

## Balancing RAM Access Time and Clock Rate Maximizes Microprocessor Throughput

---

Tuning timing relationships of high performance memories and fast buffer logic in microprocessor systems increases performance by eliminating unnecessary wait cycles

---

Stan Groves

Motorola Integrated Circuits Division  
3501 Ed Bluestein Ave, Austin, TX 78721

**T**hroughput and execution rate are of paramount importance in some systems. These systems require the most suitable microprocessor, running at the maximum usable clock rate, and the fastest available memories. More often, system cost also determines some, if not most, of the component parts used to build a system. Components are selected as the best compromise between performance and price. However, in quasi-synchronous systems, timing effects can interact so that it is not obvious just which of the various memory access times offers best performance, or whether the

system will benefit from use of high performance memories.

Although the MC68000 has an asynchronous data and address bus, in the sense that it can wait interminably for a response showing availability of requested data, the microprocessor illustrates a quasi-synchronous machine in the classical sense: internal operations are asserted and external signals are sensed at specific clock times. Owing to the internal synchronous nature of this microprocessor, all bus access times are in increments of one full clock period. It senses all input data and control lines when the clock is in

its high state and captures data or control line states when the clock goes low.

For example, as shown in the read cycle timing diagram of Fig 1, data acknowledge ( $\overline{DTACK}$ ) is asserted low prior to the falling edge of the fourth clock state (S4). As long as  $\overline{DTACK}$  is asserted a full setup time period prior to the falling edge of any clock signal, such as S4,  $\overline{DTACK}$  will be sensed during that clock period. If  $\overline{DTACK}$  is asserted low less than the required setup time prior to the falling edge of S4, a wait cycle of one full clock period, which equals two states, would be added. When  $\overline{DTACK}$  is sensed low at the end

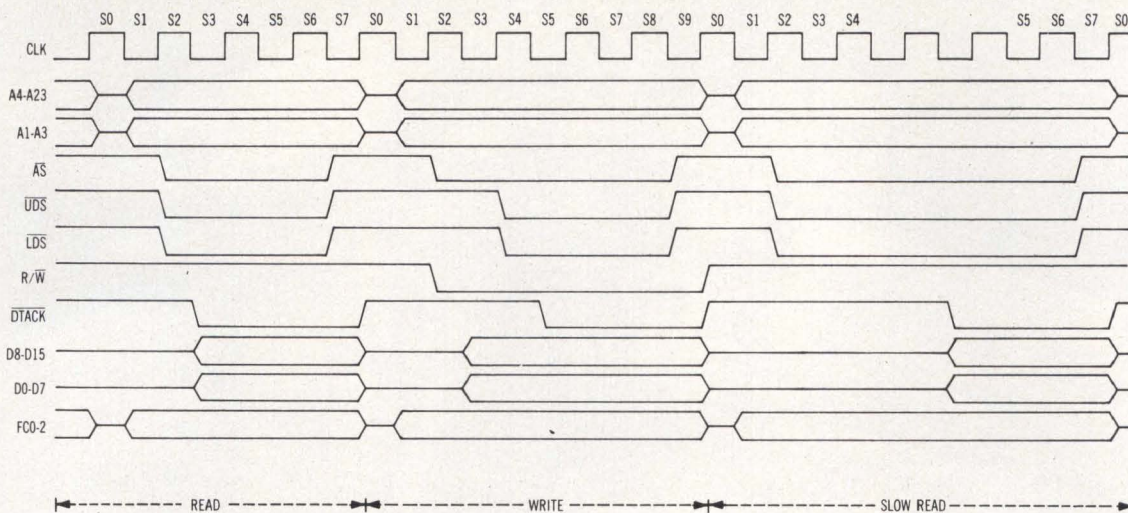


Fig 1 Read/write cycle timing diagram. As long as  $\overline{DTACK}$  is asserted one setup time period prior to falling clock edge S4, it will be sensed during that clock period. Otherwise, as shown on right side of diagram, wait cycles will be added until  $\overline{DTACK}$  is sensed

of a clock period, the state of the data bus is captured on the falling edge of the next full clock period (S6 in this instance) and only then applied to the microprocessor. As shown on the right side of Fig 1, wait cycles or wait states are added after S4 until  $\overline{DTACK}$  has been low for the full setup time prior to the falling edge of the clock signal.

Any system that uses even a small portion of the MC68000 addressing capability needs signal buffers. The delay through these buffers, the delay through the logic to generate row address select (RAS) for the random access memories (RAMs), the output delay of the address lines and address strobe ( $\overline{AS}$ ) signals, and the data port input

setup time must be considered overhead to the specified memory access time. Table 1 shows this additional overhead by listing RAM access times, typical values for the time delay through Schottky (S) and low power Schottky (LS) buffers, the critical path that generates RAS from  $\overline{AS}$ , and MC68000 data setup delay with  $\overline{AS}$  delay.

TABLE 1  
Access Time and Clock Rate Interrelationship

Specified RAM Access Times (ns)	Additional Overhead (ns)								Bus Latency Period Required (ns)	Operating Frequencies (MHz)	
	Buffers		AS RAS		MC68000 Delay (Data Setup and AS Delay)					Max, no waits	Nominal, no waits
	'S240	'LS240	'S32	'LS32	10 MHz*	8 MHz	6 MHz	4 MHz			
50	4 x 7		7		10 + 50				145	17	16
100	4 x 7		7		10 + 50				195	12.8	12
150	4 x 7		7		10 + 50				245	10.2	10
200	4 x 7		7			15 + 55			305	8.19	8
200	4 x 7	4 x 14		22		15 + 55			348	7.18	7
250	4 x 7		7			15 + 55			355	7.04	7
250		4 x 14		22			25 + 65		418	5.98	6
300		4 x 14		22			25 + 65		468	5.34	5
350		4 x 14		22				30 + 75	533	4.69	4
400		4 x 14		22				30 + 75	583	4.2	4
450		4 x 14		22				30 + 75	633	3.94	3.58

\*Projected

**TABLE 2**  
**Operation with LS Buffers and 200 ns RAMs**

Action	Clock Cycles	Time (ns)	Clock Frequency (MHz)	Performance
Instruction sequence (ideal)	17	2125	8	100%
If wait on each read (actual)	20	2500	8	85%
If reduced clock frequency	17	2429	7	87%
If only 1/2 wait cycle on each read	18.5	2313	8	92%

MC68000 delay times are from the latest data sheet showing 4-, 6-, and 8-MHz parameters with projected 10-MHz parameters. When worst-case numbers are used, the resulting bus latency period permits operation at the nominal clock rate shown in the table, provided that no wait states of one full clock period each are to be incurred.

Instruction cycle times from the MC68000 data sheet assume a nominal read cycle time of 4 clock periods, with 2½ periods allocated for bus latency, and a nominal write cycle time of 5 clock periods, with 3½ periods allocated for bus latency. When writing, ample time is available to use the less expensive LS buffers and logic. However, again referring to Table 1, to avoid incurring wait cycles in a typical system with 200-ns RAM and LS buffers, a clock frequency must be selected for which the required 348-ns latency period represents 2½ clock periods (about 7.18 MHz), or else Schottky logic must be used.

In a simple 2-instruction sequence—read data followed by write data—there are four bus accesses of which only one is a write access. This reflects a nominal time period of 17 clock cycles, or 2125 ns at 8 MHz. Table 2 shows the effect of changing the clock period and incurring wait cycles for a particular case, assuming 200-ns RAM in an 8-MHz system, where cost or other considerations require use of LS buffers. If a system uses 200-ns RAM with LS buffers at 8 MHz, reducing the clock frequency to 7 MHz would improve performance. This occurs because, when using these components at 8 MHz, full clock periods are added as wait cycles. Consequently, in the Table 2 example, 250-ns RAM with Schottky buffers could be used instead of the 200-ns RAM to achieve the same

20-cycle instruction sequence period of about 2500 ns.

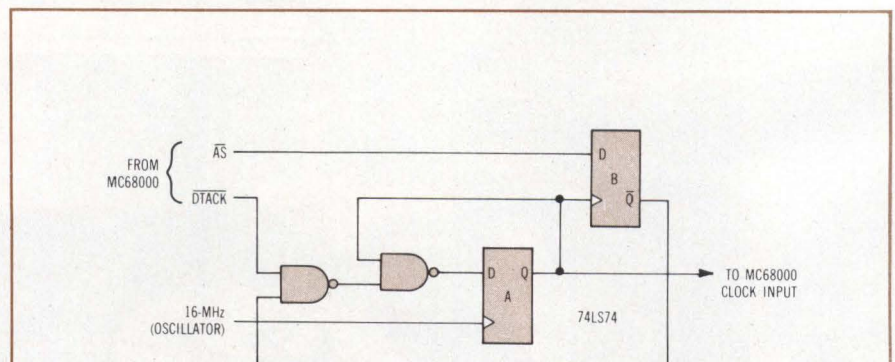
The last line of Table 2 describes operation if it were possible to add only one half of a cycle for each wait state. Using the previous example of a 2-instruction sequence requiring 17 cycles, the sequence now extends to only 18.5 cycles, instead of 20 cycles, when 200-ns RAM and LS buffers are used in an 8-MHz system.

Although the MC68000 extends bus cycles only in increments of one full clock period, the circuit shown in Fig 2 can be used to stretch S4 by unit periods of the oscillator input to flipflop A. This circuit will not stretch S2, because data strobes are not provided until S3 of a write cycle. Fig 3 clarifies the full impact of this approach by showing the combined interaction between memory access time with associated buffer logic type,

microprocessor clock frequency, and the number of wait cycles incurred.

In Fig 3, average execution time per instruction (in microseconds) of the read data, write data sequence appears on the left vertical axis. Lines sloping down and to the left reflect nominal microprocessor clock frequencies. Curves sloping down and to the right are labeled along the right axis according to memory access time (in nanoseconds) and buffer logic type. These curves include both the bus buffer overhead and the microprocessor overhead from Table 1. For each combination of logic type, memory access time, clock frequency, and number of wait cycles, Fig 3 gives the corresponding average instruction execution time.

The numbers in Table 2 were derived from Fig 3 and illustrate its use. For the typical delay parameters in Table 1, the 8.0-MHz clock line crosses the zero wait cycle between its intersection with the contour for Schottky-buffered 200-ns RAM and its intersection with the contour for 200-ns LS-buffered RAM, showing that Schottky-buffered RAM would incur no wait cycles and execute the two instructions in 2.12-μs total time, or 1.06-μs average time on the graph. Using 200-ns RAM with LS buffers incurs a single wait cycle for each access. The two instructions would execute in 2.5-μs total time for an average execution time of 1.25 μs each. However, if the clock stretching circuit of Fig 2



**Fig 2** Clock stretching circuit. Although MC68000 adds only full cycles as wait states, this circuit extends S4 by unit periods of flipflop A oscillator input, allowing half cycle wait state and thereby gaining one half of a cycle

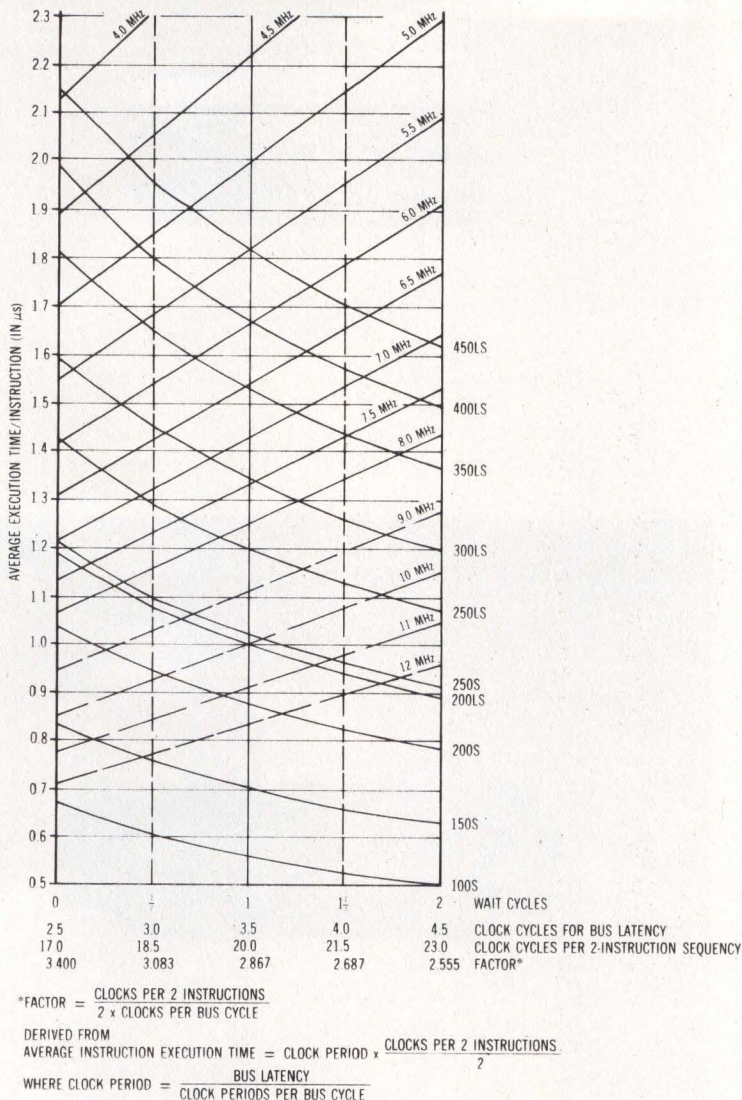


Fig 3 Composite performance chart. Intersecting contours give average instruction execution time for each combination of logic type, memory access time, clock frequency, and number of wait cycles. This diagram reflects specific logic delays of Table 1. For other delay parameters, product of factor on bottom line with total delay (memory access, buffer logic, and microprocessor) gives ordinate of contour sloping downward to the right

were used, only half a wait cycle would be incurred for 1.16- $\mu$ s average execution time.

As another example of the use of Fig 3, comparing a system with 250-ns LS-buffered memory operating at 6 MHz (1.42- $\mu$ s average instruction execution time) with a system using 300-ns LS-buffered memories operating at 6.41 MHz (1.44- $\mu$ s average instruction time) shows that both systems offer nearly equal performance—about the same

level of performance offered by 300-ns memories operating at 7.0 MHz (1.43- $\mu$ s instruction time). Consider a data communications controller with a proposed clock frequency of 9.8304 MHz ( $2^{10} \times 9600$  baud), which would require use of the 10-MHz MC68000. The clock cycle period is approximately 102 ns. If no wait states are incurred, the average simple instruction executes in 865 ns and the bus latency period is 254 ns. From Fig 3, Schottky-

buffered 150-ns RAM is recommended here. Schottky-buffered 200-ns RAM may be used with the clock stretching circuit of Fig 2 with the resulting average execution time of 941 ns offering nearly 92% of the performance obtained from the faster RAM. Similarly, 250-ns LS-buffered RAM can be used, incurring full wait cycles and an average execution time of 1017 ns, to achieve 85% of the performance offered by the 150-ns RAM.

Different circuit configurations result in different delay times reflected by the Table 1 data used to plot those contours in Fig 3 that slope downward to the right. Factors listed at the bottom of Fig 3 can be used to plot a composite performance chart for any set of delay times. Suppose, for example, the MC68000 system includes a memory management unit. Adding the memory management unit delay to the Table 1 timing values increases the bus latency period. For each combination of access time and logic type, the corresponding new bus latency period multiplied by each of the factors listed in Fig 3 identifies a new crossing on each of the Fig 3 axes.

In any system whose addressable memory even begins to approach the full capacity of the MC68000, the cost of memory far exceeds the cost of the microprocessor. Therefore, it is the microprocessor and its clock that should be tuned to the memories in use. The cost versus performance tradeoffs discussed here, with the composite performance chart of Fig 3 and the clock stretching circuit of Fig 2, determine which combination of logic type and memory access time offers best performance and allow adjustment of timing parameters to optimize performance of the components used.

**How valuable is this note to you?**

**High 716 Average 717 Low 718**

**Please circle the appropriate number in the "Comments" box on the Inquiry Card.**