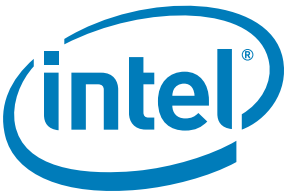


Intel® 6400/6402 Advanced Memory Buffer

Datasheet

October 2006

Reference Number: 313072-002

Contents

Figures

Tables

Revision History

Revision	Description	Date
001	Updated Release	May 2006
002	Updated Chapter 15 SPD Bits tables	October 2006

§

1 Introduction

This document is a core specification for a Fully Buffered DIMM (FB DIMM, also FBD) memory system. This document, along with the other core specifications, must be treated as a whole. Information critical to an Intel® 6400/6402 Advanced Memory Buffer (AMB) design appears in the other specifications, with specific cross-references provided.

1.1 Intel® 6400/6402 Advanced Memory Buffer Overview

The Intel 6400/6402 Advanced Memory Buffer (AMB) complies with the *FB-DIMM Architecture and Protocol Specification*. This device supports DDR2 SDRAM memory components. The AMB allows buffering of memory traffic to support large memory capacities. All memory control for the DRAM devices resides in the host, including memory request initiation, timing, refresh, scrubbing, sparing, configuration access, and power management. The AMB interface is responsible for handling FBD channel and memory requests to and from the local DIMM and for forwarding requests to other DIMMs on the FBD channel.

Fully Buffered DIMM (FBD) provides a high memory bandwidth, large capacity channel solution that has a narrow host interface. Fully Buffered DIMMs use commodity DRAMs isolated from the channel behind a buffer on the DIMM. The memory capacity is 288 devices per channel and total memory capacity scales with DRAM bit density.

The AMB will perform the following FBD channel functions:

- Supports channel initialization procedures as defined in the initialization chapter of the *FB-DIMM Architecture and Protocol Specification* to align the clocks and the frame boundaries, verify channel connectivity, and identify AMB DIMM position.
- Supports the forwarding of southbound and northbound frames, servicing requests directed to a specific AMB or DIMM, as defined in the protocol chapter, and merging the return data into the northbound frames.
- If the AMB resides on the last DIMM in the channel, the AMB initializes northbound frames.
- Detects errors on the channel and reports them to the host memory controller.
- Support the FBD configuration register set as defined in the register chapters.
- Acts as DRAM memory buffer for all read, write, and configuration accesses addressed to the DIMM.
- Provides a read buffer FIFO and a write buffer FIFO.
- Supports an SMBus protocol interface for access to the AMB configuration registers.
- Provides logic to support MemBIST and IBIST Design for Test (DFx) functions.
- Provides a register interface for the thermal sensor and status indicator.
- Functions as a repeater to extend the maximum length of FBD Links.

1.1.1 Transparent Mode for DRAM Test Support

In this mode, the AMB will provide lower speed tester access to DRAM pins through the FBD I/O pins. This allows the tester to send an arbitrary test pattern to the DRAMs. Transparent mode only supports a maximum DRAM frequency equivalent to DDR2 400.

Transparent mode functionality:

- Reconfigures FBD inputs from differential high speed link receivers to two single ended lower speed receivers (~200 MHz)
- These inputs directly control DDR2 Command/Address and input data that is replicated to all DRAMs
- Uses low speed direct drive FBD outputs to bypass high speed Parallel/Serial circuitry and provide test results back to tester

1.1.2 Debug and Logic Analyzer Interface

When optional LAI functionality is supported, the AMB can be used to support the connection of FBD links to a Logic Analyzer (LA) for debug.

AMB debug functionality:

- Reconfigures DDR2 interface to act as a Logic Analyzer Interface (LAI) to observe activity on FBD high speed links
- Triggers on programmable events in normal operation

1.1.3 DDR SDRAM

DDR2 SDRAM support:

- Supports DDR2 at speeds of 533, 667 MT/s
- Supports 256, 512, 1024, 2048 and 4096 Mb devices in x4 and x8 configurations
- 288 devices/channel (8 DIMMs/channel, 1 and 2 ranks/DIMM)
- 72-bit DDR2 SDRAM unregistered, unbuffered memory interface

1.2 AMB Block Diagram

Figure 1-1 is a conceptual block diagram of the AMB's data flow and clock domains.

1.3 Interfaces

Figure 1-1. Advanced Memory Buffer Block Diagram

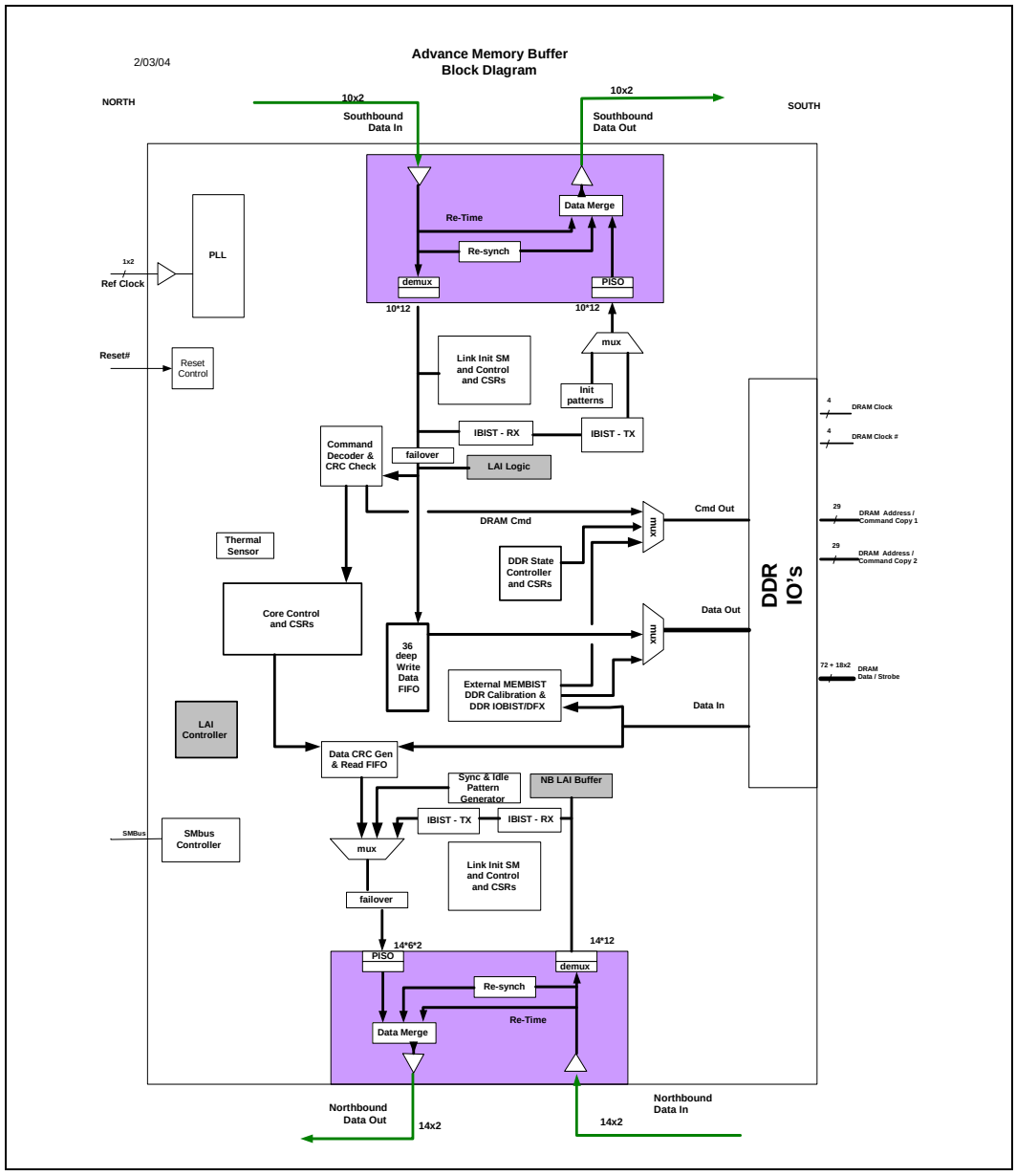
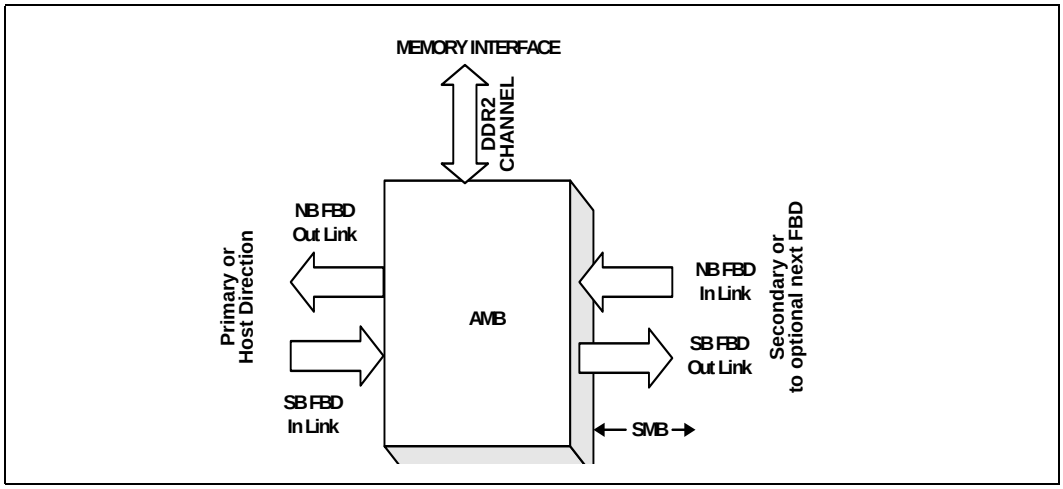


Figure 1-2 illustrates the AMB and all of its interfaces. They consist of two FBD links, one DDR2 channel, and an SMBus interface. Each FBD link connects the AMB to a host memory controller or an adjacent FBD. The DDR2 channel supports direct connection to the DDR2 SDRAMs on a Fully Buffered DIMM.

Figure 1-2. AMB Interfaces



1.3.1 FBD High-Speed Differential Point-to-Point Link (at 1.5 V) Interfaces

The AMB supports one FBD Channel interface consisting of two bidirectional link interfaces using high-speed differential point-to-point electrical signaling.

The southbound input link is 10 lanes wide and carries commands and write data from the host memory controller or the first adjacent DIMM in the host direction. The southbound output link forwards this same data to the next adjacent FBD.

The northbound input link is 13 to 14 lanes wide and carries read return data or status information from the next FB DIMM in the chain back towards the host. The northbound output link forwards this information back towards the host and multiplexes in any read return data or status information that is generated internally.

1.3.2 DDR2 Channel

The DDR2 channel on the AMB supports direct connection to DDR2 SDRAMs. The DDR2 channel supports two ranks of eight banks with 16 row/column request, 64 data signals, and eight check-bit signals. There are two copies of address and command signals to support DIMM routing and electrical requirements. Four-transfer bursts are driven on the data and check-bit lines at 800 MHz.

Propagation delays between read data/check-bit strobe lanes on a given channel can differ. Each strobe can be calibrated by hardware state machines using write/read trial and error. Hardware aligns the read data and check-bits to a single core clock.

The AMB provides four copies of the command clock phase references (CLK[3:0]) and write data/check-bit strobes (DQSs) for each DRAM nibble.

1.3.3 SMBus Slave Interface

The AMB supports an SMBus interface to allow system access to configuration registers independent of the FBD link. The AMB will never be a master on the SMBus, only a slave. Serial SMBus data transfer is supported at 100 kHz.

SMBus access to the AMB may be a requirement to boot a system. This provides a mechanism to set link strength, frequency and other parameters needed to insure robust operation given platform specific configurations. It is also required for diagnostic support when the link is down.

The SMBus address straps located on the DIMM connector are used by the AMB to get its unique ID.

More information is available in the SMBus chapter of this document.

Additionally, more detailed information about the SMBus, refer to the *System Management Bus (SMBus) Specification [HTTP://smbus.org/]*.

1.4 References

This product and datasheet are consistent with the following documents:

- *FB DIMM Architecture and Protocol Specification*, [1]
- *FBD Design for Test, Design for Validation (DFx) Specification*, [2]
- *FB4300/5300/6400 DDR2 Fully Buffered DIMM Design Specification*, [3]
- *JEDEC DDR2 SDRAM Specification*, JC 42.3 [4]
- *High Speed Differential Point-to-Point Link at 1.5 V for Fully Buffered DIMM*, [5]
- *SMBus Specification* (http://smbus.org/specs/smbus20.pdf) [6]
- Advanced Configuration and Power Interface Specification Version 2.0c (www.acpi.info) [7]
- Moisture/Reflow Sensitivity Classification for Nonhermetic Solid State Surface Mount Devices, IPC/JEDEC J-STD-020C

§

2 FBD Channel Interface

2.1 Intel 6400/6402 Advanced Memory Buffer (AMB) Support for FBD Operating Modes

The AMB may not support all operating modes documented in the *FB-DIMM Architecture and Protocol Specification* [1]. The following list defines which features/ modes are supported:

- 14 lane northbound (NB) with and without single lane Fail Over
- 13 lane NB with and without single lane Fail Over
- 10 lane southbound (SB) with and without single lane Fail Over
- Repeater mode
- LAI mode (optional feature supported in the Intel AMB)
- Transparent mode
- Recalibrate state

The following are optional FBD features not supported in the AMB:

- 12 lane NB non-ECC
- L0s low power state
- Data mask
- Variable read latency

2.2 Channel Initialization

Refer to Chapter 3, “Channel Initialization” in the *FB DIMM Architecture and Protocol Specification* [1] for FBD initialization protocol. The reset chapter covers some additional details about the initialization process.

2.3 Channel Protocol

2.3.1 General

Refer to Chapter 4, “Channel Protocol” in the *FB DIMM Architecture and Protocol Specification* [1] for FBD protocol.

2.3.2 Timeouts During TS0

The FBDLOCKTO register is used to help the AMB determine when to give up waiting for individual lanes to bit lock. The NBLINKCFG field is used to communicate when lanes are intentionally not in use. The BLTOCNT field is used to set a time out on waiting for a lane to bit lock. Lanes not bit locked by this time will be marked as failed.

2.3.3 Recalibrate State Considerations

In addition to what the *FBD Architecture and Protocol specification* describes around Recalibrate State, the AMB further requires that the host be sending NOP commands in at least the 2 frames preceding the exit from recalibrate state.

Requirement: 2 Cycles before the recalibrate counter expires, NOPs must be sent from the host.

Time	Frame Description	Recal Frame Counter
N	Sync with ERC bit set	0
N+1	NOP frame	RECALDUR
N+2	NOP frame	RECALDUR - 1
...	NOP frame	....
N+8	NOP frame	RECALDUR - 7
N+9	NOP frame - host may begin recalibration	RECALDUR - 8
...	.....	....
N + RECALDUR - 7	NOP frame - host must be finished with recalibration	8
...	.....	....
N + RECALDUR - 3	NOP frame	4
N + RECALDUR - 2	NOP frame	3
N + RECALDUR - 1	NOP frame - must be received at AMB without corruption	2
N + RECALDUR	NOP frame - must be received at AMB without corruption	1
0	Valid Sync Frame	0

Recal Frame Counter: 5 4 3 2 1 0

Rx Data: X X X NOP NOP Valid Cmd

where X represents “don’t care” data

2.3.4 Address Mapping of DDR Commands to DRAMs

DDR2 x4 Config		20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
256Mb (64Mbx4)	Row	1	X	X	RS	X	X	B1	B0	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
1KB page	Col	0	1	r/w	RS	X	X	B1	B0	X	A11	AP	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
512Mb (128Mbx4)	Row	1	X	X	RS	A13	X	B1	B0	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
1KB page	Col	0	1	r/w	RS	X	X	B1	B0	X	A11	AP	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
1Gb (256Mbx4)	Row	1	X	X	RS	A13	B2	B1	B0	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
1KB page	Col	0	1	r/w	RS	X	B2	B1	B0	X	A11	AP	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
2Gb (512Mbx4)	Row	1	X	A14	RS	A13	B2	B1	B0	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
1KB page	Col	0	1	r/w	RS	X	B2	B1	B0	X	A11	AP	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
DDR2 x8 Config		20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
256Mb (32Mbx8)	Row	1	X	X	RS	X	X	B1	B0	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
1KB page	Col	0	1	r/w	RS	X	X	B1	B0	X	X	AP	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
512Mb (64Mbx8)	Row	1	X	X	RS	A13	X	B1	B0	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
1KB page	Col	0	1	r/w	RS	X	X	B1	B0	X	X	AP	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
1Gb (128Mbx8)	Row	1	X	X	RS	A13	B2	B1	B0	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
1KB page	Col	0	1	r/w	RS	X	B2	B1	B0	X	X	AP	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
2Gb (256Mbx8)	Row	1	X	A14	RS	A13	B2	B1	B0	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0

2.3.5 FBD L0s State

The L0s state is not supported in the AMB.

2.4 Reliability, Availability, and Serviceability

Refer to Chapter 5, “Reliability, Availability and Serviceability” in the *FB DIMM Architecture and Protocol Specification* [1] for FBD RAS requirements.

2.4.1 Channel Error Detection and Logging

See for details on the error handling.

2.5 Channel Configuration

2.5.1 Re-sync and Resample Modes

A separate control is available for both the NB and SB FBD links to select between lower latency re-sample and lower jitter re-sync modes for repeating received data. Selection between these two modes is a function of platform design and configuration and should be set by BIOS prior to link initialization

The **FBDSBCFGNXT.SBRESYNCEN** and **FBDNBCFGNXT.NBRESYNCEN** bits make this selection.

Descriptions of these two modes follows.

2.5.1.1 Accumulated Tracking Effects (Re-Sample Option)

Each AMB acts as a repeater for the FBD channel. Since the data driven from each DIMM to the next DIMM may experience random or periodic phase shifts, the effect of these phase shifts must be accommodated in the design. Consider a system with three DIMMs daisy-chained together. A phase shift generated in the first DIMM will be seen by the second DIMM but will not be immediately propagated to the third DIMM. Unlike an analog buffer, the phase shift is not automatically driven to subsequent DIMMs since

the data driven to the subsequent DIMMs is re-sampled in the AMB to reduce jitter. The lowest latency implementation would use the derived clock to retransmit the outbound signal. The second DIMM will see the phase shift as a slight change in the position of the data eye at its receiver. The clock tracking loop filter in the second DIMM will measure several bit cells and may eventually determine that it should adjust the phase of its derived clock to capture the data closer to the center of the new data eye location. Only when the second DIMM makes its phase change will the effect propagate to the third DIMM. More detail of the data sampling technique may be found in the "High Speed Differential Point-to-Point Link at 1.5V for Fully Buffered DIMM Specification".

2.5.1.2 Accumulated Tracking Effects (Re-Sync Option)

An alternative implementation would place a voltage/thermal (VT) drift compensation buffer between the receiver and the transmitter section of the AMB I/O cell. The drift compensation buffer would re-synchronize the signal with a multiple of the reference clock. This buffer would have to be deep enough to handle the absolute magnitude of delay change of the daisy-chain channel over voltage and temperature. More detail of the data sampling technique may be found in the "High Speed Differential Point-to-Point Link at 1.5V for Fully Buffered DIMM Specification".

2.5.2 Other Channel Configuration Modes

Other channel electrical configuration parameters that should be set up prior to link initialization include

- Link frequency (**LINKPARNXT.CFREQ)**
- SB Transmitter drive current (**FBDSBCFGNXT.SBTXDRVCUR)**
- SB Transmitter de-emphasis values (**FBDSBCFGNXT.SBTXPREEMP)**
- NB Transmitter drive current (**FBDNBCFGNXT.NBTXDRVCUR)**
- NB Transmitter de-emphasis values (**FBDNBCFGNXT.NBTXPREEMP)**

The parameters contained in these registers are described more completely in the "High Speed Differential Point-to-Point Link at 1.5V for Fully Buffered DIMM Specification".

Additional channel configuration registers that should be set up prior to link initialization include

- First 6 bytes of the SPD parameter registers
  - **SPDPAR01NXT,SPDPAR23NXT,SPDPAR45NXT**
- FBD Bit Lock Time Out Register (**FBDLOCKTO)**

2.5.3 Lane to Lane Skew on a Channel

The FBD Channel is expected to support a maximum skew of up to 46UI. The deskew buffers on an individual AMB need to be able to support this amount of accumulated skew. The actual skew observed will be a function of the skew introduced by platform layout, DIMM layout, the number of active AMBs in the channel and skew introduced by AMBs.

2.6 Repeater Mode

The AMB may also be used as an FBD link repeater to extend distances at which links can operate. This mode can be automatically set by the BFUNC and SA pins. In this mode, the AMB functions in the same way as a regular DIMM with the exception that DRAM commands are not supported.

Link behavior is the same as for normal DIMMs

- Participates in link initialization like a normal DIMM
- Responds to Reads and Writes to AMB configuration registers
- Status is returned in response to Sync commands
- Link errors are detected and alerts generated

SMBus access is the same except for the base Slave address is different than from normal DIMM.

Slave address[6:3] = 4'b0011 for Repeaters, instead of

Slave address[6:3] == 4'b1011 for normal DIMMs

2.7 Channel Latency

The critical elements that AMB contributes to the latency calculation are the chip crossing delays.

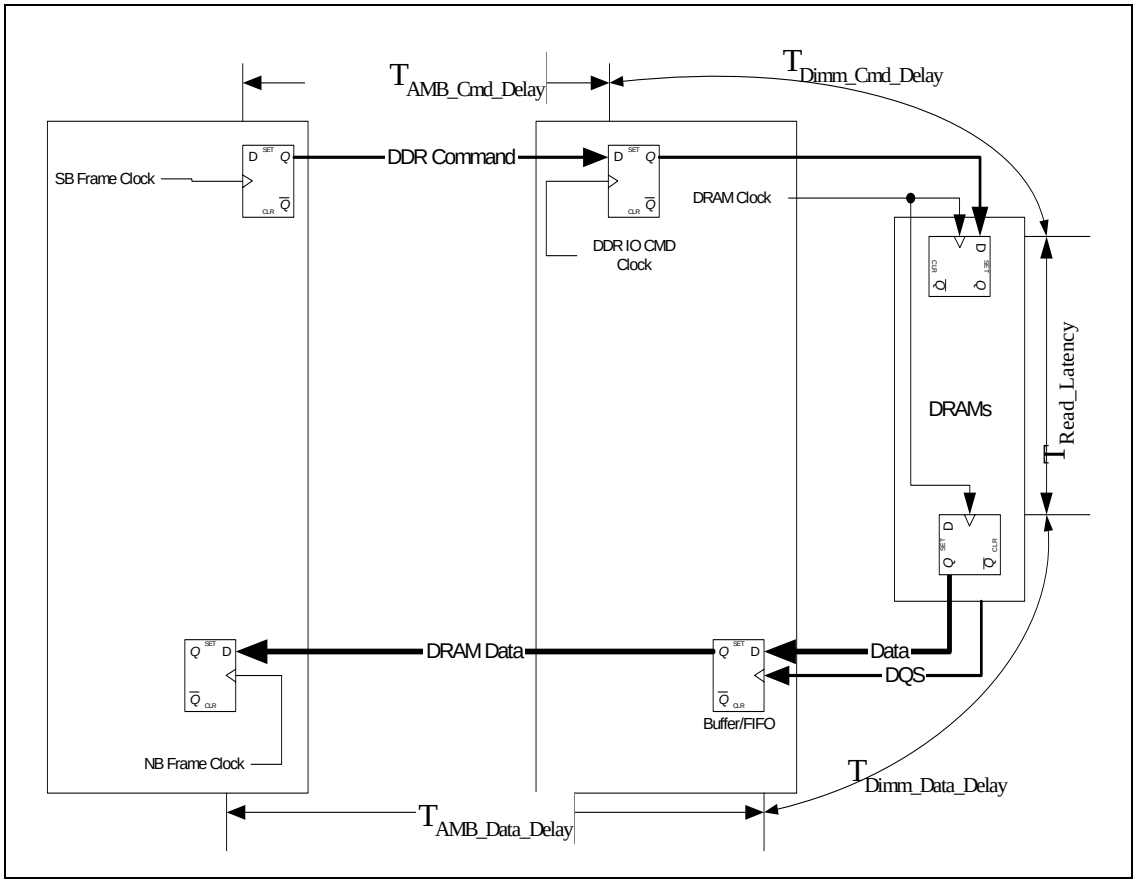
- Southbound latency contributions
  - — Delay from an input FBD link transaction to commands on the DDR interface and
  - — Pass-thru delay of forwarded Southbound FBD transactions.
- Northbound latency contributions
  - — Delay from DDR Read data input on the last DIMM to FBD link transactions and
  - — Pass-thru delay of forwarded Northbound FBD transactions back towards the host.

These timing delay values are documented in Chapter 4, "Electrical, Power, and Thermal."

2.7.1 Command to Data Delay Calculation

shows the various components that make up the over all delay through an AMB for a memory command.

Figure 2-1. Delays Through an AMB



The definition for terms used in the are given below.

UI	This is the unit interval on the FBD link. This is same as the period of the FBD link bit-rate clock.
NB:	Northbound
SB:	Southbound
	$T_{Read\_Latency}$: DRAM Parameter. T_{CAS} + $T_{Additive\_Latency}$
$T_{AMB\_Cmd\_Delay}$	This value is very specific to an AMB implementation. This is the time it takes for the command to be transferred from the FBD receiver to the DDR I/O Cluster. This includes any differences between the FBD frame clock and the DDR I/O clock that latches the command into the DDR I/O cluster. This can be different for different DRAM frequencies.
$T_{AMB\_Data\_Delay}$	This value is very specific to an AMB implementation. This is the minimum time it takes for the data that is returned from the DRAMs to be transferred from the DDR I/O cluster to the FBD I/O for transmission on the link. This includes any buffering and clocking delays for the data within the chip. This can be different for different DRAM frequencies.
$T_{Dimm\_Cmd\_Delay}$	This includes the delays for the command through the DDR I/O cluster, differences between DDR I/O CMD Clock and DRAM clock, any routing delays for the clock and command on the

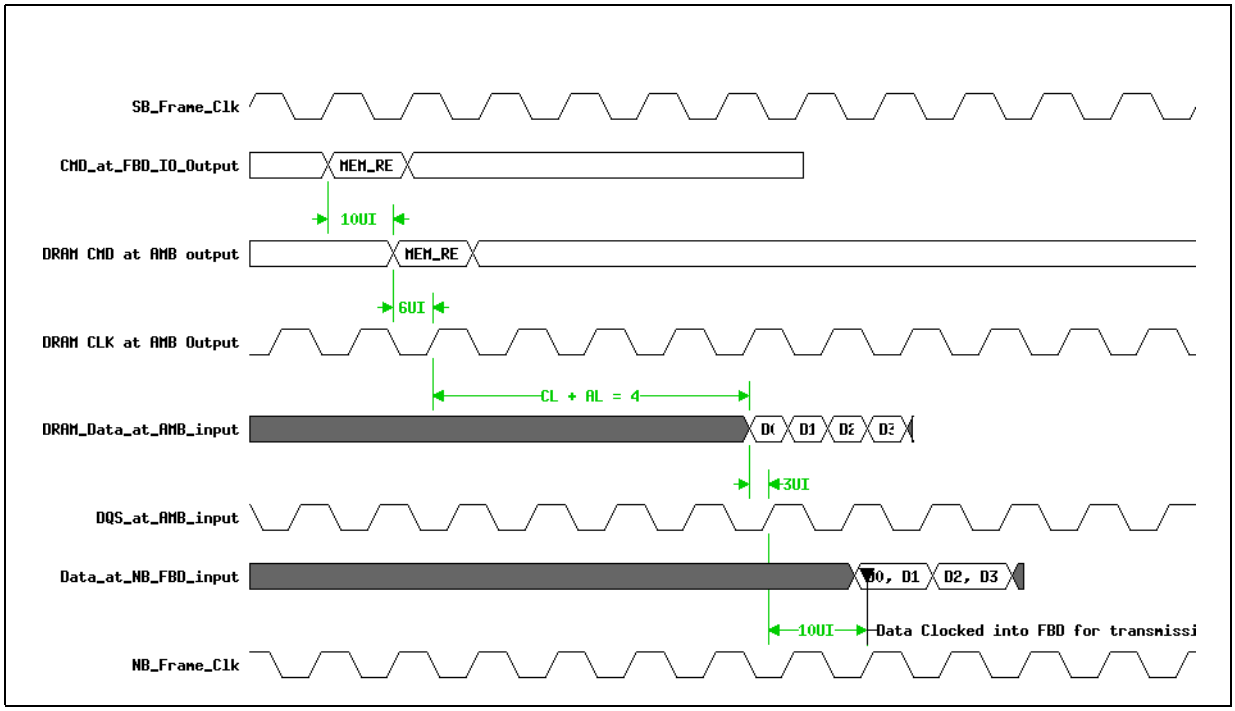
	DIMM and any set-up and hold-times in the AMB and the DRAMs.
$T_{Dimm\_Data\_Delay}$	This includes the routing delays for the data and strobes from the DRAM to the AMB, skews between the DRAMs, delays through the DDR I/O cluster and any set-up and hold-times in the AMB and DRAMs.
$T_{CMD\_To\_Data}$	This is equal to ($T_{Read\_Latency}$ + $T_{AMB\_Cmd\_Delay}$ + $T_{Dimm\_Cmd\_Delay}$ + $T_{Dimm\_Data\_Delay}$ + $T_{AMB\_Data\_Delay}$). This can be specified with 1UI granularity. This will be different for each DRAM type. It does not include any delays inside the I/O to deskew and frame align the incoming data on the SB side nor does it include the delays inside the NB I/O on the transmit side.

The T_{CAS} and $T_{Additive\_Latency}$ are specified in the DRC register (DRC.cl and DRC.al). The CMD2DATANXT register is initialized with a value equal to ($T_{CMD\_To\_Data}$ - $T_{Read\_Latency}$). This value of CMD2DATANXT is specified in the SPD EEPROM. All AMBs are expected to receive the data from the DRAMs with the calculated value of $T_{CMD\_To\_Data}$.

All AMBs power up with a default value of 5 for $T_{CMD\_To\_Data}$. This can be done either by setting the default values of DRC.CL and DRC.AL or by setting the default value of the CMD2DATACUR.FRMS to be 5. The AMBs must be able to return status and configuration register reads with this default timing. This will enable the BIOS to initialize the proper values from SPD.

The following simplified timing diagram illustrates how the last AMB uses the values specified in DRC.CL, DRC.AL, CMD2DATANXT.DLYFRMS and CMD2DATANXT.DLYFRAC.

Figure 2-2. Command to Data Delay Timing



In Figure 2-2, it is assumed that there are no routing delays on the DIMM itself. The command takes 10UI to get from SB FBD to the DDR I/O ($T_{AMB\_Cmd\_Delay}$). This includes time it takes to validate CRC of the frame and to decode the command. It is assumed DDR_IO_Cmd_Clk shown in is delayed from the SB_Frame_Clk by 10 UI. The

DRAM SCLK is placed at the center of the command window which adds an additional 6UI of delay. The DRAMs are programmed with a $T_{Read\_Latency}$ of 4. The DQS strobes are centered when they get to the AMB (3UI delay) and the data takes an additional 10 UI to propagate from the DDR I/O to the NB FBD I/O (including setup time at NB FBD and CRC generation) before it can be clocked into the NB FBD for transmission. The total delay without including the DRAM access time in this case is (10 + 6 + 3 + 10UI) 29UI. The CMD2DATANXT.DLYFRMS will be programmed with a value 2 and the CMD2DATANXT.DLYFRAC with 5UI. If the AMB supports only a 2UI granularity, then the values should be 2 frames and 6UI respectively. If an AMB does not support sub-frame delays, it is expected that the value in SPD will be round up the to the nearest frame. Any additional delays caused by rounding up should be supported by additional buffering in the DDR I/O.

The above figure shows the last arriving data from the DDR. It is expected that any data arriving earlier than this, due to differences in routing, and so forth, is buffered up in some way inside the DDR I/O. The expectation is that all AMBs (last and intermediate) unload the data from the DDR I/O with the timing specified by the $T_{CMD\_To\_DATA}$. The Last AMB is expected to be able send data on the NB links with no additional delay added, if C2DINCRCUR.INCRDLY is 0. Any additional delay needed due to the value programmed in C2DINCRCUR.INCRDLY register in the last AMB or to delay the data in intermediate AMB before the merge, is handled by FIFOs in the AMB core.

In the above figure, it is assumed that the AMB can place the clocks in position shown above. If this is not the case, then additional delays due to non-optimal placement of the clocks should be taken into account to calculate the total delay to be programmed into the SPD.

§

3 DDR Interface

3.1 Intel 6400/6402 Advanced Memory Buffer (AMB) DDR Interface Overview

The DDR interface on the AMB consists of:

- A command decoder.
- A FIFO write buffer to hold the write data before it is written to the DDR channel. The write FIFO buffer has 36 entries of 72 bits (36 x 72b). A maximum of 35 entries can be used to store DDR bursts. Write data targeted for other AMB parts on the channel will use three of the 35 entries until the target AMB is known. The write FIFO buffer fills at half of the DDR data rate, and empties at the DDR data rate. The write FIFO buffer must support an invalidate write FIFO command (FBD Soft Reset command).
- A FIFO read buffer to hold the read data so that each DIMM returns data with the same latency as the southernmost DIMM in the chain. Latency is measured in increments of core clock periods. The core clock runs at the DDR command clock rate (half the data rate frequency). The latency through the FIFO read buffer on the southernmost DIMM is expected to be zero.
- A DDR cluster which serves as a DIMM buffer by registering outbound commands and data at output flops. The cluster also captures and levelizes incoming read data.
- A reset FSM which puts the DIMM in self-refresh when reset is asserted, and exits self-refresh when reset is deasserted and southbound frame training is complete.
- A calibration FSM that automatically sets the timing for DQS receiver enable and DQS delay or equivalent DDR timing control mechanism. The DQS receiver enable calibration uses a series of DRAM read and write operations to find the center of the read DQS preamble. During normal operation the DQS receivers will be enabled at the preamble center to ensure that the DQS signal is received correctly into the AMB Component's DDR I/O circuits. The DQS delay calibration uses a series of reads and writes to align the read DQS waveform rising and falling edges to the center of the DQ data eye.
- A configuration register set to allow software to issue DRAM power up and DRAM MRS/EMRS commands, as well as a self-refresh entry command. These registers are accessible through FBD channel commands and the SMBUS interface.
- "Burst Write Interrupt" is not supported.

3.2 Data Mapping

See the protocol chapter of the *FB DIMM Architecture and Protocol Specification* for the mapping between data in DDR DRAM devices and data in FBD frame formats for 4-bit and 8-bit devices.

3.3 Command / Address Outputs

Two sets of DRAM command and address output pins are provided for loading and timing considerations. Each set drives the same DRAM commands, but the two address busses are inverted from each other in order to reduce power consumption and heat produced on the DIMM. The command and address output pin behavior is detailed below:

1. Minimum address toggling. The address associated with the last command issued on the DRAM bus is retained during DRAM NOP/Deselect commands. The address and bank bits do not revert to all 1's (or all 0's) when the command bus is idle.
2. Balanced bank and address busses. With some exceptions, the bank and address busses on the two bus copies are inverted from each other. This minimizes the current load on the V_{TT} supply regulator because the balanced address bus sinks as much current as it sources. There are exceptions to the inversion behavior to allow for commands that use one or more address bits to control DRAM functionality. Balancing can also be disabled by setting the DRC.BALDIS register field.

   Balancing Exceptions:

   a. Address bit A10 is not balanced during all read, write, and precharge commands.

   b. No address or bank bits are balanced during any MRS and EMRS commands.

   c. Column address A0 is not balanced when the DRC.SEQADD bit is set. As with A10, A0 is only not balanced during read, write, and precharge commands. This mode is intended to work with DRAMs in sequential address mode.

   d. No address or bank bits are balanced during any command when the DRC.BALDIS register field is set.
3. Balanced idle command bus. When the command bus is idle, a deselect command is issued with all chip selects high and all RAS/CAS/WE signals driven low on both command/address copies.
4. Command/Address output control with CKE. All command and address pin outputs, except for ODT, CKE, and CLK, will float one DRAM clock cycle after both CKE pins transition from high to low. The command/address pins will be driven to valid signal levels on the same cycle that either CKE pin is driven from low to high.
5. Output control during link reset. When the AMB core logic is in reset, CKE and ODT will be driven low, and CLK will run at normal levels and frequency. The remaining command/address pins will float during reset.
6. Command/Address output control in S3 mode. When the AMB core logic is in S3 power mode, all command/address outputs, including CKE, ODT, CLK, and all other command/address pins, will be driven low.
7. CSR output control of command/address. All command/address pin outputs will float when the appropriate DRC bits are set. Setting the DRC.CADIS field will float the RAS, CAS, WE, Bank, and Address pins. DRC.CSDIS controls the chip select pins. DRC.ODTDIS, DRC.CKEDIS, and DRC.CLKDIS float the ODT, CKE, and clock pins respectively.

3.3.1 CKE Output Control

The are six different functions that affect the state of the CKE outputs during normal operation:

1. DRC.CKEFRCLOW CSR. When set, this bit forces both CKE outputs low. No other CKE control function overrides this CSR. This allows firmware to prevent hardware from issuing all DRAM commands. This could be used by firmware to keep the DRAM bus idle throughout a fast reset sequence that occurs before the DRAM initialization command sequence has been initiated.
2. Self-Refresh FSM. When entering a fast reset sequence, a hardware FSM takes control of the DDR command bus, including the CKE outputs, in order to issue a series of commands to put the DRAM's into self-refresh. This FSM is overridden by the DRC.CKEFRCLOW CSR.
3. Automatic self-refresh exit with link training. When a T0 link training sequence is complete, both CKE outputs will be automatically asserted high in order to exit self-refresh. This control is inhibited by setting the DSREFTC.DISSREXIT bit. The DRC.CKEFRCLOW also overrides this automatic self-refresh exit function.
4. Self-Refresh entry DCALCSR command. The DCALCSR register can be programmed to launch an FSM that issues a single self-refresh entry command. The DRC.CKEFRCLOW bit overrides this function.
5. Channel commands. The host can directly control the CKE output state through channel command protocol. The DRC.CKEFRCLOW bit overrides this function.
6. DRC CKE0 and CKE1 CSR bits. These CKE bits in the DRC are both control and status bits for the CKE outputs. Software can write these bits to directly control the CKE outputs. These bits reflect the state of the CKE outputs one cycle after the output state is changed by one of the other control functions. The DRC.CKEFRCLOW bit overrides this function.

Channel commands that change the state of the same CKE output must be separated by at least two DRAM clock cycles. Configuration writes and channel commands that affect the same CKE output must not occur within two cycles of each other to avoid unstable CKE behavior.

3.4 DQS I/O and DM Outputs

The AMB sends and receives source synchronous differential strobes (DQS) to transfer data (DQ/CB) during write and read DRAM transactions. DQS9 through DQS17 also support the data mask (DM) function in x8 mode. Setting the MTR.WIDTH configuration register enables x8 mode. When driving DM, the timing of the transition from floating, to driving, and back to floating is unchanged, but DQS[17:9] do not toggle and instead drive a constant level. DQSP[17:9] drive low and DQSN[17:9] drive high. In x8 mode DQS[17:9] are not used to capture read data. The table below lists which DQS signals are associated with which DQ/CB pins in x8 and x4 mode.

Table 3-1. DQS Association with DQ/CB Pins in x8 and x4 Mode

DQS Pin	x4 Mode: MTR.WIDTH=0		x8 Mode: MTR.WIDTH=1	
	Output Function	Input/Output Data Mapping	Output Function	Input/Output Data Mapping
DQS17	Write DQS	CB[7:4]	DM	N/A
DQS8	Write DQS	CB[3:0]	Write DQS	CB[7:0]
DQS16	Write DQS	DQ[63:60]	DM	N/A

Table 3-1. DQS Association with DQ/CB Pins in x8 and x4 Mode

DQS Pin	x4 Mode: MTR.WIDTH=0		x8 Mode: MTR.WIDTH=1	
	Output Function	Input/Output Data Mapping	Output Function	Input/Output Data Mapping
DQS7	Write DQS	DQ[59:56]	Write DQS	DQ[63:56]
DQS15	Write DQS	DQ[55:52]	DM	N/A
DQS6	Write DQS	DQ[51:48]	Write DQS	DQ[55:48]
DQS14	Write DQS	DQ[47:44]	DM	N/A
DQS5	Write DQS	DQ[43:40]	Write DQS	DQ[47:40]
DQS13	Write DQS	DQ[39:36]	DM	N/A
DQS4	Write DQS	DQ[35:32]	Write DQS	DQ[39:32]
DQS12	Write DQS	DQ[31:28]	DM	N/A
DQS3	Write DQS	DQ[27:24]	Write DQS	DQ[31:24]
DQS11	Write DQS	DQ[23:20]	DM	N/A
DQS2	Write DQS	DQ[19:16]	Write DQS	DQ[23:16]
DQS10	Write DQS	DQ[15:12]	DM	N/A
DQS1	Write DQS	DQ[11:8]	Write DQS	DQ[15:8]
DQS9	Write DQS	DQ[7:4]	DM	N/A
DQS0	Write DQS	DQ[3:0]	Write DQS	DQ[7:0]

3.5 Refresh

The AMB is required to manage DRAM refresh during channel resets and when the auto-refresh function is enabled. During channel resets, the Self-Refresh FSM takes control of the DDR command bus and places the DRAMs in self-refresh mode. The Auto-Refresh FSM generates auto-refresh commands when the DAREFTC.AREFEN bit is set. The Self-Refresh FSM will override and take control away from the Auto-Refresh FSM when a reset event occurs.

A self-refresh entry command can also be generated by programming the DCALCSR register. The FSM that controls this function will be described in the DRAM initialization and (E)MRS command section.

3.5.1 Self-Refresh During Channel Reset

The Self-Refresh FSM is launched by a sync train error on the link, or when an electrical idle condition is detected on the link. The Self-Refresh FSM will take the DRAM's from an unknown state and put them into self-refresh mode. The AMB does not track the DRAM state during normal operation, and so has a single process for getting from any DRAM state starting point to the self-refresh state.

When launched, the Self-Refresh FSM will execute the following steps. Each step below is one of the states of the FSM. Setting the DRC.CKEFRCLOW bit will prevent the command sequence described below from being issued on the DDR bus since it will force the CKE outputs low.

1. Clear the DAREFTC.AREFEN CSR to stop the AMB auto-refresh engine if enabled.
2. Block all DRAM commands, except those initiated by the self-refresh FSM.
3. Wait until any in-process read or write commands complete, with a minimum wait time of DSREFTC.TCKE, the DRAM "Minimum CKE pulse width time" specification. In-process reads/writes must complete to ensure that the DRAM ODT control outputs are driven low. The minimum time allows for the case where self-refresh entry or power-down entry was executed just before the channel reset.
4. Assert both CKE output pins by setting the DRC.CKE0/1 CSR fields. This will have no effect on the DRAM's if the CKE pins were already asserted.
5. Wait DSREFTC.TXSNR, the DRAM's "Exit self-refresh to a non-read command" specification, to allow any in-process DRAM command to complete. This allows time to complete any command that may have been issued just before the channel reset event, such as an auto-refresh, as well as allows for self-refresh exit that may have been initiated when the self-refresh FSM asserted the CKE pins high.
6. Issue a "precharge all" command to both ranks. This guarantees that the DRAM's will be in an "idle" state.
7. Wait as required by DSREFTC.TRP, the DRAM "Precharge time."
8. Issue an auto-refresh command to both ranks. This meets the DRAM requirement that at least one auto-refresh command is issued between any self-refresh exit to self-refresh entry transition. The AMB staggers the auto-refresh commands to the two ranks by the DSRETC.DRARTIM value in order to avoid stressing the system power supply with too many DRAM's refreshing at the same time.
9. Wait as required by DAREFTC.TRFC, the DRAM "Refresh to active/refresh command time."
10. Issue a self-refresh entry command to both ranks. The AMB staggers the self-refresh entry commands to the two ranks by the DSRETC.DRSRENT value to avoid stressing the system power supply.

When the channel comes out of "fast reset" (exiting the FBD link disable state), the AMB will automatically issue a self-refresh exit command to both ranks after the FBD Link Testing State is reached and the AMB core clock is stable. Note that this does not apply when the DISSREXIT bit is set as it should be when the AMB is powering up or when exiting S3 mode. The DRC.CKEFRCLOW CSR also overrides the automatic exit command by forcing the CKE outputs to stay low.

3.5.2 Automatic Refresh

The AMB has an Auto-Refresh FSM for issuing auto-refresh commands to the DRAM's on regular intervals. This can be enabled when the DRAM bus is otherwise idle, but not during any other mode that generates DRAM commands, including DRAM power up and

initialization, DDR I/O calibration, and normal operation where the FBD channel issues DRAM commands. The DAREFTC CSR controls the refresh interval and period, and includes an enable bit to turn auto-refresh command generation on and off. The auto-refresh FSM may be integrated or separate from the MemBIST functions in hardware, but the auto-refresh function runs independently of the MemBIST state. Multiple MemBIST tests can be started and completed with the auto-refresh FSM running during, in between, and after all MemBIST tests complete.

The Auto-Refresh FSM generates auto-refresh command requests to an arbiter in hardware at regularly spaced intervals defined by the DAREFTC.TREFI CSR. For a single rank DIMM the command spacing is equal to TREFI cycles. For dual rank DIMMs the command spacing is TREFI/2, alternating between the two ranks so that each rank receives an auto-refresh command every TREFI cycles on average. Hardware abitrates between the Self-Refresh FSM command requests and commands generated by the MemBIST function. When MemBIST is not running, the auto-refresh command is immediately issued on the DDR bus. When MemBIST is running, an auto-refresh request is posted until the MemBIST FSM can interrupt its command sequence, precharge all open banks on all ranks, and all DRAM timing requirements are met so that an auto-refresh command can be accepted. The MemBIST FSM resumes its command sequence TRFC later. The MemBIST FSM must interrupt its operation within TREFI/2 in order to avoid causing an auto-refresh command to be dropped by the arbiter.

3.6 Back to Back Turnaround Time

The host controller is required to observe a turnaround time on the DRAM data pins within a DIMM when switching between read and write cycles, and when switching from reads from one rank vs. the other rank on the DIMM. The nominal turnaround time is one clock for each parameter, which is the minimum time required to prevent a collision of the postamble of the first transaction and preamble of the second transaction. Additional clocks may be required by the DIMM, especially at higher speeds.

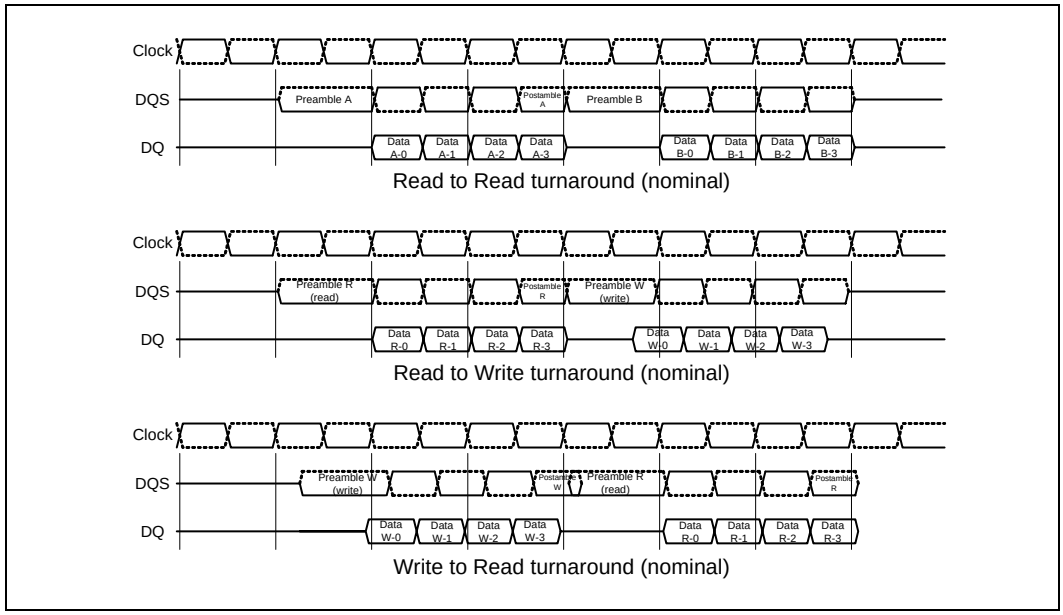
The three timing parameters are:

- Read to write turnaround time. The number of additional DRAM clocks in which the DRAM data bus must be idle between a read from either rank and a write cycle to either rank.
- Write to read turnaround time. The number of additional DRAM clocks in which the DRAM data bus must be idle between a write to one rank and a read from either rank. The write to read turnaround time to the same rank will generally dominated by the tWTR specification, which must also be observed.
- Read to read turnaround time. The number of additional DRAM clocks in which the DRAM data bus must be idle between a read from one rank and a read from the opposite rank.

These parameters are dependent on the AMB design and the DIMM PC board layout. The parameters are stored in the SPD. Each parameter is a 2 bit field allowing 0 to 3 additional clocks of turnaround time. See the *FB4300/5300/6400 DDR2 Fully Buffered DIMM Design Specification* for additional details.

Figure 3-1 shows the nominal turnaround times, with no additional clocks. Note that the DQS preamble and postamble may merge slightly when the first transaction is at worst case timings and the second transaction is at best case timings.

Figure 3-1. Nominal Turnaround Time Timing Diagram



3.7 S3 State Background Description

S3 is a platform power down state where DRAM memory contents are preserved while the rest of the platform powers down. This state is described more formally in the *Advanced Configuration and Power Interface Specification*. It is much like the Standby mode on mobile computers.

Functionally, the requirement for an AMB is that after a sequence in which:

- AMBS3 Recovery state is stored by the host,
- DRAMs are put into Self-Refresh,
- RESET# is asserted,
- Vcc (1.5 V), Vtt (0.9 V) and VddSPD(3.3 V) are powered down,

The contents of memory on the DIMM are preserved until S3 Recovery. Recovery requires that the memory on the DIMM be restored without losing data integrity.

The recovery sequence is much the same as an initial power on with the exceptions of:

- Vdd is already powered on
- DDR interface state is restored from stored register values (since no calibration which might compromise memory content is allowed)

To minimize DIMM current, VCC should be powered up during VTT transitions. The AMB determines that it is in the S3 mode by checking that VDD (1.8 V) is powered up and that VCC (1.5 V) is powered off. When in S3 the AMB drives all command/address outputs (including CKE, ODT, and CLK) to a low. This keeps the DRAM in auto-refresh and helps prevent DRAM data corruption. When coming out of the S3 mode, if the system brings VTT (0.9 V) up before VCC the AMB will drive low into the VTT pull-up circuitry. This causes significant current (approximately two times average but still normal) to flow from VTT into the AMB until VCC is powered up and the AMB comes out of S3.

3.7.1 S3 Recovery Configuration Registers

The following CSRs should be stored in non-volatile memory before entering S3 mode and restored before normal DRAM transactions begin.

- DRC
- MTR
- DSREFTC
- DAREFTC
- DDR2ODTC
- CMD2DATANXT
- S3RESTORE[15:0]
- PERSBYTE[13:0] - SPD Personality Bytes

3.8 DDR Calibration

The following sections describe these DDR calibration and initialization features:

- DRAM initialization and (E)MRS command CSR's and FSM
- DQS failure CSR
- DQS receive enable calibration
- DQS calibration

3.8.1 DRAM Initialization and (E)MRS FSM

The AMB provides a set of CSR's and an FSM that allow BIOS to manage DRAM power up initialization and set DRAM mode register bits. All commands needed for DRAM initialization can be generated, including precharge, refresh, mode register set (MRS), and extended mode register set EMRS commands. A self-refresh command can also be generated, although this is not required for initialization. The initialization/(E)MRS FSM only controls the issuing of single commands, and does not automatically initialize the DRAM. It is the responsibility of software to control the command sequence to correctly initialize the DRAM.

The set of CSR's include the DCALCSR and DCALADDR registers. The fields of these CSR's are described in detail in the configuration register chapter.

The DCALCSR is used to select the command to be issued, which ranks to select, start the FSM that issues the command, and provide completion status.

The DCALADDR sets the bank and address issued to the DRAM, and therefore defines the type of (E)MRS to be issued, including limited OCD commands. The DCALADDR can also be used to configure a precharge command as a "precharge all" command. DCALADDR[31:16] defines the DDR address bus during these commands, and DCALADDR[2:0] defines the ddr bank address bus.

The FSM that controls this function can have as few as three states: idle, issue command, and clear start bit. When the DCALCSR.START bit is set, and the DCALCSR.OPCODE bits select one of the command options, the FSM transitions from idle to the "issue command" state. After the command is issued, the FSM clears the DCALCSR.START bit and returns to the idle state. A more elaborate FSM may also be implemented. Firmware is required to control the minimum command spacing to meet all DRAM timing requirements. After setting the start bit, firmware should poll the

DCALCSR until the start bit is cleared. After the start bit is cleared, firmware waits a period of time based on DRAM command timing specifications before issuing a new command through the DCALCSR.

3.8.1.1 OCD EMRS Commands

FB DIMM DRAM timing is set up to work with OCD default calibration. Using the DCALCSR and DCALADDR registers, EMRS OCD default and OCD exit commands can be sent to the DRAMs.

Sending OCD EMRS drive(0), drive(1), or adjust DRAM commands may have implementation dependent outcomes and should not be used in normal operation.

3.8.1.2 MRS Command Example

The following example shows how to send out an MRS command:

1. Write a value of 0x02320000 to the DCALADDR csr. This will configure the address/bank bus for an MRS command with burst length 4, CAS latency 3, and write recovery 2.
2. Write a value of 0x80000003 to the DCALCSR. This selects the (E)MRS command mode and initiates the FSM that will issue the command.
3. Poll the DCALCSR until bit 31 is cleared to zero by hardware. This indicates that the FSM has completed the selected operation.

3.8.2 DQS Failure CSR

The DQSFAIL CSR allows the AMB to calibrate properly even when one or more DQS signals are missing, either through PCB or component failure. DQSFAIL is a 36 bit register, one bit for each DQS pair of each rank. Software can set this CSR to force any DQS signal to be excluded from the automatic DDR bus calibration. Hardware will also detect missing DQS signals and automatically exclude them from calibration.

3.8.3 Automatic DDR Bus Calibration

The AMB has two automatic DDR bus calibration functions that must be executed before read data can be captured reliably. These functions issue a series of write and read transactions on the DRAM bus, analyze the read data captured, and program a set of calibration results configuration registers. During subsequent operations, these configuration registers control the DDR I/O circuits and ensure proper data capture. DIMM memory contents are not preserved during calibration. Calibration can take up to several ms to complete. The following steps run the calibration:

1. Program the DCALCSR to 0x8000000C. This selects and initiates the first of two calibrations.
2. Poll the DCALCSR until bit 31 is cleared to zero by hardware.
3. Program the DCALCSR to 0x80000005. This selects/initiates the second calibration.
4. Poll the DCALCSR until bit 31 is cleared to zero by hardware.

3.8.4 Receive Enable Calibration

The DQS input receiver needs to be disabled when the DDR bus is floating (tri-stated), for example, between the read and write data transfers. Otherwise, the floating strobe would cause spurious data to be written into the read data FIFO. Also, the DQS input receiver needs to be disabled during a write so that the write data strobes do not cause unwanted data or check-bits to be written into the read data FIFO.

During a read, the DRAM's initially drive the DQS signals low for a full cycle. This is the preamble. After the preamble, the DQS signals are toggled twice per cycle, for every cycle there is a data transfer, which is determined by the configured burst length and the number of back-to-back read commands that were issued to the selected rank. After the last DQS falling edge, the DQS signal is driven low for a half cycle. This is the post-amble. After the post-amble the DQS signals are tri-stated.

The AMB automatically finds the end of the preamble of each of the 18 DQS pairs on the DDR bus. That is, it finds the location of the first waveform transition that defines the end of the preamble of each DQS pair. Once this is complete, the AMB calculates the location of the center of the preambles, and stores this information for use during read transactions. Receiver calibration is initiated by setting the DCALCSR.START CSR. Hardware clears this bit when the calibration is complete. The calibration method modifies the data contents of the DIMM.

3.8.5 DQS Delay Calibration

The DQS Delay calibration adjusts the AMB Component's on-chip delay circuits that align DQS signals to the center of their associated DQ/CB data eyes at the capture flops in the DDR I/O cluster. This maximizes the DQ/CB setup and hold time at these flops, which capture source synchronous data from the DDR data bus.

DQS delay calibration is initiated by setting the DCALCSR.START CSR. Hardware clears this bit when the calibration is complete. The calibration method modifies the data contents of the DIMM. The calibration is accomplished by issuing a series of write and read transactions, and comparing expected to captured data

3.9 DIMM Organization

The AMB supports DIMMs with 1 or 2 independent ranks (chip-selects). Each rank consists of DDR SDRAM devices or DDR SDRAM devices. Dual rank DIMMs consist of DDR SDRAM devices or DDR SDRAM devices.

The AMB DDR I/O circuits provide three main functions: an outbound command and data path, an inbound data path, and analog compensation.

Analog compensation is provided, along with configuration registers, to control and match the output impedance and slew rate of the off-chip drivers and on on-die termination. The main configuration registers are the leg override (for impedance) and slew override fields of the spdpar67cur and spdpar1011cur CSR's. The analog compensation circuits match driver characteristics to a ratio of an externally provided resistance. The CSR's control the ratio.

The outbound data path is relatively simple compared to the inbound. On the outbound command, the I/O circuits provide a minimum latency, registered path to transfer commands from the core to the DDR command bus as quickly as possible, but with tight timing control. For the DQ, DQS, and DRAM clock outputs, the I/O provides registered paths to transfer these signals to the DDR bus with the proper phase relationship to the command. The phase of the DRAM clocks (along with the DQS and DQ phase) relative to the command can be controlled by the ddr1xphsel and ca2xphsel fields of the spdpar45cur and spdpar67cur CSR's.

The inbound data path includes calibrated receiver enable circuits, calibrated DQS delay lines, and an eight entry deep levelization FIFO. Calibration is controlled by the core at power up and can take several ms to execute. Calibration involves a series of read and write operations to the DRAM. Receiver enable is calibrated on a per byte basis. DQS delay is calibrated at a coarse level on a word basis, with a fine calibration adjustment for each nibble. The fine adjustment can be dynamic so that a different calibration value can be sent from the core for each nibble depending on which rank is being accessed. The core provides a read pointer to access the contents of the FIFO. The IO I/O circuits manage the FIFO write pointer automatically, with timing based on both the receiver enable and DQS delay calibrations.

§

4 Electrical, Power, and Thermal

This chapter contains a description of the Intel 6400/6402 Advanced Memory Buffer (AMB)'s electrical DC parameters, timing parameters, power considerations, and thermal considerations.

4.1 Electrical DC Parameters

4.1.1 Absolute Maximum Ratings

Table 4-1 contains absolute maximum ratings over operating free-air temperature range (see Note 1).

Table 4-1. Absolute Maximum Ratings Over Operating Free-Air Temperature Range (See Note 1)

Symbol	Parameter	Min	Max	Unit
V_{DD}	Supply voltage DRAM Interface	–0.5	+2.3	V
$V_{IN\ (DDR2)}$, $V_{OUT\ (DDR2)}$	Voltage on any DDR2 interface pin relative to Vss (See Notes 2 and 3)	–0.5	+2.3	V
I_{INK} (V_{IN} < 0 or V_{IN} > V_{DD})	Input clamp current		±30	mA
I_{OUTK} (V_{OUT} < 0 or V_{OUT} > V_{DD})	Output clamp current		±30	mA
I_{OUT} (V_{OUT} = 0 to V_{DD})	Continuous output current		±30	mA
N/A	Continuous current through each V_{DD} or GND		±100	mA
V_{CC}	Supply voltage for Core and High Speed Interface	–0.3	+1.75	V
T_{stg}	Storage temperature range	–55	+100	°C

Notes:
1. Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.
2. The input and output negative-voltage ratings may be exceeded if the input and output clamp-current ratings are observed.This value is limited to 2.3 V maximum.

4.1.2 Operating DC Parameters

Table 4-2 contains the electrical DC parameters for the AMB part for normal operation.

Table 4-2. AMB Operating DC Electrical Parameters

Parameter	AMB Mode	Units	Min	Typ	Max
V_{CC} link / core		Volts	1.455	1.5	1.575
V_{DD}		Volts	1.7	1.8	1.9
V_{DDSPD}		Volts	3.0	3.3	3.6

Note: There will also be a V_{TT} termination supply at $V_{DDR}/2$ available on the DIMM but does not connect to the AMB.

4.1.3 AMB Power Specifications

Table 4-3 contains the AMB power specifications related to parameters stored in the SPD for the AMB .

Table 4-3. Power Values for x8 DIMMS (Sheet 1 of 3)

			533 MHz		667 MHz			
Symbol	**Conditions**	**Power Supply**	**Thermal Design**	**Max Current**	**Thermal Design**	**Max Current**	**Units**	**Notes**
Idd_Idle_0	Idle Current, single or last DIMM L0 state, idle (0 BW) Primary channel enabled, Secondary Channel Disabled CKE high. Command and address lines stable. DRAM clock active.	@1.5 V	2.1	2.2	2.4	2.6	A	
		@1.8 V	0.6	0.7	0.6	0.7	A	
		@3.3 V					A	
Idd_Idle_0 Total Power			3.5		4.0		W	
Idd_Idle_1	Idle Current, first DIMM L0 state, idle (0 BW) Primary and Secondary channels enabled CKE high. Command and address lines stable. DRAM clock active.	@1.5 V	2.7	3.0	3.1	3.4	A	
		@1.8 V	0.6	0.7	0.6	0.7	A	
		@3.3 V					A	
Idd_Idle_1 Total Power			4.6		5.1		W	
Idd_TDP_0 (for AMB spec, Not in SPD)	Active Power, TDP BW, Single or Last DIMM L0 state TDP Channel BW = 2.0GB/s@533; 2.4GB/s@667; DIMM BW = 2.0GB/s@533; 2.4GB/s@667; 67% read, 33% write. Primary channel Enabled Secondary channel Disabled CKE high. Command and Address	@1.5 V	2.4	2.6	2.8	3.0	A	
		@1.8 V	1.1	1.3	1.2	1.3	A	
		@3.3 V					A	
Idd_TDP_0 Total Power			5.2		5.8		W	
Idd_TDP_1 (for AMB spec, Not in SPD)	Active Power, TDP BW, First DIMM L0 state TDP Channel BW = 2.0GB/s@533; 2.4GB/s@667; DIMM BW =2/3 Channel BW = 1.3GB/s@533; 1.6GB/s@667; 67% read, 33% write. Primary channel Enabled Secondary channel Enabled CKE high. Command and Ad	@1.5 V	3.0	3.3	3.5	3.8	A	
		@1.8 V	0.9	1.0	0.9	1.0	A	
		@3.3 V					A	

Table 4-3. Power Values for x8 DIMMS (Sheet 2 of 3)

			533 MHz		667 MHz			
Symbol	**Conditions**	**Power Supply**	**Thermal Design**	**Max Current**	**Thermal Design**	**Max Current**	**Units**	**Notes**
Idd_TDP_1 Total Power			5.8		6.4		W	
Idd_Active_1	Active Power L0 state. 50% DRAM BW, 67% read, 33% write. Primary and Secondary channels enabled. DRAM clock active, CKE high.	@1.5 V	3.1	3.4	3.6	3.9	A	
		@1.8 V	1.2	1.3	1.2	1.3	A	
		@3.3 V					A	
Idd_Active_1 Total Power			6.4		7.1		W	
Idd_Active_2	Active Power, data pass through L0 state. 50% DRAM BW to downstream DIMM, 67% read, 33% write. Primary and Secondary channels enabled CKE high. Command and address lines stable. DRAM clock active.	@1.5 V	2.9	3.2	3.3	3.7	A	
		@1.8 V	0.6	0.7	0.6	0.7	A	
		@3.3 V					A	
Idd_Active_2 Total Power			5.0		5.6		W	
Idd_Training (for AMB spec, Not in SPD)	Training Primary and Secondary channels enabled. 100% toggle on all channel lanes DRAMs idle. 0 BW. CKE high, Command and address lines stable. DRAM clock active.	@1.5 V		3.5		4.0	A	
		@1.8 V		0.7		0.7	A	
		@3.3 V					A	
Idd_Training Total Power							W	
Idd_IBIST (for AMB spec, Not in SPD)	IBIST Over all IBIST modes DRAM Idle (0 BW) Primary channel Enabled Secondary channel Enabled CKE high. Command and Address lines stable DRAM clock active	@1.5 V		3.8		4.5	A	
		@1.8 V		0.7		0.7	A	
		@3.3 V					A	
Idd_IBIST Total Power							W	

Table 4-3. Power Values for x8 DIMMS (Sheet 3 of 3)

			533 MHz		667 MHz			
Symbol	**Conditions**	**Power Supply**	**Thermal Design**	**Max Current**	**Thermal Design**	**Max Current**	**Units**	**Notes**
Idd_MemBIST (for AMB spec, Not in SPD)	MemBIST Over all MemBIST modes >50% DRAM BW (as dictated by the AMB) Primary channel Enabled Secondary channel Enabled CKE high. Command and Address lines stable DRAM clock active	@1.5 V		3.3		3.8	A	
		@1.8 V		2.1		2.1	A	
		@3.3 V					A	
Idd_MemBIST Total Power							W	
Idd_EI (for AMB spec, Not in SPD)	Electrical Idle DRAM Idle (0 BW) Primary channel Disabled Secondary channel Disabled CKE low. Command and Address lines Floated DRAM clock active, ODT and CKE driven low	@1.5 V		2.0		2.5	A	
		@1.8 V		0.2		0.2	A	
		@3.3 V					A	
Idd_EI Total Power							W	
IDD_S3	S3 Current VDD = 1.9V VCC = 0V VTT = 0V Across process variations Across the operating TCASE temp range DIMM types, R/C A, B, C, D, E, H & J	VDD (1.8V)		75		75	mA	

Table 4-4 contains the AMB Power Specification Parameters for the Advanced Memory Buffer part in normal mode.

Table 4-4. Power Values for x4 DIMMs (Sheet 1 of 3)

			533 MHz		667 MHz		
Symbol	**Conditions**	**Power Supply**	**Thermal Design**	**Max Current**	**Thermal Design**	**Max Current**	**Units**
Idd_Idle_0	Idle Current, single or last DIMM L0 state, idle (0 BW) Primary channel enabled, Secondary Channel Disabled CKE high. Command and address lines stable. DRAM clock active.	@1.5 V	2.1	2.2	2.4	2.6	A
		@1.8 V	0.9	0.9	0.9	0.9	A
		@3.3 V					A
Idd_Idle_0 Total Power			3.9		4.4		W
Idd_Idle_1	Idle Current, first DIMM L0 state, idle (0 BW) Primary and Secondary channels enabled CKE high. Command and address lines stable. DRAM clock active.	@1.5 V	2.7	3.0	3.1	3.4	A
		@1.8 V	0.9	0.9	0.9	0.9	A
		@3.3 V					A
Idd_Idle_1 Total Power			4.9		5.5		W
Idd_TDP_0 (for AMB spec, Not in SPD)	Active Power, TDP BW, Single or Last DIMM L0 state TDP Channel BW = 2.0GB/s@533; 2.4GB/s@667; DIMM BW = 2.0GB/s@533; 2.4GB/s@667; 67% read, 33% write. Primary channel Enabled Secondary channel Disabled CKE high. Command and Address	@1.5 V	2.4	2.6	2.8	3.0	A
		@1.8 V	1.5	1.6	1.5	1.6	A
		@3.3 V					A
Idd_TDP_0 Total Power			5.9		6.5		W
Idd_TDP_1 (for AMB spec, Not in SPD)	Active Power, TDP BW, First DIMM L0 state TDP Channel BW = 2.0GB/s@533; 2.4GB/s@667; DIMM BW =2/3 Channel BW = 1.3GB/s@533; 1.6GB/s@667; 67% read, 33% write. Primary channel Enabled Secondary channel Enabled CKE high. Command and Ad	@1.5 V	3.0	3.3	3.5	3.8	A
		@1.8 V	1.3	1.4	1.3	1.4	A
		@3.3 V					A
Idd_TDP_1 Total Power			6.3		6.9		W
Idd_Active_1	Active Power L0 state. 50% DRAM BW, 67% read, 33% write. Primary and Secondary channels enabled. DRAM clock active, CKE high.	@1.5 V	3.1	3.4	3.6	3.9	A
		@1.8 V	1.6	1.7	1.6	1.7	A
		@3.3 V					A
Idd_Active_1 Total Power			6.9		7.6		W

Table 4-4. Power Values for x4 DIMMs (Sheet 2 of 3)

			533 MHz		667 MHz		
Symbol	**Conditions**	**Power Supply**	**Thermal Design**	**Max Current**	**Thermal Design**	**Max Current**	**Units**
Idd_Active_2	Active Power, data pass through L0 state. 50% DRAM BW to downstream DIMM, 67% read, 33% write. Primary and Secondary channels enabled CKE high. Command and address lines stable. DRAM clock active.	@1.5 V	2.9	3.2	3.3	3.7	A
		@1.8 V	0.9	0.9	0.9	0.9	A
		@3.3 V					A
Idd_Active_2 Total Power			5.5		6.1		W
Idd_Training (for AMB spec, Not in SPD)	Training Primary and Secondary channels enabled. 100% toggle on all channel lanes DRAMs idle. 0 BW. CKE high, Command and address lines stable. DRAM clock active.	@1.5 V		3.5		4.0	A
		@1.8 V		0.9		0.9	A
		@3.3 V					A
Idd_Training Total Power							W
Idd_IBIST (for AMB spec, Not in SPD)	IBIST Over all IBIST modes DRAM Idle (0 BW) Primary channel Enabled Secondary channel Enabled CKE high. Command and Address lines stable DRAM clock active	@1.5 V		3.8		4.5	A
		@1.8 V		0.9		0.9	A
		@3.3 V					A
Idd_IBIST Total Power							W
Idd_MemBIST (for AMB spec, Not in SPD)	MemBIST Over all MemBIST modes >50% DRAM BW (as dictated by the AMB) Primary channel Enabled Secondary channel Enabled CKE high. Command and Address lines stable DRAM clock active	@1.5 V		3.3		3.8	A
		@1.8 V		2.4		2.4	A
		@3.3 V					A
Idd_MemBIST Total Power							W

Table 4-4. Power Values for x4 DIMMs (Sheet 3 of 3)

			533 MHz		667 MHz		
Symbol	**Conditions**	**Power Supply**	**Thermal Design**	**Max Current**	**Thermal Design**	**Max Current**	**Units**
Idd_EI (for AMB spec, Not in SPD)	Electrical Idle DRAM Idle (0 BW) Primary channel Disabled Secondary channel Disabled CKE low. Command and Address lines Floated DRAM clock active, ODT and CKE driven low	@1.5 V		2.0		2.5	A
		@1.8 V		0.2		0.2	A
		@3.3 V					A
Idd_EI Total Power							W
IDD_S3	S3 Current VDD = 1.9V VCC = 0V VTT = 0V Across process variations Across the operating TCASE temp range DIMM types, R/C A, B, C, D, E, H & J	VDD (1.8V)		75		75	mA

Notes:
1. Vdd : Thermal Design = 1.845 V (+2.5%) ; Max Current = 1.900 V (+5.5%)
2. Vcc : Thermal Design = 1.530 V (+2%) ; Max Current = 1.575 V (+5%)
3. Includes all DIMM DRAM organizations (SRx4, DRx4, SRx8, DRx8) and Raw Cards (A, B, C, D, E, H, J)
4. For x8, measured with DRx8 raw card B with 39 ohm termination for command, address, and clocks, and 47 ohm termination for CS and CKE
5. For x4, measured with DRx4 raw card E with parallel 33 ohm termination for command, address, and 39 ohm termination for CS, CKE and clocks
6. Cards using smaller termination resistors will have higher powers. for example, x4 cards parallel 22 ohm termination for command and address could add as much as 0.4W to the power for all states.
7. Total DIMM current during S3 includes AMB IDD_S3 and self-refresh current of all DRAM devices.

4.2 FB-DIMM Electrical Timing Specifications

The FB-DIMM Channel link electrical interface is more completely described in the *High Speed Differential Point-to-Point Link at 1.5V for Fully Buffered DIMM* specification. Refer to this document for recommended operating conditions.

Table 4-5 contains the FB-DIMM electrical timing specifications.

Table 4-5. AMB FB-DIMM Timing/Electrical

Symbol	Parameter	Units	Min	Typ	Max	Notes
tEI Propagate	EI Assertion Pass-Thru Timing	clks			4	
tEID	EI Deassertion Pass-Thru Timing	clks			tBitLock	
tEI	EI Assertion Duration	clks	100			1
tBitLock	Bit Lock Interval	frames			119	1
tFrameLock	Frame Lock Interval	frames			154	1

Notes:
1. Defined in FB-DIMM Architecture and Protocol Spec
2. Clocks defined as core clocks = 2x SCLK input

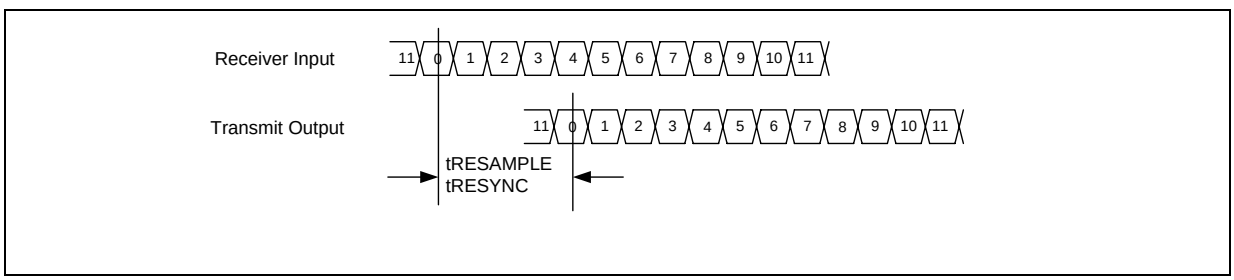
Table 4-6. AMB FB-DIMM Latency

Symbol	Parameter	Date Rate	Min	Max	Units	Comments
tC2D_AMB	CMD2DATA = 0x36 CMD2DATA = 0x40	533	18.7	22.3	nS	
		667	16.2	19	nS	
tC2D_AMB	CMD2DATA = 0x40 CMD2DATA = 0x46	533	20.5	24.2	nS	
		667	17.7	20.5	nS	
CMD2DATA	R/C B R/C B	533	0x36	0x48	nS	
		667	0x40	0x50	nS	
tRESAMPLE	Resample Delay	533	0.9	1.6	nS	1
		667	0.9	1.4	nS	1
tRESYNC	Resync Delay	533	2.3	3.9	nS	2
		667	2	3.2	nS	2

Note:
1. tRESAMPLE is the delay from the southbound input to the southbound output, or the northbound input to the northbound output when in resample mode, measured from the center of the data eye.
2. tRESYNC is the delay from the southbound input to the southbound output, or the northbound input to the northbound output when in resync mode, measured from the center of the data eye.

.

Figure 4-1. Latency Timing Diagrams



4.3 DDR2 DRAM Interface Electrical Specifications

Table 4-7 contains the electrical DC parameters for the AMB DDR2

Table 4-7. Recommended Operating Conditions for DRAM Interface

Symbol	Parameter	Min	Nom	Max	Unit
V_{DD}	Supply voltage	1.7	1.8	1.9	V
V_{REF}	Input reference voltage	0.49 * V_{DD}	0.50 * V_{DD}	0.51 * V_{DD}	mV
V_{TT}	Termination voltage	V_{REF} – 40	V_{REF}	V_{REF} + 40	mV
Single Ended Signals					
V_{IN}	Input voltage	0	-	V_{DD}	
$V_{IH(dc)}$	DC HIGH-level input voltage	V_{DD} / 2 + 100	-	V_{DD} + 300	mV
$V_{IL(dc)}$	DC LOW-level input voltage	-300	-	V_{DD} / 2 - 100	mV
$V_{IH(ac)}$	AC HIGH-level input voltage	V_{DD} / 2 + 200	-	-	mV
$V_{IL(ac)}$	AC LOW-level input voltage	-	-	V_{DD} / 2 – 200	mV
V_{OH}	Minimum Required Output Pull-up under AC Test Load	V_{DD} / 2 + 575	-	-	mV
V_{OL}	Maximum Required Output Pull-down under AC Test Load	-	-	V_{DD} / 2 - 575	mV
V_{OTR}	Output Timing Measurement Reference Level	-	0.5*V_{DD}	-	V
$I_{OH(dc)}$	Output minimum source dc current	-13.8	-	-	mA
$I_{OL(dc)}$	Output minimum sink dc current	13.8	-	-	mA
Differential Signals					
$V_{ID(dc)}$	DC differential input voltage	0.2	-	-	V
$V_{ID(ac)}$	AC differential input voltage	0.4	-	-	V
$V_{IX(ac)}$	AC differential input crossing voltage	0.5 * VDD - 0.100	-	0.5 * VDD + 0.100	V
Vr	Input timing measurement reference level		$V_{IX(ac)}$		
$V_{OX(ac)}$	AC differential output crossing voltage	0.5 * VDD - 0.100	-	0.5 * VDD + 0.100	V
$V_{OUT\ (slew)}$	Output slew-rate requirement	2.2	-	3.2	V/ns
Iih	Input leakage current (HIGH)	-	-	10	µA
Iil	Input leakage current (LOW)	-	-	10	µA
C_{IO}	Input/Output Capacitance	2.0	-	2.5	pF
R_{OUT}	Output Impedance	13	-	20	Ohms
T_C	Package surface (case) temperature for AMB	-	-	110	^oC

Notes:
1. Values highlighted in 'Red' are for reference (that is, placeholder value)
2. No V_{REF} pin on AMB
3. $V_{OUT\ (slew)}$ covers all other outputs slew rate including clock
4. Input voltage for all pins is limited to a maximum of 2.3 V.
5. VDD/2 = 1.7/2 = 850 mV; V_{OUT} = 575 mV. (V_{OUT} - VDD/2)/I_{OH} must be less than 20 Ohm for values of VOUT between VDD/2 and VDD/2 - 275 mV.
6. VDD/2 = 1.7/2 = 850 mV; V_{OUT} = 275 mV. V_{OUT}/I_{OH} must be less than 20 Ohm for values of VOUT between 0V and 275 mV.
7. C_{IO} is the Input/Output capacitance for DQ/DQS, and Output capacitance for CMD/ADDR/CK.

4.4 DDR2 Electrical Output Timing Specifications

4.4.1 Description of DQ/DQS Alignment

The DQS output rising edge aligns with the CLK output rising edge and the DQS output falling edge aligns with the CLK output falling edge. The DQ outputs are 1/4 cycle offset from the DQS outputs.

DQ/DQS inputs are edge aligned and will be skewed by internal receiver DLL. Inputs are terminated on-die by a resistive circuit during reads only.

4.4.2 Description of ADD/CMD/CNTL Outputs

ADD/CMD/CNTL outputs can be adjusted relative to CLK (see Table 4-8.) to improve setup or hold times. The value of this delay is fixed at boot time. These outputs are either aligned with CLK falling or with a certain timing offset before CLK falling. The amount of offset is implementation specific (for example, can be a constant timing offset, or a known ratio of the DRAM clock period).

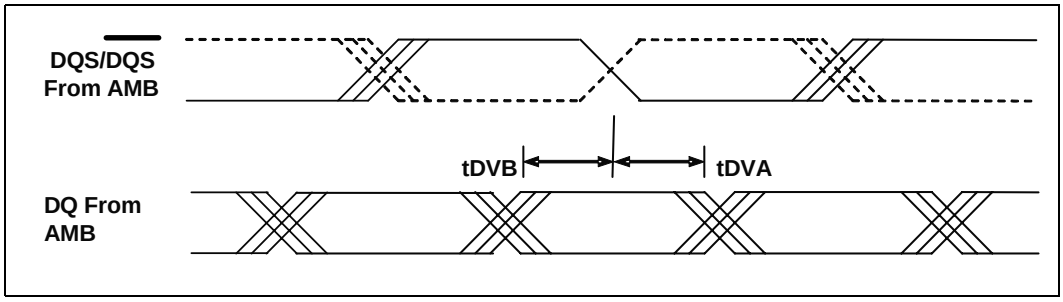
4.4.3 Test Load Specification

DDR2 timings are specified for a 25 Ohm test load terminated to Vdd/2, measured at the Advance Memory Buffer component package pins.

4.4.4 tDVA and tDVB Parameter Description

The timing parameters tDVA and tDVB indicate the time the DQ is valid after or before DQS. tDVA is used to indicate the time that Data is Valid After. tDVA is used for DQ/DQS write hold calculations (tDH). tDVB is used to indicate the time that Data is Valid Before. tDVB is used for DQ/DQS write setup calculations (tDS).

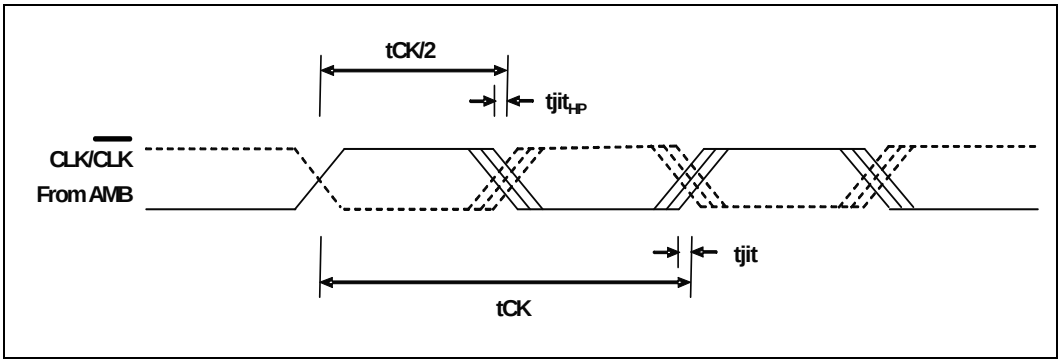
Figure 4-2. tDVA and tDVB Timing Diagram



4.4.5 tjit and $tjit_{HP}$ Parameter Description

The parameter tjit is the full period jitter, and $tjit_{HP}$ is the half-period jitter.

Figure 4-3. tjit and $tjit_{HP}$ Timing Diagram



4.4.6 tCVA, tCVB, tECVA and tECVB Parameter Description

The parameters tCVA and tCVB specify the time that command is valid after and before CLK. tCVA stands for the time that the Command is Valid After the CLK/$\overline{CLK}$ crossing point. tCVA is used for CA/CLK hold calculations (tIH). tCVB stands for the time that the Command is Valid Before the CLK/$\overline{CLK}$ crossing point. tCVB is used for CA/CLK setup calculations (tIS). Table 4-4 shows tCVA and tCVB.

tECVA and tECVB apply in early mode. For DIMMs with 36 devices, the command, address and control signals can be shifted by 1/6 clock in early mode. Table 4-5 shows tECVA and tECVB.

Figure 4-4. tCVA and tCVB Timing Diagram

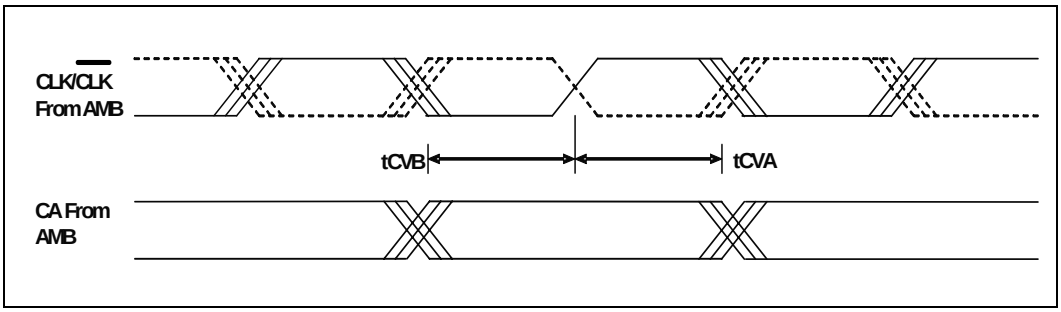
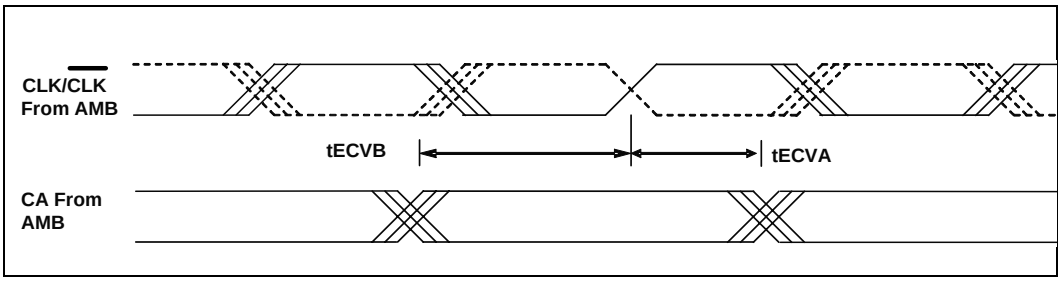


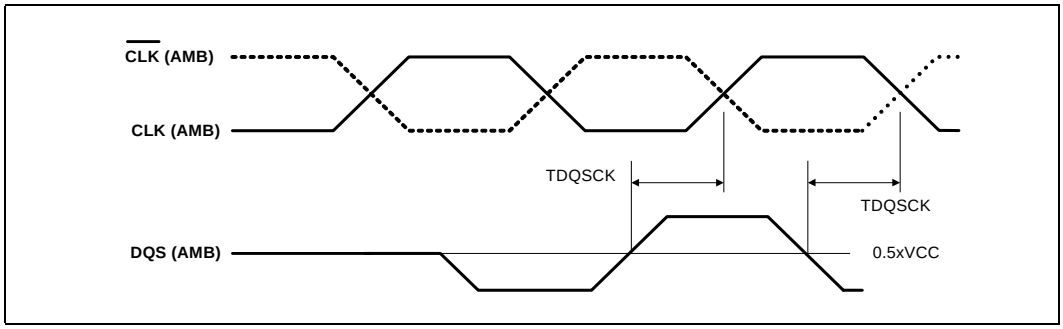
Figure 4-5. tECVA and tECVB Timing Diagram



4.4.7 tDQSCK Timing Parameter Description

tDQSCK indicated the CLK to DQS delay. This value is used for tDQSS, tDSS and tDSH timing calculations. In order to determine these numbers accurately, package parameters must be taken into account. This adjusts the minimum time by -110 ps and the maximum by -70 ps.

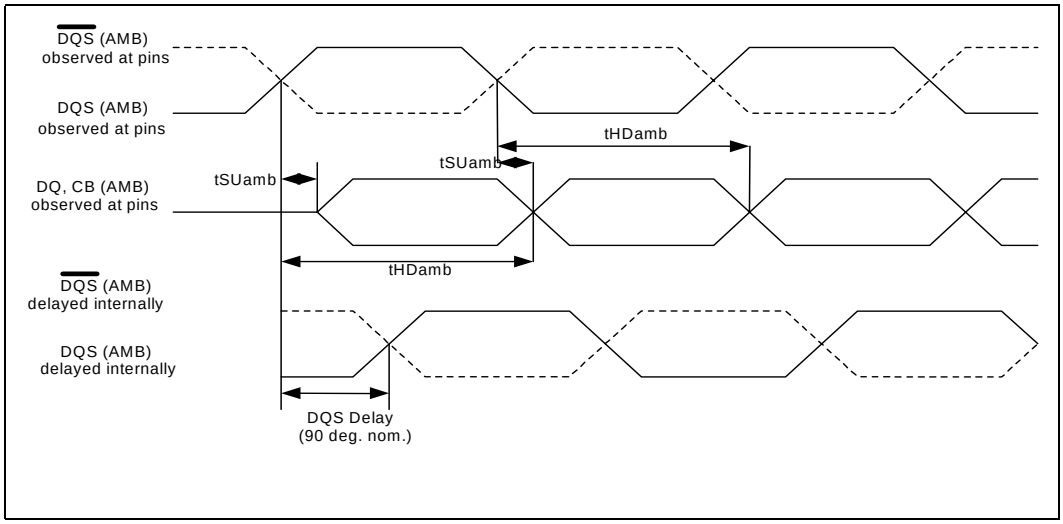
Figure 4-6. TDQSCK Timing Diagram



4.4.8 DQ and CB (ECC) Setup/Hold Relationships to/from DQS (Read Operation)

Table 4-7 shows the timing diagram for tHDamb and tSUamb. The data is launched from the DRAM "edge aligned," meaning that the DQ data signals switch coincident with the DQS strobe rising and falling edges. Internal to the Advance Memory Buffer, the DQS strobe is delayed by approximately a quarter clock, and this delayed clock is then used to capture the DQ data. Thus, the setup time tSUamb is negative, meaning that the data can arrive at the Advance Memory Buffer inputs after the strobe, and tHDamb is greater than a quarter clock, so that the data will not change until after it has been captured by the internally delayed strobe. The Advance Memory Buffer determines the correct internal delay of strobe DQS based on a search of the data eye during the initialization of the system. The tHDamb and tSUamb specifications are based on an idealized data eye, where the search delays the strobe by exactly one quarter clock. The sum of tHDamb and tSUamb is equivalent to the minimum data valid window at the Advance Memory Buffer inputs.

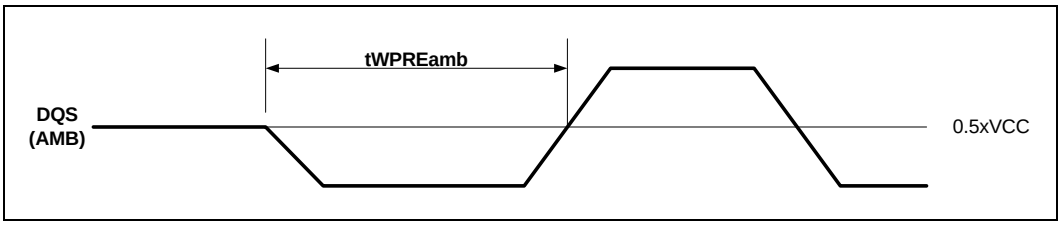
Figure 4-7. DQ and CB (ECC) Setup/Hold Relationship to/from DQS Timing Diagram



4.4.9 Write Preamble Duration

The write preamble duration is the measurement from the point when DQS and $\overline{DQS}$ start to be driven, to the crossing point of DQS and $\overline{DQS}$. Typically, to determine when DQS and $\overline{DQS}$ are starting to be driven, timing measurements are made at 0.5xVCC +/ - 50 mV, and 0.5xVCC +/- 100 mV, and then the measurements are linearly extrapolated back to 0.5xVCC.

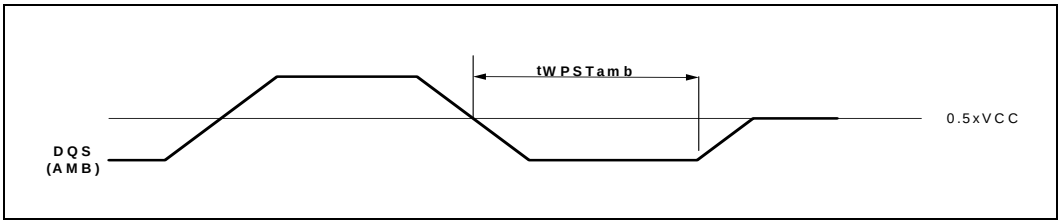
Figure 4-8. Write Preamble Duration Timing Diagram



4.4.10 Write Postamble Duration

The write postamble duration is the measurement from the crossing point of DQS and $\overline{DQS}$, to the point where DQS and $\overline{DQS}$ start to go into a high impedance state. Typically, to determine when DQS and $\overline{DQS}$ are starting to go into a high impedance state, for DQS, timing measurements are made at Vlow + 50 mV, and Vlow + 100 mV, and then the measurements are linearly extrapolated back to Vlow. For $\overline{DQS}$, timing measurements are made at Vhigh - 50 mV, and Vhigh - 100 mV, and then the measurements are extrapolated back to Vhigh.

Figure 4-9. Write Postamble Duration Timing Diagram



4.4.11 Advance Memory Buffer Component Electrical Timing Summary

Table 4-8 and Table 4-9 contain the electrical timing specifications for the Advance Memory Buffer component DDR2 interface.

Table 4-8. Advance Memory Buffer Component DDR2 Electrical Timing Specifications

Symbol	Parameter	DDR2 667		DDR2 533		Units	Fig #
		Min	Max	Min	Max		
System Memory Clock Timings							
t_{CK}	Ideal clock (CK) period	3.0		3.75		ns	
tCH	CK high time	1.35		1.70		ns	
tCL	CK low time	1.35		1.70		ns	
tjit	CK cycle to cycle Jitter		150		175	ps	4-7
$tjit_{HP}$	CK half-cycle jitter		150		175	ps	4-7
TDQSCK	Clock rising edge to DQS rising edge, or clock falling edge to DQS falling edge -- includes -110/-70 ps for package	-200	20	-210	30	ps	4-10
System Memory Address/Command/Control Signal Timings (Normal)							
tCVB	CMD/ADD/CNTL output valid before CLK/$\overline{CLK}$	1260		1615		ps	4-8
tCVA	CMD/ADD/CNTL output valid after CLK/$\overline{CLK}$ -- includes -140 ps for package	1120		1475		ps	4-8
System Memory Address/Command/Control Signal Timings (Early)							
tECVB	Early CMD/ADD/$\overline{CNTL}$ output valid before CLK/$\overline{CLK}$	1760		2240			4-9
tECVA	Early CMD/ADD/$\overline{CNTL}$ output valid after CLK/$\overline{CLK}$ -- includes -140 ps for package	620		850			4-9
System Memory Data and Strobe Signal Timings							
tDVB	DQ[63:0], CB[7:0], valid before DQS[15:0]/ DQS[15:0] crossing	575		750		ps	4-6
tDVA	DQ[63:0], CB[7:0], valid after DQS[15:0]/DQS[15:0] crossing	575		750		ps	4-6
tDOPW	DQ[63:0]. CB[7:0] Output Valid Pulse Width	1.35		1.70		ns	
tSU_{AMB}	DQ and CB Input Setup Time to DQS Crossing	-530		-700		ps	4-11
tHD_{AMB}	DQ and CB Input Hold Time After DQS Crossing	970		1180		ps	4-11
$tWPRE_{AMB}$	DQS Write Preamble Duration	2.85	3.5	3.58	4.25	ns	4-12
$tWPST_{AMB}$	DQS Write Postamble Duration	1.35	1.65	1.7	2.05	ns	4-13

4.4.12 Reference DDR2 Interface Package Trace Lengths

The following reference package trace lengths have been incorporated into the Advance Memory Buffer timings in Table 4-9.

Table 4-9. Advance Memory Buffer DDR2 Package Lengths

Signal Group	Min Length	Max Length	Units
CLK/$\overline{\text{CLK}}$	24	26	mm
Command/Address	4	20	mm
CKE, CS#	9	22	mm
ODT	10	16	mm
DQS/$\overline{\text{DQS}}$/DQ	9	13	mm

4.5 SMBUS Interface

Table 4-10. Recommended Operating Conditions for SMBUS Interface

Symbol	Parameter	Min	Typ	Max	Unit	Comments
V_{DDSPD}	Supply voltage (SMBUS)	3.0	3.3	3.6	V	3.3v ±10%
V_{IL}	SMBus signal input low voltage	-	-	0.8	V	
V_{IH}	SMBus signal input high voltage	2.1	-	V_{DDSPD}	V	
V_{OL}	SMBus signal output low voltage	-	-	0.4	V	@I_{PULLUP}
$I_{LEAK-BUS}$	Input Leakage per bus segment	-	-	±200	µA	
$I_{LEAK-PIN}$	Input Leakage per device pin	-	-	±10	µA	
I_{PULLUP}	Current sinking, V_{OL} = 0.4V	4	-	-	mA	
C_{BUS}	Capacitive load per bus segment	-	-	400	pF	
C_I	Capacitance for SMBDAT or SMBCLK pin	-	-	10	pF	
V_{NOISE}	Signal noise immunity from 10MHz to 100MHz	300	-	-	mV p-p	This is AC item applies to the high-power DC specification only

Note: Based on High-power SMBus DC Specification

4.6 Miscellaneous I/O (1.5 Volt CMOS Driver)

Table 4-11. Recommended Operating Conditions for $\overline{\text{RESET}}$ and BFUNC Pins

Symbol	Parameter	Min	Typ	Max	Unit	Comments
$V_{IH(dc)}$	DC HIGH-level input voltage	1.0	-	-	V	
$V_{IL(dc)}$	DC LOW-level input voltage	-	-	0.5	V	
ILEAK	Input leakage	-	-	± 90	µA	

4.7 Thermal Diode and Analog to Digital Converter (ADC)

A thermal diode with an analog to digital converter (ADC) is required. The thermal sensor will be used to ensure prevention of catastrophic failure.

An 8-bit register will store the temperature. The sensor measures from 0 to 127 degrees C measured in 0.5 degree increments in a register with values from 0 to 255. Internal analog nodes (anode, cathode, and so forth) of this circuit will not be brought out to package pins to keep noise at a minimum.

Refer to the *Intel® 6400/6402 Advanced Memory Buffer Thermal Mechanical Design Guide* for more information on the thermal sensor.

4.7.1 Thermal Sensor Effects on the AMB's Functional Behavior

When enabled, the results of the thermal trip points TEMPLO and TEMPMID are reported in FBD Status 0 responses following Sync commands.

If the temperature exceeds TEMPHI, errors are logged. If the TEMPHIENABLE bit in the TEMPSTAT register is set, DDR shutdown occurs and FBD links go into electrical idle mode. This over TEMPHI behavior also applies when in LAI mode. See the error chapter for a more complete description of chip behavior when TEMPHI is exceeded.

A temperature TEMPSTAT.INCREASING bit is also generated depending on whether the temp is greater or less than the last time the INCREASING bit was sampled. Sampling to create INCREASING bit is controlled by writes to register UPDATED. When temperature is above TEMPMID, this information also shows up in the FBDS0.Thermal_Trip register field and in the Status response of FBD Sync commands targeted to FBDS0.

§

5 Debug and Logic Analyzer Mode

5.1 Logic Analyzer Interface (LAI) Mode

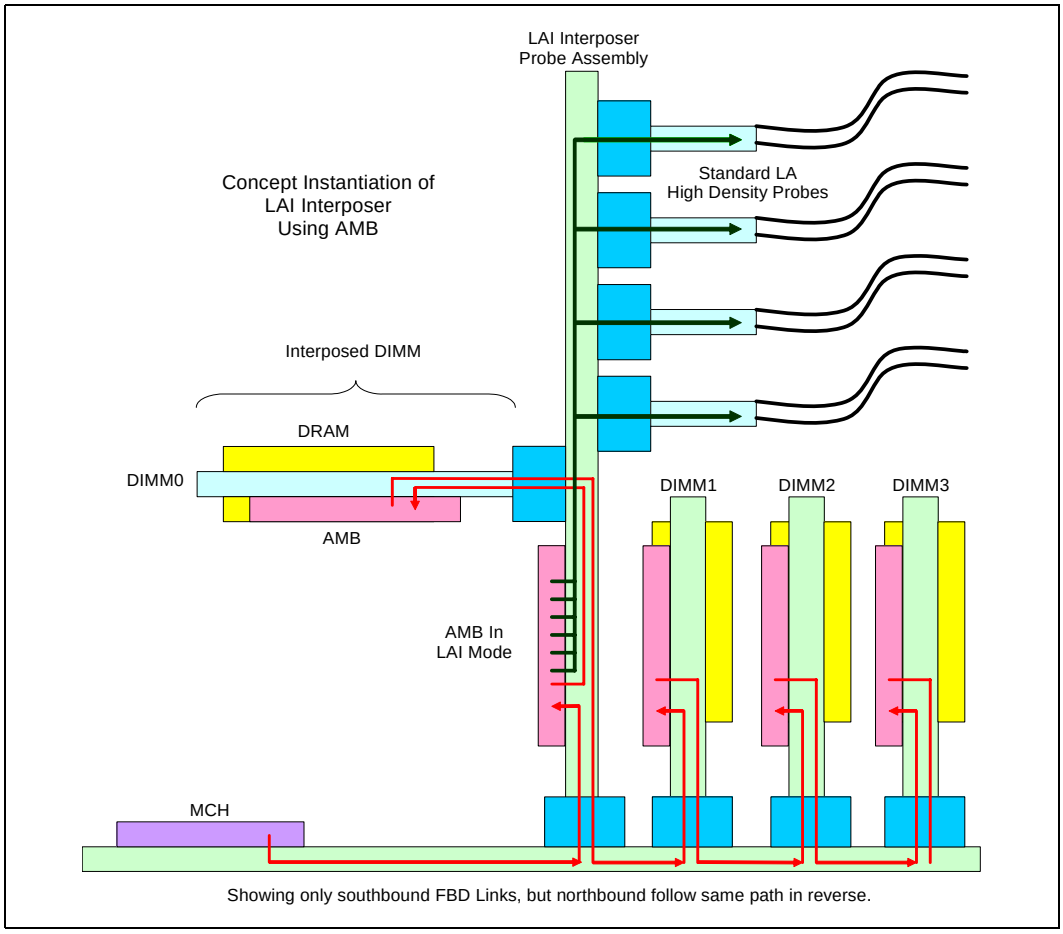
This section describes the functionality of the optional Logic Analyzer Mode for the Intel 6400/6402 Advanced Memory Buffer (AMB).

The features of the logic analyzer mode (LAI mode) in the Fully Buffered DIMM will enable select bus observability capabilities. The mode will be the basis of an LAI assembly to provide link traffic trace capability and will support design and manufacturing debug and validation needs for systems featuring FBD links. In this application, the device is programmed to perform the following functions:

- Repeater pass-through operation for southbound and northbound links
- Provides enough link protocol unwinding to allow extraction of framing boundaries, idle filtering opportunities, and pattern match triggering
- Performs logic analysis functions such as cross-triggering, match/mask, and local event detection. These functions cannot be effectively performed using a remote logic analyzer.
- Provides independent demuxed northbound and southbound traffic stream in LA compatible signal levels and timing format through reuse of the existing device DRAM I/O pins
- Provides SMB (or possibly another communications interface) access to allow external, non-intrusive access to component LAI control functions/parameters
- The AMB is commanded to power up in LAI mode when SMA[3:2] addresses are strapped to 2’b11 and LAI capability bit is enabled.

Figure 5-1 is a conceptual depiction of the AMB used in a LAI mode application.

Figure 5-1. AMB LAI Mode Usage Diagram



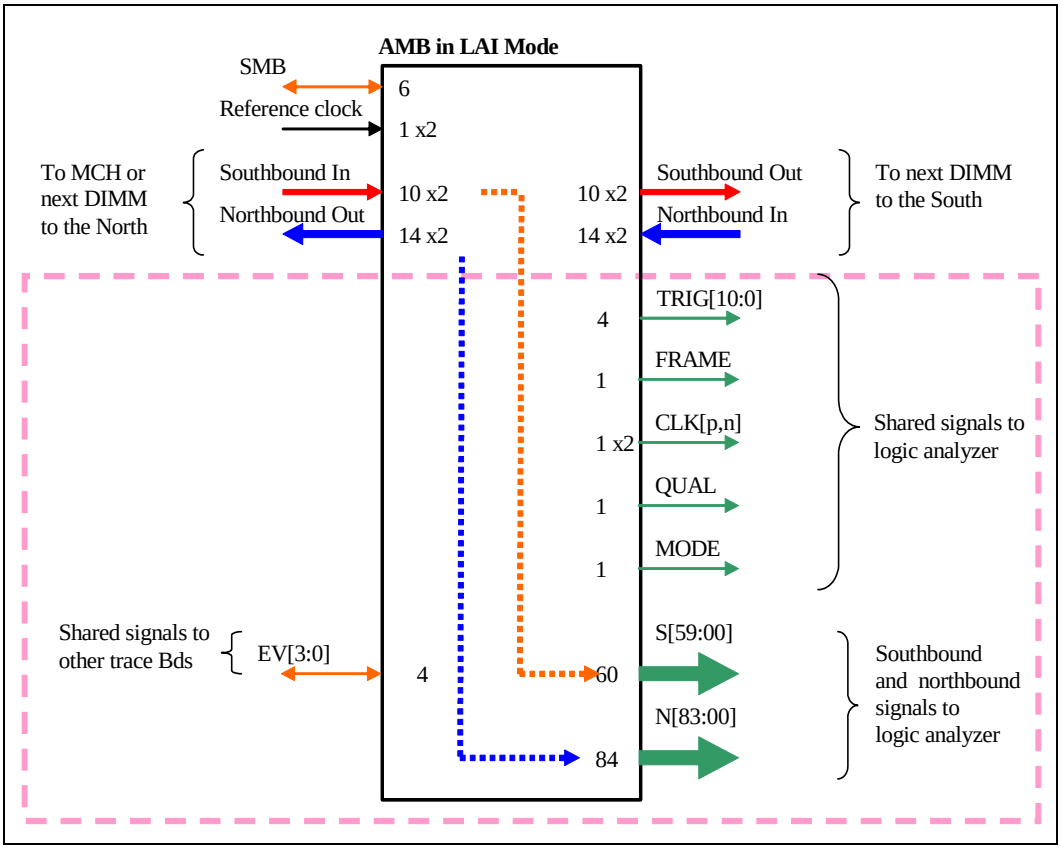
5.1.1 LAI Mode Architecture

The diagram below illustrates the AMB as a functional block when used as a LAI. The normal southbound and northbound links, the reference clock, and the SMB bus are used "as is". The channel traffic is reflected onto the DRAM interface with frame alignment for access with a logic analyzer.

To be effective in collecting useful FBD traces, the information provided to a logic analyzer must include not only a demuxed copy of the direct information transferred on the links, but also several types of derived information that a logic analyzer is not equipped to derive itself. These include specifically:

- Cross-triggering information with finer timing granularity than LA can achieve
- Simple filtering (qualified storage) opportunities recognition
- Traffic framing

Figure 5-2. AMB LAI Mode Connectivity



5.1.2 LAI Mode Clocking

To eliminate frequency drifts, the AMB on each DIMM and the chipset in an FBD channel will be provided with a reference clock that is from a clock source common to the FBD channel. A clock buffer will be placed on the LAI card to take the reference clock input and provide the required two reference clocks for the LAI AMB and the attached DIMM AMB.

5.1.3 LAI Mode Pins

The DDR pins designed in the AMB can be enabled to carry signals for the logic analyzer debug and validation of the FBD channel. The LAI mode is selected by strapping the following input pins on a AMB which has the LAI capability bit enabled.

SA[:0] = DIMM ID = 4b'11XX

The list below highlights the pins in the AMB that are dedicated for DDR, all of which are not required to operate at speed equal to the DDR data rate.

Address/Command pins are specially designed in the AMB to be able to drive double data rate to support the LAI functionality.

Table 5-1. DDR Pins Shared With LAI Functionality

Signal Type	Count	Speed (MHz)	Use in LAI Mode	Comment
DDR				
DDR DQ & DQS	108	533/667	~ LAI signal	DQS must run single ended in LAI mode
DDR Cmd/Addr	56	266/333	~ LAI signal	Most must run double speed in LAI mode Some may be dedicated to bidirectional Event Bus
Clocks to DRAMs	8	533/667	~ LAI signal	May run single ended in LAI mode
DDR Comp and analog	5	analog		Not suitable for LAI signals
Total DDR Pins	177			

Table 5-2. List of Pins Required to Enable Debug With LAI Functionality

Signal Type	Count	Speed (MHz)	Comment
Southbound			
S[59:0]	60	533/667	Data
Northbound			
N[83:0]	84	533/667	Data
MODE	1	267/333	
TRIG[10:0]	11	267/333	
CLK	4	267/333	Two Differential pairs
QUAL	1	533/667	
FRAME	1	533/667	
EV[3:0]	4	100	Bidirectional
Comp Pins	5	Analog	Required for signal integrity
Total Pins	171		

5.1.4 LAI Mode Signal Definitions

Table 5-3. LAI Mode Added Signals (Sheet 1 of 2)

Signal Type	Direction	Definition
S[59:0]	Out	Southbound demuxed (6x10) traffic
N[83:0]	Out	Northbound demuxed (6x14) traffic
MODE	Out	Link mode: 0 = after training complete 1 = before training complete
TRIG[10:0]	Out	Triggers to LA. This provides eleven unique trigger signals to the LA from the AMB LAI as a result of frame pattern matching, state matching, and/or cross-triggering events from other LAIs. This allows cooperating AMB LAIs to overcome (a) limitations of LA in recognizing serial traffic patterns that exceed LA pattern matching capabilities and, (b) relatively long latencies in cross-triggering between LA modules.

Table 5-3. LAI Mode Added Signals (Sheet 2 of 2)

Signal Type	Direction	Definition
CLK	Out	Logic analyzer reference clock (differential). This provides a LA compatible timing reference clock for capture of all signals to the LA, with fine capture phase adjust using the native LA clock edge offset capabilities.
QUAL	Out	Store qualifier: 1 = store 0 = do not store
FRAME	Out	Drives 1 when LAI pins are valid level. Should be connected to pull down on LAI board to detect when LAI pins are tri-stated and LA should ignore data.
EV[3:0]	I/O	Inter-AMB event bus for cross-triggering (wired-OR, high active, slow).This four bit event bus allows multiple AMB (and similar) LAIs to be programmed to inter-communicate locally detected matching and/or filtering opportunities events (cross-triggering and cross-qualification).

5.1.5 LAI to DDR Pin Mapping

Table 5-4 contains the LAI-to-DDR pin mapping.

Table 5-4. List of Shared DDR/LAI Pins (Sheet 1 of 2)

DDR Pin	Count	Speed Mbit/sec	LAI Mode	Comment
DQ[55:52], DQS[15], $\overline{DQS}$[15]	6	533/667	sbframe_data0[5:0], [11:6]	5:0 captured on rising edge of CLK, 11:6 captured on falling edge of CLK
DQ[59:56], DQS[7], $\overline{DQS}$[7]	6	533/667	sbframe_data1[5:0], [11:6]	
DQ[51:48], DQS[6], $\overline{DQS}$[6]	6	533/667	sbframe_data2[5:0], [11:6]	
DQ[39:36], DQS[13], $\overline{DQS}$[13]	6	533/667	sbframe_data3[5:0], [11:6]	
DQ[47:44], DQS[14], $\overline{DQS}$[14]	6	533/667	sbframe_data4[5:0], [11:6]	
DQ[35:32], DQS[4], $\overline{DQS}$[4]	6	533/667	sbframe_data5[5:0], [11:6]	
DQ[43:40], DQS[5], $\overline{DQS}$[5]	6	533/667	sbframe_data6[5:0], [11:6]	
$\overline{RAS}$B, A[10:9)B, DYBA[2:0]B	6	533/667	sbframe_data7[5:0], [11:6]	
A[6:2]B, A[0]B	6	533/667	sbframe_data8[5:0], [11:6]	
A[15:11],B A[8]B	6	533/667	sbframe_data9[5:0], [11:6]	
CKE[1:0]**B**, CS[1:0]**B,** ODT**B**,	5	267/333	trigger[10:6]	Will only change at 1/2 freq
BA[2]**A**	1	533/667	frame	1 if transferring first half of frame, 0 if second half
$\overline{CAS}$**B**, $\overline{WE}$**B**, A[7]**B**, A[1]**B**	4	100 MHz	evbus[3:0]	Inner DY bumpout rows
$\overline{CAS}$**A**	1	267/333	mode	
$\overline{RAS}$**A**	1	267/333	qual	
CKE[1:0]**A**, CS[1:0]**A**, ODT**A**, $\overline{WE}$**A**	6	267/333	trigger[5:0]	Will only change at 1/2 freq

Table 5-4. List of Shared DDR/LAI Pins (Sheet 2 of 2)

DDR Pin	Count	Speed Mbit/sec	LAI Mode	Comment
A[5:0]**A**	6	533/667	nbframe_data0[5:0], [11:6]	5:0 captured on rising edge of CLK, 11:6 captured on falling edge of CLK
A[11:6]**A**	6	533/667	nbframe_data1[5:0], [11:6]	
BA[1:0]**A,** A[15:12]**A**	6	533/667	nbframe_data2[5:0], [11:6]	
DQ[23:20], DQS[11], DQS[11]	6	533/667	nbframe_data3[5:0], [11:6]	
DQ[31:28], DQS[12], DQS[12]	6	533/667	nbframe_data4[5:0], [11:6]	
DQ[19:16], DQS[2], DQS[2]]	6	533/667	nbframe_data5[5:0], [11:6]	
DQ[27:24], DQS[3], DQS[3]	6	533/667	nbframe_data6[5:0], [11:6]	
DQ[15:12], DQS[10], DQS[10]	6	533/667	nbframe_data7[5:0], [11:6]	
DQ[7:4], DQS[9], DQS[9]	6	533/667	nbframe_data8[5:0], [11:6]	
DQ[11:8], DQS[1], DQS[1]	6	533/667	nbframe_data9[5:0], [11:6]	
DQ[3:0], DQS[0], DQS[0]	6	533/667	nbframe_data10[5:0], [11:6]	
CB[3:0], DQS[8], DQS[8]	6	533/667	nbframe_data11[5:0], [11:6]	
CB[7:4], DQS[17], DQS[17]	6	533/667	nbframe_data12[5:0], [11:6]	
DQ[63:60], DQS[16], DQS[16]	6	533/667	nbframe_data13[5:0], [11:6]	
CLK[3:2]	2	267/333 MHz	nc	Not supported in LAI Mode
$\overline{\text{CLK}}$[3:2]	2	267/333 MHz	nc	Not supported in LAI Mode
CLK[1:0]	2	267/333 MHz	LAI clock p [1:0]	
$\overline{\text{CLK}}$[1:0]#	2	267/333 MHz	LAI clock n [1:0]	
Total DDR Pins	162			No spare pins

5.1.6 FBD to LAI Signal Mapping

The following example show how an FBD Southbound Command Frame is transferred from FBD frame format to LA Interface early/late data.

The LAI interface delays the SB "A slot" by one clock to capture the FBD frame in the same "ABC" slot format that is sent from the host - rather than the "BCA" (B and C slots from host frame N-1 plus A slot from host frame N) used by the normal mode AMBs to minimize latency on the decode of slot A commands.

Table 5-5. Typical FBD Southbound Command Frame

Transfer \ Bit	9	8	7	6	5	4	3	2	1	0	
N 0	aE0	aE7	aE8	F0=0	aC20	aC16	aC12	aC8	aC4	aC0	Early Data
N 1	aE1	aE6	aE9	F1=0	aC21	aC17	aC13	aC9	aC5	aC1	
N 2	aE2	aE5	aE10	aE13	aC22	aC18	aC14	aC10	aC6	aC2	
N 3	aE3	aE4	aE11	aE12	aC23	aC19	aC15	aC11	aC7	aC3	
N 4	FE21	0	0	0	bC20	bC16	bC12	bC8	bC4	bC0	
N 5	FE20	0	0	0	bC21	bC17	bC13	bC9	bC5	bC1	
N 6	FE19	0	0	0	bC22	bC18	bC14	bC10	bC6	bC2	Late Data
N 7	FE18	0	0	0	bC23	bC19	bC15	bC11	bC7	bC3	
N 8	FE17	0	0	0	cC20	cC16	cC12	cC8	cC4	cC0	
N 9	FE16	0	0	0	cC21	cC17	cC13	cC9	cC5	cC1	
N 10	FE15	0	0	0	cC22	cC18	cC14	cC10	cC6	cC2	
N 11	FE14	0	0	0	cC23	cC19	cC15	cC11	cC7	cC3	

Output to logic analyzer is lane by lane

early data

[lane 9][lane 1][lane0]

[FE20,FE21,….,aE0] …… [bC5, bC4, aC7, aC6, aC5, aC4] [bC1, bC0, aC3, aC2, aC1, aC0]

late data

[lane 9][lane 1][lane0]

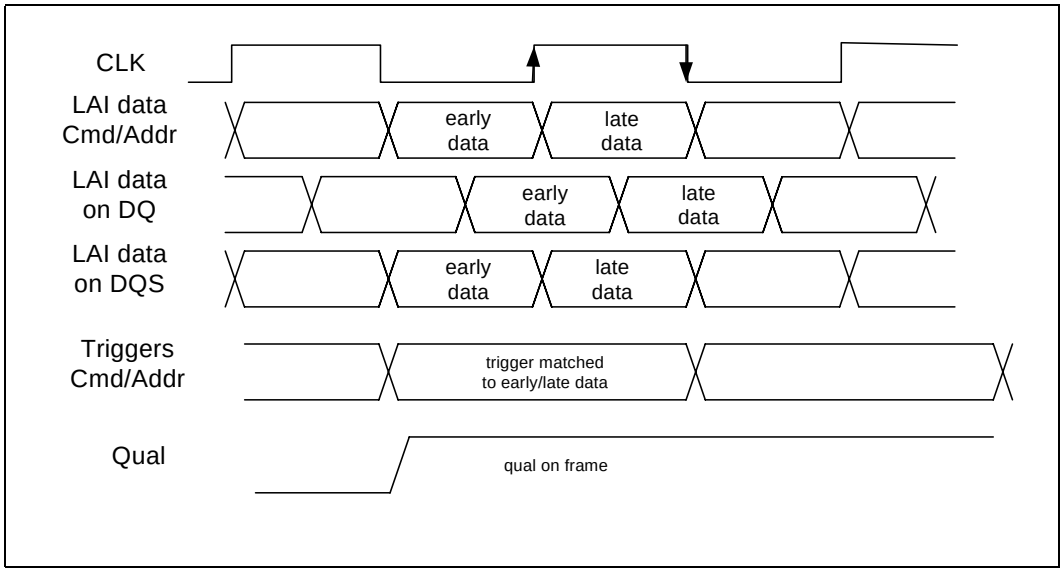
[FE14,FE15,….,FE19] … … [cC7, cC6, cC5, cC4, bC7, bC6] [cC3,cC2, cC1, cC0, bC3, bC2]

Note: CRC codes in A cmd (aE0 - aE13) are a function of FE21:FE14 of previous frame.

5.1.7 LAI to DDR Pin Timing

The phases of data presented to the logic analyzer have some odd timing due to reuse of some many different types of DDR I/O outputs to achieve the desired pin count. The LA will compensate for these predictable phase offsets.

Figure 5-3. LAI Signal Group Timing



5.1.8 LAI Features

5.1.8.1 Control and Status Registers (CSRs)

LAI mode CSRs will be accessed and programmed through SMBus when in LAI mode. See Chapter 14, "Registers," for complete details.

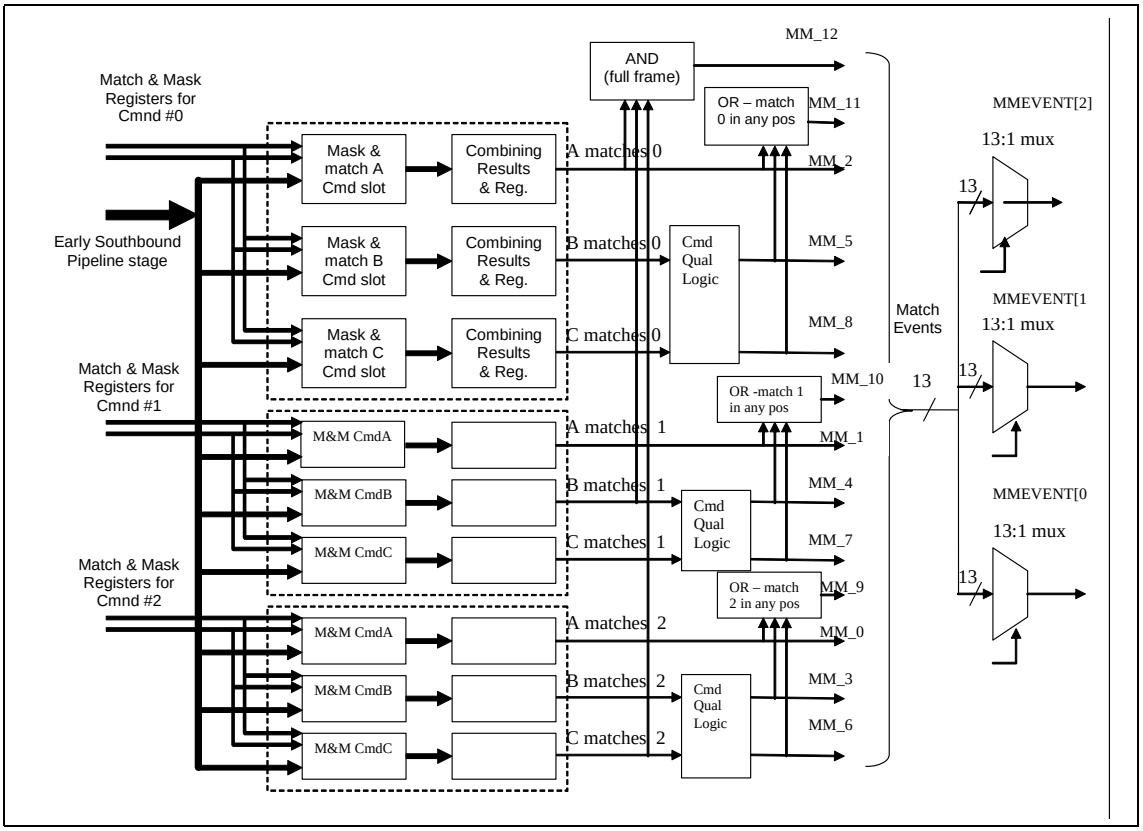
Note: LAI registers cannot be accessed in-band over the FBD link

5.1.8.2 Pattern Matching

The LAI block can pattern match on three command values at any of three command slots and combine the results into independent and combined local events. There are 13 total pattern matching events: a command value matches a command slot (9 events), a command value matches any slot (3 events), and all command values appear in the frame (1 event).

This pattern matching is also used in normal mode for error injection and NB in-band event generation.Figure 5-4 shows the LAI match and mask logic.

Figure 5-4. LAI Match and Mask Logic



5.1.8.2.1 Additional Qualification on Match/Mask

Full frame and A-slot matching is enabled part way through TS0, once FBD inputs have been aligned with the core clocking phases. Though generally, will not be used until link has completed initialization.

Slot B and Slot C pattern matching is further qualified so that true commands can be differentiated from data when not doing full frame matching.

For each mask and match pair 0, 1 or 2

- If Mask[39] = 1, then match and mask against any received data in Slot B and Slot C.
- If Mask[39] = 0 AND Match[39] = 0 then only match on commands.
  - — Ignore a match with contents of Slot B if
    Frame type is not Command or
    Frame type is command and A-command is Sync or Soft Reset
  - — Ignore a match with contents of Slot C if
    Frame type is not Command or
    Frame type is command and A-command is Sync or Soft Reset or
    B-command is Write Configuration Register

To match an FBD command that might occur in any slot including the A slot, mask out bits [38:24] and set both Mask[39] and Match[39] to 0.

- If Mask[39] = 0 AND Match[39] = 1 then only match on memory write data.
  - — Match and mask against received data in slot B and slot C if Frame type is Write Data Frame

5.1.8.3 Local Events

The mask/match features in the AMB LAI, and certain internal state and error conditions, will be used to generate local events. Global events propagated through the in-band debug and external event (EV) bus will also generate local events. All of the local events can be selected by muxes as trigger sources for LAI event signals and event bus signals. While not used in LAI mode, these events are also used in error injection and for sourcing events in the NB status frames. For exact details see the Configuration Registers Chapter.

An overview of the Local Events logic is shown below. The Match/Mask, Qualification and Event Bus blocks are described in more detail in other sections.

Figure 5-5. Local Event Mux Block Diagram

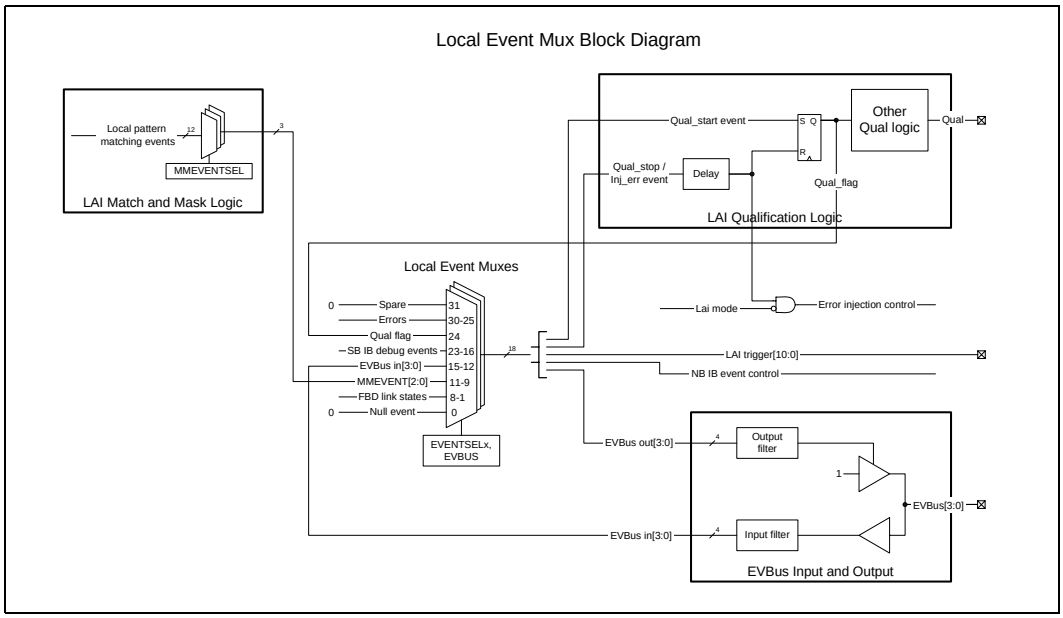


Table 5-6 shows the 32 local events. Each of these local events is logged in a configuration register and is sent to 19 32:1 muxes. The select lines for these muxes are programmed in configuration registers. Table 5-7 shows the destination (intended use) of the 19 selected events.

Table 5-6. LAI Local Events (Sheet 1 of 2)

Name	Events	Sel Addr	Description
Mask and Match	3	11:9	MMEVENT2:MMEVENT0 Slota, Slotb, Slotc, and Frame command matches - 3 events preselected from 13 possible matches
In-band debug	8	23:16	Received on SB link in-band EV[7:0]
EVBus events	4	15:12	Received on select DDR pins Event Bus EV[3:0]

Table 5-6. LAI Local Events (Sheet 2 of 2)

Name	Events	Sel Addr	Description
Initialization States	8	1:8	Disable[1], calibrate[2], training[3], testing[4], pollling[5], config[6], l0[7], l0s or recalibrate[8]
Errors	6	30:25	• SB/NB Fail Over mode [25], when unmasked: • SB CRC error[26], • Thermal overload[27], • Clock training violation (< 6 transitions in 512 UI) [28], • Unimplemented register access[29], • Other implementation specific errors[30] For the Intel 6400/6402 Advanced Memory Buffer: event[30] is the "OR" of any bit in FERR[7:4] or NERR[7:4]
Qual_Flag	1	24	
Spare	1	31	
NOP	1	0	Null Event
Total Events	32		For the Intel 6400/6402 Advanced Memory Buffer: There is enough space for 32 events

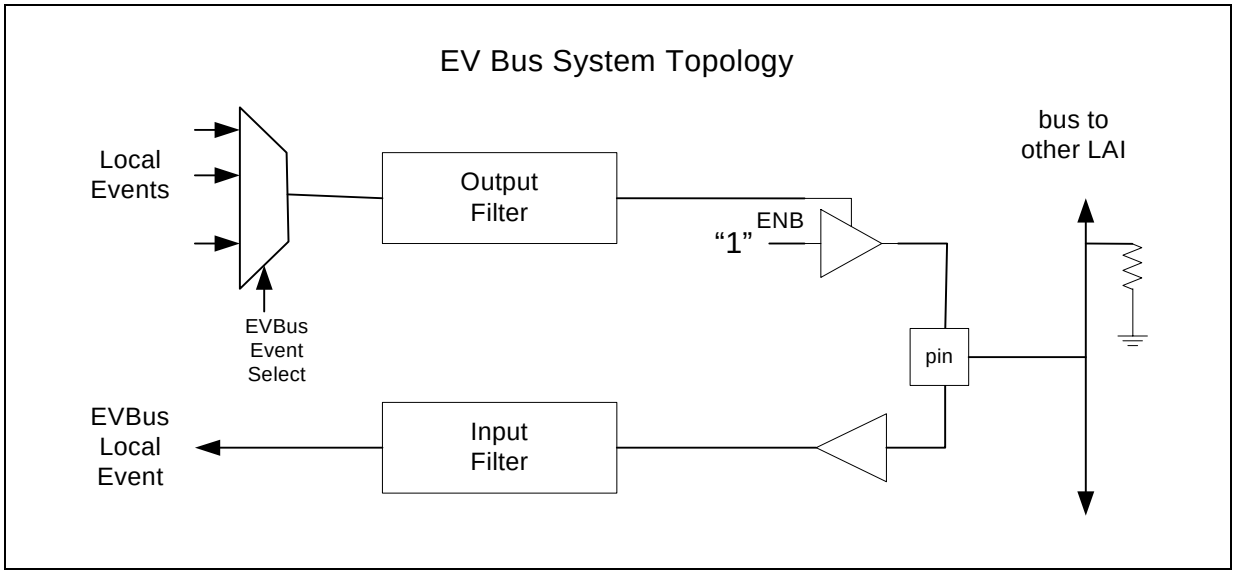
Table 5-7. LAI Event Selection

Name	Events	Description
Output event/triggers	11	Sent to LA on DDR pins
EVBus events	4	Sent on DDR pins
Inject Event NB	1	Assert NB event bit - not necessarily LAI usage
Error Injection Trigger	1	Inject errors - not necessarily LAI usage
Qual Events	2	Start and stop events for qualification signal
Total Events	19	

5.1.8.4 Event Bus

The AMB LAI mode enables four events signals (EV[3:0]) to be shared between the AMB or compatible LAI devices in a system. The signals are shared through a uniquely defined interconnect that connects all the devices to the 4-bit wide daisy chain bus. The 4 lanes are independent and carry separate events or triggers. Due to the noisy nature of the interconnect between LAI devices, filtering is required to eliminate spurious events from being introduced. A typical lane is shown in Figure 5-7 below.

Figure 5-6. EVBus Overview



Each lane in the bus can be selected by the EVTYPE parameter to be a pulse (trigger) event or a level (qualifier) event. Timing for the input and output filters is set by the EVT parameter which defines the T_{min} in core clock cycles. Both of these parameters reside in the EVBUS control register along with the local event select controls for each lane in the bus.

Pulse mode timing:

Input: Inputs are digitally filtered to reduce spurious events from bus ringing. Once a rising edge is detected a single one clock width pulse event is sent to the local events. No further pulse events are permitted until the next rising edge that occurs after the longer of either (2 * Tmin) or (T_source_event_actual + T min).

Output: An output high pulse of length Tmin or T_local event_actual (whichever is longer) is sourced whenever triggered by the selected local event. This is followed by an output low pulse of length Tmin. Further toggling by the selected local event is ignored until this output sequence is complete.

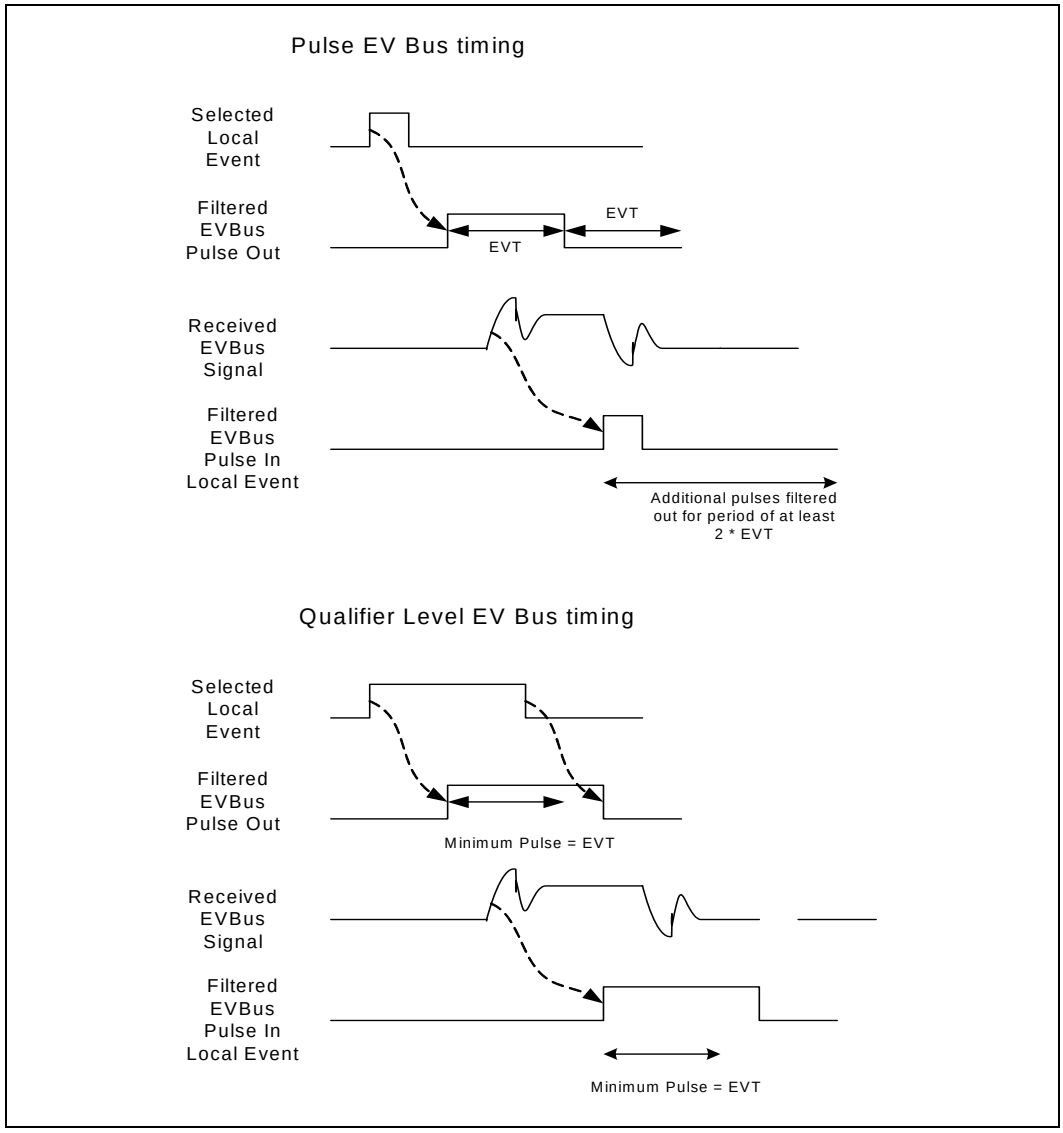
Level mode timing:

Input: Once a rising edge is detected the local event is asserted and remains asserted for Tmin or T_source_event_actual which ever is longer.

Output: An output high pulse of length Tmin or T_local event_actual (whichever is longer) is sourced whenever triggered by the selected local event.

Figure 5-7 shows the event signal timing.

Figure 5-7. Event Bus Signal Timing



The signals will be driven or captured by four DDR I/O buffers.

5.1.8.5 Qualification

The AMB in LAI mode sends a qualification signal, QUAL, on a DDR pin along with each frame. A qualified frame contains data likely to be of interest to the user of the logic analyzer. Conversely, an unqualified frame has been chosen to be filtered out because it only contains idle NOP frames or has otherwise been "flagged" as not occurring between pre-selected events. The logic analyzer can capture data when QUAL is asserted, and ignore data when QUAL is deasserted.

Once a qualified frame is seen, the QUAL signal is asserted, and it remains asserted for an additional programmable number of cycles, using a timer ranging from 0 to 63. This timer is restarted if it is already running when a new qualified frame is seen.

The error injection timer will be used to push out the triggering of the QUAL_STOP signal by N timer count of clock cycles (where N is user programmed for delay range of 0 to 63), thus increasing the length of the QUAL_FLAG interval. One QUAL_START event will enable the assertion of QUAL_FLAG and another QUAL_STOP event will disable assertion of QUAL_FLAG.

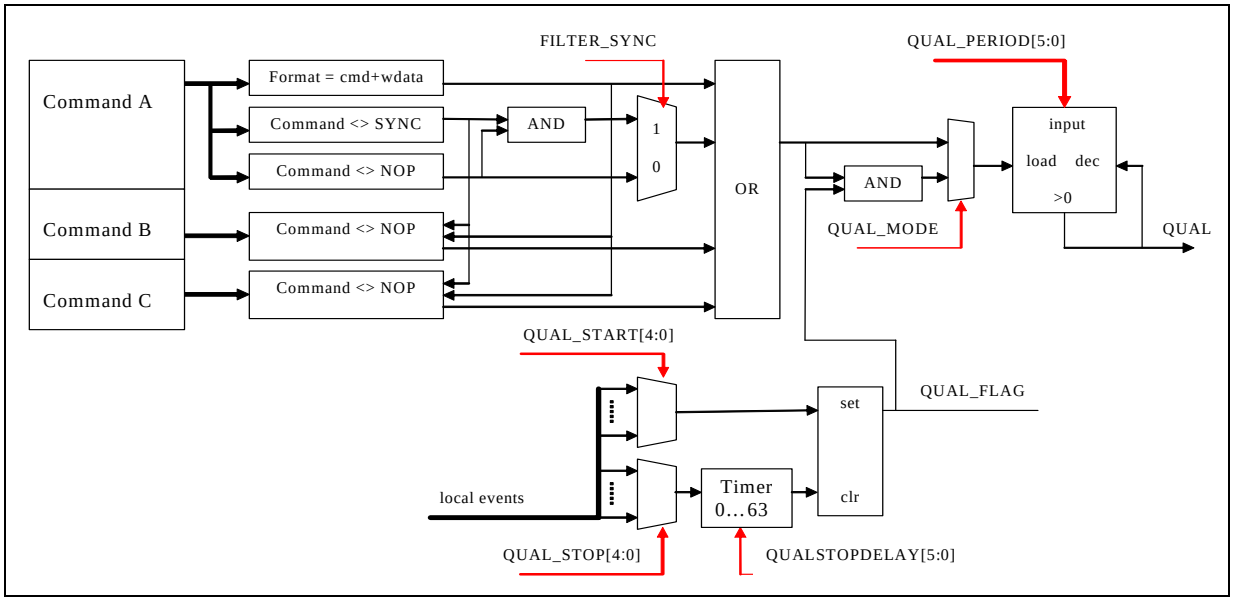
A frame of all NOPs is not a qualified frame. A sync frame is not a qualified frame if the FILTER_SYNC configuration register bit is set; otherwise it is considered qualified.

Note: The ability to determine which frames are qualified may be lost following an unmasked CRC or Few Edges error. This is the result of the architected behavior that future commands following the error will ignored. As a side effect, QUAL may stay high once these errors are detected.

If the QUAL_MODE configuration register bit is set, frames must additionally occur between QUAL_START and QUAL_STOP events to be qualified. A frame that triggers QUAL_START may also cause the QUAL signal to assert, and a frame that triggers QUAL_STOP will not be considered qualified. The QUAL_START and QUAL_STOP events are programmable and are selected from the 32 local events.

Figure 5-8 is a block diagram of the qualification signal control logic.

Figure 5-8. LAI Qualification Signal Block Diagram



5.1.9 LAI Block Diagram

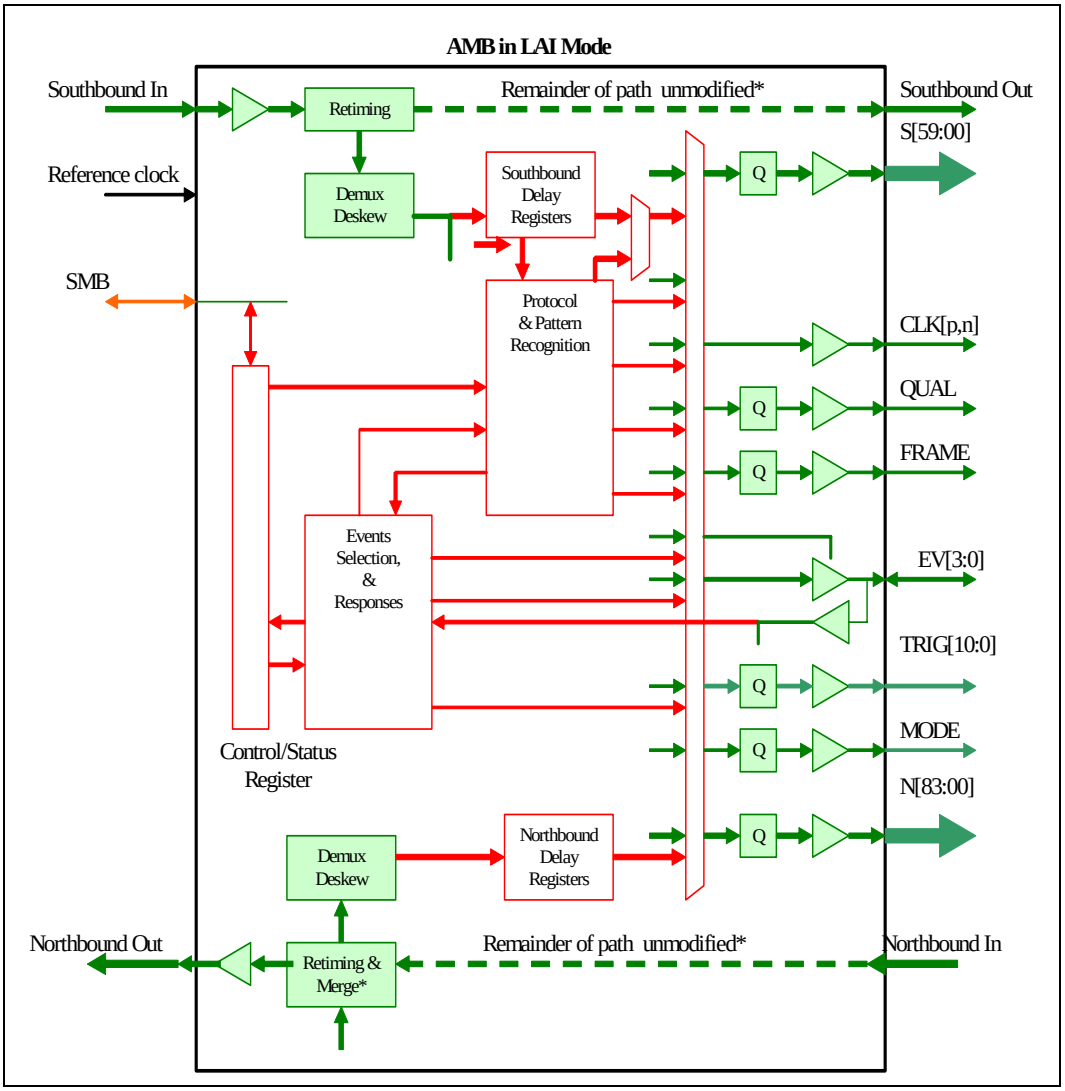
Figure 5-9 is a block diagram of the LAI implementation.

The LAI block defines a frame from the host (not the AMB) point-of-view, so the slota command is delayed by one core cycle relative to the slotb and slotc commands. The southbound delay pipeline consists of one set of core registers for slota, and one set of core registers for delayed slota, and slotb and slotc.

The protocol unwrapping and pattern recognition block takes the registered southbound frame and detects and logs any local events. Events are selected in this same cycle, registered on the core clock, and then forwarded to the DDR cluster. The southbound frame is also registered once more and forwarded to the DDR cluster.

The northbound registers the line this data to the same clock domain as the southbound data before it is forwarded to the DDR cluster.

Figure 5-9. Block Diagram of AMB in LAI Mode



5.2 Normal Mode Debug Features

5.2.1 Normal Mode Debug Triggers

Southbound command matching/masking functionality may be available in normal AMB operation. This can be used for triggering error injection or returning a signal in a northbound status frame for debug/monitoring purposes for example.

5.2.2 Error Injection

Refer to the JEDEC publication: *FB-DIMM Draft Specification: Design for Test, Design for Validation (DFx) Specification* for more information regarding Error Injection.

Selected errors of specific types may be injected internal to the AMB in response to selected in-band events or by mask/match events from commands arriving at the AMB . In the case of stuck lane errors, these are controlled directly via registers on the AMB The AMB will also forward errors injected by the host into the SB link.

5.2.2.1 Types of Injected Errors

There are several types of errors that the AMB can inject in order to enable validation and debug hardware and software mechanisms intended to deal with each error type in operating systems.

5.2.2.1.1 Errors Injected in Northbound Command Register Read and Read Data Frames

In response to a selected local event, the AMB will inject an error in frame data after or during calculating frame CRCs and transmitting the frame northbound. This is necessary to test/validate/debug HW and SW mechanisms designed to detect and deal with northbound channel soft (non-repeatable) channel transport errors.

Errors can be injected using the LFSR Idle pattern generator as a default.

5.2.2.1.2 Status Bits Injected in Status Block Sent Back in Response to Sync Command

In response to an FBD Sync command with R[1:0] = 2’b11, the AMB will return the user written FBDS3 for the next status block returned in response to a Sync command.

5.2.2.1.3 Invalid Parity Injected in Status Block Sent Back in Response to Sync Command

If the FBDS3.OVREN bit is set and FBDS3.USRPAR contains invalid parity for the data in FBDS3.USRVAL, the in response to an FBD Sync command with R[1:0] = 2’b11 and, the AMB will cause invalid parity to be passed in the next status a block returned in response to a Sync command.

5.2.2.1.4 Force Alert

In response to a selected local event, the AMB will force the beginning of alerts northbound. An error status bit indicating an injected alert error as the source will also be set. This error may be created by artificially corrupting CRC on the frame whose timing matches the error injection. This may have the same side effects as corrupting the CRC but will have a different error status. The beginning of alerts northbound will begin with the frame that would match read return slot for the corrupted frame.

For the Intel 6400/6402 Advanced Memory Buffer, the alert takes place 2 to 3 clocks after the NB response for the SB frame that caused the trigger

5.2.2.1.5 Selectively Force "Stuck On" Northbound Lanes

This capability is required to test ability of components and system to accomplish lane fail-over. Note that this is the only error injection feature which is not event driven, but rather shall be controlled directly by the STUCKL register in the AMB.

The selected lanes are "stuck" by being forced into "Electrical Idle".

5.2.2.1.6 Sourcing Northbound In-Band Event

The northbound event shall be asserted in the next transmitted FBDS0 status block following assertion of a selected local event. This approach allows any local event to be propagated past a tracing AMB LAI monitoring the FBD channel as well as for use as an event stimulus to the Host Interface logic.

§

6 Errors

6.1 Types of Errors and Responses

The Intel 6400/6402 Advanced Memory Buffer (AMB) detects link errors that could lead to data corruption. This includes CRC on commands and write data. The AMB also prevents damage to the machine state.

6.1.1 FBD Link Errors

6.1.1.1 Link Initialization Errors

Table 6-1 shows link errors that can occur during initialization.

Table 6-1. Link Errors in Initialization

Error	Response
Multi-lane failures - unable to achieve bit lock on at least 9 of 10 SB lanes	Never come out of training state
Multi-lane failures - unable to achieve bit lock on at least 12 NB lanes	Never come out of training state
NB_Data_Merge_Disable - able to achieve bit lock (pass thru link data) but internal errors prevent receiving valid link cmds or merging data	Host sets NB_Merge_Disable bit in TS2 to cause DIMM to be in a pass-thru mode both SB and NB - act as a blind repeater 1. No attempt to decode commands, no response to link register RD/WR, generate no alerts, neither generate or merge any NB traffic 2. SMBus access enabled
Single Lane failure - SB and/or NB	Support normal initialization protocol and Fail Over mode if commanded by host

6.1.1.2 Errors during Channel Operation

Table 6-2. Link Errors in Normal Operation (Sheet 1 of 2)

Error	Response
CRC Error on SB frame "A" Command - detected by 14-bit CRC (or reduced 10-bit CRC in Fail Over mode) or CRC Error on SB data or "BC" Commands in Command Frame - detected by 22-bit CRC (or reduced 10bit CRC in Fail Over mode	If CMDCRC error type enabled in EMASK Register 1. No command executed 2. 120-bit Raw SB Frame captured in RECFBD Error Log Registers 3. Type of error logged in FERR/NERR registers 4. Error/Alert Asserted bit set in FBD Status 0 register 5. Ignore future commands except Soft Channel Reset until Soft Channel Reset or Link Reset Received 6. Alert Frame sent continuously starting with NB frame in which returned data pattern would be sent if aborted command had been a config read. Alert patterns continue until Soft Channel Reset or Link Fast Reset received. ***Note:*** Will NOT close DRAM pages or place DRAM into Self Refresh until detection of Link Reset. Else ignore error
Lose transition density on channel as detected by no Sync in within 2 times the SYNCTRAININT value (typically last 84 frames). ***Note:*** Purpose of this error is not to detect violation of required transition density (6 out of 512) but to detect hang in the host and put DRAM into self refresh. Any corruption caused by lack of transitions will be detected by CRC violations.	If FEWEDGES error type enabled in EMASK Register 1. No command executed in expected Sync slot 2. 120 bit Raw SB Frame captured in RECFBD Error Log Registers 3. Type of error logged in FERR/NERR registers 4. Error/Alert Asserted bit set in FBD Status 0 register 5. DDR Self Refresh FSM triggered to put DRAMs into Self Refresh 6. Ignore future commands including Soft Channel Reset until Link Reset Received 7. Alert Frame sent continuously starting with NB frame in which returned data pattern would be sent if aborted command had been a config read. Alert patterns continue until Link Fast Reset received. Else ignore error
Write Buffer Overrun Write data received when FIFO is full	Overwriting the write buffer is a host error. The AMB does not take any action based on an overrun. DRAM writes will proceed. What data item is written following an overrun condition is not defined. An overrun could be the result of channel errors, but these errors are detectable by other means. Detection of an overrun condition is not required, but may be done by implementation dependent means for debug.
Write Buffer Underrun: not enough valid entries in Write FIFO to support write data to memory	Underrunning the write buffer is a host error. The AMB does not take any action based on an underrun. DRAM writes will proceed. What data item is written following an underrun condition is not defined. An underrun could be the result of channel errors, but these errors are detectable by other means. Detection of an underrun condition is not required, but may be done by implementation dependent means for debug.

Table 6-2. Link Errors in Normal Operation (Sheet 2 of 2)

Error	Response
Access to unimplemented register	If UNIMPLCFG error type enabled in EMASK Register 1. Drop Config Write cmds 2. Capture Addr in RECCFG register if not previously set 3. Return 0's data if Config Read (or return -1 if Read addr to unimplemented function - though currently expect all functions to be used) 4. UNIMPLCFG error type logged in FERR or NERR registers 5. Error/Alert Asserted bit set in FBD Status 0 register Else ignore error NOTE: The timing of the Error/Alert Asserted bit in FBD Status 0 response for an unimplemented register is not guaranteed relative to the corresponding Sync/Status boundaries. For example, if an unimplemented register access occurs in the frame before a Sync frame, the Error Asserted bit in FBD Status 0 may not be asserted in northbound Status corresponding to that Sync. But it will be asserted in a later Status response.
Undefined command	Undefined commands with good CRC are ignored. This is not considered an error condition, and is not logged. Treat as reserved command or Channel NOP
TID error on config writes	If the TID bit on a config write matches the value of the previous TID bit, the write is ignored. The TID bit stored in the AMB is left unchanged in this case. This error does not cause an alert frame, and is not logged. The purpose of the TID bit is to allow the host to retry a config write command following a fast reset if it does not know if it had been executed prior to an alert. If the config write had occurred the TID bit will be the same, the retried write will be ignored. If it had not occurred, the TID bit will be opposite, and the retried write to will be executed.

6.1.2 DDR Errors

Table 6-3. DDR Errors

Error	Response
Failure of software to achieve calibration	This is detected through firmware during the calibration routine. Firmware should treat the DIMM as a repeater if it is an intermediate DIMM or map it out if it is the last DIMM in the chain
DDR voltage does not power up	if normal FBD interface comes up, should at least act like a repeater. Does not bring down FBD channel - like above. DDR cmds directed at DIMM will fail to return valid responses.

6.1.3 Host Protocol Errors

AMBs are not expected to detect bad protocol from the host.

Table 6-4. Host Protocol Errors

Error	Response
Illegal combinations of commands • see Concurrent Command Delivery Rules section of the FBD Architecture and Protocol Specification	AMB response to illegal command combinations is undefined
Commands to multiple AMBs to return data in the same return frame.	If multiple AMBs attempt to return data in the same frame, the host will see the data from the northern most AMB which is providing data, as it will replace any data sent from AMBs to its south. A host controller should not produce commands which would cause multiple AMBs to respond with data in the same frame. A special case can occur where Alert frames are being sent by one or more AMBs while another AMB is returning data from a command. These cases are discussed in the Architecture and Protocol spec in the Northbound Alert Frame section.

6.1.4 Other Errors

Table 6-5. Other Errors

Error	Response
Overtemp - Temp > TEMPHI and overtemp enabled	If OVERTEMP error type enabled in EMASK Register 1. The OVERTEMP bit will be set in the FERR or NERR register as appropriate 2. Error/Alert Asserted bit set in FBD Status 0 register If TEMPHIENABLE set in TEMPSTAT register also 3. Shut down DDR channel: • Drive CKE low to the DRAMs and float the command, address, and data signals. CKE, ODT, and clock continue to be driven. The clocks to the DRAMS may be stopped after the CKE has been registered low 4. The FBD interface goes to electrical idle, with the receivers shut off to reduce power. 5. The core will continue to be clocked, and the AMB will respond to SMBus commands. This allows the host controller to determine the error condition ***Note:*** No recovery expected, just trying to prevent Si meltdown The AMB will remain in this state until the temperature is below TEMPHI and the OVERTEMP bit is reset via SMBus or a hardware reset. Else ignore error ***Note:*** A hardware reset will place the TEMPHIENABLE bit in its default state of disabled.
Injected alert	If INJCRC error type enabled in EMASK Register 1. No command executed 2. 120 bit Raw SB Frame captured in RECFBD Error Log Registers 3. Type of error logged in FERR/NERR registers 4. Error/Alert Asserted bit set in FBD Status 0 register 5. Ignore future commands except Soft Channel Reset until Soft Channel Reset or Link Reset Received 6. Alert Frame sent continuously starting with NB frame in which returned data pattern would be sent if aborted command had been a config read. Alert patterns continue until Soft Channel Reset or Link Fast Reset received. Else ignore error NOTE: this basically the same as a CRC error except that CRC is not actually corrupted
Injected error	If INJERR error type enabled in EMASK Register 1. Type of error logged in FERR/NERR registers 2. Error/Alert Asserted bit set in FBD Status 0 register Else ignore error
No REFCLK	Reset should not be released if no REFCLK present.PLL will not achieve lock, the AMB will not come out of reset.

6.2 Error Logging

6.2.1 Error Logging Procedure

There are three basic types of errors in FBD: OverTemp, Alerts and Status Only Errors.The first occurrence of any type of unmasked error are flagged in the FERR register. Multiple bits can be set in this register if multiple errors occur in the same clock period. Subsequent errors are flagged in the NERR register.

Unmasked “Alert” errors generate in-band link alert messages. All unmasked errors also set the error bit in FBDS0 that is returned in regularly scheduled in-band status response messages that occur following Sync commands.

There are error data logs associated with some of the errors. Once the first “Alert” error has been flagged in the FERR or NERR (and matching SB frame data logged), the log registers for that error remain locked until either 1) all “Alert” error bits in the FERR and/or NERR are cleared, or 2) a power-up reset. Once the first Unimplemented

Configuration Register Access error has been flagged (and matching address logged), the log registers for that error remain locked until either 1) that bit in the FERR and/or NERR cleared or 2) a power-up reset.

6.3 Fail Over Mode Support

The AMB supports single lane Fail Over mode as described in the *FB DIMM Architecture and Protocol Specification*,. This is done under host control or through the SMBus.

6.4 Failback to Pass-Thru

In general, the AMB attempts to minimize the number of Single Points Of Failure (SPOF) that could bring down the entire channel. Errors in any one lane can be mapped out with Fail Over. Errors on the DDR interface can be handled by disabling the DRAM interface and leaving the AMB in a repeater like mode. The goal is to allow the system (following a reset or fast reset sequence) to work around the bad DIMM and keep the DIMMs downstream in operation until there is time for system maintenance.

A AMB in an intermediate DIMM should continue to operate in pass-thru mode so that NB and SB data are relayed to the next links in the channel with minimal functionality. Only the following parts of the AMB need to be healthy to support this mode:

- Clock inputs and PLL circuitry to generate FBD clocks
- Minimal core logic around FBD I/O enabling and reset
  - — Reset generation
  - — Bit lock detection.
- at least N-1 FBD lanes operational in NB and SB channels

§

7 SMBus Interface

The Intel 6400/6402 Advanced Memory Buffer (AMB) has configuration registers that provide flexibility and allow for testing and optimization of the chip. Upon system reset (RESET#), configuration registers are reset to predetermined default states, representing the minimum feature set required to successfully bring up a nominal channel. It is expected that the BIOS will properly determine and program the optimal configuration settings.

For all of these registers, the AMB supports register access mechanisms through SMBus as well as through in-band channel commands.

7.1 System Management Access

System Management software in the platform can initiate system management access to the configuration registers. This can be done through SMBus accesses.

The mechanism for the Server Management (SM) software to access configuration registers is through a *SMBus Specification*, Rev. 2.0-compliant slave port. The AMB contain this slave port and allow access to the configuration registers.

SMBus operations are made up of two major steps: (1) writing information to registers within each component and (2) reading configuration registers from each component. The following sections will describe the protocol for an SMBus master to access a AMB's internal configuration registers. Refer to the *SMBus Specification*, Rev. 2.0 for the bus protocol, timings, and waveforms.

7.1.1 SMBus 2.0 Specification Compatibility

The principal requirement from the SMBus 2.0 specification is support of the "high power" bus electrical specifications described in the layer 1 (Physical layer) chapter.

For the simple register access requirements of FBD, no layer 2 (Link layer) or layer 3 (Network layer) extensions provided by the 2.0 specification are used. In particular, there is no support for Address Resolution Protocol (ARP) since FBD is using fixed addresses. Additionally, only a subset of the network packet protocols described in the specification are needed and these are described below.

AMB's are required to support read and write transactions without requiring clock stretching in order to simplify host controller requirements. For similar reasons, AMB's should not master SMBus transactions in normal operation.

7.1.2 Supported SMBus Commands

The AMB SMBus Rev. 2.0 slave ports support register reads and writes built out of the following SMBus primitive commands:

The slave address for each primitive SMBus transaction are determined from the SA pins.

- For normal FBD DIMMs:
  - Slave Address[6:3] = 4'b1011
  - Slave Address[2:0] = SA[2:0]

- For repeaters or LAI AMBs:
  - Slave Address[6:3] = 4'b0011
  - Slave Address[2:0] = SA[2:0]

Each SMBus transaction has an 8-bit command driven by the master. The format for this command is illustrated in Table 7-1 below.

Table 7-1. SMBus Command Encoding

7	6	5	4	3:2	1:0
Begin	End	Rsvd	PEC_en	Internal Command: 00 - Read DWord 01 - Write Byte 10 - Write Word 11 - Write DWord	SMBus Command: 00 - Byte 01 - Word 10 - Block 11 - *Rsvd*

The *Begin* bit indicates the first transaction of a read or write sequence.

The *End* bit indicates the last transaction of a read or write sequence.

The *PEC_en* bit enables the 8-bit PEC generation and checking logic.

The *Internal Command* field specifies the internal command to be issued by the SMBus slave logic. Note that the Internal Command must remain consistent (that is, not change) during a sequence that accesses a configuration register. Operation cannot be guaranteed if it is not consistent when the command setup sequence is done.

The *SMBus Command* field specifies the SMBus command to be issued on the bus. This field is used as an indication of the length of transfer so the slave knows when to expect the PEC packet (if enabled).

Reserved bits should be written to zero to preserve future compatibility.

7.1.3 FBD AMB Register Access Protocols

Sequences of these basic commands will initiate internal accesses to the component's configuration registers.

Each configuration read or write first consists of an SMBus write sequence which initializes the register's address. The term sequence is used since these variables may be written with a single block write or multiple word or byte writes. Once these parameters are initialized, the SMBus master can initiate a read sequence (which performs a configuration read) or a write sequence (which performs a configuration write).

Table 7-2. SMBus Protocol Addressing Fields

Address Field Name	Bits	Description
Reserved	7:0	Reserved - AMB may alias all these addresses to 00h
Dev	4:0	Reserved - AMB may alias all these addresses to 00h
Function	2:0	Function Address
Reg_Num[15:8]	7:0	Reserved - AMB may alias all these addresses to 00h
Reg_Num[7:0]	7:0	Register Address within Function

7.1.3.1 Configuration Register Read Protocol

Configuration reads are accomplished through an SMBus write(s) and later followed by an SMBus read. The write sequence is used to initialize the Bus Number, Device, Function, and Register Number for the configuration access. The writing of this information can be accomplished through any combination of the supported SMBus write commands (Block, Word, or Byte). The *Internal Command* field for each write should specify Read DWord.

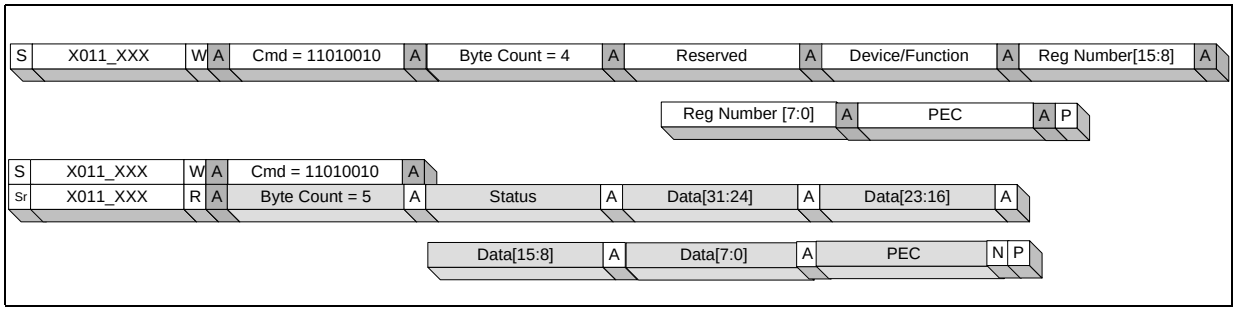
After all the information is set up, the last write (*End* bit is set) initiates an internal configuration read. If an error occurs during the internal access, the last write command will receive a NACK. A status field indicates abnormal termination and contains status information such as target abort, master abort, and time-outs. The status field encoding is defined in the following table.

Table 7-3. Status Field Encoding for SMBus Reads

Bit	Description
7	Reserved
6	Reserved
5	Reserved
4	Internal Target Abort
3:1	Reserved
0	Successful

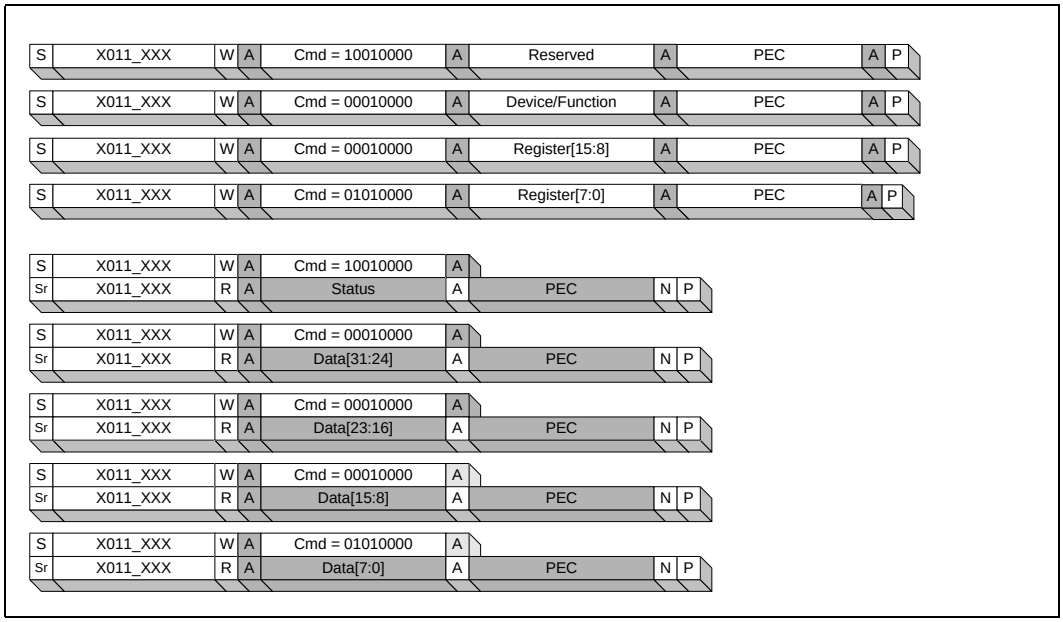
Examples of configuration reads are illustrated below. All of these examples have PEC (Packet Error Code) enabled. If the master does not support PEC, then bit 4 of the command would be cleared and there would not be a PEC phase. For the definition of the diagram conventions below, refer to the *SMBus Specification*, Rev. 2.0. For SMBus read transactions, the last byte of data (or the PEC byte if enabled) is NACKed by the master to indicate the end of the transaction. For diagram compactness, "Register Number[]" is also sometimes referred to as "Reg Number" or "Reg Num".

Figure 7-1. SMBus Configuration Read (Block Write / Block Read, PEC Enabled)



The following example uses byte reads.

Figure 7-2. SMBus Configuration Read (Write Bytes / Read Bytes, PEC Enabled)



7.1.3.2 Configuration Register Write Protocol

Configuration writes are accomplished through a series of SMBus writes. As with configuration reads, a write sequence is first used to initialize the Bus Number, Device, Function, and Register Number for the configuration access. The writing of this information can be accomplished through any combination of the supported SMBus write commands (Block, Word or Byte).

On SMBus, there is no concept of byte enables. Therefore, the Register Number written to the slave is assumed to be aligned to the length of the Internal Command. In other words, for a Write Byte internal command, the Register Number specifies the byte address. For a Write DWord internal command, the two least-significant bits of the Register Number are ignored. This is different from PCI where the byte enables are used to indicate the byte of interest.

After all the information is set up, the SMBus master initiates one or more writes which sets up the data to be written. The final write (*End* bit is set) initiates an internal configuration write. If an error occurred, the SMBus interface NACKs the last write operation just before the stop bit.

Examples of configuration writes are illustrated below. For the definition of the diagram conventions below, refer to the *SMBus Specification*, Rev. 2.0.

Figure 7-3. SMBus Configuration Double Word Write (Block Write, PEC Enabled)

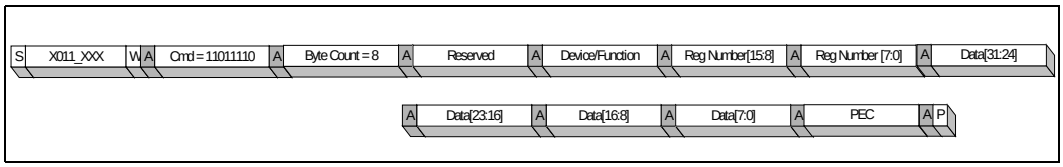


Figure 7-4. SMBus Configuration Double Word Write (Write Bytes, PEC Enabled)

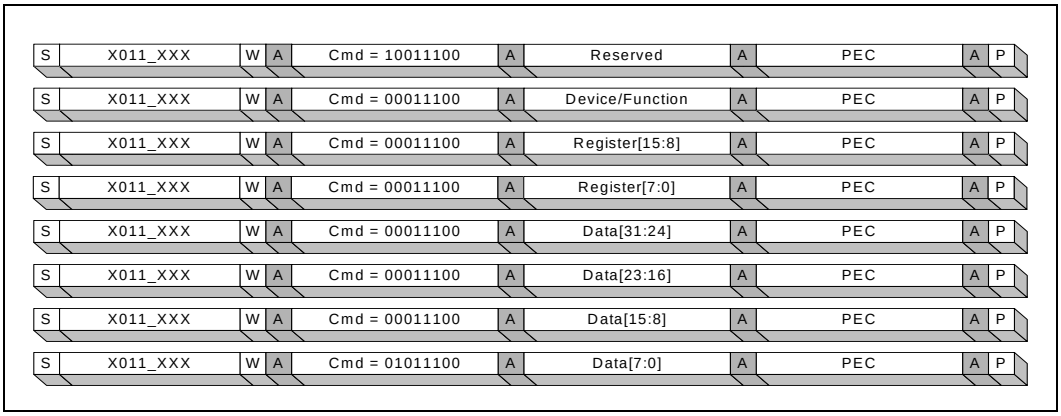


Figure 7-5. SMBus Configuration Word Write (Block Write, PEC Disabled)

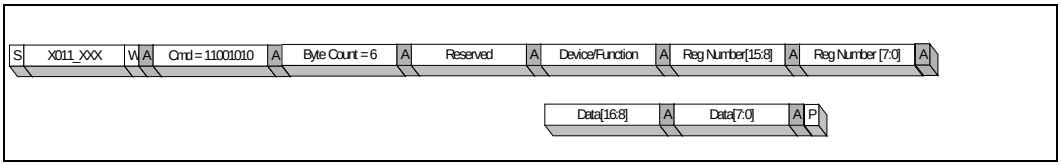
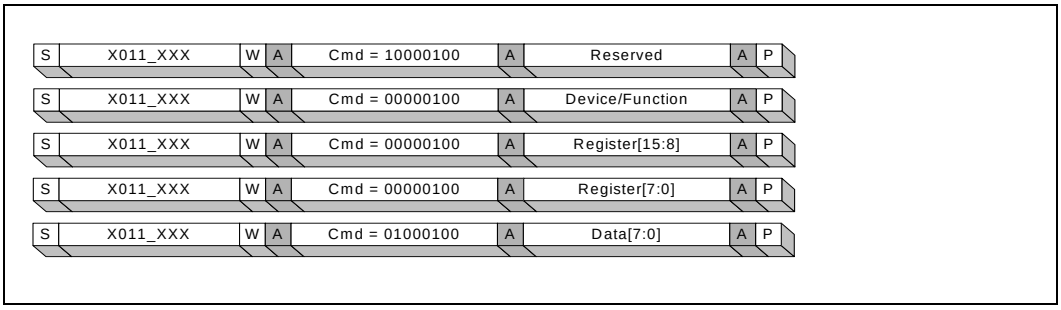


Figure 7-6. SMBus Configuration Byte Write (Write Bytes, PEC Disabled)



7.1.4 SMBus Error Handling

The SMBus slave interface handles two types of errors: internal and PEC. These errors manifest as a Not-Acknowledge (NACK) for the read command (*End* bit is set). If an internal error occurs during a configuration write, the final write command receives a NACK just before the stop bit. If the master receives a NACK, the entire configuration transaction should be reattempted.

If the master supports packet error checking (PEC) and the PEC_en bit in the command is set, then the PEC byte is checked in the slave interface. If the check indicates a failure, then the slave will NACK the PEC packet.

7.1.5 SMBus Resets

7.1.5.1 SMBus Transactions During FBD Link Fast Reset

When the FBD link transitions into Electrical Idle (disable state) from an active state, this causes a "fast" reset of all non-sticky registers in the AMB. SMB transactions underway during a "fast" reset will not complete normally.

This is not a problem since SMBus accesses are only required prior to initial link turn-on or for diagnostic access when a link can not be initialized. It is the host's responsibility to monitor for SMBus transactions during a fast reset and retry these transactions when the link is stable.

An interrupted transaction will result in the AMB as slave not properly acknowledging the Master. This protects write transactions. However, if a read transaction has proceeded to the point where the slave no longer acknowledges the master, read data can be lost when the SMBus state machine is reset. If PEC is enabled, this data loss will be detected as a PEC error.

The host restricting usage of SMBus to when the link is idle or monitoring "fast resets" and retrying transactions that are interrupted is the safest SMBus access method.

7.1.5.2 SMBus Interface State Machine Reset

The slave interface state machine can be reset by the master in two ways:

- The master holds SCL low for 25 ms cumulative. Cumulative in this case means that all the "low time" for SCL is counted between the Start and Stop bit. If this totals 25 ms before reaching the Stop bit, the interface is reset.
  - Timing is set up to be:
    * 30 ms at DDR2-667
    * 37.5 ms at DDR2-533
- The master holds SCL continuously high for 50 µs.
  - Timing is set up to be:
    * 60 µs at DDR2-667
    * 75 µs at DDR2-533

7.1.5.3 SMBus Transactions During Hard Reset

Since the configuration registers are affected by the reset pin, SMBus masters will NOT be able to access the internal registers while the system is reset.

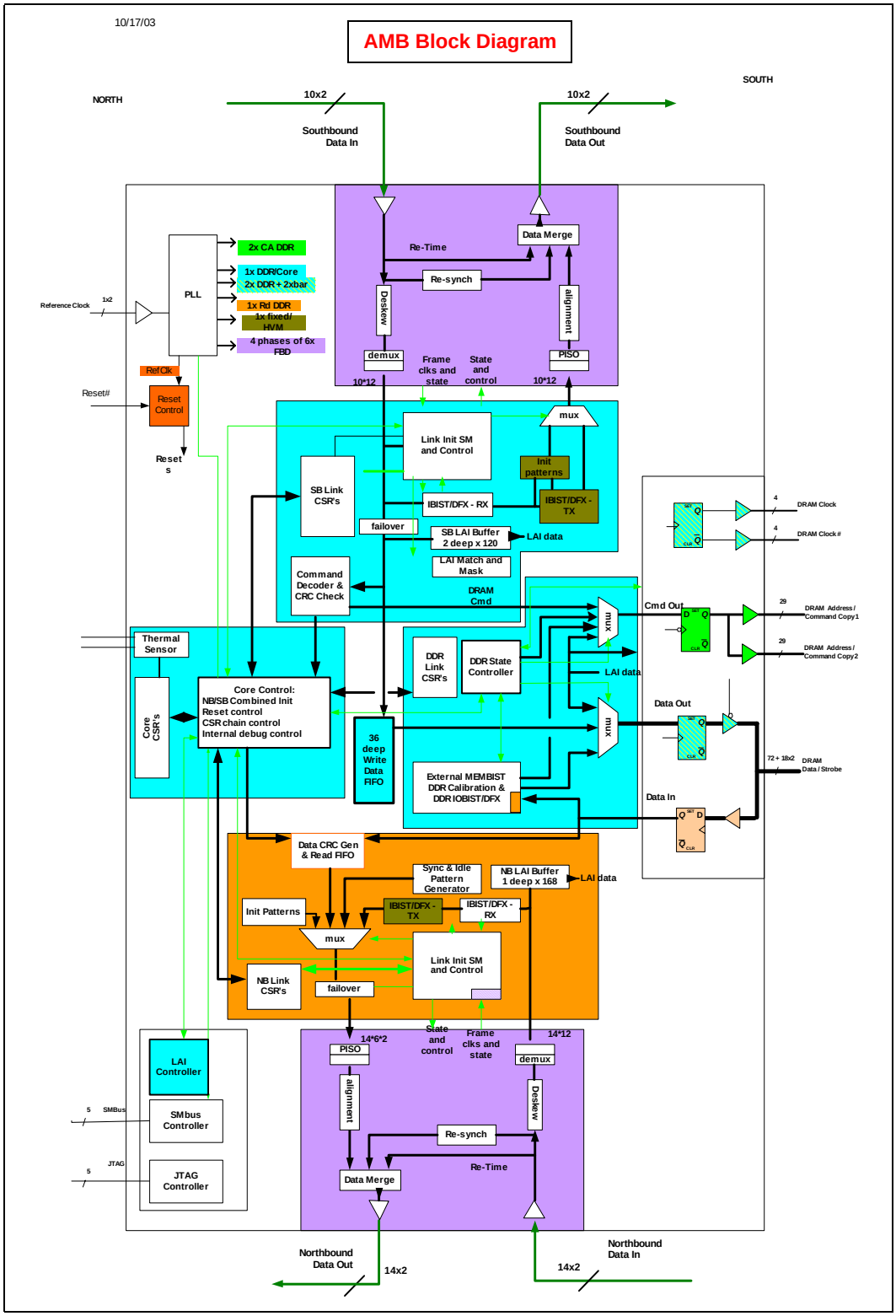
§

8 Clocking

8.1 Intel 6400/6402 Advanced Memory Buffer (AMB) Clock Domains

There are three main clock domains in the Intel 6400/6402 Advanced Memory Buffer (AMB). The FBD (Fully-Buffered DIMM) link domain is a 12x multiple of the core clock. The DDR data-rate domain is 2x the core clock. The core domain frequency equals the DDR command-rate, and is a 2x multiple of the external reference clock. The ratio between these domains remains fixed.The AMB logic in each domain is shown in Figure 8-1.

Figure 8-1. AMB Clock Domains



8.2 PLL Clocks

The PLL receives a reference clock at 1/2X, where X is the DDR command frequency. The PLL generates the internal clocks shown below in Table 8-1.

Table 8-1. PLL Clocks

Clock	Notes
DDRCA1X	This clock is unused in the AMB. It is still used for setting the alignment of the DDRCA2X clock within the PLL.
DDRCA2X	Rising edge aligned to DDRCA1X. Also allows command/address bus to run at DDR data rate in LAI mode.
DDR1X	Along with DDR2X and DDR2X#, used by DDRI/O cluster to generate DDR command clock, DDR DQS output, and DDR DQ output. This is also used as core clock.
DDR2X	Rising edge aligned to DDR1X
DDR2X#	Inverted DDR2X
DDRRD1X	Used by DDRI/O cluster to advance the DDR Read FIFO read-pointer. It is also by Northbound logic. This is driven by an independent divider and can be moved w.r.t to the core clock in 2UI increments to align it with the DDR data availability.
HVMCLK	This is a fixed clock at the same frequency as the core clock but is driven by a independent divider.
FBDCK (4)	Four phases of 6X clock. Provided to FBD Northbound, Southbound high-speed I/O clusters. These signals are connected by abutment.

There are additional PLL modes used for testing and bring up. Details are in Section 8.9, “Additional Clock Modes” on page 87.

8.3 Reference Clock

A low-jitter differential reference clock (REFCLK) is routed to the host and each DIMM from a common clock source on the system board. This reference clock uses HCSL (High-Speed Current Steering Logic) signaling and its detailed requirements are documented in the FB-DIMM Draft Specification: High Speed Differential P2P Link at 1.5V. The AMB uses the reference clock to generate internal buffer clocks and to generate the clocks to the DRAMs located on each DIMM. The frequency of the reference clocks (133 to 200 Mhz) is one half the frequency of the DRAM base clock (267 to 400 Mhz), that is, it is one half the command-rate of the DRAM devices located behind the AMB. For example, for DDR2 667 DRAM devices the reference clock frequency would be 167 MHz. The reference clock is the basis for the various Core, FBD and DDR internal clocks.

It is a requirement for the FBD channel to operate in the presence of Spread Spectrum Clocking (SSC), which is commonly used to reduce EMI. The reference clocks for FBD have to meet a jitter specification.

The reference clocks to the host and each DIMM are mesochronous, that is, they have an unknown but fixed phase relationship to each other or the memory channel. This simplifies PCB routing since no precise length matching is required. However an upper bound for the clock length mismatch is necessary since the maximum phase difference between the data sent out with the transmitter clock and the receiver clock needs to be limited in the presence of SSC. It is required that all the reference clocks for a given FBD channel originate from a single clock source, for example, a common clock synthesizer or clock oscillator, and travel through the same jitter spectrum modifying components (for example, PLL clock buffer) thereby ensuring that there is no frequency mismatch or frequency drift between FBD agents.

The PLL will also operate with the REFCLK at 100 MHz during special transparent mode testing. Since there is no high speed link operation, there can be looser requirements for jitter and no SSC.

8.4 FBD Lane Frame Clocks

Each FBD I/O lane also sources a frame clock at core frequency that is matched to the parallel data sourced by the lane. These are used during initialization to capture data in training sequences and to align data across the link.

8.5 Clock Ratios

The core, DDR and FBD link clock domains are fixed in a 1:2:12 ratio. The SMBus asynchronous subsystem need not scale. The supported clock ratios are shown in Table 8-2.

Table 8-2. AMB Clock Ratios

FBD Link Data Rate	DDR Data Rate	Core Frequency	Ref Clk	FBD Link : Core	Core : DDR
3.2 Gb/s	533 Mb/s	266 MHz	133 MHz	12 : 1	1 : 2
4.0 Gb/s	667 Mb/s	333 MHz	167 MHz	12 : 1	1 : 2

8.6 DDR DRAM Clock Support

The DDR command clocks (CLK[3:0], $\overline{\text{CLK}}$[3:0])are generated by the AMB. They operate at 1X the core frequency for DDR2.

The write strobes operate at the same frequency as the CLK/$\overline{\text{CLK}}$ signals. Write data and check bits are aligned to both the rising and falling edges of the write strobe.

The source-synchronous read strobes operate at the same rates as the write strobes. Each read strobe will be individually aligned with its portion of the data and check-bits.

8.7 SMBus

The SMBus clock is synchronized to the core clock. Data is driven into the AMB with respect to the serial clock signal. Data received on the data signal with respect to the clock signal will be synchronized to the core using a metastability hardened synchronizer guaranteeing an MTBF greater than 10^7 years. When inactive, the serial clock should be deasserted (High). The serial clock frequency is 100 kHz.

8.8 Clock Pins

Table 8-3. Clock Pins (Sheet 1 of 2)

Pin Name	Pin Description
SCK	AMB clock
$\overline{\text{SCK}}$	AMB clock (Complement)
VCCAPLL	analog power supply for PLL
VSSAPLL	analog ground for PLL
TCK	TAP clock

Table 8-3. Clock Pins (Sheet 2 of 2)

Pin Name	Pin Description
SCL	SMBus clock
CKE[1:0]{A,B}	DDR clock enables
CLK[3:0]	DDR clocks
$\overline{\text{CLK}}$[3:0]	DDR clocks (Complements)
DQS[17:0]	DDR data/check-bit strobes
$\overline{\text{DQS}}$[17:0]	DDR data/check-bit strobes (Complements)

8.9 Additional Clock Modes

8.9.1 Transparent Mode Clocking

In transparent mode, all input signals are registered in the core clock domain and all outputs are driven from the output of registers clocked by core clock. In order to achieve determinism on a tester in this mode, the feedback clock for the PLL is taken from the end of the core clock tree. This makes all timing relative to the input reference clock.

8.10 PLL Requirements

8.10.1 Jitter

The FBD link clocks are produced by a PLL that multiplies the SCLK frequency. See the *High Speed Differential Point-to-Point Link at 1.5 V for Fully Buffered DIMM Specification* and the *Circuit Architecture Specifications* for the DDR, FBD and PLL custom I/Osfor more details.

8.10.2 PLL Bandwidth Requirements

The PLL -3dB loop bandwidth shall be between fREFCLK/18 and fREFCLK/6 with a 3dB maximum peaking.

See the *High Speed Differential Point-to-Point Link at 1.5 V for Fully Buffered DIMM Specification* and the *Circuit Architecture Specifications* for the DDR, FBD and PLL custom I/Osfor more details.

8.10.3 External Reference

The PLL uses an external reference clock - described previously.

See the *High Speed Differential Point-to-Point Link at 1.5 V for Fully Buffered DIMM Specification*and the *Circuit Architecture Specifications* for the DDR, FBD and PLL custom I/Os for more details.

8.10.4 Spread Spectrum Support

The AMB PLL will support Spread Spectrum Clocking (SSC). SSC is a frequency modulation technique for EMI reduction. Instead of maintaining a constant frequency, SSC modulates the clock frequency/period along a modulation profile.The AMB is designed to support a nominal modulation frequency of 30-33 kHz with a downspread of 0.5%.

See the *High Speed Differential Point-to-Point Link at 1.5 V for Fully Buffered DIMM Specification* and the *Circuit Architecture Specifications* for the DDR, FBD and PLL custom I/Os for more details.

8.10.5 Frequency of Operation

The PLL's support a range of operation that exceeds the AMB's functional range. This allows the AMB to be tested at a higher frequency than the maximum specification to provide test guardband. Lower frequencies are supported to allow system debug. The PLL will also operate with the REFCLK at 100 MHz during transparent mode testing.

8.10.6 RESET#

The externally generated RESET# signal indicates when the core voltage is up and reference clocks are stable. The core will use an asserted RESET# to asynchronously put the AMB in reset, and to hold the AMB in reset. For details see the reset chapter.

External clocks dependent on PLL's are DDR clocks and strobes, and SMBus clock.

8.10.7 Other PLL Characteristics

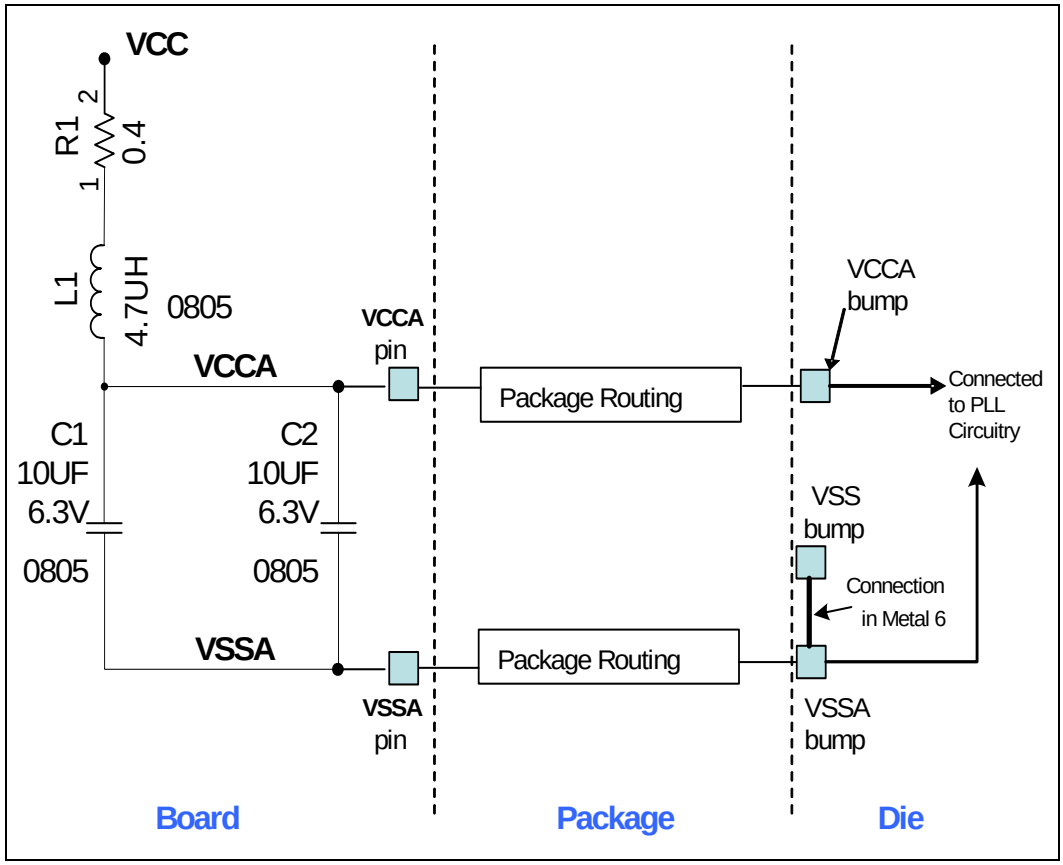
The PLL VCOs oscillate continually from power-up. At all other times, PLL output dividers track the VCO, providing pulses to the clock trees. Logic that does not receive an asynchronous reset can thus be reset "synchronously".

A "locked" PLL will only serve to prove that the feedback loop is continuous. It will not prove that the entire clock tree is continuous. The PLL is disabled for leakage test.

8.11 Analog Power Supply Pins

The incorporates one PLL. This PLL requires an Analog V_{CC} and Analog Vss pad. Therefore, there will be external LC filters for the AMB .

Figure 8-2. FBD PLL Power Supply Filter



Separate filtered power pins are available for use by a PLL if needed.

Warning: The filters are NOT to be connected to board Vss. The ground connection of the filters will be routed through the package and grounded to on-die Vss.

§

9 Reset

This chapter describes aspects of hardware reset specific to the Intel 6400/6402 Advanced Memory Buffer (AMB).

9.1 Platform Reset Functionality

The FBD channel provides a RESET# signal to initialize all AMBs on the channel. The generation of this signal is platform dependent, and may be asynchronous to the clock. The platform will assert RESET# at power up. This signal may be asserted at other times, such as a warm boot.

It is possible that platform conditions cause RESET# to be asserted at any time, including in the middle of DRAM commands. This could occur during a warm boot. Under these conditions, the AMB will be reset and the contents of memory are not guaranteed. The state of the DRAMs must be guaranteed when reinitialized for proper response.

9.1.1 Platform RESET# Requirements

- RESET# must be asserted at power up, and may also be asserted at other times such as a warm boot. Asserting RESET# at warm boot will clear all error logging registers.Asserting RESET# only at power up will allow error logging registers to be maintained through a warm boot cycle.
- Asserting RESET# at warm boot will clear all error logging registers.

There is no need to delay or lockout RESET# going to the FBD channel since the AMB will guarantee that the tDelay parameter is met. Reference Clocks must remain stable for at least 4 clock cycles after RESET# is asserted in order to allow the AMB to satisfy the tDelay requirement.

- RESET# must be asserted during power up, and for a minimum of 1 mS after the FBD channel power and reference clocks SCK/$\overline{\text{SCK}}$ are stable.
- RESET# must be asserted for a minimum of 100 uS. This will only apply if RESET# is re-asserted while power and clocks remain stable.
- After initial power on, if the reference clock frequency is changed while Reset is asserted, reset must not be deasserted until power and reference clocks SCK/$\overline{\text{SCK}}$ have been stable for at least 1 ms.

9.1.2 RESET# Requirements

RESET# is asynchronously applied to all storage elements. Assertion of RESET# does not effect AMB PLL operation. Internal clocks continue to run.

Upon assertion of RESET#:

- DRAM CKE is driven low asynchronously with minimal delay (within 1 clock, asynch path from reset to CKE).
- DRAM CLK/$\overline{\text{CLK}}$ continue to run with no short pulses generated within the tDelay period specified in JEDEC ballot 1410.01.
- DRAM CLK/$\overline{\text{CLK}}$ may be stopped after the tDelay has been satisfied.

- All internal register bits are set to their default values, including any error logging bits that are normally not reset by the channel reset.
- The initialization FSM is put into the disable state. All internal state machines are put in their default state.
- All southbound and northbound Tx outputs are put into electrical idle (EI) mode. All Tx outputs stay in EI mode until the appropriate initialization state after deassertion of RESET#.

9.1.3 Power-Up and Suspend-to-RAM Considerations

In a suspend to RAM environment the DRAMs are put into self-refresh mode, and the FBD channel power may be removed. The DRAM power supply remains active. This supply is used by the AMB DRAM interface I/O circuits. The AMB must keep the CKE pins low, without glitches through this transition.

RESET# should be asserted before channel power goes away when entering S3.

DDR control and clock signals will be pulled low during initial power up. This may be done with a voltage detection circuit. CKE must be maintained low during this time without glitches to prevent the DRAMs from exiting self refresh mode. The RESET# signal will remain low during the power-up sequence, for at least 1mS after power and clocks are stable. The CKE signals must remain low until a command is received that takes the CKE signals high. This could be an exit self refresh command, or any of the DRAM CKE commands.

9.2 Reset Types

Types of reset:

- Hard resets occur when the RESET# signal is low. This usually occurs at power up.
- Fast resets occur when there is a reset event on the primary southbound FBD Link.
- SMBus resets affect only the SMBus interface.

9.3 Pads Controlling Reset

The AMB resets are controlled by the RESET# pad and the primary southbound FBD link pads. The RESET# pad resets the chip at power up. When the primary southbound FBD link pads indicate EI, a fast reset is started.

9.3.1 RESET# Pad

The low true RESET# pad is controlled by the platform which holds it low until after power and SCK/$\overline{\text{SCK}}$ are stable. RESET# asynchronously resets most of the chip to a safe initial state. The PLLs and TAP are not reset.When RESET# goes high, logic running on REFCLK waits an appropriate amount of time and then resets the core. Logic running on REFCLK is reset by RESET# directly. As the chip comes out of reset, the Primary South FBD Link is expected to be in a reset state. As the link sequences through the first initialization sequence after power up, the AMB will not generate any DRAM commands other than to maintain CKE low and enable DRAM clocks at the appropriate time.

9.3.2 Primary FBD Link

When an EI occurs on the primary southbound FBD link a fast reset is started. This starts a handshake procedure putting the DRAMs into self-refresh mode and resetting the AMB. Fast reset does not reset the PLL, sticky flops, and sticky configuration registers.

9.4 Details

Reset details and sequences will be released in a future revision of this document.

9.4.1 Cold Power-Up Reset Sequence

1. 1.5 V, 1.8 V and 3.3 V power supplies comes up

- RESET# asserted low while power supplies are coming up
- CKE's are low upon 1.8 V power up

2. BIOS queries SPD on all the FBDs on the channel to determine operating conditions
   - channel frequency, compatible DIMMs, DRAM and AMB parameters
3. Clocks up and stable at required frequency

- Reference Clocks (SCK/$\overline{\text{SCK}}$) should be stable at least 1ms before RESET# deasserted for designs with PLL running independent of RESET#
- DRAM clocks (CLK/$\overline{\text{CLK}}$) may be toggling at this time

4. RESET# deasserted high

- CKE's to DRAMs remain low

5. No transactions for at least 200 us after RESET# deasserted for designs with PLL's tied to RESET#

- No SMBus or in-band activity during this period
- DRAM clocks should be stable at this time

6. AMB parameters critical for robust link initialization are programmed via SMBus

- Architected link registers
  - LINKPARNXT: link frequency - Note: some AMBs may use this write to trigger PLL init
  - FBDSBCFGNXT: SB transmitter drive strength, de-emphasis setting and pass-thru mode
  - FBDNBCFGNXT:NB transmitter drive strength, de-emphasis setting and pass-thru mode
  - FBDBLTO:if NB lanes are to be deconfigured
- Personality Bytes from SPD needed for link initialization
  - PERSBYTE[5:0]NXT:
    In the AMB, these match:
  - SPDPAR01NXT: various FBD IO implementation specific controls
  - SPDPAR23NXT:various FBD IO implementation specific controls
  - SPDPAR45NXT
  - remaining Personality bytes are not required for link init and may be loaded over the high speed FBD configuration register accesses

- These “Next” register values must be transferred to the matching “Current” registers before the FBD link leaves the DISABLE state
  - Updates may be done right after the NXT register is updated when link is in electrical idle. Updates must be complete before the beginning of training.

7. FBD Link is initialized including CALIBRATION state.
8. Remaining AMB configuration is loaded over high speed FBD channel

- CMD2DATA, remaining personality bytes, other SPD parameters, DRAM parameters, Errors enabled, and so forth.

9. FBD Link goes through fast reset (no CALIBRATION) to establish the desired configuration
10. DRAM interface can now be established
    a. MRS/EMRS setup using DCALCSR and DCALADDR
    b. DRAM interface calibrated using DCALCSR
    c. Optionally, MemBIST functionality can be used to test the DRAMs
    d. DRAM’s can be initialized using MemBIST
11. Refresh must now be transferred to the host

- Option 1: Use fast reset on the link with DRAMs in self-refresh
  - clear DSREFTC:DISSREXIT to enable fast self refresh exit when link is re-established
  - put the link in disable state which automatically puts the DRAMs in self refresh.
  - Start the refresh engine on the host
  - bring up the link again
  - Host starts sending refresh commands as soon as L0 state reached
- Option 2: write control register to disable auto-refresh engine followed by
  - Clear DAREFTC:AREFEN to turn off auto-refresh
  - Host then immediately takes over sending refresh commands

12. Host now has complete control of the FBD Channel

9.4.2 S3 Restore Power-Up Reset Sequence

Follow steps 1) through 9) from cold power up sequence above. Step 2, BIOS query of SPD may be skipped if these values are saved elsewhere. Either way the personality bytes from the SPD are restored to their prior values in the AMB. Once this is done the DRAM interface can be restored.

1. DRAM interface can now be restored
   a. DRC, MTR, DSREFTC, and DAREFTC register restored
   b. Stored S3RESTORE[15:0] are written back into each AMB
   - do NOT recalibrate the DRAM interface using DCALCSR
   - do NOT reinitialize the DRAMs using MemBIST
2. Refresh must now be transferred to the host as in cold power up
3. Host now has complete control of the FBD Channel and prior DRAM memory state has been preserved.

9.4.2.1 Implementation Detail

1. The RESET# pad starts out low. This asynchronously asserts all internal resets.
2. After power comes up and REFCLK stabilizes, the PLLs start generating clocks. At this time, the phase relations between the clocks are not guarantied to be correct.
3. RESET# rises. A synchronized version releases the reset on the logic running on REFCLK. The chip monitors RESET# for 100 and 200 usec RESET# must remain stable during this time. If it falls the wait starts again until a stable high occurs.
4. After a stable RESET# high is detected, the PLL is reset to bring the generated clocks into correct alignment with themselves.
5. At this point the chip waits for the PLL to indicate reset complete. When it does, the internal resets are deasserted for all clock domains except TCK.

9.4.3 Reset Sequence for a Fast Reset

Figure 9-1 below shows a fast reset sequence. Important steps are described below:

- The chip is running with the RESET# high and a full link initialization sequence has been completed at least once.
- A electrical idle is detected on the primary Southbound FBD Link.
- If not the last DIMM, forward electrical idle Southbound to the next DIMM.
- Drive logic “0’s” on Northbound transmitters.
- The AMB completes the fast reset handshake (see below).
- Reset is asserted for all non-sticky registers.
- The values in the next fields are transferred to the Current fields.
- The PLL remains in lock but the clock outputs are reset, causing them to realign.
- IF not the last DIMM, the AMB waits for an electrical idle to appear on the secondary Northbound FBD link.
- Forward electrical idle Northbound
- The chip is released from reset.
- Freezing sticky configuration registers through reset

9.4.4 Fast Reset Handshake

When a reset event is detected on the primary Southbound FBD link, the AMB does the following:

- Immediately stops accepting new DRAM commands from the link.
- Halts all on-chip algorithms, including DRAM cal and MemBIST.
- Waits 200 ns for any in-process DRAM commands to complete.
- Asserts CKE high.
- Waits 200 ns for DRAM self-refresh exit to complete.
- Sends a DRAM precharge all to all ranks.
- Waits 30 ns for precharge all to complete.
- Sends DRAM self-refresh entry commands to all ranks

9.4.5 Timing Diagrams

Figure 9-1. Cold Power-Up Reset

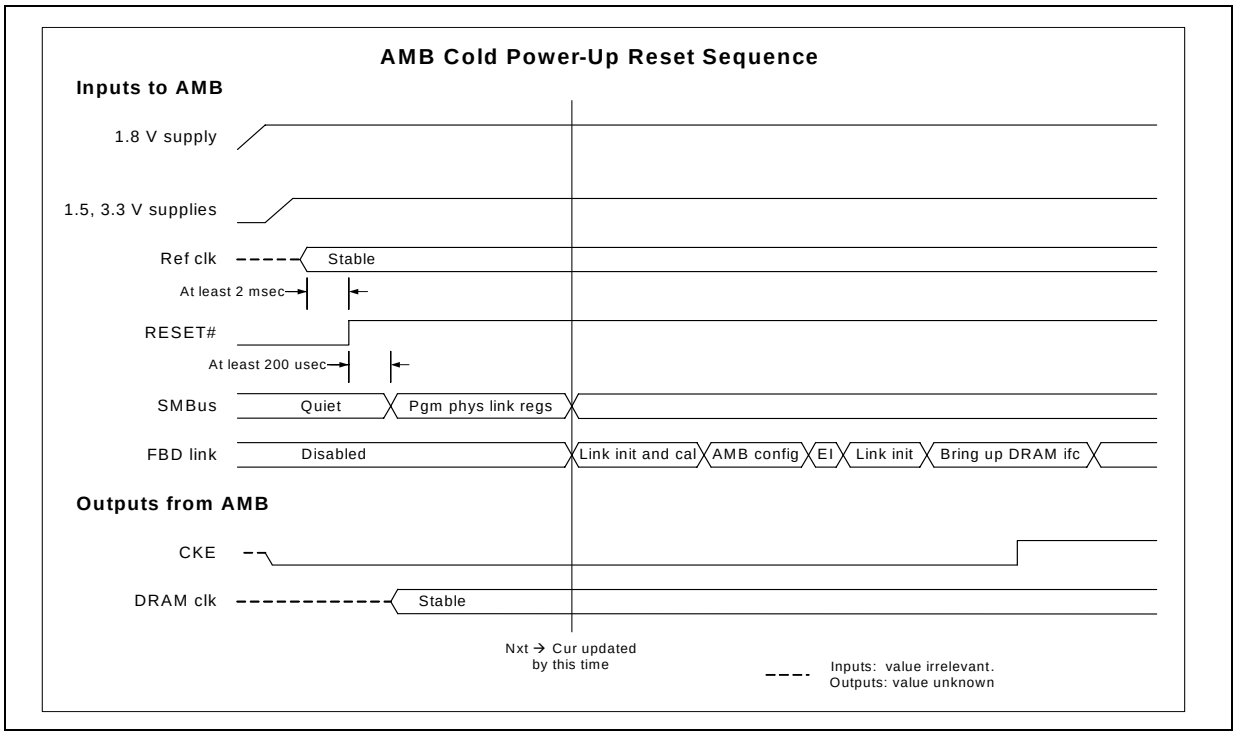
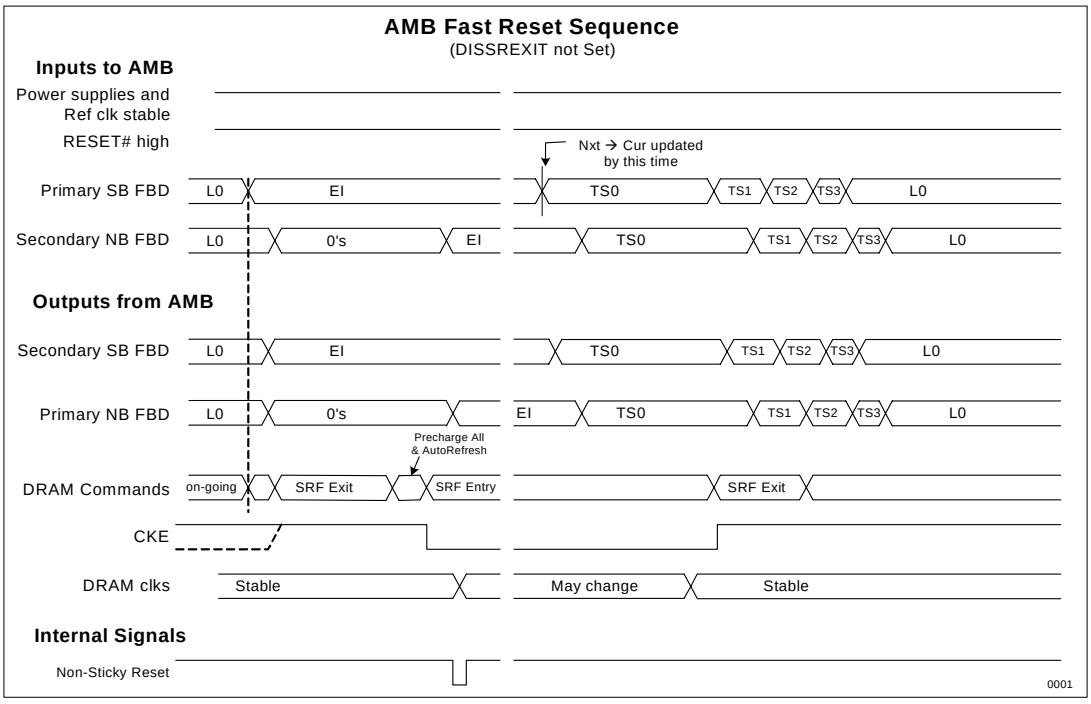


Figure 9-2. AMB Fast Reset Sequence



9.5 I/O Initialization

9.5.1 FBD Channel Initialization

Channel initialization states and uses of fast reset are described in the *FB-DIMM Architecture and Protocol Specification*.

9.5.2 DDR

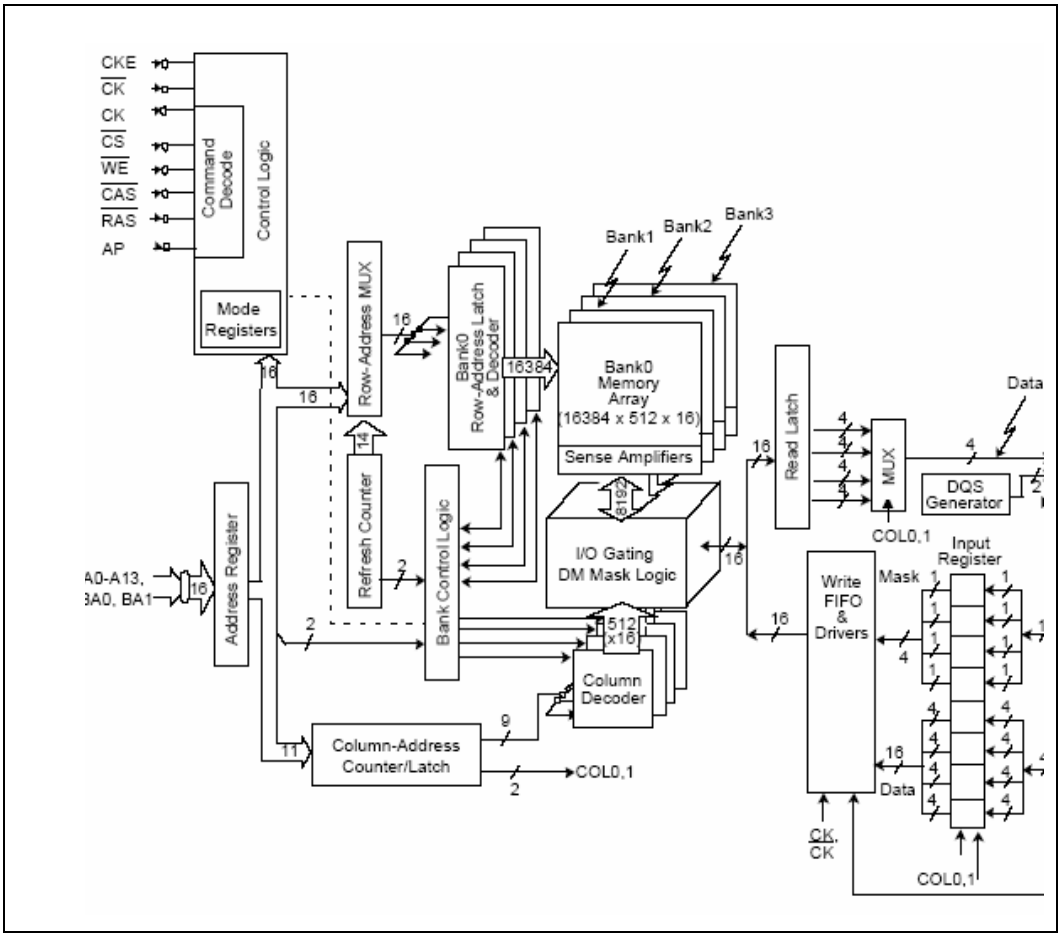
Analog compensation commences at power-up. It's completed by "RESET#" deassertion. After the FBD link reaches L0 state, setting the DRC.CKEN configuration bit enables CKE.The DDR interface is now ready for calibration.

§

10 Transparent Mode

Figure 10-1 shows the typical architecture of a DRAM with a 4 bit wide data path. The memory array operates at 100 to 200 MHz. With a pre-fetch of 4, there are 4 bits of data on each array access, allowing us to clock data in or out at 400 MHz. This 4 to 1 multiplexing and de-multiplexing is performed in the input register or output multiplexer.

Figure 10-1. DRAM Architecture



10.1 Transparent Mode

Refer to the JEDEC publication: *FB-DIMM Draft Specification: Design for Test, Design for Validation (DFx) Specification* for information regarding transparent mode.

Transparent mode is designed to allow access to the DRAM behind the AMB. In this mode high speed pins are converted into low speed pins and mapped to DRAM pins. The objective is to allow the use of existing test equipment and manufacturing processes. The tester must be capable of operation at 200 MHz. Transparent mode offers potential improvements in test capacity over traditional DIMMs. In this mode, FBD requires only 60 active pins to test the DIMM.

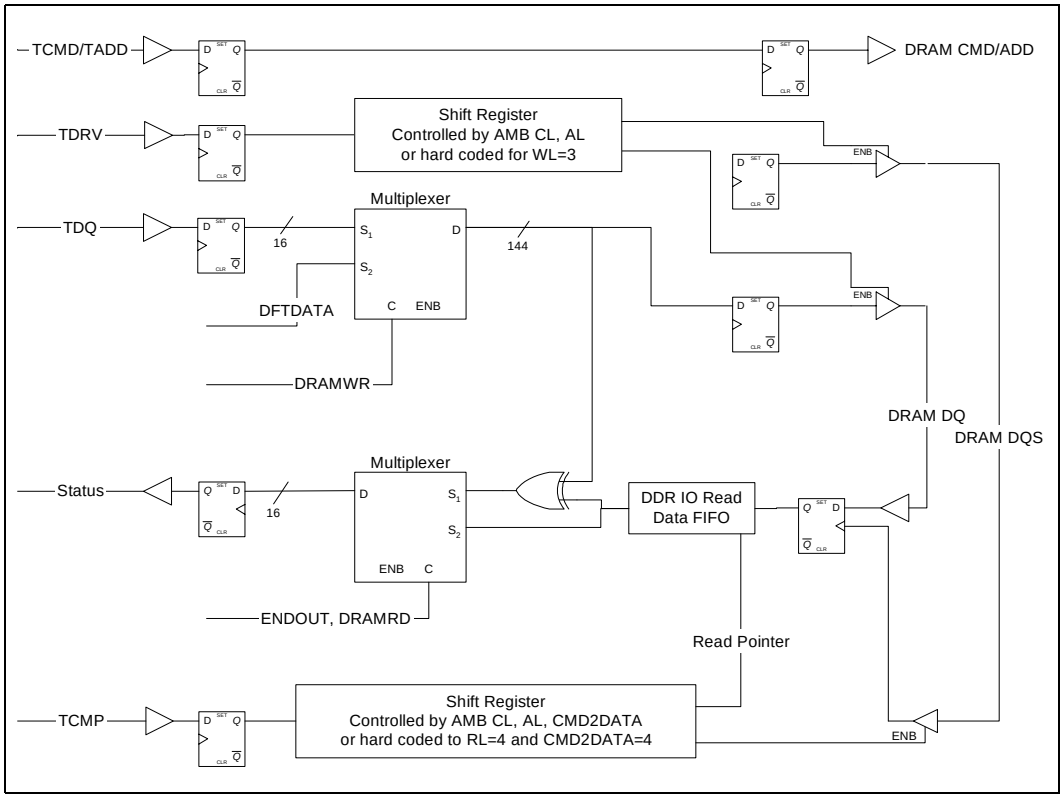
Data from the tester is 16 bits wide at 200 MHz (single data rate). The data rate is doubled and the width halved on the way to the DRAM (by clocking out 8 bits of data on the rising edge of the clock and the remaining 8 bits on the falling edge).

The tester will drive data to be written to the DRAM on a write pass and data to be compared on a read. DRAM data and the expected data from the tester is compared in the AMB. If the actual and expected data differ the pass/fail outputs will indicate which DRAM failed.

10.1.1 Block Diagram

The data paths for transparent mode will bypass all link logic and normal DRAM control logic. The DDR interfaces are used intact. Figure 10-2 is a block diagram of the Intel 6400/6402 Advanced Memory Buffer (AMB) in transparent mode.

Figure 10-2. Block Diagram for the AMB in transparent mode



10.1.2 Transparent Mode Signal Definitions

When the transparent mode is enabled. The FBD (Fully-Buffered DIMM) differential input pins designed in the AMB part become two single ended inputs. In transparent mode the FBD input pins require 0 to 500 mV swing (half of the normal differential input voltage). Input slew rates should be approximately 5 V/ns. This parameter is not critical, but it must be fast enough to be recognized by the AMB receiver. The DDR pins will operate with normal DDR2 timings and levels.

The AMB clock input pins will be used for transparent mode as well as normal mode. This also allows use of most of the existing on-chip clock distribution network.

Table 10-1. Additional Signals in Transparent Mode

Transparent Mode Signal Name	Pin Count	Frequency	Direction	Definition
TCKE[1:0]	2	200 MHz	In	Controls CKE[1:0]{A,B}
TCS#[1:0]	2	200 MHz	In	
TODT	1	200 MHz	In	Controls both ODT[1:0]{A,B}
TRAS#	1	200 MHz	In	
TCAS#	1	200 MHz	In	
TWE#	1	200 MHz	In	Controls $\overline{WE}${A,B}
TBA[2:0]	3	200 MHz	In	
TA[14:0]	15	200 MHz	In	
TDRV	1	200 MHz	In	Tester signal to drive data on writes
TCMP	1	200 MHz	In	Tester signal to compare on reads
TDQ[15:8]	8	200 MHz	In	Early data to each byte
TDQ[7:0]	8	200 MHz	In	Late data to each byte
TPF[8:0] or TDQO[15:0]	9 or 16	200 MHz	Out	Used for pass/fail (8:0) or direct access (15:0)
Sum of receivers Sum of drivers	44 16			

10.1.3 Transparent Mode to FBD Pin Mapping

The FBD pin mapping is shown in Table 10-2.

Table 10-2. Mapping of FBD Pins in Transparent Mode

FBD Pin Name	Count	Speed	Transparent Mode Signal Name	Comment
SN[0], $\overline{SN}$[0]	2	200 MHz	TCKE[1:0]	
SN[1], $\overline{SN}$[1]	2	200 MHz	TCS#[1:0]	
SN[2],	1	200 MHz	TODT	
$\overline{SN}$[2]	1	200 MHz	TRAS#	
SN[3],	1	200 MHz	TCAS#	
$\overline{SN}$[3]	1	200 MHz	TWE#	
SN[4], $\overline{SN}$[4] SN[5]	3	200 MHz	TBA[2:0]	
SN[13:6], $\overline{SN}$[13:6]	16	200 MHz	TA[15:0]	
PS[8]	1	200 MHz	TDRV	
$\overline{PS}$[8]	1	200 MHz	TCMP	
PS[7:0]	8	200 MHz	TDQ[15:8]	Early data
$\overline{PS}$[7:0]	8	200 MHz	TDQ[7:0]	Late data
SS[8:0]	9	200 MHz	TPF[8:0]	Pass/Fail mode (TRANSCFG.ENDOUT =0)
PN[13:7], SS[8:0]	16	200 MHz	TDQO[15:0]	Data output mode (TRANSCFG.ENDOUT =1)
Total Pins				

10.2 Transparent Mode Timing

10.2.1 Clock Frequency and Core Timing

The DDR2 DRAM clock frequency is 200 to 400 MHz but core timings require several clocks (or nS) to complete.

For example on DDR2 667:

- tCL, tRP and tRCD are 4 clocks or 12 ns
- tRC is 57 ns
- tRRD is 7.5 ns

DDR2 transactions are burst-oriented, reading or writing 4 or 8 words of data across 4 or 8 clock edges. Assuming a 4 bit burst, a x8 DRAM will transfer 32 bits on successive edges of 2 DRAM clock cycles. On the tester side of the interface the same 32 bits of data is transferred, 16 bits at a time, over two DRAM cycles.

10.2.2 Edge Placement Accuracy

Command, address and data edges should be reasonably close to the appropriate clock edge but with some margin of error. DRAM setup and hold times are 400 to 600 pS, while the half cycle time is at least 1250 pS. As long as the data is within 625 pS of the clock edge it will not violate setup or hold. Since there will be other error terms (DIMM trace length matching, jitter, and so forth) it is recommended the tester be accurate to ±300 pS.

During Transparent mode testing, the core clock is tied to the input reference clock by selecting a special mode in the PLL. Normally, the PLL uses an internal feedback loop for maintaining lock. In this special mode, the end of the core clock tree is fed back as the feedback clock to the PLL. This makes driving and receiving data at the pins of the chip deterministic with respect to the reference clock. This feedback mode for the PLL can be selected automatically in Transparent Mode.

10.2.3 Transparent Mode Timing

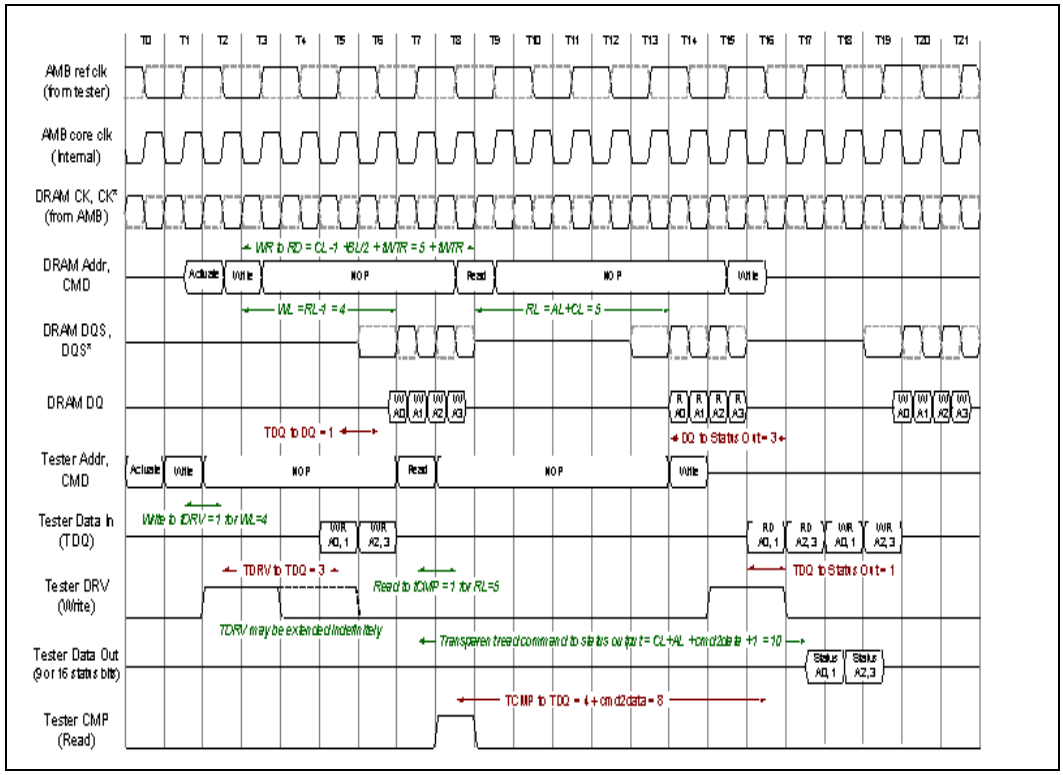
Normally, transparent mode will use DDR2-400 timing even if the DRAM is rated for faster operation. Optionally an AMB may support operation at frequencies higher than 200 MHz.

To set up transparent mode the appropriate AMB registers should be set to AL=0, CL=4 and CMD2DATA=4. Some AMBs may default to these values, in which case programming has no effect. These values establish an internal timing relationship as illustrated in the following figures. Optionally other register values may be supported for special test cases.

Actual placement of DRAM read/write or other commands is dependent on the incoming signals, not the AMB register values. Figure shows an example of a write, read, write sequence using incoming signals that correspond to WL=4 and RL=5. The timing relationships in red (normal font) are fixed relationships (established by the AMB register settings above). These edges will move together. The relationships in green (italic font) are the DRAM timings, which are under control of the tester.

Timing may be changed on the fly (for example, in the middle of a test pattern) by changing the placement of edges from the tester. DRAM mode registers can be programmed on the fly as needed by including (E)MRS commands in the tester data stream. There is no need to change AMB settings during a test. The only exception is that DRAM BL may be changed on the fly but the data logging logic may get confused if DRAM BL does not match the BL expected by the data logger. All other DRAM settings such as AL and CL may be changed at any time.

Figure 10-1. Transparent Mode Timing



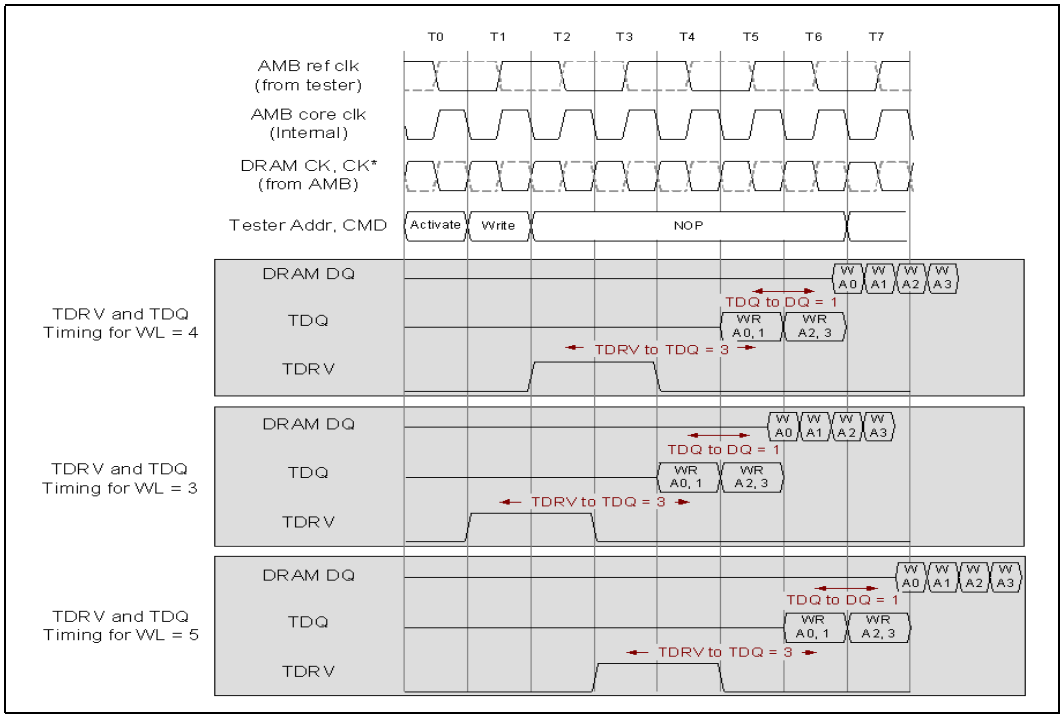
10.2.3.1 Write Timing

Figure 10-2 illustrates write timing with the tester set to WL=3, 4, and 5. There is a constant offset of three cycles from TDRV to TDQ and one cycle from TDQ to DRAM DQ. The DRAM mode registers must, of course, be set to the appropriate timing to recognize the read. For BL=8 the TDRV pulse will be 4 clocks wide rather than 2.

10.2.3.2 Extended Write Timing

The TDRV pulse may be extended indefinitely in cases where it is necessary to apply constant data to the DRAM pins. As long as TDRV remains asserted the AMB will continuously propagate data from the tester TDQ inputs through to the DRAM DQ pins (delayed by 1 cycle) as indicated below.

Figure 10-2. Transparent Mode Write Timing



10.2.3.3 Read Timing

Figure 10-3 shows read timing with the tester set to RL=3 and RL=5. Due to complexities in the handling of read data in an AMB there is a latency of several cycles from the TDRV pulse to TDQ and test status outputs. Specifically there is a constant latency of four cycles plus the programmed CMD2DATA value from TDRV to TDQ (8 cycles total using the AMB settings above) and one cycle from TDQ to DRAM DQ. An AMB may support shorter latencies but this is not required.

TCMP is latched on the rising edge of core clock. This initiates the read inside the AMB. In most cases the AMB will latch DRAM read data slightly before TDQ data is needed. The reason is most AMBs will load DDR data into a queue in the DRAM domain and unload the data on a core clock edge. TDQ is typically not needed until the DRAM data is in the core. The comparison of actual and expected data and propagation back to the tester will occur on the next core clock edge.

The timings below are for BL=4. When testing BL=8 the TCMP pulse should be 3 clocks wide rather than 1. Tester DQ data, DRAM data and the status outputs will be extended appropriately to cover the burst length.

Figure 10-3. Transparent Mode Read Timing

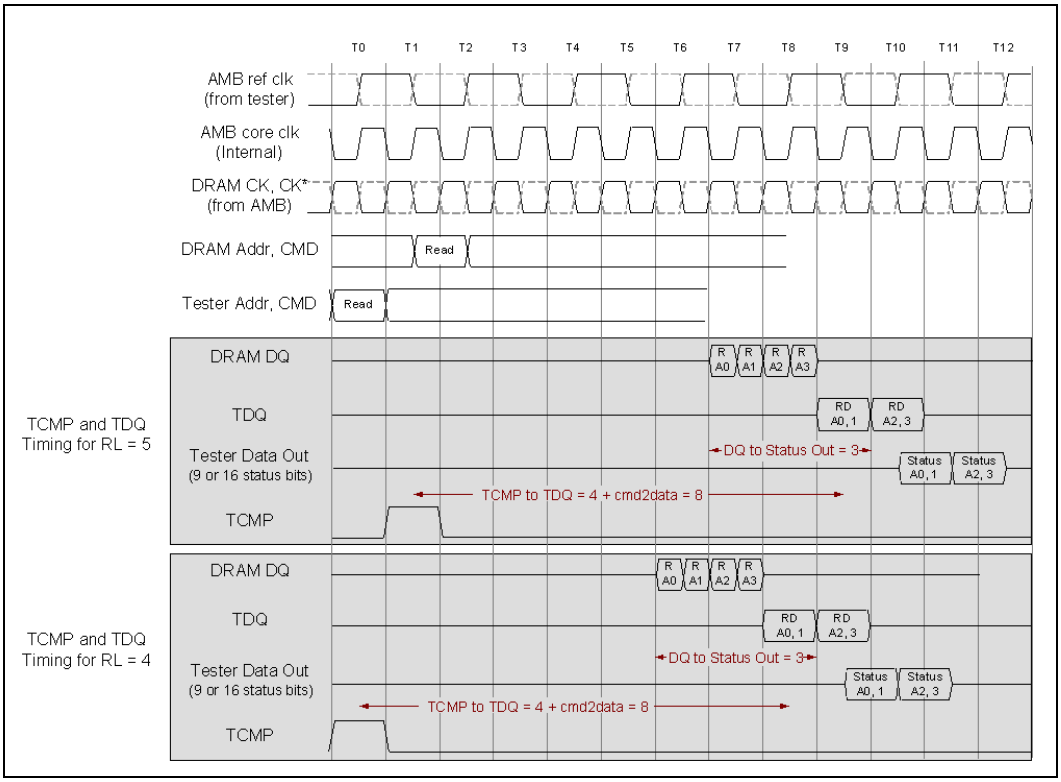
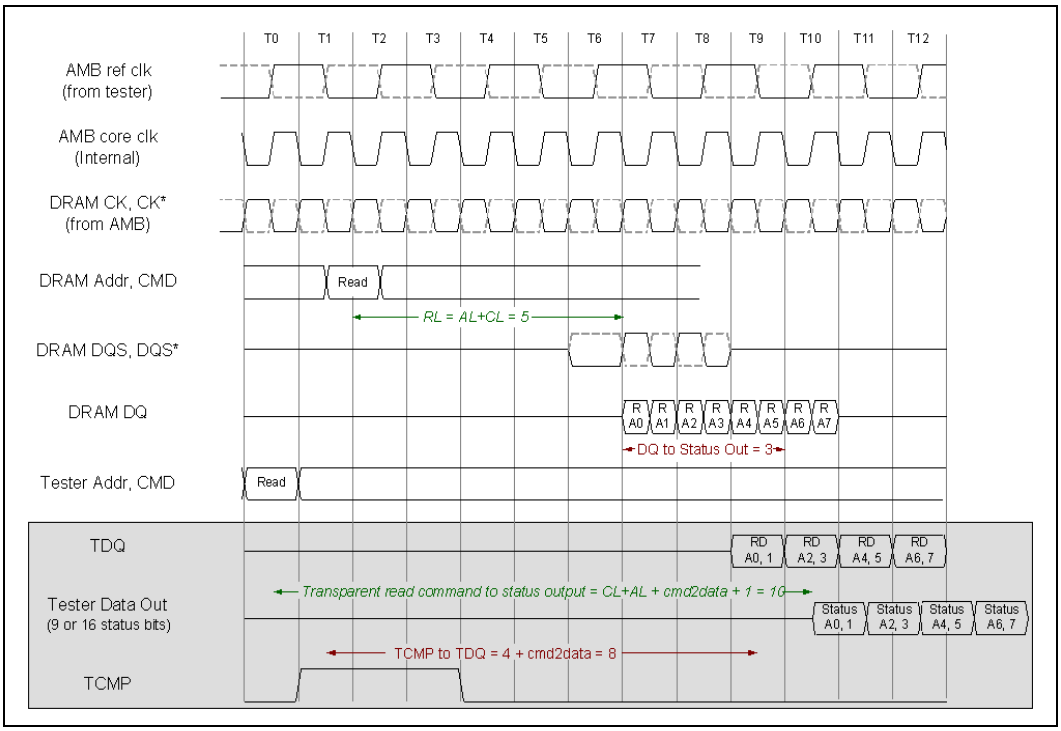


Figure 10-4. BL=8 Read Timing



10.2.4 Error Reporting

By default the status pins will be the xor of actual data from the DRAM and expected data from the tester. The AMB also stores 144 bits of DQ data. If the LGFBITS control bit is cleared the 144 bits of data will be actual data read from the DRAM. Otherwise the result of the data compare is stored. In either case the tester must still provide expected data for the AMB to properly set the status pins.

Normally the test will stop when an error occurs. It is the responsibility of the tester or test program to track errors, read error registers and stop or continue as appropriate. This may require multiple iterations of the same test to execute, collect data and re-start the test.

10.2.4.1 Multiple Failures

There are some implications if multiple failures occur in the same DRAM burst. Three control register bits determine how these failures should be captured. If the log first fail (lgffail) bit is zero (default) the AMB will record only the last failure in a burst. When the bit is set the AMB will record the first failure at the burst position matching the burst position (bstpos) setting. If the bit is set and an error is logged, no further logging will occur until the bit is cleared.

a) If a failure is detected only in one half of the burst (first and second or third and fourth data words), the error pins will indicate which DRAM or DRAMs failed. The error register will indicate which data lines failed and in which data words.

b) If a failure is detected in both halves of the burst (data word 1 or 2 and words 3 or 4), data from the second failure would overwrite data from the first failure. By default this is prevented by the log fail bit. The error pins will continue to operate correctly. If it is desired to collect data from a specific portion of the burst the burst position bits can be used to select an appropriate burst position to record. For a 4 bit burst it is possible to select data from the first or second half of the burst. For an 8 bit burst there are four positions to choose from. These mappings are illustrated in the following figure.

Table 10-3. Mapping of Burst Position Bits to Error Capture

Log First Fail	Burst Position		BL=4			
Bit0	Bit1	Bit0	1	2	3	4
0	x	x	144 bits*		or 144 bits*	
1	0	0	144 bits			
1	0	1			144 bits	

Log First Fail	Burst Position		BL=8							
Bit0	Bit1	Bit0	1	2	3	4	5	6	7	8
0	x	x	144 bits*		or 144 bits*		or 144 bits*		or 144 bits*	
1	0	0	144 bits							
1	0	1			144 bits					
1	1	0					144 bits			
1	1	1							144 bits	

* The last failure will be saved in whatever burst position it occurs.

10.2.4.2 Direct Access - Testing of Individual DRAMs

In certain cases it is desirable to test one or two DRAMs. Transparent mode allows direct access to a single x8 or two x4 DRAMs. In this mode 8 DDR DQ pins are demultiplexed onto 16 SDR status pins, providing16 bit input data path (on TDQ) and a 16 bit output data path on the status pins.

The transparent mode configuration register has one bit (ENDOUT) to enable this mode. On reads, the DRAMRD bits will select the bytes of DRAM data to be presented on the status pins. On writes the DRAMWR bits select a DRAM to receive data from the TDQ bus. A separate register holds 8 bits of default data to be applied to non-selected DRAMs in the early/even cycles and another 8 bits for late/odd cycles. The mapping of these bits to DQ selection is illustrated in the following table.

Table 10-4. Selection of 8 bit Data Paths When ENDOUT is Set

DRAM RD/WR	DQ	Early Data DQ Byte	Late Data DQ Byte
0xF (DRAM WR only)	All Bytes		
8	71:64	8	17
7	63:56	7	16
6	55:48	6	15
5	47:40	5	14
4	39:32	4	13
3	31:24	3	12
2	23:16	2	11
1	15:8	1	10
0	7:0	0	9

DRAMWR is the byte of data bus selected to receive transparent write data, and byte of data bus to be compared against transparent read data. DRAMWR allows a setting of 0xF (all ones) which sends the TDQ input data to all DQ bytes. DRAMRD is the byte of data bus selected to be output on transparent data/status pins when ENDOUT bit is set.

10.2.5 Transparent Mode IO Specifications

Listed below are the specifcations for transparent mode input and output pins.

Table 10-5. Transparent Mode FB-DIMM Interface Signaling Specifications (Sheet 1 of 2)

	Minimum	Maximum	Units
I/O voltage swing	0	500	mV
Input slew rate	2		V/ns
Input to refclk (rising or falling edges) setup time	3000		ps
Input to refclk (rising or falling edges) hold time	1000		ps
Status output valid to refclk time	-1000	+1000	ps
Vref	200	300	mV
Vil (DC)	-300	200	mV
Vil (AC)	-300	150	mV
Vih (DC)	300	900	mV

Table 10-5. Transparent Mode FB-DIMM Interface Signaling Specifications (Sheet 2 of 2)

	Minimum	Maximum	Units
Vih (AC)	350	900	mV
Voh	400		mV
Vol		100	mV
Ioh	8		mA
Iol	12		mA

Note:

1. Ioh: current into a 50-ohm external load to ground, with on-die transmitter termination enabled
2. Iol: current into a 50-ohm external load to 1.5 V supply rail, with on-die transmitter termination enabled

10.2.6 IO Implementation Guidelines

10.2.6.1 Dedicated Receivers

Simple one-stage receivers for the transparent mode have been added in parallel to the existing high-speed sampling receivers. The latter can be turned off during transparent mode, as well as the DRC and the phase interpolator, to save power and avoid noise. An internal VREF set to 0.25 V will be used, so the tester signals should oscillate between 0 and 0.5 V. The transparent mode RX should be turned off during normal mode, so as to save power/avoid noise.

10.2.6.2 Common Clock Scheme

To avoid costly implementations using strobes and FIFOs, a common clock scheme is followed, implemented in the core, where the data capture flops reside. Since transparent mode data signals from the tester are all in phase with the 100 MHz system clock, the clock used for the capture flops has to be aligned with the external system clock (or slightly delayed, to account for the propagation delay difference between data receivers and clock receiver).

Aligning core clock and external clock can be done using the HVM mode circuitry included in the PLL. The HVM clock tree will have to feed the capture flops, and one of its leaves has to be fed back to the PLL, in order to achieve adequate synchronization.

10.2.6.3 Tester Interface Clock and Data Routing

The following uncertainties have to be factored in:

- Tester board (TIU) trace mismatches
- Package trace mismatches
- FBD low speed RX propagation delay variations due to PVT
- Set up and hold of capture flops (typically <350 ps depending on process, voltage and temperature)
- Clock synchronization mismatch, core clock PLL and tree jitter (approximately ±200 pS)

At 200 MHz, there is a 5 ns data window. The potential FBD low speed receiver variation is approximately 300 ps propagation variation over process, voltage and temperature. Package and tester interface mismatches are not expected to exceed

200 ps. Flop setup variation should also be less than 200 pS. After removing 400 ps for clock uncertainties leaves 3.9 ns margin. While this is plenty of margin, some amount of trace matching should be done on the tester interface to minimize skew.

10.2.6.4 Outgoing Control Signals

Only the TX+ pin should connect to the tester. The data on TX- can be discarded (it will toggle at the same rate as TX+).

Terminating TX+ on the tester should be a given, as every tester offers this capability. Terminating TX- could be done on the tester, by having a TIU (tester board) with a route and a tester connection allocated for it. It may be cheaper to just have a 50 ohm resistor tied to ground on the TIU itself, from the TX- pins.

To avoid the crossing clock domains from core clock to FBD fast clocks, the transparent mode data flows directly through the Analog Front-end Unit of the TX.

10.2.6.5 Usage Models

10.2.6.5.1 Host Side Usage

TX: The transmitters are set the same as in normal operation (bias on, enable termination, and so forth.).

RX: RX+ and RX- signals will be independent. Incoming data will free-flow through the I/O (no flops). The routing distance from transparent_rxout pins should be the same for all capture flops. All these flops should be clocked by the dedicated HVM clock.

10.2.6.5.2 Tester Side Usage

The tester interface should have trace-matched data signals, to avoid skews > 1 ns. The tester should have 50 ohm terminations to ground for all TX pins in use (on-tester termination can be used where applicable). The tester should enable 50 ohm terminations to ground for all the signals sent to RX pins. This will guarantee reasonable signal integrity.

10.3 Transparent Mode Control and Status Registers

In transparent mode, CSRs will be accessed and programmed through SMBus. See Section 14.3.5, , “Hardware Configuration Registers,” for register descriptions.

§

11 DDR MemBIST

11.1 MemBIST Overview

The Intel 6400/6402 Advanced Memory Buffer (AMB) supports memory built-in self test (MemBIST) for memory initialization during system boot up and for testing the installed memory. During DIMM manufacturing, MemBIST may be used to apply tests at speed to test the AMB-to-DRAM interface. Table 11-1 below lists the features provided by MemBIST.

At the system level, MemBIST may be executed on multiple DIMMs simultaneously. This could be used to speed memory test during system boot.

During DIMM manufacturing, MemBIST offers a fast method to detect FBD assembly-related defects, interface defects and the majority of memory-core-related defects. This testing may be done through commands initiated across the FBD channel (in-band test initiation) or through SMBus or JTAG commands (out-of-band test initiation), making MemBIST compatible with motherboards, low cost ATE, or standalone equipment such as continuity testers. To perform in-band testing on a motherboard, the motherboard must contain an FBD-based memory subsystem.

MemBIST is primarily intended to test the AMB-to-DRAM interface and not the DRAM core. It is expected that transparent mode will be used to test the core logic in DRAMs already installed on DIMMs. For this reason, MemBIST includes primarily those features needed for interface testing. MemBIST does not include all features needed for DRAM core testing. Traditional system test methods are expected to be used for operating system or application-based testing of the memory subsystem. This may include existing memory stress tests, applications or other tests selected by the DIMM manufacturer.

DDR interface testing requires stress of the AMB DDR I/O circuitry and the DDR I/O-to-core path in the DRAM. The test needs to be able to detect static faults (such as stuck signals) as well as dynamic faults (for example, slow timing paths) in the logic.

Testing this logic requires:

- Delivering patterns at full speed (667 MT/S).
- Incrementing and decrementing addresses. The address decoders are best tested with marching or other non-linear patterns.
- Alternating data patterns (single rotating bits, checkerboards) to detect slow nodes or capacitive coupling in the data path.
- Using standard interface timing (nominal clock cycle time, setup, hold, pre and post-amble).
- Verifying ODT operation at speed

To accomplish the required testing, MemBIST has a number of modes of operation and data formats which can be chosen to meet the specific need. In addition, MemBIST supports a variety of DRAM timings and densities.

A fundamental feature of the MemBIST architecture is that, unlike transparent mode, which simply replicates 8 bits of data across the DQ bus, MemBIST can control individual bits in the 72 bit DQ bus. In addition, MemBIST can apply test patterns at a rate as high as the DRAM address rate. To accomplish this, the MemBIST architecture

provides two 72 bit words, or 144 bits of test data for each DRAM clock cycle. The data is supplied to the DRAMs on "early" and "late" phases of the DDR clock cycle. Early data is provided on the rising edge of CK. Late data is provided one half cycle later on the falling edge of CK. DRAM compare data is treated in a similar manner with appropriate clock alignment.

11.2 MemBIST Feature Summary

Table 11-1 lists the features of MemBIST in summary form. Each feature is explained in subsequent sections. The registers used to control these features are detailed in the MemBIST register section.

Table 11-1. MemBIST Feature Summary (Sheet 1 of 2)

Feature	Feature Description
Memory Address Control	
Address pattern in tests	User defined start and end physical address
	Fast X, Fast Y, Fast XY, XZY address modes
	Choice of incrementing or decrementing addresses
	Dynamic address inversion (DAI) inverts alternate addresses
X (row) address bits	Up to 16
Y (column) address bits	Up to 13 (limited by MTR:numcol to A[13:11,9:0])
Z (bank) address bits	Up to 3
Data Patterns	
Static data patterns	fixed nibble data patterns (0, 3, 5, 6, 9, A, C, F)
	144-bit user-defined data pattern
	288-bit user-defined data pattern
Dynamic data patterns	32-bit user-defined circular shifted data pattern
	Random data pattern derived from user-specified 32-bit seed using an LFSR (CRC32)
Data pattern inversion	Any data pattern can be inverted before being applied
Programmable DRAM Timing Control	
DDR2 DRAM timing	Set in AMB registers
Burst Length	4 or 8
Refresh control	DDR2 refresh intervals programmable in AMB registers
BIST Engine Control	
Fundamental commands	Write, read, read with data compare, write + read with data compare
Access method	all registers and settings accessible using FBD, JTAG, or SMBus
DRAM data width	x4 or x8
DRAM initialization and mode settings	Set by AMB registers
Execution speed	A programmable number of deselect commands may be inserted after DRAM accesses to slow down the speed of execution
Execution control	Halt on error or run to completion of test
	Test abort during the test

Table 11-1. MemBIST Feature Summary (Sheet 2 of 2)

Feature	Feature Description
Failure data access	Failure data logged and accessible using FBD, JTAG or SMBus
	Logging may be delayed to capture later failures
Algorithms Provided	
Algorithms operate over a specified address range with a fixed data pattern of 0xA only Notation for algorithm definitions: ^ = increasing address from start to end v = decreasing address from end to start W / R = Write / Read and check D / I = Data / Inverted Data number = sequence of events (x, y) = for each address, first x is applied, then y, before continuing to next address	
Scan	^ (WD_1); ^(RD_2); ^ (WI_3); ^ (RI_4)
Init	^ (WD_1)
Mats+	^(WD_1); ^(RD_2, WI_3); v(RI_4, WD_5);
MarchC–	^(WD_1); ^(RD_2, WI_3); ^(RI_4, WD_5); v(RD_6, WI_7); v(RI_8, WD_9); v(RD_{10});
Read and check	^(RD_1);
Error Logging	
Error logging	Pass / fail indicator
	Log up to 5 failing addresses
	Record up to 4 sets of 144-bit failure data
	144-bit failure data bit location accumulator marking bit failures through entire test

11.3 MemBIST Operation

11.3.1 Fundamental Operations

The MemBIST logic provides a number of operational modes which can be combined in various ways to provide a large number of useful combinations. For example, there is a mode in which it is possible to select address incrementing first by column and then by row, or the opposite of this. There is also a mode to dynamically invert every other address, which toggles all address lines to opposite states. These two independent modes can be combined to test a bank by toggling row address lines to opposite states or by toggling column address lines to opposite states. Limits on combinations are mentioned where they exist.

MemBIST also provides a number of complete operations which it can perform. MemBIST operations can be characterized as a task which MemBIST carries out automatically after being programmed for the task. The operation begins when MBCSR:start is set by the user and ends when MemBIST clears this same bit. MemBIST operations include built-in algorithms and fundamental commands.

MemBIST built-in algorithms consist of complete testing schemes which are implemented in MemBIST and which carry out complete tests for meeting specific testing objectives. They accomplish their testing completely under automatic control once the operation is started.

In addition, MemBIST has 4 fundamental commands which can be utilized directly by the user. They are write, read, read with data compare, and write followed by read with data compare. They are also used by the MemBIST built-in algorithms. Some MemBIST algorithms also utilize additional fundamental commands that are not available to users. As a result, users cannot duplicate all algorithm functionality by sequentially running individual MemBIST commands.

11.3.2 Memory Addressing

A memory test is characterized by a starting address, an ending address and a direction. The address generation logic has counters for row, column and bank addresses. MemBIST does not change the rank bit during execution. Each rank on a DIMM must be tested by a separate execution of MemBIST.

Addresses in MemBIST are logical addresses, meaning they follow the order of the DRAM external address pins. The actual arrangement of bits in the array (the physical address) will differ from the logical address. Therefore, an addressing scheme or data pattern may not be applied to the array exactly as one might think. For example, accesses to logically adjacent cells will not necessarily access physically adjacent cells. For general purpose testing such as that intended for MemBIST this is not an issue. In-depth array testing is best done on a portion of the array where the logical to physical mapping is known, or in transparent mode where one has full control of address and data sequencing.

11.3.2.1 *Address Definition*

All addresses in MemBIST have three components: a row address, a column address and a bank address. The user communicates these values to MemBIST through 32-bit registers. Column address 0 is not stored since the DRAMs are 72 bits wide and the MemBIST engine has an internal 144 bit architecture. Column address 10 is used for auto-precharge (always low in MemBIST) so it is also not specified in the address registers.

User-defined start and end addresses are constrained to contain a column address modulo the burst length. For instance, at BL=4, assume a memory access starts at column 0. The next access would start at column 4, the next at column 8 and so on (assuming Fast Y address sequencing). The address register bits reflect these constraints. BL=4 allows specification of column bits 14..2, while BL=8 allows specification of column bits 15..3.

Table 11-2. Memory Address Definition, BL=4

Address register bit																														
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
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	R	14	13	12	11	9	8	7	6	5	4	3	2	2	1	0
Row																Column													Bank		

Note: Address register bit 15 "R" = Reserved for future use

Table 11-3. Memory Address Definition, BL=8

Address register bit																													
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
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	R	15	14	13	12	11	9	8	7	6	5	4	3	2	1	0
Row																Column													Bank		

Note: Address register bit 15 "R" = Reserved for future use

11.3.2.2 Address Generation

The address generation logic and controlling register bits allow a variety of methods to traverse the address space of the DRAM. These are described in the next sections.

In these explanations, the symbol X refers to the row address. Row address lines are also called word lines. The symbol Y refers to the column address. Column address lines are also called bit lines. The symbol Z refers to the bank address. The MemBIST start address consists of a triple of row, bank, and column address (X_{start}, Z_{start}, Y_{start}) or (X_s, Z_s, Y_s) for brevity. Likewise, the end address consists of the triple (X_{end}, Z_{end}, Y_{end}) or (X_e, Z_e, Y_e). The row, bank or column symbols may also be used individually to refer to a value without being included in the address triple to which it belongs.

11.3.2.2.1 Address Sequencing Options

MemBIST provides several options to select the address space for MemBIST operation and for sequencing through the address space chosen. Table 11-4 below shows how the MemBIST addresses are generated based on various parameters that affect address generation. Detailed explanations of the entries are found in later sections.

Table 11-4. MemBIST Addressing Behavior

	Address Type (MBSCR:atype)		
Address Sequencing (MBCSR:fast)	**Range (specified in MB_START_ADDR and MB_END_ADDR[a])**	**Full (uses range of 0 to MTR limit)**	**Single (specified in MBADDR)**
XZY	Counter order[b] XZY X, Y increment[3] start to end X, Y decrement[3] end to start Z increments Z_s to Z_e if $Z_s < Z_e$ Z decrements Z_e to Z_s if $Z_s < Z_e$ Following is for Intel 6400/6402 Advanced Memory Buffer only and not required by the FBD DFx Spec Z fixed at Z_s if $Z_s = Z_e$ Z increments Z_s to MTR limit, wraps to 0, then to Z_e if $Z_s > Z_e$. Z_e to Z_s is skipped. • Z decrements Z_e to 0, wraps to MTR limit, then to Zs if $Z_s > Z_e$. Z_e to Z_s is skipped.	Counter order XZY X, Y, Z increment[c] 0 to MTR limit[d] X, Y, Z decrement MTR limit to 0	Single burst to address specified in MBADDR MBCSR:fast ignored
Fast Y	Counter order XY X, Y increment start to end X, Y decrement end to start Z fixed at Z_s. Z_e is ignored.	Counter order XY X, Y increment 0 to MTR limit X, Y decrement MTR limit to 0 Z fixed at 0	
Fast X	Counter order YX X, Y increment start to end X, Y decrement end to start Z fixed at Z_s. Z_e is ignored	Counter order YX X, Y increment 0 to MTR limit X, Y decrement MTR limit to 0 Z fixed at 0	
Fast XY	X and Y count simultaneously X, Y increment start to end X, Y decrement end to start Z fixed at Z_s. Z_e is ignored	X and Y count simultaneously X, Y increment 0 to MTR limit X, Y decrement MTR limit to 0 Z fixed at 0	

Notes:

a. It is required that $X_s < X_e$, $Y_s < Y_e$ and $Z_s < Z_e$. X_s, Z_s, Y_s and X_e, Z_e, Y_e are the values in MB_START_ADDR and MB_END_ADDR respectively

b. Counter order is counter MSB to LSB in the order given.

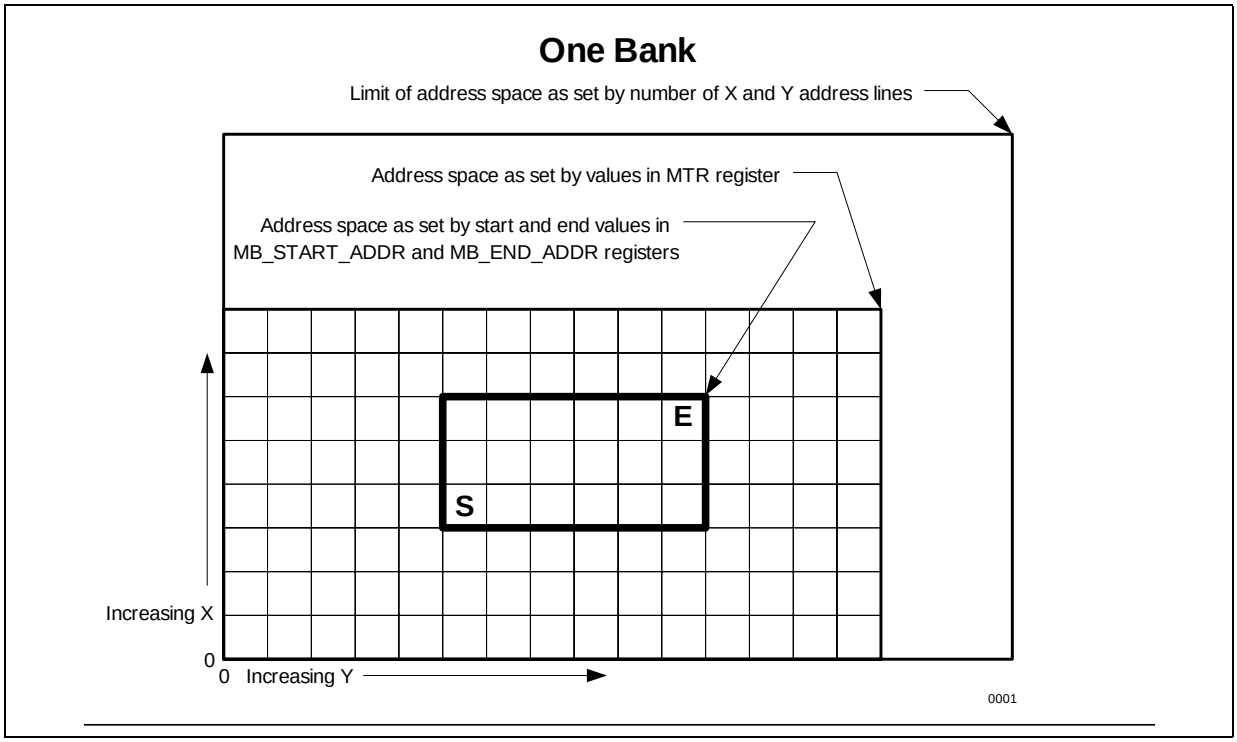
c. Increment or decrement is selected using MBCSR:adir

d. MTR limit is the value shown in MTR for this address and refers to the top of this address range.

11.3.2.3 Address Space

The address space for MemBIST is selected using the MBCSR:atype field. The choices include a single physical address specified in the MBADDR register, a range of addresses as specified using the MB_START_ADDR and MB_END_ADDR registers, and the full address space of the DRAMs as specified in the MTR register. The range and full address spaces are depicted in the next figure, Figure 11-1.

Figure 11-1. Range and Full Address Spaces



As the figure above illustrates, when the values of the MB_START_ADDR (marked **S**) and MB_END_ADDR (marked **E**) registers are used to set the address space for a MemBIST operation, the values in these registers must bear the correct relationship to each other. The start address sets the lower bound for the test address space in both the X and Y directions, and the end address sets the upper bound in both the X and Y directions. For a start and end address to define a usable address space, it is required that $X_s < X_e$, and $Y_s < Y_e$. In addition, when an address sequencing mode is chosen in which the bank field in MB_END_ADDR is significant, it is also required that $Z_s < Z_e$. (If a single physical address is to be tested, MBADDR, not MB_START_ADDR and MB_END_ADDR, is used.)

In every case in which an address endpoint is specified using a register value, MemBIST includes the address endpoint in the operation. For example, if an address range is specified using the MB_START_ADDR and MB_END_ADDR registers, the locations specified in these registers are included as part of the address range. The MemBIST operation will be performed on both of the locations specified in these registers.

11.3.2.3.1 Order and Direction of Address Sequencing

The order of sequencing through the test address space is chosen using the MBCSR:fast field. The options are Fast Y (with fixed bank), Fast X (with fixed bank), Fast XY (with fixed bank) and XZY. The terms Fast X, Fast Y, Fast XY and XZY refer to the order in which addresses are incremented or decremented during MemBIST execution. This order is given in Table 11-4 above as "counter order." Counter order of XZY, for example, indicates that the Y counter is counted most rapidly. When counter Y over or underflows due to reaching the programmed limit of its count, the overflow or

underflow is applied to the next, or Z, counter. When counter Z reaches its limit, its overflow or underflow is applied to counter X. A counter that overflows or underflows resets to its initial starting value and continues counting from that value.

When using XZY addressing, MemBIST accesses the banks specified by MB_START_ADDR and MB_END_ADDR (with range addressing) or bank 0 through the MTR limit (with full addressing).

When using XZY addressing, MemBIST accesses multiple banks.

- With range addressing (MBCSR:atype = 0x2), the banks accessed are in the inclusive range specified by MB_START_ADDR:ba and MB_END_ADDR:ba.
- With full addressing (MBCSR:atype = 0x3), the bank address is bank 0 through the MTR limit.

When using Fast X, Fast Y or Fast XY addressing, MemBIST will traverse a single bank.

- With range addressing (MBCSR:atype = 0x2), the bank is specified by MB_START_ADDR:ba. MB_END_ADDR:ba is ignored.
- With full addressing (MBCSR:atype = 0x3), the bank address is always 0.

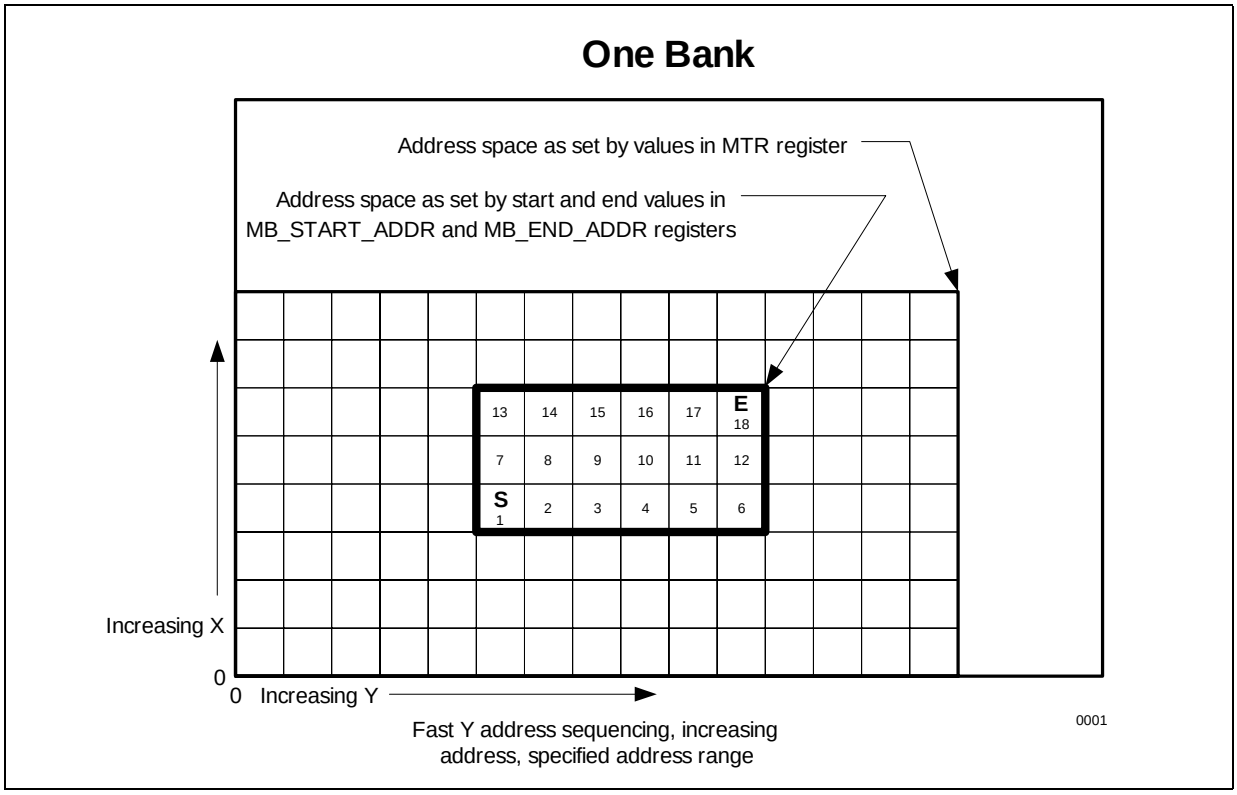
The direction of sequencing through the test address space is chosen using the MBCSR:adir field. In this field, either incrementing addresses or decrementing addresses may be selected. The beginning and ending addresses for the counters depend upon the choice of addressing type programmed in MBCSR:atype. With range addressing, the counter limits are specified in MB_START_ADDR and in MB_END_ADDR. With full addressing, the counter limits are 0 and the MTR limit for each counter. Special cases which exist for the bank address are specified in Table 11-4.

11.3.2.4 Details and Examples

11.3.2.4.1 *Fast Y*

Figure 11-2 below depicts how Fast Y with incrementing addresses cycles through a range of addresses specified by MB_START_ADDR and MB_END_ADDR. In this execution mode, Z, or bank address, is held constant (except in the case of dynamic address inversion mode, DAI, mentioned later). The order in which the locations are accessed is shown in the boxes representing the memory locations. Note that the Y (or column) address counter counts from Y_s to Y_e before incrementing the X (or row) address counter and beginning again at Y_s. This process continues until both the X and Y address counters reach their end values simultaneously. This ending condition results in the address range being covered once. The name "Fast Y" is descriptive of this testing order in which the Y address range changes faster than the X address range.

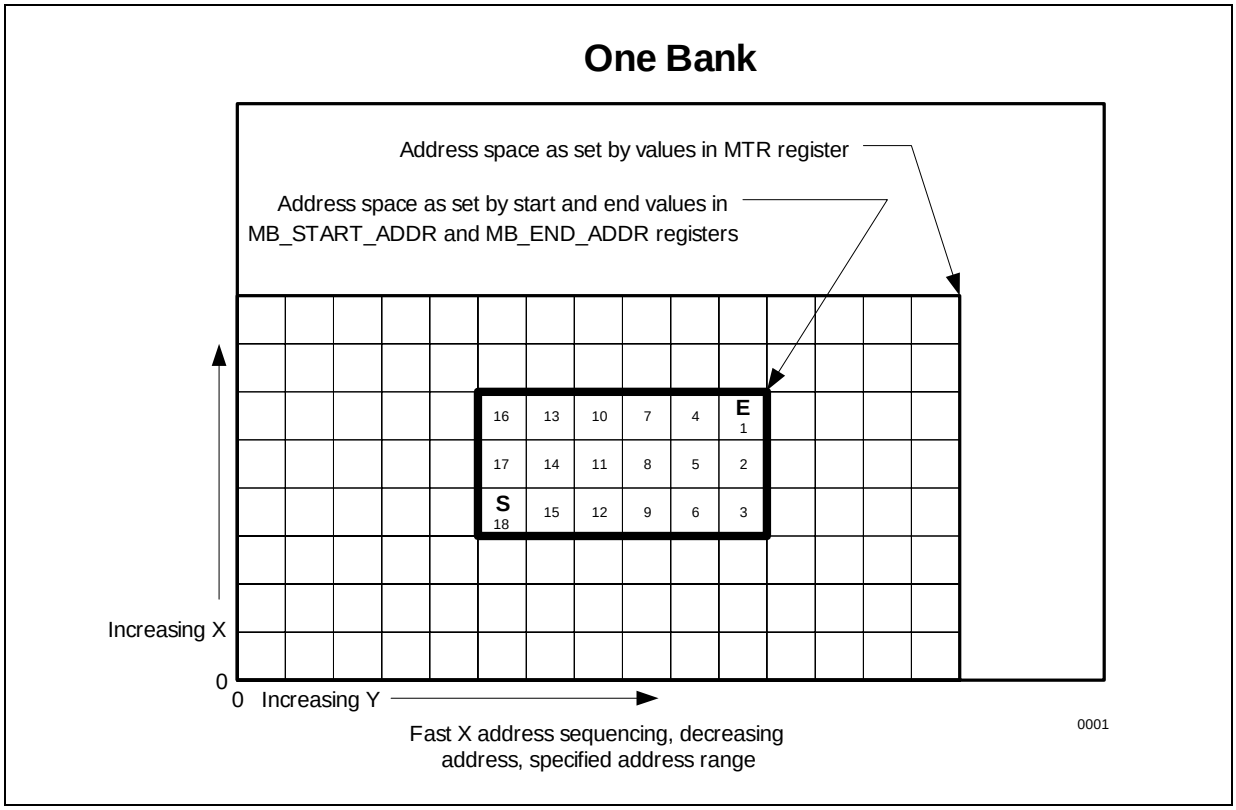
Figure 11-2. Fast Y Address Sequencing



11.3.2.4.2 Fast X

Figure 11-3 below depicts how Fast X with decrementing addresses cycles through a range of addresses specified by MB_START_ADDR and MB_END_ADDR. In this execution mode, Z, or bank address, is held constant (except in the case of dynamic address inversion mode, DAI, mentioned later). Note that the X address counter counts down from X_e to X_s before decrementing the Y address counter and beginning again at X_e. This process continues until both the X and Y address counters reach their end X_s and Y_s values simultaneously. This ending condition results in the address range being covered once. The name "Fast X" is descriptive of this testing order in which the X address range changes faster than the Y address range.

Figure 11-3. Fast X Address Sequencing

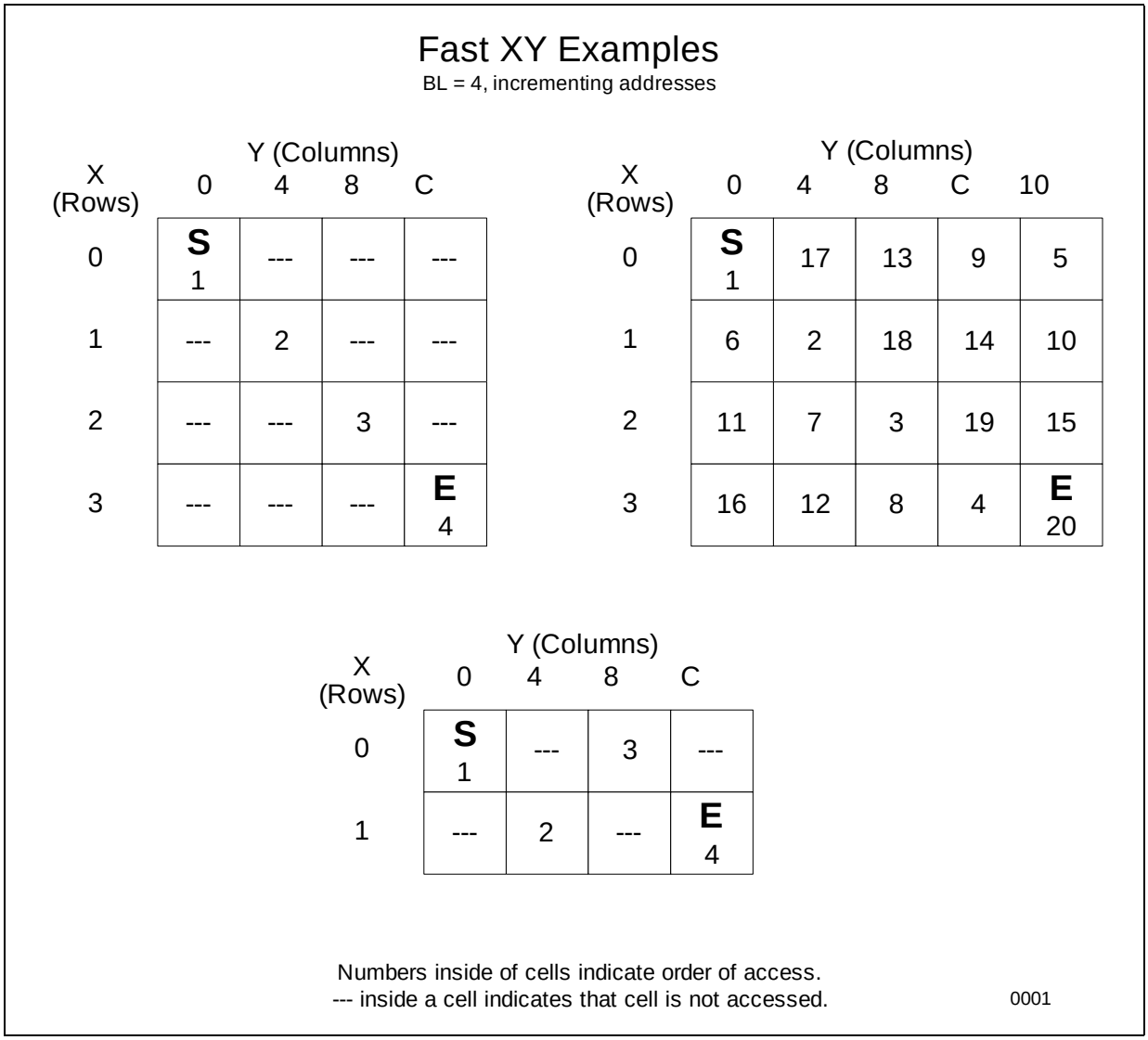


11.3.2.4.3 Fast XY

In Fast XY execution, both the X and Y address counters are changed by one count at the same time. Each of these counters begins at its starting address and increments to its ending address (or decrements from ending to starting address if so programmed) and then reloads on the next count to its initial value. As a result, test execution accesses the RAM locations in diagonal rows across the RAM. Execution stops when both the X and Y address counters reach their end values simultaneously.

In the special case when there are the same number of rows and columns in the address space, only the locations on the diagonal (whose relative row and column positions are equal) will be tested. In the general case in which the specified address range results in different numbers of rows and columns, not all locations may be accessed. Which locations are accessed in this case depends upon the X to Y ratio of the address space specified for the test. See Figure 11-4 below for several examples. Because the starting and ending addresses are always located at opposite corners of the address space, the end address is always eventually reached, and the MemBIST operation always completes.

Figure 11-4. Fast XY Address Sequencing Examples



11.3.2.4.4 XZY Addressing

In XZY (range) address sequencing the MSB-to-LSB ordering of the address counters is Row, Bank, Column, although this is not the order of the fields in the registers used for address specification for MemBIST. As a result of this ordering of the counters, first the columns in one row in the starting bank will be accessed, followed by those on the same row in the second bank, and so on until that row in all banks has been accessed. Then the columns in the next row in the first bank will be selected and the process continued.

11.3.2.4.5 Dynamic Address Inversion (DAI):

Dynamic address inversion (DAI) is provided to maximize the switching of address lines during testing. When dynamic address inversion is enabled, the address counters increment or decrement as usual, but every other address driven to the DRAMs is the logical inverse of the previous address used. The least significant address bit is not inverted since it already toggles at the address rate.

All address lines (X, Y and Z) are inverted by DAI, creating a ping-pong access pattern. This occurs in all address sequencing modes. This behavior might lead to accesses in unexpected portions of the address space. For example, DAI inverts the bank address lines even in Fast X, Fast Y and Fast XY modes, which normally have fixed bank addresses. As a result, every other access is to a different bank than the fixed bank selected by the addressing mode (either MB_START_ADDR:ba field or bank 0 for range or full addressing respectively). Between each access, the old bank must be closed and the new bank must be activated. In another example, with XZY address sequencing and range addressing, it is normal to think of accesses as being restricted to the address range specified in MB_START_ADDR and MB_END_ADDR. But with DAI enabled, the non-inverted accesses will be within the specified range, but the inverted accesses could possibly fall outside of the specified address range.

DAI can be used with any address sequencing mode. It can also be used with either incrementing or decrementing addresses. Table 11-5 gives an example of both address incrementing and address decrementing in DAI mode. This example is of XZY address sequencing with range addressing and shows only low-order bank and column address lines. Shaded rows are non-inverted, non-shaded rows show inverted addresses.

Table 11-5. Dynamic Address Inversion, XZY Address Sequencing and Range Addressing

Normal					DAI, incrementing address					DAI, decrementing address				
Bank	Column				Bank	Column				Bank	Column			
00	0	0	0	0	00	0	0	0	0	00	1	1	1	1
00	0	0	0	1	11	1	1	1	1	11	0	0	0	0
00	0	0	1	0	00	0	0	1	0	00	1	1	0	1
00	0	0	1	1	11	1	1	0	1	11	0	0	1	0
00	0	1	0	0	00	0	1	0	0	00	1	0	1	1
00	0	1	0	1	11	1	0	1	1	11	0	1	0	0
00	0	1	1	0	00	0	1	1	0	00	1	0	0	1
00	0	1	1	1	11	1	0	0	1	11	0	1	1	0
00	1	0	0	0	00	1	0	0	0	00	0	1	1	1
00	1	0	0	1	11	0	1	1	1	11	1	0	0	0
00	1	0	1	0	00	1	0	1	0	00	0	1	0	1
00	1	0	1	1	11	0	1	0	1	11	1	0	1	0
00	1	1	0	0	00	1	1	0	0	00	0	0	1	1
00	1	1	0	1	11	0	0	1	1	11	1	1	0	0
00	1	1	1	0	00	1	1	1	0	00	0	0	0	1
00	1	1	1	1	11	0	0	0	1	11	1	1	1	0
01	0	0	0	0	01	0	0	0	0	11	1	1	1	1
01	0	0	0	1	10	1	1	1	1	00	0	0	0	0
Etc.					Etc.					Etc.				

11.3.2.4.6 Address Inversion for Vtt Balancing

FB-DIMMs use a separate termination (Vtt) power supply for command/address termination. Keep in mind the AMB has two copies of the CA bus. In normal operation the AMB attempts to balance the number of high and low address lines by inverting one of the address ports. Inverting one copy of the addresses reduces Vtt power consumption. This operation is invisible to the controller and DRAM, as this is simply a static inversion of address lines.

By default this behavior is enabled for all memory accesses including during MemBIST operation. Most memory test patterns assume a particular address sequence. Vtt balancing might not be desirable in these cases. The AMB provides a control bit (DRC.BALDIS) to turn off Vtt balancing if desired. If the bit is set, no balancing will take place. If adjacent address inversion is disabled Vtt power may increase substantially, placing additional load on the system power supply.

11.3.3 Memory Data Formatting

During MemBIST operation, 144-bit data is used to write to memory and to check data read from memory. The data register MBDATA allows definition of a 144-bit data pattern. The 144-bit data will be written in two consecutive 72-bit locations within a memory burst. When operating with a burst length of 4 (BL4), early data is written to the first and third locations in the burst and late data to the second and fourth locations. An 8-word burst (BL8) is treated in similar manner, with early data written to odd-numbered locations and late data to even locations.

11.3.3.1 Static Data Formats

MemBIST provides a number of static data patterns which can be selected using bits in MBCSR. A static data pattern is one in which the data remains the same throughout the MemBIST operation. One such static data format is the fixed data pattern. The fixed data patterns are concatenated together multiple times to make up the number of bits required supply data to the DRAMs. In addition to these fixed patterns, the MBDATA registers can be used to supply 144 bits of user-defined data. The data from this static pattern is also concatenated together multiple times to make up the required number of bits for MemBIST use.

Finally, there are certain cases, such as initializing ECC bits in system memory, that require unique data in each data word. This requirement is met by using all 288 bits of the data register MBDATA for user-defined data. This usage of the data register for user-defined data occupies bits which normally are used for failure information. Because the registers normally used for failure information are occupied with the data pattern, when 288-bit user-defined data is used, no checking is performed.

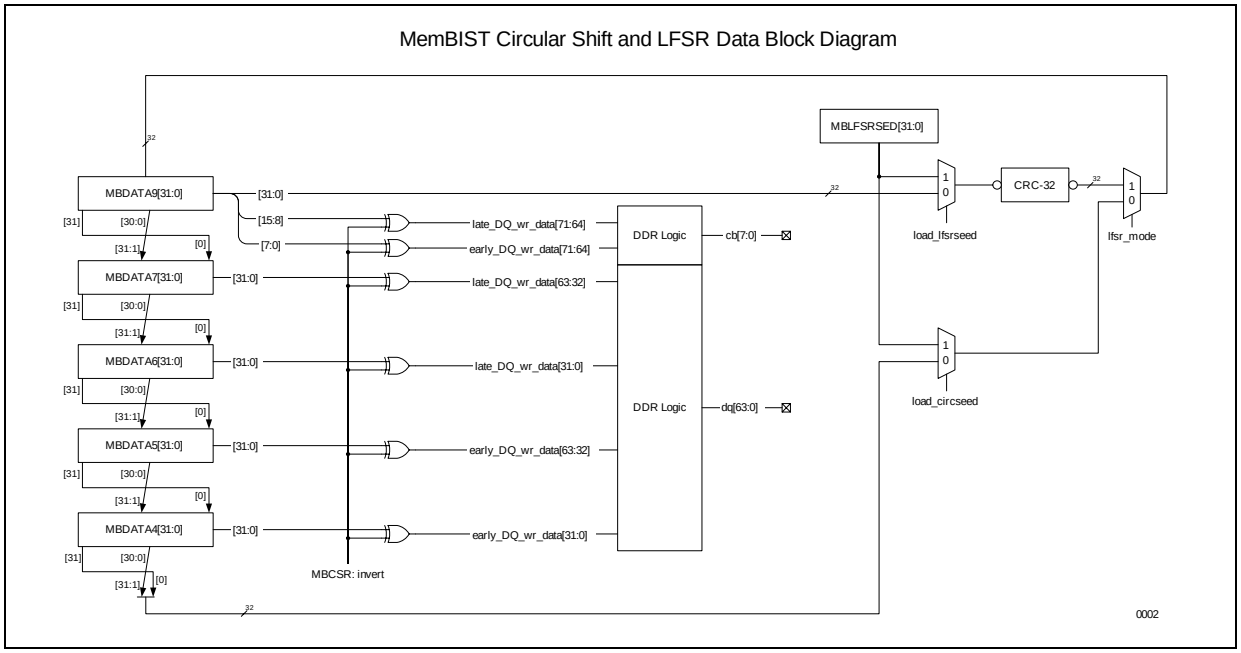
11.3.3.2 Dynamic Data Formats

In addition to the static data formats, MemBIST has the ability to provide dynamically changing data patterns. Dynamic data changes each time that a new address is accessed. These supply shifted data and pseudo-random data.

Note: The following description of circular shift and LFSR random data generation describes the Intel 6400/6402 Advanced Memory Buffer implementation. This does not match the FBD DFx description of these functions. Resolution of these differences will be decided based on potential changes to Intel 6400/6402 Advanced Memory Buffer.

The simplified block diagram of Figure 11-5 below depicts the relationship between the LFSR seed register MBLFSRSED, the MBDATA[9, 7:4] registers, and the data sent on the CB and DQ chip pins when MemBIST executes a write command in the circular shift or in the LFSR data modes. The LFSR seed register is the source of the data written to the CD and DQ chip pins (and thus to the DRAMs) in both of these modes. When MemBIST executes a read command, the data generation is the same, but the data is not sent on the CB and DQ pins. Instead, it is used for comparison against the data received from the DRAMs on the corresponding pins.

Figure 11-5. MemBIST Circular Shift and LFSR Data Block Diagram



In circular shift data mode, the 0 input of the mux at the far right is selected, since this is not LFSR mode. At the beginning of MemBIST execution in this mode, the value of the LFSR seed register is loaded into MBDATA9 by asserting the load_circseed signal shown. Registers MBDATA[7:4] are all cleared to 0 at this time, so that their previous contents are lost.

Each time that MemBIST execution requires data to be written to the DRAMs, 144 bits are taken from the 160 bits comprising MBDATA[9,7:4], as shown in the figure. The 144 bits are either inverted or not before being sent to the DDR logic to be written to the DRAMs, depending upon the state of the MBCSR:invert bit. After using the data, the MBDATA[9,7:4] registers are clocked once, shifting the data around within these registers to form new data to be used by MemBIST the next time that data is required.

For example, when MemBIST starts and MBCSR:invert is 0, the value from the LFSR seed register will be loaded into MBDATA9, and MBDATA[7:4] will be cleared, so that the first data available for MemBIST to write to the CB and DQ pins will consist of MBDATA9[7:0] sent on the early cycle on the CB pins, MBDATA9[15:8] on the late cycle on the CB pins, and 0s sent on all of the DQ pins on both the early and late cycles.

Once this data has been used, the MBDATA[9,7:4] registers are clocked. The data path between the MBDATA registers is wired so that the data undergoes a left circular shift when passing to the next register. Bits [30:0] become bits [31:1] in the receiving register, and bit [31] becomes bit [0] in the receiving register. For example, 0x8000_0001 in MBDATA9 becomes 0x0000_0003 upon being loaded into MBDATA7.

In LFSR data mode, a similar procedure is followed, with the exception that the lfsr_mode signal causes the mux on the far right of the figure to select the output of the CRC-32 block to provide the input to register MBDATA9. At the beginning of MemBIST execution, load_lfsrseed asserts once to use the value in the LFSR seed register as the initial input to the CRC-32 block. On subsequent register loads, the current value in MBDATA9 is used as the input to the CRC-32 block. Each time that MBDATA9 is loaded with a new data value from the CRC-32 block, each of MBDATA[7:4] also load new values from their inputs.

Before the first 144 bits of data are used by MemBIST in LFSR mode, 5 register loads occur. This fills all bits in MBDATA[9,7:4] with random data values before the first data is written to the DRAMs.

Table 11-6 below shows an excerpt from an example run of MemBIST executing with MBCSR:dtype = 2, circular shift data. The table illustrates the relationship between the LFSR seed register MBLFSRSED, the BMDATA[9, 7:4] registers, and the data sent on the CB and DQ chip pins when executing in the circular shift mode with MBCSR:invert = 0. The table also illustrates how the circular shift occurs in the MBDATA registers. This further illustrates the process explained and shown above.

Table 11-6. Example of Circular Data Shifting

#	Mbdata9 xxL[71:64]E[71:64]	Mbdata7 L[63:32]	Mbdata6 L[31:0]	Mbdata5 E[63:32]	Mbdata4 E[31:0]	Early CB	Late CB	Early DQ	Late DQ
1	0000_0001	0	0	0	0	01	0	0	0
2	0	0000_0002	0	0	0	0	0	0	0000_0002 _0000_0000
3	0	0	0000_0004	0	0	0	0	0	0000_0000 _0000_0004
4	0	0	0	0000_0008	0	0	0	0000_0008 _0000_0000	0
5	0	0	0	0	0000_0010	0	0	0000_0000 _0000_0010	0
6	0000_0020	0	0	0	0	20	0	0	0

In the example shown, MBLFSRSED contains 0x0000_0001 when MemBIST execution starts. MBCSR contains 0x8022_0a90. Line 1 shows the data from MBLFSRSED loaded into MBDATA9, and shows how 144 bits of the data from MBDATA[9,7:4] are sent on the CB and DQ pins when required by MemBIST. Line 2 shows the result of the circular shift of MBDATA9 into MBDATA7, as explained previously. Line 2 also shows how this second set of 144 bits of data will appear on the CB and DQ pins when it is used by MemBIST. The table shows how the original data from MBLFSRSED is transferred from register to register each time that data is required by MemBIST execution, with a one-bit left circular shift occurring at each register transfer. Line 6 shows the data from MBDATA4 arriving at MBDATA9, again with a circular shift left.

Table 11-7 below shows an excerpt from an example run of MemBIST executing with MBCSR:dtype = 3, LFSR data. The table illustrates the relationship between the LFSR seed register MBLFSRSED, the MBDATA[9, 7:4] registers, and the data sent on the CB and DQ chip pins when executing in the LFSR data mode. The table also illustrates how the circular shift occurs in the MBDATA registers during LFSR data mode with MBCSR:invert = 0. This further illustrates the process shown in the figure above.

Table 11-7. Example of LFSR Random Data

#	Mbdata9 xxL[71:64]E[71:64]	Mbdata7 L[63:32]	Mbdata6 L[31:0]	Mbdata5 E[63:32]	Mbdata4 E[31:0]	Early CB	Late CB	Early DQ	Late DQ
1	27fe_3b3d	0	0	0	0				
2	4e2e_350c	4ffc_767a	0	0	0				
3	C9a8_9bae	9c5c_6a18	9ff8_ecf4	0	0				
4	97ca_9c36	9351_375d	38b8_d431	3ff1_d9e9	0				
5	662f_8edd	2f95_386d	26a2_6ebb	7171_a862	7fe3_b3d2	dd	8e	7171_a862 _7fe3_b3d2	2f95_386d _26a2_6ebb
6	4031_1dd0	cc5f_1dba	5f2a_70da	4d44_dd76	e2e3_50c4	d0	1d	4d44_dd76 _ e2e3_50c4	cc5f_1dba _5f2a_70da

In the example shown, MBLFSRSED contains 0x93d7_8768 when MemBIST execution starts. MBCSR contains 0x8011_0b90. Lines 1 through 5 show filling of MBDATA[9,7:4] with random data generated by the CRC-32 block. For each line, newly generated random data is loaded into MBDATA9, and MBDATA[7:4] are loaded with data which is the result of a circular left shift from the register above it. Line 5 shows how 144 bits of the data from MBDATA[9,7:4] is sent on the CB and DQ pins when required by MemBIST. Line 6 shows the result of the circular shift of the contents of each MBDATA register into the next MBDATA register. Line 6 also shows how this second set of 144 bits of data will appear on the CB and DQ pins when it is used by MemBIST. The table shows how the random data from MBDATA9 is transferred from register to register each time that data is required by MemBIST execution, with a one-bit left circular shift occurring at each register transfer.

11.3.4 Algorithmic Testing

MemBIST provides several of the more common algorithmic tests. Several of these are directed at basic operation such as initializing memory or verifying proper connectivity of the AMB and DRAM on a DIMM. In addition there are a few tests that are intended to perform testing of the AMB address generators and DRAM internal address decoders, multiplexers and related logic that is not otherwise testable at speed.

The algorithm engine takes control of the MemBIST engine to carry out the algorithm steps. It does this by writing to some of the control bits in MBCSR. As a result, those control bits used by the algorithm engine are not available for setting by the user. However, the remaining control bits may be utilized by the user to control the algorithm. The MemBIST controls which may be selected by the user when using one of the built-in algorithms are:

- Address sequencing (Fast X, Fast Y, Fast XY, and XZY)
- Rank selection
- Halt or continue on error

When an algorithm is used, the data used for the test is always the fixed pattern 0xA. No other pattern may be chosen. The address space for the algorithm is always the address range specified by MB_START_ADDR and MB_END_ADDR. If testing of the full address range of the DRAM is required, then the MB_START_ADDR and MB_END_ADDR registers must be set to this address range.

When using the built-in algorithms, the initial command for the algorithm must be entered into MBCSR:cmd. For all algorithms except data retention read, the initial command to enter is write (0x1). For the data retention read algorithm, the initial command to enter is read and check (0x3).

Because the algorithm engine takes control of the MemBIST engine to carry out the algorithm steps, a guess about the algorithm step during which the first failure occurred can be formulated by examining the various MemBIST control registers after MemBIST halts on error. After the operation halts, the control register values will generally still reflect their settings when the algorithm step detected the failure.

The algorithms provided by MemBIST are specified in the following sections. The algorithms are specified using the following notation:

^ = increasing addressing. Addresses will be counted up, starting at 0 or the user-defined start address as appropriate.

v = decreasing addressing. Addresses will be counted down, starting at the end of the array or from the user-defined end address, as appropriate.

w or r = Write or Read with checking of the data against the expected data.

D or I = Data or Inverted Data

number = sequence of events

( ) = back to back operations. Example: (wD, rD) will read and then write the same cell before moving to the next cell.

11.3.4.1 Initialization Tests

Init: ^ $(wD)_1$
This is a simple memory initialization algorithm. Data is written to the array with incrementing addressing.

Read and check: ^$(rD)_1$
This test is used to verify memory contents. One use of this is data retention testing. Init may be used to write known data in the array. The tester or system can then alter an environmental condition, or simply wait for some period of time, and then read the array to see if the data changed.

Scan: ^$(wD)_1$; ^$(rD)_2$; ^ $(wI)_3$; ^ $(rI)_4$
The scan test writes data to the array then reads the data back. The second half of the test writes inverted data and reads it back. Scan is a 4N pattern, meaning test time will be 4 traversals, times the size of the array (rows * columns * banks, also known as 'N') times the average time for a read or write operation.

11.3.4.2 Memory Stress Tests

Mats+: ^$(wD)_1$; ^(rD_2, wI_3); v(rI_4, wD_5)
The Mats+ algorithm initializes the array to a known data background and steps through with a read-write-inverted-data sequence. The algorithm is performed with both incrementing and decrementing addressing. Mats+ will detect stuck at faults and basic address decoder faults. The algorithm is order 5N.

MarchC-: ^$(wD)_1$; ^(rD_2, wI_3); ^(rI_4, wD_5); v(rD_6, wI_7); v(rI_8, wD_9); v$(rD)_{10}$
The MarchC- algorithm initializes the array to a known data background and steps through the array in both count-up and count-down addressing with a read-write sequence. MarchC- tests the array decoders and basic neighboring faults. As might

be determined from the sequence numbering this is a 10N pattern.

11.3.5 Error Reporting and Control

Information about failures detected by MemBIST is reported in a variety of ways. These are summarized here and detailed in the following sections. MBCSR:pf set during a MemBIST operation indicates that a failure was detected. MBDATA can log the addresses of up to 5 failures. When the MemBIST operation is utilizing a fixed data pattern selected using MBCSR:dtype = 0x0, MBDATA also contains a failure data bit location accumulator, which logs any data bit that has seen a failure during the operation. With other dtypes, the failure data bit location accumulator may replace the address logs. MB_ERR_DATA logs the first four 144-bit data words which MemBIST detects as containing an error after the Memory Test Failure Address Pointer Register (MBFADDRPTR) has counted down to zero. MBFADDRPTR makes it possible to extract the failure data information for all repeatable failures which occur during execution of a MemBIST operation.

How the available logs in are filled depends partly upon the MemBIST configuration and partly upon how the errors detected by MemBIST occur. For example, when MemBIST is set to halt on error and a failure is detected, MemBIST does not advance to the next address. However, due to the architecture of the memory system, there may be more than one error to log from the single address that was being checked when the failure was detected. At burst length of 4, for each address there are four 72-bit bus-widths of data to be checked. Data is logged in 144-bit quantities. If a bit in the DRAM bus was stuck, this might result in an error being detected in each of the four 72-bit bus-widths of data. Even though MemBIST is set to halt on error, this would result in two 144-bit data error logs, each with a corresponding failure address log. Operating at BL=8 could result in up to four 144-bit data error logs being written with their corresponding address logs.

When MemBIST is not set to halt on error, logs will fill as errors occur. The logs may contain information from several unrelated failing addresses. The first four failures to be detected will cause the first four 144-bit failure data chunks to be logged along with their corresponding addresses. The address of the fifth error to occur will also be logged, although there is no location available to log the failure data. The sixth and succeeding errors will not be logged. During execution of a MemBIST operation, MemBIST will not overwrite a log entry that has already been filled during that operation to record a more recent failure.

11.3.5.1 Control Register Pass/Fail Bit and Halt On Error

The MemBIST pass/fail bit, MBCSR:pf, is always cleared when MemBIST execution starts. If a failure is ever detected, this bit is always set immediately. If MBCSR:halt (halt on detection of read-compare error) is set, and MBCSR:pf is then set due to detection of an error, MemBIST execution always completes the accesses to the current address and then halts without proceeding to the next address. No other MemBIST controls, including a non-zero value in MBFADDRPTR, modifies the setting of the failure bit or the halting on failure detection as described.

11.3.5.2 Failure Address Logging

The addresses of the first 5 detected failures can be logged by MemBIST. The address logs (located in MBDATA) are always cleared, and the logging pointer is always reset to start logging with the first log, each time that MemBIST starts execution of a new operation.

Failure addresses are logged in a slightly different format than the format which is used when specifying start and end addresses. To compress the failure address into 32 bits, bits that are always zero are not stored in the log. These unstored bits include AutoPrecharge Column address [10] and the least significant bits assumed by burst length. In BL=4 operation, column bit 1 in the log is replaced by a bit which distinguishes whether the error was detected in the first 144-bit data chunk (the bit is 0) or in the second chunk (the bit is 1). In like manner, for BL=8 operation, column bits 2:1 are replaced by two bits to identify the data chunk containing the error. The bits used for BL=8 are shaded in Table 11-8 below. See Section 14.5.2.3.1, "MBDATA Failure Address Mapping" for more information.

Table 11-8. Address Log to Bank, Row and Column Bit Correspondence

Address Log Register Bit																													
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
2	1	0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	14	13	12	11	9	8	7	6	5	4	3	2	1
Bank			Row																Column											

11.3.5.3 Failure Data Bit Location Accumulator

MemBIST provides a (sometimes optional) 144-bit failure data bit location accumulator, maintained in the MBDATA registers. Each bit tracks a data bit during MemBIST operation. This accumulator is automatically zeroed each time that MemBIST starts execution of a new operation.

An accumulator bit set to 1 indicates that the corresponding data bit failed to have the expected value at least once during the operation. This accumulator is used primarily for recording detected failed data lines to facilitate DRAM replacement during DIMM manufacturing.

Table 11-9 below shows the correspondence between the DRAM data bus bits and the 160 bits in the MBDATA registers used to store the failure data bit location accumulator.

Table 11-9. Failure Data Bit Location Accumulator to MBDATA Bit Correspondence

MBDATA[8]												MBDATA[3]			MBDATA[2]			MBDATA[1]			MBDATA[0]		
31		24	23		16	15		8	7		0	31		0	31		0	31		0	31		0
Unused						71		64	71		64	63		32	31		0	63		32	31		0
						Late data			Early data			Late data			Late data			Early data			Early data		

11.3.5.4 Failure Data Logging

Failure data is logged in the MB_ERR_DATA registers. These data logging registers are **not** cleared when MemBIST starts execution of a new operation, although the corresponding address logs in the MBDATA registers **are** cleared. The user desiring to discard failure data logs from a previous operation must write zeros to these registers before starting the next operation. During execution of a subsequent operation, when the first failure is detected, both an address and a data log are written, beginning with the first error log entry. Any existing data in the data logging registers used for this log entry is overwritten. The data in the other data logging registers is not affected.

Table 11-10 shows the correspondence between the failure number, the address log for this error, and the data log for this error. Which register is used to log the address of the failure depends upon settings in MBCSR:dtype and MBCSR:algo. If MBCSR:algo is non-zero, then an algorithm is selected, and fixed data is used. Address logging will occur according to the "Fixed data pattern" entries in Table 11-10.

If MBCSR:algo = 0x0 (no algorithm), then MBCSR:dtype determines the failure address error logging registers. For dtype = 0x0 (fixed), address logging will occur according the to "Fixed data pattern" entries in Table 11-10. For dtype other than 0x0 (that is, circular, random, or user data), if MBCSR:mbdata = 1, then the "Other data pattern" entries show the address logging registers. For these dtypes, if MBCSR:mbdata = 0 (accumulator), no failure address logging is available.

Table 11-10. Failure to Logging Register Correspondence

Failure number	Address logged in	Data logged in
1	Fixed data pattern: MBDATA4 Other data pattern: MBDATA0	MB_ERR_DATA0[4:0]
2	Fixed data pattern: MBDATA5 Other data pattern: MBDATA1	MB_ERR_DATA1[4:0]
3	Fixed data pattern: MBDATA6 Other data pattern: MBDATA2	MB_ERR_DATA2[4:0]
4	Fixed data pattern: MBDATA7 Other data pattern: MBDATA3	MB_ERR_DATA3[4:0]
5	Fixed data pattern: MBDATA9 Other data pattern: MBDATA8	Not logged
6 +	Not logged	Not logged

11.3.5.5 Multiple Failures

MemBIST may detect many more failures than there are available registers in which to log them. For some test purposes, the information captured by the failure data bit location accumulator may be sufficient. However, in other situations, capturing all of the failure data and failure addresses may be significant. As long as the failures are repeatable, MemBIST provides a mechanism to extract the failure information about all failures which MemBIST detects during execution of each MemBIST operation. Repeatable failures are ones which occur each time that a test is run.

MemBIST uses the MBFADDRPTR register to discard logs for certain errors, allowing other errors to be logged instead. (However, this register has no effect on setting bits in the failure data bit location accumulator.) The error logging described in the previous section actually occurs on the first through fifth errors ***after the MBFADDRPTR register equals zero***. Since this register defaults to zero, this is the normal logging behavior. However, if storing of logs for later errors is desired, the MBFADDRPTR register should simply be set to the number of earlier logs which are to be discarded.

For example, suppose that MemBIST has been executed, and 5 logs were stored in the failure address logs. Full information was available for the first four (because both address and data logs exist). To collect both the failure address and data for the fifth failure, set MBFADDRPTR to 0x4 and rerun MemBIST. Each of the first 4 logs will be discarded, and will decrement MBFADDRPTR until it reaches 0. The fifth log will be stored, as it is the first to occur after MBFADDRPTR reached 0.

This will work no matter what the position of the error is within the burst. Referencing the above example, if the fourth error is within a burst, and the fifth error is within the same burst but not in the same 144-bit chunk containing the fourth error, the fourth error will not be logged and the fifth error will be logged.

To use this feature, MemBIST must be set to continue on error by clearing MBCSR:halt. Otherwise MemBIST will halt on detecting the first failure and later failures will not be detected or logged. Neither setting of the failure bit, nor halting on failure detection is affected by a non-zero value in MBFADDRPTR.

The general procedure for using this feature is to run MemBIST, read and record the address and data failure information from the logs, and then add the number of failure logs received to the value in MBFADDRPTR to cause the failure information already captured to be ignored on the next pass. This process is repeated until MemBIST completes with zeros in the address logs. This indicates that no further errors were detected beyond those already logged. (Watch the case of decrementing addressing ending at address 0. A failure at address 0 with failure data of zero is possible.)

To generate a complete data log of all failures, use the following procedure:

1. Define starting and ending addresses, address modes and data patterns, set MBFADDRPTR to zero, set halt on error to 0 (don't halt on error).
2. Start MemBIST and check the result. If pass, exit. If fail, continue.
3. Read out the failing addresses and data. If the address and data logs are zero (and the address space covered by this pass of MemBIST does not include address 0), exit, as all failures have been seen. Otherwise, continue. (If the address space covered by this pass of MemBIST does include address 0, and it is ambiguous whether or not address 0 has failed, the ambiguity can be resolved by retesting only address 0.)
4. Add the number of complete logs (for which both address and data were read) obtained during this pass to the value in MBFADDRPTR so that these failures will not be seen again on the next pass of MemBIST.
5. Go to step 2

11.3.6 DRAM Throttling

MemBIST can generate high DRAM bandwidth and consequently, high power consumption and thermal stress. Although this is manageable in a dedicated test environment, a system's power and cooling capacity may be exceeded. For this reason MemBIST allows insertion of deselect cycles on every address change. The deselects will be inserted after each read or write command. This allows slowing down BIST execution if necessary to stay within a given power or cooling envelope.

11.3.7 Refresh Control

In normal operation, refresh commands are issued by the host rather than by the AMB. However, during MemBIST, when commands to the DRAM originate from the AMB, refresh must be provided by MemBIST. The refresh and MemBIST state machines are implemented as separate state machines, allowing refresh when MemBIST is not active. The refresh interval is programmable by setting a 15 bit refresh counter. For example, at the maximum address frequency of 400 Mhz (250 pS), 3120 clocks are needed to achieve the nominal refresh interval of 7.8 µS. The maximum interval in this case will be 81.9 µS. The default refresh value should be usable in most circumstances. The refresh interval may be programmed to other values to meet special needs.

Table 11-11. Refresh Programming

		spec (uS)	clk period (uS)	count
tREFI (15 bits)	min	0	0.0025	1
	spec	7.8	0.0025	3120
	max	81.9	0.0025	32768

11.3.8 DRAM Initialization

As in other operating modes, MemBIST requires the DRAM be initialized prior to the start of any operation. DRAM initialization is accomplished by starting the initialization engine (using the DCALCSR register). This may be done in band or out of band.

11.4 MemBIST Memory Test Examples

In general, using MemBIST will involve the following five steps:

1. **Define DRAM characteristics.** Set up AMB registers for normal DRAM operation. This includes programming the MTR register to match the geometry of the DRAMs on the DIMM. In addition, program the DRT register to match the DDR2 timing of the DRAMs installed on the DIMM. MemBIST uses settings from MTR and DRT to match its operation to the characteristics of the DRAMs being tested. Change the default refresh interval if it does not meet the test objectives. Perform any other normal initialization.
2. **Define the test address space.** Define the starting and ending physical Row/Column/Bank address range which MemBIST is to test in registers MBADDR, MB_START_ADDR, and MB_END_ADDR as required. Which of these registers is used depends upon which address range is selected in MBCSR:atype and whether or not an algorithm is selected.
3. **Enter required user test data.** Define any required user test data for this test, using registers MBDATA[9:0] and MBLFSRSED as appropriate. Which of these registers is used depends upon the setting of MBCSR:dtype, MBCSR:algo, and MBCSR:enable288.
4. **Set test parameters and start MemBIST.** Write MBCSR with MBCSR:start set and other bits set as required to select the desired MemBIST operation.
5. **Evaluate the result.** The test result is available through MBCSR:pf. Depending upon which settings were specified in MBCSR, error information is contained in the MBDATA[9:0]and MB_ERRDATA[4:0] registers.

The next sections provide specific examples of MemBIST execution for selected test cases.

11.4.1 Write a Fixed Pattern to a Range of DRAM Addresses

The following points describe the programming required to write a fixed data pattern to all of the addresses within a selected address range, and to optionally check this data, using MemBIST.

1. Set up registers for normal DRAM operation.
2. Program MB_START_ADDR and MB_END_ADDR registers to the desired address range to be tested.

3. No user data is required, as fixed data (selected in MBCSR) will be used for this test. Therefore, the MBDATA[9:0] and MBLFSRSED registers are not written.
4. Program MBCSR.

   These fields are required for the specified test:

   — Program DTYPE (bits [9:8]) to 00 to select fixed data pattern.

   — Program ATYPE (bits [7:6]) to 10 to use the address range already defined in the MB_START_ADDR and MB_END_ADDR registers.

   — Select either Rank 0 or Rank 1 by programming CS (bits [21:20]).

   — MBDATA (bit 14) and ENABLE288 (bit 15) are not relevant when fixed data has been selected. INVERT (bit 19) is unnecessary for fixed data, since the data choices include inverses for all fixed data patterns. Rewrite these fields with their default values.

   The values for these fields can be selected to choose options for use during MemBIST operation:

   — Program ABAR (bit 13) to select DAI if desired.

   — Either program CMD (bits [5:4]) to be "01" (write only without data comparison) or "11" (write followed by read with data comparison).

   — Program FAST (bits [11:10]) to select Fast X, Fast Y, Fast XY, or XZY (column->bank->row) address sequencing.

   — Program ADIR (bit 12) to select whether addresses should increment or decrement.

   — Program FIXED (bits [18:16]) to specify which fixed data pattern to apply.

   Set these control values to start the MemBIST engine.

   — Set ALGO (bits 26:24) to 0 to prevent the algorithm engine from overwriting bits it controls in MBCSR.

   — Set ABORT (bit 28) to 0.

   — Clear PF (bit 30). Hardware will set this bit if a failure is detected.

   — If desired to halt whenever there is an error, set HALT (bit 29).

   — Set START (bit 31) to 1 to start MemBIST execution.
5. Check the MemBIST results, and if checking was enabled, observe failure data or address through MBDATA registers and MB_ERR_DATA registers:

   — Check MBCSR:start. 0 means MemBIST has completed. Check MBCSR:PF. 1 means a failure has occurred.

   — Failure data bit location accumulator will be stored in MBDATA[8, 3:0].

   — Up to 5 failure addresses will be placed in MBDATA[9, 7:4]

   — Failure data will be stored in MB_ERR_DATA[3:0][4:0] registers.

11.4.2 Write Random Data to a Range of DRAM Addresses and Check

This describes writing randomly generated data to a range of DRAM addresses and checking this data.

1. Set up registers for normal DRAM operation.
2. Program MB_START_ADDR and MB_END_ADDR registers to the desired address range to be tested.

3. Random data is created by the LFSR using the seed value in MBLFSRSED. Either a value can be written, or the default value can be used. MBDATA[9:0] is unused and is therefore not written.
4. Program MBCSR.
   These fields are required for the specified test:
   - Program DTYPE (bits [9:8]) to 11 to select LFSR-generated random data.
   - Program CMD (bits [5:4]) to be 11 (write followed by read with data comparison).
   - Program ATYPE (bits [7:6]) to 10 to use the address range already defined in the MB_START_ADDR and MB_END_ADDR registers.
   - Select either Rank 0 or Rank 1 by programming CS (bits [21:20]).
   - ENABLE288 (bit 15) is not relevant when random LFSR data has been selected. INVERT (bit 19) is unnecessary for random data. Rewrite these fields with their default values.

   The values for these fields can be selected to choose options for use during MemBIST operation:
   - Program ABAR (bit 13) to select DAI if desired.
   - Program FAST (bits [11:10]) to select Fast X, Fast Y, Fast XY, or XZY (column->bank->row) address sequencing.
   - Program ADIR (bit 12) to select whether addresses should increment or decrement.
   - MBDATA (bit 14) is set to select failure address logging or failure data bit location accumulator logging in MBDATA.

   Set these control values to start the MemBIST engine.
   - Set ALGO (bits 26:24) to 0 to prevent the algorithm engine from overwriting bits it controls in MBCSR.
   - Set ABORT (bit 28) to 0.
   - Clear PF (bit 30). Hardware will set this bit if a failure is detected.
   - If desired to halt whenever there is an error, set HALT (bit 29).
   - Set START (bit 31) to 1 to start MemBIST execution.
5. Check the MemBIST results, and if a failure occurred, observe failure data or address through MBDATA registers and MB_ERR_DATA registers:
   - Check MBCSR:start. 0 means MemBIST has completed. Check MBCSR:PF. 1 means a failure has occurred.
   - Depending upon the value chosen for MBCSR:mbdata, either up to 5 failure addresses or the failure data bit location accumulator will be stored in MBDATA[8, 3:0].
   - Failure data will be stored in MB_ERR_DATA[3:0][4:0] registers.

11.4.3 Write Leaping 0s to the Full DRAM Address Range and Check

This describes writing leaping 0s to all locations in the DRAM on the DIMM and checking this data. Leaping 0s are like walking 0s except that rather than moving from bit to adjacent bit, the single 0 in a field of 1s moves from one bit to another bit far removed from it. Of every 160 writes to the DRAMs, 144 will have a 0 on some bit, and 16 will have no 0s on any bit. Refer to the tables and block diagram above to see how this works.

1. Set up registers for normal DRAM operation.

2. This example illustrates the use of MBCSR:atype = 11, so MB_START_ADDR and MB_END_ADDR are not programmed. However, these registers could alternatively be set to 0 and the maximum address in the DIMM respectively, and MBCRS:atype set to 10, to accomplish the same effect.
3. To get a field of 1s for the leaping 0 to traverse, data to the DRAMs must be inverted. This is because MBDATA[9, 7:4] are set to zeros at the start of MemBIST execution. That also means that the data programmed into MBLFSRSED must be inverted. So MBLFSRSED is set to 0x0000_0001. When inverted, this will give a single 0 in a field of 1s. Registers MBDATA[8, 3:0] are unused and are therefore not written.
4. Program MBCSR.

   These fields are required for the specified test:

   — Program DTYPE (bits [9:8]) to 10 to select circular shifted data.

   — Program CMD (bits [5:4]) to be 11 (write followed by read with data comparison).

   — Program ATYPE (bits [7:6]) to 11 to use the full address range of the DIMMs as already defined in the MTR register.

   — Select either Rank 0 or Rank 1 by programming CS (bits [21:20]).

   — INVERT (bit 19) must be set to 1 to create the field of 1s from the 0s in MBDATA[9, 7:4].

   — ENABLE288 (bit 15) is not relevant when circular shifted data has been selected. Rewrite this field with its default value of 0.

   The values for these fields can be selected to choose options for use during MemBIST operation:

   — Program ABAR (bit 13) to select DAI if desired.

   — Program FAST (bits [11:10]) to select Fast X, Fast Y, Fast XY, or XZY (column->bank->row) address sequencing.

   — Program ADIR (bit 12) to select whether addresses should increment or decrement.

   — MBDATA (bit 14) is set to select failure address logging or failure data bit location accumulator logging in MBDATA.

   Set these control values to start the MemBIST engine.

   — Set ALGO (bits 26:24) to 0 to prevent the algorithm engine from overwriting bits it controls in MBCSR.

   — Set ABORT (bit 28) to 0.

   — Clear PF (bit 30). Hardware will set this bit if a failure is detected.

   — If desired to halt whenever there is an error, set HALT (bit 29).

   — Set START (bit 31) to 1 to start MemBIST execution.
5. Check the MemBIST results, and if a failure occurred, observe failure data or address through MBDATA registers and MB_ERR_DATA registers:

   — Check MBCSR:start. 0 means MemBIST has completed. Check MBCSR:PF. 1 means a failure has occurred.

   — Depending upon the value chosen for MBCSR:mbdata, either up to 5 failure addresses or the failure data bit location accumulator will be stored in MBDATA[8, 3:0].

   — Failure data will be stored in the MB_ERR_DATA[3:0][4:0] registers.

11.4.4 Write 144-bit User-defined Pattern to a Range of Addresses and Check

This describes writing user-defined data to a range of DRAM addresses and checking this data.

1. Set up registers for normal DRAM operation.
2. Program MB_START_ADDR and MB_END_ADDR registers to the desired address range to be tested.
3. 144 bits of user-defined data is written to MBDATA[9, 7:4]. MBDATA[8, 3:0] and MBLFSRSED are unused and are therefore not written.
4. Program MBCSR.
   These fields are required for the specified test:
   - Program DTYPE (bits [9:8]) to 01 to select user-defined data.
   - ENABLE288 (bit 15) is left 0 to select 144 bit user-defined data.
   - Program CMD (bits [5:4]) to be 11 (write followed by read with data comparison).
   - Program ATYPE (bits [7:6]) to 10 to use the address range already defined in the MB_START_ADDR and MB_END_ADDR registers.
   - Select either Rank 0 or Rank 1 by programming CS (bits [21:20]).
   - INVERT (bit 19) is unnecessary for user-defined data, as the user can set the bits to the values desired directly. Set this field to 0.

   The values for these fields can be selected to choose options for use during MemBIST operation:
   - Program ABAR (bit 13) to select DAI if desired.
   - Program FAST (bits [11:10]) to select Fast X, Fast Y, Fast XY, or XZY (column->bank->row) address sequencing.
   - Program ADIR (bit 12) to select whether addresses should increment or decrement.
   - MBDATA (bit 14) is set to select failure address logging or failure data bit location accumulator logging in MBDATA[8, 3:0].

   Set these control values to start the MemBIST engine.
   - Set ALGO (bits 26:24) to 0 to prevent the algorithm engine from overwriting bits it controls in MBCSR.
   - Set ABORT (bit 28) to 0.
   - Clear PF (bit 30). Hardware will set this bit if a failure is detected.
   - If desired to halt whenever there is an error, set HALT (bit 29).
   - Set START (bit 31) to 1 to start MemBIST execution.
5. Check the MemBIST results, and if a failure occurred, observe failure data or address through MBDATA registers and MB_ERR_DATA registers:
   - Check MBCSR:start. 0 means MemBIST has completed. Check MBCSR:PF. 1 means a failure has occurred.
   - Depending upon the value chosen for MBCSR:mbdata, either up to 5 failure addresses or the failure data bit location accumulator will be stored in MBDATA[8, 3:0].
   - Failure data will be stored in MB_ERR_DATA[3:0][4:0] registers.

11.4.5 Test a Range of DRAM Addresses With March C- Algorithm

The following points describe the programming required to test all of the addresses within a selected address range using the March C- algorithm provided by MemBIST.

1. Set up registers for normal DRAM operation.
2. Program MB_START_ADDR and MB_END_ADDR registers to the desired address range to be tested. If the full address of the DIMM is to be tested, it must be programmed here.
3. No user data is required, as all built-in testing algorithms use the fixed data pattern 0xA (regardless of the pattern selected by MBCSR:fixed). Therefore, the MBDATA[9:0] and MBLFSRSED registers are not written.
4. Program MBCSR.

   These fields are required for the specified test:

   — Set ALGO (bits 26:24) to 110 to select the built-in March C- testing algorithm.

   — Program CMD (bits [5:4]) to be 01 (write only without data comparison), as this is required for the March C- algorithm.

   The values for these fields can be selected to choose options for use during MemBIST operation:

   — Select either Rank 0 or Rank 1 by programming CS (bits [21:20]).

   — Program FAST (bits [11:10]) to select Fast X, Fast Y, Fast XY, or XZY (column->bank->row) address sequencing.

   These bits are taken over by the algorithm engine to execute the algorithm. Set them to their default values.

   — FIXED (bits [18:16]).

   — ADIR (bit 12).

   — ABAR (bit 13).

   — DTYPE (bits [9:8]).

   — ATYPE (bits [7:6]).

   — ENABLE288 (bit 15)

   — INVERT (bit 19).

   — MBDATA (bit 14). This bit is irrelevant because algorithms always use fixed data, so both failure data bit lane accumulator and failure addresses are logged in MBDATA.

   Set these control values to start the MemBIST engine.

   — Set ABORT (bit 28) to 0.

   — Clear PF (bit 30). Hardware will set this bit if a failure is detected.

   — If desired to halt whenever there is an error, set HALT (bit 29).

   — Set START (bit 31) to 1 to start MemBIST execution.
5. Check the MemBIST results, and if checking was enabled, observe failure data or address through MBDATA registers and MB_ERR_DATA registers:

   — Check MBCSR:start. 0 means MemBIST has completed. Check MBCSR:PF. 1 means a failure has occurred.

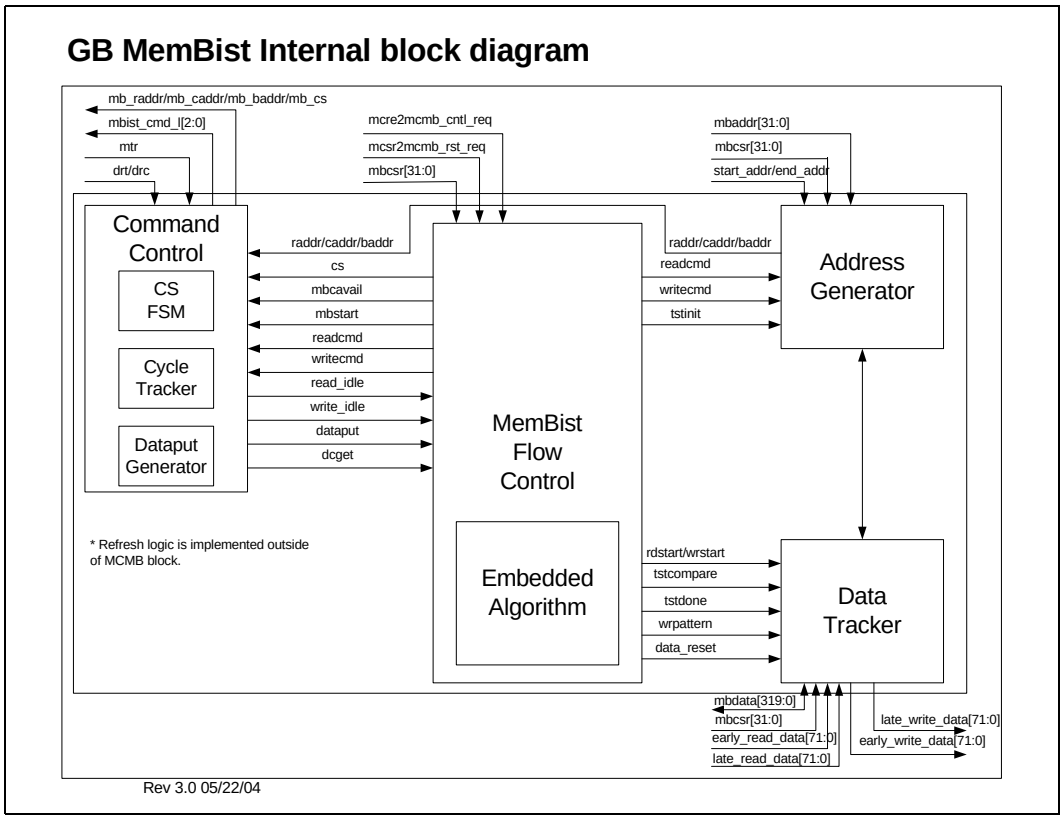
   — Failure data bit location accumulator will be stored in MBDATA[8, 3:0]. Up to 5 failure addresses will be placed in MBDATA[9, 7:4]. Use the “Fixed Data Pattern” mapping for these registers.

   — Failure data will be stored in MB_ERR_DATA[3:0][4:0] registers.

11.5 MemBIST Implementation

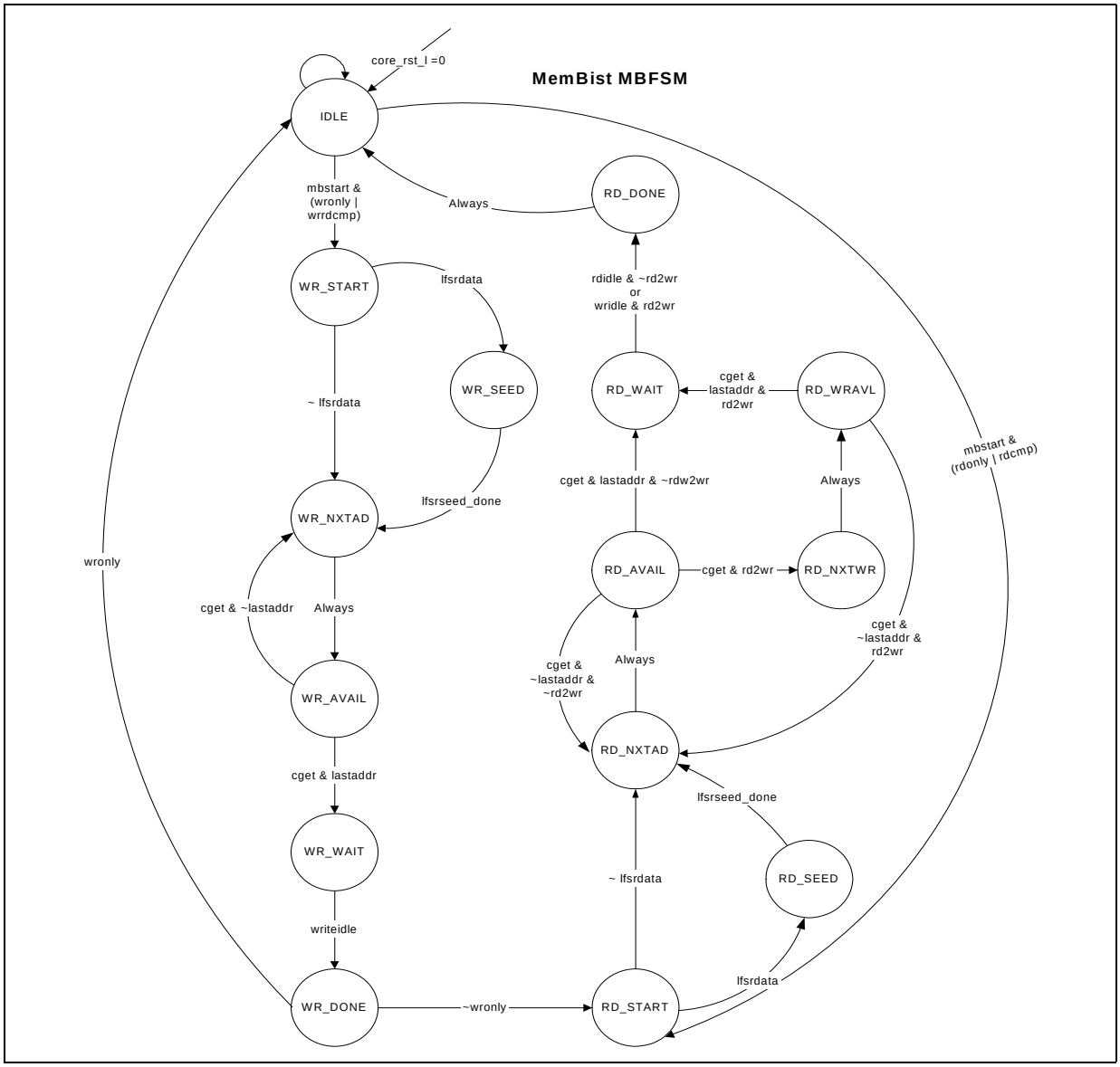
11.5.1 MemBIST Block Diagram

Figure 11-6. MemBIST Block Diagram



11.5.2 MB Flow Control State Machine

Figure 11-7. MBFSM Diagram



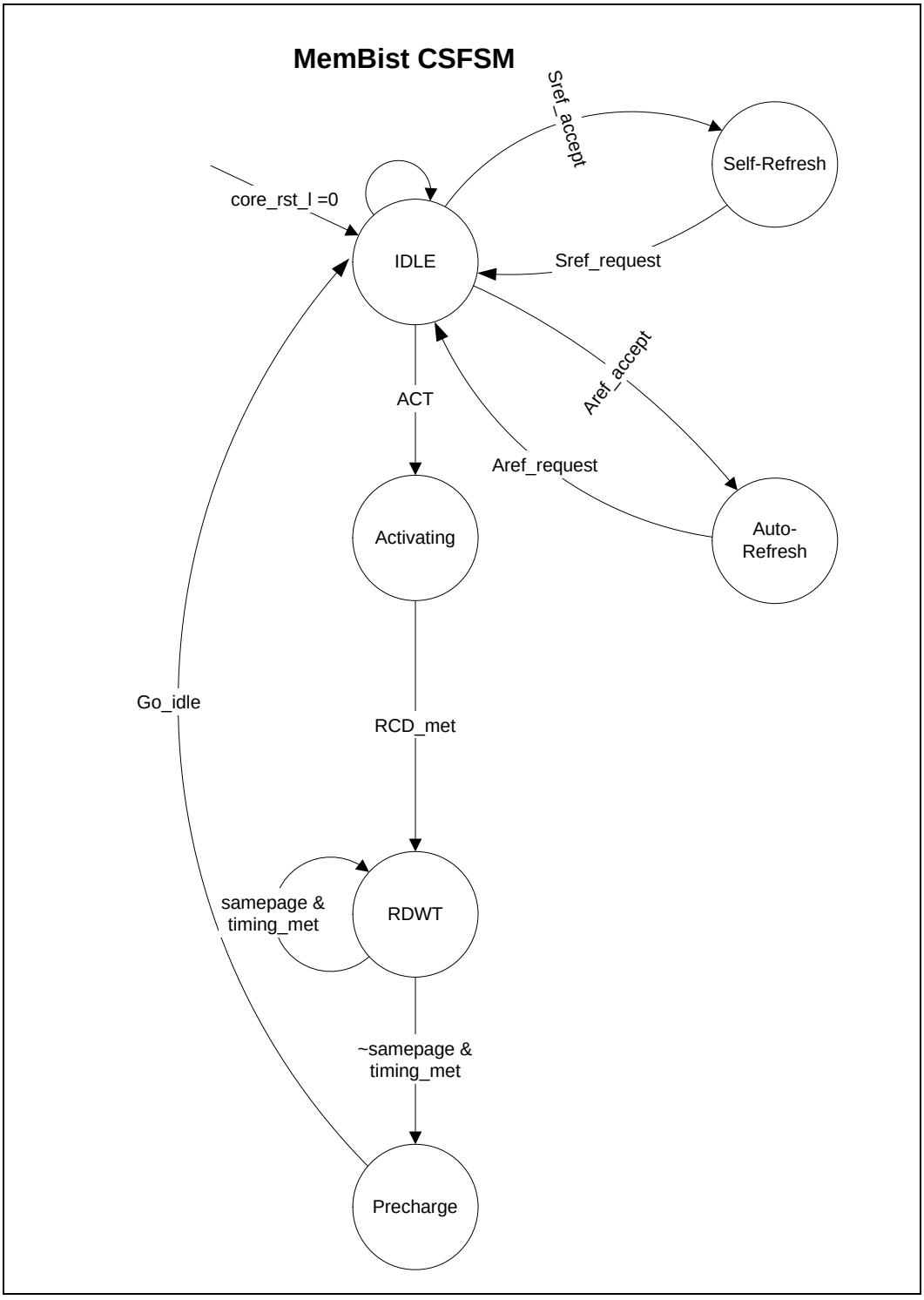
This FSM controls the MemBIST flow and generates read/write commands for MemBIST.

- When MBCSR bit[31] is programmed to begin execution, the MemBIST FSM will transition out of the IDLE state to either WR_START or RD_START, depending upon the MemBIST command programmed in MBCSR:cmd.
- In WR_START state, FSM will look at the decoding of DATA type selection. If LFSR data type generation is selected, FSM will go to WR_SEED state. If not, FSM will directly go to WR_NXTAD state.

- In WR_SEED state, MemBIST will create 5 crc32 data sets and load into MBDATA4/5/6/7/9 from the initial seed register MBLFSRSED. When 5 sets of random data are loaded into MBDATA, the FSM will transition out of this state to WR_NXTAD state.
- In the WR_NXTAD state, the next address for the operation is calculated and the address issued. The FSM then transitions to the WR_AVAIL state immediately.
- The FSM will alternate between WR_NXTAD and WR_AVAIL until all writes are issued. In the WR_AVAIL state, a write command will be issued. Once the previous cycle's DRAM timing requirements are met (indicated by cget true), the FSM leaves the WR_AVAIL state. If the write command issued was not to the last address, the FSM will transition to WR_NXTAD. If this was the last address, the FSM will go to WR_WAIT state.
- In the WR_WAIT state, the FSM will wait for the timing for the last write to be met, and then will transition to the WR_DONE state.
- If this is a write only operation, then the FSM will go back to the IDLE state. If this is a write with read comparison test, the FSM will go to the RD_START state.
- In RD_START state, FSM will look at the decoding of DATA type selection. If LFSR data type generation is selected, FSM will go to RD_SEED state. If not, FSM will directly go to RD_NXTAD state.
- In RD_SEED state, MemBIST will create 5 crc32 data sets and load into MBDATA4/5/6/7/9 from the initial seed register MBLFSRSED. When 5 sets of random data is set and loaded into MBDATA, FSM will transit out of this state to RD_NXTAD state.
- In the RD_NXTAD state, the next address for the operation is calculated and the address issued. The FSM transitions to the RD_AVAIL state immediately.
- The FSM will alternate between RD_NXTAD and RD_AVAIL until all reads are issued. In the RD_AVAIL state, a read command will be issued. Once the previous cycle's DRAM timing requirements are met (indicated by cget true), the FSM leaves the RD_AVAIL state. If the read command issued was not to the last address, the FSM will transition to RD_NXTAD. If this was the last address, the FSM will go to RD_WAIT state. If this is a back to back read/write operation, the FSM will transit from the RD_AVAIL state to the RD_NXTWR state.
- In the RD_NXTWR state, the address for the write command will be issued (which is the same as the read address previously used). The FSM then transitions to the RD_WRAVL state immediately.
- From the RD_WRAVL state, if this is the last address, the FSM will go to the RD_WAIT state after meeting the DRAM timing requirements. If this is not the last address, the FSM will go back to the RD_NXTAD state again after meeting the DRAM timing requirements.
- In the RD_WAIT state, the FSM will wait for the timing requirements for the last read or write to be met, and then transition to the RD_DONE state.
- From the RD_DONE state the FSM will always go to the IDLE state.

11.5.3 CS Finite State Machine

Figure 11-8. CS State Machine



11.5.3.1 MemBIST CSFSM

This FSM creates DRAM commands for MemBIST.

- When read or write command is available and tRP/tRC timing are met, MemBIST CSFSM will transit out of IDLE state to ACT state.
- In ACT state, FSM will wait for tRCD timing parameter qualified and go to RDWT state.
- If the next coming read or write command is in same page and DRAM timing is qualified, FSM will be looping in this state. If the next command is not in the same page or there is an auto-refresh/self-refresh request, FSM will go to PRECHARGE state.
- In PRECHARGE state, FSM always go back to IDLE state.
- In IDLE state, if there is a self refresh or an auto refresh request, FSM will wait for self refresh logic accept signal or auto refresh accept signal.

§

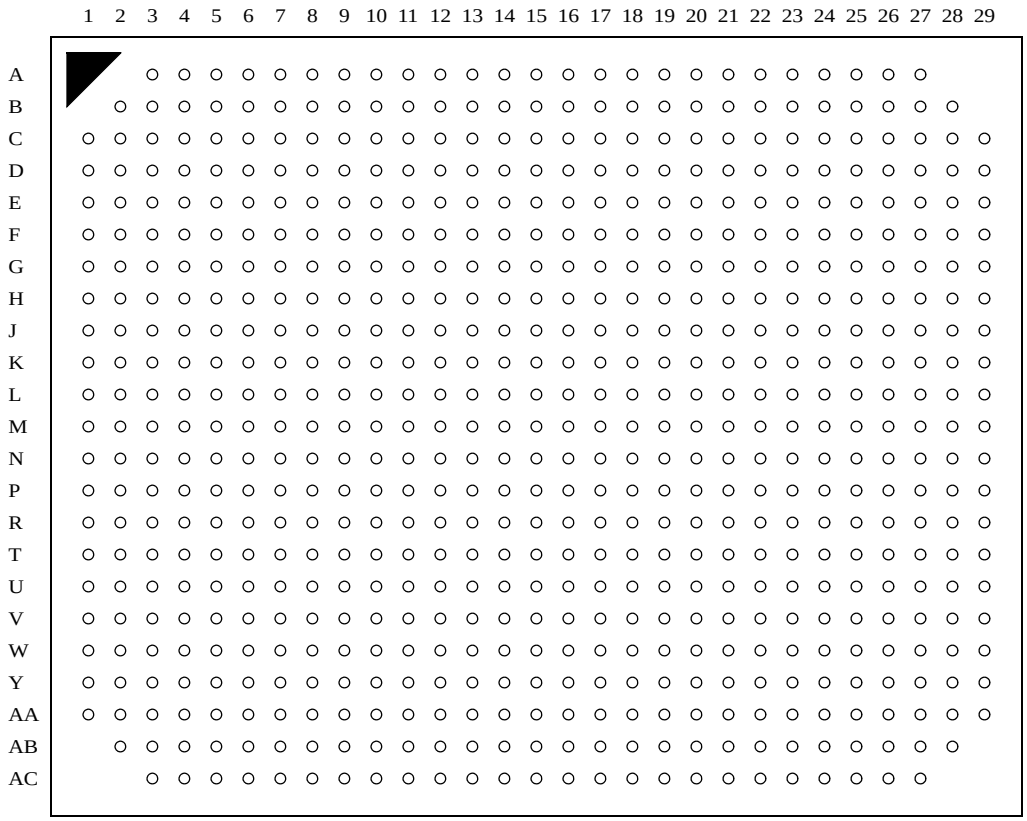
12 Ballout and Package Information

12.1 Ballout

The following section presents preliminary ballout information for the Intel 6400/6402 Advanced Memory Buffer (AMB). This ballout is subject to change and is to be used for informational purposes only.

12.2 655-Ball FBGA 0.8mm Pitch Pin Configuration

Figure 12-1. Pinout Configuration



12.3 Pin Assignments for the Advanced Memory Buffer (AMB)

Table 12-1. 655-Ball FBGA 0.8 mm Pitch - Left Side

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A			V_{SS}	DQ26	DQ12	V_{DD}	DQS10	DQ13	V_{DD}	$\overline{DQS1}$	DQ10	V_{DD}	TESTLO	V_{DD}	V_{DD}
B		V_{DD}	DQS3	$\overline{DQS3}$	V_{SS}	DQ14	$\overline{DQS10}$	V_{SS}	DQ11	DQS1	V_{SS}	DDRC	TESTLO	V_{DD}	V_{SS}
C	V_{SS}	DQS2	DQ18	V_{SS}	DQ4	$\overline{DQS9}$	V_{SS}	DQ15	DQ9	V_{SS}	DQ8	DDRC	V_{SS}	DDRC	DQS17
D	DQ19	$\overline{DQS2}$	V_{SS}	DQ16	DQ24	V_{SS}	DQS9	DQ7	V_{SS}	DQ3	DQS0	V_{SS}	$\overline{DQS8}$	DQS8	V_{DD}
E	DQ21	V_{SS}	DQ17	DQ29	V_{SS}	DQ25	DQ6	V_{SS}	DQ5	DQ1	V_{SS}	DQ0	CB1	V_{SS}	CB2
F	V_{SS}	DQ20	DQ23	V_{SS}	DQ31	DQ27	V_{SS}	TESTLO	TEST	V_{SS}	$\overline{DQS0}$	DQ2	V_{DD}	CB0	CB3
G	$\overline{DQS11}$	DQS11	NC	NC	NC	V_{SS}	DQS12	$\overline{DQS12}$	NC	NC	NC	BFUNC	RFU	RFU	RFU
H	DQ22	V_{SS}	NC	NC	NC	DQ28	DQ30	V_{SS}	NC	NC	NC	V_{SS}	V_{DD}	V_{SS}	V_{DD}
J	V_{SS}	CLK2	NC	NC	NC	BA1A	V_{SS}	CKE1A	NC	NC	NC	V_{DD}	V_{SS}	V_{DD}	V_{SS}
K	$\overline{CLK2}$	CLK0	NC	NC	NC	V_{SS}	$\overline{WEA}$	$\overline{RASA}$	NC	NC	NC	V_{SS}	V_{CC}	V_{SS}	V_{CC}
L	$\overline{CLK0}$	V_{SS}	NC	NC	NC	A0A	CKE0A	V_{SS}	NC	NC	NC	V_{CC}	V_{SS}	V_{CC}	V_{SS}
M	ODT0A	RFU	NC	NC	NC	$\overline{CASA}$	V_{SS}	BA2A	NC	NC	NC	V_{SS}	V_{CC}	V_{SS}	V_{CC}
N	$\overline{CS1A}$	$\overline{CS0A}$	NC	NC	NC	V_{SS}	BA0A	A10A	NC	NC	NC	V_{CC}	V_{SS}	V_{CC}	V_{SS}
P	A6A	V_{SS}	NC	NC	NC	A2A	A1A	A3A	NC	NC	NC	V_{SS}	V_{CC}	V_{SS}	V_{CC}
R	V_{SS}	A8A	NC	NC	NC	A11A	V_{SS}	A5A	NC	NC	NC	V_{CC}	V_{SS}	V_{CC}	V_{SS}
T	A4A	A13A	NC	NC	NC	V_{SS}	A9A	A7A	NC	NC	NC	V_{SS}	V_{CC}	V_{SS}	V_{CC}
U	PN0	$\overline{PN0}$	NC	NC	NC	A15A	A14A	A12A	NC	NC	NC	RFU	V_{CCFBD}	V_{SS}	V_{SS}
V	PN1	$\overline{PN1}$	V_{SS}	SN0	$\overline{SN0}$	V_{CCFBD}	V_{SS}	V_{CCFBD}	V_{SS}	RFU[a]	RFU[a]	V_{CCFBD}	V_{SS}	V_{SS}	V_{SS}
W	PN2	$\overline{PN2}$	V_{SS}	$\overline{SN1}$	$\overline{SN1}$	$\overline{SN3}$	$\overline{SN4}$	$\overline{SN5}$	$\overline{SN13}$	$\overline{SN12}$	$\overline{SN6}$	$\overline{SN7}$	$\overline{SN8}$	$\overline{SN9}$	$\overline{SN10}$
Y	PN3	$\overline{PN3}$	V_{SS}	SN2	SN2	SN3	SN4	SN5	SN13	SN12	SN6	SN7	SN8	SN9	SN10
AA	V_{SS}	PN4	$\overline{PN4}$	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
AB		V_{SS}	$\overline{RESET}$	$\overline{PN5}$	$\overline{PN13}$	RFU[a]	$\overline{PN12}$	$\overline{PN6}$	$\overline{PN7}$	$\overline{PN8}$	$\overline{PN9}$	V_{SSAPLL}	V_{CCAPLL}	$\overline{PN10}$	$\overline{PN11}$
AC			V_{SS}	PN5	PN13	RFU[a]	PN12	PN6	PN7	PN8	PN9	FBDRES	PLLTSTO	PN10	PN11

These pin positions are reserved for forward clocks to be used in future AMB implementations.

Table 12-2. 655-Ball FBGA 0.8 mm Pitch - Right Side

	16	17	18	19	20	21	22	23	24	25	26	27	28	29
A	V_{DD}	TEST	V_{DD}	DQ52	DQS15	V_{DD}	DQ49	$\overline{DQS6}$	V_{DD}	DQ48	DQ38	V_{DD}		
B	V_{DD}	TEST	DDRC	V_{SS}	DQS15	DQ53	V_{SS}	DQS6	DQ50	V_{SS}	DQS13	DQS13	V_{SS}	
C	$\overline{DQS17}$	V_{SS}	DDRC	DQ54	V_{SS}	DQ55	DQ51	V_{SS}	DQS7	DQ56	V_{SS}	DQ46	$\overline{DQS14}$	V_{DD}
D	CB6	CB7	V_{SS}	DQS16	DQ63	V_{SS}	DQ59	$\overline{DQS7}$	V_{SS}	DQ36	DQ44	V_{SS}	DQS14	DQ47
E	V_{SS}	CB5	$\overline{DQS16}$	V_{SS}	DQ61	DQ57	V_{SS}	DQ58	DQ39	V_{SS}	DQ33	DQ45	V_{SS}	DQ41
F	CB4	V_{DD}	DQ62	DQ60	V_{SS}	TEST	TEST	V_{SS}	DQ37	DQ35	V_{SS}	$\overline{DQS5}$	DQ43	V_{SS}
G	TEST LO	RFU	RFU	NC	NC	NC	DQS4	$\overline{DQS4}$	V_{SS}	NC	NC	NC	DQS5	DQ40
H	V_{SS}	V_{DD}	V_{SS}	NC	NC	NC	V_{SS}	DQ34	DQ32	NC	NC	NC	V_{SS}	DQ42
J	V_{DD}	V_{SS}	V_{DD}	NC	NC	NC	$\overline{RASB}$	V_{SS}	RFU	NC	NC	NC	$\overline{CLK3}$	V_{SS}
K	V_{SS}	V_{CC}	V_{SS}	NC	NC	NC	ODT0B	CS1B	V_{SS}	NC	NC	NC	$\overline{CLK1}$	CLK3
L	V_{CC}	V_{SS}	V_{CC}	NC	NC	NC	V_{SS}	$\overline{CASB}$	$\overline{WEB}$	NC	NC	NC	V_{SS}	CLK1
M	V_{SS}	V_{CC}	V_{SS}	NC	NC	NC	$\overline{CS0B}$	V_{SS}	BA1B	NC	NC	NC	CKE0B	V_{SS}
N	V_{CC}	V_{SS}	V_{CC}	NC	NC	NC	A0B	A2B	V_{SS}	NC	NC	NC	BA0B	BA2B
P	V_{SS}	V_{CC}	V_{SS}	NC	NC	NC	V_{SS}	A4B	A1B	NC	NC	NC	V_{SS}	CKE1B
R	V_{CC}	V_{SS}	V_{CC}	NC	NC	NC	A6B	V_{SS}	A10B	NC	NC	NC	A3B	V_{SS}
T	V_{SS}	V_{CC}	V_{SS}	NC	NC	NC	A11B	A9B	V_{SS}	NC	NC	NC	A7B	A5B
U	V_{SS}	V_{CCFBD}	RFU	NC	NC	NC	A8B	A15B	A14B	SA0	SCL	SDA	$\overline{PS8}$	PS8
V	V_{CCFBD}	V_{SS}	V_{CCFBD}	V_{SS}	V_{CCFBD}	RFU[a]	RFU[a]	V_{SS}	A13B	A12 B	SA2	SA1	$\overline{PS7}$	PS7
W	V_{SS}	$\overline{SS0}$	$\overline{SS1}$	$\overline{SS2}$	$\overline{SS3}$	$\overline{SS4}$	$\overline{SS9}$	$\overline{SS5}$	$\overline{SS6}$	$\overline{SS7}$	$\overline{SS8}$	V_{SS}	$\overline{PS6}$	PS6
Y	V_{SS}	SS0	SS1	SS2	SS3	SS4	SS9	SS5	SS6	SS7	SS8	V_{SS}	$\overline{PS5}$	PS5
AA	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	$\overline{PS9}$	PS9	V_{SS}
AB	V_{SS}	$\overline{SN11}$	V_{SS}	SCK	TESTLO_AB20	$\overline{PS0}$	$\overline{PS1}$	$\overline{PS2}$	$\overline{PS3}$	$\overline{PS4}$	RFU[a]	V_{DDSPD}	V_{SS}	
AC	RFU	SN11	V_{SS}	$\overline{SCK}$	TESTLO_AC20	PS0	PS1	PS2	PS3	PS4	RFU[a]	V_{SS}		

These pin positions are reserved for forward clocks to be used in future AMB implementations.

Table 12-3. Advanced Memory Buffer Signals By Ball Number (Sheet 1 of 7)

Ball No.	Signal
A3	VSS
A4	DQ26
A5	DQ12
A6	VDD
A7	DQS10
A8	DQ13
A9	VDD

Ball No.	Signal
B15	VSS
B16	VDD
B17	TESTLO
B18	DDRC_B18
B19	VSS
B20	$\overline{DQS15}$
B21	DQ53

Ball No.	Signal
C25	DQ56
C26	VSS
C27	DQ46
C28	$\overline{DQS14}$
C29	VDD
D1	DQ19
D2	DQS2

Table 12-3. Advanced Memory Buffer Signals By Ball Number (Sheet 2 of 7)

Ball No.	Signal
A10	$\overline{\text{DDQS1}}$
A11	DQ10
A12	VDD
A13	TESTLO
A14	VDD
A15	VDD
A16	VDD
A17	TEST
A18	VDD
A19	DQ52
A20	DQS15
A21	VDD
A22	DQ49
A23	DQS6
A24	VDD
A25	DQ48
A26	DQ38
A27	VDD
B2	VDD
B3	DQS3
B4	DQS3
B5	VSS
B6	DQ14
B7	$\overline{\text{DQS10}}$
B8	VSS
B9	DQ11
B10	DQS1
B11	VSS
B12	DDRC_B12
B13	TESTLO
B14	VDD
E5	VSS
E6	DQ25
E7	DQ6
E8	VSS
E9	DQ5
E10	DQ1
E11	VSS
E12	DQ0
E13	CB1
E14	VSS
E15	CB2

Ball No.	Signal
B22	VSS
B23	DQS6
B24	DQ50
B25	VSS
B26	DQS13
B27	DQS13
B28	VSS
C1	VSS
C2	DQS2
C3	DQ18
C4	VSS
C5	DQ4
C6	DQS9
C7	VSS
C8	DQ15
C9	DQ9
C10	VSS
C11	DQ8
C12	DDRC_C12
C13	VSS
C14	DDRC_C14
C15	DQS17
C16	DQS17
C17	VSS
C18	DDRC_C18
C19	DQ54
C20	VSS
C21	DQ55
C22	DQ51
C23	VSS
C24	DQS7
F14	CB0
F15	CB3
F16	CB4
F17	VDD
F18	DQ62
F19	DQ60
F20	VSS
F21	TEST
F22	TEST
F23	VSS
F24	DQ37

Ball No.	Signal
D3	VSS
D4	DQ16
D5	DQ24
D6	VSS
D7	DQS9
D8	DQ7
D9	VSS
D10	DQ3
D11	DQS0
D12	VSS
D13	DQS8
D14	DQS8
D15	VDD
D16	CB6
D17	CB7
D18	VSS
D19	DQS16
D20	DQ63
D21	VSS
D22	DQ59
D23	$\overline{\text{DQS7}}$
D24	VSS
D25	DQ36
D26	DQ44
D27	VSS
D28	DQS14
D29	DQ47
E1	DQ21
E2	VSS
E3	DQ17
E4	DQ29
G23	DQS4
G24	VSS
G25	NC
G26	NC
G27	NC
G28	DQS5
G29	DQ40
H1	DQ22
H2	VSS
H3	NC
H4	NC

Table 12-3. Advanced Memory Buffer Signals By Ball Number (Sheet 3 of 7)

Ball No.	Signal
E16	VSS
E17	CB5
E18	$\overline{\text{DQS16}}$
E19	VSS
E20	DQ61
E21	DQ57
E22	VSS
E23	DQ58
E24	DQ39
E25	VSS
E26	DQ33
E27	DQ45
E28	VSS
E29	DQ41
F1	VSS
F2	DQ20
F3	DQ23
F4	VSS
F5	DQ31
F6	DQ27
F7	VSS
F8	TESTLO
F9	TEST
F10	VSS
F11	DQS0 -> $\overline{\text{DQS0}}$
F12	DQ2
F13	VDD
J3	NC
J4	NC
J5	NC
J6	BA1A
J7	VSS
J8	CKE1A
J9	NC
J10	NC
J11	NC
J12	VDD
J13	VSS
J14	VDD
J15	VSS
J16	VDD
J17	VSS

Ball No.	Signal
F25	DQ35
F26	VSS
F27	$\overline{\text{DQS5}}$
F28	DQ43
F29	VSS
G1	$\overline{\text{DQS11}}$
G2	DQS11
G3	NC
G4	NC
G5	NC
G6	VSS
G7	DQS12
G8	DQS12
G9	NC
G10	NC
G11	NC
G12	BFUNC
G13	RFU
G14	RFU
G15	RFU
G16	TESTLO
G17	RFU
G18	RFU
G19	NC
G20	NC
G21	NC
G22	DQS4
K12	VSS
K13	VCC
K14	VSS
K15	VCC
K16	VSS
K17	VCC
K18	VSS
K19	NC
K20	NC
K21	NC
K22	ODT0B
K23	$\overline{\text{CS}}$1B
K24	VSS
K25	NC
K26	NC

Ball No.	Signal
H5	NC
H6	DQ28
H7	DQ30
H8	VSS
H9	NC
H10	NC
H11	NC
H12	VSS
H13	VDD
H14	VSS
H15	VDD
H16	VSS
H17	VDD
H18	VSS
H19	NC
H20	NC
H21	NC
H22	VSS
H23	DQ34
H24	DQ32
H25	NC
H26	NC
H27	NC
H28	VSS
H29	DQ42
J1	VSS
J2	CLK2
L21	NC
L22	VSS
L23	$\overline{\text{CASB}}$
L24	$\overline{\text{WEB}}$
L25	NC
L26	NC
L27	NC
L28	VSS
L29	CLK1
M1	ODT0A
M2	RFU
M3	NC
M4	NC
M5	NC
M6	$\overline{\text{CASA}}$

Table 12-3. Advanced Memory Buffer Signals By Ball Number (Sheet 4 of 7)

Ball No.	Signal
J18	VDD
J19	NC
J20	NC
J21	NC
J22	$\overline{\text{RASB}}$
J23	VSS
J24	RFU
J25	NC
J26	NC
J27	NC
J28	$\overline{\text{CLK3}}$
J29	VSS
K1	CLK2
K2	CLK0
K3	NC
K4	NC
K5	NC
K6	VSS
K7	WEA
K8	RASA
K9	NC
K10	NC
K11	NC
N1	$\overline{\text{CS1A}}$
N2	$\overline{\text{CS0A}}$
N3	NC
N4	NC
N5	NC
N6	VSS
N7	BA0A
N8	A10A
N9	NC
N10	NC
N11	NC
N12	VCC
N13	VSS
N14	VCC
N15	VSS
N16	VCC
N17	VSS
N18	VCC
N19	NC

Ball No.	Signal
K27	NC
K28	$\overline{\text{CLK1}}$
K29	CLK3
L1	$\overline{\text{CLK0}}$
L2	VSS
L3	NC
L4	NC
L5	NC
L6	A0A
L7	CKE0A
L8	VSS
L9	NC
L10	NC
L11	NC
L12	VCC
L13	VSS
L14	VCC
L15	VSS
L16	VCC
L17	VSS
L18	VCC
L19	NC
L20	NC
P10	NC
P11	NC
P12	VSS
P13	VCC
P14	VSS
P15	VCC
P16	VSS
P17	VCC
P18	VSS
P19	NC
P20	NC
P21	NC
P22	VSS
P23	A4B
P24	A1B
P25	NC
P26	NC
P27	NC
P28	VSS

Ball No.	Signal
M7	VSS
M8	BA2A
M9	NC
M10	NC
M11	NC
M12	VSS
M13	VCC
M14	VSS
M15	VCC
M16	VSS
M17	VCC
M18	VSS
M19	NC
M20	NC
M21	NC
M22	$\overline{\text{CS0B}}$
M23	VSS
M24	BA1B
M25	NC
M26	NC
M27	NC
M28	CKE0B
M29	VSS
R19	NC
R20	NC
R21	NC
R22	A6B
R23	VSS
R24	A10B
R25	NC
R26	NC
R27	NC
R28	A3B
R29	VSS
T1	A4A
T2	A13A
T3	NC
T4	NC
T5	NC
T6	VSS
T7	A9A
T8	A7A

Table 12-3. Advanced Memory Buffer Signals By Ball Number (Sheet 5 of 7)

Ball No.	Signal
N20	NC
N21	NC
N22	A0B
N23	A2B
N24	VSS
N25	NC
N26	NC
N27	NC
N28	BA0B
N29	BA2B
P1	A6A
P2	VSS
P3	NC
P4	NC
P5	NC
P6	A2A
P7	A1A
P8	A3A
P9	NC
T28	A7B
T29	A5B
U1	PN0
U2	PN0
U3	NC
U4	NC
U5	NC
U6	A15A
U7	A14A
U8	A12A
U9	NC
U10	NC
U11	NC
U12	RFU
U13	VCCFBD
U14	VSS
U15	VSS
U16	VSS
U17	VCCFBD
U18	RFU
U19	NC
U20	NC
U21	NC

Ball No.	Signal
P29	CKE1B
R1	VSS
R2	A8A
R3	NC
R4	NC
R5	NC
R6	A11A
R7	VSS
R8	A5A
R9	NC
R10	NC
R11	NC
R12	VCC
R13	VSS
R14	VCC
R15	VSS
R16	VCC
R17	VSS
R18	VCC
V8	VCCFBD
V9	VSS
V10	RFU[a]
V11	RFU[a]
V12	VCCFBD
V13	VSS
V14	VSS
V15	VSS
V16	VCCFBD
V17	VSS
V18	VCCFBD
V19	VSS
V20	VCCFBD
V21	RFU[a]
V22	RFU[a]
V23	VSS
V24	A13B
V25	A12B
V26	SA2
V27	SA1
V28	PS7
V29	PS7
W1	PN2

Ball No.	Signal
T9	NC
T10	NC
T11	NC
T12	VSS
T13	VCC
T14	VSS
T15	VCC
T16	VSS
T17	VCC
T18	VSS
T19	NC
T20	NC
T21	NC
T22	A11B
T23	A9B
T24	VSS
T25	NC
T26	NC
T27	NC
W17	$\overline{SS0}$
W18	SS1 -> $\overline{SS1}$
W19	SS2
W20	$\overline{SS3}$
W21	$\overline{SS4}$
W22	SS9
W23	$\overline{SS5}$
W24	$\overline{SS6}$
W25	$\overline{SS7}$
W26	$\overline{SS8}$
W27	VSS
W28	$\overline{PS6}$
W29	PS6
Y1	PN3
Y2	PN3
Y3	VSS
Y4	SN2
Y5	SN2
Y6	SN3
Y7	SN4
Y8	SN5
Y9	SN13
Y10	SN12

Table 12-3. Advanced Memory Buffer Signals By Ball Number (Sheet 6 of 7)

Ball No.	Signal	Ball No.	Signal	Ball No.	Signal
U22	A8B	W2	$\overline{PN2}$	Y11	SN6
U23	A15B	W3	VSS	Y12	SN7
U24	A14B	W4	SN1	Y13	SN8
U25	SA0	W5	$\overline{SN1}$	Y14	SN9
U26	SCL	W6	$\overline{SN3}$	Y15	SN10
U27	SDA	W7	$\overline{SN4}$	Y16	VSS
U28	PS8	W8	$\overline{SN5}$	Y17	SS0
U29	PS8	W9	SN13	Y18	SS1
V1	PN1	W10	SN12	Y19	SS2
V2	PN1	W11	$\overline{SN6}$	Y20	SS3
V3	VSS	W12	SN7	Y21	SS4
V4	SN0	W13	SN8	Y22	SS9
V5	$\overline{SN0}$	W14	$\overline{SN9}$	Y23	SS5
V6	VCCFBD	W15	$\overline{SN10}$	Y24	SS6
V7	VSS	W16	VSS	Y25	SS7
These pin positions are reserved for forward clocks to be used in future AMB implementations.					
Y26	SS8	AB7	PN12	AC19	SCK
Y27	VSS	AB8	PN6	AC20	TESTLO_AC20
Y28	$\overline{PS5}$	AB9	PN7	AC21	PS0
Y29	PS5	AB10	PN8	AC22	PS1
AA1	VSS	AB11	PN9	AC23	PS2
AA2	PN4	AB12	VSSAPLL	AC24	PS3
AA3	$\overline{PN4}$	AB13	VCCAPLL	AC25	PS4
AA4	VSS	AB14	PN10	AC26	RFU[a]
AA5	VSS	AB15	$\overline{PN11}$	AC27	VSS
AA6	VSS	AB16	VSS		
AA7	VSS	AB17	$\overline{SN11}$		
AA8	VSS	AB18	VSS		
AA9	VSS	AB19	SCK		
AA10	VSS	AB20	TESTLO_AB20		
AA11	VSS	AB21	$\overline{PS0}$		
AA12	VSS	AB22	$\overline{PS1}$		
AA13	VSS	AB23	PS2		
AA14	VSS	AB24	$\overline{PS3}$		
AA15	VSS	AB25	$\overline{PS4}$		
AA16	VSS	AB26	RFU[a]		
AA17	VSS	AB27	VDDSPD		
AA18	VSS	AB28	VSS		
AA19	VSS	AC3	VSS		
AA20	VSS	AC4	PN5		
AA21	VSS	AC5	PN13		
AA22	VSS	AC6	RFU[a]		

Table 12-3. Advanced Memory Buffer Signals By Ball Number (Sheet 7 of 7)

Ball No.	Signal	Ball No.	Signal	Ball No.	Signal
AA23	VSS	AC7	PN12		
AA24	VSS	AC8	PN6		
AA25	VSS	AC9	PN7		
AA26	VSS	AC10	PN8		
AA27	PS9	AC11	PN9		
AA28	PS9	AC12	FBDRES		
AA29	VSS	AC13	PLLTSTO		
AB2	VSS	AC14	PN10		
AB3	$\overline{\text{RESET}}$	AC15	PN11		
AB4	PN5	AC16	RFU		
AB5	PN13	AC17	SN11		
AB6	RFU[a]	AC18	VSS		
These pin positions are reserved for forward clocks to be used in future AMB implementations.					

12.4 Package Information

Figure 12-2. Bottom View

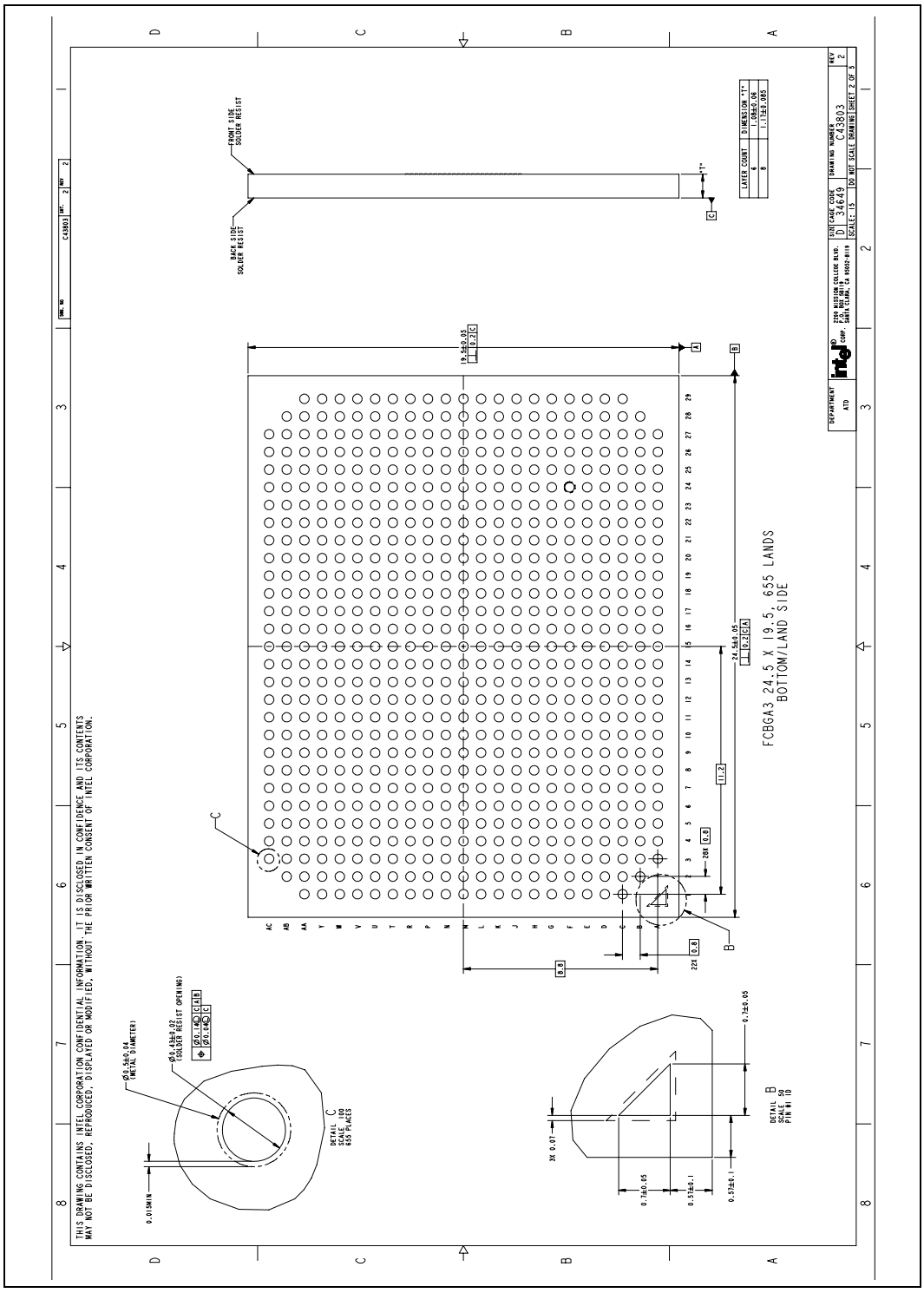


Figure 12-3. Top View

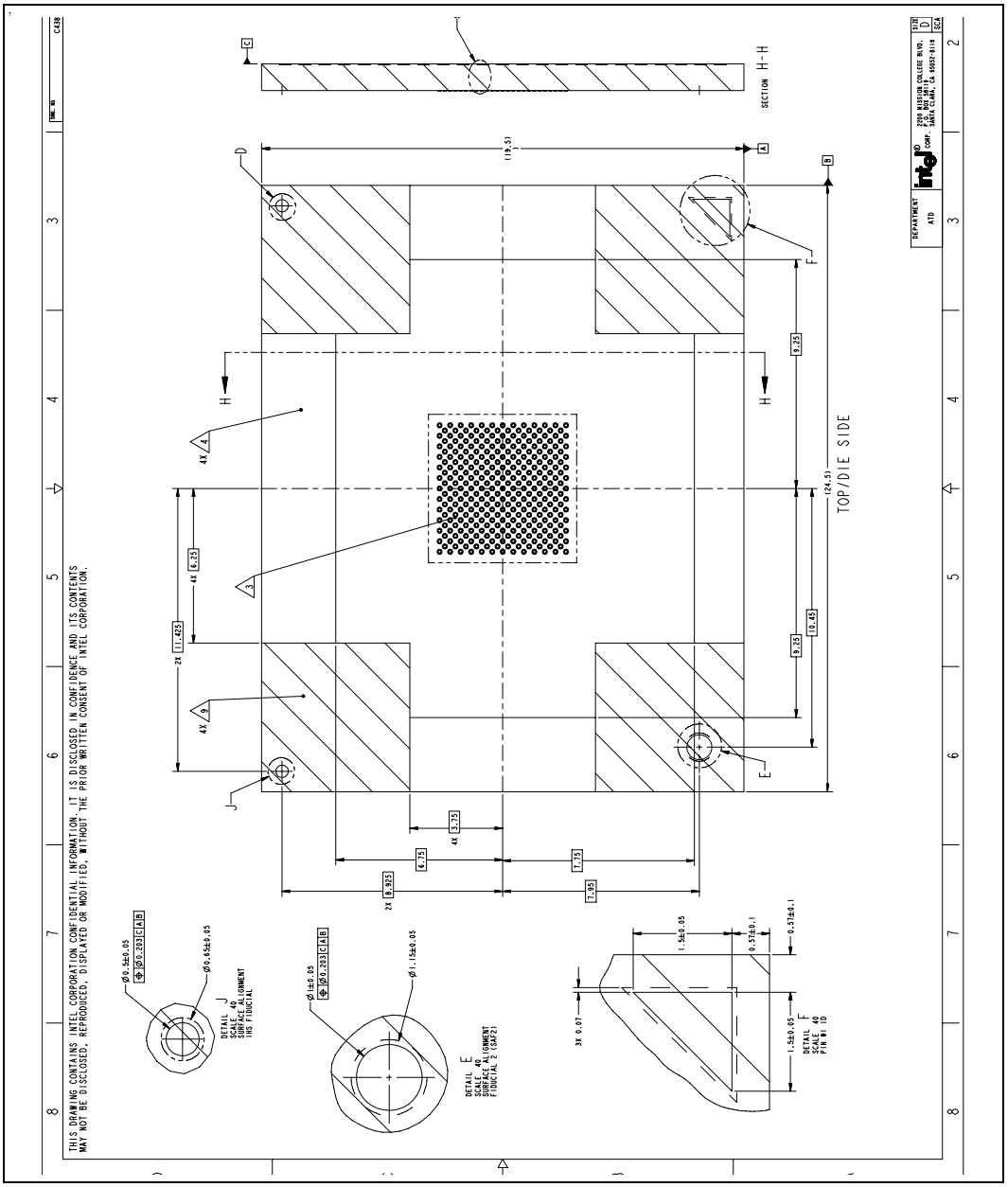
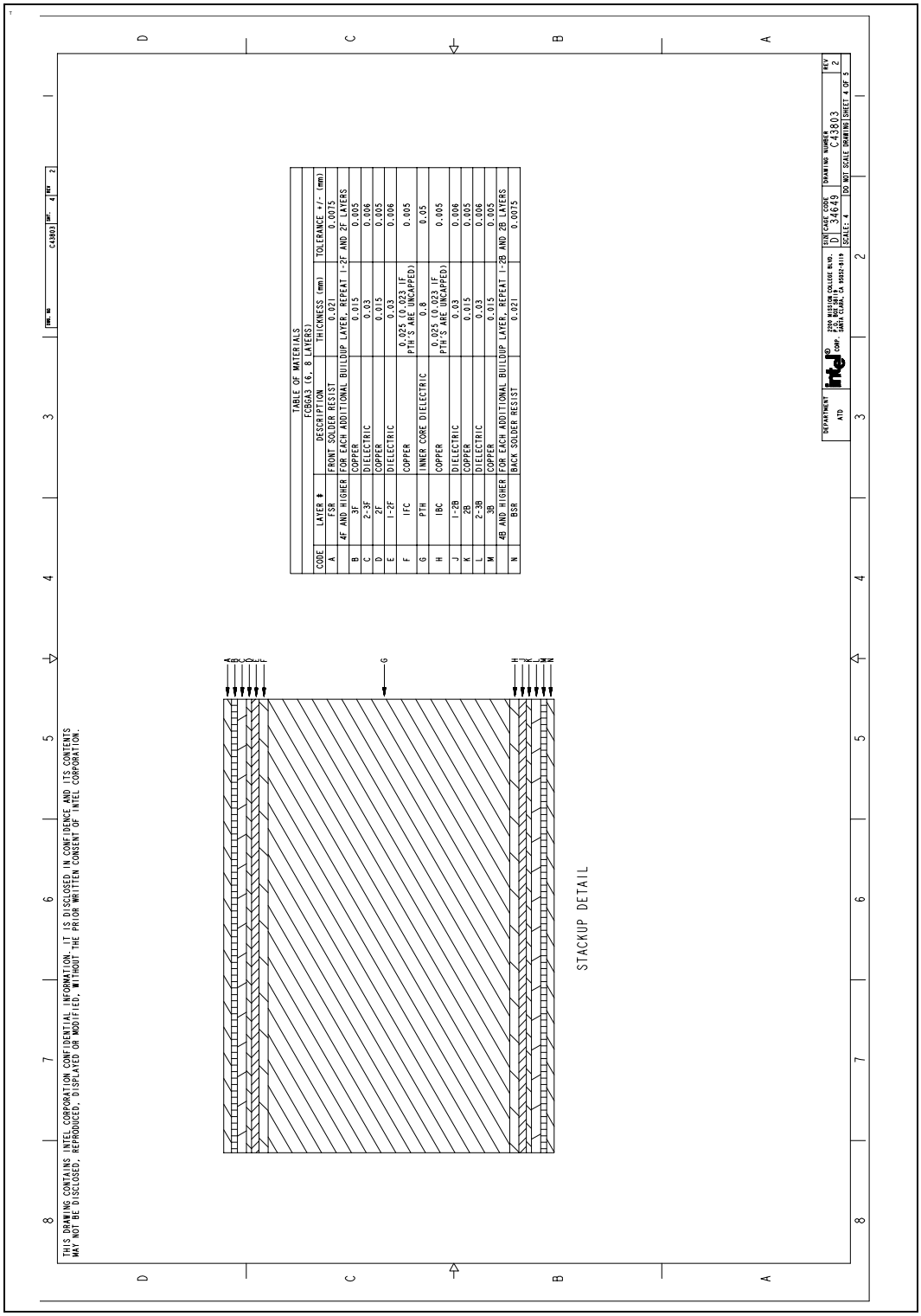


Figure 12-4. Package Stackup



§

13 Signal Lists

13.1 Conventions

The terms *assertion* and *de-assertion* are used extensively when describing signals, to avoid confusion when working with a mix of active-high and active-low signals. The term *assert*, or *assertion*, indicates that the signal is active, independent of whether the active level is represented by a high or low voltage. The term *de-assert*, or *de-assertion*, indicates that the signal is inactive.

Signal names may or may not have a "#" at appended to them. The "#" symbol at the end of a signal name indicates that the active, or asserted state occurs when the signal is at a low voltage level. When "#" is not present after the signal name the signal is asserted when at the high voltage level.

Differential pairs use the "#" to indicate the "negative" signal in the pair. The "positive" signal in When discussing data values used inside the component, the logical value is used; that is, a data value described as "1101b" would appear as "1101b" on an active-high bus, and as "0010b" on an active-low bus. When discussing the assertion of a value on the actual signal, the physical value is used; that is, asserting an active-low signal produces a "0" value on the signal.

Typical frequencies of operation for the fastest operating modes are indicated. Test guardbands are not included. No frequency is mentioned for asynchronous or analog signals.

Some signals or groups of signals have multiple versions. These signal groups may represent distinct but similar ports or interfaces, or may represent identical copies of the signal used to reduce loading effects. Table 13-1 shows the conventions used in this document.

Curly-bracketed non-trailing numerical indices, for example, "{X/Y}", represent replications of major buses. Square-bracketed numerical indices, , "[n:m]" represent functionally similar but logically distinct bus signals; each signal provides an independent control, and may or may not be asserted at the same time as the other signals in the grouping. In contrast, trailing curly-bracketed numerical indices, for example, "{x/y}" typically represent identical duplicates of a signal; such duplicates are provided for electrical reasons.

Table 13-1. Signal Naming Conventions

Convention	Expands to
RR{0/1/2}XX	Expands to: RR0XX, RR1XX, and RR2XX. This denotes similar signals on replicated buses.
RR[2:0]	Expands to: RR[2], RR[1], and RR[0]. This denotes a bus.
RR{0/1/2}	Expands to: RR2, RR1, and RR0. This denotes electrical duplicates.
RR# or RR[2:0]#	Denotes an active low signal or bus.

Table 13-2 lists the reference terminology used for signal types.

Table 13-2. Buffer Signal Types

Buffer Direction	Description
I	Input signal
O	Output signal
A	Analog
I/O	Bidirectional (input/output) signal

13.2 Intel 6400/6402 Advanced Memory Buffer (AMB) Pin Description List

Table 13-3 describes the Intel 6400/6402 Advanced Memory Buffer (AMB) packagepins.

Table 13-3. Pin Description (Sheet 1 of 2)

Signal	Type	Description
Channel Interface		
PN[13:0]	O	**Northbound Output Data:** High speed serial signal. Read path from AMB toward host on primary side of the DIMM connector.
$\overline{\text{PN}}$[13:0]	O	**Northbound Output Data Complement**
SN[13:0]	I	**Northbound Input Data:** High speed serial signal. Read path from the previous AMB toward this AMB on secondary side of the DIMM connector.
$\overline{\text{SN}}$[13:0]	I	**Northbound Input Data Complement**
PS[9:0]	I	**Southbound Input Data:** High speed serial signal. Write path from host toward AMB on primary side of the DIMM connector.
$\overline{\text{PS}}$[9:0]	I	Southbound Input Data Complement
SS[9:0]	O	**Southbound Output Data:** High speed serial signal. Write path from this AMB toward next AMB on secondary side of the DIMM connector. These output buffers are disabled for the last AMB on the channel.
$\overline{\text{SS}}$[9:0]	O	**Southbound Output Data Complement**
FBDRES	A	**External precision resistor connected to VCC.** On-die termination calibrated against this resistor.
DRAM Interface		
CB[7:0]	I/O	**Check bits**
DQ[63:0]	I/O	**Data**
DQS[17:0]	I/O	**Data Strobe:** DDR2 data and check-bit strobe.
$\overline{\text{DQS}}$[17:0]	I/O	**Data Strobe Complement:** DDR2 data and check-bit strobe complements.
A0A-A15A, A0B-A15B	O	**Address:** Used for providing multiplexed row and column address to SDRAM.
BA0A-BA2A, BA0B-BA2B	O	**Bank Active:** Used to select the bank within a rank.
$\overline{\text{RAS}}$A, $\overline{\text{RAS}}$B	O	**Row Address Strobe:** Used with $\overline{\text{CS}}$, $\overline{\text{CAS}}$, and $\overline{\text{WE}}$ to specify the SDRAM command.
$\overline{\text{CAS}}$A, $\overline{\text{CAS}}$B	O	**Column Address Strobe:** Used with $\overline{\text{CS}}$, $\overline{\text{RAS}}$, and $\overline{\text{WE}}$ to specify the SDRAM command.
$\overline{\text{WE}}$A, $\overline{\text{WE}}$B	O	**Write Enable:** Used with $\overline{\text{CS}}$, $\overline{\text{CAS}}$, and $\overline{\text{RAS}}$ to specify the SDRAM command.
$\overline{\text{CS}}$0A-$\overline{\text{CS}}$1A, $\overline{\text{CS}}$0B-$\overline{\text{CS}}$1B	O	**Chip Select:** Used with $\overline{\text{CAS}}$, $\overline{\text{RAS}}$, and $\overline{\text{WE}}$ to specify the SDRAM command. These signals are used for selecting one of two SDRAM ranks. $\overline{\text{CS}}$0 is used to select the first rank and $\overline{\text{CS}}$1 is used to select the second rank.
CKE0A-CKE1A, CKE0B-CKE1B	O	**Clock Enable:** DIMM command register enable.

Table 13-3. Pin Description (Sheet 2 of 2)

Signal	Type	Description
ODT0A, ODT0B	O	**DIMM On-Die-Termination:** Dynamic ODT enables for each DIMM on the channel.
CLK[3:0]	O	**Clock:** Clocks to DRAMs. CLK0 and CLK1 are always used. CLK2 and CLK3 are optional and may be disabled when not required.
$\overline{\text{CLK}}$[3:0]	O	**Clock Complement:** Clocks to DRAMs.
DDR Compensation		
DDRC_C14	A	**DDR Compensation Common:** Common return (ground) pin for DDRC_B18 and DDRC_C18
DDRC_B18	A	**DDR Compensation Ball** Resistor connected to Compensation Common above
DDRC_C18	A	**DDR Compensation Ball** Resistor connected to Compensation Common above
DDRC_B12	A	**DDR Compensation Ball** Resistor connected to VSS
DDRC_C12	A	**DDR Compensation Ball** Resistor connected to VDD
Clocking		
SCK	I	AMB Clock: This is one of the two differential reference clock inputs to the Phase Locked Loop in the AMB core. Phase Locked Loops in the AMB will shift this to all frequencies required by the core, DDR channels, and FBD Channel.
$\overline{\text{SCK}}$	I	AMB Clock Complement: This is the other differential reference clock input to the Phase Locked Loop in the AMB core. Phase Locked Loops in the AMB will shift this to all frequencies required by the core, DDR channels, and FBD Channel.
VCCAPLL	A	VCC: PLL Analog Voltage for the core PLL (See Chapter 8, Clocking)
VSSAPLL	A	VSS: PLL Analog Voltage for the core PLL (See Chapter 8, Clocking)
System Management		
SCL	I/O	**SMBus Clock**
SDA	I/O	**SMBus Address/Data**
SA[2:0]		**DIMM Select ID**
Reset		
RESET#		**Power Good Reset**
Miscellaneous Test		
TEST (6 pins)	NC	**Pin for debug and test. Must be floated on DIMM.**
TESTLO (3 pins)	A	**Pin for debug and test. Must be tied to Ground on DIMM**
TESTLO_AB20	A	**Pin for debug and test.** Connected to two resistors. One resistor is connected to VCCFBD, the other resistor is connected to VSS.
TESTLO_AC20	A	**Pin for debug and test.** Connected to two resistors. One resistor is connected to VCCFBD, the other resistor is connected to VSS.
Power Supplies		
VCC (2pins)	A	**1.5V nominal supply for core logic**
VCCFBD (8 pins)	A	**1.5V nominal supply for FBD high speed I/O**
VDD (24 pins)	A	**1.8V nominal supply for DDR I/O**
VSS (156 pins)	A	**Ground**
VDDSPD	A	**3.3V nominal supply for SMB receivers and ESD diodes**
Other Pins		
BFUNC	I	**Buffer Function Bit:** When BFUNC = 0, AMB is used as a regular buffer on FB-DIMM. When BFUNC = 1, AMB is used as either a repeater or a buffer for LAI function. On FB-DIMM, BFUNC is tied to Ground
RFU (18 pins)	NC	**Reserved for Future Use.** Must be floated on DIMM. RFU pins denoted by "a" are reserved for forwarded clocks in future AMB implementations.
Other No Connect Pins		
NC (129 pins)	NC	**No Connect pins**

Table 13-4 lists the signal types and pin counts.

Table 13-4. Pin Count

Signal	Pin Count
Channel Interface	97
DRAM Interface	170
DDR Compensation	5
Clocking	5
System Management	5
Reset	1
Miscellaneous Test	11
Power Supplies	213
Other Pins	19
Sub-total	526
Other No Connect Pins	129
Total	655

§

14 Registers

14.1 Access Mechanisms

The Intel 6400/6402 Advanced Memory Buffer (AMB) component supports PCI configuration space access as defined in the *PCI Local Bus Specification, Rev.2.2*. The internal registers of the AMB can be accessed in byte (8-bit), word (16-bit), or double word (32-bit) quantities. All multi-byte numeric fields use "little-endian" ordering (that is, lower addresses contain the least significant parts of the field). As a memory buffer (not a PCI device or bridge) the AMB is not fully compliant with this mechanism with respect to the standard registers (those with offsets 0-3Fh).

The AMB will not be mapped into the host system's PCI Plug and Play hierarchy, but is accessed through a method that is host controller implementation specific.

Problems will occur if the host system maps the registers into the PCI PnP hierarchy.

Configuration accesses are transported on the FBD link as configuration read and write commands, which mimic the corresponding PCI commands. The AMB responds to any device encoded in an FBD command. The AMB responds only to SM Bus requests that match the NodeID. The "Device:" mentioned in the heading of each configuration register table does not designate the PCI device; it designates the SM Bus node.

14.1.1 Conflict Resolution and Usage Model Limitations

AMB accepts configuration register reads and writes through the FBD link and through SMBus transactions. In Logic Analyzer Interface (LAI) mode, registers are not accessible through the in-band FBD link configuration read/write commands.

Registers do not incur read side-effects.

14.1.2 FBD Data on Configuration Read Returns

FBD read return data from the AMB is described in the *FB-DIMM Architecture and Protocol Specification*. Configuration reads are sent in northbound data frames. Only the bottom four bytes of this data are defined for a configuration access. The rest of the bytes in the read return are undefined. Legal CRCs are generated for these undefined inbound bytes.

14.1.3 Non-Existent Register Bits

To comply with the PCI specification, accesses to non-existent registers and bits will be treated as follows:

Table 14-1. Access to "Non-existent" Register Bits

Access to	Writes	Reads
Registers in unimplemented functions	Have no effect	AMB returns -1
Registers not listed	Have no effect	AMB returns all zeroes
Reserved bits in registers	Software must read-modify-write to preserve the value	AMB returns implementation specific values

14.1.4 Register Attribute Definition

Table 14-2. Register Attributes Definitions

Attribute	Abbreviation	Description
Read Only	RO	This bit is set by the hardware. Software can only read the bit. Writes to the register have no effect.
Write Only	WO	The bit is not implemented as a bit. The write causes some hardware event to take place. Read returns all zeroes.
Read/Write	RW	The bit can be read or written by software.
Read/Write / Clear	RWC	The bit can be either read or cleared by software. In order to clear a bit, the software must write a one to it. Writing a zero to an RWC bit will have no effect. Hardware will set this bit.
Read/Write / Set	RWS	The bit can be either read or set by software. In order to set this RWS bit, the software must write a one to it. Writing a zero to an RWS bit will have no effect. Hardware will clear this bit.
Read/Write Lock	RWL	The bit can be read and written by software. Hardware or a configuration bit can lock this bit and prevent it from being updated.
Read/Write Once	RWO	The bit can be read by software. It can also be written by software but the hardware prevents writing/setting it more than once without a prior “effective” reset. This protection applies on a byte-by-byte basis. For example, if a two-bit RWO field straddles a byte boundary and only one byte is written, then the written bit cannot be rewritten (unless reset). However, the unwritten bit can still be written once. This is a special form of RWL
Read/Restricted Write	RRW	The bit field can be read by software. Only a restricted set of values can be written through a configuration write in-band from the host. Illegal values will be aborted and dropped.
Sticky	All of the above with “ST” appended to the end	The bit is “sticky” or unchanged by a link reset. These bits can only be defaulted by a power-up reset.
Reserved	RV	This bit is reserved for future expansion and must not be written. The *PCI Local Bus Specification,* Revision 2.2 requires that reserved bits must be preserved. Any software that modifies a register that contains a reserved bit is responsible for reading the register, modifying the desired bits, and writing back the result.

14.1.5 Binary Number Notation

When references are made to binary numbers, the following notation is used:

n'bXX
n - number of digits
b - indicates binary
XX - binary value

For example, 2'b01 is two binary digital of value "01".

14.1.6 Function Mapping

The following functions are described in this chapter:

0) PCI Standard Header Identification Registers

1) FBD Link Registers

2) Implementation Specific FBD Registers

3) DDR and Miscellaneous Registers

4) Implementation Specific DDR Initialization and Calibration Registers

5) DFX Registers

6) Bring-up and Debug Registers

Table 14-3. Function Mapping Legend

Fill	Description
RegName	Required Architected Register"RegName" in these bytes
Reserved	Reserved Register for future architecture in these bytes
RegName	Optional Architected Register"RegName" in these bytes • No Implementation Specific Registers usage recommended
	Unused Register location available for Implementation Specific use

Table 14-4. Function 0: PCI Standard Header Identification Registers

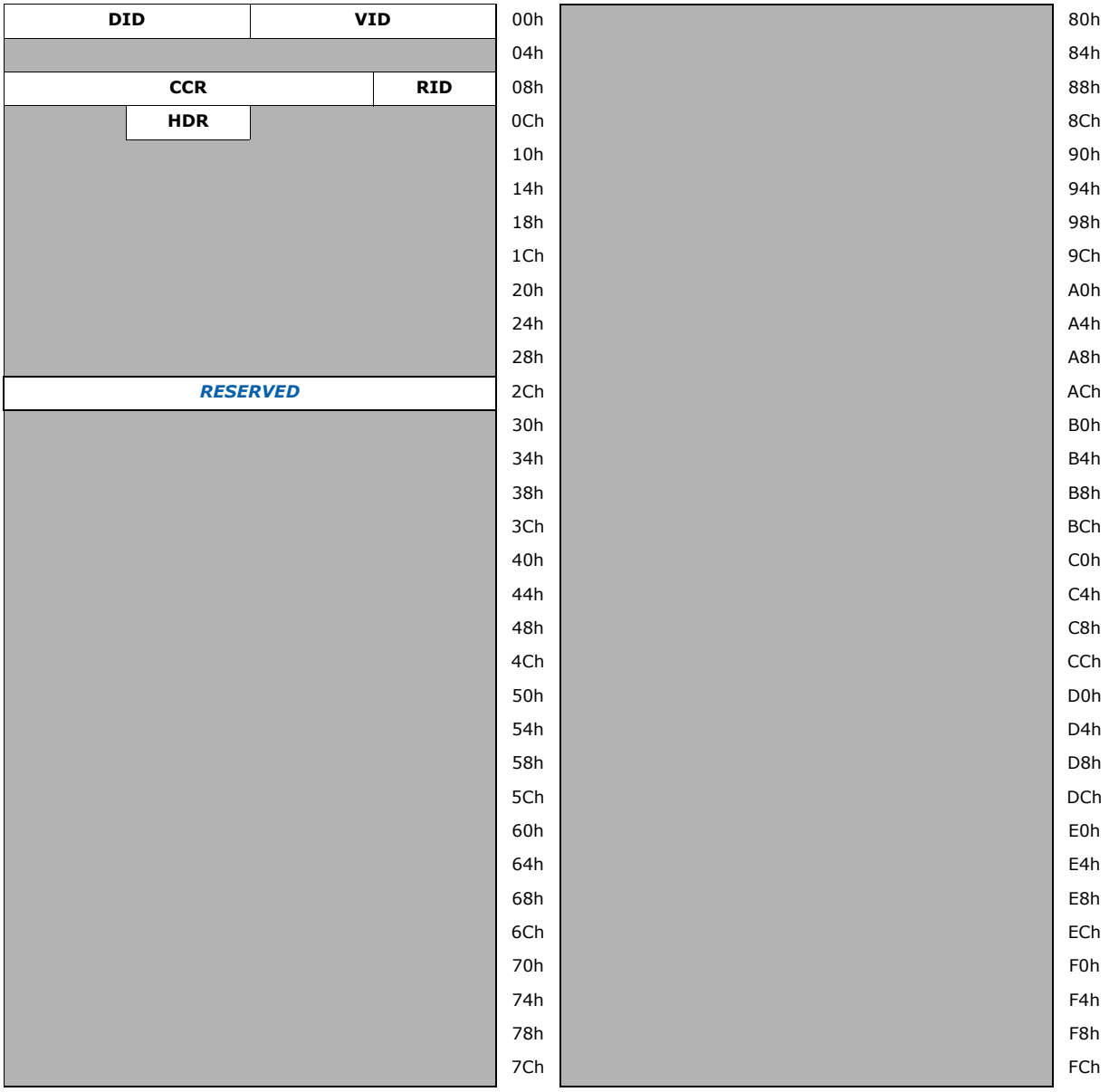


Table 14-5. Function 1: FBD Link Registers

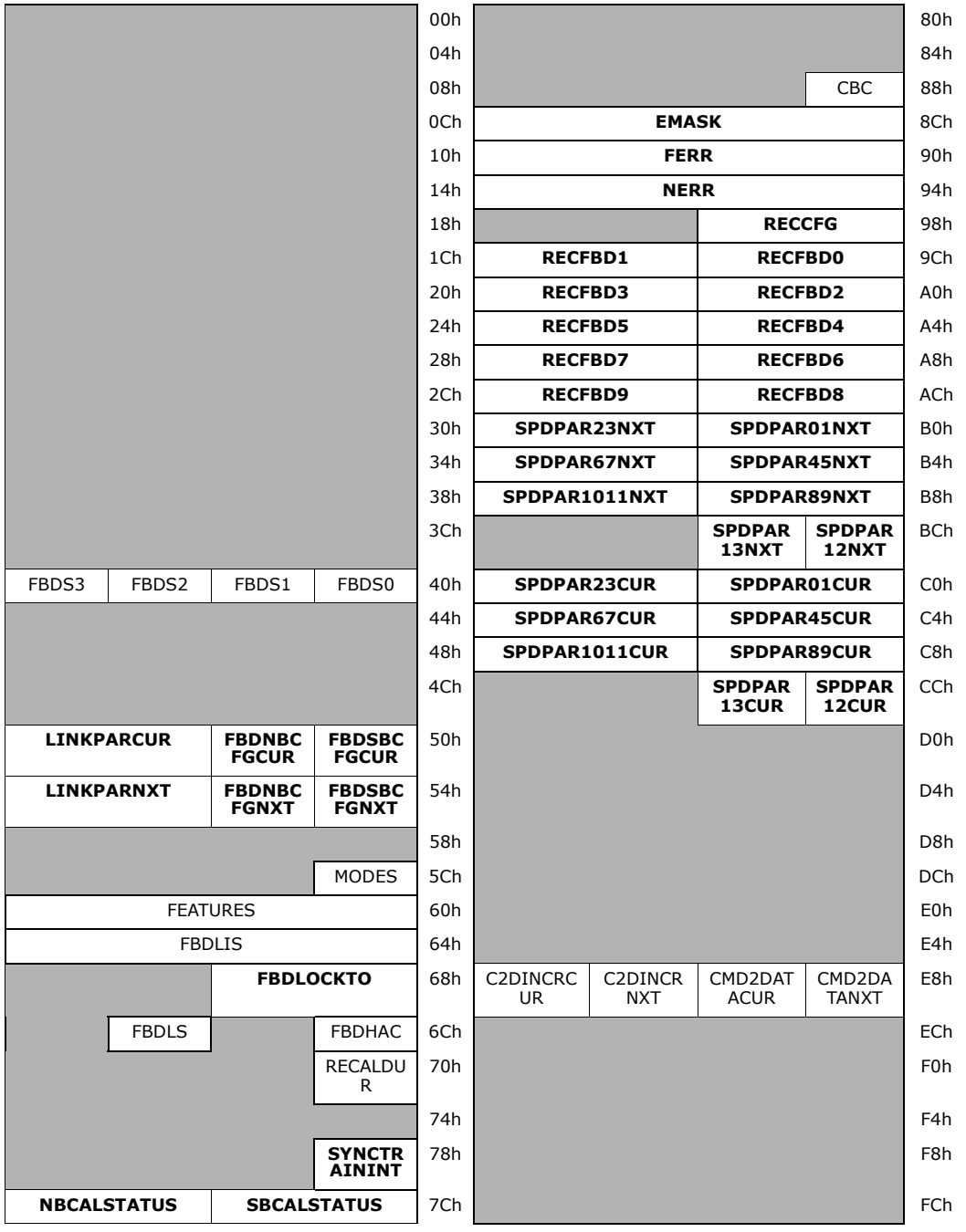


Table 14-6. Function 2: Implementation Specific FBD Registers

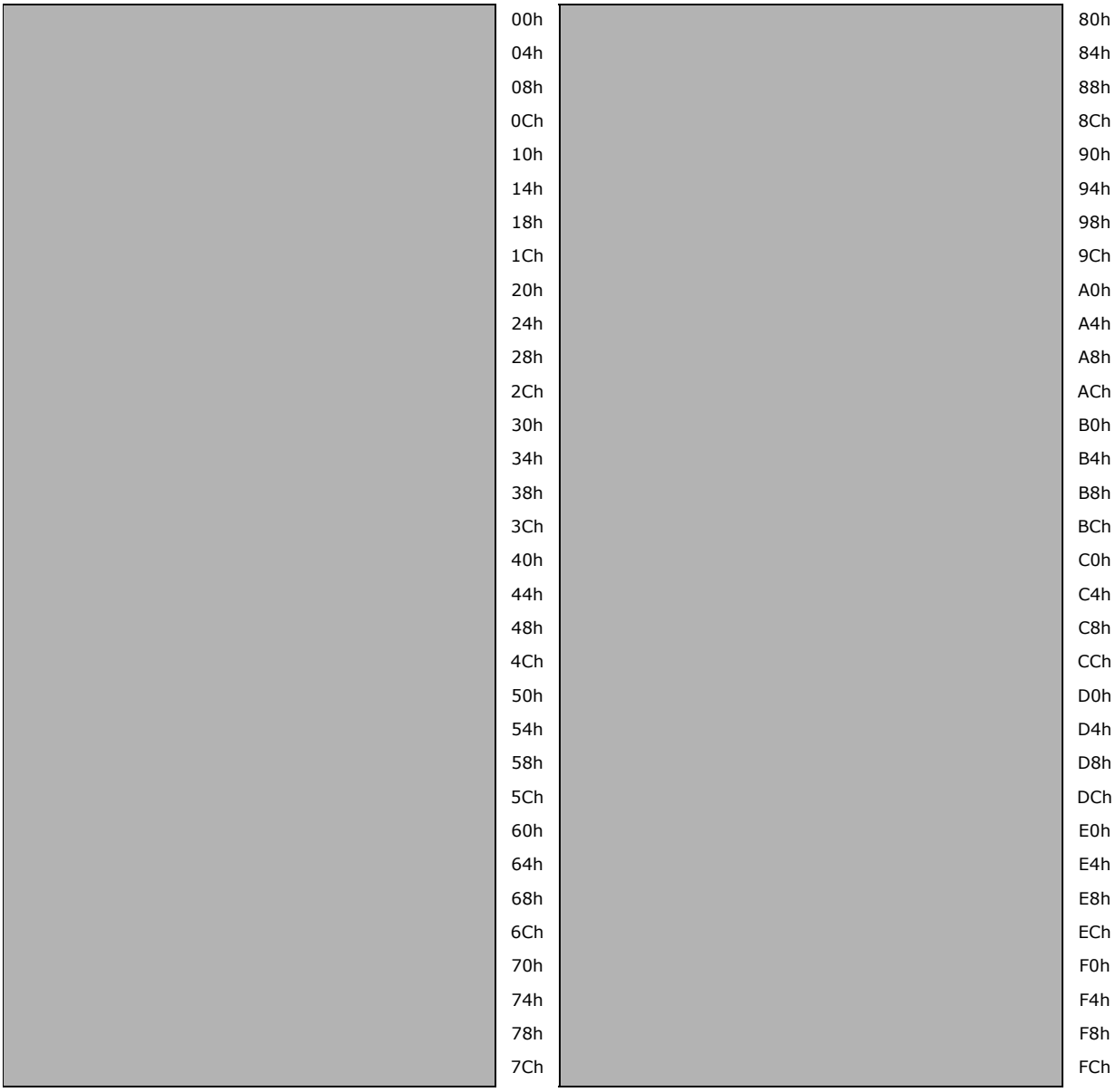


Table 14-7. Function 3: DDR and Miscellaneous Registers

				00h	UPDATED	TEMPHI	TEMPMID	TEMPLO	80h
				04h			TEMP	TEMPSTAT	84h
				08h					88h
				0Ch					8Ch
				10h					90h
				14h					94h
				18h					98h
				1Ch	MB_START_ADDR				9Ch
				20h	MB_END_ADDR				A0h
				24h	MB_LFSR				A4h
				28h	MBFADDRPTR				A8h
				2Ch					ACh
				30h	MB_ERR_DATA00				B0h
				34h	MB_ERR_DATA01				B4h
				38h	MB_ERR_DATA02				B8h
				3Ch	MB_ERR_DATA03				BCh
MBCSR				40h			MB_ERR_DATA04		C0h
MBADDR				44h					C4h
MBDATA0				48h					C8h
MBDATA1				4Ch					CCh
MBDATA2				50h					D0h
MBDATA3				54h					D4h
MBDATA4				58h					D8h
MBDATA5				5Ch					DCh
MBDATA6				60h					E0h
MBDATA7				64h					E4h
MBDATA8				68h					E8h
MBDATA9				6Ch					ECh
DAREFTC				70h					F0h
MTR	DSREFTC			74h					F4h
DRT				78h					F8h
DRC				7Ch					FCh

Table 14-8. Function 4: Implementation Specific DDR Initialization and Calibration Registers

								00h	DCALDATA59	DCALDATA58	DCALDATA57	DCALDATA56				80h
								04h	DCALDATA63	DCALDATA62	DCALDATA61	DCALDATA60				84h
								08h	DCALDATA67	DCALDATA66	DCALDATA65	DCALDATA64				88h
								0Ch	DCALDATA71	DCALDATA70	DCALDATA69	DCALDATA68				8Ch
								10h		DDBISTLM						90h
								14h		**RCVENAC**						94h
								18h	DSRETC							98h
								1Ch				DQSFAIL1				9Ch
								20h	**DRRTC00**							A0h
								24h	**DRRTC01**							A4h
								28h								A8h
								2Ch								ACh
								30h	**DQSOFCS00**							B0h
								34h	**DQSOFCS01**							B4h
								38h	**DQSOFCS10**							B8h
								3Ch	**DQSOFCS11**							BCh
DCALCSR								40h	**DQSOFCS12**	**DQSOFCS02**		**DRRTC02**				C0h
DCALADDR								44h		**DRAMDLLC**						C4h
DCALDATA3	DCALDATA2	DCALDATA1	DCALDATA0					48h	**WPTRTC0**							C8h
DCALDATA7	DCALDATA6	DCALDATA5	DCALDATA4					4Ch				**WPTRTC1**				CCh
DCALDATA11	DCALDATA10	DCALDATA9	DCALDATA8					50h	**DDQSCVDP0**							D0h
DCALDATA15	DCALDATA14	DCALDATA13	DCALDATA12					54h	**DDQSCVDP1**							D4h
DCALDATA19	DCALDATA18	DCALDATA17	DCALDATA16					58h	**DDQSCADP0**							D8h
DCALDATA23	DCALDATA22	DCALDATA21	DCALDATA20					5Ch	**DDQSCADP1**							DCh
DCALDATA27	DCALDATA26	DCALDATA25	DCALDATA24					60h	**DRRTC00**							E0h
DCALDATA31	DCALDATA30	DCALDATA29	DCALDATA28					64h								E4h
DCALDATA35	**DCALDATA34**	**DCALDATA33**	**DCALDATA32**					68h	**FIVESREG**							E8h
DCALDATA39	**DCALDATA38**	**DCALDATA37**	**DCALDATA36**					6Ch	**AAAAREG**							ECh
DCALDATA43	**DCALDATA42**	**DCALDATA41**	**DCALDATA40**					70h			**DIOMON**					F0h
DCALDATA47	**DCALDATA46**	**DCALDATA45**	**DCALDATA44**					74h			**ODTZTC**					F4h
DCALDATA51	**DCALDATA50**	**DCALDATA49**	**DCALDATA48**					78h	**DRAMISCTL**							F8h
DCALDATA55	**DCALDATA54**	**DCALDATA53**	**DCALDATA52**					7Ch				**DDR2ODTC**				FCh

Table 14-9. Function 5: DFX Registers

RESERVED		00h			80h
RESERVED		04h			84h
RESERVED		08h			88h
RESERVED		0Ch			8Ch
		10h			90h
		14h			94h
		18h			98h
		1Ch			9Ch
		20h			A0h
		24h			A4h
		28h			A8h
		2Ch			ACh
		30h			B0h
		34h			B4h
		38h	**LAI**		B8h
TRANSCFG		3Ch	**SBMATCHU**		BCh
TRANDERR1	**TRANDERR0**	40h	**SBMATCHL0**		C0h
TRANDERR3	**TRANDERR2**	44h	**SBMATCHL1**		C4h
TRANDERR5	**TRANDERR4**	48h	**SBMATCHL2**		C8h
TRANDERR7	**TRANDERR6**	4Ch	**SBMASKU**		CCh
	TRANDERR8	50h	**SBMASKL0**		D0h
		54h	**SBMASKL1**		D4h
		58h	**SBMASKL2**		D8h
		5Ch		**MMEVENTSEL**	DCh
		60h	**EVENTSEL0**		E0h
		64h	**EVENTSEL1**		E4h
		68h	**EVENTSEL2**		E8h
		6Ch			ECh
		70h	**EVENT**		F0h
		74h	**EVBUS**		F4h
		78h			F8h
		7Ch	**STUCKL**	**EICNTL**	FCh

Table 14-10. Function 6: Bring-up and Debug Registers

Register Bit Location			
RESERVED	04h	SBFIBPGCTL	84h
	10h	SBFIBRXMSK	90h
	14h	SBFIBTXSHFT	94h
	18h	SBFIBRXSHFT	98h
	1Ch	SBFIBRXLNERR	9Ch
	20h	SBFIBPATTBUF2	A0h
	24h	SBFIBPATTBUF2EN	A4h
	28h		A8h
	2Ch		ACh
	30h	SBFIBINIT	B0h
	34h	SBIBISTMISC	B4h
	38h		B8h
	3Ch		BCh
	40h	NBFIBPORTCTL	C0h
	44h	NBFIBPGCTL	C4h
	48h	NBFIBPATTBUF1	C8h
	4Ch	NBFIBTXMSK	CCh
	50h	NBFIBRXMSK	D0h
	54h	NBFIBTXSHFT	D4h
	58h	NBFIBRXSHFT	D8h
	5Ch	NBFIBRXLNERR	DCh
	60h	NBFIBPATTBUF2	E0h
	64h	NBFIBPATTBUF2EN	E4h
	68h		E8h
	6Ch		ECh
	70h	NBFIBINIT	F0h
	74h	NBIBISTMISC	F4h
	78h		F8h

14.2 PCI Standard Header Identification Registers (Function 0)

14.2.1 VID: Vendor Identification Register

This register identifies Intel as the manufacturer of the AMB.

Device: NodeID
Function: 0
Offset: 00h

Bit	Attr	Default	Description
15:0	RO	8086h	**Vendor Identification Number** The value assigned to Intel.

14.2.2 DID: Device Identification Register

These registers combined with the vendor identification register uniquely identifies the AMB Devices.

Device: NodeID
Function: 0
Offset: 02h

Bit	Attr	Default	Description
15:0	RO	A620h	**Device Identification Number** Identifies each function of the AMB

14.2.3 RID: Revision Identification Register

This register contains the revision number of the AMB.

Device: NodeID
Function: 0
Offset: 08h

Bit	Attr	Default	Description
7:0	RO	10h	**Revision Identification Number**: RID "00h" = A0 stepping "01h" = A1 stepping "02h" = A2 stepping "10h" = B0 stepping

14.2.4 CCR: Class Code Register

This register contains the Class Code for the AMB, specifying the device function.

Device: NodeID
Function: 0
Offset: 09h

Bit	Attr	Default	Description
23:16	RO	05h	**Base Class**. This field indicates the general device category. For the AMB, this field is hardwired to 05h, indicating it is a "memory controller".
15:8	RO	00h	**Sub-Class.** This field qualifies the Base Class, providing a more detailed specification of the device function. For the AMB, this field is hardwired to 00h, indicating it is a "RAM".
7:0	RO	00h	**Register-Level Programming Interface.** This field identifies a specific programming interface (if any), that device independent software can use to interact with the device. There are no such interfaces defined for "memory controllers".

14.2.5 HDR: Header Type Register

This register identifies the header layout of the configuration space.

Device: NodeID
Function: 0
Offset: 0Eh

Bit	Attr	Default	Description
7	RO	1	**Multi-function Device.** Selects whether this is a multi-function device, that may have alternative configuration layouts. The AMB has more than the 256 bytes of configuration registers allotted to a single function. Therefore, the AMB is defined to be a multifunction device, and this bit is hardwired to 1.
6:0	RO	00h	**Configuration Layout.** This field identifies the format of the 10h through 3Fh space. The AMB uses header type "00": these bits are hardwired to 00h.

14.3 FBD Link Registers (Function 1)

14.3.1 FBD Link Control and Status

14.3.1.1 FBDS0: FBD Status 0

This register contains copies of status bits returned by the AMB in the most recent northbound status frame when SYNC command R[1:0] field is 2'b00.

In the absence of SYNCs to this register, this register is not updated.

Device: NodeID
Function: 1
Offset: 40h

Bit	Attr	Default	Description
7:5	RV	0h	*Reserved*
4	RO	0h	SP: Parity: This bit contains an odd parity bit that covers the S[3:0] field.

Device: NodeID Function: 1 Offset: 40h			
Bit	**Attr**	**Default**	**Description**
3	RO	0h	**S3: Northbound Debug Event** (1 = asserted, 0 = inactive): This bit is used to communicate debug events to the host.
2:1	RO	0h	**S[2:1]: Thermal Trip:** This field indicates various thermal conditions of the AMB as follows: • 00 – Below TEMPLO • 01 – Above TEMPLO • 10 – Above TEMPMID and falling • 11 – Above TEMPMID and rising The TEMPLO threshold is generally used to inform the host to accelerate refresh events. The TEMPMID threshold is generally used to inform the host that a thermal limit has been exceeded and that thermal throttling is needed. Refer to the RAS chapter for more details on thermal management.
0	RO	0h	S0: ERROR Asserted: This bit indicates an error has been detected by the AMB. Errors can be alert or other type.

14.3.1.2 FBDS1: FBD Status 1

This register contains copies of status bits returned by the AMB in the most recent northbound status frame when SYNC command R[1:0] field is 2'b01.

Device: NodeID Function: 1 Offset: 41h			
Bit	**Attr**	**Default**	**Description**
7:5	RV	0h	*Reserved*
4	RO	0h	SP: Parity: This bit contains an odd parity bit that covers the S[3:0] field.
3:1	RV	0h	*Reserved*
0	RO	0h	S0: Data Merge Error: This bit indicates that the northbound data merge alignment logic of an intermediate AMB cannot met the timing required to merge its DRAM data into the northbound data stream when required. Refer to the initialization chapter for details.

14.3.1.3 FBDS2: FBD Status 2

This register contains copies of status bits returned by the AMB in the most recent northbound status frame when SYNC command R[1:0] field is 2'b10.

Device: NodeID Function: 1 Offset: 42h			
Bit	**Attr**	**Default**	**Description**
7:5	RV	0h	*Reserved*
4	RO	0h	SP: Parity: This bit contains an odd parity bit that covers the S[3:0] field.
3:0	RV	0h	*Reserved*

14.3.1.4 FBDS3: FBD Status 3

This register contains copies of bits that were returned by the AMB in the most recent northbound status frame when the SYNC command R[1:0] field is 2'b11. This can also be written with an override value that will be returned if selected during SYNC command.

Device: NodeID
Function: 1
Offset: 43h

Bit	Attr	Default	Description
7:6	RV	0h	*Reserved*
5	RW	0h	**OVREN: Use values written by user.** Setting this bit causes the values specified in the lower 5 bits of this register returned as-is if requested by SYNC command.
4	RW	1h	**USRPAR: User Specified parity for USRVAL**
3:0	RW	0h	**USRVAL: User Specified value**

14.3.1.5 MODES: Operating Mode

This register contains overview configuration status of chip.

Device: NodeID
Function: 1
Offset: 5Ch

Bit	Attr	Default	Description
7	RO	1	NORMAL: • 1 = Normal AMB Buffer
6	RO	0	LAI: • 1 = LAI
5	RO	0	REPEATER: • 1 = Repeater
4	RO	0	TRANSPARENT: • 1 = Transparent Test Mode
3:0	RV	0	*Reserved*

14.3.1.6 FEATURES: Capabilities

This register reports optional capabilities of this DIMM.

Device: NodeID
Function: 1
Offset: 60h

Bit	Attr	Default	Description
31:15	RV	0h	***Reserved***
14:11	RO	0011	**DDRFREQ: DDR2 frequencies supported** • 1XXX = reserved • X1XX = DDR2-800 • XX1X = DDR2-667 • XXX1 = DDR2-533

Device: NodeID Function: 1 Offset: 60h			
Bit	**Attr**	**Default**	**Description**
10	RO	0	**VARLAT: Variable Read Latency Mode** • 1 = Support Variable Read Latency on data returns • 0 = Not supported
9	RO	1	**LAI: Logic Analyzer Interface Mode** • 1 = Support remapping DDR interface as Logic Analyzer Interface • 0 = Not supported
8	RO	0	**DMASK: Data Mask for non-ECC Write Data** • 1 = Support data mask with non-ECC Write • 0 = Not supported
7	RO	0	**L0S: Low Power Link State** • 1 = Support L0s state • 0 = Not supported
6:2	RO	1Eh	**NBWC: Northbound Width Capability** • 1XXXX = 14 bits NB width supported • X1XXX = 14bits fail over to 13 bits mode supported. • XX1XX = 13 bits NB width supported • XXX1X = 13 bits fail over to 12 bits mode supported.
1:0	RO	01	**SBWC: Southbound Width Capability** • X1 = 10 SB bits: Device supports 10-bits and 10-bit fail-over to 9-bits. Both configurations deliver 72-bits of data payload frame. • 1X = Reserved

14.3.1.7 FBDLIS: FBD Link Initialization Status

This register reports FBD initialization status and is only valid when the link is up since it is not sticky.

Device: NodeID Function: 1 Offset: 64h			
Bit	**Attr**	**Default**	**Description**
31:20	RV	0	***Reserved***
19	RO	0	DATAMERGEERROR: NorthBound Data Merge Error • 1 = NB merge error
18	ROST	0	**NBMERGEDIS: NorthBound Merge Disable** **Set by TS2 packet addressed to it** • 1 = Disable NB merge • Note: state in AMB should be sticky through fast link reset until new TS2 resets bit or hard pin reset
17:12	RO	3Fh	**NBWCFG: Northbound width configuration set by TS3** • See table in *FBD Architecture & Protocol Specification* for full decoding • [5:4] = Selects 14, 13 or 12 lane operation. = Protocol Selection[1:0] out TS3 Protocol Selection[3:0] • [3:0] - Selects none or one lane to map out = NB Channel Configuration [3:0] in TS3
11:8	RO	Fh	**SBWCFG: Southbound width capability set by TS3** • See Table in *FBD Architecture & Protocol Specification* for full decoding • [3:0] - Selects none or one lane to map out
7	RV	0	***Reserved***

Device: NodeID
Function: 1
Offset: 64h

Bit	Attr	Default	Description
6	RO	0	**TS2RESP: Responded to a TS2 packet addressed to it** 1: TS2 was addressed to this AMB • In polling mode - DS matches TS2 AMB_ID value • Undefined in other states
5	RO	0	**SB2NBLBMAP: Specifies if upper or lower SB lanes are reflected in TS1** Valid only during testing phase (TS1) 1 = upper SB bit lanes 0 = lower SB lanes
4	RO	0	**LASTAMBFLAG: Indicates if this AMB is acting like the last AMB** • In Disable, Calibrate - always 0. • In training - Set if DS matches TS0 AMB_ID value • Retains value after training till next reset.
3:0	RO	0h	**LINITST: Link initialization state** Encoding is • 0000 - Disable • 0001 - Calibrate • 0010 - Training • 0011 - Testing • 0100 - Polling • 0101 - Config • 0110 - L0 • 0111 - L0s • 1000 - Recalibrate

14.3.1.8 FBDSBCFGNXT: FBD SB Link Electrical Configuration

This register contains next settings of control bits to set link electrical parameters to match link length and frequency characteristics.

Device: NodeID Function: 1 Offset: 54h			
Bit	**Attr**	**Default**	**Description**
7:5	RV	0h	*Reserved*
4:3	RWST	00	**SBTXDRVCUR:** '11' = Small '10' = reserved '01' = Regular '00' = Large
2:1	RWST	10b	**SBTXDEEMP:** 00 = 0 dB 01 = 3.5dB 10 = 6dB 11 = 9.5dB
0	RWST	1b	**SBRESYNCEN:** '1' = SB pass-thru data is in Re-sync mode '0' = SB pass-thru is in Re-sample mode.

14.3.1.9 FBDNBCFGNXT: FBD NB Link Electrical Configuration

This register contains next settings of control bits to set link electrical parameters to match link length and frequency characteristics.

Device: NodeID Function: 1 Offset: 55h			
Bit	**Attr**	**Default**	**Description**
7:5	RV	0h	*Reserved*
4:3	RWST	00	**NBTXDRVCUR:** '11' = Small '10' = reserved '01' = Regular '00' = Large
2:1	RWST	10b	**NBTXDEEMP:** 00 = 0 dB 01 = 3.5dB 10 = 6dB 11 = 9.5dB
0	RWST	1b	**NBRESYNCEN:** '1' = SB pass-thru data is in Re-sync mode '0' = SB pass-thru is in Re-sample mode.

14.3.1.10 LINKPARNXT: FBD Link Frequency

This register contains current settings of control bits to set link electrical parameters to match link length and frequency characteristics.

Device: NodeID
Function: 1
Offset: 56h

Bit	Attr	Default	Description
15:2	RV	0h	*Reserved*
1:0	RWST		CFREQ: Current Link Frequency • 11 = DDR2-800 • 10 = DDR2-667 • 01 = DDR2-533 • 00 = Uninitialized

14.3.1.11 FBDSBCFGCUR: FBD SB Link Electrical Configuration

This register contains current settings of control bits to set link electrical parameters to match link length and frequency characteristics.

Device: NodeID
Function: 1
Offset: 50h

Bit	Attr	Default	Description
7:5	RV	0h	*Reserved*
4:3	ROST	00b	**SBTXDRVCUR:** '11' = Small '10' = reserved '01' = Regular '00' = Large
2:1	ROST	10b	**SBTXDEEMP:** 00 = 0 dB 01 = 3.5dB 10 = 6dB 11 = 9.5dB
0	ROST	1	**SBRESYNCEN:** '1' = SB pass-thru data is in Re-sync mode '0' = SB pass-thru is in Re-sample mode.

14.3.1.12 FBDNBCFGCUR: FBD NB Link Electrical Configuration

This register contains current settings of control bits to set link electrical parameters to match link length and frequency characteristics.

Device: NodeID
Function: 1
Offset: 51h

Bit	Attr	Default	Description
7:5	RV	0h	*Reserved*
4:3	ROST	00	**NBTXDRVCUR:** '11' = Small '10' = reserved '01' = Regular '00' = Large
2:1	ROST	10b	**NBTXDEEMP:** 00 = 0 dB 01 = 3.5dB 10 = 6dB 11 = 9.5dB
0	ROST	1	**NBRESYNCEN:** '1' = SB pass-thru data is in Re-sync mode '0' = SB pass-thru is in Re-sample mode.

14.3.1.13 LINKPARCUR: FBD Link Frequency

This register contains current settings of control bits to set link electrical parameters to match link length and frequency characteristics

Device: NodeID
Function: 1
Offset: 52h

Bit	Attr	Default	Description
15:2	RV	0h	*Reserved*
1:0	ROST	01	CFREQ: Current Link Frequency • 11 = DDR2-800 • 10 = DDR2-667 • 01 = DDR2-533 • 00 = Reserved

14.3.1.14 FBDLOCKTO: FBD Bit Lock Time Out Register

This register contains the bit lock time out value. This value is used by the to figure out when it should stop waiting for lanes to bit lock and make forward progress. The register also contains the width that the host is going to be starting with on the NB side. This will be used to mask off lanes for initial bit lock decision

Device: NodeID
Function: 1
Offset: 68h

Bit	Attr	Default	Description
15:2	RWST	0594h	BLTOCNT: Bit Lock Time Out Counter default: 1428 frames
1:0	RWST	0h	NBLINKCFG: Northbound Link Config 00 = 14 lane 01 = 13 lane 10 = 12 lane 11 = reserved

14.3.1.15 FBDHAC: FBD Hot Add Control

This register contains control to aid in hot add functionality.

Device: NodeID
Function: 1
Offset: 6Ch

Bit	Attr	Default	Description
7:2	RV	0	*Reserved*
1	RO	0	NB_DATA_ALL_ONES_FLAG: • 1 = Receiving ones on sufficient NB lanes to support init • 0 = Not receiving calibrate handshake on NB Rx
0	RW	0	DRIVE_ONES_SB: • 1 = Enable SB Tx Outputs and drive one's • 0 = Normal operation

14.3.1.16 FBDLS: FBD Link Status

This register reports AMB FBD link status.

Device: NodeID
Function: 1
Offset: 6Eh

Bit	Attr	Default	Description
7:3	RV	0h	*Reserved*
2	RO	0	**NLQS: Northbound Lane Electrical Status** '1' = Northbound lanes are quiesced '0' = Northbound lanes are active
1	RO	0	**SLQS: Southbound Lane Electrical Status** '1' = Southbound lanes are quiesced '0' = Southbound lanes are active
0	RO	0	**LNKRDY: Link Ready** '1' = FBD link is ready to accept requests and deliver responses '0' = FBD link is not ready

14.3.1.17 RECALDUR: FBD Recalibrate Duration

This register determines the duration of the Recalibration state between 32 and 42 frames. During recalibration state all commands and CRC are ignored.

Device: NodeID
Function: 1
Offset: 70h

Bit	Attr	Default	Description
7	RV	0	*Reserved*
6:1	RW	0h	Recalibrate_Duration: This field sets the duration of the Recalibrate state once a sync command with the ERC bit. set is received. Legal values are between 32 and 42. Functionality if set outside this range is undefined. • > '42d (101010b) = undefined • = '42d (101010b) = ignore for 42 frames after Sync • .... • = 32d (100000b) = ignore for 32 frames after Sync • <32d (100000b) = undefined
0	RV	0	*Reserved*

14.3.1.18 SYNCTRAININT: SYNC Train Interval register

This register sets the typical spacing of sync commands on the link.

Device: NodeID
Function: 1
Offset: 78h

Bit	Attr	Default	Description
7:0	RWST	2Ah	SyncTrainInt: This field sets the min spacing for sync cmds default value 42d (101010b) min value 32, max value=255

14.3.1.19 SBCALSTATUS: Southbound Calibration Status

This register contains the pass/fail information of the last calibration on a per lane basis for the southbound. The AMB is expected to log the information when it is asked to go through calibration. The register is always cleared when entering the calibration state. This register will be used for debug purposes.

Device: NodeID
Function: 1
Offset: 7Ch

Bit	Attr	Default	Description
15:10	RV	0h	**Reserved**
9:0	RWST	0h	**CALSTATUS: Calibration status** 0 - Pass 1 - Fail

14.3.1.20 NBCALSTATUS: Northbound Calibration Status

This register contains the pass/fail information of the last calibration on a per lane basis for the southbound. The AMB is expected to log the information when it is asked to go through calibration. The register is always cleared when entering the calibration state. This register will be used for debug purposes.

Device: NodeID
Function: 1
Offset: 7Eh

Bit	Attr	Default	Description
15:14	RV	0h	**Reserved**
13:0	RWST	0h	**CALSTATUS: Calibration status** 0 - Pass 1 - Fail

14.3.2 SM Bus Register

Since the AMB will never be a master on the SM Bus and will never write to the EEPROM, SM Bus registers related to being a master are not required.

14.3.2.1 CBC: Chip Boot Configuration

This register reports AMB DIMM ID from pins BFUNC, SA[2:0]. SM Bus NodeID can be calculated from the DIMM ID as follows:

If DIMMID[3] = 0: (BFUNC =0) Normal DIMM

- NODEID[6:3] = "101_1" or "B"
  NODEID[2:0] = DIMMID[2:0] the value from pins SA[2:0]

If DIMMID[3] = 1: (BFUNC =1) Repeater or LAI

- NODEID[6:3] = "001_1" or "3"
  NODEID[2:0] = DIMMID[2:0] the value from pins SA[2:0]

Device: NodeID
Function: 1
Offset: 88h

Bit	Attr	Default	Description
7:4	RV	0	*Reserved*
3:0	RO	0	**DIMMID: SM Bus NodeID** Value from pins BFUNC, SA[2:0]

14.3.3 Error Registers

14.3.3.1 EMASK: Error Mask

This register masks errors in the FERR and NERR registers as well as disabling some types of error detection. A '0' in any field enables that error. A '1' in any field masks (disables) that error. Multiple bits can be set in this register. An enabled error sets error status, updates error logs, and generates link signals. A masked error does not affect error status, error logs, or link signals.

Device: NodeID
Function: 1
Offset: 8Ch

Bit	Attr	Default	Description
31:8	*RV*	*0*	***Reserved***
7	RWST	1	**Reserved**
6	RWST	1	**Reserved**
5	RWST	1	**INJERR: Error Injection has sourced an injected error bit in the status return field** (optional)
4	RWST	1	**INJALERT: Error Injection has sourced an injected alert error** (optional)
3	RWST	0	**FEWEDGES: tClk Training Violation (no sync cmd for 2x SYNCTRAININT - typically 84 frames)**
2	RWST	1	**OVERTEMP: Temp > TEMPHI and temp enabled**
1	RWST	1	**UNIMPLCFG: Unimplemented Configuration Address**
0	RWST	0	**CMDCRCERR: SB CRC Error**

14.3.3.2 FERR: First Error

This register contains bits specifying which errors occurred first as related to the FBD channel.

Device: NodeID
Function: 1
Offset: 90h

Bit	Attr	Default	Description
31:8	*RV*	*0*	***Reserved***
7	RWCST	0	**WBUFOVFL: Write Buffer overflow** - Implementation Specific - only supported for debug
6	RWCST	0	**WPLDERR: Wrong Number of Write Payloads (Buffer Underflow)** - Implementation Specific - only supported for debug
5	RWCST	0	**INJERR: Error Injection has sourced an injected error bit in the status return field** (optional)
4	RWCST	0	**INJALERT: Error Injection has sourced an injected alert error** (optional)
3	RWCST	0	**FEWEDGES: tClk Training Violation (no sync cmd for2x SYNCTRAININT - typically 84 frames)** Logs in **RECFBD** registers. Sends Alerts NB. Triggers auto self refresh SM.
2	RWCST	0	**OVERTEMP: Temp > TEMPHI and temp enabled** Fatal. AMB shuts down.
1	RWCST	0	**UNIMPLCFG: Unimplemented Configuration Address** Correctable. Logs in **RECCFG*** registers. AMB drops the command.

Device: NodeID Function: 1 Offset: 90h			
Bit	**Attr**	**Default**	**Description**
0	RWCST	0	**CMDCRCERR: SB CRC Error** Correctable. Logs in **RECFBD** registers. AMB drops the current command and sources alerts.

14.3.3.3 NERR: Successive Error

This register is used to report successive errors. More than two bits can be set in this register. This register contains bits specifying which errors occurred as related to the FBD channel.

Device: NodeID Function: 1 Offset: 94h			
Bit	**Attr**	**Default**	**Description**
31:8	*RV*	*0*	***Reserved***
7	RWCST	0	**Reserved**
6	RWCST	0	**Reserved**
5	RWCST	0	**INJERR: Error Injection has sourced an injected error bit in the status return field** (optional)
4	RWCST	0	**INJALERT: Error Injection has sourced an injected alert error** (optional)
3	RWCST	0	**FEWEDGES: tClkTraining Violation (no sync cmd for 2x SYNCTRAININT - typically 84 frames)** Logs in **RECFBD** registers. Sends Alerts NB. Triggers auto self refresh SM.
2	RWCST	0	**OVERTEMP: Temp > TEMPHI and temp enabled** Fatal. AMB shuts down.
1	RWCST	0	**UNIMPLCFG: Unimplemented Configuration Address** Correctable. Logs in **RECCFG*** registers. Drops the command.
0	RWCST	0	**CMDCRCERR: SB CRC Error** Correctable. Logs in **RECFBD** registers. Drops the current command and sources alerts.

14.3.3.4 RECCFG: Configuration Register Error Log

This register contains the received address for an unimplemented configuration register access error. The contents of this register are only valid when the error that set this register is logged in the FERR or NERR register.

Device: NodeID Function: 1 Offset: 98h			
Bit	**Attr**	**Default**	**Description**
15:13	RV	0	**Reserved**
12:10	RWST	0h	**function:**

Device: NodeID Function: 1 Offset: 98h			
Bit	**Attr**	**Default**	**Description**
9:8	RWST	0h	**size:** 00 = one byte 01 = two bytes 10 = three bytes 11 = four bytes **Note: This field only defined for errors on Config Register Writes. This field is undefined for other transactions.**
7:0	RWST	0h	**register address:**

14.3.3.5 RECFBD[9:0]: FBD Error Log

.This register contains FBD frame data received that matches the logged frame error.

Captures {BC} from frame N and A from frame N+1 where
[11:8] = A[n+1]
[7:4] = C[n] slot
[3:0] = B[n] slot

The contents of this register are only valid when one of the errors that set this register is logged in the FERR register. The contents of this register should not change until the error indication is cleared from the FERR register.

Device: NodeID Function: 1 Offset: AEh, ACh, AAh, A8h, A6h, A4h, A2h, A0h, 9Eh, 9Ch			
Bit	**Attr**	**Default**	**Description**
15:12	RV	0	**Reserved**
11:0	RWST	0	**FRMDATA: Frame data for lane n**

14.3.4 PERSONALITY BYTES Loaded From the SPD

These bytes allow for AMB implementation specific settings to be loaded in an architected way by BIOS without BIOS being aware of specific AMB requirements. Each AMB vendor defines how these bytes should be loaded for the specific DIMM being built. The values to be loaded into these bytes are stored in the SPD EEPROM on the DIMM.

The first six bytes are required to be loaded into the AMB via SMBus before link initialization to allow for configuration information needed for robust link operation.

The remaining 8 bytes must be loaded before the FBD begins normal operation.

Usage of these bytes can include

- DDR electrical parameters to optimize performance on a given DIMM
  - For example, DLL delay settings, various I/O driver slew settings,
- Enable/disable of various optimizations that may have been included in the design but can be turned off if they are not needed on this DIMM or they have unanticipated side effects - for example, power save modes, alternate clock recovery algorithms, and so forth.
- Temperature Sensor offsets

- Internal clock domain phase offsets

14.3.4.1 PERSBYTE[13:0]NXT: Personality Bytes

These bytes are loaded from SPD bytes 114:101 respectively.

(PERSBYTE13NXT = SPD byte 114, ... , PERSBYTE0NXT = SPD byte 101)

Function: 1
Offset: BDh:B0h

Bit	Attr	Default	Description
7:0	RWST	0	PData: Personality Data Byte Implementation specific registers

14.3.4.2 PERSBYTE[13:0]CUR: Personality Bytes

Function: 1
Offset: BDh:B0h

Bit	Attr	Default	Description
7:0	ROST	0	PData: Personality Data Byte Implementation specific registers

14.3.5 Hardware Configuration Registers

14.3.5.1 CMD2DATANXT: Next Value of Command to Data Delay

This register has the next value of the command to data delay. This will be used after the next fast reset. For correct DIMM operation, CMD2DATA may be limited to a subset of the architecturally valid values. The allowed values are AMB specific and may vary with frequency. Values come from SPD.

Device: NodeID
Function: 1
Offset: E8h

Bit	Attr	Default	Description
7:4	RWST	0h	**DLYFRMS: Number of frames** This specifies full frame delay part of the command to data delay. 0 - 9: Valid delays 10 - 15: Reserved
3:0	RWST	0h	**DLYFRAC: Fractional delay of command to data** This specifies fractional frame delay part of the command to data delay. 0 - 11: Specifies the delay in 1UI increments 12 - 15: Reserved

14.3.5.2 CMD2DATACUR: Current Value of Command to Data Delay

This register has the current value of the command to data delay.

Device: NodeID
Function: 1
Offset: E9h

Bit	Attr	Default	Description
7:4	ROST	0h	**DLYFRMS: Number of frames** This specifies full frame delay part of the command to data delay. 0 - 9: Valid delays 10 - 15: Reserved
3:0	ROST	0h	**DLYFRAC: Fractional delay of command to data** This specifies fractional frame delay part of the command to data delay. 0 - 11: Specifies the delay in 1UI increments 12 - 15: Reserved

14.3.5.3 C2DINCRNXT: Next Value of Command to Data Delay Increment

This register has the next value of the command to data incremental delay. This will be used after the next fast reset. This will be used by the last AMB to delay driving the data beyond that specified in the command to data delay.

Device: NodeID
Function: 1
Offset: EAh

Bit	Attr	Default	Description
7:2	RV	0h	*Reserved*
1:0	RWST	0h	**INCRDLY: Incremental Delay for command to data** 0 - 3: Specifies the incremental delay in frames

14.3.5.4 C2DINCRCUR: Current Value of Command to Data Delay Increment

This register has the current value of the command to data incremental delay. This will be used by the last AMB to delay driving the data beyond that specified.

Device: NodeID
Function: 1
Offset: EBh

Bit	Attr	Default	Description
7:2	RV	0h	*Reserved*
1:0	ROST	0h	**INCRDLY: Incremental Delay for command to data** 0 - 3: Specifies the incremental delay in frames

14.4 Implementation Specific FBD Registers (Function 2)

No register information is available for the implementation specific registers.

14.5 DDR and Miscellaneous Registers (Function 3)

14.5.1 Memory Registers

14.5.1.1 DAREFTC: DRAM Auto-Refresh Timing and Control

Device: NodeID
Function: 3
Offset: 70h

Bit	Attr	Default	Description
31	RV	0	**Reserved**
30	RWST	0	**REFDERR: refresh buffer overflow error**
29	RWST	0	**REFIERR: buffer count greater than the number of installed ranks**
28	RWST	0	**ORIDEHS: override handshake; auto-refresh wins arbitration for command bus**
27:24	RWST	0	**RBUF: number of pending refreshes for all ranks combined**
23:16	RWST	4Eh	**TRFC: DRAM refresh period**
15	RW	0	**AREFEN: auto-refresh enable**
14:0	RWST	0C30h	**TREFI: DRAM refresh interval**

14.5.1.2 DSREFTC: DRAM Self-Refresh Timing and Control

Device: NodeID
Function: 3
Offset: 74h

Bit	Attr	Default	Description
23:17	RV	0	**Reserved**
16	RWST	1	**DISSREXIT**: Disable DRAM Self-Refresh Exit when the link comes up
15:8	RWST	56h	**TXSNR**: DRAM Self-Refresh Exit to Non-Read Command Timing
7:4	RWST	Fh	**TRP: DRAM Precharge Timing**
3	RV	0	**Reserved**
2:0	RWST	7h	**TCKE**: DRAM Minimum CKE Pulse Width

14.5.1.3 MTR: Memory Technology Register

This register provides a local definition of the organization of DIMMs. This DRAM configuration information is used for MemBIST and DDR calibration.

Device: NodeID
Function: 3
Offset: 77h

Bit	Attr	Default	Description
7	RV	0	*Reserved*
6	RWST	0	WIDTH: Technology – DRAM data width Define the data width of SDRAMs within these DIMM's 0 = x4 (4 bits wide) 1 = x8 (8 bits wide)

Device: NodeID Function: 3 Offset: 77h			
Bit	**Attr**	**Default**	**Description**
5	RWST	0	NUMRANK: Technology – Number of Ranks Define the number of ranks within these DIMM's 0 = single ranked 1 = double ranked
4	RWST	0	NUMBANK: Technology – Number of Banks Define the number of banks within these DIMM's 0 = 4 banks 1 = 8 banks
3:2	RWST	00	NUMROW: Technology – Number of Rows Define the number of rows within these DIMM's 00 = 8,192 01 = 16,384 10 = 32,768 11 = 65,536
1:0	RWST	00	NUMCOL: Technology – Number of Columns Define the number of columns within these DIMM's 00 = 1,024 01 = 2,048 10 = 4,096 11 = 8,192

14.5.1.4 DRT: DRAM Timing Control

The DRAM Timing Control register is used to setup timing for MemBIST access to DRAMs.

Device: NodeID Function: 3 Offset: 78h			
Bit	**Attr**	**Default**	**Description**
31	RV	0	*Reserved*
30:29	RWST	00	TRAS: DRAM tRAS minimum required delay from active command to precharge command. Delay cycles based on JEDEC DDRII spec 45 ns for DDRII 400/533/667. Based on the latest JEDEC spec (JESD79-2, Sept 2003) for DDRII 800 MHz min tRAS is not defined yet. tRASMIN clocks delay: 00 => 18 for DDRII 800 MHz 01 => 15 for DDRII 667 MHz 10 => 12 for DDRII 533 MHz 11 => Reserved
28:27	RWST	00	TRTP: DRAM cell internal read to precharge command delay. tRTP clocks delay: 00 => 2 01 => 3 10 => 4 11 => 5

Device: NodeID
Function: 3
Offset: 78h

Bit	Attr	Default	Description
26:24	RWST	000	BBRW: Back to Back Read-Write turn around. This field determines the minimum number of CMDCLK between Read-Write commands. The purpose of these 3 bits are to control the turnaround time on the DQ bus. Regular setting will be based on BL/2 + 2 tCK. BL4: tR2W = 4 tCK BL8: tR2W = 6 tCK Command clocks apart based on the following encoding: 000 => 10 001 => 9 010 => 8 011 => 7 100 => 6 101 => 5 110 => 4 111 => 3 (stress mode, not recommended)
23	RV	0	*Reserved*
22:20	RWST	000	BBWR: Back to Back Write-Read turn around. This field determines the minimum number of CMDCLK between Write-Read commands. The purpose of these 3 bits are to control the turnaround time on the DQ bus. Regular setting will be based on (CL-1)+BL/2+tWTR. Command clocks apart based on the following encoding: 000 => 12 001 => 11 010 => 10 011 => 9 100 => 8 101 => 7 110 => 6 111 => 5 (stress mode, not recommended
19	RV	0	*Reserved*
18:16	RWST	000	TWR: Twr DRAM Write Recovery delay Overall delay clocks will be (CL+AL-1) +BL/2 + tWR from write command to precharge command. 000 => 9 001 => 8 010 => 7 011 => 6 100 => 5 101 => 4 110 => 3 111 => 2

Device: NodeID Function: 3 Offset: 78h			
Bit	**Attr**	**Default**	**Description**
15:12	RWST	0000	TRC: Trc DRAM activate to another activate delay 0000 => 26 0001 => 25 0010 => 24 0011 => 23 0100 => 22 0101 => 21 0110 => 20 0111 => 19 1000 => 18 1001 => 17 1010 => 16 1011 => 15 1100 => 14 1101 => 13 1110 => 12 1111 => 11
11:10	RWST	00	TRCD: Trcd DRAM RAS# to CAS# delay 00 => 6 01 => 5 10 => 4 11 => 3 If AL >= Trcd, Read/Write command will be issued right after ACT cycle.
9:8	RWST	00	TRP: Trp DRAM RAS# to Precharge delay 00 => 6 01 => 5 10 => 4 11 => 3
7:0	RWST	00h	NOPCNT: Programmable NOP insertion (Device Deselect actually). Number of Nops will be inserted between read/write commands to slow down Membist activities in the same page. Up to 255 clocks NOPs can be programmed to insert delay between read/write commands. If NOPs delay is programmable less than the required DRAM timing, Overall NOP delay from command to command will not be seen.

14.5.1.5 DRC: DRAM Controller Mode Register

This register controls the mode of the DRAM Controller.

Device: NodeID Function: 3 Offset: 7Ch			
Bit	**Attr**	**Default**	**Description**
31:30	RV	00	*Reserved*
29	RW	0	**INITDONE: Initialization Complete.** This scratch bit communicates software state from the AMB to BIOS. BIOS sets this bit to 1 after initialization of the DRAM memory array is complete. This bit has no effect on AMB operation.
28	RV	0	**Reserved**

Device: NodeID
Function: 3
Offset: 7Ch

Bit	Attr	Default	Description
27:24	RWST	0	**CLKDIS: clock[3:0] output disable**
23	RWST	0	**SEQADDR:** When set to 1 turns off address balancing on address bit A0 to support DRAMs programmed for Sequential Burst Type
22	RWST	0	**DQSHALFGAIN: - select for DQS differential amplifier gain. When set to 0 the amplifier gain is cut half to support differential strobes for DDR2** **Note: the sense of this field is inverted from past DDR designs so that BIOS supporting generic AMBs do not have to write a "1" to what is a "reserved" field on other AMBs**
21	RW	0	**TESTMODE:** When set to 1 the LEGSEL output of the DDR comp block selects one of eight driver legs to enable. This bit can be used in conjunction with the DRAMISCTL.DRVOVR bits to override the LEGSEL output generated by the comp block.
20	RWST	0	
19	RWST	0	**RWPRDIS: Read/Write pointer reset disable** Disables the resetting of DDR cluster FIFO read and write pointers during normal operation that occurs when a READ command finishes executing and no additional READ commands are in process.
18	RWST	1	**ODTZ: On-Die Termination Strength**. "0" Disabled "1" Enabled
17	RWST	0	**HLDDIS**: command/address hold disable Disabling hold will allow the address and bank address pins to revert to all zeros (all ones on the balanced address copy) during idle cycles. When hlddis is clear, the addresses retain the value of the last non-idle command cycle in order to reduce switching on the bus.
16	RWST	0	**BALDIS**: command/address balancing disable
15	RW	0	**CADIS**: command/address output disable
14	RW	0	**CSDIS**: chip select output disable
13	RW	0	**ODTDIS**: ODT output disable
12	RWST	1	**CKEFRCLOW**: CKE Force Low Forces CKE low. Must be cleared to enable normal DDR functionality. This bit overrides the CKE1 and CKE0 fields described below, and also overrides all channel commands and other hardware functions that would otherwise affect the state of the CKE outputs.
11	RW	0	**CKEDIS**: CKE output disable
10	RWST	0	**CKE1:** CKE output 1 control and status. Software can write to this bit to change the state of the CKE 1 output. Hardware will update this bit with the current status of the CKE1 output two core cycles after a channel command or other hardware function changes the state of the CKE1 output. '1' = CKE1 pads asserted. '0' = CKE1 pads de-asserted.
9	RWST	0	**CKE0:** CKE output 0 control and status. Software can write to this bit to change the state of the CKE 0 output. Hardware will update this bit with the current status of the CKE 0 output two core cycles after a channel command or other hardware function changes the state of the CKE 0 output. '1' = CKE0 pads asserted. '0' = CKE0 pads de-asserted.
8	RWST	0	**BL**: DRAM burst length. '1' = bl8 '0' = bl4
7:4	RWST	2h	**AL**: DRAM Additive Latency [3:0] Note: This AL value is sampled during TS2 training sequences to set timing for FBD link data returns. Changes to AL after this time will not effect FBD link data return timing until the next TS2 sequence following link reset.

Device: NodeID Function: 3 Offset: 7Ch			
Bit	**Attr**	**Default**	**Description**
3:0	RWST	3h	**CL**: DRAM CAS Latency [3:0] Note: This CL value is sampled during TS2 training sequences to set timing for FBD link data returns. Changes to CL after this time will not effect FBD link data return timing until the next TS2 sequence following link reset.

14.5.2 Memory BIST Registers

14.5.2.1 MBCSR: MemBIST Control

Architected MemBIST control interface.

Device: NodeID Function: 3 Offset: 40h			
Bit	**Attr**	**Default**	**Description**
31	RWS	0	START: Start operation: 1 => Set this bit to begin MemBIST execution. 0 => Hardware will clear this bit when MemBIST execution is completed.
30	RW	0	PF: Fail/Pass indicator: Write to 0 when start MemBIST. Hardware will set to 1 when a failure is detected. 0 => Pass 1 => Fail
29	RW	0	HALT: Halt on Error 0 => Operation will not halt due to a detected error. 1 => Operation will halt after read-compare data error is detected. MemBIST will complete the current transaction before halting. This may result in multiple errors being logged.
28	RW	0	ABORT: MemBIST test abort. When test abort bit is set, MBCSR bit 31 (Start operation, RWS) needs to be set to "0" at the same time to avoid restarting MemBIST. 0 => Normal operation. 1 => Need to abort the test during MemBIST operation. If there is any following membist test after the abort test, bit [28] needs to be cleared. The Write to set MBCSR.abort must occur at least tRFC after the Write to set MBCSR.start. Otherwise subsequent MemBIST operations may fail. tRFC value is set in DAREFTC.trfc (Function3, offset70h, bit field 23:16).
27	RW	0	SPARE:

Device: NodeID
Function: 3
Offset: 40h

Bit	Attr	Default	Description
26:24	RW	000	ALGO: Embedded Algorithm selection: Embedded Algorithm selection: 000 => No embedded algorithm is selected. Normal command will be executed from the selection of MBCSR bits field [5:4] 001 => Scan: ^ (WD1); ^(RD2); ^ (WI3); ^ (RI4) 010 => Undefined 011 => Data Retention Write or Init: ^ (WD1); 100 => Data Retention Read : ^ (RD2); 101 => Mats +: ^(WD1); ^(RD2, WI3); v(RI4, WD5); 110 => March C-: ^(WD1); ^(RD2, WI3); ^(RI4, WD5); v(RD6, WI7); v(RI8, WD9); v(RD10); 111 => Undefined
23:22	RV	00	Reserved
21:20	RW	00	CS: CS[1:0] selection in MemBIST mode 01: select Rank 0 10: select Rank 1 00: Reserved 11: Reserved
19	RW	0	INVERT: Invert data pattern when data is written out to DRAM.
18:16	RW	000	FIXED: Fixed data pattern selection for MemBIST operation 000 => 0 001 => F 010 => A 011 => 5 100 => C 101 => 3 110 => 9 111 => 6
15	RW	0	ENABLE288: Enable 288 bits user defined pattern for memory fill write only. There is no data comparison, error logger functions for 288 bits user defined data. 0 => 144 bits user defined data pattern when MBCSR[9:8] selects user defined data. 1 => 288 bits user defined data pattern when MBCSR[9:8] selects user defined data.
14	RW	0	MBDATA: Selects use of MBDATA for error log field for LFSR, Circular Shift and user defined data modes. This field has no effect on fixed data patterns. 0 => use MBDATA0/1/2/3/8 for failure data bit location accumulator. 1 => use MBDATA0/1/2/3/8 to log 5 failure addresses.
13	RW	0	ABAR: MemBIST output address compliment for FastX, FastY, and FastXY. Whenever this bit is enabled, Bank, Row, Column address will be inverted on alternate addresses as described in the MemBIST chapter. 0 => Regular addressing 1 => Dynamic address inversion
12	RW	0	ADIR: Address decode direction for FastX, Fast Y, FastXY 0 => Address increments 1 => Address decrements

Device: NodeID Function: 3 Offset: 40h			
Bit	**Attr**	**Default**	**Description**
11:10	RW	00	FAST: Address sequencing 00 => addressing with XZY toggling (column->bank->row) 01 => Fast Y with fixed bank 10 => Fast X with fixed bank 11 => Fast XY with fixed bank
9:8	RW	00	DTYPE: Data type selection: 00 => Fixed data pattern, selected by MBCSR bits 18:16 01 => 144 or 288 bits user defined data 10 => Circular shift data 11 => LFSR data, seeded from 32 bit LFSR seed register. Note: Algorithm mode only supports DTYPE = Fixed Note: Circular shift data and LFSR data type should not be used for single address operation (ATYPE = 01).
7:6	RW	00	ATYPE: Address type 00 => Reserved 01 => Single physical address operation, contained in MBADDR row/column/ bank. 10 => start/end physical address range defined in MB_START_ADDR & MB_END_ADDR registers. • In FastX, FastY and FastXY modes, only the bank specified in MB_START_ADDR will be exercised. 11 => full address range of the DIMM as defined in MTR register which specifies the number of banks, rows, and columns. • In FastX, FastY and FastXY modes, only the bank 0 will be exercised.
5:4	RW	00	CMD: Command execution: 00 => Read only without data comparison 01 => Write only 10 => Read with data comparison 11 => Write followed by Read with data comparison
3:0	RV	0	Reserved

Algorithms:

When Embedded algorithm is applied, please program the following bits at the same time.

1. Select MBCSR.cmd bit[5:4] for the initial command execution mode.
   For all algorithm choices except for Data Retention Read, select "01: write only".
   For Data Retention Read, select "10: read with data comparison".
2. Program MBCSR.fast bit[11:10] to select FastX, FastY, FastXY, or XZY.
3. Program proper start/end address registers and corresponding MTR value for DIMM type. Do not leave start and end address register as default "00" or the same value. Algorithm does not support single or full address modes.

14.5.2.2 MBADDR: Memory Test Address

The register is used by MemBIST only when testing to a single memory location.

(MBCSR.atype = 2b‘01)

Device: NodeID Function: 3 Offset: 44h			
Bit	**Attr**	**Default**	**Description**
31:16	RWST	0000h	ROW: Row Address 15:0
15	RWST	0	SPARE:
14:3	RWST	0000h	COL: Column Address BL8[14:3] <==> DRAM Column Address 15:11,9:3 BL4[14:3] <==> DRAM Column Address 14:11,9:2
2:0	RWST	000	BA: Bank Address 2:0

14.5.2.3 MBDATA[9:0]: Memory Test Data

Device: NodeID Function: 3 Offset: 6Ch, 68h, 64h, 60h,5Ch, 58h, 54h, 50h,4Ch, 48h			
Bit	**Attr**	**Default**	**Description**
31:0	RWST	0000h	DATA: see functional description below for definition by mode and register

Reg	Bit	Offset	Description by mode (note: MBCSR.dtype, MBCSR.mbdata and MBCSR.enable288 select mode)				
			Fixed Data Pattern	**144 bit User Defined Pattern**	**Circular Shift**	**LFSR**	**288 bit User defined pattern**
MBDATA9	31:0	6Ch	5th Fail address	User defined Late data [71:64] & Early data [71:64]	Word4 Circular shift data	LFSR random Late data [71:64] & Early data [71:64]	User defined Late data [71:64] (2nd burst data) & Early data [71:64] (2nd burst data)
MBDATA8	31:0	68h	Late data [71:64] & Early data [71:64] Failure bit location accumulator	5th Fail address Or Late data [71:64] & Early data [71:64] Failure bit location accumulator	5th Fail address Or Late data [71:64] & Early data [71:64] Failure bit location accumulator	5th Fail address Or Late data [71:64] & Early data [71:64] Failure bit location accumulator	User defined Late data [71:64] (1st burst data) & Early data [71:64] (1st burst data)
MBDATA7	31:0	64h	Fail address 4	User defined Late data [63:32]	DW3 Circular shift data	LFSR random Late data [63:32]	User defined Late data [63:32] (2nd burst data)
MBDATA6	31:0	60h	Fail address 3	User defined Late data [31:0]	DW2 Circular shift data	LFSR random Late data [31:0]	User defined Late data [31:0] (2nd burst data)
MBDATA5	31:0	5Ch	Fail address 2	User defined Early data [63:32]	DW1 Circular shift data	LFSR random Early data [63:32]	User defined Early data [63:32] (2nd burst data)
MBDATA4	31:0	58h	Fail address 1	User defined Early data [31:0]	DW0 Circular shift data	LFSR random Early data [31:0]	User defined Early data [31:0] (2nd burst data)
MBDATA3	31:0	54h	Late data [63:32] Failure bit location accumulator	Fail address 4 Or Late data [63:32] Failure bit location accumulator	Fail address 4 Or Late data [63:32] Failure bit location accumulator	Fail address 4 Or Late data [63:32] Failure bit location accumulator	User defined Late data [63:32] (1st burst data)

Reg	Bit	Offset	Description by mode (note: MBCSR.dtype, MBCSR.mbdata and MBCSR.enable288 select mode)				
			Fixed Data Pattern	144 bit User Defined Pattern	Circular Shift	LFSR	288 bit User defined pattern
MBDATA2	31:0	50h	Late data [31:0] Failure bit location accumulator	Fail address 3 Or Late data [31:0] Failure bit location accumulator	Fail address 3 Or Late data [31:0] Failure bit location accumulator	Fail address 3 Or Late data [31:0] Failure bit location accumulator	User defined Late data [31:0] (1st burst data)
MBDATA1	31:0	4Ch	Early data [63:32] Failure bit location accumulator	Fail address 2 Or Early data [63:32] Failure bit location accumulator	Fail address 2 Or Early data [63:32] Failure bit location accumulator	Fail address 2 Or Early data [63:32] Failure bit location accumulator	User defined Early data [63:32] (1st burst data)
MBDATA0	31:0	48h	Early data [31:0] Failure bit location accumulator	Fail address 1 Or Early data [31:0] Failure bit location accumulator	Fail address 1 Or Early data [31:0] Failure bit location accumulator	Fail address 1 Or Early data [31:0] Failure bit location accumulator	User defined Early data [31:0] (1st burst data)

Note: In the later half part of data burst length 8 test, 144 bits or 288 bits user-defined data pattern will be repeat as the same sequence of burst length 4.

14.5.2.3.1 MBDATA Failure Address Mapping

To compress the failure address into 32 bits, bits that are always zero are removed from the logging. These removed bits include AutoPrecharge Column address [10] and least significant bits assumed by burst length.

Table 14-11. MBDATA Failure Address Register Correspondence to DRAM 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
2	1	0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	see description below												
Bank			Row																Column and Chunk												

BL4: 1 bit chunk indicates the location of 2 failure burst data chunks.

The above Column plus Chunk is equal to DRAM column address as the following:

Table 14-12. BL4 Column and Chunk Correspondence to DRAM Address

Register Bit Location	12	11	10	9	8	7	6	5	4	3	2	1	0	
DRAM Col Address	14	13	12	11	9	8	7	6	5	4	3	2		
Data Chunk													1	X

- where the auto-precharge address bit 10 is assumed zero
- since data is logged in 144 bits (two chunks), address bit zero is not needed

BL8: 2 bit chunk indicates the location of 4 failure burst data chunks.

The above Column plus Chunk is equal to DRAM column address as the following:

Table 14-13. BL8 Column and Chunk Correspondence to DRAM Address

Register Bit Location	12	11	10	9	8	7	6	5	4	3	2	1	0	
DRAM Col Address	14	13	12	11	9	8	7	6	5	4	3			
Data Chunk												2	1	X

- where the auto-precharge address bit 10 is assumed zero
- since data is logged in 144 bits (two chunks), Data chunk address bit zero is not needed

14.5.2.4 MB_START_ADDR: Memory Test Start Address

MB_END_ADDR row and column address must be larger than MB_START_ADDR row and column address in either increasing or deceasing address mode.

During FastX, FastY and FastXY operation, only one memory bank is tested. Specify the desired bank in MB_START_ADDR[2:0]. MB_END_ADDR[2:0] is ignored.

This register is only used when MBCSR.atype = 2b'10, and when MBCSR.algo is non-zero.

Device: NodeID
Function: 3
Offset: 9Ch

Bit	Attr	Default	Description
31:16	RWST	0000h	ROW: MemBIST Start Row Address 15:0
15	RV	0	Reserved
14:3	RWST	0000h	COL: MemBIST Start Column Address BL8 [14:3] <==> Column Address 15:11, 9:3 BL4 [14:3] <==> Column Address 14:11, 9:2
2:0	RWST	000	BA: MemBIST Start Bank Address 2:0

14.5.2.5 MB_END_ADDR: Memory Test End Address

This register is only used when MBCSR.atype = 2b'10, and when MBCSR.algo is non-zero.

Device: NodeID
Function: 3
Offset: A0h

Bit	Attr	Default	Description
31:16	RWST	0000h	ROW: MemBIST End Row Address 15:0
15	RV	0	Reserved
14:3	RWST	0000h	COL: MemBIST End Column Address BL8 [14:3] <==> Column Address 15:11, 9:3 BL4 [14:3] <==> Column Address 14:11, 9:2
2:0	RWST	000	BA: MemBIST End Bank Address 2:0

14.5.2.6 MBLFSRSED: Memory Test Circular Shift and LFSR Seed

Device: NodeID
Function: 3
Offset: A4h

Bit	Attr	Default	Description
31:0	RWST	0000h	DMBLFSRSED:MemBIST LFSR Seed This 32 bit register will be used as the initial data seed for LFSR or Circular shift data pattern.

14.5.2.7 MBFADDRPTR: Memory Test failure Address Pointer Register

Device: NodeID
Function: 3
Offset: A8h

Bit	Attr	Default	Description
31:0	RWST	0000h	DMBFADDRPTR: This 32 bit register designates which MemBIST failures to log in the available failure address locations. The default value of this register is zero. It means MemBIST always logs beginning with the first failure. If it is programmed to hex A (10 in decimal), MemBIST will log failures starting from the11th failure. The corresponding MB_ERR_DATA0 register will log corrupted data in the first designated failure address. Note: this register does not affect the MBDATA failure bit location accumulators.

14.5.2.8 MB_ERR_DATA00: Memory Test Error Data 0 Bytes [3:0]

Stores the first 32 bits of the 1st 144 bit failure data Research: compare all register[definitions to table.]

Device: NodeID
Function: 3
Offset: B0h

Bit	Attr	Default	Description
31:0	RWST	0000h	DATA: Early failure data [31:0]

14.5.2.9 MB_ERR_DATA01: Memory Test Error Data 0 Bytes [7:4]

Stores the second 32 bits of the 1st 144 bit failure data.

Device: NodeID
Function: 3
Offset: B4h

Bit	Attr	Default	Description
31:0	RWST	0000h	DATA: Early failure data [63:32]

14.5.2.10 MB_ERR_DATA02: Memory Test Error Data 0 Bytes [11:8]

Stores the third 32 bits of the 1st 144 bit failure data.

Device: NodeID
Function: 3
Offset: B8h

Bit	Attr	Default	Description
31:0	RWST	0000h	DATA: Late failure data [31:0]

14.5.2.11 MB_ERR_DATA03: Memory Test Error Data 0 Bytes [15:12]

Stores the fourth 32 bits of the 1st 144 bit failure data.

Device: NodeID
Function: 3
Offset: BCh

Bit	Attr	Default	Description
31:0	RWST	0000h	DATA: Late failure data [63:32]

14.5.2.12 MB_ERR_DATA04: Memory Test Error Data 0 Bytes [17:16]

Stores the last 16 bits of the 1st 144 bit failure data.

Device: NodeID
Function: 3
Offset: C0h

Bit	Attr	Default	Description
15:0	RWST	0000h	DATA: Late failure data [71:64] & Early failure data [71:64]

14.5.2.13 MB_ERR_DATA10: Memory Test Error Data 1 Bytes [3:0]

Stores the first 32 bits of the 2nd 144 bit failure data.

Device: NodeID
Function: 3
Offset: C4h

Bit	Attr	Default	Description
31:0	RWST	0000h	DATA: Early failure data [31:0]

14.5.2.14 MB_ERR_DATA11: Memory Test Error Data 1 Bytes [7:4]

Stores the second 32 bits of the 2nd 144 bit failure data.

Device: NodeID
Function: 3
Offset: C8h

Bit	Attr	Default	Description
31:0	RWST	0000h	DATA: Early failure data [63:32]

14.5.2.15 MB_ERR_DATA12: Memory Test Error Data 1 Bytes [11:8]

Stores the third 32 bits of the 2nd 144 bit failure data.

Device: NodeID
Function: 3
Offset: CCh

Bit	Attr	Default	Description
31:0	RWST	0000h	DATA: Late failure data [31:0]

14.5.2.16 MB_ERR_DATA13: Memory Test Error Data 1 Bytes [15:12]

Stores the fourth 32 bits of the 2nd 144 bit failure data.

Device: NodeID
Function: 3
Offset: D0h

Bit	Attr	Default	Description
31:0	RWST	0000h	DATA: Late failure data [63:32]

14.5.2.17 MB_ERR_DATA14: Memory Test Error Data 1 Bytes [17:16]

Stores the last 16 bits of the 2nd 144 bit failure data.

Device: NodeID
Function: 3
Offset: D4h

Bit	Attr	Default	Description
15:0	RWST	0000h	DATA: Late failure data [71:64] & Early failure data [71:64]

14.5.2.18 MB_ERR_DATA20: Memory Test Error Data 2 Bytes [3:0]

Stores the first 32 bits of the 3rd 144 bit failure data.

Device: NodeID
Function: 3
Offset: D8h

Bit	Attr	Default	Description
31:0	RWST	0000h	DATA: Early failure data [31:0]

14.5.2.19 MB_ERR_DATA21: Memory Test Error Data 2 Bytes [7:4]

Stores the second 32 bits of the 3rd 144 bit failure data.

Device: NodeID
Function: 3
Offset: DCh

Bit	Attr	Default	Description
31:0	RWST	0000h	DATA: Early failure data [63:32]

14.5.2.20 MB_ERR_DATA22: Memory Test Error Data 2 Bytes [11:8]

Stores the third 32 bits of the 3rd 144 bit failure data.

Device: NodeID
Function: 3
Offset: E0h

Bit	Attr	Default	Description
31:0	RWST	0000h	DATA: Late failure data [31:0]

14.5.2.21 MB_ERR_DATA23: Memory Test Error Data 2 Bytes [15:12]

Stores the fourth 32 bits of the 3rd 144 bit failure data.

Device: NodeID
Function: 3
Offset: E4h

Bit	Attr	Default	Description
31:0	RWST	0000h	DATA: Late failure data [63:32]

14.5.2.22 MB_ERR_DATA24: Memory Test Error Data 2 Bytes [17:16]

Stores the last 16 bits of the 3rd 144 bit failure data.

Device: NodeID
Function: 3
Offset: E8h

Bit	Attr	Default	Description
15:0	RWST	0000h	DATA: Late failure data [71:64] & Early failure data [71:64]

14.5.2.23 MB_ERR_DATA30: Memory Test Error Data 3 Bytes [3:0]

Stores the first 32 bits of the 4th 144 bit failure data.

14.5.2.24 MB_ERR_DATA31: Memory Test Error Data 3 Bytes [7:4]

Stores the second 32 bits of the 4th 144 bit failure data

Device: NodeID
Function: 3
Offset: F0h

Bit	Attr	Default	Description
31:0	RWST	0000h	DATA: Early failure data [63:32]

14.5.2.25 MB_ERR_DATA32: Memory Test Error Data 3 Bytes [11:8]

Stores the third 32 bits of the 4th 144 bit failure data.

Device: NodeID
Function: 3
Offset: F4h

Bit	Attr	Default	Description
31:0	RWST	0000h	DATA: Late failure data [31:0]

14.5.2.26 MB_ERR_DATA33: Memory Test Error Data 3 Bytes [15:12]

Stores the fourth 32 bits of the 4th 144 bit failure data.

Device: NodeID
Function: 3
Offset: F8h

Bit	Attr	Default	Description
31:0	RWST	0000h	DATA: Late failure data [63:32]

14.5.2.27 MB_ERR_DATA34: Memory Test Error Data 3 Bytes [17:16]

Stores the last 16 bits of the 4th 144 bit failure data.

Device: NodeID
Function: 3
Offset: FCh

Bit	Attr	Default	Description
15:0	RWST	0000h	DATA: Late failure data [71:64] & Early failure data [71:64]

14.5.3 Thermal Sensor Registers

14.5.3.1 TEMPLO: Temperature Low Trip Point

Low trip point.

Device: NodeID
Function: 3
Offset: 80h

Bit	Attr	Default	Description
7:0	RWST	FFh	**TEMPLO: Low threshold trip point**

14.5.3.2 TEMPMID: Temperature Mid Trip Point

Mid trip point.

Device: NodeID
Function: 3
Offset: 81h

Bit	Attr	Default	Description
7:0	RWST	FFh	**TEMPMID: Mid threshold trip point**

14.5.3.3 TEMPHI: Temperature High Trip Point

High trip point.

Device: NodeID
Function: 3
Offset: 82h

Bit	Attr	Default	Description
7:0	RWST	FFh	**TEMPHI: High threshold trip point**

14.5.3.4 UPDATED: Update Temp Diff Bit

Take new temperature sample and update the temp diff bit (INCREASING).

Device: NodeID
Function: 3
Offset: 83h

Bit	Attr	Default	Description
7:1	*RV*	*00h*	***reserved***
0	RWS	0	**UPDATE:** **Write '1' =** **a. latch current temperature from the TEMP register for comparison on next UPDATE and** **b. update INCREASING bit of TEMPSTAT register based on the current temperature and the last latched temperature;** **c. Automatically clears this bit to '0' after INCREASING bit is updated;** **Write '0' - no effect**

14.5.3.5 TEMPSTAT: Thermal Sensor Status Register

This register controls and reports temperature status.

Device: NodeID
Function: 3
Offset: 84h

Bit	Attr	Default	Description
7:6	RV	0	*Reserved*
5	RWST	0	NoAutoUpdate '1' = turns off update of temp stat values so that forced values written in by firmware are not overwritten '0' = overtemp trip bits are continuously and automatically updated as normal and increasing bit is updated through UPDATE register mechanism
4	RW	0	**INCREASING** '1' = Temperature has increased since the last time UPDATE bit was set in UPDATED register. '0' = Temperature has not increased since the last time UPDATE bit was set in UPDATED register. This is reflected as the Rising/Falling value in the Thermal Trip field of northbound FBD status 0.
3	RWST	0	**OVERTEMPHI** '1' = Temperature is above or equal to TEMPHI.TRIP '0' = Temperature is below TEMPHI.TRIP
2	RWST	0	**OVERTEMPMID** '1' = Temperature is above or equal to TEMPMID.TRIP '0' = Temperature is below TEMPMID.TRIP This is reflected in the Thermal Trip value of northbound FBD status 0
1	RWST	0	**OVERTEMPLO** '1' = Temperature is above or equal to TEMPLO.TRIP '0' = Temperature is below TEMPLO.TRIP This is reflected in the Thermal Trip value of northbound FBD status 0
0	RWST	0	**TEMPHIENABLE** '1' = Allow OVERTEMPHI=1 to shut down the DDR channel, drop DDR commands, log an error, and take FBD links to EI. '0' = OVERTEMPHI=1 only logs an error.

14.5.3.6 TEMP: Temperature A/D

Current temperature reading. This 8-bit value with 0.5 degree of resolution is continuously and automatically kept up to date by AMB hardware.

Device: NodeID
Function: 3
Offset: 85h

Bit	Attr	Default	Description
7:0	RO	00h	**DEGREES: Current temperature - binary encode 0 - 127.5 degrees C**

14.6 Implementation Specific DDR Initialization and Calibration Registers (Function 4)

14.6.1 DDR Calibration

14.6.1.1 DCALCSR: DCAL Control and Status

Device: NodeID
Function: 4
Offset: 40h

Bit	Attr	Default	Description
31	RWS	0	**START: Start Operation** When set to 1 by software, the operation selected by the dcalcsr.opcode is initiated. Hardware clears this bit when the operation is complete.
30:28	RW	0	**FAIL: Completion Status** 1xx = Fail, 0xx = Pass
27	RW	0	**BASPAT:** This controls which data pattern is used for the DQS Delay calibration. Setting this field enables the use of the basic data pattern selected by the DCALCSR.PATTERN bits. When cleared, the extended data pattern specified in the DDQSCVDP and DDQSCADP registers is used.
26	RW	00	RSTREGSS: Reset DCALDATA CSR in single step calibration mode. This bit should be set during the first step of a single step calibration. It will enable hardware to clear all registers and status bits during the calibration step the same way hardware does on the first step of an automatic “all passes” calibration.
25:24	RW	0	**CHSEL:** This field defines bus folding. This function is obsolete and is not supported.
23	RW	0	**SGLSTP:** Single Step Calibration Operation Applies only to Receive enable, DQS cal, and I/O loopback “1” = Single step - a single step of the algorithm selected by the OPCODE is run by hardware. No data analysis is run. “0” = All passes - all steps of the algorithm selected by the OPCODE is run by hardware including data analysis.
22:21	RW	00	**CS: Rank select** This field corresponds to the chip select outputs: CS[1:0]. Setting a bit in this field will cause the corresponding CS pin to drive low when commands are issued on the DDR bus. This field Applies to NOP, Refresh, Precharge all, and MRS/EMRS commands. It also applies to Receive Enable, and DQS Delay cal in single step mode.
20	RV	0	*Reserved*
19	RW	0	**A0_DQSCAL***: revert to A0 DQSCAL algorithm*

Device: NodeID Function: 4 Offset: 40h			
Bit	**Attr**	**Default**	**Description**
18:16	RW	000	**PATTERN:** Basic data pattern for DQS cal and I/O loopback. This sets the burst length 4 pattern for a nibble of data. The pattern is repeated for BL8. This pattern is replicated on all nibbles of the data bus. "000" = F > 0 > F > 0 "001" = 0 > F > 0 > F "010" = A > 5 > A > 5 "011" = 5 > A > 5 > A "100" = C > 3 > C > 3 "101" = 3 > C > 3 > C "110" = 9 > 6 > 9 > 6 "111" = 6 > 9 > 6 > 9
15	RW	0	**DARWPR:** Disable FIFO reset in single pass mode. Applies only to Receiver enable, DQS cal, and I/O loopback. When set to 1, this bit inhibits the core to DDR cluster reset signal generated during the cal/hvm modes listed above. This prevents the DDR cluster synchronizer FIFO write pointer and data latches from being reset so that they can be read out of the cluster using the error monitor function. The reset signal can only be disabled in single step mode. When the ALLP bit is set to 1, the DARWPR bit has no effect.
14:4	RW	000h	**OPMODS:** Operation modifiers
3:0	RW	0h	**OPCODE:** "0000" = NOP "0001" = Refresh "0010" = Pre-Charge "0011" = MRS/EMRS "0101" = Automatic DQS Delay Calibration "1100" = Automatic Receive Enable Calibration "1101" = Self-Refresh Entry All other settings are reserved

14.6.1.2 DCALADDR: DCAL Address Register

Device: NodeID Function: 4 Offset: 44h			
Bit	**Attr**	**Default**	**Description**
31:0	RW	0000_0000h	**DCALADDR: DCAL Address and Other Information Based on Opcode.**

Table 14-14. Functional Characteristics of DCALADDR

Bit	NOP, Refresh, Pre-Charge, MRS/EMRS, and Self-Refresh Entry Commands initiated by DCALCSR
31	DRAM Address Bus 15:0
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	DRAM Bank Address Bus 2:0
1	
0	

14.6.1.3 DCALDATA[71:0]: DCAL Data Registers

Device: NodeID Function: 4 Offset: 8Fh - 48h			
Bit	**Attr**	**Default**	**Description**
7:0	RW	00h	**DCALDATA: DCAL Data and Other Information Based on Opcode.**

Table 14-15. Functional Characteristics of DCALDATA for Calibration Algorithms (Sheet 1 of 2)

Byte	Receive Enable	DQS Cal
71	Preamble status DQS17 rank1	Max delay DQS17 rank1
70	First edge position DQS17 rank1	Min delay DQS17 rank1
69	Preamble status DQS8 rank1	Max delay DQS8 rank1
68	First edge position DQS8 rank1	Min delay DQS8 rank1
67	Preamble status DQS16 rank1	Max delay DQS16 rank1
66	First edge position DQS16 rank1	Min delay DQS16 rank1
65	Preamble status DQS7 rank1	Max delay DQS7 rank1
64	First edge position DQS7 rank1	Min delay DQS7 rank1
63	Preamble status DQS15 rank1	Max delay DQS15 rank1
62	First edge position DQS15 rank1	Min delay DQS15 rank1
61	Preamble status DQS6 rank1	Max delay DQS6 rank1
60	First edge position DQS6 rank1	Min delay DQS6 rank1
59	Preamble status DQS14 rank1	Max delay DQS14 rank1
58	First edge position DQS14 rank1	Min delay DQS14 rank1
57	Preamble status DQS5 rank1	Max delay DQS5 rank1
56	First edge position DQS5 rank1	Min delay DQS5 rank1
55	Preamble status DQS13 rank1	Max delay DQS13 rank1
54	First edge position DQS13 rank1	Min delay DQS13 rank1
53	Preamble status DQS4 rank1	Max delay DQS4 rank1
52	First edge position DQS4 rank1	Min delay DQS4 rank1
51	Preamble status DQS12 rank1	Max delay DQS12 rank1
50	First edge position DQS12 rank1	Min delay DQS12 rank1
49	Preamble status DQS3 rank1	Max delay DQS3 rank1
48	First edge position DQS3 rank1	Min delay DQS3 rank1
47	Preamble status DQS11 rank1	Max delay DQS11 rank1
46	First edge position DQS11 rank1	Min delay DQS11 rank1
45	Preamble status DQS2 rank1	Max delay DQS2 rank1
44	First edge position DQS2 rank1	Min delay DQS2 rank1
43	Preamble status DQS10 rank1	Max delay DQS10 rank1
42	First edge position DQS10 rank1	Min delay DQS10 rank1
41	Preamble status DQS1 rank1	Max delay DQS1 rank1
40	First edge position DQS1 rank1	Min delay DQS1 rank1
39	Preamble status DQS9 rank1	Max delay DQS9 rank1

Table 14-15. Functional Characteristics of DCALDATA for Calibration Algorithms (Sheet 2 of 2)

Byte	Receive Enable	DQS Cal
38	First edge position DQS9 rank1	Min delay DQS9 rank1
37	Preamble status DQS0 rank1	Max delay DQS0 rank1
36	First edge position DQS0 rank1	Min delay DQS0 rank1
35	Preamble status DQS17 rank0	Max delay DQS17 rank0
34	First edge position DQS17 rank0	Min delay DQS17 rank0
33	Preamble status DQS8 rank0	Max delay DQS8 rank0
32	First edge position DQS8 rank0	Min delay DQS8 rank0
31	Preamble status DQS16 rank0	Max delay DQS16 rank0
30	First edge position DQS16 rank0	Min delay DQS16 rank0
29	Preamble status DQS7 rank0	Max delay DQS7 rank0
28	First edge position DQS7 rank0	Min delay DQS7 rank0
27	Preamble status DQS15 rank0	Max delay DQS15 rank0
26	First edge position DQS15 rank0	Min delay DQS15 rank0
25	Preamble status DQS6 rank0	Max delay DQS6 rank0
24	First edge position DQS6 rank0	Min delay DQS6 rank0
23	Preamble status DQS14 rank0	Max delay DQS14 rank0
22	First edge position DQS14 rank0	Min delay DQS14 rank0
21	Preamble status DQS5 rank0	Max delay DQS5 rank0
20	First edge position DQS5 rank0	Min delay DQS5 rank0
19	Preamble status DQS13 rank0	Max delay DQS13 rank0
18	First edge position DQS13 rank0	Min delay DQS13 rank0
17	Preamble status DQS4 rank0	Max delay DQS4 rank0
16	First edge position DQS4 rank0	Min delay DQS4 rank0
15	Preamble status DQS12 rank0	Max delay DQS12 rank0
14	First edge position DQS12 rank0	Min delay DQS12 rank0
13	Preamble status DQS3 rank0	Max delay DQS3 rank0
12	First edge position DQS3 rank0	Min delay DQS3 rank0
11	Preamble status DQS11 rank0	Max delay DQS11 rank0
10	First edge position DQS11 rank0	Min delay DQS11 rank0
9	Preamble status DQS2 rank0	Max delay DQS2 rank0
8	First edge position DQS2 rank0	Min delay DQS2 rank0
7	Preamble status DQS10 rank0	Max delay DQS10 rank0
6	First edge position DQS10 rank0	Min delay DQS10 rank0
5	Preamble status DQS1 rank0	Max delay DQS1 rank0
4	First edge position DQS1 rank0	Min delay DQS1 rank0
3	Preamble status DQS9 rank0	Max delay DQS9 rank0
2	First edge position DQS9 rank0	Min delay DQS9 rank0
1	Preamble status DQS0 rank0	Max delay DQS0 rank0
0	First edge position DQS0 rank0	Min delay DQS0 rank0

DCALDATA Receiver Enable "First edge position" byte detail	
Bit	**Description**
7:0	At the end of a successful calibration, this register holds the DRRTC setting that enables the DQS receiver as close as possible to but no earlier than the first rising DQS transition after the preamble. At the start of the calibration, this register is loaded with a value of 0xFF. During the calibration, while the "strobe toggle status" bit is low, this register will be updated with the DRRTC value for the current calibration step if the DQS is found to have a value of zero. After "strobe toggle status" goes high, this register will be updated with the DRRTC value when the DQS is found to have a value of one at a calibration step. This register will no longer be updated after the "preamble found status" bit goes high, so that it will retain the position of the rising DQS edge following immediately after the preamble.

DCALDATA Receiver Enable "Preamble status" byte detail	
Bit	**Description**
7	Strobe toggle status. Hardware sets this bit if a valid high pulse is found in the strobe waveform. The requirement is (DCALDATA.First_edge_position - last receiver enable delay value) > RCVENAC.HWIDTH
6	Preamble found status. Hardware sets this bit if the "preamble found" bit asserts at any time during the calibration.
5	Preamble found. Last receiver enable delay value meets or exceeds the preamble width requirement setting. Hardware sets this bit if: (DCALDATA.First_edge_position - last receiver enable delay value) > RCVENAC.PWIDTH
4:0	Count of "lows" minus count of "highs" found during one set of repeated tests at the last receiver enable delay setting. See DCALCSR opmods field for a description of the repeat test function.

DCALDATA DQS Cal Max Delay detail	
Bit	**Description**
7:6	reserved
5:0	At the end of a successful calibration, this field will hold the maximum DQS delay setting that results in correct data capture in the DDR I/O capture flop. This is the right edge of the DQ data eye. During the calibration, this field is updated with the DQS delay setting at each calibration step until the minimum delay setting is found and a subsequent failure to capture correct read data occurs.

Table 14-16. Functional Characteristics of DCALDATA for HVM Algorithms

Byte	I/O Loopback	DLL BIST
71:36	Not used	Not used
35	First "All Bits Failed" position and Status DQS17	Core Counter DQS17
34	First "Any Bit Failed" Position and Status DQS17	
33	First "All Bits Failed" position and Status DQS8	Core Counter DQS8
32	First "Any Bit Failed" Position and Status DQS8	
31	First "All Bits Failed" position and Status DQS16	Core Counter DQS16
30	First "Any Bit Failed" Position and Status DQS16	

Table 14-16. Functional Characteristics of DCALDATA for HVM Algorithms

Byte	I/O Loopback	DLL BIST
29	First "All Bits Failed" position and Status DQS7	Core Counter DQS7
28	First "Any Bit Failed" Position and Status DQS7	
27	First "All Bits Failed" position and Status DQS15	Core Counter DQS15
26	First "Any Bit Failed" Position and Status DQS15	
25	First "All Bits Failed" position and Status DQS6	Core Counter DQS6
24	First "Any Bit Failed" Position and Status DQS6	
23	First "All Bits Failed" position and Status DQS14	Core Counter DQS14
22	First "Any Bit Failed" Position and Status DQS14	
21	First "All Bits Failed" position and Status DQS5	Core Counter DQS5
20	First "Any Bit Failed" Position and Status DQS5	
19	First "All Bits Failed" position and Status DQS13	Core Counter DQS13
18	First "Any Bit Failed" Position and Status DQS13	
17	First "All Bits Failed" position and Status DQS4	Core Counter DQS4
16	First "Any Bit Failed" Position and Status DQS4	
15	First "All Bits Failed" position and Status DQS12	Core Counter DQS12
14	First "Any Bit Failed" Position and Status DQS12	
13	First "All Bits Failed" position and Status DQS3	Core Counter DQS3
12	First "Any Bit Failed" Position and Status DQS3	
11	First "All Bits Failed" position and Status DQS11	Core Counter DQS11
10	First "Any Bit Failed" Position and Status DQS11	
9	First "All Bits Failed" position and Status DQS2	Core Counter DQS2
8	First "Any Bit Failed" Position and Status DQS2	
7	First "All Bits Failed" position and Status DQS10	Core Counter DQS10
6	First "Any Bit Failed" Position and Status DQS10	
5	First "All Bits Failed" position and Status DQS1	Core Counter DQS1
4	First "Any Bit Failed" Position and Status DQS1	
3	First "All Bits Failed" position and Status DQS9	Core Counter DQ9
2	First "Any Bit Failed" Position and Status DQS9	
1	First "All Bits Failed" position and Status DQS0	Core Counter DQS0
0	First "Any Bit Failed" Position and Status DQS0	

DCALDATA I/O Loopback "Any Bit Failed" detail	
Bit	**Description**
7	First "First Any bit Failed" Nibble. This bit is set if the nibble associated with this register is one of the first to fail to capture data correctly during the test.
6	reserved
5:0	At the end of the test, this field will contain the minimum DQS delay setting that results in one or more bits of a burst to be captured incorrectly.

DCALDATA I/O Loopback "All Bits Failed" detail	
Bit	**Description**
7	Last "First All Bits Failed" Nibble. This bit is set if the nibble associated with this register is one of the last to capture an entire data burst incorrectly during the test.
6	reserved
5:0	At the end of the test, this field will contain the minimum DQS delay setting that results all bits of a burst to be captured incorrectly.

DCALDATA DLL BIST Core Counter detail	
Bit	**Description**
15:0	At the end of the test this two byte register will contain the number of core cycles counted from the time the associated DLL delay line outputs a "terminal count" number of self-oscillation cycles. The terminal count is defined by the DDBISTLM.TCOUNT register.

14.6.1.4 DDBISTLM: DDR DLL BIST Limits

This register contains test limits for DDR DLL BIST.

Device: NodeID
Function: 4
Offset: 90h

Bit	Attr	Default	Description
23:16	RWST	0Fh	**TCOUNT: DLL delay line output terminal count**
15:0	RWST	000Fh	**CVAR: core count variation limit**

14.6.1.5 RCVENAC: Receiver Enable Algorithm Control

This register contains controls for the preamble detection algorithm of the automatic receiver enable logic. RCVENAC.PWIDTH is used to determine if a "low" pulse in a DQS waveform is wide enough to be a preamble. RCVENAC.POFFSET is subtracted from the DCALDATA first edge position result and programmed into the DRRTC registers.

Device: NodeID
Function: 4
Offset: 94h

Bit	Attr	Default	Description
23:16	RWST	18h	**PWIDTH: Minimum preamble width limit, used to detect if a low pulse in a DQS waveform is wide enough to be a valid preamble. The default corresponds to 3/4 of a DRAM clock cycle**
15:14	RV	0h	**Reserved**
13:8	RWST	08h	**HWIDTH: Minimum high pulse width limit, used to detect if a high pulse in a DQS waveform is wide enough to indicate a strobe is toggling in a valid manner. The default corresponds to 1/4 of a DRAM clock cycle.**
7:6	RV	0h	**Reserved**
5:0	RWST	10h	**POFFSET: Preamble center offset from first rising edge, used to position the DQS receiver enable relative to the preamble edge location recorded in the DCALDATA registers. The default value corresponds to 1/2 of a DRAM clock cycle.**

14.6.1.6 DSRETC: DRAM Self-Refresh Extended Timing and Control

This register sets the timing of operations to different ranks while the auto-refresh FSM controls the DRAM command bus. This allows power intensive commands to be staggered. This register also contains the count for the auto-refresh FSM handshake time out. The FSM will wait a maximum of the time out count before taking control of the bus and issuing the command sequence to put the DRAMs into self-refresh. The RSTREQERR bit of the DSREFTC CSR will be set if the time out count is reached.

Device: NodeID
Function: 4
Offset: 98h

Bit	Attr	Default	Description
31:24	RV	0h	**Reserved**
23:16	RWST	14h	**DRSRENT: dual rank self-refresh entry timing - stagger of commands between ranks**
15:8	RWST	14h	**DRARTIM: dual rank auto-refresh timing- stagger of commands between ranks**
7:0	RWST	FFh	**TREQERR: reset handshake time out count** (counts in x16 of core clock) If times out - forces DRAMs into self-reset even if no handshake received from MemBIST or other logic after link goes into fast reset

14.6.1.7 DQSFAIL

There are two DQSFAIL registers that contain a total of 36 individual DQS failure status bits. There is one status bit for each DQS signal pair on each rank. These bits are set automatically by hardware during the receiver enable calibration if a valid DQS waveform is not detected. Hardware will not clear any bits that are set prior to the calibration even if a valid waveform is detected. Hardware uses the DQSFAIL information to exclude calibration data during the data gathering portion and/or the data analysis portion of the both the receiver enable and DQS delay calibrations. This prevents a failed DQS pin from corrupting the calibration of neighboring functional DQS pins that may share internal logic resources with a failing DQS pin.

14.6.1.8 DQSFAIL1: DQS Failure Configuration Register 1

Device: NodeID
Function: 4
Offset: 9Ch

Bit	Attr	Default	Description
7:4	RV	0h	**Reserved**
3	RWST	0	**r1dqs17: rank 1**
2	RWST	0	**r1dqs08: rank 1**
1	RWST	0	**r1dqs16: rank 1**
0	RWST	0	**r1dqs07: rank 1**

14.6.1.9 DQSFAIL0: DQS Failure Configuration Register 0

Device: NodeID
Function: 4
Offset: A0h

Bit	Attr	Default	Description
31	RWST	0	**r1dqs15: rank 1**
30	RWST	0	**r1dqs06: rank 1**
29	RWST	0	**r1dqs14: rank 1**
28	RWST	0	**r1dqs05: rank 1**
27	RWST	0	**r1dqs13: rank 1**
26	RWST	0	**r1dqs04: rank 1**
25	RWST	0	**r1dqs12: rank 1**
24	RWST	0	**r1dqs03: rank 1**
23	RWST	0	**r1dqs11: rank 1**
22	RWST	0	**r1dqs02: rank 1**
21	RWST	0	**r1dqs10: rank 1**
20	RWST	0	**r1dqs01: rank 1**
19	RWST	0	**r1dqs09: rank 1**
18	RWST	0	**r1dqs00: rank 1**
17	RWST	0	**r0dqs17: rank 0**
16	RWST	0	**r0dqs08: rank 0**
15	RWST	0	**r0dqs16: rank 0**
14	RWST	0	**r0dqs07: rank 0**
13	RWST	0	**r0dqs15: rank 0**
12	RWST	0	**r0dqs06: rank 0**
11	RWST	0	**r0dqs14: rank 0**
10	RWST	0	**r0dqs05: rank 0**
9	RWST	0	**r0dqs13: rank 0**
8	RWST	0	**r0dqs04: rank 0**
7	RWST	0	**r0dqs12: rank 0**
6	RWST	0	**r0dqs03: rank 0**
5	RWST	0	**r0dqs11: rank 0**
4	RWST	0	**r0dqs02: rank 0**
3	RWST	0	**r0dqs10: rank 0**
2	RWST	0	**r0dqs01: rank 0**
1	RWST	0	**r0dqs09: rank 0**
0	RWST	0	**r0dqs00: rank 0**

14.6.1.10 DRRTC: Receive Enable Reference Output Timing Control Registers

Note: These registers have to be saved and restored on S3.

The DRRTC is a set of three registers with DQS receiver enable window timing control for each byte on the DDR data bus. There is a single control for each byte for both ranks. A correct register setting will delay the start of the enable window so that it coincides with the middle of the DQS pre-amble. Enabling the window before or after the pre-amble would cause valid DQS edges to be missed or invalid edges or noise to be received.

The range of the enable delay, controlled by the DRRTC registers, is eight cycles, with a granularity defined by the SPDPAR06CUR.MASTCNTL register. The delay is measured from the AMC core clock edge that launches a "read" command on the DDR command bus. The minimum delay is equal to the DDR SDRAM read latency defined in the DRC.CL and DRC.AL register fields. The maximum delay is the read latency plus eight cycles. In addition to these major sources of delay, there is also a small "uncompensated delay" as shown in the formulas below.

The RCVEN fields of the DRRTC register control the delay as follows: bits [7:5] control whole clock increments, bits [4:3] control in quarter clock increments, and bits [2:0] control the sub-quarter cycle increments. Setting RCVEN to 0x0 produces the minimum delay, and 0xFF sets the maximum delay. The sub-quarter cycle delay is defined by the equations and "RCVEN_OUT" lookup table below:

Delay_Uncomp = 100ps; Note: estimate only

Delay Element = (quarter CMDCLK period - Delay_Uncomp) / (MASTCNTL + 0.5)

sub quarter cycle delay = Delay_Uncomp + (Delay Element * RCVEN_OUT[2:0])

RCVEN_OUT Lookup Table

SPDPAR06CUR MASTCNTL]	DRRTC RCVEN [2:0]							
	7	6	5	4	3	2	1	0
7	7	6	**5**	**4**	**3**	**2**	**1**	**0**
6	6	5	**5**	**4**	**3**	**2**	**1**	**0**
5	5	4	**4**	**3**	**2**	**1**	**1**	**0**
4	4	4	**3**	**3**	**2**	**1**	**1**	**0**
3	3	3	**2**	**2**	**1**	**1**	**0**	**0**
2	2	2	**2**	**1**	**1**	**0**	**0**	**0**
1	1	1	**1**	**1**	**0**	**0**	**0**	**0**
0	0	0	**0**	**0**	**0**	**0**	**0**	**0**

For example, if the SPDPAR06CUR.MASTCNTL is set to 0x7, the receiver enable delay can be varied over eight cycle in 256 steps, one step for each DRRTC RCVEN setting. If SPDPAR06CUR.MASTCNTL is set to 0x3, however, the number of steps is reduced to 128, such that half of the DRRTC RCVEN settings do not produce an increase in delay from the previous setting.

14.6.1.11 DRRTC00: Receive Enable Reference Output Timing Control Register

This register determines DQS12, 3, 11, 2, 10, 1, 9, & 0 input buffer enable timing delay

Device: NodeID
Function: 4
Offset: A4h

Bit	Attr	Default	Description
31:24	RWST	20h	**RCVEN1203: receiver enable delay for DQS12 and 3**
23:16	RWST	20h	**RCVEN1102: receiver enable delay for DQS11 and 2**
15:8	RWST	20h	**RCVEN1001: receiver enable delay for DQS10 and 1**
7:0	RWST	20h	**RCVEN0900: receiver enable delay for DQS9 and 0**

14.6.1.12 DRRTC01: Receive Enable Reference Output Timing Control Register

This register determines DQS16, 7, 15, 6, 14, 5, 13, & 4 input buffer enable timing delay.

Device: NodeID
Function: 4
Offset: A8h

Bit	Attr	Default	Description
31:24	RWST	20h	**RCVEN1607: receiver enable delay for DQS16 and 7**
23:16	RWST	20h	**RCVEN1506: receiver enable delay for DQS15 and 6**
15:8	RWST	20h	**RCVEN1405: receiver enable delay for DQS14 and 5**
7:0	RWST	20h	**RCVEN1304: receiver enable delay for DQS13 and 4**

14.6.1.13 DRRTC02: Receive Enable Reference Output Timing Control Register

This register determines DQS17 & 8 input buffer enable timing delay.

Device: NodeID
Function: 4
Offset: C4h

Bit	Attr	Default	Description
7:0	RWST	20h	**RCVEN1708: receiver enable delay for DQS17 and 8**

14.6.1.14 DQS Calibration Registers

The DQSOFCS is a group of six registers that control the fine delay used to center DQS edges to the DQ data eye during read operations. There is a delay entry for each nibble of the DDR data bus for each rank. The coarse delay is controlled by the DRAMDLLC register. The equations for the fine and coarse delays are shown below. Note that "Delay Element" and "Delay_Uncomp" are defined in the DRRTC register section. Also note that there is a separate coarse delay control for each "chunk" of the DDR I/O cluster as defined in the DRAMDLLC register section.

slvlen_not_at_max[m] = DRAMDLLC.SLVLENm[2:0] < 7; where m is the DDR I/O cluster chunk number

increment_slvlen[i,n] = slvlen_not_at_max AND (DQSOFCSi.DQSn[3:0] > 7); where i and n are the DQSOFCS register and DQS field numbers respectively

slvlen[i,m,n] = DRAMDLLC.SLVLENm[2:0] + increment_slvlen[i,n]

Programable_Delay[i,m,n] = (Delay Element) * (slvlen[i,m,n] + 0.5 + DQSOFCSi.DQSn[2:0]/8)

DQS_Delay[i,m,n] = Delay_Uncomp + Programmable_Delay[i,m,n]

Note: these registers may have to be saved and restored on S3

14.6.1.15 DQSOFCS00: DQS Calibration Register

This register determines DQS12, 3, 11, 2, 10, 1, 9, & 0 fine DQS delay when reading from rank 0.

Device: NodeID
Function: 4
Offset: B4h

Bit	Attr	Default	Description
31:28	RWST	0h	**DQS12: Fine delay**
27:24	RWST	0h	**DQS03: Fine delay**
23:20	RWST	0h	**DQS11: Fine delay**
19:16	RWST	0h	**DQS02: Fine delay**
15:12	RWST	0h	**DQS10: Fine delay**
11:8	RWST	0h	**DQS01: Fine delay**
7:4	RWST	0h	**DQS09: Fine delay**
3:0	RWST	0h	**DQS00: Fine delay**

14.6.1.16 DQSOFCS01: DQS Calibration Register

This register determines DQS16, 7, 15, 6, 14, 5, 13, & 4 fine DQS delay when reading from rank 0.

Device: NodeID
Function: 4
Offset: B8h

Bit	Attr	Default	Description
31:28	RWST	0h	**DQS16: Fine delay**
27:24	RWST	0h	**DQS07: Fine delay**
23:20	RWST	0h	**DQS15: Fine delay**
19:16	RWST	0h	**DQS06: Fine delay**
15:12	RWST	0h	**DQS14: Fine delay**
11:8	RWST	0h	**DQS05: Fine delay**
7:4	RWST	0h	**DQS13: Fine delay**
3:0	RWST	0h	**DQS04: Fine delay**

14.6.1.17 DQSOFCS02: DQS Calibration Register

This register determines DQS17 & 8 fine DQS delay when reading from rank 0.

Device: NodeID
Function: 4
Offset: C6h

Bit	Attr	Default	Description
7:4	RWST	0h	**DQS17: Fine DLL delay**
3:0	RWST	0h	**DQS08: Fine DLL delay**

14.6.1.18 DQSOFCS10: DQS Calibration Register

This register determines DQS12, 3, 11, 2, 10, 1, 9, & 0 fine DQS delay when reading from rank 1.

Device: NodeID
Function: 4
Offset: BCh

Bit	Attr	Default	Description
31:28	RWST	0h	**DQS12: Fine delay**
27:24	RWST	0h	**DQS03: Fine delay**
23:20	RWST	0h	**DQS11: Fine delay**
19:16	RWST	0h	**DQS02: Fine delay**
15:12	RWST	0h	**DQS10: Fine delay**
11:8	RWST	0h	**DQS01: Fine delay**
7:4	RWST	0h	**DQS09: Fine delay**
3:0	RWST	0h	**DQS00: Fine delay**

14.6.1.19 DQSOFCS11: DQS Calibration Register

This register determines DQS16, 7, 15, 6, 14, 5, 13, & 4 fine DQS delay when reading from rank 1.

Device: NodeID
Function: 4
Offset: C0h

Bit	Attr	Default	Description
31:28	RWST	0h	**DQS16: Fine delay**
27:24	RWST	0h	**DQS07: Fine delay**
23:20	RWST	0h	**DQS15: Fine delay**
19:16	RWST	0h	**DQS06: Fine delay**
15:12	RWST	0h	**DQS14: Fine delay**
11:8	RWST	0h	**DQS05: Fine delay**
7:4	RWST	0h	**DQS13: Fine delay**
3:0	RWST	0h	**DQS04: Fine delay**

14.6.1.20 DQSOFCS12: DQS Calibration Register

This register determines DQS17 & 8 fine DQS delay when reading from rank 1.

Device: NodeID
Function: 4
Offset: C7h

Bit	Attr	Default	Description
7:4	RWST	0h	**DQS17: Fine DLL delay**
3:0	RWST	0h	**DQS08: Fine DLL delay**

14.6.1.21 WPTRTC DDR I/O Write Pointer Timing

The two WPTRTC registers control the fine delay of the DDR I/O FIFO write pointers. The formulas for delay shown in the DQSOFCS and DRRTC register sections are identical to the write pointer delay formulas. To find the WPTRTC portion of write pointer delay, use the DQSOFCS formulas, and substitute WPTRTC fields for all the DQSOFCS fields. The only difference in the application of these formulas is that there is only one WPTRTC field per byte of the DDR I/O, whereas the DQSOFCS has a field per nibble per rank. The total write pointer delay, measured from the same reference point as the DQS receiver enable timing, is equal to the DQS receiver enable timing, including the quarter clock and sub-quarter clock delays, plus one full clock cycle, plus the coarse and fine DRAMDLLC.SLVLEN/WPTRTC delays calculated with the formulas from the DQSOFCS register section.

14.6.1.22 WPTRTC0: Write Pointer Timing Control 0

This register determines the DDR I/O FIFO write pointer fine delay timing for all DQS signals except DQS17 and DQS8 when reading from rank 0 or rank 1.

Device: NodeID
Function: 4
Offset: CCh

Bit	Attr	Default	Description
31:28	RWST	0h	**DQS1607: DQS16 and DQS7 write pointer fine delay**
27:24	RWST	0h	**DQS1506: DQS15 and DQS6 write pointer fine delay**
23:20	RWST	0h	**DQS1405: DQS14 and DQS5 write pointer fine delay**
19:16	RWST	0h	**DQS1304: DQS13 and DQS4 write pointer fine delay**
15:12	RWST	0h	**DQS1203: DQS12 and DQS3 write pointer fine delay**
11:8	RWST	0h	**DQS1102: DQS11 and DQS2 write pointer fine delay**
7:4	RWST	0h	**DQS1001: DQS10 and DQS1 write pointer fine delay**
3:0	RWST	0h	**DQS0900: DQS9 and DQS0 write pointer fine delay**

14.6.1.23 WPTRTC1: Write Pointer Timing Control 1

This register determines the DDR I/O FIFO write pointer fine delay timing for DQS17 and DQS8 signals when reading from rank 0 or rank 1.

Device: NodeID
Function: 4
Offset: D0h

Bit	Attr	Default	Description
7:4	RV	0h	**Reserved**
3:0	RWST	0h	**DQS01708: Rank 0 DQS17 and DQS8 write pointer fine delay**

14.6.1.24 DDQSCVDP and DDQSCADP

This set of 4 registers defines two 64 bit long data patterns used in the DQS Delay Calibration. They are only used when DCALCSR.BASPAT is low. The 64 bit patterns cover a data burst that is 32 DRAM clock cycles long. The DDQSCVDP registers define the "victim" pattern, and the DDQSCADP defines the "aggressor" pattern. The victim pattern is applied to one bit of each byte of the DDR data bus for 32 clock cycles, and the aggressor pattern is applied to all other bits. The victim pattern is applied in turn to each bit of each byte, creating a complete data pattern that is 8*32 data cycles long.

14.6.1.25 DDQSCVDP0: DQS DELAY CAL PATTERN 0

This register defines the last 32 bits of the 64 bit long "victim" data pattern.

Device: NodeID
Function: 4
Offset: D4h

Bit	Attr	Default	Description
31:0	RW	aaaa0a05h	**ENABLE:**

14.6.1.26 DDQSCVDP1: DQS DELAY CAL PATTERN 1

This register defines the first 32 bits of the 64 bit long "victim" data pattern.

Device: NodeID
Function: 4
Offset: D8h

Bit	Attr	Default	Description
31:0	RW	5b339c5dh	**ENABLE:**

14.6.1.27 DDQSCADP0: DQS DELAY CAL PATTERN 0

This register defines the last 32 bits of the 64 bit long "aggressor" data pattern.

Device: NodeID
Function: 4
Offset: DCh

Bit	Attr	Default	Description
31:0	RW	aaabffffh	**ENABLE:**

14.6.1.28 DDQSCADP1: DQS DELAY CAL PATTERN 1

This register defines the first 32 bits of the 64 bit long "aggressor" data pattern.

Device: NodeID Function: 4 Offset: E0h			
Bit	**Attr**	**Default**	**Description**
31:0	RW	db339ce1h	**ENABLE:**

14.6.2 Memory Interface Control

14.6.2.1 DIOMON: DDR I/O Monitor

This register monitors the legsel output of the DDR I/O topcdat chunk and controls the A/D converter in the DDR I/O used to monitor analog voltage levels.

Device: NodeID Function: 4 Offset: F0h			
Bit	**Attr**	**Default**	**Description**
15	RW	0	**ENABLE: Enable A/D converter and update vresult**
14:12	RWST	0h	**BIASSEL: A/D converter input selection**
11:8	RWST	0h	**LEGSELOUT: Legsel output of topcdat chunk**
7	RWST	0h	**DIOPWR:**
6	RV	0h	**Reserved**
5:0	RWST	00h	**VRESULT: A/D converter output**

14.6.2.2 ODTZTC: On-Die Termination Timing Control

This register controls the enable and disable timing of the on-die termination on DQ and DQS pins. Timing can be adjusted in whole clock increments, or enabled statically. The ETIMR and DTIMR fields are added in hardware to the SPDPAR13CUR.ODTZ_ETIMR and SPDPAR13CUR.ODTZ_DTIMR register fields to control termination timing during reads. The DRRTC register is also used to align the enable/disable time to when read DQ/DQS signals are expected to arrive at the input pins. The combined default values of the ODTZTC and SPDPAR13CUR fields enable termination 1/2 DRAM clock cycle before the leading edge of the read DQS preamble arrives at the DQS pin, and keeps termination enabled for 5 clock cycles in BL4 mode, and 7 cycles in BL8 mode. The DDR I/O circuits automatically disable on-die termination when the DQ/DQS pins are driving. The default register values enable termination during writes at the same time that the pins are driving, so termination is effectively off during writes.

Device: NodeID Function: 4 Offset: F4h			
Bit	**Attr**	**Default**	**Description**
15	RWST	0	**TIMORIDE: timing override. On-Die termination always on during read operations and when the bus is idle**
14:12	RWST	0h	**DTIMW: disable time after write data**
11	RV	0	**Reserved**

Device: NodeID Function: 4 Offset: F4h			
Bit	**Attr**	**Default**	**Description**
10:8	RWST	0h	**DTIMR: disable time after read data**
7:4	RWST	0h	**ETIMW: enable time before write data**
3:0	RWST	0h	**ETIMR: enable time before read data**

14.6.2.3 DRAMISCTL: Miscellaneous DRAM DDR Cluster Control

Device: NodeID Function: 4 Offset: F8h			
Bit	**Attr**	**Default**	**Description**
31:13	RV	0000h	**Reserved**
12	RWST	1	**VOXSTART: Enable the voltage output crossing control loop in the DDR I/O. This bit is AND'ed with the SPDPAR1011CUR.vox_start bit.**
11	RW	0	**AVMODE: analog validation mode**
10	RW	0	**OCDLOADENABLE: calibration load placed on incoming signals for DDR2 DRAM OCD calibration**
9	RW	0	**OCDPOLSEL: set for pull up calibration, clear for pull down**
8	RW	0	**OCDRCOMPEN:**
7:0	RWST	11h	**VREFSEL: vref selection**

Details of DRAMISCTL VREF Field	
Setting	**Description**
	TBD

14.6.2.4 DDR2ODTC: DDR2 DRAM On-Die Termination Control

The DDR2ODTC controls the behavior of the ODT output (one ODT output per command bus copy). There is a separate field to control the behavior for reads to rank0, reads to rank1, writes to rank0, and writes to rank1. Only the lsb of each field is used, and the msb has no effect. When an lsb of a field is set to one, the ODT pins will drive high, at the appropriate time relative to data on the DDR bus, when the selected transaction (read or write) is issued to the selected rank (0 or 1). If a field is set to 0, the ODT output continues to drive low during the transaction, as it does during idle cycles.

Device: NodeID Function: 4 Offset: FCh			
Bit	**Attr**	**Default**	**Description**
7:6	RWST	0h	**R1ODTWR: ODT control during writes to CS1** **x1: ODT pins will drive high** **x0: ODT pins remain driving low**
5:4	RWST	0h	**R1ODTRD: ODT control during reads to CS1**
3:2	RWST	0h	**R0ODTWR: ODT control during writes to CS0**

Device: NodeID Function: 4 Offset: FCh			
Bit	**Attr**	**Default**	**Description**
1:0	RWST	0h	**R0ODTRD: ODT control during reads to CS0**

14.6.2.5 DRAMDLLC: DDR I/O DLL Control

The formulas that show how the SLVLEN fields affect DQS delay timing are shown in the DQSOFCS register definition section. The SLVLEN fields are set by hardware during the DQS delay calibration. There are five SLVLEN fields, one for each "chunk", or two bytes, of the DDR I/O DQ pins. The SLVBYP bit can be toggled to reset the master DLL's in the DDR I/O.

Device: NodeID Function: 4 Offset: C8h			
Bit	**Attr**	**Default**	**Description**
23:22	RV	00h	*Reserved*
21	RW	0h	**SLVBYP: DQS delay bypass**
20:18	RWST	3h	**SLVLEN4: dqs17 & 8 coarse DQS delay**
17:15	RWST	3h	**SLVLEN3: dqs16, 7, 15, & 6 coarse DQS delay**
14:12	RWST	3h	**SLVLEN2: dqs14, 5, 13, & 4 coarse DQS delay**
11:9	RWST	3h	**SLVLEN1: dqs12, 3, 11, & 2 coarse DQS delay**
8:6	RWST	3h	**SLVLEN0: dqs10, 1, 9, & 0 coarse DQS delay**
5:0	RV	00h	*Reserved*

14.6.3 Firmware Support Registers

14.6.3.1 FIVESREG: Fixed 5's Pattern

Constant value used for debug.

Device: NodeID Function: 4 Offset: E8h			
Bit	**Attr**	**Default**	**Description**
31:0	RO	55555555h	**FIVES: Hardwired to 5's for read-return**

14.6.3.2 AAAAREG: Fixed A's Pattern

Constant value used for debug.

Device: NodeID Function: 4 Offset: ECh			
Bit	**Attr**	**Default**	**Description**
31:0	RO	AAAAAAAAh	**AAAAS: Hardwired to A's for read-return**

14.7 DFX Registers (Function 5)

14.7.1 Transparent Mode Registers

14.7.1.1 TRANSCFG: Transparent Mode Configuration

This register enables and controls FBD DFX transparent mode features.

Device: NodeID Function: 5 Offset: 3Ch			
Bit	**Attr**	**Default**	**Description**
31:29	RV	0h	Reserved
28	RWST	0	**ENDOUT: enable data output on transparent data/status pins when set, output status when clear**
27	RWST	0	**LGFBITS: log bits that fail the compare when set, log raw read data when clear.**
26	RWST	0	**LGFFAIL: log first failure in any burst position.**
25:24	RWST	00	**BSTPOS: burst position to log data/failed bits from when LGFFAIL bit is not set.** 0 = first burst in bl4 or bl8 mode 1 = last burst of bl4, second burst of bl8 2 = third burst of bl8 3 = last burst of bl8
23:20	RWST	0h	**DRAMRD: byte of data bus selected to be output on transparent data/status pins when ENDOUT bit is set.** **8= DQS 17 and DQS 8** **7= DQS 16 and DQS 7** **…** **0 = DQS 9 and DQS 0**
19:16	RWST	0h	**DRAMWR: byte of data bus selected to receive transparent write data, and byte of data bus to be compared against transparent read data.** **Fh = all bytes** **8= DQS 17 and DQS 8** **7= DQS 16 and DQS 7** **…** **0 = DQS 9 and DQS 0**
15:0	RWST	0000h	**DFTDATA: default data for bytes not selected by the DRAMWR field. This field has early/even data in the upper 8 bits, and late/odd data in the lower 8 bits.**

14.7.1.2 TRANDERR[8:0]: Transparent Mode Data Error Logs

This register stores data returned from DRAM byte groups on failing transparent mode tests.

Device: NodeID Function: 5 Offset: 50h, 4Eh, 4Ch, 4Ah, 48h,46h, 44h, 42h, 40h			
Bit	**Attr**	**Default**	**Description**
15:8	RWST	00h	**LATE_DATA:**
7:0	RWST	00h	**EARLY_DATA:**

14.7.1.3 TRANSCTRL: Transparent Mode Control

This register enables and controls FBD DFX transparent mode features.

Device: NodeID Function: 5 Offset: 80h			
Bit	**Attr**	**Default**	**Description**
7:1	RV	00h	Reserved
0	RWST	0	**ENTRNSPMODE: Transparency Mode Enable** 1 - Enables Transparency Mod

14.7.2 Logic Analyzer Interface (LAI) Registers

14.7.2.1 LAI: LAI Operation Modes

This register controls and reports the AMB's LAI mode and Qual controls.

Device: NodeID Function: 5 Offset: B8h			
Bit	**Attr**	**Default**	**Description**
31:16	RV	0000h	**Reserved**
15	RWST	0	**RAWMODE: data connected to LAI** 0: LAI outputs contain initialization state information **prior to L0S then lane data after L0S** 1: LAI outputs connected to FBD data inputs even though valid timing is not present
14	RV	0	**Reserved**
13	RWST	0	**QUALMODE:** Assert Qual for all non-filtered frames, or only assert Qual for all non-filtered frames between start and stop events. 0: Ignore Qual start/stop events 1: Assert Qual after a start event, and deassert Qual after a stop events
12	RWST	0	**FILTERSYNC:** Filter the frame (do not assert Qual) if the frame is a sync. 0: Disable sync filtering 1: Enable sync filtering
11:6	RV	00h	**Reserved**
5:0	RWST	3Fh	**QUALPERIOD:** Additional number of frames Qual remains asserted Power-on default to 63

14.7.2.2 SBMATCHU: Upper Southbound Match Register

This register sets the upper 8 bits of data match for three southbound commands.

Device: NodeID Function: 5 Offset: BCh			
Bit	**Attr**	**Default**	**Description**
31:24	RV	00h	**Reserved**
23:16	RWST	00h	**CMD2:** Upper 8 bits [39:32] of southbound command 2

Device: NodeID Function: 5 Offset: BCh			
Bit	**Attr**	**Default**	**Description**
15:8	RWST	00h	**CMD1:** Upper 8 bits [39:32] of southbound command 1
7:0	RWST	00h	**CMD0:** Upper 8 bits [39:32] of southbound command 0

Match and Mask bit numbering of SB frames is as follows:

- where N= 0 for slot A, 4 for slot B and 8 for slot C

Table 14-17. Bit Locations for SB Match and Mask

Xfr\lane	9	8	7	6	5	4	3	2	1	0
0 + N	B36	B32	B28	B24	B20	B16	B12	B8	B4	B0
1 + N	B37	B33	B29	B25	B21	B17	B13	B9	B5	B1
2 + N	B38	B34	B30	B26	B22	B18	B14	B10	B6	B2
3 + N	B39	B35	B31	B27	B23	B19	B15	B11	B7	B3

14.7.2.3 SBMATCHL0: Lower Southbound Match Register 0

This register sets the lower 32 bits of data match for match southbound command 0.

Device: NodeID Function: 5 Offset: C0h			
Bit	**Attr**	**Default**	**Description**
31:0	RWST	00004000h	**CMD:** Lower 32bits [31:0] of southbound command power on default = match Synch

14.7.2.4 SBMATCHL1: Lower Southbound Match Register 1

This register sets the lower 32 bits of data match for match southbound command 1.

Device: NodeID Function: 5 Offset: C4h			
Bit	**Attr**	**Default**	**Description**
31:0	RWST	00100000h	**CMD:** Lower 32bits [31:0] of southbound command power on default = match Activate

14.7.2.5 SBMATCHL2: Lower Southbound Match Register 2

This register sets the lower 32 bits of data match for match southbound command 2.

Device: NodeID
Function: 5
Offset: C8h

Bit	Attr	Default	Description
31:0	RWST	00014000h	**CMD:** Lower 32bits [31:0] of southbound command power on default = match Write Config Reg

14.7.2.6 SBMASKU: Upper Southbound Mask Register

This register sets the upper 8 bits of data mask for three southbound commands.

Device: NodeID
Function: 5
Offset: CCh

Bit	Attr	Default	Description
31:24	RV	00h	**Reserved**
23:16	RWST	00h	**CMDMASK2:** Upper 8 bits [39:32] of southbound command 2 mask 0: Do not include this bit in match comparison 1: Include this bit in match comparison
15:8	RWST	00h	**CMDMASK1:** Upper 8 bits [39:32] of southbound command 1 mask 0: Do not include this bit in match comparison 1: Include this bit in match comparison
7:0	RWST	00h	**CMDMASK0:** Upper 8 bits [39:32] of southbound command 0 mask 0: Do not include this bit in match comparison 1: Include this bit in match comparison

14.7.2.7 SBMASKL0: Lower Southbound Mask Register 0

This register sets the lower 32 bits of data mask for match southbound command 0.

Device: NodeID
Function: 5
Offset: D0h

Bit	Attr	Default	Description
31:0	RWST	031FC000h	**CMDMASK:** Lower 32bits [31:0] of southbound command mask. 0: Do not include this bit in match comparison 1: Include this bit in match comparison power on default = match Sync

14.7.2.8 SBMASKL1: Lower Southbound Mask Register 1

This register sets the lower 32 bits of data mask for match southbound command 1.

Device: NodeID
Function: 5
Offset: D4h

Bit	Attr	Default	Description
31:0	RWST	00100000h	**CMDMASK:** Lower 32bits [31:0] of southbound command mask. 0: Do not include this bit in match comparison 1: Include this bit in match comparison power on default = match Activate

14.7.2.9 SBMASKL2: Lower Southbound Mask Register 2

This register sets the lower 32 bits of data mask for match southbound command 2.

Device: NodeID
Function: 5
Offset: D8h

Bit	Attr	Default	Description
31:0	RWST	001FC000h	**CMDMASK:** Lower 32bits [31:0] of southbound command mask. 0: Do not include this bit in match comparison 1: Include this bit in match comparison power on default = match Write Config Reg

14.7.2.10 MMEVENTSEL: Match/Mask Event Selection Register

Selects 1 of 13 local match events described below for promotion to local event select. Three local events MMEVENT[2:0] can be associated with one of these local match events.

Device: NodeID
Function: 5
Offset: DCh

Bit	Attr	Default	Description
15:12	RV	0h	**Reserved**
11:8	RWST	2h	**MMEVENT2SEL:** default: match pattern 0 to slot A only for sync
7:4	RWST	Ah	**MMEVENT1SEL:** default: match pattern 1 to slot A, B, or C for activate
3:0	RWST	3h	**MMEVENT0SEL:** default: match pattern 2 to slot B only for write config reg

MM Event	Description
15:13	**Reserved**
12	**FRAMEMATCH** Bit pattern in slot A matches SBMATCH/SBMASK 0 **AND** Bit pattern in slot B matches SBMATCH/SBMASK 1 **AND** Bit pattern in slot C matches SBMATCH/SBMASK 2

MM Event	Description
11	**SLOTMATCH0** Command in slot A, B, or C matches SBMATCH/SBMASK 0
10	**SLOTMATCH1** Command in slot A, B, or C matches SBMATCH/SBMASK 1
9	**SLOTMATCH2** Command in slot A, B, or C matches SBMATCH/SBMASK 2
8	**SLOTCMATCH0** Command in slot C matches SBMATCH/SBMASK 0
7	**SLOTCMATCH1** Command in slot C matches SBMATCH/SBMASK 1
6	**SLOTCMATCH2** Command in slot C matches SBMATCH/SBMASK 2
5	**SLOTBMATCH0** Command in slot B matches SBMATCH/SBMASK 0
4	**SLOTBMATCH1** Command in slot B matches SBMATCH/SBMASK 1
3	**SLOTBMATCH2** Command in slot B matches SBMATCH/SBMASK 2
2	**SLOTAMATCH0** Command in slot A matches SBMATCH/SBMASK 0
1	**SLOTAMATCH1** Command in slot A matches SBMATCH/SBMASK 1
0	**SLOTAMATCH2** Command in slot A matches SBMATCH/SBMASK 2

14.7.2.11 EVENTSEL0: Event Selection Register

In LAI mode, selects 1 of 32 local events (see EVENT register) for 2 uses (Qual Start and Qual Stop) and sets programmable delay for QualStop. In Normal mode, selects local event for 2 uses (error injection and NB in-band event signaling) and sets programmable delay for error injection.

Device: NodeID
Function: 5
Offset: E0h

Bit	Attr	Default	Description
31:21	RV	00h	*Reserved*
20:15	RWST	3Fh	**QUALSTOPDELAY: in LAI Mode** Additional number of clocks QualStop is delayed before use (0 to 63) Power-on default to 63 **INJERRORDELAY: in Normal Mode** Additional number of clocks INJERROR is delayed before use (0 to 63) Power-on default to 63
14:10	RW	00h	**INBAND: in LAI Mode** No effect. Northbound in-band debug events can not be asserted in LAI mode **INBAND: in Normal Mode** If selected event occurs, assert the northbound in-band event bit in next sync status 0 return and in FBDS0.S3

Device: NodeID Function: 5 Offset: E0h			
Bit	**Attr**	**Default**	**Description**
9:5	RWST	00h	**QUALSTART: in LAI Mode** If selected event occurs, enable assertion of Qual until next QUALSTOP
4:0	RWST	00h	**QUALSTOP: in LAI Mode** • If selected event occurs, disable assertion of Qual until next QUALSTART **INJERROR: in Normal Mode** • If selected event occurs, inject error selected by EICNTL

14.7.2.12 EVENTSEL1: Event Selection Register

Selects 1 of 32 local events (see EVENT register) for each of six events transmitted to the LAI interface (events[5:0]).

Device: NodeID Function: 5 Offset: E4h			
Bit	**Attr**	**Default**	**Description**
31:30	RV	00	**Reserved**
29:25	RWST	0Ah	**LAITRIGGER5:** Local event selected and transmitted to LAI as TRIGGER5 **Default on power on to MM[1] event= Activate on any command**
24:20	RWST	0Bh	**LAITRIGGER4:** Local event selected and transmitted to LAI as TRIGGER4 **Default on power on to MM[2] event= Sync command in slot A**
19:15	RWST	13h	**LAITRIGGER3:** Local event selected and transmitted to LAI as TRIGGER3 **Default on power on to select in-band debug event 3**
14:10	RWST	12h	**LAITRIGGER2:** Local event selected and transmitted to LAI as TRIGGER2 **Default on power on to select in-band debug event 2**
9:5	RWST	11h	**LAITRIGGER1:** Local event selected and transmitted to LAI as TRIGGER1 **Default on power on to select in-band debug event 1**
4:0	RWST	10h	**LAITRIGGER0:** Local event selected and transmitted to LAI as TRIGGER0 **Default on power on to select in-band debug event 0**

14.7.2.13 EVENTSEL2: Event Selection Register

Selects 1 of 32 local events (see EVENT register) for each of five events transmitted to the LAI interface (events[10:6]).

Device: NodeID Function: 5 Offset: E8h			
Bit	**Attr**	**Default**	**Description**
31:30	RV	00	**Reserved**

Device: NodeID Function: 5 Offset: E8h			
Bit	**Attr**	**Default**	**Description**
29:25	RV	00h	**Reserved**
24:20	RWST	04h	**LAITRIGGER10:** Local event selected and transmitted to LAI as TRIGGER10 **Default on power on to SB Unit Testing State detect**
19:15	RWST	1Ah	**LAITRIGGER9:** Local event selected and transmitted to LAI as TRIGGER9 **Default on power on to SB CRC error detect**
14:10	RWST	0Dh	**LAITRIGGER8:** Local event selected and transmitted to LAI as TRIGGER8 **Default on power on to Event Bus[1] event**
9:5	RWST	0Ch	**LAITRIGGER7:** Local event selected and transmitted to LAI as TRIGGER7 **Default on power on to Event Bus[0] event**
4:0	RWST	09h	**LAITRIGGER6:** Local event selected and transmitted to LAI as TRIGGER6 **Default on power on to MM[0] event= Write Register in command slot B**

14.7.2.14 EVENT: Local LAI Event Register

This register sets a bit if a local event is hit. Except for QUALFLAG, most local event signal internal to the AMB may assert for only one cycle. The event bits in this register remain set once asserted so that the history of the event being set is not lost.

Note: The 5-bit select fields in the EVENTSEL0, EVENTSEL1, EVENTSEL2 and EVBUS registers use the bit mapping of this register to identify which local event to select.

- For example, if EVENTSEL1.laitrigger0[4:0] = 26 decimal then trigger0 is connected to the SB CRC error event.
- For example, if EVENTSEL1.laitrigger0[4:0] = 12 decimal then trigger0 is connected to the inbound EVBus[0] event, and so forth.

Device: NodeID Function: 5 Offset: F0h			
Bit	**Attr**	**Default**	**Description**
31	RWCST	0	**Spare*:***
30:25	RWCST	00h	ERROR EVENTS: • SB/NB failover[25], when unmasked: • SB CRC error[26], • thermal overload[27], • clock training violation (< 6 transitions in 512 UI) [28], • unimplemented register access[29], • other implementation specific errors[30] set on event and cleared by writing
24	RWCST	0	QUALFLAG:
23:16	RWCST	00h	INBANDEV: SB link in-band EV[7:0] set on event and cleared by writing

Device: NodeID Function: 5 Offset: F0h			
Bit	**Attr**	**Default**	**Description**
15:12	RWCST	0h	EV: Event Bus EV[3:0] set on event and cleared by writing
11:9	RWCST	0h	MMEVENT: MMEVENT[2:0] selected by MMEVENTSEL set on event and cleared by writing
8:1	RWST	0h	LINKST: FBD Link State: Disable[1], calibrate[2], training[3], testing[4], pollling[5], config[6], l0[7], L0s or recalibrate[8] One hot encoding of FBD link state
0	RO	0	NULL: Null Event: Bit never set

14.7.3 Error Injection Registers

14.7.3.1 EICNTL: Error Injection Control

This register controls the AMB error injection logic.

Device: NodeID Function: 5 Offset: FCh			
Bit	**Attr**	**Default**	**Description**
7	RW	0	**EIEN:Error Injection Enable** 1= Error Injection enabled 0= Error Injection disabled
6:4	RW	000	**EITYPE: Type of error injection** 111 = Reserved; 110 = Reserved 101 = Force NB Error bit on next Status return 100 = Force Alert on event 011 = Reserved 010 =Reserved 001= Corrupt NB CRC on event 000= No error injection
3:0	RV	0h	**Reserved**

14.7.3.2 STUCKL: Stuck "ON" FBD Lanes

This register selects FBD Lanes to be stuck at "Electrical Idle" following a write to this register.

Device: NodeID Function: 5 Offset: FEh			
Bit	**Attr**	**Default**	**Description**
7:4	RV	0h	*Reserved*
3:0	RWST	Fh	NBSTUCK: NB Lane to be driven to EI to simulate a failed lane • 0h = lane 0: Dh = lane 13 and > 13 = NOP

14.8 Bring-up and Debug Registers (Function 6)

14.8.1 SPAD[1:0]: Scratch Pad

These bits have no effect upon the operation of the AMB. They are intended to be used by software for tracking changes in AMB state. These two registers are in different functions.

Device: NodeID
Function: 7, 6
Offset: 7Ch

Bit	Attr	Default	Description
31:0	RW	0000_0000h	**FREE:** These bits are available for software definition.

14.8.2 Southbound FBD Intel® Interconnect BIST Registers

14.8.2.1 SBFIBPORTCTL: Southbound FBD Intel® Interconnect BIST Port Control Register

This register controls the operation of the Intel® Interconnect BIST (Intel® IBIST) logic. Please refer to *Fully-Buffered DIMM DFx Specification* for detailed description of the operation of Intel IBIST.

Device: NodeID
Function: 6
Offset: 80h

Bit	Attr	Default	Description
31:24	RV	00h	**Reserved**
23	RWST	0	**SBNBMAP: Loopback mapping bit** This bit will be sent during TS1 to the slave to specify which lanes needs to be looped back. Actual lanes looped back is specified in the FBD Architecture Spec.
22	RWST	0	**CMMSTR: Compliance Measurement Mode start** This puts the Intel IBIST logic in CMM mode and Intel IBIST TX engine will start transmitting Intel IBIST patterns.
21:12	RWCST	000h	**ERRCNT: Error Counter** Total number of frames with errors that were encountered by this port.
11:8	ROST	0h	**ERRLNNUM: Error Lane Number** This points to the first lane that encountered an error. If more than one lane reports an error in a cycle, the most significant lane number that reported the error will be logged.
7:6	RWCST	00	**ERRSTAT: Port Error Status** When Intel IBIST is started, status goes to 01 until first start delimiter is received and then goes to 00 until the end or to10 if an error occurs. 00: No error. 01: Did not receive first start delimiter. 10: Transmission error (first error). 11: Reserved for future use
5	RWST	0	**AUTOINVSWPEN: Auto Inversion sweep enable** This bit enables the inversion shift register to continuously rotate the pattern in the **FIBTXSHFT** and **FIBRXSHFT** registers. 0: Disable Auto-inversion 1: Enable Auto-inversions

Device: NodeID Function: 6 Offset: 80h			
Bit	**Attr**	**Default**	**Description**
4	RWST	0	**STOPONERR: Stop IBIST on Error** 0: Do not stop on error, only update error counter 1: Stop on error
3	RWST	0	**LOOPCON: Loop forever** 0: No looping 1: Loop forever
2	RWCST	0	**COMPLETE: IBIST done flag** This bit is set when the receive engine is done checking. 0: Not done 1: Done
1	RWST	0	**MSTRMD: Master Mode Enable** When this bit is set along with **IBISTR**, the Intel IBIST transmit engine is enabled to transmit Intel IBIST patterns. 0: Disable Master mode. 1: Enable master mode.
0	RWST	0	**IBISTR: IBIST Start** When set, Intel IBIST starts transmitting after the TS1 header is recognized during the next link initialization sequence. and **MSTRMD** bit is set.This bit also enables the receive engine to start looking for the start delimiter during TS1 training set. This bit is reset when Intel IBIST receiver is done. The Intel IBIST transmit and receive engines can be stopped by setting this bit to 0. 0: Stop IBIST transmitter 1: Start IBIST transmitter

14.8.2.2 SBFIBPGCTL: SB Intel IBIST Pattern Generator Control Register

This register contains bits to control the operation of the Intel IBIST pattern generator. All counts are in 24 bit increments.

Device: NodeID Function: 6 Offset: 84h			
Bit	**Attr**	**Default**	**Description**
31:26	RWST	0Fh	**OVRLOOPCNT: Overall Loop Count** 0h: Reserved 1h:3Fh: Number of times to loop all the patterns.
25:21	RWST	00h	**CNSTGENCNT: Constant Generator Loop Counter** 00h: Disable constant generator output 01h: 1Fh The number of times the constant generator counter pattern should loop before going to the next component. Each count represents 24 bits of 1's or 0's.
20	RWST	0	**CNSTGENSET: Constant Generator Setting** 0: Generate 0 1: Generate 1
19:13	RWST	0Fh	**MODLOOPCNT: Modulo-N Loop Counter** Each count represents 24 bits of the pattern specified by **MODPERIOD**. 00h: Disable Pattern Output 01h: 7Fh The number of times the Pattern Buffer should loop before going to the next component.

Device: NodeID Function: 6 Offset: 84h			
Bit	**Attr**	**Default**	**Description**
12:10	RWST	001b	**MODPERIOD: Period of the Modulo-N counter** Each encoding transmits a 24-bit pattern as specified below. All other encodings are reserved. 001: L/2 - 0101_0101_0101_0101_0101_0101 010: L/4 - 0011_0011_0011_0011_0011_0011 011: L/6 - 0001_1100_0111_0001_1100_0111 100: L/8 - 0000_1111_0000_1111_0000_1111 110: L/24 - 0000_0000_0000_1111_1111_1111
9:3	RWST	0Fh	**PATTLOOPCNT: Pattern Buffer Loop Counter** 00h: Disable Pattern Output 01h: 7Fh The number of times the Pattern Buffer (**FIBPATTBUF1**) should be repeated before going to the next component.
2:0	RWST	000	**PTGENORD: Pattern Generation Order** 000: Pattern Store + Modulo N Cntr + Constant Generator 001: Pattern Store + Constant Generator + Modulo N Cntr 010: Modulo N Cntr + Pattern Store + Constant Generator 011: Modulo N Cntr + Constant Generator + Pattern Store 100: Constant Generator + Pattern Store + Modulo N Cntr 101: Constant Generator + Modulo N Cntr + Pattern Store 110: Reserved 111: Reserved

14.8.2.3 SBFIBPATTBUF1: SB Intel IBIST Pattern Buffer 1 Register

This register contains the pattern bits used in Intel IBIST operations. Only the least significant 24 bits are used. This user specified pattern goes out on to the link with the least-significant 12 bits as the first frame and the most significant 12 bits as the second frame.

Device: NodeID Function: 6 Offset: 88h			
Bit	**Attr**	**Default**	**Description**
31:24	RV	00h	**Reserved**
23:0	RWST	02ccfdh	**IBPATBUF: IBIST Pattern Buffer** Pattern buffer storing the default and the user programmable pattern. Default: 02ccfdh

14.8.2.4 SBFIBTXMSK: SB Intel IBIST Transmitter Mask Register

This register enables Intel IBIST operations for individual lanes. This mask only applies to transmitters and not receivers.

Device: NodeID Function: 6 Offset: 8Ch			
Bit	**Attr**	**Default**	**Description**
31:10	RV	000000h	**Reserved**
9:0	RWST	3FFh	**TXMASK: Selects which channels to enable for testing.**

14.8.2.5 SBFIBRXMSK: SB Intel IBIST Receiver Mask Register

This register enables Intel IBIST operations for individual lanes. This mask only applies to receivers and not transmitters.

Device: NodeID
Function: 6
Offset: 90h

Bit	Attr	Default	Description
31:10	RV	000000h	**Reserved**
9:0	RWST	3FFh	**RXMASK: Selects which channels to enable for testing.**

14.8.2.6 SBFIBTXSHFT: SB Intel IBIST Transmitter Inversion Shift Register

Each bit in this register enables inverting the patterns that is driven on corresponding lanes. If AUTOINVSWPEN bit is set in port control register, the TXINVSHFT field is rotated left at the completion of each pattern buffer set.

Device: NodeID
Function: 6
Offset: 94h

Bit	Attr	Default	Description
31:10	RV	000000h	**Reserved**
9:0	RWST	3FFh	**TXINVSHFT: Transmitter Inversion Shift Register.**

14.8.2.7 SBFIBRXSHFT: SB Intel IBIST Receiver Inversion Shift Register

Each bit in this register enables inverting the patterns that is received on corresponding lanes. If AUTOINVSWPEN bit is set in port control register, the RXINVSHFT field is rotated left at the completion of each pattern buffer set.

Device: NodeID
Function: 6
Offset: 98h

Bit	Attr	Default	Description
31:10	RV	000000h	**Reserved**
9:0	RWST	3FFh	**RXINVSHFT: Receiver Inversion Shift Register.**

14.8.2.8 SBFIBRXLNERR: SB Intel IBIST Receiver Lane Error Status

This records the error status from each lane.

Device: NodeID
Function: 6
Offset: 9Ch

Bit	Attr	Default	Description
31:10	RV	000000h	**Reserved**
9:0	ROST	000h	**RXERRSTAT: Receiver Error Status**

14.8.2.9 SBFIBPATTBUF2: SB Intel IBIST Pattern Buffer 2 Register

This optional register contains the pattern bits used in Intel IBIST operations. Only the least significant 24 bits are used. This user specified pattern goes out on to the link with the least-significant 12 bits as the first frame and the most significant 12 bits as the second frame.

Device: NodeID
Function: 6
Offset: A0h

Bit	Attr	Default	Description
31:24	RV	00h	**Reserved**
23:0	RWST	fd3302h	**IBPATBUF2: IBIST Pattern Buffer 2** Pattern buffer storing the default and the user programmable pattern. Default: fd3302h

14.8.2.10 SBFIBPATT2EN: SB Intel IBIST Pattern Buffer 2 Enable

This optional register specifies which lanes will carry the pattern specified in SBFIBPATTBUFF2.

Device: NodeID
Function: 6
Offset: A4h

Bit	Attr	Default	Description
31:10	RV	00h	**Reserved**
9:0	RWST	000h	**SBFIBPATT2EN: IBIST Pattern Buffer 2 enable** Per lane enable for driving pattern buffer 2.

14.8.2.11 SBFIBINIT: SB Intel IBIST Initialization Register

This register control southbound Intel IBIST Testing.

Device: NodeID
Function: 6
Offset: B0h

Bit	Attr	Default	Description
31	RV	0	**Reserved**
30:21	RWST	0c8h	**SBTS0CNT: Southbound TS0 Count** Number of TS0 sequences to transmit.
20:13	RWST	01h	**SBTS1CNT: Southbound TS1 Count** Number of TS1 sequences to transmit. If TS1CNT[7] = 1; (TS1CNT >= 128) Intel IBIST will loop forever If TS1CNT[7] = 0; (TS1CNT <128) Intel IBIST will loop TS1CNT times
12:3	RWST	100h	**SBDISCNT: Southbound Disable State Count** Number of cycles to remain in disable state.
2	RWST	1	**SBCALIBEN: Southbound Calibration Enable** 1 - Perform FBD Calibration.
1	RV	0	**Reserved**
0	RWST	0	**SBIBISTINITEN: IBIST Initialization Enable** 1 - Start IBIST Testing with Southbound Transmitter as the host.

14.8.2.12 SBIBISTMISC: SB Intel IBIST Initialization Miscellaneous Register

This register control southbound Intel IBIST Testing.

Device: NodeID
Function: 6
Offset: B4h

Bit	Attr	Default	Description
31:	RV	00h	**Reserved**
23:20	RWST	0h	**AMBID: Value of the AMB ID field during TS0**
19:0	RWST	61a80h	**SBIBISTCALPERIOD: Number of cycles to drive 1 during Calibration**

14.8.3 Northbound FBD Intel IBIST Registers

14.8.3.1 NBFIBPORTCTL: Northbound FBD Intel IBIST Port Control Register

This register controls the operation of the Intel IBIST logic.

Device: NodeID
Function: 6
Offset: C0h

Bit	Attr	Default	Description
31:24	RV	00h	**Reserved**
23	RWST	0	**SBNBMAP: Loopback mapping bit** This bit will be sent during TS1 to specify to the slave which lanes needs to be looped back. Actual lanes looped back is specified in the FBD Architecture Spec.
22	RWST	0	**CMMSTR: Compliance Measurement Mode start** This puts the Intel IBIST logic in CMM mode and Intel IBIST TX engine will start transmitting Intel IBIST patterns.
21:12	RWCST	000h	**ERRCNT: Error Counter** Total number of frames with errors that were encountered by this port.
11:8	ROST	0h	**ERRLNNUM: Error Lane Number** This points to the first lane that encountered an error. If more than one lane reports an error in a cycle, the most significant lane number that reported the error will be logged.
7:6	RWCST	00	**ERRSTAT: Port Error Status** When Intel IBIST is started, status goes to 01 until first start delimiter is received and then goes to 00 until the end or to10 if an error occurs. 00: No error. 01: Did not receive first start delimiter. 10: Transmission error (first error). 11: Reserved
5	RWST	0	**AUTOINVSWPEN: Auto Inversion sweep enable** This bit enables the inversion shift register to continuously rotate the pattern in the **FIBTXSHFT** and **FIBRXSHFT** registers. 0: Disable Auto-inversion 1: Enable Auto-inversions
4	RWST	0	**STOPONERR: Stop IBIST on Error** 0: Do not stop on error, only update error counter 1: Stop on error
3	RWST	0	**LOOPCON: Loop forever** 0: No looping 1: Loop forever

Device: NodeID Function: 6 Offset: C0h			
Bit	**Attr**	**Default**	**Description**
2	RWCST	0	**COMPLETE: IBIST done flag** 0: Not done 1: Done
1	RWST	0	**MSTRMD: Master Mode Enable** When this bit is set along with **IBISTR**, the Intel IBIST transmit engine is enabled to transmit Intel IBIST patterns. 0: Disable Master mode. 1: Enable master mode.
0	RWST	0	**IBISTR: IBIST Start** When set, Intel IBIST starts transmitting after the TS1 header is recognized during the next link initialization sequence. and **MSTRMD** bit is set.This bit also enables the receive engine to start looking for the start delimiter during TS1 training set. This bit is reset when Intel IBIST receiver is done. The Intel IBIST transmit and receive engines can be stopped by setting this bit to 0. 0: Stop IBIST transmitter 1: Start IBIST transmitter

14.8.3.2 NBFIBPGCTL: NB Intel IBIST Pattern Generator Control Register

This register contains bits to control the operation of the Intel IBIST pattern generator. All counts are in 24 bit increments.

Device: NodeID Function: 6 Offset: C4h			
Bit	**Attr**	**Default**	**Description**
31:26	RWST	0Fh	**OVRLOOPCNT: Overall Loop Count** 0h: Reserved 1h-3Fh: Number of times to loop all the patterns.
25:21	RWST	00h	**CNSTGENCNT: Constant Generator Loop Counter** 00h: Disable constant generator output 01h: 1Fh The number of times the constant generator counter pattern should loop before going to the next component. Each count represents 24 bits of 1's or 0's.
20	RWST	0	**CNSTGENSET: Constant Generator Setting** 0: Generate 0 1: Generate 1
19:13	RWST	0Fh	**MODLOOPCNT: Modulo-N Loop Counter** Each count represents 24 bits of the pattern specified by **MODPERIOD**. 00h: Disable Pattern Output 01h: 7Fh The number of times the Pattern Buffer should loop before going to the next component.
12:10	RWST	001b	**MODPERIOD: Period of the Modulo-N counter** Each encoding transmits a 24-bit pattern as specified below. All other encodings are reserved. 001: L/2 - 0101_0101_0101_0101_0101_0101 010: L/4 - 0011_0011_0011_0011_0011_0011 011: L/6 - 0001_1100_0111_0001_1100_0111 100: L/8 - 0000_1111_0000_1111_0000_1111 110: L/24 - 0000_0000_0000_1111_1111_1111
9:3	RWST	0Fh	**PATTLOOPCNT: Pattern Buffer Loop Counter** 00h: Disable Pattern Output 01h: 7Fh The number of times the Pattern Buffer (**IBPATTBUF**) should be repeated before going to the next component.

Device: NodeID Function: 6 Offset: C4h			
Bit	**Attr**	**Default**	**Description**
2:0	RWST	000	**PTGENORD: Pattern Generation Order** 000: Pattern Store + Modulo N Cntr + Constant Generator 001: Pattern Store + Constant Generator + Modulo N Cntr 010: Modulo N Cntr + Pattern Store + Constant Generator 011: Modulo N Cntr + Constant Generator + Pattern Store 100: Constant Generator + Pattern Store + Modulo N Cntr 101: Constant Generator + Modulo N Cntr + Pattern Store 110: Reserved 111: Reserved

14.8.3.3 NBFIBPATTBUF1: NB Intel IBIST Pattern Buffer 1 Register

This register contains the pattern bits used in Intel IBIST operations. Only the least significant 24 bits are used. This user specified pattern goes out on to the link with the least-significant 12 bits as the first frame and the most significant 12 bits as the second frame.

Device: NodeID Function: 6 Offset: C8h			
Bit	**Attr**	**Default**	**Description**
31:24	RV	00h	**Reserved**
23:0	RWST	02ccfdh	**IBPATBUF: IBIST Pattern Buffer** Pattern buffer storing the default and the user programmable pattern.

14.8.3.4 NBFIBTXMSK: NB Intel IBIST Transmitter Mask Register

This register enables Intel IBIST operations for individual lanes. This mask only applies to transmitters and not receivers.

Device: NodeID Function: 6 Offset: CCh			
Bit	**Attr**	**Default**	**Description**
31:14	RV	00000h	**Reserved**
13:0	RWST	3FFFh	**TXMASK: Selects which channels to enable for testing.**

14.8.3.5 NBFIBRXMSK: NB Intel IBIST Receiver Mask Register

This register enables Intel IBIST operations for individual lanes. This mask only applies to receivers and not transmitters.

Device: NodeID Function: 6 Offset: D0h			
Bit	**Attr**	**Default**	**Description**
31:14	RV	00000h	**Reserved**
13:0	RWST	3FFFh	**RXMASK: Selects which channels to enable for testing.**

14.8.3.6 NBFIBTXSHFT: NB Intel IBIST Transmitter Inversion Shift Register

Each bit in this register enables inverting the patterns that is driven on corresponding lanes. If AUTOINVSWPEN bit is set in port control register, the TXINVSHFT field is rotated left at the completion of each pattern buffer set.

Device: NodeID
Function: 6
Offset: D4h

Bit	Attr	Default	Description
31:14	RV	00000h	**Reserved**
13:0	RWST	3FFFh	**TXINVSHFT: Transmitter Inversion Shift Register.**

14.8.3.7 NBFIBRXSHFT: NB Intel IBIST Receiver Inversion Shift Register

Each bit in this register enables inverting the patterns that is received on corresponding lanes. If AUTOINVSWPEN bit is set in port control register, the RXINVSHFT field is rotated left at the completion of each pattern buffer set.

Device: NodeID
Function: 6
Offset: D8h

Bit	Attr	Default	Description
31:14	RV	00000h	**Reserved**
13:0	RWST	3FFFh	**RXINVSHFT: Receiver Inversion Shift Register.**

14.8.3.8 NBFIBRXLNERR: NB Intel IBIST Receiver Lane Error Status

This records the error status from each lane.

Device: NodeID
Function: 6
Offset: DCh

Bit	Attr	Default	Description
31:14	RV	00000h	**Reserved**
13:0	ROST	0000h	**RXERRSTAT: Receiver Error Status**

14.8.3.9 NBFIBPATTBUF2: NB Intel IBIST Pattern Buffer 2 Register

This optional register contains the pattern bits used in Intel IBIST operations. Only the least significant 24 bits are used. This user specified pattern goes out on to the link with the least-significant 12 bits as the first frame and the most significant 12 bits as the second frame.

Device: NodeID
Function: 6
Offset: E0h

Bit	Attr	Default	Description
31:24	RV	00h	**Reserved**
23:0	RWST	fd3302h	**IBPATBUF2: IBIST Pattern Buffer 2** Pattern buffer storing the default and the user programmable pattern.

14.8.3.10 NBFIBPATT2EN: NB Intel IBIST Pattern Buffer 2 Enable

This optional register specifies which lanes will carry the pattern specified in NBFIBPATTBUFF2.

Device: NodeID
Function: 6
Offset: E4h

Bit	Attr	Default	Description
15:14	RV	00	**Reserved**
13:0	RWST	0000h	**IBPATBUF2EN: IBIST Pattern Buffer 2 enable** Per lane enable for driving pattern buffer 2.

14.8.3.11 NBFIBINIT: NB Intel IBIST Initialization Register

This register control northbound Intel IBIST Testing.

Device: NodeID
Function: 6
Offset: F0h

Bit	Attr	Default	Description
31	RWST	0	**SBIDLE: SB Link is not active** This is to enable the NB link to complete training when there is no activity on the SB side. Normally NB waits for SB init to complete before proceeding with its training. 0 - Wait for SB 1 - Do not wait for SB
30:21	RWST	0c8h	**NBTS0CNT: Northbound TS0 Count** Number of TS0 sequences to transmit.
20:13	RWST	01h	**NBTS1CNT: Northbound TS1 Count** Number of TS1 sequences to transmit. If TS1CNT[7] = 1; (TS1CNT >= 128) Intel IBIST will loop forever If TS1CNT[7] = 0; (TS1CNT <128) Intel IBIST will loop TS1CNT times
12:3	RWST	100h	**NBDISCNT: Northbound Disable State Count** Number of cycles to remain in disable state.
2	RWST	1	**NBCALIBEN: Northbound Calibrating Enable** 1 - Perform FBD Calibration.
1	RV	0	**Reserved**
0	RWST	0	**NBIBISTINITEN: IBIST Initialization Enable** 1 - Start IBIST Testing with Northbound Transmitter as the host.

14.8.3.12 NBIBISTMISC: NB Intel IBIST Initialization Miscellaneous Register

This register control northbound Intel IBIST Testing.

Device: NodeID
Function: 6
Offset: F4h

Bit	Attr	Default	Description
31	RV	00h	**Reserved**
23:20	RWST	0h	**AMBID: Value of the AMB ID field during TS0**
19:0	RWST	61a80h	**NBIBISTCALPERIOD: Number of cycles to drive 1 during Calibration**

The upper bits of this register read the raw values of four of the DIFF fuses. The values are undefined until the fuses are burned. However the values of **DIFF[49:46]** used by other logic are forced to zeros while the **FUSELCK** fuse is zero. Therefore these four fuse values only have affect after the fuses have been locked. There are “chicken bits” which allow the bits to be controlled before the fuses are locked.

§

15 SPD Bits

15.1 Access Mechanisms

The tables in this section contain the Intel recomendations for the SPD bits that will be pulled from the DIMM's non-volital RAM and programmed into the Intel 6400/6402 Advanced Memory Buffer (AMB) component. This is done by the BIOS when required. There is a table for each of the raw cards that the DIMM can be based on.

15.1.1 Raw Cards A, B, and C

Table 15-1. Raw Cards A, B, and C (Sheet 1 of 2)

	C0 Stepping		D1 Stepping		
Byte	**533MHz**	**667MHz**	**533MHz**	**667MHz**	**Description**
81	0x02	0x02	0x02	0x02	FBD channel Protocols Supported
82	0x00	0x00	0x00	0x00	
83	0x10	0x10	0x10	0x10	B2B Access Turnaround Bubble (assumes at tCK)
84	0x56	0x56	0x56	0x56	CMD2DATA (800)
85	0x40	0x40	0x40	0x40	CMD2DATA (667)
86	0x36	0x36	0x36	0x36	CMD2DATA (533)
87	0x30	0x30	0x30	0x30	TR AMB (free air JEDEC)
88	0x54	0x67	0x52	0x60	AMB DT Idle_0
89	0x6E	0x7F	0x66	0x7A	AMB DT Idle_1
90	0x60	0x6E	0x60	0x6E	AMB DT Idle_2
91	0x9A	0xAA	0x84	0xA1	AMB DT Active_1
92	0x78	0x86	0x6A	0x7F	AMB DT Active_2
93	0x00	0x00	0x00	0x00	AMB DT L0s
94	0x00	0x00	0x00	0x00	reserved (TR DRAM)
95	0x00	0x00	0x00	0x00	reserved (TR AMB)
96	0x00	0x00	0x00	0x00	reserved (TR DRAM2AMB)
97	0x00	0x00	0x00	0x00	reserved (TR AMB2DRAM)
98	0x00	0x00	0x00	0x00	Tjmax
99					See Category-Byte Table for values
100	0x00	0x00	0x00	0x00	Reserved
101	0x80	0x80	0x80	0x80	AMB Personality Bytes PRE-INIT
102	0x20	0x20	0x20	0x20	
103	0x00	0x00	0x00	0x00	
104	0x44	0x44	0x44	0x44	
105	0x00	0x00	0x04	0x04	
106	0x80	0x80	0x80	0x80	

Table 15-1. Raw Cards A, B, and C (Sheet 2 of 2)

	C0 Stepping		D1 Stepping		
Byte	533MHz	667MHz	533MHz	667MHz	Description
107	0x40	0x40	0x48	0x48	AMB Personality Bytes POST-INIT
108	0x53	0x53	0x53	0x53	
109	0x00	0x00	0x33	0x33	
110	0x00	0x00	0x40	0x40	
111	0x65	0x65	0x65	0x65	
112	0x4C	0x4C	0x4C	0x4C	
113	0x00	0x00	0x00	0x00	
114	0x05	0x05	0x10	0x10	
115	0x80	0x80	0x80	0x80	AMB Manufacturer JEDEC ID
116	0x89	0x89	0x89	0x89	
150	0x00	0x00	0x01	0x01	informal AMB content revision tag (MSB)
151	0x08	0x08	0x09	0x09	informal AMB content revision tag (LSB)

15.1.2 Raw Cards D, E, H, and J

Table 15-2. Raw Cards D, E, H, and J

	C0 Stepping		D1 Stepping		
Byte	533 MHz	667 MHz	533 MHz	667 MHz	Description
81	0x02	0x02	0x02	0x02	FBD channel Protocols Supported
82	0x00	0x00	0x00	0x00	
83	0x10	0x10	0x10	0x10	B2B Access Turnaround Bubble (assumes at tCK)
84	0x58	0x58	0x58	0x58	CMD2DATA (800)
85	0x42	0x42	0x42	0x42	CMD2DATA (667)
86	0x38	0x38	0x38	0x38	CMD2DATA (533)
87	0x30	0x30	0x30	0x30	TR AMB (free air JEDEC)
88	0x5E	0x71	0x5B	0x6A	AMB DT Idle_0
89	0x76	0x89	0x71	0x84	AMB DT Idle_1
90	0x60	0x6E	0x60	0x6E	AMB DT Idle_2
91	0xA6	0xB6	0x92	0xAF	AMB DT Active_1
92	0x84	0x92	0x71	0x8B	AMB DT Active_2
93	0x00	0x00	0x00	0x00	AMB DT L0s
94	0x00	0x00	0x00	0x00	reserved (TR DRAM)
95	0x00	0x00	0x00	0x00	reserved (TR AMB)
96	0x00	0x00	0x00	0x00	reserved (TR DRAM2AMB)
97	0x00	0x00	0x00	0x00	reserved (TR AMB2DRAM)
98	0x00	0x00	0x00	0x00	Tjmax
99					See catecategory-Byte Table for values
100	0x00	0x00	0x00	0x00	Reserved

Table 15-2. Raw Cards D, E, H, and J

	C0 Stepping		D1 Stepping		
Byte	533 MHz	667 MHz	533 MHz	667 MHz	Description
101	0x80	0x80	0x80	0x80	AMB Personality Bytes PRE-INIT
102	0x20	0x20	0x20	0x20	
103	0x00	0x00	0x00	0x00	
104	0x44	0x44	0x44	0x44	
105	0x03	0x03	0x04	0x04	
106	0x80	0x80	0x80	0x80	
107	0x48	0x48	0x48	0x48	AMB Personality Bytes POST-INIT
108	0x53	0x53	0x53	0x53	
109	0x00	0x00	0x31	0x31	
110	0x00	0x00	0x40	0x40	
111	0x65	0x65	0x65	0x65	
112	0x4C	0x4C	0x4C	0x4C	
113	0x00	0x00	0x00	0x00	
114	0x05	0x05	0x10	0x10	
115	0x80	0x80	0x80	0x80	AMB Manufacturer JEDEC ID
116	0x89	0x89	0x89	0x89	
150	0x00	0x00	0x01	0x01	informal AMB content revision tag (MSB)
151	0x08	0x08	0x09	0x09	informal AMB content revision tag (LSB)

15.1.3 Category Byte 99

Table 15-3. Category ByteJ

Raw Card	airflow impedance*		DRAM type		Heat Spreader type		Byte 99	
	value	[7:6]	Value	[5:3]	Value	[2:0]	binary	hex
A	unknown	00	planar	001	AMB only	001	b00001001	0x09
A	unknown	00	planar	001	Full DIMM	010	b00001010	0x0A
B	unknown	00	planar	001	AMB only	001	b00001001	0x09
B	unknown	00	planar	001	Full DIMM	010	b00001010	0x0A
C	unknown	00	planar	001	AMB only	001	b00001001	0x09
C	unknown	00	planar	001	Full DIMM	010	b00001010	0x0A
D/J	unknown	00	stacked	011	AMB only	001	b00011001	0x19
D/J	unknown	00	stacked	011	Full DIMM	010	b00011010	0x1A
D/J	unknown	00	dual die	010	AMB only	001	b00010001	0x11
D/J	unknown	00	dual die	010	Full DIMM	010	b00010010	0x12
E/H	unknown	00	planar	001	AMB only	001	b00001001	0x09
E/H	unknown	00	planar	001	Full DIMM	010	b00001010	0x0A

Note: *Airflow Impedance value is the result of air flow resistance measurements and bin bands defined by FBDIMM customers

(Contact your customer for measurement method/setup if you want to program known air flow impedance values otherwise program to the b00 default.)

§

A Glossary

A.1 Terms and Definitions

Term	Definition
ADC	Analog to Digital Converter
AMB	Advanced Memory Buffer
APM	Autonomous Platform Manager
ATE	Automatic Test Equipment
BIST	Built-In Self-Test
BL4/BL8	DRAM Burst Length 4/8
CAS	Column Address Strobe
CBC	Chip Boot Configuration
Chip disable	An ECC encoding specifically tailored for memory such that the data from any defective memory device can be reconstructed from some aggregate of surviving memory devices. Corrects data from failed device. The AMB employs an “x8” ECC, which means that all data from a partially or completely failed 8-bit device can be recovered without stopping the system. This same ECC provides the same level of coverage for 4-bit (“x4”) devices.
CRC	Cyclic Redundancy Code(s). Usually in reference to binary error detection algorithms.
CSR	Configuration and Status Register(s)
DDR	Double Data Rate (SDRAM)
DDR Branch	The minimum aggregation of DDR channels which operate in lock-step to support error correction. Two channels per branch supports x8 chip disable ECC. A rank spans a branch.
DDR Channel	A DDR channel consists of a data channel with 72 bits of data and an ADDR/CNTRL channel
DDR Data channel	A DDR data channel consists of 72 bits of data, divided into 18 data groups
DDR Data group	Each data group consists of 4 data signals and a differential strobe pair
DFx	Design for Test/Manufacturability/Validation
DFV	Design For Validation
DIMM	Dual In-Line Memory Module. A packaging arrangement of memory devices on a socketable substrate.
DIMM Slot	Receptacle (socket) for a DIMM. Also, the relative physical location of a specific DIMM on a DDR channel.
DIMM Stack	Dual-ranked x4 DRAM DIMM physical topology: refers to two physical rows of DRAM “stacked” one above another
DLL	Delay locked loop
DPM	Defects per million
DRAM	Dynamic Random Access Memory
DRAM Page (Row)	The DRAM cells selected by the Row Address
ECC	Error Correction Code. For the AMB, this is a chip disable code.
EEPROM	Electrically Erasable Programmable Read Only Memory
EMask, EMASK	Error Mask

Term	Definition
EM	Electromagnetic
EMI	Electromagnetic interference
FBD	Fully-Buffered DIMM
FBD Channel	Combination of 10 lane Southbound Links and 13 or 14 lane Northbound Links that make up a logical memory channel from host perspective
FBDIMM	Fully-Buffered DIMM
FERR	First Error
FIFO	First-In, First-Out. Usually in reference to a buffer implementation.
Frame	Group of bits containing commands or data
FSM	Finite State Machine
HCSL	High-speed Current Steering Logic
Host	Memory controller agent on an FBD channel
IBIST	Interconnect Built-In Self-Test (built-in interconnect test)
I/O	Input Output. Usually in reference to a bidirectional buffer cell or circuit.
ISI	Inter Symbol Interference – see section “Initialization / Clocking” for definition
JEDEC	JEDEC Solid State Technology Association (once known as the Joint Electron Device Engineering Council)
JESD79	JEDEC Standard 79, *DDR SDRAM Specification*
JTAG	Joint Test Action Group. Usually in reference to the IEEE 1149.1a boundary scan test standard.
LA	Logic Analyzer
LAI	Logic Analyzer Interface
Lane	Differential pair of receivers or transmitters
LFSR	Linear Feedback Shift Register. Usually in reference to pseudo-random bit-stream data.
Link	High speed parallel Differential Point-to-Point interface
MC	Memory interface Control (block)
MemBIST	Memory Built-In Self-Test
Mesochronous	Small ppm frequency difference.
MTBF	Mean Time Between Failures
MT/s	Mega Transfers per Second
NACK	Not-ACKnowledge. Usually in reference to SMBus communications.
NB	Northbound
NBI	Northbound Interface. (also: Northbound link control logic)
NERR	Next/Subsequent Error
Northbound	The direction of signals running from the furthest DIMM toward the host.
ODT	On-Die Termination
Page Replace aka Page Miss, Row Hit / Page Miss	An access to a row that has another page open. The page must be transferred back from the sense amps to the array, and the bank must be precharged.
Page Hit	An access to an open page, or DRAM row. The data can be supplied from the sense amps at low latency.
Page Miss (Empty Page)	An access to a page that is not buffered in sense amps and must be fetched from DRAM array.
PEC	Packet Error Code
Plesiochronous	Having the same frequency but arbitrary phase differences.
PLL	Phase Locked Loop

Term	Definition
POR	Plan Of Record
Primary	In the direction towards the Host controller
PTH	Plated Through-Hole
PVT	Process, Voltage and Temperature
Rank	A DIMM is organized as one or two physical sets of memory, called ranks. Note that single rank or dual rank is different from single-sided or double-sided, for example, a single rank DIMM build from x4 DRAM devices is actually double-sided. It is also common practice to distribute the 9 devices of an x8 DIMM between both sides of the DIMM to enhance the thermal performance of the module. The standard 4-slot DDR2 topology is limited to single rank DIMM due to loading constraints.
RAS	Reliability, Availability, Serviceability. (also: Row Address Strobe - meaning is context dependent.)
Resample	A resampler is a serial data in and serial data out node that attenuates jitter by regenerating the serial data using a clock recovered from the incoming data stream derived from a common reference clock. It also resets the voltage budget of the retransmitted data.
Resync	A resync repeater is a serial data in and serial data out node that resynchronizes data to a local clock after it has been sampled with a recovered clock derived from a common reference clock. The local clock is also generated from the same reference clock by a PLL multiplier. A drift compensation buffer is inserted between the two clock domains which absorbs the maximum link delay change over worst case voltage and temperature changes. Both the jitter and voltage budgets for the retransmitted data are reset.
RPD	Return Path Discontinuity
SB	Southbound
SBI	Southbound Interface. (also: Southbound link control logic)
SDR	Single Data Rate
SDRAM	Synchronous Dynamic Random Access Memory
Secondary	In the direction away from the Host controller
Seed, SEED	The starting or “seed” value used to algorithmically generate pseudo-random data. Usually in reference to a LFSR implementation.
Serial Present Detect (aka SMBus protocol	A 2-signal serial bus used to read and write control registers in the AMB and SDRAM
SI	Signal Integrity
SMBus, SMBUS	System Management Bus. Mastered by a system management controller to read and write configuration registers. Limited to 100 kHz.
Southbound	The direction of signals running from the host controller toward the DIMMs.
SPD	Serial Presence Detect. Usually in reference to the serial ROM on a socketable DIMM which contains information specific to the unit.
SPOF	Single Point Of Failure
SSC	Spread Spectrum Clocking. Usually used to lower average EM emissions.
SSO	Simultaneously Switching Outputs
SSTL_18	Series Stub Terminated Logic for 1.8 V
Throttled	Temporarily prohibiting memory accesses when a thermal or electrical limit has been reached.
Transparent Mode	High speed FBD linked I/Os are bypassed with lower speed CMOS I/Os to allow lower speed tester to control and test DRAMs on the DIMM.
Unit Interval	Average time interval between voltage transitions of a signal
VCO	Voltage Controlled Oscillator
V_{DDQ}	I/O buffer voltage for DDR2 buffers. Nominally 1.8 V
V_{SS}	Ground (0.0 V)

§