JEDEC STANDARD

FBDIMM: Architecture and Protocol

JESD206

SPECIAL DISCLAIMER: JEDEC has received information that certain patents or patent applications may be relevant to this standard, and, as of the publication date of this standard, no statements regarding an assurance or refusal to license such patents or patent applications have been provided.

http://www.jedec.org/download/search/FBDIMM/Patents.xls

JEDEC does not make any determination as to the validity or relevancy of such patents or patent applications. Prospective users of the standard should act accordingly.

JANUARY 2007

JEDEC SOLID STATE TECHNOLOGY ASSOCIATION

Electronic Industries Alliance

Samsung Electronics Co., Ltd.
Ex. 1025, Cover-1

NOTICE

JEDEC standards and publications contain material that has been prepared, reviewed, and approved through the JEDEC Council level and subsequently reviewed and approved by the EIA General Counsel.

JEDEC standards and publications are designed to serve the public interest through eliminating misunderstandings between manufacturers and purchasers, facilitating interchangeability and improvement of products, and assisting the purchaser in selecting and obtaining with minimum delay the proper product for use by those other than JEDEC members, whether the standard is to be used either domestically or internationally.

JEDEC standards and publications are adopted without regard to whether or not their adoption may involve patents or articles, materials, or processes. By such action JEDEC does not assume any liability to any patent owner, nor does it assume any obligation whatever to parties adopting the JEDEC standards or publications.

The information included in JEDEC standards and publications represents a sound approach to product specification and application, principally from the solid state device manufacturer viewpoint.

No claims to be in conformance with this standard may be made unless all requirements stated in the standard are met.

Inquiries, comments, and suggestions relative to the content of this JEDEC standard or publication should be addressed to JEDEC Solid State Technology Association, 2500 Wilson Boulevard, Arlington, VA 22201-3834, (703)907-7559 or www.jedec.org.

Published by

©JEDEC Solid State Technology Association 2007
2500 Wilson Boulevard
Arlington, VA 22201-3834

This document may be downloaded free of charge, however JEDEC retains the copyright on this material. By downloading this file the individual agrees not to charge for or resell the resulting material.

Price: Please refer to the current
Catalog of JEDEC Engineering Standards and Publications or call Global Engineering Documents, USA and Canada (1-800-854-7179), International (303-397-7956)

Printed in the U.S.A.
All rights reserved

PLEASE!

DON'T VIOLATE
THE
LAW!

This document is copyrighted by the JEDEC Solid State Technology Association
and may not be reproduced without permission.

Organizations may obtain permission to reproduce a limited number of copies
through entering into a license agreement. For information, contact:

JEDEC Solid State Technology Association
2500 Wilson Boulevard
Arlington, Virginia 22201-3834
or call (703) 907-7559

Special Disclaimer

JEDEC has received information that certain patents or patent applications may be relevant to this standard, and, as of the publication date of this standard, no statements regarding an assurance or refusal to license such patents or patent applications have been provided.

http://www.jedec.org/download/search/FBDIMM/Patents.xls

JEDEC does not make any determination as to the validity or relevancy of such patents or patent applications. Prospective users of the standard should act accordingly.

FBDIMM Architecture and Protocol
Contents

1 Document Organization 1

1.1 List of Terms and Abbreviations 1
1.2 Revision History 3
1.3 Related Documents 4

2 Fully Buffered DIMM Overview 5

2.1 Memory Channel 5
2.1.1 Link Widths 5
2.1.2 Topologies 6
2.1.3 AMB Addressing 7
2.1.4 SMBus Interface 8
2.2 Physical Layer 8
2.3 Clocking 8
2.3.1 Clock Distribution 9
2.3.2 Receiver Data Sampling 9
2.3.3 Voltage Temperature Compensation FIFO 10
2.3.4 Daisy-Chain Retiming and Data Merge 10
2.3.5 DRAM Clock Generation 10
2.3.6 DRAM Data Return 10
2.4 Host Memory Interface 11
2.5 Advanced Memory Buffer (AMB) 11
2.6 Options 12
2.6.1 Number of FBD Lanes 12
2.6.2 ECC and non-ECC DIMMs 13
2.6.3 Variable Read Latency 13
2.6.4 L0s State 13
2.6.5 Protocol Variants 13

3 Channel Initialization 15

3.1 Reset and Inband Control “Signals” 16
3.1.1 RESET# Signal 17
3.1.2 Inband Control “Signals” 17
3.1.3 Inband Reset Event Detector 17
3.1.4 Inband Calibrate Event Detector 18
3.2 Training Sequence Ordered-sets 18
3.3 Channel Initialization Sequence 19
3.3.1 Firmware Transition Control 23
3.3.2 AMB Internal State Variables 23
3.3.3 Disable State 24
3.3.4 Training State 26
3.3.5 Testing State 29
3.3.6 Polling State 32
3.3.7 Config State 36
3.3.8 L0 State 42
3.3.9 Calibrate State 44
3.3.10 L0s State 46
3.4 Channel Re-initialization 48
3.4.1 Enter Self Refresh FSM 48
3.4.2 Fast Reset Flow 49
3.5 Hot-add 51

Samsung Electronics Co., Ltd.
Ex. 1025, p. i

3.5.1 Hot-add AMB Reset 51
3.5.2 Hot-add AMB Calibration 51
3.5.3 Hot-add AMB Testing 52
3.5.4 Hot-add AMB Timing 52
3.5.5 Hot-Add Reset Flow 52
3.6 Hot-remove 54
3.7 Hot-replace 54

4 Channel Protocol 55

4.1 Southbound Frames 55
4.1.1 Normal Southbound Frames 55
4.1.2 Fail-over Southbound Frames 56
4.1.3 Southbound Frame Formats 57
4.2 Southbound Commands 61
4.2.1 Command Delivery Timing 61
4.2.2 Concurrent Command Delivery Rules 62
4.2.3 Command Encoding 63
4.2.4 DRAM Commands 64
4.2.5 Channel Commands 64
4.3 Northbound Frames 71
4.3.1 Northbound CRC Modes 71
4.3.2 Northbound Idle Frame 71
4.3.3 Northbound Alert Frame 73
4.3.4 Northbound Data Frames 75
4.3.5 Northbound Status Frame 80
4.4 DRAM Memory Timing 81
4.4.1 Read Timing 82
4.4.2 Write Timing 82
4.4.3 Simultaneous Read and Write Data Transfers 85
4.4.4 DRAM Bus Segment Restrictions 85

5 Reliability, Availability and Serviceability 87

5.1 Overview 87
5.2 Example Error Flows 87
5.2.1 Command Error Flow 87
5.2.2 Write Data Error Flow 87
5.2.3 Read Error Flow 88
5.3 Overview of Error Protection, Detection, Correction, and Logging 88
5.4 Error Protection and Detection Methods 90
5.4.1 CRC Logic Used on Normal Southbound Frames 90
5.4.2 Fail-over Southbound Frames 98
5.4.3 Fail-over Southbound Frame CRCs 101
5.4.4 CRC Generation: 14-bit Lane Northbound Data Frame 104
5.4.5 13-bit Lane Northbound Data Frame 107
5.4.6 12-bit Lane Northbound Data Frame 109
5.4.7 Write and Read Data ECC Error Protection 110
5.5 Southbound Error Handling at the AMB 110
5.5.1 Exiting Command Error State 111
5.6 Northbound Error Handling at the AMB 111
5.7 Northbound Error Handling at the Host 111
5.7.1 Read Return Burst Management 111
5.7.2 Read Response Data Error Handling 112
5.8 Error Logging 112

Samsung Electronics Co., Ltd.
Ex. 1025, p. ii

5.8.1 Logging of AMB Southbound Events 112
5.8.2 Logging of Host Northbound Events 112
5.9 Error Injection 112
5.10 Fail-over Mode Operation 113
5.10.1 Fail-over Mode Operation on Southbound Lanes 113
5.10.2 Fail-over Mode Operation on Northbound Lanes 113
5.11 AMB Pass-through Functionality 113
5.12 Hot Add and Remove 114
5.12.1 Hot Add Sequence 114
5.12.2 Hot Remove Sequence 114
5.12.3 Hot Replace Sequence 115
5.13 Memory Initialization 115
5.14 Thermal Trip Sensor 115

Samsung Electronics Co., Ltd.
Ex. 1025, p. iii

FBDIMM Architecture and Protocol

List of Tables

1-1 Terms and Definitions ... 1
1-2 Revision History ... 3
1-3 Related Documents ... 4
2-4 AMB Addressing ... 7
2-5 FBD DIMM Port High-Speed Signal Pin Count (Full 14-bit NB Option) ... 8
2-6 Northbound Bit Lane Options ... 12
3-7 Channel Initialization Timing Parameters ... 16
3-8 Disable State ... 25
3-9 Training Sequence – TS0 (Training) ... 26
3-10 Training Duration Estimate at DDR2-667 ... 26
3-11 Training State ... 28
3-12 SB to NB Bit Lane Mapping ... 30
3-13 Training Sequence – TS1 (Testing) ... 31
3-14 Testing State ... 32
3-15 NB_Width_Capability Field ... 33
3-16 Training Sequence – TS2 (Polling) ... 33
3-17 Polling State ... 34
3-18 Training Sequence – TS3 (Config) ... 37
3-19 Protocol Selection Field ... 37
3-20 Config State ... 38
3-21 NB Channel Width Selection ... 39
3-22 SB_Config Field ... 40
3-23 NB_Config Field ... 40
3-24 L0 State ... 43
3-25 Calibrate State ... 44
3-26 Recalibrate State ... 45
3-27 L0s State ... 47
3-28 Self Refresh Entry ... 49
3-29 Fast Reset Duration Estimate for a 12 AMB system ... 51
4-30 Common Features of Normal Southbound Frames ... 56
4-31 Common Features of Fail-over Southbound Frames ... 57
4-32 Southbound Frame Type Encoding ... 57
4-33 Command Frame Format ... 58
4-34 Command Frame with Data Format ... 59
4-35 Wdata Address Delivery ... 59
4-36 Command+Wdata Frame Format (4-bit Devices) ... 60
4-37 Command+Wdata Frame Format (8-bit Devices) ... 60
4-38 Command+Wdata Frame Format Non-ECC 8-bit Devices) ... 61
4-39 Command Encoding ... 63
4-40 DRAM Command Mapping Examples ... 64
4-41 Channel Command Encoding ... 65
4-42 Table 4-13: Per Rank CKE Command Bit Positions ... 67
4-43 Soft Channel Reset Command Frame Format ... 68
4-44 Sync Command Frame Format ... 70
4-45 NOP Frame Format ... 70
4-46 First Northbound Idle Frame Format ... 72
4-47 Second Northbound Idle Frame Format ... 72
4-48 Third Northbound Idle Frame Format ... 73
4-49 Fourth Northbound Idle Frame Format ... 73
4-50 Alert Frame Replacing Second Idle Frame ... 75
4-51 14-bit Northbound Data Frame Format (with 4-bit Devices) ... 76

Samsung Electronics Co., Ltd.
Ex. 1025, p. iv

4-52 13-bit Northbound Data Frame Format (with 8-bit Devices) 77
4-53 13-bit Northbound Fail-Over Data Frame Format (8 bit Devices) 78
4-54 12-bit Northbound Data Frame Format (with Non-ECC 16-bit Devices) 79
4-55 Northbound Register Data Frame Format 79
4-56 Northbound Status Frame Format 80
4-57 Status Bit Description 81
5-58 AMB Responses to Error Conditions 89
5-59 Host Responses to Error Conditions 89
5-60 Common Features of Normal Southbound Frames 91
5-61 Southbound Frame with 72 bit Write Data (4-bit Devices) 93
5-62 Southbound Frame with Three Commands 98
5-63 Common Features of Fail-over Southbound Frames (with 8-bit Devices) 99
5-64 Fail-over Southbound Frame With Three Commands 104
5-65 14-bit Northbound Data Frame Format (with 4-bit Devices) 105
5-66 13-bit Northbound Data Frame Format (with 8-bit Devices) 108
5-67 12-bit Northbound Data Frame Format (with 16-bit Devices) 110

Samsung Electronics Co., Ltd.
Ex. 1025, p. v

FBDIMM Architecture and Protocol

List of Figures

2-1 FBD Channel Southbound and Northbound Paths 6
2-2 Sample FBD Channel Topologies 7
2-3 Clock Distribution Block Diagram 9
3-4 AMB Initialization Flow Diagram 15
3-5 In-band Control Signal Detectors 18
3-6 Initialization Sequence Flow 20
3-7 Alternative AMB Discovery Sequence Flows 22
3-8 Figure 3-5: Bit Lane Fail over Mechanism Example 42
3-9 Recalibrate Sequence Flow 46
3-10 L0s Sequence Flow 48
3-11 Fast Reset Sequence Flow 50
3-12 Hot-Add Reset Sequence Flow if Failure 53
4-13 “A”, “B”, and “C” Command Delivery Timing 62
4-14 Idle Frame LFSR Counter Example 71
4-15 Basic DRAM Read Data Transfers on FBD (RD) 82
4-16 FBack-to-back DRAM Read Data Transfers on FBD (RD-RD) 82
4-17 Basic DRAM Write Data Transfers on FBD (WR) 83
4-18 Write FIFO Minimum Fall Through Time 84
4-19 Write FIFO Maximum Fall Through Time 84
4-20 Write FIFO Maximum Pipelining of Writes for a Single DIMM 85
4-21 Simultaneous Read and Write Data Transfers on FBD (RD-WR-RD) 85

Samsung Electronics Co., Ltd.
Ex. 1025, p. vi

FBDIMM: Architecture and Protocol

(From JEDEC Board Ballot, JCB-06-49, formulated under the cognizance of the JC-40 Committee on Digital Logic.)

1 Document Organization

The next four chapters of the FBD Channel Specification cover Channel Overview (Chapter 2), Initialization (Chapter 3), Channel Protocol (Chapter 4) and Reliability, Availability, and Serviceability (RAS) (Chapter 5).

1.1 List of Terms and Abbreviations

This document uses the following terms and abbreviations:

Note that the terms chipset and memory controller are used interchangeably throughout the rest of this document. The term motherboard is used as a generic term to describe the PCB onto which the memory controller is mounted. Actual implementations could have distributed memory controllers mounted on separate memory boards.

Table 1-1 — Terms and Definitions

TERM	Definition
AMB	Advanced Memory Buffer
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.
Bit Lane	A differential pair of signals in one direction.
D+ and D-	The D+ and D- terms used in this document are used to indicate the two conductors or signals of a differential signaling pair.
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
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
DRAM Page (Row)	The DRAM cells selected by the Row Address
DPM	Defects per million
DRAM	Dynamic Random Access Memory
ECC	Error Correction Code.
EMI	Electro-magnetic interference
FBD	Fully Buffered DIMM

Table 1-1 — Terms and Definitions (Cont'd)

TERM	Definition
Frame	Group of bits containing commands or data
HCSL	High-speed Current Steering Logic
Host	Memory controller agent on an FBD channel
ISI	Inter Symbol Interference
JEDEC	JEDEC Solid State Technology Association (once known as the Joint Electron Device Engineering Council)
JESD79	JEDEC Standard 79, DDR SDRAM Specification
Link	A dual-simplex communications path between two components. The collection of two Ports and their interconnecting bit lanes.
Mesochronous	Same frequency with unknown (but fixed) phase relation.
NB	Northbound
Northbound	The direction of signals running from the farthest DIMM toward the host.
Page Replace a.k.a. 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 pre-charged.
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.
PLL	Phase Locked Loop
Port	In physical terms, a group of transmitters and receivers physically located on the same chip that define one end of a Link.
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, e.g. 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.
Resample	A resampler repeater is a serial data in and serial data out node that attenuates jitter by re-generating 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 re-transmitted data.
Resync	A resync repeater is a serial data in and serial data out node that re-synchronizes 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 re-transmitted data are reset.
RPD	Return Path Discontinuity
SB	Southbound
SDRAM	Synchronous Dynamic Random Access Memory
SI	Signal Integrity
Serial Present Detect (aka SMBus protocol)	A 2-signal serial bus used to read and write control registers in the AMB and SDRAM

Table 1-1 — Terms and Definitions (Cont'd)

TERM	Definition
SMBus	System Management Bus. Mastered by a system management controller to read and write configuration registers. Limited to 100KHz.
Southbound	The direction of signals running from the host controller toward the DIMMs.
SSC	Spread Spectrum Clocking. Utilized to lower EMI
SSO	Simultaneously Switching Outputs
SSTL_18	Series Stub Terminated Logic for 1.8V
Throttled	Temporarily prohibiting memory accesses when a thermal or electrical limit has been reached.
Unit Interval	Average time interval between voltage transitions of a signal
Vss	Ground (0V)
Vddq	I/O buffer voltage for DDR-II buffers. Nominally 1.8V

1.2 Revision History

This document has been released in the following revisions:

Table 1-2 — Revision History

Version	Date	Changes
0.0	12/3/2003	Original revision
0.1	2/13/2004	Revision of all chapters to incorporate feedback
0.1a	5/3/2004	Added JEDEC Ballot Comments 2.1.3: added detail to SMBus addressing 3. Increased calibration time 3.1.2: added detail on determining clock train violaiton 3.3.3: Tx termination to remain on in disable state 3.3.3: Clarified disable state actions by reording table 3-8 3.3.4 & 3.3.5: Command to data fraction delay added in. 3.3.5 & 3.3.7: AMB with Last_AMB flag set loops back data 3.3.6: specified host operation in sending TS2 patterns 3.3.6: Clarified Mege_Disable bit funtionality 3.3.9.1: De-Emphasis disabled during calibrate 4.2.5 Table 4-41: corrected ERC and EL0s duration
	3/16/2005	Added Editorial changes from September 2004 JEDEC meeting Added Ballot changes from March 2005 JEDEC meeting

1.3 Related Documents

Table 1-3 — Related Documents

Document	Revision	Description
FBD Connector Specification	Na	Connector physical parameters: pinout, footprint, mechanical drawing, and requirements
FBD Signaling Specification		PTP link parameters: signaling, I/O, and AC and DC parameters
FBD AMB Specification	Na	AMB Characteristics: pinout, package type, mechanical outline, footprint, AC/DC specs, power/thermal requirements, buffer TPT, special feature requirements (e.g. thermal sensor), & basics DFT
FBD DIMM Specification	Na	FBD DIMM module parameters: multiple raw card designs, block diagrams, net topologies, routing details, timing budget, pin out, mech. Outline, stack up, and SPD requirements
SMBus	2.0	System Management Bus Specification http://smbus.org/specs/smbus20.pdf
DDRII JEDEC Spec		JEDEC DDR II SDRAM Data Sheet JC-42.3

2 Fully Buffered DIMM Overview

Higher CPU speeds are driving the need for higher memory bandwidth. Looking beyond DDR2-533, the existing "stub bus" architecture becomes unworkable. The number of DRAM devices per channel has been trending toward fewer devices per channel. This reduction in capacity reduces performance. Platform cost restrictions limit DRAM pin count increases, routing restrictions limits adding channels to maintain capacity, and differing DRAM interfaces require unique motherboard developments and products.

Fully Buffered DIMM (FBD) addresses these requirements by providing a high-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 that creates a pay-as-you-go cost structure. Memory device capacity remains high and total memory capacity scales with DRAM bit density.

This chapter describes the FBD Channel topology, physical signaling, clocking and data flow.

2.1 Memory Channel

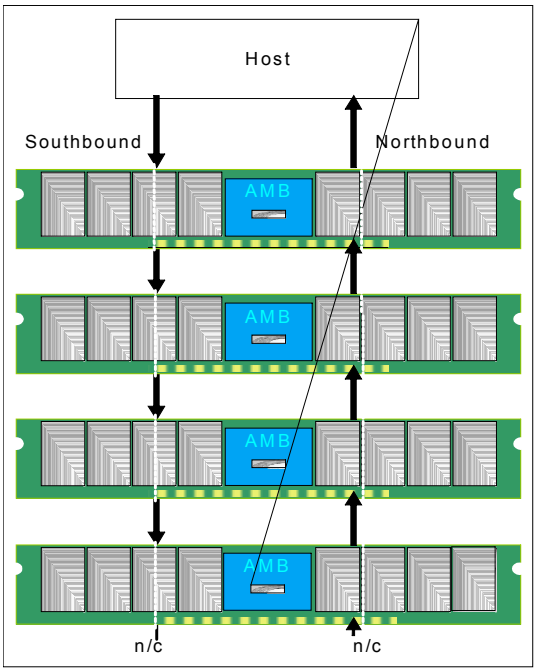
The FBD channel provides a communication path from a host controller to an array of DRAM devices. The DRAM devices are buffered behind one or more Advanced Memory Buffer (AMB) devices. The physical isolation of the DRAM devices from the channel enables the flexibility to enhance the communication path to significantly increase the reliability and availability of the memory subsystem.

2.1.1 Link Widths

Figure 2-1 shows a logical diagram of the FBD channel's southbound and northbound data paths. The channel interconnect actually consists of two unidirectional links: one in the southbound direction and the other in the northbound direction. The southbound data path running from the host to the DIMMs is 10 logical signals wide, and the northbound data path running from the DIMMs back to the host is 14, 13, or 12 logical signals wide. The width of the northbound link is application dependent. The host may support the link width or widths needed to provide the appropriate error detection coverage needed for the intended application. An AMB device on a non-ECC DIMM must support the 12 logical signal width. An AMB device on an ECC DIMM must support the 13 and 14 bit widths.

The southbound link may be operated in fail-over mode to map out a bad bit lane if supported by the host. The 14 or 13 bit width northbound link may be operated in fail-over mode to map out a bad bit lane if supported by the host. An AMB is required to support fail-over on the southbound link, and is required to support fail-over on the northbound link if ECC mode is supported (14 and 13 bit widths).

Figure 2-1 — FBD Channel Southbound and Northbound Paths

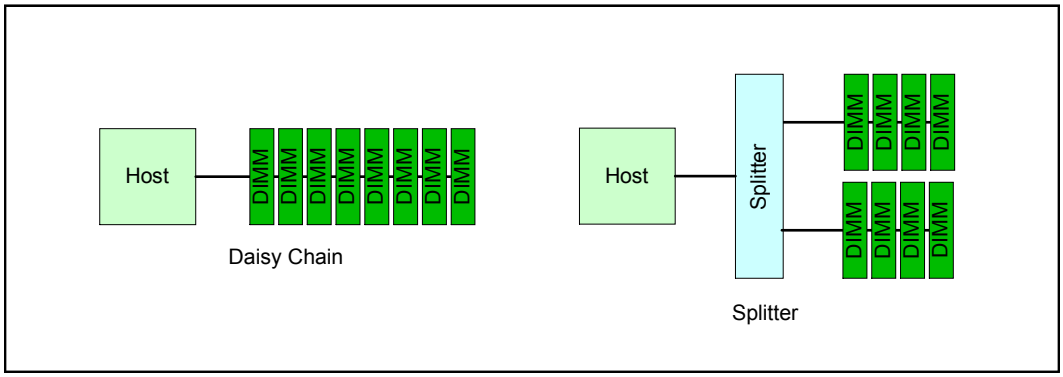


2.1.2 Topologies

The FBD channel uses a daisy-chain topology to provide expansion from a single DIMM per channel to up to 8 DIMMs per channel. The host sends data on the southbound link to the first DIMM where it is received and re-driven to the second DIMM. On the southbound data path each DIMM receives the data and again re-drives the data to the next DIMM until the last DIMM receives the data. The last DIMM in the chain initiates the transmission of data in the direction of the host (a.k.a. northbound). On the northbound data path each DIMM receives the data and re-drives the data to the next DIMM until the host is reached.

More than 8 buffer devices may be located on a channel. Up to 8 of these may be DRAM DIMMs and the remainder may be repeaters, splitters, or logic analyzer buffers. Figure 2-2 shows two possible topologies that could be supported by FBD. The Daisy Chain topology has the lowest initial cost but latency to the last DIMM is longer than in the splitter topology. The splitter topology has higher initial latency and cost but provides lower latency when fully populated. While a splitter device is theoretically possible, this specification does not define the mechanisms needed to support a splitter.

Figure 2-2 — Sample FBD Channel Topologies



2.1.3 AMB Addressing

Each AMB on the channel must have a unique identity to be addressable by the system. The four SA[3:0] signal pins on the AMB determines its function and address. Table 2-4 defines the function of the AMB based upon the value on the SA[3:0] selection pins. The most significant pins determine if the AMB is mapped into the address space as a DRAM buffer, a repeater, or a Logic Analyzer Interface Buffer. For DIMM modules the value on the lower 3 pins is determined by the value on the corresponding SA[2:0] DIMM connector pins. These are shared with the Serial Presence Detect EEPROM and give the AMB and the EPROM unique SMBus addresses. The SA[3:0] pins also indicate to the AMBs which DIMM Select (DS [3:0] field) they should respond to during FBD DRAM and Channel commands. Only the DRAM buffers and the repeaters are addressable on the FBD channel, the Logic Analyzer Interface buffers are only addressable through the SMBus. The DS[3:0] field in FBD commands is not used by the Logic Analyzer Interface buffers and this allows the DS[3:0] value of 0Fh to be used as a broadcast address to all DRAM DIMM and Repeater AMBs on the channel for Write Config Reg commands.

Table 2-4 — AMB Addressing

AMB BFUNC pin	AMB SA[2:0] pins	DIMM SA[2:0] pins	SMBus Type/ Addr	DS[3:0] command field	Status Return Lane	AMB Function
0	000	000	B0h / B1h	0	0	DRAM DIMM 0
0	001	001	B2h / B3h	1	1	DRAM DIMM 1
0	010	010	B4h / B5h	2	2	DRAM DIMM 2
0	011	011	B68 / B7h	3	3	DRAM DIMM 3
0	100	100	B8h / B9h	4	4	DRAM DIMM 4
0	101	101	BAh / BBh	5	5	DRAM DIMM 5
0	110	110	BCh / BDh	6	6	DRAM DIMM 6
0	111	111	BEh / BFh	7	7	DRAM DIMM 7
1	000	n/a	30h / 31h	8	8	Repeater 0
1	001	n/a	32h / 33h	9	9	Repeater 1
1	010	n/a	34h / 35h	A	10	Repeater 2
1	011	n/a	36h / 37h	B	11	Repeater 3
1	100	n/a	38h / 39h	n/a	none	Logic Analyzer Interface 0
1	101	n/a	3Ah / 3Bh	n/a	none	Logic Analyzer Interface 1
1	110	n/a	3Ch / 3Dh	n/a	none	Logic Analyzer Interface 2
1	111	n/a	3Eh / 3Fh	n/a	none	Logic Analyzer Interface 3

The SMBus Type/Addr column is shown with the Read/Write bit included as the LSB. It is constructed as: {[!BFUNC],[DEV TYPE],SA[2:0],RW} where BFUNC is 0 on DRAM buffers and 1 on repeaters and logic analyzer interfaces. DEV TYPE is three bit field defined to be 3b011 for buffer (AMB) devices. RW is the read/write bit.

Note that DRAM commands on the channel only specify DS[2:0] since only AMB addresses 0 through 7 are used as DIMMs. DS[3] is assumed to be 0 for these commands. The Write Config Reg and Read Config Reg commands specify DS[3:0]

The Status Return Lane is the northbound lane onto which the AMB will merge its data during a Status Frame, which is in response to a southbound Sync Frame.

By convention the AMB addresses increase the farther away from the host the AMB is located on the channel. System firmware may be simplified if the system vendor adheres to this convention.

2.1.4 SMBus Interface

Each AMB on the channel must have an SMBus controller. The SMBus controller in each AMB is a slave only controller and is used by the system as an out-of-band access path to the configuration registers. This out-of-band path is available to the system to set initialization parameters, to facilitate AMB/DIMM testing, and provide enhanced fault isolation. Conflicts between in-band and out-of-band accesses to configuration registers must be avoided at the system level.

2.2 Physical Layer

The physical layer of the FBD channel uses high-speed point-to-point links. The signaling is dual simplex, allowing simultaneous data communications in both directions, over low-voltage differential pairs. This section gives a short overview of the FBD channel physical layer. For details refer to the *FBD Signaling Specification*.

Table 2-2 shows the total number of high-speed signals and the corresponding number of physical pins required for the full 14-bit northbound FBD port option on the FBD DIMM. As the signals are differential each high-speed signal requires two pins. The high-speed signals include the 10 southbound signal pairs that carry command, address, data and CRC, and the 14 northbound signal pairs that carry status, data and CRC. In addition there is one signal pair in each direction reserved for future use as forwarded clocks. This makes the total number of high-speed signals equal to 52 per port on the FDB DIMM.

In addition to the high-speed FBD channel signals, each FBD channel agent receives a pair of differential reference clock signals from a common clock source. Each AMB must also support an SMBus port.

Table 2-5 — FBD DIMM Port High-Speed Signal Pin Count (Full 14-bit NB Option)

	Southbound		Northbound	
	Signals	**Pins**	**Signals**	**Pins**
Data path	10	20	14	28
Spare pins for forwarded clocks[a]	1	2	1	2
Total high speed channel pins	52			

a. The current FBD physical layer does not utilize forwarded clocks. To allow the potential use of forwarded clocks in a future version of the FBD physical layer, pins at the FBD DIMM connector are held in reserve for that purpose.

2.3 Clocking

This chapter describes the FBD channel clocking architecture. The two main aspects of clocking are clock distribution and data sampling at the input receivers.

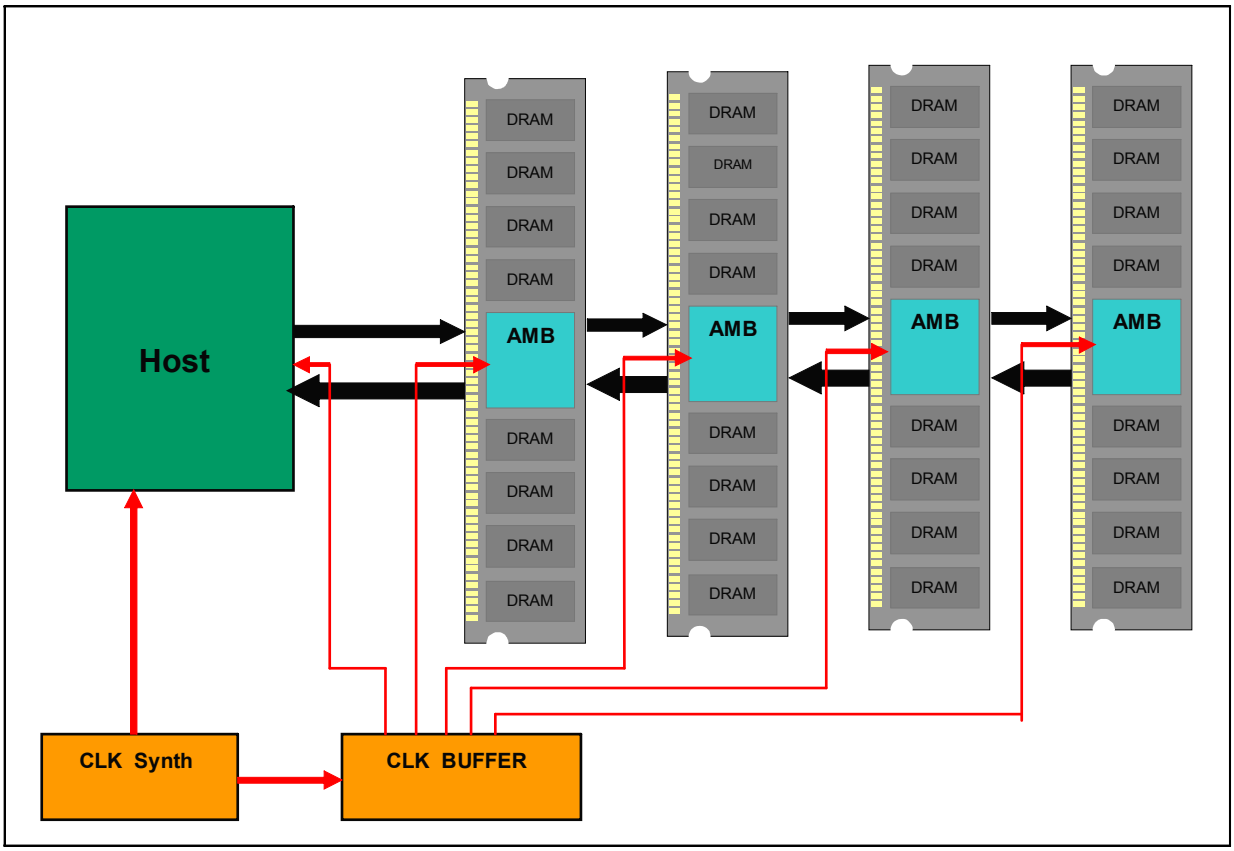
2.3.1 Clock Distribution

A low-jitter reference clock is routed to the host and each AMB from a common clock source on the system board. The reference clock is used to generate the internal buffer clocks and the clocks to the DRAMs located on each DIMM. The frequency of the reference clocks is equal to the DRAM base clock divided by two. For example for DDR2-667 DRAM devices the reference clock frequency would be 166.67 MHz. The reference clocks to the host and each AMB do not have any fixed phase relationship to each other thus simplifying PCB routing since no length matching is required. It is required that all the reference clocks for a given FBD channel originate from a single clock source, e.g. a common clock synthesizer or clock oscillator, thereby ensuring that there is no frequency mismatch or frequency drift between FBD agents. 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 requirements of the reference clock are defined in the *FBD Signaling Specification*.

Figure 2-3 shows a generic example of a clock distribution for a simple single channel FBD platform. The reference clocks for the FBD memory subsystem are generated by the same low-cost CMOS clock components used with non-FBD systems.

Figure 2-3 — Clock Distribution Block Diagram



2.3.2 Receiver Data Sampling

In order to capture the channel data, an FBD agent needs to generate receiver sampling clocks that have a known phase relationship with the received data. FBD allows significant variation in the length of each data bit lane to simplify the channel routing and minimize the length of the longest bit lane. Because of this bit lane skew each receiver needs to 'derive' its own sampling clock from the data stream. A synchronous reference clock is provided that may be used to generate these receiver sampling clocks. A future version of the FBD channel may use forwarded clocks to derive the receiver sampling clocks.

To keep the sampling clock at the receiver in sync with the received data stream without adding the latency of traditional 8b/10b encoding, the FBD physical layer will use dynamic training to track the data. This dynamic training requires a minimum edge density on the bit lanes. To guarantee this edge density in the face of worst-case data patterns the host must periodically send training frames onto the southbound channel. The south most DIMM will correspondingly generate training frames onto the northbound channel to guarantee the required edge density. A density of 6 transitions within 512 transfers or unit intervals (UI) on the channel is required for clock training.

2.3.3 Voltage Temperature Compensation FIFO

The data received on each bit lane is deskewed and passed into the core clock domain of the AMB. The core clock domain generates the clock reference to the DRAM devices and this clock is not adjusted following channel initialization. The timing of data received on each bit lane may drift relative to the core clock domain and a voltage temperature compensation FIFO (VT FIFO) is required to compensate for this drift. The FIFO is pre loaded with a few data bits during channel initialization. The number of buffered data bits in the VT FIFO varies during channel operation to compensate for temperature and voltage changes.

2.3.4 Daisy-Chain Retiming and Data Merge

For a low-latency memory channel it is critical that each data bit is received by the AMB and re-driven to the next AMB or host with minimum delay. The receiver and driver in this critical path are located inside the AMB IO Cell to minimize the delay through the AMB. Every AMB on the channel receives the data from the host even if the data is intended for an upstream AMB. The quality of the signal received by the AMB IO Cell has been degraded by channel distortion and must be re-clocked before it is sent to the next DIMM. The signal is sampled by the receive latch within a very narrow setup/hold timing window and immediately driven out with a clean open data eye. No bit lane de-skew operation is performed during each daisy-chain re-drive. Bit lane skew is allowed to accumulate as the data is sent from AMB to AMB. Latency through the AMB is significantly reduced by not performing a bit lane de-skew operation in each hop. Each AMB or the host receives the data from all of the bit lanes and performs a bit lane de-skew operation before using the data.

The northbound channel has the unique requirement that data from the intermediate DIMMs must be merged into the northbound data stream in such a way as to not introduce any bit-to-bit jitter between the data supplied from the different DIMM. This is done by a well timed multiplexer in the receiver-to-driver signal path and a common transmitter flip-flop. When the AMB logic determines that it is to supply read data on the link, it aligns the data with the northbound frame and signals the AMB IO Cell to multiplex the data into the data stream. The high-speed circuits to switch the multiplexer between the received data comparator and the data serializer are located in the AMB IO Cell. Data is always merged on frame boundaries.

2.3.5 DRAM Clock Generation

The DRAM data interface clock on each DIMM is generated internal to the AMB as a 2x multiple of the reference clock.

2.3.6 DRAM Data Return

The DRAM devices are programmed to return a burst of data on the DRAM data lines a specific number of clocks following a Read Command. The AMB receives the data along with data strobe signals from each DRAM device and captures the data into a data receive latch. The AMB drives the read data onto the northbound channel. The read data propagates through the northbound daisy chain to reach the host. From the host's point of view the sequence of commands it sends to the DRAM and the time slot (frame) that the data arrives back at the host is deterministic. The host performs a training sequence during initialization to determine the number of DRAM base clocks in the round trip. The host may internally delay the received read data to align the data to the internal host core clock.

Intermediate DIMMs must time driving their read data onto the northbound channel to align with the frame timing of the data from the farthest DIMM. Each AMB contains a read data FIFO that is adjusted by logic in the AMB to delay the local read data the appropriate amount. The exact implementation of this logic is design specific. The added delay through the FIFO on the southernmost (last) DIMM is normally set to zero by the host. The delay through the FIFO on the last DIMM may be set to a non-zero value to accommodate a slower FBD DIMM in between the last DIMM and the host.

The channel may be initialized so that the host will recieve the DRAM read data returned from any of the DIMMs on a channel arrive back at the host after the same number of DRAM base clocks. A Variable Read Latency capability is also supported which allows the DRAM data from DIMMs closer to the host to be returned earlier than DRAM data returned from the last DIMM. The host will see the round trip flight time from command to data change slightly as the dynamic tracking mechanism adjusts for thermal and voltage drift. A drift compensation FIFO may eliminate this.

2.4 Host Memory Interface

The Host Memory Interface (HMI) is a hypothetical subsystem used as a framework to define the host functions. The FBD Architecture Specification does not require an actual FBD memory interface to be implemented as described in this chapter.

The HMI is the master agent on the FBD channel and is responsible for initiating and tracking all FBD channel commands. The HMI interfaces the FBD channel to the Host Logic.

The HMI may implement a memory mapped IO mechanism or implement the conventional CFC/CF8 I/O space-based mechanism (or both) for accessing the AMB configuration registers. The HMI generates Channel commands on the FBD channel to access registers in the AMB devices.

The HMI may implement a write queue structure to track write transactions until status information from the AMBs indicate the write operations completed successful. The main purpose of the HMI write buffer is to hold onto the write data for multiple outstanding write operations in case of an error and the need to retry the write operations.

The HMI may also contain error checking and error handling functionality for inbound frames.

The HMI is encouraged but not required to provide the following functions:

It is recommended that the HMI perform error handling of read and write commands, including retrying read and write requests, and checking for errors in read return data.

It is recommended that the HMI implement the Fast Reset mechanism to affect a purge of HMI and FBD channel state and then attempt to recover from intermittent errors.

2.5 Advanced Memory Buffer (AMB)

The Advanced Memory Buffer (AMB) 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. A complete detail description of the AMB is contained in the *FBD AMB Specification*. The goal of this chapter is to provide a basic overview of the AMB interface.

The AMB is a memory interface that connects to an array of DRAM devices to the FBD channel. The AMB is a slave device on the channel responding to channel commands directed to this AMB and forwarding channel command to other AMB devices on the channel.

All memory control for the DRAM resides in the host, including memory request initiation, timing, refresh, scrubbing, sparing, configuration access, and power management.

The AMB is expected to perform the following functions:

Support channel initialization procedures as defined in the Initialization Chapter to align the clocks and the frame boundaries, and verify channel connectivity,

Support the forwarding of southbound and northbound frames, servicing requests directed to this specific AMB or DIMM, as defined in the Protocol Chapter, and merging the return data into the northbound frames,

If the AMB is the last, southern-most, DIMM, perform the responsibilities of the last AMB on the channel,

Detect errors on the channel and report them to the host memory controller,

Support the FBD configuration register set as defined in the *FBD AMB Specification* Register Chapters,

Act as a DRAM memory buffer for all read, write, and configuration accesses addressed to this DIMM,

Provide a write buffer FIFO,

Support an SMBus protocol interface for access to the AMB configuration registers, and

Provide an interface for a thermal sensor as defined in the *FBD AMB Specification*.

2.6 Options

The FBD specification defines a small number of options to support the requirements of different applications. The capabilities of the AMB are communicated to the host during the initialization process in the TS2 training pattern and in bits readable in the Features register in the AMB.

2.6.1 Number of FBD Lanes

Southbound lanes:

It is recommended that a Host Controller implement 10 bit lanes with fail-over to 9 lanes. A host controller may choose to not implement fail-over, or to only implement 9 bit lanes, as is appropriate to the application.

The AMB must support 10 southbound bit lanes with the capability to fail-over to 9 lanes.

Northbound lanes:

There are 5 modes of operation for the northbound lanes as defined in Table 2-6:

Table 2-6 — Northbound Bit Lane Options

Mode	Memory ECC	Channel CRC	Fail Over Capable
14-lane mode	8 bit ECC	12 bit CRC	Yes
14-lane failover	8 bit ECC	6 bit CRC	No (failed over)
13-lane mode	8 bit ECC	6 bit CRC	Yes
13-lane failover	8 bit ECC	No CRC	No (failed over)
12-lane mode	No ECC	6 bit CRC	No

Note: 14-lane fail over and 13-lane mode operate with the same frame format. 13-lane fail over does NOT use the same frame format as 12-lane mode. 13-lane fail over maintains memory ECC without channel CRC protection, while 12-lane mode is designed for non-ECC DIMMs.

The Host controller may implement any combination of northbound modes, depending on the RAS requirements of the application.

An AMB may support all 4 ECC northbound modes: 14-lane mode, 14-lane failover mode, 13-lanes mode, and 13-lane failover mode as a group, or support just the 12-lane mode, or support all 5 modes.

2.6.2 ECC and non-ECC DIMMs

It is desirable to minimize the number of DIMM types in the market to just two kinds, ECC and non-ECC. This reduces the flexibility that can be allowed for AMBs in order to guarantee operation of DIMMs of the proper type in all systems. The DIMM will operate in ECC mode if any of the 4 ECC northbound modes are selected during initialization. The DIMM will operate in non-ECC mode if the 12-lane mode is selected during initialization.

ECC DIMMs:

An AMB that supports ECC must support all 4 ECC northbound modes: 14-lane mode, 14-lane failover mode, 13-lanes mode, and 13-lane failover mode. This assures that the DIMM will work with all ECC host controllers, and avoid confusion in the marketplace with different levels of RAS within ECC DIMMs. In ECC mode the AMB will interpret a Command+Wdata frame as containing a 72-bit data payload.

Non-ECC DIMMs:

An AMB that supports non-ECC must support 12-lane mode and the Data Mask write operations to memory. This provides a 6-bit channel CRC on northbound frames, but no fail-over capability. In non-ECC mode the AMB will interpret a Command+Wdata frame as containing 8 Data Mask bits and a 64-bit data payload.

2.6.3 Variable Read Latency

An AMB may support the Variable Read Latency feature that allows the data from DIMMs closer to the host to provide read data sooner than DIMM farther away from the host. The Variable Read Latency feature is implemented with the Command_to_Data_Decr register in the AMB. A bit in the Features register in the AMB indicates if this feature is supported. Refer to the Initialization Chapter for details. It is possible for Variable Read Latency to be used with a mixture AMBs with and without support, especially if the closer DIMMs support it.

2.6.4 L0s State

An AMB may support the L0s low latency power saving state. A bit in the Features register in the AMB indicates if this feature is supported. If any DIMM indicates that it does not support L0s, the host controller should not attempt to put the channel into L0s. An AMB which indicates that it supports L0s must implement the L0s register and be able to respond to the L0s command and ignore inputs on lanes since they may be powered down.

2.6.5 Protocol Variants

It is desireable to provide a mechanism to support the addition of fututre protocol variants to the FB-DIMM specification. This section defines the mechanisms in the various areas of the FB-DIMM Architecture & Protocol Specification that accommodate future protocol variants. Details of the protocol variant must be captured in a protocol variant specific addendum.

2.6.5.1 SPD content

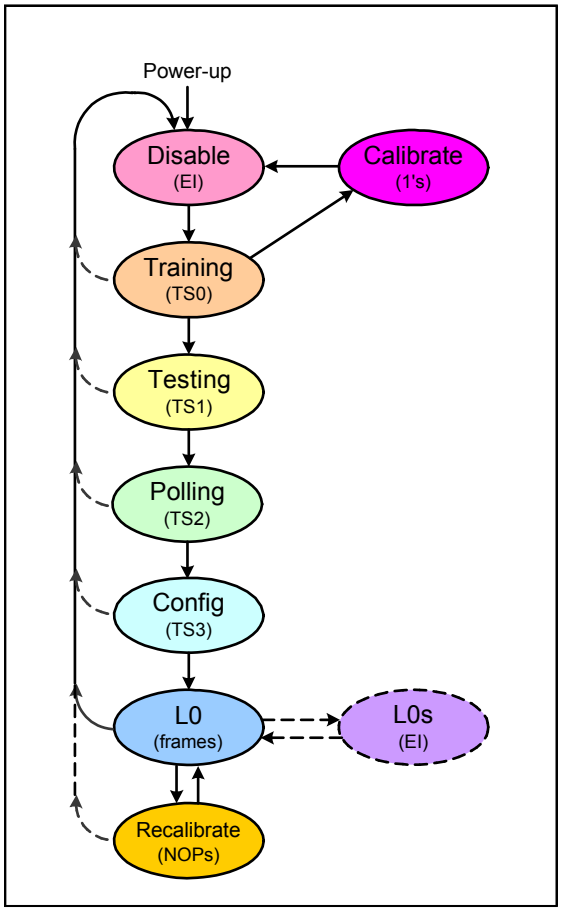
The SPD EEPROM contains two bytes that identify to the boot firmware what protocols are supported by the DIMM. A bit map is present in the "Supported Protocols" bytes to identify which protocols are supported. More than one bit may be set if the DIMM supports more than one protocol.

- Bit 0 = 1 indicates that the DDR2 Base non_ECC Protocol is supported (10 SB & 12 NB bit lanes).
- Bit 1 = 1 indicates that the DDR2 Base ECC Protocol is supported (10 SB, 13 NB & 14 NB bit lanes).
- Bit 2 = 1 indicates that the DDR3 Base non_ECC Protocol is supported (10 SB & 12 NB bit lanes).
- Bit 3 = 1 indicates that the DDR3 Base ECC Protocol is supported (10 SB, 13 NB & 14 NB bit lanes).
- Bits 4-15 are reserved for future protocol variants.

3 Channel Initialization

This chapter defines the process of initializing the FBD channel. The FBD initialization process generally follows the top to bottom sequence of state transitions shown in the high level AMB Initialization Flow diagram in Figure 3-4. The host must sequence the AMB devices through the Disable, Calibrate, (back to Disable), Training, Testing, and Polling states in order to transition the AMBs into the active channel L0 state. The value in parenthesis in each state bubble indicates the condition/activity of the links during these states.

Figure 3-4 — AMB Initialization Flow Diagram



The states in the AMB Initialization Flow diagram are:

Disable – The channel is inactive and the interface signals are in a low power Electrical Idle condition.

Training – The initial bit alignment and frame alignment training is done in this state.

Testing – Each bit lane is individually tested in this state.

Polling – The channel capabilities of the individual AMB devices are communicated in this state.

Config – The channel width configuration is communicated to the AMB devices in this state.

L0 – The channel is active and frames of information are flowing between the host and the AMB devices.

Recalibrate – The channel is momentarily idled to allow TX and Rx circuits to be recalibrated.

L0s – The channel is in a low-latency power saving condition. (Optional)

Each bit lane is initialized (mostly) independently to support fault tolerance. The transitions in the figure represent the transitions of the AMB core logic state machine and are taken when the transition event is detected on the minimum required number of southbound bit lanes. The chain of FBD links connecting the host to the AMBs must each be initialized to establish the timing for broadcasting data frames in the southbound direction and for merging data frames in the northbound direction. The AMBs on the channel are generally initialized as a group but because each AMB is individually addressable many alternate initialization sequences may be employed.

In this chapter, southbound may be abbreviated as SB and northbound may be abbreviated as NB.

The FBD channel uses the timing parameters defined in Table 3-7 while sequencing through channel initialization.

Table 3-7 — Channel Initialization Timing Parameters

Timing Parameter	Value	Description
tCalibrate	960K frames (~2ms)	Calibration Interval: This value indicates the minimum amount of time the host will provide for the channel Rx and Tx circuits to perform initial electrical calibration steps following a hardware reset event.
tClkTrain	42 frames	Clock Training Interval: This value defines the period of time in which a minimum number of transitions must occur on each bit lane in order to ensure the clock tracker circuits remain locked on the data stream. (Based on 6 transitions within 512 UI)
tBitLock	119 frames	Bit Lock Interval: This value defines the maximum time an AMB may take to acquire bit lock on incoming TS0 patterns on a bit lane by bit lane basis and pass the TS0 pattern to the Tx outputs. This parameter is not used as a timeout but is provided as a requirement to limit the time it takes for a Fast Reset sequence to complete.
tFrameLock	154 frames	Frame Lock Interval: This value defines the maximum time an AMB may take to acquire frame lock on incoming TS0 patterns and be prepared to receive a TS1 pattern. This parameter is not used as a timeout but is provided as a requirement to ensure that intermediate AMBs acquire frame lock before the host starts sending TS1 patterns.
tEIPropagate	60nS 16 frames at 533 20 frames at 667 24 frames at 800	Channel Electrical Idle Propagate: This value defines the maximum time an AMB may take to detect an Electrical Idle on the Rx inputs and propagate the Electrical Idle to the Tx outputs.
tDisable	DDR: 533 / 667 / 800: 512M: 492 / 603 / 714 1Gbit: 504 / 618 / 732 2Gbit: 540 / 663 / 786 4Gbit: 511 / 751 / 892 in Frames.	Channel Disable Interval: This value defines the minimum amount of time the host will provide for the Electrical Idle condition to propagate to all AMBs on the channel to put them into the Disable state. It is calculated as 12*tEIPropagate*2+Self Refresh Entry.

The maximum skew between lanes of the high speed interface is specified in the AMB specification.

3.1 Reset and Inband Control "Signals"

This section describes special mechanisms used to control the AMB state while the host controller sequences the FBD channel through the initialization sequence.

3.1.1 RESET# Signal

The RESET# signal acts as a hardware reset and immediately puts the AMB into a known state. The AMB Initialization FSM is put into the Disable state and the NB Tx outputs are put into Electrical Idle regardless of the state of the NB Rx inputs. All "sticky" bits are set to their default values. The CKE signals to the DRAM devices are driven inactive to turn off the DRAM output drivers. DRAM specific mechanisms in the AMB may generate additional signal transitions to the DRAM devices to make sure that they do not hang in an unknown state. The AMB specification for each DRAM technology defines any DRAM specific mechanisms. If the DRAMs were in self refresh prior to RESET# being asserted, they will remain in self refresh through the hardware reset. The host must wait until the power and the reference clock to the AMBs have been stable for greater than or equal to 1ms before transitioning the channel out of the Disable state. The relationship between supply voltage, reference clock and the RESET# signal is defined in the *FBD AMB Specification*.

3.1.2 Inband Control "Signals"

There are no dedicated control signals implemented on an FBD channel. Two different channel characteristics are exploited to deliver inband control information on the FBD channel wires when no clock timing has been established between the host and the AMBs:

Electrical Idle (EI): During normal channel operation the Tx outputs are enabled and a differential voltage is present on each bit lane. In Electrical Idle the Tx outputs source insignificant current and the termination resistors at the receiver pull both signals of the differential pair to ground. The Rx inputs can detect if both differential inputs are near ground to receive inband control information. The specification of the voltage levels used for detection is defined in the *FBD Signaling Specification*.

Clock Training Violation: During normal channel operation the southbound bit lanes contain a minimum number of transitions every tClkTrain frames to keep the clock tracking circuits on each bit lane locked to the data stream. It is the absence of these periodic bit lane transitions that is used by the host to communicate control information. The AMB detects a clock training violation by the lack of sync frames sent from the host. See the AMB specification for details.

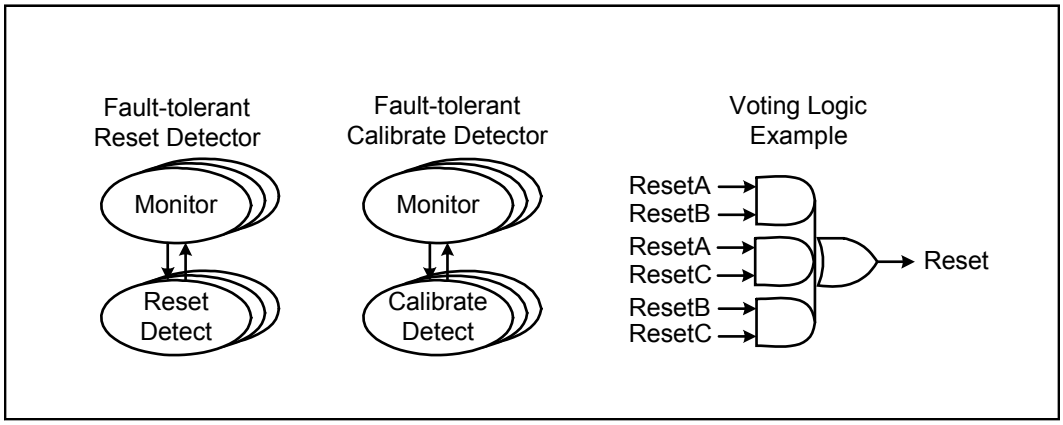
3.1.3 Inband Reset Event Detector

The host communicates a *reset event* to the AMBs by putting its SB Tx outputs into an Electrical Idle condition. These reset events do not put the AMB into a hardware reset condition, and thus allow the retention of sticky state information.

For high reliability, it is assumed that any of the channel bit lanes could be faulty and the FBD link must still function properly in the presence of a single bit lane failure (actually one in each direction). The reset event must be detectible even if any one of the bit lanes is in Electrical Idle, stuck at zero, stuck at one, or is transitioning with an erroneous pattern. To guarantee that any single bit lane failure will not prevent the reset events from being delivered, a fault-tolerant reset detection mechanism is required. The required mechanism uses three Reset Event Detectors to monitors the 3 least significant SB bit lanes. The outputs of the three reset detectors are routed to each state machine in the AMB and each state machine uses voting logic similar to the example in Figure 3-5 to decide if a reset event is present. Electrical Idle present on at least 2 of 3 bit lanes is detected as a reset event. This redundant delivery of reset guarantees that the reset event is delivered even if any of the three bit lanes is bad or if any one of the Reset Event Detectors is faulty.

The detection of a reset event is different than a hardware reset, and does not put all internal AMB registers into a known state. In all cases except the L0s state the detection of a reset event will cause the AMB internal state machines to complete any current activities, perform clean up operations, and make an orderly transition to the Disable state. In the L0s state the channel is disabled to save power and the reset event detectors are ignored. The AMB is in the L0s state for less than or equal to tClkTrain and transitions back to the L0 state where the reset event is then detected.

Figure 3-5 — In-band Control Signal Detectors



3.1.4 Inband Calibrate Event Detector

The host communicates a *calibrate event* to the AMBs by sending a continuous stream of 1's on all southbound bit lanes for a time greater than or equal to tCalibrate. A clock training violation will occur and trigger the event in the AMBs.

Following any hardware reset, the transmitters and receivers must be calibrated to adjust them for optimum performance. Before this calibration is performed only gross logic one and logic zero states can be communicated to the AMBs. Similar to the Reset Event Detectors, the Calibrate Event Detectors on each AMB monitor the 3 least significant SB bit lanes and a continuous stream of 1's for a time greater than or equal to two times tClkTrain on at least 2 of 3 bit lanes is detected as a calibration event. The Calibrate Event Detectors are used to transition the AMBs from the Training state to the Calibrate state.

3.2 Training Sequence Ordered-sets

Before the channel is initialized the bit lanes are not aligned to a frame boundary, thus preventing Channel and DRAM commands from being communicated to the AMBs. Instead of channel commands, training sequences are sent on each bit lane to communicate with the AMBs in a serial fashion. Training sequences are composed of ordered sets used for initializing bit alignment, frame alignment, bit lane testing, and to exchange Physical Layer initialization/configuration parameters. Training sequences are sent serially on all bit lanes in parallel. The host must send the training sequences bit aligned on all SB bit lanes. The host must repetitively send a stream of training sequences to the AMBs during each training state. The host generally receives a corresponding stream of training sequence from the AMBs to indicate that the SB training sequence has been correctly received. The reception of the training sequence from the AMBs provides reliable notification to the host that the requested training has completed and that the host may transition the channel to the next state. The host transitions the channel to the next state by changing the type of training sequence it is repetitively sending on the channel. The host must transition from one type of training sequence to another without any dead time between the sequences.

Training sequences are defined using 12-bit groups. Training sequence patterns are sent serially starting from the LSB (bit 0) to MSB (bit 11) within each group and in sequential group order from group 0 to n.

The training sequences and any data values in the training sequences must be detected by the AMBs on 2 or more of the 3 least significant SB bit lanes before any state transitions are taken by the AMBs. This insures that a bad bit lane or a transient bit failure does not cause the AMB to falsely respond to the sequence. Voting logic similar to that shown in Figure 3-5 may be used to filter out bit lane or transient bit failure.

After initialization, the groups of bits received on multiple SB bit lanes are combined into the frames used to communicate Channel and DRAM commands to the AMBs.

There are 4 defined training sequences:

TS0 – Used in the Training state for initial link training.

TS1 – Used in the Testing state to perform an electrical stress test of the individual channel bit lanes.

TS2 – Used during the Polling state to determine the round trip latency of the channel and to test that each individual AMB can perform data merge properly.

TS3 –Used in the Config state to communicate the channel configuration to the AMBs.

3.3 Channel Initialization Sequence

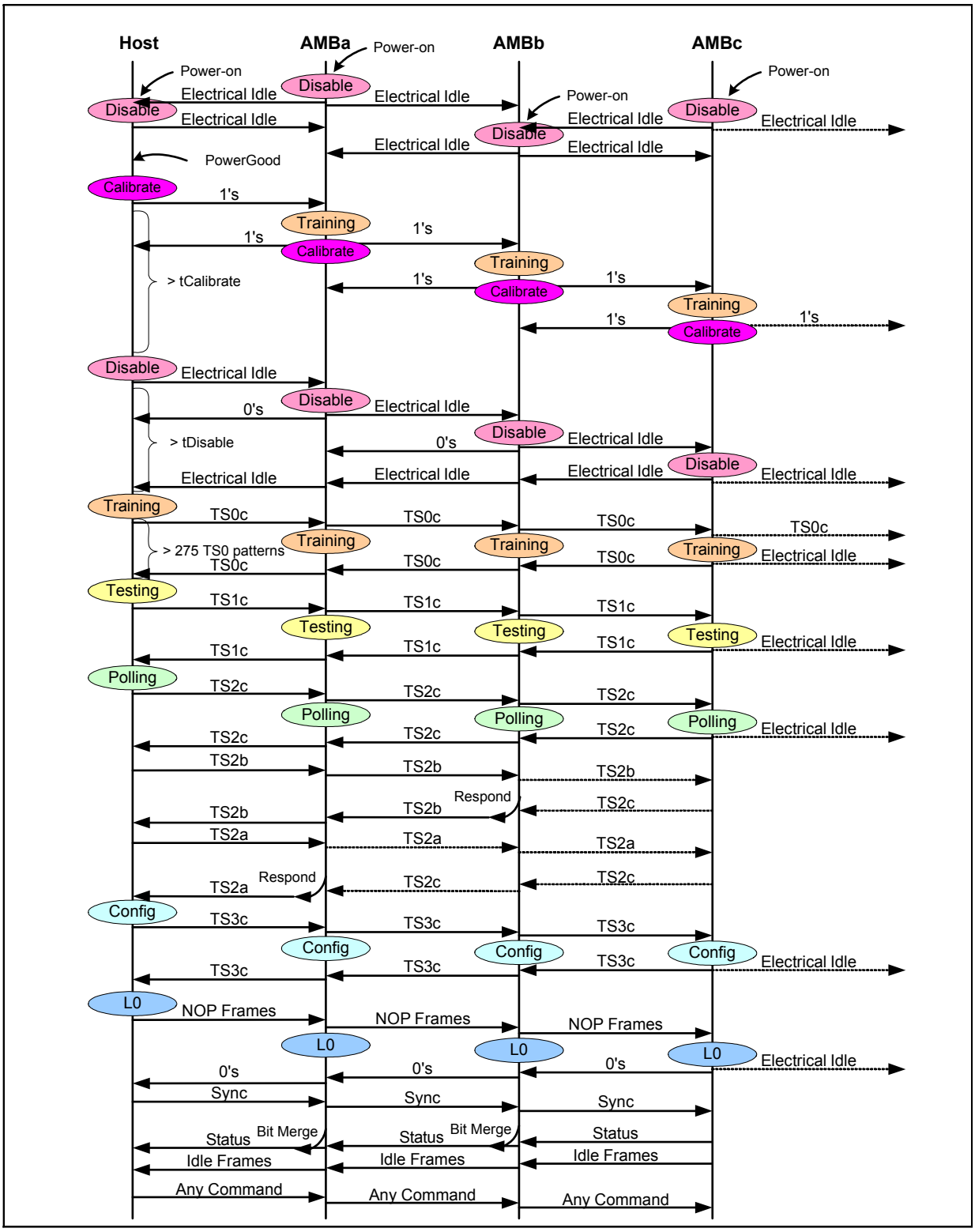
The host controller sequences the FBD channel through the initialization sequence. The AMB devices on each DIMM monitor in-band signals from the host and use events and patterns on these signals to transition from one state to another. If the channel fails to initialize properly the host may transition the channel back to the Disable state and try again a number of times before reporting a failure to the system.

It is undesirable to continuously drive high frequency signals into un-terminated transmission lines because of the EMI that is generated and the power that is wasted. To avoid this the host must return to the Disable state if the channel does not properly initialize.

Figure 3-6 shows the graphical representation of the “full” initialization sequence for a 3 AMB configuration. This representation shows the data traffic on each link. The placement down the page indicates the timing of the initialization sequence of events. The bubbles on the various timelines represent state transitions.

Note that the AMBs may be powered up in any sequence and will remain in the disabled state until instructed by the host to transition to the next state.

Figure 3-6 — Initialization Sequence Flow

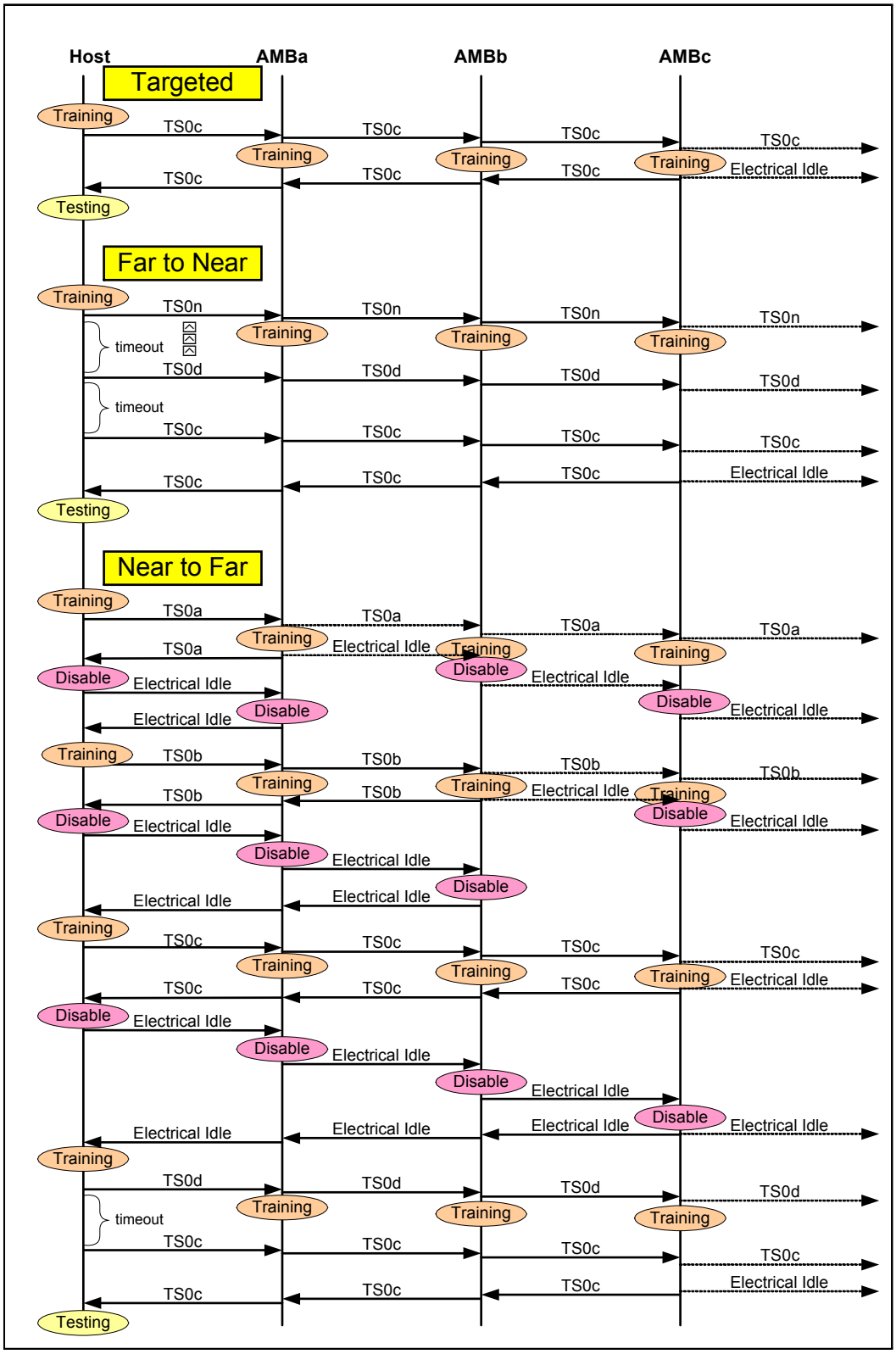


The above example illustrates a ‘targeted” AMB discovery method that uses system information such as access to the SPD EPROM on the DIMMs to determine the last DIMM position. Figure 3-7 illustrates the above “targeted” AMB discovery method and two alternative discovery methods; “near to far”, and “far to near”. Either of these alternative discovery methods may be used based on system requirements.

The “far to near” method starts at the farthest DIMM location supported by the platform and searches toward the host until an AMB is found. If no AMB is found the host must timeout after 1ms and try the next closer DIMM.

The “near to far” method adds an AMB at a time starting closest to the host. Each time the host changes from one AMB to the next the northbound clock and alignment must be reestablished. The channel must transition back to the Disable state to initialize the clock trackers so they will reestablish lock with the new northbound timing. After the host walks past the last DIMM it backs up and establishes the northbound timing using the last AMB found.

Figure 3-7 — Alternative AMB Discovery Sequence Flows



3.3.1 Firmware Transition Control

The channel initialization and configuration sequence may be controlled by a hardware state machine or directed by firmware. To provide a flexible mechanism for dealing with a variety of FBD channel failure conditions it is recommended that the channel initialization and configuration process be controlled by firmware.

It is recommended that implementation specific control registers be included in the host to allow firmware to step through the initialization steps and perform the following functions:

- Put the SB Tx outputs into Electrical Idle.
- Drive SB Tx outputs to all ones.
- Detect if the NB port is receiving Electrical Idle.
- Drive TS0 patterns with an arbitrary AMB_ID value.
- Receive TS0 patterns and read the returned AMB_ID value
- Drive TS1 patterns with an arbitrary AMB_ID value and with a sequence of electrical stress test patterns on each bit lane. Registers to hold an arbitrary 24-bits of Test Parameter values are recommended.
- Receive TS1 electrical stress test patterns and check the patterns.
- Test the NB bit lanes and report NB test results.
- Drive TS2 patterns.
- Receive TS2 patterns and determine the round trip channel delay.
- Drive TS3 patterns with channel configuration values.
- Receive TS3 patterns and check the returned values.
- Set the Last_AMB_ID value.
- Set the Hot_Add_AMB_ID value
- Set the Fast_Reset_Flag value.
- Set the Recalibrate_Duration value.
- Set the L0s_Duration value.
- Transition the channel to the L0 state and send the first Sync command.

It is expected but not required that the host will implement a Hot-Add initialization sequence in firmware after a DIMM has been added to the system. The *FBD AMB Specification* Register chapter defines the AMB configuration registers that perform the Hot-Add Initialization functions. This register model may be duplicated in the host for similar control of the entire channel.

3.3.2 AMB Internal State Variables

A number of internal flags and timers are referenced in the following sections. These flags and timers are implementation specific and included in the state tables to describe internal AMB state that may or may not be visible in defined AMB registers. These flags and timers include:

- Last_AMB_Flag – set in the last AMB to enable unique properties of the AMB in this position.
- First_Sync_Received_Flag – set to disable further initialization of the Idle/Alert Frame LFSR.

- Idle/Alert Frame LFSR – a counter in each AMB used to generate Idle and Alert frames on the NB channel.
- Alert_Flag – a flag that indicated that this AMB detected an error and is or was generating NB Alert frames.
- Recalibrate_Timer – a timer that keeps track of how long the AMB has been in the Recalibrate state.
- L0s_Timer – a timer that keeps track of how long the AMB has been in the L0s state.

3.3.3 Disable State

The channel is forced into the Disable state during hardware reset. The host may put the channel into the Disable state at any time and from any other state other than L0s by putting the three least significant Tx outputs into Electrical Idle. The host must not put the channel into the Disable state from the L0 state until any DRAM write operations have had time to complete. Channel initialization always starts in the Disable state.

State Table Definitions

- "Disable Rx inputs" means that the input receiver termination is still enabled, the high-speed receiver circuitry may be turned off to reduce power consumption, and the receivers are not required to provide any data. The Electrical Idle detectors may be on or off as specified in the state tables.
- "Disable Tx outputs" means that the current/voltage sources are turned off. The Tx termination remains on to inhibit noise on the high speed lanes for robust detection of electrical idle. Other transmitter circuitry may be turned off to reduce power consumption.

Table 3-8 defines the actions of the AMBs during the Disable state.

Table 3-8 — Disable State

	AMB
Entry Condition	Hardware reset *or* Electrical Idle detected on 2of3 LSB SB Rx inputs.
Action	Upon Hardware Reset the following actions are performed: — Terminate any commands in progress including Self Refresh entry sequence. If the DRAMs were in Self Refresh prior to the hardware reset, the Self Refresh mode will be maintained. If they were not in Self Refresh prior to the hardware reset, data may be lost. — Reset all memory elements to default values, including “sticky” bits. — Reset all logic to the default state. — Disable SB Rx inputs except Electrical Idle detectors on 3 LSB inputs — Disable SB Tx outputs — Disable NB Rx inputs including Electrical Idle detectors — Disable NB Tx outputs Upon entry from other than hardware reset — If transitioning from the Testing, Polling, Config, L0, L0s, or Recalibrate states, put the DRAMs into Self Refresh — Disable SB Rx inputs except Electrical Idle detectors on 3 LSB inputs — Disable SB Tx outputs — If Electrical Idle not yet detected on 2of3 LSB NB Rx inputs or the DRAM Self Refresh entry has not yet completed • Continue to put DRAMs into Self Refresh • Disable NB Rx inputs except Electrical Idle detectors on 3 LSB inputs • Leave NB Tx outputs enabled and drive logic 0’s — Else (EI detected on 2of3 LSB NB Rx inputs and DRAM Self Refresh entry completed) • Disable NB Rx inputs including Electrical Idle detectors • Disable NB Tx outputs • Reset “non-sticky” memory elements to their default values. Memory elements not involved in self refresh entry or EI detection could be reset earlier in this process if desired. • Reset all logic to the default state
Exit Condition & Next States	Transition to the Training state if hardware reset condition removed and a non-Electrical Idle condition detected on 2of3 LSB SB Rx inputs, *else* wait.

At the transition into the Disable state from any active state the “sticky” status bits held in status registers are frozen to reflect the state of the interface before the transition. Channel commands are required to change the individual sticky bits. Refer to the *FBD AMB Specification* Register chapter for the definition of the sticky bits in the AMB status registers.

When transitioning into the Disable state from another state other than because of a hardware reset, the AMBs will only disable NB Tx outputs if Electrical Idle is detected on 2of3 LSB NB Rx inputs. In the Disable state the last AMB detects that no AMB is south of its position and disable its NB Tx outputs. The next closer AMB to the host will detect this condition and disable its NB Tx outputs. Each intermediate AMB will continue this process until the AMB closest to the host disables its NB Tx outputs. At this point the host can be assured that all of the AMBs are in the Disable state and that all of the DRAM devices have been put into Self Refresh. In high-availability applications, the host must implement a timeout to attempt channel initialization even if the Electrical Idle condition is not detected from the channel to prevent a faulty AMB in the chain from allowing a subset of the memory subsystem from being initialized.

The channel may stay in the Disable state indefinitely while the host is held in reset by external conditions. The host must wait for the NB Rx inputs to detect that the channel is in Electrical Idle indicating that the AMBs are all in the Disable State plus a minimum of 100 frames before taking the channel out of the Disable state. In high-availability

applications, the host may proceed after a timeout period. The length of the host timeout is platform specific but must be larger than the maximum time it takes for the AMBs to transition to the Disable state. If the host Fast_Reset_Flag is reset the host must transition the channel to the Calibrate state to perform the calibration of the channel transmitters and receivers. The host drives logic ones for a period equal to or greater than tCalibrate to signal to the AMBs that they should transition to the Calibrate state and to give them time to complete the calibration operation (Full Reset). The AMBs will momentarily transition from the Disable state into the Training state until the Calibrate Event Detectors recognize the calibrate event and then transition into the Calibrate state.

3.3.4 Training State

The host drives a repetitive series of TS0 patterns to transition the AMBs from the Disable state to the Training state and to perform initial link training. The host may detect that the last AMB has acquire frame lock when TS0 patterns are received on the required number of NB Rx inputs. Table 3-9 defines the TS0 format. Bit patterns in TS0 are used to perform bit lock and frame lock. The pattern is mostly filled with an alternating 1010 pattern to align the clock trackers with the incoming data stream. The sequence generally has logic zeroes in the even bit positions and logic ones in the odd bit positions. The beginning of the sequence is identified by the header pattern shown in the table below and is used to establish the alignment of the serial data onto frame boundaries.

The AMB_ID field identifies which AMB should initiate the northbound data stream and send a TS0 pattern northbound to the host.

Table 3-9 — Training Sequence – TS0 (Training)

Group #	Description	Value [11:0]

0	TS0 Header:
	Continuous string of 9 ones identifies the start of the TS0 sequence.
		`Bit [11:0]: 12b'1011 1111 1110`
		`  *  *`
		`MSB  LSB`

1	Control: Bit [4:0] – training pattern Bit [11,9,7,5] – AMB_ID — Host indicates which AMB should return a TS0 pattern. — AMB returns this value. Bit [10,8,6] – training pattern	 Bit [4:0]: 5b'0 1010 Bit [5]: AMB_ID value bit 0 Bit [6]: 1b'0 Bit [7]: AMB_ID value bit 1 Bit [8]: 1b'0 Bit [9]: AMB_ID value bit 2 Bit [10]: 1b'0 Bit [11]: AMB_ID value bit 3
11-2	10 groups of clock training patterns	Bit [11:0]: 12b'1010 1010 1010

Table 3-10 estimates the amount of time a fully populated channel may take to complete initial training. Based on this estimate and providing a little margin, the host must send a minimum of 275 TS0 patterns before switching to TS1 patterns to give all intermediate AMBs time to acquire frame lock.

Table 3-10 — Training Duration Estimate at DDR2-667

State	Duration per AMB	Duration for 12 AMBs on the channel (Frames)	Description
EI deassertion detection	10 frames	120	Time to detect data being driven on the SB Rx inputs.
Bit lock	119 frames	1428	Time to lock on the SB input bit stream and start to forward to the next AMB.

Table 3-10 — Training Duration Estimate at DDR2-667 (Cont'd)

State	Duration per AMB	Duration for 12 AMBs on the channel (Frames)	Description
Last AMB frame lock	13 TS0 patterns	154	Time for the last AMB to deskew its bit lanes, align its core clock, and start sending TS0 patterns NB
EI deassertion detection	10 frames	120	Time to detect data being driven on the NB Rx inputs.
Bit lock	119 frames	1428	Time to lock on the NB input bit stream and start to forward to the next AMB.
Host frame lock	13 TS0 patterns	154	Time for host to align to deskew its bit lanes and align its frame boundaries
Total frames		3404	Approximate number of frames for training.
Total TS0 patterns		284	Approximate number of TS0 patterns needed for training.

Table 3-11 defines the actions of the AMBs while in the Training state. Each AMB determines its AMB_ID number using the SMBus address strap pins. Refer to section 2.1.3 for the SMBus address to AMB_ID mapping.

Table 3-11 — Training State

	AMB
Entry Condition	Entry from Disable state if reset conditions are removed and exit from Electrical Idle detected on 2of3 LSB SB Rx inputs
Action	The following actions are performed: — Enable SB Rx inputs including Electrical Idle detectors • If SB bit lock (on a bit lane by bit lane basis) – Enable SB Tx output – Forward SB data patterns • Else – Leave SB Tx output disabled — Establish core clock alignment and drive a stable clock to the DRAM devices — If the TS0 header and AMB_ID field are received on 2of3 LSB SB Rx inputs, and the AMB_ID value matches the AMB number • Set the Last_AMB_Flag • Disable SB Tx outputs (they were momentarily enabled) • Disable NB Rx inputs including Electrical Idle detectors • Enable NB Tx outputs • Loopback or generate NB TS0 on all supported bits using the Command to Data fractional frame delay — Else • Enable NB Rx inputs including Electrical Idle detectors • If NB bit lock (on a bit lane by bit lane basis) – Enable NB Tx output – Forward NB data patterns (initial NB TS0 pattern may be truncated) • Else, – Leave NB Tx output disabled
Exit Condition & Next States	Transition to Calibrate state if 1's detected on 2of3 LSB SB Rx inputs for greater than 2 times tClkTrain frames, *else*, transition to Testing state if SB TS1 header is received on 2of3 LSB SB Rx inputs, *else*, transition to Disable state if Electrical Idle detected on 2of3 LSB SB Rx inputs, *else* wait.

The host controller must send at least 4 TS0 sequences after the final supported northbound lane is detected to exit electrical idle.

3.3.4.1 Southbound Clock Training

After any reset event the southbound clock tracking state machines are initialized and must reacquire a lock on the received data timing. This is referred to as bit lock.

Southbound clock alignment is performed in the Training state. In this state all of the southbound bit lanes are enabled and the host sends frames of data filled with a pattern rich in bit transitions. During this time the clock tracking state machines on each bit lane lock their clock trackers to the transitions in the pattern. The pattern has a long sequence of alternating ones and zeroes. Proper synchronization is attained when all even data bits in this field are received as zeroes and all odd data bits are received as ones.

The TS0 pattern is not forwarded from the first AMB to the second AMB until the first AMB has locked its clock tracker. The clock tracking state machines in each AMB in turn locks onto the TS0 pattern as its input pattern stabilizes. A chain of up to 12 AMBs is supported to accommodate large memory configurations and configurations that use channel repeaters or have a logic analyzer component in the chain. Each AMB must be able to resolve clock alignment and send a stable waveform to the next southern AMB in a time less than tBitLock.

After a time greater than or equal to tBitLock times 12 the clock trackers on all functional southbound bit lanes should have locked and are tracking slow variations in the channel clock timing.

3.3.4.2 Southbound Bit Lane Deskew and Core Clock Training

Southbound bit lane deskew and core clock alignment are also performed in the Training state. The TS0 pattern has a unique header pattern that is used to identify the beginning of the ordered set. The beginning of the TS0 pattern is used to align the bit lanes even if the skew between bit lanes is greater than several frame times.

The AMB must ensure that the DRAM clock is stable at the DRAM devices before generating or forwarding the TS0 pattern to the host.

3.3.4.3 Northbound Clock Training

After any reset event the northbound clock timing must be reestablished. During northbound clock alignment, the specified AMB on the channel sends a TS0 pattern northbound on all supported bit lanes. The data pattern at the host is not stable until all of the intermediate AMBs have locked their clock trackers. Each AMB must be able to resolve clock alignment and send a stable waveform to the next northern AMB in a time less than tBitLock.

3.3.4.4 Northbound Host Bit Lane Deskew and Alignment to Core Clock

The host performs northbound bit lane deskew and alignment to its core clock while in the Training state.

Northbound frame alignment is performed in the Training state. This frame alignment is not changed after being established in the Training state. Any subsequent return timing changes are always made in full frame increments.

3.3.5 Testing State

The host drives a TS1 pattern to transition the AMBs from the Training state to the Testing state and may send an arbitrary number of TS1 patterns to test the channel.

The DRAM clock is stable at the DRAM devices before entering the Testing state. When entering the Testing state, if the DISSREXIT configuration bit is not set, the AMBs assert the CKE to all DRAMS attached to the AMBs to cause the DRAM devices to exit Self Refresh. Following hardware RESET, the DISSREXIT bit is set, preventing the DRAMs from exiting self refresh. The host controller will reset the bit prior to the fast reset in which it wants the DRAMs to exit self refresh. Commands must not be immediately sent to DDR2 SDRAMs after exiting Self Refresh until two parameters are satisfied:

tXSNR is the minimum time from exiting Self Refresh before a non-Read command is executed and is defined as tXSNR = tRFC+10ns (tRFC = refresh to active/Refresh command time = 75 to 195ns for 256Mb to 2Gb devices, device technology dependent).

tXSRD is the minimum time from exiting Self Refresh before a Read command is executed and is specified as 200 tCK.

The host must not send commands to the DDR2 SDRAM devices until the tXSNR and tXSRD parameters have been satisfied. The host must also not transition the channel back into the Disable state, which causes the AMBs to issue commands to put the DDR2 SDRAMs into Self Refresh, until the tXSNR parameter has been satisfied. The use of Self Refresh mode introduces the possibility that an internally timed refresh event can be missed when CKE is raised for exit from Self Refresh mode. Upon exit from Self Refresh, the DDR2 SDRAM requires a mini-mum of one extra auto refresh command before it is put back into Self Refresh mode.

Table 3-13 defines the format of the TS1 pattern. The TS1 pattern is used to perform an electrical stress test of the channel bit lanes and has a platform specific length.

A host supporting a transient-error-tolerant initialization sequence must at least repeat TS1 patterns SB until it detects NB properly framed TS1 headers followed by control with the proper AMB_ID on any given lane.

A host that supports persistent-fault-tolerant initialization must at least repeat TS1 patterns SB until multiple properly framed TS1 patterns on 2of3 LSBs are returned. Note that a single SB lane fault plus a single NB lane fault may prevent the host from receiving TS1 patterns on 2of3 LSBs on one of the SB to NB mapping selections. In this case

the other SB to NB mapping selection will provide TS1 patterns on 2of3 LSBs. The host could alternately test 2of4 LSBs in a specific mapping. If some lanes consistently do not return the TS1 header, the host can conclude a persistent failure exists on those lanes.

The AMB with the Last_AMB Flag set should respond by looping back the TS1 pattern northbound. The lower or upper SB bit lanes may be driven onto the NB bit lanes. The SBtoNB_Mapping bits in the NB TS1 pattern indicates which SB bit lanes are being looped back in the NB TS1 pattern. The host may use the two mapping modes to test all of the NB bit lanes even if one of the SB bit lanes is broken. The mapping of the southbound lanes onto the northbound lanes is defined in Table 3-12.

Future protocol variants may wish to increase the number of southbound bit lanes to accommodate additional control/ address information. The host must sequence the AMB devices through the Disable, Calibrate (and back to Disable), Training, Testing, Polling, and Config states in order to transition the AMBs into the active channel L0 state.

During the Testing state the additional southbound and northbound bit lane(s) must be tested using the same techniques used to test the other southbound bit lanes. The Base Protocol southbound to northbound bit lane mapping is specified in the SBtoNB_Mapping field in the TS1 training sequence and is defined in Table 3-12 below. The protocol variant specific southbound to northbound bit lane mappings are specified in the individual protocol variant specifications.

Table 3-12 — SB to NB Bit Lane Mapping

Value	NB bit lane	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000	Base lower 5 SB lanes	3	2	1	0	4	3	2	1	0	4	3	2	1	0
001	Base upper 5 SB lanes	8	7	6	5	9	8	7	6	5	9	8	7	6	5
010 to 111	Reserved	Specified in the protocol variant specifications													

The TS1 pattern contains a Test Parameter field to communicate voltage and timing margining parameters to the AMBs. The Test Parameters may be used to change the drive strength of the Tx outputs, change the input threshold voltage of the Rx inputs and vary the timing sample points of the Rx inputs.

The TS1 pattern is terminated with a pair of End Delimiters to identify that the next training sequence header should follow. An End Delimiter is the numeric sequence of “8,7,6,5,4,3” and is a pattern that must not be used in electrical stress testing patterns. Excluding the End Delimiter pattern, arbitrary stress testing bit patterns may appear in the TS1 data payload. Care should be taken to ensure that the tClkTrain parameter in not unintentionally violated.

Table 3-13 — Training Sequence – TS1 (Testing)

Group #	Description	Value [11:0]
0	TS1 Header: Continuous string of 11 ones to identify the TS1.	Bit [11:0]: 12b'1111 1111 1110
1	Control: Bit [3:0] – AMB_ID — Host fills this field with all 1's. — AMBs ignore the value in SB TS1. Last AMB returns its AMB_ID in NB TS1 Bit [6:4] – Base protocol SBtoNB_Mapping — Host specifies SB bit lane to NB bit lane mapping. — AMB returns this value to indicate the source of the NB electrical stress pattern. Bit [11:7] – reserved	Bit [3:0]: AMB_ID value Bit [6:4]: 3 bit field that specifies the southbound to northbound mapping. Bit [11:7]: 5b'0000 0
3-2	Test Parameters: Bit [23:0] – Test parameters — Host indicates test conditions to be used during this training sequence. — AMB returns this value.	Bit [23:0]: 24 bits of AMB implementation specific test parameters. Must be filled with zeroes if AMB type is unknown.
(n-4)-4	Electrical Stress Pattern: — Host sends platform specific patterns to stress the channel signaling as a rudimentary test of the signal integrity of each bit lane. A different pattern may be sent on each bit lane. — AMB returns the pattern it receives on the selected SB bit lanes onto the NB bit lanes.	Bit [(12n-83):0]: Electrical stress pattern. (12n-84) total bits located in (n-7) groups of 12 bits each.
(n-3)	End Delimiter (first delimiter, first group)	Bit [11:0]: 12b'0110 0111 1000, 3h'678
(n-2)	End Delimiter (first delimiter, last group)	Bit [11:0]: 12b'0011 0100 0101, 3h'345
(n-1)	End Delimiter (last delimiter, first group)	Bit [11:0]: 12b'0110 0111 1000, 3h'678
n	End Delimiter (last delimiter, last group)	Bit [11:0]: 12b'0011 0100 0101, 3h'345

Table 3-14 defines the actions of the AMBs while in the Testing state.

Table 3-14 — Testing State

	AMB
Entry Condition	Entry from Training state if SB TS1 header is received on 2of3 LSB SB Rx inputs.
Action	The following actions are performed: — Receive SB TS1 patterns — If DISSREXIT is not set, assert CKE high to all DRAM devices attached to the AMB — If Last_AMB_Flag is set, • Loopback SB TS1 patterns NB mapping the upper or lower SB bit lanes selected by the SBtoNB_Mapping bit onto all supported NB bit lanes using the Command to Data fractional frame delay — Else, • Forward SB data patterns • Forward NB data patterns
Exit Condition & Next States	Transition to Polling state if a SB TS2 header is received on 2of3 LSB SB Rx inputs, *else* put the DRAMs into Self Refresh and transition to Disable state if Electrical Idle detected on 2of3 LSB SB Rx inputs, *else* wait.

3.3.5.1 Bit Lane Testing

The host sends an electrical stress test pattern while in the Testing state. The TS1 sequence is an arbitrary length and terminated with a pair of End Delimiters. The last AMB will return the pattern it receives on either the lower or upper SB bit lanes onto the NB bit lanes. All AMBs expect to receive a training sequence header after receiving the pair of End Delimiters. The host checks the electrical stress test pattern returned on the NB bit lanes to test the SB and NB bit lanes and determine how to configure the channel.

3.3.6 Polling State

The host drives a TS2 pattern to transition the AMBs from the Testing state to the Polling state. The host sends a continuous stream of TS2 patterns to the last AMB to determine the round trip latency of the channel. The host may subsequently and optionally send a TS2 pattern to each intermediate AMB to test if it has aligned its NB merge data timing to the timing of the last AMB and can properly merge its data into the northbound data stream.

The host must continue to send TS2 patterns until it receives at least 4 TS2 patterns on its northbound receivers to provide enough time for AMBs to set their merge timing.

A host supporting transient-tolerant initialization must at least repeat TS2 patterns to the last AMB until it detects multiple properly framed TS2 headers followed by control with AMB_ID equal to the last AMB and incrementing Training_Sequence_ID on any given lane.

Table 3-16 defines the format of the TS2 pattern. The AMB_ID field identifies which AMB should respond by driving TS2 patterns northbound.

The NB_Merge_Disable bit tells the AMB addressed by the AMB_ID field to disable its NB merge data path and to not respond with training sequences, Status frames or Alert frames. The NB_Merge_Disable bit functionality is not limited to the polling state, but remains in effect even during L0. It is also sticky through fast reset. Once an AMB's NB_Merge_Disable bit is set, it will remain set until cleared by a future TS2 pattern targeting this AMB with the bit not set, or a hardware reset. The purpose of the bit is to disable the merge function of an AMB that cannot properly merge data, allowing the rest of the system to continue proper operation.

The Training_Sequence_ID field is used by the northbound training logic to determine the round trip response time of the channel. The host must increment the value in each successive TS2 pattern.

The NB_Width_Capability field communicates the NB bit lane configurations supported by the AMBs. Five NB channel width configurations are defined: 14-bit, 14-bit fail over, 13-bit, 13-bit fail over, and 12-bit as defined in Table 3-15. AMB devices may be implemented that implement a subset of the defined configurations. The fail over configurations are not supported unless the corresponding non-fail over configurations are supported.

Table 3-15 — NB_Width_Capability Field

Field	Value	Description
NBWC [4:0]	1XXXX	14 bit: This configuration delivers 144-bits of read data per frame.
	X1XXX	14 bit Fail Over: Device supports 14-bit fail-over to 13-bit. This configuration delivers 144-bits of read data per frame but with reduced CRC coverage.
	XX1XX	13 bit: This configuration delivers 144-bits of read data per frame.
	XXX1X	13 bit Fail Over: Device supports 13-bit fail-over to 12-bit. This configuration delivers 144-bits of read data per frame but with reduced CRC coverage.
	XXXX1	12 bit: Device supports 12-bits and no fail-over. This configuration delivers 128-bits of read data per frame. CRC coverage is embedded in the 144-bit data payload.

Table 3-16 — Training Sequence – TS2 (Polling)

Group #	Description	Value [11:0]
0	TS2 Header: Continuous string of 10 ones to identify the TS2.	Bit [11:0]: 12b’0111 1111 1110
1	Control: Bit [3:0] – AMB_ID — Host indicates which AMB should return a TS2 pattern. — AMB returns this value. Bit [4] – reserved Bit [5] – NB_Merge_Disable — Host may disable the NB merge data path in this AMB. — No AMB response to this TS2 by the addressed AMB. Bit [11:6] – reserved	Bit [3:0]: AMB_ID value Bit [4]: 1b’0 Bit [5]: 1=disable the NB merge path on this AMB. Bit [11:6]: 6b’0000 00
2	Channel round trip control: Bit [3:0] – Training_Sequence_ID — Used to associate a SB TS2 pattern with its corresponding NB TS2 response. — AMBs return this value in corresponding NB TS2 patterns. Bit [4] – reserved Bit [9:5] – NB_Width_Capability — Host sends 0’s. — AMB indicates northbound channel widths supported by the device. Bit [11:10] – reserved	Bit [3:0]: TSID, zero in initial TS2, incremented in each successive TS2s Bit [4]: 1b’0 Bit [5]: 1=12-bit NB width supported Bit [6]: 1=13-bit fail over NB width supported Bit [7]: 1=13-bit NB width supported Bit [8]: 1=14-bit fail over NB width supported Bit [9]: 1=14-bit NB width supported More than one bit may be set. Bit [11:10]: 2b’00

Table 3-16 — Training Sequence – TS2 (Polling) (Cont'd)

Group #	Description	Value [11:0]
5-3	3 groups of clock training patterns	Bit [11:0]: 12b'1010 1010 1010

Table 3-17 defines the actions of the AMBs while in the Polling state. The AMB devices communicate their northbound channel width capabilities to the host in the returned TS2 pattern. The host and AMBs determine the round trip latency of the channel while in the Polling state.

Table 3-17 — Polling State

	AMB
Entry Condition	Entry from Training state on detection of TS2 header on 2of3 LSB SB Rx inputs.
Action	The following actions are performed: — Receive SB TS2 patterns — If Last_AMB_Flag is set, • Generate TS2 patterns NB with own AMB_ID value (independent of AMB_ID value in SB TS2 pattern) onto all aligned NB bit lanes with the delay specified by the Command_to_Data value. The Command_to_Data delay may be longer than the minimum delay used with the TS1 pattern. The AMB must insert frames of logic 0's to fill the gap between the last TS1 and first TS2 sequence. • Report the channel widths supported in the width capability fields • Ignore the NB_Merge_Disable bit — Else, • Forward SB data patterns • If the AMB_ID value matches the AMB number in TS2 patterns received on 2of3 LSB SB Rx inputs and NB_Merge_Disable bit is set – Disable NB data merge logic – Forward NB data patterns • Else, – Align NB round trip timing using NB TS2 and Training_Sequence_ID on a bit lane by bit lane basis – If the AMB_ID value matches the AMB number in TS2 patterns received on 2of3 LSB SB Rx inputs and NB round trip timing has been aligned on the bit lane, - Generate TS2 patterns NB onto all aligned NB bit lanes merged with the data timing received from the last AMB. The generated TS2 patterns are merged into the data stream to prevent truncation of the TS2 streams from southern AMBs. - Report the channel widths supported in the width capability fields – Else, - Forward NB data patterns
Exit Condition & Next States	Transition to Config state if a SB TS3 header is received on 2of3 LSB SB Rx inputs, *else* put the DRAMs into Self Refresh and transition to Disable state if Electrical Idle detected on 2of3 LSB SB Rx inputs, *else* wait.

NOTE 1 Command_to_Data value is the CMD2DATA Register plus AL+CL

NOTE 2 The Command to Data timing is established in the polling state, and does not change after that point. If the CMD2DATA, AL, or CL values are changed in the AMB registers, the Command to Data timing will not change until the next fast reset.

3.3.6.1 Round Trip Latency Adjustment

The round-trip latency of the FBD channel is determined in the Polling state. The last AMB, the intermediate AMBs and the host use different mechanisms to control and determine the round-trip latency of the channel.

3.3.6.1.1 Last AMB Round-trip Latency Adjustment

The last AMB on the FBD channel controls the timing of the overall channel round-trip latency. Two control registers are used to set the timing in the last AMB:

Command_to_Data – This value sets the time from a DRAM Read command on the southbound FBD channel to when the first DRAM read data is driven onto the northbound FBD channel. The Command_to_Data value consists of the CMD2DATA register plus AL+CL. This parameter has a default value that is common to all AMB implementations as defined in the *FBD AMB Specification* and is used in the initial channel initialization sequence. This value is updated by firmware based on information contained in the SPD EPROM on the DIMM. The updated value takes effect in the next channel initialization sequence. This register is adjusted in UI granularity.

Command_to_Data_Incr – This register is used to increase the round-trip latency of the channel in the unusual cases where the DRAM access time of one or more of the intermediate DIMMs cannot meet the minimum round-trip latency established by the last DIMM. This register is only used by the last DIMM and is ignored by intermediate DIMMs. This value is updated by firmware based on feedback in the status response from the intermediate DIMMs. The updated value takes effect in the next channel initialization sequence. This register is adjusted in full frame granularity.

The sum of the two registers above is used on the last AMB to set the command to response timing for responses from the last DIMM including DRAM data, Configuration read data, and nominal status response data.

The local DRAM read latency as seen by the AMB is controlled by values loaded into the AMB by firmware. Refer to the *FBD AMB Specification* for details.

3.3.6.1.2 Intermediate AMB Response Time Alignment

Intermediate AMB northbound response time alignment is performed in the Polling state. The TS2 pattern is used to adjust the response time of the intermediate AMBs to the timing of the last DIMM. Because the round-trip latency on the channel may be longer than several TS2 patterns, the Training_Sequence_ID field in the southbound TS2 pattern must be matched with a corresponding Training_Sequence_ID number in a northbound TS2 pattern to properly determine the alignment. No bit lane deskew is performed by the intermediate AMBs. The northbound data stream on each bit lane is passed through the intermediate AMBs with a minimum of delay.

The intermediate AMBs on the FBD channel align their timing to the channel round-trip latency established by the last AMB. Two control registers are used to adjust the timing in the intermediate AMBs:

Command_to_Data – This value sets the minimum time from a DRAM Read command on the southbound FBD channel to when the first DRAM read data is available to be driven onto the northbound FBD channel. The Command_to_Data value consists of the CMD2DATA register plus AL+CL. This parameter has a default value used in the initial channel initialization sequence. This value is updated by firmware based on information contained in the SPD EPROM on the DIMM. The updated value takes effect in the next channel initialization sequence. This register is adjusted in UI granularity. This register specifies the minimum time from the DRAM Read command to the DRAM read data. The DRAM read data is normally merged into the northbound data stream to match the DRAM read timing from the last DIMM. This minimum value is used by the data merge logic to determine if the round-trip latency established by the last DIMM can be met by this DIMM.

Command_to_Data_Decr – This register is used to decrease the DRAM read latency for this DIMM. This register allows DIMMs closer to the host to return DRAM read data earlier than when the data would be returned from the last DIMM. This register is only valid on the intermediate AMBs and is ignored by the last AMB. This register subtracts full frame times from the DRAM read latency as driven onto the northbound channel. All read responses are affected by this value except the Status frames returned in response to Sync commands, which retains the same timing as the last DIMM.

The combination of the above two registers implements a Variable Read Latency capability on the FBD channel. Support of the Command_to_Data_Decr register is an option feature and a capability bit in the AMB indicates if it supports the register. For large memory configurations where the latency to the last DIMM could be significantly

longer than the latency to the DIMMs closest to the host, this mechanism can be used to allow the closer DIMMs to return their data one or more frames earlier than if the data was being sourced by the last DIMM. It is up to the host to avoid data collisions when scheduling the data returned on the northbound channel.

The initial channel training sequence uses the default value for the Command_to_Data delay and all intermediate AMBs align to the timing of the last AMB. The Command_to_Data value may be changed by firmware and the change will take effect in subsequent channel training sequences. During the subsequent channel training sequence each intermediate AMB will check to see if its locally read DRAM data will be available in time to be merged into the NB link with the new timing established by the last AMB. If the timing cannot be met the AMB sets the Data_Merge_Error bit in its status responses. Firmware must check to see if any of the Data_Merge_Error bits are set and adjust the Command_to_Data_Incr register in the last AMB to add more time to the round trip channel timing. This may take several iterations until a round trip channel delay is found that all of the AMBs can meet. Details of the procedure to add more time to the round trip channel delay are defined in the *FBD AMB Specification*.

The local DRAM read latency as seen by the AMB is controlled by values loaded into the AMB by firmware. If the DRAM read data is received by the AMB before the data must be driven onto the NB link, the AMB must delay the data the appropriate amount of time until it is driven.

Systems that support the hot add of DIMMs to the channel may require restrictions in the speeds of the DIMMs supported to guarantee that the channel could be initialized during the hot add Fast Reset sequence. The proper Command_to_Data value must be programmed into the “to be added” DIMM through the SMBus before the Fast Reset is performed in order to allow the channel to continue operation with proper timing.

3.3.6.1.3 Host Round-trip Delay Alignment

The host performs round-trip channel delay alignment while in the Polling state. The Training_Sequence_ID field in the TS2 pattern is used to measure the round-trip delay of the channel and to set a value for the memory controller to use to time when to expect read data to be returned on the channel.

3.3.7 Config State

The TS3 training sequence is used to communicate the channel configuration to the AMBs in the Config state. Table 3-18 defines the format of the TS3 pattern.

The AMB with the Last_AMB Flag set should respond by driving the TS3 pattern northbound. The SB_Config field communicates the SB bit lane to be mapped out in case of fail-over. The NB_Config field communicates the NB width/CRC mode to be used on the channel and which bit lane to map out in case of fail-over.

Table 3-18 — Training Sequence – TS3 (Config)

Group #	Description	Value [11:0]
0	TS3 Header: Continuous string of 9 ones followed by 2 zeroes to identify the TS3.	Bit [11:0]: 12b'0011 1111 1110
1	Control: Bit [3:0] – AMB_ID (returned by AMB) — Host fills this field with all 1's. — The last AMB will return its AMB_ID value Bit [4] – reserved Bit [8:5] – Protocol Selection — Host indicates the selected channel protocol — AMB returns 0's Bit [11:9] – reserved	Bit [3:0]: AMB_ID value Bit [4]: 1b'0 Bit [8:5]: Protocol selection field as defined in Table 3-19 Bit [11:9]: 3b'000
2	Channel Configuration: Bit [3:0] – SB Channel Configuration — Host indicates the selected channel configurations — AMB returns the selected channel configuration. Bit [5:4] – reserved Bit [9:6] – NB Channel Configuration — Host indicates the selected channel configurations — AMB returns the selected channel configuration. Bit [11:10] – reserved	Bit [3:0]: SB_Config (See section Section 3.3.7.3) Bit [5:4]: 2b'00 Bit [9:6]: NB_Config (See section Section 3.3.7.3) Bit [11:10]: 2b'01
5-3	3 groups of clock training patterns	Bit [11:0]: 12b'1010 1010 1010

Table 3-19 — Protocol Selection Field

Bit Position	Value	Description
PS[3:0]	0000	DDR2 Base non-ECC (10 SB & 12 NB bit lanes)
	0001	DDR2 Base ECC with 6 bit NB CRC protection (10 SB & 13 NB bit lanes)
	0010	Reserved
	0011	DDR2 Base ECC with 12 bit NB CRC protection (10 SB & 14 NB bit lanes)
	0100	DDR3 Base non-ECC (10 SB & 12 NB bit lanes)
	0101	DDR3 Base ECC with 6 bit NB CRC protection (10 SB & 13 NB bit lanes)
	0110	Reserved
	0111	DDR3 Base ECC with 12 bit NB CRC protection (10 SB & 14 NB bit lanes)
	1xxx	Reserved

Table 3-20 defines the actions of the AMBs while in the Config state.

Table 3-20 — Config State

	AMB
Entry Condition	**Entry from Polling state on detection if TS3 header received on 2of3 LSB SB Rx inputs.**
Action	The following actions are performed: — Receive SB TS3 patterns — If SB_Config field received on 2of3 LSB SB Rx inputs, map out any unused bit lanes, disable unused SB Rx inputs and unused SB Tx outputs if directed by the SB_Config field independent of the AMB_ID value. The initial SB TS3 sequence will be truncated on these bits. — If NB_Config field received on 2of3 LSB SB Rx inputs, configure NB operating mode, map out any unused bit lanes, disable unused NB Rx inputs including Electrical Idle detectors and unused NB Tx outputs if directed by the NB_Config field independent of the AMB_ID value. The in progress NB data pattern from the previous state will be truncated on these bits. — If Last_AMB_Flag is set • Generate NB TS3 patterns with own AMB_ID value on all enabled NB Tx outputs with the delay determined in the Polling State. — Else, • Forward SB data patterns on all enabled SB Tx outputs • Forward NB data patterns on all enabled NB Tx outputs
Exit Condition & Next States	Transitions to L0 state if 4 consecutive NOP frames are received on configured SB Rx inputs, *else* put the DRAMs into Self Refresh and transition to Disable state if Electrical Idle detected on 2of3 LSB SB Rx inputs, *else* wait.

3.3.7.1 Alternate Protocol Selection

Future protocol variants may be supported by the FB-DIMM specification. The host must sequence the AMB devices through the Disable, Calibrate (and back to Disable), Training, Testing, Polling, and Config states in order to transition the AMBs into the active channel L0 state.

During the Config state the protocol selected by the host is communicated to the AMBs in the Protocol Selection field in the TS3 training sequence as shown in Table 3-19 above. The selected protocol defines the SB and NB link widths for that specific protocol in the protocol variant specifications

3.3.7.2 Channel Width Selection

After the bit lanes have been tested the host configures the channel to the selected configuration. The host communicates the selected southbound and northbound channel configurations to the AMB agents during the SB TS3 sequence. The last AMB returns a copy of the configuration selections back to the host in the NB TS3 sequence. The host selects the northbound channel width based on the NB test results as defined in Table 3-21. Label “14” means 14 bit width, label “14F” means 14-bit width with fail-over to 13-bits. Label “13” means 13 bit width, label “13F” means 13-bit width with fail-over to 12-bits. Label “12” means 12-bit width. If no acceptable configuration is found the host disables the channel and reports the error to system software.

Table 3-21 — NB Channel Width Selection

NB Test Results	Valid Configura-tions for Supported Modes						
Bit 13	**Bit 12**	**Bit 11-0**	**14**	**14F**	**13**	**13F**	**12**
Good	Good	Good	Yes	-	Yes	-	Yes
Good	Good	Single	-	Yes	-	Yes	-
Good	Good	Multi	-	-	-	-	-
Good	Bad	Good	-	Yes	-	-	Yes
Good	Bad	Single	-	-	-	-	-
Good	Bad	Multi	-	-	-	-	-
Bad or n/c	Good	Good	-	-	Yes	-	Yes
Bad or n/c	Good	Single	-	-	-	Yes	-
Bad or n/c	Good	Multi	-	-	-	-	-
Bad or n/c	Bad or n/c	Good	-	-	-	Yes	Yes
Bad or n/c	Bad or n/c	Single	-	-	-	-	-
Bad or n/c	Bad or n/c	Multi	-	-	-	-	-

3.3.7.3 Bit Lane Fail-Over

Channel configuration is performed in the Config state. The selected configuration may take advantage of bit lane fail-over if supported by the AMBs. If a single SB bit lane is found at fault, the faulty bit lane is mapped out of the SB link. In the case of a SB multi-bit lane fault the channel is non-functional. If a single NB bit lane is found at fault, the faulty bit lane is mapped out of the NB link. In the case of a NB multi-bit lane fault the channel is non-functional except in the special case of the two most significant NB bit lanes with 14 and 13 bit support. In this special case the channel may be configured into 13 bit fail-over mode.

The host sends a SB_Config field and a NB_Config field in the TS3 pattern to communicate that the AMBs must map out the bad bit lane(s) and get ready for normal frame commands. Table 3-22 and Table 3-23 define the valid codes for the Config fields.

Table 3-22 — SB_Config Field

Bit Position	Value	Description
SB[3:0]	1111	All bit lanes are good (no fail-over).
	1110	Reserved.
	1101	Reserved.
	1100	Reserved.
	1011	Reserved.
	1010	Map out bit lane 10 (Optional - not in base protocol)
	1001	Map out bit lane 9.
	1000	Map out bit lane 8.
	0111	Map out bit lane 7.
	0110	Map out bit lane 6.
	0101	Map out bit lane 5.
	0100	Map out bit lane 4.
	0011	Map out bit lane 3.
	0010	Map out bit lane 2.
	0001	Map out bit lane 1.
	0000	Map out bit lane 0.

Table 3-23 — NB_Config Field

Bit Position	Value	Description
NB[3:0]	1111	All bit lanes are good (no fail-over).
	1110	Map out bit lane 14 (Optional - not in base protocol).
	1101	Map out bit lane 13 (Optional)
	1100	Map out bit lane 12.(Optional)

Table 3-23 — NB_Config Field (Cont'd)

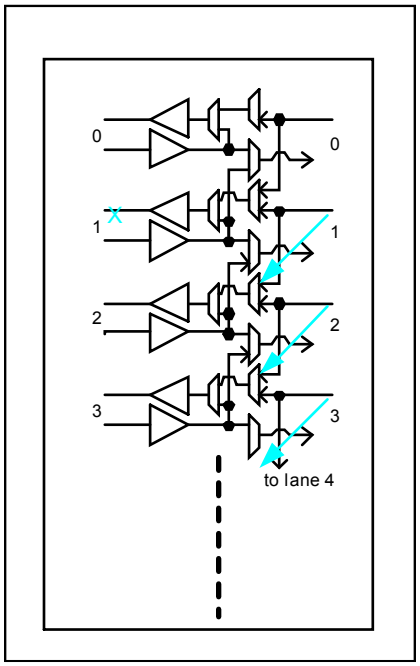
Bit Position	Value	Description
	1011	Map out bit lane 11.
	1010	Map out bit lane 10.
	1001	Map out bit lane 9.
	1000	Map out bit lane 8.
	0111	Map out bit lane 7.
	0110	Map out bit lane 6.
	0101	Map out bit lane 5.
	0100	Map out bit lane 4.
	0011	Map out bit lane 3.
	0010	Map out bit lane 2.
	0001	Map out bit lane 1.
	0000	Map out bit lane 0.

The SB_Config and NB_Config fields are delivered redundantly on all bit lanes to ensure that a faulty bit lane will not interfere with the delivery of the configuration data. If the channel test results indicate that the channel is ready for use the host transition to the L0 state after receiving a NB TS3 sequence on 2of3 LSB NB Rx inputs. If the channel test results indicate that the channel is not functional, the host transitions to the Disable state and the AMBs will follow.

3.3.7.4 Bit Lane Fail-Over Mechanism

To implement bit lane fail-over, both ends of the link must map out a bad bit lane, substitute a good bit lane, and operate in fail-over CRC checking mode. This could be done with a full crossbar switch mechanism but it is recommended that a simpler multiplexer mechanism be used. Figure 3-8 illustrates an example of a simple multiplexing scheme that implements the bit lane re-map using simple 2-input multiplexers. The example implementation could be used for either southbound or northbound links. Normally all of the bit lanes are used for data and CRC transfers. When in fail-over mode, the multiplexers in the bit positions above the failed bit lane switch to use the next higher bit lane and the information normally carried on the most significant bit lane is unavailable.

Figure 3-8 — Figure 3-5: Bit Lane Fail over Mechanism Example



3.3.8 L0 State

The L0 state is entered when the channel is ready to accept channel and DRAM commands. The host must send at least 8 NOP frames on the channel to ensure that the AMB devices have transitioned into the L0 state. In the L0 state both channel commands and DRAM commands will be accepted by the AMBs. Table 3-8 defines the actions of the AMBs while in the L0 state.

Table 3-24 — L0 State

	AMB
Entry Condition	Entry from Config state if 4 consecutive NOP frames are received on configured SB Rx inputs.
Action	The following actions are performed: — Receive SB Frames on configured SB Rx inputs — If Last_AMB_Flag is set • If First_Sync_Received_Flag is reset – If Sync command received, - Set First_Sync_Received_Flag - Reset Idle/Alert Frame LFSR, the LFSR is incremented in each subsequent clock - Send Status Response Frame NB – Else, - Send all zero frames NB • Else If Alert_Flag is set, – Send Alert frames NB • Else If Command response is requested, – Send response NB • Else, – Send Idle frames NB — Else, • Forward SB Frames • If First_Sync_Received_Flag is reset – If Sync command received, - Set First_Sync_Received_Flag - Reset Idle/Alert Frame LFSR, the LFSR is incremented in each subsequent clock - - Send Status Response Frame NB – Else, - Forward NB Frames • Else If Alert_Flag is set, – Send Alert frames NB • Else If Command response is requested, – Merge response into NB data stream • Else, – Forward NB Frames
Exit Condition & Next States	Transition to Recalibrate state when Sync command is received with Enter Recalibrate (ERC) bit set, *else* transitions to L0s state when Sync command is received with Enter L0s (EL0s) bit set, *else* put the DRAMs into Self Refresh, update timing registers to new values (these include Command_to_Data, Command_to_Data_Incr, Command_to_Data_Decr, and other clocking related delays as specified in the *FBD AMB Specification*) and transition to Disable state if Electrical Idle detected on 2of3 LSB SB Rx inputs, *else* wait.

3.3.8.1 Idle Frame Initialization

Host hardware must issue a Sync command on the channel within tClkTrain following entry into the L0 state in order to keep the clock trackers aligned. The first Sync command is used by the AMBs to reset the permuting data pattern of the Idle and Alert frames to their initial value. The first Idle frame follows immediately after the Status frame returned for the first Sync command.

3.3.9 Calibrate State

Internal calibration of the host and AMB receiver and transmitter circuits is performed in the Calibrate state. These steps include: Impedance Matching, Current Source Calibration, and Rx Offset Cancellation. These steps are performed at hardware reset and when exiting extended power saving states. This state is skipped during a Fast Reset. Table 3-25 defines the actions of the AMBs while in the Calibrate state.

Table 3-25 — Calibrate State

	AMB
Entry Condition	Entry from Training state if 1's detected on 2of3 LSB SB Rx inputs for greater than 2 times tClkTrain frames.
Action	The following actions are performed (in order): — Calibrate Tx circuits. — Enable SB Tx outputs and drive all 1's with de-emphasis disabled — Enable NB Tx outputs and drive all 1's with de-emphasis disabled — Calibrate SB Rx input circuits — Enable SB Rx inputs including Electrical Idle detectors — If non-Electrical Idle detected on 2of3 LSB NB Rx inputs • Calibrate NB Rx input circuits • Enable NB Rx inputs including Electrical Idle detectors
Exit Condition & Next States	Transition to the Disable state if Electrical Idle detected on 2of3 LSB SB Rx inputs, *else* wait.

3.3.9.1 Internal Calibration

Upon entering the Calibrate state the IO circuit calibration steps are performed. The host will continue to drive all of the bit lanes to all ones for a time greater than or equal to tCalibrate to give time for the calibration circuits to complete the calibration steps. The northbound Rx inputs should not be calibrated until all ones are received from the AMB south of its position. The IO circuits in the AMBs ignore the signals on the channel during portions of the tCalibrate time in order to perform internal calibration operations but must complete all of the IO circuit calibrations before the tCalibrate time expires. De-empasis must be disabled in the transmitter during the calibration state. After the tCalibrate delay the host must transition the channel back to the Disable state.

3.3.9.2 Recalibrate State

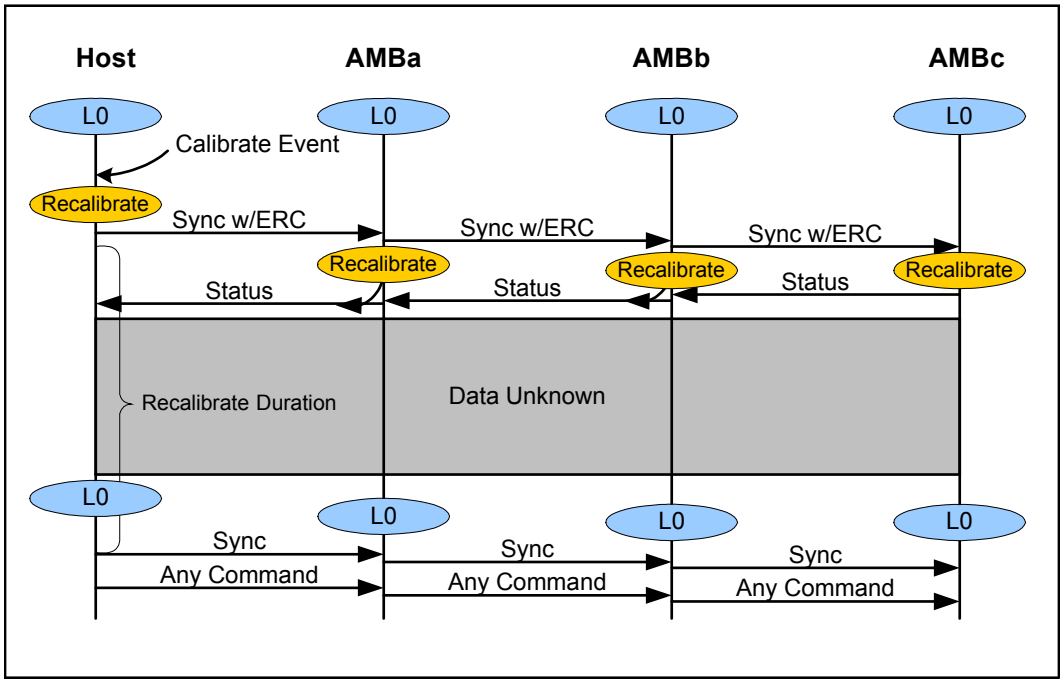
Future technologies may need a mechanism to recalibrate Tx and Rx circuitry after the channel has been initialized. The Recalibrate state provides an opportunity to perform these calibration operations during a window of time that the data on the channel is being ignored. The Recalibrate state is entered when the channel is operating in the L0 state and the Tx and Rx circuits must be recalibrated. In the Recalibrate state the FBD channel is in a quiescent state with NOP frames on the SB links and Idle frames on the NB links. After sending the Sync command with Enter Recalibrate bit set the host must send at least 8 NOP frames before recalibrating its SB Tx outputs. Recalibration must complete at least 8 frames before the end of the recalibration state as defined by the Recalibrate_Duration register to account for skew and the reenabling of clock tracking logic in the receivers. The logic state on the SB link may be corrupted for several bit times during the recalibration procedure. The AMB command decode logic must ignore the data received on the SB inputs while in the Recalibrate state and not indicate CRC errors for these frames. Table 3-26 defines the actions of the AMBs while in the Recalibrate state.

Table 3-26 — Recalibrate State

	AMB
Entry Condition	Entry from L0 state when a Sync command is received with the Enter Recalibrate bit set.
Action	The following actions are performed: — Recalibrate_Timer is initialized to Recalibrate_Duration register value — Decrement Recalibrate_Timer — If Last_AMB_Flag is set, • Ignore SB frame content • If required, calibrate SB Rx input circuits • After the Status frame has been returned for the Sync command with the Enter Recalibrate bit set, – Send Idle frames NB – If required, wait 8 frames and calibrate NB Tx output circuits — Else, • Ignore SB frame content • Forward SB data patterns • If required, calibrate SB Rx input circuits • If required, calibrate SB Tx output circuits • After the Status frame has been returned for the Sync command with the Enter Recalibrate bit set, – Forward NB data patterns – If required, wait 8 frames and calibrate NB Rx input circuits – If required, wait 8 frames and calibrate NB Tx output circuits
Exit Condition & Next States	Transitions to L0 state if Recalibrate_Timer has expired, else, put the DRAMs into Self Refresh and transition to Disable state if Electrical Idle detected on 2of3 LSB SB Rx inputs, else wait.

The recalibration operation may generate false transitions on the bit lanes. The clock tracking logic on each bit lane must ignore these false transitions. The Recalibrate_Duration value written into the AMBs specify the time from the Sync command with the Enter Recalibrate bit set to the Sync command following the recalibration operation. The Recalibrate_Duration value must be greater than or equal to 32 frames and less than or equal to 42 frames. It must not be shorter than the programmed sync interval, which specifies the minimum interval between sync frames. This minimum must be adhered to throughout the process. The host must send a Sync command as the first command after the Recalibrate state to provide bit lane training transitions for the clock training logic. Figure 3-9 is a graphical representation of the transition into and out of the Recalibrate state.

Figure 3-9 — Recalibrate Sequence Flow



3.3.10 L0s State

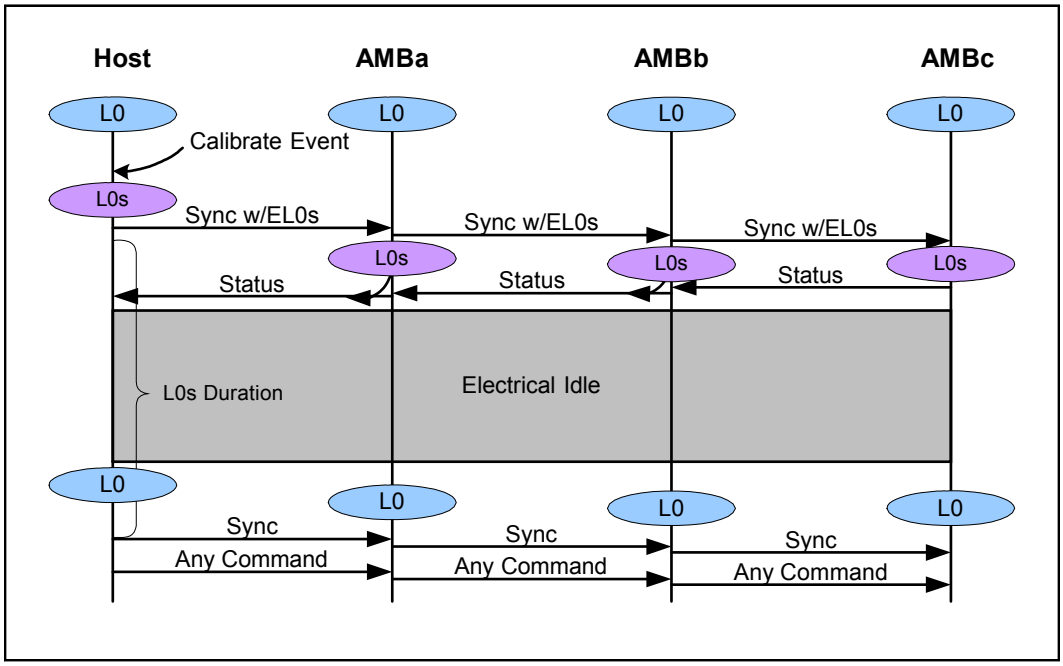
The L0s state is an optional state used in systems that implement aggressive power management. The L0s state is entered when the channel is operating in the L0 state and is momentarily transitioned into this state to reduce power. The channel is kept in the L0s state for a period of time short enough that the clock tracking logic does not lose lock on the data stream. In the L0s state the FBD channel is powered down and the DRAM CKE signals are deasserted to put the DRAMs into the Power Down mode. Table 3-27 defines the actions of the AMBs while in the L0s state.

Table 3-27 — L0s State

	AMB
Entry Condition	Entry from L0 state when a Sync command is received with the Enter L0s bit set.
Action	The following actions are performed: — L0s_Timer is initialized to L0s_Duration register value — Deassert the CKE signals to the DRAM devices (if asserted) — Disable SB Rx inputs including Electrical Idle detectors — Disable SB Tx outputs (if enabled) — Decrement L0s_Timer After the Status frame has been returned for the Sync command with the Enter L0s bit set, • Disable NB Rx inputs including Electrical Idle detectors (if enabled) • Disable NB Tx outputs If L0s_Timer is at L0s_Duration minus an implementation specific Rx and Tx power up time, • Assert the CKE signals to the DRAM devices (if they were asserted) • Enable SB Rx inputs including Electrical Idle detectors (if they were enabled) • Enable NB Tx outputs (if they were enabled) • If Last_AMB_Flag is set, – Send Idle frames NB • Else, – Enable SB Tx outputs (if they were enabled) – Forward SB frames – Enable NB Rx inputs including Electrical Idle detectors (if they were enabled) – Forward NB frames
Exit Condition & Next States	Transitions to L0 state if L0s_Timer has expired, *else* wait.

The host must send a Sync command as the first command after the L0s state to provide bit lane training transitions for the clock training logic. The CKE signals to the DRAM devices must be asserted by the AMB early enough to allow any DRAM command to be located in the frame following the Sync command without violating DRAM timing parameters. Figure 3-10 is a graphical representation of the transition into and out of the L0s state.

Figure 3-10 — L0s Sequence Flow



3.4 Channel Re-initialization

The FBD channel may need to be re-initialized after entering the L0 state. Channel re-initialization is initiated by the host in response to the detection of a channel communication error that could not be corrected with a Soft Channel Reset command. If the host detects a system condition that requires the FBD channel to be re-initialized it sequences the channel through a Fast Reset sequence. The first step of a Fast Reset is a transition to the Disable state.

3.4.1 Enter Self Refresh FSM

When the host controller puts the DRAMs into Self Refresh with an explicit Enter Self Refresh command, it is the host controller's responsibility to adhere to the requirements of self-refresh. In particular the CKE signals must have been high for three clocks before issuing the command, all pages must be closed, and the ODT signal must be low. Note that the ODT signal is shared between ranks.

When the Disabled state is entered from the Testing, Polling, Config, L0 or Recalibrate states, the DRAMs must be put into Self Refresh mode automatically by the AMB without assistance from the host controller. The AMB should not assume any particular operation or mode prior to this. There may be open DRAM pages which must be closed prior to Self Refresh entry, or an auto refresh may have just finished, requiring tRFC to be satisfied prior to entering self refresh. The DRAMs could also be in Self Refresh already.

The following algorithm assures a safe entry into Self Refresh. It does not require tracking the previous state.

Table 3-28 — Self Refresh Entry

Event	Purpose
Wait tCKE (3 clocks for DDR2), and then take all CKE signals high.	3 clocks satisfy the minimum CKE pulse width in case it had just been taken low.
Wait tXSNR (tRFC + 10nS for DDR2).	This satisfies the exit time for the case where the DRAMs were in Self Refresh mode, or the auto refresh to command time for the case where an auto refresh command was just issued.
ODT must be low during the prior two steps if the DRAMs were in self refresh mode.	Existing state machines likely satisfy this.
Issue a Precharge All Command to each rank.	Closes all pages for the case where pages were left open.
Wait tRP	
Issue an Auto Refresh Command to all ranks.	If the DRAMs were in Self Refresh mode, one auto refresh is required before putting the DRAMs back into Self Refresh mode
Wait tRFC	Time from the auto refresh to the next command
Assure ODT signals are low	Existing state machines likely satisfy this.
Issue the Enter Self Refresh Command to all ranks	
Clocks may be disabled or changed one clock later	

The host controller must not enter the TS0 training state until this sequence can be completed. The host may wait until at least 100 clocks after the NB Rx inputs detect Electrical Idle indicating the AMBs have completed the sequence, or use a timeout greater than this calculation:

tCKE + tXSNR + tRP + 2*tRFC + 230 clocks

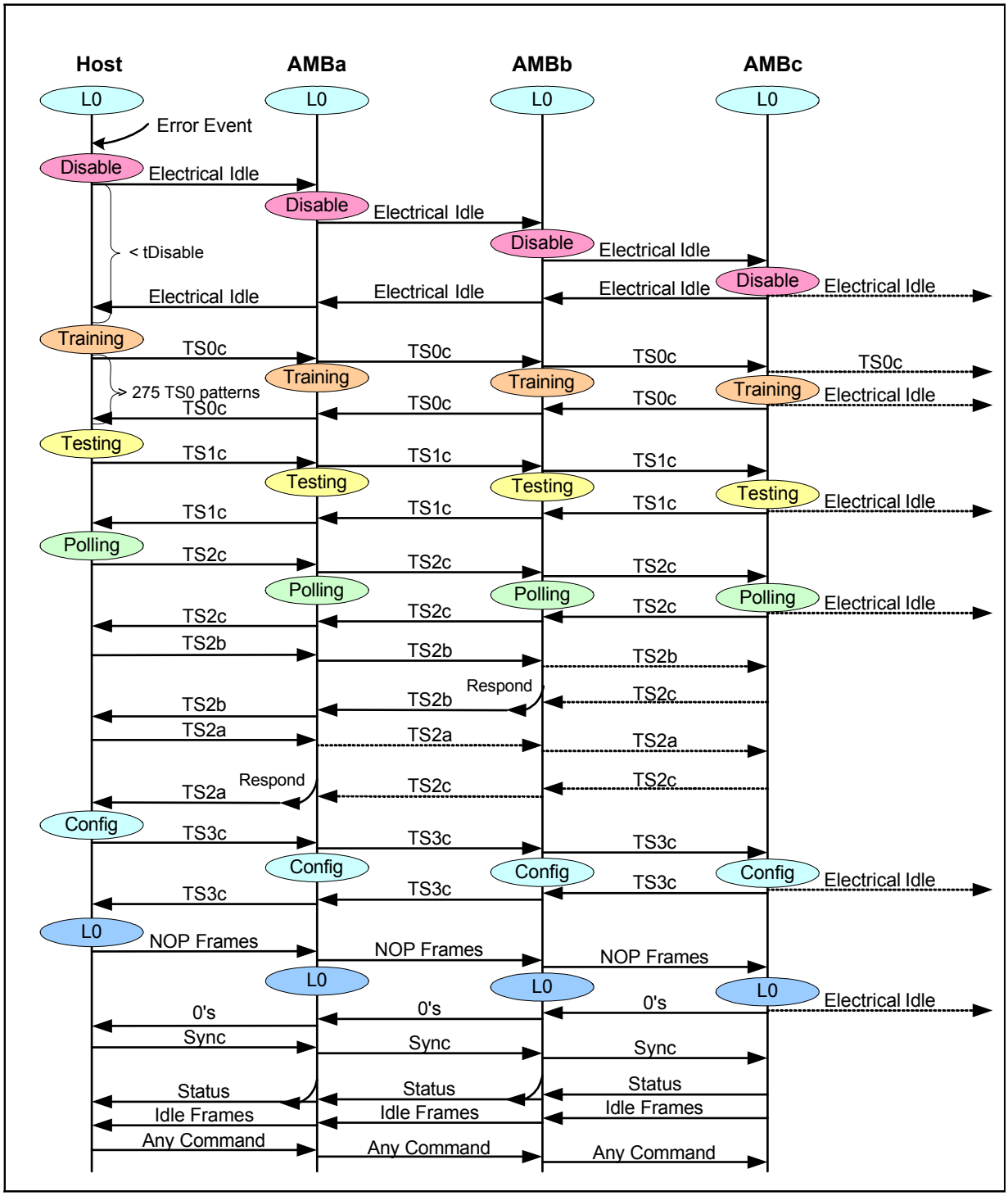
The 230 clocks consists of the time to finish up any pending operations, issue the Auto Refresh and Enter Self Refresh commands, possibly staggered between ranks, and the time for the AMB to complete its self refresh command.

The tXSNR, tRFC, and tRP parameters are programmed into the memory register section in the AMB. They are in units of DRAM clocks.

3.4.2 Fast Reset Flow

Fast Reset is a sequence generated by a host FSM with the goal to reset the channel in less than 12 microseconds to re-establish channel operation with a minimum of system interruption. The Fast Reset mechanism uses the Last_AMB_ID value in an implementation specific register as the target of the initialization sequence. The Fast Reset sequence flow is illustrated in Figure 3-11. The sequence shows a handshake to indicate that the AMBs have transitioned into the Disable state and ready to be trained. If the Electrical Idle is not received in tDisable time the host proceeds with the sequence assuming that one of the AMBs is faulty but an attempt at channel re-initialization will still be made.

Figure 3-11 — Fast Reset Sequence Flow



3.4.2.1 Table 3-21: Fast Reset Duration

The duration of a Fast Reset is made as short as possible to minimize the disruption of service to the system. The sequence and the estimated duration are summariz'ed in Table 3-29 The value of "+ 15" in the following table is an estimate of the delay through the last AMB plus the round trip flight time of the channel.

Table 3-29 — Fast Reset Duration Estimate for a 12 AMB system

State	Timing Paramete	Duration (Frames)	Description
Disable		618	Transition into the Disable state: Host puts the SB Tx outputs in Electrical Idle. The AMBs propagate the Electrical Idle SB and then NB back to the host. tEIPropogate*12 + tEIPropogate*12 + Self_Refresh_Entry.
Training	300 TS0 patterns	3600	The host enables the SB Tx outputs and sends TS0 patterns. The Last AMB returns TS0 patterns NB.
Testing		63	The host sends two TS1 patterns. The Last AMB returns TS1 patterns NB. 12*2 + 15 + 12*2
Polling		39	The host sends TS2 patterns SB. Each AMB returns TS2 patterns NB. 6*2 + 15 + 6*2
Config		39	The host sends TS3 patterns SB. The Last AMB returns TS3 patterns NB. 6*2 + 15 + 6*2
L0		23	The host sends NOP frames SB. The Last AMB returns 0's NB. 4 +15 + 4
Sync		17	The host sends a Sync to reset NB Idle LFSR. AMBs return a Status frame. 1 +15 + 1
Total		4399	Approximate number of frames for Fast Reset.
Duration		13.2	Time in microseconds at DDR2-667 speed.

3.5 Hot-add

The FBD channel does not provide a mechanism to automatically detect and report the addition of a new DIMM south of the currently active last DIMM. It is assumed the system will be notified through some means of the addition of one or more new DIMMs so that specific commands can be sent to the host controller to initialize the newly added DIMM(s) and perform a Hot-Add Reset to bring them into the channel timing domain. The Hot-Add Reset uses the host Hot_Add_AMB_ID register value as the target. If the Hot-Add Reset completed successfully the Hot_Add_AMB_ID value is copied into the host Last_AMB_ID register.

If the channel does not initialize properly the Hot-Add Reset sequence reverts back to the previous host Last_AMB_ID register value and performs the equivalent of a Fast Reset to reestablish the working channel.

It should be noted that the power to the DIMM socket must be removed before a "hot-add" DIMM is inserted or removed. Applying or removing the power to a DIMM socket is a system platform function that is beyond the scope of the FBD channel specifications.

3.5.1 Hot-add AMB Reset

Electrical Idle is present on the SB Tx outputs of the last AMB and the newly added DIMM will transition to or remain in the Disable state.

3.5.2 Hot-add AMB Calibration

Newly added DIMMs must be calibrated before attempting to bring them online. Firmware must set the Drive_Ones_SB bit in the last AMB using a Configuration Write command to enable the SB Tx outputs and drive them with logic ones. The hot added DIMMs detect the logic one state and transition into the Calibrate state and perform the calibration steps. Firmware must also direct the last AMB to calibrate its NB Rx inputs.

3.5.3 Hot-add AMB Testing

Firmware must wait tCalibrate for the calibration steps to complete. After the newly added DIMM(s) is calibrated firmware should read the NB_Data_All_Ones bit field to check that the newly added DIMM(s) went into the Calibrate state and that the required number of NB bit lanes are connected. If the newly added DIMM(s) appears to function properly firmware may initiate a Hot-Add Reset to bring the DIMM(s) on line.

3.5.4 Hot-add AMB Timing

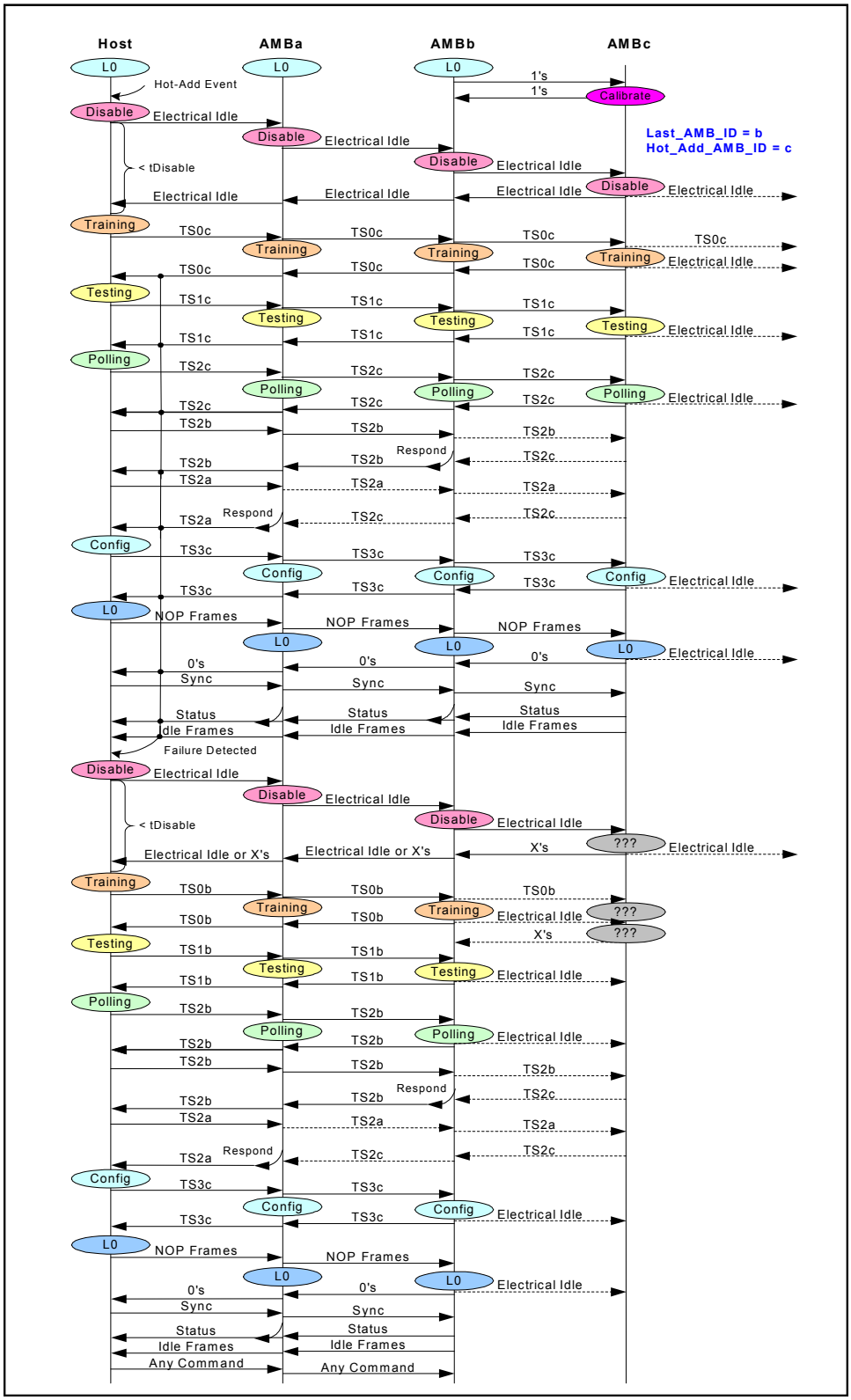
Firmware must set the Command_to_Data delay and the internal timing values in the new DIMM before adding it to the channel with a Fast Reset. This means that the firmware must read the SPD and load all the AMB specific parameters into the new DIMM after calibration and before attempting to bring it into the channel. If this is not done, during the Fast Reset, the new DIMM that becomes the last AMB will have a default value for the Command_to_Data delay and other parameters that may make the link fail initialization with merge errors from one or more of the intermediate AMBs.

3.5.5 Hot-Add Reset Flow

Hot-Add Reset is similar to the Fast Reset Sequence but with the capability to fall back to the previous configuration if the newly added DIMMs fail to initialize. Hot-Add Reset follows the Fast Reset sequence flow illustrated in Figure 3-11 if the newly added DIMM(s) initialize correctly. The Hot-Add Reset sequence follows the flow illustrated in Figure 3-12 if the newly added DIMM(s) fails to initialize. The Hot-Add Reset must re-establish channel operation within 10 microseconds if the newly added DIMM(s) fails to initialize. The Hot-Add Reset mechanism uses the Hot_Add_AMB_ID value in an implementation specific host register as the target of the initialization sequence and falls back to the Last_AMB_ID value if the newly added DIMM(s) fails to initialize.

For added confidence that the newly added DIMM has initialized correctly the host may include additional commands to the DIMMs, such as Configuration Read commands to known register values, before declaring that the newly added DIMM is functioning properly.

Figure 3-12 — Hot-Add Reset Sequence Flow if Failure



3.6 Hot-remove

In order to accomplish removal of DIMMs the host must perform a Fast Reset sequence targeted at the last DIMM that will be retained on the channel. The Fast Reset re-establish the appropriate last DIMM so that the SB Tx outputs of the last active DIMM and the SB and NB outputs of the DIMMs beyond the last active DIMM are disabled. Once the appropriate outputs are disabled the system can coordinate the procedure to remove power in preparation for physical removal of the DIMM if needed.

3.7 Hot-replace

Hot replace of DIMM is accomplished though combining the Hot-Remove and Hot-Add process.

4 Channel Protocol

The host performs all of the scheduling of the southbound and northbound data paths. The FBD DIMMs do not initiate any northbound traffic but instead respond to commands provided by the host. This protocol style results in a memory channel that has deterministic behavior (in the absence of error events) and facilitates the use of two or more FBD channels in lock stepped configurations. The host sends commands and data to the DIMMs in 120-bit (108 in fail-over mode, defined in previous chapter) southbound frames. Similarly the DIMMs return data to the host in 168-bit (156 in fail-over mode), 156-bit (144 in fail-over) or 144-bit northbound frames.

4.1 Southbound Frames

After initialization the host communicates with the AMBs on the channel using southbound frames of information containing commands and data. There are two modes of operation of the southbound channel, normal and fail-over. In normal mode the southbound link is full width and has a stronger CRC code. In fail-over mode the southbound link is reduced in width by one bit and uses a weaker CRC code. The next two sections define the southbound frame formats of the two modes and the CRC coverage of each mode.

4.1.1 Normal Southbound Frames

Normal southbound frames consist of 12 transfers of data delivered on 10 southbound bit lanes. Table 4-30 defines the common features of a normal southbound frame. Each frame starts with four transfers containing what is referred to as the “A” command. The F[1:0] field in the “A” command determines the frame type. The remainder of the “A” command includes 24 bits of command information in the C[23:0] field and a 14-bit CRC in the E[13:0] field. The E[13:0] field provides error detection coverage across the F[1:0], C[23:0] and E[13:0] fields. Because the frame type field is covered by the CRC of the “A” command, the content of the entire frame is invalid if there is a CRC error detected in the “A” command.

Each frame also contains 72-bits of command or data information that is protected by a 22-bit CRC identified as FE[21:0]. Eight bits of the 22-bit CRC, FE[21:14], are located in the 10th bit lane. The remaining 14 bits of the CRC are combined with the 14-bit CRC in the next frame using an exclusive-or function. FE0 is XOR with aE0, FE1 is XOR with aE1, and so on. This mechanism provides strong CRC protection of the last 72 bits of the frame without adding latency to the delivery of the “A” command in the current frame. The CRC of the “A” command can be checked as soon as the first 4 transfers of the frame are received and the “A” command can be used immediately without waiting for the remainder of the frame to arrive. The complexity of this scheme is relatively low compared to the benefit of minimizing DRAM access time.

Table 4-30 — Common Features of Normal Southbound Frames

Transfer	Bit									
	9	8	7	6	5	4	3	2	1	0
0	aE0	aE7	aE8	F0	aC20	aC16	aC12	aC8	aC4	aC0
1	aE1	aE6	aE9	F1	aC21	aC17	aC13	aC9	aC5	aC1
2	aE2	aE5	aE10	aE13	aC22	aC18	aC14	aC10	aC6	aC2
3	aE3	aE4	aE11	aE12	aC23	aC19	aC15	aC11	aC7	aC3
4	FE21									
5	FE20									
6	FE19									
7	FE18									
8	FE17									
9	FE16									
10	FE15									
11	FE14									
	FE0	FE7	FE8							
	FE1	FE6	FE9							
	FE2	FE5	FE10	FE13						
	FE3	FE4	FE11	FE12						

4.1.2 Fail-over Southbound Frames

Fail-over southbound frames consist of 12 transfers of data delivered on 9 southbound bit lanes. The most significant bit lane is not available to carry CRC bits in fail-over mode and the CRC code size is reduced in this mode.
Table 4-31 defines the common features of a southbound frame in fail-over mode. A total of 108 bits are delivered as part of the frame, including 10 bits of CRC. The E[9:0] field provides 10-bit CRC error detection coverage across the "A" command of this frame exclusive or'ed with the 10-bit CRC covering the 72 bit data payload of the previous frame. The CRC code provides good error detection for the period of time until the system can be serviced and the fail-over condition repaired.

Table 4-31 — Common Features of Fail-over Southbound Frames

Transfer	Bit									
	9	8	7	6	5	4	3	2	1	0
0		aE3	aE4	F0	aC20	aC16	aC12	aC8	aC4	aC0
1		aE2	aE5	F1	aC21	aC17	aC13	aC9	aC5	aC1
2		aE1	aE6	aE9	aC22	aC18	aC14	aC10	aC6	aC2
3		aE0	aE7	aE8	aC23	aC19	aC15	aC11	aC7	aC3
4										
5										
6										
7										
8										
9										
10										
11										

4.1.3 Southbound Frame Formats

FBD supports a limited number of southbound frame types as defined in Table 4-32. The F[1:0] field in the first transfer determines the frame type.

Table 4-32 — Southbound Frame Type Encoding

Frame Format	F1	F0	Comments
Command	0	0	Frame contains one or more commands plus optional data
reserved	0	1	
Command+Wdata	1	WSn	Frame contains an “A” command plus 72 bits of Wdata

The “reserved” frame type is not defined and is reserved for future use. First generation FBD implementations must not generate this frame type, and AMBs must ignore this frame type.

4.1.3.1 Command Frame Format

The Command frame contains up to three independent commands that can be executed in parallel by separate DIMMs and in some cases by the same DIMM. Bits in each command specify which DIMM should execute the command. The three commands are referred to as the “A” command, the “B” command and the “C” command. The “A” command is the first command in the frame followed by “B” and then “C”. The “A”, “B” and “C” commands are optional and must be filled with a No Operation command pattern, as defined in Table 4-41, if not used. Unused bits in the 72-bit data payload containing the “B” and “C” commands must be ignored by the AMBs but are included in the 22-bit (or 10-bit in fail-over mode) CRC calculation. The unused bits may contain debug information in some platforms.

Table 4-33 — Command Frame Format

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

4.1.3.2 Command Frame with Data Format

Specific commands, such as configuration register write commands, may need to deliver data to the AMB devices. The Command frame is used by these commands to deliver a data payload with information that cannot be encoded in the command itself. The Command frame contains an “A” command, a “B” command, and a “B” data payload. The command type of the “B” command determines if “B” data is included in the frame. Table 4-34 specifies the format of the data field that is included in a Command frame used by these commands. The data delivered by this frame format always replaces the “C” command slot and is always associated with the “B” command. The “B” data payload contains up to 32 bits of data in the D[31:0] field, and four byte enable bits in the BE[3:0] field. Unused bits in the 72-bit data payload containing the “B” command and “B” data payload must be ignored by the AMBs but are included in the 22-bit (or 10-bit in fail-over mode) CRC calculation. The unused bits may contain debug information in some platforms.

Table 4-34 — Command Frame with Data Format

Transfer	Bit									
	9	8	7	6	5	4	3	2	1	0
0	aE0	aE7	aE8	F0=0	aC20	aC16	aC12	aC8	aC4	aC0
1	aE1	aE6	aE9	F1=0	aC21	aC17	aC13	aC9	aC5	aC1
2	aE2	aE5	aE10	aE13	aC22	aC18	aC14	aC10	aC6	aC2
3	aE3	aE4	aE11	aE12	aC23	aC19	aC15	aC11	aC7	aC3
4	FE21	0	0	0	bC20	bC16	bC12	bC8	bC4	bC0
5	FE20	0	0	0	bC21	bC17	bC13	bC9	bC5	bC1
6	FE19	0	0	0	bC22	bC18	bC14	bC10	bC6	bC2
7	FE18	0	0	0	bC23	bC19	bC15	bC11	bC7	bC3
8	FE17	BE0	D28	D24	D20	D16	D12	D8	D4	D0
9	FE16	BE1	D29	D25	D21	D17	D13	D9	D5	D1
10	FE15	BE2	D30	D26	D22	D18	D14	D10	D6	D2
11	FE14	BE3	D31	D27	D23	D19	D15	D11	D7	D3

4.1.3.3 Command+Wdata Frame Format

The Command+Wdata frame is used to deliver write data to write FIFO structures on each DIMM for future transfer to the DRAM devices. The content of the data payload is not examined by the AMB. The write data is loaded into the write FIFO on the DIMM specified by the combined WSn bits from 3 consecutive Command+Wdata frames. Table 4-35 defines the order of the Wdata address bit delivery. The WS[2:0] bits identify the target DIMM. Each DIMM must speculatively capture the Wdata into an accumulation buffer before writing the Wdata into the write FIFO. Once the target DIMM is identified the data may be written into the target DIMM write FIFO. If the WS[2:0] bits identify a different target DIMM the data is discarded.

Table 4-35 — Wdata Address Delivery

Wdata Frame #	F1	F0
0	1	WS0
1	1	WS1
2	1	WS2
3	1	0

The F0 bit in Wdata Frame #3 is reserved, and should be ignored by the AMB.

Table 4-36 defines the mapping of the data pins of 18 4-bit DRAM devices to the frame data bits. In the Cn.Dn labels, Cn is the DRAM chip number and Dn is the DRAM data bit number.

Table 4-36 — Command+Wdata Frame Format (4-bit Devices)

Transfer	Bit									
	9	8	7	6	5	4	3	2	1	0
0	aE0	aE7	aE8	WSn	aC20	aC16	aC12	aC8	aC4	aC0
1	aE1	aE6	aE9	F1=1	aC21	aC17	aC13	aC9	aC5	aC1
2	aE2	aE5	aE10	aE13	aC22	aC18	aC14	aC10	aC6	aC2
3	aE3	aE4	aE11	aE12	aC23	aC19	aC15	aC11	aC7	aC3
4	FE21	C17.D0	C15.D0	C13.D0	C11.D0	C9.D0	C7.D0	C5.D0	C3.D0	C1.D0
5	FE20	C17.D1	C15.D1	C13.D1	C11.D1	C9.D1	C7.D1	C5.D1	C3.D1	C1.D1
6	FE19	C17.D2	C15.D2	C13.D2	C11.D2	C9.D2	C7.D2	C5.D2	C3.D2	C1.D2
7	FE18	C17.D3	C15.D3	C13.D3	C11.D3	C9.D3	C7.D3	C5.D3	C3.D3	C1.D3
8	FE17	C18.D0	C16.D0	C14.D0	C12.D0	C10.D0	C8.D0	C6.D0	C4.D0	C2.D0
9	FE16	C18.D1	C16.D1	C14.D1	C12.D1	C10.D1	C8.D1	C6.D1	C4.D1	C2.D1
10	FE15	C18.D2	C16.D2	C14.D2	C12.D2	C10.D2	C8.D2	C6.D2	C4.D2	C2.D2
11	FE14	C18.D3	C16.D3	C14.D3	C12.D3	C10.D3	C8.D3	C6.D3	C4.D3	C2.D3

The mapping of the data pins of nine 8-bit DRAM devices to the frame data bits is defined in Table 4-37.

Table 4-37 — Command+Wdata Frame Format (8-bit Devices)

Transfer	Bit									
	9	8	7	6	5	4	3	2	1	0
0	aE0	aE7	aE8	WSn	aC20	aC16	aC12	aC8	aC4	aC0
1	aE1	aE6	aE9	F1=1	aC21	aC17	aC13	aC9	aC5	aC1
2	aE2	aE5	aE10	aE13	aC22	aC18	aC14	aC10	aC6	aC2
3	aE3	aE4	aE11	aE12	aC23	aC19	aC15	aC11	aC7	aC3
4	FE21	C9.D0	C8.D0	C7.D0	C6.D0	C5.D0	C4.D0	C3.D0	C2.D0	C1.D0
5	FE20	C9.D1	C8.D1	C7.D1	C6.D1	C5.D1	C4.D1	C3.D1	C2.D1	C1.D1
6	FE19	C9.D2	C8.D2	C7.D2	C6.D2	C5.D2	C4.D2	C3.D2	C2.D2	C1.D2
7	FE18	C9.D3	C8.D3	C7.D3	C6.D3	C5.D3	C4.D3	C3.D3	C2.D3	C1.D3
8	FE17	C9.D4	C8.D4	C7.D4	C6.D4	C5.D4	C4.D4	C3.D4	C2.D4	C1.D4
9	FE16	C9.D5	C8.D5	C7.D5	C6.D5	C5.D5	C4.D5	C3.D5	C2.D5	C1.D5
10	FE15	C9.D6	C8.D6	C7.D6	C6.D6	C5.D6	C4.D6	C3.D6	C2.D6	C1.D6
11	FE14	C9.D7	C8.D7	C7.D7	C6.D7	C5.D7	C4.D7	C3.D7	C2.D7	C1.D7

Table 4-38 defines the mapping of the data pins of eight 8-bit DRAM devices to the frame data bits used for non-ECC applications. The DM[7:0] field carries data mask bits for each of the DRAM devices.

Table 4-38 — Command+Wdata Frame Format Non-ECC 8-bit Devices)

Transfer	Bit									
	9	8	7	6	5	4	3	2	1	0
0	aE0	aE7	aE8	WSn	aC20	aC16	aC12	aC8	aC4	aC0
1	aE1	aE6	aE9	F1=1	aC21	aC17	aC13	aC9	aC5	aC1
2	aE2	aE5	aE10	aE13	aC22	aC18	aC14	aC10	aC6	aC2
3	aE3	aE4	aE11	aE12	aC23	aC19	aC15	aC11	aC7	aC3
4	FE21	DM0	C8.D0	C7.D0	C6.D0	C5.D0	C4.D0	C3.D0	C2.D0	C1.D0
5	FE20	DM1	C8.D1	C7.D1	C6.D1	C5.D1	C4.D1	C3.D1	C2.D1	C1.D1
6	FE19	DM2	C8.D2	C7.D2	C6.D2	C5.D2	C4.D2	C3.D2	C2.D2	C1.D2
7	FE18	DM3	C8.D3	C7.D3	C6.D3	C5.D3	C4.D3	C3.D3	C2.D3	C1.D3
8	FE17	DM4	C8.D4	C7.D4	C6.D4	C5.D4	C4.D4	C3.D4	C2.D4	C1.D4
9	FE16	DM5	C8.D5	C7.D5	C6.D5	C5.D5	C4.D5	C3.D5	C2.D5	C1.D5
10	FE15	DM6	C8.D6	C7.D6	C6.D6	C5.D6	C4.D6	C3.D6	C2.D6	C1.D6
11	FE14	DM7	C8.D7	C7.D7	C6.D7	C5.D7	C4.D7	C3.D7	C2.D7	C1.D7

The mapping of four 16-bit DRAM devices would concatenate the data from each pair of 8-bit devices above into a 16-bit data field for the 16-bit devices. Each DM[7:0] bit would still correspond to 8-bits of data.

4.2 Southbound Commands

Commands are located in the “A”, “B” and “C” command slots of southbound frames. Any command may be placed in any “A”, “B”, or “C” command slot except as follows:

The Soft Channel Reset and Sync commands may only be located in the “A” command slot.

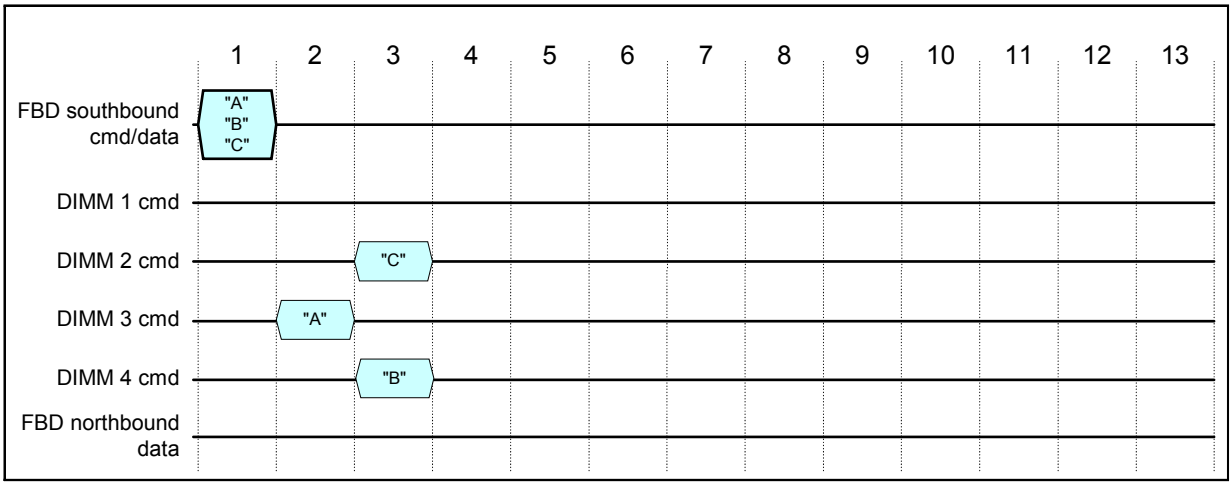
The Configuration Write command may only be placed in the “B” command slot.

4.2.1 Command Delivery Timing

A DRAM command located in the “A” command may be delivered to the DRAM devices as soon as the 14-bit (10-bits in fail-over) CRC is checked. This minimizes DRAM access latency by allowing the command to be delivered after the first 4 transfers of the frame have been received.

The “A” command is transferred immediately to the DRAM pins with minimum delay whereas the “B” and “C” command are delivered one DRAM clock later. To minimize memory access latency the read related Activate, Read (if the page is open) and explicit Precharge commands to a rank of DRAM devices should be placed in the “A” command, if possible. Table 4-13 illustrates the delivery of the three potential commands in a frame to three separate DRAM channels. Command “A” is delivered in this case to the DRAM devices on DIMM 3 as soon as the command can traverse the AMB buffer. The “B” and “C” commands are delayed and presented to two other DRAM channels on the following clock.

Figure 4-13 — "A", "B", and "C" Command Delivery Timing



4.2.2 Concurrent Command Delivery Rules

Commands may be issued in any combination, as long as they do not collide on any DRAM pin or FBD data slot, and follow a few additional rules below.

Commands from the "B" or "C" command slots of frame *n* and the "A" command slot of frame *n+1* will be executed in the AMB on the same DRAM clock. This definition of DRAM Clock is used below.

DRAM Command and Address Pins

Only one of the following commands may target a particular DIMM in the same DRAM clock due to collisions on the DRAM command and address pins. Multiple commands within this list may be issued if each targets a different DIMM, as long as there is no collision on the FBD channel northbound data bus:

Activate, Write, Read, Precharge Single, Precharge All, Auto Refresh, and Enter Self Refresh.

DRAM CKE Pins

Only one of the following commands may target a particular DIMM in the same DRAM clock due to collisions on the CKE pins:

Enter Self Refresh, Exit Self Refresh, DRAM CKE per DIMM, DRAM CKE per Rank, and Enter Power Down.

Note that DRAM CKE commands may target a single DIMM, or 4 or 8 DIMMs at once. When multiple DIMMs are targeted by a command, no other command affecting the CKE pins may be issued to any of the targeted DIMMs.

DRAM Data and Strobe Pins

Commands cannot be issued to a DIMM that would cause collisions on the DRAM data and strobe pins within a DIMM. In addition, all turnaround times for the DRAM data and strobe pins must be observed.

FBD Northbound Data Bus

Commands cannot be issued on the channel that would cause collisions on the FBD Northbound data bus. Commands that generate data on the northbound data bus are:

Read, Read Config Reg, and Sync

The order of responses must be preserved. Commands issued following a Sync command with SD > 0 must not return data before or on top of the Sync status return.

Other Restrictions

Only one outstanding configuration read or write register transaction is allowed on the channel. A configuration register read begins with the command and ends with the data being returned to the host. A configuration write begins with the command and ends when the read data would have been returned if the command were a Read Config Reg. This is the same point that an Alert Frame would be generated if there were a CRC error on the Write Config Reg command. Allowing only one outstanding configuration transaction on the bus allows for proper replay of the Write Config Reg command following an Alert Frame.

A Soft Channel Reset requires NOP commands in all other command slots in the previous DRAM clock, the current DRAM clock, and the next 4 DRAM clocks.

Only one In-band Debug event may be sent within a DRAM clock.

The host controller is responsible for state and timing of the CKE pins vs. DRAM commands based on the DRAM specifications. A DRAM command and CKE command may target the same DIMM on the same DRAM clock provided that the DRAM specifications are met.

Examples:

A DRAM command may be issued to rank 1 on the same DRAM clock as a DRAM CKE per Rank command that changes the CKE of rank 0 while retaining a 1 on the CKE of rank 1.

A DRAM command may be issued to DIMM 2 on the same DRAM clock as a DRAM CKE Command per DIMM that targets all DIMMs, but retains the state the CKEs of DIMM 2 as 1.

4.2.3 Command Encoding

Commands are encoded into the 24 bit C[23:0] fields of Command frames. Table 4-39 defines the bit mapping of an example DRAM configuration and the channel commands into the C[23:0] field.

Table 4-39 — Command Encoding

DRAM Cmnds	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Activate	DS2	DS1	DS0	1	DRAM Addr		RS	DRAM Bank & Address																
Write	DS2	DS1	DS0	0	1	1	RS	DRAM Bank & Address																
Read	DS2	DS1	DS0	0	1	0	RS	DRAM Bank & Address																
Precharge All	DS2	DS1	DS0	0	0	1	RS	X	X	X	X	1	1	1	X	X	X	X	X	X	X	X	X	X
Precharge Single	DS2	DS1	DS0	0	0	1	RS	DRAM Bank				1	1	0	X	X	X	X	X	X	X	X	X	X
Auto (CBR) Refresh	DS2	DS1	DS0	0	0	1	RS	X	X	X	X	1	0	1	X	X	X	X	X	X	X	X	X	X
Enter Self Refresh	DS2	DS1	DS0	0	0	1	RS	X	X	X	X	1	0	0	X	X	X	X	X	X	X	X	X	X
Exit Self Refresh / Exit Power Down	DS2	DS1	DS0	0	0	1	RS	X	X	X	X	0	1	1	X	X	X	X	X	X	X	X	X	X
Enter Power Down	DS2	DS1	DS0	0	0	1	RS	X	X	X	X	0	1	0	X	X	X	X	X	X	X	X	X	X
reserved	X	X	X	0	0	1	X	X	X	X	X	0	0	X	X	X	X	X	X	X	X	X	X	X

Note: The values in "X" fields in non-reserved commands above may be driven onto the DRAM device pins.

Channel Cmnds							OP3	OP2	OP1	OP0														
Debug: In-band Events	EV7	EV6	EV5	0	0	0	1	1	1	1	1	EV4	EV3	EV2	EV1	EV0	PV7	PV6	PV5	PV4	PV3	PV2	PV1	PV0
Debug: Relative Timing	PH5	PH4	PH3	0	0	0	1	1	1	1	0	PH2	PH1	PH0	RT9	RT8	RT7	RT6	RT5	RT4	RT3	RT2	RT1	RT0
Debug: Exposed Info	EX16	EX15	EX14	0	0	0	1	1	1	0	EX13	EX12	EX11	EX10	EX9	EX8	EX7	EX6	EX5	EX4	EX3	EX2	EX1	EX0
reserved	X	X	X	0	0	0	1	1	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
reserved	X	X	X	0	0	0	1	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
DRAM CKE per DIMM	DS2	DS1	DS0	0	0	0	0	1	1	1	BCST	X	X	X	X	X	DE7	DE6	DE5	DE4	DE3	DE2	DE1	DE0
DRAM CKE per Rank	DS2	DS1	DS0	0	0	0	0	1	1	0	BCST	X	X	X	X	X	D3 R1	D3 R0	D2 R1	D2 R0	D1 R1	D1 R0	D0 R1	D0 R0
Write Config Reg	DS2	DS1	DS0	0	0	0	0	1	0	1	DS3	TID	X	A10	A9	A8	A7	A6	A5	A4	A3	A2	0	0
Read Config Reg	DS2	DS1	DS0	0	0	0	0	1	0	0	DS3	X	X	A10	A9	A8	A7	A6	A5	A4	A3	A2	0	0
reserved	X	X	X	0	0	0	0	0	1	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Soft Channel Reset	X	X	X	0	0	0	0	0	1	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Sync	X	X	X	0	0	0	0	0	0	1	X	SD1	SD0	X	X	X	X	IER	ERC	EL0s	X	X	R1	R0
Channel NOP	X	X	X	0	0	0	0	0	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Note: All unused encodings are reserved. "X" values should be driven by the host to zero and ignored by the AMB.

4.2.4 DRAM Commands

DRAM commands are generated by the host to access the DRAM devices behind each AMB buffer. The host has access to the DRAM devices as if the devices were directly connected to the host. The DS[2:0] field directs the command to one of the eight possible DRAM DIMMs on the FBD channel. The AMB decodes the DRAM commands and generates the control signals to the DRAM devices. The command delivery on the DRAM address and control signals (excluding CKE) use 1n command timing. 1n command timing means that the commands are present on the DRAM pins for a single clock cycle. The exact mapping of the control signals delivered to the DRAM devices are defined in the *FBD AMB Specification*.

AMB buffers may support more than one DRAM technology. The details of the mapping of bank and address bits from the commands to the DRAM devices are specified in the *FBD AMB Specification*. An example mapping is shown in Table 4-40. For complete details of the DRAM command encoding refer to the JEDEC SDRAM data sheets.

In the following table, the RS (Rank Select) bit specifies to the AMB which memory ranks located behind the buffer should be accessed. The other labels correspond to the familiar labels in the SDRAM data sheets. Rows labeled with an "*" are speculative and may change as the JEDEC SDRAM data sheets mature.

Bit position 10 is used in the command encoding of the Precharge Single and Precharge All commands to allow the command bit to be mapped directly onto the DRAM address bit 10 to match the DRAM AP bit usage.

Table 4-40 — DRAM Command Mapping Examples

DDR2 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
4Gb (1Gbx4) *	Row	1	A15	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
4Gb (1Gbx4) *	Row	1	X	A14	RS	A13	B2	B1	B0	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
2KB page	Col	0	1	r/w	RS	X	B2	B1	B0	A12	A11	AP	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0

DRAM Read and Write commands always transfer complete bursts of data determined by the Burst Length field programmed into the DRAM MRS registers. A burst length of 4 will transfer 36 bytes and a burst length of 8 will transfer 72 bytes to/from each ECC DIMM. Non-ECC memory DIMMs support the Data Mask function.

Write accesses transfer the data from the write data FIFO located inside the AMB device on the DIMM. A register instructs the AMB when to drive the data after the Write command. The DDR2 specific Off-Chip Driver (OCD) Impedance Adjust command (EMRS access with A[9:7] = 100) also transfers data from the write data FIFO to the DRAM devices.

The host is responsible for memory ordering, FBD channel scheduling, and error handling.

4.2.5 Channel Commands

Channel commands include the Sync command, miscellaneous DRAM commands, configuration register read and write commands, and miscellaneous maintenance commands. Channel commands may include a DS[2:0] field to specify which DIMM the command is addressing, a 4-bit operation code field to define the command type, and an 11-bit address field. Table 4-41 defines the encoding of the Channel commands. The individual configuration registers are defined in the *FBD AMB Specification* Register chapter.

Table 4-41 — Channel Command Encoding

Field	Value	Description
DS [2:0] or DS [3:0]	All	DIMM Select: This field specifies which DIMM is being addressed by this command. The SMBus address straps located on the DIMM connector are used by the AMB to give each DRAM DIMM on the channel a unique ID between 0h and 7h. FBD Channel repeater buffers have an AMB ID in the range 8h to Bh. FBD Logic Analyzer buffers are transparent and do not respond to Channel or DRAM commands. The value of Fh is used as a broadcast and addresses all of the AMB devices. Notes: Only the DRAM CKE Control, Read Configuration Register, and Write Configuration Register commands use this field.
OP [3:0]	1111	In-band Debug Event: Command bit 13 = 1. This command is broadcast to all AMB and Logic Analyzer Interface buffers to deliver in-band debug events in the southbound direction. It is used primarily to augment triggering mechanisms in a buffer device acting as a Logic Analyzer Interface (LAI) repeater on the link. Eight Debug Event Bits are supported in the EV [7:0] field. A Debug Parameter Value associated with the debug event is contained in the PV [7:0] field.
		Relative Timing Exposure: Command bit 13 = 0. This command is broadcast to all AMB and Logic Analyzer Interface buffers to deliver relative timing exposure in the southbound direction. It is used primarily to facilitate multi channel trace correlation in a buffer device acting as a Logic Analyzer Interface (LAI) repeater on the link. Six phase bits in the PH[5:0] field communicate the encoded boundary transfer phase. Ten timing bits in the RT[9:0] field communicate the relative time of transfer across boundaries.
	1110	Debug Information Exposure: This command is broadcast to all AMB and Logic Analyzer Interface buffers to deliver host specific debug information in the southbound direction. It is used primarily to augment triggering mechanisms in a buffer device acting as a Logic Analyzer Interface (LAI) repeater on the link. Seventeen information bits in the EX[16:0] field communicate debug information.
	1000-1101	Reserved.
	0111	DRAM CKE per DIMM Control: This command manipulates the CKE pins on a per DRAM DIMM basis. See section **Section 4.2.5.1** for details.
	0110	DRAM CKE per Rank Control: This command manipulates the individual CKE pins on up to 4 DRAM DIMMs simultaneously. See section **Section 4.2.5.1** for details.
	0101	Write Configuration Register: This command is only allowed in the "B" command slot. The data located in the data payload of the frame is written into the configuration register addressed by the A [10:2] field. Registers are addressed in 32 bit quantities, with the byte enables providing write addressing down to a byte level. A [10:8] specifies the function number, and A [7:2] specifies the register number if using a PCI Config style addressing scheme. The host toggles the Transaction Identifier (TID) bit in each successive Write Configuration Register command. The AMB copies the TID bit into an internal Last_TID flag when it processes the Write Configuration Register command. The Last_TID flag is "sticky" and is retained in the AMB during a Reset Channel Command or Fast Reset. The AMB must ignore the Write Configuration Register command if the TID bit in the command matches the Last_TID flag value. The initial value of the Last_TID flag upon hardware RESET# is 0. All AMBs update their Last_TID flag on config writes regardless of whether they were addressed by the write. The four Byte Enables in the data payload specify which of four data bytes are written into the configuration register. The data payload is not captured as write data by the write FIFOs on the DIMMs. Only one Read or Write Configuration command may be outstanding on the channel at a time.
	0100	Read Configuration Register: The configuration register addressed by the A [10:2] field is read. A [10:8] specifies the function number, and A [7:2] specifies the register number if using a PCI Config style addressing scheme. The read data is returned in a northbound data frame transferred in the same timeslot that would be used if this command were a DRAM Read Command. The full 32-bit data value is returned within the data frame in the least significant 32 bits of the first data payload. Zeros fill the remainder of the payload. Only one Read or Write Configuration command may be outstanding on the channel at a time.
	0011	Reserved

Table 4-41 — Channel Command Encoding (Cont'd)

Field	Value	Description
	0010	Soft Channel Reset: This command is always located in the "A" command of a Command frame with the "B" and "C" command slots filled with a training pattern. The host broadcasts this command in response to an Alert frame to bring the channel into a known state without issuing a Fast Reset. This command is used to recover from lost commands due to transient bit errors on the channel. This command clears the command state of the channel. The Write FIFOs in all AMBs are invalidated and all contents are discarded.
	0001	Sync Command: This command performs two functions: 1. This command instructs all of the DIMMs to return information in a Status frame. Each DIMM provides its status information on a different bit lane. 2. The host must ensure that a sufficient number of transitions have been generated on all of the bit lanes to keep the northbound derived clock trackers locked to the data stream. This command may be issued to request a Status frame on the northbound channel with the required bit transitions. This command is always located in the "A" command of a Command frame. The data payload is not captured as write data by the write FIFOs on the DIMMs. The data payload has sufficient data transitions to keep the southbound derived clock trackers locked to the data stream when issued periodically. The requested Status frame is normally returned in the same timeslot that would be used if this command were a DRAM Read Command. The SD [1:0] field (Status Delay) allows the return status data to be delayed up to 3 frames. The IER bit indicates that commands and CRC errors should be ignored by the AMB until after the next reset. The host will send a sync packet with the IER bit set prior to a fast reset to avoid spurious CRC errors and possible spurious commands caused by the electrical idle. The ERC bit indicates that the channel should transition from the L0 state to the Recalibration state for 32 to 42 frames as specified in the Recalibrate_Duration register. The EL0s bit indicates that the channel should transition from the L0 state to the L0s state for 32 to 42 frames as specified in the L0s_Duration register. The ERC and EL0s bits are mutually exclusive and my not both be set at the same time. The R[1:0] field allows different status bit to be returned as defined in the *FBD AMB Specification* Register chapter.
	0000	No Operation (NOP): This command is a placeholder for a command when the host has no other commands to issue. This command may be located in the "A", "B", or "C" command slots.

4.2.5.1 CKE Control Commands

Two versions of the CKE control command allows for individual rank control, where up to 4 DIMMs may be targeted at once, or per DIMM control, where all 8 DIMMs can be accessed from a single command.

The CKE control commands may be in slot A, B, or C, and will affect the CKE pins for the addressed DIMM(s) with the same timing as a DRAM command, based on slot location. Multiple CKE commands may be included in one frame as long as no more than one of the commands targets any one DIMM on the same DRAM clock.

The Host Controller is responsible for CKE timing with respect to the DRAM protocol, including the explicit Self Refresh command. The AMB will not do any protocol checking.

The AMB **IS** responsible for CKE timing when entering self refresh automatically based on the link going to the Disable state or failing to receive a sync frame for 2 times tClkTrain frames. The AMB must take CKE high, wait at least the required number of clocks (3 for DDR2), and then take it low for the Enter Self Refresh command.

The **Per DIMM** CKE command allows all 8 DIMMs to be targeted by a single command. There is one bit per DIMM, with the CKE for all ranks on a DIMM affected by that bit. DE0 is used by DIMM0, DE1 for DIMM1, etc. A 1 takes CKE high and a 0 takes CKE low.

If the BCST bit is 1, the command targets all DIMMs.

If the BCST bit is 0, the command targets only one DIMM as specified by DS [2:0]. The bit encodings remain the same. I.E. if DIMM2 was addressed, it would use DE2, with the others being don't cares.

The **Per Rank** CKE command allows for individual Rank CKE control. Two bits are defined per rank, allowing up to 4 DIMMs to be addressed simultaneously. See the table below for the bit positions. A 1 takes CKE high and a 0 takes CKE low.

Table 4-42 — Table 4-13: Per Rank CKE Command Bit Positions

DIMM	Rank	Bit Position
0	0	0
0	1	1
1	0	2
1	1	3
2	0	4
2	1	5
3	0	6
3	1	7
4	0	0
4	1	1
5	0	2
5	1	3
6	0	4
6	1	5
7	0	6
7	1	7

If the BCST bit is a 1, the command targets 4 DIMMs at once. DS [2] determines whether DIMMs 3:0 or DIMMs 7:4 are targeted. DS [1:0] are ignored in this case and may contain any value.

If the BCST bit is a 0, the command targets only one DIMM as specified by DS [2:0]. The bit encodings remain the same. I.E. if DIMM2 was addressed, it would use bit 4 for rank 0 and bit 5 for rank 1 with the others being don't cares.

The AMB may float the command and address signals to the DRAMs when all CKEs are low for that DIMM in order to reduce power.

4.2.5.2 Soft Channel Reset Command

The Soft Channel Reset command may be used to attempt to recover from a transient bit failure on the channel. In the case of a minor transient bit error a single or a small group of commands may be corrupted. The AMB will detect the corruption as a CRC error and will ignore the corrupted commands and report the error to the host with Alert frames.

The host may issue a Soft Channel Reset command to acknowledge the receipt of the Alert frames and reset the command state of the AMBs. The Soft Channel Reset command must be preceded by at least 1 NOP frame and followed by at least 4 NOP frames. The AMB recognizes the Soft Channel Reset command while ignoring all others and resets its internal command state. The following actions are performed by the AMB:

- Discontinue Alert frame generation and generate Idle frames or forward NB traffic in the frame that a Status return would be located if the Soft Channel Reset command was a Sync command.
- Discard all data content in the Write FIFO
- Reset the DIMM target Write FIFO WS[2:0] field tracking state machine

The host may follow the Soft Channel Reset command (and the 4 NOP frames) with a sequence of DRAM commands to clear the command state of the DRAM devices. The sequence may look something like this:

a) Assert CKE to all ranks

b) Wait the appropriate number of clocks

c) Issue a Precharge All command to all ranks

If the Soft Channel Reset itself is corrupted the stream of Alert frames will continue and the host may perform a Fast Reset to reinitialize the channel. Table 4-43 defines the format of the Soft Channel Command frame. The training pattern in the frame provides transitions for clock tracking logic to reduce the likelihood that the channel will need to be Fast Reset in the case where the corrupted command sequence contained a Sync command. The 14-bit remainder of the 22-bit CRC is set to zero in the next frame.

Table 4-43 — Soft Channel Reset Command Frame Format

Transfer	Bit									
	9	8	7	6	5	4	3	2	1	0
0	aE0	aE7	aE8	F0=0	0	OP2=0	0	0	0	0
1	aE1	aE6	aE9	F1=0	0	OP3=0	0	0	0	0
2	aE2	aE5	aE10	aE13	0	0	OP0=0	0	0	0
3	aE3	aE4	aE11	aE12	0	0	OP1=1	0	0	0
4	1	0	1	0	1	0	1	0	1	0
5	0	1	0	1	0	1	0	1	0	1
6	1	0	1	0	1	0	1	0	1	0
7	0	1	0	1	0	1	0	1	0	1
8	1	0	1	0	1	0	1	0	1	0
9	0	1	0	1	0	1	0	1	0	1
10	1	0	1	0	1	0	1	0	1	0
11	0	1	0	1	0	1	0	1	0	1

4.2.5.3 Sync Command

The FBD channel periodically requires a minimum number of transitions on each bit lane to maintain clock recovery synchronization. The host must periodically send a Sync command on the channel to maintain the required transition density.

The Sync Command is a special case of the Command frame and contains a data field that is used to generate the minimum number of transitions required to keep the derived clock circuitry locked onto the data stream. The 10th bit lane carries a transition pattern instead of CRC information. The 14-bit remainder of the 22-bit CRC is set to zero in the next frame. All DIMMs respond to a Sync command with a Status frame located in the time slot where the first data from a DRAM read command to the last DIMM would be returned plus the number of frames specified in the status delay field SD [1:0]. The additional delay field specified by the SD [1:0] field is not allowed to change the order in which data is returned on the channel. The AMB does not support out-of-order return of response data. If a Sync command is issued before any other command that returns data, the data returned for the other command must be scheduled after the status data. Care must be taken when using the SD [1:0] field to ensure that the transition density tClkTrain is not violated.

The maximum interval between sync frames is 42 frames, in order to maintain clock recovery synchronization. The host controller must adhere to a minimum interval between sync frames to guarantee that the AMB clock recovery circuits will be adjusted. This allows the AMB to save power by switching off internal circuits between sync commands. The AMB contains a register in which the host controller programs the minimum interval between syncs

which it will send. The host controller may then send syncs at any interval between the programmed interval and 42. For example, if the host controller design can send syncs in the range of 38 to 42 frames apart, the register would be programmed to 38. The best power management for the AMB can be achieved by the host controller being as consistent as possible in its sync generation. Power Management within the AMB can have an impact on bandwidth capabilities in some platforms. The AMB specification provides information on the programming of this register as well as the default and minimum values. Following a reset, the host may ignore the minimum sync interval up until the 4th sync. The AMBs should disable any power management which times sync frames until the 4th sync is received.

The AMBs must accept and process the sync command and respond properly with a status frame regardless of when it is sent. The host controller may send additional syncs that violate the minimum sync interval. The interpolators retraining may not occur, however, and this additional sync will be ignored by the sync interval timers in the AMB. Another sync must still be sent after the minimum sync interval has expired, and before 42 frames from the previous properly timed sync.

The IER command bit directs all AMBs to ignore all commands and CRC errors until after the next reset. The host controller may send a sync with the IER command bit set to avoid spurious CRC errors which may be detected by the AMBs due to the channel going to electrical idle for the fast reset. The host should not send any commands other than NOP after the sync with the IER bit set until the next fast reset.

The ERC command bit directs all of the AMBs to transition from the L0 state to the Recalibrate state for 32 to 42 frames as specified in the Recalibrate_Duration register. During the intervening frames between Sync commands, all data on the channel will be ignored and the TX and Rx circuits in the interface may be recalibrated. Following the Status frame on the NB link, the subsequent frames may contain arbitrary data and the TX and Rx circuits in the interface may be recalibrated.

The EL0s command bit directs all of the AMBs to transition from the L0 state to the L0s state for 32 to 42 frames as specified in the L0s_Duration register. During the intervening frames between Sync commands, all data on the channel will be ignored and the interface may be placed in Electrical Idle to reduce power. Following the Status frame on the NB link, the subsequent NB frames may contain arbitrary data or Electrical Idle to reduce power. A capability bit in the AMB indicates whether the AMB supports this feature and the host must not transition the channel into the L0s state unless all of the AMBs on the channel support the feature.

The R[1:0] field specifies which configuration register in the range of 43h:40h is accessed for status information. The lower 4 bits from one of these configuration registers contain status information that may be useful to a hardware memory controller and are returned in the Status frame. The status bits from these registers are returned from all AMBs simultaneously in the same Status frame. The definitions of the configuration registers are located in the *FBD AMB Specification* Register chapter. Brief descriptions of the status bits are located in the **Section 4.3.5**.

Table 4-44 — Sync Command Frame Format

Transfer	Bit									
	9	8	7	6	5	4	3	2	1	0
0	aE0	aE7	aE8	F0=0	0	OP2=0	SD1	0	EL0s	R0
1	aE1	aE6	aE9	F1=0	0	OP3=0	0	0	ERC	R1
2	aE2	aE5	aE10	aE13	0	0	OP0=1	0	IER	0
3	aE3	aE4	aE11	aE12	0	0	OP1=0	SD0	0	0
4	1	0	1	0	1	0	1	0	1	0
5	0	1	0	1	0	1	0	1	0	1
6	1	0	1	0	1	0	1	0	1	0
7	0	1	0	1	0	1	0	1	0	1
8	1	0	1	0	1	0	1	0	1	0
9	0	1	0	1	0	1	0	1	0	1
10	1	0	1	0	1	0	1	0	1	0
11	0	1	0	1	0	1	0	1	0	1

4.2.5.4 NOP Frame

The NOP frame contains three NOP commands and is sent on the southbound link when there are no other commands to send on the channel. The frame is a normal Command frame format. The 14-bit CRC contains the remainder of the 22-bit CRC from the previous frame. Unused bits in the 72-bit data payload containing the “B” and “C” NOP commands must be ignored by the AMBs but are included in the 22-bit (or 10-bit in fail-over mode) CRC calculation. The unused bits may contain debug information in some platforms. If the unused bits are filled with 0’s the data payload CRC fields compute to zeros resulting in NOP frames completely filled with zeros after the first NOP frame.

Table 4-45 — NOP Frame Format

Transfer	Bit									
	9	8	7	6	5	4	3	2	1	0
0	aE0	aE7	aE8	F0=0	0	aOP2=0	0	0	0	0
1	aE1	aE6	aE9	F1=0	0	aOP3=0	0	0	0	0
2	aE2	aE5	aE10	aE13	0	0	aOP0=0	0	0	0
3	aE3	aE4	aE11	aE12	0	0	aOP1=0	0	0	0
4	FE21	0	0	0	0	bOP2=0	0	0	0	0
5	FE20	0	0	0	0	bOP3=0	0	0	0	0
6	FE19	0	0	0	0	0	bOP0=0	0	0	0
7	FE18	0	0	0	0	0	bOP1=0	0	0	0
8	FE17	0	0	0	0	cOP2=0	0	0	0	0
9	FE16	0	0	0	0	cOP3=0	0	0	0	0
10	FE15	0	0	0	0	c	COP0=0	0	0	0
11	FE14	0	0	0	0	0	cOP1=0	0	0	0

4.3 Northbound Frames

In the northbound direction, read return data and status information are multiplexed over the northbound data path. FBD supports four basic northbound frame types: Northbound Idle, Northbound Alert, Northbound Data and Northbound Status.

4.3.1 Northbound CRC Modes

FBD supports three northbound CRC modes to support applications that require different levels of error detection. The frames contain two 72-bit or 64-bit data payloads. Each data payload is protected by either a 12-bit CRC or a 6-bit CRC. The three supported northbound CRC modes are:

14 bit lanes: 12-bit CRC over 72-bit data payload, fail-over to 6-bit CRC

13 bit lanes: 6-bit CRC over 72-bit data payload, fail-over to ECC coverage only

12 bit lanes: 6-bit CRC over 64-bit data payload, no fail-over

The selection on the mode of operation is controlled by the host and communicated during the initialization process as defined in the Initialization chapter. Northbound CRC is only computed for Data frames. The Idle, Alert, and Status frame types drive the upper bit lanes with a known data pattern. During fail-over the host simply ignores the missing bit lanes.

4.3.2 Northbound Idle Frame

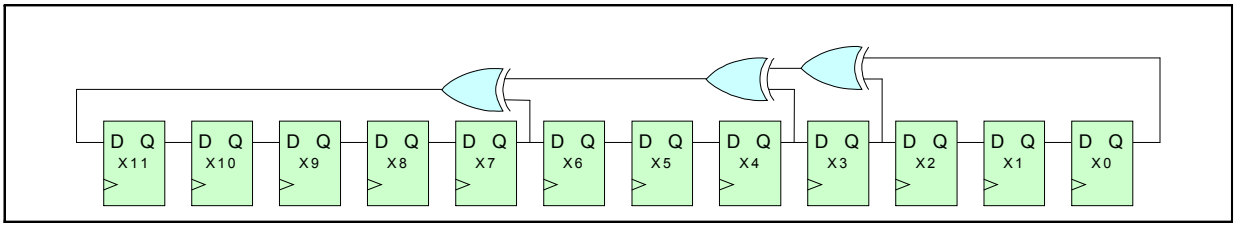
In addition to DRAM data frames, FBD supports several other northbound frame formats. Table 4-46 defines the general format of the Northbound Idle frame and contains the initial idle data pattern.

Each Idle frame contains a permuting data pattern. The last DIMM on the channel sends this permuting data pattern when not sending requested data from the DIMM. The content of the frame is designed to intentionally generate CRC errors if not in fail-over mode so that the host can easily detect when an expected Northbound Data frame with good CRC is missing. The host does not log the CRC errors generated by the Idle frames.

Host hardware will issue a Sync command on the channel immediately following entry into the L0 state to reset the permuting data pattern of the Idle and Alert frames to their initial value. The first Idle frame follows immediately after the Status frame returned for the first Sync command.

The permuting data pattern is generated by a 12-bit linear-feedback shift register (LFSR) with a polynomial of $x^{12} + x^7 + x^4 + x^3 + 1$. An example of an implementation of the counter is shown in Figure 4-14 below. The initial value of the counter is 12'b000000000001. The LFSR counter cycles through 2^{12}-1 states (4095 frames) before the pattern is repeated. Each bit of the counter is mapped onto a corresponding northbound bit lane; counter bit x0 is mapped onto bit lane 0, counter bit x1 is mapped onto bit lane 1, etc. The 13th bit lane (if implemented) contains the value of the x0 counter bit for the first 6 UI followed by the inverted x0 counter bit for the second 6 UI. The 14th bit lane (if implemented) contains the value of the x0 counter bit. The LFSR does not generate an all zero data payload.

Figure 4-14 — Idle Frame LFSR Counter Example



The permuting data pattern is sent in the frame as an added level of error detection. Table 4-46 through Table 4-49 define the format of the first four Northbound Idle frames. When in the fail-over mode that does not have CRC coverage the host may use the deterministic permuting data pattern to differentiate between a valid read data frame

and an Idle frame. This may be accomplished by detecting that the read data frame contents matches the expected Idle frame contents. To make sure that the data frame contains valid data and not an erroneous Idle frame, the host may re-read the data to verify that the data no longer matches the Idle frame permuting data pattern, or request a status response to check for errors. The re-read of the data must be scheduled to not coincide with the same frame content from the previous read to make sure that the data pattern will not match again.

Table 4-46 — First Northbound Idle Frame Format

Transfer	Bit													
	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	0	0	0	0	0	0	0	0	0	0	0	1
1	1	1	0	0	0	0	0	0	0	0	0	0	0	1
2	1	1	0	0	0	0	0	0	0	0	0	0	0	1
3	1	1	0	0	0	0	0	0	0	0	0	0	0	1
4	1	1	0	0	0	0	0	0	0	0	0	0	0	1
5	1	1	0	0	0	0	0	0	0	0	0	0	0	1
6	1	0	0	0	0	0	0	0	0	0	0	0	0	1
7	1	0	0	0	0	0	0	0	0	0	0	0	0	1
8	1	0	0	0	0	0	0	0	0	0	0	0	0	1
9	1	0	0	0	0	0	0	0	0	0	0	0	0	1
10	1	0	0	0	0	0	0	0	0	0	0	0	0	1
11	1	0	0	0	0	0	0	0	0	0	0	0	0	1

Table 4-47 — Second Northbound Idle Frame Format

Transfer	Bit													
	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0	0	0	0	0	0	0
3	0	0	1	0	0	0	0	0	0	0	0	0	0	0
4	0	0	1	0	0	0	0	0	0	0	0	0	0	0
5	0	0	1	0	0	0	0	0	0	0	0	0	0	0
6	0	1	1	0	0	0	0	0	0	0	0	0	0	0
7	0	1	1	0	0	0	0	0	0	0	0	0	0	0
8	0	1	1	0	0	0	0	0	0	0	0	0	0	0
9	0	1	1	0	0	0	0	0	0	0	0	0	0	0
10	0	1	1	0	0	0	0	0	0	0	0	0	0	0
11	0	1	1	0	0	0	0	0	0	0	0	0	0	0

Table 4-48 — Third Northbound Idle Frame Format

Transfer	Bit													
	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	1	0	0	0	0	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0	0	0	0	0	0
5	0	0	0	1	0	0	0	0	0	0	0	0	0	0
6	0	1	0	1	0	0	0	0	0	0	0	0	0	0
7	0	1	0	1	0	0	0	0	0	0	0	0	0	0
8	0	1	0	1	0	0	0	0	0	0	0	0	0	0
9	0	1	0	1	0	0	0	0	0	0	0	0	0	0
10	0	1	0	1	0	0	0	0	0	0	0	0	0	0
11	0	1	0	1	0	0	0	0	0	0	0	0	0	0

Table 4-49 — Fourth Northbound Idle Frame Format

Transfer	Bit													
	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	1	0	0	0	0	0	0	0	0	0
3	0	0	0	0	1	0	0	0	0	0	0	0	0	0
4	0	0	0	0	1	0	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0	0	0	0	0	0	0
6	0	1	0	0	1	0	0	0	0	0	0	0	0	0
7	0	1	0	0	1	0	0	0	0	0	0	0	0	0
8	0	1	0	0	1	0	0	0	0	0	0	0	0	0
9	0	1	0	0	1	0	0	0	0	0	0	0	0	0
10	0	1	0	0	1	0	0	0	0	0	0	0	0	0
11	0	1	0	0	1	0	0	0	0	0	0	0	0	0

4.3.3 Northbound Alert Frame

AMBs report detection of errors on the channel using the Northbound Alert frame. The Northbound Alert frame contains the inverse of the Idle frame data pattern. The host *may* use detection of this permuting data pattern to indicate that an error has occurred. An AMB on the channel will send this permuting data pattern after it has detected a CRC error in any southbound command frame. The AMB will continue to generate Northbound Alert frames until it receives a Soft Channel Reset command or a Fast Reset sequence. The AMB will set the Alert_Asserted status bit

in its status register to indicate that it has generated Alert frames. The Alert_Asserted status bit is also reported in subsequent Status frames to assist the host in determining which AMB generated the Alert frames. The Alert_Asserted status bit is sticky and will remain asserted until it has been cleared by a configuration write command.

The content of the frame is designed to intentionally generate CRC errors if not in fail-over mode to assist the host in detecting the difference between a Northbound Alert frame and a Northbound Data frame that has data content that matches the alert data pattern.

The permuting alert data pattern is sent in the frame as an added level of error detection. When in fail-over modes that lose all CRC coverage the host *may* use the deterministic permuting data pattern to differentiate between a valid read data frame and an Alert frame. This may be accomplished by detecting that the read data frame contents matches the unexpected Alert frame contents. To make sure that the data frame contained valid data and not an Alert frame, the host *may* schedule a hole in the northbound data stream to check that a valid Idle frame is present, or request a status response to check for errors, or re-read the data and verify that the data no longer matches the Alert frame permuting data pattern.

AMBs and repeaters may return the Alert frame to the host while the LAI buffers do not.

When an AMB generates alert frames, the other AMBs on the channel will not be aware of the alert and will continue to operate normally. Downstream AMBs (further from the host) will not see the alert pattern generated by the upstream AMB. Upstream AMBs (closer to the host) will pass the alert pattern northward as they would any data, but will not decode the data, and will not be aware that it is an alert.

The host is likely to have sent additional commands following the command which has caused the alert. The AMBs not generating an Alert will continue to process these commands, and return data on the northbound link for DRAM and config reads. If the AMB is downstream of the AMB generating the Alert, the read data will be replaced by the alert frame. If the AMB is upstream of the AMB generating the Alert, the alert frame will be replaced by the read data. This means that the host may be delayed in receiving the alert until a frame occurs in which no upstream AMB is returning data. If the host is using the lack of an alert frame at a specific time as an indication that an operation has completed successfully, it should not generate a command to another AMB that would return data during that frame.

A special case occurs with the status frame when one or more AMBs are generating alerts. Each lane of data is supplied by a different AMB for a status frame, with the host receiving merged data from up to 12 AMBs. Any AMB generating an Alert frame will replace all lanes with the Alert pattern rather than generating and passing on status. Any AMB that is not generating an Alert replaces one specific lane with its status information. This will result in the host receiving a combination of the status and alert frames on a lane by lane basis.

In all cases the host will receive the full alert frame if the frame would have otherwise been an idle frame.

The alert frame is a bit for bit inversion of what the idle frame pattern would have been for the frame. This includes inverting lanes 12 and 13 throughout the frame. The table below shows an alert frame in place of the Second Northbound Idle Frame shown above in **Table 4-47**. The sequence would be an idle frame from **Table 4-46** followed by an alert frame in **Table 4-50**.

Table 4-50 — Alert Frame Replacing Second Idle Frame

Transfer	Bit													
	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	0	1	1	1	1	1	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
2	1	1	0	1	1	1	1	1	1	1	1	1	1	1
3	1	1	0	1	1	1	1	1	1	1	1	1	1	1
4	1	1	0	1	1	1	1	1	1	1	1	1	1	1
5	1	1	0	1	1	1	1	1	1	1	1	1	1	1
6	1	0	0	1	1	1	1	1	1	1	1	1	1	1
7	1	0	0	1	1	1	1	1	1	1	1	1	1	1
8	1	0	0	1	1	1	1	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	1	1	1	1	1	1
10	1	0	0	1	1	1	1	1	1	1	1	1	1	1
11	1	0	0	1	1	1	1	1	1	1	1	1	1	1

4.3.4 Northbound Data Frames

This section defines the format of the Northbound Data frames. Each frame contains either two 72-bit data payloads or two 64-bit data payloads. A CRC code is computed across each of the 72-bit data payloads and is sent on the 12th, 13th, or 13th & 14th bit lanes if not in fail-over mode. Each data payload has its own CRC code to minimize the latency to deliver the first data payload to the host.

4.3.4.1 14-bit Lane Northbound Data Frame

This is the highest RAS mode of operation for the northbound channel. In this mode a 12-bit CRC is delivered in the E[11:0] field on the 13th & 14th northbound bit lanes during the transfer of each 72-bit data payload. Table 4-51 defines the bit positions of the CRC code in the northbound frame. This table also defines the mapping of the data from each of eighteen 4-bit DRAM devices into the Northbound Data frame. This frame format also supports 8-bit DRAM devices and the mapping of those devices is defined in Table 4-52. In the Cn.Dn labels, Cn is the DRAM chip number and Dn is the DRAM data bit number.

Table 4-51 — 14-bit Northbound Data Frame Format (with 4-bit Devices)

Xfr	Bit													
	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	E1.0	E1.11	C17.D2	C16.D0	C14.D2	C13.D0	C11.D2	C10.D0	C8.D2	C7.D0	C5.D2	C4.D0	C2.D2	C1.D0
1	E1.1	E1.10	C17.D3	C16.D1	C14.D3	C13.D1	C11.D3	C10.D1	C8.D3	C7.D1	C5.D3	C4.D1	C2.D3	C1.D1
2	E1.2	E1.9	C18.D0	C16.D2	C15.D0	C13.D2	C12.D0	C10.D2	C9.D0	C7.D2	C6.D0	C4.D2	C3.D0	C1.D2
3	E1.3	E1.8	C18.D1	C16.D3	C15.D1	C13.D3	C12.D1	C10.D3	C9.D1	C7.D3	C6.D1	C4.D3	C3.D1	C1.D3
4	E1.4	E1.7	C18.D2	C17.D0	C15.D2	C14.D0	C12.D2	C11.D0	C9.D2	C8.D0	C6.D2	C5.D0	C3.D2	C2.D0
5	E1.5	E1.6	C18.D3	C17.D1	C15.D3	C14.D1	C12.D3	C11.D1	C9.D3	C8.D1	C6.D3	C5.D1	C3.D3	C2.D1
6	E2.0	E2.11	C17.D2	C16.D0	C14.D2	C13.D0	C11.D2	C10.D0	C8.D2	C7.D0	C5.D2	C4.D0	C2.D2	C1.D0
7	E2.1	E2.10	C17.D3	C16.D1	C14.D3	C13.D1	C11.D3	C10.D1	C8.D3	C7.D1	C5.D3	C4.D1	C2.D3	C1.D1
8	E2.2	E2.9	C18.D0	C16.D2	C15.D0	C13.D2	C12.D0	C10.D2	C9.D0	C7.D2	C6.D0	C4.D2	C3.D0	C1.D2
9	E2.3	E2.8	C18.D1	C16.D3	C15.D1	C13.D3	C12.D1	C10.D3	C9.D1	C7.D3	C6.D1	C4.D3	C3.D1	C1.D3
10	E2.4	E2.7	C18.D2	C17.D0	C15.D2	C14.D0	C12.D2	C11.D0	C9.D2	C8.D0	C6.D2	C5.D0	C3.D2	C2.D0
11	E2.5	E2.6	C18.D3	C17.D1	C15.D3	C14.D1	C12.D3	C11.D1	C9.D3	C8.D1	C6.D3	C5.D1	C3.D3	C2.D1

4.3.4.2 14-bit Lane Fail-over Northbound Data Frame

When the 14 lane mode has failed over to 13 lanes, the northbound data frame is identical to the 13 bit lane frame below.

4.3.4.3 13-bit Lane Northbound Data Frame

This is the medium RAS mode of operation for the northbound channel. In this mode a 6-bit CRC is delivered in the E[5:0] field on the 13th northbound bit lane during the transfer of each 72-bit data payload. Table 4-52 defines the bit positions of the CRC code in the northbound frame. This table also defines the mapping of the data pins of nine 8-bit DRAM devices to the frame data bits. This frame format also supports 4-bit DRAM devices mapped as shown in Table 4-51 above.

Table 4-52 — 13-bit Northbound Data Frame Format (with 8-bit Devices)

xfr	Bit													
	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0		E1.0	C9.D2	C8.D4	C7.D6	C7.D0	C6.D2	C5.D4	C4.D6	C4.D0	C3.D2	C2.D4	C1.D6	C1.D0
1		E1.1	C9.D3	C8.D5	C7.D7	C7.D1	C6.D3	C5.D5	C4.D7	C4.D1	C3.D3	C2.D5	C1.D7	C1.D1
2		E1.2	C9.D4	C8.D6	C8.D0	C7.D2	C6.D4	C5.D6	C5.D0	C4.D2	C3.D4	C2.D6	C2.D0	C1.D2
3		E1.3	C9.D5	C8.D7	C8.D1	C7.D3	C6.D5	C5.D7	C5.D1	C4.D3	C3.D5	C2.D7	C2.D1	C1.D3
4		E1.4	C9.D6	C9.D0	C8.D2	C7.D4	C6.D6	C6.D0	C5.D2	C4.D4	C3.D6	C3.D0	C2.D2	C1.D4
5		E1.5	C9.D7	C9.D1	C8.D3	C7.D5	C6.D7	C6.D1	C5.D3	C4.D5	C3.D7	C3.D1	C2.D3	C1.D5
6		E2.0	C9.D2	C8.D4	C7.D6	C7.D0	C6.D2	C5.D4	C4.D6	C4.D0	C3.D2	C2.D4	C1.D6	C1.D0
7		E2.1	C9.D3	C8.D5	C7.D7	C7.D1	C6.D3	C5.D5	C4.D7	C4.D1	C3.D3	C2.D5	C1.D7	C1.D1
8		E2.2	C9.D4	C8.D6	C8.D0	C7.D2	C6.D4	C5.D6	C5.D0	C4.D2	C3.D4	C2.D6	C2.D0	C1.D2
9		E2.3	C9.D5	C8.D7	C8.D1	C7.D3	C6.D5	C5.D7	C5.D1	C4.D3	C3.D5	C2.D7	C2.D1	C1.D3
10		E2.4	C9.D6	C9.D0	C8.D2	C7.D4	C6.D6	C6.D0	C5.D2	C4.D4	C3.D6	C3.D0	C2.D2	C1.D4
11		E2.5	C9.D7	C9.D1	C8.D3	C7.D5	C6.D7	C6.D1	C5.D3	C4.D5	C3.D7	C3.D1	C2.D3	C1.D5

4.3.4.4 13-bit Lane Fail-Over Northbound Data Frame

When 13-bit lane mode has failed over and is operating on 12 lanes, each transfer consists of only the 72 bit payload with no CRC. The ECC implemented by the host is the only error detection available. Note that this frame format is NOT the same as the 12-bit Lane frame format.

Table 4-53 — 13-bit Northbound Fail-Over Data Frame Format (8 bit Devices)

xfr	Bit													
	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0			C9.D2	C8.D4	C7.D6	C7.D0	C6.D2	C5.D4	C4.D6	C4.D0	C3.D2	C2.D4	C1.D6	C1.D0
1			C9.D3	C8.D5	C7.D7	C7.D1	C6.D3	C5.D5	C4.D7	C4.D1	C3.D3	C2.D5	C1.D7	C1.D1
2			C9.D4	C8.D6	C8.D0	C7.D2	C6.D4	C5.D6	C5.D0	C4.D2	C3.D4	C2.D6	C2.D0	C1.D2
3			C9.D5	C8.D7	C8.D1	C7.D3	C6.D5	C5.D7	C5.D1	C4.D3	C3.D5	C2.D7	C2.D1	C1.D3
4			C9.D6	C9.D0	C8.D2	C7.D4	C6.D6	C6.D0	C5.D2	C4.D4	C3.D6	C3.D0	C2.D2	C1.D4
5			C9.D7	C9.D1	C8.D3	C7.D5	C6.D7	C6.D1	C5.D3	C4.D5	C3.D7	C3.D1	C2.D3	C1.D5
6			C9.D2	C8.D4	C7.D6	C7.D0	C6.D2	C5.D4	C4.D6	C4.D0	C3.D2	C2.D4	C1.D6	C1.D0
7			C9.D3	C8.D5	C7.D7	C7.D1	C6.D3	C5.D5	C4.D7	C4.D1	C3.D3	C2.D5	C1.D7	C1.D1
8			C9.D4	C8.D6	C8.D0	C7.D2	C6.D4	C5.D6	C5.D0	C4.D2	C3.D4	C2.D6	C2.D0	C1.D2
9			C9.D5	C8.D7	C8.D1	C7.D3	C6.D5	C5.D7	C5.D1	C4.D3	C3.D5	C2.D7	C2.D1	C1.D3
10			C9.D6	C9.D0	C8.D2	C7.D4	C6.D6	C6.D0	C5.D2	C4.D4	C3.D6	C3.D0	C2.D2	C1.D4
11			C9.D7	C9.D1	C8.D3	C7.D5	C6.D7	C6.D1	C5.D3	C4.D5	C3.D7	C3.D1	C2.D3	C1.D5

4.3.4.5 12-bit Lane Northbound Data Frame (Non-ECC Mode)

This is the lowest RAS mode of operation for the northbound channel. In this mode a 6-bit CRC is delivered in the E[5:0] field on the 12th northbound bit lane during the transfer of each 64-bit data payload. The data payload does not contain ECC bits and the CRC code is the only form of protection. Table 4-54 defines the bit positions of the CRC code in the northbound frame. This table also defines the mapping of the data pins of four 16-bit DRAM devices to the frame data bits. This frame format also supports 8-bit DRAM devices mapping two 8-bit devices onto the upper and lower 8 bits of the 16-bit DRAM mapping. This frame format also supports 4-bit DRAM devices mapping four 4-bit devices onto the four groups of continuous 4 bits of the 16-bit DRAM mapping.

Table 4-54 — 12-bit Northbound Data Frame Format (with Non-ECC 16-bit Devices)

Xfr	Bit													
	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0			E1.0	C4.D12	C4.D6	C4.D0	C3.D10	C3.D4	C2.D14	C2.D8	C2.D2	C1.D12	C1.D6	C1.D0
1			E1.1	C4.D13	C4.D7	C4.D1	C3.D11	C3.D5	C2.D15	C2.D9	C2.D3	C1.D13	C1.D7	C1.D1
2			E1.2	C4.D14	C4.D8	C4.D2	C3.D12	C3.D6	C3.D0	C2.D10	C2.D4	C1.D14	C1.D8	C1.D2
3			E1.3	C4.D15	C4.D9	C4.D3	C3.D13	C3.D7	C3.D1	C2.D11	C2.D5	C1.D15	C1.D9	C1.D3
4			E1.4	0	C4.D10	C4.D4	C3.D14	C3.D8	C3.D2	C2.D12	C2.D6	C2.D0	C1.D10	C1.D4
5			E1.5	0	C4.D11	C4.D5	C3.D15	C3.D9	C3.D3	C2.D13	C2.D7	C2.D1	C1.D11	C1.D5
6			E2.0	C4.D12	C4.D6	C4.D0	C3.D10	C3.D4	C2.D14	C2.D8	C2.D2	C1.D12	C1.D6	C1.D0
7			E2.1	C4.D13	C4.D7	C4.D1	C3.D11	C3.D5	C2.D15	C2.D9	C2.D3	C1.D13	C1.D7	C1.D1
8			E2.2	C4.D14	C4.D8	C4.D2	C3.D12	C3.D6	C3.D0	C2.D10	C2.D4	C1.D14	C1.D8	C1.D2
9			E2.3	C4.D15	C4.D9	C4.D3	C3.D13	C3.D7	C3.D1	C2.D11	C2.D5	C1.D15	C1.D9	C1.D3
10			E2.4	0	C4.D10	C4.D4	C3.D14	C3.D8	C3.D2	C2.D12	C2.D6	C2.D0	C1.D10	C1.D4
11			E2.5	0	C4.D11	C4.D5	C3.D15	C3.D9	C3.D3	C2.D13	C2.D7	C2.D1	C1.D11	C1.D5

4.3.4.6 Northbound Register Data Frame

The NB Register Data frame is used to return data in response to a Read Configuration command. The frame always returns 32-bits of register data. The host must select the appropriate bytes from the four data bytes delivered if fewer than four bytes are needed. Table 4-55 defines the bit positions of the register data in the northbound frame. The table shows the CRC code coverage when operation with 14 bit lanes. The CRC code coverage for 13-bit and 12-bit lane operation is the same as those defined in the 13-bit and 12-bit frame formats above.

Table 4-55 — Northbound Register Data Frame Format

Xfr	Bit													
	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	E1.0	E1.11	0	0	0	0	0	0	D30	D24	D18	D12	D6	D0
1	E1.1	E1.10	0	0	0	0	0	0	D31	D25	D19	D13	D7	D1
2	E1.2	E1.9	0	0	0	0	0	0	0	D26	D20	D14	D8	D2
3	E1.3	E1.8	0	0	0	0	0	0	0	D27	D21	D15	D9	D3
4	E1.4	E1.7	0	0	0	0	0	0	0	D28	D22	D16	D10	D4
5	E1.5	E1.6	0	0	0	0	0	0	0	D29	D23	D17	D11	D5
6	E2.0	E2.11	0	0	0	0	0	0	0	0	0	0	0	0
7	E2.1	E2.10	0	0	0	0	0	0	0	0	0	0	0	0
8	E2.2	E2.9	0	0	0	0	0	0	0	0	0	0	0	0
9	E2.3	E2.8	0	0	0	0	0	0	0	0	0	0	0	0
10	E2.4	E2.7	0	0	0	0	0	0	0	0	0	0	0	0
11	E2.5	E2.6	0	0	0	0	0	0	0	0	0	0	0	0

4.3.5 Northbound Status Frame

The Northbound Status frame defined in Table 4-56 is used to convey status information from each AMB to the host. The Status frame is returned to the host in response to a Sync command from the host. The status returned in the Status frame corresponds to the status of the AMB to commands before the Sync command. Errors that are generated by commands after the Sync command are reported in subsequent Status frames. In other words the Sync command provides a fence for status reporting. The Status frame is returned in the time slot specified by the Command_to_Data parameter plus the value in the SD [1:0] (Status Delay) field. Each AMB will merge its status into the northbound bit stream on the appropriate bit lane. The Command_to_Data value consists of the CMD2DATA register plug AL+CL.

Table 4-56 — Northbound Status Frame Format

Xfr	Bit													
	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	DBS0	DAS0	D9S0	D8S0	D7S0	D6S0	D5S0	D4S0	D3S0	D2S0	D1S0	D0S0
1	0	0	DBS1	DAS1	D9S1	D8S1	D7S1	D6S1	D5S1	D4S1	D3S1	D2S1	D1S1	D0S1
2	0	0	DBS2	DAS2	D9S2	D8S2	D7S2	D6S2	D5S2	D4S2	D3S2	D2S2	D1S2	D0S2
3	0	0	DBS3	DAS3	D9S3	D8S3	D7S3	D6S3	D5S3	D4S3	D3S3	D2S3	D1S3	D0S3
4	0	0	DBSP	DASP	D9SP	D8SP	D7SP	D6SP	D5SP	D4SP	D3SP	D2SP	D1SP	D0SP
5	0	1	0	1	0	1	0	1	0	1	0	1	0	1
6	1	0	1	0	1	0	1	0	1	0	1	0	1	0
7	0	1	0	1	0	1	0	1	0	1	0	1	0	1
8	1	0	1	0	1	0	1	0	1	0	1	0	1	0
9	0	1	0	1	0	1	0	1	0	1	0	1	0	1
10	1	0	1	0	1	0	1	0	1	0	1	0	1	0
11	0	1	0	1	0	1	0	1	0	1	0	1	0	1

The northbound Status frame contains a group of status bits from each AMB. The status bits are protected by an odd parity bit DnSP that covers the status bits from each AMB individually. This is necessary because the status from each AMB is merged "on-the-fly" into the Status Frame by each AMB and a CRC that covers all of the status bits could not be calculated within this mechanism. Each AMB drives all 12 bits delivered in the frame for its assigned bit lane, including the alternating one/zero pattern. The status bits returned by the AMBs are described in Table 4-57. There are four possible sets of status bits as selected by the R [1:0] field in the Sync command. For a complete definition of the status bits and how they are used refer to the *FBD AMB Specification* Register chapter.

See **Section 4.3.3** for the interaction between the Status Frame and Alert Frame

Table 4-57 — Status Bit Description

Field	Name	Description
FBD Status 0, R [1:0] = 00		
SP	Parity	Parity: This bit contains an odd parity bit that covers the S [3:0] field.
S3	NBDE	Northbound Debug Event (1 = asserted, 0 = inactive): This bit is used to communicate debug events to the host.
S2:S1	Thermal_Trip	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.
S0	Alert_Asserted	Alert_Asserted: This bit indicates an error has been detected by the AMB. This bit is reset by a Write Configuration command after either a Soft Channel Reset command or a Fast Reset sequence has reset the channel.
FBD Status 1, R [1:0] = 01		
SP	Parity	Parity: This bit contains an odd parity bit that covers the S [3:0] field.
S3:S1	Reserved	Reserved.
S0	Data_Merge_Error	Data Merge Error: This bit indicates that the northbound data merge alignment logic of an intermediate AMB cannot meet the timing required to merge its DRAM data into the northbound data stream when required. Refer to the Initialization chapter for details.
FBD Status 2, R [1:0] = 10		
SP	Parity	Parity: This bit contains an odd parity bit that covers the S [3:0] field.
S3:S0	Reserved	Reserved.
FBD Status 3, R [1:0] = 11		
SP	Parity	Parity: This bit contains an odd parity bit that covers the S [3:0] field.
S3:S0	Reserved	Reserved.

The AMB in the last DIMM position of the daisy chain initiates the northbound Status frame and fills the bit lane corresponding to its DIMM position with its status information and fills the remainder of the bit lanes with a zero status code and an invalid zero parity value. This is done so that the host may detect a missing status response if an AMB misinterprets the Sync command. The host is expected (but not required) to detect the status response error and reissue the Sync command to request the status again. The CRC bit lanes are filled with the same fixed pattern because the CRC is not valid in this frame type.

4.4 DRAM Memory Timing

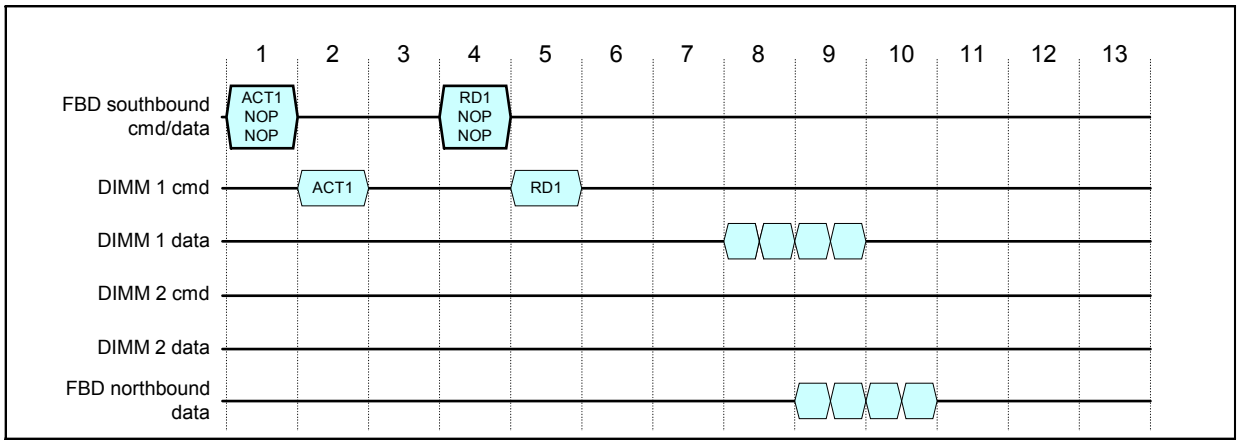
The host accesses the DRAM devices on an FBD DIMM as if they were directly connected to the host but with a few differences. First there is generally a longer than usual delay in the return data path between the DRAM and the host, and second there is a FIFO mechanism in the write data path between the host and the DRAM. The host sends “RAS” and “CAS” style commands directly to the DRAM devices. The commands on the FBD channel are delivered to the DRAM devices with a fixed delay. The host controller must deliver commands onto the FBD channel exactly as the host intends the commands to be delivered to the DRAM devices. This section illustrates the DRAM timing on

the FBD channel. The command delivery on the DRAM address and control pins use 1n command timing. 1n command timing means that the commands are present on the DRAM pins for a single clock cycle. This allows the commands present on the FBD channel to be forwarded to the DRAM channel without timing modification.

4.4.1 Read Timing

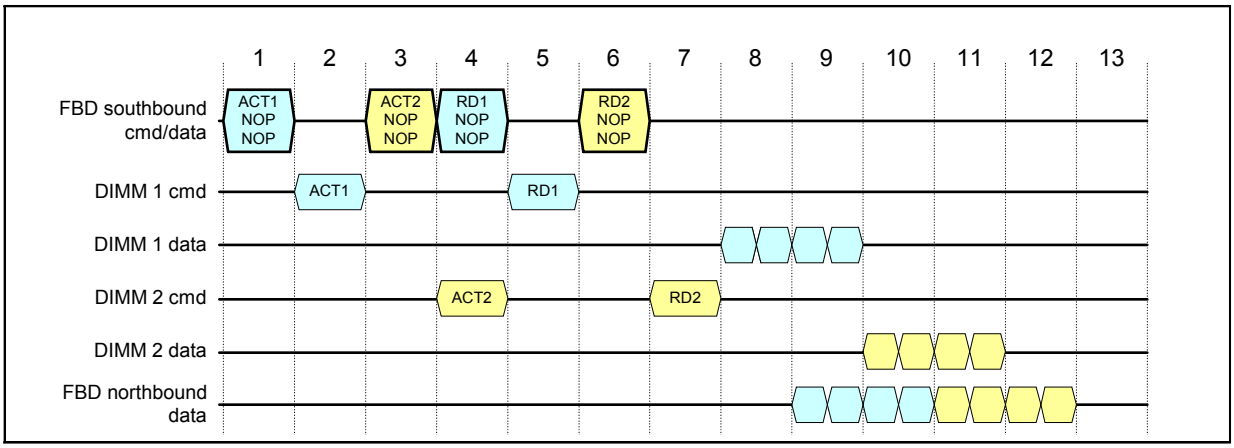
The command timing of the DRAM devices on an FBD DIMM is identical to the timing of an individual DRAM device. The RAS latency, CAS latency, etc. are controlled by the MRS values loaded into the DRAM devices. Figure 4-15 illustrates a DRAM Read operation. The data returned to the host is delayed for an interval of time determined by the propagation delay characteristics of the channel. For single DIMM configurations the timing behaves similar to a Registered SDRAM DIMM. As DIMMs are added to the channel the accumulated delay due to PCB flight time and delay through intermediate AMB components increases the delay in the return data path.

Figure 4-15 — Basic DRAM Read Data Transfers on FBD (RD)



Back-to-back reads from different DIMMs is illustrated in Figure 4-16. Unlike DDR2, the data from the separate DIMMs can be returned without a dead clock between the data bursts.

Figure 4-16 — FBack-to-back DRAM Read Data Transfers on FBD (RD-RD)

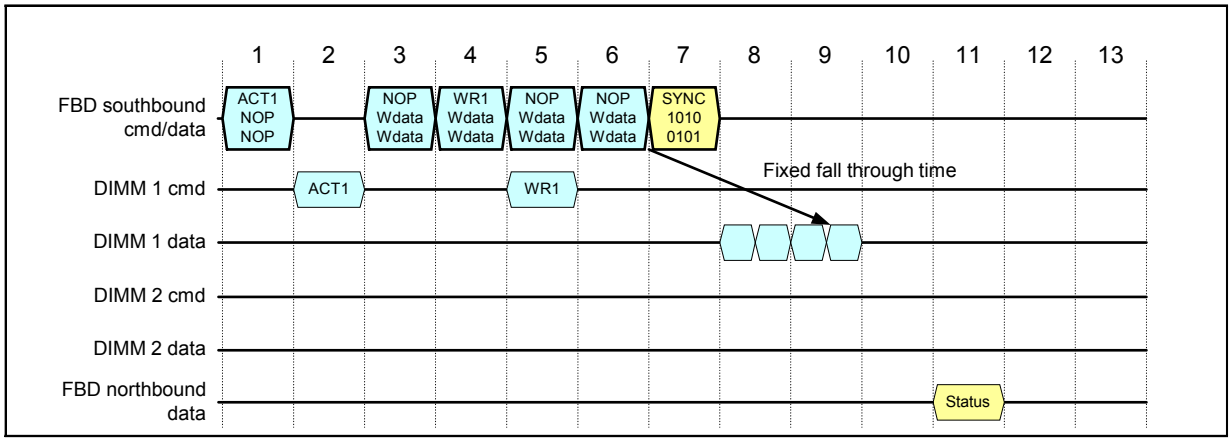


4.4.2 Write Timing

The write command timing of a DRAM device on an FBD DIMM is identical to the timing of an individual DRAM device. The Write latency is controlled by the MRS values loaded into the DRAM devices. Figure 4-17 illustrates an example DRAM Write operation. The host transfers the data to be written into a write FIFO in the AMB preceding the DRAM write transfer. The write FIFO is used to accumulate write data in the AMB so that the data can be transferred to the DRAM devices at full burst rate during the write operation. The host must be aware of the DRAM

Write latency value in order to make sure that the write data is available in the Write FIFO early enough to be delivered to the DRAMs when expected. The AMB must be aware of the DRAM Write latency value in order to deliver the write data to the DRAMs when expected

Figure 4-17 — Basic DRAM Write Data Transfers on FBD (WR)



Note that the Write command may be issued before the frame holding the last payload of data. The figure shows the shortest time between the last frame of data is driven on the southbound channel and when the data can be driven onto the DRAM data pins. The data can be loaded into the FIFO earlier than what is shown but will occupy an entry in the FIFO until used. The fixed fall through time shown defines the just-in-time arrival of the data to meet delivery to the DRAM. This just-in-time arrival time allows the controller to deliver a burst of 64 transfers to the DRAM using the 35 deep FIFO in the AMB. See Figure 4-20 for a timing diagram of the 64-transfer write operation. Writes *may* be followed by a Sync command that returns status information to indicate to the host that no errors are associated with the write operation(s). The figure shows the earliest the Sync command can be issued and report completion of the write operations. If there are errors with the write command or the write data the AMB will report the error by sending Alert frames.

Following error detection, the host *may* issue the Soft Channel Reset command to discard any data in the write FIFO. This would empty the write FIFO and put the write FIFO state machines into a known state.

4.4.2.1 Write Data FIFO

The Write Data FIFO is a data structure that is used to accumulate write data in the AMB in preparation for bursting the data to the DRAM devices. The FIFO can be filled at a maximum of half of the DRAM burst rate but is emptied at the full DRAM burst data rate. The Command+Wdata frames contain a data payload of 72-bits that is loaded into the designated write FIFO. The Command+Wdata frames are not required to be contiguous and may be separated by an arbitrary number of intervening frames. The write FIFO on each DIMM can hold thirty-five (35) 72-bit data payloads. Multiple bursts of data can be accumulated in the FIFO to amortize the read-write-read DRAM data bus turnaround penalty over a number of write operations.

The WS [2:0] bits identifying the target DIMM are delivered in four consecutive Command+Wdata frames. Each DIMM must speculatively capture the Wdata into an accumulation buffer before writing the Wdata into the write FIFO. Once the target DIMM is identified the data may be written into the target DIMM write FIFO. If the WS [2:0] bits identify a different target DIMM the data is discarded. Wdata is pushed into the FIFO in groups of 4 72-bit data payloads at a time whether the DRAM is operating in a burst length of 4 or 8.

The DRAM Write command pulls the data from the head of the FIFO and delivers it to the DRAM devices in the clock cycle determined by register settings in the AMB. Additional data can be loaded into the FIFO while data is being delivered to the DRAM. The depth of the FIFO supports a continuous burst of 64 transfers to the DRAM devices. There are 3 rules that define the behavior of the FIFO as follows:

Rule 1: Minimum fall through time

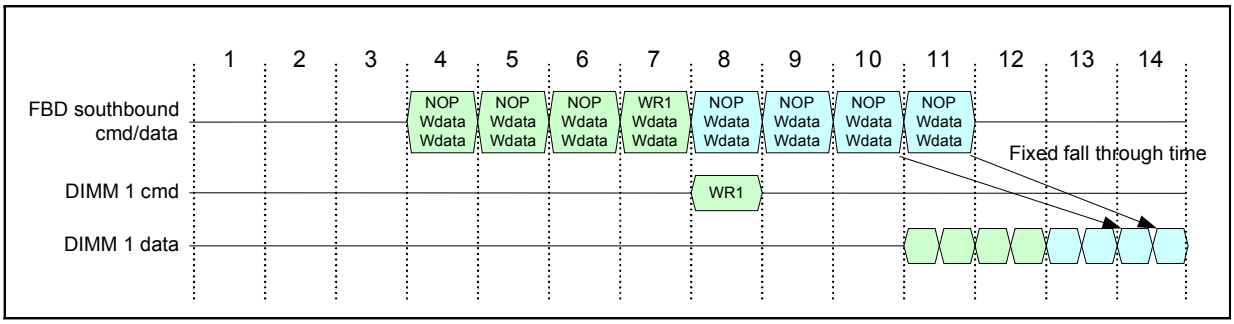
For even chunks (1st chunk is 0): 3 frame times from end of frame with even chunk until even chunk driven on DIMM data bus. The even chunk fall through is actually derived from the odd chunk fall through.

For odd chunks (2nd chunk is 1): 2.5 frame times from end of frame with odd chunk until odd chunk driven on DIMM data bus

All chunks are delayed by at least the minimum fall through time – and only the last 2 chunks to be drained can attain this minimum.

Figure 4-18 illustrates Rule 1 for a burst length of 8 transfer. The minimum fall through time is defined to enable a host controller to optimize the usage of the FIFO entries and allow a DRAM burst of 64 transfers.

Figure 4-18 — Write FIFO Minimum Fall Through Time



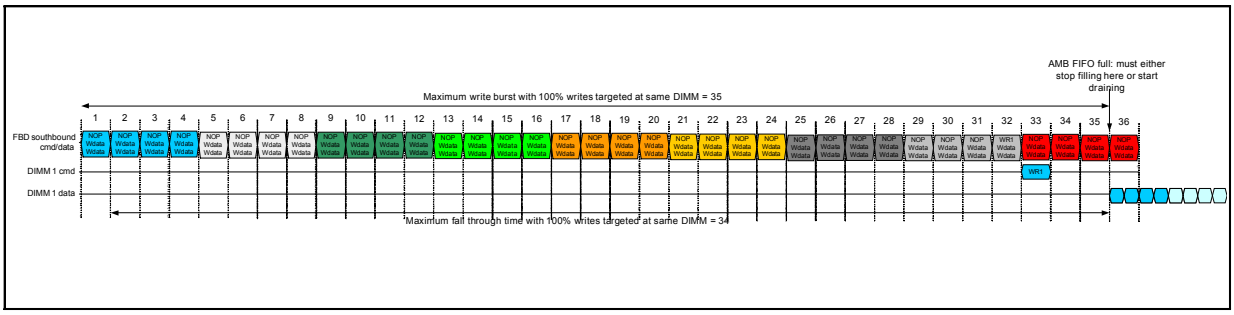
Rule 2: Maximum fall through time

Maximum number of write data frames targeted at the same DIMM before it must start draining writes from the DIMM AMB is 35. This means that the maximum fall through for 100% write data frames targeted at the same DIMM is 34 frame times as illustrated in **Figure 4-19**. Beyond this the fifo would overflow if additional data was sent to this DIMM.

Note: To perform the accumulation of four write data frames for destination decode the AMB may actually have a buffer that is effectively larger than 35 frames. The host shall not exploit this implementation feature.

Figure 4-19 illustrates Rule 2. The figure shows the maximum number of write data frames that may be pushed into the FIFO before they must be drained without causing a FIFO overrun.

Figure 4-19 — Write FIFO Maximum Fall Through Time



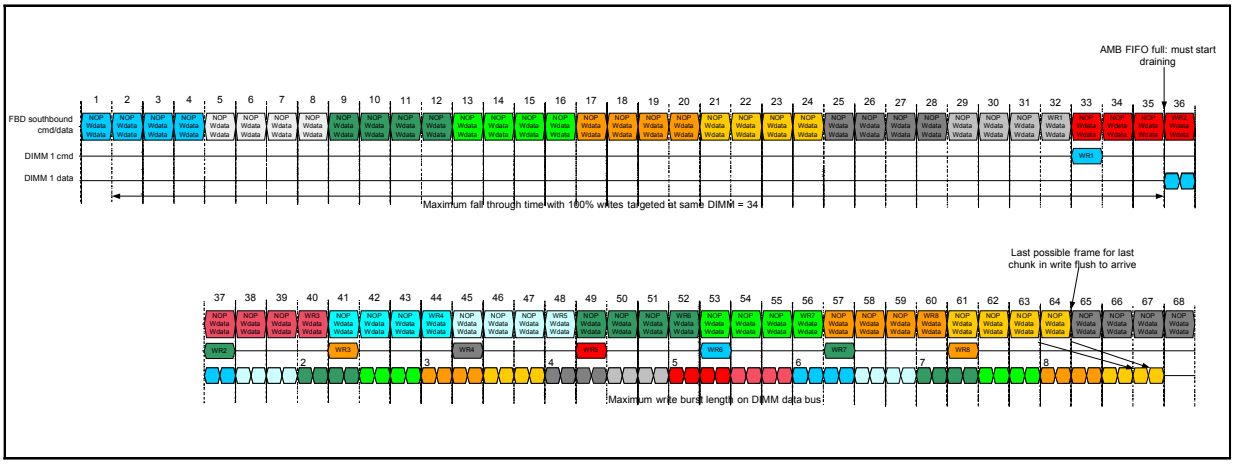
Rule 3: Maximum pipelining of writes for a single DIMM

Maximum contiguous write burst length on DIMM data bus is 32 clocks (64 transfers).

NOTE: This rule is a consequence of rules 1 & 2.

Figure 4-20 illustrates Rule 3. The figure shows the longest continuous burst (64 transfers) supported by the depth of the FIFO. The example shows the use of burst of 8 commands but burst of 4 commands could be used if supported by the DRAM devices.

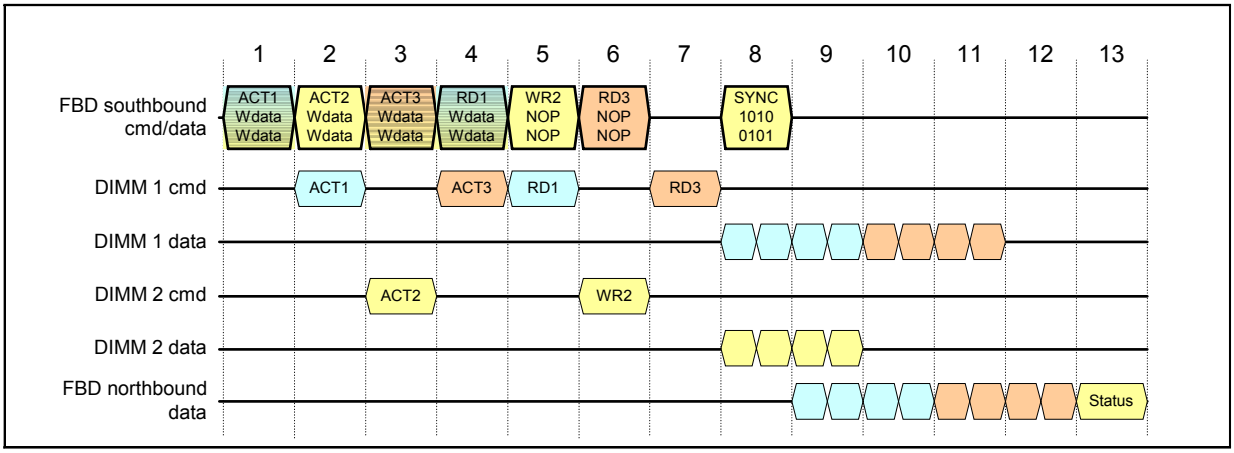
Figure 4-20 — Write FIFO Maximum Pipelining of Writes for a Single DIMM



4.4.3 Simultaneous Read and Write Data Transfers

The FBD channel provides separate data path for read completion data and write request data. Because each FBD DIMM contains an isolated DRAM channel behind the AMB component, read data from the DRAM devices on one FBD DIMM can be read at the same time that write data is being written to the DRAM devices on another FBD DIMM. Figure 4-21 shows a read transaction to one DIMM followed by a write transaction to a different DIMM, followed by a read transaction to the first DIMM.

Figure 4-21 — Simultaneous Read and Write Data Transfers on FBD (RD-WR-RD)



4.4.4 DRAM Bus Segment Restrictions

Either one or two ranks of DRAM devices may be located behind the AMB on an FBD DIMM. These devices sit on a DRAM bus segment and must observe the restrictions on the usage of the bus segment. The JEDEC SDRAM data sheets should be referenced for details of the restrictions. A dead time is required between read operations for DDR2 devices from the two separate ranks to avoid electrical conflict on the DQS and DQS# signals. The turnaround times for dead times such as read-to-read, read-to-write, and write-to-read are DIMM layout specific and are captured in the SPD EEPROM on the DIMM. These parameters are readable by firmware to direct the appropriate behavior of the host controller.

5 Reliability, Availability and Serviceability

5.1 Overview

The FBD channel specification provides comprehensive RAS support including, error detection and frame transmission retry, error logging, error injection, host add/remove of DIMMs, and the mechanisms for in-operation test and fault recovery using the Fast Reset capability of the channel. The philosophy for FBD channel reliability is to provide strong error detection of channel transaction errors, and the ability to retry the transactions after automatic hardware recovery. Both the northbound and southbound links include fail-over mechanisms that can keep the links running after any one wire fails with enough fault detection to maintain reliable operation until repair.

The FBD channel protects data from errors using CRC codes generated by both the host and the AMB. FBD provides error detection and retry mechanisms for commands and data. It further provides an Alert frame reporting mechanism whereby the host is made aware of errors found by an AMB in the command or (write) data. A Status response mechanism is provided to return to the host quick abbreviated status information from all of the AMBs simultaneously. An AMB will discard any commands or data received with a CRC error.

For reads, the read data that is returned to the host with correct ECC and/or CRC is the positive acknowledgement that all has transpired without error. If the host does not receive a read return when scheduled, or if the read return contains an error, the host *may* reissue the read command or the entire read sequence, and/or send a sync command to acquire error status from the AMBs.[1]

Error free writes are silently accepted by the AMB with no response returned to the host. Write data or any commands that are received by the AMB in error will cause the AMB to notify the host through Alert frames. Alert frames are continuously sent until acknowledged by the host with a Soft Channel Reset command or a Fast Reset sequence.

5.2 Example Error Flows

This section gives an informal overview of error handling by walking through example write and read flows. Precise details follow in subsequent sections.

5.2.1 Command Error Flow

The AMB checks for errors in all commands but cannot discriminate one failed type of command from any other type of command. All command errors are reported to the host and all subsequent commands except Soft Channel Reset are ignored. Command errors are reported to the host by a stream of Alert frames in place of normally returned frames.

Upon receiving an Alert response indicating that there was a command error, the host *may* issue a Soft Channel Reset command or a Fast Reset to attempt to recover from the error. The AMB will close all DRAM pages and place the DRAM devices into self-refresh upon detection of the Fast Reset. Following the Fast Reset the host *may* reissue all read and write transactions since the previous verified transaction completion and continue normal operation.

5.2.2 Write Data Error Flow

The AMB checks for errors in the write data by computing a 22-bit CRC covering the write data frame. When in wire fail-over mode a 10-bit CRC is available to check for link transmission errors in the write data. CRC errors detected in the write data are reported to the host the same as command errors.

1. Some read data errors might be corrected using an ECC code. The choice between retry and correction is left up to the designers of the individual host implementations.

5.2.3 Read Error Flow

A read differs from a write primarily in that the AMB provides a positive acknowledgement that there were no errors with the read command through the delivery of the read data in the specified northbound data frame. If the AMB detects an error in a read command, the AMB discards the command and Alert frames will be returned**.**

Upon receiving a read return, the host verifies that it has received the correct amount of data at the scheduled time, and checks the correctness of the data. If any of these are in error the read command *may* either be retried or the host may attempt to correct the [data] error. **Section 5.7** describes the algorithm in more detail.

5.3 Overview of Error Protection, Detection, Correction, and Logging

FBD uses several different mechanisms for error protection, detection, correction, and error logging. Error handling elements are made up of the following:

a) Southbound Commands and Data

   a. The host computes check bits for commands

      i. 14-bit CRC on a per command basis. Reduced to 10-bit CRC in fail-over mode.

   b. The host computes check bits for (write) data

      i. 22-bit CRC on a per 72-bit write burst basis. Reduced to 10-bit CRC in fail-over mode.

   c. The AMB detects CRC errors in southbound commands or (write) data, and logs information on the errors detected

      i. Command or write data errors once observed prevent the AMB from decoding any commands except the Soft Channel Reset command or until the channel is reset by the host

         - The AMB does not evaluate any ECC information sent with the (DRAM) write or attempt to correct any errors

   d. The AMB returns error status on detected errors

      i. CRC errors are reported on the northbound link by inserting Alert frames in place of other content.

      ii. Alerts continue to be sent until a Soft Channel Reset command is received or the channel is reset.

b) Northbound Read Data

   FBD supports three northbound CRC modes to support applications that require different levels of error detection and cost. The frames contain two 72-bit or 64-bit data payloads. Each data payload is protected by either a 12-bit CRC or a 6-bit CRC, with reduced protection during fail-over. The three supported northbound CRC modes are:

   14 bit lanes: 12-bit CRC over 72-bit data payload, fail-over to 6-bit CRC

   13 bit lanes: 6-bit CRC over 72-bit data payload, fail-over to ECC coverage only

   12 bit lanes: 6-bit CRC over 64-bit data payload, no fail-over

   The selection on the mode of operation is controlled by the host and communicated during the initialization process as defined in the Initialization chapter. Northbound CRC is only computed for Data frames. The Idle, Alert, and Status frame types drive the upper bit lanes with a known data pattern. During fail-over the host ignores the missing bit lanes and operates with reduced CRC coverage.

   a. The host detects an error in the data through CRC (added by the AMB when not in fail-over mode) or by ECC provided with the data when read from DRAM (provided by the host with the data when written to DRAM)

      i. The host logs the information on errors detected

ii. The host corrects the data if possible using the ECC included within the data

iii. The host takes whatever other steps deemed prudent (such as reissuing the command to see if the data error was transient or scrubbing the DRAM location if it were a correctable error)

c) Northbound Status

a. The AMB computes a parity bit over its own status information

b. The host detects an error in the northbound status return

i. The host logs the information on errors detected

ii. The host takes whatever other steps deemed prudent (such as issuing another sync command – up to a limit)

iii. If there were no errors in the status return itself then the host would log any error information reported through the status return and take whatever other steps deemed prudent.

Table 5-58 — AMB Responses to Error Conditions

Error Condition		AMB Response
Southbound Command	CRC Error	AMB discards the commands and reports the error with Alert frames. Subsequent commands are ignored until a Soft Channel Reset command is received or the channel is reset.
	tClkTrain violation	AMB reports the error with Alert frames. The DRAM devices are put into Self Refresh. Subsequent commands are ignored until a Soft Channel Reset command is received or the channel is reset.

Table 5-59 — Host Responses to Error Conditions

Error Condition		AMB Response
Northbound Status Response	Status Parity Error	The host must ignore the status returned and *may* retry the request for another status response.
	Frame CRC (non fail-over mode)	Frame CRC is not valid on northbound status responses.
Northbound Read Data Response	System Level ECC	The host *may* attempt to correct the data error using the ECC code. If the data is uncorrectable the host must discard the read data and *may* retry the read request. If the error persists the host *may* perform a Fast Reset of the channel and retry the read request.
	Data Frame = Alert Frame	The host *may* discard the read data and *may* retry the read to see if the Data Frame really was equal to the Alert Frame or *may* perform a status read to check that there is no error condition or *may* schedule an Idle frame and check to see if the Idle frame has been replaced with an Alert frame.
	Data Frame = Idle Frame	The host *may* discard the read data and *may* retry the read to see if the Data Frame really was equal to the Idle Frame.
	Frame CRC (non fail-over mode)	The host must discard the read data and *may* retry the read request. If the error persists the host *may* perform a Fast Reset of the channel and retry the read request.

As noted above, the host is the only agent that corrects errors in system data. However, to provide enhanced data integrity, the host *may* first retry a read request upon detecting a data error before attempting to correct the error.

Furthermore, the host *may* choose to patrol memory, reading memory locations and writing back corrected data for any errors detected. Such patrol "scrubbing" is orthogonal to the FBD error handling specification. It is left to the host designers to determine their own memory scrub methodology.

The error logging done by the AMB(s) and the host are designed to permit isolation of the error source.

5.4 Error Protection and Detection Methods

The goal of an error detection technique is to enable the receiver of a message transmitted through a noisy error-introducing channel to determine whether the message has been corrupted. To do this, the transmitter constructs a value called a checksum that is a function of the message, and appends it to the message. The receiver can then use the same function to calculate the checksum of the received message and compare it with the appended checksum to see if the message was correctly received.

The basic idea of the CRC algorithm is simply to treat the message as an enormous binary number, to divide it by another fixed binary number, and to make the remainder from this division the checksum. Upon receipt of the message, the receiver can perform the same division and compare the remainder with the transmitted remainder. FBD also uses a compound checksum in order to get strong error detection with very low latency for the southbound command execution. The compound checksum on the southbound channel is constructed from the exclusive-or between the "A" command checksum of that frame, and the data checksum from the previous frame. Combining the checksum from the data in the previous frame allows the controller to quickly use a command that can be CRC'ed without having to wait for all of the data in that frame to be received. Since writes are cached in the AMB, there is no performance penalty. An error in the compound checksum requires that both the command in the current frame and the data in the previous frame be treated as faulty.

There are two ways to manage Error Control:

Backward Error Control – enough information is sent to allow the receiver to detect errors, but not correct them. Upon error detection, retransmission may be requested. The simple topology and low latency of FBD makes retransmission easy when the CRC detects a fault, and this is the principle method of error control.

Forward Error Control – enough additional or redundant information is passed to the receiver, so it can not only detect, but also correct errors. This requires substantially more redundant information to be sent than required just for error detection. The DRAM error correction codes with "chip fail" used to protect against soft errors and DRAM failures also do a good job of detecting and correcting link errors. High reliability frame formats combine this DRAM centric ECC with the packet CRC in a complementary manner to get the best total fault coverage, and provide a secondary method of error control. Although it will be a rare occurrence to have the link transmission generate an ECC fault without also generating a CRC fault, memory controller designers should understand this possibility exists.

5.4.1 CRC Logic Used on Normal Southbound Frames

Normal southbound frames consist of 12 transfers of data delivered on the 10 southbound bit lanes. Table 5-60 defines the common features of a normal southbound frame. Each frame starts with four transfers containing what is referred to as the "A" command. The F[1:0] field in the "A" command determines the frame type. The remainder of the "A" command includes 24 bits of command information in the aC[23:0] field and a 14-bit compound CRC checksum in the aE[13:0] field. The aE[13:0] field provides error detection coverage across the F[1:0], aC[23:0] and aE[13:0] fields. Because the frame type field is covered by the CRC of the "A" command, the content of the entire frame is invalid if there is a CRC error detected in the "A" command. Each southbound frame also contains 72-bits which can be used for additional commands, command extensions, or data, and is protected by a 22-bit compound CRC checksum identified as FE[21:0]. Table 5-60 shows a normal southbound frame with 72-bits of write data mapped onto 4-bit memory devices. The Bxx notation in the Tables refers to bit inputs to the specific CRC generation logic block used.

To generate each southbound frame, a 22-bit CRC is generated from the 72-bit data B[71:0], and a 14-bit CRC is generated from the 26-bit "A" command F[1:0]aC[23:0]. Eight bits of the 22-bit data CRC, FE[21:14], are located in the 10th bit lane of the current southbound frame. The remaining 14 bits of the 22-bit data CRC, FE[13:0] as illustrated on the bottom of Table 5-60, are latched at the controller to be combined with the 14-bit CRC generated

from the 26-bit "A" command in the next southbound frame, using an exclusive-or function to create the compound checksum aE[13:0] which will be transmitted in the next southbound frame. The initial value of the 14-bit remainder in the first Sync command during channel initialization is zero.

Table 5-60 — Common Features of Normal Southbound Frames

Xfr	Bit									
	9	8	7	6	5	4	3	2	1	0
0	AE0	aE7	aE8	F0 B24	aC20 B23	aC16 B16	aC12 B15	aC8 B8	aC4 B7	aC0 B0
1	AE1	aE6	aE9	F1 B25	aC21 B22	aC17 B17	aC13 B14	aC9 B9	aC5 B6	aC1 B1
2	AE2	aE5	aE10	aE13	aC22 B21	aC18 B17	aC14 B13	aC10 B10	aC6 B5	aC2 B2
3	AE3	aE4	aE11	aE12	aC23 B20	aC19 B19	aC15 B12	aC11 B11	aC7 B4	aC3 B3
4	FE21	B8	B7	B6	B5	B4	B3	B2	B1	B0
5	FE20	B9	B10	B11	B12	B13	B14	B15	B16	B17
6	FE19	B26	B25	B24	B23	B22	B21	B20	B19	B18
7	FE18	B27	B28	B29	B30	B31	B32	B33	B34	B35
8	FE17	B44	B43	B42	B41	B40	B39	B38	B37	B36
9	FE16	B45	B46	B47	B48	B49	B50	B51	B52	B53
10	FE15	B62	B61	B60	B59	B58	B57	B56	B55	B54
11	FE14	B63	B64	B65	B66	B67	B68	B69	B70	B71
	FE0	FE7	FE8							
	FE1	FE6	FE9							
	FE2	FE5	FE10	FE13						
	FE3	FE4	FE11	FE12						

The generation of the compound 14-bit checksum aE[13:0] requires an XOR operation of the "A" command 14-bit CRC from this frame, with the latched 14-bits FE[13:0] CRC generated from the 72-bit data of the previous frame. The compound checksum bit aE13 for this frame comes from "A" command CRC14[13] from this frame XOR with FE13 from the 72-bit data CRC of the previous frame, aE12 for this frame comes from "A" command CRC14[12] from this frame XOR with FE12 from the 72-bit data CRC of the previous frame, and so on. This mechanism provides strong CRC protection of the 72 bits of the frame without adding latency to the delivery of the "A" command in the current frame.

GENERATION: Normal Southbound Frame: 26-bit "A" Command + 72-bit Data:

FRAME_0 = Initialization: Generate 22-bit CRC FE[21:0] from 72-bit data payload and latch FE[13:0].

FRAME_1: Generate 14-bit checksum CRC14[13:0] from 26-bit command, and 22-bit checksum CRC22[21:0] from 72-bit data. Generate the compound checksum aE[13:0] through bitwise XOR of Data FE[13:0] latched from FRAME_0 with Command CRC14[13:0] from FRAME_1. FE[13] XOR CRC14[13], FE[12] XOR CRC14[12], etc… Insert aE[13:0] and FE[21:14] into FRAME_1 with command and data and SEND Southbound FRAME_1.

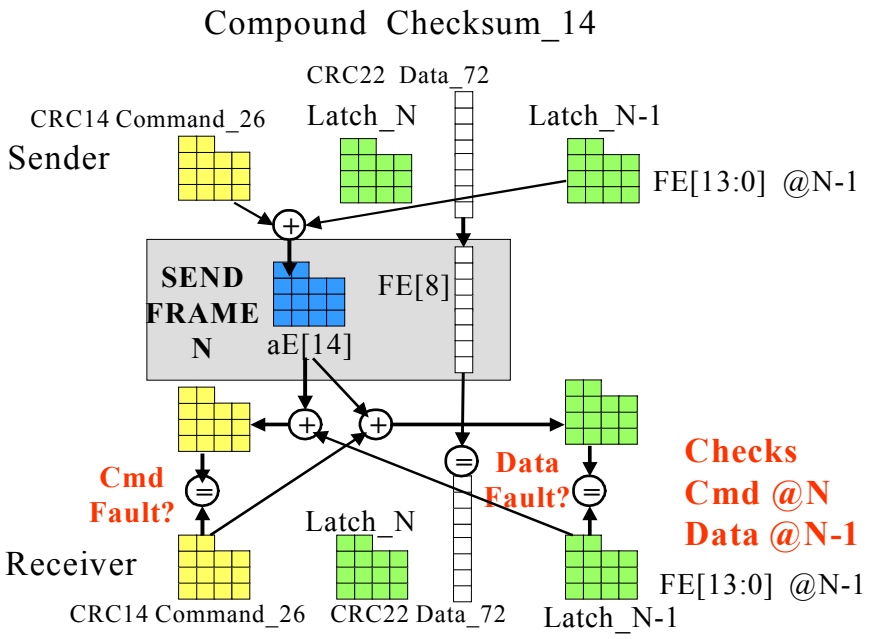
To decode the southbound frame, every AMB on the link must generate the 22-bit CRC FE[21:0] for every southbound frame, whether that frame is addressed to that specific AMB or not, and latch FE[13:0] for possible XOR with the next frame's composite checksum aE[13:0]. When a normal southbound frame is addressed to a specific AMB chip, both the 14-bit CRC from the 26-bit "A" command, and the 22-bit CRC covering the 72 bit data are generated. The receiver checks the command from FRAME_N, and completes the check of the data from FRAME_N-1.

DECODE: Normal Southbound Frame: 26-bit "A" Command + 72-bit Data:

FRAME_0 = Initialize: Generate 22-bit CRC FE[21:0] for 72-bit data payload and latch FE[13:0].

FRAME_1: Generate 14-bit checksum CRC14[13:0] from 26-bit command, and 22-bit checksum CRC22[21:0] from 72-bit data in FRAME_1. Latch CRC22[13:0] as FE[13:0] for future compound checksum checks in FRAME_2. Generate the test compound checksum TESTaE[13:0] through bitwise XOR of FE[13:0] from FRAME_0 with new aE[13:0] from FRAME_1. FE[13] XOR aE[13], FE[12] XOR aE[12], etc… If the generated test compound checksum TESTaE[13:0] matches the transmitted compound checksum aE[13:0], there are no errors in the "A" command of FRAME_1. To complete the detection of faults in the 72-bits of data from FRAME_0, bitwise XOR new CRC14[13:0] from 26-bit command with transmitted FRAME_1 aE[13:0] and compare this result to the latched FRAME_0 FE[13:0]. To start the fault detection of the 72-bits of data transferred in FRAME_1, compare transmitted FRAME_1 FE[21:14] with new CRC22[21:14] generated from the 72-bit data in FRAME_1. The completion of fault detection for the 72-bits of data transferred in FRAME_1 will be done in FRAME_2.

A fault in aE[13:0] indicates that both the "A" command in FRAME_1 could be faulted, and that the 72-bit data in FRAME_0 could be faulted. A comparison fault in transmitted FE [21:14] partial checksum indicates that the 72-bit data in FRAME_0 could be faulted.



The CRC of the "A" command can be checked as soon as the first 4 transfers of the frame are received and the "A" command can be used immediately without waiting for the remainder of the frame to arrive. The complexity of this scheme is relatively low compared to the benefit of minimizing DRAM access time. XOR'ing two CRCs of reasonable length results in only a small reduction in fault detection for each, but it does increase the number of detected faults, and hence retransmissions.

Table 5-61 defines the mapping of the data going to each of eighteen 4-bit DRAM devices for a write operation into the southbound data frame. This frame format also supports 8-bit DRAM devices as illustrated in Table 1-5 and uses the same CRC logic block input positions as in the frame in Table 5-60. In the Cn.Dn labels, Cn is the DRAM chip number and Dn is the DRAM data bit number. In these tables B[71:0] in the data payload correspond to the bit position used by the inputs to the 22-bit CRC checksum generation logic block for the (write) data, and B[25:0] in the "A" command payload correspond to the bit position used by the inputs to the 14-bit CRC checksum generation logic block for any of the commands.

Table 5-61 — Southbound Frame with 72 bit Write Data (4-bit Devices)

Xfer	Bit									
	9	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0	aE0	aE7	aE8	F0 B24	aC20 B23	aC16 B16	aC12 B15	aC8 B8	aC4 B7	aC0 B0
1	aE1	aE6	aE9	F1 B25	aC21 B22	aC17 B17	aC13 B14	aC9 B9	aC5 B6	aC1 B1
2	aE2	aE5	aE10	aE13	aC22 B21	aC18 B18	aC14 B13	aC10 B10	aC6 B5	aC2 B2
3	aE3	aE4	aE11	aE12	aC23 B20	aC19 B19	aC15 B12	aC11 B11	aC7 B4	aC3 B3
4	FE21	C17.D0 B8	C15.D0 B7	C13.D0 B6	C11.D0 B5	C9.D0 B4	C7.D0 B3	C5.D0 B2	C3.D0 B1	C1.D0 B0
5	FE20	C17.D1 B9	C15.D1 B10	C13.D1 B11	C11.D1 B12	C9.D1 B13	C7.D1 B14	C5.D1 B15	C3.D1 B16	C1.D1 B17
6	FE19	C17.D2 B26	C15.D2 B25	C13.D2 B24	C11.D2 B23	C9.D2 B22	C7.D2 B21	C5.D2 B20	C3.D2 B19	C1.D2 B18
7	FE18	C17.D3 B27	C15.D3 B28	C13.D3 B29	C11.D3 B30	C9.D3 B31	C7.D3 B32	C5.D3 B33	C3.D3 B34	C1.D3 B35
8	FE17	C18.D0 B44	C16.D0 B43	C14.D0 B42	C12.D0 B41	C10.D0 B40	C8.D0 B39	C6.D0 B38	C4.D0 B37	C2.D0 B36
9	FE16	C18.D1 B45	C16.D1 B46	C14.D1 B47	C12.D1 B48	C10.D1 B49	C8.D1 B50	C6.D1 B51	C4.D1 B52	C2.D1 B53
10	FE15	C18.D2 B62	C16.D2 B61	C14.D2 B60	C12.D2 B59	C10.D2 B58	C8.D2 B57	C6.D2 B56	C4.D2 B55	C2.D2 B54
11	FE14	C18.D3 B63	C16.D3 B64	C14.D3 B65	C12.D3 B66	C10.D3 B67	C8.D3 B68	C6.D3 B69	C4.D3 B70	C2.D3 B71
	FE0	FE7	FE8							
	FE1	FE6	FE9							
	FE2	FE5	FE10	FE13						
	FE3	FE4	FE11	FE12						

5.4.1.1 Normal Southbound Frame CRCs

The 14-bit CRC uses the polynomial P(x) = x^14+x^13+x^12+x^11+x^9+x^8+x^6+x^3+x^2+x+1. Which in the octal base is 75517, and in the communication standard hexadecimal base polynomial naming convention that drops the degree bit from the most significant position is 0x3b4f. This polynomial is primitive, has linearly independent roots, and the roots of the reciprocal polynomial are also linearly independent. This polynomial has an odd number of terms, and hence will checksum random bit faults with an even or odd number of faulty bits with approximately the same detection rate. This CRC will detect all random 1-bit, 2-bit, and 3-bit faults, and any continuous fault of 14-bits or less. The serpentine mapping onto the command frame columns provides strong detection of multi-bit faults generated along a column. Wire/column failures are considered the most common fault, and this bit mapping is optimized to detect single or adjacent pair wire faults.

14 bit CRC P(x) = x^14+x^13+x^12+x^11+x^9+x^8+x^6+x^3+x^2+x+1 over 26 bits

```
B = Bit Input.

CRC(0) = B(24) xor B(22) xor B(21) xor B(18) xor B(17) xor B(15) xor
             B(14) xor B(9) xor B(8) xor B(7) xor B(6) xor B(4) xor
             B(1) xor B(0);
CRC(1) = B(25) xor B(24) xor B(23) xor B(21) xor B(19) xor B(17) xor
             B(16) xor B(14) xor B(10) xor B(6) xor B(5) xor B(4) xor
             B(2) xor B(0) ;
CRC(2) = B(25) xor B(21) xor B(20) xor B(14) xor B(11) xor B(9) xor
             B(8) xor B(5) xor B(4) xor B(3) xor B(0);
CRC(3) = B(24) xor B(18) xor B(17) xor B(14) xor B(12) xor B(10) xor
             B(8) xor B(7) xor B(5) xor B(0);
CRC(4) = B(25) xor B(19) xor B(18) xor B(15) xor B(13) xor B(11) xor
             B(9) xor B(8) xor B(6) xor B(1);
CRC(5) = B(20) xor B(19) xor B(16) xor B(14) xor B(12) xor B(10) xor
             B(9) xor B(7) xor B(2);
CRC(6) = B(24) xor B(22) xor B(20) xor B(18) xor B(14) xor B(13) xor
             B(11) xor B(10) xor B(9) xor B(7) xor B(6) xor B(4) xor
             B(3) xor B(1) xor B(0);
CRC(7) = B(25) xor B(23) xor B(21) xor B(19) xor B(15) xor B(14) xor
             B(12) xor B(11) xor B(10) xor B(8) xor B(7) xor B(5) xor
             B(4) xor B(2) xor B(1);
CRC(8) = B(21) xor B(20) xor B(18) xor B(17) xor B(16) xor B(14) xor
             B(13) xor B(12) xor B(11) xor B(7) xor B(5) xor B(4) xor
             B(3) xor B(2) xor B(1) xor B(0);
CRC(9) = B(24) xor B(19) xor B(13) xor B(12) xor B(9) xor B(7) xor
             B(5) xor B(3) xor B(2) xor B(0);
CRC(10) = B(25) xor B(20) xor B(14) xor B(13) xor B(10) xor B(8) xor
              B(6) xor B(4) xor B(3) xor B(1);
CRC(11) = B(24) xor B(22) xor B(18) xor B(17) xor B(11) xor B(8) xor
              B(6) xor B(5) xor B(2) xor B(1) xor B(0);
CRC(12) = B(25) xor B(24) xor B(23) xor B(22) xor B(21) xor B(19) xor
              B(17) xor B(15) xor B(14) xor B(12) xor B(8) xor B(4) xor
              B(3) xor B(2) xor B(0);
CRC(13) = B(25) xor B(23) xor B(21) xor B(20) xor B(17) xor B(16) xor
              B(14) xor B(13) xor B(8) xor B(7) xor B(6) xor B(5) xor
              B(3) xor B(0);
```

The polynomial P(x) = x^22+x^14+x^13+x^12+x^7+x^3+x^2+x+1 generates the 22-bit CRC to cover the 72-bit data in the southbound packet shown in Table 5-60. Which in the octal base is 20070217, and in the communication standard hexadecimal base polynomial naming convention that drops the degree bit from the most significant position is 0x00708f. This polynomial is primitive, has linearly dependent roots, and the roots of the reciprocal polynomial are also linearly independent. This polynomial has an odd number of terms, and hence will checksum random bit faults with an even or odd number of faulty bits with approximately the same detection rate. This CRC will detect all

random 1-bit, 2-bit, and 3-bit faults, and any continuous fault of 22-bits or less. The serpentine mapping onto the data frame rows provides strong detection of multi-bit faults generated along a row, and complements the fault correction and detection capabilities of the "chip fail" ECC which is mapped down the data frame columns by the Chip.Data_bit ordering shown in Table 5-61.

22 bit CRC P(x) = x^22+x^14+x^13+x^12+x^7+x^3+x^2+x+1 over 72 bits

```
B = Bit Input.

    CRC(0) = B(70) xor B(69) xor B(66) xor B(62) xor B(61) xor B(59) xor
                 B(58) xor B(55) xor B(54) xor B(53) xor B(50) xor B(49) xor
                 B(42) xor B(39) xor B(33) xor B(32) xor B(31) xor B(29) xor
                 B(27) xor B(25) xor B(24) xor B(22) xor B(21) xor B(19) xor
                 B(18) xor B(16) xor B(15) xor B(10) xor B(9) xor B(8) xor
                 B(0);
    CRC(1) = B(71) xor B(69) xor B(67) xor B(66) xor B(63) xor B(61) xor
                 B(60) xor B(58) xor B(56) xor B(53) xor B(51) xor B(49) xor
                 B(43) xor B(42) xor B(40) xor B(39) xor B(34) xor B(31) xor
                 B(30) xor B(29) xor B(28) xor B(27) xor B(26) xor B(24) xor
                 B(23) xor B(21) xor B(20) xor B(18) xor B(17) xor B(15) xor
                 B(11) xor B(8) xor B(1) xor B(0);
    CRC(2) = B(69) xor B(68) xor B(67) xor B(66) xor B(64) xor B(58) xor
                 B(57) xor B(55) xor B(53) xor B(52) xor B(49) xor B(44) xor
                 B(43) xor B(42) xor B(41) xor B(40) xor B(39) xor B(35) xor
                 B(33) xor B(30) xor B(28) xor B(15) xor B(12) xor B(10) xor
                 B(8) xor B(2) xor B(1) xor B(0);
    CRC(3) = B(68) xor B(67) xor B(66) xor B(65) xor B(62) xor B(61) xor
                 B(56) xor B(55) xor B(49) xor B(45) xor B(44) xor B(43) xor
                 B(41) xor B(40) xor B(39) xor B(36) xor B(34) xor B(33) xor
                 B(32) xor B(27) xor B(25) xor B(24) xor B(22) xor B(21) xor
                 B(19) xor B(18) xor B(15) xor B(13) xor B(11) xor B(10) xor
                 B(8) xor B(3) xor B(2) xor B(1) xor B(0);
    CRC(4) = B(69) xor B(68) xor B(67) xor B(66) xor B(63) xor B(62) xor
                 B(57) xor B(56) xor B(50) xor B(46) xor B(45) xor B(44) xor
                 B(42) xor B(41) xor B(40) xor B(37) xor B(35) xor B(34) xor
                 B(33) xor B(28) xor B(26) xor B(25) xor B(23) xor B(22) xor
                 B(20) xor B(19) xor B(16) xor B(14) xor B(12) xor B(11) xor
                 B(9) xor B(4) xor B(3) xor B(2) xor B(1);
    CRC(5) = B(70) xor B(69) xor B(68) xor B(67) xor B(64) xor B(63) xor
                 B(58) xor B(57) xor B(51) xor B(47) xor B(46) xor B(45) xor
                 B(43) xor B(42) xor B(41) xor B(38) xor B(36) xor B(35) xor
                 B(34) xor B(29) xor B(27) xor B(26) xor B(24) xor B(23) xor
                 B(21) xor B(20) xor B(17) xor B(15) xor B(13) xor B(12) xor
                 B(10) xor B(5) xor B(4) xor B(3) xor B(2);
    CRC(6) = B(71) xor B(70) xor B(69) xor B(68) xor B(65) xor B(64) xor
                 B(59) xor B(58) xor B(52) xor B(48) xor B(47) xor B(46) xor
                 B(44) xor B(43) xor B(42) xor B(39) xor B(37) xor B(36) xor
                 B(35) xor B(30) xor B(28) xor B(27) xor B(25) xor B(24) xor
                 B(22) xor B(21) xor B(18) xor B(16) xor B(14) xor B(13) xor
                 B(11) xor B(6) xor B(5) xor B(4) xor B(3);
    CRC(7) = B(71) xor B(65) xor B(62) xor B(61) xor B(60) xor B(58) xor
                 B(55) xor B(54) xor B(50) xor B(48) xor B(47) xor B(45) xor
                 B(44) xor B(43) xor B(42) xor B(40) xor B(39) xor B(38) xor
                 B(37) xor B(36) xor B(33) xor B(32) xor B(28) xor B(27) xor
                 B(26) xor B(24) xor B(23) xor B(21) xor B(18) xor B(17) xor
                 B(16) xor B(14) xor B(12) xor B(10) xor B(9) xor B(8) xor
                 B(7) xor B(6) xor B(5) xor B(4) xor B(0);
    CRC(8) = B(66) xor B(63) xor B(62) xor B(61) xor B(59) xor B(56) xor
```

```
            B(55) xor B(51) xor B(49) xor B(48) xor B(46) xor B(45) xor
            B(44) xor B(43) xor B(41) xor B(40) xor B(39) xor B(38) xor
            B(37) xor B(34) xor B(33) xor B(29) xor B(28) xor B(27) xor
            B(25) xor B(24) xor B(22) xor B(19) xor B(18) xor B(17) xor
            B(15) xor B(13) xor B(11) xor B(10) xor B(9) xor B(8) xor
            B(7) xor B(6) xor B(5) xor B(1);
CRC(9) = B(67) xor B(64) xor B(63) xor B(62) xor B(60) xor B(57) xor
            B(56) xor B(52) xor B(50) xor B(49) xor B(47) xor B(46) xor
            B(45) xor B(44) xor B(42) xor B(41) xor B(40) xor B(39) xor
            B(38) xor B(35) xor B(34) xor B(30) xor B(29) xor B(28) xor
            B(26) xor B(25) xor B(23) xor B(20) xor B(19) xor B(18) xor
            B(16) xor B(14) xor B(12) xor B(11) xor B(10) xor B(9) xor
            B(8) xor B(7) xor B(6) xor B(2);
CRC(10) = B(68) xor B(65) xor B(64) xor B(63) xor B(61) xor B(58) xor
             B(57) xor B(53) xor B(51) xor B(50) xor B(48) xor B(47) xor
             B(46) xor B(45) xor B(43) xor B(42) xor B(41) xor B(40) xor
             B(39) xor B(36) xor B(35) xor B(31) xor B(30) xor B(29) xor
             B(27) xor B(26) xor B(24) xor B(21) xor B(20) xor B(19) xor
             B(17) xor B(15) xor B(13) xor B(12) xor B(11) xor B(10) xor
             B(9) xor B(8) xor B(7) xor B(3);
CRC(11) = B(69) xor B(66) xor B(65) xor B(64) xor B(62) xor B(59) xor
             B(58) xor B(54) xor B(52) xor B(51) xor B(49) xor B(48) xor
             B(47) xor B(46) xor B(44) xor B(43) xor B(42) xor B(41) xor
             B(40) xor B(37) xor B(36) xor B(32) xor B(31) xor B(30) xor
             B(28) xor B(27) xor B(25) xor B(22) xor B(21) xor B(20) xor
             B(18) xor B(16) xor B(14) xor B(13) xor B(12) xor B(11) xor
             B(10) xor B(9) xor B(8) xor B(4);
CRC(12) = B(69) xor B(67) xor B(65) xor B(63) xor B(62) xor B(61) xor
             B(60) xor B(58) xor B(54) xor B(52) xor B(48) xor B(47) xor
             B(45) xor B(44) xor B(43) xor B(41) xor B(39) xor B(38) xor
             B(37) xor B(28) xor B(27) xor B(26) xor B(25) xor B(24) xor
             B(23) xor B(18) xor B(17) xor B(16) xor B(14) xor B(13) xor
             B(12) xor B(11) xor B(8) xor B(5) xor B(0);
CRC(13) = B(69) xor B(68) xor B(64) xor B(63) xor B(58) xor B(54) xor
             B(50) xor B(48) xor B(46) xor B(45) xor B(44) xor B(40) xor
             B(38) xor B(33) xor B(32) xor B(31) xor B(28) xor B(26) xor
             B(22) xor B(21) xor B(17) xor B(16) xor B(14) xor B(13) xor
             B(12) xor B(10) xor B(8) xor B(6) xor B(1) xor B(0);
CRC(14) = B(66) xor B(65) xor B(64) xor B(62) xor B(61) xor B(58) xor
             B(54) xor B(53) xor B(51) xor B(50) xor B(47) xor B(46) xor
             B(45) xor B(42) xor B(41) xor B(34) xor B(31) xor B(25) xor
             B(24) xor B(23) xor B(21) xor B(19) xor B(17) xor B(16) xor
             B(14) xor B(13) xor B(11) xor B(10) xor B(8) xor B(7) xor
             B(2) xor B(1) xor B(0);
CRC(15) = B(67) xor B(66) xor B(65) xor B(63) xor B(62) xor B(59) xor
             B(55) xor B(54) xor B(52) xor B(51) xor B(48) xor B(47) xor
             B(46) xor B(43) xor B(42) xor B(35) xor B(32) xor B(26) xor
             B(25) xor B(24) xor B(22) xor B(20) xor B(18) xor B(17) xor
             B(15) xor B(14) xor B(12) xor B(11) xor B(9) xor B(8) xor
             B(3) xor B(2) xor B(1);
CRC(16) = B(68) xor B(67) xor B(66) xor B(64) xor B(63) xor B(60) xor
             B(56) xor B(55) xor B(53) xor B(52) xor B(49) xor B(48) xor
             B(47) xor B(44) xor B(43) xor B(36) xor B(33) xor B(27) xor
             B(26) xor B(25) xor B(23) xor B(21) xor B(19) xor B(18) xor
             B(16) xor B(15) xor B(13) xor B(12) xor B(10) xor B(9) xor
             B(4) xor B(3) xor B(2);
CRC(17) = B(69) xor B(68) xor B(67) xor B(65) xor B(64) xor B(61) xor
             B(57) xor B(56) xor B(54) xor B(53) xor B(50) xor B(49) xor
```

```
              B(48) xor B(45) xor B(44) xor B(37) xor B(34) xor B(28) xor
              B(27) xor B(26) xor B(24) xor B(22) xor B(20) xor B(19) xor
              B(17) xor B(16) xor B(14) xor B(13) xor B(11) xor B(10) xor
              B(5) xor B(4) xor B(3);
CRC(18) = B(70) xor B(69) xor B(68) xor B(66) xor B(65) xor B(62) xor
              B(58) xor B(57) xor B(55) xor B(54) xor B(51) xor B(50) xor
              B(49) xor B(46) xor B(45) xor B(38) xor B(35) xor B(29) xor
              B(28) xor B(27) xor B(25) xor B(23) xor B(21) xor B(20) xor
              B(18) xor B(17) xor B(15) xor B(14) xor B(12) xor B(11) xor
              B(6) xor B(5) xor B(4);
CRC(19) = B(71) xor B(70) xor B(69) xor B(67) xor B(66) xor B(63) xor
              B(59) xor B(58) xor B(56) xor B(55) xor B(52) xor B(51) xor
              B(50) xor B(47) xor B(46) xor B(39) xor B(36) xor B(30) xor
              B(29) xor B(28) xor B(26) xor B(24) xor B(22) xor B(21) xor
              B(19) xor B(18) xor B(16) xor B(15) xor B(13) xor B(12) xor
              B(7) xor B(6) xor B(5);
CRC(20) = B(71) xor B(70) xor B(68) xor B(67) xor B(64) xor B(60) xor
              B(59) xor B(57) xor B(56) xor B(53) xor B(52) xor B(51) xor
              B(48) xor B(47) xor B(40) xor B(37) xor B(31) xor B(30) xor
              B(29) xor B(27) xor B(25) xor B(23) xor B(22) xor B(20) xor
              B(19) xor B(17) xor B(16) xor B(14) xor B(13) xor B(8) xor
              B(7) xor B(6);
CRC(21) = B(71) xor B(69) xor B(68) xor B(65) xor B(61) xor B(60) xor
              B(58) xor B(57) xor B(54) xor B(53) xor B(52) xor B(49) xor
              B(48) xor B(41) xor B(38) xor B(32) xor B(31) xor B(30) xor
              B(28) xor B(26) xor B(24) xor B(23) xor B(21) xor B(20) xor
              B(18) xor B(17) xor B(15) xor B(14) xor B(9) xor B(8) xor
              B(7);
```

5.4.1.2 Normal Southbound Frame with Multiple Commands

As shown in Table 5-62, the normal southbound frame can also hold just multiple commands. One or two commands, “B” command and “C” command, can share the normal southbound frame with the mandatory “A” command. For error detection, these commands are treated just as the 72-bit data payload and use the same CRC logic block and B[71:0] input mapping.

Table 5-62 — Southbound Frame with Three Commands

Xfr	Bit									
	9	8	7	6	5	4	3	2	1	0
0	aE0	aE7	aE8	F0 B24	aC20 B23	aC16 B16	aC12 B15	aC8 B8	aC4 B7	aC0 B0
1	aE1	aE6	aE9	F1 B25	aC21 B22	aC17 B17	aC13 B14	aC9 B9	aC5 B6	aC1 B1
2	aE2	aE5	aE10	aE13	aC22 B21	aC18 B17	aC14 B13	aC10 B10	aC6 B5	aC2 B2
3	aE3	aE4	aE11	aE12	aC23 B20	aC19 B19	aC15 B12	aC11 B11	aC7 B4	aC3 B3
4	FE21	0 B8	0 B7	0 B6	bC20 B5	bC16 B4	bC12 B3	bC8 B2	bC4 B1	bC0 B0
5	FE20	0 B9	0 B10	0 B11	bC21 B12	bC17 B13	bC13 B14	bC9 B15	bC5 B16	bC1 B17
6	FE19	0 B26	0 B25	0 B24	bC22 B23	bC18 B22	bC14 B21	bC10 B20	bC6 B19	bC2 B18
7	FE18	0 B27	0 B28	0 B29	bC23 B30	bC19 B31	bC15 B32	bC11 B33	bC7 B34	bC3 B35
8	FE17	0 B44	0 B43	0 B42	cC20 B41	cC16 B40	cC12 B39	cC8 B38	cC4 B37	cC0 B36
9	FE16	0 B45	0 B46	0 B47	cC21 B48	cC17 B49	cC13 B50	cC9 B51	cC5 B52	cC1 B53
10	FE15	0 B62	0 B61	0 B60	cC22 B59	cC18 B58	cC14 B57	cC10 B56	cC6 B55	cC2 B54
11	FE14	0 B63	0 B64	0 B65	cC23 B66	cC19 B67	cC15 B68	cC11 B69	cC7 B70	cC3 B71
	FE0	FE7	FE11							
	FE1	FE6	FE10							
	FE2	FE5	FE9	FE13						
	FE3	FE4	FE8	FE12						

5.4.2 Fail-over Southbound Frames

Fail-over southbound frames consist of 12 transfers of data delivered on the 9 southbound bit lanes. Bit lane 9 is not available to carry CRC bits in fail-over mode, and the CRC code size is reduced in this mode. Table 5-63 defines the common features of a southbound frame in fail-over mode. A total of 108 bits are delivered as part of the frame, including 10 bits of CRC. The CRC code provides good error detection for the period of time until the system can be serviced and the fail-over condition repaired.

Each frame starts with four transfers containing what is referred to as the “A” command. The F[1:0] field in the “A” command determines the frame type. The remainder of the “A” command includes 24 bits of command information in the aC[23:0] field and a 10-bit compound CRC checksum in the aE[9:0] field. The aE[9:0] field provides error detection coverage across the F[1:0], aC[23:0] and aE[9:0] fields. Because the frame type field is covered by the CRC of the “A” command, the content of the entire frame is invalid if there is a CRC error detected in the “A” command. Each fail-over southbound frame also contains 72-bits which can be used for command extensions, additional commands, or data information, and is protected by a 10-bit compound CRC checksum identified as

FE[9:0] at the bottom of Table 5-63. The table shows a fail-over southbound frame with 72-bits of write data mapped onto 8-bit memory devices. The Bxx notation in the table refers to bit inputs to the specific CRC logic generation block used.

Table 5-63 — Common Features of Fail-over Southbound Frames (with 8-bit Devices)

Xfr	Bit									
	9	8	7	6	5	4	3	2	1	0
0		aE3	aE4	F0 B24	aC20 B23	aC16 B16	aC12 B15	aC8 B8	aC4 B7	aC0 B0
1		aE2	aE5	F1 B25	aC21 B22	aC17 B17	aC13 B14	aC9 B9	aC5 B6	aC1 B1
2		aE1	aE6	aE9	aC22 B21	aC18 B17	aC14 B13	aC10 B10	aC6 B5	aC2 B2
3		aE0	aE7	aE8	aC23 B20	aC19 B19	aC15 B12	aC11 B11	aC7 B4	aC3 B3
4		C9.D0 B8	C8.D0 B7	C7.D0 B6	C6.D0 B5	C5.D0 B4	C4.D0 B3	C3.D0 B2	C2.D0 B1	C1.D0 B0
5		C9.D1 B9	C8.D1 B10	C7.D1 B11	C6.D1 B12	C5.D1 B13	C4.D1 B14	C3.D1 B15	C2.D1 B16	C1.D1 B17
6		C9.D2 B26	C8.D2 B25	C7.D2 B24	C6.D2 B23	C5.D2 B22	C4.D2 B21	C3.D2 B20	C2.D2 B19	C1.D2 B18
7		C9.D3 B27	C8.D3 B28	C7.D3 B29	C6.D3 B30	C5.D3 B31	C4.D3 B32	C3.D3 B33	C2.D3 B34	C1.D3 B35
8		C9.D4 B44	C8.D4 B43	C7.D4 B42	C6.D4 B41	C5.D4 B40	C4.D4 B39	C3.D4 B38	C2.D4 B37	C1.D4 B36
9		C9.D5 B45	C8.D5 B54	C7.D5 B47	C6.D5 B48	C5.D5 B49	C4.D5 B50	C3.D5 B51	C2.D5 B52	C1.D5 B53
10		C9.D6 B62	C8.D6 B61	C7.D6 B60	C6.D6 B59	C5.D6 B58	C4.D6 B57	C3.D6 B56	C2.D6 B55	C1.D6 B54
11		C9.D7 B63	C8.D7 B64	C7.D7 B65	C6.D7 B66	C5.D7 B67	C4.D7 B68	C3.D7 B69	C2.D7 B70	C1.D7 B71
		FE3	FE4							
		FE2	FE5							
		FE1	FE6	FE9						
		FE0	FE7	FE8						

To encode each southbound frame at the controller, a 10-bit CRC is generated from the 72-bit data B[71:0], and a 10-bit CRC is generated from the 26-bit "A" command F[1:0]aC[23:0]. All 10 bits of the data CRC, FE[9:0] as illustrated on the bottom of Table 1-6, are latched at the controller to be combined with the 10-bit CRC generated from the 26-bit "A" command in the next southbound frame, using an exclusive-or function to create the compound checksum aE[9:0] which will be transmitted in the next southbound frame.

The generation of the compound 10-bit checksum aE[9:0] requires an XOR operation of the "A" command 10-bit CRC from this frame, with the latched 10-bits FE[9:0] CRC generated from the 72-bit data of the previous frame. The compound checksum bit aE9 for this frame comes from "A" command CRC10[9] from this frame XOR with FE9 from the 72-bit data CRC of the previous frame, aE8 for this frame comes from "A" command CRC10[8] from this frame XOR with FE8 from the 72-bit data CRC of the previous frame, and so on. This mechanism provides adequate CRC protection of the last 72 bits of the frame without adding latency to the delivery of the "A" command in the current frame.

GENERATION: Fail-over Southbound Frame: 26-bit "A" Command + 72-bit Data:

FRAME_0 = Initialization: Generate 10-bit CRC FE[9:0] from 72-bit data payload and latch FE[9:0].

FRAME_1: Generate 10-bit checksum CRC10[9:0] from 26-bit command, and 10-bit checksum FE[9:0] from 72-bit data. Generate the compound checksum aE[9:0] through bitwise XOR of FE[9:0] latched from FRAME_0 with CRC10[9:0] from FRAME_1. FE[9] XOR CRC10[9], FE[8] XOR CRC10[8], etc... Insert aE[9:0] into FRAME_1 with command and data and SEND fail-over Southbound FRAME_1.

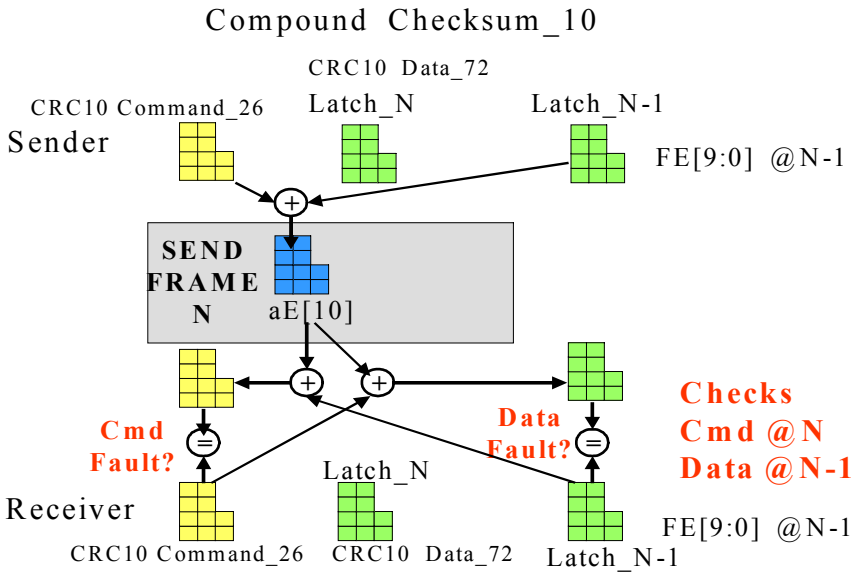
To decode the southbound frame, every AMB on the link must generate the 10-bit CRC FE[9:0] for every southbound frame, whether that frame is addressed to that specific AMB or not, and latch FE[9:0] for possible XOR with the next frame's composite checksum aE[9:0]. When a fail-over southbound frame is address to a specific AMB chip both the 10-bit CRC from the 26-bit "A" command, and the 10-bit CRC covering the 72 bit data are generated.

DECODE: Fail-over Southbound Frame: 26-bit "A" Command + 72-bit Data:

FRAME_0 = Initialize: Generate 10-bit CRC FE[9:0] for 72-bit data payload and latch FE[9:0].

FRAME_1: Generate 10-bit test checksum CRC10[9:0] from 26-bit command. Generate 10-bit checksum FE[9:0] from 72-bit data and latch for use with FRAME_2. Generate the test compound checksum TESTaE[9:0] through bitwise XOR of FE[9:0] from CRC10 on the 72-bit data latched from FRAME_0 with aE[9:0] in FRAME_1. FE[9] XOR aE[9], FE[8] XOR aE[8], etc… If the generated test compound checksum TESTaE[9:0] matches the transmitted compound checksum aE[9:0], there are no errors in the "A" command of FRAME_1. To detect faults in the transmission of the 72-bits of data from FRAME_0, bitwise XOR new CRC10[9:0] from the 26 bit command in FRAME_1 with transmitted FRAME_1 aE[9:0] and compare this result to the latched FE[9:0] from FRAME_0.

A fault in aE[9:0] indicates that both the "A" command in FRAME_1 could be faulted, and that the 72-bit data in FRAME_0 could be faulted.



The CRC of the "A" command can be checked as soon as the first 4 transfers of the frame are received and the "A" command can be used immediately without waiting for the remainder of the frame to arrive. The complexity of this scheme is relatively low compared to the benefit of minimizing DRAM access time. XOR'ing two CRCs of reasonable length results in only a small reduction in fault detection for each, but it does increase the number of detected faults, and hence retransmissions.

Table 5-63 defines the mapping of the data going to each of nine 8-bit DRAM devices for a write operation into the southbound data frame. This frame format also supports 4-bit and 16-bit DRAM devices, using the same CRC logic block input positions B[71:0]. In the Cn.Dn labels, Cn is the DRAM chip number and Dn is the DRAM data bit number.

5.4.3 Fail-over Southbound Frame CRCs

The 36 bit command payload holds one 26 bit command and the 10 bit CRC available in the fail-over operating mode. The 10-bit CRC uses the polynomial P(x) = x^10+x^9+x^7+x^6+x^4+x^3+x^2+x+1. Which in the octal base is 3337, and in the communication standard hexadecimal base polynomial naming convention which drops the degree bit from the most significant position is 0x2df. This polynomial is primitive, has linearly independent roots, and the roots of the reciprocal polynomial are also linearly independent. This polynomial has an odd number of terms, and hence will checksum random bit faults with an even or odd number of faulty bits with approximately the same detection rate. This CRC will detect all random 1-bit, 2-bit, and 3-bit faults, and any continuous fault in 10 bits or less. The serpentine mapping onto the data frame rows provides strong detection of multi-bit faults generated along a column. Wire/column failures are considered the most common fault, and this bit mapping is optimized to detect wire faults.

Logic Equations for polynomial P(x) = x^10+x^9+x^7+x^6+x^4+x^3+x^2+x+1 over 26 bits

```
B = Bit in Packet;

 CRC(0) = B(25) xor B(22) xor B(21) xor B(19) xor B(17) xor B(16) xor
               B(14) xor B(13) xor B(10) xor B(8) xor B(2) xor B(1) xor
               B(0);
   CRC(1) = B(25) xor B(23) xor B(21) xor B(20) xor B(19) xor B(18) xor
               B(16) xor B(15) xor B(13) xor B(11) xor B(10) xor B(9) xor
               B(8) xor B(3) xor B(0);
   CRC(2) = B(25) xor B(24) xor B(20) xor B(13) xor B(12) xor B(11) xor
               B(9) xor B(8) xor B(4) xor B(2) xor B(0);
   CRC(3) = B(22) xor B(19) xor B(17) xor B(16) xor B(12) xor B(9) xor
               B(8) xor B(5) xor B(3) xor B(2) xor B(0);
   CRC(4) = B(25) xor B(23) xor B(22) xor B(21) xor B(20) xor B(19) xor
               B(18) xor B(16) xor B(14) xor B(9) xor B(8) xor B(6) xor
               B(4) xor B(3) xor B(2) xor B(0);
   CRC(5) = B(24) xor B(23) xor B(22) xor B(21) xor B(20) xor B(19) xor
               B(17) xor B(15) xor B(10) xor B(9) xor B(7) xor B(5) xor
               B(4) xor B(3) xor B(1);
   CRC(6) = B(24) xor B(23) xor B(20) xor B(19) xor B(18) xor B(17) xor
               B(14) xor B(13) xor B(11) xor B(6) xor B(5) xor B(4) xor
               B(1) xor B(0);
   CRC(7) = B(24) xor B(22) xor B(20) xor B(18) xor B(17) xor B(16) xor
               B(15) xor B(13) xor B(12) xor B(10) xor B(8) xor B(7) xor
               B(6) xor B(5) xor B(0);
   CRC(8) = B(25) xor B(23) xor B(21) xor B(19) xor B(18) xor B(17) xor
               B(16) xor B(14) xor B(13) xor B(11) xor B(9) xor B(8) xor
               B(7) xor B(6) xor B(1);
   CRC(9) = B(25) xor B(24) xor B(21) xor B(20) xor B(18) xor B(16) xor
               B(15) xor B(13) xor B(12) xor B(9) xor B(7) xor B(1) xor
               B(0);
```

The 10-bit CRC for the 72-bit data uses the polynomial:

P(x) = x^10+x^9+x^7+x^6+x^4+x^3+x^2+x+1

In the octal base, this is 3337, and in the communication standard hexadecimal base polynomial naming convention, which drops the degree bit from the most significant position, it is 0x2df. This polynomial is primitive, has linearly independent roots, and the roots of the reciprocal polynomial are also linearly independent. This polynomial has an odd number of terms, and hence will checksum random bit faults with an even or odd number of faulty bits with approximately the same detection rate. This CRC will detect all random 1-bit, 2-bit, and 3-bit faults, and any continuous fault of 10 bits or less. The serpentine mapping onto the write data frame rows provides strong detection of multi-bit faults generated along a row, and complements the fault correction and detection capabilities of the ChipFail ECC that is mapped down the write data frame columns by the Chip.Data_bit ordering shown in Table 1-6.

Logic Equations for polynomial P(x) = x^22+x^14+x^13+x^12+x^7+x^3+x^2+x+1 over 72 bits

```
B = Bit in Packet;

CRC(0) = B(71) xor B(65) xor B(63) xor B(62) xor B(59) xor B(58) xor
                B(57) xor B(55) xor B(53) xor B(52) xor B(51) xor B(50) xor
                B(49) xor B(48) xor B(45) xor B(44) xor B(41) xor B(36) xor
                B(34) xor B(32) xor B(31) xor B(29) xor B(26) xor B(25) xor
                B(22) xor B(21) xor B(19) xor B(17) xor B(16) xor B(14) xor
                B(13) xor B(10) xor B(8) xor B(2) xor B(1) xor B(0);
    CRC(1) = B(71) xor B(66) xor B(65) xor B(64) xor B(62) xor B(60) xor
                B(57) xor B(56) xor B(55) xor B(54) xor B(48) xor B(46) xor
                B(44) xor B(42) xor B(41) xor B(37) xor B(36) xor B(35) xor
                B(34) xor B(33) xor B(31) xor B(30) xor B(29) xor B(27) xor
                B(25) xor B(23) xor B(21) xor B(20) xor B(19) xor B(18) xor
                B(16) xor B(15) xor B(13) xor B(11) xor B(10) xor B(9) xor
                B(8) xor B(3) xor B(0);
    CRC(2) = B(71) xor B(67) xor B(66) xor B(62) xor B(61) xor B(59) xor
                B(56) xor B(53) xor B(52) xor B(51) xor B(50) xor B(48) xor
                B(47) xor B(44) xor B(43) xor B(42) xor B(41) xor B(38) xor
                B(37) xor B(35) xor B(30) xor B(29) xor B(28) xor B(25) xor
                B(24) xor B(20) xor B(13) xor B(12) xor B(11) xor B(9) xor
                B(8) xor B(4) xor B(2) xor B(0);
    CRC(3) = B(71) xor B(68) xor B(67) xor B(65) xor B(60) xor B(59) xor
                B(58) xor B(55) xor B(54) xor B(50) xor B(43) xor B(42) xor
                B(41) xor B(39) xor B(38) xor B(34) xor B(32) xor B(30) xor
                B(22) xor B(19) xor B(17) xor B(16) xor B(12) xor B(9) xor
                B(8) xor B(5) xor B(3) xor B(2) xor B(0);
    CRC(4) = B(71) xor B(69) xor B(68) xor B(66) xor B(65) xor B(63) xor
                B(62) xor B(61) xor B(60) xor B(58) xor B(57) xor B(56) xor
                B(53) xor B(52) xor B(50) xor B(49) xor B(48) xor B(45) xor
                B(43) xor B(42) xor B(41) xor B(40) xor B(39) xor B(36) xor
                B(35) xor B(34) xor B(33) xor B(32) xor B(29) xor B(26) xor
                B(25) xor B(23) xor B(22) xor B(21) xor B(20) xor B(19) xor
                B(18) xor B(16) xor B(14) xor B(9) xor B(8) xor B(6) xor
                B(4) xor B(3) xor B(2) xor B(0);
    CRC(5) = B(70) xor B(69) xor B(67) xor B(66) xor B(64) xor B(63) xor
                B(62) xor B(61) xor B(59) xor B(58) xor B(57) xor B(54) xor
                B(53) xor B(51) xor B(50) xor B(49) xor B(46) xor B(44) xor
                B(43) xor B(42) xor B(41) xor B(40) xor B(37) xor B(36) xor
                B(35) xor B(34) xor B(33) xor B(30) xor B(27) xor B(26) xor
                B(24) xor B(23) xor B(22) xor B(21) xor B(20) xor B(19) xor
                B(17) xor B(15) xor B(10) xor B(9) xor B(7) xor B(5) xor
                B(4) xor B(3) xor B(1);
    CRC(6) = B(70) xor B(68) xor B(67) xor B(64) xor B(60) xor B(57) xor
                B(54) xor B(53) xor B(49) xor B(48) xor B(47) xor B(43) xor
                B(42) xor B(38) xor B(37) xor B(35) xor B(32) xor B(29) xor
                B(28) xor B(27) xor B(26) xor B(24) xor B(23) xor B(20) xor
```

```
            B(19) xor B(18) xor B(17) xor B(14) xor B(13) xor B(11) xor
            B(6) xor B(5) xor B(4) xor B(1) xor B(0);
CRC(7) = B(69) xor B(68) xor B(63) xor B(62) xor B(61) xor B(59) xor
            B(57) xor B(54) xor B(53) xor B(52) xor B(51) xor B(45) xor
            B(43) xor B(41) xor B(39) xor B(38) xor B(34) xor B(33) xor
            B(32) xor B(31) xor B(30) xor B(28) xor B(27) xor B(26) xor
            B(24) xor B(22) xor B(20) xor B(18) xor B(17) xor B(16) xor
            B(15) xor B(13) xor B(12) xor B(10) xor B(8) xor B(7) xor
            B(6) xor B(5) xor B(0);
CRC(8) = B(70) xor B(69) xor B(64) xor B(63) xor B(62) xor B(60) xor
            B(58) xor B(55) xor B(54) xor B(53) xor B(52) xor B(46) xor
            B(44) xor B(42) xor B(40) xor B(39) xor B(35) xor B(34) xor
            B(33) xor B(32) xor B(31) xor B(29) xor B(28) xor B(27) xor
            B(25) xor B(23) xor B(21) xor B(19) xor B(18) xor B(17) xor
            B(16) xor B(14) xor B(13) xor B(11) xor B(9) xor B(8) xor
            B(7) xor B(6) xor B(1);
CRC(9) = B(70) xor B(64) xor B(62) xor B(61) xor B(58) xor B(57) xor
            B(56) xor B(54) xor B(52) xor B(51) xor B(50) xor B(49) xor
            B(48) xor B(47) xor B(44) xor B(43) xor B(40) xor B(35) xor
            B(33) xor B(31) xor B(30) xor B(28) xor B(25) xor B(24) xor
            B(21) xor B(20) xor B(18) xor B(16) xor B(15) xor B(13) xor
            B(12) xor B(9) xor B(7) xor B(1) xor B(0);
```

5.4.3.1 Fail-over Southbound Frame with Multiple Commands

As shown in Table 5-64, the fail-over southbound frame can also hold just multiple commands. One or two commands, “B” command and “C” command, can share the fail-over southbound frame with the mandatory “A” command. For error detection, these commands are treated just as the 72-bit data payload and use the same CRC logic block and B[71:0] input mapping.

Table 5-64 — Fail-over Southbound Frame With Three Commands

Xfr	Bit									
	9	8	7	6	5	4	3	2	1	0
0		aE3	aE4	F0 B24	aC20 B23	aC16 B16	aC12 B15	aC8 B8	aC4 B7	aC0 B0
1		aE2	aE5	F1 B25	aC21 B22	aC17 B17	aC13 B14	aC9 B9	aC5 B6	aC1 B1
2		aE1	aE6	aE9	aC22 B21	aC18 B17	aC14 B13	aC10 B10	aC6 B5	aC2 B2
3		aE0	aE7	aE8	aC23 B20	aC19 B19	aC15 B12	aC11 B11	aC7 B4	aC3 B3
4		0 B8	0 B7	0 B6	bC20 B5	bC16 B4	bC12 B3	bC8 B2	bC4 B1	bC0 B0
5		0 B9	0 B10	0 B11	bC21 B12	bC17 B13	bC13 B14	bC9 B15	bC5 B16	bC1 B17
6		0 B26	0 B25	0 B24	bC22 B23	bC18 B22	bC14 B21	bC10 B20	bC6 B19	bC2 B18
7		0 B27	0 B28	0 B29	bC23 B30	bC19 B31	bC15 B32	bC11 B33	bC7 B34	bC3 B35
8		0 B44	0 B43	0 B42	cC20 B41	cC16 B40	cC12 B39	cC8 B38	cC4 B37	cC0 B36
9		0 B45	0 B46	0 B47	cC21 B48	cC17 B49	cC13 B50	cC9 B51	cC5 B52	cC1 B53
10		0 B62	0 B61	0 B60	cC22 B59	cC18 B58	cC14 B57	cC10 B56	cC6 B55	cC2 B54
11		0 B63	0 B64	0 B65	cC23 B66	cC19 B67	cC15 B68	cC11 B69	cC7 B70	cC3 B71
		FE3	FE4							
		FE2	FE5							
		FE1	FE6	FE9						
		FE0	FE7	FE8						

5.4.4 CRC Generation: 14-bit Lane Northbound Data Frame

The 14-bit lane data frame is the highest RAS mode of operation for the northbound channel. In this mode a 12-bit CRC is delivered in the E[11:0] field on the 13th & 14th northbound bit lanes during the transfer of each 72-bit data payload. Table 5-65 defines the bit positions of the CRC code in the northbound frame. This table also defines the mapping of the data from each of eighteen 4-bit DRAM devices into the Northbound Data frame. This frame format also supports 8-bit DRAM devices and the mapping of those devices is defined in Table 1-9. In the Cn.Dn labels, Cn is the DRAM chip number and Dn is the DRAM data bit number. In these tables B[71;0] correspond to the bit position used by the CRC checksum generation logic block.

Table 5-65 — 14-bit Northbound Data Frame Format (with 4-bit Devices)

Xfr	Bit													
	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	E1.0	E1.11	C17.D2 B11	C16.D0 B10	C14.D2 B9	C13.D0 B8	C11.D2 B7	C10.D0 B6	C8.D2 B5	C7.D0 B4	C5.D2 B3	C4.D0 B2	C2.D2 B1	C1.D0 B0
1	E1.1	E1.10	C17.D3 B12	C16.D1 B13	C14.D3 B14	C13.D1 B15	C11.D3 B16	C10.D1 B17	C8.D3 B18	C7.D1 B19	C5.D3 B20	C4.D1 B21	C2.D3 B22	C1.D1 B23
2	E1.2	E1.9	C18.D0 B35	C16.D2 B34	C15.D0 B33	C13.D2 B32	C12.D0 B31	C10.D2 B30	C9.D0 B29	C7.D2 B28	C6.D0 B27	C4.D2 B26	C3.D0 B25	C1.D2 B24
3	E1.3	E1.8	C18.D1 B36	C16.D3 B37	C15.D1 B38	C13.D3 B39	C12.D1 B40	C10.D3 B41	C9.D1 B42	C7.D3 B43	C6.D1 B44	C4.D3 B45	C3.D1 B46	C1.D3 B47
4	E1.4	E1.7	C18.D2 B59	C17.D0 B58	C15.D2 B57	C14.D0 B56	C12.D2 B55	C11.D0 B54	C9.D2 B53	C8.D0 B52	C6.D2 B51	C5.D0 B50	C3.D2 B49	C2.D0 B48
5	E1.5	E1.6	C18.D3 B60	C17.D1 B61	C15.D3 B62	C14.D1 B63	C12.D3 B64	C11.D1 B65	C9.D3 B66	C8.D1 B67	C6.D3 B68	C5.D1 B69	C3.D3 B70	C2.D1 B71
6	E2.0	E2.11	C17.D2 B11	C16.D0 B10	C14.D2 B9	C13.D0 B8	C11.D2 B7	C10.D0 B6	C8.D2 B5	C7.D0 B4	C5.D2 B3	C4.D0 B2	C2.D2 B1	C1.D0 B0
7	E2.1	E2.10	C17.D3 B12	C16.D1 B13	C14.D3 B14	C13.D1 B15	C11.D3 B16	C10.D1 B17	C8.D3 B18	C7.D1 B19	C5.D3 B20	C4.D1 B21	C2.D3 B22	C1.D1 B23
8	E2.2	E2.9	C18.D0 B35	C16.D2 B34	C15.D0 B33	C13.D2 B32	C12.D0 B31	C10.D2 B30	C9.D0 B29	C7.D2 B28	C6.D0 B27	C4.D2 B26	C3.D0 B25	C1.D2 B24
9	E2.3	E2.8	C18.D1 B36	C16.D3 B37	C15.D1 B38	C13.D3 B39	C12.D1 B40	C10.D3 B41	C9.D1 B42	C7.D3 B43	C6.D1 B44	C4.D3 B45	C3.D1 B46	C1.D3 B47
10	E2.4	E2.7	C18.D2 B59	C17.D0 B58	C15.D2 B57	C14.D0 B56	C12.D2 B55	C11.D0 B54	C9.D2 B53	C8.D0 B52	C6.D2 B51	C5.D0 B50	C3.D2 B49	C2.D0 B48
11	E2.5	E2.6	C18.D3 B60	C17.D1 B61	C15.D3 B62	C14.D1 B63	C12.D3 B64	C11.D1 B65	C9.D3 B66	C8.D1 B67	C6.D3 B68	C5.D1 B69	C3.D3 B70	C2.D1 B71

The 12-bit CRC uses the polynomial P(x) = x^12+x^11+x^10+x^8+x^6+x^4+x^3+x^1+1. Which in the octal base is 16533, and in the communication standard hexadecimal base polynomial naming convention that drops the degree bit from the most significant position is 0xd5b. This polynomial is primitive, has linearly independent roots, and the roots of the reciprocal polynomial are also linearly independent. This polynomial has an odd number of terms, and hence will checksum random bit faults with an even or odd number of faulty bits with approximately the same detection rate. This CRC will detect all random 1-bit, 2-bit, and 3-bit faults, and any continuous fault in B[71:0] of 12 bits or less. The serpentine mapping onto the data frame rows provides strong detection of multi-bit faults generated along a row, and complements the fault correction and detection capabilities of the ChipFail ECC that is mapped down the data frame columns by the Chip.Data_bit ordering

ENCODE:

- Logically B[71] is the MSB of the dividend for the division by divisor P(x) to get the checksum remainder.
- Input 72 data bits into the logic block with MSB starting at B[71] as mapped in Table 1-8
- Run CRC logic on this 72 bit packet to get checksum remainder CRC[11:0],
- Build the transmission packet by inserting the new checksum CRC[11:0] into E[11:0]
- SEND PACKET

DECODE:

- Run CRC logic on 72 bits B[71:0] as mapped in Table 1-8

- If CRC[11:0] does not match transmitted E[11:0] , a fault has been detected

Logic Equations for polynomial P(x) = x^12+x^11+x^10+x^8+x^6+x^4+x^3+x^1+1 over 72 bits

```
B = Bit in Packet;

    CRC(0) = B(71) xor B(70) xor B(68) xor B(67) xor B(66) xor B(63) xor
                 B(58) xor B(56) xor B(55) xor B(54) xor B(53) xor B(52) xor
                 B(49) xor B(48) xor B(46) xor B(44) xor B(42) xor B(41) xor
                 B(40) xor B(39) xor B(38) xor B(37) xor B(36) xor B(35) xor
                 B(34) xor B(29) xor B(26) xor B(25) xor B(22) xor B(21) xor
                 B(19) xor B(18) xor B(16) xor B(12) xor B(9) xor B(8) xor
                 B(7) xor B(6) xor B(3) xor B(1) xor B(0);
    CRC(1) = B(70) xor B(69) xor B(66) xor B(64) xor B(63) xor B(59) xor
                 B(58) xor B(57) xor B(52) xor B(50) xor B(48) xor B(47) xor
                 B(46) xor B(45) xor B(44) xor B(43) xor B(34) xor B(30) xor
                 B(29) xor B(27) xor B(25) xor B(23) xor B(21) xor B(20) xor
                 B(18) xor B(17) xor B(16) xor B(13) xor B(12) xor B(10) xor
                 B(6) xor B(4) xor B(3) xor B(2) xor B(0);
    CRC(2) = B(71) xor B(70) xor B(67) xor B(65) xor B(64) xor B(60) xor
                 B(59) xor B(58) xor B(53) xor B(51) xor B(49) xor B(48) xor
                 B(47) xor B(46) xor B(45) xor B(44) xor B(35) xor B(31) xor
                 B(30) xor B(28) xor B(26) xor B(24) xor B(22) xor B(21) xor
                 B(19) xor B(18) xor B(17) xor B(14) xor B(13) xor B(11) xor
                 B(7) xor B(5) xor B(4) xor B(3) xor B(1);
    CRC(3) = B(70) xor B(67) xor B(65) xor B(63) xor B(61) xor B(60) xor
                 B(59) xor B(58) xor B(56) xor B(55) xor B(53) xor B(50) xor
                 B(47) xor B(45) xor B(44) xor B(42) xor B(41) xor B(40) xor
                 B(39) xor B(38) xor B(37) xor B(35) xor B(34) xor B(32) xor
                 B(31) xor B(27) xor B(26) xor B(23) xor B(21) xor B(20) xor
                 B(16) xor B(15) xor B(14) xor B(9) xor B(7) xor B(5) xor
                 B(4) xor B(3) xor B(2) xor B(1) xor B(0);
    CRC(4) = B(70) xor B(67) xor B(64) xor B(63) xor B(62) xor B(61) xor
                 B(60) xor B(59) xor B(58) xor B(57) xor B(55) xor B(53) xor
                 B(52) xor B(51) xor B(49) xor B(45) xor B(44) xor B(43) xor
                 B(37) xor B(34) xor B(33) xor B(32) xor B(29) xor B(28) xor
                 B(27) xor B(26) xor B(25) xor B(24) xor B(19) xor B(18) xor
                 B(17) xor B(15) xor B(12) xor B(10) xor B(9) xor B(7) xor
                 B(5) xor B(4) xor B(2) xor B(0);
    CRC(5) = B(71) xor B(68) xor B(65) xor B(64) xor B(63) xor B(62) xor
                 B(61) xor B(60) xor B(59) xor B(58) xor B(56) xor B(54) xor
                 B(53) xor B(52) xor B(50) xor B(46) xor B(45) xor B(44) xor
                 B(38) xor B(35) xor B(34) xor B(33) xor B(30) xor B(29) xor
                 B(28) xor B(27) xor B(26) xor B(25) xor B(20) xor B(19) xor
                 B(18) xor B(16) xor B(13) xor B(11) xor B(10) xor B(8) xor
                 B(6) xor B(5) xor B(3) xor B(1);
    CRC(6) = B(71) xor B(70) xor B(69) xor B(68) xor B(67) xor B(65) xor
                 B(64) xor B(62) xor B(61) xor B(60) xor B(59) xor B(58) xor
                 B(57) xor B(56) xor B(52) xor B(51) xor B(49) xor B(48) xor
                 B(47) xor B(45) xor B(44) xor B(42) xor B(41) xor B(40) xor
                 B(38) xor B(37) xor B(31) xor B(30) xor B(28) xor B(27) xor
                 B(25) xor B(22) xor B(20) xor B(18) xor B(17) xor B(16) xor
                 B(14) xor B(11) xor B(8) xor B(4) xor B(3) xor B(2) xor
                 B(1) xor B(0);
    CRC(7) = B(71) xor B(70) xor B(69) xor B(68) xor B(66) xor B(65) xor
                 B(63) xor B(62) xor B(61) xor B(60) xor B(59) xor B(58) xor
                 B(57) xor B(53) xor B(52) xor B(50) xor B(49) xor B(48) xor
                 B(46) xor B(45) xor B(43) xor B(42) xor B(41) xor B(39) xor
```

```
            B(38) xor B(32) xor B(31) xor B(29) xor B(28) xor B(26) xor
            B(23) xor B(21) xor B(19) xor B(18) xor B(17) xor B(15) xor
            B(12) xor B(9) xor B(5) xor B(4) xor B(3) xor B(2) xor
            B(1);
CRC(8) = B(69) xor B(68) xor B(64) xor B(62) xor B(61) xor B(60) xor
            B(59) xor B(56) xor B(55) xor B(52) xor B(51) xor B(50) xor
            B(48) xor B(47) xor B(43) xor B(41) xor B(38) xor B(37) xor
            B(36) xor B(35) xor B(34) xor B(33) xor B(32) xor B(30) xor
            B(27) xor B(26) xor B(25) xor B(24) xor B(21) xor B(20) xor
            B(13) xor B(12) xor B(10) xor B(9) xor B(8) xor B(7) xor
            B(5) xor B(4) xor B(2) xor B(1) xor B(0);
CRC(9) = B(70) xor B(69) xor B(65) xor B(63) xor B(62) xor B(61) xor
            B(60) xor B(57) xor B(56) xor B(53) xor B(52) xor B(51) xor
            B(49) xor B(48) xor B(44) xor B(42) xor B(39) xor B(38) xor
            B(37) xor B(36) xor B(35) xor B(34) xor B(33) xor B(31) xor
            B(28) xor B(27) xor B(26) xor B(25) xor B(22) xor B(21) xor
            B(14) xor B(13) xor B(11) xor B(10) xor B(9) xor B(8) xor
            B(6) xor B(5) xor B(3) xor B(2) xor B(1);
CRC(10) = B(68) xor B(67) xor B(64) xor B(62) xor B(61) xor B(57) xor
             B(56) xor B(55) xor B(50) xor B(48) xor B(46) xor B(45) xor
             B(44) xor B(43) xor B(42) xor B(41) xor B(32) xor B(28) xor
             B(27) xor B(25) xor B(23) xor B(21) xor B(19) xor B(18) xor
             B(16) xor B(15) xor B(14) xor B(11) xor B(10) xor B(8) xor
             B(4) xor B(2) xor B(1) xor B(0);
CRC(11) = B(71) xor B(70) xor B(69) xor B(67) xor B(66) xor B(65) xor
             B(62) xor B(57) xor B(55) xor B(54) xor B(53) xor B(52) xor
             B(51) xor B(48) xor B(47) xor B(45) xor B(43) xor B(41) xor
             B(40) xor B(39) xor B(38) xor B(37) xor B(36) xor B(35) xor
             B(34) xor B(33) xor B(28) xor B(25) xor B(24) xor B(21) xor
             B(20) xor B(18) xor B(17) xor B(15) xor B(11) xor B(8) xor
             B(7) xor B(6) xor B(5) xor B(2) xor B(0);
```

5.4.5 13-bit Lane Northbound Data Frame

The 13-bit lane data frame is the medium RAS mode of operation for the northbound channel. In this mode a 6-bit CRC is delivered in the E[5:0] field on the 13th northbound bit lane during the transfer of each 72-bit data payload. Table 5-66 defines the bit positions of the CRC code in the northbound frame. This table also defines the mapping of the data pins of nine 8-bit DRAM devices to the frame data bits. This frame format also supports 4-bit DRAM devices mapped as shown in Table 5-65. In the Cn.Dn labels, Cn is the DRAM chip number and Dn is the DRAM data bit number. In these tables B[71:0] corresponds to the the bit position used by the CRC checksum generation logic block. In wire fail-over, only the DRAM ECC is available to protect against link errors.

Table 5-66 — 13-bit Northbound Data Frame Format (with 8-bit Devices)

Xfr	Bit													
	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0		E1.0	C9.D2 B11	C8.D4 B10	C7.D6 B9	C7.D0 B8	C6.D2 B7	C5.D4 B6	C4.D6 B5	C4.D0 B4	C3.D2 B3	C2.D4 B2	C1.D6 B1	C1.D0 B0
1		E1.1	C9.D3 B12	C8.D5 B13	C7.D7 B14	C7.D1 B15	C6.D3 B16	C5.D5 B17	C4.D7 B18	C4.D1 B19	C3.D3 B20	C2.D5 B21	C1.D7 B22	C1.D1 B23
2		E1.2	C9.D4 B35	C8.D6 B34	C8.D0 B33	C7.D2 B32	C6.D4 B31	C5.D6 B30	C5.D0 B29	C4.D2 B28	C3.D4 B27	C2.D6 B26	C2.D0 B25	C1.D2 B24
3		E1.3	C9.D5 B36	C8.D7 B37	C8.D1 B38	C7.D3 B39	C6.D5 B40	C5.D7 B41	C5.D1 B42	C4.D3 B42	C3.D5 B44	C2.D7 B45	C2.D1 B46	C1.D3 B47
4		E1.4	C9.D6 B59	C9.D0 B58	C8.D2 B57	C7.D4 B56	C6.D6 B55	C6.D0 B54	C5.D2 B53	C4.D4 B52	C3.D6 B51	C3.D0 B50	C2.D2 B49	C1.D4 B48
5		E1.5	C9.D7 B60	C9.D1 B61	C8.D3 B62	C7.D5 B63	C6.D7 B64	C6.D1 B65	C5.D3 B66	C4.D5 B67	C3.D7 B68	C3.D1 B69	C2.D3 B70	C1.D5 B71
6		E2.0	C9.D2 B11	C8.D4 B10	C7.D6 B9	C7.D0 B8	C6.D2 B7	C5.D4 B6	C4.D6 B5	C4.D0 B4	C3.D2 B3	C2.D4 B2	C1.D6 B1	C1.D0 B0
7		E2.1	C9.D3 B12	C8.D5 B13	C7.D7 B14	C7.D1 B15	C6.D3 B16	C5.D5 B17	C4.D7 B18	C4.D1 B19	C3.D3 B20	C2.D5 B21	C1.D7 B22	C1.D1 B23
8		E2.2	C9.D4 B35	C8.D6 B34	C8.D0 B33	C7.D2 B32	C6.D4 B31	C5.D6 B30	C5.D0 B29	C4.D2 B28	C3.D4 B27	C2.D6 B26	C2.D0 B25	C1.D2 B24
9		E2.3	C9.D5 B36	C8.D7 B37	C8.D1 B38	C7.D3 B39	C6.D5 B40	C5.D7 B41	C5.D1 B42	C4.D3 B42	C3.D5 B44	C2.D7 B45	C2.D1 B46	C1.D3 B47
10		E2.4	C9.D6 B59	C9.D0 B58	C8.D2 B57	C7.D4 B56	C6.D6 B55	C6.D0 B54	C5.D2 B53	C4.D4 B52	C3.D6 B51	C3.D0 B50	C2.D2 B49	C1.D4 B48
11		E2.5	C9.D7 B60	C9.D1 B61	C8.D3 B62	C7.D5 B63	C6.D7 B64	C6.D1 B65	C5.D3 B66	C4.D5 B67	C3.D7 B68	C3.D1 B69	C2.D3 B70	C1.D5 B71

The 6-bit CRC uses the polynomial P(x) = x^6+x^5+x^2+x^1+1. Which in the octal base is 147, and in the communication standard hexadecimal base polynomial naming convention that drops the degree bit from the most significant position is 0x27. This polynomial is primitive, has linearly independent roots, and the roots of the reciprocal polynomial are also linearly independent. This polynomial has an odd number of terms, and hence will checksum random bit faults with an even or odd number of faulty bits with approximately the same detection rate. The 72 bits in this packet is larger than the 2^6=64 bit coverage range of the polynomial, and results in slightly lower than textbook error detection of random multi-bit faults. This CRC will detect any continuous fault in B[71:0] of 6 bits or less. The serpentine mapping onto the data frame rows provides strong detection of multi-bit faults generated along a row, and complements the fault correction and detection capabilities of the ChipFail ECC that is mapped down the data frame columns by the Chip.Data_bit ordering.

ENCODE:

- Logically B[71] is the MSB of the dividend for the division by divisor P(x) to get the checksum remainder.
- Input 72 data bits into the logic block with MSB starting at B[71] as mapped in Table 1-9
- Run CRC logic on this 72 bit packet to get checksum remainder CRC[5:0],
- Build the transmission packet by inserting the new checksum CRC[5:0] into E[5:0]
- SEND PACKET

DECODE:

- Run CRC logic on 72 bits B[71:0] as mapped in Table 1-8

- If CRC[5:0] does not match transmitted E[5:0] , a fault has been detected

Logic Equations for polynomial P(x) = x^12+x^11+x^10+x^8+x^6+x^4+x^3+x^1+1 over 72 bits

```
B = Bit in Packet;

        CRC(0) = B(69) xor B(66) xor B(65) xor B(64) xor B(63) xor B(57) xor
                 B(56) xor B(54) xor B(53) xor B(52) xor B(49) xor B(48) xor
                 B(44) xor B(43) xor B(42) xor B(40) xor B(38) xor B(37) xor
                 B(36) xor B(35) xor B(34) xor B(33) xor B(31) xor B(30) xor
                 B(28) xor B(24) xor B(19) xor B(17) xor B(16) xor B(13) xor
                 B(11) xor B(9) xor B(6) xor B(3) xor B(2) xor B(1) xor
                 B(0);
    CRC(1) = B(70) xor B(69) xor B(67) xor B(63) xor B(58) xor B(56) xor
                 B(55) xor B(52) xor B(50) xor B(48) xor B(45) xor B(42) xor
                 B(41) xor B(40) xor B(39) xor B(33) xor B(32) xor B(30) xor
                 B(29) xor B(28) xor B(25) xor B(24) xor B(20) xor B(19) xor
                 B(18) xor B(16) xor B(14) xor B(13) xor B(12) xor B(11) xor
                 B(10) xor B(9) xor B(7) xor B(6) xor B(4) xor B(0);
    CRC(2) = B(71) xor B(70) xor B(69) xor B(68) xor B(66) xor B(65) xor
                 B(63) xor B(59) xor B(54) xor B(52) xor B(51) xor B(48) xor
                 B(46) xor B(44) xor B(41) xor B(38) xor B(37) xor B(36) xor
                 B(35) xor B(29) xor B(28) xor B(26) xor B(25) xor B(24) xor
                 B(21) xor B(20) xor B(16) xor B(15) xor B(14) xor B(12) xor
                 B(10) xor B(9) xor B(8) xor B(7) xor B(6) xor B(5) xor
                 B(3) xor B(2) xor B(0);
    CRC(3) = B(71) xor B(70) xor B(69) xor B(67) xor B(66) xor B(64) xor
                 B(60) xor B(55) xor B(53) xor B(52) xor B(49) xor B(47) xor
                 B(45) xor B(42) xor B(39) xor B(38) xor B(37) xor B(36) xor
                 B(30) xor B(29) xor B(27) xor B(26) xor B(25) xor B(22) xor
                 B(21) xor B(17) xor B(16) xor B(15) xor B(13) xor B(11) xor
                 B(10) xor B(9) xor B(8) xor B(7) xor B(6) xor B(4) xor
                 B(3) xor B(1);
    CRC(4) = B(71) xor B(70) xor B(68) xor B(67) xor B(65) xor B(61) xor
                 B(56) xor B(54) xor B(53) xor B(50) xor B(48) xor B(46) xor
                 B(43) xor B(40) xor B(39) xor B(38) xor B(37) xor B(31) xor
                 B(30) xor B(28) xor B(27) xor B(26) xor B(23) xor B(22) xor
                 B(18) xor B(17) xor B(16) xor B(14) xor B(12) xor B(11) xor
                 B(10) xor B(9) xor B(8) xor B(7) xor B(5) xor B(4) xor
                 B(2);
    CRC(5) = B(71) xor B(68) xor B(65) xor B(64) xor B(63) xor B(62) xor
                 B(56) xor B(55) xor B(53) xor B(52) xor B(51) xor B(48) xor
                 B(47) xor B(43) xor B(42) xor B(41) xor B(39) xor B(37) xor
                 B(36) xor B(35) xor B(34) xor B(33) xor B(32) xor B(30) xor
                 B(29) xor B(27) xor B(23) xor B(18) xor B(16) xor B(15) xor
                 B(12) xor B(10) xor B(8) xor B(5) xor B(2) xor B(1) xor
                 B(0);
```

5.4.6 12-bit Lane Northbound Data Frame

This is the lowest RAS mode of operation for the northbound channel. In this mode a 6-bit CRC is delivered in the E[5:0] field on the 13th northbound bit lane during the transfer of each 64-bit data payload. The data does not contain ECC bits and the CRC code is the only form of protection from link transmission faults. Table 1-10 defines the bit positions of the CRC code in the 12-bit lane northbound frame. This table also defines the mapping of the data pins of four 16-bit DRAM devices to the frame data bits. This frame format also supports 4-bit and 8-bit DRAM devices mapped as shown in Table 1-8 and 1-9. In these tables B[71:0] correspond to the bit position used by the CRC checksum generation logic block.

Table 5-67 — 12-bit Northbound Data Frame Format (with 16-bit Devices)

Xfr	Bit													
	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0			E1.0	C4.D12 B10	C4.D6 B9	C4.D0 B8	C3.D10 B7	C3.D4 B6	C2.D14 B5	C2.D8 B4	C2.D2 B3	C1.D12 B2	C1.D6 B1	C1.D0 B0
1			E1.1	C4.D13 B13	C4.D7 B14	C4.D1 B15	C3.D11 B16	C3.D5 B17	C2.D15 B18	C2.D9 B19	C2.D3 B20	C1.D13 B21	C1.D7 B22	C1.D1 B23
2			E1.2	C4.D14 B34	C4.D8 B33	C4.D2 B32	C3.D12 B31	C3.D6 B30	C3.D0 B29	C2.D10 B28	C2.D4 B27	C1.D14 B26	C1.D8 B25	C1.D2 B24
3			E1.3	C4.D15 B37	C4.D9 B38	C4.D3 B39	C3.D13 B40	C3.D7 B41	C3.D1 B42	C2.D11 B43	C2.D5 B44	C1.D15 B45	C1.D9 B46	C1.D3 B47
4			E1.4	0 B58	C4.D10 B57	C4.D4 B56	C3.D14 B55	C3.D8 B54	C3.D2 B53	C2.D12 B52	C2.D6 B51	C2.D0 B50	C1.D10 B49	C1.D4 B48
5			E1.5	0 B61	C4.D11 B62	C4.D5 B63	C3.D7 B64	C3.D9 B65	C3.D3 B66	C2.D13 B67	C2.D7 B68	C2.D1 B69	C1.D11 B70	C1.D5 B71
6			E2.0	C4.D12 B10	C4.D6 B9	C4.D0 B8	C3.D10 B7	C3.D4 B6	C2.D14 B5	C2.D8 B4	C2.D2 B3	C1.D12 B2	C1.D6 B1	C1.D0 B0
7			E2.1	C4.D13 B13	C4.D7 B14	C4.D1 B15	C3.D11 B16	C3.D5 B17	C2.D15 B18	C2.D9 B19	C2.D3 B20	C1.D13 B21	C1.D7 B22	C1.D1 B23
8			E2.2	C4.D14 B34	C4.D8 B33	C4.D2 B32	C3.D12 B31	C3.D6 B30	C3.D0 B29	C2.D10 B28	C2.D4 B27	C1.D14 B26	C1.D8 B25	C1.D2 B24
9			E2.3	C4.D15 B37	C4.D9 B38	C4.D3 B39	C3.D13 B40	C3.D7 B41	C3.D1 B42	C2.D11 B43	C2.D5 B44	C1.D15 B45	C1.D9 B46	C1.D3 B47
10			E2.4	0 B58	C4.D10 B57	C4.D4 B56	C3.D14 B55	C3.D8 B54	C3.D2 B53	C2.D12 B52	C2.D6 B51	C2.D0 B50	C1.D10 B49	C1.D4 B48
11			E2.5	0 B61	C4.D11 B62	C4.D5 B63	C3.D7 B64	C3.D9 B65	C3.D3 B66	C2.D13 B67	C2.D7 B68	C2.D1 B69	C1.D11 B70	C1.D5 B71

The same 6-bit CRC polynomial and logic block used for the 13 bit lane northbound data frame with 72 bits of data is re-used here for the 12 bit lane northbound data frame, with eight zeros applied to the CRC generation logic inputs:

0=B[11]=B[12]=B[35]=B[36]=B[58]=B[59]=B[60]=B[61].

5.4.7 Write and Read Data ECC Error Protection

FBD makes provision for both read and write data to be protected with system defined ECC check bits per data block by supporting 8 check bits per 64 data bits in 14 and 13 lane northbound frames. The host generates the ECC code and passes it along with the write data to the AMB. The AMB will store the ECC along with the data in the DRAM memory. The AMB will not check the ECC code for errors and the host may use whatever algorithm it chooses. This allows the host to use various complex ECC algorithms, possibly spread across multiple FBD channels. The mapping of the data and ECC bits to the DRAM components and FBD channel bit lanes can enhance the protection provided by the ECC code to cover DRAM device failures and FBD channel bit lane failures. Refer to the Southbound Command+Wdata frame format and Northbound Data frame definitions for details.

5.5 Southbound Error Handling at the AMB

Errors in southbound frames are handled using the following method:

a) Check for CRC errors in the “A” command. If the AMB detects an error in the “A” command then discard the entire frame, and marks as faulted the commands or data from the previous frame. Process as command error:

   a. Log the error. The first CRC error latches the error data contents. The AMB will save the 72-bits plus CRC bits from the previous frame and the "A" command plus CRC from the current frame.
   b. Enter Command Error state: The AMB is forced to discard [all] subsequent commands until the channel is reset
   c. Indicate error by returning Alert frames.

b) Determine if the frame is a Command, or Command+Wdata frame; evaluate each command within a frame separately.

c) Check if the command is a Sync command. If Sync then respond with Status.

d) Check if the command is targeted for this AMB then process command.
   a. If the command is an unrecognizable command then ignore the command. The AMB is not expected to do DRAM protocol checking (e.g., looking for command conflicts such as a write interrupting a read, etc.)

e) Process the next command in the command frame if any are left.

f) Process next frame.

5.5.1 Exiting Command Error State

Once an AMB has entered the Command Error state it will no longer process commands other than the Soft Channel Reset command. Indication that the AMB is in the Command Error state is made manifest by the hardware setting of the appropriate configuration register bit and returning Alert frames. The AMB will continue to operate in this mode until a Soft Channel Reset command is received or the host resets the channel.

5.6 Northbound Error Handling at the AMB

The AMB does not evaluate the data and/or ECC information provided to it by the DRAM in response to a read command; it will forward the information supplied by the DRAM unchanged with a CRC for link error detection.

An AMB does not evaluate the data and/or ECC or CRC information passing through it from AMBs further south than it.

5.7 Northbound Error Handling at the Host

This section describes how FBD handles northbound read return data (and status responses) arriving at the host. **Section 5.7.1** describes the procedure for handling read return data; while **Section 5.7.2** presents the procedure for handling read return errors.

5.7.1 Read Return Burst Management

The host begins processing a read return burst when the first expected read return data frame arrives at its interface. Data collection continues on the next frame as the host receives the second read return data and continues until the read burst is completed.

The host checks for errors in the first and each subsequent data frame by evaluating CRC and ECC codes. The host logs any errors detected at this point for possible later error analysis by software or upstream hardware (see **Section 5.8.2**).

The host *may* choose to deliver the read return data for each frame as it is processed or it *may* hold the data until several or all data chunks have been collected and evaluated.

The data for a full read burst arrives in sequential frames. A framing error occurs if a status or idle frame is received prior to a full read burst being processed. In the case of such a framing error, the host, at a minimum, logs the error condition and begins looking for the next expected read return frame.

If the host completes the above steps without encountering a framing error, the host has received a *valid* read return data burst. Otherwise, the read return burst received by the host is *invalid*. Handling of an invalid read return burst is system dependent.

5.7.1.1 Determination that a Read Was Not Returned as Scheduled

The AMBs Idle frame is continuously being delivered by the most southern AMB on an FBD channel (*when* it is not responding with read data or status as scheduled and requested by the host).

In general, the host schedules the specific frame when a read burst is to be supplied by an AMB. If the AMB fails to provide read data in the specified frame then the host will instead receive an Idle or Alert frame and will then know that there has been a channel error.

If the host receives read return data with contents that match an Idle or Alert frame it is recommended but not required that the host take steps to make sure it actually received valid data. Because the Idle and Alert frame contents change each frame (according to a methodology that both the AMB and host comprehend) the host *may* reissue the read to return data in a frame that would contain a different Idle or Alert pattern to make sure it is receiving valid data. To make sure that the data frame contained valid data and not an Alert frame, the host *may* request a status response to check for errors, schedule a hole in the northbound data stream to check that a valid Idle frame is present, or re-read the data and verify that the data no longer matches the Alert frame permuting data pattern.

5.7.2 Read Response Data Error Handling

Host behavior for the processing of valid or invalid read responses is left up to the designers of the individual host implementations.

5.8 Error Logging

This section describes FBD error logging within the AMB and host.

5.8.1 Logging of AMB Southbound Events

If an error event occurs for southbound frames at the AMB, the AMB logs information in AMB error registers AMB_FERR or AMB_NERR. The AMB will also notify the host that an error has been observed, using the status response (sent in reply to a Sync command). The host may use this error indication to recover the logged information from the AMB using configuration access commands or to initiate some other error response activity.

5.8.2 Logging of Host Northbound Events

Details on host recording of errors observed in processing read, status, or idle frames is left up to the designers of the individual host implementations.

5.9 Error Injection

FBD naturally supports deliberate injection of data errors for debug and test purposes by way of allowing the host to alter write data and associated ECC in any way that it wants. Write data that has valid CRC will be written into DRAM. On Reads the data with its (purposefully corrupted) ECC will be forwarded by the AMB for evaluation by the host.

5.10 Fail-over Mode Operation

During channel initialization each bit lane is tested to determine if it is functioning properly. If one of the southbound bit lanes, northbound bit lanes, or one bit lane in both directions is non-functional, the redundant bit lane(s) may be used to map out the bad bit lane. Operation with the redundant bit lane used to map out a bad bit lane is described as "fail-over mode."

5.10.1 Fail-over Mode Operation on Southbound Lanes

Without the redundant bit lane used for CRC protection on the southbound lanes, commands continue to be protected by the 10-bit compound checksum CRC included with each command, DRAM write data continues to be protected by system level ECC data within the write data payload and the optional 12-bit CRC across each 72-bit data block, and the configuration register write data (within a Command+Data frame) continues to be protected by a 10-bit compound checksum CRC.

5.10.2 Fail-over Mode Operation on Northbound Lanes

The 14-lane northbound frame provides a 12-bit CRC over 72-bits of data in normal operation, and a 6- bit CRC over 72-bits of data in fail-over mode.

The 13-lane frames are without the redundant bit lane used for CRC protection. The read data continues to be protected by system level ECC data within the read data payload, and the status response continues to be covered by its 10-bit compound checksum CRC.

The 12-lane frames do not have a redundant bit lane for fail-over.

5.11 AMB Pass-through Functionality

As noted earlier much of the discussion regarding AMB behavior was from the viewpoint of having only a single AMB on the channel. FBD supports from one to eight DIMMs and several additional AMB components per channel.

As outlined in the protocol chapter, in terms of data movement an AMB is responsible for:

- Receiving southbound frames from the host or another AMB and in general re-driving those frames to a more southerly AMB.
- Evaluating southbound frames for commands or data targeted to that AMB and for checking all commands and data for errors.
- Receiving northbound frames from another AMB (generally) and re-driving those frames to another northerly AMB or to the host.
- Supplying frame content for read and status responses.

Each AMB must maintain the compound checksums used on the southbound channel. As can be seen from the four simple steps above an AMB does not check [for errors in] frames moving north that have been forwarded by another AMB, the frames are either discarded and replaced by frames from this AMB (if it is responsible for providing a read response), selectively overwritten by this AMB (if this AMB is providing a status response), or simply forwarded on to the next AMB or host. Because an AMB component does not evaluate data passing northbound through it, a read response or Idle frame being delivered by a more southerly AMB at the same time as this AMB is simply discarded without error notification.

If a given AMB is the last AMB (southern most AMB) it does not receive frames from the south and thus does not forward such frames in the northerly direction. It is responsible, however, for always generating Idle frames whenever it is not providing a read response or status response frame in response to a command from the host. These frames enable easy error detection by the host whenever a read return or status return is not provided by an AMB as scheduled by the host.

Particular attention is paid to the reliability of the pass-through logic. The logic is isolated from the rest of the internal AMB functions to ensure that the pass-through mechanism is functional even if other AMB functions have failed. This improves the reliability of the channel by minimizing the amount of logic that could result in a single point of failure.

5.12 Hot Add and Remove

Hot add and remove of system components is a common feature to enhance serviceability and provide improved system availability in the face of component failures. In the systems in which FBD is deployed it is expected that the granularity of hot add/remove may be at the granularity of individual DIMMs.

The hot add or remove of a DIMM involves the logical and physical addition or removal of an entire DIMM without interruption to the operation of northerly components. Removal of a DIMM will isolate DIMMs south of the DIMM removed. FBD provides mechanisms to assist in the hot add and remove operations. The Fast Reset mechanism defined in the Initialization Chapter is instrumental in the implementation of the hot add and remove feature.

This section outlines the major steps for hot add/remove. The type and number of configuration commands per step may depend on the AMB functionality and implementation, and on the particulars of the FBD topology.

5.12.1 Hot Add Sequence

FBD is especially well suited to adding new memory to the end of existing memory. The following sequence describes the events during a hot add.

Flow:

a) User appends a new component onto the FBD channel

b) User informs firmware that a component has been appended
Firmware powers up the appended component

c) 3.Firmware sends commands to the last AMB on the channel to calibrate the newly appended DIMM
Firmware programs internal AMB timing parameters through the SMbus.

d) 4.Firmware initiates a Fast Reset to bring the new DIMM into operation and retime the link

e) 5.When the FBD link completes initialization, the host may optionally interrupt the system firmware indicating that the FBD channel is now up or wait for the system firmware to poll its status register

f) 6.Firmware configures the host for what memory (logically) resides behind the newly appended DIMM.

5.12.2 Hot Remove Sequence

The following sequence describes the events during a hot remove.

Flow:

a) User informs system that he/she wants to remove a specific DIMM

b) System removes host address range from system map (if mirrored, remaps host address range to DIMM mirrors)

c) System copies or moves data from the host address range to other locations (if not already mirrored)

d) System polls until all outstanding transactions are completed

e) System updates the Last_AMB_ID value to the “new” last DIMM on the channel.

f) System initiates a Fast Reset to shut down the selected DIMM and the FBD channel interfaces for DIMM components attached to this DIMM

g) System disconnects power to the selected DIMM

h) System signals the user so they may now remove the selected DIMM

5.12.3 Hot Replace Sequence

The actions taken during the hot replace sequence of an AMB complex is accomplished with a combination of the hot remove and hot add sequences.

5.13 Memory Initialization

The AMB contains a memory built-in self-test (MEMBIST) engine that is used to test the DRAM devices on the DIMM and initialize the contents of the DRAM devices to a known state. Refer to the *FBD DFx specification* for details.

5.14 Thermal Trip Sensor

The AMB is outfitted with a thermal sensor that measures the temperature of the AMB die. A DAC and comparator mechanism driven from a Finite State Machine in the AMB periodically adjusts its value to indicate the temperature of the die. The temperature of the AMB die can be read at any time in the Thermal Sensor register. The Thermal Trip registers can be set to signal thermal warnings whenever the value of the Thermal Sensor register is higher than the Thermal Trip register trip points. The AMB provides the warning via bits in the status response that indicates if the thermal condition has been exceeded. Refer to the FBD AMB Specification for details.

Standard Improvement Form **JEDEC JESD206**

The purpose of this form is to provide the Technical Committees of JEDEC with input from the industry regarding usage of the subject standard. Individuals or companies are invited to submit comments to JEDEC. All comments will be collected and dispersed to the appropriate committee(s).

If you can provide input, please complete this form (it can be edited in Acrobat Exchange) and return to:

JEDEC
Attn: Publications Department
2500 Wilson Blvd. Suite 220
Arlington, VA 22201-3834
Fax: 703.907.7583

1. I recommend changes to the following:

☐ Requirement, paragraph number _____

☐ Test method number _____ Paragraph number _____

The referenced paragraph number has proven to be:

☐ Unclear ☐ Too Rigid ☐ In Error

☐ Other _____

2. Recommendations for correction:

_____

_____

_____

_____

3. Other suggestions for document improvement:

_____

_____

_____

_____

Submitted by

Name: _____ Phone: _____

Company: _____ E-mail: _____

Address: _____

City/State/Zip: _____ Date: _____

Rev. 9/02

JEDEC