Rockchip
RK3308
Technical Reference Manual
Part2

Revision 1.1
Aug. 2018

Copyright 2018 © FuZhou Rockchip Electronics Co., Ltd.

Revision History

Date	Revision	Description
2018-8-10	11	Update the discription
2018-5-20	1.0	Initial Release

Table of Content

Copyright 2018 © FuZhou Rockchip Electronics Co., Ltd.

Figure Index

Table Index

Warranty Disclaimer

Rockchip Electronics Co., Ltd makes no warranty, representation or guarantee (expressed, implied, statutory, or otherwise) by or with respect to anything in this document, and shall not be liable for any implied warranties of non-infringement, merchantability or fitness for a particular purpose or for any indirect, special or consequential damages.

Information furnished is believed to be accurate and reliable. However, Rockchip Electronics Co.,Ltd assumes no responsibility for the consequences of use of such information or for any infringement of patents or other rights of third parties that may result from its use.

Rockchip Electronics Co., Ltd's products are not designed, intended, or authorized for using as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Rockchip Electronics Co., Ltd's product could create a situation where personal injury or death may occur, should buyer purchase or use Rockchip Electronics Co., Ltd's products for any such unintended or unauthorized application, buyers shall indemnify and hold Rockchip Electronics Co., Ltd and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, expenses, and reasonable attorney fees arising out of, either directly or indirectly, any claim of personal injury or death that may be associated with such unintended or unauthorized use, even if such claim alleges that Rockchip Electronics Co., Ltd was negligent regarding the design or manufacture of the part.

Copyright and Patent Right

Information in this document is provided solely to enable system and software implementers to use Rockchip Electronics Co.,Ltd 's products. There are no expressed and patent or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Rockchip Electronics Co., Ltd does not convey any license under its copyright and patent rights nor the rights of others.
All copyright and patent rights referenced in this document belong to their respective owners and shall be subject to corresponding copyright and patent licensing requirements.

Trademarks

Rockchip and Rockchip^{TM} logo and the name of Rockchip Electronics Co., Ltd's products are trademarks of Rockchip Electronics Co., Ltd. and are exclusively owned by Rockchip Electronics Co., Ltd. References to other companies and their products use trademarks owned by the respective companies and are for reference purpose only.

Confidentiality

The information contained herein (including any attachments) is confidential. The recipient hereby acknowledges the confidentiality of this document, and except for the specific purpose, this document shall not be disclosed to any third party.

Reverse engineering or disassembly is prohibited.

ROCKCHIP ELECTRONICS CO.,LTD. RESERVES THE RIGHT TO MAKE CHANGES IN ITS PRODUCTS OR PRODUCT SPECIFICATIONS WITH THE INTENT TO IMPROVE FUNCTION OR DESIGN AT ANY TIME AND WITHOUT NOTICE AND IS NOT REQUIRED TO UNDATE THIS DOCUMENTATION TO REFLECT SUCH CHANGES.

Copyright © 2018 Rockchip Electronics Co., Ltd.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electric or mechanical, by photocopying, recording, or otherwise, without the prior written consent of Rockchip Electronics Co., Ltd.

Chapter 1 Interconnect

1.1 Overview

The chip-level interconnect consists of main interconnect. It enables communication among the modules and subsystems in the device.
The main interconnect supports the following features:

- Cross-bar exchange network
- A special internal slave for accessing the configuration register
- Little-endian platform
- Embedded memory scheduler for DDR transaction generation
- QoS management for optimizing the transaction flow
- Transaction statistics for analyzing the transaction flow
- Security protection mechanism to compatible with the TrustZone technology

1.2 Block Diagram

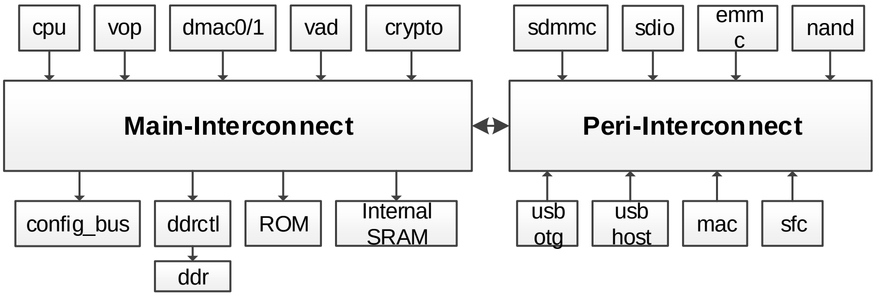


Fig. 1-1 Interconnect diagram

1.3 Function Description

1.3.1 Interconnect Services

The Interconnect supports run-time tuning, via control registers, and observed via status registers. Interconnect service networks provide access to registers.
The registers include the QosGenerator, Probe and Scheduler registers.

1.3.2 QoS(Quality of Service) management

The main interconnect is connected with all the related IPs of the system, the interface between the IP and the interconnect is called as NIU(native interface unit).
The interconnect offers 4 modes of qos management:

- None, QoSGenerator is disabled, and priority information are stuck at 0.
- Fixed, QoSGenerator drives apply a fixed urgency to read transactions, and a (possibly different) urgency to write transactions.
- Limiter, QoSGenerator behaves as in fixed mode, but limits the traffic bandwidth coming from that socket, possibly stalling requests if the initiator attempts to exceed its budget.
- Regulator, QoSGenerator promotes are demotes hurry, depending the bandwidth obtained by the initiator is below or beyond a bandwidth budget. As transactions exceeding the bandwidth limit are sent (even though demoted), the regulator mode may be considered as a softer version of the limiter mode.

Limiter Behavior
When configured in bandwidth limiter, the unit uses a 23 bit counter to measure the average bandwidth. This counter has a 1/256 byte resolution and works as follows:

- Adds the number of byte rounded up to 16 (1 -> 16) and then multiplied by 256 to the current value, each time a request is sent.
- Subtracts the Bandwidth register value every cycle. If the Counter becomes negative, force it to 0.

- If the Counter value is greater than the Saturation register value multiplied by 16*256, any incoming request is stalled until this condition disappears. Note that the Counter cannot wrap-around because the maximum value it can reach is: SaturationMax*16*256 + BurstMax*256 = 1023*4K+4K*256 = 5116K or 223 = 8192K.

The following example will show the Counter behavior: 32 byte bursts, F=400MHz, BW=200MB/s, T=0.32us. The Bandwidth register will be set to 256*200/400 = 128, and the Saturation register to 128*0.32*400/4096 = 4 (which corresponds to 64 bytes).

Regulator Behavior

When configured in bandwidth regulator, the unit uses a 23 bit counter to measure the average bandwidth. This counter has a 1/256 byte resolution and works as follows:

- Adds the number of byte rounded up to 16 and then multiplied by 256 to the current value, each time a response is received. If the result is greater than the Saturation register value multiplied by 16*256, saturation to this value is applied.
- Subtracts the Bandwidth register value every cycle. If the Counter becomes negative, force it to 0.
- If the Counter value is less than or equal to the Saturation register value multiplied by 16*256/2, the SocketMst Hurry signal will be set to the HurryHigh register, and HurryLow otherwise. Note that Urgency and Press will be also set to the same value.

The following example will show the Counter behavior: 1Kbyte bursts, F=500MHz, BW=2GB/s,T=2.048us. The Bandwidth register will be set to 256*2000/500 = 1024, and the Saturation register to 1024*2.048*500/4096 = 256 (which corresponds to 4 Kbytes).

QoS Generator Programming

Bandwidth: This log2 (socket.wData/8)+8 bits register defines the bandwidth in 1/256th byte per cycle unit. This allows a 2 MByte/s resolution at 500MHz. When thebandwidth is given in MByte/s, the value of this register will be equal to 256*BWMB/s / FMHz.

Saturation: This 10 bits register defines the number of byte used for bandwidth measurement. It is expressed in 16bytes unit (up to 16 Kbyte). Usually the integration window is given in us or in cycle: the value of this register will beequal to Bandwidh*Tus*FMHz/(256*16)or Bandwidh*Ncycle/(256*16).

The QoS management for peri-interconnect and main-interconnect inside each interconnect is independent. When master in peri-interconnect access main-interconnect, the QoS will be propagated to main-interconnect according to QoS configuration for 'peri2msch_nsp' (peri master access DDR). The masters in main-interconnect does not access peri-interconnect. The default setting of master NIUs are listing below:

Table 1-1 QoS Generator

Master NIU	Priority0/1	Register Base Address
cpu	2	0xff5c0080
vop	3	0xff5d8100
dmac0	1	0xff5d8000
dmac1	1	0xff5d8080
crypto	1	0xff5d8180
vad	3	0xff5d0080
emmc	1	0xff5e0080
mac	1	0xff5e0100
nand	1	0xff5e0180
sdio	1	0xff5e0200
sdmmc	1	0xff5e0280
sfc	1	0xff5e0300
usb_host	1	0xff5e0380
usb_otg	1	0xff5e0400

Note:

- *All master NIU QoS generator mode is 'Regulator', bandwidth is 500MB/s and saturation is 1024.*
- *The bandwidth must be calculated based on actual operating frequency.*
- *Refer to chapter 7.4 for detail register. All generators have the same register except the valid bits of 'Bandwidth' filed may be different.*

1.3.3 Memory Scheduler

Memory scheduler is a special NIU of the interconnect, it mainly deal with the transaction inside the interconnect and convert it to the transaction which the DDR protocol controller can recognize.
Following table shows the software configurable setting for the memory scheduler when the system connected to different size of DDR device.
The MSCH_DdrConf is a configurable register inside interconnect .
R: indicates Row bits
B: indicates Bank bits
C: indicates Column bits

Table 1-2 DdrConf item

0	CCCB RRRR RRRR RRRR RRRB BCCC CCCC CCCC
1	CCCR RRRR RRRR RRRR RRBB BCCC CCCC CCCC
2	CCRR RRRR RRRR RRRR RBBB CCCC CCCC CCCC
3	CCRR RCBB BRRR RRRR RRRR RCCC CCCC CCCC
4	CCRR CBBB RRRR RRRR RRRR RCCC CCCC CCCC
5	CCCR BBBR RRRR RRRR RRRR RCCC CCCC CCCC
6	CCCB BBRR RRRR RRRR RRRR RCCC CCCC CCCC
7	CCRB BBRR RRRR RRRR RRRR CCCC CCCC CCCC

In order to get best performance, some DDR controller and DRAM related timing parameters must be programmed properly in memory schedule registers.

1.3.4 Probe

The interconnect provides a service called probe to trace packet and compute traffic statics, there are totally 3 probes to monitor the memory schedule traffic statics and each can be programmed by their register. They are listed below.

Table 1-3 Probe

Probe Name	Monitor Path	Register Base Address
cpu_probe	cpu to memory schedule	0xff5c8800
bus_probe	bus to memory schedule	0xff5c8400
peri_probe	peripheral master to memory schedule	0xff5c8c00

Refer to chapter 7.4 for detail register.

1.4 Register Description

1.4.1 Internal Address Mapping

Name	Offset	Description
cpu_QosGenerator	0x0080	service_cpu + offset
dmac0_QosGenerator	0x0000	service_logic + offset
dmac1_QosGenerator	0x0080	service_logic + offset
vop_QosGenerator	0x0100	service_logic + offset
crypto_QosGenerator	0x0180	service_logic + offset
msch_Scheduler	0x0000	service_msch + offset
bus_Probe	0x0400	service_msch + offset
cpu_Probe	0x0800	service_msch + offset
peri_Probe	0x0c00	service_msch + offset
vad_QosGenerator	0x0080	service_voice + offset
emmc_QosGenerator	0x0080	service_peri + offset

Name	Offset	Description
mac_QosGenerator	0x0100	service_peri + offset
nandc_QosGenerator	0x0180	service_peri + offset
sdio_QosGenerator	0x0200	service_peri + offset
sdmmc_QosGenerator	0x0280	service_peri + offset
sfc_QosGenerator	0x0300	service_peri + offset
usbhost_QosGenerator	0x0380	service_peri + offset
usbotg_QosGenerator	0x0400	service_peri + offset

1.4.2 QoS Registers Summary

Name	Offset	Size	Reset Value	Description
QOS_Id_CoreId	0x0000	W	0xeba52304	Core id
QOS_Id_RevisionId	0x0004	W	0x00018100	Revision id
QOS_Priority	0x0008	W	0x80000202	QoS prirority
QOS_Mode	0x000c	W	0x00000003	Qos mode seletion
QOS_Bandwidth	0x0010	W	0x00000140	QoS bandwidth
QOS_Saturation	0x0014	W	0x00000040	QoS saturation
QOS_ExtControl	0x0018	W	0x00000000	Qos external control

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

1.4.3 QoS Detail Register Description

QOS_Id_CoreId
Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:8	RO	0xeba523	CoreChecksum Field containing a checksum of the parameters of the IP
7:0	RO	0x04	CoreTypeId Field identifying the type of IP

QOS_Id_RevisionId
Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:0	RO	0x00018100	RevisionId Constant

QOS_Priority
Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31	RO	0x1	Mark Backward compatibility marker when 0
30:10	RO	0x0	reserved
9:8	RW	0x2	P1 In Programmable or Bandwidth Limiter mode, the priority level for read transactions. In Bandwidth regulator mode, the priority level when the used throughput is below the threshold. In Bandwidth Regulator mode, P1 should have a value equal or greater than P0
7:2	RO	0x0	reserved
1:0	RW	0x2	P0 In Programmable or Bandwidth Limiter mode, the priority level for write transactions. In Bandwidth Regulator mode, the priority level when the used throughput is above the threshold. In Bandwidth Regulator mode, P0 should have a value equal or lower than P1

QOS_Mode

Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1:0	RW	0x3	Mode 0 = Programmable mode: a programmed priority is assigned to each read or write, 1 = Bandwidth Limiter Mode: a hard limit restricts throughput, 2 = Bypass mode: (<See SoC-specific QoS generator documentation>), 3 = Bandwidth Regulator mode: priority decreases when throughput exceeds a threshold

QOS_Bandwidth

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x140	Bandwidth In Bandwidth Limiter or Bandwidth Regulator mode, the bandwidth threshold in units of 1/256th bytes per cycle. For example, 80 MBps on a 250 MHz interface is value 0x0052. The valid bits may be different for different master NIU

QOS_Saturation

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9:0	RW	0x040	Saturation In Bandwidth Limiter or Bandwidth Regulator mode, the maximum data count value, in units of 16 bytes. This determines the window of time over which bandwidth is measured. For example, to measure bandwidth within a 1000 cycle window on a 64-bit interface is value 0x1F4

QOS_ExtControl

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	SocketQosEn Register field SocketQosEn determines how priority levels are driven when QoS generators and socket interfaces alternatively drive the levels for Urgency, Pressure, and Hurry signals: When set to 0, the QoS generator drives the levels. When set to 1, internal signals Pressure and Hurry are driven by the greater of the two levels from the socket interface or the QoS generator

1.4.4 Memory Schedule Registers Summary

Name	Offset	Size	Reset Value	Description
MSCH_Id_CoreId	0x0000	W	0xd80b4d02	Core id
MSCH_Id_RevisionId	0x0004	W	0x00018100	Revision id
MSCH_DdrConf	0x0008	W	0x00000000	Register DeviceConf stores selectors to the predefined list of DDR configurationto be programmed at initialization phase. The register has two fields, Rank0 andRank1
MSCH_DdrTiming	0x000c	W	0x1c514256	Register DdrTiming of timing register bank n stores timing settings used bymemory schedulers to compute bank and page states
MSCH_DdrMode	0x0010	W	0x00000000	Register DdrMode stores the controller behavior description
MSCH_ReadLatency	0x0014	W	0x00000028	DDR readlatency register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

Base address for memory schedule 0 is 0xffa84000 and for memory schedule 1 is 0xffa8c0000.

1.4.5 Memory schedule Detail Register Description

MSCH_Id_CoreId

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:8	RO	0xd80b4d	CoreChecksum Field containing a checksum of the parameters of the IP.
7:0	RO	0x02	CoreTypeId Field identifying the type of IP

MSCH_Id_RevisionId

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:0	RO	0x00018100	RevisionId Constant

MSCH_DdrConf

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2:0	RW	0x00	DdrConf Selection of a configuration of mappings of address bits to memory device, bank, row, and column. See the table "ddrconf" for detail.

MSCH_DdrTiming

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31	RW	0x0	BwRatio Bandwidth Ratio of the Memory Schedule and DDR If the width of DDR interface is 16bits, set BwRatio=1, and MSCH_DdrMODE. BwRatioExtended = 0 If the width of DDR interface is 8bits, set BwRatio=0, and MSCH_DdrMODE. BwRatioExtended = 1
30:26	RW	0x07	WrToRd The minimum number of scheduler clock cycles between the last DRAM Write command and a Read command. (WL x tCkD + tWTR) / tCkG
25:21	RW	0x02	RdToWr The minimum number of scheduler clock cycles between the last DRAM Read command and a Write command. DDR2: 2xtCkD / tCkG DDR3: (RL – WL +2)xtCkD / tCkG
20:18	RW	0x4	BurstLen The DRAM burst duration on the DRAM data bus in scheduler clock cycles. Also equal to scheduler clock cycles between two DRAM commands (BL/2 x tCkD) / tCkG

Bit	Attr	Reset Value	Description
17:12	RW	0x14	WrToMiss Minimum number of scheduler clock cycles between the last DRAM Write command and a new Read or Write command in another page of the same bank. The field must be set to the following value, rounded to an integer number of scheduler clock cycles: (WL ×tCkD + tWR + tRP + tRCD) / tCkG
11:6	RW	0x09	RdToMiss Minimum number of scheduler clock cycles between the last DRAM Read command and a new Read or Write command in another page of the same bank. The field must be set to the following value, rounded to an integer number of scheduler clock cycles: (tRTP + tRP + tRCD –BL ×tCkD / 2) / tCkG
5:0	RW	0x16	ActToAct Minimum number of scheduler clock cycles between two consecutive DRAM Activate commands on the same bank. The field must be set to the following value, rounded to an integer number of scheduler clock cycles: tRC / tCkG

Note: tCkG – time of noc clock cycle, tCkD – time of dram clock cycle

MSCH_DdrMode

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	BwRatioExtended Extended for MSCH_DdrTiming.BwRatio When set to 1, support for 4x Bwratio If the width of DDR interface is 16bits, set BwRatioExtended = 0, and DdrTiming .BwRatio=1 If the width of DDR interface is 8bits, set BwRatioExtended =1, and MSCH_DdrTiming.BwRatio=0
0	RW	0x0	AutoPrecharge When set to one, pages are automatically closed after each access, when set to zero, pages are left opened until an access in a different page occurs

MSCH_ReadLatency

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7:0	RW	0x0	ReadLatency The DRAM type-specific number of cycles from a scheduler request to a protocol controller response. This is a fixed value depending on the type of DRAM memory.

1.4.6 Probe Registers Summary

Name	Offset	Size	Reset Value	Description
PROBE_Id_CoreId	0x0000	W	0x54af1e06	Core id
PROBE_Id_RevisionId	0x0004	W	0x00018100	Revision id
PROBE_MainCtl	0x0008	W	0x00000000	Register MainCtl contains probe global control bits
PROBE_CfgCtl	0x000c	W	0x00000000	Register CfgCtl contains global enable and active bits. The register, which mustbe used by software before changing certain packet probe global registers
PROBE_StatPeriod	0x0024	W	0x00000000	Statistics Period
PROBE_StatGo	0x0028	W	0x00000000	Statistics begin control
PROBE_Counters_0_Src	0x0138	W	0x00000000	Register CntSrc indicates the event source used to increment the counter. Unassigned values (non-existing Press level or ExtEvent index, or unimplemented Filter) are equivalent to OFF
PROBE_Counters_0_Val	0x013c	W	0x00000000	Registers Counters_M_Val contain the statistics counter values
PROBE_Counters_1_Src	0x014c	W	0x00000000	Register CntSrc indicates the event source used to increment the counter. Unassigned values (non-existing Press level or ExtEvent index, or unimplemented Filter) are equivalent to OFF
PROBE_Counters_1_Val	0x0150	W	0x00000000	Registers Counters_M_Val contain the statistics counter values

Name	Offset	Size	Reset Value	Description
PROBE_Counters_2_Src	0x0160	W	0x00000000	Register CntSrc indicates the event source used to increment the counter. Unassigned values (non-existing Press level or ExtEvent index, or unimplemented Filter) are equivalent to OFF
PROBE_Counters_2_Val	0x0164	W	0x00000000	Registers Counters_M_Val contain the statistics counter values
PROBE_Counters_3_Src	0x0174	W	0x00000000	Register CntSrc indicates the event source used to increment the counter. Unassigned values (non-existing Press level or ExtEvent index, or unimplemented Filter) are equivalent to OFF
PROBE_Counters_3_Val	0x0178	W	0x00000000	Registers Counters_M_Val contain the statistics counter values

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

1.4.7 Probe Detail Register Description

PROBE_Id_CoreId
Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:8	RO	0x54af1e	CoreChecksum Field containing a checksum of the parameters of the IP
7:0	RO	0x06	CoreTypeId Field identifying the type of IP

PROBE_Id_RevisionId
Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:0	RO	0x00018100	RevisionId Constant

PROBE_MainCtl
Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7	RW	0x0	FiltByteAlwaysChainableEn When set to 0, filters are mapped to all statistic counters when counting bytes or enabled bytes. Therefore, only filter events mapped to even counters can be counted using a pair of chained counters.When set to 1, filters are mapped only to even statistic counters when counting bytes or enabled bytes. Thus events from any filter can be counted using a pair of chained counters
6	RO	0x0	IntrusiveMode When set to 1, register field IntrusiveMode enables trace operation in Intrusive flow-control mode. When set to 0, the register enables trace operation in Overflow flow-control mode
5	RW	0x0	StatCondDump When set, register field StatCondDump enables the dump of a statistics frame to the range of counter values set for registers StatAlarmMin, StatAlarmMax, and AlarmMode. This field also renders register StatAlarmStatus inoperative. When parameter statisticsCounterAlarm is set to False, the StatCondDump register bit is reserved
4	RW	0x0	AlarmEn When set, register field AlarmEn enables the probe to collect alarm-related information. When the register field bit is null, both TraceAlarm and StatAlarm outputs are driven to 0
3	RW	0x0	StatEn When set to 1, register field StatEn enables statistics profiling. The probe sendS statistics results to the output for signal ObsTx. All statistics counters are cleared when the StatEn bit goes from 0 to 1. When set to 0, counters are disabled
2	RW	0x0	PayloadEn Register field PayloadEn, when set to 1, enables traces to contain headers and payload. When set ot 0, only headers are reported
1	RO	0x0	TraceEn Register field TraceEn enables the probe to send filtered packets (Trace) on the ObsTx observation output
0	RW	0x0	ErrEn Register field ErrEn enables the probe to send on the ObsTx output any packet with Error status, independently of filtering mechanisms, thus constituting a simple supplementary global filter

PROBE_CfgCtl

Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved

Bit	Attr	Reset Value	Description
1	RO	0x0	Active Register field Active is used to inform software that the probe is active. Probe configuration is not allowed during the active state. This bit is raised when bit GlobalEn is set, and is cleared a few cycles after setting GlobalEn to zero (probe is Idle)
0	RW	0x0	GlobalEn Set register field GlobalEn to 1 enable the tracing and statistics collection sub-systems of the packet probe

PROBE_StatPeriod

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	StatPeriod Register StatPeriod is a 5-bit register that sets a period, within a range of 2 cycles to 2 gigacycles, during which statistics are collected before being dumped automatically. Setting the register implicitly enables automatic mode operation for statistics collection. The period is calculated with the formula: N_Cycle = 2**StatPeriodWhen register StatPeriod is set to its default value 0, automatic dump mode is disabled, and register StatGo is activated for manual mode operation. Note: When parameter statisticsCollection is set to False, StatPeriod is reserved

PROBE_StatGo

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	StatGo Writing a 1 to the 1-bit pulse register StatGo generates a statistics dump.The register is active when statistics collection operates in manual mode, that is, when register StatPeriod is set to 0.NOTE The written value is not stored in StatGo. A read always returns 0

PROBE_Counters_0_Src

Address: Operational Base + offset (0x0138)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4:0	RW	0x00	IntEvent Internal packet event 5'h00 OFF Counter disabled. 5'h01 CYCLE8 Probe clock cycles. 5'h02 IDLE Idle cycles during which no packet data is observed. 5'h03 XFER Transfer cycles during which packet data is transferred. 5'h04 BUSY Busy cycles during which the packet data is made available by the transmitting agent but the receiving agent is not ready to receive it. 5'h05 WAIT Wait cycles during a packet in which the transmitting agent suspends the transfer of packet data. 5'h06 PKT Packets. 5'h08 BYTE Total number of payload bytes. 5'h09 PRESS Clock cycles with pressure level > 0. 5'h0A PRESS Clock cycles with pressure level > 1. 5'h0B PRESS Clock cycles with pressure level > 2. 5'h10 CHAIN Carry from counter 2m to counter 2m + 1

PROBE_Counters_0_Val
Address: Operational Base + offset (0x013c)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RO	0x0000	Counters_0_Val Register Val is a read-only register that is always present. The register contains the statistics counter value either pending StatAlarm output, or when statisticscollection is suspended subsequent to triggers or signal statSuspend

PROBE_Counters_1_Src
Address: Operational Base + offset (0x014c)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4:0	RW	0x00	IntEvent Internal packet event Event | source type | Event description 5'h00 OFF Counter disabled. 5'h01 CYCLE8 Probe clock cycles. 5'h02 IDLE Idle cycles during which no packet data is observed. 5'h03 XFER Transfer cycles during which packet data is transferred. 5'h04 BUSY Busy cycles during which the packet data is made available by the transmitting agent but the receiving agent is not ready to receive it. 5'h05 WAIT Wait cycles during a packet in which the transmitting agent suspends the transfer of packet data. 5'h06 PKT Packets. 5'h08 BYTE Total number of payload bytes. 5'h09 PRESS Clock cycles with pressure level > 0. 5'h0A PRESS Clock cycles with pressure level > 1. 5'h0B PRESS Clock cycles with pressure level > 2. 5'h10 CHAIN Carry from counter 2m to counter 2m + 1

PROBE_Counters_1_Val

Address: Operational Base + offset (0x0150)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RO	0x0000	Counters_0_Val Register Val is a read-only register that is always present. The register contains the statistics counter value either pending StatAlarm output, or when statisticscollection is suspended subsequent to triggers or signal statSuspend

PROBE_Counters_2_Src

Address: Operational Base + offset (0x0160)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4:0	RW	0x00	IntEvent Internal packet event Event | source type | Event description 5'h00 OFF Counter disabled. 5'h01 CYCLE8 Probe clock cycles. 5'h02 IDLE Idle cycles during which no packet data is observed. 5'h03 XFER Transfer cycles during which packet data is transferred. 5'h04 BUSY Busy cycles during which the packet data is made available by the transmitting agent but the receiving agent is not ready to receive it. 5'h05 WAIT Wait cycles during a packet in which the transmitting agent suspends the transfer of packet data. 5'h06 PKT Packets. 5'h08 BYTE Total number of payload bytes. 5'h09 PRESS Clock cycles with pressure level > 0. 5'h0A PRESS Clock cycles with pressure level > 1. 5'h0B PRESS Clock cycles with pressure level > 2. 5'h10 CHAIN Carry from counter 2m to counter 2m + 1

PROBE_Counters_2_Val

Address: Operational Base + offset (0x0164)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RO	0x0000	Counters_0_Val Register Val is a read-only register that is always present. The register contains the statistics counter value either pending StatAlarm output, or when statisticscollection is suspended subsequent to triggers or signal statSuspend

PROBE_Counters_3_Src

Address: Operational Base + offset (0x0174)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4:0	RW	0x00	IntEvent Internal packet event Event | source type | Event description 5'h00 OFF Counter disabled. 5'h01 CYCLE8 Probe clock cycles. 5'h02 IDLE Idle cycles during which no packet data is observed. 5'h03 XFER Transfer cycles during which packet data is transferred. 5'h04 BUSY Busy cycles during which the packet data is made available by the transmitting agent but the receiving agent is not ready to receive it. 5'h05 WAIT Wait cycles during a packet in which the transmitting agent suspends the transfer of packet data. 5'h06 PKT Packets. 5'h08 BYTE Total number of payload bytes. 5'h09 PRESS Clock cycles with pressure level > 0. 5'h0A PRESS Clock cycles with pressure level > 1. 5'h0B PRESS Clock cycles with pressure level > 2. 5'h10 CHAIN Carry from counter 2m to counter 2m + 1

PROBE_Counters_3_Val
Address: Operational Base + offset (0x0178)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RO	0x0000	Counters_0_Val Register Val is a read-only register that is always present. The register contains the statistics counter value either pending StatAlarm output, or when statisticscollection is suspended subsequent to triggers or signal statSuspend

1.5 Application Notes

1.5.1 QoS setting

The VOP has the external QoS control.
It’s recommended that field 0 of QoS. ExtControl set to 1 to enable the external qos control.
And priority setting of each master kept at 1.

1.5.2 Idle request

The main interconnect supports flushing the ongoing transaction when the software needed to do so.
If the power domain need to disconnect from the main interconnect, Idle request has to be sent to NIU, the NIU will respond a ack, and when it’s ready to be disconnect, one Idle signal will be send out . Then, if the power domain still have transaction to be sent to the memory scheduler, it will be stalled by the NIU.
If the power domain is disconnected as the above flow, then CPU want to access to the power

domain, it will response error or held to CPU according to the corresponding grf register setting. The sequence is like following figure shows:

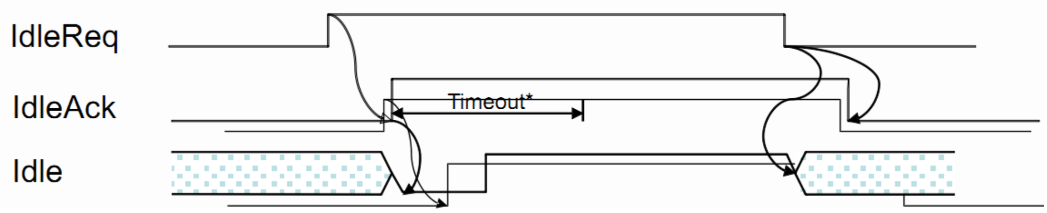


Fig. 1-2 Idle Request

The idle request is set by PMU register.

1.5.3 Basic Packet Tracing

To trace packets, the packet probe must be programmed as follows:

- Select the interesting probe listed in section 7.3.3.
- Set field *TraceEn* of register *MainCtl* to 1 to enable forwarding of traced packets to the connected observer. Optionally set field *PayloadEn* of register *MainCtl* to 1 if the packet payload should be included in the trace.
- Set field *GlobalEn* of register *CfgCtl* to 1.

1.5.4 Counting packets over a fixed period

The following programming sequence counts packets at a given probe point using statistic counter 0.

- Select the interesting probe listed in section 7.3.3
- Set field *StatEn* to 1 in register *MainCtl*.
- Set register *Counters_0_Src* to 0x6 (PKT)to count packets.
- Specify the period during which the packets should be counted by setting register StatPeriod to: $\log_2$(interval expressed in number of probe clock cycles)
- Set field *GlobalEn* of register *CfgCtl* to 1 to enable packet counting.

Once time $2^{StatPeriod}$ has elapsed, the number of packets counted is dumped to the observer and can be read from *Counters_0_Val*.

1.5.5 Measuring bandwidth

The following programming sequence example shows how a packet probe can be used to measure bandwidth at a probe point.

Some important points to note about this example are:

- Statistics counters are chained together to support the maximum theoretical bandwidth. Counter 0 is configured to count bytes; counter 1 increments when counter 0 rolls over.
- The counter values are dumped to an observer after time $2^{StatPeriod}$.

The programming sequence is as follows:

- Select the interesting probe listed in section 7.3.3
- Set register *Counters_0_Src* to 0x8 (BYTES) to count bytes.
- Set register *Counters_1_Src* to 0x10 (CHAIN) to increment when counter 0 wraps.
- Specify the period during which the bytes should be counted by setting register *StatPeriod* to: $\log_2$(interval expressed in number of probe clock cycles).
- Set field *GlobalEn* of register *CfgCtl* to 1 to enable the counting of bytes.

Once time $2^{StatPeriod}$ has elapsed, the number of packets counted is dumped to the observer and can be read from *Counters_0_Val* and *Counters_0_Val*.

Chapter 2 Dynamic Memory Controller (DMC)

2.1 Overview

The DMC includes two section, dynamic ram universal protocol controller (uPCTL) and PHY. The uPCTL SoC application bus interface supports a lowest-latency native application interface (NIF). To maximize data transfer efficiency, NIF commands transfer data without flow control. To simplify command processing, the NIF accepts addresses in rank, bank, row, column format.

The DMC supports the following features:

- Complete, integrated DDR2/DDR3/DDR3L/LPDDR2 solution
- Up to 1600 Mbps in 1:2 frequency ratio using a 400MHz controller clock and 800MHz memory clock
- Support for x8, x16 DDR2/DDR3/DDR3L/LPDDR2 memories, for a total memory data path width of 16 bits
- Up to 1 memory rank
- Transaction controllable bank management policies: open-page, close-page
- Efficient DDR protocol implementation with in-order column (Read and Write) commands and out-of-order Activate and Precharge commands
- 1T or 2T memory command timing
- Automatic power-down and self-refresh entry and exit
- Software and hardware driven self-refresh entry and exit
- Partial population of memories, where not all DDR byte lanes are populated with memory chips
- Automatic DQS gate training

2.2 Block Diagram

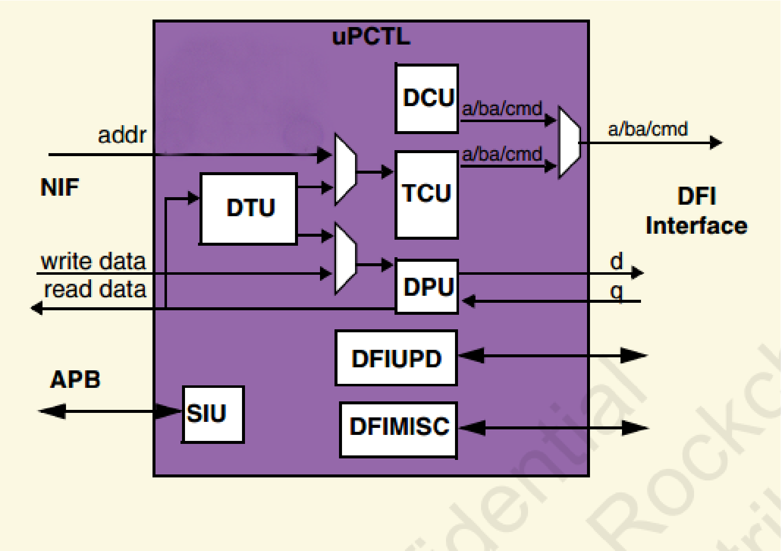


Fig. 2-1 Protocol controller architecture

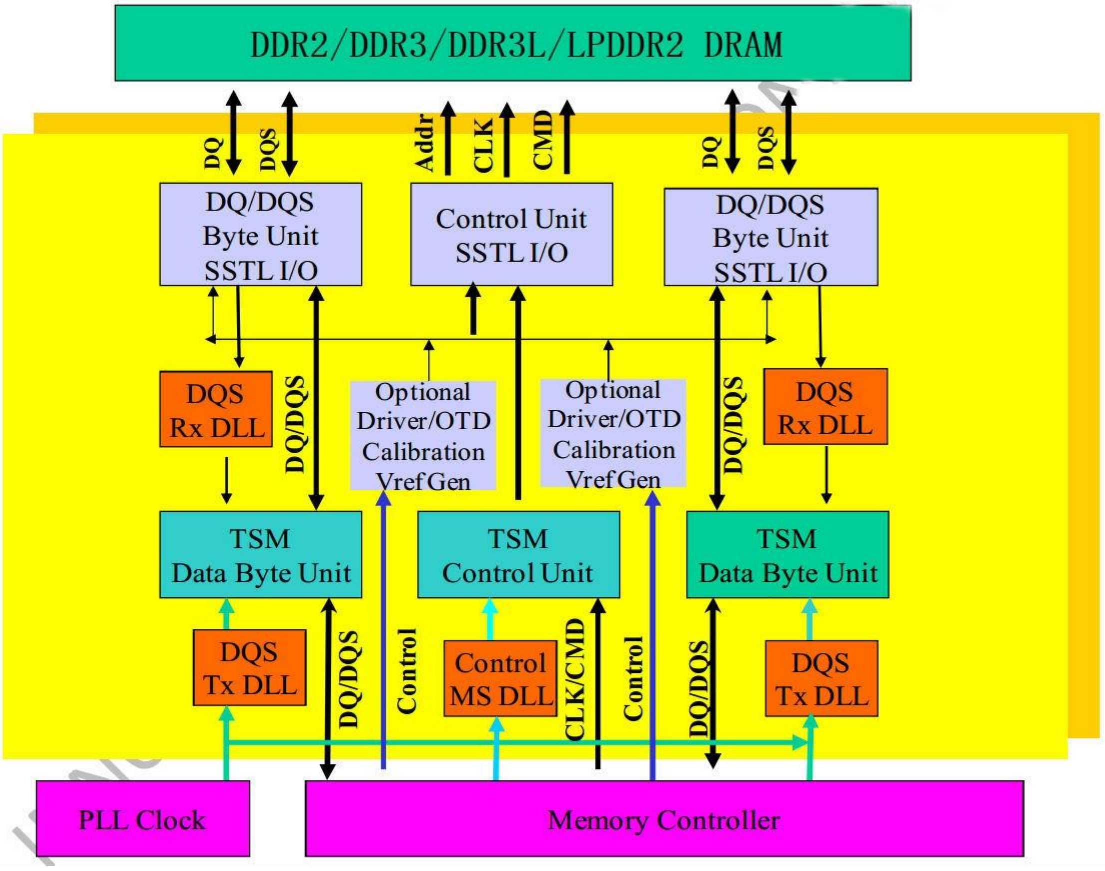


Fig. 2-2 PHY controller architecture

2.3 Function Description

2.3.1 Universal Protocol controller (uPCTL)

uPCTL operations are defined in terms of the current state of the Operational State Machine. Software can move uPCTL in any of the operational states by issuing commands via the SCTL register. Transitions from one operational state to the other occurs pass through a "transitional" state. Transitional states are exited automatically by the uPCTL after all the necessary actions required to change operational state have been completed. The current operational state of uPCTL is reported by the DDR_PCTL_STAT register and is also available from the GRF_UPCTL_STATUS0[18:16] register.

uPCTL supports the following operational states:

- Init_mem - This state is the default state entered after reset. All writable registers can be programmed. While in this state software can program uPCTL and initialize the PHY and the memories. The memories are not refreshed and data that has previously been written to the memories may be lost as a result. The Init_mem state is also used when it is desirable to stop any automatic uPCTL function that directly affects the memories, like Power Down and Refresh, or when a software reset of the memory subsystem has to be executed.
- ConFig- This state is used to suspend temporarily the normal NIF traffic and allow software to reprogram uPCTL and memories if necessary, while still keeping active the periodic generation of Refresh cycles to the memories. Power Down entry and exit sequences are possible while in ConFig state.
- Access - This is the operational state where NIF transactions are accepted by the uPCTL and converted into memory read and writes. None of the registers can be programmed except SCFG and SCTL registers.
- Low_power - Memories are in self-refresh mode. The uPCTL does not generate refresh cycles while in this state.

Access and Low_power states can be entered and exited by configuring GRF_UPCTL_CON0[4] to 1'b1. In case of conflicting software and hardware low-power commands, the resulting operational state taken by the controller can be either one of the two conflicting requests.

Following figure illustrates the operational and transitional states.

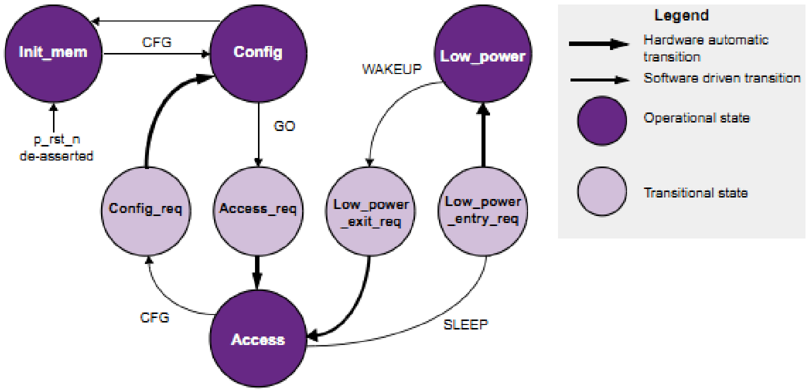


Fig. 2-3 Protocol controller architecture

2.3.2 DDR PHY

DDR PHY provides physical interface solutions for chip requiring access to DDR2/DDR3/ LPDDR2 DRAM device. It is optimized for low power and high speed applications with robust timing and small silicon area. It supports DDR2/DDR3/LPDDR2 DRAM components in the market. The PHY components contain DDR specialized functional and utility SSTL I/Os up to 800MHz, critical timing synchronization module (TSM) and a low power/jitter DLLs with programmable fine-grain control for any DRAM interface.

DDR PHY uses a DFI digital interface to connect the memory controller. DDR mux is done in the PHY block together with all related per-byte lane timing adjustment. The interface is fairly generic and support high performance input and output data flow up to 1600Mbps. With configurable timing and driving strength and ODT parameters to interface to the wide variety of DRAMs, the PHY is very flexible with advanced command capability to increase DRAM operation efficiency.

2.3.3 DDR Monitor

The DDR Monitor Module has two functions, the first function is to monitor the NIF interface to determine if read or write DDR address is within a specified range. The second function is used to monitor the DFI interface to do the statistics about DDR bandwidth and utilization.

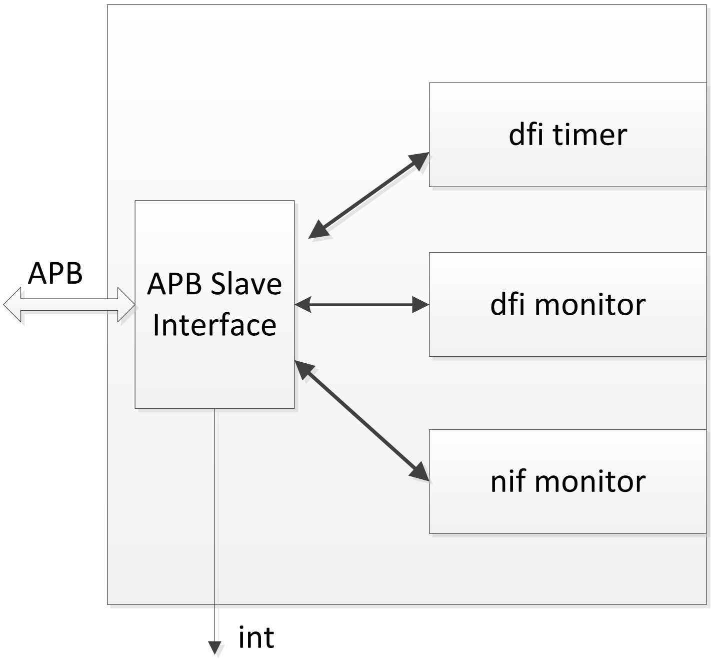


Fig. 2-4 DDR Monitor Block Diagram

DDR Monitor asserts the active-high level interrupt if correspond interrupt is enabled. The NIF monitor will monitor if there are any access to DDR while the DFI monitor do the statistics about DDR bandwidth and utilization.

2.3.4 DDR Standby

The function of DDR standby module is to set the standby mode of DDR controller, PHY and memory scheduler. The standby mode is enabled by programming DDR standby control register through APB slave interface. When there are not any bus access to DDR data and register, DDR controller will be idle. Once DDR controller is idle for a period of time, the DDR standby module will generate a low power request to DDR controller and also generate appropriate interface logic to gate the clock of DDR controller, DDR PHY and memory scheduler.

2.4 Register Description

2.4.1 Registers Summary For uPCTL

Name	Offset	Size	Reset Value	Description
DDR_PCTL_SCFG	0x0000	W	0x00000300	State Configuration Register
DDR_PCTL_SCTL	0x0004	W	0x00000000	Operational State Control Register
DDR_PCTL_STAT	0x0008	W	0x00000000	Operational State Status Register
DDR_PCTL_INTRSTAT	0x000c	W	0x00000000	Interrupt Status Register
DDR_PCTL_MCMD	0x0040	W	0x00100000	Memory Command Register
DDR_PCTL_POWCTL	0x0044	W	0x00000000	Power Up Control Register
DDR_PCTL_POWSTAT	0x0048	W	0x00000000	Power Up Status Register
DDR_PCTL_CMDTSTAT	0x004c	W	0x00000000	Command Timers Status Register
DDR_PCTL_CMDTSTATEN	0x0050	W	0x00000000	Command Timers Status Enable Register
DDR_PCTL_MRRCFG0	0x0060	W	0x00000000	Mode Register Read Configuration 0
DDR_PCTL_MRRSTAT0	0x0064	W	0x00000000	Mode Register Read Status 0 Register
DDR_PCTL_MRRSTAT1	0x0068	W	0x00000000	Mode Register Read Status 0 Register
DDR_PCTL_MCFG1	0x007c	W	0x00000000	Memory Configuration 1 Register
DDR_PCTL_MCFG	0x0080	W	0x00040020	Memory Configuration Register
DDR_PCTL_PPCFG	0x0084	W	0x00000000	Partially Populated Memories Configuration Register
DDR_PCTL_MSTAT	0x0088	W	0x00000000	Memory Status Register
DDR_PCTL_LPDDR2ZQCFG	0x008c	W	0xab0a560a	LPDDR2 ZQ Configuration Register
DDR_PCTL_DTUPDES	0x0094	W	0x00000000	DTU Status Register
DDR_PCTL_DTUNA	0x0098	W	0x00000000	DTU Number of Addresses Created Register
DDR_PCTL_DTUNE	0x009c	W	0x00000000	DTU Number of Errors Register
DDR_PCTL_DTUPRD0	0x00a0	W	0x00000000	DTU Parallel Read 0 Register
DDR_PCTL_DTUPRD1	0x00a4	W	0x00000000	DTU Parallel Read 1 Register
DDR_PCTL_DTUPRD2	0x00a8	W	0x00000000	DTU Parallel Read 2 Register
DDR_PCTL_DTUPRD3	0x00ac	W	0x00000000	DTU Parallel Read 3 Register
DDR_PCTL_DTUAWDT	0x00b0	W	0x00000290	DTU Address Width Register
DDR_PCTL_TOGCNT1U	0x00c0	W	0x00000064	Toggle Counter 1us Register

Name	Offset	Size	Reset Value	Description
DDR_PCTL_TINIT	0x00c4	W	0x000000c8	t_init Timing Register
DDR_PCTL_TRSTH	0x00c8	W	0x00000000	t_rsth Timing Register
DDR_PCTL_TOGCNT100N	0x00cc	W	0x00000001	Toggle Counter 100ns
DDR_PCTL_TREFI	0x00d0	W	0x0000004e	t_refi Timing Register
DDR_PCTL_TMRD	0x00d4	W	0x00000001	t_mrd Timing Register
DDR_PCTL_TRFC	0x00d8	W	0x00000002	t_rfc Timing Register
DDR_PCTL_TRP	0x00dc	W	0x00010006	t_trp Timing Register
DDR_PCTL_TRTW	0x00e0	W	0x00000002	t_rtw Timing Register
DDR_PCTL_TAL	0x00e4	W	0x00000000	AL Register
DDR_PCTL_TCL	0x00e8	W	0x00000004	CL Timing Register
DDR_PCTL_TCWL	0x00ec	W	0x00000003	CWL Timing Register
DDR_PCTL_TRAS	0x00f0	W	0x00000010	t_ras Timing Register
DDR_PCTL_TRC	0x00f4	W	0x00000016	t_rc Timing Register
DDR_PCTL_TRCD	0x00f8	W	0x00000006	t_rcd Timing Register
DDR_PCTL_TRRD	0x00fc	W	0x00000004	t_rrd Timing Register
DDR_PCTL_TRTP	0x0100	W	0x00000003	t_rtp Timing Register
DDR_PCTL_TWR	0x0104	W	0x00000006	t_wr Register
DDR_PCTL_TWTR	0x0108	W	0x00000004	t_wtr Timing Register
DDR_PCTL_TEXSR	0x010c	W	0x00000001	t_exsr Timing Register
DDR_PCTL_TXP	0x0110	W	0x00000001	t_xp Timing Register
DDR_PCTL_TXPDLL	0x0114	W	0x00000000	t_xpdll Timing Register
DDR_PCTL_TZQCS	0x0118	W	0x00000000	t_zqcs Timing Register
DDR_PCTL_TZQCSI	0x011c	W	0x00000000	t_zqcsi Timing Register
DDR_PCTL_TDQS	0x0120	W	0x00000001	t_dqs Timing Register
DDR_PCTL_TCKSRE	0x0124	W	0x00000000	t_cksre Timing Register
DDR_PCTL_TCKSRX	0x0128	W	0x00000000	t_cksrx Timing Register
DDR_PCTL_TCKE	0x012c	W	0x00000003	t_cke Timing Register
DDR_PCTL_TMOD	0x0130	W	0x00000000	t_mod Timing Register
DDR_PCTL_TRSTL	0x0134	W	0x00000000	Reset Low Timing Register
DDR_PCTL_TZQCL	0x0138	W	0x00000000	t_zqcl Timing Register
DDR_PCTL_TMRR	0x013c	W	0x00000002	t_mrr Timing Register
DDR_PCTL_TCKESR	0x0140	W	0x00000004	t_ckesr Timing Register
DDR_PCTL_TDPD	0x0144	W	0x00000000	t_dpd Timing Register
DDR_PCTL_TREFI_MEM_DDR3	0x0148	W	0x00000000	t_refi_mem_ddr3 Timing Register
DDR_PCTL_DTUWACTL	0x0200	W	0x00000000	DTU Write Address Control
DDR_PCTL_DTURACTL	0x0204	W	0x00000000	DTU Read Address Control Register
DDR_PCTL_DTUCFG	0x0208	W	0x00000000	DTU Configuration Control Register
DDR_PCTL_DTUECTL	0x020c	W	0x00000000	DTU Execute Control Register
DDR_PCTL_DTUWD0	0x0210	W	0x00000000	DTU Write Data #0 Register
DDR_PCTL_DTUWD1	0x0214	W	0x00000000	DTU Write Data #1 Register

Name	Offset	Size	Reset Value	Description
DDR_PCTL_DTUWD2	0x0218	W	0x00000000	DTU Write Data #2 Register
DDR_PCTL_DTUWD3	0x021c	W	0x00000000	DTU Write Data #3 Register
DDR_PCTL_DTUWDM	0x0220	W	0x00000000	DTU Write Data Mask Register
DDR_PCTL_DTURD0	0x0224	W	0x00000000	DTU Read Data #0 Register
DDR_PCTL_DTURD1	0x0228	W	0x00000000	DTU Read Data #1 Register
DDR_PCTL_DTURD2	0x022c	W	0x00000000	DTU Read Data #2 Register
DDR_PCTL_DTURD3	0x0230	W	0x00000000	DTU Read Data #3 Register
DDR_PCTL_DTULFSRWD	0x0234	W	0x00000000	DTU LFSR Seed for Write Data Generation Register
DDR_PCTL_DTULFSRRD	0x0238	W	0x00000000	DTU LFSR Seed for Read Data Generation Register
DDR_PCTL_DTUEAF	0x023c	W	0x00000000	DTU Error Address FIFO Register
DDR_PCTL_DFITCTRLDELAY	0x0240	W	0x00000002	DFI tctrl_delay Register
DDR_PCTL_DFIODTCFG	0x0244	W	0x00000000	DFI ODT Configuration
DDR_PCTL_DFIODTCFG1	0x0248	W	0x06060000	DFI ODT Timing Configuration 1 (for Latency and Length)
DDR_PCTL_DFIODTRANKMAP	0x024c	W	0x00008421	DFI ODT Rank Mapping
DDR_PCTL_DFITPHYWRDATA	0x0250	W	0x00000001	DFI tphy_wrdata Register
DDR_PCTL_DFITPHYWRLAT	0x0254	W	0x00000001	DFI tphy_wrlat Register
DDR_PCTL_DFITPHYWRDATALAT	0x0258	W	0x00000000	DFI tphy_wrdata_lat Register
DDR_PCTL_DFITRDDATAEN	0x0260	W	0x00000001	DFI trddata_en Register
DDR_PCTL_DFITPHYRDLAT	0x0264	W	0x0000000f	DFI tphy_rdlat Register
DDR_PCTL_DFITPHYUPDTYPE0	0x0270	W	0x00000010	DFI tphyupd_type0 Register
DDR_PCTL_DFITPHYUPDTYPE1	0x0274	W	0x00000010	DFI tphyupd_type1 Register
DDR_PCTL_DFITPHYUPDTYPE2	0x0278	W	0x00000010	DFI tphyupd_type2 Register
DDR_PCTL_DFITPHYUPDTYPE3	0x027c	W	0x00000010	DFI tphyupd_type3 Register
DDR_PCTL_DFITCTRLUPDMIN	0x0280	W	0x00000010	DFI tctrlupd_min Register
DDR_PCTL_DFITCTRLUPDMAX	0x0284	W	0x00000040	DFI tctrlupd_max Register
DDR_PCTL_DFITCTRLUPDDLY	0x0288	W	0x00000008	DFI tctrlupddly Register

Name	Offset	Size	Reset Value	Description
DDR_PCTL_DFIUPDCFG	0x0290	W	0x00000003	DFI Update Configuration Register
DDR_PCTL_DFITREFMSKI	0x0294	W	0x00000000	DFI Masked Refresh Interval
DDR_PCTL_DFITCTRLUPDI	0x0298	W	0x00000000	DFI tctrlupd_interval Register
DDR_PCTL_DFITRCFG0	0x02ac	W	0x00000000	DFI Training Configuration 0 Register
DDR_PCTL_DFITRSTAT0	0x02b0	W	0x00000000	DFI Training Status 0 Register
DDR_PCTL_DFITRWRLVLEN	0x02b4	W	0x00000000	DFI Training dfi_wrlvl_en Register
DDR_PCTL_DFITRRDLVLEN	0x02b8	W	0x00000000	DFI Training dfi_rdlvl_en Register
DDR_PCTL_DFITRRDLVLGATEEN	0x02bc	W	0x00000000	DFI Training dfi_rdlvl_gate_en Register
DDR_PCTL_DFISTSTAT0	0x02c0	W	0x00000000	DFI Status Status 0 Register
DDR_PCTL_DFISTCFG0	0x02c4	W	0x00000000	DFI Status Configuration 0 Register
DDR_PCTL_DFISTCFG1	0x02c8	W	0x00000000	DFI Status Configuration 1 Register
DDR_PCTL_DFITDRAMCLKEN	0x02d0	W	0x00000002	DFI tdram_clk_enable Register
DDR_PCTL_DFITDRAMCLKDIS	0x02d4	W	0x00000002	DFI tdram_clk_disable Register
DDR_PCTL_DFISTCFG2	0x02d8	W	0x00000000	DFI Status Configuration 2 Register
DDR_PCTL_DFISTPARCLR	0x02dc	W	0x00000000	DFI Status Parity Clear Register
DDR_PCTL_DFISTPARLOG	0x02e0	W	0x00000000	DFI Status Parity Log Register
DDR_PCTL_DFILPCFG0	0x02f0	W	0x00070000	DFI Low Power Configuration 0 Register
DDR_PCTL_DFITRWRLVLRESP0	0x0300	W	0x00000000	DFI Training dfi_wrlvl_resp Status 0 Register
DDR_PCTL_DFITRWRLVLRESP1	0x0304	W	0x00000000	DFI Training dfi_wrlvl_resp Status 1 Register
DDR_PCTL_DFITRWRLVLRESP2	0x0308	W	0x00000000	DFI Training dfi_wrlvl_resp Status 2 Register
DDR_PCTL_DFITRRDLVLRESP0	0x030c	W	0x00000000	DFI Training dfi_rdlvl_resp Status 0 Register
DDR_PCTL_DFITRRDLVLRESP1	0x0310	W	0x00000000	DFI Training dfi_rdlvl_resp Status 1 Register
DDR_PCTL_DFITRRDLVLRESP2	0x0314	W	0x00000000	DFI Training dfi_rdlvl_resp Status 2 Register
DDR_PCTL_DFITRWRLVLDELAY0	0x0318	W	0x00000000	DFI Training dfi_wrlvl_delay Configuration 0 Register

Name	Offset	Size	Reset Value	Description
DDR_PCTL_DFITRWRLVLDELAY1	0x031c	W	0x00000000	DFI Training dfi_wrlvl_delay Configuration 1 Register
DDR_PCTL_DFITRWRLVLDELAY2	0x0320	W	0x00000000	DFI Training dfi_wrlvl_delay Configuration 2 Register
DDR_PCTL_DFITRRDLVLDELAY0	0x0324	W	0x00000000	DFI Training dfi_rdlvl_delay Configuration 0 Register
DDR_PCTL_DFITRRDLVLDELAY1	0x0328	W	0x00000000	DFI Training dfi_rdlvl_delay Configuration 1 Register
DDR_PCTL_DFITRRDLVLDELAY2	0x032c	W	0x00000000	DFI Training dfi_rdlvl_delay Configuration 2 Register
DDR_PCTL_DFITRRDLVLGATEDELAY0	0x0330	W	0x00000000	DFI Training dfi_rdlvl_gate_delay Configuration 0
DDR_PCTL_DFITRRDLVLGATEDELAY1	0x0334	W	0x00000000	DFI Training dfi_rdlvl_gate_delay Configuration 1
DDR_PCTL_DFITRRDLVLGATEDELAY2	0x0338	W	0x00000000	DFI Training dfi_rdlvl_gate_delay Configuration 2
DDR_PCTL_DFITRCMD	0x033c	W	0x00000000	DFI Training Command Register
DDR_PCTL_IPVR	0x03f8	W	0x00000000	IP Version Register
DDR_PCTL_IPTR	0x03fc	W	0x44574300	IP Type Register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

2.4.2 Detail Register Description For uPCTL

DDR_PCTL_SCFG

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:8	RW	0x3	bbflags_timing The n_bbflags is a NIF output vector which provides combined information about the status of each memory bank. The de-assertion is based on when precharge, activates, reads/writes are scheduled by the TCU block. It may be possible to de-assert n_bbflags earlier than calculated by the TCU block. Programming bbflags_timing is used to achieve this. The maximum recommended value is: UPCTL_TCU_SED_P - TRP.t_rp. The programmed value is the maximum number of "early" cycles that n_bbflags maybe de-asserted. The actual achieved de-assertion depends on the traffic profile. In 1:2 mode the maximum allowed programmable value is 4'b0111 In 1:1 mode the value can be 4'b1111
7	RW	0x0	ac_pdd_en Enables assertion of ac_pdd to indicate to the PHY of an opportunity to switch to a Low power mode

Bit	Attr	Reset Value	Description
6	RW	0x0	nfifo_nif1_dis For Synopsys internal use only for NFIFO testing. 1'b0: Only supported setting 1'b1: For Synopsys internal use only
5:1	RO	0x0	reserved
0	RW	0x0	hw_low_power_en Enables the hardware low-power interface. Allows the system to request via c_sysreq(controlled by GRF_UPCTL_CON0[5] or GRF_UPCTL_CON0[6]) to enter the memories into Self-Refresh. The handshaking between the request and acknowledge hardware low power signals (c_sysreq and c_sysack, respectively) is always performed, but the uPCTL response depends on the value set on this register field and by the value driven on the c_active_in(controlled by GRF_UPCTL_CON0[4]) input pin. 1'b0: Disabled. Requests are always denied and uPCTL is unaffected by c_sysreq 1'b1: Enabled. Requests are accepted or denied, depending on the current operational state of uPCTL and on the value of c_active_in.

DDR_PCTL_SCTL

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2:0	RW	0x0	state_cmd Issues an operational state transition request to the uPCTL. 3'b000: INIT (move to Init_mem from Config) 3'b001: CFG (move to Config from Init_mem or Access) 3'b010: GO (move to Access from Config) 3'b011: SLEEP (move to Low_power from Access) 3'b100: WAKEUP (move to Access from Low_power) Others: Reserved

DDR_PCTL_STAT

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6:4	RO	0x0	lp_trig Reports the status of what triggered an entry to Low_power state. Is only set if in Low_power state. The individual bits report the following: - lp_trig[2]: Software driven due to SCTL.state_cmd==SLEEP. - lp_trig[1]: Hardware driven due to Hardware Low Power Interface. - lp_trig[0]: Hardware driven due to Auto Self Refresh (MCFG1.sr_idle>0). Note, if more than one trigger happens at the exact same time, more than one bit of lp_trig may be asserted high.
3	RO	0x0	reserved
2:0	RO	0x0	ctl_stat Returns the current operational state of the uPCTL. 3'b000: Init_mem 3'b001: Config 3'b010: Config_req 3'b011: Access 3'b100: Access_req 3'b101: Low_power 3'b110: Low_power_entry_req 3'b111: Low_power_exit_req Others: Reserved

DDR_PCTL_INTRSTAT

Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RO	0x0	parity_intr Indicates that a DFI parity error has been detected 1'b0: No error 1'b1: Parity error
0	RO	0x0	ecc_intr Indicates that an ECC error has been detected 1'b0: No error 1'b1: Parity error

DDR_PCTL_MCMD

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31	R/W SC	0x0	start_cmd Start command. When this bit is set to 1, the command operation defined in the cmd_opcode field is started. This bit is automatically cleared by the uPCTL after the command is finished. The application can poll this bit to determine when uPCTL is ready to accept another command. This bit cannot be cleared to 1'b0 by software.
30:28	RO	0x0	reserved
27:24	RW	0x0	cmd_add_del Set the additional delay associated with each command to 2^n internal timers clock cycles, where n is the bit field value. If n=0, the delay is 0. Max value is n=10.
23:20	RW	0x1	rank_sel Rank select for the command to be executed. 4'b0001: Rank 0 4'b0010: Rank 1 4'b0100: Rank 2 4'b1000: Rank 3 4'b0000: Reserved Multiple 1'b1s in rank_sel mean multiple ranks are selected, which is useful broadcasting commands in parallel to multiple ranks during initialization and configuration of the memories. If MCMD.cmd_opcode=RSTL, all ranks should be selected as it cannot be performed to individual ranks
19:17	RW	0x0	bank_addr refer to cmd_addr field description.

Bit	Attr	Reset Value	Description
16:4	RW	0x0000	cmd_addr If LPDDR2/LPDDR3, cmd_addr field and bank_addr field are merged into one field - lpddr23_addr: lpddr23_addr[19:4]: Mode register address and value driven on the relevant memory address bits ([19:4]), during a Mode Register Write operation, defined by cmd_opcode=MRS or a Mode Register Read operation, defined by cmd_opcode=MRR. For other values of cmd_opcode, this field is ignored. bits [19:12] correspond to OP7 .. OP0 and bits [11:4] correspond to MA7 .. MA0. If mDDR/DDR2/DDR3, consider as two separate register fields - bank_addr and cmd_addr: bank_addr[19:17]: Mode Register address driven on the memory bank address bits, BA1, BA0, during a Mode Register Set operation, defined by cmd_opcode=MRS. For other values of cmd_opcode, this field is ignored. 3'b000: MR0 (MR in DDR2) 3'b001: MR1 (EMR in DDR2) 3'b010: MR2 (EMR(2) in DDR2) 3'b011: MR3 (EMR(3) in DDR2) Others: Reserved cmd_addr[16:14]: Mode Register value driven on the memory address bits, A12 to A0, during a Mode Register Set operation defined by cmd_opcode=MRS. For other values of cmd_opcode this field is ignored. Refer to the memory specification for the correct settings of the various bits of this field during a MRS operation.

Bit	Attr	Reset Value	Description
3:0	RW	0x0	cmd_opcode Command to be issued to the memory. 4'b0000: Deselect. This is only used for timing purposes, no actual direct Deselect command is passed to the memories. 4'b0001: Precharge All (PREA) 4'b0010: Refresh (REF) 4'b0011: Mode Register Set (MRS) - is MRW in LPDDR2/LPDDR3, MRS otherwise 4'b0100: ZQ Calibration Short (ZQCS, only applies to LPDDR2/LPDDR3/DDR3) 4'b0101: ZQ Calibration Long (ZQCL, only applies to LPDDR2/LPDDR3/DDR3) 4'b0110: Software Driven Reset (RSTL, only applies to DDR3) 4'b0111: Reserved 4'b1000: Mode Register Read (MRR) - is MRR in LPDDR2/LPDDR3, is SRR in mDRR and is MPR in DDR3 4'b1001: Deep Power Down Entry (DPDE, only applies to mDDR/LPDDR2/LPDDR3) 4'b1010: DFI Controller Update (DFICTRLUPD) Others - Reserved It is recommended that DPDE and RSTL commands using the MCMD register should not be performed in Config State as automatic Refresh logic runs in Config state and may cause these commands to operate incorrectly.

DDR_PCTL_POWCTL

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	R/W SC	0x0	power_up_start Start the memory power up sequence. When this bit is set to 1'b1, uPCTL starts the CKE and RESET# power up sequence to the memories. This bit is automatically cleared by uPCTL after the sequence is completed. This bit cannot be cleared to 1'b0 by software.

DDR_PCTL_POWSTAT

Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	power_up_done Returns the status of the memory power-up sequence. 1'b0: Power-up sequence has not been performed. 1'b1: Power-up sequence has been performed.

DDR_PCTL_CMDTSTAT

Address: Operational Base + offset (0x004c)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	cmd_tstat Returns the status of the timers for memory commands. This ANDs all the command timers together. 1'b0: One or more command timers has not expired. 1'b1: All command timers have expired.

DDR_PCTL_CMDTSTATEN

Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	cmd_tstat_en Enables the generation of the status of the timers for memory commands. Is enabled before CMDTSTAT register is read. 1'b0: Disabled 1'b1: Enabled

DDR_PCTL_MRRCFG0

Address: Operational Base + offset (0x0060)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x0	mrr_byte_sel Selects which byte's data to store when performing an MRR command via MCMD. LegalValues: 0 .. 8

DDR_PCTL_MRRSTAT0

Address: Operational Base + offset (0x0064)

Bit	Attr	Reset Value	Description
31:24	RO	0x00	mrrstat_beat3 MRR/MPR read data beat 3
23:16	RO	0x00	mrrstat_beat2 MRR/MPR read data beat 2
15:8	RO	0x00	mrrstat_beat1 MRR/MPR read data beat 1
7:0	RO	0x00	mrrstat_beat0 MRR/MPR read data beat 0

DDR_PCTL_MRRSTAT1

Address: Operational Base + offset (0x0068)

Bit	Attr	Reset Value	Description
31:24	RO	0x00	mrrstat_beat7 MRR/MPR read data beat 7

Bit	Attr	Reset Value	Description
23:16	RO	0x00	mrrstat_beat6 MRR/MPR read data beat 6
15:8	RO	0x00	mrrstat_beat5 MRR/MPR read data beat 5
7:0	RO	0x00	mrrstat_beat4 MRR/MPR read data beat 4

DDR_PCTL_MCFG1

Address: Operational Base + offset (0x007c)

Bit	Attr	Reset Value	Description
31	RW	0x0	hw_exit_idle_en When this bit is programmed to 1'b1 the c_active_in pin can be used to exit from the automatic clock stop , power down or self-refresh modes.
30:25	RO	0x0	reserved
24	RW	0x0	zq_resistor_shared - 1 Denotes that ZQ resistor is shared between ranks. Means ZQinit/ZQCL/ZQCS commands are sent to one rank at a time with tZQinit/tZQCL/tZQCS timing met between commands so that commands to different ranks do not overlap. - 0 ZQ resistor is not shared.
23:16	RW	0x00	hw_idle Hardware idle period. The c_active output is driven high if the NIF is idle in Access state for hw_idle * 32 * n_clk cycles. The hardware idle function is disabled when hw_idle=0.
15:11	RO	0x0	reserved
10:8	RW	0x0	tfaw_cfg_offset Used to fine tune the tFAW setting as programmed in MCFG.tfaw_cfg tFAW= (4 + MCFG.tfaw_cfg)*tRRD - tfaw_cfg_offset
7:0	RW	0x00	sr_idle Self Refresh idle period. Memories are placed into Self-Refresh mode if the NIF is idle in Access state for sr_idle * 32 * n_clk cycles. The automatic self refresh function is disabled when sr_idle=0.

DDR_PCTL_MCFG

Address: Operational Base + offset (0x0080)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	mddr_lpddr23_clock_stop_idle Clock stop idle period in n_clk cycles. Memories are placed into clock stop mode if the NIF is idle for mddr_lpddr2_clkstop_idle n_clk cycles. The automatic clock stop function is disabled when mddr_lpddr2_clkstop_idle=0. Clock stop mode is only applicable in mDDR/LPDDR2/LPDDR3.
23:22	RW	0x0	mddr_lpddr23_en mDDR/LPDDR2 Enable. Enables support for mDDR or LPDDR2 or LPDDR3. 2'b00: mDDR/LPDDR2/LPDDR3 Disabled 2'b01: LPDDR3 Enabled 2'b10: mDDR Enabled 2'b11: LPDDR2 Enabled
21:20	RW	0x0	mddr_lpddr23_bl mDDR/LPDDR2 Burst Length. The BL setting must be consistent with the value programmed into the BL field of MR. 2'b00: BL2, Burst length of 2 (MR.BL=3'b001, mDDR only) 2'b01: BL4, Burst length of 4 (MR.BL=3'b010, for mDDR and LPDDR2 only) 2'b10: BL8, Burst length of 8 (MR.BL=3'b011, for mDDR, LPDDR2 and LPDDR3) 2'b11: BL16, Burst length of 16 (MR.BL=3'b100, for mDDR and LPDDR2 only) This value is effective only if MCFG.mddr_lpddr23_en[1:0]!=2'b00. Otherwise, MCFG.mem_bl is used to define uPCTL’s Burst Length (for DDR2/DDR3).
19:18	RW	0x1	tfaw_cfg Sets tFAW to be 4, 5 or 6 times tRRD. 2'b00: set tFAW=4*tRRD 2'b01: set tFAW=5*tRRD 2'b10: set tFAW=6*tRRD
17	RW	0x0	pd_exit_mode Selects the mode for Power Down Exit. For DDR2/DDR3, the power down exit mode setting in uPCTL must be consistent with the value programmed into the power down exit mode bit of MR0. For mDDR/LPDDR2/LPDDR3, only fast exit mode is valid. 1'b0: slow exit 1'b1: fast exit
16	RW	0x0	pd_type Sets the Power down type. 1'b0: Precharge Power Down 1'b1: Active Power Down

Bit	Attr	Reset Value	Description
15:8	RW	0x00	pd_idle Power-down idle period in n_clk cycles. Memories are placed into power-down mode if the NIF is idle for pd_idle n_clk cycles. The automatic power down function is disabled when pd_idle=0.
7	RW	0x0	mddr_lpddr2_bst_even Ensures BST commands only placed an even no. of cycles from previous column command (Read or Write). 1'b0: Disabled 1'b1: Enabled If using the "Gen 2 DDR MultiPHYs" in LPDDR2/LPDDR3, set this to 1'b1. Otherwise, set it to 1'b0.
6	RW	0x0	lpddr2_s4 Enables LPDDR2-S4 support. 1'b0: LPDDR2-S4 disabled (LPDDR2-S2 enabled) 1'b1: LPDDR2-S4 enabled
5	RW	0x1	ddr3_en Select DDR2 or DDR3 protocol. Ignored, if mDDR or LPDDR2 or LPDDR3 support is enabled. 1'b0: DDR2 Protocol Rules 1'b1: DDR3 Protocol Rules
4	RW	0x0	stagger_cs For multi-rank commands from the DCU, stagger the assertion of CS_N to odd and even ranks by one n_clk cycle. This is useful when using RDIMMs, when multi-rank commands may be interpreted as writes to control words in the register chip. 1'b0: Do not stagger CS_N 1'b1: Stagger CS_N
3	RW	0x0	two_t_en Enables 2T timing for memory commands. 1'b0: Disabled 1'b1: Enabled
2	RW	0x0	bl8int_en Setting this bit enables the BL8 interrupt function of DDR2. This is the capability to early terminate a BL8 after only 4 DDR beats by issuing the next command two cycles earlier. This functionality is only available for DDR2 memories and this setting is ignored for mDDR/LPDDR2/LPDDR3 and DDR3. 1'b0: Disabled 1'b1: Enabled

Bit	Attr	Reset Value	Description
1	RW	0x0	cke_or_en This bit is intended to be set for 4-rank RDIMMs, which have a 2-bit CKE input. If set, dfi_cke[0] is asserted to enable either of the even ranks (0 and 2), while dfi_cke[1] is asserted to enable either of the odd ranks (1 and 3). dfi_cke[3:2] are inactive (0) 1'b0: Disabled 1'b1: Enabled
0	RW	0x0	mem_bl DDR Burst Length. The BL setting in DDR2 / DDR3 must be consistent with the value programmed into the BL field of MR0. 1'b0: BL4, Burst length of 4 (MR0.BL=3'b010, DDR2 only) 1'b1: BL8, Burst length of 8 (MR0.BL=3'b011 for DDR2, MR0.BL=2'b00 for DDR3)

DDR_PCTL_PPCFG

Address: Operational Base + offset (0x0084)

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8:1	RW	0x00	rpmem_dis Reduced Population Disable bits. Setting these bits disables the corresponding NIF/DDR data lanes from writing or reading data. Lane 0 is always present, hence only 8 bits are required for the remaining lanes including the ECC lane. In 1:2 mode bit 0 of rpmem_dis covers n_wdata/n_rdata/m_ctl_d/m_phy_q[63:32], bit 1 [95:64] etc. In 1:1 mode bit 0 of rpmem_dis covers n_wdata/n_rdata/m_ctl_d/m_phy_q[31:16], bit 2 [47:32] etc. There are no restrictions on which byte lanes can be disabled, other than byte lane 0 is required. Gaps between enabled byte lanes are allowed. For each bit: 1'b0: lane exists 1'b1: lane is disabled
0	RW	0x0	ppmem_en Partially Population Enable bit. Setting this bit enables the partial population of external memories where the entire application bus is routed to a reduced size memory system. The lower half of the DRAM data bus, bit 0 up to bit UPCTL_M_DW/2-1, is the active portion when Partially Populated memories are enabled. An example of this is shown in Example 8-1 on page 263. 1'b0: Disabled 1'b1: Enabled

DDR_PCTL_MSTAT

Address: Operational Base + offset (0x0088)

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	RO	0x0	self_refresh Indicates if uPCTL, through auto self refresh, has placed the memories in Self Refresh. 1'b0: Memory is not in Self Refresh 1'b1: Memory is in Self Refresh
1	RO	0x0	clock_stop Indicates if uPCTL has placed the memories in Clock Stop. 1'b0: Memory is not in Clock Stop 1'b1: Memory is in Clock Stop
0	RO	0x0	power_down Indicates if uPCTL has placed the memories in Power Down. 1'b0: Memory is not in Power Down 1'b1: Memory is in Power-Down

DDR_PCTL_LPDDR2ZQCFG

Address: Operational Base + offset (0x008c)

Bit	Attr	Reset Value	Description
31:24	RW	0xab	zqcl_op Value to drive on memory address bits [19:12] for an automatic hardware generated ZQCL command (LPDDR2/LPDDR3). Corresponds to OP7 .. OP0 of Mode Register Write (MRW) command which is used to send ZQCL command to memory.
23:16	RW	0x0a	zqcl_ma Value to drive on memory address bits [11:4] for an automatic hardware generated ZQCL command (LPDDR2/LPDDR3). Corresponds to MA7 ..MA0 of Mode Register Write (MRW) command which is used to send ZQCL command to memory.
15:8	RW	0x56	zqcs_op Value to drive on memory address bits [19:12] for an automatic hardware generated ZQCS command (LPDDR2/LPDDR3). Corresponds to OP7 ..OP0 of Mode Register Write (MRW) command which is used to send ZQCS command to memory.
7:0	RW	0x0a	zqcs_ma Value to drive on memory address bits [11:4] for an automatic hardware generated ZQCS command (LPDDR2/LPDDR3). Corresponds to MA7 ..MA0 of Mode Register Write (MRW) command which is used to send ZQCS command to memory.

DDR_PCTL_DTUPDES

Address: Operational Base + offset (0x0094)

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved

Bit	Attr	Reset Value	Description
13	RO	0x0	dtu_rd_missing Indicates if one or more read beats of data did not return from memory.
12:9	RO	0x0	dtu_eaffl Indicates the number of entries in the FIFO that is holding the log of error addresses for data comparison
8	RO	0x0	dtu_random_error Indicates that the random data generated had some failures when written and read to the memories
7	RO	0x0	dtu_err_b7 Detected at least 1 bit error for bit 7 in the programmable data buffers
6	RO	0x0	dtu_err_b6 Detected at least 1 bit error for bit 6 in the programmable data buffers
5	RO	0x0	dtu_err_b5 Detected at least 1 bit error for bit 5 in the programmable data buffers
4	RO	0x0	dtu_err_b4 Detected at least 1 bit error for bit 4 in the programmable data buffers
3	RO	0x0	dtu_err_b3 Detected at least 1 bit error for bit 3 in the programmable data buffers
2	RO	0x0	dtu_err_b2 Detected at least 1 bit error for bit 2 in the programmable data buffers
1	RO	0x0	dtu_err_b1 Detected at least 1 bit error for bit 1 in the programmable data buffers
0	RO	0x0	dtu_err_b0 Detected at least 1 bit error for bit 0 in the programmable data buffers

DDR_PCTL_DTUNA

Address: Operational Base + offset (0x0098)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dtu_num_address Indicates the number of addresses that were created on the NIF interface during random data generation.

DDR_PCTL_DTUNE

Address: Operational Base + offset (0x009c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dtu_num_errors Indicates the number of errors that were detected on the readback of the NIF data during random data generation.

DDR_PCTL_DTUPRD0

Address: Operational Base + offset (0x00a0)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	dtu_allbits_1 Allows all the bit ones from each of the 16 received read bytes to be read in parallel. Used as part of read data eye training where a transition is required to be monitored to train the eye.
15:0	RO	0x0000	dtu_allbits_0 Allows all the bit zeros from each of the 16 received read bytes to be read in parallel. Used as part of read data eye training where a transition is required to be monitored to train the eye.

DDR_PCTL_DTUPRD1

Address: Operational Base + offset (0x00a4)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	dtu_allbits_3 Allows all the bit threes from each of the 16 received read bytes to be read in parallel. Used as part of read data eye training where a transition is required to be monitored to train the eye.
15:0	RO	0x0000	dtu_allbits_2 Allows all the bit twos from each of the 16 received read bytes to be read in parallel. Used as part of read data eye training where a transition is required to be monitored to train the eye.

DDR_PCTL_DTUPRD2

Address: Operational Base + offset (0x00a8)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	dtu_allbits_5 Allows all the bit fives from each of the 16 received read bytes to be read in parallel. Used as part of read data eye training where a transition is required to be monitored to train the eye.
15:0	RO	0x0000	dtu_allbits_4 Allows all the bit fours from each of the 16 received read bytes to be read in parallel. Used as part of read data eye training where a transition is required to be monitored to train the eye.

DDR_PCTL_DTUPRD3

Address: Operational Base + offset (0x00ac)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	dtu_allbits_7 Allows all the bit sevens from each of the 16 received read bytes to be read in parallel. Used as part of read data eye training where a transition is required to be monitored to train the eye.
15:0	RO	0x0000	dtu_allbits_6 Allows all the bit sixes from each of the 16 received read bytes to be read in parallel. Used as part of read data eye training where a transition is required to be monitored to train the eye.

DDR_PCTL_DTUAWDT

Address: Operational Base + offset (0x00b0)

Bit	Attr	Reset Value	Description
31:11	RO	0x0	reserved
10:9	RW	0x1	number_ranks Number of supported memory ranks. 2'b00: 1 rank 2'b01: 2 ranks 2'b10: 3 ranks 2'b11: 4 ranks
8	RO	0x0	reserved
7:6	RW	0x2	row_addr_width Width of the memory row address bits. 2'b00: 13 bits wide 2'b01: 14 bits wide 2'b10: 15 bits wide 2'b11: 16 bits wide
5	RO	0x0	reserved
4:3	RW	0x2	bank_addr_width Width of the memory bank address bits. 2'b00: 2 bits wide (4 banks) 2'b01: 3 bits wide (8 banks) Others: Reserved
2	RO	0x0	reserved
1:0	RW	0x0	column_addr_width Width of the memory column address bits. 2'b00: 7 bits wide 2'b01: 8 bits wide 2'b10: 9 bits wide 2'b11: 10 bits wide

DDR_PCTL_TOGCNT1U

Address: Operational Base + offset (0x00c0)

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9:0	RW	0x064	toggle_counter_1u The number of internal timers clock cycles

DDR_PCTL_TINIT

Address: Operational Base + offset (0x00c4)

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8:0	RW	0x0c8	t_init Defines the time period (in us) to hold dfi_cke and dfi_reset_n stable during the memory power up sequence. The value programmed must correspond to at least 200us. The actual time period defined is TINIT * TOGCNT1U * internal timers clock .period

DDR_PCTL_TRSTH

Address: Operational Base + offset (0x00c8)

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9:0	RW	0x000	t_rsth Defines the time period (in us) to hold the dfi_reset_n signal high after it is de-asserted during the DDR3 Power Up/Reset sequence. The value programmed for DDR3 must correspond to minimum 500us of delay. For mDDR and DDR2, this register should be programmed to 0.The actual time period defined is TRSTH * TOGCNT1U * internal timers clock period.

DDR_PCTL_TOGCNT100N

Address: Operational Base + offset (0x00cc)

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6:0	RW	0x01	toggle_counter_100n The number of internal timers clock cycles.

DDR_PCTL_TREFI

Address: Operational Base + offset (0x00d0)

Bit	Attr	Reset Value	Description
31	R/W SC	0x0	upd_ref Update Refresh counter values. Write 1'b1 to this bit whenever num_add_ref and/or t_refi register bit fields are updated. When this is set to 1'b1, the internal counters for Refresh interval and for how many refreshes performed each refresh interval get updated to use latest values of t_refi and num_add_ref respectively. It is cleared to 1'b0 automatically by hardware within a couple of clock cycles. This bit cannot be cleared to 1'b0 by software. If not set to 1'b1 when num_add_ref/t_refi are changed, then the previous values are continued to be used by the internal logic.
30:19	RO	0x0	reserved
18:16	RW	0x0	num_add_ref Number of additional Refreshes per t_refi interval. 0: 1 refresh performed every t_refi interval 1: 2 refreshes performed every t_refi interval 2: 3 refreshes performed every t_refi interval 3: 4 refreshes performed every t_refi interval 4: 5 refreshes performed every t_refi interval 5: 6 refreshes performed every t_refi interval 6: 7 refreshes performed every t_refi interval 7: 8 refreshes performed every t_refi interval For mDDR, max supported value is 6, as tRFCmax is 8*tREFI.
15:13	RO	0x0	reserved
12:0	RW	0x004e	t_refi Defines the time period (in 100ns units) of the Refresh interval. The actual time period defined is TREFI * TOGCNT100N * internal timers clock period.

DDR_PCTL_TMRD

Address: Operational Base + offset (0x00d4)

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved

Bit	Attr	Reset Value	Description
2:0	RW	0x1	t_mrd Mode Register Set command cycle time in memory clock cycles. mDDR: Time from MRS to any valid command. LPDDR2/LPDDR3: Time from MRS (MRW) to any valid command. DDR2: Time from MRS to any valid command. DDR3: Time from MRS to MRS command. mDDR Legal Values: 2 LPDDR2 Legal Values: 5 LPDDR3 Legal Values: 5 DDR2 Legal Values: 2..3 DDR3 Legal Values: 2..4

DDR_PCTL_TRFC

Address: Operational Base + offset (0x00d8)

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8:0	RW	0x002	t_rfc Refresh to Active/Refresh command time in memory clock cycles. mDDR Legal Values: 7..28 LPDDR2 Legal Values: 15..112 LPDDR3 Legal Values: 22..168 DDR2 Legal Values: 15..131 DDR3 Legal Values: 36.. 374

DDR_PCTL_TRP

Address: Operational Base + offset (0x00dc)

Bit	Attr	Reset Value	Description
31:18	RO	0x0	reserved
17:16	RW	0x1	prea_extra Additional cycles required for a Precharge All (PREA) command - in addition to t_rp. In terms of memory clock cycles mDDR Value: 0 LPDDR2 Value: Value that corresponds (tRPab -tRPpb). Rounded up in terms of memory clock cycles. Values can be 0, 1, 2. LPDDR3 Value: Value that corresponds (tRPab -tRPpb). Rounded up in terms of memory clock cycles. Values can be 0, 1, 2, 3. DDR2 Value: 1 if 8 Banks, 0 otherwise DDR3 Value: 0
15:5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4:0	RW	0x06	t_rp Precharge period in memory clock cycles. For LPDDR2/LPDDR3, this should be set to TRPpb. mDDR Legal Values: 2..3 LPDDR2 Legal Values: 3..13 LPDDR3 Legal Values: 3..20 DDR2 Legal Values: 3..7 DDR3 Legal Values: 5..14

DDR_PCTL_TRTW

Address: Operational Base + offset (0x00e0)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x2	t_rtw Read to Write turnaround time in memory clock cycle. mDDR Legal Values: 3..11 LPDDR2 Legal Values: 1..11 LPDDR3 Legal Values: 1..12 DDR2 Legal Values: 2..10 DDR3 Legal Values: 2..10

DDR_PCTL_TAL

Address: Operational Base + offset (0x00e4)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x0	t_al Additive Latency in memory clock cycles. For DDR2 this must match the value programmed into the AL field of MR1. For DDR3 this must be 0, CL-1, CL-2 depending weather the AL value in MR1 is 0,1, or 2 respectively. CL is the CAS latency programmed into MR0. For mDDR, LPDDR2 and LPDDR3, there is no AL field in the mode registers, and this setting should be set to 0 mDDR Legal Values: 0 LPDDR2 Legal Values: 0 LPDDR3 Legal Values: 0 DDR2 Legal Values: AL DDR3 Legal Values: 0, CL-1, CL-2 (depending on AL=0,1,2 in MR1)

DDR_PCTL_TCL

Address: Operational Base + offset (0x00e8)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x4	t_cl CAS Latency in memory clock cycles. If mDDR/DDR2/DDR3, the uPCTL setting must match the value programmed into the CL field of MR0. If LPDDR2/LPDDR3, the uPCTL setting must match RL (Read Latency), where RL is the value programmed into the "RL & WL"field of MR2 mDDR/DDR2/3 Legal Value: CL LPDDR2/LPDDR3 Legal Value: RL

DDR_PCTL_TCWL
Address: Operational Base + offset (0x00ec)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x3	t_cwl CAS Write Latency in memory clock cycles. For mDDR, the setting must be 1. For LPDDR2/LPDDR3 the setting must match WL (Write Latency), where WL is the value programmed into the "RL & WL"field of MR2. For DDR2 the setting must match CL-1, where CL is the value programmed into the CL field of MR0. For DDR3, the setting must match the value programmed in the memory CWL field of MR2. mDDR Legal Value: 1 LPDDR2/LPDDR3 Legal Values: WL DDR2 Legal Value: CL-1 DDR3 Legal Value: CWL

DDR_PCTL_TRAS
Address: Operational Base + offset (0x00f0)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x10	t_ras Activate to Precharge command time in memory clock cycles. mDDR Legal Values: 4..8 LPDDR2 Legal Values: 7..23 LPDDR3 Legal Values: 7..34 DDR2 Legal Values: 8..24 DDR3 Legal Values: 15..38

DDR_PCTL_TRC
Address: Operational Base + offset (0x00f4)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x16	t_rc Row Cycle time in memory clock cycles. Specifies the minimum Activate to Activate distance for accesses to same bank. mDDR Legal Values: 5..11 LPDDR2 Legal Values: 10..36 LPDDR3 Legal Values: 11..53 DDR2 Legal Values: 11..31 DDR3 Legal Values: 20..52

DDR_PCTL_TRCD

Address: Operational Base + offset (0x00f8)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x06	t_rcd Row to Column delay in memory clock cycles. Specifies the minimum Activate to Column distance. mDDR Legal Values: 2..3 LPDDR2 Legal Values: 3..13 LPDDR3 Legal Values: 4..20 DDR2 Legal Values: 3..7 DDR3 Legal Values: 5..14

DDR_PCTL_TRRD

Address: Operational Base + offset (0x00fc)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x4	t_rrd Row-to-Row delay in memory clock cycles. Specifies the minimum Activate-to-Activate distance for consecutive accesses to different banks in the same rank. mDDR Legal Values: 1..2 LPDDR2 Legal Values: 2..6 LPDDR3 Legal Values: 2..8 DDR2 Legal Values: 2..6 DDR3 Legal Values: 4..8

DDR_PCTL_TRTP

Address: Operational Base + offset (0x0100)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x3	t_rtp Read to Precharge time in memory clock cycles. Specifies the minimum distance Read to Precharge for consecutive accesses to same bank. mDDR Value: 0 LPDDR2 Legal Values: 2..4 LPDDR3 Legal Values: 2..6 DDR2 Legal Values: 2..4 DDR3 Legal Values: 3..8

DDR_PCTL_TWR

Address: Operational Base + offset (0x0104)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x06	t_wr Write recovery time in memory clock cycles. When using close page the uPCTL setting must be consistent with the WR field setting of MR0. mDDR Legal Values: 2..3 LPDDR2 Legal Values: 3..8 LPDDR3 Legal Values: 3..12 DDR2 Legal Values: 3..8 DDR3 Legal Values: 6..16

DDR_PCTL_TWTR

Address: Operational Base + offset (0x0108)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x4	t_wtr Write to Read turnaround time, in memory clock cycles. mDDR Legal Values: 1..2 LPDDR2 Legal Values: 2..4 LPDDR3 Legal Values: 4..6 DDR2 Legal Values: 2..4 DDR3 Legal Values: 3..8

DDR_PCTL_TEXSR

Address: Operational Base + offset (0x010c)

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9:0	RW	0x001	t_exsr Exit Self Refresh to first valid command delay, in memory clock cycles. For mDDR, this should be programmed to match tXSR. For LPDDR2/LPDDR3, this should be programmed to match tXSR. For DDR2, this should be programmed to match tXSRD (SRE to read-related command) as defined by the memory device specification. For DDR3, this should be programmed to match tXSDLL (SRE to a command requiring DLL locked) as defined by the memory device specification. mDDR Legal Values: 17..40 LPDDR2 Legal Values: 17..117 LPDDR3 Legal Values: 24..176 DDR2 Typical Value: 200 DDR3 Typical Value: 512

DDR_PCTL_TXP

Address: Operational Base + offset (0x0110)

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2:0	RW	0x1	t_xp Exit Power Down to first valid command delay when DLL is on (fast exit), measured in memory clock cycles. Legal Values: 1..7

DDR_PCTL_TXPDLL

Address: Operational Base + offset (0x0114)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x00	t_xpdll Exit Power Down to first valid command delay when DLL is off (slow exit), measured in memory clock cycles. mDDR/LPDDR2/LPDDR3 Value: 0 DDR2/DDR3 Legal Values: 3..63

DDR_PCTL_TZQCS

Address: Operational Base + offset (0x0118)

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6:0	RW	0x00	t_zqcs DRAM ZQ Calibration Short period, in memory clock cycles. Should be programmed to match the tZQCS timing value as defined in the memory specification. mDDR Value: 0 LPDDR2 Legal Values: 15..48 LPDDR3 Legal Values: 15..72 DDR2 Value: 0 DDR3 Typical Value: 64

DDR_PCTL_TZQCSI
Address: Operational Base + offset (0x011c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	t_zqcsi W DRAM ZQCS interval, measured in Refresh interval units. The total time period defined is TZQCSI*TREFI * TOGCNT100N * internal timers clock period. Programming a value of 0 in t_zqcsi disables the auto-ZQCS functionality in uPCTL. mDDR Value: 0 LPDDR2 Legal Values: 0..4294967295 LPDDR3 Legal Values: 0..4294967295 DDR2 Value: 0 DDR3 Legal Values: 0..4294967295 DDR3 Legal Values: 0..4294967295

DDR_PCTL_TDQS
Address: Operational Base + offset (0x0120)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x1	t_dqs Additional data turnaround time in memory clock cycles for accesses to different ranks. Used to increase the distance between column commands to different ranks, allowing more tolerance as the driver source changes on the bidirectional DQS and/or DQ signals. mDDR Legal Values: 1..12 LPDDR2 Legal Values: 1..12 LPDDR3 Legal Values: 1..12 DDR2 Legal Values: 1..12 DDR3 Legal Values: 1..12

DDR_PCTL_TCKSRE
Address: Operational Base + offset (0x0124)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	t_cksre This is the time after Self Refresh Entry that CKE is held high before going low. In memory clock cycles. Specifies the clock disable delay after SRE. In DDR3, this should be programmed to match the greatest value between 10ns and 5 memory clock periods. In LPDDR3, this should be programmed to 2. mDDR Value: 0 LPDDR2 Value: 0 LPDDR3 Value: 2 DDR2 Value: 0 DDR3 Legal Values: 5..15

DDR_PCTL_TCKSRX

Address: Operational Base + offset (0x0128)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	t_cksrx This is the time (before Self Refresh Exit) that CKE is maintained high before issuing SRX. In memory clock cycles. Specifies the clock stable time before SRX. In DDR3, This should be programmed to match the greatest value between 10ns and 5 memory clock periods. In LPDDR2/LPDDR3, this should be programmed to 2. mDDR Value: 0 LPDDR2 Value: 2 LPDDR3 Legal Values: 2 DDR2 Value: 0 DDR3 Legal Values: 5..15

DDR_PCTL_TCKE

Address: Operational Base + offset (0x012c)

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved

Bit	Attr	Reset Value	Description
2:0	RW	0x3	t_cke CKE minimum pulse width in memory clock cycles. mDDR Legal Value: 2 LPDDR2 Legal Values: 3 LPDDR3 Legal Values: 3 DDR2 Legal Value: 3 DDR3 Legal Values: 3..6

DDR_PCTL_TMOD

Address: Operational Base + offset (0x0130)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	t_mod In DDR3 mode, this is the time from MRS to any valid non-MRS command (except DESELECT or NOP) in memory clock cycles. Default Value: 0 mDDR Value: 0 LPDDR2 Value: 0 LPDDR3 Legal Values: 0 DDR2 Value: 0 DDR3 Legal Values: 0..31

DDR_PCTL_TRSTL

Address: Operational Base + offset (0x0134)

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6:0	RW	0x00	t_rstl Memory Reset Low time, in memory clock cycles. Defines the time period to hold dfi_reset_n signal low during a software driven DDR3 Reset Operation. The value programmed must correspond to at least 100ns of delay. mDDR Value: 0 LPDDR2 Value: 0 LPDDR3 Value: 0 DDR2 Value: 0 DDR3 Legal Values: 1..127

DDR_PCTL_TZQCL

Address: Operational Base + offset (0x0138)

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9:0	RW	0x000	t_zqcl DRAM ZQ Calibration Long period in memory clock cycles. If LPDDR2/LPDDR3, should be programmed to tZQCL. If DDR3, should be programmed to match the memory tZQinit timing value for the first ZQCL command during memory initialization; should be programmed to match tZQoper timing value after reset and initialization. mDDR Value: 0 LPDDR2 Legal Values: 60..192 LPDDR3 Legal Values: 60..288 DDR2 Value: 0 DDR3 Legal Values: 0..1023

DDR_PCTL_TMRR
Address: Operational Base + offset (0x013c)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x02	t_mrr Time for a Mode Register Read (MRR command from MCMD).

DDR_PCTL_TCKESR
Address: Operational Base + offset (0x0140)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x4	t_ckesr Minimum CKE low width for Self Refresh entry to exit timing in memory clock cycles. Recommended settings: - mDDR: t_ckesr=0 - LPDDR2/LPDDR3: t_ckesr=tCKESR setting from memories, rounded up in terms of memory cycles. - DDR2: t_ckesr=0 - DDR3: t_ckesr=t_cke + 1 mDDR Value: 0 LPDDR2 Legal Values: 3..8 LPDDR3 Legal Values: 3..12 DDR2 Value: 0 DDR3 Legal Values: 4..7

DDR_PCTL_TDPD
Address: Operational Base + offset (0x0144)

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9:0	RW	0x000	t_dpd Minimum Deep Power Down time. Is in terms of us. When a MCMD.DPDE command occurs, TDPD time is waited before MCMD.start_cmd can be cleared. MCMD_cmd_add_del (if any) does not start until TDPD has completed. This ensures TDPD requirement for the memory is not violated. The actual time period defined is TDPD* TOGCNT1U * internal timers clock period. Only applies for mDDR and LPDDR2 as Deep Power Down (DPD) is only valid for these memory types. For mDDR, tDPD=0, while for LPDDR2, tDPD=500 us. For LPDDR2/LPDDR3, if 500 us is waited externally by system, then set tDPD=0. mDDR Value: 0 LPDDR2 Legal Values: 0 or 500 LPDDR3 Legal Values: 0 or 500 DDR2 Legal Value: 0 DDR3 Legal Values: 0

DDR_PCTL_TREFI_MEM_DDR3

Address: Operational Base + offset (0x0148)

Bit	Attr	Reset Value	Description
31:15	RO	0x0	reserved
14:0	RW	0x0000	t_refi_mem_ddr3 DDR3 tREFI (7.8us/3.9us) time, in memory clock cycles. In DDR3 mode the value used to ensure that a maximum of 16 Refresh commands can be issued within 2 x tREFI, which is a DDR3 restriction mDDR Value: 0 LPDDR2 Values: 0 LPDDR3 Values: 0 DDR2 Value: 0

DDR_PCTL_DTUWACTL

Address: Operational Base + offset (0x0200)

Bit	Attr	Reset Value	Description
31:30	RW	0x0	dtu_wr_rank Write rank to where data is to be targeted
29	RO	0x0	reserved
28:13	RW	0x0000	dtu_wr_row Write row to where data is to be targeted
12:10	RW	0x0	dtu_wr_bank Write bank to where data is to be targeted
9:0	RW	0x000	dtu_wr_col FWrite column to where data is to be targeted

DDR_PCTL_DTURACTL

Address: Operational Base + offset (0x0204)

Bit	Attr	Reset Value	Description
31:30	RW	0x0	dtu_rd_rank Read rank from where data comes
29	RO	0x0	reserved
28:13	RW	0x0000	dtu_rd_row Read row from where data comes
12:10	RW	0x0	dtu_rd_bank Read bank from where data comes
9:0	RW	0x000	dtu_rd_col Read column from where data comes

DDR_PCTL_DTUCFG

Address: Operational Base + offset (0x0208)

Bit	Attr	Reset Value	Description
31:23	RO	0x0	reserved
22:16	RW	0x00	dtu_row_increments Number of times to increment the row address when generating random data, up to a maximum of 127 times.
15	RW	0x0	dtu_wr_multi_rd When set puts the DTU into write once multiple reads mode.
14	RW	0x0	dtu_data_mask_en Controls whether random generated data masks are transmitted. Unless enabled all data bytes are written to memory and expected to be read from memory.
13:10	RW	0x0	dtu_target_lane Selects one of the byte lanes for data comparison into the programmable read data buffer.
9	RW	0x0	dtu_generate_random Generate transfers using random data, otherwise generate transfers from the programmable write data buffers.

Bit	Attr	Reset Value	Description
8	RW	0x0	dtu_incr_banks When the column address rolls over increment the bank address until we reach and conclude bank 7.
7	RW	0x0	dtu_incr_cols Increment the column address until we saturate. Return to zero if DTUCFG.dtu_incr_banks is set to 1 and we are not at bank 7.
6:1	RW	0x00	dtu_nalen Length of the NIF transfer sequence that is passed through the uPCTL for each created address.
0	RW	0x0	dtu_enable When set, allows the DTU module to take ownership of the NIF interface: 1: DTU enabled 0: DTU disabled

DDR_PCTL_DTUECTL
Address: Operational Base + offset (0x020c)

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	R/W SC	0x0	wr_multi_rd_rst When set, resets the DTU in write once multiple reads mode, to allow a new write to be performed. This bit automatically clears.
1	R/W SC	0x0	run_error_reports When set, initiates the calculation of the error status bits. This bit automatically clears when the re-calculation is done. This is only used in debug mode to verify the comparison logic.
0	R/W SC	0x0	run_dtu When set, initiates the running of the DTU read and write transfer. This bit automatically clears when the transfers are completed

DDR_PCTL_DTUWD0
Address: Operational Base + offset (0x0210)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	dtu_wr_byte3 Write data byte
23:16	RW	0x00	dtu_wr_byte2 Write data byte
15:8	RW	0x00	dtu_wr_byte1 Write data byte
7:0	RW	0x00	dtu_wr_byte0 Write data byte

DDR_PCTL_DTUWD1
Address: Operational Base + offset (0x0214)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	dtu_wr_byte7 Write data byte
23:16	RW	0x00	dtu_wr_byte6 Write data byte
15:8	RW	0x00	dtu_wr_byte5 Write data byte
7:0	RW	0x00	dtu_wr_byte4 Write data byte

DDR_PCTL_DTUWD2
Address: Operational Base + offset (0x0218)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	dtu_wr_byte11 Write data byte
23:16	RW	0x00	dtu_wr_byte10 Write data byte
15:8	RW	0x00	dtu_wr_byte9 Write data byte
7:0	RW	0x00	dtu_wr_byte8 Write data byte

DDR_PCTL_DTUWD3
Address: Operational Base + offset (0x021c)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	dtu_wr_byte15 Write data byte
23:16	RW	0x00	dtu_wr_byte14 Write data byte
15:8	RW	0x00	dtu_wr_byte13 Write data byte
7:0	RW	0x00	dtu_wr_byte12 Write data byte

DDR_PCTL_DTUWDM
Address: Operational Base + offset (0x0220)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	dm_wr_byte0 Write data mask bit, one bit for each byte. Each bit should be 0 for a byte lane that contains valid write data.

DDR_PCTL_DTURD0
Address: Operational Base + offset (0x0224)

Bit	Attr	Reset Value	Description
31:24	RO	0x00	dtu_rd_byte3 Read byte
23:16	RO	0x00	dtu_rd_byte2 Read byte
15:8	RO	0x00	dtu_rd_byte1 Read byte
7:0	RO	0x00	dtu_rd_byte0 Read byte

DDR_PCTL_DTURD1

Address: Operational Base + offset (0x0228)

Bit	Attr	Reset Value	Description
31:24	RO	0x00	dtu_rd_byte7 Read byte
23:16	RO	0x00	dtu_rd_byte6 Read byte
15:8	RO	0x00	dtu_rd_byte5 Read byte
7:0	RO	0x00	dtu_rd_byte4 Read byte

DDR_PCTL_DTURD2

Address: Operational Base + offset (0x022c)

Bit	Attr	Reset Value	Description
31:24	RO	0x00	dtu_rd_byte11 Read byte
23:16	RO	0x00	dtu_rd_byte10 Read byte
15:8	RO	0x00	dtu_rd_byte9 Read byte
7:0	RO	0x00	dtu_rd_byte8 Read byte

DDR_PCTL_DTURD3

Address: Operational Base + offset (0x0230)

Bit	Attr	Reset Value	Description
31:24	RO	0x00	dtu_rd_byte15 Read byte
23:16	RO	0x00	dtu_rd_byte14 Read byte
15:8	RO	0x00	dtu_rd_byte13 Read byte
7:0	RO	0x00	dtu_rd_byte12 Read byte

DDR_PCTL_DTULFSRWD
Address: Operational Base + offset (0x0234)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dtu_lfsr_wseed This is the initial seed for the random write data generation LFSR (linear feedback shift register), shared with the write mask generation.

DDR_PCTL_DTULFSRRD
Address: Operational Base + offset (0x0238)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dtu_lfsr_rseed This is the initial seed for the random read data generation LFSR (linear feedback shift register), this is shared with the read mask generation. The read data mask is reconstructed the same as the write data mask was created, allowing the "on the fly comparison" ignore bytes which were not written.

DDR_PCTL_DTUEAF
Address: Operational Base + offset (0x023c)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	ea_rank Indicates the rank that the error occurred in during random data generation. There could be a number of entries in this FIFO. If FIFO is empty one reads zeroes.
29	RO	0x0	reserved
28:13	RO	0x0000	ea_row Indicates the row that the error occurred in during random data generation. There could be a number of entries in this FIFO. If FIFO is empty one reads zeroes.
12:10	RO	0x0	ea_bank Indicates the bank that the error occurred in during random data generation. There could be a number of entries in this FIFO. If FIFO is empty one reads zeroes
9:0	RO	0x000	ea_column Indicates the column address that the error occurred in during random data generation. There could be a number of entries in this FIFO. If FIFO is empty one reads zeroes.

DDR_PCTL_DFITCTRLDELAY
Address: Operational Base + offset (0x0240)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x2	tctrl_delay Specifies the number of DFI clock cycles after an assertion or desertion of the DFI control signals that the control signals at the PHY-DRAM interface reflect the assertion or de-assertion. If the DFI clock and the memory clock are not phase-aligned, this timing parameter should be rounded up to the next integer value.

DDR_PCTL_DFIODTCFG

Address: Operational Base + offset (0x0244)

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28	RW	0x0	rank3_odt_default Default ODT value of rank 3 when there is no read/write activity
27	RW	0x0	rank3_odt_write_sel Enable/disable ODT for rank 3 when a write access is occurring on this rank
26	RW	0x0	rank3_odt_write_nse Enable/disable ODT for rank 3 when a write access is occurring on a different rank
25	RW	0x0	rank3_odt_read_sel Enable/disable ODT for rank 3 when a read access is occurring on this rank
24	RW	0x0	rank3_odt_read_nsel Enable/disable ODT for rank 3 when a read access is occurring on a different rank
23:21	RO	0x0	reserved
20	RW	0x0	rank2_odt_default Default ODT value of rank 2 when there is no read/write activity
19	RW	0x0	rank2_odt_write_sel Enable/disable ODT for rank 2 when a write access is occurring on this rank
18	RW	0x0	rank2_odt_write_nse Enable/disable ODT for rank 2 when a write access is occurring on a different rank
17	RW	0x0	rank2_odt_read_sel Enable/disable ODT for rank 2 when a read access is occurring on this rank
16	RW	0x0	rank2_odt_read_nsel Enable/disable ODT for rank 2 when a read access is occurring on a different rank
15:13	RO	0x0	reserved
12	RW	0x0	rank1_odt_default Default ODT value of rank 1 when there is no read/write activity

Bit	Attr	Reset Value	Description
11	RW	0x0	rank1_odt_write_sel Enable/disable ODT for rank 1 when a write access is occurring on this rank
10	RW	0x0	rank1_odt_write_nse Enable/disable ODT for rank 1 when a write access is occurring on a different rank
9	RW	0x0	rank1_odt_read_sel Enable/disable ODT for rank 1 when a read access is occurring on this rank
8	RW	0x0	rank1_odt_read_nsel Enable/disable ODT for rank 1 when a read access is occurring on a different rank
7:5	RO	0x0	reserved
4	RW	0x0	rank0_odt_default Default ODT value of rank 0 when there is no read/write activity
3	RW	0x0	rank0_odt_write_sel Enable/disable ODT for rank 0 when a write access is occurring on this rank
2	RW	0x0	rank0_odt_write_nse Enable/disable ODT for rank 0 when a write access is occurring on a different rank
1	RW	0x0	rank0_odt_read_sel Enable/disable ODT for rank 0 when a read access is occurring on this rank
0	RW	0x0	rank0_odt_read_nsel Enable/disable ODT for rank 0 when a read access is occurring on a different rank

DDR_PCTL_DFIODTCFG1

Address: Operational Base + offset (0x0248)

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26:24	RW	0x6	odt_len_bl8_r ODT length for BL8 read transfers Length of dfi_odt signal for BL8 reads. This is in terms of SDR cycles. For BL4 reads, the length of dfi_odt is always 2 cycles shorter than the value in this register field.
23:19	RO	0x0	reserved

Bit	Attr	Reset Value	Description
18:16	RW	0x6	odt_len_bl8_w ODT length for BL8 write transfers Length of dfi_odt signal for BL8 writes. This is in terms of SDR cycles. For BL4 writes, the length of dfi_odt is always 2 cycles shorter than the value in this register field.
15:13	RO	0x0	reserved
12:8	RW	0x00	odt_lat_r ODT latency for reads Latency after a read command that dfi_odt is set. This is in terms of SDR cycles.
7:5	RO	0x0	reserved
4:0	RW	0x00	odt_lat_w ODT latency for writes Latency after a write command that dfi_odt is set. This is in terms of SDR cycles

DDR_PCTL_DFIODTRANKMAP

Address: Operational Base + offset (0x024c)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:12	RW	0x8	odt_rank_map3 Rank mapping for dfi_odt[3] Determines whether dfi_odt[3] should be asserted when the uPCTL requires to terminate each rank Bit 15=1: dfi_odt[3] will be asserted to terminate rank 3 Bit 14=1: dfi_odt[3] will be asserted to terminate rank 2 Bit 13=1: dfi_odt[3] will be asserted to terminate rank 1 Bit 12=1: dfi_odt[3] will be asserted to terminate rank 0 This field exists only if UPCTL_M_NRANKS: 4
11:8	RW	0x4	odt_rank_map2 Rank mapping for dfi_odt[2] Determines which rank access(es) will cause dfi_odt[2] to be asserted Bit 11=1: dfi_odt[2] will be asserted to terminate rank 3 Bit 10=1: dfi_odt[2] will be asserted to terminate rank 2 Bit 9=1: dfi_odt[2] will be asserted to terminate rank 1 Bit 8=1: dfi_odt[2] will be asserted to terminate rank 0 This field exists only if UPCTL_M_NRANKS: 4

Bit	Attr	Reset Value	Description
7:4	RW	0x2	odt_rank_map1 Rank mapping for dfi_odt[1] Determines which rank access(es) will cause dfi_odt[1] to be asserted Bit 7=1: dfi_odt[1] will be asserted to terminate rank 3 Bit 6=1: dfi_odt[1] will be asserted to terminate rank 2 Bit 5=1: dfi_odt[1] will be asserted to terminate rank 1 Bit 4=1: dfi_odt[1] will be asserted to terminate rank 0 This field exists only if UPCTL_M_NRANKS >
3:0	RW	0x1	odt_rank_map0 Rank mapping for dfi_odt[0] Determines which rank access(es) will cause dfi_odt[0] to be asserted Bit 3=1: dfi_odt[0] will be asserted to terminate rank 3 Bit 2=1: dfi_odt[0] will be asserted to terminate rank 2 Bit 1=1: dfi_odt[0] will be asserted to terminate rank 1 Bit 0=1: dfi_odt[0] will be asserted to terminate rank 0

DDR_PCTL_DFITPHYWRDATA

Address: Operational Base + offset (0x0250)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x01	tphy_wrdata Specifies the number of DFI clock cycles between when the dfi_wrdata_en signal is asserted to when the associated write data is driven on the dfi_wrdata signal. This has no impact on performance, only adjusts the relative time between enable and data transfer.

DDR_PCTL_DFITPHYWRLAT

Address: Operational Base + offset (0x0254)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x01	tphy_wrlat Specifies the number of DFI clock cycles between when a write command is sent on the DFI control interface and when the dfi_wrdata_en signal is asserted.

DDR_PCTL_DFITPHYWRDATALAT

Address: Operational Base + offset (0x0258)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x00	tphy_wrdata_lat Specifies the number of DFI clock cycles of latency of the DFI write data through the PHY

DDR_PCTL_DFITRDDATAEN

Address: Operational Base + offset (0x0260)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x01	trddata_en Specifies the number of DFI clock cycles from the assertion of a read command on the DFI to the assertion of the dfi_rddata_en signal.

DDR_PCTL_DFITPHYRDLAT

Address: Operational Base + offset (0x0264)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x0f	tphy_rdlat Specifies the maximum number of DFI clock cycles allowed from the assertion of the dfi_rddata_en signal to the assertion of the dfi_rddata_valid signal.

DDR_PCTL_DFITPHYUPDTYPE0

Address: Operational Base + offset (0x0270)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x010	tphyupd_type0 Specifies the maximum number of DFI clock cycles that the dfi_phyupd_req signal may remain asserted after the assertion of the dfi_phyupd_ack signal for dfi_phyupd_type=0x0. The dfi_phyupd_req signal may de-assert at any cycle after the assertion of the dfi_phyupd_ack signal.

DDR_PCTL_DFITPHYUPDTYPE1

Address: Operational Base + offset (0x0274)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x010	tphyupd_type1 Specifies the maximum number of DFI clock cycles that the dfi_phyupd_req signal may remain asserted after the assertion of the dfi_phyupd_ack signal for dfi_phyupd_type=0x1. The dfi_phyupd_req signal may de-assert at any cycle after the assertion of the dfi_phyupd_ack signal.

DDR_PCTL_DFITPHYUPDTYPE2

Address: Operational Base + offset (0x0278)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x010	tphyupd_type2 Specifies the maximum number of DFI clock cycles that the dfi_phyupd_req signal may remain asserted after the assertion of the dfi_phyupd_ack signal for dfi_phyupd_type=0x2. The dfi_phyupd_req signal may de-assert at any cycle after the assertion of the dfi_phyupd_ack signal.

DDR_PCTL_DFITPHYUPDTYPE3

Address: Operational Base + offset (0x027c)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x010	tphyupd_type3 Specifies the maximum number of DFI clock cycles that the dfi_phyupd_req signal may remain asserted after the assertion of the dfi_phyupd_ack signal for dfi_phyupd_type=0x3. The dfi_phyupd_req signal may de-assert at any cycle after the assertion of the dfi_phyupd_ack signal.

DDR_PCTL_DFITCTRLUPDMIN

Address: Operational Base + offset (0x0280)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0010	tctrlupd_min Specifies the minimum number of DFI clock cycles that the dfi_ctrlupd_req signal must be asserted.

DDR_PCTL_DFITCTRLUPDMAX

Address: Operational Base + offset (0x0284)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0040	tctrlupd_max Specifies the maximum number of DFI clock cycles that the dfi_ctrlupd_req signal can assert.

DDR_PCTL_DFITCTRLUPDDLY

Address: Operational Base + offset (0x0288)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x8	tctrlupd_dly Delay in DFI clock cycles between time a uPCTL-initiated update could be started and time uPCTL-initated update actually starts (dfi_ctrlupd_req going high).

DDR_PCTL_DFIUPDCFG

Address: Operational Base + offset (0x0290)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x1	dfi_phyupd_en 1'b0: Disabled 1'b1: Enabled
0	RW	0x1	dfi_ctrlupd_en 1'b0: Disabled 1'b1: Enabled

DDR_PCTL_DFITREFMSKI

Address: Operational Base + offset (0x0294)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	trefmski Time period of the masked Refresh interva This value is only used if TREFI==0. Defines the time period (in 100ns units) of the masked Refresh (REFMSK) interval. The actual time period defined is DFITREFMSKI* TOGCNT100N * internal timers clock period.

DDR_PCTL_DFITCTRLUPDI

Address: Operational Base + offset (0x0298)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	tctrlupd_interval If TREFI!=0, the time period is defined as DFITCTRLUPDI*TREFI * TOGCNT100N * internal timers clock period. If TREFI==0 and DFITREFMSKI!=0, the period changes to DFITCTRLUPDI*DFITREFMSKI* * TOGCNT100N * internal timers clock period. Programming a value of 0 is the same as programming a value of 1; for instance, a uPCTL-initiated update occurs every Refresh interval.

DDR_PCTL_DFITRCFG0

Address: Operational Base + offset (0x02ac)

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:16	RW	0x0	dfi_wrlvl_rank_sel Determines the value to drive on the output signal dfi_wrlvl_cs_n. The value on dfi_wrlvl_cs_n is the inverse of the setting in this field.
15:13	RO	0x0	reserved
12:4	RW	0x000	dfi_rdlvl_edge Determines the value to drive on the output signal dfi_rdlvl_edge. The value on dfi_rdlvl_edge is the same as the setting in this field.
3:0	RW	0x0	dfi_rdlvl_rank_sel Determines the value to drive on the output signal dfi_rdlvl_cs_n. The value on dfi_rdlvl_cs_n is the inverse of the setting in this field.

DDR_PCTL_DFITRSTAT0
Address: Operational Base + offset (0x02b0)

Bit	Attr	Reset Value	Description
31:18	RO	0x0	reserved
17:16	RO	0x0	dfi_wrlvl_mode Reports the value of the input signal dfi_wrlvl_mode.
15:10	RO	0x0	reserved
9:8	RO	0x0	dfi_rdlvl_gate_mode Reports the value of the input signal dfi_rdlvl_gate_mode.
7:2	RO	0x0	reserved
1:0	RO	0x0	dfi_rdlvl_mode Reports the value of the input signal dfi_rdlvl_mode.

DDR_PCTL_DFITRWRLVLEN
Address: Operational Base + offset (0x02b4)

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8:0	RW	0x000	dfi_wrlvl_en Determines the value to drive on the output signal dfi_wrlvl_en.

DDR_PCTL_DFITRRDLVLEN
Address: Operational Base + offset (0x02b8)

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8:0	RW	0x000	dfi_rdlvl_en Determines the value to drive on the output signal dfi_rdlvl_en.

DDR_PCTL_DFITRRDLVLGATEEN

Address: Operational Base + offset (0x02bc)

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8:0	RW	0x000	dfi_rdlvl_gate_en Determines the value to drive on the output signal dfi_rdlvl_gate_en.

DDR_PCTL_DFISTSTAT0

Address: Operational Base + offset (0x02c0)

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24:16	RO	0x000	dfi_data_byte_disable Reports the value of the output signal dfi_data_byte_disable.
15:6	RO	0x0	reserved
5:4	RO	0x0	dfi_freq_ratio Reports the value of the output signal dfi_freq_ratio.
3:2	RO	0x0	reserved
1	RO	0x0	dfi_init_start Reports the value of the output signal dfi_init_start.
0	RO	0x0	dfi_init_complete Reports the value of the input signal dfi_init_complete.

DDR_PCTL_DFISTCFG0

Address: Operational Base + offset (0x02c4)

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	RW	0x0	dfi_data_byte_disable_en Enables the driving of the dfi_data_byte_disable signal. The value driven on dfi_data_byte_disable is dependent on the setting of PPCFG register. 1'b0: Drive dfi_data_byte_disable to default value of all zeroes. 1'b1: Drive dfi_data_byte_disable according to value as defined by PPCFG register setting. Note: should be set to 1only after PPCFG is correctly set.
1	RW	0x0	dfi_freq_ratio_en Enables the driving of the dfi_freq_ratio signal. When enabled, the dfi_freq_ratio value driven is dependent on configuration parameter UPCTL_FREQ_RATIO: 2’b00 is driven when UPCTL_FREQ_RATIO=1; 2'b01 is driven when UPCTL_FREQ_RATIO=2. 1'b0: Drive dfi_freq_ratio to default value of 2'b00. 1'b1: Drive dfi_freq_ratio value according to how configuration parameter is set.

Bit	Attr	Reset Value	Description
0	RW	0x0	dfi_init_start Sets the value of the dfi_init_start signal. 1'b0: dfi_init_start is driven low 1'b1: dfi__init_start is driven high

DDR_PCTL_DFISTCFG1

Address: Operational Base + offset (0x02c8)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	dfi_dram_clk_disable_en_dpd Enables support of the dfi_dram_clk_disable signal with Deep Power Down (DPD). DPD is only for mDDR/LPDDR2. 1'b0: Disable dfi_dram_clk_disable support in relation to DPD 1'b1: Enable dfi_dram_clk_disable support in relation to DPD
0	RW	0x0	dfi_dram_clk_disable_en Enables support of the dfi_dram_clk_disable signal with Self Refresh (SR). 1'b0: Disable dfi_dram_clk_disable support in relation to SR 1'b1: Enable dfi_dram_clk_disable support in relation to SR

DDR_PCTL_DFITDRAMCLKEN

Address: Operational Base + offset (0x02d0)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x2	tdram_clk_enable Specifies the number of DFI clock cycles from the de-assertion of the dfi_dram_clk_disable signal on the DFI until the first valid rising edge of the clock to the DRAM memory devices, at the PHY-DRAM boundary. If the DFI clock and the memory clock are not phase aligned, this timing parameter should be rounded up to the next integer value.

DDR_PCTL_DFITDRAMCLKDIS

Address: Operational Base + offset (0x02d4)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x2	tdram_clk_disable Specifies the number of DFI clock cycles from the assertion of the dfi_dram_clk_disable signal on the DFI until the clock to the DRAM memory devices, at the PHY-DRAM boundary, maintains a low value. If the DFI clock and the memory clock are not phase aligned, this timing parameter should be rounded up to the next integer value.

DDR_PCTL_DFISTCFG2
Address: Operational Base + offset (0x02d8)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	parity_en Enables the DFI parity generation feature (driven on output signal dfi_parity_in) 1'b0: Disable DFI parity generation 1'b1: Enable DFI parity generation
0	RW	0x0	parity_intr_en Enable interrupt generation for DFI parity error (from input signal dfi_parity_error). 1'b0: Disable interrupt 1'b1: Enable interrupt

DDR_PCTL_DFISTPARCLR
Address: Operational Base + offset (0x02dc)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	R/W SC	0x0	parity_log_clr Set this bit to 1'b1 to clear the DFI Status Parity Log register (DFISTPARLOG). 1'b0: Do not clear DFI status Parity Log register 1'b1: Clear DFI status Parity Log register
0	R/W SC	0x0	parity_intr_clr Set this bit to 1'b1 to clear the interrupt generated by an DFI parity error (as enabled by DFISTCFG2.parity_intr_en). It also clears the INTRSTAT.parity_intr register field. It is automatically cleared by hardware when the interrupt has been cleared.

DDR_PCTL_DFISTPARLOG
Address: Operational Base + offset (0x02e0)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	parity_err_cnt Increments any time the DFI parity logic detects a parity error(s) (on dfi_parity_error).

DDR_PCTL_DFILPCFG0
Address: Operational Base + offset (0x02f0)

Bit	Attr	Reset Value	Description
31:28	RW	0x0	dfi_lp_wakeup_dpd Value to drive on dfi_lp_wakeup signal when Deep Power Down mode is entered. Determines the DFI's tlp_wakeup time: 4'b0000: 16 cycles 4'b0001: 32 cycles 4'b0010: 64 cycles 4'b0011: 128 cycles 4'b0100: 256 cycles 4'b0101: 512 cycles 4'b0110: 1024 cycles 4'b0111: 2048 cycles 4'b1000: 4096 cycles 4'b1001: 8192 cycles 4'b1010: 16384 cycles 4'b1011: 32768 cycles 4'b1100: 65536 cycles 4'b1101: 131072 cycles 4'b1110: 262144 cycles 4'b1111: Unlimited
27:25	RO	0x0	reserved
24	RW	0x0	dfi_lp_en_dpd Enables DFI Low Power interface handshaking during Deep Power Down Entry/Exit. 1'b0: Disabled 1'b1: Enabled
23:20	RO	0x0	reserved
19:16	RW	0x7	dfi_tlp_resp Setting for tlp_resp time. Same value is used for both Power Down and Self refresh and Deep Power Down modes. DFI 2.1 specification, recommends using value of 7 always.

Bit	Attr	Reset Value	Description
15:12	RW	0x0	dfi_lp_wakeup_sr Value to drive on dfi_lp_wakeup signal when Self Refresh mode is entered. Determines the DFI's tlp_wakeup time: 4'b0000: 16 cycles 4'b0001: 32 cycles 4'b0010: 64 cycles 4'b0011: 128 cycles 4'b0100: 256 cycles 4'b0101: 512 cycles 4'b0110: 1024 cycles 4'b0111: 2048 cycles 4'b1000: 4096 cycles 4'b1001: 8192 cycles 4'b1010: 16384 cycles 4'b1011: 32768 cycles 4'b1100: 65536 cycles 4'b1101: 131072 cycles 4'b1110: 262144 cycles 4'b1111: Unlimited
11:9	RO	0x0	reserved
8	RW	0x0	dfi_lp_en_sr Enables DFI Low Power interface handshaking during Self Refresh Entry/Exit. 1'b0: Disabled 1'b1: Enabled

Bit	Attr	Reset Value	Description
7:4	RW	0x0	dfi_lp_wakeup_pd Value to drive on dfi_lp_wakeup signal when Power Down mode is entered. Determines the DFI's tlp_wakeup time: 4'b0000: 16 cycles 4'b0001: 32 cycles 4'b0010: 64 cycles 4'b0011: 128 cycles 4'b0100: 256 cycles 4'b0101: 512 cycles 4'b0110: 1024 cycles 4'b0111: 2048 cycles 4'b1000: 4096 cycles 4'b1001: 8192 cycles 4'b1010: 16384 cycles 4'b1011: 32768 cycles 4'b1100: 65536 cycles 4'b1101: 131072 cycles 4'b1110: 262144 cycles 4'b1111: Unlimited
3:1	RO	0x0	reserved
0	RW	0x0	dfi_lp_en_pd Enables DFI Low Power interface handshaking during Power Down Entry/Exit. 1'b0: Disabled 1'b1: Enabled

DDR_PCTL_DFITRWRLVLRESP0

Address: Operational Base + offset (0x0300)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dfi_wrlvl_resp0 Reports the status of the dif_wrlvl_resp[31:0] signal.

DDR_PCTL_DFITRWRLVLRESP1

Address: Operational Base + offset (0x0304)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dfi_wrlvl_resp1 Reports the status of the dif_wrlvl_resp[63:32] signal.

DDR_PCTL_DFITRWRLVLRESP2

Address: Operational Base + offset (0x0308)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	dfi_wrlvl_resp2 Reports the status of the dif_wrlvl_resp[71:64] signal.

DDR_PCTL_DFITRRDLVLRESP0
Address: Operational Base + offset (0x030c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dfi_rdlvl_resp0 Reports the status of the dif_rdlvl_resp[31:0] signal.

DDR_PCTL_DFITRRDLVLRESP1
Address: Operational Base + offset (0x0310)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dfi_rdlvl_resp1 Reports the status of the dif_rdlvl_resp[63:32] signal.

DDR_PCTL_DFITRRDLVLRESP2
Address: Operational Base + offset (0x0314)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	dfi_rdlvl_resp2 Reports the status of the dif_rdlvl_resp[71:64] signal.

DDR_PCTL_DFITRWRLVLDELAY0
Address: Operational Base + offset (0x0318)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dfi_wrlvl_delay0 Sets the value to be driven on the signal dfi_wrlvl_delay_x[31:0].

DDR_PCTL_DFITRWRLVLDELAY1
Address: Operational Base + offset (0x031c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dfi_wrlvl_delay1 Sets the value to be driven on the signal dfi_wrlvl_delay_x[63:32].

DDR_PCTL_DFITRWRLVLDELAY2
Address: Operational Base + offset (0x0320)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	dfi_wrlvl_delay2 Sets the value to be driven on the signal dfi_wrlvl_delay_x[71:64].

DDR_PCTL_DFITRRDLVLDELAY0
Address: Operational Base + offset (0x0324)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dfi_rdlvl_delay0 Sets the value to be driven on the signal dfi_rdlvl_delay_x[31:0].

DDR_PCTL_DFITRRDLVLDELAY1
Address: Operational Base + offset (0x0328)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dfi_rdlvl_delay1 Sets the value to be driven on the signal dfi_rdlvl_delay_x[63:32].

DDR_PCTL_DFITRRDLVLDELAY2
Address: Operational Base + offset (0x032c)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	dfi_rdlvl_delay2 Sets the value to be driven on the signal dfi_rdlvl_delay_x[71:64].

DDR_PCTL_DFITRRDLVLGATEDELAY0
Address: Operational Base + offset (0x0330)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dfi_rdlvl_gate_delay0 Sets the value to be driven on the signal dfi_rdlvl_gate_delay_x[31:0].

DDR_PCTL_DFITRRDLVLGATEDELAY1
Address: Operational Base + offset (0x0334)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dfi_rdlvl_gate_delay1 Sets the value to be driven on the signal dfi_rdlvl_gate_delay_x[63:32].

DDR_PCTL_DFITRRDLVLGATEDELAY2
Address: Operational Base + offset (0x0338)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	dfi_rdlvl_gate_delay2 Sets the value to be driven on the signal dfi_rdlvl_gate_delay_x[71:64].

DDR_PCTL_DFITRCMD
Address: Operational Base + offset (0x033c)

Bit	Attr	Reset Value	Description
31	R/W SC	0x0	dfitrcmd_start DFI Training Command Start. When this bit is set to 1, the command operation defined in the dfitrcmd_opcode field is started. This bit is automatically cleared by the uPCTL after the command is finished. The application can poll this bit to determine when uPCTL is ready to accept another command. This bit cannot be cleared to 1b0 by software.
30:13	RO	0x0	reserved
12:4	RW	0x000	dfitrcmd_en DFI Training Command Enable. Selects which bits of chosen DFI Training command to drive to 1'b1.
3:2	RO	0x0	reserved
1:0	RW	0x0	dfitrcmd_opcode DFI Training Command Opcode. Select which DFI Training command to generate for one n_clk cycle: 2'b00: dfi_wrlvl_load 2'b01: dfi_wrlvl_strobe 2'b10: dfi_rdlvl_load 2'b11: Reserved.

DDR_PCTL_IPVR
Address: Operational Base + offset (0x03f8)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	ip_version ASCII value for each number in the version, followed by a *.

DDR_PCTL_IPTR
Address: Operational Base + offset (0x03fc)

Bit	Attr	Reset Value	Description
31:0	RO	0x44574300	ip_type Contains the IP's identification code, which is an ASCII value to identify the component and it is currently set to the string "DWC". This value never changes.

Notes:left channel A signals: A_DQS0, A_DQSB0, A_DQ0~A_DQ7, A_DM0
right channel A signals: A_DQS1, A_DQSB1, A_DQ8~A_DQ15, A_DM1

2.4.3 Registers Summary For DDR PHY

DDRPHY_REG0	0x0000	W	0x0000003f	DDR PHY register 00
DDRPHY_REG1	0x0004	W	0x00000004	DDR PHY register 01
DDRPHY_REG2	0x0008	W	0x00000000	DDR PHY register 02
DDRPHY_REG3	0x000c	W	0x00000022	DDR PHY register 03
DDRPHY_REG9	0x0024	W	0x00000000	DDR PHY register 09
DDRPHY_REGB	0x002c	W	0x00000060	DDR PHY register 0B
DDRPHY_REGC	0x0030	W	0x00000000	DDR PHY register 0C
DDRPHY_REG11	0x0044	W	0x00000014	DDR PHY register 11
DDRPHY_REG12	0x0048	W	0x000000a2	DDR PHY register 12
DDRPHY_REG13	0x004c	W	0x00000008	DDR PHY register 13
DDRPHY_REG14	0x0050	W	0x00000000	DDR PHY register 14
DDRPHY_REG16	0x0058	W	0x00000014	DDR PHY register 16
DDRPHY_REG17	0x005c	W	0x00000044	DDR PHY register 17
DDRPHY_REG18	0x0060	W	0x00000014	DDR PHY register 18
DDRPHY_REG20	0x0080	W	0x00000014	DDR PHY register 20
DDRPHY_REG21	0x0084	W	0x00000004	DDR PHY register 21
DDRPHY_REG26	0x0098	W	0x0000000c	DDR PHY register 26
DDRPHY_REG27	0x009c	W	0x00000000	DDR PHY register 27
DDRPHY_REG28	0x00a0	W	0x00000000	DDR PHY register 28
DDRPHY_REG2B	0x00ac	W	0x00000090	DDR PHY register 2B
DDRPHY_REG2E	0x00b8	W	0x00000004	DDR PHY register 2E
DDRPHY_REF2F	0x00bc	W	0x00000014	DDR PHY register 2F
DDRPHY_REG30	0x00c0	W	0x00000014	DDR PHY register 30
DDRPHY_REG31	0x00c4	W	0x00000004	DDR PHY register 31
DDRPHY_REG36	0x00d8	W	0x0000000c	DDR PHY register 36
DDRPHY_REG37	0x00dc	W	0x00000000	DDR PHY register 37
DDRPHY_REG38	0x00e0	W	0x00000001	DDR PHY register 38
DDRPHY_REG3B	0x00ec	W	0x00000090	DDR PHY register 3B
DDRPHY_REG3E	0x00f8	W	0x00000000	DDR PHY register 3E
DDRPHY_REF3F	0x00fc	W	0x00000014	DDR PHY register 3F
DDRPHY_REG70	0x01c0	W	0x00000007	DDR PHY register 70
DDRPHY_REG71	0x01c4	W	0x00000007	DDR PHY register 71
DDRPHY_REG72	0x01c8	W	0x00000070	DDR PHY register 72
DDRPHY_REG73	0x01cc	W	0x00000000	DDR PHY register 73
DDRPHY_REG74	0x01d0	W	0x00000000	DDR PHY register 74
DDRPHY_REG75	0x01d4	W	0x00000000	DDR PHY register 75
DDRPHY_REG76	0x01d8	W	0x00000000	DDR PHY register 76
DDRPHY_REG77	0x01dc	W	0x00000000	DDR PHY register 77
DDRPHY_REG78	0x01e0	W	0x00000000	DDR PHY register 78
DDRPHY_REG79	0x01e4	W	0x00000000	DDR PHY register 79
DDRPHY_REG7A	0x01e8	W	0x00000000	DDR PHY register 7A
DDRPHY_REG7B	0x01ec	W	0x00000000	DDR PHY register 7B
DDRPHY_REG7C	0x01f0	W	0x00000000	DDR PHY register 7C
DDRPHY_REG7D	0x01f4	W	0x00000000	DDR PHY register 7D

DDRPHY_REG7E	0x01f8	W	0x00000000	DDR PHY register 7E
DDRPHY_REG7F	0x01fc	W	0x00000000	DDR PHY register 7F
DDRPHY_REG80	0x0200	W	0x00000000	DDR PHY register 80
DDRPHY_REG81	0x0204	W	0x00000000	DDR PHY register 81
DDRPHY_REG82	0x0208	W	0x00000000	DDR PHY register 82
DDRPHY_REG83	0x020c	W	0x00000000	DDR PHY register 83
DDRPHY_REG84	0x0210	W	0x00000000	DDR PHY register 84
DDRPHY_REG85	0x0214	W	0x00000000	DDR PHY register 85
DDRPHY_REGB0	0x02c0	W	0x00000077	DDR PHY register B0
DDRPHY_REGB1	0x02c4	W	0x00000077	DDR PHY register B1
DDRPHY_REGB2	0x02c8	W	0x00000077	DDR PHY register B2
DDRPHY_REGB3	0x02cc	W	0x00000077	DDR PHY register B3
DDRPHY_REGB4	0x02d0	W	0x00000077	DDR PHY register B4
DDRPHY_REGB5	0x02d4	W	0x00000077	DDR PHY register B5
DDRPHY_REGB6	0x02d8	W	0x00000077	DDR PHY register B6
DDRPHY_REGB7	0x02dc	W	0x00000077	DDR PHY register B7
DDRPHY_REGB8	0x02e0	W	0x00000077	DDR PHY register B8
DDRPHY_REGB9	0x02e4	W	0x00000077	DDR PHY register B9
DDRPHY_REGBA	0x02e8	W	0x00000077	DDR PHY register BA
DDRPHY_REGBB	0x02ec	W	0x00000077	DDR PHY register BB
DDRPHY_REGBC	0x02f0	W	0x00000077	DDR PHY register BC
DDRPHY_REGBD	0x02f4	W	0x00000077	DDR PHY register BD
DDRPHY_REGC0	0x0300	W	0x00000000	DDR PHY register C0
DDRPHY_REGC1	0x0304	W	0x00000000	DDR PHY register C1
DDRPHY_REGC2	0x0308	W	0x00000000	DDR PHY register C2
DDRPHY_REGC3	0x030c	W	0x00000000	DDR PHY register C3
DDRPHY_REGC4	0x0310	W	0x00000000	DDR PHY register C4
DDRPHY_REGC5	0x0314	W	0x00000000	DDR PHY register C5
DDRPHY_REGC6	0x0318	W	0x00000000	DDR PHY register C6
DDRPHY_REGC7	0x031c	W	0x00000000	DDR PHY register C7
DDRPHY_REGC8	0x0320	W	0x00000000	DDR PHY register C8
DDRPHY_REGC9	0x0324	W	0x00000000	DDR PHY register C9
DDRPHY_REGCA	0x0328	W	0x00000000	DDR PHY register CA
DDRPHY_REGCB	0x032c	W	0x00000000	DDR PHY register CB
DDRPHY_REGCC	0x0330	W	0x00000000	DDR PHY register CC
DDRPHY_REGCD	0x0334	W	0x00000000	DDR PHY register CD
DDRPHY_REGCE	0x0338	W	0x00000000	DDR PHY register CE
DDRPHY_REGCF	0x033c	W	0x00000000	DDR PHY register CF
DDRPHY_REGD0	0x0340	W	0x00000000	DDR PHY register D0
DDRPHY_REGD1	0x0344	W	0x00000000	DDR PHY register D1
DDRPHY_REGD2	0x0348	W	0x00000000	DDR PHY register D2
DDRPHY_REGD3	0x034c	W	0x00000000	DDR PHY register D3
DDRPHY_REGD4	0x0350	W	0x00000000	DDR PHY register D4
DDRPHY_REGD5	0x0354	W	0x00000000	DDR PHY register D5

DDRPHY_REGF0	0x03c0	W	0x00000000	DDR PHY register F0
DDRPHY_REGF1	0x03c4	W	0x00000000	DDR PHY register F1
DDRPHY_REGFA	0x03e8	W	0x00000000	DDR PHY register FA
DDRPHY_REGFB	0x03ec	W	0x00000000	DDR PHY register FB
DDRPHY_REGFC	0x03f0	W	0x00000000	DDR PHY register FC
DDRPHY_REGFF	0x03fc	W	0x00000000	DDR PHY register FF

2.4.4 Detail Register Description For DDR PHY

DDRPHY_REG0

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x3	bit7: reserved bit6: reserved bit5: The enable signal of the high 8bits of the DQ16 bit4: The enable signal of the low 8bits of the DQ16
3	RW	0x1	reset digital core, active low
2	RW	0x1	reset analog logic, active low
1:0	RW	0x3	Reserved

DDRPHY_REG1

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:2	RW	0x01	reserved
1:0	RW	0x0	2'b00: ddr3 PHY mode 2'b01: ddr2 PHY mode 2'b10: lpddr2 PHY mode

DDRPHY_REG2

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:6	RW	0x0	2'b00: select CS0 and CS1 2'b01: select CS1 2'b10: select CS0
5:4	RW	0x0	2'b00: select CS0 and CS1 2'b01: select CS1 2'b10: select CS0
3	RW	0x0	reserved
2	RW	0x0	reserved
1	RW	0x0	DQS gating calibration bypass mode, active high
0	RW	0x0	DQS gating calibration control, active high

DDRPHY_REG3

Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6:4	RW	0x2	right channel A read ODT delay by read odt configure bypass mode
3	RO	0x0	reserved
2:0	RW	0x2	left channel A read ODT delay by read odt configure bypass mode

DDRPHY_REG9
Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	1'b0: Normal mode 1'b1: Bypass
5:0	RO	0x0	reserved

DDRPHY_REGB
Address: Operational Base + offset (0x002c)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x6	CL value DDR2/DDR3 CAS Latency LPDDR2/3 RL value
3:0	RW	0x0	AL value DDR2/DDR3 additive latency

DDRPHY_REGC
Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x0	CWL value DDR3 WRITE CAS Latency LPDDR2/3 WL value

DDRPHY_REG11
Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x14	CMD NRCOMP, except for CK/CKB. The larger the value, the stronger the drive strength.

DDRPHY_REG12
Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:3	RW	0x14	CMD PRCOMP, except for CK/CKB. The larger the value, the stronger the drive strength.
2	RO	0x0	reserved
1	RW	0x1	CMD weak pull up enable, active low
0	RW	0x0	CMD weak pull down enable, active high

DDRPHY_REG13

Address: Operational Base + offset (0x004c)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RW	0x0	CMD DLL clock phase select in bypass mode 1'b0: no delay 1'b1: 90°delay
3	RW	0x1	CMD DLL enable 1'b0: disable 1'b1: enable
2:0	RW	0x0	CMD AND ADDRESS DLL delay 3'b000: no delay 3'b001: 22.5°delay 3'b010: 45°delay 3'b011: 67.5°delay 3'b100: 90°delay 3'b101: 112.5°delay 3'b110: 135°delay 3'b111: 157.5°delay

DDRPHY_REG14

Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RW	0x0	CK DLL clock phase select in bypass mode 1'b0: no delay 1'b1: 90°delay
2:0	RW	0x0	CK DLL delay 3'b000: no delay 3'b001: 22.5°delay 3'b010: 45°delay 3'b011: 67.5°delay 3'b100: 90°delay 3'b101: 112.5°delay 3'b110: 135°delay 3'b111: 157.5°delay

DDRPHY_REG16

Address: Operational Base + offset (0x0058)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x14	CK/CKB NRCOMP. The larger the value, the stronger the drive strength.

DDRPHY_REG17

Address: Operational Base + offset (0x005c)

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6:4	RW	0x4	CMD/CK falling edge slew rate control, larger value means larger falling slew rate
3	RO	0x0	reserved
2:0	RW	0x4	CMD/CK rising edge slew rate control, larger value means larger rising slew rate

DDRPHY_REG18

Address: Operational Base + offset (0x0060)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x14	CK/CKB PRCOMP. The larger the value, the stronger the drive strength.

DDRPHY_REG20

Address: Operational Base + offset (0x0080)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x14	Left channel A NRCOMP. The larger the value, the stronger the drive strength in the scope from A_DQ0 to A_DQ7.

DDRPHY_REG21

Address: Operational Base + offset (0x0084)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x04	Left channel A read pull-down ODT. The larger the value, the smaller the pull-down resistance in the scope from A_DQ0 to A_DQ7.

DDRPHY_REG26

Address: Operational Base + offset (0x0098)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RW	0x0	Left channel A write DQ DLL phase select in bypass mode. 1'b0: no delay 1'b1: 90°delay
3	RW	0x1	Left channel A write DQ DLL enable, active HIGH.
2:0	RW	0x4	Left channel A write DQ DLL delay 3'b000: no delay 3'b001: 22.5°delay 3'b010: 45°delay 3'b011: 67.5°delay 3'b100: 90°delay 3'b101: 112.5°delay 3'b110: 135°delay 3'b111: 157.5°delay

DDRPHY_REG27

Address: Operational Base + offset (0x009c)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RW	0x0	Left channel A write DQS DLL phase select in bypass mode. 1'b0: no delay 1'b1: 90°delay
2:0	RW	0x0	Left channel A write DQS DLL delay 3'b000: no delay 3'b001: 22.5°delay 3'b010: 45°delay 3'b011: 67.5°delay 3'b100: 90°delay 3'b101: 112.5°delay 3'b110: 135°delay 3'b111: 157.5°delay

DDRPHY_REG28

Address: Operational Base + offset (0x00a0)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1:0	RW	0x0	Left channel A read DQS DLL delay 2'b00: no delay 2'b01: 22.5°delay 2'b10: 45°delay 2'b11: 67.5°delay

DDRPHY_REG2B

Address: Operational Base + offset (0x00ac)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:5	RW	0x4	Left channel A falling edge slew rate control, larger value means larger falling slew rate.
4:2	RW	0x4	Left channel A rising edge slew rate control, larger value means larger rising slew rate.
1:0	RO	0x0	reserved

DDRPHY_REG2E

Address: Operational Base + offset (0x00b8)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x04	Left channel A read pull-up ODT. The larger the value, the smaller the pull-up resistance in the scope from A_DQ0 to A_DQ7.

DDRPHY_REF2F

Address: Operational Base + offset (0x00bc)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x14	Left channel A PRCOMP. The larger the value, the stronger the drive strength in the scope from A_DQ0 to A_DQ7.

DDRPHY_REG30

Address: Operational Base + offset (0x00c0)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x14	Right channel A NRCOMP. The larger the value, the stronger the drive strength in the scope from A_DQ8 to A_DQ15.

DDRPHY_REG31

Address: Operational Base + offset (0x00c4)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x04	Right channel A read pull-down ODT. The larger the value, the smaller the pull-down resistance in the scope from A_DQ8 to A_DQ15

DDRPHY_REG36

Address: Operational Base + offset (0x00d8)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RW	0x0	Right channel A write DQ DLL phase select in bypass mode. 1'b0: no delay 1'b1: 90°delay
3	RW	0x1	Right channel A write DQ DLL enable, active HIGH.
2:0	RW	0x4	Right channel A write DQ DLL delay 3'b000: no delay 3'b001: 22.5°delay 3'b010: 45°delay 3'b011: 67.5°delay 3'b100: 90°delay 3'b101: 112.5°delay 3'b110: 135°delay 3'b111: 157.5°delay

DDRPHY_REG37

Address: Operational Base + offset (0x00dc)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RW	0x0	Right channel A write DQS DLL phase select in bypass mode. 1'b0: no delay 1'b1: 90°delay
2:0	RW	0x0	Right channel A write DQS DLL delay 3'b000: no delay 3'b001: 22.5°delay 3'b010: 45°delay 3'b011: 67.5°delay 3'b100: 90°delay 3'b101: 112.5°delay 3'b110: 135°delay 3'b111: 157.5°delay

DDRPHY_REG38

Address: Operational Base + offset (0x00e0)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1:0	RW	0x1	Right channel A read DQS DLL delay 2'b00: no delay 2'b01: 22.5°delay 2'b10: 45°delay 2'b11: 67.5°delay

DDRPHY_REG3B

Address: Operational Base + offset (0x00ec)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:5	RW	0x4	Right channel A falling edge slew rate control, larger value means larger falling slew rate.
4:2	RW	0x4	Right channel A rising edge slew rate control, larger value means larger rising slew rate.
1:0	RO	0x0	reserved

DDRPHY_REG3E

Address: Operational Base + offset (0x00f8)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	Right channel A read pull-up ODT. The larger the value, the smaller the pull-up resistance in the scope from A_DQ8 to A_DQ15.

DDRPHY_REF3F

Address: Operational Base + offset (0x00fc)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x14	Right channel A PRCOMP. The larger the value, the stronger the drive strength in the scope from A_DQ8 to A_DQ15.

DDRPHY_REG70

Address: Operational Base + offset (0x01c0)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x0	CS0 A_DM0 RX de-skew
3:0	RW	0x7	CS0 A_DM0 TX de-skew

DDRPHY_REG71

Address: Operational Base + offset (0x01c4)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x0	CS0 A_DQ0 RX de-skew
3:0	RW	0x7	CS0 A_DQ0 TX de-skew

DDRPHY_REG72

Address: Operational Base + offset (0x01c8)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	CS0 A_DQ1 RX de-skew
3:0	RW	0x0	CS0 A_DQ1 TX de-skew

DDRPHY_REG73
Address: Operational Base + offset (0x01cc)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x0	CS0 A_DQ2 RX de-skew
3:0	RW	0x0	CS0 A_DQ2 TX de-skew

DDRPHY_REG74
Address: Operational Base + offset (0x01d0)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x0	CS0 A_DQ3 RX de-skew
3:0	RW	0x0	CS0 A_DQ3 TX de-skew

DDRPHY_REG75
Address: Operational Base + offset (0x01d4)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x0	CS0 A_DQ4 RX de-skew
3:0	RW	0x0	CS0 A_DQ4 TX de-skew

DDRPHY_REG76
Address: Operational Base + offset (0x01d8)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x0	CS0 A_DQ5 RX de-skew
3:0	RW	0x0	CS0 A_DQ5 TX de-skew

DDRPHY_REG77
Address: Operational Base + offset (0x01dc)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x0	CS0 A_DQ6 RX de-skew
3:0	RW	0x0	CS0 A_DQ6 TX de-skew

DDRPHY_REG78
Address: Operational Base + offset (0x01e0)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x0	CS0 A_DQ7 RX de-skew
3:0	RW	0x0	CS0 A_DQ7 TX de-skew

DDRPHY_REG79

Address: Operational Base + offset (0x01e4)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x0	CS0 A_DQS0 RX de-skew
3:0	RW	0x0	CS0 A_DQS0 TX de-skew

DDRPHY_REG7A
Address: Operational Base + offset (0x01e8)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x0	CS0 A_DQSB0 RX de-skew
3:0	RW	0x0	CS0 A_DQSB0 TX de-skew

DDRPHY_REG7B
Address: Operational Base + offset (0x01ec)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x0	CS0 A_DM1 RX de-skew
3:0	RW	0x0	CS0 A_DM1 TX de-skew

DDRPHY_REG7C
Address: Operational Base + offset (0x01f0)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x0	CS0 A_DQ8 RX de-skew
3:0	RW	0x0	CS0 A_DQ8 TX de-skew

DDRPHY_REG7D
Address: Operational Base + offset (0x01f4)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x0	CS0 A_DQ9 RX de-skew
3:0	RW	0x0	CS0 A_DQ9 TX de-skew

DDRPHY_REG7E
Address: Operational Base + offset (0x01f8)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x0	CS0 A_DQ10 RX de-skew
3:0	RW	0x0	CS0 A_DQ10 TX de-skew

DDRPHY_REG7F
Address: Operational Base + offset (0x01fc)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x0	CS0 A_DQ11 RX de-skew
3:0	RW	0x0	CS0 A_DQ11 TX de-skew

DDRPHY_REG80
Address: Operational Base + offset (0x0200)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x0	CS0 A_DQ12 RX de-skew
3:0	RW	0x0	CS0 A_DQ12 TX de-skew

DDRPHY_REG81
Address: Operational Base + offset (0x0204)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x0	CS0 A_DQ13 RX de-skew
3:0	RW	0x0	CS0 A_DQ13 TX de-skew

DDRPHY_REG82
Address: Operational Base + offset (0x0208)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x0	CS0 A_DQ14 RX de-skew
3:0	RW	0x0	CS0 A_DQ14 TX de-skew

DDRPHY_REG83
Address: Operational Base + offset (0x020c)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x0	CS0 A_DQ15 RX de-skew
3:0	RW	0x0	CS0 A_DQ15 TX de-skew

DDRPHY_REG84
Address: Operational Base + offset (0x0210)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x0	CS0 A_DQS1 RX de-skew
3:0	RW	0x0	CS0 A_DQS1 TX de-skew

DDRPHY_REG85
Address: Operational Base + offset (0x0214)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x0	CS0 A_DQSB1 RX de-skew
3:0	RW	0x0	CS0 A_DQSB1 TX de-skew

DDRPHY_REGB0
Address: Operational Base + offset (0x02c0)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	A1 de-skew
3:0	RW	0x7	A0 de-skew

DDRPHY_REGB1
Address: Operational Base + offset (0x02c4)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	A3 de-skew
3:0	RW	0x7	A2 de-skew

DDRPHY_REGB2
Address: Operational Base + offset (0x02c8)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	A5 de-skew
3:0	RW	0x7	A4 de-skew

DDRPHY_REGB3
Address: Operational Base + offset (0x02cc)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	A7 de-skew
3:0	RW	0x7	A6 de-skew

DDRPHY_REGB4
Address: Operational Base + offset (0x02d0)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	A9 de-skew
3:0	RW	0x7	A8 de-skew

DDRPHY_REGB5
Address: Operational Base + offset (0x02d4)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	A11 de-skew
3:0	RW	0x7	A10 de-skew

DDRPHY_REGB6
Address: Operational Base + offset (0x02d8)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	A13 de-skew
3:0	RW	0x7	A12 de-skew

DDRPHY_REGB7
Address: Operational Base + offset (0x02dc)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	A15 de-skew
3:0	RW	0x7	A14 de-skew

DDRPHY_REGB8
Address: Operational Base + offset (0x02e0)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	B1 de-skew
3:0	RW	0x7	B0 de-skew

DDRPHY_REGB9
Address: Operational Base + offset (0x02e4)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	RAS# de-skew
3:0	RW	0x7	B2 de-skew

DDRPHY_REGBA
Address: Operational Base + offset (0x02e8)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	WE# de-skew
3:0	RW	0x7	CAS# de-skew

DDRPHY_REGBB
Address: Operational Base + offset (0x02ec)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	CKB de-skew
3:0	RW	0x7	CK de-skew

DDRPHY_REGBC
Address: Operational Base + offset (0x02f0)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	CKE de-skew
3:0	RW	0x7	ODT0 de-skew

DDRPHY_REGBD
Address: Operational Base + offset (0x02f4)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	CSB0 de-skew
3:0	RW	0x7	RESETN de-skew

DDRPHY_REGC0
Address: Operational Base + offset (0x0300)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x0	CS1 A_DM0 RX de-skew
3:0	RW	0x0	CS1 A_DM0 TX de-skew

DDRPHY_REGC1
Address: Operational Base + offset (0x0304)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x0	CS1 A_DQ0 RX de-skew
3:0	RW	0x0	CS1 A_DQ0 TX de-skew

DDRPHY_REGC2
Address: Operational Base + offset (0x0308)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x0	CS1 A_DQ1 RX de-skew
3:0	RW	0x0	CS1 A_DQ1 TX de-skew

DDRPHY_REGC3
Address: Operational Base + offset (0x030c)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x0	CS1 A_DQ2 RX de-skew
3:0	RW	0x0	CS1 A_DQ2 TX de-skew

DDRPHY_REGC4
Address: Operational Base + offset (0x0310)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x0	CS1 A_DQ3 RX de-skew
3:0	RW	0x0	CS1 A_DQ3 TX de-skew

DDRPHY_REGC5
Address: Operational Base + offset (0x0314)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x0	CS1 A_DQ4 RX de-skew
3:0	RW	0x0	CS1 A_DQ4 TX de-skew

DDRPHY_REGC6
Address: Operational Base + offset (0x0318)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x0	CS1 A_DQ5 RX de-skew
3:0	RW	0x0	CS1 A_DQ5 TX de-skew

DDRPHY_REGC7
Address: Operational Base + offset (0x031c)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x0	CS1 A_DQ6 RX de-skew
3:0	RW	0x0	CS1 A_DQ6 TX de-skew

DDRPHY_REGC8
Address: Operational Base + offset (0x0320)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x0	CS1 A_DQ7 RX de-skew
3:0	RW	0x0	CS1 A_DQ7 TX de-skew

DDRPHY_REGC9
Address: Operational Base + offset (0x0324)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x0	CS1 A_DQS0 RX de-skew
3:0	RW	0x0	CS1 A_DQS0 TX de-skew

DDRPHY_REGCA
Address: Operational Base + offset (0x0328)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x0	CS1 A_DQSB0 RX de-skew
3:0	RW	0x0	CS1 A_DQSB0 TX de-skew

DDRPHY_REGCB
Address: Operational Base + offset (0x032c)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x0	CS1 A_DM1 RX de-skew
3:0	RW	0x0	CS1 A_DM1 TX de-skew

DDRPHY_REGCC
Address: Operational Base + offset (0x0330)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x0	CS1 A_DQ8 RX de-skew
3:0	RW	0x0	CS1 A_DQ8 TX de-skew

DDRPHY_REGCD
Address: Operational Base + offset (0x0334)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x0	CS1 A_DQ9 RX de-skew
3:0	RW	0x0	CS1 A_DQ9 TX de-skew

DDRPHY_REGCE
Address: Operational Base + offset (0x0338)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x0	CS1 A_DQ10 RX de-skew
3:0	RW	0x0	CS1 A_DQ10 TX de-skew

DDRPHY_REGCF
Address: Operational Base + offset (0x033c)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x0	CS1 A_DQ11 RX de-skew
3:0	RW	0x0	CS1 A_DQ11 TX de-skew

DDRPHY_REGD0
Address: Operational Base + offset (0x0340)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x0	CS1 A_DQ12 RX de-skew
3:0	RW	0x0	CS1 A_DQ12 TX de-skew

DDRPHY_REGD1
Address: Operational Base + offset (0x0344)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x0	CS1 A_DQ13 RX de-skew
3:0	RW	0x0	CS1 A_DQ13 TX de-skew

DDRPHY_REGD2
Address: Operational Base + offset (0x0348)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x0	CS1 A_DQ14 RX de-skew
3:0	RW	0x0	CS1 A_DQ14 TX de-skew

DDRPHY_REGD3
Address: Operational Base + offset (0x034c)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x0	CS1 A_DQ15 RX de-skew
3:0	RW	0x0	CS1 A_DQ15 TX de-skew

DDRPHY_REGD4
Address: Operational Base + offset (0x0350)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x0	CS1 A_DQS1 RX de-skew
3:0	RW	0x0	CS1 A_DQS1 TX de-skew

DDRPHY_REGD5
Address: Operational Base + offset (0x0354)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x0	CS1 A_DQSB1 RX de-skew
3:0	RW	0x0	CS1 A_DQSB1 TX de-skew

DDRPHY_REGF0

Address: Operational Base + offset (0x03c0)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RO	0x0	Channel A High 8bit write leveling done
0	RO	0x0	Channel A Low 8bit write leveling done

DDRPHY_REGF1

Address: Operational Base + offset (0x03c4)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RO	0x0	Channel A High 8bit write leveling dqs value
3:0	RO	0x0	Channel A Low 8bit write leveling dqs value

DDRPHY_REGFA

Address: Operational Base + offset (0x03e8)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RO	0x0	Channel A High 8bit dqs gate sample dqs value(idqs) (3)
0	RO	0x0	Channel A Low 8bit dqs gate sample dqs value(idqs) (3)

DDRPHY_REGFB

Address: Operational Base + offset (0x03ec)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:5	RO	0x0	Calibration get the cyclesel configure channel A low 8bit(3)
4:3	RO	0x0	Calibration get the ophsel configure channel A low 8bit(3)
2:0	RO	0x0	Calibration get the dll configure channel A low 8bit(3)

DDRPHY_REGFC

Address: Operational Base + offset (0x03f0)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:5	RW	0x0	Calibration get the cyclesel configure channel A high 8bit(3)
4:3	RO	0x0	Calibration get the ophsel configure channel A high 8bit(3)
2:0	RO	0x0	Calibration get the dll configure channel A high 8bit(3)

DDRPHY_REGFF

Address: Operational Base + offset (0x03fc)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	Channel A High 8bit Calibration done
0	RO	0x0	Channel A Low 8bit Calibration done

2.4.5 Registers Summary For DDR Monitor

DDRMON_IP_VERSION	0x0000	W	0x00000021	IP version
DDRMON_CTRL	0x0004	W	0x00000008	DDR Monitor Control
DDRMON_INT_STATUS	0x0008	W	0x00000000	Interrupt Status
DDRMON_INT_MASK	0x000c	W	0x000001ff	Interrupt mask control
DDRMON_TIMER_COUNT	0x0010	W	0x00000000	The DFI Timer Threshold
DDRMON_FLOOR_NUMBER	0x0014	W	0x00000000	The Low Threshold in the Comparison of DDR Access
DDRMON_TOP_NUMBER	0x0018	W	0x00000000	The High Threshold in the Comparison of DDR Access
DDRMON_DFI_ACT_NUM	0x001c	W	0x00000000	DFI Active Command Number
DDRMON_DFI_WR_NUM	0x0020	W	0x00000000	DFI write Command Number
DDRMON_DFI_RD_NUM	0x0024	W	0x00000000	DFI read Command Number
DDRMON_CH0_COUNT_NUM	0x0028	W	0x00000000	Channel 0 Timer Count Number
DDRMON_DFI_ACCESS_NUM	0x002c	W	0x00000000	DFI Read and Write Command Number
DDRMON_DDR_IF_CTRL	0x0200	W	0x00000000	DDR interface Control Register
DDRMON_DDR_MSTID	0x0204	W	0x00000000	Master And AXI ID of DDR Command
DDRMON_DDR_IDMSK	0x0208	W	0x00000000	Master and AXI ID MASK Of DDR Command
DDRMON_WR_START_ADDR	0x020c	W	0x00000000	Write Start Address
DDRMON_WR_END_ADDR	0x0210	W	0x00000000	Write End Address
DDRMON_RD_START_ADDR	0x0214	W	0x00000000	Read Start Address
DDRMON_RD_END_ADDR	0x0218	W	0x00000000	Read End Address
DDRMON_DDR_FIFO0_ADDR	0x0240	W	0x00000000	DDR Controller Interface Address FIFO0
DDRMON_DDR_FIFO0_ID	0x0244	W	0x00000000	DDR Controller Interface Command ID FIFO0
DDRMON_DDR_FIFO1_ADDR	0x0248	W	0x00000000	DDR Controller Interface Address FIFO1
DDRMON_DDR_FIFO1_ID	0x024c	W	0x00000000	DDR Controller Interface Command ID FIFO1
DDRMON_DDR_FIFO2_ADDR	0x0250	W	0x00000000	DDR Controller Interface Address FIFO2
DDRMON_DDR_FIFO2_ID	0x0254	W	0x00000000	DDR Controller Interface Command ID FIFO2

DDRMON_DDR_FIFO3_ADDR	0x0258	W	0x00000000	DDR Controller Interface Address FIFO3
DDRMON_DDR_FIFO3_ID	0x025c	W	0x00000000	DDR Controller Interface Command ID FIFO3

2.4.6 Detail Register Description For DDR Monitor

DDRMON_IP_VERSION

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x21	ip_version DDR monitor IP version

DDRMON_CTRL

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable When bit 16=1, bit 0 can be written by softwar . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:4	RO	0x0	reserved
3	RW	0x1	hardware_en 1'b1: enable 1'b0: disable
2	RW	0x0	lpddr2_en 1'b1: enable 1'b0: disable
1	RW	0x0	software_en 1'b1: enable 1'b0: disable
0	RW	0x0	timer_cnt_en 1'b1: enable 1'b0: disable

DDRMON_INT_STATUS

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RO	0x0	rd_addr_hit This is the interrupt status read address hit the setting range
5	RO	0x0	wr_addr_hit This is the interrupt status write address hit the setting range
4:2	RO	0x0	reserved
1	RO	0x0	over_int This is the interrupt status of DDR read and write burst number more than high threshold
0	RO	0x0	below_int This is the interrupt status of DDR read and write burst number less than low threshold

DDRMON_INT_MASK
Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8:0	RW	0x1ff	int_mask Interrupt mask control, when bit set to 1'b1, the corresponding interrupt will disable

DDRMON_TIMER_COUNT
Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	timer_count The timer counter threshold, the statistics of DDR access only be done when timer counter is less than this threshold in hardware mode, in OSC clock cycle.

DDRMON_FLOOR_NUMBER
Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	floor_number The low threshold in the comparison of DDR access

DDRMON_TOP_NUMBER
Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	top_number The high threshold in the comparison of DDR access

DDRMON_DFI_ACT_NUM
Address: Operational Base + offset (0x001c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dfi_act_num DFI active command number in the statistics range

DDRMON_DFI_WR_NUM

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dfi_wr_num DFI write command number in the statistics range

DDRMON_DFI_RD_NUM

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dfi_rd_num DFI read command number in the statistics range

DDRMON_CH0_COUNT_NUM

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	ch0_dfi_count_num The DFI counter number in the statistics range of the channel 0, in DDR clock cycle.

DDRMON_DFI_ACCESS_NUM

Address: Operational Base + offset (0x002c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dfi_access_num DFI read and write command number in the statistics range

DDRMON_DDR_IF_CTRL

Address: Operational Base + offset (0x0200)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:3	RO	0x0	reserved
2	RW	0x0	if_mon_en 1'b1: enable 1'b0: disable

Bit	Attr	Reset Value	Description
1	RO	0x0	reserved
0	RW	0x0	direction 1'b1: read 1'b0: write

DDRMON_DDR_MSTID
Address: Operational Base + offset (0x0204)

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16:0	RW	0x00000	ddr_mstid High 7 bits: Master ID Low 10 bits: AXI command ID

DDRMON_DDR_IDMSK
Address: Operational Base + offset (0x0208)

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16:0	RW	0x00000	ddr_idmsk When bit set to high, this bit of MSTID will be masked, and does not take part in the ID comparison

DDRMON_WR_START_ADDR
Address: Operational Base + offset (0x020c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	wr_start_addr Write start address for address comparison

DDRMON_WR_END_ADDR
Address: Operational Base + offset (0x0210)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	wr_end_addr Write end address for address comparison

DDRMON_RD_START_ADDR
Address: Operational Base + offset (0x0214)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rd_start_addr Read start address for address comparison

DDRMON_RD_END_ADDR
Address: Operational Base + offset (0x0218)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rd_end_addr Read end address for address comparison

DDRMON_DDR_FIFO0_ADDR
Address: Operational Base + offset (0x0240)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	ddr_fifo0_addr DDR controller interface address FIFO0

DDRMON_DDR_FIFO0_ID
Address: Operational Base + offset (0x0244)

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16:0	RO	0x00000	ddr_fifo0_id DDR controller interface command ID FIFO0

DDRMON_DDR_FIFO1_ADDR
Address: Operational Base + offset (0x0248)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	ddr_fifo1_addr DDR controller interface address FIFO1

DDRMON_DDR_FIFO1_ID
Address: Operational Base + offset (0x024c)

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16:0	RO	0x00000	ddr_fifo1_id DDR controller interface command ID FIFO1

DDRMON_DDR_FIFO2_ADDR
Address: Operational Base + offset (0x0250)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	ddr_fifo2_addr DDR controller interface address FIFO2

DDRMON_DDR_FIFO2_ID
Address: Operational Base + offset (0x0254)

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16:0	RO	0x00000	ddr_fifo2_id DDR controller interface command ID FIFO2

DDRMON_DDR_FIFO3_ADDR
Address: Operational Base + offset (0x0258)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	ddr_fifo3_addr DDR controller interface address FIFO3

DDRMON_DDR_FIFO3_ID
Address: Operational Base + offset (0x025c)

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16:0	RO	0x00000	ddr_fifo3_id DDR controller interface command ID FIFO3

2.4.7 Registers Summary For DDR Standby

DDRSTDBY_CON0	0x0000	W	0x00000000	Control Register0
DDRSTDBY_CON1	0x0004	W	0x00000000	Control Register1
DDRSTDBY_STATUS0	0x0008	W	0x00000000	Status Register0

2.4.8 Detail Register Description For DDR Standby

DDRSTDBY_CON0
Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	idle_th idle threshold time. Measured by clk_ddr_stdby(General is ddrphy4x/4). When the time of both memory scheduler and uPCTL in idle status exceed idle_th, it will request uPCTL to enter self-refresh
15:8	RO	0x0	reserved
7	RW	0x0	msch_gate_en 1'b0: disable memory scheduler gated 1'b1: enable memory scheduler gated
6	RW	0x0	ddrphy4x_gate_en 1'b0: disable ddrphy4x gated 1'b1: enable ddrphy4x gated
5	RW	0x0	upctl_core_clk_gate_en 1'b0: disable uPCTL core_clk gated 1'b1: enable uPCTL core_clk gated
4	RW	0x0	upctl_aclk_gate_en 1'b0: disable uPCTL aclk gated 1'b1: enable uPCTL aclk gated
3	RO	0x0	reserved
2	RW	0x0	sysack_ext_dis 1'b0: exit stdby need to wait assertion of sysack 1'b1: exit stdby not need to wait assertion of sysack
1	RW	0x0	ctl_idle_en 1'b0: disable uPCTL idle when ddr_stdby is enabled 1'b1: enable uPCTL idle when ddr_stdby is enabled

Bit	Attr	Reset Value	Description
0	RW	0x0	stdby_en 1'b0: disable ddr_stdby function 1'b1: enable ddr_stdby function

DDRSTDBY_CON1

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	cg_exit_th clk gated exit threshold time. Measured by clk_ddr_stdby(General is ddrphy4x/4). If ddrphy4x_gate_en=1, cg_exit_th need consider the dll lock time of ddr phy
15:0	RW	0x0000	cg_wait_th clk gated wait threshold time. Measured by clk_ddr_stdby(General is ddrphy4x/4). Set this value to 0x0

DDRSTDBY_STATUS0

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
13	RO	0x0	sysactive 1'b0: uPCTL is idle 1'b1: uPCTL is active
12	RO	0x0	sysack 1'b0: ack the request of sysreq enter self-refresh 1'b1: ack the request of sysreq exit self-refresh
11	RO	0x0	sysreq 1'b0: request uPCTL enter self-refresh 1'b1: request uPCTL exit self-refresh
10	RO	0x0	upctl_idle 1'b0: uPCTL is active 1'b1: uPCTL is idle
9	RO	0x0	pwract 1'b0: memory scheduler is not active 1'b1: memory scheduler is active
8	RO	0x0	ddrstdby_gate 1'b0: ddrstdby global gated is disabled 1'b1: ddrstdby global gated is enabled
7	RO	0x0	stdby_en 1'b0: ddrstdby is disabled 1'b1: ddrstdby is enabled

Bit	Attr	Reset Value	Description
6:0	RO	0x00	state 7'b0000001: ST_NORMAL 7'b0000010: ST_STDBY_WAIT_EN 7'b0000100: ST_STDBY_SR_ENTRY 7'b0001000: ST_STDBY_WAIT 7'b0010000: ST_STDBY 7'b0100000: ST_STDBY2 7'b1000000: ST_STDBY_SR_EXIT

2.5 Application Notes

2.5.1 State transition of PCTL

To operate PCTL, the programmer must be familiar with the available operational states and how to transition to each state from the current state.

The following tables provide the programming sequences for moving to the various states of the state machine.

2.5.1.1 Moving to the Init_mem State

Table 2-1 moving to the Init_mem state

Step	Application	PCTL
1	Read STAT register	Returns the current uPCTL state
2	If STAT.ctl_stat = Init_mem, go to END.	
3	If STAT.ctl_stat = Config, go to Step9.	
4	If STAT.ctl_stat =Access, go to Step8.	
5	If STAT.ctl_stat = Low_power, go to Step7.	
6	Go to Step1.	PCTL is in a Transitional state and not in any of the previous operational states.
7	Write WAKEUP to SCTL.state_cmd and poll STAT.ctl_stat= Access.	Issues SRX, moves to the Access state, updates STAT.ctl_stat =Access when complete.
8	Write CFG to SCTL.state_cmd and poll STAT.ctl_stat=ConFig	PCTL stalls the NIF; completes any pending transaction; issues PREA if required; moves into the ConFig state; updates STAT.ctl_stat =ConFig when complete.
9	Write INIT to SCTL.state_cmd and poll STAT.ctl_stat=Init_mem	Moves into the Init_mem state and updates STAT.ctl_stat =Init_mem.
END		PCTL is in Init_mem state.

2.5.1.2 Moving to ConFig State

Table 2-2 moving to the Config state

Step	Application	PCTL
1	Read STAT register.	Returns the current uPCTL state.
2	If STAT.ctl_stat = Config, goto END.	
3	If STAT.ctl_stat = Low_power, go to Step6.	
4	If STAT.ctl_stat = Init_mem or Access, go to Step7.	

Step	Application	PCTL
5	GotoStep1.	PCTL is in a transitional state and is not in any of the previous operational states.
6	Write WAKEUP to CTL.state_cmd and poll STAT.ctl_stat = Access.	Issues SRX, moves to the Access state, and updates STAT.ctl_stat = Access when complete.
7	Write CFG to SCTL.state_cmd and poll STAT.ctl_stat = ConFig	PCTL stalls the NIF; completes any pending transaction; issues PREA if required; moves into the ConFig state; and updates STAT.ctl_stat = ConFig when complete.
END		PCTLis in ConFig state.

2.5.1.3 Moving to Access State

Table 2-3 moving to the Access state

Step	Application	PCTL
1	Read STAT register	Returns the current uPCTL state
2	If STAT.ctl_stat = Access, go to END.	
3	If STAT.ctl_stat = Config, go to Step9	
4	If STAT.ctl_stat = Init_mem, go to Step8	
5	If STAT.ctl_stat = Low_power, go to Step7.	
6	Go to Step1.	PCTL is in a transitional state and is not in any of the previous operational states.
7	Write WAKEUP to SCTL.state_cmd and poll STAT.ctl_stat = Access. Go to END	Issues SRX, moves to the Access state, updates STAT.ctl_stat = Access when complete.
8	Write CFG to SCTL.state_cmd and poll STAT.ctl_stat = ConFig	Moves into the ConFig state, updates STAT.ctl_stat = ConFig when complete.
9	Write GO to SCTL.state_cmd and poll STAT.ctl_stat = Access.	Moves into the Access state, updates STAT.ctl_stat= Access when complete.
END		PCTL is in Access state.

2.5.1.4 Moving to Low Power State

Table 2-4 moving to the Low_power state

Step	Application	PCTL
1	Read STAT register	Returns the current uPCTL state
2	If STAT.ctl_stat = Low_power, go to END.	
3	If STAT.ctl_stat = Access, go to Step9	
4	If STAT.ctl_stat = Config, go to Step8	
5	If STAT.ctl_stat = Init_mem, go to Step7.	
6	Go to Step1.	PCTL is in transitional state and is not in any of the previous operational states.
7	Write CFG to SCTL.state_cmd and poll STAT.ctl_stat=ConFig	Moves into the ConFig state, updates STAT.ctl_stat = ConFig when complete.
8	Write GO to SCTL.state_cmd and	Moves into the Access state, updates

Step	Application	PCTL
	poll STAT.ctl_stat= Access.	STAT.ctl_stat = Access when complete.
9	Write SLEEP to SCTL.state_cmd and poll STAT.ctl_stat = Low_power.	Issues PDX if necessary; completes any pending transactions; issues PREA command; finally, issues SRE and updates STAT.ctl_stat = Low_power.
END		PCTL is in Low Power state

2.5.2 Initialization

2.5.2.1 DMC Initialization

DDR PHY power-up reset sequence:

- Configure relative system information of GRF_UPCTL_CON0 and GRF_SOC_CON12 register. For a total memory data width of 16 bits, either GRF_UPCTL_CON0[0] or GRF_SOC_CON12[1] should be set to 1’b1, but they should not be set 1’b1 at the same time. If the external memory type is DDR3, the GRF_SOC_CON12[0] should be set to 1’b1 to match the MSCH_DdrConf, otherwise GRF_SOC_CON12[0] should be set to 1’b0. Detailed description for MSCH_DdrConf refers to Interconnect.
- Configure AL, CL… information of PHY registers.
- Configure timing registers of PCTL.
- Start PHY initialization with DDR_PCTL_DFISTCFG0[0] register of uPCTL and wait PHY initialization finish with DDR_PCTL_DFISTSTAT0[0] register of PCTL.
- Power up DRAM by DDR_PCTL_POWCTL[0] register of uPCTL and wait power up DRAM finish with DDR_PCTL_POWSATA[0] register of PCTL.
- Initiate DRAM with MRS command sent by PCTL.
- After DRAM initialization done, start PHY dqs calibration with PHYREG02 register and wait calibration finish with PHYREGFF register.
- (Optional) After dqs calibration, start write leveling training with PHYREG02 register and wait write leveling training finish with PHYJREGF0 register.
- Configure uPCTL to move to Access State.
- Start Write and Read.

2.5.2.2 DDR2 Initialization Sequence

The initialization steps for DDR2 DRAMs are as follows:

- Write Deselect command to MCMD, specifying an additional delay in terms of internal timers clock cycles of at least 400ns.
- Poll MCMD.start_cmd=0.
- Write PREA command to MCMD and poll MCMD.start_cmd=0.
- Write MR2 command to MCMD and poll MCMD.start_cmd=0.
- Write MR3 command to MCMD and poll MCMD.start_cmd=0.
- Write MR1 command to MCMD and poll MCMD.start_cmd=0.
- Write MR command to MCMD, for DLL Reset, and poll MCMD.start_cmd=0. Initiate a timer for counting 200 ddr controller clock cycles.
- Write PREA command to MCMD, and poll MCMD.start_cmd=0.
- Write REF command to MCMD, and poll MCMD.start_cmd=0.
- Repeat Step 9 at least once.
- Write MR command to MCMD, to initialize device operation, and poll MCMD.start_cmd=0.
- At least 200 ddr controller clock cycles after Step 7 (DLL Reset), write MR1 command, for OCD Default Command, to MCMD. Poll MCMD.start_cmd=0.
- Write MR1 command to MCMD, for OCD Calibration Mode Exit command. Poll MCMD.start_cmd=0.

2.5.2.3 DDR3 Initialization Sequence

The initialization steps for DDR3 DRAMs are as follows:

- Write Deselect command to MCMD, specifying an additional delay in terms of internal timers clock cycles of at least tXPR.
- Poll MCMD.start_cmd=0.
- Write MR2 command to MCMD and poll MCMD.start_cmd=0.
- Write MR3 command to MCMD and poll MCMD.start_cmd=0.

- Write MR1 command to MCMD and poll MCMD.start_cmd=0.
- Write MR0 command to MCMD, for "DLL Reset", and poll MCMD.start_cmd=0.
- Write ZQCL command to MCMD and poll MCMD.start_cmd=0. If MCFG1.zq_resistor_shared=1 ensure that ZQCL commands are sent to one rank at a time.

2.5.2.4 LPDDR2 Initialization Sequence

The initialization steps for LPDDR2 DRAMs are as follows:

- Write MRW(Reset) command to MCMD and poll MCMD.start_cmd=0. MRW(Reset) is to Mode Register Address 0x3F and the value should be 0x00. So MCMD.lpddr23_addr = 0x003F.
- Write Deselect command to MCMD, specifying an additional delay in terms of internal timers clock cycles of at least tINIT5=10 us.
- If ZQ Calibration command is supported, Write MRW(ZQ initialization calibration) command to MCMD and poll MCMD.start_cmd=0. MRW(ZQ initialization calibration) is to Mode Register Address 0x0A and the value should be 0xFF. So MCMD.lpddr23_addr = 0XFF0A. Set MCMD.cmd_add_del such that tZQINIT=1us is met before next step. If MCFG1.zq_resistor_shared=1 ensure that ZQINIT commands are sent to one rank at a time.
- Write MR2 command to MCMD and poll MCMD.start_cmd=0.
- Write MR1 command to MCMD and poll MCMD.start_cmd=0.
- Write MR3 command to MCMD and poll MCMD.start_cmd=0.

2.5.3 Low Power Operation

Low_power state can be entered/exited via following ways:

- Software control of uPCTL State machine (highest priority)
- Hardware Low Power Interface (middle priority)
- Auto Self-Refresh feature (lowest priority)

Note the priority of requests from Access to Low_power is highlighted above. The STAT.lp_trig register field reports which of the 3 requests caused the entry to Low_power state.

2.5.3.1 Software control of uPCTL State

The application can request via software to enter the memories into Self-Refresh state by issuing the SLEEP command by programming SCTL.PCTL responds to the software request by moving into the Low_power operational state and issuing the SRE command to the memories. Note that the Low_power state can only be reached from the Access state.
In a similar fashion, the application requests to exit the memories from Self-Refresh by issuing a WAKEUP command by programming SCTL. uPCTL responds to the WAKEUP command issuing SRX and restoring normal NIF address channel operation.

2.5.3.2 Hardware Low Power Interface

The hardware low power interface can also be used to enter/exit Self-Refresh. The functionality is enabled by setting SCFG.hw_low_power_en=1. Once that bit is set, the input c_sysreq(controlled by GRF_UPCTL_CON0[5] or GRF_UPCTL_CON0[6]) has the ability to trigger entry into the Low Power configuration state just like the software methodology (SCTL.state_cmd=SLEEP). A hardware Low Power entry trigger will be ignored/denied if the input c_active_in=1 (controlled by GRF_UPCTL_CON0[4]) or n_avalid=1(there exits read or write operation to DDR). It may be accepted if c_active_in=0 and n_avalid=0, depending on the current state of the PCTL.
The c_active which represents the state of uPCTL could also be used by an external Low Power controller to decide when to request a transition to low power. When MCFG1.hw_idle > 0, c_active=1'b0 indicates that the NIF has been idle for at least MCFG1.hw_idle * 32 * n_clk (PCTL's working clock) cycles while in the Access state.
When in low power the c_active output can used by an external Low Power controller to trigger a low power exit. c_active will be driven high when either c_active_in or n_avalid are high. The Low Power controller should re-enable the clocks when c_active is driven high while in the Low_power state.

2.5.3.3 Auto Power Down/Self-Refresh

The Power Down and/or Self-Refresh sequence is automatically started by uPCTL when the NIF address channel is idle for a number of cycles, depending on the programmed value in

MCFG.pd_idle and MCFG1.sr_idle.
Following table outlines the effect of these settings in conjunction with NIF being idle.

Table 2-5 The effect of pd_idle and sr_idle with NIF being idle

pd_idle	sr_idle	Memory modes	Memory Type
0	0	none	All
>0	0	Power Down	All
0	>0	Self-Refresh	All
>0	>0	Power Down -> Self Refresh3	All

Note:
1. Power Down is entered if NIF is idle for pd_idle.
*2. Following on from that, if NIF continues to be idle for a further sr_idle*32 cycles, Power Down is exited and Self-Refresh is entered.*

2.5.3.4 Removing PCTL's n_clk(PCTL's working clock)

The relationship between SRE/SRX and stopping/starting the memory clock (CK) are formalized and are accounted for automatically by PCTL. CK could be stopped after uPCTL has reached the Low_power state. The current operational state can be verified by reading STAT.ctl_stat. The CK must be started and stable before the Software or Hardware Low Power Interface attempts to take the memory out of Self-Refresh.
PCTL's n_clk can be safely removed when uPCTL is in Low Power state. The sequences outlined in following two tables should be followed for safe operation:

Table 2-6 The software sequences to remove PCTL's n_clk safely

Step	Application	PCTL
1	Write SLEEP to SCTL.state_cmd and poll STAT.ctl_stat = LOW_POWER.	Tells uPCTL to move memories into Self-Refresh and waits until this completes.
2	Write TREFI=0. Also, write DFITCRLUPDI=0 and DFIREFMSKI=0, if they are not already 0.	Stops any MC-driven DFI updates occurring internally with PCTL
3	Wait a minimum interval which is equivalent to the PCTL's Refresh Interval (previous value of TREFI*TOGCNT100N*internal timers clock period;	Ensures any already scheduled PHY/PVT updates have completed successfully.
4	Stop toggling n_clk to PCTL.	n_clk logic inside uPCTL is stopped.
end		

Table 2-7 The hardware sequences to remove PCTL's n_clk safely

Step	Application	PCTL
1	Drive c_active_in low	Confirms that system external to uPCTL can accept a Low- power request
2	Drive c_sysreq low	System Low-power request
3	Wait for uPCTL to drive c_sysack low	PCTL Low-power request acknowledgement
4	Check value of c_active when Step 3 occurs. - if c_active=1, request denied. Cannot remove n_clk. Go to END. - if c_active=0, request accepted.	PCTL low-power request status response
5	Stop toggling n_clk to PCTL	n_clk logic inside uPCTL is stopped

end		

2.5.4 PHY TX DLLs

All high speed IO signals’ phase can be adjusted by PHY TX DLLs. Following table illustrates these DLLs.

Table 2-8 DDR PHY TX DLLs Delay Step

Offset	Bit	Control Signal Phase	Default	Description
0x4c	2~0	CMD and Address	0x4	CMD DLL delay step
0x50	2~0	CK	0x0	CK DLL delay step
0x98	2~0	A_DM0, A_DQ7~A_DQ0	0x4	DM and DQ DLL Signal delay step
0xd8	2~0	A_DM1, A_DQ15~A_DQ8	0x4	
0x9c	2~0	A_DQS0, A_DQS0N	0x0	TX DQS DLL Signal delay step
0xdc	2~0	A_DQS1, A_DQS1N	0x0	

Notes: The delay between CMD/Address and CK is fixed at 180 degree for DDR2/DDR3 and can’t be adjusted through configuring 0x4c.

Step 0x0 values means no phase delay, and 0x4 increases delay phase to 90 deg, 0x7 values corresponds to maximum phase delay. All DLLs having 8 delay steps which can get 90 deg phase delay by setting 0x4.

2.5.5 PHY RX DLLs

The RX DLLs are used for sample RX DQS signals with proper phase delay and pulse edges. The DQS squelch (Rx Squelch) signal opens a window for passing RX DQS pulses, both RX DQS and DQS squelch signal phase can be adjusted by corresponding DLLs.

Table 2-9 DDR PHY RX DQS Delay Step

Offset	Bit	Control Signal Phase	Default	Description
0x0a0	1~0	DDR_DQS0,DDR_DQS0N	0x1	Read DQS DLL delay phase
0x0e0	1~0	DDR_DQS1,DDR_DQS1N	0x1	
0x0b0	7~5	left channel A DQS gating	0x1	CS0 DQS gating delay, unit x1 clock cycle
0x0f0	7~5	right channel A DQS gating	0x1	
0x0b4	7~5	left channel A DQS gating	0x1	CS1 DQS gating delay, unit x1 clock cycle
0x0f4	7~5	right channel A DQS gating	0x1	
0x0b0	4~3	left channel A DQS gating	0x1	CS0 additive and accumulative DQS gating delay, unit 4x clock cycle
0x0f0	4~3	right channel A DQS gating	0x1	
0x0b4	4~3	left channel A DQS gating	0x1	CS1 additive and accumulative DQS gating delay, unit 4x clock cycle
0x0f4	4~3	right channel A DQS gating	0x1	
0x0b0	2~0	left channel A DQS gating	0x4	CS0 additive and accumulative DQS gating delay, unit per DLL step
0x0f0	2~0	right channel A DQS gating	0x4	
0x0b4	2~0	left channel A DQS gating	0x4	CS1 additive and accumulative DQS gating delay, unit per DLL step
0x0f4	2~0	right channel A DQS gating	0x4	

2.5.6 High Speed IO Drive Strength

The tuning rage of driver resistance is 19.6ohm to +∞. By default, 0x14 is 37.6ohm for DDR3 DQ and CMD driver. When the control bit is set to be larger, the drive strength becomes stronger.

Table 2-10 CK/CMD Signal Drive Strength Register

Offset	Bit	Default	Description
0x44	4~0	0x14	adjustable CMD pull-down resistance
0x48	7~3	0x14	adjustable CMD pull-up resistance

Offset	Bit	Default	Description
0x58	4~0	0x14	adjustable CK pull-down resistance
0x60	4~0	0x14	adjustable CK pull-up resistance
0x5c	6~4	0x4	Adjust CMD/CK falling edge slew rate
0x5c	2~0	0x4	Adjust CMD/CK rising edge slew rate

Table 2-11 DM, DQ Signal Drive Strength Register

Offset	Bit	Default	Description
0x80	4~0	0x14	pull-down driving resistance for A_DQ0~A_DQ7
0xbc	4~0	0x14	pull-up driving resistance for A_DQ0~A_DQ7
0xc0	4~0	0x14	pull-down driving resistance for A_DQ8~A_DQ15
0xfc	4~0	0x14	pull-up driving resistance for A_DQ8~A_DQ15
0xac	7~5	0x4	Falling edge slew rate control for A_DQ0~A_DQ7
0xac	4~2	0x4	Rising edge slew rate control for A_DQ0~A_DQ7
0xec	7~5	0x4	Falling edge slew rate control for A_DQ8~A_DQ15
0xec	4~2	0x4	Rising edge slew rate control for A_DQ8~A_DQ15
0x84	4~0	0x4	Pull-down ODT resistance for A_DQ0~A_DQ7
0xb8	4~0	0x4	Pull-up ODT resistance for A_DQ0~A_DQ7
0xc4	4~0	0x4	Pull-down ODT resistance for A_DQ8~A_DQ15
0xf8	4~0	0x4	Pull-up ODT resistance for A_DQ8~A_DQ15

The value is larger, the drive strength is stronger.

Table 2-12 DM/DQ/DQS/CMD Driver and ODT resistance with control bit(DDR3 1.5V)

Control bit	**00000**	**00001**	**00010**	**00011**	**00100**	**00101**	**00110**	**00111**	**Unit**
Pull-up resistance	+∞	451.6	225.8	150.5	112.9	90.3	75.3	64.5	ohm
Pull-down resistance	+∞	451.6	225.8	150.5	112.9	90.3	75.3	64.5	ohm
Control bit	**01000**	**01001**	**01010**	**01011**	**01100**	**01101**	**01110**	**01111**	**unit**
Pull-up resistance	56.5	50.2	45.2	41.1	37.6	34.7	33.3	30.1	ohm
Pull-down resistance	56.5	50.2	45.2	41.1	37.6	34.7	33.3	30.1	ohm
Control bit	**10000**	**10001**	**10010**	**10011**	**10100**	**10101**	**10110**	**10111**	**unit**
Pull-up resistance	56.5	50.2	45.2	41.1	37.6	34.7	33.3	30.1	ohm
Pull-down resistance	56.5	50.2	45.2	41.1	37.6	34.7	33.3	30.1	ohm
Control bit	**11000**	**11001**	**11010**	**11011**	**11100**	**11101**	**11110**	**11111**	**unit**
Pull-up resistance	28.2	26.6	25.1	23.8	22.6	21.5	20.5	19.6	ohm
Pull-down resistance	28.2	26.6	25.1	23.8	22.6	21.5	20.5	19.6	ohm

Table 2-13 DM/DQ/DQS/CMD Driver and ODT resistance with control bit(LPDDR2 1.2V)

Control bit	**00000**	**00001**	**00010**	**00011**	**00100**	**00101**	**00110**	**00111**	**Unit**
Pull-up resistance	+∞	480.4	240.2	160.1	120.1	96.1	80.1	68.6	oh

									m
Pull-down resistance	+∞	480.4	240.2	160.1	120.1	96.1	80.1	68.6	ohm
Control bit	**01000**	**01001**	**01010**	**01011**	**01100**	**01101**	**01110**	**01111**	**unit**
Pull-up resistance	60.0	53.4	48.0	43.7	40.0	37.0	34.3	32.0	ohm
Pull-down resistance	60.0	53.4	48.0	43.7	40.0	37.0	34.3	32.0	ohm
Control bit	**10000**	**10001**	**10010**	**10011**	**10100**	**10101**	**10110**	**10111**	**unit**
Pull-up resistance	60.0	53.4	48.0	43.7	40.0	37.0	34.3	32.0	ohm
Pull-down resistance	60.0	53.4	48.0	43.7	40.0	37.0	34.3	32.0	ohm
Control bit	**11000**	**11001**	**11010**	**11011**	**11100**	**11101**	**11110**	**11111**	**unit**
Pull-up resistance	30.0	28.3	26.7	25.3	24.0	22.9	21.8	20.9	ohm
Pull-down resistance	30.0	28.3	26.7	25.3	24.0	22.9	21.8	20.9	ohm

2.5.7 PHY Low Speed Mode (200MHz)

DDR PHY supports low speed to high speed by using two operating mode: normal delay line mode up to 800Mbps or more, low power mode where we support any speed up to 533Mbps. If all TX DLLs are bypassed, the PHY will enter low power state.

The DDR PHY entries low power mode when setting DLLs into Bypass mode. Following table illustrates related register settings.

Table 2-14 Low Power DLL Setting

Offset	Bit	Default	Low power Setting	Description
0x290	4	-	-	-
	3	-	-	-
	2	0x0	0x1	right channel A TX DQ DLL in bypass mode
	1	0x0	0x1	left channel A TX DQ DLL in bypass mode
	0	0x0	0x1	CMD/CK DLL in bypass mode
0x4c	4	0x0	0x1	CMD DLL phase select
0x50	3	0x0	0x0	CK DLL phase select
0x98	4	0x0	0x1	A_DQ0~A_DQ7 TX DLL phase select
0x9c	3	0x0	0x0	A_DQS0/A_DQSB0 TX DLL phase select
0xd8	4	0x0	0x1	A_DQ8~A_DQ15 TX DLL phase select
0xdc	3	0x0	0x0	A_DQS1/A_DQSB1 TX DLL phase select

2.5.8 Per bit de-skew tuning

Per-bit de-skew is designed for compensating PCB trace mismatch, DDR PHY support skew individually adjustable for all PHY signals. There are eight steps for each bit de-skew adjusting, and the adjust resolution under different corners is shown below:

Table 2-15 per-bit de-skew tuning resolution

	ff	tt	ss
de-skew resolution	15ps	22ps	32ps

Pre-bit de-skew is realized with inverter chain delay, per-bit de-skew control signals select how much inverters are connected to data path, the minimum resolution is determined by the two inverters minimum delay.

2.5.9 DDR PHY Calibration

DDR PHY auto dqs calibration function has been implemented in the PHY. The entire training processes only need to modify the register to start and wait for finish.

The entire training process is as follows:

- PHY's register is reset, the setup is complete.
- Send the initial command to dram and complete dram initialization.
- Set the PHY's register beginning calibration.

Table 2-16 PHY calibration register

Offset	Bit	Default	Description
0x8	7~2	0x0	Other register
	1	0x0	set calibration bypass mode(1:bypass mode; 0:nomal)
	0	0x0	set calibration start (1: start; 0: stop)

- Wait for the calibration finish by PHYREGFF.
- Normal read and writes operation can begin.

2.5.10 DDR Monitor

1. DDR read or write address monitor

DDR monitor module can store 4 consecutive read or write addresses in real time. We can read these addresses by APB bus for debug when system enters abnormal state.

The steps of configuration to monitor DDR read or write address:

- Configure DDRMON_DDR_IF_CTRL.direction to select storing read or write address.
- Set DDRMON_DDR_IF_CTRL.if_mon_en to '1' to enable DDR monitor.
- When system is abnormal, we can read the register to get the current four addresses.

2. DDR access address monitor within a specified range

Sometimes we want to confirm whether DDR read or write within a specified address range, then we can configure the address range and enable this function.

The steps of configuration to monitor DDR access address within a specified range:

Configure the write address range registers DDRMON_WR_START_ADDR, DDRMON_WR_END_ADDR, and read address registers DDRMON _RD_START_ADDR, DDRMON _RD_END_ADDR.

- Enable interrupt by configure the register DDRMON_INT_MASK[6:5] to 0.
- Set DDRMON_DDR_IF_CTRL.if_mon_en to '1' to enable DDR address monitor.
- If the read or write addresses hit the range, then interrupt will assert, and we can read the interrupt status register DDRMON_INT_STATUS.

3. DDR access command statistics

This module can do the statistics about DDR access command, like write, read and active by monitoring DFI interface. There are two mode to do statistics, hardware mode and software mode. Two thresholds can be set, if read and write command number is more than high threshold, or less than low threshold, the interrupt will be asserted.

Hardware mode

In hardware mode, a dfi timer is used to specify a statistics period, the command statistics is done in the statistics period. The dfi timer is running in 24MHz. After dfi timer counts to the threshold, and update the statistics value, the dfi timer will restart automatically, and count again.

The steps of hardware mode of DDR access command statistics:

- Configure register DDRMON_CTRL.hardware_en as '1' to enable hardware mode.
- Configure register DDRMON_TIMER_COUNT to set the dfi timer count threshold, the statistics is done in the period of timer being less than the value of DDRMON_TIMER_COUNT.
- Configure register DDRMON_CTRL.lpddr23_en and DDRMON_CTRL.lpddr4_en to set the DDR mode:

Table 2-17 DDRMON_CTRL register

DDRMON_CTRL.lpddr23_en	DDRMON_CTRL.lpddr4_en	DDR mode
1	0	LPDDR2/LPDDR3
0	0	DDR3
0	1	LPDDR4

- Configure register DDRMON_FLOOR_NUMBER to specify the low threshold of interrupt, and configure register DDRMON_TOP_NUMBER to specify the high threshold of interrupt.
- Configure register DDRMON_CTRL.timer_cnt_en as '1' to start hardware mode.
- Wait for the interrupt to do following process. We also can read the read, write and active command number separately.

Software mode

In software mode, the statistics is controlled by software.

The steps of hardware mode of DDR access command statistics:

- Configure register DDRMON_CTRL.lpddr23_en and DDRMON_CTRL.lpddr4_en to set the DDR mode like hardware mode.
- Configure register DDRMON_CTRL.software_en as '1' to enable software mode statistics.
- Configure register DDRMON_CTRL.software_en as '0' to stop the statistics, and generate the statistics result. We can read the read, write and active command number separately.

2.5.11 DDR Standby Mode

The standby mode is enabled by register DDR_STDBY_CONTROL0 [0]. When DDR controller is idle, and after a period of waiting time, the standby mode will be activated, the clocks of DDR controller, PHY and memory scheduler can be gated. The waiting time is determined by the register DDR_STDBY_CONTROL0 [31:16]. It is the counter threshold by controller clock. The register DDR_STDBY_CONTROL0[7:4] is used to determine if the clock of memory schedule, PHYCTL or DDR controller will be gated when in standby status.

Chapter 3 Mobile Storage Host Controller

3.1 Overview

The Mobile Storage Host Controller is designed to support Secure Digital memory (SD-max version 3.01) with 1 bits or 4 bits data width, Multimedia Card(MMC-max version 4.51) with 1 bits or 4 bits or 8 bits data width.
The Host Controller is instantiated for SDMMC, SDIO and EMMC. The interface difference between these instances is shown in "Interface Description".
The Host Controller supports following features:

- Bus Interface Features:
  - Support AMBA AHB interface for master and slave
  - Supports internal DMA interface(IDMAC)
    - Supports 16/32-bit data transfers
    - Single-channel; single engine used for Transmit and Receive, which are mutually exclusive
    - Dual-buffer and chained descriptor linked list
    - Each descriptor can transfer up to 4KB of data in chained mode and 8KB of data in dual-buffer mode
    - Programmable burst size for optimal host bus utilization
  - Support combined single FIFO for both transmit and receive operations
  - Support FIFO size of 256x32
  - Support FIFO over-run and under-run prevention by stopping card clock

7

- Card Interface Features:
  - Support Secure Digital memory protocol commands
  - Support Secure Digital I/O protocol commands
  - Support Multimedia Card protocol commands
  - Support Command Completion Signal and interrupts to host
  - Support CRC generation and error detection
  - Support programmable baud rate
  - Support power management and power switch
  - Support card detection
  - Support write protection
  - Support hardware reset
  - Support SDIO interrupts in 1-bit and 4-bit modes
  - Support 4-bit mode in SDIO3.0
  - Support SDIO suspend and resume operation
  - Support SDIO read wait
  - Support block size of 1 to 65,535 bytes
  - Support 1-bit, 4-bit and 8-bit SDR modes
  - Support boot in 1-bit, 4-bit and 8-bit SDR modes
  - Support Packed Commands, CMD21, CMD49

8

- Clock Interface Features:
  - Support 0/90/180/270-degree phase shift operation for sample clock (cclk_in_sample) and drive clock(cclk_in_drv) relative to function clock(cclk_in) respectively
  - Support phase tuning using delay line for sample clock(cclk_in_sample) and drive clock(cclk_in_drv) relative to function clock (cclk_in) respectively. The max number of delay element number is 256

3.2 Block Diagram

The Host Controller consists of the following main functional blocks.

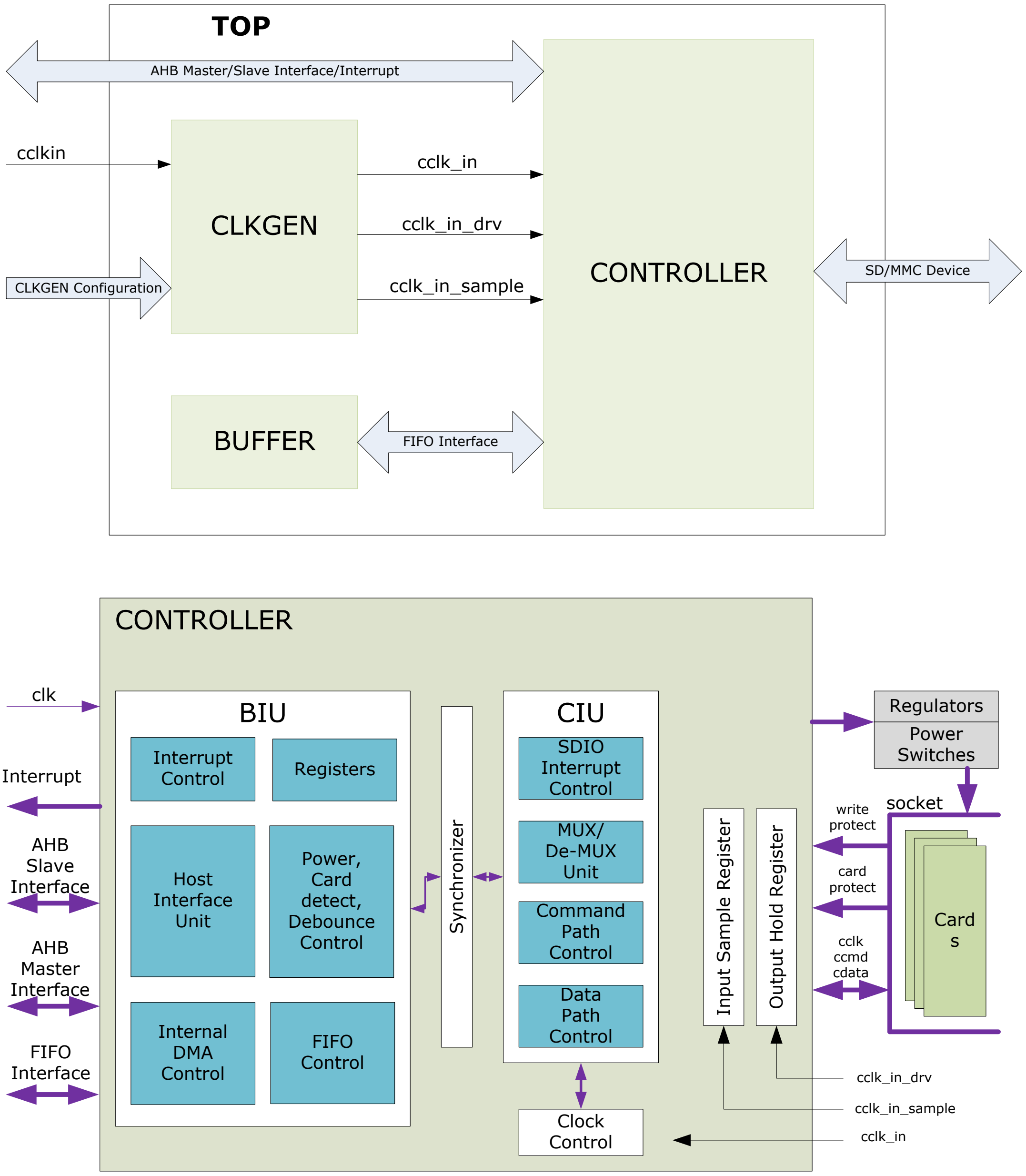


Fig. 3-1 Host Controller Block Diagram

- Clock Generate Unit(CLKGEN): generates card interface clock cclk_in/ cclk_sample/cclk_drv based on cclkin and configuration information.
- Asynchronous dual-port memory(BUFFER): Uses a two-clock synchronous read and synchronous write dual-port RAM. One of the ports is connected to the host clock, and the second port is connected to the card clock.
- Bus Interface Unit (BIU): Provides AMBA AHB interfaces for register and data read/writes.
- Card Interface Unit (CIU): Takes care of the SD/MMC protocols and provides clock management.

3.3 Function Description

3.3.1 Bus Interface Unit

The Bus Interface Unit provides the following functions:

- Host interface
- Interrupt control
- Register access
- External FIFO access
- Power control and card detection

1. Host Interface Unit

The Host Interface Unit is an AHB slave interface, which provides the interface between the SD/MMC card and the host bus.

2. Register Unit

The register unit is part of the bus interface unit; it provides read and write access to the registers.
All registers reside in the Bus Interface Unit clock domain. When a command is sent to a card by setting the start_bit, which is bit[31] of the SDMMC_CMD register, all relevant registers needed for the CIU operation are transferred to the CIU block. During this time, the registers that are transferred from the BIU to the CIU should not be written. The software should wait for the hardware to clear the start bit before writing to these registers again. The register unit has a hardware locking feature to prevent illegal writes to registers. The lock is necessary in order to avoid metastability violations, both because the host and card clock domains are different and to prevent illegal software operations.
Once a command start is issued by setting the start_bit of the SDMMC_CMD register, the following registers cannot be reprogrammed until the command is accepted by the card interface unit:

- SDMMC_CMD – Command
- SDMMC_CMDARG – Command Argument
- SDMMC_BYTCNT – Byte Count
- SDMMC_BLKSIZ – Block Size
- SDMMC_CLKDIV – Clock Divider
- SDMMC_CLKENA – Clock Enable
- SDMMC_CLKSRC – Clock Source
- SDMMC_TMOUT – Timeout
- SDMMC_CTYPE – Card Type

The hardware resets the start_bit once the CIU accepts the command. If a host write to any of these registers is attempted during this locked time, then the write is ignored and the hardware lock error bit is set in the raw interrupt status register. Additionally, if the interrupt is enabled and not masked for a hardware lock error, then an interrupt is sent to the host.
When the Card Interface Unit is in an idle state, it typically takes the following number of clocks for the command handshake, where clk is the BIU clock and cclk_in is the CIU clock:
3 (clk) + 3 (cclk_in)
Once a command is accepted, you can send another command to the CIU-which has a one-deep command queue-under the following conditions:

- If the previous command was not a data transfer command, the new command is sent to the SD/MMC card once the previous command completes.
- If the previous command is a data transfer command and if wait_prvdata_complete (bit[13]) of the Command register is set for the new command, the new command is sent to the SD/MMC card only when the data transfer completes.
- If the wait_prvdata_complete is 0, then the new command is sent to the SD/MMC card as soon as the previous command is sent. Typically, you should use this only to stop or abort a previous data transfer or query the card status in the middle of a data transfer.

3. Interrupt Controller Unit

The interrupt controller unit generates an interrupt that depends on the controller raw interrupt status, the interrupt-mask register, and the global interrupt-enable register bit. Once an interrupt condition is detected, it sets the corresponding interrupt bit in the raw interrupt status register. The raw interrupt status bit stays on until the software clears the bit by writing a 1 to the interrupt bit; a 0 leaves the bit untouched.
The interrupt port, int, is an active-high, level-sensitive interrupt. The interrupt port is active only when any bit in the raw interrupt status register is active, the corresponding interrupt mask bit is 1, and the global interrupt enable bit is 1. The interrupt port is registered in order to avoid any combinational glitches.
The int_enable is reset to 0 on power-on, and the interrupt mask bits are set to 32’h0, which

masks all the interrupts.

Notes:

Before enabling the interrupt, it is always recommended that you write 32'hffff_ffff to the raw interrupt status register in order to clear any pending unserviced interrupts. When clearing interrupts during normal operation, ensure that you clear only the interrupt bits that you serviced.

The SDIO Interrupts, Receive FIFO Data Request (RXDR), and Transmit FIFO Data Request (TXDR) are set by level-sensitive interrupt sources. Therefore, the interrupt source should be first cleared before you can clear the interrupt bit of the Raw Interrupt register. For example, on seeing the Receive FIFO Data Request (RXDR) interrupt, the FIFO should be emptied so that the "FIFO count greater than the RX-Watermark" condition, which triggers the interrupt, becomes inactive. The rest of the interrupts are triggered by a single clock-pulse-width source.

Table 3-1 Bits in Interrupt Status Register

Bits	Interrupt	Description
24	sdio_interrupt	Interrupt from SDIO card. In MMC-Ver3.3-only mode, these bits are always 0
16	Card no-busy	If card exit busy status, the interrupt happened
15	End Bit Error (read) /Write no CRC (EBE)	Error in end-bit during read operation, or no data CRC received during write operation. For MMC CMD19, there may be no CRC status returned by the card. Hence, EBE is set for CMD19. The application should not treat this as an error.
14	Auto Command Done (ACD)	Stop/abort commands automatically sent by card unit and not initiated by host; similar to Command Done (CD) interrupt. Recommendation: Software typically need not enable this for non CE-ATA accesses; Data Transfer Over (DTO) interrupt that comes after this interrupt determines whether data transfer has correctly competed.
13	Start Bit Error (SBE)	Error in data start bit when data is read from a card. In 4-bit mode, if DAT[0] line indicates start bit-that is, 0-and any of the other data bits do not have start bit, then this error is set. Busy Complete Interrupt when data is written to the card. This interrupt is generated after completion of busy driven by the card after the last data block is written into the card.
12	Hardware Locked write Error (HLE)	During hardware-lock period, write attempted to one of locked registers. When software sets the start_cmd bit in the SDMMC_CMD register, the Host Controller tries to load the command. If the command buffer is already filled with a command, this error is raised. The software then has to reload the command.
11	FIFO Underrun/ Overrun Error (FRUN)	Host tried to push data when FIFO was full, or host tried to read data when FIFO was empty. Typically this should not happen, except due to error in software. Card unit never pushes data into FIFO when FIFO is full, and pop data when FIFO is empty. If IDMAC is enabled, FIFO underrun/overrun can occur due to a programming error on MSIZE and watermark values in SDMMC_FIFOTH register.
10	Data Starvation by Host Timeout (HTO)	To avoid data loss, card clock out is stopped if FIFO is empty when writing to card, or FIFO is full when reading from card. Whenever card clock is stopped to avoid data loss, data-starvation timeout counter is started with data-timeout value. This interrupt is set if host does not fill data into FIFO during write to card,

Bits	Interrupt	Description
		or does not read from FIFO during read from card before timeout period. Even after timeout, card clock stays in stopped state, with CIU state machines waiting. It is responsibility of host to push or pop data into FIFO upon interrupt, which automatically restarts cclk_out and card state machines. Even if host wants to send stop/abort command, it still needs to ensure it has to push or pop FIFO so that clock starts in order for stop/abort command to send on cmd signal along with data that is sent or received on data line.
9	Data Read Timeout (DRTO)	In Normal functioning mode: Data read timeout (DRTO) Data timeout occurred. Data Transfer Over (DTO) also set if data timeout occurs. In Boot Mode: Boot Data Start (BDS) When set, indicates that Host Controller has started to receive boot data from the card. A write to this register with a value of 1 clears this interrupt.
8	Response Timeout (RTO)	In normal functioning mode: Response timeout (RTO) Response timeout occurred. Command Done (CD) also set if response timeout occurs. If command involves data transfer and when response times out, no data transfer is attempted by Host Controller. In Boot Mode: Boot Ack Received (BAR) When expect_boot_ack is set, on reception of a boot acknowledge pattern—0-1-0—this interrupt is asserted. A write to this register with a value of 1 clears this interrupt.
7	Data CRC Error (DCRC)	Received Data CRC does not match with locally-generated CRC in CIU. Can also occur if the Write CRC status is incorrectly sampled by the Host.
6	Response CRC Error (RCRC)	Response CRC does not match with locally-generated CRC in CIU.
5	Receive FIFO Data Request (RXDR)	Interrupt set during read operation from card when FIFO level is greater than Receive-Threshold level. Recommendation: In DMA modes, this interrupt should not be enabled. In non-DMA mode: pop RX_WMark + 1 data from FIFO.

4	Transmit FIFO Data Request (TXDR)	Interrupt set during write operation to card when FIFO level reaches less than or equal to Transmit-Threshold level.
		Recommendation:
		In DMA modes, this interrupt should not be enabled.
		In non-DMA mode:
		if (pending_bytes > (FIFO_DEPTH - TX_WMark))
		        push (FIFO_DEPTH - TX_WMark) data into FIFO
		else
		        push pending_bytes data into FIFO

3	Data Transfer Over (DTO)	Indicates Data transfer completed. Though on detection of errors-Start Bit Error, Data CRC error, and so on, DTO may or may not be set; the application must issue CMD12, which ensures that DTO is set.

Bits	Interrupt	Description
		Recommendation: In non-DMA mode, when data is read from card, on seeing interrupt, host should read any pending data from FIFO. In DMA mode, DMA controllers guarantee FIFO is flushed before interrupt. DTO bit is set at the end of the last data block, even if the device asserts MMC busy after the last data block.
2	Command Done(CD)	Command sent to card and got response from card, even if Response Error or CRC error occurs.
1	Response Error (RE)	Error in received response set if one of following occurs: ● Transmission bit != 0 ● Command index mismatch ● End-bit != 1
0	Card-Detect (CDT)	When one or more cards inserted or removed, this interrupt occurs. Software should read card-detect register to determine current card status. Recommendation: After power-on and before enabling interrupts, software should read card detect register and store it in memory. When interrupt occurs, it should read card detect register and compare it with value stored in memory to determine which card(s) were removed/inserted. Before exiting ISR, software should update memory with new card-detect value.

4. FIFO Controller Unit

The FIFO controller interfaces the external FIFO to the host interface and the card controller unit. When FIFO overrun and under-run conditions occur, the card clock stops in order to avoid data loss.

The FIFO uses a two-clock synchronous read and synchronous write dual-port RAM. One of the ports is connected to the host clock, clk, and the second port is connected to the card clock, cclk_in.

Notes: The FIFO controller does not support simultaneous read/write access from the same port. For debugging purposes, the software may try to write into the FIFO and read back the data; results are indeterminate, since the design does not support read/write access from the same port.

5. Power Control and Card Detection Unit

The register unit has registers that control the power. Power to each card can be selectively turned on or off.

The card detection unit looks for any changes in the card-detect signals for card insertion or card removal. It filters out the debounces associated with mechanical insertion or removal, and generates one interrupt to the host. You can program the debounce filter value.

On power-on, the controller should read in the card_detect port and store the value in the memory. Upon receiving a card-detect interrupt, it should again read the card_detect port and XOR with the previous card-detect status to find out which card has interrupted. If more than one card is simultaneously removed or inserted, there is only one card-detect interrupt; the XOR value indicates which cards have been disturbed. The memory should be updated with the new card-detect value.

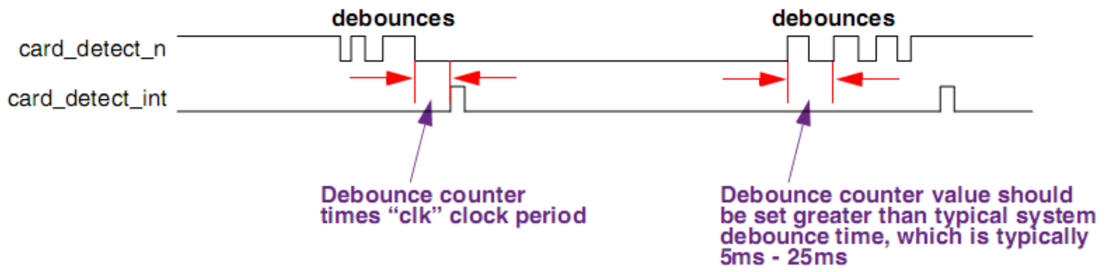

debounces
debounces
card_detect_n
card_detect_int
Debounce counter times "clk" clock period
Debounce counter value should be set greater than typical system debounce time, which is typically 5ms - 25ms

Fig. 3-2 SD/MMC Card-Detect Signal

6. DMA Interface Unit

DMA signals interface the Host Controller to an external DMA controller to reduce the software overhead during FIFO data transfers. The DMA request/acknowledge handshake is used for only data transfers. The DMA interface provides a connection to the DMA Controller. On seeing the DMA request, the DMA controller initiates accesses through the host interface to read or write into the data FIFO. The Host Controller has FIFO transmit/receive watermark registers that you can set, depending on system latency. The DMA interface asserts the request in the following cases:

- Read from a card when the data FIFO word count exceeds the Rx-Watermark level
- Write to a card when the FIFO word count is less than or equal to the Tx-Watermark level

When the DMA interface is enabled, you can use normal host read/write to access the data FIFO.

3.3.2 Card Interface Unit

The Card Interface Unit (CIU) interfaces with the Bus Interface Unit (BIU) and the devices. The host writes command parameters to the BIU control registers, and these parameters are then passed to the CIU. Depending on control register values, the CIU generates SD/MMC command and data traffic on a selected card bus according to SD/MMC protocol. The Host Controller accordingly controls the command and data path.

The following software restrictions should be met for proper CIU operation:

- Only one data transfer command can be issued at a time.
- During an open-ended card write operation, if the card clock is stopped because the FIFO is empty, the software must first fill the data into the FIFO and start the card clock. It can then issue only a stop/abort command to the card.
- When issuing card reset commands (CMD0, CMD15 or CMD52_reset) while a card data transfer is in progress, the software must set the stop_abort_cmd bit in the Command register so that the Host Controller can stop the data transfer after issuing the card reset command.
- When the data end bit error is set in the SDMMC_RINTSTS register, the Host Controller does not guarantee SDIO interrupts. The software should ignore the SDIO interrupts and issue the stop/abort command to the card, so that the card stops sending the read data.
- If the card clock is stopped because the FIFO is full during a card read, the software should read at least two FIFO locations to start the card clock.

The CIU block consists of the following primary functional blocks:

- Command path
- Data path
- SDIO interrupt control
- Clock control
- Mux/demux unit

1. Command Path

The command path performs the following functions:

- Loads clock parameters
- Loads card command parameters
- Sends commands to card bus (ccmd_out line)
- Receives responses from card bus (ccmd_in line)
- Sends responses to BIU

- Drives the P-bit on command line

A new command is issued to the Host Controller by programming the BIU registers and setting the start_cmd bit in the Command register. The BIU asserts start_cmd, which indicates that a new command is issued to the SD/MMC device. The command path loads this new command (command, command argument, timeout) and sends acknowledge to the BIU by asserting cmd_taken.

Once the new command is loaded, the command path state machine sends a command to the device bus-including the internally generated CRC7-and receives a response, if any. The state machine then sends the received response and signals to the BIU that the command is done, and then waits for eight clocks before loading a new command.

Load Command Parameters

One of the following commands or responses is loaded in the command path:

- New command from BIU – When start_cmd is asserted, then the start_cmd bit is set in the Command register.
- Internally-generated auto-stop command – When the data path ends, the stop command request is loaded.
- IRQ response with RCA 0x000 – When the command path is waiting for an IRQ response from the MMC card and a "send irq response" request is signaled by the BIU, then the send_irq_response bit is set in the control register.

Loading a new command from the BIU in the command path depends on the following Command register bit settings:

- update_clock_registers_only – If this bit is set in the Command register, the command path updates only the clock enable, clock divider, and clock source registers. If this bit is not set, the command path loads the command, command argument, and timeout registers; it then starts processing the new command.
- wait_prvdata_complete – If this bit is set, the command path loads the new command under one of the following conditions:
  - ■ Immediately, if the data path is free (that is, there is no data transfer in progress), or if an open-ended data transfer is in progress (byte_count = 0).
  - ■ After completion of the current data transfer, if a predefined data transfer is in progress.

Send Command and Receive Response

Once a new command is loaded in the command path, update_clock_registers_only bit is unset – the command path state machine sends out a command on the device bus; the command path state machine is illustrated in following figure.

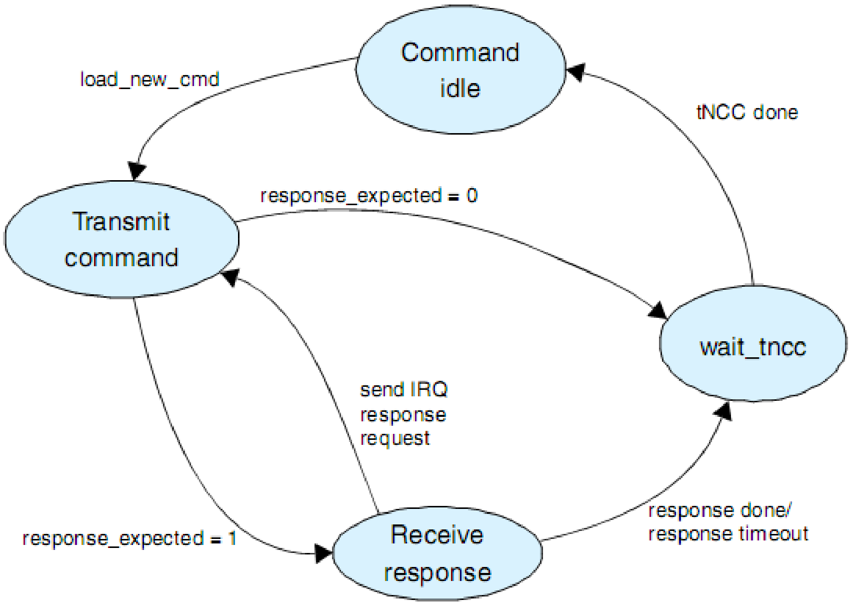


Fig. 3-3 Host Controller Command Path State Machine

The command path state machine performs the following functions, according to Command register bit values:

- send_initialization – Initialization sequence of 80 clocks is sent before sending the command.

- response_expected – Response is expected for the command. After the command is sent out, the command path state machine receives a 48-bit or 136-bit response and sends it to the BIU. If the start bit of the card response is not received within the number of clocks programmed in the timeout register, then the response timeout and command done bit is set in the Raw Interrupt Status register as a signal to the BIU. If the response-expected bit is not set, the command path sends out a command and signals a response done to the BIU; that is, the command done bit is set in the Raw Interrupt Status register.
- response_length – If this bit is set, a 136-bit response is received; if it is not set, a 48-bit response is received.
- check_response_crc – If this bit is set, the command path compares CRC7 received in the response with the internally-generated CRC7. If the two do not match, the response CRC error is signaled to the BIU; that is, the response CRC error bit is set in the Raw Interrupt Status register.

Send Response to BIU

If the response_expected bit is set in the Command register, the received response is sent to the BIU. The Response0 register is updated for a short response, and the Response3, Response2, Response1, and Response0 registers are updated on a long response, after which the Command Done bit is set. If the response is for an auto_stop command sent by the CIU, the response is saved in the Response1 register, after which the Auto Command Done bit is set.

Additionally, the command path checks for the following:

- Transmission bit = 0
- Command index matches command index of the sent command
- End bit = 1 in received card response

The command index is not checked for a 136-bit response or if the check_response_crc bit is unset. For a 136-bit response and reserved CRC 48-bit responses, the command index is reserved-that is, 111111.

Polling Command Completion Signal

The device generates the Command Completion Signal in order to notify the host controller of the normal command completion or command termination.

Command Completion Signal Detection and Interrupt to Host Processor

If the ccs_expected bit is set in the Command register, the Command Completion Signal (CCS) from the device is indicated by setting the Data Transfer Over (DTO) bit in the SDMMC_RINTSTS register. The Host Controller generates a Data Transfer Over (DTO) interrupt if this interrupt is not masked.

Command Completion Signal Timeout

If the command expects a CCS from the device—if the ccs_expected bit is set in the Command register—the command state machine waits for the CCS and remains in a wait_CCSS state. If the device fails to send out the CCS, the host software should implement a timeout mechanism to free the command and data path. The host controller does not implement a hardware timer; it is the responsibility of the host software to maintain a software timer.

In the event of a CCS timeout, the host should issue a CCSD by setting the send_ccsd bit in the CTRL register. The host controller command state machine sends the CCSD to the device and exits to an idle state. After sending the CCSD, the host should also send a CMD12 to the device in order to abort the outstanding command.

Send Command Completion Signal Disable

If the send_ccsd bit is set in the CTRL register, the host sends a Command Completion Signal Disable (CCSD) pattern on the CMD line. The host can send the CCSD while waiting for the CCS or after a CCS timeout happens.

After sending the CCSD pattern, the host sets the Command Done (CD) bit in SDMMC_RINTSTS and also generates an interrupt to the host if the Command Done interrupt is not masked.

2. Data Path

The data path block pops the data FIFO and transmits data on cdata_out during a write data transfer, or it receives data on cdata_in and pushes it into the FIFO during a read data transfer. The data path loads new data parameters—that is, data expected, read/write data transfer, stream/block transfer, block size, byte count, card type, timeout registers—whenever a data transfer command is not in progress.

If the data_expected bit is set in the Command register, the new command is a data transfer command and the data path starts one of the following:

- Transmit data if the read/write bit = 1
- Data receive if read/write bit = 0

Data Transmit

The data transmit state machine, illustrated in following figure, starts data transmission two clocks after a response for the data write command is received; this occurs even if the command path detects a response error or response CRC error. If a response is not received from the card because of a response timeout, data is not transmitted. Depending upon the value of the transfer_mode bit in the Command register, the data transmit state machine puts data on the card data bus in a stream or in block(s).

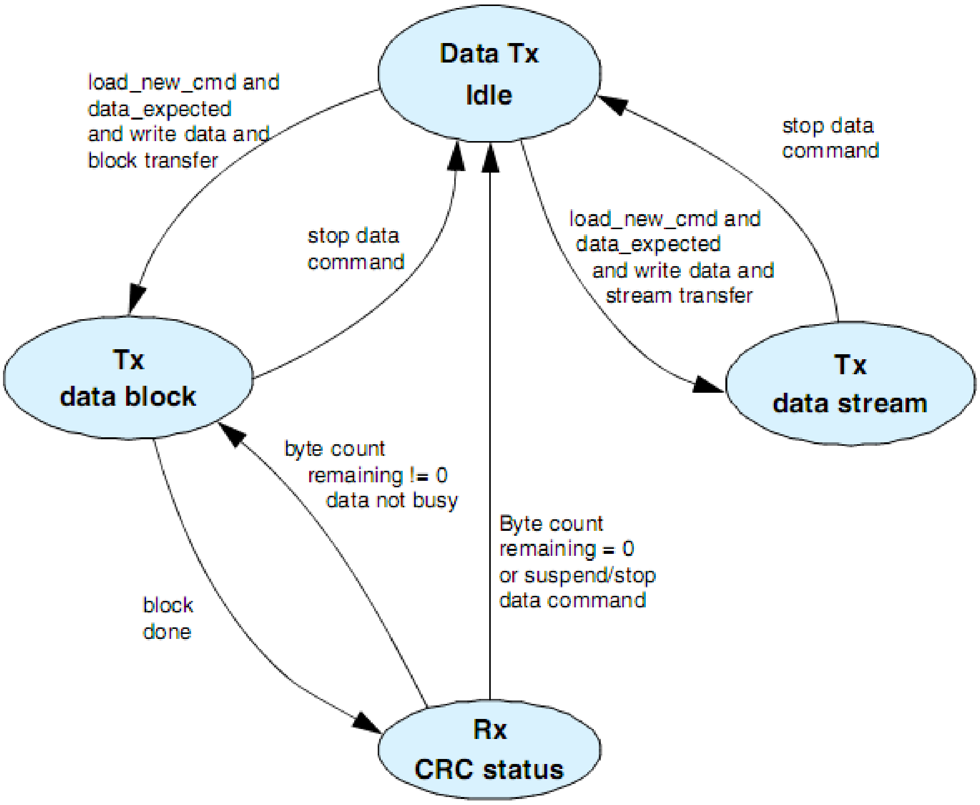

Data Tx Idle
load_new_cmd and data_expected and write data and block transfer
stop data command
stop data command
load_new_cmd and data_expected and write data and stream transfer
Tx data block
Tx data stream
byte count remaining != 0 data not busy
Byte count remaining = 0 or suspend/stop data command
block done
Rx CRC status

Fig. 3-4 Host Controller Data Transmit State Machine

Stream Data Transmit

If the transfer_mode bit in the Command register is set to 1, it is a stream-write data transfer. The data path pops the FIFO from the BIU and transmits in a stream to the card data bus. If the FIFO becomes empty, the card clock is stopped and restarted once data is available in the FIFO.

If the byte_count register is programmed to 0, it is an open-ended stream-write data transfer. During this data transfer, the data path continuously transmits data in a stream until the host software issues a stop command. A stream data transfer is terminated when the end bit of the stop command and end bit of the data match over two clocks.

If the byte_count register is programmed with a non-zero value and the send_auto_stop bit is set in the Command register, the stop command is internally generated and loaded in the command path when the end bit of the stop command occurs after the last byte of the stream write transfer matches.

This data transfer can also terminate if the host issues a stop command before all the data bytes are transferred to the card bus.

Single Block Data

If the transfer_mode bit in the Command register is set to 0 and the byte_count register value is equal to the value of the block_size register, a single-block write-data transfer occurs. The data transmit state machine sends data in a single block, where the number of bytes equals the block size, including the internally-generated CRC16.

If the SDMMC_CTYPE register bit for the selected card – indicated by the card_num value in the Command register – is set for a 1-bit, 4-bit, or 8-bit data transfer, the data is transmitted on 1, 4, or 8 data lines, respectively, and CRC16 is separately generated and transmitted for 1, 4, or 8 data lines, respectively.

After a single data block is transmitted, the data transmit state machine receives the CRC status from the card and signals a data transfer to the BIU; this happens when the data-transfer-over bit is set in the SDMMC_RINTSTS register.

If a negative CRC status is received from the card, the data path signals a data CRC error to the BIU by setting the data CRC error bit in the SDMMC_RINTSTS register.

Additionally, if the start bit of the CRC status is not received by two clocks after the end of the data block, a CRC status start bit error is signaled to the BIU by setting the write-no-CRC bit in the SDMMC_RINTSTS register.

Multiple Block Data

A multiple-block write-data transfer occurs if the transfer_mode bit in the Command register is set to 0 and the value in the byte_count register is not equal to the value of the block_size register. The data transmit state machine sends data in blocks, where the number of bytes in a block equals the block size, including the internally-generated CRC16.

If the SDMMC_CTYPE register bit for the selected card – indicated by the card_num value in the Command register – is set to 1-bit, 4-bit, or 8-bit data transfer, the data is transmitted on 1, 4, or 8 data lines, respectively, and CRC16 is separately generated and transmitted on 1, 4, or 8 data lines, respectively.

After one data block is transmitted, the data transmit state machine receives the CRC status from the card. If the remaining byte_count becomes 0, the data path signals to the BIU that the data transfer is done; this happens when the data-transfer-over bit is set in the SDMMC_RINTSTS register.

If the remaining data bytes are greater than 0, the data path state machine starts to transmit another data block.

If a negative CRC status is received from the card, the data path signals a data CRC error to the BIU by setting the data CRC error bit in the SDMMC_RINTSTS register, and continues further data transmission until all the bytes are transmitted.

Additionally, if the CRC status start bit is not received by two clocks after the end of a data block, a CRC status start bit error is signaled to the BIU by setting the write-no-CRC bit in the SDMMC_RINTSTS register; further data transfer is terminated.

If the send_auto_stop bit is set in the Command register, the stop command is internally generated during the transfer of the last data block, where no extra bytes are transferred to the card. The end bit of the stop command may not exactly match the end bit of the CRC status in the last data block.

If the block size is less than 4, 16, or 32 for card data widths of 1 bit, 4 bits, or 8 bits, respectively, the data transmit state machine terminates the data transfer when all the data is transferred, at which time the internally generated stop command is loaded in the command path.

If the byte_count is 0 – the block size must be greater than 0 – it is an open-ended block transfer. The data transmit state machine for this type of data transfer continues the block-write data transfer until the host software issues a stop or abort command.

Data Receive

The data-receive state machine, illustrated in following figure, receives data two clock cycles after the end bit of a data read command, even if the command path detects a response error or response CRC error. If a response is not received from the card because a response timeout occurs, the BIU does not receive a signal that the data transfer is complete; this happens if the command sent by the Host Controller is an illegal operation for the card, which keeps the card from starting a read data transfer.

If data is not received before the data timeout, the data path signals a data timeout to the BIU and an end to the data transfer done. Based on the value of the transfer_mode bit in the Command register, the data-receive state machine gets data from the card data bus in a stream or block(s).

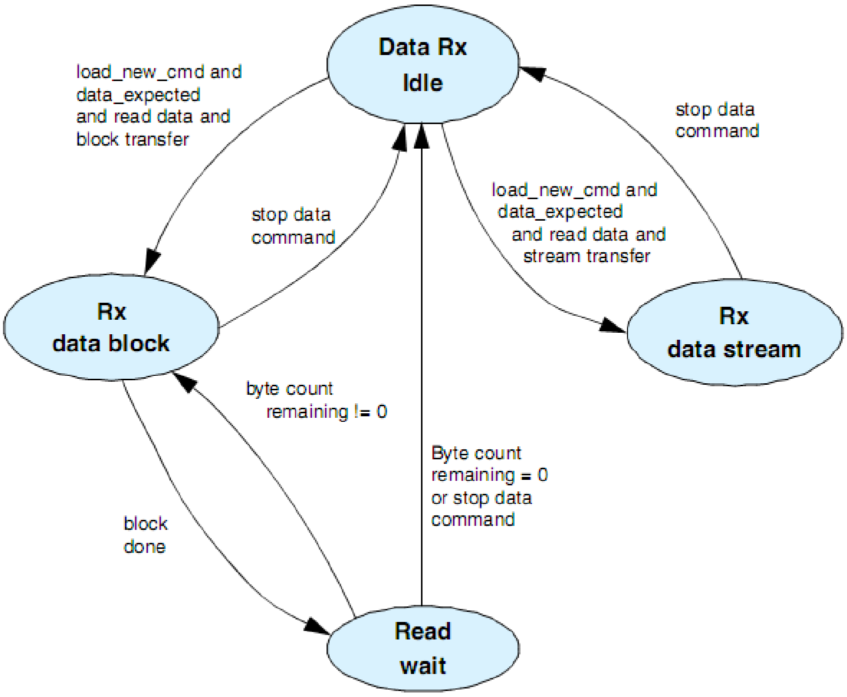

Data Rx
Idle
load_new_cmd and data_expected and read data and block transfer
stop data command
load_new_cmd and data_expected and read data and stream transfer
stop data command
Rx
data block
Rx
data stream
byte count remaining != 0
Byte count remaining = 0 or stop data command
block done
Read
wait

Fig. 3-5 Host Controller Data Receive State Machine

Stream Data Read

A stream-read data transfer occurs if the transfer_mode bit in the Command register equals 1, at which time the data path receives data from the card and pushes it to the FIFO. If the FIFO becomes full, the card clock stops and restarts once the FIFO is no longer full.

An open-ended stream-read data transfer occurs if the byte_count register equals 0. During this type of data transfer, the data path continuously receives data in a stream until the host software issues a stop command. A stream data transfer terminates two clock cycles after the end bit of the stop command.

If the byte_count register contains a non-zero value and the send_auto_stop bit is set in the Command register, a stop command is internally generated and loaded into the command path, where the end bit of the stop command occurs after the last byte of the stream data transfer is received. This data transfer can terminate if the host issues a stop or abort command before all the data bytes are received from the card.

Single-Block Data Read

A single-block read-data transfer occurs if the transfer_mode bit in the Command register is set to 0 and the value of the byte_count register is equal to the value of the block_size register. When a start bit is received before the data times out, data bytes equal to the block size and CRC16 are received and checked with the internally-generated CRC16.

If the SDMMC_CTYPE register bit for the selected card – indicated by the card_num value in the Command register – is set to a 1-bit, 4-bit, or 8-bit data transfer, data is received from 1, 4, or 8 data lines, respectively, and CRC16 is separately generated and checked for 1, 4, or 8 data lines, respectively. If there is a CRC16 mismatch, the data path signals a data CRC error to the BIU. If the received end bit is not 1, the BIU receives an end-bit error.

Multiple-Block Data Read

If the transfer_mode bit in the Command register is set to 0 and the value of the byte_count register is not equal to the value of the block_size register, it is a multiple-block read-data transfer. The data-receive state machine receives data in blocks, where the number of bytes in a block is equal to the block size, including the internally-generated CRC16.

If the SDMMC_CTYPE register bit for the selected card – indicated by the card_num value in the Command register – is set to a 1-bit, 4-bit, or 8-bit data transfer, data is received from 1, 4, or 8 data lines, respectively, and CRC16 is separately generated and checked for 1, 4, or 8 data lines, respectively.

After a data block is received, if the remaining byte_count becomes 0, the data path signals

a data transfer to the BIU.
If the remaining data bytes are greater than 0, the data path state machine causes another data block to be received. If CRC16 of a received data block does not match the internally-generated CRC16, a data CRC error to the BIU and data reception continue further data transmission until all bytes are transmitted.
Additionally, if the end of a received data block is not 1, data on the data path signals terminate the bit error to the CIU and the data-receive state machine terminates data reception, waits for data timeout, and signals to the BIU that the data transfer is complete.
If the send_auto_stop bit is set in the Command register, the stop command is internally generated when the last data block is transferred, where no extra bytes are transferred from the card; the end bit of the stop command may not exactly match the end bit of the last data block.
If the requested block size for data transfers to cards is less than 4, 16, or 32 bytes for 1-bit, 4-bit, or 8-bit data transfer modes, respectively, the data-transmit state machine terminates the data transfer when all data is transferred, at which point the internally-generated stop command is loaded in the command path. Data received from the card after that are then ignored by the data path.
If the byte_count is 0—the block size must be greater than 0—it is an open-ended block transfer. For this type of data transfer, the data-receive state machine continues the block-read data transfer until the host software issues a stop or abort command.

Auto-Stop

The Host Controller internally generates a stop command and is loaded in the command path when the send_auto_stop bit is set in the Command register.
The software should set the send_auto_stop bit according to details listed in following table.

Table 3-2 Auto-Stop Generation

Card type	Transfer type	Byte Count	send_auto_stop bit set	Comments
MMC	Stream read	0	No	Open-ended stream
MMC	Stream read	>0	Yes	Auto-stop after all bytes transfer
MMC	Stream write	0	No	Open-ended stream
MMC	Stream write	>0	Yes	Auto-stop after all bytes transfer
MMC	Single-block read	>0	No	Byte count =0 is illegal
MMC	Single-block write	>0	No	Byte count =0 is illegal
MMC	Multiple-block read	0	No	Open-ended multiple block
MMC	Multiple-block read	>0	Yes①	Pre-defined multiple block
MMC	Multiple-block write	0	No	Open-ended multiple block
MMC	Multiple-block write	>0	Yes①	Pre-defined multiple block
SDMEM	Single-block read	>0	No	Byte count =0 is illegal
SDMEM	Single-block write	>0	No	Byte count =0 illegal
SDMEM	Multiple-block read	0	No	Open-ended multiple block
SDMEM	Multiple-block read	>0	Yes	Auto-stop after all bytes transfer
SDMEM	Multiple-block write	0	No	Open-ended multiple block
SDMEM	Multiple-block write	>0	Yes	Auto-stop after all bytes transfer
SDIO	Single-block read	>0	No	Byte count =0 is illegal
SDIO	Single-block write	>0	No	Byte count =0 illegal
SDIO	Multiple-block read	0	No	Open-ended multiple block
SDIO	Multiple-block	>0	No	Pre-defined multiple block

Card type	Transfer type	Byte Count	send_auto_stop bit set	Comments
	read			
SDIO	Multiple-block write	0	No	Open-ended multiple block
SDIO	Multiple-block write	>0	No	Pre-defined multiple block

①:*The condition under which the transfer mode is set to block transfer and byte_count is equal to block size is treated as a single-block data transfer command for both MMC and SD cards. If byte_count = n*block_size (n = 2, 3, …), the condition is treated as a predefined multiple-block data transfer command. In the case of an MMC card, the host software can perform a predefined data transfer in two ways: 1) Issue the CMD23 command before issuing CMD18/CMD25 commands to the card – in this case, issue MD18/CMD25 commands without setting the send_auto_stop bit. 2) Issue CMD18/CMD25 commands without issuing CMD23 command to the card, with the send_auto_stop bit set. In this case, the multiple-block data transfer is terminated by an internally-generated auto-stop command after the programmed byte count.*

The following list conditions for the auto-stop command.

- Stream read for MMC card with byte count greater than 0 – The Host Controller generates an internal stop command and loads it into the command path so that the end bit of the stop command is sent out when the last byte of data is read from the card and no extra data byte is received. If the byte count is less than 6 (48 bits), a few extra data bytes are received from the card before the end bit of the stop command is sent.
- Stream write for MMC card with byte count greater than 0 - The Host Controller generates an internal stop command and loads it into the command path so that the end bit of the stop command is sent when the last byte of data is transmitted on the card bus and no extra data byte is transmitted. If the byte count is less than 6 (48 bits), the data path transmits the data last in order to meet the above condition.
- Multiple-block read memory for SD card with byte count greater than 0 – If the block size is less than 4 (single-bit data bus), 16 (4-bit data bus), or 32 (8-bit data bus), the auto-stop command is loaded in the command path after all the bytes are read. Otherwise, the top command is loaded in the command path so that the end bit of the stop command is sent after the last data block is received.
- Multiple-block write memory for SD card with byte count greater than 0 – If the block size is less than 3 (single-bit data bus), 12 (4-bit data bus), or 24 (8-bit data bus), the auto-stop command is loaded in the command path after all data blocks are transmitted. Otherwise, the stop command is loaded in the command path so that the end bit of the stop command is sent after the end bit of the CRC status is received.
- Precaution for host software during auto-stop – Whenever an auto-stop command is issued, the host software should not issue a new command to the SD/MMC device until the auto-stop is sent by the Host Controller and the data transfer is complete. If the host issues a new command during a data transfer with the auto-stop in progress, an auto-stop command may be sent after the new command is sent and its response is received; this can delay sending the stop command, which transfers extra data bytes. For a stream write, extra data bytes are erroneous data that can corrupt the card data. If the host wants to terminate the data transfer before the data transfer is complete, it can issue a stop or abort command, in which case the Host Controller does not generate an auto-stop command.

3. Non-Data Transfer Commands that Use Data Path

Some non-data transfer commands (non-read/write commands) also use the data path. Following table lists the commands and register programming requirements for them.

Table 3-3 Non-data Transfer Commands and Requirements

Base Address [12:8]	CMD 27	CMD 30	CMD 42	ACMD 13	ACMD 22	ACMD 51
Command register programming						
cmd_index	6'h1B	6'h1E	6'h2A	6'h0D	6'h16	6'h33
response_expect	1	1	1	1	1	1
rResponse_length	0	0	0	0	0	0

Base Address [12:8]	CMD 27	CMD 30	CMD 42	ACMD 13	ACMD 22	ACMD 51
check_response_crc	1	1	1	1	1	1
data_expected	1	1	1	1	1	1
read/write	1	0	1	0	0	0
transfer_mode	0	0	0	0	0	0
send_auto_stop	0	0	0	0	0	0
wait_prevdata_complete	0	0	0	0	0	0
stop_abort_cmd	0	0	0	0	0	0
Command Argument register programming						
	stuff bits	32-bit write protect data address	stuff bits	stuff bits	stuff bits	stuff bits
Block Size register programming						
	16	4	Num_bytes①	64	4	8
Byte Count register programming						
	16	4	Num_bytes①	64	4	8

①: *Num_bytes = No. of bytes specified as per the lock card data structure (Refer to the SD specification and the MMC specification)*

4. SDIO Interrupt Control

Interrupts for SD cards are reported to the BIU by asserting an interrupt signal for two clock cycles. SDIO cards signal an interrupt by asserting cdata_in low during the interrupt period; an interrupt period for the selected card is determined by the interrupt control state machine. An interrupt period is always valid for non-active or non-selected cards, and 1-bit data mode for the selected card. An interrupt period for a wide-bus active or selected card is valid for the following conditions:

- Card is idle
- Non-data transfer command in progress
- Third clock after end bit of data block between two data blocks
- From two clocks after end bit of last data until end bit of next data transfer command

Bear in mind that, in the following situations, the controller does not sample the SDIO interrupt of the selected card when the card data width is 4 bits. Since the SDIO interrupt is level-triggered, it is sampled in a further interrupt period and the host does not lose any SDIO interrupt from the card.

- Read/Write Resume – The CIU treats the resume command as a normal data transfer command. SDIO interrupts during the resume command are handled similarly to other data commands. According to the SDIO specification, for the normal data command the interrupt period ends after the command end bit of the data command; for the resume command, it ends after the response end bit. In the case of the resume command, the Controller stops the interrupt sampling period after the resume command end bit, instead of stopping after the response end bit of the resume command.
- Suspend during read transfer – If the read data transfer is suspended by the host, the host sets the abort_read_data bit in the controller to reset the data state machine. In the CIU, the SDIO interrupts are handled such that the interrupt sampling starts after the abort_read_data bit is set by the host. In this case the controller does not sample SDIO interrupts between the period from response of the suspend command to setting the abort_read_data bit, and starts sampling after setting the abort_read_data bit.

5. Clock Control

The clock control block provides different clock frequencies required for SD/MMC cards. The cclk_in signal is the source clock (cclk_in >= card max operating frequency) for clock divider of the clock control block. This source clock (cclk_in) is used to generate different card clock frequencies (cclk_out). The card clock can have different clock frequencies, since the card can be a low-speed card or a full-speed card. The Host Controller provides one clock signal

(cclk_out).
The clock frequency of a card depends on the following clock control registers:

- Clock Divider register – Internal clock dividers are used to generate different clock frequencies required for card. The division factor for each clock divider can be programmed by writing to the Clock Divider register. The clock divider is an 8-bit value that provides a clock division factor from 1 to 510; a value of 0 represents a clock-divider bypass, a value of 1 represents a divide by 2, a value of 2 represents a divide by 4, and so on.
- Clock Control register – cclk_out can be enabled or disabled for each card under the following conditions:
  - clk_enable – cclk_out for a card is enabled if the clk_enable bit for a card in the Clock Control register is programmed (set to 1) or disabled (set to 0).
  - Low-power mode – Low-power mode of a card can be enabled by setting the low-power mode bit of the Clock Control register to 1. If low-power mode is enabled to save card power, the cclk_out is disabled when the card is idle for at least 8 card clock cycles. It is enabled when a new command is loaded and the command path goes to a non-idle state.

Additionally, cclk_out is disabled when an internal FIFO is full – card read (no more data can be received from card) – or when the FIFO is empty – card write (no data is available for transmission). This helps to avoid FIFO overrun and underrun conditions. It is used by the command and data path to qualify cclk_in for driving outputs and sampling inputs at the programmed clock frequency for the selected card, according to the Clock Divider and Clock Source register values.
Under the following conditions, the card clock is stopped or disabled, along with the active clk_en, for the selected card:

- Clock can be disabled by writing to Clock Enable register (clk_en bit = 1).
- If low-power mode is selected and card is idle, or not selected for 8 clocks.
- FIFO is full and data path cannot accept more data from the card and data transfer is incomplete –to avoid FIFO overrun.
- FIFO is empty and data path cannot transmit more data to the card and data transfer is incomplete – to avoid FIFO underrun.

6. Error Detection

- Response
  - Response timeout – Response expected with response start bit is not received within programmed number of clocks in timeout register.
  - Response CRC error – Response is expected and check response CRC requested; response CRC7 does not match with the internally-generated CRC7.
  - Response error – Response transmission bit is not 0, command index does not match with the command index of the send command, or response end bit is not 1.
- Data transmit
  - No CRC status – During a write data transfer, if the CRC status start bit is not received two clocks after the end bit of the data block is sent out, the data path does the following:
    - Signals no CRC status error to the BIU
    - Terminates further data transfer
    - Signals data transfer done to the BIU
  - Negative CRC – If the CRC status received after the write data block is negative (that is, not 010), a data CRC error is signaled to the BIU and further data transfer is continued.
  - Data starvation due to empty FIFO – If the FIFO becomes empty during a write data transmission, or if the card clock is stopped and the FIFO remains empty for data timeout clocks, then a data-starvation error is signaled to the BIU and the data path continues to wait for data in the FIFO.
- Data receive
  - Data timeout – During a read-data transfer, if the data start bit is not received before the number of clocks that were programmed in the timeout register, the data

path does the following:

  - ◆ Signals data-timeout error to the BIU
  - ◆ Terminates further data transfer
  - ◆ Signals data transfer done to BIU

- ■ Data start bit error – During a 4-bit or 8-bit read-data transfer, if the all-bit data line does not have a start bit, the data path signals a data start bit error to the BIU and waits for a data timeout, after which it signals that the data transfer is done.
- ■ Data CRC error – During a read-data-block transfer, if the CRC16 received does not match with the internally generated CRC16, the data path signals a data CRC error to the BIU and continues further data transfer.
- ■ Data end-bit error – During a read-data transfer, if the end bit of the received data is not 1, the data path signals an end-bit error to the BIU, terminates further data transfer, and signals to the BIU that the data transfer is done.
- ■ Data starvation due to FIFO full – During a read data transmission and when the FIFO becomes full, the card clock is stopped. If the FIFO remains full for data timeout clocks, a data starvation error is signaled to the BIU (Data Starvation by Host Timeout bit is set in SDMMC_RINTSTS register) and the data path continues to wait for the FIFO to start to empty.

3.3.3 Internal Direct Memory Access Controller (IDMAC)

The Internal Direct Memory Access Controller (IDMAC) has a Control and Status Register (CSR) and a single Transmit/Receive engine, which transfers data from host memory to the device port and vice versa. The controller utilizes a descriptor to efficiently move data from source to destination with minimal Host CPU intervention. You can program the controller to interrupt the Host CPU in situations such as data Transmit and Receive transfer completion from the card, as well as other normal or error conditions.

The IDMAC and the Host driver communicate through a single data structure. CSR addresses 0x80 to 0x98 are reserved for host programming.

The IDMAC transfers the data received from the card to the Data Buffer in the Host memory, and it transfers Transmit data from the Data Buffer in the Host memory to the FIFO. Descriptors that reside in the Host memory act as pointers to these buffers.

A data buffer resides in physical memory space of the Host and consists of complete data or partial data. Buffers contain only data, while buffer status is maintained in the descriptor. Data chaining refers to data that spans multiple data buffers. However, a single descriptor cannot span multiple data.

A single descriptor is used for both reception and transmission. The base address of the list is written into Descriptor List Base Address Register (SDMMC_DBADDR @0x88). A descriptor list is forward linked. The Last Descriptor can point back to the first entry in order to create a ring structure. The descriptor list resides in the physical memory address space of the Host. Each descriptor can point to a maximum of two data buffers.

1. IDMAC CSR Access

When an IDMAC is introduced, an additional CSR space resides in the IDMAC that controls the IDMAC functionality. The host accesses the new CSR space in addition to the existing control register set in the BIU. The IDMAC CSR primarily contains descriptor information. For a write operation to the CSR, the respective CSR logic of the IDMAC and BIU decodes the address before accepting. For a read operation from the CSR, the appropriate CSR read path is enabled.

You can enable or disable the IDMAC operation by programming bit[25] in the SDMMC_CTRL register of the BIU. This allows the data transfer by accessing the slave interface on the AMBA bus if the IDMAC is present but disabled. When IDMAC is enabled, the FIFO cannot be accessed through the slave interface.

2. Descriptors

- ● Descriptor structures

The IDMAC uses these types of descriptor structures:

  - ■ Dual-Buffer Structure – The distance between two descriptors is determined by the

Skip Length value programmed in the Descriptor Skip Length (DSL) field of the Bus Mode Register (SDMMC_BMOD @0x80).

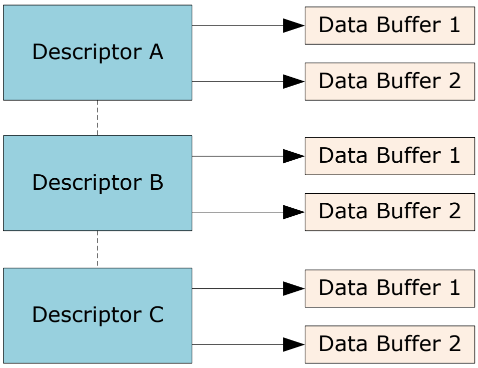


Fig. 3-6 Dual-Buffer Descriptor Structure

- Chain Structure – Each descriptor points to a unique buffer and the next descriptor.

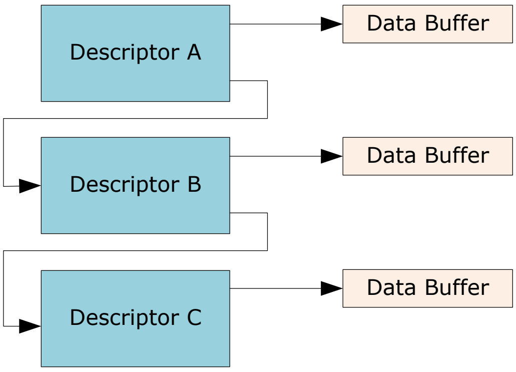


Fig. 3-7 Chain Descriptor Structure

- Descriptor formats

Following figure illustrates the internal formats of a descriptor. The descriptor addresses must be aligned to the bus width used for 32-bit AHB data buses. Each descriptor contains 16 bytes of control and status information. DES0 is a notation used to denote the [31:0] bits, DES1 to denote [63:32] bits, DES2 to denote [95:64] bits, DES3 to denote [127:96] bits.

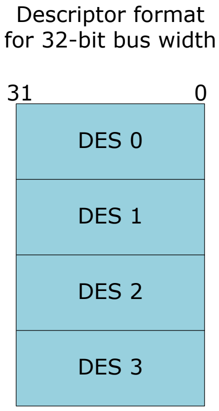


Fig. 3-8 Descriptor Formats for 32-bit AHB Address Bus Width

9

- The DES0 element in the IDMAC contains control and status information.

Table 3-4 Bits in IDMAC DES0 Element

Bit	Name	Description
31	OWN	When set, this bit indicates that the descriptor is owned by the IDMAC. When this bit is reset, it indicates that the descriptor is owned by the Host. The IDMAC clears this bit when it completes the data transfer.
30	Card Error	These error bits indicate the status of the transaction to or

Bit	Name	Description
	Summary (CES)	from the card. These bits are also present in SDMMC_RINTSTS Indicates the logical OR of the following bits: ● EBE: End Bit Error ● RTO: Response Time out ● RCRC: Response CRC ● SBE: Start Bit Error ● DRTO: Data Read Timeout ● DCRC: Data CRC for Receive ● RE: Response Error
29:6	Reserved	-
5	End of Ring (ER)	When set, this bit indicates that the descriptor list reached its final descriptor. The IDMAC returns to the base address of the list, creating a Descriptor Ring. This is meaningful for only a dual-buffer descriptor structure.
4	Second Address Chained (CH)	When set, this bit indicates that the second address in the descriptor is the Next Descriptor address rather than the second buffer address. When this bit is set, BS2 (DES1[25:13]) should be all zeros.
3	First Descriptor (FS)	When set, this bit indicates that this descriptor contains the first buffer of the data. If the size of the first buffer is 0, next Descriptor contains the beginning of the data.
2	Last Descriptor (LD)	This bit is associated with the last block of a DMA transfer. When set, the bit indicates that the buffers pointed to by this descriptor are the last buffers of the data. After this descriptor is completed, the remaining byte count is 0. In other words, after the descriptor with the LD bit set is completed, the remaining byte count should be 0.
1	Disable Interrupt on Completion (DIC)	When set, this bit will prevent the setting of the TI/RI bit of the IDMAC Status Register (IDSTS) for the data that ends in the buffer pointed to by this descriptor.
0	Reserved	-

10

- The DES1 element contains the buffer size.

Table 3-5 Bits in IDMAC DES1 Element

Bit	Name	Description
31:26	Reserved	-
25:13	Buffer 2 Size (BS2)	These bits indicate the second data buffer byte size. The buffer size must be a multiple of 2, 4, or 8, depending upon the bus widths—16, 32, and 64 respectively. In the case where the buffer size is not a multiple of 2, 4, or 8, the resulting behavior is undefined. If this field is 0, the DMA ignores this buffer and proceeds to the next buffer in case of a dual-buffer structure. This field is not valid for chain structure; that is, if DES0[4] is set.
12:0	Buffer 1 Size (BS1)	Indicates the data buffer byte size, which must be a multiple of 2, 4, or 8 bytes, depending upon the bus widths—16, 32, and 64, respectively. In the case where the buffer size is not a multiple of 2, 4, or 8, the resulting behavior is undefined. This field should not be zero. Note: If there is only one buffer to be programmed, you need to use only the Buffer 1, and not Buffer 2.

11

- The DES2 element contains the address pointer to the data buffer.

Table 3-6 Bits in IDMAC DES2 Element

Bit	Name	Description
31:26	Reserved	
25:13	Buffer 2 Size (BS2)	These bits indicate the second data buffer byte size. The buffer size must be a multiple of 2, 4, or 8, depending upon the bus widths—16, 32, and 64 respectively. In the case where the buffer size is not a multiple of 2, 4, or 8, the resulting behavior is undefined. If this field is 0, the DMA ignores this buffer and proceeds to the next buffer in case of a dual-buffer structure. This field is not valid for chain structure; that is, if DES0[4] is set.
12:0	Buffer 1 Size (BS1)	Indicates the data buffer byte size, which must be a multiple of 2, 4, or 8 bytes, depending upon the bus widths—16, 32, and 64, respectively. In the case where the buffer size is not a multiple of 2, 4, or 8, the resulting behavior is undefined. This field should not be zero. Note: If there is only one buffer to be programmed, you need to use only the Buffer 1, and not Buffer 2.

12

- The DES3 element contains the address pointer to the next descriptor if the present descriptor is not the last descriptor in a chained descriptor structure or the second buffer address for a dual-buffer structure.

Table 3-7 Bits in IDMAC DES3 Element

Bit	Name	Description
31:0	Buffer Address Pointer 2/ Next Descriptor Address (BAP2)	These bits indicate the physical address of the second buffer when the dual-buffer structure is used. If the Second Address Chained (DES0[4]) bit is set, then this address contains the pointer to the physical memory where the Next Descriptor is present. If this is not the last descriptor, then the Next Descriptor address pointer must be bus-width aligned.

3. Initialization

IDMAC initialization occurs as follows:

1) Write to IDMAC Bus Mode Register—SDMMC_BMOD to set Host bus access parameters.
2) Write to IDMAC Interrupt Enable Register—SDMMC_IDINTEN to mask unnecessary interrupt causes.
3) The software driver creates either the Transmit or the Receive descriptor list. Then it writes to IDMAC Descriptor List Base Address Register (SDMMC_DBADDR), providing the IDMAC with the starting address of the list.
4) The IDMAC engine attempts to acquire descriptors from the descriptor lists.

- Host Bus Burst Access

The IDMAC attempts to execute fixed-length burst transfers on the AHB Master interface if configured using the FB bit of the IDMAC Bus Mode register. The maximum burst length is indicated and limited by the PBL field. The descriptors are always accessed in the maximum possible burst-size for the 16-bytes to be read— 16*8/bus-width.

The IDMAC initiates a data transfer only when sufficient space to accommodate the configured burst is available in the FIFO or the number of bytes to the end of data, when less than the configured burst-length.

The IDMAC indicates the start address and the number of transfers required to the AHB Master Interface. When the AHB Interface is configured for fixed-length bursts, then it transfers data using the best combination of INCR4/8/16 and SINGLE transactions. Otherwise, in no fixed-length bursts, it transfers data using INCR (undefined length) and SINGLE transactions.

- Host Data Buffer Alignment

The Transmit and Receive data buffers in host memory must be aligned, depending on the data width.

- Buffer Size Calculations

The driver knows the amount of data to transmit or receive. For transmitting to the card, the IDMAC transfers the exact number of bytes to the FIFO, indicated by the buffer size field of DES1.

If a descriptor is not marked as last-LS bit of DES0-then the corresponding buffer(s) of the descriptor are full, and the amount of valid data in a buffer is accurately indicated by its buffer size field. If a descriptor is marked as last, then the buffer cannot be full, as indicated by the buffer size in DES1. The driver is aware of the number of locations that are valid in this case.

- Transmission

IDMAC transmission occurs as follows:

1) The Host sets up the elements (DES0-DES3) for transmission and sets the OWN bit (DES0[31]). The Host also prepares the data buffer.
2) The Host programs the write data command in the SDMMC_CMD register in BIU.
3) The Host will also program the required transmit threshold level (TX_WMark field in SDMMC_FIFOTH register).
4) The IDMAC determines that a write data transfer needs to be done as a consequence of step 2.
5) The IDMAC engine fetches the descriptor and checks the OWN bit. If the OWN bit is not set, it means that the host owns the descriptor. In this case the IDMAC enters suspend state and asserts the Descriptor Unable interrupt in the SDMMC_IDSTS register. In such a case, the host needs to release the IDMAC by writing any value to the poll demand register.
6) It will then wait for Command Done (CD) bit and no errors from BIU which indicates that a transfer can be done.
7) The IDMAC engine will now wait for a DMA interface request from BIU. This request will be generated based on the programmed transmit threshold value. For the last bytes of data which can't be accessed using a burst, SINGLE transfers are performed on AHB Master Interface.
8) The IDMAC fetches the Transmit data from the data buffer in the Host memory and transfers to the FIFO for transmission to card.
9) When data spans across multiple descriptors, the IDMAC will fetch the next descriptor and continue with its operation with the next descriptor. The Last Descriptor bit in the descriptor indicates whether the data spans multiple descriptors or not.
10) When data transmission is complete, status information is updated in SDMMC_IDSTS register by setting Transmit Interrupt, if enabled. Also, the OWN bit is cleared by the IDMAC by performing a write transaction to DES0.

- Reception

IDMAC reception occurs as follows:

1) The Host sets up the element (DES0-DES3) for reception, sets the OWN (DES0[31]).
2) The Host programs the read data command in the SDMMC_CMD register in BIU.
3) The Host will program the required receive threshold level (RX_WMark field in FIFOTH register).
4) The IDMAC determines that a read data transfer needs to be done as a consequence of step 2.
5) The IDMAC engine fetches the descriptor and checks the OWN bit. If the OWN bit is not set, it means that the host owns the descriptor. In this case the DMA enters suspend state and asserts the Descriptor Unable interrupt in the SDMMC_IDSTS register. In such a case, the host needs to release the IDMAC by writing any value to the poll demand register.
6) It will then wait for Command Done (CD) bit and no errors from BIU which indicates that a transfer can be done.
7) The IDMAC engine will now wait for a DMA interface request from BIU. This request will be generated based on the programmed receive threshold value. For the last bytes of data which can't be accessed using a burst, SINGLE transfers are performed on AHB.

8) The IDMAC fetches the data from the FIFO and transfer to Host memory.
9) When data spans across multiple descriptors, the IDMAC will fetch the next descriptor and continue with its operation with the next descriptor. The Last Descriptor bit in the descriptor indicates whether the data spans multiple descriptors or not.
10) When data reception is complete, status information is updated in SDMMC_IDSTS register by setting Receive Interrupt, if enabled. Also, the OWN bit is cleared by the IDMAC by performing a write transaction to DES0.

- Interrupts

Interrupts can be generated as a result of various events. SDMMC_IDSTS register contains all the bits that might cause an interrupt. SDMMC_IDINTEN register contains an Enable bit for each of the events that can cause an interrupt.

There are two groups of summary interrupts-Normal and Abnormal-as outlined in SDMMC_IDSTS register. Interrupts are cleared by writing a 1 to the corresponding bit position. When all the enabled interrupts within a group are cleared, the corresponding summary bit is cleared. When both the summary bits are cleared, the interrupt signal dmac_intr_o is de-asserted.

Interrupts are not queued and if the interrupt event occurs before the driver has responded to it, no additional interrupts are generated. For example, Receive Interrupt—SDMMC_IDSTS[1] indicates that one or more data was transferred to the Host buffer.

An interrupt is generated only once for simultaneous, multiple events. The driver must scan SDMMC_IDSTS register for the interrupt cause.

3.4 Register Description

3.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
SDMMC_CTRL	0x0000	W	0x00000000	Control register
SDMMC_PWREN	0x0004	W	0x00000000	Power-enable register
SDMMC_CLKDIV	0x0008	W	0x00000000	Clock-divider register
SDMMC_CLKSRC	0x000c	W	0x00000000	SD Clock Source Register
SDMMC_CLKENA	0x0010	W	0x00000000	Clock-enable register
SDMMC_TMOUT	0x0014	W	0xffffff40	Time-out register
SDMMC_CTYPE	0x0018	W	0x00000000	Card-type register
SDMMC_BLKSIZ	0x001c	W	0x00000200	Block-size register
SDMMC_BYTCNT	0x0020	W	0x00000200	Byte-count register
SDMMC_INTMASK	0x0024	W	0x00000000	Interrupt-mask register
SDMMC_CMDARG	0x0028	W	0x00000000	Command-argument register
SDMMC_CMD	0x002c	W	0x00000000	Command register
SDMMC_RESP0	0x0030	W	0x00000000	Response-0 register
SDMMC_RESP1	0x0034	W	0x00000000	Response-1 register
SDMMC_RESP2	0x0038	W	0x00000000	Response-2 register
SDMMC_RESP3	0x003c	W	0x00000000	Response-3 register
SDMMC_MINTSTS	0x0040	W	0x00000000	Masked interrupt-status register
SDMMC_RINTSTS	0x0044	W	0x00000000	Raw interrupt-status register
SDMMC_STATUS	0x0048	W	0x00000406	Status register
SDMMC_FIFOTH	0x004c	W	0x00000000	FIFO threshold register
SDMMC_CDETECT	0x0050	W	0x00000000	Card-detect register
SDMMC_WRTPRT	0x0054	W	0x00000000	Write-protect register
SDMMC_TCBCNT	0x005c	W	0x00000000	Transferred CIU card byte count

Name	Offset	Size	Reset Value	Description
SDMMC_TBBCNT	0x0060	W	0x00000000	Transferred host/DMA to/from BIU-FIFO byte count
SDMMC_DEBNCE	0x0064	W	0x00ffffff	Card detect debounce register
SDMMC_USRID	0x0068	W	0x07967797	User ID register
SDMMC_VERID	0x006c	W	0x5342270a	Synopsys version ID register
SDMMC_HCON	0x0070	W	0x00000000	Hardware configuration register
SDMMC_UHS_REG	0x0074	W	0x00000000	UHS-1 register
SDMMC_RSTN	0x0078	W	0x00000001	Hardware reset register
SDMMC_BMOD	0x0080	W	0x00000000	Bus mode register
SDMMC_PLDMND	0x0084	W	0x00000000	Poll demand register
SDMMC_DBADDR	0x0088	W	0x00000000	Descriptor list base address register
SDMMC_IDSTS	0x008c	W	0x00000000	Internal DMAC status register
SDMMC_IDINTEN	0x0090	W	0x00000000	Internal DMAC interrupt enable register
SDMMC_DSCADDR	0x0094	W	0x00000000	Current host descriptor address register
SDMMC_BUFADDR	0x0098	W	0x00000000	Current buffer descriptor address register
SDMMC_CARDTHRCTL	0x0100	W	0x00000000	Card read threshold enable register
SDMMC_BACK_END_POWER	0x0104	W	0x00000000	Back-end power register
SDMMC_EMMC_DDR_REG	0x010c	W	0x00000000	eMMC 4.5 ddr start bit detection control register
SDMMC_FIFO_BASE	0x0200	W	0x00000000	FIFO base address register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

3.4.2 Detail Register Description

SDMMC_CTRL

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:26	RO	0x0	reserved
25	RW	0x0	use_internal_dmac Present only for the Internal DMAC configuration; else, it is reserved. 1'b0: The host performs data transfers through the slave interface 1'b1: Internal DMAC used for data transfe
24:12	RO	0x0	reserved

Bit	Attr	Reset Value	Description
11	RW	0x0	ceata_device_interrupt_status 1'b0: Interrupts not enabled in CE-ATA device 1'b1: Interrupts are enabled in CE-ATA device Software should appropriately write to this bit after power-on reset or any other reset to CE-ATA device. After reset, usually CE-ATA device interrupt is disabled. If the host enables CE-ATA device interrupt, then software should set this bit
10	RW	0x0	send_auto_stop_ccsd 1'b0: Clear bit if the Host Controller does not reset the bit. 1'b1: Send internally generated STOP after sending CCSD to CE-ATA device. NOTE: Always set send_auto_stop_ccsd and send_ccsd bits together send_auto_stop_ccsd should not be set independent of send_ccsd. When set, the Host Controller automatically sends internally-generated STOP command (CMD12) to CE-ATA device. After sending internally-generated STOP command, Auto Command Done (ACD) bit in SDMMC_RINTSTS is set and generates interrupt to host if Auto Command Done interrupt is not masked. After sending the CCSD, the Host Controller automatically clears send_auto_stop_ccsd bit
9	RW	0x0	send_ccsd 1'b0: Clear bit if the Host Controller does not reset the bit 1'b1: Send Command Completion Signal Disable (CCSD) to CE-ATA device When set, the Host Controller sends CCSD to CE-ATA device. Software sets this bit only if current command is expecting CCS (that is, RW_BLK) and interrupts are enabled in CE-ATA device.Once the CCSD pattern is sent to device, the Host Controller automatically clears send_ccsd bit. It also sets Command Done (CD) bit in SDMMC_RINTSTS register and generates interrupt to host if Command Done interrupt is not masked. NOTE: Once send_ccsd bit is set, it takes two card clock cycles to drive the CCSD on the CMD line. Due to this, during the boundary conditions it may happen that CCSD is sent to the CE-ATA device, even if the device signalled CCS
8	RW	0x0	abort_read_data 1'b0: no change 1'b1: after suspend command is issued during read-transfer, software polls card to find when suspend happened. Once suspend occurs, software sets bit to reset data state-machine, which is waiting for next block of data. Bit automatically clears once data state machine resets to idle. Used in SDIO card suspend sequence

Bit	Attr	Reset Value	Description
7	RW	0x0	send_irq_response 1'b0: no change 1'b1: send auto IRQ response Bit automatically clears once response is sent. To wait for MMC card interrupts, host issues CMD40, and SDMMC Controller waits for interrupt response from MMC card(s). In meantime, if host wants SDMMC Controller to exit waiting for interrupt state, it can set this bit, at which time SDMMC Controller command state-machine sends CMD40 response on bus and returns to idle state
6	RW	0x0	read_wait 1'b0: clear read wait 1'b1: assert read wait For sending read-wait to SDIO cards
5	RW	0x0	dma_enable 1'b0: disable DMA transfer mode 1'b1: enable DMA transfer mode Even when DMA mode is enabled, host can still push/pop data into or from FIFO; this should not happen during the normal operation. If there is simultaneous FIFO access from host/DMA, the data coherency is lost. Also, there is no arbitration inside SDMMC Controller to prioritize simultaneous host/DMA access
4	RW	0x0	int_enable Global interrupt enable/disable bit: 1'b0: disable interrupts 1'b1: enable interrupts The int port is 1 only when this bit is 1 and one or more unmasked interrupts are set
3	RO	0x0	reserved
2	RW	0x0	dma_reset 1'b0: no change 1'b1: reset internal DMA interface control logic To reset DMA interface, firmware should set bit to 1. This bit is auto-cleared after two AHB clocks
1	RW	0x0	fifo_reset 1'b0: no change 1'b1: reset to data FIFO To reset FIFO pointers To reset FIFO, firmware should set bit to 1. This bit is auto-cleared after completion of reset operation

Bit	Attr	Reset Value	Description
0	RW	0x0	controller_reset 1'b0: no change 1'b1: reset SDMMC controller To reset controller, firmware should set bit to 1. This bit is auto-cleared after two AHB and two cclk_in clock cycles. This resets: a. BIU/CIU interface b. CIU and state machines c. abort_read_data, send_irq_response, and read_wait bits of Control register d. start_cmd bit of Command register Does not affect any registers or DMA interface, or FIFO or host interrupts

SDMMC_PWREN

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	power_enable Power on/off switch for the card. Once power is turned on, firmware should wait for regulator/switch ramp-up time before trying to initialize card. 1'b0: power off 1'b1: power on Bit values output to card_power_en port

SDMMC_CLKDIV

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	clk_divider0 Clock divider-0 value. Clock division is 2*n. For example, value of 0 means divide by 2*0 = 0 (no division, bypass), value of 1 means divide by 2*1 = 2, and so on

SDMMC_CLKSRC

Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved

Bit	Attr	Reset Value	Description
1:0	RW	0x0	clk_source Clock divider source for up to 16 SD cards supported. Each card has two bits assigned to it. For example, bits[1:0] assigned for card-0, which maps and internally routes clock divider[3:0] outputs to cclk_out[15:0] pins, depending on bit value. 2'b00: clock divider 0

SDMMC_CLKENA
Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16	RW	0x0	cclk_low_power Low-power control for SD card clock and MMC card clock supported. 1'b0: non-low-power mode 1'b1: low-power mode; stop clock when card in IDLE (should be normally set to only MMC and SD memory cards; for SDIO cards, if interrupts must be detected, clock should not be stopped)
15:1	RO	0x0	reserved
0	RW	0x0	cclk_enable Clock-enable control for SD card clock and MMC card clock supported. 1'b0: clock disabled 1'b1: clock enabled

SDMMC_TMOUT
Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:8	RW	0x000000	data_timeout Value for card Data Read Timeout; same value also used for Data Starvation by Host timeout. Value is in number of card output clocks cclk_out of selected card. Note: The software timer should be used if the timeout value is in the order of 100 ms. In this case, read data timeout interrupt needs to be disabled
7:0	RW	0x40	response_timeout Response timeout value. Value is in number of card output clock

SDMMC_CTYPE
Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved

Bit	Attr	Reset Value	Description
16	RW	0x0	card_width_8 Indicates if card is 8-bit. 1'b0: non 8-bit mode 1'b1: 8-bit mode
15:1	RO	0x0	reserved
0	RW	0x0	card_width Indicates if card is 1-bit or 4-bit. 1'b0: 1-bit mode 1'b1: 4-bit mode

SDMMC_BLKSIZ

Address: Operational Base + offset (0x001c)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	block_size Block size

SDMMC_BYTCNT

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	byte_count Number of bytes to be transferred; should be integer multiple of Block Size for block transfers. For undefined number of byte transfers, byte count should be set to 0. When byte count is set to 0, it is responsibility of host to explicitly send stop/abort command to terminate data transfer

SDMMC_INTMASK

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24	RW	0x0	sdio_int_mask Mask SDIO interrupts. When masked, SDIO interrupt detection for that card is disabled. A 0 masks an interrupt, and 1 enables an interrupt
23:17	RO	0x0	reserved
16	RW	0x0	data_nobusy_int_mask 1'b0: data no busy interrupt not masked 1'b1: data no busy interrupt masked

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	int_mask Bits used to mask unwanted interrupts. Value of 0 masks interrupt; value of 1 enables interrupt. bit 15: End-bit error (read)/Write no CRC (EBE) bit 14: Auto command done (ACD) bit 13: Start-bit error (SBE) bit 12: Hardware locked write error (HLE) bit 11: FIFO underrun/overrun error (FRUN) bit 10: Data starvation-by-host timeout (HTO) /Volt_switch_int bit 9: Data read timeout (DRTO) bit 8: Response timeout (RTO) bit 7: Data CRC error (DCRC) bit 6: Response CRC error (RCRC) bit 5: Receive FIFO data request (RXDR) bit 4: Transmit FIFO data request (TXDR) bit 3: Data transfer over (DTO) bit 2: Command done (CD) bit 1: Response error (RE) bit 0: Card detect (CD)

SDMMC_CMDARG

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	cmd_arg Value indicates command argument to be passed to card

SDMMC_CMD

Address: Operational Base + offset (0x002c)

Bit	Attr	Reset Value	Description
31	RW	0x0	start_cmd Start command. Once command is taken by CIU, bit is cleared. When bit is set, host should not attempt to write to any command registers. If write is attempted, hardware lock error is set in raw interrupt register. Once command is sent and response is received from SD_MMC cards, Command Done bit is set in raw interrupt register
30	RO	0x0	reserved

Bit	Attr	Reset Value	Description
29	RW	0x0	use_hold_reg Use Hold Register. 1'b0: CMD and DATA sent to card bypassing HOLD Register 1'b1: CMD and DATA sent to card through the HOLD Register Note: a. Set to 1'b1 for SDR12 and SDR25 (with non-zero phase-shifted cclk_in_drv); zero phase shift is not allowed in these modes. b. Set to 1'b0 for SDR50, SDR104, and DDR50 (with zero phase-shifted cclk_in_drv) c. Set to 1'b1 for SDR50, SDR104, and DDR50 (with non-zero phase-shifted cclk_in_drv)
28	RW	0x0	volt_switch Voltage switch bit. 1'b0: no voltage switching 1'b1: voltage switching enabled; must be set for CMD11 only
27	RW	0x0	boot_mode Boot Mode. 1'b0: mandatory boot operation 1'b1: alternate boot operation
26	RW	0x0	disable_boot Disable boot. When software sets this bit along with start_cmd, CIU terminates the boot operation. Do NOT set disable_boot and enable_boot together
25	RW	0x0	expect_boot_ack Expect Boot Acknowledge. When Software sets this bit along with enable_boot, CIU expects a boot acknowledge start pattern of 0-1-0 from the selected card
24	RW	0x0	enable_boot Enable Boot. This bit should be set only for mandatory boot mode.When Software sets this bit along with start_cmd, CIU starts the boot sequence for the corresponding card by asserting the CMD line low. Do NOT set disable_boot and enable_boot together
23	RW	0x0	ccs_expected 1'b0: command does not expect CCS from device 1'b1: command expects CCS from device If the command expects Command Completion Signal (CCS) from the CE-ATA device, the software should set this control bit. The Host Controller sets Data Transfer Over (DTO) bit in SDMMC_RINTSTS register and generates interrupt to host if Data Transfer Over interrupt is not masked

Bit	Attr	Reset Value	Description
22	RW	0x0	read_ceata_device 1'b0: Host is not performing read access (RW_REG or RW_BLK) towards CE-ATA device 1'b1: Host is performing read access (RW_REG or RW_BLK) towards CE-ATA device Software should set this bit to indicate that CE-ATA device is being accessed for read transfer. This bit is used to disable read data timeout indication while performing CE-ATA read transfers. Maximum value of I/O transmission delay can be no less than 10 seconds. The Host Controller should not indicate read data timeout while waiting for data from CE-ATA device
21	RW	0x0	update_clock_registers_only 1'b0: normal command sequence 1'b1: do not send commands, just update clock register value into card clock domain Following register values transferred into card clock domain: CLKDIV, CLRSRC, CLKENA. Changes card clocks (change frequency, truncate off or on, and set low-frequency mode); provided in order to change clock frequency or stop clock without having to send command to cards. During normal command sequence, when update_clock_registers_only=0, following control registers are transferred from BIU to CIU: CMD, CMDARG, TMOUT, CTYPE, BLKSIZ, BYTCNT. CIU uses new register values for new command sequence to card. When bit is set, there are no Command Done interrupts because no command is sent to SD_MMC_CEATA cards
20:16	RO	0x0	reserved
15	RW	0x0	send_initialization 1'b0: do not send initialization sequence (80 clocks of 1) before sending this command 1'b1: send initialization sequence before sending this command After power on, 80 clocks must be sent to card for initialization before sending any commands to card. Bit should be set while sending first command to card so that controller will initialize clocks before sending command to card. This bit should not be set for either of the boot modes (alternate or mandatory)

Bit	Attr	Reset Value	Description
14	RW	0x0	stop_abort_cmd 1'b0: neither stop nor abort command to stop current data transfer in progress. If abort is sent to function-number currently selected or not in data-transfer mode, then bit should be set to 0. 1'b1: stop or abort command intended to stop current data transfer in progress. When open-ended or predefined data transfer is in progress, and host issues stop or abort command to stop data transfer, bit should be set so that command/data state-machines of CIU can return correctly to idle state. This is also applicable for Boot mode transfers. To Abort boot mode, this bit should be set along with CMD[26]=disable_boot
13	RW	0x0	wait_prvdata_complete 1'b0: send command at once, even if previous data transfer has not completed 1'b1: wait for previous data transfer completion before sending command The wait_prvdata_complete=0 option typically used to query status of card during data transfer or to stop current data transfer; card_number should be same as in previous command
12	RW	0x0	send_auto_stop 1'b0: no stop command sent at end of data transfer 1'b1: send stop command at end of data transfer When set, SDMMC Controller sends stop command to SD_MMC cards at end of data transfer. a. when send_auto_stop bit should be set, since some data transfers do not need explicit stop commands b. open-ended transfers that software should explicitly send to stop command Additionally, when "resume" is sent to resume-suspended memory access of SD-Combo card -bit should be set correctly if suspended data transfer needs send_auto_stop. Don't care if no data expected from card
11	RW	0x0	transfer_mode 1'b0: block data transfer command 1'b1: stream data transfer command Don't care if no data expected
10	RW	0x0	wr 1'b0: read from card 1'b1: write to card Don't care if no data expected from card
9	RW	0x0	data_expected 1'b0: no data transfer expected (read/write) 1'b1: data transfer expected (read/write)

Bit	Attr	Reset Value	Description
8	RW	0x0	check_response_crc 1'b0: do not check response CRC 1'b1: check response CRC Some of command responses do not return valid CRC bits. Software should disable CRC checks for those commands in order to disable CRC checking by controller
7	RW	0x0	response_length 1'b0: short response expected from card 1'b1: long response expected from card
6	RW	0x0	response_expect 1'b0: no response expected from card 1'b1: response expected from card
5:0	RW	0x00	cmd_index Command index

SDMMC_RESP0

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	response0 Bit[31:0] of response

SDMMC_RESP1

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	response1 Register represents bit[63:32] of long response. When CIU sends auto-stop command, then response is saved in register. Response for previous command sent by host is still preserved in Response 0 register. Additional auto-stop issued only for data transfer commands, and response type is always "short" for them

SDMMC_RESP2

Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	response2 Bit[95:64] of long response

SDMMC_RESP3

Address: Operational Base + offset (0x003c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	response3 Bit[127:96] of long response

SDMMC_MINTSTS

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24	RW	0x0	sdio_interrupt Interrupt from SDIO card; SDIO interrupt for card enabled only if corresponding sdio_int_mask bit is set in Interrupt mask register (mask bit 1 enables interrupt; 0 masks interrupt). 1'b0: no SDIO interrupt from card 1'b1: SDIO interrupt from card
23:17	RO	0x0	reserved
16	RW	0x0	data_nobusy_int_status Data no busy interrupt status, high active
15:0	RW	0x0000	int_status Interrupt enabled only if corresponding bit in interrupt mask register is set. bit 15: End-bit error (read)/Write no CRC (EBE) bit 14: Auto command done (ACD) bit 13: Start-bit error (SBE) bit 12: Hardware locked write error (HLE) bit 11: FIFO underrun/overrun error (FRUN) bit 10: Data starvation-by-host timeout (HTO) /Volt_switch_int bit 9: Data read timeout (DRTO) bit 8: Response timeout (RTO) bit 7: Data CRC error (DCRC) bit 6: Response CRC error (RCRC) bit 5: Receive FIFO data request (RXDR) bit 4: Transmit FIFO data request (TXDR) bit 3: Data transfer over (DTO) bit 2: Command done (CD) bit 1: Response error (RE) bit 0: Card detect (CD)

SDMMC_RINTSTS

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24	RW	0x0	sdio_interrupt Interrupt from SDIO card; Writes to these bits clear them. Value of 1 clears bit and 0 leaves bit intact. 1'b0: no SDIO interrupt from card 1'b1: SDIO interrupt from card
23:17	RO	0x0	reserved
16	RW	0x0	data_nobusy_int_status Data no busy interrupt status, high active

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	int_status Writes to bits clear status bit. Value of 1 clears status bit, and value of 0 leaves bit intact. Bits are logged regardless of interrupt mask status. bit 15: End-bit error (read)/Write no CRC (EBE) bit 14: Auto command done (ACD) bit 13: Start-bit error (SBE) bit 12: Hardware locked write error (HLE) bit 11: FIFO underrun/overrun error (FRUN) bit 10: Data starvation-by-host timeout (HTO) /Volt_switch_int bit 9: Data read timeout (DRTO) bit 8: Response timeout (RTO) bit 7: Data CRC error (DCRC) bit 6: Response CRC error (RCRC) bit 5: Receive FIFO data request (RXDR) bit 4: Transmit FIFO data request (TXDR) bit 3: Data transfer over (DTO) bit 2: Command done (CD) bit 1: Response error (RE) bit 0: Card detect (CD)

SDMMC_STATUS

Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31	RW	0x0	dma_req DMA request signal state
30	RO	0x0	dma_ack DMA acknowledge signal state
29:17	RO	0x0000	fifo_count Number of filled locations in FIFO
16:11	RO	0x00	response_index Index of previous response, including any auto-stop sent by core
10	RO	0x1	data_state_mc_busy Data transmit or receive state-machine is busy
9	RW	0x0	data_busy Inverted version of raw selected card_data[0]. 1'b0: card data not busy 1'b1: card data busy default value is 1 or 0 depending on cdata_in
8	RW	0x0	data_3_status Raw selected card_data[3]; checks whether card is present. 1'b0: card not present 1'b1: card present default value is 1 or 0 depending on cdata_in

Bit	Attr	Reset Value	Description
7:4	RW	0x0	command_fsm_states Command FSM states: 4'h0: idle 4'h1: send init sequence 4'h2: Tx cmd start bit 4'h3: Tx cmd tx bit 4'h4: Tx cmd index + arg 4'h5: Tx cmd crc7 4'h6: Tx cmd end bit 4'h7: Rx resp start bit 4'h8: Rx resp IRQ response 4'h9: Rx resp tx bit 4'ha: Rx resp cmd idx 4'hb: Rx resp data 4'hc: Rx resp crc7 4'hd: Rx resp end bit 4'he: Cmd path wait NCC 4'hf: Wait; CMD-to-response turnaround The command FSM state is represented using 19 bits. The SDMMC_STATUS Register[7:4] has 4 bits to represent the command FSM states. Using these 4 bits, only 16 states can be represented. Thus three states cannot be represented in the SDMMC_STATUS[7:4] register. The three states that are not represented in the SDMMC_STATUS Register[7:4] are: Bit 16: Wait for CCS Bit 17: Send CCSD Bit 18: Boot Mode Due to this, while command FSM is in "Wait for CCS state" or "Send CCSD" or "Boot Mode", the Status register indicates status as 0 for the bit field [7:4]
3	RO	0x0	fifo_full FIFO is full status
2	RO	0x1	fifo_empty FIFO is empty status
1	RO	0x1	fifo_tx_watermark FIFO reached Transmit watermark level; not qualified with data transfer
0	RO	0x0	fifo_rx_watermark FIFO reached Receive watermark level; not qualified with data transfer

SDMMC_FIFOTH

Address: Operational Base + offset (0x004c)

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved

Bit	Attr	Reset Value	Description
30:28	RW	0x0	dma_mutiple_transaction_size Burst size of multiple transaction; should be programmed same as DMA controller multiple-transaction-size SRC/DEST_MSIZE. 3'b000: 1 transfers 3'b001: 4 3'b010: 8 3'b011: 16 3'b100: 32 3'b101: 64 3'b110: 128 3'b111: 256 The unit for transfer is the H_DATA_WIDTH parameter. A single transfer (dw_dma_single assertion in case of Non DW DMA interface) would be signalled based on this value. Value should be sub-multiple of (RX_WMark + 1)* (F_DATA_WIDTH/H_DATA_WIDTH) and (FIFO_DEPTH - TX_WMark)* (F_DATA_WIDTH/ H_DATA_WIDTH) For example, if FIFO_DEPTH = 16, FDATA_WIDTH == H_DATA_WIDTH Allowed combinations for MSize and TX_WMark are: MSize = 1, TX_WMARK = 1-15 MSize = 4, TX_WMark = 8 MSize = 4, TX_WMark = 4 MSize = 4, TX_WMark = 12 MSize = 8, TX_WMark = 8 MSize = 8, TX_WMark = 4 Allowed combinations for MSize and RX_WMark are: MSize = 1, RX_WMARK = 0-14 MSize = 4, RX_WMark = 3 MSize = 4, RX_WMark = 7 MSize = 4, RX_WMark = 11 MSize = 8, RX_WMark = 7 Recommended: MSize = 8, TX_WMark = 8, RX_WMark = 7

Bit	Attr	Reset Value	Description
27:16	RW	0x000	rx_wmark FIFO threshold watermark level when receiving data to card. When FIFO data count reaches greater than this number, DMA/FIFO request is raised. During end of packet, request is generated regardless of threshold programming in order to complete any remaining data. In non-DMA mode, when receiver FIFO threshold (RXDR) interrupt is enabled, then interrupt is generated instead of DMA request. During end of packet, interrupt is not generated if threshold programming is larger than any remaining data. It is responsibility of host to read remaining bytes on seeing Data Transfer Done interrupt. In DMA mode, at end of packet, even if remaining bytes are less than threshold, DMA request does single transfers to flush out any remaining bytes before Data Transfer Done interrupt is set. 12 bits-1 bit less than FIFO-count of status register, which is 13 bits. Limitation: RX_WMark <= FIFO_DEPTH-2 Recommended: (FIFO_DEPTH/2) - 1; (means greater than (FIFO_DEPTH/2) - 1) NOTE: In DMA mode during CCS time-out, the DMA does not generate the request at the end of packet, even if remaining bytes are less than threshold. In this case, there will be some data left in the FIFO. It is the responsibility of the application to reset the FIFO after the CCS timeout
15:12	RO	0x0	reserved
11:0	RW	0x000	tx_wmark FIFO threshold watermark level when transmitting data to card. When FIFO data count is less than or equal to this number, DMA/FIFO request is raised. If Interrupt is enabled, then interrupt occurs. During end of packet, request or interrupt is generated, regardless of threshold programming. In non-DMA mode, when transmit FIFO threshold (TXDR) interrupt is enabled, then interrupt is generated instead of DMA request. During end of packet, on last interrupt, host is responsible for filling FIFO with only required remaining bytes (not before FIFO is full or after CIU completes data transfers, because FIFO may not be empty). In DMA mode, at end of packet, if last transfer is less than burst size, DMA controller does single cycles until required bytes are transferred. 12 bits -1 bit less than FIFO-count of status register, which is 13 bits. Limitation: TX_WMark >= 1; Recommended: FIFO_DEPTH/2; (means less than or equal to FIFO_DEPTH/2)

SDMMC_CDETECT
Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	card_detect_n Value on card_detect_n input ports; read-only bits. 0 represents presence of card

SDMMC_WRTPRT
Address: Operational Base + offset (0x0054)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	write_protect Value on card_write_prt input port. 1 represents write protection

SDMMC_TCBCNT
Address: Operational Base + offset (0x005c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	trans_card_byte_count Number of bytes transferred by CIU unit to card. Both SDMMC_TCBCNT and SDMMC_TBBCNT share same coherency register

SDMMC_TBBCNT
Address: Operational Base + offset (0x0060)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	trans_fifo_byte_count Number of bytes transferred between Host/DMA memory and FIFO. Both SDMMC_TCBCNT and SDMMC_TBBCNT share same coherency register

SDMMC_DEBNCE
Address: Operational Base + offset (0x0064)

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:0	RW	0xffffff	debounce_count Number of host clocks (clk) used by debounce filter logic; typical debounce time is 5-25 ms

SDMMC_USRID
Address: Operational Base + offset (0x0068)

Bit	Attr	Reset Value	Description
31:0	RW	0x07967797	usrid User identification register; value set by user. Default reset value can be picked by user while configuring core before synthesis. Can also be used as scratch pad register by user. The default value is determined by Configuration Value

SDMMC_VERID

Address: Operational Base + offset (0x006c)

Bit	Attr	Reset Value	Description
31:0	RO	0x5342270a	verid Version identification register; register value is hard-wired. Can be read by firmware to support different versions of core

SDMMC_HCON

Address: Operational Base + offset (0x0070)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	HCON Configuration Dependent. Hardware configurations selected by user before synthesizing core. Register values can be used to develop configuration-independent software drivers. [0]: CARD_TYPE 1'b0: MMC_ONLY 1'b1: SD_MMC [5:1]: NUM_CARDS - 1 [6]: H_BUS_TYPE 1'b0: APB 1'b1: AHB [9:7]: H_DATA_WIDTH 3'b000: 16 bits 3'b001: 32 bits 3'b010: 64 bits others: reserved [15:10]: H_ADDR_WIDTH 0 to 7: reserved 6'd8: 9 bits 6'd9: 10 bits … 6'd31: 32 bits 6'd32 to 63: reserved [17:16]: DMA_INTERFACE 2'b00: none 2'b01: DW_DMA 2'b10: GENERIC_DMA 2'b11: NON-DW-DMA [20:18]: GE_DMA_DATA_WIDTH 3'b000: 16 bits 3'b001: 32 bits 3'b010: 64 bits others: reserved [21]: FIFO_RAM_INSIDE 1'b0: outside 1'b1: inside [22]: IMPLEMENT_HOLD_REG 1'b0: no hold register 1'b1: hold register [23]: SET_CLK_FALSE_PATH 1'b0: no false path 1'b1: false path set [25:24]: NUM_CLK_DIVIDER-1 [26]: AREA_OPTIMIZED 1'b0: no area optimization 1'b1: Area optimization

SDMMC_UHS_REG

Address: Operational Base + offset (0x0074)

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16	RW	0x0	ddr_reg DDR mode. 1'b0: non-DDR mode 1'b1: DDR mode
15:0	RO	0x0	reserved

SDMMC_RSTN

Address: Operational Base + offset (0x0078)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	card_reset Hardware reset. 1'b0: active mode 1'b1: reset

SDMMC_BMOD

Address: Operational Base + offset (0x0080)

Bit	Attr	Reset Value	Description
31:11	RO	0x0	reserved
10:8	RW	0x0	PBL Programmable Burst Length. These bits indicate the maximum number of beats to be performed in one IDMAC transaction. The IDMAC will always attempt to burst as specified in PBL each time it starts a Burst transfer on the host bus. The permissible values are 1, 4, 8, 16, 32, 64, 128 and 256. This value is the mirror of MSIZE of FIFOTH register. In order to change this value, write the required value to FIFOTH register. This is an encode value as follows. 3'b000: 1 transfers 3'b001: 4 transfers 3'b010: 8 transfers 3'b011: 16 transfers 3'b100: 32 transfers 3'b101: 64 transfers 3'b110: 128 transfers 3'b111: 256 transfers Transfer unit is either 16, 32, or 64 bits, based on HDATA_WIDTH. PBL is a read-only value and is applicable only for Data Access; it does not apply to descriptor accesses
7	RW	0x0	DE IDMAC Enable. When set, the IDMAC is enabled

Bit	Attr	Reset Value	Description
6:2	RW	0x00	DSL Descriptor Skip Length. Specifies the number of HWord/Word/Dword (depending on 16/32/64-bit bus) to skip between two unchained descriptors. This is pplicable only for dual buffer structure
1	RW	0x0	FB Fixed Burst. Controls whether the AHB Master interface performs fixed burst transfers or not. When set, the AHB will use only SINGLE, INCR4, INCR8 or INCR16 during start of normal burst transfers. When reset, the AHB will use SINGLE and INCR burst transfer operations
0	RW	0x0	SWR Software Reset. When set, the DMA Controller resets all its internal registers. It is automatically cleared after 1 clock cycle

SDMMC_PLDMND

Address: Operational Base + offset (0x0084)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	PD Poll Demand. If the OWN bit of a descriptor is not set, the FSM goes to the Suspend state. The host needs to write any value into this register for the IDMAC FSM to resume normal descriptor fetch operation. This is a write only register

SDMMC_DBADDR

Address: Operational Base + offset (0x0088)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	SDL Start of Descriptor List. Contains the base address of the First Descriptor. The LSB bits [0/1/2:0] for 16/32/64-bit bus-width) are ignored and taken as all-zero by the IDMAC internally. Hence these LSB bits are read-only

SDMMC_IDSTS

Address: Operational Base + offset (0x008c)

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved

Bit	Attr	Reset Value	Description
16:13	RW	0x0	FSM DMAC FSM present state. 4'h0: DMA_IDLE 4'h1: DMA_SUSPEND 4'h2: DESC_RD 4'h3: DESC_CHK 4'h4: DMA_RD_REQ_WAI 4'h5: DMA_WR_REQ_WAI 4'h6: DMA_RD 4'h7: DMA_WR 4'h8: DESC_CLOSE
12:10	RW	0x0	EB Error Bits. Indicates the type of error that caused a Bus Error. Valid only with fatal Bus. 3'h1: Host Abort received during transmission 3'h2: Host Abort received during reception Others: Reserved
9	RW	0x0	AIS Abnormal Interrupt Summary. Logical OR of the following: SDMMC_IDSTS[2] Fatal Bus Interrupt SDMMC_IDSTS[4] DU bit Interrupt Only unmasked bits affect this bit. This is a sticky bit and must be cleared each time a corresponding bit that causes AIS to be set is cleared. Writing a 1 clears this bit
8	RW	0x0	NIS Normal Interrupt Summary. Logical OR of the following: SDMMC_IDSTS[0] Transmit Interrupt SDMMC_IDSTS[1] Receive Interrupt Only unmasked bits affect this bit. This is a sticky bit and must be cleared each time a corresponding bit that causes NIS to be set is cleared. Writing a 1 clears this bit
7:6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5	RW	0x0	CES Card Error Summary. Indicates the status of the transaction to/from the card; also present in SDMMC_RINTSTS. Indicates the logical OR of the following bits: EBE: End Bit Error RTO: Response Timeout/Boot Ack Timeout RCRC: Response CRC SBE: Start Bit Error DRTO: Data Read Timeout/BDS timeout DCRC: Data CRC for Receive RE: Response Error Writing a 1 clears this bit. The abort condition of the IDMAC depends on the setting of this CES bit. If the CES bit is enabled, then the IDMAC aborts on a "response error"; however, it will not abort if the CES bit is cleared
4	RW	0x0	DU Descriptor Unavailable Interrupt. This bit is set when the descriptor is unavailable due to OWN bit = 0 (DES0[31] =0). Writing a 1 clears this bit
3	RO	0x0	reserved
2	RW	0x0	FBE Fatal Bus Error Interrupt. When this bit is set, the DMA disables all its bus accesses. Writing a 1 clears this bit
1	RW	0x0	RI Receive Interrupt. Indicates the completion of data reception for a descriptor. Writing a 1 clears this bit
0	RW	0x0	TI Transmit Interrupt. Indicates that data transmission is finished for a descriptor. Writing 1 clears this bit

SDMMC_IDINTEN

Address: Operational Base + offset (0x0090)

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved

Bit	Attr	Reset Value	Description
9	RW	0x0	AI Abnormal Interrupt Summary Enable. When set, an abnormal interrupt is enabled. This bit enables the following bits: SDMMC_IDINTEN[2] Fatal Bus Error Interrupt SDMMC_IDINTEN[4] DU Interrupt
8	RW	0x0	NI Normal Interrupt Summary Enable. When set, a normal interrupt is enabled. When reset, a normal interrupt is disabled. This bit enables the following bits: SDMMC_IDINTEN[0] Transmit Interrupt SDMMC_IDINTEN[1] Receive Interrup
7:6	RO	0x0	reserved
5	RW	0x0	CES Card Error summary Interrupt Enable. When set, it enables the Card Interrupt summary
4	RW	0x0	DU Descriptor Unavailable Interrupt. When set along with Abnormal Interrupt Summary Enable, the DU interrupt is enabled
3	RO	0x0	reserved
2	RW	0x0	FBE Fatal Bus Error Enable. When set with Abnormal Interrupt Summary Enable, the Fatal Bus Error Interrupt is enabled. When reset, Fatal Bus Error Enable Interrupt is disabled
1	RW	0x0	RI Receive Interrupt Enable. When set with Normal Interrupt Summary Enable, Receive Interrupt is enabled. When reset, Receive Interrupt is disabled
0	RW	0x0	TI Transmit Interrupt Enable. When set with Normal Interrupt Summary Enable, Transmit Interrupt is enabled. When reset, Transmit Interrupt is disabled

SDMMC_DSCADDR

Address: Operational Base + offset (0x0094)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	HDA Host Descriptor Address Pointer. Cleared on reset. Pointer updated by IDMAC during operation. This register points to the start address of the current descriptor read by the IDMAC

SDMMC_BUFADDR

Address: Operational Base + offset (0x0098)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	HBA Host Buffer Address Pointer. Cleared on Reset. Pointer updated by IDMAC during operation. This register points to the current Data Buffer Address being accessed by the IDMAC

SDMMC_CARDTHRCTL

Address: Operational Base + offset (0x0100)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	CardRdThreshold Card Read Threshold size
15:2	RO	0x0	reserved
1	RW	0x0	BsyClrIntEn Busy Clear Interrupt generation. 1'b0: Busy Clear Interrupt disabled 1'b1: Busy Clear Interrupt enabled Note: The application can disable this feature if it does not want to wait for a Busy Clear Interrupt. For example, in a multi-card scenario, the application can switch to the other card without waiting for a busy to be completed. In such cases, the application can use the polling method to determine the status of busy. By default this feature is disabled and backward-compatible to the legacy drivers where polling is used
0	RW	0x0	CardRdThrEn Card Read Threshold Enable. 1'b0: Card Read Threshold disabled 1'b1: Card Read Threshold enabled. Host Controller initiates Read Transfer only if CardRdThreshold amount of space is available in receive FIFO

SDMMC_BACK_END_POWER

Address: Operational Base + offset (0x0104)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	back_end_power Back end power. 1'b0: Off; Reset 1'b1: Back-end Power supplied to card application

SDMMC_EMMC_DDR_REG
Address: Operational Base + offset (0x010c)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	HALF_START_BIT Control for start bit detection mechanism inside the Host Controller based on duration of start bit; each bit refers to one slot. For eMMC 4.5, start bit can be: 1'b0: Full cycle (HALF_START_BIT=0) 1'b1: Less than one full cycle (HALF_START_BIT=1) Set HALF_START_BIT=1 for eMMC 4.5 and above; set to 0 for SD applications

SDMMC_FIFO_BASE
Address: Operational Base + offset (0x0200)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	fifo_base_addr Fifo base addr

3.5 Interface Description

The interface and IOMUX setting for SDMMC, SDIO, EMMC are shown as follows.

Table 3-8 SDMMC Interface Description

Module Pin	Direction	Pin Name	IOMUX Setting
sdmmc_cclk	O	GPIO4_D5/SDMMC_CLK	GRF_GPIO4D_IOMUX[11:10] =2’b01
sdmmc_ccmd	I/O	GPIO4_D4/SDMMC_CMD	GRF_GPIO4D_IOMUX[9:8] =2’b01
sdmmc_cdata0	I/O	GPIO4_D0/SDMMC_D0	GRF_GPIO4D_IOMUX[1:0] =2’b01
sdmmc_cdata1	I/O	GPIO4_D1/SDMMC_D1	GRF_GPIO4D_IOMUX[3:2] =2’b01
sdmmc_cdata2	I/O	GPIO4_D2/SDMMC_D2/UART2_RX_M1	GRF_GPIO4D_IOMUX[5:4] =2’b01
sdmmc_cdata3	I/O	GPIO4_D3/SDMMC_D3/UART2_TX_M1	GRF_GPIO4D_IOMUX[7:6] =2’b01
sdmmc_cdetn	I	GPIO0_A3/SDMMC_DET	GRF_GPIO0A_IOMUX[7:6] =2’b01
sdmmc_ pwren	O	GPIO4_D6/SDMMC_PWREN	GRF_GPIO4D_IOMUX[13:12] =2’b01

Notes: I=input, O=output, I/O=input/output, bidirectional

Table 3-9 SDIO Interface Description

Module Pin	Direction	Pin Name	IOMUX Setting
sdio_cclk	O	GPIO4_A5/SDIO_CLK	GRF_GPIO14A_IOMUX[11:10] =2'b01
sdio_ccmd	I/O	GPIO4_A4/SDIO_CMD	GRF_GPIO4A_IOMUX[9:8] =2'b01
sdio_cdata0	I/O	GPIO4_A0/SDIO_D0	GRF_GPIO4A_IOMUX[1:0] =2'b01
sdio_cdata1	I/O	GPIO4_A1/SDIO_D1	GRF_GPIO4A_IOMUX[3:2] =2'b01
sdio_cdata2	I/O	GPIO4_A2/SDIO_D2	GRF_GPIO4A_IOMUX[5:4] =2'b01
sdio_cdata3	I/O	GPIO4_A3/SDIO_D3	GRF_GPIO4A_IOMUX[7:6] =2'b01
sdio_intn	I	GPIO0_A0/SDIO_INTN	GRF_GPIO0A_IOMUX[1:0] =2'b01
sdio_wrpt	I	GPIO0_A1/SDIO_WRPT	GRF_GPIO0A_IOMUX[3:2] =2'b01
sdio_pwren	O	GPIO0_A2/SDIO_PWREN	GRF_GPIO0A_IOMUX[5:4] =2'b01

Notes: I=input, O=output, I/O=input/output, bidirectional

Table 3-10 EMMC Interface Description

Module Pin	Direction	Pin Name	IOMUX Setting
emmc_cclk	O	GPIO3_B1/FLASH_CLE/EMMC_CLK	GRF_GPIO3B_IOMUX[3:2] =2'b10
emmc_ccmd	I/O	GPIO3_B0/FLASH_WRN/EMMC_CMD	GRF_GPIO3B_IOMUX[1:0] =2'b10
emmc_cdata0	I/O	GPIO3_A0/FLASH_D0/EMMC_D0/SFC_SIO0	GRF_GPIO3A_IOMUX[1:0] =2'b10
emmc_cdata1	I/O	GPIO3_A1/FLASH_D1/EMMC_D1/SFC_SIO1	GRF_GPIO3A_IOMUX[3:2] =2'b10
emmc_cdata2	I/O	GPIO3_A2/FLASH_D2/EMMC_D2/SFC_WP_SIO2	GRF_GPIO3A_IOMUX[5:4] =2'b10
emmc_cdata3	I/O	GPIO3_A3/FLASH_D3/EMMC_D3/SFC_HOLD_SIO3	GRF_GPIO3A_IOMUX[7:6] =2'b10
emmc_cdata4	I/O	GPIO3_A4/FLASH_D4/EMMC_D4/SFC_CLK	GRF_GPIO3A_IOMUX[9:8] =2'b10
emmc_cdata5	I/O	GPIO3_A5/FLASH_D5/EMMC_D5/SFC_CSN0	GRF_GPIO3A_IOMUX[11:10] =2'b10
emmc_cdata6	I/O	GPIO3_A6/FLASH_D6/EMMC_D6	GRF_GPIO3A_IOMUX[13:12] =2'b10
emmc_cdata7	I/O	GPIO3_A7/FLASH_D7/EMMC_D7	GRF_GPIO3A_IOMUX[15:14] =2'b10
emmc_pwren	O	GPIO3_B3/FLASH_ALE/EMMC_PWREN/SPI1_CLK	GRF_GPIO3B_IOMUX[7:6] =2'b10

Notes: I=input, O=output, I/O=input/output, bidirectional

3.6 Application Notes

3.6.1 Card-Detect and Write-Protect Mechanism

Following figure illustrates how the SD/MMC card detection and write-protect signals are connected. Most of the SD/MMC sockets have card-detect pins. When no card is present, card_detect_n is 1 due to the pull-up. When the card is inserted, the card-detect pin is shorted to ground, which makes card_detect_n go to 0. Similarly in SD cards, when the write-protect switch is toward the left, it shorts the write_protect port to ground.

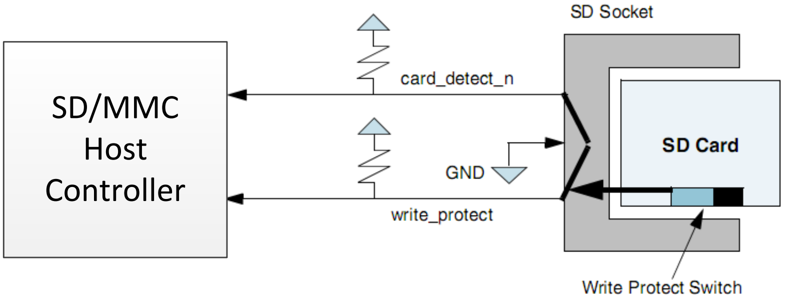


Fig. 3-9 SD/MMC Card-Detect and Write-Protect

3.6.2 SD/MMC Termination Requirement

Following Figure illustrates the SD/MMC termination requirements, which is required to pull up ccmd and cdata lines on the device bus. The recommended specification for pull-up on the ccmd line (Rcmd) is 4.7K - 100K for MMC, and 10K - 100K for an SD. The recommended pull-up on the cdata line (Rdat) is 50K - 100K.

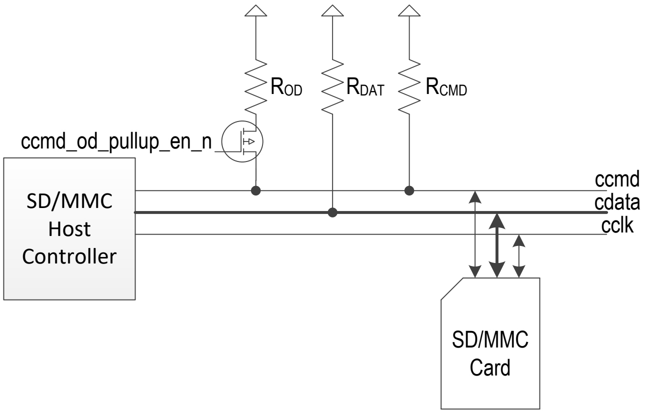


Fig. 3-10 SD/MMC Card Termination

Rcmd and Rod Calculation

The SD/MMC card enumeration happens at a very low frequency – 100-400KHz. Since the MMC bus is a shared bus between multiple cards, during enumeration open-drive mode is used to avoid bus conflict. Cards that drive 0 win over cards that drive "z". The pull-up in the command line pulls the bus to 1 when all cards drive "z". During normal data transfer, the host chooses only one card and the card driver switches to push-pull mode.

For example, if enumeration is done at 400KHz and the total bus capacitance is 200 pf, the pull-up needed during enumeration is:

2.2 RC = rise-time = 1/400KHz

R = 1/(2.2 * C * 100KHz)

  = 1/(2.2 x 200 x 10**-12 x 400 x 10**3)

  = 1/(17.6 x 10**-5)

  = 5.68K

The ROD and RCMD should be adjusted in such a way that the effective pull-up is at the maximum 5.68K during enumeration. If there are only a few cards in the bus, a fixed RCMD resister is sufficient and there is no need for an additional ROD pull-up during enumeration. You should also ensure the effective pull-up will not violate the Iol rating of the drivers.

In SD mode, since each card has a separate bus, the capacitance is less, typically in the order of 20-30pf (host capacitance + card capacitance + trace + socket capacitance). For example, if enumeration is done at 400KHz and the total bus capacitance is 20pf, the pull-up needed during enumeration is:

2.2 RC = rise-time = 1/400KHz

R = 1/(2.2 * C * 100KHz)

= 1/(2.2 x 20 x 10**-12 x 400 x 10**3)
= 1/(1.76 x 10**-5)
= 56.8K

Therefore, a fixed 56.8K permanent Rcmd is sufficient in SD mode to enumerate the cards. The driver of the SD/MMC on the "command" port needs to be only a push-pull driver. During enumeration, the SD/MMC emulates an open-drain driver by driving only a 0 or a"z" by controlling the ccmd_out and ccmd_out_en signals.

3.6.3 Software/Hardware Restriction

Before issuing a new data transfer command, the software should ensure that the card is not busy due to any previous data transfer command. Before changing the card clock frequency, the software must ensure that there are no data or command transfers in progress.

If the card is enumerated in SDR50, or DDR50 mode, then the application must program the use_hold_reg bit[29] in the SDMMC_CMD register to 1'b0 (phase shift of cclk_in_drv = 0) or 1'b1 (phase shift of cclk_in_drv>0). If the card is enumerated in SDR12 or SDR25 mode, the application must program the use_hold_reg bit[29] in the SDMMC_CMD register to 1'b1. This programming should be done for all data transfer commands and non-data commands that are sent to the card. When the use_hold_reg bit is programmed to 1'b0, the Host Controller bypasses the Hold Registers in the transmit path. The value of this bit should not be changed when a Command or Data Transfer is in progress. For more details on using use_hold_reg and the implementation requirements for meeting the Card input hold time, refer to "Recommended Usage" in following table.

Table 3-11 Recommended Usage of use_hold_reg

No.	Speed Mode	use_hold_reg	cclk_in (MHz)	clk_in_drv (MHz)	clk_divider	Phase shift
1	SDR104	1'b0	200	200	0	0
2	SDR104	1'b1	200	200	0	Tunable> 0
3	SDR50	1'b0	100	100	0	0
4	SDR50	1'b1	100	100	0	Tunable> 0
5	DDR50 (8bit)	1'b0	100	100	1	0
6	DDR50 (8bit)	1'b1	100	100	1	Tunable> 0
7	DDR50 (4bit)	1'b0	50	50	0	0
8	DDR50 (4bit)	1'b1	50	50	0	Tunable> 0
9	SDR25	1'b1	50	50	0	Tunable> 0
10	SDR12	1'b1	50	50	1	Tunable> 0

To avoid glitches in the card clock outputs, the software should use the following steps when changing the card clock frequency:

1) Before disable the clocks, ensure that the card is not busy due to any previous data command. To determine this, check for 0 in bit9 of STATUS register.

2) Update the Clock Enable register to disable all clocks. To ensure completion of any previous command before this update, send a command to the CIU to update the clock registers by setting:

- start_cmd bit
- "update clock registers only" bits
- "wait_previous data complete" bit

Wait for the CIU to take the command by polling for 0 on the start_cmd bit.

3) Set the start_cmd bit to update the Clock Divider and/or Clock Source registers, and send a command to the CIU in order to update the clock registers; wait for the CIU to take the command.

4) Set start_cmd to update the Clock Enable register in order to enable the required clocks and send a command to the CIU to update the clock registers; wait for the CIU to take the command.

In non-DMA mode, while reading from a card, the Data Transfer Over (SDMMC_RINTSTS[3]) interrupt occurs as soon as the data transfer from the card is over. There still could be some data left in the FIFO, and the RX_WMark interrupt may or may not occur, depending on the remaining bytes in the FIFO. Software should read any remaining bytes upon seeing the Data Transfer Over (DTO) interrupt. While using the external DMA interface for reading from a card, the DTO interrupt occurs only after all the data is flushed to memory by the DMA interface unit.

While writing to a card in external DMA mode, if an undefined-length transfer is selected by setting the Byte Count Register to 0, the DMA logic will likely request more data than it will send to the card, since it has no way of knowing at which point the software will stop the transfer. The DMA request stops as soon as the DTO is set by the CIU.

If the software issues a controller_reset command by setting control register bit[0] to 1, all the CIU state machines are reset; the FIFO is not cleared. The DMA sends all remaining bytes to the host. In addition to a card-reset, if a FIFO reset is also issued, then:

- Any pending DMA transfer on the bus completes correctly
- DMA data read is ignored
- Write data is unknown(x)

Additionally, if dma_reset is also issued, any pending DMA transfer is abruptly terminated. When the DMA is used, the DMA controller channel should also be reset and reprogrammed.

If any of the previous data commands do not properly terminate, then the software should issue the FIFO reset in order to remove any residual data, if any, in the FIFO. After asserting the FIFO reset, you should wait until this bit is cleared.

One data-transfer requirement between the FIFO and host is that the number of transfers should be a multiple of the FIFO data width (32bits). For example, you want to write only 15 bytes to an SD/MMC card (SDMMC_BYTCNT), the host should write 16 bytes to the FIFO or program the DMA to do 16-byte transfers. The software can still program the Byte Count register to only 15, at which point only 15 bytes will be transferred to the card. Similarly, when 15 bytes are read from a card, the host should still read all 16 bytes from the FIFO.

It is recommended that you not change the FIFO threshold register in the middle of data transfers.

3.6.4 Programming Sequence

1. Initialization

Following figure illustrates the initialization flow.

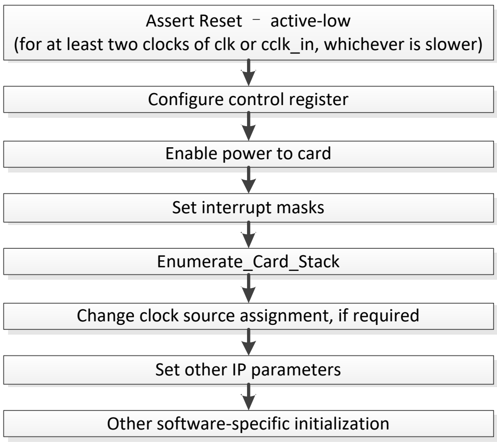

Assert Reset – active-low
(for at least two clocks of clk or cclk_in, whichever is slower)
Configure control register
Enable power to card
Set interrupt masks
Enumerate_Card_Stack
Change clock source assignment, if required
Set other IP parameters
Other software-specific initialization

Fig. 3-11 Host Controller Initialization Sequence

Once the power and clocks are stable, reset_n should be asserted(active-low) for at least two clocks of clk or cclk_in, whichever is slower. The reset initializes the registers, ports, FIFO-pointers, DMA interface controls, and state-machines in the design. After power-on reset, the software should do the following:

1) Configure control register – For MMC mode, enable the open-drain pullup by setting enable_OD_pullup(bit24) in the control register.
2) Enable power to cards – Before enabling the power, confirm that the voltage setting to the voltage regulators is correct. Enable power to the connected cards by setting the corresponding bit to 1 in the Power Enable register. Wait for the power ramp-up time.

3) Set masks for interrupts by clearing appropriate bits in the Interrupt Mask register. Set the global int_enable bit of the Control register. It is recommended that you write 0xffff_ffff to the Raw Interrupt register in order to clear any pending interrupts before setting the int_enable bit.
4) Enumerate card stack – Each card is enumerated according to card type; for details, refer to "Enumerated Card Stack". For enumeration, you should restrict the clock frequency to 400KHz.
5) Changing clock source assignment – set the card frequency using the clock-divider and clock-source registers; for details, refer to "Clock Programming". MMC cards operate at a maximum of 20MHz (at maximum of 52MHz in high-speed mode). SD mode operates at a maximum of 25MHz (at maximum of 50MHz in high-speed mode).
6) Set other parameters, which normally do not need to be changed with every command, with a typical value such as timeout values in cclk_out according to SD/MMC specifications.

- ResponseTimeOut = 0x64
- DataTimeOut = highest of one of the following:
- (10*((TAAC*Fop)+(100*NSAC))
- Host FIFO read/write latency from FIFO empty/full
- Set the debounce value to 25ms(default:0x0fffff) in host clock cycle units in the DEBNCE register.
- FIFO threshold value in bytes in the SDMMC_FIFOTH register.

13

2. Enumerated Card Stack

The card stack does the following:

- Enumerates all connected cards
- Sets the RCA for the connected cards
- Reads card-specific information
- Stores card-specific information locally

Enumeration depends on the operating mode of the SD/MMC card; the card type is first identified and the appropriate card enumeration routine is called.

1) Check if the card is connected.
2) Clear the card type register to set the card width as a single bit. For the given card number, clear the corresponding bits in the card_type register. Clear the register bit for a 1-bit, 4-bit bus width. For example, for card number=1, clear bit 0 and bit 16 of the card_type register.
3) Set clock frequency to FOD=400KHz, maximum – Program clock divider0 (bits 0-7 in the CLKDIV register) value to one-half of the cclk_in frequency divided by 400KHz. For example, if cclk_in is 20MHz, then the value is 20, 000/(2*400)=25.
4) Identify the card type; that is, SD, MMC, or SDIO.
   a. Send CMD5 first. If a response is received, then the card is SDIO
   b. If not, send CMD8 with the following Argument
      Bit[31:12] = 20'h0 //reserved bits
      Bit[11:8] = 4'b0001 //VHS value
      Bit[7:0] = 8'b10101010 //Preferred Check Pattern by SD2.0
   c. If Response is received the card supports High Capacity SD2.0 then send ACMD41 with the following Argument
      Bit[31] = 1'b0; //Reserved bits
      Bit[30] = 1'b1; //High Capacity Status
      Bit[29:24] = 6'h0; //Reserved bits
      Bit[23:0] = Supported Voltage Range
   d. If Response is received for ACMD41 then the card is SD. Otherwise the card is MMC.
   e. If response is not received for initial CMD8 then card does not support High Capacity SD2.0, then issue CMD0 followed by ACMD41 with the following Argument
      Bit[31] = 1'b0; //Reserved bits
      Bit[30] = 1'b0; //High Capacity Status
      Bit[29:24] = 6'h0; //Reserved bits
      Bit[23:0] = Supported Voltage Range

5) Enumerate the card according to the card type.
6) Use a clock source with a frequency = Fod (that is, 400KHz) and use the following enumeration command sequence:

- SD card – Send CMD0, CMD8, ACMD41, CMD2, CMD3.
- MMC – Send CMD0, CMD1, CMD2, CMD3.

14

3. Power Control

You can implement power control using the following registers, along with external circuitry:

- Control register bits card_voltage_a and card_voltage_b – Status of these bits is reflected at the IO pins. The bits can be used to generate or control the supply voltage that the memory cards require.
- Power enable register – Control power to individual cards.

Programming these two register depends on the implemented external circuitry. While turning on or off the power enable, you should confirm that power supply settings are correct. Power to all cards usually should be disabled while switching off the power.

4. Clock Programming

The Host Controller supports one clock sources. The clock to an individual card can be enabled or disabled. Registers that support this are:

- SDMMC_CLKDIV – Programs individual clock source frequency. SDMMC_CLKDIV limited to 0 or 1 is recommended.
- SDMMC_CLKSRC – Assign clock source for each card.
- SDMMC_CLKENA – Enables or disables clock for individual card and enables low-power mode, which automatically stops the clock to a card when the card is idle for more than 8 clocks.

The Host Controller loads each of these registers only when the start_cmd bit and the Update_clk_regs_only bit in the SDMMC_CMD register are set. When a command is successfully loaded, the Host Controller clears this bit, unless the Host Controller already has another command in the queue, at which point it gives an HLE(Hardware Locked Error). Software should look for the start_cmd and the Update_clk_regs_only bits, and should also set the wait_prvdata_complete bit to ensure that clock parameters do not change during data transfer. Note that even though start_cmd is set for updating clock registers, the Host Controller does not raise a command_done signal upon command completion.

The following shows how to program these registers:

1) Confirm that no card is engaged in any transaction; if there is a transaction, wait until it finishes.
2) Stop all clocks by writing xxxx0000 to the SDMMC_CLKENA register. Set the start_cmd, Update_clk_regs_only, and wait_prvdata_complete bits in the SDMMC_CMD register. Wait until start_cmd is cleared or an HLE is set; in case of an HLE, repeat the command.
3) Program the SDMMC_CLKDIV and SDMMC_CLKSRC registers, as required. Set the start_cmd, Update_clk_regs_only, and wait_prvdata_complete bits in the SDMMC_CMD register. Wait until start_cmd is cleared or an HLE is set; in case of an HLE, repeat the command.
4) Re-enable all clocks by programming the SDMMC_CLKENA register. Set the start_cmd, Update_clk_regs_only, and wait_prvdata_complete bits in the SDMMC_CMD register. Wait until start_cmd is cleared or an HLE is set; in case of an HLE, repeat the command.

15

5. No-Data Command With or Without Response Sequence

To send any non-data command, the software needs to program the SDMMC_CMD register @0x2C and the SDMMC_CMDARG register @0x28 with appropriate parameters. Using these two registers, the Host Controller forms the command and sends it to the command bus. The Host Controller reflects the errors in the command response through the error bits of the SDMMC_RINTSTS register.

When a response is received – either erroneous or valid – the Host Controller sets the command_done bit in the SDMMC_RINTSTS register. A short response is copied in Response Register0, while along response is copied to all four response registers @0x30, 0x34, 0x38, and 0x3C. The Response3 register bit 31 represents the MSB, and the Response0 register bit

0 represents the LSB of a long response.

For basic commands or non-data commands, follow these steps:

1) Program the Command register @0x28 with the appropriate command argument parameter.
2) Program the Command register @0x2C with the settings in following table.

Table 3-12 Command Settings for No-Data Command

Parameter	Value	Description
Default		
start_cmd	1	-
use_hold_reg	1/0	Choose value based on speed mode being used;ref to “use_hold_reg” on SDMMC_CMD register
update_clk_regs_only	0	No clock parameters update command
data_expected	0	No data command
card number	0	Actual card number(one controller only connect one card, the num is No. 0)
cmd_index	command-index	-
send_initialization	0	Can be 1, but only for card reset commands, such as CMD0
stop_abort_cmd	0	Can be 1 for commands to stop data transfer, such as CMD12
response_length	0	Can be 1 for R2(long) response
response_expect	1	Can be 0 for commands with no response; for example, CMD0, CMD4, CMD15, and so on
User-selectable		
wait_prvdata_complete	1	Before sending command on command line, host should wait for completion of any data command in process, if any (recommended to always set this bit, unless the current command is to query status or stop data transfer when transfer is in progress)
check_response_crc	1	If host should crosscheck CRC of response received

1) Wait for command acceptance by host. The following happens when the command is loaded into the Host Controller:
   - Host Controller accepts the command for execution and clears the start_cmd bit in the SDMMC_CMD register, unless one command is in process, at which point the Host Controller can load and keep the second command in the buffer.
   - If the Host Controller is unable to load the command – that is, a command is already in progress, a second command is in the buffer, and a third command is attempted – then it generates an HLE (hardware-locked error).
2) Check if there is an HLE.
3) Wait for command execution to complete. After receiving either a response from a card or response timeout, the Host Controller sets the command_done bit in the SDMMC_RINTSTS register. Software can either poll for this bit or respond to a generated interrupt.

4) Check if response_timeout error, response_CRC error, or response error is set. This can be done either by responding to an interrupt raised by these errors or by polling bits 1, 6, and 8 from the SDMMC_RINTSTS register @0x44. If no response error is received, then the response is valid. If required, the software can copy the response from the response registers @0x30-0x3C.

Software should not modify clock parameters while a command is being executed.

6. Data Transfer Commands

Data transfer commands transfer data between the memory card and the Host Controller. To send a data command, the Host Controller needs a command argument, total data size, and block size. Software can receive or send data through the FIFO.

Before a data transfer command, software should confirm that the card is not busy and is in a transfer state, which can be done using the CMD13 and CMD7 commands, respectively.

For the data transfer commands, it is important that the same bus width that is programmed in the card should be set in the card type register @0x18.

The Host Controller generates an interrupt for different conditions during data transfer, which are reflected in the SDMMC_RINTSTS register @0x44 as:

1) Data_Transfer_Over (bit 3) – When data transfer is over or terminated. If there is a response timeout error, then the Host Controller does not attempt any data transfer and the “Data Transfer Over” bit is never set.
2) Transmit_FIFO_Data_request (bit 4) – FIFO threshold for transmitting data was reached; software is expected to write data, if available, in FIFO.
3) Receive_FIFO_Data_request (bit 5) – FIFO threshold for receiving data was reached; software is expected to read data from FIFO.
4) Data starvation by Host timeout (bit 10) – FIFO is empty during transmission or is full during reception. Unless software writes data for empty condition or reads data for full condition, the Host Controller cannot continue with data transfer. The clock to the card has been stopped.
5) Data read timeout error (bit 9) – Card has not sent data within the timeout period.
6) Data CRC error (bit 7) – CRC error occurred during data reception.
7) Start bit error (bit 13) – Start bit was not received during data reception.
8) End bit error (bit 15) – End bit was not received during data reception or for a write operation; a CRC error is indicated by the card.

Conditions 6, 7, and 8 indicate that the received data may have errors. If there was a response timeout, then no data transfer occurred.

7. Single-Block or Multiple-Block Read

Steps involved in a single-block or multiple-block read are:

1) Write the data size in bytes in the SDMMC_BYTCNT register @0x20.
2) Write the block size in bytes in the SDMMC_BLKSIZ register @0x1C. The Host Controller expects data from the card in blocks of size SDMMC_BLKSIZ each.
3) Program the SDMMC_CMDARG register @0x28 with the data address of the beginning of a data read.
4) Program the Command register with the parameters listed in following table. For SD and MMC cards, use CMD17 for a single-block read and CMD18 for a multiple-block read. For SDIO cards, use CMD53 for both single-block and multiple-block transfers.

Table 3-13 Command Setting for Single or Multiple-Block Read

Parameter	Value	Description
Default		
start_cmd	1	-
use_hold_reg	1/0	Choose value based on speed mode being used;ref to “use_hold_reg” on SDMMC_CMD register
update_clk_regs_only	0	No clock parameters update command
card number	0	Actual card number(one

Parameter	Value	Description
		controller only connect one card, the num is No.0)
send_initialization	0	Can be 1, but only for card reset commands, such as CMD0
stop_abort_cmd	0	Can be 1 for commands to stop data transfer, such as CMD12
send_auto_stop	0/1	-
transfer_mode	0	Block transfer
read_write	0	Read from card
data_expected	1	Data command
response_length	0	Can be 1 for R2(long) response
response_expect	1	Can be 0 for commands with no response; for example, CMD0, CMD4, CMD15, and so on
User-selectable		
cmd_index	command-index	-
wait_prvdata_complete	1	0: Sends command immediately 1: Sends command after previous data transfer ends
check_response_crc	1	0: Host Controller should not check response CRC 1: Host Controller should check response CRC

After writing to the SDMMC_CMD register, the Host Controller starts executing the command; when the command is sent to the bus, the command_done interrupt is generated.

- Software should look for data error interrupts; that is, bits 7, 9, 13, and 15 of the SDMMC_RINTSTS register. If required, software can terminate the data transfer by sending a STOP command.
- Software should look for Receive_FIFO_Data_request and/or data starvation by host timeout conditions. In both cases, the software should read data from the FIFO and make space in the FIFO for receiving more data.
- When a Data_Transfer_Over interrupt is received, the software should read the remaining data from the FIFO.

16

8. Single-Block or Multiple-Block Write

Steps involved in a single-block or multiple-block write are:

1) Write the data size in bytes in the BYTCNT register @0x20.
2) Write the block size in bytes in the SDMMC_BLKSIZ register @0x1C; the Host Controller sends data in blocks of size SDMMC_BLKSIZ each.
3) Program SDMMC_CMDARG register @0x28 with the data address to which data should be written.
4) Write data in the FIFO; it is usually best to start filling data the full depth of the FIFO.
5) Program the Command register with the parameters listed in following table.

Table 3-14 Command Settings for Single or Multiple-Block Write

Parameter	Value	Description
Default		
start_cmd	1	-
use_hold_reg	1/0	Choose value based on speed mode being used;ref to “use_hold_reg” on

Parameter	Value	Description
		SDMMC_CMD register
update_clk_regs_only	0	No clock parameters update command
card number	0	Actual card number(one controller only connect one card, the num is No. 0)
send_initialization	0	Can be 1, but only for card reset commands, such as CMD0
stop_abort_cmd	0	Can be 1 for commands to stop data transfer, such as CMD12
send_auto_stop	0/1	-
transfer_mode	0	Block transfer
read_write	1	Write to card
data_expected	1	Data command
response_length	0	Can be 1 for R2(long) response
response_expect	1	Can be 0 for commands with no response; for example, CMD0, CMD4, CMD15, and so on
User-selectable		
cmd_index	command-index	-
wait_prvdata_complete	1	0: Sends command immediately 1: Sends command after previous data transfer ends
check_response_crc	1	0: Host Controller should not check response CRC 1: Host Controller should check response CRC

After writing to the SDMMC_CMD register, Host Controller starts executing a command; when the command is sent to the bus, a command_done interrupt is generated.

- Software should look for data error interrupts; that is, for bits 7, 9, and 15 of the SDMMC_RINTSTS register. If required, software can terminate the data transfer by sending the STOP command.
- Software should look for Transmit_FIFO_Data_Request and/or timeout conditions from data starvation by the host. In both cases, the software should write data into the FIFO.
- When a Data_Transfer_Over interrupt is received, the data command is over. For an open-ended block transfer, if the byte count is 0, the software must send the STOP command. If the byte count is not 0, then upon completion of a transfer of a given number of bytes, the Host Controller should send the STOP command, if necessary. Completion of the AUTO-STOP command is reflected by the Auto_command_done interrupt – bit 14 of the SDMMC_RINTSTS register. A response to AUTO_STOP is stored in SDMMC_RESP1 @0x34.

9. Stream Read

A stream read is like the block read mentioned in "Single-Block or Multiple-Block Read", except for the following bits in the Command register:

```
transfer_mode = 1; //Stream transfer
cmd_index = CMD20;
```

A stream transfer is allowed for only a single-bit bus width.

10. Stream Write

A stream write is exactly like the block write mentioned in "Single-Block or Multiple-Block

Write", except for the following bits in the Command register:
transfer_mode = 1;//Stream transfer
cmd_index = CMD11;
In a stream transfer, if the byte count is 0, then the software must send the STOP command. If the byte count is not 0, then when a given number of bytes completes a transfer, the Host Controller sends the STOP command. Completion of this AUTO_STOP command is reflected by the Auto_command_done interrupt. A response to an AUTO_STOP is stored in the SDMMC_RESP1 register@0x34.
A stream transfer is allowed for only a single-bit bus width.

11. Packed Commands

In order to reduce overhead, read and write commands can be packed in groups of commands—either all read or all write—that transfer the data for all commands in the group in one transfer on the bus.
Packed commands can be of two types:

- Packed Write: CMD23 →CMD25
- Packed Read: CMD23 → CMD25 → CMD23 → CMD18

Packed commands are put in packets by the application software and are transparent to the core.

12. Sending Stop or Abort in Middle of Transfer

The STOP command can terminate a data transfer between a memory card and the Controller, while the ABORT command can terminate an I/O data transfer for only the SDIO_IOONLY and SDIO_COMBO cards.

- Send STOP command – Can be sent on the command line while a data transfer is in progress; this command can be sent at any time during a data transfer.

You can also use an additional setting for this command in order to set the Command register bits (5-0) to CMD12 and set bit 14 (stop_abort_cmd) to 1. If stop_abort_cmd is not set to 1, the Controller does not know that the user stopped a data transfer. Reset bit 13 of the Command register (wait_prvdata_complete) to 0 in order to make the Controller send the command at once, even though there is a data transfer in progress.

- Send ABORT command – Can be used with only an SDIO_IOONLY or SDIO_COMBO card. To abort the function that is transferring data, program the function number in ASx bits (CCCR register of card, address 0x06, bits (0-2) using CMD52.

17

13. Read_Wait Sequence

Read_wait is used with only the SDIO card and can temporarily stall the data transfer—either from function or memory—and allow the host to send commands to any function within the SDIO device. The host can stall this transfer for as long as required. The Host Controller provides the facility to signal this stall transfer to the card. The steps for doing this are:

1) Check if the card supports the read_wait facility; read SRW (bit 2) of the CCCR register @0x08. If this bit is 1, then all functions in the card support the read_wait facility. Use CMD52 to read this bit.
2) If the card supports the read_wait signal, then assert it by setting the read_wait (bit 6) in the SDMMC_CTRL register @0x00.
3) Clear the read_wait bit in the SDMMC_CTRL register.

18

14. Controller/DMA/FIFO Reset Usage

- Controller reset – Resets the controller by setting the controller_reset bit (bit 0) in the SDMMC_CTRL register; this resets the CIU and state machines, and also resets the BIU-to-CIU interface. Since this reset bit is self-clearing, after issuing the reset, wait until this bit is cleared.
- FIFO reset - Resets the FIFO by setting the fifo_reset bit (bit 1) in the SDMMC_CTRL register; this resets the FIFO pointers and counters of the FIFO. Since this reset bit is self-clearing, after issuing the reset, wait until this bit is cleared.

In external DMA transfer mode, even when the FIFO pointers are reset, if there is a DMA transfer in progress, it could push or pop data to or from the FIFO; the DMA itself completes correctly. In order to clear the FIFO, the software should issue an additional FIFO reset and clear any FIFO underrun or overrun errors in the SDMMC_RAWINTS register caused by the DMA transfers after the FIFO was reset.

15. Card Read Threshold

When an application needs to perform a Single or Multiple Block Read command, the application must program the SDMMC_CARDTHRCTL register with the appropriate Card Read Threshold size (CardRdThreshold) and set the Card Read Threshold Enable (CardRdThrEnable) bit to 1'b1. This additional programming ensures that the Host controller sends a Read Command only if there is space equal to the CardRDThreshold available in the Rx FIFO. This in turn ensures that the card clock is not stopped in the middle a block of data being transmitted from the card. The Card Read Threshold can be set to the block size of the transfer, which guarantees that there is a minimum of one block size of space in the RxFIFO before the controller enables the card clock. The Card Read Threshold is required when the Round Trip Delay is greater than 0.5cclk_in period.

16. Error Handling

The Host Controller implements error checking; errors are reflected in the SDMMC_RAWINTS register@0x44 and can be communicated to the software through an interrupt, or the software can poll for these bits. Upon power-on, interrupts are disabled (int_enable in the SDMMC_CTRL register is 0), and all the interrupts are masked (bits 0-31 of the SDMMC_INTMASK register; default is 0).
Error handling:

- Response and data timeout errors – For response timeout, software can retry the command. For data timeout, the Host Controller has not received the data start bit – either for the first block or the intermediate block – within the timeout period, so software can either retry the whole data transfer again or retry from a specified block onwards. By reading the contents of the SDMMC_TCBCNT later, the software can decide how many bytes remain to be copied.
- Response errors – Set when an error is received during response reception. In this case, the response that copied in the response registers is invalid. Software can retry the command.
- Data errors – Set when error in data reception are observed; for example, data CRC, start bit not found, end bit not found, and so on. These errors could be set for any block-first block, intermediate block, or last block. On receipt of an error, the software can issue a STOP or ABORT command and retry the command for either whole data or partial data.
- Hardware locked error – Set when the Host Controller cannot load a command issued by software. When software sets the start_cmd bit in the SDMMC_CMD register, the Host Controller tries to load the command. If the command buffer is already filled with a command, this error is raised. The software then has to reload the command.
- FIFO underrun/overrun error – If the FIFO is full and software tries to write data in the FIFO, then an overrun error is set. Conversely, if the FIFO is empty and the software tries to read data from the FIFO, an underrun error is set. Before reading or writing data in the FIFO, the software should read the fifo_empty or fifo_full bits in the Status register.
- Data starvation by host timeout – Raised when the Host Controller is waiting for software intervention to transfer the data to or from the FIFO, but the software does not transfer within the stipulated timeout period. Under this condition and when a read transfer is in process, the software should read data from the FIFO and create space for further data reception. When a transmit operation is in process, the software should fill data in the FIFO in order to start transferring data to the card.
- CRC Error on Command – If a CRC error is detected for a command, the CE-ATA device does not send a response, and a response timeout is expected from the Host Controller. The ATA layer is notified that an MMC transport layer error occurred.

Notes: During a multiple-block data transfer, if a negative CRC status is received from the device, the data *path signals a data CRC error to the BIU by setting the data CRC error bit in the SDMMC_RINTSTS register. It then continues further data transmission until all the bytes are transmitted.*

3.6.5 Voltage Switching

The Host Controller supports SD 3.0 Ultra High Speed (UHS-1) and is capable of voltage switching in SD-mode, which can be applied to SD High-Capacity (SDHC) and SD Extended Capacity (SDXC) cards. UHS-1 supports only 4-bit mode.

However, whether the IO voltage of 1.8v supported or not is depended on the SoC design.

SD 3.0 UHS-1 supports the following transfer speed modes for UHS-50 and/or UHS-104 cards:

- DS – default-speed up to 25MHz, 3.3V signaling
- HS – high-speed up to 50MHz, 3.3V signaling
- SDR12 – SDR up to SDR 25MHz, 1.8V signaling
- SDR25 – SDR up to 50MHz, 1.8V signaling
- SDR50 – SDR up to 100MHz, 1.8V signaling
- DDR50 – DDR up to 50MHz, 1.8V signaling

Voltage selection can be done in only SD mode. The first CMD0 selects the bus mode-either SD mode or SPI mode. The card must be in SD mode in order for 1.8V signaling mode to apply, during which time the card cannot be switched to SPI mode or 3.3V signaling without a power cycle.

If the System BIOS in an embedded system already knows that it is connected to an SD 3.0 card, then the driver programs the Controller to initiate ACMD41. The software knows from the response of ACMD41 whether or not the card supports voltage switching to 1.8V.

- If bit 32 of ACMD41 response is 1'b1: card supports voltage switching and next command-CMD11-invokes voltage switching sequence. After CMD11 is started, the software must program the IO voltage selection register based on the soc architecture.
- If bit 32 of ACMD41 response is 1'b0: card does not support voltage switching and CMD11 should not be started.

If the card and host controller accept voltage switching, then they support UHS-1 modes of data transfer. After the voltage switch to 1.8V, SDR12 is the default speed.

Since the UHS-1 can be used in only 4-bit mode, the software must start ACMD6 and change the card data width to 4-bit mode; ACMD6 is driven in any of the UHS-1 speeds. If the host wants to select the DDR mode of data transfer, then the software must program the SDMMC_DDR_REG register in the CSR space with the appropriate card number.

To choose from any of the SDR or DDR modes, appropriate values should be programmed in the SDMMC_CLKDIV register.

1. Voltage Switch Operation

The Voltage Switch operation must be performed in SD mode only.

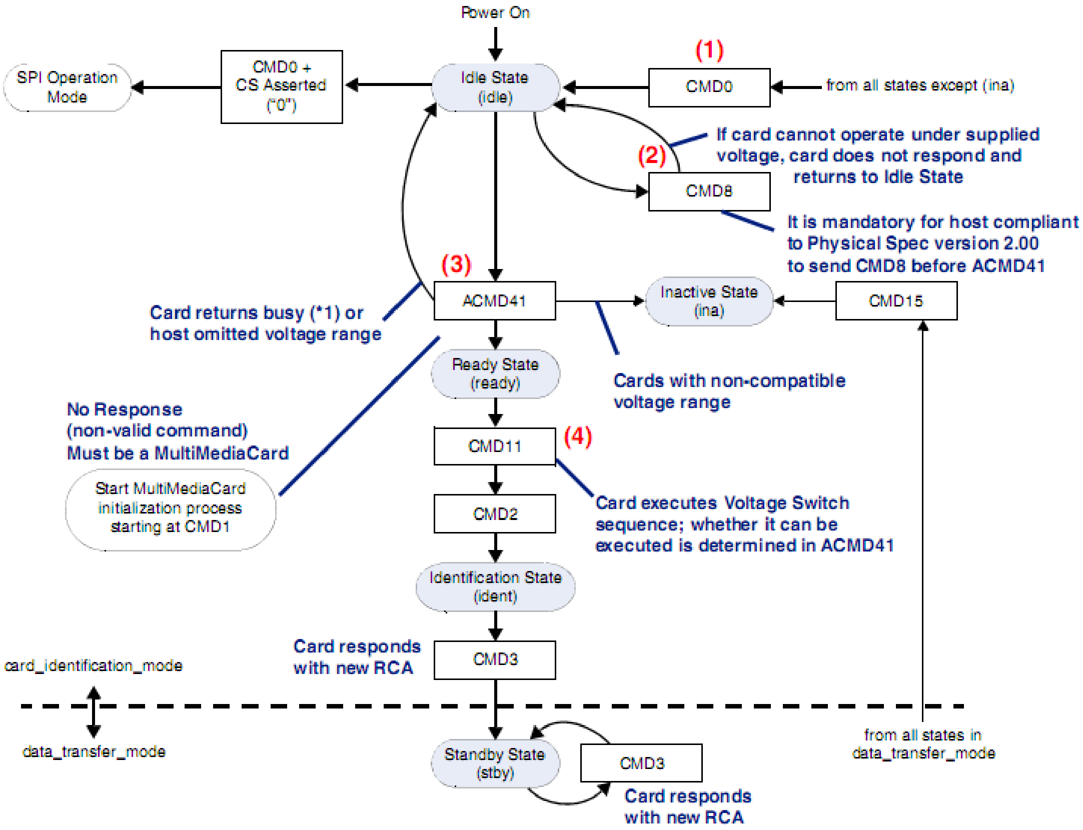
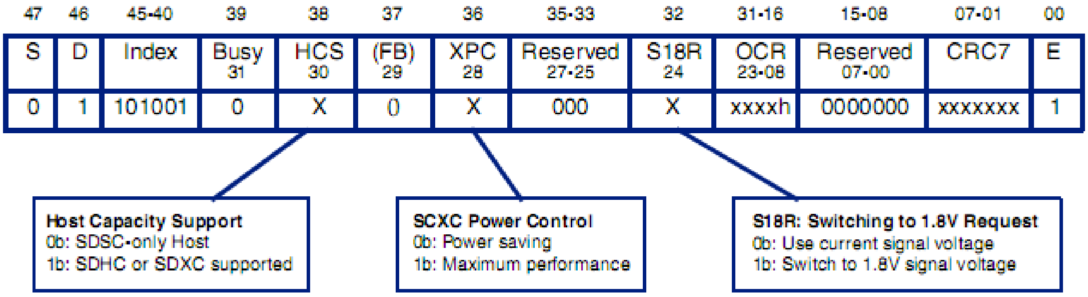

Power On
(1)
CMD0
from all states except (ina)
SPI Operation Mode
CMD0 + CS Asserted ("0")
Idle State (idle)
(2)
CMD8
If card cannot operate under supplied voltage, card does not respond and returns to Idle State
It is mandatory for host compliant to Physical Spec version 2.00 to send CMD8 before ACMD41
(3)
ACMD41
Inactive State (ina)
CMD15
Card returns busy (*1) or host omitted voltage range
Ready State (ready)
Cards with non-compatible voltage range
No Response (non-valid command) Must be a MultiMediaCard
Start MultiMediaCard initialization process starting at CMD1
CMD11
(4)
CMD2
Card executes Voltage Switch sequence; whether it can be executed is determined in ACMD41
Identification State (ident)
Card responds with new RCA
CMD3
card_identification_mode
data_transfer_mode
Standby State (stby)
CMD3
Card responds with new RCA
from all states in data_transfer_mode

Fig. 3-12 Voltage Switching Command Flow Diagram

The following outlines the steps for the voltage switch programming sequence

1) Software Driver starts CMD0, which selects the bus mode as SD.
2) After the bus is in SD card mode, CMD8 is started in order to verify if the card is compatible with the SD Memory Card Specification, Version 2. 00. CMD8 determines if the card is capable of working within the host supply voltage specified in the VHS (19:16) field of the CMD; the card supports the current host voltage if a response to CMD8 is received.
3) ACMD 41 is started. The response to this command informs the software if the card supports voltage switching; bits 38, 36, and 32 are checked by the card argument of ACMD41; refer to following figure.



47	46	45-40	39	38	37	36	35-33	32	31-16	15-08	07-01	00
S	D	Index	Busy 31	HCS 30	(FB) 29	XPC 28	Reserved 27-25	S18R 24	OCR 23-08	Reserved 07-00	CRC7	E
0	1	101001	0	X	0	X	000	X	xxxxh	0000000	xxxxxxx	1

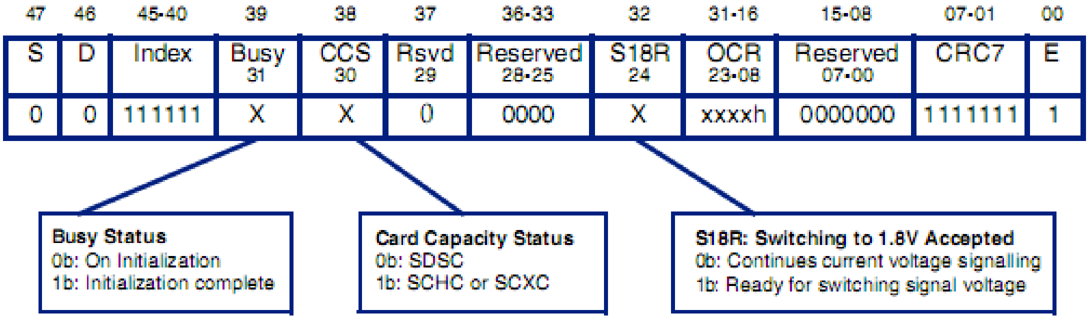
Host Capacity Support
0b: SDSC-only Host
1b: SDHC or SDXC supported

SCXC Power Control
0b: Power saving
1b: Maximum performance

S18R: Switching to 1.8V Request
0b: Use current signal voltage
1b: Switch to 1.8V signal voltage

Fig. 3-13 ACMD41 Argument

- Bit 30 informs the card if host supports SDHC/SDXC or not; this bit should be set to 1'b1.
- Bit 28 can be either 1 or 0.
- Bit 24 should be set to 1'b1, indicating that the host is capable of voltage switching; refer to following figure.



47	46	45-40	39	38	37	36-33	32	31-16	15-08	07-01	00
S	D	Index	Busy 31	CCS 30	Rsvd 29	Reserved 28-25	S18R 24	OCR 23-08	Reserved 07-00	CRC7	E
0	0	111111	X	X	0	0000	X	xxxxh	0000000	1111111	1

Busy Status
0b: On Initialization
1b: Initialization complete

Card Capacity Status
0b: SDSC
1b: SCHC or SCXC

S18R: Switching to 1.8V Accepted
0b: Continues current voltage signalling
1b: Ready for switching signal voltage

Fig. 3-14 ACMD41 Response(R3)

- Bit 30 – If set to 1'b1, card supports SDHC/SDXC; if set to 1'b0, card supports only SDSC
- Bit 24 – If set to 1'b1, card supports voltage switching and is ready for the switch
- Bit 31 – If set to 1'b1, initialization is over; if set to 1'b0, means initialization in process

4) If the card supports voltage switching, then the software must perform the steps discussed for either the "Voltage Switch Normal Scenario" or the "Voltage Switch Error Scenario".

19

2. Voltage Switch Normal Scenario

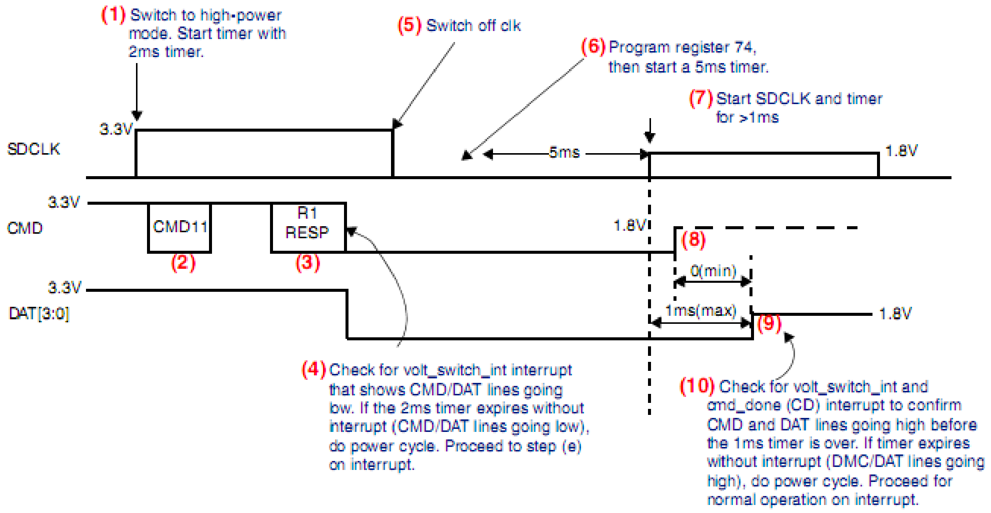

(1) Switch to high-power mode. Start timer with 2ms timer.
(5) Switch off clk
(6) Program register 74, then start a 5ms timer.
(7) Start SDCLK and timer for >1ms
3.3V
SDCLK
5ms
1.8V
3.3V
CMD
CMD11
(2)
R1 RESP
(3)
1.8V
(8)
0(min)
3.3V
DAT[3:0]
1ms(max)
(9)
1.8V
(4) Check for volt_switch_int interrupt that shows CMD/DAT lines going low. If the 2ms timer expires without interrupt (CMD/DAT lines going low), do power cycle. Proceed to step (e) on interrupt.
(10) Check for volt_switch_int and cmd_done (CD) interrupt to confirm CMD and DAT lines going high before the 1ms timer is over. If timer expires without interrupt (DMC/DAT lines going high), do power cycle. Proceed for normal operation on interrupt.

Fig. 3-15 Voltage Switch Normal Scenario

1) The host programs SDMMC_CLKENA—cclk_low_power register—with zero (0) for the corresponding card, which makes the host controller move to high-power mode. The application should start a timer with a recommended value of 2ms; this value of 2 ms is determined as below:Total clk required for CMD11 = 48 clks
 Total clk required for RESP R1 = 48 clks
 Maximum clk delay between MCD11 end to start of RESP1 = 60 clks
 Total = 48+48 + 60 = 160
 Minimum frequency during enumeration is 100 KHz; that is, 10us
 Total time = 160 * 10us = 1600us = 1. 6ms ~ 2ms
2) The host issues CMD11 to start the voltage switch sequence. Set bit 28 to 1'b1 in CMD when setting CMD11; for more information on setting bits, refer to "Boot Operation".
3) The card returns R1 response; the host controller does not generate cmd_done interrupt on receiving R1 response.
4) The card drives CMD and DAT [3:0] to low immediately after the response. The host controller generates interrupt (VOLT_SWITCH_INT) once the CMD or DAT [3:0] line goes low. The application should wait for this interrupt. If the 2ms timer expires without an interrupt (CMD/DAT lines going low), do a power cycle.

Note: Before doing a power cycle, switch off the card clock by programming SDMMC_CLKENA register

Proceed to step (5) on getting an interrupt (VOLT_SWITCH_INT).

Note: This interrupt must be cleared once this interrupt is received. Additionally, this interrupt should not be masked during the voltage switch sequence.

If the timer expires without interrupt (CMD/DAT lines going low), perform a power cycle. Proceed to step (5) on interrupt.

1) Program the SDMMC_CLKENA, cclk_enable register, with 0 for the corresponding card; the host stops supplying SDCLK.
2) Program Voltage register to the required values for the corresponding card. The application should start a timer > 5ms.
3) After the 5ms timer expires, the host voltage regulator is stable. Program SDMMC_CLKENA, cclk_enable register, with 1 for the corresponding card; the host starts providing SDCLK at 1. 8V; this can be at zero time after Voltage register has been programmed. When the SDMMC_CLKENA register is programmed, the application should start another timer > 1ms.
4) By detecting SDCLK, the card drives CMD to high at 1. 8V for at least one clock and then stops driving (tri-state); CMD is triggered by the rising edge of SDCLK (SDR timing).
5) If switching to 1. 8V signaling is completed successfully, the card drives DAT [3:0] to high at 1. 8V for at least one clock and then stops driving (tri-state); DAT [3:0] is triggered by the rising edge of SDCLK (SDR timing). DAT[3:0] must be high within 1ms from the start of SDCLK.
6) The host controller generates a voltage switch interrupt (VOLT_SWITCH_INT) and a command done (CD) interrupt once the CMD and DAT[3:0] lines go high. The application should wait for this interrupt to confirm CMD and DAT lines going high before the 1ms timer is done.

If the timer expires without the voltage switch interrupt (VOLT_SWITCH_INT), a power cycle should be performed. Program the SDMMC_CLKENA register to stop the clock for the corresponding card number. Wait for the cmd_done (CD) interrupt. Proceed for normal operation on interrupt. After the sequence is completed, the host and the card start communication in SDR12 timing.

3. Voltage Switch Error Scenario

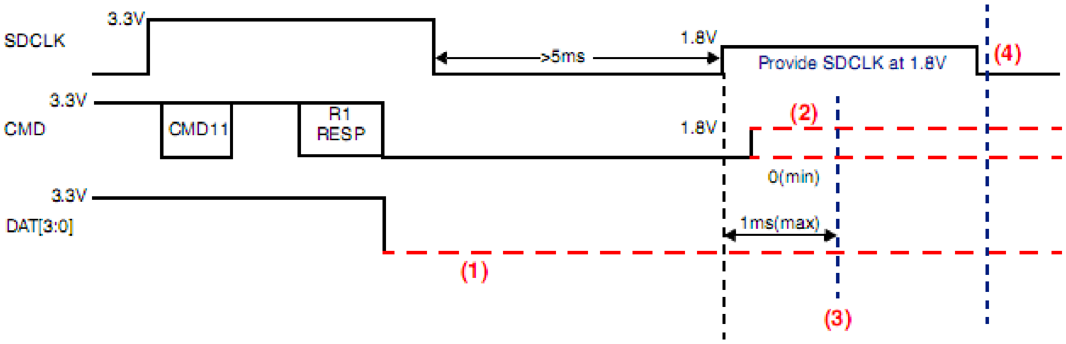


Fig. 3-16 Voltage Switch Error Scenario

1) If the interrupt (VOLT_SWITCH_INT) does not come, then the 2 ms timer should time out and a power cycle should be initiated.

Note: Before performing a power cycle, switch off the card clock by programming SDMMC_CLKENA register; no cmd_done (CD) interrupt is generated.

Additionally, if the card detects a voltage error at any point in between steps (5) and (7) in the card keeps driving DAT[3:0] to low until card power off.

2) CMD can be low or tri-state.
3) The host controller generates a voltage switch interrupt once the CMD and DAT[3:0] lines go high. The application should check for an interrupt to confirm CMD and DAT lines going high before the 1 ms timer is done.

If the 1 ms timer expires without interrupt (VOLT_SWITCH_INT) and cmd_done (CD), a power cycle should be performed. Program the SDMMC_CLKENA register to stop SDCLK of the corresponding card. Wait for the cmd_done interrupt. Proceed for normal operation on interrupt.

4) If DAT[3:0] is low, the host drives SDCLK to low and then stops supplying the card power.

Note: The card checks voltages of its own regulator output and host signals to ensure they are less than 2. 5V. Errors are indicated by (1) and (2).

- If voltage switching is accepted by the card, the default speed is SDR12.

- Command Done is given:
  - If voltage switching is properly done, CMD and DAT line goes high.
  - If switching is not complete, the 1ms timer expires, and the card clk is switched off.

Note: No other CMD should be driven before the voltage switching operation is completed and Command Done is received.

- The application should use CMD6 to check and select the particular function; the function appropriate-speed should be selected.

After the function switches, the application should program the correct value in the CLKDIV register, depending on the function chosen. Additionally, if Function 0x4 of the Access mode is chosen—that is, DDR50, then the application should also program 1'b1 in DDR_REG for the card number that has been selected for DDR50 mode.

3.6.6 Back-End Power

Each device needs one bit to control the back-end power supply for an embedded device; this bit does not control the VDDH of the host controller. A back_end_power register enables software programming for back-end power. The value on this register is output to the back_end_power signal, which can be used to switch power on and off the embedded device.

3.6.7 DDR Operation

1. 4-bit DDR Programming Sequence

DDR programming should be done only after the voltage switch operation has completed. The following outlines the steps for the DDR programming sequence:

1) Once the voltage switch operation is complete, the user must program voltage selection register to the required values for the corresponding card.

- To start a card to work in DDR mode, the application must program a bit of the newly defined SDMMC_UHS_REG[16] register with a value of 1'b1.
- The bit that the user programs depends on which card is to be accessed in DDR mode.

2) To move back to SDR mode, a power cycle should be run on the card—putting the card in SDR12 mode—and only then should SDMMC_UHS_REG[16]be set back to 1'b0 for the appropriate card.

20

2. 8-bit DDR Programming Sequence

The following outlines the steps for the 8-bit DDR programming sequence:

1) The cclk_in signal should be twice the speed of the required cclk_out. Thus, if the cclk_out signal is required to be 50 MHz, the cclk_in signal should be 100 MHz.
2) The CLKDIV register should always be programmed with a value higher than zero (0); that is, a clock divider should always be used for 8-bit DDR mode.
3) The application must program the SDMMC_UHS_REG[16] register (DDR_REG bits) by assigning it with a value of 1 for the bit corresponding to the card number; this causes the selected card to start working in DDR mode.
4) Depending on the card number, the SDMMC_CTYPE [31:16] bits should be set in order to make the host work in the 8-bit mode.

21

3. eMMC4.5 DDR START Bit

The eMMC4.5 changes the START bit definition in the following manner:

1) Receiver samples the START bit on the rising edge.
2) On the next rising edge after sampling the START bit, the receiver must sample the data.
3) Removes requirement of the START bit and END bit to be high for one full cycle.

Notes: The Host Controller does not support a START bit duration higher than one clock cycle. START bit durations of one or less than one clock cycle are supported and can be defined at the time of startup by programming the EMMC_DDR_REG register.

Following figure illustrates cases for the definition change of the START bit with eMMC4.5; it also illustrates how some of these cases can fail in sampling when higher-value delays are considered for I/O PADs.

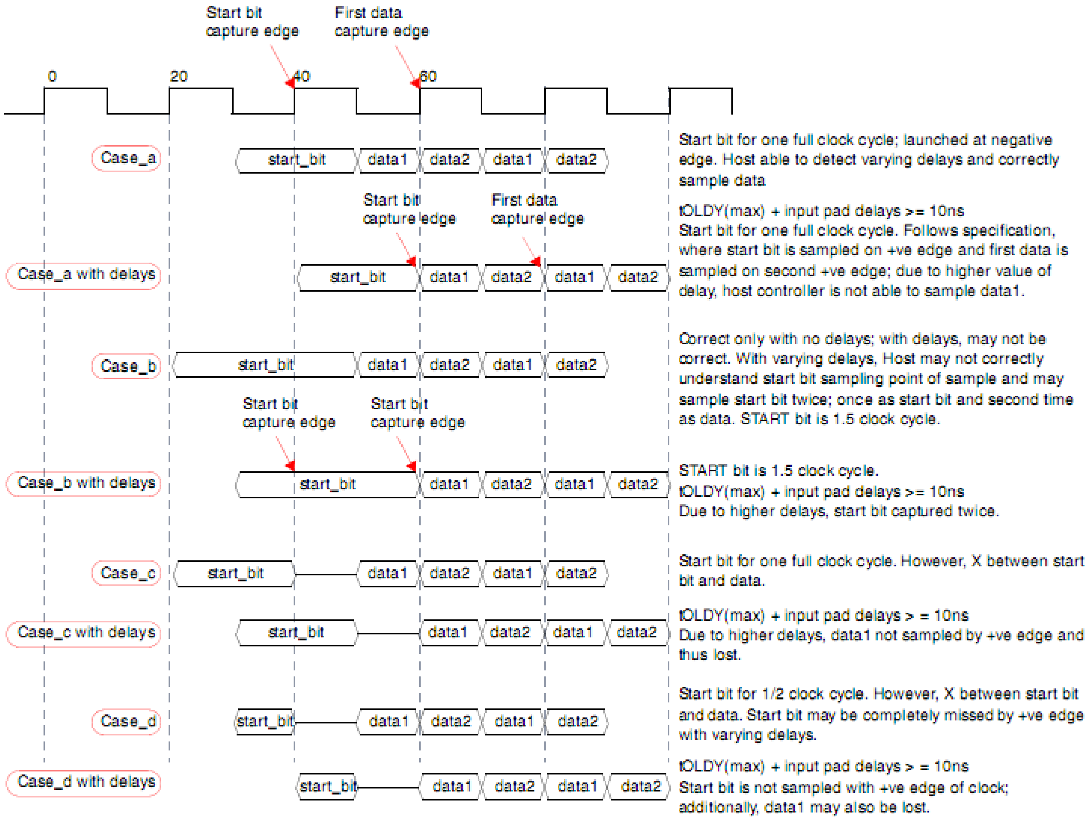

Start bit capture edge
First data capture edge
0
20
40
60
Case_a
start_bit
data1
data2
data1
data2
Start bit for one full clock cycle; launched at negative edge. Host able to detect varying delays and correctly sample data
Start bit capture edge
First data capture edge
Case_a with delays
tOLDY(max) + input pad delays >= 10ns
Start bit for one full clock cycle. Follows specification, where start bit is sampled on +ve edge and first data is sampled on second +ve edge; due to higher value of delay, host controller is not able to sample data1.
Case_b
Correct only with no delays; with delays, may not be correct. With varying delays, Host may not correctly understand start bit sampling point of sample and may sample start bit twice; once as start bit and second time as data. START bit is 1.5 clock cycle.
Start bit capture edge
Start bit capture edge
Case_b with delays
START bit is 1.5 clock cycle.
tOLDY(max) + input pad delays >= 10ns
Due to higher delays, start bit captured twice.
Case_c
Start bit for one full clock cycle. However, X between start bit and data.
Case_c with delays
tOLDY(max) + input pad delays > = 10ns
Due to higher delays, data1 not sampled by +ve edge and thus lost.
Case_d
Start bit for 1/2 clock cycle. However, X between start bit and data. Start bit may be completely missed by +ve edge with varying delays.
Case_d with delays
tOLDY(max) + input pad delays > = 10ns
Start bit is not sampled with +ve edge of clock; additionally, data1 may also be lost.

Fig. 3-17 CASES for eMMC 4.5 START bit

4. Reset Command/Moving from DDR50 to SDR12

To reset the mode of operation from DDR50 to SDR12, the following sequence of operations has to be done by the application:

1) Issue CMD0.

When CMD0 is received, the card changes from DDR50 to SDR12.

2) Program the SDMMC_CLKDIV register with an appropriate value.

3) Set DDR_REG to 0.

Note: The Voltage register should not be programmed to 0 while switching from DDR50 to SDR12, since the card is still operating in 1.8V mode after receiving CMD0.

3.6.8 H/W Reset Operation

When the RST_n signal goes low, the card enters a pre-idle state from any state other than the inactive state.

H/W Reset Programming Sequence

The following outlines the steps for the H/W reset programming sequence:

11) Program CMD12 to end any transfer in process.

12) Wait for DTO, even if no response is sent back by the card.

13) Set the following resets:

- DMA reset–SDMMC_CTRL[2]
- FIFO reset –SDMMC_CTRL[1] bits

Note: The above steps are required only if a transfer is in process.

14) Program the CARD_RESET register with a value of 0; this can be done at any time when the card is connected to the controller. This programming asserts the RST_n signal and resets the card.

15) Wait for minimum of 1 µs or cclk_in period, whichever is greater

16) After a minimum of 1 µs, the application should program a value of 0 into the CARD_RESET register. This de-asserts the RST_n signal and takes the card out of reset.

17) The application can program a new CMD only after a minimum of 200 µs after the de-assertion of the RST_n signal, as per the MMC 4.41 standard.

Note: For backward compatibility, the RST_n signal is temporarily disabled in the card by default. The host may need to set the signal as either permanently enabled or permanently disabled before it uses the card.

3.6.9 FBE Scenarios

An FBE occurs due to an AHB error response on the AHB bus. This is a system error, so the software driver should not perform any further programming to the Host. The only recovery mechanism from such scenarios is to do one of the following:

- Issue a hard reset by asserting the reset_n signal
- Do a program controller reset by writing to the CTRL[0] register

22

1. FIFO Overflow and Underflow

During normal data transfer conditions, FIFO overflow and underflow will not occur. However if there is a programming error, then FIFO overflow/underflow can result. For example, consider the following scenarios.

- For transmit: PBL=4, Tx watermark = 1. For the above programming values, if the FIFO has only one location empty, it issues a dma_req to IDMAC FSM. Due to PBL value=4, the IDMAC FSM performs 4 pushes into the FIFO. This will result in a FIFO overflow interrupt.
- For receive: PBL=4, Rx watermark = 1. For the above programming values, if the FIFO has only one location filled, it issues a dma_req to IDMAC FSM. Due to PBL value=4, the IDMAC FSM performs 4 pops to the FIFO. This will result in a FIFO underflow interrupt.

The driver should ensure that the number of bytes to be transferred as indicated in the descriptor should be a multiple of 4bytes with respect to H_DATA_WIDTH=32. For example, if the BYTCNT = 13, the number of bytes indicated in the descriptor should be 16 for H_DATA_WIDTH=32.

2. Programming of PBL and Watermark Levels

The DMAC performs data transfers depending on the programmed PBL and threshold values.

Table 3-15 PBL and Watermark Levels

PBL (Number of transfers)	Tx/Rx Watermark Value
1	greater than or equal to 1
4	greater than or equal to 4
8	greater than or equal to 8
16	greater than or equal to 16
32	greater than or equal to 32
64	greater than or equal to 64
128	greater than or equal to 128
256	greater than or equal to 256

3.6.10 Variable Delay/Clock Generation

Variable delay mechanism for the cclk_in_drv is optional, but it can be useful in order to meet a range of hold-time requirements across modes. Variable delay mechanism for the cclk_in_sample is mandatory and is required to achieve the correct sampling point for data.

cclk_in/cclk_in_sample/ cclk_in_drv is generated by Clock Generation Unit (CLKGEN) with variable delay mechanism, which includes Phase Shift Unit and Delay Line Unit selectable.

The Phase Shift Unit can shift cclk_in_sample/cclk_in_drv by 0/90/180/270-degree relative to cclk_in, controlled by sample_degree/drv_degree.

The Delay Line Unit can shift cclk_in_sample/cclk_in_drv in the unit of 40ps~80ps for every delay element. The delay unit number is determined by sample_delaynum/drv_delaynum, and enabled by sample_sel/drv_sel.

cclk_in is generated by cclkin divided by 2. cclk_in_drv and cclk_in_sample clocks are phase-shifted with delayed versions of cclk_in. All clocks are recommended to have a 50% duty cycle; DDR modes must have 50% duty cycles.

The architecture is as follows.

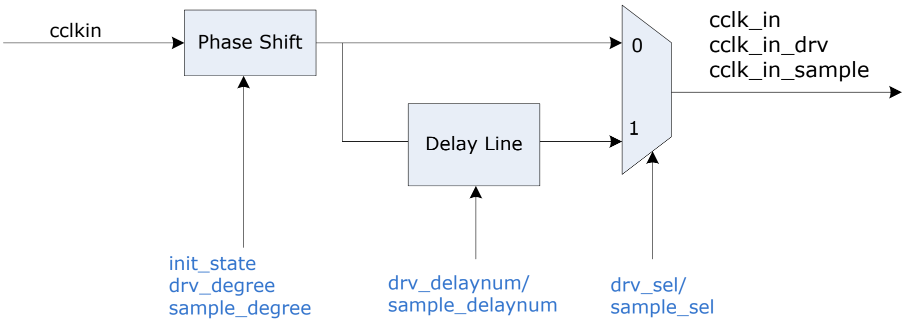


Fig. 3-18 Clock Generation Unit

The control signals for different Host Controller instance are shown as follows:

Table 3-16 Configuration for SDMMC Clock Generation

Signal Name	Source	Default	Description
init_state	CRU_SDMMC_CON0[0]	0	Soft initial state for phase shift.
drv_degree[1:0]	CRU_SDMMC_CON0[2:1]	2	Phase shift for cclk_in_drv. 0: 0-degree 1: 90-degree 2: 180-degree 3: 270-degree
drv_delaynum[7:0]	CRU_SDMMC_CON0[10:3]	0	Element number in delay line for cclk_in_drv
drv_sel	CRU_SDMMC_CON0[11]	0	cclk_in_drv source selection: 0: use clock after phase_shift 1: use clock after phase_shift and delay line
sample_degree[1:0]	CRU_SDMMC_CON1[2:1]	0	Phase shift for cclk_in_sample. 0: 0-degree 1: 90-degree 2: 180-degree 3: 270-degree
sample_delaynum[7:0]	CRU_SDMMC_CON1[10:3]	0	Element number in delay line for cclk_in_sample
sample_sel	CRU_SDMMC_CON1[11]	0	cclk_in_sample source selection: 0: use clock after phase_shift 1: use clock after phase_shift and delay line

Table 3-17 Configuration for SDIO Clock Generation

Signal Name	Source	Default	Description
init_state	CRU_SDIO_CON0[0]	0	Soft initial state for phase shift.
drv_degree[1:0]	CRU_SDIO_CON0[2:1]	2	Phase shift for cclk_in_drv. 0: 0-degree 1: 90-degree 2: 180-degree 3: 270-degree
drv_delaynum[7:0]	CRU_SDIO_CON0[10:3]	0	Element number in delay line for cclk_in_drv
drv_sel	CRU_SDIO_CON0[11]	0	cclk_in_drv source selection: 0: use clock after phase_shift 1: use clock after phase_shift and delay line
sample_degree[1:0]	CRU_SDIO_CON1[2:1]	0	Phase shift for cclk_in_sample. 0: 0-degree 1: 90-degree 2: 180-degree 3: 270-degree

Signal Name	Source	Default	Description
sample_delaynum[7:0]	CRU_SDIO_CON1[10:3]	0	Element number in delay line for cclk_in_sample
sample_sel	CRU_SDIO_CON1[11]	0	cclk_in_sample source selection: 0: use clock after phase_shift 1: use clock after phase_shift and delay line

Table 3-18 Configuration for EMMC Clock Generation

Signal Name	Source	Default	Description
init_state	CRU_EMMC_CON0[0]	0	Soft initial state for phase shift.
drv_degree[1:0]	CRU_EMMC_CON0[2:1]	2	Phase shift for cclk_in_drv. 0: 0-degree 1: 90-degree 2: 180-degree 3: 270-degree
drv_delaynum[7:0]	CRU_EMMC_CON0[10:3]	0	Element number in delay line for cclk_in_drv
drv_sel	CRU_EMMC_CON0[11]	0	cclk_in_drv source selection: 0: use clock after phase_shift 1: use clock after phase_shift and delay line
sample_degree[1:0]	CRU_EMMC_CON1[2:1]	0	Phase shift for cclk_in_sample. 0: 0-degree 1: 90-degree 2: 180-degree 3: 270-degree
sample_delaynum[7:0]	CRU_EMMC_CON1[10:3]	0	Element number in delay line for cclk_in_sample
sample_sel	CRU_EMMC_CON1[11]	0	cclk_in_sample source selection: 0: use clock after phase_shift 1: use clock after phase_shift and delay line

The following outlines the steps for clock generation sequence:

1) Assert init_state to soft reset the CLKGEN.
2) Configure drv_degree/sample_degree.
3) If fine adjustment required, delay line can be used by configuring drv_delaynum/sample_delaynum and drv_sel/sample_sel.
4) Dis-assert init_state to start CLKGEN.

3.6.11 Variable Delay Tuning

Tuning is defined by SD and MMC cards to determine the correct sampling point required for the host, especially for the speed modes SDR104 and HS200 where the output delays from the cards can be up to 2 UI. Tuning is required for other speed modes-such as DDR50-even though the output delay from the card is less than one cycle.

Command for tuning is different for different cards.

- SD Memory Card:
  - CMD19 – SD card for SDR50 and SDR104 speed modes. Tuning data is defined by card specifications.
  - CMD6 – SD card for speed modes not supporting CMD19. Tuning data is the 64byte SD status.
- Multimedia Card:
  - CMD21 – MMC card for HS200 speed mode. Tuning data is defined by card specifications.
  - CMD8 – MMC card for speed modes not supporting CMD21. Tuning data is 512 byte ExtCSD data.

The following is the procedure for variable delay tuning:

1) Set a phase shift of 0-degree on cclk_in_sample.
2) Send the Tuning command to the card; the card in turn sends an R1 response on the CMD line and tuning data on the DAT line.
3) If the host sees any of the errors—start bit error, data crc error, end bit error, data read time-out, response crc error, response error—then the sampling point is incorrect.
4) Send CMD12 to bring the host controller state machines to idle.

- The card may treat CMD12 as an invalid command because the card has successfully sent the tuning data, and it cannot send a response.
- The host controller may generate a response time-out interrupt that must be cleared by software.

5) Repeat steps 2) to 4) by increasing the phase shift value or delay element number on cclk_in_sample until the correct sampling point is received such that the host does not see any of the errors.
6) Mark this phase shift value as the starting point of the sampling window.
7) Repeat steps 2 to 4 by increasing the phase shift value or delay element number on cclk_in_sample until the host sees the errors starting to come again or the phase shift value reaches 360-degree.
8) Mark the last successful phase shift value as the ending point of the sampling window.

A window is established where the tuning block is matched. For example, for a scenario where the tuning block is received correctly for a phase shift window of 90-degree and 180-degree, then an appropriate sampling point is established as 135-degree. Once a sampling point is established, no errors should be visible in the tuning block.

3.6.12 Package Command

In order to reduce overhead, read and write commands can be packed in groups of commands-either all read or all write-that transfer the data for all commands in the group in one transfer on the bus.

Packed commands can be of two types:

- Packed Write: CMD23 → CMD25
- Packed Read: CMD23 → CMD25 → CMD23 → CMD18

Packed commands are put in packets by the application software and are transparent to the core. For more information on packed commands, refer to the eMMC specification.

3.6.13 Card Detection Method

There are many methods for SDMMC/SDIO device detection.

- Method1: Using SDMMC_CDETECT register, which is value on card_detect_n input port. 0 represents presence of card.
- Method2: Using card detection unit in Host Controller, outputting host interrupt. The card detection unit looks for any changes in the card-detect signals for card insertion or card removal. It filters out the debounces associated with mechanical insertion or removal, and generates one interrupt to the host. You can program the debounce filter value in SDMMC_DEBNCE [23:0]. Following figure illustrates the timing for card-detect signals.

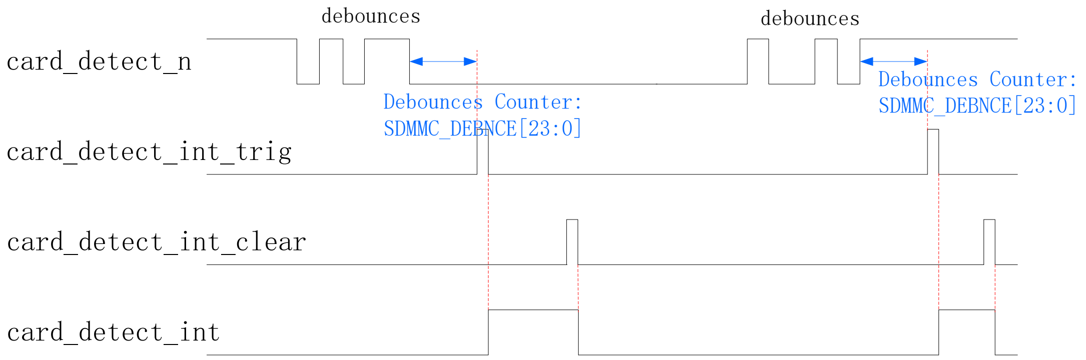


Fig. 3-19 Card Detection Method 2

- Method3: Using card detection unit in DETEC_GRF, outputting sdmmc_detect_dual_edge_int connecting to IRQ[112]. Similar to Method2, except that the debounce time is configurable by DETECT_GRF_SDMMC_DET_COUNTER; and the insertion/removal detection interrupt can be enabled or cleared respectively. The detailed register information is:

Table 3-19 Register for SDMMC Card Detection Method 3

Signal Name	Source	Default	Description
sd_detectn_rise_edge_irq_en	DETECT_GRF_SDMMC_DETECT_CON[0]	0	sdmmc detect_n signal rise edge interrupt enable. 1: enable 0: disable
sd_detect_fall_edge_detect_en	DETECT_GRF_SDMMC_DETECT_CON[1]	0	sd_detect_falling_edge enable 0: disable 1: enable
sd_detect_rising_edge_dectect_status	DETECT_GRF_SDMMC_DETECT_STATUS[0]	0	sd_detect_rising_edge status 0: disable 1: enable
sd_detect_fall_edge_detect_status	DETECT_GRF_SDMMC_DETECT_STATUS[1]	0	sd_detect_falling_edge status 0: disable 1: enable
sd_detect_rising_edge_dectect_clr	DETECT_GRF_SDMMC_DETECT_CLR[0]	0	sd_detect_rising_edge clear 0: disable 1: enable
sd_detect_fall_edge_detect_clr	DETECT_GRF_SDMMC_DETECT_CLR[1]	0	sd_detect_falling_edge clear 0: disable 1: enable

- Method4: Using filtered card_detect_n with the debounce time DETECT_GRF_SDMMC_DET_COUNTER for interrupt source, connecting to IRQ [111] directly.

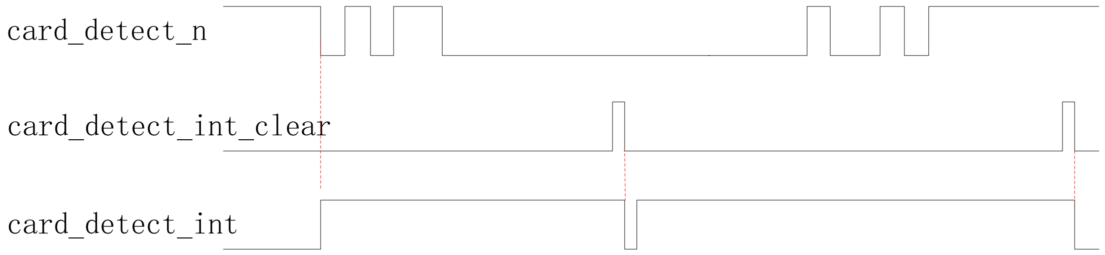


Fig. 3-20 Card Detection Method 4

Chapter 4 Process-Voltage-Temperature Monitor (PVTM)

4.1 Overview

The Process-Voltage-Temperature Monitor (PVTM) is used to monitor the chip performance variance caused by chip process, voltage and temperature.
PVTM supports the following features:

- A clock oscillation ring is integrated and used to generate a clock like signal, the frequency of this clock is determined by the cell delay value of clock oscillation ring circuit.
- A frequency counter is used to measure the frequency of the clock oscillation ring.
- Follow PVTM blocks are supported:
  - core_pvtm, used near Cortex-A35
  - pmu_pvtm, used near PMU

4.2 Block Diagram

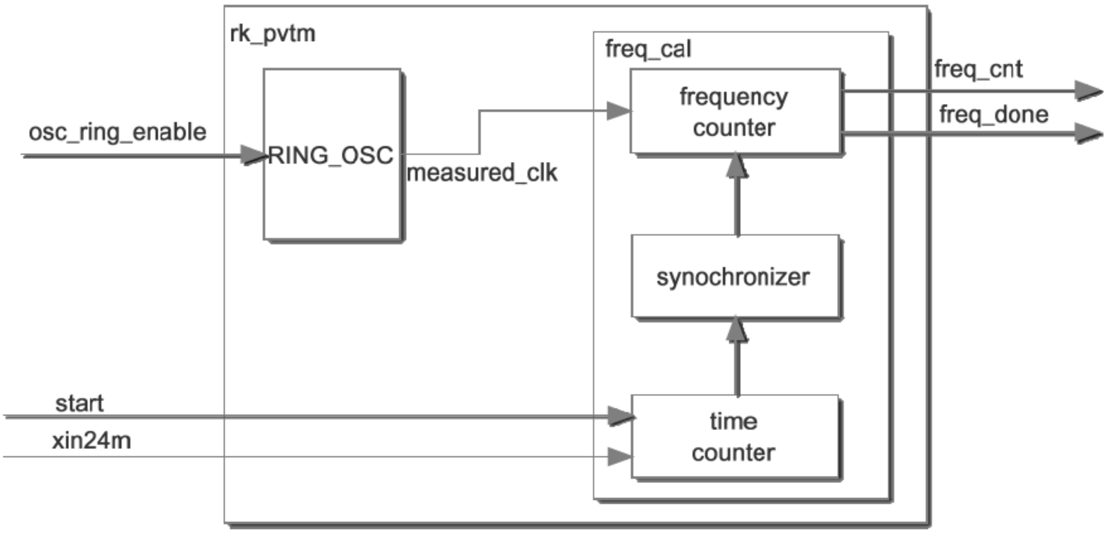


Fig. 4-1 PVTM Block Diagram

The PVTM include two main blocks:

- RING_OSC, it is composed with inverters with odd number, which is used to generate a clock. core_pvtm support 3clock oscillation rings in RING_OSC, and finally select a clock output by signal osc_ring_sel.
- Freq_cal, it is used to measure the frequency of clock which generated from the RING_SOC block.

4.3 Function Description

4.3.1 Frequency Calculation

A clock is generated by the RING_OSC, and a frequency fixed clock (24MHz) is used to calculate the cycles of the clock. Suppose the time period is 1s, then the clock period of RING_OSC clock is T= 1/clock_counter(s), the cell delay value is T/2.

4.3.2 Control Source and Result Destination

The pvtm is controlled by CRU and GRF, and the monitor result is got by GRF. Following tables shows the PVTM control source and result destination.

Copyright 2018 © FuZhou Rockchip Electronics Co., Ltd.

Table 4-1 core_pvtm control source and result destination

Interface	Reset value	Control Source/Result Destination
xin24m	0x0	CRU_CLKGATE0_CON[4], clock gating control
resetn	0x1	CRU_SOFTRST1_CON[1], reverse connect to resetn, active high
start	0x0	CORE_GRF_COREPVTM_CON0[0], active high
osc_ring_enable	0x0	CORE_GRF_COREPVTM_CON0[1], active high
osc_ring_sel	0x0	CORE_GRF_COREPVTM_CON0[3:2]
cal_cnt	0x0	CORE_GRF_COREPVTM_CON1[31:0]
freq_done	0x0	CORE_GRF_COREPVTM_STATUS0[0]
freq_cnt	0x0	CORE_GRF_COREPVTM_STATUS1[31:0]

Table 4-2 pmu_pvtm control source and result destination

Interface	Reset value	Control Source/Result Destination
xin24m	0x0	CRU_CLKGATE4_CON[4], clock gating control
resetn	0x1	CRU_SOFTRST5_CON[14], reverse connect to resetn, active high
start	0x1	GRF_PVTM_CON0[0], active high
osc_ring_enable	0x1	GRF_PVTM_CON0[1], active high
cal_cnt	0x0	GRF_PVTM_CON1[31:0]
freq_done	0x0	GRF_PVTM_STATUS0[0]
freq_cnt	0x0	GRF_PVTM_STATUS1[31:0]

4.3.3 pmu_pvtm usage

A clock divided from pmu_pvtm oscillation ring is used in low power mode, which can replace the function of 32KHz clock source by configure CRU_CLKSEL_CON2[9:8]. The division factor is configured by GRF_PVTM_CON0[11:2].

4.4 Application Notes

4.4.1 PVTM Usage Flow

1. Enable the frequency fixed clock xin24m.
2. Reset the pvtm.
3. Set osc_ring_enable '1' to enable the generated clock.
4. Set osc_ring_sel to select the clock oscillation ring (Only core_pvtm need)
5. Configure the cal_cnt to an appropriate value.
6. Set start '1' to calculate the cycles of the generated clock.
7. Wait the freq_done is asserted, then get the value of freq_cnt. The period of RING_OSC clock is T= cal_cnt*(Period of 24MHz clock)/freq_cnt, the cell delay value is T/2.

Chapter 5 USB2.0 Host

5.1 Overview

USB2.0 host controller supports fully USB2.0 functions with one EHCI host controller and one OHCI host controller, and each host controller has one USB port. OHCI host controller only supports full-speed and low-speed mode and is used for full-speed devices and low-speed devices. EHCI only supports high-speed mode and is used for high-speed devices. OHCI host controller and EHCI host controller shares the same USB port, EHCI host controller will auto select the owner (OHCI or EHCI) of this USB port depending on the speed mode of attached devices, when selecting OHCI as owner, OHCI host controller will serve for the attached device; when selecting EHCI as owner, EHCI host controller will serve for the attached device.
USB2.0 Host Controller supports the following features:

- Compatible Specifications
  - Universal Serial Bus Specification, Revision 2.0
  - Enhanced Host Controller Interface Specification (EHCI), Revision 1.0
  - Open Host Controller Interface Specification (OHCI), Revision 1.0a
- Support High-speed (480Mbps), Full-speed (12Mbps) and Low-speed (1.5Mbps)

5.2 Block Diagram

USB2.0 Host Controller comprises with:

- EHCI Host Controller: Perform High-speed transactions
- OHCI Host Controller: Perform full/low-speed transactions
- Port Routing Control: Select EHCI Host Controller or OHCI Host Controller

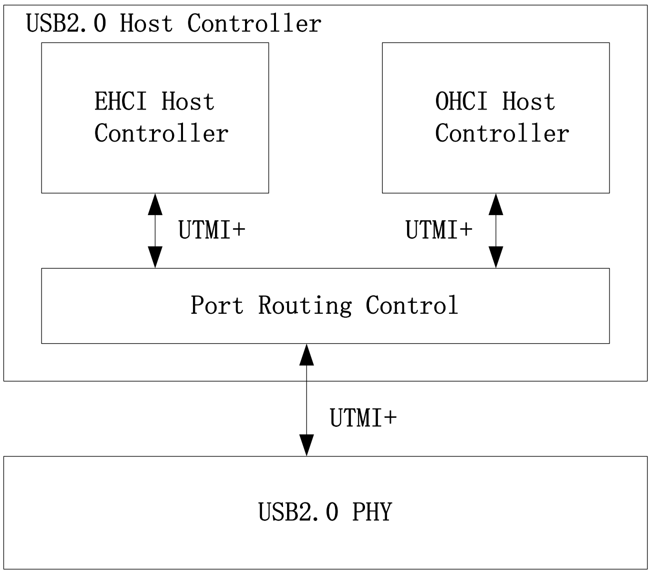


Fig. 5-1 USB2.0 Host Controller Block Diagram

5.3 Function Description

5.3.1 EHCI Host Controller

It performs descriptors and data read or write from or to system memory and packs or un-pack USB transactions from or to UTMI+ interface defined in EHCI specification for high-speed data transmission.

5.3.2 OHCI Host Controller

It performs descriptors and data read/write from/to system memory and packs or un-pack USB transactions from or to UTMI+ interface defined in OHCI specification for full-speed or low-speed data transmission.

5.3.3 Port Routing Control

As part of logic in the EHCI host controller, it is used to auto-select EHCI or OHCI host controller to serve the attached device depending on the speed of the attached device.

5.4 Register Description

5.4.1 Internal Address Mapping

Slave address can be divided into different length for different usage, which is shown as follows.

Table 5-1 USB2.0 Host Controller Address Mapping

Base Address	Device	Address Length	Offset Address Range
0xff440000	EHCI	64K BYTE	0x0000 ~ 0xffff
0xff450000	OHCI	64K BYTE	0x0000 ~ 0xffff
0xff000000	GRF	32K BYTE	0x0000~0x7fff
0xff008000	USBPHY_GRF	12K BYTE	0x0000~0x2fff
0xff00b000	DETECT_GRF	4K BYTE	0x0000~0x0fff

General Register File for USB2.0 host controller is spread into three modules which have continuous address. Please refer to “Chapter GRF” for detailed information about GRF, USBPHY_GRF and DETECT_GRF.

5.4.2 Register Summary and Detail Description

EHCI and OHCI register definitions, please refer to Enhanced Host Controller Interface Specification (EHCI), Revision 1.0 and Open Host Controller Interface Specification (OHCI), Revision 1.0a.

This USB2.0 host controller shares a combo PHY with USB OTG, and the host controller uses the port 1 of the combo PHY. The PHY has a register configuration interface with total width of 384 bits, and bit[191:0] are related to OTG while bit[383:192] are related to host controller. For host controller, these 192 configuration bits are programmed in USBPHY_GRF. Please refer to “Chapter GRF” for detailed bit assignment for these 192 bits. Detailed function description of these configuration bits are as follows.

Table 5-2 USB2.0 Host Controller PHY Configuration Interface Description

Bit	Attr	Reset Value	Description
383	RW	0x0	Single ended disconnect detection enable.
382:379	RW	0x0	Digital debug interface, keeping the default value is greatly appreciated. These bits will never be used by users normally.
378:376	RW	0x0	COMP mode selection.
375:373	RW	0x0	Disconnect bias current tuning bits, more ones will promote the disconnect working bandwidth.
372:320	RW	0x0	Digital debug interface, keeping the default value is greatly appreciated. These bits will never be used by users normally.
319	RW	0x0	vbus voltage level detection function power down, active high.

Bit	Attr	Reset Value	Description

318:
315
	RW	0xD	HOST disconnect detection trigger point configure, only used in HOST mode:
			4b'0000: 625mv;
			4b'0001: 675mv;
			4b'0010: 612.5mv;
			4b'0011: 575mv;
			4b'0100: 550mv;
			4b'0101: 600mv;
			4b'0110: 537.5mv;
			4b'0111: 500mv;
			4b'1000: 600mv;
			4b'1001: 650mv;
			4b'1010: 587.5mv;
			4b'1011: 550mv;
			4b'1100: 575mv;
			4b'1101: 625mv(default);
			4b'1110: 562.5mv;
			4b'1111: 525mv.

314: 312	RW	0x0	Digital debug interface, keeping the default value is greatly appreciated. These bits will never be used by users normally.
311	RW	0x0	Bypass squelch trigger point auto configure in chirp modes, active high.
310	RW	0x0	Half bit pre-emphasize enable, active high. "1" represent half bit pre-emphasis, "0" for full bit.
309	RW	0x0	Digital debug interface, reserved.
308	RW	0x0	PLL divider ration option, keeping the default value is greatly appreciated.
307	RW	0x0	HS slew rate tuning bits. Please refer to bit[196:195] for more information.
306: 304	RW	0x0	vbus_valid reference tuning.
303: 301	RW	0x0	session_end reference tuning.
300: 298	RW	0x0	B_sessionvalid reference tuning.
297: 295	RW	0x0	A_sessionvalid reference tuning.
294	RW	0x0	Force output vbus_valid asserted, active high.
293	RW	0x0	Force output session_end asserted, active high.
292	RW	0x0	Force output B_sessionvalid asserted, active high.
291	RW	0x0	Force output A_sessionvalid asserted, active high.
290	RW	0x1	Turn off differential receiver in suspend mode to save power, active low.

Bit	Attr	Reset Value	Description
289; 258	RW	0x51555555	Digital debug interface, keeping the default value is greatly appreciated. These bits will never be used by users normally.
257	RW	0x1	Digital debug interface.
256: 255	RW	0x0	Digital debug interface, keeping the default value is greatly appreciated. These bits will never be used by users normally.
254	RW	0x0	Digital debug interface.
253: 250	RW	0x0	Digital debug interface, keeping the default value is greatly appreciated. These bits will never be used by users normally.
249	RW	0x1	A port ODT auto refresh bypass, active low, this register should only be used when bit[235:234] were set to "11". In bypass mode, customer can configure driver strength through bit[233:229].
248	RW	0x0	BG output voltage reference adjust, keeping the default value is greatly appreciated.

247:
245
	RW	0x0	Compensation current tuning reference:
			3'b000: 400mV(default);
			3'b001: 362.5mV;
			3'b010: 375mV;
			3'b011: 387.5mV;
			3'b100: 412.5mV;
			3'b101: 425mV;
			3'b110: 437.5mV;
			3'b111: 450mV.

244:
242
	RW	0x0	Bias current tuning reference:
			3'b000: 200mV(default);
			3'b001: 212.5mV;
			3'b010: 225mV;
			3'b011: 237.5mV;
			3'b100: 250mV;
			3'b101: 187.5mV;
			3'b110: 175mV;
			3'b111: 162.5mV.

241:
239
	RW	0x0	ODT compensation voltage reference:
			3'b000: 268mV(default);
			3'b001: 262mV;
			3'b010: 250mV;
			3'b011: 237.5mV;
			3'b100: 275mV;
			3'b101: 281mV;
			3'b110: 293mV;
			3'b111: 300mV.

238: 237	RW	0x0	Battery charging related registers, keeping the default value is greatly appreciated.
236	RO	0x0	Reserved.

Bit	Attr	Reset Value	Description
235: 234	RW	0x0	Auto compensation bypass, "11" will bypass current and ODT compensation, customers can set the driver strength and current manually. For larger HS/FS/LS slew rate, give more "1" for bit[233:229].
233: 229	RW	0x17	HS/FS driver strength tuning, "11111" represent the largest slew rate and "10000" represents the smallest slew rate.
228: 221	RW	0x3f	HS eye height tuning bits. More zeros represent bigger eye.
220: 219	RW	0x0	Digital debug interface, keeping the default value is greatly appreciated. These bits will never be used by users normally.
218: 211	RW	0x0	Digital debug interface, keeping the default value is greatly appreciated. These bits will never be used by users normally.
210	RW	0x1	Enable current compensation, active high.
209	RW	0x1	Enable resistance compensation, active high.

208:
205
	RW	0xc	port squelch trigger point configuration:
			4b'0000: 112.5mv;
			4b'0001: 150mv;
			4b'0010: 87.5mv;
			4b'0011: 162.5mv;
			4b'0100: 100mv;
			4b'0101: 137.5mv;
			4b'0110: 75mv;
			4b'0111: 150mv;
			4b'1000: 125mv;
			4b'1001: 162.5mv;
			4b'1010: 100mv;
			4b'1011: 175mv;
			4b'1100: 150mv(default);
			4b'1101: 187.5mv;
			4b'1110: 125mv;
			4b'1111: 200mv.

204: 203	RW	0x0	Registers for non-driving state control. non-driving state is controlled by op-mode by default, when bit[203] is configured with "1", user can control non-driving state trough bit[204].
202: 200	RW	0x5	USB Tx Clock phase configure, 3'b000 represent the earliest phase, and 3'b111 the latest, single step delay is 256ps.
199: 197	RW	0x0	USB Rx Clock phase configure, 3'b000 represent the earliest phase, and 3'b111 the latest, single step delay is 256ps.
196: 195	RW	0x3	combine with bit[307]. {bit[307],bit[196:195]} form three HS slew rate tuning bits. More one represents larger slew rate, 111 the maximum and 001 the minimum. 000 will shut down the high speed driver output.

Bit	Attr	Reset Value	Description
194: 192	RW	0x0	HS eye diagram adjust, open HS pre-emphasize function to increase HS slew rate, only used when large cap loading is attached. 3'b001: open pre-emphasize in sof or eop state; 3'b010: open pre-emphasize in chirp state; 3'b100: open pre-emphasize in non-chirp state; 3'b111: always open pre-emphasize; other combinations: reserved.
191:0	-	-	These 192 bits are related to OTG. Please refer to "Chapter USB OTG".

5.5 Interface Description

Table 5-3 USB2.0 PHY Interface Description

Module Pin	Direction	Pin Name	Descriptions
USB0ID	I	USB_ID	USB2.0 PHY OTG Port ID, left unused for TypeC
USB0PN	I/O	USB0_DM	USB2.0 PHY OTG Port PN
USB0PP	I/O	USB0_DP	USB2.0 PHY OTG Port PP
VBUS	I	USB_VBUS	USB2.0 PHY OTG Port VBUS
USB1PN	I/O	USB1_DM	USB2.0 PHY Host Port PN
USB1PP	I/O	USB1_DP	USB2.0 PHY Host Port PP
USBRBIAS	I/O	USB_EXTR	USB2.0 PHY Shared RBIAS

Notes: I=input, O=output, I/O=input/output, bidirectional

5.6 Application Notes

5.6.1 Special Setting

5.6.2 Program flow

Please refer to Enhanced Host Controller Interface Specification (EHCI), Revision 1.0 and Open Host Controller Interface Specification (OHCI), Revision 1.0a.

Chapter 6 USB OTG2.0

6.1 Overview

USB OTG 2.0 is a Dual-Role Device controller, which supports both device and host functions and is fully compliant with OTG Supplement to USB2.0 specification and support high-speed (480Mbps), full-speed (12Mbps), low-speed (1.5Mbps) transfer.
USB OTG 2.0 is optimized for portable electronic devices, point-to-point applications (no hub, direct connection to device) and multi-point applications to devices.

6.1.1 Features

- Compliant with the OTG Supplement to the USB2.0 Specification
- Operates in High-Speed, Full-Speed and Low-Speed mode (host mode only)
- Support 9 channels in host mode
- 9 Device mode endpoints in addition to control endpoint 0, 4 in, 3 out and 2 IN/OUT
- Built-in one 1024x35 bits FIFO
- Internal DMA with scatter/gather function
- Supports packet-based, dynamic FIFO memory allocation for endpoints for flexible, efficient use of RAM
- Support dynamic FIFO sizing

6.2 Block Diagram

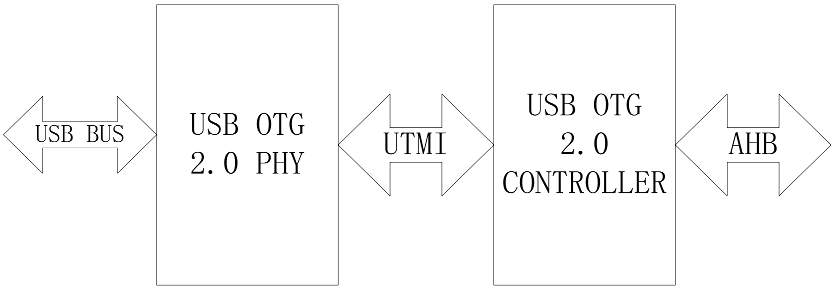


Fig.6-1 USB OTG 2.0 Architecture

The Fig shows the architecture of USB OTG 2.0. It is broken up into two separate units: USB OTG 2.0 controller and USB OTG 2.0 PHY. The two units are interconnected with UTMI interface.

6.3 USB OTG 2.0 Controller

The USB OTG 2.0 Controller can active as USB2.0 host controller or USB2.0 device controller basing on ID status of Micron-AB receptacle.
As USB2.0 host controller, it uses one transmit FIFO for all non-periodic out transactions and one transmit FIFO for all periodic out transactions as transmit buffers to hold the data to be transmitted over USB, and use one receive FIFO for all periodic and non-periodic transactions to hold the received data from the USB until it is transferred to the system memory by DMA.
As USB2.0 device controller, it also uses a single receive FIFO to receive the data for all the out endpoints from USB, and individual transmit FIFOs for each IN endpoint.

6.4 USB OTG 2.0 PHY

The USB OTG 2.0 PHY handles the low-level USB protocol and signaling from controller and differential pairs. It includes functions such as data serialization and de-serialization, bit stuffing and clock recovery and synchronization. Its feature contains:

- provide dual UTMI ports
- OTG0 Support UART Bypass Function
- Fully compliant with USB specifications Rev 2.0
- Supports 480Mbps (HS), 12Mbps (FS) & 1.5Mbps(LS) serial data transmission

Copyright 2018 © FuZhou Rockchip Electronics Co., Ltd.

- Supports low latency hub mode with 40 bit time around trip delay
- 8-bit or 16-bit UTMI interface compliant with UTMI+ specification level 3 Rev 1.
- Loop back BIST mode supported
- Built-in I/O and ESD structure
- On-die self-calibrated HS/FS/LS termination
- 24MHz crystal oscillator with integrated phase-locked loop (PLL) oscillator
- Dual 3.3V / 1.2V supply

6.5 UART BYPASS FUNCITON

UART bypass mode is used for software team debugging by transmitting UART signals over USB differential pairs. When in UART bypass mode, UART2 can transmit UART protocol signals over USB differential pairs and USB is not functional; Otherwise, UART2 use normal UART interface and USB is functional.

6.6 Register Description

6.6.1 Internal Address Mapping

Slave address can be divided into different length for different usage, which is shown as follows.

Table 6-1 USB OTG Address Mapping

Base Address	Device	Address Length	Offset Address Range
0xff400000	OTG	256K BYTE	0x00000 ~ 0x3ffff
0xff000000	GRF	32K BYTE	0x0000~0x7fff
0xff008000	USBPHY_GRF	12K BYTE	0x0000~0x2fff
0xff00b000	DETECT_GRF	4K BYTE	0x0000~0x0fff

General Register File for USB OTG is spread into three modules which have continuous address. Please refer to “Chapter GRF” for detailed information about GRF, USBPHY_GRF and DETECT_GRF.

6.6.2 Register Summary

Name	Offset	Size	Reset Value	Description
USBOTG_GOTGCTL	0x0000	W	0x00000000	Control and Status Register
USBOTG_GOTGINT	0x0004	W	0x00000000	Interrupt Register
USBOTG_GAHBCFG	0x0008	W	0x00000000	AHB Configuration Register
USBOTG_GUSBCFG	0x000c	W	0x00001400	USB Configuration Register
USBOTG_GRSTCTL	0x0010	W	0x80000000	Reset Register
USBOTG_GINTSTS	0x0014	W	0x00000000	Interrupt Register
USBOTG_GINTMSK	0x0018	W	0x00000000	Interrupt Mask Register
USBOTG_GRXSTSR	0x001c	W	0x00000000	Receive Status Debug Read Register
USBOTG_GRXSTSP	0x0020	W	0x00000000	Receive Status Read and Pop Register
USBOTG_GRXFSIZ	0x0024	W	0x00000000	Receive FIFO Size Register
USBOTG_GNPTXFSIZ	0x0028	W	0x00000000	Non-Periodic Transmit FIFO Size Register
USBOTG_GNPTXSTS	0x002c	W	0x00000000	Non-Periodic Transmit FIFO/Queue Status Register

Name	Offset	Size	Reset Value	Description
USBOTG_GI2CCTL	0x0030	W	0x11000000	I2C Address Register
USBOTG_GPVNDCTL	0x0034	W	0x00000000	PHY Vendor Control Register
USBOTG_GGPIO	0x0038	W	0x00000000	General Purpose Input/Output Register
USBOTG_GUID	0x003c	W	0x00000000	User ID Register
USBOTG_GSNPSID	0x0040	W	0x00004f54	Core ID Register
USBOTG_GHWCFG1	0x0044	W	0x00000000	User HW Config1 Register
USBOTG_GHWCFG2	0x0048	W	0x00000000	User HW Config2 Register
USBOTG_GHWCFG3	0x004c	W	0x00000000	User HW Config3 Register
USBOTG_GHWCFG4	0x0050	W	0x00000000	User HW Config4 Register
USBOTG_GLPMCFG	0x0054	W	0x00000000	Core LPM Configuration Register
USBOTG_GPWRDN	0x0058	W	0x00000000	Global Power Down Register
USBOTG_GDFIFOCFG	0x005c	W	0x00000000	Global DFIFO Software Config Register
USBOTG_GADPCTL	0x0060	W	0x00000000	ADP Timer, Control and Status Register
USBOTG_HPTXFSIZ	0x0100	W	0x00000000	Host Periodic Transmit FIFO Size Register
USBOTG_DIEPTXFn	0x0104	W	0x00000000	Device Periodic Transmit FIFO-n Size Register
USBOTG_HCFG	0x0400	W	0x00000000	Host Configuration Register
USBOTG_HFIR	0x0404	W	0x00000000	Host Frame Interval Register
USBOTG_HFNUM	0x0408	W	0x0000ffff	Host Frame Number/Frame Time Remaining Register
USBOTG_HPTXSTS	0x0410	W	0x00000000	Host Periodic Transmit FIFO/Queue Status Register
USBOTG_HAINT	0x0414	W	0x00000000	Host All Channels Interrupt Register
USBOTG_HAINTMSK	0x0418	W	0x00000000	Host All Channels Interrupt Mask Register
USBOTG_HPRT	0x0440	W	0x00000000	Host Port Control and Status Register
USBOTG_HCCHARn	0x0500	W	0x00000000	Host Channel-n Characteristics Register
USBOTG_HCSPLTn	0x0504	W	0x00000000	Host Channel-n Split Control Register
USBOTG_HCINTn	0x0508	W	0x00000000	Host Channel-n Interrupt Register
USBOTG_HCINTMSKn	0x050c	W	0x00000000	Host Channel-n Interrupt Mask Register
USBOTG_HCTSIZn	0x0510	W	0x00000000	Host Channel-n Transfer Size Register
USBOTG_HCDMAn	0x0514	W	0x00000000	Host Channel-n DMA Address Register

Name	Offset	Size	Reset Value	Description
USBOTG_HCDMABn	0x051c	W	0x00000000	Host Channel-n DMA Buffer Address Register
USBOTG_DCFG	0x0800	W	0x08200000	Device Configuration Register
USBOTG_DCTL	0x0804	W	0x00002000	Device Control Register
USBOTG_DSTS	0x0808	W	0x00000000	Device Status Register
USBOTG_DIEPMSK	0x0810	W	0x00000000	Device IN Endpoint common interrupt mask register
USBOTG_DOEPMSK	0x0814	W	0x00000000	Device OUT Endpoint common interrupt mask register
USBOTG_DAINT	0x0818	W	0x00000000	Device All Endpoints interrupt register
USBOTG_DAINTMSK	0x081c	W	0x00000000	Device All Endpoint interrupt mask register
USBOTG_DTKNQR1	0x0820	W	0x00000000	Device IN token sequence learning queue read register1
USBOTG_DTKNQR2	0x0824	W	0x00000000	Device IN token sequence learning queue read register2
USBOTG_DVBUSDIS	0x0828	W	0x00000b8f	Device VBUS discharge time register
USBOTG_DVBUSPULSE	0x082c	W	0x00000000	Device VBUS Pulsing Timer Register
USBOTG_DTHRCTL	0x0830	W	0x08100020	Device Threshold Control Register
USBOTG_DIEPEMPMSK	0x0834	W	0x00000000	Device IN endpoint FIFO empty interrupt mask register
USBOTG_DEACHINT	0x0838	W	0x00000000	Device each endpoint interrupt register
USBOTG_DEACHINTMSK	0x083c	W	0x00000000	Device each endpoint interrupt register mask
USBOTG_DIEPEACHMSKn	0x0840	W	0x00000000	Device each IN endpoint -n interrupt Register
USBOTG_DOEPEACHMSKn	0x0880	W	0x00000000	Device each out endpoint-n interrupt register
USBOTG_DIEPCTL0	0x0900	W	0x00008000	Device control IN endpoint 0 control register
USBOTG_DIEPINTn	0x0908	W	0x00000000	Device Endpoint-n Interrupt Register
USBOTG_DIEPTSIZn	0x0910	W	0x00000000	Device endpoint n transfer size register
USBOTG_DIEPDMAn	0x0914	W	0x00000000	Device endpoint-n DMA address register
USBOTG_DTXFSTSn	0x0918	W	0x00000000	Device IN endpoint transmit FIFO status register

Name	Offset	Size	Reset Value	Description
USBOTG_DIEPDMABn	0x091c	W	0x00000000	Device endpoint-n DMA buffer address register
USBOTG_DIEPCTLn	0x0920	W	0x00000000	Device endpoint-n control register
USBOTG_DOEPCTL0	0x0b00	W	0x00000000	Device control OUT endpoint 0 control register
USBOTG_DOEPINTn	0x0b08	W	0x00000000	Device endpoint-n control register
USBOTG_DOEPTSIZn	0x0b10	W	0x00000000	Device endpoint n transfer size register
USBOTG_DOEPDMAn	0x0b14	W	0x00000000	Device Endpoint-n DMA Address Register
USBOTG_DOEPDMABn	0x0b1c	W	0x00000000	Device endpoint-n DMA buffer address register
USBOTG_DOEPCTLn	0x0b20	W	0x00000000	Device endpoint-n control register
USBOTG_PCGCR	0x0e00	W	0x200b8000	Power and clock gating control register
USBOTG_EPBUF0	0x1000	W	0x00000000	Device endpoint 0 / host out channel 0 address
USBOTG_EPBUF1	0x2000	W	0x00000000	Device endpoint 1 / host out channel 1 address
USBOTG_EPBUF2	0x3000	W	0x00000000	Device endpoint 2 / host out channel 2 address
USBOTG_EPBUF3	0x4000	W	0x00000000	Device endpoint 3 / host out channel 3 address
USBOTG_EPBUF4	0x5000	W	0x00000000	Device endpoint 4 / host out channel 4 address
USBOTG_EPBUF5	0x6000	W	0x00000000	Device endpoint 5 / host out channel 5 address
USBOTG_EPBUF6	0x7000	W	0x00000000	Device endpoint 6 / host out channel 6 address
USBOTG_EPBUF7	0x8000	W	0x00000000	Device endpoint 7 / host out channel 7 address

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

6.6.3 Detail Register Description

USBOTG_GOTGCTL

Address: Operational Base + offset (0x0000)

Control and Status Register

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved

Bit	Attr	Reset Value	Description
27	RW	0x0	ChirpEn Chirp on enable This bit when programmed to 1'b1 results in the core asserting chirp_on before sending an actual Chirp "K" signal on USB. This bit is present only if OTG_BC_SUPPORT = 1. If OTG_BC_SUPPORT != 1, this bit is a reserved bit.
26:22	RO	0x00	MultValidBc Multi Valued ID pin Battery Charger ACA inputs in the following order: Bit 26 - rid_float Bit 25 - rid_gnd Bit 24 - rid_a Bit 23 - rid_b Bit 22 - rid_c These bits are present only if OTG_BC_SUPPORT = 1. Otherwise, these bits are reserved and will read 5'h0.
21	RO	0x0	reserved
20	RW	0x0	OTGVer OTG version Indicates the OTG revision. 0: OTG Version 1.3. In this version the core supports Data line pulsing and VBus pulsing for SRP. 1: OTG Version 2.0. In this version the core supports only Data line pulsing for SRP.
19	RO	0x0	BSesVld B-session valid Indicates the Device mode transceiver status. 0: B-session is not valid. 1: B-session is valid. In OTG mode, you can use this bit to determine if the device is connected or disconnected. Note: If you do not enabled OTG features (such as SRP and HNP), the read reset value will be 1.The vbus assigns the values internally for non-SRP or non-HNP configurations.
18	RO	0x0	ASesVld A-session valid Indicates the Host mode transceiver status. 0: A-session is not valid 1: A-session is valid Note: If you do not enabled OTG features (such as SRP and HNP), the read reset value will be 1.The vbus assigns the values internally for non-SRP or non-HNP configurations.

Bit	Attr	Reset Value	Description
17	RO	0x0	DbnTime Long/short debounce time Indicates the debounce time of a detected connection. 0: Long debounce time, used for physical connections (100 ms + 2.5 us) 1: Short debounce time, used for soft connections (2.5 us)
16	RO	0x0	ConIDSts Connector ID Status Indicates the connector ID status on a connect event. 0: The core is in A-Device mode 1: The core is in B-Device mode
15:12	RO	0x0	reserved
11	RW	0x0	DevHNPEn Device HNP Enable The application sets this bit when it successfully receives a SetFeature. SetHNPEnable command from the connected USB host. 0: HNP is not enabled in the application 1: HNP is enabled in the application
10	RW	0x0	HstSetHNPEn Host set HNP enable The application sets this bit when it has successfully enabled HNP (using the SetFeature.SetHNPEnable command) on the connected device. 0: Host Set HNP is not enabled 1: Host Set HNP is enabled
9	RW	0x0	HNPReq HNP request The application sets this bit to initiate an HNP request to the connected USB host. The application can clear this bit by writing a 0 when the Host Negotiation Success Status Change bit in the OTG Interrupt register (GOTGINT.HstNegSucStsChng) is set. The core clears this bit when the HstNegSucStsChng bit is cleared. 0: No HNP request 1: HNP request
8	RO	0x0	HstNegScs Host Negotiation Success The core sets this bit when host negotiation is successful. The core clears this bit when the HNP Request (HNPReq) bit in this register is set. 0: Host negotiation failure 1: Host negotiation success
7:2	RO	0x0	reserved

Bit	Attr	Reset Value	Description
1	RW	0x0	SesReq Session Request The application sets this bit to initiate a session request on the USB. The application can clear this bit by writing a 0 when the Host Negotiation Success Status Change bit in the OTG Interrupt register (GOTGINT.HstNegSucStsChng) is set. The core clears this bit when the HstNegSucStsChng bit is cleared. If you use the USB 1.1 Full-Speed Serial Transceiver interface to initiate the session request, the application must wait until the VBUS discharges to 0.2 V, after the B-Session Valid bit in this register (GOTGCTL.BSesVld) is cleared. This discharge time varies between different PHYs and can be obtained from the PHY vendor. 0: No session request 1: Session request
0	RO	0x0	SesReqScs Session Request Success The core sets this bit when a session request initiation is successful. 0: Session request failure 1: Session request success

USBOTG_GOTGINT
Address: Operational Base + offset (0x0004)
Interrupt Register

Bit	Attr	Reset Value	Description
31:21	RO	0x0	reserved
20	W1C	0x0	MultiValueChg Multi-Valued input changed This bit when set indicates that there is a change in the value of at least one ACA pin value. This bit is present only if OTG_BC_SUPPORT = 1, otherwise it is reserved.
19	W1C	0x0	DbnceDone Debounce Done The core sets this bit when the debounce is completed after the device connected. The application can start driving USB reset after seeing this interrupt. This bit is only valid when the HNP Capable or SRP Capable bit is set in the Core USB Configuration register (GUSBCFG.HNPCap or GUSBCFG.SRPCap, respectively).
18	W1C	0x0	ADevTOUTChg A-Device Timeout Change The core sets this bit to indicate that the A-device has timed out while waiting for the B-device to connect.

Bit	Attr	Reset Value	Description
17	W1C	0x0	HstNegDet Host Negotiation Detected The core sets this bit when it detects a host negotiation request on the USB
16:10	RO	0x0	reserved
9	W1C	0x0	HstNegSucStsChng Host Negotiation Success Status Change The core sets this bit on the success or failure of a USB host negotiation request. The application must read the Host Negotiation Success bit of the OTG Control and Status register (GOTGCTL.HstNegScs) to check for success or failure
8	W1C	0x0	SesReqSucStsChng Session Request Success Status Change The core sets this bit on the success or failure of a session request. The application must read the Session Request Success bit in the OTG Control and Status register (GOTGCTL.SesReqScs) to check for success or failure.
7:3	RO	0x0	reserved
2	W1C	0x0	SesEndDet Session End Detected The core sets this bit when the utmisrp_bvalid signal is deasserted
1:0	RO	0x0	reserved

USBOTG_GAHBCFG
Address: Operational Base + offset (0x0008)
AHB Configuration Register

Bit	Attr	Reset Value	Description
31:23	RO	0x0	reserved

Bit	Attr	Reset Value	Description
22	RW	0x0	NotiAllDmaWrit Notify All Dma Write Transactions This bit is programmed to enable the System DMA Done functionality for all the DMA write Transactions corresponding to the Channel/Endpoint. This bit is valid only when GAHBCFG.RemMemSupp is set to 1. GAHBCFG.NotiAllDmaWrit = 1. HSOTG core asserts int_dma_req for all the DMA write transactions on the AHB interface along with int_dma_done, chep_last_transact and chep_number signal informations. The core waits for sys_dma_done signal for all the DMA write transactions in order to complete the transfer of a particular Channel/Endpoint. GAHBCFG.NotiAllDmaWrit = 0. HSOTG core asserts int_dma_req signal only for the last transaction of DMA write transfer corresponding to a particular Channel/Endpoint. Similarly, the core waits for sys_dma_done signal only for that transaction of DMA write to complete the transfer of a particular Channel/Endpoint.
21	RW	0x0	RemMemSupp Remote Memory Support This bit is programmed to enable the functionality to wait for the system DMA Done Signal for the DMA Write Transfers. GAHBCFG.RemMemSupp=1. The int_dma_req output signal is asserted when HSOTG DMA starts write transfer to the external memory. When the core is done with the Transfers it asserts int_dma_done signal to flag the completion of DMA writes from HSOTG. The core then waits for sys_dma_done signal from the system to proceed further and complete the Data Transfer corresponding to a particular Channel/Endpoint. GAHBCFG.RemMemSupp=0. The int_dma_req and int_dma_done signals are not asserted and the core proceeds with the assertion of the XferComp interrupt as soon as the DMA write transfer is done at the HSOTG Core Boundary and it does not wait for the sys_dma_done signal to complete the DATA transfers.
20:9	RO	0x0	reserved

Bit	Attr	Reset Value	Description
8	RW	0x0	PTxFEmpLvl Periodic TxFIFO Empty Level Indicates when the Periodic TxFIFO Empty Interrupt bit in the Core Interrupt register (GINTSTS.PTxFEmp) is triggered. This bit is used only in Slave mode. 0: GINTSTS.PTxFEmp interrupt indicates that the Periodic TxFIFO is half empty 1: GINTSTS.PTxFEmp interrupt indicates that the Periodic TxFIFO is completely empty
7	RW	0x0	NPTxFEmpLvl Non-Periodic TxFIFO Empty Level This bit is used only in Slave mode. In host mode and with Shared FIFO with device mode, this bit indicates when the Non-Periodic TxFIFO Empty Interrupt bit in the Core Interrupt register GINTSTS.NPTxFEmp) is triggered. With dedicated FIFO in device mode, this bit indicates when IN endpoint Transmit FIFO empty interrupt (DIEPINTn.TxFEmp) is triggered. Host mode and with Shared FIFO with device mode: 1'b0: GINTSTS.NPTxFEmp interrupt indicates that the Non-Periodic TxFIFO is half empty 1'b1: GINTSTS.NPTxFEmp interrupt indicates that the Non-Periodic TxFIFO is completely empty Dedicated FIFO in device mode: 1'b0: DIEPINTn.TxFEmp interrupt indicates that the IN Endpoint TxFIFO is half empty 1'b1: DIEPINTn.TxFEmp interrupt indicates that the IN Endpoint TxFIFO is completely empty
6	RO	0x0	reserved
5	RW	0x0	DMAEn DMA Enable 0: Core operates in Slave mode 1: Core operates in a DMA mode This bit is always 0 when Slave-Only mode has been selected.

Bit	Attr	Reset Value	Description
4:1	RW	0x0	HBstLen Burst Length/Type This field is used in both External and Internal DMA modes. In External DMA mode, these bits appear on dma_burst[3:0] ports, External DMA Mode defines the DMA burst length in terms of 32-bit words: 4'b0000: 1 word 4'b0001: 4 words 4'b0010: 8 words 4'b0011: 16 words 4'b0100: 32 words 4'b0101: 64 words 4'b0110: 128 words 4'b0111: 256 words Others: Reserved Internal DMA Mode AHB Master burst type: 4'b0000: Single 4'b0001: INCR 4'b0011: INCR4 4'b0101: INCR8 4'b0111: INCR16 Others: Reserved
0	RW	0x0	GlblIntrMsk Global Interrupt Mask The application uses this bit to mask or unmask the interrupt line assertion by itself. Irrespective of this bit's setting, the interrupt status registers are updated by the core. 1'b0: Mask the interrupt assertion to the application. 1'b1: Unmask the interrupt assertion to the application.

USBOTG_GUSBCFG
Address: Operational Base + offset (0x000c)
USB Configuration Register

Bit	Attr	Reset Value	Description
31	RW	0x0	CorruptTxpacket Corrupt Tx packet This bit is for debug purposes only. Never set this bit to 1.

Bit	Attr	Reset Value	Description
30	RW	0x0	ForceDevMode Force Device Mode Writing a 1 to this bit forces the core to device mode irrespective of utmiotg_iddig input pin. 1'b0: Normal Mode 1'b1: Force Device Mode After setting the force bit, the application must wait at least 25 ms before the change to take effect. When the simulation is in scale down mode, waiting for 500 us is sufficient. This bit is valid only when OTG_MODE = 0, 1 or 2. In all other cases, this bit reads 0.
29	RW	0x0	ForceHstMode Force Host Mode Writing a 1 to this bit forces the core to host mode irrespective of utmiotg_iddig input pin. 1'b0: Normal Mode 1'b1: Force Host Mode After setting the force bit, the application must wait at least 25 ms before the change to take effect. When the simulation is in scale down mode, waiting for 500 us is sufficient. This bit is valid only when OTG_MODE =0, 1 or 2. In all other cases, this bit reads 0.
28	RW	0x0	TxEndDelay Tx End Delay Writing a 1 to this bit enables the TxEndDelay timers in the core. 1'b0: Normal mode 1'b1: Introduce Tx end delay timers
27	RW	0x0	IC_USBTrafCtl IC_USB TrafficPullRemove Control When this bit is set, pullup/pulldown resistors are detached from the USB during traffic signaling. This bit is valid only when configuration parameter OTG_ENABLE_IC_USB = 1 and register field GUSBCFG.IC_USBCap is set to 1.
26	RW	0x0	IC_USBCap IC_USB-Capable The application uses this bit to control the IC_USB capabilities. 1'b0: IC_USB PHY Interface is not selected. 1'b1: IC_USB PHY Interface is selected. This bit is writable only if OTG_ENABLE_IC_USB=1 and OTG_FSPHY_INTERFACE!=0.The reset value depends on the configuration parameter OTG_SELECT_IC_USB when OTG_ENABLE_IC_USB = 1. In all other cases, this bit is set to 1'b0 and the bit is read only.

Bit	Attr	Reset Value	Description
25	RW	0x0	ULPIIfDis ULPI Interface Protect Disable Controls circuitry built into the PHY for protecting the ULPI interface when the link tri-states STP and data. Any pull-ups or pull-downs employed by this feature can be disabled. Please refer to the ULPI Specification for more detail. 1'b0: Enables the interface protect circuit 1'b1: Disables the interface protect circuit
24	RW	0x0	IndPassThrough Indicator Pass Through Controls whether the Complement Output is qualified with the Internal Vbus Valid comparator before being used in the Vbus State in the RX CMD. Please refer to the ULPI Specification for more detail. 1'b0: Complement Output signal is qualified with the Internal VbusValid comparator. 1'b1: Complement Output signal is not qualified with the Internal VbusValid comparator.
23	RW	0x0	IndComple Indicator Complement Controls the PHY to invert the ExternalVbusIndicator input signal,generating the Complement Output. Please refer to the ULPI Specification for more detail 1'b0: PHY does not invert ExternalVbusIndicator signal 1'b1: PHY does invert ExternalVbusIndicator signal
22	RW	0x0	TermSelDLPulse TermSel DLine Pulsing Selection This bit selects utmi_termselect to drive data line pulse during SRP. 1'b0: Data line pulsing using utmi_txvalid (default). 1'b1: Data line pulsing using utmi_termsel.
21	RW	0x0	ULPIExtVbusIndicator ULPI External VBUS Indicator This bit indicates to the ULPI PHY to use an external VBUS over-current indicator. 1'b0: PHY uses internal VBUS valid comparator. 1'b1: PHY uses external VBUS valid comparator. (Valid only when RTL parameter OTG_HSPHY_INTERFACE = 2 or 3)

Bit	Attr	Reset Value	Description
20	RW	0x0	ULPIExtVbusDrv ULPI External VBUS Drive This bit selects between internal or external supply to drive 5V on VBUS, in ULPI PHY. 1'b0: PHY drives VBUS using internal charge pump (default). 1'b1: PHY drives VBUS using external supply. (Valid only when RTL parameter OTG_HSPHY_INTERFACE = 2 or 3)
19	RW	0x0	ULPIClkSusM ULPI Clock SuspendM This bit sets the ClockSuspendM bit in the Interface Control register on the ULPI PHY. This bit applies only in serial or carkit modes. 1'b0: PHY powers down internal clock during suspend. 1'b1: PHY does not power down internal clock. (Valid only when RTL parameter OTG_HSPHY_INTERFACE = 2 or 3)
18	RW	0x0	ULPIAutoRes ULPI Auto Resume This bit sets the AutoResume bit in the Interface Control register on the ULPI PHY. 1'b0: PHY does not use AutoResume feature. 1'b1: PHY uses AutoResume feature. (Valid only when RTL parameter OTG_HSPHY_INTERFACE = 2 or 3)
17	RW	0x0	ULPIFsLs ULPI FS/LS Select The application uses this bit to select the FS/LS serial interface for the ULPI PHY. This bit is valid only when the FS serial transceiver is selected on the ULPI PHY. 1'b0: ULPI interface 1'b1: ULPI FS/LS serial interface (Valid only when RTL parameters OTG_HSPHY_INTERFACE = 2 or 3 and OTG_FSPHY_INTERFACE = 1, 2, or 3)
16	RW	0x0	OtgI2CSel UTMIFS or I2C Interface Select The application uses this bit to select the I2C interface. 1'b0: UTMI USB 1.1 Full-Speed interface for OTG signals 1'b1: I2C interface for OTG signals This bit is writable only if I2C and UTMIFS were specified for Enable I2C Interface (parameter OTG_I2C_INTERFACE = 2). Otherwise, reads return 0.

Bit	Attr	Reset Value	Description
15	RW	0x0	PhyLPwrClkSel PHY Low-Power Clock Select Selects either 480-MHz or 48-MHz (low-power) PHY mode. In FS and LS modes, the PHY can usually operate on a 48-MHz clock to save power. 1'b0: 480-MHz Internal PLL clock 1'b1: 48-MHz External Clock In 480 MHz mode, the UTMI interface operates at either 60 or 30-MHz, depending upon whether 8- or 16-bit data width is selected. In 48-MHz mode, the UTMI interface operates at 48 MHz in FS and LS modes. This bit drives the utmi_fsls_low_power core output signal, and is valid only for UTMI+ PHYs.
14	RO	0x0	reserved
13:10	RW	0x5	USBTrdTim USB Turnaround Time Sets the turnaround time in PHY clocks. Specifies the response time for a MAC request to the Packet FIFO Controller (PFC) to fetch data from the DFIF(SPRAM). This must be programmed to 4'h5: When the MAC interface is 16-bit UTMI+. 4'h9: When the MAC interface is 8-bit UTMI+. Note: The values above are calculated for the minimum AHB frequency of 30MHz. USB turnaround time is critical for certification where long cables and 5-Hubs are used, so if you need the AHB to run at less than 30 MHz, and if USB turnaround time is not critical, these bits can be programmed to a larger value.
9	RW	0x0	HNPCap HNP-Capable The application uses this bit to control the HNP capabilities. 0: HNP capability is not enabled. 1: HNP capability is enabled. This bit is writable only if an HNP mode was specified for Mode of Operation (parameter OTG_MODE). Otherwise, reads return 0.
8	RW	0x0	SRPCap SRP-Capable The application uses this bit to control the SRP capabilities. If the core operates as a non-SRP-capable B-device, it cannot request the connected A-device (host) to activate VBUS and start a session. 0: SRP capability is not enabled. 1: SRP capability is enabled. This bit is writable only if an SRP mode was specified for Mode of Operation in coreConsultant (parameter OTG_MODE). Otherwise, reads return 0.

Bit	Attr	Reset Value	Description
7	RW	0x0	DDRSel ULPI DDR Select The application uses this bit to select a Single Data Rate (SDR) or Double Data Rate (DDR) or ULPI interface. 0: Single Data Rate ULPI Interface, with 8-bit-wide data bus 1: Double Data Rate ULPI Interface, with 4-bit-wide data bus
6	RW	0x0	PHYSel USB 2.0 High-Speed PHY or USB 1.1 Full-Speed Serial Transcei The application uses this bit to select either a high-speed UTMI+ or ULPI PHY, or a full-speed transceiver. 0: USB 2.0 high-speed UTMI+ or ULPI PHY 1: USB 1.1 full-speed serial transceiver If a USB 1.1 Full-Speed Serial Transceiver interface was not selected (parameter OTG_FSPHY_INTERFACE = 0), this bit is always 0, with Write Only access. If a high-speed PHY interface was not selected (parameter OTG_HSPHY_INTERFACE = 0), this bit is always 1, with Write Only access. If both interface types were selected in coreConsultant (parameters have non-zero values), the application uses this bit to select which interface is active, and access is Read and Write.
5	RW	0x0	FSIntf Full-Speed Serial Interface Select The application uses this bit to select either a unidirectional or bidirectional USB 1.1 full-speed serial transceiver interface. 0: 6-pin unidirectional full-speed serial interface 1: 3-pin bidirectional full-speed serial interface If a USB 1.1 Full-Speed Serial Transceiver interface was not selected (parameter OTG_FSPHY_INTERFACE = 0), this bit is always 0, with Write Only access. If a USB 1.1 FS interface was selected (parameter OTG_FSPHY_INTERFACE! = 0), then the application can set this bit to select between the 3- and 6-pin interfaces, and access is Read and Write.
4	RW	0x0	ULPI_UTMI_Sel ULPI or UTMI+ Select The application uses this bit to select either a UTMI+ interface or ULPI Interface. 0: UTMI+ Interface 1: ULPI Interface This bit is writable only if UTMI+ and ULPI was specified for High-Speed PHY Interface(s) in coreConsultant configuration (parameter OTG_HSPHY_INTERFACE = 3). Otherwise, reads return either 0 or 1, depending on the interface selected using the OTG_HSPHY_INTERFACE parameter.

Bit	Attr	Reset Value	Description
3	RW	0x0	PHYIf PHY Interface The application uses this bit to configure the core to support a UTMI+ PHY with an 8- or 16-bit interface. When a ULPI PHY is chosen, this must be set to 8-bit mode. 0: 8 bits 1: 16 bits This bit is writable only if UTMI+ and ULPI were selected (parameter OTG_HSPHY_DWIDTH = 3). Otherwise, this bit returns the value for the power-on interface selected during configuration.
2:0	RW	0x0	TOutCal HS/FS Timeout Calibration The number of PHY clocks that the application programs in this field is added to the high-speed/full-speed inter-packet timeout duration in the core to account for any additional delays introduced by the PHY. This can be required, because the delay introduced by the PHY in generating the line state condition can vary from one PHY to another. The USB standard timeout value for high-speed operation is 736 to 816 (inclusive) bit times. The USB standard timeout value for full-speed operation is 16 to 18 (inclusive) bit times. The application must program this field based on the speed of enumeration. The number of bit times added per PHY clock are: High-speed operation: One 30-MHz PHY clock = 16 bit times One 60-MHz PHY clock = 8 bit times Full-speed operation: One 30-MHz PHY clock = 0.4 bit times One 60-MHz PHY clock = 0.2 bit times One 48-MHz PHY clock = 0.25 bit times

USBOTG_GRSTCTL
Address: Operational Base + offset (0x0010)
Reset Register

Bit	Attr	Reset Value	Description
31	RO	0x1	AHBIdle AHB Master Idle Indicates that the AHB Master State Machine is in the IDLE condition.
30	RO	0x0	DMAReq DMA Request Signal Indicates that the DMA request is in progress. Used for debug.
29:11	RO	0x0	reserved

Bit	Attr	Reset Value	Description
10:6	RW	0x00	TxFNum TxFIFO Number This is the FIFO number that must be flushed using the TxFIFO Flush bit. This field must not be changed until the core clears the TxFIFO Flush bit. 5'h0: Non-periodic TxFIFO flush in Host mode; Non-periodic TxFIFO flush in device mode when in shared FIFO operation. Tx FIFO 0 flush in device mode when in dedicated FIFO mode. 5'h1: Periodic TxFIFO flush in Host mode: Periodic TxFIFO 1 flush in Device mode when in shared FIFO operation; TXFIFO 1 flush in device mode when in dedicated FIFO mode. 5'h2: Periodic TxFIFO 2 flush in Device mode when in shared FIFO operation: TXFIFO 2 flush in device mode when in dedicated FIFO mode. ... 5'hF: Periodic TxFIFO 15 flush in Device mode when in shared FIFO operation: TXFIFO 15 flush in device mode when in dedicated FIFO mode. 5'h10: Flush all the transmit FIFOs in device or host mode.
5	R/W SC	0x0	TxFFlsh TxFIFO Flush This bit selectively flushes a single or all transmit FIFOs, but cannot do so if the core is in the midst of a transaction. The application must write this bit only after checking that the core is neither writing to the TxFIFO nor reading from the TxFIFO. Verify using these registers: Read NAK Effective Interrupt ensures the core is not reading from the FIFO. Write GRSTCTL.AHBIdle ensures the core is not writing anything to the FIFO. Flushing is normally recommended when FIFOs are re-configured or when switching between Shared FIFO and Dedicated Transmit FIFO operation. FIFO flushing is also recommended during device endpoint disable. The application must wait until the core clears this bit before performing any operations. This bit takes eight clocks to clear, using the slower clock of phy_clk or hclk.
4	R/W SC	0x0	RxFFlsh RxFIFO Flush The application can flush the entire RxFIFO using this bit, but must first ensure that the core is not in the middle of a transaction. The application must only write to this bit after checking that the core is neither reading from the RxFIFO nor writing to the RxFIFO. The application must wait until the bit is cleared before performing any other operations. This bit requires 8 clocks (slowest of PHY or AHB clock) to clear.

Bit	Attr	Reset Value	Description
3	R/W SC	0x0	INTknQFlsh IN Token Sequence Learning Queue Flush This bit is valid only if OTG_EN_DED_TX_FIFO = 0. The application writes this bit to flush the IN Token Sequence Learning Queue.
2	W1 C	0x0	FrmCntrRst Host Frame Counter Reset The application writes this bit to reset the (micro)frame number counter inside the core. When the (micro)frame counter is reset, the subsequent SOF sent out by the core has a (micro)frame number of 0.
1	R/W SC	0x0	Reset A write to this bit issues a soft reset to the otg_power_dn module of the core.

Bit	Attr	Reset Value	Description
0	R/W SC	0x0	CSftRst Core Soft Reset Resets the hclk and phy_clock domains as follows: Clears the interrupts and all the CSR registers except the following register bits: CGCCTL.RstPdwnModule PCGCCTL.GateHclk PCGCCTL.PwrClmp PCGCCTL.StopPPhyLPwrClkSelclk GUSBCFG.PhyLPwrClkSel GUSBCFG.DDRSel GUSBCFG.PHYSel GUSBCFG.FSIntf GUSBCFG.ULPI_UTMI_Sel GUSBCFG.PHYIf HCFG.FSLSPclkSel DCFG.DevSpd GGPIO GPWRDN GADPCTL All module state machines (except the AHB Slave Unit) are reset to the IDLE state, and all the transmit FIFOs and the receive FIFO are flushed. Any transactions on the AHB Master are terminated as soon as possible, after gracefully completing the last data phase of an AHB transfer. Any transactions on the USB are terminated immediately. When Hibernation or ADP feature is enabled, the PMU module is not reset by the Core Soft Reset. The application can write to this bit any time it wants to reset the core. This is a self-clearing bit and the core clears this bit after all the necessary logic is reset in the core, which can take several clocks, depending on the current state of the core. Once this bit is cleared software must wait at least 3 PHY clocks before doing any access to the PHY domain (synchronization delay). Software must also must check that bit 31 of this register is 1 (AHB Master is IDLE) before starting any operation. Typically software reset is used during software development and also when you dynamically change the PHY selection bits in the USB configuration registers listed above. When you change the PHY, the corresponding clock for the PHY is selected and used in the PHY domain. Once a new clock is selected, the PHY domain has to be reset for proper operation.

USBOTG_GINTSTS
Address: Operational Base + offset (0x0014)
Interrupt Register

Bit	Attr	Reset Value	Description
31	W1C	0x0	WkUpInt Resume/Remote Wakeup Detected Interrupt Wakeup Interrupt during Suspend(L2) or LPM(L1) state. During Suspend(L2): Device Mode: This interrupt is asserted only when Host Initiated Resume is detected on USB. Host Mode: This interrupt is asserted only when Device Initiated Remote Wakeup is detected on USB. During LPM(L1): Device Mode: This interrupt is asserted for either Host Initiated Resume or Device Initiated Remote Wakeup on USB. Host Mode: This interrupt is asserted for either Host Initiated Resume or Device Initiated Remote Wakeup on USB.
30	W1C	0x0	SessReqInt Session Request/New Session Detected Interrupt In Host mode, this interrupt is asserted when a session request is detected from the device. In Host mode, this interrupt is asserted when a session request is detected from the device. In Device mode, this interrupt is asserted when the utmisrp_bvalid signal goes high.
29	W1C	0x0	DisconnInt Disconnect Detected Interrupt This interrupt is asserted when a device disconnect is detected.
28	W1C	0x0	ConIDStsChng Connector ID Status Change This interrupt is asserted when there is a change in connector ID status.
27	W1C	0x0	LPM_Int LPM Transaction Received Interrupt Device Mode: This interrupt is asserted when the device receives an LPM transaction and responds with a non-ERRORed response. Host Mode: This interrupt is asserted when the device responds to an LPM transaction with a non-ERRORed response or when the host core has completed LPM transactions for the programmed number of times (GLPMCFG.RetryCnt). This field is valid only if the Core LPM Configuration register's LPMCapable (LPMCap) field is set to 1.
26	RO	0x0	PTxFEmp Periodic TxFIFO Empty This interrupt is asserted when the Periodic Transmit FIFO is either half or completely empty and there is space for at least one entry to be written in the Periodic Request Queue. The half or completely empty status is determined by the Periodic TxFIFO Empty Level bit in the Core AHB Configuration register (GAHBCFG.PTxFEmpLvl).

Bit	Attr	Reset Value	Description
25	RO	0x0	HChInt Host Channels Interrupt The core sets this bit to indicate that an interrupt is pending on one of the channels of the core (in Host mode). The application must read the Host All Channels Interrupt (HAINT) register to determine the exact number of the channel on which the interrupt occurred, and then read the corresponding Host Channel-n Interrupt (HCINTn) register to determine the exact cause of the interrupt. The application must clear the appropriate status bit in the HCINTn register to clear this bit.
24	RO	0x0	PrtInt Host Port Interrupt The core sets this bit to indicate a change in port status of one of the ports in Host mode. The application must read the Host Port Control and Status (HPRT) register to determine the exact event that caused this interrupt. The application must clear the appropriate status bit in the Host Port Control and Status register to clear this bit.
23	RW	0x0	ResetDet Reset Detected Interrupt The core asserts this interrupt in Device mode when it detects a reset on the USB in Partial Power-Down mode when the device is in Suspend. This interrupt is not asserted in Host mode.
22	W1C	0x0	FetSusp Data Fetch Suspended This interrupt is valid only in DMA mode. This interrupt indicates that the core has stopped fetching data for IN endpoints due to the unavailability of TxFIFO space or Request Queue space. This interrupt is used by the application for an endpoint mismatch algorithm. For example, after detecting an endpoint mismatch, the application: Sets a global non-periodic IN NAK handshake, Disables In endpoints, Flushes the FIFO, Determines the token sequence from the IN Token Sequence Learning Queue, Re-enables the endpoints, Clear the global non-periodic IN NAK handshake. If the global non-periodic IN NAK is cleared, the core has not yet fetched data for the IN endpoint, and the IN token is received: the core generates an "IN token received when FIFO empty" interrupt. The OTG then sends the host a NAK response. To avoid this scenario, the application can check the GINTSTS.FetSusp interrupt, which ensures that the FIFO is full before clearing a global NAK handshake. Alternatively, the application can mask the IN token received when FIFO empty interrupt when clearing a global IN NAK handshake.

Bit	Attr	Reset Value	Description
21	W1C	0x0	incompIP Incomplete Periodic Transfer In Host mode, the core sets this interrupt bit when there are incomplete periodic transactions still pending which are scheduled for the current microframe. Incomplete Isochronous OUT Transfer (incompISOOUT) The Device mode, the core sets this interrupt to indicate that there is at least one isochronous OUT endpoint on which the transfer is not completed in the current microframe. This interrupt is asserted along with the End of Periodic Frame Interrupt (EOPF) bit in this register.
20	W1C	0x0	incompISOIN Incomplete Isochronous IN Transfer The core sets this interrupt to indicate that there is at least one isochronous IN endpoint on which the transfer is not completed in the current microframe. This interrupt is asserted along with the End of Periodic Frame Interrupt (EOPF) bit in this register. Note: This interrupt is not asserted in Scatter/Gather DMA mode.
19	RO	0x0	OEPInt OUT Endpoints Interrupt The core sets this bit to indicate that an interrupt is pending on one of the OUT endpoints of the core (in Device mode). The application must read the Device All Endpoints Interrupt (DAINT) register to determine the exact number of the OUT endpoint on which the interrupt occurred, and then read the corresponding Device OUT Endpoint-n Interrupt (DOEPINTn) register to determine the exact cause of the interrupt. The application must clear the appropriate status bit in the corresponding DOEPINTn register to clear this bit.
18	RO	0x0	IEPInt IN Endpoints Interrupt The core sets this bit to indicate that an interrupt is pending on one of the IN endpoints of the core (in Device mode). The application must read the Device All Endpoints Interrupt (DAINT) register to determine the exact number of the IN endpoint on which the interrupt occurred, and then read the corresponding Device IN Endpoint-n Interrupt (DIEPINTn) register to determine the exact cause of the interrupt. The application must clear the appropriate status bit in the corresponding DIEPINTn register to clear this bit.

Bit	Attr	Reset Value	Description
17	W1C	0x0	EPMis Endpoint Mismatch Interrupt Note: This interrupt is valid only in shared FIFO operation. Indicates that an IN token has been received for a non-periodic endpoint, but the data for another endpoint is present in the top of the Non-periodic Transmit FIFO and the IN endpoint mismatch count programmed by the application has expired.
16	W1C	0x0	RstrDoneInt Restore Done Interrupt The core sets this bit to indicate that the restore command after Hibernation was completed by the core. The core continues from Suspended state into the mode dictated by PCGCCTL.RestoreMode field. This bit is valid only when Hibernation feature is enabled.
15	W1C	0x0	EOPF End of Periodic Frame Interrupt Indicates that the period specified in the Periodic Frame Interval field of the Device Configuration register (DCFG.PerFrInt) has been reached in the current microframe.
14	W1C	0x0	ISOOutDrop Isochronous OUT Packet Dropped Interrupt The core sets this bit when it fails to write an isochronous OUT packet into the RxFIFO because the RxFIFO does not have enough space to accommodate a maximum packet size packet for the isochronous OUT endpoint.
13	W1C	0x0	EnumDone Enumeration Done The core sets this bit to indicate that speed enumeration is complete. The application must read the Device Status (DSTS) register to obtain the enumerated speed.
12	W1C	0x0	USBRst USB Reset The core sets this bit to indicate that a reset is detected on the USB.
11	W1C	0x0	USBSusp USB Suspend The core sets this bit to indicate that a suspended was detected on the USB. The core enters the Suspended state when there is no activity on the utmi_linestate signal for an extended period of time.
10	W1C	0x0	ErlySusp Early Suspend The core sets this bit to indicate that an Idle state has been detected on the USB for 3ms.

Bit	Attr	Reset Value	Description
9	W1C	0x0	I2CINT I2C Interrupt The core sets this interrupt when I2C access is completed on the I2C interface. This field is used only if the I2C interface was enabled. Otherwise, reads return 0.
8	W1C	0x0	ULPICKINT ULPI Carkit Interrupt This field is used only if the Carkit interface was enabled. Otherwise, reads return 0. The core sets this interrupt when a ULPI Carkit interrupt is received. The core's PHY sets ULPI Carkit interrupt in UART or Audio mode. I2C Carkit Interrupt (I2CCKINT) This field is used only if the I2C interface was enabled. Otherwise, reads return 0.The core sets this interrupt when a Carkit interrupt is received. The core's PHY sets the I2C Carkit interrupt in Audio mode.
7	RO	0x0	GOUTNakEff Global OUT NAK Effective Indicates that the Set Global OUT NAK bit in the Device Control register DCTL.SGOUTNak), set by the application, has taken effect in the core. This bit can be cleared by writing the Clear Global OUT NAK bit in the Device Control register (DCTL.CGOUTNak).
6	RO	0x0	GINNakEff Global IN Non-Periodic NAK Effective Indicates that the Set Global Non-periodic IN NAK bit in the Device Control register (DCTL.SGNPInNak), set by the application, have taken effect in the core. That is, the core has sampled the Global IN NAK bit set by the application. This bit can be cleared by clearing the Clear Global Nonperiodic IN NAK bit in the Device Control register (DCTL.CGNPInNak). This interrupt does not necessarily mean that a NAK handshake is sent out on the USB. The STALL bit takes precedence over the NAK bit.
5	RO	0x0	NPTxFEmp Non-Periodic TxFIFO Empty This interrupt is valid only when OTG_EN_DED_TX_FIFO = 0. This interrupt is asserted when the Non-periodic TxFIFO is either half or completely empty, and there is space for at least one entry to be written to the Non-periodic Transmit Request Queue. The half or completely empty status is determined by the Non-periodic TxFIFO Empty Level bit in the Core AHB Configuration register(GAHBCFG.NPTxFEmpLvl).
4	RO	0x0	RxFLvl RxFIFO Non-Empty Indicates that there is at least one packet pending to be read from the RxFIFO.

Bit	Attr	Reset Value	Description
3	W1C	0x0	Sof Start of (micro)Frame In Host mode, the core sets this bit to indicate that an SOF (FS), micro-SOF(HS), or Keep-Alive (LS) is transmitted on the USB. The application must write a 1 to this bit to clear the interrupt. In Device mode, in the core sets this bit to indicate that an SOF token has been received on the USB. The application can read the Device Status register to get the current (micro)frame number. This interrupt is seen only when the core is operating at either HS or FS.
2	RO	0x0	OTGInt OTG Interrupt The core sets this bit to indicate an OTG protocol event. The application must read the OTG Interrupt Status (GOTGINT) register to determine the exact event that caused this interrupt. The application must clear the appropriate status bit in the GOTGINT register to clear this bit.
1	W1C	0x0	ModeMis Mode Mismatch Interrupt The core sets this bit when the application is trying to access: A Host mode register, when the core is operating in Device mode; A Device mode register, when the core is operating in Host mode. The register access is completed on the AHB with an OKAY response, but is ignored by the core internally and does not affect the operation of the core.
0	RO	0x0	CurMod Current Mode of Operation Indicates the current mode. 1'b0: Device mode 1'b1: Host mode

USBOTG_GINTMSK
Address: Operational Base + offset (0x0018)
Interrupt Mask Register

Bit	Attr	Reset Value	Description
31	RW	0x0	WkUpIntMsk Resume/Remote Wakeup Detected Interrupt Mask
30	RW	0x0	SessReqIntMsk Session Request/New Session Detected Interrupt Mask
29	RW	0x0	DisconnIntMsk Disconnect Detected Interrupt Mask
28	RW	0x0	ConIDStsChngMsk Connector ID Status Change Mask
27	RW	0x0	LPM_IntMsk LPM Transaction Received Interrupt Mask

Bit	Attr	Reset Value	Description
26	RW	0x0	PTxFEmpMsk Periodic TxFIFO Empty Mask
25	RW	0x0	HChIntMsk Host Channels Interrupt Mask
24	RW	0x0	PrtIntMsk Host Port Interrupt Mask
23	RW	0x0	ResetDetMsk Reset Detected Interrupt Mask
22	RW	0x0	FetSuspMsk Data Fetch Suspended Mask
21	RW	0x0	incomplPMsk_incompISOOUTMsk Incomplete Periodic Transfer Mask(Host only) Incomplete Isochronous OUT Transfer Mask(Device only)
20	RW	0x0	incompISOINMsk Incomplete Isochronous IN Transfer Mask
19	RW	0x0	OEPIntMsk OUT Endpoints Interrupt Mask
18	RW	0x0	IEPIntMsk IN Endpoints Interrupt Mask
17	RW	0x0	EPMisMsk Endpoint Mismatch Interrupt Mask
16	RW	0x0	RstrDoneIntMsk Restore Done Interrupt Mask This field is valid only when Hibernation feature is enabled.
15	RW	0x0	EOPFMsk End of Periodic Frame Interrupt Mask
14	RW	0x0	ISOOutDropMsk Isochronous OUT Packet Dropped Interrupt Mask
13	RW	0x0	EnumDoneMsk Enumeration Done Mask
12	RW	0x0	USBRstMsk USB Reset Mask
11	RW	0x0	USBSuspMsk USB Suspend Mask
10	RW	0x0	ErlySuspMsk Early Suspend Mask
9	RW	0x0	I2CIntMsk I2C Interrupt Mask
8	RW	0x0	ULPICKINTMsk_I2CCKINTMsk ULPI Carkit Interrupt Mask (ULPICKINTMsk) I2C Carkit Interrupt Mask (I2CCKINTMsk)
7	RW	0x0	GOUTNakEffMsk Global OUT NAK Effective Mask

Bit	Attr	Reset Value	Description
6	RW	0x0	GINNakEffMsk Global Non-periodic IN NAK Effective Mask
5	RW	0x0	NPTxFEmpMsk Non-periodic TxFIFO Empty Mask
4	RW	0x0	RxFLvlMsk Receive FIFO Non-Empty Mask
3	RW	0x0	SofMsk Start of (micro)Frame Mask
2	RW	0x0	OTGIntMsk OTG Interrupt Mask
1	RW	0x0	ModeMisMsk Mode Mismatch Interrupt Mask
0	RO	0x0	reserved

USBOTG_GRXSTSR
Address: Operational Base + offset (0x001c)
Receive Status Debug Read Register

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24:21	RO	0x0	FN Frame Number (Device Only)This is the least significant 4 bits of the (micro) frame number in which the packet is received on the USB. This field is supported only when isochronous OUT endpoints are supported.
20:17	RO	0x0	PktSts Packet Status Indicates the status of the received packet(Host Only) 4'b0010: IN data packet received 4'b0011: IN transfer completed (triggers an interrupt) 4'b0101: Data toggle error (triggers an interrupt) 4'b0111: Channel halted (triggers an interrupt) Others: Reserved Indicates the status of the received packet(Device only) 4'b0001: Global OUT NAK (triggers an interrupt) 4'b0010: OUT data packet received 4'b0011: OUT transfer completed (triggers an interrupt) 4'b0100: SETUP transaction completed (triggers an interrupt) 4'b0110: SETUP data packet received Others: Reserved

Bit	Attr	Reset Value	Description
16:15	RO	0x0	DPID Data PID Indicates the Data PID of the received packet 2'b00: DATA0 2'b10: DATA1 2'b01: DATA2 2'b11: MDATA
14:4	RW	0x000	BCnt Byte Count Indicates the byte count of the received data packet.
3:0	RO	0x0	ChNum_EPNum Channel Number(Host) Endpoint Number(Device) (Host Only) Indicates the channel number to which the current received packet belongs. (Device Only) Indicates the endpoint number to which the current received packet belongs.

USBOTG_GRXSTSP
Address: Operational Base + offset (0x0020)
Receive Status Read and Pop Register

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24:21	RO	0x0	FN Frame Number (Device Only) This is the least significant 4 bits of the (micro)frame number in which the packet is received on the USB. This field is supported only when isochronous OUT endpoints are supported.
20:17	RO	0x0	PktSts Packet Status Indicates the status of the received packet(Host Only) 4'b0010: IN data packet received 4'b0011: IN transfer completed (triggers an interrupt) 4'b0101: Data toggle error (triggers an interrupt) 4'b0111: Channel halted (triggers an interrupt) Others: Reserved Indicates the status of the received packet(Device only) 4'b0001: Global OUT NAK (triggers an interrupt) 4'b0010: OUT data packet received 4'b0011: OUT transfer completed (triggers an interrupt) 4'b0100: SETUP transaction completed (triggers an interrupt) 4'b0110: SETUP data packet received Others: Reserved

Bit	Attr	Reset Value	Description
16:15	RO	0x0	DPID Data PID Indicates the Data PID of the received OUT data packet 2'b00: DATA0 2'b10: DATA1 2'b01: DATA2 2'b11: MDATA
14:4	RO	0x000	BCnt Byte Count Indicates the byte count of the received data packet.
3:0	RO	0x0	ChNum_EPNum Channel Number(Host) Endpoint Number(Device) (Host Only) Indicates the channel number to which the current received packet belongs. (Device Only) Indicates the endpoint number to which the current received packet belongs.

USBOTG_GRXFSIZ
Address: Operational Base + offset (0x0024)
Receive FIFO Size Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	RxFDep RxFIFO Depth This value is in terms of 32-bit words. Minimum value is 16. Maximum value is 32,768. The power-on reset value of this register is specified as the Largest Rx Data FIFO Depth. If Enable Dynamic FIFO Sizing was deselected, these flops are optimized, and reads return the power-on value. If Enable Dynamic FIFO Sizing was selected, you can write a new value in this field. You can write a new value in this field. Programmed values must not exceed the power-on value.

USBOTG_GNPTXFSIZ
Address: Operational Base + offset (0x0028)
Non-Periodic Transmit FIFO Size Register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	NPTxFDep Non-periodic TxFIFO For host mode, this field is always valid. For Device mode, this field is valid only when OTG_EN_DED_TX_FIFO==0. This value is in terms of 32-bit words. Minimum value is 16. Maximum value is 32,768 This field is determined by Enable Dynamic FIFO Sizing. OTG_DFIFO_DYNAMIC = 0: These flops are optimized, and reads return the power-on value. OTG_DFIFO_DYNAMIC = 1: The application can write a new value in this field. Programmed values must not exceed the power-on value. The power-on reset value of this field is specified by OTG_EN_DED_TX_FIFO: OTG_EN_DED_TX_FIFO = 0:The reset value is the Largest Non-periodic Tx Data FIFO Depth parameter, OTG_TX_NPERIO_DFIFO_DEPTH. OTG_EN_DED_TX_FIFO = 1:The reset value is parameter OTG_TX_HNPERIO_DFIFO_DEPTH.
15:0	RW	0x0000	NPTxFStAddr Non-periodic Transmit RAM For host mode, this field is always valid. This field contains the memory start address for Non-periodic Transmit FIFO RAM. This field is determined by Enable Dynamic FIFO Sizing(OTG_DFIFO_DYNAMIC): OTG_DFIFO_DYNAMIC = 0:These flops are optimized, and reads return the power-on value. OTG_DFIFO_DYNAMIC = 1:The application can write a new value in this field. Programmed values must not exceed the power-on value. The power-on reset value of this field is specified by Largest Rx Data FIFO Depth (parameter OTG_RX_DFIFO_DEPTH).

USBOTG_GNPTXSTS
Address: Operational Base + offset (0x002c)
Non-Periodic Transmit FIFO/Queue Status Register

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved

Bit	Attr	Reset Value	Description
30:24	RO	0x00	NPTxQTop Top of the Non-periodic Transmit Request Queue Entry in the Non-periodic Tx Request Queue that is currently being processed by the MAC. Bits [30:27]: Channel/endpoint number Bits [26:25]: 2'b00: IN/OUT token 2'b01: Zero-length transmit packet (device IN/host OUT) 2'b10: PING/CSPLIT token 2'b11: Channel halt command Bit [24]: Terminate (last entry for selected channel/endpoint)
23:16	RO	0x00	NPTxQSpcAvail Non-periodic Transmit Request Queue Space Available Indicates the amount of free space available in the Non-periodic Transmit Request Queue. This queue holds both IN and OUT requests in Host mode. Device mode has only IN requests. 8'h0: Non-periodic Transmit Request Queue is full 8'h1: 1 location available 8'h2: 2 locations available n: n locations available (0 <=n <= 8) Others: Reserved
15:0	RO	0x0000	NPTxFSpcAvail Non-periodic TxFIFO Space Avail Indicates the amount of free space available in the Non-periodic TxFIFO. Values are in terms of 32-bit words. 16'h0: Non-periodic TxFIFO is full 16'h1: 1 word available 16'h2: 2 words available 16'hn: n words available (where 0 <=n <=32,768) 16'h8000: 32,768 words available Others: Reserved

USBOTG_GI2CCTL
Address: Operational Base + offset (0x0030)
I2C Address Register

Bit	Attr	Reset Value	Description
31	R/W SC	0x0	BsyDne I2C Busy/Done The application sets this bit to 1'b1 to start a request on the I2C interface. When the transfer is complete, the core de-asserts this bit to 1'b0. As long as the bit is set, indicating that the I2C interface is busy, the application cannot start another request on the interface.

Bit	Attr	Reset Value	Description
30	RW	0x0	RW Read/Write Indicator Indicates whether a read or write register transfer must be performed on the interface. Read/write bursting is not supported for registers. 1'b1: Read 1'b0: Write
29	RO	0x0	reserved
28	RW	0x1	I2CDatSe0 I2C DatSe0 USB Mode Selects the FS interface USB mode. 1'b1: VP_VM USB mode 1'b0: DAT_SE0 USB mode
27:26	RW	0x0	I2CDevAdr I2C Device Address Selects the address of the I2C Slave on the USB 1.1 full-speed serial transceiver that the core uses for OTG signaling. 2'b00: 7'h2C 2'b01: 7'h2D 2'b10: 7'h2E 2'b11: 7'h2F
25	RW	0x0	I2CSuspCtl I2C Suspend Control Selects how Suspend is connected to a full-speed transceiver in I2C mode. 1'b0: Use the dedicated utmi_suspend_n pin 1'b1: Use an I2C write to program the Suspend bit in the PHY register
24	RO	0x1	Ack I2C ACK Indicates whether an ACK response was received from the I2C Slave. This bit is valid when BsyDne is cleared by the core, after application has initiated an I2C access. 1'b0: NAK 1'b1: ACK
23	RW	0x0	I2CEn I2C Enable Enables the I2C Master to initiate I2C transactions on the I2C interface
22:16	RW	0x00	Addr I2C Address This is the 7-bit I2C device address used by software to access any external I2C Slave, including the I2C Slave on a USB 1.1 OTG full-speed serial transceiver. Software can change this address to access different I2C Slaves.

Bit	Attr	Reset Value	Description
15:8	RW	0x00	RegAddr I2C Register Addr This field programs the address of the register to be read from or written to.
7:0	RW	0x00	RWData I2C Read/Write Data After a register read operation, this field holds the read data for the application. During a write operation, the application can use this register to program the write data to be written to a register. During writes, this field holds the write data.

USBOTG_GPVNDCTL
Address: Operational Base + offset (0x0034)
PHY Vendor Control Register

Bit	Attr	Reset Value	Description
31	R/W SC	0x0	DisUlpiDrvr Disable ULPI Drivers This field is used only if the Carkit interface was enabled in coreConsultant (parameter OTG_ULPI_CARKIT = 1). Otherwise, reads return 0. The application sets this bit when it has finished processing the ULPI Carkit Interrupt (GINTSTS.ULPICKINT). When set, the core disables drivers for output signals and masks input signal for the ULPI interface. Core clears this bit before enabling the ULPI interface.
30:28	RO	0x0	reserved
27	R/W SC	0x0	VStsDone VStatus Done The core sets this bit when the vendor control access is done. This bit is cleared by the core when the application sets the New Register Request bit (bit 25).
26	RO	0x0	VStsBsy VStatus Busy The core sets this bit when the vendor control access is in progress and clears this bit when done.
25	R/W SC	0x0	NewRegReq New Register Request The application sets this bit for a new vendor control access.
24:23	RO	0x0	reserved
22	RW	0x0	RegWr Register Write Set this bit for register writes, and clear it for register reads.
21:16	RW	0x00	RegAddr Register Address The 6-bit PHY register address for immediate PHY Register Set access. Set to 6'h2F for Extended PHY Register Set access.

Bit	Attr	Reset Value	Description
15:8	RW	0x00	VCtrl UTMI+ Vendor Control Register Address The 4-bit register address a vendor defined 4-bit parallel output bus. Bits 11:8 of this field are placed on utmi_vcontrol[3:0]. ULPI Extended Register Address (ExtRegAddr) The 6-bit PHY extended register address.
7:0	RW	0x00	RegData Register Data Contains the write data for register write. Read data for register read, valid when VStatus Done is set.

USBOTG_GGPIO
Address: Operational Base + offset (0x0038)
General Purpose Input/Output Register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	GPO General Purpose Output This field is driven as an output from the core, gp_o[15:0]. The application can program this field to determine the corresponding value on the gp_o[15:0] output.
15:0	RO	0x0000	GPI General Purpose Input This field's read value reflects the gp_i[15:0] core input value.

USBOTG_GUID
Address: Operational Base + offset (0x003c)
User ID Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	UserID Application-programmable ID field.

USBOTG_GSNPSID
Address: Operational Base + offset (0x0040)
Core ID Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00004f54	CoreID Release number of the core being used

USBOTG_GHWCFG1
Address: Operational Base + offset (0x0044)
User HW Config1 Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	epdir Endpoint Direction This 32-bit field uses two bits per endpoint to determine the endpoint direction. Endpoint Bits [31:30]: Endpoint 15 direction Bits [29:28]: Endpoint 14 direction ... Bits [3:2]: Endpoint 1 direction Bits[1:0]: Endpoint 0 direction (always BIDIR) Direction 2'b00: BIDIR (IN and OUT) endpoint 2'b01: IN endpoint 2'b10: OUT endpoint 2'b11: Reserved

USBOTG_GHWCFG2
Address: Operational Base + offset (0x0048)
User HW Config2 Register

Bit	Attr	Reset Value	Description
31	RO	0x0	OTG_ENABLE_IC_USB IC_USB mode specified for mode of operation (parameter OTG_ENABLE_IC_USB). To choose IC_USB_MODE, both OTG_FSPHY_INTERFACE and OTG_ENABLE_IC_USB must be 1.
30:26	RO	0x00	TknQDepth Device Mode IN Token Sequence Learning Queue Depth Range: 0-30
25:24	RO	0x0	PTxQDepth Host Mode Periodic Request Queue Depth 2'b00: 2 2'b01: 4 2'b10: 8 Others: Reserved
23:22	RO	0x0	NPTxQDepth Non-periodic Request Queue Depth 2'b00: 2 2'b01: 4 2'b10: 8 Others: Reserved
21	RO	0x0	reserved
20	RO	0x0	MultiProcIntrpt Multi Processor Interrupt Enabled 1'b0: No 1'b1: Yes

Bit	Attr	Reset Value	Description
19	RO	0x0	DynFifoSizing Dynamic FIFO Sizing Enabled 1'b0: No 1'b1: Yes
18	RO	0x0	PerioSupport Periodic OUT Channels Supported in Host Mode 1'b0: No 1'b1: Yes
17:14	RO	0x0	NumHstChnl Number of Host Channels Indicates the number of host channels supported by the core in Host mode. The range of this field is 0-15: 0 specifies 1 channel, 15 specifies 16 channels.
13:10	RO	0x0	NumDevEps Number of Device Endpoints Indicates the number of device endpoints supported by the core in Device mode in addition to control endpoint 0. The range of this field is 1-15.
9:8	RO	0x0	FSPhyType Full-Speed PHY Interface Type 2'b00: Full-speed interface not supported 2'b01: Dedicated full-speed interface 2'b10: FS pins shared with UTMI+ pins 2'b11: FS pins shared with ULPI pins
7:6	RO	0x0	HSPhyType High-Speed PHY Interface Type 2'b00: High-Speed interface not supported 2'b01: UTMI+ 2'b10: ULPI 2'b11: UTMI+ and ULPI
5	RO	0x0	SingPnt Point-to-Point 1'b0: Multi-point application (hub and split support) 1'b1: Single-point application (no hub and no split support)
4:3	RO	0x0	OtgArch Architecture 2'b00: Slave-Only 2'b01: External DMA 2'b10: Internal DMA Others: Reserved

Bit	Attr	Reset Value	Description
2:0	RO	0x0	OtgMode Mode of Operation 3'b000: HNP- and SRP-Capable OTG (Host and Device) 3'b001: SRP-Capable OTG (Host and Device) 3'b010: Non-HNP and Non-SRP Capable OTG (Host and Device) 3'b011: SRP-Capable Device 3'b100: Non-OTG Device 3'b101: SRP-Capable Host 3'b110: Non-OTG Host Others: Reserved

USBOTG_GHWCFG3
Address: Operational Base + offset (0x004c)
User HW Config3 Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	DfifoDepth DFIFO Depth This value is in terms of 32-bit words. Minimum value is 32 Maximum value is 32,768
15	RO	0x0	OTG_ENABLE_LPM LPM mode specified for Mode of Operation (parameter OTG_ENABLE_LPM).
14	RO	0x0	OTG_BC_SUPPORT This bit indicates the HS OTG controller support for Battery Charger. 0 : No Battery Charger Support 1 : Battery Charger support present.
13	RO	0x0	OTG_ENABLE_HSIC HSIC mode specified for Mode of Operation (parameter OTG_ENABLE_HSIC). Value Range: 0-1 1: HSIC-capable with shared UTMI PHY interface 0: Non-HSIC-capable
12	RO	0x0	OTG_ADP_SUPPORT This bit indicates whether ADP logic is present within or external to the HS OTG controller 0: No ADP logic present with HSOTG controller 1: ADP logic is present along with HSOTG controller.
11	RO	0x0	RstType Reset Style for Clocked always Blocks in RTL 1'b0: Asynchronous reset is used in the core 1'b1: Synchronous reset is used in the core

Bit	Attr	Reset Value	Description
10	RO	0x0	OptFeature Optional Features Removed Indicates whether the User ID register, GPIO interface ports, and SOF toggle and counter ports were removed for gate count optimization by enabling Remove Optional Features? 1'b0: No 1'b1: Yes
9	RO	0x0	VndctlSupt Vendor Control Interface Support 1'b0: Vendor Control Interface is not available on the core. 1'b1: Vendor Control Interface is available.
8	RO	0x0	I2CIntSel I2C Selection 1'b0: I2C Interface is not available on the core. 1'b1: I2C Interface is available on the core.
7	RO	0x0	OtgEn OTG Function Enabled The application uses this bit to indicate the OTG capabilities. 1'b0: Not OTG capable 1'b1: OTG Capable
6:4	RO	0x0	PktSizeWidth Width of Packet Size Counters 3'b000: 4 bits 3'b001: 5 bits 3'b010: 6 bits 3'b011: 7 bits 3'b100: 8 bits 3'b101: 9 bits 3'b110: 10 bits Others: Reserved
3:0	RO	0x0	XferSizeWidth Width of Transfer Size Counters 4'b0000: 11 bits 4'b0001: 12 bits ... 4'b1000: 19 bits Others: Reserved

USBOTG_GHWCFG4
Address: Operational Base + offset (0x0050)
User HW Config4 Register

Bit	Attr	Reset Value	Description
31	RO	0x0	SGDMA Scatter/Gather DMA 1'b1: Dynamic configuration

Bit	Attr	Reset Value	Description
30	RO	0x0	SGDMACon Scatter/Gather DMA configuration 1'b0: Non-Scatter/Gather DMA configuration 1'b1: Scatter/Gather DMA configuration
29:26	RO	0x0	INEps Number of Device Mode IN Endpoints Including Control Endpoint Range 0 -15 0:1 IN Endpoint 1:2 IN Endpoints .... 15:16 IN Endpoints
25	RW	0x0	DedFifoMode Enable Dedicated Transmit FIFO for device IN Endpoints 1'b0: Dedicated Transmit FIFO Operation not enabled. 1'b1: Dedicated Transmit FIFO Operation enabled.
24	RW	0x0	SessEndFltr session_end Filter Enabled 1'b0: No filter 1'b1: Filter
23	RW	0x0	BValidFltr "b_valid" Filter Enabled 1'b0: No filter 1'b1: Filter
22	RO	0x0	AValidFltr "a_valid" Filter Enabled 1'b0: No filter 1'b1: Filter
21	RO	0x0	VBusValidFltr "vbus_valid" Filter Enabled 1'b0: No filter 1'b1: Filter
20	RO	0x0	IddgFltr "iddig" Filter Enable 1'b0: No filter 1'b1: Filter
19:16	RO	0x0	NumCtlEps Number of Device Mode Control Endpoints in Addition to Endpoin Range: 0-15

Bit	Attr	Reset Value	Description
15:14	RO	0x0	PhyDataWidth UTMI+ PHY/ULPI-to-Internal UTMI+ Wrapper Data Width When a ULPI PHY is used, an internal wrapper converts ULPI to UTMI+. 2'b00: 8 bits 2'b01: 16 bits 2'b10: 8/16 bits, software selectable Others: Reserved
13:7	RO	0x0	reserved
6	RO	0x0	EnHiber Enable Hibernation 1'b0: Hibernation feature not enabled 1'b1: Hibernation feature enabled
5	RO	0x0	AhbFreq Minimum AHB Frequency Less Than 60 MHz 1'b0: No 1'b1: Yes
4	RO	0x0	EnParPwrDown Enable Partial Power Down 1'b0: Partial Power Down Not Enabled 1'b1: Partial Power Down Enabled
3:0	RO	0x0	NumDevPerioEps Number of Device Mode Periodic IN Endpoints Range: 0-15

USBOTG_GLPMCFG
Address: Operational Base + offset (0x0054)
Core LPM Configuration Register

Bit	Attr	Reset Value	Description
31	RW	0x0	InvSelHsic HSIC-Invert Select HSIC The application uses this bit to control the HSIC enable/disable. This bit overrides and functionally inverts the if_sel_hsic input port signal. If the core operates as non-HSIC-capable, it can only connect to non-HSIC-capable PHYs. If the core operates as HSIC-capable, it can only connect to HSICcapable PHYs. If the if_sel_hsic input signal is 1: 1'b1: HSIC capability is not enabled. 1'b0: HSIC capability is enabled, If InvSelHsic = 1'b0: HSIC capability is enabled. If the if_sel_hsic input signal is 0: 1'b1: HSIC capability is enabled, 1'b0: HSIC capability is not enabled. This bit is writable only if an HSIC mode was specified for Mode of Operation (parameter OTG_ENABLE_HSIC). This bit is valid only if OTG_ENABLE_HSIC is enabled.
30	RW	0x0	HSICCon HSIC-Connect The application must use this bit to initiate the HSIC Attach sequence. Host Mode: Once this bit is set, the host core configures to drive the HSIC Idle state (STROBE = 1 & DATA = 0) on the bus. It then waits for the device to initiate the Connect sequence. Device Mode: Once this bit is set, the device core waits for the HSIC Idle line state on the bus. Upon receiving the Idle line state, it initiates the HSIC Connect sequence. This bit is valid only if OTG_ENABLE_HSIC is 1, if_sel_hsic = 1 and InvSelHSIC is 0. Otherwise, it is read-only.
29:28	RO	0x0	reserved
27:25	RO	0x0	LPM_RetryCnt_Sts LPM Retry Count Status Number of LPM host retries remaining to be transmitted for the current LPM sequence.
24	RW	0x0	SndLPM Send LPM Transaction Host Mode: When the application software sets this bit, an LPM transaction containing two tokens, EXT and LPM, is sent. The hardware clears this bit once a valid response (STALL, NYET, or ACK) is received from the device or the core has finished transmitting the programmed number of LPM retries. Note: This bit must only be set when the host is connected to a local port.

Bit	Attr	Reset Value	Description
23:21	R/W SC	0x0	LPM_Retry_Cnt LPM Retry Count When the device gives an ERROR response, this is the number of additional LPM retries that the host performs until a valid device response (STALL, NYET, or ACK) is received.
20:17	RW	0x0	LPM_Chnl_Indx LPM Channel Index The channel number on which the LPM transaction must be applied while sending an LPM transaction to the local device. Based on the LPM channel index, the core automatically inserts the device address and endpoint number programmed in the corresponding channel into the LPM transaction.
16	RO	0x0	L1ResumeOK Sleep State Resume OK Indicates that the application or host can start a resume from the Sleep state. This bit is valid in the LPM Sleep (L1) state. It is set in Sleep mode after a delay of 50 us (TL1Residency). The bit is reset when SlpSts is 0 1'b1: The application/core can start resume from the Sleep state 1'b0: The application/core cannot start resume from the Sleep state

Bit	Attr	Reset Value	Description
15	RO	0x0	SlpSts Port Sleep Status Device Mode: This bit is set as long as a Sleep condition is present on the USB bus. The core enters the Sleep state when an ACK response is sent to an LPM transaction and the timer TL1TokenRetry has expired. To stop the PHY clock, the application must set the Port Clock Stop bit, which asserts the PHY Suspend input pin. The application must rely on SlpSts and not ACK in CoreL1Res to confirm transition into sleep. The core comes out of sleep: When there is any activity on the USB line_state, When the application writes to the Remote Wakeup Signaling bit in the Device Control register (DCTL.RmtWkUpSig) or when the application resets or soft-disconnects the device. Host Mode: The host transitions to the Sleep (L1) state as a side-effect of a successful LPM transaction by the core to the local port with an ACK response from the device. The read value of this bit reflects the port's current sleep status. The core clears this bit after: The core detects a remote L1 Wakeup signal, The application sets the Port Reset bit or the Port L1Resume bit in the HPRT register or The application sets the L1Resume/ Remote Wakeup Detected Interrupt bit or Disconnect Detected Interrupt bit in the Core Interrupt register (GINTSTS.L1WkUpInt or GINTSTS.DisconnInt, respectively). Values: 1'b0: Core not in L1 1'b1: Core in L1
14:13	RO	0x0	CoreL1Res LPM Response Device Mode: The core's response to the received LPM transaction is reflected in these two bits. Host Mode: The handshake response received from the local device for LPM transaction. 11: ACK 10: NYET 01: STALL 00: ERROR (No handshake response)

Bit	Attr	Reset Value	Description
12:8	RW	0x00	HIRD_Thres HIRD Threshold Device Mode: The core asserts L1SuspendM to put the PHY into Deep Low-Power mode in L1 when the HIRD value is greater than or equal to the value defined in this field (GLPMCFG.HIRD_Thres[3:0]), and HIRD_Thres[4] is set to 1'b1. Host Mode: The core asserts L1SuspendM to put the PHY into Deep Low-Power mode in L1 when HIRD_Thres[4] is set to 1'b1. HIRD_Thres[3:0] specifies the time for which resume signaling is to be reflected by the host TL1HubDrvResume2) on the USB when it detects device-initiated resume. HIRD_Thres must not be programmed with a value greater than 4'b1100 in Host mode, because this exceeds maximum TL1HubDrvResume2. Sl. No HIRD_Thres[3:0] Host mode resume time(us) 1 4'b0000 60 2 4'b0001 135 3 4'b0010 210 4 4'b0011 285 5 4'b0100 360 6 4'b0101 435 7 4'b0110 510 8 4'b0111 585 9 4'b1000 660 10 4'b1001 735 11 4'b1010 810 12 4'b1011 885 13 4'b1100 960 14 4'b1101 invalid 15 4'b1110 invalid 16 4'b1111 invalid

Bit	Attr	Reset Value	Description
7	RW	0x0	EnblSlpM Enable utmi_sleep_n For ULPI interface: The application uses this bit to write to the function control [7] in the L1 state, to enable the PHY to go into Low Power mode. For the host, this bit is valid only in Local Device mode. 1'b0: Writes to the ULPI Function Control Bit[7] are disabled. 1'b1: The core is enabled to write to the ULPI Function Control Bit[7], which enables the PHY to enter Low-Power mode. Note: When a ULPI interface is configured, enabling this bit results in a write to Bit 7 of the ULPI Function Control register. The ULPI PHY supports writing to this bit, and in the L1 state asserts SleepM when utmi_l1_suspend_n cannot be asserted. When a ULPI interface is configured, this bit must always be set if you are using the ULPI PHY. Note: For ULPI interface, do not clear this bit during the resume. For all other interfaces: The application uses this bit to control utmi_sleep_n assertion to the PHY in the L1 state. For the host, this bit is valid only in Local Device mode. 1'b0: utmi_sleep_n assertion from the core is not transferred to the external PHY. 1'b1: utmi_sleep_n assertion from the core is transferred to the external PHY when utmi_l1_suspend_n cannot be asserted.
6	RW	0x0	bRemoteWake RemoteWakeEnable Host Mode: The remote wakeup value to be sent in the LPM transaction's wIndex field. Device Mode: This field is updated with the received bRemoteWake LPM token's bmAttribute when an ACK/NYET/STALL response is sent to an LPM transaction.

Bit	Attr	Reset Value	Description
5:2	RW	0x0	HIRD Host-Initiated Resume Duration Host Mode: The value of HIRD to be sent in an LPM transaction. This value is also used to initiate resume for a durationL1HubDrvResume1 for host initiated resume. Device Mode: This field is updated with the Received LPM Token HIRD bmAttribute when an ACK/NYET/STALL response is sent to an LPM transaction Sl. No HIRD[3:0] THIRD (us) 1 4'b0000 50 2 4'b0001 125 3 4'b0010 200 4 4'b0011 275 5 4'b0100 350 6 4'b0101 425 7 4'b0110 500 8 4'b0111 575 9 4'b1000 650 10 4'b1001 725 11 4'b1010 800 12 4'b1011 875 13 4'b1100 950 14 4'b1101 1025 15 4'b1110 1100 16 4'b1111 1175
1	RW	0x0	AppL1Res LPM response programmed by application Handshake response to LPM token pre-programmed by device application software. The response depends on GLPMCFG.LPMCap. If GLPMCFG.LPMCap is 1'b0, the core always responds with a NYET. If GLPMCFG.LPMCap is 1'b1, the core responds as follows: 1: ACK. Even though an ACK is pre-programmed, the core responds with an ACK only on a successful LPM transaction. The LPM transaction is successful if: There are no PID/CRC5 errors in both the EXT token and the LPM token (else ERROR); A valid bLinkState = 0001B (L1) is received in the LPM transaction (else STALL); No data is pending in the Transmit queue (else NYET) 0: NYET. The pre-programmed software bit is overridden for response to LPM token when:(1)The received bLinkState is not L1 (STALL response); (2)An error is detected in either of the LPM token packets due to corruption (ERROR response).

Bit	Attr	Reset Value	Description
0	RW	0x0	LPMCap LPM-Capable The application uses this bit to control the LPM capabilities. If the core operates as a non-LPM-capable host, it cannot request the connected device/hub to activate LPM mode. If the core operates as a non-LPM-capable device, it cannot respond to any LPM transactions. 1'b0: LPM capability is not enabled. 1'b1: LPM capability is enabled. This bit is writable only if an LPM mode was specified for Mode of Operation (parameter OTG_ENABLE_LPM). Otherwise, reads return 0.

USBOTG_GPWRDN
Address: Operational Base + offset (0x0058)
Global Power Down Register

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28:24	RO	0x00	MultValIdBC Multi Valued ID pin Battery Charger ACA inputs in the following order: Bit 26 - rid_float. Bit 25 - rid_gnd Bit 24 - rid_a Bit 23 - rid_b Bit 22 - rid_c These bits are present only if OTG_BC_SUPPORT = 1. Otherwise, these bits are reserved and will read 5'h0.
23	W1C	0x0	ADPInt ADP Interupt This bit is set whenever there is a ADP event.
22	RO	0x0	BSessVld B Session Valid This field reflects the B session valid status signal from the PHY. 1'b0: B-Valid is 0. 1'b1: B-Valid is 1. This bit is valid only when GPWRDN.PMUActv is 1.

Bit	Attr	Reset Value	Description
21	RO	0x0	IDDIG This bit indicates the status of the signal IDDIG. The application must read this bit after receiving GPWRDN.StsChngInt and decode based on the previous value stored by the application. Indicates the current mode. 1'b1: Device mode 1'b0: Host mode This bit is valid only when GPWRDN.PMUActv is 1.
20:19	RO	0x0	LineState This field indicates the current linestate on USB as seen by the PMU module. 2'b00: DM = 0, DP = 0. 2'b01: DM = 0, DP = 1. 2'b10: DM = 1, DP = 0. 2'b11: Not-defined. This bit is valid only when GPWRDN.PMUActv is 1.
18	RW	0x0	StsChngIntMsk Mask For StsChng Interrupt
17	W1C	0x0	StsChngInt This field indicates a status change in either the IDDIG or BSessVld signal. 1'b0: No Status change 1'b1: status change detected After receiving this interrupt the application should read the GPWRDN register and interpret the change in IDDIG or BSesVld with respect to the previous value stored by the application.
16	RW	0x0	SRPDetectMsk Mask For SRPDetect Interrupt
15	W1C	0x0	SRPDetect This field indicates that SRP has been detected by the PMU. This field generates an interrupt. After detecting SRP during hibernation the application should not restore the core. The application should get into the initialization process. 1'b0: SRP not detected 1'b1: SRP detected
14	RW	0x0	ConnDetMsk Mask for ConnectDet interrupt This bit is valid only when OTG_EN_PWROPT = 2.
13	W1C	0x0	ConnectDet This field indicates that a new connect has been detected 1'b0: Connect not detected 1'b1: Connect detected This bit is valid only when OTG_EN_PWROPT = 2.

Bit	Attr	Reset Value	Description
12	RW	0x0	DisconnectDetectMsk Mask For DisconnectDetect Interrupt This bit is valid only when OTG_EN_PWROPT = 2.
11	W1C	0x0	DisconnectDetect This field indicates that Disconnect has been detected by the PMU. This field generates an interrupt. After detecting disconnect during hibernation the application must not restore the core, but instead start the initialization process. 1'b0: Disconnect not detected 1'b1: Disconnect detected This bit is valid only when OTG_EN_PWROPT = 2.
10	RW	0x0	ResetDetMsk Mask For ResetDetected interrupt. This bit is valid only when OTG_EN_PWROPT = 2.
9	W1C	0x0	ResetDetected This field indicates that Reset has been detected by the PMU module. This field generates an interrupt. 1'b0: Reset Not Detected 1'b1: Reset Detected This bit is valid only when OTG_EN_PWROPT = 2.
8	RW	0x0	LineStageChangeMsk Mask For LineStateChange interrupt This bit is valid only when OTG_EN_PWROPT = 2.
7	W1C	0x0	LnStsChng Line State Change This interrupt is asserted when there is a Linestate Change detected by the PMU. The application should read GPWRDN.Linestate to determine the current linestate on USB. 1'b0: No LineState change on USB 1'b1: LineState change on USB This bit is valid only when GPWRDN.PMUActv is 1.This bit is valid only when OTG_EN_PWROPT = 2.
6	RW	0x0	DisableVBUS The application should program this bit if HPRT0.PrtPwr was programmed to 0 before entering Hibernation. This is to indicate PMU whether session was ended before entering Hibernation. 1'b0: HPRT0.PrtPwr was not programmed to 0. 1'b1: HPRT0.PrtPwr was programmed to 0.

Bit	Attr	Reset Value	Description
5	RW	0x0	PwrDnSwtch Power Down Switch This bit indicates to the VDD switch is in ON/OFF state 1'b0: ON state 1'b1: OFF state *Note: This bit must not be written to during normal mode of operation.*
4	RW	0x0	PwrDnRst_n Power Down ResetN The application must program this bit to reset the OTG core during the Hibernation exit process or during ADP when powering up the core (in case the OTG core was powered off during ADP process). 1'b1: in normal operation 1'b0: reset *Note: This bit must not be written to during normal mode of operation.*
3	RW	0x0	PwrDnClmp Power Down Clamp The application must program this bit to enable or disable the clamps to all the outputs of the OTG core module to prevent the corruption of other active logic. 1'b0: Disable PMU power clamp 1'b1: Enable PMU power clamp
2	RW	0x0	Restore The application should program this bit to enable or disable restore mode from the PMU module. 1'b0: in normal mode of operation 1'b1: in restore mode *Note: This bit must not be written to during normal mode of operation. This bit is valid only when OTG_EN_PWROPT = 2.*
1	RW	0x0	PMUActv PMU Active This is bit is to enable or disable the PMU logic. 1'b0: Disable PMU module 1'b1: Enable PMU module *Note: This bit must not be written to during normal mode of operation.*

Bit	Attr	Reset Value	Description
0	RW	0x0	PMUIntSel PMU Interrupt Select When the hibernation functionality is selected using the configuration option OTG_EN_PWR_OPT = 2, a write to this bit with 1'b1 enables the PMU to generate interrupts to the application. During this state all interrupts from the core module are blocked to the application. Note: This bit must be set to 1'b1 before the core is put into hibernation 1'b0: Internal interrupt is selected 1'b1: the external OTG PMU interrupt is selected Note: This bit must not be written to during normal mode of operation.

USBOTG_GDFIFOCFG
Address: Operational Base + offset (0x005c)
Global DFIFO Software Config Register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	EPInfoBaseAddr This field provides the start address of the EP info controller.
15:0	RW	0x0000	GDFIFOCfg This field is for dynamic programming of the DFIFO Size. This value takes effect only when the application programs a non-zero value to this register. The core does not have any corrective logic if the FIFO sizes are programmed incorrectly.

USBOTG_GADPCTL
Address: Operational Base + offset (0x0060)
ADP Timer, Control and Status Register

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28:27	R/W SC	0x0	AR Access Request 2'b00 Read/Write Valid (updated by the core) 2'b01 Read 2'b10 Write 2'b11 Reserved
26	RW	0x0	AdpTmoutMsk ADP Timeout Interrupt Mask When this bit is set, it unmasks the interrupt because of AdpTmoutInt. This bit is valid only if OTG_Ver = 1'b1(GOTGCTL[20]).

Bit	Attr	Reset Value	Description
25	RW	0x0	AdpSnsIntMsk ADP Sense Interrupt Mask When this bit is set, it unmasks the interrupt due to AdpSnsInt. This bit is valid only if OTG_Ver = 1'b1(GOTGCTL[20]).
24	RW	0x0	AdpPrbIntMsk ADP Probe Interrupt Mask When this bit is set, it unmasks the interrupt due to AdpPrbInt. This bit is valid only if OTG_Ver = 1'b1(GOTGCTL[20]).
23	W1C	0x0	AdpTmoutInt ADP Timeout Interrupt This bit is relevant only for an ADP probe. When this bit is set, it means that the ramp time has completed (GADPCTL.RTIM has reached its terminal value of 0x7FF). This is a debug feature that allows software to read the ramp time after each cycle. This bit is valid only if OTG_Ver = 1'b1.
22	W1C	0x0	AdpSnsInt ADP Sense Interrupt When this bit is set, it means that the VBUS voltage is greater than VadpSns value or VadpSns is reached. This bit is valid only if OTG_Ver = 1'b1 (GOTGCTL[20]).
21	W1C	0x0	AdpPrbInt ADP Probe Interrupt When this bit is set, it means that the VBUS voltage is greater than VadpPrb or VadpPrb is reached. This bit is valid only if OTG_Ver = 1'b1 (GOTGCTL[20]).
20	RW	0x0	ADPEn ADP Enable When set, the core performs either ADP probing or sensing based on EnaPrb or EnaSns. This bit is valid only if OTG_Ver = 1'b1 (GOTGCTL[20]).
19	R/W SC	0x0	ADPRes ADP Reset When set, ADP controller is reset. This bit is auto-cleared after the reset procedure is complete in ADP controller. This bit is valid only if OTG_Ver = 1'b1 (GOTGCTL[20]).
18	RW	0x0	EnaSns Enable Sense When programmed to 1'b1, the core performs a sense operation. This bit is valid only if OTG_Ver = 1'b1 (GOTGCTL[20]).
17	RW	0x0	EnaPrb Enable Probe When programmed to 1'b1, the core performs a probe operation. This bit is valid only if OTG_Ver = 1'b1 (GOTGCTL[20]).

Bit	Attr	Reset Value	Description
16:6	RO	0x000	RTIM RAMP TIME These bits capture the latest time it took for VBUS to ramp from VADP_SINK to VADP_PRB. The bits are defined in units of 32 kHz clock cycles as follows: 0x000 - 1 cycles 0x001 - 2 cycles 0x002 - 3 cycles and so on till 0x7FF - 2048 cycles A time of 1024 cycles at 32 kHz corresponds to a time of 32 msec. (Note for scaledown ramp_timeout = prb_delta == 2'b00 => 200 cycles prb_delta == 2'b01 => 100 cycles prb_delta == 2'b01 => 50 cycles prb_delta == 2'b01 => 25 cycles.)
5:4	RW	0x0	PrbPer Probe Period These bits sets the TadpPrd as follows: 2'b00 - 0.625 to 0.925 sec (typical 0.775 sec) 2'b01 - 1.25 to 1.85 sec (typical 1.55 sec) 2'b10 - 1.9 to 2.6 sec (typical 2.275 sec) 2'b11 - Reserved (PRB_PER is also scaledown prb_per== 2'b00 => 400 ADP clocks prb_per== 2'b01 => 600 ADP clocks prb_per== 2'b10 => 800 ADP clocks prb_per==2'b11 => 1000 ADP clocks)
3:2	RW	0x0	PrbDelta Probe Delta These bits set the resolution for RTIM value. The bits are defined in units of 32 kHz clock cycles as follows: 2'b00 - 1 cycles 2'b01 - 2 cycles 2'b10 - 3 cycles 2'b11 - 4 cycles For example if this value is chosen to 2'b01, it means that RTIM increments forevery three 32Khz clock cycles.

Bit	Attr	Reset Value	Description
1:0	RW	0x0	PrbDschg Probe Discharge These bits set the times for TadpDschg. These bits are defined as follows: 2'b00 4 msec (Scaledown 2 32Khz clock cycles) 2'b01 8 msec (Scaledown 4 32Khz clock cycles) 2'b10 16 msec (Scaledown 8 32Khz clock cycles) 2'b11 32 msec (Scaledown 16 32Khz clock cycles)

USBOTG_HPTXFSIZ
Address: Operational Base + offset (0x0100)
Host Periodic Transmit FIFO Size Register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	PTxFSize Host Periodic TxFIFO Depth This value is in terms of 32-bit words. Minimum value is 16 Maximum value is 32,768 The power-on reset value of this register is specified as the Largest Host Mode Periodic Tx Data FIFO Depth (parameter OTG_TX_HPERIO_DFIFO_DEPTH). If Enable Dynamic FIFO Sizing was deselected (parameter OTG_DFIFO_DYNAMIC = 0), these flops are optimized, and reads return the power-on value. If Enable Dynamic FIFO Sizing was selected (parameter OTG_DFIFO_DYNAMIC = 1), you can write a new value in this field. Programmed values must not exceed the power-on value set.
15:0	RW	0x0000	PTxFStAddr Host Periodic TxFIFO Start Address The power-on reset value of this register is the sum of the Largest Rx Data FIFO Depth and Largest Non-periodic Tx Data FIFO Depth specified. These parameters are: In shared FIFO operation: OTG_RX_DFIFO_DEPTH + OTG_TX_NPERIO_DFIFO_DEPTH. In dedicated FIFO mode: OTG_RX_DFIFO_DEPTH + OTG_TX_HNPERIO_DFIFO_DEPTH. If Enable Dynamic FIFO Sizing was deselected (parameter OTG_DFIFO_DYNAMIC = 0), these flops are optimized, and reads return the power-on value. If Enable Dynamic FIFO Sizing was selected(parameter OTG_DFIFO_DYNAMIC = 1), you can write a new value in this field. Programmed values must not exceed the power-on value.

USBOTG_DIEPTXFn

Address: Operational Base + offset (0x0104)
Device Periodic Transmit FIFO-n Size Register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	INEP1TxFDep IN Endpoint TxFIFO Depth This value is in terms of 32-bit words. Minimum value is 16 Maximum value is 32,768 The power-on reset value of this register is specified as the Largest IN Endpoint FIFO number Depth (parameter OTG_TX_DINEP_DFIFO_DEPTH_n)(0 < n <= 15). If Enable Dynamic FIFO Sizing was deselected (parameter OTG_DFIFO_DYNAMIC = 0), these flops are optimized, and reads return the power-on value. If Enable Dynamic FIFO Sizing was selected (parameter OTG_DFIFO_DYNAMIC = 1), you can write a new value in this field. Programmed values must not exceed the power-on value.
15:0	RW	0x0000	INEP1TxFStAddr IN Endpoint FIFO1 Transmit RAM Start Address This field contains the memory start address for IN endpoint Transmit FIFOn (0 < n <= 15). The power-on reset value of this register is specified as the Largest Rx Data FIFO Depth (parameter OTG_RX_DFIFO_DEPTH). OTG_RX_DFIFO_DEPTH + SUM 0 to n-1 (OTG_DINEP_TXFIFO_DEPTH_n) For example start address of IN endpoint FIFO 1 is OTG_RX_DFIFO_DEPTH + OTG_DINEP_TXFIFO_DEPTH_0. The start address of IN endpoint FIFO 2 is OTG_RX_DFIFO_DEPTH + OTG_DINEP_TXFIFO_DEPTH_0 + OTG_DINEP_TXFIFO_DEPTH_1. If Enable Dynamic FIFO Sizing? was deselected (parameter OTG_DFIFO_DYNAMIC = 0), these flops are optimized, and reads return the power-on value. If Enable Dynamic FIFO Sizing was selected (parameter OTG_DFIFO_DYNAMIC = 1), and you have programmed a new value for RxFIFO depth, you can write that value in this field. Programmed values must not exceed the power-on value set.

USBOTG_HCFG

Address: Operational Base + offset (0x0400)
Host Configuration Register

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved

Bit	Attr	Reset Value	Description
26	RW	0x0	PerSchedEna Enable Periodic Scheduling Applicable in Scatter/Gather DMA mode only. Enables periodic scheduling within the core. Initially, the bit is reset. The core will not process any periodic channels. As soon as this bit is set, the core will get ready to start scheduling periodic channels and sets HCFG.PerSchedStat. The setting of HCFG.PerSchedStat indicates the core has enabled periodic scheduling. Once HCFG.PerSchedEna is set, the application is not supposed to again reset the bit unless HCFG.PerSchedStat is set. As soon as this bit is reset, the core will get ready to stop scheduling periodic channels and resets HCFG.PerSchedStat. In non Scatter/Gather DMA mode, this bit is reserved.
25:24	RW	0x0	FrListEn Frame List Entries The value in the register specifies the number of entries in the Frame list. This field is valid only in Scatter/Gather DMA mode.

23	RW	0x0	DescDMA
			Enable Scatter/gather DMA in Host mode
			When the Scatter/Gather DMA option selected during configuration of the RTL, the application can set this bit during initialization to enable the Scatter/Gather DMA operation. NOTE: This bit must be modified only once after a reset. The following combinations are available for programming:
			GAHBCFG.DMAEn=0, HCFG.DescDMA=0 => Slave mode
			GAHBCFG.DMAEn=0, HCFG.DescDMA=1 => Invalid
			GAHBCFG.DMAEn=1, HCFG.DescDMA=0 => Buffered DMA mode
			GAHBCFG.DMAEn=1, HCFG.DescDMA=1 => Scatter/Gather DMA mode
			In non-Scatter/Gather DMA mode, this bit is reserved.

22:16	RO	0x0	reserved
15:8	RW	0x00	ResValid Resume Validation Period This field is effective only when HCFG.Ena32KHzS is set. It controls the resume period when the core resumes from suspend. The core counts the ResValid number of clock cycles to detect a valid resume when this is set.
7	RW	0x0	Ena32KHzS Enable 32-KHz Suspend Mode This bit can only be set if the USB 1.1 Full-Speed Serial Transceiver Interface has been selected. If USB 1.1 Full-Speed Serial Transceiver Interface has not been selected, this bit must be zero. When the USB 1.1 Full-Speed Serial Transceiver Interface is chosen and this bit is set, the core expects the 48-MHz PHY clock to be switched to 32 KHz during a suspend.

Bit	Attr	Reset Value	Description
6:3	RO	0x0	reserved
2	RW	0x0	FSLSSupp FS- and LS-Only Support The application uses this bit to control the core enumeration speed. Using this bit, the application can make the core enumerate as a FS host, even if the connected device supports HS traffic. Do not make changes to this field after initial programming. 1'b0: HS/FS/LS, based on the maximum speed supported by the connected device 1'b1: FS/LS-only, even if the connected device can support HS
1:0	RW	0x0	FSLSPclkSel FS/LS PHY Clock Select 2'b00: PHY clock is running at 30/60 MHz 2'b01: PHY clock is running at 48 MHz Others: Reserved

USBOTG_HFIR
Address: Operational Base + offset (0x0404)
Host Frame Interval Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	FrInt Frame Interval The value that the application programs to this field specifies the interval between two consecutive SOFs (FS) or micro-SOFs (HS) or Keep-Alive tokens (HS). This field contains the number of PHY clocks that constitute the required frame interval. The default value set in this field for a FS operation when the PHY clock frequency is 60MHz. The application can write a value to this register only after the Port Enable bit of the Host Port Control and Status register (HPRT.PrtEnaPort) has been set. If no value is programmed, the core calculates the value based on the PHY clock specified in the FS/LS PHY Clock Select field of the Host Configuration register (HCFG.FSLSPclkSel). Do not change the value of this field after the initial configuration. 125 us * (PHY clock frequency for HS) 1 ms * (PHY clock frequency for FS/LS)

USBOTG_HFNUM
Address: Operational Base + offset (0x0408)
Host Frame Number/Frame Time Remaining Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	FrRem Frame Time Remaining Indicates the amount of time remaining in the current microframe (HS) or frame (FS/LS), in terms of PHY clocks. This field decrement on each PHY clock. When it reaches zero, this field is reloaded with the value in the Frame Interval register and a new SOF is transmitted on the USB.
15:0	RO	0xffff	FrNum Frame Number This field increment when a new SOF is transmitted on the USB, and is reset to 0 when it reaches 16'h3FFF. This field is writable only if Remove Optional Features was not selected (OTG_RM_OTG_FEATURES = 0). Otherwise, reads return the frame number value.

USBOTG_HPTXSTS
Address: Operational Base + offset (0x0410)
Host Periodic Transmit FIFO/Queue Status Register

Bit	Attr	Reset Value	Description
31:24	RO	0x00	PTxQTop Top of the Periodic Transmit Request Queue This indicates the entry in the Periodic Tx Request Queue that is currently being processes by the MAC. This register is used for debugging. Bit [31]: Odd/Even (micro)frame 1'b0: send in even (micro)frame 1'b1: send in odd (micro)frame Bits [30:27]: Channel/endpoint number Bits [26:25]: Type 2'b00: IN/OUT 2'b01: Zero-length packet 2'b10: CSPLIT 2'b11: Disable channel command Bit [24]: Terminate (last entry for the selected channel/endpoint)
23:16	RO	0x00	PTxQSpcAvail Periodic Transmit Request Queue Space Available Indicates the number of free locations available to be written in the Periodic Transmit Request Queue. This queue holds both IN and OUT requests. 8'h0: Periodic Transmit Request Queue is full 8'h1: 1 location available 8'h2: 2 locations available n: n locations available (0 <=n <= 16) Others: Reserved

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	PTxFSpcAvail Periodic Transmit Data FIFO Space Available Indicates the number of free locations available to be written to in the Periodic TxFIFO. Values are in terms of 32-bit words 16'h0: Periodic TxFIFO is full 16'h1: 1 word available 16'h2: 2 words available 16'hn: n words available (where 0 . n . 32,768) 16'h8000: 32,768 words available Others: Reserved

USBOTG_HAINT
Address: Operational Base + offset (0x0414)
Host All Channels Interrupt Reigster

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RO	0x0000	HAINT Channel Interrupts One bit per channel: Bit 0 for Channel 0, bit 15 for Channel 15

USBOTG_HAINTMSK
Address: Operational Base + offset (0x0418)
Host All Channels Interrupt Mask Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	HAINTMsk Channel Interrupt Mask One bit per channel: Bit 0 for channel 0, bit 15 for channel 15

USBOTG_HPRT
Address: Operational Base + offset (0x0440)
Host Port Control and Status Register

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:17	RO	0x0	PrtSpd Port Speed Indicates the speed of the device attached to this port. 2'b00: High speed 2'b01: Full speed 2'b10: Low speed 2'b11: Reserved

Bit	Attr	Reset Value	Description
16:13	RW	0x0	PrtTstCtl Port Test Control The application writes a nonzero value to this field to put the port into a Test mode,and the corresponding pattern is signaled on the port. 4'b0000: Test mode disabled 4'b0001: Test_J mode 4'b0010: Test_K mode 4'b0011: Test_SE0_NAK mode 4'b0100: Test_Packet mode 4'b0101: Test_Force_Enable Others: Reserved
12	R/W SC	0x0	PrtPwr Port Power The application uses this field to control power to this port (write 1'b1 to set to 1'b1and write 1'b0 to set to 1'b0), and the core can clear this bit on an over current condition. 1'b0: Power off 1'b1: Power on
11:10	RO	0x0	PrtLnSts Port Line Status Indicates the current logic level USB data lines Bit [10]: Logic level of D+ Bit [11]: Logic level of D
9	RO	0x0	reserved
8	RW	0x0	PrtRst Port Reset When the application sets this bit, a reset sequence is started on this port. The application must time the reset period and clear this bit after the reset sequence is complete. 1'b0: Port not in reset 1'b1: Port in reset To start a reset on the port, the application must leave this bit set for at least the minimum duration mentioned below, as specified in the USB 2.0 specification, Section 7.1.7.5. The application can leave it set for another 10 ms in addition to the required minimum duration, before clearing the bit, even though there is no maximum limit set by the USB standard. High speed: 50 ms Full speed/Low speed: 10 ms

Bit	Attr	Reset Value	Description
7	R/W SC	0x0	PrtSusp Port Suspend The application sets this bit to put this port in Suspend mode. The core only stops sending SOFs when this is set. To stop the PHY clock, the application must set the Port Clock Stop bit, which asserts suspend input pin of the PHY. The read value of this bit reflects the current suspend status of the port. This bit is cleared by the core after a remote wakeup signal is detected or the application sets the Port Reset bit or Port Resume bit in this register or the Resume/Remote Wakeup Detected Interrupt bit or Disconnect Detected Interrupt bit in the Core Interrupt register (GINTSTS.WkUpInt or GINTSTS.DisconnInt, respectively). 1'b0: Port not in Suspend mode 1'b1: Port in Suspend mode
6	R/W SC	0x0	PrtRes Port Resume The application sets this bit to drive resume signaling on the port. The core continues to drive the resume signal until the application clears this bit. If the core detects a USB remote wakeup sequence, as indicated by the Port Resume/Remote Wakeup Detected Interrupt bit of the Core Interrupt register (GINTSTS.WkUpInt), the core starts driving resume signaling without application intervention and clears this bit when it detects a disconnect condition. The read value of this bit indicates whether the core is currently driving resume signaling. 1'b0: No resume driven 1'b1: Resume driven When LPM is enabled and the core is in the L1 (Sleep) state, setting this bit results in the following behavior: The core continues to drive the resume signal until a pre-determined time specified in the GLPMCFG.HIRD_Thres[3:0] field. If the core detects a USB remote wakeup sequence, as indicated by the Port L1 Resume/Remote L1 Wakeup Detected Interrupt bit of the Core Interrupt register (GINTSTS.L1WkUpInt), the core starts driving resume signaling without application intervention and clears this bit at the end of the resume. The read value of this bit indicates whether the core is currently driving resume signaling. 1'b0: No resume driven 1'b1: Resume driven

Bit	Attr	Reset Value	Description
5	W1C	0x0	PrtOvrCurrChng Port Overcurrent Change The core sets this bit when the status of the Port Over-current Active bit (bit 4) in this register changes.
4	RO	0x0	PrtOvrCurrAct Port Overcurrent Active Indicates the overcurrent condition of the port. 1'b0: No overcurrent condition 1'b1: Overcurrent condition
3	W1C	0x0	PrtEnChng Port Enable/Disable Change The core sets this bit when the status of the Port_Enable bit[2] of this register changes.
2	W1C	0x0	PrtEna Port Enable A port is enabled only by the core after a reset sequence, and is disabled by an over-current condition, a disconnect condition, or by the application clearing this bit. The application cannot set this bit by a register write. It can only clear it to disable the port. This bit does not trigger any interrupt to the application. 1'b0: Port disabled 1'b1: Port enabled
1	W1C	0x0	PrtConnDet Port Connect Detected The core sets this bit when a device connection is detected to trigger an interrupt to the application using the Host Port Interrupt bit of the Core Interrupt register (GINTSTS.PrtInt). The application must write a 1 to this bit to clear the interrupt.
0	RO	0x0	PrtConnSts Port Connect Status 0: No device is attached to the port. 1: A device is attached to the port.

USBOTG_HCCHARn
Address: Operational Base + offset (0x0500)
Host Channel-n Characteristics Register

Bit	Attr	Reset Value	Description
31	R/W SC	0x0	ChEna Channel Enable When Scatter/Gather mode is enabled 1'b0: Indicates that the descriptor structure is not yet ready. 1'b1: Indicates that the descriptor structure and data buffer with data is setup and this channel can access the descriptor. When Scatter/Gather mode is disabled, This field is set by the application and cleared by the OTG host. 1'b0: Channel disabled 1'b1: Channel enabled
30	R/W SC	0x0	ChDis Channel Disable The application sets this bit to stop transmitting/receiving data on a channel, even before the transfer for that channel is complete. The application must wait for the Channel Disabled interrupt before treating the channel as disabled.
29	RW	0x0	OddFrm Odd Frame This field is set (reset) by the application to indicate that the OTG host must perform a transfer in an odd (micro) frame. This field is applicable for only periodic (isochronous and interrupt) transactions. 1'b0: Even (micro)frame 1'b1: Odd (micro)frame This field is not applicable for Scatter/Gather DMA mode and need not be programmed by the application and is ignored by the core.
28:22	RW	0x00	DevAddr Device Address This field selects the specific device serving as the data source or sink.

Bit	Attr	Reset Value	Description
21:20	RW	0x0	MC_EC Multi Count (MC) / Error Count (EC) When the Split Enable bit of the Host Channel-n Split Control register (HCSPLTn.SpltEna) is reset (1'b0), this field indicates to the host the number of transactions that must be executed per microframe for this periodic endpoint. For non-periodic transfers, this field is used only in DMA mode, and specifies the number packets to be fetched for this channel before the internal DMA engine changes arbitration. 2'b00: Reserved This field yields undefined results. 2'b01: 1 transaction 2'b10: 2 transactions to be issued for this endpoint per micro-frame 2'b11: 3 transactions to be issued for this endpoint per micro-frame When HCSPLTn.SpltEna is set (1'b1), this field indicates the number of immediate retries to be performed for a periodic split transactions on transaction errors. This field must be set to at least 2'b01.
19:18	RW	0x0	EPType Endpoint Type Indicates the transfer type selected. 2'b00: Control 2'b01: Isochronous 2'b10: Bulk 2'b11: Interrupt
17	RW	0x0	LSpdDev Low-Speed Device This field is set by the application to indicate that this channel is communicating to a low-speed device.
16	RO	0x0	reserved
15	RW	0x0	EPDir Endpoint Direction Indicates whether the transaction is IN or OUT. 1'b0: OUT 1'b1: IN
14:11	RW	0x0	EPNum Endpoint Number Indicates the endpoint number on the device serving as the data source or sink.
10:0	RW	0x000	MPS Maximum Packet Size Indicates the maximum packet size of the associated endpoint.

USBOTG_HCSPLTn

Address: Operational Base + offset (0x0504)
Host Channel-n Split Control Register

Bit	Attr	Reset Value	Description
31	RW	0x0	SpltEna Split Enable The application sets this field to indicate that this channel is enabled to perform split transactions.
30:17	RO	0x0	reserved
16	RW	0x0	CompSplt Do Complete Split The application sets this field to request the OTG host to perform a complete split transaction.
15:14	RW	0x0	XactPos Transaction Position This field is used to determine whether to send all, first, middle, or last payloads with each OUT transaction. 2'b11: All. This is the entire data payload is of this transaction (which is less than or equal to 188 bytes). 2'b10: Begin. This is the first data payload of this transaction (which is larger than 188 bytes). 2'b00: Mid. This is the middle payload of this transaction (which is larger than 188bytes). 2'b01: End. This is the last payload of this transaction (which is larger than 188 bytes).
13:7	RW	0x00	HubAddr Hub Address This field holds the device address of the transaction translator's hub.
6:1	RO	0x0	reserved
0	RW	0x0	PrtAddr Port Address This field is the port number of the recipient transaction translator.

USBOTG_HCINTn
Address: Operational Base + offset (0x0508)
Host Channel-n Interrupt Register

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
13	W1C	0x0	DESC_LST_ROLLIntr Descriptor rollover interrupt This bit is valid only when Scatter/Gather DMA mode is enabled. The core sets this bit when the corresponding channel's descriptor list rolls over. For non-Scatter/Gather DMA mode, this bit is reserved.

Bit	Attr	Reset Value	Description
12	W1C	0x0	XCS_XACT_ERR Excessive Transaction Error This bit is valid only when Scatter/Gather DMA mode is enabled. The core sets this bit when 3 consecutive transaction errors occurred on the USB bus. XCS_XACT_ERR will not be generated for Isochronous channels. For non-Scatter/Gather DMA mode, this bit is reserved.
11	W1C	0x0	BNAIntr BNA (Buffer Not Available) Interrupt This bit is valid only when Scatter/Gather DMA mode is enabled. The core generates this interrupt when the descriptor accessed is not ready for the Core to process. BNA will not be generated for Isochronous channels. For non-Scatter/Gather DMA mode, this bit is reserved.
10	W1C	0x0	DataTglErr Data Toggle Error In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core.
9	W1C	0x0	FrmOvrun Frame Overrun In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core
8	W1C	0x0	BblErr Babble Error In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core.
7	W1C	0x0	XactErr Transaction Error Indicates one of the following errors occurred on the USB: CRC check failure, Timeout, Bit stuff error, False EOP. In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core.
6	WO	0x0	NYET NYET Response Received Interrupt In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core.
5	W1C	0x0	ACK ACK Response Received/Transmitted Interrupt In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core.
4	W1C	0x0	NAK NAK Response Received Interrupt In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core.

Bit	Attr	Reset Value	Description
3	W1C	0x0	STALL STALL Response Received Interrupt In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core.
2	W1C	0x0	AHBErr AHB Error This is generated only in DMA mode when there is an AHB error during AHB read/write. The application can read the corresponding channel's DMA address register to get the error address.
1	W1C	0x0	ChHltd Channel Halted In non-Scatter/Gather DMA mode, it indicates the transfer completed abnormally either because of any USB transaction error or in response to disable request by the application or because of a completed transfer. In Scatter/Gather DMA mode, this indicates that transfer completed due to any of the following: EOL being set in descriptor, AHB error, Excessive transaction errors, In response to disable request by the application, Babble, Stall, Buffer Not Available (BNA)
0	W1C	0x0	XferCompl Transfer Completed For Scatter/Gather DMA mode, it indicates that current descriptor processing got completed with IOC bit set in its descriptor. In non-Scatter/Gather DMA mode, it indicates that Transfer completed normally without any errors.

USBOTG_HCINTMSKn
Address: Operational Base + offset (0x050c)
Host Channel-n Interrupt Mask Register

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
13	RW	0x0	DESC_LST_ROLLIntrMsk Descriptor rollover interrupt Mask register This bit is valid only when Scatter/Gather DMA mode is enabled. In non-Scatter/Gather DMA mode, this bit is reserved.
12	RO	0x0	reserved
11	RW	0x0	BNAIntrMsk BNA (Buffer Not Available) Interrupt mask register This bit is valid only when Scatter/Gather DMA mode is enabled. In non-Scatter/Gather DMA mode, this bit is reserved.

Bit	Attr	Reset Value	Description
10	RW	0x0	DataTglErrMsk Data Toggle Error Mask This bit is not applicable in Scatter/Gather DMA mode.
9	RW	0x0	FrmOvrunMsk Frame Overrun Mask This bit is not applicable in Scatter/Gather DMA mode.
8	RW	0x0	BblErrMsk Babble Error Mask This bit is not applicable in Scatter/Gather DMA mode.
7	RW	0x0	XactErrMsk Transaction Error Mask This bit is not applicable in Scatter/Gather DMA mode
6	RW	0x0	NyetMsk NYET Response Received Interrupt Mask This bit is not applicable in Scatter/Gather DMA mode.
5	RW	0x0	AckMsk ACK Response Received/Transmitted Interrupt Mask This bit is not applicable in Scatter/Gather DMA mode.
4	RW	0x0	NakMsk NAK Response Received Interrupt Mask This bit is not applicable in Scatter/Gather DMA mode.
3	RW	0x0	StallMsk STALL Response Received Interrupt Mask This bit is not applicable in Scatter/Gather DMA mode.
2	RW	0x0	AHBErrMsk AHB Error Mask Note: This bit is only accessible when OTG_ARCHITECTURE = 2
1	RW	0x0	ChHltdMsk Channel Halted Mask
0	RW	0x0	XferComplMsk Transfer Completed Mask This bit is valid only when Scatter/Gather DMA mode is enabled. In non-Scatter/Gather DMA mode, this bit is reserved.

USBOTG_HCTSIZn
Address: Operational Base + offset (0x0510)
Host Channel-n Transfer Size Register

Bit	Attr	Reset Value	Description
31	RW	0x0	DoPng Do Ping This bit is used only for OUT transfers. Setting this field to 1 directs the host to do PING protocol. Note: Do not set this bit for IN transfers. If this bit is set for IN transfers it disables the channel.

Bit	Attr	Reset Value	Description
30:29	RW	0x0	Pid PID The application programs this field with the type of PID to use for the initial transaction. The host maintains this field for the rest of the transfer. 2'b00: DATA0 2'b01: DATA2 2'b10: DATA1 2'b11: MDATA (non-control)/SETUP (control)
28:19	RW	0x000	PktCnt Packet Count This field is programmed by the application with the expected number of packets to be transmitted (OUT) or received (IN).The host decrements this count on every successful transmission or reception of an OUT/IN packet. Once this count reaches zero, the application is interrupted to indicate normal completion. The width of this counter is specified as Width of Packet Counters (parameter OTG_PACKET_COUNT_WIDTH).
18:0	RW	0x00000	XferSize Transfer Size For an OUT, this field is the number of data bytes the host sends during the transfer. For an IN, this field is the buffer size that the application has Reserved for the transfer. The application is expected to program this field as an integer multiple of the maximum packet size for IN transactions (periodic and non-periodic). The width of this counter is specified as Width of Transfer Size Counters (parameter OTG_TRANS_COUNT_WIDTH).

USBOTG_HCDMAn
Address: Operational Base + offset (0x0514)
Host Channel-n DMA Address Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMAAddr DMA Address This field holds the start address in the external memory from which the data for the endpoint must be fetched or to which it must be stored. This register is incremented on every AHB transaction.

USBOTG_HCDMABn
Address: Operational Base + offset (0x051c)
Host Channel-n DMA Buffer Address Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	HCDMABn Holds the current buffer address This register is updated as and when the data transfer for the corresponding end point is in progress. This register is present only in Scatter/Gather DMA mode. Otherwise this field is reserved.

USBOTG_DCFG
Address: Operational Base + offset (0x0800)
Device Cconfiguration Register

Bit	Attr	Reset Value	Description
31:26	RW	0x02	ResValid Resume Validation Period This field controls the period when the core resumes from a suspended. When this bit is set, the core counts for the Resume Valid number of clock cycles to detect a valid resume. This field is effective only when DCFG.Ena32KHzSusp is set.
25:24	RW	0x0	PerSchIntvl Periodic Scheduling Interval PerSchIntvl must be programmed only for Scatter/Gather DMA mode. Description: This field specifies the amount of time the Internal DMA engine must allocate for fetching periodic IN endpoint data. Based on the number of periodic endpoints, this value must be specified as 25, 50 or 75% of (micro) frame. When any periodic endpoints are active, the internal DMA engine allocates the specified amount of time in fetching periodic IN endpoint data. When no periodic endpoints are active, then the internal DMA engine services non-periodic endpoints, ignoring this field. After the specified time within a (micro) frame, the DMA switches to fetching for non-periodic endpoints. 2'b00: 25% of (micro) frame. 2'b01: 50% of (micro) frame. 2'b10: 75% of (micro) frame. 2'b11: Reserved.

Bit	Attr	Reset Value	Description

23	RW	0x0	DescDMA
			Enable Scatter/Gather DMA in Device mode
			When the Scatter/Gather DMA option selected during configuration of the RTL, the application can set this bit during initialization to enable the Scatter/Gather DMA operation. NOTE: This bit must be modified only once after a reset. The following combinations are available for programming:
			GAHBCFG.DMAEn=0,DCFG.DescDMA=0 => Slave mode
			GAHBCFG.DMAEn=0,DCFG.DescDMA=1 => Invalid
			GAHBCFG.DMAEn=1,DCFG.DescDMA=0 => Buffered DMA mode
			GAHBCFG.DMAEn=1,DCFG.DescDMA=1 => Scatter/Gather DMA mode

22:18	RW	0x08	EPMisCnt IN Endpoint Mismatch Count This field is valid only in shared FIFO operation. The application programs this filed with a count that determines when the core generates an Endpoint Mismatch interrupt (GINTSTS.EPMis). The core loads this value into an internal counter and decrements it. The counter is reloaded whenever there is a match or when the counter expires. The width of this counter depends on the depth of the Token Queue.
17:13	RO	0x0	reserved
12:11	RW	0x0	PerFrInt Periodic Frame Interval Indicates the time within a (micro) frame at which the application must be notified using the End Of Periodic Frame Interrupt. This can be used to determine if all the isochronous traffic for that (micro) frame is complete. 2'b00: 80% of the (micro) frame interval 2'b01: 85% 2'b10: 90% 2'b11: 95%
10:4	RW	0x00	DevAddr Device Address The application must program this field after every SetAddress control command.
3	RW	0x0	Ena32KHzS Enable 32-KHz Suspend Mode When the USB 1.1 Full-Speed Serial Transceiver Interface is chosen and this bit is set, the core expects the 48-MHz PHY clock to be switched to 32 KHz during a suspend. This bit can only be set if USB 1.1 Full-Speed Serial Transceiver Interface has been selected. If USB 1.1 Full-Speed Serial Transceiver Interface has not been selected, this bit must be zero.

Bit	Attr	Reset Value	Description
2	RW	0x0	NZStsOUTHShk Non-Zero-Length Status OUT Handshake The application can use this field to select the handshake the core sends on receiving a nonzero-length data packet during the OUT transaction of a control transfer's Status stage. 1'b1: Send a STALL handshake on a nonzero-length status OUT transaction and do not send the received OUT packet to the application. 1'b0: Send the received OUT packet to the application (zero-length or non-zero length) and send a handshake based on the NAK and STALL bits for the endpoint in the Device Endpoint Control register.
1:0	RW	0x0	DevSpd Device Speed Indicates the speed at which the application requires the core to enumerate, or the maximum speed the application can support. However, the actual bus speed is determined only after the chirp sequence is completed, and is based on the speed of the USB host to which the core is connected. 2'b00: High speed (USB 2.0 PHY clock is 30 MHz or 60 MHz) 2'b01: Full speed (USB 2.0 PHY clock is 30 MHz or 60 MHz) 2'b10: Reserved 2'b11: Full speed (USB 1.1 transceiver clock is 48 MHz)

USBOTG_DCTL
Address: Operational Base + offset (0x0804)
Device Control Register

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16	RW	0x0	NakOnBble Set NAK automatically on babble The core sets NAK automatically for the endpoint on which babble is received.

Bit	Attr	Reset Value	Description
15	RW	0x0	IgnrFrmNum Ignore frame number for isochronous endpoints in case of Scatter Do NOT program IgnrFrmNum bit to 1'b1 when the core is operating in Threshold mode. Note: When Scatter/Gather DMA mode is enabled this feature is not applicable to high speed, high-bandwidth transfers. When this bit is enabled, there must be only one packet per descriptor. 0: The core transmits the packets only in the frame number in which they are intended to be transmitted. 1: The core ignores the frame number, sending packets immediately as the packets are ready. Scatter/Gather: In Scatter/Gather DMA mode, when this bit is enabled, the packets are not flushed when an ISOC IN token is received for an elapsed frame. When Scatter/Gather DMA mode is disabled, this field is used by the application to enable periodic transfer interrupt. The application can program periodic endpoint transfers for multiple micro) frames. 0: Periodic transfer interrupt feature is disabled; the application must program transfers for periodic endpoints every (micro)frame 1: Periodic transfer interrupt feature is enabled; the application can program transfers for multiple (micro) frames for periodic endpoints. In non-Scatter/Gather DMA mode, the application receives transfer complete interrupt after transfers for multiple (micro) frames are completed.
14:13	RW	0x1	GMC Global Multi Count GMC must be programmed only once after initialization. Applicable only for Scatter/Gather DMA mode. This indicates the number of packets to be serviced for that end point before moving to the next end point. It is only for non-periodic endpoints. 2'b00: Invalid. 2'b01: 1 packet. 2'b10: 2 packets. 2'b11: 3 packets. When Scatter/Gather DMA mode is disabled, this field is reserved, and reads 2'b00.
12	RO	0x0	reserved

Bit	Attr	Reset Value	Description
11	RW	0x0	PWROnPrgDone Power-On Programming Done The application uses this bit to indicate that register programming is completed after a wake-up from Power Down mode.
10	WO	0x0	CGOUTNak Clear Global OUT NAK A write to this field clears the Global OUT NAK.
9	WO	0x0	SGOUTNak Set Global OUT NAK A write to this field sets the Global OUT NAK. The application uses this bit to send a NAK handshake on all OUT endpoints. The application must set this bit only after making sure that the Global OUT NAK Effective bit in the Core Interrupt Register (GINTSTS.GOUTNakEff) is cleared.
8	WO	0x0	CGNPInNak Clear Global Non-periodic IN NAK A write to this field clears the Global Non-periodic IN NAK.
7	WO	0x0	SGNPInNak Set Global Non-periodic IN NAK A write to this field sets the Global Non-periodic IN NAK. The application uses this bit to send a NAK handshake on all non-periodic IN endpoints. The core can also set this bit when a timeout condition is detected on a non-periodic endpoint in shared FIFO operation. The application must set this bit only after making sure that the Global IN NAK Effective bit in the Core Interrupt Register (GINTSTS.GINNakEff) is cleared.
6:4	RW	0x0	TstCtl Test Control 3'b000: Test mode disabled 3'b001: Test_J mode 3'b010: Test_K mode 3'b011: Test_SE0_NAK mode 3'b100: Test_Packet mode 3'b101: Test_Force_Enable Others: Reserved
3	RO	0x0	GOUTNakSts Global OUT NAK Status 1'b0: A handshake is sent based on the FIFO Status and the NAK and STALL bit settings. 1'b1: No data is written to the RxFIFO, irrespective of space availability. Sends a NAK handshake on all packets, except on SETUP transactions. All isochronous OUT packets are dropped

Bit	Attr	Reset Value	Description
2	RO	0x0	GNPINNakSts Global Non-periodic IN NAK Status 1'b0: A handshake is sent out based on the data availability in the transmit FIFO. 1'b1: A NAK handshake is sent out on all non-periodic IN endpoints, irrespective of the data availability in the transmit FIFO.
1	RW	0x0	SftDiscon Soft Disconnect The application uses this bit to signal the core to do a soft disconnect. As long as this bit is set, the host does not see that the device is connected, and the device does not receive signals on the USB. The core stays in the disconnected state until the application clears this bit. 1'b0: Normal operation. When this bit is cleared after a soft disconnect, the core drives the phy_opmode_o signal on the UTMI+ to 2'b00, which generates a device connect event to the USB host. When the device is reconnected, the USB host restarts device enumeration. 1'b1: The core drives the phy_opmode_o signal on the UTMI+ to 2'b01, which generates a device disconnect event to the USB host.
0	RW	0x0	RmtWkUpSig Remote Wakeup Signaling When the application sets this bit, the core initiates remote signaling to wake the USB host. The application must set this bit to instruct the core to exit the Suspend state. As specified in the USB 2.0 specification, the application must clear this bit 1ms after setting it. If LPM is enabled and the core is in the L1 (Sleep) state, when the application sets this bit, the core initiates L1 remote signaling to wake up the USB host. The application must set this bit to instruct the core to exit the Sleep state. As specified in the LPM specification, the hardware automatically clears this bit 50 us (TL1DevDrvResume) after being set by the application. The application must not set this bit when GLPMCFG bRemoteWake from the previous LPM transaction is zero.

USBOTG_DSTS
Address: Operational Base + offset (0x0808)
Device Status Register

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved

Bit	Attr	Reset Value	Description
21:8	RW	0x0000	SOFFN Frame or Micro-frame Number of the Received SOF When the core is operating at high speed, this field contains a micro-frame number. When the core is operating at full or low speed, this field contains a frame number.
7:4	RO	0x0	reserved
3	RW	0x0	ErrticErr Erratic Error The core sets this bit to report any erratic errors (phy_rxvalid_i/phy_rxvldh_i or phy_rxactive_i is asserted for at least 2ms, due to PHY error) seen on the UTMI+. Due to erratic errors, the OTG core goes into Suspended state and an interrupt is generated to the application with Early Suspend bit of the Core Interrupt register (GINTSTS.ErlySusp). If the early suspend is asserted due to an erratic error, the application can only perform a soft disconnect recover.
2:1	RW	0x0	EnumSpd Enumerated Speed Indicates the speed at which the OTG core has come up after speed detection through a chirp sequence. 2'b00: High speed (PHY clock is running at 30 or 60 MHz) 2'b01: Full speed (PHY clock is running at 30 or 60 MHz) 2'b10: Low speed (PHY clock is running at 48 MHz, internal phy_clk at 6 MHz) 2'b11: Full speed (PHY clock is running at 48 MHz) Low speed is not supported for devices using a UTMI+ PHY.
0	RW	0x0	SuspSts Suspend Status In Device mode, this bit is set as long as a Suspend condition is detected on the USB. The core enters the Suspended state when there is no activity on the utmi_linestate signal for an extended period of time. The core comes out of the suspend: When there is any activity on the utmi_linestate signal, When the application writes to the Remote Wakeup Signaling bit in the Device Control register (DCTL.RmtWkUpSig).

USBOTG_DIEPMSK
Address: Operational Base + offset (0x0810)
Device IN Endpoint common interrupt mask register

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
13	RW	0x0	NAKMsk NAK interrupt Mask
12:10	RO	0x0	reserved

Bit	Attr	Reset Value	Description
9	RW	0x0	BNAInIntrMsk BNA Interrupt Mask
8	RW	0x0	TxfifoUndrnMsk Fifo Underrun Mask
7	RO	0x0	reserved
6	RW	0x0	INEPNakEffMsk IN Endpoint NAK Effective Mask
5	RW	0x0	INTknEPMisMsk IN Token received with EP Mismatch Mask
4	RW	0x0	INTknTXFEmpMsk IN Token Received When TxFIFO Empty Mask
3	RW	0x0	TimeOUTMsk Timeout Condition Mask
2	RW	0x0	AHBErrMsk AHB Error Mask
1	RW	0x0	EPDisbldMsk Endpoint Disabled Interrupt Mask
0	RW	0x0	XferComplMsk Transfer Completed Interrupt Mask

USBOTG_DOEPMSK
Address: Operational Base + offset (0x0814)
Device OUT Endpoint common interrupt mask register

Bit	Attr	Reset Value	Description
31:15	RO	0x0	reserved
14	RW	0x0	NYETMsk NYET Interrupt Mask
13	RW	0x0	NAKMsk NAK Interrupt Mask
12	RW	0x0	BbleErrMsk Babble Interrupt Mask
11:10	RO	0x0	reserved
9	RW	0x0	BnaOutIntrMsk BNA interrupt Mask
8	RW	0x0	OutPktErrMsk OUT Packet Error Mask
7	RO	0x0	reserved
6	RW	0x0	Back2BackSETup Back-to-Back SETUP Packets Received Mask Applies to control OUT endpoints only.
5	RO	0x0	reserved
4	RW	0x0	OUTTknEPdisMsk OUT Token Received when Endpoint Disabled Mask Applies to control OUT endpoints only.

Bit	Attr	Reset Value	Description
3	RW	0x0	SetUPMsk SETUP Phase Done Mask Applies to control endpoints only.
2	RW	0x0	AHBErrMsk AHB Error
1	RW	0x0	EPDisbldMsk Endpoint Disabled Interrupt Mask
0	RW	0x0	XferComplMsk Transfer Completed Interrupt Mask

USBOTG_DAINT
Address: Operational Base + offset (0x0818)
Device All Endpoints interrupt register

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	OutEPInt OUT Endpoint Interrupt Bits One bit per OUT endpoint: Bit 16 for OUT endpoint 0, bit 31 for OUT endpoint 15
15:0	RO	0x0000	InEpInt IN Endpoint Interrupt Bits One bit per IN Endpoint: Bit 0 for IN endpoint 0, bit 15 for endpoint 15

USBOTG_DAINTMSK
Address: Operational Base + offset (0x081c)
Device All Endpoint interrupt mask register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	OutEpMsk OUT EP Interrupt Mask Bits One per OUT Endpoint: Bit 16 for OUT EP 0, bit 31 for OUT EP 15
15:0	RW	0x0000	InEpMsk IN EP Interrupt Mask Bits One bit per IN Endpoint: Bit 0 for IN EP 0, bit 15 for IN EP 15

USBOTG_DTKNQR1
Address: Operational Base + offset (0x0820)
Device IN token sequence learning queue read register1

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	EPTkn Endpoint Token Four bits per token represent the endpoint number of the token: Bits [31:28]: Endpoint number of Token 5 Bits [27:24]: Endpoint number of Token 4 .... Bits [15:12]: Endpoint number of Token 1 Bits [11:8]: Endpoint number of Token 0
7	RO	0x0	WrapBit Wrap Bit This bit is set when the write pointer wraps. It is cleared when the learning queue is cleared.
6:5	RO	0x0	reserved
4:0	RO	0x00	INTknWPtr IN Token Queue Write Pointer

USBOTG_DTKNQR2
Address: Operational Base + offset (0x0824)
Device IN token sequence learning queue read register2

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	EPTkn Endpoint Token Four bits per token represent the endpoint number of the token: Bits [31:28]: Endpoint number of Token 13 Bits [27:24]: Endpoint number of Token 12 .... Bits [7:4]: Endpoint number of Token 7 Bits [3:0]: Endpoint number of Token 6

USBOTG_DVBUSDIS
Address: Operational Base + offset (0x0828)
Device VBUS discharge time register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0b8f	DVBUSDis Device VBUS Discharge Time Specifies the VBUS discharge time after VBUS pulsing during SRP. This value equals: VBUS discharge time in PHY clocks / 1,024.The value you use depends whether the PHY is operating at 30 MHz (16-bit data width) or 60 MHz (8-bit data width). Depending on your VBUS load, this value can need adjustment.

USBOTG_DVBUSPULSE
Address: Operational Base + offset (0x082c)
Device VBUS Pulsing Timer Register

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x000	DVBUSPulse Device VBUS Pulsing Time Specifies the VBUS pulsing time during SRP. This value equals: VBUS pulsing time in PHY clocks / 1,024.The value you use depends whether the PHY is operating at 30 MHz (16-bit data width) or 60 MHz (8-bit data width).

USBOTG_DTHRCTL
Address: Operational Base + offset (0x0830)
Device Threshold Control Register

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27	RW	0x1	ArbPrkEn Arbiter Parking Enable This bit controls internal DMA arbiter parking for IN endpoints. When threshold is enabled and this bit is set to one, then the arbiter parks on the IN endpoint for which there is a token received on the USB. This is done to avoid getting into under-run conditions. By default the parking is enabled.
26	RO	0x0	reserved
25:17	RW	0x008	RxThrLen Receive Threshold Length This field specifies Receive threshold size in DWORDS. This field also specifies the amount of data received on the USB before the core can start transmitting on the AHB. The threshold length has to be at least eight DWORDS. The recommended value for ThrLen is to be the same as the programmed AHB Burst Length (GAHBCFG.HBstLen).
16	RW	0x0	RxThrEn Receive Threshold Enable When this bit is set, the core enables threshold in the receive direction.
15:13	RO	0x0	reserved

Bit	Attr	Reset Value	Description
12:11	RW	0x0	AHBThrRatio AHB Threshold Ratio These bits define the ratio between the AHB threshold and the MAC threshold for the transmit path only. The AHB threshold always remains less than or equal to the USB threshold, because this does not increase overhead. Both the AHB and the MAC threshold must be DWORD-aligned. The application needs to program TxThrLen and the AHBThrRatio to make the AHB Threshold value DWORD aligned. If the AHB threshold value is not DWORD aligned, the core might not behave correctly. When programming the TxThrLen and AHBThrRatio, the application must ensure that the minimum AHB threshold value does not go below 8 DWORDS to meet the USB turnaround time requirements. 2'b00: AHB threshold = MAC threshold 2'b01: AHB threshold = MAC threshold / 2 2'b10: AHB threshold = MAC threshold / 4 2'b11: AHB threshold = MAC threshold / 8
10:2	RW	0x008	TxThrLen Transmit Threshold Length This field specifies Transmit threshold size in DWORDS. This field also forms the MAC threshold and specifies the amount of data, in bytes, to be in the corresponding endpoint transmit FIFO before the core can start transmit on the USB. When the value of AHBThrRatio is 2'h00, the threshold length must be at least 8 DWORDS. If the AHBThrRatio is nonzero, the application must ensure that the AHB threshold value does not go below the recommended 8 DWORDs. This field controls both isochronous and non-isochronous IN endpoint thresholds. The recommended value for ThrLen is to be the same as the programmed AHB Burst Length (GAHBCFG.HBstLen).
1	RW	0x0	ISOThrEn ISO IN Endpoints Threshold Enable When this bit is set, the core enables threshold for isochronous IN endpoints.
0	RW	0x0	NonISOThrEn Non-ISO IN Endpoints Threshold Enable When this bit is set, the core enables threshold for Non Isochronous IN endpoints.

USBOTG_DIEPEMPMSK
Address: Operational Base + offset (0x0834)
Device IN endpoint FIFO empty interrupt mask register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	InEpTxfEmpMsk IN EP Tx FIFO Empty Interrupt Mask Bits These bits act as mask bits for DIEPINTn.TxFEmp interrupt One bit per IN Endpoint: Bit 0 for IN endpoint 0 ... Bit 15 for endpoint 15

USBOTG_DEACHINT
Address: Operational Base + offset (0x0838)
Device each endpoint interrupt register

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	EchOutEPInt OUT Endpoint Interrupt Bits One bit per OUT endpoint: Bit 16 for OUT endpoint 0 ... Bit 31 for OUT endpoint 15
15:0	RO	0x0000	EchInEpInt IN Endpoint Interrupt Bits One bit per IN Endpoint: Bit 0 for IN endpoint 0 ... Bit 15 for endpoint 15

USBOTG_DEACHINTMSK
Address: Operational Base + offset (0x083c)
Device each endpoint interrupt register mask

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	EchOutEpMsk OUT EP Interrupt Mask Bits One per OUT Endpoint: Bit 16 for IN endpoint 0 ... Bit 31 for endpoint 15
15:0	RW	0x0000	EchInEpMsk IN EP Interrupt Mask Bits One bit per IN Endpoint: Bit 0 for IN endpoint 0 ... Bit 15 for endpoint 15

USBOTG_DIEPEACHMSKn
Address: Operational Base + offset (0x0840)

Device each IN endpoint -n interrupt Register

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
13	RW	0x0	NAKMsk NAK interrupt Mask
12:10	RO	0x0	reserved
9	RW	0x0	BNAInIntrMsk BNA interrupt Mask
8	RW	0x0	TxfifoUndrnMsk Fifo Under run Mask
7	RO	0x0	reserved
6	RW	0x0	INEPNakEffMsk IN Endpoint NAK Effective Mask
5	RW	0x0	INTknEPMisMsk IN Token received with EP Mismatch Mask
4	RW	0x0	INTknTXFEmpMsk IN Token Received When TxFIFO Empty Mask
3	RW	0x0	TimeOUTMsk Timeout Condition Mask(Non-isochronous endpoints)
2	RW	0x0	AHBErrMsk AHB Error Mask
1	RW	0x0	EPDisbldMsk Endpoint Disabled Interrupt Mask
0	RW	0x0	XferComplMsk Transfer Completed Interrupt Mask

USBOTG_DOEPEACHMSKn
Address: Operational Base + offset (0x0880)
Device each out endpoint-n interrupt register

Bit	Attr	Reset Value	Description
31:15	RO	0x0	reserved
14	RW	0x0	NYETMsk NYET interrupt Mask
13	RW	0x0	NAKMsk NAK interrupt Mask
12	RW	0x0	BbleErrMsk Babble interrupt Mask
11:10	RO	0x0	reserved
9	RW	0x0	BnaOutIntrMsk BNA interrupt Mask
8	RW	0x0	OutPktErrMsk OUT Packet Error Mask
7	RO	0x0	reserved

Bit	Attr	Reset Value	Description
6	RW	0x0	Back2BackSETup Back-to-Back SETUP Packets Received Mask Applies to control OUT endpoints only.
5	RO	0x0	reserved
4	RW	0x0	OUTTknEPdisMsk OUT Token Received when Endpoint Disabled Mask Applies to control OUT endpoints only.
3	RW	0x0	SetUPMsk SETUP Phase Done Mask Applies to control endpoints only.
2	RW	0x0	AHBErrMsk AHB Error
1	RW	0x0	EPDisbldMsk Endpoint Disabled Interrupt Mask
0	RW	0x0	XferComplMsk Transfer Completed Interrupt Mask

USBOTG_DIEPCTL0
Address: Operational Base + offset (0x0900)
Device control IN endpoint 0 control register

Bit	Attr	Reset Value	Description
31	R/W SC	0x0	EPEna Endpoint Enable When Scatter/Gather DMA mode is enabled, for IN endpoints this bit indicates that the descriptor structure and data buffer with data ready to transmit is setup. When Scatter/Gather DMA mode is disabled-such as in buffer-pointer based DMA mode-this bit indicates that data is ready to be transmitted on the endpoint. The core clears this bit before setting the following interrupts on this endpoint: Endpoint Disabled; Transfer Completed.
30	R/W SC	0x0	EPDis Endpoint Disable The application sets this bit to stop transmitting data on an endpoint, even before the transfer for that endpoint is complete. The application must wait for the Endpoint Disabled interrupt before treating the endpoint as disabled. The core clears this bit before setting the Endpoint Disabled Interrupt. The application must set this bit only if Endpoint Enable is already set for this endpoint.
29:28	RO	0x0	reserved

Bit	Attr	Reset Value	Description
27	WO	0x0	SNAK Set NAK A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also set this bit for an endpoint after a SETUP packet is received on that endpoint.
26	WO	0x0	CNAK Clear NAK A write to this bit clears the NAK bit for the endpoint.
25:23	RO	0x0	reserved
22	RW	0x0	TxFNum TxFIFO Number For Shared FIFO operation, this value is always set to 0, indicating that control IN endpoint 0 data is always written in the Non-Periodic Transmit FIFO. For Dedicated FIFO operation, this value is set to the FIFO number that is assigned to IN Endpoint 0.
21	R/W SC	0x0	Stall STALL Handshake The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority.
20	RO	0x0	reserved
19:18	RO	0x0	EPType Endpoint Type Hardcoded to 00 for control
17	RO	0x0	NAKSts NAK Status Indicates the following: 1'b0: The core is transmitting non-NAK handshakes based on the FIFO status 1'b1: The core is transmitting NAK handshakes on this endpoint. When this bit is set, either by the application or core, the core stops transmitting data, even if there is data available in the TxFIFO. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.
16	RO	0x0	reserved
15	RO	0x1	USBActEP USB Active Endpoint This bit is always set to 1, indicating that control endpoint 0 is always active in all configurations and interfaces.

Bit	Attr	Reset Value	Description
14:11	RW	0x0	NextEp Next Endpoint Applies to non-periodic IN endpoints only. Indicates the endpoint number to be fetched after the data for the current endpoint is fetched. The core can access this field, even when the Endpoint Enable (EPEna) bit is not set. This field is not valid in Slave mode. Note: This field is valid only for Shared FIFO operations.
10:2	RO	0x0	reserved
1:0	RW	0x0	MPS Maximum Packet Size Applies to IN and OUT endpoints. The application must program this field with the maximum packet size for the current logical endpoint. 2'b00: 64 bytes 2'b01: 32 bytes 2'b10: 16 bytes 2'b11: 8 bytes

USBOTG_DIEPINTn
Address: Operational Base + offset (0x0908)
Device Endpoint-n Interrupt Register

Bit	Attr	Reset Value	Description
31:15	RO	0x0	reserved
14	W1C	0x0	NYETIntrpt NYET interrupt The core generates this interrupt when a NYET response is transmitted for a non-isochronous OUT endpoint.
13	W1C	0x0	NAKIntrpt NAK interrupt The core generates this interrupt when a NAK is transmitted or received by the device. In case of isochronous IN endpoints the interrupt gets generated when a zero length packet is transmitted due to un-availability of data in the TXFifo.
12	W1C	0x0	BbleErrIntrpt BbleErr (Babble Error) interrupt The core generates this interrupt when babble is received for the endpoint.
11	W1C	0x0	PktDrpSts Packet Dropped Status This bit indicates to the application that an ISOC OUT packet has been dropped. This bit does not have an associated mask bit and does not generate an interrupt. Dependency: This bit is valid in non-Scatter/Gather DMA mode when periodic transfer interrupt feature is selected.

Bit	Attr	Reset Value	Description
10	RO	0x0	reserved
9	W1C	0x0	BNAIntr BNA (Buffer Not Available) Interrupt The core generates this interrupt when the descriptor accessed is not ready for the Core to process, such as Host busy or DMA done Dependency: This bit is valid only when Scatter/Gather DMA mode is enabled.
8	W1C	0x0	TxfifoUndrn FIFO Underrun Applies to IN endpoints only. The core generates this interrupt when it detects a transmit FIFO under-run condition for this endpoint. Dependency: This interrupt is valid only when both of the following conditions are true: Parameter OTG_EN_DED_TX_FIFO==1; Threshold is enabled; OUT Packet Error(OutPktErr). Applies to OUT endpoints only. This interrupt is asserted when the core detects an overflow or a CRC error for an OUT packet. Dependency: This interrupt is valid only when both of the following conditions are true: Parameter OTG_EN_DED_TX_FIFO==1; Threshold is enabled.
7	W1C	0x0	TxFEmp Transmit FIFO Empty This bit is valid only for IN Endpoints. This interrupt is asserted when the TxFIFO for this endpoint is either half or completely empty. The half or completely empty status is determined by the TxFIFO Empty Level bit in the Core AHB Configuration register (GAHBCFG.NPTxFEmpLvl)).
6	W1C	0x0	INEPNakEff IN Endpoint NAK Effective Applies to periodic IN endpoints only. This bit can be cleared when the application clears the IN endpoint NAK by writing to DIEPCTLn.CNAK. This interrupt indicates that the core has sampled the NAK bit set (either by the application or by the core). The interrupt indicates that the IN endpoint NAK bit set by the application has taken effect in the core. This interrupt does not guarantee that a NAK handshake is sent on the USB. A STALL bit takes priority over a NAK bit. This bit is applicable only when the endpoint is enabled. Back-to-Back SETUP Packets Received (Back2BackSETup) Applies to Control OUT endpoints only. This bit indicates that the core has received more than three back-to-back SETUP packets for this particular endpoint.

Bit	Attr	Reset Value	Description
5	W1C	0x0	INTknEPMis IN Token Received with EP Mismatch Applies to non-periodic IN endpoints only. Indicates that the data in the top of the non-periodic TxFIFO belongs to an endpoint other than the one for which the IN token was received. This interrupt is asserted on the endpoint for which the IN token was received. Status Phase Received For Control Write (StsPhseRcvd) This interrupt is valid only for Control OUT endpoints and only in Scatter Gather DMA mode. This interrupt is generated only after the core has transferred all the data that the host has sent during the data phase of a control write transfer, to the system memory buffer. The interrupt indicates to the application that the host has switched from data phase to the status phase of a Control Write transfer. The application can use this interrupt to ACK or STALL the Status phase, after it has decoded the data phase. This is applicable only in case of Scatter Gather DMA mode.
4	W1C	0x0	INTknTXFEmp IN Token Received When TxFIFO is Empty Indicates that an IN token was received when the associated TxFIFO periodic/non-periodic) was empty. This interrupt is asserted on the endpoint for which the IN token was received. OUT Token Received When Endpoint Disabled (OUTTknEPdis) Indicates that an OUT token was received when the endpoint was not yet enabled. This interrupt is asserted on the endpoint for which the OUT token was received.
3	W1C	0x0	TimeOUT Timeout Condition In shared TX FIFO mode, applies to non-isochronous IN endpoints only. In dedicated FIFO mode, applies only to Control IN endpoints. In Scatter/Gather DMA mode, the TimeOUT interrupt is not asserted. Indicates that the core has detected a timeout condition on the USB for the last IN token on this endpoint. SETUP Phase Done (SetUp) Applies to control OUT endpoints only. Indicates that the SETUP phase for the control endpoint is complete and no more back-to-back SETUP packets were received for the current control transfer. On this interrupt, the application can decode the received SETUP data packet.
2	W1C	0x0	AHBErr AHB Error Applies to IN and OUT endpoints. This is generated only in Internal DMA mode when there is an AHB error during an AHB read/write. The application can read the corresponding endpoint DMA address register to get the error address.

Bit	Attr	Reset Value	Description
1	W1C	0x0	EPDisbld Endpoint Disabled Interrupt Applies to IN and OUT endpoints. This bit indicates that the endpoint is disabled per the application's request.
0	W1C	0x0	XferCompl Transfer Completed Interrupt Applies to IN and OUT endpoints. When Scatter/Gather DMA mode is enabled: For IN endpoint this field indicates that the requested data from the descriptor is moved from external system memory to internal FIFO. For OUT endpoint this field indicates that the requested data from the internal FIFO is moved to external system memory. This interrupt is generated only when the corresponding endpoint descriptor is closed, and the IOC bit for the corresponding descriptor is set. When Scatter/Gather DMA mode is disabled, this field indicates that the programmed transfer is complete on the AHB as well as on the USB, for this endpoint.

USBOTG_DIEPTSIZn
Address: Operational Base + offset (0x0910)
Device endpoint n transfer size register

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved

Bit	Attr	Reset Value	Description
30:29	RW	0x0	MC Multi Count Applies to IN endpoints only. For periodic IN endpoints, this field indicates the number of packets that must be transmitted per microframe on the USB. The core uses this field to calculate the data PID for isochronous IN endpoints. 2'b01: 1 packet 2'b10: 2 packets 2'b11: 3 packets For non-periodic IN endpoints, this field is valid only in Internal DMA mode. It specifies the number of packets the core must fetch for an IN endpoint before it switches to the endpoint pointed to by the Next Endpoint field of the Device Endpoint-n Control register (DIEPCTLn.NextEp). Received Data PID (RxDPID) Applies to isochronous OUT endpoints only. This is the data PID received in the last packet for this endpoint. 2'b00: DATA0 2'b01: DATA2 2'b10: DATA1 2'b11: MDATA SETUP Packet Count (SUPCnt).Applies to control OUT Endpoints only. This field specifies the number of back-to-back SETUP data packets the endpoint can receive. 2'b01: 1 packet 2'b10: 2 packets 2'b11: 3 packets
28:19	RW	0x000	PktCnt Packet Count Indicates the total number of USB packets that constitute the Transfer Size amount of data for this endpoint. The power-on value is specified for Width of Packet Counters during configuration (parameter OTG_PACKET_COUNT_WIDTH). IN Endpoints: This field is decremented every time a packet (maximum size or short packet) is read from the TxFIFO. OUT Endpoints: This field is decremented every time a packet (maximum size or short packet) is written to the RxFIFO.

Bit	Attr	Reset Value	Description
18:0	RW	0x00000	XferSize Transfer Size This field contains the transfer size in bytes for the current endpoint. The power-on value is specified for Width of Transfer Size Counters during configuration (parameter OTG_TRANS_COUNT_WIDTH). The core only interrupts the application after it has exhausted the transfer size amount of data. The transfer size can be set to the maximum packet size of the endpoint, to be interrupted at the end of each packet. IN Endpoints: The core decrements this field every time a packet from the external memory is written to the TxFIFO.OUT Endpoints: The core decrements this field every time a packet is read from the RxFIFO and written to the external memory.

USBOTG_DIEPDMAn
Address: Operational Base + offset (0x0914)
Device endpoint-n DMA Address Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMAAddr DMA Address Holds the start address of the external memory for storing or fetching endpoint data. Note: For control endpoints, this field stores control OUT data packets as well as SETUP transaction data packets. When more than three SETUP packets are received back-to-back, the SETUP data packet in the memory is overwritten. This register is incremented on every AHB transaction. The application can give only a DWORD-aligned address. When Scatter/Gather DMA mode is not enabled, the application programs the start address value in this field. When Scatter/Gather DMA mode is enabled, this field indicates the base pointer for the descriptor list.

USBOTG_DTXFSTSn
Address: Operational Base + offset (0x0918)
Device IN endpoint transmit FIFO status register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	INEPTxFSpcAvail IN Endpoint TxFIFO Space Avail Indicates the amount of free space available in the Endpoint TxFIFO.Values in terms of 32-bit words. 16'h0: Endpoint TxFIFO is full 16'h1: 1 word available 16'h2: 2 words available 16'hn: n words available (where 0 . n . 32,768) 16'h8000: 32,768 words available Others: Reserved

USBOTG_DIEPDMABn

Address: Operational Base + offset (0x091c)

Device endpoint-n DMA buffer address register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	DMABufferAddr DMA Buffer Address Holds the current buffer address. This register is updated as and when the data transfer for the corresponding end point is in progress. This register is present only in Scatter/Gather DMA mode. Otherwise this field is reserved.

USBOTG_DIEPCTLn

Address: Operational Base + offset (0x0920)

Device endpoint-n control register

Bit	Attr	Reset Value	Description
31	R/W SC	0x0	EPEna Endpoint Enable Applies to IN and OUT endpoints. When Scatter/Gather DMA mode is enabled, For IN endpoints this bit indicates that the descriptor structure and data buffer with data ready to transmit is setup. For OUT endpoint it indicates that the descriptor structure and data buffer to receive data is setup. When Scatter/Gather DMA mode is enabled-such as for buffer-pointer based DMA mode: For IN endpoints, this bit indicates that data is ready to be transmitted on the endpoint ; For OUT endpoints, this bit indicates that the application has allocated the memory to start receiving data from the USB. The core clears this bit before setting any of the following interrupts on this endpoint: SETUP Phase Done, Endpoint Disabled, Transfer Completed. Note: For control endpoints in DMA mode, this bit must be set to be able to transfer SETUP data packets in memory.

Bit	Attr	Reset Value	Description
30	R/W SC	0x0	EPDis Endpoint Disable Applies to IN and OUT endpoints. The application sets this bit to stop transmitting/receiving data on an endpoint, even before the transfer for that endpoint is complete. The application must wait for the Endpoint Disabled interrupt before treating the endpoint as disabled. The core clears this bit before setting the Endpoint Disabled interrupt. The application must set this bit only if Endpoint Enable is already set for this endpoint.
29	WO	0x0	SetD1PID Set DATA1 PID Applies to interrupt/bulk IN and OUT endpoints only. Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA1.This field is applicable both for Scatter/Gather DMA mode and non-Scatter/Gather DMA mode. Set Odd (micro) frame (SetOddFr). Applies to isochronous IN and OUT endpoints only. Writing to this field sets the Even/Odd (micro) frame (EO_FrNum) field to odd (micro)frame. This field is not applicable for Scatter/Gather DMA mode.
28	WO	0x0	SetD0PID Set DATA0 PID Applies to interrupt/bulk IN and OUT endpoints only. Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA0.This field is applicable both for Scatter/Gather DMA mode and non-Scatter/Gather DMA mode. In non-Scatter/Gather DMA mode: Set Even (micro) frame (SetEvenFr) Applies to isochronous IN and OUT endpoints only. Writing to this field sets the Even/Odd (micro) frame (EO_FrNum) field to even (micro) frame. When Scatter/Gather DMA mode is enabled, this field is reserved. The frame number in which to send data is in the transmit descriptor structure. The frame in which to receive data is updated in received descriptor structure.
27	WO	0x0	SNAK Set NAK Applies to IN and OUT endpoints. A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also set this bit for OUT endpoints on a Transfer Completed interrupt, or after a SETUP is received on the endpoint.
26	WO	0x0	CNAK Clear NAK Applies to IN and OUT endpoints. A write to this bit clears the NAK bit for the endpoint.

Bit	Attr	Reset Value	Description
25:22	RW	0x0	TxFNum TxFIFO Number Shared FIFO Operation: non-periodic endpoints must set this bit to zero. Periodic endpoints must map this to the corresponding Periodic TxFIFO number. 4'h0: Non-Periodic TxFIFO; Others: Specified Periodic TxFIFO number. Note: An interrupt IN endpoint can be configured as a non-periodic endpoint for applications such as mass storage. The core treats an IN endpoint as a non-periodic endpoint if the TxFNum field is set to 0. Otherwise, a separate periodic FIFO must be allocated for an interrupt IN endpoint using coreConsultant, and the number of this FIFO must be programmed into the TxFNum field. Configuring an interrupt IN endpoint as a non-periodic endpoint saves the extra periodic FIFO area. Dedicated FIFO Operation: these bits specify the FIFO number associated with this endpoint. Each active IN endpoint must be programmed to a separate FIFO number. This field is valid only for IN endpoints.
21	RW	0x0	Stall STALL Handshake Applies to non-control, non-isochronous IN and OUT endpoints only. The application sets this bit to stall all tokens from the USB host to this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Only the application can clear this bit, never the core. Applies to control endpoints only. The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.
20	RW	0x0	Snp Snoop Mode Applies to OUT endpoints only. This bit configures the endpoint to Snoop mode. In Snoop mode, the core does not check the correctness of OUT packets before transferring them to application memory.
19:18	RW	0x0	EPType Endpoint Type Applies to IN and OUT endpoints. This is the transfer type supported by this logical endpoint. 2'b00: Control 2'b01: Isochronous 2'b10: Bulk 2'b11: Interrupt

Bit	Attr	Reset Value	Description
17	RO	0x0	NAKSts NAK Status Applies to IN and OUT endpoints. Indicates the following: 1'b0: The core is transmitting non-NAK handshakes based on the FIFO status. 1'b1: The core is transmitting NAK handshakes on this endpoint. When either the application or the core sets this bit: The core stops receiving any data on an OUT endpoint, even if there is space in the RxFIFO to accommodate the incoming packet. For non-isochronous IN endpoints: The core stops transmitting any data on an IN endpoint, even if there data is available in the TxFIFO. For isochronous IN endpoints: The core sends out a zero-length data packet, even if there data is available in the TxFIFO. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.
16	RO	0x0	DPID Endpoint Data PID Applies to interrupt/bulk IN and OUT endpoints only. Contains the PID of the packet to be received or transmitted on this endpoint. The application must program the PID of the first packet to be received or transmitted on this endpoint, after the endpoint is activated. The applications use the SetD1PID and SetD0PID fields of this register to program either DATA0 or DATA1 PID. 1'b0: DATA0 1'b1: DATA1 This field is applicable both for Scatter/Gather DMA mode and non-Scatter/Gather DMA mode. Even/Odd (Micro)Frame (EO_FrNum) In non-Scatter/Gather DMA mode: Applies to isochronous IN and OUT endpoints only. Indicates the (micro) frame number in which the core transmits/receives isochronous data for this endpoint. The application must program the even/odd (micro) frame number in which it intends to transmit/receive isochronous data for this endpoint using the SetEvnFr and SetOddFr fields in this register. 1'b0: Even (micro)frame 1'b1: Odd (micro)frame When Scatter/Gather DMA mode is enabled, this field is reserved. The frame number in which to send data is provided in the transmit descriptor structure. The frame in which data is received is updated in receive descriptor structure.

Bit	Attr	Reset Value	Description
15	R/W SC	0x0	USBActEP USB Active Endpoint Applies to IN and OUT endpoints. Indicates whether this endpoint is active in the current configuration and interface. The core clears this bit for all endpoints (other than EP 0) after detecting a USB reset. After receiving the SetConfiguration and SetInterface commands, the application must program endpoint registers accordingly and set this bit.
14:11	RW	0x0	NextEp Next Endpoint Applies to non-periodic IN endpoints only. Indicates the endpoint number to be fetched after the data for the current endpoint is fetched. The core can access this field, even when the Endpoint Enable (EPEna) bit is low. This field is not valid in Slave mode operation. Note: This field is valid only for Shared FIFO operations.
10:0	RW	0x000	MPS Maximum Packet Size Applies to IN and OUT endpoints. The application must program this field with the maximum packet size for the current logical endpoint. This value is in bytes.

USBOTG_DOEPCTL0

Address: Operational Base + offset (0x0b00)

Device control OUT endpoint 0 control register

Bit	Attr	Reset Value	Description
31	R/W SC	0x0	EPEna Endpoint Enable When Scatter/Gather DMA mode is enabled, for OUT endpoints this bit indicates that the descriptor structure and data buffer to receive data is setup. When Scatter/Gather DMA mode is disabled (such as for buffer-pointer based DMA mode)-this bit indicates that the application has allocated the memory to start receiving data from the USB. The core clears this bit before setting any of the following interrupts on this endpoint: SETUP Phase Done, Endpoint Disabled, Transfer Completed. Note: In DMA mode, this bit must be set for the core to transfer SETUP data packets into memory.
30	WO	0x0	EPDis Endpoint Disable The application cannot disable control OUT endpoint 0.
29:28	RO	0x0	reserved

Bit	Attr	Reset Value	Description
27	WO	0x0	SNAK Set NAK A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also set bit on a Transfer Completed interrupt, or after a SETUP is received on the endpoint.
26	WO	0x0	CNAK Clear NAK A write to this bit clears the NAK bit for the endpoint.
25:22	RO	0x0	reserved
21	R/W SC	0x0	Stall STALL Handshake The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit or Global OUT NAK is set along with this bit, the STALL bit takes priority. Irrespective of this bit’s setting, the core always responds to SETUP data packets with an ACK handshake.
20	RW	0x0	Snp Snoop Mode This bit configures the endpoint to Snoop mode. In Snoop mode, the core does not check the correctness of OUT packets before transferring them to application memory.
19:18	RO	0x0	EPType Endpoint Type Hardcoded to 2’b00 for control.
17	RO	0x0	NAKSts NAK Status Indicates the following: 1'b0: The core is transmitting non-NAK handshakes based on the FIFO status. 1'b1: The core is transmitting NAK handshakes on this endpoint. When either the application or the core sets this bit, the core stops receiving data, even if there is space in the RxFIFO to accommodate the incoming packet. Irrespective of this bit setting, the core always responds to SETUP data packets with an ACK handshake.
16	RO	0x0	reserved
15	RO	0x0	USBActEP USB Active Endpoint This bit is always set to 1, indicating that a control endpoint 0 is always active in all configurations and interfaces.
14:2	RO	0x0	reserved

Bit	Attr	Reset Value	Description
1:0	RO	0x0	MPS Maximum Packet Size The maximum packet size for control OUT endpoint 0 is the same as what is programmed in control IN Endpoint 0. 2'b00: 64 bytes 2'b01: 32 bytes 2'b10: 16 bytes 2'b11: 8 bytes

USBOTG_DOEPINTn
Address: Operational Base + offset (0x0b08)
Device endpoint-n control register

Bit	Attr	Reset Value	Description
31:15	RO	0x0	reserved
14	W1C	0x0	NYETIntrpt NYET interrupt The core generates this interrupt when a NYET response is transmitted for a non-isochronous OUT endpoint.
13	W1C	0x0	NAKIntrpt NAK interrupt The core generates this interrupt when a NAK is transmitted or received by the device. In case of isochronous IN endpoints the interrupt gets generated when a zero length packet is transmitted due to un-availability of data in the TXFifo.
12	W1C	0x0	BbleErrIntrpt BbleErr (Babble Error) interrupt The core generates this interrupt when babble is received for the endpoint.
11	W1C	0x0	PktDrpSts Packet Dropped Status This bit indicates to the application that an ISOC OUT packet has been dropped. This bit does not have an associated mask bit and does not generate an interrupt. Dependency: This bit is valid in non-Scatter/Gather DMA mode when periodic transfer interrupt feature is selected.
10	RO	0x0	reserved
9	W1C	0x0	BNAIntr BNA (Buffer Not Available) Interrupt The core generates this interrupt when the descriptor accessed is not ready for the Core to process, such as Host busy or DMA done. Dependency: This bit is valid only when Scatter/Gather DMA mode is enabled.

Bit	Attr	Reset Value	Description
8	W1C	0x0	TxfifoUndrn FIFO Underrun Applies to IN endpoints only. The core generates this interrupt when it detects a transmit FIFO under-run condition for this endpoint. Dependency: This interrupt is valid only when both of the following conditions are true: Parameter OTG_EN_DED_TX_FIFO==1, Threshold is enabled, OUT Packet Error (OutPktErr). Applies to OUT endpoints only. This interrupt is asserted when the core detects an overflow or a CRC error for an OUT packet. Dependency: This interrupt is valid only when both of the following conditions are true: Parameter OTG_EN_DED_TX_FIFO==1, Threshold is enabled.
7	W1C	0x0	TxFEmp Transmit FIFO Empty This bit is valid only for IN Endpoints. This interrupt is asserted when the TxFIFO for this endpoint is either half or completely empty. The half or completely empty status is determined by the TxFIFO Empty Level bit in the Core AHB Configuration register (GAHBCFG.NPTxFEmpLvl)).
6	W1C	0x0	INEPNakEff IN Endpoint NAK Effective Applies to periodic IN endpoints only. This bit can be cleared when the application clears the IN endpoint NAK by writing to DIEPCTLn.CNAK. This interrupt indicates that the core has sampled the NAK bit set (either by the application or by the core). The interrupt indicates that the IN endpoint NAK bit set by the application has taken effect in the core. This interrupt does not guarantee that a NAK handshake is sent on the USB. A STALL bit takes priority over a NAK bit. This bit is applicable only when the endpoint is enabled. Back-to-Back SETUP Packets Received (Back2BackSETup) Applies to Control OUT endpoints only. This bit indicates that the core has received more than three back-to-back SETUP packets for this particular endpoint.
5	W1C	0x0	INTknEPMis IN Token Received with EP Mismatch Applies to non-periodic IN endpoints only. Indicates that the data in the top of the non-periodic TxFIFO belongs to an endpoint other than the one for which the IN token was received. This interrupt is asserted on the endpoint for which the IN token was received. Status Phase Received For Control Write (StsPhseRcvd) This interrupt is valid only for Control OUT endpoints and only in Scatter Gather DMA mode.

Bit	Attr	Reset Value	Description
4	W1C	0x0	INTknTXFEmp IN Token Received When TxFIFO is Empty Indicates that an IN token was received when the associated TxFIFO periodic/non-periodic) was empty. This interrupt is asserted on the endpoint for which the IN token was received. OUT Token Received When Endpoint Disabled (OUTTknEPdis) Indicates that an OUT token was received when the endpoint was not yet enabled. This interrupt is asserted on the endpoint for which the OUT token was received.
3	W1C	0x0	TimeOUT Timeout Condition In shared TX FIFO mode, applies to non-isochronous IN endpoints only. In dedicated FIFO mode, applies only to Control IN endpoints. In Scatter/Gather DMA mode, the TimeOUT interrupt is not asserted. Indicates that the core has detected a timeout condition on the USB for the last IN token on this endpoint. SETUP Phase Done (SetUp). Applies to control OUT endpoints only. Indicates that the SETUP phase for the control endpoint is complete and no more back-to-back SETUP packets were received for the current control transfer. On this interrupt, the application can decode the received SETUP data packet.
2	W1C	0x0	AHBErr AHB Error Applies to IN and OUT endpoints. This is generated only in Internal DMA mode when there is an AHB error during an AHB read/write. The application can read the corresponding endpoint DMA address register to get the error address.
1	W1C	0x0	EPDisbld Endpoint Disabled Interrupt Applies to IN and OUT endpoints. This bit indicates that the endpoint is disabled per the application's request.
0	W1C	0x0	XferCompl Transfer Completed Interrupt Applies to IN and OUT endpoints. When Scatter/Gather DMA mode is enabled. For IN endpoint this field indicates that the requested data from the descriptor is moved from external system memory to internal FIFO. For OUT endpoint this field indicates that the requested data from the internal FIFO is moved to external system memory. This interrupt is generated only when the corresponding endpoint descriptor is closed, and the IOC bit for the corresponding descriptor is set. When Scatter/Gather DMA mode is disabled, this field indicates that the programmed transfer is complete on the AHB as well as on the USB, for this endpoint.

USBOTG_DOEPTSIZn
Address: Operational Base + offset (0x0b10)
Device endpoint n transfer size register

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30:29	RW	0x0	MC Multi Count Applies to IN endpoints only. For periodic IN endpoints, this field indicates the number of packets that must be transmitted per microframe on the USB. The core uses this field to calculate the data PID for isochronous IN endpoints. 2'b01: 1 packet 2'b10: 2 packets 2'b11: 3 packets For non-periodic IN endpoints, this field is valid only in Internal DMA mode. It specifies the number of packets the core must fetch for an IN endpoint before it switches to the endpoint pointed to by the Next Endpoint field of the Device Endpoint-n Control register (DIEPCTLn.NextEp). Received Data PID (RxDPID) Applies to isochronous OUT endpoints only. This is the data PID received in the last packet for this endpoint. 2'b00: DATA0 2'b01: DATA2 2'b10: DATA1 2'b11: MDATA SETUP Packet Count (SUPCnt).Applies to control OUT Endpoints only. This field specifies the number of back-to-back SETUP data packets the endpoint can receive. 2'b01: 1 packet 2'b10: 2 packets 2'b11: 3 packets
28:19	RW	0x000	PktCnt Packet Count Indicates the total number of USB packets that constitute the Transfer Size amount of data for this endpoint. The power-on value is specified for Width of Packet Counters during coreConsultant configuration (parameter OTG_PACKET_COUNT_WIDTH). IN Endpoints: This field is decremented every time a packet (maximum size or short packet) is read from the TxFIFO. OUT Endpoints: This field is decremented every time a packet (maximum size or short packet) is written to the RxFIFO.

Bit	Attr	Reset Value	Description
18:0	RW	0x00000	XferSize Transfer Size This field contains the transfer size in bytes for the current endpoint. The power-on value is specified for Width of Transfer Size Counters during coreConsultant configuration (parameter OTG_TRANS_COUNT_WIDTH). The core only interrupts the application after it has exhausted the transfer size amount of data. The transfer size can be set to the maximum packet size of the endpoint, to be interrupted at the end of each packet. IN Endpoints: The core decrements this field every time a packet from the external memory is written to the TxFIFO. OUT Endpoints: The core decrements this field every time a packet is read from the RxFIFO and written to the external memory.

USBOTG_DOEPDMAn
Address: Operational Base + offset (0x0b14)
Device Endpoint-n DMA Address Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMAAddr DMA Address Holds the start address of the external memory for storing or fetching endpoint data. Note: For control endpoints, this field stores control OUT data packets as well as SETUP transaction data packets. When more than three SETUP packets are received back-to-back, the SETUP data packet in the memory is overwritten. This register is incremented on every AHB transaction. The application can give only a DWORD-aligned address. When Scatter/Gather DMA mode is not enabled, the application programs the start address value in this field. When Scatter/Gather DMA mode is enabled, this field indicates the base pointer for the descriptor list.

USBOTG_DOEPDMABn
Address: Operational Base + offset (0x0b1c)
Device endpoint-n DMA buffer address register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	DMABufferAddr DMA Buffer Address Holds the current buffer address. This register is updated as and when the data transfer for the corresponding end point is in progress. This register is present only in Scatter/Gather DMA mode. Otherwise this field is reserved.

USBOTG_DOEPCTLn
Address: Operational Base + offset (0x0b20)
Device endpoint-n control register

Bit	Attr	Reset Value	Description
31	R/W SC	0x0	EPEna Endpoint Enable Applies to IN and OUT endpoints. When Scatter/Gather DMA mode is enabled, For IN endpoints this bit indicates that the descriptor structure and data buffer with data ready to transmit is setup. For OUT endpoint it indicates that the descriptor structure and data buffer to receive data is setup. When Scatter/Gather DMA mode is enabled-such as for buffer-pointer based DMA mode: For IN endpoints, this bit indicates that data is ready to be transmitted on the endpoint; For OUT endpoints, this bit indicates that the application has allocated the memory to start receiving data from the USB. The core clears this bit before setting any of the following interrupts on this endpoint: SETUP Phase Done, Endpoint Disabled, Transfer Completed. Note: For control endpoints in DMA mode, this bit must be set to be able to transfer SETUP data packets in memory.
30	R/W SC	0x0	EPDis Endpoint Disable Applies to IN and OUT endpoints. The application sets this bit to stop transmitting/receiving data on an endpoint, even before the transfer for that endpoint is complete. The application must wait for the Endpoint Disabled interrupt before treating the endpoint as disabled. The core clears this bit before setting the Endpoint Disabled interrupt. The application must set this bit only if Endpoint Enable is already set for this endpoint.
29	RO	0x0	SetD1PID Field0001 Abstract Applies to interrupt/bulk IN and OUT endpoints only. Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA1.This field is applicable both for Scatter/Gather DMA mode and non-Scatter/Gather DMA mode. Set Odd (micro) frame (SetOddFr). Applies to isochronous IN and OUT endpoints only. Writing to this field sets the Even/Odd (micro) frame (EO_FrNum) field to odd (micro) frame. This field is not applicable for Scatter/Gather DMA mode.

Bit	Attr	Reset Value	Description
28	WO	0x0	SetD0PID Set DATA0 PID Applies to interrupt/bulk IN and OUT endpoints only. Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA0.This field is applicable both for Scatter/Gather DMA mode and non-Scatter/Gather DMA mode. In non-Scatter/Gather DMA mode: Set Even (micro) frame (SetEvenFr) Applies to isochronous IN and OUT endpoints only. Writing to this field sets the Even/Odd (micro) frame (EO_FrNum) field to even (micro) frame. When Scatter/Gather DMA mode is enabled, this field is reserved. The frame number in which to send data is in the transmit descriptor structure. The frame in which to receive data is updated in received descriptor structure.
27	WO	0x0	SNAK Set NAK Applies to IN and OUT endpoints. A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also set this bit for OUT endpoints on a Transfer Completed interrupt, or after a SETUP is received on the endpoint.
26	WO	0x0	CNAK Clear NAK Applies to IN and OUT endpoints. A write to this bit clears the NAK bit for the endpoint.
25:22	RW	0x0	TxFNum TxFIFO Number Shared FIFO Operation: non-periodic endpoints must set this bit to zero. Periodic endpoints must map this to the corresponding Periodic TxFIFO number. 4'h0: Non-Periodic TxFIFO; Others: Specified Periodic TxFIFO number. Note: An interrupt IN endpoint can be configured as a non-periodic endpoint for applications such as mass storage. The core treats an IN endpoint as a non-periodic endpoint if the TxFNum field is set to 0. Otherwise, a separate periodic FIFO must be allocated for an interrupt IN endpoint using coreConsultant, and the number of this FIFO must be programmed into the TxFNum field. Configuring an interrupt IN endpoint as a non-periodic endpoint saves the extra periodic FIFO area. Dedicated FIFO Operation: these bits specify the FIFO number associated with this endpoint. Each active IN endpoint must be programmed to a separate FIFO number. This field is valid only for IN endpoints.

Bit	Attr	Reset Value	Description
21	RW	0x0	Stall STALL Handshake Applies to non-control, non-isochronous IN and OUT endpoints only. The application sets this bit to stall all tokens from the USB host to this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Only the application can clear this bit, never the core. Applies to control endpoints only. The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.
20	RW	0x0	Snp Snoop Mode Applies to OUT endpoints only. This bit configures the endpoint to Snoop mode. In Snoop mode, the core does not check the correctness of OUT packets before transferring them to application memory.
19:18	RW	0x0	EPType Endpoint Type Applies to IN and OUT endpoints. This is the transfer type supported by this logical endpoint. 2'b00: Control 2'b01: Isochronous 2'b10: Bulk 2'b11: Interrupt
17	RO	0x0	NAKSts NAK Status Applies to IN and OUT endpoints. Indicates the following: 1'b0: The core is transmitting non-NAK handshakes based on the FIFO status. 1'b1: The core is transmitting NAK handshakes on this endpoint. When either the application or the core sets this bit: The core stops receiving any data on an OUT endpoint, even if there is space in the RxFIFO to accommodate the incoming packet.

Bit	Attr	Reset Value	Description
16	RO	0x0	DPID Endpoint Data PID Applies to interrupt/bulk IN and OUT endpoints only. Contains the PID of the packet to be received or transmitted on this endpoint. The application must program the PID of the first packet to be received or transmitted on this endpoint, after the endpoint is activated. The applications use the SetD1PID and SetD0PID fields of this register to program either DATA0 or DATA1 PID. 1'b0: DATA0 1'b1: DATA1 This field is applicable both for Scatter/Gather DMA mode and non-Scatter/Gather DMA mode. Even/Odd (Micro) Frame (EO_FrNum). In non-Scatter/Gather DMA mode: Applies to isochronous IN and OUT endpoints only. Indicates the (micro)frame number in which the core transmits/receives isochronous data for this endpoint. The application must program the even/odd (micro) frame number in which it intends to transmit/receive isochronous data for this endpoint using the SetEvnFr and SetOddFr fields in this register. 1'b0: Even (micro)frame 1'b1: Odd (micro)frame When Scatter/Gather DMA mode is enabled, this field is reserved. The frame number in which to send data is provided in the transmit descriptor structure. The frame in which data is received is updated in receive descriptor structure.
15	R/W SC	0x0	USBActEP USB Active Endpoint Applies to IN and OUT endpoints. Indicates whether this endpoint is active in the current configuration and interface. The core clears this bit for all endpoints (other than EP 0) after detecting a USB reset. After receiving the SetConfiguration and SetInterface commands, the application must program endpoint registers accordingly and set this bit.
14:11	RW	0x0	NextEp Next Endpoint Applies to non-periodic IN endpoints only. Indicates the endpoint number to be fetched after the data for the current endpoint is fetched. The core can access this field, even when the Endpoint Enable (EPEna) bit is low. This field is not valid in Slave mode operation. Note: This field is valid only for Shared FIFO operations.

Bit	Attr	Reset Value	Description
10:0	RW	0x000	MPS Maximum Packet Size Applies to IN and OUT endpoints. The application must program this field with the maximum packet size for the current logical endpoint. This value is in bytes.

USBOTG_PCGCR
Address: Operational Base + offset (0x0e00)
Power and clock gating control register

Bit	Attr	Reset Value	Description
31:14	RW	0x0802e	RestoreValue Restore Value (Applicable only when Hibernation is enabled (OTG_EN_PWROPT=2). Defines port clock select for different speeds. [31] if_dev_mode - 1: Device mode, core restored as device - 0: Host mode, core restored as host [30:29] p2hd_prt_spd (PRT speed) - 00: HS - 01: FS - 10: LS - 11: Reserved [28:27] p2hd_dev_enum_spd (Device enumerated speed) - 00: HS - 01: FS (30/60 MHz clk) - 10: LS - 11: FS (48 MHz clk) [26:20] mac_dev_addr (MAC device address) Device address [19] mac_termselect (Termination selection) - 0: HS_TERM (Program for High Speed) - 1: FS_TERM (Program for Full Speed) [18:17] mac_xcvrselect (Transceiver select) - 00: HS_XCVR (High Speed) - 01: FS_XCVR (Full Speed) - 10: LS_XCVR (Low Speed) - 11: LFS_XCVR (Reserved) [16] sh2pl_prt_ctl[0] - 1: prt_power enabled - 0: prt_power disabled [15:14] prt_clk_sel (Refer prt_clk_sel table)

Bit	Attr	Reset Value	Description
13	RW	0x0	EssRegRestored Essential Register Values Restored (Applicable only when Hibernation is enabled (OTG_EN_PWROPT=2). When a value of 1 is written to this field, it indicates that register values of essential registers have been restored.
12:10	RO	0x0	reserved
9	RO	0x0	RestoreMode Restore Mode (Applicable only when Hibernation is enabled (OTG_EN_PWROPT=2). The application should program this bit to specify the restore mode during RESTORE POINT before programming PCGCCTL.EssRegRest bit is set. Host Mode: 1'b0: Host Initiated Resume, Host Initiated Reset 1'b1: Device Initiated Remote Wake up Device Mode: 1'b0: Device Initiated Remote Wake up 1'b1: Host Initiated Resume, Host Initiated Reset
8	RW	0x0	ResetAfterSusp Reset After Suspend Applicable in Partial power-down mode. In partial power-down mode of operation, this bit needs to be set in host mode before clamp is removed if the host needs to issue reset after suspend. If this bit is not set, then the host issues resume after suspend. This bit is not applicable in device mode and non-partial power-down mode. In Hibernation mode, this bit needs to be set at RESTORE_POINT before PCGCCTL.EssRegRestored is set. In this case, PCGCCTL.restore_mode needs to be set to wait_restore.
7	RO	0x0	L1Suspended Deep Sleep This bit indicates that the PHY is in deep sleep when in L1 state.
6	RO	0x0	PhySleep PHY in Sleep This bit indicates that the PHY is in the Sleep state.
5	RW	0x0	Enbl_L1Gating Enable Sleep Clock Gating When this bit is set, core internal clock gating is enabled in Sleep state if the core cannot assert utmi_l1_suspend_n. When this bit is not set, the PHY clock is not gated in Sleep state.
4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3	RW	0x0	RstPdwnModule Reset Power-Down Modules This bit is valid only in Partial Power-Down mode. The application sets this bit when the power is turned off. The application clears this bit after the power is turned on and the PHY clock is up.
2	RW	0x0	PwrClmp Power Clamp This bit is valid only in Partial Power-Down mode (OTG_EN_PWROPT = 1). The application sets this bit before the power is turned off to clamp the signals between the power-on modules and the power-off modules. The application clears the bit to disable the clamping before the power is turned on.
1	RW	0x0	GateHclk Gate Hclk The application sets this bit to gate hclk to modules other than the AHB Slave and Master and wakeup logic when the USB is suspended or the session is not valid. The application clears this bit when the USB is resumed or a new session starts.
0	RW	0x0	StopPclk Stop Pclk The application sets this bit to stop the PHY clock (phy_clk) when the USB is suspended, the session is not valid, or the device is disconnected. The application clears this bit when the USB is resumed or a new session starts.

USBOTG_EPBUF0
Address: Operational Base + offset (0x1000)
Device endpoint 0 / host out channel 0 address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	EPBUF0 From0x1000 to 0x2000, EPBUF for endport0

USBOTG_EPBUF1
Address: Operational Base + offset (0x2000)
Device endpoint 1 / host out channel 1 address

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	EPBUF1 From0x2000 to 0x3000, EPBUF for endport1

USBOTG_EPBUF2
Address: Operational Base + offset (0x3000)
Device endpoint 2 / host out channel 2 address

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	EPBUF2 From0x3000 to 0x4000, EPBUF for endport2

USBOTG_EPBUF3
Address: Operational Base + offset (0x4000)
Device endpoint 3 / host out channel 3 address

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	EPBUF3 From0x4000 to 0x5000, EPBUF for endport3

USBOTG_EPBUF4
Address: Operational Base + offset (0x5000)
Device endpoint 4 / host out channel 4 address

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	EPBUF4 From0x5000 to 0x6000, EPBUF for endport4

USBOTG_EPBUF5
Address: Operational Base + offset (0x6000)
Device endpoint 5 / host out channel 5 address

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	EPBUF5 From0x6000 to 0x7000, EPBUF for endport5

USBOTG_EPBUF6
Address: Operational Base + offset (0x7000)
Device endpoint 6 / host out channel 6 address

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	EPBUF6 From0x7000 to 0x8000, EPBUF for endport6

USBOTG_EPBUF7
Address: Operational Base + offset (0x8000)
Device endpoint 7 / host out channel 7 address

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	EPBUF7 From0x8000 to 0x9000, EPBUF for endport7

6.6.4 PHY Configuration Interface

This USB OTG shares a combo PHY with USB2.0 host controller, and the OTG uses the port 0 of the combo PHY. The PHY has a register configuration interface with total width of 384 bits, and bit[191:0] are related to OTG while bit[383:192] are related to host controller. Similar to host controller, bit[191:0] configuration bits are programmed in USBPHY_GRF. Please refer to "Chapter GRF" for detailed bit assignment for these 192 bits. Detailed function description of these configuration bits are as follows.

Table 6-2 USB OTG PHY Configuration Interface Description

Bit	Attr	Reset Value	Description
383: 192	-	-	These 192 bits are related to OTG. Please refer to "Chapter USB2 HOST".
191	RW	0x0	Single ended disconnect detection enable.
190: 187	RW	0x0	Digital debug interface, keeping the default value is greatly appreciated. These bits will never be used by users normally.
186: 184	RW	0x0	COMP mode selection.
183: 181	RW	0x0	Disconnect bias current tuning bits, more ones will promote the disconnect working bandwidth.
180: 128	RW	0x0	Digital debug interface, keeping the default value is greatly appreciated. These bits will never be used by users normally.
127	RW	0x0	vbus voltage level detection function power down, active high.

126:
123
	RW	0xD	HOST disconnect detection trigger point configure, only used in HOST mode:
			4b'0000: 625mv;
			4b'0001: 675mv;
			4b'0010: 612.5mv;
			4b'0011: 575mv;
			4b'0100: 550mv;
			4b'0101: 600mv;
			4b'0110: 537.5mv;
			4b'0111: 500mv;
			4b'1000: 600mv;
			4b'1001: 650mv;
			4b'1010: 587.5mv;
			4b'1011: 550mv;
			4b'1100: 575mv;
			4b'1101: 625mv(default);
			4b'1110: 562.5mv;
			4b'1111: 525mv.

122: 120	RW	0x0	Digital debug interface, keeping the default value is greatly appreciated. These bits will never be used by users normally.
119	RW	0x0	Bypass squelch trigger point auto configure in chirp modes, active high.
118	RW	0x0	Half bit pre-emphasize enable, active high. "1" represent half bit pre-emphasis, "0" for full bit.
117	RW	0x0	Digital debug interface, reserved.

Bit	Attr	Reset Value	Description
116	RW	0x0	PLL divider ration option, keeping the default value is greatly appreciated.
115	RW	0x0	HS slew rate tuning bits. Please refer to bit[4:3] for more information.
114:112	RW	0x0	vbus_valid reference tuning.
111:109	RW	0x0	session_end reference tuning.
108:106	RW	0x0	B_sessionvalid reference tuning.
105:103	RW	0x0	A_sessionvalid reference tuning.
102	RW	0x0	Force output vbus_valid asserted, active high.
101	RW	0x0	Force output session_end asserted, active high.
100	RW	0x0	Force output B_sessionvalid asserted, active high.
99	RW	0x0	Force output A_sessionvalid asserted, active high.
98	RW	0x1	Turn off differential receiver in suspend mode to save power, active low.
97:66	RW	0x51555555	Digital debug interface, keeping the default value is greatly appreciated. These bits will never be used by users normally.
65	RW	0x1	Digital debug interface.
64:63	RW	0x0	Digital debug interface, keeping the default value is greatly appreciated. These bits will never be used by users normally.
62	RW	0x0	Digital debug interface.
61:58	RW	0x0	Digital debug interface, keeping the default value is greatly appreciated. These bits will never be used by users normally.
57	RW	0x1	A port ODT auto refresh bypass, active low, this register should only be used when bit[43:42] were set to "11". In bypass mode, customer can configure driver strength through bit[41:37].
56	RW	0x0	BG output voltage reference adjust, keeping the default value is greatly appreciated.

55:53	RW	0x0	Compensation current tuning reference:
			3'b000: 400mV(default);
			3'b001: 362.5mV;
			3'b010: 375mV;
			3'b011: 387.5mV;
			3'b100: 412.5mV;
			3'b101: 425mV;
			3'b110: 437.5mV;
			3'b111: 450mV.

Bit	Attr	Reset Value	Description

52:50	RW	0x0	Bias current tuning reference:
			3'b000: 200mV(default);
			3'b001: 212.5mV;
			3'b010: 225mV;
			3'b011: 237.5mV;
			3'b100: 250mV;
			3'b101: 187.5mV;
			3'b110: 175mV;
			3'b111: 162.5mV.

49:47	RW	0x0	ODT compensation voltage reference:
			3'b000: 268mV(default);
			3'b001: 262mV;
			3'b010: 250mV;
			3'b011: 237.5mV;
			3'b100: 275mV;
			3'b101: 281mV;
			3'b110: 293mV;
			3'b111: 300mV.

46:45	RW	0x0	Battery charging related registers, keeping the default value is greatly appreciated.
44	RO	0x0	Reserved.
43:42	RW	0x0	Auto compensation bypass, "11" will bypass current and ODT compensation, customers can set the driver strength and current manually. For larger HS/FS/LS slew rate, give more "1" for bit[233:229].
41:37	RW	0x17	HS/FS driver strength tuning, "11111" represent the largest slew rate and "10000" represents the smallest slew rate.
36:29	RW	0x3f	HS eye height tuning bits. More zeros represent bigger eye.
28:27	RW	0x0	Digital debug interface, keeping the default value is greatly appreciated. These bits will never be used by users normally.
26:19	RW	0x0	Digital debug interface, keeping the default value is greatly appreciated. These bits will never be used by users normally.
18	RW	0x1	Enable current compensation, active high.
17	RW	0x1	Enable resistance compensation, active high.

Bit	Attr	Reset Value	Description

16:13	RW	0xc	port squelch trigger point configuration:
			4b'0000: 112.5mv;
			4b'0001: 150mv;
			4b'0010: 87.5mv;
			4b'0011: 162.5mv;
			4b'0100: 100mv;
			4b'0101: 137.5mv;
			4b'0110: 75mv;
			4b'0111: 150mv;
			4b'1000: 125mv;
			4b'1001: 162.5mv;
			4b'1010: 100mv;
			4b'1011: 175mv;
			4b'1100: 150mv(default);
			4b'1101: 187.5mv;
			4b'1110: 125mv;
			4b'1111: 200mv.

12:11	RW	0x0	Registers for non-driving state control. non-driving state is controlled by op-mode by default, when bit[203] is configured with "1", user can control non-driving state trough bit[204].
10:8	RW	0x5	USB Tx Clock phase configure, 3'b000 represent the earliest phase, and 3'b111 the latest, single step delay is 256ps.
7:5	RW	0x0	USB Rx Clock phase configure, 3'b000 represent the earliest phase, and 3'b111 the latest, single step delay is 256ps.
4:3	RW	0x3	combine with bit[115]. {bit[115],bit[4:3]} form three HS slew rate tuning bits. More one represents larger slew rate, 111 the maximum and 001 the minimum. 000 will shut down the high speed driver output.
2:0	RW	0x0	HS eye diagram adjust, open HS pre-emphasize function to increase HS slew rate, only used when large cap loading is attached. 3'b001: open pre-emphasize in sof or eop state; 3'b010: open pre-emphasize in chirp state; 3'b100: open pre-emphasize in non-chirp state; 3'b111: always open pre-emphasize; other combinations: reserved.

6.7 Interface description

Table 6-3 USB OTG 2.0 Interface Description

Module Pin	Direction	Pin Name	IOMUX Setting
USB0PN	A	USB0_DM	NS
USBRBIAS	A	USB_EXTR	NS
USB0PP	A	USB0_DP	NS
VBUS	A	USB_VBUS	NS
USB0ID	A	USB_ID	NS
DRVVBUS	DP	GPIO0_C5/OTG_DRVBUS	GRF_SOC_CON0[6]=1

***Note**: **A**—Analog pad ;**AP**—Analog power; **AG**—Analog ground ;**DP**—Digital power ;**DG**–Digital ground;*

6.8 Application Note

6.8.1 Suspend Mode

When PHY is in suspend state

- COMMONONN = 1’b1, 480M clock invalid.
- COMMONONN = 1’b0, 480M clock output available.

Please refer to “Chapter GRF” for configuration details.

6.8.2 Relative GRF Registers

USBPHY contains 384-bit registers to configure USB PHY. These bits are used to adjust DP/DM SI. Please refer to “Chapter GRF” for more details.

Chapter 7 Video Output Processor (VOP)

7.1 Overview

Video Output Processor is a video process engine and a display interface from memory frame buffer to display device(LCD panel). VOP is connected to an AHB bus through an AHB slave and AXI bus through an AXI master. The register setting is configured through the AHB slave interface and the display frame data is read through the AXI master interface.

7.1.1 Features

VOP supports the following features:

- Display interface
  - Parallel RGB LCD Interface:18-bit(RGB666),16-bit(RGB565)
  - Max output resolution
    - 1280x800 for MCU/RGB
- Display process
  - Background layer
    - programmable 24-bit color
  - Win0 layer
    - RGB888, ARGB888, RGB565, YCbCr422, YCbCr420, YCbCr444
    - 1/8 to 8 scaling-down and scaling-up engine
    - Support virtual display
    - 256 level alpha blending (pre-multiplied alpha support)
    - Transparency color key
    - YCbCr2RGB(rec601-mpeg/rec601-jpeg/rec709)
    - RGB2YCbCr(BT601/BT709)
  - Win1 layer
    - RGB888, ARGB888, RGB565
    - Support virtual display
    - 256 level alpha blending (pre-multiplied alpha support)
    - Transparency color key
    - RGB2YCbCr(BT601/BT709)
- Others
  - Win0 layer and Win1 layer overlay exchangeable
  - Support RGB or YUV domain overlay
  - BCSH(Brightness,Contrast,Saturation,Hue adjustment)
  - BCSH:YCbCr2RGB(rec601-mpeg/rec601-jpeg/rec709)
  - BCSH:RGB2YCbCr(BT601/BT709)
  - Support Gamma adjust for panel
  - Support dither down allegro RGB888to666 RGB888to565 & dither down frc(configurable) RGB888to666
  - Blank and black display
  - Standby mode
  - Support DMA stop mode
  - Support all layers reg_done separately

Copyright 2018 © FuZhou Rockchip Electronics Co., Ltd.

7.2 Block Diagram

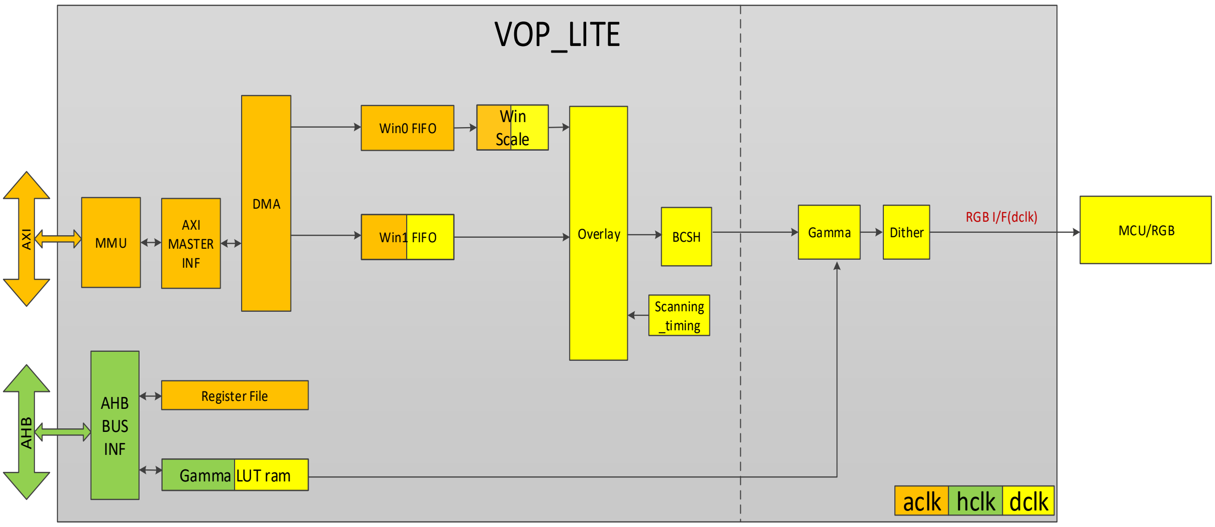


Fig. 7-1 VOP Block Diagram

7.3 Function Description

7.3.1 Data Format

VOP master read the frame data from the frame buffer in the system memory. There are total 6 formats supported in three layers.

- Win0: RGB888, ARGB888, RGB565, YCbCr422_SP, YCbCr420_SP, YCbCr444_SP
- Win1: RGB888, ARGB888, RGB565

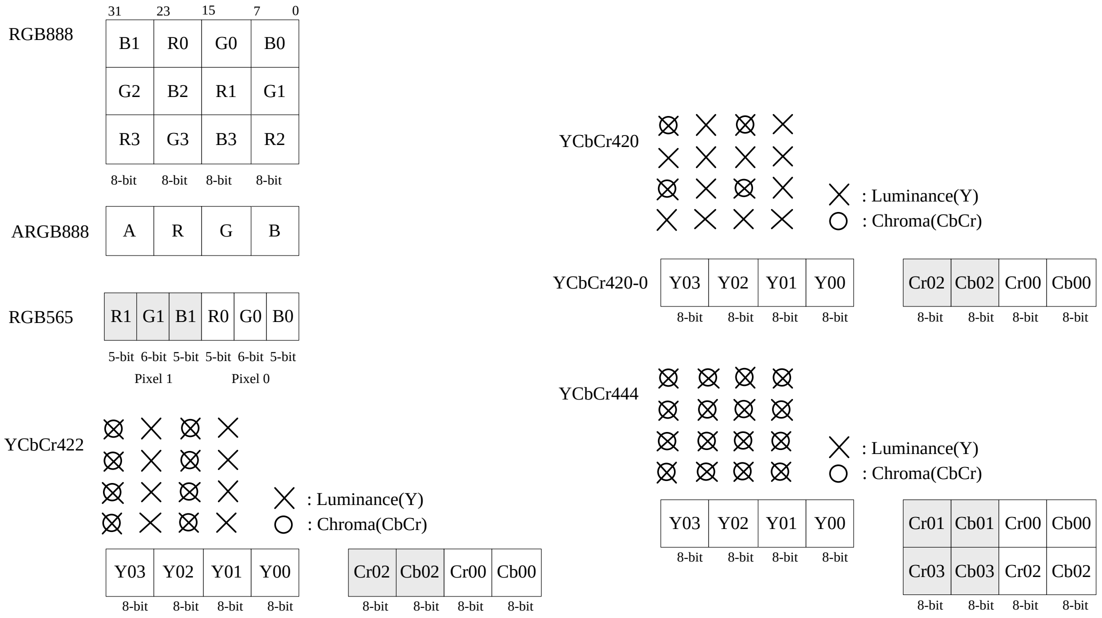
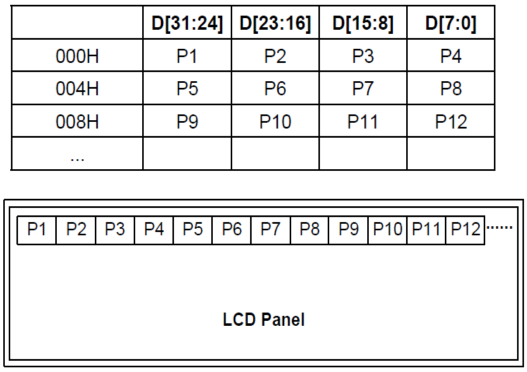


Fig. 7-2 VOP Frame Buffer Data Format



	D[31:24]	D[23:16]	D[15:8]	D[7:0]
000H	P1	P2	P3	P4
004H	P5	P6	P7	P8
008H	P9	P10	P11	P12
...				

Fig. 7-3 VOP Win1 Palette (8bpp)

Data SWAP function

There are several swap options for different frame data formats.
All the data swap types are in the following table.

Table 7-1 VOP Data Swap of Win0 and Win1

Data-swap	RB swap	Alpha swap	Y-M8 swap	CbCr swap
Win0	yes	yes	yes	yes
Win1	yes	yes	No	No

7.3.2 Data path

There are two data input path for VOP to get display layers' pixel data. One is internal DMA;the other is direction path interface.

1.Internal DMA

Internal DMA can fetch the pixel data through AXI bus from system memory (DDR) for all the display layers. Data fetching is driven by display output requirement.

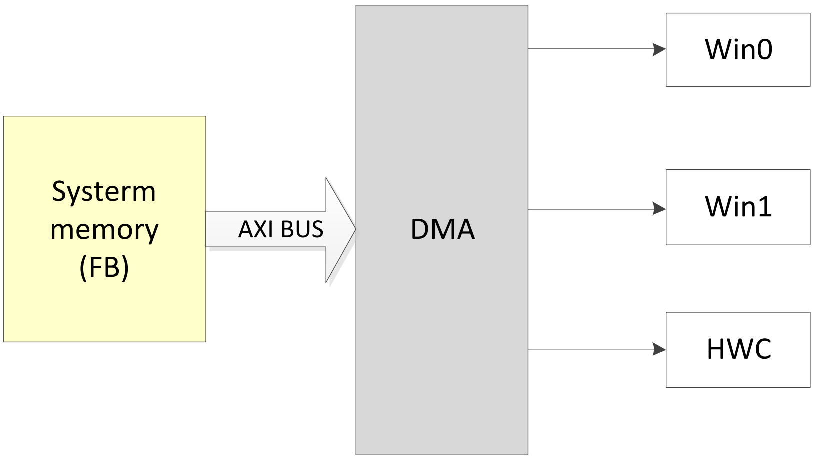


Fig. 7-4 VOP Internal DMA

7.3.3 Virtual display

Virtual display is supported in Win0 and Win1. The active image is part of the virtual (original) image in frame buffer memory. The virtual width is indicated by setting WIN0/WIN1_VIR_STRIDE for different data format.

The virtual stride should be multiples of word (32-bit). That means dummy bytes in the end of virtual line if the real pixels are not 32-bit aligned.

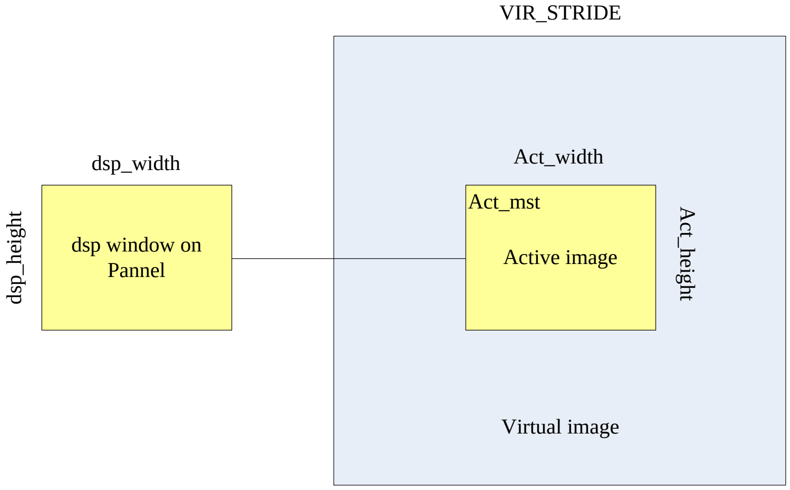


Fig. 7-5 VOP Virtual Display Mode

7.3.4 Scaling

The scaling operation is the image resizing process of data transfer from the frame buffer memory to LCD panel.

Horizontal and vertical scaling factor should be set according the window scaling ratio.

1. Scaling factor

Because the Chroma data may have different sampling rate with Luma data in the memory format of YCbCr422/YCbCr420. The scaling factor of Win0 has two couples of factor registers:

VOP_WIN0_SCL_FACTOR_Y/VOP_WIN0_SCL_FACTOR_CBR

Software calculates the scaling factor value using the following equations:

$$y\_rgb\_vertical\_factor = \left(\frac{VOP\_WIN0\_ACT\_INFO[31:16]}{VOP\_WIN0\_DSP\_INFO[31:16]}\right) \times 2^{12}$$

$$y\_rgb\_horizontal\_factor = \left(\frac{VOP\_WIN0\_ACT\_INFO[15:0]}{VOP\_WIN0\_DSP\_INFO[15:0]}\right) \times 2^{12}$$

$$yuv422\_yuv444\_Cbr\_vertical\_factor = \left(\frac{VOP\_WIN0\_ACT\_INFO[31:16]}{VOP\_WIN0\_DSP\_INFO[31:16]}\right) \times 2^{12}$$

$$yuv420\_Cbr\_vertical\_factor = \left(\frac{VOP\_WIN0\_ACT\_INFO[31:16]/2}{VOP\_WIN0\_DSP\_INFO[31:16]}\right) \times 2^{12}$$

$$yuv444\_Cbr\_horizontal\_factor = \left(\frac{VOP\_WIN0\_ACT\_INFO[15:0]}{VOP\_WIN0\_DSP\_INFO[15:0]}\right) \times 2^{12}$$

$$yuv422\_yuv420\_Cbr\_horizontal\_factor = \left(\frac{VOP\_WIN0\_ACT\_INFO[15:0]/2}{VOP\_WIN0\_DSP\_INFO[15:0]}\right) \times 2^{12}$$

2. Scaling start point offset

The x and y start point of the generated pixels can be adjusted, the offset value is in the range of 0 to 0.99.

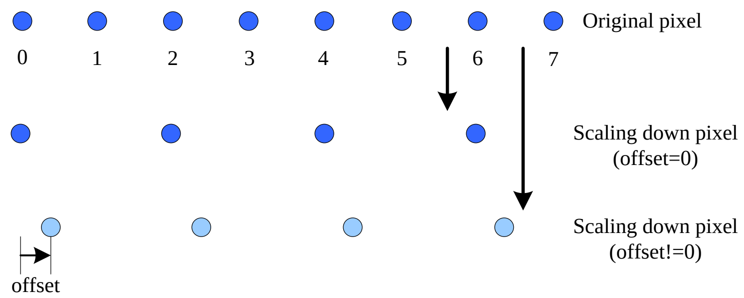


Fig. 7-6 VOP Scaling Down Offset

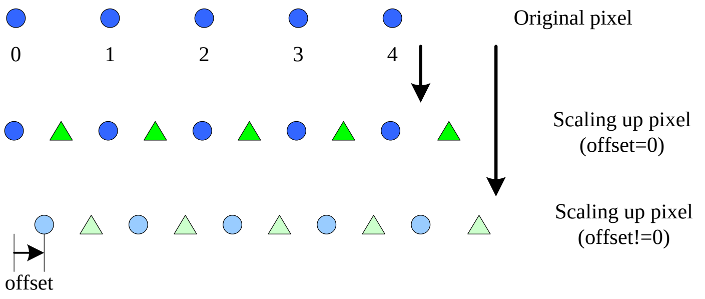


Fig. 7-7 VOP Scaling Up Offset

Table 7-2 VOP Scaling Start Point Offset Registers

scaling down/up start point offset	Offset variable	Register
Win0 YRGB vertical scaling offset	Win0_YRGB_vscl_offset	Win0_SCL_OFFSET [32:24]
Win0 YRGB horizontal scaling offset	Win0_YRGB_hscl_offset	Win0_SCL_OFFSET [23:16]
Win0 Cbr vertical scaling offset	Win0_CBR_vscl_offset	Win0_SCL_OFFSET [15:8]
Win0 Cbr horizontal scaling offset	Win0_CBR_hscl_offset	Win0_SCL_OFFSET[7:0]

7.3.5 Overlay

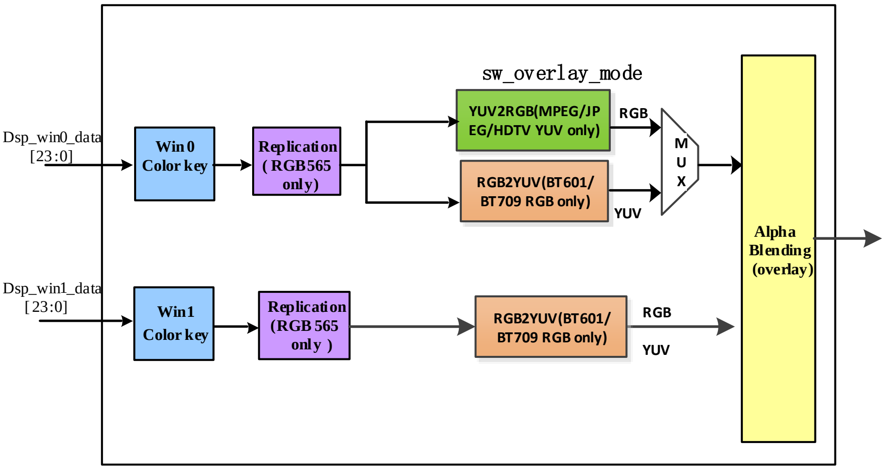

sw_overlay_mode
Dsp_win0_data [23:0]
Win0 Color key
Replication (RGB565 only)
YUV2RGB(MPEG/JPEG/HDTV YUV only)
RGB2YUV(BT601/ BT709 RGB only)
RGB
YUV
M U X
Alpha Blending (overlay)
Dsp_win1_data [23:0]
Win1 Color key
Replication (RGB 565 only)
RGB2YUV(BT601/ BT709 RGB only)
RGB
YUV

Fig. 7-8 VOP Overlay Block Diagram

1. Overlay display

There are totally 3 layers for overlay display: Background, Win0 and Win1.

The background is a programmable solid color layer, which is always the bottom of the display screen.

The two middle layers are Win0 and Win1. Win1 is on the top of Win0 in default setting, setting VOP_DSP_CTRL2[3] to '1' can let Win0 be on the top of Win1.

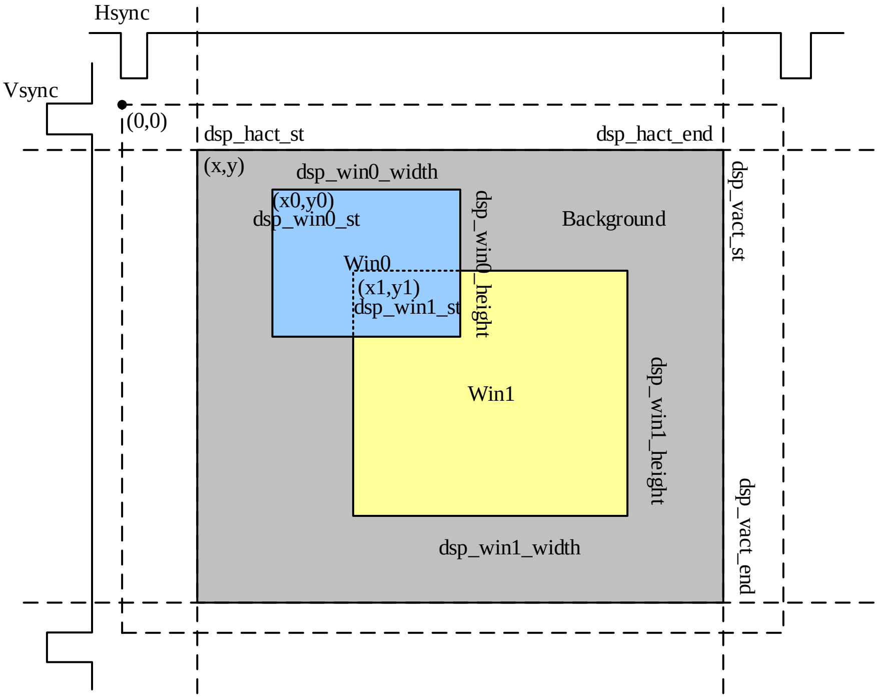


Fig. 7-5 VOP Overlay Display

2. Transparency color key

There are specific registers(VOP_WIN0_COLOR_KEY,VOP_WIN1_COLOR_KEY) for Win0 and Win1 layer to configure the color key value.The two transparency color key can be active at the same time.

The pixel color value is compared to the transparency color key before final display. The transparency color key value defines the pixel data considered as the transparent pixel. The pixel values with the source color key value are pixels not visible on the screen, and the under layer pixel values or solid background color are visible.

Transparency color key is done after the scaling module . So transparency color key can only be used in non-scaling mode.

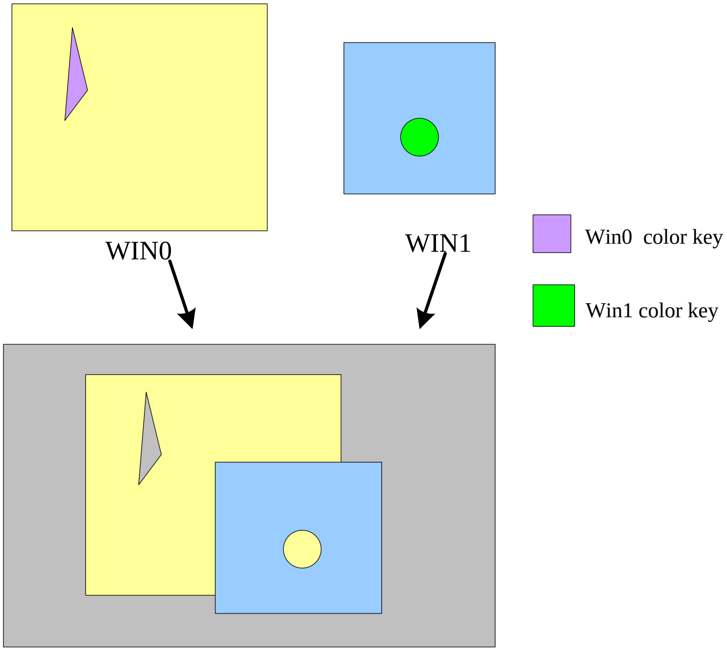
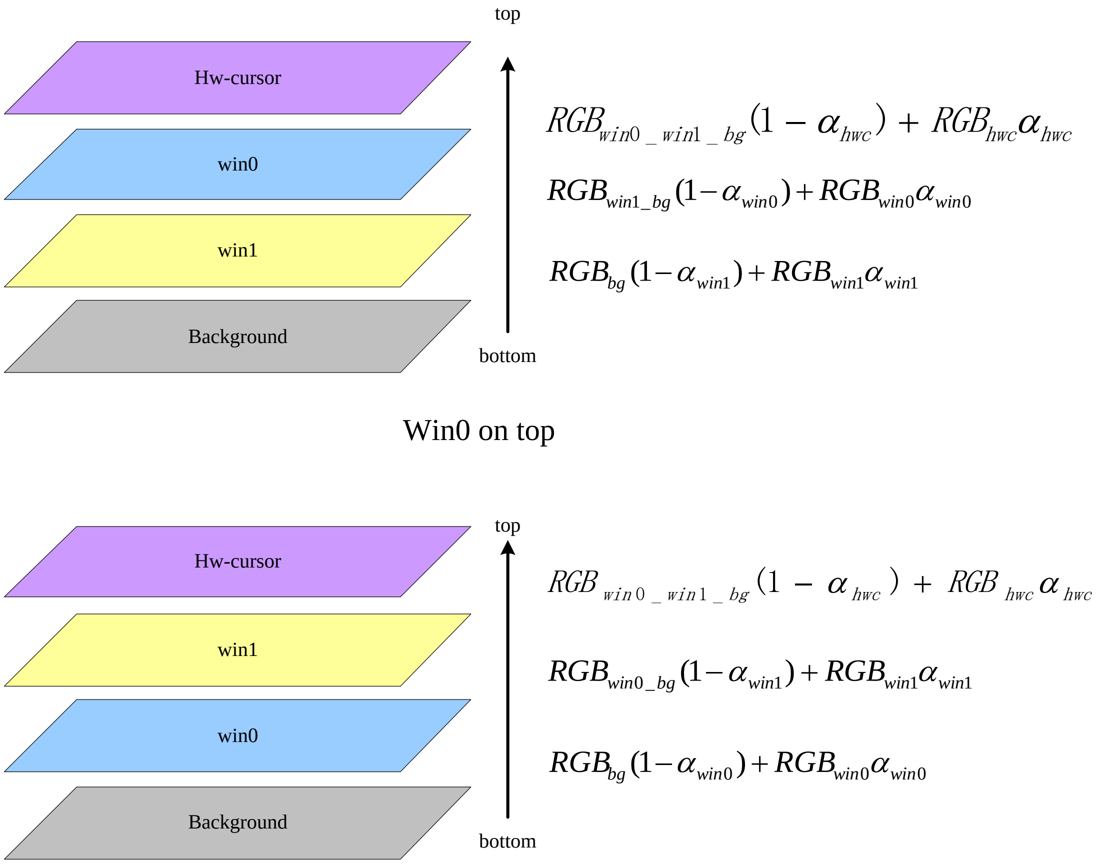


Fig. 7-6 VOP Transparency Color Key

3. Alpha Blending

Two blending modes are supported. One is per-pixel (ARGB) mode; the other is user-specified mode. In ARGB mode, the alpha value is in the ARGB data (Win0 and Win1 normal mode only). In user-specified mode, the alpha value comes from the register (WINX_ALPHA_CTRL[11:4]).

Pre-multiplied alpha are supported for per-pixel alpha in Win0 and Win1, for Pre-multiplied alpha, the SRC data has already been multiplied with alpha value.


top
Hw-cursor
win0
win1
Background
bottom
$RGB_{win0\_win1\_bg}(1-\alpha_{hwc})+RGB_{hwc}\alpha_{hwc}$
$RGB_{win1\_bg}(1-\alpha_{win0})+RGB_{win0}\alpha_{win0}$
$RGB_{bg}(1-\alpha_{win1})+RGB_{win1}\alpha_{win1}$
top
Hw-cursor
win1
win0
Background
bottom
$RGB_{win0\_win1\_bg}(1-\alpha_{hwc})+RGB_{hwc}\alpha_{hwc}$
$RGB_{win0\_bg}(1-\alpha_{win1})+RGB_{win1}\alpha_{win1}$
$RGB_{bg}(1-\alpha_{win0})+RGB_{win0}\alpha_{win0}$

Win0 on top

Win1 on top

Fig. 7-7 VOP Alpha blending

4. RGB OVERLAY and YCbCr OVERLAY

VOP has a signal named sw_overlay_mode at DSP_CTRL2[4] to decide overlay in RGB or YUV color space.

RGB OVERLAY

All layers overlay in RGB color space. If win0 is YUV420/422/444, it can be converted to RGB888 by YUV2RGB use MPEG,JPEG and HDTV formula.

YCbCr OVERLAY

All layers overlay in YCbCr color space. When win0/1 input picture format is RGB, it can be converted to YUV444 using BT601 or BT709.

7.3.6 BCSH

The BCSH block is used for brightness, contrast, saturation and hue adjustment to YCbCr format image.

The following diagram shows the BCSH adjustment processing. All the factors should be set in the VOP registers.

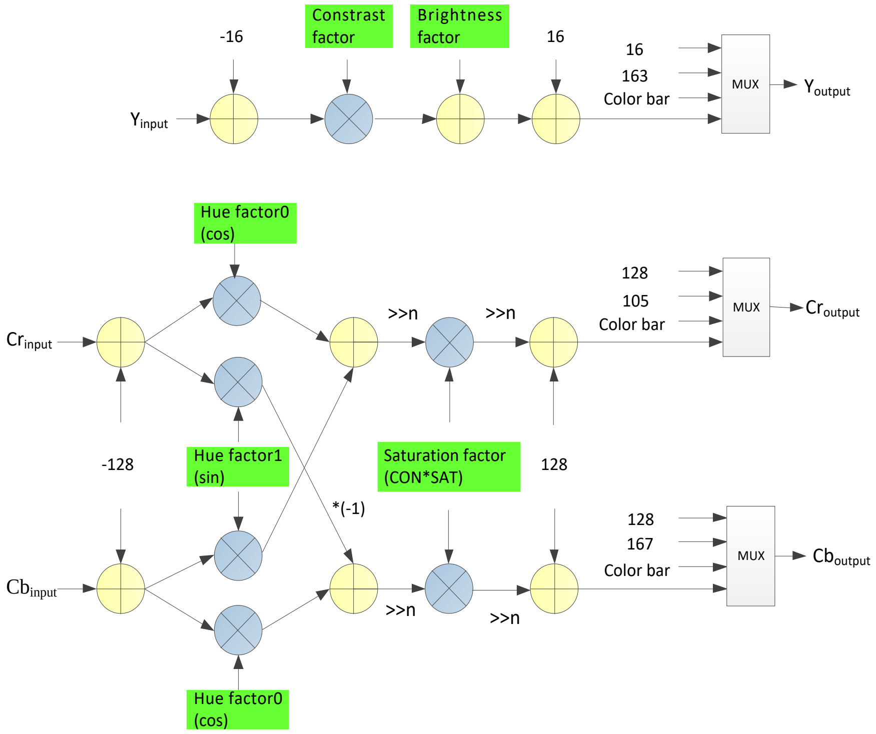


Fig. 7-8 VOP BCSH Diagram

7.3.7 Gamma Correction

Gamma correction is necessary because most monitors don't have a linear relationship between the voltage and the brightness, which results in your scene looking like it has too much contrast and the light falling off from the source outward, happens too quickly. The result can also be problematic if you are going into a composition program.
You can correct this by "Gamma Correction", which allows you to display the images and textures on your computer in an accurate manner.
Your screen is not linear, in that it displays the brightness unevenly. As a result, the image looks to be more high contrast than it should, you end up adding more lights or turning up the intensity, or you don't use the lighting in a realistic way that matches well with live action scenes. It also creates problems for you if you use software composition.
It consumes 256x8bit LUT for each channel. You can write gamma correction LUT through register" DSP_LUT_ADDR" one by one after set dsp_lut_en = 0.

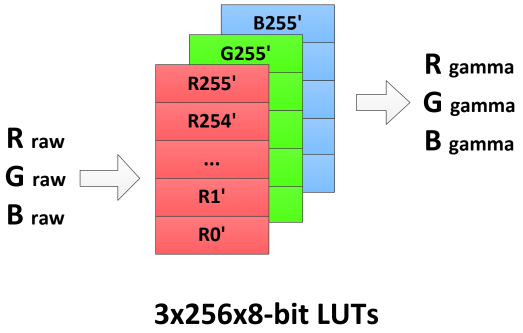


Fig. 7-9 VOP Gamma LUTs

7.3.8 Replication and dither down

1 Replication (dither up)

If the interface data bus is wider than the pixel format size, by programming the pixel components replication active/inactive, the MSB is replicated to the LSB of the interface data bus or the LSB is filled with 0s.

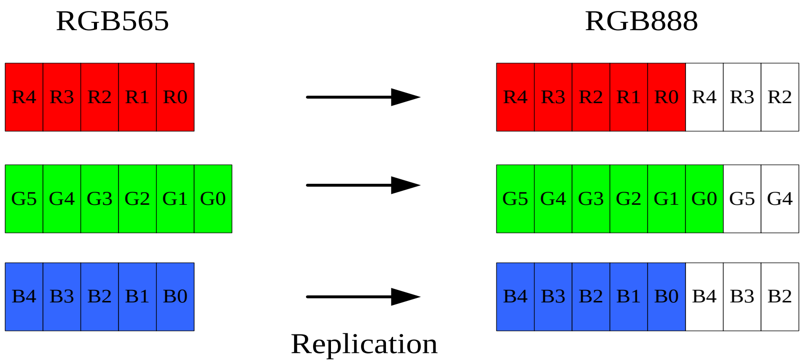


Fig. 7-10 VOP Replication

Dithering is used to improve the quality of display the pixel data in a lower color depth on the LCD panel. The Dithering algorithm is based on the (x,y) pixel position, the value of removed bits and frame counter.

2 Dither down

Here we support three kinds of dithering arithmetic. RGB888 to RGB666 has two ways, one is allegro, the other is FRC. RGB888 to RGB565 has only one arithmetic, which is allegro. When Dithering is not enabled, the MSBs of the pixel color components are output on the interface data bus if the interface data bus is smaller than the pixel format size.

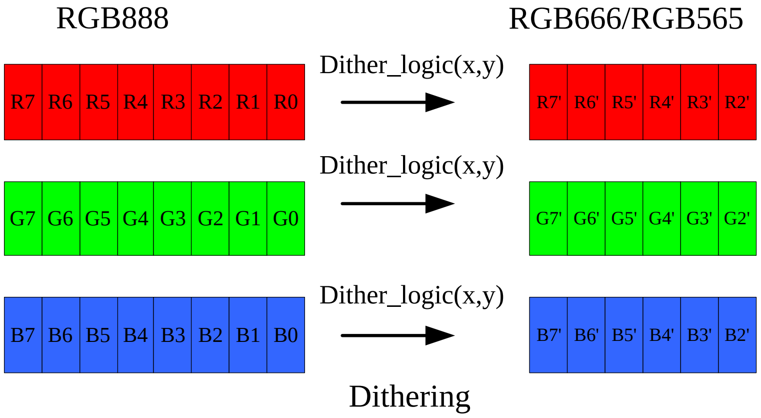


Fig. 7-11 vop dithering

The following figure is the recommended pattern picture in vop, you can configure different value of reg 0x170~0x184, to change the pattern picture.(default value is recommended values below)

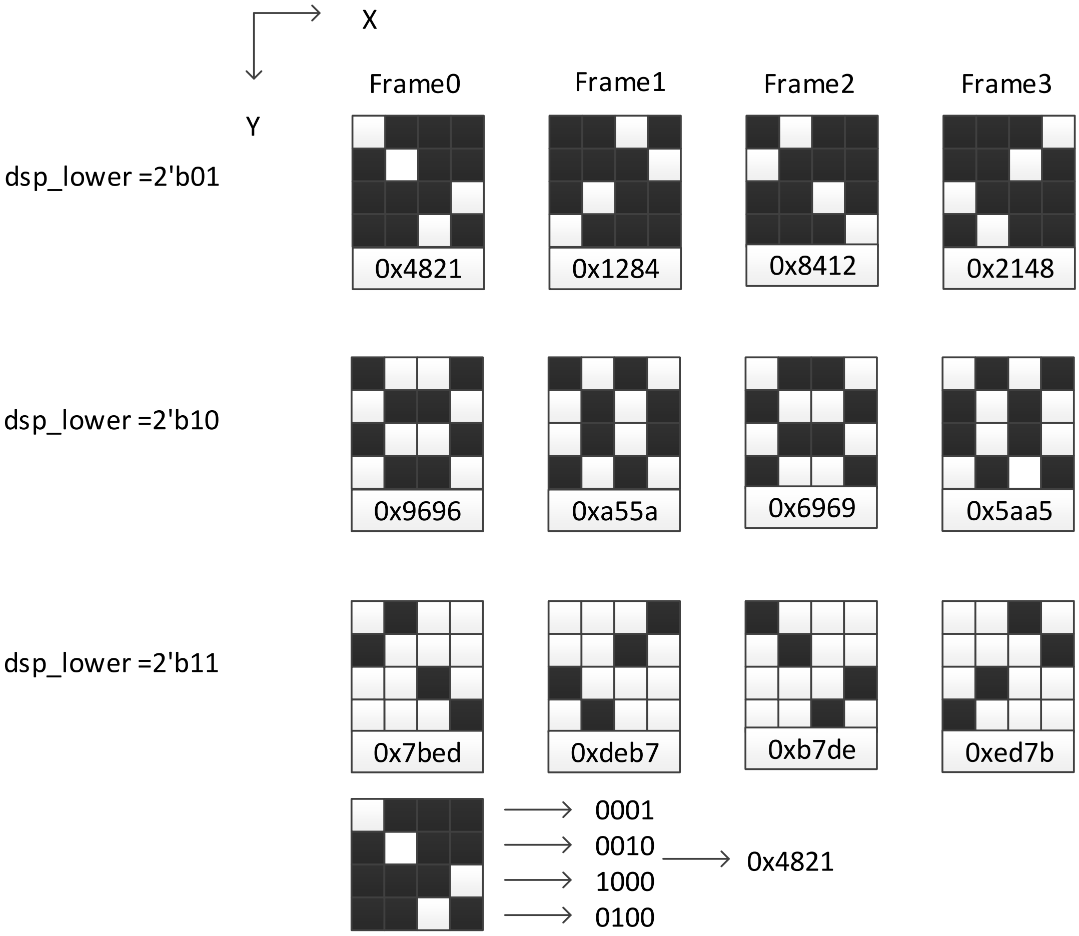


Fig. 7-12 frc pattern diagram

Recommended pattern:
0x170 : 0x12844821
0x174 : 0x21488412
0x178 : 0xa55a9696
0x17f : 0x5aa56969
0x180 : 0xdeb77bed
0x184 : 0xed7bb7de

7.4 Register Description

7.4.1 Internal Address Mapping

Slave address can be divided into different length for different usage, which is shown as follows.

7.4.2 Registers Summary

Name	Offset	Size	Reset Value	Description
VOP_LITE_REG_CFG_DONE	0x0000	W	0x00000000	Register config done flag
VOP_LITE_VERSION	0x0004	W	0x00000000	Version for vop
VOP_LITE_DSP_BG	0x0008	W	0x00000000	Display control register1
VOP_LITE_MCU	0x000c	W	0x00000000	MCU control register
VOP_LITE_SYS_CTRL0	0x0010	W	0x00000000	System control register
VOP_LITE_SYS_CTRL1	0x0014	W	0x00000000	Axi Bus interface control register
VOP_LITE_SYS_CTRL2	0x0018	W	0x00006000	System control register1 for immediate reg

Name	Offset	Size	Reset Value	Description
VOP_LITE_DSP_CTRL0	0x0020	W	0x00000101	Display control register0
VOP_LITE_DSP_CTRL1	0x0024	W	0x00000000	Display control register1
VOP_LITE_DSP_CTRL2	0x0028	W	0x00004000	Display control register2
VOP_LITE_VOP_STATUS	0x002c	W	0x00000000	Some vop module status
VOP_LITE_LINE_FLAG	0x0030	W	0x00000000	Line flag config register
VOP_LITE_INTR_EN	0x0034	W	0x00000000	Interrupt enable register
VOP_LITE_INTR_CLEAR	0x0038	W	0x00000000	Interrupt clear register
VOP_LITE_INTR_STATUS	0x003c	W	0x00000000	Interrupt raw status and interrupt status
VOP_LITE_WIN0_CTRL0	0x0050	W	0x00000000	Win0 control register0
VOP_LITE_WIN0_CTRL1	0x0054	W	0x00021220	Win0 control register1
VOP_LITE_WIN0_COLOR_KEY	0x0058	W	0x00000000	Win0 color key register
VOP_LITE_WIN0_VIR	0x005c	W	0x01400140	Win0 virtual stride
VOP_LITE_WIN0_YRGB_MST0	0x0060	W	0x00000000	Win0 YRGB memory start address 0
VOP_LITE_WIN0_CBR_MST0	0x0064	W	0x00000000	Win0 Cbr memory start address 0
VOP_LITE_WIN0_ACT_INFO	0x0068	W	0x00ef013f	Win0 active window width/height
VOP_LITE_WIN0_DSP_INFO	0x006c	W	0x00ef013f	Win0 display width/height on panel
VOP_LITE_WIN0_DSP_ST	0x0070	W	0x000a000a	Win0 display start point on panel
VOP_LITE_WIN0_SCL_FACTOR_YRGB	0x0074	W	0x10001000	Win0 YRGB scaling factor
VOP_LITE_WIN0_SCL_FACTOR_CBR	0x0078	W	0x10001000	Win0 CBR scaling factor
VOP_LITE_WIN0_SCL_OFFSET	0x007c	W	0x00000000	Win0 scaling start point offset
VOP_LITE_WIN0_ALPHA_CTRL	0x0080	W	0x00000000	Blending control register
VOP_LITE_WIN1_CTRL0	0x0090	W	0x00000000	Win1 control register0
VOP_LITE_WIN1_CTRL1	0x0094	W	0x00000580	Win1 control register0
VOP_LITE_WIN1_VIR	0x0098	W	0x00000000	Win1 virtual stride
VOP_LITE_WIN1_MST	0x00a0	W	0x00000000	Win1 memory start address
VOP_LITE_WIN1_DSP_INFO	0x00a4	W	0x00ef013f	Win1 display width/height on panel
VOP_LITE_WIN1_DSP_ST	0x00a8	W	0x000a000a	Win1 display start point on panel
VOP_LITE_WIN1_COLOR_KEY	0x00ac	W	0x00000000	Win1 color key register
VOP_LITE_WIN1_ALPHA_CTRL	0x00bc	W	0x00000000	Blending control register

Name	Offset	Size	Reset Value	Description
VOP_LITE_DSP_HTOTAL_HS_END	0x0100	W	0x014a000a	Panel scanning horizontal width and hsync pulse end point
VOP_LITE_DSP_HACT_ST_END	0x0104	W	0x000a014a	Panel active horizontal scanning start point and end point
VOP_LITE_DSP_VTOTAL_VS_END	0x0108	W	0x00fa000a	Panel scanning vertical height and vsync pulse end point
VOP_LITE_DSP_VACT_ST_END	0x010c	W	0x000a00fa	Panel active vertical scanning start point and end point
VOP_LITE_DSP_VS_ST_END_F1	0x0110	W	0x00000000	Vertical scanning start point and vsync pulse end point of even filed in interlace mode
VOP_LITE_DSP_VACT_ST_END_F1	0x0114	W	0x00000000	Vertical scanning active start point and end point of even filed in interlace mode
VOP_LITE_BCSH_CTRL	0x0160	W	0x00000000	Brightness/Contrast enhancement/Saturation/Hue control
VOP_LITE_BCSH_COL_BAR	0x0164	W	0x00000000	Colorbar YUV value
VOP_LITE_BCSH_BCS	0x0168	W	0x00000000	Brightness/Contrast enhancement/Saturation
VOP_LITE_BCSH_H	0x016c	W	0x00000000	Hue
VOP_LITE_FRC_LOWER01_0	0x0170	W	0x12844821	Frc algorithm configuration register 1
VOP_LITE_FRC_LOWER01_1	0x0174	W	0x21488412	Frc algorithm configuration register 2
VOP_LITE_FRC_LOWER10_0	0x0178	W	0xa55a9696	Frc algorithm configuration register 3
VOP_LITE_FRC_LOWER10_1	0x017c	W	0x5aa56969	Frc algorithm configuration register 4
VOP_LITE_FRC_LOWER11_0	0x0180	W	0xdeb77bed	Frc algorithm configuration register 5
VOP_LITE_FRC_LOWER11_1	0x0184	W	0xed7bb7de	Frc algorithm configuration register 6
VOP_LITE_MCU_RW_BYPASS_PORT	0x018c	W	0x00000000	MCU write/read port address
VOP_LITE_DBG_REG_SCAN_LINE	0x0190	W	0x00000000	Current line number of dsp timing
VOP_LITE_BLANKING_VALUE	0x01f4	W	0x01000000	The value of vsync blanking
VOP_LITE_FLAG_REG_FRM_VALID	0x01f8	W	0x00000000	Flag reg value after frame valid
VOP_LITE_FLAG_REG	0x01fc	W	0x00000000	Flag reg value before frame valid

Name	Offset	Size	Reset Value	Description
VOP_LITE_GAMMA_LUT_ADDR	0x0a00	W	0x00000000	Access entry for GAMMA LUT memory(size is word only)

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

7.4.3 Detail Register Description

VOP_LITE_REG_CFG_DONE

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:6	RO	0x0	reserved
5	RW	0x0	reg_load_sys_en vop system register config done flag In the first setting of the register, the new value was saved into the mirror register. When all the system register config finish(all reg except win0 win1), writing this register to enable the copyright of the mirror register to real register. Then register would be updated at the start of every frame.
4:3	RW	0x0	reserved
2	RW	0x0	reg_load_win1_en vop win1 register config done flag In the first setting of the register, the new value was saved into the mirror register. When all the win1 register config finish, writing this register to enable the copyright of the mirror register to real register. Then register would be updated at the start of every frame.
1	RW	0x0	reg_load_win0_en vop win0 register config done flag In the first setting of the register, the new value was saved into the mirror register. When all the win0 register config finish, writing this register to enable the copyright of the mirror register to real register. Then register would be updated at the start of every frame.
0	WO	0x0	reg_load_global_en vop register config done flag In the first setting of the register, the new value was saved into the mirror register. When all the register config finish, writing this register to enable the copyright of the mirror register to real register. Then register would be updated at the start of every frame.

VOP_LITE_VERSION

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:24	RO	0x00	major IP major vertion used for IP structure
23:16	RO	0x00	minor minor vertion big feature change under same structure
15:0	RO	0x0000	build rtl current svn number

VOP_LITE_DSP_BG
Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:16	RW	0x00	dsp_bg_red Background Red color
15:8	RW	0x00	dsp_bg_green Background Green color
7:0	RW	0x00	dsp_bg_blue Background Blue color

VOP_LITE_MCU
Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31	RW	0x0	mcu_type MCU LCD output SELECT
30	RW	0x0	mcu_bypass MCU LCD BYPASS MODE Select
29	RW	0x0	mcu_rs MCU LCD RS Select
28	RW	0x0	mcu_frame_st Write"1": MCU HOLD Mode Frame Start Read: MCU standby HOLD status
27	RW	0x0	mcu_hold_mode MCU HOLD Mode Select
26	RW	0x0	mcu_clk_sel MCU_CLK_SEL for MCU bypass 1'b1: MCU BYPASS sync with DCLK 1'b0: MCU BYPASS sync with HCLK
25:20	RW	0x00	mcu_rw_pend MCU_RW signal end point (0-63)
19:16	RW	0x0	mcu_rw_pst MCU_RW signal start point (0-15)
15:10	RW	0x00	mcu_cs_pend MCU_CS signal end point (0-63)

Bit	Attr	Reset Value	Description
9:6	RW	0x0	mcu_cs_pst MCU_CS signal start point (0-15)
5:0	RW	0x00	mcu_pix_total MCU LCD Interface writing period (1-63)

VOP_LITE_SYS_CTRL0
Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:0	RO	0x0	reserved

VOP_LITE_SYS_CTRL1
Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:21	RO	0x0	reserved
20:16	RW	0x00	sw_axi_max_outstand_num Max number of AXI read outstanding
15:13	RO	0x0	reserved
12	RW	0x0	sw_axi_max_outstand_en AXI read oustanding limited enable bit 1'b0: disable 1'b1: enable
11:8	RW	0x0	sw_noc_hurry_threshold Noc hurry threshold
7	RO	0x0	reserved
6:5	RW	0x0	sw_noc_hurry_value Noc hurry level
4	RW	0x0	sw_noc_hurry_en Noc hurry mode enable bit 1'b0: disable 1'b1: enable
3	RO	0x0	reserved
2:1	RW	0x0	sw_noc_qos_value Noc QoS level
0	RW	0x0	sw_noc_qos_en Noc QoS mode enable bit 1'b0: disable 1'b1: enable

VOP_LITE_SYS_CTRL2
Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:15	RO	0x0	reserved

Bit	Attr	Reset Value	Description
14	RW	0x1	fs_addr_mask_en frame start mask bit 1'b0: disable 1'b1: enable
13	RW	0x1	imd_global_regdone_en global config done enalbe 1'b0: DISABLE 1'b1: ENALBE
12	RW	0x0	imd_dsp_timing_imd 1'b0: timing reg valid immediately 1'b1: timing reg valid after frame start
11:8	RO	0x0	reserved
7	RW	0x0	sw_io_pad_clk_sel IO pad clk select bit 1'b0: normal dclk out 1'b1: gating dclk out
6	RW	0x0	imd_dsp_data_out_mode 1'b0: normal output mode 1'b1: output 24'b0
5	RO	0x0	reserved
4	RW	0x0	imd_yuv_clip YCrCb clip 1'b0: disable, YCbCr no clip 1'b1: enable, YCbCr clip before YCbCr2RGB *Y clip: 16~235, CBCR clip: 16~239
3	RW	0x0	imd_dsp_out_zero DEN、HSYNC、VSYNC output software control 1'b0: normal output 1'b1: all output '0' means:{hsync,vsync,den}={000}
2	RW	0x0	imd_vop_dma_stop VOP DMA stop mode 1'b0: disable 1'b1: enable * If DMA is working, the stop mode would not be active until current bus transfer is finished.
1	RW	0x0	imd_vop_standby_en VOP standby mode Writing "1" to turn VOP into standby mode, All the layer would disable and the data transfer from frame buffer memory would stop at the end of current frame. The output would be blank. When writing "0" to this bit, standby mode would disable and the VOP go back to work immediately. 1'b0: disable 1'b1: enable

Bit	Attr	Reset Value	Description
			* Black display is recommended before setting standby mode enable.
0	RW	0x0	imd_auto_gating_en VOP layer axi-clk auto gating enable 1'b0: disable 1'b1: enable

VOP_LITE_DSP_CTRL0
Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
13	RW	0x0	sw_core_clk_sel 1'b0: select dclk sclk 1'b1: select dclk_div sclk_div
12:5	RO	0x0	reserved
4	RW	0x0	rgb_den_pol DEN polarity 1'b0: positive 1'b1: negative when this channel dclk_en = 0, this signal used for control corresponding bit
3	RW	0x0	rgb_vsync_pol VSYNC polarity 1'b0: negative 1'b1: positive when this channel dclk_en = 0, this signal used for control corresponding bit
2	RW	0x0	rgb_hsync_pol HSYNC polarity 1'b0: negative 1'b1: positive when this channel dclk_en = 0, this signal used for control corresponding bit
1	RW	0x0	rgb_dclk_pol rgb dclk invert select bit 1'b0: rgb dclk inv disable 1'b1: rgb dclk inv enable
0	RW	0x1	rgb_dclk_en rgb dclk enable bit 1'b0: rgb dclk disable 1'b1: rgb dclk enable

VOP_LITE_DSP_CTRL1
Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:0	RO	0x0	reserved

VOP_LITE_DSP_CTRL2

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:16	RW	0x0	dsp_out_mode Display output format 4'b0001: Parallel 18-bit RGB666 output {6'b0,R[5:0],G[5:0],B[5:0]} 4'b0010: Parallel 16-bit RGB565 output {8'b0,R[4:0],G[5:0],B[4:0]} 4'b0100: Serial 2x16-bit RGB888x {8'b0,G[7:0],B[7:0]} + {16'b0,R[7:0]} 4'b1000: Serial 3x8-bit RGB888 {16'b0, B[7:0]}+{16'b0,G[7:0]}+{16'b0,R[7:0]} 4'b1100: Serial 3x8-bit RGB888 + dummy {16'b0, B[7:0]}+{16'b0,G[7:0]}+{16'b0,R[7:0]} + dummy Others: Reserved.
15	RW	0x0	dsp_black_en Black display mode When this bit enable, the pixel data output is all black(0x000000)
14	RW	0x1	dsp_blank_en Blank display mode When this bit enable, the hsync/vsync/den output is blank. means: {hsync,vsync,den}={110}
13	RO	0x0	reserved
12	RW	0x0	dsp_rg_swap Display output red and green swap enable 1'b0: RGB 1'b1: GRB
11	RW	0x0	dsp_rb_swap Display output red and blue swap enable 1'b0: RGB 1'b1: BGR
10	RO	0x0	reserved
9	RW	0x0	dsp_bg_swap Display output blue and green swap enable 1'b0: RGB 1'b1: RBG
8	RW	0x0	dither_down Dither-down enable 1'b0: disable 1'b1: enable

Bit	Attr	Reset Value	Description
7	RW	0x0	dither_down_sel dither down mode select 1'b0: allegro 1'b1: FRC
6	RW	0x0	dither_down_mode Dither-down mode 1'b0: RGB888 to RGB565 1'b1: RGB888 to RGB666
5	RW	0x0	dsp_lut_en Display LUT ram enable 1'b0: disable 1'b1: enable *This bit should be "0" when CPU updates the LUT, and should be "1" when Display LUT mode enable.
4	RW	0x0	sw_overlay_mode 1'b0: overlay in rgb domain 1'b1: overlay in yuv domain
3	RW	0x0	dsp_win0_top Win0 and Win1 position swap 1'b0: win1 on the top of win0 1'b1: win0 on the top of win1
2	RW	0x0	dither_up dither up RGB565 to RGB888 enable 1'b0: disable 1'b1: enable
1	RW	0x0	interlace_field_pol Interlace field polarity 1'b0: normal 1'b1: invert
0	RW	0x0	dsp_interlace Interlace display enable 1'b0: disable 1'b1: enable *This mode is related to the TVE output, the display timing of odd field must be set correctly. (vop_dsp_vs_st_end_f1/vop_dsp_vact_end_f1)

VOP_LITE_VOP_STATUS

Address: Operational Base + offset (0x002c)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RO	0x0	dma_stop_valid
3	RO	0x0	reserved

Bit	Attr	Reset Value	Description
2	RO	0x0	int_raw_dma_finish dma finish raw singal 1'b0: not finish 1'b1: finish
1	RO	0x0	idle_mmu_ff1 mmu idle status 1'b0: busy 1'b1: idle
0	RO	0x0	dsp_blanking_en_async_aff2 1'b0: blanking disable 1'b1: blanking enable

VOP_LITE_LINE_FLAG

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	dsp_line_flag1_num Line number of the Line flag interrupt 1 The display line number when the flag interrupt 1 occur, the range is (0~ DSP_VTOTAL-1).
15:12	RO	0x0	reserved
11:0	RW	0x000	dsp_line_flag0_num Line number of the Line flag interrupt 0 The display line number when the flag interrupt occur, the range is (0~ DSP_VTOTAL-1).

VOP_LITE_INTR_EN

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:10	RO	0x0	reserved
9	RW	0x0	dma_frm_fsh_intr_en dma frame finish interrupt enable 1'b0: DISABLE 1'b1: ENABLE
8	RW	0x0	dsp_hold_valid_intr_en display hold valid interrupt enable 1'b0: DISABLE 1'b1: ENABLE

Bit	Attr	Reset Value	Description
7	RW	0x0	win1_empty_intr_en win1 data empty interrupt enable 1'b0: DISABLE 1'b1: ENABLE
6	RW	0x0	win0_empty_intr_en win0 data empty interrupt enable 1'b0: DISABLE 1'b1: ENABLE
5	RW	0x0	bus_error_intr_en Bus error Interrupt enable 1'b0: DISABLE 1'b1: ENABLE
4	RW	0x0	line_flag1_intr_en Line flag 1 Interrupt enable 1'b0: DISABLE 1'b1: ENABLE
3	RW	0x0	line_flag0_intr_en Line flag 0 Interrupt enable 1'b0: DISABLE 1'b1: ENABLE
2	RW	0x0	addr_same_intr_en memory start addr same interruption enable 1'b0: DISABLE 1'b1: ENABLE
1	RW	0x0	fs1_intr_en Frame start 1 interrupt enable 1'b0: DISABLE 1'b1: ENABLE
0	RW	0x0	fs0_intr_en Frame start 0 interrupt enable 1'b0: DISABLE 1'b1: ENABLE

VOP_LITE_INTR_CLEAR

Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9	W1C	0x0	dma_frm_fsh_intr_clr dma frame finish interrupt clear
8	W1C	0x0	dsp_hold_valid_intr_clr display hold valid interrupt clear
7	W1C	0x0	win1_empty_intr_clr win1 data empty interrupt clear

Bit	Attr	Reset Value	Description
6	W1C	0x0	win0_empty_intr_clr win0 data empty interrupt clear
5	W1C	0x0	bus_error_intr_clr Bus error Interrupt clear
4	W1C	0x0	line_flag1_intr_clr Line flag 1 Interrupt clear
3	W1C	0x0	line_flag0_intr_clr Line flag 0 Interrupt clear
2	W1C	0x0	addr_same_intr_clr memory start address same interruption clear
1	W1C	0x0	fs1_intr_clr Frame start 1 interrupt clear
0	W1C	0x0	fs0_intr_clr Frame start interrupt 0 clear

VOP_LITE_INTR_STATUS

Address: Operational Base + offset (0x003c)

Bit	Attr	Reset Value	Description
31:26	RO	0x0	reserved
25	RO	0x0	dma_frm_fsh_intr_raw_sts dma frame finish interrupt raw status
24	RO	0x0	dsp_hold_valid_intr_raw_sts display hold valid interrupt raw status
23	RO	0x0	win1_empty_intr_raw_sts win1 data empty interrupt raw status
22	RO	0x0	win0_empty_intr_raw_sts win0 data empty interrupt raw status
21	RO	0x0	bus_error_intr_raw_sts Bus error Interrupt raw status
20	RO	0x0	line_flag1_intr_raw_sts Line flag 1 Interrupt raw status
19	RO	0x0	line_flag0_intr_raw_sts Line flag 0 Interrupt raw status
18	RW	0x0	addr_same_intr_raw_sts memory start addr same interruption raw status
17	RO	0x0	fs1_intr_raw_sts Frame start interrupt raw status(when memory start addr are same,no interruption)
16	RO	0x0	fs0_intr_raw_sts Frame start raw interrupt status
15	RO	0x0	mmu_intr_status mmu irq interrupt status
14:10	RO	0x0	reserved

Bit	Attr	Reset Value	Description
9	RO	0x0	dma_frm_fsh_intr_sts dma frame finish interrupt status
8	RO	0x0	dsp_hold_valid_intr_sts display hold valid interrupt status
7	RO	0x0	win1_empty_intr_sts win1 data empty interrupt status
6	RO	0x0	win0_empty_intr_sts win0 data empty interrupt status
5	RO	0x0	bus_error_intr_sts Bus error Interrupt status
4	RO	0x0	line_flag1_intr_sts Line flag 1 Interrupt status
3	RO	0x0	line_flag0_intr_sts Line flag 0 Interrupt status
2	RW	0x0	addr_same_intr_sts memory start addr same interruption status
1	RO	0x0	fs1_intr_sts Frame status(when memory start addr are same, no interruption)
0	RO	0x0	fs0_intr_sts Frame start 0 interrupt status

VOP_LITE_WIN0_CTRL0

Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19	RW	0x0	win0_cbr_deflick Win0 Cbr deflick mode 1'b0: disable 1'b1: enable
18	RW	0x0	win0_yrgb_deflick Win0 YRGB deflick mode 1'b0: disable 1'b1: enable
17:16	RO	0x0	reserved
15	RW	0x0	win0_uv_swap Win0 CbCr swap 1'b0: CrCb 1'b1: CbCr
14	RW	0x0	win0_mid_swap Win0 Y middle 8-bit swap 1'b0: Y3Y2Y1Y0 1'b1: Y3Y1Y2Y0

Bit	Attr	Reset Value	Description
13	RW	0x0	win0_alpha_swap Win0 RGB alpha swap 1'b0: ARGB 1'b1: RGBA
12	RW	0x0	win0_rb_swap Win0 RGB Red and Blue swap 1'b0: RGB 1'b1: BGR
11:10	RW	0x0	win0_csc_mode Win0 YUV2RGB Color space conversion: 2'b00/11: mpeg 2'b01: hd 2'b10: jpeg reused by win0 r2y color space conversion: 2'bX0: BT601 2'bX1: BT709
9	RW	0x0	win0_no_outstanding Win0 AXI master read outstanding 1'b0: enable 1'b1: disable
8	RW	0x0	win0_interlace_read Win0 interlace read mode 1'b0: disable 1'b1: enable
7:4	RO	0x0	reserved
3:1	RW	0x0	win0_data_fmt Win0 source data Format 3'b000: ARGB888 3'b001: RGB888 3'b010: RGB565 3'b100: YCbCr420 3'b101: YCbCr422 3'b110: YCbCr444 others: reserved
0	RW	0x0	win0_en Win0 enable bit 1'b0: Win0 layer disable 1'b1: Win0 layer enable

VOP_LITE_WIN0_CTRL1

Address: Operational Base + offset (0x0054)

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved

Bit	Attr	Reset Value	Description
19:16	RW	0x2	sw_win0_cbr0_rid axi read id of win0 cbr channel
15:12	RW	0x1	sw_win0_yrgb0_rid axi read id of win0 yrgb channel
11	RO	0x0	reserved
10:8	RW	0x2	win0_cbr_axi_gather_num
7:4	RW	0x2	win0_yrgb_axi_gather_num
3:2	RW	0x0	win0_dma_burst_length DMA read Burst length 2'b00: burst16 (burst 15 in rgb888 pack mode) 2'b01: burst8 (burst 12 in rgb888 pack mode) 2'b10: burst4 (burst 6 in rgb888 pack mode)
1	RW	0x0	win0_cbr_axi_gather_en win0 cb/cr dma channel AXI read transfer gather 1'b0: disable 1'b1: enable
0	WO	0x0	win0_yrgb_axi_gather_en win0 yrgb dma channel AXI read transfer gather 1'b0: disable 1'b1: enable

VOP_LITE_WIN0_COLOR_KEY

Address: Operational Base + offset (0x0058)

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24	RW	0x0	win0_key_en Win0 transparency color key enable 1'b0: disable 1'b1: enable
23:0	RW	0x000000	win0_key_color Win0 key color

VOP_LITE_WIN0_VIR

Address: Operational Base + offset (0x005c)

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28:16	RW	0x0140	win0_cbr_vir_stride Number of words of Win0 cbr Virtual width UV420: ceil(win0_cbr_vir_stride/4) UV422: ceil(win0_cbr_vir_stride/4) UV444: ceil(win0_cbr_vir_stride/2)
15:13	RO	0x0	reserved

Bit	Attr	Reset Value	Description
12:0	RW	0x0140	win0_yrgb_vir_stride Win0 Virtual stride Number of words of Win0 Virtual width ARGB888: win0_yrgb_vir_stride RGB888: (win0_yrgb_vir_stride*3/4) + win0_yrgb_vir_stride%3) RGB565: ceil(win0_yrgb_vir_stride/2) YUV: ceil(win0_yrgb_vir_stride/4)

VOP_LITE_WIN0_YRGB_MST0

Address: Operational Base + offset (0x0060)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	win0_yrgb0_mst win0 YRGB frame buffer memory start address 0

VOP_LITE_WIN0_CBR_MST0

Address: Operational Base + offset (0x0064)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	win0_cbr0_mst win0 CBR frame buffer memory start address 0

VOP_LITE_WIN0_ACT_INFO

Address: Operational Base + offset (0x0068)

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28:16	RW	0x00ef	win0_act_height Win0 active(original) window height win_act_height = (win0 vertical size-1)
15:13	RO	0x0	reserved
12:0	RW	0x013f	win0_act_width Win0 active(original) window width win_act_width = (win0 horizontial size-1)

VOP_LITE_WIN0_DSP_INFO

Address: Operational Base + offset (0x006c)

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26:16	RW	0x0ef	dsp_win0_height Win0 display window height win0_dsp_height = (win0 vertical size-1)
15:11	RO	0x0	reserved
10:0	RW	0x13f	dsp_win0_width Win0 display window width win0_dsp_width = (win0 horizontal size-1)

VOP_LITE_WIN0_DSP_ST

Address: Operational Base + offset (0x0070)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x00a	dsp_win0_yst Win0 vertical start point(y) of the Panel scanning
15:12	RO	0x0	reserved
11:0	RW	0x00a	dsp_win0_xst Win0 horizontal start point(x) of the Panel scanning

VOP_LITE_WIN0_SCL_FACTOR_YRGB

Address: Operational Base + offset (0x0074)

Bit	Attr	Reset Value	Description
31:16	RW	0x1000	win0_vs_factor_yrgb Win0 YRGB vertical scaling factor: factor=((VOP_WIN0_ACT_INFO[31:16])/(VOP_WIN0_DSP_INFO[31:16]))*2^12
15:0	RW	0x1000	win0_hs_factor_yrgb Win0 YRGB horizontal scaling factor: factor=((VOP_WIN0_ACT_INFO[15:0])/(VOP_WIN0_DSP_INFO[15:0]))*2^12

VOP_LITE_WIN0_SCL_FACTOR_CBR

Address: Operational Base + offset (0x0078)

Bit	Attr	Reset Value	Description
31:16	RW	0x1000	win0_vs_factor_cbr Win0 CBR vertical scaling factor YCbCr420: factor=((VOP_WIN0_ACT_INFO[31:16]/2)/(VOP_WIN0_DSP_INFO[31:16]))*2^12 YCbCr422,YCbCr444: factor=((VOP_WIN0_ACT_INFO[31:16])/(VOP_WIN0_DSP_INFO[31:16]))*2^12
15:0	RW	0x1000	win0_hs_factor_cbr Win0 CBR horizontal scaling factor YCbCr422,YCbCr420: factor=((VOP_WIN0_ACT_INFO[15:0]/2)/(VOP_WIN0_DSP_INFO[15:0]))*2^12 YCbCr444: factor=((VOP_WIN0_ACT_INFO[15:0])/(VOP_WIN0_DSP_INFO[15:0]))*2^12

VOP_LITE_WIN0_SCL_OFFSET

Address: Operational Base + offset (0x007c)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	win0_vs_offset_cbr Cbr Vertical scaling start point offset (0x00~0xff)/0x100 = 0~0.99
23:16	RW	0x00	win0_vs_offset_yrgb Y Vertical scaling start point offset (0x00~0xff)/0x100 = 0~0.99
15:8	RW	0x00	win0_hs_offset_cbr Cbr Horizontal scaling start point offset (0x00~0xff)/0x100 = 0~0.99
7:0	RW	0x00	win0_hs_offset_yrgb Y Horizontal scaling start point offset (0x00~0xff)/0x100 = 0~0.99

VOP_LITE_WIN0_ALPHA_CTRL

Address: Operational Base + offset (0x0080)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:4	RW	0x00	win0_alpha_value Win0 alpha blending value
3	RW	0x0	win0_alpha_sat_mode win0 alpha satuation mode select 1'b0: alpha value no change 1'b1: alpha = alpha + alpha[7]
2	RW	0x0	win0_alpha_pre_mul win1 alpha Premultiplied control 1'b0: Non-premultiplied alpha 1'b1: Premultiplied alpha
1	RW	0x0	win0_alpha_mode Win0 alpha mode 1'b0: user-defined alpha 1'b1: per-pixel alpha
0	RW	0x0	win0_alpha_en Win0 alpha blending enable 1'b0: disable 1'b1: enable

VOP_LITE_WIN1_CTRL0

Address: Operational Base + offset (0x0090)

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
13	RW	0x0	win1_alpha_swap Win1 RGB alpha swap 1'b0: ARGB 1'b1: RGBA

Bit	Attr	Reset Value	Description
12	RW	0x0	win1_rb_swap Win1 RGB Red and Blue swap 1'b0: RGB 1'b1: BGR
11:10	RO	0x0	reserved
9	RW	0x0	win1_no_outstanding Win1 AXI master read outstanding 1'b0: enable 1'b1: disable
8	RW	0x0	win1_interlace_read Win1 interlace read mode 1'b0: disable 1'b1: enable
7	RO	0x0	reserved
6:4	RW	0x0	win1_data_fmt Win1 source Format 3'b000: ARGB888 3'b001: RGB888 3'b010: RGB565 others: reserved
3:1	RO	0x0	reserved
0	RW	0x0	win1_en Win1 enable bit 1'b0: Win1 layer disable 1'b1: Win1 layer enable

VOP_LITE_WIN1_CTRL1

Address: Operational Base + offset (0x0094)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:8	RW	0x5	sw_win1_rid axi read id of win1 channel
7:4	RW	0x8	win1_axi_gather_num
3:2	RW	0x0	win1_dma_burst_length DMA read Burst length 2'b00: burst16 (burst 15 in rgb888 pack mode) 2'b01: burst8 (burst 12 in rgb888 pack mode) 2'b10: burst4 (burst 6 in rgb888 pack mode)
1	RO	0x0	reserved
0	RW	0x0	win1_axi_gather_en win1 dma channel AXI read transfer gather 1'b0: disable 1'b1: enable

VOP_LITE_WIN1_VIR
Address: Operational Base + offset (0x0098)

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12:0	RW	0x0000	win1_vir_stride Win1 virtual stride Number of words of Win1 Virtual width ARGB888: win1_vir_width RGB888: (win1_vir_width*3/4) + (win1_vir_width%3) RGB565: ceil(win1_vir_width/2)

VOP_LITE_WIN1_MST
Address: Operational Base + offset (0x00a0)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	win1_mst Win1 frame buffer memory start address

VOP_LITE_WIN1_DSP_INFO
Address: Operational Base + offset (0x00a4)

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26:16	RW	0x0ef	dsp_win1_height Win1 display window height win1_dsp_height = (win1 dsp vertical size-1)
15:11	RO	0x0	reserved
10:0	RW	0x13f	dsp_win1_width Win1 display window width win1_dsp_width = (win1 dsp horizontal size-1)

VOP_LITE_WIN1_DSP_ST
Address: Operational Base + offset (0x00a8)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x00a	dsp_win1_yst Win1 vertical start point(y) of the Panel scanning
15:12	RO	0x0	reserved
11:0	RW	0x00a	dsp_win1_xst Win1 horizontal start point(x) of the Panel scanning

VOP_LITE_WIN1_COLOR_KEY
Address: Operational Base + offset (0x00ac)

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved

Bit	Attr	Reset Value	Description
24	RW	0x0	win1_key_en Win1 transparency color key enable 1'b0: disable 1'b1: enable
23:0	RW	0x000000	win1_key_color Win1 key color

VOP_LITE_WIN1_ALPHA_CTRL

Address: Operational Base + offset (0x00bc)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:4	RW	0x00	win1_alpha_value Win1 alpha blending value
3	RW	0x0	win1_alpha_sat_mode win1 alpha satuation mode select 1'b0: alpha value no change 1'b1: alpha = alpha + alpha[7]
2	RW	0x0	win1_alpha_pre_mul win1 alpha Premultiplied control 1'b0: Non-premultiplied alpha 1'b1: Premultiplied alpha
1	RW	0x0	win1_alpha_mode Win1 alpha mode 1'b0: user-defined alpha 1'b1: per-pixel alpha
0	RW	0x0	win1_alpha_en Win1 alpha blending enable 1'b0: disable 1'b1: enable

VOP_LITE_DSP_HTOTAL_HS_END

Address: Operational Base + offset (0x0100)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x14a	dsp_htotal Panel display scanning horizontal period
15:12	RO	0x0	reserved
11:0	RW	0x00a	dsp_hs_end Panel display scanning hsync pulse width

VOP_LITE_DSP_HACT_ST_END

Address: Operational Base + offset (0x0104)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved

Bit	Attr	Reset Value	Description
27:16	RW	0x00a	dsp_hact_st Panel display scanning horizontal active start point
15:12	RO	0x0	reserved
11:0	RW	0x14a	dsp_hact_end Panel display scanning horizontal active end point

VOP_LITE_DSP_VTOTAL_VS_END
Address: Operational Base + offset (0x0108)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x0fa	dsp_vtotal Panel display scanning vertical period
15:12	RO	0x0	reserved
11:0	RW	0x00a	dsp_vs_end Panel display scanning vsync pulse width

VOP_LITE_DSP_VACT_ST_END
Address: Operational Base + offset (0x010c)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x00a	dsp_vact_st Panel display scanning vertical active start point
15:12	RO	0x0	reserved
11:0	RW	0x0fa	dsp_vact_end Panel display scanning vertical active end point

VOP_LITE_DSP_VS_ST_END_F1
Address: Operational Base + offset (0x0110)

Bit	Attr	Reset Value	Description
31:0	RO	0x0	reserved

VOP_LITE_DSP_VACT_ST_END_F1
Address: Operational Base + offset (0x0114)

Bit	Attr	Reset Value	Description
31:0	RO	0x0	reserved

VOP_LITE_BCSH_CTRL
Address: Operational Base + offset (0x0160)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	sw_bcsh_r2y_en 1'b0: bypass 1'b1: enable

Bit	Attr	Reset Value	Description
6	RW	0x0	sw_bcsh_y2r_en 1'b0: bypass 1'b1: enable
5:4	RW	0x0	sw_bcsh_y2r_csc_mode Color space conversion: 2'b00/11: mpeg 2'b01: hd 2'b10: jpeg
3:2	RW	0x0	video_mode 2'b00: black 2'b01: blue 2'b10: color bar 2'b11: normal video
1	RW	0x0	sw_bcsh_r2y_csc_mode Color space conversion: 1'b0: BT601 1'b1: BT709
0	RW	0x0	bcsh_en 1'b0: bcsh bypass 1'b1: bcsh enable

VOP_LITE_BCSH_COL_BAR
Address: Operational Base + offset (0x0164)

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:16	RW	0x00	color_bar_v
15:8	RW	0x00	color_bar_u
7:0	RW	0x00	color_bar_y

VOP_LITE_BCSH_BCS
Address: Operational Base + offset (0x0168)

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24:16	RW	0x000	sat_con Saturation*Contrast*256:{0,1.992*1.992}
15:8	RW	0x00	contrast Contrast*256:{0,1.992}
7:6	RO	0x0	reserved
5:0	RW	0x00	brightness Brightness:{-128,127}

VOP_LITE_BCSH_H
Address: Operational Base + offset (0x016c)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:8	RW	0x00	cos_hue cos hue value
7:0	RW	0x00	sin_hue sin hue value

VOP_LITE_FRC_LOWER01_0

Address: Operational Base + offset (0x0170)

Bit	Attr	Reset Value	Description
31:16	RW	0x1284	lower01_frm1
15:0	RW	0x4821	lower01_frm0

VOP_LITE_FRC_LOWER01_1

Address: Operational Base + offset (0x0174)

Bit	Attr	Reset Value	Description
31:16	RW	0x2148	lower01_frm3
15:0	RW	0x8412	lower01_frm2

VOP_LITE_FRC_LOWER10_0

Address: Operational Base + offset (0x0178)

Bit	Attr	Reset Value	Description
31:16	RW	0xa55a	lower10_frm1
15:0	RW	0x9696	lower10_frm0

VOP_LITE_FRC_LOWER10_1

Address: Operational Base + offset (0x017c)

Bit	Attr	Reset Value	Description
31:16	RW	0x5aa5	lower10_frm3
15:0	RW	0x6969	lower10_frm2

VOP_LITE_FRC_LOWER11_0

Address: Operational Base + offset (0x0180)

Bit	Attr	Reset Value	Description
31:16	RW	0xdeb7	lower11_frm1
15:0	RW	0x7bed	lower11_frm0

VOP_LITE_FRC_LOWER11_1

Address: Operational Base + offset (0x0184)

Bit	Attr	Reset Value	Description
31:16	RW	0xed7b	lower11_frm3
15:0	RW	0xb7de	lower11_frm2

VOP_LITE_MCU_RW_BYPASS_PORT

Address: Operational Base + offset (0x018c)

Bit	Attr	Reset Value	Description
31:0	RW	0x0	MCU write/read port address

VOP_LITE_DBG_REG_SCAN_LINE

Address: Operational Base + offset (0x0190)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RO	0x000	scan_line_num

VOP_LITE_BLANKING_VALUE

Address: Operational Base + offset (0x01f4)

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24	RW	0x1	blanking_value_config_en
23:0	WO	0x000000	sw_blanking_value

VOP_LITE_FLAG_REG_FRM_VALID

Address: Operational Base + offset (0x01f8)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	flag_reg_frm_valid valid by frame start

VOP_LITE_FLAG_REG

Address: Operational Base + offset (0x01fc)

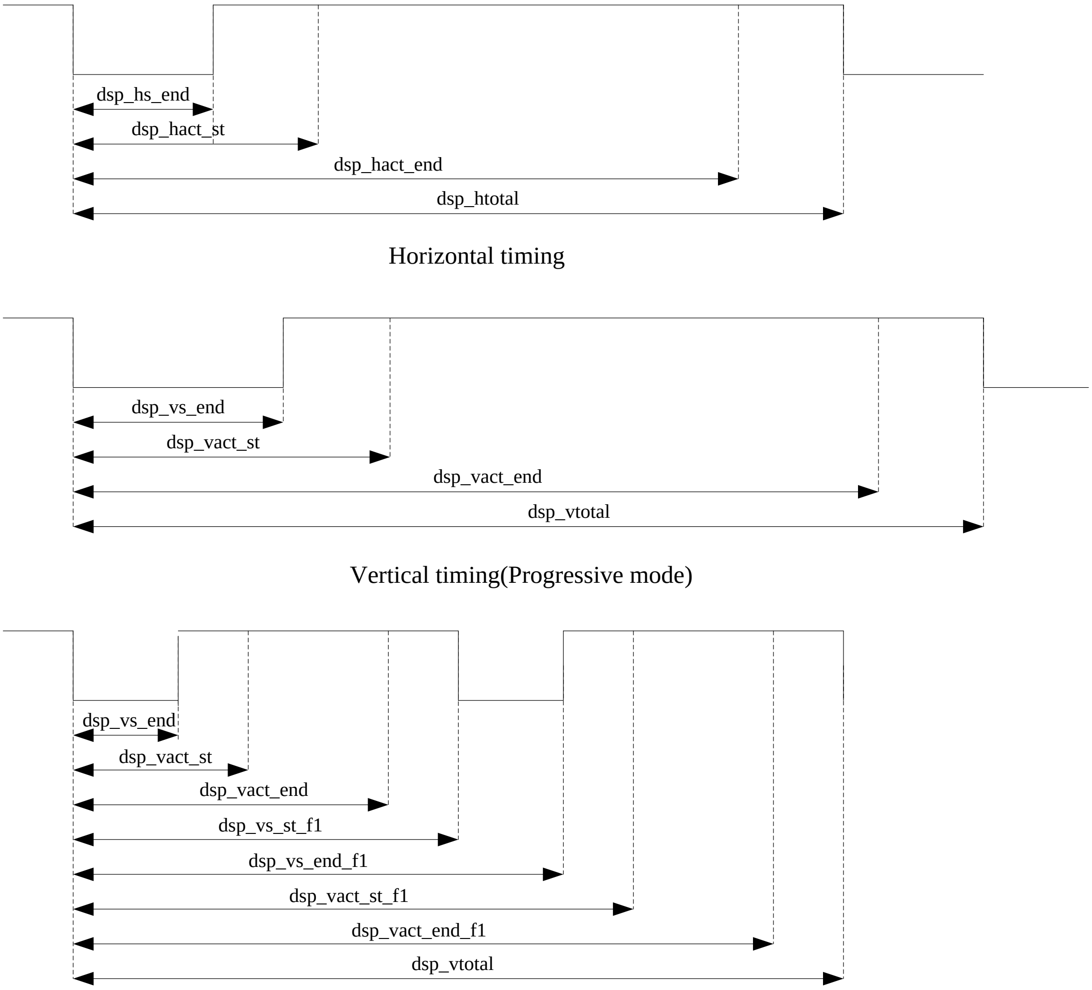
Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	flag_reg

VOP_LITE_GAMMA_LUT_ADDR

Address: Operational Base + offset (0x0a00)

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:0	RW	0x000000	gamma_lut_addr note: SIZE: 24X256 used for panel GAMMA adjustment, base address: 0x0a00 -- 0x0dff

7.5 Timing Diagram



Horizontal timing

Vertical timing(Progressive mode)

Vertical timing(Interlace mode)

Fig. 7-13 VOP RGB interface timing setting

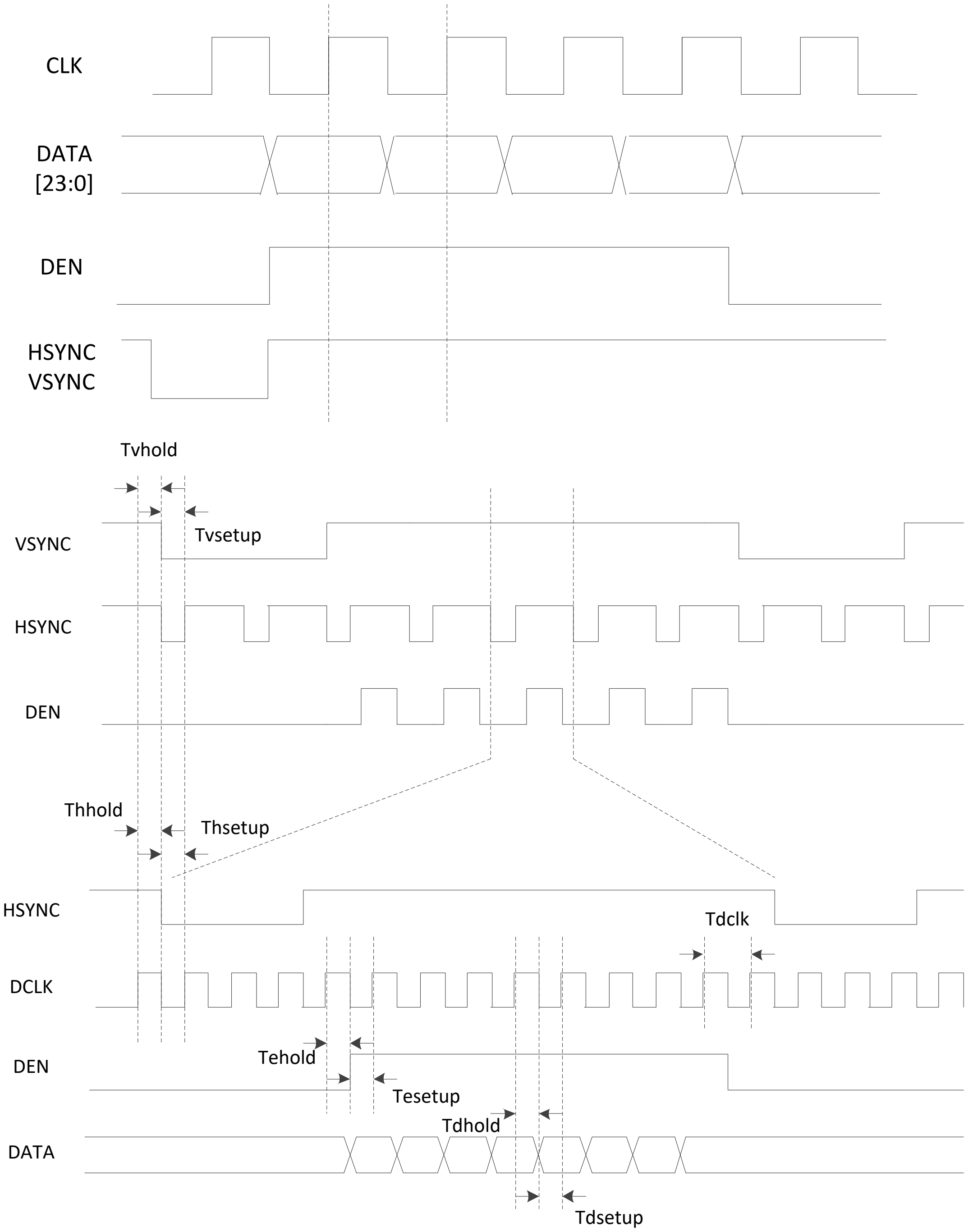


Fig. 7-18 VOP RGB Interface Timing(SDR)

7.6 Interface Description

7.6.1 VOP Outputs

VOP supports RGB, MCU output. the MCU output interface is same as RGB interface. VOP is suitable for different display mode by different usage, which is shown as follows.

Table 7-3 VOP Control Pins Definition

Display mode	***RGB Parallel 18-bit***	***RGB Parallel 16-bit***
DCLK	DCLK	DCLK
VSYNC	VSYNC	VSYNC
HSYNC	HSYNC	HSYNC
DEN	DEN	DEN

Display mode	RGB Parallel 18-bit	RGB Parallel 16-bit
DATA	DATA[17:0]	DATA[15:0]

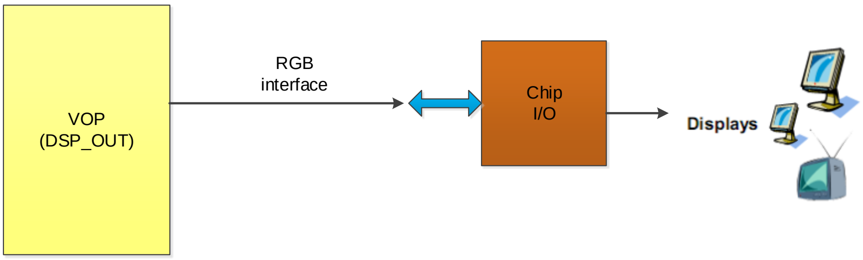


Fig. 7-14 VOP Display output for peripherals

7.7 Application Notes

7.7.1 DMA transfer mode

There are three DMA transfer modes for loading win0 or win1 frame data determined by following parameters(X=0,1):
dma_burst_length
winX_no_outstanding
winX_gather_en
winX_gather_thres

7.7.2 auto outstanding transfer mode(random transfer)

When winX_no_outstanding is 0, multi-bursts transfer command could be sent out to AXI master interface continuously if the internal memory has enough space to store new data. The continuous random burst number is in the range of 1 to 4, mainly depending on the empty level of internal memory, dma_burst_length, data format and active image width.

1. configured outstanding transfer mode(fixed transfer)

When winX_gather_en is 1, fixed-number of bursts transfer command should be sent out to AXI master interface continuously if the internal memory has enough space to store new data. The fixed-number is determined by winX_gather_thres. Since the internal memory size is limited, there is some restriction for the winX_gather_thres as follows.

Table 7-4 Gather configuration for all format

Gather Threshold	dma_burst_length =2'b00(burst16)	dma_burst_length =2'b01(burst8)	dma_burst_length =2'b10(burst4)
YCbCr420 YCbCr422 YCbCr444	0	0,1,2	0,1,2,3
ARGB888 RGB888 RGB565	0,1,2,3	0,1,2,3	0,1,2,3
8BPP	0,1,2,3	0,1,2,3	0,1,2,3

7.7.3 GAMMA LUT

When dsp_lut_en is 0, the DSP LUT data should be refreshed by software. i.e, writing dsp lut data to the internal memory with the start address DSP_LUT_MST. The memory size is

256x24, i.e, lower 24bits valid, and the writing data number is determined by software.

7.7.4 DMA control（QoS/Hurry/Outstanding）

If you want to get higher priority for VOP to access external memory when the frame data is urgent, a QoS and hurry request can be generated and sent out basing on the configured values:
sw_noc_qos_en: SYS_CTRL1[0]
sw_noc_qos_value: SYS_CTRL1[2:1]
sw_noc_hurry_en: SYS_CTRL1[4]
sw_noc_hurry_value: SYS_CTRL1[6:5]
sw_noc_hurry_threshold: SYS_CTRL1[11:8]
sw_axi_max_outstand_en: SYS_CTRL1[12]
sw_axi_max_outstand_num: SYS_CTRL1[20:16]
QoS request for higher bus priority for win1
NOC hurry for higher bus priority for VIO when win0 needs higher priority.
Max Outstanding num is configurable, but limited at 31 when mmu enables or 32 when mmu disables.

7.7.5 Interrupt

VOP interrupt is comprised of 10 interrupt sources:

- dsp hold interrupt
- frame start0 interrupt
- frame start1 interrupt
- address same interrupt
- line flag0 interrupt
- line flag1 interrupt
- bus error interrupt
- win0 empty interrupt
- win1 empty interrupt
- dma finish interrupt

Every interrupt has independent interrupt enable signal(VOP_INT_EN), interrupt clear signal(VOP_INT_CLR) and interrupt raw status signal(VOP_INT_STATUS).
There is only one interrupt combined with all this interrupt signals to CPU, and it high active.
Dma finish interrupt is used for changing DDR frequency. This interrupt will be asserted when dma finish getting all the Pixel DATA every frame. This interrupt is asserted at the end of the frame.
Address same interrupt will be asserted at this kind of scenario that all the memory start address configured in VOP reg are same compared with the former frame. This interrupt is asserted at the beginning of frame start.
The difference between frame start0 interrupt and frame start1 interrupt is that frame start0 interrupt will be asserted every frame, while the frame start1 interrupt will be masked when all the memory start address are same.

7.7.6 RGB display mode

RGB display mode is used for RGB panel display. It is a continuous frames display mode.

Copyright 2018 © FuZhou Rockchip Electronics Co., Ltd.

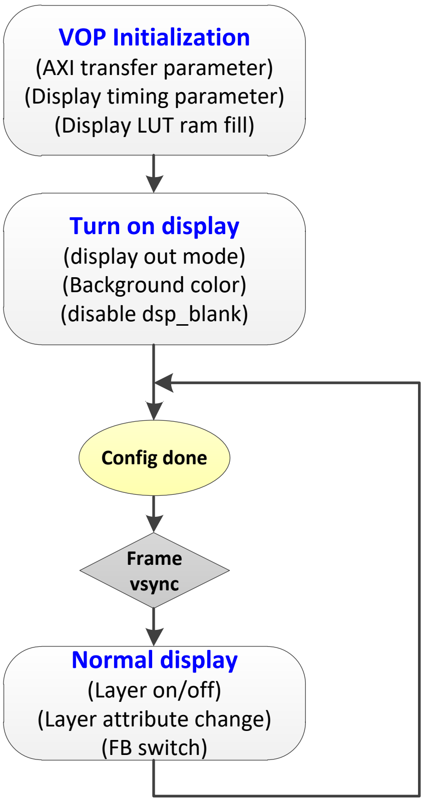


Fig. 7-15 VOP RGB Mode Programming Flow

1.VOP initialization

VOP initialization should be done before turning display on.
First, AXI bus parameter (VOP_SYS_CTRL1) should be set for DMA transfer.
Second, display panel/interface timing should be set for display output. The registers are: VOP_DSP_HTOTAL_HS_END/VOP_DSP_HACT_ST_END/VOP_DSP_VTOTAL_HS_END/VOP_DSP_VACT_ST_END/VOP_DSP_VS_ST_END_F1/VOP_DSP_VACT_ST_END_F1.

2.Background display

Before normal display, the background display could be turn on.
First, set display output mode (VOP_DSP_CTRL0/1) according to display device.
Second, disable dsp_blank mode, which would not be enable until frame synchronization.
Finally, writing '1' to "VOP_REG_CFG_DONE" register then all the frame-sync registers will be enable at the beginning of next frame.

3.Normal display

In normal display, all the display layers' attribute could be different according display scenario. So there is a programming loop in this mode.
First, configure all the display layers' attribute registers for the change of image format, location, size, scaling factor, alpha and overlay and so on. Those register would not be enable until frame synchronization.
Finally, write 1 to "VOP_REG_CFG_DONE" register then all the frame-sync registers will be enable at the beginning of next frame.

7.7.7 Immediately control register

There are two type registers in VOP, one is effective immediately, the other is effective by frame sync. Effective immediately registers list as follows, other registers are all effective by frame sync.

Table 7-5 effective immediately register table

register address	description
0x0008	background
0x0018	All display and control signal registers
0xe0~0xf4	Frc configuration bits

7.7.8 Output Polarity Control

There are two channel outputs(RGB MCU , some may not support because SOC do not have the related interface), every channel has its own xxx_dclk_en, xxx_dclk_pol, xxx_hsync_pol,

xxx_vsync_pol, xxx_den_pol.
The xxx_dclk_en should be set to 1 when output select the xxx channel, and the other channel's xxx_dclk_en should be set to 0 to gate the output clk and data.
When using RGB panel, the dclk should be tied to "0" or "1" in some scenarios.
In this case, you should enable sw_io_pad_clk_sel, to tie dclk to "0". If enable rgb_dclk_pol at the same time, the dclk will tie to "1".

7.7.9 Some special control

- The blanking value of VSYNC could be configured through reg BLANKING_VALUE;
- The current scan line number could be read through reg_addr 0x190;
- There is a FLAG_REG which is readable and writable. This FLAG_REG does not for config function , our software staff may use it in the future. After writing a meaningful 32bit value to FLAG_REG, we can read it before frame valid through reading 0x1fc value, and can get the same value after frame valid reading 0x1f8 value;

Chapter 8 Voice Activity Detect (VAD)

8.1 Overview

Voice Activity Detect(VAD) is used to detect the amplitude of voice which is received by Analog Mic, I2S Digital Mic or PDM digital Mic when SoC is in low power mode. If the amplitude of voice is over threshold, the VAD will assert interrupt to wake up SoC, then SoC will exit low power mode.

VAD supports the following features:

- Support AHB bus interface
- Support read voice data from I2S_8CH_0~3/PDM
  - Support to configure the voice source address
  - Support to configure increment or fixed for the direction of voice data address
  - Support 5 group of DMA request and acknowledge
  - Support transfer 1~8 burst per DMA request
  - Support read 1~8 Mic voice data, and only support single Mic voice detection, user can select any Mic voice data to detect the amplitude of voice
  - Support 16/24 bits voice data
- Support voice amplitude detection
  - Support an Amplifier for the voice data
  - Support a IIR high pass filter for the voice frequency band, and the filter coefficient can be configured
  - Support a voice detect threshold that take the ambient noise to account
- Support configure Audio Codec register after voice detection event
  - Support to set enable or disable of the configuration action
  - Support to configure 1~32 registers, and the width is 8
- Support Multi-Mic array data storing
  - Buffer memory is shared with Internal SRAM
  - The start and end address of storing can be configured
  - When current storing address is up to end address, it will loop to start address and overlap previous data, it will also assert a flag
  - Support 3 data storing mode: mode 0 start storing data after the voice detect event, mode 1 start storing after VAD is enabled and mode 2 do not storing data
- Support a level combined interrupt
  - Support voice detect interrupt
  - Support time out interrupt
  - Support transfer error interrupt

8.2 Block Diagram

VAD comprises with:

- ahb_master: AHB Master Interface
- ahb_sram_if: AHB Slave Interface
- vad_reg_bank : Register bank
- dmac_engine: DMA control engine
- vad_det : Voice detection

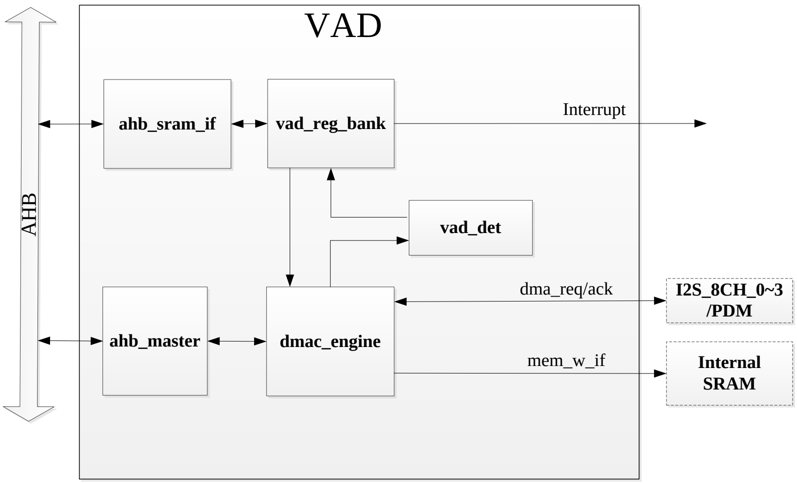


Fig. 8-1 VAD Block Diagram

8.3 Function Description

8.3.1 DMAC_ENGINE

dmac_engine is used to read voice data from one of I2S_8CH_0~3 or PDM, and it can store all channels data to Internal SRAM. The voice data can be 16 or 24 bits:

- When it's 16 bits, it must be half word transfer mode that low 16 bits in a word for left channel and high 16 bits in a word for right channel.
- When it's 24bits, it must be word transfer mode that only 24 bits data is valid in a word, and it support left or right justified

dmac_engine also select and send one channel data to vad_det for voice detection. vad_det only support 16 bits data to detect the amplitude of voice, so when the voice data is 24 bits, user can use the high or low 16 bits in 24 bits.

- When use high bits, the data value will be divided by 256.
- When use low bits, the data value will be saturation to 16 bits.

8.3.2 VAD_DET

vad_det is used to detect the amplitude of voice.

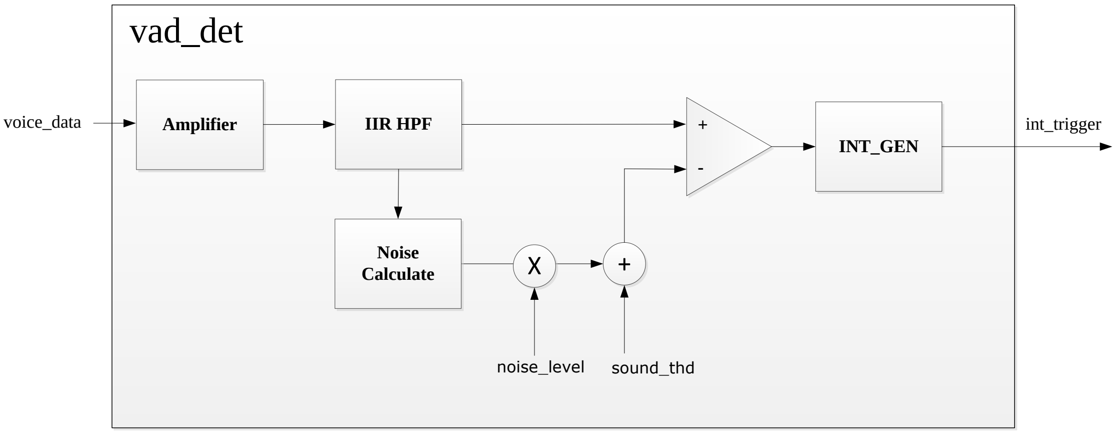


Fig. 8-2 vad_det Block Diagram

Amplifier

voice_amplitude_amplified=gain*voice_amplitude_original/32.

IIR HPF

There is a high pass filter for the human voice frequency band, the filter is a two order direct I type IIR. The following formula describes:

y(n)=-a1*y(n-1)-a2*y(n-2)+b0*x(n)+b1*x(n-1)+b2*x(n-2)

The coefficient a1, a2, b0, b1 and b2 are all quantified by multiplying 16384 and represented as 16 bits, the result is follow registers that can be configured: iir_anum_0,

iir_anum_1, iir_anum_2, iir_aden_1 and iir_aden_2.
The output of HPF need some time to achieve convergence after VAD is enabled.

Noise Calculate

VAD support a voice detection threshold that take the ambient noise to account:

- VAD calculate the average amplitude of voice data within noise_sample_num samples, the result is regard as the noise value of one frame.
- VAD find the minimum noise value within noise_frm_num frames, the result is regard as the noise_min(minimum noise value). User can configure min_noise_find_mode to change the mode to find the minimum noise.
- The noise_min will be smooth updated to noise_abs(current noise value), the formula is as follow: noise_abs= (noise_abs * noise_alpha + noise_min * (256-noise_alpha)/256. noise_abs will be updated once every frame. noise_abs also can be configured directly, and it is not clear until VAD is reset.

Voice Detect Threshold

The final threshold is sound_thd + noise_abs * noise_level.

INT_GEN

When equal or greater than vad_con_thd continuous sample over the threshold, the voice detection interrupt will be asserted.

8.4 Register Description

8.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
VAD_CONTROL	0x0000	W	0x03000000	Control register
VAD_VS_ADDR	0x0004	W	0x00000000	Voice source address register
VAD_ACODEC_BASE_ADDR	0x0008	W	0x00000000	Audio Codec base address register
VAD_OD_ADDR0	0x000c	W	0x00000000	Offset of Audio Codec address register0
VAD_OD_ADDR1	0x0010	W	0x00000000	Offset of Audio Codec address register1
VAD_OD_ADDR2	0x0014	W	0x00000000	Offset of Audio Codec address register2
VAD_OD_ADDR3	0x0018	W	0x00000000	Offset of Audio Codec address register3
VAD_OD_ADDR4	0x001c	W	0x00000000	Offset of Audio Codec address register4
VAD_OD_ADDR5	0x0020	W	0x00000000	Offset of Audio Codec address register5
VAD_OD_ADDR6	0x0024	W	0x00000000	Offset of Audio Codec address register6
VAD_OD_ADDR7	0x0028	W	0x00000000	Offset of Audio Codec address register7
VAD_D_DATA0	0x002c	W	0x00000000	Audio Codec data register0
VAD_D_DATA1	0x0030	W	0x00000000	Audio Codec data register1
VAD_D_DATA2	0x0034	W	0x00000000	Audio Codec data register2
VAD_D_DATA3	0x0038	W	0x00000000	Audio Codec data register3
VAD_D_DATA4	0x003c	W	0x00000000	Audio Codec data register4
VAD_D_DATA5	0x0040	W	0x00000000	Audio Codec data register5
VAD_D_DATA6	0x0044	W	0x00000000	Audio Codec data register6
VAD_D_DATA7	0x0048	W	0x00000000	Audio Codec data register7
VAD_TIMEOUT	0x004c	W	0x00000000	Timeout register
VAD_RAM_START_ADDR	0x0050	W	0x00000000	RAM start address register
VAD_RAM_END_ADDR	0x0054	W	0x00000000	RAM end address register
VAD_RAM_CUR_ADDR	0x0058	W	0x00000000	RAM current address register
VAD_DET_CON0	0x005c	W	0x00024020	Detect control register0
VAD_DET_CON1	0x0060	W	0x00ff0064	Detect control register1
VAD_DET_CON2	0x0064	W	0x3bf5e663	Detect control register2
VAD_DET_CON3	0x0068	W	0x3bf58817	Detect control register3
VAD_DET_CON4	0x006c	W	0x382b8858	Detect control register4
VAD_DET_CON5	0x0070	W	0x00000000	Detect control register5
VAD_INT	0x0074	W	0x00000000	VAD interrupt register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

8.4.2 Detail Register Description

VAD_CONTROL

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:29	RW	0x0	vad_det_channel Index of the channal for voice detect, from channel 0 to channel 7
28	RW	0x0	voice_24bit_sat The mode of voice 24bit data change to 16bit 1'b0: Get the high 16bit data(divided by 256) 1'b1: Saturation from 24bit to 16bit
27	RW	0x0	voice_24bit_align_mode Align mode of channel 24bit width 1'b0: 8~31bits is valid 1'b1: 0~23bits is valid
26	RW	0x0	voice_channel_bitwidth 1'b0: 16bits 1'b1: 24bits
25:23	RW	0x6	voice_channel_num Voice channel number, the value N means N+1 channel
22	RW	0x0	config_after_det_en Configure the Audio Codec after voice detection 1'b0: Disable 1'b1: Enable
21:20	RW	0x0	vad_mode 2'h0: Begin to store the data after voice detect 2'h1: Begin to store the data after VAD is enable 2'h2: Don't store the data 2'h3: Reserved
19:15	RW	0x00	acodec_cfg_reg_num Audio Codec configuration register number, the value N means N+1 data
14	RW	0x0	source_fixaddr_en Direction of source address 1'b0: Increment 1'b1: Fixed
13:10	RW	0x0	incr_length INCR burst length, 0~15 is valid. It is valid when source_burst is set to 3'h1.
9:7	RW	0x0	source_burst_num Source burst number per dma_req, the value N means N+1 burst

Bit	Attr	Reset Value	Description
6:4	RW	0x0	source_burst 3'h0: SINGLE 3'h1: INCR 3'h3: INCR4 3'h5: INCR8 3'h7: INCR16 Others: Reserved
3:1	RW	0x0	source_select Voice source select 3'h0: I2S0_8CH 3'h1: I2S1_8CH 3'h2: I2S2_8CH 3'h3: I2S3_8CH 3'h4: PDM Others: Reserved
0	RW	0x0	vad_en VAD enable 1'b0: Disable 1'b1: Enable

VAD_VS_ADDR

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	vs_addr Voice source address

VAD_ACODEC_BASE_ADDR

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	acodec_base_addr Audio Codec base address for configuration

VAD_OD_ADDR0

Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	offset_acodec_addr_3 Offset of Audio Codec address for the number 3 configuration
23:16	RW	0x00	offset_acodec_addr_2 Offset of Audio Codec address for the number 2 configuration
15:8	RW	0x00	offset_acodec_addr_1 Offset of Audio Codec address for the number 1 configuration
7:0	RW	0x00	offset_acodec_addr_0 Offset of Audio Codec address for the number 0 configuration

VAD_OD_ADDR1

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	offset_acodec_addr_7 Offset of Audio Codec address for the number 7 configuration
23:16	RW	0x00	offset_acodec_addr_6 Offset of Audio Codec address for the number 6 configuration
15:8	RW	0x00	offset_acodec_addr_5 Offset of Audio Codec address for the number 5 configuration
7:0	RW	0x00	offset_acodec_addr_4 Offset of Audio Codec address for the number 4 configuration

VAD_OD_ADDR2

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	offset_acodec_addr_11 Offset of Audio Codec address for the number 11 configuration
23:16	RW	0x00	offset_acodec_addr_10 Offset of Audio Codec address for the number 10 configuration
15:8	RW	0x00	offset_acodec_addr_9 Offset of Audio Codec address for the number 9 configuration
7:0	RW	0x00	offset_acodec_addr_8 Offset of Audio Codec address for the number 8 configuration

VAD_OD_ADDR3

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	offset_acodec_addr_15 Offset of Audio Codec address for the number 15 configuration
23:16	RW	0x00	offset_acodec_addr_14 Offset of Audio Codec address for the number 14 configuration
15:8	RW	0x00	offset_acodec_addr_13 Offset of Audio Codec address for the number 13 configuration
7:0	RW	0x00	offset_acodec_addr_12 Offset of Audio Codec address for the number 12 configuration

VAD_OD_ADDR4

Address: Operational Base + offset (0x001c)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	offset_acodec_addr_19 Offset of Audio Codec address for the number 19 configuration
23:16	RW	0x00	offset_acodec_addr_18 Offset of Audio Codec address for the number 18 configuration

Bit	Attr	Reset Value	Description
15:8	RW	0x00	offset_acodec_addr_17 Offset of Audio Codec address for the number 17 configuration
7:0	RW	0x00	offset_acodec_addr_16 Offset of Audio Codec address for the number 16 configuration

VAD_OD_ADDR5

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	offset_acodec_addr_23 Offset of Audio Codec address for the number 23 configuration
23:16	RW	0x00	offset_acodec_addr_22 Offset of Audio Codec address for the number 22 configuration
15:8	RW	0x00	offset_acodec_addr_21 Offset of Audio Codec address for the number 21 configuration
7:0	RW	0x00	offset_acodec_addr_20 Offset of Audio Codec address for the number 20 configuration

VAD_OD_ADDR6

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	offset_acodec_addr_27 Offset of Audio Codec address for the number 27 configuration
23:16	RW	0x00	offset_acodec_addr_26 Offset of Audio Codec address for the number 26 configuration
15:8	RW	0x00	offset_acodec_addr_25 Offset of Audio Codec address for the number 25 configuration
7:0	RW	0x00	offset_acodec_addr_24 Offset of Audio Codec address for the number 24 configuration

VAD_OD_ADDR7

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	offset_acodec_addr_31 Offset of Audio Codec address for the number 31 configuration
23:16	RW	0x00	offset_acodec_addr_30 Offset of Audio Codec address for the number 30 configuration
15:8	RW	0x00	offset_acodec_addr_29 Offset of Audio Codec address for the number 29 configuration
7:0	RW	0x00	offset_acodec_addr_28 Offset of Audio Codec address for the number 28 configuration

VAD_D_DATA0

Address: Operational Base + offset (0x002c)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	acodec_data_3 Audio Codec data for the number 3 configuration
23:16	RW	0x00	acodec_data_2 Audio Codec data for the number 2 configuration
15:8	RW	0x00	acodec_data_1 Audio Codec data for the number 1 configuration
7:0	RW	0x00	acodec_data_0 Audio Codec data for the number 0 configuration

VAD_D_DATA1

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	acodec_data_7 Audio Codec data for the number 7 configuration
23:16	RW	0x00	acodec_data_6 Audio Codec data for the number 6 configuration
15:8	RW	0x00	acodec_data_5 Audio Codec data for the number 5 configuration
7:0	RW	0x00	acodec_data_4 Audio Codec data for the number 4 configuration

VAD_D_DATA2

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	acodec_data_11 Audio Codec data for the number 11 configuration
23:16	RW	0x00	acodec_data_10 Audio Codec data for the number 10 configuration
15:8	RW	0x00	acodec_data_9 Audio Codec data for the number 9 configuration
7:0	RW	0x00	acodec_data_8 Audio Codec data for the number 8 configuration

VAD_D_DATA3

Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	acodec_data_15 Audio Codec data for the number 15 configuration
23:16	RW	0x00	acodec_data_14 Audio Codec data for the number 14 configuration
15:8	RW	0x00	acodec_data_13 Audio Codec data for the number 13 configuration
7:0	RW	0x00	acodec_data_12 Audio Codec data for the number 12configuration

VAD_D_DATA4

Address: Operational Base + offset (0x003c)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	acodec_data_19 Audio Codec data for the number 19 configuration
23:16	RW	0x00	acodec_data_18 Audio Codec data for the number 18 configuration
15:8	RW	0x00	acodec_data_17 Audio Codec data for the number 17 configuration
7:0	RW	0x00	acodec_data_16 Audio Codec data for the number 16 configuration

VAD_D_DATA5

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	acodec_data_23 Audio Codec data for the number 23 configuration
23:16	RW	0x00	acodec_data_22 Audio Codec data for the number 22 configuration
15:8	RW	0x00	acodec_data_21 Audio Codec data for the number 21 configuration
7:0	RW	0x00	acodec_data_20 Audio Codec data for the number 20 configuration

VAD_D_DATA6

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	acodec_data_27 Audio Codec data for the number 27 configuration
23:16	RW	0x00	acodec_data_26 Audio Codec data for the number 26 configuration
15:8	RW	0x00	acodec_data_25 Audio Codec data for the number 25 configuration
7:0	RW	0x00	acodec_data_24 Audio Codec data for the number 24 configuration

VAD_D_DATA7

Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	acodec_data_31 Audio Codec data for the number 31 configuration
23:16	RW	0x00	acodec_data_30 Audio Codec data for the number 30 configuration

Bit	Attr	Reset Value	Description
15:8	RW	0x00	acodec_data_29 Audio Codec data for the number 29 configuration
7:0	RW	0x00	acodec_data_28 Audio Codec data for the number 28 configuration

VAD_TIMEOUT

Address: Operational Base + offset (0x004c)

Bit	Attr	Reset Value	Description
31	RW	0x0	work_timeout_en Work timeout enable 1'b0: Disable 1'b1: Enable
30	RW	0x0	idle_timeout_en Idle timeout enable 1'b0: Disable 1'b1: Enable
29:20	RW	0x000	work_timeout_thd work timeout threshold, the unit is one cycle of hclk
19:0	RW	0x00000	idle_timeout_thd Idle timeout threshold, the unit is one cycle of hclk

VAD_RAM_START_ADDR

Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	ram_begin_addr RAM start address to store voice data, the address must be double word alignment

VAD_RAM_END_ADDR

Address: Operational Base + offset (0x0054)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	ram_end_addr RAM end address to store voice data, the address must be double word alignment

VAD_RAM_CUR_ADDR

Address: Operational Base + offset (0x0058)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	ram_cur_addr RAM current address to store voice data, The last valid double word data is at address ram_cur_addr-0x8. When the ram_loop_flag is valid, the valid voice data will be ram_cur_addr ~ ram_end_addr ~ loop to ram_begin_addr ~ ram_cur_addr-0x8. When the ramp_loop_flag is not valid, the valid voice data will be ram_begin_addr ~ ram_cur_addr-0x8

VAD_DET_CON0

Address: Operational Base + offset (0x005c)

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:16	RW	0x02	vad_con_thd When continuous sample number(>=vad_con_thd) exceed threshold, then assert the vad_det interrupt, the value N means N+1. When this value is higher, the voice detect condition is more strict
15	RO	0x0	reserved
14:12	RW	0x4	noise_level Noise level, valid value is 0x1~0x6 when this value is higher, the voice detect condition is more strict
11:10	RO	0x0	reserved
9:0	RW	0x020	gain The gain control of voice data amplifier, the value of gain is signed and is valid from -512 to 511. voice_amplitude_amplified=gain*voice_amplitude_original/32.

VAD_DET_CON1

Address: Operational Base + offset (0x0060)

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30	RW	0x0	min_noise_find_mode Minimal noise value find mode 1'b0: Always find the value at the range of noise_frm_num 1'b1: When receive N frame: if N is less than noise_frm_num, find the value at the range of N; if N is more than noise_frm_num, find the value at the range of noise_frm_num
29	RW	0x0	clean_noise_at_begin 1'b0: The noise will be clean only at the begin of the first time vad is enable after reset 1'b1: The noise will be clean every time at the begin of vad is enable
28	RW	0x0	force_noise_clk_en Force noise calculate clk enable 1'b0: The clock will be auto gating for low power 1'b1: The clock will be always enable
27:26	RO	0x0	reserved
25:16	RW	0x0ff	noise_sample_num The number of sample in one frame to calculate the noise, the value N means N+1 sample. When this value is higher, the voice detect condition is more strict
15:0	RW	0x0064	sound_thd Initial sound threshold when this value is higher, the voice detect condition is more strict

VAD_DET_CON2
Address: Operational Base + offset (0x0064)

Bit	Attr	Reset Value	Description
31:16	RW	0x3bf5	iir_anum_0 IIR numerator coefficient b0
15:8	RW	0xe6	noise_alpha The update smooth speed of noise When this value is lower, the voice detect condition is more strict
7	RO	0x0	reserved
6:0	RW	0x63	noise_frm_num The number of frame to calculate the noise, the value N means N+1 frame. When this value is lower, the voice detect condition is more strict

VAD_DET_CON3
Address: Operational Base + offset (0x0068)

Bit	Attr	Reset Value	Description
31:16	RW	0x3bf5	iir_anum_2 IIR numerator coefficient b2
15:0	RW	0x8817	iir_anum_1 IIR numerator coefficient b1

VAD_DET_CON4

Address: Operational Base + offset (0x006c)

Bit	Attr	Reset Value	Description
31:16	RW	0x382b	iir_aden_2 IIR demoninator coefficient a2
15:0	RW	0x8858	iir_aden_1 IIR demoninator coefficient a1

VAD_DET_CON5

Address: Operational Base + offset (0x0070)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	iir_result Voice real time data after IIR filter
15:0	RW	0x0000	noise_abs Noise abs value

VAD_INT

Address: Operational Base + offset (0x0074)

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9	RW	0x0	vad_idle vad idle flag 1'b0: Not idle 1'b1: Idle
8	RO	0x0	ramp_loop_flag RAM adress loop flag 1'b0: not loop 1'b1: loop
7	W1C	0x0	work_timeout_int Work timeout interrupt 1'b0: interrupt not generated 1'b1: interrupt generated
6	W1C	0x0	idle_timeout_int Idle timeout interrupt 1'b0: interrupt not generated 1'b1: interrupt generated

Bit	Attr	Reset Value	Description
5	RW	0x0	error_int Error interrupt 1'b0: interrupt not generated 1'b1: interrupt generated
4	W1C	0x0	vad_det_int VAD detect interrupt 1'b0: interrupt not generated 1'b1: interrupt generated
3	RW	0x0	work_timeout_int_en Wrok timeout interrupt enable 1'b0: Disable 1'b1: Enable
2	RW	0x0	idle_timeout_int_en Idle timeout interrupt enable 1'b0: Disable 1'b1: Enable
1	RW	0x0	error_int_en Error interrupt enable 1'b0: Disable 1'b1: Enable
0	RW	0x0	vad_det_int_en VAD detect interrupt enable 1'b0: Disable 1'b1: Enable

8.5 Application Notes

8.5.1 VAD usage flow

VAD usage flow is as following figure.

- Step 4, step5, step9 and step14 are optional, user should consider the power consumption and keyword detection accuracy for these steps.
- For step4, just power down the unused ADC in Audio Codec or unused external digital mic. The I2S_8CH or PDM is still work at normal channel, the voice data of unused channels are zero.
- Step9 power up ADC in Audio Codec or external digital mic which is power down at step4. It’s controlled by VAD or hardware after voice detection event, CPU needn’t to configure it.

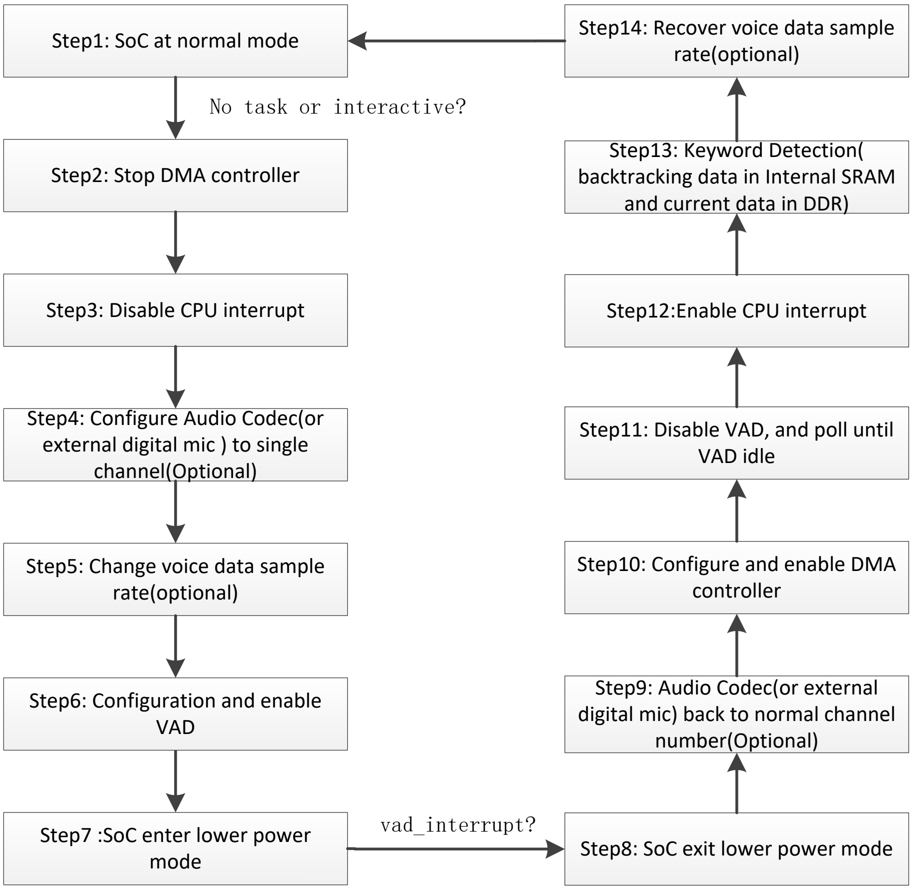

Step1: SoC at normal mode
No task or interactive?
Step2: Stop DMA controller
Step3: Disable CPU interrupt
Step4: Configure Audio Codec(or external digital mic) to single channel(Optional)
Step5: Change voice data sample rate(optional)
Step6: Configuration and enable VAD
Step7 :SoC enter lower power mode
vad_interrupt?
Step8: SoC exit lower power mode
Step9: Audio Codec(or external digital mic) back to normal channel number(Optional)
Step10: Configure and enable DMA controller
Step11: Disable VAD, and poll until VAD idle
Step12:Enable CPU interrupt
Step13: Keyword Detection(backtracking data in Internal SRAM and current data in DDR)
Step14: Recover voice data sample rate(optional)

Fig. 8-3 VAD usage flow

8.5.2 VAD configuration usage flow

1. Set CRU_CLKGATE_CON14[10]=0x0 to enable the VAD clock.
2. Set VAD_VS_ADDR.vs_addr=I2S0_8CH base address + I2S_8CH_INCR_RXDR.
3. Set VAD_ACODEC_BASE_ADDR, VAD_OD_ADDR0~7 and VAD_D_DATA0~7.

This step is used to configure some registers of Audio Codec refer to Audio Codec Application note about enable ADC.

In detail, user need to set acodec_base_addr, offset_acodec_addr_0~31 and acodec_data_0~31(if the number that need to be configured is N, user only need to set offset_acodec_addr_0 ~ offset_acodec_addr_N-1 and acodec_data_0 ~ acodec_data_N-1). acodec_data_0 will be configured to the register with offset address offset_acodec_addr_0, and the others are the same. The registers will be configured to Audio Codec from low to high order.

Because the unit of offset address is word, so the actually register address will be Audio Codec base address + 4*offset_acodec_addr_x. Following is an example.

Set VAD_ACODEC_BASE_ADDR.acodec_base_addr=Audio Codec base address + 0x340(user can add an offset to base address).

Set VAD_OD_ADDR0=0x37271707:
offset_acodec_addr_0=0x7
offset_acodec_addr_1=0x17
offset_acodec_addr_2=0x27
offset_acodec_addr_3=0x37

Set VAD_D_DATA0=0x50505050:
acodec_data_0=0x50
acodec_data_1=0x50
acodec_data_2=0x50
acodec_data_3=0x50

As a result after configuration:

register[ACODEC base address + 0x340 + 0x7*4]=0x50
register[ACODEC base address + 0x340 + 0x17*4]=0x50
register[ACODEC base address + 0x340 + 0x27*4]=0x50
register[ACODEC base address + 0x340 + 0x37*4]=0x50
User can add the configuration follow this principle.

4. Set VAD_RAM_BEGIN_ADDR.ram_begin_addr and VAD_RAM_END_ADDR.ram_end_addr, the address should match with the Internal SRAM sharing scheme.

5. Adjust the sensitivity of voice activity detect by setting follow registers:
VAD_DET_CON0.noise_level
VAD_DET_CON0.vad_con_thd
VAD_DET_CON1.noise_sample_num
VAD_DET_CON2.noise_frm_num
VAD_DET_CON2.noise_alpha

6. Set the iir_anum_0~3 and iir_aden_1~2 to adjust the IIR HPF coefficient.
For 48Khz sample rate：
iir_anum_0: 0x382d
iir_anum_1: 0x8fa5
iir_anum_2: 0x382d
iir_aden_1: 0x909b
iir_aden_2: 0x3150
For 16Khz sample rate（default）：
iir_anum_0: 0x3bf5
iir_anum_1: 0x8817
iir_anum_2: 0x3bf5
iir_aden_1: 0x8858
iir_aden_2: 0x382b
For 8Khz sample rate：
iir_anum_0: 0x3e9f
iir_anum_1: 0x82c2
iir_anum_2:0x3e9f
iir_aden_1: 0x82c9
iir_aden_2: 0x3d46

7. Set DET_CON5.noise_abs to ambient noise which is calculated by software(optional).

8. Set VAD_INT.vad_det_int_en=0x1 to enable the interrupt.

9. Set VAD_DET_CON1.sound_thd=0xffff to disable detection at the beginning after VAD is enabled.

10. Set VAD_CONTROL register:
Set source_select=0x0, select I2S0_8CH
Set source_burst=0x3, select INCR4 burst type
Set source_burst_num=0x0, select 1 burst transfer per DMA request
Set acodec_cfg_reg_num=0xN, the register number that need configure to Audio Codec is N
Set vad_mode=0x0, select Mode 0
Set config_after_det_en=0x1, enable Audio Codec configuration
Set voice_channel_num=0x7, all voice channel number is 8
Set voice_channel_bitwidth=0x0, voice data width is 16 bits
Set vad_det_channel=0x0, use channel 0 to voice activity detect
Set vad_en=0x1, enable VAD

11. Delay about 4ms, Set VAD_DET_CON1.sound_thd to appropriate value.

12. After above setting, VAD will start to work and system can enter low power mode.

Note1: If user don’t configure Audio Codec(or external digital mic) to single channel, the step 3 can be ignored.

8.5.3 Timeout configuration usage flow

1. Set VAD_TIMEOUT.idle_timeout_thd=0xfffff, set VAD_TIMEOUT.idle_timeout_en=0x1, set VAD_INT.idle_timeout_int=0x1. After above setting, a counter is increase at AHB clock when dmac_engine is idle, the counter will be clear to 0 once dmac_engine start to read voice data. An interrupt will be asserted when the counter up to idle_timeout_thd. This idle

timeout is used for I2S/PDM work fail(Don’t assert DMA request for a long time).
2. Set VAD_TIMEOUT.work_timeout_thd=0x3ff, set VAD_TIMEOUT.work_timeout_en=0x1, set VAD_INT.work_timeout_int=0x1. After above setting, a counter is increase at AHB clock when dmac_engine is busy, the counter will be clear to 0 once dmac_engine is idle. An interrupt will be asserted when the counter up to work_timeout_thd. This work timeout is used for bus transmission congestion(a burst transferring is not completed for a long time).

Chapter 9 Crypto

9.1 Overview

Crypto is a hardware accelerator for encrypting or decrypting. It supports the most commonly used algorithm: DES/3DES, AES, SHA1, SHA256, MD5 and PKA.
The Crypto supports following features:

- Support Link List Item (LLI) DMA transfer;
- Support SHA-1, SHA-256/224, SHA-512/384, MD5 with hardware padding;
- Support HMAC ofSHA-1, SHA-256, SHA-512, MD5 with hardware padding;
- Support AES-128, AES-192, AES-256 encrypt & decrypt cipher;
- Support DES & TDES cipher;
- Support AES ECB/CBC/OFB/CFB/CTR/CTS/XTS/CCM/GCM/CBC-MAC/CMAC mode;
- Support DES/TDES ECB/CBC/OFB/CFB mode;
- Support up to 4096bits PKA mathematical operations for RSA/ECC;
- Support up to 8-channels configuration;
- Support 256-bits OTP device root key hardwire calculation;
- Support Up to 256 bits TRNG Output.

9.2 Block Diagram

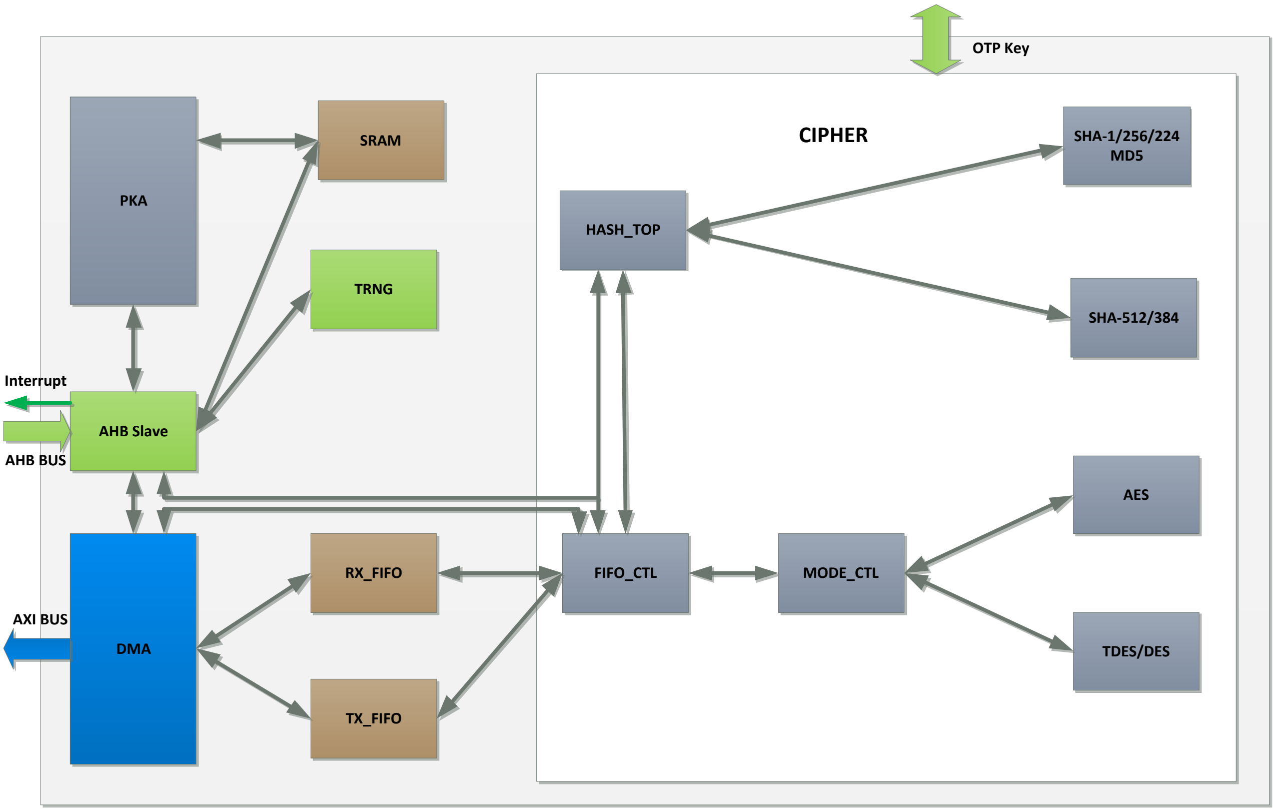


Fig. 9-1Crypto Architecture

Crypto contains several modules : AHB_Slave, DMA, CIPHER, PKA, TRNG.

AHB_Slave

AHB_Slave is used to configure registers. This module is in HCLK domain.

DMA

DMA is used to transfer data from external memory to RX_FIFO, or from TX_FIFO to external memory. DMA uses 64-bits AXI3 protocol with max burst length to 16.   LLI transfer is also supported for performance and convenience consideration. This module is in ACLK domain.

CIPHER

CIPHER contains AES, DES/TDES and HASH engines. And it also supports various mode operations. The source data is either from RX_FIFO , or from other engine output. The result data is sending either to TX_FIFO, or Registers in module AHB_Slave. This module is in CLK_CORE domain.

PKA

PKA is used to accelerate mathematical operations for big numbers. It supports - Modular arithmetic (addition, subtraction, multiplication and division), Regular arithmetic (addition, subtraction, multiplication and division), Modular inversion, Modular exponentiation, Logical operations (AND, OR, XOR, SHIFT). PKA has a SRAM which is used to store source, result and intermediate data for PKA operations. The software driver could use PKA operations to implement complicate calculation, such as RSA, ECC etc. It could support up to 4096 bits RSA modular exponentiation calculation. This module is in CLK_PKA domain.

TRNG

TRNG is used to collects random bits from the ring oscillator, up to 256 random bits per time. This module is in HCLK domain.

9.3 Register description

9.3.1 Internal Address Mapping

Slave address can be divided into different length for different usage, which is shown as follows.

9.3.2 Registers Summary

Name	Offset	Size	Reset Value	Description
CRYPTO_CLK_CTL	0x0000	W	0x00000001	Clock Control Register
CRYPTO_RST_CTL	0x0004	W	0x00000000	Reset Control Register
CRYPTO_DMA_INT_EN	0x0008	W	0x00000000	DMA Interrupt Enable Register
CRYPTO_DMA_INT_ST	0x000c	W	0x00000000	DMA Interrupt Status Register
CRYPTO_DMA_CTL	0x0010	W	0x00000000	DMA Control Register
CRYPTO_DMA_LLI_ADDR	0x0014	W	0x00000000	DMA LIST Start Address Register
CRYPTO_DMA_ST	0x0018	W	0x00000000	DMA Status Register
CRYPTO_DMA_STATE	0x001c	W	0x00000000	DMA State Register
CRYPTO_DMA_LLI_RADDR	0x0020	W	0x00000000	DMA LLI Read Address Register
CRYPTO_DMA_SRC_RADDR	0x0024	W	0x00000000	DMA Source Data Read Address Register
CRYPTO_DMA_DST_WADDR	0x0028	W	0x00000000	DMA Destination Data Read Address Register
CRYPTO_DMA_ITEM_ID	0x002c	W	0x00000000	DMA Descriptor ID Register
CRYPTO_FIFO_CTL	0x0040	W	0x00000003	FIFO Control Register
CRYPTO_BC_CTL	0x0044	W	0x00000000	Block Cipher Control Register
CRYPTO_HASH_CTL	0x0048	W	0x00000004	Hash Control Register
CRYPTO_CIPHER_ST	0x004c	W	0x00000000	Cipher Status Register
CRYPTO_CIPHER_STATE	0x0050	W	0x00000400	Cipher Current State Register
CRYPTO_CHn_IV_0	0x0100	W	0x00000000	Channel n IV Register 0.Channel n range from 0 to 7. CHn_IV_0 address = 0x0100 + 0x10 * n. For CTR Mode, IV stands for counter. For XTS Mode, IV stands for tweak.

Name	Offset	Size	Reset Value	Description
CRYPTO_CHn_IV_1	0x0104	W	0x00000000	Channel n range from 0 to 7. CHn_IV_1 address = 0x0104 + 0x10 * n. For CTR Mode, IV stands for counter. For XTS Mode, IV stands for tweak.
CRYPTO_CHn_IV_2	0x0108	W	0x00000000	Channel n range from 0 to 7. CHn_IV_2 address = 0x0108 + 0x10 * n. For CTR Mode, IV stands for counter. For XTS Mode, IV stands for tweak.
CRYPTO_CHn_IV_3	0x010c	W	0x00000000	Channel n range from 0 to 7. CHn_IV_3 address = 0x010c + 0x10 * n. or CTR Mode, IV stands for counter. For XTS Mode, IV stands for tweak.
CRYPTO_CHn_KEY_0	0x0180	W	0x00000000	Channel n range from 0 to 7. CHn_KEY_0 address = 0x0180 + 0x10 * n
CRYPTO_CHn_KEY_1	0x0184	W	0x00000000	Channel n range from 0 to 7. CHn_KEY_1 address = 0x0184 + 0x10 * n.
CRYPTO_CHn_KEY_2	0x0188	W	0x00000000	Channel n range from 0 to 7. CHn_KEY_2 address = 0x0188 + 0x10 * n.
CRYPTO_CHn_KEY_3	0x018c	W	0x00000000	Channel n range from 0 to 7.CHn_KEY_3 address = 0x018c + 0x10 * n.
CRYPTO_CHn_PKEY_0	0x0200	W	0x00000000	Channel n range from 0 to 7. CHn_PKEY_0 address = 0x0200 + 0x10 * n.
CRYPTO_CHn_PKEY_1	0x0204	W	0x00000000	Channel n range from 0 to 7. CHn_PKEY_1 address = 0x0204 + 0x10 * n.
CRYPTO_CHn_PKEY_2	0x0208	W	0x00000000	Channel n range from 0 to 7. CHn_PKEY_2 address = 0x208 + 0x10 * n.
CRYPTO_CHn_PKEY_3	0x020c	W	0x00000000	Channel n range from 0 to 7. CHn_PKEY_3 address = 0x020c + 0x10 * n.
CRYPTO_CHn_PC_LEN_0	0x0280	W	0x00000000	Channel n range from 0 to 7. CHn_PC_LEN_0 address = 0x0280 + 0x8 * n. Length in byte unit.

Name	Offset	Size	Reset Value	Description
CRYPTO_CHn_PC_LEN_1	0x0284	W	0x00000000	Channel n range from 0 to 7.CHn_PC_LEN_1 address = 0x0284 + 0x8 * n. Length in byte unit.
CRYPTO_CHn_ADA_LEN_0	0x02c0	W	0x00000000	Channel n range from 0 to 7. CHn_ADA_LEN_0 address = 0x02c0 + 0x8 * n. Length in byte unit.
CRYPTO_CHn_ADA_LEN_1	0x02c4	W	0x00000000	Channel n range from 0 to 7. CHn_ADA_LEN_1 address = 0x02c4 + 0x8 * n. Length in byte unit.
CRYPTO_CHn_IV_LEN_0	0x0300	W	0x00000000	Channel n range from 0 to 7. CHn_IV_LEN_0 address = 0x0300 + 0x4 * n. Length in byte unit. Up to 16 byte IV for GCM.
CRYPTO_CHn_TAG_0	0x0320	W	0x00000000	Channel n range from 0 to 7. CHn_TAG_0 address = 0x0320 + 0x10 * n. When the corresponding TAG_VALID is high, TAG value is valid.
CRYPTO_CHn_TAG_1	0x0324	W	0x00000000	Channel n range from 0 to 7. CHn_TAG_1 address = 0x0324 + 0x10 * n. When the corresponding TAG_VALID is high, TAG value is valid.
CRYPTO_CHn_TAG_2	0x0328	W	0x00000000	Channel n range from 0 to 7. CHn_TAG_2 address = 0x0328 + 0x10 * n. When the corresponding TAG_VALID is high, TAG value is valid.
CRYPTO_CHn_TAG_3	0x032c	W	0x00000000	Channel n range from 0 to 7. CHn_TAG_3 address = 0x032c + 0x10 * n. When the corresponding TAG_VALID is high, TAG value is valid.
CRYPTO_HASH_DOUT_0	0x03a0	W	0x00000000	HASH Data Output Register 0
CRYPTO_HASH_DOUT_1	0x03a4	W	0x00000000	HASH Data Output Register 1
CRYPTO_HASH_DOUT_2	0x03a8	W	0x00000000	HASH Data Output Register 2
CRYPTO_HASH_DOUT_3	0x03ac	W	0x00000000	HASH Data Output Register 3
CRYPTO_HASH_DOUT_4	0x03b0	W	0x00000000	HASH Data Output Register 4
CRYPTO_HASH_DOUT_5	0x03b4	W	0x00000000	HASH Data Output Register 5
CRYPTO_HASH_DOUT_6	0x03b8	W	0x00000000	HASH Data Output Register 6
CRYPTO_HASH_DOUT_7	0x03bc	W	0x00000000	HASH Data Output Register 7

Name	Offset	Size	Reset Value	Description
CRYPTO_HASH_DOUT_8	0x03c0	W	0x00000000	HASH Data Output Register 8
CRYPTO_HASH_DOUT_9	0x03c4	W	0x00000000	HASH Data Output Register 9
CRYPTO_HASH_DOUT_10	0x03c8	W	0x00000000	HASH Data Output Register 10
CRYPTO_HASH_DOUT_11	0x03cc	W	0x00000000	HASH Data Output Register 11
CRYPTO_HASH_DOUT_12	0x03d0	W	0x00000000	HASH Data Output Register 12
CRYPTO_HASH_DOUT_13	0x03d4	W	0x00000000	HASH Data Output Register 13
CRYPTO_HASH_DOUT_14	0x03d8	W	0x00000000	HASH Data Output Register 14
CRYPTO_HASH_DOUT_15	0x03dc	W	0x00000000	HASH Data Output Register 15
CRYPTO_TAG_VALID	0x03e0	W	0x00000000	TAG Valid Register
CRYPTO_HASH_VALID	0x03e4	W	0x00000000	HASH Output Valid Register
CRYPTO_VERSION	0x03f0	W	0x01000001	CRYPTO Version Number Register
CRYPTO_RNG_CTL	0x0400	W	0x0000000c	RNG Control Register
CRYPTO_RNG_SAMPLE_CNT	0x0404	W	0x00000000	RNG Sample Counter Register
CRYPTO_RNG_DOUT_0	0x0410	W	0x00000000	RNG Data Output Register 0
CRYPTO_RNG_DOUT_1	0x0414	W	0x00000000	RNG Data Output Register 1
CRYPTO_RNG_DOUT_2	0x0418	W	0x00000000	RNG Data Output Register 2
CRYPTO_RNG_DOUT_3	0x041c	W	0x00000000	RNG Data Output Register 3
CRYPTO_RNG_DOUT_4	0x0420	W	0x00000000	RNG Data Output Register 4
CRYPTO_RNG_DOUT_5	0x0424	W	0x00000000	RNG Data Output Register 5
CRYPTO_RNG_DOUT_6	0x0428	W	0x00000000	RNG Data Output Register 6
CRYPTO_RNG_DOUT_7	0x042c	W	0x00000000	RNG Data Output Register 7
CRYPTO_RAM_CTL	0x0480	W	0x00000000	RAM Control Register
CRYPTO_RAM_ST	0x0484	W	0x00000001	RAM Status Register
CRYPTO_DEBUG_CTL	0x04a0	W	0x00000000	PKA Debug Control Register
CRYPTO_DEBUG_ST	0x04a4	W	0x00000001	PKA Debug Status Register
CRYPTO_DEBUG_MONITOR	0x04a8	W	0x0000feef	PKA Debug Monitor Bus Register
CRYPTO_PKA_MEM_MAP0	0x0800	W	0x00000000	PKA Memory Map 0 Register
CRYPTO_PKA_MEM_MAP1	0x0804	W	0x00000000	PKA Memory Map 1 Register
CRYPTO_PKA_MEM_MAP2	0x0808	W	0x00000000	PKA Memory Map 2 Register
CRYPTO_PKA_MEM_MAP3	0x080c	W	0x00000000	PKA Memory Map 3 Register
CRYPTO_PKA_MEM_MAP4	0x0810	W	0x00000000	PKA Memory Map 4 Register
CRYPTO_PKA_MEM_MAP5	0x0814	W	0x00000000	PKA Memory Map 5 Register
CRYPTO_PKA_MEM_MAP6	0x0818	W	0x00000000	PKA Memory Map 6 Register
CRYPTO_PKA_MEM_MAP7	0x081c	W	0x00000000	PKA Memory Map 7 Register
CRYPTO_PKA_MEM_MAP8	0x0820	W	0x00000000	PKA Memory Map 8 Register
CRYPTO_PKA_MEM_MAP9	0x0824	W	0x00000000	PKA Memory Map 9 Register
CRYPTO_PKA_MEM_MAP10	0x0828	W	0x00000000	PKA Memory Map 10 Register
CRYPTO_PKA_MEM_MAP11	0x082c	W	0x00000000	PKA Memory Map 11 Register

Name	Offset	Size	Reset Value	Description
CRYPTO_PKA_MEM_MAP12	0x0830	W	0x00000000	PKA Memory Map 12 Register
CRYPTO_PKA_MEM_MAP13	0x0834	W	0x00000000	PKA Memory Map 13 Register
CRYPTO_PKA_MEM_MAP14	0x0838	W	0x00000000	PKA Memory Map 14 Register
CRYPTO_PKA_MEM_MAP15	0x083c	W	0x00000000	PKA Memory Map 15 Register
CRYPTO_PKA_MEM_MAP16	0x0840	W	0x00000000	PKA Memory Map 16 Register
CRYPTO_PKA_MEM_MAP17	0x0844	W	0x00000000	PKA Memory Map 17 Register
CRYPTO_PKA_MEM_MAP18	0x0848	W	0x00000000	PKA Memory Map 18 Register
CRYPTO_PKA_MEM_MAP19	0x084c	W	0x00000000	PKA Memory Map 19 Register
CRYPTO_PKA_MEM_MAP20	0x0850	W	0x00000000	PKA Memory Map 20 Register
CRYPTO_PKA_MEM_MAP21	0x0854	W	0x00000000	PKA Memory Map 21 Register
CRYPTO_PKA_MEM_MAP22	0x0858	W	0x00000000	PKA Memory Map 22 Register
CRYPTO_PKA_MEM_MAP23	0x085c	W	0x00000000	PKA Memory Map 23 Register
CRYPTO_PKA_MEM_MAP24	0x0860	W	0x00000000	PKA Memory Map 24 Register
CRYPTO_PKA_MEM_MAP25	0x0864	W	0x00000000	PKA Memory Map 25 Register
CRYPTO_PKA_MEM_MAP26	0x0868	W	0x00000000	PKA Memory Map 26 Register
CRYPTO_PKA_MEM_MAP27	0x086c	W	0x00000000	PKA Memory Map 27 Register
CRYPTO_PKA_MEM_MAP28	0x0870	W	0x00000000	PKA Memory Map 28 Register
CRYPTO_PKA_MEM_MAP29	0x0874	W	0x00000000	PKA Memory Map 29 Register
CRYPTO_PKA_MEM_MAP30	0x0878	W	0x00000000	PKA Memory Map 30 Register
CRYPTO_PKA_MEM_MAP31	0x087c	W	0x00000000	PKA Memory Map 31 Register
CRYPTO_PKA_OPCODE	0x0880	W	0x00000000	PKA Operation Code Register
CRYPTO_N_NP_TO_T1_ADDR	0x0884	W	0x000ff820	N_NP_TO_T1_ADDR Register

Name	Offset	Size	Reset Value	Description
CRYPTO_PKA_STATUS	0x0888	W	0x00000001	PKA Status Register
CRYPTO_PKA_SW_RESET	0x088c	W	0x00000000	software reset of PKA
CRYPTO_PKA_L0	0x0890	W	0x00000000	PKA Length 0 Register
CRYPTO_PKA_L1	0x0894	W	0x00000000	PKA Length 1 Register
CRYPTO_PKA_L2	0x0898	W	0x00000000	PKA Length 2 Register
CRYPTO_PKA_L3	0x089c	W	0x00000000	PKA Length 3 Register
CRYPTO_PKA_L4	0x08a0	W	0x00000000	PKA Length 4 Register
CRYPTO_PKA_L5	0x08a4	W	0x00000000	PKA Length 5 Register
CRYPTO_PKA_L6	0x08a8	W	0x00000000	PKA Length 6 Register
CRYPTO_PKA_L7	0x08ac	W	0x00000000	PKA Length 7 Register
CRYPTO_PKA_PIPE_RDY	0x08b0	W	0x00000001	PKA pipe is ready for new opcode.
CRYPTO_PKA_DONE	0x08b4	W	0x00000001	PKA Done Register
CRYPTO_PKA_MON_SELECT	0x08b8	W	0x00000000	PKA Monitor Select Register
CRYPTO_PKA_DEBUG_REG_EN	0x08bc	W	0x00000000	PKA Debug Enable Register
CRYPTO_DEBUG_CNT_ADDR	0x08c0	W	0x00000000	Debug Counter Address Register
CRYPTO_DEBUG_EXT_ADDR	0x08c4	W	0x00000000	Debug Extra Address Register
CRYPTO_PKA_DEBUG_HALT	0x08c8	W	0x00000000	PKA Debug Halt State Register
CRYPTO_PKA_MON_READ	0x08d0	W	0x0000feef	PKA Monitor Read Register
CRYPTO_PKA_INT_ENA	0x08d4	W	0x00000000	PKA Interrupt Enable Register
CRYPTO_PKA_INT_ST	0x08d8	W	0x00000000	PKA Interrupt Status Register
CRYPTO_SRAM_ADDR	0x1000	W	0x00000000	SRAM Base Address

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

9.3.3 Detail Register Description

CRYPTO_CLK_CTL

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16	WO	0x0	write_enable When bit n=1, the corresponding bit n-16 can be written by software; When bit n=0, the corresponding bit n-16 can't be written by software.
15:1	RO	0x0	reserved
0	RW	0x1	auto_clkgate_en CRYPTO will gate unused Block Cipher and HASH module automatically. 1'b0: disable; 1'b1: enable.

CRYPTO_RST_CTL

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18:16	RW	0x0	write_enable When bit n=1, the corresponding bit n-16 can be written by software; When bit n=0, the corresponding bit n-16 can't be written by software.
15:3	RO	0x0	reserved
2	R/W SC	0x0	sw_pka_reset Software set this bit to start a reset to PKA module. After the reset is done, CRYPTO will clear this bit.
1	R/W SC	0x0	sw_rng_reset Software set this bit to start a reset to TRNG module. After the reset is done, CRYPTO will clear this bit.
0	R/W SC	0x0	sw_cc_reset Software set this bit to start a reset to Symmetric Cipher and HASH module. After the reset is done, CRYPTO will clear this bit.

CRYPTO_DMA_INT_EN

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	zero_len_int_en 1'b1: enable; 1'b0: disable.
5	RW	0x0	list_err_int_en 1'b1: enable; 1'b0: disable.
4	RW	0x0	src_err_int_en 1'b1: enable; 1'b0: disable.
3	RW	0x0	dst_err_int_en 1'b1: enable; 1'b0: disable.
2	RW	0x0	src_item_done_int_en 1'b1: enable; 1'b0: disable.
1	RW	0x0	dst_item_done_int_en 1'b1: enable; 1'b0: disable.

Bit	Attr	Reset Value	Description
0	RW	0x0	list_done_int_en 1'b1: enable; 1'b0: disable.

CRYPTO_DMA_INT_ST
Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	W1C	0x0	zero_len 1'b1: indicate that DMA has met an 0 byte source transfer length in list descriptors. After the bit is read, the application should write 1 to clear this bit for next time use. 1'b0: nothing.
5	RO	0x0	reserved
4	W1C	0x0	src_err 1'b1: indicate that DMA has met an error response when transfer source data. The state machine will exit current transfer and then return to IDLE state. After the bit is read, the application should write 1 to clear this bit for next time use. 1'b0: nothing.
3	W1C	0x0	dst_err 1'b1: indicate that DMA has met an error response when transfer destination data. The state machine will exit current transfer and then return to IDLE state. After the bit is read, the application should write 1 to clear this bit for next time use; 1'b0: nothing.
2	W1C	0x0	src_item_done 1'b1:Indicate that DMA has completed a read transfers which the current list descriptor pointed to. After the bit is read, the application should write 1 to clear this bit for next time use. 1'b0: nothing.
1	W1C	0x0	dst_item_done 1'b1: indicate that DMA has completed a write transfers which the current list descriptor pointed to. After the bit is read, the application should write 1 to clear this bit for next time use; 1'b0: nothing.
0	W1C	0x0	list_done 1'b1: indicate that DMA has completed all the transfers which the list descriptors pointed to. After the bit is read, the application should write 1 to clear this bit for next time use ; 1'b0: nothing.

CRYPTO_DMA_CTL
Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:18	RO	0x0	reserved
17:16	WO	0x0	write_enable When bit n=1, the corresponding bit n-16 can be written by software; When bit n=0, the corresponding bit n-16 can't be written by software.
15:2	RO	0x0	reserved
1	R/W SC	0x0	dma_restart If DMA data for next stage is not ready, application could pause DMA by descriptor commands. DMA will stop prefetching next descriptor. The application could restart DMA by asserting this bit when DMA data for next state is ready. Crypto will continue with previous transfer, and clear the bit automatically.
0	R/W SC	0x0	dma_start DMA asserts the bit to start DMA transfer, then Crypto will clear the bit automatically.

CRYPTO_DMA_LLI_ADDR

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dma_lli_addr When DMA_CTL start asserted, Crypto will read the address to get the 1'st descriptor. It should be 8-bytes align. We suggest dma_lli_addr 64-byte align for best performance consideration.

CRYPTO_DMA_ST

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	dma_busy 1'b1: dma busy 1'b0: dma idle

CRYPTO_DMA_STATE

Address: Operational Base + offset (0x001c)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved

5:4	RO	0x0	dma_lli_state
			For debug use only.
			2'b00: IDLE STATE;
			2'b01: FETCH STATE;
			2'b10: WORK STATE;
			Others: Reserved.

Bit	Attr	Reset Value	Description

3:2	RO	0x0	dma_src_state
			For debug use only.
			2'b00: IDLE STATE;
			2'b01: LOAD STATE;
			2'b10: WORK STATE;
			Others: Reserved.

1:0	RO	0x0	dma_dst_state
			For debug use only.
			2'b00: IDLE STATE;
			2'b01: LOAD STATE;
			2'b10: WORK STATE;
			Others: Reserved.

CRYPTO_DMA_LLI_RADDR
Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dma_lli_raddr For debug use only. It indicates the current dma lli read address.

CRYPTO_DMA_SRC_RADDR
Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dma_src_raddr For debug use only. It indicates the current dma source read address.

CRYPTO_DMA_DST_WADDR
Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dma_dst_waddr For debug use only. It indicates the current dma destination write address.

CRYPTO_DMA_ITEM_ID
Address: Operational Base + offset (0x002c)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	dma_item_id For debug use only. It indicates the current descriptor ID.

CRYPTO_FIFO_CTL

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:18	RO	0x0	reserved
17:16	WO	0x0	write_enable When bit n=1, the corresponding bit n-16 can be written by software; When bit n=0, the corresponding bit n-16 can't be written by software.
15:2	RO	0x0	reserved
1	RW	0x1	dout_byteswap 1'b1: little endian; 1'b0: big endian.
0	RW	0x1	din_byteswap 1'b1: little endian; 1'b0: big endian.

CRYPTO_BC_CTL

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:26	RO	0x0	reserved
25:16	WO	0x000	write_enable When bit n=1, the corresponding bit n-16 can be written by software; When bit n=0, the corresponding bit n-16 can't be written by software.
15:10	RO	0x0	reserved

9:8	RW	0x0	bc_cipher_sel
			2'b00: AES;
			2'b10: DES;
			2'b11: TDES;
			Others:Reserved.

Bit	Attr	Reset Value	Description

7:4	RW	0x0	mode
			For AES,
			4'h0: ECB;
			4'h1: CBC;
			4'h2: CTS;
			4'h3: CTR;
			4'h4: CFB;
			4'h5: OFB;
			4'h6: XTS;
			4'h7: CCM;
			4'h8: GCM;
			4'h9: CMAC;
			4'hA: CBC-MAC;
			Others: Reserved.
			For TDES/DES,
			4'h0: ECB;
			4'h1: CBC;
			4'h4: CFB;
			4'h5: OFB;
			Others: Reserved.

3:2	RW	0x0	key_size
			For AES,
			2'b00: 128 bit;
			2'b01: 192 bit;
			2'b10: 256 bit;
			2'b11: reserved;
			For TDES/DES, it is reserved.

1	RW	0x0	decrypt 1'b1: decrypt; 1'b0: encrypt.
0	RW	0x0	bc_enable 1'b1: enable; 1'b0: disable.

CRYPTO_HASH_CTL

Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:16	WO	0x00	write_enable When bit n=1, the corresponding bit n-16 can be written by software; When bit n=0, the corresponding bit n-16 can't be written by software.
15:8	RO	0x0	reserved

7:4	RW	0x0	hash_cipher_sel
			4'h0: SHA-1;
			4'h1: MD-5;
			4'h2: SHA-256;
			4'h3: SHA-224;
			4'h8: SHA-512;
			4'h9: SHA-384;
			4'hA: SHA-512/224;
			4'hB: SHA-512/256;
			Others: Reserved.

3	RW	0x0	hmac_enable Crypto supports HMAC-SHA1, HMAC-SHA256, HMAC_SHA512. 1'b1: enable; 1'b0: disable.
2	RW	0x1	hw_pad_enable 1'b1: enable; 1'b0: disable.
1	RW	0x0	hash_src_sel 1'b1: from TX-FIFO; 1'b0: from RX-FIFO.
0	RW	0x0	hash_enable 1'b1: enable; 1'b0: disable.

CRYPTO_CIPHER_ST

Address: Operational Base + offset (0x004c)

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	RO	0x0	otp_key_valid Indicate if otp_key is valid. 1'b1: valid 1'b0: invalid
1	RO	0x0	hash_busy 1'b1: busy 1'b0: idle
0	RO	0x0	block_cipher_busy 1'b1: busy 1'b0: idle

CRYPTO_CIPHER_STATE

Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:15	RO	0x0	reserved

14:10	RO	0x01	hash_state
			For debug use only,
			5'h01: IDLE State;
			5'h02: IPAD State;
			5'h04: TEXT State;
			5'h08: OPAD State;
			5'h10: OPAD EXT State.

9:8	RO	0x0	gcm_state
			For debug use only,
			2'b00: IDLE State;
			2'b01: PRE State;
			2'b10: NA State;
			2'b11: PC State.

7:6	RO	0x0	ccm_state
			For debug use only,
			2'b00: IDLE State;
			2'b01: PRE State;
			2'b10: NA State;
			2'b11: PC State.

5:4	RO	0x0	parallel_state
			For debug use only,
			2'b00: IDLE State;
			2'b01: PRE State;
			2'b10: BULK State;
			Others: Reserved.

3:2	RO	0x0	mac_state
			For debug use only,
			2'b00: IDLE State;
			2'b01: PRE State;
			2'b10: BULK State;
			Others: Reserved.

1:0	RO	0x0	serial_state
			For debug use only,
			2'b00: IDLE State;
			2'b01: PRE State;
			2'b10: BULK State;
			2'b11: Reserved.

CRYPTO_CHn_IV_0

Address: Operational Base + offset (0x0100)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	chn_iv_0 Channel N IV[127:96].

CRYPTO_CHn_IV_1
Address: Operational Base + offset (0x0104)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	chn_iv_1 Channel N IV[95:64].

CRYPTO_CHn_IV_2
Address: Operational Base + offset (0x0108)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	chn_iv_2 Channel N IV[63:32].

CRYPTO_CHn_IV_3
Address: Operational Base + offset (0x010c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	chn_iv_3 Channel N IV[31:0].

CRYPTO_CHn_KEY_0
Address: Operational Base + offset (0x0180)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	chn_key_0 Channel N Key[127:96].

CRYPTO_CHn_KEY_1
Address: Operational Base + offset (0x0184)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	chn_key_1 Channel N Key[95:64].

CRYPTO_CHn_KEY_2
Address: Operational Base + offset (0x0188)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	chn_key_2 Channel N Key[63:32].

CRYPTO_CHn_KEY_3
Address: Operational Base + offset (0x018c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	chn_key_3 Channel N Key[31:0].

CRYPTO_CHn_PKEY_0
Address: Operational Base + offset (0x0200)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	chn_pkey_0 Channel N PKey[127:96].

CRYPTO_CHn_PKEY_1
Address: Operational Base + offset (0x0204)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	chn_pkey_1 Channel N PKey[95:64].

CRYPTO_CHn_PKEY_2
Address: Operational Base + offset (0x0208)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	chn_pkey_2 Channel N PKey[63:32].

CRYPTO_CHn_PKEY_3
Address: Operational Base + offset (0x020c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	chn_key_3 Channel N PKey[31:0].

CRYPTO_CHn_PC_LEN_0
Address: Operational Base + offset (0x0280)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	chn_pc_len_0 Channel N Plain/Cipher Text Length[31:0]

CRYPTO_CHn_PC_LEN_1
Address: Operational Base + offset (0x0284)

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28:0	RW	0x00000000	chn_pc_len_1 Channel N Plain/Cipher Text Length[60:32].

CRYPTO_CHn_ADA_LEN_0
Address: Operational Base + offset (0x02c0)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	chn_ada_len_0 Channel N additional data Length[31:0].

CRYPTO_CHn_ADA_LEN_1

Address: Operational Base + offset (0x02c4)

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28:0	RW	0x00000000	chn_ada_len_1 Channel N additional data Length[60:32].

CRYPTO_CHn_IV_LEN_0

Address: Operational Base + offset (0x0300)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	chn_iv_len Channel N initial vector length.

CRYPTO_CHn_TAG_0

Address: Operational Base + offset (0x0320)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	chn_tag_0 Channel N tag[127:96].

CRYPTO_CHn_TAG_1

Address: Operational Base + offset (0x0324)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	chn_tag_1 Channel N tag[95:64].

CRYPTO_CHn_TAG_2

Address: Operational Base + offset (0x0328)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	chn_tag_2 Channel N tag[63:32].

CRYPTO_CHn_TAG_3

Address: Operational Base + offset (0x032c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	chn_tag_3 Channel N tag[31:0].

CRYPTO_HASH_DOUT_0

Address: Operational Base + offset (0x03a0)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	hash_dout_0 0'th output word for all hash function, in big endian

CRYPTO_HASH_DOUT_1
Address: Operational Base + offset (0x03a4)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	hash_dout_1 1'th output word for all hash function, in big endian

CRYPTO_HASH_DOUT_2
Address: Operational Base + offset (0x03a8)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	hash_dout_2 2'th output word for all hash function, in big endian

CRYPTO_HASH_DOUT_3
Address: Operational Base + offset (0x03ac)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	hash_dout_3 3'th output word for all hash function, in big endian. This is MD5 last output word. HASH_DOUT_4 ~ HASH_DOUT_15 is invalid data for MD5.

CRYPTO_HASH_DOUT_4
Address: Operational Base + offset (0x03b0)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	hash_dout_4 4'th output word for all hash function, in big endian. This is SHA-1 last output word. HASH_DOUT_5 ~ HASH_DOUT_15 is invalid data for SHA-1.

CRYPTO_HASH_DOUT_5
Address: Operational Base + offset (0x03b4)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	hash_dout_5 5'th output word for all hash function, in big endian

CRYPTO_HASH_DOUT_6
Address: Operational Base + offset (0x03b8)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	hash_dout_6 6'th output word for all hash function, in big endian

CRYPTO_HASH_DOUT_7

Address: Operational Base + offset (0x03bc)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	hash_dout_7 7'th output word for all hash function, in big endian. This is SHA-256/224 last output word. HASH_DOUT_8 ~ HASH_DOUT_15 is invalid data for SHA-256/224.

CRYPTO_HASH_DOUT_8

Address: Operational Base + offset (0x03c0)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	hash_dout_8 8'th output word for all hash function, in big endian

CRYPTO_HASH_DOUT_9

Address: Operational Base + offset (0x03c4)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	hash_dout_9 9'th output word for all hash function, in big endian

CRYPTO_HASH_DOUT_10

Address: Operational Base + offset (0x03c8)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	hash_dout_10 10'th output word for all hash function, in big endian

CRYPTO_HASH_DOUT_11

Address: Operational Base + offset (0x03cc)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	hash_dout_11 11'th output word for all hash function, in big endian.

CRYPTO_HASH_DOUT_12

Address: Operational Base + offset (0x03d0)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	hash_dout_12 12'th output word for all hash function, in big endian

CRYPTO_HASH_DOUT_13

Address: Operational Base + offset (0x03d4)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	hash_dout_13 13'th output word for all hash function, in big endian

CRYPTO_HASH_DOUT_14

Address: Operational Base + offset (0x03d8)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	hash_dout_14 14'th output word for all hash function, in big endian

CRYPTO_HASH_DOUT_15

Address: Operational Base + offset (0x03dc)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	hash_dout_15 15'th output word for all hash function, in big endian. This is SHA-512, SHA-384, SHA-512/224, SHA-512/256 last output word.

CRYPTO_TAG_VALID

Address: Operational Base + offset (0x03e0)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	W1C	0x0	ch7_tag_valid When channel 7 tag calculation is finished, CRYPTO will set this bit. After the bit is read by application, it should be cleared by writing 1 to this bit. 1'b1: channel 7 tag is valid 1'b0: channel 7 tag is invalid
6	W1C	0x0	ch6_tag_valid When channel 6 tag calculation is finished, CRYPTO will set this bit. After the bit is read by application, it should be cleared by writing 1 to this bit. 1'b1: channel 6 tag is valid 1'b0: channel 6 tag is invalid
5	W1C	0x0	ch5_tag_valid When channel 5 tag calculation is finished, CRYPTO will set this bit. After the bit is read by application, it should be cleared by writing 1 to this bit. 1'b1: channel 5 tag is valid 1'b0: channel 5 tag is invalid
4	W1C	0x0	ch4_tag_valid When channel 4 tag calculation is finished, CRYPTO will set this bit. After the bit is read by application, it should be cleared by writing 1 to this bit. 1'b1: channel 4 tag is valid 1'b0: channel 4 tag is invalid

Bit	Attr	Reset Value	Description
3	W1C	0x0	ch3_tag_valid When channel 3 tag calculation is finished, CRYPTO will set this bit. After the bit is read by application, it should be cleared by writing 1 to this bit. 1'b1: channel 3 tag is valid 1'b0: channel 3 tag is invalid
2	W1C	0x0	ch2_tag_valid When channel 2 tag calculation is finished, CRYPTO will set this bit. After the bit is read by application, it should be cleared by writing 1 to this bit. 1'b1: channel 2 tag is valid 1'b0: channel 2 tag is invalid
1	W1C	0x0	ch1_tag_valid When channel 1 tag calculation is finished, CRYPTO will set this bit. After the bit is read by application, it should be cleared by writing 1 to this bit. 1'b1: channel 1 tag is valid 1'b0: channel 1 tag is invalid
0	W1C	0x0	ch0_tag_valid When channel 0 tag calculation is finished, CRYPTO will set this bit. After the bit is read by application, it should be cleared by writing 1 to this bit. 1'b1: channel 0 tag is valid 1'b0: channel 0 tag is invalid

CRYPTO_HASH_VALID

Address: Operational Base + offset (0x03e4)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	W1C	0x0	hash_valid When HASH calculation is finished, CRYPTO will set this bit. After the bit is read by application, it should be cleared by writing 1 to this bit. 1'b1: HASH_DOUT is valid 1'b0: HASH_DOUT is invalid

CRYPTO_VERSION

Address: Operational Base + offset (0x03f0)

Bit	Attr	Reset Value	Description
31:0	RW	0x01000001	version_num Version number: V1.0.0.1.

CRYPTO_RNG_CTL

Address: Operational Base + offset (0x0400)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21:16	WO	0x00	write_enable When bit n=1, the corresponding bit n-16 can be written by software; When bit n=0, the corresponding bit n-16 can't be written by software.
15:6	RO	0x0	reserved
5:4	RW	0x0	rng_len 2'b00: 64 bit 2'b01: 128 bit 2'b10: 192 bit 2'b11: 256 bit
3:2	RW	0x3	ring_sel There are 4 osc rings choice to decide the rng output data. 2'b00: fatest osc ring 2'b01: slower than osc ring 0 2'b10: slower than osc ring 1 2'b11: slowest osc ring
1	RW	0x0	rng_enable 1'b1: enable 1'b0: disable
0	R/W SC	0x0	rng_start The application triggers this bit to start collect rng output data. After rng is started, CRYPTO will clear the bit automatically. 1'b1: start 1'b0: do nothing

CRYPTO_RNG_SAMPLE_CNT
Address: Operational Base + offset (0x0404)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	rng_sample_cnt RNG collects osc ring output bit every rng_sample_cnt time. The value of rng_sample_cnt affects RNG output data rate, the value more bigger, the rate more slower.

CRYPTO_RNG_DOUT_0
Address: Operational Base + offset (0x0410)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rng_dout_0 The 32'th osc ring bit is captured in RNG_DOUT_0.bit31.

CRYPTO_RNG_DOUT_1
Address: Operational Base + offset (0x0414)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rng_dout_1 The 64'th osc ring bit is captured in RNG_DOUT_1.bit31. If RNG_CTL.rng_len = 0x00, the last valid bit of RNG is stored in RNG_DOUT_1.bit31, and RNG_DOUT_2 ~ RNG_DOUT_7 are invalid.

CRYPTO_RNG_DOUT_2

Address: Operational Base + offset (0x0418)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rng_dout_2 The 96'th osc ring bit is captured in RNG_DOUT_2.bit31.

CRYPTO_RNG_DOUT_3

Address: Operational Base + offset (0x041c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rng_dout_3 The 128'th osc ring bit is captured in RNG_DOUT_3.bit31. If RNG_CTL rng_len = 0x01, the last valid bit of RNG is stored in RNG_DOUT_3.bit31, and RNG_DOUT_4 ~ RNG_DOUT_7 are invalid.

CRYPTO_RNG_DOUT_4

Address: Operational Base + offset (0x0420)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rng_dout_4 The 160'th osc ring bit is captured in RNG_DOUT_4.bit31.

CRYPTO_RNG_DOUT_5

Address: Operational Base + offset (0x0424)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rng_dout_5 The 192'th osc ring bit is captured in RNG_DOUT_5.bit31. If RNG_CTL.rng_len = 0x02, the last valid bit of RNG is stored in RNG_DOUT_5.bit31, and RNG_DOUT_6~ RNG_DOUT_7 are invalid.

CRYPTO_RNG_DOUT_6

Address: Operational Base + offset (0x0428)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rng_dout_6 The 224'th osc ring bit is captured in RNG_DOUT_6.bit31.

CRYPTO_RNG_DOUT_7

Address: Operational Base + offset (0x042c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rng_dout_7 The 256'th osc ring bit is captured in RNG_DOUT_7.bit31. If RNG_CTL.rng_len = 0x03, the last valid bit of RNG is stored in RNG_DOUT_7.bit31.

CRYPTO_RAM_CTL

Address: Operational Base + offset (0x0480)

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16	WO	0x0	write_enable When bit n=1, the corresponding bit n-16 can be written by software; When bit n=0, the corresponding bit n-16 can't be written by software.
15:1	RO	0x0	reserved
0	RW	0x0	ram_pka_rdy Indicate whether ram is controlled by PKA engine. 1'b0: ram is controlled by CPU 1'b1: ram is controlled by CRYPTO PKA engine

CRYPTO_RAM_ST

Address: Operational Base + offset (0x0484)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x1	clk_ram_rdy Indicate whether clk_ram is stable, and ready for use. 1'b0: not stable 1'b1: stable

CRYPTO_DEBUG_CTL

Address: Operational Base + offset (0x04a0)

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16	WO	0x0	write_enable When bit n=1, the corresponding bit n-16 can be written by software; When bit n=0, the corresponding bit n-16 can't be written by software.
15:1	RO	0x0	reserved
0	RW	0x0	pka_debug_mode 1'b1: PKA is in debug mode 1'b0: PKA is in normal mode

CRYPTO_DEBUG_ST

Address: Operational Base + offset (0x04a4)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x1	pka_debug_clk_en For debug use only, 1'b1: enable 1'b0: disable

CRYPTO_DEBUG_MONITOR

Address: Operational Base + offset (0x04a8)

Bit	Attr	Reset Value	Description
31:0	RW	0x0000feef	pka_monitor_bus For debug use only.

CRYPTO_PKA_MEM_MAP0

Address: Operational Base + offset (0x0800)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:2	RW	0x000	memory_map0 Memory map 0 [11:2], bit[1:0] is stuck to 0.
1:0	RO	0x0	reserved

CRYPTO_PKA_MEM_MAP1

Address: Operational Base + offset (0x0804)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:2	RW	0x000	memory_map1 Memory map 1 [11:2], bit[1:0] is stuck to 0.
1:0	RO	0x0	reserved

CRYPTO_PKA_MEM_MAP2

Address: Operational Base + offset (0x0808)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:2	RW	0x000	memory_map2 Memory map 2 [11:2], bit[1:0] is stuck to 0.
1:0	RO	0x0	reserved

CRYPTO_PKA_MEM_MAP3

Address: Operational Base + offset (0x080c)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:2	RW	0x000	memory_map3 Memory map 3 [11:2], bit[1:0] is stuck to 0.
1:0	RO	0x0	reserved

CRYPTO_PKA_MEM_MAP4

Address: Operational Base + offset (0x0810)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:2	RW	0x000	memory_map4 Memory map 4 [11:2], bit[1:0] is stuck to 0.
1:0	RO	0x0	reserved

CRYPTO_PKA_MEM_MAP5

Address: Operational Base + offset (0x0814)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:2	RW	0x000	memory_map5 Memory map 5 [11:2], bit[1:0] is stuck to 0.
1:0	RO	0x0	reserved

CRYPTO_PKA_MEM_MAP6

Address: Operational Base + offset (0x0818)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:2	RW	0x000	memory_map6 Memory map 6 [11:2], bit[1:0] is stuck to 0.
1:0	RO	0x0	reserved

CRYPTO_PKA_MEM_MAP7

Address: Operational Base + offset (0x081c)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:2	RW	0x000	memory_map7 Memory map 7 [11:2], bit[1:0] is stuck to 0.
1:0	RO	0x0	reserved

CRYPTO_PKA_MEM_MAP8

Address: Operational Base + offset (0x0820)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:2	RW	0x000	memory_map8 Memory map 8 [11:2], bit[1:0] is stuck to 0.
1:0	RO	0x0	reserved

CRYPTO_PKA_MEM_MAP9

Address: Operational Base + offset (0x0824)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:2	RW	0x000	memory_map9 Memory map 9 [11:2], bit[1:0] is stuck to 0.
1:0	RO	0x0	reserved

CRYPTO_PKA_MEM_MAP10

Address: Operational Base + offset (0x0828)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:2	RW	0x000	memory_map10 Memory map 10 [11:2], bit[1:0] is stuck to 0.
1:0	RO	0x0	reserved

CRYPTO_PKA_MEM_MAP11

Address: Operational Base + offset (0x082c)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:2	RW	0x000	memory_map11 Memory map 11 [11:2], bit[1:0] is stuck to 0.
1:0	RO	0x0	reserved

CRYPTO_PKA_MEM_MAP12

Address: Operational Base + offset (0x0830)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:2	RW	0x000	memory_map12 Memory map 12 [11:2], bit[1:0] is stuck to 0.
1:0	RO	0x0	reserved

CRYPTO_PKA_MEM_MAP13

Address: Operational Base + offset (0x0834)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:2	RW	0x000	memory_map13 Memory map 13 [11:2], bit[1:0] is stuck to 0.
1:0	RO	0x0	reserved

CRYPTO_PKA_MEM_MAP14

Address: Operational Base + offset (0x0838)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:2	RW	0x000	memory_map14 Memory map 14 [11:2], bit[1:0] is stuck to 0.
1:0	RO	0x0	reserved

CRYPTO_PKA_MEM_MAP15

Address: Operational Base + offset (0x083c)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:2	RW	0x000	memory_map15 Memory map 15 [11:2], bit[1:0] is stuck to 0.
1:0	RO	0x0	reserved

CRYPTO_PKA_MEM_MAP16

Address: Operational Base + offset (0x0840)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:2	RW	0x000	memory_map16 Memory map 16 [11:2], bit[1:0] is stuck to 0.
1:0	RO	0x0	reserved

CRYPTO_PKA_MEM_MAP17

Address: Operational Base + offset (0x0844)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:2	RW	0x000	memory_map17 Memory map 17 [11:2], bit[1:0] is stuck to 0.
1:0	RO	0x0	reserved

CRYPTO_PKA_MEM_MAP18

Address: Operational Base + offset (0x0848)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:2	RW	0x000	memory_map18 Memory map 18 [11:2], bit[1:0] is stuck to 0.
1:0	RO	0x0	reserved

CRYPTO_PKA_MEM_MAP19
Address: Operational Base + offset (0x084c)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:2	RW	0x000	memory_map19 Memory map 19 [11:2], bit[1:0] is stuck to 0.
1:0	RO	0x0	reserved

CRYPTO_PKA_MEM_MAP20
Address: Operational Base + offset (0x0850)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:2	RW	0x000	memory_map20 Memory map 20 [11:2], bit[1:0] is stuck to 0.
1:0	RO	0x0	reserved

CRYPTO_PKA_MEM_MAP21
Address: Operational Base + offset (0x0854)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:2	RW	0x000	memory_map21 Memory map 21 [11:2], bit[1:0] is stuck to 0.
1:0	RO	0x0	reserved

CRYPTO_PKA_MEM_MAP22
Address: Operational Base + offset (0x0858)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:2	RW	0x000	memory_map22 Memory map 22 [11:2], bit[1:0] is stuck to 0.
1:0	RO	0x0	reserved

CRYPTO_PKA_MEM_MAP23
Address: Operational Base + offset (0x085c)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:2	RW	0x000	memory_map23 Memory map 23 [11:2], bit[1:0] is stuck to 0.
1:0	RO	0x0	reserved

CRYPTO_PKA_MEM_MAP24

Address: Operational Base + offset (0x0860)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:2	RW	0x000	memory_map24 Memory map 24 [11:2], bit[1:0] is stuck to 0.
1:0	RO	0x0	reserved

CRYPTO_PKA_MEM_MAP25

Address: Operational Base + offset (0x0864)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:2	RW	0x000	memory_map25 Memory map 25 [11:2], bit[1:0] is stuck to 0.
1:0	RO	0x0	reserved

CRYPTO_PKA_MEM_MAP26

Address: Operational Base + offset (0x0868)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:2	RW	0x000	memory_map26 Memory map 26 [11:2], bit[1:0] is stuck to 0.
1:0	RO	0x0	reserved

CRYPTO_PKA_MEM_MAP27

Address: Operational Base + offset (0x086c)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:2	RW	0x000	memory_map27 Memory map 27 [11:2], bit[1:0] is stuck to 0.
1:0	RO	0x0	reserved

CRYPTO_PKA_MEM_MAP28

Address: Operational Base + offset (0x0870)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:2	RW	0x000	memory_map28 Memory map 28 [11:2], bit[1:0] is stuck to 0.
1:0	RO	0x0	reserved

CRYPTO_PKA_MEM_MAP29

Address: Operational Base + offset (0x0874)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:2	RW	0x000	memory_map29 Memory map 29 [11:2], bit[1:0] is stuck to 0.
1:0	RO	0x0	reserved

CRYPTO_PKA_MEM_MAP30

Address: Operational Base + offset (0x0878)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:2	RW	0x000	memory_map30 Memory map 30 [11:2], bit[1:0] is stuck to 0.
1:0	RO	0x0	reserved

CRYPTO_PKA_MEM_MAP31

Address: Operational Base + offset (0x087c)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:2	RW	0x000	memory_map31 Memory map 30 [11:2], bit[1:0] is stuck to 0.
1:0	RO	0x0	reserved

CRYPTO_PKA_OPCODE

Address: Operational Base + offset (0x0880)

Bit	Attr	Reset Value	Description
31:27	WO	0x00	opcode Defines the PKA operation 5'h04: Add,Inc 5'h05: Sub,Dec,Neg 5'h06: ModAdd,ModInc 5'h07: ModSub,ModDec,ModNeg 5'h08: AND,TST0,CLR0 5'h09: OR,COPY,SET0 5'h0A: XOR,FLIP0,INVERT,COMPARE 5'h0B: SHR0 5'h0D: SHR1 5'h0E: SHL0 5'h0F: SHL1 5'h10: MulLow 5'h11: ModMul 5'h12: ModMulN 5'h13: ModExp 5'h14: Division 5'h15: Div 5'h16: ModDiv 5'h00: Terminate
26:24	WO	0x0	len The virtual length address 0-7. Virtual address 0 point to PKA_L0. Virtual address 1 point to PKA_L1. ... Virtual address 7 point to PKA_L7.
23:18	WO	0x00	reg_a Operand A virtual address 0-15. Virtual address 0 point to PKA_MEM_MAP0. Virtual address 1 point to PKA_MEM_MAP1. ... Virtual address 15 point to PKA_MEM_MAP15.
17:12	WO	0x00	reg_b Operand B virtual address 0-15. Virtual address 0 point to PKA_MEM_MAP0. Virtual address 1 point to PKA_MEM_MAP1. ... Virtual address 15 point to PKA_MEM_MAP15.
11:6	WO	0x00	reg_r Result register virtual address 0-15. Virtual address 0 point to PKA_MEM_MAP0. Virtual address 1 point to PKA_MEM_MAP1. ... Virtual address 15 point to PKA_MEM_MAP15.

Bit	Attr	Reset Value	Description
5:0	WO	0x00	tag Tag.

CRYPTO_N_NP_TO_T1_ADDR

Address: Operational Base + offset (0x0884)

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:15	RW	0x1f	reg_t1 Virtual address of temporary register number 1. Virtual address 0 point to PKA_MEM_MAP0. Virtual address 1 point to PKA_MEM_MAP1. ... Virtual address 15 point to PKA_MEM_MAP15.
14:10	RW	0x1e	reg_t0 Virtual address of temporary register number 0. Virtual address 0 point to PKA_MEM_MAP0. Virtual address 1 point to PKA_MEM_MAP1. ... Virtual address 15 point to PKA_MEM_MAP15.
9:5	RW	0x01	reg_np Virtual address of register np. Virtual address 0 point to PKA_MEM_MAP0. Virtual address 1 point to PKA_MEM_MAP1. ... Virtual address 15 point to PKA_MEM_MAP15.
4:0	RW	0x00	reg_n Virtual address of register n. Virtual address 0 point to PKA_MEM_MAP0. Virtual address 1 point to PKA_MEM_MAP1. ... Virtual address 15 point to PKA_MEM_MAP15.

CRYPTO_PKA_STATUS

Address: Operational Base + offset (0x0888)

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:14	RO	0x00	tag Tag of the Last Operation
13:9	RO	0x00	opcode The last OPCODE
8	RO	0x0	pka_cpu_busy PKA is busy memory control is by PKA.
7	RO	0x0	modinv_of_zero Modular inverse of zero flag

Bit	Attr	Reset Value	Description
6	RO	0x0	alu_sign_out Sign of the last operation(MSB)
5	RO	0x0	alu_carry Carry of the last ALU operation
4	RO	0x0	div_by_zero Division by 0
3	RO	0x0	alu_mod_ovflw Modular overflow flag
2	RO	0x0	alu_out_zero ALU out is 0.
1	RO	0x0	pka_busy PKA is busy.
0	RO	0x1	pipe_is_busy PKA ready signal 1'b0: pipe full 1'b1: PKA ready for new command

CRYPTO_PKA_SW_RESET
Address: Operational Base + offset (0x088c)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	WO	0x0	pka_sw_reset PKA software reset the reset mechanism will take about four PKA clocks until the reset line is de-asserted.

CRYPTO_PKA_L0
Address: Operational Base + offset (0x0890)

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12:0	RW	0x0000	pka_l0 PKA length 0, in bit unit

CRYPTO_PKA_L1
Address: Operational Base + offset (0x0894)

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12:0	RW	0x0000	pka_l1 PKA length 1, in bit unit

CRYPTO_PKA_L2
Address: Operational Base + offset (0x0898)

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12:0	RW	0x0000	pka_l2 PKA length 2, in bit unit

CRYPTO_PKA_L3
Address: Operational Base + offset (0x089c)

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12:0	RW	0x0000	pka_l3 PKA length 3, in bit unit

CRYPTO_PKA_L4
Address: Operational Base + offset (0x08a0)

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12:0	RW	0x0000	pka_l4 PKA length 4, in bit unit

CRYPTO_PKA_L5
Address: Operational Base + offset (0x08a4)

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12:0	RW	0x0000	pka_l5 PKA length 5, in bit unit

CRYPTO_PKA_L6
Address: Operational Base + offset (0x08a8)

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12:0	RW	0x0000	pka_l6 PKA length 6, in bit unit

CRYPTO_PKA_L7
Address: Operational Base + offset (0x08ac)

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12:0	RW	0x0000	pka_l7 PKA length 7, in bit unit

CRYPTO_PKA_PIPE_RDY
Address: Operational Base + offset (0x08b0)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x1	pka_pipe_rdy PKA pipe is ready for new opcode.

CRYPTO_PKA_DONE

Address: Operational Base + offset (0x08b4)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x1	pka_done PKA operation is completed and pipe is empty.

CRYPTO_PKA_MON_SELECT

Address: Operational Base + offset (0x08b8)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x0	pka_mon_select PKA monitor select which PKA fsm monitor is being output.

CRYPTO_PKA_DEBUG_REG_EN

Address: Operational Base + offset (0x08bc)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	pka_debug_reg_en Enable all the debug mechanism when set.

CRYPTO_DEBUG_CNT_ADDR

Address: Operational Base + offset (0x08c0)

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	R/W SC	0x00000	debug_cnt_addr The clock counter initial values. clock is disabled when counter expires. Triggered when pka_debug_en is set.

CRYPTO_DEBUG_EXT_ADDR

Address: Operational Base + offset (0x08c4)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	WO	0x0	debug_ext_addr Disable the debug Mechanism

CRYPTO_PKA_DEBUG_HALT

Address: Operational Base + offset (0x08c8)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	pka_debug_halt In debug mode: PKA is in halt state.

CRYPTO_PKA_MON_READ
Address: Operational Base + offset (0x08d0)

Bit	Attr	Reset Value	Description
31:0	RO	0x0000feef	pka_mon_read This is the PKA monitor bus register output.

CRYPTO_PKA_INT_ENA
Address: Operational Base + offset (0x08d4)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	pka_int_ena 1'b1: enable pka interrupt 1'b0: disable pka interrupt

CRYPTO_PKA_INT_ST
Address: Operational Base + offset (0x08d8)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	W1C	0x0	pka_int_st Indicate that PKA operation completes. After the bit is read, the application should write 1 to clear this bit for next time use.

CRYPTO_SRAM_ADDR
Address: Operational Base + offset (0x1000)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sram_addr Sram address starts from 0x1000 to 0x1fff. When RAM_CTL.ram_pka_rdy == 0, application could access sram. Otherwise, application can't.

9.4 Application Note

9.4.1 Clock &Reset

There are 4 clock domains in Crypto. The clock and reset signals are described in the following table .

Table 9-1Crypto Clock & Reset Description

Signal	Attr	Description
hclk	clock	AHB clock
aclk	clock	AXI master clock
clk_core	clock	Cipher work clock

Signal	Attr	Description
clk_pka	clock	PKA work clock
hresetn	reset	Asynchronously assert, synchronously de-assert to hclk,low active
aresetn	reset	Asynchronously assert, synchronously de-assert to aclk,low active
resetn_core	reset	Asynchronously assert, synchronously de-assert to clk_core,low active
resetn_pka	reset	Asynchronously assert, synchronously de-assert to clk_pka,low active

Each function need different clocks. The applications could gate the un-used clock to save power. Please see the following table for detail information.

Table 9-2Crypto Clock & Reset Description

Operation	HCLK	ACLK	CLK_CORE	CLK_PKA
AES	ON	ON	ON	OFF
DES/TDES	ON	ON	ON	OFF
HASH/HMAC	ON	ON	ON	OFF
PKA	ON	OFF	OFF	ON
TRNG	ON	OFF	OFF	OFF

Even when CLK_CORE is on, Crypto is doing some cipher job. And Crypto could still be able to automatically gate most parts of un-used blocks to save more power, if CRYPTO_CLK_CTL.auto_clkgate_en is set to '1' . The default value for this bit is also '1'. Application could do a soft reset to a certain clock domain. Please refer to "Chapter CRU" for more details.

9.4.2 Performance

Cipher performance is shown in the following table.

Table 9-3Crypto Performance Description

Algorithm	block size (Byte)	clk_core frequency (Mhz)	cycle	serial max throughput (MBps)	parallel max throughput (MBps)
DES	8	200	18	88	352
TDES	8	200	55	29	116
AES-128	16	200	12	266	1066
AES-192	16	200	14	228	914
AES-256	16	200	16	200	800
SHA-1	64	200	81	158	NA
MD5	64	200	65	196	NA
SHA-256/224	64	200	65	196	NA
SHA-512/384/ 512_224/ 512_256	128	200	81	316	NA

There are 2 column throughput rates in the table, 1 is serial mode, the other is parallel mode. In parallel mode, there are 4 engines working at the same time . So the speed is 4 times than serial mode. Parallel mode includes ECB/CTR/XTS both encryption and decryption mode, CFB/CBC/CTS only decryption mode. Other modes are serial. HASH doesn't have parallel mode.

For PKA, the cycles for each calculation are not certain. It depends on the parameters. Take RSA-2048 for example, it takes about 28M cycles to finish a calculation. PKA can run 300 Mhz. It means it can run over 10 times per second.

9.4.3 DMA

DMA supports Link List Item (LLI) DMA transaction.

- Each item contains 8 bytes, and start address should be 8 bytes align;
- We suggest that DATA start address is 8 bytes align;
- Total DATA length is byte align;

- Support segmenting HASH/HMAC DATA into multi sections. We suggest that each section DATA length is a multiple of 64 bytes, except this section is the last section.

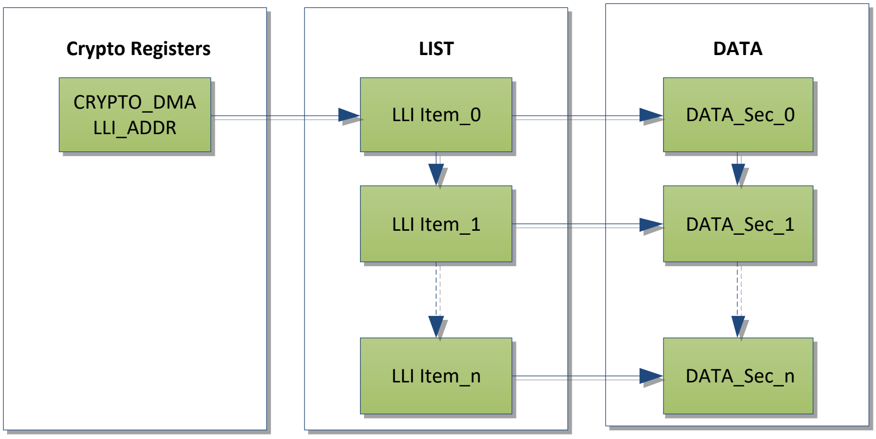


Fig. 9-2 LLI DMA Usage

As shown in the Figure above, Register CRYPTO_DMA_LLI_ADDR points to 1'st LLI item in external memory. Each LLI item contains DATA address, length, control information and next LLI item pointer, except the last LLI item. The last item doesn't have the next LLI Item pointer. After the last LLI item is finished, DMA will go to idle state.
LLI item definition is shown in the following table.

Table 9-4LLI Item Description

offset	Def	Description
0x00	SRC_ADDRESS[31:0]	source data start address
0x04	SRC_LENGTH[31:0]	source data length，in byte unit
0x08	DST_ADDRESS[31:0]	destination data start address
0x0c	DST_LENGTH[31:0]	destination data length，in byte unit
0x10	USER_DEFINE[31:0]	used in cipher block
0x14	reserve	reserve
0x18	DMA_CTRL[31:0]	used in DMA block
0x1c	next address[31:0]	next LLI item address. When DMA_CTRL.LAST = '1'，NEXT_ADDRESS is invalid.

DMA_CTRL: the definition is shown in the following table.

Table 9-5LLI Item dma_ctl Description

Bit	Def	Definition
[31:24]	ITEM_ID[7:0]	used to identify LLI items.
[23:16]	reserve	
[15:11]	reserve	
10	source_item_done enable	When source data fetch is completed, CRYPTO_DMA_INT_ST.source_item_done will assert if this bit is set.
9	reserved	reserved
8	list_done enable	When all LLI items transfer is completed, CRYPTO_DMA_INT_ST.list_done will assert if this bit is set.
[7:2]	reserved	reserved
1	PAUSE	indicate DMA will hold on after executes current item. DMA won't go on unless CRYPTO_DMA_CTL.restart is configured.

Bit	Def	Definition
0	LAST	indicate current item is the last one. After executes current item, DMA will return to IDLE state.

Table 9-6LLI Item user_define Description

Bit	Signal	Description
31:9	Reserved	Reserved
8	otpkey_sel	otpkey select. 1: select otpkey; 0: select register key
7	Privacy_sel	pkey select. 1: select pkey; 0: select key;
6:4	Chnl_num	channel number, from 0 to 7.
3	String_attr	indicate current item's attribution. 0: ADA; 1: PC(plaintext or ciphertext).
2	String_last	indicate current item is the string last item
1	String_start	indicate current item is the string first item
0	Cipher_start	indicate current item is the cipher first item

9.4.4 Multi-Channel Map

There are 8-channel configurations for AES or DES/TDES operation. For different key-size, the map is different. Please find the register map in the following table.

Table 9-7LLI Item user_define Description

Cipher sel	otpkey sel	privacy sel	chnl num	key	iv(tag/…)
AES-128/ DES	0	0	0	CH0_KEY0-3/ CH0_KEY0-1	CH0_IV0-3/ CH0_IV0-1
AES-128/ DES	0	0	n	CHn_KEY0-3/ CHn_KEY0-1	CHn_IV0-3/ CHn_IV0-1
AES-128/ DES	0	0	7	CH7_KEY0-3/ CH7_KEY0-1	CH7_IV0-3/ CH7_IV0-1
AES-128/ DES	0	1	0	CH0_PKEY0-3/ CH0_PKEY0-1	CH0_IV0-3/ CH0_IV0-1
AES-128/ DES	0	1	n	CHn_PKEY0-3/ CHn_PKEY0-1	CHn_IV0-3/ CHn_IV0-1
AES-128/ DES	0	1	7	CH7_PKEY0-3/ CH7_PKEY0-1	CH7_IV0-3/ CH7_IV0-1
AES-128/ DES	1	NA	0	OTP_KEY[255:128]	CH0_IV0-3/ CH0_IV0-1
AES-128/ DES	1	NA	n	OTP_KEY[127:0]	CHn_IV0-3/ CHn_IV0-1
AES-128/ DES	1	NA	7	OTP_KEY[127:0]	CH7_IV0-3/ CH7_IV0-1
AES-192/ TDES	0	0	0	CH0_KEY0-3, CH1_KEY0-1	CH0_IV0-3/ CH0_IV0-1
AES-192/ TDES	0	0	1	CH2_KEY0-3, CH3_KEY0-1	CH1_IV0-3/ CH1_IV0-1
AES-192/ TDES	0	0	2	CH4_KEY0-3, CH5_KEY0-1	CH2_IV0-3/ CH2_IV0-1
AES-192/ TDES	0	0	3	CH6_KEY0-3, CH7_KEY0-1	CH3_IV0-3/ CH3_IV0-1
AES-192/ TDES	0	1	0	CH0_PKEY0-3, CH1_PKEY0-1	CH0_IV0-3/ CH0_IV0-1
AES-192/ TDES	0	1	1	CH2_PKEY0-3, CH3_PKEY0-1	CH1_IV0-3/ CH1_IV0-1

Cipher sel	otpkey sel	privacy sel	chnl num	key	iv(tag/…)
AES-192/ TDES	0	1	2	CH4_PKEY0-3, CH5_PKEY0-1	CH2_IV0-3/ CH2_IV0-1
AES-192/ TDES	0	1	3	CH6_PKEY0-3, CH7_PKEY0-1	CH3_IV0-3/ CH3_IV0-1
AES-192/ TDES	0	NA	4-7	not supported	not supported
AES-192/ TDES	1	NA	0	OTP[255:64]	CH0_IV0-3/ CH0_IV0-1
AES-192/ TDES	1	NA	1-7	not supported	not supported
AES-256	0	0	0	CH0_KEY0-3, CH1_KEY0-3	CH0_IV0-3/ CH0_IV0-1
AES-256	0	0	1	CH2_KEY0-3, CH3_KEY0-3	CH1_IV0-3/ CH1_IV0-1
AES-256	0	0	2	CH4_KEY0-3, CH5_KEY0-3	CH2_IV0-3/ CH2_IV0-1
AES-256	0	0	3	CH6_KEY0-3, CH7_KEY0-3	CH3_IV0-3/ CH3_IV0-1
AES-256	0	1	0	CH0_PKEY0-3, CH1_PKEY0-3	CH0_IV0-3/ CH0_IV0-1
AES-256	0	1	1	CH2_PKEY0-3, CH3_PKEY0-3	CH1_IV0-3/ CH1_IV0-1
AES-256	0	1	2	CH4_PKEY0-3, CH5_PKEY0-3	CH2_IV0-3/ CH2_IV0-1
AES-256	0	1	3	CH6_PKEY0-3, CH7_PKEY0-3	CH3_IV0-3/ CH3_IV0-1
AES-256	0	NA	4-7	not supported	not supported
AES-256	1	NA	0	OTP[255:0]	CH0_IV0-3/ CH0_IV0-1
AES-256	1	NA	1-7	not supported	not supported

In AES-XTS mode, there are 2 keys, and only AES-128 and AES-256 mode are. Please refer to the following table for detail information.

Table 9-8LLI Item user_define Description

Cipher sel	otpkey sel	privacy sel	chnl num	key1	key2	tweak
AES-128	0	0	0	CH0_KEY0-3	CH4_KEY0-3	CH0_IV0-3
AES-128	0	0	1	CH1_KEY0-3	CH5_KEY0-3	CH1_IV0-3
AES-128	0	0	2	CH2_KEY0-3	CH6_KEY0-3	CH2_IV0-3
AES-128	0	0	3	CH3_KEY0-3	CH7_KEY0-3	CH3_IV0-3
AES-128	0	1	0	CH0_PKEY0-3	CH4_PKEY0-3	CH0_IV0-3
AES-128	0	1	1	CH1_PKEY0-3	CH5_PKEY0-3	CH1_IV0-3
AES-128	0	1	2	CH2_PKEY0-3	CH6_PKEY0-3	CH2_IV0-3
AES-128	0	1	3	CH3_PKEY0-3	CH7_PKEY0-3	CH3_IV0-3
AES-128	0	NA	4-7	not supported	not supported	not supported
AES-128	1	NA	NA	not supported	not supported	not supported
AES-256	0	0	0	CH0_KEY0-3, CH1_KEY0-3	CH4_KEY0-CH5_KEY3	CH0_IV0-3
AES-256	0	0	1	CH2_KEY0-3, CH3_KEY0-3	CH6_KEY0-CH7_KEY3	CH1_IV0-3
AES-256	0	1	0	CH0_PKEY0-3, CH1_PKEY0-3	CH4_PKEY0-CH5_PKEY3	CH0_IV0-3

Cipher sel	otpkey sel	privacy sel	chnl num	key1	key2	tweak
AES-256	0	1	1	CH2_PKEY0-3, CH3_PKEY0-3	CH6_PKEY0- CH7_PKEY3	CH1_IV0-3
AES-256	0	NA	2-7	not supported	not supported	not supported
AES-256	1	NA	NA	not supported	not supported	not supported

Note: The difference between CHn_KEY and CHn_PKEY is that: CHn_KEY could be read/write, CHn_PKEY could be write , but can't be read . The read value for CHn_PKEY is all '0'.

9.4.5 HASH Data Path

HASH and AES could run in parallel way. There are 2 paths lead to AES-HASH function. One is AES-HASH-RX mode, the other is AES-HASH-TX mode.

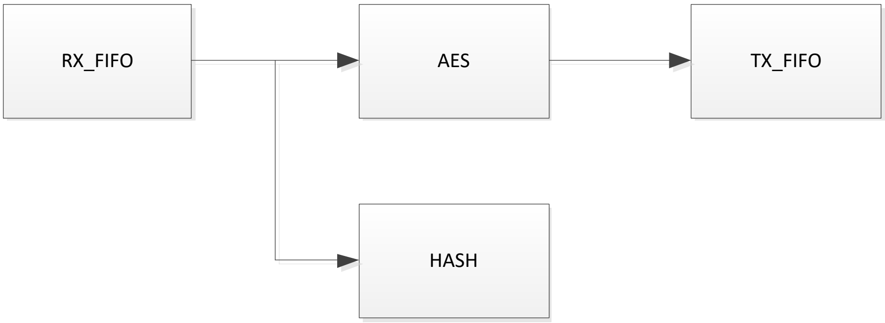


Fig. 9-3 AES-HASH-RX mode

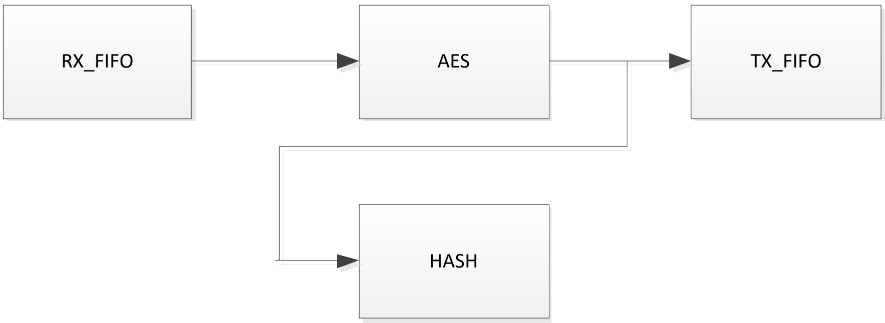


Fig. 9-4 AES-HASH-TX mode

As shown in the figures above, we could facilitate operations in some cases. For example, secure boot, we need both AES and HASH operations for the same blocks of data. The data HASH gets from RX_FIFO or TX_FIFO is byteswaped if the byteswap function is configured.

9.4.6 Program Steps

The application could succeed various crypto operations if they program properly.

- Program the LLI address to DMA_LLI_ADDR;
- Program KEY, IV, or other parameter if needed;
- Program BC_CTL or HASH_CTL for control information;
- Prepare LLI Item;
- Enable interrupt, or do nothing;

All these operations could be in any order.

- Program DMA_CTL.start to start the operation;

This step should be the last configuration step. After this register is configured, other registers should not be changed.

- Wait interrupt asserted, or just poll the DMA_INT_ST bits;
- Program DMA_INT_ST to clear interrupt status, and get the result.

The application could also use LLI.pause when the next LLI item is not ready. After the new item is prepared, the application could program DMA_CTL.restart to continue previous operation.