M68000
MICRO MINUTES

MM- 422-03

THE MC68661 SURPASSES THE 2651 IN FLEXIBILITY AND FUNCTIONALITY

A Solution for Intelligent Terminal Serial Communications

The MC68661 Enhanced Programmable Communications Interface (EPCI) is a pin-for-pin compatible, upgraded version of the Signetics 2651 Programmable Communications Interface (PCI). The MC68661 enhancements include:

- Three baud rate sets (A, B, C) compared to set C only for the 2651 (Table 1).
- Improved support for asynchronous and synchronous operations (Tables 2 and 3).
- Upgraded bus specifications (Table 4).

For new data communications system designs, the MC68661 offers increased flexibility and functionality over the 2651. By providing higher baud rates and greater accuracy, the MC68661 will satisfy a wider variety of system specifications. Also, the enhanced asynchronous and synchronous operation modes will simplify programming and the improved bus characteristics will allow quicker, less complex designs. Moreover, existing 2651 systems can be upgraded to the MC68661 with no hardware changes, and only minor software modifications to take advantage of the MC68661's new features. Since cost differences are minimal, the MC68661 is the clear choice for data communication system designers.

Table 1 - MC68661 Baud Rate Sets

MC68661		MC68661		MC68661 & 2651	
Set A		Set B		Set C	
Baud Rate	%Error	Baud Rate	%Error	Baud Rate	%Error
50	none	45.5	0.005	50	none
75	none	50	none	75	none
110	-0.01	75	none	110	none
134.5	none	110	-0.01	134.5	0.016
150	none	134.5	none	150	none
200	none	150	none	300	none
300	none	300	none	600	none
600	none	600	none	1200	none
1050	0.196	1200	none	1800	none
1200	none	1800	-0.19	2000	0.253
1800	-0.19	2000	-0.26	2400	none
2000	-0.26	2400	none	3600	none
2400	none	4800	none	4800	none
4800	none	9600	none	7200	none
9600	none	19200	none	9600	none
19200	none	38400	none	19200	none

MOTOROLA
Austin, Texas

Data Comm
MC68661

Table 2 - Enhanced Asynchronous Operations

Function	2651	MC68661	MC68661 Advantages
Terminate Async Transmission	Disable Tx (CR0=0) when TxEmt=0 Then drop RTS (CR5 =0) when TxEmt transitions from 0 to 1	Drop RTS (CR5=0) when TxRdy=0	Simplifies programming
Break Detect	Set FE (SR5=1) and transfer null char. to Rx holding reg.	Set FE (SR5=1) and transfer null char. to Rx holding reg. Also pin 25 can be BR output by setting MR27-24	Simplifies H/W break detect
External Jam Sync	none	Pin 9 can be ext. sync by setting MR27-24	Allows External syncronization
No. of stop bits	One or two as selected by MR17	One only	Increased clock skew tolerance

Table 3 - Enhanced Bisync Support

Function	2651	MC68661	MC68661 Advantages
DLE stuffing in Transparent mode	None- must be done in software	Auto DLE stuffing when a DLE is loaded, except when send DLE command is active (CR3=1)	Simplifies programming
Termination of send DLE command	Must clear command (CR3=0) on next TxRdy=0	One time command - automatic CR3 reset	Simplifies programming
Syncl stripping in double sync non-transparent mode	Only 1st Syncl of Syncl-Syncl pair is stripped.	Both Syncl's of pair are stripped	Simplifies programming
Clear DLE detect	Disable receiver (CR2=0) or reset error flags (CR4=1)	Disable receiver (CR2=0), reset error flags (CR4=1), or automatically cleared on 2nd char after DLE	Simplifies programming
DLE Detect operation	DLE detect is indicated for DLE-DLE or DLE-Syncl (SR3=1)	DLE detect is not indicated for DLE-DLE or DLE-Syncl (SR3=0)	Preferred operation

Table 3 - AC and DC Electrical Characteristics

Specification	2651	MC68661	MC68661 Advantage
Data Bus Drivers (I_{ol}, I_{oh})	Sink 1.6mA Source 100uA	Sink 2.2 mA Source 400uA	Increased Drive capability and/or relaxed timings
Chip Enable (T_{ce})	300 ns min	250 ns min	|
Address Setup (T_{as})	20 ns min	10 ns min	|
Address Hold (T_{ah})	20 ns min	10 ns min	|
R/W Control Setup (T_{cs})	20 ns min	10 ns min	|
R/W Control Hold (T_{ch})	20 ns min	10 ns min	|
Data Setup Write (T_{ds})	225 ns min	150 ns min	|
Data delay for Read (T_{dd})	250 ns max with Cl=100 pf	200 ns max with Cl=150 pf	| |
Data bus Tri-state after Read (T_{df})	150 ns max with Cl=100 pf	100 ns max with Cl=150 pf	|
Chip Enable to Chip Enable Delay (T_{ced})	700 ns min with Cl=100 pf	600 ns min with Cl=150 pf	| |
TxC or RxC High	500 ns min	480 ns min	|
Tx delay from Falling Edge of Txc (T_{xd})	650 ns max Cl=100 pf	650 ns max Cl=150 pf	|

MC68000 PACKAGING

Motorola has offered the MC68000 and MC68010 for some time now in 64-Pin DIP's, 68-Pin type B and C LCC's (Leadless Chip Carriers), and 68-Pin PGA's (Pin Grid Array),and will soon be offered in Plastic "Quad Paks" (plastic LCC).

The Pin Grid Array is being offered as an option to the larger dual-in-line package. The PGA for the '000 and '010 is approximately 1 square inch versus three square inches for the DIP. Also, the PGA offers an alternative to the LCC , and is competitive with the LCC socket system. Several advantages of the PGA over the LCC are:

1. Printed circuit boards may be constructed with conventional materials.
2. Ease of direct PCB solder mounting.
3. Socketing mechanism utilizes proven technology and offers higher reliability.
4. Will provide an upward migration path to the MC68020, which will be offered in PGA.

FOR NEW DESIGNS, THE PGA SHOULD BE EMPHASIZED OVER THE LCC.

64, 68, and 114 Terminal PGA Socket Vendors

The following vendors have been identified as suppliers of PGA sockets, or will support manufacture of these sockets at a customer's request. Motorola can NOT recommend a specific vendor's socket. This list may not be complete.

Advance Interconnections, Warwick, RI (401)885-0485
Augat, Inc., Attleboro, MA (617)222-2202
Robertson Nugent, Inc., Albany, IN (812)945-0211
Textool Products, Irving, TX (214)259-2676 (100 position PGA ZIP -5230 series)
Yamaichi (Nepenthe Distribution Inc.), (415)856-9332 (NP35 series ZIF)
Datak Corp., Sparks, NV (702)359-7660 (PCB design aids only)
AMP, Inc., Harrisburg, PA. (717)657-4110

68 Terminal Type B, LCC Socket Vendors

For those customers who wish to use the LCC package, the following is a non-extensive list of vendors of the LCC Type B sockets. Once again, Motorola does NOT recommend a particular vendor's socket. The list is for information purposes only.

AMP, Inc., Harrisburg, PA (717)657-4110
Methode Electronics Inc., Chicago, IL (312)867-9600
Textool Products, Irving, TX (214)259-2676

(continued)

Quad Pack (Plastic LCC) Socket Vendors

The following vendors supply sockets which will allow use of the type "FN" packages to appear in 1Q85.

Burndy Corp., Norwalk, CN. (203)852-6293
Plastronics Inc., Irving, TX. (214)721-1212

M68000
MICRO MINUTES

MM- 422-07

THE GREAT CONTEXT SWITCH DEBATE

Intel has again been doing what they are now famous for -- pushing sensationalism. Their program to promote the iAPX 286 is extremely aggressive, and seems to be an all-out effort with corporate blessing and money. It is a direct attack on the MC68000, and an attempt to slow the 68000's gains in the 16-bit marketplace. **There is however, as yet, no technological or business reason for Intel, National, or any other manufacturer to unseat the 68000 from its dominant position as the processor with the cleanest, most straighforward architecture, and the proven performance leader!**

One of the claims Intel is making is that the '286 makes context switches faster than the 68K. This is not the meat of the matter. Their logic flows something like this: The MC68000 has sixteen 32-bit registers (=512 bits). The '286 has eight "general" registers and four segment registers, all 16-bits (=192 bits). Intel's context switch comparisons make each machine flush the entire register set, rather than a more equitable equivalent register set (12 16-bit registers each). Actually, the MC68000 is substantially more efficient, performing a context switch at a rate of about 68 nanoseconds/bit (8 MHz processor) versus 118 nanoseconds/bit for an 8 MHz 286 (with a 16 MHz clock).

Doesn't it make sense that if you are to flush 256 bits it will take longer than to flush less than half of that number? If one were to follow this logic to its conclusion, the ideal 16-bit microprocessor, with the fastest context switch times, would have the programmer's model illustrated here:

What Intel is not pointing out, is that the '286 suffers from the same fate that the 8086 met -- too few, and too small registers. They don't point out the constant need for swapping register contents throughout a '286 program to compensate for the small, dedicated register set. Is it more important to move all of the registers at the beginning of a program a little more slowly (ala MC68000), or to be swapping those registers constantly throughout the program (ala '286)? In other words, would you rather change your 16 quarts of oil once a year, or change 4 quarts of oil each week?

Intel wants the world to believe that they have developed an advanced 16-bit microprocessor, yet it has the same restrictions (inherited from the 8-bit world) which have choked 8086 programmers for years. The MC68000 was designed by computer architects. Some other machines look like they were designed by computer archaeologists.

M68000 MICRO MINUTES

MM- 422-10 - A

MOTOROLA CORNERS THE MICROPROCESSOR DATA COMMUNICATIONS MARKET

Motorola has more data communications devices than any other semiconductor manufacturer. Motorola's total offerings of microprocessor peripherals that support data communications are listed in Table 1. Table 2 gives a comparison of functionality (# of registers) and features between the parts most suited for M68000 applications. For more information on data communications in general, and specific data communications devices, consult the Motorola Digital Data Communications Guide, BR244.

Table 1 - Motorola Data Communications Parts

MC6850	ACIA	- Asynchronous Communications Interface Adapter
MC6852	SSDA	- Synchronous Serial Data Adapter
MC6854	ADLC	- Advanced Data Link Controller
MC68120	IPC	- Intelligent Peripheral Controller
MC68562	DUSCC	- Dual Channel Universal Serial Communications Controller
MC68652	MPCC	- Multi-Protocol Communications Controller
MC68653	PGC	- Polynomial Generator Checker
MC68661	EPCI	- Enhanced Programmable Communications Interface
MC68681	DUART	- Dual Channel Universal Asynchronous Reciever/Transmitter
MC68901	MFP	- Multi-Function Peripheral

Table 2 - M68000 Data Communications Devices Comparison

	68562	68652	68661	68681	68901
	DUSCC	MPCC	EPCI	DUART	MFP
68000 Interface	YES	NO	NO	YES	YES
Asynchronous	YES	NO	YES	YES	YES
Stop Bits					
1, 1.5, 2	YES	---	YES	YES	YES
Increments	YES	---	NO	YES	YES
Parity	YES	---	YES	YES	YES
Synchronous	YES	YES	YES	NO	YES
BOP	YES	YES	NO	---	NO
BCP	YES	YES	YES	---	YES
Channels	2	1	1	2	1
Max Rate (Mbits/sec)	4	2	1	1	0.95
Buffering (Rcv/Xmit)	4/4	1/1	2/2	4/2	1/1
Data Bus Width	8	8/16	8	8	8
Prog. Baud Rates	YES	NO	YES	YES	NO
CRC	YES	YES	NO	NO	NO
Interrupts (V & AV)	V&A	A	A	V	V
Interrupt Registers	5	0	0	4	16
Registers	27	4	12	17	24
Programmable Timer	YES	NO	NO	YES	YES (4)
Character Length (bits)	5-8	1-8	5-8	5-8	5-8
Protocols (S= Supported on-chip, C= Capable of performing)					
BOP					
ADCCP	C	C	-	-	-
HDLC	C	C	-	-	-
SDLC	S	C	-	-	-
X.25	C	C	-	-	-
BCP					
BISYNC	S	C	C	-	C
DDCMP	C	C	C	-	C
# of Pins	40/48	40	28	40	48

M68000 MICRO MINUTES

MM- 422-13

MC68000 and MC68010 BLOCK MOVE CODE SEQUENCE

Many times we have been asked by customers why the M68000 family doesn't include a "Block Move" instruction, when so many of our competitors do. The M68000 instruction set provides primitives for general purpose use instead of a lot of limited-use dedicated instructions. The register set in the programming model is also general purpose. To clarify this point, the following is a 30 byte code sequence for performing block moves:

```
        LEA       SOURCE-48,A5
        LEA       DESTINATION+LENGTH,A6
        MOVE.W    #LENGTH,D0
   LOOP MOVEM.L   0(A5,D0),D1-D7/A0-A4
        MOVEM.L   D1-D7/A0-A4,-(A6)
        SUB.W     #48,D0
        BNE.S     LOOP
```

This code sequence uses all of the M68000 registers except A7, the stack pointer. A5 is set-up to point to the source string, while A6 points to the end of the destination string. D0 is used as both an offset for A5, and as a counter to monitor the block move. Each pass through the loop moves 48 bytes of data (twelve 32-bit long words).

Using this code, a 12.5 MHz MC68000 or MC68010 system with no wait states (i.e. 4 clock cycle reads and writes) can move data at a rate of 2.5+ Mbytes/second. By way of contrast, the 8089 from Intel, as a dedicated I/O processor, can move data at only 1.25 Mbytes/second.

The slightly less efficient, but more flexible 22 byte code stub presented next, takes advantage of the MC68010's "loop mode" to achieve a rate of 2.27 Mbytes/second on a 12.5 MHz processor:

```
        LEA       SOURCE,A0
        LEA       DESTINATION,A1
        MOVE.W    #COUNT-1,D0
   LOOP MOVE.L    (A0)+,(A1)+
        DBRA      D0,LOOP
```

Here, only 3 registers are utilized. The source and destination string addresses are set-up in registers A0 and A1 respectively, and the count (in long words), adjusted to serve as a loop counter is put in D0. The tight loop then simply moves source to destination until the count is exhausted. This loop will move a maximum of 256kbytes for each iteration. Note that once the loop is entered, the instruction stream fetches cease, and only data transactions appear in the external world. Remember, that this last code achieves this throughput only on the MC68010, a MC68000 achieves "only" 1.67Mbytes/second here.

WHAT DOES A 16-BIT MPU SYSTEM COST?
Is the MPU price tag alone the real picture?

There have been several price decreases announced by vendors of 16-bit processors (including Motorola), but making a decision based on price alone distorts the real issue, which is system cost.

Figure 1 shows a typical MPU system. The MPU generates address, data, and control signals required to interface memory and I/O. Both Intel (prior to the iAPX286), Zilog and National have designed microprocessors which fit in compact packages -- but, the customer must demultiplex the address and data before standard ROMs and RAMs can be interfaced to these processors. This demultiplexing circuitry is not needed on the M68000 family processors.

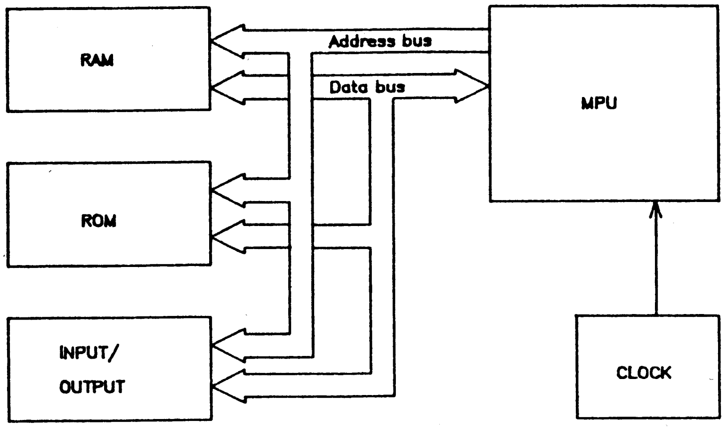


Figure 1. Typical Microprocessor System

Figure 2 shows the MC68000 and its clock generator. Seperate address, data and control buses are provided. This allows the designer to directly interface to memory or I/O.

Figure 3 shows the i8086 clock generator and required demultiplexors, Figure 4 the Z8000, Figure 5 the National 16032 and Figure 6 the iAPX286. Note how many packages are required on each to implement an MPU with seperate address, data, and control buses. The MC68000 obviously allows a cleaner, simpler design, with resultant lower costs and higher reliability.

Figure 2 – MC68000 and Clock

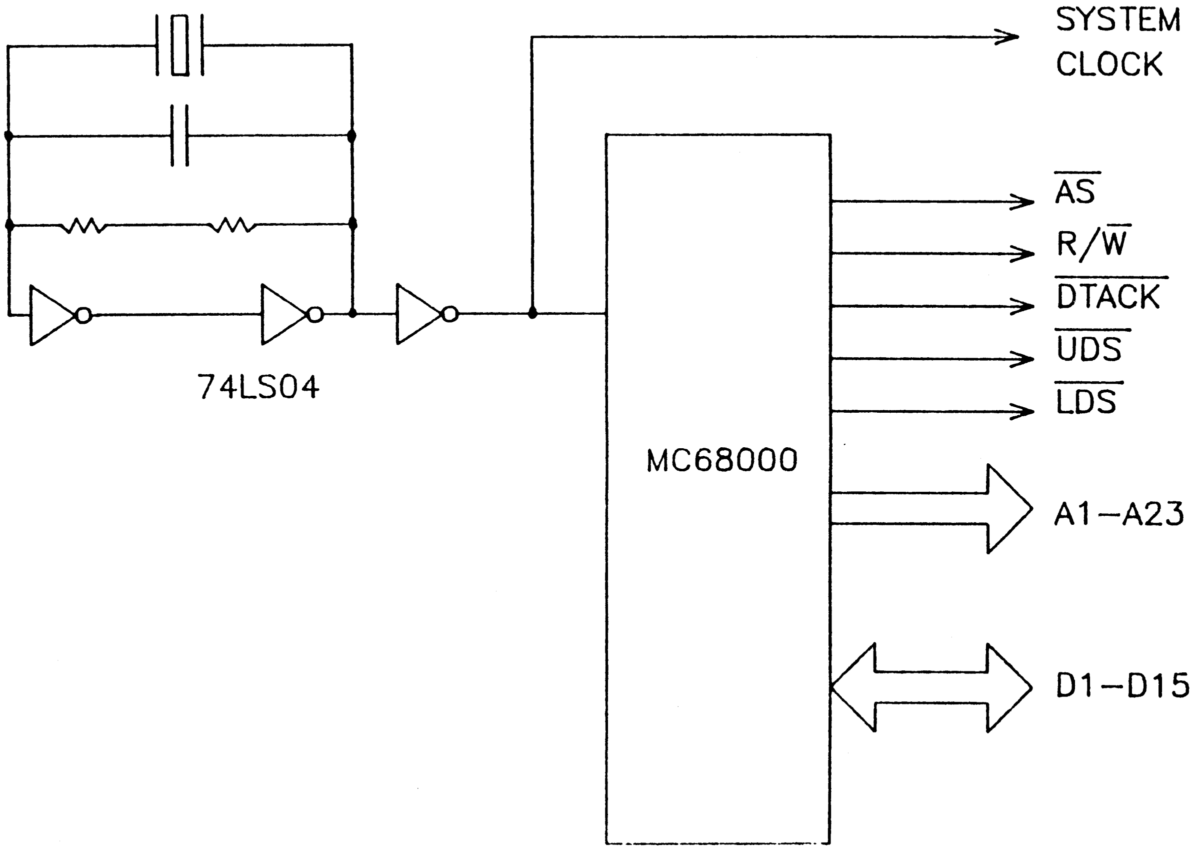


Fig. 3 - i8086, Clock & Demux

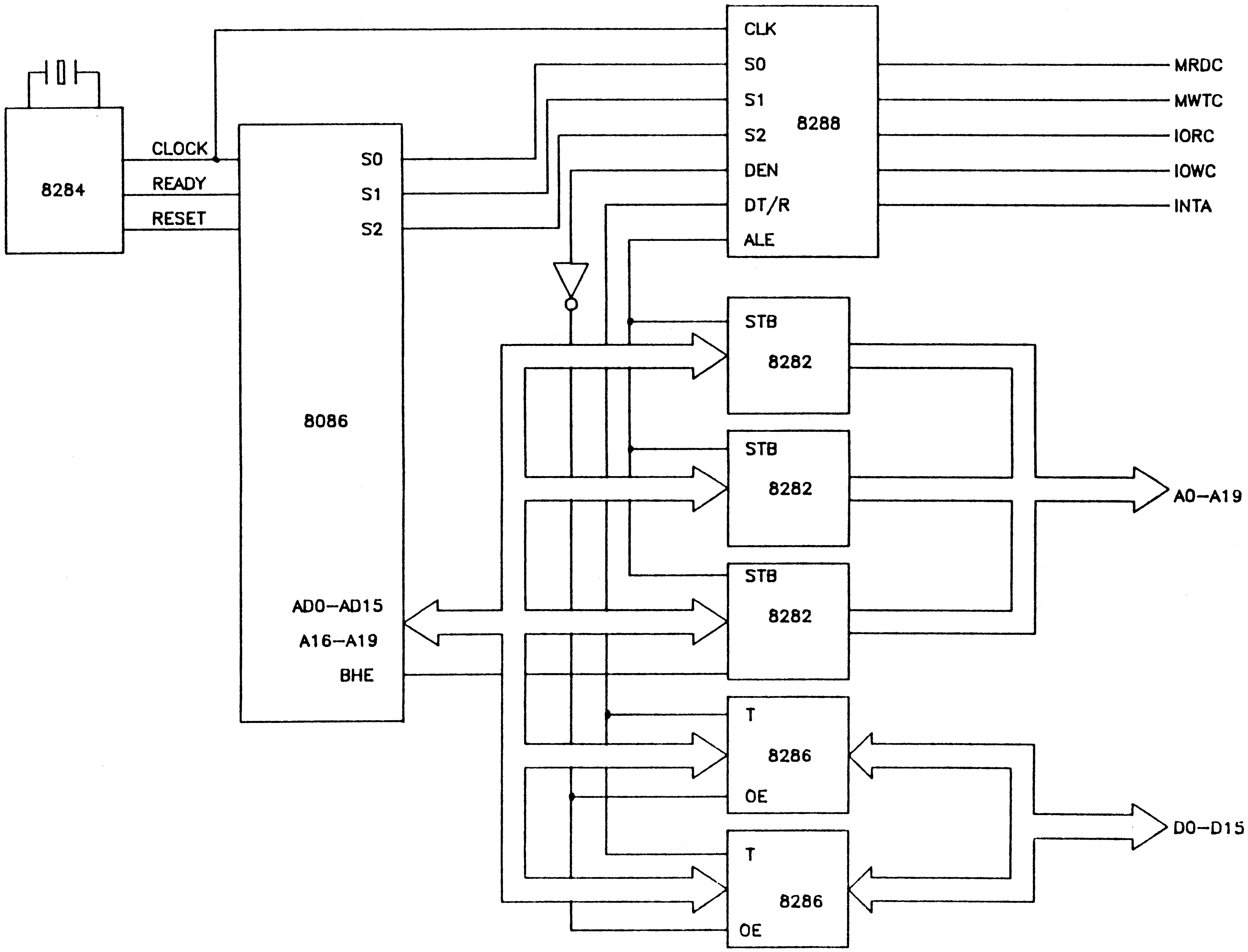


Fig. 4 – Z8000, Clock & Demux

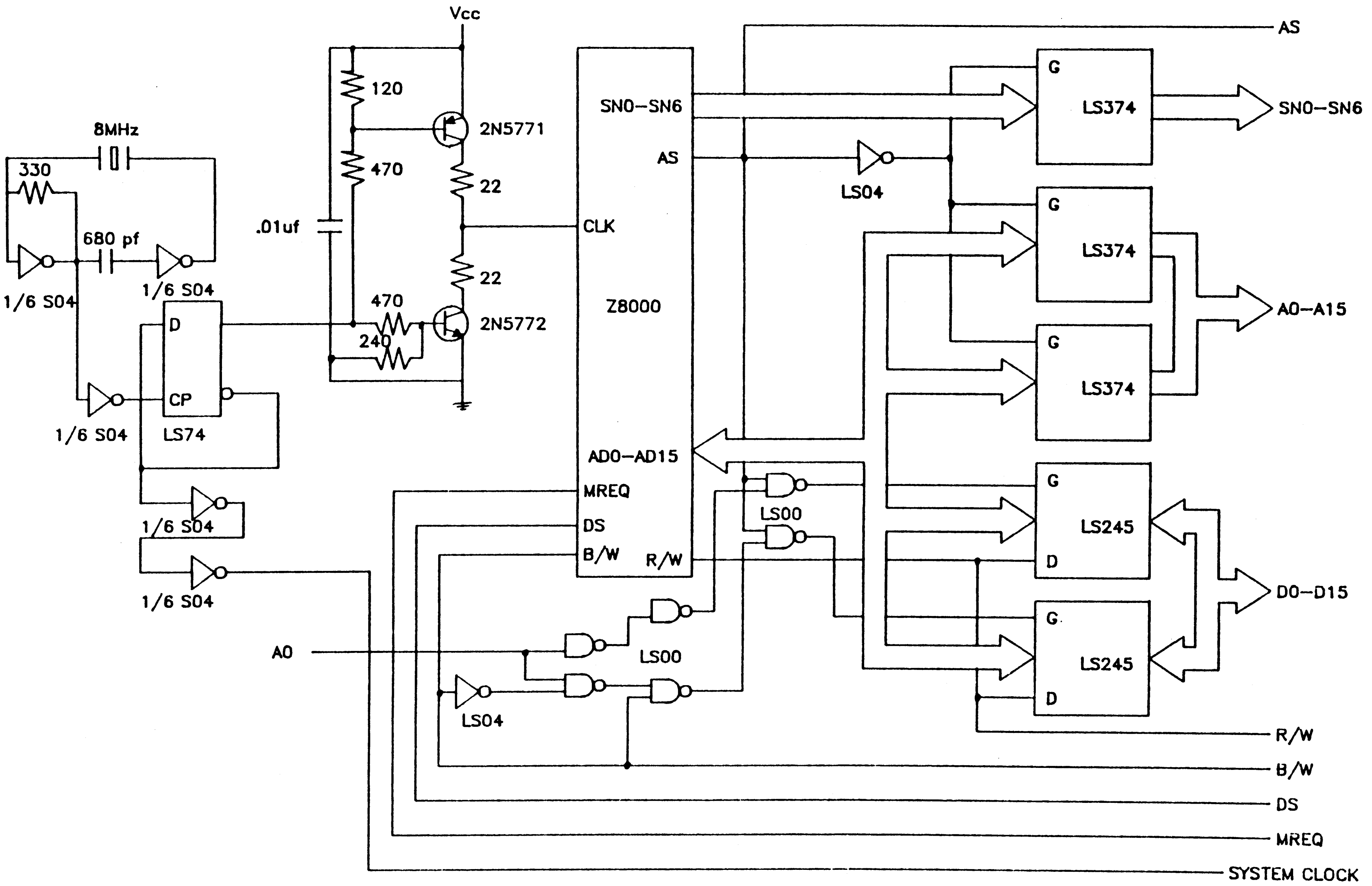


Fig. 5 – NS16032, Clock & Demux

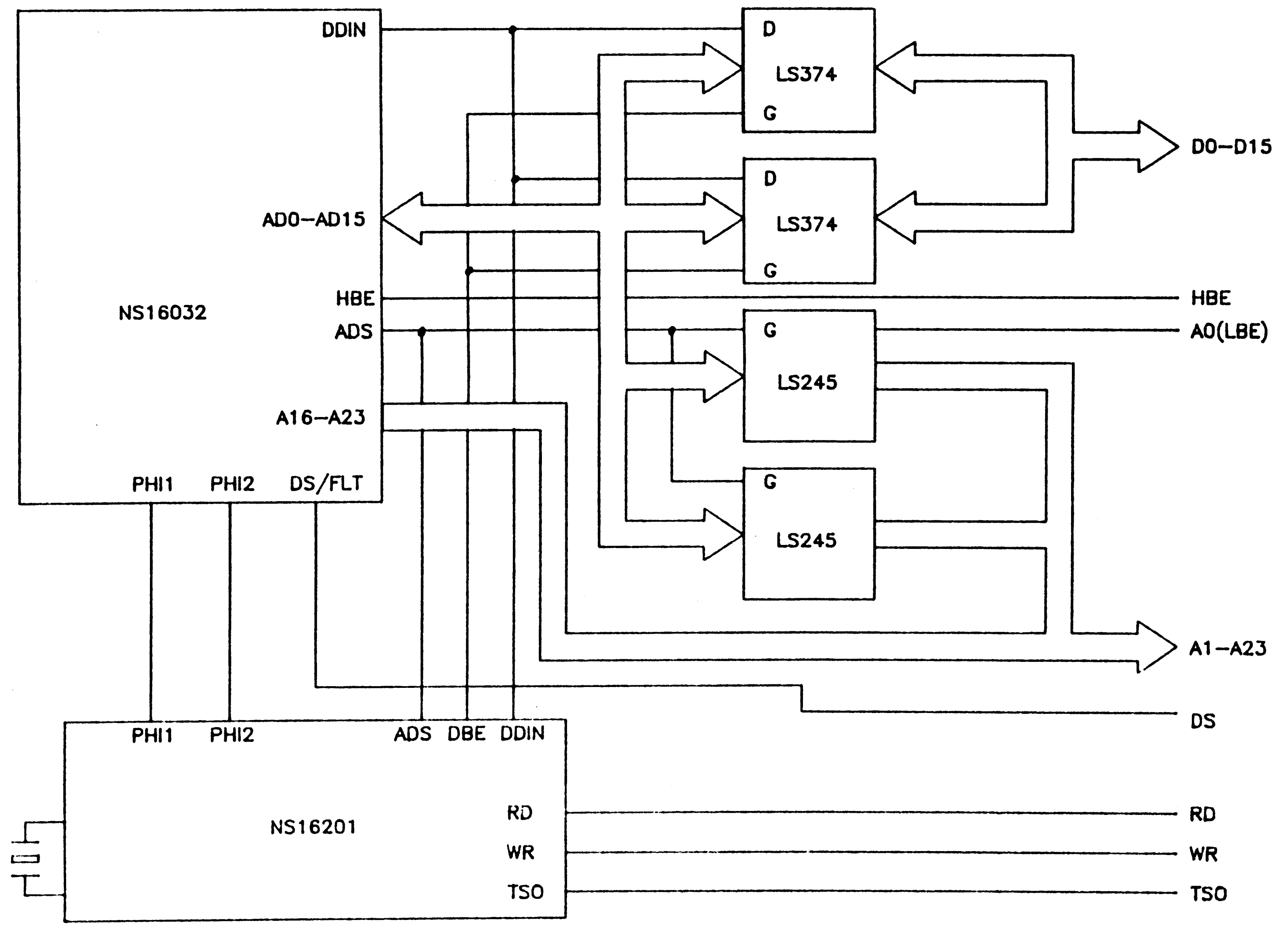
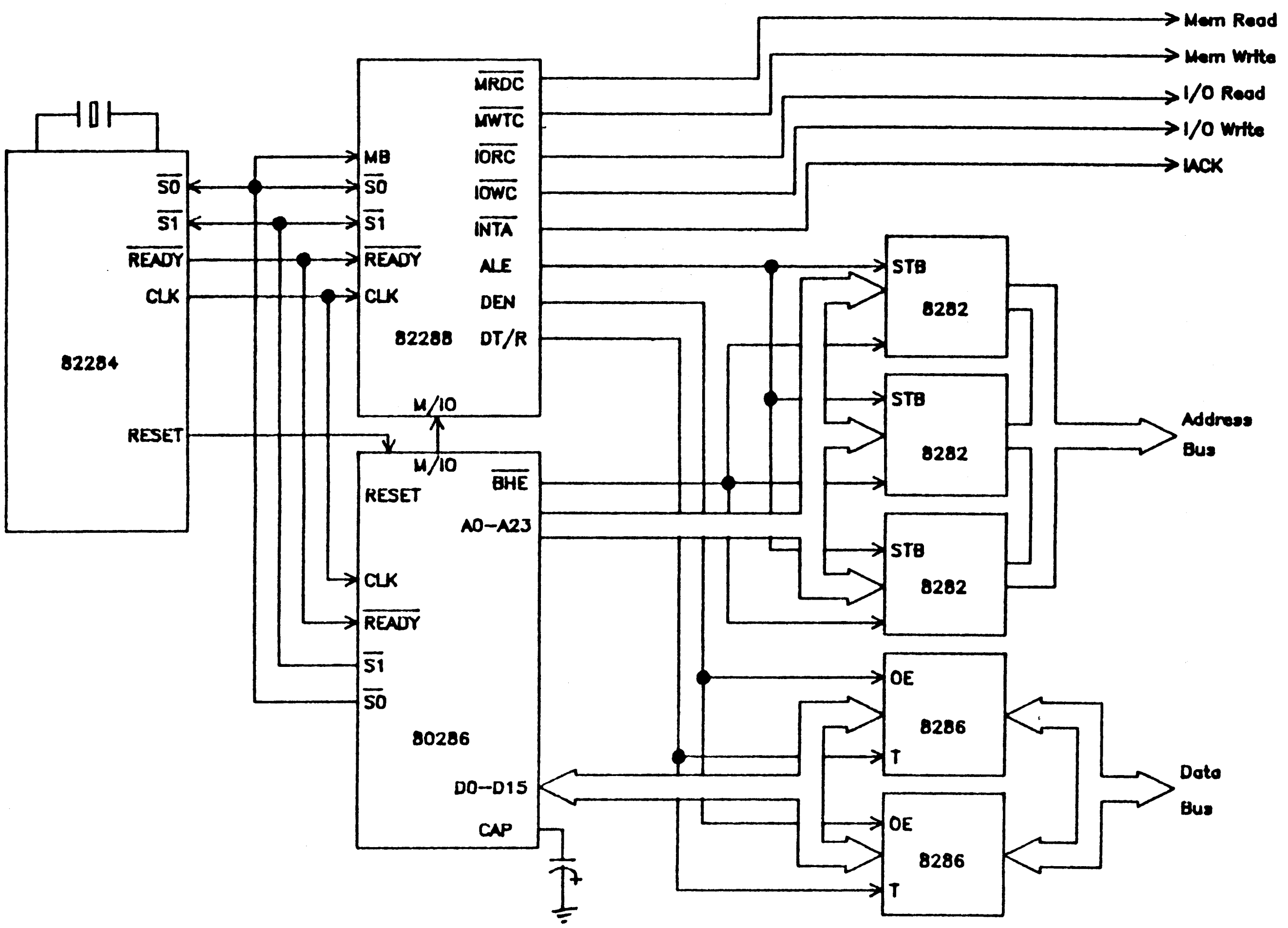


Fig. 6 – iAPX286, Clock & Latches



•Note: An 8259A must be added to allow Interrupt capability.

M68000
MICRO MINUTES

MM- 422-16

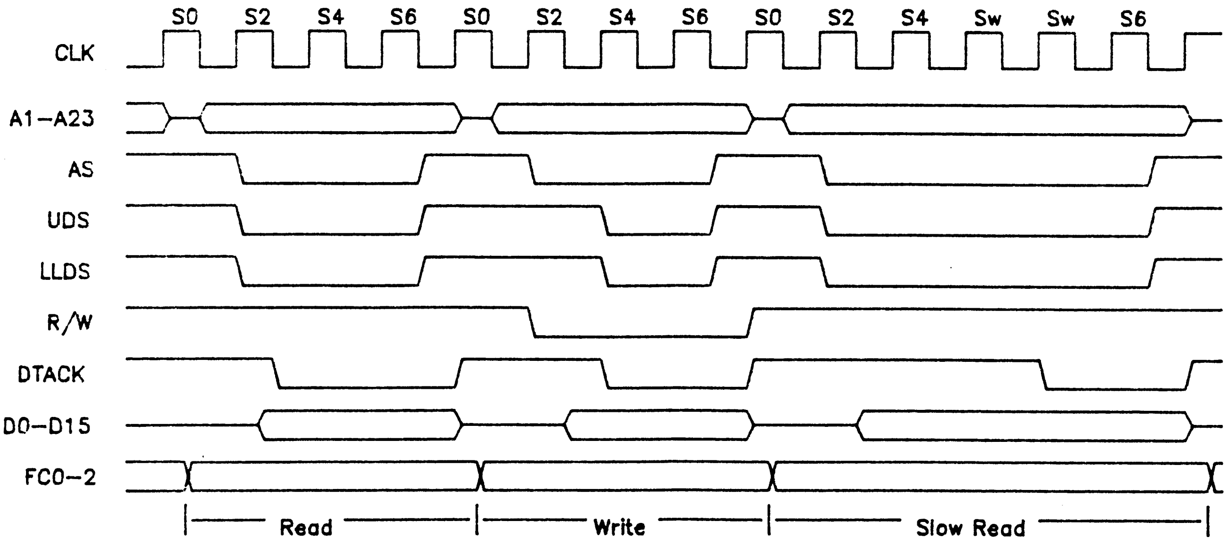
MEMORY SPEED REQUIREMENTS FOR 16-BIT MPUs

Questions have been received on the MC68000 concerning access times in relation to other 16-bit MPUs such as the i8086, iAPX286, Z8000 and NS16032. Note that other suppliers will have the same questions when they introduce higher speed versions of their MPUs (10 MHz). This Micro-Minutes details the questions and the answers concerning memory access times for 16-bit MPUs.

Background Specifications
Figure 1 shows read and write cycle timing for the MC68000. A basic MC68000 read cycle is initiated by the rising edge of S2 which causes AS to fall. If DTACK is asserted prior to the falling edge of S4, no wait states are inserted and data is strobed into the MPU on the falling edge of S6, terminating the bus cycle.

Timing for write cycles is similar. If DTACK is returned to the MPU before the falling edge of S4, the processor terminates the bus cycle on the falling edge of S6 and assumes that data was strobed into memory.

Figure 1 - MC68000 Read and Write Cycle Timing Diagram



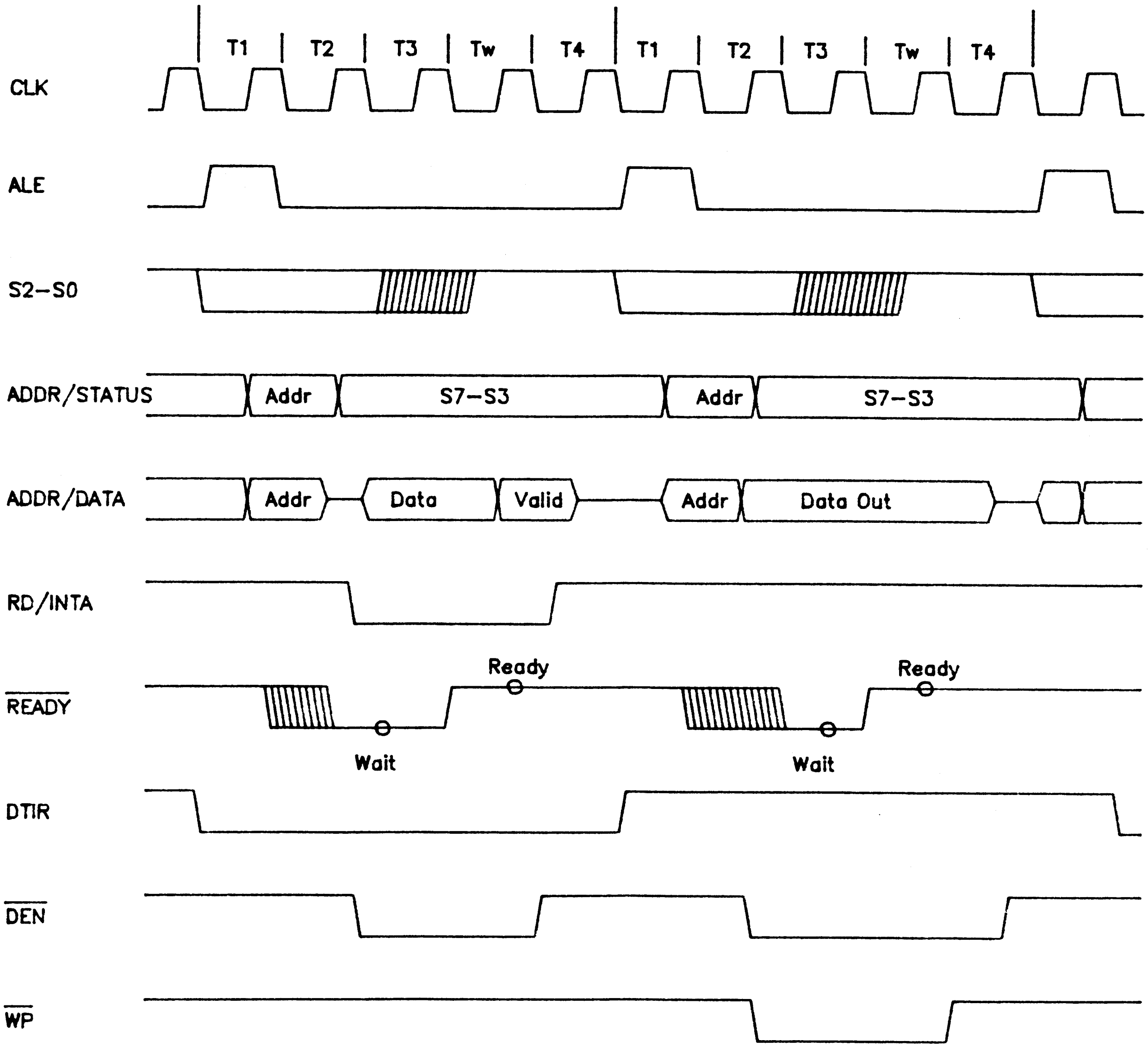
MOTOROLA
Austin, Texas

Processors - Competitive

Memory Speeds

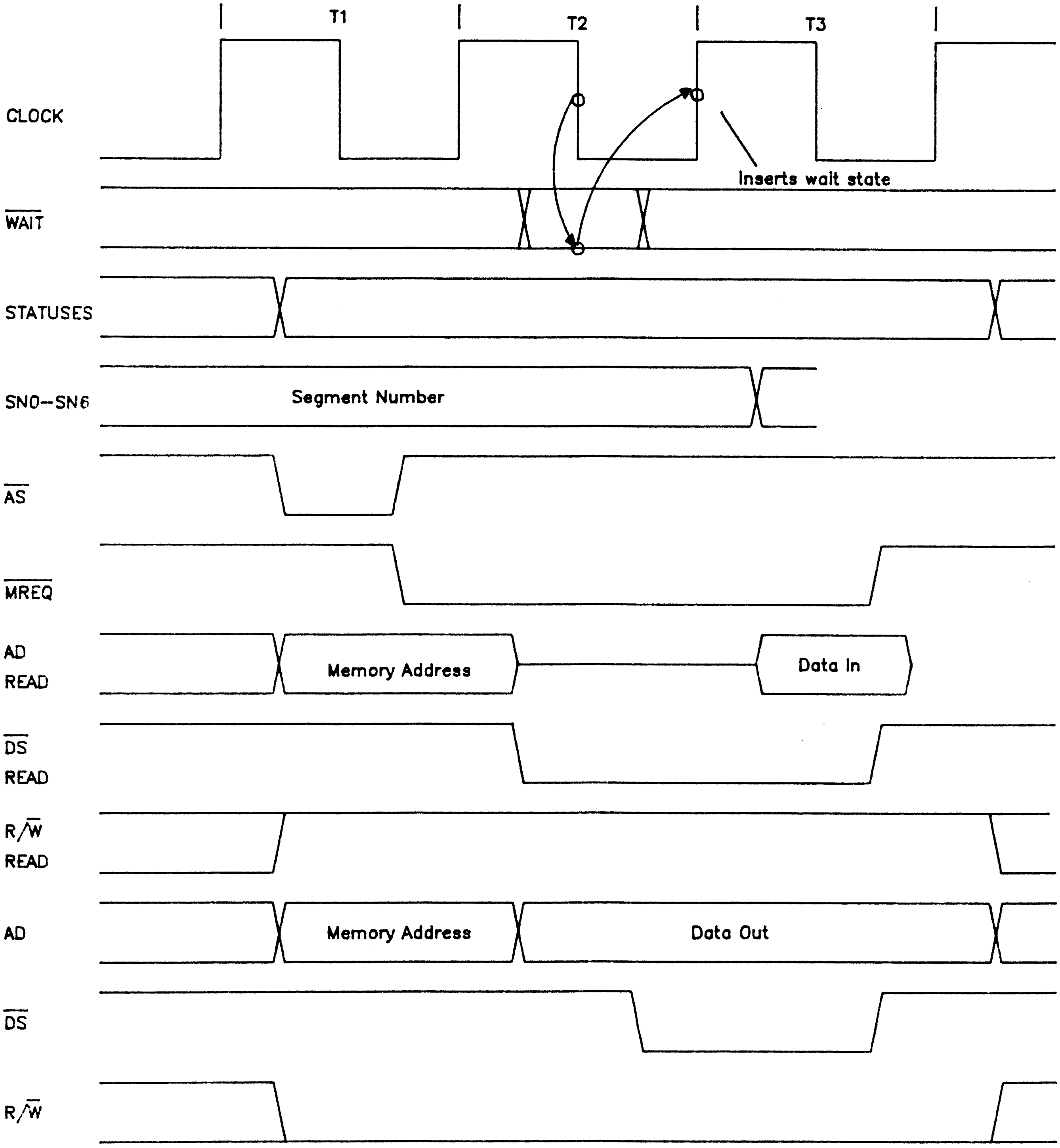
Figure 2 shows that i8086 bus cycles consist of four clock cycles. The falling edge of ALE strobes the address into parallel latches during T1. The data is strobed into the MPU on the falling edge of T3 if READY was asserted on the rising edge of T3. If READY is not asserted, wait states are inserted. Write operation timing is identical.

Figure 2 - i8086 Read and Write Cycle Timing Diagram



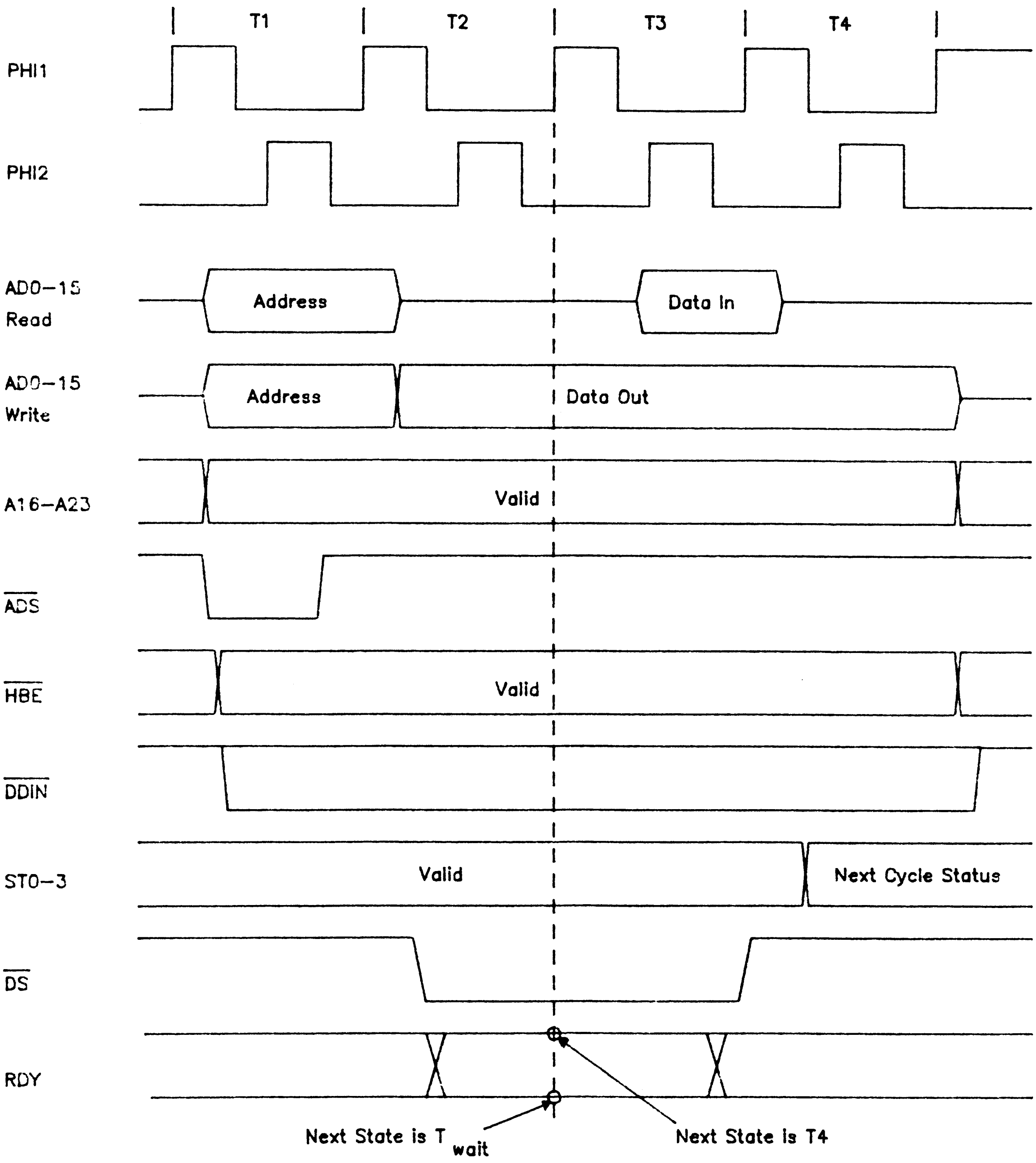
The basic Z8000 bus cycle consists of three clock periods as shown in Figure 3. AS is asserted to enable transparent latches. The rising edge of AS initiates the bus cycle. If WAIT is asserted on the falling edge of T2, a wait state (TW) will be inserted. WAIT is sampled on the falling edge of TW; if it is active, then additional wait states will occur. If WAIT is negated, then T3 will occur, terminating the cycle.

Figure 3 - Z8000 Read and Write Cycle Timing Diagram



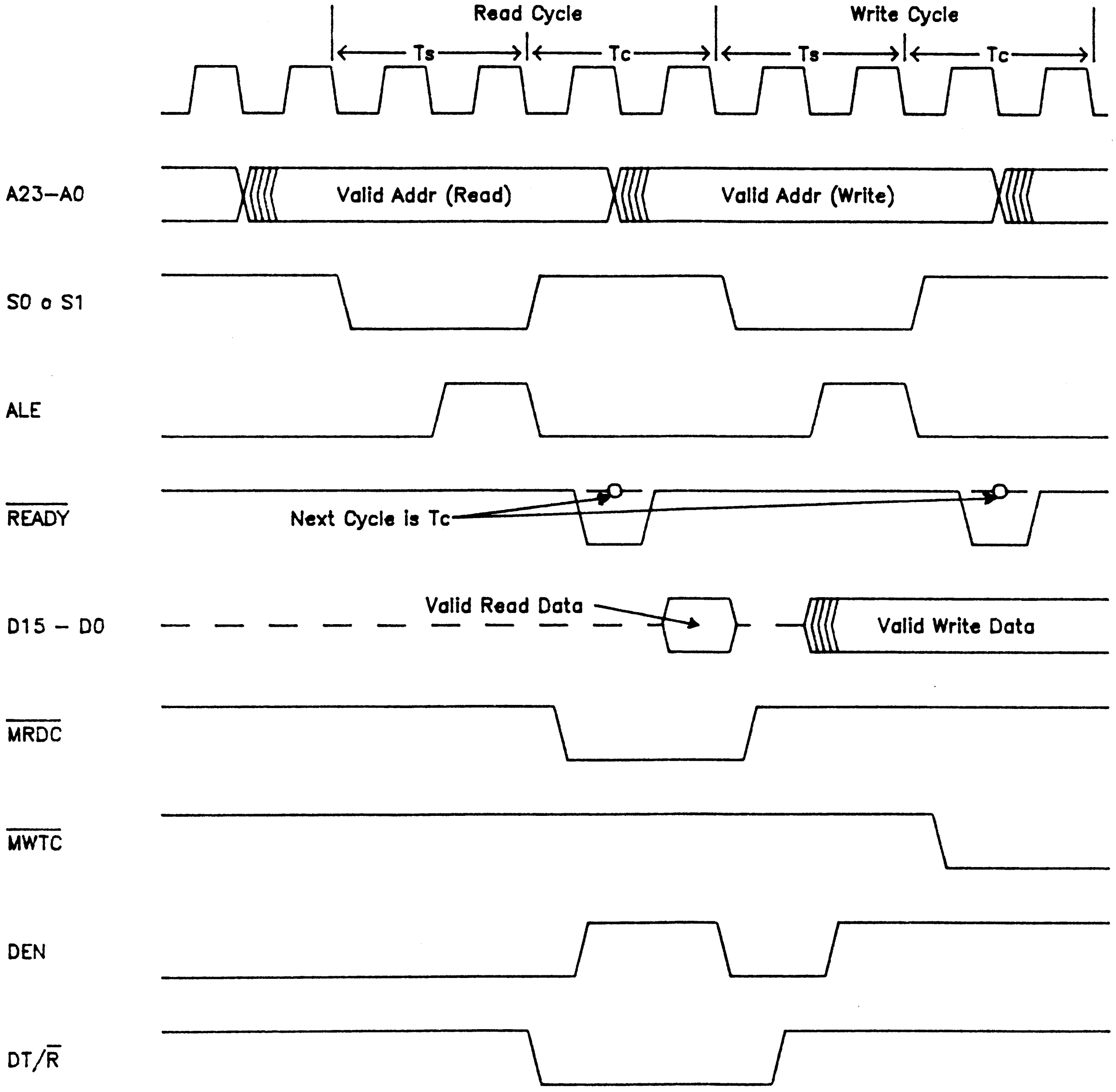
A basic cycle in the NS16032 MPU is performed in four cycles of the PHI1 clock. During T1 the processor places addresses on the bus and asserts a transparent latch control signal (ADS). At T2 the processor prepares to either accept or place data on the bus. At the rising edge of T3, the RDY line is sampled, and if inactive, the next cycle will be a wait cycle. Otherwise, on read cycles, data is latched toward the end of T3, and T4 terminates the cycle.

Figure 4 - NS16032 Read and Write Cycle Timing Diagram



Basic read and write cycles for the iAPX286 are shown in Figure 6. Here, the bus cycle is performed in 4 clock cycles of the system clock (CLK). Due to the "pipelining" of addresses, the addresses become valid sometime prior to the start of the bus cycle. The status lines (S0 and S1) indicate the type of cycle to be run. During the T_S portion of the bus cycle, addresses are propagated to memory, and during the T_C portion, the desired command is actually executed. The ALE signal may be used to control a transparent latch to allow addresses to remain valid to memory and at the same time, start changing for the next cycle. Read data is latched at the end of T_C.

Figure 6 - iAPX286 Read and Write Cycles Timing Diagram



Specification Summary

Note that each of these processors requires three clock periods with an AS (ALE, ADS) delay and a data setup time required except for the iAPX286. Equations 1 through 5 and table 1 sumarize the access times required for the MC68000 (1), i8086 (2), Z8000 (3), NS16032 (4) and iAPX286 (5).

$$t_{acc} = 2.5 \times t_{cyc} - t_{CHSL} - t_{DICL} \qquad (1)$$

$$t_{acc} = 2.5 \times t_{cyc} - t_{CHLL} - t_{DUCL} \qquad (2)$$

$$t_{acc} = 2.5 \times t_{cyc} - t_{dC(ASf)} - t_{sDI(C)} \qquad (3)$$

$$t_{acc} = 3 \times t_{cyc} - t_{Ahv} - t_{DIs} - t_{clr} \qquad (4)$$

$$t_{acc} = 5 \times t_{cyc} - t_{addr\ valid\ delay} - t_{data\ setup} \qquad (5)$$

Table 1 shows that variances an access time are so minor that similar speed RAMs must be used for similar clock frequencies.

Table 1 - Access Time (nsec) Summary

MPU	Clock Rate					
	4MHz	5MHz	6MHz	8MHz*	10Mhz*	12.5Mhz*
MC68000	515	405	321	237.5	180	135
i8086	580	455	385.7	277.5	215	165
Z8000	495	420	336.7	234	200	150
NS16032	596	476	356	251	176	116
iAPX286	475	400	315	242.5	180	130

* i8086 and Z8000 access times are calculations based on preliminary numbers. NS16032 and iAPX286 access times are straight line calculations using delay and setup times for the 8 MHz parts.

Applications Impact

These processors and the associated memory and I/O may be configured in one of two ways -- a single board system or a bus oriented system. Note that the i8086, Z8000 and NS16032 must pay a speed penalty of 30 ns to demultiplex the addresses and data. This reduces the available access time. The iAPX286 system shown (from their data sheet order number 210253-004 page 34), will not be able to utilize the full access times given, but for the sake of clarity, these numbers will be used).

The single board system, shown in Figure 5, allows the designer to use slower RAMs but the number of devices is limited due to board space, drive capability, etc. Table 2 shows the available access timing assuming demultiplexed address and data buses. Table 3 shows the required memory speed to run bus cycles without wait states (450 or 350 ns for EPROMs/ROMs and 300, 250, 200, 150 or 120 ns for RAMs).

Figure 5 - Single Board System Block Diagram

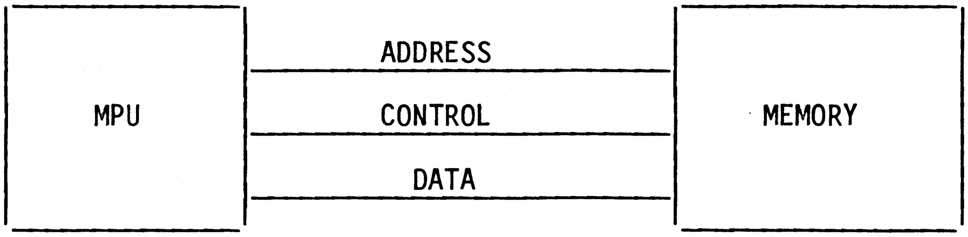


Table 2 - Access Time (nsec) Summary for Single Board Applications

MPU	Clock Rate					
	4MHz	5MHz	6MHz	8MHz*	10Mhz*	12.5Mhz*
MC68000	515	405	321	237.5	180	135
i8086	550	425	355.7	247.5	185	135
Z8000	465	390	316.7	204	170	120
NS16032	566	446	326	221	146	86
iAPX286	475	400	315	242.5	180	130

* i8086 and Z8000 access times are calculations based on preliminary numbers. NS16032 and iAPX286 access times are straight line calculations using delay and setup times for the 8 MHz parts.
Assumes 30 nanosecond impact for demultiplexing address and data buses on i8086, Z8000 and NS16032.

Table 3 - Maximum Allowable Memory Speeds for Single Board Applications

MPU	Clock Rate					
	4MHz	5MHz	6MHz	8MHz*	10Mhz*	12.5Mhz*
MC68000	450	350	300	200	150	120
i8086	450	350	350	200	150	120
Z8000	450	350	300	200	170	120
NS16032	450	350	300	200	120	**
iAPX286	450	350	300	200	150	120

* i8086 and Z8000 access times are calculations based on preliminary numbers. NS16032 access times are straight line calculations.
** Needs memories faster than 120 nanoseconds.

The multiboard system shown in Figure 6, allows for expansion as required by the end customer. The disadvantage is the delay required for a signal to travel through buffers and the bus to the destination. Table 4 shows the available access times assuming demultiplexed address and data buffer delays (35 ns), and bus delays (20 ns). Table 5 shows the required memory speed to run bus cycles without wait states.

Figure 6 - Multi-Board System Block Diagram

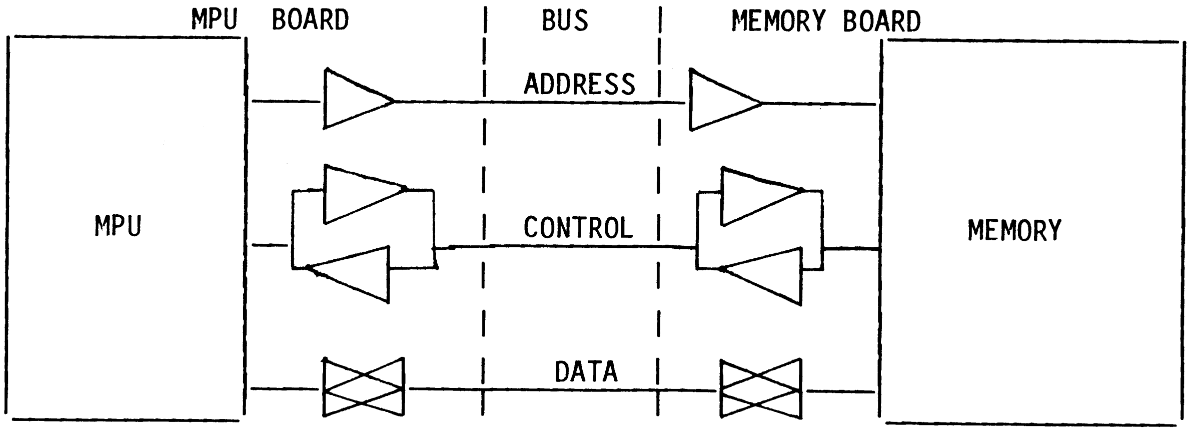


Table 4 - Access Time (nsec) Summary for Multi-Board Applications

MPU			Clock Rate			
	4MHz	5MHz	6MHz	8MHz*	10Mhz*	12.5Mhz*
MC68000	460	350	266	182.5	125	70
i8086	495	370	300.7	192.5	130	70
Z8000	410	335	251.7	149	115	65
NS16032	511	391	271	166	91	31
iAPX286	420	345	260	187.5	125	75

* i8086 and Z8000 access times are calculations based on preliminary numbers. NS16032 and iAPX286 access times are straight line calculations using delay and setup times for the 8 MHz parts.
Assumes 30ns impact for demultiplexing address and data buses on i8086, Z8000 and NS16032; 35ns bus buffering delays, and 20ns bus delays for all four processors.

Table 5 - Maximum Allowable Memory Speeds for Multi-Board Applications

MPU			Clock Rate			
	4MHz	5MHz	6MHz	8MHz*	10Mhz*	12.5Mhz*
MC68000	450	350	250	150	120	**
i8086	450	350	300	150	120	**
Z8000	350	300	250	120	**	**
NS16032	450	350	250	150	**	**
iAPX286	350	300	200	150	120	**

* i8086 and Z8000 access times are calculations based on preliminary numbers. NS16032 and iAPX286 access times are straight line calculations using delay and setup times for the 8 MHz parts.
** Needs memories faster than 120 nanoseconds.

Summary

The data presented shows that there is no appreciable difference in memory speed requirements for the MC68000, i8086, Z8000, NS16032 and iAPX286 processors, regardless of what the other manufacturers may be trying to say. Remember that the MC68000 has the highest performance of all of these machines, so dollar-for-dollar **THE MC68000 GIVES THE BIGGEST BANG FOR THE BUCK!!!**

M68000
MICRO MINUTES

MM- 422-26-A

DOES A 32 BIT DATA BUS MEAN PERFORMANCE?
A COMPARISON OF THE NATIONAL 32032 WITH THE 68000 FAMILY

National Semiconductor has recently made big claims with the announcement of their 32032 which is a "full" 32-bit version of their 16032. Although the 32032 offers no significant performance advantage over the 16032. National claims to be able to offer a 32-bit solution today.

The truth is that the National 32032 is little more than a 16-bit processor with a 32-bit data bus. It it true that National has working silicon on the 32032 as evidenced by an operational wirewrap board at Wescon. The fact is, however, that several important architectural flaws prevent the 32032 from providing a complete 32-bit solution.

It is important to note that the core of the 32032 is the 16032 processor. The only performance gain the 32032 can tout is the increase in performance by doubling the bus bandwidth which National has admitted to provide only an additional 15% - 23% true improvement. The 16032 processor was originally designed for a 16-bit bus so, in expanding to the 32-bit bus, no internal changes were made to take advantage of the higher bus bandwidth. As a result the 32032 makes very inefficient use of the bus as it spends much of its time inside its execution unit (i.e. it's instruction bound). This may be an advantage for multi-master systems but indicates that the CPU is limited by its low instruction processing capability. Performance estimates indicate that three 32032's are required to approach 32-bit performance of the 68020 (Table 1).

Another architectural limitation is that the two co-processors that National offers (16081 FPP, and 16082 PMMU) are the very same parts from the 16032 and thus support only 16 bit data paths. A fundamental concept of the co-processor is that it is an extension of the CPU architecture. It is easy to see that the 32032 is limited by the co-processor family to operate with a 16-bit data bus. This half bandwidth bus to the floating point unit is a significant bottleneck for system performance. In addition, National has announced that these co-processors will NOT be redesigned to take advantage of the 32 bit data bus. We can only surmise that the design resources do not exist to support such an undertaking thus the National commitment to a true 32 bit family is negligible.

As in the 16032 processor, National still maintains a 24-bit program counter. As an important element in any architecture, the limitation of the PC to 24 bits restricts the total architecture to 24 bits. This flaw, in and of itself, prohibits the 32032 from being a true 32-bit machine.

MOTOROLA
Austin, Texas

Processors - Competitive

32032

The 32032 uses only 24 bits of physical address in order to keep the pin count down to 48 pins. This restriction severely limits the addressing range and really only provides the user with 32-bit data manipulation capability. With the trend for providing large virtual addresses for multiple users, the 16 Mbyte limitation of the 32032 is restrictive to say the least. No such limitation exists with the 68020's 4 gigabyte addressing capability.

National has recently announced that 12 MHz 32032's will be available 1Q84. National has conceded that the 16000 was initially designed to operate at 10 MHz so 12 MHz parts are going to require significant effort. Motorola, on the other hand, currently offers the 68000 family processors at 12.5 MHz and will offer the 68020 at 16 MHz. Nonetheless, to achieve the performance of the 68020, as seen by the benchmark in Table 1, the 32032 would have to operate at 36 MHz, an absolutely unrealistic design goal.

What then does the 68020 have to offer that is so much better than the 32032? First, the 68020 was designed at the outset to be a high performance microprocessor. To paraphrase one of the system designers, "Whenever there was a design trade-off we opted for performance". This is typified in the 3.5 MIP (average instruction stream) estimates for the 68020 which is roughly a 3x performance increase over the 68000. National, on the other hand, offers only about a 15% - 23% performance increase over the 16032.

Motorola has a commitment to offer 32-bit peripherals such as the 68881 and a paged memory management unit. The Floating Point Processor (68881) offers a significant performance increase when coupled with the 68020. Unlike the National floating point solution, the 68881 offers a complete IEEE standard implemented completely in hardware. As documented by National, the NS16081 only supports a few floating point operations to the IEEE specification, and requires roughly 20 - 30 kbytes of support code to implement a system comparable to the 68881.

The 68881 offers everything in hardware. In addition, the National co-processor scheme does not allow concurrent operation of the CPU and the floating point unit due to the handshaking mechanism involved in data transfers. So, while performing time consuming floating point operations on the National part, the CPU must wait. The 68881 however, due to the general purpose nature of the 68020 co-processor interface, can release the CPU to continue the instruction flow thus allowing true concurrent operation.

Motorola will also offer demand paged memory management support for the 68020 that translates all 32 bits of address, provides access protection, and performs with minimal system overhead. More information on this part can be provided by Motorola by obtaining a non-disclosure agreement.

The key to the 68020 is performance. The Table 1 shows times for a typical 32-bit arithmetic instructon using National's 16032 and 32032 both with and without MMU and Motorola's 68010 and 68020 with and without MMU. In all cases, one wait state is added for MMU translation time.

RD

A MIP calculation is given for each processor for various addressing modes. An average MIP figure is then calculated over all of the addressing modes. Note that this is a very rough performance analysis as a single instruction is used over common addressing modes rather than a representative instruction mix. In any case, these numbers do show relative performance between CPU's with the 68020 as the clear cut performance leader. Figures 1 and 2 show, in bar chart form, the composite information found in Table 1.

Of particular interest is the fact that the 68010 and the 32032 perform at roughly the same MIP rate. This is due to the fact that the 32032 internal design is for 16-bit performance and the fact that Motorola provides the 12.5 MHz execution rate of the 68010.

National makes the claim to be the first commercially available 32-bit microprocessor. Do not be fooled into believing that 32-bit is synonymous with performance. The MIP calculations are proof that it takes more than a 32-bit data path to provide the high performance associated with a 32-bit processor. To summarize:

1. The 32032 provides the same performance as the 68010. This is only 15% - 20% greater than the 16032 and 3.5x less than the 68020.

2. The 32032 is only capable of a 24-bit address space limiting the user to 16 Mbytes of virtual addresses (same as our 16-bit 68000's).

3. The only 32032 32-bit feature is the data bus. This feature alone offers a modest improvement in performance yet doubles the memory cost in the system.

4. National only offers 16-bit co-processors, an important architectural element, with no plans to support 32-bit co-processors.

5. The National 16082 Memory Management Unit only operates with 24-bit addresses thus limiting the system to 16 Mbytes of real and virtual memory space.

RD

FIGURE 1
PROCESSOR PERFORMANCE
(NO MMU)

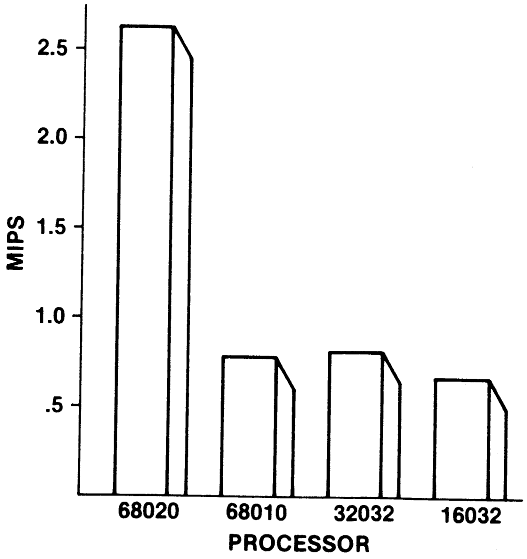
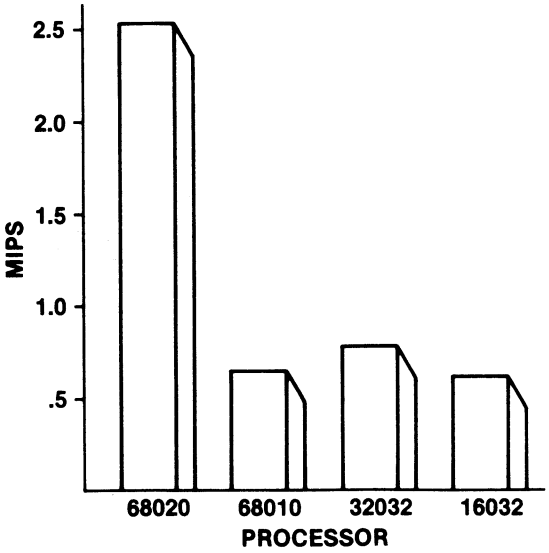


FIGURE 2
PROCESSOR PERFORMANCE
(WITH MMU)



MIP COMPARISON FOR 32-BIT ADD, COMMON ADDRESSING MODES [1], [2]

CPU	R,R	(R),R	R,(R)	-(R),-(R)	ABS,R	d(R:4,R),R	d(R),R	d(d(R),R	(R)+,R	-(R),R
68020	8.30	2.78	2.78	1.39	2.78	1.85	2.38	1.04	2.78	2.38
68020+PMMU	8.30	2.38	2.08	1.11	2.38	1.67	2.08	.93	2.38	2.08
68010	2.08	.89	.63	.42	.69	.39	.69	.37	.89	.78
68010+MMU	1.79	.74	.50	.33	.54	.34	.57	.30	.74	.66
32032	2.50	.83	.48	.34	.91	.50	.83	.59	.63	.63
32032+MMU	2.50	.77	.42	.31	.83	.48	.77	.53	.59	.59
16032	2.50	.63	.29	.24	.67	.42	.63	.40	.50	.50
16032+MMU	2.50	.56	.25	.21	.59	.38	.56	.34	.45	.45

AVERAGE MIPS FOR ADD INSTRUCTION

CPU	MIPS RATE
68020	2.63
68020+MMU	2.54
68010	.79
68010+MMU	.65
32032	.82
32032+MMU	.78
16032	.68
16032+MMU	.63

ASSUMPTIONS : 16032 and 32032 - 10 MHz Operation
68010 - 12.5 MHz Operation
68020 - 16.67 MHz Operation
All fetches hit in cache or queue.

TABLE 1

[1] 68020 instruction timings are calculated on microcode as of April 1983. Many instruction times have been changed due to constant optimization, and therefore may be different for production parts.

[2] National times were obtained via timing formulas contained in the "NS16000 Programmer's Reference Manual".

MC68020 SWEEPS the EDN BENCHMARK SERIES

This MICROminutes contains the results of the standard EDN benchmark series, and a "Burst" benchmark series for the MC68020 compared to the MC68010, NS32032, and iAPX286.

The timings (all in microseconds) were obtained as follows:

(1) The MC68020 and MC68010 times were obtained by actually running the code on a board.

(2) The iAPX286 times are Intel published times with the following exceptions: the timings for both the Linked List and QuickSort benchmarks vary from Intel published timings because they were recoded to remove the artificial 64k boundry imposed by Intel. In other words, the code will now run in a "normal" operating system environment.

(3) The 32032 times for the EDN series were obtained by applying a 22% performance improvement to previously published 32016 (aka 16032) timings. The Burst benchmark timings are hand calculated from the National 32032 timing tables.

It is interesting to note that the MC68010 was sufficient to better the times of the '286 on any of the EDN series, and was enough to beat even the 32032 on all execpt the Linked List and Matrix Inversion benchmarks (were the 32032 has special address modes/instructions to assist it). An average of all of the EDN benchmarks shows that the MC68010 performed at a respectable 42.1% of the MC68020, the NS32032 at only 30.1%, and the iAPX286 at only 25.1%. Our 16-bit virtual processor beat both of the highest performance processors from the competition.

The next set of benchmarks are burst rates for the performance of single activities (i.e. looping on a single instruction). The values above the graph elements are the MIPS rates for each test. Once again the MC68020 level of performance was rarely needed to better the competition. The average performance of the processors tested for all activites compared to the MC68020 shows the MC68010 at 22.4% of the '020, the iAPX286 at 20.2%, and the NS32032 bringing up the rear at 18.6%. The superior performance of the MC68020 in this sort of test shows the power of the instruction cache.

It is through the results of benchmarks like this that we at Motorola can honestly state the we have the highest performance microprocessor known to man...the MC68020 - **The 32-bit Performance Standard.**

Advantages of Upgrading an MC68000 to an MC68010

There are several ways a system's performance can be upgraded. Some are software related, such as lowering operating system overhead, obtaining better quality high-level language compilers, wisely designing application programs, and coding applications more efficiently. Others are hardware related, such as adding memory, improving I/O channel data rates, increasing mass storage speed and capacity, reducing memory access times, and upgrading the system processor's clock frequency.

A Look at the Obvious

When considering an MC68000 system upgrade to higher performance, the obvious thought is to redesign for a higher frequency MC68000. For example, a current MC68000 system running at 10 MHz could be redesigned to run at 12.5 MHz, thereby increasing system throughput by 25%. The "obvious solution", however, is not necessarily the most appropriate or cost-effective once several factors are taken into consideration and alternative solutions examined.

The speed-up of a system clock will not be effective unless the system's memory access time is also improved (see figure 1). The performance of the MC68000 is strictly limited by the bus speed, and if no improvement in memory speeds are available, then an increase in system clock speed will lead to negligible improvement in the overall result. A 10 MHz processor running with no "wait states" utilizes a 400ns bus cycle (4 clocks X 100 ns/clock). This **same** bus cycle timing, however, leads to a wait cycle on a 12.5 MHz processor (4 clocks x 80 ns/clock + 80 ns of idle time). Thus, the **bus** performance is **exactly** the same, but the faster processor is idled for one complete clock cycle. Since a decrease in the bus cycle time provides a directly proportional increase in processor throughput (until, of course, the memory cycle becomes faster than the fastest processor bus cycle), the 12.5 MHz processor has no relative performance advantage over the 10 MHz system. The bottom line, then, is that in order to be effective, a higher speed proccesor must run with fewer or the same number of "wait states". This normally requires a redesign of the memory subsystem to improve the memory access time.

Now to The Data Sheet...

Refering to the MC68000 Data Manual (ADI-814-R4), the memory access requirements for the various speed processors can be examined. The effective memory access time (t_{accs}) of the MC68000 to a memory array (from assertion of Address Strobe [$\overline{AS}$] to data valid) is:

$$t_{accs} = t_{ch} + 2\ t_{cyc} - t_{chsl} - t_{dicl} + (n * t_{cyc})$$

where: (numbers in parentheses indicate data sheet timing references)

t_{ch} (#3) is the clock high time (system dependent).

t_{cyc} (#1) is the clock period of the processor clock.

t_{chsl} (#9) is the delay time from the rising edge of the clock to the assertion of address strobe.

t_{dicl} (#27) is the data input set-up time prior to the falling edge of the clock.

n is the number of wait cycles in the system.

Assuming a symetric clock (50% duty cycle), the memory speed required for a no "wait-state" bus cycle for a 10 MHz MC68000 processor is 185ns (50 + 200 - 55 - 10 + 0). This bus speed can easily be realized with readily available 150ns dynamic RAMs and careful system design. However, with the same assumptions, the memory speed required for a no "wait-state" bus cycle on a 12.5 MHz processor is reduced to 135ns (40 + 160 - 55 - 10 + 0) which presents an obvious problem to the cost-conscious system designer - lack of cost-effective, large capacity 100ns RAMs!

Memory access times are not the only difficulty encountered with the faster clock speeds. In a similar vein, the design of an efficient 12.5 MHz system is more difficult than that of a 10 MHz system, since much more careful attention must be paid to the physical design of the board in order to account for the higher frequency signals present, and the increased sensitivity to transient phenomena.

And Now...an Alternative:

An "painless" alternative means to **effectively** increase system performance, is to upgrade to the MC68010 processor. The MC68010 at equal clock frequencies will run from 8% to 50% faster than an MC68000 without any user code changes. The speed-ups are due to several microcode enhancements: many 32-bit operations, conditional branches, multiply, divide, and other miscellaneous instructions run faster. Systems which use memory management can have **dramatic** improvements with slight operating system changes utilizing a few new MC68010 instructions such as "Move to/from Address Space" (MOVES).

Systems may see a significant improvement if they heavily utilize multiply, divide, and looping operations. Loops run from 23% to 80% faster once the microcode sets up the automatic "loop mode". Such loops benefit particular functions such as block moves, character matching and general string manipulation operations, and multiple-precision binary and packed BCD arithmetic. The new MC68010 multiply is 14 clocks faster, and the divide is 32 clocks faster than the MC68000. Programs utilizing (or with the potential of utilizing) such operations can obtain an increase in performance easily exceeding 10%.

An additional "plus" of the MC68010 is the provision of a clear path for the upgrade of current operating systems to full virtual operating systems utilizing the sophisticated virtual memory processing capabilities of the MC68010 (which is the same virtual environment offered by the 32-bit MC68020).

And, as an Added Bonus...

Since the MC68010 is pin-for-pin compatible with the MC68000, NO hardware redesign is necessary. Only very minor software changes may have to be made depending on operating system conventions. The MC68010 differs from the MC68000 in that: 1) a generic "vector word" has been added to the MC68010 stack frame; and 2) the MC68000's MOVE SR,[ea] has been made a privileged operation.

Easy software solutions for these two minor differences are: 1) any routines which build exception stacks (e.g. those which dispatch a routine via an RTE instruction) are modified to account for the four word stack frame (the MC68000 uses a three word stack frame); and 2) an exception handler is added to provide for privilege violations generated by the execution of the MOVE SR,[ea] instructions in the USER state (local Motorola representatives can supply a debugged handler to suit the requirements of any O.S.). Major operating systems have been ported from the MC68000 to the MC68010 in less than a single day, reflecting the trivial changes required in the supervisory level code.

The Conclusion:

The bottom line is, by upgrading an MC68000 system to an MC68010 system, an increase in system performance is obtained which is equal to that which a system redesign from 10 MHz to 12.5 MHz would provide, but with significantly less design cost and effort. The "speed-only" upgrade could only achieve, at best, a 25% system improvement, and only if the system memory access time is significantly improved. The MC68010 upgrade offers from 8% to 50% improvement. Note that the speed gained by changing to the MC68010 is achieved with NO change in memory speeds, NO board redesign, and NO higher speed parts installed in the system as would be required to upgrade a system to a 12.5 MHz part.

M68000 MICRO MINUTES

MM- 444-04

PGA PINOUTS FOR MC68440/442/450

Attached is the pinout for the MC68440, MC68442, and MC68450 DMA Controllers in 68-pin Pin Grid Array (PGA) packaging. One quick glance at the diagram will assure even the most hardened pessimist that all three parts are upward pin-for-pin compatible.

Basic Features

	MC68440	MC68442	MC68450
Bus Size (bits):	8/16	8/16	16
Operand Size (bits):	8/16	8/16	8/16/32
Word Transfers to/fm Odd Addresses:	NO	NO	YES
Address Sequencing:	UP/NO	UP/NO	UP/DOWN/NO
"MOVEP" Addressing:	NO	NO	YES
Continue Mode:	YES	YES	YES
Chaining Modes:	NO	NO	YES
Peripheral Requested Channel Restart:	YES	YES	NO
Channels:	2	2	2
Addressing Range:	16 Mbytes	4 Gbytes	16 Mbytes

Pin	Function		
	'440	'442	'450
A1	NC	FC3	
B1	A13/D5		
C1	A11/D3		
D1	A10/D2		
E1	A8/D0		
F1	A7		
G1	A6		
H1	A5		
J1	A3		
K1	NC	A25	
K2	$\overline{\text{BR}}$		
K3	$\overline{\text{UAS}}$		
K4	$\overline{\text{DBEN}}$		
K5	NC	A24	$\overline{\text{REQ3}}$
K6	NC	A26	$\overline{\text{REQ2}}$
K7	$\overline{\text{REQ0}}$		
K8	NC	A28	$\overline{\text{PCL3}}$
K9	$\overline{\text{PCL1}}$		
K10	$\overline{\text{DTACK}}$		
J10	$\overline{\text{UDS}}$/A0		
H10	$\overline{\text{AS}}$		
G10	R/$\overline{\text{W}}$		
F10	NC		
E10	$\overline{\text{CS}}$		
D10	CLK		
C10	$\overline{\text{IACK}}$		
B10	NC	A30	$\overline{\text{ACK3}}$
A10	$\overline{\text{ACK0}}$		
A9	$\overline{\text{BEC1}}$		
A8	FC2		
A7	FC1		
A6	A23/D15		
A5	A22/D14		
A4	A20/D12		

Pin	Function		
	'440	'442	'450
A3	A19/D11		
A2	A17/D9		
B2	A15/D7		
C2	A12/D4		
D2	A9/D1		
E2	GND		
F2	Vcc		
G2	A4		
H2	A2		
J2	$\overline{\text{BG}}$		
J3	$\overline{\text{OWN}}$		
J4	$\overline{\text{HIBYTE}}$		
J5	$\overline{\text{DDIR}}$		
J6	$\overline{\text{REQ1}}$		
J7	NC	A29	$\overline{\text{PCL2}}$
J8	$\overline{\text{PCL0}}$		
J9	NC	A27	
H9	$\overline{\text{BGACK}}$		
G9	$\overline{\text{LDS/DS}}$		
F9	GND		
E9	Vcc		
D9	$\overline{\text{DONE}}$		
C9	$\overline{\text{IRQ}}$		
B9	NC	A31	$\overline{\text{ACK2}}$
B8	$\overline{\text{BEC2}}$		
B7	$\overline{\text{BEC0}}$		
B6	FC0		
B5	A21/D13		
B4	A18/D0		
B3	A16/D8		
C3	A14/D6		
H3	A1		
H8	$\overline{\text{DTC}}$		
C8	$\overline{\text{ACK1}}$		