CONTROL AND REFRESH OF MADRAMs

National Semiconductor
Memory Application
Mike Evans *
May 1978

The name MADRAM refers to any multiplexed address dynamic random access memory, such as the MM5290.

A MADRAM controller must have:

- Address multiplexing facility common to all MADRAMs
- $\overline{RAS}$ to the selected bank only (rather than $\overline{CAS}$ to save power)
- $\overline{CAS}$ common to all MADRAMs
- $\overline{WE}$ common to all MADRAMs
- Refresh of all rows with $\overline{RAS}$ every refresh period

Also it would be desirable to have:

- Page mode capability
- Read modify write capability
- Ability to transparently refresh
- Ability to distributive refresh
- Ability to burst refresh, with circuitry to ensure that refreshing of all rows occurs
- Ability to initiate MADRAM with 8 refresh cycles after power switch on
- To operate in conjunction with microprocessors with minimum extra IC packages
- Operation with DMA
- Operation in large systems
- Adaptability for all speeds of MADRAMs
- Adaptability to 4K, 16K, and 64K MADRAMs

General Purpose Controller for all MADRAMs

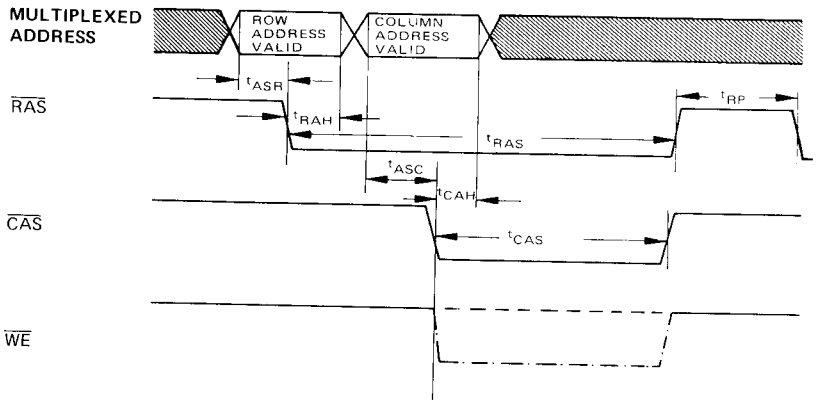
Most control circuits contain Johnson counters to sequentially produce in order, $\overline{RAS}$, ROW to COLUMN ENABLE, then $\overline{CAS}$, all from processor READ/WRITE signals. There are a number of limitations with this method, first it is much more frequency limited due to all the interrelationships imposed by the RAM timing specifications, and second it is not very versatile. For example, page mode operation is difficult with $\overline{RAS}$ low continuously, while $\overline{CAS}$ is pulsed for each new column address. Read-modify-write is likewise difficult.

The circuit of Figure 1 was designed to satisfy most of the requirements listed for a MADRAM controller and by using it in various applications only one extra SSI chip is required to interface between the processor and controller.

Memory Access

The $\overline{MEMORY\ ACCESS}$ input goes low to access the RAM, and for normal operation (non page mode) this must remain low only for the first negative going clock pulse of the clock input. From this negative going clock pulse, $\overline{RAS}$ commences on one of the selected four $\overline{RAS}$ output drives. (The row address will already be enabled on the multiplex address bus.)

The required MADRAM timing sequence is shown below:



* Refer to introduction

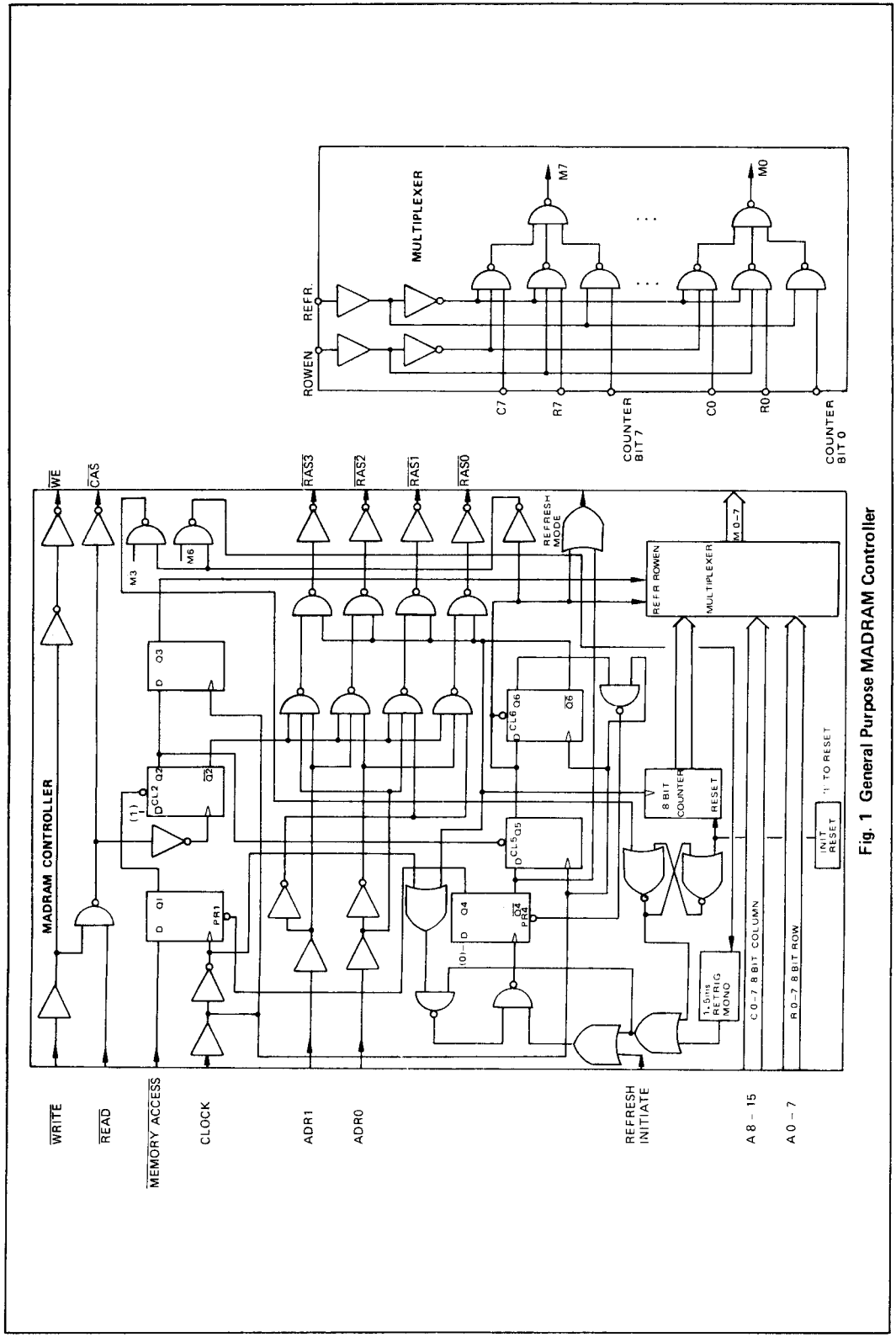
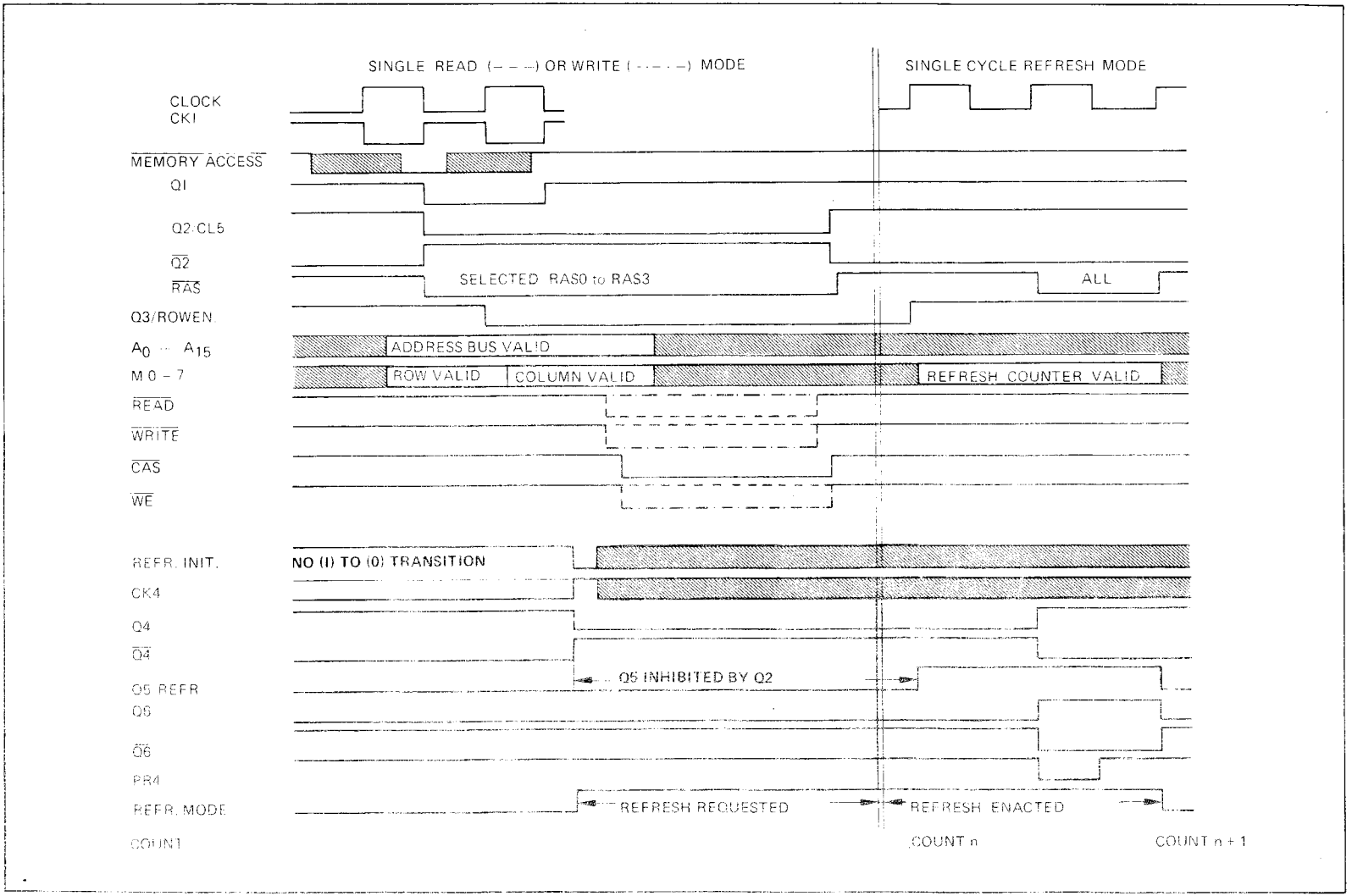


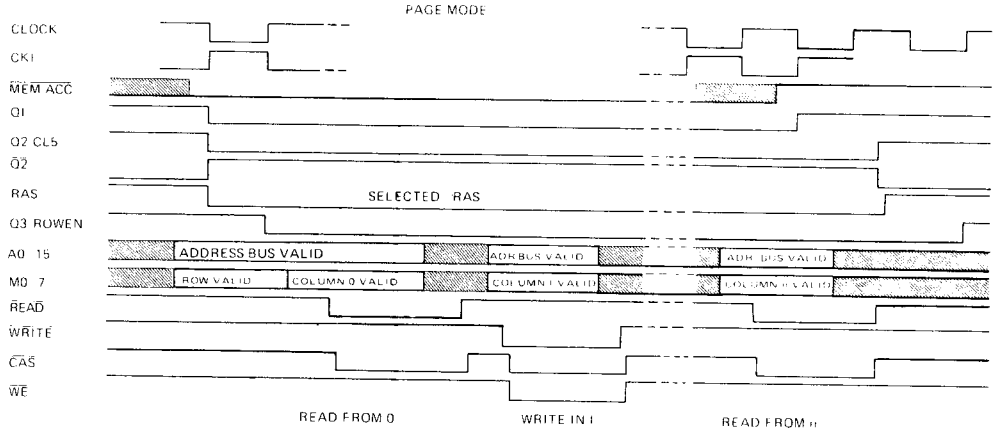
Fig. 1 General Purpose MADRAM Controller

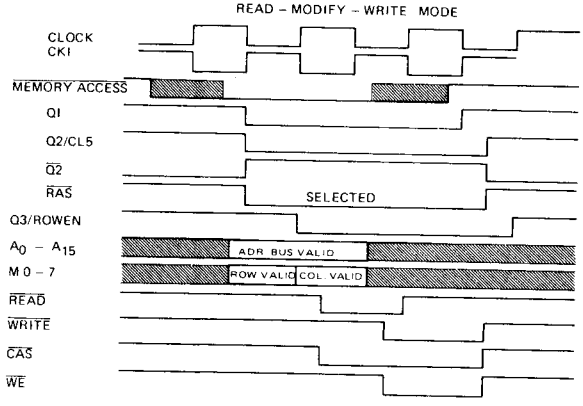

SINGLE READ (– – –) OR WRITE (– · – · –) MODE
SINGLE CYCLE REFRESH MODE
CLOCK
CK1
MEMORY ACCESS
Q1
Q2/CL5
Q2
RAS
SELECTED RAS0 to RAS3
ALL
Q3/ROWEN
A0 ··· A15
ADDRESS BUS VALID
M 0 – 7
ROW VALID
COLUMN VALID
REFRESH COUNTER VALID
READ
WRITE
CAS
WE
REFR. INIT.
NO (1) TO (0) TRANSITION
CK4
Q4
Q4
Q5 REFR
Q5 INHIBITED BY Q2
Q5
Q6
PR4
REFR. MODE
REFRESH REQUESTED
REFRESH ENACTED
COUNT
COUNT n
COUNT n + 1

On the first positive going edge of CLOCK after this, the column address will be enabled, and this has to occur before any $\overline{READ}$ or $\overline{WRITE}$ signal. Next the $\overline{READ}$ or $\overline{WRITE}$ signal outputs $\overline{CAS}$ to all RAMs, and also $\overline{WE}$ if the signal is $\overline{WRITE}$. This means that t_{ACC} after the $\overline{CAS}$ leading edge, the data will be valid for a read cycle, whereas with the Johnson counter method, an extra 2 clock pulses minimum are required between $\overline{READ}$ and $\overline{CAS}$ leading edges. This means a faster RAM may be necessary for that method. The bank selected by $\overline{RAS}$ is determined from the states of address inputs ADR0 and ADR1.

In page mode operation, if ADDRESS STROBE is kept low, the selected $\overline{RAS}$ remains low, and $\overline{CAS}$ occurs for every $\overline{READ}$ or $\overline{WRITE}$ input. Note that refreshing is inhibited at any time any $\overline{RAS}$ is low.

To perform READ–MODIFY–WRITE sequences, if a normal read operation is started, and then any time before $\overline{READ}$ trailing edge, $\overline{WRITE}$ leading edge begins, then selected $\overline{RAS}$ and $\overline{CAS}$ remain low, first for read, and then $\overline{WE}$ low.

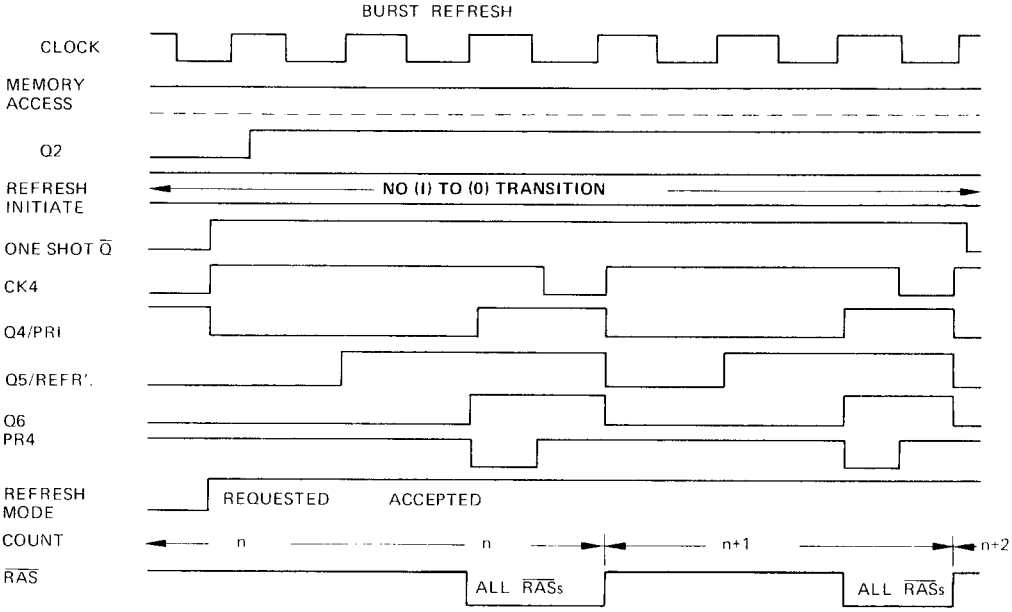




Refreshing

A negative going edge on REFRESH INITIATE causes one refresh cycle to begin. This can be for any type of refresh mode, i.e., transparent, distributive or burst modes. After the REFRESH INITIATE signal, the refresh cycle occupies the next two clock periods, during which time normal memory access is inhibited, and an 8 bit refresh counter is outputted on to the multiplexed address bus. An output, REFRESH MODE, goes high from the commencement of REFRESH INITIATE until the cycle is completed, so that this output can be used to hold a processor during refresh as occurs during a distributive or burst refresh. At the end of each refresh cycle the 8 bit counter is incremented.

In the transparent mode, the REFRESH INITIATE occurs when the system is not accessing the RAM, for example, during an instruction fetch cycle from ROM/PROM, when a cycle steal is performed. As long as the refresh cycle finishes before the next memory access cycle, the refresh is transparent. Usually only part of one SSI chip is sufficient extra circuitry to output to REFRESH INITIATE that an instruction fetch cycle is beginning. So although REFRESH MODE goes high, the processor sequence is not delayed. It is necessary that 128 of these cycles must occur every 2 ms.



In distributive mode, a constant frequency faster than 128/2ms, i.e., 64 kHz, is applied to REFRESH INITIATE. If a read or write cycle is already in progress, refresh commences at the end of this cycle. Once refresh is initiated, a read or write cycle is inhibited. Because the processor is held by REFRESH MODE, a small precentage of time is lost, this being about 128 double clock periods every 2 ms.

This also applies for the burst mode when a burst of refresh cycles is made to occur if for example, page mode is continually being used, meaning other refresh modes are not possible. This burst mode facility is also provided internal to the controller circuitry. If there are less than 128 refresh cycles in each consecutive 1.7 ms, then a retriggerable one shot, triggered from bit 7 of the refresh counter, goes high, causing a burst of refresh cycles until bit 7 next triggers the one shot. Thus if only 100 refresh cycles occurred in the 1.7 ms since the one shot was last triggered, a burst of 28 refresh cycles will occur, and the MADRAM cannot be accessed by the system for this period.

There is also a facility to provide 8 consecutive refresh cycles immediately after power switch on as all MADRAMs need up to 8 cycles after switch on before providing valid information. The refresh counter and a simple RS flip flop are reset at switch on, and the refresh initiate signal is applied internally 8 times until the refresh counter bit 4 permanently sets the flip flop.

INS8060

Figure 2 shows an INS8060 (SC/MP) microprocessor system driving four banks of 4K MADRAMs. The first 16K bytes of memory addresses will be to ROMs/PROMs or I/Os. The three SC/MP outputs NWDS, NRDS and NADS can be connected directly to $\overline{\text{WRITE}}$, $\overline{\text{READ}}$, and $\overline{\text{MEMORY ACCESS}}$. Refresh can be transparent if NENOUT is connected to REFRESH INITIATE, because while NENOUT is low, SC/MP is not using the data or address busses. As the refresh cycle will be finished in less than three clock periods, a transparent refresh will have occurred because SC/MP will still not be ready to use the busses. As a precautionary measure, REFRESH MODE output can be connected to an open collector inverter whose output is wired-ORed to NBREQ and NENIN, to prevent SC/MP commencing a new cycle.

INS8080A

Figure 3 shows the 8080A and the controller in a transparent refresh mode, using the instruction fetch from ROM/PROM condition, so that the MADRAMs are refreshed one row during this period. If D5 and D7 of the data bus are NANDed with SYNC, then when all three are high an instruction fetch cycle is about to commence and REFRESH INITIATE goes low to start the refresh cycle, taking up to three clock periods. The 8080 may be just ready to commence another cycle. As a precaution, REFRESH MODE is connected to HOLD of the 8080 in case a burst mode occurs for any reason. Note that if distributed mode is preferred, an oscillator (or clock divider) is connected to REFRESH INITIATE, and its frequency must be faster than 64 kHz. Three banks of MADRAM are shown. Note that $\overline{\text{WR}}$ is connected directly to $\overline{\text{WRITE}}$, DBIN is inverted to $\overline{\text{READ}}$, and SYNC is inverted to $\overline{\text{MEMORY ACCESS}}$.

8085A

With this microprocessor it is necessary to set $\overline{\text{MEMORY ACCESS}}$ high at least one negative going clock pulse before the trailing edge of $\overline{\text{CAS}}$. Thus if a flip flop, whose output is connected to $\overline{\text{MEMORY ACCESS}}$, is triggered low from the leading edge of ALE and then set high by $\overline{\text{CAS}}$, the 8085 will also interface with this controller, as shown in Figure 4. $\overline{\text{WE}}$ and $\overline{\text{RD}}$ can be directly connected to $\overline{\text{WRITE}}$ and $\overline{\text{READ}}$, and CKOUT inverted to CLOCK. Instruction fetch cycle stealing is not so easy but can be done as shown; if not, use distributive refresh. Note that it is not necessary to latch AD0–7 from the data bus.

16 Bit Microprocessors

The new 16 bit microprocessors like the 8086 are likely to be used with the 64K MADRAM because 20 address bits are available with this microprocessor, see Figure 5. Each group of four banks can be selected by connecting A16 to ADR0 and A17 to ADR1 using A18 and A19 to control $\overline{\text{MEMORY ACCESS}}$, because if $\overline{\text{MEMORY ACCESS}}$ remains high, so do all RAS's for that controller. With the 64K MADRAM, 256 rows need to be refreshed every 4 ms, so as long as 128 rows are refreshed every 2 ms, there is no need to alter the burst refresh circuitry. The RD, WE, CKOUT and ALE of the 8086 are similar to the 8085A.

DMA

The controller will also interface with a DMA as long as 128 rows are still refreshed every 2 ms, probably in a burst mode, but REFRESH MODE must inhibit both the DMA and the processor. Page mode control is accomplished by keeping $\overline{\text{MEMORY ACCESS}}$ low. Note that the Mostek 16K MADRAM has a t_{RAS} maximum of 10,000 ns, and with the 3 part the page mode cycle time of 225 ns means that only 36 consecutive cycles can be performed. At present our specification is 32000 ns meaning all 128 columns can be cycled with one $\overline{\text{RAS}}$ low.

From all these diverse applications, it can be seen that this design of MADRAM controller is very versatile, satisfying all the requirements listed.

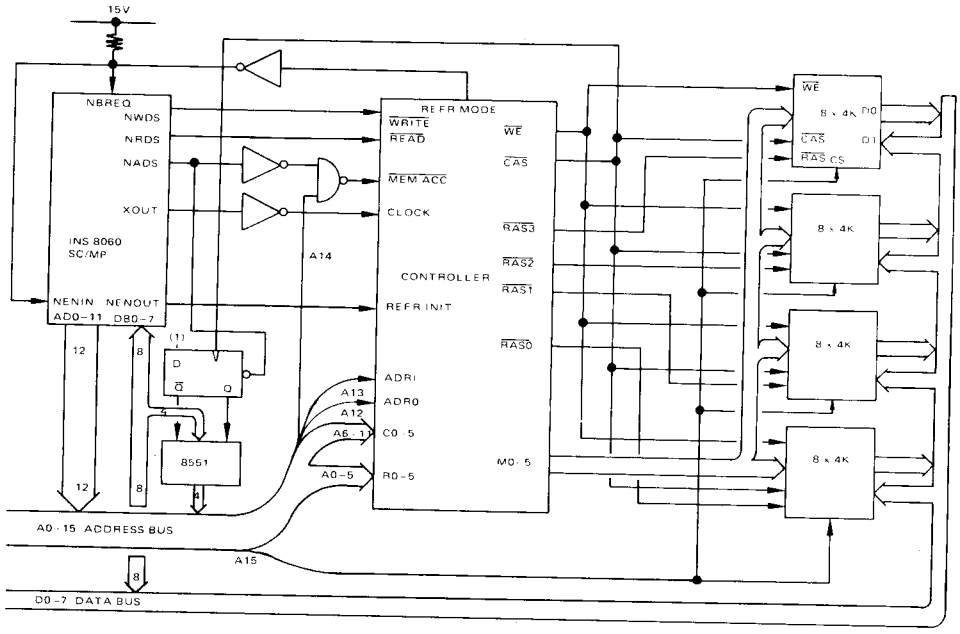


FIG 2 INS 8060 WITH CONTROLLER DRIVING 4 BANKS OF 4Ks

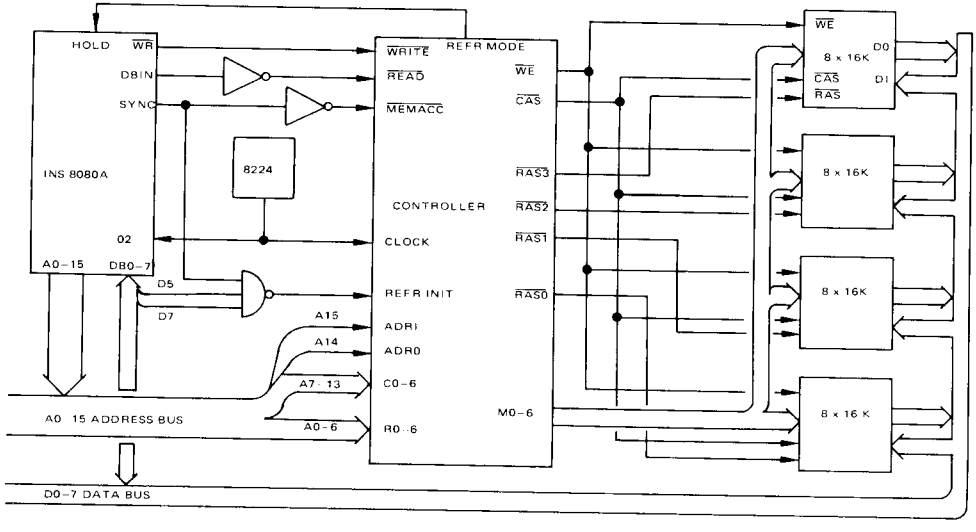


FIG 3 INS 8080A WITH CONTROLLER DRIVING 4 BANKS OF 16Ks

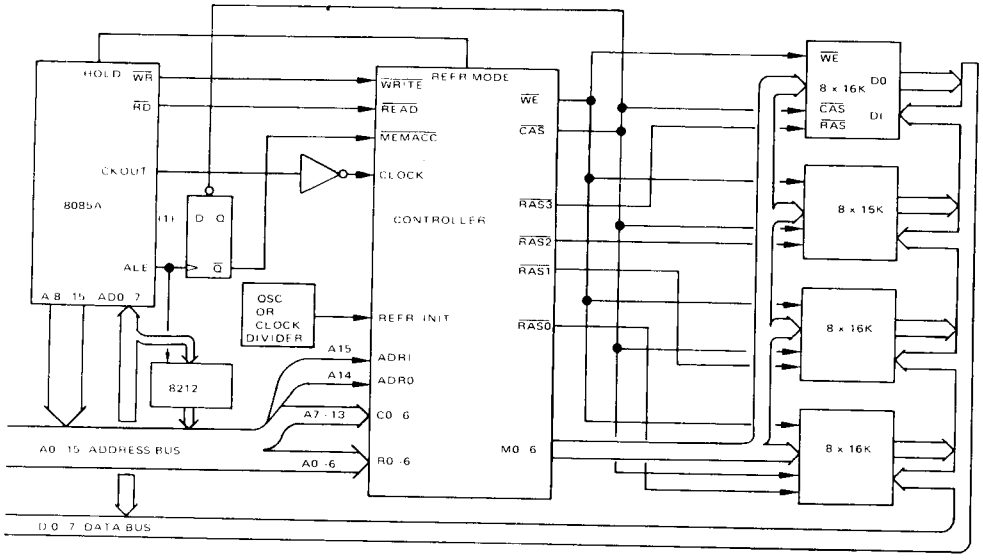


FIG 4 8085A WITH CONTROLLER DRIVING 4 BANKS OF 16Ks

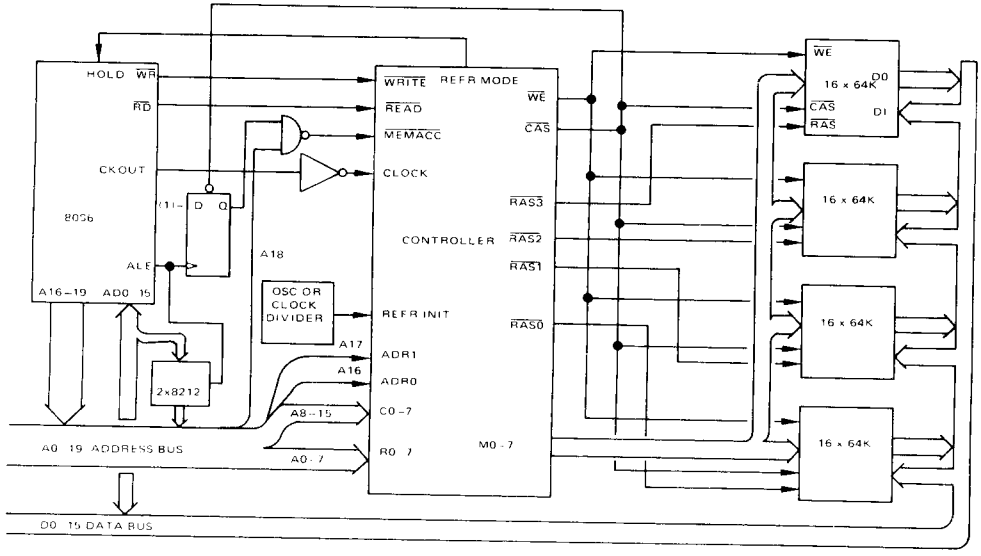


FIG 5 8086 WITH CONTROLLER DRIVING 4 BANKS OF 64Ks