intel®

AB-35

APPLICATION BRIEF

DRAM Refresh/Control with the 80186/80188

STEVE FARRER
APPLICATIONS ENGINEER

August 1990

Order Number: 270524-002

Information in this document is provided in connection with Intel products. Intel assumes no liability whatsoever, including infringement of any patent or copyright, for sale and use of Intel products except as provided in Intel's Terms and Conditions of Sale for such products.

Intel retains the right to make changes to these specifications at any time, without notice. Microcomputer Products may have minor variations to this specification known as errata.

*Other brands and names are the property of their respective owners.

†Since publication of documents referenced in this document, registration of the Pentium, OverDrive and iCOMP trademarks has been issued to Intel Corporation.

Contact your local Intel sales office or your distributor to obtain the latest specifications before placing your product order.

Copies of documents which have an ordering number and are referenced in this document, or other Intel literature, may be obtained from:

Intel Corporation
P.O. Box 7641
Mt. Prospect, IL 60056-7641

or call 1-800-879-4683

COPYRIGHT © INTEL CORPORATION, 1996

DRAM REFRESH/ CONTROL WITH THE 80186/80188

CONTENTS

In many low-cost 80186/80188 designs, dynamic memory offers an excellent cost/performance advantage. However, DRAM interfacing is often complicated by the need to perform memory refreshing. This application brief describes how to use the Timer and DMA functionality of the 80186/80188 to perform memory refresh.

THEORY OF OPERATION

Dynamic RAM refreshing is accomplished by strobing a ROW address to every ROW of the DRAM within a given period of time. One way to do this is to perform periodic sequential reads to the DRAM using a DMA controller and a Timer. This can be achieved with the 80186/188 by Programming Timer 2 and one of the DMA channels such that the timer generated one DMA cycle approximately every 15 micro-seconds. Please note that this is a single row refresh method and not a burst refresh. Single row refreshing reduces the bus overhead considerably when compared to burst refreshing.

The control logic of the DRAM is such that a RAS (row address strobe) occurs on every memory read, regardless of the address. This is necessary because the DMA channel is cycling through the entire 1 MByte address space and the address of the refresh cycle does not always fall within the range of the DRAM bank.

Although the address may be outside the DRAM range, the lower address bits continue to change and roll over to provide the row address.

READY LOGIC WITH MEMORY

Since the DMA controller is cycling through the entire 1 MByte address space, care must be taken to ensure that a READY signal is available for all addresses. One way to do this is to use only the internal wait state generator for memory areas and to strap the SRDY and ARDY pins HIGH. Whenever a refresh cycle occurs outside of a predefined internal wait state area, the external ready pins, which are active HIGH, will complete the bus cycle.

If it is necessary to use the external ready signals for certain memory regions, then it will be necessary to add logic which will generate a ready signal whenever the address of a refresh cycle falls where there is no memory. This can easily be accomplished by either decoding a couple of high order address lines, or by AND-ing all the chip selects so that READY goes active whenever all the memory chip selects are inactive (i.e. the cycle is not in a valid memory region).

BUS OVERHEAD

The absolute maximum overhead can be calculated at a given speed by taking the number of refresh cycles divided by the total number of bus cycles for a given period of time. At 8 MHz these values can be calculated as follows:

$$\frac{\text{2 bus cycles}}{\text{15.2 } \mu\text{s/500 ns}} \times 100 = \text{6.6\% maximum overhead}$$

In reality, the bus overhead associated with the DMA cycles is much lower due to the instruction prefetch queue. When a DMA cycle is requested by the timer for a refresh cycle, the Bus Interface Unit honors the request on the next bus cycle boundary (with the exception of LOCKed bus cycles and odd aligned accesses). Typically this time is idle time on the bus and the impact on the overall performance is extremely small. The following table shows more realistic data which was acquired by running 6 different benchmarks with and without the DMA channel enabled to provide refresh every 15.2μs.

BENCHMARK RESULTS @ 8 MHz

	Minimum	Maximum	Average
80186	1.3%	5.9%	2.5%
80188	2.4%	6.5%	3.4%

The programs which showed the highest bus overhead tended to be very bus intensive. Also note that at faster frequencies the bus overhead becomes even less.

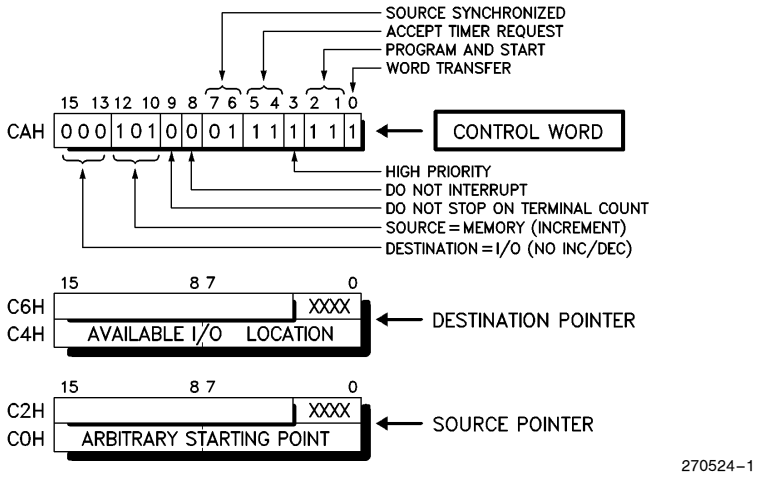
DMA OPERATION

The DMA controller is programmed to be source synchronized with the TC (transfer count) bit cleared. This ensures that the DMA controller never reaches a final count. The source pointer continues to increment through memory on every cycle. When FFFFFH is reached, the address rolls over to 00000H

The programming values for the DNA registers are shown in Figure 1. The source pointer may be initialized to any location since the starting location of the refresh is arbitrary.

The value of the Transfer Count register is also arbitrary since the TC bit is not set. The DMA channel will continue to run cycles upon request from Timer 2 even after the Transfer Count register has reached zero. Once zero is reached, the Transfer Count register will roll over to FFFFH and continue to count down.

The destination pointer may be set to any available memory or I/O location. This pointer must be set so that it neither increments nor decrements. Otherwise, the address of the deposit cycle would cycle through memory or I/O doing writes which could possibly be destructive. Thus the INC and DEC bits of the control register should be cleared.



NOTES:
1. Locations of registers are relative to the base address of the peripheral control block. The offsets shown are for Channel 0.
2. The byte/word bit is a don't care in a 80188 system. In a 80186 system this bit should be set to a 1 to represent word transfers.
3. The transfer count register is located at offset C8H. It is not necessary to program this register.

Figure 1. DMA Registers

TIMER OPERATION

Timer 2 must be programmed to generate a DMA request every time a row must be refreshed. Since we are not using a burst refresh, the refresh time is divided up evenly among the number of rows. For a 2 ms refresh DRAM with 128 rows, the time between rows equals 15.62 microseconds.

When setting the count value of the timer, keep in mind the timer clock is operating at one-fourth the CPU clock frequency. Thus, the equation for setting the timer count is:

$$\frac{(\text{CPU CLOUT FREQ}) \times (\text{Time Between ROWS})}{4} = \text{COUNT\_\_VALUE (decimal)}$$

For an 8 MHz clock, programming the Maximum Count Register to 1EH provides a 15.2 μs refresh. This programming is indicated in Figure 2.

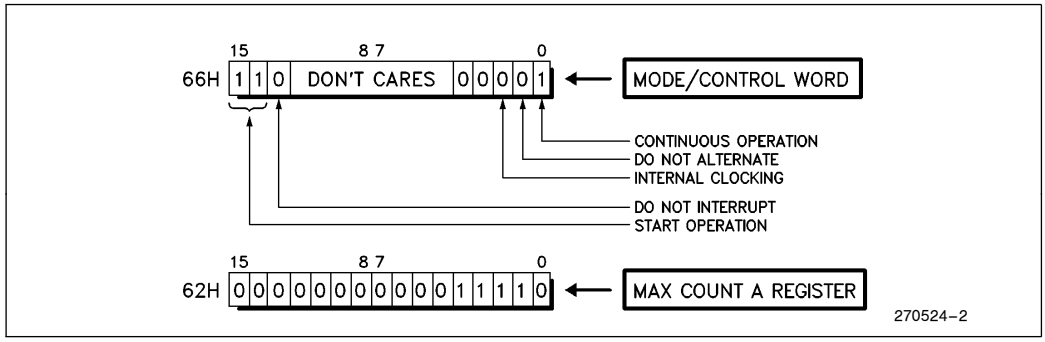


Figure 2. Timer 2 Registers Programmed for a 15.2 μs Refresh at 8 MHz

EXAMPLE 1: DRAM CONTROL WITH A DELAY LINE

This is the most straight forward way of implementing the RAS and CAS logic. A RAS signal is generated by either RD or WR going active while the address is within the corresponding range. Normally the logic for RAS would also go active for a refresh cycle status, but since this information is not available on the 80186/80188, a RAS must be generated for every RD and WR, regardless address.

The MUX signal is used to change from the RAS address to the CAS address after latching with RAS. This is accomplished by using a delay line which generates a MUX signal by a fixed number of nano-seconds after RAS is generated. The important timing here is the necessary hold time for the row address into the DRAM.

The MUX signal is initially HIGH which sends the A side (see Figure 3) Row address through the multiplexer to the DRAM. This address consists of A0 through A7. The B address (A8 through A16) is selected when MUX goes LOW. The system shown in Figure 3 represents that of an 80188 system.

For an 80186 system, the A address would start at A1. The least significant address line A0 along with BHE would be used to decode WE into WEH and WEL which will be shown in the second example. Also, the 186 DMA must be set to do word transfers so that the address is incremented by 2 after each refresh cycle. This is necessary to ensure Al increments by 1 every refresh cycle.

CAS is generated in the same manner by delaying the MUX signal a fixed number of nano-seconds. Typically CAS goes inactive at the same time as RAS to ensure a valid CAS precharge time before the next DRAM access. The 80186/188 chip selects are used to ensure that CAS only goes active when the address falls within the DRAM bank range, and to ensure that CAS does not go active during I/O cycles.

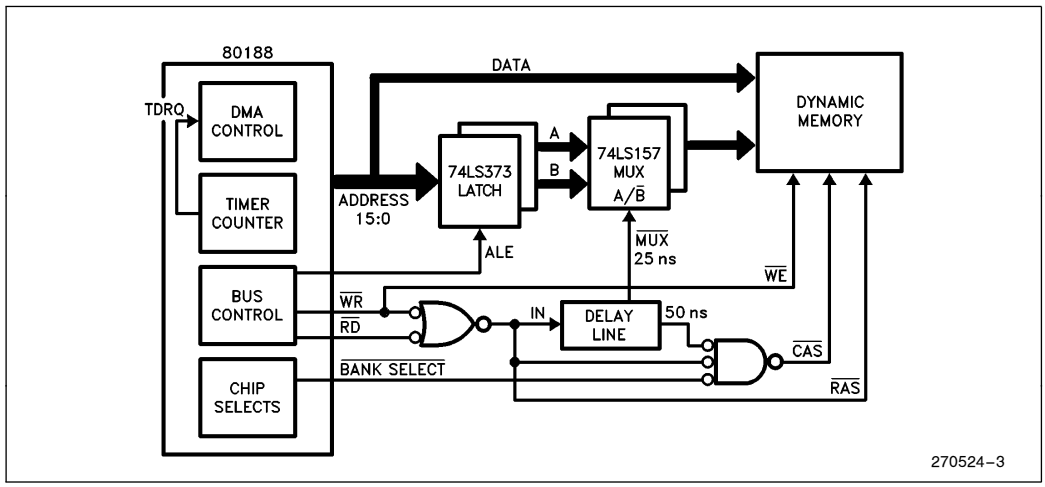


Figure 3. Using A Delay Line for DRAM Control

EXAMPLE 2: DRAM CONTROL WITH A PAL*

This design uses a PAL to generate all the control logic for the DRAM array. Internal feedback is used on the signals to control the timing and states of the $\overline{RAS}$, $\overline{MUX}$ and $\overline{CAS}$ signals.

This design uses 256k X 4 DRAMs. With minor changes to the PAL equations this design could just as easily make use of 64k X 1, 64k X 4, or 256k X 1 DRAMs.

The $\overline{RAS}$ signal is generated off ALE going LOW, bus cycle status active, and $\overline{PRE\_RAS}$ being active. The $\overline{PRE\_RAS}$ signal is necessary to ensure that a $\overline{RAS}$ is not accidentally generated when S2-SO are becoming valid and ALE has not yet gone HIGH in T4 phase 2. $\overline{PRE\_RAS}$ does not go active until ALE has gone HIGH.

$\overline{RAS}$ is initiated for every memory read and write regardless of the bus cycle address. This ensures a row refresh when the refresh address falls outside of the DRAM bank and also a refresh to both banks simultaneously so that the frequency of the refresh can be set for the number of rows in one bank of DRAM.

The $\overline{UCS}$ (Upper Chip Select) from the 80186/188 is used to disable DRAM signals when the processor is attempting to access upper memory control ROM. Thus the portion of memory used by the $\overline{UCS}$ (maximum 256k) is unavailable in the upper DRAM. However, the $\overline{RAS}$ signal must still be allowed during $\overline{UCS}$ access to ensure refreshing when the DMA refresh cycle occurs in the $\overline{UCS}$ region.

$\overline{MUX}$ is generated off T2 phase 1 and $\overline{RAS}$ active. $\overline{MUX}$ will remain low until the current $\overline{RAS}$ signal goes inactive during T3 phase 2.

$\overline{CAS0}$ and $\overline{CAS1}$ are generated off $\overline{MUX}$ being active and T2 phase 2 of the bus cycle. $\overline{CAS}$ goes inactive at the start of T4 phase 2.

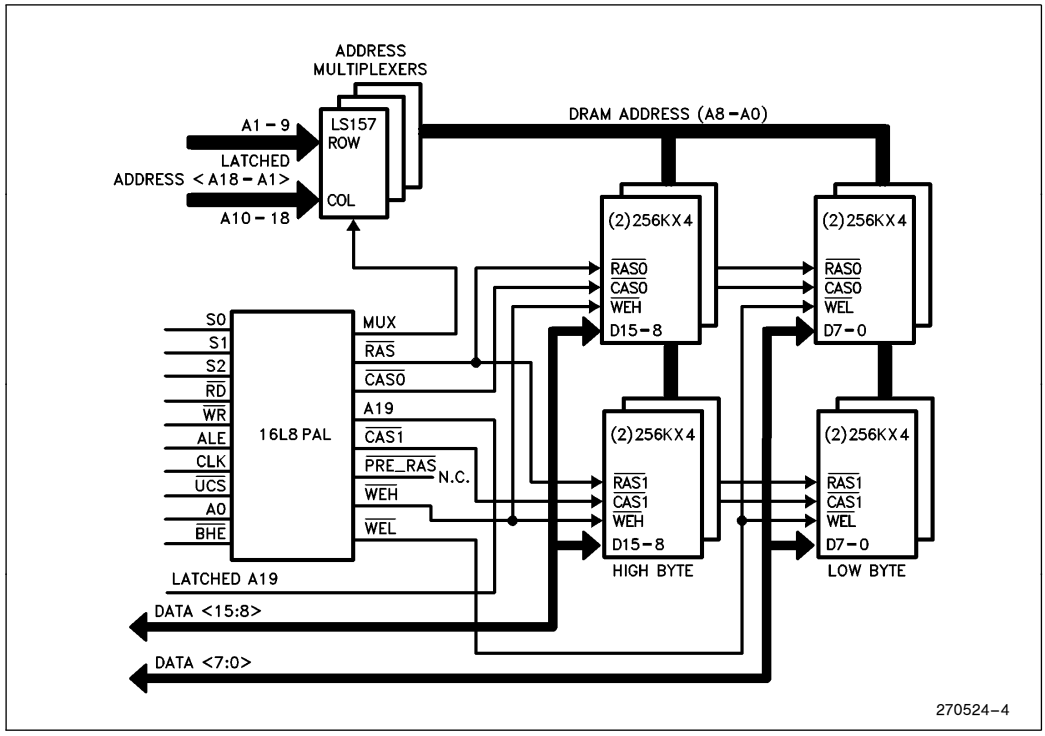


Figure 4. Using a PAL for DRAM Control

*PAL is a registered trademark of Monolithic Memories.

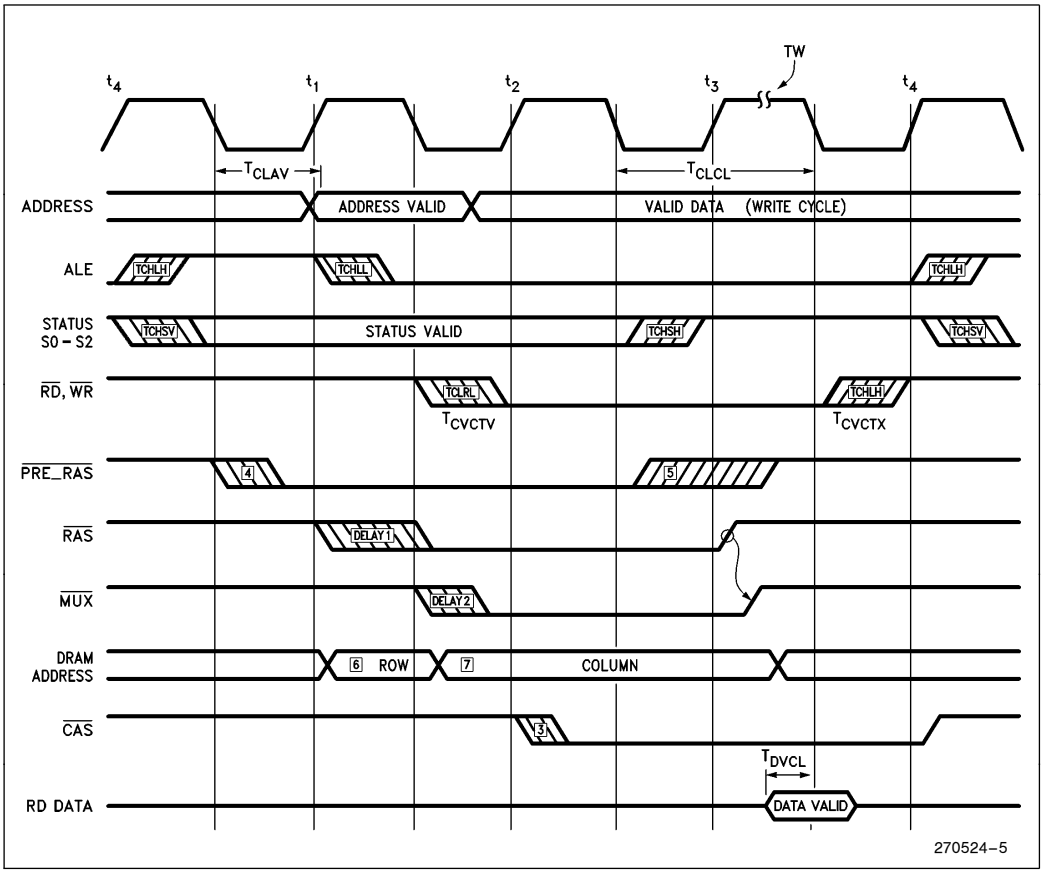


Figure 5. Timing Diagram for PAL DRAM Controller

PAL EQUATIONS FOR 80186 SYSTEM

```
PRE__RAS  =  ALE * S2 * S1 * S0 +          ;INSTRUCTION FETCH
             ALE * S2 * S1 * S0 +          ;READ DATA/REFRESH
             ALE * S2 * S1 * S0 +          ;WRITE DATA
             PRE__RAS * S2 * S1 * S0 +     ;KEEP PRE__RAS VALID
             PRE__RAS * S2 * S1 * S0 +     ; WHILE STATUS
             PRE__RAS * S2 * S1 * S0       ;IS VALID
RAS       =  PRE__RAS * ALE * S2 * S1 * S0 +   ;INSTRUCTION FETCH
             PRE__RAS * ALE * S2 * S1 * S0 +   ;READ DATA/REFRESH
             PRE__RAS * ALE * S2 * S1 * S0 +   ;WRITE DATA
             RAS * CLK                         ;KEEP ACTIVE DURING T3A
MUX       =  RAS * CLK +
             RAS * MUX
CAS0      =  A19 * MUX * CLK * RAS +
             CAS0 * RD +
             CAS0 * WR +
             CAS0 * CLK
CAS1      =  A19 *UCS * MUX * CLK * RAS +
             CAS1 * RD +
             CAS1 * WR +
             CAS1 * CLK
WEL       =  WR * AO
WEH       =  WR * BHE
```

TIMING EQUATIONS

	8 MHz	10 MHz
TCLAV	55	50
TCHLH	35	30
TCHLL	35	30
TCHSV	55	45
TCLSH	65	50
TCLRL/TCVCTV	70	56
TCLRH	55	44
TDVCL	20	15

The following equations are with reference to given clock edge. The edge in reference is indicated by the first element in the equation: T3 ↑ = rising edge of T3 clock ↓T1 = falling edge of T1 clock.

```
DELAY 1 = T1 ↑ + TCHLL + (PAL DELAY)
DELAY 2 = ↓ T2 + (PAL DELAY)
DELAY 3 = T2 ↑ + (PAL DELAY)
DELAY 4 = ↓ T1 + (PAL DELAY)
DELAY 5 = ↓ T3 + TCLSH + (PAL DELAY)
DELAY 6 = ↓ T1 + TCLAV + (MUX DELAY)
DELAY 7 = ↓ T2 + DELAY 2 + (MUX DELAY)
```

ACCESS TIME FROM $\overline{RAS}$ = 2.5 (TCLCL) − DELAY 1 − TDVCL

ACCESS TIME FROM $\overline{CAS}$ = 1.5 (TCLCL) − DELAY 3 − TDVCL