LSI LOGIC

www.DataSheet4U.com

MiniRISC™ MR4010 Superscalar Microprocessor

Reference Device

Contents

November 1996

www.DataSheet4U.com

Copyright © 1996 by LSI Logic Corporation. All rights reserved.

www.DataSheet4U.com

www.DataSheet4U.com

www.DataSheet4U.com

Figures

www.DataSheet4U.com

www.DataSheet4U.com

Tables

www.DataSheet4U.com

www.DataSheet4U.com

www.DataSheet4U.com

www.DataSheet4U.com

Overview

The MiniRISC MR4010 Microprocessor Reference Device is a chip implementation of the MiniRISC CW4010 Microprocessor core and shell. The MR4010 contains the following circuitry:

- The CW4010 shell, which is an unencrypted Verilog model containing the CW4010 core, the Multiply/Divide unit, Instruction cache (Icache), Data cache (Dcache), Memory Management Unit (MMU), and a Writeback Buffer
- A DRAM Controller (DRAMC) that controls the memory bus and an external synchronous DRAM array
- An SCbus/Lbus Converter (SCLC) that controls the Local I/O bus and external Lbus devices
- A Phase-Locked Loop (PLL) circuit that supplies clock inputs to the other modules in the MR4010

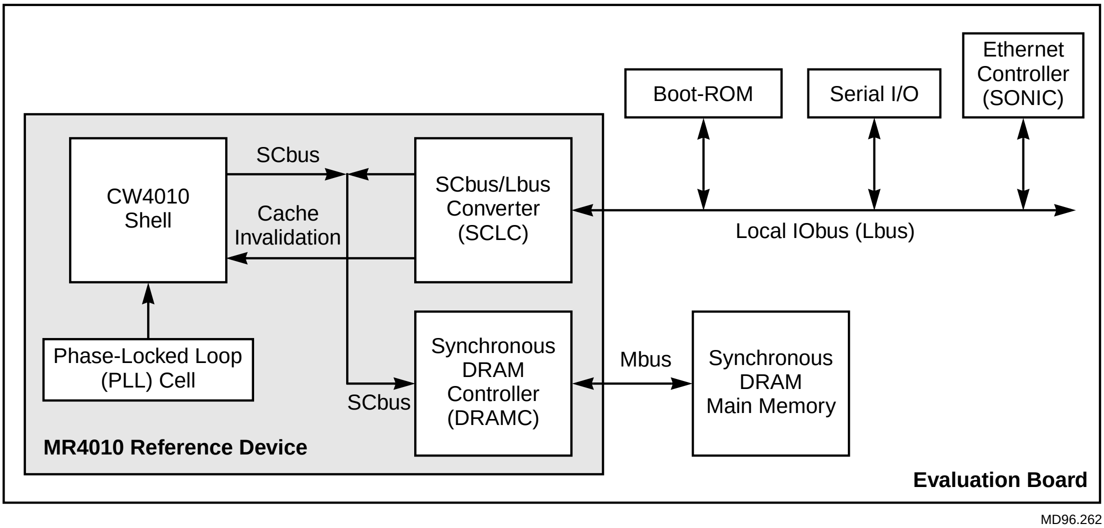
The MR4010 uses the maximum configuration CW4010 shell. You can disable optional modules by programming the Configuration Register in the CW4010 core's Coprocessor 0 (CP0). "MR4010 Functional Blocks," starting on page 9, provides further information about the different elements of the MR4010.

The MR4010 is housed on an evaluation board that allows you to use and test the microprocessor. In addition to the MR4010 Reference Device, the board also contains the DRAM array and Lbus facilities for plugging in devices such as a Boot-ROM, serial I/O devices, and an external Ethernet Controller. Figure 1 provides a block diagram of the MR4010 evaluation board circuitry.

www.DataSheet4U.com

www.DataSheet4U.com

Figure 1
Block Diagram of
MR4010 and Evaluation
Board Circuitry


Boot-ROM
Serial I/O
Ethernet Controller (SONIC)
CW4010 Shell
SCbus
Cache Invalidation
SCbus/Lbus Converter (SCLC)
Local IObus (Lbus)
Phase-Locked Loop (PLL) Cell
SCbus
Synchronous DRAM Controller (DRAMC)
Mbus
Synchronous DRAM Main Memory
MR4010 Reference Device
Evaluation Board
MD96.262

1 MR4010 Features

The MR4010 Reference Device has the following features:

- Superscalar microprocessor supports the MIPS-II 32-bit instruction set:
  - Executes up to two instructions per clock cycle
  - Four-deep write buffer
  - Load scheduling
  - R3000/R4000 compatible mode for Exception Return and Status Register
- 32-bit timer (R4000 compatible)
- SCbus watchdog timer with error reporting features
- Full internal scan testing
- Local IObus (Lbus) interface
  - Subset of the VLbus (486 bus); does not have I/O space, data/code, INTA, support for burst transactions
  - Demultiplexed 32-bit address bus and 32-bit data bus

www.DataSheet4U.com

www.DataSheet4U.com

- Direct interface to the SONIC™ Ethernet Controller
- Synchronous DRAM Controller, with 64-bit wide data transfer, interfaces to the following 16-Mbit DRAMs:
  - 1-Mword x 16-bit DRAM devices in an 8-Mbyte or 16-Mbyte configuration
  - 2-Mword x 8-bit DRAM devices in a 16-Mbyte or 32-Mbyte configuration
  - 4-Mword x 4-bit DRAM devices in an 32-Mbyte or 64-Mbyte configuration
- PLL circuit for system clock
- 3.3 V operation
- Up to 66 MHz microprocessor clock
- Packaged in a 299-pin CPGA (ceramic pin-grid array package)
- Maximum CW4010 configuration:
  - Direct-mapped or two-way set-associative Icache and Dcache
  - 1-Kbyte, 2-Kbyte, 4-Kbyte, or 8-Kbyte cache sets; organized as either direct-mapped (single set) cache with maximum cache size of 8 Kbytes, or as two-way set-associative cache with a maximum cache size of 16 Kbytes.
  - Memory management unit with 32-entry fully associative TLB (translation lookaside buffer)
  - Fast multiplier supporting multiply-accumulate operations

www.DataSheet4U.com

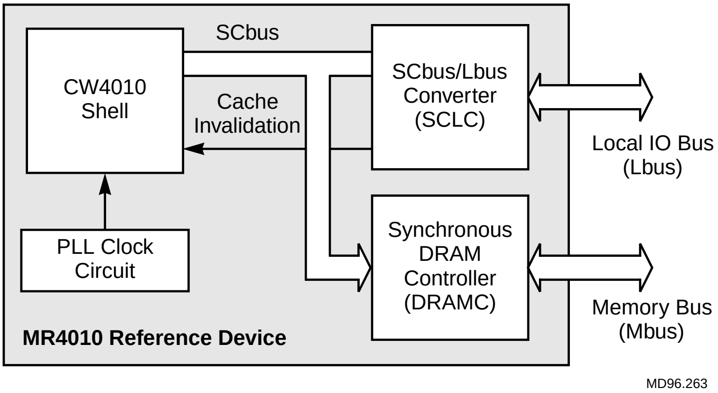
2 MR4010 Functional Blocks

This section describes the functional blocks that make up the MR4010 Reference Device:

- CW4010 shell
- SCbus/Lbus Converter
- DRAM Controller
- PLL clock circuit

Figure 2 shows the relationship between these blocks.

Figure 2 MR4010 Reference Device Block Diagram



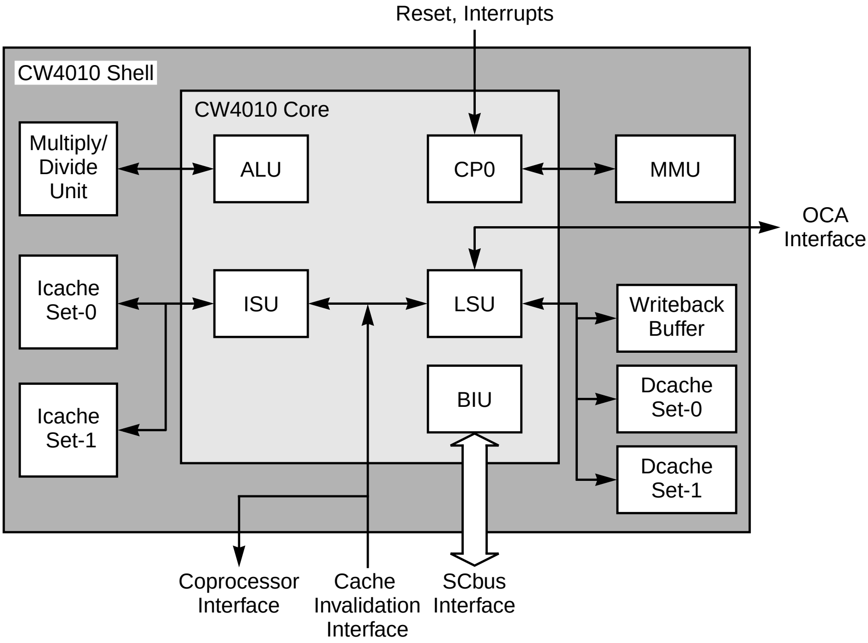
www.DataSheet4U.com

www.DataSheet4U.com

2.1 CW4010 Shell

The CW4010 shell consists of the CW4010 core and a number of optional modules, including Icache, Dcache, the Multiply/Divide unit, the MMU (Memory Management Unit), and the Writeback Buffer. You can modify the modules in the shell to fit your own ASIC design. Figure 3 shows a block diagram of the shell modules.

Figure 3
Block Diagram of CW4010 Shell Modules and CW4010 Core



2.1.1 CW4010 Core

The CW4010 core is part of LSI Logic's CoreWare® Library. It is an encrypted synthesizable Verilog model. As shown in Figure 3, it contains the basic microprocessor elements:

- Arithmetic Logic Unit (ALU)
- Instruction Scheduling Unit (ISU)
- Load/Store Unit (LSU)
- Bus Interface Unit (BIU)
- Coprocessor 0 (CP0)

The core is a hardmacro. It remains the same for each ASIC design, and can easily be reused.

www.DataSheet4U.com

www.DataSheet4U.com

The CW4010 core executes all MIPS-II 32-bit based instructions except for multiply/divide instructions, which are handled by the Multiply/Divide unit. The microprocessor is implemented as efficient dual six-stage pipelines. The pipelines have the traditional instruction fetch and execution stages. An additional queuing (Q) stage (one of the instruction fetch stages) removes the penalty cycle when a branch instruction is executed. Standard MIPS compiled code for R3000 and R4000 processors runs on the CW4010 core.

The CW4010 has an extended instruction set and implements ADDCIU (Add with Circular Mask Immediate Unsigned), MADD/MADDU (Multiply Accumulate (Unsigned)), MSUB/MSUBU (Multiply Subtract (Unsigned)), FFS/FFC (Find First Set/Clear Bit), SELSR/SELSL (Select and Shift Right/Left), WAITI (Wait Interrupt) and FLUSHID (Flush Instruction/Data Cache) instructions.

2.1.2 Multiply/Divide

The multiply/divide module supports Multiply-Add/Subtract operations as well as Multiply and Divide. The Multiply instruction executes in three cycles. The Multiply-Add/Subtract instruction is optimized to two cycles.

2.1.3 Memory Management Unit (MMU)

The MMU has 32 single-page entries, which are a subset of the R4000 32-bit addressing mode. Each page is individually specified to be 4 Kbyte or 16 Mbyte and may be cached or uncached.

2.1.4 Writeback Buffer

The CW4010 uses this buffer when the Dcache operates in Writeback mode. When a cache miss occurs at a dirty line, the dirty data is written into the Writeback Buffer instead of the main memory. This reduces the latency of the cache refill for missed addresses. Data in the Writeback Buffer is written into the main memory after the refill is completed.

2.1.5 Caches

The MR4010 has separate instruction and data caches—Icache and Dcache—that are part of the CW4010 shell. Both caches can be organized as direct-mapped or two-way set-associative caches. The cache controllers support configurations of 1, 2, 4, or 8 Kbytes for each set.

www.DataSheet4U.com

www.DataSheet4U.com

Thus, the smallest supported configuration is a 1 Kbyte direct-mapped cache, and the largest is a 16 Kbyte two-way set-associative cache, with 8 Kbytes per set. You can select between Writeback and Writethrough modes. You can also configure the Dcache for scratch pad RAM mode.

2.2 Synchronous DRAM Controller (DRAMC)

The DRAM Controller is part of the MR4010 Reference Device external to the CW4010 shell. It generates DRAM transactions according to requests from the CW4010 core or from the SCLC module. The DRAM Controller also generates initialization cycles and refresh cycles for DRAM.

2.3 SCbus to Local I/O Bus (Lbus) Controller (SCLC)

The SCLC module is part of the MR4010 Reference Device external to the CW4010 shell. It provides an interface between the internal CW4010 microprocessor bus, (SCbus), and the external Local I/O bus (Lbus). The Lbus connects boot-ROM, serial I/O devices, and the Ethernet Controller to the MR4010.

The CW4010 SCbus is a 32-bit address, 64-bit data bus. The Lbus, which is a subset of the industrial standard VLbus, is a 32-bit address, 32-bit data bus. The CW4010 uses the SCLC module to access devices on the Lbus. Devices on the Lbus access the DRAM main memory through the SCLC module and the DRAM Controller.

The CW4010 microprocessor generally has ownership of the SCbus and the Lbus. When a device on the Lbus wants to access the DRAM, it asserts the bus hold request signal on the Lbus. The SCLC module detects the asserted signal and then asserts the bus hold request to the CW4010. The CW4010 asserts the grant signal to the SCLC module, and the SCLC module then asserts the hold acknowledge signal to the Lbus device.

2.4 PLL Clock Circuit

The PLL circuit is part of the MR4010 Reference Device external to the CW4010 shell. It drives the clock signals to the CW4010 shell and the other modules that are part of the MR4010.

www.DataSheet4U.com

www.DataSheet4U.com

3 MR4010 Programming Model

The term ‘programming model’ refers to the way data is arranged in registers and in memory. You will find information on these subjects in the following areas of this technical summary:

- ♦ “System Configuration” on page 37
- ♦ “DRAM Modes and Programmable Configurations” on page 59
- ♦ “SCbus Timeout Watchdog Timer” on page 85
- ♦ “Cache Configuration and Maintenance” on page 87

In addition, *MiniRISC CW4010 Superscalar Microprocessor Core Technical Manual* provides information about the Memory Management Unit and Coprocessor 0 (CP0).

4 Signal Descriptions

This section describes the MR4010 signals in the following groupings:

- ♦ SCbus Interface

  The interface between the CW4010 shell and the DRAMC and the SCLC module implemented by means of the SCbus.

  External buffering required for certain SCbus signals.
- ♦ CW4010 Shell Interface

  Other signals that interface between the CW4010 shell and the DRAMC and SCLC.
- ♦ Mbus Interface

  The interface between the DRAM Controller and the DRAM array, with inputs and outputs referenced to the MR4010 Reference Device.
- ♦ Lbus Interface

  The interface between the SCLC and devices on the Lbus, with inputs and outputs referenced to the MR4010 Reference Device.
- ♦ Phase-Locked Loop (PLL) Interface

  The interface between the PLL clock generator and the CW4010 shell, with inputs and outputs referenced to the CW4010 shell.

www.DataSheet4U.com

www.DataSheet4U.com

- Test Signals

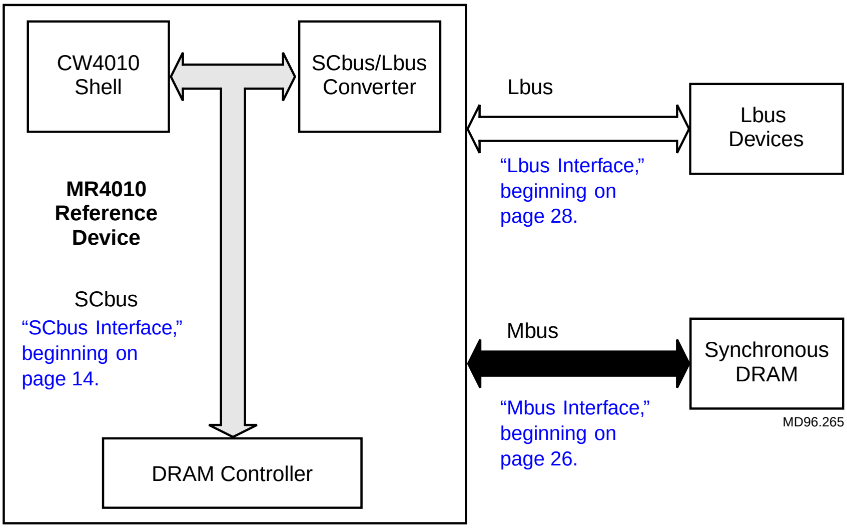
  Input pins that allow LSI Logic to test the MR4010.
- Core Monitor Signals

  Output signals that allow you to monitor the behavior of the CW4010 core.

Figure 4 shows the three major buses.

Each signal definition contains the mnemonic and the full signal name. Active LOW signals have an ‘n’ suffix, for example, SCRESETn. Active HIGH signals have a ‘p’ suffix, for example, MDQMp. ‘Assert’ means to drive the signal TRUE or active. ‘Deassert’ means to drive the signal FALSE or inactive.

Figure 4
MR4010 Buses


CW4010
Shell
SCbus/Lbus
Converter
MR4010
Reference
Device
SCbus
“SCbus Interface,” beginning on page 14.
DRAM Controller
Lbus
Lbus
Devices
“Lbus Interface,” beginning on page 28.
Mbus
Synchronous
DRAM
“Mbus Interface,” beginning on page 26.
MD96.265

4.1 SCbus Interface

Figure 5 shows how the CW4010 shell uses the SCbus to interface with the MR4010 Reference Device’s SCLC module and DRAM Controller. In the interface between the CW4010 and the SCLC, either module can function as the bus master or slave. In the interface between the CW4010 and the DRAM Controller, the CW4010 is always the master.

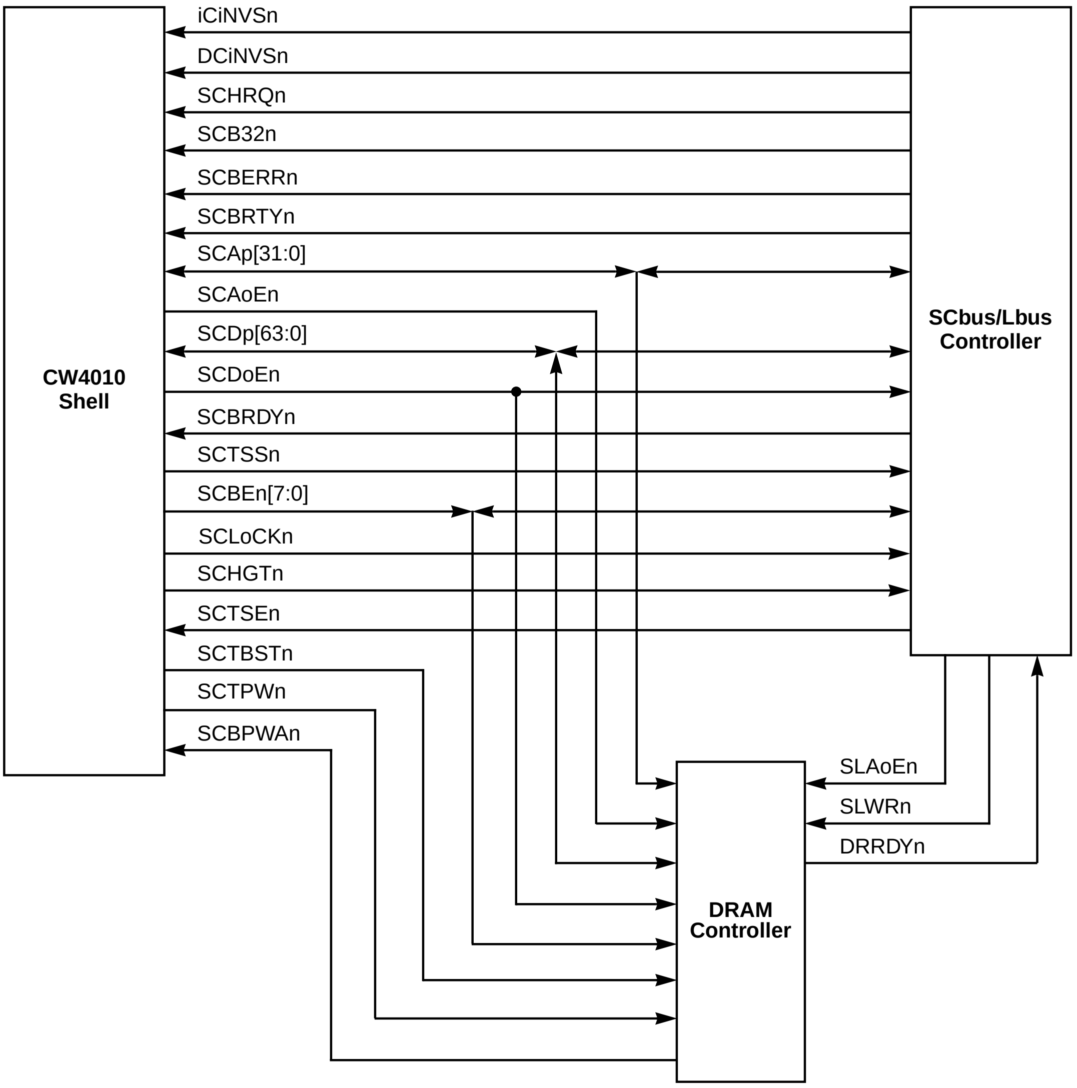
Figure 5 also shows the cache invalidation signals input to the CW4010 shell from the SCLC, and the address and write enable signal interface and the bus ready signal interface between the DRAMC and the SCLC.

www.DataSheet4U.com

www.DataSheet4U.com

You will need to provide external buffering for certain SCbus signals. "External Buffering for SCbus Signals" on page 21 provides information on this subject.

Figure 5
SCbus Interface



www.DataSheet4U.com

www.DataSheet4U.com

SCHRQn **Bus Hold Request** **Input to Shell from SCLC**

SCHRQn indicates that a device on the Lbus is requesting ownership of the SCbus. Bus hold request has the highest priority during bus arbitration. However, it cannot break continuous transactions of in-page writes and burst read/write transactions if those transactions are supported by an asserted SCTSEn, and SCHRQn must wait until SCTSEn is deasserted.

SCB32n **32-Bit Bus Width Sizing** **Input to Shell from SCLC**

SCB32n indicates that the external bus slave on the SC bus needs 32-bit bus sizing. The CW4010 core samples this signal on the rising edge of the clock that synchronizes the SCbus ready signal, SCBRDYn. If SCB32n is asserted for a 64-bit transaction, which is a doubleword or part of a burst transaction, the bus interface unit in the CW4010 core generates a subsequent 32-bit word transaction and packs data to 64 bits for a read transaction or unpacks data to 32 bits for a write transaction.

SCBERRn **Bus Error** **Input to Shell from SCLC**

The Lbus master device asserts SCBERRn to terminate the current transaction when a bus error occurs. If SCBRDYn, or the bus retry signal, SCBRTYn, is asserted at the same time as SCBERRn, SCBERRn has higher priority. SCBERRn is reported to the CP0 and the CP0 generates an exception.

SCBRTYn **Bus Retry** **Input to Shell from SCLC**

The Lbus master device asserts SCBRTYn when the current transaction has been terminated unsuccessfully and must be retried later. The control state goes back once to the idle state, then all bus requests are arbitrated again. If there are no higher priority requests and the Lbus master has asserted SCTSEn, there is one idle state between the first transaction and a retry transaction. If SCBRDYn and SCBRTYn are asserted at the same time, SCBRTYn has the higher priority.

SCAp[31:0] **Address Bus** **Bidirectional between Shell and SCLC**
Input to DRAMC

SCAp[31:0] is the 32-bit address bus for instruction fetch and data read/write operations. The bus signals are valid only when the address output enable signal, SCAoEn, is asserted. The enable signal remains valid throughout the

www.DataSheet4U.com

www.DataSheet4U.com

operation until SCBRDYn, SCBRTYn, or SCBERRn is asserted. The CW4010 asserts the signals on this bus and outputs them to the SCLC or the DRAMC. The Lbus master can also assert SCAp[31:0] and output them to the CW4010 shell through the SCLC.

SCAoEn **Address Output Enable** **Output from Shell to DRAMC**
When the CW4010 asserts this signal, it indicates that the address bus lines, SCAp[31:0], are valid. The signal remains active throughout the bus transaction. SCAoEn also enables SCTBSTn, SCTBEn, and SCTPWn. This signal is not valid at the same time as SLAoEn, which is the Address Output Enable signal output from the SCLC shell to the DRAMC described on page 20.

SCDp[63:0] **Data Bus** **Bidirectional between Shell, SCLC, and DRAMC**
SCDp[63:0] are the data bus signals. They are output from the shell for data read/write operations and for data Writeback to the Dcache. They are input to the shell for data read and instruction fetch transactions. The CW4010 shell samples the signals on the rising edge of the clock when SCBRDYn is asserted.The signals are valid throughout a write transaction where the CW4010 is writing to DRAM through the DRAMC, or the Lbus device is writing to the CW4010 or DRAMC through the SCLC. Byte ordering is little endian.

SCDoEn **Data Output Enable** **Output from Shell to SCLC and DRAMC**
The CW4010 asserts SCDoEn throughout a write transaction and outputs it to the SCLC or the DRAMC. The signal indicates that the current transaction is a write transaction, and it also enables data output. It performs the same function for a CW4010 write transaction to DRAM that SLWRn (page 20) performs for an SCLC write transaction to DRAM.

SCTSSn **Transaction Start Strobe** **Output from Shell to SCLC**
The CW4010 asserts SCTSSn for one clock cycle at the beginning of a transaction to indicate that a transaction has started. If the next transaction begins immediately, the CW4010 asserts SCTSSn continuously.

www.DataSheet4U.com

www.DataSheet4U.com

SCBEn[7:0] **Byte Enable** **Bidirectional between Shell and SCLC and input to DRAMC**

SCBEn[7:0] indicate which byte positions are valid for a read or write transaction. The CW4010 asserts the signals and outputs them to the SCLC or the DRAMC. The Lbus device can also assert the signals and input them to the CW4010 through the SCLC. Only one of the signals is asserted during a byte read or byte write transaction. All signals are asserted for a doubleword or burst transaction.

SCTBSTn **Burst Transaction** **Output from Shell to DRAMC**

The CW4010 asserts SCTBSTn and outputs it to the DRAMC to indicate that a transaction is taking place during which four doublewords will be moved, and that the first doubleword is currently being moved. It deasserts the signal after the first word has been transferred and during singleword transactions.

SCLoCKn **Bus Lock** **Output from Shell to SCLC**

The CW4010 asserts SCLoCKn to indicate that it wishes to lock the SCbus and restrict ownership. The CW4010 asserts the signal when a read transaction is started by executing a LoadLink instruction in an uncached area or a Writethrough cached area. It deasserts the signal just before a write transaction is started by executing a StoreConditional instruction. During read and write transactions, the CW4010 asserts the signal continuously, preventing ownership from changing during one of these transactions. If a StoreConditional transaction hits the Dcache in a Writeback cached read while SCLoCKn is asserted, an incorrect condition exists, and the CW4010 deasserts SCLoCKn without completing any bus transactions.

SCTPWn **Next Transaction is In-Page Write** **Output from Shell to DRAMC**

When asserted, this signal indicates that the next transaction is in the same DRAM page as defined in the Configuration Register. When the CW4010 asserts SCTPWn, a maximum of four write transactions take place one after the other, even if there is an instruction fetch request or data read request. If there are four continuous write transactions, the CW4010 asserts SCTPWn from the first through the last (fourth)

www.DataSheet4U.com

www.DataSheet4U.com

transaction. The CW4010 asserts SCTPWn from the beginning of one in-page write transaction to the end of that transaction. The write buffer in the CW4010's LSU checks to see if the subsequent write request is in the same page.

SCHGTn **Bus Hold Grant** **Output from Shell to SCLC**

The CW4010's bus interface unit enters the hold state and asserts SCHGTn to indicate that it is releasing SCbus ownership in response to a bus hold request (SCHRQn) from one of the devices on the Lbus.

SCTSEn **Transaction Start Enable** **Input to Shell from SCLC**

SCTSEn enables or disables a new SCbus transaction. Transaction requests are arbitrated only when SCTSEn is asserted. The Lbus device must deassert then assert the signal when SCBRDYn is asserted to allow an idle cycle between the two transactions. During the time SCTSEn is deasserted, the CW4010's bus interface unit repeats the idle state.

SCBRDYn **Bus Ready** **Input to Shell from SCLC**

The SCLC asserts SCBRDYn when the current transaction is terminated, indicating that the SCbus is available. The signal remains active (LOW) until the next transaction starts. The SCLC deasserts the signal to indicate that the SCbus is not available. The SCLC receives a bus-ready signal, DRRDYn from the DRAMC (see page 20), merges DRRDYn with its own bus ready signal, and drives SCBRDYn, which is output to the CW4010 shell.

SCBPWAn **Bus In-Page Write Accept** **Input to Shell from DRAMC**

The DRAMC asserts SCBPWAn to indicate that it accepts in-page write transactions. The CW4010 samples the signal on the rising edge of the clock that synchronizes SCBRDYn. If the CW4010 has not asserted SCTPWn, asserting or deasserting SCBPWAn has no significance.

www.DataSheet4U.com

www.DataSheet4U.com

iCiNVSn **Icache Invalidation Strobe** **Input to Shell from SCLC**

The SCLC asserts this signal to indicate that the Icache invalidation address bus is valid and there is no need for a snooping sequence. If the cache tag is not coincident with higher address bits, the line is not invalidated.

DCiNVSn **Dcache Invalidation Strobe** **Input to Shell from SCLC**

The SCLC asserts this signal to indicate that the Dcache invalidation address bus is valid and there is no need for a snooping sequence. If the cache tag is not coincident with higher address bits, the line is not invalidated.

SLAoEn **Address Output Enable** **Output from SCLC to DRAMC**

When the SCLC asserts this signal, it indicates that the address bus lines, SCAp[31:0], are valid. The signal remains active throughout the bus transaction. SCAoEn also enables SCTBSTn, SCTBEn, and SCTPWn. This signal is not valid at the same time as SCAoEn, which is the Address Output Enable signal output from the CW4010 shell to the SCLC described on page 17.

SLWRn **SCLC Write Enable** **Output from SCLC to DRAMC**

The SCLC asserts this signal throughout a DRAM write operation and outputs it to the DRAMC. It performs the same function for a CW4010 write transaction to DRAM that SLWRn (page 17) performs for an SCLC write transaction to DRAM.

DRRDYn **DRAM Ready** **Output from DRAMC to SCLC**

The DRAMC asserts DRRDYn when the current DRAM transaction is terminated, indicating that the bus is available. The signal remains active (LOW) until the next transaction starts. The DRAMC outputs the signal to the SCLC, which merges DRRDYn with its own bus ready signal (see page 19), and drives SCBRDYn, which is output to the CW4010 shell.

www.DataSheet4U.com

www.DataSheet4U.com

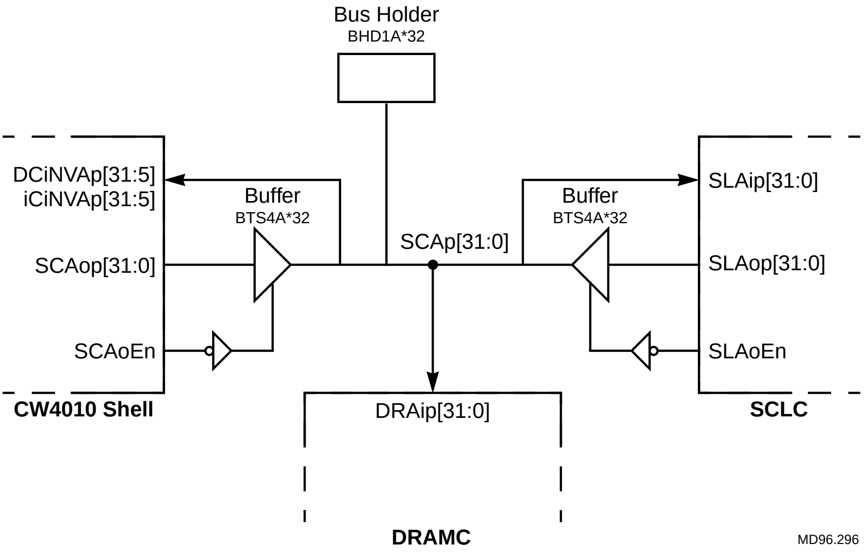
4.2 External Buffering for SCbus Signals

You must provide external buffering for certain SCbus signals, including

- Address bus SCAp[31:0]
- Address output enable signal SCAoEn and SLAoEN
- SC data bus SCDp[63:0]
- Data output enable SCDoEn
- SCbus byte enable SCBEn[7:0]

Figure 6 shows an example of a buffer configuration in which the bidirectional address bus is buffered at the SCLC and CW4010 ends by BTS4A*32 3-state buffers. When the CW4010 asserts SCAoEn, the signal enables the buffer at the CW4010 end. When the SCLC asserts SLAoEn, the signal enables the buffer at the SCLC end.

Figure 6 Buffering for SCAp[31:0] Address Bus

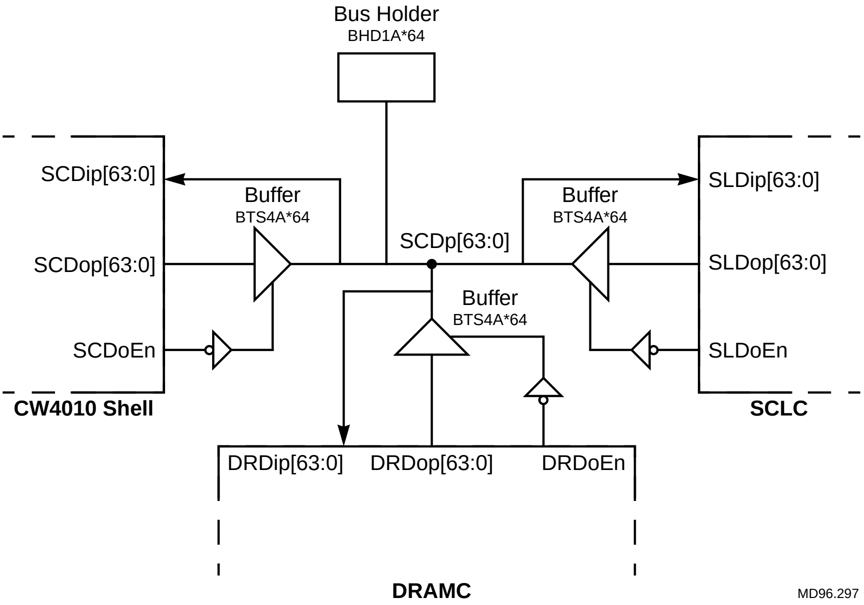


www.DataSheet4U.com

www.DataSheet4U.com

Figure 7 shows an example of a buffer configuration in which the SC data bus is buffered at the SCLC and CW4010 ends by BTS4A*64 3-state buffers. When the CW4010 asserts SCDoEn, the signal enables the buffer at the CW4010 end. When the SCLC asserts SLDoEn, the signal enables the buffer at the SCLC end. A BTS4A*64 buffer also buffers the data output from the DRAMC. This buffer is enabled when the DRAMC asserts DRDoEn.

Figure 7
Buffering for SCDp[63:0] Data Bus

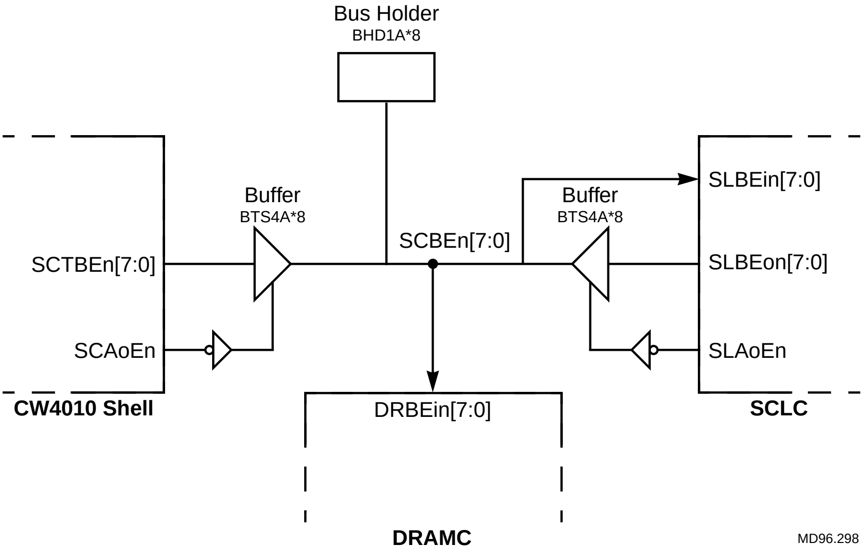


www.DataSheet4U.com

www.DataSheet4U.com

Figure 8 shows an example of a buffer configuration in which the SC byte enable signals, SCTBEn[7:0], are buffered at the SCLC and CW4010 ends by BTS4A*8 3-state buffers. When the CW4010 asserts SCAoEn, the signal enables the buffer at the CW4010 end. When the SCLC asserts SLAoEn, the signal enables the buffer at the SCLC end.

Figure 8
Buffering for SCBEn[7:0] Byte Enable



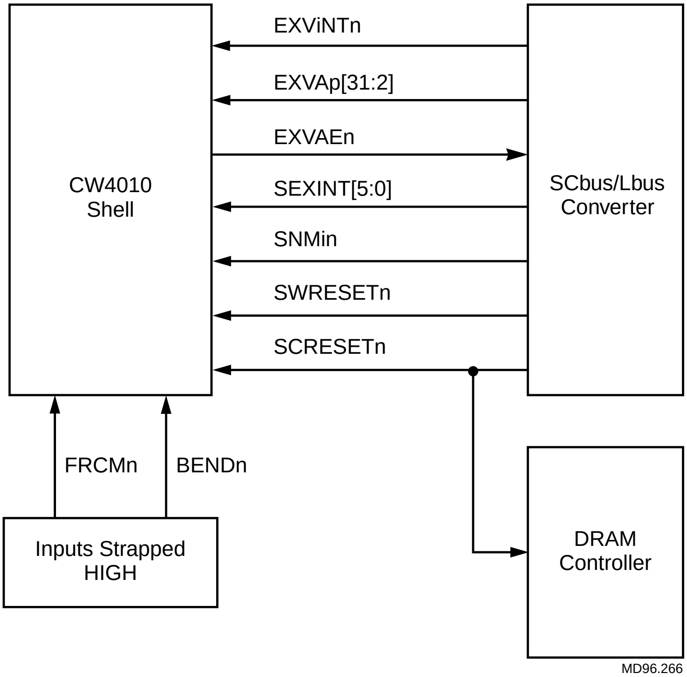
www.DataSheet4U.com

www.DataSheet4U.com

4.3 CW4010 Shell Interface

Figure 9 shows the internal interface that links the CW4010 shell, the SCLC, and the DRAMC. "SCbus Interface" beginning on page 14 describes the SCbus interface between these modules.

Figure 9
Shell Interface Overview



EXViNTn **External Vectored Input** **Input to Shell from SCLC**
The SCLC drives this signal. When the CW4010 shell receives the signal, it generates an external interrupt exception.

EXVAp[31:2]

External Vectored Interrupt Address **Input to Shell**
The CW4010 shell accepts the external vectored interrupt address when the SCLC asserts EXVApEn. The CW4010 writes the address directly into the program counter. The address bus must remain stable until EXVApEn is asserted.

EXVAEn **EXVAp Enable** **Output from Shell to SCLC**
This is the enable signal for the vectored interrupt address. The CW4010 asserts this signal to acknowledge the address.

www.DataSheet4U.com

www.DataSheet4U.com

SEXiNTn[5:0]

External Interrupt Input [5:0] **Input to Shell**

The DRAMC or SCLC asserts one of the SEXiNTn signals to cause the CP0 in the CW4010 core to generate an interrupt exception. The assertion is registered in the IP field of the CW4010 Cause Register. The SCLC should continue to assert the signals until the exception routine has serviced the interrupt.

The CW4010 does not recognize interrupts if the interrupt enable bit in the Status Register is not set. The CW4010 can therefore disable individual interrupt inputs by clearing the related bits. However, the interrupt inputs are still registered in the IP field of the Cause Register.

External Interrupt Input [5:0] are synchronized to the system clock, SCLKp, internally in the CW4010 shell.

SNMin **Nonmaskable Interrupt** **Input to Shell**

This input is synchronized internally to the system clock, SCLKp. When the SCLC asserts this signal, the CW4010 recognizes a nonmaskable interrupt. The CP0 then generates a nonmaskable interrupt exception (0xBFC0 0000).

SWRESETn **Warm Reset** **Input to Shell**

This input initiates a warm reset for the MR4010. Inside the MR4010, this signal is synchronized to the system clock, SCLKp. The MR4010 enters the warm reset condition when the SCLC asserts SWRESETn and immediately exits from the warm reset when the SCLC deasserts SWRESETn.

SCRESET **Cold Reset** **Input to Shell**

This input initiates a cold reset for the MR4010. Inside the MR4010, this signal is synchronized to the system clock, SCLKp. The MR4010 enters the cold reset condition when the SCLC asserts SCRESETn and immediately exits from the cold reset when the SCLC deasserts SCRESETn.

FRCMn **Force Cache Miss (Strap Input)** **Input to Shell**

This input is used for system debug. Under normal operating conditions, you should strap it HIGH. To use it for debug, you should assert it by tying it LOW. When LOW, it forces a cache miss for the Icache and the

www.DataSheet4U.com

www.DataSheet4U.com

Dcache in the CW4010 shell. The CW4010 treats this event as an access to an uncached area. The CW4010 can then read and write all instructions and data as uncached, regardless of the memory segment and the MMU.

BENDn **Big Endian (Strap Input)** **Input to Shell**
This input affects the byte positions for sizing and load/store data alignment. When the input is LOW (asserted), the CW4010 uses big-endian addressing. The MR4010 uses only little-endian addressing, so you tie this input HIGH.

4.4 Mbus Interface

Figure 10 shows the 89 Mbus signals that the MR4010 uses to connect the MR4010 DRAM Controller to the synchronous DRAMs in the main memory array. Inputs and outputs are referenced to the DRAM Controller.

MAp[11:0] **Multiplexed Memory Address Bus** **Output**
These multiplexed signals carry row and column addresses. MRASn strobes the row addresses into the DRAMs, and MCASn strobes the column addresses. "Address Bit Assignment" on page 58 provides detailed information about the address bus.

During memory initialization, the DRAM Controller uses MAp[11:0] to write the 12-bit Mode Register in each DRAM.

MDp[63:0] **Memory Data Bus** **Bidirectional**
This 64-bit bidirectional data bus carries data between the MR4010 and the memory array. The direction of data flow is controlled by MWEn.

MCSn[1:0] **Memory Chip Select** **Output**
MCSn[1:0] select between Banks 0 and 1 in the DRAM array. The DRAM Controller asserts MSCn0 to select the DRAMs that make up Bank 0, and asserts MSCn1 to select the DRAMs in Bank 1. If only one bank of DRAMs is installed, the DRAM Controller asserts MCSn0.

MRASn **Memory Row Address Strobe** **Output**
The MR4010 asserts MRASn to strobe memory row addresses into the memory devices.

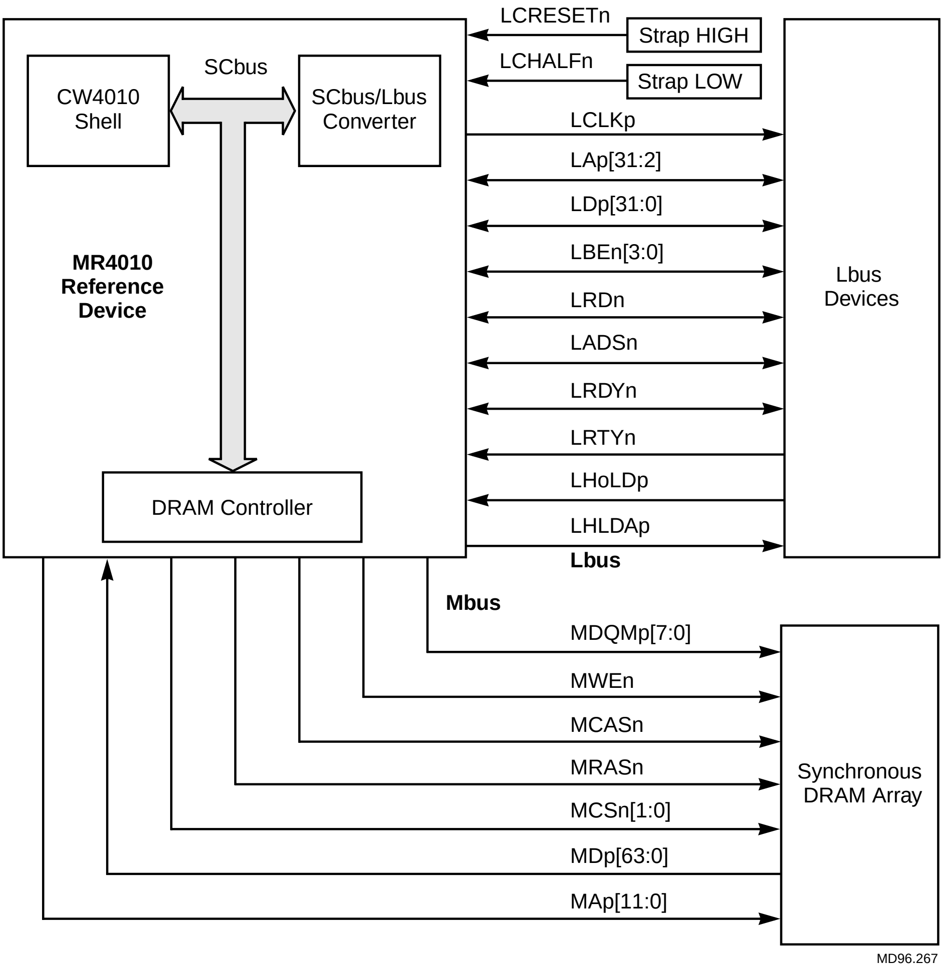
www.DataSheet4U.com

www.DataSheet4U.com

MCASn **Memory Column Address Strobe** **Output**
The MR4010 asserts MCASn to strobe memory column addresses into the memory devices.

MWEn **Memory Write Enable** **Output**
The MR4010 asserts MWEn to enable a write operation and deasserts MWEn to enable a read operation.

Figure 10
Mbus and Lbus
Interface



www.DataSheet4U.com

www.DataSheet4U.com

MDQMp[7:0] **Memory Data Enable/Mask** **Output**

This is an 8-bit data mask used only during write operations. When asserted, each bit of the mask selects one byte of data, as shown in the examples below, to enable write operations in individual bytes of the data word. During read operations, the DRAM Controller asserts all the mask bits to select all data bytes.

Mask	Byte Selected	DRAM Byte/Number
	8-Bit Wide DRAM	
MDQMp 7	MDp[63:56]	7
MDQMp 6	MDp[55:48]	6
MDQMp 5	MDp[47:40]	5
MDQMp 4	MDp[39:32]	4
MDQMp 3	MDp[31:24]	3
MDQMp 2	MDp[23:16]	2
MDQMp 1	MDp[15:8]	1
MDQMp 0	MDp[7:0]	0

Mask	Byte Selected	DRAM Byte/Number
	16-Bit Wide DRAM	
MDQMp 7	MDp[63:56]	3 (Upper Byte)
MDQMp 6	MDp[55:48]	3 (Lower Byte)
MDQMp 5	MDp[47:40]	2 (Upper Byte)
MDQMp 4	MDp[39:32]	2 (Lower Byte)
MDQMp 3	MDp[31:24]	1 (Upper Byte)
MDQMp 2	MDp[23:16]	1 (Lower Byte)
MDQMp 1	MDp[15:8]	0 (Upper Byte)
MDQMp 0	MDp[7:0]	0 (Lower Byte

4.5 Lbus Interface

Figure 10 shows the 75 Lbus signals that connect the SCLC in the MR4010 with external Lbus devices. The MR4010 functions as the bus master to access external devices on the Lbus, such as boot-ROM, serial devices, and the Ethernet Controller. These Lbus devices can also function as bus master to access the DRAM through the SCLC and the DRAMC.

This section describes the signals on the Lbus. Inputs and outputs are referenced to the MR4010. Since the MR4010 or an Lbus device can be bus master, some signals that are typically unidirectional, such as the read signal LRDn, are bidirectional. When the MR4010 is bus master and asserts LRDn, it enables a read operation in one of the Lbus devices. When an Lbus device is bus master, it can assert LRDn to read data from the DRAM.

www.DataSheet4U.com

www.DataSheet4U.com

LCRESETn **LCLK Divider Reset** **Input**

You can use this input for testing. Normally you should strap it HIGH (deasserted) on the system board, which means the signal is deasserted.

LCHALFn **Lbus Clock Speed** **Input**

This signal sets the clock speed for the Lbus. When a device on the Lbus drives this signal LOW, it divides the SCbus clock (SCLKp) by two and the MR4010 outputs a clock signal (LCLKp) that is half the frequency of SCLKp. When the signal is HIGH, it divides the clock by four, and the MR4010 outputs a clock signal (LCLKp) that is one quarter the frequency of SCLKp.

LCLKp **Lbus Clock** **Output**

This output is derived from the SCbus clock, SCLKp. The Lbus clock rate is either half or quarter the clock rate of SCLKp, depending on the state of the LCHALFn input to the MR4010.

LAp[31:2] **Lbus Address Bus** **Bidirectional**

When the MR4010 is master of the Lbus, it outputs the address that is used to access one of the devices on the Lbus. If one of the devices on the Lbus is bus master, it inputs the address that the MR4010 uses to access the DRAM.

LDp[31:0] **Lbus Data Bus** **Bidirectional**

This 32-bit bidirectional data bus transfers data between the devices on the Lbus and the MR4010. The read/write signal, LRDn*,* controls the direction of data flow on the Lbus.

LBEn[3:0] **Lbus Byte Enables** **Bidirectional**

When the master device drives these signals active (LOW), they enable data on the Lbus, as shown below. The read/write signal, LRDn, controls the direction of data flow.

Byte Enable Signal	Byte Bits	Byte Number
LBEn 3	LDp[31:24]	Byte 3
LBEn 2	LDp[23:16]	Byte 2
LBEn 1	LDp[15:8]	Byte 1
LBEn 0	LDp[7:0]	Byte 0

www.DataSheet4U.com

www.DataSheet4U.com

LRDn **Lbus Read** **Bidirectional**

The master device asserts this signal to enable a read operation and deasserts it to enable a write operation. When the MR4010 asserts LHLDAp and grants bus ownership to an Lbus master device, the master device inputs this signal to the MR4010. When the MR4010 is the bus master, it inputs this signal to the Lbus device. The initiating device must synchronize this signal to the Lbus clock, LCLKp.

LADSn **Lbus Address Strobe** **Bidirectional**

This signal strobes the Lbus addresses. The bus master asserts it at the first LCLKp cycle of a transaction. When the MR4010 asserts LHLDAp and grants bus ownership to an Lbus master device, the master device inputs this signal to the MR4010. When the MR4010 is the bus master, it inputs this signal to the Lbus device. The initiating device must synchronize this signal to the Lbus clock, LCLKp.

LRDYn **Lbus Data Ready** **Bidirectional**

When it is asserted, this signal terminates a transaction. When the MR4010 asserts LHLDAp and grants bus ownership to an Lbus master device, the master device inputs this signal to the MR4010. When the MR4010 is the bus master, it inputs this signal to the Lbus device. The initiating device must synchronize this signal to the Lbus clock, LCLKp

LRTYn **Lbus Retry** **Input**

When an Lbus master device asserts this signal and inputs it to the MR4010, the MR4010 temporarily aborts any transaction in progress and initiates the transaction again later. The initiating Lbus device must synchronize the signal to the rising edge of LCLKp.

LHoLDp **Lbus Hold Request** **Input**

An Lbus device asserts LHoLDp to request ownership of the Lbus. The initiating device must synchronize the signal to the rising edge of LCLKp.

www.DataSheet4U.com

www.DataSheet4U.com

LHLDAp **Lbus Hold Acknowledge** **Output**

The MR4010 asserts this signal in response to an LHoLDp input from an Lbus device. When asserted, the signal grants a bus hold and allows the Lbus device to take bus ownership.

4.6 Phase-Locked Loop (PLL) Clock Signals

The PLL circuit generates the clock inputs for the CW4010 shell and for the other modules that are part of the MR4010. The PLL is part of the MR4010 Reference Device, as shown in Figure 1 on page 7. This section describes the PLL signals. "PLL Circuit," on page 36 provides further information. The test signals associated with the PLL circuit are not for general use and are therefore deasserted by strapping them LOW if they are active-high signals, and strapping them HIGH if they are active-low signals.

PLLREFp **System Clock Reference** **Input**

This is the system reference clock, input to the CW4010 by the PLL circuit.

PLLENp **VCO Enable(1)/Disable(0)** **Strapped Input**

This signal is input to the PLL circuit and enables the PLL circuit when it is active (HIGH). It is strapped HIGH on the main circuit board so that the PLL circuit is always enabled.

PLLLP2p **VCO Input and Loop Filter** **Filter Pin**

You must connect an RC (resistor/capacitor) circuit for the PLL filter between pins PLLLP2p and PLLAGND on the PLL circuit, as shown in Figure 11, on page 36.

PLLiDDTp **Test Enable Input** **Input to PLL Circuit**

This signal enables test inputs when it is active (HIGH). It is strapped LOW on the main circuit board.

PLLCTop **Test Counter** **Open Output from PLL Circuit**

This signal is an open pin on the board.

PLLCTRn **Test Counter Reset** **Input to PLL Circuit**

This signal is strapped LOW on the board.

PLLTSTp **Test Enable** **Input to PLL Circuit**

This signal is strapped LOW on the board, which means that testing is generally disabled.

www.DataSheet4U.com

www.DataSheet4U.com

PLLTDp **Test Data (Clock)** **Input to PLL Circuit**
This signal is strapped LOW on the board.

PLLVDD **PLL Power** **Input to PLL Circuit**
This signal provides VDD power.

PLLVSS **PLL Ground** **Ground**
This is the ground for the PLL circuit.

PLLAGND **PLL Analog Ground** **Ground**
This is the analog ground for the PLL circuit. You must connect an RC (resistor/capacitor) circuit for the PLL filter between pins PLLLP2p and PLLAGND on the PLL circuit, as shown in Figure 11 on page 36.

Test Signals

There are nine pins on the MR4010 chip that allow designers at LSI Logic to test devices on the board using an LSI tester. When the pins are not being used for testing, you must deassert all inputs by strapping active-high signals LOW and active-low signals HIGH. You must leave all outputs open. Inputs and outputs are referenced to and from the MR4010.

SCANCRip **CW4010 Core Scan** **Input**
This input allows you to initiate a scan of the CW4010 core. Strap it LOW on the board.

SCANCRop **CW4010 Core Scan** **Output**
This output allows you to read the results of the CW4010 core scan. Leave it open on the board.

SCANKZip **MR4010 Peripherals Scan** **Input**
This input allows you to initiate a scan of the peripheral circuitry that is part of the MR4010 Reference Device. Strap it LOW on the board.

SCANKZop **MR4010 Peripherals Scan** **Output**
This output allows you to read the results of the MR4010 peripherals scan. Leave it open on the board.

SCANENBp **Global MR4010 Scan Enable** **Input**
This input allows you to initiate a global scan of the MR4010. Strap it LOW on the board.

www.DataSheet4U.com

www.DataSheet4U.com

SCANMONp **Scan Mode Monitor** **Output**

This output allows you to monitor the scan mode. Leave it open on the board.

ZSTATEn **Global 3-State Control** **Input**

This input is reserved for factory use during testing. Strap it HIGH on the board.

PARAMOUTp **Parametric Nand Tree** **Output**

This output allows you to check the parametric NAND tree. It is reserved for factory use during testing. Leave it open on the board.

TESTMp **Test Mode for Scan** **Input**

This input is reserved for factory use during testing. Strap it LOW on the board through a 10 K resistor.

4.7 CW4010 Core Monitor Signals

There are 11 pins that enable you to monitor the behavior of the CW4010 core. When you are not using the outputs to monitor the core, make sure they are open.

BRLiKFn **CW4010 Branchlikely of Even Slot is False** **Output**

The CW4010 asserts BRLiKFn when the Branch Likely instruction is in an even slot and it is false. If, at this time, a coprocessor has a valid instruction in the EX stage, the instruction must be cancelled. It is not necessary to check whether the instruction in the EX stage is in an even or odd slot, since the CW4010 asserts BRLiKFn only when the Branch Likely instruction is in the even slot. If the Branch Likely instruction in the even slot is not taken, the instruction in the odd slot must be nullified although it has already been started.

MCLKp **Internal Clock Monitor** **Output**

This output from the internal clock allows you to check the clock phase. When you are not using the pin to check the clock, the output should be open.

PCANCRn **CW4010 Pipeline Cancel at CR Stage** **Output**

When one or more exceptions occur, the pipeline is cancelled at the CR stage and the CW4010 asserts PCANCRn. Coprocessor pipelines must be cancelled to prevent a second execution of the coprocessor instruction under either of the following conditions: when the coprocessor returns from an exception handler; or when the

www.DataSheet4U.com

www.DataSheet4U.com

coprocessor has finished executing an LWCz instruction that caused a TLB (translation lookaside buffer) miss. The WB stage is not cancelled when PCANCRn is asserted.

PCANoDDn **CW4010 Pipeline Cancel to Odd** **Output**

PCANoDDn is valid only when PCANCRn is asserted. The signal informs coprocessors whether the cancellation is for an odd or even slot. When the CW4010 asserts the signal, cancellation applies to the odd slot. When it deasserts the signal, cancellation applies to both even and odd slots.

The coprocessor must track which slot it is executing in based on the CPXoDDn signal. When the CW4010 asserts both PCANCRn and PCANoDDn and the coprocessor instruction is in the odd slot, the instruction must be cancelled. When the CW4010 asserts PCANCRn and deasserts PCANoDDn, the coprocessor instruction must be cancelled regardless of the slot in which it is operating. This signal is valid at the CR stage of the pipeline.

PSTALLn **CW4010 Pipeline Stall Monitor** **Output**

The CW4010 asserts this signal to indicate that all stages of the CW4010 pipelines are stalled. Pipelines must be stalled when they are executing instructions. This signal is valid at any stage of the pipelines.

SCAoEn **SCbus Address Output Enable** **Output**

The CW4010 asserts SCA0En to indicate that the address output bus SCAop[31:0] lines are valid. The CW4010 asserts the signal when the BIU is performing an SCbus transaction, and the signal remains active throughout the operation.

SCBRDYn **SCbus Bus Ready** **Output**

The SCLC asserts SCBRDYn when the current transaction is terminated. When asserted, it indicates that the SCbus is available. The signal remains active (LOW) until the next transaction starts. The SCLC then deasserts the signal to indicate that the SCbus is not available.

www.DataSheet4U.com

www.DataSheet4U.com

SCDoEn **SCbus Data Output Enable** **Output**

The CW4010 asserts SCDoEN to indicate that the data output signals SCDop[63:0] are valid. The CW4010 asserts the signal throughout the write transaction to indicate that the current transaction is a write transaction and to enable data output.

SCiFETn **SCbus Instruction Fetch** **Output**

SSCiFETn indicates that the BIU is fetching an instruction for monitoring purposes. The CW4010 drives the signal low at this time and outputs it to external logic.

SCTBSTn **SCbus Burst Transaction** **Output**

The CW4010 uses burst transactions to the DRAM Controller. When the CW4010 asserts SCTBSTn, it indicates that a DRAM transaction is taking place during which four doublewords will be moved, and that currently the first doubleword is being moved. The CW4010 deasserts the signal after the first word has been transferred and during singleword transactions.

SCTSSn **SCbus Transaction Start Strobe** **Output**

When the CW4010 asserts SCTSSn, it indicates that a transaction has started. The CW4010 asserts the signal for one clock cycle at the beginning of a transaction. If the transaction lasts through one cycle and the next transaction begins immediately, the CW4010 asserts SCTSSn continuously.

SUSPEXn **CW4010 Suspend EX Stage** **Output**

The CW4010 Instruction Scheduler Unit (ISU) asserts SUSPEXn to request coprocessors to suspend the instruction in the EX stage. The instruction in the EX stage must be held until the ISU deasserts SUSPEXn. Instructions in the CR and WB stages must be completed. This signal is valid at the EX stage of the pipeline.

www.DataSheet4U.com

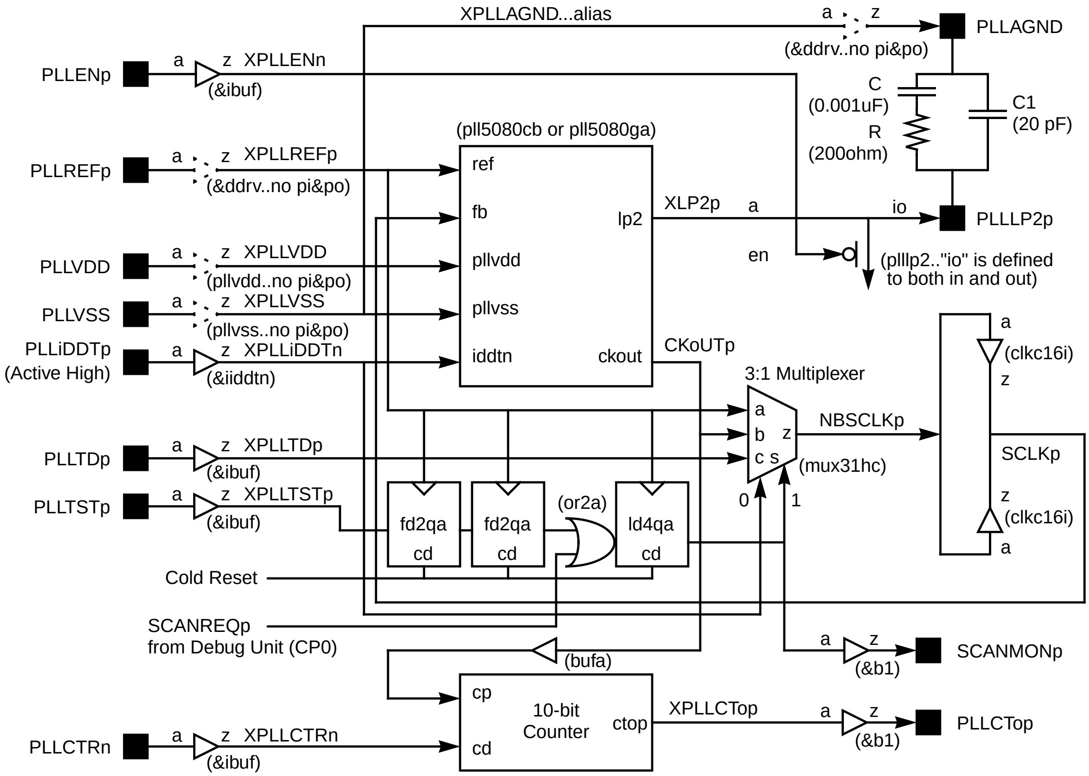
www.DataSheet4U.com

5 PLL Circuit

Figure 11 shows the layout of the MR4010 PLL circuit. This circuit is an LSI Logic PLL cell (pll5080cb) used for cell-based design. The system clock drives the PLLREFp input. You need to provide capacitance devices and a resistor between PLLLP2p and PLLAGND. You must connect other PLL circuit inputs to VDD or GND. SCANMONp and PLLCTop outputs must be open.

For more information about the PLL circuit, refer to the LSI Logic *LCB500K Preliminary Design Manual*.

Figure 11
MR4010 PLL Circuit
Diagram



www.DataSheet4U.com

www.DataSheet4U.com

6 System Configuration

MR4010 has a number of features that allow you to modify the system configuration. This section describes the Configuration and Cache Control (CCC) Register, which is part of the CW4010 core, and several Lbus registers, which are part of the SCLC module. You can also configure the DRAM, as described in “DRAM Modes and Programmable Configurations” on page 59.

6.1 CW4010 CCC Register

The Configuration and Cache Control (CCC) Register is part of the CW4010 coprocessor, CP0. It allows you to use software to configure various pieces of the core design, such as the BIU, the TLB, and the controllers for the Icache and Dcache.

You can read from the CCC Register or write to it using the MFC0 and MTC0 instructions described in Table 2, which starts on page 44. The register’s address in CP0 is ‘16.’ Figure 12 shows the bit configuration of the CCC Register. All bits are initialized to 0 at reset, so that the caches are not available until the register is programmed.

Figure 12
CW4010 CCC Register

31 28	27	26	25	24	23	22	21	20	19	18	17	16
R	SDB	IR1	EVI	CMP	IIE	DIE	MUL	MAD	TMR	BGE	IE0	IE1

15 14	13	12	11 10	9	8 7	6	5	4	3	2	1	0
IS[1:0]	DE0	DE1	DS[1:0]	IPWE	IPWS[1:0]	TE	WB	SR0	SR1	ISC	TAG	INV

MD96.269

R **Reserved** **[31:28]**
This field is reserved. The bits are cleared to 0.

SDB **Scan Debug** **27**
This bit enables the Scan Debug mode. Setting the bit to 1 enables the mode and clearing it to 0 disables the mode.

www.DataSheet4U.com

www.DataSheet4U.com

IR1 **Icache Scratchpad RAM** **26**

This bit enables Set-1 of the Icache to be used as scratchpad RAM. Setting the bit to 1 enables scratchpad RAM mode. Clearing it to 0 disables scratchpad RAM mode.

EVI **External Vectored Interrupt** **25**

This bit enables and disables external vectored interrupt. Setting the bit to 1 enables the interrupt and clearing it to 0 disables the interrupt.

CMP **R3000 Compatibility** **24**

This bit enables and disables R3000 Compatibility mode. Setting the bit to 1 enables the mode and clearing it to 0 disables the mode.

IIE **Icache Invalidate Enable** **23**

This bit enables and disables the Icache invalidate request. Setting the bit to 1 enables the request and clearing it to 0 disables the request.

DIE **Dcache Invalidate Enable** **22**

This bit enables and disables the Dcache invalidate request. Setting the bit to 1 enables the request and clearing the bit to 0 disables the request.

MUL **Multiplier** **21**

This bit enables and disables the hardware multiplier. Setting the bit to 1 enables the multiplier and clearing the bit to 0 disables the multiplier.

MAD **Multiply Accumulate** **20**

This bit allows the multiplier to support accumulate extensions. Setting the bit to 1 enables the feature and clearing the bit disables the feature. When this bit is set, MUL must also be set.

TMR **Timer** **19**

Setting this bit to 1 enables the timer facility associated with the CW4010 core's Count and Compare Registers. When this bit is set, and the value of the Count Register equals the value of the Compare Register, interrupt bit IP7 in the Cause Register is set. IP7 causes an interrupt in the next execution cycle, provided that interrupts are enabled by setting the Interrupt Enable bit in the Mode

www.DataSheet4U.com

www.DataSheet4U.com

Register to 1 and clearing the Error Level and Exception Level bits in the Mode Register to 0.

BGE **BIU Bus Enable Grant** **18**

This bit enables and disables the BIU bus grant. Setting this bit to 1 enables the external bus master. Clearing it to 0 allows the CW4010 core to ignore the external bus master.

IE0 **Icache Set-0 Enable** **17**

This bit enables and disables Set-0 of the Icache. Setting the bit to 1 enables Set-0 and clearing it to 0 disables Set-0.

IE1 **Icache Set-1 Enable** **16**

This bit enables and disables Set-1 of the Icache. Setting the bit to 1 enables Set-1 and clearing it to 0 disables Set-1.

IS[1:0] **Icache Size** **[15:14]**

The IS[1:0] field determines the size of each Icache set. The field settings are defined as follows:

IS1	IS0	Cache Set Size
0	0	1 Kbyte
0	1	2 Kbyte
1	0	4 Kbyte
1	1	8 Kbyte

DE0 **Dcache Set-0 Enable** **13**

This bit enables and disables Set-0 of the Dcache. Setting the bit to 1 enables Set-0 and clearing it to 0 disables Set-0.

DE1 **Dcache Set-1 Enable** **12**

This bit enables and disables Set-1 of the Dcache. Setting the bit to 1 enables Set-1 and clearing it to 0 disables Set-1.

www.DataSheet4U.com

www.DataSheet4U.com

DS[1:0] **Dcache Size** **[15:14]**

The DS[1:0] field determines the size of each Dcache set. The field settings are defined as follows:

DS1	DS0	Cache Set Size
0	0	1 Kbyte
0	1	2 Kbyte
1	0	4 Kbyte
1	1	8 Kbyte

IPWE **In-Page Write Enable** **9**

This bit enables and disables in-page write operations. Setting the bit to 1 enables in-page write and clearing it to 0 disables in-page write.

IPWS[1:0] **In-Page Write Size** **[8:7]**

The IPWS[1:0] field determines the size of the Icache set. The field settings are defined as follows:

IPWS1	IPWS0	In-Page Write Size
0	0	1 Kbyte
0	1	2 Kbyte
1	0	4 Kbyte
1	1	8 Kbyte

TE **TLB Enable** **6**

This bit enables and disables the TLB. Setting the bit to 1 enables the TLB and clearing the bit to 0 disables the TLB.

WB **Writeback** **5**

This bit defines operation for addresses not mapped by the TLB. Setting the bit to 1 enables a Writeback operation and clearing it to 0 enables a Writethrough operation.

SR0 **Scratchpad RAM Mode Set-0** **4**

This bit enables and disables scratchpad RAM mode for Set-0 of the Dcache. Setting the bit to 1 enables scratchpad mode and clearing it to 0 disables scratchpad mode.

www.DataSheet4U.com

www.DataSheet4U.com

SR1 **Scratchpad RAM Mode Set-1** **3**

This bit enables and disables scratchpad RAM mode for Set-1 of the Dcache. Setting the bit to 1 enables scratchpad mode and clearing it to 0 disables scratchpad mode.

IsC **Isolate Cache** **2**

This bit enables isolate cache mode. This means that stores to the cache are not propagated to external memory. Setting the bit to 1 enables the mode and clearing it to 0 disables the mode.

TAG **Tag Test Mode** **1**

This bit enables and disables Tag Test Mode, which is used for cache maintenance. Setting the bit to 1 enables the mode, which means that load and store operations access the Tag RAMs, and sample the tag bits Tag Data, Hit, Writeback (Dcache only), and Valid. Clearing the bit to 0 disables Tag Test Mode.

INV **Invalidate Cache Mode** **0**

This bit enables and disables Cache Invalidate Mode, which is used for cache maintenance. Setting the bit to 1 enables the mode, which means that the software must invalidate all lines in the Icache and the Dcache. After reset, zeros must be written into all Tags for both sets of the Icache and Dcache. Clearing the bit to 0 disables Invalidate Cache Mode.

6.2 Lbus Controller Registers

The Lbus controller has three 32-bit registers that store information about SCbus errors and interrupts. They are the SCbus Error Address Register, the SCbus Error Status Register, and the External Vectored Interrupt Register: You must access these registers through *kseg1*. Access to an unused address causes an SCbus timeout error. Table 1 shows the phys-

www.DataSheet4U.com

www.DataSheet4U.com

ical and virtual addresses of these registers. “SCbus Timeout Watchdog Timer” on page 85 provides further information on this subject.

Table 1
Bus Error Internal Registers

Name	Virtual Address	Physical Address	Function
SCbus Error Address Register	0x B010 0000	0x 1010 0000	Contains the SCbus error address
SCbus Error Status Register	0x B010 0004	0x 1010 0004	Contains the error status information for the SCbus
External Vectored Interrupt Register	0x B010 0008	0x 1010 0008	Contains the vectored interrupt for the SCbus

7 MR4010 Memory Map

Figure 13 shows the memory map of the MR4010 where the MR4010 is bus master, and where an Lbus device is bus master and the MR4010 is a slave. In both cases, address spaces are linear 4-Gbyte spaces. Lbus master devices cannot access MR4010 internal memory mapped registers.

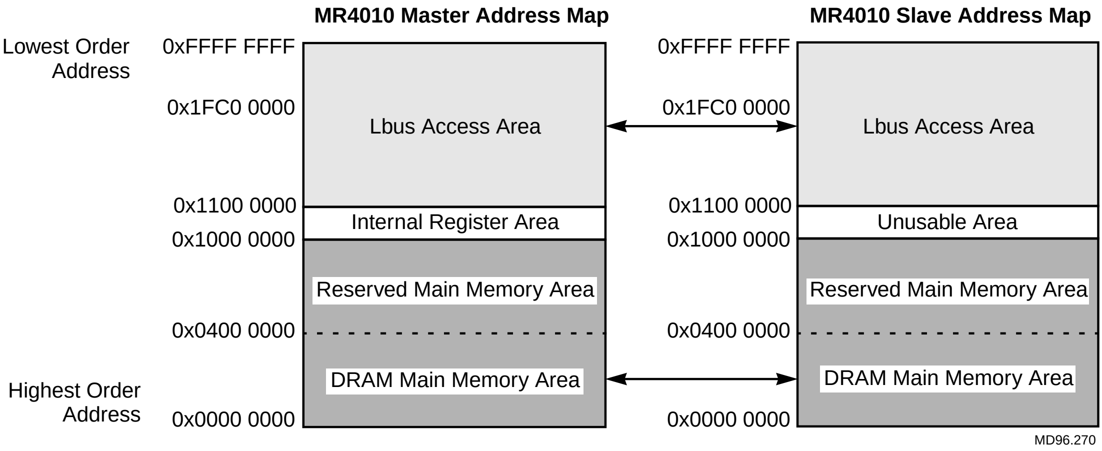
Synchronous DRAM main memory that is interfaced to the MR4010 is located at address space 0x0000 0000 through 0x03FF FFFF. The MR4010 works as an Lbus slave device for this 64-Mbyte memory space. There is no guarantee that memory devices exist in the entire 64-Mbyte area. Software, in the form of a setup/bootstrap utility or equivalent, must check installed memory size when the system is initialized. The upper 192-Mbyte space is reserved as an extended main memory area.

MR4010 internal registers for DRAM Controller and error reporting are located in the Internal Registers area between addresses 0x1000 0000 and 0x10FF FFFF. These registers must be accessed through *kseg1*, the uncached-unmapped area. The virtual address for these registers is 0xB000 0000 through 0xB0FF FFFF.

www.DataSheet4U.com

www.DataSheet4U.com

Figure 13
MR4010 Master/Slave
Memory Map



8 CW4010 Instruction Set Summary

Table 2 describes the instructions that make up the MR4010 instruction set. The chip supports 32-bit MIPS-II instructions and also implements additional extended instructions. The instructions are arranged alphabetically within the following functional groups:

- Load and store, on page 44
- Load linked, on page 45
- ALU immediate, on page 45
- ALU three-operand register type, on page 46
- Shift, on page 47
- Multiply/divide, on page 48
- CW4010 extended computational, on page 48
- Jump, on page 50
- Branch, on page 50
- Trap, on page 52
- Special, on page 52

www.DataSheet4U.com

www.DataSheet4U.com

- Coprocessor, on page 52
- CP0, on page 53
- Cache maintenance, on page 54

Table 2
MR4010 (CW4010)
Instruction Set
Summary

Instruction	Format and Description
Load and Store Instructions	
Load Byte	`LB rt, offset(base)` Sign extend the 16-bit `offset` and add to the contents of register base to form address. Sign-extend the contents of addressed byte and load into `rt`.
Load Byte Unsigned	`LBU rt, offset(base)` Sign extend the 16-bit `offset` and add to the contents of register base to form address. Zero-extend the contents of addressed byte and load into `rt`.
Load Halfword	`LH rt, offset(base)` Sign extend the 16-bit `offset` and add to the contents of register base to form address. Sign-extend the contents of addressed halfword and load into `rt`.
Load Halfword Unsigned	`LHU rt, offset(base)` Sign extend the 16-bit `offset` and add to the contents of register base to form address. Zero-extend contents of addressed halfword and load into `rt`.
Load Word	`LW rt, offset(base)` Sign extend the 16-bit `offset` and add to the contents of register base to form address, and load the addressed word into `rt`.
Load Word Left	`LWL rt, offset(base)` Sign extend the 16-bit `offset` and add to the contents of register base to form address. Shift addressed word left so that addressed byte is leftmost byte of a word. Merge bytes from memory with contents of register `rt` and load result into register `rt`.
Load Word Right	`LWR rt, offset(base)` Sign extend the 16-bit `offset` and add to the contents of register base to form address. Shift addressed word right so that addressed byte is rightmost byte of a word. Merge bytes from memory with contents of register `rt` and load result into register `rt`.
Store Byte	`SB rt, offset(base)` Sign extend the 16-bit `offset` and add to the contents of register base to form address. Store least-significant byte of register `rt` at addressed location.

(Sheet 1 of 11)

www.DataSheet4U.com

www.DataSheet4U.com

Table 2 (Cont.)
MR4010 (CW4010)
Instruction Set
Summary

Instruction	Format and Description
Load and Store Instructions (continued)	
Store Halfword	`SH rt, offset(base)` Sign extend the 16-bit `offset` and add to the contents of register base to form address. Store least-significant halfword of register `rt` at addressed location.
Store Word	`SW rt, offset(base)` Sign extend the 16-bit `offset` and add to the contents of register base to form address. Store contents of register `rt` at addressed location.
Store Word Left	`SWL rt, offset(base)` Sign extend the 16-bit `offset` and add to the contents of register base to form address. Shift contents of register `rt` left so that the leftmost byte of the word is in the position of the addressed byte. Store word containing shifted bytes into word at addressed byte.
Store Word Right	`SWR rt, offset(base)` Sign extend the 16-bit `offset` and add to the contents of register base to form address. Shift contents of register `rt` right so that the rightmost byte of the word is in the position of the addressed byte. Store word containing shifted bytes into word at addressed byte.
Load Linked Instructions	
Load Linked	`LL rt, offset(base)` Sign extend the 16-bit `offset` and add to the contents of the register base to form the address. Load the addressed word into register `rt`.
Store Conditional	`SC rt, offset(base)` Sign extend the 16-bit `offset` and add to the contents of the register base to form the address. Conditionally store register `rt` at the address, based on whether the load-link has been “broken.”
Synchronize	`SYNC` Complete all outstanding load and store instructions before allowing any new load or store instruction to start.
ALU Immediate Instructions	
Add Immediate	`ADDI rt, rs, immediate` Add 16-bit, sign-extended `immediate` to register `rs` and place 32-bit result in register `rt`. Trap on two’s complement overflow.
Add Immediate Unsigned	`ADDIU rt, rs, immediate` Add 16-bit, sign-extended `immediate` to register `rs` and place 32-bit result in register `rt`. Do not trap on overflow.

(Sheet 2 of 11)

www.DataSheet4U.com

www.DataSheet4U.com

Table 2 (Cont.)
MR4010 (CW4010)
Instruction Set
Summary

Instruction	Format and Description
ALU Immediate Instructions (continued)	
Set on Less than Immediate	`SLTI rt, rs, immediate` Compare 16-bit, sign-extended `immediate` with register `rs` as signed 32-bit integers. Result = 1 if `rs` is less than immediate; otherwise result = 0. Place result in register `rt`.
Set on Less than Immediate Unsigned	`SLTIU rt, rs, immediate` Compare 16-bit, sign-extended `immediate` with register `rs` as unsigned 32-bit integers. Result = 1 if `rs` is less than `immediate`; otherwise result = 0. Place result in register `rt`.
AND Immediate	`ANDI rt, rs, immediate` Zero-extend 16-bit `immediate`, AND with contents of register `rs`, and place result in register `rt`.
OR Immediate	`ORI rt, rs, immediate` Zero-extend 16-bit `immediate`, OR with contents of register `rs`, and place result in register `rt`.
Exclusive OR Immediate	`XORI rt, rs, immediate` Zero-extend 16-bit `immediate`, exclusive OR with contents of register `rs`, and place result in register `rt`.
Load Upper Immediate	`LUI rt, immediate` Shift 16-bit `immediate` left 16 bits. Set least-significant 16 bits of word to zeros. Store result in register `rt`.
ALU Three-Operand Register Type Instructions	
Add	`ADD rd, rs, rt` Add contents of registers `rs` and `rt` and place 32-bit result in register `rd`. Trap on two's complement overflow.
Add Unsigned	`ADDU rd, rs, rt` Add contents of registers `rs` and `rt` and place 32-bit result in register `rd`. Do not trap on overflow.
Subtract	`SUB rd, rs, rt` Subtract contents of registers `rt` from `rs` and place 32-bit result in register `rd`. Trap on two's complement overflow.
Subtract Unsigned	`SUBU rd, rs, rt` Subtract contents of register `rt` from `rs` and place 32-bit result in register `rd`. Do not trap on overflow.
Set on Less than	`SLT rd, rs, rt` Compare contents of registers `rt` and `rs` (as signed, 32-bit integers). If register `rs` is less than `rt`, `rd` = 1; otherwise, `rd` = 0.

(Sheet 3 of 11)

www.DataSheet4U.com

www.DataSheet4U.com

Table 2 (Cont.)
MR4010 (CW4010)
Instruction Set
Summary

Instruction	Format and Description
ALU Three-Operand Register Type Instructions (continued)	
Set on Less than Unsigned	SLTU rd, rs, rt Compare contents of registers rt and rs (as unsigned, 32-bit integers). If register rs is less than rt, rd = 1; otherwise, rd = 0.
AND	AND rd, rs, rt Bitwise AND contents of registers rs and rt and place result in register rd.
OR	OR rd, rs, rt Bitwise OR contents of registers rs and rt and place result in register rd.
Exclusive OR	XOR rd, rs, rt Bitwise exclusive OR contents of registers rs and rt and place result in register rd.
NOR	NOR rd, rs, rt Bitwise NOR contents of registers rs and rt and place result in register rd.
Shift Instructions	
Shift Left Logical	SLL rd, rt, shamt Shift contents of register rt left by shamt bits, inserting zeros into low-order bits. Place 32-bit result in register rd.
Shift Right Logical	SRL rd, rt, shamt Shift contents of register rt right by shamt bits, inserting zeros into high-order bits. Place 32-bit result in register rd.
Shift Right Arithmetic	SRA, rd, rt, shamt Shift contents of register rt right by shamt bits, sign-extending the high-order bits. Place 32-bit result in register rd.
Shift Left Logical Variable	SLLV rd, rt, rs Shift contents of register rt left. Low-order 5 bits of register rs specify the number of bits to shift. Insert zeros into low-order bits of rt and place 32-bit result in register rd.
Shift Right Logical Variable	SRLV rd, rt, rs Shift contents of register rt right. Low-order 5 bits of register rs specify the number of bits to shift. Insert zeros into high-order bits of rt and place 32-bit result in register rd.
Shift Right Arithmetic Variable	SRAV rd, rt, rs Shift contents of register rt right. Low-order 5 bits of register rs specify the number of bits to shift. Sign-extend the high-order bits of rt and place 32-bit result in register rd.

(Sheet 4 of 11)

www.DataSheet4U.com

www.DataSheet4U.com

Table 2 (Cont.)
MR4010 (CW4010)
Instruction Set
Summary

Instruction	Format and Description
Multiply/Divide Instructions	
Multiply	MULT rs, rt Multiply contents of registers rs and rt as two's complement values. Place the 64-bit results in special registers EntryHi and EntryLo. (The EntryLo and EntryHi Registers are read/write registers that access the TLB.)
Multiply Unsigned	MULTU rs, rt Multiply contents of registers rs and rt as unsigned values. Place 64-bit results in special registers EntryHi and EntryLo.
Divide	DIV rs, rt Divide contents of registers rs by the contents of rt as two's complement values. Place the 32-bit quotient in special register EntryLo and the 32-bit remainder in EntryHi.
Divide Unsigned	DIVU rs, rt Divide contents of registers rs by the contents of rt as unsigned values. Place the 32-bit quotient in special register EntryLo and the 32-bit remainder in EntryHi.
Move from HI	MFHI rd Move contents of special register EntryHi to register rd.
Move from LO	MFLO rd Move contents of special register EntryLo to register rd.
Move to HI	MTHI rs Move contents of register rs to special register EntryHi.
Move to LO	MTLO rs Move contents of register rd to special register EntryLo.
CW4010 Extended Computational Instructions	
Add Circular Immediate	ADDCIU rt, rs, immediate The 16-bit immediate is sign extended and added to the contents of general register rs, with the result masked by the value in CP0's CMask Register according to the formula: $rt = (rs_{31...cmask} \mid\mid (rs+signextended\_imed)_{cmask-1...0})$
Find First Set Bit	FFS rd, rs Starting at the most significant bit in register rs, find the first bit which is set to 1, and return the bit number in register rd. If no bit is set, return with all bits of rd set to 1.
Find First Clear Bit	FFC rd, rs Starting at the most significant bit in register rs, find the first bit which is set to 0, and return the bit number in register rd. If no bit is set, return with all bits of rd set to 1.

(Sheet 5 of 11)

www.DataSheet4U.com

www.DataSheet4U.com

Table 2 (Cont.)
MR4010 (CW4010)
Instruction Set
Summary

Instruction	Format and Description
CW4010 Extended Computational Instructions (continued)	
Select and Shift Right	`SELSR rd, rs, rt` Using register `rs` and `rt` as a 64-bit register pair, and the contents of the CP0's Rotate Register as the shift count, shift the register pair `rs`//`rt` right the number of bits specified in the Rotate Register, and place the least significant 32-bit value in result register `rd`.
Select and Shift Left	`SELSL rd, rs, rt` Using register `rs` and `rt` as a 64-bit register pair, and the contents of the CP0's Rotate Register as the shift count, shift the register pair `rs`//`rt` left the number of bits specified in the Rotate Register, and place the most significant 32-bit value in result register `rd`.
Multiply/Add	`MADD rs, rt` Multiply contents of registers `rs` and `rt` as two's complement values. Add 64-bit results to the contents of the EntryLo Register and EntryHi Register, and place the results in EntryLo and EntryHi. (The EntryLo and EntryHi Registers are read/write registers that access the TLB.)
Multiply/Add Unsigned	`MADDU rs, rt` Multiply contents of registers `rs` and `rt` as unsigned values. Add 64-bit results to the contents of the EntryLo Register and EntryHi Register, and place the results in EntryLo and EntryHi.
Multiply/Subtract	`MSUB rs, rt` Multiply contents of registers `rs` and `rt` as two's complement values. Subtract the 64-bit results from the contents of the EntryLo Register and EntryHi Register, and place the results in EntryLo and EntryHi.
Multiply/Subtract Unsigned	`MSUBU rs, rt` Multiply contents of registers `rs` and `rt` as unsigned values. Subtract the 64-bit results from the contents of the EntryLo Register and EntryHi Register, and place the results in EntryLo and EntryHi.
Minimum	`MIN rd, rs, rt` Compare the contents of registers `rs` and `rt` as two's complement values. The smaller value is stored in register `rd`.
Maximum	`MAX rd, rs, rt` Compare the contents of registers `rs` and `rt` as two's complement values. The larger value is stored in register `rd`.

(Sheet 6 of 11)

www.DataSheet4U.com

www.DataSheet4U.com

Table 2 (Cont.)
MR4010 (CW4010)
Instruction Set
Summary

Instruction	Format and Description
Jump Instructions	
Jump	`J target` Shift 26-bit target address left two bits, combine with four high-order bits of PC, and jump to address with a one-instruction delay.
Jump and Link	`JAL target` Shift 26-bit target address left two bits, combine with four high-order bits of PC, and jump to address with a one-instruction delay. Place address of instruction following delay slot in r31 (link register).
Jump Register	`JR rs` Jump to address contained in register `rs` with a one-instruction delay.
Jump and Link Register	`JALR rs, rd` Jump to address contained in register `rs` with a one-instruction delay. Place address of instruction following delay slot in `rd`.
Branch Instructions	
Branch on Equal	`BEQ rs, rt, offset` Branch to target address if register `rs` is equal to register `rt`. (See Footnote [1] at the end of the table.)
Branch on Not Equal	`BNE rs, rt, offset` Branch to target address if register `rs` does not equal register `rt`.
Branch on Less than or Equal to Zero	`BLEZ rs, offset` Branch to target address if register `rs` is less than or equal to 0.
Branch on Greater than Zero	`BGTZ rs, offset` Branch to target address if register `rs` is greater than 0.
Branch on Less than Zero	`BLTZ rs, offset` Branch to target address if register `rs` is less than 0.
Branch on Greater than or Equal to Zero	`BGEZ rs, offset` Branch to target address if register `rs` is greater than or equal to 0.
Branch on Less than Zero and Link	`BLTZAL rs, offset` Place address of instruction following delay slot in register r31 (link register). Branch to target address if register `rs` is less than 0.
Branch on Greater than or Equal to Zero and Link	`BGEZAL rs, offset` Place address of instruction following delay slot in register r31 (link register). Branch to target address if register `rs` is greater than or equal to 0.
Branch on Equal Likely	`BEQL rs, rt, offset` Branch to target address if register `rs` is equal to register `rt`.
Branch on Not Equal Likely	`BNEL rs, rt, offset` Branch to target address if register `rs` does not equal register `rt`.

(Sheet 7 of 11)

www.DataSheet4U.com

www.DataSheet4U.com

Table 2 (Cont.)
MR4010 (CW4010)
Instruction Set
Summary

Instruction	Format and Description
Branch Instructions (continued)	
Branch on Less than or Equal to Zero Likely	`BLEZL rs, offset` Branch to target address if register `rs` is less than or equal to 0.
Branch on Greater than Zero Likely	`BGTZL rs, offset` Branch to target address if register `rs` is greater than 0.
Branch on Less than Zero Likely	`BLTZL rs, offset` Branch to target address if register `rs` is less than 0.
Branch on Greater than or Equal to Zero Likely	`BGEZL rs, offset` Branch to target address if register `rs` is greater than or equal to 0.
Branch on Less than Zero and Link Likely	`BLTZALL rs, offset` Place address of instruction following delay slot in register r31 (link register). Branch to target address if register `rs` is less than 0.
Branch on Greater than or Equal to Zero and Link Likely	`BGEZALL rs, offset` Place address of instruction following delay slot in register r31 (link register). Branch to target address if register `rs` is greater than or equal to 0.
Trap Instructions	
Trap on Equal	`TEQ rs, rt` Trap if register `rs` is equal to register `rt`.
Trap on Equal Immediate	`TEQI rs, immediate` Trap if register `rs` is equal to the immediate value.
Trap on Greater than or Equal	`TGE rs, rt` Trap if register `rs` is greater than or equal to register `rt`.
Trap on Greater than or Equal Immediate	`TGEI rs, immediate` Trap if register `rs` is greater than or equal to the immediate value.
Trap on Greater than or Equal Unsigned	`TGEU rs, rt` Trap if register `rs` is greater than or equal to register `rt`.
Trap on Greater than or Equal Immediate Unsigned	`TGEIU rs, immediate` Trap if register `rs` is greater than or equal to the immediate value.
Trap on Less than	`TLT rs, rt` Trap if register `rs` is less than register `rt`.
Trap on Less than Immediate	`TLTI rs, immediate` Trap if register `rs` is less than the immediate value.
Trap on Less than Unsigned	`TLTU rs, rt` Trap if register `rs` is less than register `rt`.

(Sheet 8 of 11)

www.DataSheet4U.com

www.DataSheet4U.com

Table 2 (Cont.)
MR4010 (CW4010)
Instruction Set
Summary

Instruction	Format and Description
Trap Instructions (continued)	
Trap on Less than Immediate Unsigned	`TLTIU rs, immediate` Trap if register `rs` is less than the immediate value.
Trap if Not Equal	`TNE rs, rt` Trap if register `rs` is not equal to `rt`.
Trap if Not Equal Immediate	`TNEI rs, immediate` Trap if register `rs` is not equal to the immediate value.
Special Instructions	
System Call	SYSCALL Initiate system call trap, immediately transferring control to exception handler.
Breakpoint	BREAK Initiate breakpoint trap, immediately transferring control to exception handler.
Coprocessor Instructions	
Load Word to Coprocessor	`LWCz rt, offset(base)` Extend the sign of the 16-bit `offset` and add the `offset` to the contents of the general register base to form a 32-bit unsigned effective address. The word at the memory location specified is loaded into coprocessor register `rt` of the coprocessor unit `z`.
Store Word from Coprocessor	`SWCz rt, offset(base)` Extend the sign of the 16-bit `offset` and add the `offset` to the contents of the general register base to form a 32-bit unsigned effective address. The contents of coprocessor register `rt` of the coprocessor unit `z` are stored at the address specified by the 32-bit unsigned effective address.
Move to Coprocessor	`MTCz rt, rd` Load the contents of general register `rt` into the `rd` register of coprocessor unit `z`.
Move from Coprocessor	`MFCz rt, rd` Load the contents of the `rd` register of coprocessor unit `z` into general register `rt`.
Move Control to Coprocessor	`CTCz rt, rd` Load the contents of general register `rt` into the control register `rd` of coprocessor unit `z`.
Move Control from Coprocessor	`CFCz rt, rd` Load the contents of the control register `rd` of coprocessor unit `z` into general register `rt`.

(Sheet 9 of 11)

www.DataSheet4U.com

www.DataSheet4U.com

Table 2 (Cont.)
MR4010 (CW4010)
Instruction Set
Summary

Instruction	Format and Description
Coprocessor Instructions (continued)	
Coprocessor Operation	`COPz cofun` Initiate a coprocessor operation that may specify and reference the coprocessor's internal registers or change the state of the coprocessor's condition line, but does not change the state within the processor or the cache memory.
Branch on Coprocessor z True (Likely)	`BCzT offset, (BCzTL offset)` Compute a branch target address by adding address of instruction to the 16-bit `offset` (shifted left two bits and sign-extended to 32 bits). Branch to the target address (with a delay of one instruction) if coprocessor z's condition line is true. In a Branch Likely, the delay slot instruction is not executed when the branch is not taken.
Branch on Coprocessor z False (Likely)	`BCzF offset, (BCzFL offset)` Compute a branch target address by adding address of instruction to the 16-bit `offset` (shifted left two bits and sign-extended to 32 bits). Branch to the target address (with a delay of one instruction) if coprocessor z's condition line is false. In a Branch Likely, the delay slot instruction is not executed when the branch is not taken.
CP0 Instructions	
Move to CP0	`MTC0 rt, rd` Load contents of CPU register `rt` into CP0 register `rd`.
Move from CP0	`MFC0 rt, rd` Load contents of CP0 register `rd` into CPU register `rt`.
Read Indexed TLB Entry	`TLBR` Load the EntryHi and EntryLo registers with the TLB entry pointed to by the Index register.
Write Indexed TLB Entry	`TLBWI` Load TLB entry pointed to by the Index register with the contents of the EntryHi and EntryLo registers.
Write Random TLB Entry	`TLBWR` Load TLB entry pointed to by the Random register with the contents of the EntryHi and EntryLo registers.
Probe TLB for Matching Entry	`TLBP` Load the Index register with the address of the TLB entry whose contents match the EntryHi and EntryLo registers. If no TLB entry matches, set the high-order bit of the Index register.

(Sheet 10 of 11)

www.DataSheet4U.com

www.DataSheet4U.com

Table 2 (Cont.)
MR4010 (CW4010)
Instruction Set
Summary

Instruction	Format and Description
CP0 Instructions (continued)	
Exception Return[2]	`ERET (R4000 Mode)` Load the PC from ErrorEPC(SR2=1:Error Exception) or EPC(SR2=0:Exception) and clear ERL bit (SR2=1) or EXL bit (SR2=0) in the Status Register. SR2 is Status register bit[2].
Restore From Exception[2]	`RFE (R3000 Mode)` Restore previous interrupt mask and mode bits of the Status register into current status bits. Restore old status bits into previous status bits.
Wait for Interrupt	`WAITI` Stop execution of instructions and places the processor into a power save (stall) condition until a hardware interrupt, NMI (nonmaskable interrupt), or reset is received.
Cache Maintenance Instructions	
Flush Icache	`FLUSHI` Flush Icache. 256 stall cycles will be needed.
Flush Dcache	`FLUSHD` Flush Dcache. 256 stall cycles will be needed.
Flush Icache & Dcache	`FLUSHID` Flush both Icache and Dcache in 256 stall cycles.
Writeback	`WB offset(base)` Write back a Dcache line addressed by offset+GPR[base]. This instruction applies to both Dcache sets.
(Sheet 11 of 11)	

1. All branch-instruction target addresses are computed as follows: add the address of instruction in the delay slot and the 16-bit offset (shifted left two bits and sign-extended to 32 bits). All branches occur with a delay of one instruction.
2. These two instructions cannot both be legal at the same time. The one that is not legal causes a reserved instruction exception.

www.DataSheet4U.com

www.DataSheet4U.com

9 DRAM Controller and Memory Bus

This section describes the synchronous DRAM Controller and the memory bus. It defines:

- DRAM types compatible with the MR4010
- Address space available for the DRAM
- Memory interface
- Memory address bit assignment
- Programmable features of the DRAM, including the DRAM Mode Register and DRAM Controller Configuration Register
- DRAM refresh requirements, and the DRAM Controller Refresh Register and Refresh counter
- DRAM commands
- Initializing the DRAM
- Timing requirements for the different DRAM transactions

9.1 DRAM Types and Available DRAM Address Area

The MR4010 interfaces directly to synchronous DRAMs without any glue logic through a 64-bit memory data bus. When the DRAM is arranged in two banks, the chip select signals MCSn[1:0] select between the two banks as described on page 26.

www.DataSheet4U.com

www.DataSheet4U.com

Table 3 shows different DRAM configurations and the address ranges assigned to the memory banks. There is no programmable feature that defines the DRAM size and configuration. The utility setup/bootstrap program should check the amount of installed DRAM when the system is initially powered up.

Table 3
DRAM
Configurations

DRAM Type	Number of Banks	Number of DRAMs	Memory Size	CS0 Bank Area Address Range	CS1 Bank Area Address Range
1 Mbyte x 16	1	4	8 Mbyte	0x0000 0000 – 0x007F FFFF	None
1 Mbyte x 16	2	8	16 Mbyte	0x0000 0000 – 0x007F FFFF	0x0200 0000 – 0x027F FFFF
2 Mbyte x 8	1	8	16 Mbyte	0x0000 0000 – 0x00FF FFFF	None
2 Mbyte x 8	2	16	32 Mbyte	0x0000 0000 – 0x00FF FFFF	0x0200 0000 – 0x02FF FFFF
4 Mbyte x 4	1	16	32 Mbyte	0x0000 0000 – 0x01FF FFFF	None
4 Mbyte x 4	2	32	64 Mbyte	0x0000 0000 – 0x01FF FFFF	0x0200 0000 – 0x03FF FFFF

9.2 Memory Interface

Figure 14 shows the interface between MR4010 Mbus and the DRAMs. In the example shown, eight 16-bit DRAM devices are arranged in two memory banks to provide 16 Mbytes of memory. This is the configuration shown in line 2 of Table 3. As shown in Table 3, this configuration does not have continuous memory space.

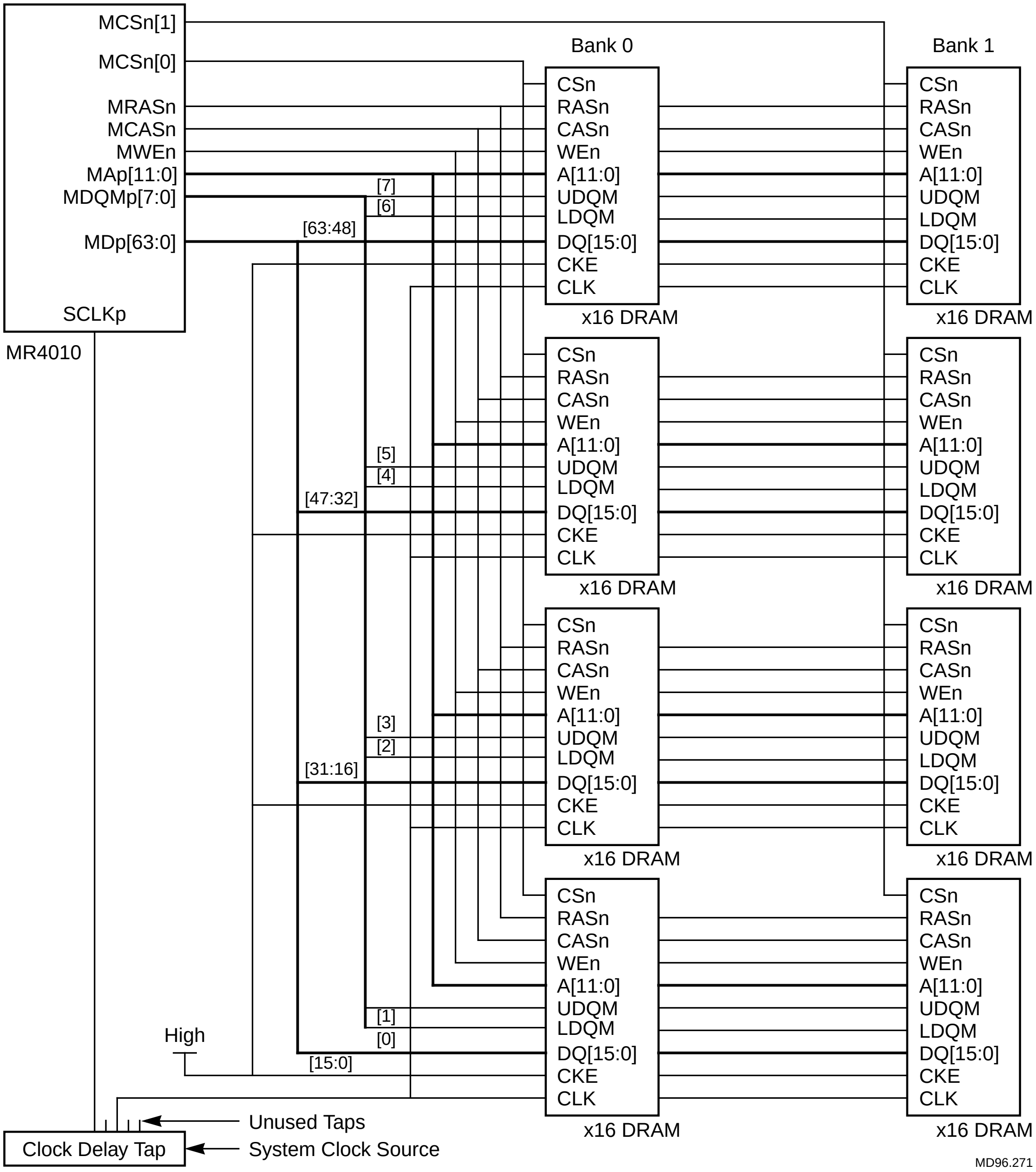
A clock delay tap provides the clock input for the DRAMs. The clock enable (CKE) inputs to the DRAMs are tied HIGH, which means that they are always asserted.

The MR4010 selects between Bank 0 and Bank 1 of the DRAM by means of the chip select signals, MCSn[1:0]. It asserts MCSn0 to select the four DRAMs in Bank 0, and MCSn1 to select the four DRAMs in Bank 1. The MR4010 distributes address (MAp[11:0]), row address strobe and column address strobe (MRAS and MCAS) and the write enable signal (MWEn) to all DRAMs. Data (MDp[63:0]) and the data mask (MDQMp[7:0]) are distributed to each byte in the DRAM array, with MDQMp7 masking byte 7 (bits [63:56]), and so forth.

www.DataSheet4U.com

www.DataSheet4U.com

Figure 14
MR4010 Interface with DRAM



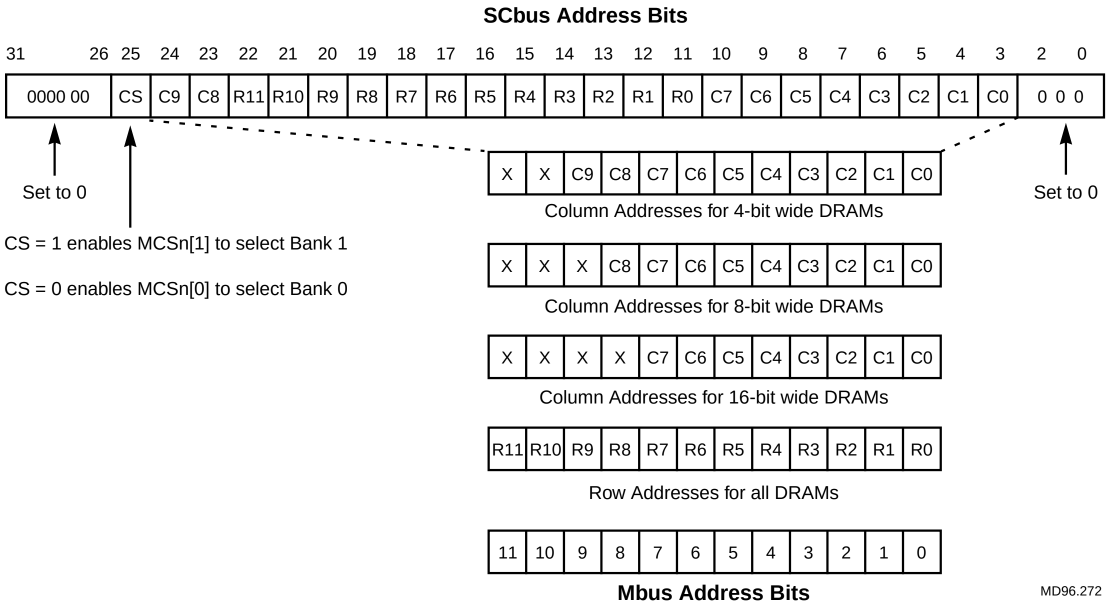
www.DataSheet4U.com

www.DataSheet4U.com

9.3 Address Bit Assignment

The DRAM Controller in the MR4010 derives the DRAM addresses, MAp[11:0], from the 32 SCbus address bits output by the bus master, which may be the CW4010 core or the SCLC module. The controller outputs the address bits on the Mbus, assigning the bits as shown in Figure 15. The byte select signals, MDQMp[7:0], are derived directly from the byte enable signals, SCTBEn[7:0].

Figure 15
SCbus DRAM Address
Bit Assignment


SCbus Address Bits
31 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 0
0000 00 | CS | C9 | C8 | R11 | R10 | R9 | R8 | R7 | R6 | R5 | R4 | R3 | R2 | R1 | R0 | C7 | C6 | C5 | C4 | C3 | C2 | C1 | C0 | 0 0 0
Set to 0
Set to 0
CS = 1 enables MCSn[1] to select Bank 1
CS = 0 enables MCSn[0] to select Bank 0
X | X | C9 | C8 | C7 | C6 | C5 | C4 | C3 | C2 | C1 | C0
Column Addresses for 4-bit wide DRAMs
X | X | X | C8 | C7 | C6 | C5 | C4 | C3 | C2 | C1 | C0
Column Addresses for 8-bit wide DRAMs
X | X | X | X | C7 | C6 | C5 | C4 | C3 | C2 | C1 | C0
Column Addresses for 16-bit wide DRAMs
R11 | R10 | R9 | R8 | R7 | R6 | R5 | R4 | R3 | R2 | R1 | R0
Row Addresses for all DRAMs
11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0
Mbus Address Bits
MD96.272

www.DataSheet4U.com

www.DataSheet4U.com

The MAp[11:0] 12-bit address bus multiplexes row and column addresses. Table 4 lists the SCbus address and Mbus address bit assignments.

Table 4
SCbus Address and Mbus Address Bit Assignment

SCbus Address Bits	Address Bit Function	Mbus Address Bits
SCTBEn[7:0]	Byte selection during write operations	MDQMp[7:0]
SCAp[10:3][1]	Column addresses C[7:0]	MAp[7:0]
SCAp[22:12]	Row addresses R[11:0]	MAp[11:0]
SCAp[24:23]	Column addresses C[9:8]	MAp[9:8]
SCAp[25]	Chip selection	MCSn[1:0]
SCAp[31:26][2]	Not used	Not used

[1] These bits are used for column addresses in 16-bit wide DRAMs. They are used in conjunction with SC[23] for 8-bit wide DRAMs, and with SC[24:23] for 4-bit wide DRAMs. Since SC[10:3] supply column addresses, the DRAM page size is 1 Kbyte for all DRAMs.

[2] To access the DRAM, SC[31:26] must be set to 0. Otherwise, the DRAM controller will not respond to an SCbus transaction.

9.4 DRAM Modes and Programmable Configurations

The MR4010 DRAM Controller supports a number of programmable modes. Each DRAM has a Mode Register and the DRAM Controller has a Configuration Register. You can select different modes by setting the appropriate bits in the two registers. The modes you can program are:

- Cache Writethrough and Writeback

  Cache Writethrough mode allows all data that is updated in the cache to be updated at the same time in external memory. In Cache Writeback mode, main memory is only updated when the cache line is reallocated or is explicitly flushed.

- Burst Length

  Defines the number of words to be output or input during read and write cycles. In the MR4010, the Burst Length field is set to 0.

- Wrap Type

  Specifies the order in which burst data is addressed. This mode does not apply in the MR4010, since the Burst Length field is set to 0.

www.DataSheet4U.com

www.DataSheet4U.com

- CAS Latency

  Defines the number of clock cycles that must occur before data is available.

- Auto Precharge

  This mode is not used in the MR4010.

9.4.1 DRAM Mode Register

Figure 16 shows the format of the 12-bit DRAM Mode Register. This register is programmed by a mode write command.

Figure 16
DRAM Mode Register
Format

11 10	9	8 7	6 4	3	2 0
R	WB	R	CL	WT	BL

MD96.273

When you power up the DRAM, the boot program precharges the DRAM devices. You should refer to the documentation supplied with the DRAM for further information on precharging. After precharge, you must program the Mode Register using the following procedure.

Step 1. Before you program the Mode Registers, set the MRS bit in the DRAM Controller Configuration Register (see page 62).

Step 2. Program the timing correctly.

Step 3. Initiate a Mode Register write cycle by executing a write transaction to one of the following addresses. The data stored should be all zeros.

CAS Latency Setting	Physical Address	Virtual Address
1	0x 0010 8000	0x A010 8000
2	0x 0011 0000	0x A011 0000
3	0x 0011 8000	0x A011 8000

R **Reserved** **[11:10] [8:7]**
These bits are not used and are set to 0.

www.DataSheet4U.com

www.DataSheet

WB **Cache Writethrough and Writeback** **9**

You can select Cache Writethrough mode by setting this bit to 1. Setting the bit to 0 selects Writeback mode. However, there is no significance to Writethrough and Writeback modes in MR4010 transactions, since the burst length is one word.

CL **CAS Latency** **[6:4]**

You can select among one-, two-, and three-cycle modes by programming bits [6:4] in the DRAM Mode Register. You should select one of the following settings (all other combinations of the bit settings are reserved):

Bit 6	Bit 5	Bit 4	Cache Latency Modes[1]
0	0	1	1
0	1	0	2
0	1	1	3

[1] You must set the same Cache Latency mode in the DRAM Controller Configuration Register.

Bits [21:20] in the DRAM Controller Configuration Register also select the cycle modes. You must set or clear the bits in both registers to select the required mode.

WT **Wrap Type** **3**

This mode is not used in the MR4010, so you should clear this bit to 0. Setting the bit to 1 enables interleaved accesses; clearing it to 0 enables sequential accesses. Since burst length is one word for the MR4010, wrap type has no meaning. Sequential mode is compatible with SCbus burst ordering.

BL **Burst Length** **[2:0]**

You can select single-cycle mode by clearing bits [2:0] of this register to 0b000. The SC bus requests four doublewords as a burst block. With a data bus width of 64 bits, the MR4010 supports the request with multiple CAS accesses.

www.DataSheet4U.com

www.DataSheet4U.com

9.4.2 DRAM Controller Configuration Register

The DRAM Controller Configuration Register allows you to configure various features of the DRAM. The virtual and physical addresses for the register are shown below:

Virtual Address	Physical Address
0x B000 0000	0x 1000 0000

Figure 17 shows the format recommended for the DRAM Controller Configuration Register.

Figure 17
DRAM Controller Configuration Register Format

31	30	29	28	27 22	21 20	19 18	17 16
R	PC	MRS	REF	R	CL	R	RCD

15 12	11	10 8	7	6 4	3 2	1	0
RC	R	RAS	R	DAL	R	RP3	DPL2

MD96.274

R **Reserved** **31 [27:22] [19:18] 11 7 [3:2]**
These bits are not used and you should clear them to 0.

PC **Precharge Command** **30**
This bit enables the manual precharge command. If the CPU sets the bit to 1, the DRAM Controller generates one precharge command cycle for both banks. The CPU sets the bit at power up. Initialization clears the bit automatically.

MRS **Mode Register Set** **29**
If the CPU sets the bit to 1, the subsequent Store Word operation to the DRAM area generates a Mode Register Set command. The row address bits in the SC address bus (SCbits [22:11]) select the addressed location during this type of operation. SCbits[31:23] and SCbits [10:0], which are the mode bits, must be set to 0. The CPU

www.DataSheet4U.com

www.DataShe

clears the MRS bit when the word operation has been completed.

REF **Refresh Cycle** **28**

This bit enables the manual refresh cycle request (REF). If the CPU sets it to 1, one refresh cycle is generated for both memory banks. This bit is cleared automatically when the refresh cycle has been completed. You can also generate REF using the refresh counter, as described in "DRAM Refresh" on page 67.

CL **CAS Latency** **[21:20]**

You can set CAS Latency by programming the bits in this field. You should select one of the following settings:

Bit 21	Bit 20	Cache Latency Cycles
0	1	1
1	0	2
1	1	3

Although you can define all DRAM timing parameters independently, as described in the surrounding paragraphs, CAS Latency defines relationship between other timing parameters. Table 5 on page 67 shows the relationships between CAS Latency, DRAM frequency, and other configuration settings.

You must also set the CAS Latency bits in the DRAM Mode Register to reflect the same CAS Latency value.

RCD **Active RAS to Read/Write Command Period Cycles** **[17:16]**

You can program the bits in this field to select the number of active clock cycles for a read or write operation. You can select one of the following settings:

Bit 17	Bit 16	Active Clock Cycles
0	1	1
1	0	2
1	1	3

www.DataSheet4U.com

www.DataShee

RC **Refresh to Refresh/Active Command Period Cycles [15:12]**

This field allows you to select the number of active read/write cycles between refresh cycles. You can program these bits as follows:

Bit 15	Bit 14	Bit 13	Bit 12	Active Read/Write Cycles
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10

RAS **Active to Precharge Command Period [10:8]**

This field allows you to select the number of clock cycles that RAS should stay active until the memory has been precharged. You can program the bits as follows:

Bit 10	Bit 9	Bit 8	Active RAS Cycles
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

www.DataSheet4U.com

www.DataShee

DAL **Data In to Active/Refresh Command Period** **[6:4]**

This field allows you to select the number of active clock cycles between the time data input is valid until the refresh command is asserted. You can set the field as follows:

Bit 6	Bit 5	Bit 4	Active DAL Cycles
0	1	1	3
1	0	0	4
1	0	1	5

RP3 **Precharge to Active Command Period** **1**

This bit allows you to select the number of active clock cycles in the period between precharge and an active read or write command. If you set the bit to 1, there are three clock cycles. If you clear the bit to 0, there are two clock cycles.

DPL2 **Data In to Precharge Command Period** **0**

This bit allows you to select the number of active clock cycles in the period between the input of valid data to the assertion of the precharge command. If you set the bit to 1, there are two clock cycles. If you clear the bit to 0, there is one clock cycle.

The relationship between latency and frequency varies, depending on the DRAM specification. Table 5 shows an example of the timing parameters for three NEC DRAMS—uPD4516821, uPD4516421, and uPD4516161 DRAMs. For the fastest access time, you should use a

www.DataSheet4U.com

www.DataSheet4U.com

DRAM with a maximum clock frequency of 10 ns. Refer to the NEC user’s manual for further information.

www.DataSheet4U.com

www.DataSheet4U.com

Table 5
Relationship Between
Frequency and Latency

Clock Frequencies			Settings						
(-10)[1]	(-12)[2]	(-13)[3]	CL	RCD	RC	RAS	DAL	RP	DPL
66 MHZ	55 MHz	50 MHz	2	2	7	5	3	2	1
33 MHz	27 MHz	25 MHz	1	1	4	3	2(3)[4]	1(2)[5]	1

1. Maximum clock frequency for the 10 ns version of the DRAM.
2. Maximum clock frequency for the 12 ns version of the DRAM.
3. Maximum clock frequency for the 13 ns version of the DRAM.
4. NEC recommends 2 clock cycles for DAL when CL is set to 1. However, the MR4010 DRAM Controller requires 3 clock cycles for DAL.
5. NEC recommends 1 clock cycle for RP when CL is set to 1. However, the MR4010 DRAM controller requires 2 clock cycles for RP.

9.5 DRAM Refresh

The DRAM Controller needs to refresh the 2048 rows in the synchronous DRAM every 32 milliseconds. The controller also needs to set up a 15,625 ns (15.625 µs) refresh interval. For example, if the maximum clock frequency is 66 MHz, the controller must issue a DRAM refresh command every 1,041 clock cycles.

The DRAM Controller has an 11-bit refresh interval timer that generates the refresh command. The refresh interval timer, shown in Figure 18, consists of one 11-bit register, referred to as the Refresh Register, which stores the refresh interval time; and one 11 bit-binary count down register, referred to as the Refresh Counter, which stores the refresh counter value, and is decremented by each system clock input. The Refresh Register address is shown below:

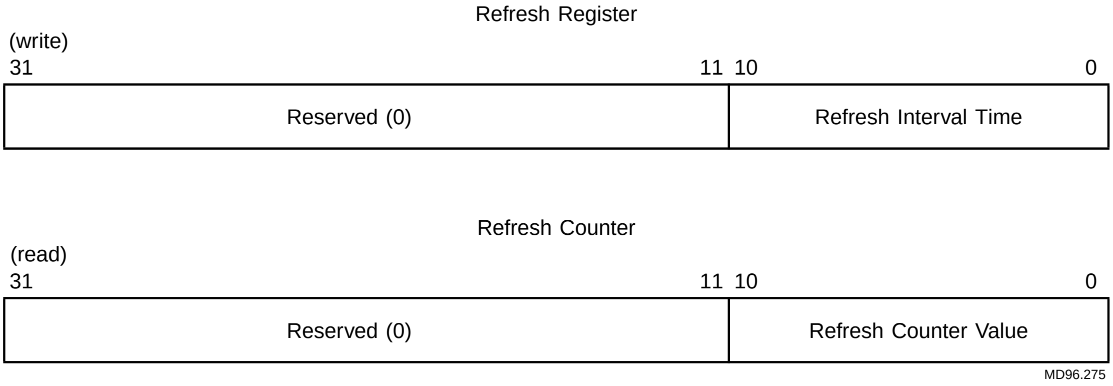
Virtual Address	Physical Address
0x B000 0004	0x 1000 0004

When the system is initialized, the DRAM Controller writes the Refresh Interval Time data into the Refresh Register. The same data is stored in the Refresh Counter as the Refresh Counter Value. The DRAM Controller reads the contents of both registers only during testing.

www.DataSheet4U.com

www.DataSheet4U.com

Figure 18
DRAM Refresh Interval
Timer



After a cold reset, the counter stops counting. Once the DRAM Controller has written the value for the Refresh Interval Time into the Refresh Register, the counter loads the same initial value and start counting by decrementing the initial value by 1 at each clock input. When the counter has counted down to 1, the DRAM Controller sets the REF bit in the MR4010 Configuration Register requesting a refresh command. The initial value is then reloaded and the process starts again. Note that the counter never counts down to 0. If a DRAM transaction is proceeding when the DRAM Controller issues the refresh command, the status of the refresh command is ‘pending,’ and a refresh command cycle is generated when the preceeding transaction has been completed. Only a cold reset can stop the refresh counter.

The setting of the Refresh Register is derived from the DRAM clock cycle value, the required refresh interval (15,625), and the CAS latency setting (CL), which determines the number of clock cycles required before data is available. Table 6 lists Refresh Register programming values for two microprocessors (66 MHz, and 50 MHz). You can calculate value A by dividing the refresh interval by the microprocessor’s clock cycle time. You can calculate the value programmed into the Refresh Register by subtracting the number of clock cycles required (a function of the CL

www.DataSheet4U.com

www.DataSheet4U.com

setting) from value A. In the first example shown, the Refresh Register should be set to 1031 (0x407).

Table 6
Refresh Register
Programming Values

Clock Frequency	Clock Cycle Time	Value (A)[1]	Number of Clock Cycles Required (B)[2]	Refresh Register Programmed Value[3] Decimal (Hex)	CL Setting
66 MHz	15 ns	1041	10	1031 (0x407)	2
50 MHz	20 ns	781	8	773 (0x305)	1

1. Value A is derived from the required refresh interval time (15,625 ns) divided by the clock cycle time (12.5 ns, and so forth).
2. Number of clock cycles required is a function of the CL setting.
3. The Refresh Register programmed value is derived from Value A minus Value B (the required number of clock cycles).

Register setting for the 80 MHz, 12.5 ns DRAM.

Register Bits	31	30	29	28	27-11	10	9	8	7	6	5	4	3	2	1	0
Binary Setting	0	0	0	0	x	1	0	0	1	1	0	1	0	1	1	0
Hex Value	0				x	4			D				6			
Decimal Value	Not used					1 2 3 8										

9.6 DRAM Commands

This section describes the DRAM commands used by MR4010 DRAM Controller. They are the chip select commands (MCSn[1:0]), row and column addresses strobes (RASn and CASn), and the write enable command (MWEn). The DRAM Controller does not use the DRAM’s Self-Refresh Entry Command and Burst Stop Command. In addition, for a No Operation (Nop), the DRAM Controller deasserts the chip select outputs MCSn[1:0] and the other control signals.

www.DataSheet4U.com

www.DataSheet4U.com

Table 7 summarizes the settings of the Mbus control signals and the DRAM commands they generate. The term *State* applies to the DRAM Controller’s internal state machine; SCAn indicates the SCbus address bit n associated with the Memory bus address bit, MApn; ~ indicates an inverted signal; ( ) indicates a don’t care condition, but one in which the signals are output.

Table 7
Summary of DRAM Commands and Mbus Control Signals

Command	**State**	**MCSn[1]**	**MCSn[0]**	**MRASn**	**MCASn**	**MWEn**	**MAp[11]**	**MAp[10]**	**MAp[9:0]**
No operation[1]	Idle	High	High	High	High	High	(SCA22)	(SCA21)	(SCA20:11)
Mode Register set	mrw[2]	Low	Low	Low	Low	Low	SCA22	SCA21	SCA[20:11]
Row active	ra[3]	~SCA25	SCA25	Low	High	High	SCA22	SCA21	SCA[20:11]
Precharge[4]	pc[5]	Low	Low	Low	High	Low	SCA22	High	SCA[20:11]
Write[6]	rwc(wr)[7]	~SCA25	SCA25	High	Low	Low	SCA22	Low	SCA24,23,[10:3]
Read[6]	rwc(rd)[8]	~SCA25	SCA25	High	Low	High	SCA22	Low	SCA24,23,[10:3]
CAS before RAS refresh	cbr[9]	Low	Low	Low	Low	High	SCA22	SCA21	SCA[20:11]

1. MCSn[1:0] must both be kept high for no-operation conditions
2. mrw = Mode Register Write
3. ra = Row Active
4. Both banks are always precharged
5. pc = Precharge
6. When Write or Read commands are sent for a burst transaction, MAp[1:0] are incremented by the order of wrap around, starting from the requested address, for example, 01, 10, 11, then 00
7. rwc(wr) = write
8. rwc(rd) = read
9. cbr = CAS before RAS

www.DataSheet4U.com

www.DataSheet4U.com

9.7 Initializing the DRAM and Programming the Mode Register

Before the DRAM Controller can access the DRAM for a normal read or write transaction, the boot program must initialize the DRAM through the DRAM Controller. After power on, the DRAM Controller goes through the following initialization process:

- precharges the DRAM
- programs the DRAM's Mode Register
- refreshes the DRAM array twice

The CPU can initiate this process by:

1. Programming the DRAM Configuration Register.
2. Programming the DRAM Mode Register by entering one of the following words at the address shown:

CAS Latency (CL)	Physical Address	Virtual Address
1	0x0010 8000	0xA010 8000
2	0x0011 0000	0xA011 0000
3	0x0011 8000	0xA011 8000

3. Programming the DRAM Refresh Register.

www.DataSheet4U.com

www.DataSheet4U.com

Figure 19 shows the timing requirements for the DRAM initialization sequence. Table 8 lists the signals referenced in Figure 19 and in subsequent timing diagrams. The signals are in alphabetical order.

www.DataSheet4U.com

www.DataSheet4U.com

Table 8
Timing Signals

Signal Name	Description	Other References
AoEREQp (internal)	Address Output Enable Request	
DCiNVSn	Dcache Invalidation Strobe	See *MiniRISC CW4010 Superscalar Microprocessor Core Technical Manual.*
DoEREQp (internal)	Data Output Enable Request	
DRAMC state	State of the DRAM Controller	
DRRDY	Data Ready	
ICiNVSn	Icache Invalidation Strobe	See *MiniRISC CW4010 Superscalar Microprocessor Core Technical Manual.*
LADSn	Lbus Address Strobe	Signals with an L prefix are Lbus signals. You will find more detailed information about these signals in "Lbus Interface" on page 28.
LAoEn	Lbus Address Enable	
LA(o)p	Lbus Address (Output from MR4010)	
LBEn	Lbus Byte Enable	
LCLKp	Lbus Clock	
LDp	Lbus Data	
LDip	Lbus Write Data	
LDop	Lbus Read Data	
LDoEn	Lbus Data Output Enable	
LHLDAp	Lbus Hold Acknowledge	
LHoLDp	Lbus Hold Request	
LRD(o)n	Lbus Data (Output to MR4010)	
LRDYn	Lbus Data Ready	
LRDYoEn	Lbus Data Ready Output Enable	
LSLRDYin (internal)	Lbus Sampled Ready	
MAp	Memory Address	Signals with an M prefix are Mbus (memory bus) signals. You will find more detailed information about these signals in "Mbus Interface" on page 26.
MCASn	Memory Column Address Strobe	
MCSn	Memory Chip Select	
MDp	Memory Data	
MDQMp	Memory Data Enable/Mask	
MRASn	Memory Row Address Strobe	
MWEn	Memory Write Enable	

(Sheet 1 of 2)

www.DataSheet4U.com

www.DataSheet4U.com

Table 8 (Cont.)
Timing Signals

Signal Name	Description	Other References
SCAp	SCbus Address	Signals with an SC prefix are CW4010 core SCbus signals. You will find more detailed information about these signals in the LSI Logic Technical Manual *MiniRISC CW4010 Superscalar Microprocessor Core.*
SCBEn (SCTBEn)	SCbus Enable	
SCBRDYn	SCbus Ready	
SCDp	SCbus Data	
SCDoEn	SCbus Data Output Enable	
SCHGTn	SCbus Hold Grant	
SCHRQn	SCbus Hold Request	
SCLKp	System Clock	
SCTBSTn	SCbus Burst Transaction	
SCTPWn	SCbus Next Transaction is in Write Page	
SCTSSn	SCbus Transaction Start Strobe	
SCTSEn	SCbus Transaction Start Enable	
SLDoEn	SCLC SCbus Data Output Enable	

(Sheet 2 of 2)

Table 9 lists abbreviations that appear in the timing diagrams

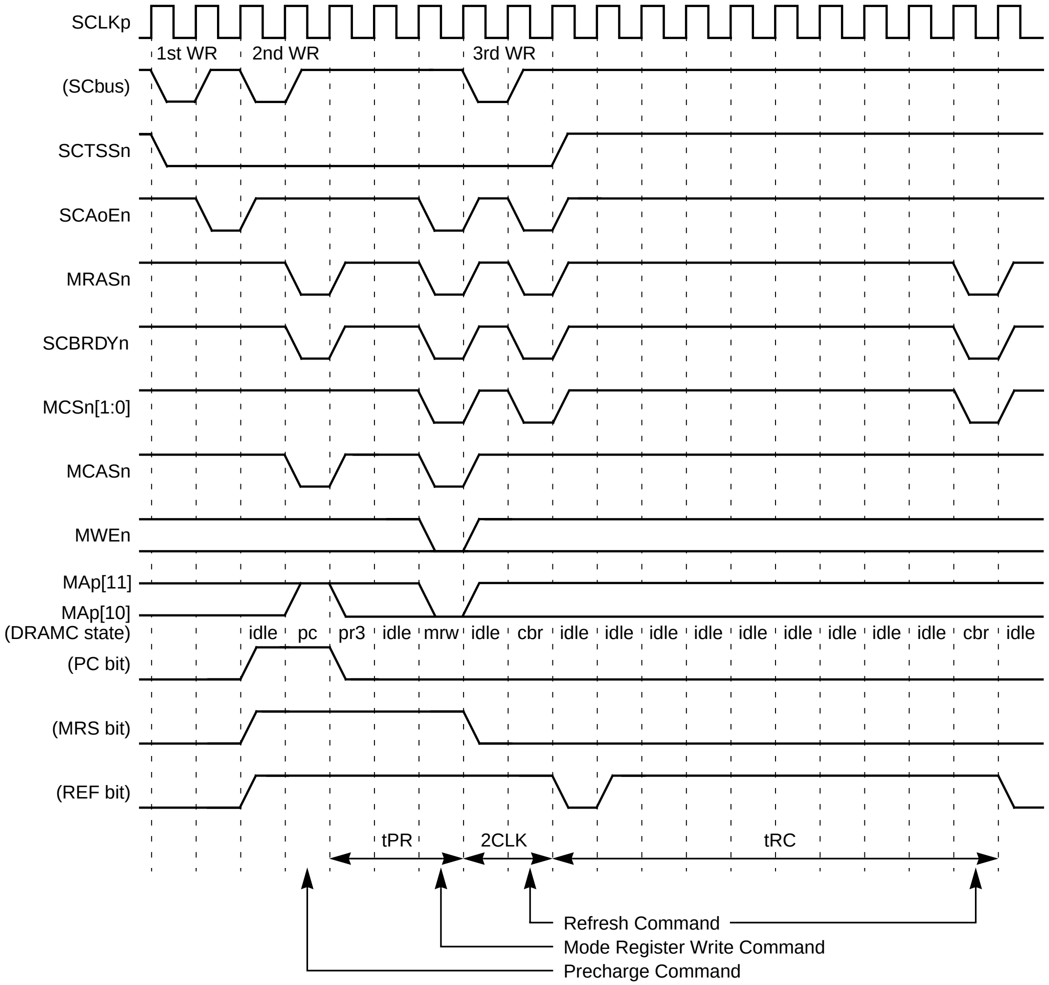
Table 9
Timing Diagram
Abbreviations

Abbreviation	Description	Other References
MRS	Mode Register Set	page 62
PC	Precharge Command	page 62
REF	Refresh Cycle	page 63

www.DataSheet4U.com

www.DataSheet4U.com

Figure 19
Timing Requirements
for the DRAM
Initialization Sequence



Notes:

- 1st WR is the write to the DRAM Configuration Register.
- 2nd WR is the write to the DRAM Mode Register.
- 3rd WR is the write to the DRAM Refresh Register.
- tPR = 3, tRC = 10.

MD96.276

www.DataSheet4U.com

www.DataSheet4U.com

9.8 DRAM Transactions

Once the DRAM Controller has initialized the DRAM, it can initiate various types of DRAM accesses. This section shows the timing requirements for three typical DRAM accesses:

- Figure 20 shows the timing for a single burst read transaction.
- Figure 21 (page 78) shows the timing for two continuous write transactions.
- Figure 22 (page 79) shows the timing for a burst write transaction.

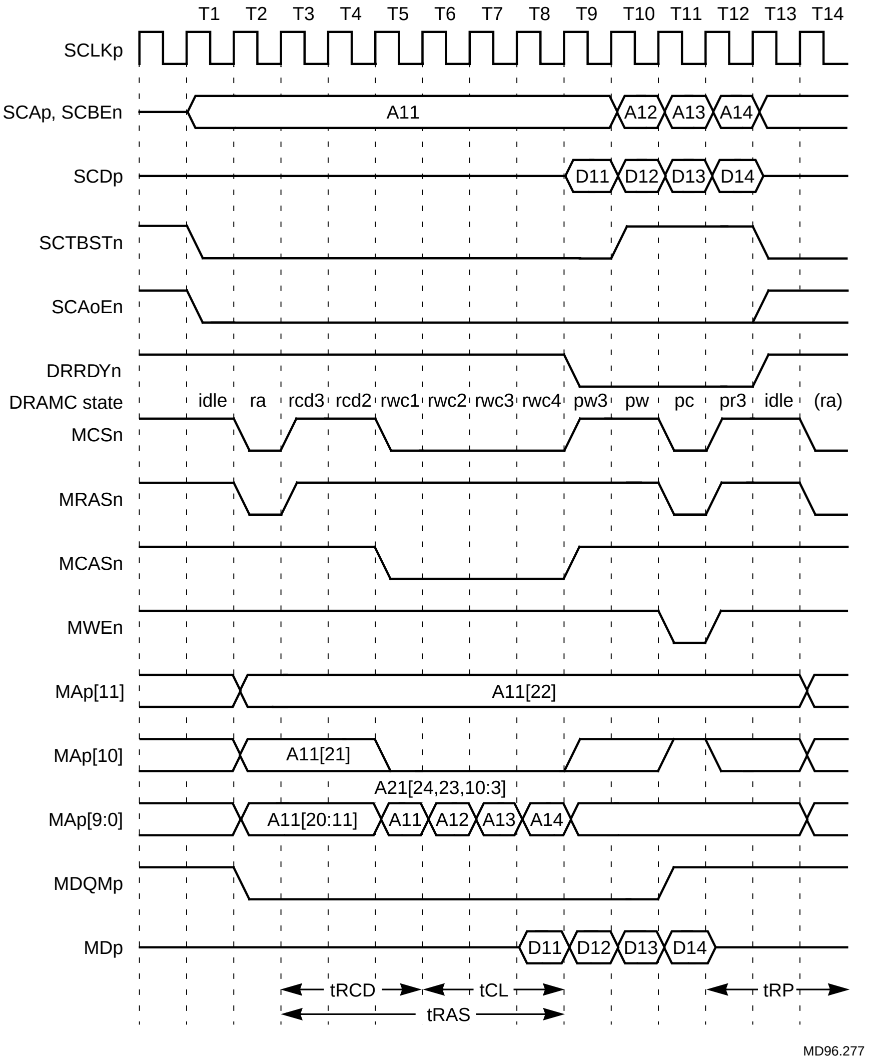
In all cases, the DRAM is an 80x, 10 ns device. Other timing parameters for this device are as follows:

- CAS Latency (CL) = 3. (Refer to page 63 for further information about CL.)
- Active RAS to Read/Write Command Period Cycles (RCD) = 3. (Refer to page 63 for further information about RCD.)
- Refresh to Refresh/Active Command Period Cycles (RC) = 8. (Refer to page 64 for further information about RC.)
- Active to Precharge Command Period (RAS) = 6. (Refer to page 64 for further information about RAS.)
- Precharge to Active Command Period (RP) = 3. (Refer to page 65 for further information about RP.)
- Data In to Precharge Command Period (DPL) = 2. (Refer to page 65 for further information about DPL.)
- Data In to Active/Refresh Command Period (DAL) = 5. (Refer to page 65 for further information about DAL.)

www.DataSheet4U.com

www.DataSheet4U.com

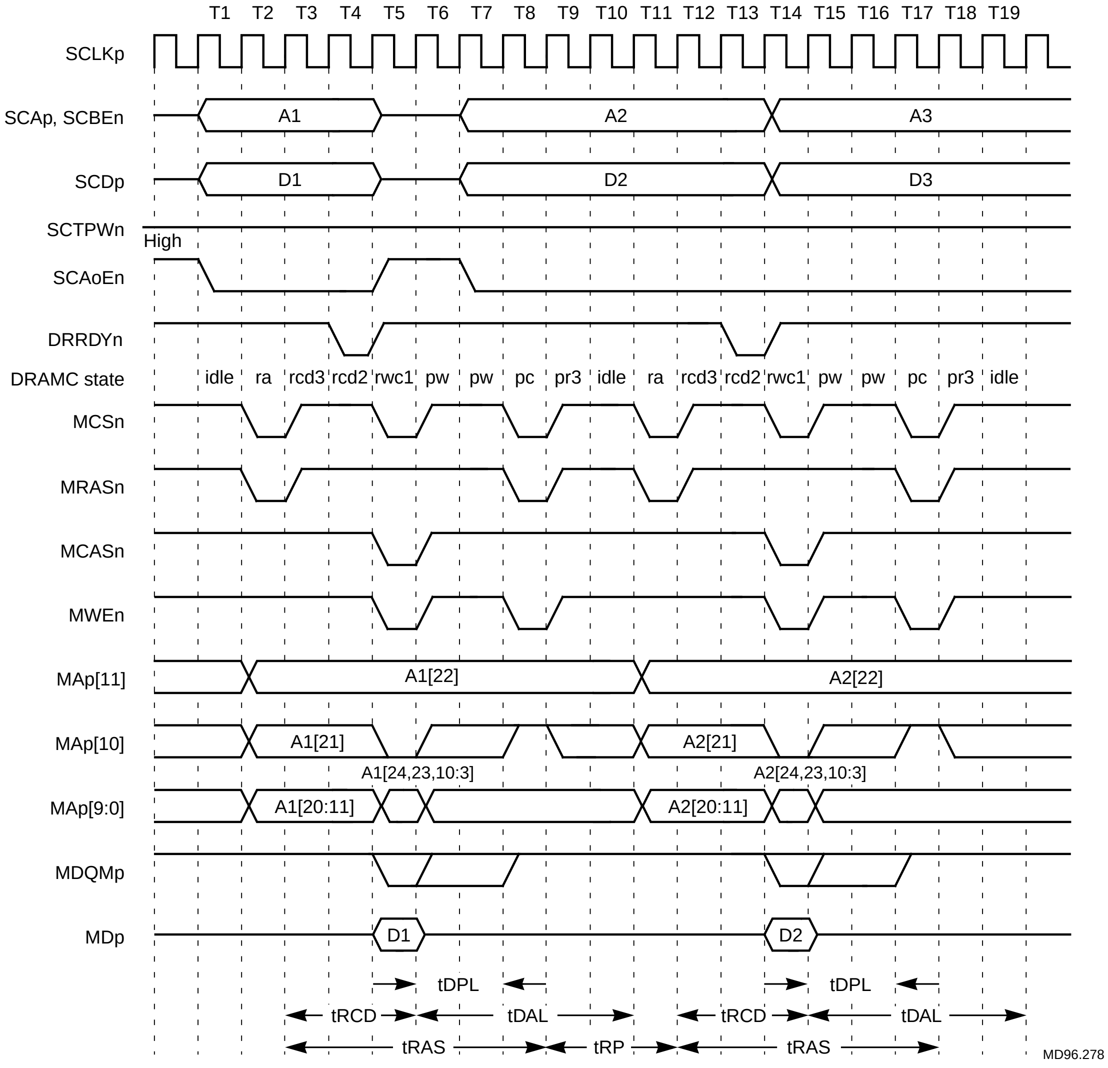
Figure 20
Single Burst Read
Transaction



www.DataSheet4U.com

www.DataSheet4U.com

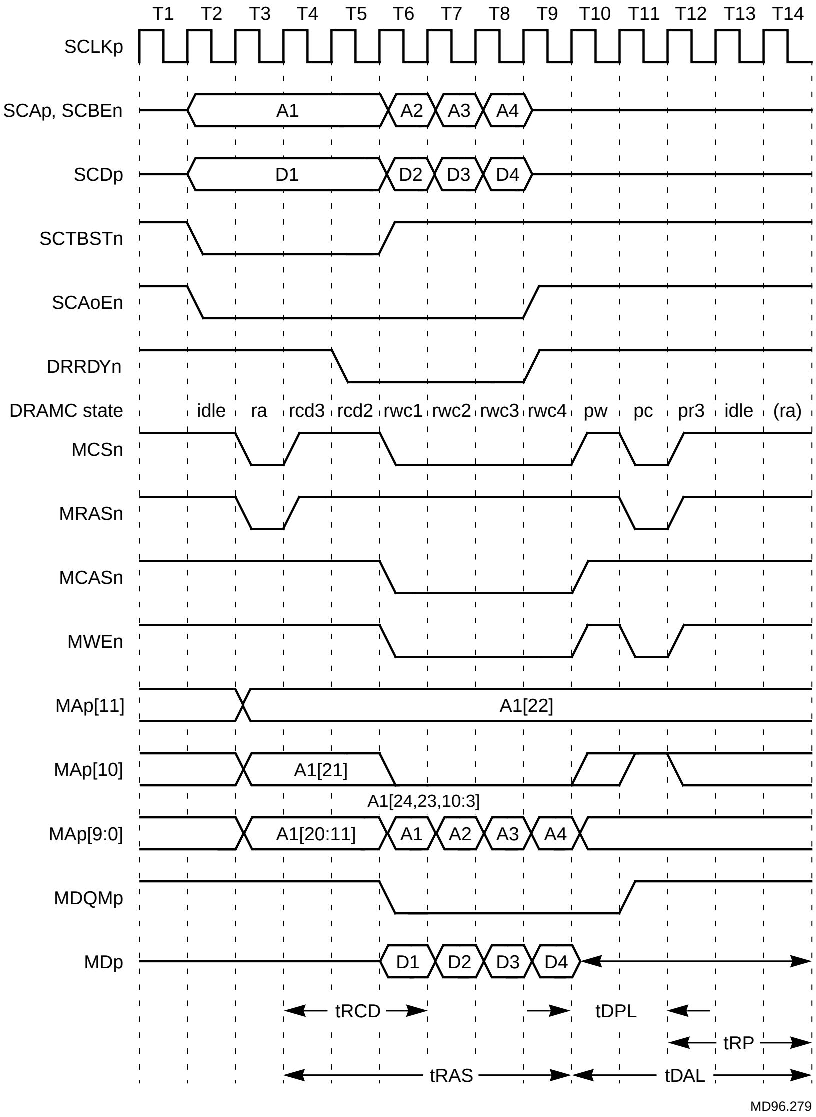
Figure 21
Two Continuous Single
Write Transactions



www.DataSheet4U.com

www.DataSheet4U.com

Figure 22
Burst Write
Transaction



www.DataSheet4U.com

www.DataSheet4U.com

10 Local I/O bus and SCbus/Lbus Converter Module

This section discusses the Lbus, and describes how the MR4010 interacts with the Lbus through the SCLC module.

10.1 Lbus Features

The Lbus is a subset of VLbus or 486 bus, which has a demultiplexed 32-bit address bus and a 32-bit data bus. “Lbus Interface” on page 28 provides a list of Lbus signals.

There are some differences between the Lbus and the VLbus. The Lbus:

- Has no I/O space
- Has no Interrupt Acknowledge Cycle
- Supports only single transactions, does not support burst transactions
- Uses the HOLD/HLDA type of bus arbitration
- Has a Bus Retry Input
- Uses the Lbus Clock, LCLKp

The Lbus is synchronized by the Lbus clock, LCLKp, which is derived from the CW4010 system clock, SCLKp. The MR4010 asserts the LCHALFn signal to divide the SCLKp by 2, and drives LCHALFn HIGH to divide the clock by 4. The MR4010 then outputs the LCLKp to the Lbus.

The MR4010 can function as the Lbus master or the Lbus slave. If the MR4010 is master, it starts an Lbus transaction while LHLDAp is deasserted. If an Lbus device wants to control the Lbus and initiate a bus transaction, it must first take ownership of the bus by issuing a bus hold request (by asserting LHoLDp) to the MR4010. The MR4010 returns a bus hold acknowledge signal (by asserting LHLDAp) to the Lbus device, granting bus ownership. When this occurs, the Lbus device may initiate Lbus transactions.

The Lbus master starts a transaction on the Lbus by asserting the Address Strobe, (LADSn). At this time, the master must also drive valid information on the address bus and the byte enable lines. The Lbus

www.DataSheet4U.com

www.DataSheet4U.com

master uses LRDn signal to control the direction of the data transfer. The master must present the appropriate level on this signal at the same time it asserts strobe signal LRDn. During a write transaction, the master must also drive valid data on the data bus at this time.

When the transaction has been successfully completed, the selected slave device asserts LRDYn, indicating that the Lbus is ready for another transaction. The master must continue to drive all signals until it samples LRDYn. If the transaction is a read, the slave device must place valid data on the bus before it asserts LRDYn.

10.2 MR4010 as Master on the Lbus

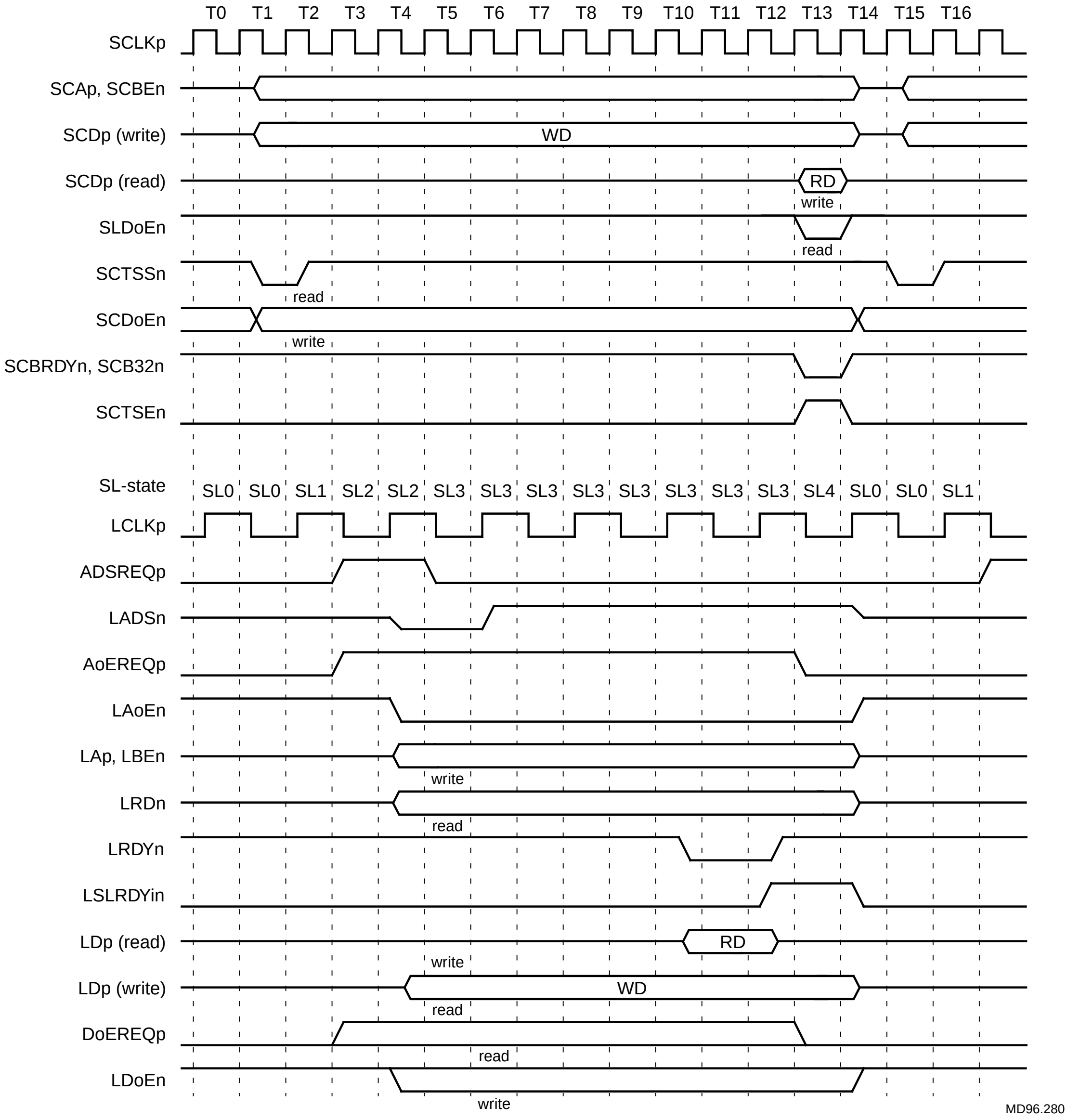
The MR4010 is the master of the Lbus when the CW4010 accesses an address in the Lbus area located in the physical address range 0x1100 0000 through 0xFFFF FFFF. The Lbus device must assert a data ready or bus retry signal and input it to the MR4010 within 256 SCLKp cycles. Otherwise, the SCbus watchdog timer terminates the SCbus transaction by asserting a bus error signal. Figure 23 shows the timing requirements for an Lbus transaction generated by the CW4010.

In the example shown, the CW4010 initiates a SCbus transaction at T1. The SCLC module, which is part of the MR4010, checks the phase LCLKp clock. At T4 and T5, the SCLC asserts address strobe, LADSn. During a write transaction, the SCLC must output data on the Lbus on the rising edge of LCLKp. The Lbus transaction starts at T4. At T12, the SCLC samples the LRDYn signal on the rising edge of LCLKp. The SCLC asserts the SCbus data ready signal, SCBRDYn, at T13. At the same time it asserts the bus sizing request signal, SCB32n. During a read transaction, the SCLC samples data on the LBus when it samples LRDYn. If the transaction is a write transaction, the CW4010 places data on the SCbus at T13.

www.DataSheet4U.com

www.DataSheet4U.com

Figure 23
Timing Requirements for an SCbus-to-Lbus Transaction



www.DataSheet4U.com

www.DataSheet4U.com

10.3 MR4010 as Slave on the Lbus

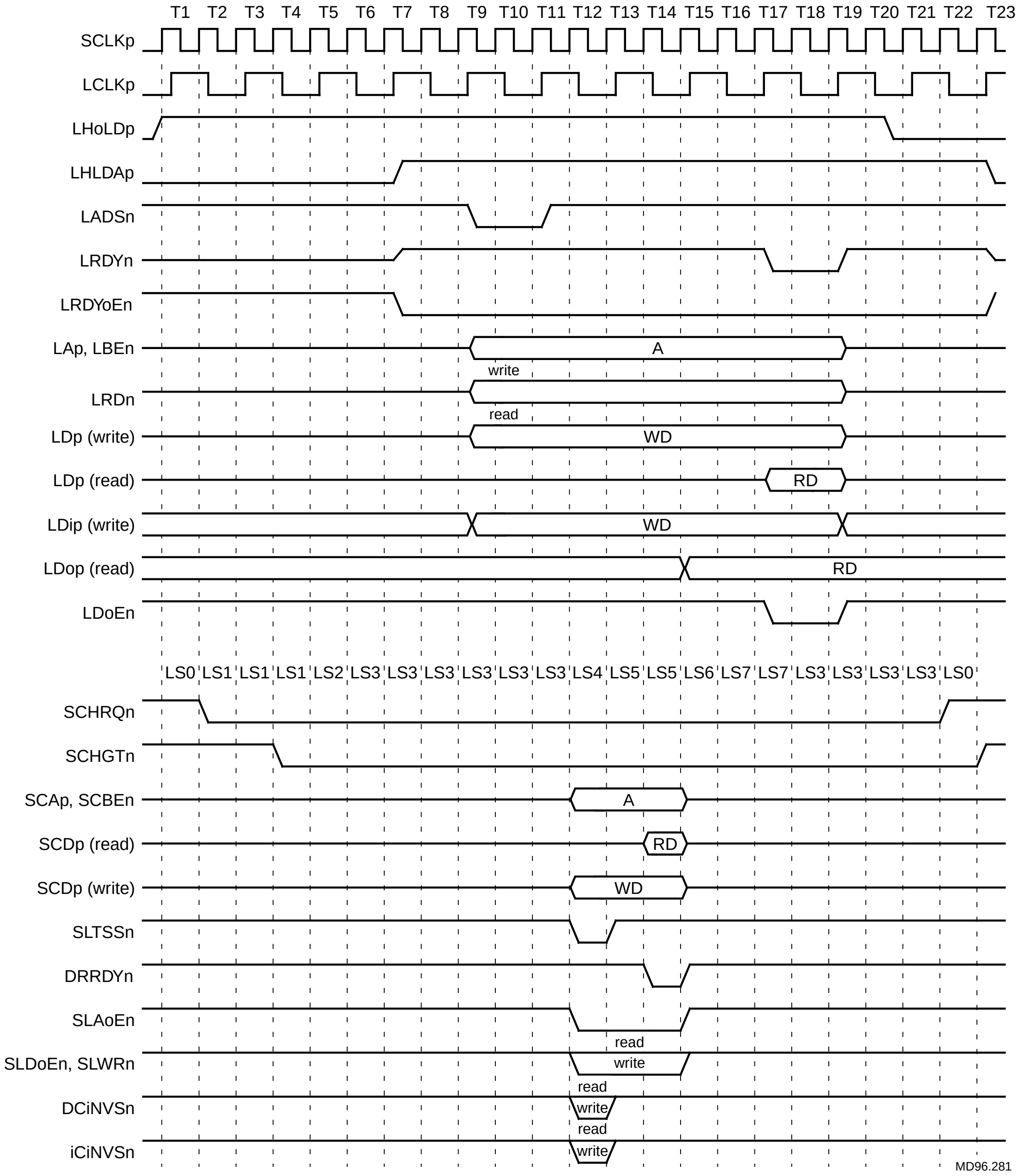
MR4010 functions as a slave on the Lbus when an Lbus device, such as a SONIC Ethernet Controller, initiates a bus transaction. If, for example, the Ethernet Controller wishes to access the system DRAM, it does this through the DRAM Controller, which is part of the MR4010. In this case, the MR4010 acts as a slave memory controller. The address being accessed must fall in the range 0x 0000 0000 through 0x 03FF FFFF. The MR4010 does not assert the data ready signal, since the transaction is treated as a read/write between an Lbus master and an Lbus slave. Figure 24 shows the timing requirements for this type of transaction.

At T1, the MR4010 samples LHoLDp on the rising edge of LCLKp. At T2, the SCLC module, which is part of the MR4010, asserts SCbus hold request, SCHRQn. The CW4010 asserts the SCbus hold grant signal, SCHGTn, at T4. At T7, the SCLC module asserts the Lbus hold acknowledge signal, LHLDAp, on the rising edge of LCLKp. While LHLDAp is asserted, the SCLC module asserts LRDYoEn to drive LRDYn. At T9 or later, the Lbus master (in this example the SONIC Ethernet Controller) starts an Lbus transaction. The SCLC samples LADSn on the rising edge of LCLKp. If the signal is asserted, the SCLC module knows the Ethernet Controller has initiated an Lbus transaction. At T12, the SCLC module decodes sampled address inputs and starts an SCbus transaction if the address is in the DRAM area. The DRAM Controller asserts the data ready signal, DRRDYn, when a transaction is completed. At T17 and T18, the SCLC module asserts LRDYn and the Lbus transaction is completed.

www.DataSheet4U.com

www.DataSheet4U.com

Figure 24
Timing Requirements
for Lbus-to-SCbus
Transaction



www.DataSheet4U.com

www.DataSheet4U.com

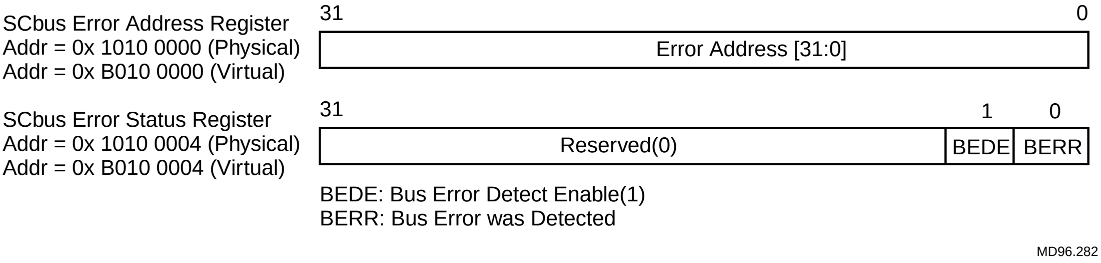
10.4 SCbus Timeout Watchdog Timer

The SCLC module in the MR4010 has a watchdog timer that it uses to time out SCbus transactions. The timer monitors the number of clock cycles for each SCbus transaction generated by the CW4010. It does not care about SCbus transactions to SCLC internal registers, or SCbus transactions generated by the SCLC module, since these transactions will never result in a timeout.

When the CW4010 initiates an SCbus transaction, the transaction must be completed within 256 SCLKp clock cycles, timed from the cycle in which SCTSSn is asserted until the cycle in which SCBRDYn or SCBRTYn is asserted. If the transaction takes longer than 256 clock cycles, the timer terminates the transaction by asserting SCBERRn, which causes a bus error exception.

The MR4010 has two registers that control SCbus timeout errors. As shown in Figure 25, they are the Error Status Register and the Error Address Register. The status register stores bus error detect enable (BEDE) and bus error detected (BERR) bits. The SCbus watchdog timer starts when the BEDE bit is set. If a timeout error occurs, the timer sets the BERR bit. During the time that BERR is set, the address register stores the SCbus transactions address and further bus error detection is inhibited. The SCLC can clear the bit by writing a 1 to it. A cold reset clears both bits.

Figure 25
SCbus Error Address and Status Register Bit Format


SCbus Error Address Register
Addr = 0x 1010 0000 (Physical)
Addr = 0x B010 0000 (Virtual)
31
0
Error Address [31:0]
SCbus Error Status Register
Addr = 0x 1010 0004 (Physical)
Addr = 0x B010 0004 (Virtual)
31
1
0
Reserved(0)
BEDE
BERR
BEDE: Bus Error Detect Enable(1)
BERR: Bus Error was Detected
MD96.282

www.DataSheet4U.com

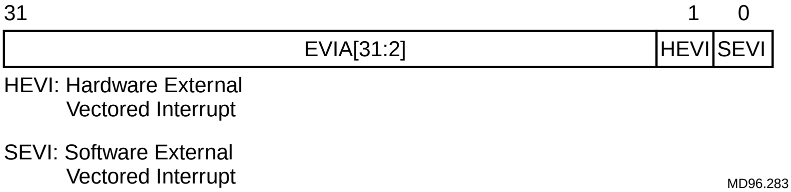
www.DataSheet4U.com

10.5 External Vectored Interrupt (EVInt) Support

The CW4010 has a special interrupt exception input feature called External Vectored Interrupt. The SCLC module in the MR4010 provides test support for this feature with the EXVI control register shown in Figure 26.

Figure 26
External Vectored Interrupt Register Bit Format

EXVI Register
Addr = 0x 1010 0008 (Physical)
Addr = 0x B010 0008 (Virtual)


31
1
0
EVIA[31:2]
HEVI
SEVI
HEVI: Hardware External Vectored Interrupt
SEVI: Software External Vectored Interrupt
MD96.283

When the CW4010 reads an exception vector address from EXVAp[31:2], it writes the address to the program counter. The EVIA[31:2] bits in the EXVI Register are connected to the EXVAp[31:2] bus to provide the vector address. When the CW4010 accepts an EVInt exception, it clears the EVIA bits to zero indicating that the timing was correct when the CW4010 sampled EXVAp[31:2].

If software sets the SEVI bit in the EXVI Register, the EVInt input of CW4010 is asserted and causes an exception. External Vectored Interrupts are enabled in the CCC Register, and interrupts are enabled in the Status Register. The software must write the extended address to the EVIA bit at the same time that it sets the SEVI bit.

The nonmaskable interrupt input to MR4010, NMI, can be used to cause an external vectored interrupt, EVInt. This bit selects the function of the NMI/EVInt pin. If the bit is cleared to 0, the pin generates a nonmaskable interrupt (NMI). If the bit is set to 1, the pin generates an external vectored interrupt (EVInt). The address is still supplied by the EVIA bits. If HEVI is cleared to 0, the falling edge of NMI causes an NMI exception. If HEVI is set to 1, the falling edge of NMI causes an EVInt exception provided that the interrupt enable bit the Status Register is set.

A cold reset clears all bits of the EXVI Register.

www.DataSheet4U.com

www.DataSheet4U.com

11 Cache Configuration and Maintenance

This section describes the Icache and Dcache configurations, and explains how to maintain the caches after power is turned on.

11.1 Cache Configuration

MR4010 takes advantage of the largest Icache and Dcache available. You can use the CCC Register in the CW4010 CP0 to program certain features of the caches. This allows you to evaluate the performance of different cache configurations and select the one most appropriate for your application. You can configure the Icache and Dcache independently. You can program the CCC Register to implement the following features:

- Select the cache operating size. Smaller cache configurations need wider tag bits. The MR4010 uses the maximum number of words for the maximum configuration and the widest tag bits for the minimum configuration. To set the size, you program bits IS[1:0] for the Icache, and DS[1:0] for the Dcache, as shown in Table 8.
- Select between direct-mapped and two-way set-associative caching. To do this you program bits IE[1:0] for the Icache and DE[1:0] for the Dcache, as shown in Table 8. IE1 and IE0 enable Icache Set-1 and Set-0, respectively, and DE1 and DE0 enable Dcache Set-1 and Set-0, respectively. In the example shown in Table 8, Set-0 is enabled for both the Icache and the Dcache when you require direct mapping, and Set-1 is disabled for both caches. When you select two-way set-associative caching, both sets are enabled for both caches. Note that when you select two-way set-associative caching, total cache capacity is doubled, since you are using both cache sets.
- Configure the Dcache as scratchpad RAM. Prior to configuring a set associativity as scratchpad RAM, you must use cache isolation mode to program the corresponding tag memory to contain the desired physical addresses. When using isolate cache mode, stores to cache are not propagated to external memory. To initiate isolate cache mode, you set bit IsC in the CCC Register.

  Once this process is complete, you can configure the Dcache as scratchpad RAM by programming bits DE0 and SR0 to configure

www.DataSheet4U.com

www.DataSheet4U.com

Dcache Set-0, and DE1 and SR1 to configure Dcache Set-1, as shown in Table 9.

- Configure the Icache as scratchpad RAM. Prior to configuring Set-1 as scratchpad RAM, you must use cache isolation mode to program the tag memory to contain the desired physical addresses. In addition, you must program the corresponding data fields to contain the instruction code which is to remain resident in the cache.

  Once this process is complete, you can configure the Icache as scratchpad RAM by programming bits IE1 and IR1 to configure Icache Set-1, as shown in Table 10.

Table 10
Cache Size and Accessing

	Bit Settings			
Icache	**IE1**	**IE0**	**IS[1:0]**	
Dcache	**DE1**	**DE0**	**DS[1:0]**	**Configuration**
	0	0	X X[1]	No cache
	0	1	0 0	1 Kbyte direct mapped
	0	1	0 1	2 Kbyte direct mapped
	0	1	1 0	4 Kbyte direct mapped
	0	1	1 1	8 Kbyte direct mapped
	1	1	0 0	2 Kbyte two-way set-associative
	1	1	0 1	4 Kbyte two-way set-associative
	1	1	1 0	8 Kbyte two-way set-associative
	1	1	1 1	16 Kbyte two-way set-associative

[1] The setting of these bits does not matter.

Table 11
Dcache Scratchpad RAM Configuration

	Bit Settings		
Dcache Set-0	**DE0**	**SR0**	
Dcache Set-1	**DE1**	**SR1**	**Configuration**
	0	X	Disabled
	1	0	Cache memory
	1	1	Scratchpad RAM

www.DataSheet4U.com

www.DataSheet4U.com

Table 12
Icache Scratchpad RAM Configuration

	Bit Settings		
Icache Set-1	IE1	IR1	Configuration
	0	X	Disabled
	1	0	Cache memory
	1	1	Scratchpad RAM

11.2 Cache Maintenance

When power is turned on to the MR4010, valid bits in the CCC Register have random values. After you have programmed the CCC Register to select a cache configuration, you must make sure that Cache Tag valid bits are cleared during the reset initialization period.

CW4010 has the following instructions that you can use to flush the caches:

- FLUSHID flushes the ICache and the Dcache
- FLUSHI flushes the ICache
- FLUSHD flushes the DCache

These instructions do not have any operand. To invalidate ICache and DCache during reset initialization, use FLUSHID. Each flush instruction causes stall cycles for 256 clock cycles regardless of cache size. You must execute the instructions from the kseg1 uncached and unmapped area.

12 Organization of Specific Internal Signals

This section describes the organization of the MR4010's clock circuitry that controls the MR4010's clock inputs and outputs, and MR4010's synchronization circuitry that handles exception inputs.

12.1 Clock Circuitry

The PLL circuit must supply the CW4010 core with the system clock, SCLKp. Figure 27 shows how the PLL output is distributed to internal MR4010 modules, such as the DRAM Controller and the SCLC, as well as to the CW4010 core itself. The phase time of the SCLKp inputs is the same for all internal modules.

www.DataSheet4U.com

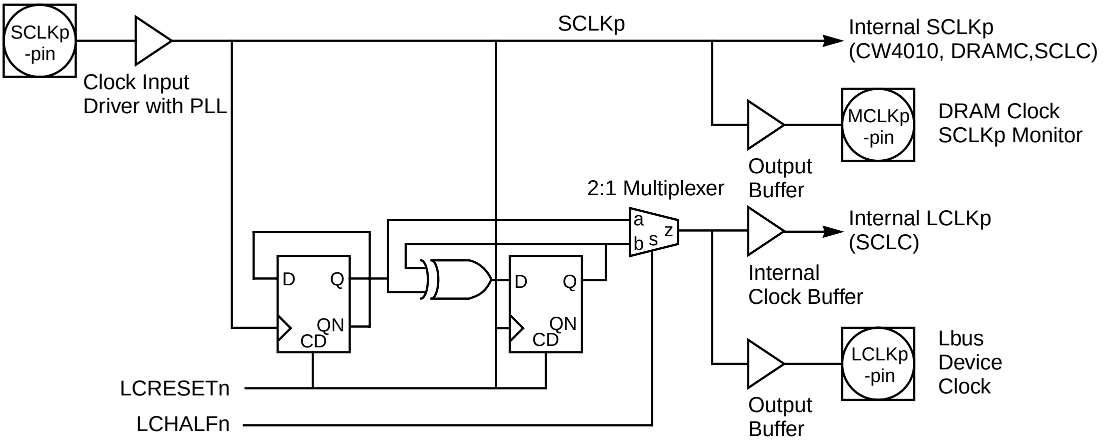
www.DataSheet4U.com

The MR4010 buffers SCLKp and outputs it as MCLKp, which monitors the internal clock, defines relative AC specifications for SCLKp synchronized inputs and outputs, and may be used as the DRAM clock.

The MR4010 generates the clock for the Lbus by dividing the SCLKp either by 2 or by 4. SCLKp is passed through a two-stage D-type flip-flop, as shown in Figure 27, and output to a 2:1 multiplexer, which is controlled by the LCHALFn input. When LCHALFn is asserted, on pin ‘s’ of the multiplexer, the multiplexer outputs the clock signal on pin ‘a’ and passes the clock signal through the second stage of the flip-flop for a second time. LCHALFn is then deasserted, and output ‘z’ from the multiplexer outputs a clock signal that is 1/4 of the original SCLKp. If LCHALFn is deasserted when SCLKp is initially input to the multiplexer, output ‘z’ outputs a clock signal that is 1/2 of the original SCLKp. The Lbus clock, LCLKp is buffered and used as an internal clock for the SCLC. It is also output on the Lbus to provide the clock for Lbus devices. Devices on the Lbus sample all inputs on the rising edge of LCLKp, and synchronize all outputs to the rising edge of LCLKp.

Table 13 summarizes the clock generation process. Figure 28 shows the timing requirements for the CW4010 and Lbus clocks.

Figure 27
MR4010 PLL Clock Circuitry



Note: LCHALFn: 0 = 1/2, 1 = 1/4. Rev. A: 1 = 1/2, 1 = 1/4.

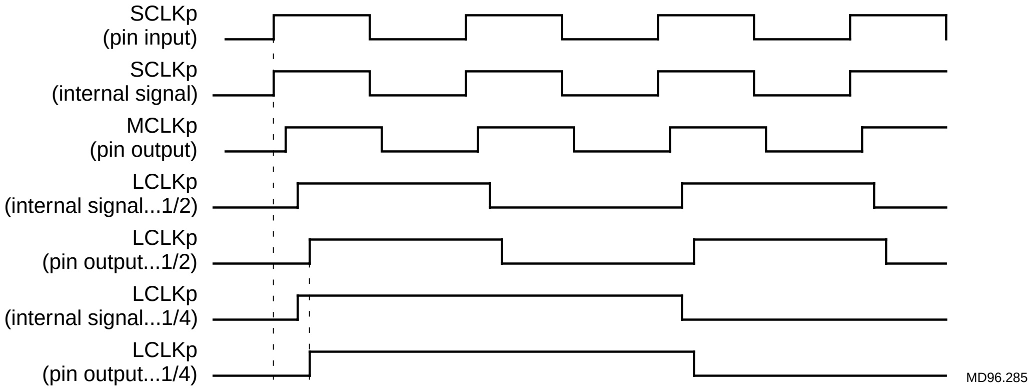
www.DataSheet4U.com

www.DataSheet4U.com

Table 13
Summary of MR4010
Clocks

Clock Name	Source	Frequency	Comments
SCLKp	Pin input SCLKp	DC to 66 MHz	CW4010 clock
MCLKp	SCLKp	Same as SCLKp frequency	DRAM clock, SCLKp monitor clock
LCLKp	SCLKp divided by 2, or SCLKp divided by 4	1/2 or 1/4 of SCLKp frequency	Lbus clock

Figure 28
Timing Requirements
for the CW4010 and
Lbus Clocks



12.2 Exception Inputs

Exception inputs to the MR4010 may be asynchronous. These inputs include:

- Cold reset exception input, SCRESETn
- Warm reset exception input, SWRESETn
- Nonmaskable interrupt exception, SNMin
- External interrupt exceptions, SEXTiNTn[5:0]

The SCLC module in the MR4010 has a synchronization circuit that synchronizes these inputs to the system clock, SCLKp. As shown in Figure 29, the synchronization circuit consists of a series of D-type flip-flops that are clocked on the rising edge of SCLKp. The exception

www.DataSheet4U.com

www.DataSheet4U.com

inputs reset the first stage, Flip-Flop A. On the rising edge of SCLKp, the Q output from A is passed to the D-input of Flip-Flop B. The next SCLKp input clocks this stage, and the Q output from B is passed to the D-input of the final stage, which outputs synchronous exception signals on the rising edge of the third SCLKp. Figure 30 shows the timing requirements for the synchronization circuit.

Figure 29
Exception Inputs
Synchronization
Circuitry

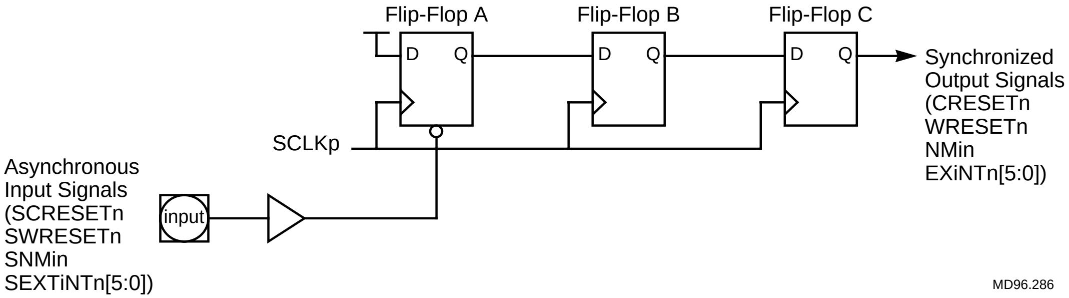
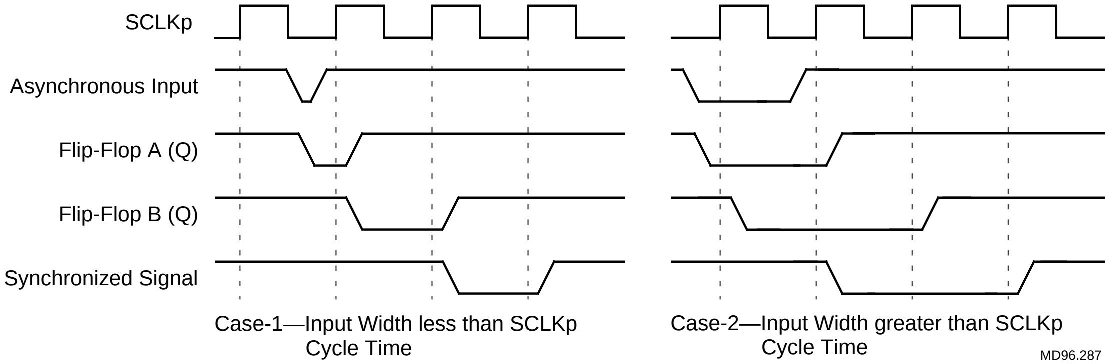


Figure 30
Timing Requirements
for Synchronization
Circuit



www.DataSheet4U.com

www.DataSheet4U.com

13 Electrical Characteristics

This section defines the electrical characteristics of the MR4010 Reference Device.

Table 14 lists the absolute maximum ratings of the MR4010.

Table 14
Absolute Maximum Ratings

Symbol	Parameter	Limits (Referenced to VSS)	Unit
VDD	DC supply	-0.3 to +3.9	V (Volts)
VIN	Input voltage	-1.0 to VDD +0.3	V (Volts)
VIN	5 V compatible input voltage	-1.0 to +6.5	V (Volts)
IIN	DC input current	±10	mA (milliamperes)[1]
TSTG	Storage temperature range	-65 to +150	˚C (degrees Centigrade)

[1] Except for power pins.

Table 15 lists the recommended operating conditions for the MR4010.

Table 15
Recommended Operating Conditions

Symbol	Parameter	Limits (Referenced to VSS)	Unit
VDD	DC supply, commercial	3.3 ± 5%	Volts
TC	Case temperature	0 to 85	˚C (degrees Centigrade)

Table 16 lists the capacitance of the MR4010’s input and output signals.

Table 16
Input/Output Capacitance

Symbol	Parameter	Limits (Referenced to VSS)	Unit
CIN	Input capacitance	5	pF (picafarads)
COUT	Output capacitance	10	pF
CIO	I/O buffer capacitance	15	pF

www.DataSheet4U.com

www.DataSheet4U.com

Table 17 lists the MR4010’s DC characteristics.

Table 17
DC Characteristics

			Limits			
Symbol	**Parameter**	**Condition**	**Min.**[1]	**Typ.**[2]	**Max.**[3]	**Unit**
VIL	Input voltage low		-0.5	—	0.8	V (Volts)
VIH	Input voltage high		2.0	—	VDD + 0.3	V
VOL	Output voltage low		—	0.2	0.4	V
VOH	Output voltage high		2.4	VDD - 0.3	—	V
IIL	Input leakage current	VDD = max. VIN = VDD or VSS	-10	1	±10	A (microamperes)
IOZ	3-state output leakage current	VDD = max. VIN = VDD or VSS	-10	1	±10	A

1. Minimum
2. Typical
3. Maximum

Table 18 lists the AC timing specifications for the MR4010. Figure 31 shows timing relationships. The specifications are valid in the temperature range 0–85 ˚C case; VDD 3.3 V, ±5%. Setup and hold times, which are relevant only for inputs to the MR4010, are referenced to the rising edge of the system clock (SCLKp) or the Lbus clock (LCLKp). The valid maximum times are equivalent to hold time for the MR4010’s outputs. They are not relevant for the inputs. They are referenced to the rising edge of SCLKp or LCLKp. For 3-state signals, valid maximum times include the period from high Z to valid and valid to high Z. (Z indicates the 3-state or ‘off’ condition of the signal.)

www.DataSheet4U.com

www.DataSheet4U.com

The buffer types referenced, *b8rp*, and so on, are LSI Logic ASIC macro cells. Output timing is calculated in all instances for a capacitance of 60 picofarads (pF). You can get detailed information about these cells from the LSI Logic *LCB500K Preliminary Design Manual*.

Table 18
MR4010 AC Timing Specifications

				Input Timing		
Signal Name	**Reference Clock**	**Input/Output (I/O)**	**Buffer Type**	**Setup**	**Hold**	**Output Timing Valid Max.**
MAp[11:0	SCLKp	O	b8rp	—	—	7.5 ns
MDp[63:0]	SCLKp	I/O	bd2c	4.5 ns	2 ns	9.5 ns
MCSn[1:0]	SCLKp	O	b6r	—	—	7.5 ns
MRASn	SCLKp	O	b8rp	—	—	7.5 ns
MCASn	SCLKp	O	b8rp	—	—	7.5 ns
MWEn	SCLKp	O	b8rp	—	—	7.5 ns
MDQMp[7:0]	SCLKp	O	b2	—	—	7.5 ns
LCLKp	SCLKp	O	b12	—	—	13 ns
LAp[31:2]	LCLKp	I/O	bd4crf	5 ns	2 ns	10 ns
LDp[31:0]	LCLKp	I/O	bd4crf	9 ns	2 ns	10 ns
LBEn[3:0]	LCLKp	I/O	bd4crf	5 ns	2 ns	10 ns
LRDn[1]	LCLKp	I/O	bd4crf	5 ns	2 ns	10 ns
LADSn[1]	LCLKp	I/O	bd4crf	5 ns	2 ns	10 ns
LRDYn[1]	LCLKp	I/O	bd4crf	5 ns	2 ns	10 ns
LRTYn[1]	LCLKp	I	ibuff	5 ns	2 ns	—
LHoLDp[1]	LCLKp	I	ibuff	5 ns	2 ns	—
LHLDAp[1]	LCLKp	O	b2	—	—	10 ns
SCRESETn[1, 2]	SCLKp	I	schmitcf	8 ns	2 ns	—
SWRESETn[1, 2]	SCLKp	I	ibuff	8 ns	2ns	—
SNMin[1, 2]	SCLKp	I	ibuff	8 ns	2 ns	—
SEXINTn[5:0][1, 2]	SCLKp	I	ibuff	8 ns	2 ns	—

1. Setup and hold times guaranteed by design.
2. These are asynchronous inputs that are synchronized in the MR4010, as described in "External Vectored Interrupt (EVInt) Support" on page 86. Input setup and hold times specify the times these signals are sampled.

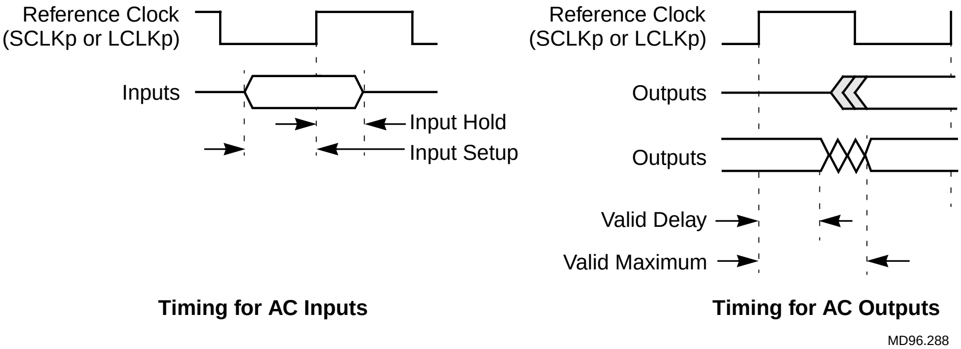
www.DataSheet4U.com

www.DataSheet4U.com

The following parameters are critical and you should check them carefully.

1. Mbus outputs valid minimum—DRAM requirement time is 1 ns.
2. Lbus outputs valid minimum—related data hold-time parameters for Lbus devices.

Figure 31
AC Timing for MR4010
Inputs and Outputs



14 Package Information

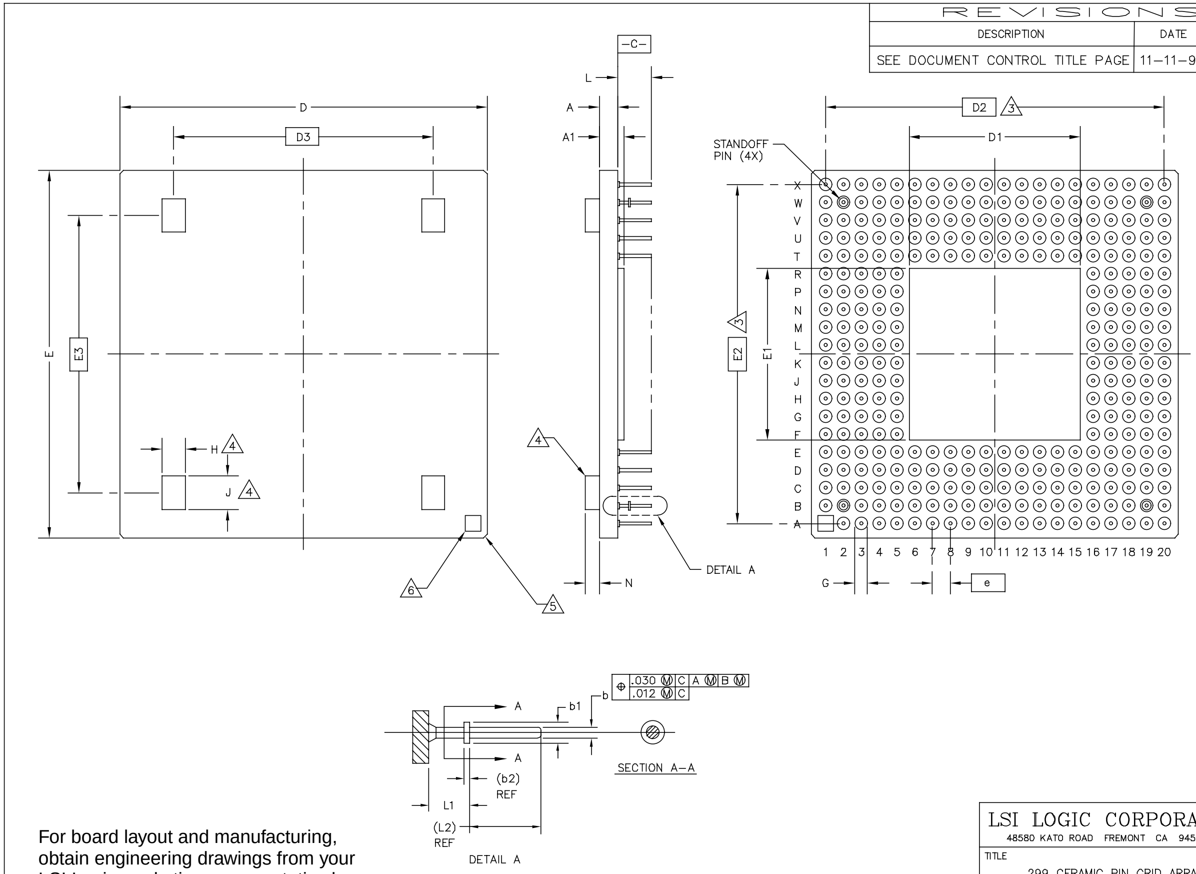
This section provides packaging information for the MR4010 Reference Device. Figure 32 shows the mechanical layout and dimensions, and the pin locations (A1, and so on). Table 19 lists the pin assignments.

www.DataSheet4U.com

Figure 32
Mechanical Drawing of the 299-Pin CPGA (FT)
MR4010 Device

REVISIONS

DESCRIPTION	DATE	REV.
SEE DOCUMENT CONTROL TITLE PAGE	11–11–96	7



For board layout and manufacturing, obtain engineering drawings from your LSI Logic marketing representative by requesting the outline drawing for package code FT.

LSI LOGIC CORPORATION
48580 KATO ROAD FREMONT CA 94539

TITLE
299 CERAMIC PIN GRID ARRAY

CODE FT MD96.FTa

DWG NO. JZ01–000130–00 REV. 7

SCALE 3 : 1 SHEET 1 of 2

www.DataSheet4U.com

www.DataSheet4U.com

Figure 32 (Cont.)
Mechanical Drawing of the 299-Pin CPGA (FT)
MR4010 Device

DIMENSIONS IN INCHES

SYM	MIN	NOM	MAX	NOTES
A	0.091	0.101	0.111	
A1	–	–	0.136	2
b	0.016	0.018	0.020	
b1	0.045	0.050	0.055	
b2	0.008 REF			
D	2.044	2.060	2.076	
D1	–	–	0.960	2
D2	1.900 BSC			3
D3	1.455 BSC			
D3 *	1.700 BSC			
E	2.044	2.060	2.076	
E1	–	–	0.960	2
E2	1.900 BSC			3
E3	1.555 BSC			
E3 *	1.670 BSC			
e	0.100 BSC			
G	0.060	0.070	0.080	
H	–	–	0.130	
J	0.188 REF			
J *	0.130 REF			
L	0.185	0.190	0.195	
L1	0.065	0.070	0.075	
L2	0.120 REF			
N	–	–	0.080	
TOTAL NO. OF PINS	299			

NOTES:

1 THIS DRAWING IS APPLICABLE TO LSI PART NUMBERS JA01–000130–01 THROUGH –40.

2 ENVELOPE FOR METAL SEAL.

3 SIZE OF ARRAY: 20 X 20; JEDEC CODE AM (LARGE OUTLINE).

4 CHIP CAPACITOR ENVELOPE, OPTIONAL.
* CHAP CAPACITOR LOCATION FOR LSI LOGIC PART NUMBERS JA01–000130–22,–24,–36 ONLY.

5 CORNER DETAIL IS LSI LOGIC CORPORATION OPTION.

6 INDEX MARK IS LSI LOGIC CORPORATION OPTION.

7. PLATING: ALL EXPOSED SURFACES SHALL BE GOLD PLATED OVER NICKEL PER LSI SPECIFICATION JB00–000002–00.

LSI LOGIC CORPORATION
48580 KATO ROAD FREMONT CA 94539

TITLE
299 CERAMIC PIN GRID ARRAY

CODE FT

DWG NO. JZ01–000130–00
REV. 7

SCALE

SHEET 2 of 2

www.DataSheet4U.com

www.DataSheet4U.com

www.DataSheet4U.com

Table 19
MR4010 Pin
Assignments

Pad #	Pin #	Signal Name	Pad #	Pin #	Signal Name	Pad #	Pin #	Signal Name
1	C3	V_{DD2}	37	B13	LDp8	73	D19	SEXiNTn0
2	E5	V_{DD2}	38	C11	LDp9	74	E18	SEXiNTn1
3	B3	LAp14	39	A14	LDp10	75	D20	SEXiNTn2
4	E6	LAp13	40	C12	LDp11	76	E19	SEXiNTn3
5	C4	LAp15	41	C13	LDp12	77	F18	SEXiNTn4
6	D6	LAp16	42	D12	V_{SS}	78	E20	SEXiNTn5
7	D5	LAp17	43	B14	LDp13	79	G17	V_{SS2}
8	E7	LAp18	44	E12	LDp14	80	F19	V_{DD2}
9	B4	LAp19	45	A15	LDp15	81	H16	V_{SS2}
10	C5	LAp20	46	D13	LDp16	82	G18	BRLiKFn
11	B5	LAp21	47	C14	LDp17	83	H17	PCANCRn
12	A5	LAp22	48	E13	LDp18	84	F20	PCANoDDn
13	C6	LAp23	49	B15	LDp19	85	J16	PSTALLn
14	B6	LAp24	50	A16	V_{DD2}	86	G19	SCAoEn
15	D7	LAp25	51	C15	LDp20	87	J17	SCBRDYn
16	A6	LAp26	52	B16	LDp21	88	H18	SCDoEn
17	C7	V_{DD2}	53	A17	LDp22	89	J18	SCTBSTn
18	E8	LAp27	54	C16	LDp23	90	G20	SCTSSn
19	B7	LAp28	55	B17	LDp24	91	K18	SCiFETn
20	D8	LAp29	56	D16	LDp25	92	H19	SUSPEXn
21	A7	LAp30	57	D14	LDp26	93	K16	V_{DD2}
22	E9	LAp31	58	C17	LDp27	94	J19	V_{SS2}
23	C8	LBEn0	59	E14	LDp28	95	K17	V_{DD2}
24	D9	LBEn1	60	B18	LDp29	96	K19	V_{DD2}
25	B8	LBEn2	61	D15	LDp30	97	L17	V_{DD2}
26	C9	LBEn3	62	B19	LDp31	98	L19	MCLKp
27	A8	LDp0	63	E15	V_{DD2}	99	L16	MDp0
28	C10	LDp1	64	D17	V_{DD2}	100	M19	MDp1
29	B9	LDp2	65	C18	V_{DD2}	101	L18	MDp2
30	E10	LDp3	66	E16	V_{DD2}	102	N20	MDp3
31	B10	LDp4	67	C19	V_{DD2}	103	M18	MDp4
32	D10	V_{DD2}	68	F16	BENDn	104	N19	MDp5
33	B11	V_{DD2}	69	D18	FRCMn	105	M17	MDp6
34	D11	LDp5	70	F17	SCRESETn	106	P20	MDp7
35	B12	LDp6	71	E17	SWRESETn	107	M16	MDp8
36	E11	LDp7	72	G16	SNMin	108	N18	MDp9

(Sheet 1 of 3)

www.DataSheet4U.com

www.DataSheet4U.com

Table 19 (Cont.)
MR4010 Pin
Assignments

Pad #	Pin #	Signal Name
109	N17	V_{DD2}
110	P19	V_{DD2}
111	N16	V_{SS}
112	R20	V_{SS}
113	P18	MDp10
114	R19	MDp11
115	T20	MDp12
116	R18	MDp13
117	T19	MDp14
118	U20	MDp15
119	T18	MDp16
120	U19	MDp17
121	P17	MDp18
122	T17	MDp19
123	P16	MDp20
124	U18	MDp21
125	R17	MDp22
126	V19	MDp23
127	R16	V_{DD2}
128	V18	V_{DD2}
129	U17	V_{DD2}
130	T16	V_{DD2}
131	W19	MDp24
132	T15	MDp25
133	W18	MDp26
134	U15	MDp27
135	V17	MDp28
136	T14	MDp29
137	U16	MDp30
138	W17	MDp31
139	V16	MDp32
140	W16	MDp33
141	X16	MDp34
142	V15	MDp35
143	W15	MDp36
144	U14	V_{DD2}
145	T13	MDp37
146	X15	MDp38
147	U13	MDp39
148	V14	MDp40
149	T12	MDp41
150	W14	MDp42
151	U12	SCANMONp
152	X14	SCANKZop
153	V12	SCANKZip
154	V13	SCANENBp
155	V11	SCANCRop
156	W13	SCANCRip
157	T11	PARAMOUTp
158	X13	ZSTATEn
159	U11	TESTMp
160	W12	V_{SS2}
161	U10	V_{DD2}
162	W11	V_{DD2}
163	T10	PLLVSS
164	W10	PLLLP2p
165	V10	PLLAGND
166	W9	PLLVDD
167	V9	PLLREFp
168	W8	V_{SS2}
169	U9	PLLCTRn
170	X7	PLLCTop
171	T9	PLLENp
172	V8	PLLTDp
173	U8	PLLTSTp
174	W7	PLLiDDTp
175	T8	V_{SS2}
176	X6	V_{DD2}
177	V7	MDp43
178	W6	MDp44
179	X5	MDp45
180	V6	MDp46
181	W5	MDp47
182	X4	MDp48
183	V5	MDp49
184	W4	MDp50
185	U7	MDp51
186	U5	MDp52
187	T7	MDp53
188	V4	MDp54
189	U6	MDp55
190	W3	MDp56
191	T6	V_{DD2}
192	V3	V_{DD2}
193	U4	V_{DD2}
194	T5	V_{DD2}
195	W2	MDp57
196	R5	MDp58
197	V2	MDp59
198	R4	MDp60
199	U3	MDp61
200	P5	MDp62
201	T4	MDp63
202	U2	MRASn
203	T3	MCASn
204	U1	MWEn
205	T2	MAp0
206	R3	MAp1
207	T1	MAp2
208	P4	MAp3
209	R2	V_{SS}
210	N5	V_{SS}
211	P3	V_{DD2}
212	N4	V_{DD2}
213	R1	MAp4

(Sheet 2 of 3)

www.DataSheet4U.com

www.DataSheet4U.com

Table 19 (Cont.)
MR4010 Pin
Assignments

Pad #	Pin #	Signal Name	Pad #	Pin #	Signal Name	Pad #	Pin #	Signal Name
214	M5	MAp5	229	J2	MDQMp3	244	F3	LADSn
215	P2	MAp6	230	K3	MDQMp4	245	E1	LAp2
216	M4	MAp7	231	H1	MDQMp5	246	E2	LAp3
217	N3	MAp8	232	J3	MDQMp6	247	E3	LAp4
218	M3	MAp9	233	H2	MDQMp7	248[1]	G4	LAp5
219	P1	MAp10	234	J4	V_{SS2}	249	D2	LAp6
220	L3	MAp11	235	H3	LCRESETn	250	G5	LAp7
221	N2	MCSn0	236	J5	LCHALFn	251	D3	LAp8
222	L5	MCSn1	237	G1	LRTYn	252	F4	LAp9
223	M2	V_{SS2}	238	H4	LHoLDp	253	C2	LAp10
224	L4	V_{DD2}	239	G2	LRDYn	254	E4	LAp11
225	L2	V_{DD2}	240	H5	LRDn	255	B2	LAp12
226	K4	MDQMp0	241	G3	V_{DD2}	256	F5	V_{DD2}
227	K2	MDQMp1	242	F1	LHLDAp	257	D4	V_{DD2}
228	K5	MDQMp2	243	F2	LCLKp	286	—	—

(Sheet 3 of 3)

[1] Pad 248/Pin G4 is provided as an extra I/O for the user. However, it cannot be tested by LSI Logic Corporation, because the maximum I/O test capability is 256.

Notes:

- The following pins are used for power input VDD: A3, A9, A11, A13, A18, A20, B1, C20, D1, J20, K1, L20, M1, V20, W1, X2, X8, X10, X12, X18, X20.
- The following pins are used for power input VSS: A2, A4, A10, A12, A19, B20, C1, H20, J1, K20, L1, M20, N1, V1, W20, X1, X3, X9, X11, X17, X19.

www.DataSheet4U.com

www.DataSheet4U.com

Notes

www.DataSheet4U.com

www.DataSheet4U.com

Notes

www.DataSheet4U.com

www.DataSheet4U.com

Notes

Headquarters

LSI Logic Corporation
North American Headquarters
Milpitas CA
Tel: 408.433.8000
Fax: 408.433.8989

LSI Logic Europe plc
European Headquarters
United Kingdom
Tel: 44.1344.426544
Fax: 44.1344.481039

LSI Logic K.K.
Headquarters
Tokyo Japan
Tel: 81.3.5463.7821
Fax: 81.3.5463.7820

To receive product literature, call us at 1-800-574-4286 (U.S. and Canada); +32.11.300.531 (Europe); 408.433.7700 (outside U.S., Canada, and Europe) and ask for Department JDS; or visit us at http://www.lsilogic.com

ISO 9000 Certified

♻ Printed on
Recycled Paper

Printed in USA
Order No. C15017
1296.500.G
Doc. No. DB09-000028-00

LSI Logic logo design and CoreWare are registered trademarks, and MiniRISC is a trademark of LSI Logic Corporation. All other brand and product names may be trademarks of their respective companies.

LSI Logic Corporation reserves the right to make changes to any products and services herein at any time without notice. LSI Logic does not assume any responsibility or liability arising out of the application or use of any product or service described herein, except as expressly agreed to in writing by LSI Logic; nor does the purchase, lease, or use of a product or service from LSI Logic convey a license under any patent rights, copyrights, trademark rights, or any other of the intellectual property rights of LSI Logic or of third parties.

www.DataSheet4U.com