

4.16.5 HDR: Header Type Register (D16, F2)

This register identifies the header layout of the configuration space.

Device: 16 Function: 2 Offset: 0Eh			
Bit	Attr	Default	Description
7	RO	1	Mfd: Multi-function Device Selects whether this is a multi-function device, that may have alternative configuration layouts.
6:0	RO	00h	CfgLayout: Configuration Layout This field identifies the format of the configuration header layout for a PCI-to-PCI bridge from bytes 10h through 3Fh. For this device default is 00h, indicating a conventional type 00h PCI header.

4.16.6 SVID: Subsystem Vendor Identification Register (D16, F2)

This register identifies the manufacturer of the system. This 16-bit register combined with the Device Identification Register uniquely identifies any PCI device.

Device: 16 Function: 2 Offset: 2Ch			
Bit	Attr	Default	Description
15:0	RWO	8086h	VIN: Vendor Identification Number The default value specifies Intel. Each byte of this register will be writable once. Second and successive writes to a byte will have no effect.

4.16.7 SID: Subsystem Identity (D16, F2)

This register identifies the system.

Device: 16 Function: 2 Offset: 2Eh - 2Fh			
Bit	Attr	Default	Description
15:0	RWO	8086h	SID: Subsystem Identification Number The default value specifies Intel. Each byte of this register will be writable once. Second and successive writes to a byte will have no effect.

4.16.8 GLOBAL_FERR: Global First Error Register (D16, F2)

When the “local” FERRST register captures the error, it closes the log register and signals the recorded error to the GLOBAL_FERR.

Device: 16 Function: 2 Offset: 40h - 43h			
Bit	Attr	Default	Description
31	RWCST	0	Global_FERR_31 Internal NB Fatal Error
30	RV	0	Reserved
29	RWCST	0	Global_FERR_29 FSB B Fatal Error
28	RWCST	0	Global_FERR_28 FSB A Fatal Error
27	RWCST	0	Global_FERR_27 IMI D Fatal Error
26	RWCST	0	Global_FERR_26 IMI C Fatal Error
25	RWCST	0	Global_FERR_25 IMI B Fatal Error
24	RWCST	0	Global_FERR_24 IMI A Fatal Error
23	RWCST	0	Global_FERR_23 PCI Express A1 (Device 7) Fatal Error
22	RWCST	0	Global_FERR_22 PCI Express A0 (Device 6) Fatal Error
21	RWCST	0	Global_FERR_21 PCI Express B1 (Device 5) Fatal Error
20	RWCST	0	Global_FERR_20 PCI Express B0 (Device 4) Fatal Error
19	RWCST	0	Global_FERR_19 PCI Express C1 (Device 3) Fatal Error
18	RWCST	0	Global_FERR_18 PCI Express C0 (Device 2) Fatal Error
17	RWCST	0	Global_FERR_17 PCI Express D (Device 1) Fatal Error
16	RWCST	0	Global_FERR_16 HI Fatal Error
15	RWCST	0	Global_FERR_15 Internal NB Non-Fatal Error
14	RV	0	Reserved

Device: 16 Function: 2 Offset: 40h - 43h			
Bit	Attr	Default	Description
13	RWCST	0	Global_FERR_13 FSB B Non-Fatal Error
12	RWCST	0	Global_FERR_12 FSB A Non-Fatal Error
11	RWCST	0	Global_FERR_11 IMI D Non-Fatal Error
10	RWCST	0	Global_FERR_10 IMI C Non-Fatal Error
9	RWCST	0	Global_FERR_09 IMI B Non-Fatal Error
8	RWCST	0	Global_FERR_08 IMI A Non-Fatal Error
7	RWCST	0	Global_FERR_07 PCI Express A1 (Device 7) Non-Fatal Error
6	RWCST	0	Global_FERR_06 PCI Express A0 (Device 6) Non-Fatal Error
5	RWCST	0	Global_FERR_05 PCI Express B1 (Device 5) Non-Fatal Error
4	RWCST	0	Global_FERR_04 PCI Express B0 (Device 4) Non-Fatal Error
3	RWCST	0	Global_FERR_03 PCI Express C1 (Device 3) Non-Fatal Error
2	RWCST	0	Global_FERR_02 PCI Express C0 (Device 2) Non-Fatal Error
1	RWCST	0	Global_FERR_01 PCI Express D (Device 1) Non-Fatal Error
0	RWCST	0	Global_FERR_00 HI Non-Fatal Error

4.16.9 GLOBAL_NERR: Global Next Error Register (D16, F2)

Once an error has been logged in the GLOBAL_FERR, subsequent errors are logged in the GLOBAL_NERR register.

Device: 16 Function: 2 Offset: 44h - 47h			
Bit	Attr	Default	Description
31	RWCST	0	Global_NERR_31 Internal NB Fatal Error
30	RV	0	Reserved
29	RWCST	0	Global_NERR_29 FSB B Fatal Error
28	RWCST	0	Global_NERR_28 FSB A Fatal Error
27	RWCST	0	Global_NERR_27 IMI D Fatal Error
26	RWCST	0	Global_NERR_26 IMI C Fatal Error
25	RWCST	0	Global_NERR_25 IMI B Fatal Error
24	RWCST	0	Global_NERR_24 IMI A Fatal Error
23	RWCST	0	Global_NERR_23 PCI Express A1 (Device 7) Fatal Error
22	RWCST	0	Global_NERR_22 PCI Express A0 (Device 6) Fatal Error
21	RWCST	0	Global_NERR_21 PCI Express B1 (Device 5) Fatal Error
20	RWCST	0	Global_NERR_20 PCI Express B0 (Device 4) Fatal Error
19	RWCST	0	Global_NERR_19 PCI Express C1 (Device 3) Fatal Error
18	RWCST	0	Global_NERR_18 PCI Express C0 (Device 2) Fatal Error
17	RWCST	0	Global_NERR_17 PCI Express D (Device 1) Fatal Error
16	RWCST	0	Global_NERR_16 HI Fatal Error
15	RWCST	0	Global_NERR_15 Internal NB Non-Fatal Error
14	RV	0	Reserved

Device: 16 Function: 2 Offset: 44h - 47h			
Bit	Attr	Default	Description
13	RWCST	0	Global_NERR_13 FSB B Non-Fatal Error
12	RWCST	0	Global_NERR_12 FSB A Non-Fatal Error
11	RWCST	0	Global_NERR_11 IMI D Non-Fatal Error
10	RWCST	0	Global_NERR_10 IMI C Non-Fatal Error
9	RWCST	0	Global_NERR_09 IMI B Non-Fatal Error
8	RWCST	0	Global_NERR_08 IMI A Non-Fatal Error
7	RWCST	0	Global_NERR_07 PCI Express A1 (Device 7) Non-Fatal Error
6	RWCST	0	Global_NERR_06 PCI Express A0 (Device 6) Non-Fatal Error
5	RWCST	0	Global_NERR_05 PCI Express B1 (Device 5) Non-Fatal Error
4	RWCST	0	Global_NERR_04 PCI Express B0 (Device 4) Non-Fatal Error
3	RWCST	0	Global_NERR_03 PCI Express C1 (Device 3) Non-Fatal Error
2	RWCST	0	Global_NERR_02 PCI Express C0 (Device 2) Non-Fatal Error
1	RWCST	0	Global_NERR_01 PCI Express D (Device1) Non-Fatal Error
0	RWCST	0	Global_NERR_00 HI Non-Fatal Error

4.16.10 EXSMRAMC - Expansion System Management RAM Control (D16, F2)

This register indicates whether an invalid SMRAM access has taken place. It is located in a different function than the other SMM related registers because it needs to be written by software.

Device: 16 Function: 2 Offset: 60h			
Bit	Attr	Default	Description
7	RWC	0	E_SMERR: Invalid SMRAM Access This bit is set when CPU has accessed the defined memory ranges in Extended SMRAM (High Memory and T-segment) while not in SMM space and with the D-OPEN bit = 0. It is software's responsibility to clear this bit. The software must write a 1 to this bit to clear it.
6:0	RV	0h	Reserved

4.16.11 FSB{A/B}_FERR: FSB First Error Register(D16, F2)

For a description of the Errors (Fxx) refer to Table 6-39 "Errors Detected by the NB" on page 328.

Device: 16 Function: 2 Offset: (80h - 81h), (A0h - A1h)			
Bit	Attr	Default	Description
15:11	RV	0h	Reserved
10	RWCST	0	F10Err: FSB hang Fatal
9	RWCST	0	F9Err: Data Parity Error (for 64-bit Intel® Xeon® processor MP with 1MB L2 cache & Dual-Core Intel® Xeon® processor 7000 sequence)
8	RWCST	0	F8Err: Unsupported Front Side Bus Transaction Fatal
7	RWCST	0	F7Err: FSB Protocol Error Fatal
6	RWCST	0	F6Err: Data Strobe Glitch Fatal
5	RWCST	0	F5Err: Address Strobe Glitch Fatal
4	RWCST	0	F4Err: Request/Address Parity Error Fatal

Device: 16 Function: 2 Offset: (80h - 81h), (A0h - A1h)			
Bit	Attr	Default	Description
3	RWCST	0	F3Err: Detected MCERR from a processor Nonfatal
2	RWCST	0	F2Err: Detected BINIT from a processor Nonfatal
1	RWCST	0	F1Err: Correctable ECC Error Nonfatal
0	RWCST	0	F0Err: Multi-Bit ECC Error Nonfatal

4.16.12 FSB{A/B}_NERR: FSB Next Fatal Error Register (D16, F2)

This register logs all FSB subsequent errors after the FERR_FSB has logged the 1st fatal error.

Device: 16 Function: 2 Offset: (82h - 83h), (A2 h- A3h)			
Bit	Attr	Default	Description
15:11	RV	0	Reserved
10	RWCST	0	F10Err: FSB hang
9	RWCST	0	F9Err: Data Parity Error (for 64-bit Intel® Xeon® processor MP with 1MB L2 cache & Dual-Core Intel® Xeon® processor 7000 sequence)
8	RWCST	0	F8Err: Unsupported Front Side Bus Transaction
7	RWCST	0	F7Err: FSB Protocol Error
6	RWCST	0	F6Err: Data Strobe Glitch
5	RWCST	0	F5Err: Address Strobe Glitch
4	RWCST	0	F4Err: Request/Address Parity Error
3	RWCST	0	F3NextErr: Detected MCERR
2	RWCST	0	F2NextErr: Detected BINIT
1	RWCST	0	F1NextErr: Correctable ECC Error
0	RWCST	0	F0NextErr: Multi-Bit ECC Error

4.16.13 RECFSB{A/B}_LOG: Recoverable FSB Error Log Register (D16, F2)

The following error log registers captures the syndrome information of the 32 bytes containing the data in which the data error was detected when a non-Fatal error is logged in the FSB_FERR register.

Device: 16 Function: 2 Offset: (84h - 87h), (A4h - A7h)			
Bit	Attr	Default	Description
31:24	ROST	0h	SYNDROME1: Syndrome bits of 3rd clock of data
23:16	ROST	0h	SYNDROME3: Syndrome bits of 4th clock of data
15:8	ROST	0h	SYNDROME0: Syndrome bits of 1st clock of data
7:0	ROST	0h	SYNDROME2: Syndrome bits of 2nd clock of data

4.16.14 NRECFSB{A/B}_LOG0: Non-recoverable FSB Error Log 0 Register (D16, F2)

FSB Log registers for non-recoverable errors when a fatal error is logged in its corresponding FSB_FERR Register.

Device: 16 Function: 2 Offset: (88h - 8Bh), (A8h - ABh)			
Bit	Attr	Default	Description
31:3	ROST	0h	ADDR: ADDR _A [31:3] fields of the FSB
2	ROST	0h	ADS: ADS field from the FSB
1:0	ROST	0h	RCV_PARITY: Address parity received

4.16.15 NRECFSB{A/B}_LOG1: Non-recoverable FSB Error Log 1 Register (D16, F2)

FSB Log registers for non-recoverable errors when a fatal error is logged in its corresponding FSB_FERR Register.

Device: 16 Function: 2 Offset: (8Ch - 8Fh), (ACh - AFh)			
Bit	Attr	Default	Description
31:15	ROST	0h	ADDRB : ADDRb[19:3] fields of the FSB
14	ROST	0	INFO
13	ROST	0h	DROVE_ADS : Indicates whether the NB drove this ADS
12:8	ROST	0h	REQA : REQa[4:0] fields of the FSB
7:0	ROST	0h	ADDR : ADDRa[39:32] fields of the FSB

4.16.16 NRECFSB{A/B}_LOG2: Non-recoverable FSB Error Log 2 Register (D16, F2)

FSB Log registers for non-recoverable errors when a fatal error is logged in its corresponding FSB_FERR Register.

Device: 16 Function: 2 Offset: (90h - 93h), (B0 - B3h)			
Bit	Attr	Default	Description
31:25	RV	0h	<i>Reserved</i>
24:20	ROST	0h	REQB : REQb[2:0] fields of the FSB
19:0	ROST	0h	ADDRB : ADDRb[39:20] fields of the FSB

4.16.17 EMASK_FSB{A/B}: FSB Error Mask Register (D16, F2)

A '0' in any field enables that error.

Device: 16 Function: 2 Offset: (94h - 95h), (B4 - B5h)			
Bit	Attr	Default	Description
15:11	RV	0h	Reserved
10	RW	1	F10Msk : FSB hang
9	RW	1	F9Msk : Data Parity Error (for 64-bit Intel® Xeon® processor MP with 1MB L2 cache &)Dual-Core Intel® Xeon® processor 7000 sequence
8	RW	1	F8Msk : Unsupported Front Side Bus Transaction
7	RW	1	F7Msk : FSB Protocol Error
6	RW	1	F6Msk : Data Strobe Glitch
5	RW	1	F5Msk : Address Strobe Glitch
4	RW	1	F4Msk : Request/Address Parity Error
3	RW	1	F3Msk : Detected MCERR
2	RW	1	F2Msk : Detected BINIT#
1	RW	1	F1Msk : Correctable ECC Error
0	RW	1	F0Msk : Multi-Bit ECC Error

4.16.18 ERR0_FSB{A/B}: FSB Error 0 Mask Register (D16, F2)

This register enables the signaling of Err[0] when an error flag is set. Note that one and only one error signal should be enabled ERR0_FSB, ERR1_FSB, ERR2_FSB, MCERR_FSB, and ICHRST# for each of the corresponding bits.

Device: 16 Function: 2 Offset: (96h - 97h), (B6h - B7h)			
Bit	Attr	Default	Description
15:11	RV	0h	Reserved
10	RWST	0	F10Err0 : FSB hang
9	RWST	0	F9Err0 : Data Parity Error (for 64-bit Intel® Xeon® processor MP with 1MB L2 cache & Dual-Core Intel® Xeon® processor 7000 sequence)
8	RWST	0	F8Err0 : Unsupported Front Side Bus Transaction
7	RWST	0	F7Err0 : FSB Protocol Error
6	RWST	0	F6Err0 : Data Strobe Glitch
5	RWST	0	F5Err0 : Address Strobe Glitch
4	RWST	0	F4Err0 : Request/Address Parity Error

Device: 16 Function: 2 Offset: (96h - 97h), (B6h - B7h)			
Bit	Attr	Default	Description
3	RWST	0	F3Err0: Detected MCERR
2	RWST	0	F2Err0: Detected BINIT#
1	RWST	0	F1Err0: Correctable ECC Error
0	RWST	0	F0Err0: Multi-Bit ECC Error

4.16.19 ERR1_FSB{A/B}: FSB Error 1 Mask Register (D16, F2)

This register enables the signaling of Err[1] when an error flag is set. Note that one and only one error signal should be enabled ERR0_FSB, ERR1_FSB, ERR2_FSB, MCERR_FSB, and ICHRST# for each of the corresponding bits.

Device: 16 Function: 2 Offset: (98h - 99h), (B8 - B9)			
Bit	Attr	Default	Description
15:11	RV	0h	Reserved
10	RWST	0	F10Err1: FSB hang
9	RWST	0	F9Err1: Data Parity Error (for 64-bit Intel® Xeon® processor MP with 1MB L2 cache & Dual-Core Intel® Xeon® processor 7000 sequence)
8	RWST	0	F8Err1: Unsupported Front Side Bus Transaction
7	RWST	0	F7Err1: FSB Protocol Error
6	RWST	0	F6Err1: Data Strobe Glitch
5	RWST	0	F5Err1: Address Strobe Glitch
4	RWST	0	F4Err1: Request/Address Parity Error
3	RWST	0	F3Err1: Detected MCERR
2	RWST	0	F2Err1: Detected BINIT#
1	RWST	0	F1Err1: Correctable ECC Error
0	RWST	0	F0Err1: Multi-Bit ECC Error

4.16.20 ERR2_FSB{A/B}: FSB Error 2 Mask Register (D16, F2)

This register enables the signaling of Err[2] when an error flag is set. Note that one and only one error signal should be enabled ERR0_FSB, ERR1_FSB, ERR2_FSB, MCERR_FSB, and ICHRST# for each of the corresponding bits.

Device: 16 Function: 2 Offset: (9Ah - 9Bh), (BAh - BBh)			
Bit	Attr	Default	Description
15:11	RV	0h	Reserved
10	RWST	0	F10Err2: FSB hang
9	RWST	0	F9Err2: Data Parity Error (for 64-bit Intel® Xeon® processor MP with 1MB L2 cache & Dual-Core Intel® Xeon® processor 7000 sequence)
8	RWST	0	F8Err2: Unsupported Front Side Bus Transaction
7	RWST	0	F7Err2: FSB Protocol Error
6	RWST	0	F6Err2: Data Strobe Glitch
5	RWST	0	F5Err2: Address Strobe Glitch
4	RWST	0	F4Err2: Request/Address Parity Error
3	RWST	0	F3Err2: Detected MCERR
2	RWST	0	F2Err2: Detected BINIT#
1	RWST	0	F1Err2: Correctable ECC Error
0	RWST	0	F0Err2: Multi-Bit ECC Error

4.16.21 MCERR_FSB{A/B}: FSB MCERR Mask Register (D16, F2)

This register enables the signaling of MCERR when an error flag is set. Note that one and only one error signal should be enabled ERR0_FSB, ERR1_FSB, ERR2, MCERR_FSB, and ICHRST# for each of the corresponding bits.

Device: 16 Function: 2 Offset: (9Ch - 9Dh), (BC - BDh)			
Bit	Attr	Default	Description
15:11	RV	0h	Reserved
10	RW	0	F10MCErr: FSB hang
9	RW	0	F9MCErr: Data Parity Error (for 64-bit Intel® Xeon® processor MP with 1MB L2 cache & Dual-Core Intel® Xeon® processor 7000 sequence)
8	RW	0	F8MCErr: Unsupported Front Side Bus Transaction
7	RW	0	F7MCErr: FSB Protocol Error
6	RW	0	F6MCErr: Data Strobe Glitch
5	RW	0	F5MCErr: Address Strobe Glitch

Device: 16 Function: 2 Offset: (9Ch - 9Dh), (BC - BDh)			
Bit	Attr	Default	Description
4	RW	0	F4MCErr: Request/Address Parity Error
3	RW	0	F3MCErr: Detected MCERR
2	RW	0	F2MCErr: Detected BINIT#
1	RW	0	F1MCErr: Correctable ECC Error
0	RW	0	F0MCErr: Multi-Bit ECC Error. Set for 64-bit Intel® Xeon® processor MP with 1MB L2 cache & Dual-Core Intel® Xeon® processor 7000 sequence

4.16.22 ICHRST_FSB{A/B}: FSB ICHRST Mask Register (D16, F2)

This register enables the signaling of ICHRST# when an error flag is set. Note that one and only one error signal should be enabled ERR0_FSB, ERR1_FSB, ERR2, MCERR_FSB, and ICHRST# for each of the corresponding bits.

Device: 16 Function: 2 Offset: BEh, 9Eh			
Bit	Attr	Default	Description
15:11	RV	0h	Reserved
10	RW	0	F10Rst: FSB hang
9	RW	0	F9Rst: Data Parity Error (for 64-bit Intel® Xeon® processor MP with 1MB L2 cache & Dual-Core Intel® Xeon® processor 7000 sequence)
8	RW	0	F8Rst: Unsupported Front Side Bus Transaction
7	RW	0	F7Rst: FSB Protocol Error
6	RW	0	F6Rst: Data Strobe Glitch
5	RW	0	F5Rst: Address Strobe Glitch
4	RW	0	F4Rst: Request/Address Parity Error
3	RW	0	F3Rst: Detected MCERR
2	RW	0	F2Rst: Detected BINIT#
1	RW	0	F1Rst: Correctable ECC Error
0	RW	0	F0Rst: Multi-Bit ECC Error

4.16.23 INT_FERR: Internal First Fatal Error Register (D16, F2)

INT_FERR latches the first NB internal error, whether fatal or non-fatal. All subsequent errors get logged in the INT_NERR.

Device: 16 Function: 2 Offset: C1h			
Bit	Attr	Default	Description
7	RWCST	0	T2Err: Uncorrectable Memory Error Non-fatal Error Signaled by the NB after a request returns uncorrectable data at least 2 times from an IMI. This error is signaled only after any reconstruction efforts made by RAID or Mirroring logic (if enabled) have also attempted to reconstruct the data. Since poison is maintained, this error is considered non-fatal.
6	RWCST	0	T4Err: Coherency Violation Error Fatal
5	RWCST	0	T3Err: Multi-Tag Hit from CDC Fatal Not detected.
4	RWCST	0	T8Err: Poison leaving a poison domain Fatal An uncorrectable error marked as poison is leaving an error detection domain and no longer has an indication that the data is incorrect.
3	RWCST	0	T7Err: UnCorrectable Data Error during a Re-silvering Process Non-fatal
2	RWCST	0	T6Err: Poison being passed on to another poison capable domain Non-fatal
1	RWCST	0	T5Err: Address Map Error Non-fatal
0	RWCST	0	T1Err: Correctable ECC Error Non-fatal

4.16.24 INT_NERR: Internal Next Fatal Error Register (D16, F2)

For details on errors listed on this table, refer to the Table 6-39 “Errors Detected by the NB” on page 328.

Device: 16 Function: 2 Offset: C3h			
Bit	Attr	Default	Description
7	RWCST	0	T2Err: Uncorrectable Memory Error
6	RWCST	0	T4NextErr: Coherency Violation Error Fatal
5	RWCST	0	T3NextErr: Multi-Tag Hit from CDC Fatal. Not detected.
4	RWCST	0	T8NextErr: Poison leaving a poison domain
3	RWCST	0	T7NextErr: UnCorrectable Data Error during a Re-silvering Process Non-fatal
2	RWCST	0	T6Err: Poison being passed on to another poison capable domain Non-fatal
1	RWCST	0	T5NextErr: Address Map Error Non-fatal
0	RWCST	0	T1NextErr: Correctable ECC Error Non-fatal

4.16.25 NRECNCB: Non-recoverable NB Error Log Register (D16, F2)

Device: 16 Function: 2 Offset: C4h - C7h			
Bit	Attr	Default	Description
31:7	RV	0000h	Reserved
6:0	ROST	0h	FailedCdcId: CDC Entry of Failed Location

4.16.26 RECNB: Recoverable NB Data Log Register (D16, F2)

Device: 16 Function: 2 Offset: C8h - CBh			
Bit	Attr	Default	Description
31:7	RV	0h	Reserved
6:0	ROST	0h	FailedCdclid: CDC Entry of Failed Location

4.16.27 EMASK_INT: Internal Error Mask Register (D16, F2)

For details on errors listed on this table, refer to the Table 6-39 “Errors Detected by the NB” on page 328. A ‘0’ in any field enables that error.

Device: 16 Function: 2 Offset: CCh			
Bit	Attr	Default	Description
7	RW	1	T2Msk: Uncorrectable Memory Error Signaled by the NB after a request returns uncorrectable data at least 2 times from an IMI. This error is signaled only after any reconstruction efforts made by RAID or Mirroring logic (if enabled) have also attempted to reconstruct the data. Since poison is maintained, this error is considered non-fatal.
6	RW	1	T4Msk: Coherency Violation Error
5	RV	0	Reserved
4	RW	1	T8Msk: Poison leaving a poison domain Must be set if ECC disabled. Set for 64-bit Intel® Xeon® processor MP with 1MB L2 cache & Dual-Core Intel® Xeon® processor 7000 sequence .
3	RW	1	T7Msk: UnCorrectable Data Error during a Re-silvering Process
2	RW	1	T6Msk: Poison being passed on to another poison capable domain Must be set if ECC disabled. Set for 64-bit Intel® Xeon® processor MP with 1MB L2 cache & Dual-Core Intel® Xeon® processor 7000 sequence
1	RW	1	T5Msk: Address Map Error
0	RW	1	T1Msk: Correctable ECC Error Must be set if ECC disabled. Set for 64-bit Intel® Xeon® processor MP with 1MB L2 cache & Dual-Core Intel® Xeon® processor 7000 sequence .

4.16.28 ERR0_INT: Internal Error 0 Mask Register (D16, F2)

This register enables the signaling of Err[0] when an error flag is set. Note that one and only one error signal should be enabled. Note that one and only one error signal should be enabled in the ERR0_INT, ERR1_INT, ERR2_INT, and MCERR_INT for each of the corresponding bits.

Device: 16 Function: 2 Offset: D0h			
Bit	Attr	Default	Description
7	RWST	0	T2Err0: Uncorrectable Memory Error
6	RWST	0	T4Err0: Coherency Violation Error
5	RV	0	Reserved
4	RWST	0	T8Err0: Poison leaving a poison domain
3	RWST	0	T7Err0: UnCorrectable Data Error during a Re-silvering Process
2	RWST	0	T6Err: Poison being passed on to another poison capable domain
1	RWST	0	T5Err0: Address Map Error
0	RWST	0	T1Err0: Correctable ECC Error

4.16.29 ERR1_INT: Internal Error 1 Mask Register (D16, F2)

This register enables the signaling of Err[1] when an error flag is set. Note that one and only one error signal should be enabled. Note that one and only one error signal should be enabled in the ERR0_INT, ERR1_INT, ERR2_INT, and MCERR_INT for each of the corresponding bits.

Device: 16 Function: 2 Offset: D1h			
Bit	Attr	Default	Description
7	RWST	0	T2Err1: Uncorrectable Memory Error
6	RWST	0	T4Err1: Coherency Violation Error
5	RWST	0	T3Err1: Multi-Tag Hit from CDC
4	RWST	0	T8Err1: Poison leaving a poison domain
3	RWST	0	T7Err1: UnCorrectable Data Error during a Re-silvering Process
2	RWST	0	T6Err: Poison being passed on to another poison capable domain
1	RWST	0	T5Err1: Address Map Error
0	RWST	0	T1Err1: Correctable ECC Error

4.16.30 ERR2_INT: Internal Error 2 Mask Register (D16, F2)

This register enables the signaling of Err[2] when an error flag is set. Note that one and only one error signal should be enabled. Note that one and only one error signal should be enabled in the ERR0_INT, ERR1_INT, ERR2_INT, and MCERR_INT for each of the corresponding bits.

Device: 16 Function: 2 Offset: D2h			
Bit	Attr	Default	Description
7	RWST	0	T2Err2: Uncorrectable Memory Error
6	RWST	0	T4Err2: Coherency Violation Error
5	RWST	0	T3Err2: Multi-Tag Hit from CDC
4	RWST	0	T8Err2: Poison leaving a poison domain
3	RWST	0	T7Err2: UnCorrectable Data Error during a Re-silvering Process
2	RWST	0	T6Err: Poison being passed on to another poison capable domain
1	RWST	0	T5Err2: Address Map Error
0	RWST	0	T1Err2: Correctable ECC Error

4.16.31 MCERR_INT: Internal MCERR Mask Register (D16, F2)

This register enables the signaling of MCERR when an error flag is set. Note that one and only one error signal should be enabled. Note that one and only one error signal should be enabled in the ERR0_INT, ERR1_INT, ERR2_INT, and MCERR_INT for each of the corresponding bits.

Device: 16 Function: 2 Offset: D3h			
Bit	Attr	Default	Description
7	RWST	0	T2MCErr: Uncorrectable Memory Error
6	RWST	0	T4MCErr: Coherency Violation Error
5	RWST	0	T3MCErr: Multi-Tag Hit from CDC
4	RWST	0	T8MCErr: Poison leaving a poison domain
3	RWST	0	T7MCErr: UnCorrectable Data Error during a Re-silvering Process
2	RWST	0	T6MCErr: Poison being passed on to another poison capable domain
1	RWST	0	T5MCErr: Address Map Error
0	RWST	0	T1MCErr: Correctable ECC Error

4.16.32 RECINT_LOG0: Recoverable Internal Error Log 0 Register (D16, F2)

The following error log registers capture information about the transaction or access that encountered an internal NB error when a non-Fatal or a Fatal error is logged in the INT_FERR or when certain errors are logged in the IMI_FERR or IMI_NERR registers.

Device: 16 Function: 2 Offset: D8h - DBh			
Bit	Attr	Default	Description
31:6	RWST	0h	ADDR31T6: Bits [31:6] of the address
5:2	RWST	0h	ADDR5T2: Bits [5:2] of the address
1	RWST	0	WRITE: '1' if the access was a write
0	RWST	0	DRAM: '1' if the access mapped to DRAM

4.16.33 RECINT_LOG1: Recoverable Internal Error Log 1 Register (D16, F2)

The following error log registers captures information about the transaction or access that encountered an internal NB error when a non-Fatal or a Fatal error is logged in the INT_FERR.

Device: 16 Function: 2 Offset: DCh - DFh			
Bit	Attr	Default	Description
31:24	RWST	0h	TAGRAM Zero padded, cpip_rq4_readinfo_ff[top:96]
23	RWST	0	GOTDATA Internal indication of the status of the line in the internal cache.
22	RWST	0	NP: Non-Posted '1' if non-posted.
21	RWST	0	BURN: '1' if the entry was bypassed from an FSB but not used.
20	RV	0	Reserved
19:16	RWST	0h	IWB_CNT: How many IWB's were still needed.
15	RV	0	Reserved
14:8	RWST	0h	CDCID
7:0	RWST	0	ADDR39T32: Bits [39:32] of the address

4.16.34 **RECINT_LOG2: Recoverable Internal Error Log 2 Register (D16, F2)**

The following error log registers captures information about the transaction or access that encountered an internal NB error when a non-Fatal or a Fatal error is logged in the INT_FERR.

Device: 16 Function: 2 Offset: E0h - E3h			
Bit	Attr	Default	Description
31:0	RWST	0000h	TAGRAM cpip_rq4_readinfo_ff[31:0]

4.16.35 **RECINT_LOG3: Recoverable Internal Error Log 3 Register (D16, F2)**

The following error log registers captures information about the transaction or access that encountered an internal NB error when a non-Fatal or a Fatal error is logged in the INT_FERR.

Device: 16 Function: 2 Offset: E4h - E7h			
Bit	Attr	Default	Description
31:0	RWST	0000h	TAGRAM cpip_rq4_readinfo_ff[63:32]

4.16.36 **RECINT_LOG4: Recoverable Internal Error Log 4 Register (D16, F2)**

The following error log registers captures information about the transaction or access that encountered an internal NB error when a non-Fatal or a Fatal error is logged in the INT_FERR.

Device: 16 Function: 2 Offset: E8h - EBh			
Bit	Attr	Default	Description
31:0	RWST	0000h	TAGRAM cpip_rq4_readinfo_ff[95:64]

4.16.37 INTLOGC: Internal Error Log Control (D16, F2)

The following error log registers controls error logging of the internal error log registers RECINT_LOG[4:0].

Device: 16 Function: 2 Offset: ECh - EFh			
Bit	Attr	Default	Description
31	RWST	0	Valid: '1' if the error logs are valid
30:24	RV	0	Reserved
23	RO	0	TRIG: Set if a log is currently being made. (Logs are expected to change in a few clocks.)
22:16	RO	0	TRIG_CDCID: If set, the bit indicates the CDCID that will be logged. Could be inaccurate if a previously logged recoverable error exists, but the current logging is being done for a non-recoverable error.
15	RWST	0	NREC: Enables non-recoverable error logging. Cleared by hardware after logging a non-recoverable error and REARM is clear.
14	RWST	0	REC: Enables recoverable error logging. Cleared by hardware after logging a recoverable error and REARM is clear.
13	RWST	0	DIAG: Enables diagnostic logging. Cleared by hardware when logging finished.
12	RWST	0	REARM: When clear, hardware will log one error of one type, then clear the appropriate enable bit and not log further errors of the same severity.
11:8	RWST	0	GLOB_EV: Set to enable logging on a global event. Hardware clears on any of global event, recoverable, or non-recoverable error logged unless REARM set.
7	RWST	0	DEBUG: Set to cause information from the DEBUG_CDCID from CCAM to be logged instead information from the INTLOGC.CDCID.
6:0	RWST	0	CDCID: CDCID to log information for when DEBUG field clear.

4.17 Miscellaneous (Device 17, Function 0)

4.17.1 VID: Vendor Identification Register (D17, F0)

This register identifies Intel as the manufacturer of the NB. Writes to this register have no effect.

Device 17 Function: 0 Offset: 00 - 01h			
Bit	Attr	Default	Description
15:0	RO	8086h	Vendor Identification Number The value assigned to Intel.

4.17.2 DID: Device Identification Register (D17, F0)

This register combined with the Vendor Identification register uniquely identifies the NB Function in the event that a driver is required. Writes to this register have no effect.

Device 17 Function: 0 Offset: 02 - 03h			
Bit	Attr	Default	Description
15:0	RO	2613h	Device Identification Number Identifies each function of the NB.

4.17.3 RID: Revision Identification Register (D17, F0)

This register contains the revision number of the Intel® E8501 chipset North Bridge (NB).

Device 17 Function: 0 Offset: 08h			
Bit	Attr	Default	Description
7:4	RO	0h	Major_rev: Major Revision Steppings which require all masks to be regenerated. 1: B stepping
3:0	RO	0h	Minor_rev: Minor Revision Incremented for each stepping which does not modify all masks. Reset for each major revision. 0: x0 stepping 1: x1 stepping 2: x2 stepping

4.17.4 CCR: Class Code Register (D17, F0)

This register contains the Class Code for the device.

Device 17 Function: 0 Offset: 09 - 0Bh			
Bit	Attr	Default	Description
23:16	RO	06h	BaseClass: Base Class. This field indicates the general device category. For the NB, this field is hardwired to 06h, indicating it is a "Bridge Device".
15:8	RO	00h	SubClass: Sub-Class. This field qualifies the Base Class, providing a more detailed specification of the device function. For this device default is 00h, indicating "Host Bridge".
7:0	RO	00h	RLPI: Register-Level Programming Interface. This field identifies a specific programming interface (if any), that device independent software can use to interact with the device. There are no such interfaces defined for "Host Bridge" types, and this field is hardwired to 00h.

4.17.5 HDR: Header Type Register (D17, F0)

This register identifies the header layout of the configuration space.

Device 17 Function: 0 Offset: 0Eh			
Bit	Attr	Default	Description
7	RO	1	Mfd: Multi-function Device. Selects whether this is a multi-function device, that may have alternative configuration layouts.
6:0	RO	00h	CfgLayout: Configuration Layout. This field identifies the format of the configuration header layout for a PCI-to-PCI bridge from bytes 10h through 3Fh. For this device default is 00h, indicating a conventional type 00h PCI header.

4.17.6 SVID: Subsystem Vendor Identification Register (D17, F0)

This register identifies the manufacturer of the system. This 16-bit register combined with the Device Identification Register uniquely identifies any PCI device.

Device: 17 Function: 0 Offset: 2Ch			
Bit	Attr	Default	Description
15:0	RWO	8086h	VIN: Vendor Identification Number The default value specifies Intel. Each byte of this register will be writable once. Second and successive writes to a byte will have no effect.

4.17.7 SID: Subsystem Identity (D17, F0)

This register identifies the system.

Device: 17 Function: 0 Offset: 2Eh - 2Fh			
Bit	Attr	Default	Description
15:0	RWO	8086h	SID: Subsystem Identification Number The default value specifies Intel. Each byte of this register will be writable once. Second and successive writes to a byte will have no effect.

4.17.8 FSBDC[A]: Front Side Bus Data Control (D17, F0)

This register controls the Front Side Buses. There is one register for each bus and they must be programmed identically.

Device: 17 Function: 0 Offset: A4h			
Bit	Attr	Default	Description
31:5	RV	XX	Reserved.
4	RW	0	POISONON: Enable poison generation. Must be set for 64-bit Intel® Xeon® processor MP with up to 8MB L3 cache

Device: 17 Function: 0 Offset: A4h			
Bit	Attr	Default	Description
3	RW	0	PARCHKEN: 1: Enable DP[3:0] checking. Must be set for 64-bit Intel® Xeon® processor MP with 1MB L2 cache & Dual-Core Intel® Xeon® processor 7000 sequence support.
2	RW	0	ECCCHKEN: 1: Enable ECC check and correct. Must be set for 64-bit Intel® Xeon® processor MP with up to 8MB L3 cache.
1	RW	1	DRVECC: 1: Drive DEP[7:0] onto FSB. Must be set for 64-bit Intel® Xeon® processor MP with up to 8MB L3 cache.
0	RW	0	DRVPAR: 1: Drive DP[3:0] parity bits onto FSB. Must be set for 64-bit Intel® Xeon® processor MP with 1MB L2 cache & Dual-Core Intel® Xeon® processor 7000 sequence support.

4.17.9 FSB[A]AC2: Front Side Bus Control2 (D17,F0)

This is the second register for controlling the Front Side Buses. There is one register for each bus and they must be programmed identically.

Device: 17 Function: 0 Offset: B0h			
Bit	Attr	Default	Description
31:24	RV	xxx	Reserved
23:20	RW	6h	CMPL_LIMIT: CMPLQ high water mark Suggested Value: 5h
19:0	RV	xxx	Reserved

4.17.10 FSB[A]AC: Front Side Bus Control (D17, F0)

This register controls the Front Side Buses. There is one register for each bus and they must be programmed identically.

Device: 17 Function: 0 Offset: B8h			
Bit	Attr	Default	Description
31:27	RV	XX	Reserved.
26	RW	0	CPUREQ_EXT: BPRI Optimization Enable 1: Enables CPU to get another transaction onto the bus if a BREQ change occurred. Helps get more BWL/BWL back to back cases possible. Suggested Value: 1 for 64-bit Intel® Xeon® processor MP, 0 for Dual-Core Intel® Xeon® processor 7000 sequence
25:15	RV	XX	Reserved.
14	RW	0	ARB_ADS: ADS Arbitration Optimization Enable 0: The NB doesn't look at ADS to BPRI the bus faster. 1: The NB can take ADS into account to BPRI the bus faster. Suggested Value: 1
13	RW	0	ARB_BREQ: Arbitration Streaming Optimization Enable 0: The NB will not use the BREQ# pins to drive ADS# early. 1: The NB will observe BREQ# to determine when the processor is requesting the bus. When no processor is requesting the bus, the NB (priority agent) may drive ADS# one cycle after it drives BPRI#. Suggested Value: 1 for 64-bit Intel® Xeon® processor MP, 0 for Dual-Core Intel® Xeon® processor 7000 sequence
12:8	RV	XX	Reserved.
7	RW	0	DPSDIS: Disable Enhanced Defer 0: The NB will listen to DPS# and can complete transactions with IDS#. Only if strapped for 64-bit Intel® Xeon® processor MP with up to 8MB L3 cache 1: The NB will complete deferred transactions with Deferred Reply rather than IDS#
6	RW	0	MEDEN: Modified Enhanced Defer Enable 0: The NB will assert OOD# at the same time as IDS#. 1: The NB may assert OOD# later than IDS#. Should be set for 64-bit Intel® Xeon® processor MP with up to 8MB L3 cache
5:0	RW	20h	CRQDEPTH: Snoop Request Queue Depth The NB will assert flow control when the CRQ reaches this threshold. Suggested Value: 10h

4.18 Miscellaneous (Device 17, Function 1)

4.18.1 VID: Vendor Identification Register (D17, F1)

This register identifies Intel as the manufacturer of the NB. Writes to this register have no effect.

Device 17 Function: 1 Offset: 00 - 01h			
Bit	Attr	Default	Description
15:0	RO	8086h	Vendor Identification Number The value assigned to Intel.

4.18.2 DID: Device Identification Register (D17, F1)

This register combined with the Vendor Identification register uniquely identifies the NB Function in the event that a driver is required. Writes to this register have no effect.

Device 17 Function: 1 Offset: 02 - 03h			
Bit	Attr	Default	Description
15:0	RO	2614h	Device Identification Number Identifies each function of the NB.

4.18.3 RID: Revision Identification Register (D17, F1)

This register contains the revision number of the Intel® E8501 chipset North Bridge (NB).

Device 17 Function: 1 Offset: 08h			
Bit	Attr	Default	Description
7:4	RO	0h	Major_rev: Major Revision Steppings which require all masks to be regenerated. 1: B stepping
3:0	RO	0h	Minor_rev: Minor Revision Incremented for each stepping which does not modify all masks. Reset for each major revision. 0: x0 stepping 1: x1 stepping 2: x2 stepping

4.18.4 CCR: Class Code Register (D17, F1)

This register contains the Class Code for the device.

Device: 17 Function: 1 Offset: 09 - 0Bh			
Bit	Attr	Default	Description
23:16	RO	06h	BaseClass: Base Class This field indicates the general device category. For the NB, this field is hardwired to 06h, indicating it is a "Bridge Device".
15:8	RO	00h	SubClass: Sub-Class This field qualifies the Base Class, providing a more detailed specification of the device function. For this device default is 00h, indicating "Host Bridge".
7:0	RO	00h	RLPI: Register-Level Programming Interface This field identifies a specific programming interface (if any), that device independent software can use to interact with the device. There are no such interfaces defined for "Host Bridge" types, and this field is hardwired to 00h.

4.18.5 HDR: Header Type Register (D17, F1)

This register identifies the header layout of the configuration space.

Device: 17 Function: 1 Offset: 0Eh			
Bit	Attr	Default	Description
7	RO	1	Mfd: Multi-function Device Selects whether this is a multi-function device, that may have alternative configuration layouts.
6:0	RO	00h	CfgLayout: Configuration Layout This field identifies the format of the configuration header layout for a PCI-to-PCI bridge from bytes 10h through 3Fh. For this device default is 00h, indicating a conventional type 00h PCI header.

4.18.6 SVID: Subsystem Vendor Identification Register (D17, F1)

This register identifies the manufacturer of the system. This 16-bit register combined with the Device Identification Register uniquely identifies any PCI device.

Device: 17 Function: 1 Offset: 2Ch			
Bit	Attr	Default	Description
15:0	RWO	8086h	VIN: Vendor Identification Number The default value specifies Intel. Each byte of this register will be writable once. Second and successive writes to a byte will have no effect.

4.18.7 SID: Subsystem Identity (D17, F1)

This register identifies the system.

Device: 17 Function: 1 Offset: 2Eh - 2Fh			
Bit	Attr	Default	Description
15:0	RWO	8086h	SID: Subsystem Identification Number The default value specifies Intel. Each byte of this register will be writable once. Second and successive writes to a byte will have no effect.

4.18.8 FSBDC[B]: Front Side Bus Data Control (D17, F1)

This register controls the Front Side Buses. There is one register for each bus and they must be programmed identically.

Device: 17 Function: 1 Offset: A4h			
Bit	Attr	Default	Description
31:5	RV	XX	Reserved.
4	RW	0	POISONON: Enable poison generation. Must be set for 64-bit Intel® Xeon® processor MP with up to 8MB L3 cache.
3	RW	0	PARCHKEN: 1: Enable DP[3:0] checking. Must be set for 64-bit Intel® Xeon® processor MP with 1MB L2 cache & Dual-Core Intel® Xeon® processor 7000 sequence support.
2	RW	0	ECCCHKEN: 1: Enable ECC check and correct. Must be set for 64-bit Intel® Xeon® processor MP with up to 8MB L3 cache.
1	RW	1	DRVECC: 1: Drive DEP[7:0] onto FSB. Must be set for 64-bit Intel® Xeon® processor MP with up to 8MB L3 cache.
0	RW	0	DRVPAR: 1: Drive DP[3:0] parity bits onto FSB. Must be set for 64-bit Intel® Xeon® processor MP with 1MB L2 cache & Dual-Core Intel® Xeon® processor 7000 sequence support.

4.18.9 FSB[B]AC2: Front Side Bus Control2 (D17, F1)

This is the second register for controlling the Front Side Buses. There is one register for each bus and they must be programmed identically.

Device: 17 Function: 1 Offset: B0h			
Bit	Attr	Default	Description
31:24	RV	xxx	<i>Reserved</i>
23:20	RW	6h	CMPL_LIMIT: CMPLQ high water mark. Suggested Value: 5h
19:0	RV	xxx	<i>Reserved</i>

4.18.10 FSB[B]AC: Front Side Bus Control (D17, F1)

This register controls the Front Side Buses. There is one register for each bus and they must be programmed identically.

Device: 17 Function: 1 Offset: B8h			
Bit	Attr	Default	Description
31:27	RV	XX	<i>Reserved.</i>
26	RW	0	CPUREQ_EXT: BPRI Optimization Enable 1: Enables CPU to get another transaction onto the bus if a BREQ change occurred. Helps get more BWL/BWL back to back cases possible. Suggested Value: 1
25:15	RV	XX	<i>Reserved.</i>
14	RW	0	ARB_ADS: ADS Arbitration Optimization Enable 0: The NB doesn't look at ADS to BPRI the bus faster. 1: The NB can take ADS into account to BPRI the bus faster. Suggested Value: 1
13	RW	0	ARB_BREQ: Arbitration Streaming Optimization Enable 0: The NB will not use the BREQ# pins to drive ADS# early. 1: The NB will observe BREQ# to determine when the processor is requesting the bus. When no processor is requesting the bus, the NB (priority agent) may drive ADS# one cycle after it drives BPRI#. Suggested Value: 1

Device: 17 Function: 1 Offset: B8h			
Bit	Attr	Default	Description
12:8	RV	XX	Reserved.
7	RW	0	DPSDIS: Disable Enhanced Defer 0: The NB will listen to DPS# and can complete transactions with IDS#. Only if strapped for 64-bit Intel® Xeon® processor MP with up to 8MB L3 cache. 1: The NB will complete deferred transactions with Deferred Reply rather than IDS#.
6	RW	0	MEDEN: Modified Enhanced Defer Enable 0: The NB will assert OOD# at the same time as IDS#. 1: The NB may assert OOD# later than IDS#. Should be set for 64-bit Intel® Xeon® processor MP with up to 8MB L3 cache.
5:0	RW	2Dh	CRQDEPTH: Snoop Request Queue Depth The NB will assert flow control when the CRQ reaches this threshold. Suggested Value: 1Dh

4.18.11 EXP_GCTRL: PCI Express Global Control Register

This 32-bit global register in the NB implements chipset specific operations for generalized control of all PCI Express events and activity such as Power Management. There is only one register for all PCI Express ports.

Device: 17 Function: 1 Offset: DCh			
Bit	Attr	Default	Description
31:16	RV	0	Reserved.
15:2	RV	3FFFh	Reserved.
1	RWST	0	PME_TURN_OFF: Send PME Turn Off Message^a When set, the NB will issue a PME Turn Off Message to all enabled PCI Express ports. The NB will clear this bit once the Message is sent.
0	RWC	0	PME_TO_ACK: Received PME Time Out Acknowledge Message^a The NB sets this bit when it receives a PME TO_ACK Message from all enabled PCI Express ports. Software will clear this bit when it handles the Acknowledge

NOTES:

- Note that PME_TURN_OFF and PME_TO_ACK bits are defined in the EXP_GCTRL register (global) since the intent was to have one and only one of these status bits unique for all the seven PCI Express ports in the NB.

4.19 Miscellaneous (Device 17, Function 2)

4.19.1 VID: Vendor Identification Register (D17, F2)

This register identifies Intel as the manufacturer of the NB. Writes to this register have no effect.

Device: 17 Function: 2 Offset: 00 - 01h			
Bit	Attr	Default	Description
15:0	RO	8086h	Vendor Identification Number The value assigned to Intel.

4.19.2 DID: Device Identification Register (D17, F2)

This register combined with the Vendor Identification register uniquely identifies the NB Function in the event that a driver is required. Writes to this register have no effect.

Device: 17 Function: 2 Offset: 02 - 03h			
Bit	Attr	Default	Description
15:0	RO	2615h	Device Identification Number Identifies each function of the NB.

4.19.3 RID: Revision Identification Register (D17, F2)

This register contains the revision number of the Intel® E8501 chipset North Bridge (NB).

Device: 17 Function: 2 Offset: 08h			
Bit	Attr	Default	Description
7:4	RO	0h	Major_rev: Major Revision Steppings which require all masks to be regenerated. 1: B stepping
3:0	RO	0h	Minor_rev: Minor Revision Incremented for each stepping which does not modify all masks. Reset for each major revision. 0: x0 stepping 1: x1 stepping 2: x2 stepping

4.19.4 CCR: Class Code Register (D17, F2)

This register contains the Class Code for the device.

Device 17 Function: 2 Offset: 09 - 0Bh			
Bit	Attr	Default	Description
23:16	RO	06h	BaseClass: Base Class This field indicates the general device category. For the NB, this field is hardwired to 06h, indicating it is a "Bridge Device".
15:8	RO	00h	SubClass: Sub-Class This field qualifies the Base Class, providing a more detailed specification of the device function. For this device default is 00h, indicating "Host Bridge".
7:0	RO	00h	RLPI: Register-Level Programming Interface This field identifies a specific programming interface (if any), that device independent software can use to interact with the device. There are no such interfaces defined for "Host Bridge" types, and this field is hardwired to 00h.

4.19.5 HDR: Header Type Register (D17, F2)

This register identifies the header layout of the configuration space.

Device 17 Function: 2 Offset: 0Eh			
Bit	Attr	Default	Description
7	RO	1	Mfd: Multi-function Device Selects whether this is a multi-function device, that may have alternative configuration layouts.
6:0	RO	00h	CfgLayout: Configuration Layout This field identifies the format of the configuration header layout for a PCI-to-PCI bridge from bytes 10h through 3Fh. For this device default is 00h, indicating a conventional type 00h PCI header.

4.19.6 SVID: Subsystem Vendor Identification Register (D17, F2)

This register identifies the manufacturer of the system. This 16-bit register combined with the Device Identification Register uniquely identifies any PCI device.

Device 17 Function: 2 Offset: 2Ch			
Bit	Attr	Default	Description
15:0	RWO	8086h	VIN: Vendor Identification Number The default value specifies Intel. Each byte of this register will be writable once. Second and successive writes to a byte will have no effect.

4.19.7 SID: Subsystem Identity (D17, F2)

This register identifies the system.

Device 17 Function: 2 Offset: 2Eh - 2Fh			
Bit	Attr	Default	Description
15:0	RWO	8086h	SID: Subsystem Identification Number The default value specifies Intel. Each byte of this register will be writable once. Second and successive writes to a byte will have no effect.

4.19.8 IMI_HPTIM: Independent Memory Interface Hot Plug Timer (D17, F2)

This register is reserved for IMI Hot Plug and Redundancy Recovery Software. The timings produced by this register are based on a 166 MHz FSB clock rate. The values must be scaled appropriately for other frequencies. As opposed to the other Hot Plug Registers, there is only one of these registers.

Device: 17 Function: 2 Offset: B0h			
Bit	Attr	Default	Description
31:17	RV	0h	Reserved.
16	RWC	0	TIMER: Timer Interrupt This bit is set whenever IMI_HPINT is asserted due to this register. This bit should be inspected by software whenever it handles an MP Hot-Plug Interrupt. Software must clear this bit by writing a 1 to it.
15	RW	0	ENABLE: Interrupt Enable When set, IMI_HPINT is asserted when the count elapses.
14:5	RW	000h	COUNT: The maximum number of 12.5 ms intervals that will elapse between setting the ENABLE bit and IMI_HPINT is asserted. The delay will be between $12.582912 * (\text{COUNT} - 1)$ and $12.582912 * \text{COUNT}$. The maximum delay is 12.8 seconds.
4:0	RW	0h	PRESCALE: This field defines the number of FSB clocks before the COUNT field is decremented. COUNT is decremented every 2^{PRESCALE} clocks. Values other than 21 (15h) should be used for validation purposes only.

4.19.9 REDUN: Memory Redundancy Control (D17, F2)

This register enables RAID and Mirroring as described in Section 6.3.5.1, “Memory Mirroring” and Section 6.3.5.2, “Memory RAID”.

Device: 17 Function: 2 Offset: B4h			
Bit	Attr	Default	Description
31:19	RV	0	Reserved
18	RW	0	HCP: HardCode Poison When set by software, overwrite uncorrectable memory data with hardcoded value for poison. Reduces debug visibility. Must be set: 1h
17	RW	0	SD: Scrub Disable When set, disables raid/mirror scrub operation when memory returns uncorrectable data.
16	RW	0	RF: Retry Forever When set, the NB will not time out due to retries to the IMI links.
15:2	RV	0	Reserved
1	RW	0	RAID: RAID Mode Enabled When this bit is set, parity is calculated across sets of 3 cache lines that appear on different IMIs and stored in the fourth IMI. If any IMI fails, the NB will reconstruct the data from the other 3 IMIs.
0	RW	0	MIRROR: Mirroring Enabled The IMIR registers change how memory reads and writes are performed in mirroring mode, but this bit is set by software to enable secondary aspects of mirroring (such as error detection). Software must set this bit before the NB can handle any memory accesses to mirrored memory. When this bit is set, every IMI is paired with another and data is written to both. If one IMI fails, the NB will deliver data from the other. This bit must be set whenever firmware allows memory accesses and IMIR[j].IMG0WAY[] != IMIR[j].IMG1WAY[] for all i=0-3 and j=0-5. If this bit is not set, proper error sequencing is not guaranteed.

§

5 System Address Map

This chapter describes the system address maps in memory space, I/O space, and PCI configuration register map.

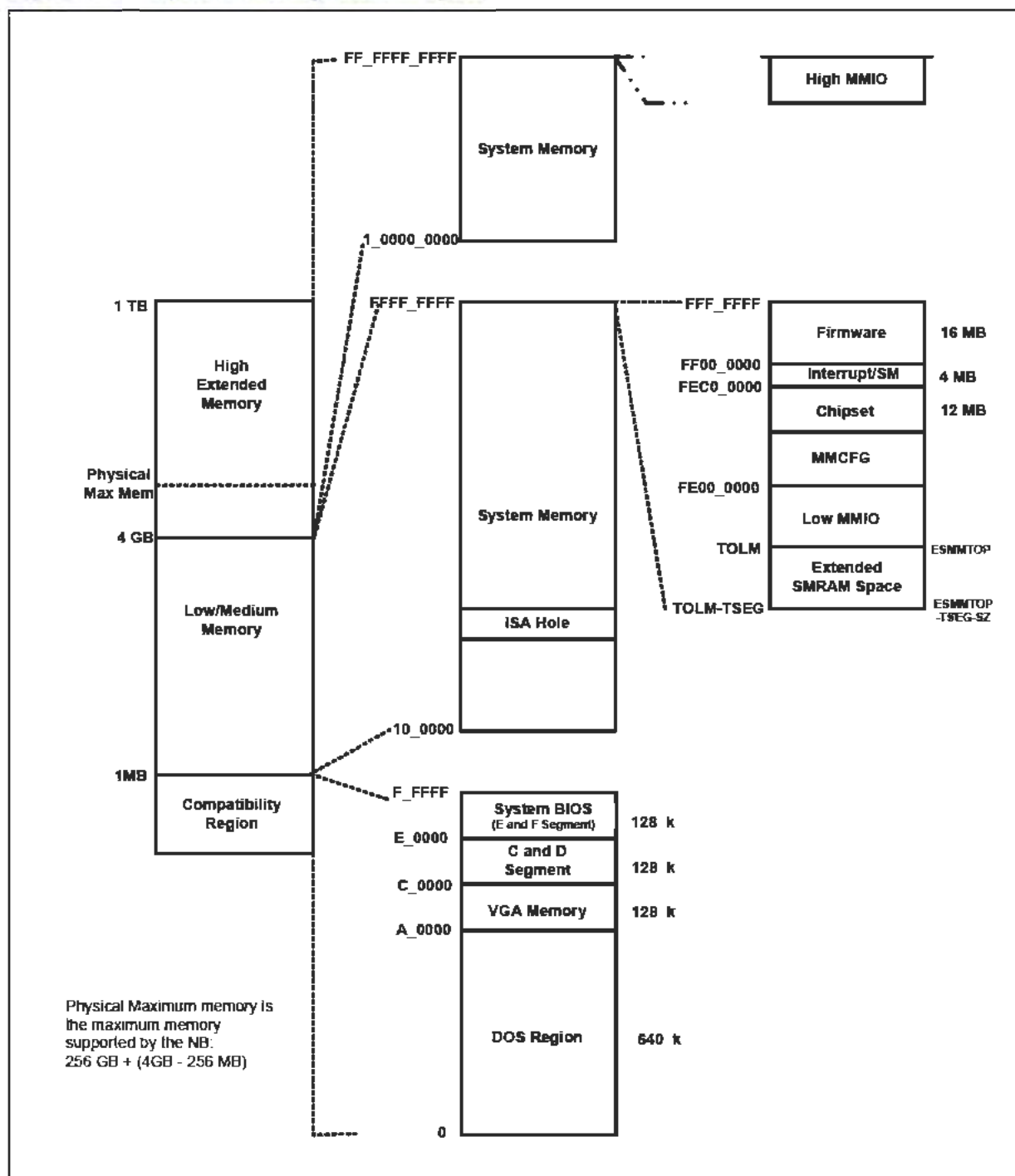
5.1 Memory Map

The Intel® E8501 chipset platform supports 40 bits of memory address space. All key components and interfaces also support the 40-bit address space.

- The 64-bit Intel® Xeon® processor MP provides address bits for a 40-bit address space.
- 40-bit local address supported over the Independent Memory Interface (IMI) for the memory space.
- 32- and 64-bit address bit formats supported for PCI Express interface.

The chipset treats accesses to various address ranges in different ways. There are fixed ranges like the compatibility region below 1 MB, interrupt delivery range, and the system region located in the 32 MB directly below 4 GB. In addition, there is a variable region for Memory Mapped I/O. The locations of these ranges in the memory map are illustrated in [Figure 5-1](#).

Figure 5-1. System Memory Address Space



5.1.1 Compatibility Region

This is the range from 0-1 MB (0_0000 to F_FFFF). Requests to the Compatibility region are directed to main memory, the Compatibility Bus, or the VGA device. Any physical DRAM that would be addressed by requests in this region that are mapped to the Compatibility Bus is not recovered. This region is divided into four ranges. Regions below 1M that are mapped to memory are accessible by the processors and by any I/O bus.

Note: The DRAM that has a physical address between 0-1 MB must not be recovered or relocated or reflected. This range must always be available to the OS as DRAM, even if at times addresses in this range are sent to the Compatibility Bus, VGA or other non-DRAM areas.

5.1.1.1 Microsoft MS-DOS* Region

Microsoft MS-DOS applications execute in the lowest 640 KB, in the address range 0h to 9_FFFFh. This range is always mapped to main memory.

5.1.1.2 VGA Memory Range

The 128 KB Video Graphics Adapter Memory range (A_0000h to B_FFFFh) can be mapped to the VGA device which may be on any PCI Express or Hub Interface or it can be mapped to main memory (if it is mapped to SMM space). At power-on this space is mapped to the HI port.

Note that the NB chipset does not support the historical MDA (Monochrome Display Adapter) space which would have appeared in the VGA region.

This region can be redirected by BIOS to point to any bus which has a VGA card. If the VGAEN bit is set in one of the NB.BCTRL configuration registers associated with the PCI Express port, then transactions in this space are sent to that PCI Express port. Note that the VGAEN bit can only be set in one and only one of the NB.BCTRL registers. For more information on the BCTRL registers, refer to [Section 4.11.27, “BCTRL\[7:1\]: Bridge Control Register \(D1-7, F0\)” on page 4-100](#).

If any VGAEN bits are set, all the ISAEN bits must be set.

This 128 KB region may also be used for mapping SMM space. The SMM range can overlay the VGA range in the A and B segments. If the SMM range overlaps an enabled VGA range then the state of the SMMEM# signal determines where accesses to the SMM Range are directed. SMMEM# asserted directs the accesses to the memory and SMMEM# deasserted directs the access to the PCI Express bus where VGA has been mapped.

This region cannot be cached.

5.1.1.3 C and D Segments

Writes and reads may be directed to different destinations in the range C_0000 to D_FFFF. Typically, these blocks were used to shadow ISA device BIOS code. For the NB these regions are used to provide address space to PCI devices requiring memory space below 1 MB. The range is divided into 8 sub-ranges. These ranges are defined by NB.PAM registers (similar to MAR registers in the 870 chipset). There is one PAM register for each sub-range that defines the routing of reads and writes (Section 4.15.8, "PAM[6:0]: Programmable Attribute MAP (D16, F1)" on page 4-177).

Table 5-1. PAM Settings

PAM value[5:4]	writes go to	reads go to	Result
00	HI1.5	HI1.5	Mapped to the compatibility PCI
01	HI1.5	Main Memory	Memory Write Protect
10	Main Memory	HI1.5	In-line Shadowed
11	Main Memory	Main Memory	Mapped to main memory

The power-on default for these segments is mapped read/write to the HI1.5 (ICH5) which forwards the access to the Compatibility Bus. Software should not set cacheable memory attributes for any of these ranges unless both reads and writes are mapped to main memory. Chipset functionality is not guaranteed if this region is cached in any mode other than both reads and writes being mapped to main memory.

For locks to this region, the NB will complete, but does not guarantee the atomicity of locked-access to this range when writes and reads are mapped to separate destinations.

If inbound accesses are expected, the C and D segments MUST be programmed to send accesses to DRAM.

5.1.1.4 System BIOS (E and F Segments)

The 128 KB region from E0000h to F_FFFFh is divided into 5 sub-ranges. Read/write attributes defined in the NB.PAM registers may be used to direct accesses to the HI1.5 (which will access the FWH) or main memory. At power-on, this area is mapped read/write to the HI1.5. The PAM settings for accesses in this region are defined in Table 5-1, "PAM Settings".

For locks to this region, the NB will complete, but does not guarantee the atomicity of locked access to this range when writes and reads are mapped to separate destinations.

If inbound accesses are expected, the E and F segments MUST be programmed to send accesses to DRAM.

5.1.2 Low/Medium Memory

The low/medium memory regions range from 1 MB to 4 GB. It consists of sub-regions for Firmware, Processor memory mapped functions, and NB chipset specific registers.

5.1.2.1 System Memory

See [Section 5.1.4, “Main Memory Region”](#).

5.1.2.2 ISA Hole

An ISA “window” can be optionally opened up between 15 MB and 16 MB by setting the NB.FDHC.HEN bit. If this bit is set, accesses to this region will be sent to the ICH5 via the HI1.5. Otherwise, it is decoded as a system memory access. The memory behind this hole cannot be reclaimed.

5.1.2.3 Extended SMRAM Space (TSEG)

SMM space allows system management software to partition a region in main memory that is only accessible by system management software. When the SMM range is enabled, memory in this range is not exposed to the Operating System. The SMM range is accessed only when the processor is in SMM-mode. This is an extended SMM range that is different than the SMM space that may reside that overlaps VGA space. The register fields that define the NB.EXSMRC.TSEG_SZ and NB.EXSMRTOP.ESMMTOP. The TSEG SMM space starts at ESMMTOP-TSEG_SZ and ends at ESMMTOP. This region may be 128KB, 256KB, or 1MB in size, depending on the TSEG_SZ field. ESMMTOP is relocatable to accommodate software that wishes to configure the TSEG SMM space before MMIO space is known. The ESMMTOP will default to the same value as TOLM.

If SMM is enabled, the chipset allows accesses to this range only when the SMMEM# signal on the FSB is asserted with the request.

If SMMEM# is deasserted, accesses to the SMM Range are master aborted. If SMMEM# is asserted the access is routed to main memory. The NB will use the SMM enable and range registers to determine where to route the access.

The NB will not support a locked-access that crosses an SMM boundary. Firmware should not create data structures that span this boundary.

SMM main memory is protected from Inbound accesses.

Note: In order to make cacheable SMM possible, the chipset must accept EWB's (BWL's) and must absorb IWB (HITM) data regardless of the condition of the SMMEM# pin. Because of this, care must be used when attempting to cache SMM space. The chipset/platform cannot protect against processor which attempts to illegally access SMM space that is modified in another processor's cache. Any software that creates such a condition (for example, by corrupting the page table) will jeopardize the protective properties of SMM.

See [Table 5-9 “Decoding Processor Requests to SMM and VGA Spaces”](#) on page 5-257 for details on SMM decoding.

5.1.2.4 Memory Mapped Configuration Region

There is one relocatable memory mapped configuration regions on the NB. The FSB address defines the configuration register to be accessed and the FSB data either returns or provides register contents. As opposed to CF8/CFC-based configuration accesses, this mechanism is atomic. The memory mapped configuration region is compatible with the PCI Express enhanced configuration mechanism. It is a 256MB window that maps to PCI Express registers. (Both in the chipset and south of the chipset.) The location of this MMCFG window is defined by the NB.EXP_ECBASE register (Section 4.15.13).

The default value of NB.EXP_ECBASE maps these regions into a region where there will be no wasted memory that is lost behind it. The default value for the PCI Express registers is 0100_0000. If this range is moved, the following recommendations will enable reclaiming the memory that is lost to MMCFG accesses.

1. MMCFG range is mapped to a legal location within the range between TOLM and 4GB. Since ranges must not overlap other legal ranges, it is safest to put this range between TOLM and the lowest real MMIO range (The current default is in these ranges).

OR

2. Put the regions above 4GB (above the top of memory and not overlapping above 4GB MMIO space).

This space is also accessible inbound. To protect the chipset under normal operation, inbound accesses can be protected by the NB.EXP_CTRL.DIS_INB_CFG bit (Section 4.11.28).

BIOS/software must ensure there are no outstanding configuration accesses or memory accesses to the old and new MMCFG range addresses when relocating this range.

5.1.2.5 Low Memory Mapped I/O (MMIO)

This is the first of the two NB memory mapped I/O ranges. The low memory mapped I/O range is defined to be between TOLM and FE00_0000h. This low MMIO region is further subdivided between the PCI Express and HI1.5 (ICH5) ports. The following table shows the registers used to define the MMIO ranges for each PCI Express/HI device. These registers are compatible with PCI Express and the PCI-to-PCI bridge specifications. Note that all subranges must be contained in the low memory mapped I/O range (between TOLM and FE00_0000h). In other words, the lowest base address must be above TOLM and the highest LIMIT register must be below FE00_0000h. Subranges must also not overlap each other.

Table 5-2. Low Memory Mapped I/O^a (Sheet 1 of 2)

I/O Port	NB Base	NB Limit
HI (ICH5)	N/A ^b	
PCI Express D Memory	MBASE1 (MBASE in device 1)	MLIMIT1 (MLIMIT in device 1)
PCI Express D Prefetchable Memory	PMBASE1 (MBASE in device 1)	PMLIMIT1 (MLIMIT in device 1)
PCI Express C0 Memory	MBASE2	MLIMIT2
PCI Express C0 Prefetchable Memory	PMBASE2	PMLIMIT2
PCI Express C1 Memory	MBASE3	MLIMIT3

Table 5-2. Low Memory Mapped I/O^a (Sheet 2 of 2)

I/O Port	NB Base	NB Limit
PCI Express C1 Prefetchable Memory	PMBASE3	PMLIMIT3
PCI Express B0 Memory	MBASE4	MLIMIT4
PCI Express B0 Prefetchable Memory	PMBASE4	PMLIMIT4
PCI Express B1 Memory	MBASE5	MLIMIT5
PCI Express B1 Prefetchable Memory	PMBASE5	PMLIMIT5
PCI Express A0 Memory	MBASE6	MLIMIT6
PCI Express A0 Prefetchable Memory	PMBASE6	PMLIMIT6
PCI Express A1 Memory	MBASE7	MLIMIT7
PCI Express A1 Prefetchable Memory	PMBASE7	PMLIMIT7

NOTES:

- This table assumes NB.PMLU and NB.PMBU are 0's. Otherwise, the prefetchable memory space will be located in high MMIO space.
- NB does not need base/limit for ICH5 because subtractive decoding will send the accesses to the ICH5. This is OK for software also, since the ICH5 is considered part of the same bus as the NB.

The NB will decode addresses in this range and route them to the appropriate H11.5 or PCI Express port. If the address is in the low MMIO range, but is not contained in any of the PCI Express base and limit ranges, it will be routed to the H11.5.

If the NB.PMLU and NB.PMBU registers are greater than 0, then the corresponding prefetchable region will be located in the high MMIO range instead.

5.1.2.6 Chipset Specific Range

The address range FE00_0000h - FEBF_FFFFh region is reserved for chipset specific functions.

- FE60_C000h - FE6F_FFFFh: This range is used for memory mapped NB registers. They are accessible only from the FSB. These registers are fixed since they are needed early during the boot process. The registers include:
 - 4 Scratch Pad Registers
 - 4 Sticky Scratch Pad Registers
 - 4 Boot flag registers
 - 4 Repeatability registers

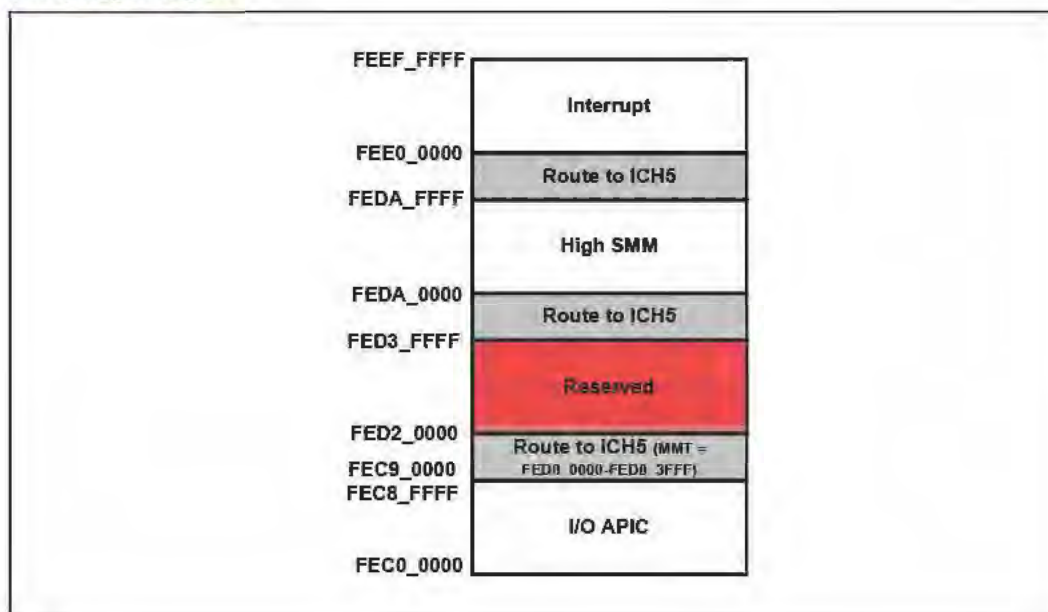
These registers are described in the NB Configuration Register Chapter.

Only the first 4 bytes of each 1 K block in this 16 K range is accepted by the NB. The NB will issue requests to the remainder of this region to the ICH unless they map into one of the relocatable regions such as MMCFG. The mechanism for this range can be the same as it is for the memory mapped configuration accesses.

5.1.2.7 Interrupt/LT Region

This 4 MB range is used for processor specific applications. It lies between FEC0_0000h and FEEF_FFFFh and is split into four 1 MB segments.

Figure 5-2. Interrupt Region



5.1.2.7.1 I/O APIC Controller Range

This address range FEC0_0000 to FEC8_FFFF is used to communicate with the IOAPIC controllers in the Intel 6700PXH/6702PXH 64-bit PCI Hub, or ICH5.

The APIC ranges are hard coded. Reads and writes to each IOAPIC region should be sent to the appropriate HII.5 or PCI Express as indicated below.

Table 5-3. IOAPIC Address Mapping

IOAPIC0 (HII.5)	0_FEC0_0000 to 0_FEC7_FFFF ^a
IOAPIC1 (PCI Express D)	0_FEC8_0000 to 0_FEC8_0FFF
IOAPIC2 (PCI Express C0)	0_FEC8_1000 to 0_FEC8_1FFF
IOAPIC3 (PCI Express C1)	0_FEC8_2000 to 0_FEC8_2FFF
IOAPIC4 (PCI Express B0)	0_FEC8_3000 to 0_FEC8_3FFF
IOAPIC5 (PCI Express B1)	0_FEC8_4000 to 0_FEC8_4FFF
IOAPIC6 (PCI Express A0)	0_FEC8_5000 to 0_FEC8_5FFF
IOAPIC7 (PCI Express A1)	0_FEC8_6000 to 0_FEC8_6FFF
Reserved	0_FEC8_7000 to 0_FEC8_FFFF

NOTES:

a. IOAPIC range is hard coded on the ICH5

For hot-plug IOAPIC support, it is recommended that software use the standard MMIO range to communicate to the Intel 6700PXH 64-bit PCI Hub. To accomplish this, the PXH.MBAR must be programmed within the PCI Express device MMIO region.

Inbound accesses to this memory range should also be routed to the I/O APIC controllers. This could happen if software configures MSI devices to send MSI's to an I/O APIC controller.

5.1.2.7.2 High SMM Range

If high SMM space is enabled by EXSMRC.H_SMRAME, then requests to the address range from FEDA_0000 to FEDB_FFFF will be aliased down to the physical address of A_0000 to B_FFFF. The HIGHSMM space allows cacheable accesses to the compatible (legacy) SMM space. In this range, the chipset will accept EWB's (BWL's) regardless of the SMMEM# pin. Also, if there is an implicit writeback (HITM with data), the chipset will update memory with the new data (regardless of the SMMEM# pin). Note that if the HIGHSMM space is enabled, the aliased SMM space of A_0000-B_FFFF will be disabled.

Note: In order to make cacheable SMM possible, the chipset must accept EWB's (BWL's) and must absorb IWB (HITM) data regardless of the condition of the SMMEM# pin. Because of this, care must be used when attempting to cache SMM space. The chipset/platform cannot protect against processors who attempt to illegally access SMM space that is modified in another processor's cache. Any software that creates such a condition (for example, by corrupting the page table) will jeopardize the protective properties of SMM.

5.1.2.7.3 Interrupt Range

Requests to the address range FEE0_0000 to FEEF_FFFF are used to deliver interrupts. Memory reads or write transactions to this range are illegal from the processor. The processor issues interrupt transactions to this range. Inbound interrupts from the PCI Express devices in the form of memory writes to this range are converted by the NB to FSB interrupt requests.

5.1.2.7.4 Reserved Ranges

The NB will master abort requests to the remaining addresses in the interrupt range. This is the FEF0_0000h - FEF7_FFFFh or FED4_0000h - FED7_FFFFh ranges. This can be done by sending the request to the compatibility bus (Hub Interface) to be master aborted. In the event the ICH5 adds new registers or functionality in this range in the future, this provides the most likely chance that it will work.

5.1.2.8 Firmware Range

The Intel® E8501 chipset platform allocates 16MB of firmware space from FF00_0000h to FFFF_FFFFh. Requests in this range are directed to the Compatibility Bus. The ICH5 will route these to its FWH interface. This range is accessible from any FSB.

5.1.3 High Extended Memory

This is the range above 4 GB. The range from 4 GB to NB.IMIR5.LIMIT is mapped to system memory. There can also be a memory mapped I/O region is located at the top of the address space. (Just below 1TB). This is described in [Section 5.1.3.2](#).

5.1.3.1 System Memory

See [Section 5.1.4, "Main Memory Region"](#).

5.1.3.2 High MMIO

The high memory mapped I/O region is located above the top of memory as defined by NB.IMIR5.LIMIT. These NB.PMBU and NB.PMLU registers in each PCI Express configuration device determine whether there is memory mapped I/O space above the top of memory. If an access is above IMIR5.LIMIT and it falls within the NB.PMBU+PMBASE/ NB.PMLU+PMLIMIT range, it should be routed to the appropriate PCI Express port. For accesses above IMIR5.LIMIT (and above 4GB) that are not in a high MMIO region, they should be master aborted.

5.1.3.3 Extended Memory (not applicable)

The range of memory just below 4 GB from TOLM to 4 GB (Low MMIO, Chipset, Interrupt) do not map to memory. If nothing is done, the DRAM memory behind the TOLM to 4 GB range will be unused.

On NB Chipset, the extended memory does not have any meaning because reclamation mechanism is handled by hardware and NB will use IMIR5.LIMIT to indicate the top of usable memory.

5.1.4 Main Memory Region

5.1.4.1 Application of Coherency Protocol

The NB applies the coherency protocol to any accesses to main memory. [Table 5-8](#) defines the conditions under which processor transactions are routed to main memory. [Table 5-11](#) defines the conditions under which inbound transactions are routed to main memory. Application of the coherency protocol includes snooping the other FSB.

Two exceptions are the C_0000 - F_FFFFh and the legacy SMM ranges. (C_0000h-F_FFFFh may not necessarily route both reads and writes to memory, legacy SMM range may target non-memory when not in SMM mode.) These exceptions will not apply the coherency protocol. The NB may malfunction if processors issue coherent transactions such as BRIL's, IWB's, and EWB's to ranges that do not call for application of the coherency protocol to both reads and writes. In addition, the chipset can not guarantee coherency for these ranges regardless of the types of transactions that are issued. Software must not set cacheable attributes for these areas (Only UC and WC attributes are not cacheable).

5.1.4.2 Routing Memory Requests

When a request appears on the FSB, PCI Express link, or Hub Interface, and it does not fall in any of the previously mentioned regions, it is compared against the IMIR.LIMIT registers on the NB and XMB.

The NB.IMIR.LIMIT registers will decode an access into a specific interleaving range. Within the interleaving range, the NB.IMIR.LIMIT register indicates which IMI the address is associated with. In the event that a mirroring event is occurring, memory writes may be associated with 2 IMI's.

The XMB will further decode the address to determine which rank the access is targeted for. The XMB contains the same XMB.IMIR.LIMIT registers as the NB, which helps it identify if the access belongs to that XMB. It also helps with converting the memory (compressing gaps due to

MMIO or IMI interleaving policies). The XMB.DMIR.LIMIT registers also decode the access into a specific interleaving range, which helps decode which rank the access will be directed to. The XMB will also convert the address into the appropriate DDR address, RAS, and CAS signals.

5.1.5 Main Memory interleaving

Memory interleaving can be described at two levels. The lowest level of memory interleaving is at the XMB level where DIMMs/Ranks are interleaved. The highest level of memory interleaving is between IMI ports. Interleaving can be at a fine or coarse granularity. Fine granularity interleaves at the cacheline boundary and provides optimal performance by evenly distributing memory accesses when addresses are accessed somewhat linearly. It helps eliminate hot spots. Coarse granularity interleaves on a boundary that is much coarser than a cacheline, for example, a different address range for each IMI (which some may call 1:1 interleaving). Coarse interleaving enables certain RAS features. A combination of fine and coarse interleaving will probably achieve the optimal mix of performance and RAS.

The NB supports 6 interleave ranges to interleave across the IMI's. Each range supports 4 way (4:1), 2 way (2:1), or 1 way (1:1) interleaving. The NB.IMIR registers contain the address of the top of the interleave range, the primary and secondary IMI ports (for mirroring) for each way.

The XMB tracks the same 6 interleave ranges. In addition, the XMB supports 5 interleave ranges to interleave across the 8 DIMM ranks. The XMB also comprehends gaps in the address to do MMIO space and IMI interleaving. The MMIO gap is accounted for using the TOLM register and is described in detail in [Section 5.1.5.4, "Recovering Main Memory Behind Other Regions"](#). The other gaps due to IMI interleaving are handled by setting the interleave range associated with that address range to not point to any ranks on the XMB.

The interleaving scheme also comprehends mirroring across IMI's, RAID, and DIMM pair sparing. For mirroring, each way of each NB.IMIR register contains a primary and secondary port. Writes are sent to both ports. Reads are sent to one port depending. For RAID, interleaving must use 4 interleaving ranges to enable RAID striping. For sparing (on the XMB), the XMB.DIMIRs reflect the live DIMM ranks. If a sparing event occurs, the DMIR will be automatically updated with the spare DIMM rank.

5.1.5.1 Interleaving Registers

The NB and XMB contain the following registers to support the ranges to do IMI and DIMM rank interleaving.

Table 5-4. Interleaving Registers

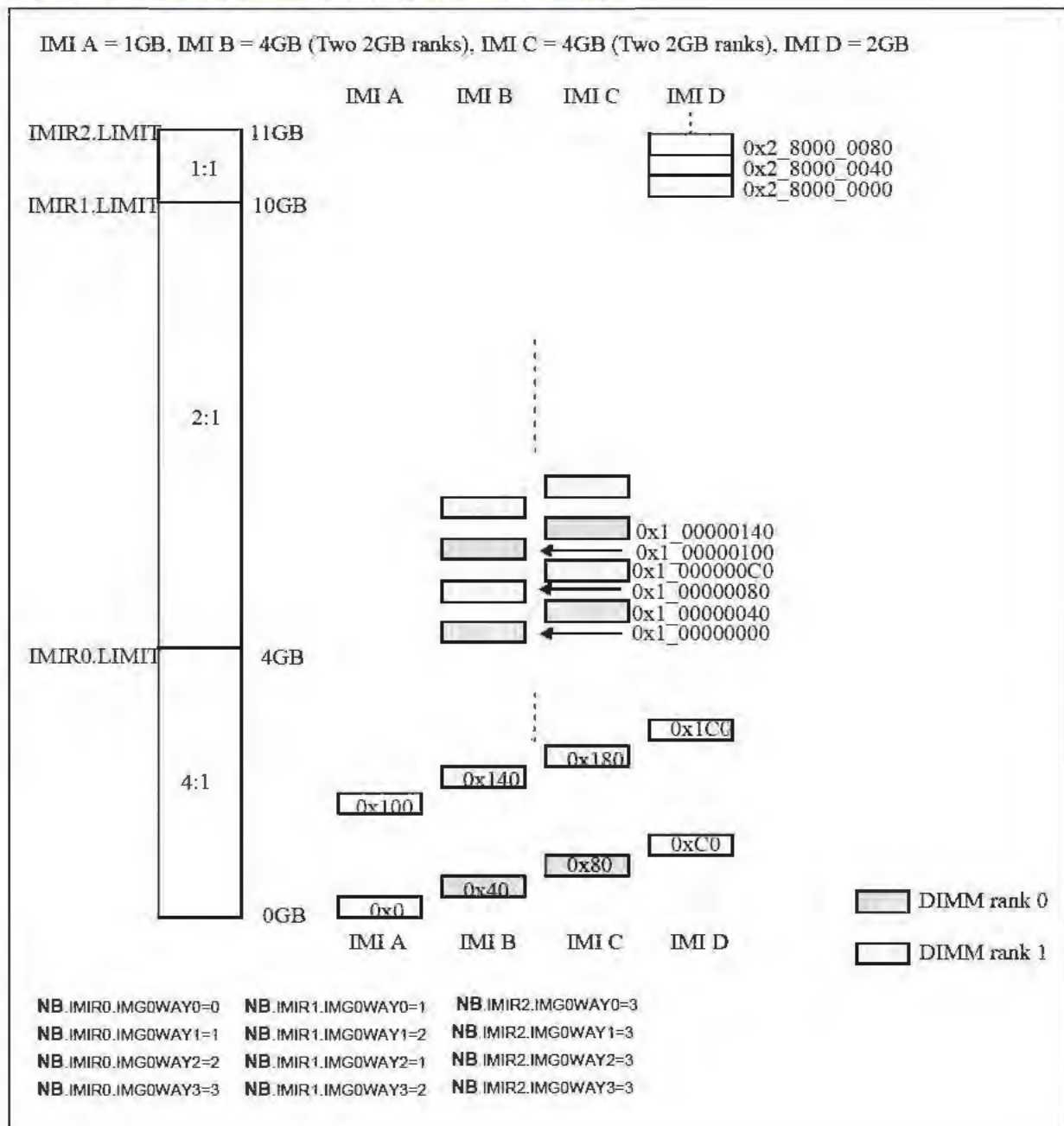
Register Name	Field	Description
NB.IMIR[5:0]	LIMIT RFB (Read From Both Images) IMG1WAY3 IMG1WAY2 IMG1WAY1 IMG1WAY0 IMG0WAY3 IMG0WAY2 IMG0WAY1 IMG0WAY0	IMIR0 Defines the interleave policy for the range from 0 to IMIR0.LIMIT. IMIR[5:1] Defines the interleave policy for the range from IMIR[N-1].LIMIT to IMIR[N].LIMIT
NB.AIMIR[5:0]	ADJLIMIT	Scratch pad registers reserved for BIOS to use to store interleave information.
XMB.IMIR[5:0]	LIMIT WAY[3:0]	IMIR0 Defines the interleave policy for the range from 0 to IMIR0.LIMIT. IMIR[5:1] Defines the interleave policy for the range from IMIR[N-1].LIMIT to IMIR[N].LIMIT.
XMB.DMIR[6:0]	LIMIT ENABLE RANK[3:0]	DIMIR0 Defines the interleave policy for the range from 0 to DIMIR0.LIMIT. DMIR[5:1] Defines the interleave policy for the range from DMIR[N-1].LIMIT to DMIR[N].LIMIT. WAY defines whether a memory request participates in this IMIR range. RANK defines which rank is set to which WAY.

5.1.5.2 Fine grain Interleave Example

This example shows how interleaving can be a fine granularity (per cacheline). Cacheline interleaving provides optimal performance by evenly distributing memory accesses. This example is not meant to be an extensive definition of interleaving, but is more to illustrate the mechanism. For example, this example does not show memory reclamation of space behind MMIO. For details on memory reclamation, please see Section 5.1.5.4, “Recovering Main Memory Behind Other Regions”.

For this example, BIOS has chosen to first interleave across all 4 IMI's until the smallest IMI has no more memory. The second range is 2:1 interleaved across IMI1 and IMI2 until these IMI's have no more memory. The third range is 1-way interleaved because only IMI3 has any memory left. Each rectangle constitutes a cacheline. In the case of IMI1 and IMI2 where there are multiple ranks (smallest granularity that the XMB will interleave), different shadings indicate which rank is being accessed.

Figure 5-3. Simple Example of Fine Grained Interleave

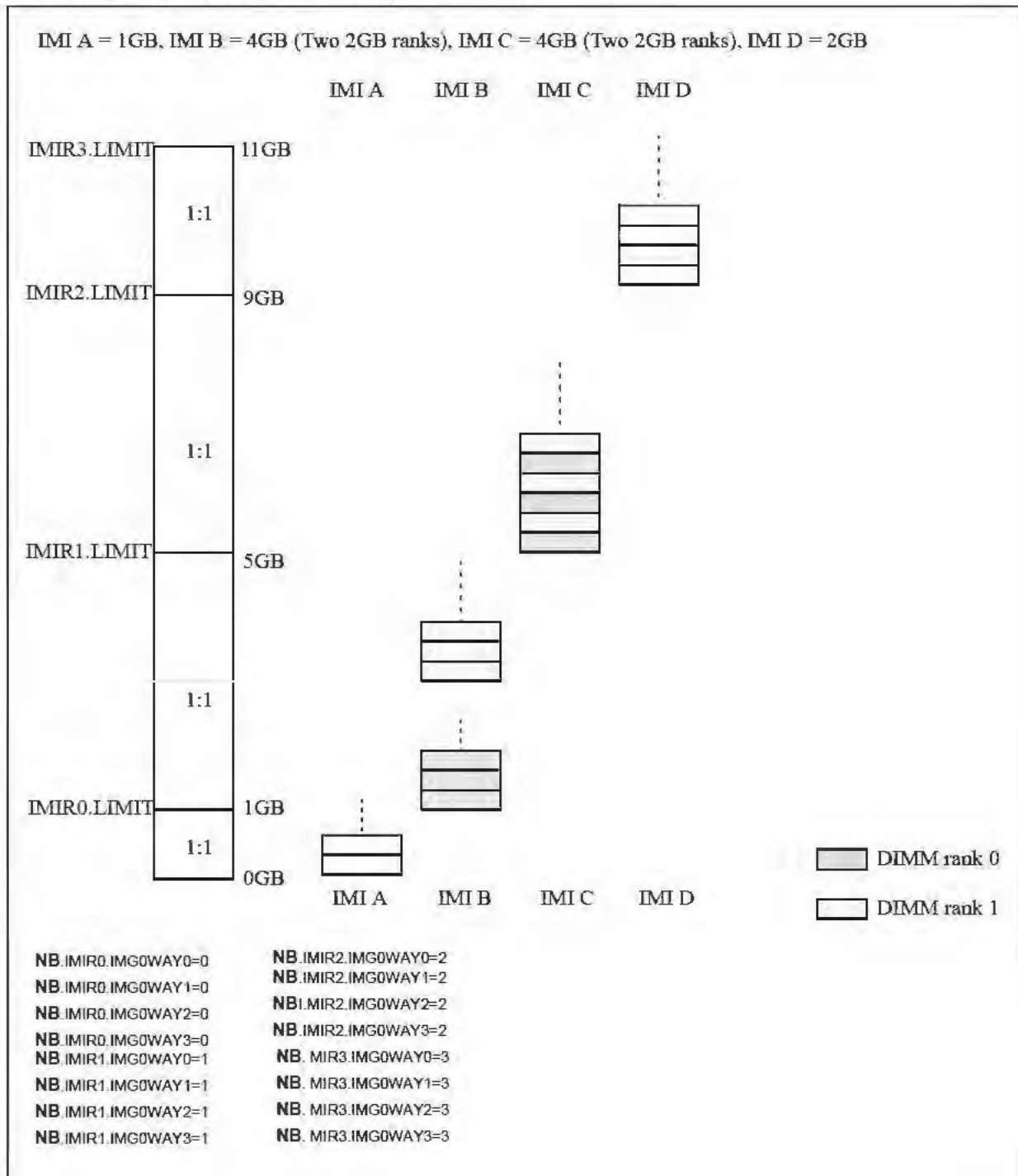


5.1.5.3 Coarse Interleave Example

This example shows how a coarse grain granularity works. It uses the same mechanism as fine grained granularity, except all regions are 1-way interleaved. As in the previous example, each rectangle constitutes a cacheline. In the case of IMI B and IMI C where there are multiple ranks (smallest granularity that the XMB will interleave), different shadings indicate which rank is being accessed.

The map on the left shows the NB interleaving ranges. Each range sends all accesses to a given IMI A. For example, the first range which is bounded by address 0 and NB.IMIR0.LIMIT will send all accesses to IMI B. The next level of interleaving is at the XMB. XMB can also support limited coarse grained interleaving. Since the XMB is limited to 5 DIMM rank interleave ranges, coarse interleaving must be limited to 5 ranges for all DIMM ranks. In this example, XMB B (IMI B) coarsely interleaves between the 2 DIMM ranks. The light dotted line in the address map indicates additional XMB boundaries to differentiated between rank 0 and rank 1. XMB C (IMI C) is still cacheline interleaved.

Figure 5-4. Example of Coarse Grained Interleave



5.1.5.4 Recovering Main Memory Behind Other Regions

The NB will reclaim memory that is lost behind the region between TOLM and 4 GB. To do this, the NB and XMB will automatically adjust the internal LIMIT values (referred to as the adjusted limit ADJLIMIT). The ADJLIMIT value that is used in the range comparators will be:

```
IF (IMIR.LIMIT > TOLM, THEN ADJLIMIT = IMIR.LIMIT + MMIO_size
ELSE ADJLIMIT = IMIR.LIMIT
```

Note: TOLM will always be below 4 GB.

$MMIO_size = 4\text{ GB} - TOLM[15:12]$ (corresponds to $A[31:28]$)

In addition, the XMB will logically perform an address conversion (subtraction) for addresses above the MMIO or other gap. For example, addresses above MMIO will be converted to Address minus the size of the MMIO gap (MMIO_size). Accesses above MMIO and a gap due to global IMI interleaving policies would be converted to Address minus MMIO_size minus size of the interleave range. The following is a logical description of the same XMB conversion.

```
IMIR_Size = LIMIT[N] - LIMIT[N-1]
IF System_addr > Any unmapped IMIR range
OtherGaps = Sum of Unmapped IMIR_Size below System_addr
IF (System_Addr > 4GB, THEN AdjustedAddr = SystemAddr - MMIO_size - OtherGaps
ELSE AdjustedAddr = System_Addr - OtherGaps
```

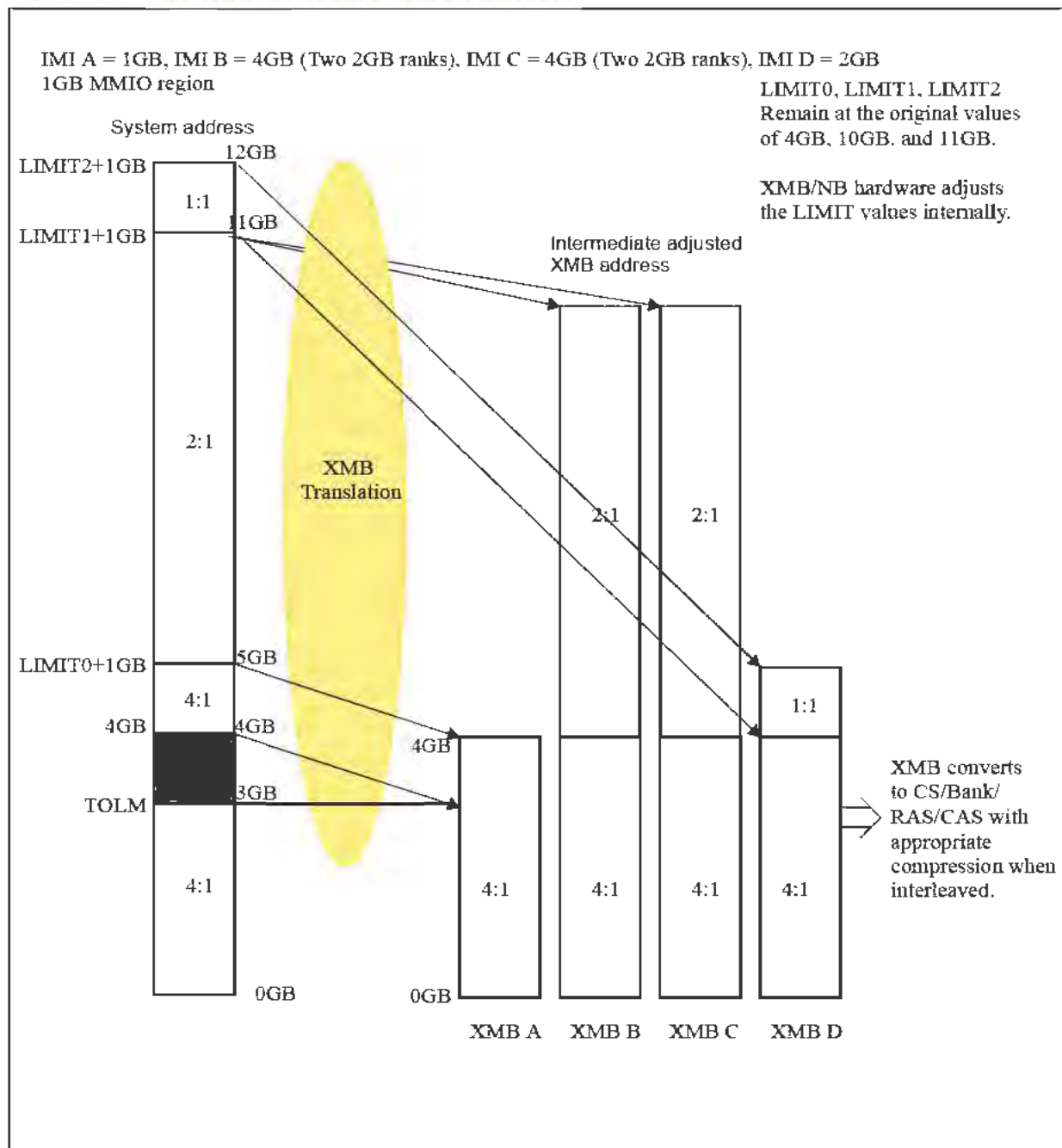
Software only needs to program the TOLM register in the NB and XMB. It does not need to adjust the LIMIT registers. In addition, there is a scratch pad register NB.AIMIR that software can use to store pertinent interleave information.

Example of reclaiming MMIO space:

The following example shows how a 1 GB MMIO (TOLM to 4 GB) range would be reclaimed. For this example, the NB/XMB internally adjusts the LIMIT values above 4 GB to $IMIR.LIMIT + MMIO_size = IMIR.LIMIT + 1\text{ GB}$ (1 GB is the MMIO_size). If there were IMIR.LIMIT's below TOLM, they would not have needed any adjustments. In addition, the XMB converts the addresses above 4 GB to $System_Address - 1\text{ GB}$ (1 GB is the MMIO_size). Addresses below 4 GB do not need to be adjusted. For addresses above 11 GB, the XMB also subtracts the size of the unmapped IMIR range, which is 6 GB in this example. (Size of the range limited by LIMIT1). Note that the example shows conceptually how the address is processed, but does not necessarily correspond to the exact implementation.

For this picture, LIMIT0 is the same as IMIR0.LIMIT, LIMIT1 is IMIR1.LIMIT, etc.

Figure 5-5. Example of Reclaiming MMIO Memory



5.1.5.5 Interleaving

A number of memory RAS features involve removing or adding memory to the system. Care must be exercised in the interleaving definition to balance RAS robustness with optimal interleaving for performance. For example, the interleaving shouldn't be set up so that removing a DIMM rank results in removing all system memory.

5.1.5.5.1 Interleaving with sparing

If DIMM sparing (without graceful degradation) is desired, at configuration time, a DIMM rank must be set aside to replace a defective DIMM rank. This spare DIMM must not be included in the IMIR and DMIR settings.

5.1.5.5.2 The Effects of Graceful Degradation & Sparing on Interleaving

Graceful degradation is when the memory associated with a bad DIMM rank is removed by software deallocating the memory. If memory is in a region that is pinned (i.e. cannot be removed), an unrelated DIMM pair can be chosen to be deallocated. Once it is deallocated, it can be converted into a spare DIMM.

This feature relies on restrictions to hardware configurations and interleaving policies to be robust. For example, If the DIMMs are interleaved so that none are a good candidate for being a spare, then a spare cannot be made. Software can follow the suggestions below to increase the chance that there will be a good candidate for a spare.

Interleaving restrictions (SW):

1. For each XMB, select a rank that can become a spare. This rank must be the largest of all populated ranks.
 - a. If there are any split DIMM's, then the spare must be a split DIMM.
 - b. Since all DIMM's will meet minimum DIMM size restriction, any DIMM can be selected to be predestined to be a spare.
2. Interleave the remaining DIMMs. The amount of memory allowable below 4 GB depends on the size of the spare rank. The memory below 4 GB must be able to fit into the spare. For example, if the spare DIMM pair holds 1 GB, then the remaining DIMM's must only have 1 GB of memory below 4 GB.

Put the spare rank above 4 GB. It must be non-interleaved (i.e. 1:1). Restrictions on hardware include:

1. If there are any double sided DIMM's, then there needs to be at least 2 double-sided DIMMs.
2. Minimum memory capacity must be larger than 4 GB.
3. Each XMB must have at least 2 DIMM pairs.
4. Restrict the use of small DIMM's. The DIMM size restrictions are dependent the number of XMB's and the IMI interleaving. The following table shows the smallest DIMM size allowable for different configurations. These restrictions are to guarantee that even with limited interleaving to allow for this functionality, there will be full memory.

Table 5-5. Minimum DIMM Size Restriction

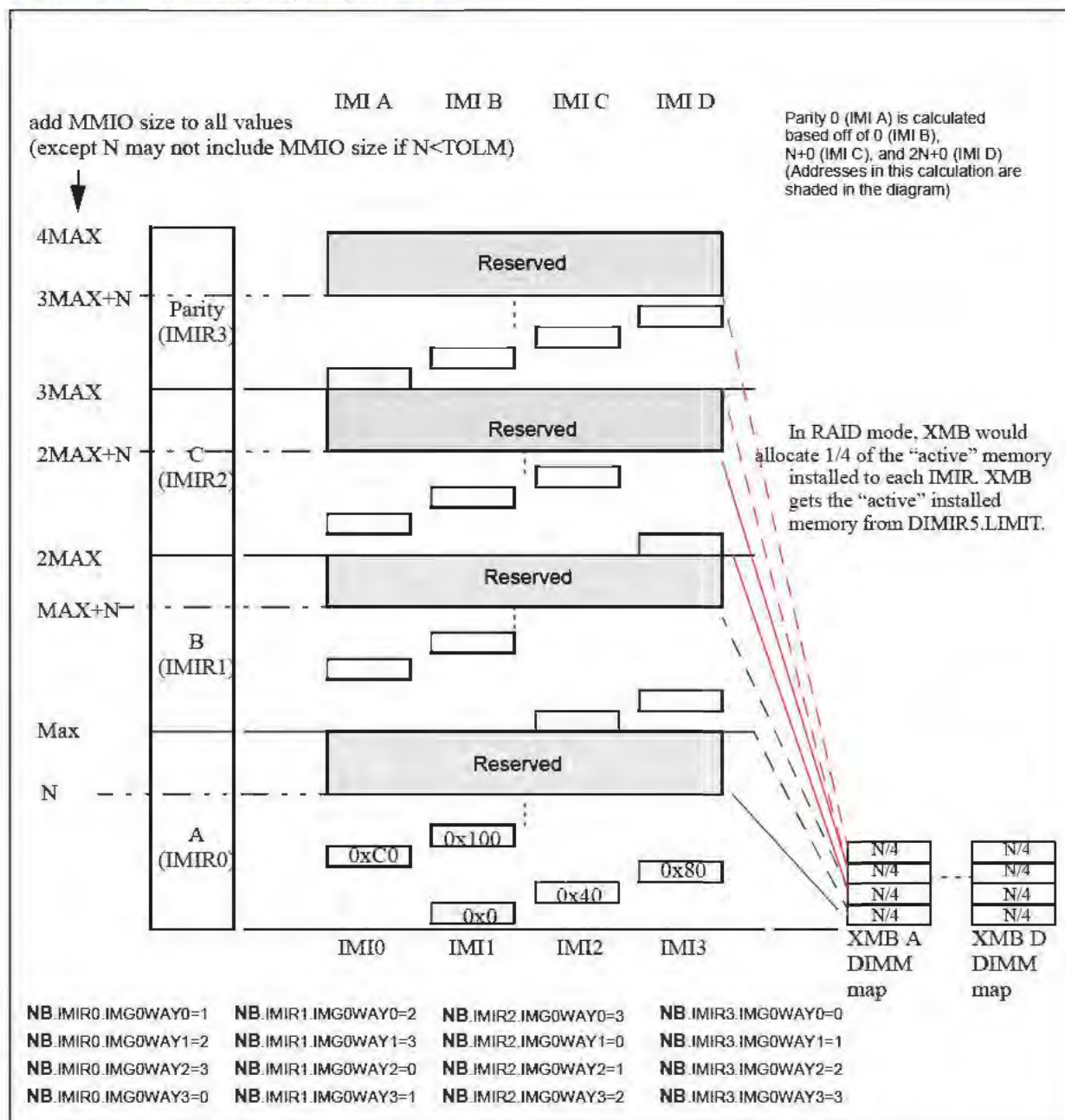
Number of XMBs	IMI Interleaving	Smallest DIMM Allowed
4	4:1	1GB
2/4	2:1	2GB
1/2/4	1:1	4GB

5.1.5.5.3 Interleaving with RAID

When Intel® E8501 chipset is in RAID mode, all 4 IMI's must be populated symmetrically. Therefore, all IMI's can be 4:1 interleaved. The reason for 4 IMIR ranges is because the starting point of each range is shifted to enable the RAID striping.

The following diagram shows how memory should be interleaved in RAID mode. This scheme includes space that is reserved in each range to accommodate capacity upgrades. A capacity upgrade with RAID would include replacing memory cards one by one with larger cards while still assuming the original smaller size. Once all 4 cards have been upgraded, a memory hot add is performed to bring the extra memory online. For systems that are not planning to be upgraded, the reserved space is not necessary. In this diagram, "N" is equal to the amount of memory per XMB that is currently installed. "MAX" represents the maximum amount of memory that is allowed per XMB after an upgrade.

Figure 5-6. Address Mapping for RAID



5.1.5.5.4 Interleaving with Mirroring

Only the NB is aware of mirroring. The NB.IMIR registers have WAY fields defined for both image 0 and image 1 (IMG{1/0}WAY[3:0]). The memory on both IMI's that are being mirrored should be the same size, so the IMI interleaving algorithm will result in the same settings for both IMI's. The only difference is that one of the IMI's is image 0 and the other is image 1. For example, if IMI A and IMI B are mirrored images and IMI C and IMI D are mirrored images, software could cacheline interleave between IMI A/B and IMI C/D by setting the IMIR register as follows:

```
IMIR.IMG0WAY0 = IMI A
IMIR.IMG1WAY0 = IMI B
IMIR.IMG0WAY1 = IMI C
IMIR.IMG1WAY1 = IMI D
IMIR.IMG0WAY2 = IMI A
IMIR.IMG1WAY2 = IMI B
IMIR.IMG0WAY3 = IMI C
IMIR.IMG1WAY3 = IMI D
```

5.1.6 General Interleaving Guidelines

1. There can be no enabled XMB.DMIRs above a disabled DMIR. For example, if DMIR[3] is disabled, DMIR[5:4] must be disabled.
2. All memory mapped I/O above 4GB (PMBASE/PMLIMIT/PMLU/PMBU) must be higher than the highest IMIR.LIMIT value plus the MMIO hole.
3. For IMIR.LIMIT values above 4GB, the value in IMIR.LIMIT cannot exceed IMIR.LIMIT minus the size of the MMIO hole, since the NB will internally adjust the LIMIT value to IMIR.LIMIT+MMIO hole.
4. Unused NB.IMIR registers should have their IMIR.LIMIT programmed to the same as the LIMIT of the highest functional IMIR. Similarly, unused XMB.DMIR and XMB.IMIR registers should have their LIMIT set to the same as the highest function IMIR/DMIR.
5. For optimal performance when doing 2-way interleaving only on the XMB, the XMB DMIR ways (called RANK[3:0]) should be programmed so that consecutive ways do not go to the same rank.

For example, if you have two ranks "m" and "n", the four ways should be programmed as follows:

Way 0	Way 1	Way 2	Way 3	OR	Way 0	Way 1	Way 2	Way 3
n	m	n	m		m	n	m	n

Similarly, if a NB is doing 2:1 interleaving (2 IMI's in the IMIR range), the ways (NB.IMIR.IMG*WAY*) should be programmed so consecutive ways do not go to the same rank.

5.2 Memory Address Disposition

5.2.1 Registers Used for Address Routing

5.2.1.1 Intel® E8501 Chipset North Bridge (NB) Registers

Table 5-6. Intel® E8501 Chipset North Bridge (NB) Memory Mapping Registers

Name	Function
IMIR[5:0]	Memory Interleaving Registers (IMI interleaving)
AIMIR[5:0]	Scratch pad register for software to use related to memory interleaving. For example, software can write MMIO gap adjusted limits here to aid in subsequent memory RAS operations.
PAM[6:0]	Defines attributes for ranges in the C and D segments. Supports shadowing by routing reads and writes to memory of I/O.
FDHC	Contains HEN which enables the ISA hold.
SMRAM	SMM Control.
EXSMRC, EXSMRTOP, EXSMRAMC	Extended SMM Control.
BCTRL	Contains VGAEN and ISAEN for each PCI Express.
TOLM	Top of low memory. Everything between TOLM and 4GB will not be sent to memory.
EXP_ECBASE	Base of the memory mapped configuration region that maps to all PCI Express registers.
MBASE (Dev 1-7)	Base address for memory mapped I/O to PCI Express ports 1-7.
IMILIMIT (Dev 1-7)	Limit address for memory mapped I/O to PCI Express ports 1-7.
PMBASE (Dev 1-7)	Base address for memory mapped I/O to prefetchable memory of PCI Express ports 1-7 ^a .
PMLIMIT (Dev 1-7)	Limit address for memory mapped I/O to prefetchable memory of PCI Express ports 1-7.
PMBU (Dev 1-7)	Prefetchable Memory Base (Upper 32 bits) - Upper address bits to the base address of prefetchable memory space. If the prefetchable memory is below 4 GB, this register will be set to all 0's.
PMLU (Dev 1-7)	Prefetchable Memory Limit (Upper 32 bits) - Upper address bits to the limit address of prefetchable memory space. If the prefetchable memory is below 4 GB, this register will be set to all 0's.
PCICMD (Dev 1-7)	MAE (Memory Access Enable) bit enables the memory and pre-fetchable ranges.

NOTES:

- a. The chipset treats memory and prefetchable memory the same. These are just considered 2 apertures to the PCI Express port.

5.2.1.2 XMB Registers

Table 5-7. XMB Memory Mapping Registers

Name	Function
IMIR[5:0]	Memory Interleaving Registers (IM) interleaving).
TOLM	Top of low memory. Used in aiding in the XMB address conversion.
DMIR[4:0]	Memory Interleaving Registers (DIMM interleaving).
MTR	Memory Technology Register.

5.2.2 Address Disposition for Processor

The following tables define the address disposition for the NB. Table 5-8 defines the disposition of outbound requests entering the NB on the FSB. For address dispositions of PCI Express/HI devices, please refer to the specifications for the Intel 6700PXH/6702PXH 64-bit PCI Hub and Intel ICH5. Table 5-8 defines how the NB completes outbound requests. Table 5-11 defines the disposition of inbound requests entering the NB on I/O bus. In these tables, an address listed as “A to B” can be interpreted as A <= Address <= B

Table 5-8. Address Disposition for Processor (Sheet 1 of 4)

Address Range	Conditions	NB Behavior	XMB Behavior
DOS	0 to 09FFFFh	Coherent Request to Main Memory. Route to main memory according to NB.IMIR registers. Apply Coherence Protocol.	Route the memory request to the correct rank based on the XMB.IMIR and XMB.DMIR registers.
SMM/VGA	0A0000h to 0BFFFFh, SMM not enabled, and VGAEN=0	Send to HI to be master aborted.	N/A
	0A0000h to 0BFFFFh, SMM not enabled, and VGAEN=1	VGA: Non-coherent request to PCI Express or Hub Interface based on BCTRL and various SMM registers.	N/A
	0A0000h to 0BFFFFh, SMM enabled and valid access to SMM space ^a	Non-coherent request to main memory. (Coherency does not need to be guaranteed. Coherency protocol can be followed if it simplifies implementation). Route to appropriate XMB according to NB.IMIR registers.	Route the memory request to the correct rank based on the XMB.IMIR and XMB.DMIR register.
	(0A0000h to 0BFFFFh and SMM enabled, and denied access to SMM space and VGAEN=0	Send to HI to be master aborted.	N/A
	(0A0000h to 0BFFFFh and SMM enabled, and denied access to SMM space and VGAEN=1	VGA: Non-coherent request to PCI Express or Hub Interface based on BCTRL registers.	N/A

Table 5-8. Address Disposition for Processor (Sheet 2 of 4)

Address Range	Conditions	NB Behavior	XMB Behavior
C and D BIOS segments (see Table 5-1 for a definition of PAM encodings)	0C0000h to 0DFFFFh and PAM=11	Non-coherent request to main memory. (Coherency does not need to be guaranteed. Coherency protocol can be followed if it simplifies implementation). Route to appropriate XMB according to NB.IMIR registers.	Route the memory request to the correct rank based on the XMB.IMIR and XMB.DMIR register.
	Write to 0C0000h to 0DFFFFh and PAM=10	Non-coherent request to main memory. (Coherency does not need to be guaranteed. Coherency protocol can be followed if it simplifies implementation). Route to appropriate XMB according to NB.IMIR registers.	Route the memory request to the correct rank based on the XMB.IMIR and XMB.DMIR register.
	Read to 0C0000h to 0DFFFFh and PAM=01	Non-coherent request to main memory. (Coherency does not need to be guaranteed. Coherency protocol can be followed if it simplifies implementation). Route to appropriate XMB according to NB.IMIR registers.	Route the memory request to the correct rank based on the XMB.IMIR and XMB.DMIR register.
	Read to 0C0000h to 0DFFFFh and PAM=10	Issue request to Hub interface.	
	Write to 0C0000h to 0DFFFFh and PAM=01		
	0C0000h to 0DFFFFh and PAM=00		

Table 5-8. Address Disposition for Processor (Sheet 3 of 4)

Address Range	Conditions	NB Behavior	XMB Behavior
E and F BIOS segments (see Table 5-1 for a definition of PAM encodings)	0E0000h to 0FFFFFh and PAM=11	Non-coherent request to main memory. (Coherency does not need to be guaranteed. Coherency protocol can be followed if it simplifies implementation). Route to appropriate XMB according to NB.IMIR registers.	Route the memory request to the correct rank based on the XMB.IMIR and XMB.DMIR register.
	Write to 0E0000h to 0FFFFFh and PAM=10	Non-coherent request to main memory. (Coherency does not need to be guaranteed. Coherency protocol can be followed if it simplifies implementation). Route to appropriate XMB according to NB.IMIR registers.	Route the memory request to the correct rank based on the XMB.IMIR and XMB.DMIR register.
	Read to 0E0000h to 0FFFFFh and PAM=01	Non-coherent request to main memory. (Coherency does not need to be guaranteed. Coherency protocol can be followed if it simplifies implementation). Route to appropriate XMB according to NB.IMIR registers.	Route the memory request to the correct rank based on the XMB.IMIR and XMB.DMIR register.
	Read to 0E0000h to 0FFFFFh and PAM=10	Issue request to Hub Interface	
	Write to 0E0000h to 0FFFFFh and PAM=01		
	0E0000h to 0FFFFFh and PAM=00		
Low/Medium Memory	10_0000 to EF_FFFF	Coherent Request to Main Memory. Route to main memory according to NB.IMIR registers. Apply Coherence Protocol.	Route the memory request to the correct rank based on the XMB.IMIR and XMB.DMIR register.
ISA Hole	F0_0000 to FF_FFFF and FDHC.HEN = 0	Coherent Request to Main Memory. Route to main memory according to NB.IMIR registers. Apply Coherence Protocol.	Route the memory request to the correct rank based on the XMB.IMIR and XMB.DMIR register.
	F0_0000 to FF_FFFF and FDHC.HEN = 1	Issue request to Hub Interface	
Low/Medium Memory (cont)	100_0000 <= Addr < ESMMTOP-TSEG_SZ OR ESMMTOP <= Addr < TOLM	Coherent Request to Main Memory. Route to main memory according to NB.IMIR registers. Apply Coherence Protocol.	Route the memory request to the correct rank based on the XMB.IMIR and XMB.DMIR register.
Extended SMRAM Space	ESMMTOP-TSEG_SZ <= Addr < ESMMTOP and (valid SMM access or SMM range disabled)	Coherent Request to Main Memory. Route to main memory according to NB.IMIR registers. Apply Coherence Protocol.	Route the memory request to the correct rank based on the XMB.IMIR and XMB.DMIR register.
	ESMMTOP-TSEG_SZ <= Addr < ESMMTOP and denied access to SMM space	Master Abort.	
Low MMIO	TOLM <= Addr < FE00_0000 and falls into a legal BASE/LIMIT range	Request to PCI Express based on <MBASE/IMILIMIT and PMBASE/PMLIMIT> registers.	
	TOLM <= Addr < FE00_0000 and not in a legal BASE/LIMIT range	Send to HI to be master aborted.	

Table 5-8. Address Disposition for Processor (Sheet 4 of 4)

Address Range	Conditions	NB Behavior	XMB Behavior
PCI Express MMCFG	HECBASE <= Addr < HECBASE+256MB	Convert to a configuration access and route according to the Configuration Access Disposition.	
NB specific	FE00_0000h to FEBF_FFFFh AND valid NB memory mapped register address	Issue configuration access to memory mapped register inside NB.	
	FE00_0000h to FEBF_FFFFh AND NOT a valid NB memory mapped register address	Send to HI to be master aborted.	
I/O APIC registers	FEC0_0000 to FECB_FFFFh	Non-coherent request to PCI Express or HI based on Table 5-3, "IOAPIC Address Mapping".	
ICH/ICH timers	FEC9_0000h to FED1_FFFF	Issue request to Hub Interface.	
High SMM	FEDA_0000h to FE0B_FFFF and (valid SMM access or SMM range disabled)	Coherent Request to Main Memory. Route to main memory according to NB.IMIR registers. Apply Coherence Protocol.	Route the memory request to the correct rank based on the XMB.IMIR and XMB.DMIR register.
	FEDA_0000h to FEDB_FFFF and access denied to SMM space	Master Abort.	
Interrupt	interrupt transaction to FEE0_0000h to FEEF_FFFFh (not really memory space)	Route to appropriate FSB(s). See Interrupt Chapter for details on interrupt routing.	
	memory transaction to FEE0_0000h to FEEF_FFFFh	Send to HI to be master aborted.	
Firmware	FF00_0000h to FFFF_FFFFh	Issue request to Hub Interface.	
High Memory	1_0000_0000 to MIR5.LIMIT ^b (max FF_FFFF_FFFF)	Coherent Request to Main Memory. Route to main memory according to NB.IMIR registers. Apply Coherence Protocol.	Route the memory request to the correct rank based on the XMB.IMIR and XMB.DMIR register.
High MMIO	PMBU+PMBASE <= Addr <= PMLU+PMLIMIT	Route request to appropriate PCI Express port.	
All others	All Others (subtractive decoding)	Issue request to Hub Interface.	

NOTES:

- Access is in SMM space, SMMEM# pin is asserted, and D_OPEN, D_CLS, D_LCK permit access type (code or data). See Table 5-9 "Decoding Processor Requests to SMM and VGA Spaces" on page 5-257
- This is the top of legal memory. Even if IMIR5 is not a valid MIR range, it will still be the top of memory because unused IMIR's should have the LIMIT's set to match the highest functional IMIR.

5.2.2.1 Processor Access to SMM Space

The FSB has a SMMEM# signal that qualifies the request asserted as having access to a system management memory. The SMM register defines SMM space that may fall in one of three ranges: A0000-BFFFFh, Extended SMRAM Space, or High SMRAM Space. The NB prevents illegal processor access to SMM memory. This is accomplished by routing memory requests from processors as a function of request address, code or data access, the SMMEM# signal accompanying request and the settings of the NB.SMRAM, NB.EXSMRC, and NB.BCTRL registers. Table 5-9 defines request type and Attribute field for each case. Illegal accesses are routed to the Compatibility Bus where they are Master Aborted. SMMEM# only affects NB behavior if it falls in an enabled SMM space. Note that the D_CLS only applies to the Legacy (A0000-BFFFFh) SMM space. The bold values indicate the reason SMM access was denied.

D_CLS closes off the SMM space to all accesses, except for code reads with SMMEM# asserted. EWB/TWB's will also be accepted.

Table 5-9. Decoding Processor Requests to SMM and VGA Spaces

SMM region	Address	BRLC?	EWB/WB	SMMEM#	D_OPEN	D_CLS	VGAEN	Enabled? ^a	Access Granted to SMM Space?
Legacy VGA/SMM	A_0000h <= A <= B_FFFFh	X	X	X	X	X	0	N	Master abort
		X	X	X	X	X	1	N	To VGA
		N	N	0	0	X	N	Y	Denied - HI
		N	N	0	0	X	Y	Y	Denied - VGA port
		X	X	X	1	0	X	Y	Yes
		X	X	X	1	1	X	Y	Illegal programming - HI (master abort)
		X	X	1	0	0	X	Y	Yes
		N	N	1	0	1	N	Y	Denied - HI
		N	N	1	0	1	Y	Y	Denied - VGA port
		X	Y	X	X	X	X	X	N/A ^b
		Y	N	1	0	1	X	Y	Yes
Extended SMRAM Space (TSEG)	(TOLM-TSEG <= A <= TOLM)	X	X	X	X	X	X	N	Access to DRAM
		X	N	0	0	X	X	Y	Denied
		X	N	0	1	X	X	Y	Yes
		X	N	1	0	X	X	Y	Yes
		X	Y	X	X	X	X	X	Yes
HIGHSMM	FEDA_0000h <= A <= FEDB_7FFFh	X	X	X	X	X	0	N	Master abort
		X	N	0	0	X	X	Y	Denied
		X	N	0	1	X	X	Y	Yes
		X	N	1	0	X	X	Y	Yes
		X	Y	X	X	X	X	X	Yes

NOTES:

- See Table 5-10 "Enabled SMM Ranges" for the definition of an enabled space
- Software must not cache this region.

Table 5-10 defines when an SMM range is enabled. All the enable bits G_SMFRAME, H_SMRAM_EN, and TSEG_EN are located in the NB.EXSMRC register.

Table 5-10. Enabled SMM Ranges

Global Enable G_SMFRAME	High Enable H_SMRAM_EN	TSEG Enable TSEG_EN	Legacy SMM Enabled?	HIGHSMM Enabled	Extended SMRAM Space (TSEG) Enabled?
0	X	X	No	No	No
1	0	0	Yes	No	No
1	0	1	Yes	No	Yes
1	1	0	No	Yes	No
1	1	1	No	Yes	Yes

5.2.3 Inbound Transactions

In general, inbound I/O transactions are decoded and dispositioned similarly to processor transactions. The key differences are in SMM space, memory mapped configuration space and interrupt policies.

For all table entries where an access is forwarded to HI to be master aborted, if an access comes from HI, the NB may master abort a transaction without forwarding it back to the HI.

Table 5-11. Address Disposition for Inbound Transactions (Sheet 1 of 3)

Address Range	Conditions	NB Behavior	XMB Behavior
DOS	0 to 09FFFFh	Coherent Request to Main Memory. Route to main memory according to NB.IMIR registers. Apply Coherence Protocol.	Route the coherent request to the correct rank based on the XMB.IMIR and XMB.DMIR registers.
SMM/VGA	0A0000h to 0BFFFFh, SMM not enabled, and VGAEN=0	Send to HI to be master aborted.	N/A
	(0A0000h to 0BFFFFh and SMM enabled and VGASE=0	Send to HI to be master aborted.	N/A
	0A0000h to 0BFFFFh and VGAEN=1	VGA: Non-coherent request to PCI Express or Hub Interface based on BCTRL and various SMM registers.	N/A
C, D, E, and F BIOS segments (see Table 5-1 for a definition of PAM encodings)	0C0000h to 0FFFFFFh and PAM = 11 ^a	Non-coherent request to main memory. (Coherency does not need to be guaranteed. Coherency protocol can be followed if it simplifies implementation.) Route to appropriate XMB according to NB.IMIR registers.	Route the coherent request to the correct rank based on the XMB.IMIR and XMB.DMIR registers.
Low/Medium Memory	10_0000 to EF_FFFF	Coherent Request to Main Memory. Route to main memory according to NB.IMIR registers. Apply Coherence Protocol.	Route the coherent request to the correct rank based on the XMB.IMIR and XMB.DMIR registers.

Table 5-11. Address Disposition for Inbound Transactions (Sheet 2 of 3)

Address Range	Conditions	NB Behavior	XMB Behavior
ISA Hole	F0_0000 to FF_FFFF and FDHC.HEN = 0	Coherent Request to Main Memory. Route to main memory according to NB.IMIR registers. Apply Coherence Protocol.	Route the coherent request to the correct rank based on the XMB.IMIR and XMB.DMIR registers.
	F0_0000 to FF_FFFF and FDHC.HEN = 1	Issue request to Hub Interface.	
Low/Medium Memory (cont)	100_0000 <= Addr < TOLM-TSEG_SZ	Coherent Request to Main Memory. Route to main memory according to NB.IMIR registers. Apply Coherence Protocol.	Route the coherent request to the correct rank based on the XMB.IMIR and XMB.DMIR registers.
Extended SMRAM Space	ESMMTOP-TSEG_SZ <= Addr < ESMMTOP	TSEG SMM enabled: Master Abort. TSEG SMM disabled: Issue request to main memory.	
Low MMIO	TOLM <= Addr < FE00_0000 and falls into a legal BASE/LIMIT range	Request to PCI Express based on <IMIBASE/IMILIMIT and PMBASE/PMLIMIT> registers.	
	TOLM <= Addr < FE00_0000 and not in a legal BASE/LIMIT range	Send to HI to be master aborted.	
PCI Express MMCFG	HECBASE <= Addr < HECBASE+256MB	Inbound MMCFG is allowed when dis_inb_cfg configuration bit is cleared, and, the NB does not modify addresses of peer-to-peer transactions. Inbound MMCFG is not allowed when dis_inb_cfg configuration bit is set. This access should be routed as a general memory mapped I/O access, which should ultimately master abort, since it should not hit any legal MMIO range.	
NB specific	FE00_0000h to FEBF_FFFFh AND valid NB memory mapped register address	Inbound MMCFG is not allowed. This access should be routed as a general memory mapped I/O access, which should ultimately master abort, since it should not hit any legal MMIO range.	
	FE00_0000h to FEBF_FFFFh AND NOT a valid NB memory mapped register address	Send to HI to be master aborted.	
I/O APIC registers	FEC0_0000 to FEC8_FFFFh	Non-coherent request to PCI Express or HI based on Table 5-3 "IOAPIC Address Mapping"	
ICH/ICH timers	FEC9_0000h to FED1_FFFF	Issue request to Hub Interface.	
High SMM	FEDA_0000h to FEDB_FFFF	Master Abort.	
Interrupt	Inbound write to FEE0_0000h - FEEF_FFFFh	Route to appropriate FSB(s). See Interrupt Chapter for details on interrupt routing.	
	memory transaction to FEE0_0000h to FEEF_FFFFh	Send to HI to be master aborted.	

Table 5-11. Address Disposition for Inbound Transactions (Sheet 3 of 3)

Address Range	Conditions	NB Behavior	XMB Behavior
Firmware	FF00_0000h to FFFF_FFFFh	Master abort.	
High Memory	1_0000_0000 to MIR5.LIMIT (max FF_FFFF_FFFF)	Coherent Request to Main Memory. Route to main memory according to NB.IMIR registers. Apply Coherence Protocol.	Route the coherent request to the correct rank based on the XMB.IMIR and XMB.DMIR registers.
High MMIO	PMBU+PMBASE <= Addr <= PMLU+PMLIMIT	Route request to appropriate PCI Express port.	
All others	All Others (subtractive decoding)	Issue request to Hub Interface.	

NOTES:

- a. Other combinations of PAM's are not allowed if inbound accesses to this region can occur. Just like Cayuse, chipset functionality is not guaranteed.

5.3 I/O Address Map

The I/O address map is separate from the memory map and is primarily used to support legacy code/drivers that use I/O mapped accesses rather than memory mapped I/O accesses. Except for the special addresses listed in Section 5.3.1, "Special I/O Addresses", I/O accesses are decoded by range and sent to the appropriate HI/PCI Express port, which will route the I/O access to the appropriate device.

5.3.1 Special I/O Addresses

There are two classes of I/O addresses that are specifically decoded by the NB:

- I/O addresses used for VGA controllers.
- I/O addresses used for the PCI Configuration Space Enable (CSE) protocol. The I/O addresses 0CF8h and 0CFCh are specifically decoded as part of the CSE protocol.

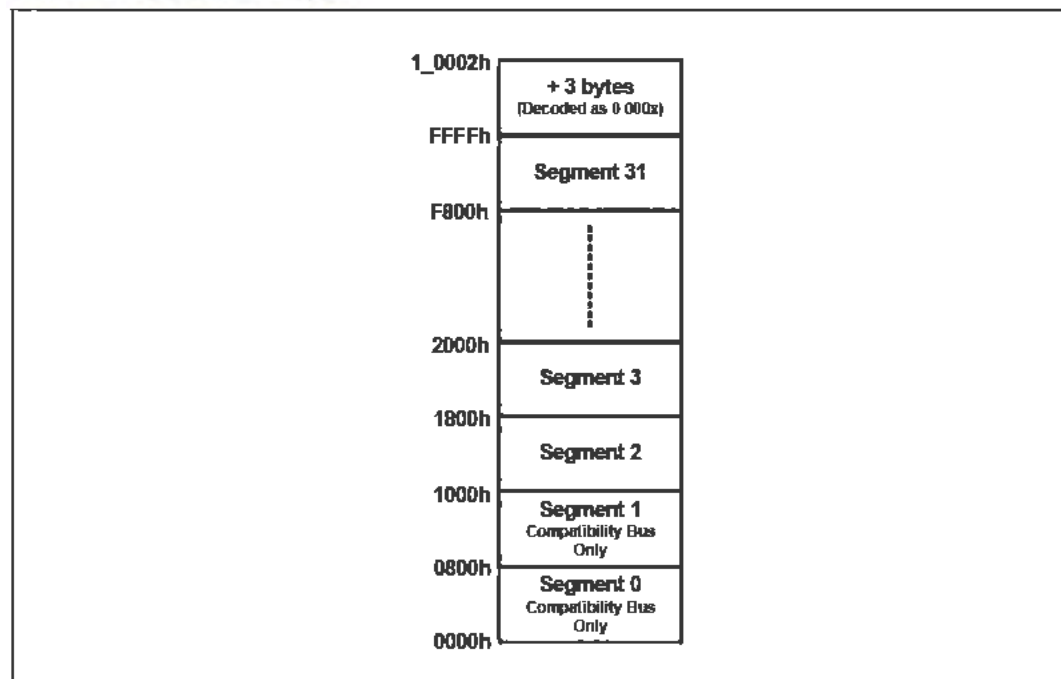
Historically, the 64 KB I/O space actually was 64 K+3 bytes. For the extra 3 bytes, A#[16] is asserted. The NB decodes only A#[15:3] when the request encoding indicates an I/O cycle. Therefore accesses with A#[16] asserted are decoded as if they were accesses to address 0 and be forwarded to the Compatibility Bus.

At power-on, all I/O accesses are mapped to the Compatibility Bus.

5.3.2 Outbound I/O Access

The NB chipset allows I/O addresses to be mapped to resources supported on the I/O buses underneath the NB. This I/O space is partitioned into 32 x 2KB segments. Each of the I/O buses can have from 0 to 32 segments mapped to it. Each PCI bus gets contiguous blocks. All PCI busses must be assigned contiguous blocks. The lowest block, from 0 to 0FFFh, is always sent to the Compatibility Bus. For further details on the legacy I/O ranges that the ICH decodes in the lowest block, please refer to *Intel® 82801EB I/O Controller Hub 5 (ICH5) and Intel® 82801ER I/O Controller Hub 5 (ICH5R) Datasheet*. The ICH5 Datasheet is available at: <http://developer.intel.com/design/chipsets/datashts>.

Figure 5-7. System I/O Address Space



5.3.2.1 Outbound I/O Transaction Routing

The NB applies these routing rules in order:

(A[2:0] for the following is not physically present on the FSB, but are calculated from BE[7:0]).

1. I/O addresses used for VGA controllers on PCI Express:

If one of the BCTRL.VGAEN bit is set, and each addressed byte is in the following range:

A[9:0] = 3B0h-3BBh, 3C0h-3DFh. (A[15:10] are ignored for this decode.)

The access is sent to the PCI Express port that VGAEN is set for. Note that only one VGAEN bit can be set.

For example, a 2 byte read starting at X3BBh includes X3BB-X3BCh (X can be any number since A[15:10] are ignored).

Since A[9:0] = 3BCh which is not one of the VGA bytes, the access is not routed to VGA.

2. Configuration accesses: If a request is a DW accesses to 0CF8h (See CFGDAT register) or 1-4 B accesses to 0CFCh with configuration space is enabled (See bit 31 of CFGADR register), the request is considered a configuration access. Configuration accesses are routed based on the bns and device numbers.

3. ISA Aliases: If the BCTRL.ISAEN is set, addresses X100-X3FFh, X500-X7FFh, X900-XBFF, and XD00-XFFFh will result in the access being sent out to the Hub Interface (ICH5). This is the top 768B in each 1KB block. 100 = 256B 3FF = 1K-1

4. I/O defined by IOBASE/IOLIMIT: If PCICMD.IOAE is set in any of the NB PCI Express devices and the I/O address falls in this range: (IOBASE <= address <= IOLIMIT), then the access will be sent to the PCI Express port that is defined by the correct IOBASE/IOLIMIT registers.

5. Otherwise: the I/O Read/Write is sent to HI (ICH5).

5.3.3 Inbound I/O Accesses

Inbound I/O's are supported only for peer-to-peer accesses and are decoded the same as processor initiated I/O's. Inbound I/O accesses to the NB (i.e. CF8/CFC) are not supported and will receive a Master Abort response.

§

6 Functional Description

This chapter will cover the Intel® E8501 chipset north bridge (NB) functionality and its interface components, including Front Side Bus, IMI, PCI Express and HII.5. The *Intel® E8501 Chipset North Bridge (NB) Datasheet* shall only cover the NB component. Details on the rest of the platform components can be found in each component's respective documentation.

6.1 Processor Support

The NB is the priority agent for the two FSB's and is optimized for two 64-bit Intel® Xeon® Processor MP on each bus. The Intel® E8501 chipset does not allow mixing of 64-bit Intel® Xeon® processor MP with up to 8MB L3 cache, 64-bit Intel® Xeon® processor MP with 1MB L2 cache or Dual-Core Intel® Xeon® processor 7000 sequence within the same platform.

Table 6-1. FSB Feature Summary

Feature	Description
Cache Line Size	64B
Clock Rate	Control@166 MT/s, Address@333 MT/s, Data@667 MT/s. Control@200 MT/s, Address@400 MT/s, Data@800 MT/s
Addresses	NB supports 40 address bits.
Enhanced Defer	NB responds using Defer phase when DPS is asserted.
Modified Enhanced Defer	NB typically drives IDS# 5 clocks before OOD#.
Maximum Deferred Transactions	32
Maximum In-Order Transactions	12
ECC/Parity	NB supports both DEP[7:0]# and DP[3:0]# but not at the same time.
BRIL.OWN	NB ignores the OWN bit.
BWLD	64-bit Intel® Xeon® processor MP processors do not issue this transaction.
BWL.INVLD	Disabled and treated as a BWL.
Threads/cores	NB fully supports 4 logical agents per FSB.
Modified response following clean snoop (buried HITM)	NB supports this for debug, but not for production. The solution will not be performance optimized. Buried HITM is inconsistent with enhanced defer
Processor Tristate	A[39:36] at reset (see Section 4.14.12, "POC_FSB(A/B): Power-on Configuration (D16_F0)" on page 168).

6.1.1 Clock Phasing

The FSB clocks run in phase.

6.1.2 Arbitration Phase

The following table shows the arbitration phase signals and the NB usage.

Table 6-2. Arbitration Signals

Signal	Usage
BPRI#	NB asserts this signal when it will drive requests that require a snoop or when it wants to stall the FSB, or to avoid oversubscription of certain internal resources. Processors do not drive this signal.
BNR#	NB may assert this signal for debug.
BREQ[3:0]#	The NB never drives these signals, except for BREQ[0]# at reset. These pins are connected to the processor BR# pins in a rotating order. The NB uses an active BREQ# from a processor to determine if it should release BPRI# and return the bus to the processors.
LOCK#	The NB never asserts LOCK#. The NB observes LOCK# and will do all transactions with LOCK# asserted atomically. LOCK# on I/O reads or writes is an illegal transaction.

6.1.3 Symmetric/Priority Agent Arbitration Policy

NB asserts BPRI# as necessary for flow control, and this overrides any arbitration policy on BPRI#.

NB asserts BPRI# for NB transactions only when it has transactions to drive to the bus, and it's not giving priority to CPU agents. NB limits the length of time it will assert BPRI by FSB[1:0]AC.TNBREQ[2:0]. When a CPU asserts a BREQ# signal on the bus, NB will give the bus to the CPUs as long as NB has no transactions to do. Otherwise, NB ensures BPRI stays deasserted for a period of time controlled by FSB[1:0]AC.CPUREQ[1:0]. If NB sees a BREQ change while BPRI is released, FSB[1:0]AC.CPUREQ_EXT controls whether NB allows an extra transaction to occur from the CPUs, to try to make back-to-back BWL's more likely.

NB's arbitration policy is such that as long as CPUs drive their BREQ signals efficiently (releasing it if they have no transactions to perform) and drive ADS as soon as possible when the protocol allows, that NB and the CPUs can share the bus with no dead cycles under any traffic pattern.

6.2 Request Phase

Table 6-3 defines how NB decodes transactions initiated by processors and Table 6-5 defines how NB drives transactions. NB self-samples the common-clock signals. NB does not sample the address or data signals but instead tracks what it has driven internally.

Reserved Read and Write transactions are not issued by 64-bit Intel® Xeon® processor MP. NB will complete these transactions as the most similar well defined transactions in order to enable subsequent transactions to complete. NB will respond to Reserved Read (ReqA[0]=0) as a NOP, returning data from the addressed memory. NB will respond to Reserved Write (ReqA[0]=1) as a BTM.

BWL with DEN is not issued by 64-bit Intel® Xeon® processor MP. The NB will not defer any BWL.

If a Deferred Reply is decoded, but the NB has not issued it, the NB will treat it as NOP and set FSB_FERR.F8 (refer to Table 6-39 “Errors Detected by the NB” on page 6-328).

If DEN is not asserted when it is shown to be 1 in Table 6-3, the NB will defer anyway. The “Unsupported FSB Transaction” bit will be set in the FERRST or SERRST register if these are decoded.

If an address/request parity error is detected, the NB will attempt to complete the decoded bus transaction on a best effort basis as if no error has been detected and at the same time log an F4 error. The NB will remain operational after such an error.

NB does require ASZ# (ReqA#[4:3] for memory transactions) to be correct for correct operation.

DSZ# (ReqB#[4:3] for most transactions) has no effect on NB operation.

In the following table “x” indicates that the NB will treat either 0 or 1 the same way: the bit is ignored. “0/1” indicates that the NB will treat 0 different from 1. Both 0 and 1 should be validated for both “x” and “0/1” entries.

Table 6-3. Processor Initiated Transactions Supported by Intel® E8501 Chipset North Bridge (NB) (Sheet 1 of 2)

Transaction	Variant	Support	Ab combinations decoded						
			BE#[7:0]	LOCK#	DEN#	DPS#	Ab[5]	Ab[6]	Ab[7] SMMEN#
BRIL	BRIL	yes	FFh	0	1	0/1	0/1 (OWN)		0/1
	BRIL.OWN	yes	FFh	0	1	0/1	0/1 (OWN)		0/1
	BIL	yes	0	0	1	0/1	x (OWN)		0/1
BRLC	BRLC	yes	contiguous	0	1	0/1	x (OWN)		0/1
BRLD	BRLD	yes	contiguous	0	1	0/1	x (OWN)		0/1
	BLR	yes	contiguous	1	1	x	x	x	0/1
BWL	INVLD#0, NZ	yes	FFh	0	0	0	x	0 (INVLD)	0/1
	INVLD#0, Z	yes	00h	0	0	0	x	0 (INVLD)	0/1
	BWL.INVLD NZ	yes	FFh	0	0	0	x	1 (INVLD)	0/1
	BWL.INVLD Z	yes	00h	0	0	0	x	1 (INVLD)	0/1
	DEN#	no	any	0	1	1	x	x	0
BWIL	BLW	yes	?	1	0	0	x	0	0/1
	BWIL	yes	any	0	0/1	1	x	0	0/1
IACK	IACK	yes	1h	0	1	0/1	0	0	x
SPEC	NOP	yes	0h	0	1	0/1	0	0	x
	Shut Down	yes	1h	0	1	0/1	0	0	x
	Flush	yes	2h	0	1	0/1	0	0	x
	Halt	yes	3h	0	1	0/1	0	0	x
	SYNC	yes	4h	0	1	0/1	0	0	x
	Stop Clock	yes	5h	0	1	0/1	0	0	x
	Stop Grant Ack	yes	6h	0	1	0/1	0	0	x
	SMIACK	yes	7h	0	1	0/1	0	0	x
	XTPR Update	yes	8h	0	0	0	0	0	x
BTM	BTM	yes	FFh	?	0	0	0	0	x

Table 6-3. Processor Initiated Transactions Supported by Intel® E8501 Chipset North Bridge (NB) (Sheet 2 of 2)

			Ab combinations decoded						
INT/EOI	INT	yes	"3h,Fh"	0	0	0	0	0/1	x
	EOI	yes	1h	0	0	0	1	1	x
IOR	IOR	yes	contiguous	0	1	1	x	x	x
IOW	IOW	yes	4B boundary	0	1	1	x	x	x
Reserved	Reserved Read	no							
	Reserved Write	no							

The following tables show how the NB decodes the 2nd phase of the address.

Table 6-4. Intel® E8501 Chipset North Bridge (NB) Decoding of 2nd Phase of Address

Ab	Signal	Definition	NB Usage
39:36	reserved	Reserved	Does not affect NB operation.
35:32	reserved	Snoop Count	Does not affect NB operation.
31	ATTR[7]#	XTPR Disable	Redirection Disable for XTPR update transaction, ignored otherwise
30:29	ATTR[6:5]#	Logical ID	NB uses this field to differentiate transactions from different logical processors (threads or cores) within a CPU package.
28	ATTR[4]#	Undefined	Does not affect NB operation
27:24	ATTR[3:0]#	Task Priority	Interrupt Redirection Task Priority for XTPR update transaction
		Caching Attributes	NB ignores.
23	DID[7]#	Priority/Symmetric Agent	0 since processors are symmetric agent
22:21	DID[6:5]#	Agent ID	Returned on defer Reply Or IDS#
20:16	DID[4:0]#	Transaction ID	Returned on defer reply or IDS#.
15:8	BE[7:0]#	byte enables	Indicates which bytes are valid in a data transfer.
7	EXF[4]#	SMMEM#	The NB will generally disallow access to SMM memory unless this bit is set. However, when SMM memory is made cacheable, writes may be evicted without this bit set.
6	EXF[3]#	SPLCK#	If LOCK# is asserted, the NB ignores this signal. It uses LOCK# to decode locked reads and writes.
		INVLD#	If the transaction is a BWL, INVLD#=1 indicates that the processor will invalidate the line
		INT[1]#	If the transaction is a Interrupt or EOI, differentiates different interrupt types
5	EXF[2]#	OWN#	NB ignores
		INT[0]#	If the transaction is a Interrupt or EOI, differentiates different interrupt types
4	EXF[1]#	DEN#	The NB needs to defer a transactions to ensure coherency, and the DEN is not set, the NB will defer anyway and will detect error F8.
3	EXF[0]#	DPS#	If set, the NB may complete the transaction with a deferred phase (IDS# assertion).

6.2.1 16-Thread Support

The NB supports up to 16 total threads automatically, in case of 2 dual-core CPUs per FSB of 2 threads each.

6.2.2 Intel® E8501 Chipset North Bridge (NB) Requests

Table 6-5 and Table 6-6 show what the NB drives on the FSB. Table 6-5 shows REQ and address driven for the first cycle. Table 6-6 shows the 2nd phase of the address. Any bits not specifically mentioned are driven to 0.

Table 6-5. Encoding of Transactions Initiated by the Intel® E8501 Chipset North Bridge (NB)

Common Bus Transaction	NB Transaction	REQa	REQb ^d	Aa[39:3]
Deferred Reply	Deferred Reply	00000	11011 ^b	A[39:24]=0
			11000 ^c	A[23:16] = DID[7:0] of original transaction A[15] = Early indication of HIT# A[13] = Debug information (DBGMATCH) A[12:8] = Debug information (PTQID[4:0]) A[3] = A[3] of original request
Zero Length Bus Read Invalidate	BIL	zz010 ^d	11000	Address from other FSB, PCI Express, or Hub Interface
Bus Read Line Code	BRLC	zz100	11000	Address from other FSB, PCI Express, or Hub Interface
Bus Read Line Data	BRLD	zz110	11000	Address from other FSB
Interrupt	INT	01001	11100	0_FEEy_yyy0

NOTES:

- RCNT[3:2] are driven on REQb[4:3] if FSB[1:0]AC.DRVCNT is set
- 64B of data will follow
- Less than 64B of data will follow.
- zz = ASZ[1:0]

Table 6-6. Encoding 2nd Phase of Address for Transactions Initiated by Intel® E8501 Chipset North Bridge (NB) (Sheet 1 of 2)

Ab	Signal	Definition	NB Drives
39:36	Reserved	Reserved	Debug information: SRCINFO[3:0] if FSB[1:0]AC.DRVCNT = 1, else 0000.
35:32	Reserved	Snoop Count	Debug information: SCNT[3:0] if FSB[1:0]AC.DRVCNT = 1, else 0000.
31	ATTR[7]#	XTPR Disable	0
30:29	ATTR[6:5]#	Logical ID	0
28:27	ATTR[4]#	Undefined	RCNT[1:0] is driven when the FSB[1:0]AC.DRVCNT bit is set.
26:24	ATTR[2:0]#	Caching Attributes	111 (writeback)
23	DID[7]#	Agent ID	1

Table 6-6. Encoding 2nd Phase of Address for Transactions Initiated by Intel® E8501 Chipset North Bridge (NB) (Sheet 2 of 2)

Ab	Signal	Definition	NB Drives
22:16	DID[6:0]#	Agent ID, Transaction ID	Debug information: CDCID[6:0]
15:8	BE[7:0]#	Byte Enables	11111111 Data present 00000000 No data
7	EXF[4]#	SMMEN#	0
6	EXF[3]#	INT1	0
5	EXF[2]#	INT0	A[5] on IPI for redirected interrupt, otherwise 0.
4	EXF[1]#	DEN#	0
3	EXF[0]#	DPS#	0

6.2.3 Snoop Phase

The snoop phase consists of the following signals: HIT#, HITM#, DEFER#.

The NB can retry a deferred reply that it initiated in order to cause the processor to retry it to fix some BIL conflict cases and buried HITM cases.

The NB requires that any agent that asserts HITM# on a transaction must drive data. It may not drive HITM# and then respond with no data when given a TRDY#. If it does not provide data, the NB cannot guarantee correct operation.

Table 6-7. Snoop Phase Signals

Signal	Usage
HIT#, HITM#	For a deferred BRLC and BRLD, if HIT# is asserted on the local FSB or the remote FSB (if snooped), the shared status will be driven during the deferred completion. Either DHIT# will be driven active during defer phase, or HIT# will be driven during snoop phase of the deferred reply. NB only snoop stalls as a debug function. If HITM# is asserted (and HIT# de-asserted) on a processor transaction, NB will response with an implicit-write back.
DEFER#	The NB will assert DEFER# followed by a DEFER or RETRY response. The NB never gives a DEFER response to requests that it initiates.

6.2.4 Response Phase

Table 6-3 defines the responses that the NB gives to each FSB transaction. Table 6-5 defines the responses the NB gives to each NB transaction. The conditions are evaluated in the order given, that is the higher condition has priority.

When a request decode is in doubt, the NB will respond in the way that is most likely to keep these processors running. For example, the NB will respond to the transaction it decodes whether an Address/Request Parity error is detected or not. When a transaction is decoded that is not supported by the NB, the NB will treat them as the closest transaction it has been designed to support.

If a coherency violation occurs, the NB will complete the transaction as best it can (e.g. deferring memory reads even if DEN=0, giving IWB response to illegal HITM's), but any data delivered after the violation may not be correct.

Table 6-8. Intel® E8501 Chipset North Bridge (NB) Responses to Processor Transactions

Transaction Type	Conditions	Response
BIL	IF modified snoop response (Coherency Protocol Violation: set Unsupported Bus transaction bit in FERRST or SERRST) (can't have illegal data size; is defined to be zero length)	IWB
	ELSEIF NB has established a system lock	Retry
	ELSEIF BIL Conflict	Defer
	ELSE	Defer
BRIL, BRLC, BRLO	IF modified snoop response	lwb
	ELSEIF NB has established a system lock	Retry
	ELSE	Defer
BLR	IF modified snoop response	lwb
	ELSEIF (NB has not yet established a system lock) OR (NB has ordered another BLR from either bus first, and that BLR is not yet complete)	Retry
	ELSE	Normal Data
BWL	All	No Data
BWIL	IF modified snoop response	lwb
	ELSEIF NB has established a system lock	Retry
	ELSE	Defer
BLW	All	No Data
IOR, IOW	IF NB has established a system lock.	Retry
	ELSE	Defer
Reserved Ignore	All	No Data
Reserved Write	IF REQa[2:0] == 011 (treat as BWL)	No Data
	ELSE (treat as BTM)	No Data
Reserved Read	all (treat as NOP)	No Data
IACK, HALT Shut Down, Stop Grant Ack.	The NB has established a system lock.	Retry
	ELSE	Defer
NOP, FLUSH, SYNCH, SMIACK, XTPR Update, BTM	All	No Data
INT	IF NB has established a system lock and interrupt is redirectable	Retry
	ELSE	No Data
EOI	IF NB has established a system lock	Retry
	ELSE	No Data
Deferred Reply	If the NB did not issue. The processor may not issue this transaction.	No Data

Table 6-4 defines the various outcomes of the defer response.

Table 6-9. Defer Response Outcomes

Conditions	Response
If buried HITM#.)	Deferred Reply with Retry response
Elseif BIL conflict	Deferred reply with retry response
Elseif DPS was asserted with transaction and FSB[1:0]AC.MEDEN=0 and not a BIL	Deferred Phase with IDS# and OOD# always asserted together (Enhanced Defer)
Elseif DPS was asserted with transaction and FSB[1:0]AC.MEDEN=1 and not a BIL	Deferred Phase with IDS# and OOD# sometimes asserted separately (Modified Enhanced Defer)
Elseif NB will transfer data to processor	Deferred Reply with Normal Data response
Elseif	Deferred Reply with No Data response

Table 6-10. Intel® E8501 Chipset North Bridge (NB) Responses to Intel® E8501 Chipset North Bridge (NB) Transactions

Transaction Type	Conditions	Response
Deferred Reply	If Buried HITM# or BIL conflict is detected	Retry
	Elseif original transaction requires that NB transfer data to the processor	Normal Data
	Else	No Data
BIL, BRLD, BRLC	HITM# asserted	IWB
	Else	No Data
INT	always	No Data

6.2.5 Defer Phase

Table 6-11. Defer Phase Signals

Signal	Intel® E8501 Chipset North Bridge (NB) Usage
IDS#	A strobe that indicates the NB is completing a deferred transaction. The chipset returns ownership to the requesting processor when IDS# is asserted. That is, the NB no longer has snoop responsibility for an ADS asserted subsequent to an IDS# assertion. When IDS# is active, ID[7:0] is valid. If the NB will not return data for the transaction, the transaction is completed. If the NB will return data, it will come with the concurrent or next OOD# assertion. Only the NB drives IDS#, so it may be asserted once every two clocks.
ID[7:0]#	Valid while IDS# is asserted. The first phase carries the transaction ID sent in the request phase of the deferred transaction. The second phase of ID carries information about the snoop results of the transaction. The NB asserts DHIT# for BRLD's if the address potentially remains shared in another agent. That is, when HIT# was asserted in the snoop phase for the original transaction or when it was snooped on the other bus. If FSB[1:0]AC.DRVDCNT is set, the NB will drive DCNT on ID[7:5]#. DCNT is defined to be (number of DDEL# assertions - number of OOD# assertions) on the cycle before IDS#. If FSB[1:0]AC.DRVDCBD is set, the NB will drive debug information onto the ID[7:0]# signals such that: IDa[7] = CDCID[6] IDb[7:5] = CDCID[5:3] IDb[2:0] = CDCID[2:0] NB never asserts DHITM# in response to a BRIL.OWN.
OOD#	This is a data phase signal. See Table 6-13.

6.2.5.1 Modified Enhanced Defer

The NB has an OOD# pin that enables IDS# assertion to give advance notice that a data transfer will occur. IDS# commits the order of deferred phase data transfers, but OOD# indicates when data will be driven.

After reset, the NB will drive OOD# coincident with IDS#. The bus protocol is defined such that this is equivalent to enhanced defer with no early data indication. The processor can sample IDS# but not OOD#. Once the processor is configured to sample both IDS# and OOD#, the FSB {A/B}_AC.MEDEN bit will be set to enable the NB to drive IDS# before OOD#.

When data is returned from the other FSB (Remote HITM#/IWB), the NB will not drive IDS# until after TRDY# has been asserted on the remote FSB. If NB asserted IDS# before TRDY# on the remote bus, IDS# might be asserted on both FSB's for a remote HITM, but TRDY# could not enable the IWB transfer on either bus due to the deadlock rule.

When data is returned from memory, the NB will drive IDS# as soon as it can after getting a read return header on an IMI (assuming coherency has already been completed) or after sampling TRDY# for an IWB from the other FSB.

6.2.6 Data Phase

6.2.6.1 Data Transfer Sizes

Table 6-12 defines limitations on the transfer sizes and alignments supported by the Intel® E8501 chipset north bridge (NB). The NB does not support all possible transfer sizes defined by Intel® Xeon® processor family. The NB supports all alignments generated by the 64-bit Intel® Xeon® processor MP profile. When the NB cannot support a length, the length is aliased to a supported length and no error is logged.

Address alignment does not affect which bytes are transferred, but only their order.

Table 6-12. Processor Transfer Sizes and Alignment limitations of the Intel® E8501 Chipset North Bridge (NB) (Sheet 1 of 2)

Transaction Type	LEN#,BE#	Alignment	Comment
BRLC	64	Any 8B	Data is returned in critical word first order.
	1-8		Not needed for 64-bit Intel® Xeon® processor MP.
	0		
BRLD	64	Any 8B	Data is returned in critical word first order.
	1-8		
	0		
BRIL	64	Any 8B	Only 64B for 64-bit Intel® Xeon® processor MP.
BIL	0	Any 8B	
BLR,BLW	1-8	Any 8B	Validation must cover cases where the sum of the BE in the BLR's or BLW's total up to 16 bytes.

Table 6-12. Processor Transfer Sizes and Alignment limitations of the Intel® E8501 Chipset North Bridge (NB) (Sheet 2 of 2)

Transaction Type	LEN#,BE#	Alignment	Comment
BWIL	64	Any 8B	Data is received in critical word first order
	1-8		Any combination of byte enables are legal. The BE do not need to be contiguous and any number from 0-8 may be asserted. For example, patterns such as 0001_1111, 0001_0101, 0101_0101 or 1010_1111 are allowed. Note: Downstream devices may not be able to support these unrestricted patterns of BE if the address maps to a MMCFG region.
	0		NB will assert TRDY for memory writes with zero data length but the processor will not assert any data transfer signals.
BWL	64B	Any 8B	NB will assert TRDY for memory writes with zero data length but the processor will not assert any data transfer signals.
	0		
I/O read	1-4 byte	Any 8B	the NB ignores the LEN field. Therefore, if LEN is other than "8 bytes or less" the transaction will be handled as usual and no error will be detected.
I/O write	1-4 byte	Any 8B	the NB ignores the LEN field. Therefore, if LEN is other than "8 bytes or less" the transaction will be handled as usual and no error will be detected. BE may NOT cross 4B boundary.
IACK	n/a	n/a	Reqb[2:0] =000, Address ignored.
SPEC	n/a	n/a	Reqb[2:0] =001, Address ignored.
BTM	n/a	n/a	Reqb[2:0] =000, Address ignored.
INT	2B or 4B	n/a	Address field holds other information.
EOI	1B	n/a	Address is meaningless.
XTPR_UPDATE	?	n/a	Address is meaningless.

6.2.6.2 Data Transfer Signals

Table 6-13 shows the signals used for data transfers.

Table 6-13. Data Transfer Signals (Sheet 1 of 2)

Signal	NB Usage
H_{A/B}_TRDY#	Signals that a symmetric agent may transfer either Write or Implicit-Writeback data.
H_{A/B}_DBSY#	The data bus is owned by some agent that cycle. Does not imply that data is necessarily being transferred that cycle.
H_{A/B}_DRDY#	data is valid. NB asserts for read data.
H_{A/B}_D[63:0]#	64 bits of data transferred at 4x the bus clock frequency.
H_{A/B}_DEP[7:0]#	ECC coverage over D[63:0]. Transferred at 4x the bus clock frequency.
H_{A/B}_DP[3:0]#	Odd parity over DBI and D[63:0]. DP must be disabled if DEP is used.

Table 6-13. Data Transfer Signals (Sheet 2 of 2)

Signal	NB Usage
H_{A/B}_DBI[3:0]#	Each DBI bit is associated with 16 data lines and 2 ECC lines. If most of any of the 18 bits in a group are active, the NB inverts the group and the corresponding DBI bit is set. Note that DBI will cover all quad pumped signals including DEP. Transferred at 4x the bus clock frequency.
H_{A/B}_OOD#	Indicates that the next or concurrent data transfer will be for deferred phase data delivery.
H_{A/B}_DSTBp#[3:0], H_{A/B}_DSTBn#[3:0]	Strobes for D[63:0]#, DEP[7:0]# and DBI[3:0]#.

6.2.7 Error Signals

The NB observes BINIT# and can be configured to drive BINIT# in certain error situations, including propagating BINIT# to the other FSB.

The “observed BINIT#” bit will be set in the FERR register and this may be configured to reset the system via ICHRST#.

The NB can be configured to assert MCERR# on the other bus when a processor asserts MCERR# (see Section 4.16.21).

6.2.8 Bus Assumptions

- BRLC's and BRLD's from the processor that have no BE asserted will get a No-Data Response. A PCI Express or Hub Interface request will be issued, but there are configuration bits to drop these zero length transactions in the I/O subsystem of NB.
- A locked request that is retried by the NB must be reissued unless a fatal processor error occurs.

6.2.9 FSB Coherency Assumptions

The following line state assumptions can be made following the snoop phase of a given transaction on a FSB populated with fully enabled 64-bit Intel® Xeon® processor MP.

Table 6-14. 64-bit Intel® Xeon® Processor MP FSB Coherency Assumptions (Sheet 1 of 2)

Transaction type	Possible Bus States after Snoop		
	Snoop Result = Modified ^a	Snoop Result = Shared ^b	Snoop Result = Clean ^c
Processor Initiated BRLC	SI	SI	MESI
NB initiated BRLC	I	SI	SI ^d
Processor Initiated BRLD	MESI	SI	MESI
NB initiated BRLD	I	SI	I
BRIL.OWN	M	Can't Happen ^e	MESI
BRIL	MESI	Can't Happen	MESI
Processor Initiated BIL	Can't Happen ^f		

Table 6-14. 64-bit Intel® Xeon® Processor MP FSB Coherency Assumptions (Sheet 2 of 2)

	Possible Bus States after Snoop		
NB initiated BIL	I	Can't Happen	I
BWL INVLD=0	Can't Happen, not snooped		MESI ⁹
BWL INVLD=1	Can't Happen, not snooped		I
BWIL	I	Can't Happen	I
BLR	SI	SI	SI
BLW	Can't happen ^h	I	I

NOTES:

- HIT# = 0, HITM# = 1. Some agent currently has the line in modified state, but will reduce to shared or invalid
- HIT# = 1, HITM# = 0. Some agent intends to keep the line in shared state
- HIT# = 0, HITM# = 0. The line is invalid in all processor caches
- The shared result is only possible when 64-bit Intel® Xeon® processor MP L2 is disabled. The NB will assume I when the CDCC.BRLCI bit is set. This should typically be set.
- A shared response indicates the agent intends to keep the line in shared state which is illegal for an invalidating transaction
- The 64-bit Intel® Xeon® processor MP does not issue BIL, even when L2 is disabled. It will back invalidate BIL's from Prescott on the IFSB
- If attribute was WC, NB could assume I, but NB does not use attributes for coherency assumptions
- Must be preceded by a BLR that would invalidate

6.2.10 Power-on Configuration

6.2.10.1 POC Register

If a bit is set in the POC register, it will drive the corresponding address line during reset. See Section 4.14.12, "POC_FSB{A/B}: Power-on Configuration (D16, F0)" on page 168 and Section 4.14.13, "POC_AUX{A/B}: CPU Tristate Control (D16, F0)" on page 170 for the register definition. This includes processor tristate on A[39:36].

6.2.10.2 BREQ0

The NB asserts BREQ[0]# during RESET# on each bus to enable processors to determine their AgentID. In a configuration with 2 processor sockets on each bus, BREQ[0]# should be routed to BREQ[0]# on the first socket (end agent) and BREQ[1]# of the second socket (Middle agent) of each bus so that they use AgentID 0 and 1 on each FSB.

6.2.10.3 Pins that Do Not Connect to NB

The NB does not connect to A20M#, PMI#, SLPCK#, IGNNE#, IERR#, SMI# or STPCLK#.

6.3 Independent Memory Interface (IMI)

6.3.1 Topology

The NB masters 1 to 4 Independent Memory Interface (IMI) Ports, each of which connects to one XMB. The IMI's are half-width (one byte of payload/transfer) outbound and full-width (2 Bytes of payload/transfer) inbound. Each IMI supports up to 1 Terabyte (2⁴⁰ Bytes) of addressability. The IMI's are operated independently, that is a given cache line request is serviced by 1 IMI. The NB does not support cascaded, or lock-stepped IMI's.

6.3.2 Physical Layer

The physical layer will be based on Scalable Differential electricals. As mentioned above, the NB issues half-width packets outbound and full-width packets inbound. The returning inbound link must be the same frequency and generated from the same clock buffer. This allows NB to perform bit de-skewing once at reset rather than periodically.

The NB drives the IMI outbound at 2.67 GHz or 3.2 GHz (16 times the front side bus clock rate). Since an outbound packet takes 8 transfers, commands can be issued at 333 MHz or 400 MHz. A single Write command transfer takes 9 packets, so writes can be issued at 167 MHz or 200 MHz.

The NB can accept responses inbound at the same rate as the outbound (2.67 GHz or 3.2 GHz). Since the inbound data packet is 4 transfers, the NB can accept one-read returns from each IMI at 667 MHz or 800 MHz. Write acknowledges are accepted at 333 MHz or 400 MHz.

6.3.3 Memory Space

MemRds and MemWrs are issued to memory space. The IMI has more address bits than the FSB's, so NB IMI addresses will be the same as FSB, PCI Express or HII.5 addresses. NB will always drive Mem_Address[41:40] to zero in IMI packets.

The NB has a set of IMI Interleave Range (IMIR) registers which define the IMI to which a memory request should be issued ([Section 4.15.15, "IMIR\[5:0\]: IMI Interleave Range, \(D16, F1\)" on page 188](#)). The IMIR registers partition the memory address space at 256 MB boundaries. Within each IMI Interleave Range, cache lines may be interleaved in different ways to support optimum interleaving of arbitrary (non-symmetrical) memory populations.

6.3.4 Configuration Space

NB issues ConfigWrs and ConfigRds to Configuration space. Only one Configuration Space access can be outstanding on the IMI at a time. Configuration accesses are limited to 1, 2 or 4 bytes in size. Any combination of contiguous Byte enables is supported.

6.3.4.1 Device Assignment

The NB assigns device numbers to the 4 IMI's and their corresponding XMBs as described in [Table 4-8 on page 60](#). These all appear on Bus 0.

The XMBDEV register defines the XMB device used for each IMI. They will default to the correct fixed values for the attached XMBs so that BIOS need not perform any enumeration. NB BIOS should not modify these registers, and does not need to read them. The only usage of these registers is for non-NB specific software (e.g. Hot-Plug driver) to determine the device number of an XMB attached to a IMI.

The NB does not support cascaded XMBs, which would require multiple devices.

The IMI requires that the NB toggle a bit in the header for each configuration access so that the NB can distinguish a new command from a retry. The IMI allows only one configuration command outstanding at a time.



Normally, the NB responds to XMB configuration requests for register offsets < 40h whether the XMB is present or not. Most of these registers are read-only so any IMI function can be accessed to provide the data. The NB will return the XMB value for the DID register. The SVID registers are writable once, so it is possible that software could write these registers. The NB maintains separate SID and SVID registers for the XMB functions. When the IMIHPC.XMBHDR bit is set, the NB forwards the accesses to the XMB to allow examination of the revision ID of the XMB.

The NB will master-abort any configuration requests to the XMB when the corresponding IMI is in a state that does not support XMB configuration accesses.

Figure 6-1. Software Visible IMI States

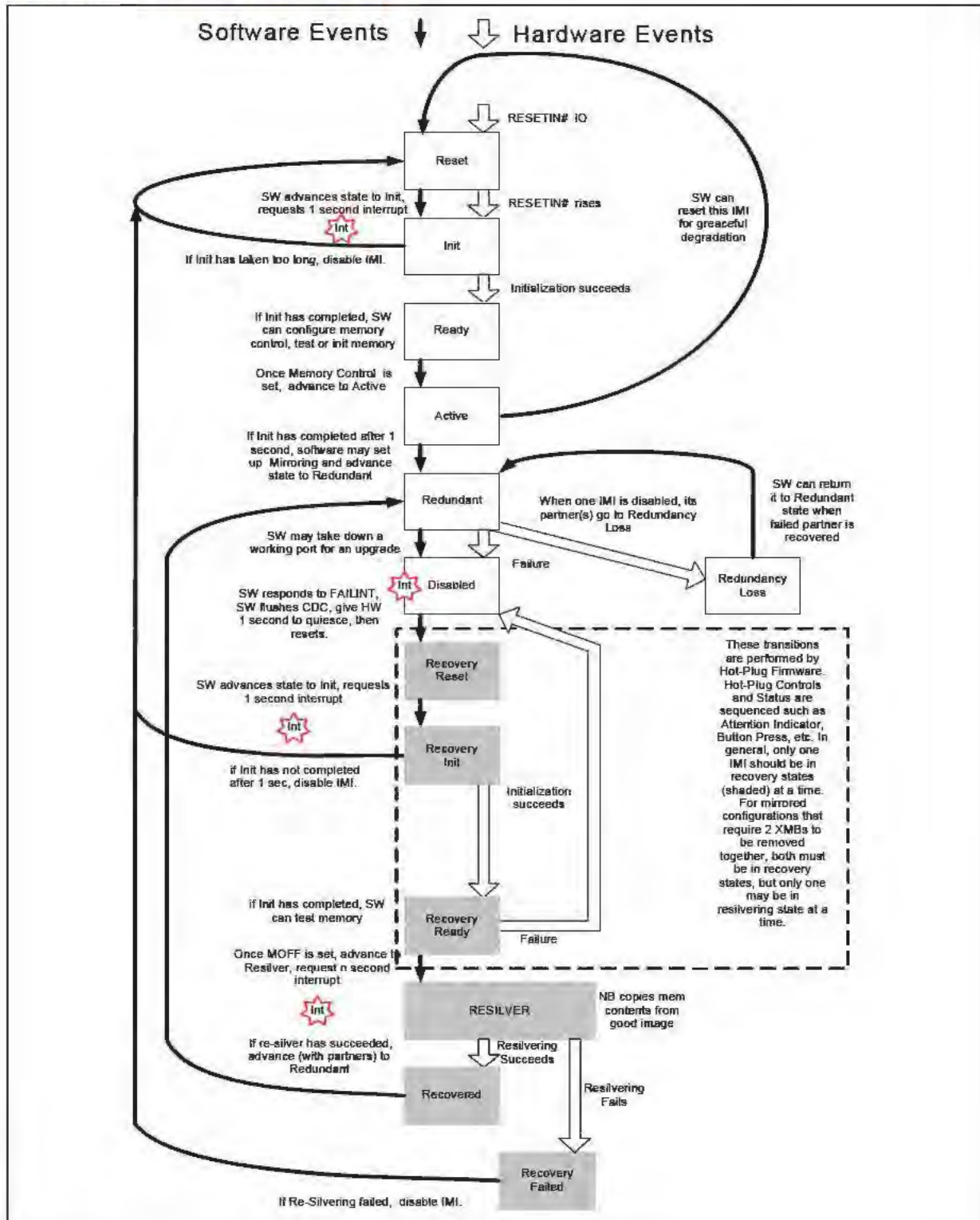


Table 6-15. Software Visible IMI State (Sheet 1 of 2)

State	NB Behavior	Supported IMI activity	SW Usage
Reset	Outputs are disabled. Link, Physical, and Transaction Layers are reset. Configuration reads and writes to this IMI are still supported, but configuration writes to the associated XMB will be Master Aborted. IMI_{A,B,C,D}_RST# is asserted.	None	Software writes this state to force an IMI into reset. Software should place any IMI that is not operational, and which is not currently being considered for recovery in this state. When in this state, IMIHPC.PWRCTL Section 4.13.10, "IMIHPC: IMI Hot-Plug Control (D{8,10,12,14}, F0)" on page 150 can be set so that cards may be added or removed.
Init	NB performs the initialization steps. Configuration reads and writes to this IMI are still supported, but configuration writes to the associated XMB will be Master Aborted.	HW controlled init ops. No system accesses.	Software writes this state to start IMI initialization. Software may not access physical memory behind the XMB nor XMB configuration.
Ready	The IMI sets this state when packet framing is completed, indicating that information can be sent and received on this IMI. Offset must be set before memory configuration accesses will work.	Configuration accesses to XMB.	Software may not write this value, but checks it to determine that the IMI is operational. Software may transition from this state to Reset.
Active	Link is fully configured and operational.	All	Software sets this state when the NB and XMB have been configured to support memory accesses. First, the memory on this link must be de-allocated so that no new memory requests are issued.
Redundant	This implies that the NB is performing Mirroring. The data on this port is duplicated on others either by Mirroring and no error has been detected on this IMI that causes the NB to ignore its contents. The NB can still continue operation if this IMI is taken out of service. Failures that would otherwise be logged as uncorrectable or fatal are now logged as correctable. Inband Fatal or Uncorrectable Signals from the XMB produce the effect of the Correctable Error signal.	All	Software writes this state to indicate that the contents behind an IMI can be reconstructed from the data on other IMI's.
Disabled	A Retry time-out or Fatal Error inband signal from XMB has occurred. This IMI is disabled. The NB is still operational. The NB is directing memory accesses to other IMI's, but it may take some time before this IMI reaches a quiescent state. Configuration accesses to this IMI are still supported, but configuration accesses to the associated XMB will master-abort. The NB will transition to this state from the redundant state when this IMI fails.	Transactions enqueued in the IMI may or may not be initiated. Any outstanding transactions complete. No new transactions from other ports are initiated.	Software may also write this value to take down a working redundant port for an upgrade. Software may not access physical memory behind the XMB nor XMB configuration.
Redundancy Loss	This IMI is no longer redundant. One of the IMI's with which this IMI was redundant is now disabled.	All	Software can return this IMI to redundant state when the all the other IMI's are recovered.

Table 6-15. Software Visible IMI State (Sheet 2 of 2)

State	NB Behavior	Supported IMI activity	SW Usage
Recovery Reset	Same as Reset State.	None	Software writes this value to clear the IMI in preparation for recovery. Software can force entry to this state whenever there are partners for this IMI. Once in this state, IMIHPC.PWRCTL can be set so that cards may be added or removed.
Recovery Init	Same as Initialization State.	HW controlled init ops. No system accesses.	Software writes this value to release the IMI reset and initiate recovery.
Recovery Ready	Similar to Ready State, but memory accesses are still being directed to/from other IMI's. Config accesses are enabled to IMI and to XMB.	Config Accesses, but not memory operations.	Software may not write this value. It checks this state to determine whether the NB has completed Recovery Init. SPD reads, XMB Address configuration, Calibration and Memory Test Engine.
Resilver	The NB is reconstructing the memory image on this IMI from its partners in Redundancy Loss State.	Re-silvering writes, System writes.	Software may set this state when the NB and XMB have been configured to support memory accesses. Software writes this value to initiate re-construction of the memory contents for this IMI.
Recovered	The images in the NB are consistent and ready to be put in redundant state. New reads and writes are already being processed the same as in the Redundant State. Any partners of this IMI in Redundancy Loss state should be returned at the same time.	All	Software may not write this value. It checks it to determine whether re-silvering succeeded.
Recovery Failed	An uncorrectable error occurred while re-silvering. The image on this IMI is not consistent with images on others and this IMI should not be advanced to Redundant state. Configuration reads and writes to this IMI are still supported, but configuration writes to the associated XMB will be master aborted.	None	Software may not write this value. It checks it to determine whether re-silvering succeeded. Software may attempt another recovery attempt or force this IMI into Reset State.

6.3.5 Memory RAS

6.3.5.1 Memory Mirroring

In the mirrored configuration, one of the IMI ports will be designated the primary image and the other the secondary (for an overview on mirroring, refer to [Section 2.6.3](#)). Both ports will be issued the same write request and both ports will respond to the request (in a ping-pong fashion). The primary designation will toggle between the two IMI ports on detected correctable memory responses. The primary designation will become fixed on a specific port on an uncorrectable error.

Memory Mirroring Write Operation- For the case of memory writes, the write request is issued to both IMI ports. The NB must successfully collect both responses before considering the write completion. If the write request fails on an IMI port, the write will be re-issued to both ports with exactly the same transaction ID. If the retried write is acknowledged by only one port, switch to the single image (no mirror available).

Memory Mirroring Read Operation- For the case of memory reads, the read request does not go to both mirrors. At the successful completion of the read response, the read response is allowed to complete on the front side bus, PCI Express or HII.5 interface.

In the case of a detected error, the action of the NB is as follows:

- In the case of a detected Correctable Error, the definition of the Primary image will toggle and the read will be issued on the “new” primary image.
- In the case of a detected Uncorrectable Error, the definition of the Primary image will “hard fail” to the other image. In this case, the “failed” image will never become primary until service has replaced the FRU (Field Replaceable Units) and “re-silvered” the images.

In the event of a switch between the primary and secondary port designations, the NB will ensure that the IMI ports are idle at the time of the switch.

Copy Engine (re-silvering the images) - In the event that the memory configuration has changed (as when a failed memory FRU has been replaced), the NB uses a copy engine to walk memory from the primary image and write it back to both ports. When the copy engine is enabled, all writes are issued to both ports.

The steps to copy (or re-silver two memory images) are as follows:

1. IMI Link goes through a training and initialization process.
2. XMB registers are initialized.
3. Memory behind the XMB is tested and initialized (“0” is written with valid ECC).
4. NB copy engine is enabled. At this time writes start being issued to both images.
5. When copy is completed, re-enable mirroring. (i.e. Failures of the primary memory image are no longer FATAL errors).

Correctable Errors - The NB will just correct the data and provide to the requester. The XMB will also correct the data and write back into memory.

Reasons to degrade to one memory image:

- Receiving two uncorrectable errors (not including poisoned data) for the same access.
- Receiving Fatal Error Signal from an XMB.
- Software disabling an IMI interface.

Re-enabling of memory mirroring:

- Replacement of a Failed Memory FRU via the Hot-Plug procedure.
- Software/Firmware will configure and test a memory card FRU. If the memory card matches in size with the current image, the re-silvering process can be enabled. Upon successful completion of the copying process, the two images can be re-enabled.

6.3.5.2 Memory RAID

Memory RAID Read Operation

Normal Case - Read Requests for a given Cache Line can be issued just as in the normal memory configuration to a given address with the following changes:

- RAID Address decoding must be done to route to the appropriate IMI Port.
- Address conflict check logic must take into account not only an address match for a write in progress, but also an address match of the Memory RAID associative lines. As indicated in Table 6-16, a read of "Cache Line 0" must be delayed to occur after any of the following writes of Cache Line 0.

Reconstructive Case - In the case of a failed IMI port, there are two cases for reading a cache line (for this discussion, we assume that IMI Port D has failed):

- To read Cache Line 0, the NB performs a normal read.
- To read Cache Line 2n (for example when IMI Port D is failed), is to read checkline(2n, n, 0), cache line n and cache line 0 and re-construct cache line 2n, as in the following equation:

$$cacheline(2n) = checkline \oplus cacheline(n) \oplus cacheline(0)$$

Memory RAID Write Operation

Normal Case - Write Request for a given cache line requires the cache line to be written and the check line to be updated accordingly. The steps of the write of cache line 0 are as follows:

1. Read Cache Line 2n and n.
2. Calculate Checkline, which is the exclusive OR of cache line 0, n, 2n.
3. Write Cache Line 0 and Check Line (2n,n,0).

Reconstructive Case - In the case of a failed IMI port, there are two cases for writing a cache line (for this discussion, we assume that IMI Port D has failed):

- To write Cache Line 0, the NB must read its checkline and cacheline 0 (old value); recalculate checkline, write new value of checkline and cacheline0. The calculation of the checkline is as follows:

$$checkline(new) = checkline(old) \oplus cacheline0(old) \oplus cacheline0(new)$$

- To write Cache Line 2 (assuming that IMI Port D is failed), is to update the checkline(2n+2, n+2, 2):

$$checkline(2n+2, n+2, 2) = cacheline(2n+2) \oplus cacheline(n+2) \oplus cacheline(2(new))$$

Memory RAID Re-silvering Operation

The re-silvering process for Memory RAID requires the NB to read from the other three IMI ports, regenerate the failed IMI Link data and then write the reconstructed data back. To reconstruct the failed IMI (for this example, IMI port D is being re-silvered), use the following equation:

$$IMID = IMIC \oplus IMIB \oplus IMLA$$

Note: As in the Memory Mirroring Case, all write operations must also happen to maintain cache coherency of the platform.

When the RAID re-silvering reaches terminal count, the re-silver link is available for normal operation. The size of the re-silvering is determined by the base and size programmed in IMIR3.

Cache Line Striping

For Memory RAID, the NB will use all four IMI Ports and provide a check cache line for every three cache lines. Table 6-16 describes how the IMI ports stripe the cache line. Also note the “n” in the table is equal to a fourth of all the memory available in the platform. For example, if a platform had 16GB of memory, “n” = 4 GB, 12 GB would be used as system address space and 4 GB is used for “Check lines”.

Table 6-16. Memory RAID IMI Port Striping

IMI RAID Associated Addresses	IMI Port D	IMI Port C	IMI Port B	IMI Port A
0	Cache Line 2n	Cache Line n	Cache Line 0	Check Line 2n, n, 0 (Check Line address - 3n)
1	Cache Line n+1	Cache Line 1	Check Line 2n+1, n+1, 1 (Check Line address - 3n+1)	Cache Line 2n+1
2	Cache Line 2	Check Line 2n+2, n+2, 2 (Check Line address - 3n+2)	Cache Line 2n+2	Cache Line n+2
3	Check Line 2n+3, n+3, 3 (Check Line address - 3n+3)	Cache Line 2n+3	Cache Line n+3	Cache Line 3

The following pseudo code provides a description of the algorithm to map a front side bus address onto an IMI Port. For readability, the code description ignores the TOLM calculation and the possibility of memory addition in the RAID mode.

If (in RAID mode) then

begin

n = 1/4(sum of total available DIMM memory if not in RAID mode)

if (n > FSB address >= 0) then use MIR0 Mapping register

else if (2n > FSB address >= n) then use MIR1 Mapping register

else if (3n > FSB address >= 2n) then use MIR2 Mapping register

else NOT a memory address and route to I/O (i.e. FSB address >= 3n)

*/*******

IMI Mapping Calculation

******/*

/ Determines ROW in Table 9-47 */*

RAID Address Offset = (FSB Address [7:6]);

/ Determine Column and IMI Port in Table 9-47 */*

if (n > FSB address >= 0) then IMI Port # = (1 + RAID Address Offset) && 011b

else if (2n > FSB address >= n) then IMI Port # = (2 + RAID Address Offset) && 011b

else then IMI Port # = (3 + RAID Address Offset) && 011b

end

Else (Map System Address per the description in Main Memory Interleaving subsection in the System Address Map chapter).

The check lines for the platform are at a “3n” base for the system address map. Refer to Table 6-17.

Table 6-17. Raid Related Address for the Independent Memory Interface Port

IMI RAID Associated Address	IMI Port D	IMI Port C	IMI Port B	IMI Port A
0	$0, n, 2n, 3n$	$0, n, 2n, 3n$	$0, n, 2n, 3n$	$0, n, 2n, 3n$
1	$1, n+1, 2n+1, 3n+1$	$1, n+1, 2n+1, 3n+1$	$1, n+1, 2n+1, 3n+1$	$1, n+1, 2n+1, 3n+1$
2	$2, n+2, 2n+2, 3n+2$	$2, n+2, 2n+2, 3n+2$	$2, n+2, 2n+2, 3n+2$	$2, n+2, 2n+2, 3n+2$
3	$3, n+3, 2n+3, 3n+3$	$3, n+3, 2n+3, 3n+3$	$3, n+3, 2n+3, 3n+3$	$3, n+3, 2n+3, 3n+3$

Memory Upgrade Support for the Memory RAID

The NB/XMB provide the ability to allow memory upgrade operations while in the RAID configurations. To allow this possibility the user must program the IMIR registers and have holes in the system address map. An example of this map follows:

```

-----FF_FFFF_FFFFh-----
High MMIO Range
-----4M+LMMIO_SIZE-----
System Parity lines and upgrade of Parity lines.
-----3M+LMMIO_SIZE+N-----
System Memory
-----3M+LMMIO_SIZE-----
Reserved for Memory Upgrade
-----2M+LMMIO_SIZE+N-----
System Memory
-----2M+LMMIO_SIZE-----
Reserved for Memory Upgrade
-----1M+LMMIO_SIZE+N-----
System Memory
-----1M+LMMIO_SIZE-----
Reserved for Memory Upgrade
-----LMMIO_SIZE+N-----
System Memory
-----01_0000_0000-----
Low MMIO HOLE
-----TOLM (01_0000_0000-LMMIO_SIZE)-----
System Memory and Legacy Ranges
-----00_0000_0000-----

```

Note: In the example map above, “N” is 1/4 the sum of all the currently installed memory in the platform and “M” represents the maximum amount of memory that is allowed per XMB after an upgrade. “M” is set by BIOS according to system vendor and/or user settings.

Table 6-18 shows the IMIR configuration for this example system map. The user of the platform might prefer not to allow for the possibility of a memory upgrade (this is due to certain OS limitations). The size of the re-silvering is determined by the base and size programmed in IMIR3 LIMIT.

Table 6-18. Raid Memory Upgrade IMIR Possible Settings

IMIR	LIMIT	Primary (data)				Secondary (parity)			
		Way3	Way2	Way1	Way0	Way3	Way2	Way1	Way0
0	M	0	3	2	1	3	2	1	0
1	2M	1	0	3	2	3	2	1	0
2	3M	2	1	0	3	3	2	1	0
3	(3M+N) ^a	3	2	1	0	3	2	1	0

NOTES:

a. This is due to accurately calculating the top of parity check blocks.

Figure 6-2 and Figure 6-3 illustrate the pre- and post-upgrade in memory RAID.

Figure 6-2. Pre-Upgrade Memory Map

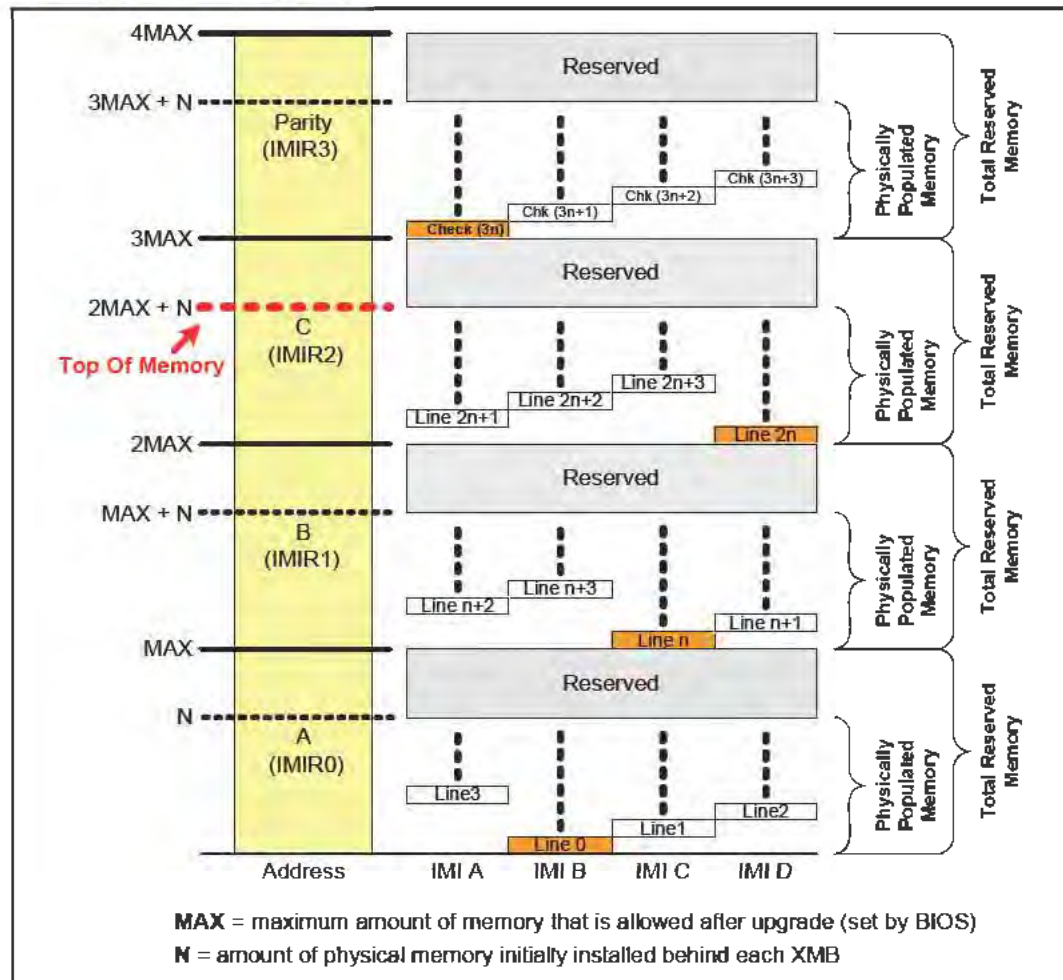
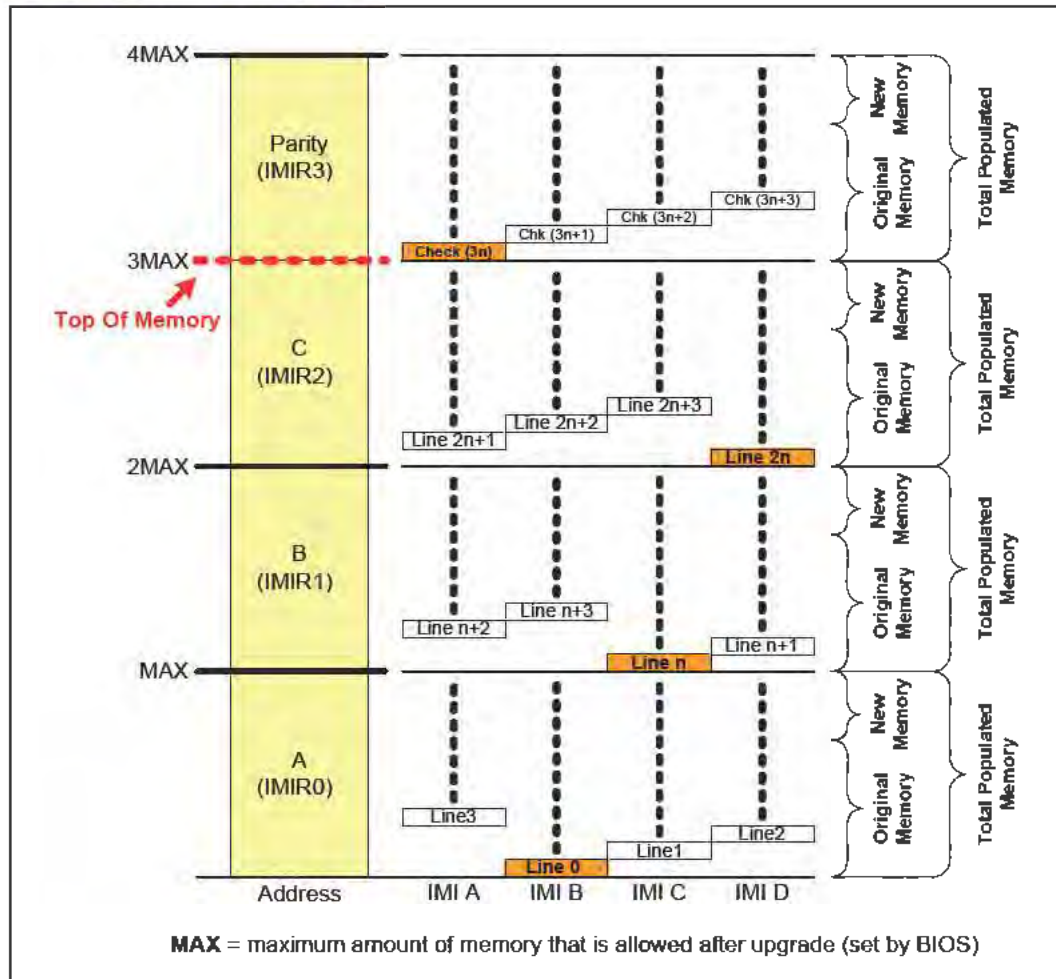


Figure 6-3. Post-Upgrade Memory Map



6.3.5.3 Patrol/Demand Scrubbing and DIMM Sparing

These RAS features are located entirely in the XMBs. For a description of these features, refer to *Intel® E8501 Chipset eXternal Memory Bridge (XMB) Datasheet*.

6.3.6 Initialization

The NB/XMB initialization sequence is mostly linear. The NB does not implement time-outs for each phase, but there is one time-out for the entire sequence. If the initialization protocol fails to advance, a time-out will cause the link to be reset and try again. When the time-out occurs, the NB will set IMIST.State to “disabled”. If the initialization succeeds, IMIST.State will be set to “Ready” (See [Section 4.13.9, “IMIST: IMI Status \(D {8,10,12,14}, F0\)” on page 148](#)). Once software configures the NB and XMB to handle memory transactions it writes IMIHPC.NextState to “Active”. This will be reflected in IMIST.State.

6.3.6.1 Initialization Triggers

All IMI’s are initialized whenever the NB is reset. An individual IMI is reset when IMIHPC.State is set to “Reset” or “RecoveryReset” for that IMI. Software will reset the IMI to perform Hot-Plug operations. No error detection by the NB directly causes an IMI to reset (an error may cause the system to be reset, in which case RSTIN# will be asserted).

6.3.7 Power Management

Intel® E8501 chipset does not support ACPI power-down of memory. Ports which are not connected to an XMB or connected to a failed XMB can be powered down by setting IMIHPC.STATE to Reset. This will hold the IMI in reset in which state drivers and receivers are not enabled. When in that state, the IMIHCP.PWRCTL can be set, which removes power from the XMB. This is a safe state that allows a memory card to be added without notification.

6.3.8 IMI Hot-Plug

The NB supports memory Hot-Plug feature that includes hot removal and hot addition.

These Hot-Plug operations are visible to the OS if the NB is operating in normal mode (non-redundant mode). Any Hot-Plug operation requires support from the OS including allocation and de-allocation of system memory.

If the NB is operating in mirroring or RAID mode in full redundant strength (not in redundancy fail over mode), the Hot-Plug operations are non-visible to the OS. Firmware/BIOS, however, is required to preform the required functionality for seamless addition and removal of XMB memory.

DIMM sparing (regular sparing or zero overhead sparing) on the DDR channels within an XMB is not within the scope of this document. Readers should refer to the *Intel® E8501 Chipset eXternal Memory Bridge (XMB) Datasheet* for information on this topic.

6.3.8.1 Hot-Plug Standard Usage Model

The NB IMI memory Hot-Plug mechanism is designed to allow the system platform to provide the same visual indicators, buttons, and control/status registers as PCI Standard Hot-Plug Controller (SHPC) Specification and PCI Express Hot-Plug standard usage model. The NB IMI memory Hot-Plug mechanism requires support from Intel® E8501 chipset-specific firmware to follow Hot-Plug standard usage model. Intel® E8501 chipset-specific firmware and interrupt service routines together are referred to as software (SW) in this section.

The memory board is a field-replaceable unit or Hot-Pluggable unit. Each memory board implements an attention button, power LED (green), indicator/attention LED (amber), and a Mechanical Retention Latch (MRL) switch to support the standard usage model and to support surprise removal without damage. In this chapter, green LED and power LED are used interchangeably. Indication/attention LED and amber LED are also used interchangeably.

6.3.8.2 Hot-Plug Registers

Each IMI has an IMIHPC (Independent Memory Interface Hot-Plug Control) register to control:

1. LED operations
2. Power to the XMB
3. Hot-Plug interrupt enabling mask
4. Next state of the IMI

Each IMI also has a IMIST (Independent Memory Interface Status) register that provides status for:

- Hot-Plug interrupt events
- Current state of the IMI

The NB supports only one IMI Hot-Plug operation at a time. The IMIST register has sufficient information to determine which IMI is being sequenced through the Hot-Plug operation. The IMIHPC register has interrupt enable/disable capability to prevent events from other IMI's from intruding on the current Hot-Plug operation. On completion of any Hot-Plug operation, software must check the status of other IMI's to handle any pending events.

6.3.8.3 Interrupt Support for Hot-Plug

6.3.8.3.1 Event Interrupts

The NB has a pin, IMI_HPINT, dedicated to IMI Hot-Plug interrupts. The 4 IMIST registers have sufficient status information to determine the cause of any IMI_HPINT interrupt.

Interrupt names and associated events triggering IMI_HPINT include:

- XMBINT: XMB has sent an “assert IMI_HPINT” in-band signal on the Independent Memory Interface. Status register in XMB provides the more interrupt information for SW.
- IMIRLINT: MRL sensor change detected whenever the MRL is transition between open and closed positions.
- ATNINT: Attention button pressed is detected for this IMI.
- FAILINT: An IMI has entered the disabled state.
- PRSINT: Present detect changed when PRSNT# changes state.
- PWRINT: Power fault asserted in an IMI.

6.3.8.3.2 Hot-Plug Timer Interrupt

The IMI_HPTIM register controls a timer dedicated to IMI Hot-Plug software that will interrupt after intervals ranging from 12.5 ms to 12.5 sec. Instead of individual timers for each IMI, there is only 1 IMI_HPTIM timer for SW use.

6.3.9 NB IMI States

NB IMIHPC and IMIST registers define the software-visible states of the NB IMI. Transitions from one state to another may be controlled by software or the NB. Software can observe these states via the IMIST register and force into a state via the IMIHPC register.

Typically, the NB only supports one IMI at a time in the Recovery states. Candidates for recovery wait in the “Disabled” state until they are serviced. Configuration accesses are master aborted when the IMI is not in a fully operational state. Otherwise, an access to a redundant failed port (by error handling, or Hot-Plug/ recovery software) could result in an unpredictable behavior.

6.3.10 Hot Swap

The NB Memory Port Hot-Plug capability and RAS operating modes enable a hot swap process which consists of a hot removal and a hot addition of a memory board. Hot swap enables the system to electrically isolate a memory board (with hardware error or by user specification) while the system continues to run (such that the memory board could be replaced without taking the system off-line).

6.3.11 NB RAS Operating Modes

Once the RAS operating mode (Normal, Mirroring or RAID) is programmed at boot time, it should not be changed during the Hot-Plug process.

If a system is operating in mirroring mode, either with mirroring pairs {(IMI A, IMI B), (IMI C, IMI D)} or {(IMI A, IMI C), (IMI B, IMI D)}, one XMB/IMI from each mirroring pair (for a total of 2) could be hot removed sequentially. Since all 4 IMI's are participated in providing redundancy, only one XMB/IMI could be hot removed. Once the IMI slot is powered down by the system, it could be replenished by a memory board with adequate capacity/capability by hot addition.

6.3.12 Memory Hot Removal

There are two mechanisms to trigger a hot removal sequence:

- Hardware initiated hot removal
- Software/user initiated hot removal

6.3.12.1 Hardware Initiated Hot Removals

In this case, NB detects an interrupt and reports it via IMIST register. The interrupt is due to hardware error or surprise removal of XMB card). These interrupts include:

- XMB interrupt: XMB has sent an “assert IMI_HPINT” in-band signal on the Independent Memory Interface. An example is XMB leaky bucket threshold overflow event (for normal mode only).
- IMI entering disabled state in mirroring or RAID mode only
- Mechanical Retention Latch (MRL) sensor changed
- Presence detect change
- Power fault detection

6.3.12.2 Software/User Initiated Hot Removals

This mechanism is mainly for preventative maintenance or add-on upgrade purposes. NB could support at least two approaches:

- Pushing the attention button on a XMB card to indicate hot removal intention
- User requests hot removal through system software user interface

The following three subsections discuss hot removal flows for three operating modes:

- 1. Normal (non-redundancy),**
- 2. Mirroring**
- 3. RAID**

Proper interrupt enable bits are assumed to be programmed correctly by software (Intel® E8501 chipset-specific firmware and interrupt handling routines).

6.3.12.2.1 Normal Mode (Graceful Degradation)

Figure 6-4 illustrates a hot removal flow in normal mode. The IMI_x is the IMI to be hot removed where x could be A, B, C or D. IMI_x is at active state (IMIST[x].state = active) entering the flow. There are five entry points in this flow: 1, 2, 3, 4 and 5. Each represents an independent entry point.

At point 1, the XMB detects a leaky bucket threshold overflow and sends a IMI in-band “assert IMI_HPINT” signal to the NB. The NB sets IMIST[x].XMBINT that triggers a IMI_HPINT, software sets the amber attention LED via IMIHPC[x].ATNLED to indicate operational problem and processes this interrupt and exits.

At point 2, the software initiates a graceful degradation to remove IMI_x and set the indication LED. A technician can initiate the removal process via pushing the attention button at point 5.

Prompted by the attention LED, the user/technician pushes the attention button following the standard usage model. The attention button pressed event triggers IMI_HPINT interrupt. The servicing software sets the green power LED to blinking state via IMIHPC[x].PWRLED and turns off attention LED. It also clears the IMIST[x].ATNINT and sets IMI_HPTIM timer to interrupt after five seconds and then exits. The NB will set IMI_HPTIM.TIMER after the programmed five second period and triggers a IMI_HPINT. If the button is pushed again during this five second period, the user indicates intention NOT to remove this IMI per standard Hot-Plug usage model. In that case, software turns off IMIHPC[x].PWRLED, clears the IMIST[x].ATNINT and processes the event and exits.

If the user’s intention is confirmed after the five second period, the software and OS will validate the removal request against system policy (e.g. pinned memory that can not be de-allocated or physical security policy).

Once the request is validated or the NB detects a power fault (entry point C with proper software handling), or the MRL sensor changed (entry point D with proper software handling), OS will de-allocate the system memory in this IMI/XMB. Software will flush NB CDC via setting CDCC.FLUSH bit which starts a NB CDC flushing process. When the NB completes the flushing process, it sets CDCS.FLUSH.DONE bit. Software can use the IMI_HPTIM to poll this completion bit periodically. Recommended waiting time is 25 ms.

After the CDC flush is completed, software should set IMIHPC[x].state to reset state. The NB will put the IMI into reset state. Software then turns off power to the IMI/XMB via IMIHPC[x].PWRCTL. The Power LED is then turned off via IMIHPC[x].PWRLED by software. At this point, the IMI port is in reset and power to the XMB slot is turned off. Since the power LED is off, the user can remove the XMB per standard usage model.