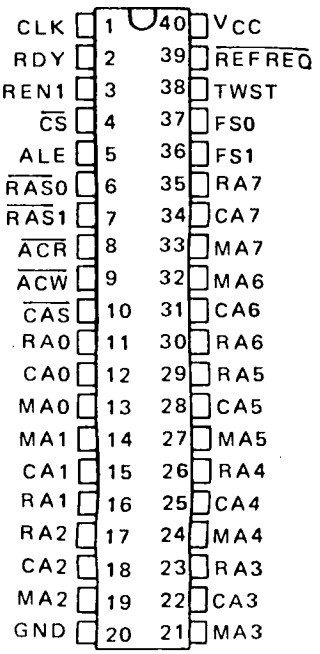
TMS 4500A DYNAMIC RAM CONTRO

JANU

- Controls Operation of 8K/16K/32K/64K Dynamic RAMs
- Creates Static RAM Appearance
- One Package Contains Address Multiplexer, Refresh Control, and Timing Control
- Directly Addresses and Drives Up to 256K Bytes of Memory Without External Drivers
- Operates from Microprocessor Clock
  - No Crystals, Delay Lines, or RC Networks
  - Eliminates Arbitration Delays
- Refresh May Be Internally or Externally Initiated
- Versatile
  - Strap-Selected Refresh Rate
  - Synchronous, Predictable Refresh
  - Selection of Distributed, Transparent, and Cycle-Steal Refresh Modes
  - Interfaces Easily to Popular Microprocessors
- Strap-Selected Wait State Generation for Microprocessor/Memory Speed Matching
- Ability to Synchronize or Interleave Controller with the Microprocessor System (Including Multiple Controllers)
- Three-State Outputs Allow Multiport Memory Configuration

TMS 4500A
40-PIN 600-MIL PLASTIC
DUAL-IN-LINE PACKAGE
(TOP VIEW)



description

The TMS 4500A is a monolithic DRAM system controller designed to provide address multiplexing, timing, control and refresh/access arbitration functions to simplify the interface of dynamic RAMs to microprocessor systems.

The controller contains a 16-bit multiplexer that generates the address lines for the memory device from the 16 system address bits and provides the strobe signals required by the memory to decode the address. An 8-bit refresh counter generates the 256-row addresses required for refresh.

A refresh timer is provided that generates the necessary timing to refresh the dynamic memories and assure data retention.

The TMS 4500A also contains refresh/access arbitration circuitry to resolve conflicts between memory access requests and memory refresh cycles. The TMS 4500A is offered in a 40-pin, 600-mil dual-in-line plastic package and is guaranteed for operation from 0°C to 70°C.

Texas Instruments

PRODUCT PREVIEW

This document contains information on a product under development. Texas Instruments reserves the right to change or discontinue this product without notice.

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 225012 • DALLAS, TEXAS 75265

SCG690A

TMS4500A Dynamic RAM Controller User's Manual

Memory Support

TEXAS INSTRUMENTS

TMS4500A Dynamic RAM Controller User's Manual

TEXAS INSTRUMENTS

IMPORTANT NOTICES

Texas Instruments reserves the right to make changes at any time in order to improve design and to supply the best product possible.

TI cannot assume any responsibility for any circuits shown or represent that they are free from patent infringement.

Second Printing
Reprinted (January 1985)

Third Printing
Reprinted (September 1986)

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

1. INTRODUCTION

Dynamic RAMs have many advantages such as low cost, high density, and low power dissipation. These features are important to every system but the problems associated with interfacing to dynamic memories sometimes seem to outweigh the advantages. Often additional circuitry is required to handle such things as address multiplexing, intricate timings, and refreshing. This additional circuitry sometimes prevents dynamic memories from becoming cost effective in small systems. The TMS 4500A dynamic memory controller solves the interfacing problems of dynamic RAMs so that even small systems can benefit from the advantages of dynamic RAMs.

The TMS 4500A provides address multiplexing, cycle timing, and refreshing for all multiplexed address dynamic RAMs. This controller operates directly from the system clock and needs no crystals, RC timing circuits, or delay lines. Refresh-access arbitration is handled synchronously with the system clock to eliminate arbitration delays and metastability problems associated with asynchronous operation.

The purpose of this manual is to describe the functional operation of the TMS 4500A and how it can be used in a microprocessor system. Section 1 provides an introduction to the TMS 4500A DRAM controller, Section 2 describes the basic operation of the controller, Section 3 provides information on typical implementation such as required signals and wait states, Section 4 provides design criteria for key memory and microprocessor specifications, and Section 5 provides some actual application examples.

2. FUNCTIONAL DESCRIPTION

The TMS 4500A consists of six basic building blocks: address and select latches, refresh rate generator, refresh counter, the multiplexer, the arbiter, and the timing and control block (See Figure 1)

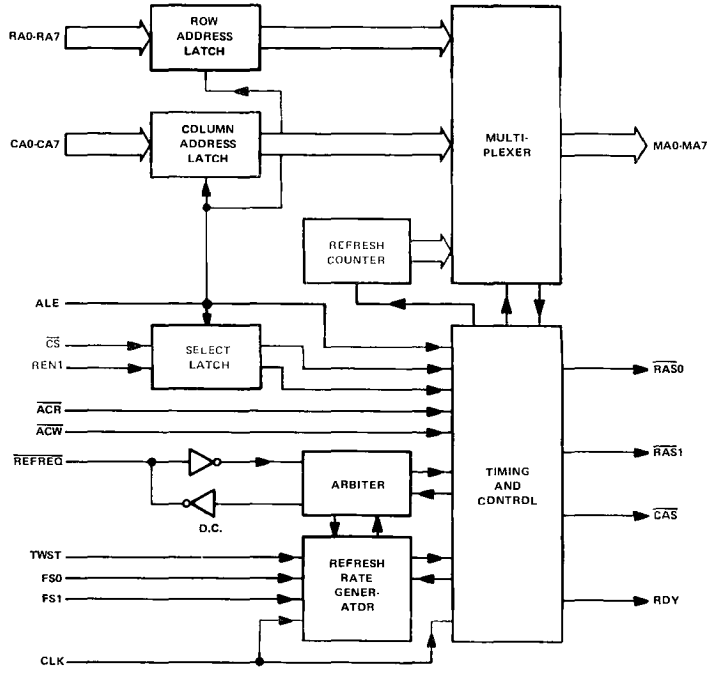


FIGURE 1 — TMS 4500A BLOCK DIAGRAM

2.1 ADDRESS AND SELECT LATCHES

The address and select latches allow the DRAM controller to be used without external latches in systems which multiplex address and data on the same line. The row address latches are transparent, meaning that while ALE is high, the output at MA0-MA7 follows inputs RA0-RA7 (with the assumption that a refresh cycle is not in progress).

2.2 REFRESH RATE GENERATOR

The refresh rate generator is a counter that indicates to the arbiter that it is time for a refresh cycle. The counter divides the clock frequency according to the configuration straps as shown in Table 1. The counter is reset when a refresh cycle is requested or when TWST, FS1, and FS0 are low. The configuration straps allow matching of memories to the system access time. Upon power-up it is necessary to provide a reset signal by driving all three straps to the controller low to initialize internal counters. A system's low-active, power on reset ($\overline{\text{RESET}}$) can be used to accomplish this by connecting it to those straps which are desired high during operation.

2.3 REFRESH COUNTER

The refresh counter is an eight bit counter that contains the address of the row to be refreshed. The counter is decremented after each refresh cycle.

2.4 MULTIPLEXER

The multiplexer provides the DRAM array with row, column, and refresh addresses at the proper times. Its inputs are the address latches and the refresh counter. The outputs provide up to sixteen multiplexed addresses on eight lines. The multiplexer can drive address inputs for up to 32 DRAM devices.

2.5 ARBITER

The arbiter provides two operational cycles: access and refresh. The arbiter resolves conflicts between cycle requests and cycles in execution and schedules the inhibited cycle when used in cycle steal mode.

TABLE 1 — STRAP CONFIGURATION

STRAP INPUT MODES			WAIT STATES FOR MEMORY ACCESS	REFRESH RATE	MINIMUM CLK FREQ. (MHz)	REFRESH FREQ. (kHz)	CLOCK CYCLES FOR EACH REFRESH
TWST	FS1	FS0					
L	L	L†	0	EXTERNAL		REFREQ	4
L	L	H	0	CLK÷31	1.984	64 - 95‡	3
L	H	L	0	CLK÷46	2.944	64 - 85‡	3
L	H	H	0	CLK÷61	3.904	64 - 82§	4
H	L	L	1	CLK÷46	2.944	64 - 85‡	3
H	L	H	1	CLK÷61	3.904	64 - 80‡	4
H	H	L	1	CLK÷76	4.864	64 - 77‡	4
H	H	H	1	CLK÷91	5.824	64 - 88•	4

† This strap configuration resets the Refresh Timer circuitry

‡ Upper figure in refresh frequency is the frequency that is produced if the minimum CLK frequency of the next select state is used

§ Refresh frequency if CLK frequency is 5 MHz

• Refresh frequency if CLK frequency is 8 MHz

2.6 TIMING AND CONTROL BLOCK

The timing control block executes the operational cycle at the request of the arbiter. It provides the DRAM array with the $\overline{RAS}$ and $\overline{CAS}$ signals and provides the CPU with a RDY signal. It controls the multiplexer during all cycles. It resets the refresh rate generator and decrements the refresh counter during refresh cycles.

3. TYPICAL IMPLEMENTATION

3.1 REQUIRED SIGNALS FROM MICROPROCESSOR

The signals re .·· ɔd by the TMS 4500A and usually gener · · by the microprocessor are ALE and $\overline{ACR}$ or $\overline{ACW}$ (· · will be used to refer to either $\overline{ACR}$ or $\overline{A}$·. ∴. The falling edge of ALE latches the sixteen address inputs, $\overline{CS}$ and $\overline{REN1}$. If $\overline{CS}$ is valid, a memory access cycle will be initiated. Memory address outputs (MA0-MA7) will present the row address that was latched on RA0-RA7 and a $\overline{RAS}$ output will go low ($\overline{RAS0}$ or $\overline{RAS1}$ depending on the state of REN1). The ACX signal should be driven low next as this will control the outputs MA0-MA7 to become the column address. After a short delay, the $\overline{CAS}$ output will go low. This sequence is shown in Figure 2.

NOTE

In all of the following examples involving the TMS 4500A, the values used for the parameters are preliminary values and subject to change.

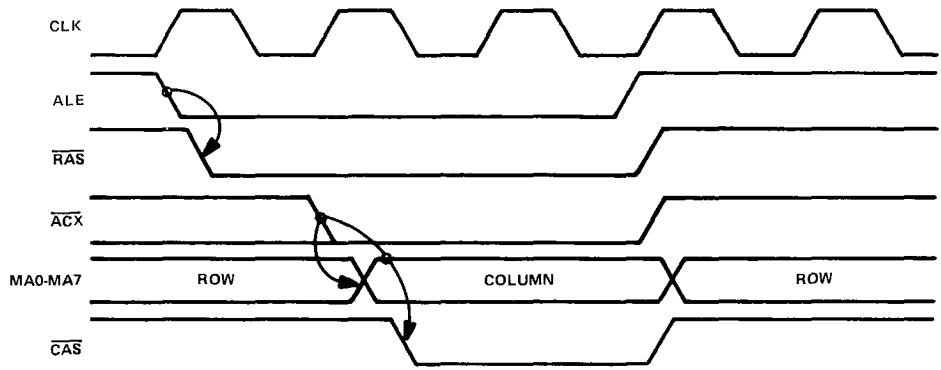


FIGURE 2 — SEQUENCE OF TIMINGS FOR TYPICAL ACCESS CYCLE

The ALE to $\overline{ACX}$ delay must be long enough to guarantee the row address hold time of the memory being used. If ALE and $\overline{ACX}$ are dropped at the same time, the hold time from $\overline{RAS}$ to address change is a minimum of 25 ns. This hold time is only sufficient for the -15 and -20 speed range parts.

The delay from $\overline{ACX}$ to $\overline{CAS}$ depends on the delay time to change to the column addresses. Internal to the TMS 4500A is a circuit that enables the latched column address to memory address outputs and enables $\overline{CAS}$ when the column address has had sufficient time to become stable. The time required to change the memory address outputs depends on the capacitive load of the address inputs of the memory devices; therefore the $\overline{CAS}$ circuitry must insure address settling prior to $\overline{CAS}$ activation. This feature is illustrated in Figure 3.

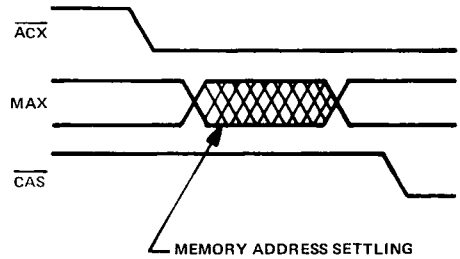


FIGURE 3 — DELAY FROM $\overline{ACX}$ TO $\overline{CAS}$

3.2 RDY SIGNAL FROM CONTROLLER

The RDY signal from the controller indicates to the CPU to enter a wait state. There are two instances which will cause RDY to go low. The first case will occur when the TMS 4500A is strapped to insert a wait state for every memory cycle (TWST = 1). This enables the controller to work with slower memories by inserting a wait period into the CPU memory cycle to allow for longer access times (See Figure 4)

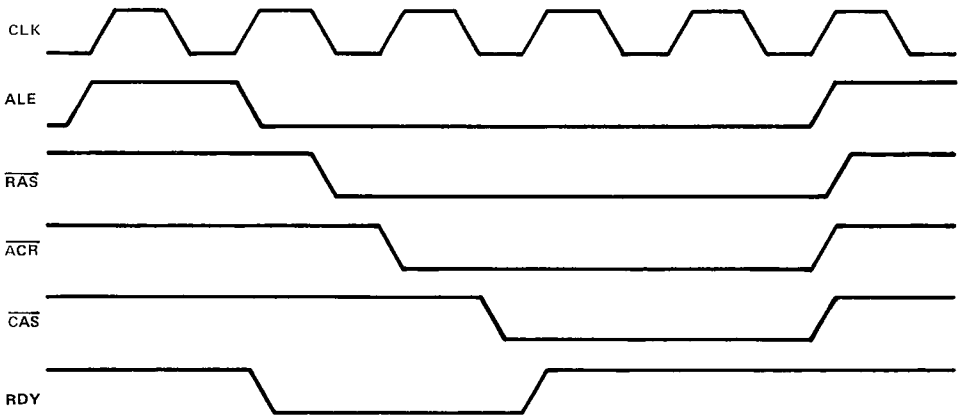


FIGURE 4 — ADDING 1 WAIT STATE TO EVERY MEMORY CYCLE

The other instance when RDY will go low is when the controller is actively engaged in a refresh cycle and cannot permit the access cycle to initiate immediately. Due to the synchronous operation of the TMS 4500A controller, there are only a few possibilities for refresh-access arbitration. In order to fully understand its operation, the following examples are given.

3.3 REFRESH - ACCESS ARBITRATION EXAMPLES

First let us examine the case of an access request while a refresh cycle is taking place (See Figure 5). The RDY line will go low when ALE goes low and the refresh cycle will continue. After the refresh is completed and the precharge time is satisfied, $\overline{RAS}$ will go low to service the access. The RDY line will go high when the access cycle is initiated.

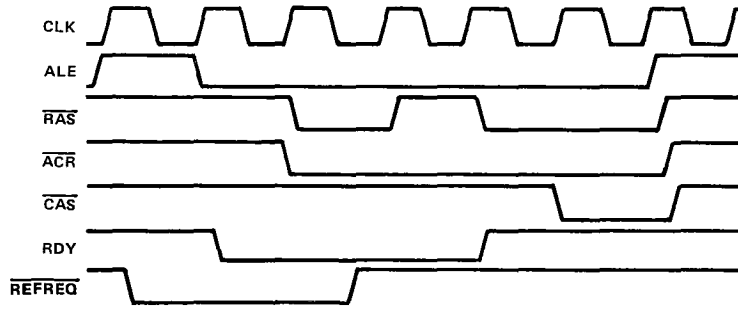


FIGURE 5 — ACCESS CYCLE HELD OFF BY REFRESH CYCLE (3 CYCLE REFRESH).

Another situation arises when a refresh becomes due during consecutive access cycles (See Figure 6). The current access cycle will be completed and then a refresh will take place. The RDY line will go low as soon as the ALE line drops on the second access request. When the refresh cycle is complete, the second access is performed. The RDY line will then go high and future access requests will be accepted.

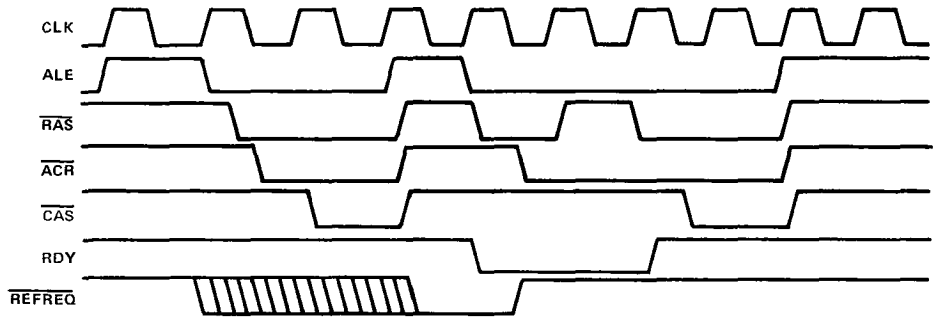
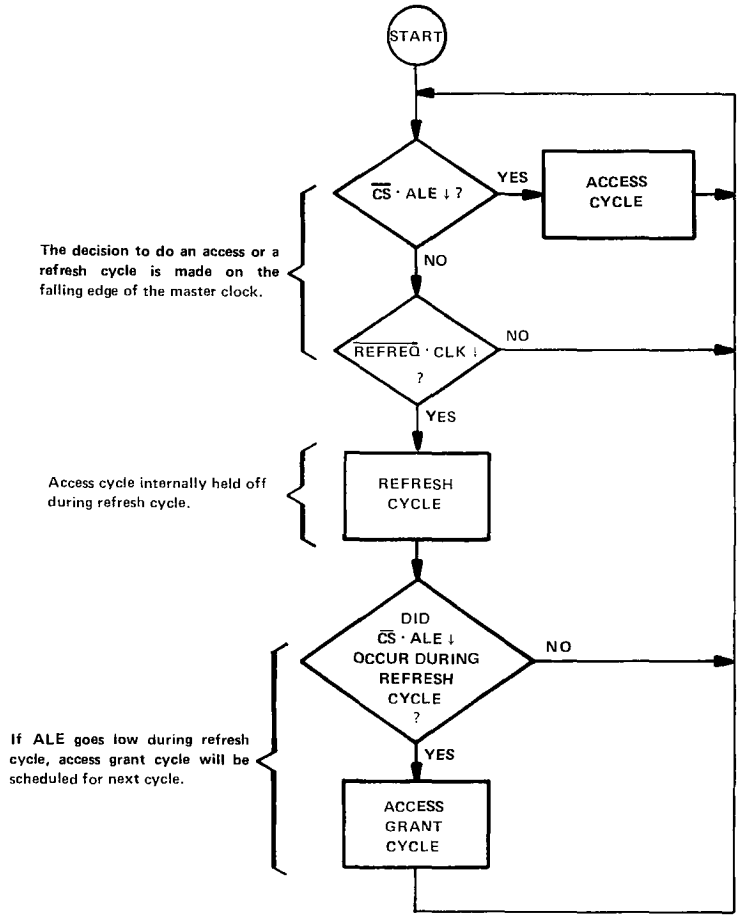


FIGURE 6 — REFRESH OCCURRING BETWEEN CONSECUTIVE ACCESS CYCLES.

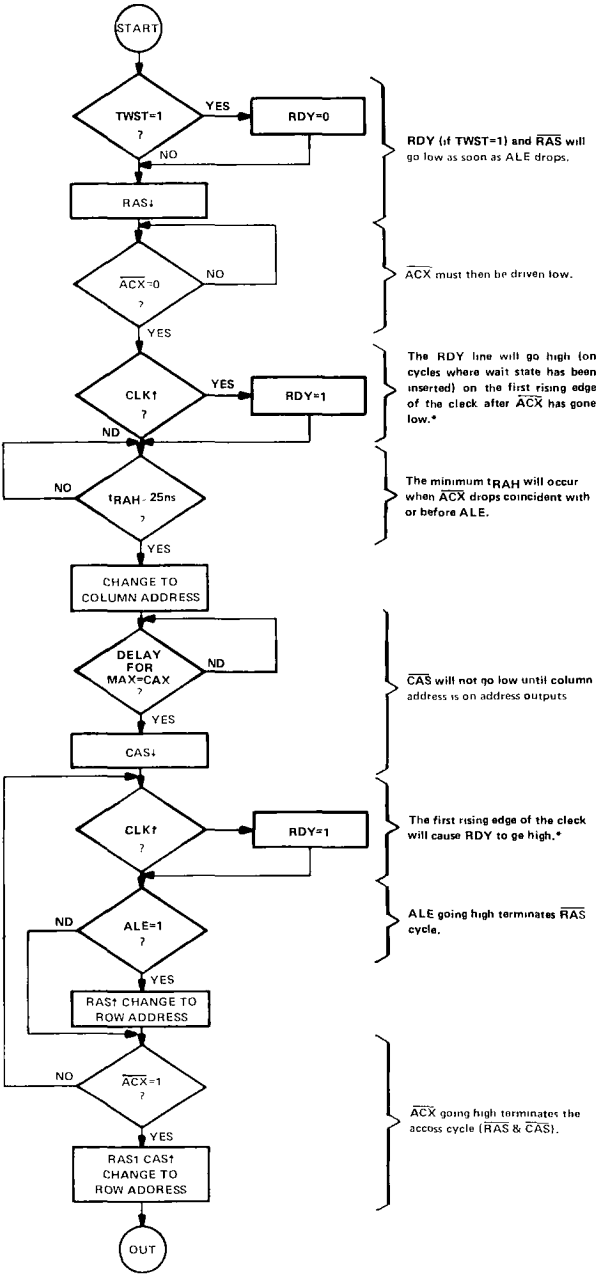
The key to solving the arbitration problem between refresh and access cycles is making the operation synchronous. The internal refresh request on the TMS 4500A always happens on the rising edge of the master clock. The decision whether the cycle is an access or a refresh operation is made on the next falling edge of the clock. For this reason, ALE must avoid the region between 10 ns prior to the high to low clock transition and 10 ns after this edge.

A flow chart that provides a general overview of DRAM controller operation is shown in Figure 7. This flow chart shows the sequence of events under different refresh and access request situations. Each of the types of cycles: access, refresh and access grant (access cycle held off by refresh) are further explained with a flow chart for each cycle type. The access cycle is shown in Figure 8, the refresh cycle in Figure 9, and the access grant cycle is shown in Figure 10.



ALE↓ and CLK↓ are guaranteed not to occur simultaneously by proper phasing of the clock.

FIGURE 7 — FLOW CHART SHOWING GENERAL OVERVIEW OF DRAM CONTROLLER OPERATION



* The first rising edge of the CLK could occur before or after $\overline{CAS}$ has gone low.

FIGURE 8 — FLOW CHART OF ACCESS CYCLE

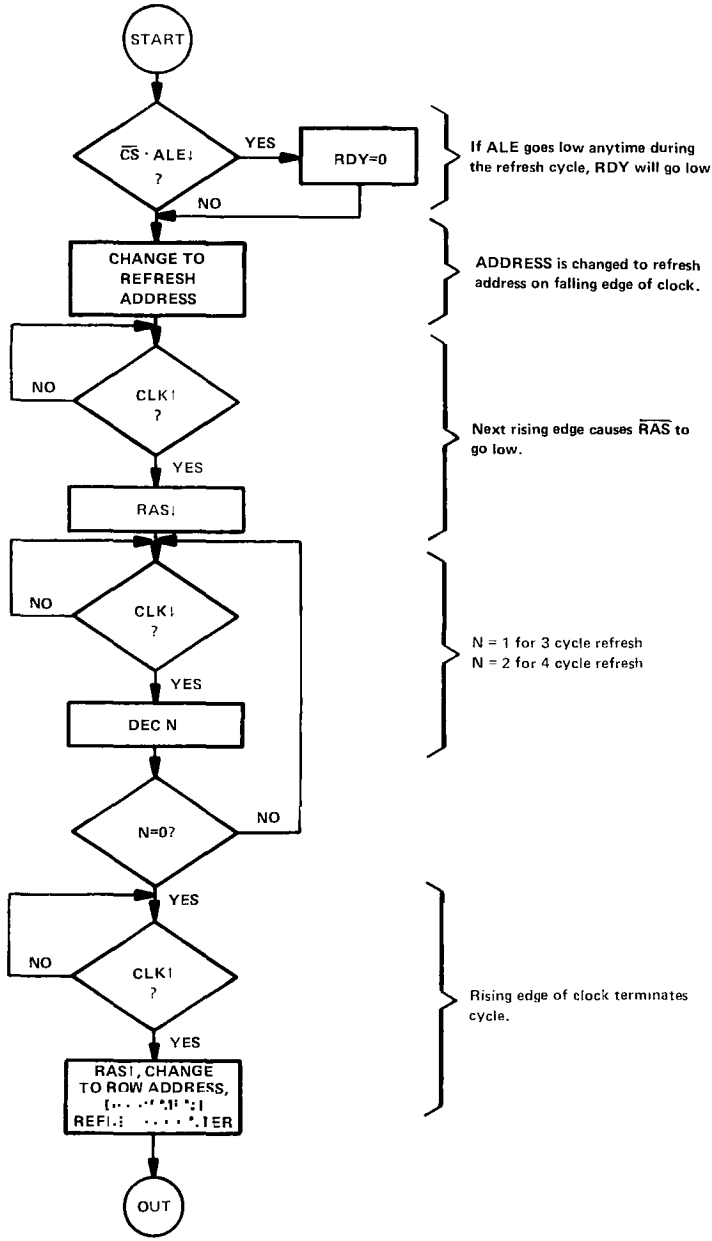


FIGURE 9 — FLOW CHART OF REFRESH CYCLE

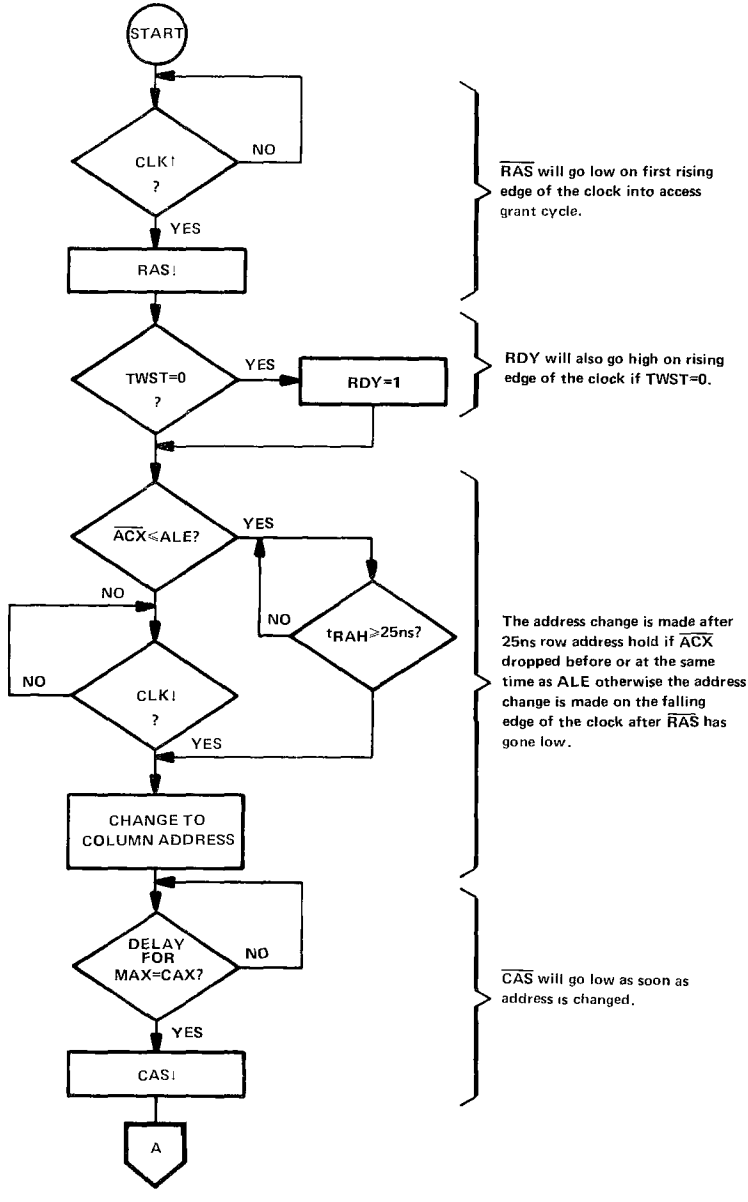


FIGURE 10 – FLOW CHART OF ACCESS GRANT CYCLE (SHEET 1 OF 2)

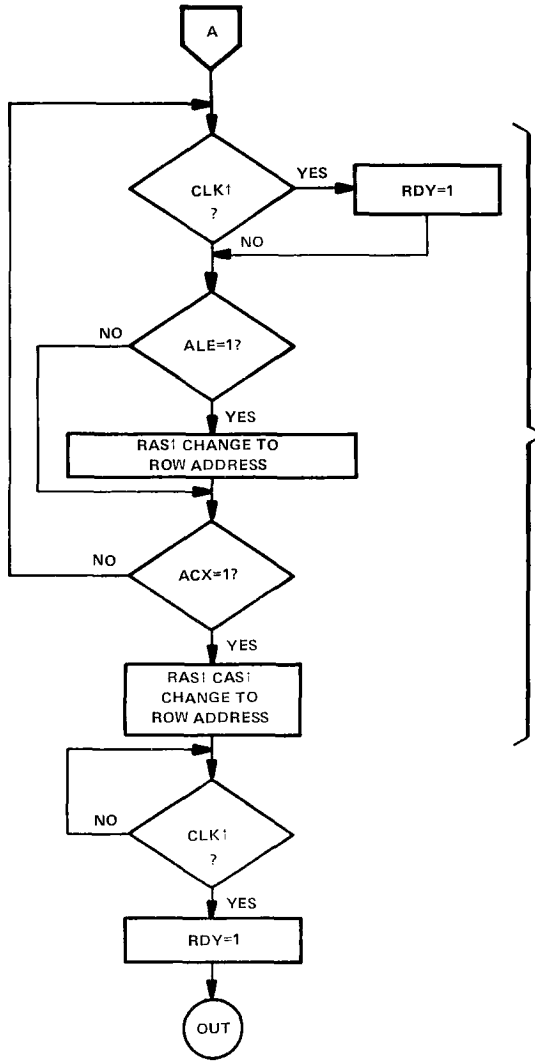


FIGURE 10 – FLOW CHART OF ACCESS GRANT CYCLE (SHEET 2 OF 2)

4. DESIGN CRITERIA

To use the TMS 4500A DRAM controller as an interface between dynamic memory and any microprocessor, certain key specifications should be examined. Each microprocessor has unique timing signals and dynamic memory timing requirements vary with device type and speed. To guarantee a proper match between memory and processor, these specifications should be checked:

1. Refresh time
2. Memory precharge time
3. ALE to clock relationship
4. Row address setup and hold time
5. Data valid to write enable time
6. Read access time

4.1 REFRESH TIME

The refresh generated by the TMS 4500A is a distributed refresh. The time between refresh cycles is determined by the clock frequency and the setting of the frequency select straps (FS0, FS1, TWST). Once the clock speed has been selected the straps can be selected for proper refresh time (See Table 1).

Alternate refresh modes are possible by using external refresh requests. The $\overline{\text{REFREQ}}$ line is an open collector output and can be driven by an external · collector device to request refresh cycles (Due to the maximum required low-level input voltage on $\overline{\text{REQ}}$ of 1.2 V, this input can be driven low by using a diode in series with a standard TTL output). This allows refresh cycles to be requested by the microprocessor or during times when the dynamic memory will not be accessed (opcode decode time, halt states, or when accessing to other memory). Another possibility, if memory size is large enough to require two controllers, would be to interleave the controllers so that one controller will refresh while the other is performing an access. In either case, refresh is guaranteed by the refresh timer if its time period expires. The only case when refresh is totally dependent on external requests is when all speed selection straps are low (The speed selection straps should not be re-configured after initial reset as the counter may not detect the correct count).

4.2 MEMORY PRECHARGE TIME

The precharge time ($\overline{\text{RAS}}$ high time) is the time needed between accesses to allow internal nodes of the RAM to charge to their correct reference levels. If precharge time is not sufficient, the RAMs may not function properly. The $\overline{\text{RAS}}$ high time is equal to the time from ALE or $\overline{\text{ACX}}$ high to ALE low on access cycles, equal to one clock cycle on access grant cycles, and is determined by ALE or $\overline{\text{ACX}}$ high to clock high on refresh cycles.

4.3 ALE TO CLK RELATIONSHIP

During a memory cycle, the ALE signal should go high to allow the address to change and then go low to latch the valid memory address. The falling edge of ALE will then initiate an access cycle if $\overline{\text{CS}}$=0. The high to low transition of ALE should not occur within 10 ns before and 10 ns after a falling edge of CLK. $\overline{\text{RAS}}$ should be made high for one negative transition of CLK so that the controller will arbitrate a refresh if a refresh is due. If ALE is coincident with the falling edge of CLK, the clock should be inverted or ALE delayed.

4.4 ROW ADDRESS SETUP AND HOLD TIME

The address must be present at RA0-RA7 and CA0-CA7 10 ns before ALE is driven low in order for this address to be latched. Similarly, a delay must be used to guarantee 0 ns row address setup time since the delay from RAX to MAX ($t_{RAV-MAV}$ = 45 ns) can be greater than the delay from ALE to RAS ($t_{AEL-REX}$ = 40 ns). To guarantee 0 ns setup time, the address should be valid at least 10 ns before ALE goes low (32 chips, 160 pF address load).

The row address hold time is also a key parameter to consider. This hold time is determined by the spacing from ALE to $\overline{\text{ACX}}$. $\overline{\text{ACX}}$ changes the address from row to column when it is driven low. If $\overline{\text{ACX}}$

goes low before or coincident with ALE, internal interlocking will provide at least 25 ns hold time from $\overline{RAS}$ to address change. 25 ns hold time is long enough for -15 and -20 speed memories but if a longer hold time is required, the delay from ALE to $\overline{ACX}$ is given by:

$$t_{AEL-ACL} = t_{h(AR)} + 20 \text{ ns}$$

where $t_{h(AR)}$ = hold time required

4.5 DATA VALID TO WRITE ENABLE

Unlike static memories, dynamic RAMs require data-in to be valid when the write signal goes low; or in the case of an early write, data must be valid when $\overline{CAS}$ goes low. The write line to the memory is routed externally from the controller to allow more versatility (R/M/W or early W possible). Depending on the microprocessor's timing of write data and write enable, the write enable may need to be delayed before it is presented to the memory.

Early-write cycles are advantageous to microprocessor systems if they can be used. An early-write cycle is a cycle in which the write line goes low before $\overline{CAS}$ goes low (in this case data-in must be valid before $\overline{CAS}$ goes low). The advantage comes from the fact that the DRAM output buffers never turn on in this type of a cycle. This allows the data-in and data-out of the DRAM to be tied to the same bus without the use of bus transceivers

4.6 READ ACCESS TIME

The access time of a memory device is a critical parameter in selecting memory speed. To insure read data will be valid, neither access time from $\overline{RAS}$ nor access time from $\overline{CAS}$ can be violated. In many cases using the TMS 4500A controller, the $\overline{RAS}$ to $\overline{CAS}$ spacing will be greater than 50 ns so the access time from $\overline{CAS}$ becomes the critical parameter. The required access time can be found by calculating the time from $\overline{ACX}$ to when data must be valid in the system. From this time, subtract the maximum delay from $\overline{ACX}$ to $\overline{CAS}$ and the result will be the required access time from $\overline{CAS}$. The required access time on delayed access cycles (access grant cycles) can be different. If ACX drops after ALE drops, $\overline{CAS}$ is generated from the clock instead of dependent on the $\overline{ACX}$ signal. To calculate $\overline{CAS}$ access time on access,grant cycles, the time from CLK low (the edge that causes $\overline{CAS}$) to when data must be valid is used instead of the time from $\overline{ACX}$ to data valid. From this the delay from CLK low to $\overline{CAS}$ low is subtracted to yield the required access time from $\overline{CAS}$. If data bus buffers are used in the data path, their propagation delay must be considered. When $\overline{ACX}$ drops coincident or prior to ALE, $\overline{CAS}$ is generated by the rising clock edge that enables $\overline{RAS}$ (allowing row address hold and column address setup times).

5. APPLICATION EXAMPLES

5.1 8085A EXAMPLE

An example of an application using an 8085A microprocessor is shown in Figure 11. This example has 16 4116s organized as 32K bytes of memory. The IO/$\overline{M}$ and A15 decoding for $\overline{CS}$ ensure that the memory controller is only selected when a memory request in the address range from 8000_H-$FFFF_H$ is performed. The clock from the 8085A can be used directly as the clock for the controller. For the purpose of this example we will assume the 8085A will be operating at 3.125 MHz (maximum allowed by 8085A). The timing diagram is shown in Figure 12.

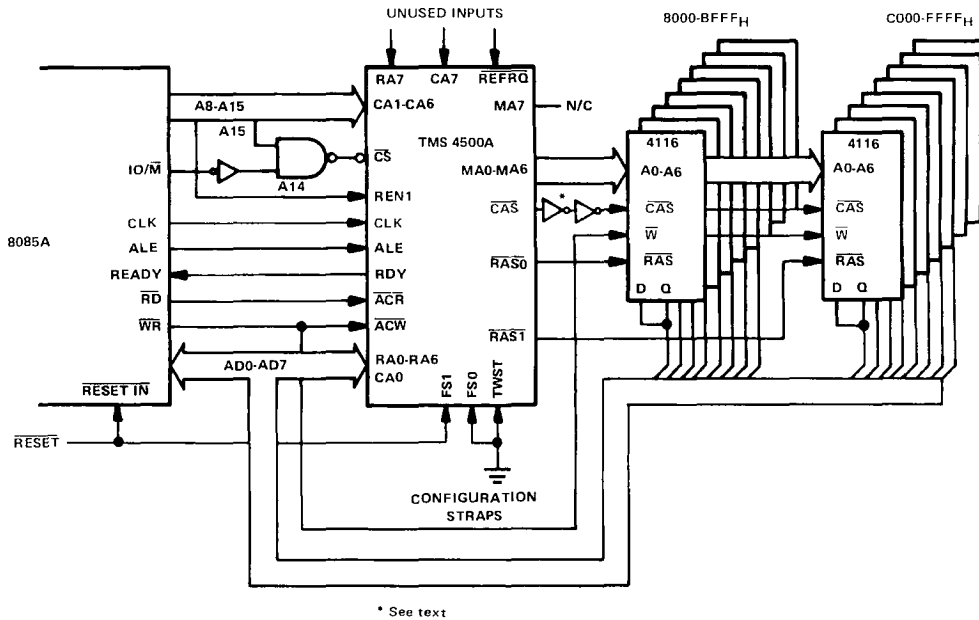
To illustrate the design criteria previously listed, we will go through the checklist and select the proper memory speed at this processor speed:

1. Refresh time

From Table 1, a clock frequency of 3.125 MHz would require straps.
FSO = 0, FS1 = 1, TWST = 0
This would yield a refresh rate with a frequency of approximately 68 kHz.
At this setting, a refresh cycle takes 3 clock cycles.

2 Memory precharge time

The memory precharge time must be calculated for each of the three cycle types (access, refresh, access grant) to be sure the minimum time is satisfied.



* See text

FIGURE 11 — 8085A INTERFACE TO TMS 4500A CONTROLLER

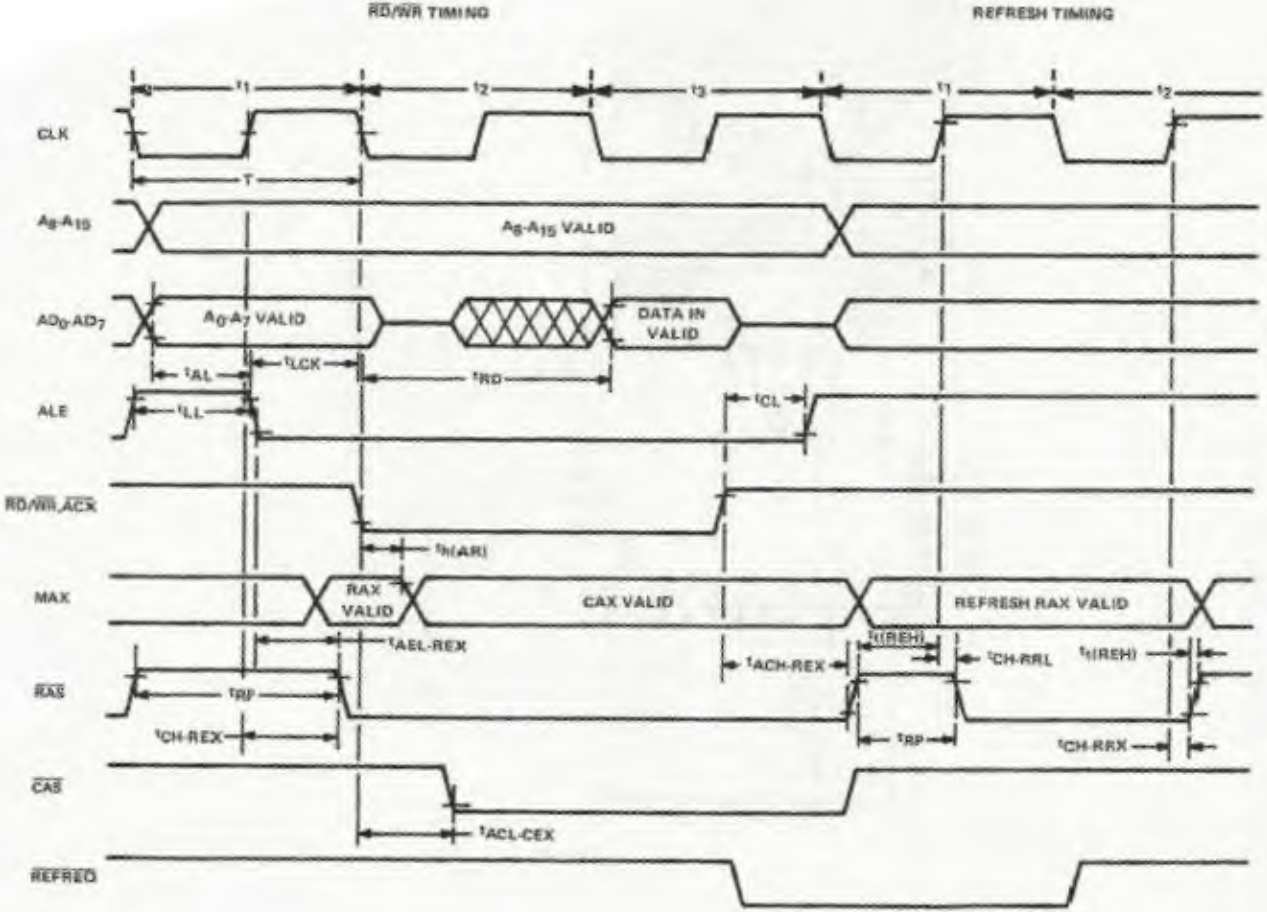


FIGURE 12 — 8085A TIMING DIAGRAM

a. Access cycles

The precharge time for access cycles is given by.

$$t_{RP} = t_{CL} + t_{LL} + t_{AEL\text{-}REX} - t_{ACH\text{-}REX} - t_{t(REH)}$$

where t_{RP} = memory precharge time
t_{CL} = trailing edge of control to leading edge of ALE (MIN, 8085A Spec.)
t_{LL} = ALE width (MIN, 8085A Spec.)
$t_{AEL\text{-}REX}$ = ALE low to $\overline{RAS}$ starting low (MAX, 4500A Spec.*).
$t_{ACH\text{-}REX}$ = $\overline{ACX}$ high to $\overline{RAS}$ starting high (MAX, 4500A Spec.)
$t_{t(REH)}$ = $\overline{RAS}$ high transistion time (MAX, 4500A Spec.)

and T = clock period = $1/f_{CLOCK}$ = $1/3.125 \times 10^6$ = 320 ns
t_{CL} = T - 110 = 50 ns
t_{LL} = T - 20 = 140 ns
$t_{AEL\text{-}REX}$ = 36 ns
$t_{ACH\text{-}REX}$ = 50 ns
$t_{t(REH)}$ = 25 ns

thus t_{RP} = (50 + 140 - 36 - 50 - 25) ns
Therefore, t_{RP} = 151 ns precharge time.

*This value should otherwise be a minimum value; however as all propagation delays on a given chip will tend to track each other, the maximum value is multiplied by a skew factor to reflect variations in same chip propagation delays. The skew factor for the TMS 4500A is 0.9. All values (followed by an asterisk) are obtained by multiplying the specified maximum value by 0.9.

b Access grant cycles

The precharge time for access grant cycles is given by:

$$t_{RP} = T + t_{CH\text{-}REX} - t_{CH\text{-}RRX} - t_{t(REH)}$$

where T = clock period
$t_{CH\text{-}REX}$ = CLK high to access $\overline{RAS}$ starting low (MAX, 4500A Spec.*)
$t_{CH\text{-}RRX}$ = CLK high to refresh $\overline{RAS}$ starting high (MAX, 4500A Spec.)
$t_{t(REH)}$ = $\overline{RAS}$ high transition time (MAX, 4500A Spec.)

*multiply the specified maximum value by 0.9.

thus t_{RP} = (320 + 63 - 45 - 25) ns
Therefore, t_{RP} = 313 ns precharge time.

c. Refresh cycles

The precharge time for refresh cycles is given by

$$t_{RP} = T - t_{CT} - t_{ACH\text{-}REX} - t_{t(REH)} + t_{CH\text{-}RRL}$$

where T = clock period
t_{CT} = CLK high to control high (Intel Application Note AP-38)
$t_{ACH\text{-}REX}$ = $\overline{ACX}$ high to $\overline{RAS}$ starting high (MAX, 4500A Spec.)
$t_{t(REH)}$ = $\overline{RAS}$ high transition time (MAX, 4500A Spec.)
$t_{CH\text{-}RRL}$ = CLK high to refresh $\overline{RAS}$ starting low (MAX, 4500A Spec.)

*multiply the specified maximum value by 0.9.

thus t_{RP} = (320 - 60 - 50 - 25 + 54) ns
Therefore, t_{RP} = 239 ns

The previous mathematical analysis provides a worst case precharge time of approximately 150 ns; this occurs during repeated access cycles. Since the prechage time of a TMS 4116-25 is 150 ns, the precharge condition for this device has been met.

3. ALE to CLK relationship

Recall that the ALE low transition must not occur within 10 ns of the CLK low transition. For an 8085A, the ALE high transition is iniated by a low transition of the CLK. ALE then goes low again at a time (t_{LCK}) before the next low CLK transition. This time interval (t_{LCK}) is:

$$t_{LCK} = \tfrac{1}{2}T - 60 \text{ ns (MIN, 8085A Spec.)}$$

where t_{LCK} = ALE low during CLK high

thus $t_{LCK} = (160 - 60)$ ns
Therefore, $t_{LCK} = 100$ ns

This value meets the ALE to CLK relationship condition.

4. Row address setup and hold time

The row address setup time is:

$$t_{AL} = \tfrac{1}{2}T - 50 \text{ ns (MIN, 8085A Spec.)}$$

where t_{AL} = Address valid before trailing edge of ALE

thus $t_{AL} = (160 - 50)$ ns
Therefore, $t_{AL} = 110$ ns

Since this value is greater than the 10 ns requirement, the row address setup time condition has been met.

The hold time from $\overline{RAS}$ to address change can be calculated using the 8085A's delay from ALE to $\overline{RD}$ or $\overline{WR}$ (t_{LC}) and setting this equal to the hold time equation given for the 4500A.

As $t_{AEL-ACL} = t_{h(AR)} + 20$ ns (4500A Spec.)

and $t_{LC} = \tfrac{1}{2}T - 30$ ns (8085A Spec.)

then $t_{AEL-ACL} = t_{LC}$
$t_{h(AR)} + 20 \text{ ns} = \tfrac{1}{2}T - 30$ ns
$t_{h(AR)} = \tfrac{1}{2}T - 50$ ns

thus $t_{h(AR)} = (160 - 50)$ ns
Therefore, $t_{h(AR)} = 110$ ns (with $T = 320$ ns)

The row address setup and hold times are easily satisfied for 4116-25's whose hold time ($t_{h(AR)}$) equals 35 ns minimum.

5. Data valid to write enable

Checking the data valid to write enable specifications shows this can be a restriction for some 8085A systems. In the example in Figure 11, the $\overline{WR}$ signal from the 8085A drives the $\overline{ACW}$ on the 4500A and also drives the $\overline{W}$ inputs on all the memory devices. Because of the delay of the 4500A from $\overline{ACW}$ to $\overline{CAS}$, the write operations are always early write operations (this enables D and Q to be tied together). The minimum delay for $\overline{ACW}$ low to $\overline{CAS}$ low is specified at 45 ns ($t_{ACL-CEX}$). The delay of the 8085A from write to data valid is a maximum of 40 ns with a 150 pF load. This says the loading on each data line must be no greater than 150 pF or the data will not be valid when $\overline{CAS}$ goes low after minimum delay. The loading of the dynamic memory would be 12 pF (D and Q tied together) for each device.

Add to this the capacitive loading of the DRAM controller (10 pF) and any other devices that are on the data bus (EPROMs, ROMs, etc.). As long as the total loading on the data bus is less than 150 pF, the system will operate without the inverters on $\overline{CAS}$ shown in Figure 11.

If the loading is greater than 150 pF, two inverters can be added to the $\overline{CAS}$ line between the 4500A and the memory bank (see Figure 11) to ensure that the data is valid before $\overline{CAS}$ goes low.

6. Read access time

In order to calculate the required access time for access cycles, determine the time from $\overline{ACR}$ to data valid (t_{RD}).

As $t_{RD} = (3/2 + N)T - 180$ ns (MAX, 8085A Spec.)

where N = Number of wait states required = 0 (for this example)
T = clock period = 320 ns

thus $t_{RD} = [(3/2)(320) - 180]$ ns
Therefore, $t_{RD} = 300$ ns

Next, subtract the delay from ACR to CAS ($t_{ACL-CEX} + t_{t(CEL)}$).
This value equals 130 ns with 320 pF loading.

$t_{RD} - (t_{ACL-CEX} + t_{t(CEL)})$
300 ns - 130 ns = 170 ns (access time from $\overline{CAS}$)

In order to calculate the access time requirements for access grant cycles, determine the time from CLK to data valid from the equation:

$$t_{LDR} - t_{LCK} = (4/2)T - 180 - [(\tfrac{1}{2})T - 60]$$ (8085A Spec.)

where T = 320 ns

thus $t_{LDR}-t_{LCK} = 4/2\ (320) - 180 - [(\tfrac{1}{2})\ 320 - 60]$

Therefore, $t_{LDR}-t_{LCK} = 360$ ns

Subtract the delay from CLK low to $\overline{CAS}$ ($t_{CL-CEX} + t_{t(CEL)} = 175$ ns with 80 pF loading).

360 ns − 175 ns = 185 ns (access time from $\overline{CAS}$)

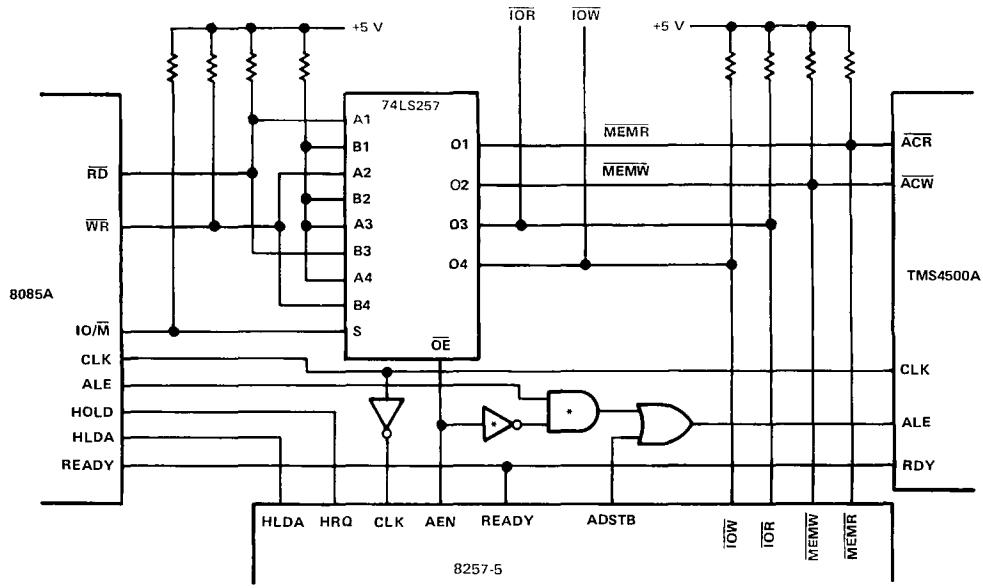
In this case, the fastest access time is required on access cycles. To guarantee the 170 ns access time from CAS, -25, -20, or -15 devices could be used since the longest access time of these devices is 165 ns for the 4116-25.

5.2 DMA AND 8085A EXAMPLE

A partial schematic of a DMA interface with the 8085A and 4500A DRAM controller is shown in Figure 13. On a DMA cycle, the 8257 will request that the 8085A enter a hold state. The 8085A will return with a HLDA when it is ready to release control. The 8257 will then drive AEN high to disable the 8085A's control of $\overline{ACR}$, $\overline{ACW}$, and ALE. The additional delays of the added circuitry must be considered when evaluating system compatability.

The same design criteria steps used to calculate a microprocessor's compatability should be used to check a DMA device. In this case ADSTB replaces ALE, and $\overline{MEMR}$ and $\overline{MEMW}$ replace $\overline{RD}$ and $\overline{WR}$ in the calculations for each step. If data is valid early enough from the DMA device, and extended writes (see 8257 Spec.) are used, a longer access time without wait states would be allowed.

Latches would probably be needed for both upper 8 and lower 8 address lines since the 8085A multiplexes address and data on lower 8 lines and the 8257 multiplexes upper address bits on data lines. These latches are not needed for the 4500 controller since it latches the address internally but most other types of devices require the address and $\overline{CS}$ throughout an entire cycle.



* Not required for 8085As that hold ALE low during HOLD.

FIGURE 13 — 8085A INTERFACE INCLUDING DMA CHANNEL

5.3 Z-80 EXAMPLE

The Z-80 microprocessor provides a $\overline{RFSH}$ signal and a 7-bit refresh address to provide for refresh requirements of dynamic RAMs. For this reason, the Z-80 is sometimes understood as the microprocessor that most easily supports dynamic memory. Th · eatures can be useful but they do have some limitations that must be considered. First of all, the ·· · H does not insure refresh during DMA situations. In a case where an external source has been granted the bus, it is the responsibility of the external source to refresh the memory. Secondly, the refresh address is only 7 bits which will not refresh 256 cycle, 64K dynamic RAMs. For these reasons, in some cases it is desirable to provide refresh timing and refresh address independent of the Z-80 microprocessor.

The same design steps mentioned in the design criteria were used to evaluate the compatibility of the Z-80 Microprocessor to the TMS 4500A. An example of a Z-80 interface to the TMS 4500A is shown in Figure 14. A15 is used to select the upper 32K bytes of memory configured with 16 4116's. Using $\overline{RFSH}$ signal generated by the Z-80 makes the refresh totally transparent in doing a refresh after every fetch cycle. If the refresh is not required by the Z-80 (DMA transfers), the 4500 will guarantee refresh. This procedure does refresh the memory at a higher rate than is needed which will increase the power dissipation of the memory array. If power dissipation is a major concern, the refresh can be handled completely by the controller (wait states could occur).

The circuit in Figure 14 will work up to the maximum speed of the Z-80 microprocessor (2.5 MHz) with -15 or -20 RAMs. The parameter that does not allow -25's to be used is the row address hold time. The $\overline{RD}$ signal can occur simultaneously with the $\overline{MREQ}$ signal which would result in a row address hold time of 25 ns (-25's require 35 ns).

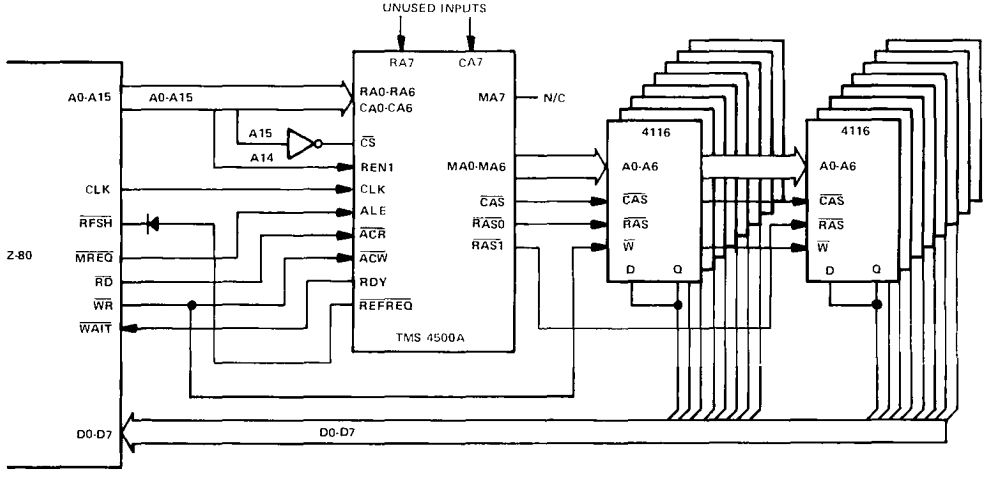


FIGURE 14 — Z-80 INTERFACE TO TMS 4500A CONTROLLER

If -25's are to be used, the circuit of Figure 15 is necessary. The flip-flop added to the RD path is necessary to delay RD long enough to satisfy -25 row address hold time. The addition of this circuitry enables -25's to be used at 2.5 MHz.

Even greater performance can be acheived if the Z-80A microprocessor is used. To find the limitations of the circuit in Figure 14 with the Z-80A, the design criteria must again be examined. There are two specifications that will limit operation at higher speeds. memory precharge time and access time from CAS

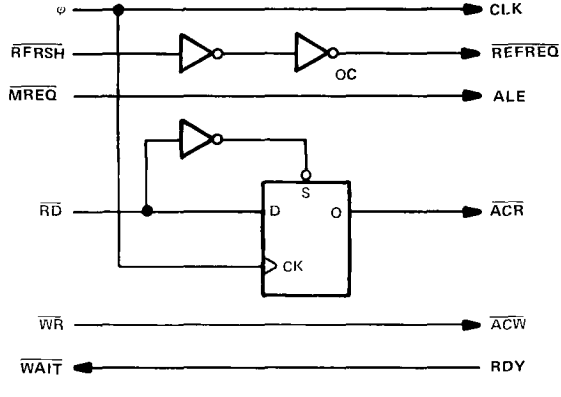


FIGURE 15 — Z-80 INTERFACE TO TMS 4500A ALLOWING 4116-25 MEMORIES

Precharge time on access cycles with the Z-80A is given by:

$$t_{RP} = t_{W(\overline{MRH})} + t_{AEL\text{-}REX} - t_{AEH\text{-}REX} - t_{t(REH)}$$

where $t_{W(\overline{MRH})} = t_{W(\emptyset H)} + t_f - 20$ ns (Z-80 Spec.)
$t_{W(\emptyset H)} = 110$ ns (MIN. Z-80A Spec.)
$t_f = 30$ ns
$t_{W(\overline{MRH})} = (110 + 30 - 20)$ ns $= 120$ ns
$t_{AEL\text{-}REX} - t_{AEH\text{-}REX} = -5$ ns
$t_{t(REH)} = 25$ ns

Therefore, $t_{RP} = (120 - 5 - 25)$ ns $= 90$ ns

The previous analysis shows that the -15 precharge time (100 ns) is not satisfied when operating at 4MHz. If the restriction is made that the duty cycle of the clock is at least 40%, the maximum frequency allowable to meet the required precharge time for -15 devices can be determined.

As $t_{RP} = (t_{W(\emptyset H)} + t_f - 20\text{ ns}) + t_{AEL\text{-}REX} - t_{AEH\text{-}REX} - t_{t(REH)}$

then $t_{RP} = t_{W(\emptyset H)} - 20$ ns

and $t_{W(\emptyset H)} = .4t_{CYCLE}$
$t_{RP} = 100$ ns (for -15s)

thus $100\text{ ns} = .4t_{CYCLE} - 20$ ns
$.4\ t_{CYCLE} = 120$ ns
Therefore, $t_{CYCLE} = 300$ ns ($f = 3.3$ MHz for -15s)

In a similar manner, the maximum frequency allowable to meet the required precharge time for -20 devices can be determined.

As $t_{RP} = t_{W(\emptyset H)} - 20$ ns

and $t_{W(\emptyset H)} = .4\ t_{CYCLE}$
$t_{RP} = 120$ ns (for -20s)

thus $120\text{ ns} = .4t_{CYCLE} - 20$ ns
$.4\ t_{CYCLE} = 140$ ns
Therefore, $t_{CYCLE} = 350$ ns ($f = 2.9$ MHz for -20s)

The second factor that limits high speed operation is the access time from $\overline{CAS}$. Instruction fetch cycles (M1) are the worst case when examining access time. The time from $\overline{ACR}$ to when data must be valid (t_{CD}) for the circuit in Figure 14 can be determined as follows:

$$t_{CD} = 3/2\ T - t_{S(D)} - t_{DL(RD)}$$

where $t_{S(D)} = 35$ ns (Z-80A Spec.)
$t_{DL(RD)} = 95$ ns (Z-80A Spec.)

thus $t_{CD} = 3/2T - 130$ ns

Next, subtract the delay from $\overline{ACX}$ to $\overline{CAS}$ to obtain the access time from $\overline{CAS}$.

$$t_{a(C)} = t_{CD} - t_{AEL\text{-}CEL} - t_{t(CEL)}$$

where $t_{AEL-CEL} + t_{t(CEL)} = 215$ ns (4500A Spec.)
$t_{CD} = 3/2T - 130$ ns

thus $t_{a(C)} = 3/2T - 130\text{ ns} - 215\text{ ns}$
Therefore, $t_{a(C)} = 3/2T - 345$ ns

Finally, determine the access time from $\overline{CAS}$ for the devices being considered (-15 and -25).

For -15s 100 ns = 3/2T - 345 ns
3/2T = 445 ns
$\therefore$ T = 297 ns (f = 3.37 MHz for -15s)

For -20s. 135 ns = 3/2T - 345 ns
3/2T = 480 ns
$\therefore$ T = 320 ns (f = 3.125 MHz for -20s)

The -15 devices are limited to 3.37 MHz and the -20 are limited to 3.125 MHz

Now that frequency limits have been determined for memory precharge time and access time from $\overline{CAS}$ for -15 and -20 devices, select the more restrictive frequency for each speed range of the device. In this example, the access time from $\overline{CAS}$ determines the frequency limit on both devices. The -15 will operate up to 3.3 MHz and the -20 will operate up to 2.9 MHz. The previous analysis should make it apparent that if the Z-80A is to be operated at 4MHz, additional circuity will be required (See Figure 16).

The circuit in Figure 16 allows the DRAM controller to operate at 4 MHz with – 15 memories. The higher speed is possible because the clock to the 4500A is twice as fast as that to the Z-80. This allows the refresh to be completed by the end of the refresh half of a fetch cycle. Although the 4500A takes four clock cycles to complete the refresh, the Z-80 sees only two clock cycles during that time and transparent refresh is maintained. $\overline{REFREQ}$ is gated to $\overline{CS}$ to deselect the controller during refresh since $\overline{MREQ}$ is active during this time and would initiate an access to the refresh address unless some form of gating is used to prevent access during refresh. The circuit in Figure 16 works up to 4 MHz (limited by microprocessor) with —15's and —20's. The timing diagram for this system is provided in Figure 17.

A summary of maximum operating frequency for the Z-80 and Z-80A versus various speed 4116's and circuit configurations is given in Table 2.

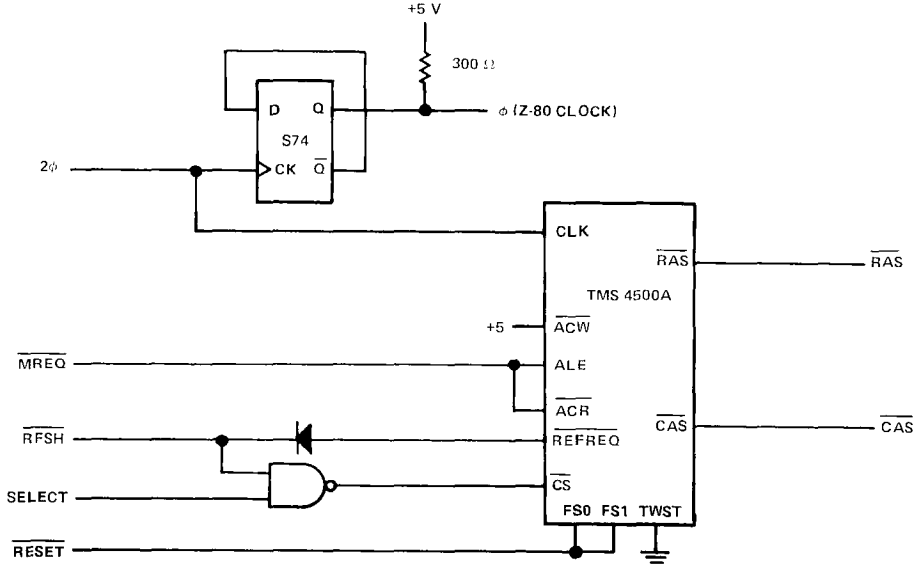


FIGURE 16 – INTERFACE FOR 4 MHz Z-80

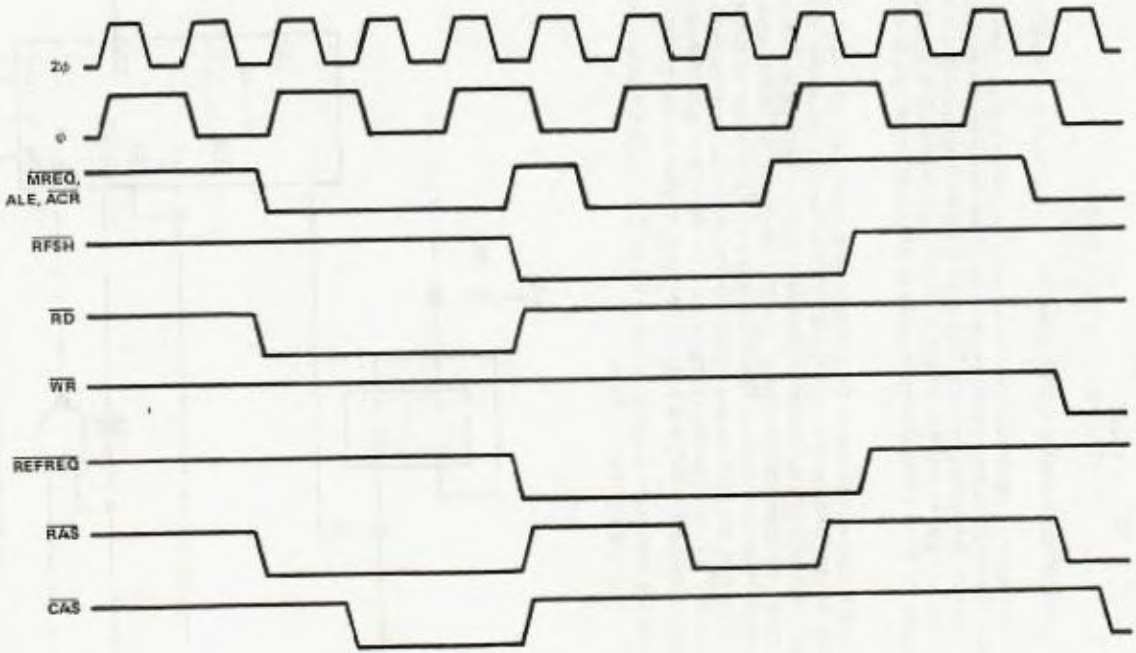


FIGURE 17 — TIMING DIAGRAM FOR 4MHz Z-80

TABLE 2 — SUMMARY OF Z-80 OPERATION

μP	4116	Maximum Operating Frequency (MHz)	
		Figure 14	Figure 15
Z-80	-15	2.5	2.5
	-20	2.5	2.5
	-25	—	2.5
		Figure 14	Figure 16
Z-80A	-15	3.3	4.0
	-20	2.9	4.0
	-25	—	3.0

5.4 TMS 9900 EXAMPLE

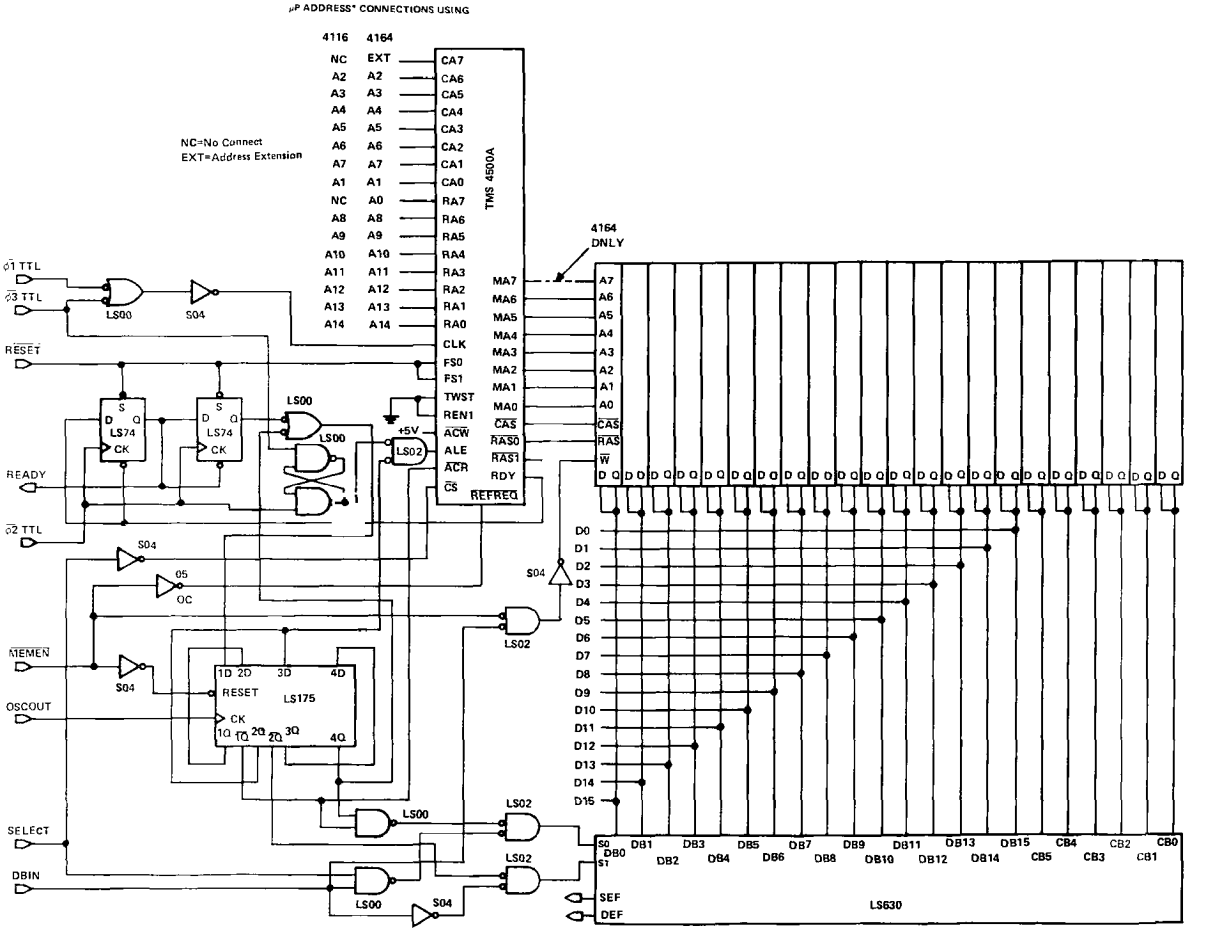
A TMS 9900 microprocessor interface to the TMS 4500A with error detection and correction is shown in Figure 18. This circuit which uses the SN74LS630 EDAC chip will operate up to 3 MHz with -20 or -15 dynamic RAMs. The timing diagram helps to illustrate how the circuit works (See Figure 19). ALE is generated by $\overline{\phi 2}$ TTL and 2Q of the 74LS175 (The 74LS175 is wired to act as a shift register). $\overline{ACR}$ is driven by the 1Q output of the shifter. These two signals will both go low at approximately the same time to start a memory cycle.

This interface makes the refresh totally transparent to the microprocessor by using $\overline{MEMEN}$ to request refresh cycles. $\overline{MEMEN}$ will always stay high for at least 2 CLK cycles (4 controller cycles) and will never go low for longer than 3 memory cycles. This insures that under microprocessor control, refresh cycles will always have enough time to be completed and will always occur often enough to satisfy the minimum refresh rate.

The only instance when wait states could occur is if the microprocessor entered a hold condition and no external source drives $\overline{MEMEN}$. This would leave the refresh responsibility on the DRAM controller. If the microprocessor returned from the hold state (while refresh is in progress) and requested a memory cycle, the RDY line would go low and insert a wait state in the first memory cycle. The RDY line from the 4500A must be delayed since the 9900 samples the RDY line during $\overline{\phi 1}$ high time.

The S0 and S1 signals generate the proper timing for the 74LS630 EDAC chip. Whenever both S0 and S1 are low, the EDAC chip will generate check bits to reflect the data on the 16 data bits. On a write cycle, the data must be valid from the microprocessor and a delay for the generation of the check bits must take place before the $\overline{CAS}$ line (early write) is allowed to go low and latch data into the RAMs.

On a read cycle, a delay must be added to allow the EDAC chip to detect and correct any errors that have occurred. This process is sequenced by S0 and S1. When S0 goes high, the 630 will read data and check bits from the memory array. When S1 is switched from high to low the data is latched on the 630 and data is interrogated for errors. If a single bit error is found, the error will be corrected and single error flag (SEF) set. If a double bit error is found, the 74LS630 will set the double error flag (DEF) and not attempt error correction. When S0 goes low the data will be output to the data bus. All of the error detection and correction takes place within the CPU memory cycle without wait states.



* Note: The TMS 9900 uses the convention that A0-A14 are the most significant to least significant bits of the address.

FIGURE 18 — TMS 9900 TO TMS 4500A INTERFACE

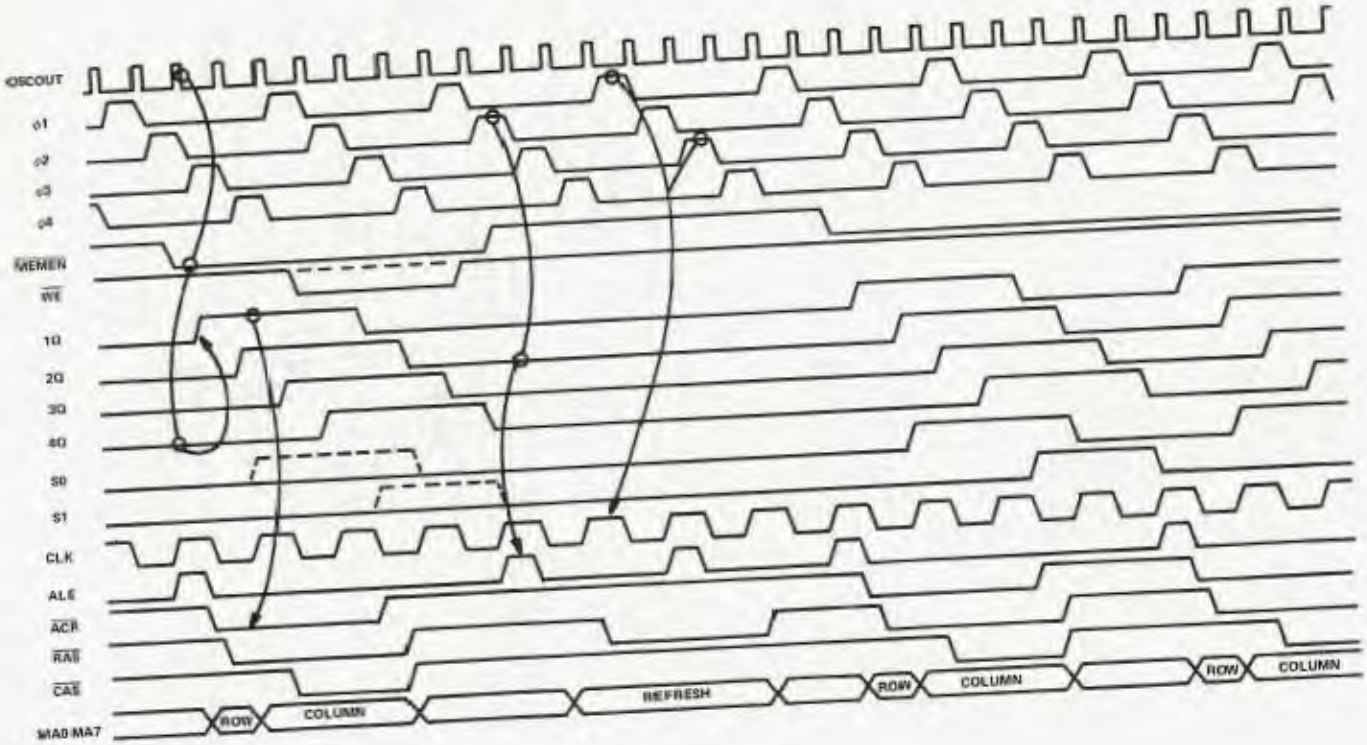


FIGURE 19 — TIMING DIAGRAM FOR TMS 9900 INTERFACE

5.5 MC68000, 8086, Z-8001, AND TMS 9995 EXAMPLES

Several application examples illustrating the interface between the TMS 4500A and several other popular 16-bit microprocessors (MC68000, 8086, Z-8001, and TMS 9995) are given in this section. Diagrams showing the required interface and a timing diagram is provided for each example. The maximum speed achievable in each case can be calculated using the design steps previously listed (Refer to Section 3). These examples show the versatility of the TMS 4500A dynamic memory controller.

5.5.1 MC68000 Example

The interfaces between the MC68000 microprocessor and the TMS 4500A memory controller is shown in Figure 20. This configuration uses the $\overline{CAS}$ output from the 4500A to signal data transfer acknowledge ($\overline{DTACK}$) to the 68000. This eliminates the need for the RDY line from the 4500 since the microprocessor will wait for $\overline{CAS}$ before completing the memory cycle. This also enables the 68000 to perform read/modify/write bus cycles on the TAS instruction. The timing diagrams are given in Figure 21.

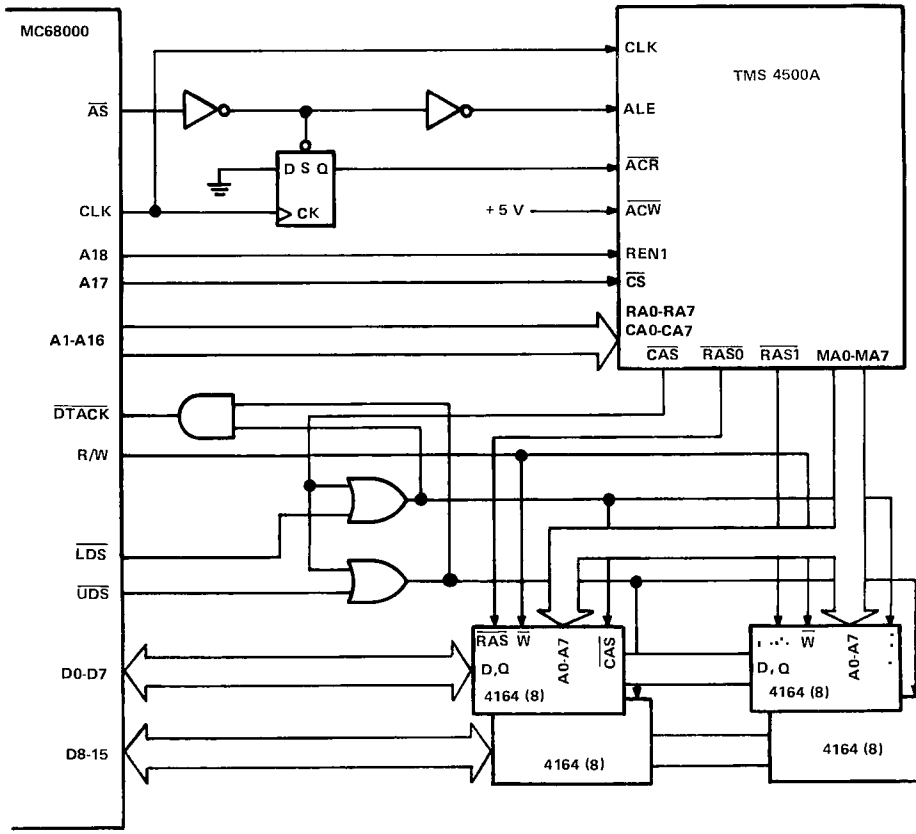
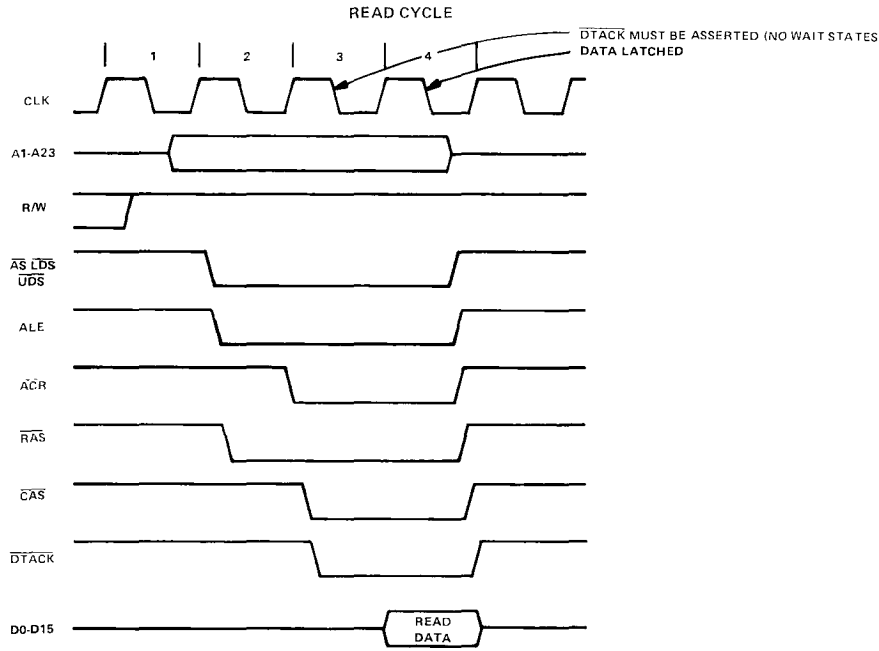


FIGURE 20 — MC68000 TO TMS 4500A 128K x 16 MEMORY INTERFACE



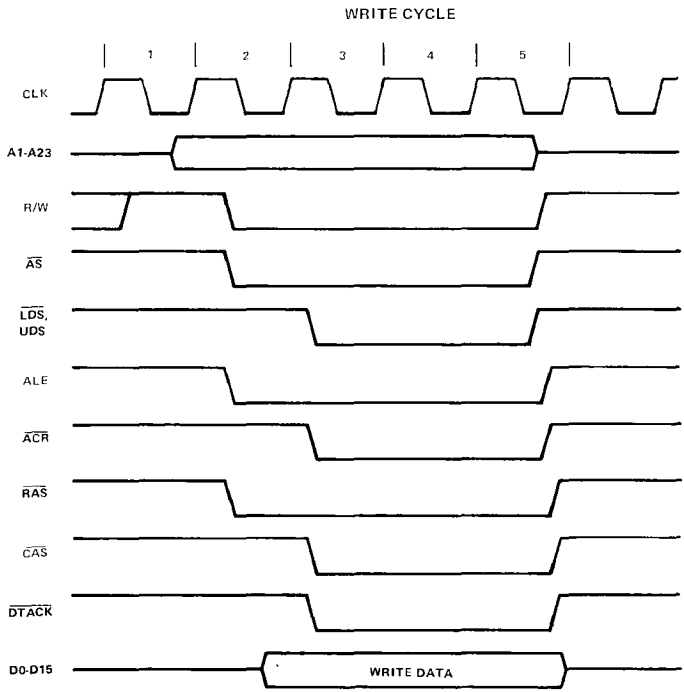


FIGURE 21 — MC68000 TIMING DIAGRAMS

READ/MODIFY/WRITE CYCLE

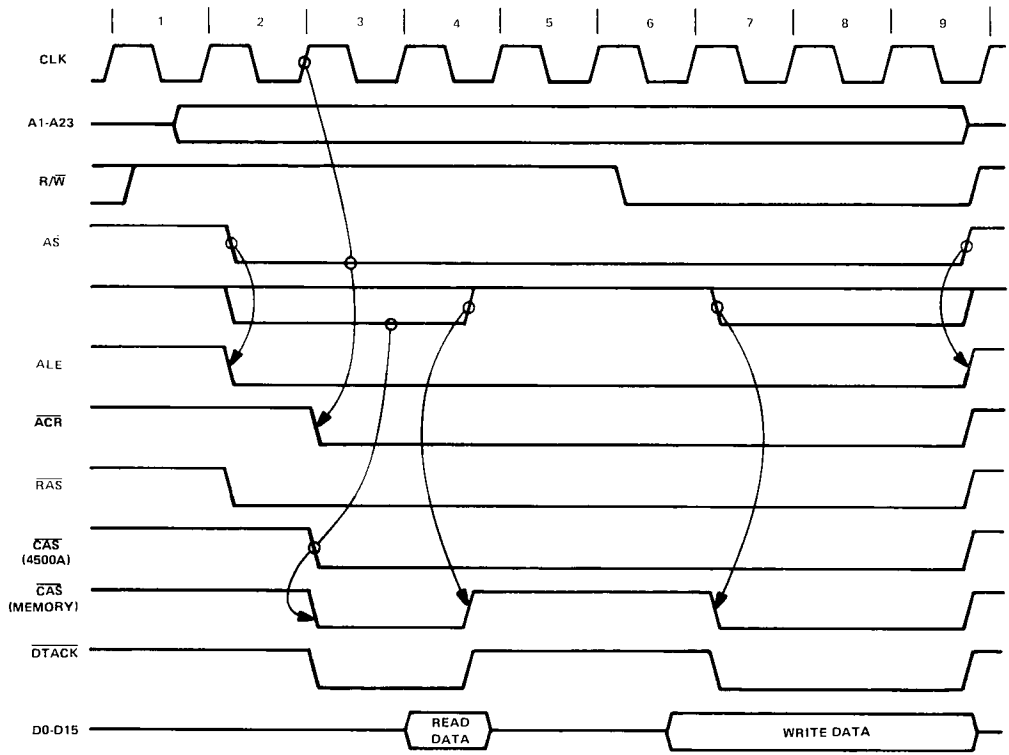


FIGURE 21 (Continued) – MC68000 TIMING DIAGRAM

5.5.2 8086 Example

An example illustrating the interface between the 8086 microprocessor and the TMS 4500A memory controller is given in Figure 22. The 8086 requires the RDY to be delayed which can be accomplished by a D flip-flop in its path. The write signal for the upper byte and the lower byte must be generated by $\overline{BHE}$ and A0. This enables writes to only the upper or lower byte as desired. The timing diagram is given in Figure 23.

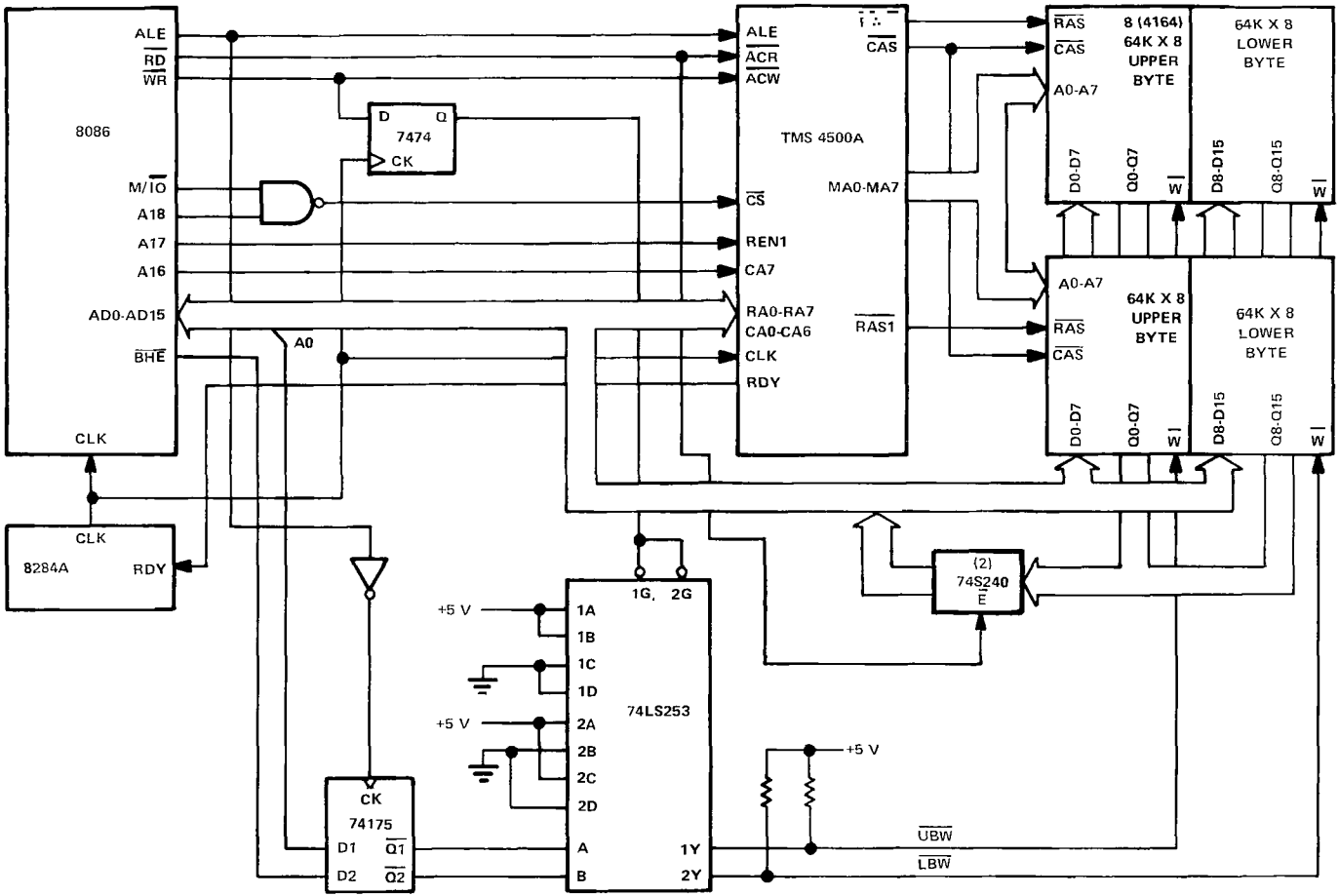


FIGURE 22 — 8086 TO TMS 4500A MEMORY INTERFACE

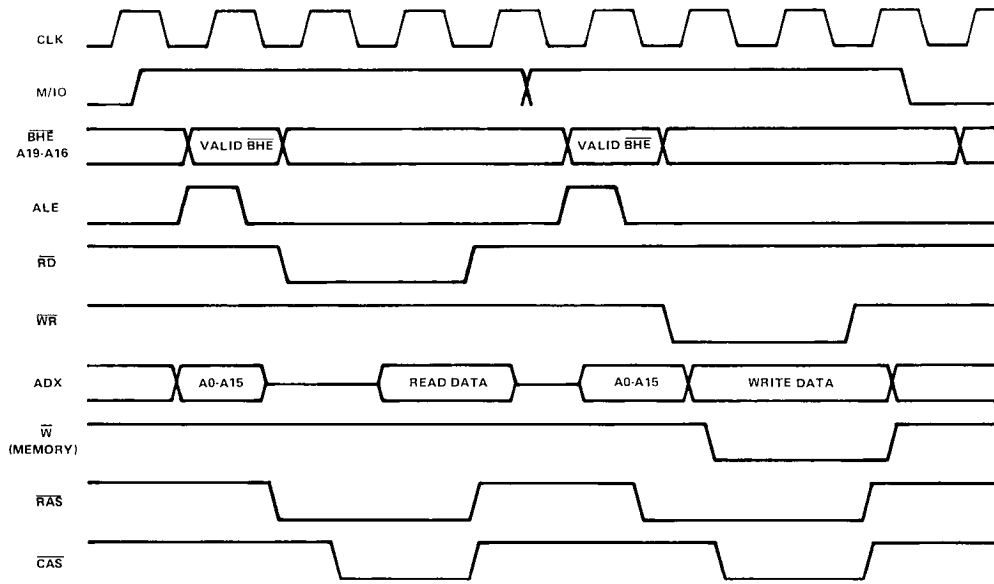


FIGURE 23 – 8086 TIMING DIAGRAM

5.5.3 Z-8001 Example

A system using the Z-8001 microprocessor and the two TMS 4500A memory controllers is given in Figure 24. This system uses a separate controller for the upper and lower bytes. Using two controllers reduces the transition times of address and control lines by reducing the load on each line. The upper or lower byte control (or both) is latched by the $\overline{CS}$ on the 4500A. A single controller could be used if the upper and lower byte con· were latched and gated with the $\overline{CAS}$ outputs. This figure also shows the use of a diode to drive the · $\overline{REQ}$ line on the controller. This could take the place of an open collector device. The timing diagram for this system is given in Figure 25.

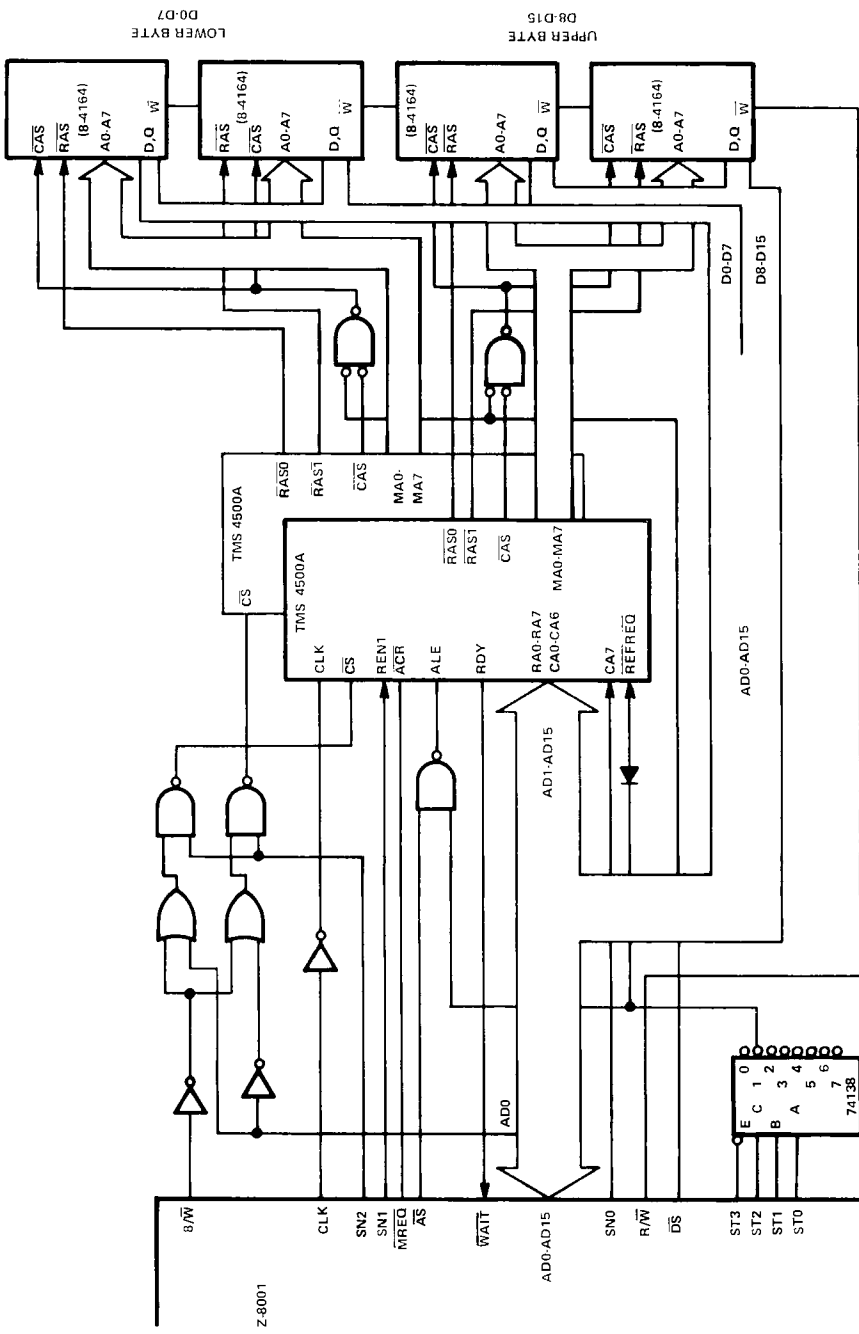


FIGURE 24 — Z-8001 TO TMS 4500A MEMORY INTERFACE

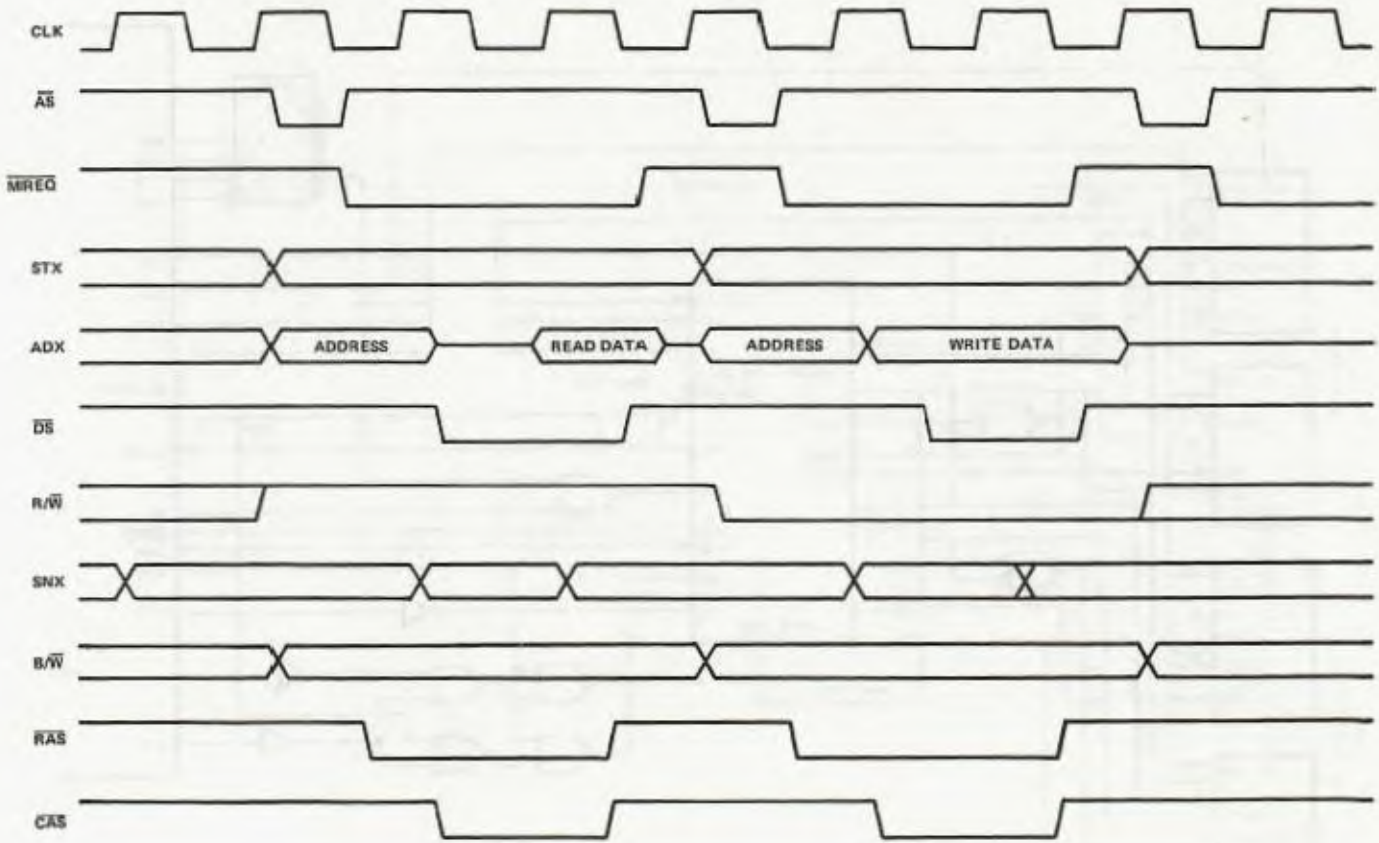


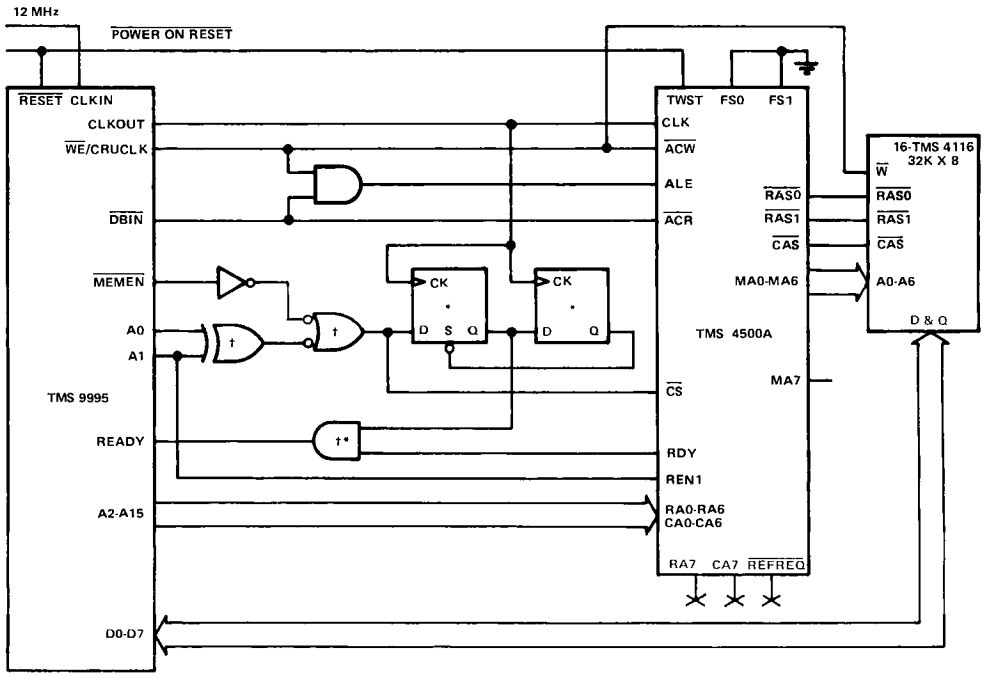
FIGURE 25 — Z-8001 TIMING DIAGRAM

5.5.4 TMS 9995 Example

The next example shows the 4500A configured for operation with the TMS 9995 (See Figure 26). The 9995 has an internal, on-chip memory and so care must be taken not to map any external memory into the internal memory space. Since the internal memory is mapped into the address space above F000$_H$ and the lower memory space is generally used for reset, interrupt, and special use vectors, the middle 32K of memory space has been chosen for the external DRAM. This is accomplished by XORing the two most significant address bits, A0 and A1. (It should be noted here that the TMS 9995 uses the convention that for both address and data lines, the "0" bit is the most significant). This output from the XOR is gated with $\overline{ME}$ to produce $\overline{CS}$ for the 4500A so that the DRAM will be selected when addressing memory locations 4000$_H$ through BFFF$_H$.

Another item of interest is the circuitry shown with asterisks. This circuitry is used to speed up the RDY signal that is to be presented to the TMS 9995. A maximum delay of −3 ns from $\overline{DBIN}$ low to setup is required with CLKIN = 12 MHz. This precludes using the RDY signal from the 4500A since is used to initiate the DRAM cycle. In order to provide RDY to the 9995, the logic shown with asterisks checks the $\overline{CS}$ line to see if DRAM is being selected and clocks a low signal to the READY line of the TMS 9995. If the 4500A is in a refresh cycle, the RDY output will hold the processor for subsequent cycles. The clocked logic thus inserts one wait state on all DRAM accesses. The parts indicated must be Schottky to obtain the shortest propagation delays. The circuitry shown with asterisks can be eliminated completely if the automatic wait state feature of the TMS 9995 is used; however this precludes taking advantage of faster memories that may be used elsewhere in the memory space. This circuit may not be required if the TMS9995 is run with a slower CLKIN frequency.

A timing diagram for the TMS 9995 interface is given in Figure 27.



* These devices must be added if automatic wait state feature of TMS 9995 is not used.
† These devices must be Schottky if automatic wait state feature of TMS 9995 is not used.

FIGURE 26 — TMS 9995 TO TMS 4500A MEMORY INTERFACE

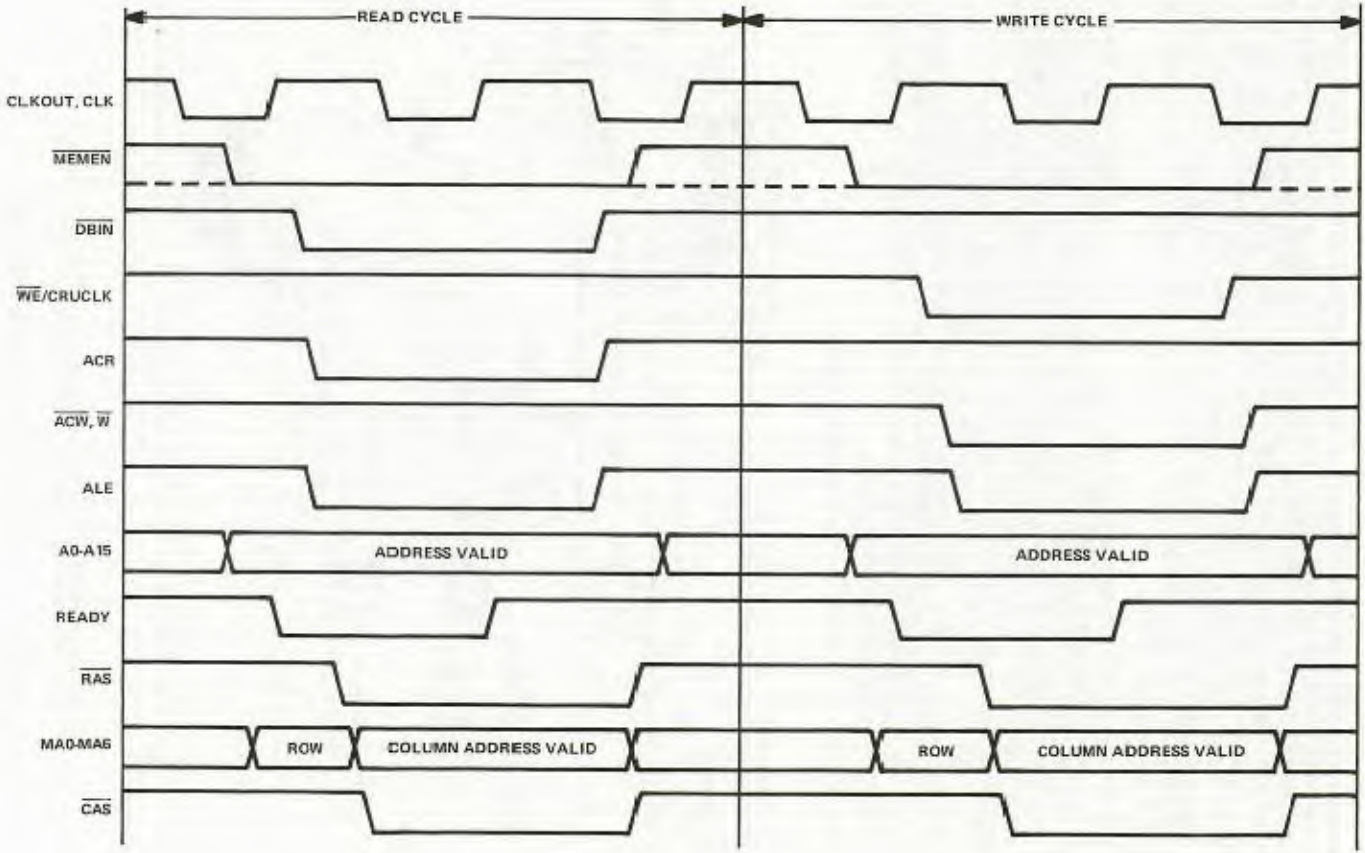


FIGURE 27 — TMS 9995 TO TMS 4500A TIMING DIAGRAM

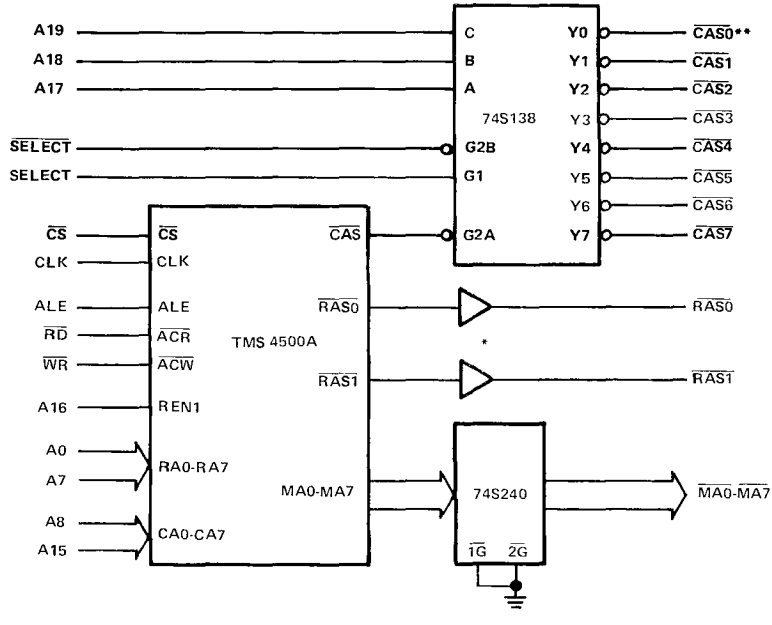
5.6 MEMORY BANK EXPANSION EXAMPLE

In many applications, the user may want to drive more than two banks of memory with one TMS 4500A DRAM controller. This can be easily accomplished as demonstrated in the following illustrations. The primary considerations are additional timing delays and loading of the controller signals.

Figure 28 shows one method that is probably the most cost effective. The higher order address lines are needed to select which of eight $\overline{CAS}$ strobes will be activated and then each $\overline{CAS}$ strobe goes to two banks of memory (one on $\overline{RAS0}$ and the other on $\overline{RAS1}$). This scheme can drive up to 16 banks of 64K DRAMS (1M words). This looks excellent on first examination; however, there is one major criticism to this approach: the system power requirements are increased significantly because of the activation of half of the banks on every access. This is due to the fact that $\overline{RAS}$ is used to select the device and $\overline{CAS}$ acts as an output enable. The design given in Figure 28 is therefore a power consuming implementation.

Figure 29 shows two possible designs using decoders. The first uses the activation of $\overline{RAS1}$ to provide the refresh to all banks (since with REN1 strapped low the only time $\overline{RAS1}$ is active is during refresh). The second design uses the fact that both $\overline{RAS0}$ and $\overline{RAS1}$ are active low during refresh to activate the RAS outputs to the memory array.

Figure 30 shows the implementation with a TBP 18S030 (32 x 8) PROM. This design again uses $\overline{RAS1}$ active to indicate refresh and $\overline{RAS0}$ to indicate an access cycle. As drawn each of the four $\overline{CAS}$ signals would drive two banks of memory. The program for the TBP 185030 PROM is given in Table 3.



* RAS signals should be appropriately buffered according to loading.

** Each CAS output drives two banks (one on $\overline{RAS0}$ and the other on $\overline{RAS1}$). Although this may be a minimum parts solution it has the disadvantage of high power dissipation. This is caused by having each $\overline{RAS}$ signal drive half the array (since maximum power occurs due to $\overline{RAS}$ transitions rather than $\overline{CAS}$ transitions.

FIGURE 28 — MULTIPLE BANK IMPLEMENTATION WITH 74S138
1 OF 8 DECORDER (16 POSSIBLE BANKS)

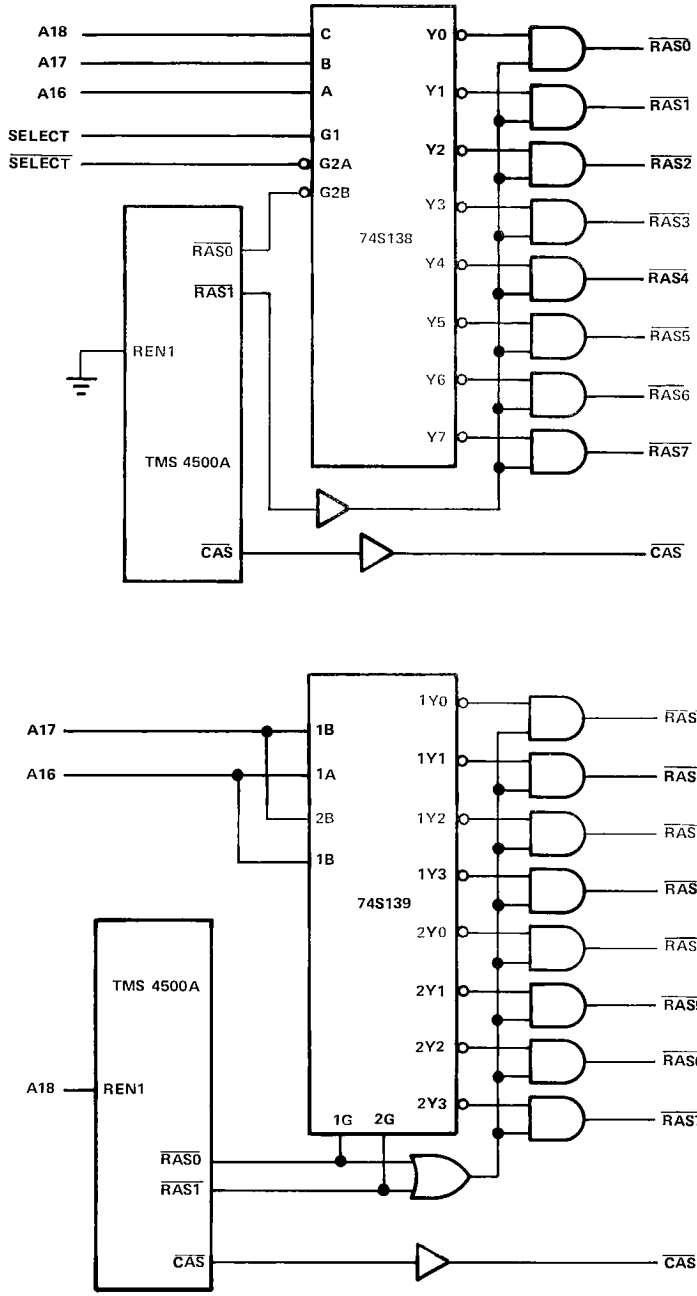
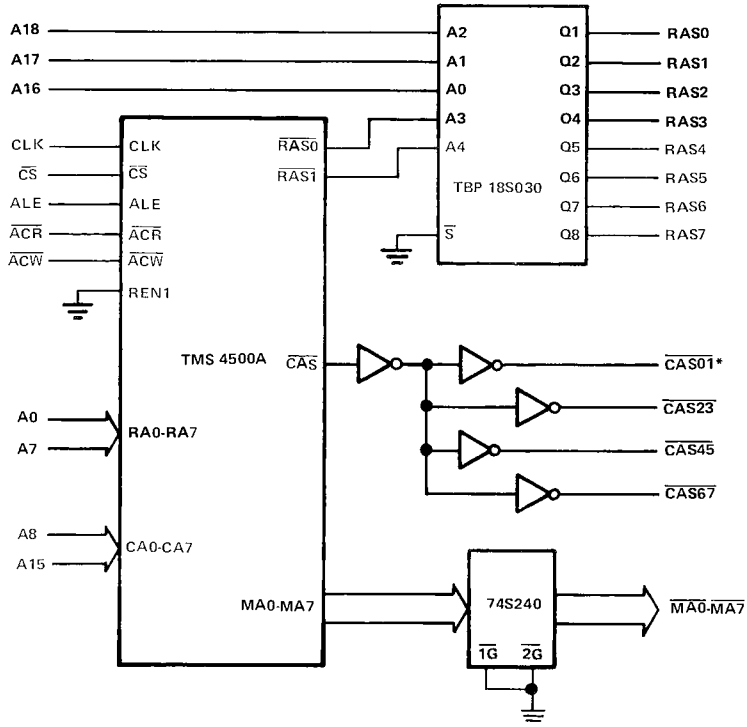


FIGURE 29 — POSSIBLE 8 BANK MEMORY IMPLEMENTATION



* The number of generated $\overline{CAS}$ signals may be increased depending on loading.

FIGURE 30 – POSSIBLE MULTIPLE BANK IMPLEMENTATION WITH TBP 18S030 PROM (8 BANKS)

TABLE 3 — PROGRAM FOR THE TBP 18S030 (32 x 8) PROM

	ADDRESS					OUTPUTS								PROGRAM-MED DATA	
$\overline{S}$	A4	A3	A2	A1	A0	Q8	Q7	Q6	Q5	Q4	Q3	Q2	Q1		
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	1	1	1	1	1	1	1	F	F
0	0	1	0	0	1	1	1	1	1	1	1	1	1	F	F
0	0	1	0	1	0	1	1	1	1	1	1	1	1	F	F
0	0	1	0	1	1	1	1	1	1	1	1	1	1	F	F
0	0	1	1	0	0	1	1	1	1	1	1	1	1	F	F
0	0	1	1	0	1	1	1	1	1	1	1	1	1	F	F
0	0	1	1	1	0	1	1	1	1	1	1	1	1	F	F
0	0	1	1	1	1	1	1	1	1	1	1	1	1	F	F
0	1	0	0	0	0	1	1	1	1	1	1	1	0	F	E
0	1	0	0	0	1	1	1	1	1	1	1	0	1	F	D
0	1	0	0	1	0	1	1	1	1	1	0	1	1	F	B
0	1	0	0	1	1	1	1	1	1	0	1	1	1	F	7
0	1	0	1	0	0	1	1	1	0	1	1	1	1	E	F
0	1	0	1	0	1	1	1	0	1	1	1	1	1	D	F
0	1	0	1	1	0	1	0	1	1	1	1	1	1	B	F
0	1	0	1	1	1	0	1	1	1	1	1	1	1	7	F
0	1	1	0	0	0	1	1	1	1	1	1	1	1	F	F
0	1	1	0	0	1	1	1	1	1	1	1	1	1	F	F
0	1	1	0	1	0	1	1	1	1	1	1	1	1	F	F
0	1	1	0	1	1	1	1	1	1	1	1	1	1	F	F
0	1	1	1	0	0	1	1	1	1	1	1	1	1	F	F
0	1	1	1	0	1	1	1	1	1	1	1	1	1	F	F
0	1	1	1	1	0	1	1	1	1	1	1	1	1	F	F
0	1	1	1	1	1	1	1	1	1	1	1	1	1	F	F
1	X	X	X	X	X	Z	Z	Z	Z	Z	Z	Z	Z		

The last example illustrates some of the flexibility that may be achieved using the TMS 4500A DRAM controller. An application required 128K of RAM in a system with a 16K EPROM such that the 16K EPROM should always be accessable as well as 1 bank of 16K RAM, the other banks (16K each) were to be paged into the remaining 32K address space of the processor.

The following memory map (See Figure 31) was devised where A16 and A17 were high order addresses that were selected by writing to an I/O port.

A17	A16	A15	A14	SELECTED MEMORY
0	0	0	0	EPROM
0	0	0	1	RAM0
0	0	1	0	RAM4
0	0	1	1	RAM1
0	1	0	0	EPROM
0	1	0	1	RAM0
0	1	1	0	RAM2
0	1	1	1	.
1	0	0	0	E . ..
1	0	0	1	RAM0
1	0	1	0	RAM4
1	0	1	1	RAM5
1	1	0	0	EPROM
1	1	0	1	RAM0
1	1	1	0	RAM6
1	1	1	1	RAM7

NOTE: RAM X is the particular 16K byte of RAM to be active in that address region

FIGURE 31 — MEMORY MAP

One possible solution involves using 8 banks of 16K DRAM devices and decoding the select logic to enable the correct bank. Another possibility is to use 2 sets of 64K DRAM each containing 4 of the required banks. Figure 32 shows a possible implementation that also indicates some of the flexibility that may be achieved when using the TMS 4500A DRAM Controller.

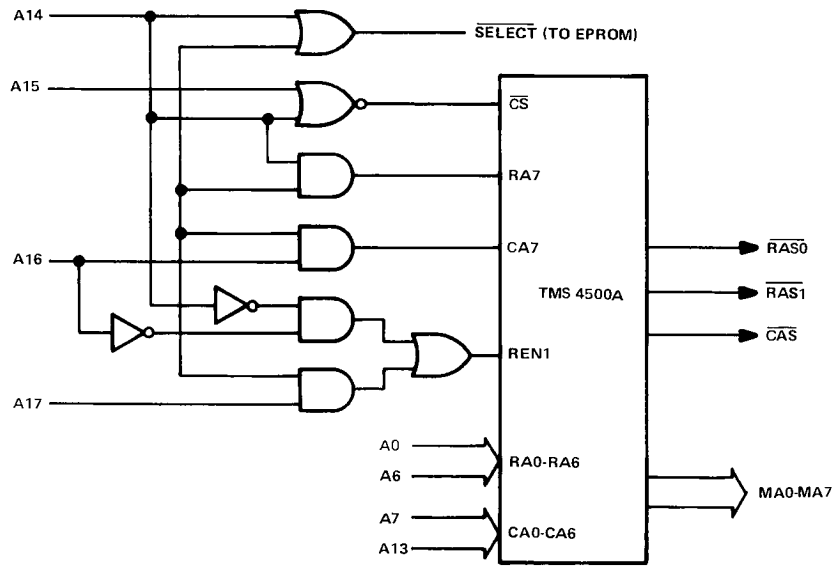


FIGURE 32 — BANK EXPANSION EXAMPLE SHOWING TMS 4500A FLEXIBILITY