33 MHz 386 System Design Considerations

SHAHZAD BAQAI
KIYOSHI NISHIDE

May 1990
33 MHz 386 SYSTEM
DESIGN CONSIDERATIONS

CONTENTS

1.0 INTRODUCTION ...................................... 1

2.0 HIGH SPEED SYSTEM DESIGN
CONSIDERATIONS ....................................... 3
   2.1 Overview of High Speed Effects ..................... 3
   2.2 Transmission Line Effects ........................... 6
   2.3 Reflection ......................................... 7
   2.4 Cross Talk ......................................... 12
   2.5 Skew ............................................... 12
   2.6 D.C. Loading ....................................... 13
   2.7 A.C. (Capacitive) Loading .......................... 13
   2.8 Derating Curve ..................................... 13
   2.9 High Speed Clock Circuits .......................... 14

3.0 DESIGN EXAMPLE ................................. 16
   3.1 System Architecture for High Speeds ............... 16
   3.2 CPU Subsystem ..................................... 16
   3.3 DRAM Subsystem .................................. 16
   3.4 Cache Subsystem .................................. 18
   3.5 I/O-EPROM Subsystem ............................. 24

APPENDIX A
Schematics ............................................. A-1

APPENDIX B
State Diagrams and Palcodes ......................... B-1

APPENDIX C
Timing Diagrams ....................................... C-1

APPENDIX D
Timing Equations ...................................... D-1

APPENDIX E
References ............................................. E-1
SECTION I. INTRODUCTION

The 386 DX Microprocessor is an advanced 32-bit microprocessor designed using Intel's CHMOS IV process for applications which require very high performance. It is optimized for multitasking operating systems. The 32-bit register and data paths support 32-bit address and data types allowing up to four gigabytes of physical memory and 64 terabytes of virtual memory to be addressed. The integrated memory management and protection architecture includes address translation registers, advanced multitasking hardware and a protection mechanism to support operating systems. In addition, the 386 DX microprocessor allows the simultaneous running of DOS with other operating systems.

Instruction pipelining, on chip address translation and high bus bandwidth ensure short average instruction execution times and high system throughput. To facilitate high performance system hardware designs, the 386 DX microprocessor bus interface offers address pipelining, dynamic data bus sizing and direct byte enable signals for each byte of the data bus.

This Application Note is intended to show how to complete a successful design of a 'Core' system using the 386 DX-33, the 33 MHz clock version. A Core system is a minimum system configuration, in this case comprising the CPU, the 82385 32-bit Cache controller, Dynamic and Static RAM and an I/O mechanism with which to communicate with the CPU.

The Application Note examines the design techniques necessary when executing a design at this frequency. Many of the methods used at lower frequencies, such as 16 MHz and 20 MHz, are no longer valid at this higher frequency. Phenomena, whose effects are negligible at the lower frequencies, must be taken into account in the design. The physical positioning of components relative to each other plays a significant part in the success of the design, since transmission line effects (reflection, radiation) are no longer negligible.
Figure 1-1. Functional Signal Groups

Figure 1-2. CLK2 Signal and Internal Processor Clock
SECTION II. HIGH SPEED SYSTEM DESIGN CONSIDERATIONS

2.1 Overview Of High Speed Effects

This section is included as a brief overview of general issues that are applicable to both higher and lower frequencies of circuit design.

The CHMOS IV 386 DX CPU differs from previous HMOS microprocessors in that its power dissipation is primarily capacitive; there is almost no DC power dissipation. Power dissipation depends mostly on frequency. This fact is used in designs where power consumption is critical.

Power dissipation can be distinguished as either internal (logic) power or I/O (bus) power. Internal power varies with operating frequency and to some extent with wait states and software. Internal power increases with supply voltage also. Process variations in manufacturing affect internal power, although to a lesser extent than with NMOS processes.

I/O power, which accounts for roughly one-fifth of the total power dissipation, varies with frequency and voltage. It also depends on capacitive bus load. Capacitive bus loadings for all output pins are specified in the 386 DX CPU data sheet. The 386 DX CPU output valid delays will increase if these loadings are exceeded. The addressing pattern of the software can affect I/O power by changing the effective frequency at the address pins. The variation in frequency at the data pins tends to be smaller; thus varying data patterns should not cause a significant change in power dissipation.

POWER AND GROUND PLANES

Power and ground planes must be used in 386 DX CPU systems to minimize noise. Power and ground lines have inherent inductance and capacitance, therefore an impedance \( z = (L/C)^{1/2} \). The total characteristic impedance for the power supply can be reduced by adding more lines. This effect is illustrated in 2.1 which shows that two lines in parallel have half the impedance of one. To reduce the impedance even further, the user should add more lines. In the limit, an infinite number of parallel lines, or a plane, results in the lowest impedance. Planes also provide the best distribution of power and ground.

The 386 DX CPU has 20 V\(_{\text{CC}}\) pins and 21 V\(_{\text{SS}}\) (ground) pins. All power and ground pins must be connected to a plane. Ideally, the 386 DX CPU is located at the center of the board, to take full advantage of these planes. Although the 386 DX CPU generally demands less power than the 80286, the possibility of power surges is increased due to higher frequency and pin count. Peak-to-peak noise on V\(_{\text{CC}}\) relative to V\(_{\text{SS}}\) should be maintained at no more than 400 mV, and preferably to no more than 200 mV.

DECOUPLING CAPACITORS

The switching activity of one device can propagate to other devices through the power supply. For example, in the TTL NAND gate of Figure 2.2, both Q3 and Q4 transistors are on for a short time when the output is switching. This increased load causes a negative spike on V\(_{\text{CC}}\) and a positive spike on ground.
In synchronous systems in which many gates switch simultaneously, the result is significant noise on the power and ground lines.

Decoupling capacitors placed across the device between Vcc and ground reduce Voltage spikes by supplying the extra current needed during switching. These capacitors should be placed close to their devices because the inductance or connection lines negates their effect.

When selecting decoupling capacitors, the user should provide 0.01 microfarads for each device and 0.1 microfarads for every 20 gates. Radio-frequency capacitors must be used; they should be distributed evenly over the board to be most effective. In addition, the board should be decoupled from the external supply line with a 2.2 microfarad capacitor.

Chip capacitors (surface-mount) are preferable because they exhibit lower inductance and require less total board space. They should be connected as in Figure 2.3. Leaded capacitors can also be used if the leads are kept as short as possible. Six leaded capacitors are required to match the effectiveness of one chip capacitor, but because only a limited number can fit around the 386 DX, the configuration in Figure 2.4 results.
HIGH FREQUENCY DESIGN CONSIDERATIONS

At high signal frequencies, the transmission line properties of signal paths in a circuit must be considered. Reflections, interference, and noise become significant in comparison to the high-frequency signals. They can cause false signal transitions, data errors, and input voltage level violations. These errors can be transient and therefore difficult to debug. In this section, some high-frequency design issues are discussed. Their effects and ways to minimize will be introduced in the next section.

REFLECTION AND LINE TERMINATION

Input voltage level violations are usually due to voltage spikes that raise input voltage levels above the maximum limit (overshoot) and below the minimum limit (undershoot). These voltage levels can cause excess current on input gates that results in permanent damage to the device. Even if no damage occurs, most devices are not guaranteed to function as specified if input voltage levels are exceeded.

Signal lines are terminated to minimize signal reflections and prevent overshoot and undershoot. If the round-trip signal path delay is greater than the rise time or fall time of the signal, terminate the line. If the line is not terminated, the signal reaches its high or low level before reflections have time to dissipate, and overshoot and undershoot occur. There are a few termination techniques that are used in different applications, these will be discussed in the next section.

INTERFERENCE

Interference is the result of electrical activity in one conductor causing transient voltages to appear in another conductor. It increases with frequency and closeness of the two conductors.

There are two types of interference to consider in high frequency circuits: electromagnetic interference (EMI) and electrostatic interference (ESI).

EMI (also called crosstalk) is caused by the magnetic field that exists around any current carrying conductor. The magnetic flux from one conductor can induce current in another conductor, resulting in transient voltage. Several precautions can minimize EMI.

Running a ground line between two adjacent lines wherever they traverse a long section of the circuit board. The ground line should be grounded at both ends.

Running ground line between the lines of an address bus or a data bus if either of the following conditions exist.
— The bus is on an external layer of the board.
— The bus is on an internal layer but not sandwiched between power and ground planes that are at most 10 mils away.

Avoiding closed loops in signal paths (see Figure 2.5). Closed loops cause excessive current and create inductive noise, especially in the circuitry enclosed by a loop.

Figure 2-5. Avoid Closed-Loop Signal Paths

ESI is caused by the capacitive coupling of two adjacent conductors. The conductors act as the plates of a capacitor; a charge built up on one induces the opposite charge on the other.

The following steps reduce ESI:

Separating signal lines so that capacitive coupling becomes negligible.

Running a ground line between two lines to cancel the electrostatic fields.

LATCHUP

Latchup is a condition in a CMOS circuit in which VCC becomes shorted to Vss. Intel's CHMOS IV process is immune to latchup under normal operating conditions. Latchup can be triggered when the voltage limits on I/O pins are exceeded, causing internal PN junctions to become forward biased. The following guidelines help prevent latchup:

Observing the maximum rating for input voltage on I/O pins.

Never applying power to an 386 DX CPU pin or a device connected to an 386 DX CPU pin before applying power to the 386 DX CPU itself.

Preventing overshoot and undershoot on I/O pins by adding line termination and by designing to reduce noise and reflection on signal lines.
THERMAL CHARACTERISTICS

The thermal specification for the 386 DX CPU defines the maximum case temperature. This section describes how to ensure that an 386 DX CPU system meets this specification.

Thermal specifications for the 386 DX CPU are designed to guarantee a tolerable temperature at the surface of the 386 DX CPU chip. This temperature (called the junction temperature) can be determined from external measurements using the known thermal characteristics of the package. Two equations for calculating junction temperature are as follows:

\[ T_j = T_a + (\@ja \times PD) \]
\[ T_j = T_c + (\@jc \times PD) \]

where:
- \( T_j \) = Junction Temperature
- \( @ja \) = Junction to ambient temperature coeff.
- \( T_c \) = Case Temperature
- \( T_a \) = Ambient Temperature
- \( @jc \) = Junction to Case
- PD = Power Dissipation temperature coeff.

Case temperature calculations offer several advantages over ambient temperature calculations.

Case temperature is easier to measure accurately than ambient temperature because the measurement is localized to a single point (top center of the package).

The worst-case junction temperature (\( T_j \)) is lower when calculated with case temperature for the following reasons:
- The junction-to-case thermal coefficient (\( @jc \)) is lower than the junction-to-ambient thermal coefficient (\( @ja \)); therefore, calculated junction temperature varies less with power dissipation (PD).
- \( @ja \) is not affected by airflow in the system; \( @ja \) varies with air flow.

With the case-temperature specification, the designer can either set the ambient temperature or use fans to control case temperature. Finned heat sinks or conductive cooling may also be used in environments where the use of fans is precluded. To approximate the case temperature for various environments, the two equations above should be combined by setting the junction temperature equal for both, resulting in this equation:

\[ T_a = T_c - ((@ja - @jc) \times PD) \]

The current data sheet should be consulted to determine the values of \( @ja \) (for the system’s air flow) and ambient temperature that will yield the desired case temperature. Whatever the conditions are, the case temperature is easy to verify.

2.2 Transmission Line Effects

As a general rule, any interconnection is considered a transmission line when the time required for the signal to travel the length of the interconnection is greater than one-eighth of the signal rise time. (True K. M., “Reflection: Computations and Waveforms, The Interface Handbook”, Fairchild Corp, Mountain View, CA, 1975, Ch. 3). As frequencies increase, designers must account for the negative effects associated with transmission lines. The section that follows will attempt to describe these effects and provide some suggestions for minimizing their negative effect on the system.

Before describing each effect, it is important to know how to characterize a trace on different types of transmission lines. This includes knowing the characteristic impedance of a trace, \( Z_o \), and the propagation delay for a given trace, \( t_{pd} \). These parameters will be used in determining what effects must be accounted for and to select component values used in minimizing the effects.

TRANSMISSION LINES TYPES

Although many types of transmission lines (conductors) exist, those most commonly used on the printed circuit boards are microstrip lines, strip lines, printed circuit traces, side-by-side conductors and flat conductors.

MICRO STRIP LINES

The micro strip trace consists of a signal plane that is separated from a ground plane by a dielectric as shown in Figure 2.6. G-10 fiber-glass epoxy, which is most common, has an \( \varepsilon_r = 5 \) where \( \varepsilon_r \) is the dielectric constant of the insulation. Let:
- \( w \) = the width of the signal line (inches)
- \( t \) = the thickness of copper
- \( h \) = the height of dielectric for controlled impedance (inches)
The characteristic impedance $Z_0$, is a function of dielectric constant and the geometry of the board. This is given by:

$$Z_0 = \frac{(87/(\varepsilon_r + 1.41))^{1/2} \ln (5.98/0.8 \, w + t) \, \Omega}{\varepsilon_r}$$

where $\varepsilon_r$ is the relative dielectric constant of the board material.

The propagation delay ($t_{pd}$) associated with the trace is a function of the dielectric only.

$$t_{pd} = 1.017 \, (0.475\varepsilon_r + 0.87) \, \frac{\text{ns}}{\text{ft}}$$

### STRIP LINES

A strip line is a strip conductor centered in a dielectric medium between two voltage planes. The characteristic impedance is given by:

$$Z_0 = \frac{60}{(\varepsilon_r)^{1/2}} \ln (5.98b/(0.8W + t)) \, \Omega$$

where $b$ = distance between the planes for the controlled impedance as shown in Figure 2.10

The propagation delay is given by:

$$t_{pd} = 1.017 \, (\varepsilon_r) \, \frac{\text{ns}}{\text{ft}}$$

Typical values of the characteristic impedance and propagation delay of these types of lines are:

- $Z_0 = 50\, \Omega$
- $t_{pd} = 2 \, \text{ns/ft}$ (or 6 in/ns)

### 2.3 Reflection

The first effect is reflection. As the name indicates it is the reflection of a signal as it propagates down the trace. The reflection results from a mismatch in impedance. The impedance of a transmission line is a function of the geometry of the line, its distance from the ground plane, and the loads long the line. Any discontinuity in the impedance will cause reflections.
Impedance mismatch occurs between the transmission line characteristic impedance and the input or output impedance of the devices that are connected to the line. The result is that the signals are reflected back and forth on the line. These reflections can attenuate or reinforce the signal depending upon the phase relationships. The results of these reflections include overshoot, undershoot, ringing and other undesirable effects.

At lower edge rates, the effects of these reflections are not severe. However at higher rates, the rise time of the signal is short with respect to the propagation delay. Thus it can cause problems as shown in Figure 2-8.

Overshoot occurs when the voltage level exceeds the maximum (upper) limit of the output voltage, while undershoot occurs when the level passes below the minimum (lower) limit. These conditions can cause excess current on the input gates which results in permanent damage to the device.

The magnitude of a reflection is usually represented in terms of a reflection coefficient. This is illustrated in the following equations:

\[
T = \frac{V_r}{V_i} = \text{Reflected voltage/incident voltage}
\]

\[
T_{load} = \frac{(Z_{load} - Z_0)/(Z_{load} + Z_0)}{(Z_{load} + Z_0)/(Z_{load} - Z_0)}
\]

\[
T_{source} = \frac{(Z_{source} - Z_0)/(Z_{source} + Z_0)}{(Z_{source} + Z_0)/(Z_{source} - Z_0)}
\]

Reflections voltage \( V_f \) is given by \( V_f \), the voltage incident at the point of the reflections, and the reflection coefficient.

The model transmission line can now be completed. In Figure 2-9, the voltage seen at point A is given by the following equation:

\[
V_a = V_s + \frac{Z_o}{Z_o + Z_0}
\]

This voltage \( V_s \) enters the transmission line at “A” and appears at “B” delayed by \( t_{pd} \).
where $x = \text{distance along the transmission line from point "A"}$ and $H(t)$ is the unit step function. The waveforms encounter the loads $Z_L$, and this may cause reflection. The reflected wave enters the transmission line at “B” and appears at point “A” after time delay ($\text{tpd}$):

$$V_1 = T_{\text{load}} \cdot V_b$$

This phenomenon continues infinitely, but it is negligible after 3 or 4 reflections. Hence:

$$V_{r2} = T_{\text{source}} \cdot V_{r1}$$

Each reflected waveform is treated as a separate source that is independent of the reflection coefficient at that point and the incident waveform. Thus the waveform from any point and on the transmission line and at any given time is as follows:

$$V(x,t) = \left(\frac{Z_O}{Z_O + Z_L}\right) V_s(t - \frac{2L}{v} t) H(t - \frac{2L}{v}) + T_1 \left[ V_s(t - (2L + x)/v) H(t - (2L + x)/v) \right] + T_1 T_2 \left[ V_s(t - (4L + x)/v) H(t - (4L + x)/v) \right] + T_1 T_2 T_3 \left[ V_s(t - (6L + x)/v) H(t - (6L + x)/v) \right] + ...$$

Each reflection is added to the total voltage through the unit step function $H(t)$. The above equation can be rewritten as follows:

$$V(x,t) = \left(\frac{Z_O}{Z_O + Z_L}\right) V_s(t - \frac{t_{\text{pd}(2L)}}{v} t) H(t - \frac{t_{\text{pd}(2L)}}{v}) + T_1 \left[ V_s(t - \frac{t_{\text{pd}(2L + x)}}{v} t) H(t - \frac{t_{\text{pd}(2L + x)}}{v}) \right] + T_1 T_2 \left[ V_s(t - \frac{t_{\text{pd}(4L + x)}}{v} t) H(t - \frac{t_{\text{pd}(4L + x)}}{v}) \right] + T_1 T_2 T_3 \left[ V_s(t - \frac{t_{\text{pd}(6L + x)}}{v} t) H(t - \frac{t_{\text{pd}(6L + x)}}{v}) \right] + ...$$

**HOW TO MINIMIZE**

There are several techniques which can be employed to further minimize the effects caused by an impedance mismatch during the layout process:

1. **Impedance Matching**
2. **Daisy Chaining**
3. **Avoid 90° Corners**
4. **Minimize the Number of Vias**

**IMPEDANCE MATCHING**

Impedance matching is the process of matching the impedance of the source or load to the impedance of the trace. This matching is accomplished using a technique called termination. Termination makes the effective source or load impedance, seen by the trace, to be approximately equal to the characteristic impedance of the trace. Before terminating a line one must determine if termination is required. This is done by a simple calculation. If the propagation delay down a trace from source to destination is greater than or equal to one-third the signals rise time, termination is needed. (i.e. $T_{\text{pd}} \geq \frac{1}{3} t_r$). The rise time is the 0%-100% rise time specified for the source. If this value is specified for 10%-90% or 20%-80%, it must be scaled by multiplying the specified value by 1.25 or 1.67, respectively. The propagation delay is calculated by multiplying the trace propagation delay, $t_{\text{pd}}$, described earlier by the trace length.

Once it is determined that termination is needed, use the equation described earlier to calculate the trace’s characteristic impedance. The specification sheets for the load can be consulted to determine the load impedance, $Z_L$. These values are needed to select the component values used to terminate.

The next chore is selecting the type of termination to use. In this section we will examine 4 different techniques and point out the advantages and disadvantages. Figure 2.10 shows the four types of termination and the corresponding component values.

Parallel termination, shown in Figure 2.10(a), is a good technique to maintain the waveform. The waveform at the load is a perfect image of the waveform at the source. In addition there is no added propagation delay associated with this technique. The disadvantage of this technique is that it requires a fair amount of additional power and it is not suggested for characteristic impedances of less than 100 ohms because of the large d.c. current required.

Thevenin termination, shown in Figure 2.10(b), is another option. This technique also requires a large amount of power, but does not have the restrictions for characteristic impedance. This technique is very good at removing overshoot and undershoot while not adding any additional delay. Another advantage is that the trace can be biased toward Vcc or GND by selecting the appropriate resistor values. This can help maintain fast edges on important signal transitions.
<table>
<thead>
<tr>
<th>Name</th>
<th>Circuitry</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
</table>
| Parallel | ![Parallel Circuit](#) | Waveform at receiver is almost perfect image of input  
Bipolar/Advanced CMOS  
No added TPD | High power dissipation  
ZO $> 100\Omega$, else D.C. current limit |
| Thevenin | ![Thevenin Circuit](#) | Good overshoot and undershoot suppression  
Bipolar or Bipolar/CMOS systems  
No added TPD | High power dissipation |

Figure 2-10(a). Termination Techniques

<table>
<thead>
<tr>
<th>Name</th>
<th>Circuitry</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
</table>
| Series | ![Series Circuit](#) | Low power consumption  
CMOS—CMOS Systems  
Easy to adjust signal amplitude to match switching threshold | Added TPD |
| A.C.   | ![A.C. Circuit](#) | Low—medium power dissipation (capacitor blocks D.C. coupling of signal)  
No added delays  
High-speed CMOS families | Two added components |

Figure 2-10(b). Termination Techniques
Series termination, shown in Figure 2-10(b), is a very easy technique of matching impedance. It only requires a resistor and very little additional power is required. In addition the resistor value can be selected to provide constructive or destructive reflections and thus alter the signal amplitude to match the switching threshold. The major disadvantage of this technique is the added delay it introduces.

The fourth technique is A.C. termination, shown in Figure 2-10(b). It requires a small amount of additional power, this is decreased over parallel termination by the introduction of the capacitor, and adds no extra delay to the path. The major disadvantage is that it requires two extra components.

After examining the systems needs and selecting a termination technique, the impedance values determined earlier, $Z_o$ and $Z_L$, can be used to determine the component values to implement the termination. These values should be seen as a starting point and may be altered to remove a specific problem experienced on a signal or to bias signals in an appropriate fashion.

**DAISY CHAINING**

Another technique of minimizing reflections is to daisy-chain signals, shown in Figure 2-11. This means to run a single trace from a source and to distribute the loads along this trace. The alternative is to run multiple traces from the source to each load. Each trace will have reflections of its own and these will be transmitted down the other traces once they have returned to the source. To manage such a system separate termination would be required for each branch. To eliminate these multiple terminators from T-connections, high frequency designs are routed as daisy chains.

Because each gate provides its own impedance load along the chain, it is necessary to distribute these loads evenly along the length of the chain. Hence, the impedance along the chain will change in a series of steps and is easier to match. The overall speed of this line is faster and predictable. Also all loads should be placed at equal distances (regular intervals).

**90 DEGREE ANGLES**

Eliminating 90° angles also minimizes reflections. It is much more desirable to use 45° or 135° angles as shown in Figure 2-12.
VIAS (FEED THROUGH CONNECTIONS)

Another impedance source that degrades high frequency circuit performance is the via. Expert layout techniques can reduce vias to avoid reflection sites on PCBs.

Following these guidelines will not guarantee elimination of all reflections, but they will minimize the number and size.

2.4 Cross Talk

Cross talk is another negative effect of transmission lines. It is a problem at high frequencies because, as operating frequency increase, the signal wavelength become comparable to the length of the interconnections on the PC board. In general, interference such as cross talk, occurs when electrical activity in one conductor causes a transient voltage to appear in another conductor. Main factors that increase interference in any circuit are:

1. Variation of current and voltage in the lines causes frequency interference. This interference increases with increase in frequency.
2. Coupling occurs when conductors are in close proximity.

Cross talk is the phenomenon of a signal in one trace producing a similar signal in an adjacent trace. It may not be a carbon copy of the original signal. It may only be occasional noise that corrupts the integrity of the second signal. The easiest way to minimize crosstalk is to eliminate or at least minimize the number of parallel traces. Parallel traces can be on a single layer or on adjacent signal layers.

There are three ways that parallel traces can couple and thereby produce a signal or at least influence the signal on a second trace. These methods of coupling are inductive, radiative, and capacitive. Inductive coupling is where the two traces act as inductors. The field produced by a signal in one trace induces a current in the second trace. Radiative coupling occurs when the two parallel traces act as a dipole, an antenna. One radiates a signal and the other receives it, thus corrupting the signal already present on the trace. The final method is capacitive coupling. Two parallel traces separated by a dielectric act as a capacitor. If both traces are in a high state and one transitions to a low. The capacitor will try to maintain the high and thus cause a slow transition time on the second trace. These effects can be minimized by reducing the number of parallel traces.

HOW TO MINIMIZE

When laying out a board for a high speed 386 DX based system, several guidelines should be followed to minimize crosstalk. Some of them are as follows:

1. To reduce crosstalk, it is necessary to minimize the common impedance paths.
2. Run a ground line between two adjacent lines. The lines should be grounded at both ends.
3. Separate the address and data busses by a ground line. This technique may however be expensive due to large number of address and data lines.
4. Remove closed loop signal paths which create inductive noise.
5. Capacitive coupling can be reduced by reducing the number of parallel traces. Parallel traces can be minimized by insuring that signals on adjacent signal layers run orthogonal, perpendicular. Ground planes or traces can be inserted to provide shielding. A ground plane between signal layers eliminates any coupling that could occur. On a single trace, a ground trace can be run between traces to prevent coupling.

In some instances it is necessary to run traces parallel to each other. In these cases try to make the distance as short as possible and choose signals in which the transition time is not as critical so that the coupling effects do not produce problems. In addition the coupling can be minimized by increasing the spacing between parallel traces.

2.5 Skew

Skew is another effect of transmission lines. This is very important in a synchronous system. Long traces add propagation delay. A longer trace or a load placed further down a trace will experience more delay than a short trace or loads very close to the source. This must be taken into account when doing the worst case timing analysis. In a system where events must occur synchronous to a clock signal, it is important to make sure the signal is available to all input a sufficient amount of time prior to the corresponding clock edge. When performing the component placement this is one of the considerations that must be accounted for.

These guidelines have always been recommended for board design; however, they are much more important at higher frequencies. At the slower frequencies designers could ignore these practices occasionally and not experience difficulties. This is not the case at higher frequencies.
2.6 DC Loading

To maintain proper logic levels, all digital signal outputs have a maximum load, they are capable of driving. DC loading is the constant current required by an input in either the high or the low state. It limits the ability of a device driving the bus to maintain proper logic levels. For a 386 DX based system, a careful analysis must be performed to ensure that in a worst case situation no loading limits are exceeded. Even if a bus is loaded slightly beyond its worst case limit, it might cause problems if a batch of parts whose input loading is close to maximum is encountered. Proper logic level will then fail to be maintained and unreliable operation may result. Marginal loading problems are particularly insidious, since the effect is often erratic operation and non-repetitive errors that are extremely difficult to track down. For both the high and low logic levels, the sum of the currents required by all the inputs and the leakage currents of all outputs (drivers) on the bus must be added together. This sum must be less than the output capability of the weakest driver. Since the 386 DX is a CMOS device having negligible dc loading, the main contributors to dc loading will be the TTL devices.

2.7 AC Loading

The AC or capacitive loading is caused by the input capacitance of each device and limits the speed at which a device driving a bus signal can change the state from high to low or low to high. Designers of microprocessor systems have traditionally calculated load capacitance of their systems by determining the number of devices and their individual capacitance loading attached to a signal plus the amount of trace capacitance. Typically, the trace capacitance was a set “lumped” number of pf (i.e. 2 pf to 3 pf per inch) when it is thought of at all. This lumped method is a general rule-of-thumb which generates a good first pass approximation. For low frequency designs, the lumped method works since system and component margins are large enough to cover any minor differences due to the approximation.

For high frequency designs, the component and system margins are no longer available to the designer. With less than 1 ns of margin, even the amount of trace capacitance can make a circuit path critical.

A more accurate calculation of capacitive loading can be derived by modeling the device loads and system traces as a series of Transmission Lines Theory. Transmission Line Theory provides a more accurate picture of system loading in high frequency systems. In addition, it allows new factors such as inductance and the effect of reflections upon the quality of the signal waveform to be factored into consideration.

2.8 Derating Curve and Its Effects:

A derating curve is a graph that plots the output buffer against the capacitive load. The curve is used to analyze a signal delay without necessitating a simulation every time the processor’s loading changes. This graph assumes the lumped capacitance model to calculate the total capacitance. The delay in the graph should be added to the specified AC timing value for the device that is driving the load. The derating curve is different for different devices because each device has different output buffers.

A derating curve is generated by tying the chip’s output buffers to a range of capacitors. The voltage and resistance values chosen for the output buffers are at the highest specified temperature and are rising (worst case) values. The value of the capacitors centres around the AC timing values for the chip. For 33 MHz and above, this is 50 pF. Since the AC timing specifications are measured for a signal reaching 1.5 V. A curve is then drawn from the range of time and capacitance values, with 50 pF representing the average and with nominal or zero derating. These curves are valid only for 50 pF–150 pF load range. Beyond this range the output buffers are not characterized. The the derating curve for the 386 DX are shown in 2-13. These curves use the lumped capacitance model for circuit capacitance measurements and must be modified slightly when doing worst-case calculations that involve transmission line effects.

![Figure 2-13. Typical Output Valid Delay Versus Load Capacitance at Maximum Operating Temperature (C_L = 120 pF)](image)
2.9 High Speed Clock Circuits

For performance at high frequencies, the clock signal (CLK2) for the 386 DX CPU must be free of noise and within the specifications listed in the 386 DX CPU data sheet. Achieving the proper clock routing around a 33 MHz printed circuit board is delicate because a myriad of problems, some of them subtle, can arise if guidelines are not followed. For example, fast clock edges cause reflections from high impedance terminations. These reflections can cause significant signal degradation in systems operating at 33 MHz clock rates. This section covers some design guidelines which should be observed to properly lay out the clock lines for efficient 386 DX operation.

• Since the rise/fall time of the clock signal is typically in the range of 2-4 ns, the reflections at this speed could result in undesirable noise and unacceptable signal degradation. The degree of reflections depends on the impedance of the traces of the clock connections. These reflections can be optimized by terminating the CLK2 output with proper terminations and by keeping length of the traces as short as possible. The preferred method is to connect all of the loads via a single trace as shown in Figure 2-14, thus avoiding the extra stubs associated with each load. The loads should be as close to one another as possible. Multiple clock sources should be for distributed loads.
• A less desirable method is the star connection layout in which the clock traces branch to the load as closely as possible (Figure 2-15). In this layout, the stubs should be kept as short as possible. The maximum allowable length of the traces depends upon the frequency and the total fanout, but the length of all the traces in the star connection should be equal. Lengths of less than one inch are recommended. In this method the CLK2 signal is terminated by a series resistor. The resistor value is calculated by measuring the total capacitive load on the CLK2 signal and referring to Figure 2-16. If the total capacitive load is less than 80 pF, the user should add capacitors to make up the difference. Because of the high frequency of CLK2, the terminating resistor must have low inductance; carbon resistors are recommended.
• Use an oscilloscope to compare the CLK2 waveform with those in Figure 2-17.
**Figure 2-16. CLK2 Series Termination**

\[ C_L = C_{IN}^{(386)} + C_{IN}^{(387)} + C_{IN}^{(PALs)} + \ldots + C_{BOARD}. \]

- \( C_{BOARD} \) is calculated from layout and board parameters; thickness, dielectric constant, distance to ground/VCC planes.
- Termination resistor must be low inductance type. Recommend carbon filled type.

**Figure 2-17. CLK2 Waveforms**
SECTION III. DESIGN EXAMPLE

At higher processor speeds the window of time available to perform specific tasks become very small. This window can be equated to multiples of the CLK2 period. Within this time signals must be supplied from a source and reach a destination in time to meet any setup requirements. At 16 MHz the CLK2 period is 31 ns. At 33 MHz it shrinks to half this value, 15 ns. The longer time allowed the use of slower logic families and the delays associated with longer traces. As the window decreases system designers have to practice more care in the selection of logic families and in the choices made for component placement and signal routing on PCBs. This section attempts to list the signal paths whose worst case timing analysis results in very small margins and therefore require closer attention from designers to guarantee that all a.c. timing specifications are met.

This section also includes a sample design based on 33 MHz version of the 386 DX. It should not be taken as a recommended design. The circuit is used only to highlight the design considerations for high speed systems.

3.1 System Architecture

Figure 3.1 shows the system block diagram. It has four major subsystems.
1) CPU subsystem
2) DRAM subsystem
3) Cache subsystem
4) ROM and I/O subsystem

The system has 1 megabytes of Page-Mode DRAMS (60 ns RAS access time), 128 kilobytes of EPROMS (200 ns access time), an 8259A-2, and an 82510. The cache subsystem is optional. Schematics and PAL codes are given in appendix A and B respectively.

3.2 CPU Subsystem

The CPU subsystem consists of the 386 DX microprocessor, a clock and reset circuitry, and bus control logic. Clean and proper clock is very important in the designs at high frequencies.

RESET STATE MACHINE

This state machine is used to generate three control signals, namely RESET, REFREQ and CLK. The CLK signal is half of the CPU clock, CLK2 and is used mainly in I/O and EPROM subsystem.

RESET is generated through the input from RESET triggering circuitry (as shown in the CPU schematic). The min RESET Setup and Hold time for operation at 33 MHz are 5 ns and 2 ns respectively.

A 61.44 KHz clock is used to produce a synchronous refresh request (REFREQ) signal for the DRAM controller, which employ a transparent, distributed, DRAM refresh technique that allows the processor and cache to run while the refresh cycle is in progress.

3.3 DRAM Subsystem

An non-interleaved DRAM system is used in the sample board, which simplifies the design. Since the board provide caching, the performance of DRAM subsystem is outweighed by the simplicity and economy of the design. It employs a transparent, distributed, DRAM refresh technique which allows the processor and cache to run while the refresh cycle is in progress. It uses the 3-state capability of the 16R8-7 and the 74ACT258 to multiplex the refresh address. A further consideration is the choice of DRAM devices. If one uses a memory device such as the AAA2801 (which supports a CAS# before RAS# refresh and provides an internal refresh counter) further simplifications can be made in both the circuitry and the control logic.

DRAM CONTROL STATE MACHINE

The state machine is implemented with three 16R8-type E-speed PALs (see page 4 of the schematics). E-speed PALs must be used since the CLK2 frequency, 66.67 MHz, is higher than the maximum clock frequency of the D-speed PALs.

In order to generate DRAM control signals with smallest delay from the CLK2 edges, all state machines are implemented as Moore machines. The state machines flip-flops generate most of the DRAM control signals directly. This is an expensive design approach in terms of hardware but allows signal timings and skews to be fine tuned.

DRAM CYCLES—NO CACHE CONFIGURATION

Pages C-1 through C-4 show examples of DRAM cycles. In order to hide the DRAM page hit-or-miss decision time, the DRAM controller always tries to put the 386 DX in pipelined mode. The first read cycle requires only two wait states since RAS# has been precharged (see page C-1). The second cycle takes only two clock cycles. The second cycle is a pipelined, page-hit read cycle, which is the best case. The third cycle is a pipelined, page-hit read cycle. This cycle requires one wait state. DRAMs capture data at the falling edge of CAS# during Early Write cycles. The 386 DX drives
Figure 3-1. Block Diagram
valid write data at the rising edge in the middle of Tip (edge C) with a max prop delay of 24 ns (T12 max). This means that the CAS# is generated after the rising edge in the middle of the second T2p (edge A). CAS# is, therefore, generated at the end of RAS# hold time with respect to CAS# (if the next cycle is a page miss, RAS# will go inactive at the end of the current write cycle), and so on.

The fifth cycle is a page miss, which is actually detected at the end of the fourth cycle (page C-2). Since the DRAM controller must wait for minimum RAS# precharge time, the fifth cycle requires three wait states. The sixth cycle is also a page miss. This cycle, however, requires only two wait states because the miss was detected early enough in the previous cycle to have RAS# precharged by the end of the T1p. If the seventh cycle is another page miss, it will require three wait states.

The eighth cycle is ended with T2i. Consequently, the ninth cycle must wait for minimum RAS# precharge time and requires three wait states.

A DRAM refresh cycle is shown on page C-4. The DRAM address multiplexer output is disabled, and the refresh address counter output is enabled. The cycle does a RAS# only refresh cycle where only RAS# is asserted with a proper refresh address. After the refresh cycle is completed, a read cycle which has been suspended due to the refresh is resumed.

**STATE DIAGRAMS**

Pages B-1 through B-11 show state diagrams of the DRAM controller. The precharge state machine on page B-2 measures the required RAS# precharge time and CAS#-to-RAS# precharge time. The CAS#-READY# state machine on page B-2 implements a pin strap option of having or not having the 82385. For no cache configuration, the Cache variable must be forced low.

**TIMING CALCULATIONS**

Timing equations are described on pages D-1 and D-2. Their corresponding results are given on pages D-3 through D-7.

Capacitive load on the 386 DX address bus was assumed to be less than 85 pF. Capacitive load on the DRAM address bus was calculated to be less than 22 pF.

### 3.4 CACHE Subsystem

At 33 MHz DRAM speeds are not fast enough to design zero wait state memory systems. A cache can be used to take advantage of the higher performance available from the higher speed 386 DX microprocessors. The cache takes advantage of the faster SRAM while keeping system costs down by using the cheaper but slower DRAMs.

Details of the cache subsystem are shown on Figure 3.2 and 3.3. The 82385 address and data busses are interfaced to the 386 DX address and data busses via 74AS574s and 74AS646s. Static RAMs (20 ns access time) are used for the cache memory.
Figure 3-2. Block Diagram of Cache Subsystem
Figure 3-3. Address Valid Delay for Cache Subsystem
In selecting SRAM there are several types one can choose to use. Some SRAM require a latch for the address and a transceiver for the data. Others have an OE#, output enable, signal and incorporate the transceiver on chip. The third type is called integrated SRAM and these contain both the latch and the transceiver on chip. However, there are two timing paths that dictate the speed selection within each type. Figure 3.4 shows a typical system configuration using each type.

![Figure 3.4(a) SRAM w/o OE#](image1)

![Figure 3.4(b) SRAM with OE# Control](image2)

![Figure 3-4. (c) Integrated SRAM](image3)

The critical times for the SRAM are the SRAM OE# to data delay and the SRAM address to data delay. The following analysis applies to SRAMs with an OE# signal as shown in Figure 3.4b. First examine the path of OE# to data. This path must be completed within 2 CLK periods. The COE# signal from the 385 Cache Controller must be valid and the SRAM must drive data onto the data bus so that the data setup time of the 386 DX CPU is met.

\[
2 \times \text{CLK2 period} - \text{t25b 82385 COE# valid delay (max)} - \text{SRAM access time (OE# to data)} - \text{t2} \times 386 \text{ DX data setup} \geq 0
\]

Using the specified values from the data sheets reveals that the SRAM must have an OE# to data delay of 10ns or less. The other path is for the address to become available and data to reach the 386 DX CPU. This path has 4 CLK2 periods. The 385 Cache Controller must supply the CALEN signal to pass the address to the SRAM and then the SRAM must drive the data on the data bus so that the data setup time is met on the 386 DX CPU.

\[
4 \times \text{CLK2 period} - \text{t21b 82385 CALEN valid delay (max)} - \text{tpd (x373 latch)} - \text{SRAM access time (address to data)} - \text{t2} \times 386 \text{ DX data setup} \geq 0
\]

Once again using the data sheet the access time can be determined. Depending on the type of transparent latch the SRAM needs an address to data access time of 20ns or 25ns. If an F series 373 is used the faster 20ns SRAM must be used, but if an FCT373a or PCT373a is used the 25ns SRAM is sufficient.

The A20 path is another path with a small margin. The reason is the AND gate that many designers insert to provide 1MB wraparound of address in real mode. Figure 3.5 shows the circuit block diagram. A20 must leave the 386 DX and reach the 385 Cache Controller within 2 CLK2 periods.
To meet this timing the propagation delay of the AND gate must be less than 6ns. This dictates the use of a 74AS08 gate or faster device.

Analysis of the LOCK # path also shows a small margin. The reason is the OR gate that many designers insert to disable the LOCK # signal to the 385 Cache Controller. This allows locked accesses to be cached. Figure 3.6 shows the circuit block diagram. LOCK # must leave the 386 DX and reach the 385 Cache Controller within 2 CLK2 periods.
To meet this timing the propagation delay of the OR gate must be less than 6ns. This dictates the use of a 74AS32 gate or faster device.

The final path examined here is the NA path. Recently designers have selected to use an I/O port and an OR gate to disable pipelining selectively. Figure 3.7 shows the circuit block diagram. NA must leave the 386 DX and reach the 385 Cache Controller within 2 CLK2 periods.

Using the specified values in the appropriate data sheets results in the need for the propagation delay of the OR gate must be no greater than 5.8ns. This dictates the use of a 74AS32 gate or faster device.

This list is not meant to be exhaustive. It is merely meant to highlight a few of the critical timings. Each designer should perform a thorough timing analysis of the system they are designing to verify that all timing requirements are met.

In addition to the specified timing parameters in the data sheets, designers should account for propagation delays introduced by the trace and by capacitive loading. The propagation delay added by the trace is explained in the section on transmission line effects and supplies an equation to determine the amount of delay.

![Figure 3-7. Critical Timing NA](image)
Another factor that becomes more important at higher frequencies is loading. DC loading and especially capacitive loading must be considered during the design stage. If the board is to be assembled and tested in stages, then the DC loads should be considered for all configurations of the board. Most termination techniques require additional current. If a board has a marginal loading situation, one is limited in one’s choices of termination techniques. If a capacitive loading problem exists, the timing situations can become extremely difficult at higher frequencies. If timing is critical, do not overload the capacitance at which a device was tested. If a device is overloaded, derating must be taken into consideration.

Capacitive loading also introduces a delay on signals. Many components including the 386 DX include a capacitive derating curve in the data sheet. To use the curve in the 386 DX data sheet, the capacitive load must be calculated. This is done by summing the input capacitances of all devices driven by a given output from the 386 Microprocessor. Find this value on the X-axis of the derating curve in the data sheet and move up till the derating curve is intersected. Then move at a right angle to the left until intersecting the Y-axis. A value of nom+ or nom− something is found. This is the nominal value plus or minus some amount. The nominal value is the value found in the data sheet. Add the offset from the curve to this nominal value to get the resulting delay corresponding to the capacitive loading in the system. Note: The trace capacitance was not included in this calculation. It is accounted for in the trace propagation delay mentioned earlier.

**DRAM CYCLES WITH 82385 ENABLED**

When the 82385 is enabled (the CACHE variable of the state machine on page B-2 is forced High), the DRAM controller inserts one extra wait state in all read cycles. This extra time is needed to allow a cache update cycle to occur after each cache read miss cycle. During a cache update cycle, the read data from DRAMs must propagate through the 74AS646 and the 74F245 (optional) and must be ready for a SRAM write cycle with enough setup time.
Figure 3-8. Block Diagram of I/O, EPROM Subsystem
NOTE:
Create BS16 for 386 Using DEN and EPROM (Synch to CLK)

Figure 3-9. Control Logic for I/O, EPROM Subsystem
Figure 3-10. ADS# Should Be Synchronized to Guarantee Recognition
APPENDIX B
STATE DIAGRAMS AND PALCODES

MEMCS# = M/IO# • ADS# • [A31-0E1FFFFFFF..00000000]
LMEMCS# = MEMCS# + mreq
module \texttt{RESET\_GEN} flag `-r3`

\texttt{reset_pal} device `\texttt{PIER8}`;

\texttt{x = .I.}; \quad \texttt{"ABEL don't care symbol"}
\texttt{c = .C.}; \quad \texttt{"ABEL clocking input symbol"}

\textbf{\texttt{Inputs}}

\texttt{CLK2 pin 1: \texttt{"CLK2}}
\texttt{RESET pin 2: \"Signal from reset circuitry\}
\texttt{clk\_61 pin 9: \"61.44kHz clock\}

\textbf{\texttt{Outputs}}

\texttt{REFREQ pin 12: \texttt{"REFREQ, sync 61.44kHz clock}}
\texttt{REFTMP pin 13: \"Temporary stage in sync of 61.44kHz clk\}
\texttt{CLK- pin 16: \texttt{"CLK-}}
\texttt{CLK pin 17: \texttt{"CLK = CLK2 / 2}}
\texttt{RESET pin 19: \texttt{"RESET}}

\textbf{equations}

\begin{align*}
    \texttt{CLK} & := (\texttt{CLK \# (\texttt{REFTMP \& \texttt{RESET})}}); \\
    \texttt{CLK-} & := \texttt{CLK}; \\
    \texttt{REFTMP} & := \texttt{RESTRIG}; \\
    \texttt{RESET} & := \texttt{RESTMP}; \\
    \texttt{REFREQ} & := \texttt{REFTMP};
\end{align*}

\textbf{test_vectors}

\begin{align*}
    \{\texttt{CLK2, CLK\_61, RESTRIG, CLK, CLK-, RESTMP, REST, RFQTMP, REFREQ}\} & := \\
    \{\texttt{CLK, CLK-, RESTMP, REST, RFQTMP, REFREQ}\} \\
    \begin{array}{cccccccccccc}
        C & C & R & C & R & A & R & R & R & C & A & R & R & R \\
        L & L & E & L & E & E & E & E & E & L & L & E & E & E \\
        K & K & S & K & S & S & O & F & K & K & S & S & O & F \\
        T & T & T & T & T & T & T & T & T & - & T & T & T & T & T \\
        6 & R & M & T & M & E & M & T & M & E \\
        1 & 1 & P & P & Q & P & P & Q & G & G & G & G & G & G & G
    \end{array}
\end{align*}

\begin{align*}
    \{c, x, 1, x, x, 1, x, 1, x, x, 1, x, 1, x, x\} & \rightarrow \{x, x, 1, x, x, 1, x, x, 1, x, 1, x, 1, x, 1\}; \\
    \{c, x, 1, x, x, 1, x, x, 1, x, 1, x, 1, x, 1\} & \rightarrow \{x, x, 1, x, x, 1, x, 1, x, x, 1, x, x, 1, x, x\}; \\
    \{c, x, 0, x, x, 1, x, x, 1, x, x, 1\} & \rightarrow \{x, x, 0, x, x, 1, x, x, 1, x, x, 1\}; \\
    \{c, x, x, 1, x, x, x, x, x, x, 1, x, x, x, x\} & \rightarrow \{1, x, x, x, x, x, x, x, x, x, x, x, x, x, x\}; \\
    \{c, x, x, x, x, 0, 1, x, x, x, x\} & \rightarrow \{1, x, x, x, x, x, x, x, x, x, x, x, x, x, x\}; \\
    \{c, x, 1, x, 1, x, 1, x, x, 1, x, 1, x, 1, x, 1\} & \rightarrow \{x, 1, x, x, x, 1, x, 1, x, x, 1, x, x, 1, x, x\}; \\
    \{c, x, 0, x, 0, x, 0, x, 0, x, 0\} & \rightarrow \{0, x, x, x, x, 0, x, x, x, x, x, x, x, x, x\}; \\
    \{c, x, x, 0, x, x, 0, x, x, x\} & \rightarrow \{1, 0, x, x, x, 0, x, x, x, x, x, x, x, x, x\};
\end{align*}
[c, x, x, 1, x, 1, x, x, x] → [0, 1, x, x, x, x];
[c, x, 0, x, x, x, x, x, x] → [x, x, 0, x, x, x];  * restmp gen
[c, x, x, x, x, x, x, x, x] → [x, x, x, 0, x, x];  * reset gen
[c, x, 1, x, x, x, x, x, x] → [x, x, 1, x, x, x];
[c, x, x, x, x, 1, x, x, x] → [x, x, x, 1, x, x];
[c, 0, x, x, x, x, x, x, x] → [x, x, x, 0, x, x];  * 61.44KHz clk
[c, x, x, x, x, x, x, 0, x] → [x, x, x, x, x, x, 0];
[c, 1, x, x, x, x, x, x, x] → [x, x, x, x, x, 1, x];
[e, x, x, x, x, x, x, 1, x] ← [x, x, x, x, 1, x];

end RESET_GEN;

ABEL(tm) 3.10 - Document Generator 14-Feb-90 09:53 AM
RESET_GENERATION LOGIC - INTEL CORPORATION
Equations for Module RESET_GEN

Device RESET_PAL

- Reduced Equations:
  1CLK := (CLK & !RESET # CLK & !RESTMP);
  1CLK := (!CLK);
  1RESTMP := (!RESTMP);
  1RESET := (!RESTMP);
  1REFQMP := (!CLK_61);
  1REFREQ := (!REFQMP);

PAL Codes: RESET (Continued)
Device RESET_PAL

PAL Codes: RESET (Continued)
module ADDR_DEC flag '-.v3'

title 'ADDRESS_DECODE LOGIC - INTEL CORPORATION'

ADDR_PAL
device 'PI6LE8';

x = .x;  /*NIEL don't care symbol*/

inputs

AD5- pin 1: "M05"  
M10- pin 2: "M10"  
A31 pin 3: "Aaddr bit 31"  
A30 pin 4: "Aaddr bit 30"  
A29 pin 5: "Aaddr bit 29"  
A6 pin 9: "Aaddr bit 6"  

outputs

MEMS- pin 18: "Memory chip select"  
SKCS- pin 15: "Isb000 chip select"  
S10CS- pin 14: "Isb10 chip select"  
ESROM pin 13: "ESROM chip select"  
LMEMS- pin 12: "Latched/unlatched memory chip select"  

equations

LMEMS- = !AD5- & M10- & !A31 & !A30 & !A29;  
LMEMS- = (AD5- & M10- & A31 & A30 & A29) & !req;  
SKCS- = !M10- & !A41;  
S10CS- = !M10- & !B1;  
ESROM- = M10- & A31 & A30 & A29;

test_vectors

[ [AD5-, M10-, A30, A29, A6, req, MEMS-] -> [MEMS-, SKCS-, S10CS-, ESROM-] ]

* A M A A A A M M S S E
  * 0 3 5 6 r e e e e P
  * s 1 0 9 e M E E C C N
  * - O C C M S D
  * - S S S S M
  * - - - - -

[0, 1, 0, 0, 0, x, 0, 0] -> [0, 0, 1, 1, 1];
[0, 0, 0, 0, 0, x, x, 0] -> [0, x, 1, 1, 1];
[1, x, x, x, x, x, x, 0] -> [1, x, x, x, x];
[1, x, x, x, x, 1, 0] -> [1, x, x, x, x]  'LMEMS-
[0, 1, 0, 0, 0, x, x, x] -> [0, x, 1, 1, 1];

and ADDR_DEC;

240725–93

PAL Codes: Address Decoder

B-9
PAL Codes: Address Decoder (Continued)

- Reduced Equations:

!MEMCS- = (IA29 & IA30 & IA31 & IADS- & M_IO-);
!LMEMCS- = (req # IA29 & IA30 & IA31 & IADS- & M_IO-);
!_59CS- = (IA6 & IM_IO-);
!_5IOCS- = (A6 & IM_IO-);
!EPROM- = (A29 & A30 & A31 & M_IO-);
Chip diagram for Module ADDR_DEC

Device ADDR_PAL

PAL Codes: Address Decoder (Continued)
module PAGE_MODE_DRAM_CTRL_I flag `-r3'
title ‘PAGE MODE DRAM CONTROLLER - PAL 1, INTEL CORPORATION'

PASE1 device  ‘P1680':
  x = .X.;  "ABEL 'don't care' symbol
  c = .C.;  "ABEL 'clocking input' symbol

* Inputs
  CLK2 pin 1:  "D0386 CLK2
  CLK pin 2:  "Processor Clock
  MEMCS- pin 3:  "Memory Chip Select
  LMEMCS- pin 4:  "Latched/Unlatched Memory Chip Select
  HIT- pin 5:  "DRAM Page Hit Signal
  CAS- pin 6:  "Column Address Strobe
  DRAMRDY- pin 7:  "DRAM Ready Signal
  refreq pin 8:  "Refresh Request Signal
  RESET pin 9:  "System Reset

* Outputs
  RAS- pin 12:  "Row Address Strobe
  NA- pin 13:  "Next Address Signal
  precharge pin 14:  "RAS Precharge Signal
  a pin 15:  """
  wait- pin 16:  "delays RAS- until refresh address is valid
  CAL pin 17:  "Column Address Latch
  refresh pin 18:  "Refresh Signal (active once refresh is acknowledged.)

unused pin 19:  "

state diagram [RAS-, NA-]

state [1, 1]; if precharge then [1, 1] else
  if (CLK & refresh & wait-) then [0, 1] else
    if (CLK & LMEMCS- & refresh) then [0, 0] else [1, 1];
state [0, 0]; if RESET then [1, 1] else
  if CAS- then [0, 0] else
    if (CLK & MEMCS- & HLT-)
      CLK & LMEMCS- & ODRAWDY- #
    if (CLK & refresh & ODRAWDY--) then [1, 1] else [0, 0];
state [0, 1]; if RESET then [1, 1] else
  if (CLK & refresh) then [1, 1] else [0, 1];
state [1, 0]; goto [1, 1];

state diagram [precharge, a]

state [0, 0]; if (RAS-) then [0, 1] else [0, 0];
state [0, 1]; if (RESET) then [0, 0] else
  if RAS- then [1, 1] else [0, 1];
state [1, 1]; goto [1, 0];
state [1, 0]; if (CAS-) then [0, 0] else [1, 0];

PAL Codes: DRAM 1

240725-94
state_diagram [CAL]

state [1]: if (NA & CAS-) then [0] else [1];
state [0]: if (RESET) then [1] else

state [0, 0]: if (CLK & refreq) then [1, 0] else [0, 0];
state [1, 0]: if (RESET) then [0, 0] else
state [1, 1]: if (RESET) then [0, 1] else [1, 1];
state [0, 1]: if (RESET) then [0, 1] else

if (CLK & refreq) then [0, 0] else [0, 1];

test_vectors

((CLKE, CLKMEMS-, MEMS-, HIT-, CAS-, DRAMRDY-, refreq, RESET) ->
(BAS-, NA-, precharge, CAL, refresh))

" C M L H D r R B p C r"
" L E M I A R e E A P A e"
" K M E T S A f S S S e f f"
" 2 C M E r M r E c r c r"
" S C R e T h e a s a s"
" S O G a s h h"
" Y r h r r g"
" e e e e"

[c, x, x, x, x, x, 1, x, 1] -> [1, 1, x, 1, 0];
[c, x, x, x, x, x, 1, x, 1] -> [1, 1, x, 1, 0];
[c, 1, 1, 1, 1, x, 1, 0] -> [1, 1, x, 1, 0];
[c, 0, 1, 1, 1, 1, 0] -> [1, 1, x, 1, 0]; *Ti, phase 1
[c, 1, 1, 1, 1, 0] -> [1, 1, x, 1, 0]; *Phase 2
[c, 1, 1, 1, 0, 1, 0] -> [1, 1, x, 1, 0]; *Ti, Read, Non-Pipelined
[c, 0, 0, x, x, 1, 0, 0] -> [1, 1, 0, 1, 0];
[c, 1, 0, 0, x, x, 1, 0, 0] -> [1, 1, 0, 1, 0]; *T2
[c, 0, 0, 0, x, x, 1, 0, 0] -> [0, 0, 0, 0, 0];
[c, 0, 0, 0, x, x, 1, 0, 0] -> [0, 0, 0, 0, 0]; *T2P
[c, 0, 0, 0, x, x, 0, 1, 0] -> [0, 0, 0, 0, 0]; *Page Hit
[c, 0, 0, 1, 0, 0, 0, 0, 0] -> [0, 0, 0, 0, 0]; *T2P
[c, 0, 0, 0, 0, 0, 0, 0, 0] -> [0, 0, 0, 0, 0]; *TIP, Read, Pipelined
[c, 1, 0, 0, 0, 0, 0, 0, 0] -> [0, 0, 0, 0, 0];
[c, 1, 1, 0, 0, 0, 0, 0, 0] -> [0, 0, 0, 0, 0]; *T2P
[c, 0, 0, 0, 0, 0, 0, 0, 0] -> [0, 0, 0, 0, 0];
[c, 1, 0, 0, 0, 0, 0, 0, 0] -> [0, 0, 0, 0, 0]; *TIP, Write
[c, 0, 1, 0, 0, 0, 0, 0, 0] -> [0, 0, 0, 0, 0];
[c, 1, 0, 0, 0, 0, 0, 0, 0] -> [0, 0, 0, 0, 0]; *T2P
[c, 0, 0, 0, 0, 0, 0, 0, 0] -> [0, 0, 0, 0, 0];
[c, 1, 0, 0, 0, 0, 0, 0, 0] -> [0, 0, 0, 0, 0]; *T2P
[c, 0, 0, 0, 0, 0, 0, 0, 0] -> [0, 0, 0, 0, 0];
[c, 0, 0, 0, 0, 0, 0, 0, 0] -> [0, 0, 0, 0, 1];

PAL Codes: DRAM 1 (Continued)
PAL Codes: DRAM 1 (Continued)
PAL Codes: DRAM 1 (Continued)
Device PAGE1

P16R8

CLK2 1
CLK 2
MEMCS~ 3
LMEMCS~ 4
H4~ 5
CAS~ 6
DRAMRDY~ 7
refresh 8
RAS~ 9
10
11

PAL Codes: DRAM 1 (Continued)
PAL Codes: DRAM 2

Device PAGE2

- Reduced Equations:

  !CAS- := (CAS- & CLK & DRAMRD- & IRESET & !a & !b
  # (CACHE & DRAMRD- & IRESET & a & b & !wr
  # DRAMRD- & IAS- & IRESET & a & !b & !wr
  # !CAS- & ICLK & IRESET & a & b
  # !CAS- & DRAMRD- & IRESET & a
  # !CAS- & CLK & DRAMRD- & !MUXOE- & !IAS- & a & b);

  !DRAMRD- := (CAS- & CLK & DRAMRD- & IRESET & !a & !b
  # !CAS- & ICLK & !DRAMRD- & IRESET & a & b
  # !CAS- & CLK & DRAMRD- & IRESET & !a & b
  # !CAS- & CLK & DRAMRD- & IRESET & a & !wr
  # !CACHE & !CAS- & CLK & DRAMRD- & !RESET & a);

  !a := (CAS- & ICLK & DRAMRD- & IRESET & !a & !b
  # CAS- & CLK & DRAMRD- & IAS- & IRESET & a & b & !wr);

  !b := (CAS- & ICLK & DRAMRD- & IRESET & !a & !b
  # CAS- & DRAMRD- & IAS- & IRESET & a & !b
  # (CACHE & CAS- & DRAMRD- & IRESET & a & !b
  # CAS- & DRAMRD- & IRESET & a & b & !wr
  # !CAS- & CLK & DRAMRD- & !IDEC- & IAS- & IRESET & a & b & !fresh
  # !CAS- & ICLK & DRAMRD- & IRESET & !a & !b
  # !CACHE & !CAS- & CLK & DRAMRD- & !RESET & a & !b & !wr);

  !MUXOE- := (MUXOE- & IREF-
  # REN- & !r
  # !MUXOE- & IRESET
  # DRAMRD- & !MUXOE- & !IAS-
  # !IDEC- & !MUXOE- & !CAS-
  # !MUXOE- & Irefresh
  # !ICLK & !MUXOE-);

  !REF- := (MUXOE- & IRESET & !r);

  !r := (MUXOE- & IREF- & IRESET & !r
  # CLK & !MUXOE- & !CAS- & IREF- & !RESET);

240725--98
Device PAGE2

P16R8

<table>
<thead>
<tr>
<th>Pin</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CLK2</td>
</tr>
<tr>
<td>2</td>
<td>CLK</td>
</tr>
<tr>
<td>3</td>
<td>RAS~</td>
</tr>
<tr>
<td>4</td>
<td>MEMCS~</td>
</tr>
<tr>
<td>5</td>
<td>HIT~</td>
</tr>
<tr>
<td>6</td>
<td>CACHE</td>
</tr>
<tr>
<td>7</td>
<td>hwr</td>
</tr>
<tr>
<td>8</td>
<td>refresh</td>
</tr>
<tr>
<td>9</td>
<td>RESET</td>
</tr>
<tr>
<td>10</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
</tr>
</tbody>
</table>

PAL Codes: DRAM 2 (Continued)
module PAGE_MODE_DRAM_CTRL_2 flag `r3'
title 'PAGE MODE DRAM CONTROLLER - PAL 2, INTEL CORPORATION'

PAGE2 device "P1688":
x = .X.;  "ABEL 'don't care' symbol"
c = .C.;  "ABEL 'clocking input' symbol"

" Inputs
CLK2 pin 1: *B08B6 CLK2
CLK pin 2: *Processor Clock
RAS- pin 3: *Row Address Strobe
MEMCS- pin 4: *Memory Chip Select
HIT- pin 5: *DRAM Page Hit Signal (unused)
CACHE pin 6: *Hi when 368 is used; otherwise, low
lw pin 7: *Latched Write/Read
refresh pin 8: *Refresh Signal
RESET pin 9: *System Reset

" Outputs
CAS- pin 12: *Column Address Strobe
DRAMRDY- pin 13: *DRAM Ready
a pin 14: *
b pin 15: *
unused pin 16: *
MUX- pin 17: *DRAM Address Multiplexer Output Enable
REF- pin 18: *Enables refresh counter instead of MUX
r pin 19: *
cstate = [CAS-, DRAMRDY-, a, b]; 'Idle
start = [ 1, 1, 0, 0]; 'Idle
wait = [ 1, 0, 0, 1]; 'CAS- Active
active = [ 0, 0, 1, 1]; 'CAS- and DRAMRDY- Active
inactive_1 = [ 1, 1, 1, 0]; 'Page Hit, CAS- and DRAMRDY-
inactive_2 = [ 1, 1, 0, 0]; 'Page Hit, CAS- and DRAMRDY-

illegal_a = [0, 0, 0, 1];
illegal_b = [0, 0, 1, 0];
illegal_c = [0, 1, 0, 0];
illegal_d = [0, 1, 0, 1];
illegal_e = [1, 0, 0, 0];
illegal_f = [1, 0, 0, 1];
illegal_g = [1, 0, 1, 0];
illegal_h = [1, 0, 1, 1];
illegal_i = [1, 1, 0, 0];
illegal_j = [1, 1, 0, 1];

muxstate = [MUX-, REF-, r];
enabled = [ 0, 1, 1]; 'Multiplexer Outputs Enabled

PAL Codes: DRAM 2 (Continued)

240725–99

B-19
PAL Codes: DRAM 2 (Continued)

disabled_2 = [1, 1, 1, 0];  "Multiplexer Outputs Disabled"
disabled_3 = [1, 0, 0, 0];  "Refresh Address Enabled"
disabled_4 = [0, 1, 0, 0];  "Refresh Address Enabled"

state_diagram cstate

state idle:  if (CLK & TRAS- & MUXVIE-) then start else idle;
state start:  if RESE! then idle else
             if (CLK & TEACHE # CLK & lw) then active else
             if CLK then wait else start;
state wait:  if RESET then idle else
             if CLK then active else wait;
state active:  if RESE! then idle else
             if (CLK & MEMS- & HAS- #
              CLK & refresh) then idle else
             if (CLK & MEMS- & TRAS-) then inactive_1
             else active;
state inactive_1:  if RESE! then idle else
             if (CLK & TRAS- # lw) then inactive_2 else
             if (TRAS & lhwr & CACHE) then start else
             if (lw & cach) then wait else
             inactive_1;
state inactive_2:  if RESE! then idle else
             if CLK then active else inactive_2;

state invalid_a:  goto idle;
state invalid_b:  goto idle;
state invalid_c:  goto idle;
state invalid_d:  goto idle;
state invalid_e:  goto idle;
state invalid_f:  goto idle;
state invalid_g:  goto idle;
state invalid_h:  goto idle;
state invalid_i:  goto idle;
state invalid_j:  goto idle;

state_diagram muxstate

state enabled:  if (CLK & refresh & HAS- & MEMS- #
                CLK & refresh & TRAS- & (DORMOD=1)
                then disabled_1 else enabled;
state disabled_1:  if (RESET) then enabled else disabled_2;
state disabled_2:  if (RESET) then enabled else
                if (CLK & TRAS-) then disabled_3 else disabled_2;
state disabled_3:  if (RESET) then enabled else disabled_4;
state disabled_4:  goto enabled;
state invalid_2:  goto enabled;
state invalid_3:  goto enabled;
state invalid_4:  goto enabled;
state invalid_5:  goto enabled;

240725-A0
test_vectors

```
[[CLK3,CLK,MEMCS,\{wr,hit\},DAS,refresh,RESET,CACHE] ->
[SAS,DRAM0T,HXUE,HBT-]]
```

```
* C C M I H R r R C C D M R
* L L E w I a E A A R U E
* K K M r f S F S C S A X F
* Z C - - r E M - M D G -
* S - - S - E - E -
* h - Y - -
```

```
[c, x, x, 0, x, x, x, 1, 0] -> [1, 1, 0, 1] *Cache disabled
[c, x, x, 0, x, x, x, 1, 0] -> [1, 1, 0, 1] *T1
[c, x, x, 0, x, x, x, 1, 0] -> [1, 1, 0, 1] *T1
[c, x, x, 0, x, x, x, 1, 0] -> [1, 1, 0, 1] *T2
[c, x, x, 0, x, x, x, 1, 0] -> [1, 1, 0, 1] *T2
[c, x, x, 0, x, x, x, 1, 0] -> [1, 1, 0, 1] *T2P
[c, x, x, 0, x, x, x, 1, 0] -> [1, 1, 0, 1] *T2P
```

PAL Codes: DRAM 2 (Continued)
| c, 0, 1, 1, 0, 0, 0, 0, 0 | 0, 0, 1, 0, 0, 0, 0, 0, 0 |
| c, 0, 0, 0, 1, 0, 0, 0, 0, 0 | 0, 0, 0, 1, 0, 0, 0, 0, 0 |
| c, 0, 0, 0, 0, 1, 0, 0, 0, 0 | 0, 0, 0, 0, 1, 0, 0, 0, 0 |
| c, 0, 1, 0, 0, 0, 0, 0, 0, 0 | 0, 1, 0, 0, 0, 0, 0, 0, 0 |
| c, 1, 0, 0, 0, 0, 0, 0, 0, 0 | 1, 0, 0, 0, 0, 0, 0, 0, 0 |
| c, 0, 1, 0, 0, 0, 0, 0, 0, 0 | 0, 1, 0, 0, 0, 0, 0, 0, 0 |
| c, 1, 1, 0, 0, 0, 0, 0, 0, 0 | 1, 1, 0, 0, 0, 0, 0, 0, 0 |
| c, 0, 0, 0, 0, 0, 1, 0, 0, 0 | 0, 0, 0, 0, 0, 1, 0, 0, 0 |
| c, 0, 0, 0, 0, 0, 0, 1, 0, 0 | 0, 0, 0, 0, 0, 0, 1, 0, 0 |
| c, 0, 0, 0, 0, 0, 0, 0, 1, 0 | 0, 0, 0, 0, 0, 0, 0, 1, 0 |
| c, 0, 0, 0, 0, 0, 0, 0, 0, 1 | 0, 0, 0, 0, 0, 0, 0, 0, 1 |

**Note:** The table above contains PAL codes for DRAM 2. Each row represents a set of input conditions and the corresponding output logic. The columns are labeled with specific inputs, and the output is indicated by the final column. The table is continued from the previous page.
PAL Codes: DRAM 2 (Continued)

```
[c, x, x, 0, x, x, x, 1] -> [1, 1, 0, 1];
[c, 0, 1, 0, x, 1, 0, 0] -> [1, 1, 0, 1]; *TI
[c, 1, 1, 0, x, 1, 0, 0] -> [1, 1, 0, 1]; *TI, Read
[c, 1, 0, 0, x, 1, 0, 0] -> [1, 1, 0, 1]; *T2
[c, 0, 1, 0, x, 0, 0, 0] -> [0, 1, 0, 1];
[c, 1, 1, 0, x, 0, 0, 0] -> [0, 1, 0, 1]; *T2P
[c, 1, 0, 0, 0, 0, 0, 0] -> [0, 1, 0, 1];
[c, 0, 0, 0, 0, 0, 0, 0] -> [0, 0, 0, 1]; *T2P
[c, 1, 0, 0, 0, 0, 0, 0] -> [0, 1, 0, 1]; *T2P
[c, 1, 0, 0, 0, 0, 0, 0] -> [0, 1, 0, 1]; *T1P, Read
[c, 0, 0, 0, 0, 0, 0, 0] -> [0, 0, 0, 1]; *T2P
[c, 0, 0, 0, 0, 0, 0, 0] -> [0, 0, 0, 1];
[c, 0, 0, 0, 0, 0, 0, 0] -> [0, 0, 0, 1]; *T2P
[c, 0, 0, 0, 0, 0, 0, 0] -> [0, 0, 0, 1];
```

end PAGE_MODE_DRAM_CTRL_2;

240725–A3
module PAGE_MODE_DRAM_CTRL_3 #Flag 'r3'

title 'PAGE MODE DRAM CONTROLLER - PAL 3, INTEL CORPORATION'

PAG3 device 'P14R8';

x = .X;  "ABEL 'don't care' symbol

y = .Y;  "ABEL 'clocking input' symbol

" Inputs

CLK2 pin 1: "80386 CLK2
CLK pin 2: "Processor Clock
ADS pin 3: "Address Strobe
MEMS pin 4: "Memory Chip Select
WR pin 5: "Write/Read
RDY- pin 6: "System Ready
DRMRY pin 7: "DRAM Ready

" Outputs

TZ2 pin 12: "active during T2, T2b, and T3
TIP pin 13: "active during T1
WE pin 14: "DRAM Write Enable
DEN pin 15: "DRAM Data Bus Transceiver Enable
DR pin 16: "DRAM Data Bus Transceiver R/W Direction signal
lwr pin 17: "Latched Write/Read
nreq pin 18: "Latched Memory Chip Select

" state_diagram [TZ2, TIP-]

state [0, 0]: if (CLK & TAPS) then [0, 1] else [1, 0];
state [0, 1]: if (CLK & TAPS) then [0, 1] else [1, 0];
state [1, 0]: if (CLK & TAPS) then [0, 1] else [1, 0];
state [1, 1]: if (CLK & TAPS) then [1, 0] else [0, 1];

state_diagram [WE-]

state [0]: if (CLK & MEMS & WR & T2- #
1wr & T1P) then [0] else [1];
state [1]: if (RESI) then [1] else [0];

state_diagram [DDN-]

state [0]: if (CLK & MEMS- & 1wr & T2- #
nreq & T1P) then [0] else [1];

240725-A4

PAL Codes: DRAM 3
state [0]: if RESET then [1] else if (CLK & READY) then [1] else [0];
state_diagram (DRM)
state [1]: if (CLK & MEMS- & WR & T2L- #

1wr & T1IP-) then [0] else [1];
state [0]: if (RESET) then [1] else if (CLK & ODN- & 1wr) then [1] else [0];
state_diagram [1wr]
state [0]: if (CLK & MEMS- & WR) then [1] else [0];
state [1]: if (RESET) then [0] else if (T1IP & MEMS- #

READY & 2wr) then [0] else [1];
state_diagram [mem]
state [0]: if (CLK & MEMS-) then [1] else [0];
state [1]: if RESET then [0] else if (READY & MEMS-) then [0] else [1];
test_vectors

((CLK,CLK_A0,WR,MEMS-,READY-,RESET) ->

(T2K, TIP-, ODN-, 1wr, WR & OTR, 1we))

= C C A W M R R T T O W D R

= L L D R E E E 1 1 E W E T T E R

= K K S M A S X P N r - R e

= 2 = C O E = -- = q

= 2 = S Y T = =

[0, 0, 0, 0, 0, 0, 0, 0] -> [1, 1, 1, 0, 0, 0, 0, 0]
[0, 1, 1, 1, 1, 1, 1, 1] -> [0, 0, 1, 1, 1, 1, 1, 1]
[0, 1, 1, 1, 1, 1, 1, 1] -> [0, 0, 1, 1, 1, 1, 1, 1] *T1
[0, 1, 1, 1, 1, 1, 1, 1] -> [0, 0, 1, 1, 1, 1, 1, 1] *T1
[0, 1, 1, 1, 1, 1, 1, 1] -> [0, 0, 1, 1, 1, 1, 1, 1] *T2
[0, 1, 1, 1, 1, 1, 1, 1] -> [0, 0, 1, 1, 1, 1, 1, 1] *T2
[0, 1, 1, 1, 1, 1, 1, 1] -> [0, 0, 1, 1, 1, 1, 1, 1] *T2P
[0, 1, 1, 1, 1, 1, 1, 1] -> [0, 0, 1, 1, 1, 1, 1, 1] *T2P
[0, 1, 1, 1, 1, 1, 1, 1] -> [0, 0, 1, 1, 1, 1, 1, 1] *T2P
[0, 1, 1, 1, 1, 1, 1, 1] -> [0, 0, 1, 1, 1, 1, 1, 1] *T2P
[0, 1, 1, 1, 1, 1, 1, 1] -> [0, 0, 1, 1, 1, 1, 1, 1] *T2P
[0, 1, 1, 1, 1, 1, 1, 1] -> [0, 0, 1, 1, 1, 1, 1, 1] *T2P
[0, 1, 1, 1, 1, 1, 1, 1] -> [0, 0, 1, 1, 1, 1, 1, 1] *T2P
[0, 1, 1, 1, 1, 1, 1, 1] -> [0, 0, 1, 1, 1, 1, 1, 1] *T2P
[0, 1, 1, 1, 1, 1, 1, 1] -> [0, 0, 1, 1, 1, 1, 1, 1] *T2P
[0, 1, 1, 1, 1, 1, 1, 1] -> [0, 0, 1, 1, 1, 1, 1, 1] *T2P
PAL Codes: DRAM 3 (Continued)
PAL Codes: DRAM 3 (Continued)

Device PAGE3

- Reduced Equations:

  $1\overline{T}_{XV} := \overline{CLK} \& \overline{RESET} \& \overline{1\text{TIP-}} \& \overline{T_{2X-}}$

  # $\overline{READY-} \& \overline{RESET} \& \overline{1\text{TIP-}} \& \overline{1T_{2X-}}$

  # $\overline{ADS-} \& \overline{CLK} \& \overline{1\text{TIP-}} \& \overline{T_{2X-}}$;

  $1\text{TIP-} := (\overline{CLK} \& \overline{RESET} \& \overline{1\text{TIP-}} \& \overline{T_{2X-}}$

  # $\overline{ADS-} \& \overline{CLK} \& \overline{1\text{READY-}} \& \overline{RESET} \& \overline{1\text{TIP-}} \& \overline{1T_{2X-}}$;

  $1\text{WE} := (\overline{READY-} \& \overline{RESET} \& \overline{1\text{WE}}$

  # $\overline{CLK} \& \overline{RESET} \& \overline{1\text{WE}}$

  # $\overline{1\text{TIP-}} \& \overline{WE-} \& \overline{1\text{wr}}$

  # $\overline{CLK} \& \overline{\text{MEMCS-}} \& \overline{T_{2X-}} \& \overline{WE-} \& \overline{WR}$;

  $1\text{DEN-} := (\overline{DEN-} \& \overline{READY-} \& \overline{RESET}$

  # $\overline{CLK} \& \overline{DEN-} \& \overline{RESET}$

  # $\overline{CLK} \& \overline{DEN-} \& \overline{1\text{TIP-}} \& \overline{\text{req}}$

  # $\overline{DEN-} \& \overline{T_{2X-}} \& \overline{\text{req}}$

  # $\overline{CLK} \& \overline{DEN-} \& \overline{\text{MEMCS-}} \& \overline{T_{2X-}} \& \overline{WR}$;

  $1\text{IDTR} := (\overline{IDTR} \& \overline{RESET} \& \overline{1\text{wr}}$

  # $\overline{IDEN-} \& \overline{IDTR} \& \overline{RESET}$

  # $\overline{CLK} \& \overline{IDTR} \& \overline{RESET}$

  # $\overline{DTA} \& \overline{1\text{TIP-}} \& \overline{1\text{wr}}$

  # $\overline{CLK} \& \overline{DTA} \& \overline{\text{MEMCS-}} \& \overline{T_{2X-}} \& \overline{WR}$;

  $1\text{Iwr} := (\overline{\text{READY-}} \& \overline{1\text{wr}}$

  # $\overline{\text{MEMCS-}} \& \overline{\text{READY-}}$

  # $\overline{RESET} \& \overline{1\text{wr}}$

  # $\overline{WR} \& \overline{1\text{wr}}$

  # $\overline{\text{MEMCS-}} \& \overline{1\text{wr}}$

  # $\overline{CLK} \& \overline{1\text{Iwr}}$;

  $1\text{req} := (\overline{\text{MEMCS-}} \& \overline{\text{READY-}}$

  # $\overline{\text{RESET}} \& \overline{\text{req}}$

  # $\overline{\text{MEMCS-}} \& \overline{\text{req}}$

  # $\overline{CLK} \& \overline{1\text{req}}$;
Device PAGE3

PAL Codes: DRAM 3 (Continued)
module PAGE_MODE_DRAM_CTRL_4 flag '.r3'
title 'PAGE MODE DRAM CONTROLLER - PAL 4, INTEL CORPORATION'

PAGE4 device 'PIER8':

x = .X.;    " ABEL 'don't care' symbol
C = .C.;    " ABEL 'clocking input' symbol

" Inputs
CLOCK pin 1;
00 pin 2;
01 pin 3;
02 pin 4;
03 pin 5;
04 pin 6;
05 pin 7;
06 pin 8;
07 pin 9;
OE pin 11;

" Outputs
A0 pin 12;
A1 pin 13;
A2 pin 14;
A3 pin 15;
A4 pin 16;
A5 pin 17;
A6 pin 18;
A7 pin 19;

addr = [A7..A0];
equations
addr := addr + 1;

end PAGE_MODE_DRAM_CTRL_4:

PAL Codes: DRAM 4
PAL Codes: DRAM 4 (Continued)

- Reduced Equations:

\[ \text{PAGE} \]:

\[
\begin{align*}
\text{A7} & := (\text{AO} \land \text{A1} \land \text{A2} \land \text{A3} \land \text{A4} \land \text{A5} \land \text{A6} \land \text{A7}) \\
& \land (\text{AO} \land \text{A7}) \\
& \land (\text{A2} \land \text{A7}) \\
& \land (\text{A3} \land \text{A7}) \\
& \land (\text{A4} \land \text{A7}) \\
& \land (\text{A5} \land \text{A7}) \\
& \land (\text{A6} \land \text{A7}); \\
\text{A6} & := (\text{AO} \land \text{A1} \land \text{A2} \land \text{A3} \land \text{A4} \land \text{A5} \land \text{A6}) \\
& \land (\text{AO} \land \text{A6}) \\
& \land (\text{A3} \land \text{A6}) \\
& \land (\text{A4} \land \text{A6}) \\
& \land (\text{A5} \land \text{A6}); \\
\text{A5} & := (\text{AO} \land \text{A1} \land \text{A2} \land \text{A3} \land \text{A4} \land \text{A5}) \\
& \land (\text{AO} \land \text{A5}) \\
& \land (\text{A3} \land \text{A5}) \\
& \land (\text{A4} \land \text{A5}); \\
\text{A4} & := (\text{AO} \land \text{A1} \land \text{A2} \land \text{A3} \land \text{A4}) \\
& \land (\text{AO} \land \text{A4}) \\
& \land (\text{A2} \land \text{A4}) \\
& \land (\text{A3} \land \text{A4}); \\
\text{A3} & := (\text{AO} \land \text{A1} \land \text{A2} \land \text{A3}) \\
& \land (\text{AO} \land \text{A3}) \\
& \land (\text{AO} \land \text{A1}) \\
& \land (\text{AO} \land \text{A2}); \\
\text{A2} & := (\text{AO} \land \text{A2} \land \text{AO} \land \text{A2}) \\
& \land (\text{AO} \land \text{A1} \land \text{A2}); \\
\text{A1} & := (\text{AO} \land \text{A2}) \\
\text{AO} & := (\text{AO});
\end{align*}
\]
Device PAGE4

P16R8

CLOCK  1  20
00  2  19  A7
01  3  18  A6
02  4  17  A5
03  5  16  A4
04  6  15  A3
05  7  14  A2
06  8  13  A1
07  9  12  A0
10  11  OE

240725–68

end of module PAGE_MODE_DRAM_CTRL_4

PAL Codes: DRAM 4 (Continued)
module IO_CTRL_1 flag `-r3`
title `I/O BUS CONTROLLER - PAL 1, INTEL CORPORATION`

IO1 device `PAL12`

x  = .X;  " AEL 'don't care' symbol"
c  = .C;  " AEL 'clocking input' symbol"

" Inputs
CLK  pin 1;  "Processor Clock"
RESET pin 2;  "System Reset"
MRDC- pin 3;  "Memory (EPROM) read Command"
IORC- pin 4;  "I/O Read Command"
IOWC- pin 5;  "I/O Write Command"
INTA- pin 6;  "Interrupt Acknowledge"
DEN- pin 7;  "I/O Bus Data Transceiver Enable"
IORC- pin 8;  "I/O EPROM Ready"
L5D1CS- pin 9;  "REDD Chip Select"
GEN- pin 11;  "PAL output Enable"
L5RCS- pin 12;  "RED2 Chip Select"
LEPROM- pin 13;  "EPROM Chip Select"
unused_0 pin 10;  *
unused_1 pin 19;  *

" Outputs
delay pin 14;  *
sz pin 15;  *
s1 pin 16;  *
s0 pin 17;  *

dstate = [delay, sz, sl, s0];
delay  = [1, 1, 1, 1];
sz     = [1, 1, 0, 1];
sl     = [1, 0, 1, 1];
s0      = [1, 0, 1, 1];

state_diagram dstate

state idle:  if (!DEN- & !MRDC- & !DEN- & !IORC- &
               IORD- & !IOWC- & !INTA-) then start
              else idle;
state start:  if (L5D1CS- & !IOWC-) then wait_14 else
              if (L5RCS- & !IORC-) then wait_13 else
              if (L5RCS- & !IO Rw-) then wait_11 else
              if (!LEPROM- # !IOWC- & !IORC- # !INTA-) then wait_10;
state wait_14: goto wait_13;
state wait_13: goto wait_12;
state wait_12: goto wait_11;
state wait_11: goto wait_10;
state wait_10: goto active;
state active:  if (!IORC-) then idle else active;

end IO_CTRL_1;
- Reduced Equations:

```plaintext
!d0 := ((!d1) & !d0 & s0 & s1 & s2) & delay & s0 & !s1 & !s2);
!d1 := delay & s1 & s2
  # !d0 & !d1 & !s0 & !s1 & !s2
  # !d0 & !d1 & !s0 & !s1 & !s2
  # !d0 & !d1 & !s0 & !s1 & !s2

!s0 := delay & !s0 & !s1 & !s2
  # delay & !s0 & !s1 & !s2
  # !d0 & !d1 & !s0 & !s1 & !s2
  # !d0 & !d1 & !s0 & !s1 & !s2
  # !d0 & !d1 & !s0 & !s1 & !s2
  # !d0 & !d1 & !s0 & !s1 & !s2
```

PAL Codes: IO-1 (Continued)
Device IO1

PAL Codes: IO-1 (Continued)
module IO_CTRL_2 flag '-r3'
title '10 BUS CONTROLLER - PAL 2, INTEL CORPORATION'

IO2 device 'PS6R6';
x = .X;  * ABEL 'don't care' symbol
c = .C;  * ABEL 'clocking input' symbol

* Inputs

  CLK    pin 1: *Processor Clock
  RESET  pin 2: *System Reset
  LMI0   pin 3: *Latched M10
  LDC    pin 4: *Latched OC1
  LWR    pin 5: *Latched W10
  LALAE  pin 6: *Latched ALE
  LSACK- pin 7: *82510 Chip Select
  LSACK+ pin 8: *82510 Chip Select
  LEPROM- pin 9: *EPROM Chip Select
  OEIM   pin 11: *PAL Output Enable
  RDY+   pin 12: *1/O-EPROM Ready (n-1)
  RDY510 pin 13: *1/O-EPROM Ready (n-2)

* Outputs

  recovery pin 13: *1/O Recovery Time
  s1 pin 14: *
  s0 pin 15: *
  OREIC+ pin 16: *1/O Read Command
  OREIC- pin 17: *1/O Write Command
  MRDC+ pin 18: *Memory (EPROM) Read Command

rstate = {recovery, s1, s0};
idle = {0, 1, 0};
active = {0, 1, 1};
inactive_0 = {1, 1, 1};
inactive_1 = {1, 0, 1};
inactive_2 = {1, 0, 0};
inactive_3 = {1, 1, 0};
illegal_a = {0, 0, 0};
illegal_b = {0, 0, 1};

state_diagram rstate

state idle: if (lmi0c # lmic-) then active else idle;
state active: if (lmi0c # lmic-) then inactive_0 else active;
state inactive_0: goto inactive_1;
state inactive_1: goto inactive_2;
state inactive_2: goto inactive_3;
state inactive_3: goto idle;
state illegal_a: goto idle;
state illegal_b: goto idle;

240725–B4

state_diagram [lmi0c-]

state [1]: if (lmi0c & lmi0c & lmi0 & lmi0 & (lmi0cs- # lmi0cs-))
then [0] else [1];
state [0]: if (reset) then [1] else
if (lmi0cs- & (not rdys10- # not rdys-)) then [1] else [0];

state_diagram [lmi0c-]

state [1]: if (lmi0c & lmi0c & lmi0c & lmi0cs- # lmi0cs-)
then [0] else [1];
state [0]: if (reset) then [1] else
if (rdys- then [1] else [0]);

state_diagram [rdys-]

state [1]: if (lmi0 & lmi0 & lmi0c & lmi0c) then [0] else [1];
state [0]: if (reset) then [1] else
if (rdys- then [1] else [0]);

end IO_CTRL_2;

"I"

PAL Codes: IO-2
PAL Codes: IO-2 (Continued)

- Reduced Equations:

  recovery := (recovery & !s) # !IORC- & !IOWC- & !recovery # !s0 & s1);  
  !s1 := (recovery & s0); 
  !s0 := (recovery & !s0 # !s1 # !IORC- & !IOWC- & !s0); 
  !IOWC- := (IOWC- & !RESET & ?ry510 & ?ry-  
    # !IOWC- & !L510C- & !RESET & ?ry-  
    # !IOWC- & !L510C- & !LOC & !LMIO & !LWR & !recovery  
    # !IOWC- & !L510C- & !LOC & !LMIO & !LWR & !recovery); 
  !IORC- := (IORC- & !RESET & ?ry-  
    # !IORC- & !L510C- & !LOC & !LMIO & !LWR & !recovery  
    # !IORC- & !L510C- & !LOC & !LMIO & !LWR & !recovery); 
  !MRODC := (MRODC- & !RESET & ?ry-  
    # !LARE & !LEPRM- & !LMIO & !LWR & !MRODC-);
Chip diagram for Module IO_CTRL_2

Device IO2

end of module IO_CTRL_2

PAL Codes: IO-2 (Continued)
module IO_CTRL_3 flag '-r3'
title '80 BUS CONTROLLER - PAL 2, INTEL CORPORATION'

I03 device 'PS966';

x = .X;  " ABEL 'don't care' symbol
y = .L;  " ABEL 'clocking input' symbol

" Inputs

CLK  pin 1;  *Processor Clock
RESET pin 2;  *System Reset
LMIO pin 3;  *Latched M/I0
LDC  pin 4;  *Latched D/C
LWR  pin 5;  *Latched W/#
LALE pin 6;  *Latched ALE
LSI/OCs- pin 7;  *8255 Chip Select
LSI/OCs+ pin 8;  *8283A-Z Chip Select
LEPROM- pin 9;  *EPROM Chip Select
GEN-  pin 11;  *PAL Output Enable
ndy- pin 12;  *I/O EPROM Ready (n-I)
JANDY- pin 19;  *I/O EPROM Ready

" Outputs

INTA- pin 13;  *Interrupt Acknowledge
st0  pin 14;
DEN- pin 15;  *I/O Bus Transceiver Enable
st1  pin 16;
DTN pin 17;  *I/O Bus Transceiver Direction
st2  pin 18;

state_diagram [INTA-, st0]

state [1, 1]; if (!LMIO & !LDC & !LWR & !LALE) then [1, 0] else [1, 1];
state [1, 0]; if RESET then [1, 1] else
if !LALE then [0, 0] else [1, 0];
state [0, 0]; if RESET then [1, 1] else
if !rdy- then [1, 1] else [0, 0];
state [0, 1]; goto [1, 1];

state_diagram [DEN-, st1]

state [1, 1]; if LALE & (!LEPROM- & !LSI/OCs- & !LSI/OCs-) then [1, 0] else
if !INTA- then [0, 0] else [1, 1];
state [1, 0]; if RESET then [1, 1] else
if !LALE then [0, 0] else [1, 0];
state [0, 0]; if RESET then [1, 1] else
if !rdy- then [1, 1] else [0, 0];
state [0, 1]; goto [1, 1];

state_diagram [DTN, st2]

state [1, 1]; if LALE & (!LEPROM- & !LSI/OCs- & !LSI/OCs-) & LWR then [0, 1]
else [1, 1];
state [0, 1]; if RESET then [1, 1] else
if !JANDY- then [0, 0] else [0, 1];
state [0, 0]; goto [1, 1];
state [1, 0]; goto [1, 1];

end IO_CTRL_3;

PAL Codes: IO-3

240725-B7
Device IO3

Reduced Equations:

\[
\text{INTA}_i := \left( \text{INTA}_0 \land \text{RESET} \land \text{rdy} \land \text{st} \land \text{st}_0 \right)
\]

\[
\text{INTA}_0 := \left( \text{RESET} \land \text{rdy} \land \text{st} \land \text{st}_0 \right)
\]

\[
\text{INTA}_1 := \left( \text{DEN} \land \text{RESET} \land \text{rdy} \land \text{st}_1 \right)
\]

\[
\text{INTA}_2 := \left( \text{DEN} \land \text{RESET} \land \text{rdy} \land \text{st}_2 \right)
\]

\[
\text{DEN} := \left( \text{DEN} \land \text{RESET} \land \text{rdy} \land \text{st}_1 \right)
\]

\[
\text{DEN} := \left( \text{DEN} \land \text{RESET} \land \text{rdy} \land \text{st}_2 \right)
\]

\[
\text{DEN} := \left( \text{DEN} \land \text{RESET} \land \text{rdy} \land \text{st}_0 \right)
\]

\[
\text{DEN} := \left( \text{DEN} \land \text{RESET} \land \text{rdy} \land \text{st}_1 \right)
\]

\[
\text{DEN} := \left( \text{DEN} \land \text{RESET} \land \text{rdy} \land \text{st}_2 \right)
\]

\[
\text{DEN} := \left( \text{DEN} \land \text{RESET} \land \text{rdy} \land \text{st}_0 \right)
\]

\[
\text{DEN} := \left( \text{DEN} \land \text{RESET} \land \text{rdy} \land \text{st}_1 \right)
\]

\[
\text{DEN} := \left( \text{DEN} \land \text{RESET} \land \text{rdy} \land \text{st}_2 \right)
\]

PAL Codes: IO-3 (Continued)
Device IO3

PAL Codes: IO-3 (Continued)
module IO_CTRL_4 flag 'r3'
title 'ID BUS CONTROLLER - PAL 2, INTEL CORPORATION'

IO4 device 'PIB386';
x = .X; * ABEL 'don't care' symbol
c = .C; * ABEL 'clocking input' symbol

* Inputs

CLK pin 1; *Processor Clock
RESET pin 2; *System Reset
LMID pin 3; *Latched M/DI
LOC pin 4; *Latched D/C
LWR pin 5; *Latched W/R
LAE pin 6; *Latched AL
delay pin 7; *Delay Signal For Wait State Generation
unused_0 pin 8; *
unused_1 pin 9; *
GEN pin 11; *PAL Output Enable
unused_2 pin 12; *
unused_4 pin 13; *

* Outputs

IORDY pin 13; */D-PROM Ready
rdy_ pin 14; */D-PROM Ready (n-1)
rdb_1 pin 15; */D-PROM Ready (n-2)
nc_0 pin 16; *
nc_1 pin 17; *
nc_2 pin 18; *

rstate = [IORDY, rdy_, rdb_1];
idle = [1, 1, 1];
rdy2 = [1, 1, 0];
rdb1 = [1, 0, 1];
rdb0 = [0, 1, 1];
illegal_a = [1, 0, 0];
illegal_b = [0, 0, 0];
illegal_c = [0, 0, 1];
illegal_d = [0, 1, 0];

state_diagram rstate

state idle:
if (LMID & LIDC & LWR & LALE) then rdy else
state rdy2:
if idelay then rdy2 else idle;
state rdb1:
if RESET then idle else
state rdb0:
goto idle;
state illegal_a:
goto idle;
state illegal_b:
goto idle;
state illegal_c:
goto idle;

240725-C0

PAL Codes: IO-4
state illegal_d: goto idle;
end lut_ctrl_d;
"2"
Device IO4

PAL Codes: IO-4 (Continued)
module LADDR_DEC flag '-S'
title 'LOCAL_DECODE_LOGIC - INTEL CORPORATION'

LADDR_PAL device "\x60\x68\x6e\x68";

x = .I.;  *ABEL don't care symbol
c = .C.;  *ABEL clocking input symbol
h = .I.;  *logic 3
l = 0;   *logic 0

"Inputs"
ADS- pin 1;  *ADS#
M IO- pin 2;  *M/IO#
A31 pin 3;  *Addr bit 31
A30 pin 4;  *Addr bit 30
A29 pin 5;  *Addr bit 29

"Outputs"
X16- pin 18;  *indicates x 16-bit access
LBA- pin 17;  *local bus access
NCA- pin 16;  *non-cache access

equations
\[ \text{X16-} = (\text{ADS-} \& \text{M IO-} \& \text{A31} \& \text{A30} \& \text{A29}) \]
\[ \text{LBA-} = h; \]
\[ \text{NCA-} = h; \]

end LADDR_DEC;

PAL Codes: Local Decoder
Device LADDR_PAL

end of module LADDR_DEC

PAL Codes: Local Decoder (Continued)
module READY flag 'r3'

title 'READY_LOGIC - INTEL CORPORATION'

RDY device 'P166B':

" Inputs

DRAMRDY- pin 1: "DRAM READY"
RDY- pin 2: "ID/EPROM READY"
RDYEN- pin 3: "RDYEN of 82385"
RDY385- pin 4: "READY# of 82385"
RDY37- pin 5: "READY# of 8237"
CACHE pin 6: "High if cache exits; otherwise, low"

" Outputs

READY- pin 12: "READY# for 82386"
BREADY- pin 13: "BREADY# for 82385"

equations


end READY;

240725-CS

PAL Codes: Ready

240725-CS
Device RDY

PAL Codes: Ready (Continued)
APPENDIX C
TIMING EQUATIONS

DRAM Cycle (R/W Hit/Miss)
DRAM Cycle (Page Miss)
Cache Cycle (Continued)
Cache Cycle (Continued)
<table>
<thead>
<tr>
<th>386</th>
<th>385</th>
<th>CLK2</th>
<th>CLK</th>
<th>ADS</th>
<th>A0</th>
<th>W/R</th>
<th>N/A</th>
<th>CALEN</th>
<th>CS</th>
<th>CME</th>
<th>COE</th>
<th>DOA</th>
<th>CT/R</th>
<th>BAO</th>
<th>BACP</th>
<th>BACOE</th>
<th>BAOEI</th>
<th>BW/R</th>
<th>NTG</th>
<th>RAS</th>
<th>CAS</th>
<th>RMA</th>
<th>READV</th>
<th>RDSR7E</th>
<th>DSN</th>
<th>DOA</th>
<th>WE</th>
<th>DT/R</th>
<th>LDS</th>
<th>OEE</th>
<th>BT/R</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>T1</td>
<td>T1</td>
<td>T1</td>
<td>T2</td>
<td>T1</td>
<td>T2</td>
<td>T1P</td>
<td>T2P</td>
<td>T1P</td>
<td>T1P</td>
<td>T1P</td>
<td>T1P</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Cache Cycle (Continued)**

- **Read, Cache Hit**
- **Read, Cache Miss**
- **Write, Cache Miss**

**Timelines:**
- **T1:** DRAM Read
- **T2:** DRAM Write, Page Miss
- **T1P:** DRAM Write, Page Hit

**Signals:**
- CLK: Clock
- ADS: Address
- A0: Address 0
- W/R: Write/Read
- N/A: Not Available
- CALEN: Cycle Address Enable
- CS: Chip Select
- CME: Command Enable
- COE: Command Output Enable
- DOA: Data Output
- CT/R: Command Test/Read
- BAO: Bank Address
- BACP: Bank Address Control
- BACOE: Bank Address Control Enable
- BAOEI: Bank Address Output Enable
- BW/R: Bank Write/Read
- NTG: No Toggling
- RAS: Row Active
- CAS: Column Active
- RMA: Row Mode Select
- READV: Read Enable
- RDSR7E: Row Disable
- DSN: Data Signal
- DOA: Data Output
- WE: Write Enable
- DT/R: Data Test/Read
- LDS: Load Signal
- OEE: Output Enable
- BT/R: Bank Test/Read
Cache Cycle (Continued)
EPROM and I/O Cycles

240725–84

240725–85

C-10
EPROM and I/O Cycles (Continued)
EPROM and I/O Cycles (Continued)
EPROM and I/O Cycles (Continued)
EQUATIONS FOR DRAM TIMINGS (NO CACHE CONFIGURATION):

Read and Write Cycles (Common Parameters):

- **tRC**: Random Read or Write Cycle Time
  \[ \text{CLK2} \times 10 \]

- **tRP**: RAS# Precharge Time
  \[ \text{CLK2} \times 4 \]

- **tRAS**: RAS# Pulse Width
  \[ \text{CLK2} \times 4 \]
  
  A random DRAM cycle may have a RAS# pulse which is only four CLK2 periods wide. This is the case if the cycle is followed by Idle cycles (DRAMs not selected or Ti's) or a DRAM page miss.

- **tCAS (Read)**: CAS# Pulse Width
  \[ \text{CLK2} \times 3 \]
  
  CAS# pulses can be as narrow as three CLK2 cycles during Page Mode read cycles.

- **tCAS (Write)**: CAS# Pulse Width
  \[ \text{CLK2} \times 2 \]
  
  CAS# pulses can be as narrow as two CLK2 cycles during Page Mode write cycles.

- **tASC**: Column Address Setup Time
  \[ \text{min} (\text{CLK2} \times 2 + \text{AS32.tphl.min} - \text{Delay.max} - \text{ACT258.StoZ.tpl.max} - \text{ACT258.Cap.Derating} - \text{AS373.GtoO.tpd.max} - \text{ACT258.toZ.tpl.max} - \text{ACT258.Cap.Derating}) \]

  The Column Address becomes valid as RAS# switches from High to Low or as the 386 address becomes valid while RAS# is already Low (i.e., Page Mode, Pipelined cycles).

- **tCAH**: Column Address Hold Time
  \[ \text{CLK2} + \text{AS373.GtoO.tpd.min} + \text{ACT258.toZ.tpl.min} - \text{AS32.tphl.max} \]

  The CAL (Column Address Latch) signal is activated one CLK2 period after the active-going edge of CAS#.

- **tAR**: Column Address Hold Time to RAS#
  \[ \text{CLK2} \times 3 + \text{AS373.GtoO.tpd.min} + \text{ACT258.toZ.tpl.min} - \text{RAS.Delay.max} \]

- **tRCD**: RAS# to CAS# Delay Time
  \[ \text{CLK2} \times 2 + \text{AS32.tphl.min} - \text{RAS.Delay.max} \]

- **tRAD**: RAS# to Column Address Delay Time
  \[ \text{(min)} \text{ ACT258.StoZ.tpl.min} + \text{Delay.min} - \text{RAS.Delay.max} \]
  \[ \text{(max)} \text{ ACT258.StoZ.tpl.max} + \text{Delay.max} + \text{ACT258.Cap.Derating} - \text{RAS.Delay.min} \]

- **tSH**: RAS# Hold Time
  \[ \text{CLK2} \times 2 - \text{AS32.tphl.max} + \text{RAS.Delay.min} \]

  The worst case occurs when a DRAM Page miss or Idle is detected at the end of the current DRAM Page miss cycle.

- **tCSH**: CAS# Hold Time
  \[ \text{CLK2} \times 2 + \text{AS32.tphl.min} - \text{RAS.Delay.max} \]

- **tCRP**: CAS# to RAS# Precharge Time
  \[ \text{CLK2} \times 2 + \text{RAS.Delay.min} - \text{AS32.tphl.max} \]

  This is guaranteed by the DRAM control state machine.

- **tASR**: Row Address Setup Time
  \[ \text{CLK2} \times 2 - \text{t6.max} - \text{386.Cap.Derating} - \text{ACT258.ItoZ.max} - \text{ACT258.Cap.Derating} - \text{H124.tpd.min} + \text{H125.tpd.min} + \text{PAL.tco.min} + \text{RAS.Delay.min} \]

- **tRAH**: Row Address Hold Time
  \[ \text{ACT258.StoZ.tpl.min} + \text{Delay.min} - \text{RAS.Delay.max} \]

- **tT**: Transition Time (Rise and Fall)

- **tREF**: Refresh Period

- **tREF2**: Refresh Period
Read Cycles:

- **tRAC**: Access Time
  \[
  \text{CLK2} \times 6 - H124.tpd.max - H125.tpd.max - \text{PAL.tco.max} - t21.min - F245.max - \text{RAS.Delay.max}
  \]

- **tCAC**: Access Time from CAS#
  \[
  \text{CLK2} \times 3 - H124.tpd.max - \text{PAL.tco.max} - \text{AS32.tphl.max} - \text{t21.min} - F245.max
  \]

- **tAA**: Access Time from Address
  \[
  \text{CLK2} \times 6 - t6.max - 386\text{Cap.Derating} - \text{AS373.ttoO.max} - \text{ACT258.ttoZ.tp.max} - \text{ACT258.Cap.Derating} - \text{t21.min} - F245.max
  \]

- **tRCS**: Read Command Setup Time
  \[
  \text{CLK2} + \text{AS32.tphl.min}
  \]

- **tRCH**: Read Command Hold Time to CAS#
  \[
  \text{CLK2} - \text{AS32.tphl.max}
  \]

- **tRRH**: Read Command Hold Time to RAS#
  \[
  \text{CLK2} - \text{RAS.Delay.max}
  \]

- **tOFF**: Output Buffer Turn-off Time
  \[
  \text{CLK2} \times 2 + F245.tzh.min
  \]

Write Cycles:

- **tWCS**: Write Command Setup Time
  \[
  \text{CLK2} \times 3 + \text{AS32.tphl.min}
  \]

- **tWCH**: Write Command Hold Time
  \[
  \text{CLK2} \times 2 - \text{AS32.tphl.max}
  \]

- **tWCR**: Write Command Hold Time to RAS#
  \[
  \text{CLK2} \times 6 - \text{RAS.Delay.max}
  \]

- **tWP**: Write Command Pulse Width
  \[
  \text{CLK2} \times 5
  \]

- **tRWL**: Write Command to RAS# Lead Time
  \[
  \text{CLK2} \times 5 + \text{RAS.Delay.min}
  \]

- **tCWL**: Write Command to CAS# Lead Time
  \[
  \text{CLK2} \times 5
  \]

Page Mode Cycles:

- **tPC**: Page Mode Cycle Time
  \[
  \text{CLK2} \times 4
  \]

- **tRACP**: Page Mode RAS# Pulse Width
  \[
  \text{CLK2} \times 4
  \]

- **tRSW**: RAS# to Second WE# Delay Time
  \[
  \text{CLK2} \times 7 + \text{RAS.Delay.max}
  \]

- **tCP**: CAS# Precharge Time
  \[
  \text{CLK2}
  \]

- **tWI**: Write Invalid Time
  \[
  \text{CLK2}
  \]

- **tCAP**: Access Time from Column Precharge Time
  \[
  \text{CLK2} \times 4 - H124.tpd.max - H125.tpd.max - \text{PAL.tco.max} - \text{t21.min} - F245.max
  \]
### 80386 A.C. SPECIFICATIONS

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>t1</td>
<td>CLKS Period</td>
<td>16.00</td>
<td>68.50</td>
</tr>
<tr>
<td>t2a</td>
<td>CLK2 High Time</td>
<td>6.25</td>
<td></td>
</tr>
<tr>
<td>t2b</td>
<td>CLK2 High Time</td>
<td>4.65</td>
<td></td>
</tr>
<tr>
<td>t3a</td>
<td>CLK2 Low Time</td>
<td>6.25</td>
<td></td>
</tr>
<tr>
<td>t3b</td>
<td>CLK2 Low Time</td>
<td>4.50</td>
<td></td>
</tr>
<tr>
<td>t4</td>
<td>CLK2 Fall Time</td>
<td>4.00</td>
<td></td>
</tr>
<tr>
<td>t5</td>
<td>CLK2 Rise Time</td>
<td>4.00</td>
<td></td>
</tr>
<tr>
<td>t6</td>
<td>A2-A51 Valid Delay</td>
<td>4.00</td>
<td>15.00</td>
</tr>
<tr>
<td>t7</td>
<td>A2-A51 Float Delay</td>
<td>4.00</td>
<td>20.00</td>
</tr>
<tr>
<td>t8</td>
<td>B6D-H6D, LOCK$ Valid Delay</td>
<td>4.00</td>
<td>15.00</td>
</tr>
<tr>
<td>t9</td>
<td>B6D-H6D, LOCK$ Float Delay</td>
<td>4.00</td>
<td>20.00</td>
</tr>
<tr>
<td>t10</td>
<td>W#/W, W/104, D/C#, ACS# Valid Delay</td>
<td>4.00</td>
<td>35.00</td>
</tr>
<tr>
<td>t11</td>
<td>W#/W, W/104, D/C#, ACS# Float Delay</td>
<td>4.00</td>
<td>25.00</td>
</tr>
<tr>
<td>t12</td>
<td>DD-D3I Write Data Valid Delay</td>
<td>5.00</td>
<td>24.00</td>
</tr>
<tr>
<td>t13</td>
<td>DD-D3I Float Delay</td>
<td>4.00</td>
<td>17.00</td>
</tr>
<tr>
<td>t14</td>
<td>HIDA Valid Delay</td>
<td>4.00</td>
<td>20.00</td>
</tr>
<tr>
<td>t15</td>
<td>H# Setup Time</td>
<td>5.00</td>
<td></td>
</tr>
<tr>
<td>t16</td>
<td>H# Hold Time</td>
<td>3.00</td>
<td></td>
</tr>
<tr>
<td>t17</td>
<td>HOLD# Setup Time</td>
<td>5.00</td>
<td></td>
</tr>
<tr>
<td>t18</td>
<td>HOLD# Hold Time</td>
<td>3.00</td>
<td></td>
</tr>
<tr>
<td>t19</td>
<td>READY# Setup Time</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>t20</td>
<td>READY# Hold Time</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>t21</td>
<td>DO-DOI Read Setup Time</td>
<td>5.00</td>
<td></td>
</tr>
<tr>
<td>t22</td>
<td>DO-DOI Read Hold Time</td>
<td>3.00</td>
<td></td>
</tr>
<tr>
<td>t23</td>
<td>HOLD Setup Time</td>
<td>11.00</td>
<td></td>
</tr>
<tr>
<td>t24</td>
<td>HOLD Hold Time</td>
<td>3.00</td>
<td></td>
</tr>
<tr>
<td>t25</td>
<td>RES# Setup Time</td>
<td>8.00</td>
<td></td>
</tr>
<tr>
<td>t26</td>
<td>RES# Hold Time</td>
<td>3.00</td>
<td></td>
</tr>
<tr>
<td>t27</td>
<td>SNL, INTR Setup Time</td>
<td>5.00</td>
<td></td>
</tr>
<tr>
<td>t28</td>
<td>SNL, INTR Hold Time</td>
<td>5.00</td>
<td></td>
</tr>
<tr>
<td>t29</td>
<td>LED, ERR# Setup Time</td>
<td>5.00</td>
<td></td>
</tr>
<tr>
<td>t30</td>
<td>LED, ERR#, BSY# Hold Time</td>
<td>4.00</td>
<td></td>
</tr>
</tbody>
</table>

### PLL SPECIFICATIONS

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>tse</td>
<td>Input or Feedback Setup Time</td>
<td>7.00</td>
<td></td>
</tr>
<tr>
<td>tco</td>
<td>Clock to Output</td>
<td>3.00</td>
<td>6.50</td>
</tr>
</tbody>
</table>

### ROW ACCESS LATCH SPECIFICATIONS

74FCT244AH (177)

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>50 pF</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>tp1b</td>
<td>Do to On Propagation Delay</td>
<td>3.00</td>
<td>6.50</td>
<td></td>
</tr>
<tr>
<td>tph1</td>
<td>0 to On Propagation Delay</td>
<td>3.00</td>
<td>6.50</td>
<td></td>
</tr>
<tr>
<td>tp1b</td>
<td>St to On Propagation Delay</td>
<td>6.00</td>
<td>8.00</td>
<td></td>
</tr>
<tr>
<td>tpd1</td>
<td>Setup Time</td>
<td>6.00</td>
<td>8.00</td>
<td></td>
</tr>
<tr>
<td>t1</td>
<td>Hold Time</td>
<td>2.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>t30</td>
<td></td>
<td>3.00</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

Timings for No Cache Configuration

*AP-442 D-3*
### Row Address Comparator Specifications

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>tp1b</td>
<td>An or Bn to Q Propagation Delay</td>
<td>1.50</td>
<td>5.50</td>
</tr>
<tr>
<td>tp2h</td>
<td>G to Hn or H to An Propagation Delay</td>
<td>1.50</td>
<td>5.50</td>
</tr>
<tr>
<td>tp3h</td>
<td>G to Hn or H to An Propagation Delay</td>
<td>1.50</td>
<td>4.60</td>
</tr>
<tr>
<td>tp4h</td>
<td>H to Q Propagation Delay</td>
<td>1.50</td>
<td>4.60</td>
</tr>
</tbody>
</table>

### Drum Address Multiplexer Specifications

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>tp1b</td>
<td>&amp; to In Propagation Delay</td>
<td>1.00</td>
<td>11.50</td>
</tr>
<tr>
<td>tp2h</td>
<td># to In Propagation Delay</td>
<td>1.00</td>
<td>9.50</td>
</tr>
<tr>
<td>tp3h</td>
<td>In to In Propagation Delay</td>
<td>1.00</td>
<td>9.50</td>
</tr>
<tr>
<td>tp4h</td>
<td>In to In Propagation Delay</td>
<td>1.00</td>
<td>6.00</td>
</tr>
</tbody>
</table>

### Data Transceiver Specifications

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>tp1b</td>
<td>An to Bn or Bn to An Propagation Delay</td>
<td>2.50</td>
<td>7.00</td>
</tr>
<tr>
<td>tp2h</td>
<td>G to Hn or H to An Propagation Delay</td>
<td>2.50</td>
<td>7.00</td>
</tr>
<tr>
<td>tsh</td>
<td>Output Enable Time</td>
<td>3.00</td>
<td>8.00</td>
</tr>
<tr>
<td>ts1</td>
<td>Output Enable Time</td>
<td>3.50</td>
<td>9.50</td>
</tr>
<tr>
<td>tpxs</td>
<td>Output Disable Time</td>
<td>3.00</td>
<td>7.50</td>
</tr>
<tr>
<td>tpx</td>
<td>Output Disable Time</td>
<td>2.00</td>
<td>7.50</td>
</tr>
</tbody>
</table>

### Column Address Latch Specifications

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>tp1b</td>
<td>On to Off Propagation Delay</td>
<td>3.00</td>
<td>6.00</td>
</tr>
<tr>
<td>tp2h</td>
<td>Off to On Propagation Delay</td>
<td>3.00</td>
<td>6.00</td>
</tr>
<tr>
<td>tp3h</td>
<td>G to On Propagation Delay</td>
<td>3.00</td>
<td>11.50</td>
</tr>
<tr>
<td>tp4h</td>
<td>On to Off Propagation Delay</td>
<td>4.00</td>
<td>7.50</td>
</tr>
<tr>
<td>tsh</td>
<td>Setup Time</td>
<td>2.00</td>
<td></td>
</tr>
<tr>
<td>ts</td>
<td>Hold Time</td>
<td>3.00</td>
<td></td>
</tr>
</tbody>
</table>

### RAM Delay

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>tp</td>
<td>Propagation Delay</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Timings for No Cache Configuration (Continued)
## DRAM Timing Requirements

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>tpd</td>
<td>Propagation Delay</td>
<td>1.50</td>
<td>1.40</td>
</tr>
<tr>
<td>tpdh</td>
<td>Propagation Delay (High)</td>
<td>1.50</td>
<td>1.40</td>
</tr>
</tbody>
</table>

### Read and Write Cycles (Common Parameters):

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>tRCD</td>
<td>Row Cycle Delay</td>
<td>13.00</td>
<td>14.00</td>
</tr>
<tr>
<td>tRSC</td>
<td>Random Slow Cycle Time</td>
<td>10.00</td>
<td>12.00</td>
</tr>
<tr>
<td>tRP</td>
<td>Random Precharge Time</td>
<td>40.00</td>
<td>50.00</td>
</tr>
<tr>
<td>tRDH</td>
<td>DAS6 Pulse Width</td>
<td>40.00</td>
<td>50.00</td>
</tr>
<tr>
<td>tRAS</td>
<td>CAS Latency (Read)</td>
<td>40.00</td>
<td>50.00</td>
</tr>
<tr>
<td>tRAS</td>
<td>CAS Pulse Width (Read)</td>
<td>30.00</td>
<td>50.00</td>
</tr>
<tr>
<td>tRAS</td>
<td>Column Address Setup Time</td>
<td>9.70</td>
<td>10.00</td>
</tr>
<tr>
<td>tRAS</td>
<td>Column Address Hold Time</td>
<td>14.20</td>
<td>14.60</td>
</tr>
<tr>
<td>tRAS</td>
<td>Column Address Hold Time to RAS#</td>
<td>50.00</td>
<td>70.00</td>
</tr>
<tr>
<td>tRAS</td>
<td>Column Address Setup Time to RAS#</td>
<td>10.00</td>
<td>10.00</td>
</tr>
<tr>
<td>tRAS</td>
<td>RAS to Row Address Delay Time</td>
<td>21.40</td>
<td>21.60</td>
</tr>
<tr>
<td>tRAS</td>
<td>RAS Hold Time</td>
<td>3.80</td>
<td>4.20</td>
</tr>
<tr>
<td>tRAS</td>
<td>CAS Hold Time</td>
<td>9.20</td>
<td>10.00</td>
</tr>
<tr>
<td>tRAS</td>
<td>CAS to RAS Precharge Time</td>
<td>24.20</td>
<td>24.40</td>
</tr>
<tr>
<td>tRAS</td>
<td>Row Address Setup Time</td>
<td>5.45</td>
<td>3.45</td>
</tr>
<tr>
<td>tRAS</td>
<td>Row Address Hold Time</td>
<td>8.00</td>
<td>3.00</td>
</tr>
<tr>
<td>tRAS</td>
<td>Transition Time (High and Fall)</td>
<td>5.00</td>
<td>5.00</td>
</tr>
<tr>
<td>tRAS</td>
<td>Refresh Period</td>
<td>100.00</td>
<td>100.00</td>
</tr>
</tbody>
</table>

### Read Cycles:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>tRAC</td>
<td>Access Time</td>
<td>68.25</td>
<td>68.25</td>
</tr>
<tr>
<td>tRAC</td>
<td>Access Time from CAS#</td>
<td>51.75</td>
<td>51.75</td>
</tr>
<tr>
<td>tA</td>
<td>Access Time from Address</td>
<td>43.75</td>
<td>43.75</td>
</tr>
<tr>
<td>tRAS</td>
<td>Read Command Setup Time</td>
<td>14.00</td>
<td>14.00</td>
</tr>
<tr>
<td>tRAS</td>
<td>Read Command Setup Time - CAS#</td>
<td>9.20</td>
<td>9.20</td>
</tr>
<tr>
<td>tRAS</td>
<td>Read Command Hold Time to RAS#</td>
<td>15.00</td>
<td>15.00</td>
</tr>
<tr>
<td>tRF</td>
<td>Output Buffer Turn-off Time</td>
<td>33.00</td>
<td>33.00</td>
</tr>
</tbody>
</table>

### Write Cycles:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>tWCS</td>
<td>Write Command Setup Time</td>
<td>48.00</td>
<td>48.00</td>
</tr>
<tr>
<td>tWCS</td>
<td>Write Command Hold Time</td>
<td>24.25</td>
<td>24.25</td>
</tr>
<tr>
<td>tWCS</td>
<td>Write Command Hold Time to RAS#</td>
<td>50.00</td>
<td>50.00</td>
</tr>
<tr>
<td>tWF</td>
<td>Write Command Pulse Width</td>
<td>50.00</td>
<td>50.00</td>
</tr>
<tr>
<td>tWC</td>
<td>Write Command to CAS# Lead Time</td>
<td>3.00</td>
<td>3.00</td>
</tr>
<tr>
<td>tWCL</td>
<td>Write Command to CAS# Lead Time</td>
<td>7.00</td>
<td>7.00</td>
</tr>
<tr>
<td>tWCS</td>
<td>Data-in Setup Time</td>
<td>8.75</td>
<td>8.75</td>
</tr>
<tr>
<td>tWS</td>
<td>Data-in Hold Time</td>
<td>28.20</td>
<td>28.20</td>
</tr>
<tr>
<td>tWSR</td>
<td>Data-in Hold Time to RAS#</td>
<td>97.50</td>
<td>97.50</td>
</tr>
</tbody>
</table>

### Page Mode Cycles:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>tPC</td>
<td>Page Mode Cycle Time</td>
<td>23.00</td>
<td>23.00</td>
</tr>
<tr>
<td>tPDC</td>
<td>Page Mode DAS6 Pulse Width</td>
<td>60.00</td>
<td>60.00</td>
</tr>
<tr>
<td>tPDC</td>
<td>Page Mode DAS6 Precharge Time</td>
<td>160.00</td>
<td>160.00</td>
</tr>
<tr>
<td>tPC</td>
<td>CAS Precharge Time</td>
<td>5.00</td>
<td>5.00</td>
</tr>
<tr>
<td>tPC</td>
<td>Write Invalid Time</td>
<td>15.00</td>
<td>15.00</td>
</tr>
<tr>
<td>tPC</td>
<td>Access Time from Column Precharge Time</td>
<td>30.25</td>
<td>4.25</td>
</tr>
</tbody>
</table>

---

Timings for No Cache Configuration (Continued)
### ADDRESS DECODER REQUIREMENTS

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>tpd</td>
<td>Available Propagation Delay</td>
<td>8.75</td>
<td></td>
</tr>
</tbody>
</table>

### ROW ADDRESS COMPARATOR REQUIREMENTS

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>tpd</td>
<td>Available Propagation Delay</td>
<td>8.75</td>
<td></td>
</tr>
</tbody>
</table>

### RAS SETUP TIME

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>tRAS</td>
<td>Available RAS Setup Time</td>
<td>3.25</td>
<td></td>
</tr>
</tbody>
</table>

### QUAD TTL TO 10K ohm ECL TRANSLATOR MC1SN124

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>tpd</td>
<td>Propagation Delay</td>
<td>2.75</td>
<td>3.25</td>
</tr>
</tbody>
</table>

### QUAD 10K ohm ECL to TTL TRANSLATOR MC10H5125

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>tpd</td>
<td>Propagation Delay</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

### DELAY ELEMENT

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>tpd</td>
<td>Propagation Delay</td>
<td>4.00</td>
<td>6.00</td>
</tr>
</tbody>
</table>

Timings for No Cache Configuration (Continued)
### DRAM SPECIFICATIONS

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>RDC</td>
<td>121.00</td>
<td>125.00</td>
</tr>
<tr>
<td>LREF</td>
<td>55.00</td>
<td>56.00</td>
</tr>
<tr>
<td>TRAS</td>
<td>60.00</td>
<td>100000</td>
</tr>
<tr>
<td>TCHAS</td>
<td>21.00</td>
<td>15.00</td>
</tr>
<tr>
<td>TCAS</td>
<td>5.00</td>
<td>25.00</td>
</tr>
<tr>
<td>TACQ</td>
<td>5.00</td>
<td>25.00</td>
</tr>
<tr>
<td>TACM</td>
<td>6.00</td>
<td>15.00</td>
</tr>
<tr>
<td>TCSR</td>
<td>40.00</td>
<td>55.00</td>
</tr>
<tr>
<td>TRCQ</td>
<td>6.00</td>
<td>45.00</td>
</tr>
<tr>
<td>TRCD</td>
<td>4.00</td>
<td>20.00</td>
</tr>
<tr>
<td>TREAD</td>
<td>15.00</td>
<td>25.00</td>
</tr>
<tr>
<td>TWRH</td>
<td>15.00</td>
<td>25.00</td>
</tr>
<tr>
<td>TCSR</td>
<td>40.00</td>
<td>70.00</td>
</tr>
<tr>
<td>TCSR</td>
<td>3.00</td>
<td>15.00</td>
</tr>
<tr>
<td>TCHS</td>
<td>2.00</td>
<td>15.00</td>
</tr>
<tr>
<td>TST</td>
<td>3.00</td>
<td>25.00</td>
</tr>
</tbody>
</table>

### CAPACITIVE LOAD TIMING DERATING FOR 74ACT536

<table>
<thead>
<tr>
<th>Load Capacitance (pF)</th>
<th>Additional Propagation Delay (ns)</th>
</tr>
</thead>
<tbody>
<tr>
<td>60.00</td>
<td>0.26 (p = 0.02625q - 1.3125)</td>
</tr>
<tr>
<td>80.00</td>
<td>0.79</td>
</tr>
<tr>
<td>100.00</td>
<td>0.89 (p = 0.022q - 1.3125)</td>
</tr>
<tr>
<td>120.00</td>
<td>1.33</td>
</tr>
<tr>
<td>140.00</td>
<td>2.11</td>
</tr>
<tr>
<td>160.00</td>
<td>2.21</td>
</tr>
<tr>
<td>180.00</td>
<td>2.45</td>
</tr>
<tr>
<td>200.00</td>
<td>3.09</td>
</tr>
<tr>
<td>220.00</td>
<td>3.83 (p = 0.01666q + 0.1666)</td>
</tr>
<tr>
<td>240.00</td>
<td>4.17</td>
</tr>
<tr>
<td>260.00</td>
<td>4.30</td>
</tr>
<tr>
<td>280.00</td>
<td>4.35</td>
</tr>
<tr>
<td>300.00</td>
<td>5.17</td>
</tr>
</tbody>
</table>

### DRAM ADDRESS BUS TIMING DERATING

<table>
<thead>
<tr>
<th>Reason</th>
<th>Capacitive Load (pF)</th>
<th>Additional Propagation Delay (ns)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRAM Address Inputs</td>
<td>160.00</td>
<td></td>
</tr>
<tr>
<td>E250 Output Microstrip/Strip Lines</td>
<td>60.00</td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td>220.00</td>
<td>3.80</td>
</tr>
</tbody>
</table>

Timings for No Cache Configuration (Continued)
EQUATIONS FOR DRAM TIMINGS (82385 Active):

**Read and Write Cycles (Common Parameters):**

- **tRC:** Random Read or Write Cycle Time  
  \( \text{CLK2} \times 10 \)

- **tRP:** RAS \( \Rightarrow \) Precharge Time  
  \( \text{CLK2} \times 4 \)

- **tRAS:** RAS \( \Rightarrow \) Pulse Width  
  \( \text{CLK2} \times 4 \)

A random DRAM cycle may have a RAS \( \Rightarrow \) pulse which is only four CLK2 periods wide. This is the case if the cycle is followed by Idle cycles (DRAMs not selected or TI's) or a DRAM page miss.

- **tCAS (Read):** CAS \( \Rightarrow \) Pulse Width  
  \( \text{CLK2} \times 5 \)
  
  CAS \( \Rightarrow \) pulses can be as narrow as five CLK2 cycles during Page Mode read cycles.

- **tCAS (Write):** CAS \( \Rightarrow \) Pulse Width  
  \( \text{CLK2} \times 2 \)
  
  CAS \( \Rightarrow \) pulses can be as narrow as two CLK2 cycles during Page Mode write cycles.

- **tASC:** Column Address Setup Time  
  \( \text{min} (\text{CLK2} \times 2 + \text{AS32.tphl.min} - \text{Delay.max} - \text{ACT258.StoZ.tpl.min} - \text{ACT258.Cap.Derating}, \text{CLK2} \times 3 + \text{AS32.tphl.min} - \text{t6.max} - 386.\text{Cap.Derating} - \text{AS373.GtoO.tpl.max} - \text{ACT258.StoZ.tpl.max} - \text{ACT258.Cap.Derating}) \)

  The Column Address becomes valid as RAS \( \Rightarrow \) switches from High to Low or as the 386 address becomes valid while RAS \( \Rightarrow \) is already Low (i.e., Page Mode, Pipelined cycles)

- **tCAH:** Column Address Hold Time  
  \( \text{CLK2} + \text{AS373.GtoO.tpl.min} + \text{ACT258.StoZ.tpl.min} - \text{AS32.tphl.max} \)

  The CAL (Column Address Latch) signal is activated one CLK2 period after the active-going edge of CAS \( \Rightarrow \).

- **tAR:** Column Address Hold Time to RAS \( \Rightarrow \)  
  \( \text{CLK2} \times 3 + \text{AS373.GtoO.tpl.min} + \text{ACT258.StoZ.tpl.min} - \text{RAS.Delay.max} \)

- **tRCD:** RAS \( \Rightarrow \) to CAS \( \Rightarrow \) Delay Time  
  \( \text{CLK2} \times 2 + \text{AS32.tphl.min} - \text{RAS.Delay.max} \)

- **tRAD:** RAS \( \Rightarrow \) to Column Address Delay Time  
  (min) \( \text{ACT258.StoZ.tpl.min} + \text{Delay.min} - \text{RAS.Delay.max} \)
  (max) \( \text{ACT258.StoZ.tpl.max} + \text{Delay.max} + \text{ACT258.Cap.Derating} - \text{RAS.Delay.min} \)

- **tRSH:** RAS \( \Rightarrow \) Hold Time  
  \( \text{CLK2} \times 2 - \text{AS32.tphl.max} + \text{RAS.Delay.min} \)

  The worst case occurs when a DRAM Page miss or Idle is detected at the end of the current DRAM Page miss cycle.

- **tCSH:** CAS \( \Rightarrow \) Hold Time  
  \( \text{CLK2} \times 6 + \text{AS32.tphl.min} - \text{RAS.Delay.max} \)

- **tCRP:** CAS \( \Rightarrow \) to RAS \( \Rightarrow \) Precharge Time  
  \( \text{CLK2} \times 2 + \text{RAS.Delay.min} - \text{AS32.tphl.max} \)

  This is guaranteed by the DRAM control state machine.

- **tASR:** Row Address Setup Time  
  \( \text{CLK2} \times 2 - \text{t6.max} - 386.\text{Cap.Derating} - \text{ACT258.StoZ.tpl.max} - \text{ACT258.Cap.Derating} + \text{H124.tpl.min} + \text{H125.tpl.min} + \text{PAL.tco.min} + \text{RAS.Delay.min} \)

- **tRAH:** Row Address Hold Time  
  \( \text{ACT258.StoZ.tpl.min} + \text{Delay.min} - \text{RAS.Delay.max} \)

- **tT:** Transition Time (Rise and Fall)

- **tREF:** Refresh Period

- **tREF2:** Refresh Period
Read Cycles:

\[ t_{RAC}: \text{Access Time} \]
\[ \text{CLK2} \times 8 - H124.tpd.max - H125.tpd.max - \]
\[ \text{PAL.tco.max} - F245.max - \text{AS646.tpd.max} - \]
\[ F245.max - \text{RAS.Delay.max} - \text{SRAM.tDW} - \text{CLK2} + \]
\[ 385.t22a.min \]

\[ t_{CAC}: \text{Access Time from CAS} \# \]
\[ \text{CLK2} \times 5 - H124.tpd.max - H125.tpd.max - \]
\[ \text{PAL.tco.max} - \text{AS32.tphl.max} - F245.max - \]
\[ \text{AS646.tpd.max} - F245.max - \text{SRAM.tDW} - \text{CLK2} + \]
\[ 385.t22a.min \]

\[ t_{AA}: \text{Access Time from Address} \]
\[ \text{CLK2} \times 8 - 16.max - \text{386.Cap.Derating} - \]
\[ \text{AS373.tphl.max} - \text{ACT258.ttoZ.tpd.max} - \]
\[ \text{ACT258.Cap.Derating} - F245.max - \text{AS646.tpd.max} - \]
\[ F245.max - \text{SRAM.tDW} - \text{CLK2} + 385.t22a.min \]

\[ t_{RCS}: \text{Read Command Setup Time} \]
\[ \text{CLK2} + \text{AS32.tphl.min} \]

\[ t_{RCH}: \text{Read Command Hold Time to CAS} \# \]
\[ \text{CLK2} - \text{AS32.tphl.max} \]

\[ t_{RHR}: \text{Read Command Hold Time to RAS} \# \]
\[ \text{CLK2} - \text{RAS.Delay.max} \]

\[ t_{OFF}: \text{Output Buffer Turn-off Time} \]
\[ \text{CLK2} \times 2 + F245.tzh.min \]

Write Cycles:

\[ t_{WCS}: \text{Write Command Setup Time} \]
\[ \text{CLK2} \times 3 + \text{AS32.tphl.min} \]

\[ t_{WCH}: \text{Write Command Hold Time} \]
\[ \text{CLK2} \times 2 - \text{AS32.tphl.max} \]

\[ t_{WCR}: \text{Write Command Hold Time to RAS} \# \]
\[ \text{CLK2} \times 6 - \text{RAS.Delay.max} \]

\[ t_{WP}: \text{Write Command Pulse Width} \]
\[ \text{CLK2} \times 5 \]

\[ t_{RWL}: \text{Write Command to RAS} \# \text{ Lead Time} \]
\[ \text{CLK2} \times 5 + \text{RAS.Delay.min} \]

\[ t_{CWL}: \text{Write Command to CAS} \# \text{ Lead Time} \]
\[ \text{CLK2} \times 5 \]

\[ t_{DS}: \text{Data-in Setup Time} \]
\[ \text{CLK2} \times 3 + H124.tp.min + H125.tp.min + \]
\[ \text{AS32.tphl.min} - 385.t43c.max - \]
\[ \text{AS646.GotoO.tpd.max} - F245.tpd.max \]

\[ t_{DH}: \text{Data-in Hold Time} \]
\[ \text{CLK2} \times 2 + F245.tpz.min - \text{AS32.tphl.max} \]

\[ t_{DHR}: \text{Data-in Hold Time to RAS} \# \]
\[ \text{CLK2} \times 6 + F245.tpz.max + \text{RAS.Delay.min} \]

Page Mode Cycles:

\[ t_{PC}: \text{Page Mode Cycle Time} \]
\[ \text{CLK2} \times 6 \]

\[ t_{RAPC}: \text{Page Mode RAS} \# \text{ Pulse Width} \]
\[ \text{CLK2} \times 4 \]

\[ t_{RSW}: \text{RAS} \# \text{ to Second WE} \# \text{ Delay Time} \]
\[ \text{CLK2} 	imes 7 - \text{RAS.Delay.max} \]

\[ t_{CP}: \text{CAS} \# \text{ Precharge Time} \]
\[ \text{CLK2} \]

\[ t_{WI}: \text{Write Invalid Time} \]
\[ \text{CLK2} \]

\[ t_{CAP}: \text{Access Time from Column Precharge Time} \]
\[ \text{CLK2} \times 6 - H124.tpd.max - H125.tpd.max - \]
\[ \text{PAL.tco.max} - F245.max - \text{AS646.tpd.max} - \]
\[ F245.max - \text{SRAM.tDW} - \text{CLK2} + 385.t22a.min \]
# DRAM Timing Requirements

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>For 80386-33</th>
<th>Timing Margin (DSM 2801-06)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Minimum</td>
<td>Maximum</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Minimum</td>
<td>Maximum</td>
</tr>
<tr>
<td><strong>Read and Write Cycles (Common Parameters):</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tRH</td>
<td>Random Read or Write Cycle Time</td>
<td>150.00</td>
<td>29.00</td>
</tr>
<tr>
<td>tRP</td>
<td>RAS$ Precharge Time</td>
<td>60.00</td>
<td>5.00</td>
</tr>
<tr>
<td>tRAS</td>
<td>RAS$ Pulse Width</td>
<td>60.00</td>
<td>0.00</td>
</tr>
<tr>
<td>tCAS</td>
<td>CAS$ Pulse Width (Read)</td>
<td>75.00</td>
<td>64.00</td>
</tr>
<tr>
<td>tRAS</td>
<td>CAS$ Pulse Width (Write)</td>
<td>50.00</td>
<td>25.00</td>
</tr>
<tr>
<td>tRAC</td>
<td>Column Address Setup Time</td>
<td>9.70</td>
<td>9.70</td>
</tr>
<tr>
<td>tRH</td>
<td>Column Address Hold Time</td>
<td>14.20</td>
<td>8.20</td>
</tr>
<tr>
<td>tRAS#</td>
<td>Column Address Hold Time to RAS$</td>
<td>50.00</td>
<td>10.00</td>
</tr>
<tr>
<td>tRC#</td>
<td>RAS$ to CAS$ Delay Time</td>
<td>31.00</td>
<td>21.30</td>
</tr>
<tr>
<td>tREAD</td>
<td>RAS$ to Column Address Delay Time</td>
<td>5.00</td>
<td>1.00</td>
</tr>
<tr>
<td>tRP$</td>
<td>RAS$ Hold Time</td>
<td>21.30</td>
<td>4.70</td>
</tr>
<tr>
<td>tC#</td>
<td>CAS$ Hold Time</td>
<td>31.00</td>
<td>51.00</td>
</tr>
<tr>
<td>tCEP</td>
<td>CAS$ to RAS$ Precharge Time</td>
<td>24.20</td>
<td>21.20</td>
</tr>
<tr>
<td>tAAE</td>
<td>Row Address Setup Time</td>
<td>6.20</td>
<td>4.20</td>
</tr>
<tr>
<td>tRAS#</td>
<td>Row Address Hold Time</td>
<td>5.00</td>
<td>3.00</td>
</tr>
<tr>
<td><strong>ST</strong></td>
<td>Transition Time (Rise and Fall)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>tREF</td>
<td>Refresh Period</td>
<td></td>
<td></td>
</tr>
<tr>
<td>tREF2</td>
<td>Refresh Period</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Read Cycles:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tREAC</td>
<td>Access Time</td>
<td>67.50</td>
<td>7.50</td>
</tr>
<tr>
<td>tREAC</td>
<td>Access Time from CAS$</td>
<td>16.70</td>
<td>5.70</td>
</tr>
<tr>
<td>tRAA</td>
<td>Access Time from Address</td>
<td>37.70</td>
<td>5.70</td>
</tr>
<tr>
<td>tRCE</td>
<td>Read Command Setup Time</td>
<td>30.00</td>
<td>20.00</td>
</tr>
<tr>
<td>tRCH</td>
<td>Read Command Hold Time to CAS$</td>
<td>9.20</td>
<td>9.00</td>
</tr>
<tr>
<td>tRCH</td>
<td>Read Command Hold Time to RAS$</td>
<td>15.00</td>
<td>15.00</td>
</tr>
<tr>
<td>tOFF</td>
<td>Output Suffer Turn-off Time</td>
<td>33.00</td>
<td>16.00</td>
</tr>
<tr>
<td><strong>Write Cycles:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tWCE</td>
<td>Write Command Setup Time</td>
<td>46.00</td>
<td>46.00</td>
</tr>
<tr>
<td>tWCE</td>
<td>Write Command Hold Time</td>
<td>24.20</td>
<td>24.20</td>
</tr>
<tr>
<td>tWCH</td>
<td>Write Command Hold Time to RAS$</td>
<td>9.00</td>
<td>50.00</td>
</tr>
<tr>
<td>tWCH</td>
<td>Write Command Pulse Width</td>
<td>75.00</td>
<td>70.00</td>
</tr>
<tr>
<td>tWCH</td>
<td>Write Command to RAS$ Lead Time</td>
<td>75.00</td>
<td>62.00</td>
</tr>
<tr>
<td>tWCH</td>
<td>Write Command to CAS$ Lead Time</td>
<td>75.00</td>
<td>70.00</td>
</tr>
<tr>
<td>tDIS</td>
<td>Data-in Setup Time</td>
<td>25.00</td>
<td>9.00</td>
</tr>
<tr>
<td>tDCH</td>
<td>Data-in Hold Time</td>
<td>31.70</td>
<td>26.70</td>
</tr>
<tr>
<td>tDIP#</td>
<td>Data-in Hold Time to RAS$</td>
<td>97.00</td>
<td>87.00</td>
</tr>
<tr>
<td><strong>Page Mode Cycles:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tPCE</td>
<td>Page Mode Cycle Time</td>
<td>93.00</td>
<td>93.00</td>
</tr>
<tr>
<td>tPCE</td>
<td>Page Mode Pulse Width</td>
<td>60.00</td>
<td>60.00</td>
</tr>
<tr>
<td>tPCE#</td>
<td>Page Mode to Second WE Delay Time</td>
<td>105.00</td>
<td></td>
</tr>
<tr>
<td>tC#</td>
<td>CAS$ Precharge Time</td>
<td>15.00</td>
<td>10.00</td>
</tr>
<tr>
<td>tCE</td>
<td>Write Invalid Time</td>
<td>15.00</td>
<td></td>
</tr>
<tr>
<td>tCEP</td>
<td>Access Time from Column Precharge Time</td>
<td>37.50</td>
<td>3.50</td>
</tr>
</tbody>
</table>
APPENDIX E
REFERENCES

REFERENCES


