# Master bridges

## AM\_ADBASE: Base address register

## AM\_ADMASK: Address mask register

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 are programmable only if address range programmability is enabled on the master through NocStudio property. When programmability is disabled, these are read-only registers with values specified through NocStudio’s add\_range properties.

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 |
| LLC | 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.

An address range can be selectively disabled for any access or can be designated as read-only or write-only access using AM\_ADBASE[4:3] and AM\_ADMASK[3]. 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 |

## AM\_RELOC

Not applicable for current release

## AM\_TOCFG

This register is used to configure response timeouts.

AM\_TOCFG[8] needs to be set for timeout tracking to be enabled. When tis 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] 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 timedout and an interrupt is raised along with the slave ID to which the timed out request was sent.



## 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.

## AM\_CGC

Hysteresis counter specifies the number of consecutive inactive cycles which are needed for activity based clock gating of the master bridge to be triggered.

## 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.

## AM\_CLKCROSS\_RETRAIN

When set to 1’b1, the phase aligned N:1 or 1:N clock crosser logic between the host and noc clock domains will retrain itself to a new value of N. When the retraining of the logic is complete, it will reflect a 1’b0 in the same bit (bit 0). This bit has no effect when there is no phase aligned N:1 or 1:N clock crosser present in the bridge.

Note that no traffic must be present or expected on these interfaces at the time of setting the retraining bit. The new value of N, that is, the new phase aligned ratio’ed clock must be available at the host\_clk interface pin at the time of setting the retraining bit.

## AM\_STS

|  |  |
| --- | --- |
| Fields | Description |
| rof | 1’b1: Indicates that Maximum supported number of read commands are outstanding waiting for response and no more requests can be accepted.  1’b0: Indicates that the master bridge can accept more read requests |
| wof | 1’b1: Indicates that Maximum supported number of write commands are outstanding waiting for response and no more requests can be accepted.  1’b0: Indicates that the master bridge can accept more write requests |
| roe | 1’b1: No read commands have outstanding responses  1’b0: There are read commands outstanding from this master |
| woe | 1’b1: No write commands have outstanding responses  1’b0: There are write commands outstanding from this master |
| ars | 1’b1: Indicates that AR channel is stalled on a Hazard. |
| aws | 1’b1: Indicates that AW channel is stalled on a Hazard. |
| aro | 1’b1: Indicates that read commands are outstanding to the slave specified in OSSLV register |
| awo | 1’b1: Indicates that read commands are outstanding to the slave specified in OSSLV register |

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

## AM\_BRIDGE\_ID

Unique identifier assigned to the master bridge

## 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.

## 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.

|  |  |
| --- | --- |
| Fields | Description |
| e0 | ARADDR did not find a match in the master bridges address table and a decode error was issued |
| e1 | A decode error response was received from a slave device |
| e2 | A slave error response was received from a slave device |
| e3 | A WRAP command marked as non-modifiable (ARCACHE[0]=0) was detected |
| e4 | An AR command of FIXED burst type was detected |
| e5 | An AR command matched against multiple entries in the address table |
| e6 | Read response timeout occurred. With timeout enabled, a response wasn’t received within the expected interval. |
| e7 | A WRAP command not equal to cache line size of 64 Bytes was detected. |

e16 - e23 are WR channel counter parts of the above errors.

## 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.

## 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

## 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

## AM\_INTM

Interrupt mask register. Individual bit positions match 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.

## AM\_SYSCOREQ

This is the coherency connect request register. For agents that receive snoops (ACE and ACE-lite+DVM), the agent must be connected to the coherency controllers in order to receive snoops. If the SYSCOREQ is set up as I/O pins, this register is not accessible.

To connect to the coherency system, a value of one should be written to the register. This should be written if the agent is currently disconnected from coherency, which means SYSCOACK is set to zero. Coherency must be connected before the bridge will be able to process coherent or DVM accesses. The agent should expect to handle snoops as soon as they initiate a request to connect.

The disconnect process is used for power management or other shutdown scenarios. Before an agent can disconnect, it must make sure it will no longer need to accept snoops. For an ACE agent, this means all dirty lines have been written back to memory and the write responses have been returned. At that point, the agent can disconnect by writing a zero to the SYSCOREQ register. This will initiate a disconnect in the system. During the disconnect window, snoops may still arrive and must be handled appropriate. No new coherent or DVM requests should be made during this window. The window ends when the SYSCOACK register goes high. At that point the disconnect was successful and no new snoops will arrive.

## AM\_SYSCOACK

This is the coherency connection status. This value indicates whether the agent is connected to the coherency system. This is a read-only register. It is used in combination with the AM\_SYSCOREQ register.

## AM\_HASH\_FUNC

These register 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.

**Event filters and counters**

## 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.

## 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.

## AM\_CCMD0

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

## 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

## AM\_CNTR0

64-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.

## 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.

# Slave Bridge

## AS\_CGC

Hysteresis counter specifies the number of consecutive inactive cycles which are needed for activity based clock gating of the slave bridge to be triggered.

## 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.

## AS\_CLKCROSS\_RETRAIN

When set to 1’b1, the phase aligned N:1 or 1:N clock crosser logic between the host and noc clock domains will retrain itself to a new value of N. When the retraining of the logic is complete, it will reflect a 1’b0 in the same bit (bit 0). This bit has no effect when there is no phase aligned N:1 or 1:N clock crosser present in the bridge.

Note that no traffic must be present or expected on these interfaces at the time of setting the retraining bit. The new value of N, that is, the new phase aligned ratio’ed clock must be available at the host\_clk interface pin at the time of setting the retraining bit.

## AS\_STS

Slave bridge status bits.

|  |  |
| --- | --- |
| Fields | Description |
| wof | 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 |
| rof | 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 device can expect more read commands from NoC |
| woe | There are no write commands outstanding to the attached slave device |
| roe | There are no read commands outstanding to the attached slave device |

## AS\_BRIDGE\_ID

Unique identifier assigned to the slave bridge

## AS\_ERR

|  |  |
| --- | --- |
| Fields | Description |
| e0 | Decode error response received from slave device for read command |
| e1 | Slave error response received from slave device for read command |
| e2 | RID from read response produces a destination which is not present in the routing table |
| e3 | Interleaved read response. This can occur if interleaved read response is received from a slave device for which a de-interleaver was not specified |
| e4 | A read command which was marked as non-modifiable was modified by the slave bridge |
| e16 | Decode error response received from slave device for write command |
| e17 | Slave error response received from slave device for write command |
| e18 | BID from write response produces a destination which is not present in the routing table |
| e19 | A write command which was marked as non-modifiable was modified by the slave bridge |

## 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.

**Command statistic capture register not part of Jun 6 2014 release.**

# Parity/ECC registers

## Routers

### RPERR

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.

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.

### RPERRM

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.

## Bridges

## BTPERR

Transmit bridge parity error status register. One register bits 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.

## BTPERRM

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.

## BRPERR0

Receive bridge parity error status register monitoring parity errors on enabled layers from 0 to 7. 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.

## BRPERR1

Similar to BRERR0, but logging errors on enabled layers from 8 to 15.

## BRPERRM0

Mask register for receive bridge parity error interrupts from register BRPERR0. 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.

## BRPERRM1

Mask register for receive bridge parity error interrupts from register BRPERR1.