Table 13-47. YUV Input and Output Format Support (continued)

Format	Input Support	Output Support	Bits per pixel
YUV422 Semi-planar	YES	YES (only when Input is YUV422)	16

13.15.3 External Signals

There are no external signals pinned out for this module.

13.15.4 Initialization and Application Information

The Dewarp Engine cannot be used as a standalone module. It can only be used in conjunction with the Image Signal Processor (ISP) module.

13.16 Machine Learning Neural Processing Unit (NPU)

13.16.1 Overview

The Neural Processing Unit (NPU) provides hardware acceleration for AI/ML workloads and vision functions. The IP provides enhanced performance for real-time use cases with hardware support for the OpenVX API.

13.16.2 Block Diagram

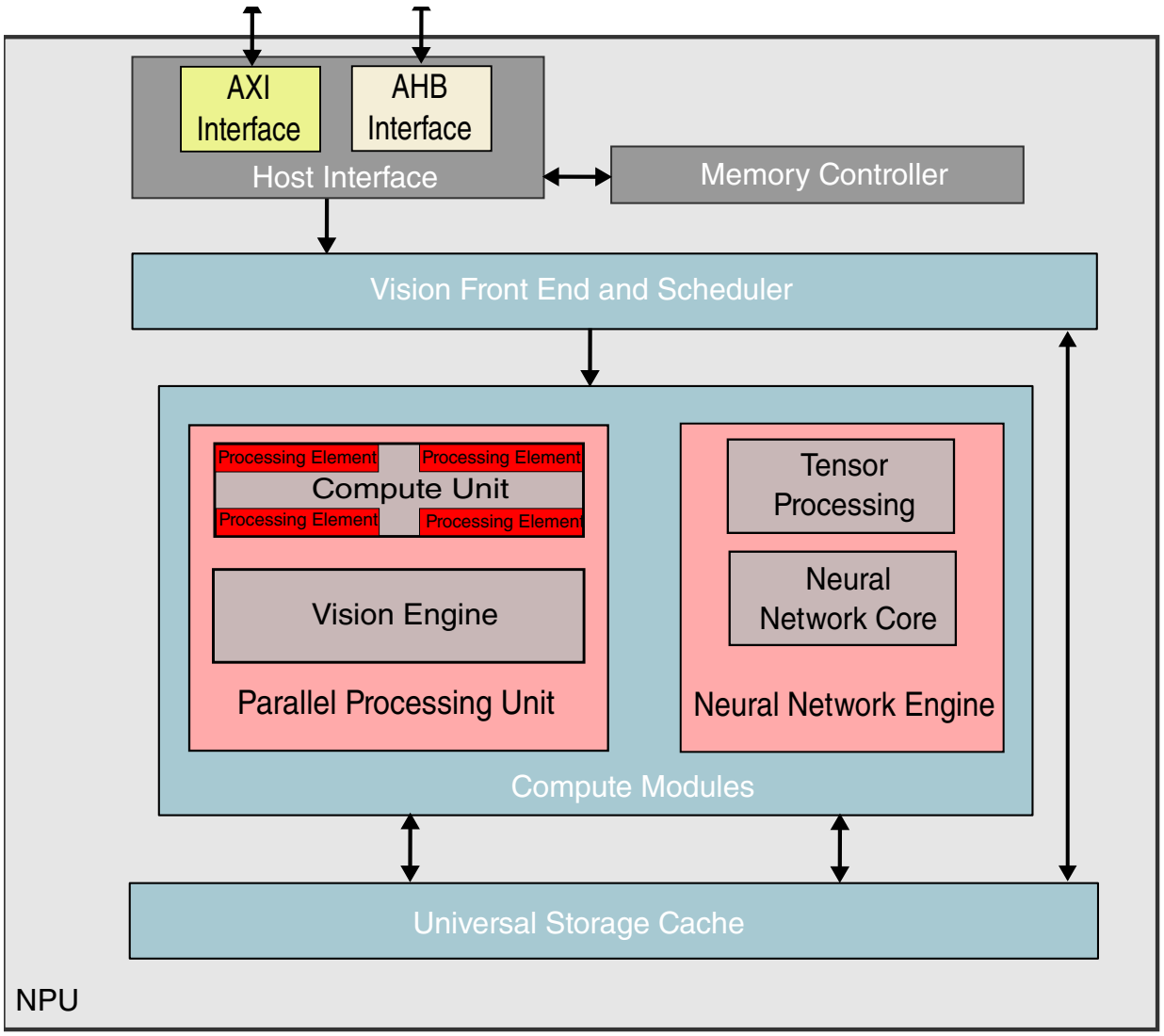


Figure 13-93. NPU High-level Block Diagram

13.16.2.1 Block Descriptions

The main functional blocks of the NPU core are described in the table below.

Table 13-48. NPU Functional Blocks

NPU Block	Description
Host Interface	Allows the NPU to communicate with external memory and the CPU through the AXI / AHB bus. In this block data crosses clock domain boundaries.

Table continues on the next page...

Table 13-48. NPU Functional Blocks (continued)

NPU Block	Description
Memory Controller	Internal memory management unit that controls the block-to-host memory request interface.
Vision Front End	Inserts high level primitives and commands into the vision pipeline.
Neural Network Core	Provides parallel convolution MAC for recognition functions using 8 or 16 bits integer.
Tensor Processing Fabric	Provides data preprocessing and supports compression and pruning for multi-dimensional array processing for Neural Nets.
Compute Unit	SIMD processor programmable execution unit that perform as a Compute Unit. The NPU block has 1 vec4 Parallel Processor Unit which also acts as 4 Processing Elements.
Vision Engine	Provides advanced image processing functions.
Universal Storage Cache	Cache shared between the Vision Front End and the Parallel Processing Unit.

13.16.3 Features

Key features of the NPU block include:

- OpenVX compliance, including extensions
- Convolutional Neural Network acceleration
- IEEE 32-bit floating-point pipeline in PPU shaders.
- Ultra-threaded parallel processing unit
- Low bandwidth at both high and low data rates
- Low CPU loading
- MMU functionality supported
- Performance Counters for DMA Profiling
- Data transfers between Neural Network Engines and the Parallel Processing Unit, with 256KB SRAM as local storage
- Neural Network Engine and Parallel Processing Unit synchronization with hardware semaphore

13.16.3.1 API Support and Architecture Features

The following table describes API support and architectural features of the NPU block.

Table 13-49. NPU Architecture Features

Feature	NPU Support
Primary APIs	OpenVX with Neural Network Extensions
Drivers	OpenVX

13.16.4 Functional Description

13.16.4.1 Parallel Processing Unit

The Parallel Processing Unit (PPU) features are noted below:

Table 13-50. Parallel Processing Unit Features

Feature	Supported
Parallel Processor Execution unit	1 compute unit, SIMD4, SFP32 Trans
FP denorm and rounding options	Denorms are flushed to zero. Supports round to nearest even and round to zero.
Maximum number of instructions	1 M (with I-cache)
Maximum number of streams	8
Maximum number of threads inflight per PPU core	256
Image Load/Store Instructions	Yes
Conditional branch support	GT, LT, EQ, GE, LE, NE, ISNAN, ISFINITY, ISINFINITY, ISNORMAL, AND, OR,XOR, NOT, ANYMSB, ALLMSB, SELMSB
Control flow instructions	Yes
Instruction rate	1-cycle throughput for all instructions
Standard derivatives	Yes
Integer pipeline	Support 8, 16, 32-bit integer operations
Universal Storage Cache	16 KB flexible allocation for both local shared memory and L1 cache

13.16.4.2 Neural Network Engine

The Neural Network Engine support features are in the table below:

Table 13-51. Neural Network Engine Features

Feature	Supported
MAC per cycle	1152
Applications	Classification, Detection, Segmentation
Network Topologies	Linear, MIMO, Fully Connected, Fully Convolutional
Network Size	2048 (Software driver limitation)
Inference Engine	TensorFlow Lite, Arm NN, ONNX, eIQ™ Inference with DeepViewRT
Configurable and Programmable	Yes
Network parameter deep compression	Yes
RNN Support	Yes

13.16.4.3 Tensor Processor

The Tensor Processor fabric support features are in the table below:

Table 13-52. Tensor Processor Features

Feature	Supported
Data format support	8/16-bit integer
Zero Weight Skipping	Yes, saving execution time
Intermediate Data Shuffling	Yes, saving CPU overhead
Local Memory for NN Weights andIntermediate Data Buffering	Yes
Local Buffer Configurations	0KB
Pooling	Max, average
Unpooling	Yes
Activation	ReLU, Leaky ReLU (LUT for other types)
Normalization	Yes
Region Proposal Support	Yes

13.16.4.4 Image Format Support

OpenVX image formats are supported. Packed formats are supported so that multiple pixels can be loaded in one shot.

13.16.4.4.1 OpenVX Support

Support for OpenVX includes the following capabilities:

Table 13-53. OpenVX Hardware Support

Feature	Supported
Enhanced Vision Instructions	EVIS 2 Instruction Set
Max kernel size	Up to 11x11, programmable
3x3 Image Filter modes	HorzMin3, HorzMed3, HorzMax3, VertMin3, VertMed3, VertMax3
Interpolation support	Lerp
Histogram support	IndexAdd, AtomicAdd
8-bit Dot Products	32x1, 16x2, 8x4, 4x8
16-bit Dot Products	16x1, 8x2 , 4x4, 2x8
Bit Manipulation modes	BitExtract
Packed Image Load/Store instructions	ImgLoad, ImgStore
Other EVIS instructions	AbsDiff, AccSq, MulShift, Clamp
Wide Uniform support	512-bit uniform reads per instruction

13.16.4.4.2 OpenVX Image Formats

The following OpenVX image formats do not require conversion.

Table 13-54. Vision Formats for OpenVX

OpenVX Formats	Description
VX_DF_IMAGE_VIRT	A virtual image of no defined type.
VX_DF_IMAGE_RGB	A single plane of 24-bit pixel as 3 interleaved 8-bit units of R then G then B data. This uses the BT709 full range by default.
VX_DF_IMAGE_RGBX	A single plane of 32-bit pixel as 4 interleaved 8-bit units of R then G then B data, then a don't care byte. This uses the BT709 full range by default.
VX_DF_IMAGE_NV12	A 2-plane YUV format of Luma (Y) and interleaved UV data at 4:2:0 sampling. This uses the BT709 full range by default.
VX_DF_IMAGE_NV21	A 2-lane YUV format of Luma (Y) and interleaved VU data at 4:2:0 sampling. This uses the BT709 full range by default.
VX_DF_IMAGE_UYVY	A single plane of 32-bit macro pixel of U0, Y0, V0, Y1 bytes. This uses the BT709 full range by default.
VX_DF_IMAGE_YUYV	A single plane of 32-bit macro pixel of Y0, U0, Y1, V0 bytes. This uses the BT709 full range by default.
VX_DF_IMAGE_IYUV	A 3 plane of 8-bit 4:2:0 sampled Y, U, V planes. This uses the BT709 full range by default.
VX_DF_IMAGE_YUV4	A 3 plane of 8 bit 4:4:4 sampled Y, U, V planes. This uses the BT709 full range by default.
VX_DF_IMAGE_U8	A single plane of unsigned 8-bit data. The range of data is not specified, as it may be extracted from a YUV or generated.
VX_DF_IMAGE_U16	A single plane of unsigned 16-bit data. The range of data is not specified, as it may be extracted from a YUV or generated.
VX_DF_IMAGE_S16	A single plane of signed 16-bit data. The range of data is not specified, as it may be extracted from a YUV or generated.
VX_DF_IMAGE_U32	A single plane of unsigned 32-bit data. The range of data is not specified, as it may be extracted from a YUV or generated.
VX_DF_IMAGE_S32	A single plane of unsigned 32-bit data. The range of data is not specified, as it may be extracted from a YUV or generated.

13.16.4.4.3 Vision Formats Requiring Conversion

Formats with eight (8) bits (or a multiple of 8 bits) per component typically do not require any conversion. Formats which do not have a component size which is a multiple of 8-bits usually require conversion. A Vision instruction can be used to do conversion if needed; conversion costs one instruction. Conversion will not be done if the application does not require it.

13.16.4.5 Power Modes

The NPU supports four (4) programmable, software-controlled power states that affect power consumption of the NPU. These states are: ON, IDLE, SUSPEND, and OFF.

Table 13-55. NPU Power States

Power State	Workload	Description
ON	Full	NPU runs at full clock speed. No clock gating.
IDLE	Minimal	NPU runs at 1/64 clock speed. No clock gating.
SUSPEND	None	NPU runs at 1/64 clock speed. AXI/core/SH are clock gated. FE stopped.
OFF	None	Same as SUSPEND, plus when this state is switched out then the NPU hardware is re-initialized. External power to the NPU can be disabled.

The kernel layer manages the power state transitions, and each state corresponds to an operating mode of the NPU:

1. Active mode
   - Power state is ON.
   - NPU is actively processing commands. One or more blocks are not in idle mode.
2. Idle mode
   - Power state is IDLE.
   - NPU is not processing any commands. All modules in the pipeline are in idle state.
   - The input clocks are on, but most of the NPU internal clocks are automatically gated off.
   - Frequency skipping is turned on with 1/64 pulse.
   - SoC reduces ACLK and HCLK frequency by 20x external to the NPU.
   - The NPU can start command processing instantly without any delay.
3. Standby mode
   - Power state is SUSPEND.
   - All input clocks to the NPU are shut off.
   - NPU clocks must be turned on and given enough time to reach idle state. Please refer to Resets section for information on the timing.
   - Once the NPU is in idle state, command processing can start.
4. Sleep mode
   - Power state is OFF.
   - The entire NPU is powered off through power gating.
   - The SoC must go through a power-on sequence in order to bring the NPU to idle state before starting command processing.

13.16.4.5.1 Low Power State Control

This NPU supports two low power state control schemes. One is the AXI Low Power state control scheme specified in the AXI spec. The other is the NPU Power Management scheme, controlled through a register which is accessible from the AHB.

13.16.4.5.1.1 AXI Low-Power Interface

The AXI Low-Power Interface is standardized in the AXI specification, and no interactions are required. However, it does not provide finer control over how the power management is configured. When exiting from the low-power state, the AQHiClock Control Register is initialized to the default value, enabling all functional blocks.

13.16.4.5.1.2 Power Management

There are two registers (AQHiClockControl and AQHiIdle) in the Host Interface (HI) module dedicated for power management. Software can control power management through these registers. Unlike the AXI Low-Power Interface, no automatic sequencing is involved in this scheme. The software needs to make sure that a block is in idle before shutting down the clock to the block. This scheme also offers a register interface to the clock scaling mechanism available in the graphics core.

13.16.4.6 Clock Domains

There are four independent clock domains in the NPU, and they are detailed in the table below.

Table 13-56. NPU Clocks

Clock	Signal name	Description
Core Clock	clkCore	Core clock is the main clock domain used by the NPU modules other than the PPU, NN, and USC.
PPU Clock	clkSh	The PPU clock domain contains the PPU, NN, and USC.
AHB Clock	hclk	The AHB Clock is the AHB Interface clock.
AXI Clock	aclk	The AXI Clock is the AXI Interface clock.

NOTE

The PPU Clock (clkSh) is the same source as the Core Clock (clkCore).

The communication between the different domains in NPU occurs mostly by way of asynchronous FIFOs. All clocks are asynchronous to each other. There is no communication between AHB and AXI clock domains.

The Core Clock in the NPU core can be scaled down dynamically without reprogramming its source. This scaling is controlled through the use of an internal register.

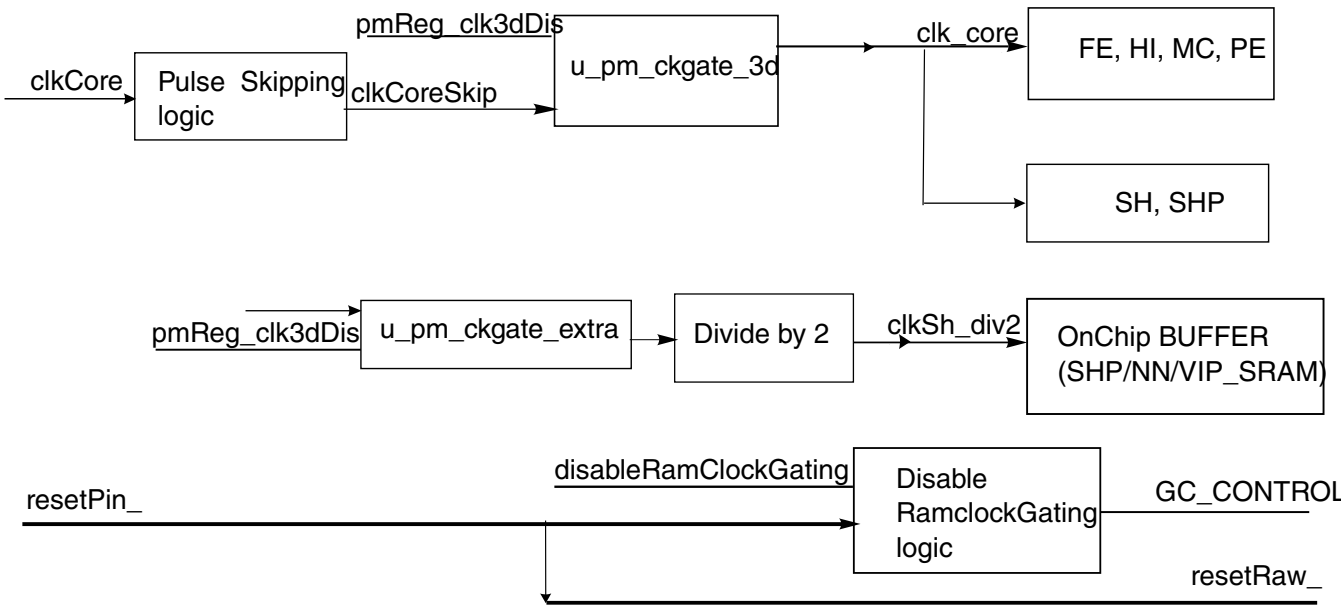


Figure 13-94. NPU Core Clock Logic Block Diagram

13.16.4.6.1 Clock Gating

Automatic localized clock gating is employed in the NPU to minimize dynamic power consumption.

13.16.4.6.1.1 Second Level Clock Gating

Block level clock gating is implemented in a majority of the blocks. If a block and the interface to the block are both idle, then the clock of that block will be gated automatically. This feature can be disabled by software.

13.16.4.6.2 Clock Disabling

The output of one of the clock gating blocks is for the Parallel Processing Unit (clkSh), and the output of the other clock gating block is for the rest of the NPU blocks (clkCore). The clock branches can be shut down independently through software control by setting the proper register bits. Please see Clock Control Register for more information.

13.16.4.6.3 Core Clock Frequency Scaling

A basic period of 64 clock cycles is chosen as the basis for clock skipping. Any number of clocks within that 64 clock period can be skipped. This yields a fine granularity of 1.6% (1/64).The scaling factor is a 7-bit value, which represents the number of clocks NOT to skip (the reset value is 64). Pulses will be skipped uniformly along the 64-cycle period to smooth out the power demand of the power supply.

Scaling is controlled by software through the use of control registers. The software determines the number of cycles not to skip within a 64-cycle period. The frequency scaling value is double-buffered as well, and it is updated at the beginning of a 64-cycle sequence. The following diagram shows an example of pulse skipping for the case of a 16 cycle period.(The actual period is 64 cycles, but 16 was chosen for ease of illustration).

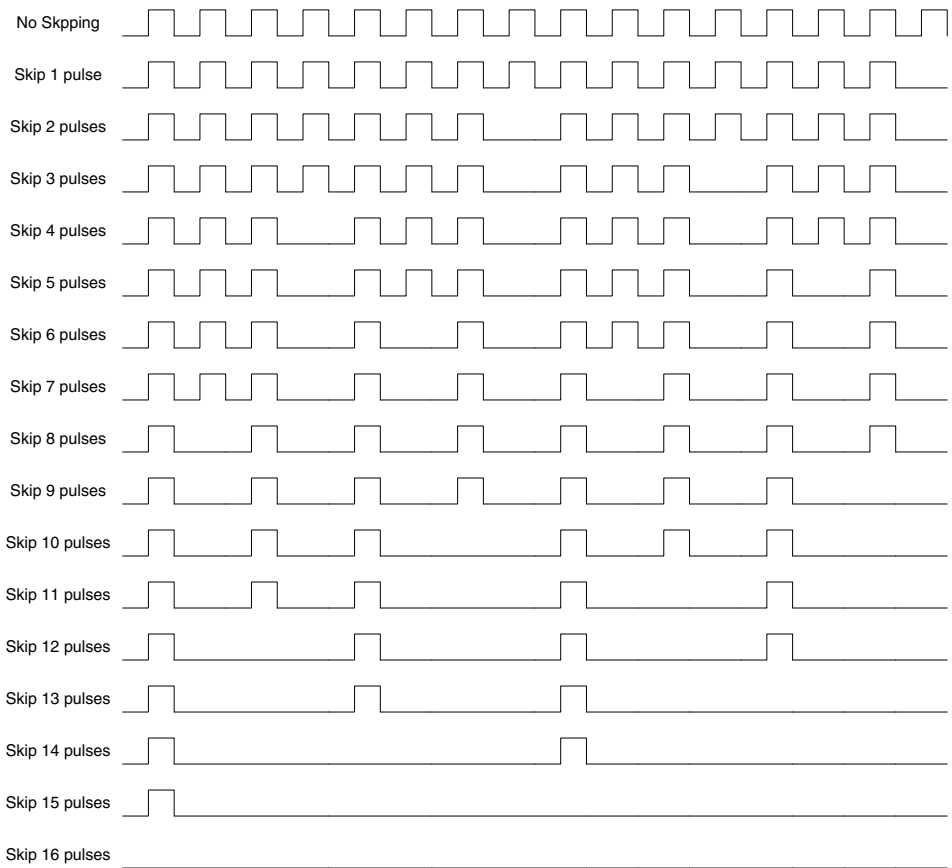


Figure 13-95. Pulse Skipping Example - Timing Diagram

Reducing the frequency of the core clock should not be a problem on the AXI side. The AXI interface is designed to handle any ratio of AXI clock vs. core clock. The AXI side can be stalled as well if the core clock side cannot accept the return data fast enough.

13.16.4.6.4 Clock Control Register

All clock control registers reside in the host clock domain.

Table 13-57. AQHiClockControl Register - Select Clock Control Bits

Bitfield	Access	Name	Description
0	R/W	CLK3D_DIS	Software core clock disable signal for core (clkCore) and 3D (clkSh) clocks. When set to “1”, these clocks are frozen. Reset value = 1’b0
1	R/W	CLK2D_DIS	Software core clock disable signal. For this core both bits CLK3D_DIS and CLK2D_DIS should be controlled by software. The AXI interface clock is the only block not stalled at that point. Reset value = 1’b0
8:2	R/W	FSCALE_VAL	Frequency scaling value. Reset value = 7’d64 (no pulse skipping)
9	R/W	FSCALE_CMD_LOAD	Frequency scaling load command. When writing a “1” to this bit, it updates the frequency scale factor with the value FSCALE_VAL[6:0]. The bit must be set back to “0” after that. Reset value = 1’b0

13.16.4.7 Resets

The NPU supports asynchronous resets that must be asserted for a minimum of of 32 core clock cycles (slowest clock). After deassertion, 128 core clock cycles (slowest clock) is required before starting any activity on AHB. During the 32 cycle assertion + 128 cycle wait, the core clock supplied into the NPU may be kept running. The clock may be gated before assertion, but because it is an asynchronous reset, it is not required.

13.16.4.8 Interrupts

The IP can send an interrupt signal to the host processor. The signal name is xaq2_intr, and it remains asserted until the host processor clears the interrupt by reading the interrupt acknowledge register (AQIntrAcknowledge). The interrupt is level triggered. The SOC should not use edge to detect the interrupt, as using edge to detect the GPU/NPU interrupt will miss some back to back interrupts. Each bit of AQIntrAcknowledge represents one of the 32 possible events that the NPU can signal to the host processor. By setting or clearing the bits of the interrupt enable register (AQIntrEnbl) the programmer can control which of those events will generate an interrupt. The bits of these registers have no predefined meaning, except for bit 31 which is used for AXI_BUS_ERROR. Instead, all of them are used by and controlled through the NPU driver. The NPU driver puts events with different event IDs (0~30) in the command buffer, and the NPU will generate an interrupt with the corresponding interrupt bit as set by the event ID in AQIntrAcknowledge. The NPU kernel driver and its interrupt

service routine (ISR) will process the interrupt accordingly to synchronize the driver/CPU with NPU. The meaning of the NPU interrupt bits are entirely private to the NPU driver. The event IDs (0~30) are arbitrarily used by the NPU driver thus the interrupt bits in AQIntrAcknowledge are also arbitrary. Chip designers need only to ensure that the xaq2_intr signal is connected to their interrupt controller. Since AQIntrEnbl is set to 0 at reset, the xaq2_intr signal will never be asserted during the chip validation phase.

13.16.5 Initialization

13.16.5.1 Power Off Sequence

The Power Off sequence for NPU is as follows:

- Chip software issues a request to go into "OFF" state or the NPU internal timer triggers a transition into the "OFF" state.
- NPU software then executes all the required preliminary steps to put the block into "OFF" state.
- NPU software issues a callback to gate the NPU clock and NPU power.
- Chip software initiates a power down sequence to the chip controller HW.
- Chip hardware asserts an isolation signal to turn on isolation from the NPU.
- Chip hardware asserts a power gate signal to turn off power to the NPU.
- Upon return of the hardware acknowledgement, chip software returns call-back status to NPU software.
- Upon receipt of a successful callback, NPU software puts the block into the "OFF" state.

13.16.5.2 Power On Sequence

The Power On sequence for NPU is as follows:

- Chip software requests a transition from the "OFF" state to the "ON" power state.
- NPU software issues a call back to enable clock and power for the NPU.
- Chip software initiates a power up sequence to the chip controller hardware.
- Chip hardware de-asserts the power gate signal to turn on power to the NPU block.
- Upon hardware acknowledgement, the chip Hardware turns on the clocks to the NPU block.
- Chip software performs an AHB domain reset.
- Chip Hardware de-asserts the isolation signal to turn off isolation from the NPU block.
- Chip software returns call-back status to NPU software.

- Upon a successful return, NPU Software performs an AHB write that will soft reset the core and AXI domains.
- NPU software enables FE and waits for commit

13.16.6 Memory Map and register definition

This section includes the NPU module memory map and detailed descriptions of key accessible registers.

The following are four modules that can be accessed via the AHB bus:

- Host Interface (HI)
- Power Management (PM)
- MMU Secure Zone (MMU)
- Memory Controller (MC)

The following table describes the registers, their corresponding AHB modules and addresses.

Table 13-58. Register Addresses and corresponding AHB Module

AHB Module	AHB Byte Offset	GPU DWord Offset
HI	0x000-0x0A8	0x0000-0x002A
PM	0x100-0x010C	0x0040-0x0043
TZ	0x388-0x3AC	0x00E2-0x00EA
MC	0x414-0x42C	0x0105-0x010B

Host Interface: The Host Interface is a bridge between the Memory Controller and the AXI interface. It also acts as a bridge between the core and the AHB bus. The Host Interface's register set contain the control settings for the clocks, AXI interface and interrupts. It also contains the identification registers and some counters that are used for debug. Users can access the Host Interface registers via the AHB bus.

Power Management Registers: The Power Management register set controls clock gating within the core. The graphics core allows the user to control the clock gating of each internal module independently of the other modules. Users can access the registers via the AHB Bus.

MMU Secure Zone Registers: These registers provide hardware mechanisms and base address information for access control.

Memory Controller Registers: All memory accesses that go to or from the AXI bus pass through the Memory Controller. It connects the Host Interface to the rest of the core. The Memory Controller's register set contains the controls for the memory interface, the

settings for the virtual memory page table, and the values of the offset registers. It also contains a variety of debug registers that are set by other blocks within the core. Users can access the Memory Controller's registers via the AHB bus.

NOTE

NPU functionality can be supported using SW interface.

13.16.6.1 NPU register descriptions

13.16.6.1.1 NPU memory map

NPU base address: 3850_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Clock Control Register (AQHiClockControl)	32	RW	0000_0900h
4h	Idle Status Register (AQHiIdle)	32	RO	7FFF_FFFFh
Ch	AXI Status Register (AQAxiStatus)	32	RO	0000_0000h
10h	Interrupt Acknowledge Register (AQIntrAcknowledge)	32	RO	0000_0000h
14h	Interrupt Enable Register (AQIntrEnbl)	32	RW	0000_0000h
78h	Total Cycles Register (gcTotalCycles)	32	RW	0000_7000h
100h	Module Power Level Control Register (gcModulePowerControls)	32	RW	0014_0020h
10Ch	Pulse eater Control Register (gcPulseEater)	32	RW	0159_0880h
388h	MMU Control Register (gcregMMUAHBControl)	32	RWONCE	0000_0000h
38Ch	MMU Table Array Base Lower 32-bit Address Register (gcregMMUAHBTableArrayBaseAddressLow)	32	RW	0000_0000h
390h	MMU Table Array Base Higher 32-bit Address Register (gcregMMUAHBTableArrayBaseAddressHigh)	32	RW	0000_0000h
394h	MMU Table Array Size Control Register (gcregMMUAHBTableArraySize)	32	RW	0000_FFFFh
398h	MMU NonSecure Address Register (gcregMMUAHBSafeNonSecureAddress)	32	RW	0000_0000h
39Ch	MMU Secure Address Register (gcregMMUAHBSafeSecureAddress)	32	RW	0000_0000h
3A4h	Command Buffer Control Register (gcregCmdBufferAHBCtrl)	32	WO	0000_0000h
3A8h	MMU Host Interface Control Register (gcregHiAHBControl)	32	RW	0000_0000h
3ACh	MMU AXI Configuration Register (gcregAxiAHBConfig)	32	RW	0022_2200h
414h	Memory Debug Register (AQMemoryDebug)	32	RW	3C00_0000h
42Ch	Register Timing Control Register (AQRegisterTimingControl)	32	RW	0003_0000h
654h	Command Buffer Base Address Register (AQCmdBufferAddr)	32	WO	0000_0000h
664h	Command Decoder Address Register (AQFEDebugCurCmdAdr)	32	RO	0000_0000h

13.16.6.1.2 Clock Control Register (AQHiClockControl)

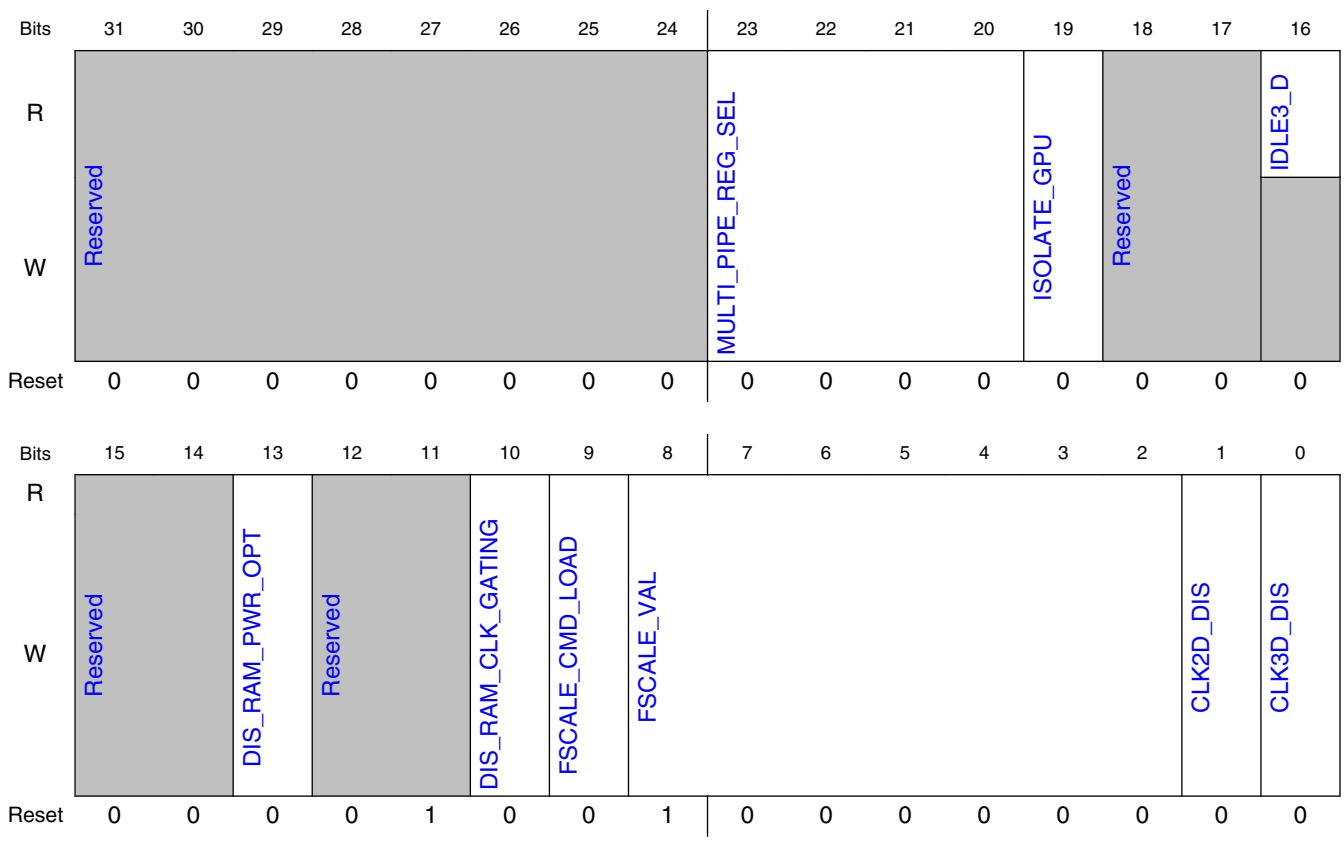
13.16.6.1.2.1 Offset

Register	Offset
AQHiClockControl	0h

13.16.6.1.2.2 Function

This register is used for clock control

13.16.6.1.2.3 Diagram



13.16.6.1.2.4 Fields

Field	Function
31-24	Reserved

Table continues on the next page...

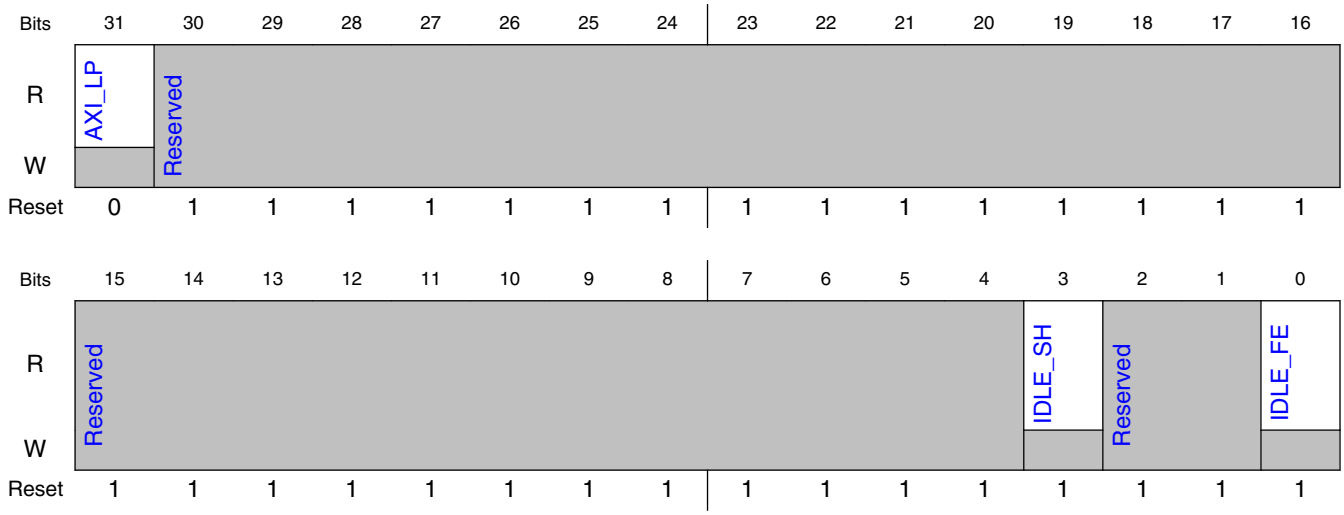
Field	Function
—	
23-20 MULTI_PIPE_REG_SEL	Multiple Pipe Register Select This bit is determines which HI/MC to use while reading registers.
19 ISOLATE_GPU	Isolate GPU This bit is used for power on/off, isolation only for multi-core GPUs. Please refer to Initialization section for information on its usage.
18-17 —	Reserved
16 IDLE3_D	3D pipe is idle
15-14 —	Reserved
13 DIS_RAM_PWR_OPT	Disable RAM power optimization
12-11 —	Reserved
10 DIS_RAM_CLK_GATING	Disable clock gating for RAMs
9 FSCALE_CMD_LOAD	Core clock frequency scale value enable If this bit is set and FSCALE_VAL=0 (an invalid combination), the HREADYOUT output signal will get stuck to 0. 1b - The frequency scale factor is updated with the value FSCALE_VAL[6:0]. The bit sets back to 0 after that.
8-2 FSCALE_VAL	Core clock frequency scale value
1 CLK2D_DIS	Disable 2D clock This bit is connected to the software clock disable signal. For this core, both bits CLK3D_DIS and CLK2D_DIS should be controlled by software. The AXI interface clock is the only block not stalled at that point.
0 CLK3D_DIS	Disable 3D clock Software core clock disable signal for 3D modules (clk_3d) clock. 1b - The clock is frozen

13.16.6.1.3 Idle Status Register (AQHildle)

13.16.6.1.3.1 Offset

Register	Offset
AQHiIdle	4h

13.16.6.1.3.2 Diagram



13.16.6.1.3.3 Fields

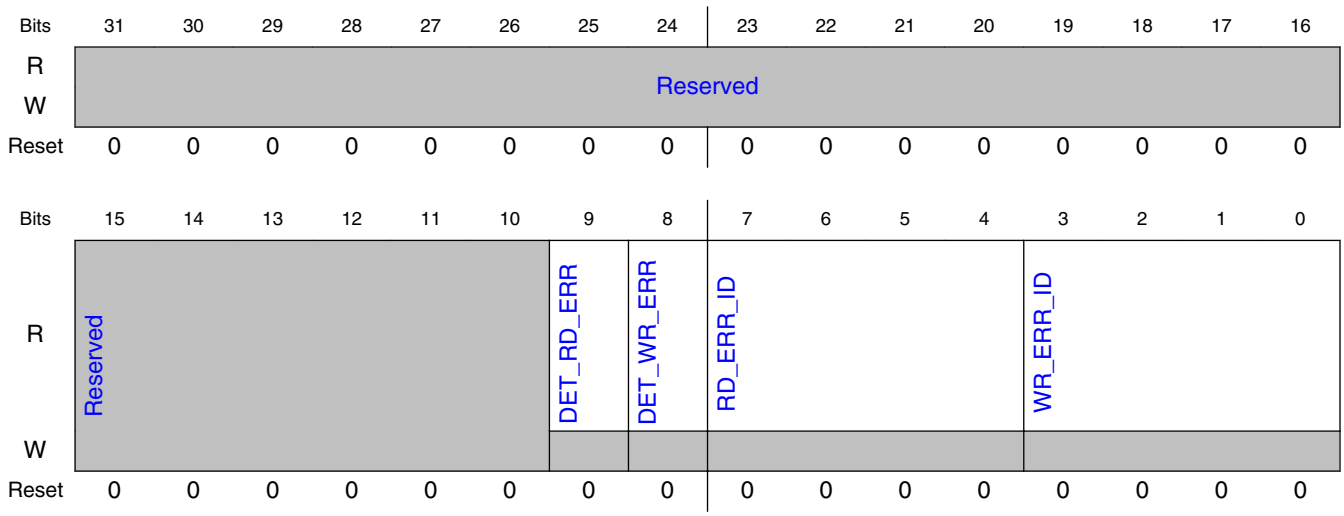
Field	Function
31 AXI_LP	AXI is in low power mode
30-4 —	Reserved
3 IDLE_SH	SH is idle
2-1 —	Reserved
0 IDLE_FE	FE is idle

13.16.6.1.4 AXI Status Register (AQAxiStatus)

13.16.6.1.4.1 Offset

Register	Offset
AQAxiStatus	Ch

13.16.6.1.4.2 Diagram



13.16.6.1.4.3 Fields

Field	Function
31-10 —	Reserved
9 DET_RD_ERR	Detect Read Error 1b - Detect read error
8 DET_WR_ERR	Detect Write Error 1b - Detect write error
7-4 RD_ERR_ID	Read Error ID
3-0 WR_ERR_ID	Write Error ID

13.16.6.1.5 Interrupt Acknowledge Register (AQIntrAcknowledge)

13.16.6.1.5.1 Offset

Register	Offset
AQIntrAcknowledge	10h

13.16.6.1.5.2 Function

Each bit represents a corresponding event being triggered. Reading from this register clears the outstanding interrupt.

13.16.6.1.5.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	INTR_VEC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	INTR_VEC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.16.6.1.5.4 Fields

Field	Function
31-0 INTR_VEC	Interrupt Vector For each interrupt event, 0=Clear, 1=InterruptActive Bit 31 is AXI_BUS_ERROR, 0 = No Error

13.16.6.1.6 Interrupt Enable Register (AQIntrEnbl)

13.16.6.1.6.1 Offset

Register	Offset
AQIntrEnbl	14h

13.16.6.1.6.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	INTR_ENBL_VEC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	INTR_ENBL_VEC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.16.6.1.6.3 Fields

Field	Function
31-0 INTR_ENBL_VEC	Interrupt Vector Enable In this register, each bit enables a corresponding event. 0: Disable interrupt 1: Enable Interrupt

13.16.6.1.7 Total Cycles Register (gcTotalCycles)

13.16.6.1.7.1 Offset

Register	Offset
gcTotalCycles	78h

13.16.6.1.7.2 Function

This register is a free running counter. It can be reset by writing 0 to it.

13.16.6.1.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CYCLES															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CYCLES															
Reset	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0

13.16.6.1.7.4 Fields

Field	Function
31-0 CYCLES	Cycles

13.16.6.1.8 Module Power Level Control Register (gcModulePowerControls)

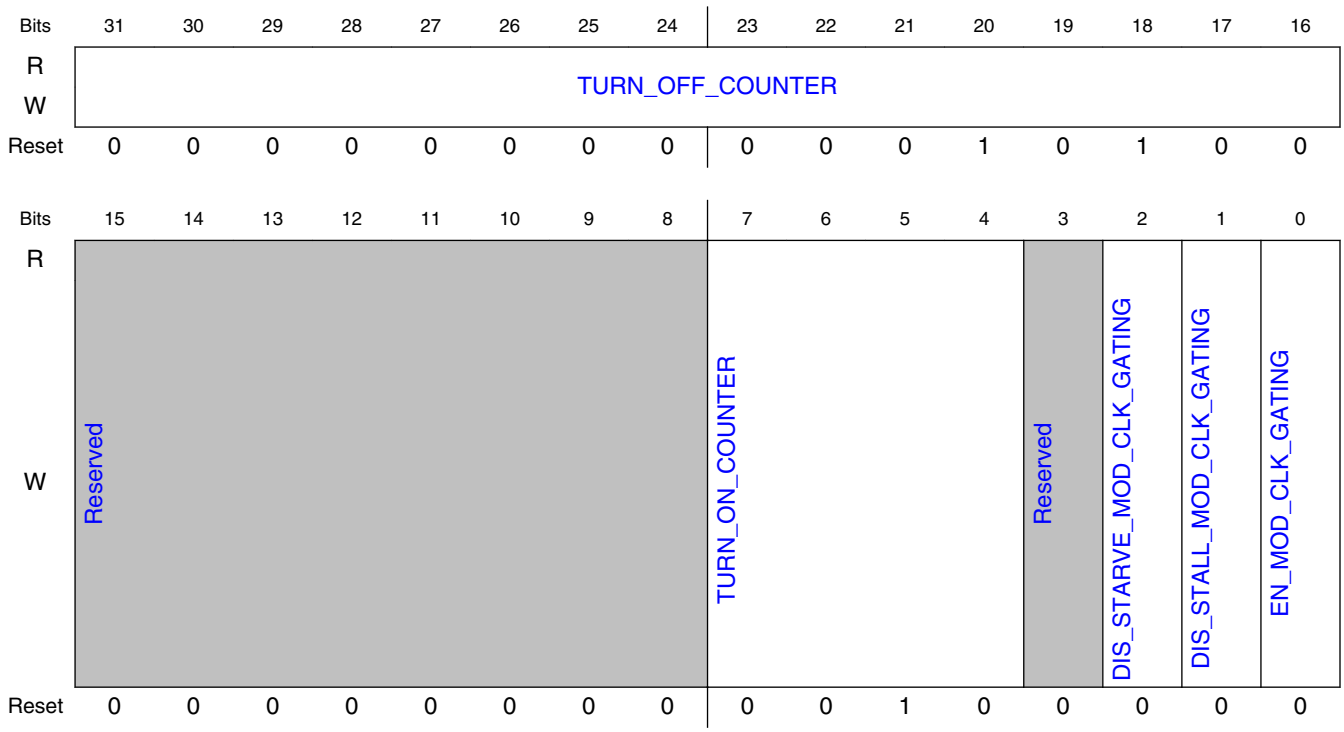
13.16.6.1.8.1 Offset

Register	Offset
gcModulePowerControls	100h

13.16.6.1.8.2 Function

This is a control register for module level power controls

13.16.6.1.8.3 Diagram



13.16.6.1.8.4 Fields

Field	Function
31-16 TURN_OFF_COUNTER	Counter value for clock gating the module if the module is idle for this amount of clock cycles
15-8 —	Reserved
7-4 TURN_ON_COUNTER	Number of clock cycles to wait after turning on the clock
3 —	Reserved
2 DIS_STARVE_MOD_CLK_GATING	Disables module level clock gating for starve/idle condition
1 DIS_STALL_MOD_CLK_GATING	Disables module level clock gating for stall condition
0	Enables module level clock gating

Field	Function
EN_MOD_CLK_GATING	

13.16.6.1.9 Pulse eater Control Register (gcPulseEater)

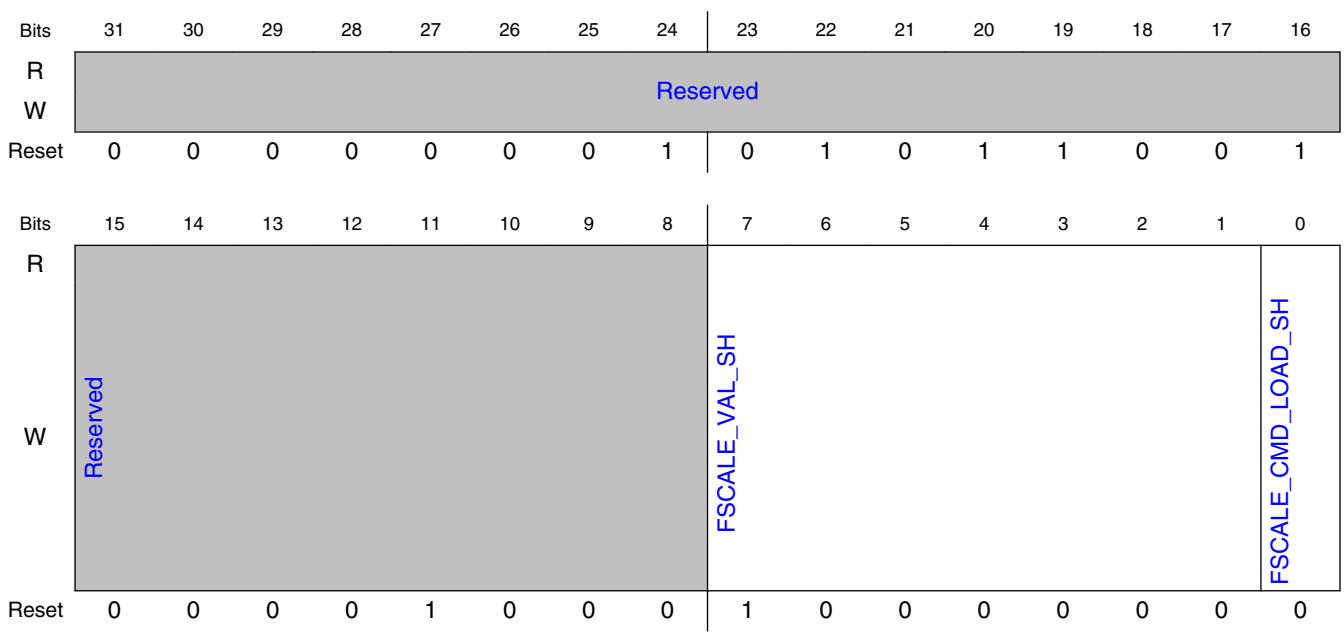
13.16.6.1.9.1 Offset

Register	Offset
gcPulseEater	10Ch

13.16.6.1.9.2 Function

This register shows the Pulse eater parameters

13.16.6.1.9.3 Diagram



13.16.6.1.9.4 Fields

Field	Function
31-8 —	Reserved

Table continues on the next page...

Field	Function
7-1 FSCALE_VAL_SH	Fscale value for shader
0 FSCALE_CMD_LOAD_SH	Fscale_cmd_load for shader

13.16.6.1.10 MMU Control Register (gcregMMUAHBControl)

13.16.6.1.10.1 Offset

Register	Offset
gcregMMUAHBControl	388h

13.16.6.1.10.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															MMU
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.16.6.1.10.3 Fields

Field	Function
31-1 —	Reserved
0 MMU	Enable the MMU For security reasons, once the MMU is enabled it cannot be disabled anymore 0b - Disable 1b - Enable

13.16.6.1.11 MMU Table Array Base Lower 32-bit Address Register (gcregMMUAHBTableArrayBaseAddressLow)

13.16.6.1.11.1 Offset

Register	Offset
gcregMMUAHBTableArrayBaseAddressLow	38Ch

13.16.6.1.11.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.16.6.1.11.3 Fields

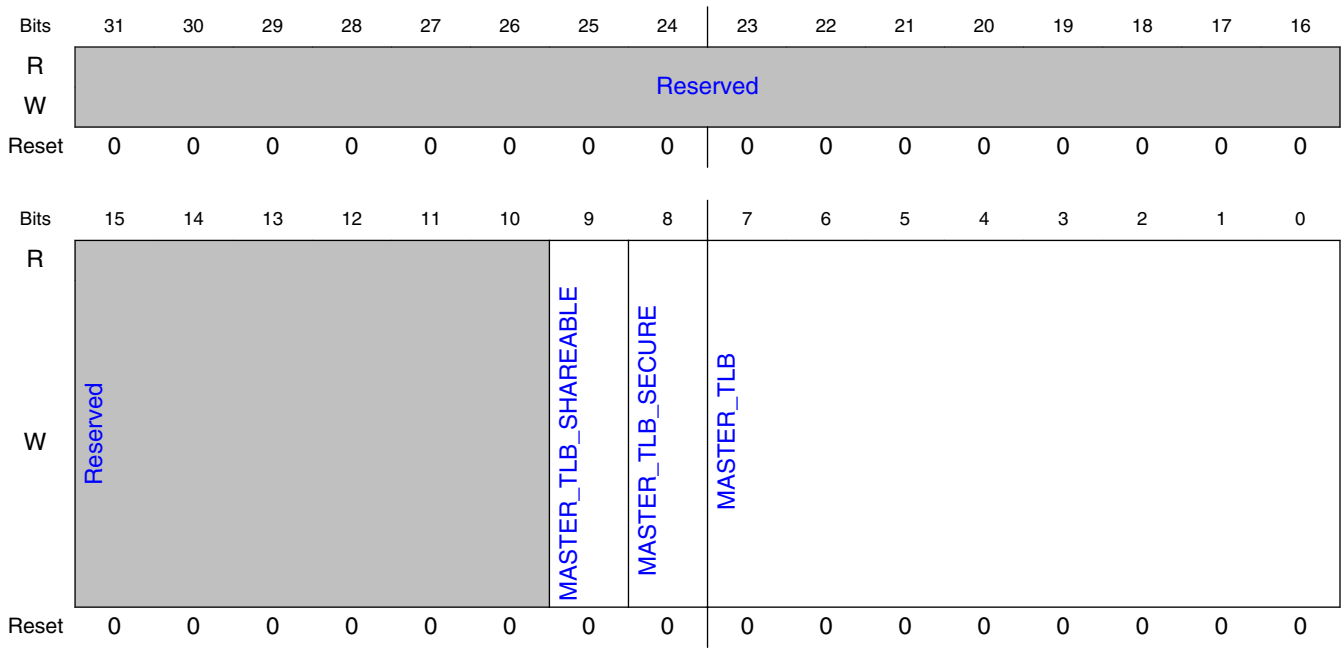
Field	Function
31-0 ADDRESS	Address

13.16.6.1.12 MMU Table Array Base Higher 32-bit Address Register (gcregMMUAHBTableArrayBaseAddressHigh)

13.16.6.1.12.1 Offset

Register	Offset
gcregMMUAHBTableArrayBaseAddressHigh	390h

13.16.6.1.12.2 Diagram



13.16.6.1.12.3 Fields

Field	Function
31-10 —	Reserved
9 MASTER_TLB_SHAREABLE	Bit that defines whether the master TLB address is shareable or not
8 MASTER_TLB_SECURE	Bit that defines whether the master TLB address is secure or not
7-0 MASTER_TLB	Upper 8-bits of the master TLB address to form a true 40-bit address

13.16.6.1.13 MMU Table Array Size Control Register (gcregMMUAHBTableArraySize)

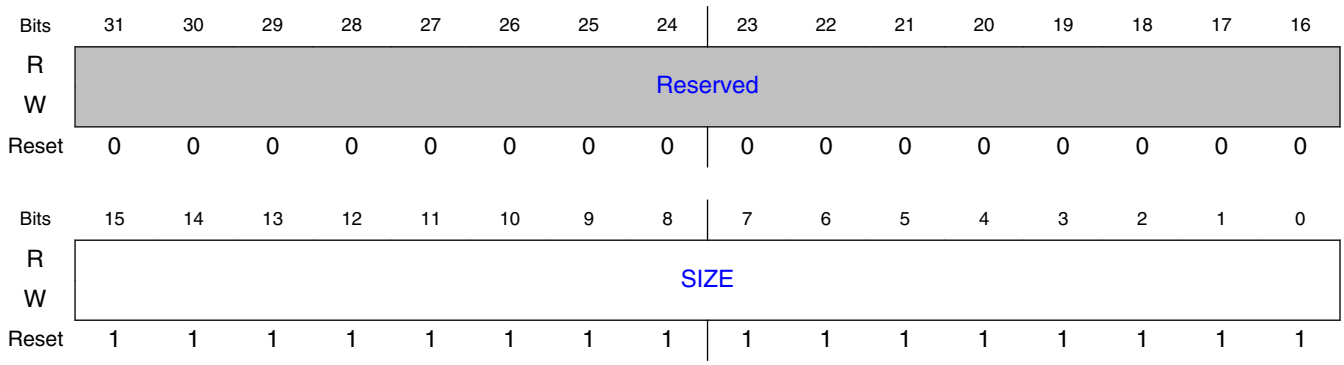
13.16.6.1.13.1 Offset

Register	Offset
gcregMMUAHBTableArraySize	394h

13.16.6.1.13.2 Function

16 bit MMUTableArraySize

13.16.6.1.13.3 Diagram



13.16.6.1.13.4 Fields

Field	Function
31-16 —	Reserved
15-0 SIZE	Size

13.16.6.1.14 MMU NonSecure Address Register (gcregMMUAHBSafeNonSecureAddress)

13.16.6.1.14.1 Offset

Register	Offset
gcregMMUAHBSafeNonSecureAddress	398h

13.16.6.1.14.2 Function

A 64-byte address that will acts as a 'safe' zone. Any address that would cause an exception is routed to this safe zone. Reads will happen and writes will go to this address, but with a write-enable of 0.

This register can only be programmed once after a reset - any attempt to write to this register after the initial write-after-reset will be ignored. This is in NonSecure memory.

13.16.6.1.14.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.16.6.1.14.4 Fields

Field	Function
31-0 ADDRESS	Address

13.16.6.1.15 MMU Secure Address Register (gcregMMUAHBSafeSecureAddress)

13.16.6.1.15.1 Offset

Register	Offset
gcregMMUAHBSafeSecureAddress	39Ch

13.16.6.1.15.2 Function

A 64-byte address that will acts as a 'safe' zone. Any address that would cause an exception is routed to this safe zone. Reads will happen and writes will go to this address, but with a write-enable of 0.

This register can only be programmed once after a reset - any attempt to write to this register after the initial write-after-reset will be ignored. This is in Secure memory.

13.16.6.1.15.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.16.6.1.15.4 Fields

Field	Function
31-0 ADDRESS	Address

13.16.6.1.16 Command Buffer Control Register (gcregCmdBufferAHBCtrl)

13.16.6.1.16.1 Offset

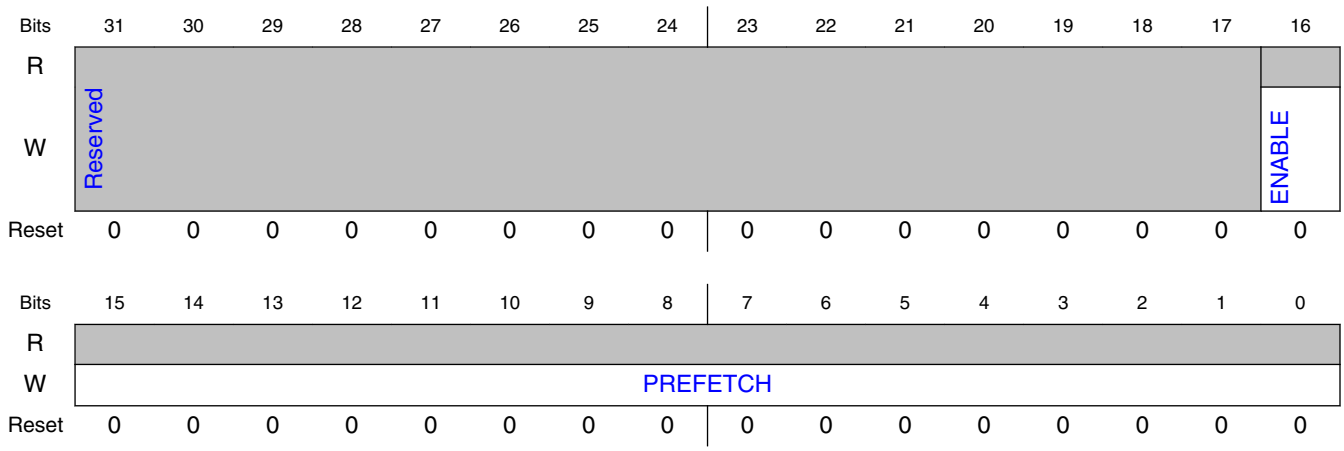
Register	Offset
gcregCmdBufferAHBCtrl	3A4h

13.16.6.1.16.2 Function

NOTE

Since this is a write only register, it has no reset value

13.16.6.1.16.3 Diagram



13.16.6.1.16.4 Fields

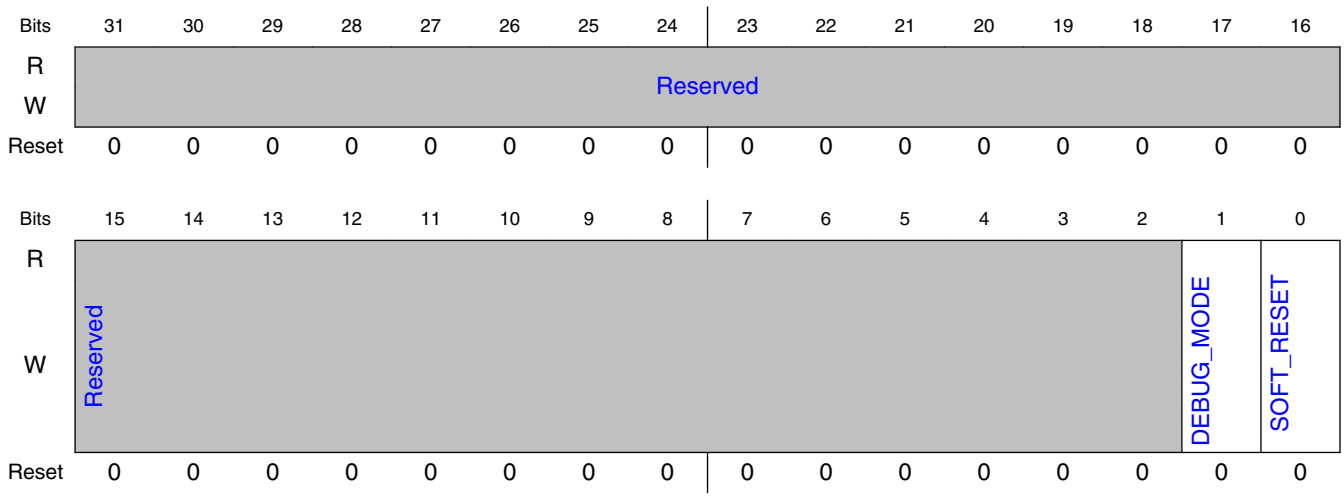
Field	Function
31-17 —	Reserved
16 ENABLE	Enable the command parser 0b - Disable 1b - Enable
15-0 PREFETCH	Prefetch Number of 64-bit words to fetch from the command buffer

13.16.6.1.17 MMU Host Interface Control Register (gcregHiAHBControl)

13.16.6.1.17.1 Offset

Register	Offset
gcregHiAHBControl	3A8h

13.16.6.1.17.2 Diagram



13.16.6.1.17.3 Fields

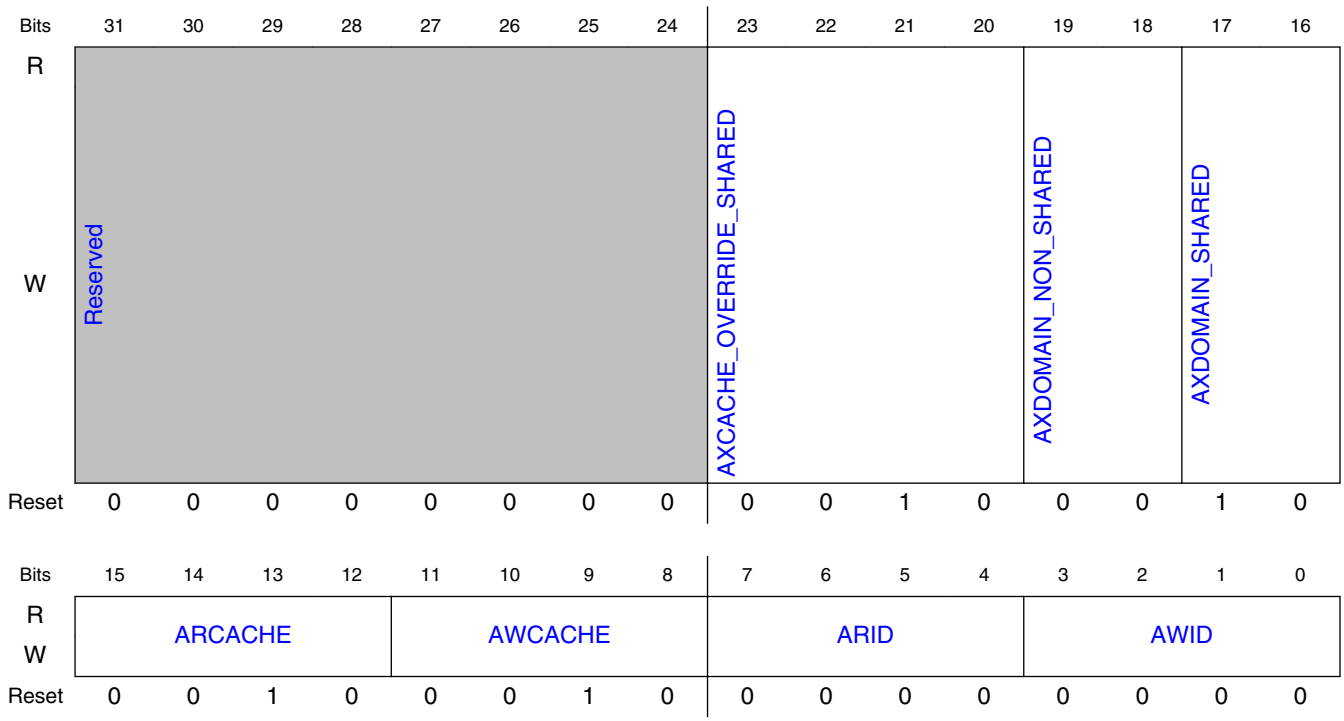
Field	Function
31-2 —	Reserved
1 DEBUG_MODE	Debug Mode Enable debug mode if disabled debug registers return 0xFFFF FFFF 0b - Disable 1b - Enable
0 SOFT_RESET	Soft Reset Soft resets the IP 0b - Disable 1b - Enable

13.16.6.1.18 MMU AXI Configuration Register (gcregAxiAHBConfig)

13.16.6.1.18.1 Offset

Register	Offset
gcregAxiAHBConfig	3ACh

13.16.6.1.18.2 Diagram



13.16.6.1.18.3 Fields

Field	Function
31-24 —	Reserved
23-20 AXCACHE_OVERRIDE_SHARED	Ax Cache value Configure AxCACHE walue for shareable request
19-18 AXDOMAIN_NON_SHARED	Ax Domain value Configure AxDOMAIN walue for non-shareable request
17-16 AXDOMAIN_SHARED	Ax Domain value Configure AxDOMAIN walue for shareable request
15-12 ARCACHE	AR Cache value Set ARCACHE[3:0] value
11-8 AWCACHE	AW Cache value Set AWCACHE[3:0] value
7-4 ARID	

Table continues on the next page...

Field	Function
3-0 AWID	

13.16.6.1.19 Memory Debug Register (AQMemoryDebug)

13.16.6.1.19.1 Offset

Register	Offset
AQMemoryDebug	414h

13.16.6.1.19.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								MAX_OUTSTANDING_READS							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.16.6.1.19.3 Fields

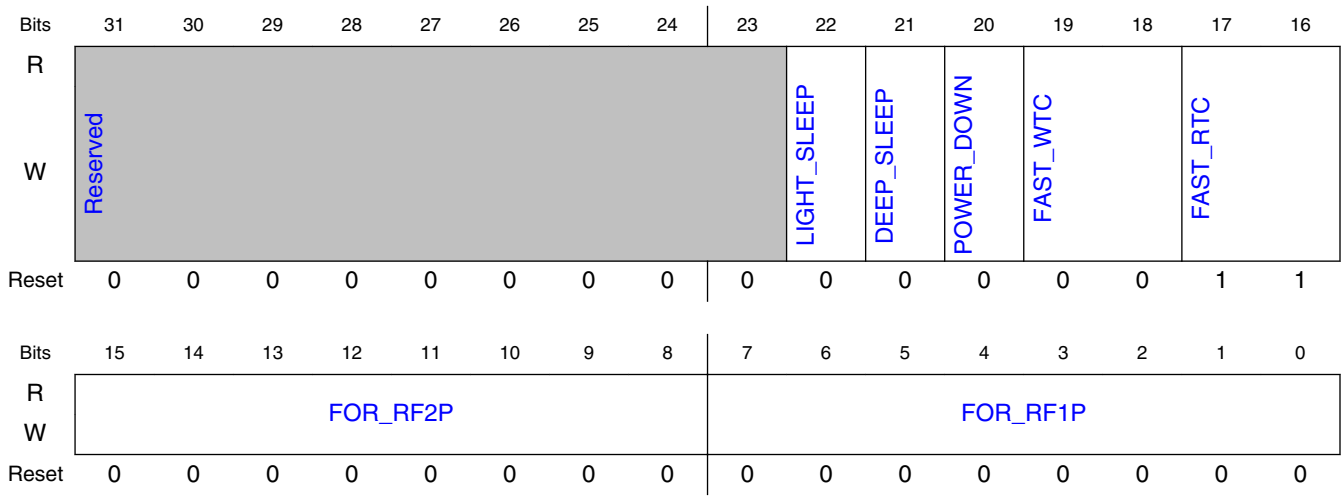
Field	Function
31-8 —	Reserved
7-0 MAX_OUTSTANDING_READS	Maximum Outstanding Reads Limits the total number of outstanding read requests

13.16.6.1.20 Register Timing Control Register (AQRegisterTimingControl)

13.16.6.1.20.1 Offset

Register	Offset
AQRegisterTimingControl	42Ch

13.16.6.1.20.2 Diagram



13.16.6.1.20.3 Fields

Field	Function
31-23 —	Reserved
22 LIGHT_SLEEP	Light sleep Bit to allow SOC to manage sleep for embedded memories
21 DEEP_SLEEP	Deep sleep Bit to allow SOC to manage sleep for embedded memories
20 POWER_DOWN	Power Down Memory
19-18 FAST_WTC	WTC for fast RAM
17-16 FAST_RTC	RTC for fast RAM
15-8 FOR_RF2P	For 2 port RAM
7-0 FOR_RF1P	For 1 port RAM

13.16.6.1.21 Command Buffer Base Address Register (AQCmdBufferAddr)

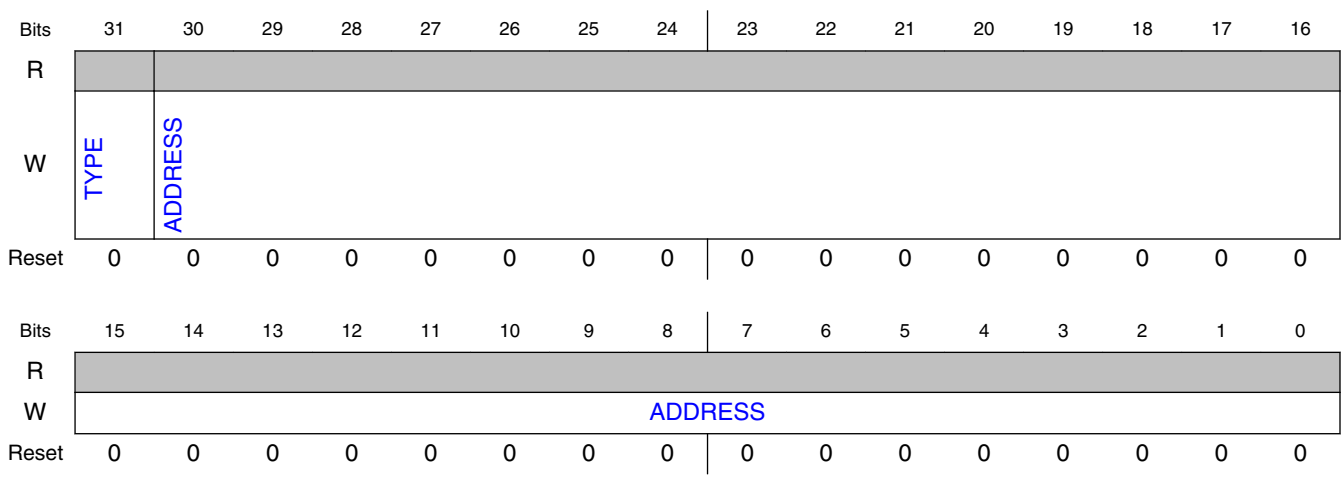
13.16.6.1.21.1 Offset

Register	Offset
AQCmdBufferAddr	654h

13.16.6.1.21.2 Function

The address must be 64-bit aligned, physical address. To check the value of the current fetch address use AQFEDebugCurCmdAdr. Since this is a write-only register, it has no set reset value.

13.16.6.1.21.3 Diagram



13.16.6.1.21.4 Fields

Field	Function
31 TYPE	Type 0b - System 1b - Virtual System
30-0 ADDRESS	Address

13.16.6.1.22 Command Decoder Address Register (AQFEDebugCurCmdAdr)

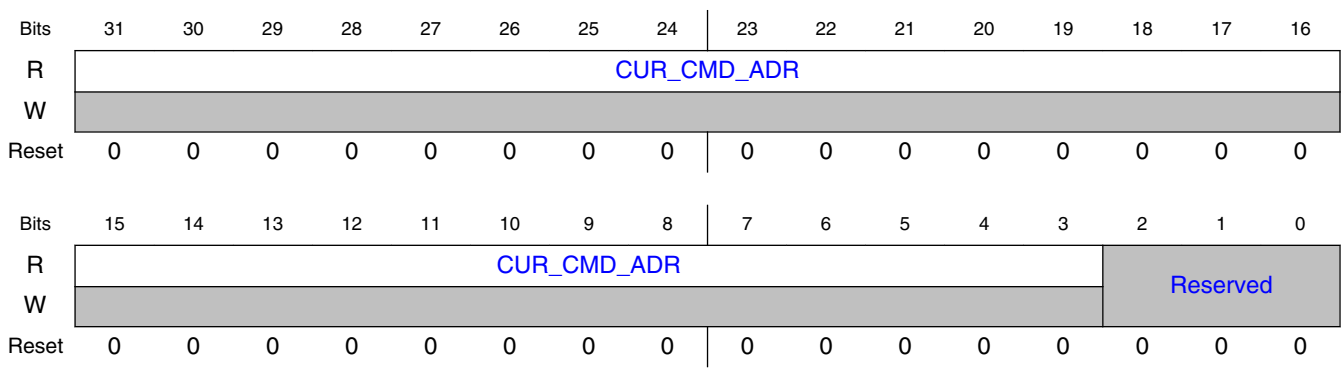
13.16.6.1.22.1 Offset

Register	Offset
AQFEDebugCurCmdAdr	664h

13.16.6.1.22.2 Function

This is a read-only register

13.16.6.1.22.3 Diagram



13.16.6.1.22.4 Fields

Field	Function
31-3 CUR_CMD_ADR	Command decoder Address
2-0 —	Reserved