# ISA/486



The 4021 CHIPSet

Databook

Revision 1.0

P R E L I M I N A R Y

9000-3808



CHIPS®

T-49-17-01



**Trademark Acknowledgement**

CHIPS® and the CHIPS logotype are registered trademarks of Chips and Technologies, Inc.

ISA/486™, 4021™, 4025™, SuperState V™, and Wingine™ are trademarks of Chips and Technologies, Inc.

IBM® and PC/AT® are registered trademarks and PS/2™ is a trademark of International Business Machines, Inc.



Chips and Technologies, Inc.
3050 Zanker Road
San Jose, California 95134
Phone: 408-434-0600

# The 4021 CHIPSet™

T-49-17-01

## Product Brief

The 4021 two-chip CHIPSet™ with CPU is designed for use in PC/AT® systems. It is 100 percent compatible with 486DX, 486DX2, 486SX, or 386DX processors or Super386™ CPUs without any component changes. The performance to cost ratio is optimized for each CPU option. This design allows the system designer flexibility and produces an easily upgradable system.

The 4021 provides a low cost high-perfomance non-cache solution for 486SX systems. It offers maximum performance benefits through the use of a CHIPS® -patented page interleaving algorithm.

The 4021 CHIPSet architecture provides an alternate, faster, and efficient bus interface for speed-sensitive devices like video, disks, or networks. This is to overcome the bus speed bottlenecks such as those associated with EISA/ISA systems that can degrade system performance.

## Features

- Supports 486DX™, 486DX2™, 486SX™, and Super386™ CPUs
- Supports 487 and 4167 in 486 CPU mode and 367, 3167 in 386 CPU mode.
- SuperState™ V support for the Super38605
- Burst Mode support in cache and DRAM modes
- Zero wait state Weitek® support
- 64K, 128K, 256K, and 512K cache size
- Direct-mapped, write back operation, with pipeline allocate on read miss
- Internal tag comparator
- Zero wait state writes
- 2-1-1-1 burst for cache reads to the 486 and 486SX
- Two-way interleaved data RAMs
- Posted writes to DRAM
- 3-1-1-1 burst read with 486SX CPU at 20MHz with no cache
- 256K, 1M, and 4M deep DRAM support
- Support 4Mx4 DRAMs with 12/10 or 11/11 addressing
- Support for 9-bit DRAM SIMS or 36-bit DRAM SIMMs
- Page mode RAM access with option of using page interleaving
- Up to 8 banks of 36-bit DRAMs
- CAS before RAS refresh for low power consumption
- Option of using asynchronous or synchronous AT clock source
- Classic, hidden, or disabled refresh
- Fast 12MHz AT bus support
- Dual-ported VRAM interface
- Local bus master and slave support

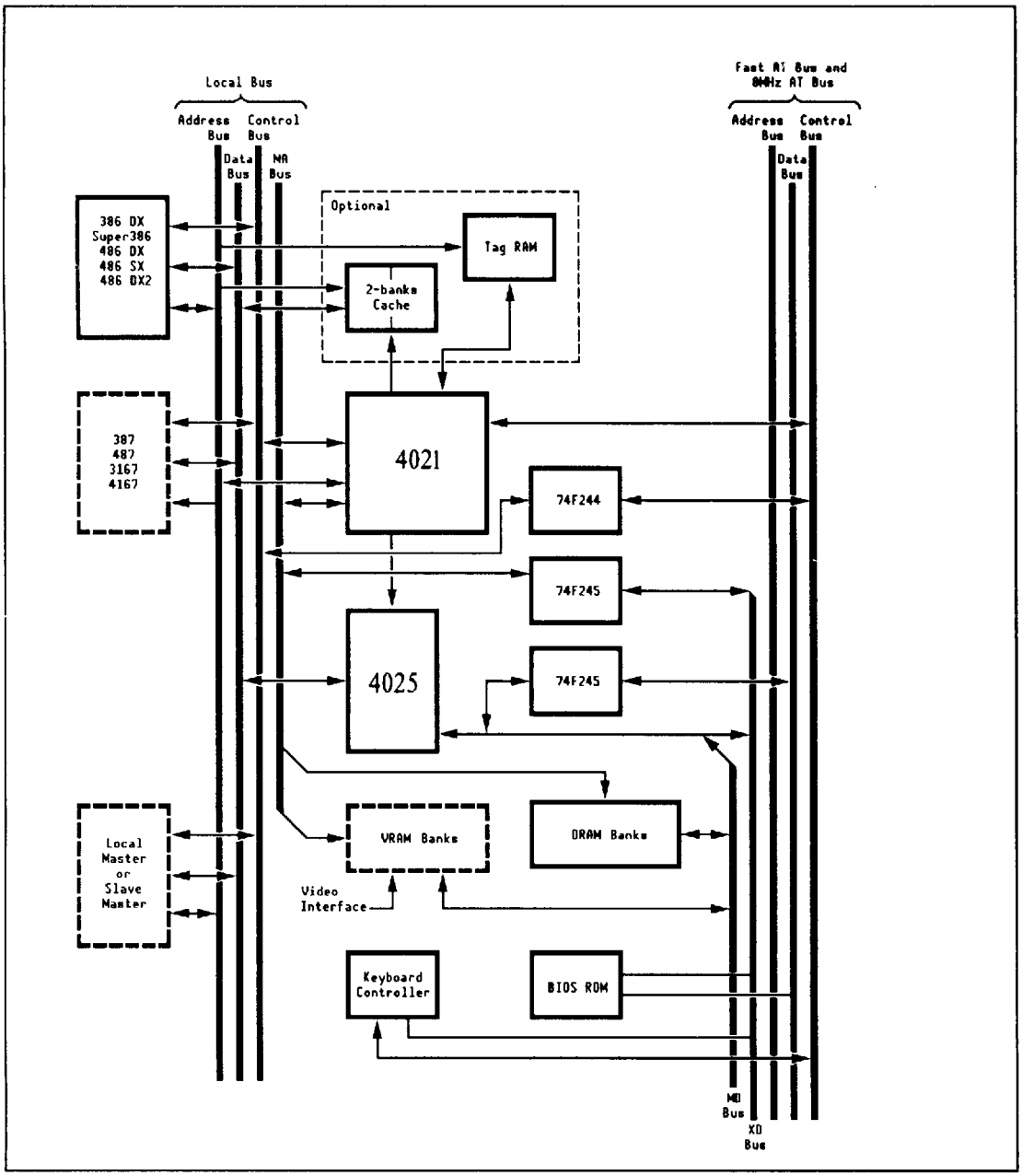
The 4021 CHIPSet has the following features:

Figure 1 shows a block diagram of the 4021 CHIPSet.



***Figure 1.*** *The ISA/486 4021 CHIPSet Block Diagram*

T-49-17-01



# Contents

T-49-17-01

T-49-17-01

## Chapter 7
## 4021 Cache Operation



## Chapter 8
## System Bus Arbitration

T-49-17-01

## Chapter 9
## The ISA Bus



## Chapter 10
## Address Mapping



## Chapter 11
## SuperState V™

T-49-17-01

## Chapter 12
## Advanced Device Interface



## Chapter 13
## Configuration Registers

## List of Figures

T-49-17-01



CHIPS & TECHNOLOGIES INC

## List of Registers

CHIPS & TECHNOLOGIES INC

T-49-17-01

## List of Tables

CHIPS & TECHNOLOGIES INC

CHAPTER 1

T-49-17-01

# Introduction

## Introducing The 4021 CHIPSet

The 4021 CHIPSet supports a diverse system architecture from a low cost 16MHz 486SX to the high-performance cache architecture of a 33 MHz 486DX or 40 MHz 386DX with the best performance and reliability. It provides the option of using the Fast AT bus or the Advanced Device Interface, which is independent of the system CPU.

The 4021 CHIPSet allows for a modular design for use with the 486SX CPU and the high-performance 386DX or 486DX CPU markets. Four basic systems are described in this section:

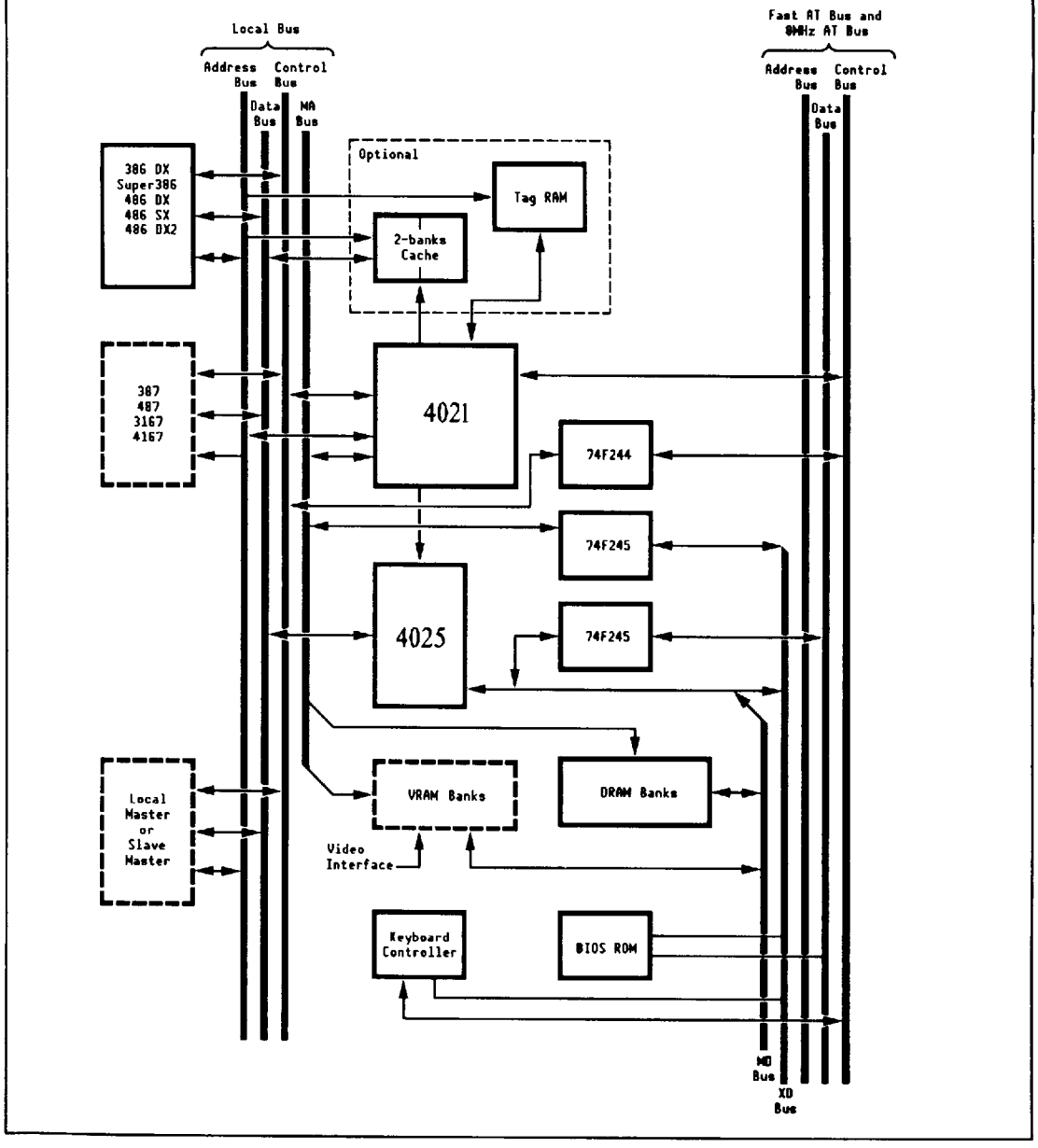
- Basic 486 non-cache system (16-25 MHz)
- 486 cache system (25-33 MHz)
- Cache 386DX or Super386 system (40 MHz)
- System with Advanced Device Interface

Figure 1-1 shows a system block diagram of the 4021 CHIPSet. Each of the dotted blocks in Figure 1-1 can be used as a building block within a unique product design.

*Figure 1-1. System Block Diagram*

T-49-17-01





T-49-17-01



## The Basic 486 Non-cache System (16 - 25 MHz)

The basic 486 non-cache system consists of the following components:

- 486SX, 486DX, and 486DX2 CPU
- 4021 CHIPSet
- System memory (DRAM)
- Keyboard controller
- BIOS ROM

The 486 non-cache system is a simple, low cost system that can yield performance comparable to a cache-based system. It can be designed with a 4-layer baby AT form factor system board. The system chip count is minimum in this configuration but it can be upgraded to any of the advanced device interfaces and a cache system.

## 486 Cache System (25-33 MHz)

The 486 cache system consists of the following components:

- 486SX, 486DX, and 486DX2 CPU
- 4021 CHIPSet
- System memory (DRAM)
- Data and tag RAM
- Keyboard controller
- BIOS ROM

This 486 cache system is the same as the basic 486 non-cache system but with the addition of cache (data and tag RAM). It can be designed with a 4-layer baby AT form factor system board. It can also be upgraded to any of the advanced device interfaces.

## Cache 386DX or Super386 System (25 - 40 MHz)

The cache 386DX or Super386 system consists of the following components:

- 386DX CPU
- 4021 CHIPSet
- System memory (DRAM)
- Data and tag RAM
- Keyboard controller
- BIOS ROM

This system can be designed with a 4-layer baby AT form factor system board, and can be upgraded to any of the advanced device interfaces.

T-49-17-01



# System with Advanced Device Interfaces

The 4021 CHIPSet provides four system memory interfaces to connect to external devices. The devices would normally interface through the EISA\ISA bus, like video, disk, or network cards.

## Local Bus Master

The 4021 CHIPSet supports a local bus master without requiring any external logic. The local bus master must meet the system CPU timing. The local bus master can access system memory at the system CPU speed and can achieve burst transfer rates of 200 Mbytes/sec. The CHIPSet handles all the arbitration between the local master and the various devices, and other buses.

## Local Bus Slave

The 4021 CHIPSet allows a local slave to map multiple system memory or I/O areas to the local slave memory.

## System VRAM Interface

The 4021 CHIPSet provides a high-speed interface for up to two banks of VRAMs. This improves the performance of a VGA chip over a standard AT or local interface type VGA.

## Fast AT Bus Interface

The fast AT bus interface is designed for external devices that need a fast, sustained transfer rate of up to 24 Mbytes/sec without major changes in the interface logic from the standard AT interface. A performance gain is achieved by either increasing the AT clock speed, or by forcing zero wait states for I/O cycles. The fast AT bus interface can be programmed for video memory, video I/O, and disk I/O space.

# Functional Description



This section provides block diagrams of the 4021 and 4025 CHIPSets. Figure 1-1 shows the 4021 block diagram. Figure 1-2 shows the 4025 block diagram.

T-49-17-01

***Figure 1-2.*** *4021 Block Diagram*

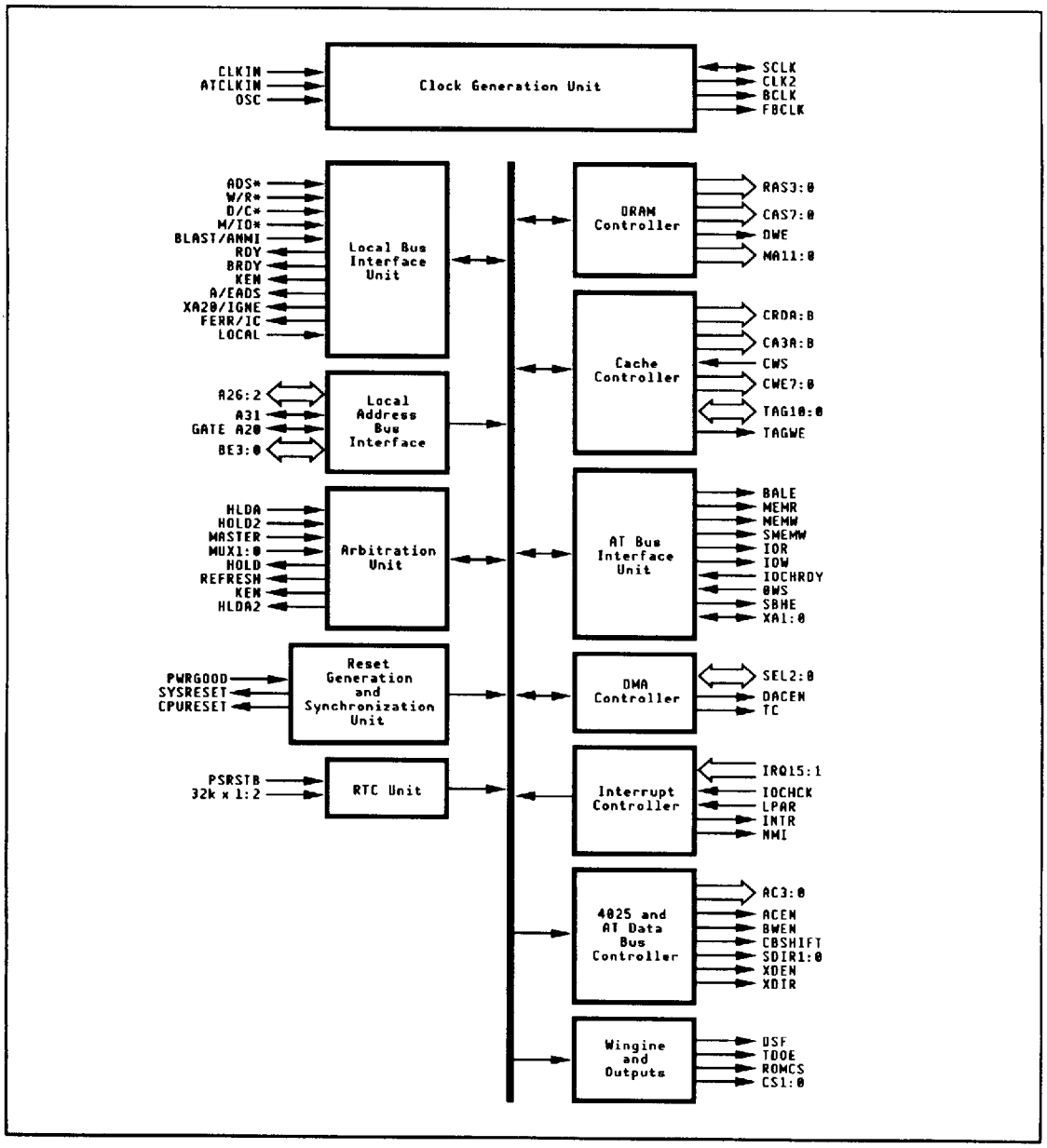
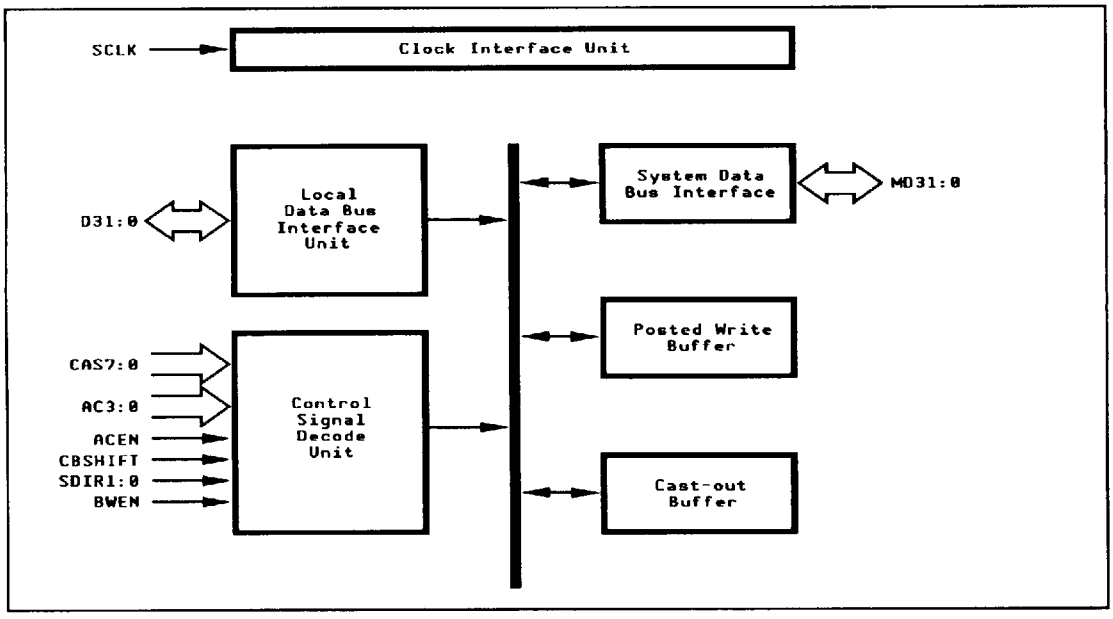


*Figure 1-3. 4025 Block Diagram*

T-49-17-01



T-49-17-01

# Pin Description

## The 4021 CHIPSet Pinouts

The 4021 CHIPSet is composed of a 4021 chip and a 4025 chip. Figure 2-1 shows the 4021 chip. Figure 2-2 shows the 4025 chip. Table 2-1 identifies the pin assignments for the 4021, and Table 2-2 identifies the pin assignments for the 4025.

***Figure 2-1.*** *4021 Pinout (Top View)*

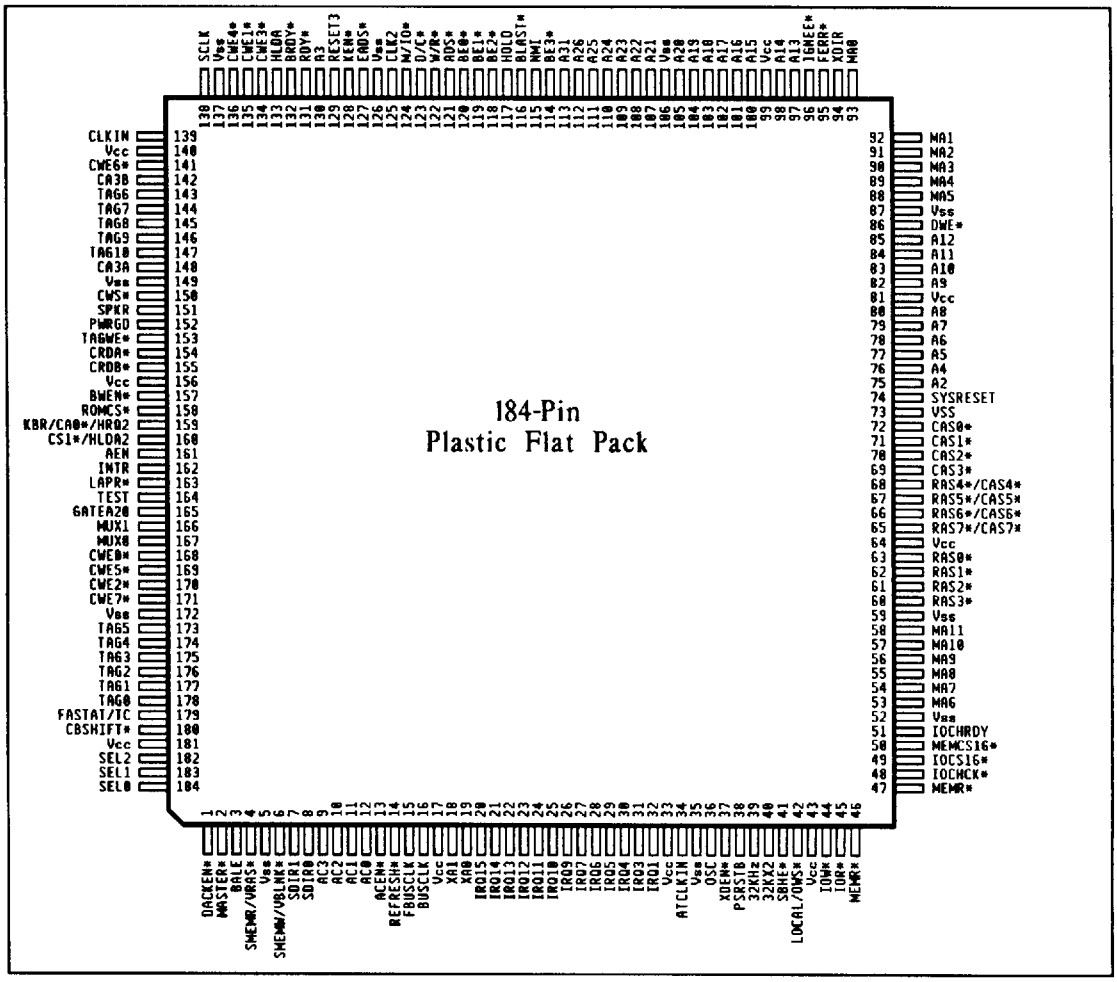


*Figure 2-2. 4025 Pinout (Top View)*

T-49-17-01

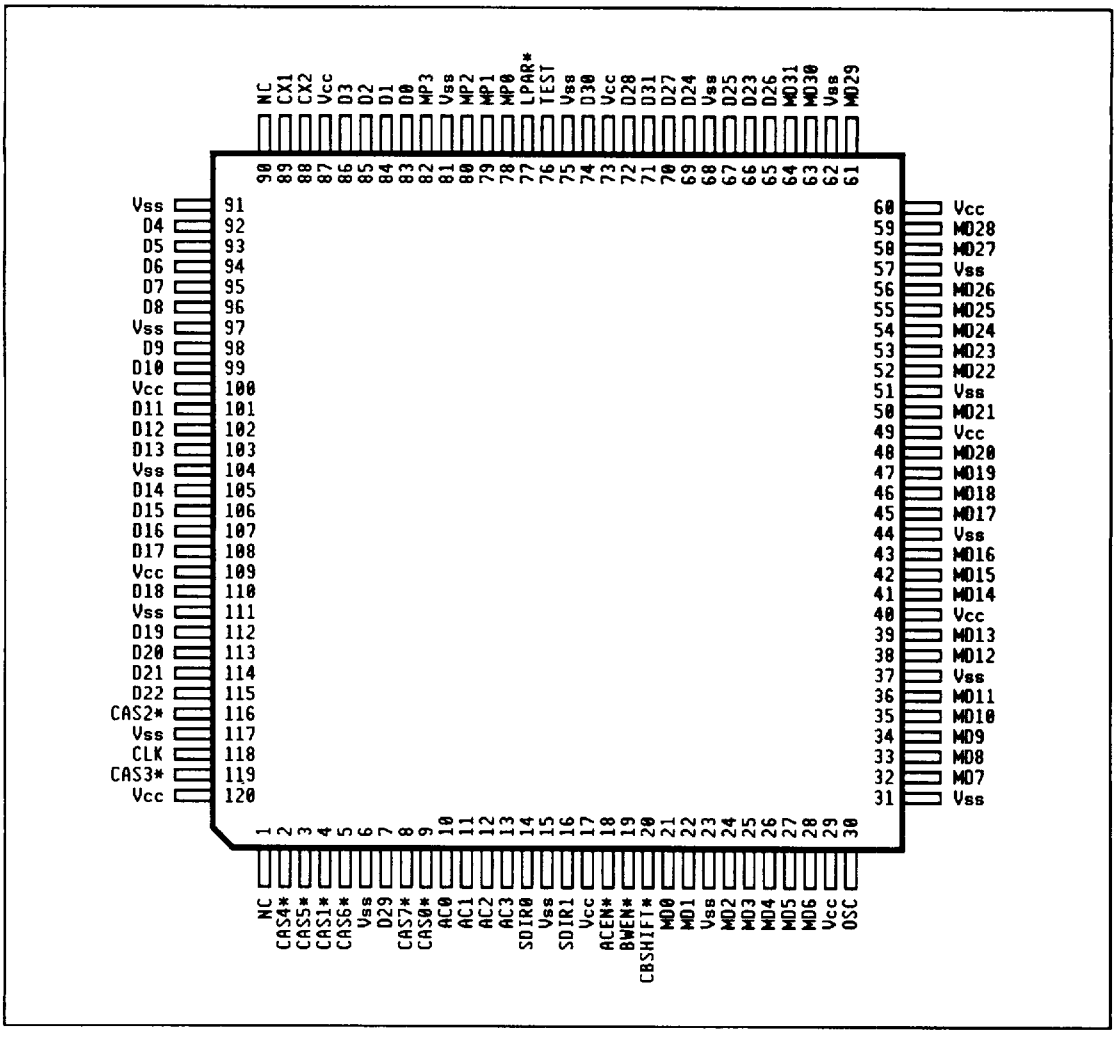
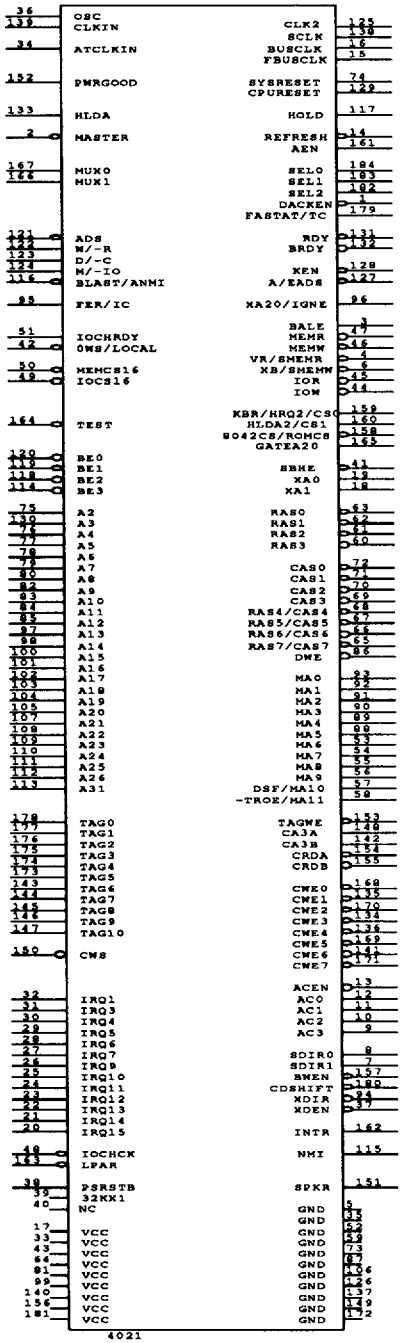


**Figure 2-3.** *The 4021 Schematic Symbols*

T-49-17-01





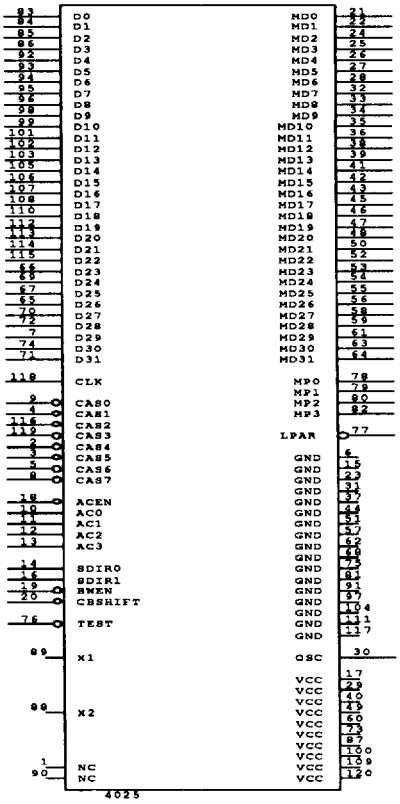



*Table 2-1. 4021 Pin Assignments*

T-49-17-01

| Signal | Pin No. | Signal | Pin No. | Signal | Pin No. |
|---|---|---|---|---|---|
| 32KHZ | 39 | BE0* | 120 | HLDA | 133 |
| NC | 40 | BE1* | 119 | HOLD | 117 |
| A2 | 75 | BE2* | 118 | IGNEE* | 96 |
| A3 | 130 | BE3* | 114 | INTR | 162 |
| A4 | 76 | BLAST* | 116 | IOCHCK* | 48 |
| A5 | 77 | BRDY* | 132 | IOCHRDY | 51 |
| A6 | 78 | BUSCLK | 16 | IOCS16* | 49 |
| A7 | 79 | BWEN* | 157 | IOR* | 45 |
| A8 | 80 | CA3A | 148 | IOW* | 44 |
| A9 | 82 | CA3B | 142 | IRQ1 | 32 |
| A10 | 83 | CAS0* | 72 | IRQ3 | 31 |
| A11 | 84 | CAS1* | 71 | IRQ4 | 30 |
| A12 | 85 | CAS2* | 70 | IRQ5 | 29 |
| A13 | 97 | CAS3* | 69 | IRQ6 | 28 |
| A14 | 98 | CBSHIFT* | 180 | IRQ7 | 27 |
| A15 | 100 | CLKIN | 139 | IRQ9 | 26 |
| A16 | 101 | CLK2 | 125 | IRQ10 | 25 |
| A17 | 102 | CRDA* | 154 | IRQ11 | 24 |
| A18 | 103 | CRDB* | 155 | IRQ12 | 23 |
| A19 | 104 | CS1*/HLDA2 | 160 | IRQ13 | 22 |
| A20 | 105 | CWE0* | 168 | IRQ14 | 21 |
| A21 | 107 | CWE1* | 135 | IRQ15 | 20 |
| A22 | 108 | CWE2* | 170 | KBR/CS0*/HRQ2 | 159 |
| A23 | 109 | CWE3* | 134 | KEN* | 128 |
| A24 | 110 | CWE4* | 136 | LOCAL/0WS* | 42 |
| A25 | 111 | CWE5* | 169 | LPAR* | 163 |
| A26 | 112 | CWE6* | 141 | M/IO* | 124 |
| A31 | 113 | CWE7* | 171 | MA0 | 93 |
| AC0 | 12 | CWS* | 150 | MA1 | 92 |
| AC1 | 11 | D/C* | 123 | MA2 | 91 |
| AC2 | 10 | DACKEN* | 1 | MA3 | 90 |
| AC3 | 9 | DWE* | 86 | MA4 | 89 |
| ACEN* | 13 | EADS* | 127 | MA5 | 88 |
| ADS* | 121 | FASTAT/TC | 179 | MA6 | 53 |
| AEN | 161 | FBUSCLK | 15 | MA7 | 54 |
| ATCLKIN | 34 | FERR* | 95 | | |
| BALE | 3 | GATEA20 | 165 | | |

*Table 2-1. 4021 Pin Assignments (continued)*

T-49-17-01



| Signal | Pin No. | Signal | Pin No. | Signal | Pin No. |
|---|---|---|---|---|---|
| MA8 | 55 | ROMCS* | 158 | $V_{cc}$ | 17 |
| MA9 | 56 | SBHE* | 41 | $V_{cc}$ | 33 |
| MA10/DSF | 57 | SCLK | 138 | $V_{cc}$ | 43 |
| MA11/TROE* | 58 | SDIR0 | 8 | $V_{cc}$ | 64 |
| MASTER* | 2 | SDIR1 | 7 | $V_{cc}$ | 81 |
| MEMCS16* | 50 | SEL0 | 184 | $V_{cc}$ | 99 |
| MEMR* | 47 | SEL1 | 183 | $V_{cc}$ | 140 |
| MEMW* | 46 | SEL2 | 182 | $V_{cc}$ | 156 |
| MUX0 | 167 | SMEMR*/VRAS* | 4 | $V_{cc}$ | 181 |
| MUX1 | 166 | SMEMW*/XREQ* | 6 | $V_{ss}$ | 5 |
| NMI | 115 | SPKR | 151 | $V_{ss}$ | 35 |
| OSC | 36 | SYSRESET | 74 | $V_{ss}$ | 52 |
| PSRSTB | 38 | TAG0 | 178 | $V_{ss}$ | 59 |
| PWRGOOD | 152 | TAG1 | 177 | $V_{ss}$ | 73 |
| RAS0* | 63 | TAG2 | 147 | $V_{ss}$ | 87 |
| RAS1* | 62 | TAG3 | 176 | $V_{ss}$ | 106 |
| RAS2* | 61 | TAG4 | 175 | $V_{ss}$ | 126 |
| RAS3* | 60 | TAG5 | 174 | $V_{ss}$ | 137 |
| RAS4*/CAS4* | 68 | TAG6 | 173 | $V_{ss}$ | 149 |
| RAS5*/CAS5* | 67 | TAG7 | 143 | $V_{ss}$ | 172 |
| RAS6*/CAS6* | 66 | TAG8 | 144 | W/R* | 122 |
| RAS7*/CAS7* | 65 | TAG9 | 145 | XA0 | 19 |
| RDY* | 131 | TAG10 | 146 | XA1 | 18 |
| REFRESH* | 14 | TAGWE* | 153 | XDEN* | 37 |
| RESET3 | 129 | TEST | 164 | XDIR | 94 |

*Table 2-2. 4025 Pin Assignments*

T-49-17-01



| Signal | Pin No. | Signal | Pin No. | Signal | Pin No. |
|---|---|---|---|---|---|
| AC0 | 10 | D18 | 110 | MD21 | 50 |
| AC1 | 11 | D19 | 112 | MD22 | 52 |
| AC2 | 12 | D20 | 113 | MD23 | 53 |
| AC3 | 13 | D21 | 114 | MD24 | 54 |
| ACEN* | 18 | D22 | 115 | MD25 | 55 |
| BWEN* | 19 | D23 | 66 | MD26 | 56 |
| CAS0* | 9 | D24 | 69 | MD27 | 58 |
| CAS1* | 4 | D25 | 67 | MD28 | 59 |
| CAS2* | 116 | D26 | 65 | MD29 | 61 |
| CAS3* | 119 | D27 | 70 | MD30 | 63 |
| CAS4* | 2 | D28 | 72 | MD31 | 64 |
| CAS5* | 3 | D29 | 7 | MP0 | 78 |
| CAS6* | 5 | D30 | 74 | MP1 | 79 |
| CAS7* | 8 | D31 | 71 | MP2 | 80 |
| CBSHIFT* | 20 | LPAR* | 77 | MP3 | 82 |
| CLK | 118 | MD0 | 21 | NC | 1 |
| CX1 | 89 | MD1 | 22 | NC | 90 |
| CX2 | 88 | MD2 | 24 | OSC | 30 |
| D0 | 83 | MD3 | 23 | SDIR0 | 14 |
| D1 | 84 | MD4 | 26 | SDIR1 | 16 |
| D2 | 85 | MD5 | 27 | TEST | 76 |
| D3 | 86 | MD6 | 28 | $V_{cc}$ | 17 |
| D4 | 92 | MD7 | 32 | $V_{cc}$ | 40 |
| D5 | 93 | MD8 | 33 | $V_{cc}$ | 49 |
| D6 | 94 | MD9 | 34 | $V_{cc}$ | 60 |
| D7 | 95 | MD10 | 35 | $V_{cc}$ | 73 |
| D8 | 96 | MD11 | 36 | $V_{cc}$ | 87 |
| D9 | 98 | MD12 | 38 | $V_{cc}$ | 100 |
| D10 | 99 | MD13 | 39 | $V_{cc}$ | 109 |
| D11 | 101 | MD14 | 41 | $V_{cc}$ | 120 |
| D12 | 102 | MD15 | 42 | $V_{cc}$ | 29 |
| D13 | 103 | MD16 | 43 | $V_{ss}$ | 6 |
| D14 | 105 | MD17 | 45 | $V_{ss}$ | 13 |
| D15 | 106 | MD18 | 46 | $V_{ss}$ | 23 |
| D16 | 107 | MD19 | 47 | $V_{ss}$ | 31 |
| D17 | 108 | MD20 | 48 | $V_{ss}$ | 37 |

*Table 2-2. 4025 Pin Assignments (continued)*



| Signal | Pin No. | Signal | Pin No. | Signal | Pin No. |
|---|---|---|---|---|---|
| $V_{ss}$ | 44 | $V_{ss}$ | 68 | Vss | 104 |
| $V_{ss}$ | 51 | Vss | 75 | Vss | 111 |
| $V_{ss}$ | 57 | Vss | 81 | Vss | 117 |
| $V_{ss}$ | 62 | Vss | 97 | Vss | 91 |

## Signal Description

The following sections describe all of the pin signals for the 4021 CHIPSet and the 4025 CHIPSet. An asterisk (*) indicates that the signal is active when low. The signal direction input (I), output (O), or bidirectional (I/O) is also noted.

T-49-17-01

## Clocks

| Signal Name | Pin No. | Type | Description |
|---|---|---|---|
| OSC | 36 | I | 14.31818MHz clock input for the timer<br>From an oscillator or the output of the crystal circuit in the 4025. |
| CLKIN | 139 | I | CPU clock input from oscillator<br>In 1x clock mode this is a 1x clock. In 2x clock mode this is a 2x clock. |
| CLK2 | 125 | I | 2x clock pin<br>Normally receives a 2x clock. This is required for 386 operation and when using the 486 in 2x DRAM modes. If only 1x DRAM modes are used with the 486, this pin can receive a 1x clock. |
| SCLK | 138 | I | 1x clock pin<br>Input from the external clock drivers. This clock should be at the same phase and delay as the 486 clock. |
| ATCLKIN | 34 | I | Oscillator input for an asynchronous AT clock<br>May be divided by 1, 2, or 3 internally to form the 2x AT state machine clock. It will be generally connected to a 16MHz oscillator if FASTAT is not implemented, and 48MHz if FASTAT is implemented. If CLKIN is used as the source for the ISA bus clocks this pin need not be connected (it should be pulled up). |
| BUSCLK | 16 | O | AT bus clock<br>May be derived from CLKIN or ATCLKIN. It is normally close to 8 MHz. Between 6 and 8.33 MHz is recommended. |
| FBUSCLK | 15 | O | Fast AT bus clock<br>Goes to the fast slots when FASTAT is implemented on the board. This pin will switch between the normal and fast AT clocks. The fast clock will normally be about 12 MHz. |

## Resets

T-49-17-01

| Signal Name | Pin No. | Type | Description |
|---|---|---|---|
| PWRGOOD | 152 | I | Power up reset input<br>Asynchronous. When low, all outputs are Hiz and all inputs are internally forced high (there are a few exceptions which are noted in the RESET section of this document). Generates CPURESET and SYSRESET. |
| CPURESET | 129 | O | Reset to the CPU only<br>Active for hard resets (initiated by PWRGOOD) and soft resets (soft resets may be redirected to ANMI* for the Super386). CPURESET will go low during the second half of a T state to properly sync the 386 and Super386 to the clock phase. |
| SYSRESET | 74 | O | Reset to everything except the CPU<br>Activated only by PWRGOOD low. Has the same timing as CPURESET. |

## Arbitration

T-49-17-01

| Signal Name | Pin No. | Type | Description |
|---|---|---|---|
| HOLD | 117 | O | HOLD to the CPU<br>Synchronous to the clock. |
| HLDA | 133 | I | HOLD acknowledge from the CPU. |
| KBR*/CS0*/<br>HRQ2 | 159 | I/O | Hold request from a local bus master<br>This is a synchronous input. This pin is multiplexed for two more functions. These are not time multiplexed functions and hence only one of them may be used in a system. The other functions are a programmable chip select or the keyboard reset input from the keyboard controller. This pin function option is selected by a configuration register. At powerup it is an unused input, and need not be pulled low. It is configured as HRQ2 by software. If used as a chip select a 10K pullup should be connected to prevent the peripheral from being selected before the pin is configured by software. |
| CS1*/HLDA2* | 160 | O | Hold acknowledge to a local bus master<br>This signal is synchronous to the 1x clock. This pin may optionally be converted to a programmable chip select by a configuration register. At powerup this pin will be HLDA2*, and will be high, so no pullup is necessary when used as a chip select. |
| MASTER* | 2 | I | ISA bus signal indicating that an ISA master has the bus. |
| REFRESH* | 14 | I/O | Refresh<br>Active for refresh cycles. A bus master may pull this pin low to cause a refresh cycle. Must have an external pullup. Pseudo open collector (drives high for a short period of time before floating). |
| AEN | 161 | O | Address enable<br>High during DMA cycles to indicate that I/O devices should disable their decode. Also high for refresh cycles. Low for AT master cycles. |

T-49-17-01

| Signal Name | Pin No. | Type | Description |
|---|---|---|---|
| SEL2:0 | 182, 183, 184 | O | Multiplexer enable signal<br>The SEL1:0 are used as the select lines on the DREQ multiplexer. SEL2:0 are used along with DACKEN* to encode the DACKs* and FLUSH*. |

| SEL2:0 | Decoded Output | SEL2:0 | Decoded Output |
|---|---|---|---|
| 0 | DACK0* | 4 | FLUSH* |
| 1 | DACK1* | 5 | DACK5* |
| 2 | DACK2* | 6 | DACK6* |
| 3 | DACK3* | 7 | DACK7* |

| | | | |
|---|---|---|---|
| DACKEN* | 1 | O | DACK* enable<br>This is the output enable for the external 3 to 8 demultiplexer for DACKx* and FLUSH*. |
| MUX1:0 | 166/167 | I | DREQx and TURBO multiplexed inputs<br>An external 74F153 multiplexer encodes the DREQx and TURBO signal as the MUX1:0 for the CHIPSet. The enable of the external multiplexer is controlled by SEL1:0. When TURBO is low, the CPU runs slower, as defined by the Performance Control logic. The encoded MUX1:0 output is as follows: |

| SEL1:0 | MUX0 | MUX1 |
|---|---|---|
| 0 | DREQ0 | TURBO |
| 1 | DREQ1 | DREQ5 |
| 2 | DREQ2 | DREQ6 |
| 3 | DREQ3 | DREQ7 |

| | | | |
|---|---|---|---|
| FASTAT/TC | 179 | O | DMA terminal count<br>When FASTAT mode is implemented, this pin is the FASTAT pin (low for normal speed, high for high speed) when the CPU or local bus master has control. When a DMA channel has the bus (including master cycles) this pin is terminal count. Two F08 gates and an inverter on the outside decode the separate signals. These gates are only required when FASTAT mode is implemented in a system. |

## CPU Control

T-49-17-01

| Signal Name | Pin No. | Type | Description |
|---|---|---|---|
| ADS* | 121 | I/O | CPU address strobe<br>Input during CPU or local master cycle. Output during DMA and ISA master cycles for 38605 internal cache invalidation, and local bus slaves. |
| W/R* | 122 | I/O | CPU write/read signal<br>Input during CPU or local master cycle. Output during DMA and AT master cycles for 38605 cache invalidation and for local bus slaves. |
| D/C* | 123 | I/O | CPU data/code signal<br>Input during CPU or local master cycle. Output during DMA and AT master cycles for 38605 cache invalidation and for local bus slaves. |
| M/IO* | 124 | I/O | CPU memory and I/O signal<br>Input during CPU or local master cycle. Output during DMA and AT master cycles for 38605 cache invalidation and for local bus slaves. |
| RDY* | 131 | I/O | Pseudo open collector READY<br>Input during local slave cycle. Output during 386 CPU, non-cachable 486 cycles, local master or AT cycles. |
| BRDY* | 132 | I/O | Burst ready. Input during local slave cycle. READY output during 486 cachable cycles. |
| KEN* | 128 | O | Cache enable to the CPU<br>The 4021 drives this pin low at the start of all cycles. It will be driven high only for non-cacheable cycles. |
| BLAST*/ANMI* | 116 | | Burst last/Alternate nonmaskable interrupt request.<br>**486:** Burst last from the CPU. When low, the cycle will be the last cycle of a burst.<br>**38605:** ANMI* output pin. Controlled by the ANMI* timer and soft reset redirection for 38605 Super state functions. 38605 CPU needs a 10k pull-up on this signal.<br>**386:** unused input. |
| EADS*/AADS* | 127 | I/O | External ADS/Alternate address strobe.<br>**486:** External ADS. Driven during DMA and bus master memory writes. Used to invalidate the 486 CPU internal cache line.<br>**38605:** AADS* input indicating the difference between user and SuperState V accesses.<br>**386:** unused input. |

T-49-17-01

## Coprocessor Control

| Signal Name | Pin No. | Type | Description |
|---|---|---|---|
| IGNEE*/XA20 | 96 | I/O | Ignore numeric error/Gated A20<br>**486:** Ignore numeric error. Connects to the 486 or 487SX CPUs.<br>**38605 and 386:** XA20. Gated A20 for the ISA bus. Must be buffered by an external F245. |
| FERR*/INTR | 95, 162 | I/O | Floating-point error/Clear interrupt<br>**486:** Floating-point error. Used to generate IRQ13.<br>**38605 and 386:** INTCLR to the coprocessor PAL. Active for I/O writes to F0 and F1 to clear the coprocessor interrupt. |

## ISA Bus Signals

| Signal Name | Pin No. | Type | Description |
|---|---|---|---|
| BALE | 3 | O | ISA bus ALE. |
| MEMR* | 47 | I/O | ISA bus memory read<br>Output during CPU access to ISA bus. Input for ISA master cycles. |
| MEMW* | 46 | I/O | ISA bus memory write<br>Output during CPU access to ISA bus. Input for ISA master cycles. |
| SMEMR*/ VRAS* | 4 | O | Memory read strobe/VRAM RAS . Dual function, selected by a configuration register.<br>**SMEMR*** Memory read strobe for the bottom megabyte. Always driven.<br>**VRAS*** VRAM RAS for Windows video support. |
| SMEMW*/ XREQ* | 6 | I/O | Memory write strobe/blank video controller signal. Dual function, selected by a configuration register.<br>**SMEMW*** Memory write strobe for the bottom megabyte. Always driven.<br>**XREQ*** Blank signal from the video controller (possibly modified to include the field information). Used for the windows video support to determine when to do a VRAM transfer cycle. Horizontal and vertical information is derived from this signal. |

T-49-17-01

| Signal Name | Pin No. | Type | Description |
|---|---|---|---|
| IOR* | 45 | I/O | ISA I/O read<br>Input for ISA master cycles. |
| IOW* | 44 | I/O | ISA I/O write<br>Input for ISA master cycles. |
| IOCHRDY | 51 | I/O | Ready indicator from the ISA bus. Driven low for accesses by DMA and ISA masters to local bus peripherals, and sometimes for ISA master accesses to system memory. |
| LOCAL*/0WS* | 42 | I | ISA slave can accept a 0WS cycle/local bus slave can handle the cycle.<br>Dual function. At the beginning of a CPU access this is the LOCAL pin. A local bus slave drives this signal to indicate to the 4021 that it will handle the cycle. During an AT cycle, it is pulled low by ISA slaves which can accept a 0WS cycle. This signal is synchronous to the bus clock. The LOCAL* function must be enabled by a configuration register. |
| MEMCS16* | 50 | I/O | 16-bit ISA memory indicator<br>The 4021 drives it during DMA and ISA Master cycles when its local DRAM is addressed. |
| IOCS16* | 49 | I | 16-bit ISA I/O indicator. |

T-49-17-01

## Address

| Signal Name | Pin No. | Type | Description |
|---|---|---|---|
| BE3:0* | 114, 118-120 | I/O | Byte enables from the CPU<br>Driven for AT Master and DMA accesses. |
| A2, A3<br>A8-A4<br>A12-A9<br>A14-13<br>A19-A15<br>A21-A23 | 75, 130<br>80-76<br>85-82<br>98-97<br>104-100<br>109-107 | I/O | Local address bus from the CPU<br>These pins are outputs during DMA cycles. A7:2 are output for classic refresh cycles. |
| A20 | 105 | I/O | Input during 486 CPU cycles and output for DMA cycles. See the GATEA20 section for more details. |
| A26:24, A31 | 110, 113<br>112, 111 | I/O | These are inputs for CPU cycles and are driven low when DMA or an AT master has control to provide the invalidation address to the CPU. |
| GATEA20 | 165 | I/O | GATEA20 signal to/from the keyboard controller With 8042 GATEA20 emulation and all of the processor types, there are several uses and connections for this pin. See *Chapter 4: The CS4021 GATEA20 Function* for more details. |
| SBHE* | 41 | I/O | ISA bus byte high enable<br>Output except during master cycles. |
| XA1:0 | 18-19 | I/O | Address bits created from the byte enables and bus conversion logic. They are buffered by an external 245 before going to the ISA bus. Input during ISA master cycles. |
| IGNEE*/XA20 | 96 | I/O | Muxed with the IGNEE pin (see the coprocessor section above). XA20 is used in 386 and 38605 systems. |
| ROMCS*/<br>8042CS* | 158 | O | This signal is the OR of ROM chip select and 8042 chip select. Active only for CPU cycles. |

T-49-17-01

## Cache

| Signal Name | Pin No. | Type | Description |
|---|---|---|---|
| CA3a,<br>CA3b | 148<br>142 | O | Address line A3 for each of the cache banks<br>In order to achieve a full speed burst using standard SRAMs, A3 must be controlled separately for the two banks of interleaved SRAMs. |
| CRDa*,<br>CRDb* | 154<br>155 | O | Cache read<br>Connected to OE* of cache data chips. One goes to each interleaved bank. |
| CWE7:0* | 171, 141,<br>169, 136,<br>134, 170,<br>135, 168 | O | Cache write enables<br>CWE3:0* are for the even doubleword, and CWE7:4* for the odd double word. One for each byte. Connected directly to the cache data RAM WE* pins. |
| CWS* | 150 | I | Cache write strobe<br>Connected to an advanced 1x clock for the 486, and directly to the 1x clock for the 386 and 38605. It provides the pulse for the CWE* pins. |
| TAGWE* | 153 | O | Tag RAM write enable<br>Connected to the WE* pins of the tag RAM(s). |
| TAG5:0<br>TAG10:6 | 173-178<br>147-143 | I/O | Tag bits 0-10<br>Bit 10 is the write back DIRTY bit. Tied directly to Tag RAM data pins. These pins are used to write the new tag data, to read the tag address during a write back, and as the data path for testing the tag RAMs. They are also the inputs to the comparator to generate the internal hit signal. |

## DRAM

T-49-17-01

| Signal Name | Pin No. | Type | Description |
|---|---|---|---|
| RAS3:0* | 60-63 | O | One RAS for each bank<br>Buffered with F244s (one gate each). |
| RAS7:4*/<br>CAS7:4* | 65-68 | O | Dual function pins:<br>4 RAS mode: CAS7:4* used for banks 1 and 3.<br>8 RAS mode: RAS7:4* must be externally buffered. |
| CAS3:0* | 69-72 | I/O | CAS3:0* are used for banks 0 and 2 and the video RAMs in 4 RAS mode. They are used for all banks in 8 RAS mode. Must be externally buffered. |
| DWE* | 86 | O | DRAM write enable<br>Must be externally buffered. |
| MA5:0<br>MA7:6 | 88-93<br>54-53 | I/O | DRAM address bits 7-0; I/O data bus. Must be buffered with F244s. |
| MA11:8 | 58-55 | O | DRAM address bits 8-10<br>Must be buffered with F244s. MA10 and MA11 are dual function pins. MA10 can be programmed to be the VRAM DSF pin for Wingine. MA11 can be the VRAM transfer request for Wingine. |

## Buffer Control

T-49-17-01

| Signal Name | Pin No. | Type | Description |
|---|---|---|---|
| ACEN* | 13 | O | Action code enable<br>Active during ISA bus cycles. Provides the latch timing during ISA reads. Active for the entire write cycles. Always low for DMA and ISA master cycles. |
| AC3:0 | 9-12 | O | Action codes, controlling the 4025 data buffer function. |
| SDIR0 | 8 | O | SD bus direction for SD7:0. 0=from SD to MD bus. |
| SDIR1 | 7 | O | SD bus direction for SD15:8. 0=from SD to MD bus. |
| BWEN* | 157 | O | Write buffer enable<br>This signal, when low, causes the write buffer in the 4025 to clock in the data from the local bus on the rising edge of the 1X clock. It is normally low, and goes high when the 4025 should NOT clock in new data. |
| CBSHIFT* | 180 | O | Castout buffer fifo shift<br>When low the 4 dword deep castout buffer in the 4025 is shifted on the rising edge of SCLK. Used to shift data in and out of the castout buffer. |
| XDIR | 94 | O | XD bus direction<br>1=SD to XD (the default direction), 0=XD to SD. |
| XDEN* | 37 | O | Enables the 245 between the MA bus and the XD bus. |

T-49-17-01

## Interrupts

| Signal Name | Pin No. | Type | Description |
|---|---|---|---|
| IRQ1,IRQ7:3, IRQ15:9 | 32, 27-31 20-26 | I | Interrupts<br>IRQ13 is for the Weitek in a 486 system. It is the input from the coprocessor PAL in 386 and 38605 systems. It is ORed with the internal 486 coprocessor interrupt. |
| INTR | 162 | O | Interrupt request to the CPU. |
| NMI | 115 | O | Non-maskable interrupt to the CPU. |
| IOCHCK* | 48 | I | AT bus error indication signal. |
| LPAR* | 163 | I | Local parity error signal from the 4025. |
| SPKR | 151 | O | Speaker output. |

## Real-Time Clock

| Signal Name | Pin No. | Type | Description |
|---|---|---|---|
| PSRSTB | 38 | I/O | Power strobe for the RTC<br>Indicates that no battery power exists and the CMOS RAM is corrupted. |
| 32KHZ | 39 | I | 32KHz clock input<br>Input side of the Xtal circuit. |
| No Connection | 40 | O | No connection. |

## Test, Power, and Ground

| Signal Name | Pin No. | Type | Description |
|---|---|---|---|
| TEST* | 164 | I | Test mode; always one. |
| $V_{cc}$ | 17, 33, 43, 64, 81, 99, 140, 156, 181 | | Power. |
| GND ($V_{ss}$) | 5, 35, 52, 59, 73, 87, 106, 126, 137, 149, 172 | | Ground. |

## 4025 Pin Description

T-49-17-01

| Signal Name | Pin No. | Type | Description |
|---|---|---|---|
| D31:0 | 71-74, 7, 72, 70, 65, 67, 69, 66,115-112, 110, 108-105, 103-101, 99, 98, 96-92, 86-83 | I/O | Local data bus. |
| MD31:0 | 64, 63, 61, 59, 58, 56-52, 50, 48-45, 43-41, 39, 38, 36-32, 28-26, 23, 24, 22, 21 | I/O | Memory data bus. |
| MP3:0 | 82, 80, 79, 78 | I/O | DRAM parity bits. |
| LPAR* | 77 | O | Local parity error<br>This is the OR of the 4-byte parity error signals. This signal will go inactive on the next memory read with proper parity for the byte which caused the error. |
| CLK | 118 | I | 1x clock for write buffer and read buffer. |
| CAS7:0* | 8, 5, 3, 2, 119, 116, 4, 95 | I | CAS lines from the 4021, with the byte enables encoded. Parity is checked on the rising edge of CAS for DRAM read cycles. Only the bytes for which the CAS is active have their parity checked. The CAS lines are also used to control the read latch for CPU DRAM reads. |
| ACEN* | 18 | I | Action code enable<br>This signal follows the AT IOR*, MEMR*, or INTA* signals for CPU AT bus reads. For CPU AT bus writes, it goes low before the IOW* or MEMW* signals to provide data setup time. During DMA and bus master cycles, it is always low. |
| AC3:0 | 13-10 | I | Encoded action codes<br>Used for data bus steering and bus conversion. |

T-49-17-01

| Signal Name | Pin No. | Type | Description |
|---|---|---|---|
| SDIR0 | 14 | I | SD bus direction for SD7:0. Used for bus steering of the MD7:0 lines. |
| SDIR1 | 16 | I | SD bus direction for SD15:8. Used for bus steering of the MD15:8 lines. |
| BWEN* | 19 | I | Buffered Write enable<br>When low, the 4025 clocks data from the local data bus into the write buffer (on the rising edge of the clock). This signal is normally low and goes high to hold data in the write buffer. |
| CBSHIFT* | 20 | I | Cast out buffer shift<br>When low the cast out buffer is shifted one position on the rising edge of the clock. Used to both shift in and out the data. |

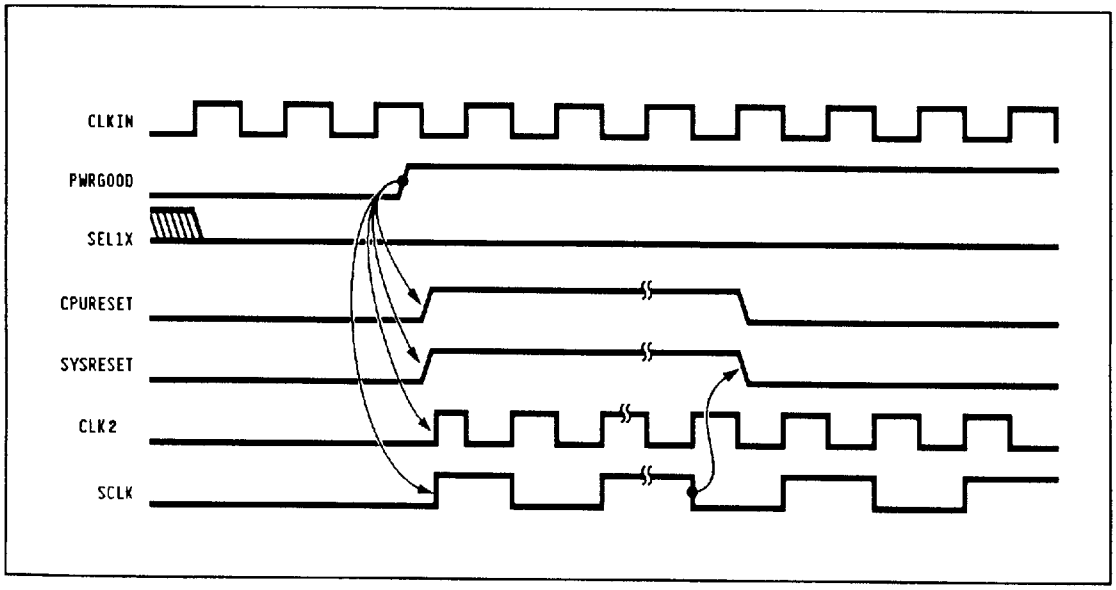
T-49-17-01

| Signal Name | Pin No. | Type | Description |
|---|---|---|---|
| CX1 | 89 | I | 14.31818MHz Xtal input. |
| CX2 | 88 | O | 14.31818MHz Xtal output. |
| OSC | 30 | O | 14.31818MHz output. Drives the bus and the 4021. |
| TEST* | 76 | I | Test mode. |
| $V_{cc}$ | 17, 40, 49, 60, 73, 87, 100, 109, 120, 29 | | Power. |
| GND | | | Ground. |
| NC | 90, 1 | | No connection. |

# The 4021 RESET Logic

The 4021 CHIPSet receives two reset inputs: PWRGOOD and KBRESET*. PWRGOOD is the power-ON reset signal, usually derived from the power supply or a reset switch. Only PWRGOOD initiates a power-up reset clearing all internal registers to their default state. It generates both CPURESET and SYSRESET as shown in Figure 3-1 and 3-2.

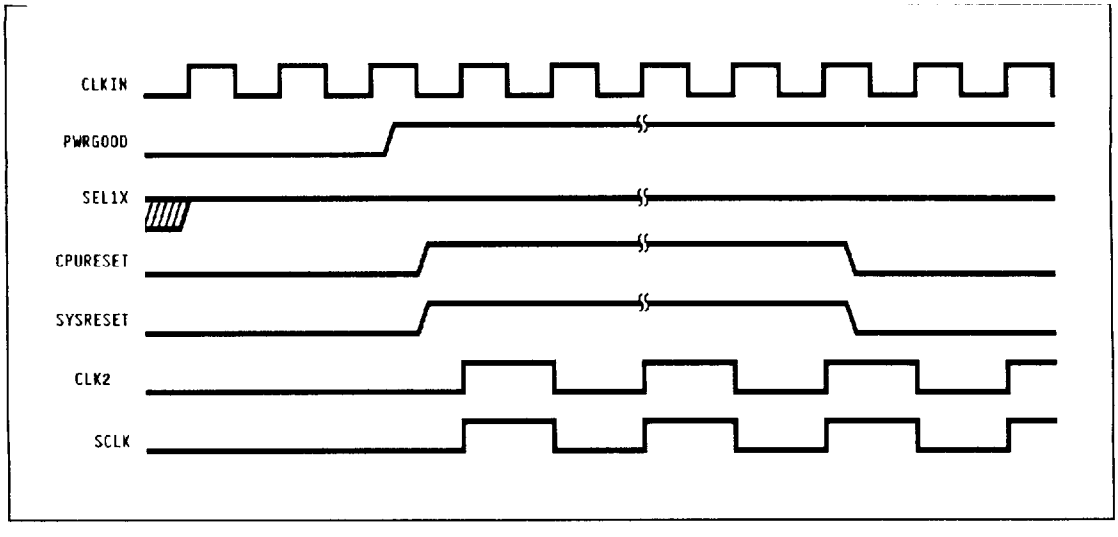
***Figure 3-1.*** *Generating CPURESET and SYSRESET*



T-49-17-01

*Figure 3-2. PWRGOOD Reset*






KBRESET* is the CPU reset request signal from the keyboard controller. It only generates CPURESET. Both PWRGOOD and KBRESET* are asynchronous inputs. KBRESET* is synchronized in the 4021 CHIPSet and sampled on every SCLK rising edge. KBRESET* is muxed with HRQ2 and CS0*. The system must be configured to use any one of these functions.

CPURESET is the reset signal to the system CPU only. SYSRESET is intended for resetting the rest of the system, and is only active on power-up, or when the Reset button is pushed. The CPURESET and SYSRESET falling edges occur during Pi2 to synchronize the 386, 387, and 2x Super386 CPUs to the proper clock phase.

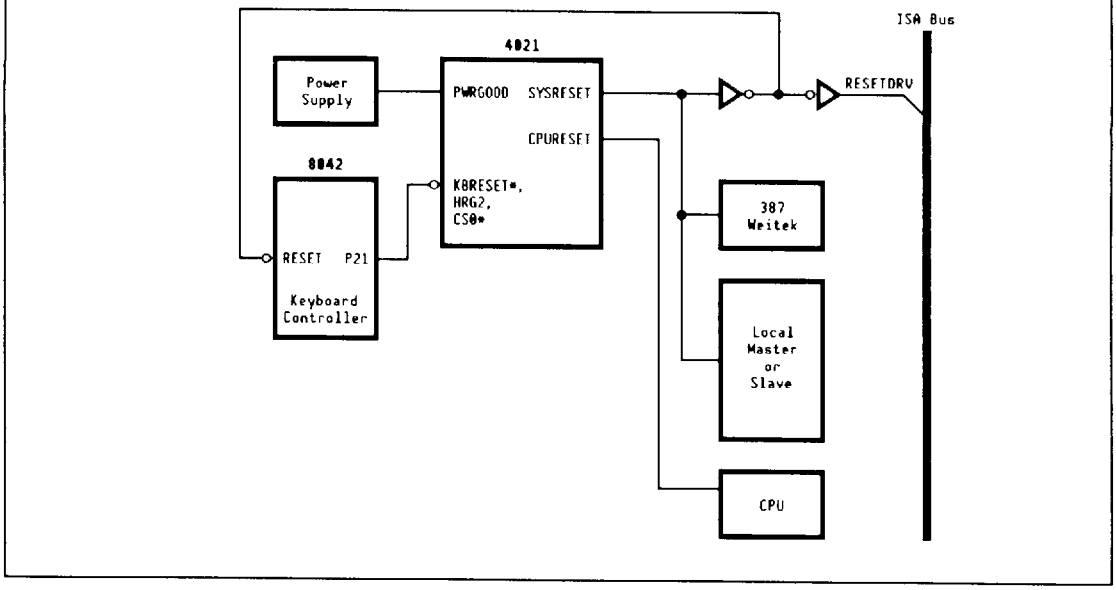
CPURESET and SYSRESET signals last for at least 1mS after PWRGOOD goes high. This allows proper initialization of peripherals, such as the 8042 keyboard controller, and to allow the 486 VCO to stabilize. This extended reset interval occurs only at power-up or Reset button closures, and does not affect performance.

The system level RESET connections are shown in Figure 3-3. The SYSRESET to the keyboard controller requires an inverter.

**Figure 3-3.** *System Level RESET Connections*

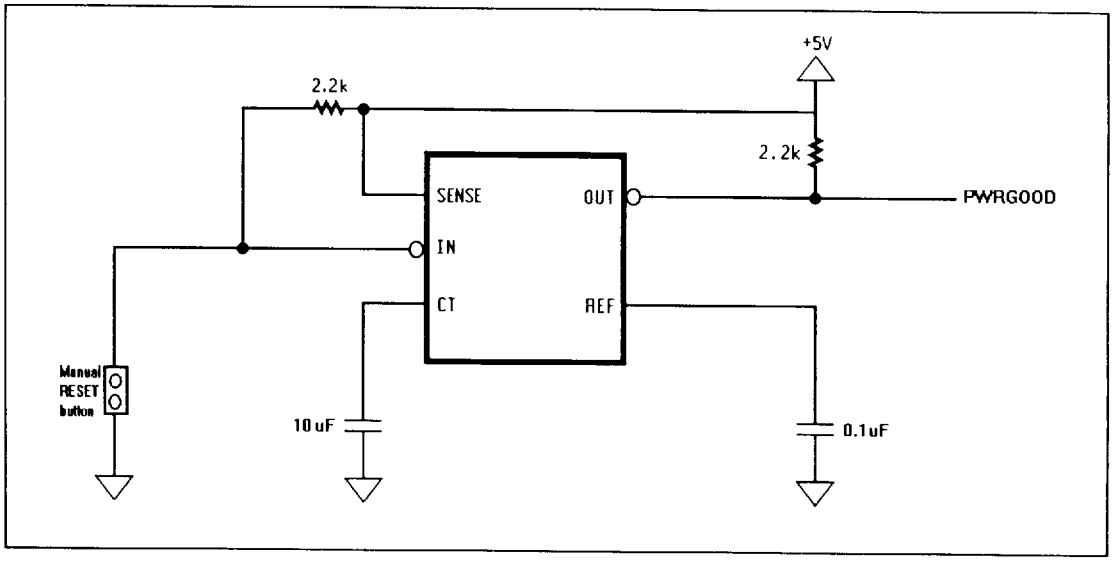
T-49-17-01






The PWRGOOD reset to the 4021 CHIPSset can be connected directly from the PWRGOOD signal of the power supply or from a voltage sense circuit. An example is shown in Figure 3-4.

**Figure 3-4.** *Example of PWRGOOD Reset Circuit*



T-49-17-01

## Powering Up the Chip

The 4021 contains a real-time clock, and thus is never really powered down. Therefore, to prevent power consumption when the the system is powered off (PWRGOOD is low), all of the outputs are floated, and all inputs are forced to a high state.

The following exceptions occur for signals that keep the real-time clock, as well as those that configure the CHIPSet on power-up:

- 32KHz (an input) is not forced high.
- PWRGOOD is not forced high.
- CLK2 and NMI (SEL1X input at powerup) is not forced high at power-up.
- CPURESET and SYSRESET are floated and have an internal pull-up resistor to keep the CPU and the rest of the system under reset.

The NMI signal from the 4021 CHIPSet is muxed during PWRGOOD high and SYSRESET high (active) as the SEL1X input to the 4021 CHIPSet.

The SEL1X signal provides the power-up clock input type information. After PWRGOOD goes high, SEL1X remains an input, is sampled and latched internally. Once SEL1X is sampled, the CHIPSet can determine if the input clock is a 2x or a 1x clock.

CPURESET and SYSRESET are driven high for 1mS after PWRGOOD goes high (active).

## CPURESET Generation

CPURESET is generated at power-up and may be generated by a software request when the CPU must exit protected mode. A configuration register bit disables all the alternate methods of generating CPURESET, making it functionally identical to SYSRESET. This mode is used on the Super386 when in SuperState V mode. The alternate sources of CPURESET generate an ANMI* instead. See *Chapter11: SuperState V* for more detailed information regarding SuperState V operation.

Four methods of generating CPURESET are provided:

T-49-17-01

1. KBRESET* input.
2. Simulated 8042 generating KBRESET* internally.
3. Port 92 bit 0.
4. Shutdown cycle from the CPU.

KBRESET* is from the 8042. It is normally high, and pulses low when the 8042 is issued a command by the software to issue a CPU reset. The 4021 samples PWRGOOD on every SCLK rising edge. When it is sampled active, it generates KBRESET* seven SCLKs later.

The simulated 8042 receives the I/O commands to the 8042 and provides a KBRESET* signal within the chip. The 4021 ORs the KBRESET* signal with the KBRESET* from the input pin. By blocking the 8042 KBRESET* commands, performance is improved.

Port 92, bit 0 (compatible with Micro Channel designs), provides a CPU reset when it goes from 0 to 1.

CHIPS & TECHNOLOGIES INC

*Note:* Port 92 bit 0 must actually transition from 0 to 1.

Immediately following a system reset, writing a 1 is sufficient, but the next time, a 0 must be written first. The rising edge of IOW* writes the data into Port 92. When bit 0 goes from 0 to 1, a CPURESET is generated 206 SCLKs later.

When the CPU does a SHUTDOWN bus cycle, a reset request is generated. The 4021 generates a one wait-state RDY* and generates a CPURESET three SCLKs later. The SHUTDOWN is encoded as shown in Table 3-1.

T-49-17-01

*Table 3-1. SHUTDOWN Bus Cycle Encoding*



| CPU | M/IO* | D/C* | W/R* | Address | BE3:0* | BE3* | BE2* | BE1* | BE0* |
|---|---|---|---|---|---|---|---|---|---|
| 486 | 0 | 0 | 1 | 00000000 | E | 1 | 1 | 1 | 0 |
| 386/Super386 | 1 | 0 | 1 | 00000000 | E | 1 | 1 | 1 | 0 |

The pulse width of CPURESET is at least 16 SCLKs for a "soft" reset, and slightly more than 1mS for a "hard" reset (initiated by PWRGOOD). The 486 and the 386 CPUs require the "soft" resets to be at least 15 clocks.

The 386 requires 80 clocks to initiate a selftest. A self test is never initiated for a "soft" reset because this has a severe performance impact. The 387 PAL prevents the self-test from occurring on soft resets.

T-49-17-01

# The GATEA20 Function

## GATEA20 Function

The GATEA20 function masks the CPU-generated address A20 to the system memory and drives it low, if GATEA20 is low. This is done by the software in a 286 IBM AT. The software, when required to set A20 low (to make the address roll back to zero at the 1Mb boundary), would write to the keyboard controller which would in turn generate the GATEA20 signal. The masking of the CPU-generated address A20 is achieved by simply logical ANDing of the GATEA20 from the keyboard controller and A20 from the CPU. This gated signal is now the final A20 address to the system memory.

## GATEA20 in a 486 and 38605 CPU

The gating of address bit 20 with GATEA20 is handled by the 486 and 38605 CPUs. The existence of a cache in the CPU dictates that the gating be done in the CPU since the CPU cache must receive the "gated" version. The 4021 CHIPSet provides these CPUs with the gating signal GATEA20 which is an OR function of all the sources that can generate GATEA20. This is called A20M* in these CPUs. The CPU ANDs it with the raw CPU A20 to provide the "gated" version. This gated version is what is driven out of these CPUs as A20.

## GATEA20 in a 386 and 38600 CPU

The 386 and 38600 do not gate address bit A20 and GATEA20 internally like the 486 or 38605. Consequently, the 4021 does the gating for the system memory and ISA bus. Local bus slaves can get the CPU A20 directly if they do not use the first 64K or the first 64K above 1 MB in the memory map.

For the 386 and 38600 the 4021 also requires two pins, GATEA20 from the keyboard controller and the gated A20 to the ISA bus (XA20).

These two schemes use different logic inside the 4021 and have different system level connections for A20M*/XA20 as shown in Figures 4-1 to 4-3, and as described in Table 4-1.

*Figure 4-1. 486 CPU GATEA20 Connection*





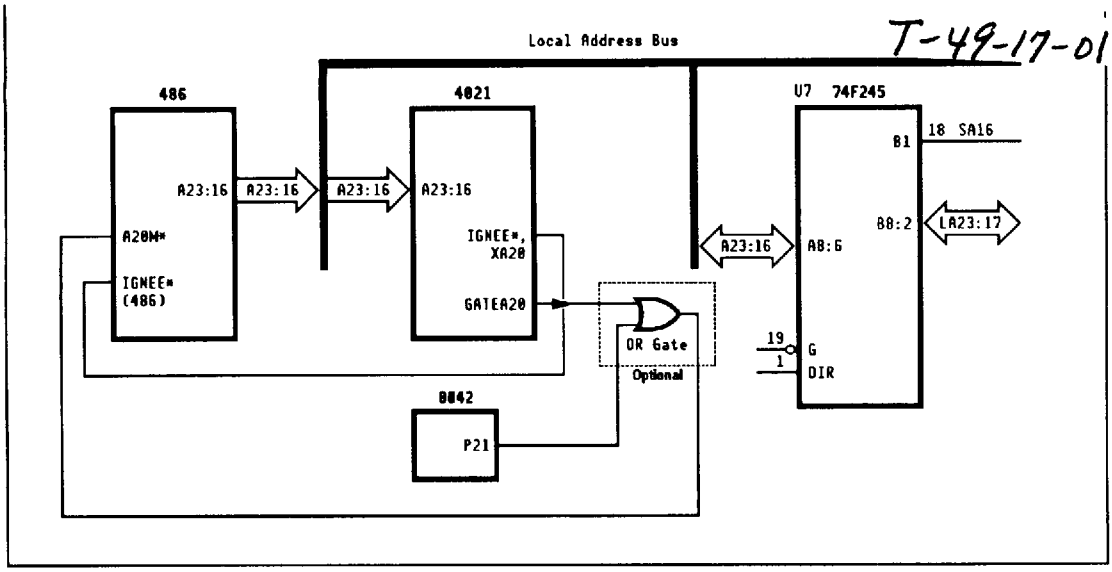



*Figure 4-2. Super386 CPU GATEA20 Connection*

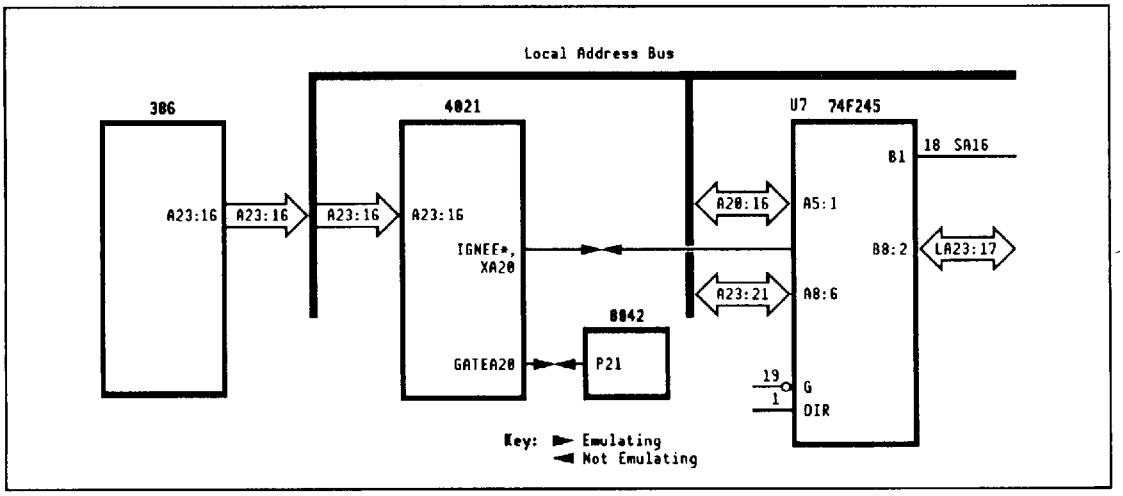


*Figure 4-3. 38605 CPU GATEA20 Connection*

T-49-17-01



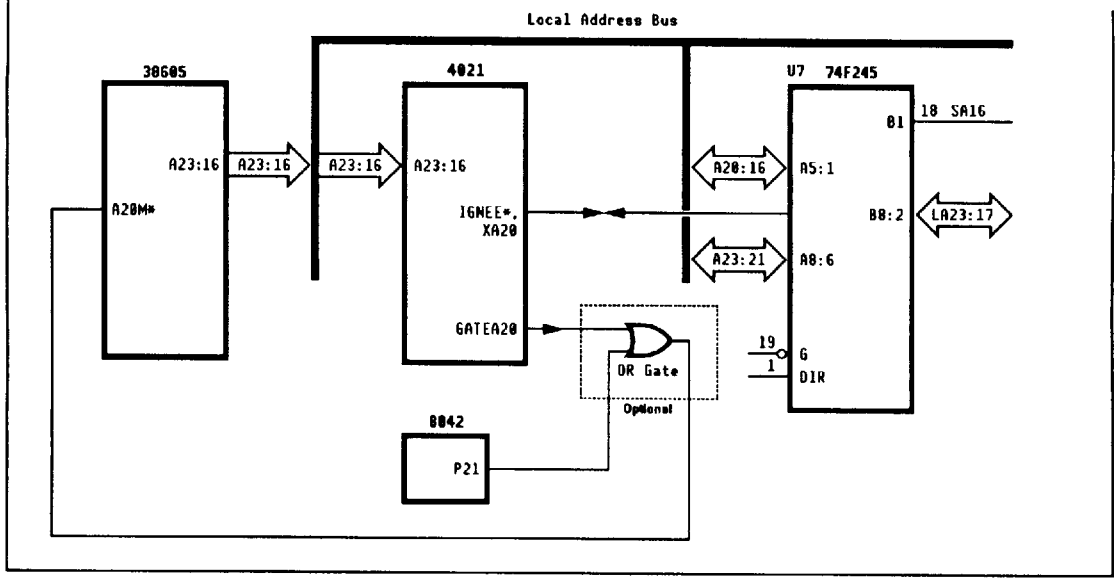



*Table 4-1. A20 Generation Sources*

| 4021 Pin | Generic | 386 | 38605 | 486 |
|---|---|---|---|---|
| A20 during CPU/Loc master cycle | Input | Input | Input | Input |
| XA20 during CPU/Loc master cycle | Output | Output | Output | Output (-IGNEE) |
| A20 during DMA cycle | Output | Output | Output | Output |
| XA20 during DMA cycle | Output | Output | Output | Output (-IGNEE) |
| A20 during ISA master cycle | Input (don't care) | Output | Output | Input |
| XA20 during ISA master cycle | Input (don't care) | Input | Input | Output (-IGNEE) |

T-49-17-01

## GATEA20 Generation Sources

The 4021 supports fast GATEA20, as well as optionally emulating the keyboard controller's GATEA20. GATEA20 is controlled from three sources. The sources are ORed together in the 4021. They include:

- Port 92 bit 1
- The keyboard controller (8042)
- The Simulated keyboard controller (8042)

Port 92 bit 1 is the PS2 type "Fast" GATEA20 bit. The I/O port bit is used directly.

If used, the keyboard controller (8042) is ORed with the 4021 GATEA20 signal, externally in the case of the 486 and 38605, and internally in the case of the 386.

If the simulated keyboard controller mode is ON, GATEA20 is controlled by the keyboard I/O ports. It may also be set or reset from a configuration register, regardless of the state of the simulated keyboard controller mode. Initially, this bit is high (like the 8042).

The keyboard controller powers up with GATEA20 high, which is required for the system to boot properly from location FFFFFFF0. The keyboard controller logic in the 4021 CHIPSet powers up high, for proper emulation. The Port 92 GATEA20 bit powers up low. In the boot process, the BIOS takes the keyboard controller GATEA20 low. When a program enters protected mode, one of the GATEA20 sources is taken high. After exiting protected mode, the same source is taken low again.

## Simulated Keyboard Controller (8042)

The CPU reset and GATEA20 functions are normally done by the 8042 keyboard controller. The CPU issues I/O commands to the 8042 keyboard controller to manipulate these signals. The 4021 may be configured to monitor these signals and simulate the RESET and GATEA20 functions of the keyboard controller. Optionally, the commands may be blocked to the keyboard controller in order to speed up the operation.

The commands are written to the keyboard controller through address 64. The commands, which effect GATEA20 and KBRESET*, are shown in Table 4-2.



*Table 4-2. Keyboard Controller Commands*

T-49-17-01

| I/O Address | Command | Action |
|---|---|---|
| 64 | AA | Self Test. KBRESET* and GATEA20 go high. |
| 64 | D1 | Write Output Port. Next byte written to Port 60 is the data. |
| 60 | -- | Bit 0=KBRESET*, bit 1=GATEA20. |
| 64 | Fx | Pulse output ports.<br>F0, F1, F4, F5, F8, F9, FC, and FD take GATEA20 high.<br>All of the even ones cause a KBRESET*. |

To take GATEA20 high, a program could use either the D1 or Fx codes (pulsing GATEA20 will pulse it low, then high, leaving it high). To take GATEA20 low, Write Output Port must be used. The selftest must be decoded simply because it effects the pin.

A CPU reset requires the Fx command. Using the D1 command to initiate the reset keeps it low forever, revivable only with a hard reset.

The Fx and D1 commands are optionally blocked from the keyboard controller (F3, F7, FB and FF are not blocked). The 4021 does this by not issuing an IOW* for them. If a D1 command occurs and is blocked, the following write to the Port 60 data register is also blocked. If the D1 command is aborted and another command is issued without Port 60 being written, that command is not blocked (unless it is another D1 or one of the blocked Fx commands). Port 60 writes are only blocked if they follow D1 commands.

## GATEA20 System Level Considerations

The A20 hookups for each CPU are as follows:

**486 CPU** — The A20 gating is done by the CPU. A20 from the CPU goes to the 4021, and to F245 which drives LA20 onto the local bus.

If the GATEA20 emulation is not done in the 4021, then the GATEA20 signal from the 8042 and the GATEA20 signal from the 4021, is ORed externaly with the 8042's GATEA20 then fed to the 486.

If the GATEA20 emulation is done (recommended) in the 4021, the external OR gate is not required.

**386 CPU** — A20 is gated in the 4021. The CPU A20 pin goes to the 4021, where it is ANDed with GATEA20 and used internally as well as sent out to the XA20 pin of the ISA bus.



GATEA20 from the 4021 is always connected directly to the 8042 through a resistor. When the emulation is off, the 8042 drives the pin and the 4021 receives it and ORs it with the internal fast GATEA20 sources.



**38605** — The board level hookup for A20 is the same as for the 386 because the 386 and 38605 generally shares the same motherboard. A20 is gated in the CPU chip. It comes into the 4021, does not get gated, is used internally and is sent out the XA20 pin to the ISA bus.

GATEA20 is connected exactly the same as the 486. The A20 board hookup also identical for both the 386 and 38605.

**38605 and 386 CPU** — GATEA20 is handled differently for the two CPUs. If the 8042 GATEA20 function is not emulated a jumper is required across the OR gate to switch between the 386 and 38605 CPU. If the emulation is used, the connections are the same and no jumper or OR gate is required.

A system may have GATEA20 connected in either of two ways:

- Non-simulated 8042: GATEA20 from the 8042 and 4021 is ORed together and sent to the CPU. In the case of a 386, the 8042 GATEA20 signal goes to the 4021 directly.
- Simulated 8042: GATEA20 from the 8042 and keyboard are connected together and sent to the CPU.



GATEA20 must be high after a hardware reset. The 8042 comes up high to account for the reset. With the simulated 8042 hookup, the 4021 must also come up high. To do this, the simulated 8042 GATEA20 powers up high.

If the 8042 will not be simulated, this bit must be taken low by writing to it through the configuration register (the simulated 8042 GATEA20 bit may be accessed directly through a configuration register). The basic algorithm is as follows:

```
If 8042 not simulated

    If 386 CPU
        disable GATEA20 output driver
    endif

    Set the Emulated 8042 GATEA20 low through configuration register before any software is
    executed which requires GATEA20 low.
else

    Enable 8042 GATEA20 emulation and blocking of 8042 commands before executing any
    software that would take GATEA20 low.
```

T-49-17-01

# The System Clocks

## The 4021 System Clocks

The 4021 CHIPSet contains the following clocks:

- CPU clocks
- AT bus clocks
- 14MHz clock
- 32KHz clock

The clocks are summarized in the following tables.

**Table 5-1.** *CPU Clocks*

| Signal | Description |
|---|---|
| CLKIN | Input from the oscillator. 2x clock in the 2x clock mode, 1x clock in the 1x clock mode. |
| CLK2 | 2x clock input is used for phase syncronization of reset in the 386 system, and for DRAM timing in a 486 non-cache mode. In a system requiring only a 1x clock, this can receive the same clock as SCLK. |
| SCLK | 1x clock input pin. |
| CWS* | Cache write strobe. Used for the CWE* signal timing. It is the 1x clock advanced by approximately 5nS. Not used in DRAM only systems. |

*Table 5-2. AT Bus Clocks*



| Signal | Description |
|---|---|
| ATCLKIN | Input clock for AT state machine. A 16, 24, or 48MHz oscillator should be used to provide an 8MHz bus. 48MHz will provide an 8MHz clock for "normal" AT and 12 MHz for FASTAT. This oscillator is optional since the AT clock source may be derived from CLKIN. |
| BUSCLK | Normal AT bus clock. Always running at the "normal" rate. |
| FBUSCLK | Fast AT bus clock. Switches between the "normal" and "fast" rates. |

*Table 5-3. 14MHz Clock*

| Signal | Description |
|---|---|
| OSC | 14.31818MHz oscillator input used for the system timer (8254). |

*Table 5-4. 32KHz Clock*

| Signal | Description |
|---|---|
| 32KHz | Real-time clock 32KHz pin. This clock is also used for the 1mS 486 reset pulse and the SuperState V timer. |

The 4025 contains the 14.31818MHz crystal circuit. This clock is not used inside the 4025. If an oscillator is used in place of a crystal, the 4025 pins need not be connected. The 4025 receives SCLK, the 1x CPU clock, which it uses for many of its internal operations.

Figures 5-1 and 5-2 show board level clock connections for the various CPUs.

*Figure 5-1. 2x Clock Connection*



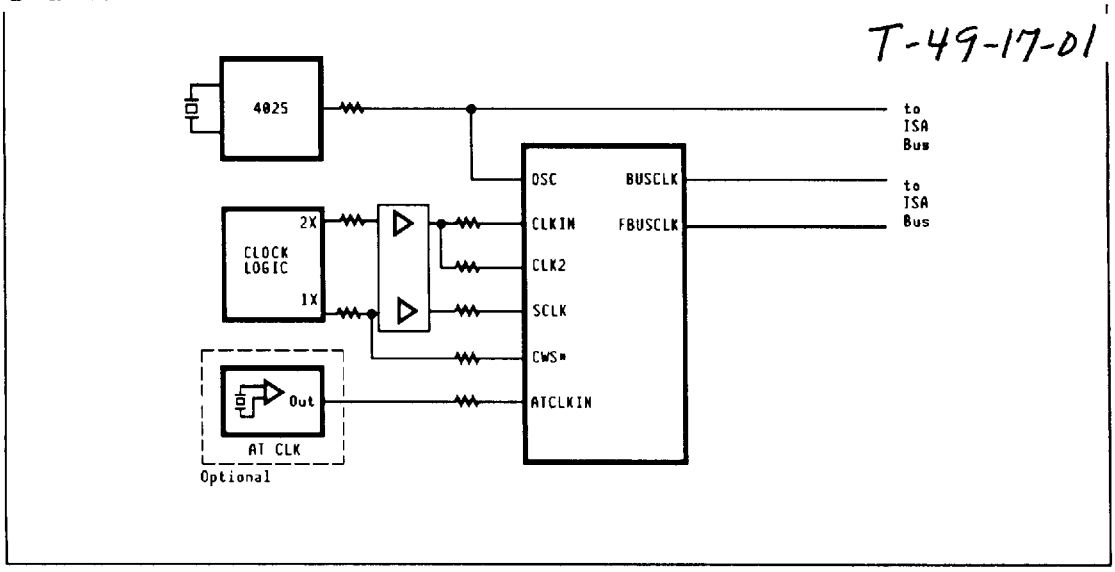
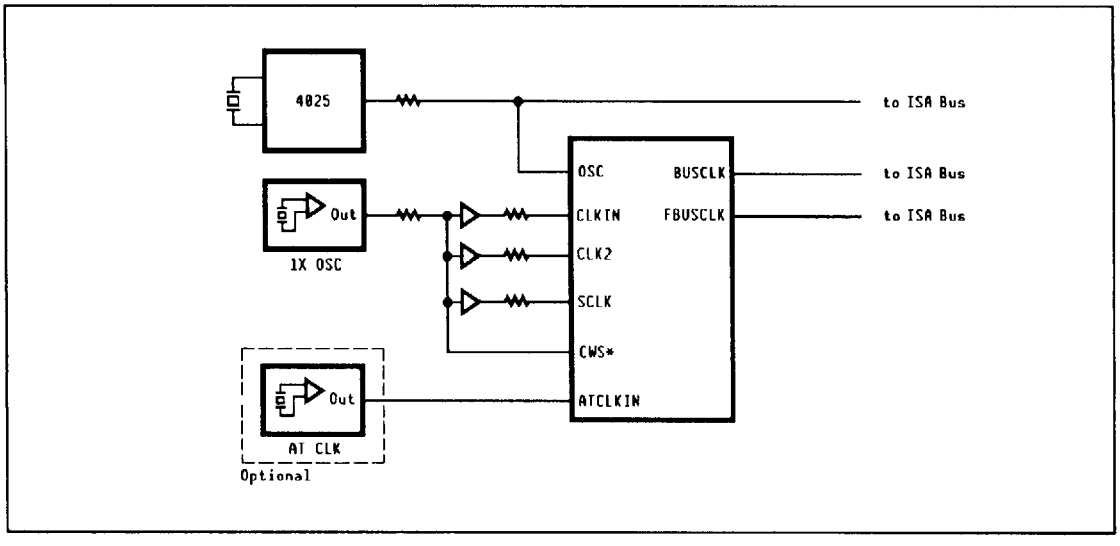



*Figure 5-2. 1x Clock Connection*



# CPU Clocks



The 4021 can operate internally with a 1x clock or both a 1x clock and a 2x clock.

The 2x clock is used in the following systems:

- 486 (16, 20, and 25MHz non-cache systems)
- 386 (all systems)
- 38605 (high-speed systems can optionally use a 1x clock)

In 2x clock mode, the 386 CPU or the 486 CPU can be used. For a 386 system, a 2x clock is provided to the CLK2 input of the 4021 and 386. A 1x clock is provided to the SCLK input of the 4021 and to the 4025. For a 486 system, a 1x clock is connected to the SCLK input of the 4021, the CPU and the 4025. The 2x clock is still connected to the CLK2 input.

The 1x clock is used for almost all the logic in the 4021. The 2x clock is used for the DRAM timings.

The 1x clock mode is used in the following systems:

- 486 (25 and 33MHz cache and non-cache systems)
- 38605 (high-speed systems)

The 1x clock mode cannot be used on 386 CPUs.

In 1x clock mode, the 1x clock is fed into the SCLK input of the chip from an external clock driver. A 1x or 2x clock should be connected to the CLKIN and CLK2 inputs. CLKIN can be devided to produce various AT BUSCLK frequencies. CLK2 is used in generating RESET (SYSRESET/CPURESET).

Most of the internals of the 4021 use only the 1x CPU clock. The 4021 internals that use the 2x clock are:

- The DRAM controller CAS state machine in the "3-1-1-1 and 3-2-2-2" modes only.
- Some AT bus clock sources.

The RAS state machine and other DRAM control logic use the 1x clock in all modes. The AT bus clocks can be sourced from the CLKIN pin (with 2x or 1x) or from the ATCLK input pin.

The clock mode is sampled by the NMI pin at power up. It is tied to a pullup or pulldown, and is sampled shortly after PWRGOOD goes high. The result is latched internally and called SEL1X.

Software can read the state of SEL1X through a configuration bit. The BIOS uses this information to indicate which DRAM modes can be used and which clock source to use for the AT bus clocks.

The 486 cannot have its clock switched while it is operating. Slowing down the 486 to simulate an 8MHz AT (the DeTurbo function) is done by using the HOLD and FLUSH* pins.



## AT Bus Clocks

T-49-17-01

There are two AT bus clock output pins: one for the normal AT bus slots (BUSCLK), and one for the fast AT bus slots (FBUSCLK). When FASTAT is not implemented on the system board, the FBUSCLK pin is not used.

The BIOS sets two frequencies for the AT clock. One for normal AT speed cycles, and one for fast AT cycles. The frequency for BUSCLK remains fixed at the normal AT speed. FBUSCLK switches between the normal AT speed and the fast AT speed reflect the timing of the current ISA bus cycle. The switch occurs before TS at the beginning of an AT cycle.

The AT clocks have two sources with several dividers for each:

- CLKIN: 1x or 2x CPU clock directly from the oscillator.
- ATCLKIN: Special oscillator for the ISA bus. Should be used when an exact frequency is required.

The default source is CLK2 because ATCLKIN may not have an oscillator connected. Using CLK2 avoids the need for a separate oscillator, but does not always provide the ideal frequencies.

The following tables list the BUSCLK or FBUSCLK rate for each clock selection when using CLK2 with a 2x clock, CLKIN with a 1x clock, and ATCLK as the source. The "selection" column shows the divider which is used to obtain BCLK, which is 2x BUSCLK. The frequencies in the tables are the BUSCLK and FBUSCLK rates. The 1.5 and 2.5 divisions use the negative and positive edge of the clock.

Table 5-5 shows the ISA clock frequencies for CLKIN. The table values are provided for various CPU operating speeds. In 2x mode, CLKIN is twice the rate shown.

**Table 5-5.** *ISA Clock Frequencies for CLKIN with a 2x Clock Source*



| Selection | 16 MHz | 20 MHz | 25 MHz | 33 MHz | 40 MHz | 50 MHz |
|---|---|---|---|---|---|---|
| CLKIN | 16 | 20 | 25 | 33 | 40 | 50 |
| CLKIN/1.5 | 10.7† | 13.3† | 16.6 | 22.2 | 26.7 | 33.3 |
| CLKIN/2 | 8‡ | 10 | 12.5 | 16.3 | 20 | 25 |
| CLKIN/2.5 | 6.4 | 8‡ | 10 | 13.3 | 16 | 20 |
| CLKIN/3 | 5.3 | 6.7 | 8.33 | 11.1 | 13.3† | 16.7 |
| CLKIN/4 | 4 | 5 | 6.25 | 8.3 | 10 | 12.5† |
| CLKIN/5 | 3.2 | 4 | 5 | 6.67 | 8 | 10 |
| CLKIN/6 | 2.67 | 3.3 | 4.17 | 5.56 | 6.67 | 8.33‡ |

‡ Recommended for BUSCLK
† Recommended for FBUSCLK

Table 5-6 shows ISA clock frequencies with CLKIN with a 1x clock source. The CPU operating speed is CLKIN speed in 1x mode.

**Table 5-6.** *ISA Clock Frequencies for CLKIN with a 1x Clock Source*

| Selection | 16 MHz* | 20 MHz* | 25 MHz | 33 MHz | 40 MHz | 50 MHz |
|---|---|---|---|---|---|---|
| CLKIN/1 | 8† | 10‡ | 12.5‡ | 16.7 | 20 | 25 |
| CLKIN/1.5 | 5.3 | 6.7‡ | 8.3 | 11.1‡ | 13.3‡ | 16.7 |
| CLKIN/2 | 4 | 5 | 6.25 | 8.3 | 10 | 12.5 |
| CLKIN/2.5 | 3.2 | 4 | 5 | 6.7 | 8 | 10 |
| CLKIN/3 | 2.67 | 3.3 | 4.17 | 5.5 | 6.7 | 8.3 |
| CLKIN/4 | 2 | 2.5 | 3.1 | 4.13 | 5 | 6.25 |
| CLKIN/5 | 1.6 | 2 | 2.5 | 3.33 | 4 | 5 |
| CLKIN/6 | 1.33 | 1.67 | 2.1 | 2.75 | 3.33 | 4.2 |

* At 16 and 20 MHz, a 2x clock is normally used.
‡ Recommended for BUSCLK
† Recommended for FBUSCLK

Table 5-7 shows clock frequencies for CLKIN with an external ATCLK source.



*Table 5-7. ISA Clock Frequencies for CLKIN with External ATCLK Source*

T-49-17-01

| | ATCLK Frequency | | |
|---|---|---|---|
| **Selection** | **16 MHz** | **24 MHz** | **48 MHz** |
| ATCLK/1 | 8 | 12 | 24 |
| ATCLK/2 | 4 | 6 | 12 |
| ATCLK/3 | 2.7 | 4 | 8 |

Tables 5-8 and 5-9 list the programming recommendations for each CPU frequency. Many factors are involved in deciding which clock speed to use in a system; for example, whether 8.33 MHz is acceptable for the standard clock, what speed of fast clock is acceptable, the cost of an oscillator, etc. The recommendations are based on the following assumptions:

- The ATCLK oscillator should be avoided, if possible.
- The ideal standard clock is 8 MHz with 8.33 MHz acceptable.
- The ideal fast clock is 12.5 MHz (11.1 MHz is too slow and 13.3 MHz is too fast to be included in the "Recommended" column, but these values are acceptable as the secondary choice).
- ATCLK is 16 MHz when FASTAT is not implemented (Table 5-8).
- ATCLK is 48 MHz when FASTAT is implemented (Table 5-9).

T-49-17-01

*Table 5-8. ISA Clock Programming Recommendations (FASTAT Not Implemented)*



| CPU Frequency | CLKIN Frequency | Recommended Standard Clock | Secondary Choice Standard Clock |
|---|---|---|---|
| 16 MHz 2x | 32 MHz | CLKIN/2 (8) | ATCLK/1 (8) |
| 20 MHz 2x | 40 MHz | CLKIN/2.5 (8) | ATCLK/1 (8) |
| 25 MHz 2x | 50 MHz | CLKIN/3 (8.33) | ATCLK/1 (8) |
| 33 MHz 2x | 66 MHz | CLKIN/4 (8.33) | ATCLK/1 (8) |
| 40 MHz 2x | 80 MHz | CLKIN/5 (8) | ATCLK/1 (8) |
| 25 MHz 1x | 25 MHz | CLKIN/1.5 (8.3) | ATCLK/1 (8) |
| 33 MHz 1x | 33 MHz | CLKIN/2 (8.3) | ATCLK/1 (8) |
| 40 MHz 1x | 40 MHz | CLKIN/2.5 (8) | ATCLK/1 (8) |

Note that the ATCLK oscillator is never required for the standard clock. A 48 MHz ATCLK oscillator may be used with a divide by 3 also.

*Table 5-9. ISA Clock Programming Recommendations (FASTAT Implemented)*

| CPU Frequency | CLKIN Frequency | Recommended Standard Clock | Recommended Fast Clock | Secondary Choice Standard Clock | Secondary Choice Fast Clock |
|---|---|---|---|---|---|
| 16 MHz 2x | 32 MHz | CLKIN/2 (8) | ATCLK/2 (12) | CLKIN/2 (8) | CLKIN/1.5 (10.7) |
| 20 MHz 2x | 40 MHz | ATCLK/3 (8) | ATCLK/2 (12) | CLKIN/2.5 (8) | CLKIN/1.5 (13.3) |
| 25 MHz 2x | 50 MHz | CLKIN/3 (8.33) | CLK2/2 (12.5) | ATCLK/3 (8) | ATCLK/2 (12) |
| 33 MHz 2x | 66 MHz | ATCLK/3 (8) | ATCLK/2 (12) | CLKIN/4 (8.33) | CLKIN/2.5 (13.3) |
| 40 MHz 2x | 80 MHz | ATCLK/3 (8) | ATCLK/2 (12) | CLKIN/5 (8) | CLKIN/3 (13.3) |
| 25 MHz 1x | 25 MHz | CLKIN/1.5 (8.3) | CLKIN/1 (12.5) | ATCLK/3 (8) | ATCLK/2 (12) |
| 33 MHz 1x | 33 MHz | ATCLK/3 (8) | ATCLK/2 (12) | CLKIN/2 (8.3) | CLKIN/1.5 (11.1) |
| 40 MHz 1x | 40 MHz | ATCLK/3 (8) | ATCLK/2 (12) | CLKIN/2.5 (8) | CLKIN/1.5 (13.3) |

Note that if ATCLK is recommended for the FAST clock, it is also shown for the standard clock. If the ATCLK oscillator is on the board, it can be used for both frequencies.

## 14.31818 MHz Clocks

The 4025 has an oscillator circuit for the 14.31818 MHz. It does not use the clock internally. It allows a crystal to be used instead of an oscillator. The 4025 output can directly drive the ISA bus. If an oscillator is used, the 4025 circuit is not required. By connecting the oscillator to X1, the 4025 circuit can be used as a buffer to drive the AT bus, eliminating the need for a separate buffer.

The 4021 receives the 14.31818 MHz "OSC" signal, divides it by 12 internally, and sends it to the 8254 timer logic which is integrated in the 4021. The 14.31818 MHz is not used elsewhere within the chip.

T-49-17-01

## 32KHz Clock

The 32KHz clock (actually 32.768 KHz) is used for the real-time clock and is always running. When the power is OFF, it draws power from the battery.

The 32KHz clock is also used for the 486 1mS reset timer, which is used on power up reset and for the SuperState V timer.

## Clock Speed Check

The clock speed check is a new feature to allow the BIOS to check system speeds. Each of the system clock inputs may be connected to a divider chain whose output may be read directly from software. The divider chain is an 11-bit ripple counter. The output toggles with every 1024 cycle of the clock. When the 32KHz clock is selected, only one flip-flop is used in the divider, so the period is not excessive. Configuration register 0D bits 2:0 select which clock is used as the source. The choices are as follows:

- CLKIN
- SCLK
- ATCLKIN
- OSC (14.31818 MHz)
- 32KHz

The 14.31313MHz or 32KHz clock can be used for determining the speed of the others. The output of the divider chain can be read through configuration register 0D, bit 3 or Port 92, bit 7. When this feature is disabled (000 on the select bits), Port 92, bit 7 contains 0. Table 5-10 lists how often the bit toggles for each clock frequency.

*Table 5-10. Clock Speed Check Timing Period*

T-49-17-01



| Clock | Toggle Period (uS) |
|---|---|
| 14.31818 MHz | 71.52 |
| 16 MHz | 64.0 |
| 20 MHz | 51.2 |
| 24 MHz | 42.67 |
| 25 MHz | 40.96 |
| 32 MHz | 32.0 |
| 33 MHz | 30.72 |
| 40 MHz | 25.6 |
| 48 MHz | 21.33 |
| 50 MHz | 20.48 |
| 66.6 MHz | 15.36 |
| 80 MHz | 12.8 |
| 100 MHz | 10.24 |
| 32KHz (/2) | 30.52 |

CHAPTER 6

T-49-17-01

# The DRAM Controller

The 4021 CHIPSet employs page mode, page interleaving, and Multi-RAS active techniques. These techniques may be used independently or together.

## Page Mode

Page mode is always used in this CHIPSet for CPU accesses for bursts and between bursts. Page mode simply means keeping RAS low while reading or writing multiple words within a DRAM page by providing only a new column address and toggling CAS.

## Page Interleaving

Page interleaving involves interleaving DRAM banks on a DRAM page boundary; this effectively doubles the page size of the DRAM banks. To achieve the best performance from this technique, the page interleaving is done such that the two pages are always consecutive pages rather than being discontinous. Page interleaving gains the most advantage when used with multi-RAS active, but also has a performance gain with single RAS active since many cycles which would be page miss cycles are now RAS inactive cycles.

## Multi-RAS Active

Multi-RAS active allows multiple banks of DRAM to have pages open simultaneously, allowing a page hit in any open page. When combined with page interleaving this allows faster cycles when the CPU is bouncing between several areas (code, data, and stack, for instance), or when looping across a page boundary. Where possible in the 4021, multi-RAS active is automatic. Two RASes may be ctive only if the two banks of DRAMs have separate CAS lines. When a page must be opened in a bank of DRAMs, the RAS of a currently active bank will only go high if the new bank shares CAS lines with it. The 8 RAS mode has only one set of CAS lines, so multi-RAS is not possible.

# Blocks and Banks



A *bank* is a 36 bit wide DRAM array with a common RAS. There are either 4 or 8 supported depending on the mode.

T-49-17-01

A *block* is a programmable decode. The CHIPSet has 4 blocks. Each block supports 1 bank in 4 RAS mode, and 2 banks in 8 RAS mode. The block contains the starting address and depth of the DRAMs in its banks.

The DRAM controller is optimized for a non-cache 486 system and a cache based 386 and 486 system. The timing constraints to achieve optimum performance for each of these systems are different at different system speeds. To use the optimized timing for a non-cache based system, the input clock frequency must be twice the system speed. All timing variations required for a non-cache or cache based system are covered with five different timing modes that vary the wait states. In some timing modes, the RAS to CAS timing may also be varied.

The first two modes are optimized for a non-cache based 486sx system, providing a very fast burst read cycle. The system speed is limited to 25 MHz and below (due to DRAM limitations). These modes use the 2x clock input to the CHIPSet.

The other three modes are optimized for cache based, higher speed systems (25- 40 MHz) and use the 1x clock input to the 4021. They may also be used without the cache in 486 systems, providing a very good performance to cost ratio.

Table 6-1 summarizes the five DRAM timing modes. The "n-n-n-n" numbers indicate the number of cycles for each word of the 486 burst. The fastest possible burst is 2-1-1-1 which means the first word is received in two clocks (end of the first T2), and the next three words are received after one clock each. For a cache based system, it would mean the bursting to the cache and CPU for read miss cycles. For the 386 and Super386 bursting is used only for cache line fills.

**Table 6-1.** *DRAM Timing Modes*

| Mode | CPU Speeds | Use with cache? | Clock Mode | Burst Page Hit | Burst RAS Inactive | Burst Page Miss | Back-to- Back Writes |
|---|---|---|---|---|---|---|---|
| 0 | 16, 20 | No | 2x | 3-1-1-1 | 5-1-1-1 | 7-1-1-1 | 0ws |
| 1 | 25 | No | 2x | 3-2-2-2 | 5-2-2-2 | 7-2-2-2 | 0ws |
| 2 | 25 cache | Yes | 1x or 2x | 4-2-2-2 | 6-2-2-2 | 9-2-2-2 | 0ws |
| 3 | 33 non-cache | No | 1x or 2x | 4-2-2-2 | 6-2-2-2 | 9-2-2-2 | 0ws |
| 4 | 33, 40 cache | Yes | 1x or 2x | 5-3-3-3 | 7-3-3-3 | 11-3-3-3 | 0ws |

# The DRAM Non-Cache Modes

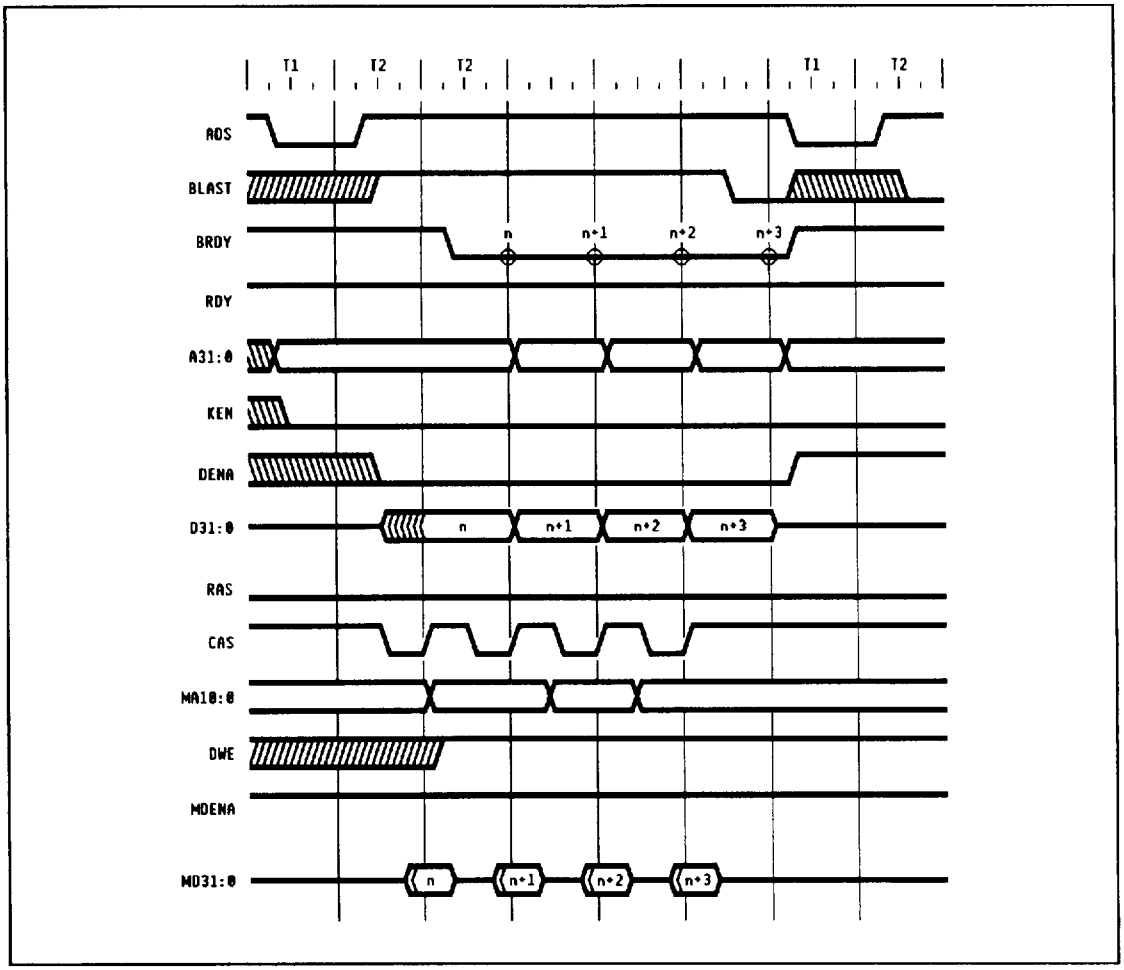
CHIPS & TECHNOLOGIES INC

The DRAM non-cache modes are specifically designed for systems with a 486sx CPU at 16, 20, and 25 MHz rates. The DRAMs are cycled at a rate high enough to provide burst data on every clock cycle. This method utilizes the full potential of a page mode DRAM.

T-49-17-01

In mode 0, the 2x clock is used to cycle CAS in a CPU burst read (see Figures 6-1 to 6-4). At the beginning of a T2 state, the CASx is active for a complete 2x clock period. Data from the DRAM is latched by the 4025 CHIPSet at the end of each CAS. The system CPU reads it from the 4025 at the end of every T2 state; a burst rate of 3-1-1-1 is achieved in a 486 system.

***Figure 6-1.*** *DRAM - Page Hit Read Burst*



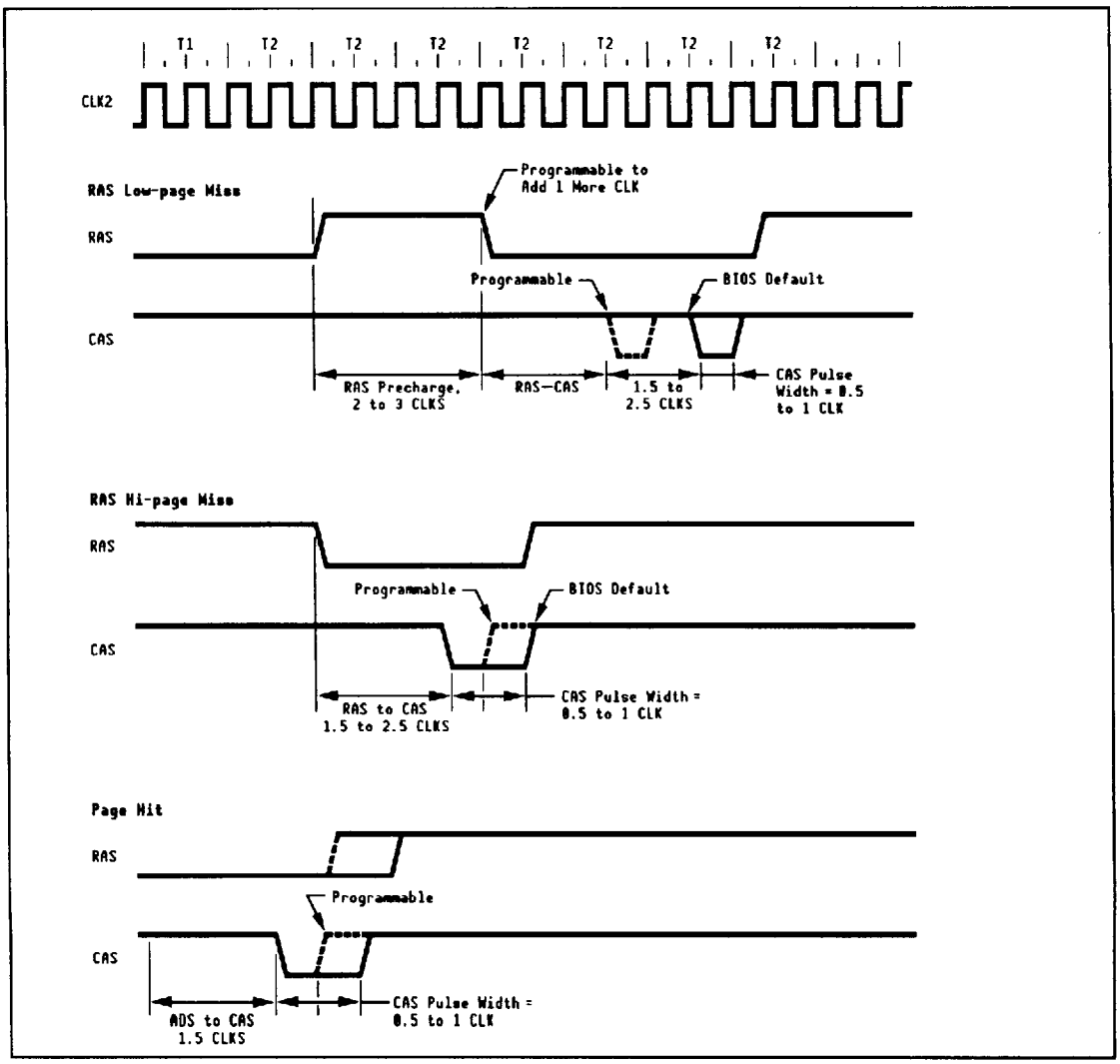
Since the CAS is cycled on every clock in a burst mode read cycle, the CAS precharge and the CAS access time of the DRAM are critical timing parameters to be considered in a system design.

T-49-17-01

In mode 1, the 2x clock is used but the CAS is cycled on every T-state boundary. Data from the DRAM is latched by the 4025 CHIPSet at the end of each CAS; the system CPU reads it at the end of the next T2 state and a burst rate of 3-2-2-2 is achieved in a 486 system. The only difference between modes 0 and 1 is the CAS low and high times on reads. Mode 1 doubles it to one T-state each. This eases up on the page mode cycle time and CAS low time parameters.

A separate configuration bit allows the RAS to CAS timing to be extended from the normal 1.5 T-states to 2.5 T-states for read cycles. This timing modifier bit can be used for DRAMs with slower RAS access time.

***Figure 6-2.*** *DRAM - Programmable Options*



The 4025 latches the data from the DRAM and is used to hold the data to the CPU while the DRAM controller begins the read to the next word during a burst. The latching of the DRAM data is with byte resolution and is controlled by the CAS7:0* signal. A closer look at this latch describes the extent of DRAM utilization. The DRAM data latch in the 4025 goes transparent when CASx* is low and SCLK is high. This latch closes only on the rising edge of the CASx* signal.

Writes are identical in both modes. The data is posted in the 4025 write buffer, allowing the first write (with the write buffer being empty) to complete in zero wait states. Back to back writes are an important consideration in the 486 since its internal cache is write through, making write page hits a higher probability. This write buffering scheme allows back to back writes to finish in zero wait states as long as page hits are maintained. For system CPU writes to DRAM, the 4025 CHIPSet latches the local data bus at the end of the first T2 and the system CPU write cycle is completed in zero waits. The actual writing of the data to the DRAMs does not occur until the next cycle when CAS goes active at the end of T1 (of the next CPU cycle). CAS remains active until the end of T2 to allow the local data bus to propogate through the 4025 CHIPSet and become stable on the DRAM data bus.

Because CAS is delayed for writes, an extra cycle must be inserted for a read which immediately follows a write. There are no cycles added for write cycles which follow a read.

*Figure 6-3. Posted Writes*

T-49-17-01



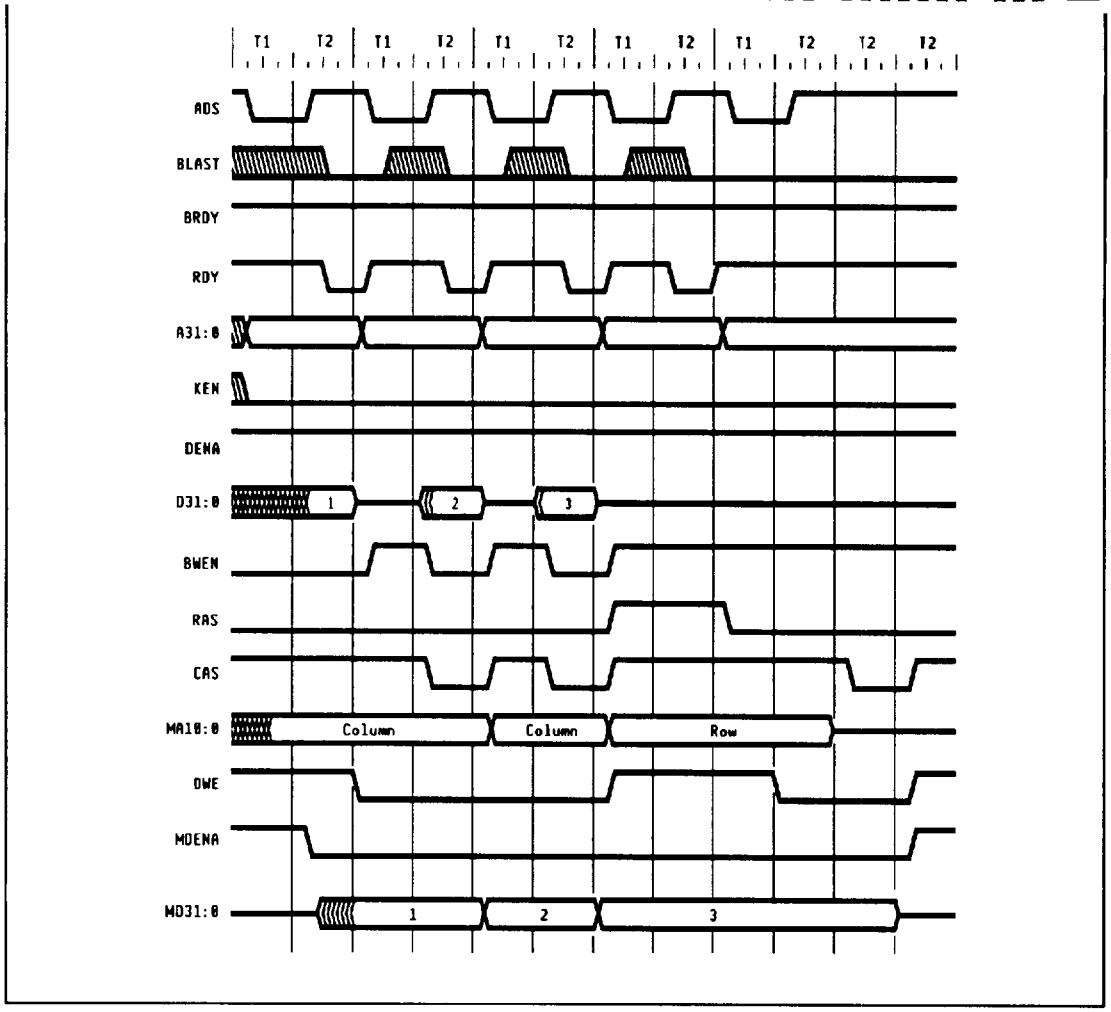


*Figure 6-4. DRAM Write Miss—Read Hit Burst*

T-49-17-01



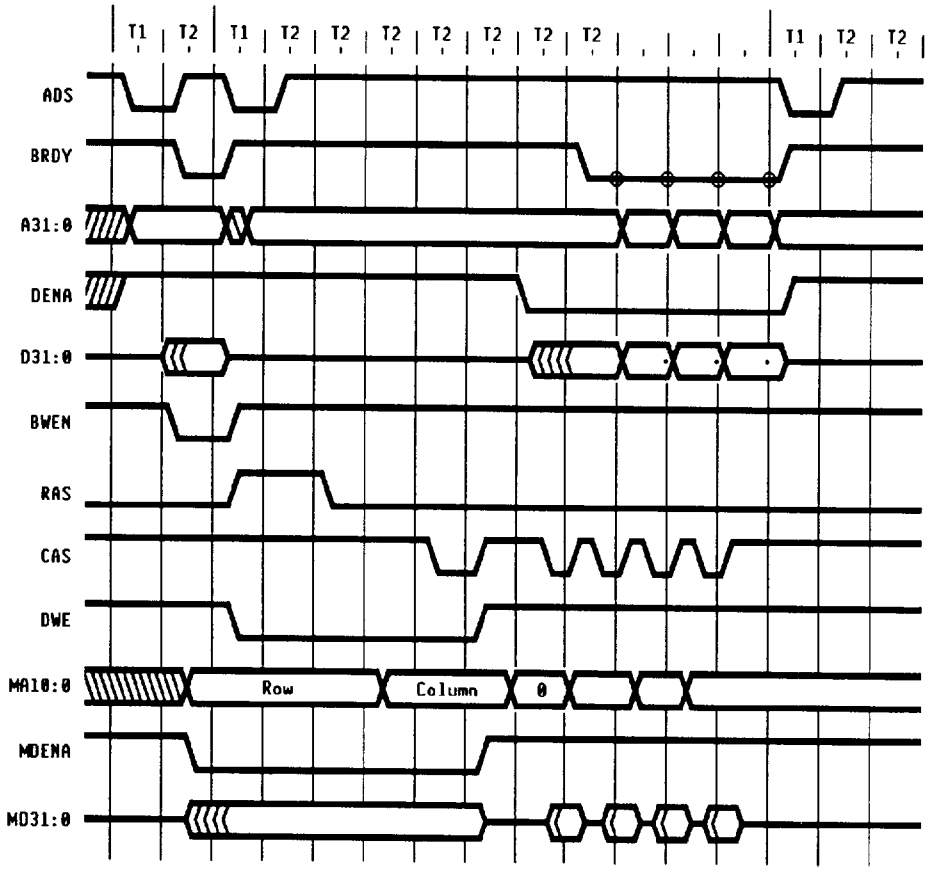


Table 6-2 shows the DRAM types required for the different CPU speeds (16 MHz, 20 MHz, etc.) in mode 0 and mode 1. The CAS access time, tCAC, is the critical DRAM parameter. It is shown and must be met for any DRAM speed.

*Table 6-2. Non-Cache Modes and DRAM Timing Parameters* T-49-17-01



| DRAM Type | 16 MHz | 20 MHz | 25 MHz |
|---|---|---|---|
| | DRAM tCAC | | |
| | 25ns | 20ns | 15ns |
| -80s | mode 0, mode 1 | mode 0, mode 1 | mode 1 |
| -70s | mode 0, mode 1 | mode 0, mode 1 | mode 0, mode 1 |
| -60s | mode 0, mode 1 | mode 0, mode 1 | mode 0, mode 1 |

## DRAM Cache Modes

The DRAM cache modes are designed for cache-based systems and for system speeds of 25 MHz and up. They also perform well in a DRAM-only system with the 486.

For a 386 CPU, the system is configured in the 2x mode. The CHIPSet uses only the 1x clock for all internal timings.

For a 486 CPU, the system can be configured in the 2x mode or the 1x mode, but only the 1x clock is used for all internal timings.

The DRAM read cycle in the cache mode uses CAS that is active for multiples of T-state boundaries rather than half T-state boundaries. This allows more address decode and MA bus setup time at the start of the cycle making the fastest burst a 4-2-2-2.

In mode 2, the 1x clock is used to cycle CAS on every 1x clock (see Figure 6-5) . Data from the DRAM is latched by the chipset (4025) at the end of each CAS and the system CPU reads it on at the end of every T2 state Since the CAS is cycled on every 1x clock in a burst mode read cycle, the CAS precharge and the CAS access time of the DRAM are less critical in this mode.

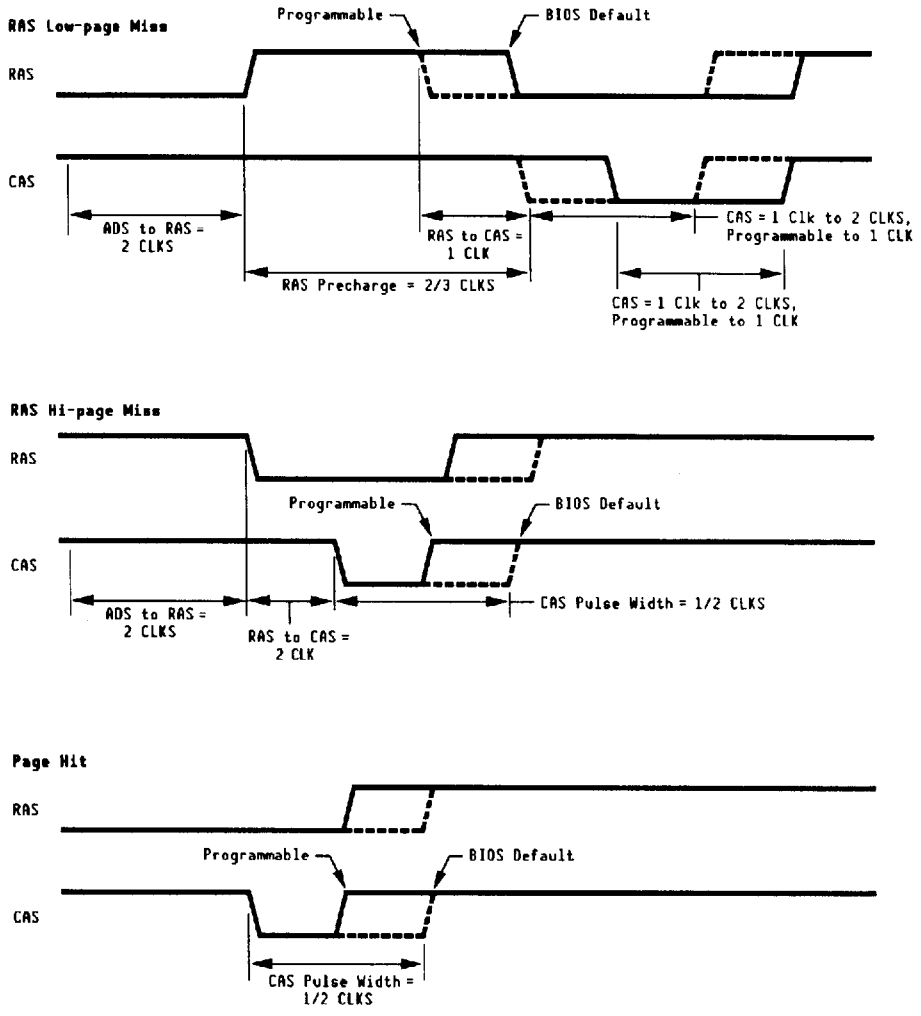
In mode 3, the 1x clock is used but the CAS is cycled active for two 1x clk periods (see Figure 6-5). In this mode, data from the DRAM is latched by the 4025 CHIPSet at the end of each CAS. The system CPU reads it at the end of the next T2 state. The only difference between mode 2 and mode 3 is the CAS low time on reads. Mode 3 doubles it to two T-states each. This eases up on the page mode cycle time and CAS access time parameters.

In these two modes the RAS to CAS timing is always fixed to one T-state for read cycles, unlike mode 1 and mode 0.

T-49-17-01

**Figure 6-5.** *DRAM Cache Modes — RAS/Cache Relation*

CHIPS & TECHNOLOGIES INC



The 4025 latches the data from the DRAM and is used to hold the data to the CPU while the DRAM controller reads the next word during a burst. The latching of the DRAM data is with byte resolution and is controlled by the CAS7:0* signal. The latch is transparent when CAS* lines for a particular byte is low and the SCLK is high. The latch closes on the rising edge of CAS*. The path from CAS*, until the latch closes, should be greater than the CAS* access time of the DRAM (tCAC) as shown in Table 6-3.

CHIPS & TECHNOLOGIES INC

***Table 6-3.*** *Non-Cache Modes and DRAM Timing Parameters*

T-49-17-01

| DRAM Type | 20MHz | 25 MHz | 33 MHz | 40 MHz |
|---|---|---|---|---|
| | DRAM tCAC | | | |
| | 25ns | 25ns | 20ns | 20ns |
| -80 | mode2 | mode2 | mode3 | mode3 |
| -70 | mode2 | mode2 | mode3 | mode3 |
| -60 | mode2 | mode2 | mode2 | mode3 |
| -50 | mode2 | mode2 | mode2 | mode3 |

Mode 2 is the basic cache timing mode, with modes 3 and 4 simply adding wait states for higher system clock speeds. Mode 4 is specifically for 50 MHz and is not implemented.

Write cycles are similar to modes 0 and 1. The data is held in the write buffer and CAS goes low one T-state after T2 on a page hit. In this way, zero wait state writes are maintained within page hits on back to back writes.

## Hidden Refresh Cycles

When a refresh request occurs, the DRAM controller tries to transition the refresh into idle cycles. Writes are posted by the DRAM controller. The actual DRAM write cycle extends into the next T1 or Ti state. The DRAM controller already knows whether the write is followed by a least one idle cycle before it is able to begin the refresh. Writes are often followed by more writes or burst reads (reads often have to wait for the 486 internal write buffer to clear out before going to the bus). It is useful to know whether one or more idle cycles follow a write. The refresh begins only when the write cycle to the DRAM is finished.

On read cycles, the DRAM controller actually finishes one T-state before the end of the burst. The refresh cycle is initiated before the read to the local bus is finished.

Read cycles are more likely to be followed by idle cycles since the CPU is often waiting for the read.



The hidden refresh algorithm operates as follows:

- If an idle cycle is occurring when the refresh request comes in, do the refresh.
- If a read cycle is occurring when the refresh request comes in, put the refresh on the tail end of the read.
- If a write is occurring when the refresh request comes in, at the end of the write, do the refresh.



CAS before RAS refresh is used. The RAS refreshes are staggered to reduce power supply spikes. Since the DRAM controller can have only two active RASes, at least two RASes must therefore be inactive. The inactive RAS banks will be refreshed first because they already have their RAS precharge time met.

## DRAM Bank Configurations

The DRAM decoding section is designed to be extremely versatile. 9-bit SIMMs or 36-bit single or double bank type of SIMMs may be used. This leads to the two basic DRAM bank configurations: the 4RAS mode, and the 8RAS mode.

### 4 RAS Mode

The 4 RAS mode has the following features:

- 4 banks of DRAMs supported (in addition to the VRAM bank).
- Blocks and banks are equivalent in this mode.
- Multi RAS active Page interleaving (2 way).
- Banks 0 and 1; 2 and 3; 1 and 2; or 0 and 3 may be interleaved.
- Each block (and bank) may contain a different DRAM depth (a pair must be the same for interleaving).

*Table 6-4. 4 RAS Mode RAS and CAS Usage*

T-49-17-01



| Block Number | RAS Used | CAS Used |
|---|---|---|
| Block 0 | RAS0* | CAS3:0* |
| Block 1 | RAS1* | CAS7:4* |
| Block 2 | RAS2* | CAS3:0* |
| Block 3 | RAS3* | CAS7:4* |
| VRAM | VRAS* | CAS3:0* |

## 8 RAS Mode

The 8 RAS mode has the following features:

- 8 banks of DRAMs supported (in addition to the VRAM bank).
- Designed for 4 double bank SIMMs.
- Blocks contain 2 banks in this mode.
- Single RAS active page interleaving (2 way).
- A double bank SIMM is automatically page interleaved.
- Two single bank SIMMs may be interleaved as in 4 RAS mode.
- Each block may contain a different DRAM depth.
- Each block may contain a single or double bank.

When double bank SIMMs are used, each block represents a double bank SIMM. Single and double bank SIMMs can be mixed with full addressability. If a block contains a single bank, it must use the lower numbered RAS* pin, which will occur automatically with double bank SIMMs.

Table 6-5 shows the RAS and CAS usage in the 8 RAS mode.



*Table 6-5. 8 RAS Mode RAS and CAS Usage*

T-49-17-01

| Block Number | RAS Used (one bank installed) | RASes Used (two banks installed) | CASes Used |
|---|---|---|---|
| Block 0 | RAS0* | RAS0* and RAS4* | CAS3:0* |
| Block 1 | RAS1* | RAS1* and RAS5* | CAS3:0* |
| Block 2 | RAS2* | RAS2* and RAS6* | CAS3:0* |
| Block 3 | RAS3* | RAS3* and RAS7* | CAS3:0* |
| VRAM | VRAS* | VRAS* | CAS3:0* |

Three common board level configurations are as follows:

- 4 banks of SIMMs, page interleaved if a pair of banks is the same size.
  4 RAS, 8 CAS used. Allows Multi-RAS active page interleaving.
- 2 banks of SIMMs, page interleaved if same size, page mode, if different size.
  2 RAS, 8 CAS used. Allows Multi-RAS active page interleaving.
- 8 banks. Page interleaving of same size banks.
  8 RAS, 4 CAS used. Page mode. A double bank is page interleaved. Allows 4 double bank SIMMs (or 8 separate banks).

Figures 6-6, 6-7, and 6-8 illustrate 4 and 8 RAS modes.

*Figure 6-6. 4 RAS Mode with Two Banks*

T-49-17-01

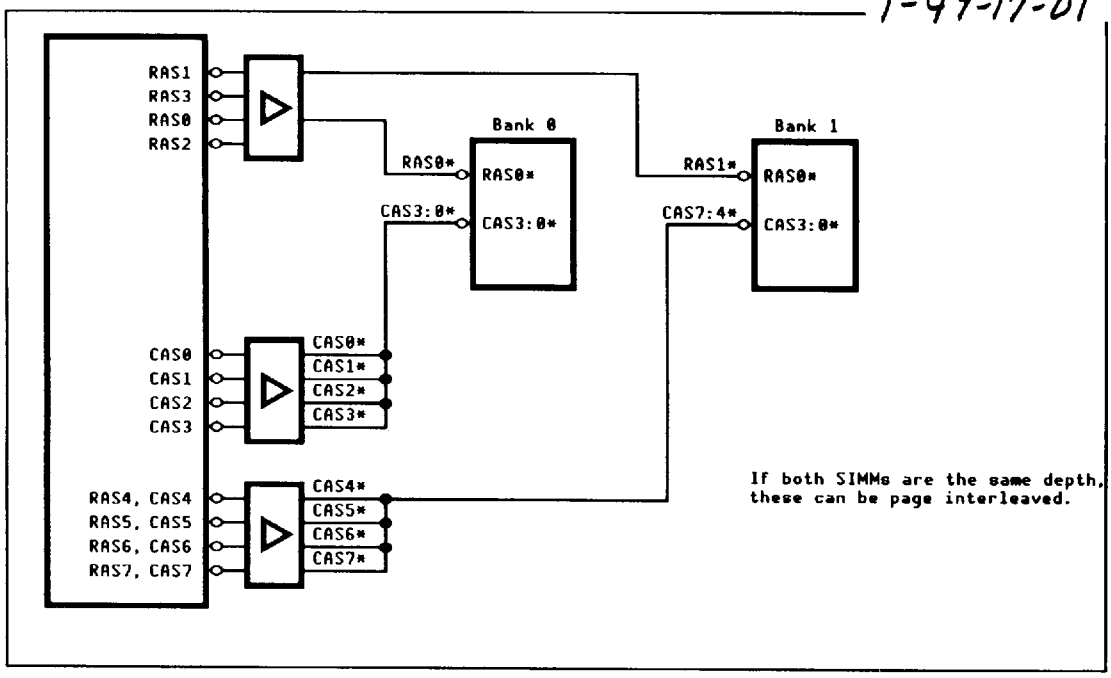



*Figure 6-7. 4 RAS Mode with Four Banks*

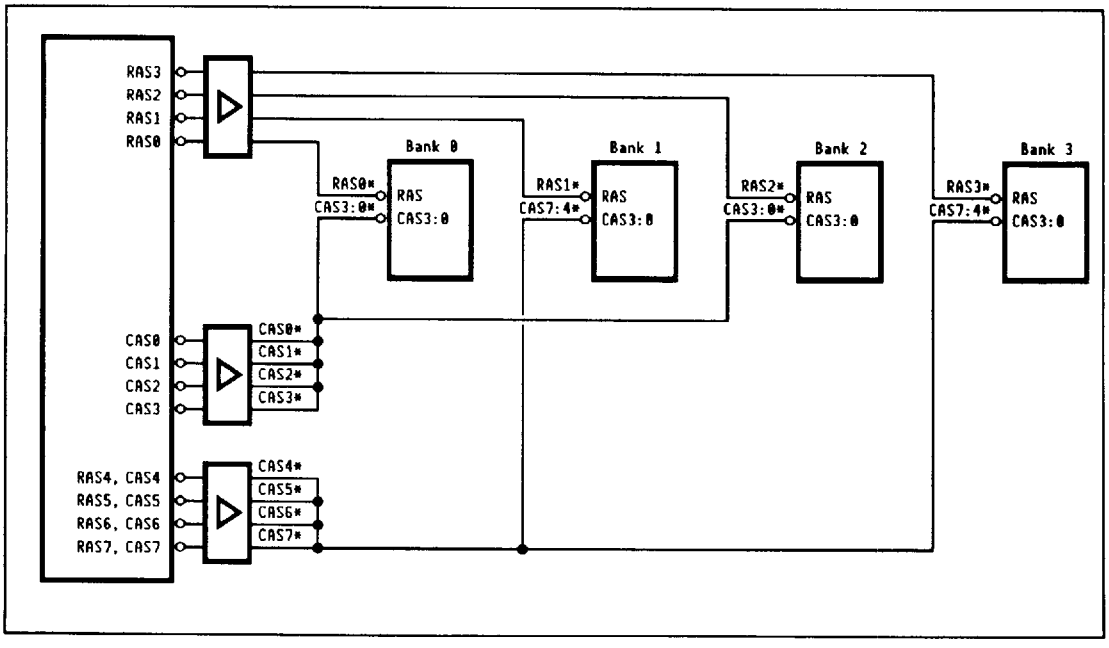
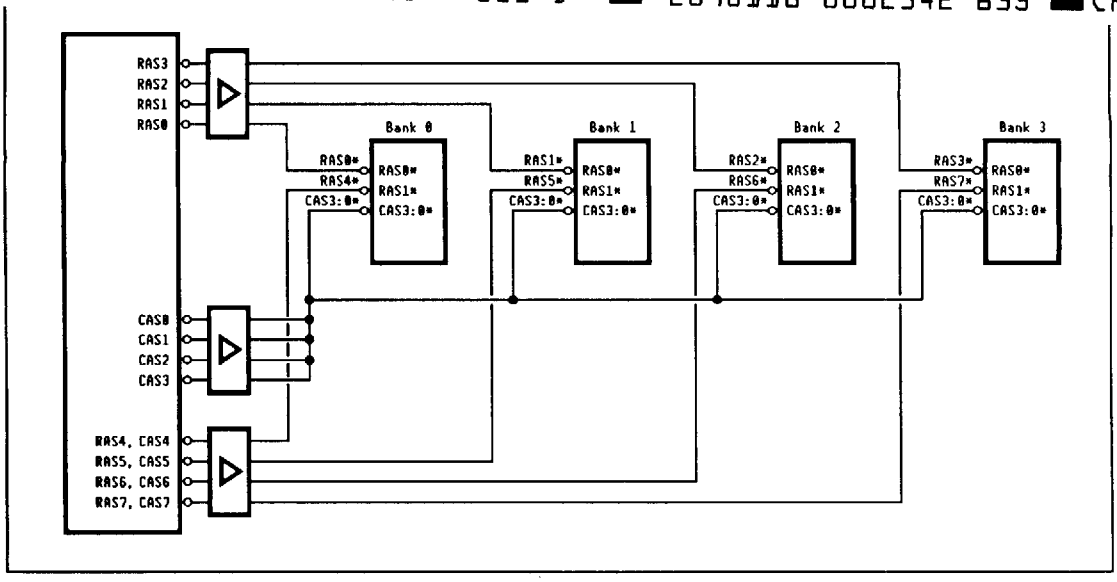


*Figure 6-8. 8 RAS Mode with Eight Banks*

T-49-17-01






## DRAM Block Programming

Each block has the following programming:

- DRAM depth: 256K, 1MB, or 4MB.
- Single or Double bank (8 RAS mode only).
- Interleave or non-interleave.
- Starting Address: A20-26. Must start on a block size boundary (a 256K deep single bank must start on a 1MB boundary, a 256K deep double bank must start on a 2MB boundary).

Because the blocks must be on a block size boundary, the largest block must be placed at the bottom of memory to avoid a hole in the memory map. Table 6-6 lists the addressing restrictions.

*Table 6-6. DRAM Block Starting Address*

T-49-17-01



| DRAM SIMM Depth | Banks per Block | Size of Block | Address Bits Programmed | Placement Boundary Size |
|---|---|---|---|---|
| 256K | Single | 1 MB | A20-26 | 1 MB |
| 256K | Double | 2 MB | A21-26 | 2 MB |
| 1 MB | Single | 4 MB | A22-26 | 4 MB |
| 1 MB | Double | 8 MB | A23-26 | 8 MB |
| 4 MB | Single | 16 MB | A24-26 | 16 MB |
| 4 MB | Double | 32MB | A25-26 | 32 MB |

## DRAM Address Muxing

Table 6-7 shows the DRAM address (MA) muxing. The column address is the same for all configurations, since this is the speed path. However, A2 and A3 must go through an internal burst counter. Table 6-6 shows the ROW address choices and which addresses are used for each DRAM configuration.

A11 is a special case for 4Mx4 DRAMs, some of which will have 12 row addresses and 10 column addresses. A23, which is provided on column address 10, is repeated on row address 11. This allows 11/11 and 12/10 addressing chips to be used transparently, and even be mixed.

MA9 contains A20 for 256K DRAMs. Therefore, the VRAM bank can use 512K deep VRAMs or externally decode two banks of 256K VRAMs from the single VRAS*.

MA10 is DSF during VRAM cycles. It is low when VRAS* goes low. This includes CPU accesses to the VRAM and refresh cycles. For VRAM transfer cycles, it is controlled by the split transfer mode bit in combination with some other logic.

MA11 is TROE* during VRAM cycles. For CPU accesses, it is high when VRAS* falls, low when CAS* falls, and remain low for the duration of CAS*. When RAS* falls, it is used by the VRAMs to determine whether to do a regular or Transfer cycle. When CAS* is low, it is used as the output enable (which is overridden by WE* like all x4 DRAMs).

*Table 6-7. Address Multiplexing*

T-49-17-01



| MA | Column Address All Sizes | Row Address Choices | Row Address Assignment for: 256K Non-Interleave | 265K Interleave | 1 MB Non-Interleave | 1 MB Interleave | 4 MB Non-Interleave | 4 MB Interleave |
|---|---|---|---|---|---|---|---|---|
| 0 | 2 | 11, 20 | 11 | 20 | 20 | 20 | 20 | 20 |
| 1 | 3 | 12, 21 | 12 | 12 | 12 | 12 | 12 | 21 |
| 2 | 4 | 13 | 13 | 13 | 13 | 13 | 13 | 13 |
| 3 | 5 | 14 | 14 | 14 | 14 | 14 | 14 | 14 |
| 4 | 6 | 15 | 15 | 15 | 15 | 15 | 15 | 15 |
| 5 | 7 | 16 | 16 | 16 | 16 | 16 | 16 | 16 |
| 6 | 8 | 17 | 17 | 17 | 17 | 17 | 17 | 17 |
| 7 | 9 | 18 | 18 | 18 | 18 | 18 | 18 | 18 |
| 8 | 10 | 19 | 19 | 19 | 19 | 19 | 19 | 19 |
| 9 | 11 | 20, 21, 22 | 20 | 20 | 21 | 22 | 21 | 22 |
| 10 | 23 | 22, 24 | x | x | x | x | 22 | 24 |
| 11 | - | 23 | - | - | x | x | 23 | 23 |
| Interleave bit | | | - | 11 | - | - | - | 12 |
| Page size | | | 1K | 2K | 2K | 4K | 2K | 4K |

## DRAM Cycles for DMA/Master Cycles

For DMA and master cycles, the DRAM controller operates in two different ways, depending on whether Wingine Transfer cycles are enabled. With Wingine transfer cycles disabled, the MEMR* and MEMW* signals generate RAS* asynchronously, and the cycles can complete without adding wait states. With Wingine transfer cycles enabled, MEMR* and MEMW* must be synchronized, because the DMA or master access must be arbitrated with the transfer cycles. VRAM transfer cycles cannot be delayed until the master is finished.

Two configuration bits determine whether IOCHRDY is pulled low for Master Reads and Write cycles. At lower clock rates, wait states must be added if Wingine is enabled. Wait states are not required for DMA cycles since they are much longer.

ISA masters which ignore IOCHRDY may not work when Wingine is enabled.

CHAPTER 7

T-49-17-01

# 4021 Cache Operation

The 4021 cache is smaller and faster than system memory. The cache memory enhances the system memory performance and acts as a buffer between the fast local bus and the slower system memory bus. Therefore, having cache memory allows the system designer to use slower system memory (at a lower cost) without sacrificing the high bandwidth of the local bus.

## 4021 Cache Architecture and Terminology

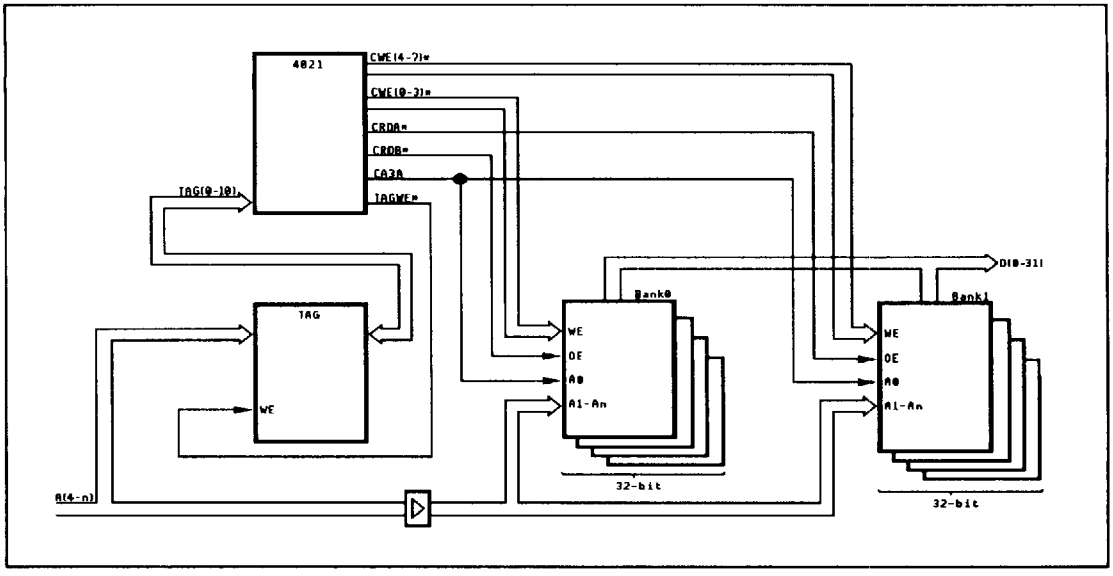
The cache is implemented using two sets of external static RAMs called the data RAMs and the tag RAMs. The tag RAMs contain the local bus address of the data stored in the data RAMs. If a local bus device needs to access system memory, the address in the tag RAMs informs the 4021 cache controller whether the accessed location is in the data RAMs or not. If it is in the data RAMs, it is a cache hit. If it is not in the data RAMs, and the 4021 needs to fetch it from system memory, then it is a cache miss.

The cache hit comparator is in the 4021 whereas the tag RAMs are external. The local bus address from A2 to A15, for a 64K cache, addresses the tag RAMs. The tag RAM data goes to the 4021 where it is compared with the higher local bus address bits. The highest address bits, which are not part of the tag data, are all compared to 0.

The 4021 cache architecture is optimized for the 486 burst read cycle (see Figure 7-1). The data RAMs are organized as two banks. The two banks are interleaved about address A2. Each bank gets a separate set of control signals, the data RAM output enable CRDa and CRDb. The write enable bits contain the byte information. This requires four write enables for each bank (CWE3:0 and CWE7:4).

*Figure 7-1. Cache Architecture*

T-49-17-01




The cache line size is always 16 bytes. A cache line is all the bytes in the data RAM that each entry in the tag points to in the 4021. A complete cache line must be updated for any cache read miss.

A *dirty line* is a cache line that has not been updated in system memory. The valid data is only in the cache. In the 4021, even if one byte in a line is updated, it will be flagged as a dirty line.

A *dirty bit* is set if it points to a dirty line. The 4021 requires one tag data bit to perform this function.

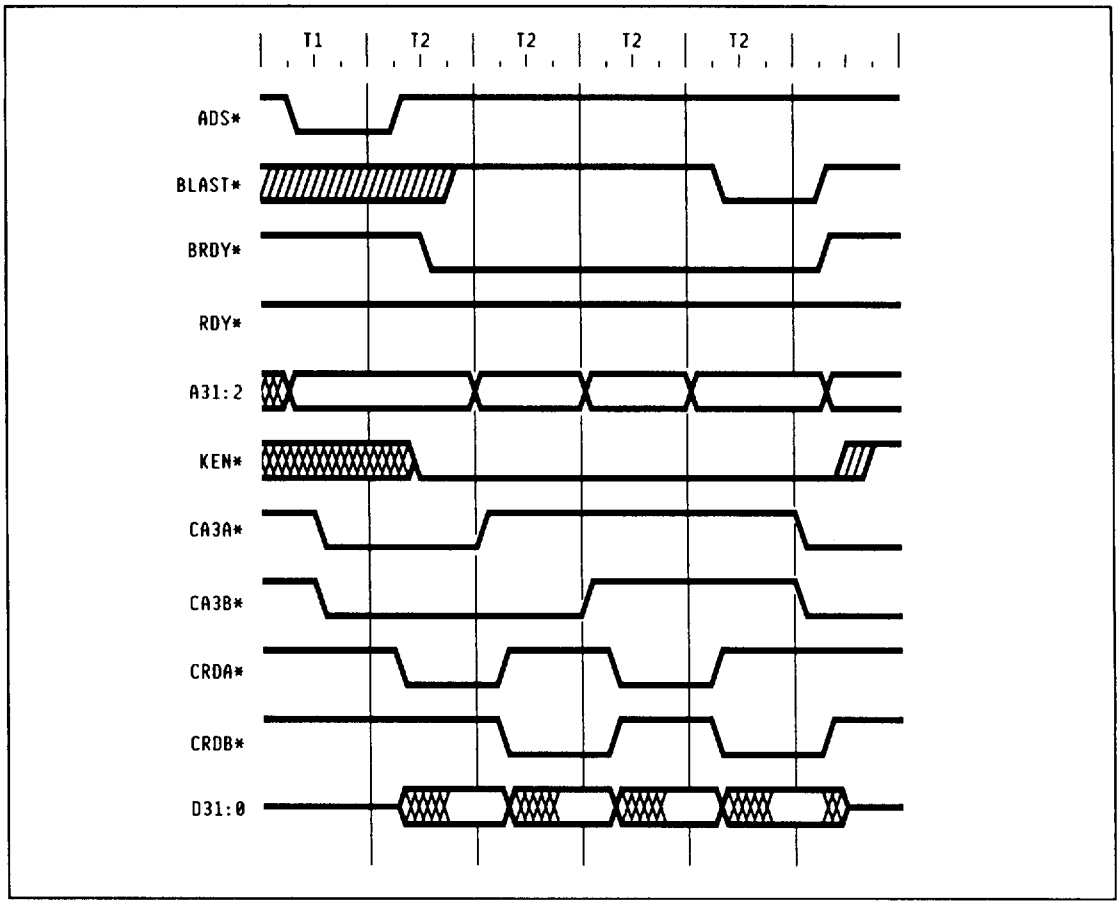
## Cache Read Hit Cycle

The cache read hit cycles are zero wait cycles. The hit/miss decision must be made before RDY*/BRDY* is returned to the CPU at the end of the first T2 (see Figure 7-2). The output of internal tag comparator generates the BRDY* signal for a 486 CPU and a RDY* for a 386 CPU, if it compares the accessed location to be a cache hit. The data to the CPU, in this case, is provided by the data RAMs.

If the cache is enabled, the 4021 always assumes the read cycles to be a cache hit. The CRDx signal goes active (output enable to the data RAMs) at the end of T1, and at the end of the first T2, the CRDx goes inactive. If the cycle is a cache read miss, RDY*/BRDY* generation is delayed until the data is fetched from the accessed memory. On a 486 read cycle, BRDY* remains low until the clock edge where BLAST* is detected active and the CRDx toggles on every T2.

*Figure 7-2. Cache Read Hit*

T-49-17-01




For optimum memory bandwidth utilization, the cache memory requires doing a burst read at a 2-1-1-1 rate in a 486 CPU based system; a new double word of data must be provided on each clock.

A 486 burst cycle is for four double words, so only addresses A2 and A3 need to change. The remaining address bits remain the same throughout the burst cycle. The addresses A2:3 always change in a sequence of 0-4-8-C, 4-0-C-8, 8-C-0-4, or C-8-0-4. The CPU changes A2 and A3 only after sampling BRDY* active at the end of every T2.

If the CPU-generated burst address of A2:3 is used, the address access time from the data RAMs is not sufficient to meet the data setup requirement of the CPU. In this case, wait states are needed, effecting the bandwidth gained in using the 486 burst cycle. The subsequent address, A2:3, in a burst read cycle (after the first read), is thus generated by the 4021. The cache architecture is optimized for the 486 burst read cycle. The 4021 has two banks of cache that are interleaved on a double word basis (address A2). Address A2 is used to control the output enable to the data RAMs and alternates the reads between the two banks of RAMs. The address access time requirement of the data RAMs is reduced by having separate output enables and address A3 to each bank.

Each of the SRAM banks has a separate CRD* and CA3 line. CA3 is the least significant address bit that goes to the SRAMs and the signal for bank 0 is CA3a and for bank 1 is CA3b. Similarly, the CRD for the two banks are the CRDa and the CRDb.

For the first read in the burst cycle, the value of A3 from the CPU is passed onto both CA3a and CA3b. At the beginning of the first T2, one of the CRD* pins goes low, as determined by A2 from the CPU (CRD0* goes low if A2=0; CRD1* goes low if A2=1). At the end of the first T2, the CRDa* and CRDb signals toggles, reading the other banks of SRAMs. At the same time, the CA3 for the bank that was just read is flipped to prepare for the third read. This allows two clocks of address access time. At the end of the second T2, the CRDs toggles again and the other CA3 is changed. This happens one more time to complete the burst. Figure 7-6 illustrates the 486 burst cycle.

## Cache Write Hit Cycle

The cache operates in write back mode or write through mode.

**Write Back Mode** - This mode requires an extra tag bit called the dirty bit, for each cache line. The dirty bit informs the 4021 if the cache line is updated in the system memory or not.

The data is always written to the data RAMs. The dirty bit is checked. If it is clean (zero), the tag must be rewritten to make it dirty (one). This makes the cycle one wait state. If it is dirty, the tag is not rewritten and the cycle completes in zero wait states. There is no write to the system memory for a write hit cycle in the write back mode.

**Write Through Mode** - The data is always written to SRAM and to system memory. RDY*/BRDY* is not returned by the cache controller. The internal DRAM controller returns RDY*/BRDY*. If the write buffer in the 4025 is empty, RDY*/BRDY* is returned in zero wait states. If the write buffer is busy, the cycle is completed only after the data is latched in the 4025 write buffer.

The 4021 performs better in write back mode than it does in write through mode. In write back mode, writing then reading a dirty location in the cache always occurs in zero wait cycles. In a write through mode, because all writes are updated to cache and system memory, the CPU must wait until the write to the system memory is completed (unless writes to system memory are buffered). For the write through mode to be effective, buffered writes must be completed as fast as the cache write. This is possible only if the writes are within the same DRAM page.



## Cache Read Miss Cycle



The tag comparator in the 4021 begins a read miss cycle if the tag data does not compare with the accessed location. Only CRDx signals to the data RAMs have similar timing as in a read hit cycle, since the CRSx signal is generated even before the hit or miss decision is made. One of the CRDx* signals will be low during the first T2. It will be taken high when the 4021 discovers that it is a read miss cycle. The RDY*/BRDY* is not active for this case and wait states are added until data from the system memory can be read. Therefore, the 4021 drives ready only when it has data available.

To read system memory, the 4021 goes through the following steps:

1. Determine whether the address is cacheable in the second level cache.

   The cacheable decision comes from the DRAM bank decodes, the 4021 configuration registers that define the non-cache regions, and the bits to determine whether the shadow area is cachable or not. If the access is not from system memory, then it is always non-cacheable (AT memory is not cached).

2. Determine if the 4021 is in the write through or the write back mode.

   **If write through mode:** Read cache line from the system memory to the CPU and write it to the data RAMs (cache linefill cycle). For a 486 CPU, the data is sent to the CPU and the data RAMs simultaneously (see the following section titled *Conditions*). For the 386 CPU, the cache line is first written to the data RAMs, and then the CPU requested double word is read from the cache.

   If write back mode and cache line is dirty:

   Write the dirty line from the data RAM to system memory (cache castout cycle) into the 4025 buffer chip. If the castout buffer is occupied, the 4021 waits for it to be freed up. When the buffer is available, the data is transferred from the SRAMs to the 4025 in four clock cycles. The 4025 latches the data on the rising edge of each clock.

   Now read the new line from system memory to the CPU and write it to the data RAMs. For the 486 CPU, the data is sent to the CPU and the data RAMs simultaneously. Note that for the 386 CPU, the cache line is first written to the data RAMs and then the CPU requested double word is read from the cache.

   The BRDY*/RDY* is returned to the CPU and it can now begin a new cycle while in the background the castout cache line in the 4025 is written into the system memory.

If write back mode and cache line is clean: T-49-17-01

Read cache line from the system memory to the CPU and write it to the data RAMs. For a 486 CPU, the data is sent to the CPU and the data RAMs simultaneously. Note that for the 386 CPU, the cache line is first written to the data RAMs, and then the CPU requested double word is read from the cache.

3. Update the tag RAM.

   The tag RAM is updated simultaneously with the line fill or the castout cycle; as the case may be.



## Conditions

The 486 may not be caching the access, however, and may not read all four double words (if PCD is high or a locked transfer is occurring or if it is a 386 system). In this case, the 486 will finish its burst early, after one or two accesses (for a 386 after one access). If this happens, the CPU may come out with another access immediately.

To avoid this, the cache controller monitors BLAST* or the 386 mode bit on each cycle. If BLAST*/386 mode bit goes low prior to the end of the third transfer, RDY*/BRDY* is not generated for the rest of the cache line fill. This allows the line to be filled while the CPU waits for the final data word. After the cache is filled, the proper word is read back out of the cache and RDY*/BRDY* is returned. The sequence below illustrates a burst of 2.

First word:

BLAST* high
Data written to cache and BRDY* generated.

Second word:

BLAST* goes low.
Data written to cache. BRDY* not generated.

Third word:

Data written to cache. BRDY* not generated.

Fourth word:

Data written to cache. BRDY* not generated.

Finish cycle:

Data read from cache (second word)
BRDY* generated.

The address of the word that must be read from the cache to finish the cycle is taken from A2 and A3 from the CPU. It is stable at this point. This information is used to generate CA3a, CA3b, and the proper CRD*.

T-49-17-01

## Cache Write Miss

During a write miss, the data is written only to the system memory the data and the tag RAMs are not updated. If the target is a local peripheral or the AT bus, the cache controller is not activated. If the target is local DRAM, the DRAM controller takes the cycle. If the write buffer is available, it is stored there and the cycle ends with zero wait states. The data buffer stores the data on the rising edge of SCLK clock.

***Figure 7-3.*** *Write Hit Clean*

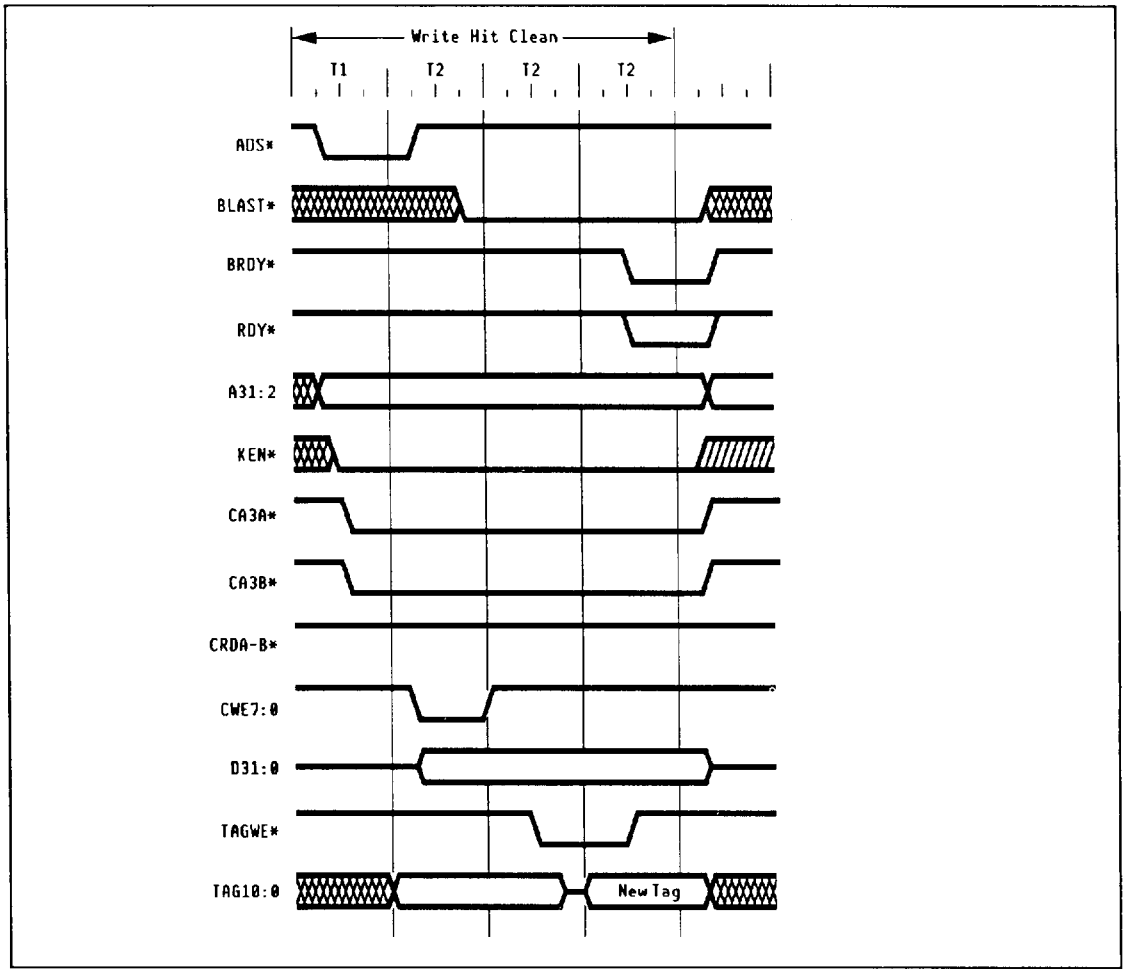
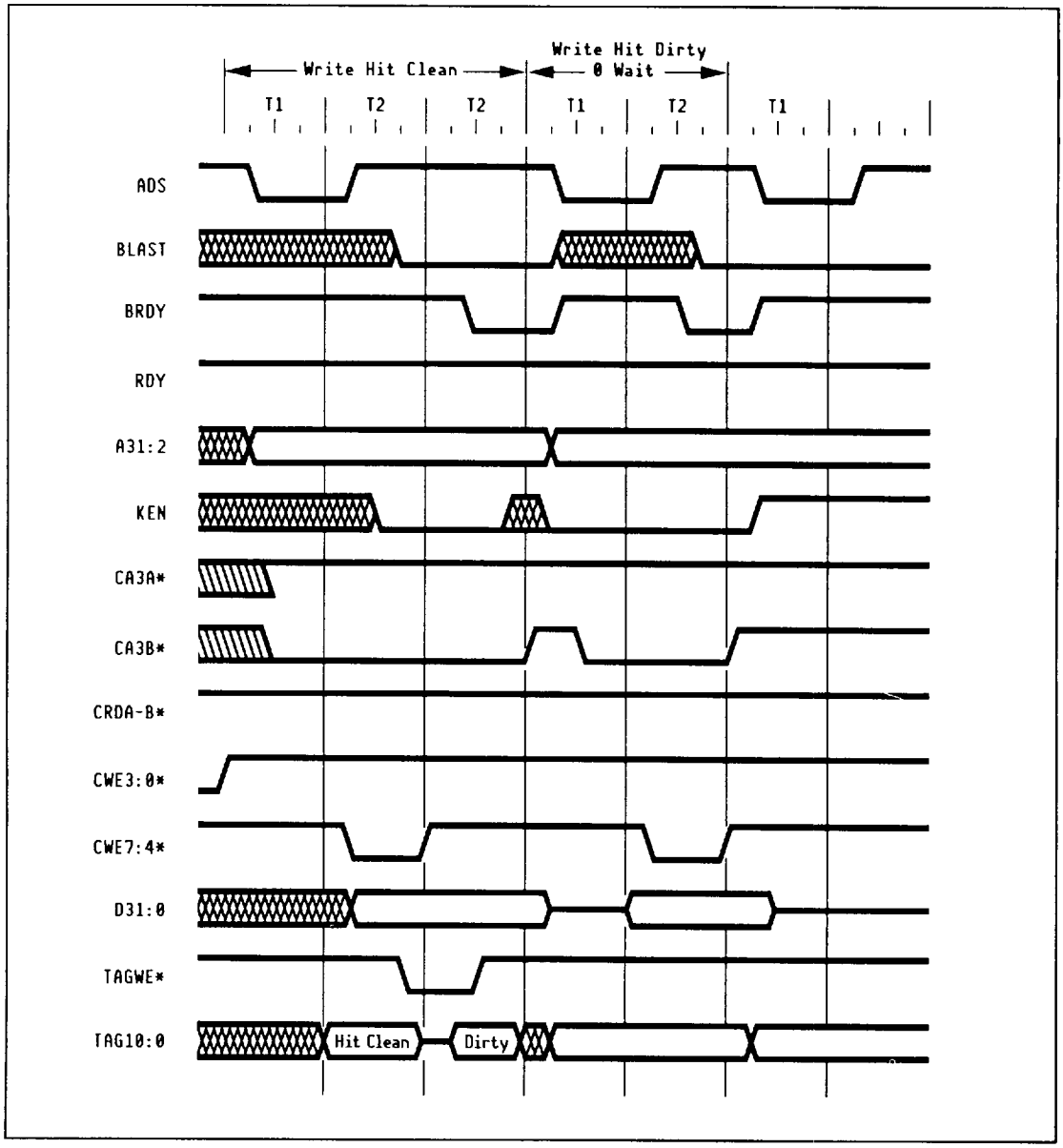


*Figure 7-4. Write Hit Dirty*

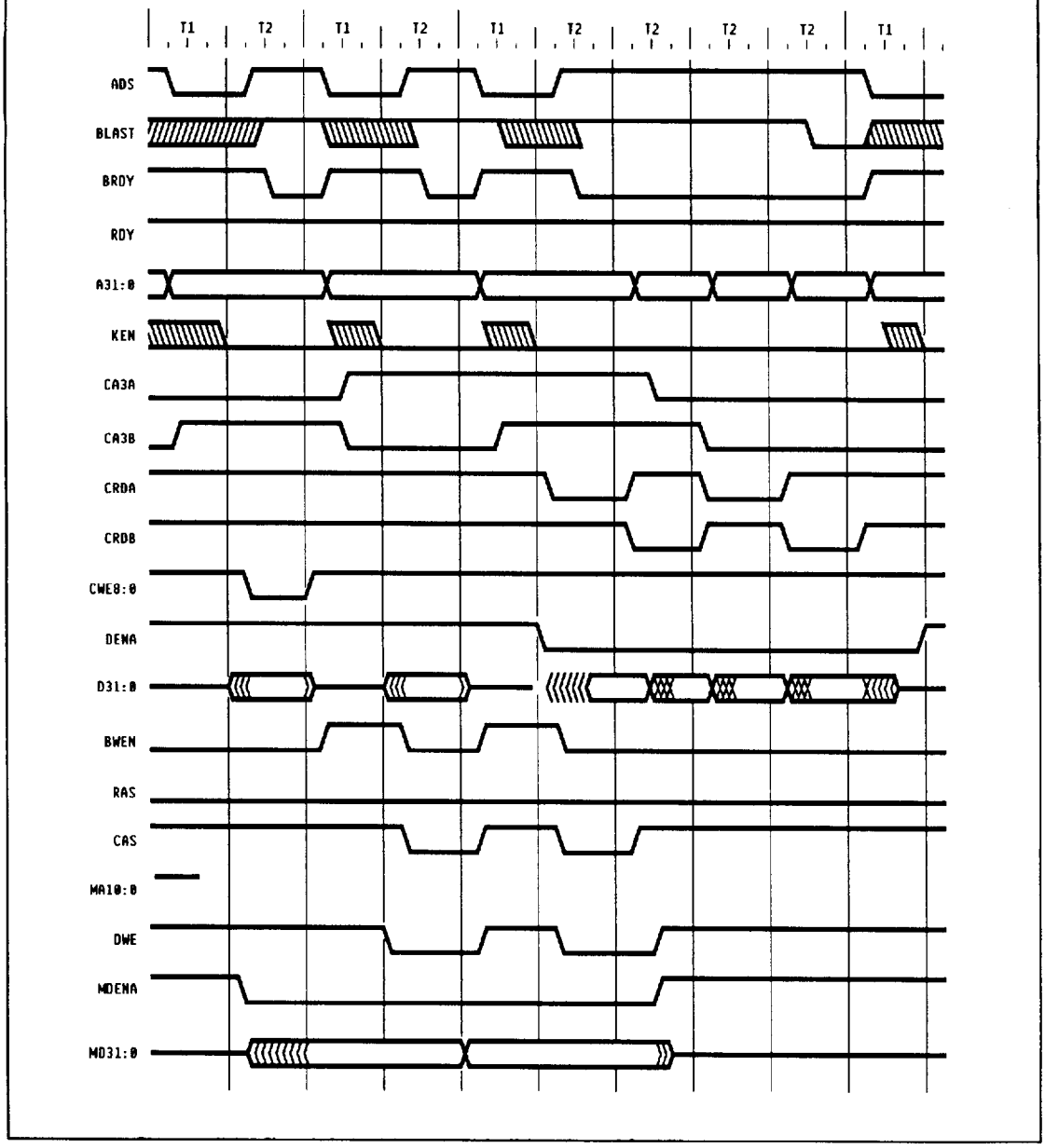
T-49-17-01



T-49-17-01

***Figure 7-5.*** *Write Through Cache Write Hit, Cache Write Miss and Read Hit*

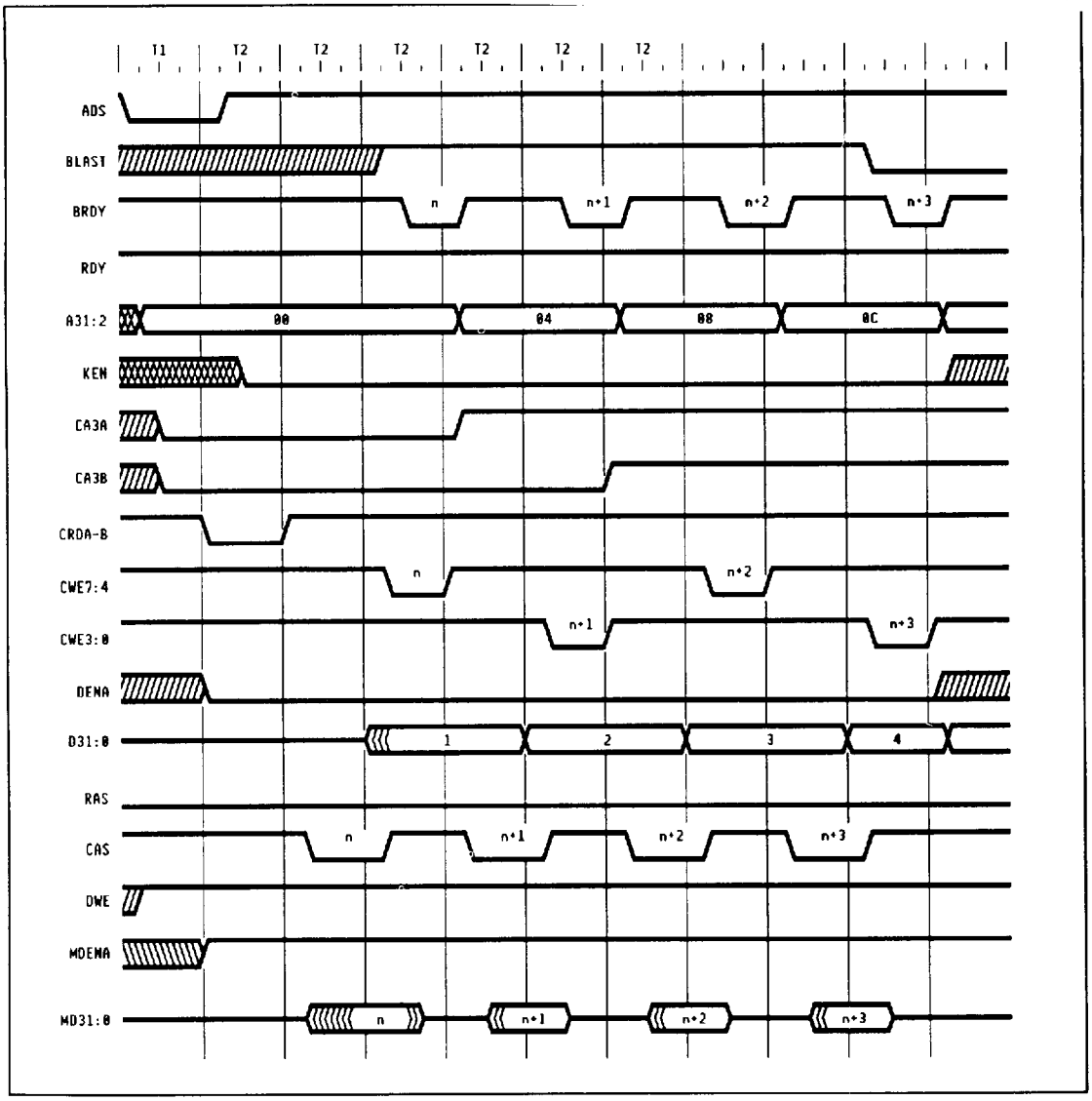




T-49-17-01

*Figure 7-6. Read Burst Line File*





T-49-17-01

*Figure 7-7. Castout — Read Hit*



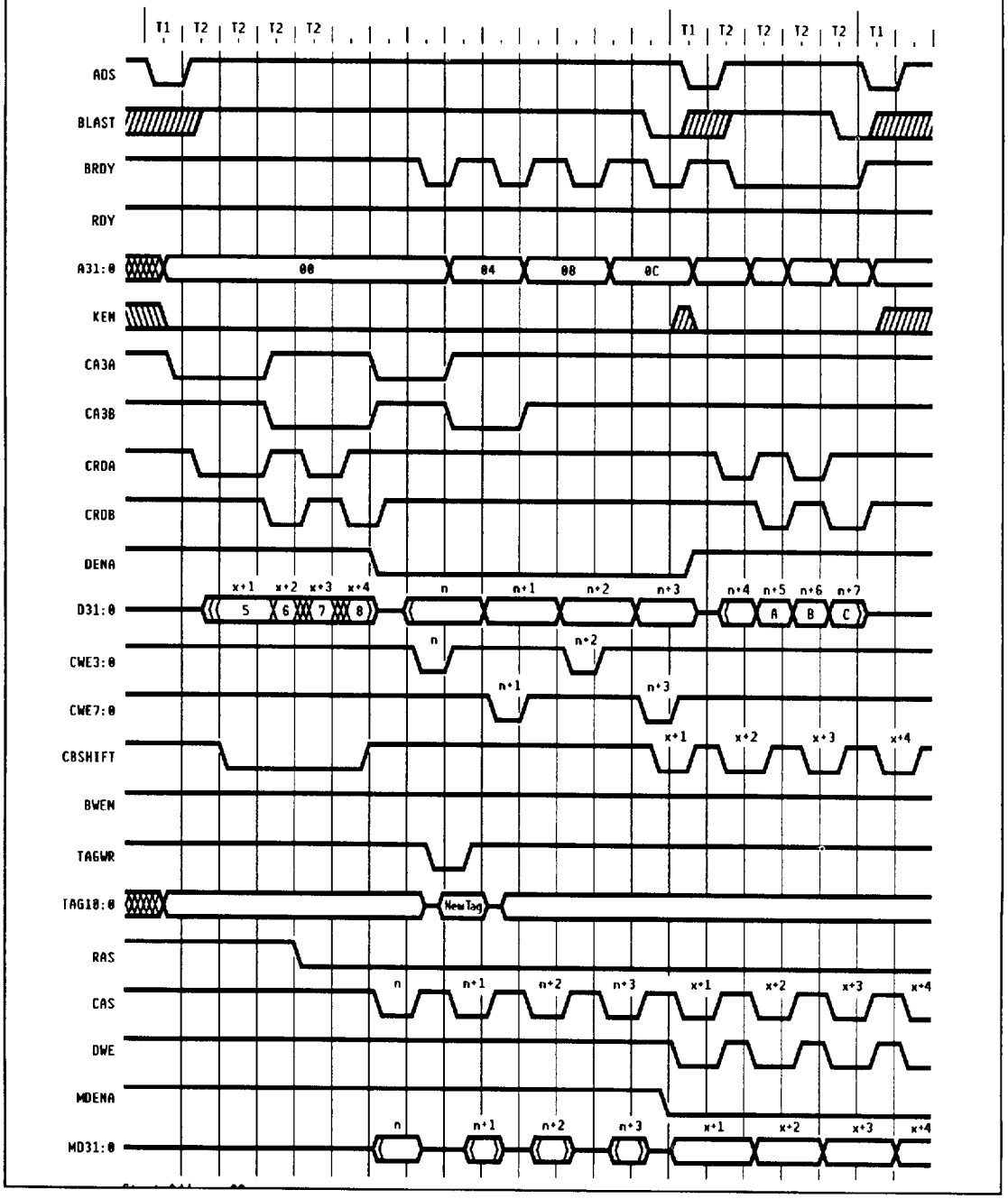
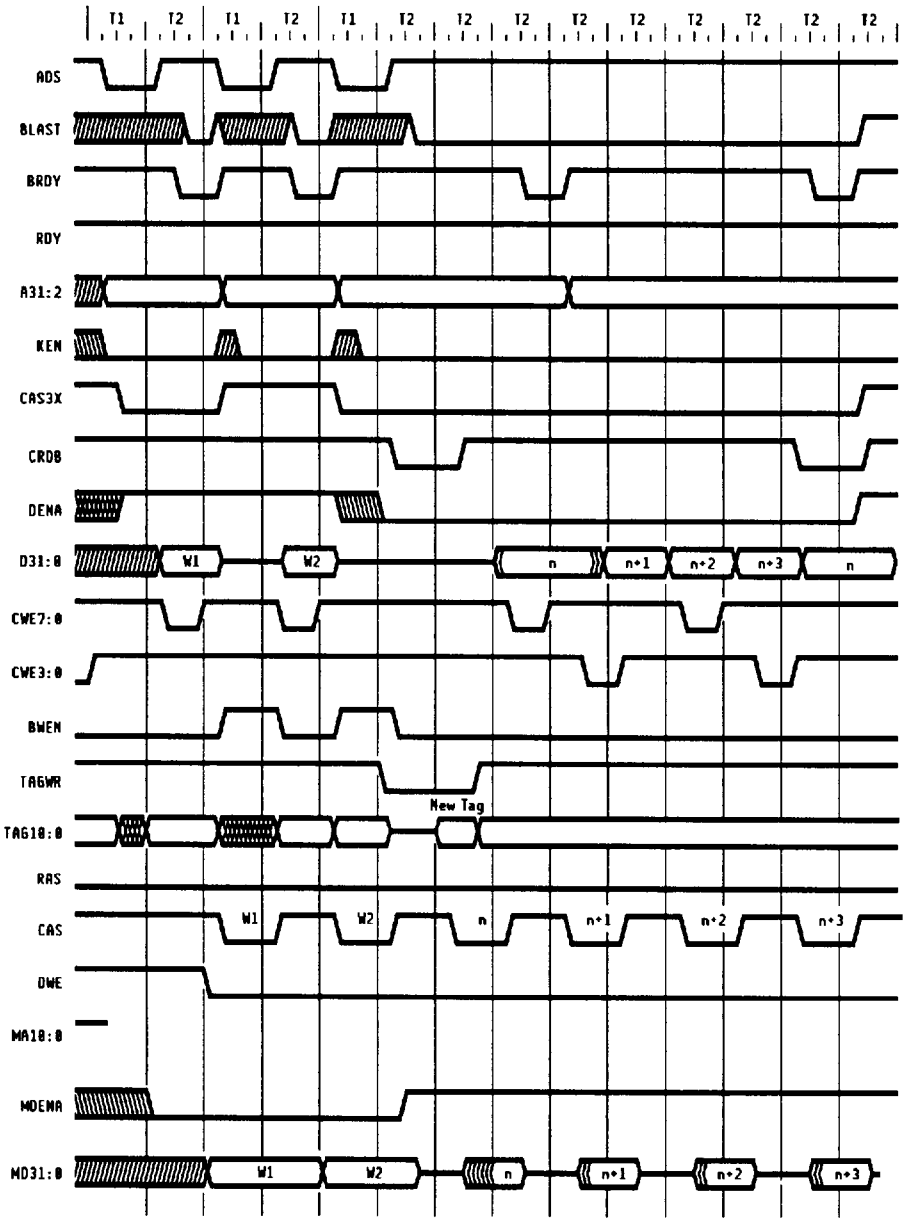


*Figure 7-8. Cache Write Miss — Cache Read Miss*

T-49-17-01





T-49-17-01

# Cache Mode and Initialization



To facilitate initialization of the cache and disabling of the cache, the following cache control bits have been defined.

**WRMODE Write Mode** - Selects the write-hit policy.

| | |
|---|---|
| 0 | write through |
| 1 | Write back |

**ENCACHE Enable Cache** - Enables or disables cache.

| | |
|---|---|
| 0 | Disable cache |
| 1 | Enable cache |

**INITCACHE Initialize cache** - All cacheable memory read cycles update the cache. No castouts are performed. Write hits are handled as write through. This mode should be used when the cache is disabled.

| | |
|---|---|
| 0 | Normal operation |
| 1 | Initialize cache mode |

**FRZCDIR Freeze cache directory**

| | |
|---|---|
| 0 | Normal cache operation |
| 1 | No directory update. No updates on cache miss. No castouts occur either. This mode emulates direct static RAM. |

## Initializing the Cache

The tag RAM comes up with random data, including the dirty bit. Since the cache contains no valid bits, the cache must be initialized by the BIOS. Once the cache is initialized, it is always valid. The cache is initialized by allowing the normal read-miss process to fill the cache with valid data. This is done by reading a block of memory which is the size of the cache. Read-hits must be disabled to prevent an uninitialized tag from accidently indicating a hit, and force the read cycle to fill the cache with valid data. Castouts must be disabled to prevent invalid data from being cast out into the DRAM during initialization.

When set, the INITCACHE bit overrides the FRZCDIR bit, which should be set to 0. It also overrides the WRMODE bit, which may be set to 1 or 0, and may be set to the desired mode throughout the procedure.

T-49-17-01

## Initialize Cache Algorithm

ENCACHE=0, FRZCDIR=0, INITCACHE=0 to start. WRMODE=X. Set INITCACHE=1.

Read a block of cacheable memory that is at least as large as the cache size.

At the same time, set ENCACHE=1, INITCACHE=0, and WRMODE to desired mode.

**Disabling the Cache:** Disabling a write back cache cannot be done by just turning off the ENCACHE bit in the 4021 CHIPSet register. This is because data is valid only in the cache and not in the DRAM. If the disabling of the write back cache is not handled carefully, it can result in a system crash, if there is dirty data in the cache. To avoid this, a mechanism is provided to dump the dirty data back into the DRAMs. This involves reading a cacheable memory block twice the size of the cache. Twice the size is required to make sure each location gets a read-miss, which causes a cast-out if the dirty bit is set. The cache is then disabled. No writes should occur during this process.

**Disabling the Cache: Write Through:** ENCACHE=1, FRZCDIR=0, INITCACHE=0, WRMODE=0 to start (normal write through mode). Set ENCACHE=0.

**Disabling the Cache: Write Back**: ENCACHE=1, FRZCDIR=0, INITCACHE=0, WRMODE=1 to start (normal write back mode).

Read a block of cacheable memory that is at least twice as large as the cache size. No cacheable write cycles should be done during this time until after the next step. Set ENCACHE=0.

**Switching between Write Back to Write Through:** To switch from write through to write back, nothing special need be done, just flip the bit. To switch from write back to write through, the dirty data must be cast out. A block of cacheable memory twice the size of the cache must be read, the WRMODE bit should be set to a 0.

**Switching from Write Through to Write Back**: ENCACHE=1, FRZCDIR=0, INITCACHE=0, WRMODE=0 to start (normal write back mode). Set WRMODE=1.

**Switching from Write Back to Write Through**: ENCACHE=1, FRZCDIR=0, INITCACHE=0, WRMODE=1 to start (normal write back mode).

Read a block of cacheable memory that is at least twice as large as the cache size. No cacheable write cycles should be done during this time until after the NEXT step. Set WRMODE=0.

**Freezing the Cache Directory**: While the cache is operating in write back or write through mode, the directory can be frozen to prevent new data from going to the cache. Writes occur normally, including updating the dirty bit, if necessary. Read misses do not update the cache data or tag RAMs. This bit may be turned ON and OFF at any time, except when initializing the cache. A possible use is for preventing large data moves from thrashing the cache.

T-49-17-01

**Freezing and Unfreezing the Cache Directory**: This bit can be changed without causing cache coherency problems.



## Cacheability and Write Protection

Only local DRAM is cacheable in either the 486 or secondary cache. Shadow RAM is cached unless defined as non-cachable area. Shadow RAM can be write protected, but this causes a problem in the 486 cache. It is not possible to write protect the 486 cache.

Several situations that occur in ATs require shadowed BIOSes to be write protected:

- Some programs search for ROMs by writing a location and reading it back to see if the data has changed. Write protection properly simulates a ROM in this case.
- Some adapter boards have write-only registers memory mapped in the BIOS areas.
- Some programs do a floppy disk verify (checking the CRC) by aiming the DMA address at the ROM.

The first two situations cause problems for the 486 cache. Problems can be avoided in the following ways:

- **Solution 1.** Do not write protect the BIOS. This results in the highest performance but does not handle the problems listed above.
- **Solution 2.** Drive EADS* low on writes to write-protected areas. This is only a partial solution and it does not work in all cases.
- **Solution 3.** Cache only in secondary cache, but still bursting to the 486. This is the recommended solution. It works 100 percent but it causes the BIOS to run a little slower.

All these solutions are supported by the 4021 CHIPSet.

**Solution 1** is implemented by not write protecting the BIOS area.

**Solution 2** is implemented by driving EADS* low when a write to a write protected area occurs. This invalidates the 486 cache line. When the 486 next reads the data, it will read it from the secondary cache or DRAM, which are write protected. There is no need to use AHOLD, etc. to drive the address. The 486 is invalidated based on the address it is driving. This type of access goes to the ISA bus, so there is sufficient time to drive EADS*. It can be driven at the end of the bus cycle, at the same time as READY. The 486 samples EADS* at the end of the final T2.

The problem with this solution is that writes are posted in the 486. By the time the invalidation occurs, the 486 could have already read the location from cache. The 486 can perform many instructions before the write gets to the bus (causing the invalidation). The four write buffers contribute to this, and the 486 may be in HOLD at the time. 

**Solution 2** should be used for the 38605 CPU because it invalidates on write hits. The EADS* signal is ignored since it is only sampled during HLDA. T-49-17-

**Solution 3**, caching only in secondary cache is the recommended solution. It is implemented by driving KEN* low for the first word. KEN* is driven high for the final word. When this is done, the 486 does not place the data in its cache. In most cases, the data goes to the prefetch queue, so bursting the data is beneficial. The secondary cache and DRAM are write protected.

A configuration bit determines whether solution 2 or 3 is used.

## KEN* Generation

KEN* is sampled by the 486 one clock before the first ready of a transfer. If it is low, the 486 does a cache line fill (assuming the 486 internal cache is enabled, the page is cacheable, etc). The 486 again samples KEN* one clock before the final ready of the cache line fill. If KEN* is low, the data is put in the cache. If KEN* is high, the data is discarded.

On the first transfer, KEN* is sampled at the start of every T2. The 486 drives BLAST* high or low but otherwise takes no action until READY is received. KEN* can change states as often as desired before READY (however, it must meet setup and hold times for each clock). For a 0WS read, the value of KEN* at the end of the first T1 is used. It is not possible to determine cacheability by this time, so KEN* is driven low for T1 (except in the Weitek coprocessor). If the access is a cache hit, KEN* should be driven low. If the access is a cache miss, KEN* needs to be taken high since only cache hits finish in 0WS. As soon as the cacheable/non-cacheable decision is made, KEN* is driven appropriately. KEN* is set up for page hits by the end of the first T2 for 3-1-1-1 DRAM mode.

For the Weitek coprocessor in 0WS Weitek mode, KEN* is driven high with A31 asynchronously. In 1WS Weitek mode, KEN* is clocked at the end of T1.

For write protected areas, KEN* is optionally taken high for the last access of a burst.

If a 486 CPU is used, Wingine cycles are cached in the 486 internal cache (not in the secondary cache). Using internal cache forces string moves to occur in four reads, then four writes to DRAM rather than in read/write/read/write order. This increases the VRAM page hit rate from 0 to 75 percent. It is the only case where the secondary cache does not follow KEN* for cacheability. KEN* will be low, but the secondary cache will consider the VRAM to be non-cacheable.

## DMA/AT Master Access

The DMA controller and AT masters perform memory operations by pulling MEMR* or MEMW* low. The memory controller must synchronize these signals with the CPU clock and perform the requested memory operation. With the write-back cache enabled for AT master or DMA read cycle, the valid data must be read from the cache on read hits. The cache contains the only valid copy of the data. For a cache read miss, the valid data comes from the DRAM.

For a memory write cycle, the data is always written to the DRAM. It is also written to the SRAM if it is a cache hit. The dirty bit is not updated in the case of a clean line.

The 4025 selects the cache data bus or the DRAM data bus to act as input to the AT data bus. The 4025 also handles the 32-bit to 16-bit, or 8-bit AT data bus translation.

An EADS* signal is generated for all DMA and AT master memory write cycles to invalidate the 486 internal cache. The 4021 also drives A25:24, A31, and optionally A26 low when HLDA1 is active because the DMA controller and AT master only provide A23:0. This is required for invalidating the 486 or 38605 cache. A30:27 are driven low by an external TTL driver.

# Cache Coherency

Coherency must be maintained in the primary and secondary caches when local masters, ISA masters, and DMA cycles occur. Cache coherency is handled differently for secondary cache, 486 primary cache, and Super386 ICACHE.

**Secondary cache**: Coherency is maintained by always writing the data in the cache on write hits, regardless of the current master. There are no valid bits in the tag, so every cache location must be valid at all times after initialization. Coherency on reads is maintained by always reading the data out of the cache on read hits, to prevent stale data from being read from the DRAM. This is required because the secondary cache is a write-back cache.

**486 Primary cache**: Coherency is maintained by driving EADS* low for all local master, ISA master, and DMA memory writes. This invalidates the addressed cache line in the 486. The address is already provided on the local bus by the DMA/ISA master. For this reason, the A24:31 signals are driven low externally when ISA masters and the DMA controller have the bus.

**Super386 ICACHE:** The Super386 EADS* pin is compatible with the 486. EADS* can also invalidate an internal cache line when HLDA is active: ADS* goes active, and the status signals indicate a memory write. The 4021 EADS* pin is used as AADS* for the Super386. Coherency is maintained by driving ADS* for ISA master and DMA memory writes. The status signals, W/R*, D/C*, and M/IO*, are also driven. The 4021 generates a one wait state RDY* to maintain the local bus protocol.



*Note:* The 4021 CHIPSet does not generate the RDY*/BRDY* signals if the DMA/ISA master access is to a local bus slave. When a local master has control, it must drive ADS* or EADS* and the status signals as part of the standard protocol to invalidate the CPU internal cache.



## Changing a DRAM Write-Protect Bit

Write-protect bits can be changed without having to reinitilize the cache. With a write-back cache, the write protection is handled in a different way.

If an area is write enabled in the cache, and written to so that it becomes a dirty line, valid data is held in the cache, and stale data is in the DRAM. If DRAM is then write protected, future writes to the area will return a 0WS READY. The dirty bit is set (the ready logic in the 4021 ignores the write-protect decode when the line is dirty), and the data is not written to the data RAM.

To write to DRAM, the CWEx signals are not generated. The AT state machine will not stop operating because BRDY*/RDY* are active (on a normal write-protect cycle the AT bus would be written to). Eventually, when the data is cast out of the cache, it is written to the DRAM, even though it is "write protected." Castouts ignore the write-protect decode. Since the data was written when the area was write enabled, this is what should occur.

# SRAM Requirements

The 4021 cache controller uses standard SRAMs for tag and data. The data RAMs are D-word interleaved in all configurations (64 bits of SRAM are required to support interleaving). This allows 486 systems to do a full 2-1-1-1 burst, and allows all systems (386 or 486) to cast out a dirty line of data to the 4025 buffer at one T-state per D-word. Separate CA3, CRD*, and CWE* signals are supplied for each of the interleaved D-words. The CWE* signals contain the byte enble information. There are four for each cache bank, or eight total. CA3 and CRD* are separate from each interleaved D-word to provide the interleaved read timing.

## Cache Critical Paths

T-49-17-01

The OE access time of the SRAMs is a critical path. It consists of the clock to CRD* timing, the SRAM tOE timing, and the CPU setup time.

The critical path for the SRAM address access is the A3 to CA3 delay time. It runs parallel to the external address lines, which goes through an external F244.

The write data hold time in a 486 is only 3 ns from the end of the write cycle. To achieve a 0 wait write hit, the write enable to the SRAM is moved 5 ns earlier by using an external delay line. This is at the expense of the data setup time of the SRAM. The SRAMs require a setup time, tDW, of less than 10 ns for a 33MHz 486 system.

The write pulse width is the critical parameter for writes. Nominally, at 33 MHz, it will be 15 ns.

The tag RAM read path is critical. The hit or miss decision must be made early in T2. The worst case is the write cycles, where CWE* goes low about 5 ns before the middle of T2.

The two most practical cache sizes are 64K and 256K because they only require eight data RAMs. 128K and 512K require 16 SRAMs (less if non-standard SRAMs are used). Table 7-1 shows common data SRAM configurations. Table 7-2 shows tag SRAM configurations for the 486 CPU.

***Table 7-1.*** *Data SRAM Configurations*

| | Data RAMs | |
|---|---|---|
| **Cache Size** | **Quantity** | **Type** |
| 64 Kbytes | 8 | 8Kx8 |
| 128 Kbytes | 16 | 16Kx4/8kx8 |
| 256 Kbytes | 8 | 32Kx8 |
| 512 Kbytes | 16 | 64Kx4/32kx8 |

*Table 7-2. Tag SRAM Configurations*

T-49-17-01

| Cache Size | Tag Depth | 8-Bit Tag Qty | 8-Bit Tag Type | 8-Bit Tag Cacheability | 9-Bit Tag Qty | 9-Bit Tag Type | 9-Bit Tag Cacheability | 11-Bit Tag Qty | 11-Bit Tag Type | 11-Bit Tag Cacheability |
|---|---|---|---|---|---|---|---|---|---|---|
| 64 Kbytes | 4K | 2 | 4Kx4 | 8MB | -[3] | -[3] | 16MB | 3[2] | 4Kx42 | 64MB2 |
| 128 Kbytes | 8K | 1 | 8Kx8 | 16MB | 1[2] | 8Kx92 | 32MB[2] | 2[2] | 8Kx82 | 128MB[2] |
| 256 Kbytes | 16K | 2[2] | 16Kx4[2] | 32MB2 | -[3] | -[3] | 64MB | 3 | 16Kx4 | 256MB[1] |
| 512 Kbytes | 32K | 1[2] | 32Kx82 | 64MB[2] | 1 | 32Kx9 | 128MB | 2 | 32Kx8 | 512MB[1] |

Notes:
1. Maximum addressability of the 4021 is 128M for local DRAM.
2. Most common configurations.
3. 64K and 256K 9-bit tag is blank because there are no common 9-bit RAMs of this size.

Tables 7-3 and 7-4 show the cacheability range for all combinations of cache size and tag width.

*Table 7-3. 486 CPU Data and Tag SRAM Speed Requirements*

| Parameter | Description | 20MHz | 25MHz | 33MHz |
|---|---|---|---|---|
| | **Data RAMS** | | | |
| **Recommended SRAM speed** | | 35 | 35 | 25 |
| tAA | Address access time | 47 | 37 | 28 |
| tOE | OE* access time | 28 | 20 | 16 |
| tWP | Write pulse width | 23 | 19 | 15 |
| tDW | Data valid to end of write | 30 | 15 | 10 |
| | **Tag RAMs** | | | |
| tAA | Address access time | 35.5 | 22 | 15 |

T-49-17-01

*Table 7-4. 386 CPU Data and Tag SRAM Speed Requirements*



| Parameter | Description | 20MHz | 25MHz | 33MHz |
|---|---|---|---|---|
| | **Data RAMs** | | | |
| **Recommended SRAM speed** | | 35 | 25 | 25 |
| tAA | Address access time | 45 | 34 | 29 |
| tOE | OE* access time | 27 | 20 | 16 |
| tWP | Write pulse width | 23 | 19 | 15 |
| tDW | Data valid to end of write | 35 | 33 | 19 |
| | **Tag RAMs** | | | |
| tAA | Address access time | 35 | 29 | 23 |

The tag RAM may be 8, 9, or 11 bits wide. The tag width affects only the cacheable address range of the DRAM. A wider tag is required for a small cache to achieve the same cacheable area.

Table 7-5 shows the tag values for different cache sizes.

T-49-17-01



***Table 7-5.*** *Tag Comparison Table*

| Tag | 64K | 128K | 256K | 512K |
|---|---|---|---|---|
| Tag0 | A16 | A26 | A26 | A26 |
| Tag1 | A17 | A17 | - | - |
| Tag2 | A18 | A18 | A18 | - |
| Tag3 | A19 | A19 | A19 | A19 |
| Tag4 | A20 | A20 | A20 | A20 |
| Tag5 | A21 | A21 | A21 | A21 |
| Tag6 | A22 | A22 | A22 | A22 |
| Tag7 | A23 | A23 | A23 | A23 |
| Tag8 | A24 | A24 | A24 | A24 |
| Tag9 | A25 | A25 | A25 | A25 |
| Tag10 | Dirty | Dirty | Dirty | Dirty |
| | **Tag bits to disable:** | | | |
| 8-bit tag | 7, 8, 9 | 8, 9, 0 | 9, 0, 1 | 0, 1, 2 |
| 9-bit tag | 8, 9 | 9, 0 | 0, 1 | 1, 2 |
| 11-bit tag | none | none | 1 | 1, 2 |

## 4021 Write Buffers

The 4021 has two separate write paths: one for posted writes and the other for cache castouts. They are called the *posted write buffer* and the *castout buffer*, respectively.

**Posted Write Buffer** - Write buffering is used in both cache and non-cached modes. The address portion of the write buffer is contained in the 4021 chip, and the data portion is contained in the 4025 chip. The write buffer is a one-level buffer that can hold one CPU write cycle or one cache write miss cycle.

The 4021 provides a BWEN* signal to control clocking the data. The 4025 samples BWEN* on each rising edge of the CPU clock (the start of each T-state). When BWEN* is low, new data from the D bus is clocked in. When BWEN* is high, the flip-flops retain their old data.

The write buffer is the speed critical path in the D to MD direction. It is implemented as flip-flops to provide best control of finishing one write and starting the next one. On a CPU write, the data is latched into the flip-flops on the final T2. The 486 CPU provides only 3 ns of hold time here before floating. On back-to-back writes (page hits), the DRAM data hold time has been satisfied just before the new data is clocked in. The write buffer allows zero wait state back-to-back writes from the CPU (see Figures 7-3 and 7-4 shown earlier).



## Castout Write Buffer



The castout buffer is four double words and is used to store a dirty cache line when it is being replaced with a new line from DRAM. It receives data from the cache SRAMs during a dirty line castout. There is no simultaneous activity of reading in data from the local data bus and writing to the DRAM. Four double words are put into the buffer. Then these four double words are taken out to write to the DRAM. The castout buffer control signal consists of CBSHIFT*, which is a shift enable. If it is low, on the rising edge of SCLK the buffer shifts a double word to the MD bus. The action codes control the output enable of the shift register onto the MD bus.

During a castout, CBSHIFT* goes low and stays low for four clock cycles. When writing to DRAM it goes low at the end of each DRAM write (see Figure 7-5 shown earlier).

The write buffer and castout buffer are always cleared before allowing DMA or an AT bus master to take control. This is done by blocking HLDA1 until they are cleared.

## Tag and Data SRAM Testing

The 4021 has test modes for both the cache tag and data RAMs. The purpose of these modes is for the BIOS software to determine the size of the cache and test its functionality before enabling it.

Testing the data RAMs causes them to appear as static RAM within the test window. The CPU reads and writes to them as if they were static RAM banks. Testing the tag is more complicated since the CPU does not have direct access to the tag data. The tag data is stored in a register within the 4021, which the CPU reads and writes through I/O ports 22 and 23.

There are two locations at which the test window can be placed. One option places it in the bottom one MB of the address space, allowing it to be used in real mode. The other places it at the 1MB point where it must be used in protected mode. Register 22, bit 3 determines which mode is used. The addresses are shown in Table 7-6.

*Table 7-6. Cache Test Window Location*

T-49-17-01



| Reg 22, Bit 3 | Cache Size Bits | Test Window Address Range |
|---|---|---|
| 0 | 64K, 128K, 256K | 40000-7FFFF (256K to 512K area) |
| 0 | 512K | 20000-9FFFF (128K to 640K area) |
| 1 | All | 100000-17FFFF (1M to 1. 5MB area) |

The only effect of the cache size bits in either test mode is to determine the window size when the window is in the bottom MB. This allows more available DRAM when testing a cache that is 256K or smaller. The full 512K test window is required for tag sizing to determine whether the cache is 256K or 512K.

The CPU addresses go directly to the tag and data RAMs (through an F244 for the data RAMs). The test mode does no mapping of any kind. When there is a 512K test window in the bottom MB of memory, the address mapping is shifted, as shown in Table 7-7.

*Table 7-7. 512K Cache Test Mode Mapping When in the Lower MB*

| CPU Address | Physical SRAM Address |
|---|---|
| 80000-9FFFF | 00000-1FFFF |
| 20000-7FFFF | 20000-7FFFF |

## Data SRAM Testing

When Data SRAM testing mode is enabled, the cache appears as static RAM within the test window. The tag comparator is disabled (by virtue of the cache being disabled), and hits are forced whenever a memory cycle is done within the test window. All timing is the same as normal cache hits. Reads are zero wait states. Bursts are 2-1-1-1. Writes are also zero wait states and have the timing of a write hit dirty. The DRAM controller sees write cycles as cache hits, and is activated. All cycles outside the test window acts like a non-cacheable area; no line fills, etc.

The tag test registers, indexes 23 and 24, are actually separate registers for reading and writing. After writing index 23, reading it back gives the value stored in the register during the last read from the test window with tag testing enabled.

When data SRAM test mode is enabled:

T-49-17-01

- Reads within the test window reads the SRAMs and are 0WS. Bursts are 2-1-1-1.
- Writes within the test window write the SRAMs and are 0WS.
- The tag is ignored.
- No castouts occur.



- Memory accesses outside the test window go to DRAM, the AT bus, etc. and are not cached.
- No line fills occur.
- No tag writes occur.
- The test window decode is substituted for the tag hit signal.
- The DRAM sees test window accesses as cache hits.
- Writes do not go to DRAM regardless of WRMODE.
- The cache size bits and tag width bits are ignored, except to determine the window size, as indicated in the above table.
- The KEN* signal is not effected by this mode.

The cache mode bits must be set as follows:

ENCACHE = 0
INITCACHE = 0
FRZCDIR = 0
WRMODE=X

To size the data SRAM:

1. Set the test window to the desired location. If it is in the lower 1MB, set the cache size to 512K to get the full 512 Kbyte test window size.
2. Set the cache mode bits as indicated above.
3. Disable the 486 internal cache.
4. Set the data SRAM test mode ON.
5. Test to see if and where the SRAM repeats itself.

To test the RAM:



1. Set the test window to the desired location. If it is in the lower 1MB, set the cache size to 512K to get the full 512Kbyte test window size. A 256K window may be used by programming any other cache size if it is already known that the cache is less than 512K.
2. Set the cache mode bits as indicated above. T-49-17-01
3. Disable the 486 internal cache.
4. Set the data SRAM test mode ON.
5. Do a RAM test within the test window.

## Tag Testing

Tag testing is a little more complicated than data RAM testing. Like the data RAM test mode, tag testing involves the CPU reading and writing to the test window. The CPU cannot directly read or write the tag RAM, however. Registers at index 23 and 24 hold the data for tag testing.

On a CPU read from the test window, the tag data is written into index registers 23 and 24 at the end of T2. This is a 0WS read, and the CPU receives invalid data. The CPU must read indexes 23 and 24 to get the data.

On a CPU write cycle to the test window, data from index 23 and 24 gets written to the tag RAMs. The timing is identical to that of a "Write Hit Clean." It is either one or two wait states depending on the "Tag Write Timing" bit (index 20 bit 0). Before doing the write to the test window, the CPU writes indexes 23 and 24 with the desired data.

Note that there is one tag location for every 16 bytes of data. The tag RAMs get A4 as the lowest address bit. An access to any byte within the 16-byte, word, or D-word within the 16-byte tag line will read or write the tag RAM. Consecutive tag RAM locations can be accessed by incrementing the CPU address by 16 bytes (10H).

The cache mode bits must be set as indicated for the data SRAM test mode. The data RAMs are disabled during this time.

The following occurs when tag test mode is enabled:

- Reads within the test window latches the tag data at the end of T2. The cycle will be 0WS. The CPU reads garbage data. RDY* is returned, not BRDY*.
- Writes within the test window write the test data to the tag with the same timing as a Write Hit Dirty (one or two wait states).
- The tag comparator is disabled.

T-49-17-01

- No castouts occur.
- Memory accesses outside the test window go to DRAM, the AT bus, etc. and are not cached.



- No line fills occur.
- The test window decode is substituted for the tag hit signal.
- The DRAM sees test window accesses as cache hits.
- Writes will not go to DRAM regardless of WRMODE.
- The cache size bits and tag width bits are ignored, except to determine the window size, as indicated in the Table 7-7.
- The KEN* signal is not effected by this mode.

To size the tag RAM:

1. Set the test window to the desired location. If it is in the lower 1MB, set the cache size to 512K to get the full 512 Kbyte test window size.
2. Set the cache mode bits as indicated above.
3. Disable the 486 internal cache.
4. Set the tag test mode ON.
5. Test to see if and where the SRAM repeats itself, incrementing by 10H for each tag location, reading, and writing the tag as explained below.

To test the RAM:

1. Set the test window to the desired location. If it is in the lower 1MB, set the cache size to 512 K to get the full 512 Kbyte test window size. A 256K window can be used by programming any other cache size if it is already known that the cache is less than 512 K.
2. Set the cache mode bits as indicated above.
3. Disable the 486 internal cache to avoid getting fooled.
4. Set the tag test mode ON.
5. Do a RAM test within the test window, reading, and writing the tag as explained below.

To write a tag location:

1. Write the desired data to Indexes 23 and 24.
2. Write to the desired memory location in the test window.

To read a tag location: T-49-17-01

1. Read the desired memory location in the test window. Ignore the data.
2. Read the data from Indexes 23 and 24.

CHAPTER 8

T-49-17-01

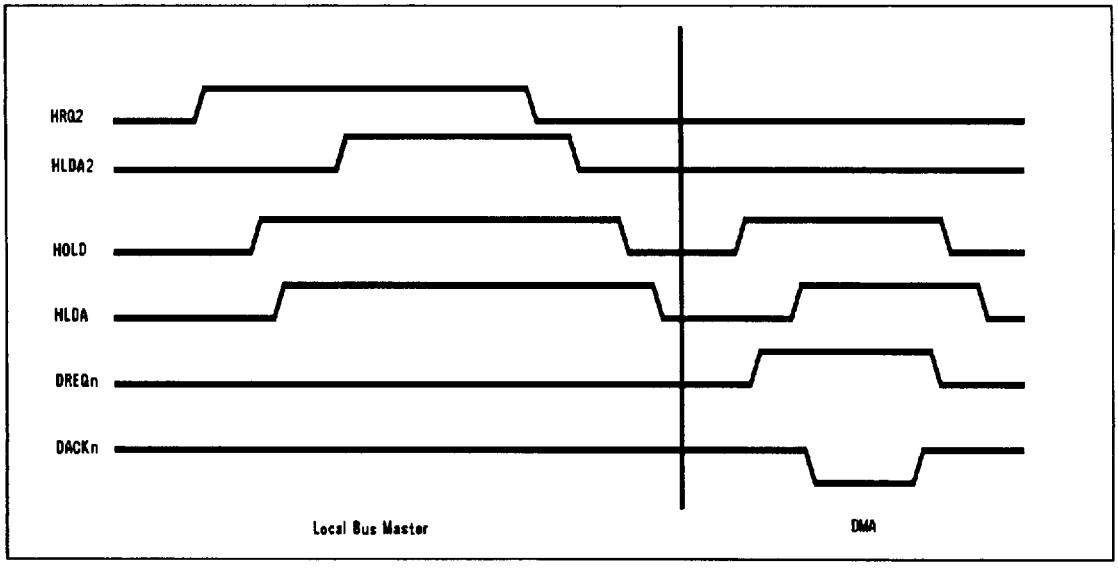
# System Bus Arbitration

The 4021 controls the access between the various masters and the system resources. The 4021 controls the bus by arbitrating between the masters requesting the system buses. It is responsible for giving control to the various bus masters at the appropriate time. The 4021 arbitrates between the following masters:

- CPU
- Refresh
- DMA controller (including ISA masters)
- Local bus master

The CPU is the default master. It gets control when no other master has the bus. The HOLD/HLDA signals operate differently from the other owners. The HOLD signal goes from the arbitration logic to the CPU, requesting control of the local bus. The CPU response to the HOLD is the HLDA signal which goes to the arbitration logic in the 4021 indicating that the CPU has given up the bus (see Figure 8-1).

***Figure 8-1.*** *Bus Arbitration*



The other bus masters request the bus. HRQn goes from the requesting master to the arbitration logic. The arbitration logic sends back HLDAn to give it control. The master takes HRQn low when it is finished indicating to the 4021 that it does not control the bus anymore (see Figure 8-1). Tables 8-1 and 8-2 show the arbitration signals for each master, as well at the current master settings.



*Table 8-1. Arbitration Signals for Each Master*

| Device | Input to Arbiter | Output from Arbiter |
|---|---|---|
| CPU | HLDA | HOLD |
| ISA classic refresh | HRQ0 (internal) | HLDA0 (internal) |
| DMA/ISA master | HRQ1 (internal) | HDLA1 (internal) |
| Local master | HRQ2 | HLDA2 |

*Table 8-2. Indication of Current Master*

| Master | HLDA | HLDA0 | HLDA1 | HLDA2* | MASTER* |
|---|---|---|---|---|---|
| CPU | 0 | 0 | 0 | 1 | 1 |
| ISA classic refresh | 1 | 1 | 0 | 1 | 1 |
| DMA | 1 | 0 | 1 | 1 | 1 |
| ISA master | 1 | 0 | 1 | 1 | 0 |
| Local master | 1 | 0 | 0 | 0 | 1 |
| No one (transitioning) | 1 | 0 | 0 | 1 | 1 |

HRQ0 and HLDA0 are internal to the 4021 and go between the refresh logic and arbitration logic. HRQ1 and HLDA1 are also internal to the 4021 and go between the DMA controllers and the arbitration logic. HRQ2 and HLDA2* are for a local bus master. They are muxed with chip selects and KBRESET*.

All inputs to the arbitration logic have equal priority, and they cannot preempt each other. If more than one device requests the bus, the first device to gain control must complete its operation before the second can begin. The CPU is left in hold between the first and second devices gaining control.

While HRQ1/HLDA1 are active, AT DMA or AT master activity can take place. The data buffers "turn around" to service a master which is on the AT bus, instead of the local bus.

HRQ2 and HLDA2* are for a local bus master. These pins are shared with CS0* and CS1*. Configuration bits determine the function. HRQ2 is forced low internally whenever the pin is not programmed for the HRQ2 function. At power-up, the pin is programmed as "No Function" and HRQ2 is forced low.

HLDA2* is active low for power-up reasons. This pin could be connected to either a local master or the chip select of an I/O device. At power-up, the HLDA2* function is selected, and the signal is high. If HLDA2* is connected to a chip select, being high at power-up, this signal will not accidently select the device. Figure 8-2 shows the DRQ and DACK interface.

T-49-17-01

The HOLD and CPURESET is arbitrated such that both are never active at once. If a request for one occurs while the other is in progress, the request is delayed. This arbitration will not cut into the 16 clock pulse width requirement of CPURESET. The 486 does not require this arbitration, but it will still be present.

## Pin Multiplexing

CHIPS & TECHNOLOGIES INC

Several sets of pins are multiplexed in the 4021 CHIPSet to allow a 184-pin package to be achieved. In most cases, all functions are available at once.

HRQ2 and HLDA2 are multiplexed with the programmable chip selects and KBRESET*. At this time, a choice must be made: local master chipselects, or KBRESET*. The majority of systems do not have a local master, so the chip selects are available. KBRESET* is only required in a system in which the 8042 KBRESET* function cannot be emulated by the 4021.

TC is time multiplexed with FASTAT. During CPU accesses, the pin is FASTAT. During DMA cycles, the pin is TC. Two F08 gates and an inverter externally create the separate signals. These gates are only needed if the FASTAT function is implemented on the board. If FASTAT is not implemented, the signals are not multiplexed; and the pin is dedicated to TC.

SMEMR* and SMEMW are multiplexed with VRAS* and XREQ*, which are used for the Windows Video function. When in Windows video mode, SMEMR* and SMEMW* are generated externally.

MA10 and DSF are multiplexed. For 4Mx1 or 1Mx4 DRAMs, this is MA10. For VRAMs this is DSF. This is a special function select output to the VRAMs. This indicates to the VRAMs whether to do a full transfer cycle or a split transfer cycle.

MA11 and TROE* are multiplexed. For 4Mx4 DRAMs with 12/10 addressing, this is MA11. For VRAMs this is TROE*, which is used for transferring a row of data to the shift register. This pin must be high for refresh cycles.

0WS* is time multiplexed with LOCAL*. The pin is LOCAL at the binning of each T-state. It is 0WS* during AT bus accesses. During DMA and AT master cycles, it is not used at all. Configuration bits allow the LOCAL and 0WS functions to be disabled independently in the chip which would dedicate the pin to the other function (or dedicate it to nothing at all).

The DREQs, DACKs*, Turbo, and FLUSH* are multiplexed to the following six pins:



- SEL 2:0 Multiplex and demultiplex selects
- DACKEN* DACK enable for the 138
- MUX 1:0 Multiplexer inputs

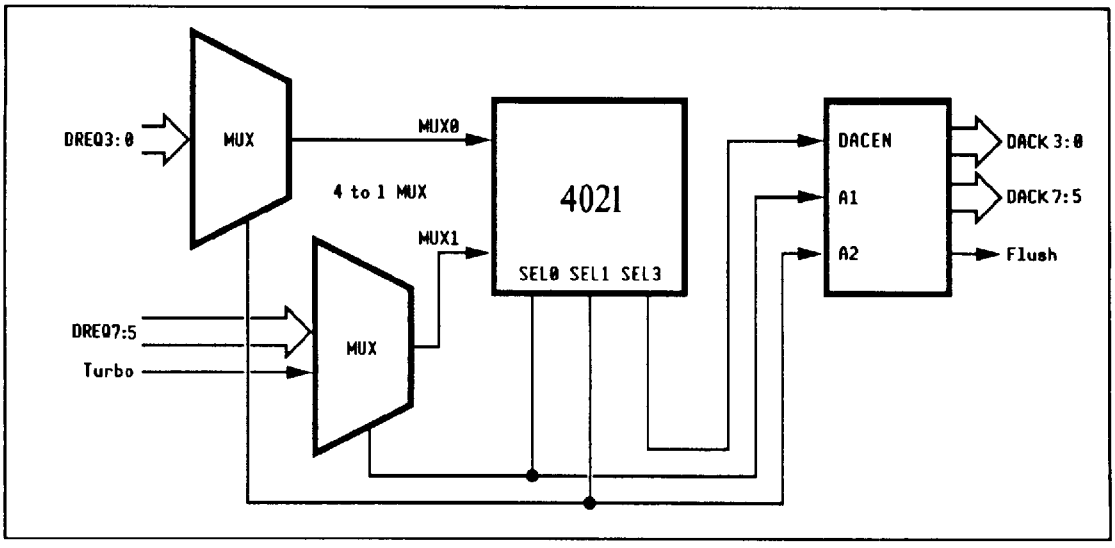


An F153 dual 4-1 multiplex receives the seven DREQs and TURBO pins and channels them down to two pins (MUX 1:0). SEL0 and SEL1 are used to select the multiplex. These pins are scanned at 14.3 MHz. The data is placed in eight latches as it is scanned. The outputs of the latches go to the IPC and TURBO logic.

When the IPC generates a DACK*, the SEL pins immediately encode the proper DACK* number, and DACKEN* goes low after a setup time. The SEL lines are frozen until the DACK goes away. Externally, an F138 decodes the SEL 2:0 and DACK* lines, and generates the peoper DACK*. Duirng this time, the DREQs are not being scanned, but the DREQ which corresponds to the active DACK is continuously sampled and sent directly to the IPC. A DREQ is more time sensitive when the channel is active than at other times.

Since there are only seven DACK signals and eight outputs of the F138, the extra one can be used for FLUSH*. When a flush is requested, a flip flop is set. While scanning the DREQs, the next time the count gets to four, it stops for a few clocks while DACKEN* is driven low then back high to generate a FLUSH* pulse. This signal is asynchronous to the 486, and need only be present for one clock. Note that if a DMA or AT master has the bus, the FLUSH* is delayed until after the DMA or AT master gives up the bus. Refer to *Chapter 13: Configuration Registers*, Index Register 27 for more information.

***Figure 8-2.*** *DRQ and DACK System Interface*



## Performance Control

Performance control refers to slowing the CPU down. A 486 can execute code too rapidly for some speed sensitive software. The 4021 makes no attempt to reduce speed via the CPU clock since this requires the CPU to be in reset for 1M second. Instead, it puts the CPU in hold for a programmable percentage of time. This is implemented as follows:

- On every refresh period, the CPU is put in HOLD for a programmable period of time. The FLUSH* pin is also pulled low during this time to make sure the CPU does not continue to execute from its internal cache. The length of the HOLD pulse is programable to provide the desired degradation in performance. The intent is to match the speed of an 8MHz AT in order to allow some games and copy protection programs to work properly. The length of the HOLD pulse has to be varied based on the system speed, to maintain an 8Mhz system performance.

After the HOLD length is set, the slow mode may be enabled and disabled by enabling the 'go slow' chipset register or pressing the DeTurbo button. This is an OR function, either method slows the system down (they must all be disabled for full speed).

When classic AT refresh is used, the HOLD, which is generated for the refresh, is used and extended for the programmed length. When AT hidden refresh or disabled AT refresh is used, HOLD is generated specifically for performance control.

The FLUSH* and HOLD functions can be enabled separately, but they should be used together to achieve more uniform performance degration to system speed ratio.

The DeTurbo button signal is time multiplexed with the DREQs. The Turbo Switch connects to the external 153. FLUSH* is time multiplexed with the DACK* signals. The CPU FLUSH* pin connects to the external 138.

## Refresh

The 4021 provides three modes of refresh: classic, hidden, and disabled. Classic refresh uses the traditional method of putting the CPU on hold and then performing the refresh of the AT bus and local DRAM. Hidden refresh does not put the CPU on hold but performs the refresh while the CPU continues to operate out of the second level cache. Disabled refresh can be used when AT cards do not require refresh (the usual case). Local DRAM is still refreshed.

T-49-17-01

## Classic AT Refresh

The refresh logic requests the bus from the arbitration unit by taking HRQ0 high. When it gets the bus (HLDA0 high), AT and DRAM refresh are both performed. The CPU is on hold, so there is a performance penalty. The address is supplied by the 4021 on XA0:1 and A2:7.

## AT Hidden Refresh

The refresh request goes to the DRAM and AT refresh sections. The DRAM controller performs a refresh between other DRAM cycles, using as little time as possible to perform the refresh. The AT refresh is done when the AT state machine is not busy. The CPU is not put on HOLD. This requires an external LS590 or 393 and 244. AT hidden refresh does not happen when HLDA1 is active. If a refresh request occurs when HLDA1 is active or is about to go active, the AT hidden refresh is delayed until HLDA1 goes low again. Likewise, HLDA1 is delayed until an AT hidden refresh is finished. If the CPU requires access to the AT bus while hidden refresh is in progress, wait states are inserted until the refresh is complete.

### Disabled AT Refresh

Disabled AT refresh is the same as AT hidden refresh, except that no AT refresh cycle is done on the AT bus. This does not require the LS590. It is common for a system to have no boards that require AT refresh.

Master refresh (a refresh cycle initiated by an ISA master) is done the same way in all modes. Both the ISA bus and the DRAM are refreshed together. The address for master refresh is supplied as indicated in the following table.

***Table 8-3.*** *Address Source for Master Refresh*

| AT Refresh Mode | Address Source |
|---|---|
| Disabled | XA0-1, A2-7 |
| Classic | XA0-1, A2-7 |
| Hidden | External LS590 |

For a master refresh, the ISA master pulls the REFRESH* pin low. The 4021 detects this and drives the refresh address (unless hidden refresh is enabled; then the external LS590 drives it). It also drives MEMR* and SMEMR* low for two BUSCLKs (extended if IOCHRDY is low). The master takes REFRESH* high one BUSCLK after MEMR* goes high.



*Note:* The 4021 performs a master refresh cycle even when AT refresh is disabled. The ISA master waits for MEMR* to go low then high before releasing REFRESH*.

T-49-17-01

Disabled refresh has the highest performance. Hidden refresh is almost as high as disabled, and in the common benchmarks is probably exactly the same. Classic refresh has a performance penalty, but not as much in a 486 as it is in a 386 system since the 486 cache masks a lot of the performance loss.

T-49-17-01

# The ISA Bus

The AT bus controller is designed to perform like an IBM AT. The following enhancements have been added:

- Different CPU and bus clocks (transparent to devices on the bus) are allowed.
- Default wait states and command delays (though not compatible with the AT) are allowed to tailor the system for custom applications and provide a fall back for unforeseen difficulties.

## General ISA Bus Support

An AT bus cycle starts at the end of the first or second CPU T2 state under the following condtions: (see Figure 9-1)

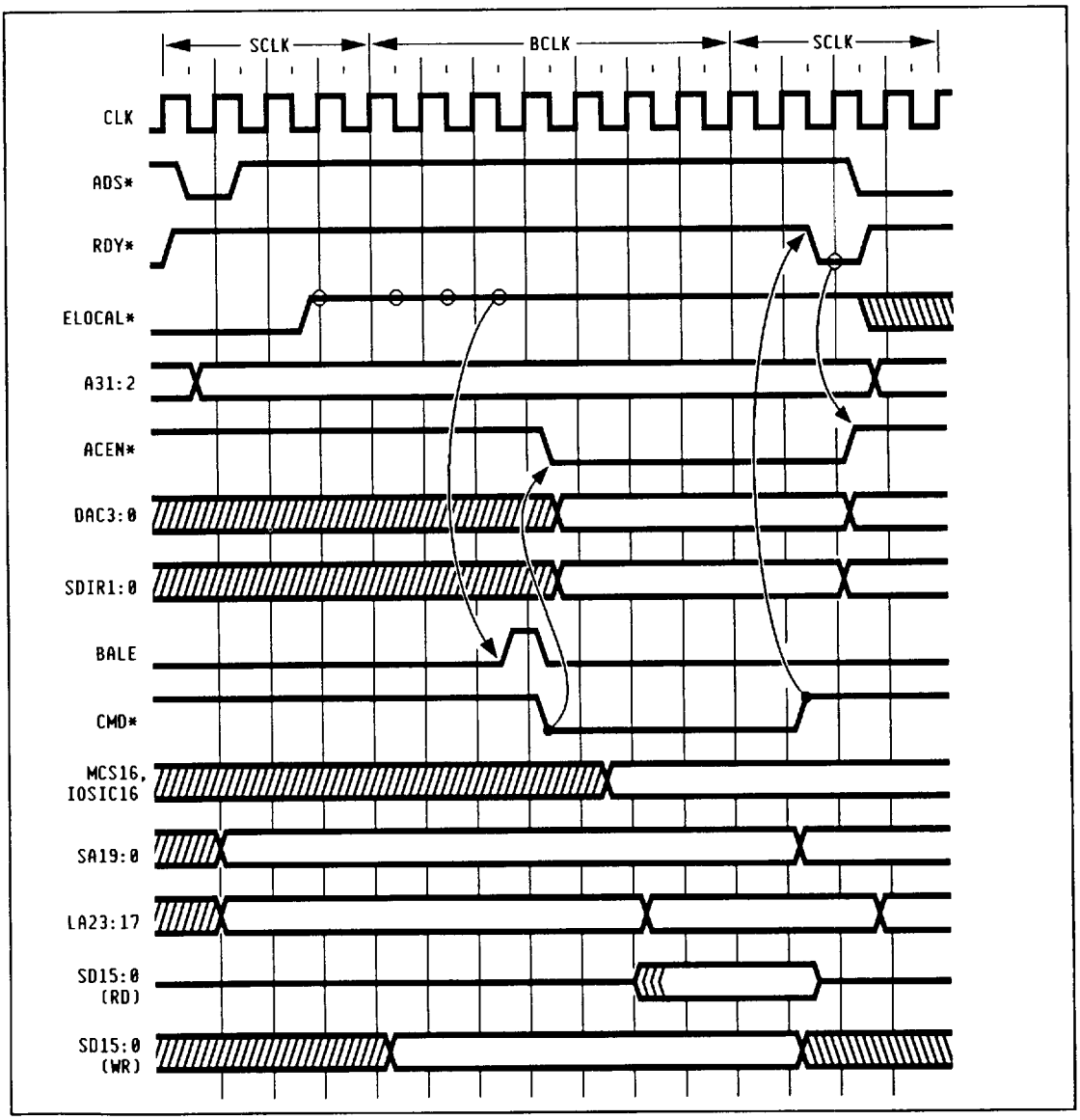
- When LOCAL* is sampled in its inactive state (this is an internal DRAM decode)
- When the external LOCAL* pin is inactive
- When the LOCAL* signal is not forced internally from the video or other decode.

Normally, the end of the first T2 is used by the LOCAL* sampling. It can be delayed until the second T2 to allow local peripherals to have more decode time at higher speeds.

At the end of the first CPU T2 state, the internal AT enable is sampled on the first falling edge of BUSCLK. AT-TS will begin on the next rising edge. At the end of the final AT-TC, the internal AT active signal will be sampled on the first rising edge of the CPU clock, and READY* will be generated one CPU clock later.

*Figure 9-1. DMA 8-bit CPU for 16-bit AT Cycle*

T-49-17-01




BALE is generated in the middle of AT-TS (the second BUSCLK falling edge from the end of T2). BALE goes inactive on the next rising edge of BUSCLK. It is, therefore, 1/2 BUSCLK cycle long. This BUSCLK rising edge marks the beginning of the first AT-TC state, or what is known as the command phase. The command phase is where the commands (MEMR*, MEMW*, IOR*, IOW* and INTA*) are generated. Command delays also begin at this point. Command delays hold off the generation of the command signals in 1/2 BUSCLK increments. That is, a cycle

with one command delay generates its command in the middle of the first AT-TC. A cycle with two command delays generates the command at the end of the first AT-TC, etc.



Wait states are measured in whole BUSCLK increments. A one wait state cycle has a command that is active for one BUSCLK (the first one that you do not count) plus one more BUSCLK (the wait state), or 2 BUSCLK cycles. Actually, the command will be two BUSCLK cycles long minus the command delays.

After the programmed number of wait states have been counted out, the cycle ends. Wait states can be added by a slave device deasserting IOCHRDY. On the AT-TC cycle that would be the end (because programmed wait states have run out), IOCHRDY is sampled. If it is found to be inactive (a peripheral card has driven it low, which is the inactive state), one more AT-TC cycle is inserted and the AT cycle does not terminate. Again, IOCHRDY is sampled, and this process repeats (indefinitely, until IOCHRDY is active). IOCHRDY is sampled at the first 1/4 of the AT-TC cycle. This is the first falling edge of BCLK in the cycle.

T-49-17-01

## I/O Accesses to Internal 4021

The 4021 data path is through the lower DRAM address lines, MA0:7. A bus driver, which is controlled by XDEN* drives the data between the MA and XD busses during I/O cycles to I/O internal to the 4021.

Because of this, one of these I/O cycles cannot occur during a DRAM buffered write or VRAM transfer cycle. To prevent this from happening, an ISA cycle cannot begin until the write buffer is empty and any pending VRAM transfer cycle is completed. Likewise, a VRAM transfer cycle cannot begin during an ISA cycle. DRAM refresh may occur since MA0:7 are not used for refresh.

XDEN* goes active for:

- I/O reads from ports 61 and 92.
- I/O reads from ports 23 and 27 when the respective index register points to an internal register.
- I/O reads from an IPC ports: 00-0F, 20-21, 40-43, 71, 80-8F, A0-A1, and C0-DF.
- Interrupt acknowledge cycles.
- All I/O writes that become AT bus cycles.

Ports 60, 64, and 70 are write only in the 4021, and reads will NOT activate XDEN*. 387 cycles and cycles claimed by the local bus do not activate XDEN* since they do not become ISA cycles (see Figure 9-2 and 9-3).

*Figure 9-2. CHIP Register Read to DRAM Read*

T-49-17-01

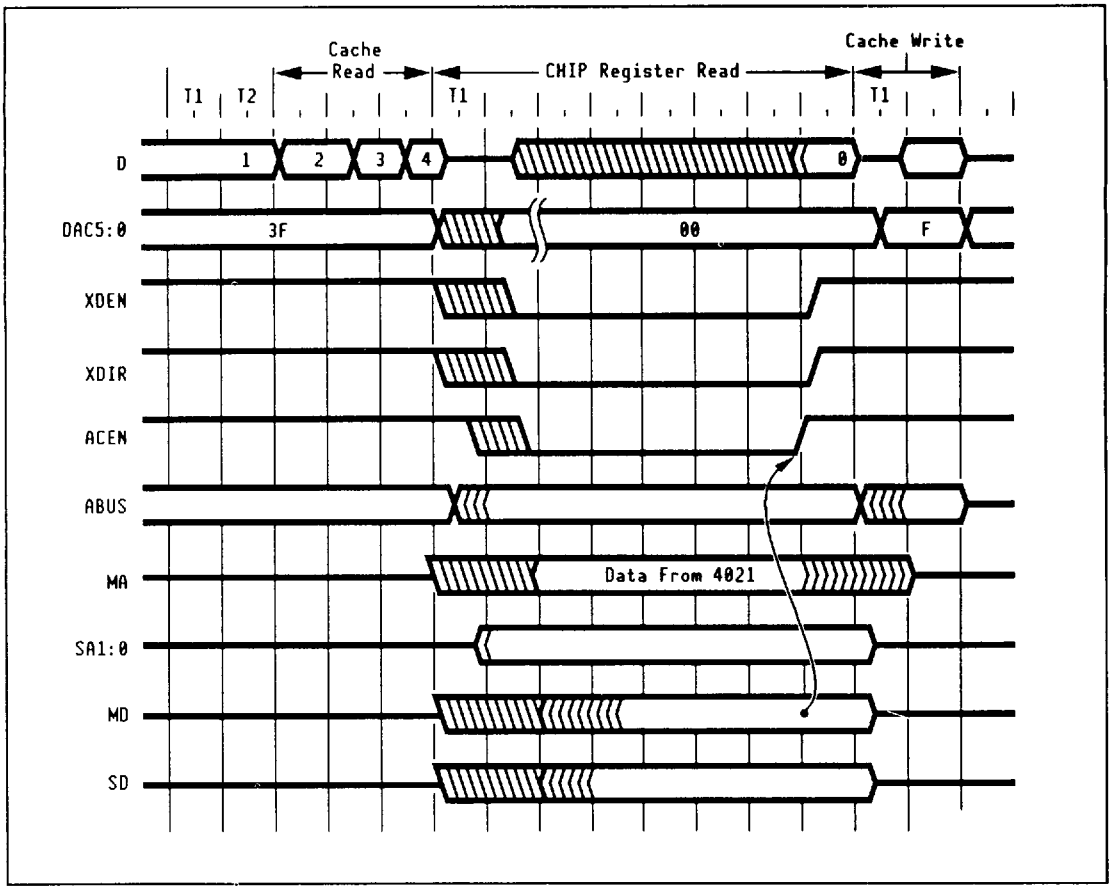
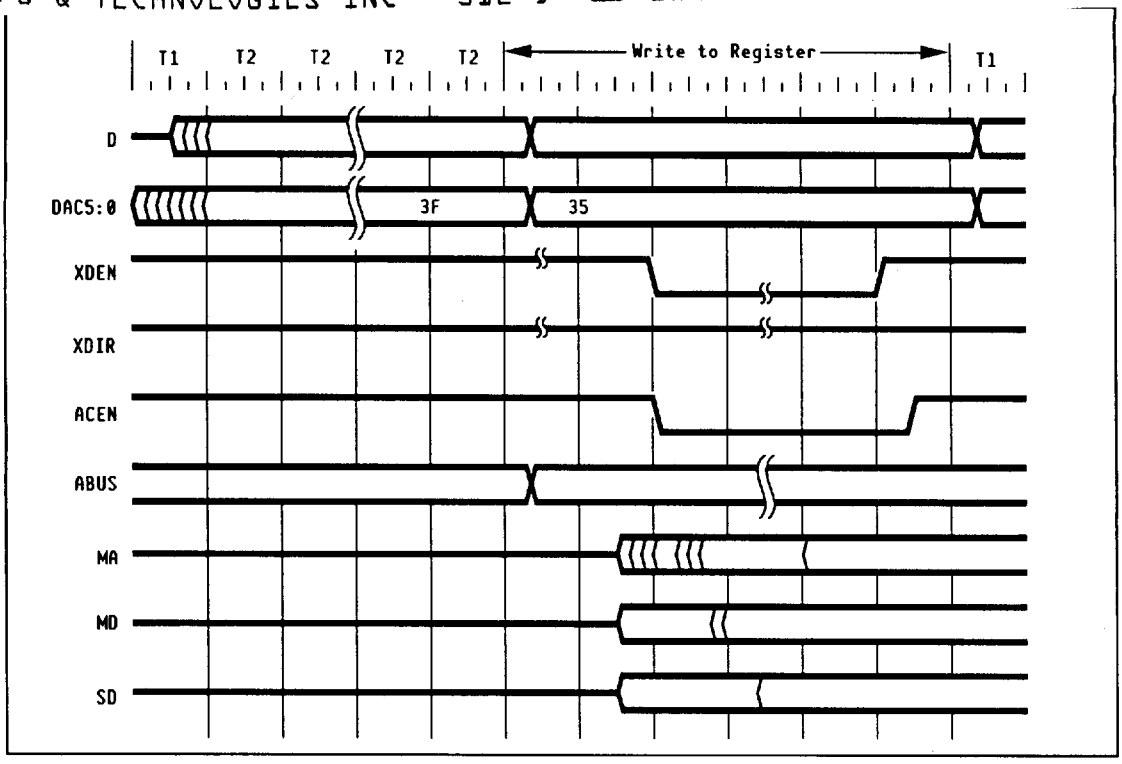


*Figure 9-3. CHIP Register Write*

T-49-17-01






XDEN* has the following timing:

- For external I/O reads XDEN* goes low with IOR*, and goes high on the clock in which RDY* is driven low (slightly after IOR* goes back high).
- For interrupt acknowledge cycles, XDEN* goes low with INTA* and goes high on the clock in which RDY* is driven low.
- For I/O write cycles, XDEN* goes low at the end of the first T2 (assuming the AT cycle is not delayed by a DRAM buffered write or VRAM transfer cycle) and goes high for 2 to 3 SCLKs (depending on synchronization) after IOW* goes high.

The early timing for write cycles is necessary for writes to the keyboard controller. The 8042 reset and GATEA20 emulation logic uses the value of the data written to these ports to determine whether the IOW* pulse should be blocked or not.

## XDIR Decoding

T-49-17-01

XDIR drives the direction control between SD0-7 and XD0-7. It is normally high, which drives data from the SD to XD bus, and goes low for read cycle from the XD bus. It will go low with IOR* or MEMR* and go high on the clock where RDY* is generated. XDIR also controls the direction of the XD to MA buffer. XDIR goes active for the following ranges:

- I/O READS 0000-00FF (up through A15 decoded)
- Interrupt acknowledge cycles (all)
- Memory READS — When ROMCS* is active; the addresses are programmable in blocks from C0000-FFFFF

# Configuration Registers Affecting the ISA Bus

Table 9-1 shows the configuration registers that can be used to control the ISA bus.

***Table 9-1.*** *Configuration Registers Affecting the ISA Bus*

| Register | Description |
|---|---|
| 05 | ISA bus command delays. |
| 06 | ISA wait states & address hold. |
| 07 | ISA bus clock selection. |
| 19-1A | DRAM shadow RAM control. |
| 1B | ROMCS* control. |
| 28-2F | I/O decodes #0 and #1 (may be programmed for FASTAT). |
| 30-35 | Memory Decode #0 and #1 (may be programmed for FASTAT). |
| 36 | FASTAT fixed decode control. |

T-49-17-01

# DMA/AT Master Access

CHIPS & TECHNOLOGIES INC

The DMA controller and AT masters conduct memory operations by pulling MEMR* or MEMW* low. The memory controller must synchronize these to the CPU clock, and perform the requested memory operation. Since this is a write-back cache, data must be read from the cache if it is a cache hit. If it is a miss, the data is read from the DRAM. For a memory write cycle, the data is always written to the DRAM. It is also written to the SRAM if it is a cache hit. The dirty bit is not updated in the case of a clean line, since the line can still be considered "clean."

An EADS* is generated for all DMA and AT master memory write cycles to invalidate the 486 internal cache. The 4021 also drives A31 and A26:23 low when HLDA is high since the DMA controller and AT master only provide A23:0. This is required for invalidating the 486 or 38605 cache. A30:27 are driven low by an external 244.

On read cycles CAS must remain low until MEMR* goes high. This is because the read latch is not in the circuit for these reads. There is an F245 between the DRAMs and the AT bus (the data must also pass through the 4025 during some reads). On read cycles, the controller has about 180nS from the time MEMR* goes low until the data is required on the AT bus.

The timing diagrams shown in Figures 9-4 through 9-9 identify the DRAM timing for DMA and master cycles.

---

***Note:*** RAS* goes low asynchronously with MEMR* or MEMW*. Clock synchronizing is done before the mux is switched.

---

T-49-17-01

*Figure 9-4. Master Write to Local Bus*



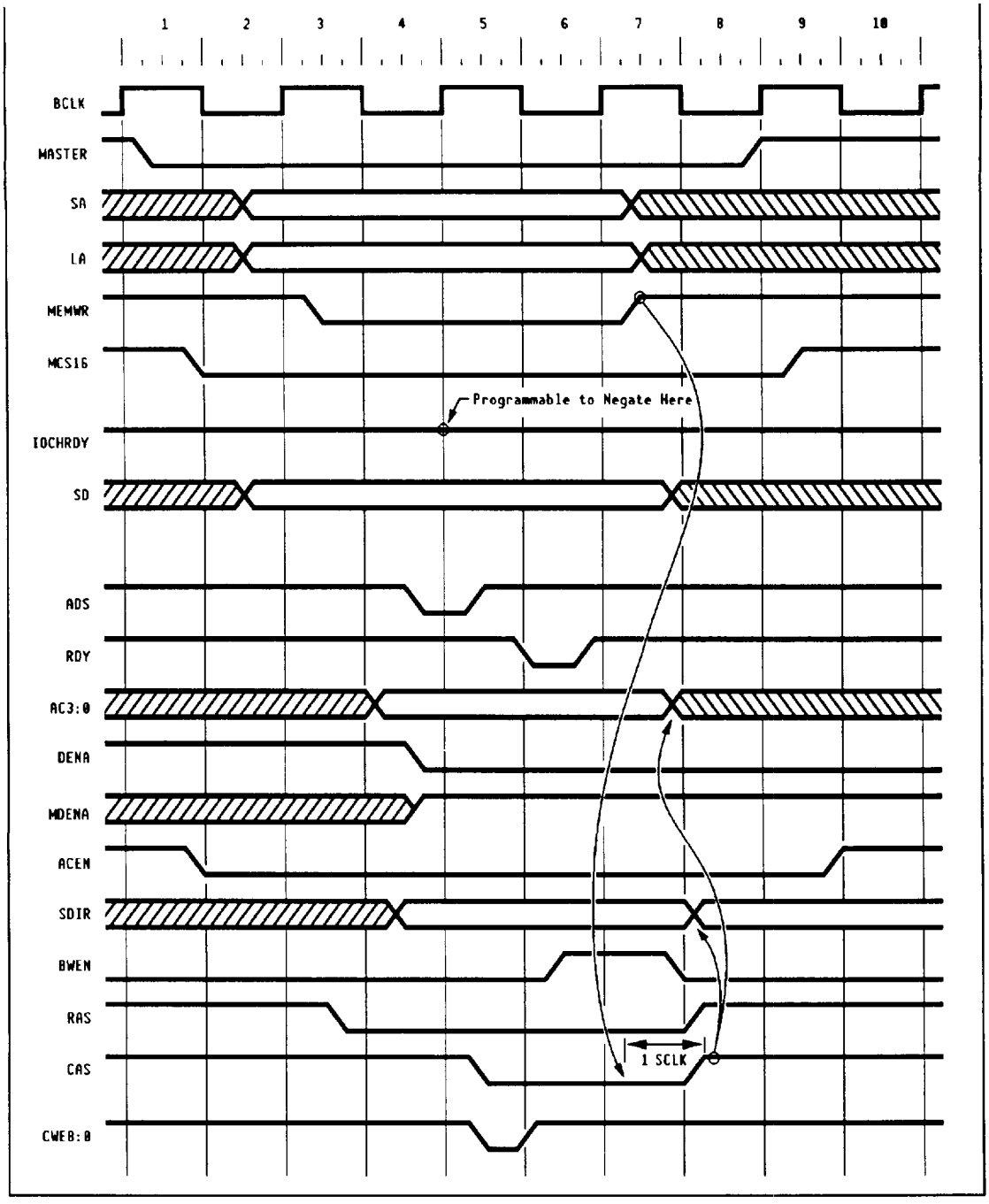


***Figure 9-5.*** *Master Read from Local Bus*

T-49-17-01

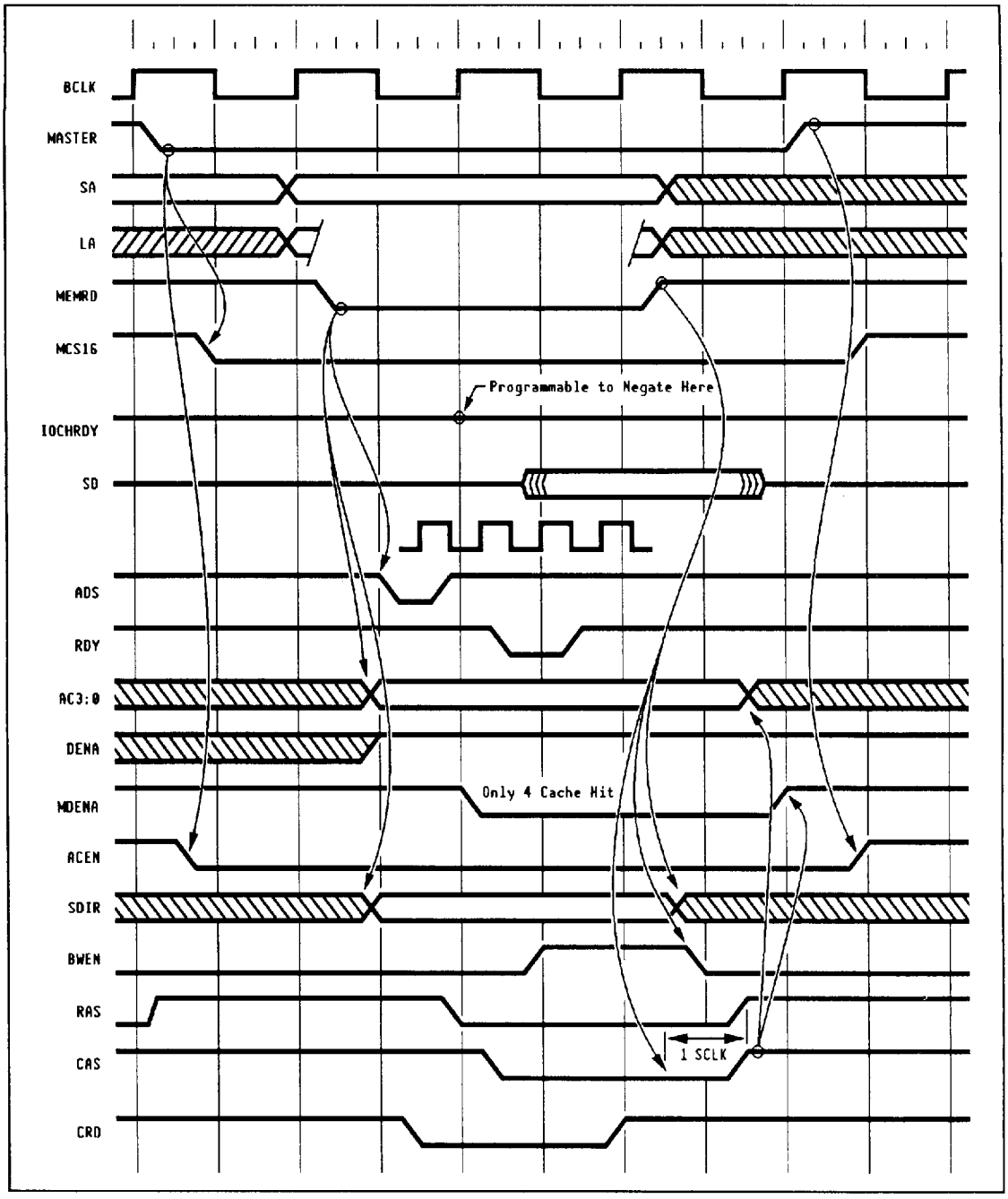


*Figure 9-6. 8-bit DMA Read from System Memory*

T-49-17-01

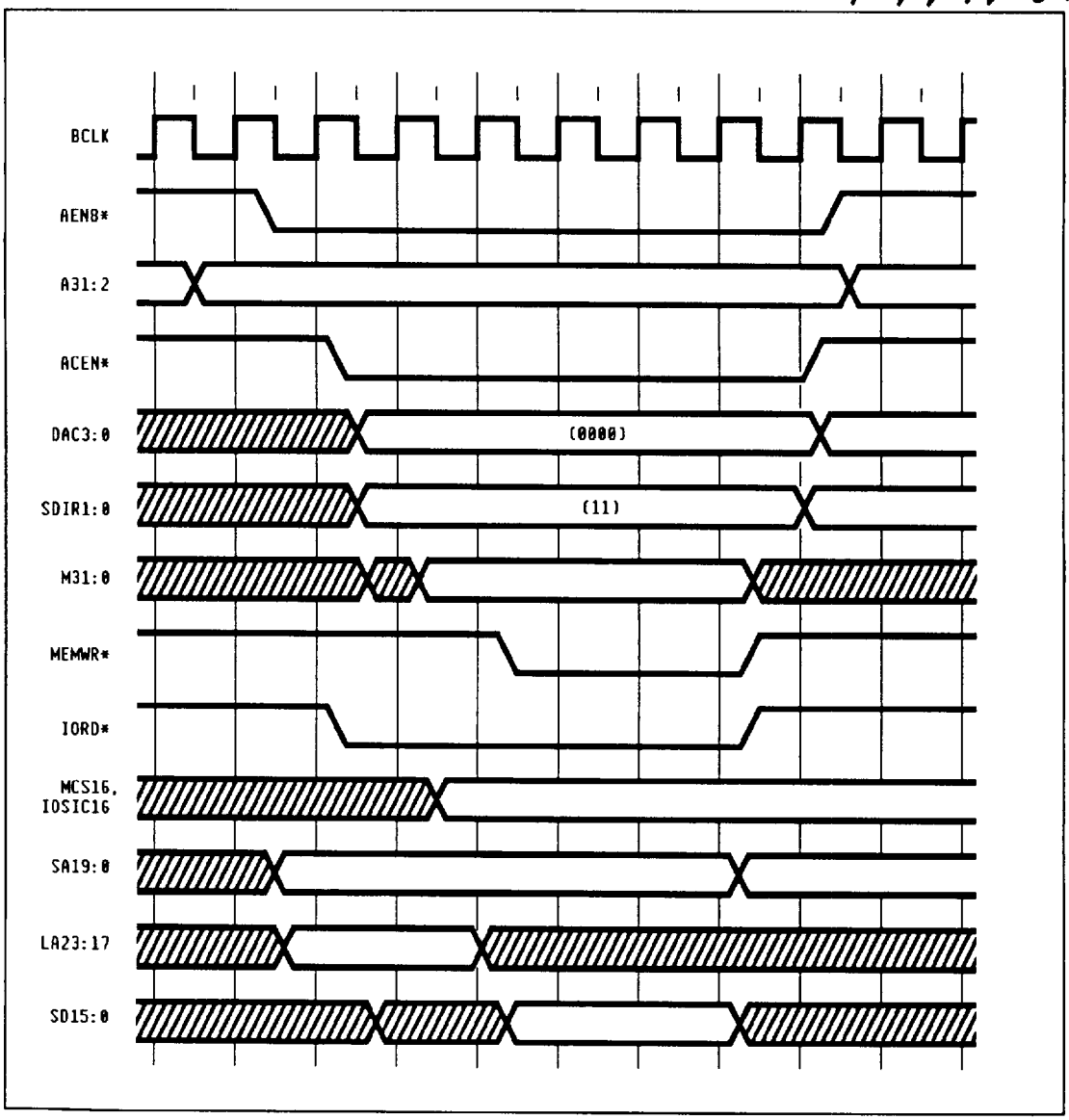


*Figure 9-7. 8-bit DMA Write to System Memory*

T-49-17-01



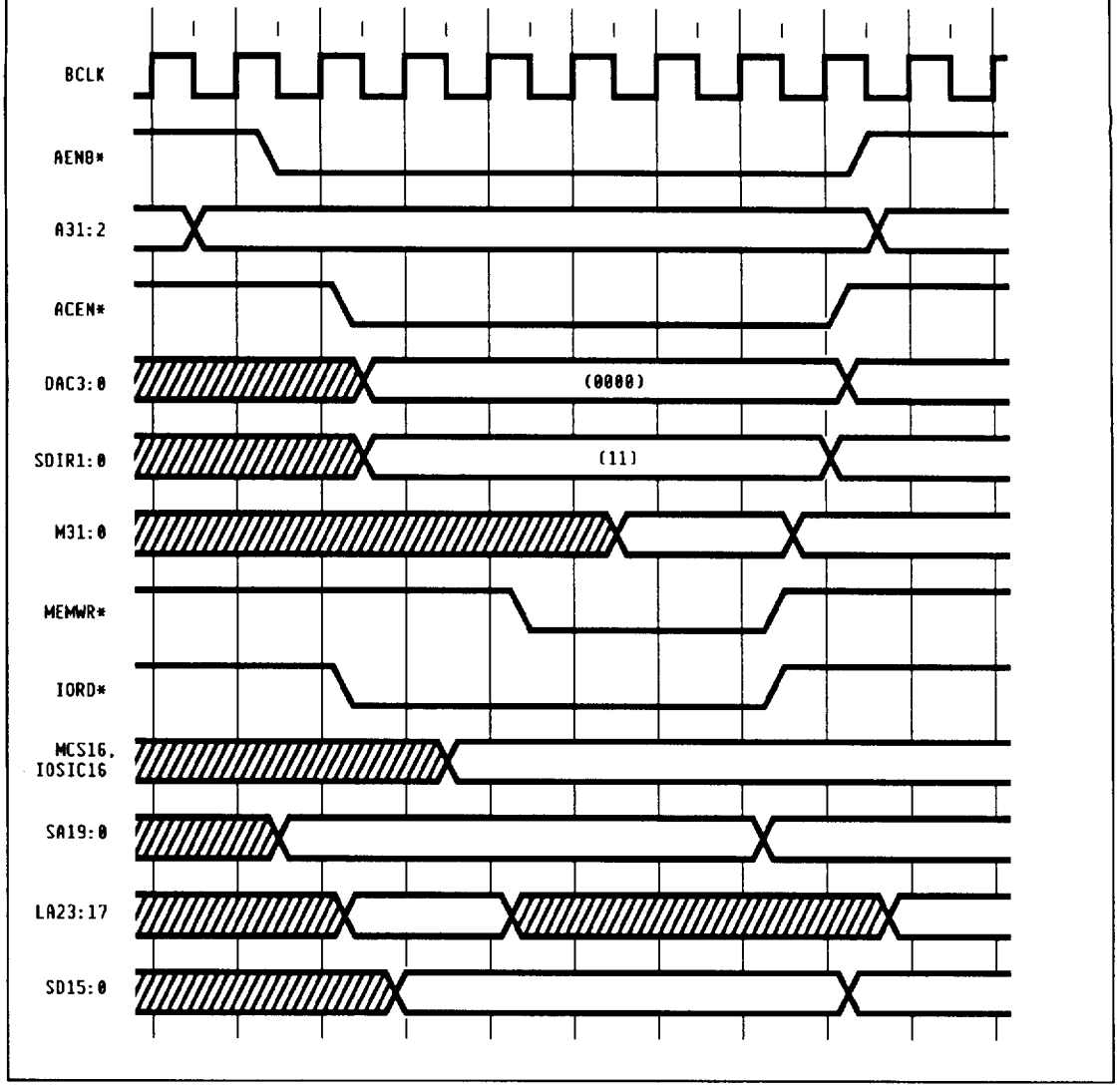


*Figure 9-8. 8-bit DMA for 16-bit AT Cycle*

T-49-17-01

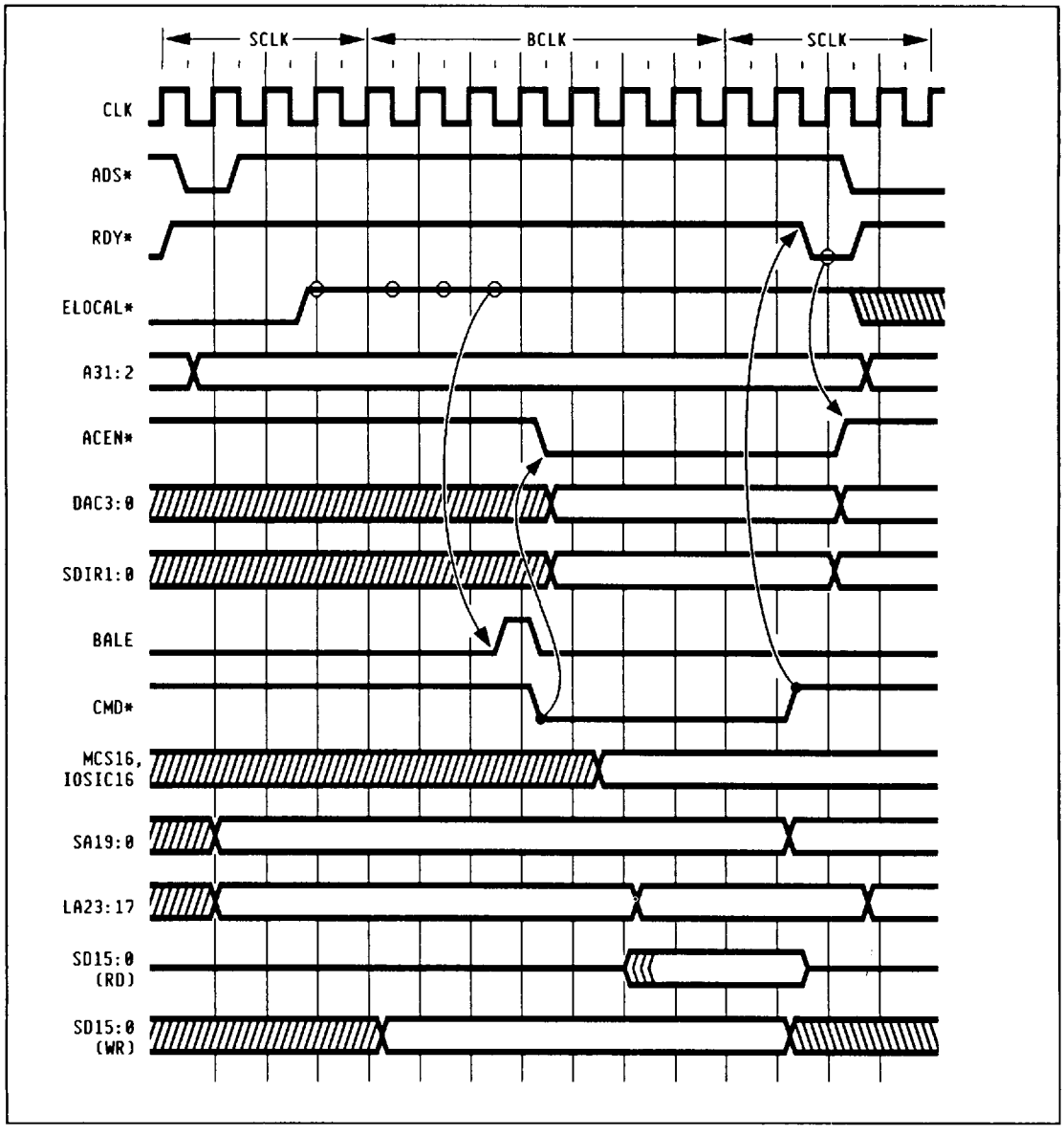
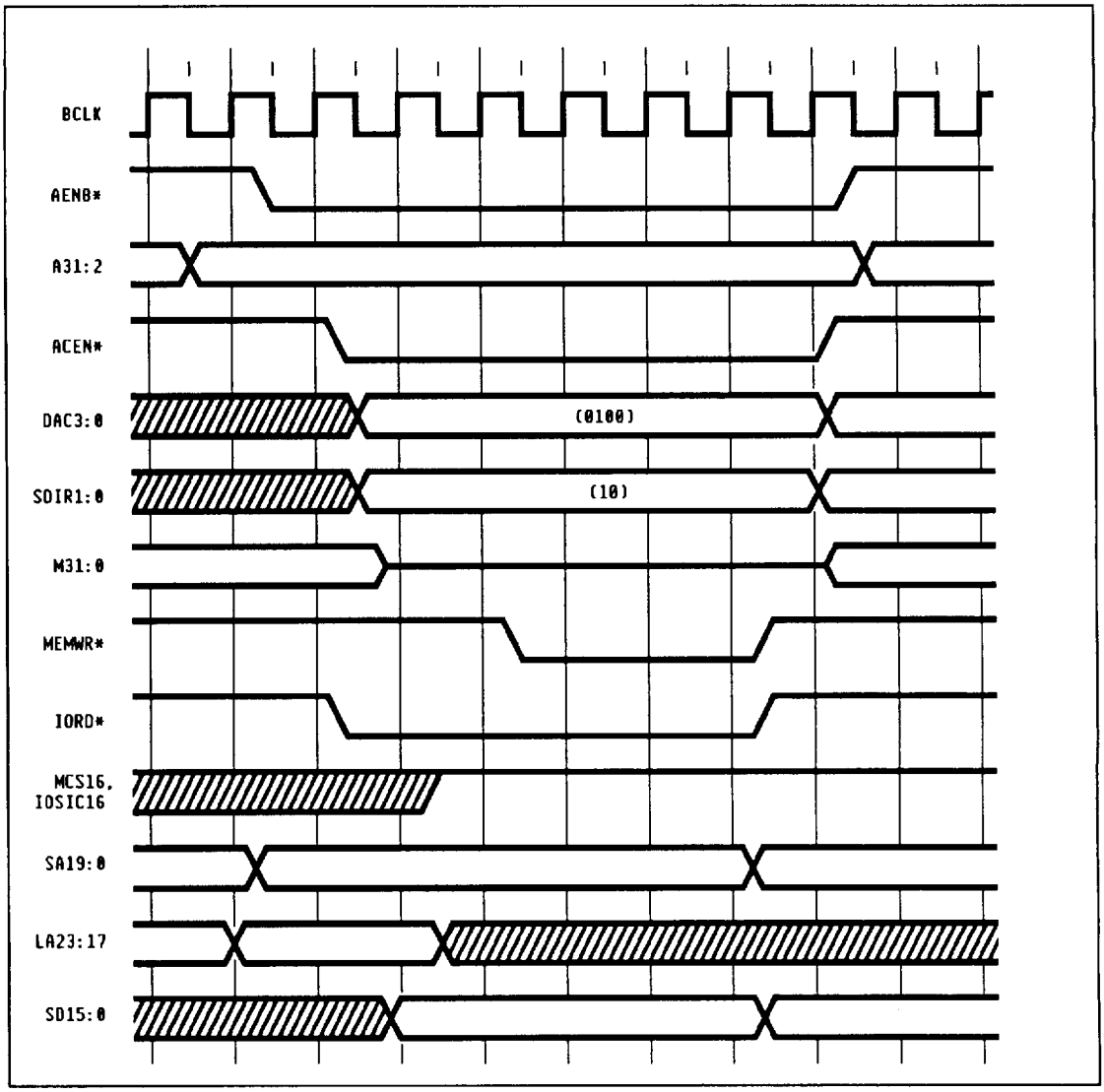


*Figure 9-9. 8-bit DMA Write to 8-bit AT Memory*

T-49-17-01



# Coprocessor Support

T-49-17-01

Table 9-2 lists the coprocessors supported for each CPU.



***Table 9-2.*** *Coprocessors Supported for each CPU*

| CPU | Supported Coprocessors |
|---|---|
| 486sx | 487sx, Weitek 4167 |
| 486 | Internal, Weitek 4167 |
| 386, Super386 | 387, Weitek 3167 |

## 486 and 486SX with External Numeric Processors

The 486SX does not contain an internal numeric processor. The upgrade for the 486SX is to add or replace it with a 487SX. In any case, the 4021 handles the coprocessor exactly the same way.

The 4021 contains exception handling logic for the 486, which generates IRQ13 internally and FERR* to the 486.

## 387 and Super387 Numeric Processors

The following support is done by the 4021 for the 387 CPU:

- Ready Generation on 387 I/O cycles (addresses 800000F8-800000FF).
- Coprocessor ERROR* handling.

The READY generation is handled by the 4021. There are two options:

- Generate a one wait state READY on all 387 I/O cycles.
- Don't generate READY at all, allowing external logic to do it.

The 387 has a READYO* pin which generates a one wait state READY* for 387 reads, and in most cases a 0WS READY for writes. Writes are one wait state for certain instructions. The READYO* pin is not tristate, which poses a problem in most systems. The 4021 gets around this by optionally generating a one wait state READY for all 387 cycles. External logic can optionally be used to generate a READY. Coprocessor I/O cycles are detected by the 4021 by I/O cycles with A31 high.

The ERROR* handling is done by an external PAL. The PAL does the following:

1. Checks for the presence of the 387 and informs the 386 of this through the ERROR* pin on all CPU resets (power-up and soft).
2. Generates the IRQ13 and 386 BUSY* signals based on BUSY* and ERROR* from the 387 and INTCLR from the 4021.
3. Toggles BUSY* to the CPU occasionally in the event that a 387 is not present (this avoids CPU hangs with some software while checking for the existence of the coprocessor).
4. Prevents the CPU from doing a self test on soft resets.
5. Combines the Weitek interrupt with the 387 interrupt on IRQ13. The Weitek interrupt is masked if the Weitek 3167 is not present, which is detected through a hardware pin.

T-49-17-01

The 4021 generates the INTCLR signal, an active high write strobe for I/O ports F0 and F1. INTCLR is active for the duration of the IOW* strobe for these addresses. When the 386 or Super386 is selected as the CPU type, this signal is provided on the FERR* pin.

## Weitek 3167 and 4167 Numeric Processors

The Weitek 4167 is supported with full 0WS operation except for really high clock speeds. It supports one wait state operation.

The speed related problem is generating KEN*. It must be high for Weitek cycles, and must meet a 5nS setup time to the end of T1 for 0WS operation. To do this, A31 high will force KEN* high asynchronously in 0WS mode. In 1WS mode, A31 is clocked at the end of T1 and gets to the 486 by the end of the first T2.

Externally, RDYOUT* is driven onto the 486 RDY* pin with a tristate driver enabled with MCS* from the 4167. For 0WS mode, TCR* is tied high. For 1WS mode, it is tied to PCD from the 486 (allows newer software, which makes the 4167 area non-cacheable internal to the 486 to run in 0WS).

IRQ13 from the 4167 is tied to the IRQ13 pin. This pin is internally masked off by a configuration bit, then ORed with the 486 coprocessor interrupt circuit before going to the interrupt controller. The configuration bit masks off the external IRQ13 when no Weitek is present.

The presence or absence of the 4167 is determined by software. If it is not present a ready time-out, it will be generated by the 4021.

The 3167 is handled in a similar manner as the 4167. The 3167 READY* is tristate and can be connected directly to the RDY* from the 4021. The AF32* pin is not used since the 4021 internally decodes Weitek cycles. The 3167 interrupt is connected to the 387 PAL instead of the 4021 directly. When the 386 or Super386 CPU is selected, the IRQ13 pin is not masked internally by the Weitek present bit. It is used for both the 387 and Weitek interrupt. The masking is done in the PAL.

CHAPTER 10

# Address Mapping

T-49-17-01

The address mapping is separated into I/O address mapping and the memory mapping.

## I/O Address Mapping

All I/O accesses go to the ISA bus except for cycles claimed by local bus slaves (by pulling the LOCAL* pin low or by an internal local decode). All accesses to the 4021 internal I/O and configuration registers are done via the ISA bus.

The 4021 has the following internal I/O decodes:

| | |
|---|---|
| **IPC functions** | 00-0F, 20-21, 40-43, 70-71, 80-8F, A0-A1, and C0-DF |
| **Config Reg Windows** | 22, 23, 26, & 27 |
| **Port B** | 61 |
| **NMI Mask** | 70 |
| **CPU fast reset, GATEA20** | 92 |
| **Coprocessor ERROR*** | F0-F1 |
| **Coprocessor** | 800000F8-800000FF |
| **8042 chip select** | Ports 60 & 64 |
| **Hard Disk Data port** | 1F0 |
| **Video I/O: MDA** | 3B0-3BB and 3BF |
| **Video I/O: EGA** | 03C0-03C5, 03CA-03CF |
| **Video I/O: CGA** | 03D0-03DF |
| **Palette DAC** | 03C6-03C9 |
| **Two Programmable decodes** | |

Internal Decodes are used for the following purposes:

- Select I/O ports which are internal to the 4021. There are many I/O ports (see *Chapter 13: Configuration Registers* for all the I/O Ports).
- Generate 8042CS*/ROMCS*. The 8042 chip select and ROM chip select are the same pin (see the section titled ROMCS*, at the end of this chapter, for more details).
- Generate Programmable Chip Selects. There are two programmable chip selects, discussed below.
- Select I/O areas for FASTAT (both fast clock and 0WS I/O). There are bits for video I/O, hard disk data port, plus the two programmable ranges (see *Chapter 12: Advanced Device Interface* for information about the FASTAT interface).
- Select I/O areas to appear on the local bus. There are bits for the video I/O (three independent ranges), and the two programmable ranges (see *Chapter 12: Advanced Device Interface* for more information about the Local Bus Support, and the section titled Programmable I/O Ranges later in this chapter).
- Generate RDY* for the 387 I/O area. A one wait state READY* is optionally generated.

Refer to Figure 10-1 for the I/O and configuration map of the CHIPSet.

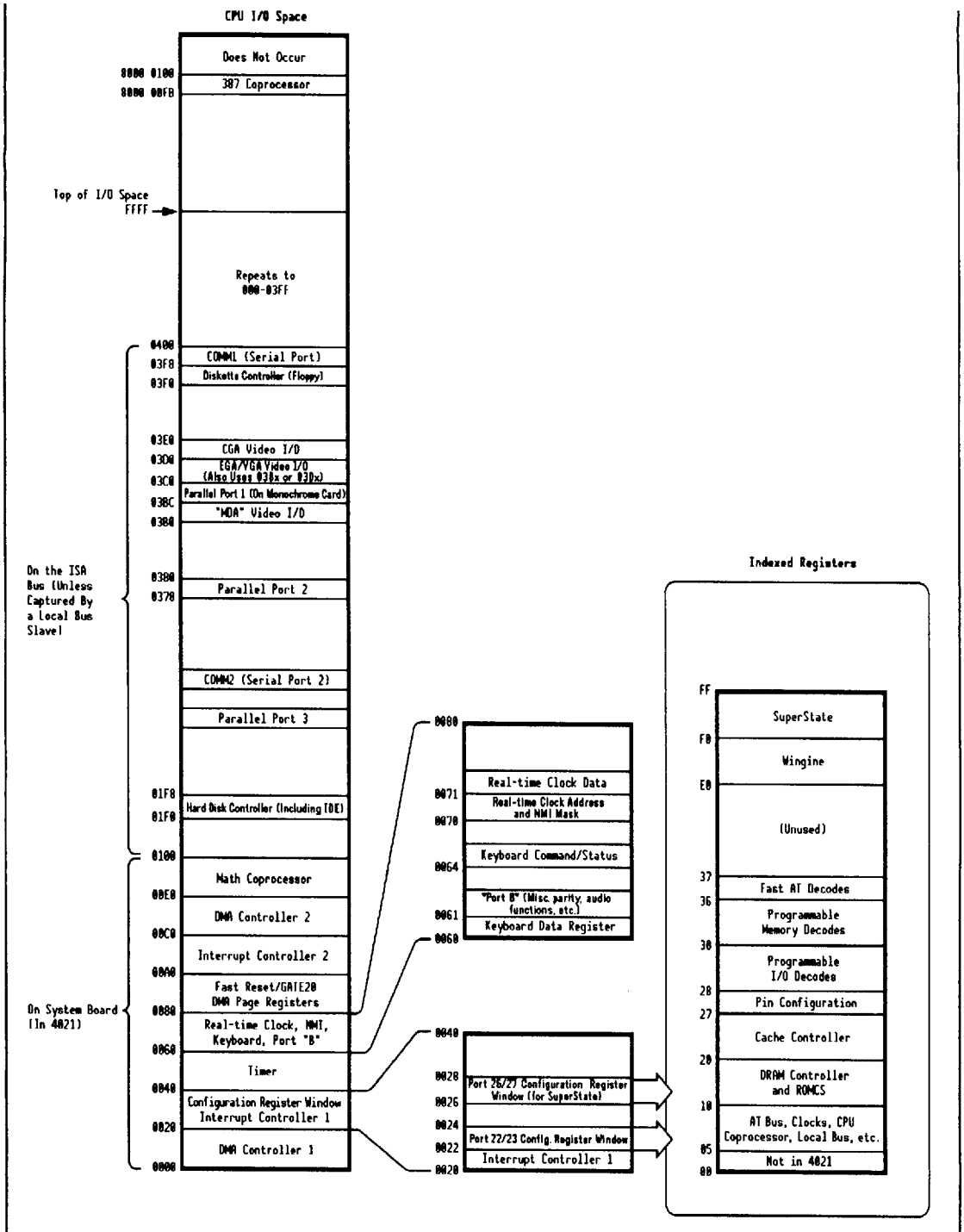
Many of the decodes used set an attribute for an I/O range such as the FASTAT clock. A conflict can arise if more than one decode points to the same place, such as a programmable decode being a superset of a fixed decode. In these cases, the attributes are ORed together. For instance:

- If a programmable decode is set to 1F0-1F7 and has the FASTAT clock bit set to 0, and the FASTAT clock bit for the HD data port (port 1F0) is a 1, accesses to 1F0 will use the fast clock. Accesses to 1F2-1F7 will use the slow clock, however. This is true for the 0WS and local bits as well.

T-49-17-01

**Figure 10-1.** *I/O and Configuration Register Map*





## 8042CS*

T-49-17-01

This is a decode of I/O ports 60 and 64. These ports are always on the XD bus. When the Fast GATEA20 and Fast CPU reset emulations are enabled, the IOW* command is sometimes withheld for these I/O ports; but the chip select is always generated. The 8042 chip select is combined with the ROM chip select. the 8042CS* section of the decode logic must include the M/IO* signal so that it is only generated for I/O cycles. It is not generated at all for DMA or ISA master cycles because it is not known before the command whether the address is a memory or I/O address.

## Some Fixed Decodes

There are two commonly used ranges which have fixed decodes in the 4021 chip with configuration bits to enable or disable the FASTAT functions and force the ranges to the local bus. These are the VGA and the hard disk data register. The decodes and the function of the bits are described in Table 10-1.

***Table 10-1.*** *I/O Fixed Decodes for Local Bus and FASTAT Functions*

| Area | I/O Range | On Local Bus | ISA Fast Clock | Force 0WS Read | Force 0WS Write |
|---|---|---|---|---|---|
| Video, MDA section | 03B0-03BF | 18 bit 4 | 36 bit 6 | 36 bit 0 | 36 bit 1 |
| Video, EGA section | 03C0-03C5<br>03CA-03CF | 18 bit 4 or<br>18 bit 5 | 36 bit 6 | 36 bit 0 | 36 bit 1 |
| Video, CGA section | 03D0-03DF | 18 bit 5 | 36 bit 6 | 36 bit 0 | 36 bit 1 |
| Video, palette DAC | 03C6-03C9 | 18 bit 6 | _ | _ | _ |
| Hard disk data register | 1F0 | _ | 36 bit 7 | 36 bit 2 | 36 bit 3 |

Four Video areas are defined as follows:

- The "MDA section" is the I/O area used by the original monochrome display adapter and is used by monochrome boards; which now include graphics modes in addition to the original character only mode. It includes the 6845 registers, and is also optionally used by EGAs and VGAs.
- The "CGA Section" is the I/O area used by the original "Color Graphics Adapter" and is used by color cards. It includes the 6845 registers, and is also optionally used by EGAs and VGAs.
- The "EGA Section" is new to the EGA. It is always used by the EGA and VGA card. Those cards also use either the MDA or CGA ranges for the 6845 registers, etc. The original EGA has a 4-pin dipswitch, which among other things, determines which I/O range is used. The range not used may be used by a second display card (either monochrome or color).
- The "Palette DAC" section is new to the VGA. It is unused in the EGA. It is kept separate because the palette chip is generally a separate chip with an 8-bit interface. Because of this, it is not on the local bus for local bus VGAs. Most palette chips do not accept FASTAT cycles either, and the speed of palette DAC writes generally does not affect performance.

## Programmable I/O Ranges

Two programmable ranges can be used for several purposes, such as to generate chip selects, to define an area to be on the local bus, and/or specify a range of I/O for the FASTAT functions. The same two decodes are used for all of these functions. Each address bit (0:9) has a compare bit and a mask bit. If the mask bit is a 1, that bit position is ignored in the decode. Each decode has a set of attribute bits to determine which function(s) it is currently being used for.

Having a separate compare and mask bit provides a great deal of versatility for the programmable ranges. The mask bits can be used to make the ranges show up for 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, or 1024 I/O ports. In addition, the decode may be active for non-contiguous ranges. For example, by setting the MASK 8, 2, 1, and 0 bits to a 1 and programming an address of 2F8, the decode will be active for 2F8-2FF and 3F8-3FF (the two COMM ports). Table 10-2 lists how the decode and mask bits are used.

***Table 10-2.*** *Programmable I/O Decode Compare and Mask Bits*

| Compare Bit | Mask Bit | Function |
|---|---|---|
| 0 | 0 | Compare address bit to 0 |
| 1 | 0 | Compare address bit to 1 |
| X | 1 | Ignore bit (active for either case) |

The lower two address bits are decoded two separate ways. For the chip selects, the XA0 and XA1 signals are used. XA0 can change during a bus conversion cycle and the chip select must reflect this change. For the FASTAT and LOCAL functions, the address decode must become available before XA0 and XA1 become valid. Therefore, BE3:0* are used to generate A0 and A1. These do not change during a bus conversion cycle, so the lowest address provided for the byte enables will be used to decode FASTAT and LOCAL* throughout the bus cycle.

For example:



A programmable decode is set for a single byte at location 100. It is used both to generate a chip select and enable the FASTAT clock. If a word is read from location 100 and IOCS16* is not pulled, the access, which consists of two byte reads from location 100 then 101, will both be done with the FASTAT clock. The chip select only goes active for the read from 100.



## Programmable Chip Selects

If the local bus master is not used (and it rarely is), HRQ2 and HLDA2* pins may be reconfigured as programmable I/O chip selects. Each chip select has a programmable 10-bit address compare and 10-bit address mask as discussed earlier. Each chip select may also have the following programmable options:

- **Disabled or enabled during ISA master accesses** — This bit determines whether or not the chip select is allowed to go active for ISA masters. The chip selects are always disabled during DMA.
- **Chip select or strobe** — If programmed as a chip select, the pin will go active strictly based on the address (including M/IO*, and W/R*). It will glitch as the address changes, but will have positive setup and hold times to the I/O command. If programmed as a strobe, the decode will be ANDed with the I/O command. This will produce a write strobe for a 374 used as an output port, a read strobe for a 244 used as an input port, etc.
- **Active for writes/read** — These bits determine whether the pin will be active for I/O reads, I/O writes, both, or neither. Typically, a chip select will be active for both reads and writes. When programmed as a strobe, it is generally desired to have it active for either reads or writes, but not both.
- **One special case occurs** — If the pin is set to be a decode, programmed to be active for reads but not writes or vice-versa, and programmed to be active by an AT master. When the master is in control, the 4021 cannot activate the chip select until after the command is active, since it does not know whether a read or a write will occur. The signal acts like a strobe, and not a chip select in this case. It is programmed to be active for both reads and writes, chip select timing will be used.

# Memory Address Mapping



By default, a memory cycle goes to the ISA bus, but very few actually manage to get there. Most are claimed by the Cache/DRAM controller. Some are also claimed by local slaves or the Weitek coprocessor.

T-49-17-01

The 4021 has the following internal I/O decodes:

| | |
|---|---|
| **Video I/O low** | A0000-AFFFF |
| **Video I/O high** | B0000-BFFFF |
| **Adapter ROM areas** | C0000-C3FFFF, C4000-C7FFFF, C8000-CBFFF, CC000-CFFFF |
| **Kitchen Sink area** | D0000-DFFFF |
| **BIOS extension area** | E0000-EFFFF |
| **BIOS main area** | F0000-FFFFF |
| **Weitek area** | C0000000-C1FFFFFF (decoded as A31=1, A25=0) |
| **High ROM area** | FFFF0000-FFFFFFFF (decoded as A31=1, A25=1) |
| **Cache Test Window** | 40000-7FFFF or 20000-9FFFF |
| **4 DRAM block decodes** | |
| **1 VRAM block decode** | |
| **2 Programmable decodes** | |

Internal decodes are used for the following purposes:

- Select the local DRAM and VRAM.
- Select the test window for cache data and tag RAMs.
- Generate ROMCS*. The 8042 chip select and ROM chip select are the same pin. See the ROMCS* section that follows.
- Select memory areas for FASTAT (both fast clock and 0WS I/O). There are bits for the low and high Video areas plus the two programmable ranges. See *Chapter 12: Advanced Device Interface* for more details on FASTAT.
- Select memory areas to appear on the local bus. There are bits for the low and high video areas, and the two programmable ranges. See *Chapter 12: Advanced Device Interface* for details on the Local Bus Support, and the section above titled Programmable I/O Ranges.
- Force KEN* high for the Weitek accesses.

Like the I/O decode, the memory decodes are used to set an attribute for a memory range. A conflict can arise if more than one decode points to the same place, such as a programmable decode being a superset of a fixed decode. In these cases, the attributes are ORed together. For instance:

The standard DRAM configuration registers have C0000-C3FFF shadowed and C4000-CFFFF not shadowed. A programmable decode is set to C0000-CFFFF. If the "Hole in DRAM" bit is set to a 0, the decode has no effect on the DRAM

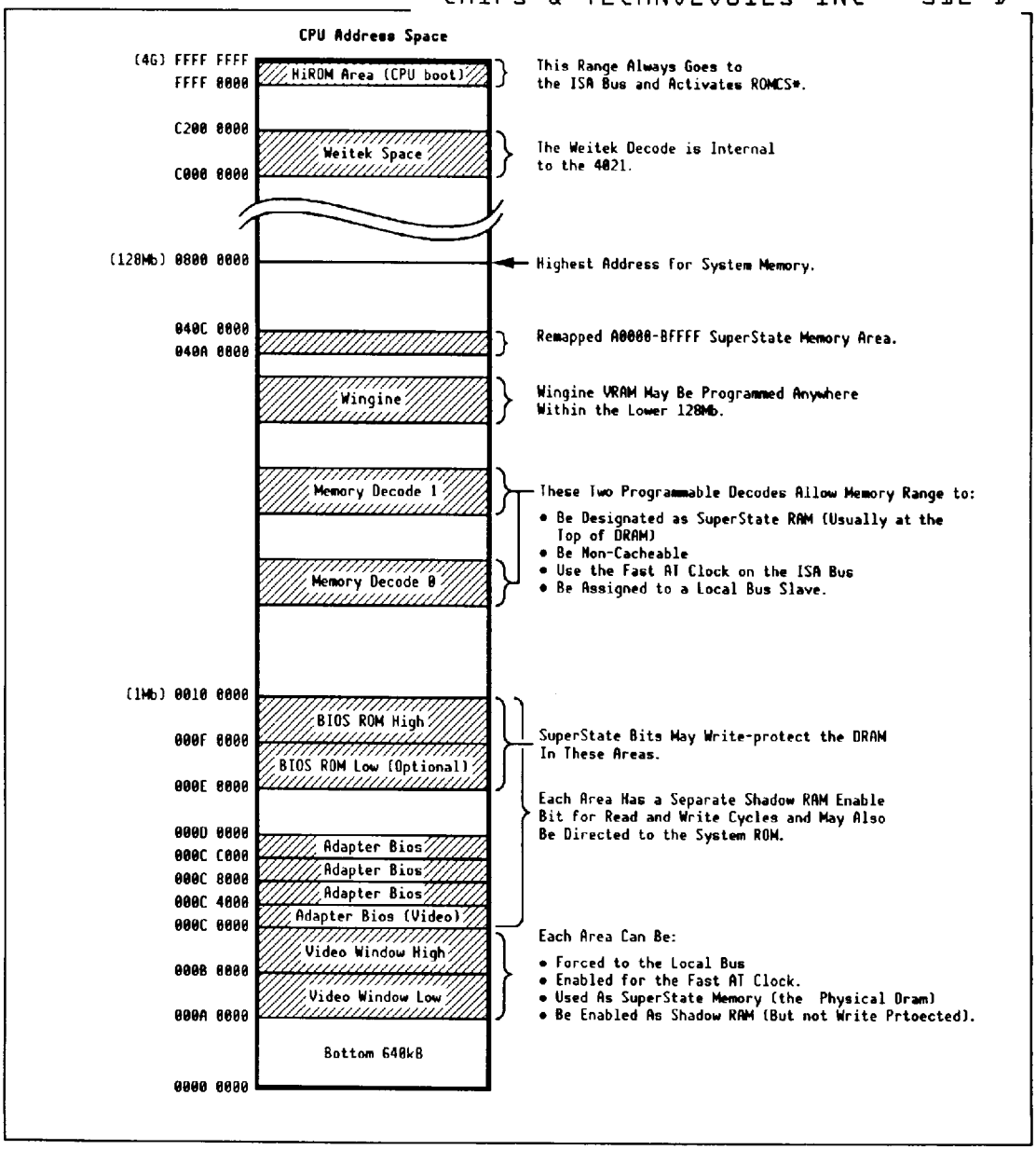
mapping. C0000-C3FFF continues to be shadowed, and C4000-CFFFF continues to go to the bus. If the programmable decode "Hole in DRAM" bit is set to a 1, the entire range is NOT shadowed and goes to the ISA bus unless captured by a local bus slave or the address is above 16M.

T-49-17-01

Figure 10-2 shows the typical memory map.

***Figure 10-2.*** *Memory Mapping Programming Options*





Note that "shadow RAM" bits are inverted before they are ORed with the "Hole in DRAM" bits. Either can indicate that the access is NOT from the DRAM FASTAT Fixed Ranges. There are fixed decodes for the A0000-AFFFF and B0000-BFFFF ranges. Configuration bits decide whether each range uses the normal or FASTAT clock. Table 10-3 identifies the fixed memory decodes.



***Table 10-3.*** *Fixed Memory Decodes*

| Address Range | FASTAT Bit |
|---|---|
| A0000-AFFFF | Index 36 bit 4 |
| B0000-BFFFF | Index 36 bit 5 |

## Shadow RAM

There are also fixed decodes for the shadow RAM and the ROM chip select. They determine which of seven areas go to local DRAM, or have the ROMCS* pin activated. Each of the seven ranges has a bit to determine whether reads go to the local DRAM or not, whether writes go to the local DRAM or not, and whether ROMCS* is active (which only occurs if the cycle goes to the ISA bus). These bits allow for all combinations of shadow RAM, write protection, etc. as indicated in Table 10-3 above.

In addition, the A0000 and B0000 ranges also have shadow bits. There are no separate read and write bits for these, since there is no need to write protect these areas. These areas are rarely shadowed since this is the video area. The physical DRAM at these locations may be used as SuperState V memory.

## ROMCS*

In most systems, 64K bytes or ROM will be used, at location F0000-FFFFF. The 4021 has the ability to allow up to 256K bytes of ROM, however, by programming any or all of the memory ranges between C0000 and FFFFF to activate ROMCS*. The most common areas to add to the ROM are:

| | |
|---|---|
| Extended BIOS area: | E0000-EFFFF |
| Video BIOS area: | C0000-C3FFF or C0000-C7FFF |

Note that a cycle will only get to the BIOS ROM if it becomes an ISA cycle. A cycle captured by local DRAM or a local bus slave will not access the system ROM. The address range of FFFF0000-FFFFFFFF will always activate ROMCS*. This range should not normally be captured by LOCAL DRAM or a local bus slave. Shadow RAM is controlled separately for reads and writes, and is the method used for write protecting it.

The ROMCS* pin is muxed with the 8042CS* signal. During CPU or local master accesses it is active for either a memory access to the ROM range or an I/O access to the 8042 ports. The ROM must connect MEMR* to its output enable to prevent it from turning on during 8042 cycles. The ROM cannot be accessed by DMA or a local master, however, shadow RAM can be accessed.



## Programmable Memory Ranges



There are two general purpose programmable memory ranges which allow several attributes to be set for blocks of memory. The ranges may be 64K, 128K, 512K, 1M, 2M, 4M, 8M, or 16M in size, and must be on a boundary which is equal to their size. If programmed for a 1M size, it must appear on a 1M boundary. Attribute bits allocate the memory range as SuperState memory, a "Hole in DRAM," a non-cacheable area, a FASTAT clock area, or a local bus slave decode.

A26:16 for each range may be programmed. A31 is decoded as a 0, and A30:27 are not included in the decode since these signals do not enter the 4021. The size of the range specifies which of the decode bits are ignored. This is done according to the following table.

***Table 10-4.*** *Programmable Memory Range Options*

| Range Size | Bits Use in Decode | Bits Ignored in Decode | Placement Boundary |
|---|---|---|---|
| 64K | 26 25 24 23 22 21 20 19 18 17 16 | none | 64K |
| | 26 25 24 23 22 21 20 19 18 17 | 16 | 128K |
| | 26 25 24 23 22 21 20 19 18 | 17 16 | 256K |
| | 26 25 24 23 22 21 20 19 | 18 17 16 | 512K |
| | 26 25 24 23 22 21 20 | 19 18 17 16 | 1M |
| | 26 25 24 23 22 21 | 20 19 18 17 16 | 2M |
| | 26 25 24 23 22 | 21 20 19 18 17 16 | 4M |
| | 26 25 24 23 | 22 21 20 19 18 17 16 | 8M |
| | 26 25 24 | 23 22 21 20 19 18 17 16 | 16M |

If all of the attribute bits are zeros, the programmable range has no effect. The following describes each attribute bit: T-49-17-01

**SuperState** — This bit is used in conjunction with the "Hole in DRAM" bit. If this bit is set, the local DRAM (if any) at this location is accessible from SuperState even if the Hole in DRAM bit is set. To allocate a section of DRAM as SuperState memory, both the Hole in Dram and SuperState bits should be set. Non-SuperState accesses to this area will go to the ISA bus (if below 16M). SuperState accesses (AADS* active) will go to the DRAM.

**Hole** — Hole in DRAM. This allows a section of DRAM to be disabled. There are several uses for this bit. If a peripheral on the local bus or ISA bus requires a specific memory range that would normally be occupied by DRAM, a hole may be placed in the DRAM to allow the peripheral to be accessed. It may also be used in conjunction with the SuperState bit to allocate a sectioin of DRAM as SuperState memory.

This bit has no effect if there is no DRAM (as specified by the DRAM configuration registers) at the address programmed for this programmable range. This mechanism may be used to make a hole in DRAM in the A0000-FFFFF range of the bottom megabyte, however a more convenient method is already provided for this using Index Register 18, 19, and 1A; providing separate bits for each of nine sections of that memory.

**Non-cache** — This bit will set the memory range as non-cache in both the 486 internal cache and the secondary cache. Note that only Local DRAM is cached in either cache, so it is unnecessary to use this mechanism to make ISA or local bus slave memmory areas non-cached. Wingine memory has a separate cache bit in the Wingine registers.

**FASTAT** — This bit specifies the memory range as using the fast AT clock. The operating frequency of both normal and fast ISA accesses is set in Register 07. This bit only has an effect if the cycle actually makes it to the ISA bus. For instance, if local DRAM or a local bus slave is accessed by the memory refrence, the cycle will not go to the ISA bus and this bit will have no effect. FASTAT only operates during CPU and local master cycles (as opposed to DMA and ISA master cycles).

**LOCAL** — This bit forces the memory range to a local bus slave without the need for the slave to pull the LOCAL* pin low. This may be used to save logic on a system where a local slave appears on the motherboard. When the LOCAL bit is set, this memory range is effectively ORed with the local pin internally, allowing either to force the cycle to a local bus slave. Note that this bit and the LOCAL* pin only prevents the ISA bus from taking the memory cycle. It does not prevent the DRAM controller from taking it. If there is local DRAM at the location that is to be used by a local bus slave, the Hole in DRAM bit, described earlier, should also be set. In general, if the LOCAL bit is set, the HOLE bit should also be set.

T-49-17-01

*Table 10-5. Programmable Memory Range Configuration Registers*



| Dec 0 Index | Dec 1 Index | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|
| 30 | 33 | A23 | A22 | A21 | A20 | A19 | A18 | A17 | A16 |
| 31 | 34 | Size 3 | Size 2 | Size 1 | Size 0 | — | A26 | A25 | A24 |
| 32 | 35 | — | local | FASTAT | non-cache | hole | — | — | SuperState |

# SuperState V™

T-49-17-0

SuperState V is a Super386 mode beyond the normal 386 and 486 modes of operation. It is a method for system software to get control of the system at any time without running into compatibility problems. The ISA486sx CHIPSet supports SuperState V in several ways:

- Provides SuperState V memory which is accessible only from SuperState V.
- Allows for an alternate window into the configuration registers using ports 26 and 27 so that SuperState V code may access configuration registers without disturbing user code port 22/23 accesses.
- Enables a locking mechanism for some of the configuration registers to prevent user code from getting to the SuperState V code or features.
- Includes an additional timer to enter SuperState V in a programmable amount of time.
- Redirects CPU soft resets into SuperState V calls to prevent exits from protected mode from effecting SuperState V operation.

## SuperState V Memory Space

Since SuperState V memory has to be protected from user accesses, a separate SuperState V memory region has to be defined. Two regions of local memory can be set up for this purpose. The first region is the local memory area 0A0000H - 0BFFFFH. This area of the memory map is normally reserved for the video frame buffer, and as such, the local DRAM at this location is not used. Either 64KB of SuperState V memory, or 128KB of SuperState V memory, is made available in this address range. Lock bits are provided separately for the two address ranges: 0A0000H - 0AFFFFH and 0B0000H - 0BFFFFH. The lock bits allow the memory to be accessed with SuperState V accesses (AADS* and ADS* both go active), but not user accesses (ADS* only goes active). Thus, ADS* accesses to B0000 will go to the video (on either the ISA bus or the local bus) and SuperState V accesses will go to the local DRAM.

A problem occurs if there is a local bus VGA. The VGA will not know that it should not respond to the SuperState V access. To get around this, A26 is ANDed internally with a signal which is optionally low for SuperState V cycles. This allows the physical DRAM at A0000-BFFFF to be accessed with a CPU address of

040A0000-040BFFFF, which will not cause the local but VGA to respond. This only works if there is no more than 64MB of local DRAM in the system. If there is more than 64MB, this feature must be disabled, and the A0000-BFFFF DRAM cannot be used as SuperState V memory when there is a local VGA. When greater than 64MB is installed (which is only possible in 8 RAS mode), it is unlikely that the 128K of RAM will be missed.

T-49-17-01

```
        If 64MB of local DRAM
        Set the "SuperState V A26 Mask" bit ON.
        Access the SuperState V memory at 040A0000-04BFFFF.
else
        Set the "SuperState V A26 Mask" bit OFF.
        If there is a local Master
                SuperState V memory at A0000-BFFFF is not available
        else
                Access the SuperState V memory at 000A0000-000BFFFF
        endif
endif
```



For certain applications, like power management, disk cache emulation, etc., 128KB of SuperState V memory is not sufficient. In order to provide a larger physical address space for SuperState V, a top-of-the-local-memory hole can be defined via the two programmable hole registers as another region. The minimum size of the hole is 64KB and the maximum size is 64MB. Although both hole registers (referred to as memory decode registers) can be used for this purpose, it is sufficient to just program one of these registers to define the memory hole which can be associated with SuperState V memory.

By programming the SuperState V memory to be at the top of local memory, the user (applications, DOS, etc.) only sees the memory below this hole as the physical memory available. Both the 0A0000H - 0BFFFFH and top-of-the-local-memory hole can be programmed for SuperState V memory accesses. However; together they form a disjoint physical memory address space. The SuperState V software that wants to take advantage of both the regions must be aware of this disjointed address space.

SuperState V memory is non-cacheable. By defining the 0A0000H - 0AFFFFH and 0B0000h - 0BFFFFH address ranges as SuperState V memory, they are automatically defined as non-cacheable memory regions. When the top-of-the-local-memory hole is defined as SuperState V memory (via the memory decode configuration registers), a separate control bit has to be set to make that memory non-cacheable. If this is not done, SuperState V memory at that address range becomes cacheable and operation is unpredictable.

# SuperState V Configuration Register Access (Port 26/27) T-49-17-01



For SuperState V, a second window into the configuration registers has been added. This is at I/O ports 26 and 27. They work exactly like ports 22 and 23, and access the same set of configuration registers. This second window is required to avoid the problem of a SuperState V access splitting a user code access to a configuration register. The situation is as follows:

- User code is attempting to read a configuration register. It writes port 22 and then reads port 23. Between the write and read, a SuperState V call occurs. SuperState V code accesses a configuration register through port 22/23 then returns. When the user code read from port 23 occurs, it reads bad data since the proper address is no longer in port 22, and two accesses to port 23 are done without a port 22 access between.



With the existence of the port 26/27 window, port 22 is not disturbed by SuperState V code which accesses configuration registers.

With port 26/27, there is also a level of protection added to prevent the user code from mucking with SuperState V. This is achieved by having port 22/23 and port 26/27 accesses follow the following rules:

- Port 26/27 accesses are protected such that they can only be accessed from SuperState V code.
- Some configuration registers (SuperState V Timer, ANMI source, etc.) can only be written through the port 26/27 window and are read only for port 22/23 accesses.
- Writes to some registers may selectively only occur through port 26/27. For example, writes can be written in both windows and subsequently can be made to be selectively written only from the port 26/27 window. Reads are always allowed from the port 22/23 window. There are the DRAM configuration and some other configuration registers which would allow a user program to manipulate SuperState V memory via reprogramming the configuration registers.

At power up, the port 26/27 window is accessible from user mode. The registers which are selectable appear for both port 22/23 and port 26/27 accesses. This allows the entire system to be configured as normal. Port 22/23 may be used as normal to configure the DRAM, etc. Port 26/27 must be used to access the SuperState V registers.

To protect the system, the SuperState V driver will set the registers that are selectable (DRAM configuration, etc.) to only appear for port 26/27 accesses. It will then protect 26/27 accesses, making them only accessible from SuperState V. The write that does the protection can be done from user code. Ports 26 and 27 disappear at the end of the write.

Once protected, ports 26 and 27 can be accessed only in SuperState V. This gives SuperState V access to all of the configuration registers. The 4021 also gives it the ability to unprotect whatever it wants to, by turning off the protection bits. Since I/O accesses from SuperState V do not generate AADS*; they are simply normal ADS* cycles. The 4021 detects SuperState V entry by seeing an ADS*=0 and AADS*=0 bus cycle; the first SuperState V access. It detects SuperState V exit by an ADS* code fetch (D/C* going low) bus cycle; the first non-SuperState V code access.

There is only one read signal, since all registers are always readable from both port 22/23 and 26/27. The write signals are for:

T-49-17-01

- "Unprotected" register that are always writeable from both port 22/23 and 26/27.
- "Selectively protected" registers in which a configuration bit determines whether they are writeable from both port 22/23 and 26/27 or only writeable from 26/27.
- "Protected" registers which are always only writeable from port 26/27.



## SuperState V Locking

In order to prevent accesses from user mode into SuperState V memory and the SuperState V timer (the SuperState V resources), a number of lock bits have been provided in the ISA486sx. These bits are all reset (in their unlocked states) at power up. After the BIOS has set up the various configuration registers, it can set the lock bits for different SuperState V resources which prevents accesses to these resources from non-SuperState V accesses. Lock bits are provided for the following resources:

- Memory address range 0A0000H - 0AFFFFH.
- Memory address range 0B0000H - 0BFFFFH.
- Memory address range 0EFFC0H - 0EFFC7H. This is actually a write-protect bit. The user can read and determine where the SuperState V memory space is located. But, by write protecting this range (one of the SuperState descriptor locations), the user cannot:
  1. Change the descriptor and thus cannot get control over SuperState V accesses.
  2. Obtain access to the SuperState V memory space since it is protected by lock bits.

  In the ISA486sx CHIPSet, this write protects the E0000-EFFFF range.
- Memory address range 0FFFC0H - 0FFFC7H. This is also a write protect bit, and is the second location where the SuperState V descriptor can be located. In the ISA486sx CHIPSet, this write protects the F0000 - FFFFF range.
- Port 26/27 accesses. Access to these ports are locked from user code. They are only accessible when in SuperState V mode. This prevents the user from being able to change configuration registers (like the SuperState V locking register which contains all these locks, the SuperState V timer, etc.), which are only accessible from port 26/27.

- Port 22/23 write operations to selective configuration registers. These configuration registers are index registers 10-17 (DRAM configuration) and 30-35 (memory decodes) which prevent the user code from changing the local memory configuration and thus sneaking into SuperState V memory areas.



The SuperState V entry point write protection is handled in a special way in the ISA486sx CHIPSet. The write protect decode is one of the most critical paths in the 4021. Having two special decodes which go all the way down to eight bytes would add to this critical path. Therefore, the existing 64K write protect decodes for E0000-EFFFF and F0000-FFFFF are used, and the SuperState V lock bits offer an additional method of write protecting the area.

Configuration Register 1A already provides a means of write protecting the E0000-EFFFF and F0000-FFFFF areas. These are actually "write enable" bits, where a 0 write protects (sends the cycle to the ISA bus) and a 1 write enables (send the cycle to the DRAMs). The SuperState V locks bits will be inverted and ANDed with the existing configuration register 1A bits. While SuperState V code is executing, the write protection for this area is controlled solely by the SuperState V lock bits. This gives the SuperState V code the ability to write to these areas without knowing where the shadow RAM write protect bits are (which is system-dependent).

## SuperState V Timer

In order for SuperState V code to schedule events, a special SuperState V timer is provided. Access to this timer is possible only through ports 26/27. The timer is enabled when a value is written to it. When the count goes down to 0, the timer generates an ANMI. Reading this timer will give the current count. Writing to this timer while it is counting will restart the count from the value which is written. This will not cause an ANMI unless it timed out just before it was written. Writing a 0 stops the timer without generating an ANMI.

The SuperState V timer is an 8-bit timer. It uses a 1 KHz clock (the 32.768 KHz clock divided by 32) for counting. The timer can provide a count from 0.976 ms to 249 ms with a resolution of 0.976 ms.

Reading the timer is a much more difficult hardware task than it initially appears. Since the read pulse is asynchronous to the clock which decrements the counter, it is possible to read the count just as it is decrementing; causing a metastable which could settle differently for different bits. This can lead to a totally erroneous value being read. To get around this, the software should write port 26, then read port 27 continously until the same value is read twice in a row. To avoid the metastable from going all the way to the CPU, the data from the timer is latched at the beginning of the read pulse allowing the entire read pulse length to become stable.

# ANMI Register

The 4021 contains an ANMI register which indicates which ANMI sources are interrupting an allows them to be cleared independently. It also contains a global ANMI mask. There is one bit for each ANMI source. When read, that bit will be a 1 if the source is generating an ANMI. When written, writing a 1 to a bit will clear the ANMI. Writing a 0 to a bit will have no effect on it. T-49-17-01

The global ANMI mask gates off the ANMI after the sources are ORed. It is basically an AND function before—ANMI leaves the chip. It will not effect status of the individual ANMI sources. It will also not effect the other bits in the register.

## SuperState V Soft Reset Redirection

In the SuperState V mode of operation, resets have to be handled in a special manner. In a system which is using SuperState V, and as such has SuperState V code, resets cannot be made to come into the system at any arbitrary time. Resetting the CPU while in SuperState V mode can leave the system in an unknown and uncontrollable state. The following two situations can occur on the ISA/486SX system:

- **SuperState V code exists:** All soft resets (Port 92 bit 0, Simulated 8042, and KBRESET*) in such a system are programmed, collectively, to generate ANMI and thus cause entry into SuperState V. The hard reset, PWRGOOD, causes a reset to the CPU.
- **SuperState V code does not exit:** In a system which is not supporting SuperState V, all reset requests actually generate a CPU RESET.

A configuration bit determines whether CPU soft reset requests generate a CPURESET or an ANMI.

CHAPTER 12

T-49-17-01

# Advanced Device Interface

The advanced device interface has been designed for high speed interface to peripheral devices on the ISA bus. The following three modes for high speed interfaces are provided by the 4021 chip:

- FASTAT interface
- LOCAL master/slave interface
- VRAM interface

## FASTAT Interface

The 4021 CHIPSet has FASTAT interface logic; it provides higher speed accesses on the ISA bus. There are two independent ways of speeding up accesses:

- Speeding up the ISA bus clock
- Forcing zero wait state I/O cycles

These functions are controlled independently. One or both functions can be used for any address range. Speeding up the clock applies to both memory and I/O accesses. Forcing zero wait states is used only for input and output (there is already a standard method of providing 0WS memory cycles).

## Using FASTAT With Existing ISA Bus Cards

The following three types of ISA bus cards are available for the FAST AT interface:

- **FASTAT Capable** — cards in which some or all of their memory and I/O space can execute faster cycles (either faster clock or 0WS I/O). Any portions of the card that cannot execute fast cycles must be at least FASTAT Tolerant compatible.

- **FASTAT Tolerant** — cards that cannot execute fast cycles but can tolerate the timing of fast cycles occurring to other peripherals. They can accept a faster BUSCLK signal that switches between normal and fast speeds. They can also tolerate the faster BALE signal, commands, and address setup time that occur on the fast slots. You can place FASTAT Tolerant cards in either the "normal" or "fast" slots.
- **FASTAT Intolerant** (XE "FASTAT Intolerant") cards cannot handle the bus timings that are present on the fast slots even though no cycles occurs with fast timing. There are not very many cards that cannot handle those bus timings. You must plug these cards into the "normal" speed slots.



To maintain ISA compatibility, you can split the ISA slots into the following two sets:



- Normal AT interface
- FASTAT interface slots

The FASTAT interface can also support the normal AT interface because the FASTAT mode is programable. To provide options for both types of AT interface on the same system, some external logic is required (F245 chip, two F32 gates, two F08 gates, and an inverter to the board). This additional logic blocks the fast commands from accessing the "normal" cards that are FASTAT Intolorent when a slot is accessed by using FASTAT cycles.

Fast slots receive FBUSCLK which switches between fast and normal clock rates for each access. FBUSCLK is always running at the frequency of the current ISA cycle (or most recent ISA cycle). BALE and the commands are always synchronous to this clock for CPU generated cycles.

Commands and BALE are received from the 4021 CHIPSet. They become active for all ISA cycles whether fast or slow timing is used.

Normal slots receive BUSCLK which is always executing at normal speed.

Buffered Commands and BALE that are blocked during fast cycles pass through during normal speed cycles.

## Fast ISA Bus Clock

The fast ISA method speeds up the ISA bus clock. It also increases the data transfer rate in the AT bus. Configuration registers determine which memory and I/O address ranges receive the fast bus clock and which ones receive the normal bus clock. The default setting allows memory and I/O address ranges to receive the normal bus clock. Only address ranges that have been programmed to receive the fast bus clock will receive the fast bus clock.

You can program normal and fast bus clocks independently. Use either CLKIN or ATCLK as an input. If you use CLKIN, you do not need to provide a separate oscillator. However, CLKIN may not always provide ideal frequencies.

The ISA bus has two clocks: BUSCLK and FBUSCLK. T-49-17-01

CHIPS & TECHNOLOGIES INC

### BUSCLK

BUSCLK is the standard ISA bus clock at about 8 MHz. Its speed is set at configuration time. The speed remains constant unless changed by using configuration software.

### FBUSCLK

FBUSCLK is the fast ISA bus clock. It accesses the "fast" slots and switches between normal and fast frequencies. During normal speed accesses, it is running at the same frequency as the BUSCLK signal. During any fast access, FBUSCLK executes at the fast rate.

FBUSCLK represents the clock frequency that is sent to the ISA state machine in the 4021 CHIPSet. The state machine usually runs at a 2x clock. The ISA clock is only switched when required. At the end of an ISA cycle FBUSCLK is always set at its current speed. At the start of an ISA cycle, a signal that is the OR of all of the decodes programmed for FASTAT clock is sampled. FBUSCLK then determines whether the standard or fast ISA clock is used. If the clock is switched, one of the phases of the idle cycle before TS is stretched during switching.

Clock switching adds time to the cycle. Therefore, fast clock cycles only improve performance when consecutive ISA cycles use the fast clock. This occurs frequently on many ISA peripherals, such as disk controllers and video displays. An average time of about 55 ns is needed each time the clock is switched.

When the fast clock is operating, the FASTAT signal is high. It prevents the BALE signal and commands from getting to the standard speed slots.

## Zero Wait State Input/Output

Because the 0WS signal is sampled on the same clock edge that generates the 16-bit I/O command, standard ISA timing does not provide for zero wait state I/O. The slave device does not know that an I/O cycle occurs before the command becomes active. Since the 4021 CHIPSet knows what occurs in advance, it pulls the 0WS* signal internally for the I/O slave. To make the I/O command last for a certain period of time, the normal command delay is also eliminated; thus making the timing exactly the same as zero wait state memory. The command width is 125 ns nominal.

Like the fast ISA clock, configuration registers determine the I/O address ranges that receive zero wait state cycles. They are programmable separately for I/O reads and writes. Like the fast ISA clock, the FASTAT signal is high for these cycles and blocks the BALE signal and the commands from the normal slots.

Unlike the fast ISA clock, there is no clock switching overhead involved. On each I/O cycle the logic determines whether it will or will not force 0WS.



## DMA and ISA Master Cycles



Fast AT cycles are only performed when the CPU, or a local master, is in control of the local bus. For DMA, ISA Master, and Refresh cycles, the standard speed clock is used and the FASTAT signal is low. The MASTER pins are kept separately on the standard and fast slots to determine the following:

- When an ISA master has control of the bus
- Which bus slots it is in

This information is used to control the direction of the F245 that drives the standard speed slot commands. However, the bus clock of the ISA master can be increased for higher speed systems independent of the FASTAT signal. This can improve the ISA master performance.

## FASTAT Programmable Registers

The 4021 chip has configuration registers that control the following:

- I/O and memory decodes to enable the FASTAT functions for specific ranges.
- Fast and slow clock speed and source selection.

For the programmable decodes there are a total of two memory and two I/O. Each can be programmed to enable multiple functions, but can only be programmed for one address range at a time.

**FASTAT clock programming:**

Fixed decodes (one bit for each selects FAST clock) T-49-17-01

- A0000-AFFFF memory range
- B0000-BFFFF memory range
- VGA I/O range (ports 3B0-03BB, 03BF-3C5, 03CA-03DF (all but palette DAC))
- Hard disk data port (I/O address 1F0)

**Programmable decodes**

- 2 memory ranges
- 2 I/O ranges



**0WS I/O forcing** (separately selectable for reads and writes)

Fixed decodes

- VGA I/O range
- Hard disk data port (I/O address 1F0)

Programmable decodes

- 2 I/O ranges

## Board Level Implementation

The F245 blocks the commands from the normal slots when the FASTAT signal is high. When the FASTAT signal is low, the F245 drives the commands toward the standard slots. The command signals for the fast slots are connected directly to the 4021 chip. Therefore, for ISA master in the normal slots, the MASTER* signal from the normal slot is used to change the direction of the F245 and the command signals is now driven from the AT master to the 4021 chip. The MASTER* signal from the fast slot and the normal slot is logically ORed and connected to the MASTER* input pin of the 4021 chip.

The gates (F32) on SMEMR* and SMEMW* block those commands during any fast cycles. Since these signals are always sent from the 4021 chip, a bidirectional driver is not required.

# Local Bus Support

The 4021 CHIPSet supports both local bus slaves and local bus masters to allow higher performance peripherals. It is designed to work with local bus devices on the motherboard or proprietary slots, and to support the emerging VL-Bus standard.

The VL-Bus standard is from the Video Electronics Standards Association (VESA). It specifies a standard interface, including a connector, for local bus slaves and masters. Up to three slots are supported. Adapter cards are expected for Video, Disk, and LAN interfaces, as well as other applications.

The following local bus support is provided: T-49-17-01

**A Clocking scheme** which provides a 1x clock with little or no skew to the local bus device or slots even when a 386 CPU is used. The VL-Bus requires this 1x clock. A 2x clock with little or no skew is also available on the motherboard (but not required by a 486 cache system).



**Support for the LDEV*** signal to allow a local bus slave to claim a bus cycle. For timing reasons, each VL-Bus slot provides a Totem-Pole output on LDEV*, which are ORed together with an F21 (for 3 slots) along with 0WS* from the ISA bus and fed to the 0WS*/LOCAL* pin on the 4021. During ISA cycles, this is the 0WS* pin, and is driven high by all local bus slaves. During the first 2 or 3 T-states of each bus cycle, this pin is LOCAL* which has the exact same definition as LDEV* on the VL-Bus.

The 4021 samples LOCAL* at the end of either the first or second T2 of each bus cycle, as specified by a configuration register. If it is low at the sample point, the cycle does not go to the ISA bus. Not that the LOCAL* signal does not take the cycle away from the cache or DRAM controllers because to do so would greatly effect performance at higher speeds. A Hole in DRAM may be programmed using either of the programmable memory decodes in the 4021 if a local bus slave is to reside where there is local DRAM present. This is all in accordance with the VL-Bus specification.

**Internal I/O and Memory decodes for Local Bus Slaves.** The two programmable I/O decodes and two programmable memory decodes may be used to force a cycle to the local bus. In addition, there are five bits in Register 18 to allow the video memory and I/O ranges to be forced to the local bus. These eliminate the need for a local bus slave to generate the LOCAL* (LDEV*) signal (although a VL-Bus slave is required to generate LDEV*). The programmable memory decodes also allow holes in DRAM to be programmed, if the local bus slave occupies a memory area that would normally be occupied by local DRAM.

**Generation of local bus control signals for DMA and ISA Master cycles.** During DMA and ISA Master cycles, the LOCAL* signal as well as the internal local bus decodes are checked to see if the addressed device is on the CPU local bus. If it is, an ADS* is generated to read or write the local bus device. The data is passed through the 4025 to or from the ISA bus. IOCHRDY is pulled low by the 4021 until the local bus slave completes the cycle. On reads from the local bus the data is latched in the 4025 when the salve generates RDY* and held there until the DMA controller or ISA master takes the command signal high.

HRQ2 and HLDA2* supported for Local Bus Masters. HRQ2 and HLDA2* go into the internal 4021 arbitration logic along with the DMA controller and refresh requests. A local master may gain access to the bus by taking HRQ2 high and waiting for HLDA2* to go low. When it has the bus it drives the address and control signals as if it is a CPU and may do any type of bus cycle. The VL-Bus calls these signals LREQ* and LGNT*, and has a separate set for each slot. For a single slot LREQ* may be inverted and connected to HRQ2 and HLDA2* may be used for LGNT*. For multiple slots a PAL is required to arbitrate between the sets of LGNTs and LREQs*. To fully support the VL-Bus specification, the local bus

master must be pre-empted off of the bus when another device (specifically DMA in this system) requires control. In the current revision of the 4021 CHIPSet this is done with a small amount of external logic.

The following figure shows the basic logic required to support a 3-slot VL-Bus system. Address and/or Data bus buffers may be required at higher frequencies. This will depend on loading and the number of slots. See the VL-Bus specification for more information. The PAL arbitrates between the three sets of LREQ*/LGNT* signals, and creates the RDYRTN* signal from LRDY*.



***Figure 12-1.*** *ISA/486 VESA Support Implementation*



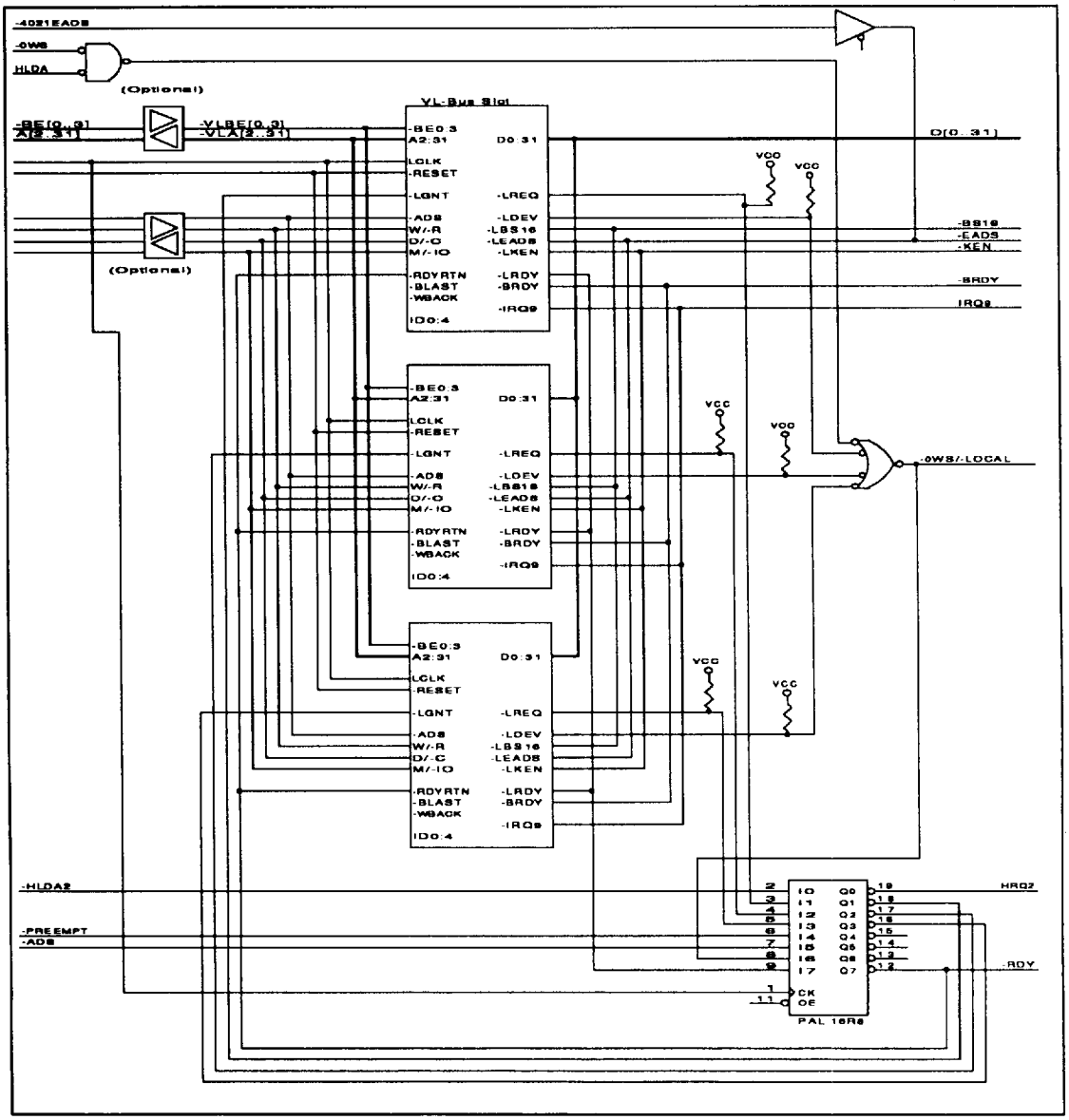
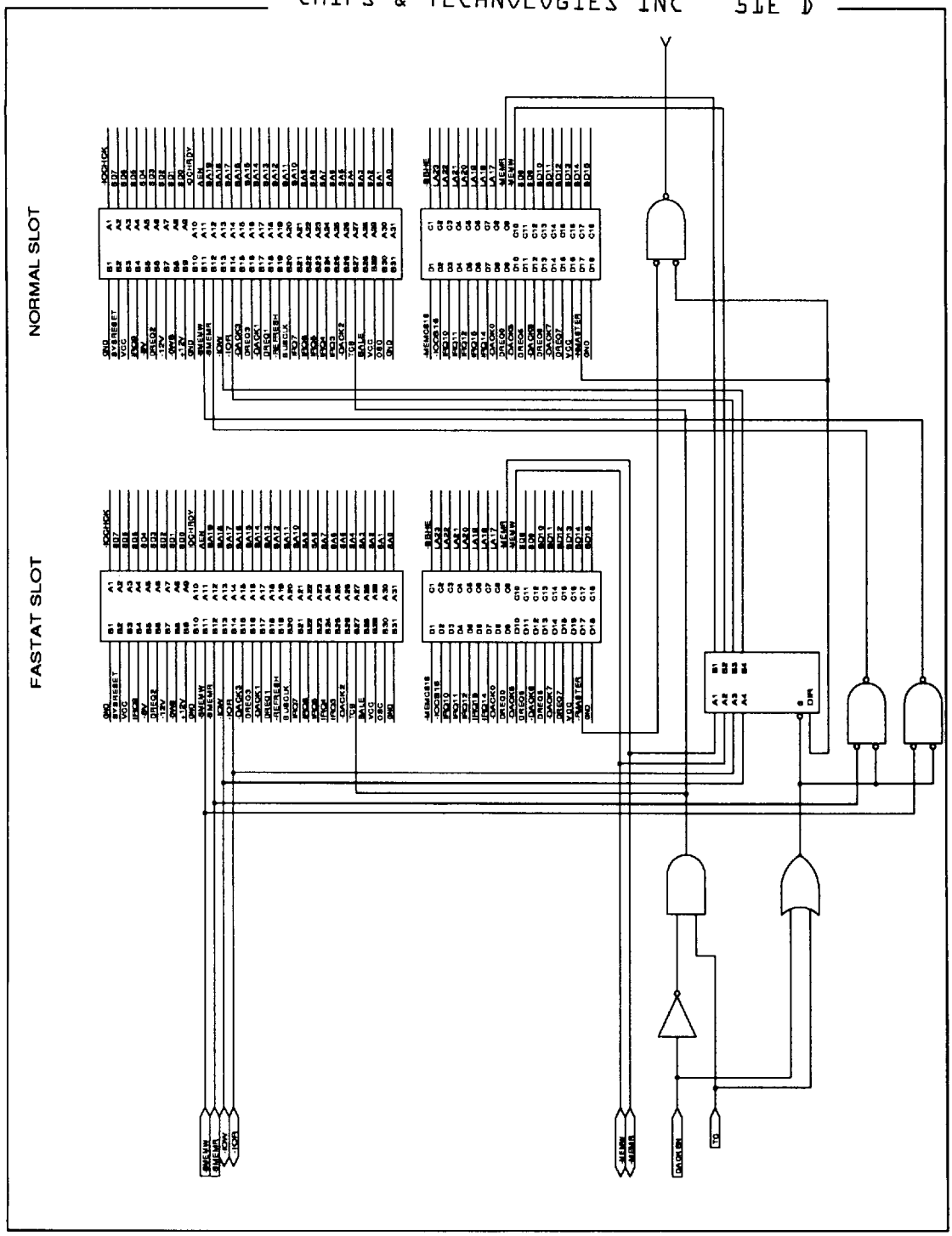


*Figure 12-2. FASTAT Bus Board Level Implementation*

T-49-17-01

CHIPS & TECHNOLOGIES INC 51E D



## Local Masters

Local Masters can be considered as the second CPU in the local bus. Local Masters gain control through HRQ2 and HLDA2. They perform normal 386 or 486 type bus cycles. The CHIPSset responds to these as if they were sent from the system CPU. When the CPU has a 486 chip, the 4021 chip can optionally drive the BLAST* signal low internally when HLDA2 is high. This eliminates the need for the LOCAL bus master to drive it low externally. If the local master supports the BLAST pin, the pin can be connected. The 4021 chip then responds as if it were interfaced with the 486 chip with burst enabled.

When a local master has control, the 4021 chip continues to drive the BRDY* and the RDY* signals as if the CPU were in control. For 386 and Super386 CPUs, the RDY* signal is generated instead of the BRDY* signal. For 486 CPUs, the LOCAL master must receive the RDY* and the BRDY* signals.

## Local Slaves

Local bus slaves drive the LOCAL* signal (time muxed with the 0WS* signal) to indicate they handle this cycle. This implementation applies to memory and I/O cycles. The LOCAL* pin only prevents the AT controller from taking the cycle, but does not prevent the DRAM or cache controllers from conducting an access since there is no time for the DRAM decode. Local peripherals should not be programmed to respond to an area to which the system memory is programmed to respond. Conflicts can be handled by disabling DRAM for the area the local slave wants to occupy. This task can be accomplished by using the shadow RAM control bits in the DRAM controller, or the two programmable memory decode ranges that have an option bit for "Hole in DRAM."

The 4021 chip has several decodes that can be used to force an I/O or memory range to the LOCAL bus. If this is done, the LOCAL* pin need not be driven externally. The ranges are as follows:

**Fixed Memory Decodes**

| | |
|---|---|
| Lower Video | A0000-AFFFF |
| Upper Video | B0000-BFFFF |

**Fixed Video Decodes**

| | |
|---|---|
| MDA & EGA video I/O | 03B0-03BB, 03BF, 03C0-03C5, 03CA-03CF |
| CGA & EGA areas | 03D0-03DF, 03C0-03C5, 03CA-03CF |
| Video Palette DAC | 03C6-03C9 |

**Programmable decodes**

two memory decodes, 64K to 64M in size
two I/O decodes, 1 to 1K bytes in size

Each fixed decode has a bit indicating whether the I/O or memory range should be forced to the local bus or not. The programmable ranges are used for many purposes, one of which is forcing the range to the local bus. Each has a programmable address, a programmable size (the I/O ranges are done with an address mask field) and several attributes to indicate what the decode does. One of these is a "LOCAL" bit which, when set, forces the range to the local bus.

T-49-17-01

For AT master and DMA cycles to local peripherals, the 4021 can optionally generate a local bus cycle for these peripherals. M/IO*, D/C*, W/R*, and BE0-3* are always generated from the AT commands and address for AT masters and DMA. The optional part is the generation of ADS* and data steering.

The 4021 has a local memory and a local I/O decode signal internally. When the AT command goes low, it knows which one to use. If it determines that the cycle is for a local bus peripheral, it will pull IOCHRDY low and drive ADS* for one T-state. It will then wait for RDY* or BRDY* to be driven by the local slave. When it receives one of the READYs, it will take IOCHRDY high. Proper data steering is done by the CS4025 from the action codes received from the 4021. The action codes and steering are the same as for DMA and master accesses from the cache RAMs. The data steering and ADS* generation can be controlled separately. They are global, and not selectable on a per device basis.



## VRAM Interface

The 4021 interface to the VRAM is provided by the normal DRAM read/write cycle and the VRAM Transfer cycles.

## Wingine™ Concept

The concept behind Windows Video support (Wingine) is to have a bank of memory controlled by the system memory controller comprised of dual ported VRAMs (Wingine memory). The system memory controller is optimized for high speed accesses from the CPU. Having it control the video memory allows for very high speed accesses to video memory much faster than achieved by standard VGA chips, even if they are on the local bus.

The serial port on the VRAMs is controlled by the Wingine chip, which is a VGA that has been modified specifically for Wingine support. It can also function as a standard high performance VGA, which is required in systems for compatibility. The Wingine chip performs all of the timing associated with the video display.

In a typical system, the software switches back and forth between normal VGA video and Wingine video. In normal DOS applications the VGA is used. When an application or operating system is envoked which has a driver for Wingine, it switches to the Wingine video mode. When exiting the application or operating system, the system is switched back to the normal VGA. The following environments may take advantage of Wingine:

1. Windows
2. Presentation Manager
3. X Windows
4. Other GUI environments
5. Graphics Applications such as CAD programs

T-49-17-01



The Wingine memory is a linear array of memory, implemented with VRAMs. During normal CPU accesses to the array, it looks just like system DRAM. In the 4021 it is the fifth block of RAM, having its own memory decode. The Transfer cycle (serial shift register capability) of the VRAMs is used to provide a high speed 32-bit wide data stream to the video RAMDACs.

Since the VRAMs are connected to the normal DRAM interface, which is 32 bits, and optimized for speed, the video information can be quickly updated. This is important for the higher resolution modes. The 1024x768, 256 color mode uses 3/4 of a megabyte of memory. This implementation is NOT VGA compatible, so it requires special software drivers for each of the environments mentioned earlier. Regular VGA logic is also present for compatibility. During that time, this VRAM is not used for video, and all special functions associated with it are disabled. Optionally, the VGA may take over control of the VRAMs for normal VGA operation. This requires isolation buffers to the memory control signals and data bus.

## 4021 Implementation

Wingine™ support consists of an extra bank of DRAMs that is controlled by the 4021 DRAM controllers in addition to the 4 or 8 banks (depending on mode) that are normally supported. 256Kx4 Video DRAMs (VRAMs) are connected to this bank.

The 4021 treats the VRAMs like normal DRAMs with the following exceptions:

- The timing mode may be programmed separately.
- MA10 is DSF and MA11 is TROE*, which has some special requirements for all cycles, including normal reads, writes, and refreshes.
- Transfer cycles must be performed when requested by the Wingine VGA.

The shift register in the VRAM holds one row of data from its DRAMs. In most cases, it is sufficient for two horizontal scan lines. The 4021 chip must perform a special "Transfer Cycle" during horizontal blanking to transfer a new row into the VRAM shift register. This is a standard RAS-CAS cycle with the TROE* pin (of the VRAM) low when RAS falls. The 4021 chip contains an internal counter to determine the row that needs to be transfered.

The 4021 chip receives a signal that is some form of composite BLANK from the video chip to determine when to perform a transfer cycle. The pin is called XREQ*.

The 4021 chip has about 4 us to accomplish this task. It receives the following information from the XREQ* signal:

T-49-17-01

| Event | Action |
|---|---|
| Horizontal Blanking | Do a transfer cycle |
| Vertical Blanking | Reset the counter |
| Odd/Even Field | Transfer odd/even rows |



The odd/even field is for interlaced mode only, which is not the normal mode. In interlaced mode, the 4021 supplies a transfer address of 0, 2, 4, etc., for the even field, and 1, 3, 5, etc., for the odd field. In non-interlaced mode, the 4021 counts normally i.e., 0, 1, 2, etc.

VRAMs are accessed by the 4021 like standard DRAMs. However, they cannot be counted to the same speed. The 4021 is designed to use VRAMs as slow as 100 ns with a tCAC of 30 ns. The VRAMs are programmed to be addressed above the system memory, possibly with a hole to prevent normal accesses. They cannot be interleaved with any other DRAM. The VRAM bank uses the regular CAS lines CAS0-3*. There is also an access enable bit to prevent accidental writing. This bit can be turned ON and OFF by the video driver program. Unlike normal video, it is acceptable to cache this VRAM. Due to the timing requirements of the cache logic, VRAM can only be cached in the 486 internal cache for Wingine. Wingine memory does not benefit from caching, since blocks of data are normally moved and the same data is not read multiple times. Caching can actually slow the system down by thrashing the cache; tossing the useable cache data out of the cache. The VRAM benefits most from caching due to the reordering of cycles on a string move which may occur from cache line fills. Instead of READ-WRITE-READ-WRITE, the cycles will be READ-READ-READ-READ-WRITE-WRITE-WRITE-WRITE to the VRAMs, causing page hits 75% of the time instead of all page misses. The 4021, with secondary caches, causes this reordering because of its four double-word line size.

The 4021 has support for 1Mbyte or 2Mbytes of VRAMs. This is 256Kx32, or 512Kx32 which can be implemented with one or two banks of VRAMs. 2Mbytes is only required for the 1280x1024 8-bits per pixel video mode. The following table shows allowable VRAM configurations for the 4021. Note that two banks of 256K parts is not directly supported. The 4021 treats it the same as one bank of 512K parts and the banks must be separated externally.

CHAPTER 13



# Configuration Registers

T-49-17-01

This section summarizes the configuration registers of the 4021 CHIPSet. The following table lists all the 4021 I/O ports.

**Table 13-1.** *4021 I/O Port Summary*

| Ports | Description |
|---|---|
| 00-0F | DMA controller #1 (8-bit DMA) |
| 20-21 | Interrupt controller #1 (IRQ7:0) |
| 22 | Configuration register address port |
| 23 | Configuration register data |
| 26 | SuperState V configuration registers address port |
| 27 | SuperState V configuration register data |
| 40-43 | Timer chip (8254) |
| 60 | Keyboard data port |
| 61 | "Port B" |
| 64 | Keyboard command/status port |
| 70 | Real-time clock address port and NMI mask |
| 71 | Real-time clock data port |
| 80-8F | DMA page registers |
| 92 | Fast CPU reset and GATEA20 |
| A0-A1 | Interrupt controller #2 (IRQ8-15) |
| C0-DF | DMA controller #2 (16-bit DMA) |
| F0-F1 | 387 error reset |
| 800000F8-800000FF | Coprocessor ports |

**Port 00-60** *I/O Port Addresses 00-60*

T-49-17-01

CHIPS & TECHNOLOGIES INC 51E D

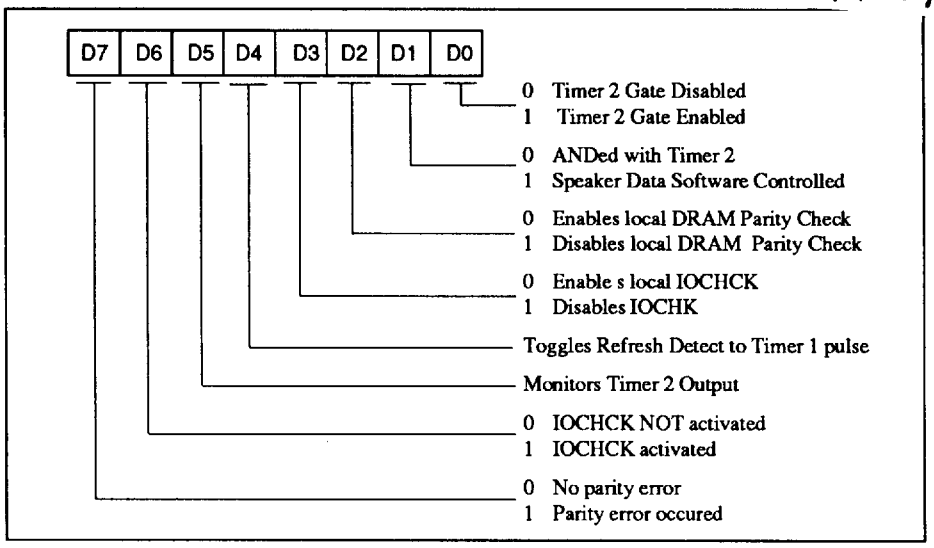
| Port Address | Name | Bits | Description |
|---|---|---|---|
| 00-0F | DMA controller #1 (8-bit DMA) | | These ports are contained in the IPC. The DMA controller will not respond to accesses to 10-1F (in the original AT 00-0F repeats at 10-1F). |
| 20-21 | Interrupt controller #1 (IRQ0-7) | | These ports are contained in the IPC. The interrupt controller does NOT respond to ports 22-3F. |
| 22 | Configuration register address port | | Write only port that holds the address of the Chips and Technologies index register to be accessed through I/O port 22. This register must be written before each access to port 23, even if the same index register is being accessed twice in a row. |
| 23 | Configuration register data | | Accessing this port accesses the configuration register pointed to by port 23. A second access to port 23 without writing port 22 in between will be ignored. If port 22 points to a register contained in the CPU chip, a read from this port will pull LOCAL* low, and the cycle will be done internally. Write cycles will always go to the bus. |
| 26 | SuperState V configuration register address port | | This is an alternate port into the Chips and Technologies index registers that is reserved for SuperState V code. It prevents SuperState V from splitting a port 22/23 access by user code and corrupt the 22/23 register data (a SuperState V call could come between a port 22 and port 23 access).<br><br>Port 26/27 accesses can be protected to prevent accesses from user code. Some index registers are accessible only through 26/27 and hidden from accesses to 22/23. Port 26 is the address port and works just like port 22, but it has a separate address storage location from port 22. |
| 27 | SuperState V configuration register data | | Data port for SuperState V configuration register accesses. The index register pointed to by the last write to port 27 will be read or written through this port. |
| 40-43 | Timer chip (8254) | | These ports are contained in the IPC. The timer does not respond to ports 44-4F. |
| 60 | Keyboard data port | | Used for keyboard GATEA20 and Fast Reset function. |

# I/O Port Address 61

This "B Port" is an AT compatible port with miscellaneous information. Bits 0-3 are read and write. Bits 4-7 are read only. The default value is 0F.

***Port 61*** *B Port - AT Compatible*

T-49-17-01




*Bits:* 0 Timer 2 is for the speaker. If this bit is 1, the timer is enabled, and if programmed to do so, will produce a square wave of the desired frequency. When this bit is 0, the timer output will be high R/W.

1 This bit is ANDed with the output of Timer 2 to produce the signal actually sent to the speaker. When the gate is low (bit 0 above), this bit gives direct software control of the speaker.

2 A 0 enables local DRAM parity checking. A 1 disables local DRAM parity checking and clears the local parity error flip-flop. This bit is inverted and sent to the active low preset of a flip-flop. The Q output is PCK*, and is fed to the NMI logic. A parity error clocks the flip-flop to 0. Also, an index register bit is used to block local DRAM parity errors. It prevents the flip-flop from being clocked. The flip-flop is an F74 on the AT, where the preset has precedence on Q and Clear has precedence on Q.

3 Enable IOCHCK. 0 enables the local IOCHCK interrupt.

4 This read only bit toggles on each refresh. It should toggle whenever Timer 1 produces a pulse (about every 15 us). This should be done even if ISA refresh is disabled. Some software uses this as a time delay.

5 This read only bit allows software to monitor the output of Timer 2; which is ANDed with bit 1 of this register to produce the speaker signal.

6 A 1 indicates that IOCHCK* has occured, and a 0 indicates no occurance.

7 A 1 indicates a local parity error has occured, and a 0 indicates no parity error.

# I/O Port Addresses 70-8F

T-49-17-01

The following table summarizes the port addresses 70-8F.

**Port 70-8F** *I/O Port Addresses 70-8F*

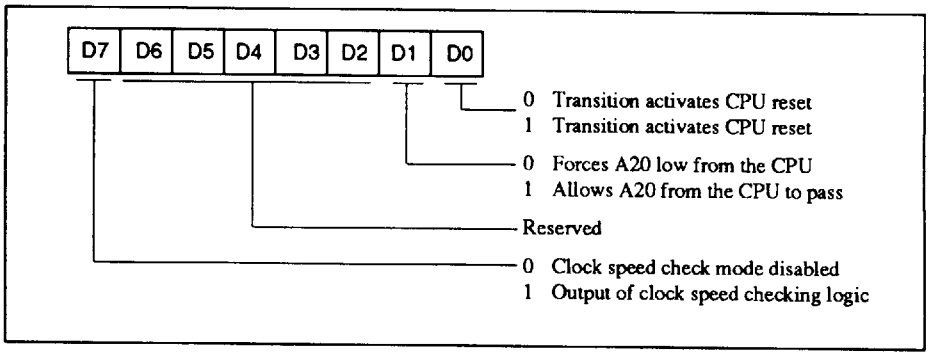
| Port Address | Name | Bits | Description |
|---|---|---|---|
| 70 | Real-time clock address port and NMI mask | 0-6 | The value written to these write only bits becomes the address of RTC/CMOS RAM that will be read or written through Port 71. |
| | | 7 | This bit is inverted and ANDed with the NMI sources (the OR of several sources). The result of the AND function is NMI to the CPU. |
| 71 | Real-time clock data port | | Reading or writing to this port will read or write the RTC register pointed to by the last write to Port 70. |
| 80-8F | DMA page registers | | These ports are contained in the IPC. They provide A23:16 for DMA accesses (A23:17 in the case of 16-bit DMA). |
| | | | 80 Not used. This port is written to by BIOS routines to indicate the BIOS status. A pair of HEX 7 segment LEDs are often put on test boards to display this information. |
| | | | 81 Channel 2 page register |
| | | | 82 Channel 3 page register |
| | | | 83 Channel 1 page register |
| | | | 84-86 Not used |
| | | | 87 Channel 0 page register |
| | | | 88 Not used |
| | | | 89 Channel 6 page register |
| | | | 8A Channel 7 page register |
| | | | 8B Channel 5 page register |
| | | | 8C-8E Not used |
| | | | 8F Refresh page register. This register is placed on A23:16 during refresh cycles (classic AT refresh only). |

## I/O Port Address 92

T-49-17-01

This Fast CPU reset and GATEA20 port is active only if Index 2B bit 6 is set to 1.

***Port 92*** *Fast CPU Reset and GATEA20*




*Bits:* 0 A 0 to 1 transition activates a CPU reset.

1 ORed with other GATEA20 signals (from 8042, for instance). A 0 forces A20 low from the CPU (assuming all other GATEA20s are low). A 1 allows A20, from the CPU, to pass through.

6:2 Reserved. Read as 0.

7 This read only bit is the output of the divider chain for clock speed checking (see Configuration Register 0D). When the clock check mode is disabled, this bit reads as 0.

0 Clock speed check disabled
1 Output of clock speed checking logic

# I/O Port Addresses A0-FF

T-49-17-01

The following table summarizes the port addresses A0-FF.

*Table 13-2. I/O Port Addresses A0-FF*

| Port Address | Name | Bits | Description |
|---|---|---|---|
| A0-A1 | Interrupt controller #2 (IRQ8-15) | | These ports are contained in the IPC. The interrupt controller does not respond to ports A2-BF. |
| C0-DF | DMA controller #2 (16-bit DMA) | | These ports are contained in the IPC. Only the even numbered ports are used. Reads or writes to the odd numbered ports will access the same register as its corresponding even numbered ports. |
| F0-F1 | 387 error reset | | Writing to either F0 or F1 causes the error latch (which also generates IRQ13) to be cleared. The data is ignored. |
| 800000F8-800000FF | Coprocessor ports | | These ports are used to access the 387. This is the only I/O with A31 high. In this CHIPSet, this will only occur with the 38605 CPU since the 486 coprocessor cycles do not appear on the local bus. The 4021 produces a one wait state ready for I/O with A31 high when configured to do so by index register 09 bit 0. |

The following table summarizes the 4021 index registers.



***Table 13-3.*** *4021 Index Register Summary*



| Index | Description |
|---|---|
| 05 | AT bus command delays |
| 06 | AT refresh, wait states, address hold |
| 07 | Clock selection |
| 08 | Performance control |
| 09 | Performance control enable |
| 0A | Soft reset and GATEA20 |
| 0B | CPU and coprocessor |
| OC | Local bus configuration |
| 0D | Clock speed check and clock status |
| 0E-0F | Do not exist |
| 10 | DRAM timing |
| 11 | DRAM setup |
| 12 | DRAM configuration |
| 13 | DRAM configuration |
| 14 | DRAM block 0 starting address |
| 15 | DRAM block 1 starting address |
| 16 | DRAM block 2 starting address |
| 17 | DRAM block 3 starting address |
| 18 | Video area shadow and local bus control |
| 19 | DRAM shadow read enable |
| 1A | DRAM shadow write enable |
| 1B | ROMCS enable |
| 1C-1F | Do not exist |
| 20 | Cache information |
| 21 | Cache information |
| 22 | Cache testing control |
| 23 | Data port for tag testing, bits 0-7 |
| 24 | Data port for tag testing, bits 8-10 |
| 25-26 | Does not exist |
| 27 | Pin configuration |
| 28 | I/O decode #0 address low |
| 29 | Chip select #0 mask low |
| 2A | I/O decode #0 address and mask high |
| 2B | I/O decode #0 configuration |

*Table 13-3. 4021 Index Register Summary (continued)*

T-49-17-01

| Index | Description |
|---|---|
| 2C | Chip select #1 address low |
| 2D | Chip select #1 mask low |
| 2E | I/O decode #1 address and mask high |
| 2F | I/O decode #1 configuration |
| 30 | Memory decode #0 address low |
| 31 | Memory decode #0 address high, block size |
| 32 | Memory decode #0 configuration |
| 33 | Memory decode #1 address low |
| 34 | Memory decode #1 address high, block size |
| 35 | Memory decode #1 configuration |
| 37-DF | Does not exist |
| 36 | Fast AT function |
| E0 | VGA Wingine control port |
| E1 | Does not exist |
| E2 | Wingine mode |
| E3 | Wingine VRAM configuration |
| E4 | VRAM starting address |
| E5 | Wingine VRAM timing mode |
| E6-EF | Does not exist |
| F0 | CHIPSet ID |
| F1 | CHIPSet revision level |
| F2 | SuperState V locking |
| F3 | Does not exist |
| F4 | SuperState V timer |
| F5-F7 | Does not exist |
| F8 | ANMI register 0 |
| F9-FF | Does not exist |

## Index Register 05

T-49-17-01

This AT bus command delay defaults to 05. Two bits are selected for each cycle type:

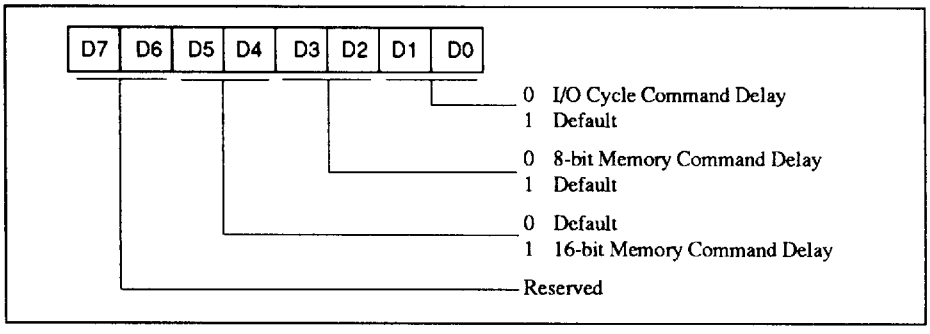
00 0 BLCK delay (command active at falling edge of BALE - default for 16-bit memory)

01 1 BCLK delay (default for all cycles except 16-bit memory)

10 2 BCLK dealy

11 3 BCLK delay

***Index 05*** *AT Bus Command Delay*




*Bits:* 1:0 I/O cycle command delay. Default is 1.

3:2 8-bit memory command delay. Default is 1.

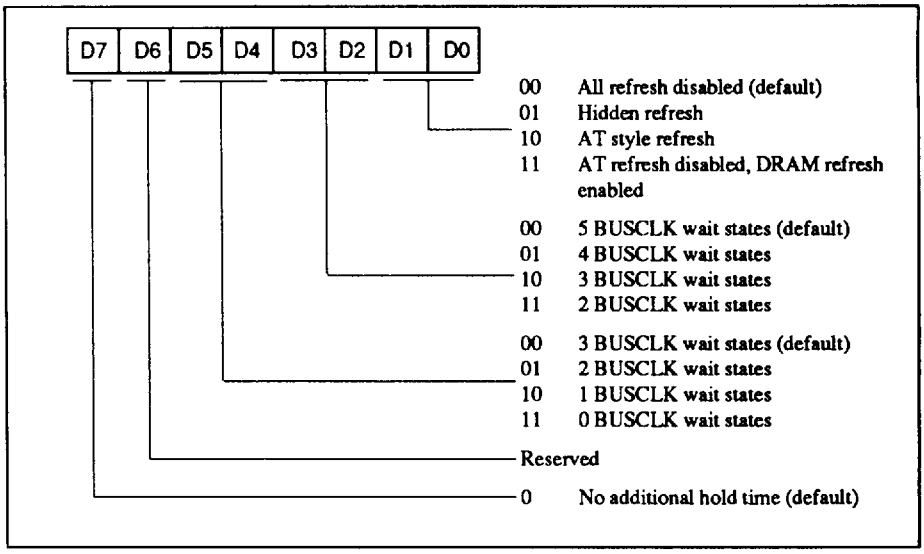
5:4 16-bit memory command delay. Default is 0.

7:6 Reserved. Write as 0 (32-bit AT bus is not implemented). Default is 0.

## Index Register 06

This AT refresh, wait state, and address hold register defaults to 00.

***Index 06*** *AT Refresh, Wait State, and Address Hold*




*Bits:* 1:0 Refresh Mode
- 00 All refresh disabled (default)
- 01 Hidden refresh (requires LS590 for refresh address)
- 10 AT style refresh (puts CPU in HOLD)
- 11 AT refresh disabled, DRAM refresh enabled

3:2 8-bit AT Bus Wait States
- 00 5 BUSCLK wait states (default)
- 01 4 BUSCLK wait states
- 10 3 BUSCLK wait states
- 11 2 BUSCLK wait states

5:4 16-bit AT Bus Wait States
- 00 3 BUSCLK wait states (default)
- 01 2 BUSCLK wait states
- 10 1 BUSCLK wait states
- 11 0 BUSCLK wait states

6 Reserved

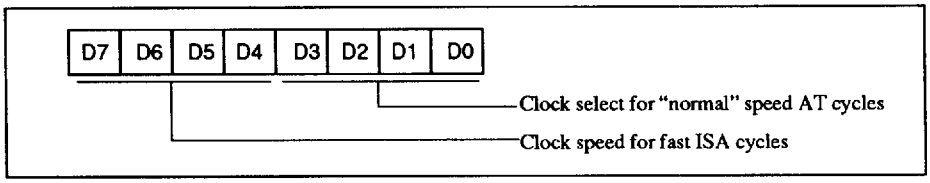
7 AT Bus Address Hold Time (ready delayed by an extra clock after the AT command goes inactive)

## Index Register 07

T-49-17-01

This register selects the ISA bus clock.

***Index 07*** *ISA Bus Clock Selection*




*Bits:* 3:0 Defaults to 00. Clock for "normal" speed AT cycles. This register sets the speed for BCLK, which is 2x BUSCLK. It can be sourced from the separate ATCLOCK pin (16 MHz), CLKIN, or SCLK (in the case that only a 1x clock is available).

| | | | |
|---|---|---|---|
| 0000 | CLKIN/6 | 1000 | (Reserved) |
| 0001 | CLKIN/5 | 1001 | (Reserved) |
| 0010 | CLKIN/4 | 1010 | (Reserved) |
| 0011 | CLKIN/3 | 1101 | ATCLK/3 |
| 0100 | CLKIN/2.5 | 1100 | (Reserved) |
| 0101 | CLKIN/2 | 1101 | ATCLK/2 |
| 0110 | CLKIN/1.5 | 1110 | (Reserved) |
| 0111 | (Reserved) | 1111 | ATCLK/1 |

7:4 Clock speed for Fast ISA cycles. The speed selected is for BCLK during fast cycles, which is 2x FBUSCLK. FBUSCLK switches between the "normal" and "fast" speed on a bus cycle-by-bus cycle basis. A 0010 value should be programmed for CLK2/4. 0000 repeats this speed to allow the register to power up to 0s.

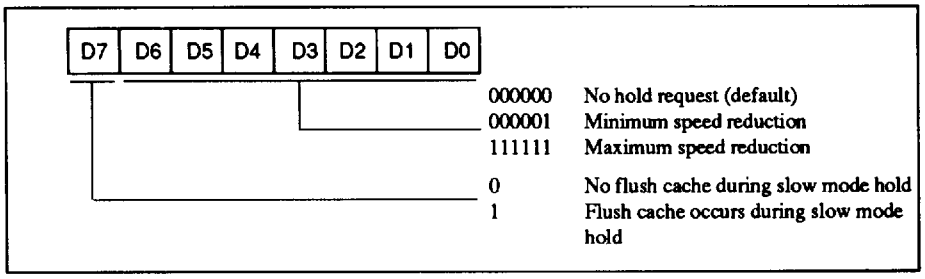
| | | | |
|---|---|---|---|
| 0000 | CLKIN/4 (default) | 1000 | (Reserved) |
| 0001 | (Reserved) | 1001 | (Reserved) |
| 0010 | CLKIN/4 | 1010 | (Reserved) |
| 0011 | CLKIN/3 | 1011 | (Reserved) |
| 0100 | CLKIN/2.5 | 1100 | (Reserved) |
| 0101 | CLKIN/2 | 1101 | ATCLK/2 |
| 0110 | CLKIN/1.5 | 1110 | (Reserved) |
| 0111 | CLKIN/1 | 1111 | ATCLK/1 |

## Index Register 08

T-49-17-01

This register regulates the performance control of the system.

***Index 08*** *Performance Control*




*Bits:* 6:0 Default is set to 00. CPU hold pulse width. These bits set the amount of time in which the CPU is kept in hold following each AT bus refresh. Once the count is set, the mode must be enabled by a separate register or the "turbo" button. The number below, multiplied by 4, is the number of SCLKs (1x CPU clocks) in which the CPU is kept in hold. This occurs about every 15 us.

| | |
|---|---|
| 000000 | No hold request (default) |
| 000001 | Minimum speed reduction |
| 111111 | Maximum speed reduction (dead stop with no means of recovery) |

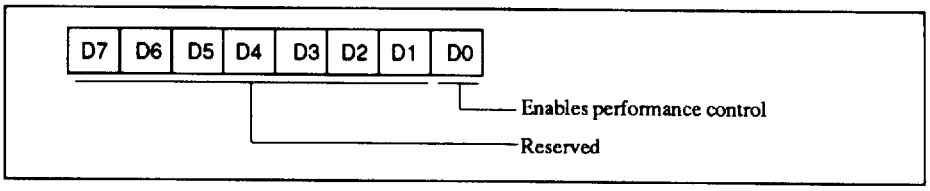
7 486 cache flush. When set to 0, it does not flush cache during slow mode HOLD. When set to 1, the cache flush occurs during each slow mode hold request. Setting this bit to 1 prevents the 486 from running out of internal cache during the slow mode hold.

T-49-17-01

## Index Register 09

This register enables/disables the performance control. The default setting is 00.

**Index 09** *Performance Control Enable*




*Bits:* 0 Deturbo (performance control enable) enables the performance control programmed in Register 0A. This bit is ORed with the other slow mode bit (turbo switch).

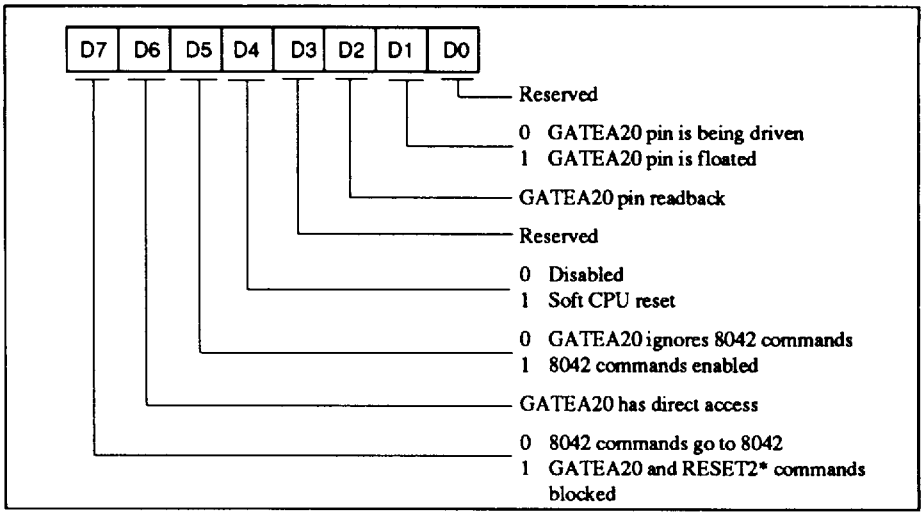
7:1 Reserved

## Index Register 0A

T-49-17-01

This soft reset and GATEA20 register is default set to 00. Note that the soft CPU reset redirection to ANMI is in the SuperState V locking register (F2).

***Index 0A*** *Soft Reset and GATEA20*




*Bits:* 0 Reserved

1 GATEA20 output pin disable
- 0 GATEA20 pin is being driven.
- 1 GATEA20 pin is floated, and used as an input pin. This should be done for 386 CPU when the 8042 GATEA20 function is not being emulated. Also, it should be done to test whether the 8042 is connected directly to this pin.

2 GATEA20 pin readback. This is a read-only bit that reads the status of the GATEA20 pin. It can be used by the BIOS to determine whether the 8042 is connected directly to the 4021 GATEA20 pin.

3 Reserved

4 8042 RESET2 emulation
- 0 Disabled
- 1 The emulated 8042 KBRESET* function will cause a soft CPU reset.

5 8042 GATEA20 emulation
- 0 The emulated 8042 GATEA20 logic ignores 8042 commands.
- 1 8042 commands are enabled to the emulated 8042 GATEA20 logic.

6 Emulated 8042 GATEA20 bit direct access. Reading this bit gives the state of the emulated 8042 GATEA20 bit. Writing sets the value.

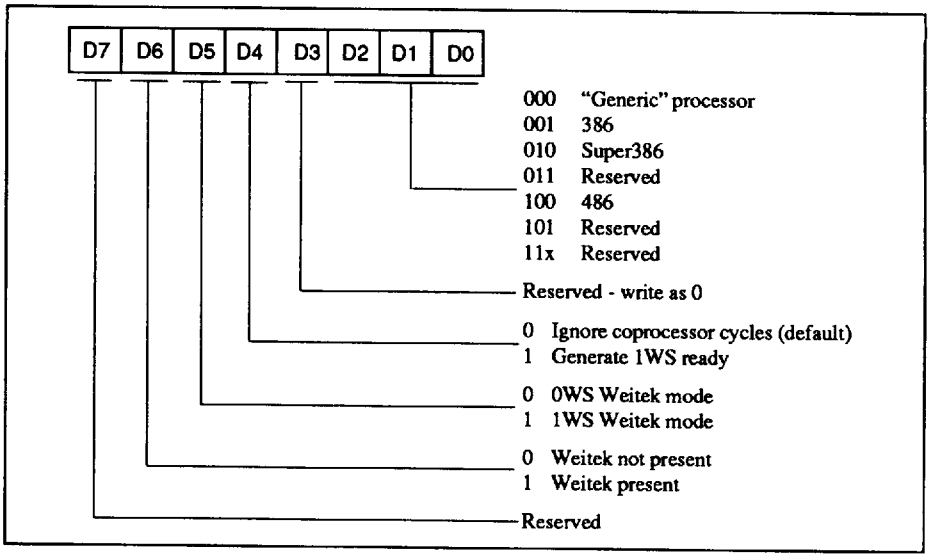
7 Disable IOW* to 8042 for emulated commands.
- 0 All 8042 commands go to the 8042.
- 1 GATEA20 and RESET2 commands to the 8042 have IOW* blocked to speed up the operation of the 8042 commands.

# Index Register 0B

T-49-17-01

This CPU and coprocessor register defaults to 00.

***Index 0B*** *CPU and Coprocessor*




*Bits:* 2:0 Processor type
000 "Generic" processor. Allows system to boot regardless of which processor is installed (default).
001 386
010 Super386
011 Reserved
100 486
101 Reserved
11x Reserved

3 Reserved, write as 0.

4 387 coprocessor ready
0 Ignore coprocessor cycles (default)
1 Generate one wait state ready

5 Weitek mode
0 Zero wait state Weitek mode. A31 passed async to KEN*.
1 One wait state Weitek mode. A31 clocked to KEN*.

6 Weitek present. Currently, this bit is used for the IRQ13 mask. This mask is for the IRQ13 input pin when the processor type is 486. It does not effect the 487 IRQ13 generation from the FERR* pin.
0 Weitek not present (disable IRQ13 for 486 CPU).
1 Weitek present (enable IRQ13).

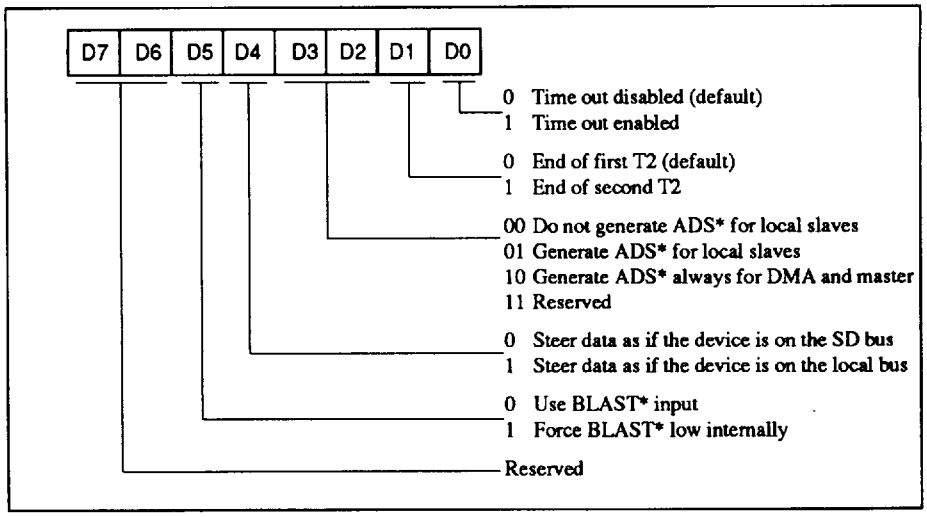
7 Reserved

## Index Register 0C

T-49-17-01

This local bus configuration register defaults to 00.

***Index 0C*** *Local Bus Configuration*




*Bits:* 0 Local bus time-out
- 0 Not time out (default)
- 1 Time out enabled

1 LOCAL* sample point
- 0 End of first T2 (default)
- 1 End of second T2. This delays the start of all ISA bus accesses.

3:2 ADS* generation for DMA or master accesses to local peripherals.
- 00 Do not generate ADS* for local slaves.
- 01 Generate ADS* for local slaves and sample RDY* and BRDY*.
- 10 Generate ADS* always for DMA and master.
- 11 Reserved

4 Data steering for local bus peripherals.
- 0 Steer data as if the device is on the SD bus.
- 1 Steer data as if the device is on the local bus.

5 Force BLAST* low when a local bus master is in control. This avoids the need to have the local master drive it low. BLAST* is always forced low when any CPU other than the 486 is selected.
- 0 Use BLAST* input in the same way as when the 486 is in control.
- 1 Force BLAST* low internally.

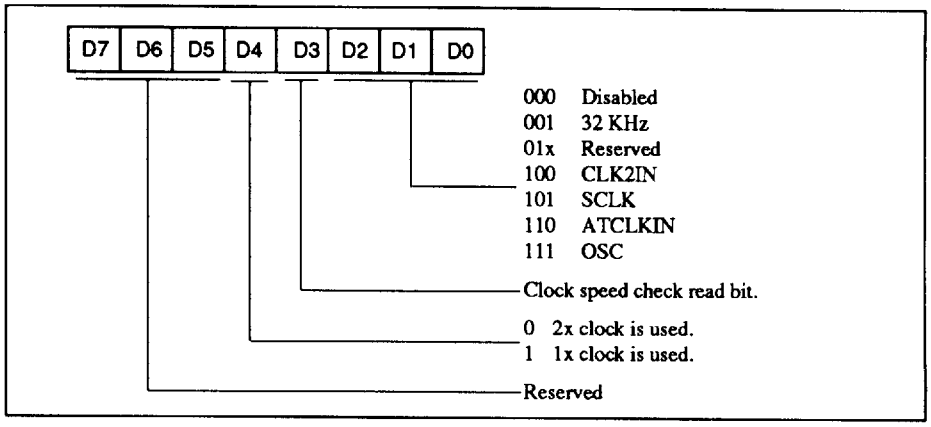
7:6 Reserved

## Index Register 0D

T-49-17-01

This clock speed check and mode register defaults to 00.

***Index 0D*** *Clock Speed Check and Clock Status*




*Bits:* 2:0 Clock speed check select. These bits select which clock is used for a speed check.
- 000 Disabled (Port 92 bit 7 reads as 0)
- 001 32 KHz (one flip-flop divider)
- 01x Reserved
- 100 CLK2IN
- 101 SCLK
- 110 ATCLKIN
- 111 OSC (14.31818)

3 CLock speed check read bit. This bit is the output of the divider chain which is an 11-bit ripple counter.

4 SEL1X status. This bit reads the status of the SEL1X latch which is sampled at RESET. Writes are ignored.
- 0 2x clock is used.
- 1 1x clock is used.

7:5 Reserved

## Index Register 0E-0F

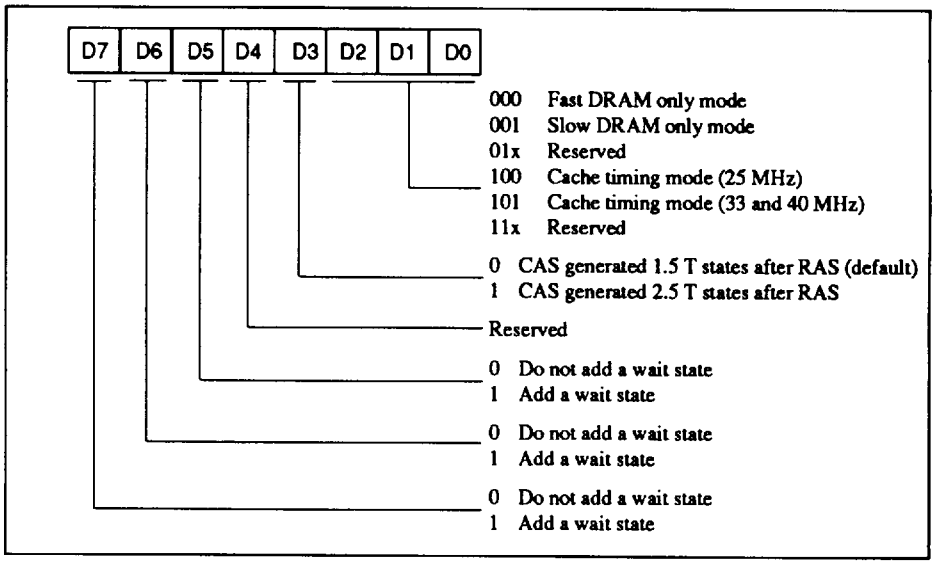
These index registers do not exist.

## Index Register 10

T-49-17-01

This DRAM timing register defaults to 00.

***Index 10*** *DRAM Timing*




*Bits:* 2:0 Timing Mode. 000 and 001 can only be used in 2x clock mode.
- 000 Fast DRAM only mode (3-1-1-1)
- 001 Slow DRAM only mode (3-2-2-2)
- 01x Reserved
- 100 Cache timing mode (4-2-2-2) (25 MHz)
- 101 Cache timing mode (5-3-3-3) one extra wait state (33 & 40 MHz)
- 11x Reserved

3 RAS to CAS timing in DRAM only mode.
- 0 CAS generated 1.5 T states after RAS (default).
- 1 CAS generated 2.5 T states after RAS (for slower RAMs).

4 Reserved

*Bits:* 5 Refresh wait state. Add a wait state for refresh cycles. Used at low CPU clock frequencies when local DRAM staggered refresh cannot complete in the time provided by a standard ISA refresh. This will add a wait state to standard refresh cycles when Classic Refresh is enabled, and all master cycles. It will not effect hidden refresh.

T-49-17-01

0 Do not add a wait state

1 Add a wait state



6 Master read wait state. Add a wait state for ISA master reads from local DRAM. Used at low CPU clock frequencies and when Wingine is enabled.

0 Do not add a wait state.

1 Add a wait state.

7 Master write wait state. Add a wait state for ISA master writes to local DRAM. Used at very low CPU clock frequencies and when Wingine is enabled.

0 Do not add a wait state.

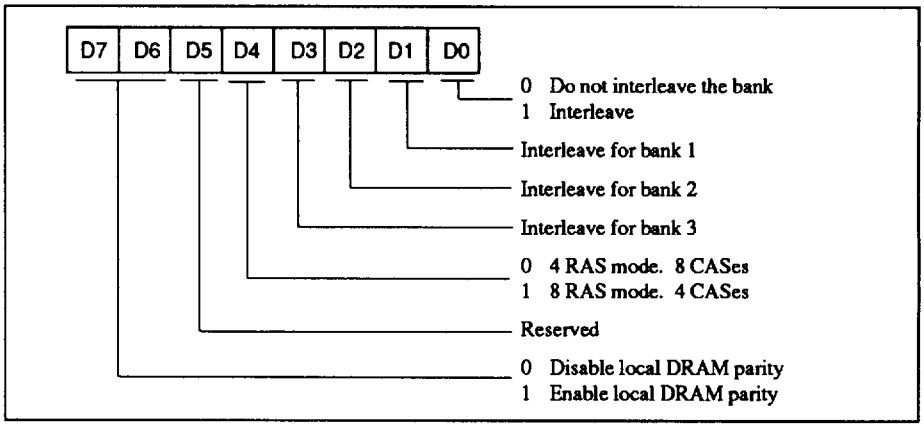
1 Add a wait state.

## Index Register 11

T-49-17-0

This DRAM setup register defaults to 00.

***Index 11*** *DRAM Setup*




*Bits:* 0 Interleave for bank zero. If a bank is interleaved, its address range is doubled, and it is active only when the interleave bit (A11 or A12) compares. Banks 0 and 2 compare the interleave bit to 0. Bnaks 1 and 3 compare it to 1. Two banks must be interleaved at the same address for proper operation. See *Chapter 6: The DRAM Controller*, the subsection titled *Page Interleaving* for more information on what must be done to interleave properly.

0 Do not interleave the bank.
1 Interleave.

1 Interleave for bank 1.

2 Interleave for bank 2.

3 Interleave for bank 3.

4 DRAM decode mode.

0 4 RAS mode. 8 CASes.
1 8 RAS mode. 4 CASes.

6:5 Reserved

7 Enable local DRAM parity. This bit provides an additional way to disable parity over and above I/O Port 61. Both this bit and I/O Port 61 bit 2 must be enabled (Port 61 must be set to 0). The NMI mask must also be set to send the NMI to he CPU (I/O Port 70 bit 7).

0 Disabled
1 Enabled

T-49-17-01

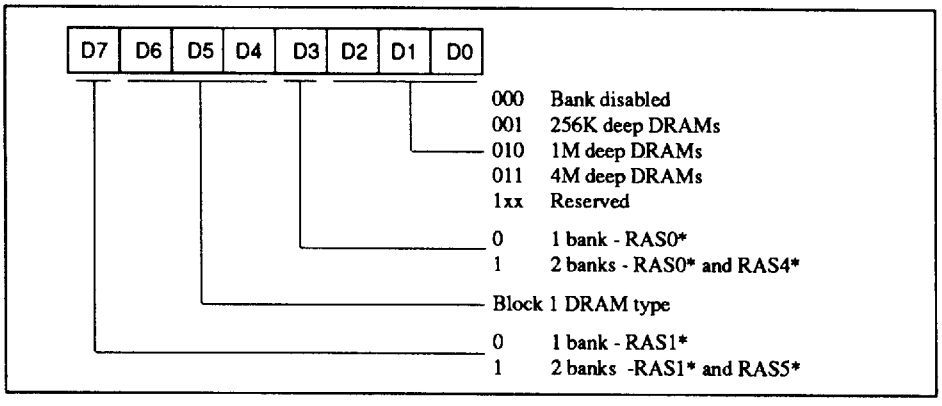
## Index Register 12

This DRAM configuration register regulates the settings for blocks 0 and 1, and defaults to 00.

4 RAS mode: Block 0 uses RAS0*.
Block 1 uses RAS1*.

8 RAS mode: Block 0 uses RAS0* and RAS4*.
Block 1 uses RAS1* and RAS5*.

***Index 12*** *DRAM Configuration Blocks 0 and 1*




*Bits:* 2:0 Block 0 DRAM type.
- 000 Bank disabled
- 001 256K deep DRAMs (256Kx1 or 256Kx4)
- 010 1M deep DRAMs (1Mx1 or 1Mx4)
- 011 4M deep DRAMs (4Mx1 or 4Mx4)
- 1xx Reserved

3 Number of banks installed in block 0.
- 0 1 bank - RAS0*
- 1 2 banks - RAS0* and RAS4*. Available in 8 RAS mode only.

6:4 Block 1 DRAM type. Bit definitions same as block 0.

7 Number of banks installed in block 1.
- 0 1 bank - RAS1*
- 1 2 banks - RAS1* and RAS5*. Available in 8 RAS mode only.

## Index Register 13

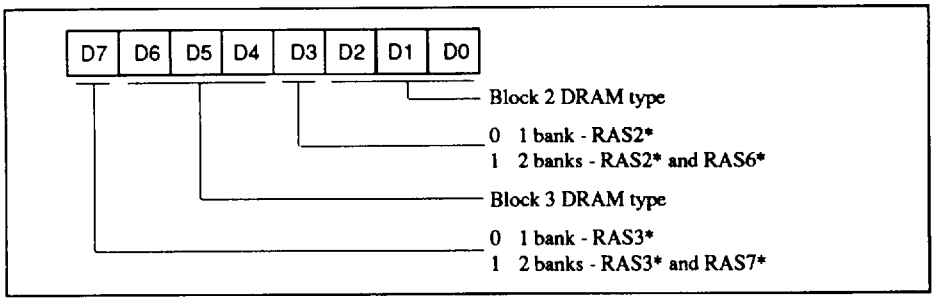
T-49-17-01

This DRAM configuration register regulates the settings for blocks 2 and 3, and defaults to 00.

4 RAS mode: Block 2 uses RAS2*.
Block 3 uses RAS3*.

8 RAS mode: Block 2 uses RAS2* and RAS6*.
Block 3 uses RAS3* and RAS7*.

***Index 13*** *DRAM Configuration Blocks 2 and 3*




*Bits:* 2:0 Block 2 DRAM type. Bit definitions same as block 0.

3 Number of banks installed in block 2.
0 1 bank - RAS2*
1 2 banks - RAS2* and RAS6*. Available in 8 RAS mode only.

6:4 Block 3 DRAM type. Bit definitions same as block 0.

7 Number of banks installed in block 3.
0 1 bank - RAS3*
1 2 banks - RAS3* and RAS7*. Available in 8 RAS mode only.

T-49-17-01

## Index Registers 14 through 17

These DRAM configuration registers regulate the settings for blocks 0-3, and the default settings are 00.

*Index:*
- 14 DRAM block 0 starting address.
- 15 DRAM block 1 starting address.
- 16 DRAM block 2 starting address.
- 17 DRAM block 3 starting address.

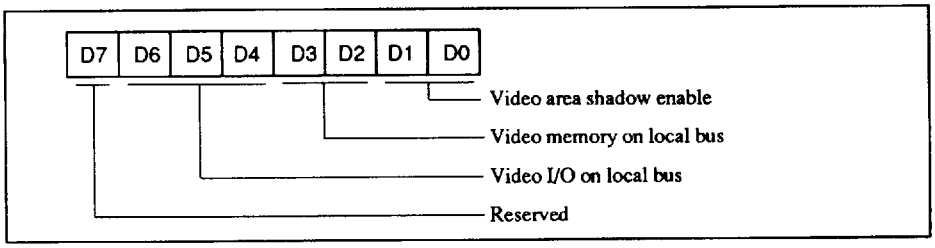
*Bits:*
- 6:0 A26-A20 of starting address
- 7 Reserved

T-49-17-01

## Index Register 18

This video area shadow and local bus control register defaults to 00.

***Index 18*** *Video Area Shadow and Local Bus Control*




*Bits:* 1:0 Video area shadow enable. If a bit is 0, accesses from that location come from the ISA bus. If a bit is 1, accesses come from local DRAM. These bits, when set to 1, override bits 2 and 3. For the 38605, this memory can be used as SuperState RAM. Take care when these bits are set to 1 that the same memory area is not used for two things at once.
Bit 0 A0000-AFFFF shadow enable
Bit 1 B0000-BFFFF shadow enable

3:2 Video memory on local bus. Setting these bits to 1 forces the LOCAL* pin internally for the address range. This function can be used even if the LOCAL* pin is disabled. Turning on shadow RAM for this area overrides these bits.
Bit 2 A0000-AFFFF
Bit 3 B0000-BFFFF

6:4 Video I/O on local bus. Setting these bits to 1 forces the LOCAL* pin for I/O accesses to the specified ranges. The MDA, CGA, and palette DAC areas are controlled separately. Setting either the MDA or CGA bits will cause the middle space (EGA) to be on the local bus.
Bit 4 MDA and EGA areas (03B0-03BB, 3Bf, 03C0-03C5, 03CA-03CF).
Bit 5 CGA and EGA areas (03D0-03DF, 03C0-03C5, 03CA-03CF).
Bit 6 Palette DAC area (03C6-03C9).

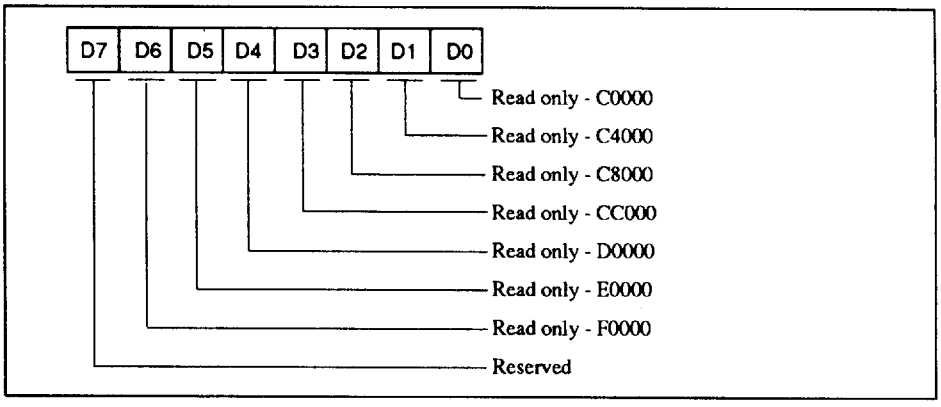
7 Reserved

## Index Register 19

T-49-17-01

This DRAM shadow read enable register defaults to 00. If a bit is set to 0, reads from that location come from the ISA bus. If a bit is set to 1, reads from that location come from the local DRAM. The default is all OFF.

***Index 19*** *DRAM Shadow Read Enable*



T-19-17-01

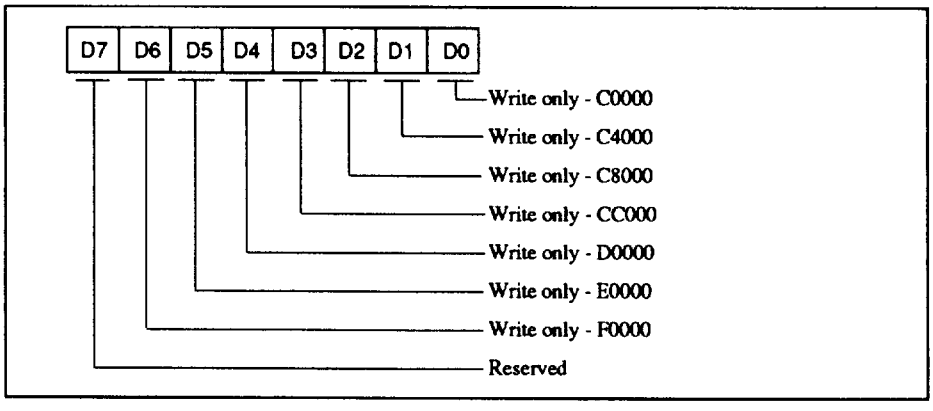
## Index Register 1A

This DRAM shadow write enable register defaults to 00. If a bit is set to 0, writes to that location go to the ISA bus. If a bit is set to 1, writes to that location go to the local DRAM. The default is all OFF.

The bit assignments for both registers are:

| | |
|---|---|
| 0 | C0000-C3FFF |
| 1 | C4000-C7FFF |
| 2 | C8000-CBFFF |
| 3 | CC000-CFFFF |
| 4 | D0000-DFFFF |
| 5 | E0000-EFFFF |
| 6 | F0000-FFFFF |
| 7 | Reserved |

***Index 1A*** *DRAM Shadow Write Enable*

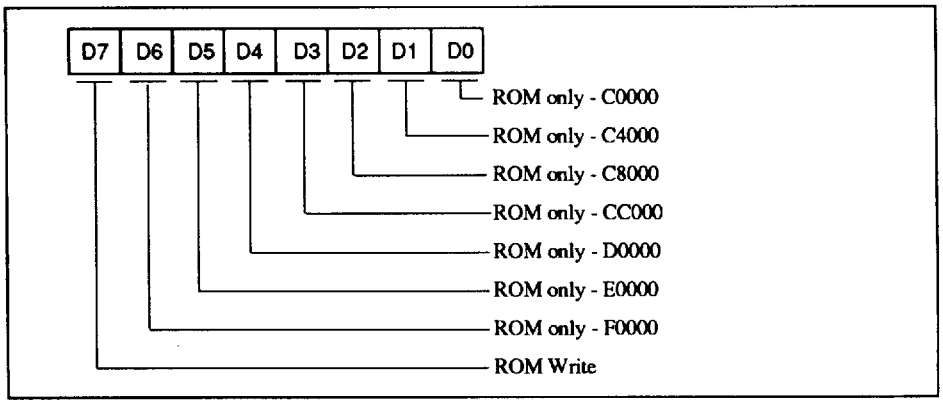


## Index Register 1B

This ROMCS enable register defaults to 60. If a bit is set to 1, ISA bus reads from that location activate ROMCS*. If a bit is set to 0, ROMCS* is not activated. ROMCS* will not be activated if an access is directed to local DRAM.

***Index 1B*** *ROMCS Enable*

T-49-01



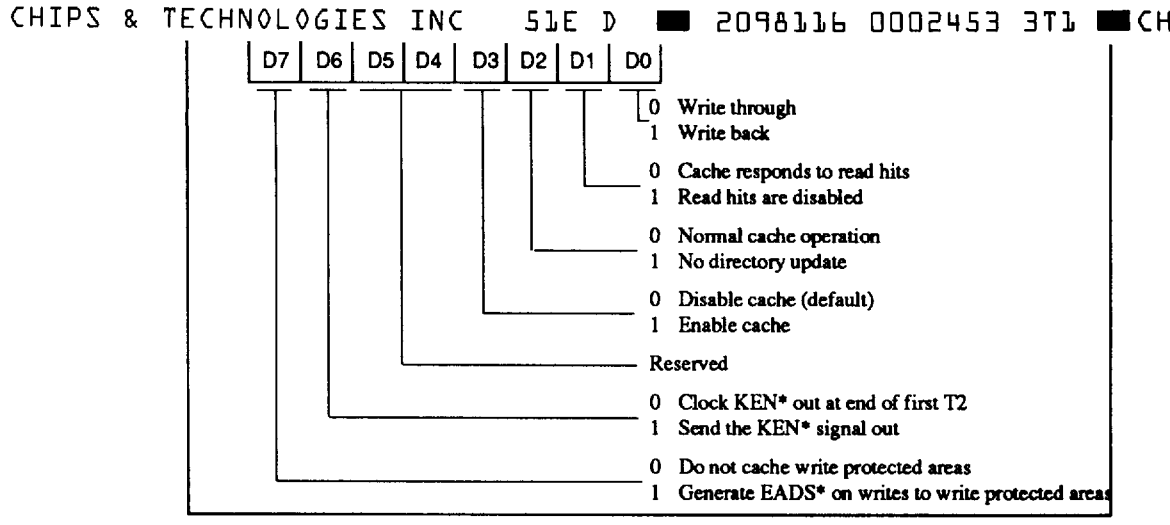

## Index Register 1C-1F

These index registers do not exist.

# Index Register 20

This cache information register is set to the default setting of 00. T-49-17-01

***Index 20*** *Cache Information I*




*Bits:* 0 Write mode: selects the write-hit policy.
- 0 Write through
- 1 Write back

1 Initialize cache. All cacheable memory read cycles update the cache. No castouts are performed. Write-hit cycles are executed as write through. This mode should only be invoked when the cache is disabled.
- 0 Cache responds to read hits
- 1 Read hits are disabled

2 Freeze cache directory
- 0 Normal cache operation
- 1 No directory update

3 Cache enable
- 0 Disable cache (default)
- 1 Enable cache

5:4 Reserved

6 Local bus slave speed support
- 0 Clock KEN* signal (local DRAM indicator portion)
- 1 Send the KEN* signal (local DRAM indicator portion) out asynchronously during the first T2. Allows one wait state non-cached local bus slaves.

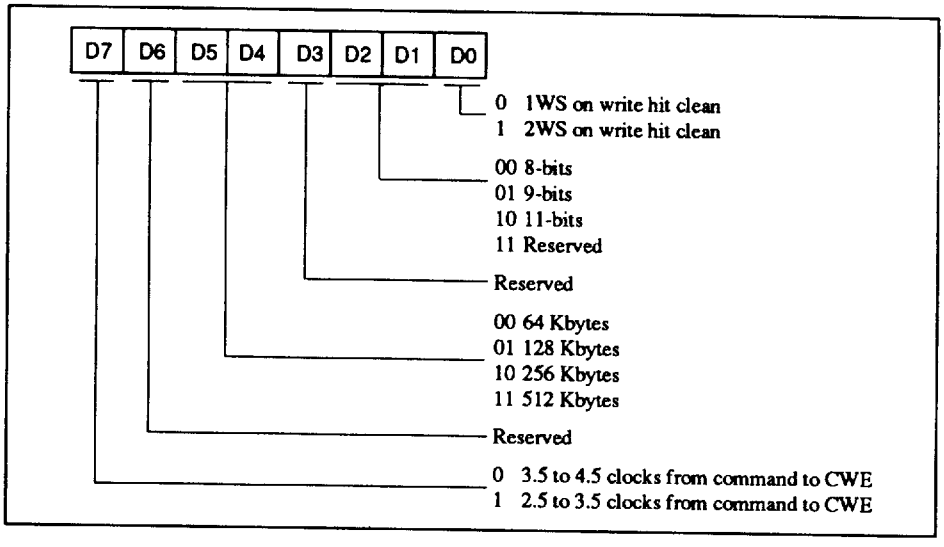
7 Write protect method
- 0 Do not cache write protected areas in the 486 (KEN* goes high for the fourth transfer of a burst).
- 1 Generate EADS* on writes to write protected areas. Allow them to be cached in the 486.

## Index Register 21

T-49-17-01

This register also provides cache information and is set to a default setting of 00.

***Index 21*** *Cache Information II*




*Bits:* 0 TAG write timing
- 0 One wait state on write hit clean (short tag write).
- 1 Two wait states on write hit clean (long tag write).

2:1 Tag width
- 00 8-bits (default)
- 01 9-bits
- 10 11-bits
- 11 Reserved

3 Reserved, write as zero.

5:4 Cache size
- 00 64 Kbytes (default)
- 01 128 Kbytes
- 10 256 Kbytes
- 11 512 Kbytes

6 Reserved, write as zero.

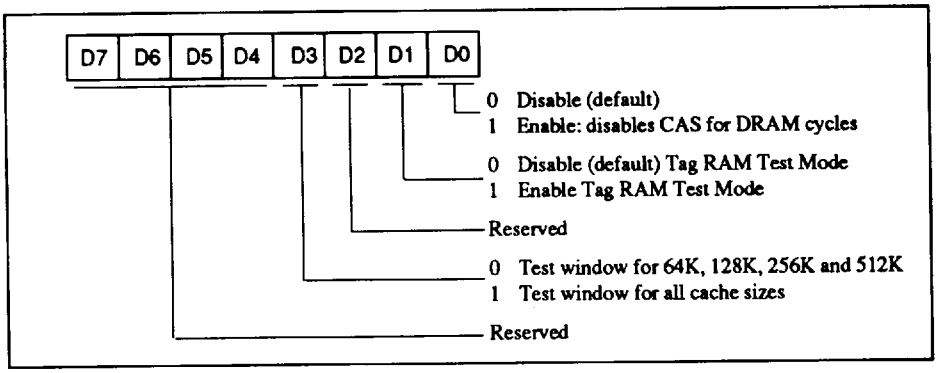
7 CWE* timing during DMA/master cycles.
- 0 3.5 to 4.5 clocks from command to CWE (25/33 MHz).
- 1 2.5 to 3.5 clocks from command to CWE (16/20 MHz).

## Index Register 22

T-49-17-01

This cache testing control register is set to a default setting of 00.

***Index 22*** *Cache Testing Control*




*Bits:* 0 Data SRAM test mode
- 0 Disable (default)
- 1 Enable: disables CAS for DRAM cycles within the test window

1 Tag RAM test mode
- 0 Disable (default)
- 1 Enable: disables CAS for DRAM cycles within the test window

2 Reserved

3 SRAM and TAG RAM test window select.
- 0 Test window for 64K, 128K, and 256K cache: 40000-7FFFF; for 512K cache: 20000-9FFFF.
- 1 Test window for all cache sizes: 100000-17FFFF (at 1M point).

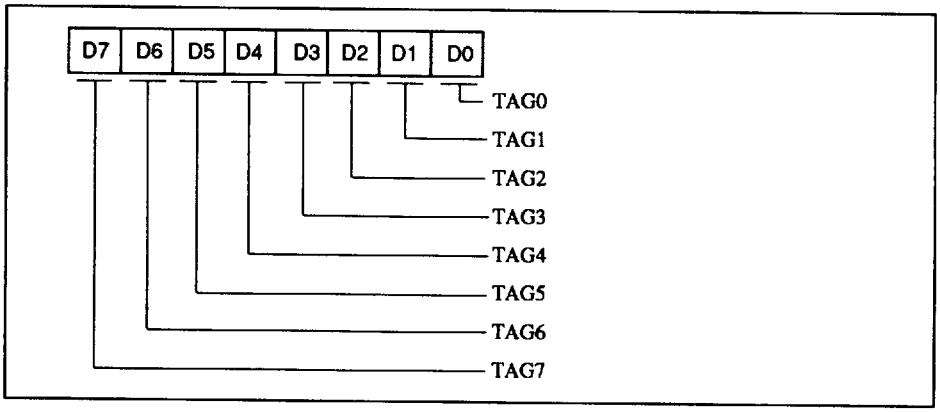
7:4 Reserved

## Index Register 23

T-49-17-01

This data port register defines the tag testing settings for bits 0-7. It has a default setting of 00. This register is for internal comparator mode only. When tag test mode is enabled, writes to the test window cause the value in this register to be written to external TAG SRAM bits 0-7. Reads from the test window cause TAG bits 0-7 to be read from the external TAG SRAM into this register.

***Index 23*** *Data Port Tag Testing Bits 0-7*



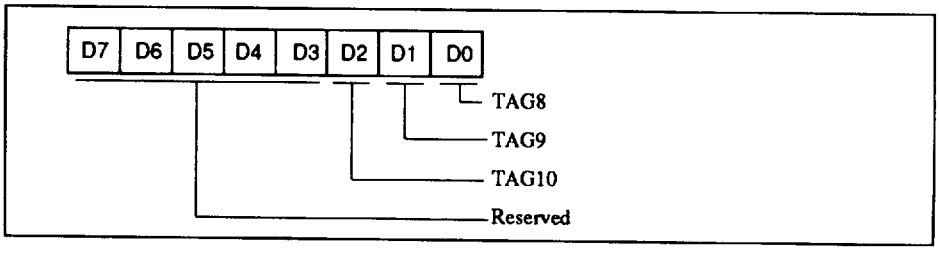

## Index Register 24

This is another data port register that defines the tag testing settings for bits 8-10. It has a default setting of 00. See the above description for Index Register 23 for an explanation of the register.

0:2 TAG bits 8-10

3:7 Reserved

***Index 24*** *Data Port Tag Testing Bits 8-10*



## Index Register 25 and 26

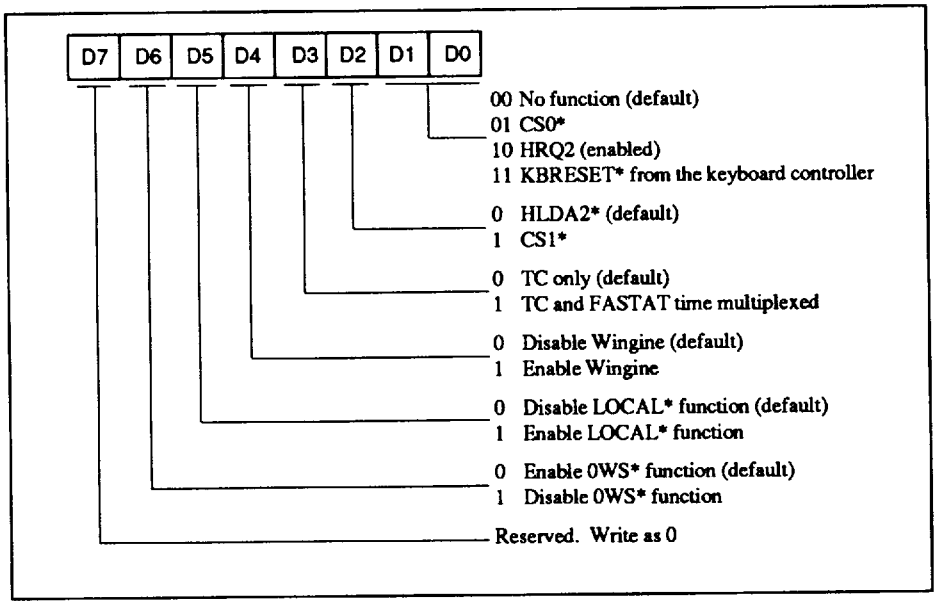
T-49-17-01

These index registers do not exist.

## Index Register 27

This pin configuration register has a default setting of 00.

***Index 27*** *Pin Configuration*




*Bits:* 1:0 HRQ2/CS0* pin.
- 00 No function (masked off HRQ2) - default
- 01 CS0*
- 10 HRQ2 (enabled)
- 11 KBRESET* from the keyboard controller

2 HLDA2*/CS1* pin
- 0 HLDA2* (default)
- 1 CS1*

3 TC/FASTAT signal
- 0 TC only (default)
- 1 TC and FASTAT time multiplexed

*Bits:* 4 Windows video
- 0 Disable Wingine: SMEMR* and SMEMW* are driven (default).
- 1 Enable Wingine: XREQ* and VRAS* pins replace the above.

5 Enable LOCAL* function
- 0 Disable LOCAL* function (default).
- 1 Enable LOCAL* function.

T-49-17-01

6 Disable 0WS* function
- 0 Enable 0WS* function (default).
- 1 Disable 0WS* function.

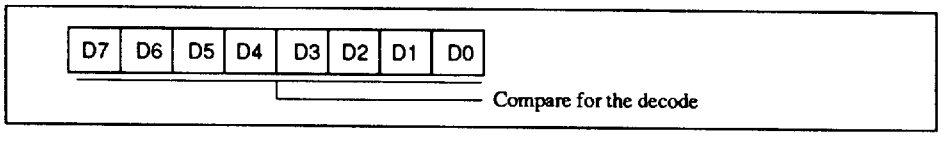
7 Reserved, write as zero.

CHIPS & TECHNOLOGIES INC 51E D

## Index Register 28

This I/O decode #0 address low register defaults to 00.

***Index 28*** *I/O Decode #0 Address Low*

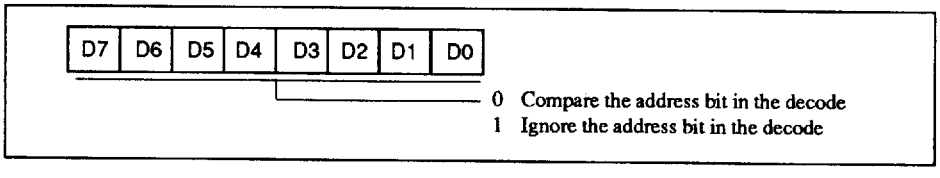



*Bits:* 7:0 Address bits 7:0 are to be compared for the decode.

## Index Register 29

This Chip select #0 mask low register defaults to 00.

***Index 29*** *Chip Select #0 Mask Low*



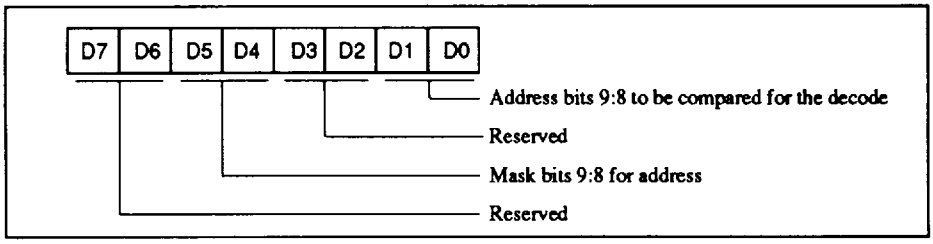

*Bits:* 7:0 Mask bits 7:0 for address.
- 0 Compare the address bit in the decode
- 1 Ignore the address bit in the decode

## Index Register 2A

T-49-17-01

This I/O decode #0 address and mask high register defaults to 00.

***Index 2A*** *I/O Decode #0 Address and Mask High*




*Bits:* 1:0 Address bits 9:8 to be compared for the decode.

3:2 Reserved

5:4 Mask bits 9:8 for address.

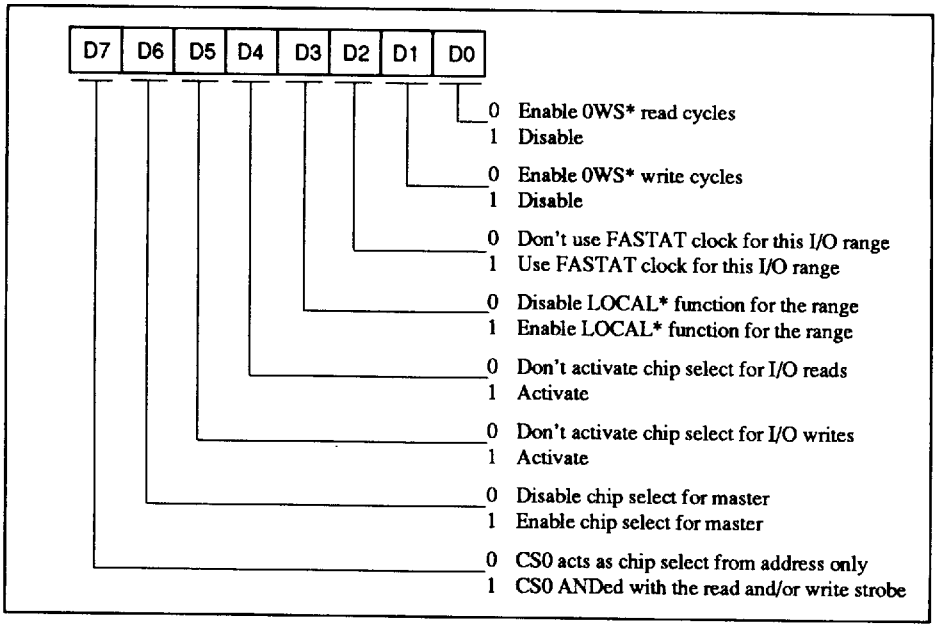
7:6 Reserved

# Index Register 2B

T-49-17-01

This I/O decode #0 configuration register defaults to 00.

***Index 2B*** *I/O Decode #0 Configuration*




*Bits:* 0 Force 0WS* for read cycles
- 0 Enable
- 1 Disable

1 Force 0WS* for write cycles
- 0 Enable
- 1 Disable

2 FASTAT clock
- 0 Don't use FASTAT clock for this I/O range
- 1 Use FASTAT clock for this I/O range

3 Force local bus I/O range
- 0 Disable LOCAL* function for the range
- 1 Enable LOCAL* function for the range

4 Activate chip select for I/O reads with limitations
- 0 Don't activate
- 1 Activate

5 Activate chip select for I/O writes with limitations
- 0 Don't activate
- 1 Activate

*Bits:* 6 Allow chip select for ISA master accesses
- 0 Disable chip select for ISA master
- 1 Enable chip select for ISA master

T-49-17-01

7 Chip select or strobe
- 0 CS0 acts as a chip select decoded from address only
- 1 CS0 is ANDed with the read and/or write strobe



## Index Register 2C through 2F

Index 2C is an I/O decode #0 address low register that defaults to 00. See Index Register 28 for a description of the CS0 bits.

Index 2D is a chip select #1 mask low register that defaults to 00. See Index Register 29 for a description of the CS0 bits.

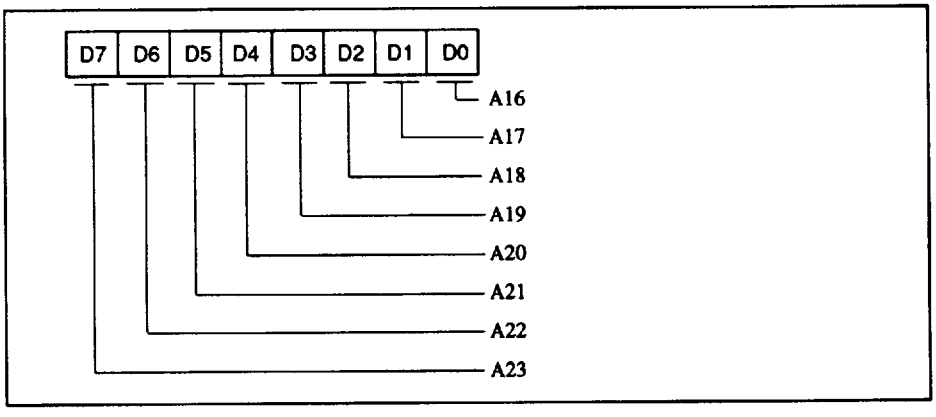
Index 2E is an I/O decode #1 address and mask high register that defaults to 00. See Index Register 2A for a description of the CS0 bits.

Index 2F is an I/O decode #1 configuration register that defaults to 00. See Index Register 2B for a description of the CS0 bits.

## Index Register 30

This memory decode #0 address low register defaults to 00.

***Index 30*** *Memory Decode #0 Address Low*

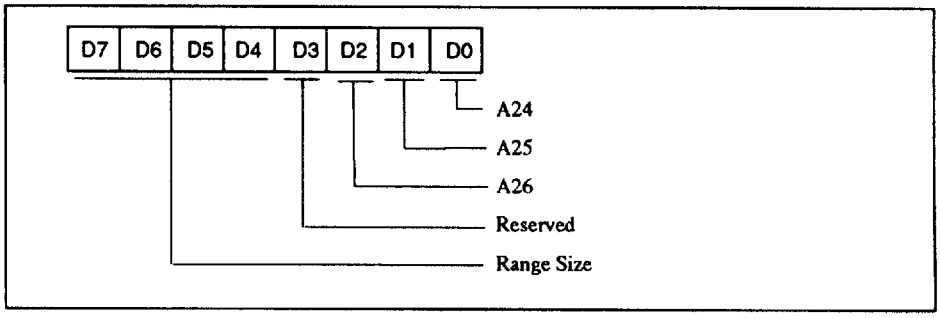


## Index Register 31

T-49-17-01

This memory decode #0 address high block size register defaults to 00.

***Index 31*** *Memory Decode #0 Address High Block Size*




*Bits:* 2:0 Address bits 26:24

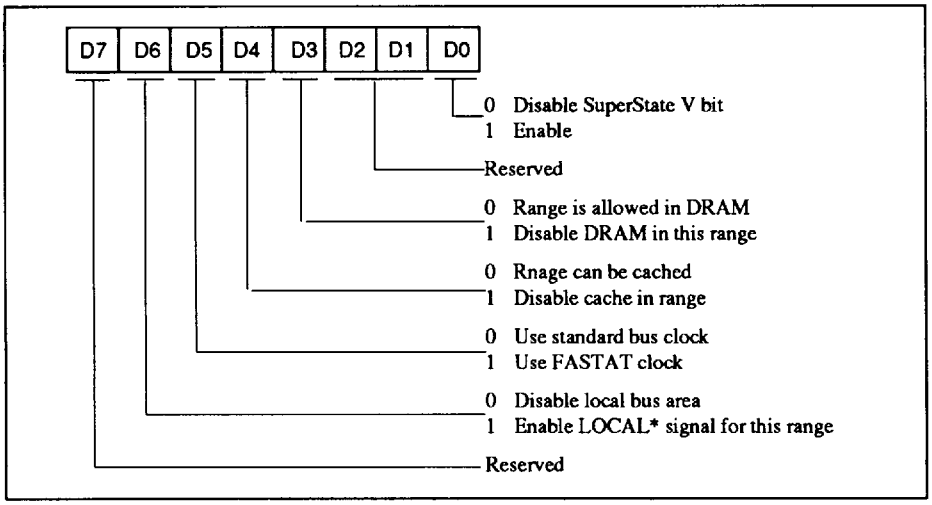
3 Reserved

7:4 Range size
0000 64K byte range
0001 128K byte range
0010 256K byte range
0011 512K byte range
0100 1Mbyte range
0101 2Mbyte range
0110 4Mbyte range
0111 8Mbyte range
1000 16Mbyte range
1001 32Mbyte range
1010 64Mbyte range
1011 Reserved - all memory is reserved for SuperState V
11xx Reserved

## Index Register 32

This memory decode #0 configuration register defaults to 00.

***Index 32*** *Memory Decode #0 Configuration*




*Bits:* 0 SuperState V bit
- 0 Disabled
- 1 DRAM at this address is SuperState V memory. When this bit is one, the "Hole in DRAM" (which should also be set) is ignored for SuperState V accesses. The effect of this is that the DRAM is accessible only from SuperState V.

2:1 Reserved

3 Hole in DRAM
- 0 Range is allowed in DRAM
- 1 DRAM within this range is disabled. The access will go to the ISA bus unless claimed by a local slave.

4 Non-cacheable range
- 0 Range can be cached.
- 1 Disable cache in range.

5 FASTAT clock (if cycle goes to AT bus)
- 0 Use standard bus clock.
- 1 Use FASTAT clock.

*Bits:* 6 Local bus area

T-49-17-01

0 Disabled.

1 Force the LOCAL* signal for this range. Accesses to this range will be on the local bus. Setting this bit will NOT prevent a local DRAM access. The memory range must be either outside of the local DRAM decode or bit 3 above must also be set.

7 Reserved



## Index Registers 33 through 35

Index 33 is a memory decode #1 address low register that defaults to 00.

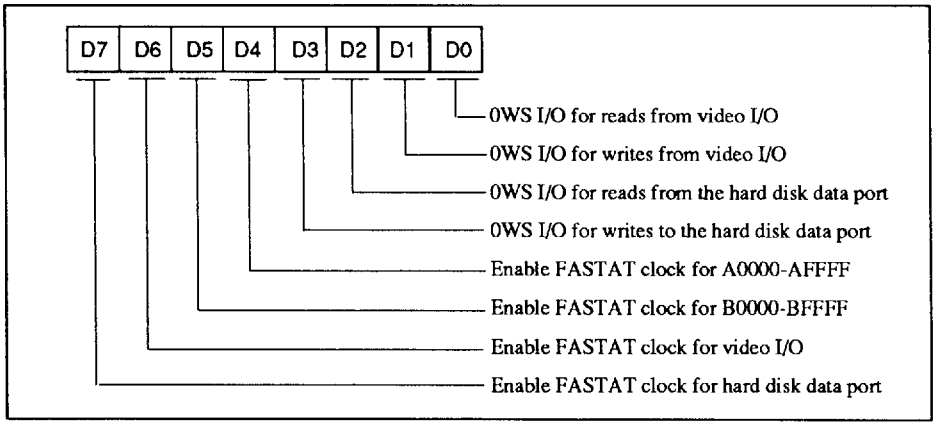
Index 34 is a memory decode #1 address high block size register that defaults to 00.

Index 35 is a memory decode #1 configuration register that defaults to 00.

## Index Register 36

This FASTAT function defaults to 00. Bits 0-3 determine whether the zero wait state I/O function is enabled for the given function. If enabled, accesses will have no command delay and zero wait state will be forced internally providing a command length of one BUSCLK (the same as a zero wait state 16-bit memory cycle).

***Index 36*** *FASTAT Function*



T-49-17-01

*Bits:* 0 Zero wait state I/O for reads from video I/O.

1 Zero wait state I/O for writes to video I/O.

2 Zero wait state I/O reads from the hard disk data port.

3 Zero wait state I/O writes to the hard disk data port.

Bits 4-7 determine whether the FASTAT clock is used for the given memory or I/O range.

0 Use normal AT clock

1 Use FASTAT clock

If a programmable decode happens to be programmed for the same range, the bits are ORed (either can force FASTAT).

4 Enable FASTAT clock for A0000-AFFFF.

5 Enable FASTAT clock for B0000-BFFFF.

6 Enable FASTAT clock for video I/O.

7 Enable FASTAT clock for hard disk data port.

## Index Register E0

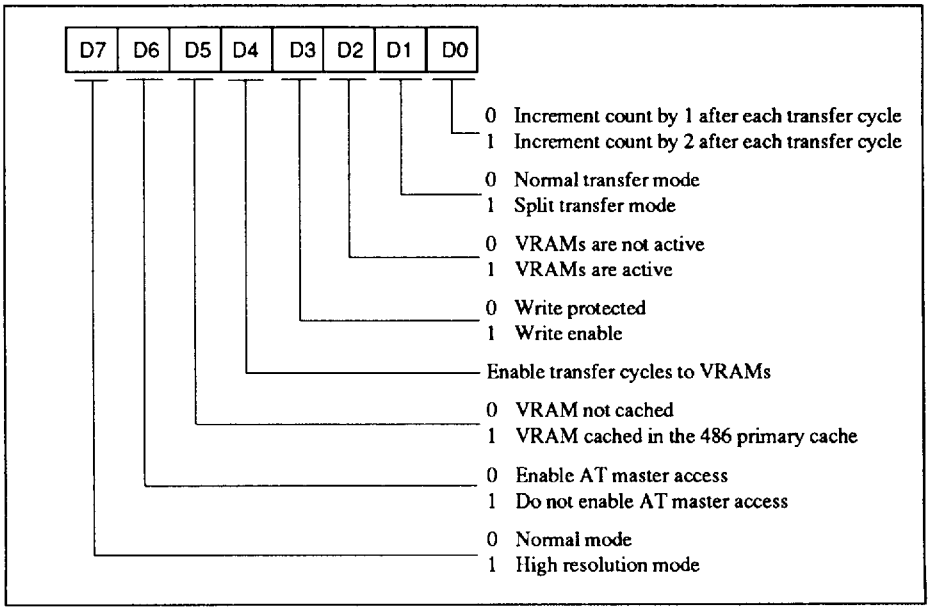
CHIPS & TECHNOLOGIES INC 51E D

This VGA Wingine control port defaults to 00. This register is simply an 8-bit read and write configuration register in the 4021. It is write-only in the VGA chip and exists here for read back purposes. The individual bits are defined, but are not listed here because they pertain only to the VGA chip.

## Index Register E2

This Wingine mode register defaults to 00.

***Index E2*** *Wingine Mode Register*



T-49-17-01

*Bits:* 0 Transfer count increment value.
- 0 Increment count by 1 after each transfer cycle.
- 1 Increment count by 2 after each transfer cycle.

1 Transfer cycle mode.
- 0 Normal mode
- 1 Split transfer mode

2 Enable VRAMs for refresh, etc.
- 0 VRAMs are not active
- 1 VRAMs are active

3 Enable writes to the VRAMs. The driver can choose to enable writes only when the driver itself is running, thus protecting it from other software. This avoids the need of having a gap between main memory and Wingine memory.
- 0 Write protected
- 1 Write enable



4 Enable transfer cycles to VRAMs. If this bit is not set, the VRAM can still be accessed as normal memory, but there will be no video. When this bit is enabled, there is a slight CPU performance penalty due to the overhead of the transfer cycles. In addition, ISA master accesses to local DRAM will be synchronized before starting the DRAM cycle. This will require a wait state to be added to the master cycle.
- 0 Totally hidden video mode
- 1 Enable transfer cycles

5 Cache VRAM. VRAMs are only cached in the 486 primary cache.
- 0 VRAM not cached.
- 1 VRAM cached in the 486 primary (internal) cache.

6 Enable AT master access to VRAM.
- 0 Enable AT master access.
- 1 Do not enable AT master access.

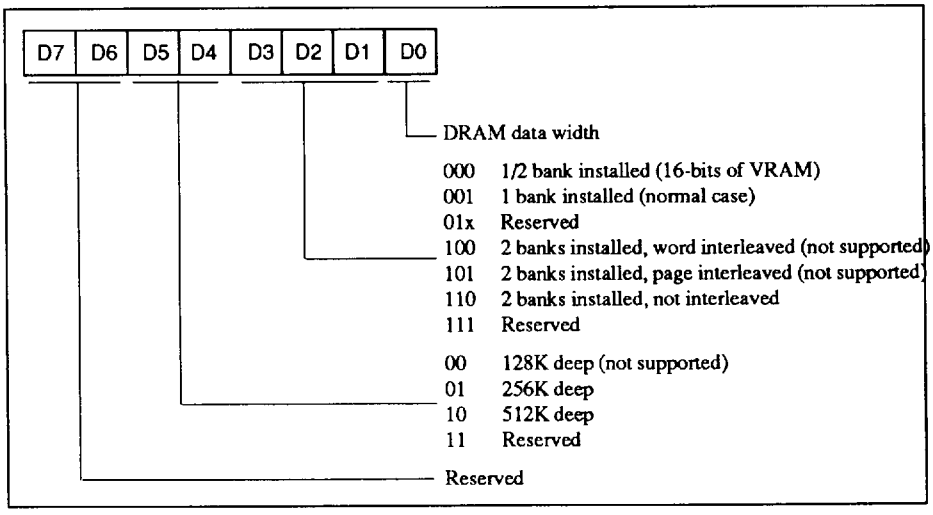
7 Transfer counter shifting. Specifies how the output of the transfer counter is mapped to the VRAMs.
- 0 Normal mode (CNT0 - Col 8, CNT1 - Row 8, etc.)
- 1 High resolution (1280x1024) mode (CNT0-> Col 8, CNT1-> Row 1, etc.)

T-49-17-01

## Index Register E3

This Wingine VRAM configuration register defaults to 00. This information has the same definition in all CHIPSets. The main purpose is to provide information to the Wingine driver as to the VRAM configuration. In the 4021, the only other purpose it serves is in determining the VRAM decode size (1 Mbyte or 2 Mbytes).

***Index E3*** *Wingine VRAM Configuration Register*




*Bits:* 0 DRAM data width. The BIOS should always write this bit as one, indicating 32-bit.

3:1 VRAM bank configuration. Only two configurations are valid in the 4021. Bit 1 is read and write; it is written by the BIOS with the current configuration. Bits 2 and 3 always read as zeros. Not all combinations of bits 5:1 are supported. See the section on Wingine for more information.

- 000 1/2 bank installed (16-bits of VRAM)
- 001 One bank installed (the normal case)
- 01x Reserved
- 100 Two banks installed, word interleaved (not supported)
- 101 Two banks installed, page interleaved (not supported)
- 110 Two banks installed, non-interleaved
- 111 Reserved

5:4 VRAM size. Only 256K and 512K are valid in the 4021. These bits are used with the VRAM bank configuration bits to indicate the size of the VRAM decode (1M or 2M) and provide information to the Wingine driver.

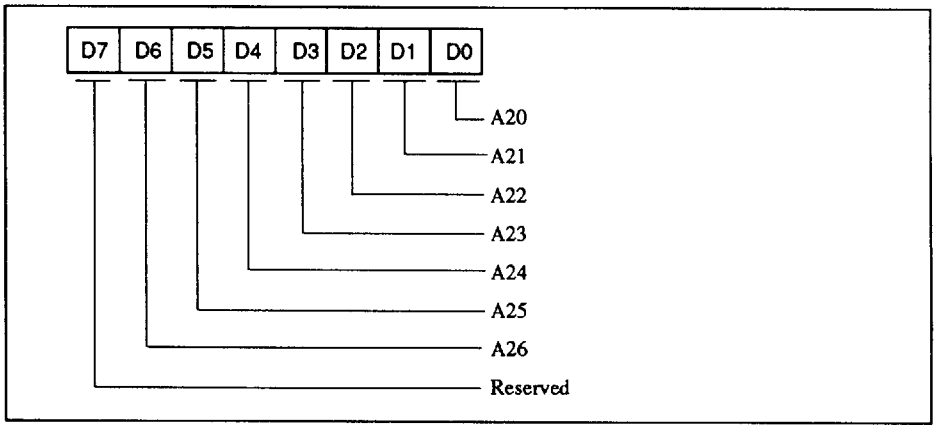
- 00 128K deep (not supported)
- 01 256K deep
- 10 512K deep
- 11 Reserved

7:6 Reserved

## Index Register E4

This VRAM starting address register defaults to 00.

***Index E4*** *VRAM Starting Address Register*




*Bits:* 6:0 A26:20 of starting address.

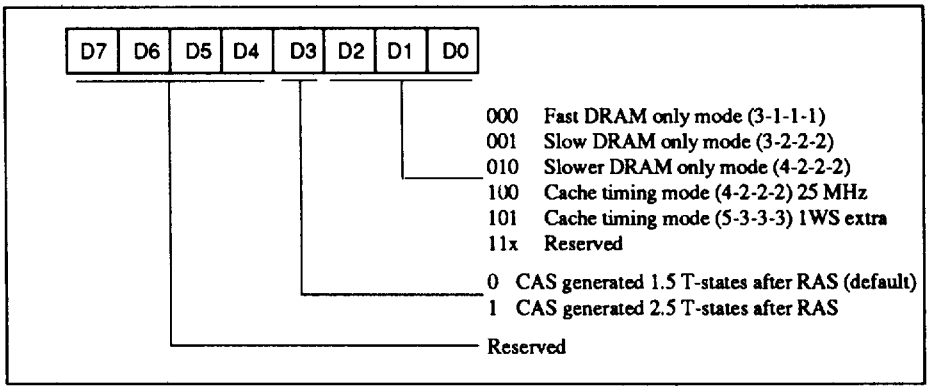
7 Reserved, always read as zero.

## Index Register E5

T-49-17-01

This Wingine VRAM timing mode register defaults to 00. This register is CHIPSet specific; and will not normally be needed by the Wingine driver.

***Index E5*** *Wingine VRAM Timing Mode Register*




*Bits:* 2:0 VRAM timing mode. Bit 2 must be the same as the normal DRAMs.
- 000 Fast DRAM only mode (3-1-1-1)
- 001 Slow DRAM only mode (3-2-2-2)
- 010 Slower DRAM only mode (4-2-2-2). CAS toggles on T-state boundaries.
- 100 Cache timing mode (4-2-2-2) 25 MHz
- 101 Cache timing mode (5-3-3-3) one extra wait state 33 and 40 MHz
- 11x Reserved

3 RAS to CAS timing in DRAM only mode.
- 0 CAS generated 1.5 T-states after RAS (default).
- 1 CAS generated 2.5 T-states after RAS (for slower RAMs)

7:4 Reserved

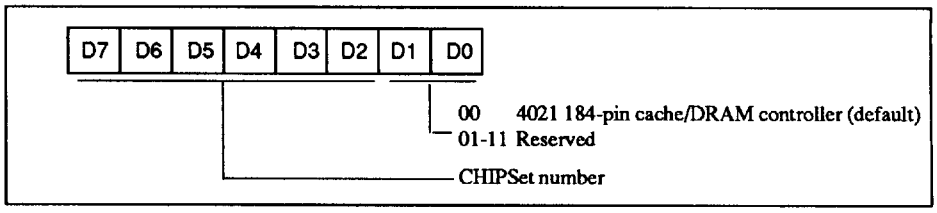
## Index Register F0



This CHIPSet identification register defaults to 04. This register contains a unique code for each CHIPSet. Bits 7:2 of this register are read only. Bits 1:0 are read and write.

***Index F0*** *CHIPSet Identification Register*

T-49-17-0




*Bits:* 1:0 Variations of the CHIPSet. These bits are read and write.

| | |
|---|---|
| 00 | 4021 184-pin cache/DRAM controller (default) |
| 01-11 | Reserved |

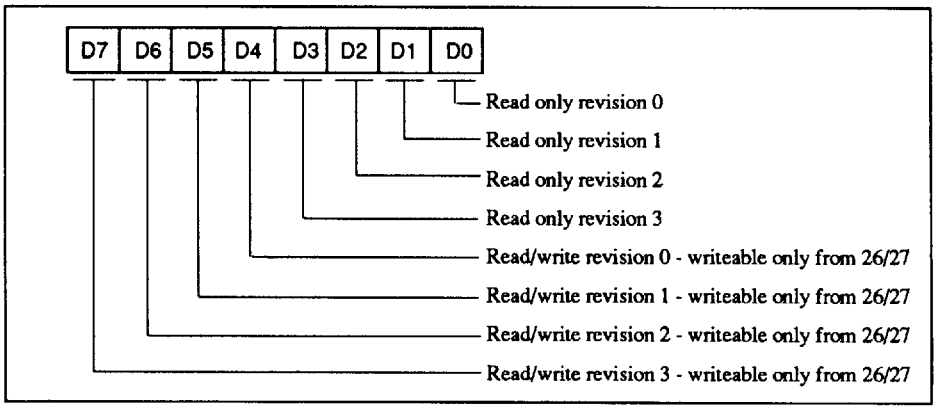
7:2 CHIPSet number

| | |
|---|---|
| 0000 | ISA486 |
| 01xx | ISA486sx |

## Index Register F1

This CHIPSet revision level register provides the revision level of all CHIPSets.

***Index F1*** *CHIPSet Revision Level Register*




*Bits:* 3:0 Read only revision level of the CHIPSet for all CHIPSets.

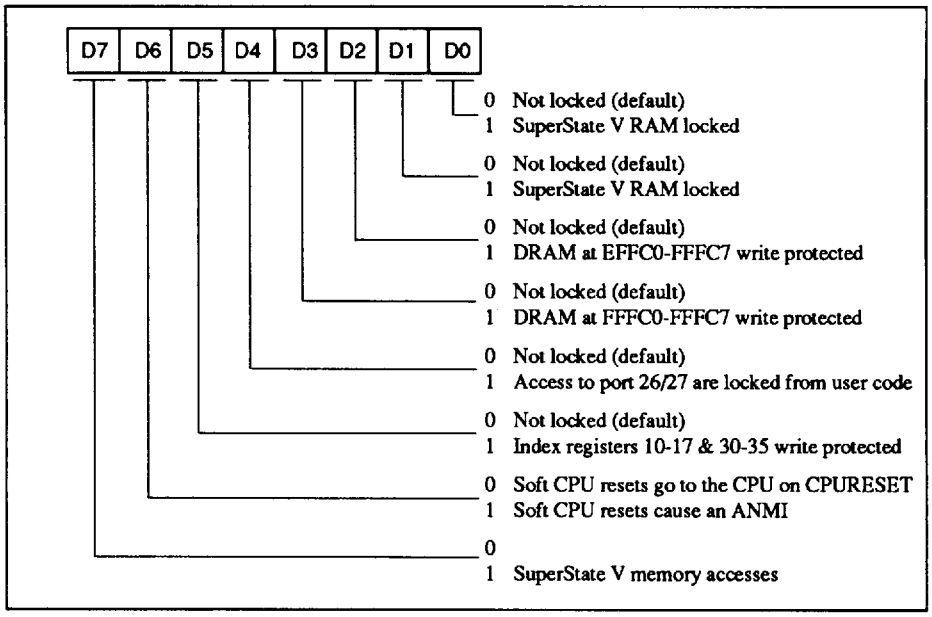
7:4 Read/write revision information. The BIOS will write additional revision information here.

## Index Register F2

T-49-17-01

This SuperState V locking register defaults to 00. This register is writeable through the Port 26/27 window only.

***Index F2*** *SuperState V Locking Register*




*Bits:* 0 SuperState V RAM lock for A0000-AFFFF.
- 0 Not locked (default)
- 1 SuperState V RAM is locked and cannot be accessed from ADS* cycles. This bit also sets the area to non-cacheable.

1 SuperState V RAM lock for B0000-BFFFF.
- 0 Not locked (default)
- 1 SuperState V RAM is locked and cannot be accessed from ADS* cycles. This bit also sets the are to non-cacheable.

2 SuperState V lower entry point write protect.
- 0 Not locked (default)
- 1 DRAM at EFFC0-EFFC7 is write protected. Since the write protection decode is critical for cache writes, this bit will write protect E0000-EFFFF.

T-49-17-01

*Bits:* 3 SuperState V upper entry point write protect.
- 0 Not locked (default)
- 1 DRAM at FFFC0-FFFC7 is write protected. Since the write protection decode is critical for cache writes, this bit will write protect F0000-FFFFF.

4 Port 26/27 lock.



- 0 Not locked (default)
- 1 Accesses to port 26/27 are locked from user code. they are accessible between the time an AADS* occurs and an ADS* code fetch occurs.

5 Register lock.
- 0 Not locked (default)
- 1 Index registers 10-17 (Dram configuration) and 30-35 (memory decodes) are write protected from port 22/23 accesses. They are still accessible from port 26/27 accesses.

6 Soft CPU reset redirection.
- 0 Soft CPU resets go to the CPU on CPURESET.
- 1 Soft CPU resets cause an ANMI.

7 Mask operation.
- 0
- 1 For SuperState V memory accesses (AADS* and ADS* active).

## Index Register F3

This register is reserved for SuperState V secondary lock configuration. It is not implemented on the 4021.

## Index Register F4

This register is writeable through the Port 26/27 window only.

This SuperState V timer register defaults to 00 (not counting). Writing a value to this register initiates a timer that counts down at about 1KHz (32.768KHz divided by 32) and generates an ANMI. The value of the timer must be rewritten each time since it is not saved in hardware. Reading this register will give the current count. Writing this register while it is counting will restart the count at the value written. Writing a zero will stop the count without generating an ANMI.

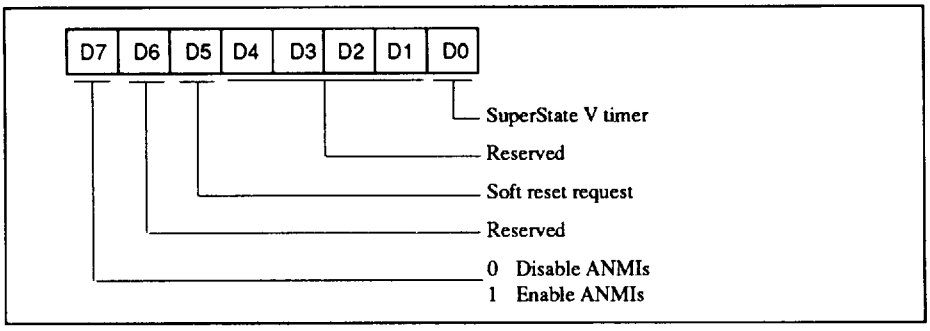
## Index Registers F5 through F7

These registers are reserved for more timers or other SuperState V functions.

## Index Register F8

This zero ANMI register defaults to 00. It identifies the source of any pending ANMI from within the chip, and allows it to be cleared. Each bit (6:0) represents an ANMI source. When the register is read, any bit which is one indicates that source is causing an ANMI. Writing a one to a bit clears that ANMI source. Bits written to zero have no affect on the ANMI source. Bit 7 is the ANMI mask which masks the final ANMI output, after the sources are ORed.

***Index F8*** *Zero ANMI Register*




*Bits:* 0 SuperState V timer.

4:1 Reserved

5 Soft reset request (redirected CPURESET).

6 Reserved

7 ANMI global mask. Can disable all ANMI sources from this chip.
- 0 Disable ANMIs
- 1 Enable ANMIs

## Index Register F9 through FB

These ANMI registers (1-3) are reserved, but can be implemented on other chips.

CHAPTER 14

T-49-17-01

# AC/DC Characteristics

The tables and figures in this section describe the operating environment and signal timings required by the 4021 and 4025.

## General Characteristics

For guaranteed operation, we recommend using the ISA/486 within the ranges specified in the following table.

**Table 14-1.** *Recommended Operating Conditions*

| Symbol | Parameter | Min. | Max. | Unit | Notes |
|---|---|---|---|---|---|
| $V_{CC}$ | Supply Voltage | 4.50 | 5.25 | V | |
| $V_I$ | Input Voltage | -0.50 | 5.50 | V | |
| $V_O$ | Output Voltage | -0.50 | 5.50 | V | |
| $T_{OP}$ | Operating Temperature | 0.00 | 70.00 | C | |
| $T_{STG}$ | Storage Temperature | -40.00 | 125.00 | C | |

The following table identifies the most extreme conditions tolerated by the chip without permanent device damage. Note that these are not continuous operating conditions; the chip may be damaged if operated near these limits for a prolonged period of time. Functional operation should be restricted to the conditions described under Recommended Operating Conditions.



***Table 14-2.*** *Maximum Tolerated Conditions*



| Symbol | Parameter | Min. | Max. | Unit |
|---|---|---|---|---|
| $V_{CC}$ | Supply Voltage | -0.50 | 5.50 | V |
| $T_A$ | Ambient Temperature | 0.00 | 70.00 | C |

The following table shows the I/O capacitance values for 5V operation.

***Table 14-3.*** *Capacitance*

| Symbol | Parameter | Min. | Max. | Unit | Test Condition |
|---|---|---|---|---|---|
| $C_{IN}$ | Input Capacitance | | 10.00 | pF | $f_C$ = 1 MHz and 70 pf load |
| $C_{I/O}$ | I/O Capacitance | | 10.00 | pF | Unmeasured pins |

# DC Characteristics

The following table provides the DC operating characteristics of the ISA486 TTL signals.

T-49-17-01

***Table 14-4.*** *DC Characteristics for TTL Signals*

| Symbol | Parameter | Min. | Max. | Unit | Test Condition |
|---|---|---|---|---|---|
| $V_{IL}$ | Input Low Voltage | -0.50 | 0.80 | V | |
| $V_{IH}$ | Input High Voltage | 2.00 | Vcc + 0.5 | V | |
| $V_{OL}$ | Output Low Voltage | | 0.40 | V | $I_{OL}$ = 4.0 mA |
| $V_{OH}$ | Output High Voltage | | 2.40 | V | $I_{OH}$ = -4.0 mA |
| $V_{CL}$ | Clock Output Low | | 0.40 | V | $I_{OL}$ = 4.0 mA |
| $V_{CH}$ | CLock Output High | 2.40 | | V | $I_{OH}$ = -4.0 mA |
| $I_{IL}$ | Input Leakage Current | -10.00 | 20.00 | uA | $V_I$ = Vcc to 0 V |
| $I_{OL}$ | Output Leakage Current | -10.00 | 20.00 | uA | $V_O$ |
| Icc | Vcc Supply Current | | 100.00 | mA | 1xCLK = 33 MHz |

The following table provides the DC operating characteristics fo the ISA486 CMOS signals.

***Table 14-5.*** *DC Characteristics for CMOS Signals*

| Symbol | Parameter | Min. | Max. | Unit | Test Condition |
|---|---|---|---|---|---|
| $V_{IL}$ | Input Low Voltage | -0.50 | 0.60 | V | |
| $V_{IH}$ | Input High Voltage | 3.50 | | V | |
| $V_{OL}$ | Output Low Voltage | | 0.40 | V | $I_{OL}$ = 4.0 mA |
| $V_{OH}$ | Output High Voltage | Vcc 0.5 | 2.40 | V | $I_{OH}$ = -4.0 mA |
| $V_{CL}$ | Clock Output Low | | 0.40 | V | $I_{OL}$ = 4.0 mA |
| $V_{CH}$ | Clock Output High | Vcc -0.5 | | V | $I_{OH}$ = -4.0 mA |
| $I_{IL}$ | Input Leakage Current | -10.00 | 10.00 | uA | $V_I$ = Vcc to 0 V |
| $I_{OL}$ | Output Leakage Current | -10.00 | 20.00 | uA | $V_O$ = Vcc to 0.45 V |
| Icc | Vcc Supply Current | | 100.00 | mA | 1xCLK = 33 MHz |

# 4021 AC Characteristics



The following tables show the AC operating characteristics for 16 MHz, 20 MHz, 25 MHz, 33 MHz, 40 MHz, and 50 MHz.

T-49-17-01

***Table 14-6.*** *CPU Interface Timing*

| Symbol | Parameter | 16 MHz | | 20 MHz | | 25 MHz | | 33 MHz | | 40 MHz | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Min. | Max. | Min. | Max. | Min. | Max. | Min. | Max. | Min. | Max. |
| t99 | CLK2 to SCLK delay | | 2 | | 2 | | 2 | | 2 | | 2 |
| t100 | A, BE0-3, M/IO, D/C, WR, ADS setup | 5 | | 5 | | 5 | | 5 | | 5 | |
| t101a | HLDA setup | 5 | | 5 | | 5 | | 5 | | 5 | |
| t101b | HLDA hold | 3 | | 3 | | 3 | | 3 | | 3 | |
| t102 | BLAST setup to CLK | 20 | | 18 | | 13 | | 10 | | 8 | |
| t103 | EADS delay from CLK | 3 | 40 | 3 | 40 | 3 | 30 | 3 | 24 | 3 | |
| t103 | A, ADS, D/C, M/IO, W/R delay from CLK for 38605 only | 3 | 20 | 3 | 20 | 3 | 20 | 3 | 20 | 3 | 10 |
| t104 | KEN delay from CLK | 3 | 20 | 3 | 20 | 3 | 20 | 3 | 20 | 3 | 10 |
| t104a | KEN delay from Address | 3 | 12 | 3 | 12 | 3 | 10 | 3 | 9 | 3 | 1 |
| t105 | RDY delay from CLK | 3 | 40 | 3 | 40 | 3 | 30 | 3 | 23 | 3 | 18 |
| t105a | BRDY delay from CLK | 3 | 40 | 3 | 40 | 3 | 30 | 3 | 25 | 3 | |
| t106 | HOLD delay from CLK | 3 | 20 | 3 | 20 | 3 | 20 | 3 | 19 | 3 | 10 |
| t106a | HLDA2 delay | | 20 | | 20 | | 20 | | 20 | | 10 |
| t106b | AADS setup | 5 | | 5 | | 5 | | 5 | | 5 | |
| t196c | AADS hold | 5 | | 5 | | 5 | | 5 | | 5 | |
| t107 | FLUSH, A20M, NMI, INTR delay from CLK | 3 | 20 | 3 | 20 | 3 | 20 | 3 | 19 | 3 | 10 |
| t108 | PWRGOOD, KBRESET setup to CLK | 5 | | 5 | | 5 | | 5 | | 5 | |
| t108a | PWRGOOD, KBRESET hold to CLK | 3 | | 3 | | 3 | | 3 | | 3 | |
| t109 | CPURESET setup to CLK | 10 | | 10 | | 5 | | 5 | | 5 | |
| t109a | CPURESET hold to CLK | 3 | | 3 | | 3 | | 3 | | 3 | |

T-49-17-01

*Figure 14-1. CLK2 to SCLK 4021 Timing*



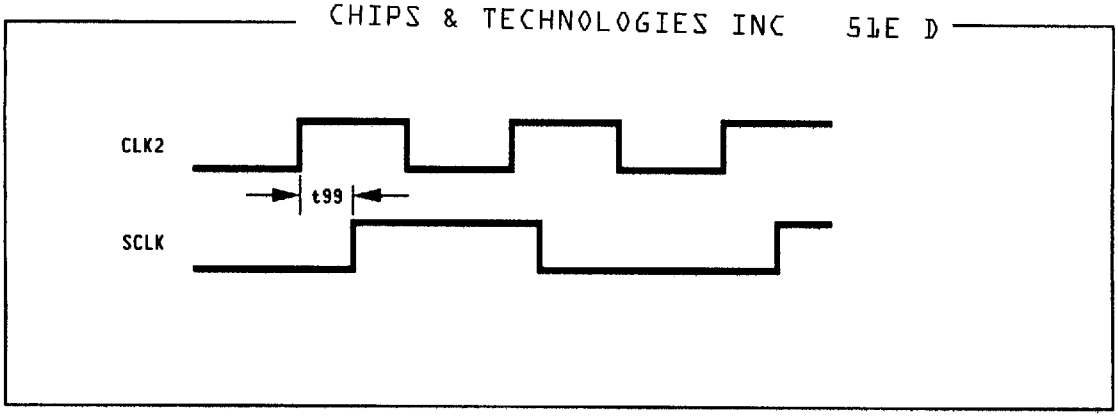



*Figure 14-2. 4021 Input Timing*

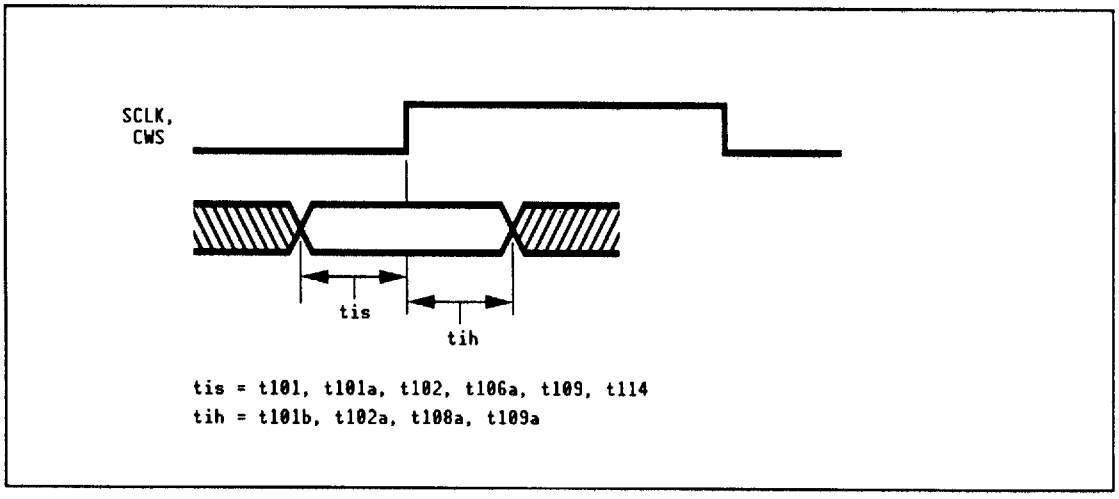


**Table 14-7.** *DRAM Timing*

| Symbol | Parameter | 16 MHz | | 20 MHz | | 25 MHz | | 33 MHz | | 40 MHz | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Min. | Max. | Min. | Max. | Min. | Max. | Min. | Max. | Min. | Max. |
| t112 | 4021 CLK to RAS delay | 3 | 20 | 3 | 20 | 3 | 20 | 3 | 20 | 3 | 14 |
| t114 | 4021 CLK to CAS delay | 5 | 15 | 5 | 15 | 5 | 15 | 5 | 15 | 5 | 15 |
| t116 | 4021 CLK to DWE delay | 3 | 20 | 3 | 20 | 3 | 20 | 3 | 20 | 3 | 15 |
| t120 | 4021 CLK to MA0-11 delay | | 20 | | 20 | | 20 | | 20 | | 18 |
| t121 | 4021 A to MA on page hit | | 27 | | 27 | | 27 | | 27 | | 20 |
| t130 | 4021 CLK to BWEN delay | 4 | 16 | 4 | 16 | 4 | 16 | 4 | 16 | 4 | 14 |
| t131 | 4021 CLK to AC1 delay | | 18 | | 18 | | 18 | | 18 | | 16 |
| t132 | 4021 CLK to AC0 delay | | 20 | | 20 | | 20 | | 20 | | 20 |
| t133 | 4021 CLK to AC2 delay | | 20 | | 20 | | 20 | | 20 | | 15 |
| t133 | 4021 CLK to CBSHIFT delay | | 20 | | 20 | | 20 | | 20 | | 15 |

**Figure 14-3.** *4021 Output Timing*

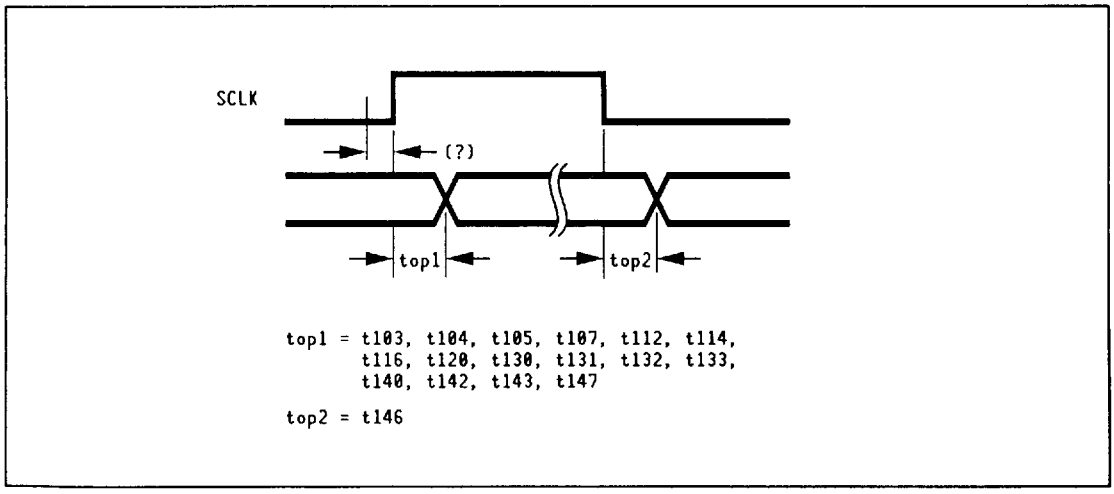


*Figure 14-4. 4021 Timing for Local Bus Address to MA Bus Delay, CA3x*



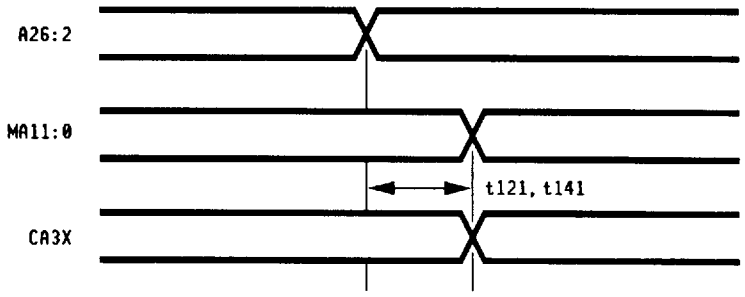



*Table 14-8. Cache Timing*

T-49-17-01

| Symbol | Parameter | 16 MHz | | 20 MHz | | 25 MHz | | 33 MHz | | 40 MHz | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Min. | Max. | Min. | Max. | Min. | Max. | Min. | Max. | Min. | Max. |
| t140 | 4020 SCLK to CA3x | 2 | 30 | 2 | 30 | 2 | 30 | 2 | 30 | 2 | 20 |
| t141 | 4020 CA3x prop delay | | 15 | | 15 | | 15 | | 15 | | 15 |
| t142 | CRDx Cache OE delay from CLK high | 2 | 20 | 2 | 20 | 2 | 20 | 2 | 12 | 2 | 10 |
| t143 | CWE* delay from CWS CLK | 5 | 10 | 5 | 10 | 5 | 10 | 5 | 8 | 5 | 8 |

*Table 14-9. Tag Timing*

| Symbol | Parameter | 16 MHz | | 20 MHz | | 25 MHz | | 33 MHz | | 40 MHz | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Min. | Max. | Min. | Max. | Min. | Max. | Min. | Max. | Min. | Max. |
| t145 | TAG data read setup to BRDY/RDY low | 30 | | 30 | | 26 | | 20 | | 18 | |
| t147 | TAG data output delay | | 23 | | 23 | | 23 | | 23 | | 23 |

*Figure 14-5. 4021 Timing for TAG Data to RDY/BRDY Setup*

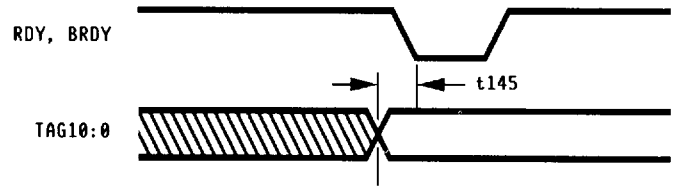


*Table 14-10. AT Bus Timing*

T-49-17-01

CHIPS & TECHNOLOGIES INC 51E D

| Symbol | Parameter | 16 MHz | | 20 MHz | | 25 MHz | | 33 MHz | | 40 MHz | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Min. | Max. | Min. | Max. | Min. | Max. | Min. | Max. | Min. | Max. |
| t161 | REF setup to BCLK high | 10 | | 10 | | 10 | | 10 | | 10 | |
| t163 | AEN delay from addrs | | 30 | | 30 | | 30 | | 30 | | 30 |
| t164 | BCLK to BALE active/inactive | 3 | 15 | 3 | 15 | 3 | 15 | 3 | 15 | 3 | 15 |
| t165 | BCLK to MEMx delay | 3 | 15 | 3 | 15 | 3 | 15 | 3 | 15 | 3 | 15 |
| t166 | BCK to IOx delay | 3 | 15 | 3 | 15 | 3 | 15 | 3 | 15 | 3 | 15 |
| t167 | IOCHRDY setup to BCLK | 10 | | 10 | | 10 | | 10 | | 10 | |
| t168 | BE0-3 to XA0-1 delay from SCLK | | 25 | | 25 | | 25 | | 25 | | 25 |
| t169 | BE0-3 to SBHE delay from SCLK | | 25 | | 25 | | 25 | | 25 | | 25 |
| t170 | MC16, IOCS16 setup to BCLK | 25 | | 25 | | 25 | | 25 | | 25 | |
| t171 | MCS16, IOCS16 hold to BCLK | 5 | | 5 | | 5 | | 5 | | 5 | |
| t172 | 0WS setup to BCLK low | 20 | | 20 | | 20 | | 20 | | 20 | |
| t173 | 0WS hold from BCLK high | 5 | | 5 | | 5 | | 5 | | 5 | |
| t175 | IRQ pulse width | 100 | | 100 | | 100 | | 100 | | 100 | |
| t176 | DREQn pulse width (OSC CLKs) | 8 | | 8 | | 8 | | 8 | | 8 | |
| t177 | DACKn valid from BCLK low | | 100 | | 100 | | 100 | | 100 | | 100 |
| t178 | MUX0-1 setup to OSC | 10 | | 10 | | 10 | | 10 | | 10 | |
| t179 | SEL 0-3 delay | 3 | | 10 | 3 | 10 | 3 | 10 | 3 | 10 | 3 |
| t180 | SEL 0-2 setup from BCLK (DREQ) | 10 | | 10 | | 10 | | 10 | | 10 | |
| t181 | TC delay from BCLK low | | 50 | | 50 | | 50 | | 50 | | 50 |
| t185 | IOCHCK setup | 5 | | 5 | | 5 | | 5 | | 5 | |

*Table 14-11. Miscellaneous Timing*

| Symbol | Parameter | 16 MHz | | 20 MHz | | 25 MHz | | 33 MHz | | 40 MHz | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Min. | Max. | Min. | Max. | Min. | Max. | Min. | Max. | Min. | Max. |
| t186 | ROMCS delay | | 20 | | 20 | | 20 | | 20 | | 20 |
| t187 | Chip select 0-1 delay | | 15 | | 15 | | 15 | | 15 | | 15 |
| t187 | AC2:1 valid to AT bus control valid | 3 | 20 | 3 | 20 | 3 | 20 | 3 | 20 | 3 | 20 |
| t188 | ACEN valid from cmd valid | 2 | 20 | 2 | 20 | 3 | 20 | 3 | 20 | 3 | 20 |

# 4025 AC Characteristics



The following tables describe the AC characteristics for the 4025 CHIPSet.

**Table 14-12.** *Data Buffer Timing*

T-49-17-01

| Symbol | Parameter | 16 MHz | | 20 MHz | | 25 MHz | | 33 MHz | | 40 MHz | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Min. | Max. | Min. | Max. | Min. | Max. | Min. | Max. | Min. | Max. |
| t200 | 4025 MD to D delay | 3 | 18 | 3 | 18 | 3 | 13 | 3 | 18 | 3 | 15 |
| t201 | 4025 MD to D delay (read latch giving trans.) | 3 | 30 | 3 | 30 | 3 | 30 | 3 | 30 | 3 | 30 |
| t202 | 4025 MD bus data setup to Read Latch closing | 1 | | 1 | | 1 | | 1 | | 1 | |
| t203 | 4025 AC1 low to D bus valid | | 20 | | 20 | | 20 | | 20 | | 20 |
| t204 | 4025 CAS high to MD/MP bus setup (DRAM read) | 5 | | 5 | | 5 | | 5 | | 5 | |
| t210 | 4025 AC0 low to M bus valid | 3 | 20 | 3 | 20 | 3 | 20 | 3 | 20 | 3 | 20 |
| t211 | 4025 MD bus hold after AC0 high | 3 | | 3 | | 3 | | 3 | | 3 | |
| t212 | 4025 AC0 low to MP valid | 3 | 20 | 3 | 20 | 3 | 20 | 3 | 20 | 3 | 20 |
| t220 | 4025 AC2 low to MD bus valid | 5 | 25 | 5 | 25 | 5 | 25 | 5 | 25 | 5 | 25 5 |
| t221 | 4025 CAS low hold time from CLK for DRAM read | 7 | | 7 | | 7 | | 7 | | 7 | |
| t222 | 4025 MP/MD hold from CAS edge | 5 | | 5 | | 5 | | 5 | | 5 | |
| t230 | 4025 D bus data setup to BWEN low and clock high | 5 | | 5 | | 5 | | 5 | | 5 | |
| t231 | 4025 D bus data hold after BWEN low and clock high | 3 | | 3 | | 3 | | 3 | | 3 | |
| t232 | 4025 BWEN and clock high to MD bus new data valid | | 25 | | 25 | | 25 | | 25 | | 25 |
| t233 | 4025 MD bus data hold after BWEN low and clock high | 6 | | 6 | | 6 | | 6 | | 6 | |
| t240 | AC1 high to D high Z | | 15 | | 15 | | 15 | | 15 | | 15 |
| t241 | AC0 high to MD high Z | | 25 | | 25 | | 25 | | 25 | | 25 |
| t250 | MD setup to ACEN low to high | 10 | | 10 | | 10 | | 10 | | 10 | |
| t251 | MD hold to ACEN low to high | 5 | | 5 | | 5 | | 5 | | 5 | |
| t252 | AC2:1 valid to data valid for AT cycle | 3 | 35 | 3 | 35 | 3 | 35 | 3 | 35 | 3 | 35 |
| t253 | MD low to high, high to low byte stearing (AT cycle) | | 30 | | 30 | | 30 | | 30 | | 30 |

T-49-17-01

*Figure 14-6. 4025 Output Timing*



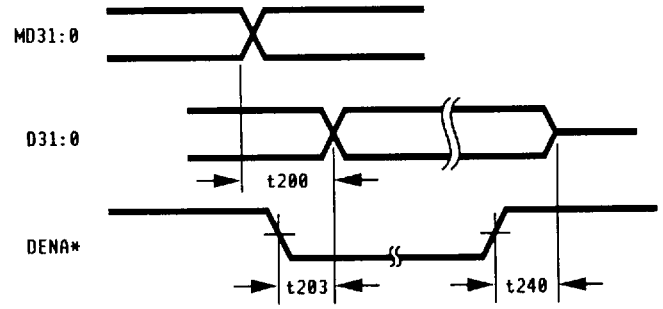



*Figure 14-7. 4025 Input Timing*

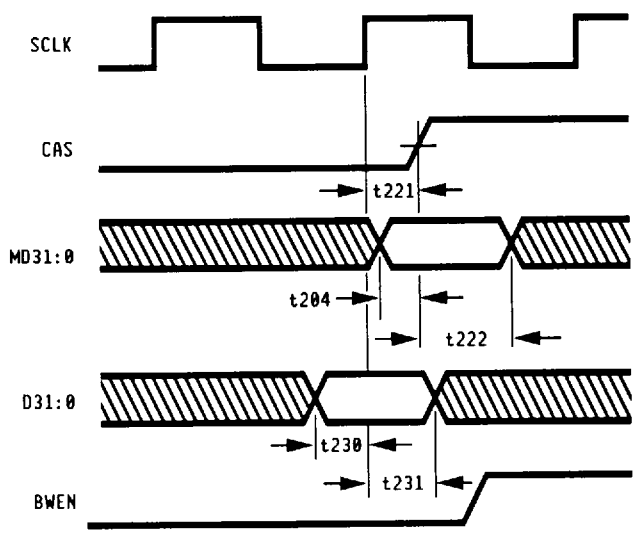


**Figure 14-8.** *AT Bus Control Signal Setup and Hold*

T-49-17-01



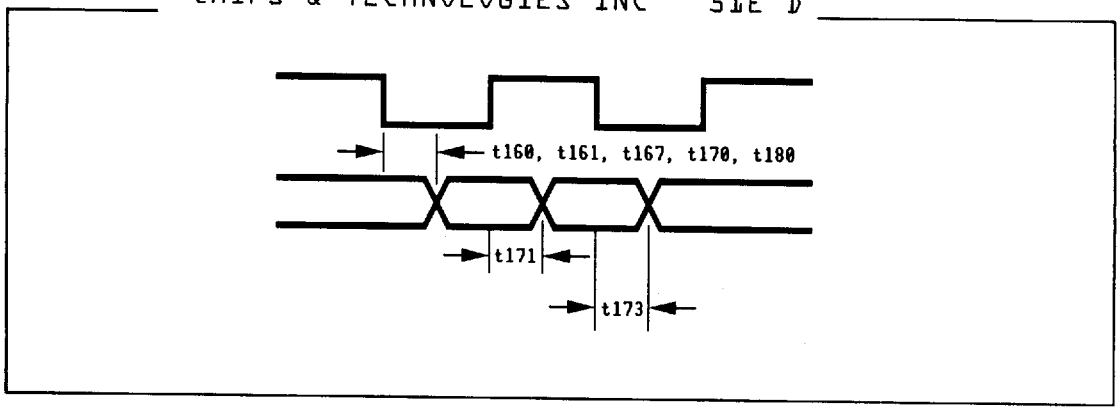



**Figure 14-9.** *4025 Output Timing*

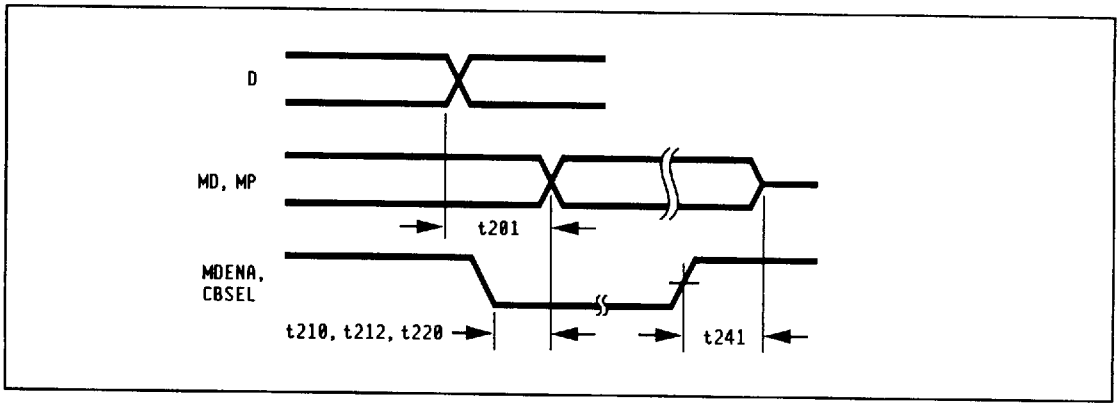



**Figure 14-10.** *4025 Output Timing*

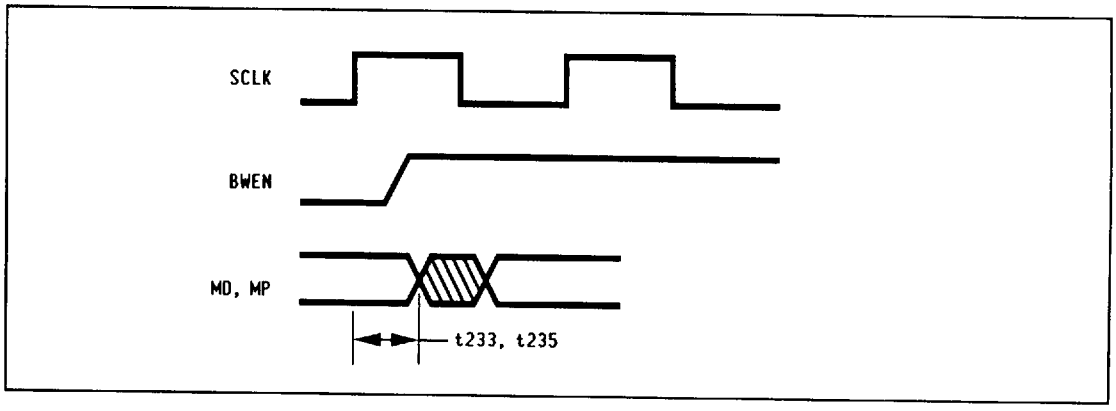
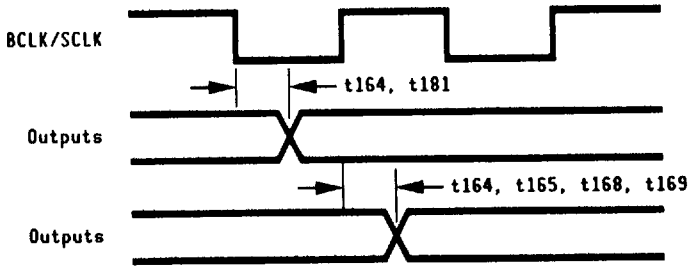


*Figure 14-11. AT Bus Control Signal Delays*

T-49-17-01





# Mechanical Specifications

T-49-17-01

The ISA/486 is a two-chip CHIPSet packaged with the 4021 184-pin plastic flat pack and the 4025 120-pin plastic flat pack. The dimensions for each package are shown in Figure 15-1 and 15-2.

**Figure 15-1.** *4021 184-Pin Plastic Flat Pack*

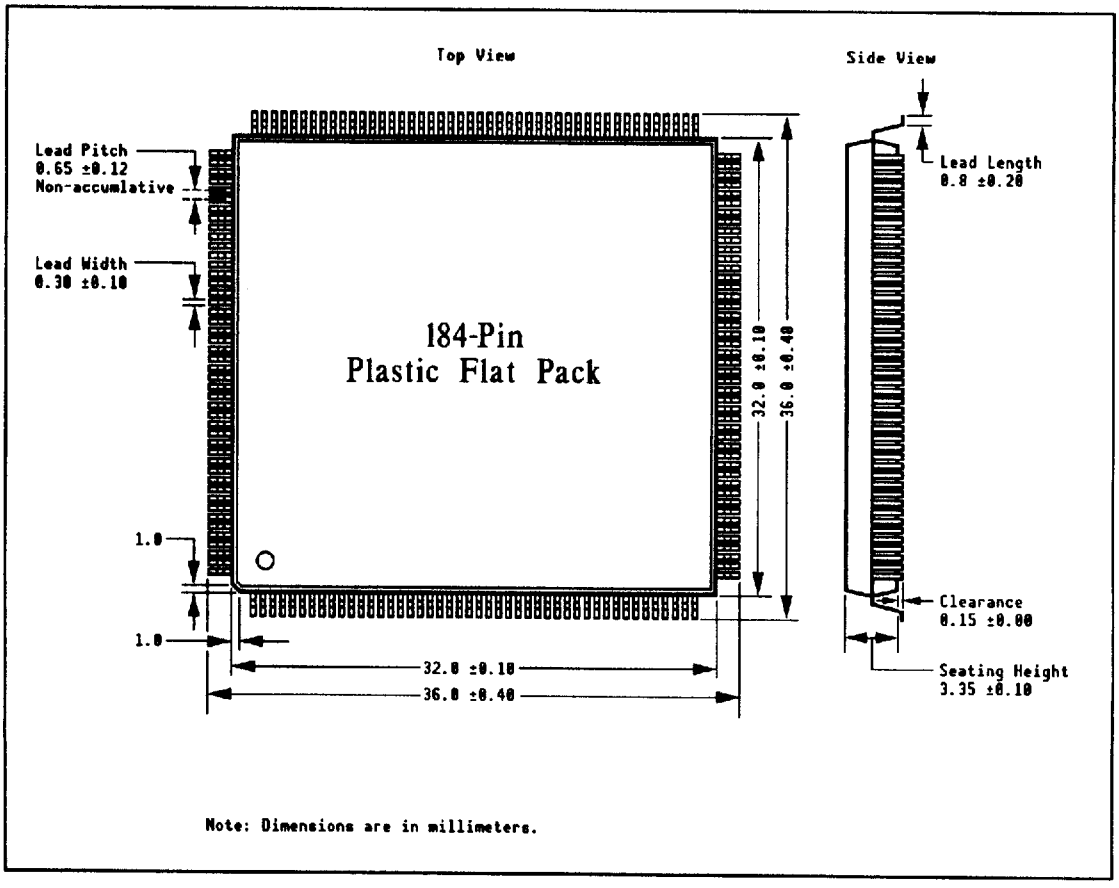
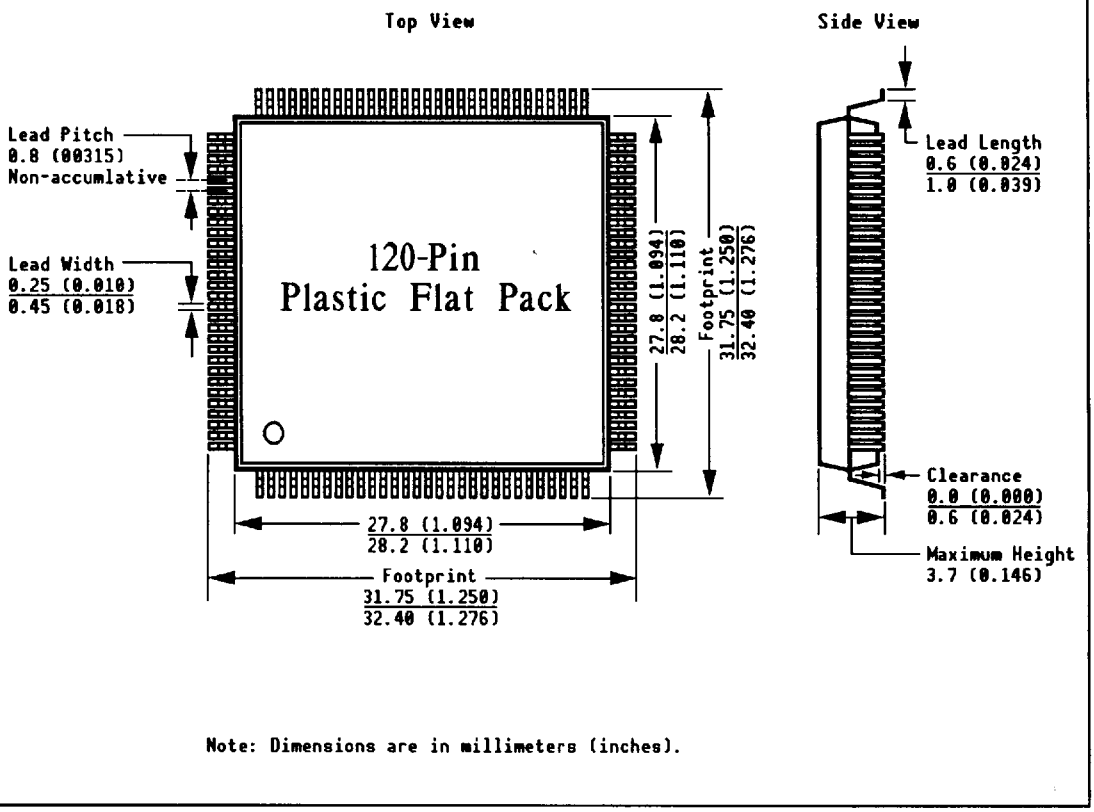


*Figure 15-2. 4025 120-Pin Flat Pack*

T-49-17-01

CHIPS & TECHNOLOGIES INC 51E D




Note: Dimensions are in millimeters (inches).