# Programmers Model

NetSpeed NoC delivers a rich set of registers used for NoC control, debug and performance monitoring of the NoC. This section describes common registers available to SoC designers. The complete list of registers implemented for a specific design can be found in the noc\_reference\_manual.html under the <project> folder when the RTL is generated by NocStudio.

Registers are divided into the following categories:

* Router registers
* Bridge registers
* AMBA registers
  + AXI Master registers
  + AXI Slave registers
  + Protocol Converter registers
    - AHB2AXI
    - AXI2AHB
    - APB

## Router Registers

### RID – Router ID

This register holds layer and position information for the router. It is a read-only register. It can be used for debugging software access to the NoC elements by confirming that a read has successfully targeted the correct NoC element.

Attribute: R

Security: Non-secure

Bit field description:

* **ONE** [24] - One
* **ZERO** [23:21] - Zeroes
* **POS** [20:5] - 16-bit position ID of this router in the NoC
* **LAYER** [4:0] - 5-bit identifier of the NoC layer on which this router is located

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | | ONE | ZERO | | | POS | | | | | | | | | | | | | | | | LAYER | | | | |

Table 1 ID register

### RPERR – Router Parity Error

There is one register for each router port capturing parity error events occurring on the port. Parity errors are monitored on router physical link and also on data read from VC buffers of the router. Error status bits are sticky. First detected error while the status bit is in cleared state sets the bit. The bit needs to be explicitly cleared using zero write, before another error can be logged for that status bit. Following fields of information transported over the NoC are monitored for error at router ports. [FATAL] all bits in this register are classified as fatal for interrupt purpose.

1. Data Parity: Parity is checked over multiple segments of data in each flit. Parity error in any segment will be recorded in the data parity status bit. Note that parity is checked on data only if parity mode error check is enabled on the router's layer. In ECC mode, data parity is not monitored on each router.
2. User sideband parity: Similar to data field above.
3. Packet control parity: Parity over start of packet, end of packet, byte valid and data valid fields of a flit.
4. Routing information parity: Parity over routing information carried in every flit.
5. Credit parity: Parity monitored over credits returned downstream port.

Attribute: WZC

Security: Non-secure

Bit field description:

* **RI\_3** [31] - 1'b1: Parity Error in VC 3 Buffer Routing Information
* **PK\_3** [30] - 1'b1: Parity Error in VC 3 Buffer Packet Delineation Controls
* **SB\_3** [29] - 1'b1: Parity Error in VC 3 Buffer User Sideband
* **D\_3** [28] - 1'b1: Parity Error in VC 3 Buffer Data
* **RI\_2** [27] - 1'b1: Parity Error in VC 2 Buffer Routing Information
* **PK\_2** [26] - 1'b1: Parity Error in VC 2 Buffer Packet Delineation Controls
* **SB\_2** [25] - 1'b1: Parity Error in VC 2 Buffer User Sideband
* **D\_2** [24] - 1'b1: Parity Error in VC 2 Buffer Data
* **RI\_1** [23] - 1'b1: Parity Error in VC 1 Buffer Routing Information
* **PK\_1** [22] - 1'b1: Parity Error in VC 1 Buffer Packet Delineation Controls
* **SB\_1** [21] - 1'b1: Parity Error in VC 1 Buffer User Sideband
* **D\_1** [20] - 1'b1: Parity Error in VC 1 Buffer Data
* **RI\_0** [19] - 1'b1: Parity Error in VC 0 Buffer Routing Information
* **PK\_0** [18] - 1'b1: Parity Error in VC 0 Buffer Packet Delineation Controls
* **SB\_0** [17] - 1'b1: Parity Error in VC 0 Buffer User Sideband
* **D\_0** [16] - 1'b1: Parity Error in VC 0 Buffer Data
* **CR** [4] - 1'b1: Parity Error in Link Credit from Downstream Router
* **RI** [3] - 1'b1: Parity Error in Link Routing Information
* **PK** [2] - 1'b1: Parity Error in Link Packet Delineation Controls
* **SB** [1] - 1'b1: Parity Error in Link User Sideband
* **D** [0] - 1'b1: Parity Error in Link Data

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RI\_3 | PK\_3 | SB\_3 | D\_3 | RI\_2 | PK\_2 | SB\_2 | D\_2 | RI\_1 | PK\_1 | SB\_1 | D\_1 | RI\_0 | PK\_0 | SB\_0 | D\_0 | u | | | | | | | | | | | CR | RI | PK | SB | D |

Table 2 RPERR register

### RPERRM – Router Parity Error Mask

One mask register bit for each parity status bit in RPERR. When mask bit is set, corresponding parity error does not cause an interrupt. Default state is reset for all mask bits, allowing interrupt on any parity error event

Attribute: RW

Security: Non-secure

Bit field description:

* **RI\_3** [31] - Mask Parity Error in VC 3 Buffer Routing Information.
* **PK\_3** [30] - Mask Parity Error in VC 3 Buffer Packet Delineation Controls.
* **SB\_3** [29] - Mask Parity Error in VC 3 Buffer User Sideband.
* **D\_3** [28] - Mask Parity Error in VC 3 Buffer Data.
* **RI\_2** [27] - Mask Parity Error in VC 2 Buffer Routing Information.
* **PK\_2** [26] - Mask Parity Error in VC 2 Buffer Packet Delineation Controls.
* **SB\_2** [25] - Mask Parity Error in VC 2 Buffer User Sideband.
* **D\_2** [24] - Mask Parity Error in VC 2 Buffer Data.
* **RI\_1** [23] - Mask Parity Error in VC 1 Buffer Routing Information.
* **PK\_1** [22] - Mask Parity Error in VC 1 Buffer Packet Delineation Controls.
* **SB\_1** [21] - Mask Parity Error in VC 1 Buffer User Sideband.
* **D\_1** [20] - Mask Parity Error in VC 1 Buffer Data.
* **RI\_0** [19] - Mask Parity Error in VC 0 Buffer Routing Information.
* **PK\_0** [18] - Mask Parity Error in VC 0 Buffer Packet Delineation Controls.
* **SB\_0** [17] - Mask Parity Error in VC 0 Buffer User Sideband.
* **D\_0** [16] - Mask Parity Error in VC 0 Buffer Data.
* **CR** [4] - Mask Parity Error in Link Credit from Downstream Router.
* **RI** [3] - Mask Parity Error in Link Routing Information.
* **PK** [2] - Mask Parity Error in Link Packet Delineation Controls.
* **SB** [1] - Mask Parity Error in Link User Sideband.
* **D** [0] - Mask Parity Error in Link Data.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RI\_3 | PK\_3 | SB\_3 | D\_3 | RI\_2 | PK\_2 | SB\_2 | D\_2 | RI\_1 | PK\_1 | SB\_1 | D\_1 | RI\_0 | PK\_0 | SB\_0 | D\_0 | u | | | | | | | | | | | CR | RI | PK | SB | D |

Table 3. RPERRM register.

### RCGC – Router Clock Gating Count

Programmable interval used by coarse clock gating logic in routers. This count determines the consecutive number of idle cycles after which a router output port initiates coarse clock gating of the local port clock and de-asserts the 'busy' signal to the downstream router. This signal indicates inactivity to the downstream router and allows it to initiate coarse clock gating of its corresponding input port.

Attribute: RW

Security: Non-secure

Bit field description:

* **HYSTERESIS\_COUNTER** [31:0] - Hysteresis counter

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| HYSTERESIS\_COUNTER | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Table 4 RCGC register

### RCGO – Router Clock Gating Override

This register is used by coarse grained clock gating logic. This register can be set to override coarse clock gating for the entire router. Coarse clock gating for selective routers can be overridden by locally setting this register, if the user does not want incur and aggregate coarse clock gating cycle penalty over a "fast path/critical path" through the NoC.

Attribute: RW

Security: Non-secure

Bit field description:

* **FPO** [0] -   
  1'b1: Coarse clock gating is locally disabled (for fast path)  
  1'b0: Coarse clock gating is locally enabled

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | FPO |

Table 5. RCGO register

### RE – Router Event

This register tracks the interrupt or error events that can occur in the router. The only interrupt event is the event counter overflow. This register is readable, and can be cleared by performing a write with the write data bits set to 0 for the bits that should be cleared.

Attribute: WZC

Security: Non-secure

Bit field description:

* **KLU** [16] -   
  1'b1: Traffic destined for K link which is unavailable
* **JLU** [15] -   
  1'b1: Traffic destined for J link which is unavailable
* **ILU** [14] -   
  1'b1: Traffic destined for I link which is unavailable
* **HLU** [13] -   
  1'b1: Traffic destined for H link which is unavailable
* **SLU** [12] -   
  1'b1: Traffic destined for South link which is unavailable
* **WLU** [11] -   
  1'b1: Traffic destined for West link which is unavailable
* **ELU** [10] -   
  1'b1: Traffic destined for East link which is unavailable
* **NLU** [9] -   
  1'b1: Traffic destined for North link which is unavailable
* **PGE** [8] -   
  1'b1: Power gating error, traffic received after router committed to power down
* **OVFO** [2] -   
  1'b1: In this status bit indicates that the router output event counter has overflowed (32'hFFFFFFFF -> 32'dh0), this is a sticky status bit  
  1'b0: To clear
* **CSR\_PARERR** [1] -   
  1'b1: Parity error in config/status registers
* **OVFI** [0] -   
  1'b1: In this status bit indicates that the router input event counter has overflowed (32'hFFFFFFFF -> 32'dh0), this is a sticky status bit  
  1'b0: To clear

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | | | | | | | | | | KLU | JLU | ILU | HLU | SLU | WLU | ELU | NLU | PGE | u | | | | | OVFO | CSR\_PARERR | OVFI |

Table 6. RE register

### REC – Router Event Counter

This register holds the event counter. The value can be read to determine the current count value. The value can be written to initialize the counter. When events trigger a count, the counter will increment. When the counter increments at its highest value, it will roll over to zero and the overflow will mark the Router Event Interrupt Status register, which could trigger an interrupt.

Attribute: RW

Security: Non-secure

Bit field description:

* **EVENT\_CNTR** [31:0] - 32'bit event incrementing counter. Rollover from 32'hFFFFF -> 32'd0 sets the rollover status bit RE

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| EVENT\_CNTR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Table 7 REC register

### RECC – Router Event Counter Control

This register is used to select which hardware events will increment the event counter.

Attribute: RW

Security: Non-secure

Bit field description:

* **EVT** [9:8] -   
  11: Generates count event when VC has valid data, but is stalled  
  10: Generates count event on every flit received for the selected input port and selected input VCs, this can be used to count total flits received on a router input port  
  01: Generates count event on every EOP received for the selected input port and selected input VCs, this can be used to count packets received on a router input port  
  00: Disable
* **INP** [6:4] - Input port on which the event is captured
* **IVC** [1:0] -   
  11: Input VC 3  
  10: Input VC 2  
  01: Input VC 1  
  00: Input VC 0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | | | | | | | | | | | | | | | | | EVT | | u | INP | | | u | | IVC | |

Table 8. RECC register

### REM – Router Event Mask

This register is used to select whether the interrupt events in the Router Event Interrupt Status register should send an interrupt when asserted. If the corresponding bit is set to 1, an interrupt will not be sent. This register can be read and written to.

Attribute: RW

Security: Non-secure

Bit field description:

* **MK** [16] -   
  1'b1: Mask KLU error interrupt
* **MJ** [15] -   
  1'b1: Mask JLU error interrupt
* **MI** [14] -   
  1'b1: Mask ILU error interrupt
* **MH** [13] -   
  1'b1: Mask HLU error interrupt
* **MS** [12] -   
  1'b1: Mask SLU error interrupt
* **MW** [11] -   
  1'b1: Mask WLU error interrupt
* **ME** [10] -   
  1'b1: Mask ELU error interrupt
* **MN** [9] -   
  1'b1: Mask NLU error interrupt
* **PGM** [8] -   
  1'b1: Mask PGE error interrupt
* **OVFOM** [2] -   
  1'b1: Masks or disables an interrupt from being generated by the output event count overflow status bit (RE)  
  1'b0: Enables an interrupt to be generated when event counter status bit is set
* **CSR\_PARERRM** [1] -   
  1'b1: Mask CSR parity error interrupt
* **OVFIM** [0] -   
  1'b1: Masks or disables an interrupt from being generated by the input event count overflow status bit (RE)  
  1'b0: Enables an interrupt to be generated when event counter status bit is set

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | | | | | | | | | | MK | MJ | MI | MH | MS | MW | ME | MN | PGM | u | | | | | OVFOM | CSR\_PARERRM | OVFIM |

Table 9. REM register

### RIVCS – Router Input VC Status

This register indicates the current status of a single input port of a router. Each register tracks the status of up to 4 virtual channels for the input port. There are 8 RIVCS per router, one for each router's input port.

Attribute: R

Security: Non-secure

Bit field description:

* **V\_3** [31] -   
  1'b1: Head flit valid (buffer ready) in VC 3
* **F\_3** [30] -   
  1'b1: Buffer full in VC 3
* **B\_3** [29] -   
  1'b1: Indicates that the head flit of the VC 3 is of the 'QoS Barrier' type  
  1'b0: Indicates that the head flit of the VC 3 is of the 'QoS Normal' type
* **S\_3** [28] -   
  1'b1: Indicates that the head flit is a start of packet. This also indicates that this input VC 3 has not yet acquired its corresponding output VC  
  1'b0: Indicates that the head flit is not a start of packet. Also indicates that this input VC 3 has already acquired the VC on the output port
* **UP\_3** [27] -   
  1'b1: Indicates that the flit accumulator on this VC 3 for upsizing to an output port is currently holding a flit  
  1'b0: Indicates that either the upsizing accumulator is empty or there is no upsizing from the VC 3
* **OUTP\_3** [26:24] - Value indicates the router output port to which the packet at the head of the VC 3 is destined to: 3'd0: N, 3'd1: E, 3'd2: W, 3'd3:S, 3'd4:H, 3'd5: I, 3'd6: J, 3'd7: K
* **V\_2** [23] -   
  1'b1: Head flit valid (buffer ready) in VC 2
* **F\_2** [22] -   
  1'b1: Buffer full in VC 2
* **B\_2** [21] -   
  1'b1: Indicates that the head flit of the VC 2 is of the 'QoS Barrier' type  
  1'b0: Indicates that the head flit of the VC 2 is of the 'QoS Normal' type
* **S\_2** [20] -   
  1'b1: Indicates that the head flit is a start of packet. This also indicates that this input VC 2 has not yet acquired its corresponding output VC  
  1'b0: Indicates that the head flit is not a start of packet. Also indicates that this input VC 2 has already acquired the VC on the output port
* **UP\_2** [19] -   
  1'b1: Indicates that the flit accumulator on this VC 2 for upsizing to an output port is currently holding a flit  
  1'b0: Indicates that either the upsizing accumulator is empty or there is no upsizing from the VC 2
* **OUTP\_2** [18:16] - Value indicates the router output port to which the packet at the head of the VC 2 is destined to: 3'd0: N, 3'd1: E, 3'd2: W, 3'd3:S, 3'd4:H, 3'd5: I, 3'd6: J, 3'd7: K
* **V\_1** [15] -   
  1'b1: Head flit valid (buffer ready) in VC 1
* **F\_1** [14] -   
  1'b1: Buffer full in VC 1
* **B\_1** [13] -   
  1'b1: Indicates that the head flit of the VC 1 is of the 'QoS Barrier' type  
  1'b0: Indicates that the head flit of the VC 1 is of the 'QoS Normal' type
* **S\_1** [12] -   
  1'b1: Indicates that the head flit is a start of packet. This also indicates that this input VC 1 has not yet acquired its corresponding output VC  
  1'b0: Indicates that the head flit is not a start of packet. Also indicates that this input VC 1 has already acquired the VC on the output port
* **UP\_1** [11] -   
  1'b1: Indicates that the flit accumulator on this VC 1 for upsizing to an output port is currently holding a flit  
  1'b0: Indicates that either the upsizing accumulator is empty or there is no upsizing from the VC 1
* **OUTP\_1** [10:8] - Value indicates the router output port to which the packet at the head of the VC 1 is destined to: 3'd0: N, 3'd1: E, 3'd2: W, 3'd3:S, 3'd4:H, 3'd5: I, 3'd6: J, 3'd7: K
* **V\_0** [7] -   
  1'b1: Head flit valid (buffer ready) in VC 0
* **F\_0** [6] -   
  1'b1: Buffer full in VC 0
* **B\_0** [5] -   
  1'b1: Indicates that the head flit of the VC 0 is of the 'QoS Barrier' type  
  1'b0: Indicates that the head flit of the VC 0 is of the 'QoS Normal' type
* **S\_0** [4] -   
  1'b1: Indicates that the head flit is a start of packet. This also indicates that this input VC 0 has not yet acquired its corresponding output VC  
  1'b0: Indicates that the head flit is not a start of packet. Also indicates that this input VC 0 has already acquired the VC on the output port
* **UP\_0** [3] -   
  1'b1: Indicates that the flit accumulator on this VC 0 for upsizing to an output port is currently holding a flit  
  1'b0: Indicates that either the upsizing accumulator is empty or there is no upsizing from the VC 0
* **OUTP\_0** [2:0] - Value indicates the router output port to which the packet at the head of the VC 0 is destined to: 3'd0: N, 3'd1: E, 3'd2: W, 3'd3:S, 3'd4:H, 3'd5: I, 3'd6: J, 3'd7: K

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| V\_3 | F\_3 | B\_3 | S\_3 | UP\_3 | OUTP\_3 | | | V\_2 | F\_2 | B\_2 | S\_2 | UP\_2 | OUTP\_2 | | | V\_1 | F\_1 | B\_1 | S\_1 | UP\_1 | OUTP\_1 | | | V\_0 | F\_0 | B\_0 | S\_0 | UP\_0 | OUTP\_0 | | |

Table 10. RIVCS register

### ROEC – Router Output Event Counter

This register holds the output event counter. The value can be read to determine the current count value. The value can be written to initialize the counter. When events trigger a count, the counter will increment. When the counter increments at its highest value, it will roll over to zero and the overflow will mark the Router output Event Interrupt Status register, which could trigger an interrupt.

Attribute: RW

Security: Non-secure

Bit field description:

* **EVENT\_CNTR** [31:0] - 32'bit event incrementing counter. Rollover from 32'hFFFFF -> 32'd0 sets the rollover status bit RE

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| EVENT\_CNTR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Table 11. ROEC register

### ROECC – Router Output Event Counter Control

This register is used to select which hardware events will increment the event counter.

Attribute: RW

Security: Non-secure

Bit field description:

* **EVT** [10:8] -   
  100: Port stalled. Input flits are available for the port, but no output VC has credit  
  011: Generates count event when flits are available to be sent to output VC, but the VC has no credit  
  010: Generates count event on every flit sent on the selected output port and selected output VCs, this can be used to count total flits sent on a router output port  
  001: Generates count event on every EOP sent on the selected output port and selected output VCs, this can be used to count packets sent on a router output port  
  000: Disable
* **OP** [6:4] - Output port on which the event is captured
* **OVC** [3:0] - Bit map to select output VCs to monitor events on

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | | | | | | | | | | | | | | | | EVT | | | u | OP | | | OVC | | | |

Table 12. ROECC register

### ROVCS – Router Output VC Status

This register indicates the current status of one of the output ports of a router. Each register tracks the status of up to 4 virtual channels for the output port. There are 8 ROVCS per router, one for each router's output port (only 5 are active registers, while the other 3 are reserved).

Attribute: R

Security: Non-secure

Bit field description:

* **RSV\_3** [27] - Reserved
* **VB\_3** [26] -   
  1'b1: Indicates that this output VC 3 is currently locked to the corresponding VC on one of the input ports.  
  1'b0: Indicates that this output VC 3 is free and can be acquired by the corresponding VC on one of the input ports for the transmission of a packet.
* **CE\_3** [25] -   
  1'b1: Indicates that this output VC 3 has no credit for transmission of flits to the downstream link.  
  1'b0: Indicates that credit is available for transmission to downstream link.
* **CF\_3** [24] -   
  1'b1: Indicates that the credit level with this VC 3 is at the maximum provisioned value.
* **RSV\_2** [19] - Reserved
* **VB\_2** [18] -   
  1'b1: Indicates that this output VC 2 is currently locked to the corresponding VC on one of the input ports.  
  1'b0: Indicates that this output VC 2 is free and can be acquired by the corresponding VC on one of the input ports for the transmission of a packet.
* **CE\_2** [17] -   
  1'b1: Indicates that this output VC 2 has no credit for transmission of flits to the downstream link.  
  1'b0: Indicates that credit is available for transmission to downstream link.
* **CF\_2** [16] -   
  1'b1: Indicates that the credit level with this VC 2 is at the maximum provisioned value.
* **RSV\_1** [11] - Reserved
* **VB\_1** [10] -   
  1'b1: Indicates that this output VC 1 is currently locked to the corresponding VC on one of the input ports.  
  1'b0: Indicates that this output VC 1 is free and can be acquired by the corresponding VC on one of the input ports for the transmission of a packet.
* **CE\_1** [9] -   
  1'b1: Indicates that this output VC 1 has no credit for transmission of flits to the downstream link.  
  1'b0: Indicates that credit is available for transmission to downstream link.
* **CF\_1** [8] -   
  1'b1: Indicates that the credit level with this VC 1 is at the maximum provisioned value.
* **RSV\_0** [3] - Reserved
* **VB\_0** [2] -   
  1'b1: Indicates that this output VC 0 is currently locked to the corresponding VC on one of the input ports.  
  1'b0: Indicates that this output VC 0 is free and can be acquired by the corresponding VC on one of the input ports for the transmission of a packet.
* **CE\_0** [1] -   
  1'b1: Indicates that this output VC 0 has no credit for transmission of flits to the downstream link.  
  1'b0: Indicates that credit is available for transmission to downstream link.
* **CF\_0** [0] -   
  1'b1: Indicates that the credit level with this VC 0 is at the maximum provisioned value.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | RSV\_3 | VB\_3 | CE\_3 | CF\_3 | u | | | | RSV\_2 | VB\_2 | CE\_2 | CF\_2 | u | | | | RSV\_1 | VB\_1 | CE\_1 | CF\_1 | u | | | | RSV\_0 | VB\_0 | CE\_0 | CF\_0 |

Table 13 ROVCS register

## Streaming Bridge registers

### BRPERR0 – Bridge Receive Parity Error 0

Receive bridge parity error status register monitoring parity errors on enabled layers from 0 to 7 (BRPERR0), and from 8 to 15 (BRPERR1). Parity/ECC error are monitored and captured for physical link to the bridge on each NoC layer. Following fields are monitored.

1. Data ECC/Parity: Parity/ECC is checked over multiple segments of data in each flit. An error in any segment will be recorded in the data ECC/parity error status bit. In ECC mode, single bit errors are corrected and the event is recorded.
2. User sideband ECC/parity: Similar to data field above.
3. Packet control parity: Parity over start of packet, end of packet, byte valid and data valid fields of a flit.

This register makes use of the logical layer mapping (and not the physical layer mapping). For the physical to logical table, please refer to the Physical to Logical Layer Mapping section in the help.

Correctable errors will raise interrupt\_nfatal if fatal/nonfatal interrupt mode is configured. All other error types are considered fatal.

Attribute: WZC

Security: Non-secure

Bit field description:

* **PK0** [4] - Parity error in packet delineation controls in layer 0
* **SBC0** [3] - Correctable single bit user sideband error (only ECC) in layer 0
* **SB0** [2] - Uncorrectable User sideband ECC/parity error in layer 0
* **DC0** [1] - Correctable single bit data error (only ECC) in layer 0
* **D0** [0] - Uncorrectable Data ECC/parity error in layer 0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | u | | | | | | | | u | | | | | | | | u | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | | | u | | | | | | | | u | | | | | | | | u | | | PK0 | SBC0 | SB0 | DC0 | D0 |

Table 14 BRPERR0 register

### BRPERR1 – Bridge Receive Parity Error 1

Receive bridge parity error status register monitoring parity errors on enabled layers from 0 to 7 (BRPERR0), and from 8 to 15 (BRPERR1). Parity/ECC error are monitored and captured for physical link to the bridge on each NoC layer. Following fields are monitored.

1. Data ECC/Parity: Parity/ECC is checked over multiple segments of data in each flit. An error in any segment will be recorded in the data ECC/parity error status bit. In ECC mode, single bit errors are corrected and the event is recorded.
2. User sideband ECC/parity: Similar to data field above.
3. Packet control parity: Parity over start of packet, end of packet, byte valid and data valid fields of a flit.

This register makes use of the logical layer mapping (and not the physical layer mapping). For the physical to logical table, please refer to the Physical to Logical Layer Mapping section in the help.

Correctable errors will raise interrupt\_nfatal if fatal/nonfatal interrupt mode is configured. All other error types are considered fatal.

Attribute: WZC

Security: Non-secure

Bit field description: The same as BRPERR0 when applicable

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | u | | | | | | | | u | | | | | | | | u | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | | | u | | | | | | | | u | | | | | | | | u | | | | | | | |

Table 15 BRPERR1 register

### BRPERRM0 – Bridge Receive Parity Error Mask 0

Mask registers for receive bridge parity error interrupts from register BRPERR0 and BRPERR1. One mask register bit for each parity status bit in BRPERR. When mask bit is set, corresponding parity error does not cause an interrupt. Default state is reset for all mask bits, allowing interrupt on any parity error event.

This register makes use of the logical layer mapping (and not the physical layer mapping). For the physical to logical table, please refer to the Physical to Logical Layer Mapping section in the help.

Attribute: RW

Security: Non-secure

Bit field description:

* **PK0** [4] - Mask Parity error in packet delineation controls in layer 0
* **SBC0** [3] - Mask Correctable single bit user sideband error (only ECC) in layer 0
* **SB0** [2] - Mask User sideband ECC/parity error in layer 0
* **DC0** [1] - Mask Correctable single bit data error (only ECC) in layer 0
* **D0** [0] - Mask Data ECC/parity error in layer 0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | u | | | | | | | | u | | | | | | | | u | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | | | u | | | | | | | | u | | | | | | | | u | | | PK0 | SBC0 | SB0 | DC0 | D0 |

Table 16 BRPERRM0 register

### BRPERRM1 - Bridge Receive Parity Error Mask 1

Mask registers for receive bridge parity error interrupts from register BRPERR0 and BRPERR1. One mask register bit for each parity status bit in BRPERR. When mask bit is set, corresponding parity error does not cause an interrupt. Default state is reset for all mask bits, allowing interrupt on any parity error event.

This register makes use of the logical layer mapping (and not the physical layer mapping). For the physical to logical table, please refer to the Physical to Logical Layer Mapping section in the help.

Attribute: RW

Security: Non-secure

Bit field description: The same as BRPERRM0 when applicable

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | u | | | | | | | | u | | | | | | | | u | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | | | u | | | | | | | | u | | | | | | | | u | | | | | | | |

Table 17 BRPERRM1 register

### BRS – Bridge Receive Status

These registers track the status of the bridge's receive FIFOs from the NoC. Since there is up to 16 layers of the NoC, there are 16 registers. Each register tracks the status of one virtual channel, with up to 4 virtual channels per layer.

Attribute: R

Security: Non-secure

Bit field description:

* **F\_3** [29] - Buffer full for VC 3 Layer 0
* **B\_3** [28] - Head flit barrier state for VC 3 Layer 0
* **S\_3** [27] - Head flit sop for VC 3 Layer 0
* **V\_3** [26] - Head flit (buffer ready) for VC 3 Layer 0
* **OUTI\_3** [25:24] - Head flit output interface for VC 3 Layer 0
* **F\_2** [21] - Buffer full for VC 2 Layer 0
* **B\_2** [20] - Head flit barrier state for VC 2 Layer 0
* **S\_2** [19] - Head flit sop for VC 2 Layer 0
* **V\_2** [18] - Head flit (buffer ready) for VC 2 Layer 0
* **OUTI\_2** [17:16] - Head flit output interface for VC 2 Layer 0
* **F\_1** [13] - Buffer full for VC 1 Layer 0
* **B\_1** [12] - Head flit barrier state for VC 1 Layer 0
* **S\_1** [11] - Head flit sop for VC 1 Layer 0
* **V\_1** [10] - Head flit (buffer ready) for VC 1 Layer 0
* **OUTI\_1** [9:8] - Head flit output interface for VC 1 Layer 0
* **F\_0** [5] - Buffer full for VC 0 Layer 0
* **B\_0** [4] - Head flit barrier state for VC 0 Layer 0
* **S\_0** [3] - Head flit sop for VC 0 Layer 0
* **V\_0** [2] - Head flit (buffer ready) for VC 0 Layer 0
* **OUTI\_0** [1:0] - Head flit output interface for VC 0 Layer 0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | F\_3 | B\_3 | S\_3 | V\_3 | OUTI\_3 | | u | | F\_2 | B\_2 | S\_2 | V\_2 | OUTI\_2 | | u | | F\_1 | B\_1 | S\_1 | V\_1 | OUTI\_1 | | u | | F\_0 | B\_0 | S\_0 | V\_0 | OUTI\_0 | |

Table 18 BRS register

### BRUS – Bridge Receive Upsizer Status

This register tracks the status of the bridge receiver upsizing and downsizing structure. It can be used with the other status registers to check for packets that are still occupying the bridge. Each of the host's receiving interfaces, up to 4, can have upsizing/downsizing logic, and this register tracks the status of all 4 interfaces.

Attribute: R

Security: Non-secure

Bit field description:

* **V\_D** [3] - Interface D upsizer/downsizer valid
* **V\_C** [2] - Interface C upsizer/downsizer valid
* **V\_B** [1] - Interface B upsizer/downsizer valid
* **V\_A** [0] - Interface A upsizer/downsizer valid

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | V\_D | V\_C | V\_B | V\_A |

Table 19 BRUS register

### BTPERR – Bridge Transmit Parity Error

Transmit bridge parity error status register. One register bit per layer, to monitor error in credit return signals from the downstream port. Error status bits are sticky. First detected error while the status bit is in cleared state sets the bit. The bit needs to be explicitly cleared using zero write, before another error can be logged for that status bit.

Attribute: WZC

Security: Non-secure

Bit field description:

* **L15** [15] -   
  1'b1: Credit parity error on layer 15
* **L14** [14] -   
  1'b1: Credit parity error on layer 14
* **L13** [13] -   
  1'b1: Credit parity error on layer 13
* **L12** [12] -   
  1'b1: Credit parity error on layer 12
* **L11** [11] -   
  1'b1: Credit parity error on layer 11
* **L10** [10] -   
  1'b1: Credit parity error on layer 10
* **L9** [9] -   
  1'b1: Credit parity error on layer 9
* **L8** [8] -   
  1'b1: Credit parity error on layer 8
* **L7** [7] -   
  1'b1: Credit parity error on layer 7
* **L6** [6] -   
  1'b1: Credit parity error on layer 6
* **L5** [5] -   
  1'b1: Credit parity error on layer 5
* **L4** [4] -   
  1'b1: Credit parity error on layer 4
* **L3** [3] -   
  1'b1: Credit parity error on layer 3
* **L2** [2] -   
  1'b1: Credit parity error on layer 2
* **L1** [1] -   
  1'b1: Credit parity error on layer 1
* **L0** [0] -   
  1'b1: Credit parity error on layer 0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | | | | | | | | | | | L15 | L14 | L13 | L12 | L11 | L10 | L9 | L8 | L7 | L6 | L5 | L4 | L3 | L2 | L1 | L0 |

Table 20 BTPERR register

### BTPERRM – Bridge Transmit Parity Error Mask

Mask register for transmit bridge parity error interrupts. One mask register bit for each parity status bit in BTPERR. When mask bit is set, corresponding parity error does not cause an interrupt. Default state is reset for all mask bits, allowing interrupt on any parity error event.

Attribute: RW

Security: Non-secure

Bit field description:

* **L15** [15] -   
  1'b1: Interrupt Mask Credit parity error on layer 15
* **L14** [14] -   
  1'b1: Interrupt Mask Credit parity error on layer 14
* **L13** [13] -   
  1'b1: Interrupt Mask Credit parity error on layer 13
* **L12** [12] -   
  1'b1: Interrupt Mask Credit parity error on layer 12
* **L11** [11] -   
  1'b1: Interrupt Mask Credit parity error on layer 11
* **L10** [10] -   
  1'b1: Interrupt Mask Credit parity error on layer 10
* **L9** [9] -   
  1'b1: Interrupt Mask Credit parity error on layer 9
* **L8** [8] -   
  1'b1: Interrupt Mask Credit parity error on layer 8
* **L7** [7] -   
  1'b1: Interrupt Mask Credit parity error on layer 7
* **L6** [6] -   
  1'b1: Interrupt Mask Credit parity error on layer 6
* **L5** [5] -   
  1'b1: Interrupt Mask Credit parity error on layer 5
* **L4** [4] -   
  1'b1: Interrupt Mask Credit parity error on layer 4
* **L3** [3] -   
  1'b1: Interrupt Mask Credit parity error on layer 3
* **L2** [2] -   
  1'b1: Interrupt Mask Credit parity error on layer 2
* **L1** [1] -   
  1'b1: Interrupt Mask Credit parity error on layer 1
* **L0** [0] -   
  1'b1: Interrupt Mask Credit parity error on layer 0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | | | | | | | | | | | L15 | L14 | L13 | L12 | L11 | L10 | L9 | L8 | L7 | L6 | L5 | L4 | L3 | L2 | L1 | L0 |

Table 21 BTPERRM register

### BTRL – Bridge Transmit Rate Limiter

This is a register per host interface of Tx Bridge for QoS, used to control the rate of Traffic injection from host to the NoC.

Attribute: RW

Security: Non-secure

Bit field description:

* **EN** [20] -   
  1'b1: Rate limit logic enable; rate limiter logic is used for arbitration only.  
  1'b0: Rate limit logic disable
* **CNT** [19:16] - Max Count Value for Token. Anytime the token count is greater than zero, the host gets qualified to inject message into NoC.
* **WT** [11:0] - Starting Weight, for traffic issue to the NoC from the host interface.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | | | | | | EN | CNT | | | | r | | | | WT | | | | | | | | | | | |

Table 22 BTRL register

### BTUS\_0 – Bridge Transmit Upsizer Status 0

These two registers (BTUS\_0 and BTUS\_1) track the status of the bridge transmitter upsizer/downsize structure. They can be used with the other status registers to check for packets that are still occupying the bridge. Each NoC layer, up to 16, can have upsizing/downsizing logic, and these 2 registers track the status of all 16 layers (BTUS\_0 from 0 to 7 and BTUS\_1 from 8 to 15).

Attribute: R

Security: Non-secure

Bit field description:

* **L7\_D** [31] - Interface upsizer status for interface D, Layer 7
* **L7\_C** [30] - Interface upsizer status for interface C, Layer 7
* **L7\_B** [29] - Interface upsizer status for interface B, Layer 7
* **L7\_A** [28] - Interface upsizer status for interface A, Layer 7
* **L6\_D** [27] - Interface upsizer status for interface D, Layer 6
* **L6\_C** [26] - Interface upsizer status for interface C, Layer 6
* **L6\_B** [25] - Interface upsizer status for interface B, Layer 6
* **L6\_A** [24] - Interface upsizer status for interface A, Layer 6
* **L5\_D** [23] - Interface upsizer status for interface D, Layer 5
* **L5\_C** [22] - Interface upsizer status for interface C, Layer 5
* **L5\_B** [21] - Interface upsizer status for interface B, Layer 5
* **L5\_A** [20] - Interface upsizer status for interface A, Layer 5
* **L4\_D** [19] - Interface upsizer status for interface D, Layer 4
* **L4\_C** [18] - Interface upsizer status for interface C, Layer 4
* **L4\_B** [17] - Interface upsizer status for interface B, Layer 4
* **L4\_A** [16] - Interface upsizer status for interface A, Layer 4
* **L3\_D** [15] - Interface upsizer status for interface D, Layer 3
* **L3\_C** [14] - Interface upsizer status for interface C, Layer 3
* **L3\_B** [13] - Interface upsizer status for interface B, Layer 3
* **L3\_A** [12] - Interface upsizer status for interface A, Layer 3
* **L2\_D** [11] - Interface upsizer status for interface D, Layer 2
* **L2\_C** [10] - Interface upsizer status for interface C, Layer 2
* **L2\_B** [9] - Interface upsizer status for interface B, Layer 2
* **L2\_A** [8] - Interface upsizer status for interface A, Layer 2
* **L1\_D** [7] - Interface upsizer status for interface D, Layer 1
* **L1\_C** [6] - Interface upsizer status for interface C, Layer 1
* **L1\_B** [5] - Interface upsizer status for interface B, Layer 1
* **L1\_A** [4] - Interface upsizer status for interface A, Layer 1
* **L0\_D** [3] - Interface upsizer status for interface D, Layer 0
* **L0\_C** [2] - Interface upsizer status for interface C, Layer 0
* **L0\_B** [1] - Interface upsizer status for interface B, Layer 0
* **L0\_A** [0] - Interface upsizer status for interface A, Layer 0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| L7\_D | L7\_C | L7\_B | L7\_A | L6\_D | L6\_C | L6\_B | L6\_A | L5\_D | L5\_C | L5\_B | L5\_A | L4\_D | L4\_C | L4\_B | L4\_A | L3\_D | L3\_C | L3\_B | L3\_A | L2\_D | L2\_C | L2\_B | L2\_A | L1\_D | L1\_C | L1\_B | L1\_A | L0\_D | L0\_C | L0\_B | L0\_A |

Table 23 BTUS\_0 register

### BTUS\_1 - Bridge Transmit Upsizer Status 1

These two registers (BTUS\_0 and BTUS\_1) track the status of the bridge transmitter upsizer/downsize structure. They can be used with the other status registers to check for packets that are still occupying the bridge. Each NoC layer, up to 16, can have upsizing/downsizing logic, and these 2 registers track the status of all 16 layers (BTUS\_0 from 0 to 7 and BTUS\_1 from 8 to 15).

Attribute: R

Security: Non-secure

Bit field description:

* **L15\_D** [31] - Interface upsizer status for interface D, Layer 15
* **L15\_C** [30] - Interface upsizer status for interface C, Layer 15
* **L15\_B** [29] - Interface upsizer status for interface B, Layer 15
* **L15\_A** [28] - Interface upsizer status for interface A, Layer 15
* **L14\_D** [27] - Interface upsizer status for interface D, Layer 14
* **L14\_C** [26] - Interface upsizer status for interface C, Layer 14
* **L14\_B** [25] - Interface upsizer status for interface B, Layer 14
* **L14\_A** [24] - Interface upsizer status for interface A, Layer 14
* **L13\_D** [23] - Interface upsizer status for interface D, Layer 13
* **L13\_C** [22] - Interface upsizer status for interface C, Layer 13
* **L13\_B** [21] - Interface upsizer status for interface B, Layer 13
* **L13\_A** [20] - Interface upsizer status for interface A, Layer 13
* **L12\_D** [19] - Interface upsizer status for interface D, Layer 12
* **L12\_C** [18] - Interface upsizer status for interface C, Layer 12
* **L12\_B** [17] - Interface upsizer status for interface B, Layer 12
* **L12\_A** [16] - Interface upsizer status for interface A, Layer 12
* **L11\_D** [15] - Interface upsizer status for interface D, Layer 11
* **L11\_C** [14] - Interface upsizer status for interface C, Layer 11
* **L11\_B** [13] - Interface upsizer status for interface B, Layer 11
* **L11\_A** [12] - Interface upsizer status for interface A, Layer 11
* **L10\_D** [11] - Interface upsizer status for interface D, Layer 10
* **L10\_C** [10] - Interface upsizer status for interface C, Layer 10
* **L10\_B** [9] - Interface upsizer status for interface B, Layer 10
* **L10\_A** [8] - Interface upsizer status for interface A, Layer 10
* **L9\_D** [7] - Interface upsizer status for interface D, Layer 9
* **L9\_C** [6] - Interface upsizer status for interface C, Layer 9
* **L9\_B** [5] - Interface upsizer status for interface B, Layer 9
* **L9\_A** [4] - Interface upsizer status for interface A, Layer 9
* **L8\_D** [3] - Interface upsizer status for interface D, Layer 8
* **L8\_C** [2] - Interface upsizer status for interface C, Layer 8
* **L8\_B** [1] - Interface upsizer status for interface B, Layer 8
* **L8\_A** [0] - Interface upsizer status for interface A, Layer 8

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| L15\_D | L15\_C | L15\_B | L15\_A | L14\_D | L14\_C | L14\_B | L14\_A | L13\_D | L13\_C | L13\_B | L13\_A | L12\_D | L12\_C | L12\_B | L12\_A | L11\_D | L11\_C | L11\_B | L11\_A | L10\_D | L10\_C | L10\_B | L10\_A | L9\_D | L9\_C | L9\_B | L9\_A | L8\_D | L8\_C | L8\_B | L8\_A |

Table 24 BTUS\_1 register

### P – QoS Profile

This register describes the weight value of each QoS supported at the bridge. Each byte of this register must be greater than or equal to 3. Each transmitting bridge supports up to 16 QoS profiles. Each QoS is composed of pri and weight, however only the weight is programmable, therefore is part of the registers.

QoS data is composed of four registers, P0, P1, P2 and P3, each of which contains the weight of four profiles. Depending upon how many QoS profiles are enabled, the appropriate bits in the following registers are available.

Attribute: RW

Security: Non-secure

Bit field description:

* **WT\_QOS\_3** [31:24] - Weight of QoS profile 3
* **WT\_QOS\_2** [23:16] - Weight of QoS profile 2
* **WT\_QOS\_1** [15:8] - Weight of QoS profile 1
* **WT\_QOS\_0** [7:0] - Weight of QoS profile 0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| WT\_QOS\_3 | | | | | | | | WT\_QOS\_2 | | | | | | | | WT\_QOS\_1 | | | | | | | | WT\_QOS\_0 | | | | | | | |

Table 25 P register

### RXE – Bridge Receive Interrupt Event

This register tracks the interrupt events in the receive portion of the streaming bridge. It resets to 0, but as these conditions occur, the corresponding bits are set to 1. This register can be read and can also be cleared by sending a write with bits set to 0 for the bits that should be cleared.

There are four events that can signal an interrupt. If the host sends more credits than the streaming bridge can take, it will signal an interrupt to indicate a protocol violation has occurred. Each interface has its own status bit. These interrupts cannot be masked.

Attribute: WZC

Security: Non-secure

Bit field description:

* **EVC1\_OFLW** [6] - Event counter1 overflow. This event can be masked so that no interrupt is sent on an overflow condition.
* **PARITY\_ERR** [5] - Register parity error interrupt
* **EVC\_OFLW** [4] - Event counter overflow. This event can be masked so that no interrupt is sent on an overflow condition.
* **CRC\_OFLW\_D** [3] - Credit counter overflow for interface D
* **CRC\_OFLW\_C** [2] - Credit counter overflow for interface C
* **CRC\_OFLW\_B** [1] - Credit counter overflow for interface B
* **CRC\_OFLW\_A** [0] - Credit counter overflow for interface A

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | EVC1\_OFLW | PARITY\_ERR | EVC\_OFLW | CRC\_OFLW\_D | CRC\_OFLW\_C | CRC\_OFLW\_B | CRC\_OFLW\_A |

Table 26 RXE register

### RXEM – Receive Event Interrupt Mask

This register is used to decide which of the error/interrupt events specified in the Receive Interrupt Status register should trigger an interrupt. Since only the events in bit 4 can be masked, only bit 4 is used in this register.

Attribute: RW

Security: Non-secure

Bit field description:

* **EVC1\_OFLW\_MASK** [6] -   
  1'b1: When is set to 1, the corresponding interrupt event will not send an interrupt to the system.  
  1'b0: The corresponding interrupt event will send an interrupt to the system.
* **PARITY\_ERR\_MASK** [5] - Interrupt mask for register parity error.
* **EVC\_OFLW\_MASK** [4] -   
  1'b1: When is set to 1, the corresponding interrupt event will not send an interrupt to the system.  
  1'b0: The corresponding interrupt event will send an interrupt to the system.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | EVC1\_OFLW\_MASK | PARITY\_ERR\_MASK | EVC\_OFLW\_MASK | u | | | |

Table 27 RXEM register

### RXID – Receive Bridge ID

This register holds a unique 8-bit identifier for the receiving bridge. It is a read-only register. It can be used for debugging software access to the NoC elements by confirming that a read has successfully targeted the correct NoC element.

Attribute: R

Security: Non-secure

Bit field description:

* **ZEROES** [15:8] - Forced to zero
* **ID** [7:0] - A unique 8-bit identifier assigned to the bridge to uniquely identify it on the NoC. It is equal to the corresponding TXID 8-bit identifier on the Tx side of the bridge.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | | | | | | | | | | | ZEROES | | | | | | | | ID | | | | | | | |

Table 28 RXID register

### TXE – Transmit Event

This register tracks error or interrupt conditions. It resets to 0, but as these conditions occur, the corresponding bits are set to 1. This register can be read and can also be cleared by sending a write with bits set to 0 for the bits that should be cleared. This register works in combination with the Transmit Interrupt Mask register to determine when an interrupt is transmitted.

Attribute: WZC

Security: Non-secure

Bit field description:

* **PARITY\_ERR** [8] - Register parity error interrupt.
* **FIFO\_OVERFLOW\_D** [7] - Host interface FIFO D overflow. Indicates that one of the per-interface FIFOs at the transmitting bridge to NoC has overflowed. This event will always trigger an interrupt and cannot be masked
* **FIFO\_OVERFLOW\_C** [6] - Host interface FIFO C overflow. Indicates that one of the per-interface FIFOs at the transmitting bridge to NoC has overflowed. This event will always trigger an interrupt and cannot be masked
* **FIFO\_OVERFLOW\_B** [5] - Host interface FIFO B overflow. Indicates that one of the per-interface FIFOs at the transmitting bridge to NoC has overflowed. This event will always trigger an interrupt and cannot be masked
* **FIFO\_OVERFLOW\_A** [4] - Host interface FIFO A overflow. Indicates that one of the per-interface FIFOs at the transmitting bridge to NoC has overflowed. This event will always trigger an interrupt and cannot be masked
* **EVENT\_CNTR\_OVERFLOW** [3] -   
  1'b1: Sets if the event counter overflows, this event can be masked so that no interrupt is sent on an overflow condition
* **TRANS\_ILLEGAL\_DEST\_QOS** [2] -   
  1'b1: Sets if a transaction is received from bridge for which there is no entry present in the vcmap, i.e. the destination and/or QoS is not supported, this is a decode error. This event can be masked to not send an interrupt, but the packet will be dropped in the bridge.
* **SOP\_AFTER\_SOP** [1] -   
  1'b1: Sets if a SOP is received after SOP, this event will always trigger an interrupt and cannot be masked.
* **TRANS\_WITHOUT\_SOP** [0] -   
  1'b1: Sets if a transaction is initiated w/o SOP, this event will always trigger an interrupt and cannot be masked.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | | | | | | | | | | | | | | | | | | PARITY\_ERR | FIFO\_OVERFLOW\_D | FIFO\_OVERFLOW\_C | FIFO\_OVERFLOW\_B | FIFO\_OVERFLOW\_A | EVENT\_CNTR\_OVERFLOW | TRANS\_ILLEGAL\_DEST\_QOS | SOP\_AFTER\_SOP | TRANS\_WITHOUT\_SOP |

Table 29 TXE register

### TXEM – Transmit Event Mask

This register is used to decide which of the error/interrupt events specified in the Transmit Interrupt Status register should trigger an interrupt. Since only the events in bit 2 and 3 can be masked, only bit 2 and 3 are used in this register. When one of the bits in this register is set to 1, the corresponding interrupt event will not send an interrupt to the system.

Attribute: RW

Security: Non-secure

Bit field description:

* **PARITY\_ERR\_MASK** [8] - Interrupt mask for register parity error
* **EVENT\_CNTR\_OVERFLOW** [3] - Interrupt mask for event counter overflow
* **TRANS\_ILLEGAL\_DEST\_QOS** [2] - Interrupt mask for illegal destination QoS

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | | | | | | | | | | | | | | | | | | PARITY\_ERR\_MASK | u | | | | EVENT\_CNTR\_OVERFLOW | TRANS\_ILLEGAL\_DEST\_QOS | u | |

Table 30 TXEM register

### TXID – Transmit Bridge ID

This register holds a unique 8-bit identifier for the transmitting bridge. It is a read-only register. It can be used for debugging software access to the NoC elements by confirming that a read has successfully targeted the correct NoC element.

Attribute: R

Security: Non-secure

Bit field description:

* **ZEROES** [15:8] - Forced to zero
* **ID** [7:0] - A unique 8-bit identifier assigned to the bridge to uniquely identify it on the NoC. It is equal to the corresponding RXID 8-bit identifier on the Rx side of the bridge.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | | | | | | | | | | | ZEROES | | | | | | | | ID | | | | | | | |

Table 31 TXID register

## AMBA Master Bridge

### AM\_ADBASE

These registers specify the base addresses and masks of different slave ranges accessible from this master. One base, mask, and reloc register set per address range assigned to the master. These registers can be individually designated as read-only or read-write based on NocStudio property assigned to address ranges.

Even if the register is read-only, the range can be disabled using the appropriate bits described below which are always programmable. A slave address range is specified using the above base address and mask pair. An address on the AR or AW channel has a match against a range if it satisfies the equation

AxADDRS & AM\_ADMASK [i] == AM\_ADBASE [i]

Note that programmed 'base' must already factor the 'mask'. The base should not have a 1'b1 bit where the corresponding mask bit is 1'b0. What this means is that the programmed base should already have performed a bit-wise AND operation with the 'mask'. An address which doesn't match any range results in a decode error response. Note that programming of these registers must ensure that an address matches only against one range. Match against multiple ranges is a fatal error and will raise an interrupt.

Address ranges are specified at 64B cache line boundary. Lower six bits if AM\_ADBASE and AM\_ADMASK are used for specifying access permissions on an address range.

AM\_ADBASE [5:0]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **5** | **4** | **3** | **2** | **1** | **0** |
| X | DI | R/Wn | I | NS | P |

AM\_ADMASK [5:0]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **5** | **4** | **3** | **2** | **1** | **0** |
| X | X | Valid | I | NS | P |

Bits [2:0] act as value and mask for checking against AxPROT of an incoming command. A command is allowed access to a range if

AxPROT & AM\_ADMASK [2:0] == AM\_ADBASE [2:0] & AM\_ADMASK [2:0]

If the above check fails, then the command is denied access to the range and decode error response is returned. The encoding is specified below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **AM\_ADBASE [4] Disable** | **AM\_ADMASK [3] RD/WRn valid** | **AM\_ADBASE [3] RD/WRn** |  | **Interpretation** |
| 1 | X | X |  | range disabled |
| 0 | 1 | 1 |  | Read only |
| 0 | 1 | 0 |  | Write only |
| 0 | 0 | X |  | read/write |

Attribute: RW

Security: Secure access only

Bit field description:

* **BASE\_ADDRESS** [63:6] - Base address
* **LLC** [5] - LLC disable
* **DI** [4] -   
  1'b1: Address range disabled
* **R\_Wn** [3] -   
  1'b1: Read enabled to range  
  1'b0: Write enabled to range
* **I** [2] -   
  1'b1: Instruction
* **NS** [1] -   
  1'b1: Non-secure
* **P** [0] -   
  1'b1: Privileged

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| BASE\_ADDRESS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BASE\_ADDRESS | | | | | | | | | | | | | | | | | | | | | | | | | | LLC | DI | R\_Wn | I | NS | P |

Table 32 AM\_ADBASE register

### AM\_ADMASK

See AM\_ADBASE.

Attribute: RW

Security: Secure access only

Bit field description:

* **MASK** [63:6] - Mask
* **RSV** [5] - Reserved
* **TM** [4] -   
  1'b1: Enable Trusted Master behavior for secure transactions
* **VAL** [3] -   
  1'b1: R\_Wn field is valid
* **I** [2] -   
  1'b1: Instruction field is valid
* **NS** [1] -   
  1'b1: Non-secure field is valid
* **P** [0] -   
  1'b1: Privileged field is valid

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| MASK | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MASK | | | | | | | | | | | | | | | | | | | | | | | | | | RSV | TM | VAL | I | NS | P |

Table 33 AM\_ADMASK register

### AM\_ADRELOCSLV

Register used to relocate a master address to slave address.

Attribute: R

Security: Secure access only

Bit field description:

* **SLV\_RELOC** [31:6] - Slave Reloc

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SLV\_RELOC | | | | | | | | | | | | | | | | | | | | | | | | | | u | | | | | |

Table 34 AM\_ADRELOCSLV register

### AM\_ADRELOCSYS

Register used to relocate a master address to system address.

Attribute: R

Security: Secure access only

Bit field description:

* **SYS\_RELOC** [31:6] - System Reloc

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SYS\_RELOC | | | | | | | | | | | | | | | | | | | | | | | | | | u | | | | | |

Table 35 AM\_ADRELOCSYS register

### AM\_AROVRD

AR override.

Attribute: RW

Security: Secure access only

Bit field description:

* **arqos\_enb** [23:20] - 1'b1 indicates bit positions where ARQOS value is overridden. 1'b0 indicates bit positions where ARQOS is unchanged.
* **arqos\_val** [19:16] - Value to override incoming ARQOS
* **arprot\_enb** [14:12] - 1'b1 indicates bit positions where ARPROT value is overridden. 1'b0 indicates bit positions where ARPROT is unchanged.
* **arprot\_val** [10:8] - Value to override incoming ARPROT
* **arcache\_enb** [7:4] - 1'b1 indicates bit positions where ARCACHE value is overridden. 1'b0 indicates bit positions where ARCACHE is unchanged.
* **arcache\_val** [3:0] - Value to override incoming ARCACHE

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | | | arqos\_enb | | | | arqos\_val | | | | u | arprot\_enb | | | u | arprot\_val | | | arcache\_enb | | | | arcache\_val | | | |

Table 36 AM\_AROVRD register

### AM\_AWOVRD

AW override.

Attribute: RW

Security: Secure access only

Bit field description:

* **awqos\_enb** [23:20] - 1'b1 indicates bit positions where AWQOS value is overridden. 1'b0 indicates bit positions where AWQOS is unchanged.
* **awqos\_val** [19:16] - Value to override incoming AWQOS
* **awprot\_enb** [14:12] - 1'b1 indicates bit positions where AWPROT value is overridden. 1'b0 indicates bit positions where AWPROT is unchanged.
* **awprot\_val** [10:8] - Value to override incoming AWPROT
* **awcache\_enb** [7:4] - 1'b1 indicates bit positions where AWCACHE value is overridden. 1'b0 indicates bit positions where AWCACHE is unchanged.
* **awcache\_val** [3:0] - Value to override incoming AWCACHE

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | | | awqos\_enb | | | | awqos\_val | | | | u | awprot\_enb | | | u | awprot\_val | | | awcache\_enb | | | | awcache\_val | | | |

Table 37 AM\_AWOVRD register

### AM\_BRIDGE\_ID

Unique identifier assigned to the master bridge.

Attribute: R

Security: Non-secure

Bit field description:

* **ZEROES** [15:8] - Forced to zero
* **ID** [7:0] - Unique bridge ID

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | | | | | | | | | | | ZEROES | | | | | | | | ID | | | | | | | |

Table 38 AM\_BRIDGE\_ID register

### AM\_CADDR

This register is part of statistics gathering on the AR and AW command channels. This is the address value which is checked against AR, AW command channels in conjunction with the mask below to filter commands for statistics gathering.

Attribute: RW

Security: Non-secure

Bit field description:

* **CAPTURE\_ADDR** [63:0] - Capture address

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| CAPTURE\_ADDR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CAPTURE\_ADDR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Table 39 AM\_CADDR register

### AM\_CADDRMSK

If command address on the AR, AW channel logically ANDed with this mask is equal to the value specified in AM\_CADDR, then an address match has occurred. Note that only lowest significant bits equal to the master's address width are used in the comparison.

Attribute: RW

Security: Non-secure

Bit field description:

* **CAPTURE\_ADDR\_MASK** [63:0] - Capture address mask

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| CAPTURE\_ADDR\_MASK | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CAPTURE\_ADDR\_MASK | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Table 40 AM\_CADDRMSK register

### AM\_CCMD0

Values of command fields/pins that are compared against AR, AW, R, W channel interface signals to filter commands/events for statistics gathering. Two selections can be made for statistics gathering, counting filtered commands or measuring latency of filtered commands.

Attribute: RW

Security: Non-secure

Bit field description:

* **TYP** [32] -   
  1'b1: Count captured command  
  1'b0: Count response latency of captured command
* **INTFID** [30:28] -   
  001: AW (can count captured event or response latency)  
  000: AR (can count captured event or response latency)
* **VAL** [25] -   
  1'b1: Valid
* **RDY** [24] -   
  1'b1: Ready
* **LOC** [23] -   
  1'b1: Lock
* **PROT** [22:20] - Prot
* **QOS** [19:16] - QoS
* **CACHE** [15:12] - Cache
* **BAR** [9:8] - Bar
* **DOMAIN** [5:4] - Domain
* **SNOOP** [3:0] - Snoop

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | TYP |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | INTFID | | | u | | VAL | RDY | LOC | PROT | | | QOS | | | | CACHE | | | | u | | BAR | | u | | DOMAIN | | SNOOP | | | |

Table 41 AM\_CCMD0 register

### AM\_CCMD1

Values of command fields/pins that are compared against AR, AW, R, W channel interface signals to filter commands/events for statistics gathering. Two selections can be made for statistics gathering, counting filtered commands or measuring latency of filtered commands.

Attribute: RW

Security: Non-secure

Bit field description:

* **TYP** [32] -   
  1'b1: Count response latency of captured command  
  1'b0: Count captured command
* **INTFID** [30:28] -   
  001: AW (can count captured event or response latency)  
  000: AR (can count captured event or response latency)
* **VAL** [25] -   
  1'b1: Valid
* **RDY** [24] -   
  1'b1: Ready
* **LOC** [23] -   
  1'b1: Lock
* **PROT** [22:20] - Prot
* **QOS** [19:16] - QoS
* **CACHE** [15:12] - Cache
* **BAR** [9:8] - Bar
* **DOMAIN** [5:4] - Domain
* **SNOOP** [3:0] - Snoop

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | TYP |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | INTFID | | | u | | VAL | RDY | LOC | PROT | | | QOS | | | | CACHE | | | | u | | BAR | | u | | DOMAIN | | SNOOP | | | |

Table 42 AM\_CCMD1 register

### AM\_CCMDMSK0

If Command fields on AR, AW channel logically ANDed with this mask are equal to the corresponding command field values in AM\_CCMD0 then a command match has occurred. Address and command value match occurring together constitute events for the statistics counters.

Attribute: RW

Security: Non-secure

Bit field description:

* **VAL** [25] -   
  1'b1: Valid
* **RDY** [24] -   
  1'b1: Ready
* **LOC** [23] -   
  1'b1: Lock
* **PROT** [22:20] - Prot
* **QOS** [19:16] - QoS
* **CACHE** [15:12] - Cache
* **BAR** [9:8] - Bar
* **DOMAIN** [5:4] - Domain
* **SNOOP** [3:0] - Snoop

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | VAL | RDY | LOC | PROT | | | QOS | | | | CACHE | | | | u | | BAR | | u | | DOMAIN | | SNOOP | | | |

Table 43 AM\_CCMDMSK0 register

### AM\_CCMDMSK1

If Command fields on AR, AW channel logically ANDed with this mask are equal to the corresponding command field values in AM\_CCMD0 then a command match has occurred. Address and command value match occurring together constitute events for the statistics counters.

Attribute: RW

Security: Non-secure

Bit field description:

* **VAL** [25] -   
  1'b1: Valid
* **RDY** [24] -   
  1'b1: Ready
* **LOC** [23] -   
  1'b1: Lock
* **PROT** [22:20] - Prot
* **QOS** [19:16] - QoS
* **CACHE** [15:12] - Cache
* **BAR** [9:8] - Bar
* **DOMAIN** [5:4] - Domain
* **SNOOP** [3:0] - Snoop

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | VAL | RDY | LOC | PROT | | | QOS | | | | CACHE | | | | u | | BAR | | u | | DOMAIN | | SNOOP | | | |

Table 44 AM\_CCMDMSK1 register

### AM\_CFG

Configures the master bridge's support for autowake of power domains.

When set, master bridge halts a request and issues wakeup requests for power domains that need to powered up to complete the transaction. The power domains should support auto wake. When reset, master bridge issues DECERR for any transaction which has dependent power domains in sleep state.

Attribute: RW

Security: Non-secure

Bit field description:

* **AW** [0] -   
  1'b1: Autowake enabled  
  1'b0: Autowake disabled

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | AW |

Table 45 AM\_CFG register.

### AM\_CGC

Programmable interval used by coarse clock gating logic in master bridge. This interval is used to generate heart beat pulses using noc\_clk on that bridge. These heart beat pulses are broadcast to each local clock gating domain within the bridge where they are synchronized to the CG domain's clock. Four consecutive heart beat pulses in the CG domain is used as the inactivity/idle interval to initiate coarse clock gating of the CG domain.

Attribute: RW

Security: Non-secure

Bit field description:

* **HYSTERESIS\_COUNTER** [31:0] - Hysteresis counter

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| HYSTERESIS\_COUNTER | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Table 46 AM\_CGC register

### AM\_CGO

Fast path override, when set to 1'b1 will cause the clock gating logic to be disabled. 1'b1 will allow activity-based clock gating to be performed on the master bridge.

Attribute: RW

Security: Non-secure

Bit field description:

* **FPO** [0] -   
  1'b1: Clock gating enabled  
  1'b0: Clock gating is disabled

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | FPO |

Table 47 AM\_CGO register.

### AM\_CNTR0

32-bit counter which is used to count the captured statistics events. This counter can hold the count of commands filtered on the AR, AW channels. When measuring command latency, this counter holds the denominator or sum of number of cycles between command and response for multiple commands over which latency is measured.

Attribute: RW

Security: Non-secure

Bit field description:

* **CNTR** [31:0] - Counter

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CNTR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Table 48 AM\_CNTR0 register

### AM\_CNTR1

32-bit counter which is used to count the captured statistics events. This counter can hold the count of commands filtered on the AR, AW channels. When measuring command latency, this counter holds the denominator or sum of number of cycles between command and response for multiple commands over which latency is measured.

Attribute: RW

Security: Non-secure

Bit field description:

* **CNTR** [31:0] - Counter

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CNTR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Table 49 AM\_CNTR1 register.

### AM\_ERA

This is the address on AR channel for which a decode error was detected. This corresponds to the status register bit e0 in AM\_ERR.

Attribute: R

Security: Non-secure

Bit field description:

* **READ\_DECERR\_ADDRS** [63:0] - Read decerr address

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| READ\_DECERR\_ADDRS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| READ\_DECERR\_ADDRS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Table 50 AM\_ERA register

### AM\_ERR

These error status bits record the first error event and have to be cleared by writing a 1'b0 before new errors are recorded.

Attribute: WZC

Security: Non-secure

Bit field description:

* **E55** [55] -   
  1'b1: [FATAL] R Channel Cpkt Fifo Parity Error
* **E54** [54] -   
  1'b1: [FATAL] CRCD Channel Crid Fifo Parity Error
* **E53** [53] -   
  1'b1: [FATAL] Ack Channel Rack Fifo Parity Error
* **E52** [52] -   
  1'b1: [FATAL] Ack Channel Wack Fifo Parity Error
* **E51** [51] -   
  1'b1: [FATAL] Rx Fifo Parity Err
* **E50** [50] -   
  1'b1: [FATAL] Write Reorder Buffer Parity Err
* **E49** [49] -   
  1'b1: [FATAL] Read Reorder Buffer Parity Err
* **E48** [48] -   
  1'b1: [FATAL] Widtbl Entry Parity Err
* **E47** [47] -   
  1'b1: [FATAL] Ridtbl Entry Parity Err
* **E46** [46] -   
  1'b1: [FATAL] CDDATA Parity Err
* **E45** [45] -   
  1'b1: [FATAL] WDATA Parity Err
* **E44** [44] -   
  1'b1: [FATAL] AWADDR Parity Err
* **E43** [43] -   
  1'b1: [FATAL] AW Parity Err
* **E42** [42] -   
  1'b1: [FATAL] ARADDR Parity Err
* **E41** [41] -   
  1'b1: [FATAL] AR Parity Err
* **E40** [40] -   
  1'b1: [FATAL] Indicates that portcheck detected error (SIB mode only)
* **E35** [35] -   
  1'b1: [FATAL] Parity error in configuration/status registers
* **E34** [34] -   
  1'b1: [FATAL] Traffic sent to a noc layer which is power gate
* **E33** [33] -   
  1'b1: Capture counter1 overflow
* **E32** [32] -   
  1'b1: Capture counter0 overflow
* **E24** [24] -   
  1'b1: [FATAL] Unexpected narrow write detected
* **E23** [23] -   
  1'b1: [FATAL] Write WRAP not equal to supported cacheline size
* **E22** [22] -   
  1'b1: Write response timeout
* **E21** [21] -   
  1'b1: [FATAL] Write address multi-hit
* **E20** [20] -   
  1'b1: [FATAL] Write exclusive split
* **E19** [19] -   
  1'b1: Non-modifiable WRAP
* **E18** [18] -   
  1'b1: Write slave error
* **E17** [17] -   
  1'b1: Write address decode error from slave
* **E16** [16] -   
  1'b1: Local write address decode error
* **E8** [8] -   
  1'b1: [FATAL] Unexpected narrow read detected
* **E7** [7] -   
  1'b1: [FATAL] Read WRAP not equal to supported cacheline size: A WRAP command of unsupported cache line size was detected
* **E6** [6] -   
  1'b1: Read response timeout: Read response timeout occurred. With timeout enabled, a response wasn't received within the expected interval
* **E5** [5] -   
  1'b1: [FATAL] Read address multi-hit: An AR command matched against multiple entries in the address table
* **E4** [4] -   
  1'b1: [FATAL] Read exclusive split: An AR command of FIXED burst type was detected
* **E3** [3] -   
  1'b1: Non-modifiable WRAP: A WRAP command marked as non-modifiable (ARCACHE [0] =0) was detected
* **E2** [2] -   
  1'b1: Read slave error: A slave error response was received from a slave device
* **E1** [1] -   
  1'b1: Read address decode error from slave: A decode error response was received from a slave device
* **E0** [0] -   
  1'b1: Local read address decode error: ARADDR did not find a match in the master bridges address table and a decode error was issued

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | E55 | E54 | E53 | E52 | E51 | E50 | E49 | E48 | E47 | E46 | E45 | E44 | E43 | E42 | E41 | E40 | u | | | | E35 | E34 | E33 | E32 |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | | E24 | E23 | E22 | E21 | E20 | E19 | E18 | E17 | E16 | u | | | | | | | E8 | E7 | E6 | E5 | E4 | E3 | E2 | E1 | E0 |

Table 51 AM\_ERR register

### AM\_EWA

This is the address on AW channel for which a decode error was detected. This corresponds to the status register bit e16 in AM\_ERR.

Attribute: R

Security: Non-secure

Bit field description:

* **WRITE\_DECERR\_ADDRS** [63:0] - Write decerr address

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| WRITE\_DECERR\_ADDRS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| WRITE\_DECERR\_ADDRS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Table 52 AM\_EWA register

### AM\_FSPEI

Flop Structure Parity Error Injection. This register is readable/writeable by software so hardware can inject errors.

Attribute: RW

Security: Secure access only

Bit field description:

* **R\_CPKT\_FIFO\_ERR\_INJ** [46] -   
  1'b1: Inject parity error in R channel cpkt fifo.
* **CRID\_FIFO\_ERR\_INJ** [45] -   
  1'b1: Inject parity error in CRCD channel CRID fifo.
* **RACK\_FIFO\_ERR\_INJ** [44] -   
  1'b1: Inject parity error in ACK channel RACK fifo.
* **WACK\_FIFO\_ERR\_INJ** [43] -   
  1'b1: Inject parity error in ACK channel WACK fifo.
* **RXFIFO\_ERR\_INJ** [42] -   
  1'b1: Inject parity error in rx fifo specified by RXFIFO\_LAYER and RXFIFO\_VC.
* **RXFIFO\_LAYER** [41:38] - Rx Fifo Virtual Channel to force parity error on
* **RXFIFO\_VC** [37:36] - Rx Fifo Virtual Channel to force parity error on
* **WROB\_ERR\_INJ** [35] -   
  1'b1: Inject parity error in wrob entry specified by WROB\_ENTRY
* **WROB\_ENTRY** [34:27] - Write Response Buffer Entry to force parity error on
* **RROB\_ERR\_INJ** [26] -   
  1'b1: Inject parity error in rrob entry specified by RROB\_ENTRY
* **RROB\_ENTRY** [25:18] - Read Response Buffer Entry to force parity error on
* **WIDTBL\_ERR\_INJ** [17] -   
  1'b1: Inject parity error in widtbl entry specified by WIDTBL\_ENTRY
* **WIDTBL\_ENTRY** [16:9] - Write Ch Aidtbl Entry to force parity error on
* **RIDTBL\_ERR\_INJ** [8] -   
  1'b1: Inject parity error in ridtbl entry specified by RIDTBL\_ENTRY
* **RIDTBL\_ENTRY** [7:0] - Read Ch Aidtbl Entry to force parity error on

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | | | | | | | | | | R\_CPKT\_FIFO\_ERR\_INJ | CRID\_FIFO\_ERR\_INJ | RACK\_FIFO\_ERR\_INJ | WACK\_FIFO\_ERR\_INJ | RXFIFO\_ERR\_INJ | RXFIFO\_LAYER | | | | RXFIFO\_VC | | WROB\_ERR\_INJ | WROB\_ENTRY | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| WROB\_ENTRY | | | | | RROB\_ERR\_INJ | RROB\_ENTRY | | | | | | | | WIDTBL\_ERR\_INJ | WIDTBL\_ENTRY | | | | | | | | RIDTBL\_ERR\_INJ | RIDTBL\_ENTRY | | | | | | | |

Table 53 AM\_FSPEI register

### AM\_FSPEL

Error logging register for flop structure parity errors. Bits 7:0 log the ridtbl entry number for the last ridtbl parity error, corresponding to AM\_ERR bit 47. Bits 15:8 log the widtbl entry number for the last widtbl parity error, corresponding to AM\_ERR bit 48. Bits 23:16 log the rrob entry for the last read reorder buffer parity error, corresponding to AM\_ERR bit 49, and bits 31:24 log the wrob entry for the last write reorder buffer parity error, corresponding to AM\_ERR bit 50. Bits 43 to 45 provide fault information for miscellaneous flop structures on the master bridge. This register is writeable only by hw, readable only by software.

Attribute: R

Security: Non-secure

Bit field description:

* **RXFIFO\_PARITY\_ERR\_LAYER** [37:34] - Rx Fifo Parity Error Layer
* **RXFIFO\_PARITY\_ERR\_VC** [33:32] - Rx Fifo Parity Error Virtual Channel
* **WROB\_PARITY\_ERR\_ENTRY** [31:24] - Write Response Buffer Parity Error Entry
* **RROB\_PARITY\_ERR\_ENTRY** [23:16] - Read Response Buffer Parity Error Entry
* **WIDTBL\_PARITY\_ERR\_ENTRY** [15:8] - Write Aidtbl Parity Error Entry
* **RIDTBL\_PARITY\_ERR\_ENTRY** [7:0] - Read Aidtbl Parity Error Entry

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | RXFIFO\_PARITY\_ERR\_LAYER | | | | RXFIFO\_PARITY\_ERR\_VC | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| WROB\_PARITY\_ERR\_ENTRY | | | | | | | | RROB\_PARITY\_ERR\_ENTRY | | | | | | | | WIDTBL\_PARITY\_ERR\_ENTRY | | | | | | | | RIDTBL\_PARITY\_ERR\_ENTRY | | | | | | | |

Table 54 AM\_FSPEL register

### AM\_HASH\_FUNC

These registers are used for programmable hash functions. They are the size of the SYSTEM address width, minus the 6-bit offset bits. Any reprogramming of these values can require the slave to understand the new hashing function in order to successfully compress the address space.

Attribute: RW

Security: Secure access only

Bit field description:

* **HASH** [31:6] - Hash bits
* **F** [0] -   
  1'b1: Force hash bit. When hash function register is programmed to 0, this will cause the corresponding hash bit to be forced to 1'b1. A non-zero hash function will cause inversion of the corresponding hash bit.  
  1'b0: Default. When hash function register is programmed to 0, the corresponding hash bit will be 1'b0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| HASH | | | | | | | | | | | | | | | | | | | | | | | | | | u | | | | | F |

Table 55 AM\_HASH\_FUNC register

### AM\_INTM

Interrupt mask register. Individual bit position matches the error bit positions in AM\_ERR. When an INTM bit is set, occurrence of the corresponding error event will not cause an interrupt to be raised. When 1'b0, error event will cause interrupt to be raised.

Attribute: RW

Security: Non-secure

Bit field description:

* **E55** [55] -   
  1'b1: R Channel Cpkt Fifo Parity Intr Mask
* **E54** [54] -   
  1'b1: CRCD Channel Crid Fifo Parity Intr Mask
* **E53** [53] -   
  1'b1: Ack Channel Rack Fifo Parity Intr Mask
* **E52** [52] -   
  1'b1: Ack Channel Wack Fifo Parity Intr Mask
* **E51** [51] -   
  1'b1: Rx Fifo Parity Intr Mask
* **E50** [50] -   
  1'b1: Write Reorder Buffer Parity Intr Mask
* **E49** [49] -   
  1'b1: Read Reorder Buffer Parity Intr Mask
* **E48** [48] -   
  1'b1: Widtbl Parity Intr Mask
* **E47** [47] -   
  1'b1: Ridtbl Parity Intr Mask
* **M46** [46] -   
  1'b1: CDDATA Parity Intr Mask
* **M45** [45] -   
  1'b1: WDATA Parity Intr Mask
* **M44** [44] -   
  1'b1: AWADDR Parity Intr Mask
* **M43** [43] -   
  1'b1: AW Parity Intr Mask
* **M42** [42] -   
  1'b1: ARADDR Parity Intr Mask
* **M41** [41] -   
  1'b1: AR Parity Intr Mask
* **M40** [40] -   
  1'b1: Mask interrupt for SIB portcheck error (SIB mode only)
* **M35** [35] -   
  1'b1: Mask interrupt on csr parity errors
* **M34** [34] -   
  1'b1: Mask interrupt on traffic to PG layer
* **M33** [33] -   
  1'b1: Counter 1 overflow interrupt mask
* **M32** [32] -   
  1'b1: Counter 0 overflow interrupt mask
* **M24** [24] -   
  1'b1: Mask interrupt for write channel
* **M23** [23] -   
  1'b1: Mask interrupt for write channel
* **M22** [22] -   
  1'b1: Mask interrupt for write channel
* **M21** [21] -   
  1'b1: Mask interrupt for write channel
* **M20** [20] -   
  1'b1: Mask interrupt for write channel
* **M19** [19] -   
  1'b1: Mask interrupt for write channel
* **M18** [18] -   
  1'b1: Mask interrupt for write channel
* **M17** [17] -   
  1'b1: Mask interrupt for write channel
* **M16** [16] -   
  1'b1: Mask interrupt for write channel
* **M8** [8] -   
  1'b1: Mask interrupt for read channel
* **M7** [7] -   
  1'b1: Mask interrupt for read channel
* **M6** [6] -   
  1'b1: Mask interrupt for read channel
* **M5** [5] -   
  1'b1: Mask interrupt for read channel
* **M4** [4] -   
  1'b1: Mask interrupt for read channel
* **M3** [3] -   
  1'b1: Mask interrupt for read channel
* **M2** [2] -   
  1'b1: Mask interrupt for read channel
* **M1** [1] -   
  1'b1: Mask interrupt for read channel
* **M0** [0] -   
  1'b1: Mask interrupt for read channel

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | E55 | E54 | E53 | E52 | E51 | E50 | E49 | E48 | E47 | M46 | M45 | M44 | M43 | M42 | M41 | M40 | u | | | | M35 | M34 | M33 | M32 |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | | M24 | M23 | M22 | M21 | M20 | M19 | M18 | M17 | M16 | u | | | | | | | M8 | M7 | M6 | M5 | M4 | M3 | M2 | M1 | M0 |

Table 56 AM\_INTM register

### AM\_LATNUM0

This register is programmed with the number of commands over which latency is to be measured. When this register counts down to 0, latency measurement is complete and average latency can be computed using:

Average command latency = Value in AM\_CNTR0/Value which was programmed in AM\_LATNUM0

There are two sets of counters available for gathering statistics. AM\_CCMD1, AM\_CCMDMSK1, AM\_CNTR1, AM\_LATNUM1 constitute the second bank of counters and are similar to the above set.

Attribute: RW

Security: Non-secure

Bit field description:

* **CNTR** [31:0] - Counter

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CNTR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Table 57 AM\_LATNUM0 register

### AM\_LATNUM1

This register is programmed with the number of commands over which latency is to be measured. When this register counts down to 0, latency measurement is complete and average latency can be computed using:

Average command latency = Value in AM\_CNTR0/Value which was programmed in AM\_LATNUM0

There are two sets of counters available for gathering statistics. AM\_CCMD1, AM\_CCMDMSK1, AM\_CNTR1, AM\_LATNUM1 constitute the second bank of counters and are similar to the above set.

Attribute: RW

Security: Non-secure

Bit field description:

* **CNTR** [31:0] - Counter

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CNTR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Table 58 AM\_LATNUM1 register

### AM\_OSSLV

This register is used to check if there are any outstanding read/write commands to a slave specified by field slvid. NocStudio provides a table of slvids corresponding to the slave ports accessible from a master bridge. Outstanding status is reflected in AM\_STS.

Attribute: RW

Security: Non-secure

Bit field description:

* **SLVID** [15:0] - A slave ID associated with the current master for command outstanding status

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | | | | | | | | | | | SLVID | | | | | | | | | | | | | | | |

Table 59 AM\_OSSLV register

### AM\_SIB\_PORT\_DIS\_[sib\_name]\_M

Indicates which of the SIB ports are disabled. When a port is disabled, all output signals are driven low, and all input signals are ignored.

Attribute: RW

Security: Secure access only

Bit field description:

* **Port\_disable**[x] - Enable/Disable bit for connected master bridge on SIB. 1 means disabled and 0 means enabled.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | | | | | | | | | | | port disable | | | | | | | | | | | | | | | |

Table 60 AM\_SIB\_PORT\_DIS\_[sib\_name]\_M register.

### AM\_STS

When reordering is disabled on the master bridge, hazard stall occurs if the master tries to access a new slave device while response from a different slave is outstanding on the same AID.

This is because the responses can arrive out of order and the bridge is not equipped to correct the order. Without re-order buffers, hazard stalls also occur if a new large command needs to be split while there are older commands outstanding, or a large command just finished sending all its split segments but all responses have not returned yet.

When reordering is enabled, stall due to hazard occurs if a new command arrives, whose NoC QoS is different from the NoC QoS of commands outstanding on that AID.

Attribute: R

Security: Non-secure

Bit field description:

* **AWO** [7] -   
  1'b1: Write commands are outstanding to the slave specified in OSSLV register
* **ARO** [6] -   
  1'b1: Read commands are outstanding to the slave specified in OSSLV register
* **AWS** [5] -   
  1'b1: AW channel is stalled on hazard
* **ARS** [4] -   
  1'b1: AR channel is stalled on hazard
* **WOE** [3] -   
  1'b1: There are no write commands outstanding from the attached master device
* **ROE** [2] -   
  1'b1: There are no read commands outstanding from the attached master device
* **WOF** [1] -   
  1'b1: Maximum supported number of write commands are outstanding waiting for response and no more requests can be accepted  
  1'b0: Master bridge can accept more write requests
* **ROF** [0] -   
  1'b1: Maximum supported number of read commands are outstanding waiting for response and no more requests can be accepted  
  1'b0: Master bridge can accept more read requests

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | | | | | | | | | | | | | | | | | | | AWO | ARO | AWS | ARS | WOE | ROE | WOF | ROF |

Table 61 AM\_STS register

### AM\_TOCFG

This register is used to configure response timeouts.

AM\_TOCFG [8] (En) needs to be set for timeout tracking to be enabled. When this bit is 1'b0, no timestamps are recorded to generate timeout interrupts. A 64-bit free running counter is used to time the response interval.

AM\_TOCFG [5:0] (TI) specifies the lower bit index into this counter, from where 2-bits are picked up and recorded as the arrival time stamp of every incoming AR and AW command. If response for a command does not return before the current time stamp rolls to arrival time stamp minus 1, the response is assumed to have timed-out and an interrupt is raised along with the slave ID to which the timed-out request was sent.

When changing the TI field, first write to the register with the En field cleared, then write a second time with the TI field to its new value, then a 3rd write to restore the En field to Enabled. During this update while the En field is cleared, existing timers will be cancelled, and new timer starts will be inhibited.

Attribute: RW

Security: Non-secure

Bit field description:

* **EN** [8] -   
  1'b1: Enabled timeout tracking, a 64-bit free running counter is used to time the response interval.  
  1'b0: No timestamps are recorded to generate timeout interrupts
* **TI** [5:0] - Timer index, index of a 64-bit counter from where timestamp is picked

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | | | | | | | | | | | | | | | | | | EN | u | | TI | | | | | |

Table 62 AM\_TOCFG register

### AM\_TOSLVID

AR slvid and AW slvid fields indicate slave IDs to which a read, write response timeout was detected.

Note that slvid encoding is not same as the bridge ID of the slave. NocStudio provides a table mapping the slvids to the actual slave ports accessible from the master bridge.

Attribute: R

Security: Non-secure

Bit field description:

* **AW\_SLVID** [31:16] - Slave ID of timed out AW request
* **AR\_SLVID** [15:0] - Slave ID of timed out AR request

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| AW\_SLVID | | | | | | | | | | | | | | | | AR\_SLVID | | | | | | | | | | | | | | | |

Table 63 AM\_TOSLVID register

## AMBA Slave Bridge

### AS\_AROVRD

AR Overrides.

Attribute: RW

Security: Secure access only

Bit field description:

* **arqos\_enb** [23:20] - 1'b1 indicates bit positions where ARQOS value is overridden. 1'b0 indicates bit positions where ARQOS is unchanged.
* **arqos\_val** [19:16] - Value to override incoming ARQOS
* **arprot\_enb** [14:12] - 1'b1 indicates bit positions where ARPROT value is overridden. 1'b0 indicates bit positions where ARPROT is unchanged.
* **arprot\_val** [10:8] - Value to override incoming ARPROT
* **arcache\_enb** [7:4] - 1'b1 indicates bit positions where ARCACHE value is overridden. 1'b0 indicates bit positions where ARCACHE is unchanged.
* **arcache\_val** [3:0] - Value to override incoming ARCACHE

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | | | arqos\_enb | | | | arqos\_val | | | | u | arprot\_enb | | | u | arprot\_val | | | arcache\_enb | | | | arcache\_val | | | |

Table 64 AS\_AROVRD register

### AS\_AWOVRD

AW Overrides.

Attribute: RW

Security: Secure access only

Bit field description:

* **awqos\_enb** [23:20] - 1'b1 indicates bit positions where AWQOS value is overridden. 1'b0 indicates bit positions where AWQOS is unchanged.
* **awqos\_val** [19:16] - Value to override incoming AWQOS
* **awprot\_enb** [14:12] - 1'b1 indicates bit positions where AWPROT value is overridden. 1'b0 indicates bit positions where AWPROT is unchanged.
* **awprot\_val** [10:8] - Value to override incoming AWPROT
* **awcache\_enb** [7:4] - 1'b1 indicates bit positions where AWCACHE value is overridden. 1'b0 indicates bit positions where AWCACHE is unchanged.
* **awcache\_val** [3:0] - Value to override incoming AWCACHE

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | | | awqos\_enb | | | | awqos\_val | | | | u | awprot\_enb | | | u | awprot\_val | | | awcache\_enb | | | | awcache\_val | | | |

Table 65 AS\_AWOVRD register

### AS\_BRIDGE\_ID

Unique identifier assigned to the slave bridge.

Attribute: R

Security: Non-secure

Bit field description:

* **ZEROES** [15:8] - Forced to zero
* **ID** [7:0] - Unique bridge ID

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | | | | | | | | | | | ZEROES | | | | | | | | ID | | | | | | | |

Table 66 AS\_BRIDGE\_ID register

### AS\_CCMD

Not applicable for current release.

Attribute: RW

Security: Non-secure

Bit field description:

* **intfid** [30:28] -   
  001: AW (can count captured event or response latency)  
  000: AR (can count captured event or response latency)
* **val** [25] -   
  1'b1: Valid
* **rdy** [24] -   
  1'b1: Ready

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | intfid | | | u | | val | rdy | u | | | | | | | | | | | | | | | | | | | | | | | |

Table 67 AS\_CCMD register

### AS\_CCMDMSK

Not applicable for current release.

Attribute: RW

Security: Non-secure

Bit field description:

* **val** [25] -   
  1'b1: Valid
* **rdy** [24] -   
  1'b1: Ready

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | val | rdy | u | | | | | | | | | | | | | | | | | | | | | | | |

Table 68 AS\_CCMDMSK register

### AS\_CGC

Programmable intervals used by coarse clock gating logic. This interval is used to generate heart beat pulses using noc\_clk on that bridge. These heart beat pulses are broadcast to each local clock gating domain within the bridge where they are synchronized to the CG domain's clock. Four consecutive heart beat pulses in the CG domain is used as the inactivity/idle interval to initiate coarse clock gating of the CG domain.

Attribute: RW

Security: Non-secure

Bit field description:

* **HYSTERESIS\_COUNTER** [31:0] - Hysteresis counter

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| HYSTERESIS\_COUNTER | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Table 69 AS\_CGC register

### AS\_CGO

Fast path override, when set to 1'b1 will cause the clock gating logic to be disabled. 1'b1 will allow activity-based clock gating to be performed on the slave bridge.

Attribute: RW

Security: Non-secure

Bit field description:

* **FPO** [0] -   
  1'b1: Clock gating enabled  
  1'b0: Clock gating is disabled

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | FPO |

Table 70 AS\_CGO register

### AS\_CNTR

Not applicable for current release.

Attribute: R

Security: Non-secure

Bit field description:

* **CNTR** [31:0] - Counter

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CNTR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Table 71 AS\_CNTR register

### AS\_ERR

These error status bits record the first error event and have to be cleared by writing a 1'b0 before new errors are recorded.

Attribute: WZC

Security: Non-secure

Bit field description:

* **E48** [48] -   
  1'b1: [FATAL] R Ch Deinterleaver Data Buffer Parity Err
* **E47** [47] -   
  1'b1: [FATAL] R Ch Deinterleaver Cmdtbl Parity Err
* **E46** [46] -   
  1'b1: [FATAL] R Ch Flush Fifo Parity Err
* **E45** [45] -   
  1'b1: [FATAL] B Ch Drain Fifo Parity Err
* **E44** [44] -   
  1'b1: [FATAL] Ack Ch Rack Reorder Buffer Parity Err
* **E43** [43] -   
  1'b1: [FATAL] Ack Ch Wack Reorder Buffer Parity Err
* **E42** [42] -   
  1'b1: [FATAL] CRCD Ch Reorder Buffer Parity Err
* **E41** [41] -   
  1'b1: [FATAL] Rx Fifo Parity Err
* **E40** [40] -   
  1'b1: [FATAL] B Ch Cmdtbl Parity Err
* **E39** [39] -   
  1'b1: [FATAL] R Ch Cmdtbl Parity Err
* **E38** [38] -   
  1'b1: [FATAL] ACADDR Parity error
* **E37** [37] -   
  1'b1: [FATAL] AC Parity error
* **E36** [36] -   
  1'b1: [FATAL] BRESP Parity error
* **E35** [35] -   
  1'b1: [FATAL] RRESP Parity error
* **E34** [34] -   
  1'b1: [FATAL] RDATA Parity error
* **E33** [33] -   
  1'b1: [FATAL] Parity error in config/status registers
* **E32** [32] -   
  1'b1: [FATAL] Traffic sent to a noc layer which is power gated
* **E19** [19] -   
  1'b1: Write command modified: A write command which was marked as non-modifiable was modified by the slave bridge
* **E18** [18] -   
  1'b1: [FATAL] Unknown write response destination: BID from write response produces a destination which is not present in the routing table
* **E17** [17] -   
  1'b1: Write slave error response: Slave error response received from slave device for write command
* **E16** [16] -   
  1'b1: Write decode error response: Decode error response received from slave device for write command
* **E4** [4] -   
  1'b1: Read command modified: A read command which was marked as non-modifiable was modified by the slave bridge
* **E3** [3] -   
  1'b1: [FATAL] Interleaved read response: Interleaved read response. This can occur if interleaved read response is received from a slave device for which a de-interleaver was not specified
* **E2** [2] -   
  1'b1: [FATAL] Unknown read response destination: RID from read response produces a destination which is not present in the routing table
* **E1** [1] -   
  1'b1: Read slave error response: Slave error response received from slave device for read command
* **E0** [0] -   
  1'b1: Read decode error response: Decode error response received from slave device for read command

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | | | | | | | | E48 | E47 | E46 | E45 | E44 | E43 | E42 | E41 | E40 | E39 | E38 | E37 | E36 | E35 | E34 | E33 | E32 |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | | | | | | | E19 | E18 | E17 | E16 | u | | | | | | | | | | | E4 | E3 | E2 | E1 | E0 |

Table 72 AS\_ERR register

### AS\_FSPEI

Error injection register for flop structure parity errors. Bits 7:0. This register is readable/writeable by sw, and used by hw to inject errors.

Attribute: RW

Security: Secure access only

Bit field description:

* **R\_CH\_DINT\_DATABUF\_ERR\_INJ** [39] -   
  1'b1: Inject parity error into R Ch Deinterleaver Data Buffer.
* **R\_CH\_DINT\_CMDTBL\_ERR\_INJ** [38] -   
  1'b1: Inject parity error into R Ch Deinterleaver Cmdtbl.
* **R\_CH\_DINT\_CMDTBL\_ENTRY** [37:30] - R Ch Deinterleaver Cmdtbl Entry to force parity error on
* **R\_CH\_FLUSH\_FIFO\_ERR\_INJ** [29] -   
  1'b1: Inject parity error into R Ch Flush Fifo.
* **B\_CH\_DRAIN\_FIFO\_ERR\_INJ** [28] -   
  1'b1: Inject parity error into B Ch Drain Fifo.
* **RACK\_ROB\_ENTRY\_ERR\_INJ** [27] -   
  1'b1: Inject parity error into ACK ch RACK ROB.
* **WACK\_ROB\_ENTRY\_ERR\_INJ** [26] -   
  1'b1: Inject parity error into ACK ch WACK ROB.
* **CRCD\_ROB\_ENTRY\_ERR\_INJ** [25] -   
  1'b1: Inject parity error into CRCD ch ROB.
* **RXFIFO\_ERR\_INJ** [24] -   
  1'b1: Inject parity error in rx fifo specified by RXFIFO\_LAYER and RXFIFO\_VC.
* **RXFIFO\_LAYER** [23:20] - Rx Fifo Virtual Channel to force parity error on
* **RXFIFO\_VC** [19:18] - Rx Fifo Virtual Channel to force parity error on
* **B\_CH\_CMDTBL\_ERR\_INJ** [17] -   
  1'b1: Inject parity error into B Ch Cmdtbl entry specified by B\_CH\_CMDTBL\_ENTRY
* **B\_CH\_CMDTBL\_ENTRY** [16:9] - B Ch Cmdtbl Entry to force parity error on
* **R\_CH\_CMDTBL\_ERR\_INJ** [8] -   
  1'b1: Inject parity error into R Ch Cmdtbl entry specified by R\_CH\_CMDTBL\_ENTRY
* **R\_CH\_CMDTBL\_ENTRY** [7:0] - R Ch Cmdtbl Entry to force parity error on

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | | | | | | | | | | | | | | | | | R\_CH\_DINT\_DATABUF\_ERR\_INJ | R\_CH\_DINT\_CMDTBL\_ERR\_INJ | R\_CH\_DINT\_CMDTBL\_ENTRY | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| R\_CH\_DINT\_CMDTBL\_ENTRY | | R\_CH\_FLUSH\_FIFO\_ERR\_INJ | B\_CH\_DRAIN\_FIFO\_ERR\_INJ | RACK\_ROB\_ENTRY\_ERR\_INJ | WACK\_ROB\_ENTRY\_ERR\_INJ | CRCD\_ROB\_ENTRY\_ERR\_INJ | RXFIFO\_ERR\_INJ | RXFIFO\_LAYER | | | | RXFIFO\_VC | | B\_CH\_CMDTBL\_ERR\_INJ | B\_CH\_CMDTBL\_ENTRY | | | | | | | | R\_CH\_CMDTBL\_ERR\_INJ | R\_CH\_CMDTBL\_ENTRY | | | | | | | |

Table 73 AS\_FSPEI register

### AS\_FSPEL

Error logging register for flop structure parity errors. This reg is written only by hw and read only by sw. It has meaning only if the corresponding bit in as\_err is set.

Attribute: R

Security: Non-secure

Bit field description:

* **RXFIFO\_PARITY\_ERR\_LAYER** [29:26] - Rx Fifo Parity Error Layer
* **RXFIFO\_PARITY\_ERR\_VC** [25:24] - Rx Fifo Parity Error Virtual Channel
* **R\_CH\_DINT\_CMDTBL\_PERR\_ENTRY** [23:16] - R Ch Deinterleaver Cmdtbl Parity Error Entry
* **B\_CH\_CMDTBL\_PERR\_ENTRY** [15:8] - B Ch Cmdtbl Parity Error Entry
* **R\_CH\_CMDTBL\_PARITY\_ERR\_ENTRY** [7:0] - R Ch Cmdtbl Parity Error Entry

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | RXFIFO\_PARITY\_ERR\_LAYER | | | | RXFIFO\_PARITY\_ERR\_VC | | R\_CH\_DINT\_CMDTBL\_PERR\_ENTRY | | | | | | | | B\_CH\_CMDTBL\_PERR\_ENTRY | | | | | | | | R\_CH\_CMDTBL\_PARITY\_ERR\_ENTRY | | | | | | | |

Table 74 AS\_FSPEL register.

### AS\_INTM

Interrupt mask register. Individual bit positions match the error bit positions in AS\_ERR. When an INTM bit is set, occurrence of the corresponding error event will not cause an interrupt to be raised. When 1'b0, error event will cause interrupt to be raised.

Attribute: RW

Security: Non-secure

Bit field description:

* **E47** [47] -   
  1'b1: R Ch Deinterleaver Cmdtbl Parity Err Intr Mask
* **E46** [46] -   
  1'b1: R Ch Flush Fifo Parity Err Intr Mask
* **E45** [45] -   
  1'b1: B Ch Drain Fifo Parity Err Intr Mask
* **E44** [44] -   
  1'b1: Ack Ch Rack Reorder Buffer Parity Err Intr Mask
* **E43** [43] -   
  1'b1: Ack Ch Wack Reorder Buffer Parity Err IntrMask
* **E42** [42] -   
  1'b1: CRCD Ch Reorder Buffer Parity Err Intr Mask
* **E41** [41] -   
  1'b1: Rx Fifo Parity Err Intr Mask
* **E40** [40] -   
  1'b1: B Ch Cmdtbl Parity Err Intr Mask
* **E39** [39] -   
  1'b1: R Ch Cmdtbl Parity Err Intr Mask
* **M38** [38] - ACADDR parity interrupt Mask
* **M37** [37] - AC parity interrupt Mask
* **M36** [36] - BRESP parity interrupt Mask
* **M35** [35] - RRESP parity interrupt Mask
* **M34** [34] - RDATA parity interrupt Mask
* **M33** [33] - Mask interrupt on csr parity errors
* **M32** [32] - Mask interrupt on traffic to PG layer
* **M19** [19] - Mask interrupts for write channel
* **M18** [18] - Mask interrupts for write channel
* **M17** [17] - Mask interrupts for write channel
* **M16** [16] - Mask interrupts for write channel
* **M4** [4] - Mask interrupts for read channel
* **M3** [3] - Mask interrupts for read channel
* **M2** [2] - Mask interrupts for read channel
* **M1** [1] - Mask interrupts for read channel
* **M0** [0] - Mask interrupts for read channel

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | | | | | | | | | E47 | E46 | E45 | E44 | E43 | E42 | E41 | E40 | E39 | M38 | M37 | M36 | M35 | M34 | M33 | M32 |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | | | | | | | M19 | M18 | M17 | M16 | u | | | | | | | | | | | M4 | M3 | M2 | M1 | M0 |

Table 75 AS\_INTM register

### AS\_STS

Slave bridge status bits.

Attribute: R

Security: Non-secure

Bit field description:

* **ROE** [3] -   
  1'b1: There are no read commands outstanding to the attached slave device
* **WOE** [2] -   
  1'b1: There are no write commands outstanding to the attached slave device
* **ROF** [1] -   
  1'b1: Maximum number of supported read commands are outstanding to the attached slave device awaiting response, no more read commands will be issued to slave till responses are received.  
  1'b0: Slave bridge can accept more read commands from the NoC
* **WOF** [0] -   
  1'b1: Maximum number of supported write commands are outstanding to the attached slave device awaiting response, no more write commands will be issued to slave till responses are received.  
  1'b0: Slave device can expect more write commands from NoC

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | ROE | WOE | ROF | WOF |

Table 76 AS\_STS register

## Regbus Registers

### AM\_NOCVER\_ID

Version identifier for the NoC. This read-only register is available only on the regbus master. This register is not available on pother master bridges and access will result in decode error response.

Attribute: R

Bit field description:

* **NOC\_VERSION\_ID** [31:0] - NoC version ID

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| NOC\_VERSION\_ID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Table 77 AM\_NOCVER\_ID register

## Protocol Converter Registers

AHB Bridge

### AHBM\_CTL

Control Register.

Attribute: RW

Bit field description:

* WNP [0] - 1'b1: Force non-posted writes.  
   - 1'b0: Not forced to non-posted writes.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | WNP |

Table 78 AHBM\_CTL register

### AHBM\_IM

Interrupt Mask Register.

Attribute: RW

Bit field description:

* EWRP [2] - 1'b1: Mask interrupt due to Illegal WRAP.
* WER [1] - 1'b1: Mask interrupt due to Write error.
* RER [0] - 1'b1: Mask interrupt due to Read error.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | EWRP | WER | RER |

Table 79 AHBM\_IM register

### AHBM\_STS

Status Register.

Attribute: WZC

Bit field description:

* EWRP [2] - 1'b1: Illegal WRAP detected.
* WER [1] - 1'b1: Write error detected.
* RER [0] - 1'b1: Read error detected.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | EWRP | WER | RER |

Table 80 AHBM\_STS register

### AHBS\_CTL

Control Register.

Attribute: RW

Bit field description:

* REG [16] - 1'b1: Issue error response on unaligned read address.  
   - 1'b0: Issue OK response even if unaligned address was aligned and   
   issued.
* ZEROES [15:0] - Zeroes

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | | | | | | | | | | REG | ZEROES | | | | | | | | | | | | | | | |

Table 81 AHBS\_CTL register

### AHBS\_IM

Interrupt Mask Register

Attribute: RW

Bit field description:

* REM [17] - 1'b1: No interrupt raised on RE.  
   - 1'b0: Interrupt raised on RE.
* WEM [16] - 1'b1: No interrupt raised on WE.  
   - 1'b0: Interrupt raised on WE.
* ZEROES [15:0] - Zeroes

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | | | | | | | | | REM | WEM | ZEROES | | | | | | | | | | | | | | | |

Table 82 AHBS\_IM register

### AHBS\_STS

Status Register.

Attribute: WZC

Bit field description:

* RE [17] - 1'b1: Unaligned read address was received.
* WE [16] - 1'b1: Unaligned write address (wrt asize)., or partial write strobes (wrt   
   asize) was received.
* ZEROES [15:0] - Zeroes

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | | | | | | | | | RE | WE | ZEROES | | | | | | | | | | | | | | | |

Table 83 AHBS\_STS register

APB Bridge

### APBSLV\_BRIDGE\_ID

Bridge ID register.

Attribute: R

Bit field description:

* **ZEROES** [15:8] - Zeroes
* **ID** [7:0] - Unique bridge id.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | | | | | | | | | | | ZEROES | | | | | | | | ID | | | | | | | |

Table 84 APBSLV\_BRIDGE\_ID register

### APBSLV\_BRIDGE\_VERSION

Bridge version register.

Attribute: R

Bit field description:

* **VER** [3:0] - Bridge version.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | VER | | | |

Table 85 APBSLV\_BRIDGE\_VERSION register

### APBSLV\_SLVS\_SLEEP\_STATUS

Slave sleep status register.

Attribute: RW

Bit field description:

* **STS\_15** [15] - 1'b1: Slave is in sleep mode  
   - 1'b0: Slave is active
* **STS\_14** [14] - 1'b1: Slave is in sleep mode  
   - 1'b0: Slave is active
* **STS\_13** [13] - 1'b1: Slave is in sleep mode  
   - 1'b0: Slave is active
* **STS\_12** [12] - 1'b1: Slave is in sleep mode  
   - 1'b0: Slave is active
* **STS\_11** [11] - 1'b1: Slave is in sleep mode  
   - 1'b0: Slave is active
* **STS\_10** [10] - 1'b1: Slave is in sleep mode  
   - 1'b0: Slave is active
* **STS\_9** [9] - 1'b1: Slave is in sleep mode  
   - 1'b0: Slave is active
* **STS\_8** [8] - 1'b1: Slave is in sleep mode  
   - 1'b0: Slave is active
* **STS\_7** [7] - 1'b1: Slave is in sleep mode  
   - 1'b0: Slave is active
* **STS\_6** [6] - 1'b1: Slave is in sleep mode  
   - 1'b0: Slave is active
* **STS\_5** [5] - 1'b1: Slave is in sleep mode  
   - 1'b0: Slave is active
* **STS\_4** [4] - 1'b1: Slave is in sleep mode  
   - 1'b0: Slave is active
* **STS\_3** [3] - 1'b1: Slave is in sleep mode  
   - 1'b0: Slave is active
* **STS\_2** [2] - 1'b1: Slave is in sleep mode  
   - 1'b0: Slave is active
* **STS\_1** [1] - 1'b1: Slave is in sleep mode  
   - 1'b0: Slave is active
* **STS\_0** [0] - 1'b1: Slave is in sleep mode  
   - 1'b0: Slave is active

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| u | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| u | | | | | | | | | | | | | | | | STS\_15 | STS\_14 | STS\_13 | STS\_12 | STS\_11 | STS\_10 | STS\_9 | STS\_8 | STS\_7 | STS\_6 | STS\_5 | STS\_4 | STS\_3 | STS\_2 | STS\_1 | STS\_0 |

Table 86 APBSLV\_SLVS\_SLEEP\_STATUS register