Cyrix® M II™ DATA BOOK

February 26, 1999

Updates for this manual can be obtained from
Cyrix Web site: www.cyrix.com.

Order Number: 94329-00
Cyrix Corporation
2703 North Central Expressway
Richardson, Texas 75080-2010
United States of America

M II DATA BOOK REVISION HISTORY

Date	Version	Revision
2/26/99	2.9	Updated device list.
1/8/1999	2.8	Revision history included.
1/5/1999	2.7	Page 5-9. Changed Power rating for 225 and 233 MHz and added 250 MHz line to Table 5-3.
12/4/1998	2.6	Page 4-5. Typos in Table 4-5, first column
12/4/1998		Page 2-33. Reworded text and clarified tables
12/4/1998		Page 3-6. Added VCC2H/L# pin Page 3-6
12/3/1998	2.5	Page 3-7 replaced with pages 3-7a and 3-7b. Added new definitions for CLKMUL1/2 for Rev 2.3.1and above devices
10/29/1998	2.4	Changed eight line of Table 4-5 on page 4-5 WAS: ICCSM Active ICC NOW: ICC Suspend Mode ICC Added the word "Suspend" to . Now reads ICCSM Active Suspend ICC
10/22/1998	2.3	Modified drawing on page 5-7 to show chip heat sink correctly, dimensions A and A1.
10/22/1998		Caption for Figure 5-4 changed: WAS: Figure 5-4 296-Pin SPGA Package B NOW: Figure 5-4 296-Pin Flip-Chip C4 SPGA Package B
10/ 22, 1998	2.2	Caption for Table 5-4 Page 5-8 WAS: Table 5-4 , 296-Pin SPGA Package B Dimensions NOW: Table 5-4 , 296-Pin Flip-Chip C4 SPGA Package B Dimensions
6/17/ 1998	2.15	Pages 4-8 through 4-13 Added AC timing data for 83 MHz bus.

Date	Version	Revision
5/5/1998	2.1	Page 2-11 Added Address Region Registers and Region Control Registers to System Register Set paragraphs. Added Model Specific Registers to System Register Set paragraphs.
5/5/1998		Pages 2-38 through 2-41 Changed MSR Registers Numbers to MSR Registers Indexes. For example MSR11 register was changed to MSR(11). Modified Table 2-22.
5/5/1998		Page 2-58 First paragraph. TR3 though TR5 can now be accessed as MSR(3) though MSR(5). Second paragraph. Unified cache description clarified and expanded.
4/24/98	2.0	Added 225 MHz, and 300 MHz Devices. Changed part numbers as follows on pages 4-5 and A-2. Also changed corresponding Icc, Iccsm, Active Power ratings WAS: 233 MHz (M II -300) 66/233 250 MHz (M II -333) 100/300 NOW: 225 MHz (M II -300) 75/225 233 MHz (M II -300) 66/233 250 MHz (M II -333) 100/250 300 MHz (M II -350) 100/300

MII™ PROCESSOR

Enhanced High Performance CPU

Introduction

♦ Enhanced Sixth-Generation Architecture

- MII-300 and higher
- 64K 4-Way Unified Write-Back Cache
- 2 Level TLB (16 Entry L1, 384 Entry L2)
- Branch Prediction with a 512-entry BTB
- Enhanced Memory Management Unit
- Scratchpad RAM in Unified Cache
- Optimized for both 16- and 32-Bit Code
- High Performance 80-Bit FPU

♦ X86 Instruction Set Includes MMX™ Instructions

- Compatible with MMX™ Technology
- Runs Windows® 95, Windows 3.x, Windows NT, DOS, UNIX®, OS/2®, Solaris®, and others

♦ Other Features

- Socket 7 Pinout Compatible
- 2.9 V Core, 3.3 V I/O
- Flexible Core/Bus Clock Ratios (2x, 2.5x, 3x, 3.5x)
- Leverages Existing Socket Infrastructure

The Cyrix MII™ processor is an enhanced processor with high speed performance. This processor has a 64K unified write-back cache, a two- level TLB and a 512-entry BTB. The MII CPU contains a scratchpad RAM feature, supports performance monitoring, and allows caching of both SMI code and SMI data. It delivers high 16- and 32-bit performance while running Windows 95, Windows NT, OS/2, DOS, UNIX, and other operating systems.

The MII processor achieves top performance through the use of two optimized superpipelined integer units, an on-chip floating point unit, and a 64 KByte unified write-back cache. The superpipelined architecture reduces timing constraints and increase frequency scalability. Advanced architectural techniques include register renaming, out-of-order completion, data dependency removal, branch prediction and speculative execution.

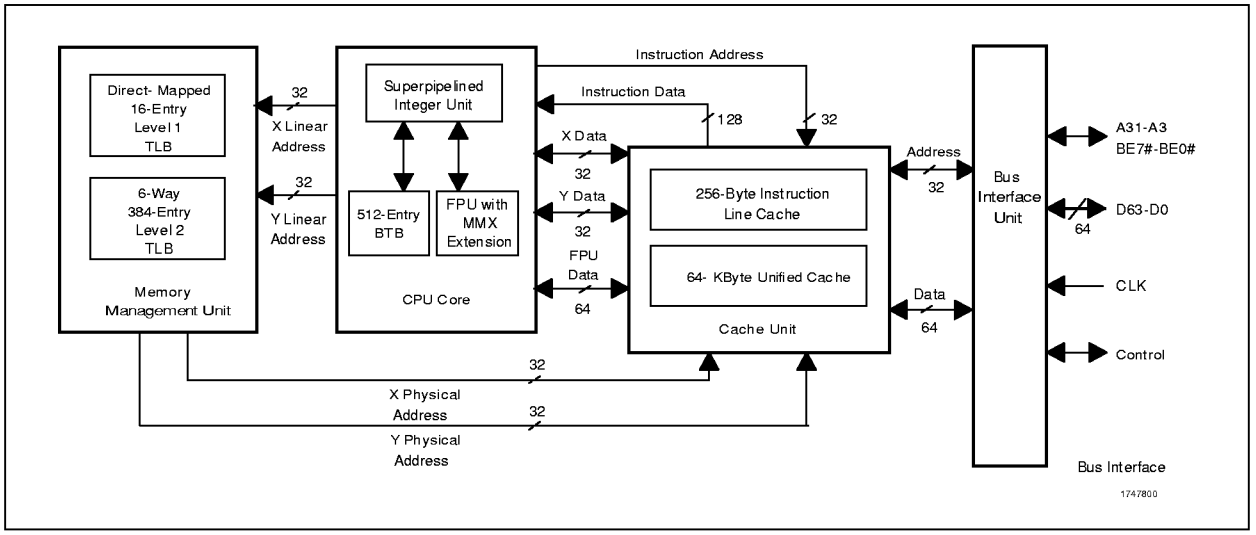
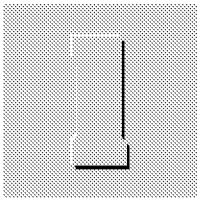
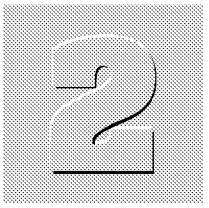
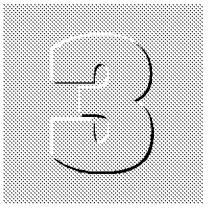
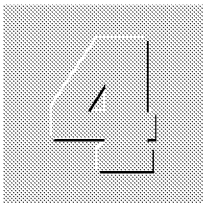
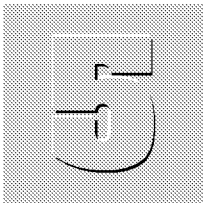
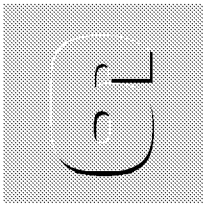


TABLE OF CONTENTS

Cyrix®
Advancing the Standards

1. ARCHITECTURE OVERVIEW

The Cyrix MII™ processor operates at higher frequencies than the 6x86MX™ processors. The MII processor, based on the proven 6x86 core, is superscalar in that it contains two separate pipelines that allow multiple instructions to be processed at the same time. The use of advanced processing technology and superpipelining (increased number of pipeline stages) allow the MII CPU to achieve high clocks rates.

Through the use of unique architectural features, the MII processor eliminates many data dependencies and resource conflicts, resulting in optimal performance for both 16-bit and 32-bit x86 software.

For maximum performance, the MII CPU contains two caches, a large unified 64 KByte 4-way set associative write-back cache and a small high-speed instruction line cache.

To provide support for multimedia operations, the cache can be turned into a scratchpad RAM memory on a line by line basis. The cache area set aside as scratchpad memory acts as a private memory for the CPU and does not participate in cache operations.

Within the MII processor there are two TLBs, the main L1 TLB and the larger L2 TLB. The direct-mapped L1 TLB has 16 entries and the 6-way associative L2 TLB has 384 entries.

The on-chip FPU has been enhanced to process MMX™ instructions as well as the floating point instructions. Both types of instructions execute in parallel with integer instruction processing. To facilitate FPU operations, the FPU features a 64-bit data interface, a four-deep instruction queue and a six-deep store queue.

The CPU operates using a split rail power design. The core runs on a 2.9 volt power supply, to minimize power consumption. External signal level compatibility is maintained by using a 3.3 volt power supply for the I/O interface.

For mobile systems and other power sensitive applications, the MII processor incorporates low power suspend mode, stop clock capability, and system management mode (SMM).

1.1 Major Functional Blocks

The MII processor consists of four major functional blocks, as shown in the overall block diagram on the first page of this manual:

- Memory Management Unit
- CPU Core
- Cache Unit
- Bus Interface Unit

The CPU contains the superpipelined integer unit, the BTB (Branch Target Buffer) unit and the FPU (Floating Point Unit).

The BIU (Bus Interface Unit) provides the interface between the external system board and the processor's internal execution units. During a memory cycle, a memory location is selected through the address lines (A31-A3 and BE7# -BE0#). Data is passed from or to memory through the data lines (D63-D0).

Each instruction is read into 256-Byte Instruction Line Cache. The Cache Unit stores the most recently used data and instructions to allow fast access to the information by the Integer Unit and FPU.

The CPU core requests instructions from the Cache Unit. The received integer instructions are decoded by either the X or Y processing pipelines within the superpipelined integer unit. If the instruction is a MMX or FPU instruction it is passed to the floating point unit for processing.

As required data is fetched from the 64-KByte unified cache. If the data is not in the cache it is accessed via the bus interface unit from main memory.

The Memory Management Unit calculates physical addresses including addresses based on paging.

Physical addresses are calculated by the Memory Management Unit and passed to the Cache Unit and the Bus Interface Unit (BIU).

1.2 Integer Unit

The Integer Unit (Figure 1-1) provides parallel instruction execution using two seven-stage integer pipelines. Each of the two pipelines, X and Y, can process several instructions simultaneously.

The Integer Unit consists of the following pipeline stages:

- Instruction Fetch (IF)
- Instruction Decode 1 (ID1)
- Instruction Decode 2 (ID2)
- Address Calculation 1 (AC1)
- Address Calculation 2 (AC2)
- Execute (EX)
- Write-Back (WB)

The instruction decode and address calculation functions are both divided into superpipelined stages.

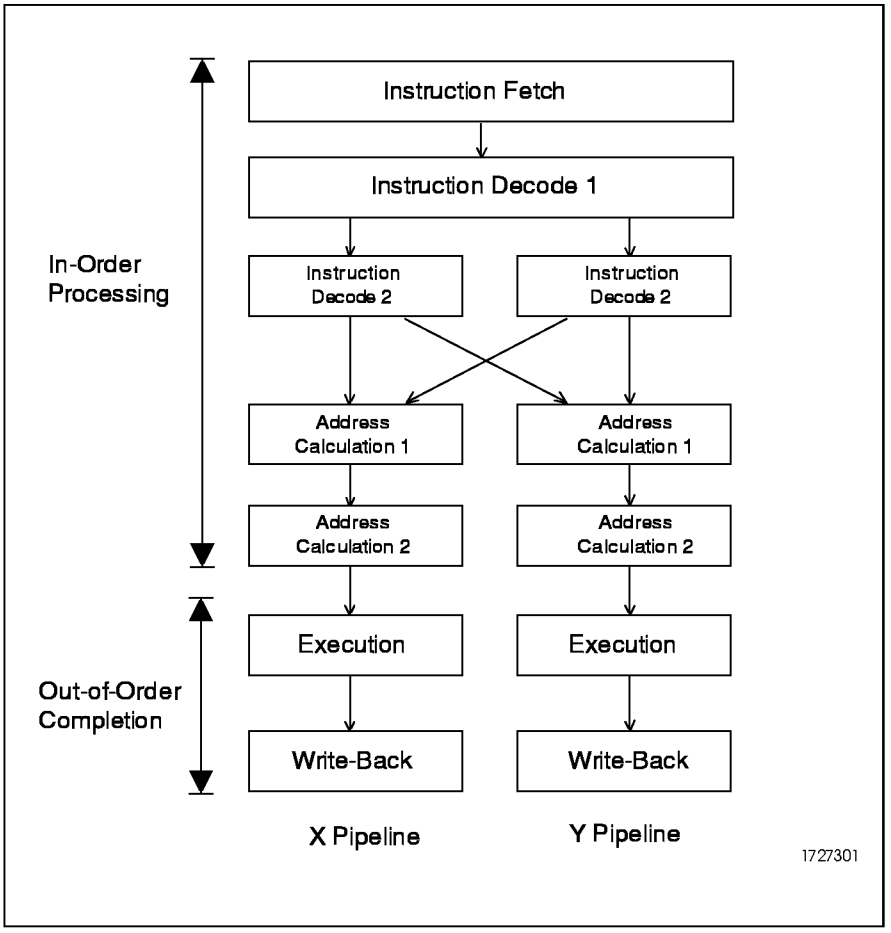


Figure 1-1. Integer Unit

1.2.1 Pipeline Stages

The Instruction Fetch (IF) stage, shared by both the X and Y pipelines, fetches 16 bytes of code from the cache unit in a single clock cycle. Within this section, the code stream is checked for any branch instructions that could affect normal program sequencing.

If an unconditional or conditional branch is detected, branch prediction logic within the IF stage generates a predicted target address for the instruction. The IF stage then begins fetching instructions at the predicted address.

The superpipelined Instruction Decode function contains the ID1 and ID2 stages. ID1, shared by both pipelines, evaluates the code stream provided by the IF stage and determines the number of bytes in each instruction. Up to two instructions per clock are delivered to the ID2 stages, one in each pipeline.

The ID2 stages decode instructions and send the decoded instructions to either the X or Y pipeline for execution. The particular pipeline is chosen, based on which instructions are already in each pipeline and how fast they are expected to flow through the remaining pipeline stages.

The Address Calculation function contains two stages, AC1 and AC2. If the instruction refers to a memory operand, the AC1 calculates a linear memory address for the instruction.

The AC2 stage performs any required memory management functions, cache accesses, and register file accesses. If a floating point instruction is detected by AC2, the instruction is sent to the FPU for processing.

The Execute (EX) stage executes instructions using the operands provided by the address calculation stage.

The Write-Back (WB) stage is the last IU stage. The WB stage stores execution results either to a register file within the IU or to a write buffer in the cache control unit.

1.2.2 Out-of-Order Processing

If an instruction executes faster than the previous instruction in the other pipeline, the instructions may complete out of order. All instructions are processed in order, up to the EX stage. While in the EX and WB stages, instructions may be completed out of order.

If there is a data dependency between two instructions, the necessary hardware interlocks are enforced to ensure correct program execution. Even though instructions may complete out of order, exceptions and writes resulting from the instructions are always issued in program order.

1.2.3 Pipeline Selection

In most cases, instructions are processed in either pipeline and without pairing constraints on the instructions. However, certain instructions are processed only in the X pipeline:

- Branch instructions
- Floating point instructions
- Exclusive instructions

Branch and floating point instructions may be paired with a second instruction in the Y pipeline.

Exclusive Instructions cannot be paired with instructions in the Y pipeline. These instructions typically require multiple memory accesses. Although exclusive instructions may not be paired, hardware from both pipelines is used to accelerate instruction completion. Listed below are the M II CPU exclusive instruction types:

- Protected mode segment loads
- Special register accesses (Control, Debug, and Test Registers)
- String instructions
- Multiply and divide
- I/O port accesses
- Push all (PUSHA) and pop all (POPA)
- Intersegment jumps, calls, and returns

1.2.4 Data Dependency Solutions

When two instructions that are executing in parallel require access to the same data or register, one of the following types of data dependencies may occur:

- Read-After-Write (RAW)
- Write-After-Read (WAR)
- Write-After-Write (WAW)

Data dependencies typically force serialized execution of instructions. However, the M II CPU implements three mechanisms that allow parallel execution of instructions containing data dependencies:

- Register Renaming
- Data Forwarding
- Data Bypassing

The following sections provide detailed examples of these mechanisms.

1.2.4.1 Register Renaming

The M II CPU contains 32 physical general purpose registers. Each of the 32 registers in the register file can be temporarily assigned as one of the general purpose registers defined by the x86 architecture (EAX, EBX, ECX, EDX, ESI, EDI, EBP, and ESP). For each register write operation a new physical register is selected to allow previous data to be retained temporarily. Register renaming effectively removes all WAW and WAR dependencies. The programmer does not have to consider register renaming as register renaming is completely transparent to both the operating system and application software.

Example #1 – Register Renaming Eliminates Write-After-Read (WAR) Dependency

A WAR dependency exists when the first in a pair of instructions reads a logical register, and the second instruction writes to the same logical register. This type of dependency is illustrated by the pair of instructions shown below:

X PIPE	Y PIPE
(1) MOV BX, AX	(2) ADD AX, CX
BX ← AX	AX ← AX + CX

Note: In this and the following examples the original instruction order is shown in parentheses.

In the absence of register renaming, the ADD instruction in the Y pipe would have to be stalled to allow the MOV instruction in the X pipe to read the AX register.

The M II CPU, however, avoids the Y pipe stall (Table 1-2). As each instruction executes, the results are placed in new physical registers to avoid the possibility of overwriting a logical register value and to allow the two instructions to complete in parallel (or out of order) rather than in sequence.

Table 1-1. Register Renaming with WAR Dependency

Instruction	Physical Register Contents					Action	
	Reg0	Reg1	Reg2	Reg3	Reg4	Pipe	
(Initial)	AX	BX	CX				
MOV BX, AX	AX		CX	BX		X	Reg3 ← Reg0
ADD AX, CX			CX	BX	AX	Y	Reg4 ← Reg0 + Reg2

Note: The representation of the MOV and ADD instructions in the final column of Table 1-2 are completely independent.

Example #2 – Register Renaming Eliminates Write-After-Write (WAW) Dependency

A WAW dependency occurs when two consecutive instructions perform writes to the same logical register. This type of dependency is illustrated by the pair of instructions shown below:

X PIPE	Y PIPE
(1) ADD AX, BX	(2) MOV AX, [mem]
AX ← AX + BX	AX ← [mem]

Without register renaming, the MOV instruction in the Y pipe would have to be stalled to guarantee that the ADD instruction in the X pipe would write its results to the AX register first.

The M II CPU uses register renaming and avoids the Y pipe stall. The contents of the AX and BX registers are placed in physical registers (Table 1-3). As each instruction executes, the results are placed in new physical registers to avoid the possibility of overwriting a logical register value and to allow the two instructions to complete in parallel (or out of order) rather than in sequence.

Table 1-2. Register Renaming with WAW Dependency

Instruction	Physical Register Contents				Action	
	Reg0	Reg1	Reg2	Reg3	Pipe	
(Initial)	AX	BX				
ADD AX, BX		BX	AX		X	Reg2 ← Reg0 + Reg1
MOV AX, [mem]		BX		AX	Y	Reg3 ← [mem]

Note: All subsequent reads of the logical register AX will refer to Reg 3, the result of the MOV instruction.

1.2.4.2 Data Forwarding

Register renaming alone cannot remove RAW dependencies. The M II CPU uses two types of data forwarding in conjunction with register renaming to eliminate RAW dependencies:

- Operand Forwarding
- Result Forwarding

Operand forwarding takes place when the first in a pair of instructions performs a move from register or memory, and the data that is read by the first instruction is required by the second instruction. The M II CPU performs the read operation and makes the data read available to both instructions simultaneously.

Result forwarding takes place when the first in a pair of instructions performs an operation (such as an ADD) and the result is required by the second instruction to perform a move to a register or memory. The M II CPU performs the required operation and stores the results of the operation to the destination of both instructions simultaneously.

Example #3 – Operand Forwarding Eliminates Read-After-Write (RAW) Dependency

A RAW dependency occurs when the first in a pair of instructions performs a write, and the second instruction reads the same register. This type of dependency is illustrated by the pair of instructions shown below in the X and Y pipelines:

X PIPE	Y PIPE
(1) MOV AX, [mem]	(2) ADD BX, AX
AX ← [mem]	BX ← AX + BX

The M II CPU uses operand forwarding and avoids a Y pipe stall (Table 1-4). Operand forwarding allows simultaneous execution of both instructions by first reading memory and then making the results available to both pipelines in parallel.

Table 1-3. Example of Operand Forwarding

Instruction	Physical Register Contents				Action	
	Reg0	Reg1	Reg2	Reg3	Pipe	
(Initial)	AX	BX				
MOV AX, [mem]		BX	AX		X	Reg2 ← [mem]
ADD BX, AX			AX	BX	Y	Reg3 ← [mem] + Reg1

Operand forwarding can only occur if the first instruction does not modify its source data. In other words, the instruction is a move type instruction (for example, MOV, POP, LEA). Operand forwarding occurs for both register and memory operands. The size of the first instruction destination and the second instruction source must match.

Example #4 – Result Forwarding Eliminates Read-After-Write (RAW) Dependency

In this example, a RAW dependency occurs when the first in a pair of instructions performs a write, and the second instruction reads the same register. This dependency is illustrated by the pair of instructions in the X and Y pipelines, as shown below:

X PIPE	Y PIPE
(1) ADD AX, BX	(2) MOV [mem], AX
AX ← AX + BX	[mem] ← AX

The M II CPU uses result forwarding and avoids a Y pipe stall (Table 1-5). Instead of transferring the contents of the AX register to memory, the result of the previous ADD instruction (Reg0 + Reg1) is written directly to memory, thereby saving a clock cycle.

Table 1-4. Result Forwarding Example

Instruction	Physical Register Contents			Action	
	Reg0	Reg1	Reg2	Pipe	
(Initial)	AX	BX			
ADD AX, BX		BX	AX	X	Reg2 ← Reg0 + Reg1
MOV [mem], AX		BX	AX	Y	[mem] ← Reg0 + Reg1

The second instruction must be a move instruction and the destination of the second instruction may be either a register or memory.

1.2.4.3 Data Bypassing

In addition to register renaming and data forwarding, the M II CPU implements a third data dependency-resolution technique called data bypassing. Data bypassing reduces the performance penalty of those memory data RAW dependencies that cannot be eliminated by data forwarding.

Data bypassing is implemented when the first in a pair of instructions writes to memory and the second instruction reads the same data from memory. The M II CPU retains the data from the first instruction and passes it to the second instruction, thereby eliminating a memory read cycle. Data bypassing only occurs for cacheable memory locations.

Example #1 - Data Bypassing with Read-After-Write (RAW) Dependency

In this example, a RAW dependency occurs when the first in a pair of instructions performs a write to memory and the second instruction reads the same memory location. This dependency is illustrated by the pair of instructions in the X and Y pipelines as shown below:

X PIPE	Y PIPE
(1) ADD [mem], AX	(2) SUB BX, [mem]
[mem] ← [mem] + AX	BX ← BX - [mem]

The M II CPU uses data bypassing and stalls the Y pipe for only one clock by eliminating the Y pipe's memory read cycle (Table 1-6). Instead of reading memory in the Y pipe, the result of the previous instruction ([mem] + Reg0) is used to subtract from Reg1, thereby saving a memory access cycle.

Table 1-5. Example of Data Bypassing

Instruction	Physical Register Contents			Action	
	Reg0	Reg1	Reg2	Pipe	
(Initial)	AX	BX			
ADD [mem], AX	AX	BX		X	[mem] ← [mem] + Reg0
SUB BX, [mem]	AX		BX	Y	Reg2 ← Reg1 - {[mem] + Reg0}

1.2.5 Branch Control

Branch instructions occur on average every four to six instructions in x86-compatible programs. When the normal sequential flow of a program changes due to a branch instruction, the pipeline stages may stall while waiting for the CPU to calculate, retrieve, and decode the new instruction stream. The M II CPU minimizes the performance degradation and latency of branch instructions through the use of branch prediction and speculative execution.

1.2.5.1 Branch Prediction

The M II CPU uses a 512-entry, 4-way set associative Branch Target Buffer (BTB) to store branch target addresses. The M II CPU has 1024-entry branch history table. During the fetch stage, the instruction stream is checked for the presence of branch instructions. If an unconditional branch instruction is encountered, the M II CPU accesses the BTB to check for the branch instruction's target address. If the branch instruction's target address is found in the BTB, the M II CPU begins fetching at the target address specified by the BTB.

In case of conditional branches, the BTB also provides history information to indicate whether the branch is more likely to be taken or not taken. If the conditional branch instruction is found in the BTB, the M II CPU begins fetching instructions at the predicted target address. If the conditional branch misses in the BTB, the M II CPU predicts that the branch will not be taken, and instruction fetching continues with the next sequential instruction.

The decision to fetch the taken or not taken target address is based on a four-state branch prediction algorithm.

Once fetched, a conditional branch instruction is first decoded and then dispatched to the X pipeline only. The conditional branch instruction proceeds through the X pipeline and is then resolved in either the EX stage or the WB stage. The conditional branch is resolved in the EX stage, if the instruction responsible for setting the condition codes is completed prior to the execution of the branch. If the instruction that sets the condition codes is executed in parallel with the branch, the conditional branch instruction is resolved in the WB stage.

Correctly predicted branch instructions execute in a single core clock. If resolution of a branch indicates that a misprediction has occurred, the M II CPU flushes the pipeline and starts fetching from the correct target address. The M II CPU prefetches both the predicted and the non-predicted path for each conditional branch, thereby eliminating the cache access cycle on a misprediction. If the branch is resolved in the EX stage, the resulting misprediction latency is four cycles. If the branch is resolved in the WB stage, the latency is five cycles.

Since the target address of return (RET) instructions is dynamic rather than static, the M II CPU caches target addresses for RET instructions in an eight-entry return stack rather than in the BTB. The return address is pushed on the return stack during a CALL instruction and popped during the corresponding RET instruction.

1.2.5.2 Speculative Execution

The M II CPU is capable of speculative execution following a floating point instruction or predicted branch. Speculative execution allows the pipelines to continuously execute instructions following a branch without stalling the pipelines waiting for branch resolution. The same mechanism is used to execute floating point instructions (see Section 1.6) in parallel with integer instructions.

The M II CPU is capable of up to four levels of speculation (i.e., combinations of four conditional branches and floating point operations). After generating the fetch address using branch prediction, the CPU checkpoints the machine state (registers, flags, and processor environment), increments the speculation level counter, and begins operating on the predicted instruction stream.

Once the branch instruction is resolved, the CPU decreases the speculation level. For a correctly predicted branch, the status of the checkpointed resources is cleared. For a branch misprediction, the M II processor generates the correct fetch address and uses the checkpointed values to restore the machine state in a single clock.

In order to maintain compatibility, writes that result from speculatively executed instructions are not permitted to update the cache or external memory until the appropriate branch is resolved. Speculative execution continues until one of the following conditions occurs:

1) A branch or floating point operation is decoded and the speculation level is already at four.
2) An exception or a fault occurs.
3) The write buffers are full.
4) An attempt is made to modify a non-checkpointed resource (i.e., segment registers, system flags).

1.3 Cache Units

The M II CPU employs two caches, the Unified Cache and the Instruction Line Cache (Figure 1-2, Page 1-15). The main cache is a 4-way set-associative 64-KByte unified cache. The unified cache provides a higher hit rate than using equal-sized separate data and instruction caches. While in Cyrix SMM mode both SMM code and data are cacheable.

The instruction line cache is a fully associative 256-byte cache. This cache avoids excessive conflicts between code and data accesses in the unified cache.

1.3.1 Unified Cache

The 64-KByte unified write-back cache functions as the primary data cache and as the secondary instruction cache. Configured as a four-way set-associative cache, the cache stores up to 64 KBytes of code and data in 2048 lines. The cache is dual-ported and allows any

two of the following operations to occur in parallel:

- Code fetch
- Data read (X pipe, Y pipeline or FPU)
- Data write (X pipe, Y pipeline or FPU)

The unified cache uses a pseudo-LRU replacement algorithm and can be configured to allocate new lines on read misses only or on read and write misses.

1.3.2 Instruction Line Cache

The fully associative 256-byte instruction line cache serves as the primary instruction cache. The instruction line cache is filled from the unified cache through the data bus. Fetches from the integer unit that hit in the instruction line cache do not access the unified cache. If an instruction line cache miss occurs, the instruction line data from the unified cache is transferred to the instruction line cache and the integer unit, simultaneously.

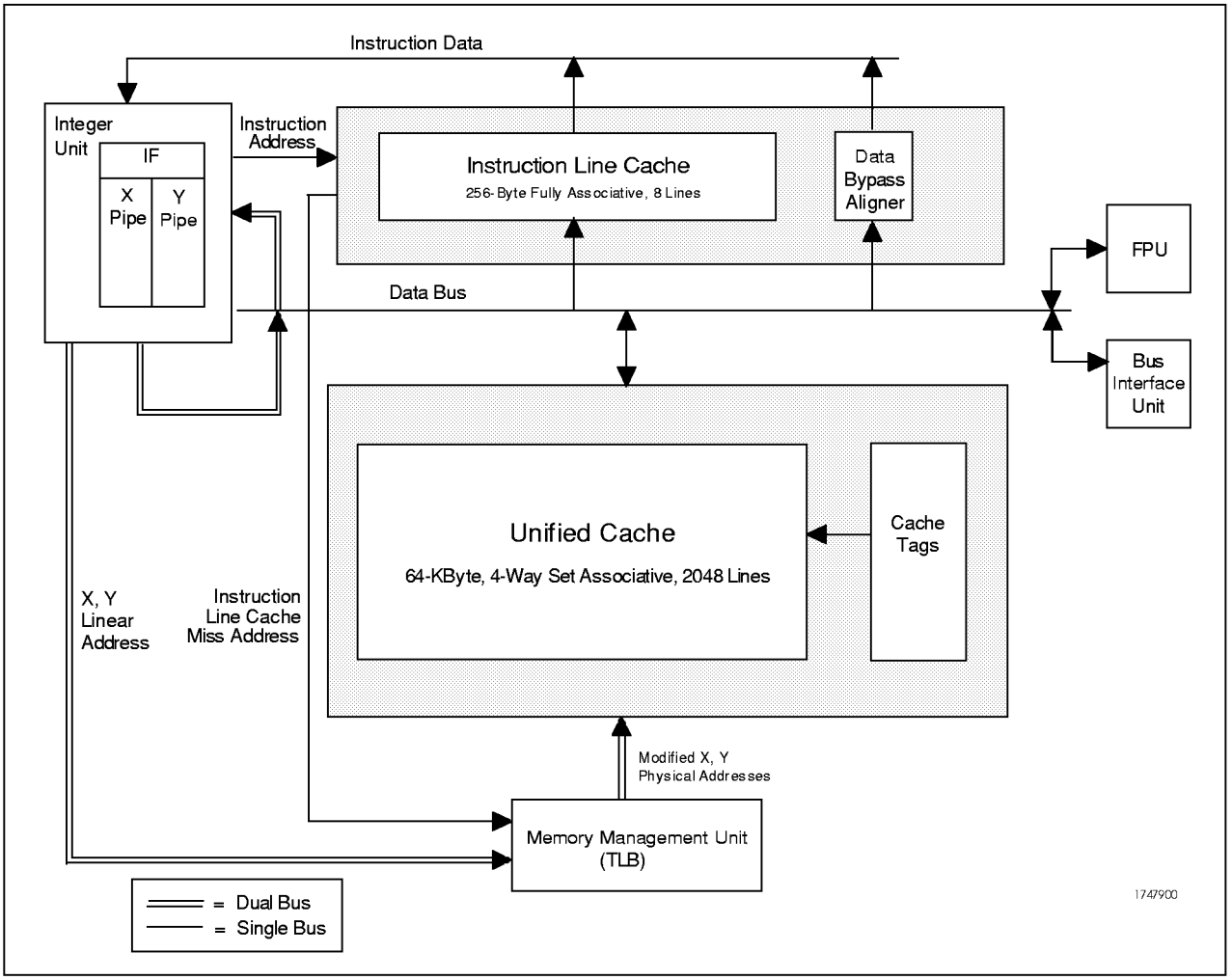


Figure 1-2. Cache Unit Operations

The instruction line cache uses a pseudo-LRU replacement algorithm. To ensure proper operation in the case of self-modifying code, any writes to the unified cache are checked against the contents of the instruction line cache. If a hit occurs in the instruction line cache, the appropriate line is invalidated.

1.4 Memory Management Unit

The Memory Management Unit (MMU), shown in Figure 1-3, translates the linear address supplied by the IU into a physical address to be used by the unified cache and the bus interface. Memory management procedures are x86 compatible, adhering to standard paging mechanisms.

Within the M II CPU there are two TLBs, the main L1 TLB and the larger L2 TLB. The 16-entry L1 TLB is direct mapped and holds 42 lines. The 384-entry L2 TLB is 6-way associative and hold 384 lines. The DTE is located in memory.

Scratch Pad Cache Memory

The M II CPU has the capability to "lock down" lines in the L1 cache on a line by line basis. Locked down lines are treated as private memory for use by the CPU. Locked down memory does not participate in hardware--cache coherency protocols.

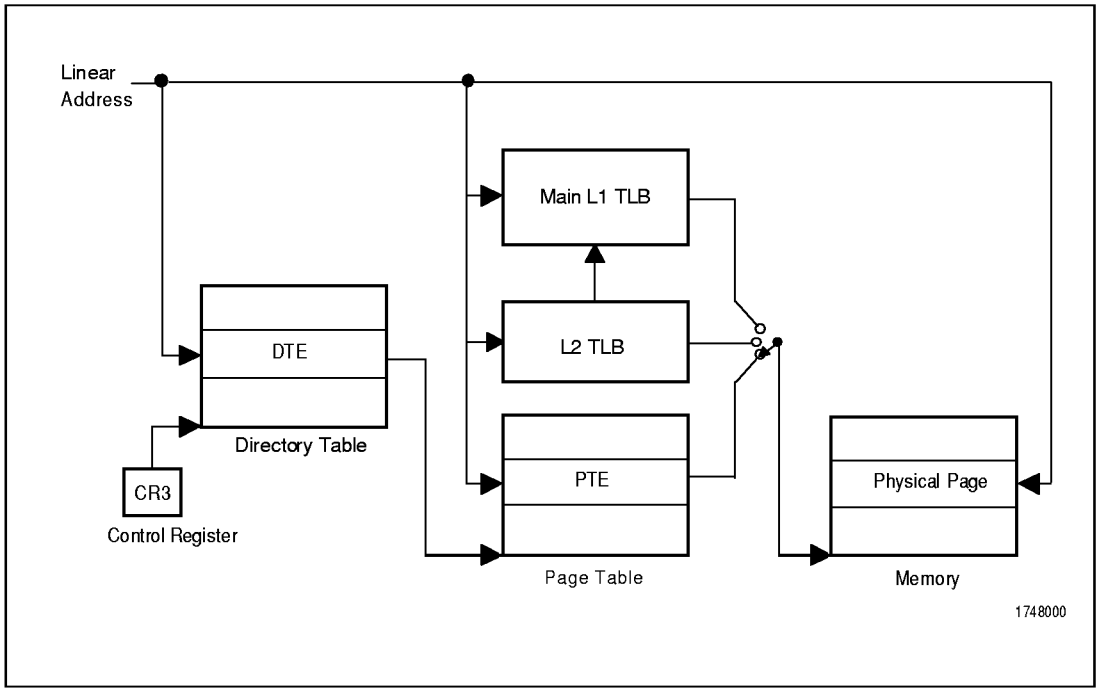


Figure 1-3. Paging Mechanism within the Memory Management Unit

Cache locking is controlled through use of the RDMSR and WRMSR instructions.

1.5 Floating Point Unit

The M II Floating Point Unit (FPU) processes floating point and MMX instructions. The FPU interfaces to the integer unit and the cache unit through a 64-bit bus. The M II FPU is x87 instruction set compatible and adheres to the IEEE-754 standard. Since most applications contain FPU instructions mixed with integer instructions, the M II FPU achieves high performance by completing integer and FPU operations in parallel.

FPU ParallelExecution

The M II CPU executes integer instructions in parallel with FPU instructions. Integer instructions may complete out of order with respect to the FPU instructions. The M II CPU maintains x86 compatibility by signaling exceptions and issuing write cycles in program order.

As previously discussed, FPU instructions are always dispatched to the integer unit's X pipeline. The address calculation stage of the X pipeline checks for memory management exceptions and accesses memory operands used by the FPU. If no exceptions are detected, the M II CPU checkpoints the state of the CPU and, during AC2, dispatches the floating point instruction to the FPU instruction queue. The M II CPU can then complete any subsequent integer instructions speculatively and out of order relative to the FPU instruction and relative to any potential FPU exceptions which may occur.

As additional FPU instructions enter the pipeline, the M II CPU dispatches up to four FPU instructions to the FPU instruction queue. The M II CPU continues executing speculatively and out of order, relative to the FPU queue, until the M II CPU encounters one of the conditions that causes speculative execution to halt. As the FPU completes instructions, the speculation level decreases and the checkpointed resources are available for reuse in subsequent operations. The M II FPU also uses a set of six write buffers to prevent stalls due to speculative writes.

1.6 Bus Interface Unit

The Bus Interface Unit (BIU) provides the signals and timing required by external circuitry. The signal descriptions and bus interface timing information is provided in Chapters 3 and 4 of this manual.

2.4.4 M II Configuration Registers

The M II configuration registers are used to enable features in the M II CPU. These registers assign non-cached memory areas, set up SMM, provide CPU identification information and control various features such as cache write policy, and bus locking control. There are four groups of registers within the M II configuration register set:

- 7 Configuration Control Registers (CCRx)
- 8 Address Region Registers (ARRx)
- 8 Region Control Registers (RCRx)

Access to the configuration registers is achieved by writing the register index number for the configuration register to I/O port 22h. I/O port 23h is then used for data transfer.

Each I/O port 23h data transfer must be preceded by a valid I/O port 22h register index selection. Otherwise, the current 22h, and the second and later I/O port 23h operations communicate through the I/O port to produce external I/O cycles. All reads from I/O port 22h produce external I/O cycles. Accesses that hit within the on-chip configuration registers do not generate external I/O cycles.

After reset, configuration registers with indexes C0-CFh and FC-FFh are accessible. To prevent potential conflicts with other devices which may use ports 22 and 23h to access their registers, the remaining registers (indexes D0-FBh) are accessible only if the MAPEN(3-0) bits in CCR3 are set to 1h. See Figure 2-16 (Page 2-29) for more information on the MAPEN(3-0) bit locations.

If MAPEN[3-0] = 1h, any access to indexes in the range 00-FFh will not create external I/O bus cycles. Registers with indexes C0-CFh, FC- FFh are accessible regardless of the state of MAPEN[3-0]. If the register index number is outside the C0-CFh or FC-FFh ranges, and MAPEN[3-0] are set to 0h, external I/O bus cycles occur. Table 2-11 (Page 2-25) lists the MAPEN[3-0] values required to access each M II configuration register. All bits in the configuration registers are initialized to zero following reset unless specified otherwise.

2.4.4.1 Configuration Control Registers

(CCR0 - CCR6) control several functions, including non-cacheable memory, write-back regions, and SMM features. A list of the configuration registers is listed in Table 2-11 (Page 2-25). The configuration registers are described in greater detail in the following pages.

Table 2-11. M II CPU Configuration Registers

REGISTER NAME	ACRONYM	REGISTER INDEX	WIDTH (Bits)	MAPEN VALUE NEEDED FOR ACCESS
Configuration Control 0	CCR0	C0h	8	x
Configuration Control 1	CCR1	C1h	8	x
Configuration Control 2	CCR2	C2h	8	x
Configuration Control 3	CCR3	C3h	8	x
Configuration Control 4	CCR4	E8h	8	1
Configuration Control 5	CCR5	E9h	8	1
Configuration Control 6	CCR6	EAh	8	1
Address Region 0	ARR0	C4h - C6h	24	x
Address Region 1	ARR1	C7h - C9h	24	x
Address Region 2	ARR2	CAh - CCh	24	x
Address Region 3	ARR3	CDh - CFh	24	x
Address Region 4	ARR4	D0h - D2h	24	1
Address Region 5	ARR5	D3h - D5h	24	1
Address Region 6	ARR6	D6h - D8h	24	1
Address Region 7	ARR7	D9h - DBh	24	1
Region Control 0	RCR0	DCh	8	1
Region Control 1	RCR1	DDh	8	1
Region Control 2	RCR2	DEh	8	1
Region Control 3	RCR3	DFh	8	1
Region Control 4	RCR4	E0h	8	1
Region Control 5	RCR5	E1h	8	1
Region Control 6	RCR6	E2h	8	1
Region Control 7	RCR7	E3h	8	1

Note: x = Don't Care

7	6	5	4	3	2	1	0
Reserved	*Reserved*	*Reserved*	*Reserved*	*Reserved*	*Reserved*	NC1	*Reserved*

Figure 2-13. M II Configuration Control Register 0 (CCR0)

Table 2-12. CCR0 Bit Definitions

BIT POSITION	NAME	DESCRIPTION
1	NC1	No Cache 640 KByte - 1 MByte If = 1: Address region 640 KByte to 1 MByte is non-cacheable. If = 0: Address region 640 KByte to 1 MByte is cacheable.

Note: Bits 0, 2 through 7 are reserved.

7	6	5	4	3	2	1	0
SM3	*Reserved*	*Reserved*	NO_LOCK	*Reserved*	SMAC	USE_SMI	*Reserved*

Figure 2-14. M II Configuration Control Register 1 (CCR1)

Table 2-13. CCR1 Bit Definitions

BIT POSITION	NAME	DESCRIPTION
7	SM3	SMM Address Space Address Region 3 If = 1: Address Region 3 is designated as SMM address space.
4	NO_LOCK	Negate LOCK# If = 1: All bus cycles are issued with LOCK# pin negated except page table accesses and interrupt acknowledge cycles. Interrupt acknowledge cycles are executed as locked cycles even though LOCK# is negated. With NO_LOCK set, previously noncacheable locked cycles are executed as unlocked cycles and therefore, may be cached. This results in higher performance. Refer to Region Control Registers for information on eliminating locked CPU bus cycles only in specific address regions.
2	SMAC	System Management Memory Access If = 1: Any access to addresses within the SMM address space, access system management memory instead of main memory. SMI# input is ignored. Used when initializing or testing SMM memory. If = 0: No effect on access.
1	USE_SMI	Enable SMM and SMIACT# Pins If = 1: SMI# and SMIACT# pins are enabled. If = 0: SMI# pin ignored and SMIACT# pin is driven inactive.

Note: Bits 0, 3, 5 and 6 are reserved.

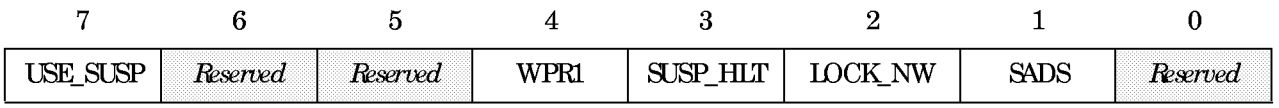


Figure 2-15. M II Configuration Control Register 2 (CCR2)

Table 2-14. CCR2 Bit Definitions

BIT POSITION	NAME	DESCRIPTION
7	USE_SUSP	Use Suspend Mode (Enable Suspend Pins) If = 1: SUSP# and SUSPA# pins are enabled. If = 0: SUSP# pin is ignored and SUSPA# pin floats.
4	WPR1	Write-Protect Region 1 If = 1: Designates any cacheable accesses in 640 KByte to 1 MByte address region are write protected.
3	SUSP_HLT	Suspend on Halt If = 1: Execution of the HLT instruction causes the CPU to enter low power suspend mode.
2	LOCK_NW	Lock NW If = 1: NW bit in CR0 becomes read only and the CPU ignores any writes to the NW bit. If = 0: NW bit in CR0 can be modified.
1	SADS	If = 1: CPU inserts an idle cycle following sampling of BRDY# and inserts an idle cycle prior to asserting ADS#

Note: Bits 0, 5 and 6 are reserved.

7	6	5	4	3	2	1	0
MAPEN3	MAPEN2	MAPEN1	MAPEN0	*Reserved*	LINBRST	NMI_EN	SMI_LOCK

Figure 2-16. M II Configuration Control Register 3 (CCR3)

Table 2-15. CCR3 Bit Definitions

BIT POSITION	NAME	DESCRIPTION
7 - 4	MAPEN(3-0)	MAP Enable If = 1h: All configuration registers are accessible. If = 0h: Only configuration registers with indexes C0-CFh, FEh and FFh are accessible.
2	LINBRST	If = 1: Use linear address sequence during burst cycles. If = 0: Use "1 + 4" address sequence during burst cycles. The "1 + 4" address sequence is compatible with Pentium's burst address sequence.
1	NMI_EN	NMI Enable If = 1: NMI interrupt is recognized while servicing an SMI interrupt. NMI_EN should be set only while in SMM, after the appropriate SMI interrupt service routine has been setup.
0	SMI_LOCK	SMI Lock If = 1: The following SMM configuration bits can only be modified while in an SMI service routine: CCR1: USE_SMI, SMAC, SM3 CCR3: NMI_EN CCR6: N, SMM_MODE ARR3: Starting address and block size. Once set, the features locked by SMI_LOCK cannot be unlocked until the RESET pin is asserted.

Note: Bit 3 is reserved.

7	6	5	4	3	2	1	0
CPUID	*Reserved*	*Reserved*	*Reserved*	*Reserved*	IORT2	IORT1	IORT

Figure 2-17. M II Configuration Control Register 4 (CCR4)

Table 2-16. CCR4 Bit Definitions

BIT POSITION	NAME	DESCRIPTION
7	CPUID	Enable CPUID instruction. If = 1: the ID bit in the EFLAGS register can be modified and execution of the CPUID instruction occurs as documented in section 6.3. If = 0: the ID bit in the EFLAGS register can not be modified and execution of the CPUID instruction causes an invalid opcode exception.
2 - 0	IORT(2-0)	I/O Recovery Time Specifies the minimum number of bus clocks between I/O accesses: 0h = 1 clock delay 1h = 2 clock delay 2h = 4 clock delay 3h = 8 clock delay 4h = 16 clock delay 5h = 32 clock delay (default value after RESET) 6h = 64 clock delay 7h = no delay

Note: Bits 3 - 6 are reserved.

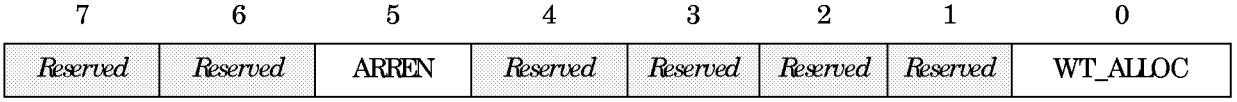


Figure 2-18. M II Configuration Control Register 5 (CCR5)

Table 2-17. CCR5 Bit Definitions

BIT POSITION	NAME	DESCRIPTION
5	ARREN	Enable ARR Registers If = 1: Enables all ARR registers. If = 0: Disables the ARR registers. If SM3 is set, ARR3 is enabled regardless of the setting of ARREN.
0	WT_ALLOC	Write-Through Allocate If = 1: New cache lines are allocated for read and write misses. If = 0: New cache lines are allocated only for read misses.

Note: Bits 1 - 3 and 6 - 7 are reserved.

7	6	5	4	3	2	1	0
Reserved	N	*Reserved*	*Reserved*	*Reserved*	*Reserved*	WP_ARR3	SMM_MODE

Figure 2-19. M II Configuration Control Register 6 (CCR6)

Table 2-18. CCR6 Bit Definitions

BIT POSITION	NAME	DESCRIPTION
6	N	Nested SMI Enable bit: If operating in Cyrix enhanced SMM mode and: If = 1: Enables nesting of SMI's If = 0: Disable nesting of SMI's. This bit is automatically CLEARED upon entry to every SMM routine and is SET upon every RSM. Therefore enabling/disabling of nested SMI can only be done while operating in SMM mode.
1	WP_ARR3	If = 1: Memory region defined by ARR3 is write protected when operating outside of SMM mode. If = 0: Disable write protection for memory region defined by ARR3. Reset State = 0.
0	SMM_MODE	If = 1: Enables Cyrix Enhanced SMM mode. If = 0: Disables Cyrix Enhanced SMM mode.

Note: Bit 1 is reserved.

2.4.4.2 Memory Address Region Registers

The Address Region Registers (ARR0 - ARR7) are paired with the Region Control Registers (RCR0-RCR7) and specify up to eight memory address regions. Using ARR/RCR pairs, these regions can be designated as non-cacheable, write through, write locking and weak locking. Register pairs ARR7/ RCR7 are unique and can define attributes for all of system main memory.

The 24-bit ARR registers (Figure 2-20) are divided into the 20-bit BASE ADDRESS and 4-bit SIZE fields. The fields define the size and base addresses for the memory regions. The base address must be on a block size boundary. For example, if a 128-KByte block is used, the base addresses are allow to be 0, 128 and, 256 KBytes, and so forth.

The meaning of the SIZE fields are listed in Table 2-20. (Page 2-34). If the SIZE field is zero, the address region is zero sized and thus disabled.

The ARR registers are accessed using I/O ports 22h and 23h. To read or write to a complete ARR register, three I/O port cycles are required. Each byte is assigned an index value (Table 2-19, Page 2-34).

A region is noncacheable, if defined by ARR/RCR pair even if KEN# is active.

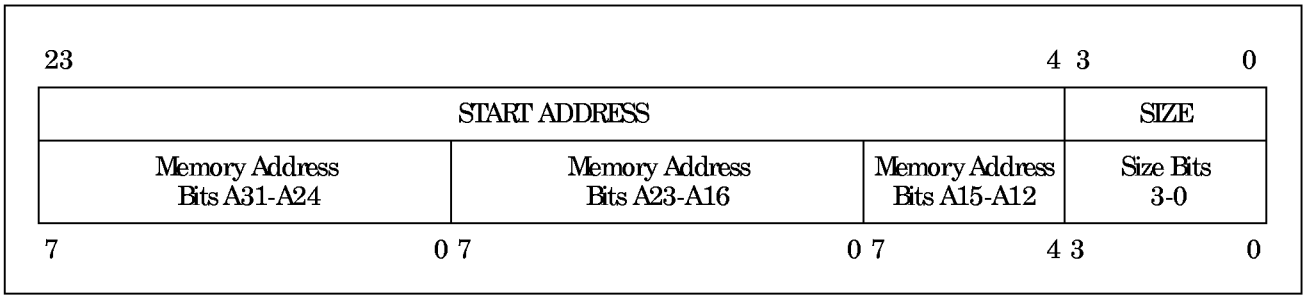


Figure 2-20. Address Region Registers (ARR0 - ARR7)

Table 2-19. ARR0 - ARR7 Register Index Assignments

ARR Register	Register Bytes		
	Base Memory Address Field A31 - A24	Base Memory Address Field A23 - A16	Base Memory Address Field A15 - A12, and SIZE Field
ARR0	C4h	C5h	C6h
ARR1	C7h	C8h	C9h
ARR2	CAh	CBh	CCh
ARR3	CDh	CEh	CFh
ARR4	D0h	D1h	D2h
ARR5	D3h	D4h	D5h
ARR6	D6h	D7h	D8h
ARR7	D9h	DAh	DBh

Table 2-20. SIZE Field Bit Definition

SIZE (3-0)	BLOCK SIZE ARR0-6	BLOCK SIZE ARR7
0h	Disabled	Disabled
1h	4 KBytes	256 KBytes
2h	8 KBytes	512 KBytes
3h	16 KBytes	1 MBytes
4h	32 KBytes	2 MBytes
5h	64 KBytes	4 MBytes
6h	128 KBytes	8 MBytes
7h	256 KBytes	16 MBytes

SIZE (3-0)	BLOCK SIZE ARR0-6	BLOCK SIZE ARR7
8h	512 KBytes	32 MBytes
9h	1 MBytes	64 MBytes
Ah	2 MBytes	128 MBytes
Bh	4 MBytes	256 MBytes
Ch	8 MBytes	512 MBytes
Dh	16 MBytes	1 GBytes
Eh	32 MBytes	2 GBytes
Fh	4 GBytes	4 GBytes

2.4.4.3 Region Control Registers

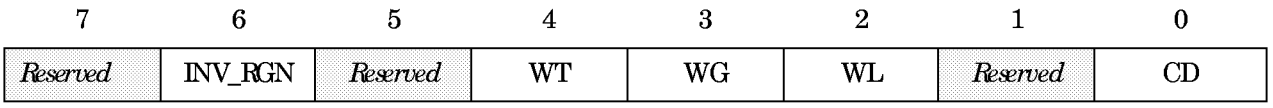
The Region Control Registers (RCR0 - RCR7) are paired with the Address Region Registers (ARR0 -ARR7). Each RCR register specifies the attributes associated with a particular address regions. These attributes include: cacheability, weak locking, write gathering, and cache write through policies.The bit definitions for the region control registers are shown in Figure 2-21 (Page 2-36) and in Table 2-21 (Page 2-36).

If an address is accessed that is not in a memory region defined by the ARRx registers, the following conditions will apply:

- If the memory address is cached, write-back is enabled if WB/WT# is returned high.
- Writes are not gathered
- Strong locking takes place
- The memory access is cached, if KEN# is returned asserted.

Overlapping Conditions Defined. If two regions specified by ARRx registers overlap and conflicting attributes are specified, the following attributes take precedence:

- Write-back is disabled
- Writes are not gathered
- Strong locking takes place
- The overlapping regions are non-cacheable.



7	6	5	4	3	2	1	0
Reserved	INV_RGN	*Reserved*	WT	WG	WL	*Reserved*	CD

Figure 2-21. Region Control Registers (RCR0-RCR7)

Table 2-21. RCR0-RCR7 Bit Definitions

BIT POSITION	NAME	DESCRIPTION
6	INV_RGN	Inverted Region. If =1, applies controls specified in RCRx to all memory addresses outside the region specified in corresponding ARR. Applicable to RCR0-RCR6 only.
4	WT	Write-Through. If =1, defines the address region as write-through instead of write-back.
3	WG	Write Gathering. If =1, enables write gathering for the associated address region.
2	WL	Weak Locking. If =1, enables weak locking for that address region.
0	CD	Cache Disable. If =1, defines the address region as non-cacheable.

Note: Bits 1, 5 and 7 are reserved.

Inverted Region (INV_RGN). Setting INV-RGN applies the controls in RCRx to all the memory addresses outside the specified address region ARRx. This bit effects RCR0-RCR6 and not RCR7.

Write Through (WT). Setting WT defines the address region as write-through instead of write-back, assuming the region is cacheable. Regions where system ROM are loaded (shadowed or not) should be defined as write-through. This bit works in conjunction with the CR0_NW and PWT bits and the WB/WT# pin to determine write-through or write-back cacheability.

Write Gathering (WG). Setting WG enables write gathering for the associated address region. Write gathering allows multiple byte, word, or Dword sequential address writes to accumulate in the on-chip write buffer. As instructions are executed, the results are placed in a series of output buffers. These buffers are gathered into the final output buffer.

When access is made to a non-sequential memory location or when the 8-byte buffer becomes full, the contents of the buffer are written on the external 64-bit data bus. Performance is enhanced by avoiding as many as seven memory write cycles.

WG should not be used on memory regions that are sensitive to write cycle gathering. WG can be enabled for both cacheable and non-cacheable regions.

Weak Locking (WL). Setting WL enables weak locking for the associated address region. During weak locking all bus cycles are issued with the LOCK# pin negated (except when page table access occur and during interrupt acknowledge cycles.)

Interrupt acknowledge cycles are executed as locked cycles even though LOCK# is negated. With WL set previously non-cacheable locked cycles are executed as unlocked cycles and therefore, may be cached, resulting in higher CPU performance.

Note that the NO_LOCK bit globally performs the same function that the WL bit performs on a single address region. The NO_LOCK bit of CCR1 enables weak locking for the entire address space. The WL bit allows weak locking only for specific address regions. WL is independent of the cacheability of the address region.

Cache Disable (CD). Cache Disable - If set, defines the address region as non-cacheable. This bit works in conjunction with the CR0_CD and PCD bits and the KEN# pin to determine line cacheability. Whenever possible, the ARR/RCR combination should be used to define non-cacheable regions rather than using external address decoding and driving the KEN# pin as the M II can better utilize its advanced techniques for eliminating data dependencies and resource conflicts with non-cacheable regions defined on-chip.

2.5 Model Specific Registers

The CPU contains several Model Specific Registers (MSRs) that provide time stamp, performance monitoring and counter event functions. Access to a specific MSR through an index value in the ECX register as shown in Table 2-22 below.

Table 2-22. Machine Specific Register

REGISTER DESCRIPTION	ECX VALUE
Test Data	3h
Test Address	4h
Command/Status	5h
Time Stamp Counter (TSC)	10h
Counter Event Selection and Control Register	11h
Performance Counter #0	12h
Performance Counter #1	13h

The MSR registers can be read using the RDMSR instruction, opcode 0F32h. During an MSR register read, the contents of the particular MSR register, specified by the ECX register, is loaded into the EDX:EAX registers.

The MSR registers can be written using the WRMSR instruction, opcode 0F30h. During a MSR register write the contents of EDX:EAX are loaded into the MSR register specified in the ECX register.

The RDMSR and WRMSR instructions are privileged instructions and are also used to setup scratch pad lock (Page 2-61).

2.6 Time Stamp Counter

The Time Stamp Counter (TSC) Register MSR(10) is a 64-bit counter that counts the internal CPU clock cycles since the last reset. The TSC uses a continous CPU core clock and will continue to count clock cycles even when the M II is suspend mode or shutdown.

The TSC can be accessed using the RDMSR and WRMSR instructions. In addition, the TSC can be read using the RDTSC instruction, opcode 0F31h. The RDTSC instruction loads the contents of the TSC into EDX:EAX. The use of the RDTSC instruction is restricted by the Time Stamp Disable, (TSD) flag in CR4. When the TSD flag is 0, the RDTSC instruction can be executed at any privilege level. When the TSD flag is 1, the RDTSC instruction can only be executed at privilege level 0.

2.7 Performance Monitoring

Performance monitoring allows counting of over a hundred different event occurrences and durations. Two 48-bit counters are used: Performance Monitor Counter 0 and Performance Monitor Counter 1. These two performance monitor counters are controlled by the Counter Event Control Register MSR(11). The performance monitor counters use a continuous CPU core clock and will continue to count clock cycles even when the M II CPU is in suspend mode or shutdown.

2.8 Performance Monitoring Counters 1 and 2

The 48-bit Performance Monitoring Counters (PMC) Registers MSR(12), MSR(13) count events as specified by the counter event control register.

The PMCs can be accessed by the RDMSR and WRMSR instructions. In addition, the PMCs can be read by the RDPMC instruction, opcode 0F33h. The RDPMC instruction loads the contents of the PMC register specified in the ECX register into EDX:EAX. The use of RDPMC instructions is restricted by the Performance Monitoring Counter Enable, (PCE) flag in C4.

When the PCE flag is set to 1, the RDPMC instruction can be executed at any privilege level. When the PCE flag is 0, the RDPMC instruction can only be executed at privilege level 0.

2.8.1 Counter Event Control Register

Register MSR(11) controls the two internal counters, #0 and #1. The events to be counted have been chosen based on the micro-architecture of the M II processor. The control register for the two event counters is described in Figure 2-21 (Page 2-36) and Table 2-23 (Page 2-40).

2.8.1.1 PM Pin Control

The Counter Event Control register MSR(11) contains PM control fields that define the PM0 and PM1 pins as counter overflow indicators or counter event indicators. When defined as event counters, the PM pins indicate that one or more events occurred during a particular clock cycle and do not count the actual events.

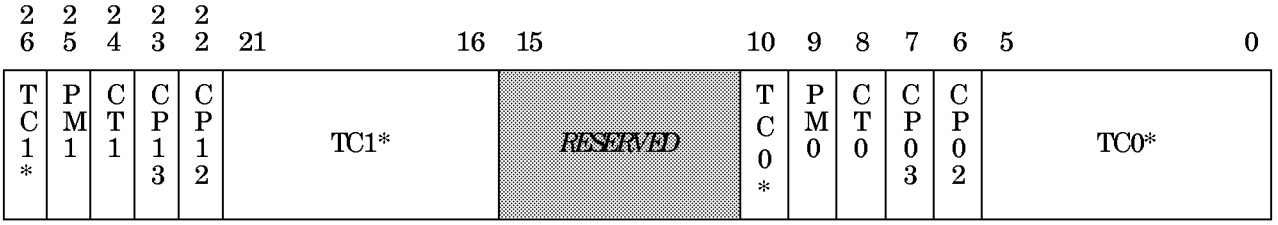
When defined as overflow indicators, the event counters can be preset with a value less the 2^{48}-1 and allowed to increment as events occur. When the counter overflows the PM pin becomes asserted.

2.8.1.2 Counter Type Control

The Counter Type bit determines whether the counter will count clocks or events. When counting clocks the counter operates as a timer.

2.8.1.3 CPL Control

The Current Privilege Level (CPL) can be used to determine if the counters are enabled. The CP02 bit in the MSR(11) register enables counting when the CPL is less than three, and the CP03 bit enables counting when CPL is equal to three. If both bits are set, counting is not dependent on the CPL level; if neither bit is set, counting is disabled.



*Note: Split Fields

Figure 2-22. Counter Event Control Register

Table 2-23. Counter Event Control Register Bit Definitions

BIT POSITION	NAME	DESCRIPTION
25	PM1	Define External PM1 Pin If = 1: PM1 pin indicates counter overflows If = 0: PM1 pin indicates counter events
24	CT1	Counter #1 Counter Type If = 1: Count clock cycles If = 0: Count events (reset state).
23	CP13	Counter #1 CPL 3 Enable If = 1: Enable counting when CPL=3. If = 0: Disable counting when CPL=3. (reset state)
22	CP12	Counter #1 CPL Less Than 3 Enable If = 1: Enable counting when CPL < 3. If = 0: Disable counting when CPL < 3. (reset state)
26, 21 - 16	TC1(5-0)	Counter #1 Event Type Reset state = 0
9	PM0	Define External PM0 Pin If = 1: PM0 pin indicates counter overflows If = 0: PM0 pin indicates counter events
8	CT0	Counter #0 Counter Type If = 1: Count clock cycles If = 0: Count events (reset state).
7	CP03	Counter #0 CPL 3 Enable If = 1: Enable counting when CPL=3. If = 0: Disable counting when CPL=3. (reset state)
6	CP02	Counter #0 CPL Less Than 3 Enable If = 1: Enable counting when CPL < 3. If = 0: Disable counting when CPL < 3. (reset state)
10, 5 - 0	TC0(5-0)	Counter #0 Event Type Reset state = 0

Note: Bits 10 - 15 are reserved.

2.8.2 Event Type and Description

The events that can be counted by the performance monitoring counters are listed in Table 2-24. Each of the 127 event types is assigned an event number.

A particular event number to be counted is placed in one of the MSR(11) Event Type fields. There is a separate field for counter #0 and #1.

The events are divided into two groups. The occurrence type events and duration type events. The occurrence type events, such as hardware interrupts, are counted as single events. The duration type events such as "clock while bus cycles are in progress" count the number of clock cycles that occur during the event.

During occurrence type events, the PM pins are configured to indicate the counter has incremented The PM pins will then assert every time the counter increments in regards to an occurrence event. Under the same PM control, for a duration event the PM pin will stay asserted for the duration of the event.

Table 2-24. Event Type Register

NUMBER	COUNTER 0	COUNTER 1	DESCRIPTION	TYPE
00h	yes	yes	Data Reads	Occurrence
01h	yes	yes	Data Writes	Occurrence
02h	yes	yes	Data TLB Misses	Occurrence
03h	yes	yes	Cache Misses: Data Reads	Occurrence
04h	yes	yes	Cache Misses: Data Writes	Occurrence
05h	yes	yes	Data Writes that hit on Modified or Exclusive Liens	Occurence
06h	yes	yes	Data Cache Lines Written Back	Occurrence
07h	yes	yes	External Inquiries	Occurrence
08h	yes	yes	External Inquires that hit	Occurrence
09h	yes	yes	Memory Accesses in both pipes	Occurrence
0Ah	yes	yes	Cache Bank conflicts	Occurrence
0Bh	yes	yes	Misaligned data references	Occurrence
0Ch	yes	yes	Instruction Fetch Requests	Occurrence
0Dh	yes	yes	L2 TLB Code Misses	Occurrence
0Eh	yes	yes	Cache Misses: Instruction Fetch	Occurrence
0Fh	yes	yes	Any Segment Register Load	Occurrence
10h	yes	yes	Reserved	Occurrence
11h	yes	yes	Reserved	Occurrence
12h	yes	yes	Any Branch	Occurrence

Table 2-24. Event Type Register (Continued)

NUMBER	COUNTER 0	COUNTER 1	DESCRIPTION	TYPE
13h	yes	yes	BTB hits	Occurrence
14h	yes	yes	Taken Branches or BTB hits	Occurrence
15h	yes	yes	Pipeline Flushes	Occurrence
16h	yes	yes	Instructions executed in both pipes	Occurrence
17h	yes	yes	Instructions executed in Y pipe	Occurrence
18h	yes	yes	Clocks while bus cycles are in progress	Duration
19h	yes	yes	Pipe Stalled by full write buffers	Duration
1Ah	yes	yes	Pipe Stalled by waiting on data memory reads	Duration
1Bh	yes	yes	Pipe Stalled by writes to not-Modified or not-Exclusive cache lines.	Duration
1Ch	yes	yes	Locked Bus Cycles	Occurrence
1Dh	yes	yes	I/O Cycles	Occurrence
1Eh	yes	yes	Non-cacheable Memory Requests	Occurrence
1Fh	yes	yes	Pipe Stalled by Address Generation Interlock	Duration
20h	yes	yes	Reserved	
21h	yes	yes	Reserved	
22h	yes	yes	Floating Point Operations	Occurrence
23h	yes	yes	Breakpoint Matches on DR0 register	Occurrence
24h	yes	yes	Breakpoint Matches on DR1 register	Occurrence
25h	yes	yes	Breakpoint Matches on DR2 register	Occurrence
26h	yes	yes	Breakpoint Matches on DR3 register	Occurrence
27h	yes	yes	Hardware Interrupts	Occurrence
28h	yes	yes	Data Reads or Data Writes	Occurrence
29h	yes	yes	Data Read Misses or Data Write Misses	Occurrence
2Bh	yes	no	MMX Instruction Executed in X pipe	Occurrence
2Bh	no	yes	MMX Instruction Executed in Y pipe	Occurrence
2Dh	yes	no	EMMS Instruction Executed	Occurrence
2Dh	no	yes	Transition Between MMX Instruction and FP Instructions	Occurrence
2Eh	no	yes	Reserved	
2Fh	yes	no	Saturating MMX Instructions Executed	Occurrence
2Fh	no	yes	Saturations Performed	Occurrence
30h	yes	no	Reserved	
31h	yes	no	MMX Instruction Data Reads	Occurrence
32h	yes	no	Reserved	
32h	no	yes	Taken Branches	Occurrence
33h	no	yes	Reserved	
34h	yes	no	Reserved	
34h	no	yes	Reserved	
35h	yes	no	Reserved	

Table 2-24. Event Type Register (Continued)

NUMBER	COUNTER 0	COUNTER 1	DESCRIPTION	TYPE
35h	no	yes	Reserved	
36h	yes	no	Reserved	
36h	no	yes	Reserved	
37h	yes	no	Returns Predicted Incorrectly	Occurrence
37h	no	yes	Return Predicted (Correctly and Incorrectly)	Occurrence
38h	yes	no	MMX Instruction Multiply Unit Interlock	Duration
38h	no	yes	MODV/MOVQ Store Stall Due to Previous Operation	Duration
39h	yes	no	Returns	Occurrence
39h	no	yes	RSB Overflows	Occurrence
3Ah	yes	no	BTB False Entries	Occurrence
3Ah	no	yes	BTB Miss Prediction on a Not-Taken Back	Occurrence
3Bh	yes	no	Number of Clock Stalled Due to Full Write Buffers While Executing	Duration
3Bh	no	yes	Stall on MMX Instruction Write to E or M Line	Duration
3C - 3Fh	yes	yes	Reserved	Duration
40h	yes	yes	L2 TLB Misses (Code or Data)	Occurrence
41h	yes	yes	L1 TLB Data Miss	Occurrence
42h	yes	yes	L1 TLB Code Miss	Occurrence
43h	yes	yes	L1 TLB Miss (Code or Data)	Occurrence
44h	yes	yes	TLB Flushes	Occurrence
45h	yes	yes	TLB Page Invalidates	Occurrence
46h	yes	yes	TLB Page Invalidates that hit	Occurrence
47h	yes	yes	Reserved	
48h	yes	yes	Instructions Decoded	Occurrence
49h	yes	yes	Reserved	

4.0 ELECTRICAL SPECIFICATIONS

4.1 Electrical Connections

This section provides information on electrical connections, absolute maximum ratings, recommended operating conditions, DC characteristics, and AC characteristics. All voltage values in Electrical Specifications are measured with respect to V_{SS} unless otherwise noted.

The MII CPU operates using two power supply voltages—one for the I/O (3.3 V) and one for the core (2.9 V).

4.1.1 Power and Ground Connections and Decoupling

Testing and operating the MII CPU requires the use of standard high frequency techniques to reduce parasitic effects. The high clock frequencies used in the MII CPU and its output buffer circuits can cause transient power surges when several output buffers switch output levels simultaneously. These effects can be minimized by filtering the DC power leads with low-inductance decoupling capacitors, using low impedance wiring, and by utilizing all of the V_{CC} and GND pins. The MII CPU contains 296 pins with 25 pins connected to V_{CC2} (2.9 volts), 28 pins connected to V_{CC3} (3.3 volts), and 53 pins connected to V_{SS} (ground).

4.1.2 Pull-Up/ Pull-Down Resistors

Table 4-1 lists the input pins that are internally connected to pull-up and pull-down resistors. The pull-up resistors are connected to V_{CC} and the pull-down resistors are connected to V_{SS}. When unused, these inputs do not require connection to external pull-up or pull-down resistors. The SUSP# pin is unique in that it is connected to a pull-up resistor only when SUSP# is not asserted.

Table 4-1. Pins Connected to Internal Pull-Up and Pull-Down Resistors

SIGNAL	PIN NO.	RESISTOR
BRDYC#	Y3	20-kΩ pull-up
CKMUL0	Y33	20-kΩ pull-down (see text)
CKMUL1	X34	20-kΩ pull-up (see text)
Reserved	AN35	20-kΩ pull-down
Reserved	W35	20-kΩ pull-up
SMI#	AB34	20-kΩ pull-up
SUSP#	Y34	20-kΩ pull-up (see text)
TCK	M34	20-kΩ pull-up
TDI	N35	20-kΩ pull-up
TMS	P34	20-kΩ pull-up
TRST#	Q33	20-kΩ pull-up

4.1.3 Unused Input Pins

All inputs not used by the system designer and not listed in Table 4-1 should be connected either to ground or to V_{CC}. Connect active--high inputs to ground through a 10 kΩ (± 10%) pull-down resistor and active-low inputs to V_{CC} through a 10 kΩ (± 10%) pull-up resistor to prevent possible spurious operation.

4.1.4 NC and Reserved Pins

Pins designated NC have no internal connections. Pins designated RESV or RESERVED should be left disconnected. Connecting a reserved pin to a pull-up resistor, pull-down resistor, or an active signal could cause unexpected results and possible circuit malfunctions.

4.2 Absolute Maximum Ratings

The following table lists absolute maximum ratings for the M II CPU processors. Stresses beyond those listed under Table 4-2 limits may cause permanent damage to the device. These are stress ratings only and do not imply that operation under any conditions other than those listed under "Recommended Operating Conditions" Table 4-3 (Page 4-3) is possible. Exposure to conditions beyond Table 4-2 may (1) reduce device reliability and (2) result in premature failure even when there is no immediately apparent sign of failure. Prolonged exposure to conditions at or near the absolute maximum ratings may also result in reduced useful life and reliability.

Table 4-2. Absolute Maximum Ratings

PARAMETER	MIN	MAX	UNITS	NOTES
Operating Case Temperature	-65	110	°C	Power Applied
Storage Temperature	-65	150	°C	
Supply Voltage, V_{CC3}	-0.5	4.0	V	
Supply Voltage, V_{CC2}	-0.5	3.3	V	
Voltage On Any Pin	-0.5	V_{CC3} + 0.5	V	Not to exceed Vcc3 max
Input Clamp Current, I_{IK}		10	mA	Power Applied
Output Clamp Current, I_{OK}		25	mA	Power Applied

4.3 Recommended Operating Conditions

Table 4-3 presents the recommended operating conditions for the MII CPU device.

Table 4-3. Recommended Operating Conditions

PARAMETER	MIN	MAX	UNITS	NOTES
T_C Operating Case Temperature	0	70	°C	Power Applied
V_{CC3} Supply Voltage (3.3 V)	3.135	3.465	V	
V_{CC2} Supply Voltage (2.9 V)	2.8	3.0	V	
V_{IH} High-Level Input Voltage (except CLK)	2.00	3.55	V	
V_{IH} CLK High-Level Input Voltage	2.0	5.5	V	
V_{IL} Low-Level Input Voltage	-0.3	0.8	V	
I_{OH} High-Level Output Current		-1.0	mA	$V_O=V_{OH(MIN)}$
I_{OL} Low-Level Output Current		5.0	mA	$V_O=V_{OL(MAX)}$

4.4 DC Characteristics

Table 4-4. DC Characteristics (at Recommended Operating Conditions) 1 of 2

PARAMETER	MIN	TYP	MAX	UNITS	NOTES
V_{OL} Low-Level Output Voltage			0.4	V	I_{OL} = 5 mA
V_{OH} High-Level Output Voltage	2.4			V	I_{OH} = -1 mA
I_I Input Leakage Current For all pins (except those listed in Table 4-1).			±15	µA	$0 < V_{IN} < V_{CC3}$ Note 1
I_{IH} Input Leakage Current For all pins with internal pull-downs.			200	µA	V_{IH} = 2.4 V Note 1
I_{IL} Input Leakage Current For all pins with internal pull-ups.			-400	µA	V_{IL} = 0.45 V Note 1
C_{IN} Input Capacitance			15	pF	f = 1 MHz*
C_{OUT} Output Capacitance			20	pF	f = 1 MHz*
C_{IO} I/O Capacitance			25	pF	f = 1 MHz*
C_{CLK} CLK Capacitance			15	pF	f = 1 MHz*

*Note: Not 100% tested.

Table 4-5. DC Characteristics (at Recommended Operating Conditions) 2 of 2

PARAMETER	ICC2 MAX	ICC3 MAX	UNITS	NOTES
I_{CC} Active I_{CC}				Notes 1, 2
75/225 MHz (MII-300)	7480	100	mA	
66/233 MHz (MII-300)	7740	100		
83/250 MHz (MII-333)	8350	100		
I_{CCSM} Active I_{CC}				Notes 1, 2, 3
75/225 MHz (MII-300)	52	100	mA	
66/233 MHz (MII-300)	54	100		
83/250 MHz (MII-333)	57	100		
I_{CCSS} Standby I_{CC}				Notes 1, 2, 4
0 MHz (Suspended/CLK Stopped)	30	50.0	mA	

Notes:
1. These values should be used for power supply design. Maximum I_{CC} is determined using the worst-case instruction sequences and functions at maximum Vcc.
2. Frequency (MHz) ratings refer to the internal clock frequency.
3. All inputs at 0.4 or V_{CC3} - 0.4 (CMOS levels). All inputs held static except clock and all outputs unloaded (static I_{OUT} = 0 mA).
4. All inputs at 0.4 or V_{CC3} - 0.4 (CMOS levels). All inputs held static and all outputs unloaded (static I_{OUT} = 0 mA).

Table 4-6. Power Dissipation

PARAMETER	POWER		UNITS	NOTES
	TYP	MAX		
Active Power Dissipation				Note 1
75/225 MHz (MII-300)	12.5	20.7	W	
66/233 MHz (MII-300)	13.0	21.6		
83/250 MHz (MII-333)	14.0	23.3		
Suspend Mode Power Dissipation				Notes 1, 2
75/225 MHz (MII-300)		0.150	W	
66/233 MHz (MII-300)		0.152		
83/250 MHz (MII-333)		0.157		
Standby Mode Power Dissipation				Notes 1, 3
0 MHz (Suspended/CLK Stopped)		0.070	W	

Notes:
1. Systems must be designed to thermally dissipate the maximum active power dissipation. Maximum power is determined using the worst-case instruction sequences and functions with Vcc2 = 2.9 V and Vcc3 = 3.3 V.
2. All inputs at 0.4 or V_{CC3} - 0.4 (CMOS levels). All inputs held static except clock and all outputs unloaded (static I_{OUT} = 0 mA).
3. All inputs at 0.4 or V_{CC3} - 0.4 (CMOS levels). All inputs held static and all outputs unloaded (static I_{OUT} = 0 mA).

4.5 AC Characteristics

Tables 4-7 through 4-12 (Pages 4-8 through 4-11) list the AC characteristics including output delays, input setup requirements, input hold requirements and output float delays. These measurements are based on the measurement points identified in Figure 4-1 (Page 4-7) and Figure 4-2 (Page 4-8). The rising clock edge reference level V_{REF} and other reference levels are shown in Table 4-7. Input or output signals must cross these levels during testing.

Figure 4-1 shows output delay (A and B) and input setup and hold times (C and D). Input setup and hold times (C and D) are specified minimums, defining the smallest acceptable sampling window a synchronous input signal must be stable for correct operation.

The JTAG AC timing is shown in Table 4-13 (Page 13) supported by Figures 4-6 (Page 4-13) though 4-8 (Page 4-14).

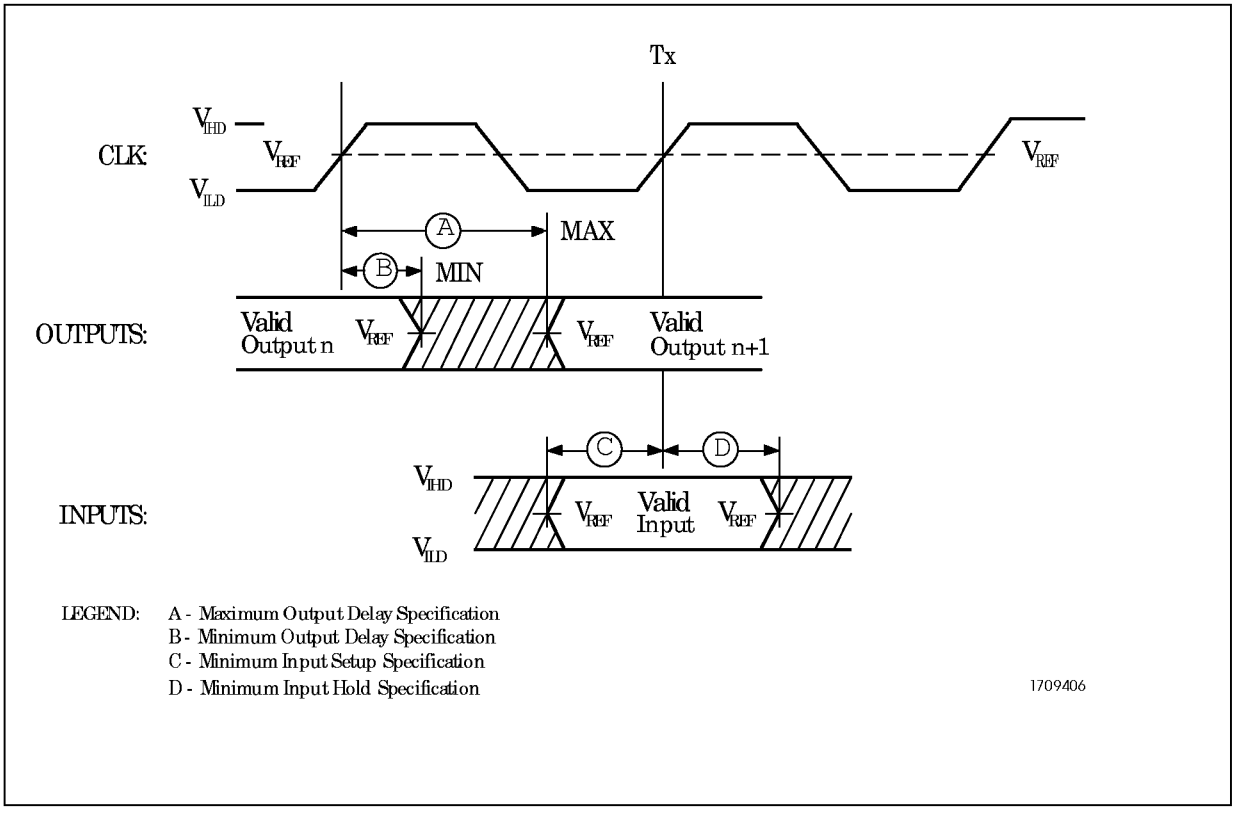


Figure 4-1. Drive Level and Measurement Points for Switching Characteristics

Table 4-7. Drive Level and Measurement Points for Switching Characteristics

SYMBOL	VOLTAGE (Volts)
V_{REF}	1.5
V_{IHD}	2.3
V_{ILD}	0

Note: Refer to Figure 4-1.

Table 4-8. Clock Specifications

T_{CASE} = 0°C to 70°C, See Figure 4-2

	PARAMETER	60-MHz BUS		66-MHz BUS		75-MHz BUS		83-MHz BUS		UNITS
		MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	
f	CLK Frequency		60		66.6		75		83	MHz
T1	CLK Period	16.67		15.0		13.33		12.0		ns
T2	CLK Period Stability		±250		±250		±250		±250	ps
T3	CLK High Time	4.0		4.0		4.0		4.0		ns
T4	CLK Low Time	4.0		4.0		4.0		4.0		ns
T5	CLK Fall Time	0.15	1.5	0.15	1.5	0.15	1.5	0.15	1.5	ns
T6	CLK Rise Time	0.15	1.5	0.15	1.5	0.15	1.5	0.15	1.5	ns

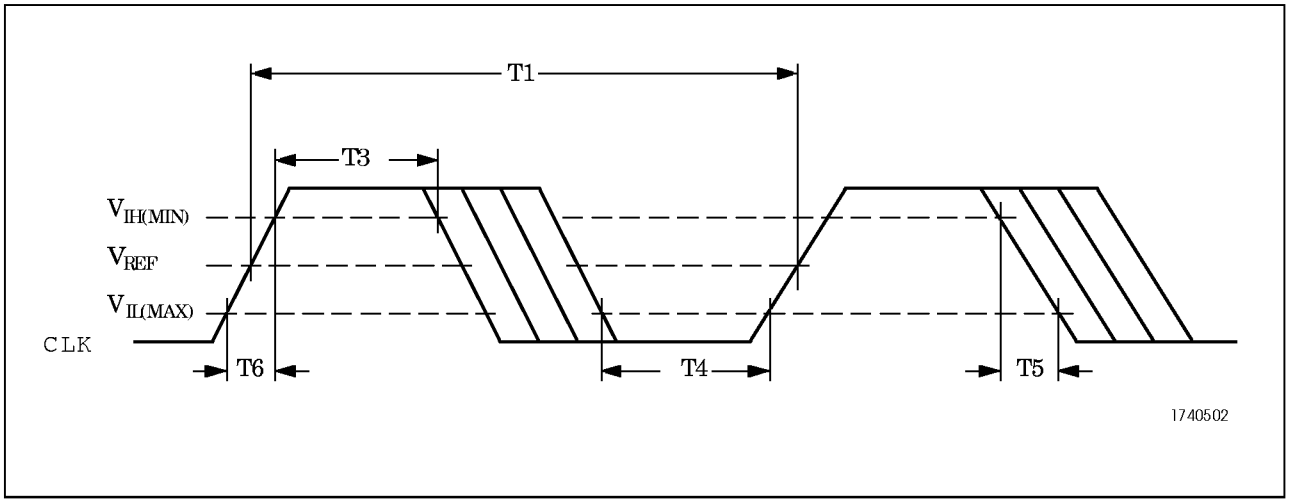


Figure 4-2. CLK Timing and Measurement Points

Table 4-9. Output Valid Delays

C_L = 50 pF, T_{case} = 0°C to 70°C, See Figure 4-3

	PARAMETER	60-MHz BUS		66-MHz BUS		75-MHz BUS		83-MHz BUS		UNITS
		MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	
T7a	A31-A3	1.0	7.0	1.0	6.3	1.0	6.3	1.0	5.7	ns
T7b	BE7#-BE0#, CACHE#, D/C#, LOCK#, PCD, PWT, SCYC, SMIACT#, W/R#	1.0	7.0	1.0	7.0	1.0	7.0	1.0	6.0	ns
T7c	ADS#	1.0	7.0	1.0	6.0	1.0	6.0	1.0	5.5	ns
T7d	M/IO#	1.0	7.0	1.0	5.9	1.0	5.9	1.0	5.5	ns
T8	ADSC#	1.0	7.0	1.0	7.0	1.0	7.0	1.0	6.5	ns
T9	AP	1.0	8.5	1.0	8.5	1.0	8.5	1.0	7.5	ns
T10	APCHK#, PCHK#, FERR#	1.0	8.3	1.0	7.0	1.0	7.0	1.0	6.5	ns
T11	D63-D0, DP7-DP0 (Write)	1.3	7.5	1.3	7.5	1.3	7.5	1.3	7.0	ns
T12a	HIT#	1.0	8.0	1.0	6.8	1.0	6.8	1.0	6.0	ns
T12b	HITM#	1.1	6.0	1.1	6.0	1.1	6.0	1.1	5.5	ns
T13a	BREQ	1.0	8.0	1.0	8.0	1.0	8.0	1.0	7.0	ns
T13b	HLDA	1.0	8.0	1.0	6.8	1.0	6.8	1.0	6.0	ns
T14	SUSPA#	1.0	8.0	1.0	8.0	1.0	8.0	1.0	7.0	ns

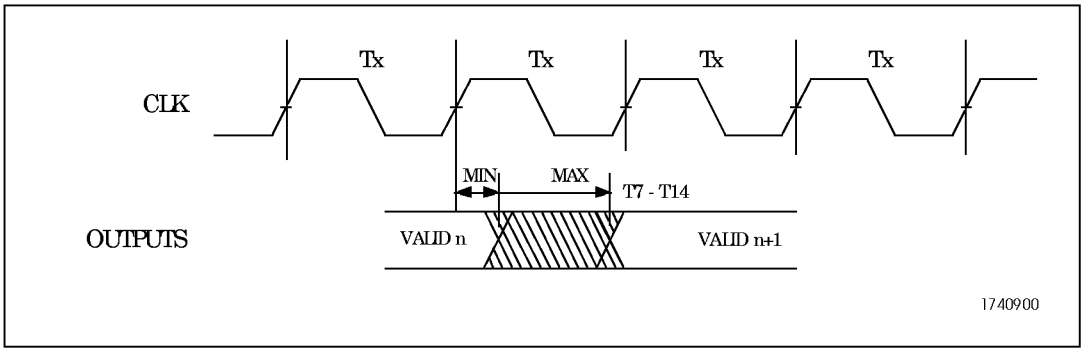


Figure 4-3. Output Valid Delay Timing

Table 4-10. Output Float Delays

C_L = 50 pF, T_{case} = 0°C to 70°C, See Figure 4-5

	PARAMETER	60-MHz BUS		66-MHz BUS		75-MHz BUS		83-MHz BUS		UNITS
		MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	
T15	A31-A3, ADS#, BE7#-BE0#, CACHE#, D/C#, LOCK#, PCD, PWT, SCYC, SMIACT#, W/R#		10.0		10.0		10.0		10.0	ns
T16	AP		10.0		10.0		10.0		10.0	ns
T17	D63-D0, DP7-DP0 (Write)		10.0		10.0		10.0		10.0	ns

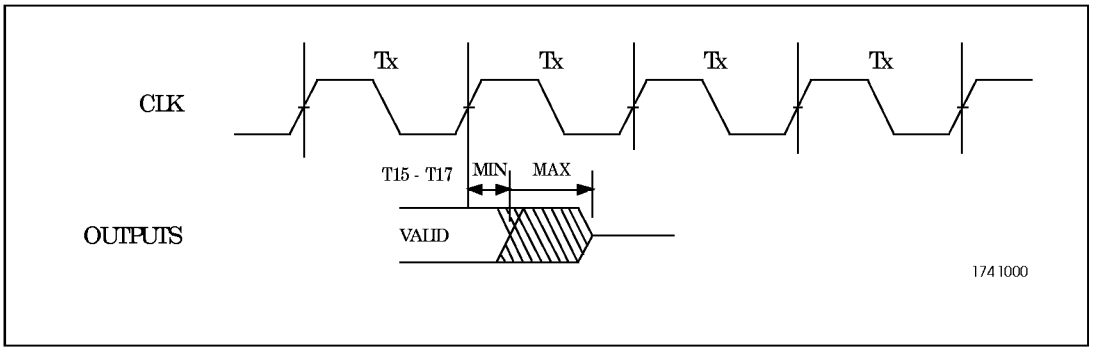


Figure 4-4. Output Float Delay Timing

Table 4-11. Input Setup Times

T_{case} = 0°C to 70°C, See Figure 4-5

SYMBOL	PARAMETER	60-MHz BUS	66-MHz BUS	75-MHz BUS	83-MHz BUS	UNITS
		MIN	MIN	MIN	MIN	
T18a	A20M#,	5.0	5.0	3.3	3.0	ns
T18b	FLUSH#, IGNNE#, SUSP#	5.0	5.0	3.3	3.0	ns
T19a	AHOLD, BOFF#	5.0	5.0	3.3	3.0	ns
T19b	HOLD	5.0	5.0	3.3	3.0	ns
T20	BRDY#	5.0	5.0	3.3	3.0	ns
T21	BRDYC#	5.0	5.0	3.3	3.0	ns
T22a	A31-A3, AP, BE7#-BE0#,	5.0	5.0	3.3	3.0	ns
T22b	AP	5.0	5.0	3.3	3.0	ns
T22c	D63-D0 (Read), DP7-DP0 (Read)	3.0	3.0	3.0	2.7	ns
T23a	EADS#	5.0	5.0	5.0	4.5	ns
T23b	INV	5.0	5.0	5.0	4.5	ns
T24	INTR, NMI, RESET, SMI#, WM_RST	5.0	5.0	5.0	4.5	ns
T25a	EWBE#, NA#, WB/WT#	4.5	4.5	3.0	2.7	ns
T25b	KEN#	4.5	4.5	3.0	2.7	ns

Table 4-12. Input Hold Times

T_{case} = 0°C to 70°C, See Figure 4-5

SYMBOL	PARAMETER	60-MHz BUS	66-MHz BUS	75-MHz BUS	83-MHz BUS	UNITS
		MIN	MIN	MIN	MIN	
T27	A20M#, FLUSH#, IGNNE#, SUSP#	1.0	1.0	1.0	1.0	ns
T28a	AHOLD, BOFF#	1.0	1.0	1.0	1.0	ns
T28b	HOLD	1.0	1.0	1.0	1.0	ns
T29	BRDY#	1.0	1.0	1.0	1.0	ns
T30	BRDYC#	1.0	1.0	1.0	1.0	ns
T31a	A31-A3, AP, BE7#-BE0#,	1.0	1.0	1.0	1.0	ns
T31b	AP	1.0	1.0	1.0	1.0	ns
T31c	D63-D0 (Read), DP7-DP0 (Read)	2.0	1.5	1.5	1.5	ns
T32	EADS#, INV	1.0	1.0	1.0	1.0	ns
T33	INTR, NMI, RESET, SMI#, WM_RST	1.0	1.0	1.0	1.0	ns
T34	EWBE#, KEN#, NA#, WB/WT#	1.0	1.0	1.0	1.0	ns

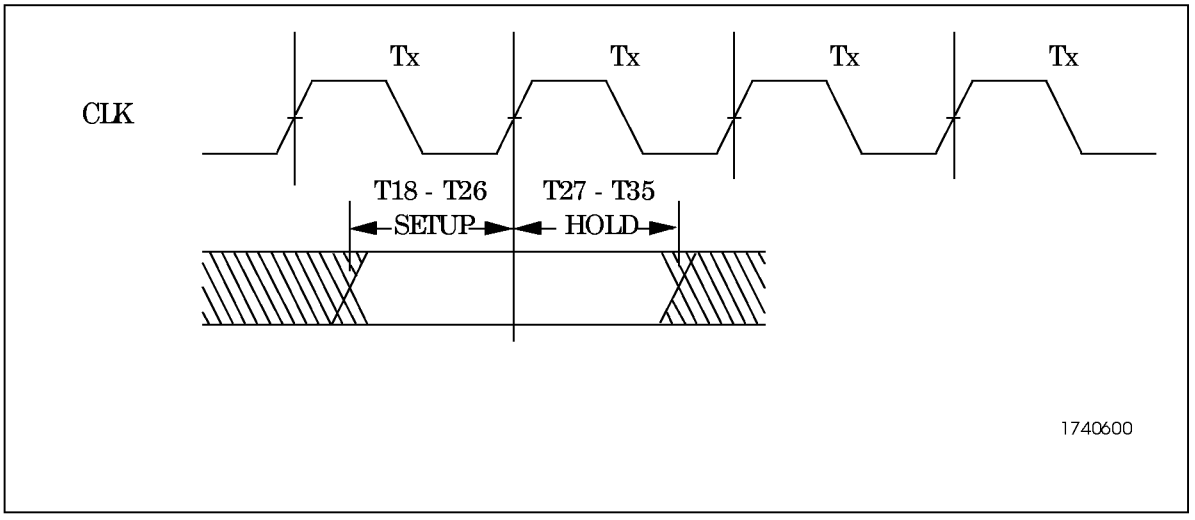


Figure 4-5. Input Setup and Hold Timing

Table 4-13. JTAG AC Specifications

SYMBOL	PARAMETER	ALL BUS FREQUENCIES		UNITS	FIGURE
		MIN	MAX		
	TCK Frequency (MHz)		20	ns	
T36	TCK Period	50		ns	4-6
T37	TCK High Time	25		ns	4-6
T38	TCK Low Time	25		ns	4-6
T39	TCK Rise Time		5	ns	4-6
T40	TCK Fall Time		5	ns	4-6
T41	TDO Valid Delay	3	20	ns	4-7
T42	Non-test Outputs Valid Delay	3	20	ns	4-7
T43	TDO Float Delay		25	ns	4-7
T44	Non-test Outputs Float Delay		25	ns	4-7
T45	TRST# Pulse Width	40		ns	4-8
T46	TDI, TMS Setup Time	20		ns	4-7
T47	Non-test Inputs Setup Time	20		ns	4-7
T48	TDI, TMS Hold Time	13		ns	4-7
T49	Non-test Inputs Hold Time	13		ns	4-7

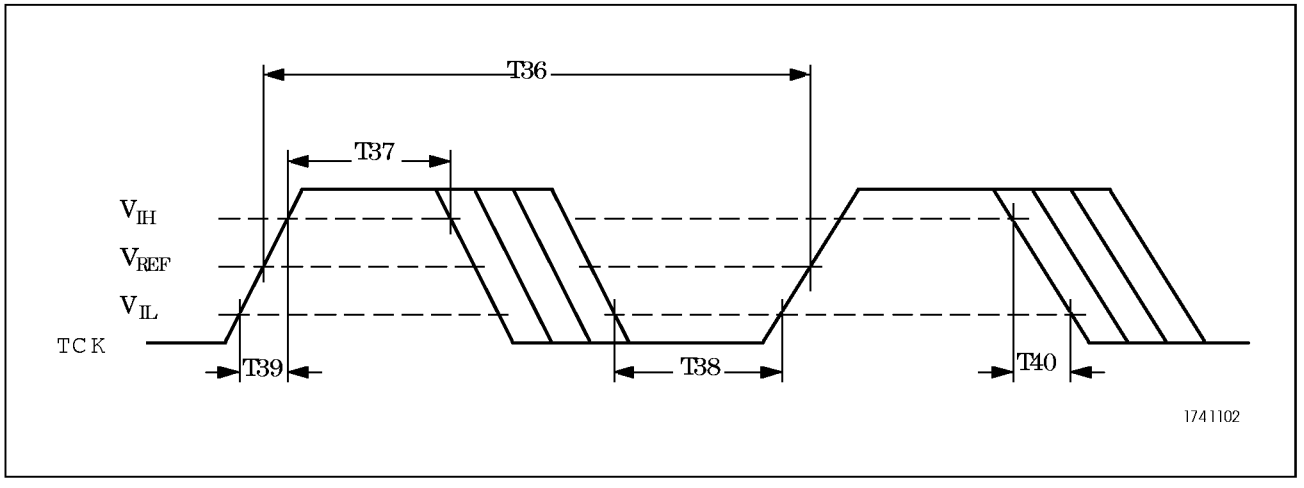


Figure 4-6. TCK Timing and Measurement Points

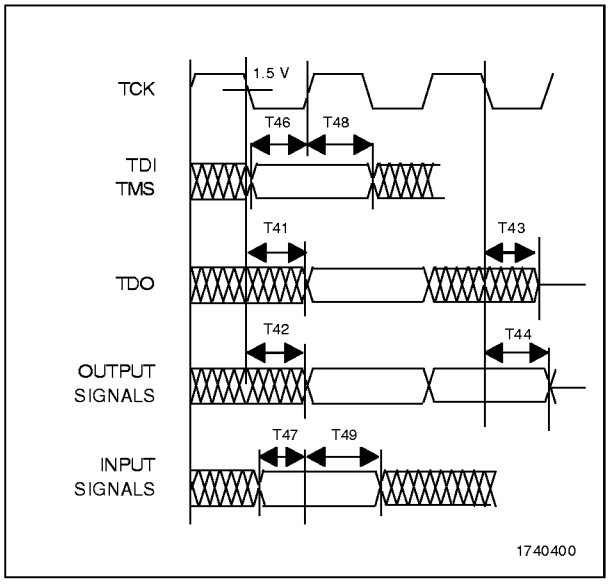


Figure 4-7. JTAG Test Timings

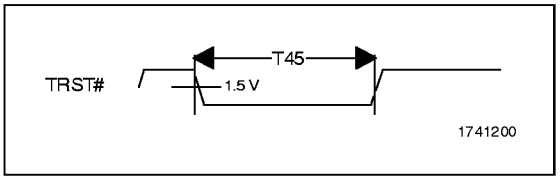


Figure 4-8. Test Reset Timing

5.0 MECHANICAL SPECIFICATIONS

5.1 296-Pin SPGA Package

The pin assignments for the M II CPU in a 296-pin SPGA package are shown in Figure 5-1. The pins are listed by signal name in Table 5-1(Page 5-3) and by pin number in Table 5-2 (Page 5-4). Dimensions are shown in Figure 5-2 (Page 5-5) and Table 5-3 (Page 5-6).

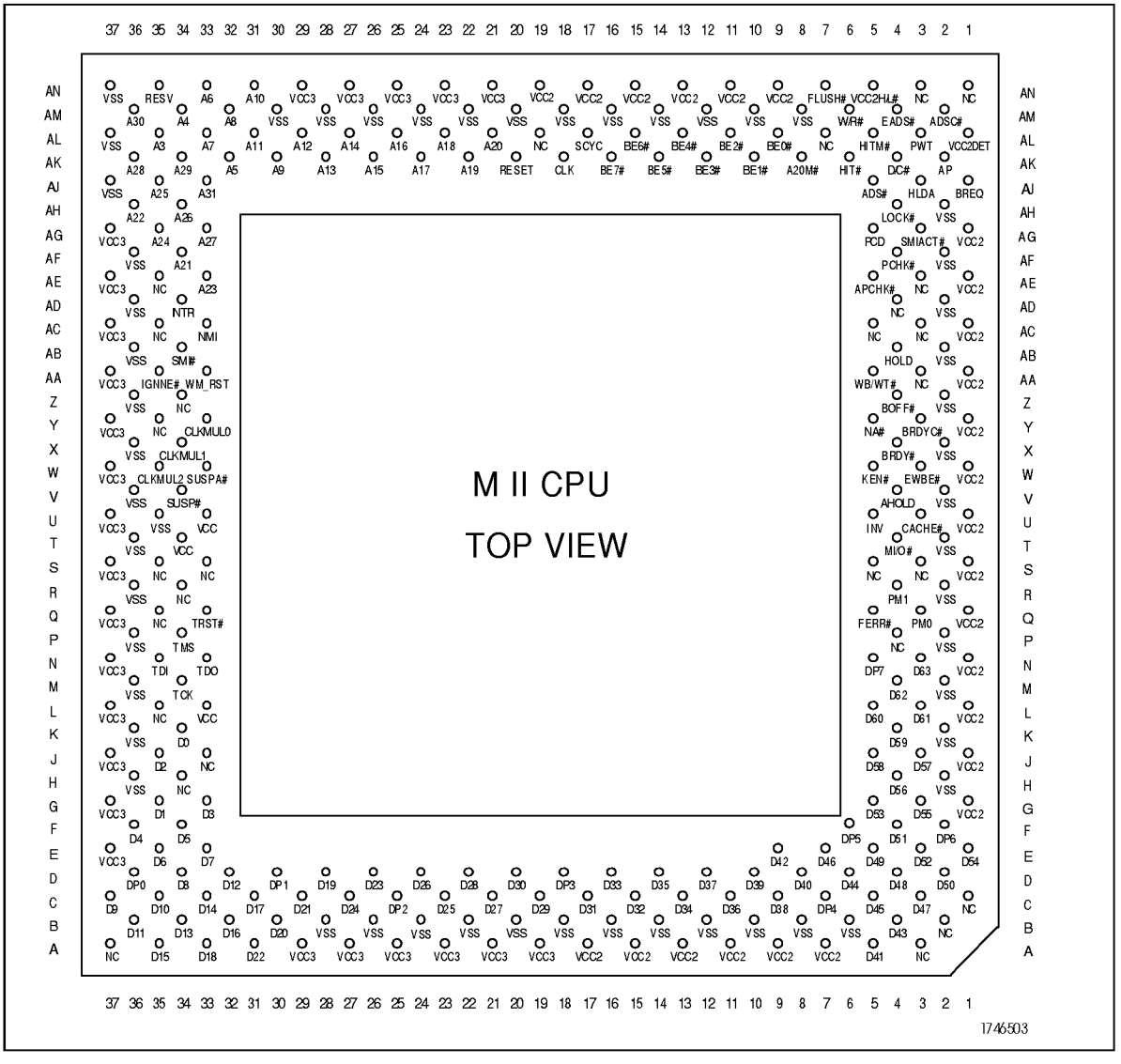


Figure 5-1. 296-Pin SPGA Package Pin Assignments

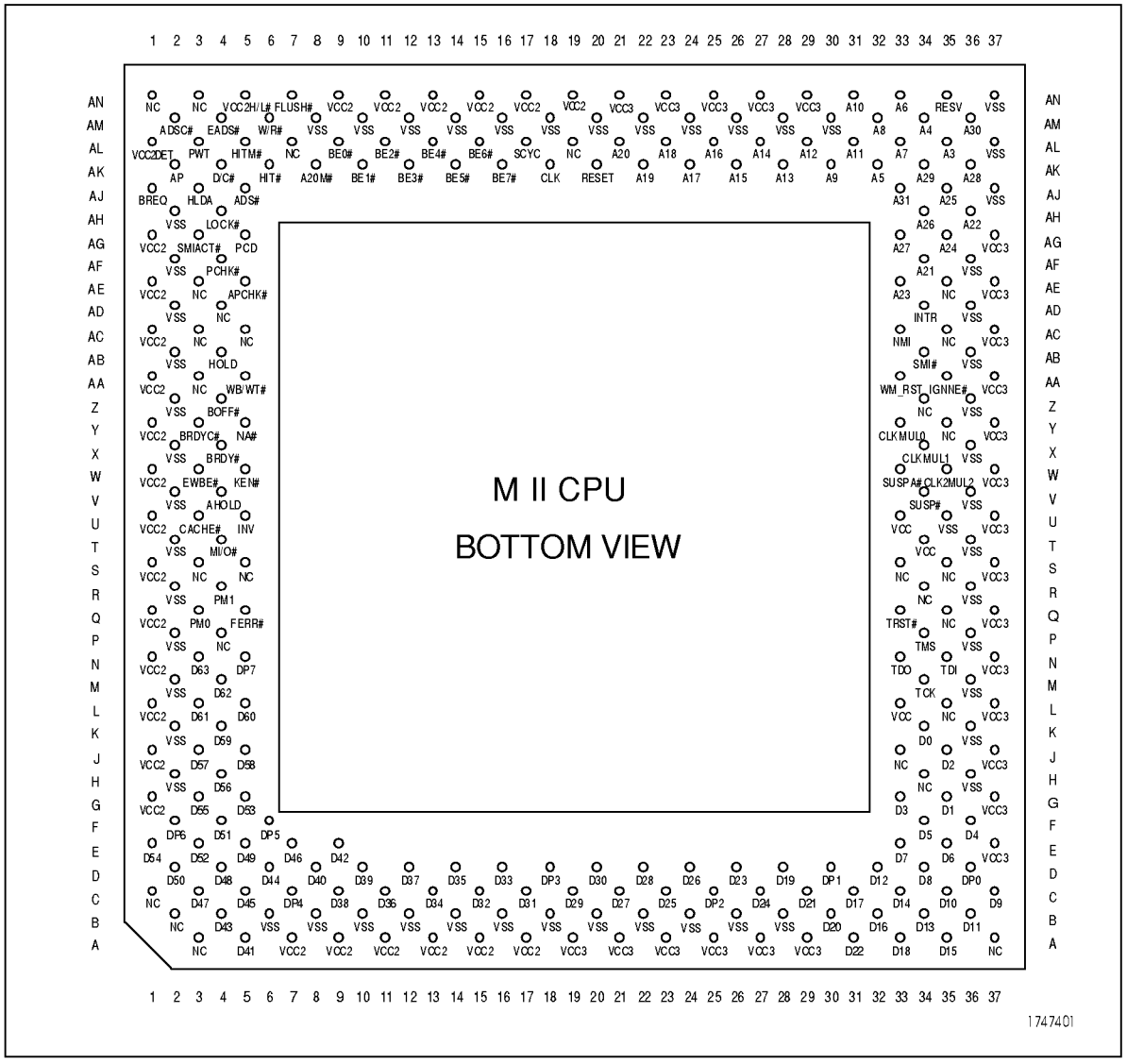


Figure 5-2. 296-Pin SPGA Package Pin Assignments (Bottom View)

Table 5-1. 296-Pin SPGA Package Signal Names Sorted by Pin Number

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
A3	NC	C29	D21	J35	D2	U35	Vss	AE35	NC	AL21	A20
A5	D41	C31	D17	J37	Vcc3	U37	Vcc3	AE37	Vcc3	AL23	A18
A7	Vcc2	C33	D14	K2	Vss	V2	Vss	AF2	Vss	AL25	A16
A9	Vcc2	C35	D10	K4	D59	V4	AHOLD	AF4	PCHK#	AL27	A14
A11	Vcc2	C37	D9	K34	D0	V34	SUSP#	AF34	A21	AL29	A12
A13	Vcc2	D2	D50	K36	Vss	V36	Vss	AF36	Vss	AL31	A11
A15	Vcc2	D4	D48	L1	Vcc2	W1	Vcc2	AG1	Vcc2	AL33	A7
A17	Vcc2	D6	D44	L3	D61	W3	EWBE#	AG3	SMIACT#	AL35	A3
A19	Vcc3	D8	D40	L5	D60	W5	KEN#	AG5	PCD	AL37	Vss
A21	Vcc3	D10	D39	L33	Vcc3	W33	SUSPA#	AG33	A27	AM2	ADSC#
A23	Vcc3	D12	D37	L35	NC	W35	CLKMUL2	AG35	A24	AM4	EADS#
A25	Vcc3	D14	D35	L37	Vcc3	W37	Vcc3	AG37	Vcc3	AM6	W/R#
A27	Vcc3	D16	D33	M2	Vss	X2	Vss	AH2	Vss	AM8	Vss
A29	Vcc3	D18	DP3	M4	D62	X4	BRDY#	AH4	LOCK#	AM10	Vss
A31	D22	D20	D30	M34	TCK	X34	CLKMUL1	AH34	A26	AM12	Vss
A33	D18	D22	D28	M36	Vss	X36	Vss	AH36	A22	AM14	Vss
A35	D15	D24	D26	N1	Vcc2	Y1	Vcc2	AJ1	BREQ	AM16	Vss
A37	NC	D26	D23	N3	D63	Y3	BRDYC#	AJ3	HLDA	AM18	Vss
B2	NC	D28	D19	N5	DP7	Y5	NA#	AJ5	ADS#	AM20	Vss
B4	D43	D30	DP1	N33	TDO	Y33	CLKMUL0	AJ33	A31	AM22	Vss
B6	Vss	D32	D12	N35	TDI	Y35	NC	AJ35	A25	AM24	Vss
B8	Vss	D34	D8	N37	Vcc3	Y37	Vcc3	AJ37	Vss	AM26	Vss
B10	Vss	D36	DP0	P2	Vss	Z2	Vss	AK2	AP	AM28	Vss
B12	Vss	E1	D54	P4	NC	Z4	BOFF#	AK4	D/C#	AM30	Vss
B14	Vss	E3	D52	P34	TMS	Z34	NC	AK6	HIT#	AM32	A8
B16	Vss	E5	D49	P36	Vss	Z36	Vss	AK8	A20M#	AM34	A4
B18	Vss	E7	D46	Q1	Vcc2	AA1	Vcc2	AK10	BE1#	AM36	A30
B20	Vss	E9	D42	Q3	PM0	AA3	NC	AK12	BE3#	AN1	NC
B22	Vss	E33	D7	Q5	FERR#	AA5	WB/WT#	AK14	BE5#	AN3	NC
B24	Vss	E35	D6	Q33	TRST#	AA33	WM_RST	AK16	BE7#	AN5	NC
B26	Vss	E37	Vcc3	Q35	NC	AA35	IGNNE#	AK18	CLK	AN7	FLUSH#
B28	Vss	F2	DP6	Q37	Vcc3	AA37	Vcc3	AK20	RESET	AN9	Vcc2
B30	D20	F4	D51	R2	Vss	AB2	Vss	AK22	A19	AN11	Vcc2
B32	D16	F6	DP5	R4	PM1	AB4	HOLD	AK24	A17	AN13	Vcc2
B34	D13	F34	D5	R34	NC	AB34	SMI#	AK26	A15	AN15	Vcc2
B36	D11	F36	D4	R36	Vss	AB36	Vss	AK28	A13	AN17	Vcc2
C1	NC	G1	Vcc2	S1	Vcc2	AC1	Vcc2	AK30	A9	AN19	Vcc2
C3	D47	G3	D55	S3	NC	AC3	NC	AK32	A5	AN21	Vcc3
C5	D45	G5	D53	S5	NC	AC5	NC	AK34	A29	AN23	Vcc3
C7	DP4	G33	D3	S33	NC	AC33	NMI	AK36	A28	AN25	Vcc3
C9	D38	G35	D1	S35	NC	AC35	NC	AL1	Vcc2DET	AN27	Vcc3
C11	D36	G37	Vcc3	S37	Vcc3	AC37	Vcc3	AL3	PWT	AN29	Vcc3
C13	D34	H2	Vss	T2	Vss	AD2	Vss	AL5	HITM#	AN31	A10
C15	D32	H4	D56	T4	MI/O#	AD4	NC	AL7	NC	AN33	A6
C17	D31	H34	NC	T34	Vcc3	AD34	INTR	AL9	BE0#	AN35	VCC2H/L#
C19	D29	H36	Vss	T36	Vss	AD36	Vss	AL11	BE2#	AN37	Vss
C21	D27	J1	Vcc2	U1	Vcc2	AE1	Vcc2	AL13	BE4#		
C23	D25	J3	D57	U3	CACHE#	AE3	NC	AL15	BE6#		
C25	DP2	J5	D58	U5	INV	AE5	APCHK#	AL17	SCYC		
C27	D24	J33	NC	U33	Vcc3	AE33	A23	AL19	NC		

Table 5-2. 296-Pin SPGA Package Signal Names Sorted by Signal Names

Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin
A3	AL35	CLKMUL1	X34	D47	C3	NC	S3	Vcc2	AA1	Vss	B26
A4	AM34	CLKMUL2	W35	D48	D4	NC	S5	Vcc2	AC1	Vss	B28
A5	AK32	D/C#	AK4	D49	E5	NC	S33	Vcc2	AE1	Vss	H2
A6	AN33	D0	K34	D50	D2	NC	S35	VCC2H/L#	AN5	Vss	H36
A7	AL33	D1	G35	D51	F4	NC	Y35	Vcc2	AG1	Vss	K2
A8	AM32	D2	J35	D52	E3	NC	Z34	Vcc2	AN9	Vss	K36
A9	AK30	D3	G33	D53	G5	NC	AA3	Vcc2	AN11	Vss	M2
A10	AN31	D4	F36	D54	E1	NC	AC3	Vcc2	AN13	Vss	M36
A11	AL31	D5	F34	D55	G3	NC	AC5	Vcc2	AN15	Vss	P2
A12	AL29	D6	E35	D56	H4	NC	AC35	Vcc2	AN17	Vss	P36
A13	AK28	D7	E33	D57	J3	NC	AD4	Vcc2	AN19	Vss	R2
A14	AL27	D8	D34	D58	J5	NC	AE3	Vcc3	A19	Vss	R36
A15	AK26	D9	C37	D59	K4	NC	AE35	Vcc3	A21	Vss	T2
A16	AL25	D10	C35	D60	L5	NC	AL7	Vcc3	A23	Vss	T36
A17	AK24	D11	B36	D61	L3	NC	AL19	Vcc3	A25	Vss	U35
A18	AL23	D12	D32	D62	M4	NC	AN1	Vcc3	A27	Vss	V2
A19	AK22	D13	B34	D63	N3	NC	AN3	Vcc3	A29	Vss	V36
A20	AL21	D14	C33	DP0	D36	NC	AN5	Vcc3	E37	Vss	X2
A20M#	AK8	D15	A35	DP1	D30	NMI	AC33	Vcc3	G37	Vss	X36
A21	AF34	D16	B32	DP2	C25	PCD	AG5	Vcc3	J37	Vss	Z2
A22	AH36	D17	C31	DP3	D18	PCHK#	AF4	Vcc3	L33	Vss	Z36
A23	AE33	D18	A33	DP4	C7	PM0	Q3	Vcc3	L37	Vss	AB2
A24	AG35	D19	D28	DP5	F6	PM1	R4	Vcc3	N37	Vss	AB36
A25	AJ35	D20	B30	DP6	F2	PWT	AL3	Vcc3	Q37	Vss	AD2
A26	AH34	D21	C29	DP7	N5	RESET	AK20	Vcc3	S37	Vss	AD36
A27	AG33	D22	A31	EADS#	AM4	SCYC	AL17	Vcc3	T34	Vss	AF2
A28	AK36	D23	D26	EWBE#	W3	SMI#	AB34	Vcc3	U33	Vss	AF36
A29	AK34	D24	C27	FERR#	Q5	SMIACT#	AG3	Vcc3	U37	Vss	AH2
A30	AM36	D25	C23	FLUSH#	AN7	SUSP#	V34	Vcc3	W37	Vss	AJ37
A31	AJ33	D26	D24	HIT#	AK6	SUSPA#	W33	Vcc3	Y37	Vss	AL37
ADS#	AJ5	D27	C21	HITM#	AL5	TCK	M34	Vcc3	AA37	Vss	AM8
ADSC#	AM2	D28	D22	HLDA	AJ3	TDI	N35	Vcc3	AC37	Vss	AM10
AHOLD	V4	D29	C19	HOLD	AB4	TDO	N33	Vcc3	AE37	Vss	AM12
AP	AK2	D30	D20	IGNNE#	AA35	TMS	P34	Vcc3	AG37	Vss	AM14
APCHK#	AE5	D31	C17	INTR	AD34	TRST#	Q33	Vcc3	AN21	Vss	AM16
BE0#	AL9	D32	C15	INV	U5	Vcc2	A7	Vcc3	AN23	Vss	AM18
BE1#	AK10	D33	D16	KEN#	W5	Vcc2	A9	Vcc3	AN25	Vss	AM20
BE2#	AL11	D34	C13	LOCK#	AH4	Vcc2	A11	Vcc3	AN27	Vss	AM22
BE3#	AK12	D35	D14	MI/O#	T4	Vcc2	A13	Vcc3	AN29	Vss	AM24
BE4#	AL13	D36	C11	NA#	Y5	Vcc2	A15	Vcc2DET	AL1	Vss	AM26
BE5#	AK14	D37	D12	NC	A3	Vcc2	A17	Vss	B6	Vss	AM28
BE6#	AL15	D38	C9	NC	A37	Vcc2	G1	Vss	B8	Vss	AM30
BE7#	AK16	D39	D10	NC	B2	Vcc2	J1	Vss	B10	Vss	AN37
BOFF#	Z4	D40	D8	NC	C1	Vcc2	L1	Vss	B12	W/R#	AM6
BRDY#	X4	D41	A5	NC	H34	Vcc2	N1	Vss	B14	WB/WT#	AA5
BRDYC#	Y3	D42	E9	NC	J33	Vcc2	Q1	Vss	B16	WM_RST	AA33
BREQ	AJ1	D43	B4	NC	L35	Vcc2	S1	Vss	B18		
CACHE#	U3	D44	D6	NC	P4	Vcc2	U1	Vss	B20		
CLK	AK18	D45	C5	NC	Q35	Vcc2	W1	Vss	B22		
CLKMUL0	Y33	D46	E7	NC	R34	Vcc2	Y1	Vss	B24		

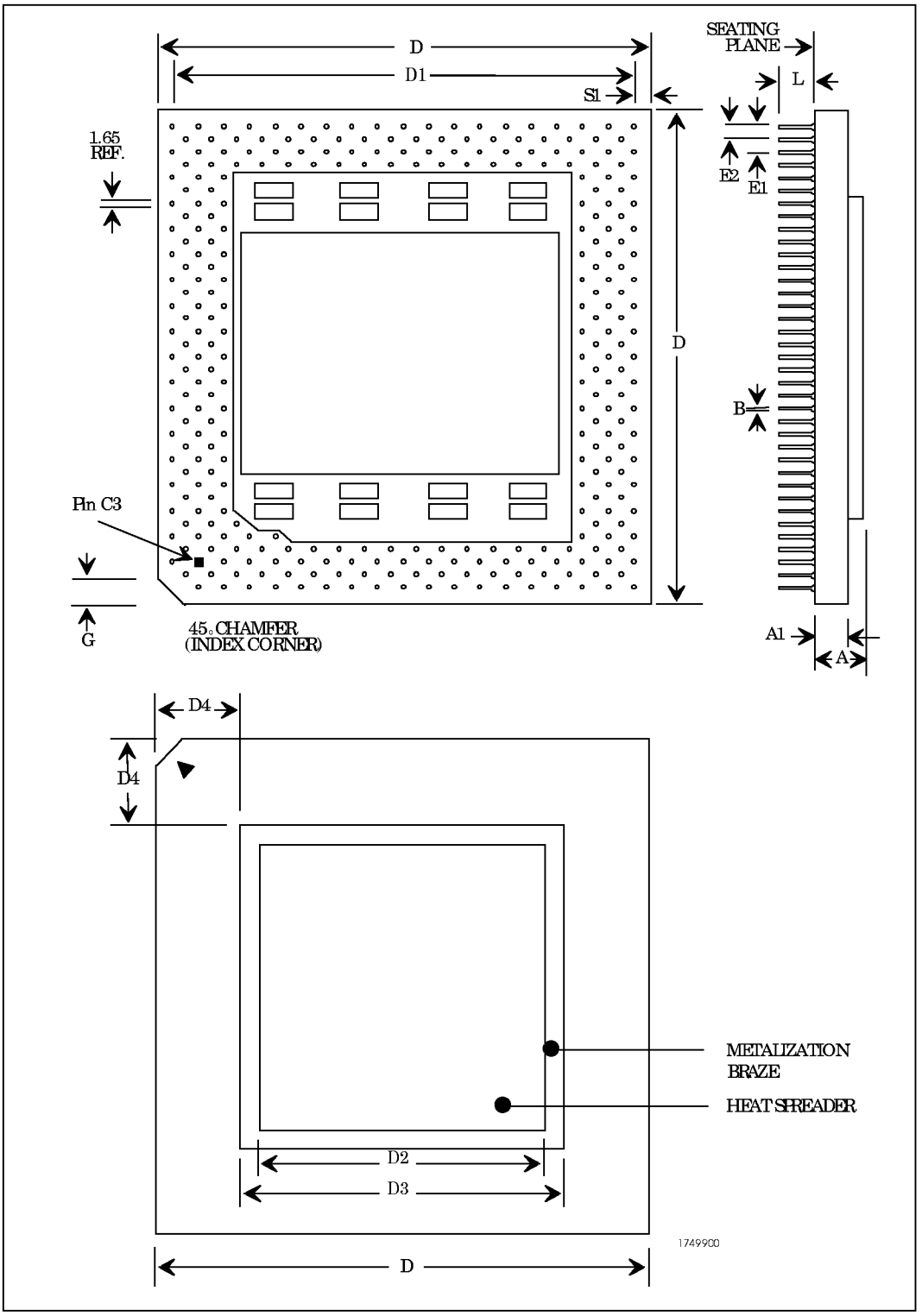


Figure 5-3. 296-Pin SPGA Package A

Table 5-3. 296-Pin SPGA Package A

SYMBOL	DESCRIPTION OF MEASURMENT	MILLIMETERS		INCHES	
		MIN	MAX	MIN	MAX
A	Seating plane to highest point on heat spreader	3.43	4.34	0.135	0.171
A1	Seating plane to highest point on package	2.51	3.07	0.099	0.121
B	Pin diameter	0.43	0.51	0.017	0.020
D	Overall package dimension	49.28	49.91	1.940	1.965
D1	Outer pin center to outer pin center	45.47	45.97	1.790	1.810
D2	Vert. and hoz. heat spreader, measured edge to edge.	31.37	32.13 Sq.	1.235	1.265
D3	Top metalization vert. and hoz., measured edge to edge	33.43	34.42	1.316	1.355
D4	Top metalization to top edge.	7.49	6.71	0.295	0.264
E1	First row/column linear pin spacing	2.41	2.67	0.095	0.105
E2	Next row/column linear pin spacing	1.14	1.40	0.045	0.055
G	Linear chamfer distance	1.52	2.29	0.060	0.090
L	Pin length, tip to seating plane	2.97	3.38	0.117	0.133
S1	Outer pin center to edge of package	1.65	2.16	0.065	0.085

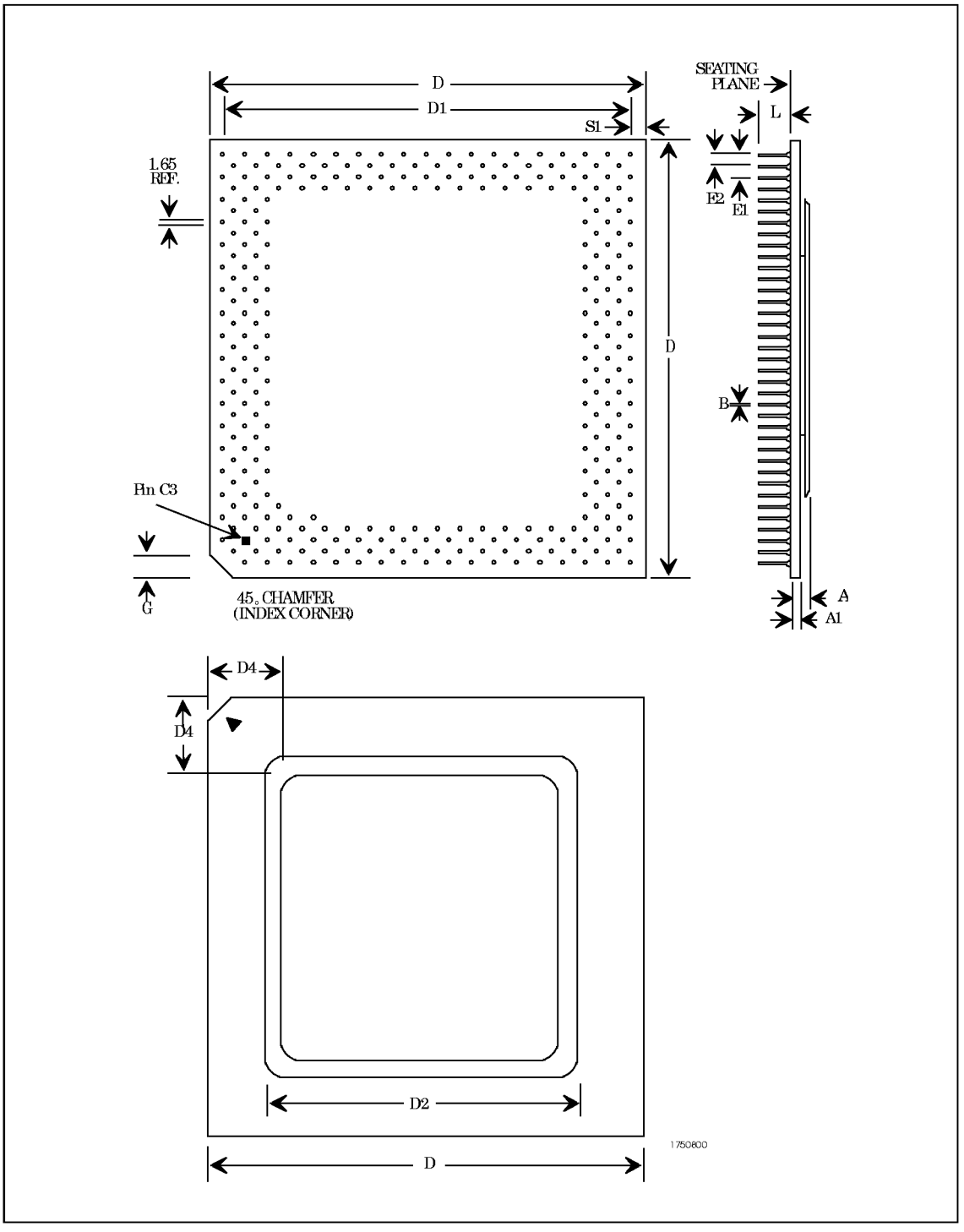


Figure 5-4. 296-Pin "Flip-Chip C4" SPGA Package B

Table 5-4. 296-Pin "Flip-Chip C4" SPGA Package B Dimensions

SYMBOL	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	3.80	4.50	0.150	0.177
A1	1.62	1.98	0.064	0.078
B	0.43	0.51	0.017	0.020
D	49.28	49.91	1.940	1.965
D1	45.47	45.97	1.790	1.810
D2	36.75 Sq.	37.25 Sq.	1.447	1.467
E1	2.41	2.67	0.095	0.105
E2	1.14	1.40	0.045	0.055
G	1.52	2.29	0.060	0.090
L	2.97	3.38	0.117	0.133
S1	1.65	2.16	0.065	0.085

5.2 Thermal Resistances

Three thermal resistances can be used to idealize the heat flow from the junction of the M II CPU to ambient:

θ_{JC} = thermal resistance from junction to case in °C/W

θ_{CS} = thermal resistance from case to heatsink in °C/W,

θ_{SA} = thermal resistance from heatsink to ambient in °C/W,

$\theta_{CA} = \theta_{CS} + \theta_{SA}$, thermal resistance from case to ambient in °C/W.

$T_C = T_A + P * \theta_{CA}$ (where T_A = ambient temperature and P = power applied to the CPU).

To maintain the case temperature under 70°C during operation θ_{CA} can be reduced by a heat-sink/fan combination. (The heatsink/fan decreases θ_{CA} by a factor of three compared to using a heatsink alone.) The required θ_{CA} to maintain 70°C is shown in Table 5-4. The designer should ensure that adequate air flow is maintained to control the ambient temperature (T_A).

Table 5-3. Required θ_{CA} to Maintain 70°C Case Temperature

Frequency (MHz)	Power* (W)	θ_{CA} For Different Ambient Temperatures				
		25°C	30°C	35°C	40°C	45°C
150	16.7	2.68	2.39	2.09	1.79	1.49
166	18.1	2.48	2.20	1.92	1.65	1.37
188	20.6	2.17	1.93	1.69	1.45	1.20
200	22.0	2.04	1.81	1.58	1.35	1.13
225	24.9	1.87	1.66	1.45	1.24	1.03
233	25.5	1.81	1.61	1.41	1.20	1.00
250	27.6	1.63	1.45	1.27	1.09	0.91

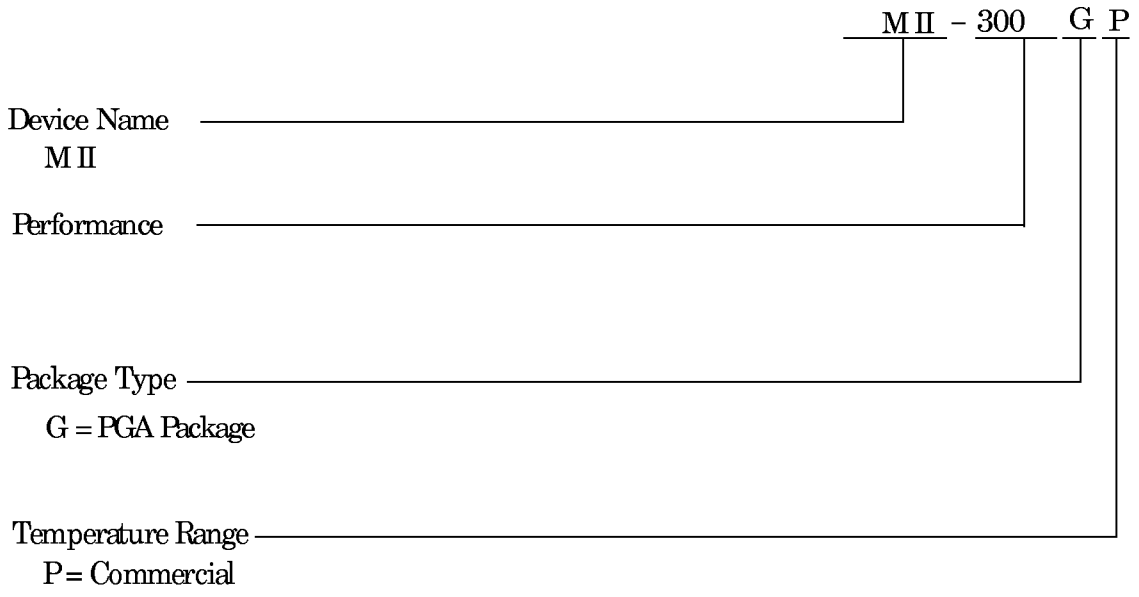
*Note: Power based on Max Active Power values from Table 4-6, Page 4-5.
Refer to the Cyrix Application AP105 titled "Thermal Design Considerations" for more information.

A typical θ_{JC} value for the M II 296-pin PGA-package value is 0.5 °C/W.

[redacted]

Cyrix®
Advancing the Standards

Ordering Information



Note: For further information concerning Performance Ratings, visit our website at www.cyrix.com.

The Cyrix M II CPU part numbers are listed below.

Cyrix M II™ Part Numbers

PART NUMBER	CLOCK MULTIPLIER	FREQUENCY (MHz)	
		BUS	INTERNAL
MII - 300GP	3.0	75	225
MII - 300GP	3.5	66	233
MII - 333GP	3.0	83	250

INDEX

Cyrix Worldwide Distributors

United States

Arrow
(800) 579-6633

Bell Industries
(800) 297-4948

DSS
(888) 377-1377

Hamilton
(800) 778-2668

Pioneer
(800) 585-7466

Wyle
(800) 588-9953

Europe

Cyrix Freephone Response Centre:

Germany	0130 813839
France	0800 908498
U.K.	0800 137305

All other countries:
+44 (0) 1756 702815

Austria

Karma
+43 1 27773100

Czech Republic

Karma Czech A.S.
+42 (0)2 66610268

Denmark

Karma ApS
+45 (0)3871 5056

Microtronica
+45 (0)44 508100

Finland

Karma Finland
+358 (0)9 2706221

Microtronica OY
+358 (0)9 7775751

France

Arrow Computer Products
+33 (0)1 60926060

Karma S.A.R.L.
+33 (0)1 46745656

Germany

Frank & Walter
+49 (0)531 2118 0

Karma Computer GmbH
+49 (0)2241 96260

Peacock AG
+49 (0)29 57790

Greece

Oktabit LTD. (Karma)
+30 (0)1 6011901

Italy

Karma Italia SRL.
+39 (0)2 3822051

Netherlands

Karma B.V.
+31 (0)70 3906545

Norway

Microtronica A/S
+47 (0)22 376020

Poland

Karma Zp. z.o.o
+48 (0)22 6180983

Portugal

Karma Portugal
+351 (0)1 4227180

Russia

Karma
+7 095 2825365

Spain

Karma Components S.A.
+34 (9)3 4738855

Sweden

Microtronica AB
+46 (0)8 6807800

Switzerland

Karma Components AG
+41 (0)41 7995799

Turkey

Karma
+90 (0)212 2333030

United Kingdom

Flashpoint
+44 (0)1628 606060

Karma UK Ltd
+44 (0)181 9009155

Microtronica
+44 (0)1189633872

Middle East & Africa

Turkey

Karma
+90 (0)212 2333030

South Africa

CDC (Karma)
+27 (0)11 7896410

United Arab Emirates

Karma Middle East
+971 (0)4 817680

Asia Pacific

Australia

Alepine Peripherals Pty Ltd
+61-2-9418-3322

Westan
+61-3-9563-6775

Hong Kong

AVT Industrial Ltd.
+852-2428-0008
E-mail:
francis@avt.com.hk
Web Site:
www.avt.com.hk

Daiwa System Ltd.
+852-2341-3351
E-mail:
stanleyn@daiwahk.com
Web Site:
www.daiwahk.com

India (Bombay)

American Components (S) Pte Ltd
+65-842-4100

Japan

Asahi Glass Co., Ltd.
+81-6-373-5852

H.Y. Associates Co., Ltd.
+81-3-3929-7111

Inno Micro Corporation
+81-45-476-7508

Kawasho Corporation
+81-3-3578-5192

Korea

Damon Electronics
+82-2588-5353

Malaysia

Planet Technology (M) Sdn Bhd
+60-3-987-9988
E-mail: ptmsb@po.jaring.my

Singapore

ARB Technologies
+65-281-3966
E-mail:
avispeng@singnet.com.sg

Cinergi Technology & Devices Pte Ltd
+65-778-8211
E-mail:
lkwong1@cyberway.com.sg

Taiwan

Princeton Technology Co.
+886-2-917-8856

Siltrontech Electronics Corp.
+886-2-652-2277

Thailand

Wati Intertrade Co. Ltd.
+662-719-7460
E-mail: ericlim@ksc9.th.com

Latin America

Brazil

GDE Inc. do Brasil Ltda.
+ 55 11 273.3300

Slice Com. Imp. Exp. de Componentes Eletr. Ltda.
+ 55 11 684.4697

Broad Marketing (Miami)
(305) 597-7700

MPS (Miami)
(305) 513-0222

Cyrix Worldwide Distributors

Cyrix U.S. Product Information

General Sales and Technical Support

800 462 9749 Sales and Technical Support
E-mail: tech_support@cyrix.com
Web: www.cyrix.com/support

Channel Sales and Technical Support

Cyrix Direct Connect (U.S. Channel Program)
800 215 6823 Sales and Literature ORders
800 340 0953 Technical Support
E-mail: tech_connect@cyrix.com
Web: www.cyrix.com/channel

Cyrix®
A National Semiconductor Company

Cyrix Corporation
2703 N. Central Expressway
Richardson, TX 75085
800 462 9749 Tel
972 968 8388 Tel

www.cyrix.com