

Application Note 42

The first inverter is slowed down due to the k times larger load, while the second inverter benefits from the k times larger transistor scale-up to drive the larger load C_L . Setting the derivative of the propagation delay equation to zero, the optimal value of k is \sqrt{n} , derived by setting dt_{pd}/dk to zero and the optimal propagation delay for the circuit in figure 1 is $2 \times \sqrt{n} \times t_{pdo}$. Comparing the delay of a single inverter driving C_L versus a single inverter plus buffer, it is clear that inserting a buffer makes sense only when $n > 4$, which denotes an equivalent fanout of four gates.

CMOS BUFFER DESIGN — DIGITAL APPROACH

Conventionally in digital systems design, off-chip loads are very large. Hence buffer chips for driving these large capacitive loads are required. These have been conventionally implemented in CMOS using a chain of inverters, as shown in figure 2. The MOSFETS in each inverter stage are scaled up linearly with respect to the previous stage so that each inverter drives a progressively larger load. An optimal design is obtained by scaling up all stages with a constant factor k . This results in an identical delay per stage given by $t_{pd\ STAGE} = k \times t_{pdo}$. To achieve the same delay for the last stage, it is necessary

$$\text{that } t_{pd} = k \times t_{pdo} + \frac{n}{k} \times t_{pdo} = \left(k + \frac{n}{k}\right) \times t_{pdo}.$$

Thus the propagation delay is given by $t_{pd} = n \times k \times t_{pdo}$ or substituting the expression for n ,

$$t_{pd} = t_{pdo} \times I_N(x) \times \frac{k}{I_N(k)}.$$

Setting the derivative of this expression to zero, the optimal value of k is e and the optimal propagation delay is given by :

$$t_{pd(OPT)} = e \times I_N(x) \times t_{pd} = e \times I_N\left(\frac{C_L}{C_i}\right) \times t_{pdo}$$

Thus in order to buffer large capacitance with CMOS logic, it is necessary to cascade an even number of inverters, with each successive inverter being larger than the preceding one, eventually leading to an inverter that will drive the required load capacitance, at the required frequency. However, when doing this, the minimum propagation that can be achieved is proportional to the minimum gate delay of the process.

As an example, if the ratio of output to input load capacitance were 1000 then the propagation delay would be 1000 times that of the process' minimum gate delay in the case of an unbuffered approach, 63 times that of the process' minimum gate delay in the case of a single inverter approach and 19 times that of the process' minimum gate delay in the case of the multiple inverter buffer approach. It is easy to conclude that there is a real but process limited reduction in delay obtained with multiple inverter based buffers when driving very large capacitive loads. Each inverter stage represents an additional delay in the gating process because in order for a single gate to switch, the input must slew more than half of the supply voltage. Today the fastest available CMOS buffer (a member of the FCT-E logic family), has managed to drive a 50pF load with a propagation delay of 3.2ns.

HIGH SPEED BUFFER DESIGN — ANALOG APPROACH

Octal buffers are eight bit logic devices that are capable of driving load capacitance several times larger than their input capacitance. As discussed in the above section, these buffers are typically implemented in CMOS logic (digital approach) and made to be TTL compatible by sizing the input devices appropriately. By using an unique analog circuit approach that does not require cascaded logic gates, we at Micro Linear have produced an octal transceiver (ML65245), which offers a propagation delay of less than 1.5ns, while switching at 50MHz, into a 50pF load. It achieves its low and predictable propagation delay by using feedback techniques to produce an output that follows the input within a couple of hundred milli-volts. If the output voltage is not close to the input, then the feedback will source enough current to the load capacitance to correct the discrepancy.

The basic architecture of this analog approach is shown in figure 3. It is implemented in a 1.5μ BiCMOS process with all the active devices being NPNs — the fastest devices available in this process. In this circuit there are two paths to the output. Assume for the moment that the switches shown in figure 3 are closed. One path sources current to the load capacitance when the signal is asserted and the other path sinks current from the output when the signal is negated. The assertion path is the emitter follower path consisting of the level shift transistor Q1, the output transistor Q2 and the bias resistor R8. It sources current to

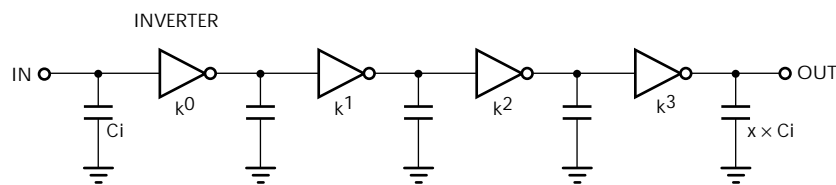


Figure 2. Multiple Inverter Buffer Architecture

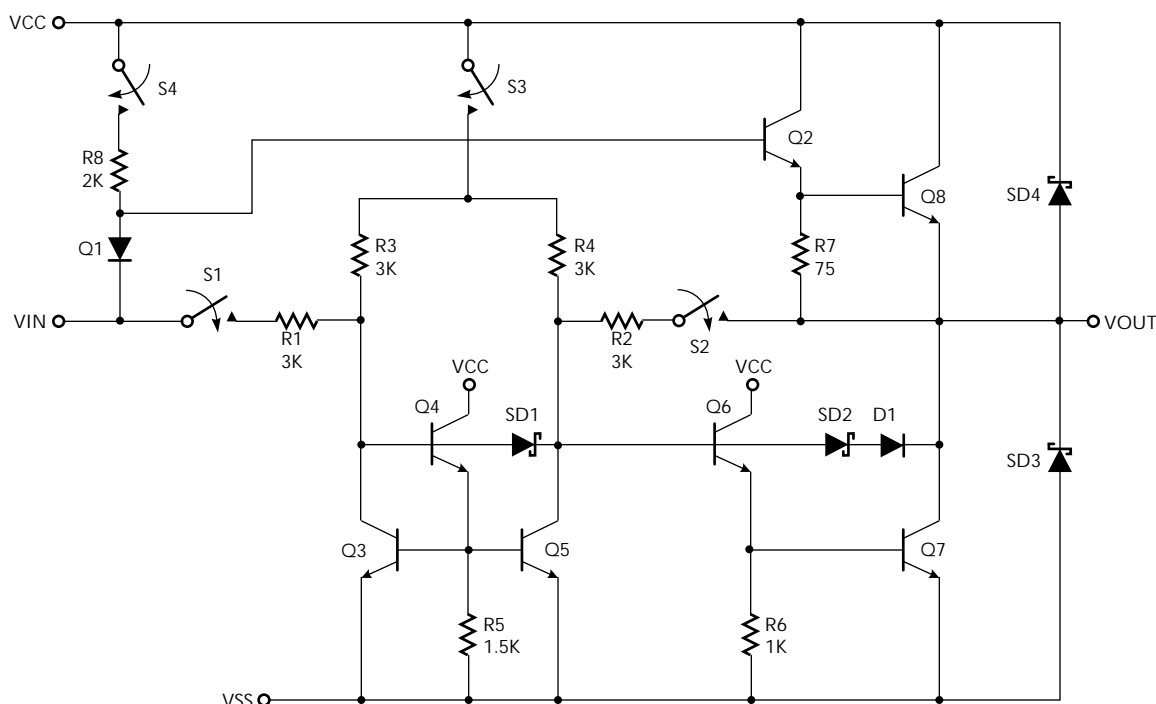


Figure 3. Analog Buffer Architecture.

the output through the 75 ohm resistor R7 which is bypassed by Q8 during fast input transients. The negation path is a current differencing amplifier connected in a follower configuration. The active components in this amplifier are transistors Q3-Q7. R3-R6 are bias resistors and R1 & R2 are feedback resistors. The key to understanding the operation of the current differencing amplifier is realizing that the current in transistors Q3 and Q5 are the same at all times and that the voltages at the bases of Q4 and Q6 are roughly the same. If the output is higher than the input, then an error current will flow through R2. This error current will flow into the base of Q6 and be multiplied by beta2 to the collector of Q7, thus closing the loop. The larger the discrepancy between the output and the input, the larger the feedback current and the harder Q7 sinks current from the load capacitor.

A number of MOSFETS have been used to tri-state outputs and minimize power in disabled buffers. The function of some of these MOSFETS have been included in figure 3 in the form of switches. S1 and S2 ensure that the buffers are tristated when they are open; S3 and S4 ensure that disabled buffers draw no current from V_{CC} . Other switches not shown in the diagram are used to pull the bases of Q4, Q6 and Q2 to ground when the buffer is disabled.

Also shown in the diagram are some diodes that are used as clamps. Two schottky diodes connected to the output SD3 and SD4 protect the output voltage from sustained excursions above and below ground. These also double as the electrostatic discharge protection for both the inputs and outputs of buffers since they are connected back to back in the layout. SD1 keeps the base of Q6 from going to ground. This helps the negation circuit to recover more quickly when the signal is asserted. SD2 and D1 are used as output clamps. They keep the output transistor Q7 from saturating. If Q7 was allowed to saturate, its base would draw lots of current from V_{CC} , and its recovery characteristics would be poor because of the excess charge that would be stored in its base.

The analog circuit implementation of a buffer/transceiver results in a number of advantages. The output rise and fall times closely match those of the input waveform while the output responds almost immediately to the input, with very low skew. Also oscillatory ground bounce is significantly reduced with this approach, as compared to CMOS transceivers, due to the bipolar output structure which damps the output ringing. Another advantage is that the resistor R7 in figure 3 acts like a termination resistor in some cases. This 75 Ω resistor is in series with the output and therefore helps suppress noise caused by transmission line effects such as reflections from mismatched impedances.

Application Note 42

PERFORMANCE CHARACTERISTICS

The analog buffer circuit approach discussed above has been fabricated in silicon as the ML65245 and proven to be successful. To verify the 1.5ns propagation delay through the part, a special printed circuit board was laid out and the buffer was made to drive different load capacitance at different frequencies. Figure 4 shows the input and output waveforms of the buffer driving a single 50pF capacitance at 33MHz. In all buffering applications, some degradation with load capacitance is to be expected. When driving larger load capacitance, the buffer must supply more current to the load in order to maintain a given output slew rate. In Micro Linear's analog buffer, the amount of current that can be supplied to the load is limited by the characteristics of the output NPNs. As the current in these transistors increases, their current gains tend to decrease and the amount of base current available is limited by the bias resistors on the

chip. However, on the ML65245, peak currents to the load capacitance can exceed 150mA. The degradation of the ML65245 with capacitance is shown in figure 5. This data was taken with one line switching at 15MHz.

Also of interest are the input output characteristics of the ML65245. Since CMOS transceivers are gauged only by their output characteristics, the analog architecture we have chosen is unique. Its output characteristics depend on the difference between the input and the output. The output voltage and input current versus input voltage are shown in figure 6. Notice that the output is clamped for inputs less than approximately 0.3 volts and greater than V_{CC} minus 0.7. Also note that the input draws no current when the input voltage is about 3 volts. This means that unlike a CMOS buffer in which the output voltage cannot be determined when the input is allowed to float, the output of the ML65245 has no undetermined state. The output voltage versus output current characteristics of the

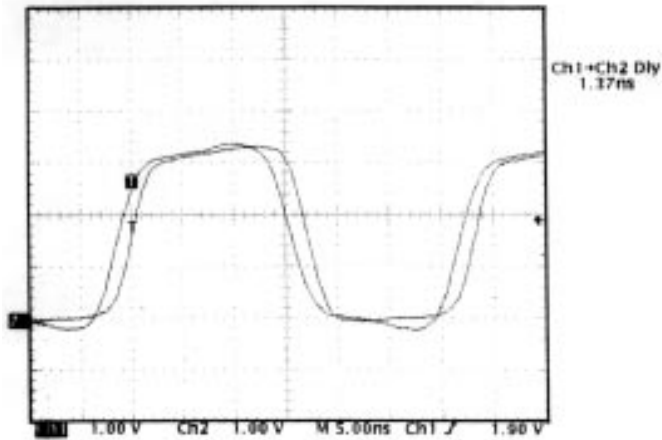


Figure 4. Typical Output Waveform.

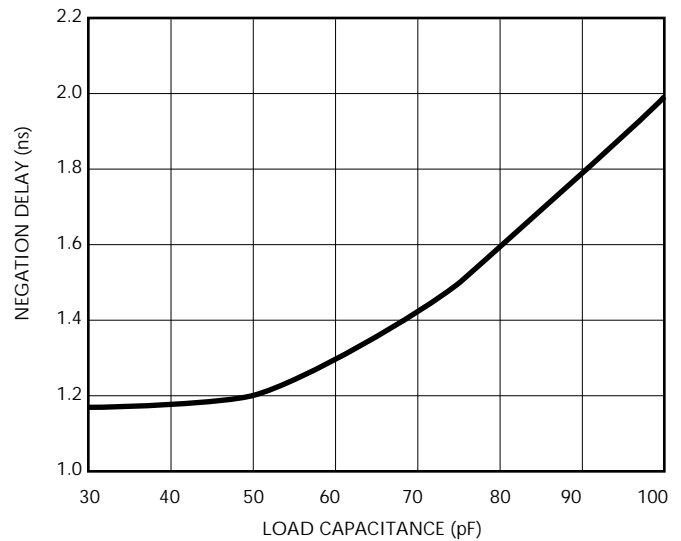


Figure 5. Propagation Delay versus Load Capacitance

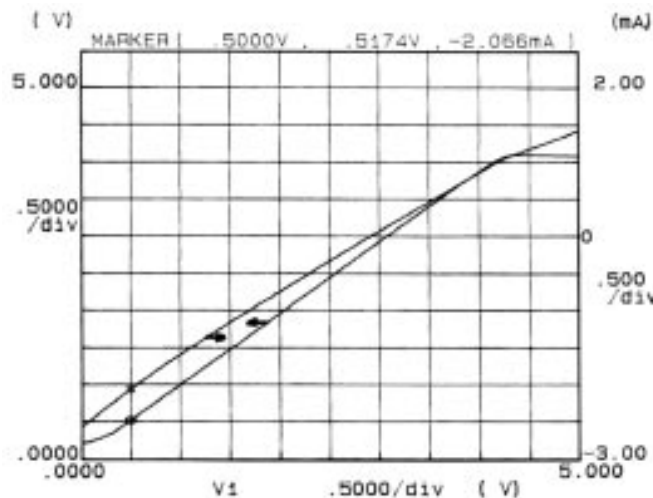


Figure 6. I_{IN} and V_{OUT} vs. V_{IN} .

ML65245 are also important. These are shown in figure 7a. In this case, the input voltage was fixed at two volts and the output current was swept from -50mA to 50mA . In a CMOS buffer application, the system designer would expect that these characteristics would be the same for any input voltage that caused the output to be high. However, in this case that would be wrong, because in the ML65245 the bias and feedback currents available to the output transistors in both paths depend on the input voltage, and the output voltage is clamped slightly above ground and slightly below V_{CC} . Figures 7b and 7c show V_{OH} vs. I_{OH} . In both cases, the outputs can sink or source more than 200mA .

Another interesting performance advantage comes from the fact that the output devices are bipolar and not CMOS. NPN bipolar devices are fabricated to carry current in one direction; from the collector to the emitter. They will carry

current in the opposite direction, but the current gain will be dramatically worse. CMOS devices are symmetric with respect to the gate. When a CMOS gate is "on", it will conduct current equally well from the drain to the source or from the source to the drain. This dramatically affects the operation of the buffer during signal assertion when a capacitor with some initial condition is made to discharge to ground through the lead inductance of the output bond wire, the output device, and the lead inductance of the ground bond wire. Since the CMOS device carries current equally well in both directions, the output tends to oscillate at the L, C, resonant frequency when it reaches ground. In the ML65245, the output does not tend to oscillate since after the initial undershoot, the resistance of the output device becomes much greater than the resistance during the initial discharge of the load capacitor. This is illustrated in figure 8.

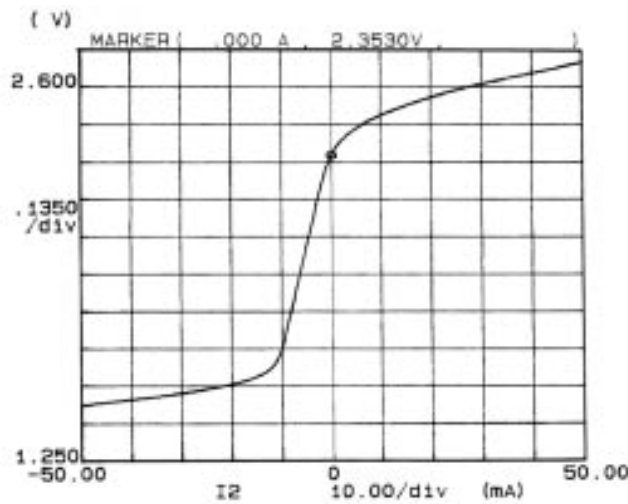


Figure 7a. V_{OUT} vs. I_{OUT} .

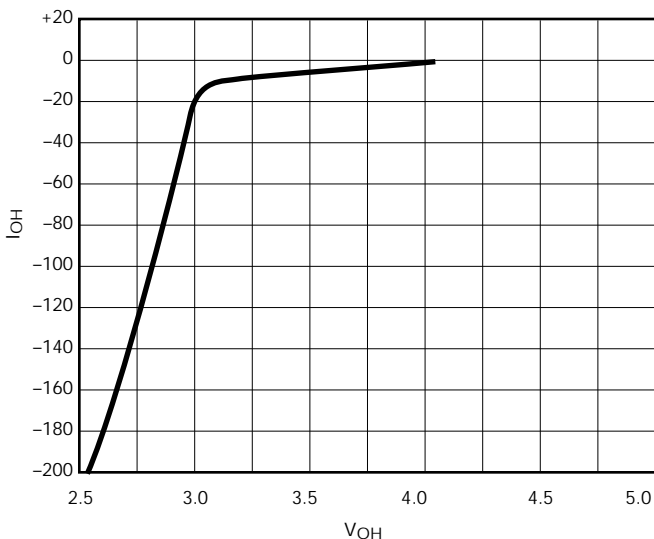


Figure 7b. V_{OH} vs. I_{OH} .

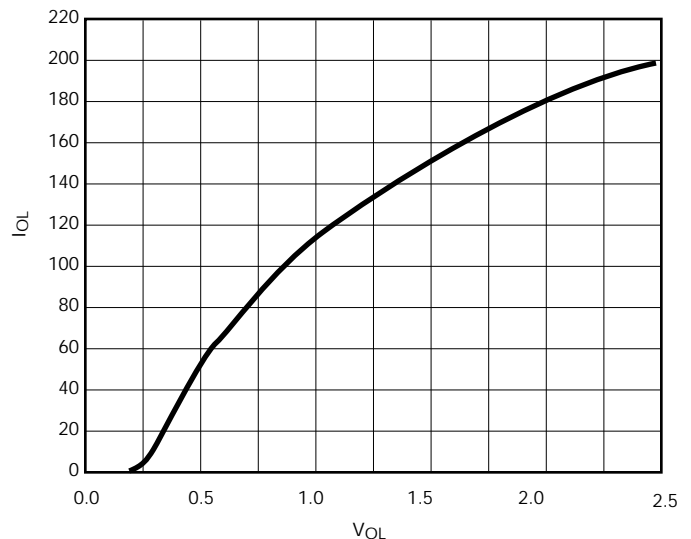
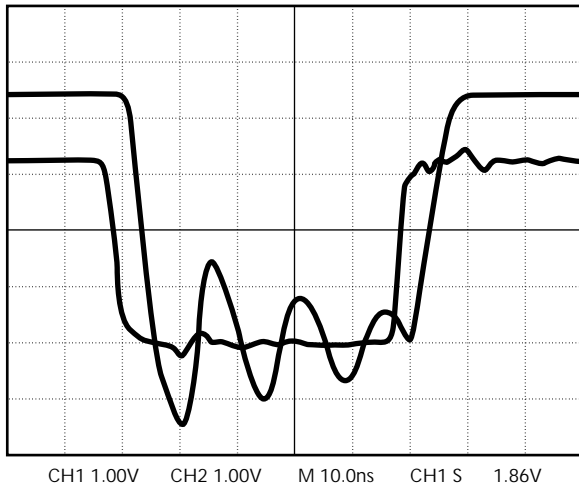
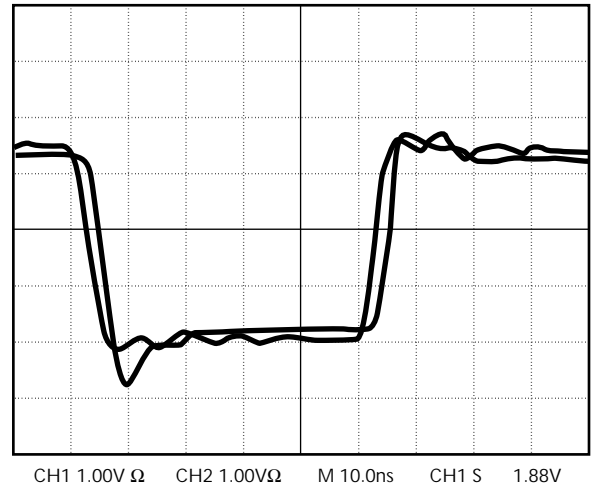


Figure 7c. V_{OL} vs. I_{OL} .



(a) FCT Buffer



(b) ML65245 Buffer

Figure 8. Buffer Output Switching with Four Typical Loads.

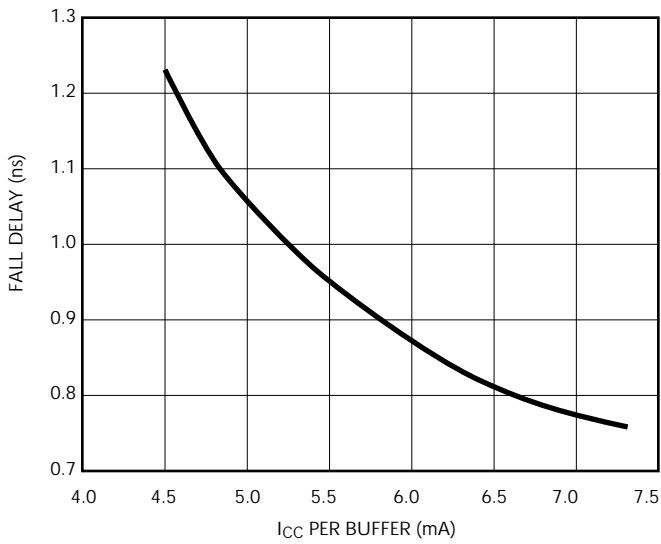


Figure 9. Negation Delay vs. I_{CC}

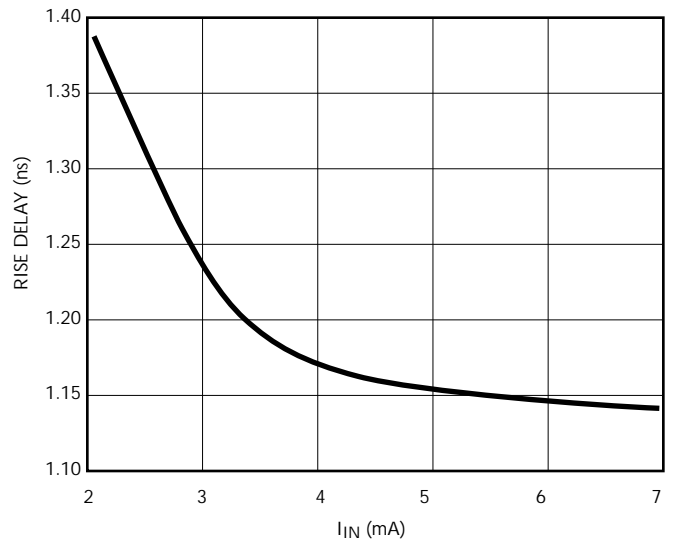


Figure 10. Assertion Delay versus I_{IN} .

Figures 8 (a) and (b) were both obtained using identical test jigs. Four outputs were made to switch loads of 50pF at 1MHz. The only difference was the part used. It is easy to see that there is a significant threat of double clocking using the CMOS part because after the initial transition, the ringing in figure 8 (a) comes back up to the TTL threshold.

LIMITATIONS OF THE ANALOG APPROACH

The limitations of this analog approach to digital transceiver design can be broken down into two questions. First, what limits the performance of the existing circuit design?; and second, what impact does this have on digital systems design?. What ultimately limits the performance of the analog buffer as a circuit is the process it was fabricated on. The goal of this circuit is to minimize propagation delay. Since there are two paths from the input to the output, the propagation delay can be broken down into assertion and negation delays. The speed limitation is a matter of charging and discharging internal capacitances with a given amount of idle current. Various bias currents in the buffer have a great impact on the propagation delay. Referring to the figure 3, the values of R1 through R4 determine the bias current in the current differencing amplifier that makes up the negation path. Decreasing these resistors speeds up that amplifier at the expense of power consumption. Negation delay versus I_{CC} is shown in figure 9. Similarly, the assertion path is largely dependent on the quiescent output current and the bias resistor R8. Decreasing this resistor decreases the assertion delay at the expense of input bias current. This relationship is shown in figure 10.

Another limitation of the analog buffer is that other digital functions are not readily adaptable to this scheme. The ML65245 performs the function V_{OUT} equals V_{IN} . In order to create an analog inverter, one would have to implement the function V_{OUT} equals V_{OH} minus V_{IN} . In order to implement this function with an amplifier, some additional information is needed. The chip would have to have a pin for the user to input the voltage V_{OH} . Hence, industry standard pin out could not be used. Another limitation along these lines is that latched buffers would be difficult without additional delay since there is no logic level

comparison being made in the existing buffer. Because of this, the Micro Linear family of buffers consists of only three products — the ML65245 (octal transceiver), the ML65244 (FCT244 compatible octal buffer), and the ML65541 (FCT541 compatible octal buffer).

In addition to the limitations of the buffer as a circuit, there are issues that the systems designer should be aware of when using this type of buffer in his system. First, since the buffer is analog, it conforms to the analog function $V_{OUT} = V_{IN}$. For example if V_{IN} is 1.5 volts, that is where the output will want to be. If the system designer inputs a waveform that has overshoot, undershoot, ringing, and glitches, the output will have all of these to the extent that its bandwidth and large signal response can reproduce them. This may have a negative impact on the digital system in which the buffer is operating. However, it is difficult to say whether the effect will be greater using the ML65245 than it would be if a standard CMOS buffer were exposed to the same input glitches. Hence, such glitches should always be avoided in the overall system design.

THE ANALOG ADVANTAGE

Because its output follows its input, the Ultra Fast Buffer (UFB) series has several advantages over conventional logic buffers.

- It is FAST, 1.5ns delay at 50pF load. The linear amplifier imposes a small delay, and the rest of the delay is due to driving the capacitive load.
- Because it is very fast and because there is a well defined, stable linear amplifier delay from input to output, skew between outputs is very low, typically 0.25ns or less for matched output loading.
- Because the output is well controlled as the input ground bounce is exceptionally low, (less than 400mV, typical) even at 1.5ns propagation delay. This removes a significant worry item from your design list.
- The resistor in the output damps reflections and noise from other parts of the system, also helping to keep down system noise.
- It is inherently 3.3 volt compatible. The output follows the input. If the input swings between 0 and 3.3 volts, the output will swing between 0 and 3.3 volts.

SYSTEM APPLICATIONS

There are a wide variety of existing and future needs for high speed buffer/transceiver, especially in the main memory and cache memory designs of very high speed processor systems like Pentium™, PowerPC™, MIPs R4000™, Sparc™, etc. If the digital system designer can save several nanoseconds by using a higher speed buffer, then she has the option of maintaining system performance by using a cheaper memory or increasing system performance without spending more for a faster memory.

BUFFERING MAIN MEMORY

An example of a main memory application for the Intel PCI chipset with the Pentium™ processor is shown in figure 11. This discussion is only intended as a general reference. For details please refer to the appropriate Intel documentation. This system has a 66MHz host processor and a 33MHz main (DRAM) memory bus. The main memory row and column addresses (RAS & CAS) and write enable (WE) signals are provided by the PCMC chip (PCI Cache and Memory Controller). The DRAM SIMMs inputs present a heavy load to the PCMC and must be buffered. Three buffered copies of the address signals and write enable are required to drive the six row array. Using the ML65245 to buffer these signals gives the system designer extra margin to be able to use memory modules slower than the normally required 50/70ns modules. The burst read (page-hit) performance is typically 7-4-4-4 at

66MHz for 70ns DRAMs or 6-3-3-3 at 66MHz for 50ns DRAMs. This usually translates to significantly higher costs. With the speed improvement offered by the ML65245, a 6-3-3-3 burst with 60ns DRAMs may be achievable. This kind of main memory application for the ML65245 could potentially extend to other kinds of processor systems which do not require latched buffering.

Figure 12 shows a main memory design example with the ML65245 for the MIPs R4X00 RISC processor based system without secondary cache. As shown in the figure the ML65245 could be used as a data I/O transceiver or as an address buffer. The faster propagation delay essentially translates to a faster main memory access which allows for cost savings by using slower CDRAMs or DRAMs.

BUFFERING CACHE MEMORY

With the advent of higher power operating systems like Windows NT, NeXTStep, Cairo, etc, RISC processor designs like MIPs R4000 series are gaining momentum. In these systems the interface to secondary cache is a critical path in the address and bus control pins. Referring to figure 13, any propagation delay saved in the buffer translates to a slower SRAM access requirement. Consider a CPU design where the secondary cache bus operates at 75MHz. Table 1 examines the timing allocations for each step of the cache RAM t_{AA} .

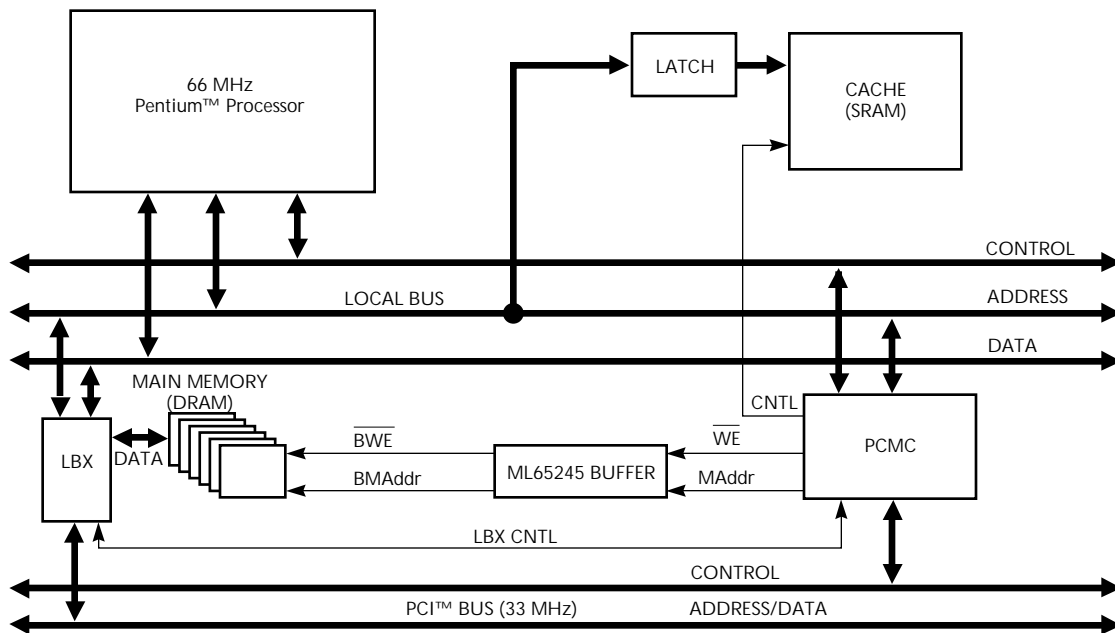


Figure 11. Main Memory Buffering for a Pentium System.

TABLE 1: TIMING ALLOCATIONS FOR CACHE RAM

	TWO CYCLES	CLK TO ADDR	BUFFER t_{pd}	DERATING	CPU SETUP	t_{AA}
FCT-E	26.5ns	-7ns	-3.5ns	-3ns	-3ns	10ns
ML65245	26.5ns	-7ns	-1.5ns	-3ns	-3ns	12ns

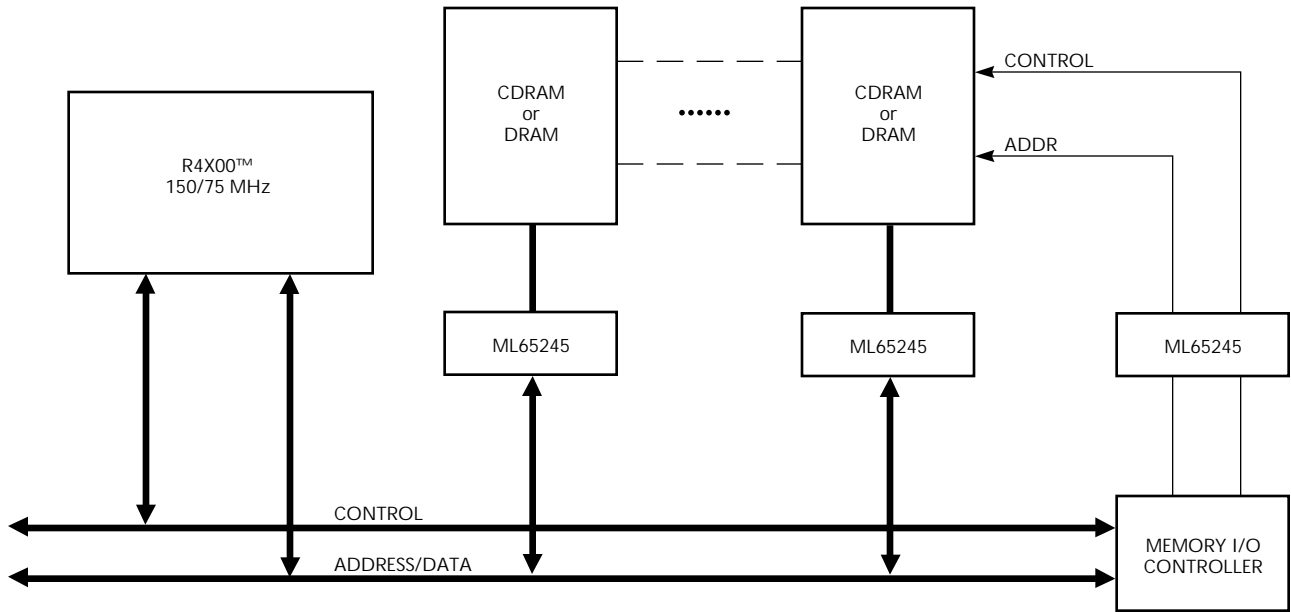


Figure 12. Main Memory Buffering for Non-cache MIPs R4000 System.

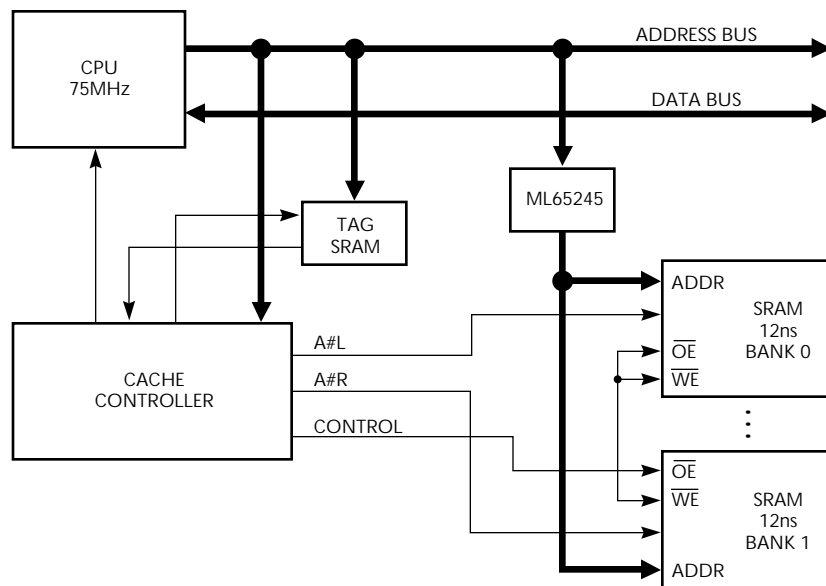


Figure 13. Cache Memory Buffering for a 75MHz CPU System with a Latched Address Bus.

Application Note 42

Based on the above calculations, for a zero-wait-state cache performance, with minimal board space utilization, it is clear that with the fastest FCT buffer, 10ns SRAMs would be required for the cache memory, while with the ML65245 high speed buffer one could use 12ns SRAMs. This difference becomes even more significant in higher speed (100 + MHz) systems where the cache access times could be on the order of 6ns to 8ns. This access time difference could very well mean the difference between using expensive BiCMOS SRAMs versus less expensive CMOS SRAMs.

EXAMPLE COST ANALYSIS OF A R4000 SECONDARY CACHE MODULE BLOCK DESIGN

Shown below in figure 14 is a 256KB secondary cache module block for the R4000 family of RISC processors. It is built on a multilayer epoxy laminate substrate using eleven 16K x 4 SRAMs and two 74FBT2827 buffer/drivers. Generally four identical cache module blocks compromise a full secondary cache in a R4000 based 50MHz/75MHz, zero-wait state system. Table 2 shows a comparative cost analysis of building this cache module using the 74FBT2827 (which have a propagation delay of 4.0ns) versus using the ML65245 (which has a propagation delay of 1.5ns). The SRAM access time t_{AA} is the Cache Module speed minus Buffer t_{pd} . For purposes of this analysis it is assumed that the cost of the two buffers are more or less the same and the SRAM cost is based on distributor prices. This is just intended to show the relative order of savings, rather than the absolute value.

Hence it is clear that the high speed ML65245 results in significant cost savings through the use of slower SRAMs, as shown.

A SYNCHRONOUS DRAM MODULE

High performance computer systems with 66MHz and faster buses are starting to use synchronous DRAMs. These DRAMs are clock driven and have registered inputs and outputs. A typical 1 meg by 32 synchronous DRAM module would use 18 synchronous 1 meg by 4 DRAMs. This means each DRAM address line has 18 capacitive loads. The address and control pins for these 18 DRAMs must be buffered to keep this loading from the CPU, particularly because a system uses up to four modules.

Buffering the address and control lines for a synchronous DRAM module at 66MHz is not easy, as we can see from examining typical timing numbers. Assuming the address arrives at the module at 8ns after the rising edge of the clock and that the DRAM has a 3ns clock setup time., this leaves us only $15 + (8 + 3) = 4ns$ for buffering. If we do not make the 4ns speed, we must add a wait state with its performance degradation. We could use a conventional 74FCT244D at 3.8ns delay, but this would leave us with only 0.2ns margin, not counting flight delay on the module at 0.25ns per inch. If we use a ML65244 at 1.5ns, we have 2.5ns margin for flight delay and design margin.

TABLE 2: COMPARATIVE COST ANALYSIS

MODULE t_{AA}	SRAM t_{AA} W/ 2827 BUFFER	SRAM COST W/ 2827 BUFFER	SRAM t_{AA} W/ ML65245 BUFFER	SRAM COST W/ ML65245 BUFFER	COST SAVINGS W/ THE ML65245	SAVINGS PER PART
12ns	10.5ns	—	8ns	\$1,562	\$00	—
15ns	13.5ns	\$1,562	11ns	\$1,177	\$385	\$130
17ns	15.5ns	\$1,177	13ns	\$858	\$319	\$106
20ns	18.5ns	\$858	16ns	\$330	\$528	\$176
25ns	23.5ns	\$275	21ns	\$275	—	—

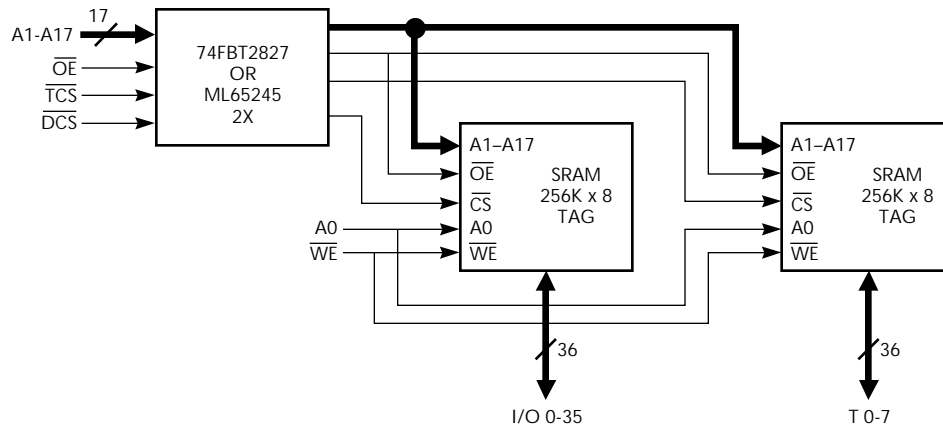


Figure 14. R4000 Secondary Cache Module Block Diagram.

THE UFB IN HIGH SPEED BUS DESIGNS

The UFB solves many difficult timing problems in high speed bus designs. Let us examine high speed memory subsystems to identify these problems. The block diagram of figure 15 shows a CPU with fast SRAM memory subsystem. The SRAM memory subsystem could be the main memory for high speed DSP processor such as the 320CX0, or it could be the external cache for a 386, 486, Pentium or other high speed RISC or CISC CPU.

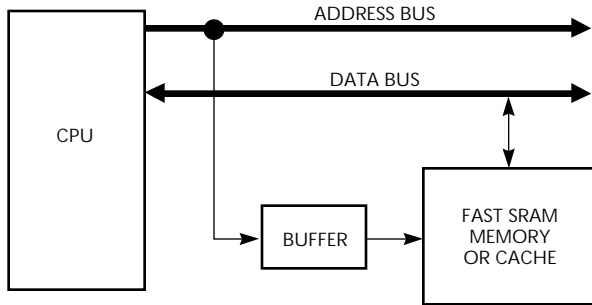


Figure 15. CPU Memory System Block Diagram.

In this block diagram, the address lines to the SRAMs are driven by a buffer. This buffer reduces the capacitive load on the CPU address bus. This is usually necessary if the SRAM system has 8 or more chips, because otherwise the capacitive loading of the SRAMs would significantly slow down the address bus for the whole system.

Figure 16 shows a general timing diagram for a CPU memory read data transfer. This timing diagram is common to 386, 486 and Pentium CPUs, and also applies to other CPUs such as the 320C30 DSP CPU and many RISC processors using different signal names. The timing diagram shows a single read data transfer. This could also be the first word of a multi-word burst data transfer.

The CPU generates the memory address after a delay of t_{ADDR} from the start of the cycle. t_{ADDR} is usually about half a clock period. The address buffer adds a delay of t_{BUFFER} to the address signals. The SRAM adds a read access delay, t_{AA} . Finally, there is the data setup time required by the CPU, t_{DS} . An additional time for printed circuit trace delay called flight time (not shown), must be added. This delay, t_{TRACE} , represents the finite propagation delay of electrical signals along the printed circuit board traces. The sum of all these delays must be less than the time available, which is two clock periods in this case. The design timing margin, t_{MARG} , represents this excess time.

A good timing margin might be 5% of the time available. As CPU and bus speeds increase, this timing margin is harder to achieve. Table 3 illustrates this problem by showing hypothetical timing margins for x86 processors from 386 to Pentium.

The timing margin for the 386 system using 7ns buffer is generous. The speed of the memory subsystem is determined primarily by the 386 CPU in this case. In the 486 case, the propagation delays of the buffer and the traces are starting to become significant, but are still manageable. In the Pentium case, the propagation delay of the buffer becomes critical. If 12ns SRAMs are used, the timing margin becomes negative: you didn't make it in time. Using ultra fast buffers, however, converts this negative margin to a positive one.

If 12ns SRAMs are used, the timing margin becomes negative: you didn't make it in time. Using an ultra fast buffer, however, converts this negative margin to a positive one.

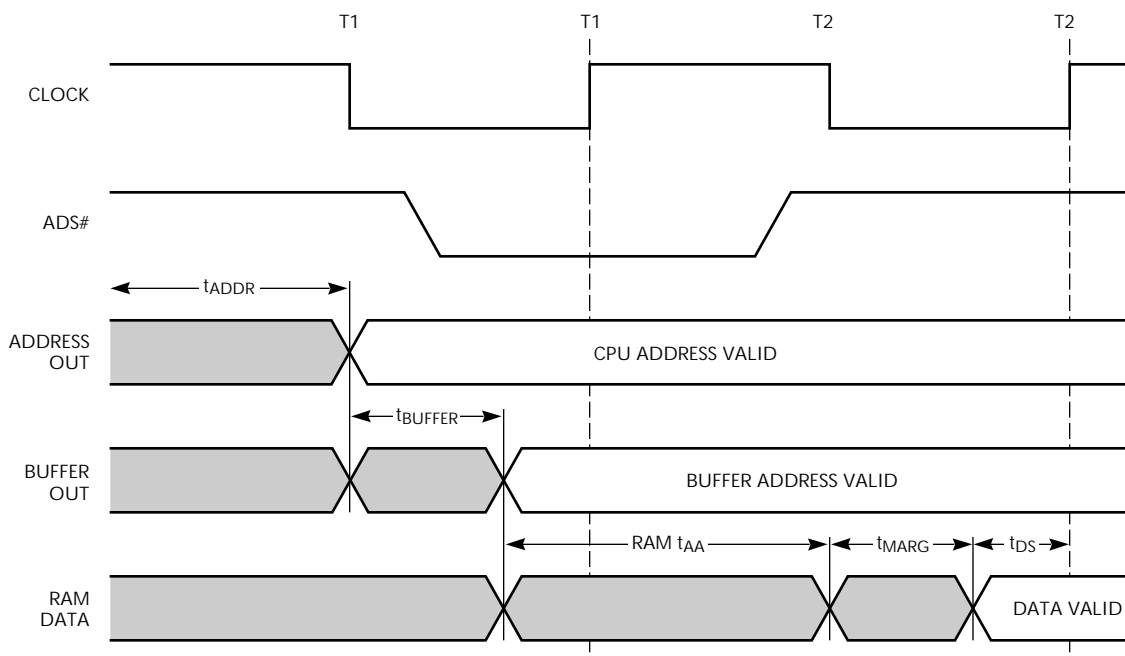


Figure 16. Memory Data Transfer Timing Diagram.

TABLE 3: DESIGN MARGIN FOR X86 CPU MEMORY SUBSYSTEMS

PARAMETER	386	486	PENTIUM	PENTIUM + UFB	UNITS
CPU Clock	16	33	66	66	MHz
Clock Cycle Time	62.5	30	15	15	ns
Total Time (2 Cycles)	125	60	30	30	ns
t_{ADDR}	40	16	10	10	ns
t_{BUFFER}	7	5	3.8	1.5	ns
t_{TRACE}	3	2	1.5	1.5	ns
SRAM t_{AA}	35	25	12	12	ns
CPU t_{DS}	10	5	3.8	3.8	ns
Total	95	53	31.1	28.8	ns
Margin, t_{MARG}	30	7	-1.1	+1.2	ns
Timing Margin, Percent	24%	11.7%	-3.6%	+4.0%	ns

Another way to view the significance of buffer delay is as a percentage of the bus clock period for various bus speeds, as shown in Table 4. This table shows that the 74FCT244D delay becomes more than 10% of the bus clock cycle time above 25 MHz. The ML65244 delay just reaches 10% at 66MHz. Even at 100MHz, the ML65244 delay is still only 15% of the bus clock cycle time.

Table 4: Buffer Delay versus Clock Period

BUS CLOCK FREQ. MHz	BUS CLOCK PERIOD ns	FCT244D % AT 3.8ns	ML65244 % AT 1.5ns
10	100	3.8	1.5
20	50	5.6	3.0
25	40	9.5	3.8
33	30	12.7	5.0
40	25	15.2	6.0
50	20	19.0	7.5
66	15	25.3	10.0
80	12.5	30.4	12.0
100	10	38.0	15.0

CACHE DESIGN: THE VALUE OF SPEED

Many systems need to re-power the address lines to the cache memory to drive the memory address capacitance. If you use a conventional high speed logic buffer such as the 74FCT244D, this usually means adding a wait state to the cache access time to compensate for the buffer delay. This degrades cache performance. L2 caches for 486 and Pentium systems use four word lines. This is typical of most other CISC and RISC CPU's. Using a ML65244 can save you from having to add this wait state.

A zero wait state design will transfer four words from the L2 cache to the CPU on a cache miss. This will take 5 clock cycles for minimum, zero wait state timing, also called 2111 timing. A one wait state design will require 6 clock cycles, or 3111 timing. The speed degradation versus wait states for four word line caches is shown in Table 5. Note that a single wait state degrades the CPU effective clock speed from 90MHz to 75MHz.

Table 5: Cache Speed versus Wait States

WAIT STATES	TIMING	SPEED	CPU CLOCK MHz	NET CLOCK MHz
0	2111	100%	90	66
1	3111	83%	75	55
2	4111	71%	64	47
3	5111	63%	56	41

There is some argument today that adding one wait state to an L2 cache does not seriously degrade speed. This is an argument from necessity because in most designs you have to add a wait state because you have no fast buffer to save you the necessary time. To the degree that you need an L2 cache, to that same degree, you need it fast. Adding a wait state causes a 17% speed reduction if your software is heavily using the L2 cache. This is a real problem if you are paying extra money for a fast CPU, only to lose your performance gain in the L2 cache timing.

The degradation in system speed shown in Table 3 assumes that the L2 cache performance limits CPU performance. This is a reasonable assumption, and becomes more reasonable with time. L2 cache use depends on the CPU L1 cache miss rate. The L1 cache miss rate depends on CPU internal speed and program statistics. As CPU internal speeds rise, the L1 cache miss frequency in misses per second will rise as well, for a given program. For example, a 100MHz Pentium with a 1.5:1 clock multiplier will generate cache misses at 1.5 times the rate of a 66MHz CPU. For a constant 66MHz bus speed, this means that the L2 cache will be accessed 1.5 times more often. Also, the more efficient you make the CPU internally, the more the L2 cache determines your performance.

A PENTIUM™ CACHE EXAMPLE

The 66MHz Pentium™ CPU uses an L2 cache. A zero wait state L2 cache has 2111 timing, achieving zero wait usually requires 66MHz burst mode SRAMs. The UFB allows you to make a 2111, zero wait state L2 cache using 12 nanosecond 32k x 8 asynchronous SRAMs.

When the L1 cache in the CPU misses, the CPU gets a four word burst of data from the L2 cache. The first cycle of a four word burst read is the critical cycle for timing. Figure 17 shows a timing diagram for this cycle. The timing margin, t_{MARG} , must be positive, and is calculated in Table 6.

In the first clock period, T1, the CPU puts the address on the bus and asserts ADS#. For zero wait states, the L2 cache returns the first word of data at the end of T2. This two cycle first access corresponds to the 2 in the 2111 timing. The remaining three words come out on each successive clock cycle.

The address sequencing for the four words is unusual but well suited to L2 cache design. The second word has the same address as the first but with the least significant bit inverted. The third word has the same address as the first with the second least significant bit inverted, and the third address is the same as the first with both least significant bits inverted.

Figure 18 shows a block diagram of the Pentium L2 cache design. This design uses two banks of 32k x 8 SRAMs, for a 512 KByte cache.

The cache works by running the two banks in overlap. Both banks access during the first T2 cycle, T2a, and the cache control logic enables the output of bank 0 or 1 depending on the state of the least significant bit of the address. The CPU supplies the address for both banks through the ML65245's.

Figure 19 shows a timing diagram of the four word burst sequence. At the end of T2a, the CPU clocks in the data from the first bank. The cache controller turns off the first bank output enable and turns on the second bank's output enable. The second bank supplies the second word of data at the end of the second T2 cycle, T2b.

Also at the end of T2a, the cache control logic clocks the address from the CPU into both FCT374's. The second least significant bit of the address is inverted as it is clocked into the 374's. This is the correct address for 3 and 4 of the four word burst. the four word burst.

After clocking the address into the 374's, the cache control logic turns off 245's for the first bank and turns on its 374's. This starts the first bank accessing the third word of the burst, to be put on the bus in T2d. The first bank has two cycles of time to get ready for T2c. The second bank switches from its 245's at the end of T2b, in the same manner as the first bank, and it generates its word of data for T2d.

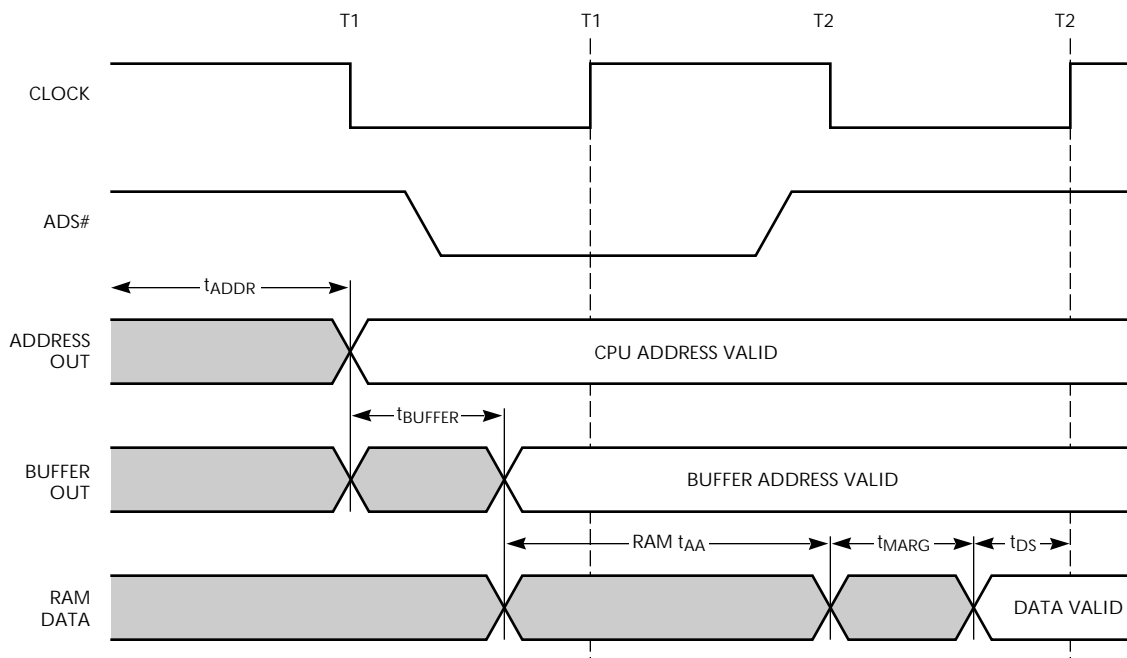


Figure 17. Pentium™ First L2 Cycle Timing Diagram.

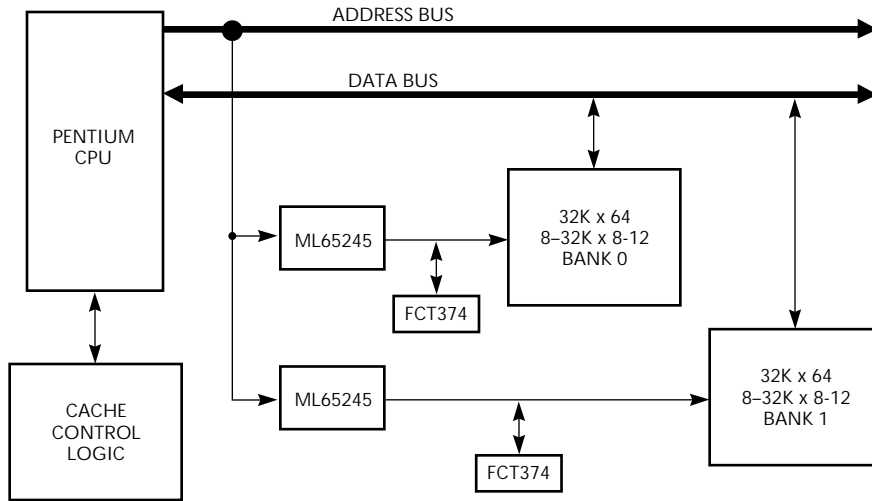


Figure 18. Pentium L2 Cache Block Diagram.

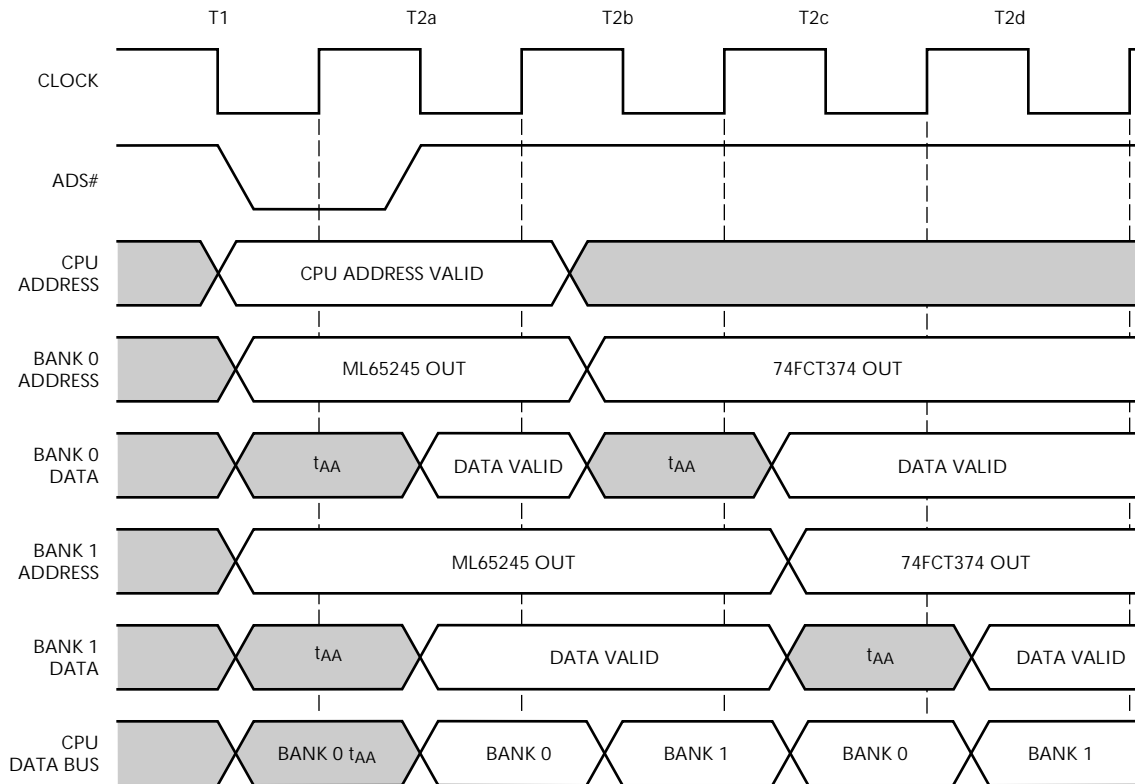


Figure 19. L2 Cache Burst Timing Diagram.

Table 6: Timing Margin for L2 Cache

SOURCE	VALUE, ns
Availabe Time: 2 clocks	+30
CPU Clock to Output	-8.0
CPU Loading Delay	-2.0
ML65245 Delay	-1.5
Trace Delay	-1.5
SRAM Access Delay	-12
CPU Data Setup Tome	-3.8
Total Delays	-28.8
Margin, t_{MARG}	+1.2

Although this design is suggestive rather than complete, it serves to indicate the possibilities of these devices. They allow you to consider designs that would otherwise be impossible, such as a zero wait state, 2111 cache for the Pentium using plain 32K x 8 SRAMs.

OTHER APPLICATIONS

The UFB can save 2.3 nanoseconds or more in bus designs. This feature strongly recommends the UFB to many designs where saving this time can mean performance improvement and design margin. Below is a list of just a few of these possible applications.

- DRAM Module Address Buffers
- Cache Module Address Buffers
- Big cache address buffers: 1 + MByte
- CPU Data Bus Transceivers
- PCI Bus Address/Data Buffers
- DSP: Address buffer for large SRAM
- ATE: probe buffers, pin drivers

DISCLAIMER

FAIRCHILD SEMICONDUCTOR RESERVES THE RIGHT TO MAKE CHANGES WITHOUT FURTHER NOTICE TO ANY PRODUCTS HEREIN TO IMPROVE RELIABILITY, FUNCTION OR DESIGN. FAIRCHILD DOES NOT ASSUME ANY LIABILITY ARISING OUT OF THE APPLICATION OR USE OF ANY PRODUCT OR CIRCUIT DESCRIBED HEREIN; NEITHER DOES IT CONVEY ANY LICENSE UNDER ITS PATENT RIGHTS, NOR THE RIGHTS OF OTHERS.

LIFE SUPPORT POLICY

FAIRCHILD'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF FAIRCHILD SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and (c) whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury of the user.
2. A critical component in any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.