Using ECC Memory Configuration and Error Injection and Detection with the Tsi107™

80C2000_AN005_03

November 3, 2009

6024 Silver Creek Valley Road San Jose, California 95138
Telephone: (408) 284-8200 • FAX: (408) 284-3572
Printed in U.S.A.

About this Document

One of the most important aspects of system memory is ensuring the integrity of the data contained within. Configuring the memory array to contain check bits is the most common method. Two common implementations that take advantage of the extra bits are parity and Error Detection and Correction (also called ECC for Error Correcting Code). The Tsi107 PowerPC™ provides hardware support for both implementations.

For basic integrity checking and notification of corrupt data, the parity checking feature of the bridge is sufficient. However, in applications where the presence of a corrupt bit in memory can cause a serious situation, and the delay in recovering from this situation is unacceptable, the use of ECC is desired. The ECC controller in the Tsi107 has the capability of detecting and repairing one corrupt bit in a word at the memory interface speed, and detecting and generating a notification of the presence of multiple corrupt bits.

This application note is intended to provide the user the necessary understanding and processes required to initialize the Tsi107 memory controller, configure the error detection registers for error notification and intentionally generate errors in the memory array in order to validate the detection and correction circuitry.

Most of the information contained herein, has been extracted from the Tsi107 User Manual. The content flow has been changed in order to aid the user in gaining a better understanding of the registers and operation of the SDRAM memory interface.

This document discusses the following topics:

- “Tsi107 Memory Controller Architecture”
- “SDRAM In-Line ECC”
- “Debug Registers”
- “Watchpoint Control Register (WP_CONTROL)”
- “How the error injection logic is put to use”
- “Sample Code”

Revision History

80C2000_AN005_03, Formal, November 2009

This version of the document was rebranded as IDT. It does not include any technical changes.

80C2000_AN005_02, Formal, March 2007

This version fixed a typographical error.

80C2000_AN005_01, Formal, March 2004

This is the first release of the application note.

1. Tsi107 Memory Controller Architecture

The Tsi107 integrates a high-performance memory controller that controls processor and PCI interactions to local memory. The Tsi107 supports various types of DRAM and ROM/Flash configurations as local memory.

- SDRAM
  - SDRAMs must comply with the JEDEC specification for SDRAM
  - High-bandwidth bus (32- or 64-bit data bus) to SDRAM
  - One-Mbyte to 1-GB SDRAM memory—1 to 8 chip selects for SDRAM bank sizes ranging from 1 MB to 512 MB per bank
  - Supports page mode SDRAMs—four open pages simultaneously
  - Programmable timing for SDRAMs
- DRAM—fast page mode (FPM) and extended data out (EDO)
  - High-bandwidth bus (32- or 64-bit data bus) to DRAM
  - One-Mbyte to 1-GB DRAM memory space
  - One to eight chip selects of 4-, 16-, 64- or 128-Mbit memory devices
  - Programmable timing for FPM and EDO
- ROM/Flash
  - 144 Mbytes of ROM/Flash space can be divided between the PCI bus and the memory bus (8 Mbytes each)
  - Supports 8-bit asynchronous ROM or 64-bit burst-mode ROM
  - Configurable data path—8-, 32-, or 64-bit
  - Supports bus-width writes to Flash
- Port X—The ROM/Flash controller can interface any device that can be controlled with an address and data field (communication devices, DSPs, general purpose I/O devices, or registers). Some devices may require a small amount of external logic to properly generate address strobes and chip selects.

  - 8-bit Port X
  - 32-bit Port X
  - 64-bit Port X—the floating-point (FPU) unit in the CPU must be present for 64-bit writes
- Data path buffering—72 bits (64-bit data and 8-bit parity)
  - Reduces loading on the internal processor core bus
  - Reduces loading of the drivers of the memory system
  - Reduces signal trace delay known as time-of-flight (TOF)
- Parity—Supports normal parity and read-modify-write (RMW)
- Error checking and correction (ECC)—64-bit only
  - DRAM ECC—Located in the central control unit (CCU)
  - SDRAM ECC—Located in-line with the data path buffers

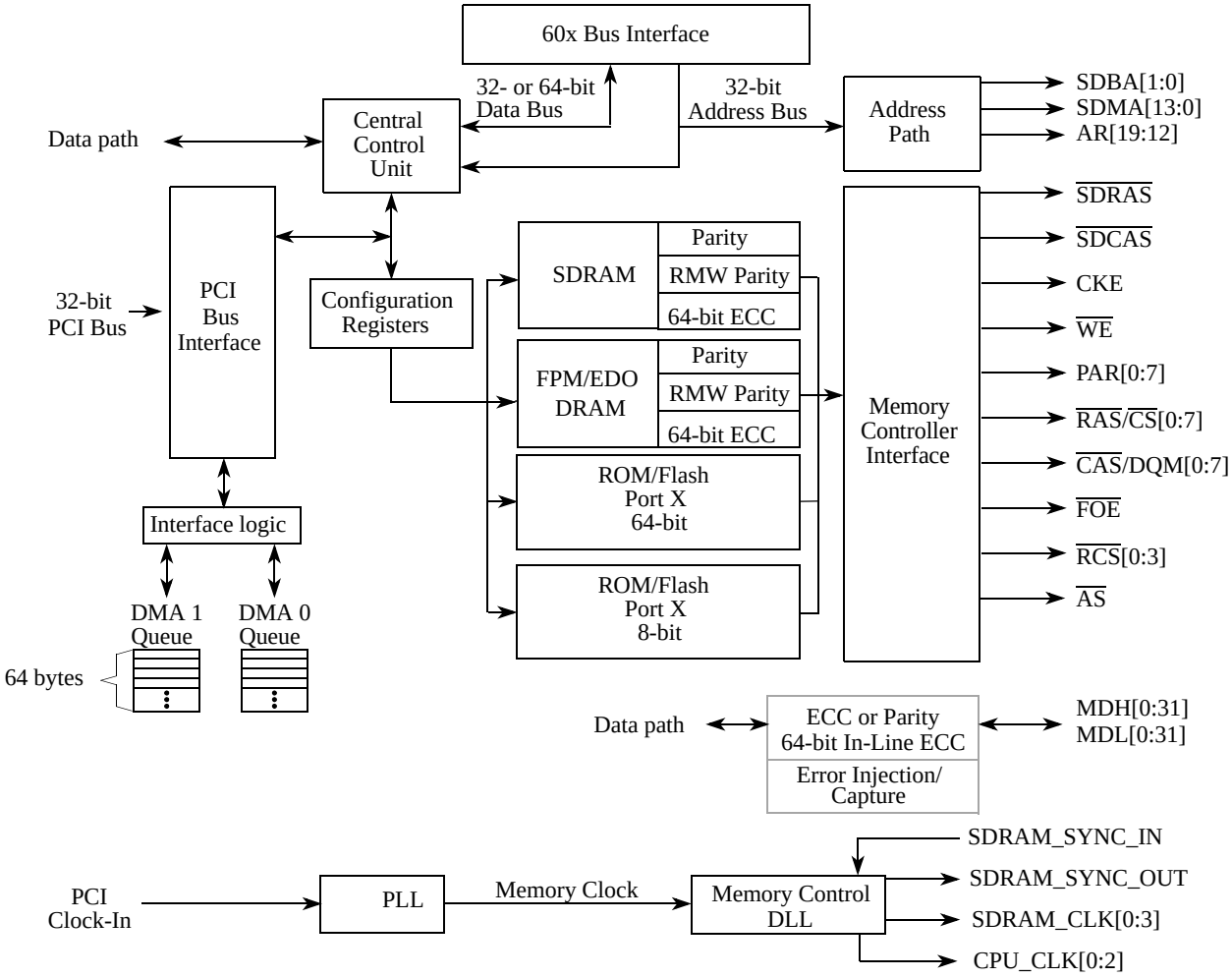
The Tsi107 is designed to control a 32- or 64-bit data path to main memory (SDRAM or DRAM). The Tsi107 can also be configured to check parity or ECC on memory reads. Parity checking and generation can be enabled with 4 parity bits for a 32-bit data path or 8 parity bits for 64-bit data path. Concurrent ECC is only generated for a 64-bit data path with 8 syndrome bits. The Tsi107 supports SDRAM or DRAM bank sizes from 1 to 512 Mbytes and provides bank start address and end address configuration registers. However the Tsi107 does not support mixed SDRAM or DRAM configurations. The Tsi107 can be configured so that the appropriate row and column address multiplexing occurs for each physical bank. Addresses (DRAM or SDRAM) and bank selects (SDRAM only) are provided through a 15-bit interface for SDRAM and 13-bit interface for DRAM.

ROM/Flash systems are supported by up to 24 address bits, 4 bank selects, 1 write enable and 1 output enable.

Figure 1 is a block diagram of the memory interface.

SDRAM is the most commonly available type of memory used with the Tsi107. For this reason, this application note will focus on the use of SDRAM in an ECC application.

Figure 1: Block Diagram for Memory Interface



1.1 SDRAM Interface Operation

Figure 2 shows an internal block diagram of the SDRAM interface of the Tsi107.

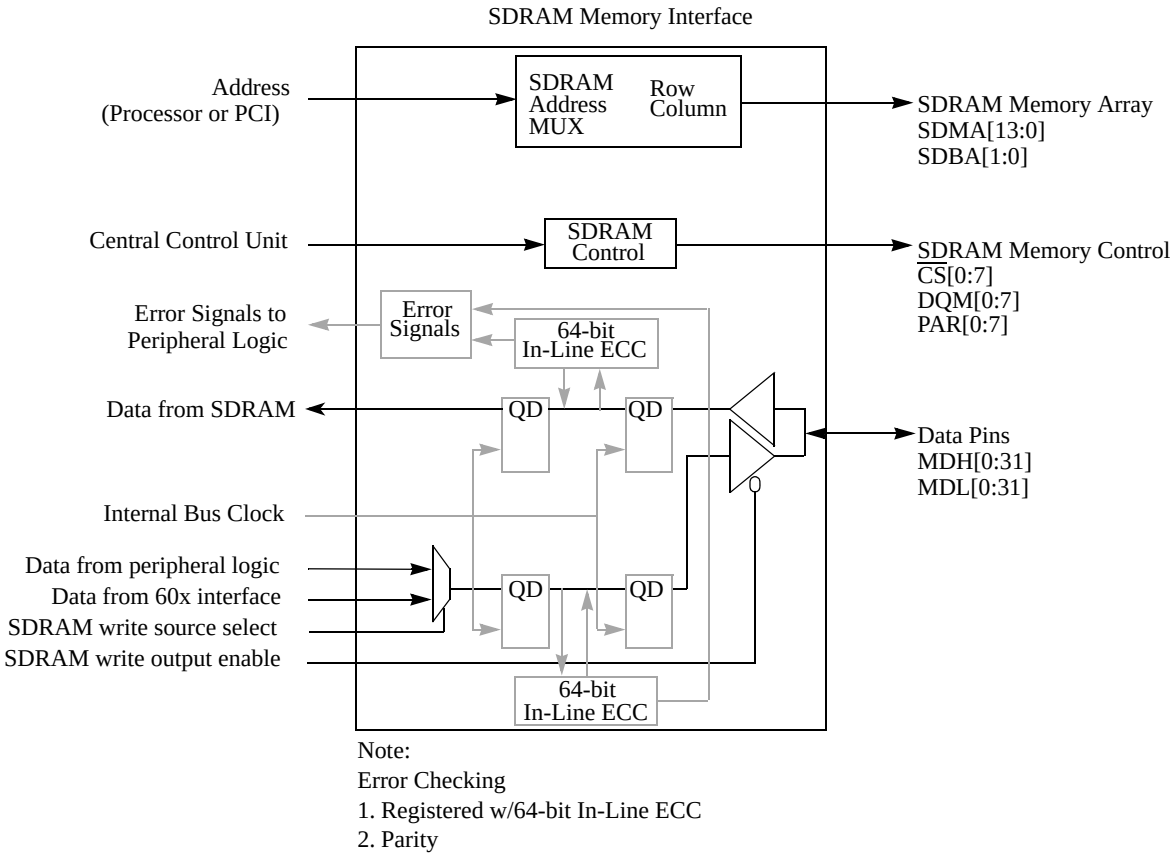
The Tsi107 provides control functions and signals for JEDEC-compliant SDRAM. The Tsi107 supplies the SDRAM_CLK[0:3] to be distributed to the SDRAM. These clocks are the same frequency and are in phase with the memory bus clock. The other control signals provided for the interface are a write enable signal (WE), a row address strobe signal (SDRAS), a column address strobe signal (SDCAS), a memory clock enable signal (CKE),

The SDRAM memory bus can be configured to be 64 bits (72 bits with parity) requiring a four-beat SDRAM data burst, or configured to be 32 bits (36 bits with parity) requiring an eight-beat SDRAM data burst.

The data width of the device determines its density and the physical bank size. Eight chip select signals (CS[0:7]) support up to eight banks of memory. The row, column and bank multiplexing is flexible enough to allow the banks to be built of x1, x4, x8, x16, or x32 SDRAMs as they become available. Collectively, these interface signals allow a total of 1 GB of addressable memory.

The thirteen row/column multiplexed address signals (SDMA[12:0]) in conjunction with two bank select signals (SDBA[1:0]) provide SDRAM addressing for up to 64 M. SDMA[13] is only used on the ROM/FLASH port in addressing as AR[23].

Figure 2: SDRAM Interface



Note:
Error Checking
1. Registered w/64-bit In-Line ECC
2. Parity

Programmable CAS latency is supported for data read operations and is set through the SDMODE parameter (MCCR4 bits 14-12). For write operations, the first beat of write data is supplied concurrent with the write command. The memory design must be byte-selectable for writes using the Tsi107's DQM outputs.

The Tsi107 allows for four simultaneous open pages for page mode operation. The number of clocks for which the pages are maintained open is programmable by the BSTOPRE and PGMAX parameters. Page register allocation uses a “least recently used” (LRU) algorithm.

Table 1 below shows how the SDRAS,SDCAS and bank select bits are multiplexed into the physical addresses of the SDRAM interface to convert from the linear addressing mode of the 60x architecture to the SDRAM architecture.

Table 1: SDRAM Address Multiplexing SDMA[12:0] and SDBA[1:0] —64-Bit Mode

Row x Col x Bank		msb Physical Address lsb																															
		0-2			3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
11x10x2	SDRAS										BA0	10	9	8	7	6	5	4	3	2	1	0											
	SDCAS								9	8	BA0												7	6	5	4	3	2	1	0			
11x9x2	SDRAS										BA0	10	9	8	7	6	5	4	3	2	1	0											
	SDCAS									8	BA0												7	6	5	4	3	2	1	0			
11x8x2	SDRAS										BA0	10	9	8	7	6	5	4	3	2	1	0											
	SDCAS										BA0												7	6	5	4	3	2	1	0			
13x10x2	SDRAS								11	12	BA0	10	9	8	7	6	5	4	3	2	1	0											
	SDCAS						9	8			BA0												7	6	5	4	3	2	1	0			
13x9x2	SDRAS								11	12	BA0	10	9	8	7	6	5	4	3	2	1	0											
	SDCAS							8			BA0												7	6	5	4	3	2	1	0			
13x8x2	SDRAS								11	12	BA0	10	9	8	7	6	5	4	3	2	1	0											
	SDCAS										BA0												7	6	5	4	3	2	1	0			

Table 1: SDRAM Address Multiplexing SDMA[12:0] and SDBA[1:0] —64-Bit Mode *(Continued)*

Row x Col x Bank		msb Physical Address lsb																																
		0-2			**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**
12x11x4[a]	SDRAS								11	BA1	BA0	10	9	8	7	6	5	4	3	2	1	0												
	SDCAS					11	9	8		BA1	BA0												7	6	5	4	3	2	1	0			
12x10x4	SDRAS								11	BA1	BA0	10	9	8	7	6	5	4	3	2	1	0												
	SDCAS						9	8		BA1	BA0												7	6	5	4	3	2	1	0			
12x9x4	SDRAS								11	BA1	BA0	10	9	8	7	6	5	4	3	2	1	0												
	SDCAS							8		BA1	BA0												7	6	5	4	3	2	1	0			
11x8x4 or 12x8x4	SDRAS								11	BA1	BA0	10	9	8	7	6	5	4	3	2	1	0												
	SDCAS									BA1	BA0												7	6	5	4	3	2	1	0			
13x11x4[a]	SDRAS							12	11	BA1	BA0	10	9	8	7	6	5	4	3	2	1	0												
	SDCAS				11	9	8			BA1	BA0												7	6	5	4	3	2	1	0			
13x10x4	SDRAS							12	11	BA1	BA0	10	9	8	7	6	5	4	3	2	1	0												
	SDCAS					9	8			BA1	BA0												7	6	5	4	3	2	1	0			
13x9x4	SDRAS							12	11	BA1	BA0	10	9	8	7	6	5	4	3	2	1	0												
	SDCAS						8			BA1	BA0												7	6	5	4	3	2	1	0			

Table 1: SDRAM Address Multiplexing SDMA[12:0] and SDBA[1:0] —64-Bit Mode *(Continued)*

Row x Col x Bank		msb Physical Address lsb																														
		0-2			3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
13x8x4	SDRAS							12	11	BA1	BA0	10	9	8	7	6	5	4	3	2	1	0											
	$\overline{\text{SDCAS}}$									BA1	BA0												7	6	5	4	3	2	1	0			

a. For SDRAMs with 11 column bits, SDMA10 is driven low during the $\overline{\text{SDCAS}}$ phase to indicate a read or write without autoprecharge and SDMA11 is used as the 11th column bit.

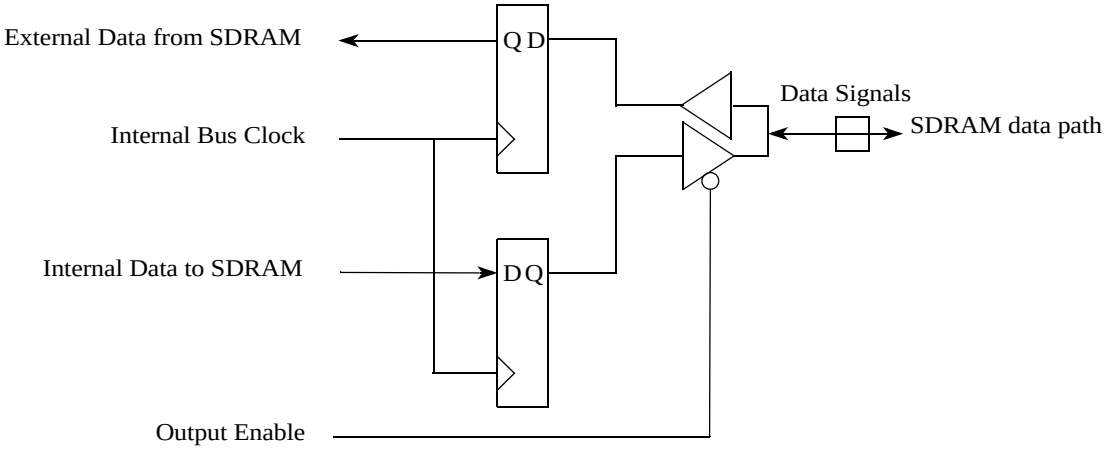
1.2 SDRAM Memory Data Interface

To reduce loading on the data bus, the Tsi107 features on-chip buffers between the 60x data bus and the memory data bus. There are sixty four bidirectional data signals (MDH[0:31] and MDL[0:31]), and eight bidirectional data parity signals (PAR[0:7]). The PAR[0:7] signals also perform as ECC syndrome bits when the controller is running in ECC mode.

Eight SDRAM data in/out mask signals (DQM[0:7]) are provided to control the byte lane selection for 32- and 64-bit accesses. The architecture is constructed with the premise that an 8-bit SDRAM device has one DQM signal and eight data signals (DQ[0:7]), and a 16-bit SDRAM device has two DQM signals associated to specific halves of the sixteen data signals (DQ[0:7] and DQ[8:15]).

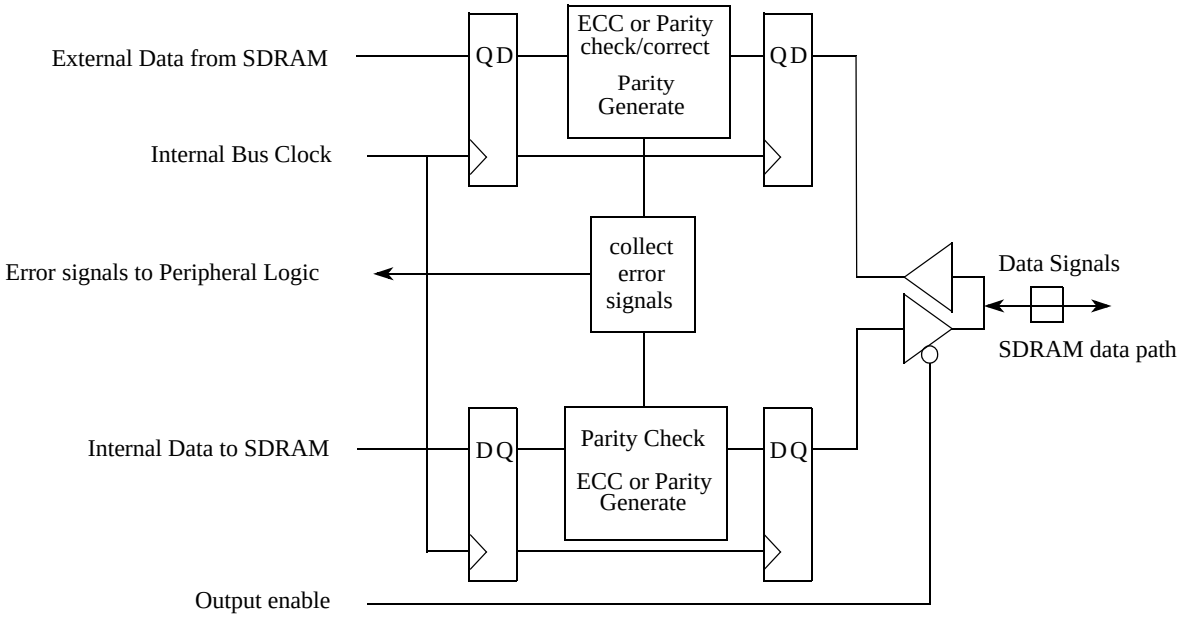
The Tsi107 supports two types of internal data path buffering for the SDRAM data interface, registered and in-line. Registered buffer mode is the default mode for the Tsi107 and has a latency of one clock cycle. Figure 3 shows the basic structure of the registered interface. This structure is used to correct the setup and hold times associated with the interface.

Figure 3: SDRAM Registered Memory Interface



The use of ECC or parity for data error management requires that the data be held for one clock cycle while the respective syndrome code or parity bit generation and checking logic perform their function. This required the introduction of an additional set of latches as shown in Figure 4. The detailed operation of the In-line ECC function is described in Section 2.

Figure 4: SDRAM In-Line ECC/Parity Memory Interface



1.3 Memory Datapath Configuration and Control Registers

The Memory Configuration and Control Registers (MCCR 1 to 4) are used to set up the operating parameters of the memory interface. The configuration information can be split into two categories, memory data path parameters and memory device operating parameters. Table 2 lists the names and locations of the parameters that determine the data path buffer mode and the parity or ECC operation of the Tsi107. Following the table are descriptions of the individual bits. The memory device operating parameters are device specific. The bit fields and detailed descriptions are contained in Section 1.4 on page 16.

Table 3 describes the parameter settings for the available SDRAM data path buffer options. Note that combinations of configuration register bit settings that are not specified in Table 3 have undefined behavior.

This application note concerns itself with the last entry of Table 3, titled “In-line, ECC enabled”. For information on the remaining configurations, refer to the *Tsi107 User Manual*.

Table 2: Memory Data Path Parameters

Bit Name	Register and Offset	Bit Number in Register
RAM_TYPE	MCCR1 @F0	17
EDO	MCCR2 @F4	16
PCKEN	MCCR1 @F0	16
INLINE_WR_EN	MCCR2 @F4	19
INLINE_RD_EN	MCCR2 @F4	18
INLINE_PAR_NOT_ECC	MCCR2 @F4	20
BUF_TYPE[0]	MCCR4 @FC	22
BUF_TYPE[1]	MCCR4 @FC	20
RMW_PAR	MCCR2 @F4	0
ECC_EN	MCCR2 @F4	17
MEM_PARITY_ECC_EN	ErrEnR1 @C0	2
MB_ECC_ERR_EN	ErrEnR2 @C4	3

1.3.1 RAM TYPE MCCR1 bit 17

The RAM TYPE bit controls the operation of the memory controller. This determination is necessary as the controller is required to communicate some initialization information to SDRAM type devices, but not to FPM or EDO type devices.

0) Indicates synchronous DRAM (SDRAM).
1) Indicates FPM DRAM or EDO DRAM (depending on the setting for MCCR2[EDO])

For the purpose of this app note, this bit must be cleared.

Note that this bit must be cleared (selecting DRAM or SDRAM) before the in-line or registered buffer mode bits in MCCR4 are set.

1.3.2 EDO MCCR2 bit 16

The EDO enable allows the memory controller to differentiate between FPM and EDO type devices. See the *Tsi107 User Manual*, Section 6.3, “FPM or EDO DRAM Interface Operation” for more information.
0) Indicates standard DRAMs
1) Indicates EDO DRAMs

For the purpose of this app note, this bit must be cleared.

1.3.3 PCKEN MCCR1 bit 16

This bit enables parity checking/generation in the memory interface.
0) Disables parity checking and parity generation for transactions to DRAM/EDO/SDRAM memory. Note that this bit must be cleared for SDRAM memory when operating in in-line buffer mode (MCCR4[BUF_TYPE[0–1]] = 0b10) and in-line parity/ECC is enabled with MCCR2[INLINE_RD_EN] = 1.
1) Enables parity checking and generation for all registered mode memory transactions to DRAM/EDO/SDRAM memory when using the registered mode of operation.

For the purpose of this app note, this bit must be cleared.

1.3.4 INLINE_WR_EN MCCR2 bit 19

This bit is called In-line parity error reporting enable and is used for SDRAM only. This bit controls whether the Tsi107 uses the in-line parity hardware to report 60x bus parity errors on writes to memory. Note that the buffer type selector in MCCR4 must be set to in-line buffer mode (MCCR4[BUF_TYPE[0–1]] = 0b10) to enable the in-line ECC/parity logic.
0) In-line 60x bus parity error reporting disabled.
1) In-line 60x bus parity error reporting enabled.

For the purpose of this app note, this bit must be cleared.

1.3.5 INLINE_RD_ EN MCCR2 bit 18

This is a multi-function bit and is called “In-line read parity or ECC check/correction enable”. This bit controls whether the Tsi107 uses the ECC/parity checking and/or correction hardware in the in-line data path to report ECC or parity errors on memory system read operations. This bit activates different parity/ECC checking/correction hardware than that controlled by ECC_EN and PCKEN. Read parity/ECC checking can be enabled for SDRAM systems running in in-line buffer mode (MCCR4[BUF_TYPE[0–1]] = 0b10) only. Also, note that when the INLINE_RD_EN bit is used in conjunction with the INLINE_PAR_NOT_ECC bit, the INLINE_PAR_NOT_ECC bit selects between parity or ECC on the memory data bus.
0) In-line memory bus read parity/ECC error reporting disabled
1) In-line memory bus read parity/ECC error reporting enabled. Note that MCCR1[PCKEN] must be cleared when this bit is set.

For the purpose of this app note, this bit must be set.

1.3.6 INLINE_PAR_NOT_ECC

This bit selects between the ECC and parity checking/correction mechanisms of the in-line data path when performing memory reads. This bit is applicable for SDRAM systems running in in-line buffer mode only (MCCR4[BUF_TYPE[0–1]] = 0b10), and when INLINE_RD_EN = 1.
0) Tsi107 uses ECC on the memory data bus.
1) Tsi107 uses parity on the memory data bus.

For the purpose of this app note, this bit must be cleared.

1.3.7 BUF_TYPE[0] MCCR4 bit 22 and BUF_TYPE[1] MCCR4 bit 20

The most significant bit of the memory data bus buffer-type field. BUF_TYPE[0] is used with bit 20 (BUF_TYPE[1]) to configure the internal memory data path buffering scheme as follows:
BUF_TYPE[0–1]:
00) Reserved
01) Registered buffer mode (default)
10) In-line buffer mode; SDRAM only
11) Reserved

The Tsi107 must be configured for in-line buffer mode in order to use the in-line ECC/parity logic for SDRAM. The in-line ECC and parity hardware allow the Tsi107 to check and generate parity on the 60x bus and check, correct and generate ECC or parity on the external SDRAM memory bus. See Section 1.2, “SDRAM Memory Data Interface” for more information.

For the purpose of this app note, BUF_TYPE[0] must be set, and BUF_TYPE[1] must be cleared.

1.3.8 RMW_PAR MCCR2 bit 0

Read-modify-write (RMW) parity enable. This bit controls how the Tsi107 writes parity bits to DRAM/EDO/SDRAM. Note that this bit does not enable parity checking and generation. PCKEN must be set to enable parity checking. Also note that this bit and ECC_EN cannot both be set to 1. See Table 3 for valid combinations.
0) RMW parity disabled
1) RMW parity enabled. Note that this bit must be set for SDRAM systems that use in-line ECC in combination with (MCCR2[ECC_EN] = 0, MCCR4[BUF_TYPE[0–1]] = 0b10, and MCCR2[INLINE_PAR_NOT_ECC]] = 0).

For the purpose of this app note, this bit must be set.

1.3.9 ECC_EN MCCR2 bit 17

This bit controls whether the Tsi107 uses ECC for transactions to system memory that is constructed of FPM or EDO devices. ECC_EN can be set only for systems using FPM/EDO memory. Note that the ECC_EN parameter overrides the PCKEN parameter. Also note that this bit and RMW_PAR cannot both be set, and it is illegal to set this bit with the combination of EDO = 1 and REGISTERED = 1. Systems using SDRAM use a different (in-line) ECC hardware and therefore, must have ECC_EN = 0.
0) ECC disabled
1) ECC enabled

For the purpose of this app note, this bit must be cleared.

1.3.10 Mem_parity_ECC_en ErrEnR1 bit 2

This bit is called Memory Parity and ECC Error Enable. It enables the detection of system memory read parity errors that occur on accesses to system memory or those that occur that exceed the ECC single-bit error threshold. The threshold is set in the ECC Single-bit Error Trigger Register, and the error occurrence is counted in the ECC Single-bit Error Counter Register. The fact that the threshold was exceeded is indicated in Error Detection Register 1, bit 2.
0) Memory read parity/ECC single-bit threshold disabled
1) Memory read parity/ECC single-bit threshold enabled

For the purpose of this app note, this bit must be set.

1.3.11 MB_ECC_ERR_EN ErrEnR2 bit 3

This bit is called the Multi-bit ECC Error Enable. This bit enables the detection of ECC multibit errors. The occurrence is reported in Error Detection Register 2, bit 3.
0) ECC multi-bit error detection disabled
1) ECC multi-bit error detection enabled

For the purpose of this app note, this bit must be set.

Table 3: SDRAM System Configurations

RAM_TYPE	EDO	PCKEN	INLINE_WR_EN	INLINE_RD_EN	INLINE_PAR_NOT_ECC	BUF_TYPE[0]	BUF_TYPE[1]	RMW_PAR	ECC_EN	MEM_PARITY_ECC_EN	MB_ECC_ERR_EN	Description	Supports 60x Bus Parity on Processor Reads from SDRAM	Supports 60x Bus Parity on Processor Writes from SDRAM
0	0	0	0	0	0	0	1	0	0	0	0	Registered, no ECC or parity	No	No
0	0	1	0	0	0	0	1	0	0	1	0	Registered buffer parity	Yes	No
0	0	1	0	0	0	0	1	1	0	1	0	Registered buffer RMW parity	Yes	No
0	0	0	0	0	0	1	0	0	0	0	0	In-line, no parity	No	No
0	0	0	1	1	1	1	0	0	0	1	0	In-line, parity enabled	Yes	No
0	0	0	0	1	1	1	0	1	0	1	0	In-line, RMW parity enabled	Yes	Yes
0	0	0	0	1	0	1	0	1	0	1	1	In-line, ECC enabled	No	No

1.4 Bit settings for SDRAM System Configurations

In order to configure a fully operational memory system, a number of memory device specific parameters are required to be programmed into the MCCR registers. These parameters tell the controller when to make write data valid on the memory bus and when to expect read data from the memory devices on the bus.

The registers will be examined sequentially from MCCR1 to MCCR4, however, when the user actually writes to the Tsi107, MCCR1 will need to be written a second time with the “MEMGO” bit set to command the controller to begin initializing the SDRAM memory devices with the correct control patterns.

The intent of this app note is to explain to the reader what parameters are required in the related bit fields. The compilation of the data word to be written into the register is left up to the reader. The bit field descriptions have been focused on SDRAM devices and so are different from those found in the *Tsi107 User Manual*.

1.4.1 Memory Page Mode Register

The 1-byte memory page mode register, shown in Figure 5 and Table 4, contains the PGMAX parameter which controls how long the Tsi107 retains the currently accessed page (row) in memory.

Figure 5: Memory Page Mode Register—0xA3

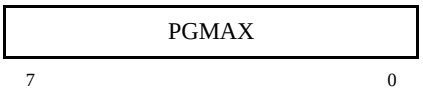


Table 4: Bit Settings for Memory Page Mode Register—0xA3

Bits	Name	Reset Value	Description
7–0	PGMAX	All 0s	For SDRAM configurations, the value of PGMAX multiplied by 64 determines the activate to precharge interval (sometimes called row active time or t_{RAS}) for retained page mode. When programmed to 0x00, page mode is disabled.

The 1-byte memory page mode register (MPMR) contains the PGMAX parameter that controls how long the Tsi107 retains the currently accessed page (row) in memory. The PGMAX parameter specifies the activate-to-precharge interval (sometimes called row active time or tRAS). The PGMAX value is multiplied by 64 to generate the actual number of clock cycles for the interval. When PGMAX is programmed to 0x00, page mode is disabled.The value for PGMAX depends on the specific SDRAM devices used, the ROM system, and the operating frequency of the Tsi107. When the interval specified by PGMAX expires, the Tsi107 must close the active page by issuing a precharge bank command. PGMAX must be sufficiently less than the maximum row active time for the SDRAM device to ensure that the issuing of a precharge command is not stalled by a memory access. When PGMAX expires during a memory access, the Tsi107 must wait for the access to complete before issuing the precharge command to the SDRAM. In the worst case, the Tsi107 initiates a memory access one clock cycle before PGMAX expires. If ROM is located on the memory bus, the longest access that could potentially stall a precharge is a burst read from ROM. If ROM is located on the PCI bus, the longest memory access is a burst read from the SDRAM.

The Tsi107 also requires two clock cycles to issue a precharge bank command to the SDRAM device so the PGMAX interval must be further reduced by two clock cycles. Therefore, PGMAX should be programmed according to the following equation:

$PGMAX < [t_{RAS}(MAX) – (\text{worst case memory access}) – 2] / 64$

For example, consider a system with a memory bus clock frequency of 66 MHz using SDRAMs with a maximum row active time ($t_{RAS}(MAX)$) of 100 us. The maximum number of clock cycles between activate bank and precharge bank commands is 66 MHz x 100 us = 6600 clock cycles.

If the system uses 8-bit ROMs on the memory bus, a processor burst read (a 32-byte cache line read) from ROM (a non-bursting ROM device) follows the timing shown in Figure 6-60. Also affecting the ROM access time is MCCR2[TS_WAIT_TIMER]. The minimum time allowed for ROM devices to enter high impedance is two clock cycles. TS_WAIT_TIMER adds clocks (n–1) to the minimum disable time. This delay is enforced after all ROM accesses preventing any other memory access from starting. Therefore a burst read from an 8-bit ROM (worst case access time) takes:

{[(ROMFAL + 2) x 8 + 3] x 4} + [2 + (TS_WAIT_TIMER – 1)] clock cycles

So, if MCCR1[ROMFAL] = 4 and MCCR2[TS_WAIT_TIMER] = 3, the interval for a local processor burst read from an 8-bit ROM takes

{[(4 + 2) x 8 + 3] x 4} + [2 + (3 – 1)] = 204 + 4 = 208 clock cycles.

Plugging the values into the PGMAX equation above,

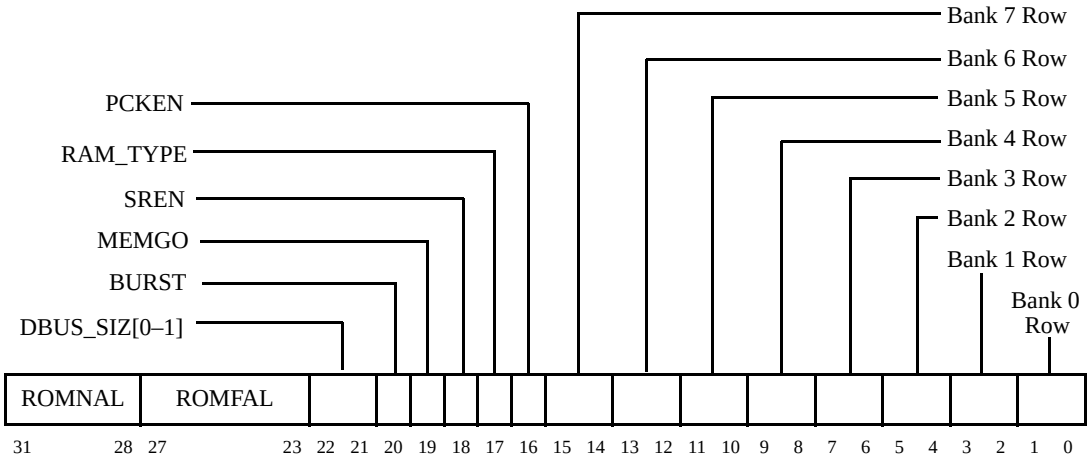
PGMAX < (6600 – 208 – 2) ÷ 64 = 99.8 clock cycles.

The value stored in PGMAX would be 0b0110_0011 (or 99 clock cycles).

1.4.2 Memory Control Configuration Register 1

Figure 6 and Table 5 show the memory control configuration register 1 (MCCR1) format and bit settings.

Figure 6: Memory Control Configuration Register 1 (MCCR1)—0xF0



SDRAM related bit fields in MCCR1 that require attention:

bit 18 SREN - If the system will support power down modes, this bit selects who becomes responsible for refreshing the SDRAM during sleep.

bit 17 RAMTYPE - this bit needs to be cleared during initialization to select SDRAM.

bit 16 PCKEN - this bit needs to be cleared for ECC operation.

bits 15 to 0 Bank n row - these need to be configured for the row/column/bank architectures of the SDRAM devices being used in the system.

Table 5: Bit Settings for MCCR1—0xF0

Bits	Name	Reset Value	Description
31–28	ROMNAL	All 1s	For burst-mode ROM and Flash reads, ROMNAL controls the next access time. The maximum value is 0b1111 (15). The actual cycle count is three cycles more than the binary value of ROMNAL. For Flash writes, ROMNAL measures the write pulse recovery (high) time. The maximum value is 0b1111 (15). The actual cycle count is four cycles more than the binary value of ROMNAL.
27–23	ROMFAL	All 1s	For nonburst ROM and Flash reads, ROMFAL controls the access time. For burst-mode ROMs, ROMFAL controls the first access time. The maximum value is 0b11111 (31). For the 64-bit and 32-bit configurations, the actual cycle count is three cycles more than the binary value of ROMFAL. For the 8-bit configuration, the actual cycle count is two cycles more than the binary value of ROMFAL. For Flash writes, ROMFAL measures the write pulse low time. The maximum value is 0b11111 (31). The actual cycle count is two cycles more than the binary value of ROMFAL.

Table 5: Bit Settings for MCCR1—0xF0 *(Continued)*

Bits	Name	Reset Value	Description
22–21	DBUS_SIZ[0–1]	xx	Read-only. This field indicates the state of the memory data path width. The value of this field is determined by the reset configuration signals [DL[0], $\overline{\text{FOE}}$]. Used with DBUS_SIZ2 (stored in MCCR4[17]) as shown below. DBUS_SIZ[0–2]: For ROM/Flash chip select #0 ($\overline{\text{RCSO}}$): 00n 32-bit data bus n1n 8-bit data bus 10n 64-bit data bus For ROM/Flash chip select #1 ($\overline{\text{RCS1}}$): 0n0 32-bit data bus nn1 8-bit data bus 1n0 64-bit data bus For ROM/Flash chip select #2 ($\overline{\text{RCS2}}$), ROM/Flash chip select #3 ($\overline{\text{RCS3}}$), and (S)DRAM: 0nn 32-bit data bus 1nn 64-bit data bus
20	BURST	0	Burst mode ROM timing enable 0 Indicates standard (nonburst) ROM access timing 1 Indicates burst-mode ROM access timing
19	MEMGO	0	RAM interface logic enable. Note that this bit must not be set until all other memory configuration parameters have been appropriately configured by boot code. 0 Tsi107 RAM interface logic disabled 1 Tsi107 RAM interface logic enabled
18	SREN	0	Self-refresh enable. Note that if self refresh is disabled, the system is responsible for preserving the integrity of DRAM/EDO/SDRAM during sleep mode. 0 Disables the DRAM/EDO/SDRAM self refresh during sleep mode 1 Enables the DRAM/EDO/SDRAM self refresh during sleep mode
17	RAM_TYPE	1	RAM type 0 Indicates synchronous DRAM (SDRAM) 1 Indicates DRAM or EDO DRAM (depending on the setting for MCCR2[EDO]) Note that this bit must be cleared (selecting DRAM or SDRAM) before the in-line or registered buffer mode bits in MCCR4 are set.

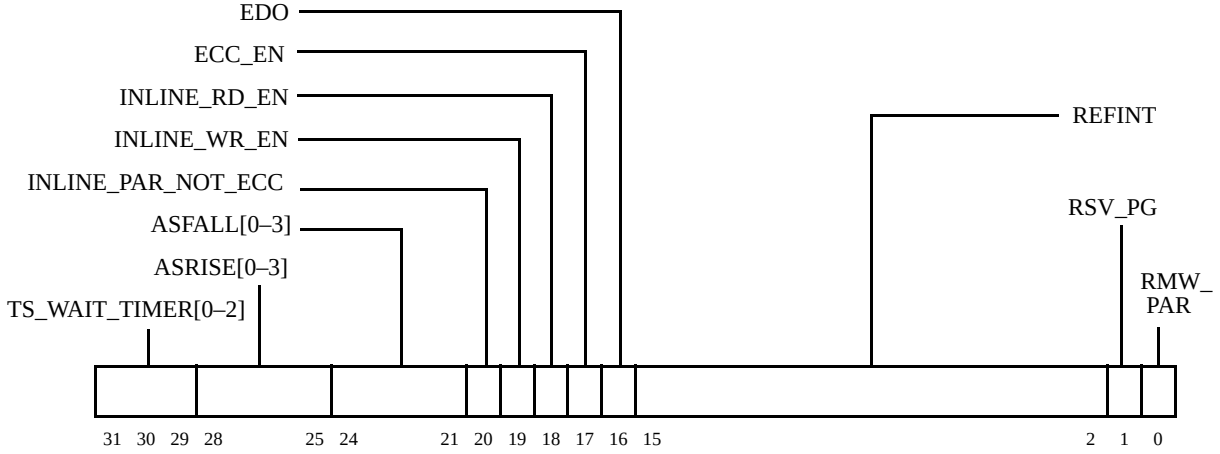
Table 5: Bit Settings for MCCR1—0xF0 *(Continued)*

Bits	Name	Reset Value	Description
16	PCKEN	0	Memory interface parity checking/generation enable 0 Disables parity checking and parity generation for transactions to DRAM/EDO/SDRAM memory. Note that this bit must be cleared for SDRAM memory when operating in in-line buffer mode (MCCR4[BUF_TYPE[0–1]] = 0b10) and in-line parity/ECC is enabled with MCCR2[INLINE_RD_EN] = 1. 1 Enables parity checking and generation for all registered mode memory transactions to DRAM/EDO/SDRAM memory.
15–14	Bank 7 row	00	RAM bank 7 row address bit count. These bits indicate the number of row address bits that are required by the RAM devices in bank 7. For FPM/EDO DRAM configurations (RAM_TYPE = 1), the encoding is as follows: 00 9 row bits 01 10 row bits 10 11 row bits 11 12 or 13 row bits For SDRAM configurations (RAM_TYPE = 0), the encoding is as follows: 00 12 row bits by n column bits by 4 logical banks (12 × n × 4) or 11 row bits by n column bits by 4 logical banks (11 × n × 4) 01 13 row bits by n column bits by 2 logical banks (13 × n × 2) or 12 row bits by n column bits by 2 logical banks (12 × n × 2) 10 13 row bits by n column bits by 4 logical banks (13 × n × 4) 11 11 row bits by n column bits by 2 logical banks (11 × n × 2)
13–12	Bank 6 row	00	RAM bank 6 row address bit count. See the description for Bank 7 row (bits 15–14).
11–10	Bank 5 row	00	RAM bank 5 row address bit count. See the description for Bank 7 row (bits 15–14).
9–8	Bank 4 row	00	RAM bank 4 row address bit count. See the description for Bank 7 row (bits 15–14).
7–6	Bank 3 row	00	RAM bank 3 row address bit count. See the description for Bank 7 row (bits 15–14).
5–4	Bank 2 row	00	RAM bank 2 row address bit count. See the description for Bank 7 row (bits 15–14).
3–2	Bank 1 row	00	RAM bank 1 row address bit count.See the description for Bank 7 row (bits 15–14).
1–0	Bank 0 row	00	RAM bank 0 row address bit count. See the description for Bank 7 row (bits 15–14).

1.4.3 Memory Control Configuration Register 2—0xF4

Figure 7 and Table 6 show the memory control configuration register 2 (MCCR2) format and bit settings.

Figure 7: Memory Control Configuration Register 2 (MCCR2)—0xF4



SDRAM related bit fields in MCCR2 that require attention:

bit 20 Inline_par_not_ECC - This bit needs to be cleared for ECC operation.

bit 19 Inline_WR_EN - This bit needs to be cleared for ECC operation.

bit 18 Inline_RD_EN - This bit needs to be set for ECC operation.

bit 17 ECC_EN - This bit needs to be cleared for ECC operation with SDRAM.

bit 16 EDO - This bit needs to be cleared for ECC operation.

bit 15-2 REFINT - The memory interface supplies CAS before RAS (CBR) refreshes to SDRAM at the refresh interval specified by MCCR2[REFINT]. When REFINT expires, the Tsi107 issues a precharge and then a refresh command to the SDRAM devices. The value stored in REFINT should allow for a potential collision between memory accesses and refresh cycles. In the worst case, the refresh must wait the number of clock cycles required by the longest access. For example, if ROM is located on the 60x/memory bus and a ROM access is in progress at the time a refresh operation needs to be performed, the refresh must wait until the ROM access has completed. If ROM is located on the 60x/memory bus, the longest access that could potentially stall a refresh is a burst read from ROM. If ROM is located on the PCI bus, the longest memory access is a burst read from the SDRAM. The Tsi107 also has to wait for a precharge command (to close any open pages) before it can issue the refresh command. the Tsi107 requires two clock cycles to issue a precharge to an internal bank; with two pages open simultaneously, this equates to four extra clock cycles that must be taken off the refresh interval. Finally, the Tsi107 must wait for the PRETOACT interval to pass before issuing the refresh command.

Therefore, REFINT should be programmed according to the following equation:

REFINT < (per row refresh interval) – (worst case memory access) – (PRETOACT) – 4

Consider a typical SDRAM device with a refresh period of 32 ms for a 2K cycle. This means that it takes 32 ms to refresh each internal bank and each internal bank has 2K rows. To refresh the whole SDRAM (two internal banks, 4K rows) it takes 64 ms. The refresh time per row is 32 ms ÷ 2048 rows (or 64 ms ÷ 4096 rows) = 15.6 µs. If the 60x bus clock is running at 66 MHz, the number of clock cycles per row refresh is 15.6 µs x 66 MHz = 1030 clock cycles.

If the system uses 8-bit ROMs on the 60x/memory bus, a burst read from ROM will follow the timing shown in Figure 8 below. Also affecting the ROM access time is MCCR2[TS_WAIT_TIMER]. The minimum time allowed for ROM devices to enter high impedance is two clock cycles. TS_WAIT_TIMER adds clocks (n–1) to the minimum disable time. This delay is enforced after all ROM accesses preventing any other memory access from starting. Therefore, a burst read from an 8-bit ROM will take:

{[(ROMFAL + 2) x 8 + 3] x 4 + 5} + [2 + (TS_WAIT_TIMER – 1)] clock cycles

So, if MCCR1[ROMFAL] = 4 and MCCR2[TS_WAIT_TIMER] = 3, the interval for a 60x burst read from an 8-bit ROM will take:

{[(4 + 2) x 8 + 3] x 4 + 5} + [2 + (3 – 1)] = 209 + 4 = 213 clock cycles

Plugging the values into the REFINT equation above:

REFINT < 1030 – 213 – 2 – 4 = 811 clock cycles

The value stored in REFINT should be 0b00 0011 0010 1010 (or 810 clock cycles).

bit 0 RMW_PAR - This bit needs to be set for ECC operation

Figure 8: 8-Bit ROM/Flash Interface—Burst Read Timing.

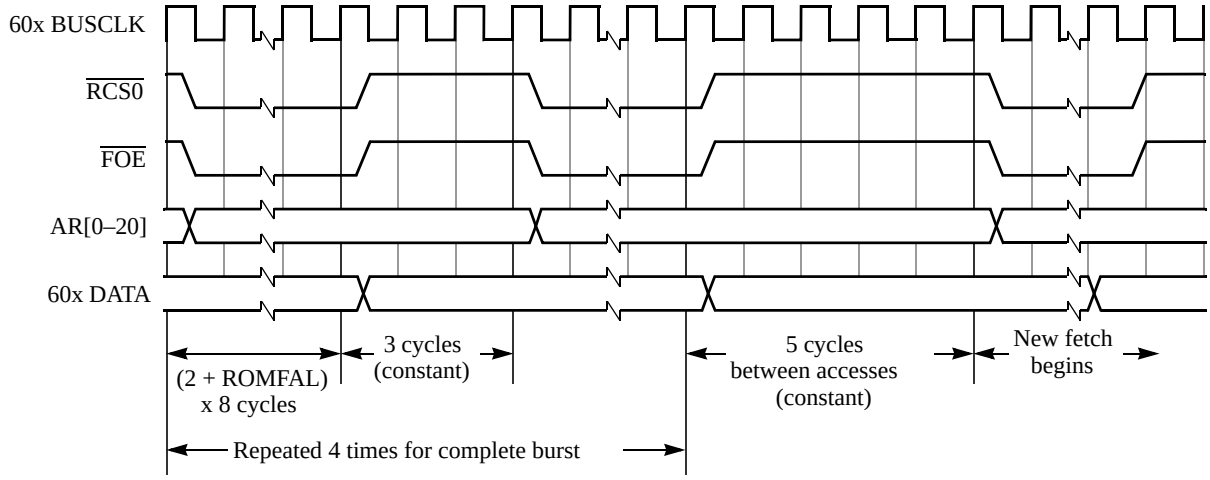


Table 6: Bit Settings for MCCR2—0xF4

<table>
<tr><th>Bits</th><th>Name</th><th>Reset Value</th><th>Description</th></tr>
<tr><td>31–29</td><td>TS_WAIT_ TIMER[0–2]</td><td>000</td><td>Transaction start wait states timer. The minimum time allowed for ROM/Flash/Port X devices to enter high impedance is 2 memory system clocks. TS_WAIT_TIMER[0–2] adds wait states before the subsequent transaction starts in order to account for longer disable times of a ROM/Flash/Port X device. This delay is enforced after all ROM and Flash accesses, delaying the next memory access from starting (for example, DRAM after ROM access, SDRAM after Flash access, ROM after Flash access).
Note that this parameter is supported for SDRAM systems only. For EDO/FPM DRAM systems, TS_WAIT_TIMER[0–2] must = 000.
<table>
<tr><th rowspan="2">Bits</th><th colspan="3">Wait States for ROM High Impedance</th></tr>
<tr><th>Reads with wide data path (32 or 64-bit)</th><th>Reads with gather data path in flow-through or registered buffer mode (8, 16, 32-bit)</th><th>All writes[1, 2] and reads with gather data path in in-line buffer mode (8, 16, 32,-bit)</th></tr>
<tr><td>000</td><td>2 clocks</td><td>5 clocks</td><td>6 clocks</td></tr>
<tr><td>001</td><td>2 clocks</td><td>5 clocks</td><td>6 clocks</td></tr>
<tr><td>010</td><td>3 clocks</td><td>5 clocks</td><td>6 clocks</td></tr>
<tr><td>011</td><td>4 clocks</td><td>5 clocks</td><td>6 clocks</td></tr>
<tr><td>100</td><td>5 clocks</td><td>6 clocks</td><td>7 clocks</td></tr>
<tr><td>101</td><td>6 clocks</td><td>7 clocks</td><td>8 clocks</td></tr>
<tr><td>110</td><td>7 clocks</td><td>7 clocks</td><td>7 clocks</td></tr>
<tr><td>111</td><td>8 clocks</td><td>9 clocks</td><td>10 clocks</td></tr>
</table>
Note 1. In this context, Flash writes are defined as any write to $\overline{\text{RCS}}$[0-3].
Note 2: For Flash writes, add the write recovery time, ROMNAL, to the given wait states for ROM high-impedance time.</td></tr>
<tr><td>28–25</td><td>ASRISE[0–3]</td><td>0000</td><td>$\overline{\text{AS}}$ rise time. These bits control the rising edge timing of the $\overline{\text{AS}}$ signal for the Port X interface. For more information, see “Port X Interface” in the Tsi107 User Manual.
0000 Disables $\overline{\text{AS}}$ signal generation
0001 1 clock
0010 2 clocks
0011 3 clocks
...
1111 15 clocks</td></tr>
</table>

Table 6: Bit Settings for MCCR2—0xF4 *(Continued)*

Bits	Name	Reset Value	Description
24–21	ASFALL[0–3]	0000	$\overline{AS}$ fall time. These bits control the falling edge timing of the $\overline{AS}$ signal for the Port X interface. For more information, see “Port X Interface” in the *Tsi107 User Manual*. 0000 0 clocks ($\overline{AS}$ asserted coincident with the chip select) 0001 1 clock 0010 2 clocks 0011 3 clocks ... 1111 15 clocks
20	INLINE_PAR_ NOT_ECC	0	In-line parity —not ECC. This bit selects between the ECC and parity checking/correction mechanisms of the in-line data path when performing memory reads. This bit is applicable for SDRAM systems running in in-line buffer mode (MCCR4[BUF_TYPE[0–1]] = 0b10) only, and when INLINE_RD_EN = 1. 0 Tsi107 uses ECC on the memory data bus. 1 Tsi107 uses parity on the memory data bus.
19	INLINE_WR_EN	0	In-line parity error reporting enable. For SDRAM only. This bit controls whether the Tsi107 uses the in-line parity hardware to report 60x bus parity errors on writes to memory. Note that the buffer type selector in MCCR4 must be set to in-line buffer mode (MCCR4[BUF_TYPE[0–1]] = 0b10) to enable the in-line ECC/parity logic. 0 In-line 60x bus parity error reporting disabled 1 In-line 60x bus parity error reporting enabled
18	INLINE_RD_ EN	0	In-line read parity or ECC check/correction enable. This bit controls whether the Tsi107 uses the ECC/parity checking and/or correction hardware in the in-line data path to report ECC or parity errors on memory system read operations. This bit activates different parity/ECC checking/correction hardware than that controlled by ECC_EN and PCKEN. Read parity/ECC checking can be enabled for SDRAM systems running in in-line buffer mode (MCCR4[BUF_TYPE[0–1]] = 0b10) only. Also, note that the INLINE_PAR_NOT_ECC bit selects between parity or ECC on the memory data bus when this bit is set. 0 In-line memory bus read parity/ECC error reporting disabled 1 In-line memory bus read parity/ECC error reporting enabled. Note that MCCR1[PCKEN] must be cleared when this bit is set.
17	ECC_EN	0	ECC enable. This bit controls whether the Tsi107 uses ECC for transactions to system memory. ECC_EN should be set only for systems using FPM/EDO memory. Note that the ECC_EN parameter overrides the PCKEN parameter. Also note that this bit and RMW_PAR cannot both be set, and it is illegal to set this bit with EDO = 1 and REGISTERED = 1. Systems using SDRAM use a different (in-line) ECC hardware and therefore, must have ECC_EN = 0. For more information, see “SDRAM Interface Operation” in the *Tsi107 User Manual*. 0 ECC disabled 1 ECC enabled

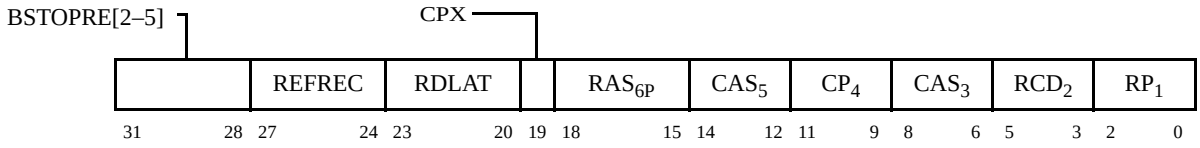
Table 6: Bit Settings for MCCR2—0xF4 *(Continued)*

Bits	Name	Reset Value	Description
16	EDO	0	EDO enable. This bit indicates the type of DRAMs for the Tsi107 memory interface. For more information, see “FPM or EDO DRAM Interface Operation” in the *Tsi107 User Manual*. 0 Indicates standard DRAMs 1 Indicates EDO DRAMs
15–2	REFINT	All 0s	Refresh interval. These bits directly represent the number of clock cycles between CBR refresh cycles. One row is refreshed in each RAM bank during each CBR refresh cycle. The value for REFINT depends on the specific RAMs used and the operating frequency of the Tsi107. For more information, see “FPM or EDO DRAM Interface Operation” in the *Tsi107 User Manual*. Note that the period of the refresh interval must be greater than the read/write access time to ensure that read/write operations complete successfully.
1	RSV_PG	0	Reserve page register. If this bit is set, the Tsi107 reserves one of the four page registers at all times. This is equivalent to only allowing three simultaneous open pages. 0 Four open page mode (default) 1 Reserve one of the four page registers at all times
0	RMW_PAR	0	Read-modify-write (RMW) parity enable. This bit controls how the Tsi107 writes parity bits to DRAM/EDO/SDRAM. Note that this bit does not enable parity checking and generation. PCKEN must be set to enable parity checking. Also note that this bit and ECC_EN cannot both be set to 1. For more information, see “FPM or EDO DRAM Parity and RMW Parity” in the *Tsi107 User Manual*. 0 RMW parity disabled 1 RMW parity enabled. Note that this bit must be set for SDRAM systems that use in-line ECC (MCCR2[ECC_EN] = 0, MCCR4[BUF_TYPE[0–1]] = 0b10, and MCCR2[INLINE_PAR_NOT_ECC]] = 0).

1.4.4 Memory Control Configuration Register 3—0xF8

Figure 9 and Table 7 show memory control configuration register 3 (MCCR3) format and bit settings.

Figure 9: Memory Control Configuration Register 3 (MCCR3)—0xF8



SDRAM related bit fields in MCCR3 that require attention:

bits 31-28 BSTOPRE[2:5] - The BSTOPRE values are determined by the customer's user code and applications depending upon how long it is necessary to keep an SDRAM page open. The BSTOPRE parameter is composed of BSTOPRE[0–1] (bits 21–20 of MCCR2), BSTOPRE[2–5] (bits 31–28 of MCCR3), and BSTOPRE[6–9] (bits 3–0 of MCCR4). BSTOPRE controls the burst-to-precharge interval. BSTOPRE is similar to PGMAX in that when it expires, the Tsi107 must generate a precharge command. However, BSTOPRE is a much shorter duration counter that gets reloaded every time a read or write command is issued to the SDRAM devices.

The page open duration counter is loaded with BSTOPRE[0:9] every time the page is accessed (including page hits). When the counter expires (or when PGMAX expires) the open page is closed with a precharge bank command. Page hits can occur at any time in the interval specified by BSTOPRE.

The BSTOPRE interval can be optimized for the particular Tsi107-based system implementation. If memory accesses are typically to the same rows within an active page, then a longer BSTOPRE interval would improve performance. Alternately, if memory accesses are typically to several locations, spanning multiple pages, then a shorter duration for BSTOPRE is in order. This allows for a precharge to close the active page before a subsequent access activates another page. Page mode is disabled by clearing the PGMAX or BSTOPRE[0:9] parameters.

Page mode can dramatically reduce access latencies for page hits. Depending on the memory system design and timing parameters, using page mode can save clock cycles from subsequent burst accesses that hit in an active page.

bits 27-24 REFREC - This field is the refresh recovery interval. This parameter is the number of clock cycles from a refresh command until a bank activate command is allowed. The interval between a refresh and a new command is TRC (row cycle time). A typical SDRAM data sheet shows $T_{RC(MIN)}$ = 96 ns. To put this time in clock cycles, for a 66-MHz bus, 96 ns x 66 MHz = 6.3 clock cycles. Rounding up to 7, the value stored to REFREC should be 0b0111

bits 23-20 RDLAT - This field is the data latency from a read command. This parameter controls the number of clock cycles from a read command until the first data beat is valid on the 60x data bus. The read command is issued with the assertion of SDCAS. Therefore, this parameter equates to the CAS latency (T_{CAC}) plus whatever latency is imposed by the data buffers. Note that data latency is programmable for read operations (RDLAT). For write operations, the first valid data beat coincides with the SDRAM write command. A typical SDRAM data sheet says that CAS latency is 30 ns. To put this time in clock cycles, for a 66-MHz bus, 30 ns x 66 MHz = 1.98 clock cycles (rounding up, 2 clock cycles). If the system uses registered data buffers, there is a one clock delay between data valid on the memory bus and data valid on the 60x bus, so the value stored to RDLAT should be 2 + 1 = 3 clock cycles (0b0011).

Table 7: Bit Settings for MCCR3—0xF8

Bits	Name	Reset Value	Description
31–28	BSTOPRE[2–5]	0000	Burst to precharge—bits 2–5. For SDRAM only. These bits, together with BSTOPRE[0–1] (bits 19–18 of MCCR4), and BSTOPRE[6–9] (bits 3–0 of MCCR4), control the open page interval. The page open duration counter is reloaded with BSTOPRE[0–9] every time the page is accessed (including page hits). When the counter expires, the open page is closed with a SDRAM-precharge bank command. For more information, see “SDRAM Page Mode” in the *Tsi107 User Manual*.
27–24	REFREC	0000	Refresh to activate interval. For SDRAM only. These bits control the number of clock cycles from an SDRAM-refresh command until an SDRAM-activate command is allowed. For more information, see “SDRAM Refresh” in the *Tsi107 User Manual*. 0001 1 clock 0010 2 clocks 0011 3 clocks ... ... 1111 15 clocks 0000 16 clocks

Table 7: Bit Settings for MCCR3—0xF8 *(Continued)*

Bits	Name	Reset Value	Description
23–20	RDLAT	0000	Data latency from read command. For SDRAM only. These bits control the number of clock cycles from an SDRAM-read command until the first data beat is available on the data bus. RDLAT values greater than 6 clocks are not supported. For more information, see “SDRAM Power-on Initialization” in the *Tsi107 User Manual*. Note that for SDRAM, this value must be programmed to a valid value (from the reset value). 0000 Reserved 0001 1 clock 0010 2 clocks 0011 3 clocks 0100 4 clocks 0101 5 clocks 0110 6 clocks 0111 Reserved (not supported) ... ... 1111 Reserved (not supported)
19	CPX	0	$\overline{CAS}$ write timing modifier. For DRAM/EDO only. When set, this bit adds one clock cycle to the $\overline{CAS}$ precharge interval (CP_4 + 1) and subtracts one clock cycle from the $\overline{CAS}$ assertion interval for page mode access (CAS_5 - 1) for write operations to DRAM/EDO. Note that this requires $CAS_5 \geq 2$. Read operations are unmodified. For more information, see “FPM or EDO DRAM Interface Timing” in the *Tsi107 User Manual*. 0 $\overline{CAS}$ write timing is unmodified 1 $\overline{CAS}$ write timing is modified as described above
18–15	RAS_{6P}	0000	$\overline{RAS}$ assertion interval for CBR refresh. For DRAM/EDO only. These bits control the number of clock cycles $\overline{RAS}$ is held asserted during CBR refresh. The value for RAS_{6P} depends on the specific DRAMs used and the frequency of the memory interface. For more information, see “FPM or EDO DRAM Refresh” in the *Tsi107 User Manual*. 0001 1 clock 0010 2 clocks 0011 3 clocks ... ... 1111 15 clocks 0000 16 clocks

Table 7: Bit Settings for MCCR3—0xF8 *(Continued)*

Bits	Name	Reset Value	Description
14–12	CAS_5	000	$\overline{CAS}$ assertion interval for page mode access. For DRAM/EDO only. These bits control the number of clock cycles $\overline{CAS}$ is held asserted during page mode accesses. The value for CAS_5 depends on the specific DRAMs used and the frequency of the memory interface. Note that when ECC is enabled, CAS_5 + CP_4 must equal four clock cycles. For more information, see “FPM or EDO DRAM Interface Timing” in the *Tsi107 User Manual*. 001 1 clock 010 2 clocks 011 3 clocks ... ... 111 7 clocks 000 8 clocks
11–9	CP_4	000	$\overline{CAS}$ precharge interval. For DRAM/EDO only. These bits control the number of clock cycles that $\overline{CAS}$ must be held negated in page mode (to allow for column precharge) before the next assertion of $\overline{CAS}$. Note that when ECC is enabled, CAS_5 + CP_4 must equal four clock cycles. For more information, see “FPM or EDO DRAM Interface Timing” in the *Tsi107 User Manual*. 001 1 clock 010 2 clocks 011 reserved ... ... 111 Reserved 000 Reserved
8–6	CAS_3	000	$\overline{CAS}$ assertion interval for the first access. For DRAM/EDO only. These bits control the number of clock cycles $\overline{CAS}$ is held asserted during a single beat or during the first access in a burst. The value for CAS_3 depends on the specific DRAMs used and the frequency of the memory interface. For more information, see “FPM or EDO DRAM Interface Timing” in the *Tsi107 User Manual*. 001 1 clock 010 2 clocks 011 3 clocks ... ... 111 7 clocks 000 8 clocks

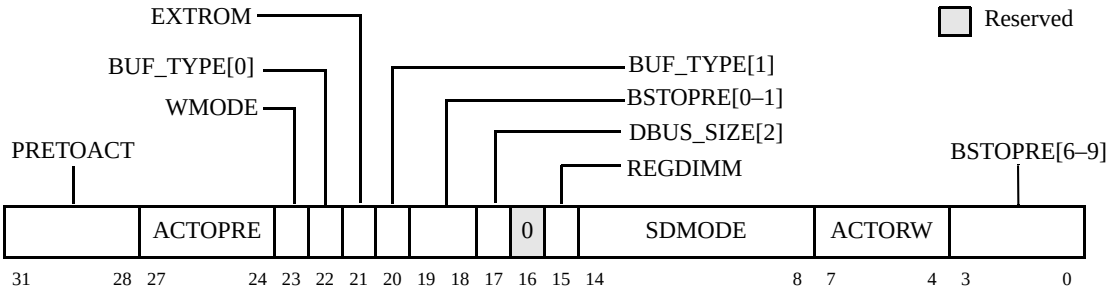
Table 7: Bit Settings for MCCR3—0xF8 *(Continued)*

Bits	Name	Reset Value	Description
5–3	RCD_2	000	$\overline{RAS}$ to $\overline{CAS}$ delay interval. For DRAM/EDO only. These bits control the number of clock cycles between the assertion of $\overline{RAS}$ and the first assertion of $\overline{CAS}$. The value for RCD_2 depends on the specific DRAMs used and the frequency of the memory interface. However, RCD_2 must be at least two clock cycles. For more information, see "FPM or EDO DRAM Interface Timing" in the *Tsi107 User Manual*. 001 Reserved 010 2 clocks 011 3 clocks ... ... 111 7 clocks 000 8 clocks
2–0	RP_1	000	$\overline{RAS}$ precharge interval. For DRAM/EDO only. These bits control the number of clock cycles that $\overline{RAS}$ must be held negated (to allow for row precharge) before the next assertion of $\overline{RAS}$. Note that RP_1 must be at least two clock cycles and no greater than 5 clock cycles. For more information, see "FPM or EDO DRAM Interface Timing" in the *Tsi107 User Manual*. 010 2 clocks 011 3 clocks 110 4 clocks 101 5 clocks All others: reserved

1.4.5 Memory Control Configuration Register 4—0xFC

Figure 10 and Table 8 show memory control configuration register 4 (MCCR4) format and bit settings.

Figure 10: Memory Control Configuration Register 4 (MCCR4)—0xFC



SDRAM related bit fields in MCCR4 that require attention:

bits 31-28 PRETOACT - This field is the number of clock cycles from a precharge command until a bank activate command is allowed. A typical SDRAM data sheet shows that T_{RP} is the minimum row precharge time. That is, the time after a precharge command is issued before the bank activate command can begin. T_{RP} for a typical SDRAM device is 30 ns. To put this time in clock cycles, for a 66-MHz bus, 30 ns x 66 MHz = 1.98 clock cycles. Rounding up to 2, the value stored to PRETOACT should be 0b0010.

bits 27-24 ACTOPRE - This field is the number of clock cycles from a bank activate command until a precharge command is allowed. In a typical SDRAM data sheet, $T_{RAS(MIN)}$ is the specification for minimum row active time. Every bank activate command must satisfy the $T_{RAS(MIN)}$ specification before a precharge command can be issued to that active bank. For a typical SDRAM device, $T_{RAS(MIN)}$ is 70 ns. To put this time in clock cycles, for a 66-MHz bus, 70 ns x 66 MHz = 4.62 clock cycles. Rounding up to 5, the value stored to ACTOPRE should be 0b0101.

bits 22 and 20 BUF_TYPE[0-1] - These bits must be set to 0b10 for SDRAM ECC mode.

bits 19-18 and 3-0 BSTOPRE[0-1] and BSTOPRE[6-9] - See also **BSTOPRE[2-5]** "bits 31-28 BSTOPRE[2:5] - The BSTOPRE values are determined by the customer's user code and applications depending upon how long it is necessary to keep an SDRAM page open. The BSTOPRE parameter is composed of BSTOPRE[0–1] (bits 21–20 of MCCR2), BSTOPRE[2–5] (bits 31–28 of MCCR3), and BSTOPRE[6–9] (bits 3–0 of MCCR4). BSTOPRE controls the burst-to-precharge interval. BSTOPRE is similar to PGMAX in that when it expires, the Tsi107 must generate a precharge command. However, BSTOPRE is a much shorter duration counter that gets reloaded every time a read or write command is issued to the SDRAM devices." on page 27.

bit 15 REGDIMM - This bit must be set depending on the designer's selection of SDRAM type. The choices are Registered (set to 1) or normal (set to 0).

bits 14-8 SDMODE - This field specifies the SDRAM mode register data to be written to the SDRAM array during power-up configuration. The Tsi107 has certain restrictions for the SDMODE bits. In fact, the only variable in the SDMODE fields is the CAS latency parameter.

The opcode should be 0b0 0000 for normal operation.

In a typical SDRAM data sheet, T_{CAC} is the specification for CAS latency. T_{CAC} is programmable on most SDRAM devices to one, two, or three clock cycles depending on operating frequency of the device. A typical SDRAM data sheet says that CAS latency is 30 ns. To put this time in clock cycles, for a 66-MHz bus, 30 ns x 66 MHz = 1.98 clock cycles. Rounding up to 2, the value for CAS latency should be 0b0010.

Finally, the wrap type must be 0b0 for sequential type wrapping and the wrap length must be 0b010 for four (doubleword) beats for each access.

Therefore, the complete value stored to SDMODE should be 0b0000 0010 0010.

Note that for 64-Mbit SDRAMs, the SDRAM mode register data is actually a 14-bit field. On the Tsi107, the two-most-significant bits are forced to 0 and appended to MCCR4[SDMODE].

bits 7-4 ACTORW - This field controls the number of clocks cycles from a bank activate command until a read or write command is allowed. It must be at least two clock cycles.

In a typical SDRAM data sheet, T_{RCD} is the specification for the RAS-to-CAS delay time. For SDRAM command encoding, RAS corresponds to the bank activate command and CAS corresponds to the read or write command.

For a typical SDRAM device, T_{RCD} = 25 ns. To put this time in clock cycles, for a 66-MHz bus, 25 ns x 66 MHz = 1.65 clock cycles. Rounding up to 2, the value for ACTORW should be 0b0010.

Table 8: Bit Settings for MCCR4—0xFC

Bits	Name	Reset Value	Description
31–28	PRETOACT	0000	Precharge to activate interval. For SDRAM only. These bits control the number of clock cycles from an SDRAM-precharge command until an SDRAM-activate command is allowed. For more information, see “SDRAM Power-on Initialization” in the *Tsi107 User Manual*. 0001 1 clock 0010 2 clocks 0011 3 clocks ... ... 1111 15 clocks 0000 16 clocks
27–24	ACTOPRE	0000	Activate to precharge interval. For SDRAM only. These bits control the number of clock cycles from an SDRAM-activate command until an SDRAM-precharge command is allowed. For more information, see “SDRAM Power-on Initialization” in the *Tsi107 User Manual*. 0001 1 clock 0010 2 clocks 0011 3 clocks ... ... 1111 15 clocks 0000 16 clocks
23	WMODE	0	Length of burst for 32-bit data. Applies to 32-bit data path mode only. Determines whether the burst ROMs can accept eight beats in a burst or only four. In 32-bit data path mode, burst transactions require data beats. If the burst ROM can only accept four beats per burst, the memory controller must perform two transactions to the ROM. 0 Four beats per burst (default) 1 Eight beats per burst
22	BUF_TYPE[0]	0	Most significant bit of the memory data bus buffer type field. BUF_TYPE[0] is used with bit 20 below (BUF_TYPE[1]) to configure the internal memory data path buffering scheme as follows: BUF_TYPE[0–1]: 00 Reserved 01 Registered buffer mode (default) 10 In-line buffer mode; SDRAM only 11 Reserved The Tsi107 must be configured for in-line buffer mode in order to use the in-line ECC/parity logic for SDRAM. The in-line ECC and parity hardware allow the Tsi107 to check/generate parity on the 60x bus and check/correct/generate ECC or parity on the external SDRAM memory bus. For more information, see “SDRAM Memory Data Interface” and “FPM or EDO Memory Data Interface” in the *Tsi107 User Manual*.
21	EXTROM	0	Extended ROM space enable 0 Extended ROM disabled 1 Extended 128 Mbytes of local ROM memory space enabled

Table 8: Bit Settings for MCCR4—0xFC *(Continued)*

Bits	Name	Reset Value	Description
20	BUF_TYPE[1]	1	Least significant bit of the memory data bus buffer type field. BUF_TYPE[1] is used with bit 22 above (BUF_TYPE[0]) to configure the internal memory data path buffering scheme as described for bit 22.
19–18	BSTOPRE[0–1]	00	Burst to precharge—bits 0–1. For SDRAM only. These bits, together with BSTOPRE[2–5] (bits 31–28 of MCCR3), and BSTOPRE[6–9] (bits 3–0 of MCCR4), control the open page interval. The page open duration counter is reloaded with BSTOPRE[0–9] every time the page is accessed (including page hits). When the counter expires, the open page is closed with a SDRAM-precharge bank command. For more information, see “Memory Interface” in the *Tsi107 User Manual*.
17	DBUS_SIZE[2]	0	See description for bits 22–21 of MCCR1.
16	—	0	Reserved
15	REGDIMM	0	Registered DIMMs. Memory data and parity data path buses configured for registered DIMMs. For SDRAM only. When enabled (REGDIMM = 1), SDRAM write data and parity are delayed by one cycle on the memory bus with respect to the SDRAM control signals (for example, $\overline{\text{SDRAS}}$, $\overline{\text{SDCAS}}$, $\overline{\text{WE}}$). 0 Normal DIMMs 1 Registered DIMMs selected

Table 8: Bit Settings for MCCR4—0xFC *(Continued)*

Bits	Name	Reset Value	Description
14–8	SDMODE	All 0s	SDRAM mode register. For SDRAM only. These bits specify the SDRAM mode register data to be written to the SDRAM array during power-up configuration. Note that the SDRAM mode register 'opcode' field is not specified and is forced to b'0_0000' by the Tsi107 when the mode registers are written. Bits 14–12 CAS latency 000 Reserved 001 1 010 2 011 3 100 Reserved 101 Reserved 110 Reserved 111 Reserved Bit 11 Wrap type 0 Sequential. Default for Tsi107 1 Interleaved - Reserved Bits 10–8 Burst length 000 Reserved 001 Reserved 010 4 011 8 100 Reserved 101 Reserved 110 Reserved 111 Reserved
7–4	ACTORW	0000	Activate to read/write interval. For SDRAM only. These bits control the number of clock cycles from an SDRAM-activate command until an SDRAM-read or SDRAM-write command is allowed. For more information, see "SDRAM Power-on Initialization" in the *Tsi107 User Manual*. 0001 Reserved 0010 2 clocks (minimum for flow-through or registered data interfaces) 0011 3 clocks (minimum for in-line ECC/parity data interfaces) ... ... 1111 15 clocks 0000 16 clocks

Table 8: Bit Settings for MCCR4—0xFC *(Continued)*

Bits	Name	Reset Value	Description
3–0	BSTOPRE[6–9]	0000	Burst to precharge—bits 6–9. For SDRAM only. These bits, together with BSTOPRE[0–1] (bits 19–18 of MCCR4), and BSTOPRE[2–5] (bits 31–28 of MCCR3), control the open page interval. The page open duration counter is reloaded with BSTOPRE[0–9] every time the page is accessed (including page hits). When the counter expires, the open page is closed with a SDRAM-precharge bank command. For more information, see “Memory Interface” in the *Tsi107 User Manual*.

2. SDRAM In-Line ECC

As an alternative to simple parity, the Tsi107 supports ECC for the data path between the Tsi107 and system memory. ECC not only allows the Tsi107 to detect errors in the memory data path but also to correct single-bit errors in the 64-bit data path. Note that ECC is not supported for systems using a 32-bit memory data bus. Since the ECC syndrome bits are calculated on a 64-bit basis, the Tsi107 performs a read-modify-write cycle to the memory in order to complete sub-double word write operations.

The Tsi107 supports concurrent ECC for the memory data path, and parity for the local processor data path. ECC and parity may be independently enabled or disabled.

The in-line ECC data path option allows the Tsi107 to detect and automatically correct single bit ECC errors; and detect multiple bit ECC errors with only a one clock cycle penalty on the CPU and PCI memory read operations; and generate parity for the processor data bus. For CPU and PCI memory write operations, parity can be checked automatically on the processor data bus and ECC generated for the memory bus.

Other memory access errors are not guaranteed to be detected or corrected. Multiple-bit errors are always reported if detected. When a single-bit error occurs, the value in the ECC single bit error counter register is incremented and compared to the ECC single bit error trigger register. If the values are not equal, no error is reported, but the error is corrected. If the values are equal, then the error is corrected and is reported. Thus, the single-bit error registers may be programmed so that minor faults with memory are corrected and ignored, but a catastrophic memory failure generates an interrupt.

The Memory Data Path Error Capture Monitor Registers (DH at offsets 0xF_F00C and 0xF0C; DL at offsets 0xF_F010 and 0xF10; and Parity at offsets 0xF_F014 and 0xF14), are used to latch the data that contains ECC or parity errors from either the 60x bus or the memory data/parity bus. These registers load automatically when the error detectors are set. The individual error detection bits are contained in the PCI Status Register (offset 0x06) bits 12, 13 and 15, the Error Detection Register 1 (offset 0xC1) bits 7-4 and 2-0, and the Error Detection Register 2 (offset 0xC5) bits 5-3 and 0. These bits indicate which type of error has been detected. See Section 2.1, “ECC Single-Bit Error Registers,” for more information on these registers.

The SDRAM ECC Syndrome Encoding table only explains how ECC works on the Tsi107 for SDRAM. The syndrome encodings are usually based on a modified Hamming code and can be found in many computer science texts and journals from the IEEE Computer Society and the ACM. The ECC uses a syndrome byte to encode bit positions of 1s within a double-word. Table 9 and Table 10 show the codes for the Tsi107 implementation of the SDRAM ECC Syndrome Encoding. The Xs mark the bits that are encoded.

When a unit of data is prepared for storage in memory, a code (composed of syndrome bits) that describes the bit sequence in the data is calculated and stored along with the unit of data. For a 64-bit double-word, eight bits are required to accurately describe the bit sequence. When a double-word is requested for reading, the memory controller reads the data and syndrome from memory. It then re-calculates the syndrome bits based on the data read from memory. The newly generated syndrome bits are then compared to the syndrome bits that were retrieved with the data. If the codes match, the data is free of errors and is sent to the processor. If the codes do not match, the controller determines if there is a single erroneous bit or if multiple erroneous bits are present. If it is a single bit error, the controller corrects the bad bit before forwarding the data to the processor. No attempt is made to correct the data that is still in memory as it is expected that it will eventually be overwritten by new data. If it is a multiple bit error, the controller cannot determine which bits are bad, so an error is flagged. The memory read cycle still completes on the 60x bus, with the double-word that was retrieved from memory.

Table 9: The Tsi107 SDRAM ECC Syndrome Encoding (Data Bits 0:31)

Syndrome Bit	Data Bit																															
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**
0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x				x				x				x				x
1	x				x				x				x				x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
2		x				x				x				x			x				x				x				x			
3			x				x				x				x			x				x				x				x		

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

Table 10: The Tsi107 SDRAM ECC Syndrome Encoding (Data Bits 32:63)

Syndrome Bit	Data Bit																															
	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
0			x				x				x				x					x				x				x		x		
1				x				x				x				x	x				x				x						x	
2	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		x				x				x						x
3	x				x				x				x						x				x				x		x	x	x	x
4		x	x	x		x	x	x		x	x	x		x	x	x												x	x	x	x	x
5					x	x	x	x					x	x	x	x	x	x	x	x	x	x	x	x								
6									x	x	x	x	x	x	x	x	x	x	x	x					x	x	x	x	x	x	x	x
7	x	x					x	x			x	x	x	x							x	x	x	x	x	x	x	x		x	x	x

2.1 ECC Single-Bit Error Registers

The ECC single-bit error registers are composed of two 8-bit registers called the error counter and error trigger registers and are used to control the reporting of ECC single-bit errors. See Section 2.1.3 for more information on the ECC error reporting behavior of the Tsi107.

2.1.1 ECC Single-Bit Error Counter Register—0xB8

The ECC single-bit error counter is shown in Figure 11. This register maintains a count of the number of single-bit errors that have been detected. It is a read/write register that is cleared to 0x00 at reset and whenever any data is written to it.

Figure 11: ECC Single-Bit Error Counter Register—0xB8

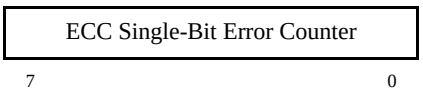


Table 11: Bit Settings for ECC Single-Bit Error Counter Register—0xB8

Bits	Name	Reset Value	Description
7–0	ECC single-bit error counter	All 0s	These bits maintain a count of the number of ECC single-bit errors that have been detected and corrected. If this value equals the value contained in the ECC single-bit error trigger register, then an error is reported (provided ErrEnR1[2] = 1).

2.1.2 ECC Single-Bit Error Trigger Register—0xB9

The ECC single-bit error trigger, shown in Figure 12, provides a threshold value that, when equal to the single-bit error count, triggers the Tsi107 error reporting logic.

Figure 12: ECC Single-Bit Error Trigger Register—0xB9

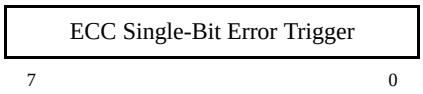


Table 12: Bit Settings for ECC Single-Bit Error Trigger Register—0xB9

Bits	Name	Reset Value	Description
7–0	ECC single-bit error trigger	All 0s	These bits provide the threshold value for the number of ECC single-bit errors that are detected before reporting an error condition. If the value of the single bit error counter register equals the value of this register, then an error is reported (provided ErrEnR1[2] = 1). If this register = 0x00, then no single bit error is ever generated.

2.1.3 Handling of Memory Interface Errors

The memory interface of the Tsi107 detects read parity, ECC, memory select, and refresh overflow errors. The parity and ECC errors are discussed above in Section 2. Since the ECC logic corrects single-bit errors, they are reported only when the number of errors in the ECC single-bit error counter register equals the threshold value in the ECC single-bit error trigger register, or if a multi-bit error occurs. A memory select error occurs when a local memory transaction address falls outside of the physical memory boundaries as programmed in the memory boundary registers. A refresh overflow error occurs when no refresh transaction occurs within the equivalent of 16 refresh cycles.

In all cases, if the memory transaction is initiated by a PCI master, ErrDR1[3] is set; if the memory transaction is initiated by the processor, ErrDR1[3] is cleared.

ErrDR2[7] is cleared to indicate that the error address in the processor/PCI error address register is valid. If the ECC single-bit error trigger threshold is reached, then the error address indicates the address of the most recent ECC single-bit error. Note that when a parity or ECC error occurs on the last beat of a transaction and another transaction to the same page has started, the Tsi107 cannot provide the error address and the corresponding bus status. In these cases, ErrDR2[7] is set to indicate that the error address in the processor/PCI error address register is not valid. The Tsi107 cannot provide the error address and the bus status for refresh overflow errors, so ErrDR2[7] is set for these errors as well.

If the transaction is initiated by the 60x processor by a PCI master with bit 6 of the PCI command register cleared, the error status information is latched, but the transaction continues and terminates normally.

When the Tsi107 is configured to perform an ECC check on every memory read cycle, it also generates the ECC check data on every memory write cycle. When a single-bit ECC error occurs, the ECC single-bit error counter register is incremented by one, and its value is compared to the value in the ECC single-bit error trigger register. If the values are equal, ErrDR1[2] is set. In addition to single-bit errors, the Tsi107 detects all 2-bit errors, all errors within a nibble (one-half byte), and any other multi-bit error that does not alias to either a single-bit error or no error. When a multi-bit ECC error occurs, ErrDR2[3] is set.

After system power-up and memory control register initialization, the memory array must be initialized with a sequence of writes to establish the correct syndrome bits for every location. The consequence to skipping this step is the immediate notification of a multi-bit error condition when reads of the memory array begin.

3. Debug Registers

The Tsi107 provides several features to aid in starting-up and debugging the system. This section describes the memory data path error injection/capture function. This capability permits the injection of single and multi-bit stuck-at faults onto the 60x or memory data and parity buses, and allow the capture of the contents of those buses upon the detection of the ECC or parity error.

3.1 Debug Register Summary

The debug registers in the Tsi107 are the six memory data path diagnostic registers consisting of three error injection mask registers and three error capture monitor registers. The debug registers are mapped as follows:

- Embedded utilities memory block (EUMBBAR) for local bus accesses at offsets 0xF_F000 to 0xF_FFFF
- Embedded utilities peripheral control and status registers (PCSRBAR) for PCI bus accesses at offsets 0xF00 to 0xFFF

Note that this data path diagnostic register space is also shared with the watchpoint registers described in Chapter 16, Programmable I/O and Watchpoint of the *Tsi107 User Manual*. The offsets to the individual registers are defined in Table 13. Note that while these registers are mapped from local bus offset 0xF_F000 to 0xF_F014 (PCI offset 0xF00 to 0xF14), the watchpoint registers are mapped from the local bus offset 0xF_F018 though 0xF_F042 (PCI offset 0xF18 to 0xF42).

Table 13: Memory Data Path Diagnostic Register Offsets

Local Bus Offset	PCI Bus Offset	Size (bytes)	Program Access Size (bytes)	Register	Register Access	Reset Value
0xF_F000	0xF00	4	4	MDP_ERR_INJ_MASK_DH	R/W	0x0000_0000
0xF_F004	0xF04	4	4	MDP_ERR_INJ_MASK_DL	R/W	0x0000_0000
0xF_F008	0xF08	4	1, 2, or 4	MDP_ERR_INJ_MASK_PAR	R/W	0x0000_0000
0xF_F00C	0xF0C	4	4	MDP_ERR_CAP_MON_DH	R	0x0000_0000
0xF_F010	0xF10	4	4	MDP_ERR_CAP_MON_DL	R	0x0000_0000
0xF_F014	0xF14	4	1, 2, or 4	MDP_ERR_CAP_MON_PAR	R/W	0x0000_0000

3.2 Memory Data Path Error Injection/Capture

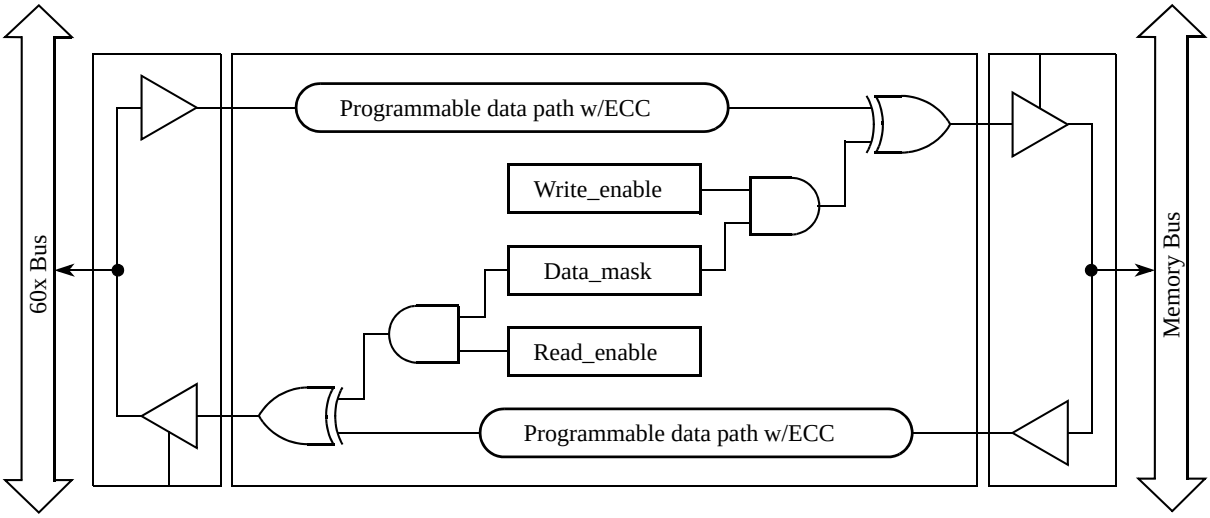
The Tsi107 provides hardware to exercise and debug the on-chip ECC and parity logic by allowing the user to inject multi-bit stuck-at faults onto the 60x or memory data/parity buses and to capture the data/parity upon the receipt of an ECC or parity error. The most common use of this type of feature is in the validation of error notification logic as part of a built-in diagnostic utility.

The memory data path error injection/capture system is programmed via the six memory data path diagnostic registers. These registers allow the user to program error injection masks and to monitor ECC/parity error capture data. The memory data path diagnostic registers are accessible from either the local bus or PCI port. All memory data path diagnostic registers are reset to zero, 0x0000_0000, unless otherwise specified.

The memory data path error injection masks are used to inject errors onto the 60x bus or the memory data/parity buses as shown in Figure 13. Separate mask registers are provided for the high data, low data, and parity buses. The masks are bit-wise inverting; a 0b1 in the mask causes the corresponding bit on the data/parity bus to be inverted. The masks are applied to the read and/or write data path by read and write mask enable bits. These registers can be read or written and are initialized to 0x0000_0000.

Note that WP_CONTROL[WP_ERR_INJ] must equal 0b10 to enable the feature of memory data path error injection/capture.

Figure 13: Functional Diagram of Memory Data Path Error Injection



3.2.1 DH Error Injection Mask Register

Figure 14 shows the bits of the DH error injection mask register. This mask affects the MDH[0:31] signals.

Figure 14: DH Error Injection Mask (MDP_ERR_INJ_MASK_DH) — Offsets 0xF_F000, 0xF00

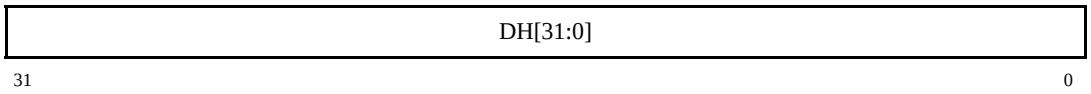


Table 14 shows the bit definitions of the DH error injection mask register.

Table 14: DH Error Injection Mask Bit Field Definitions

Bits	Name	Reset Value	R/W	Description
31–0	DH[31:0]	all 0s	R/W	Error injection mask for memory data path data bus high

3.3 DL Error Injection Mask Register

Figure 15 shows the bits of the DL error injection mask register. This mask affects the MDL[0:31] signals.

Figure 15: DL Error Injection Mask (MDP_ERR_INJ_MASK_DL)—Offsets 0xF_F004, 0xF04

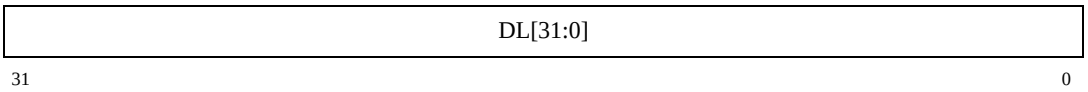


Table 15 shows the bit definitions of the DL error injection mask register

Table 15: DL Error Injection Mask Bit Field Definitions

Bits	Name	Reset Value	R/W	Description
31–0	DL[31:0]	all 0s	R/W	Error injection mask for memory data path data bus low

3.4 Parity Error Injection Mask Register

Figure 16 shows the bits of the DH error injection mask register and Table 16 shows the bit definitions.

Figure 16: Parity Error Injection Mask (MDP_ERR_INJ_MASK_PAR) — Offsets 0xF_F008, 0xF08

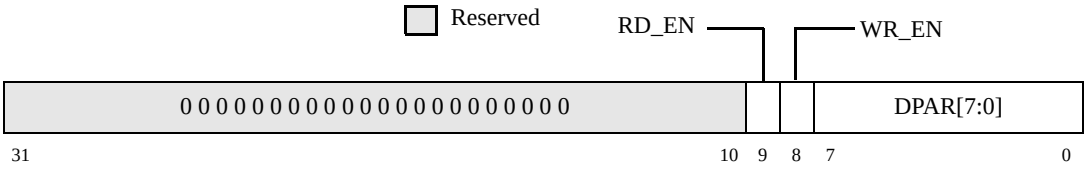


Table 16: Parity Error Injection Mask Bit Field Definitions

Bits	Name	Reset Value	R/W	Description
31–8	—	all 0s	Read	Reserved
9	RD_EN	0	R/W	Memory data path read enable bit 0 Disables error injection for reads 1 Enables error injection onto the 60x data bus during reads from local memory
8	WR_EN	0	R/W	Memory data path write enable bit 0 Disables error injection for writes 1 Enables error injection onto the memory data bus during writes to local memory
7–0	DPAR[7:0]	0b0000_0000	R/W	Error injection mask for memory data parity bus

3.5 Memory Data Path Error Capture Monitor Registers

The memory data-path error capture monitors are used to latch data that causes ECC or parity errors from either the 60x bus or the memory data/parity bus. Separate monitors are provided for the high data, low data, and parity buses. The monitors are read-only. They are loaded automatically when the error detection registers detect any of the following:

- A memory read parity error; EDR1[2]
- Single-bit ECC error trigger exceeded; EDR1[2]
- A multi-bit ECC error; EDR2[3]
- A write parity error; EDR2[2]

When memory data path parity/ECC error data is loaded into the monitors, the capture flag in the MDP_ERR_CAP_MON_PAR is also set. This control bit remains set until explicitly cleared by the software. The set Capture Flag prevents subsequent errors from overwriting the data from the first failure. The Capture Flag is the only bit in the memory data path error capture monitors that is read/write.

3.6 DH Error Capture Monitor Register

Figure 17 shows the bits of the DH error capture monitor register.

Figure 17: DH Error Capture Monitor (MDP_ERR_CAP_MON_DH) — Offsets 0xF_F00C, 0xF0C

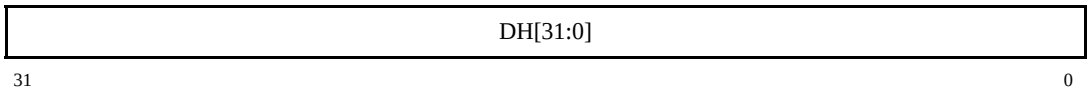


Table 17 shows the bit definitions of the DH error capture monitor register.

Table 17: DH Error Capture Monitor Bit Field Definitions

Bits	Name	Reset Value	R/W	Description
31–0	DH[31:0]	all 0s	Read	Capture monitor for memory data path data bus high

3.7 DL Error Capture Monitor Register

Figure 18 shows the bits of the DL error capture monitor register.

Figure 18: DL Error Capture Monitor (MDP_ERR_CAP_MON_DL) — Offsets 0xF_F010, 0xF10

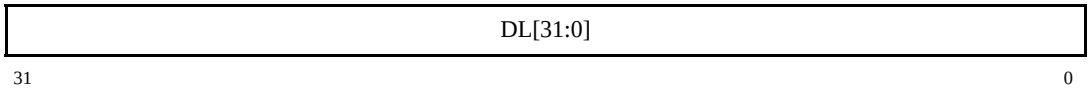


Table 18 shows the bit definitions of the DL error capture monitor register.

Table 18: DL Error Capture Monitor Bit Field Definitions

Bits	Name	Reset Value	R/W	Description
31–0	DL[31:0]	all 0s	Read	Capture monitor for memory data path data bus low

3.8 Parity Error Capture Monitor Register

Figure 19 shows the bits of the parity error capture monitor register and Table 19 shows the bit definitions.

Figure 19: Parity Error Capture Monitor (MDP_ERR_CAP_MON_PAR) — Offsets0xF_F014, 0xF14

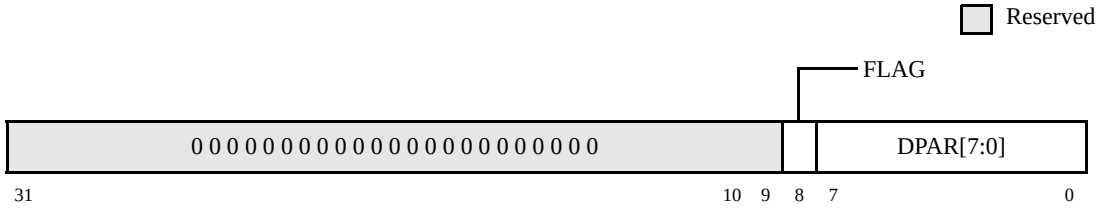


Table 19: Parity Error Capture Monitor Bit Field Definitions

Bits	Name	Reset Value	R/W	Description
31–9	—	all 0s	Read	Reserved
8	FLAG	0	R/W	Capture flag 0 No data captured 1 Data valid
7–0	DPAR[7:0]	0b0000_0000	Read	Capture monitor for memory data path data parity bus

4. Watchpoint Control Register (WP_CONTROL)

The watchpoint control register configures the watchpoint facility. It has fields that allow the user to enable the watchpoint facility, enable watchpoint interrupts, initialize the watchpoint counters, select the driver modes for TRIG_OUT, and set the watchpoint mode of operation. Figure 20 shows the format of WP_CONTROL. Table 20 shows the bit field definitions for WP_CONTROL. Detailed bit field information can be found in the *Tsi107 User Manual*.

The Watchpoint Control Register is included in this application note because bits [3-2], WP_ERR_INJ[0–1] must be initialized to 0b10 in order for the error injection logic to be enabled.

Figure 20: Watchpoint Control Register (WP_CONTROL)— Offsets 0xF_F048, 0xF48

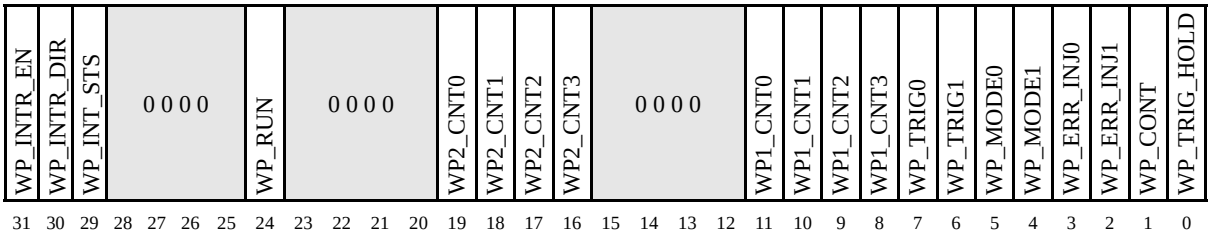


Table 20: Watchpoint Control Register Bit Field Definitions

Bits	Name	Reset Value	R/W	Description
31	WP_INTR_EN	0	R/W	0 Watchpoint interrupt disabled 1 Watchpoint interrupt enabled
30	WP_INTR_DIR	0	R/W	0 Local interrupt (INT asserted on a watchpoint interrupt) 1 PCI interrupt (INTA asserted on a watchpoint interrupt)
29	WP_INTR_STS	0	Read/Write 1 to clear	0 A watchpoint interrupt has not occurred. 1 A watchpoint interrupt has occurred.
28–25	—	0b0000	R	Reserved
24	WP_RUN	0	R/W	0 Stop a watchpoint scan. 1 Start a watchpoint scan.
23–20	—	0b0000	R	Reserved
19–16	WP2_CNT[0–3]	0b0000	R/W	
15–12	—	0b0000	R	Reserved
11–8	WP1_CNT[0–3]	0b0000	R/W	
7–6	WP_TRIG[0–1]	0b00	R/W	0xTRIG_OUT is disabled; TRIG_OUT is in high-impedance state. 10TRIG_OUT is enabled as an active high output. 11TRIG_OUT is enabled as an active low output.
5–4	WP_MODE[0–1]	0b00	R/W	
3–2	WP_ERR_INJ[0–1]	0b00	R/W	The watchpoint error injection field is used to enable the memory data path error injection/capture feature for ECC testability. 00 Watchpoint Monitor enabled 01 Reserved 10 Memory Data Path Error Injection/Capture enabled 11 Reserved

Table 20: Watchpoint Control Register Bit Field Definitions *(Continued)*

Bits	Name	Reset Value	R/W	Description
1	WP_CONT	0	R/W	0 One-shot scan mode 1 Continuous scan mode
0	WP_TRIG_HOLD	0	R/W	0 Trigger hold disabled 1Trigger hold enabled

5. How the error injection logic is put to use

The injection logic causes bits set in the mask registers to become inverted in a read or write transaction while in transit to the destination.

Initialization of the procedure would begin with clearing the Data Valid flag in the Parity Error Capture Monitor Register. This bit must be cleared prior to every transaction that is expected to cause an error.

For corruption to occur on the write path, the error injection logic would be enabled, and the bit(s) desired for corruption set in the appropriate mask register(s). Then a write to memory transaction would be initiated by the local processor. The write to memory with a bit set in a data path mask registers would corrupt the data being written to memory after the syndrome bits were calculated. The corrupted data would be stored in memory along with the syndrome code for the uncorrupted data. The result of the corruption would be observed during a read of that memory location. The error injection logic is disabled, and a memory read request issued by the local processor. The memory controller would read the data from the location including the syndrome bits, calculate the syndrome code for the data just read and compare it with the code obtained from the read. The data would be transferred in its uncorrupted form to the processor.

Intentional corruption in a read transaction would take the following process. A data pattern is written to memory with the error injection logic disabled. The contents in memory would be the intended value and along with the data would also be stored the correct syndrome code. Prior to the read transaction, the error injection logic would be enabled, and the desired bit(s) set in the mask register(s). The read transaction would be issued, and the Tsi107 memory controller would fetch the data and syndrome bits. The data would be corrupted by the masked bits in the read path, and a new syndrome code would be generated and compared to the code received from memory.

The ECC logic assumes that the syndrome code bits received from memory are the correct entity and that the data received from memory is the corrupt entity.

A corrupted syndrome code bit cannot be resolved into a single bit error and therefore would be reported as a multi-bit error.

Since the codes read from memory in either test did not match the calculated code, one of several actions would take place.

If the controller concluded that there was only one data bit in error, the bit would be corrected based on the syndrome code, increment the ECC Single-bit Error Counter Register, compare the resulting value against the ECC Single-bit Error Trigger Register and if necessary set the ECC Single-bit Error Exceeded bit in Error Detection Register 1. The read transaction would also be completed with the corrected data being sent to the read requestor.

If the controller concluded a multi-bit error had taken place, the transaction would be completed with the data as it was retrieved from memory, and the ECC Multi-bit Error bit being asserted in Error Detection Register 2.

In either case, the data deemed to be corrupt, along with its syndrome code can be obtained from the DH, DL and Parity Error Capture Monitor Registers.

6. Sample Code

Sample assembler code may be obtained from Appendix C of the *Tsi107 User Manual*. The code is not specific to the SDRAM interface but rather encompasses the whole bridge.

IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD-PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers who are designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only to develop an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third-party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising from your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Disclaimer Rev.1.01 Jan 2024)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit www.renesas.com/contact-us/.

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.