1977

AN-775

Application Note

M6800 SYSTEMS
UTILIZING THE MC6875 CLOCK GENERATOR/DRIVER

Prepared by
Stephen R. Bookout
Microprocessor Applications/Systems Engineering

This application note describes the use of the MC6875 clock generator/driver in M6800 based systems. Design examples will demonstrate the capabilities of the driver in systems using slow and/or dynamic memories. Multiprocessing and DMA methods are also covered.

MOTOROLA *Semiconductor Products Inc.*

UTILIZING THE MC6875 CLOCK GENERATOR/DRIVER

INTRODUCTION

Previous methods of implementing MPU clocks ranged from discrete components to expensive hybrid schemes with hardware fixes for dynamic and slow memory handshaking.

The MC6875 is a monolithic MPU clock driver containing the dynamic and slow memory handshaking logic. A series resonant crystal with a center frequency of four times the desired MPU operating frequency is all that is required for most systems. For systems in which frequency stability is not critical, R-C networks may be used.

This application note describes the MC6875 clock generator/driver and illustrates its use in M6800 microprocessor systems. Included in the design examples are slow memory, dynamic memory and DMA (multiprocessor) examples.

PART DESCRIPTION

The MC6875 clock generator/driver is contained in a 16 pin dual-in-line package. The part uses Schottky devices to provide the high speed needed for fast rise and fall times and reduced propagation delays. Frequency is

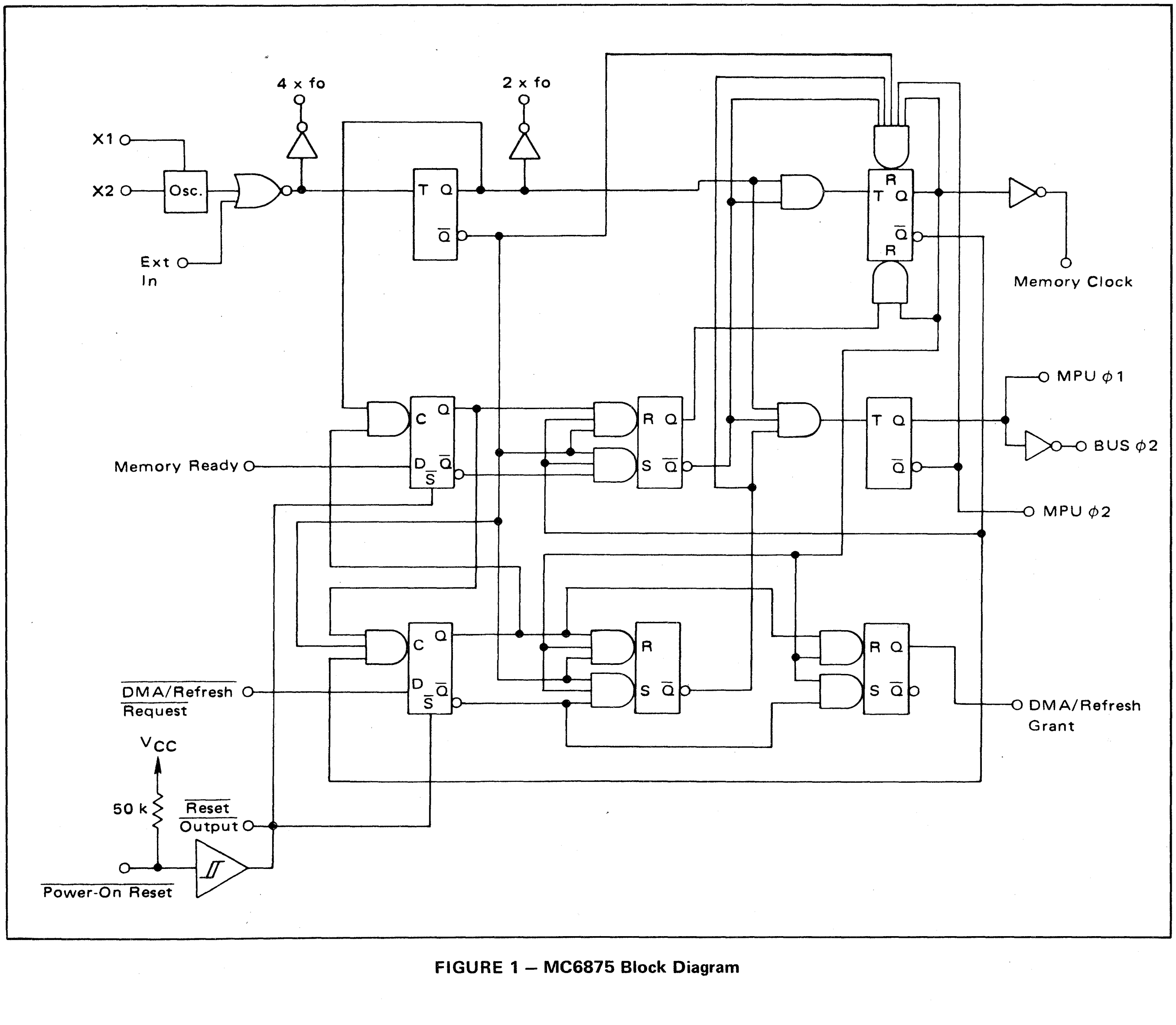


FIGURE 1 — MC6875 Block Diagram

Circuit diagrams external to Motorola products are included as a means of illustrating typical semiconductor applications; consequently, complete information sufficient for construction purposes is not necessarily given. The information in this Application Note has been carefully checked and is believed to be entirely reliable. However, no responsibility is assumed for inaccuracies. Furthermore, such information does not convey to the purchaser of the semiconductor devices described any license under the patent rights of Motorola Inc. or others.

controlled by a series resonant crystal or, alternatively, either an RC or LC network may be used. The resonant frequency of the oscillator is buffered and divided to produce the 4 x fo and 2 x fo outputs. The 2 x fo is then used to produce the Memory Clock, MPU $\phi 1$, MPU $\phi 2$ and Bus $\phi 2$. Memory Ready and $\overline{\text{Refresh Request}}$ inputs are internally sampled on alternate edges of 2 x fo enabling the stretch of either $\phi 1$ or $\phi 2$. All outputs are capable of driving high capacitance loads typical of unbuffered systems. The MPU $\phi 1$ and $\phi 2$ signal outputs provide the necessary V_{OH} (V_{CC} - 0.6 V) and V_{OL} (V_{SS} + 0.4 V) capable of driving two MPUs.

The functional block diagram of the internal logic of the MC6875 is illustrated in Figure 1.

SIGNAL DESCRIPTION

4 x fo, 2 x fo

A free running oscillator at four times (two times) the MPU's clock rate useful for a system sync signal.

$\overline{\text{DMA/Ref Req}}$

An asynchronous input used to freeze the MPU clocks in the $\phi 1$ high, $\phi 2$ low state for dynamic memory refresh or cycle steal DMA (Direct Memory Access).

DMA/Ref Grant

A synchronous output used to synchronize the refresh or DMA operation to the MPU.

Memory Ready

An asynchronous input used to freeze the MPU clocks in the $\phi 1$ low, $\phi 2$ high state for slow memory interface.

MPU $\phi 1$, MPU $\phi 2$

Capable of driving the $\phi 1$ and $\phi 2$ inputs on two MC6800s.

Bus $\phi 2$

An output nominally in phase with MPU $\phi 2$ having MC8T26 type drive capability which follows MPU $\phi 2$.

Memory Clock

An output nominally in phase with MPU $\phi 2$ which free runs during a refresh request cycle.

$\overline{\text{Power-On-Reset}}$

A Schmitt trigger input which controls $\overline{\text{Reset}}$. A capacitor to ground is required to set the desired time constant. Internal 50 kΩ resistor to V_{CC}.

$\overline{\text{Reset}}$

An output to the MPU and I/O devices.

X1, X2

Provision to attach a series resonant crystal or RC network.

Ext In

Allows driving by an external TTL signal to synchronize the MPU to an external system.

CAPABILITIES

Slow memory access, dynamic memory refresh and DMA are three areas in which the MC6875 can be used to manipulate and control the MPU timing.

Slow memory access is performed by stretching MPU $\phi 2$, Memory Clock and Bus $\phi 2$ in the high state while stretching MPU $\phi 1$ in the low state. Memory Ready is the control signal used to stretch these signals. Memory Ready is normally high and is active (low) only when addresses are valid to a slow memory or slow peripheral. It should be noted that at higher clock frequencies (above 1 MHz) many of the ROMs, RAMs and peripheral parts with access times sufficient for 1 MHz operation will be classified as Slow Parts needing this interface. The timing relationships are given in Figure 2. Memory Ready is sampled internally on the falling edge of 2 x fo. To stretch $\phi 2$, Memory Ready must be in its low state within the required minimum setup time, and held low for the minimum hold time, with respect to the falling edge of 2 x fo corresponding to the leading of $\phi 2$ (see MC6875 Data Sheet). Returning Memory Ready to its high state prior to the minimum high setup time referenced to a falling edge of 2 x fo will result in terminating the stretch on the following falling edge of 2 x fo.

A method of generating Memory Ready is illustrated in Figure 3. $\overline{\text{CS}}$ is an active low signal developed from the address decode including VMA used to enable the RAM array. This signal is inverted (CS) and used to hold

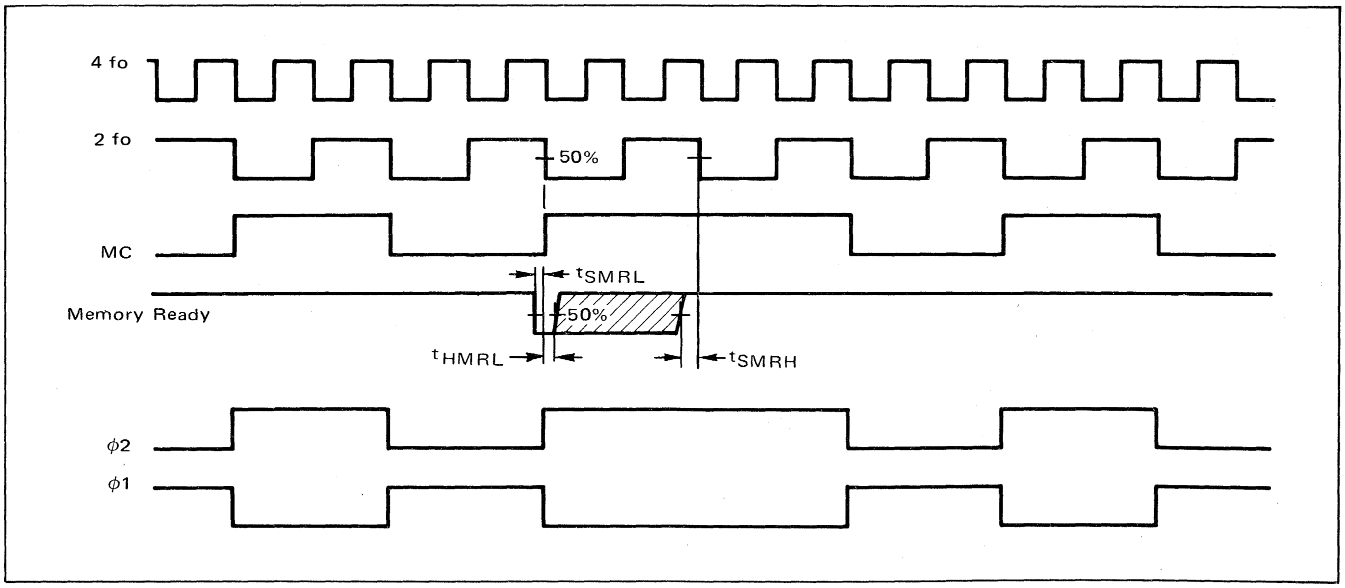


FIGURE 2 — Slow Memory Timing

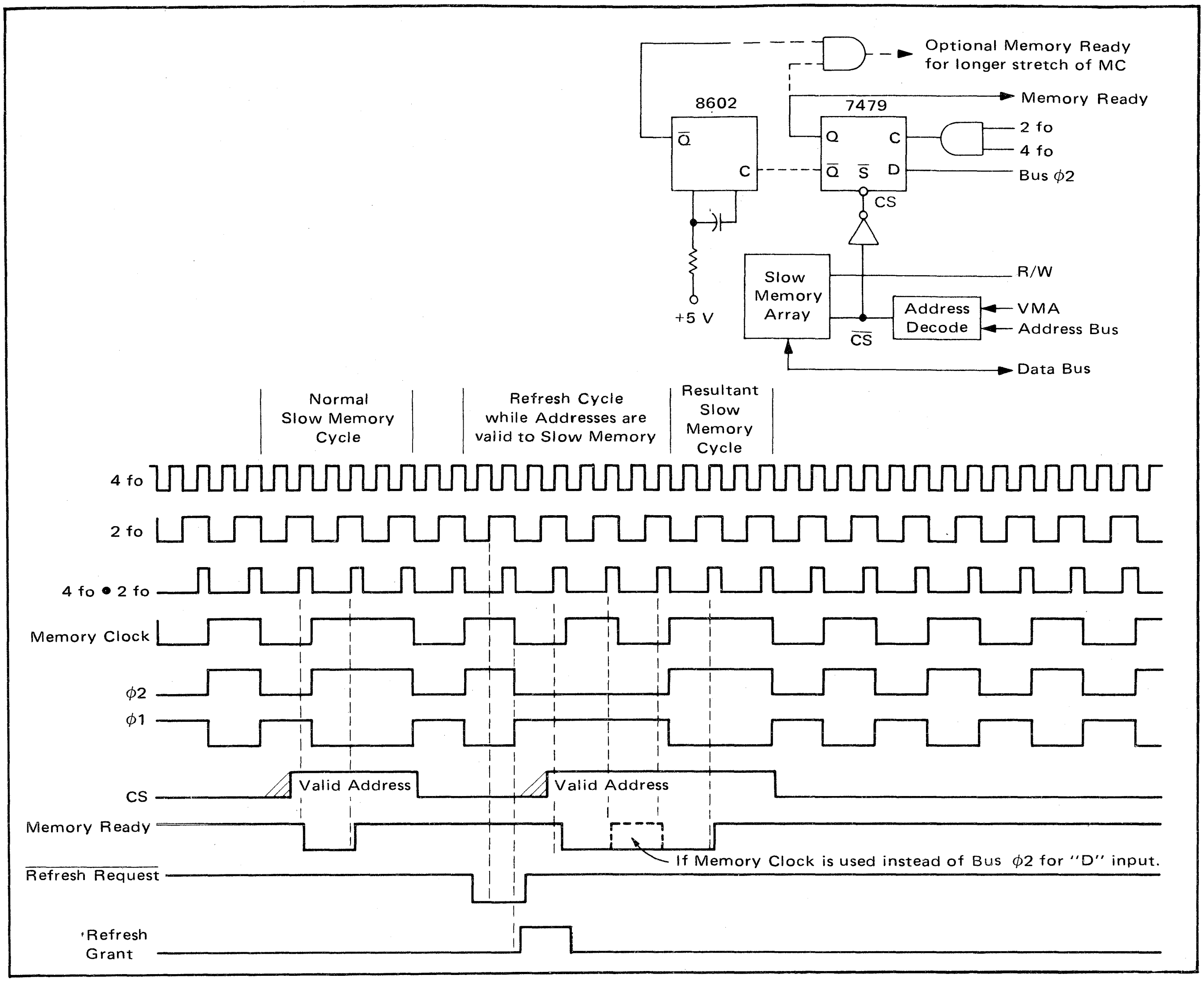


FIGURE 3 – Generation of Memory Ready

Memory Ready at a logic "1" when addresses are not valid to the slow memory. When addresses are valid the CS signal releases the $\overline{S}$ input to the MC7479 D flip-flop and allows 2 x fo ANDed 4 x fo to clock the present value of Bus $\phi2$ on to the Memory Ready line. This scheme will stretch $\phi2$ high for an additional 1/2 MPU cycle. If additional access time is needed a one shot may be added as indicated by the dotted lines.

Dynamic Memory refresh can be done by cycle stealing using the Refresh Request and Refresh Grant functions of the MC6875. The clock generator will stretch $\phi1$ in the high state and $\phi2$ in the low state allowing a refresh cycle to occur within the $\phi1$ time. Figure 4 illustrates the Refresh Request and Refresh Grant timing requirements for the MC6875. Refresh Request is internally sampled on the positive or leading edge of 2 x fo. To be recognized Refresh Request must be an active (low) prior to the minimum setup time and held low for the minimum hold time, referenced to the leading edge of 2 x fo occurring during the high portion of Memory Clock. If this is performed $\phi1$ will be stretched for a total of 1-1/2 MPU cycles providing Refresh Request is returned to a "1" level prior to the minimum setup time preceding the next leading edge of 2 x fo. Since Refresh Request is an asynchronous signal, Refresh Grant is provided by the MC6875 to indicate to the board requesting refresh that the request has been recognized. Thus the inactive edge of the Refresh Request signal can occur synchronously with Memory Clock. In Figure 5 Refresh Request is generated by clocking a D flip-flop with a refresh clock whose period is the required refresh rate. Refresh Grant is returned from the MC6875 to clock another D flip-flop which enables the Request flip-flop to be reset when the negative (leading) edge of the Row Address Strobe (RAS) occurs. The reset is disabled when the next leading edge of Memory Clock is encountered. Figure 6 illustrates the timing relationship of these signals.

The three basic methods of doing DMA include cycle stealing, multiplexing and halting the processor. Cycle stealing is done in the same manner as dynamic memory refresh. Refresh Request and Refresh Grant become DMA Request and DMA Grant. When performing DMA by cycle stealing it is important to observe the maximum stretch time the MPU can tolerate. Figure 7 illustrates the timing of DMA transfers by cycle stealing. It should be noted that the DMA controllers must provide the control signals R/W and VMA as well as the address and data lines. If the DMA Bus interface is wire ORed with the MPU Bus, the

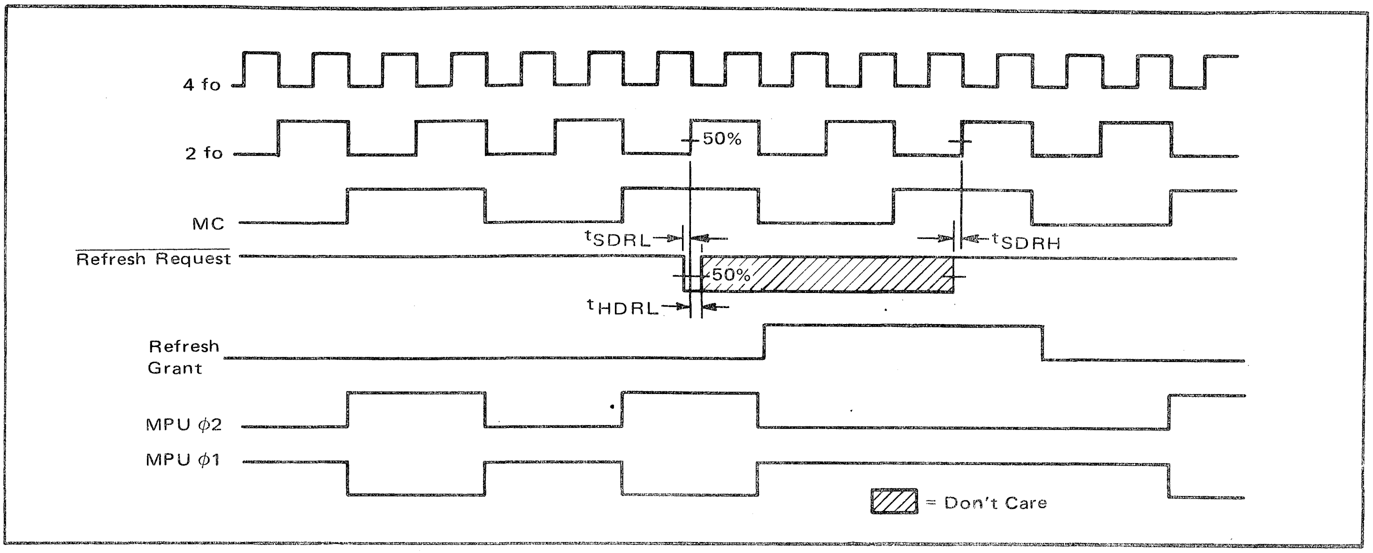


FIGURE 4 — $\overline{\text{Refresh Request}}$/Refresh Grant Timing

MPU control pin (TSC) can be used to force the MPU Bus drivers to high impedence state. Another alternative would be to use a Bus Switch such as the MC3449.

Multiplexed DMA results in the highest DMA transfer rate since the DMA operation is invisible to the MPU. Multiplexed DMA timing is given in Figure 8. This method requires memory access times fast enough to allow a complete read or write cycle to occur within 1/2 MPU cycle. A sample dual processor design given later will illustrate this technique.

DMA by halting the processor is illustrated in the timing diagram of Figure 9. In this mode the MPU may be halted as long as necessary to perform the DMA as the MPU clock signals are not stretched. This technique is useful when Burst DMA is desired. The DMA controller must provide the halt signal (active low) to the MPU. The MPU will finish the current instruction and respond with a positive transition of BA (Bus available) when the instruction is finished. The DMA controller may then take control of the Bus for transfer.

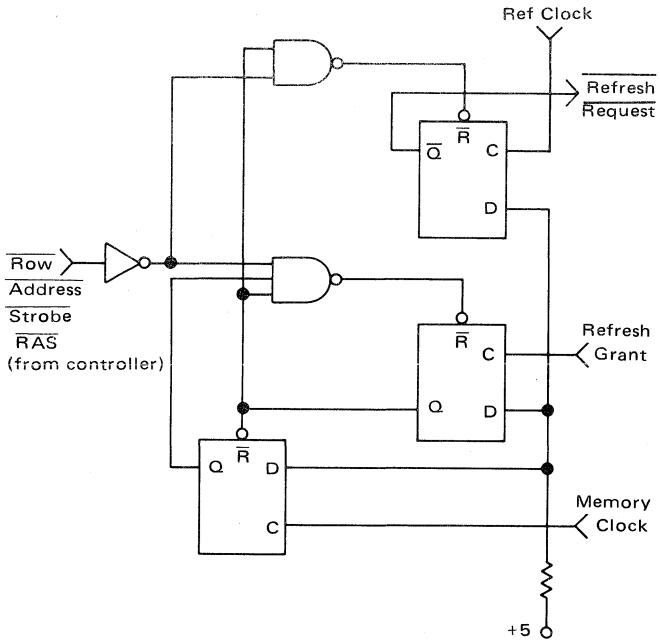


FIGURE 5 — $\overline{\text{Refresh Request}}$ Generation

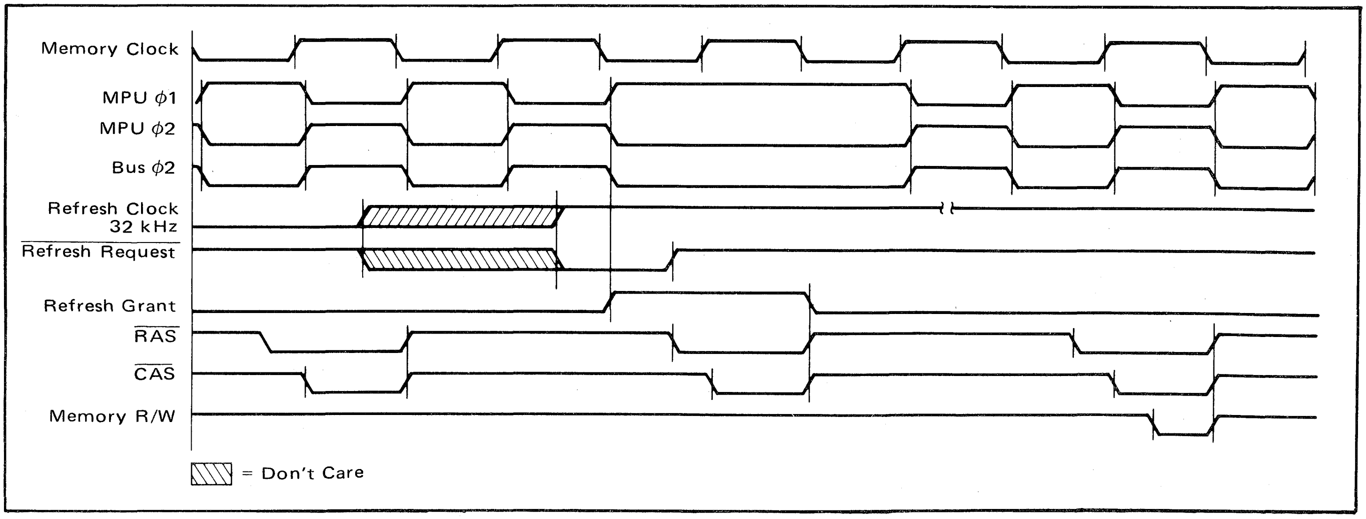


FIGURE 6 — Timing Relationships

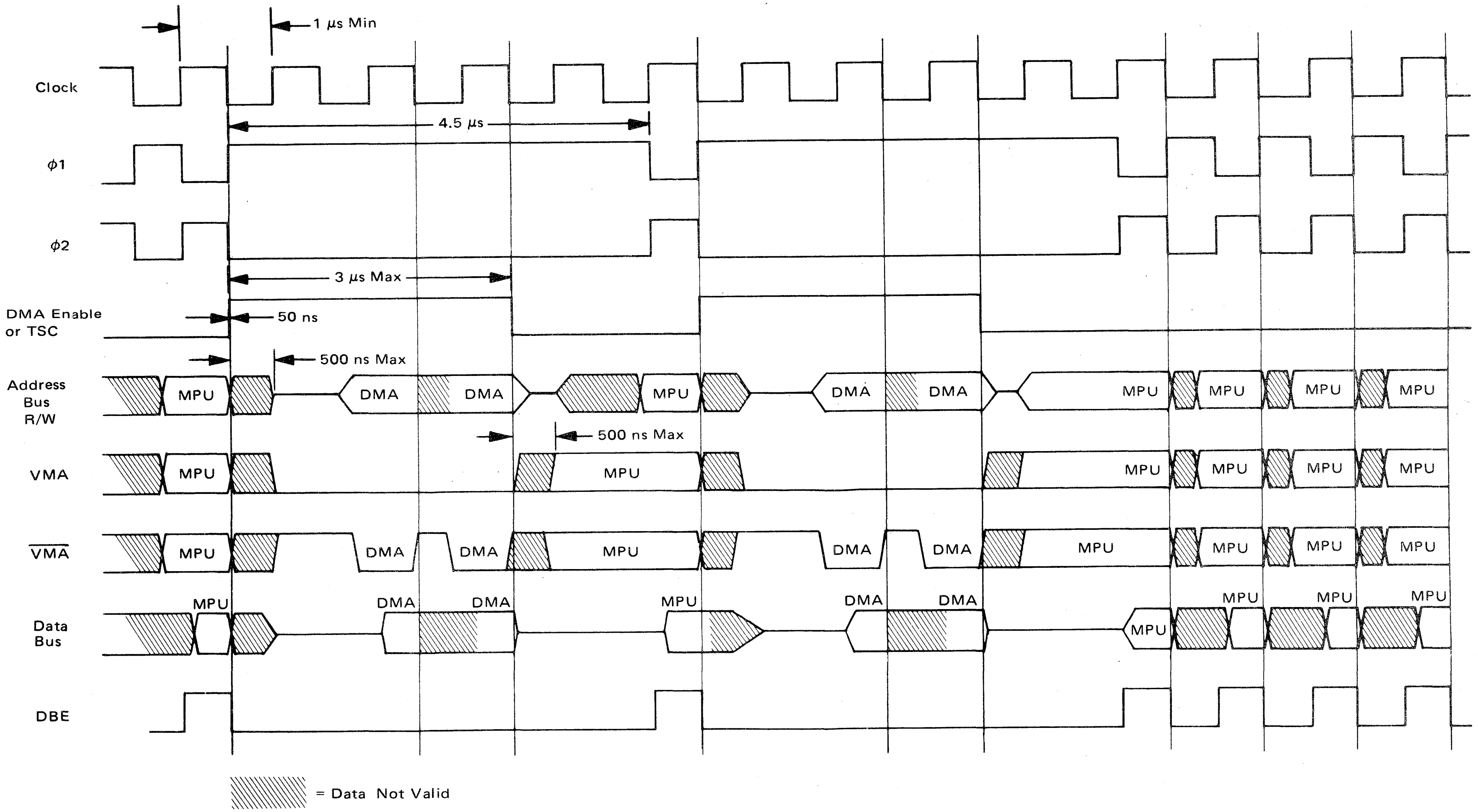


FIGURE 7 — Timing of DMA Transfers by Cycle Stealing

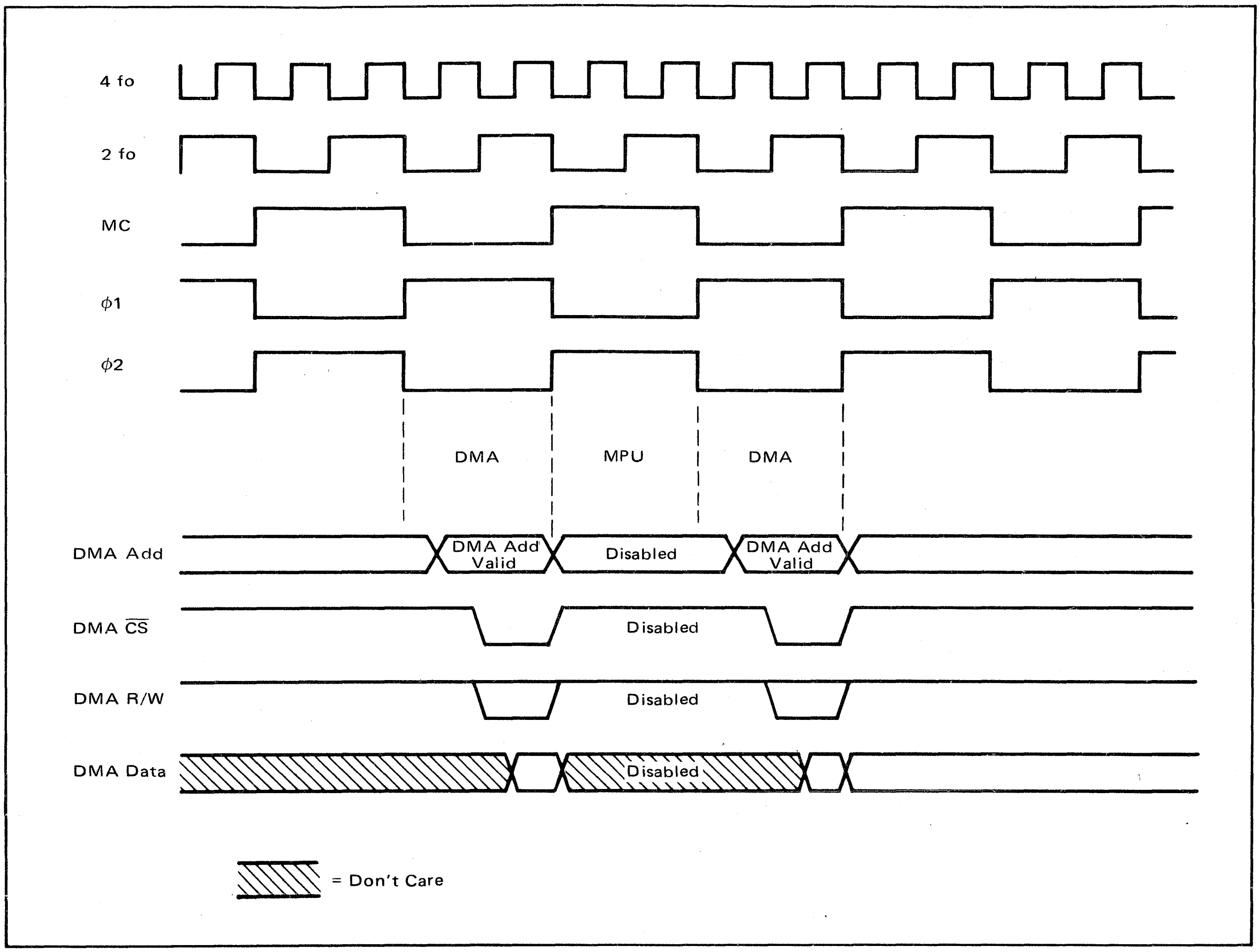


FIGURE 8 — DMA Multiplexed

DESIGN EXAMPLES

Typical Buffered System

In the block diagram of Figure 10 the MC6875 is connected in a typical buffered MPU system. The MC6875 should be located such that the signal path is less than two inches for the MPU ϕ1 and ϕ2. The damping resistors shown at the MPU ϕ1 and ϕ2 inputs should be located as close to the MPU inputs as possible. The value can range from 0 to 30 ohms but the optimum value range is 10 to 20 ohms. These resistors damp the overshoot and ringing typically found in these systems. They also extend the rise and fall times and reduce non-overlap time of the ϕ1 and ϕ2 signals. The block labelled DBE stretch is an optional circuit used with memory peripheral parts requiring longer data hold times. DBE stretch circuits are given in Figures 11a and 11b. These are basic stretch schemes and may be modified to suit the hold-time requirements. DBE may be stretched up to the maximum time allowed by the MPU specification used.

The block labelled Bus Control Logic of Figure 10 is expanded in Figure 12. As shown, this logic controls receiver/driver sections of the MC8T26. The Read Enable (Receiver) is an active low signal enabled only when R/$\overline{W}$ is high and Bus ϕ2 is high (Read Cycle). The write Enable (Driver) is an active high signal enabled only when R/$\overline{W}$ is low, MPU DBE is high and the MPU is not in the halt mode ($\overline{BA}$ high). MPU DBE is used to provide the data hold times required by some memories such as the early 2102.

2048 x 8 Bit Slow Memory Design Using Silicon Gate MOS 2102

The very first step in designing a memory system is to develop a timing diagram showing the relationship of the MPU timing and the required memory timing for both the Read and Write cycles. Once this is done the designer can easily develop the controller which consists of address decode, R/$\overline{W}$ and various strobe signals. Figure 13 illustrates these relationships. The memory $\overline{CE}$ can be presented to the memory array as soon as the MPU addresses are valid. As described earlier, CE should be used to develop the memory ready signal stretching ϕ2 1.5 μs. R/$\overline{W}$ to the memory array must wait at least 200 ns after $\overline{CE}$ for a write cycle. This can be accomplished by incorporating Bus ϕ2 into the Memory R/$\overline{W}$ signal. See schematic of Figure 14.

16K x 8 Dynamic Memory Using MCM6604

Dynamic memory system design has been complicated due to the involved controller. The controller has to

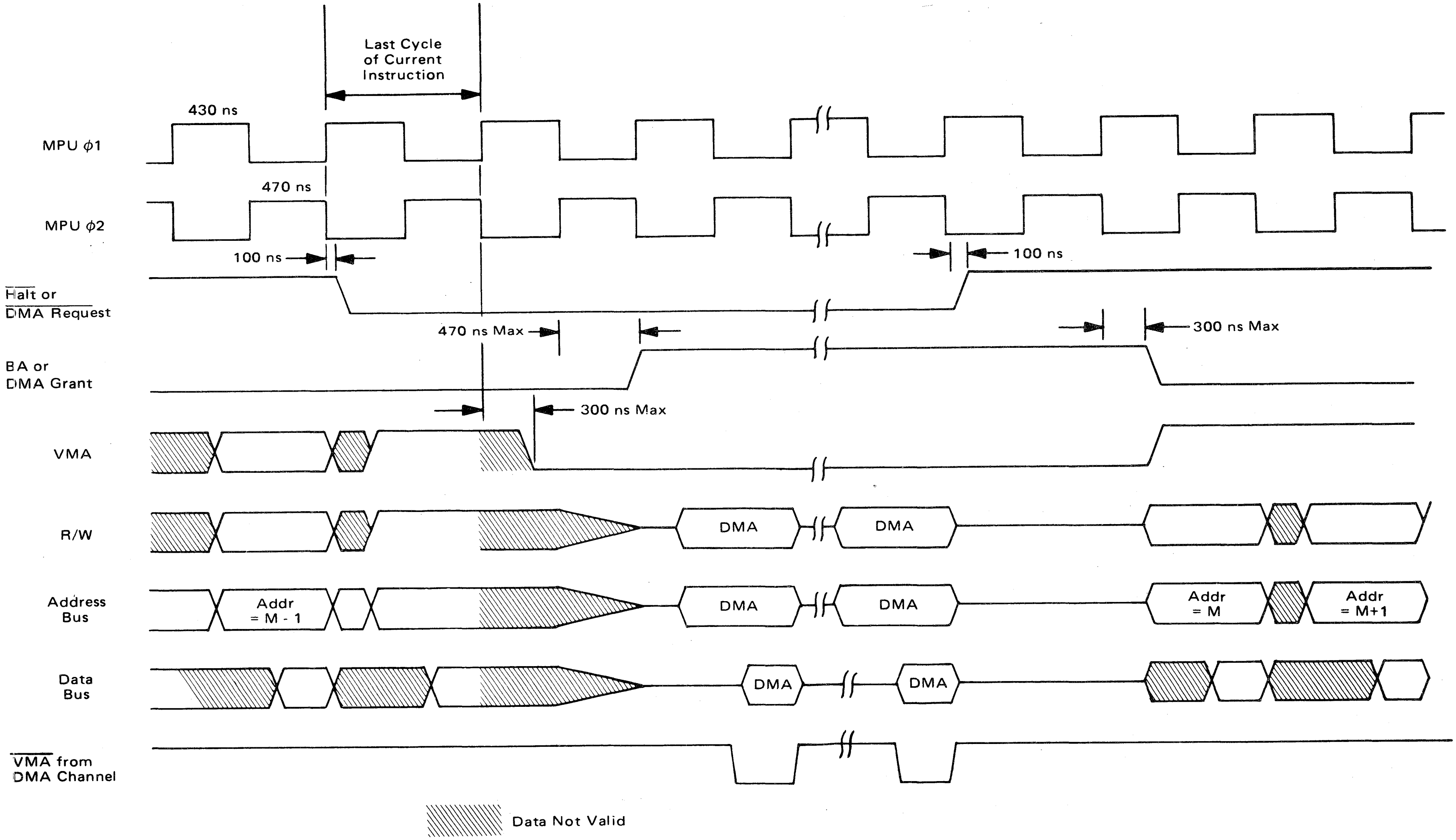


FIGURE 9 – Timing of DMA Transfers by Halting the Microprocessor

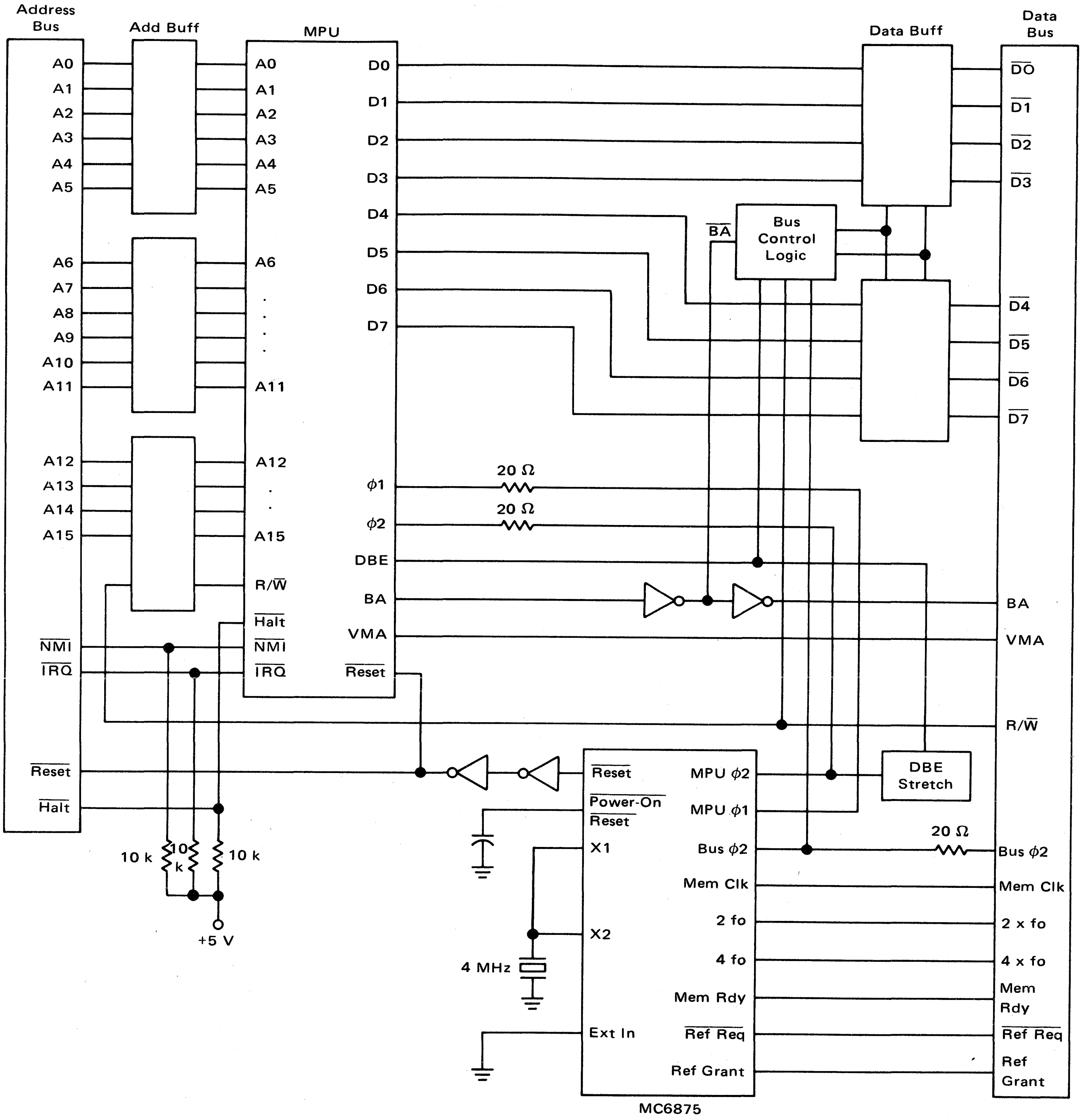


FIGURE 10 — Block Diagram

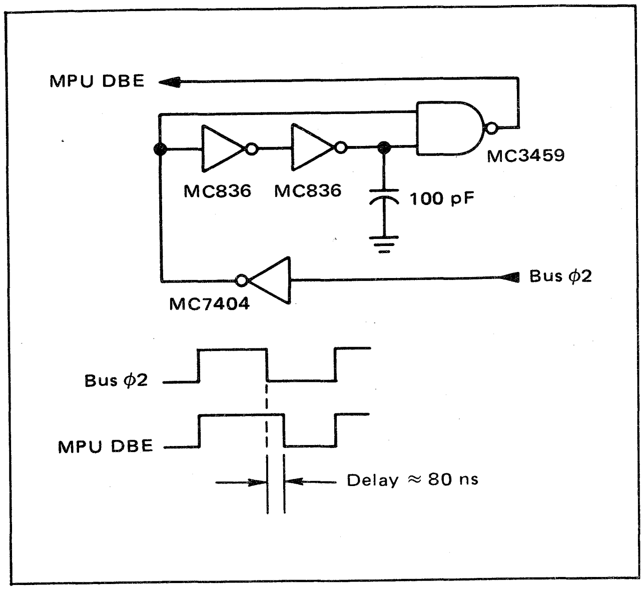


FIGURE 11a – DBE Stretch Circuit (Half Shot Method)

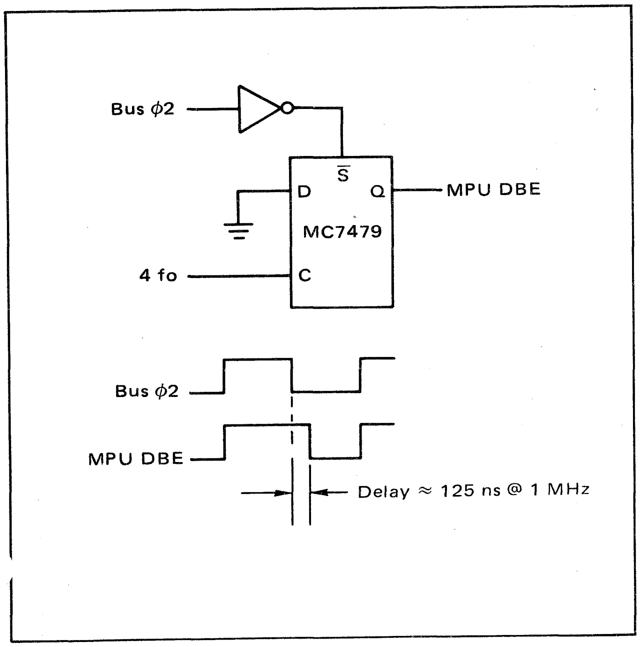


FIGURE 11b – DBE Stretch Circuit (Flip-Flop Method)

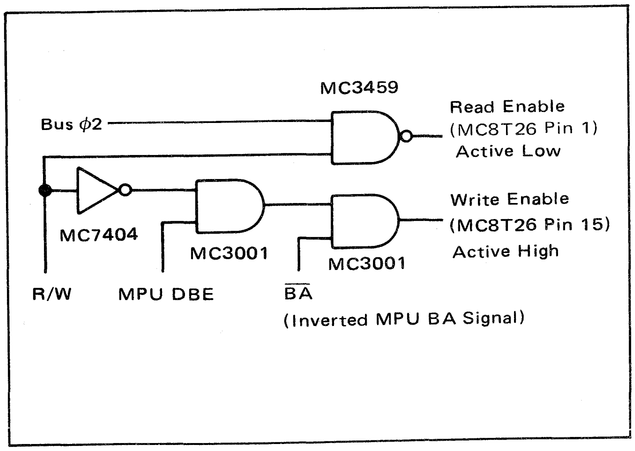


FIGURE 12 – Bus Control Logic

provide refresh, R/W and Row and column address strobes. The MC3480, in conjunction with the MC3232A and MC6875 handles all of this with ease. The block diagram in Figure 15 describes a typical 16K system employing the MC3480. The MC3480 is intended for use with the MC3232A (Address multiplexer and refresh counter) and the MCM6604 4K dynamic RAM. The delay circuit may be manipulated to configure the output timing for other memory types.

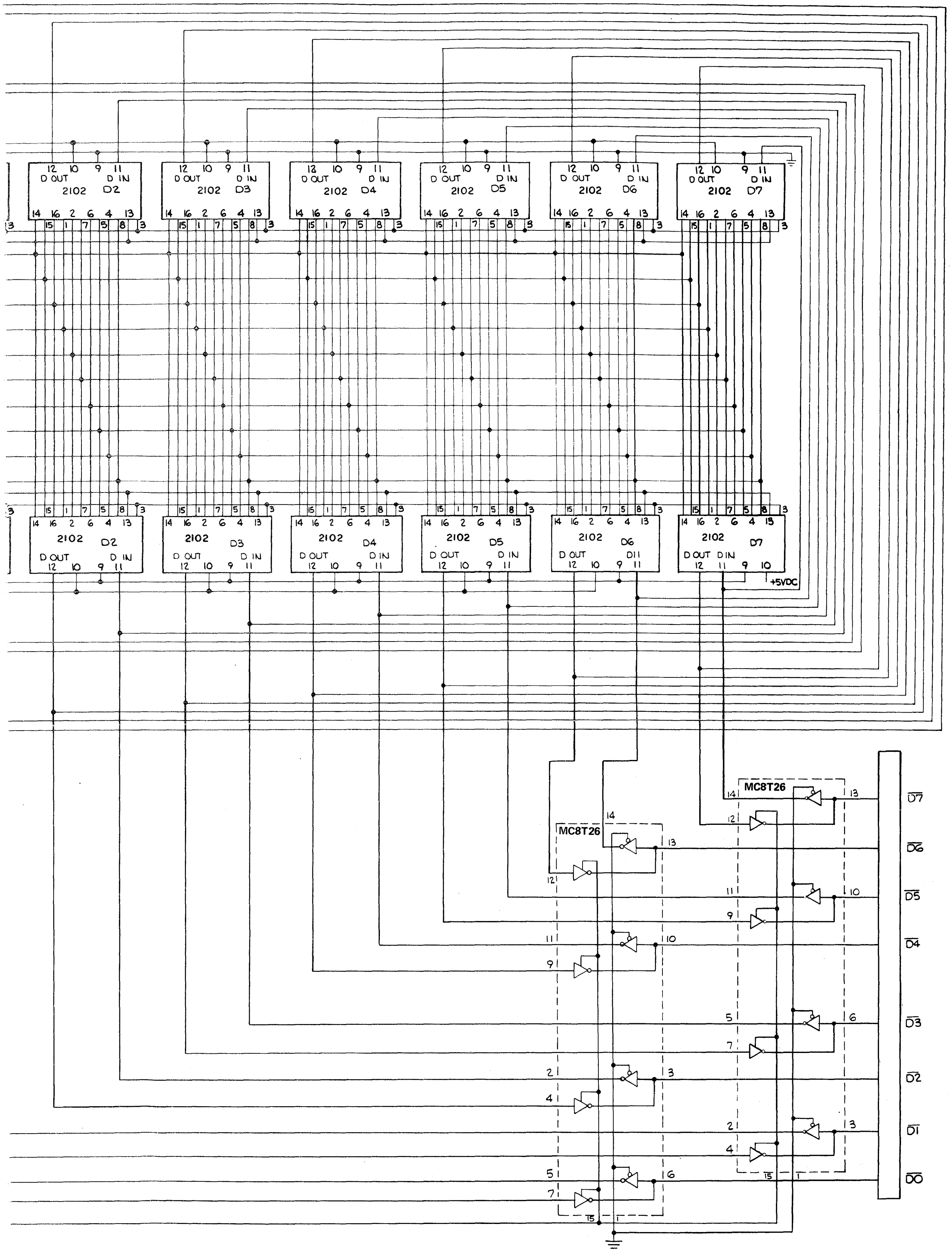
Figure 16 is the timing diagram for the MPU, MC3480 and MCM6604. The time delayed inputs, t1 through t5, may be generated by delay lines, one shots, counters or combinational logic, depending on the speed of operation and signals available. One method of generating these delays employing a shift register is shown in Figure 17. The shift register is clocked on the positive edge of 4 x fo and is pre-set on the logical AND of 4 x fo, 2 x fo and fo. Figure 18 uses combinational logic to develop the delayed inputs to the MC3480. With this technique the MC6875 must run slower (3.32 MHz Crystal) to align the 4 x fo and 2 x fo signals with the required time delays.

Multiplexed Dual Processor System

Several methods exist for designing dual or multi-MPU systems. The multiplexed scheme is the most popular due to its high processing rate. The block diagram of Figure 19 illustrates the technique of swapping the MPU ϕ1 and ϕ2 signals and multiplexing operations on a common memory array. The total system in actuality would contain three buses. One bus for each MPU containing ROM, RAM and I/O and the third bus containing the common RAM. As stated earlier the access time of the RAM and/or I/O used on the common bus will determine the clock period. The same techniques used here for accessing the common area would also be used for DMA access. Figures numbered 20 and 21 indicate the buffered dual processor board and common memory board. Not shown are the main buses containing the required ROM, RAM and I/O. The MPU board is straightforward; note that the MC6875 is directly driving both MPUs. In Figure 21, MC3449s are controlling the address and data bus. ϕ2 controls the selection of Bus 1 or Bus 2. When ϕ2 is low, Bus 2 is selected and when ϕ2 is high Bus 1 is selected. C̄S̄ is then developed by decoding address bits A8 through A15. C̄S̄ is then used as a chip select to memory and as an enable to the MC3449s controlling the data bus entry. The R/W̄ line is then used to control the direction of the Data Bus. Because of the timing requirement of the MCM6810AL1 and propagation delay of the MC3449s along with the Address and Data buffers, the MC6875 must run slightly slower (approximately 840 kHz). This technique can be expanded by adding other MPUs sharing common memory with the first MPU. Thus MPU1 becomes a master and the other MPUs are slaves sharing common memory only with the master. (Note: All addresses must be unique.)

Dual Processor Using Halt

This technique, as described earlier, is illustrated in the block diagram of Figure 22. For simplicity, it is shown with MPU1 as a master with access to Bus 2 as well as Bus 1. In a user system both MPUs may have access to either bus. In this application access is gained to the second bus



ow Memory Using 2102's

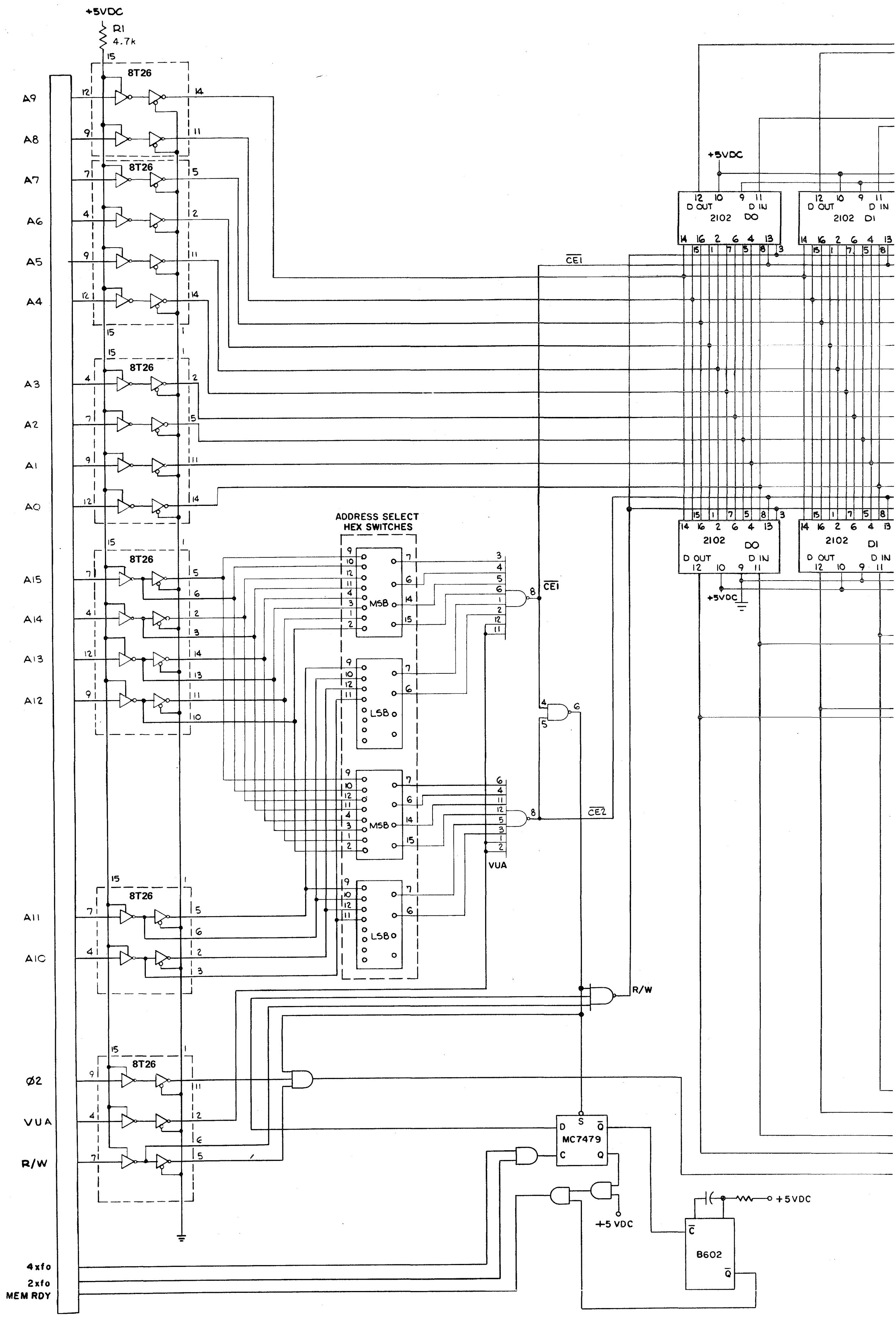


FIGURE 14 — 2K S

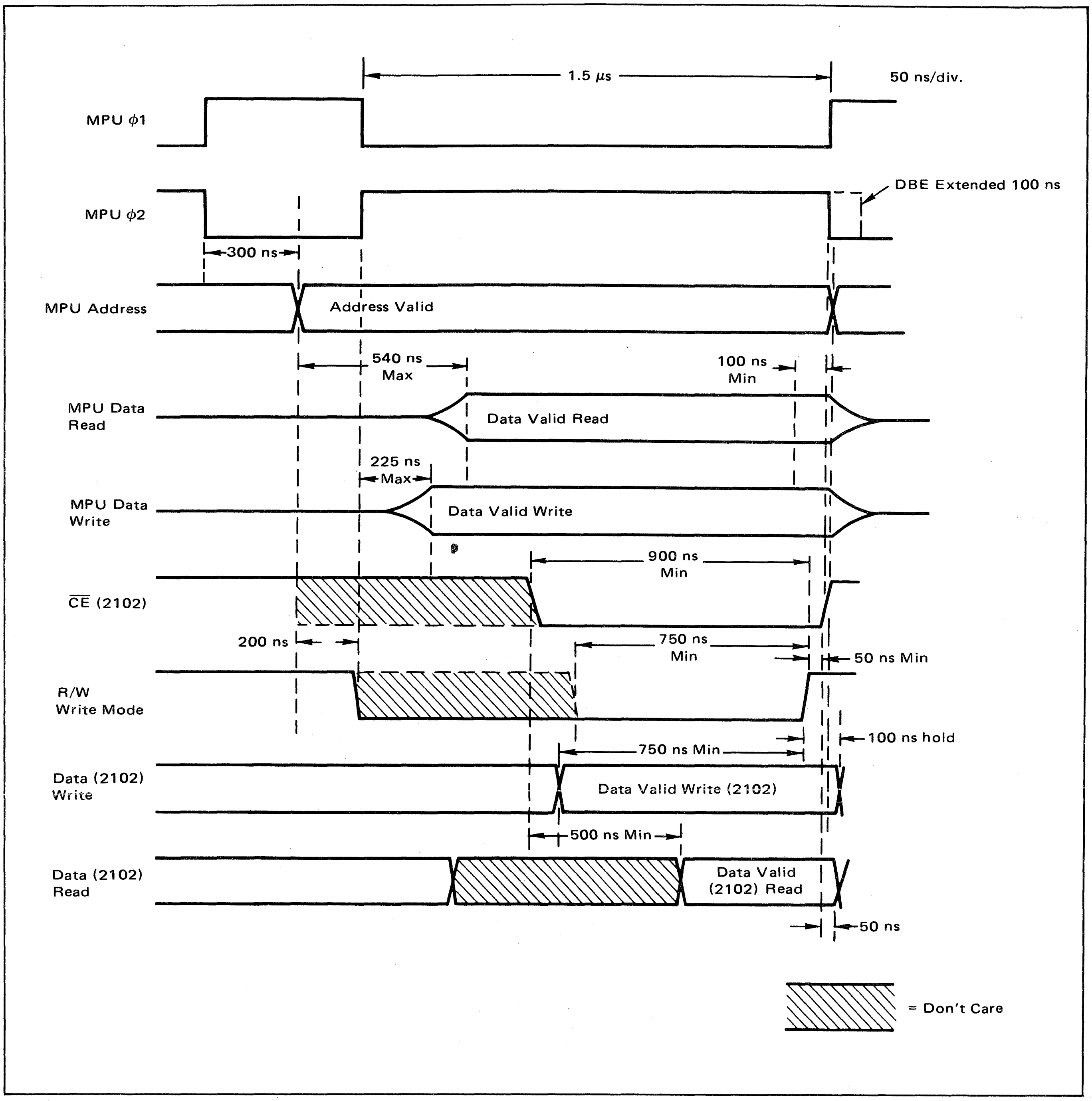


FIGURE 13 — 2102 Timing

by halting MPU2 using a PIA on Bus 1. When MPU2 finishes the current instruction BA will return high switching control of Bus 2 to MPU1 and at the same time indicating to MPU1 that the bus is available. This technique can be used in conjunction with the multiplexed scheme in multiprocessing applications; however, NMOS drivers may have to be used to expand the drive capability of the MC6875.

Design And Layout Considerations

Certain precautions must be taken when designing an MC6875 into an M6800 system. It is recommended that:

1. The MC6875 be located such that the MPU ϕ1 and ϕ2 signal paths are within 2" of the MPU.
2. Damping resistors within the ranges of 10 to 30 ohms should be located as close as possible to the MPU ϕ1 and ϕ2 input pins of the M6800.
3. $\overline{\text{Refresh Request}}$ and Memory Ready be pulled up when not in use.
4. Crystal, RC or L-C controlling networks be located as close as possible to the corresponding inputs of the MC6875.
5. Ground loops be avoided and high frequency bypass capacitor used directly at the MC6875. (0.1 μF ceramic disk)
6. The External Input be grounded if not being used.

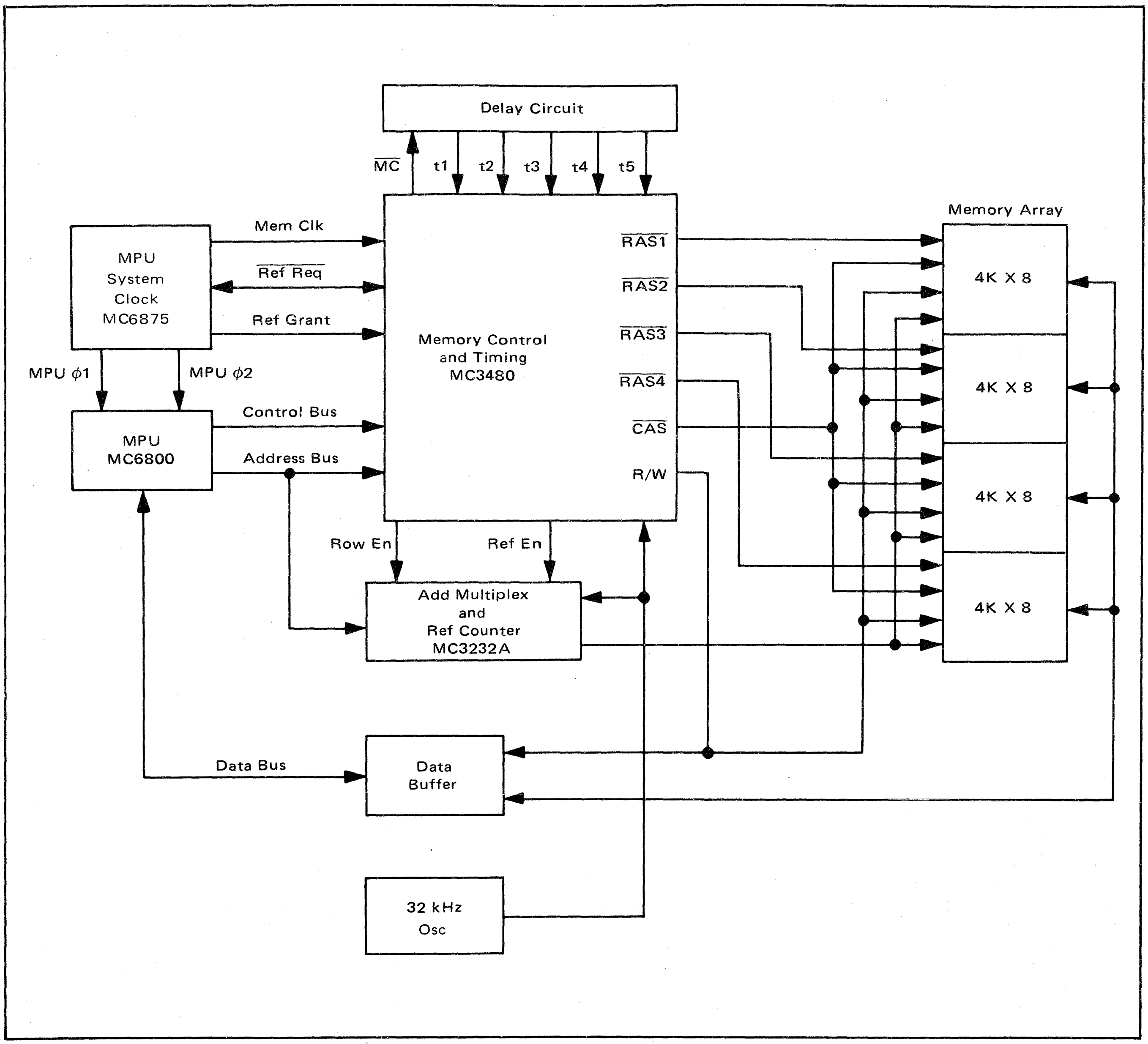


FIGURE 15 – 16K X 8 Memory System Employing MCM6604 (4K RAM)

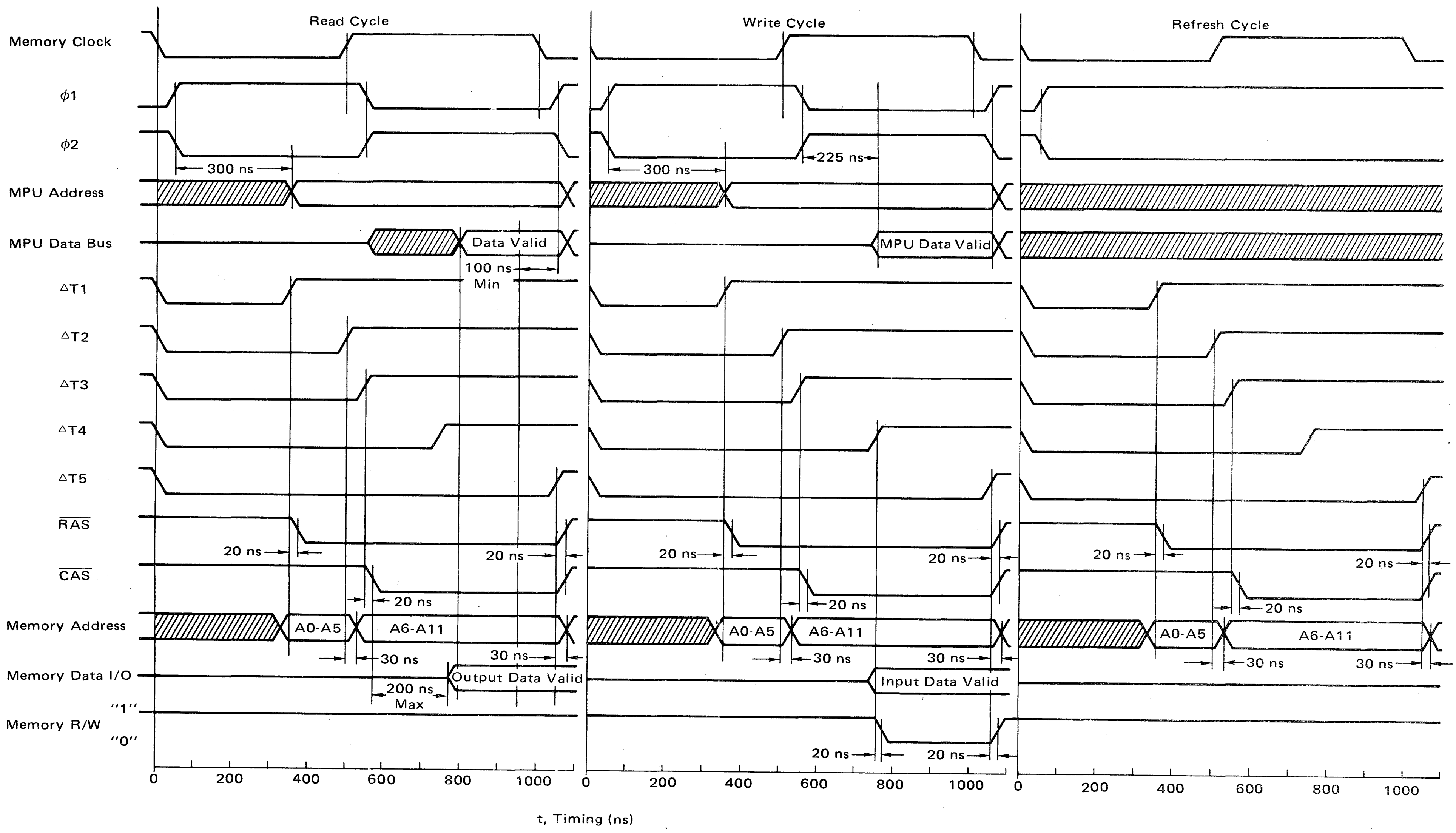
7. If Dynamic Memory is used the Reset output should be buffered and the resulting Reset be ORed with a debounced Master Reset signal. This is needed since the Power-On-Reset input will disable the dynamic memory refresh.

8. TTL and NMOS loads should not exceed the maximum capability of the MC6875.

9. Crystals be selected with equivalent series resistance of 35 to 60 ohms and that can tolerate a circuit load capacitance of 12.5 to 19 pF. These crystals may be purchased from Tyco or CTS Knights Inc.

CONCLUSION

The MC6875 is a very versatile, reliable and inexpensive clock. It can be tailored to the users' system with a minimum of handshaking logic. As shown earlier, the 2 x fo and 4 x fo outputs are useful in developing various control signals needed with certain memory and I/O parts. The high drive capability make it useful in buffered, unbuffered and dual MPU systems. The clock stretching capability make it useful in Dynamic Memory, slow memory and DMA applications.


Read Cycle
Write Cycle
Refresh Cycle
Memory Clock
ϕ1
ϕ2
MPU Address
MPU Data Bus
ΔT1
ΔT2
ΔT3
ΔT4
ΔT5
RAS
CAS
Memory Address
Memory Data I/O
Memory R/W
"1"
"0"
300 ns
300 ns
225 ns
Data Valid
MPU Data Valid
100 ns
Min
20 ns
20 ns
30 ns
200 ns
Max
A0-A5
A6-A11
Output Data Valid
Input Data Valid
0
200
400
600
800
1000
t, Timing (ns)

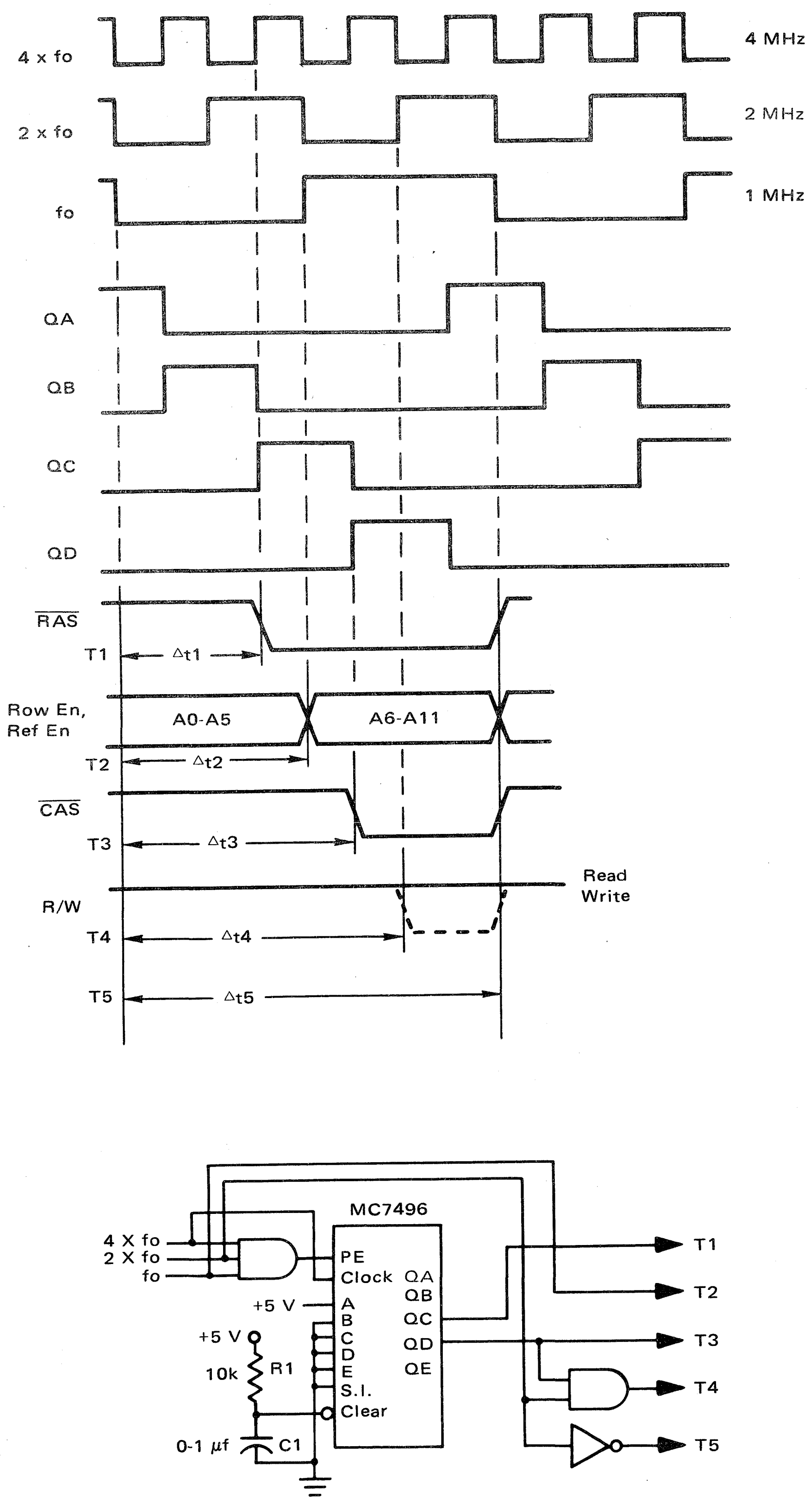


FIGURE 17 — Shift Register Delay Scheme

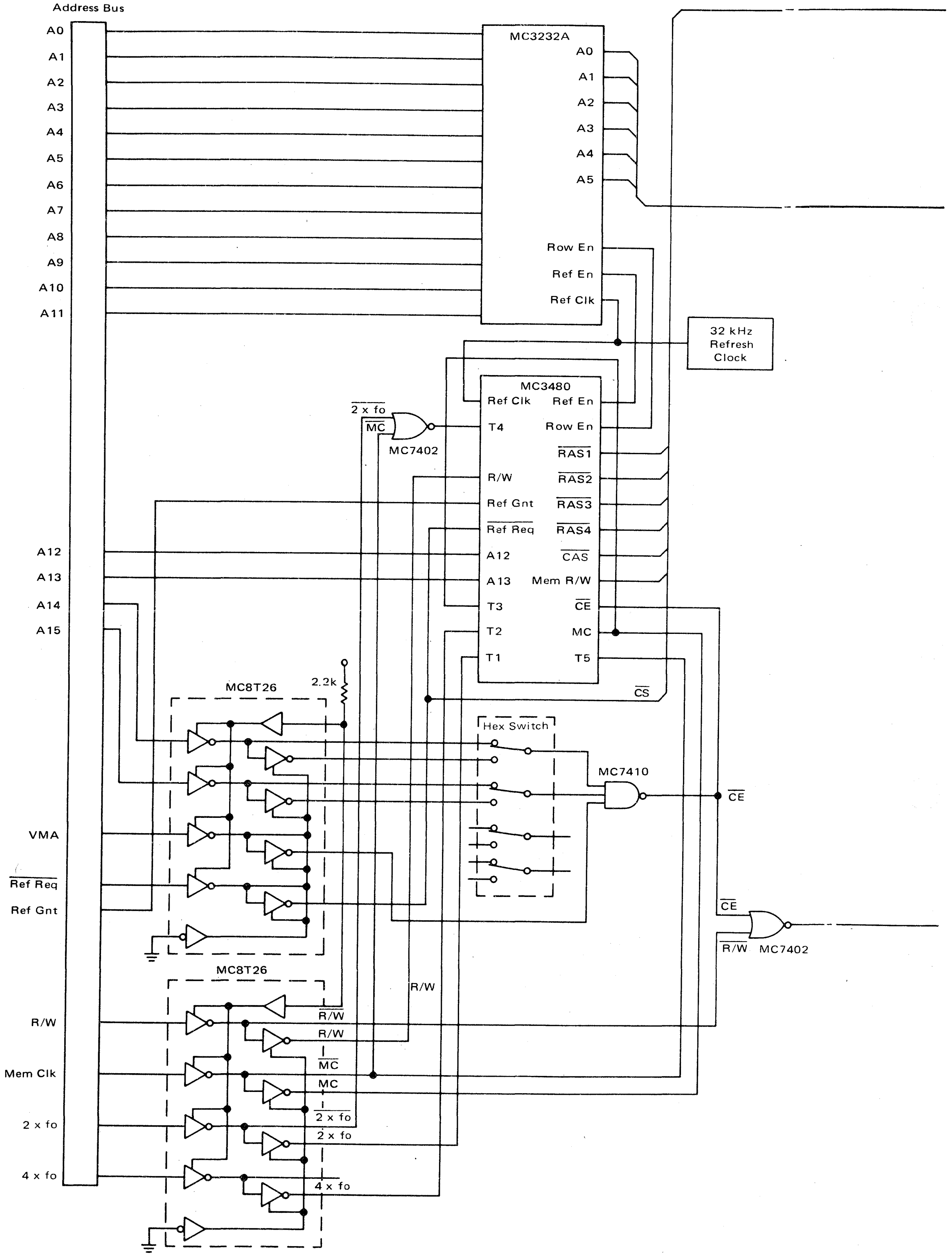
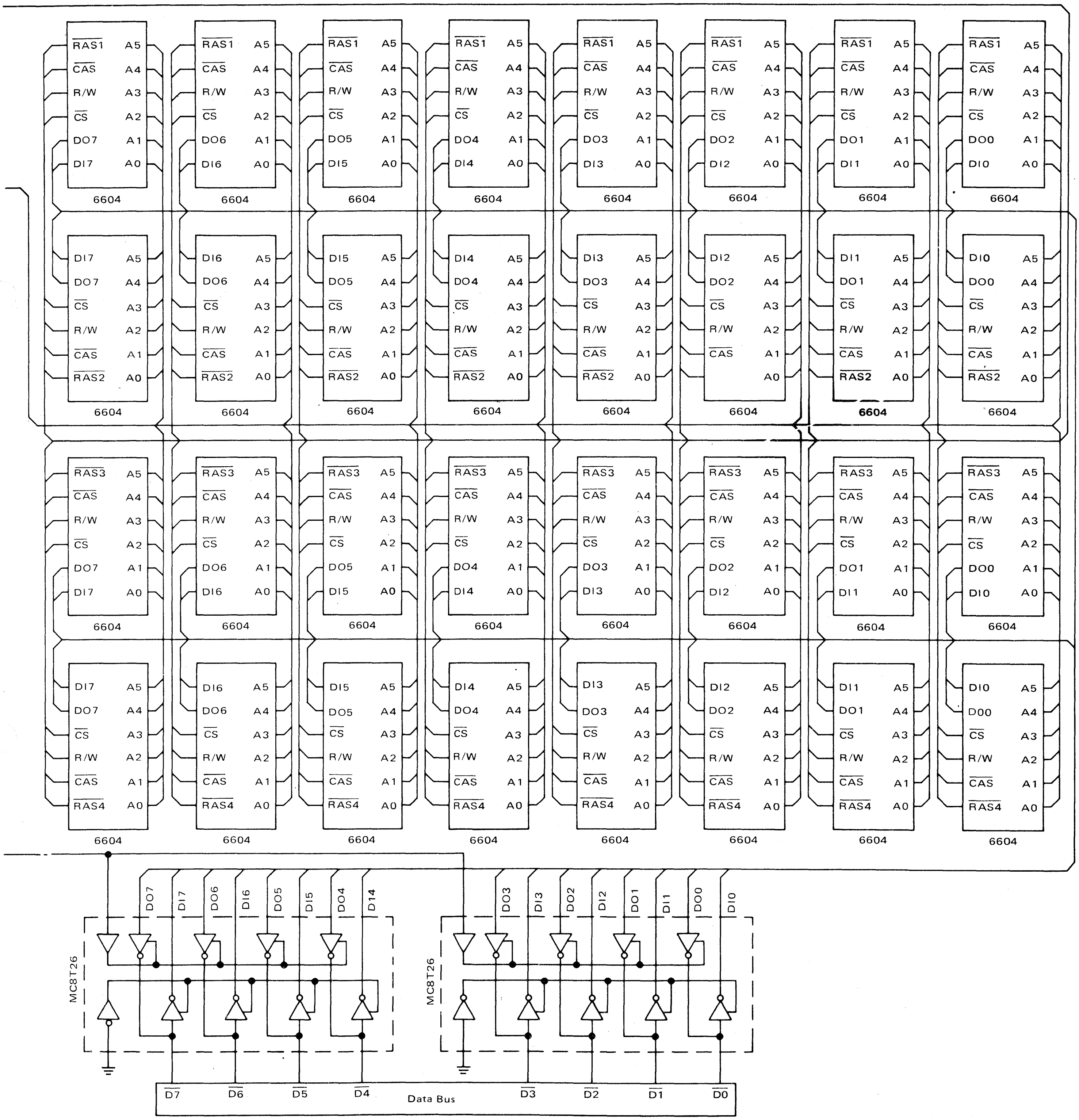


FIGURE 18 – 16K Dynamic RAM Using 4 fo and 2 fo Timing



RAS1
CAS
R/W
CS
DO7
DI7
DO6
DI6
DO5
DI5
DO4
DI4
DO3
DI3
DO2
DI2
DO1
DI1
DO0
DI0
A5
A4
A3
A2
A1
A0
6604
RAS2
RAS3
RAS4
MC8T26
MC8T26
D7
D6
D5
D4
Data Bus
D3
D2
D1
D0

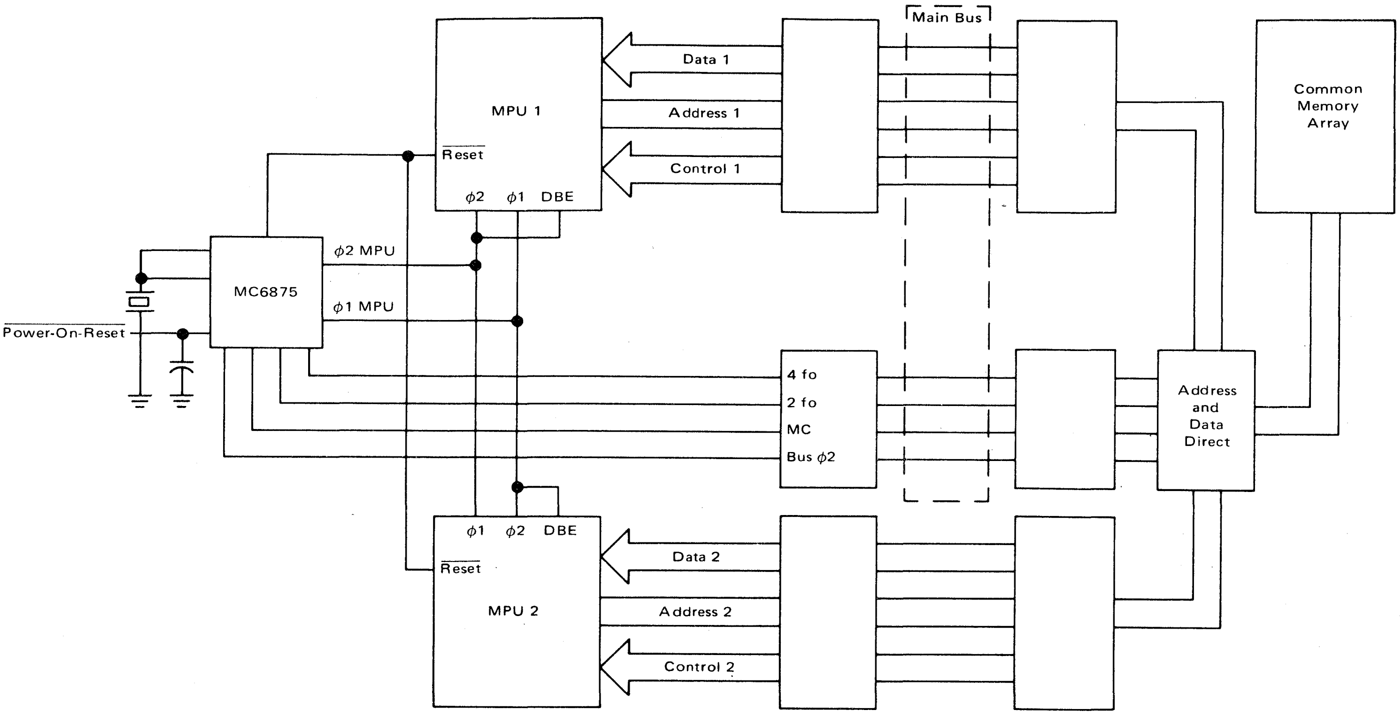


FIGURE 19 – Dual Processor Block Diagram

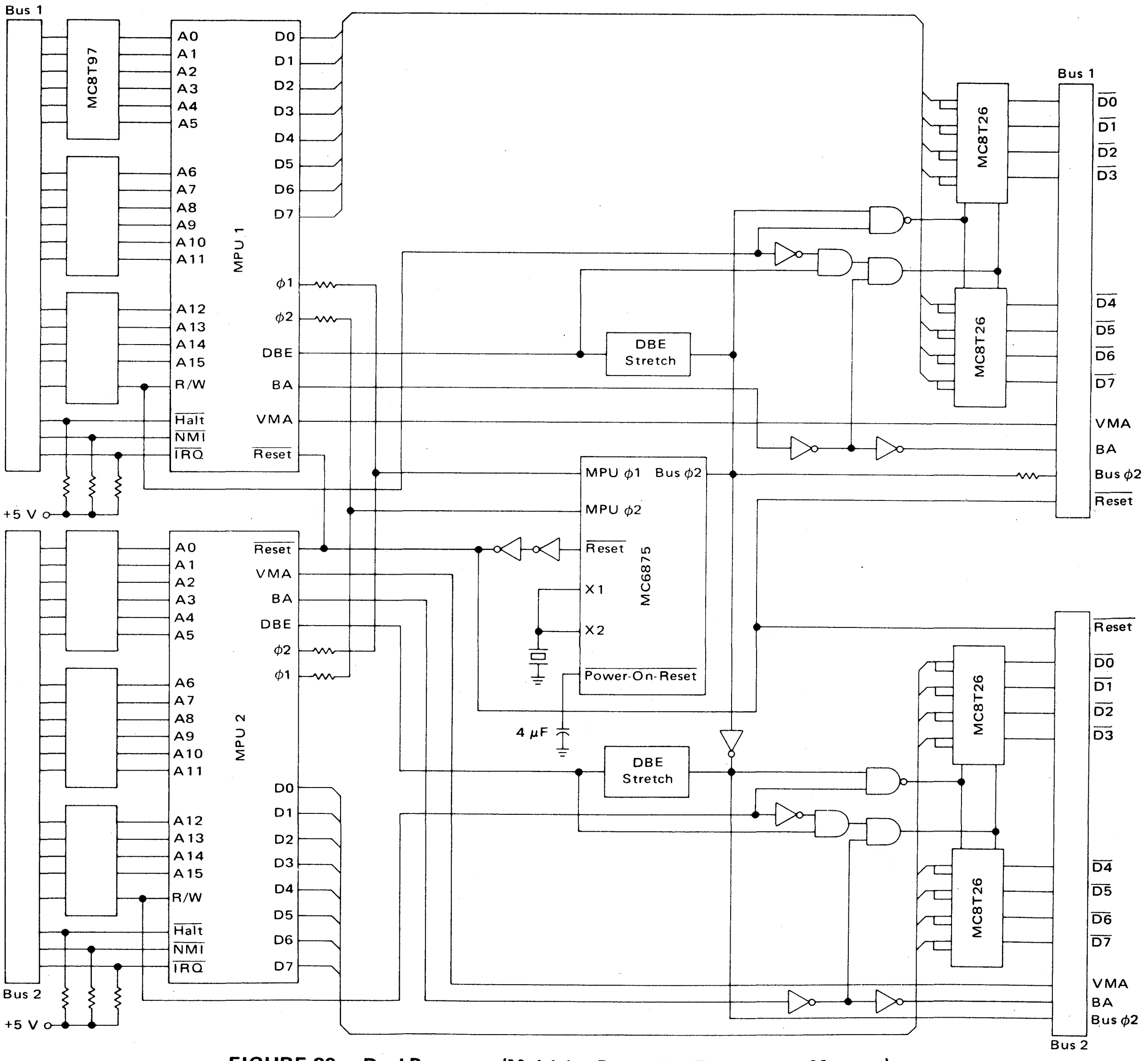


FIGURE 20 – Dual Processor (Multiplex Processing To Common Memory)

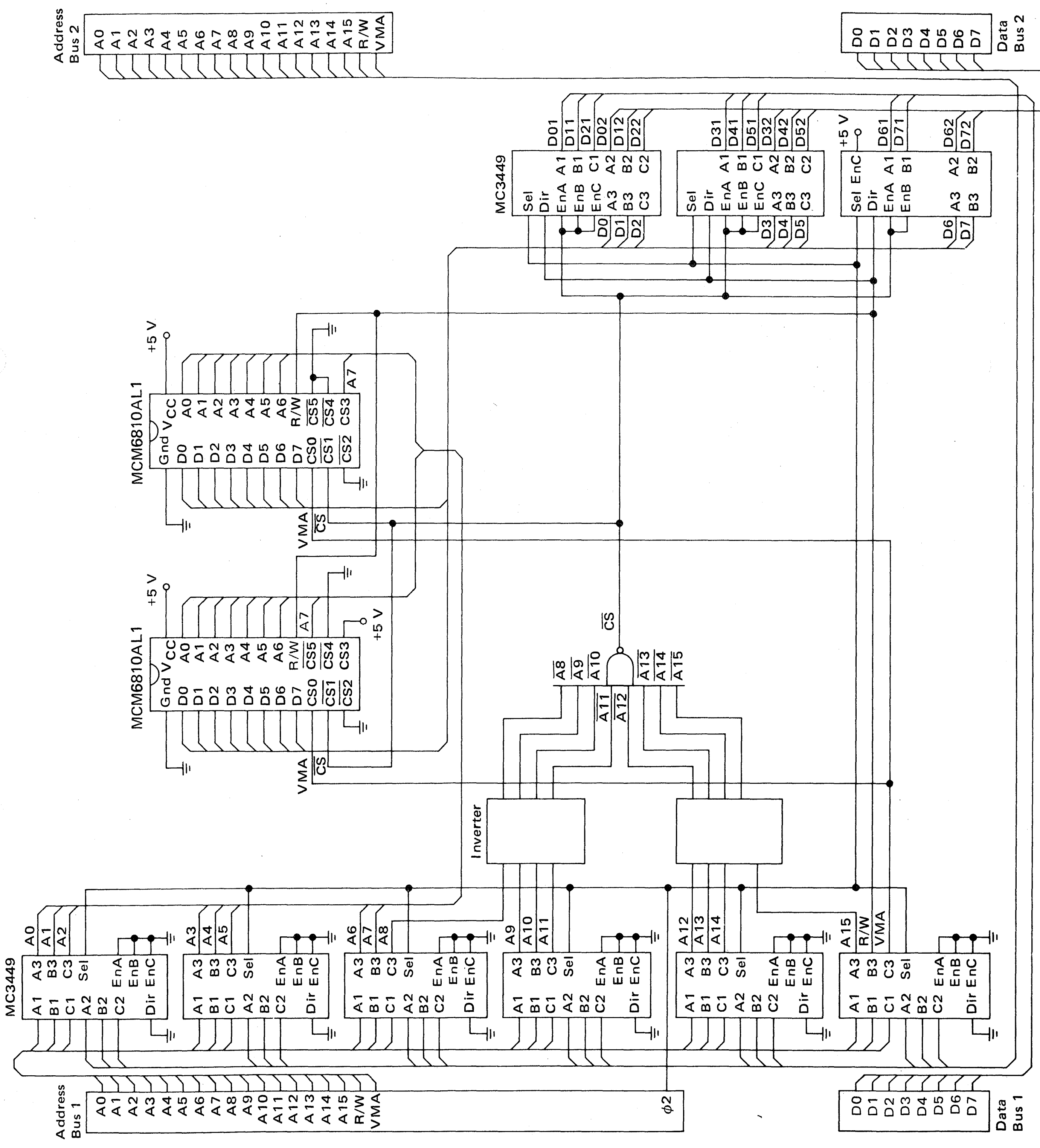


FIGURE 21 — Common Memory For Dual Processor (Multiplexing)

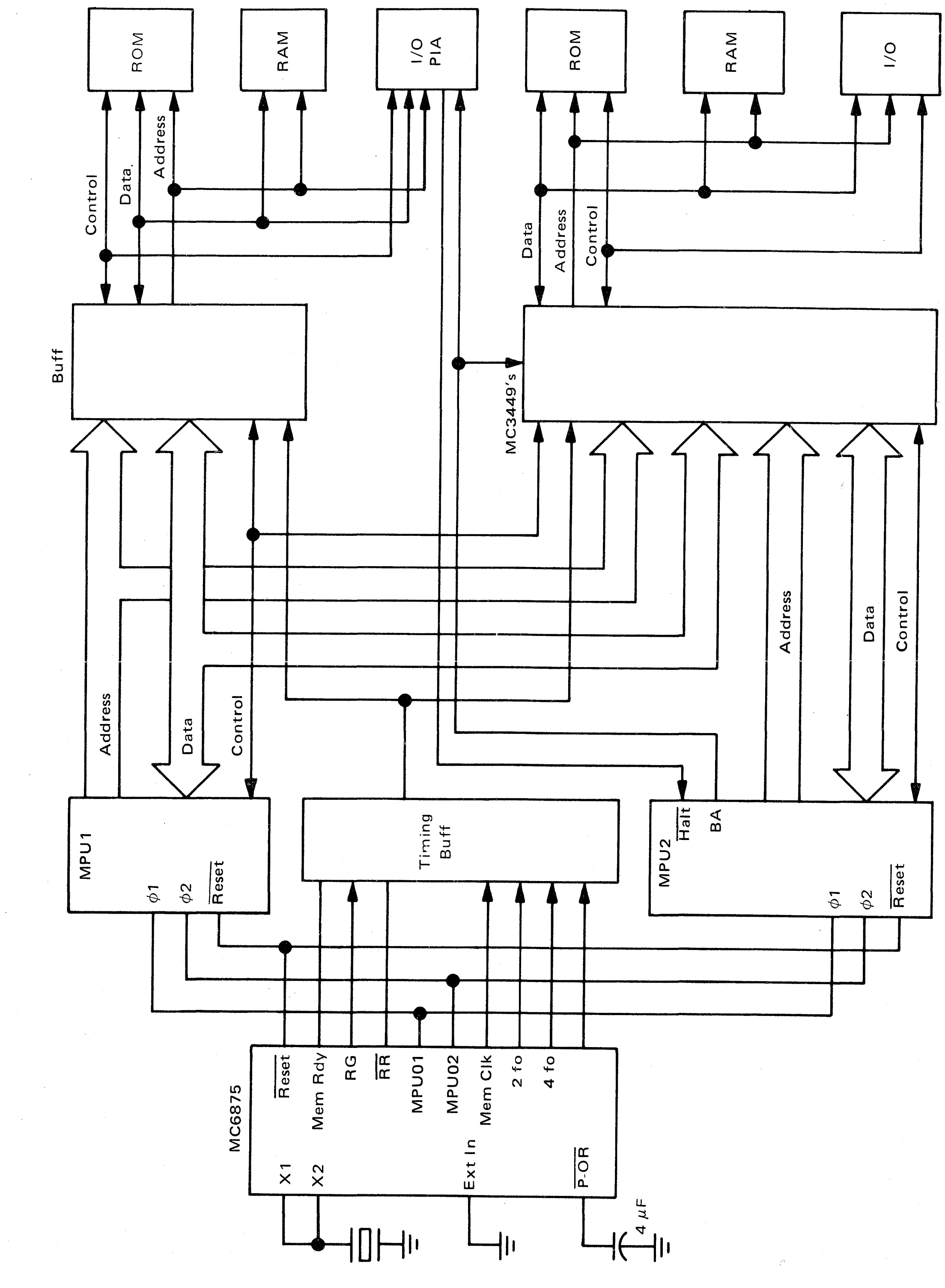


FIGURE 22 — Dual Processor (DMA) Using Halt

MOTOROLA Semiconductor Products Inc.

Printed in Switzerland 185-177/5.0