AN1063

Application Note

DRAM Controller for the MC68340

This document describes a design concept for a 16.78 MHz dynamic random-access memory (DRAM) controller with hardware refresh for the MC68340. The design has been simulated successfully but has not been built in hardware. The DRAM controller and memory system are implemented using programmable logic arrays (PLAs), combinatorial logic, and 1M x 4 bit 100 ns DRAMs.

The purpose of this design is to provide a low cost high-performance main memory subsystem for the MC68340. Since a cost versus performance tradeoff always occurs, the cost was minimized given a performance level of only one wait state per memory access and a hardware refresh. The MC68340 contains several on-chip modules that traditionally require additional chips. When possible the design uses these MC68340 features to reduce complexity and cost. These features include a timer module, the programmable chip-select signals, and the clock generation circuitry.

OVERVIEW

Figure 1 shows a block diagram of the memory system, including the MC68340 processor. The 16 bit wide memory provides 8 Mbytes of DRAM. Although this design uses 8 Mbytes of DRAM, it can easily be modified for smaller amounts of memory. The DRAM select (DRAMSEL) signal is generated by one of the MC68340 chip-select signals. The MC68340 chip-select circuitry decodes DRAMSEL directly from the address (A31 – A0) and function code (FC3 – FC0) signals. The memory system starts either a read or a write access after receiving an asserted DRAMSEL.

Refresh requests are generated by timer 1 of the MC68340. Each time a refresh is required, this timer asserts a signal (TOUT1) for two processor clock cycles. The DRAM controller services the request by performing a refresh cycle.

The MC68340 supports both 8- and 16-bit data accesses. The DRAM controller supports 8-bit accesses by using the SIZ0 signal to determine the size of the data access.

HARDWARE DESIGN

The overall system consists of the MC68340 and the three-component DRAM system. The DRAM system consists of the address multiplexers, the DRAM array, and the control logic.

The DRAM array is organized into four columns; however, only one column is active during a read or write access. For this reason, there are four separate row address strobe (RASn) and column address strobe (CASn) signals. All four columns of the array are active during refresh cycles.

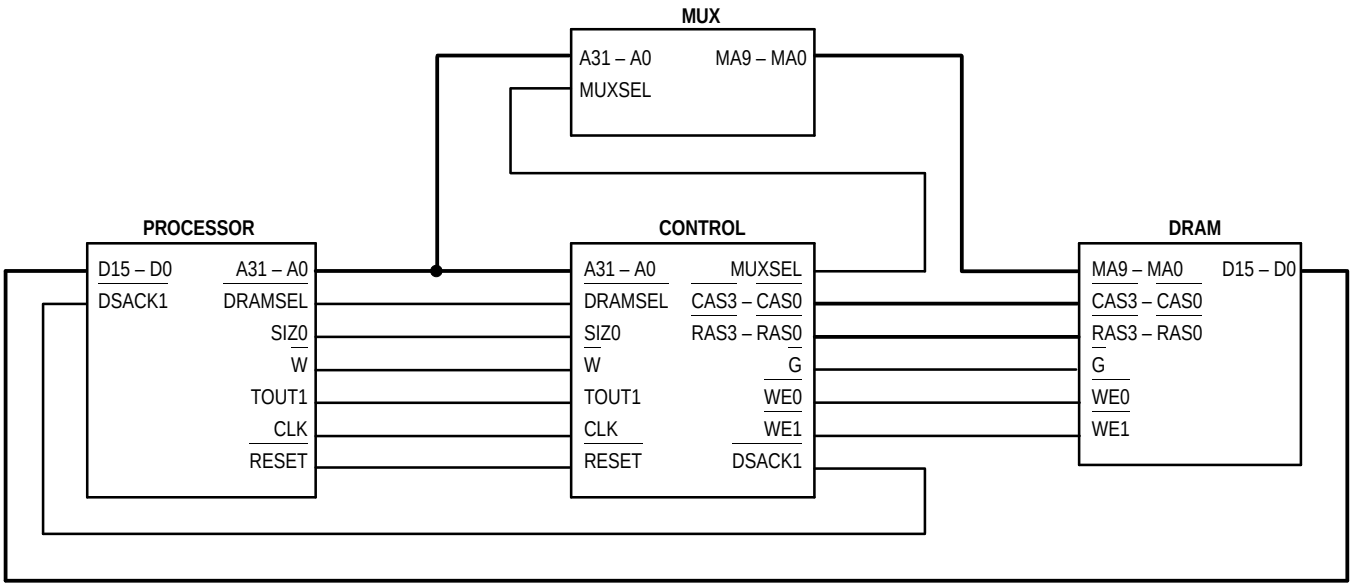


Figure 1. MC68340 with Memory System

Figure 2 shows the basic read cycle timing for the DRAM system. The address from the MC68340 is multiplexed to provide row and column addresses. The DRAM control logic starts an access by asserting RASn after receiving DRAMSEL. CASn is asserted after the DRAM controller moves into state 1 (T1). The state machine of the controller changes states on the falling edge of the system clock. DSACK1 is asserted during MC68340 bus state W1 to terminate the cycle with one wait state.

Figure 3 shows the write cycle timing for the DRAM system. The timing is the same as a read cycle except one or both of the write enable (WEn) signals are asserted.

The refresh cycle timing is shown in Figure 4. TOUT1 is latched to generate the refresh request (RFQ) signal, which causes a CAS before RAS refresh cycle to be performed after the present read or write cycle finishes. CASn is asserted during state T0, and RASn is asserted after the DRAM controller moves into state T4. The DRAM chip internally supplies the address of the row to be refreshed.

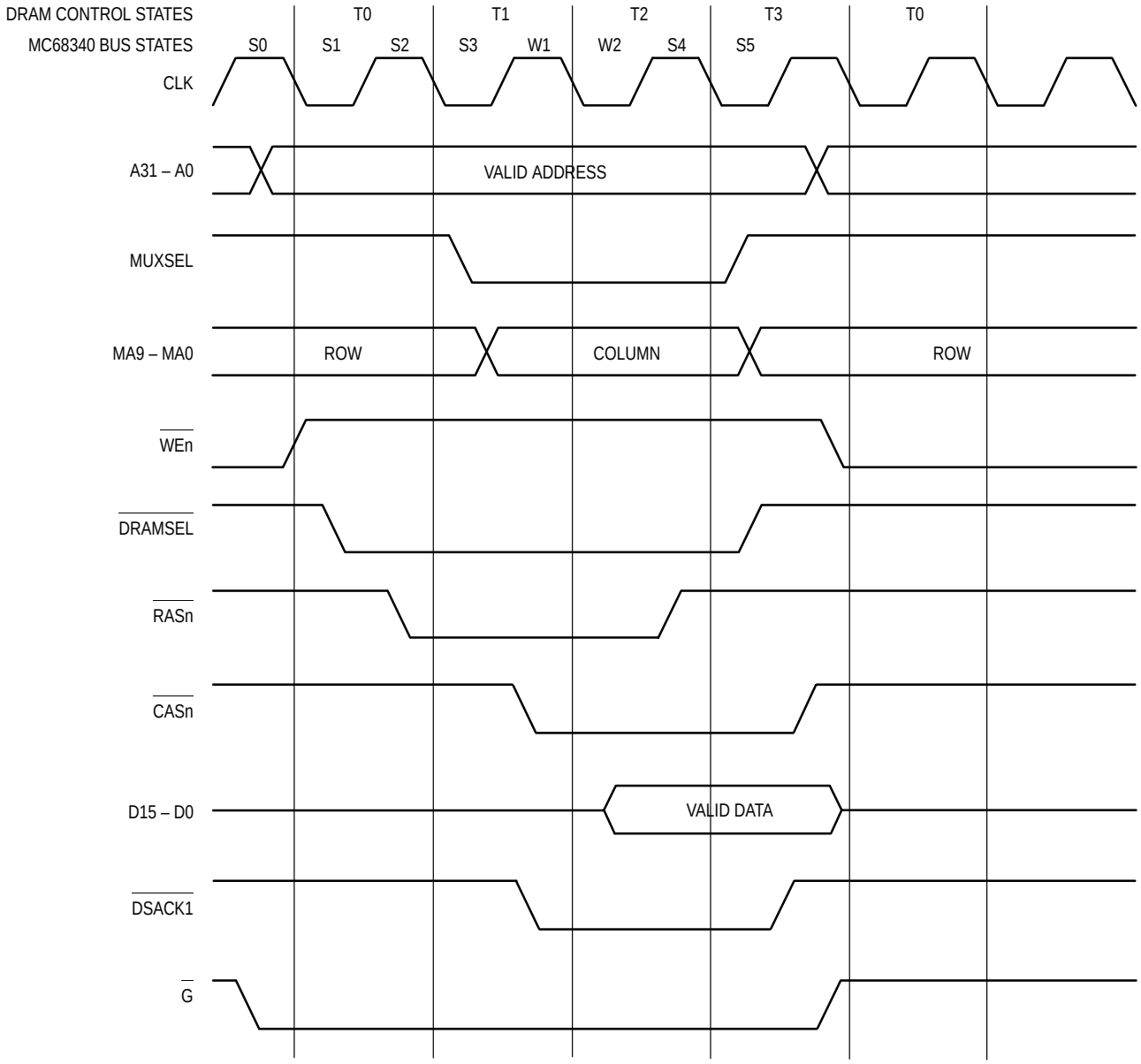


Figure 2. 16.78 MHz Read Cycle Timing Diagram

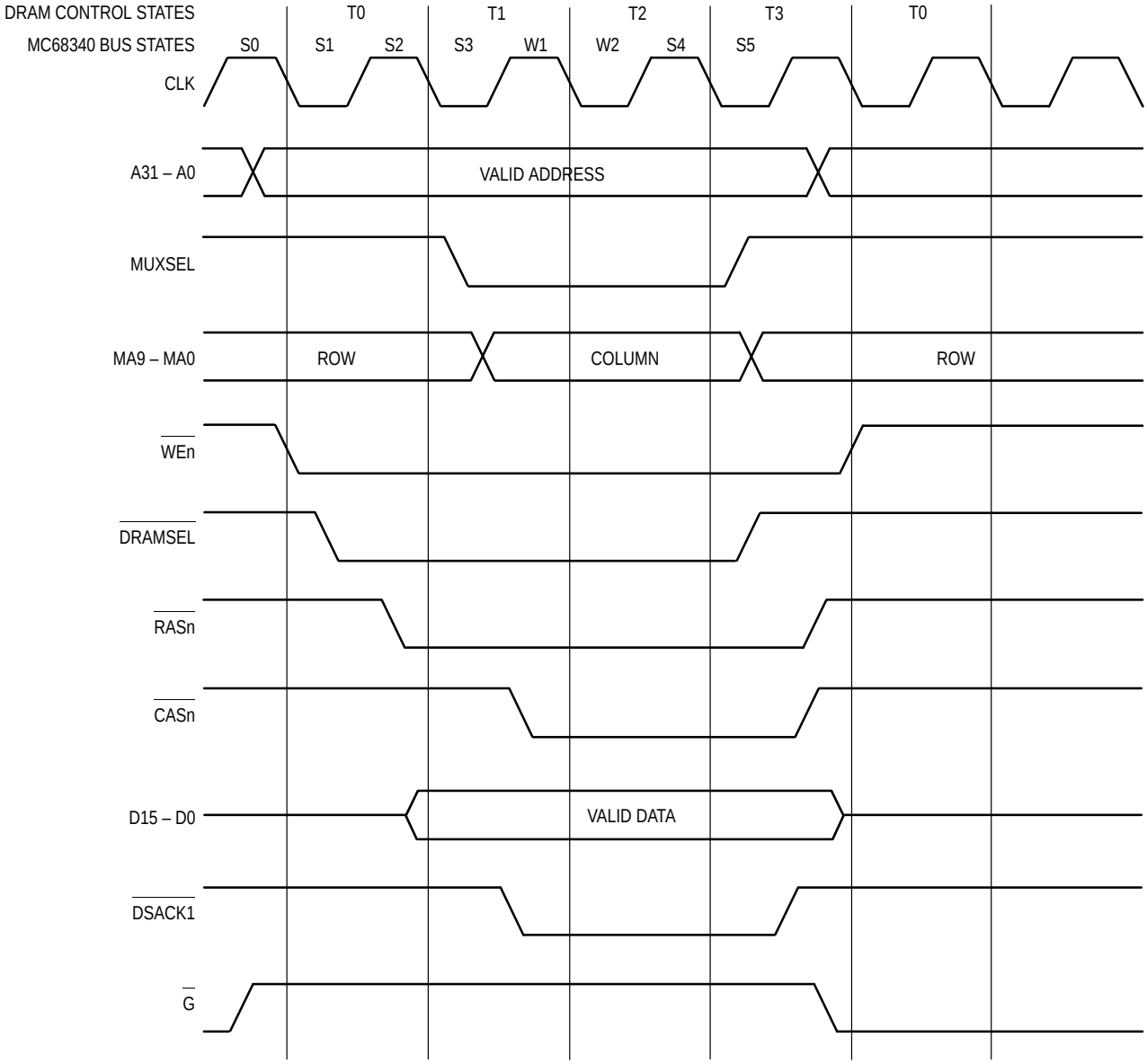


Figure 3. 16.78 MHz Write Cycle Timing Diagram

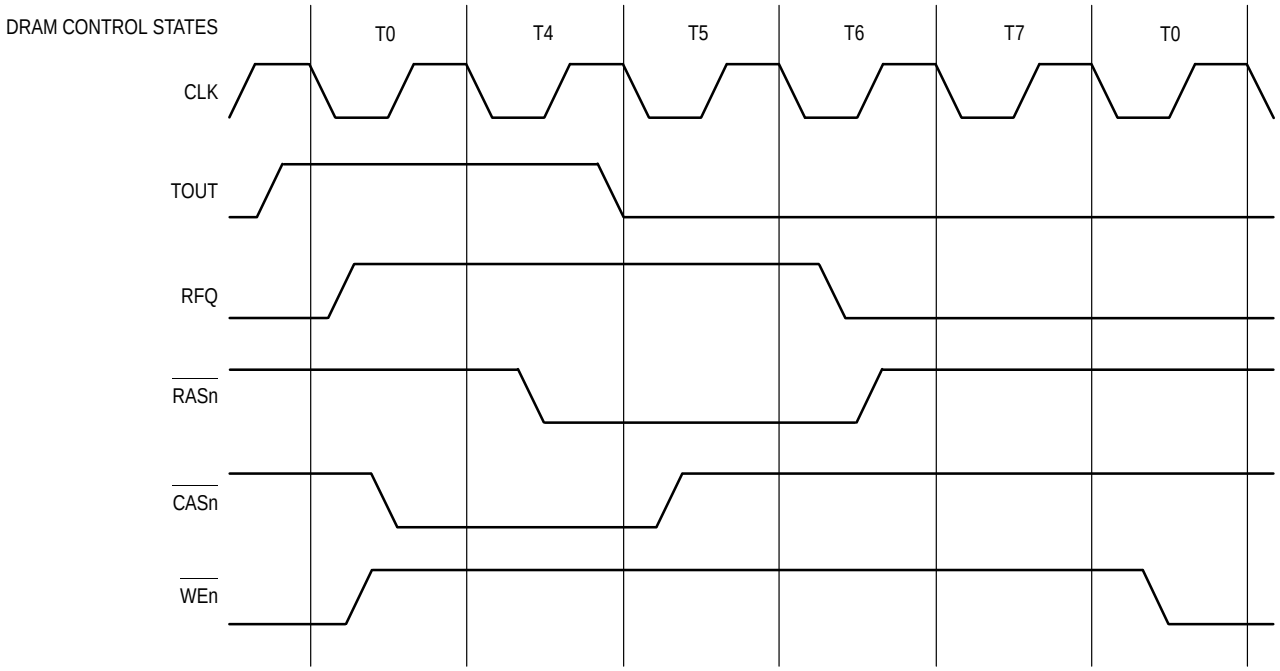
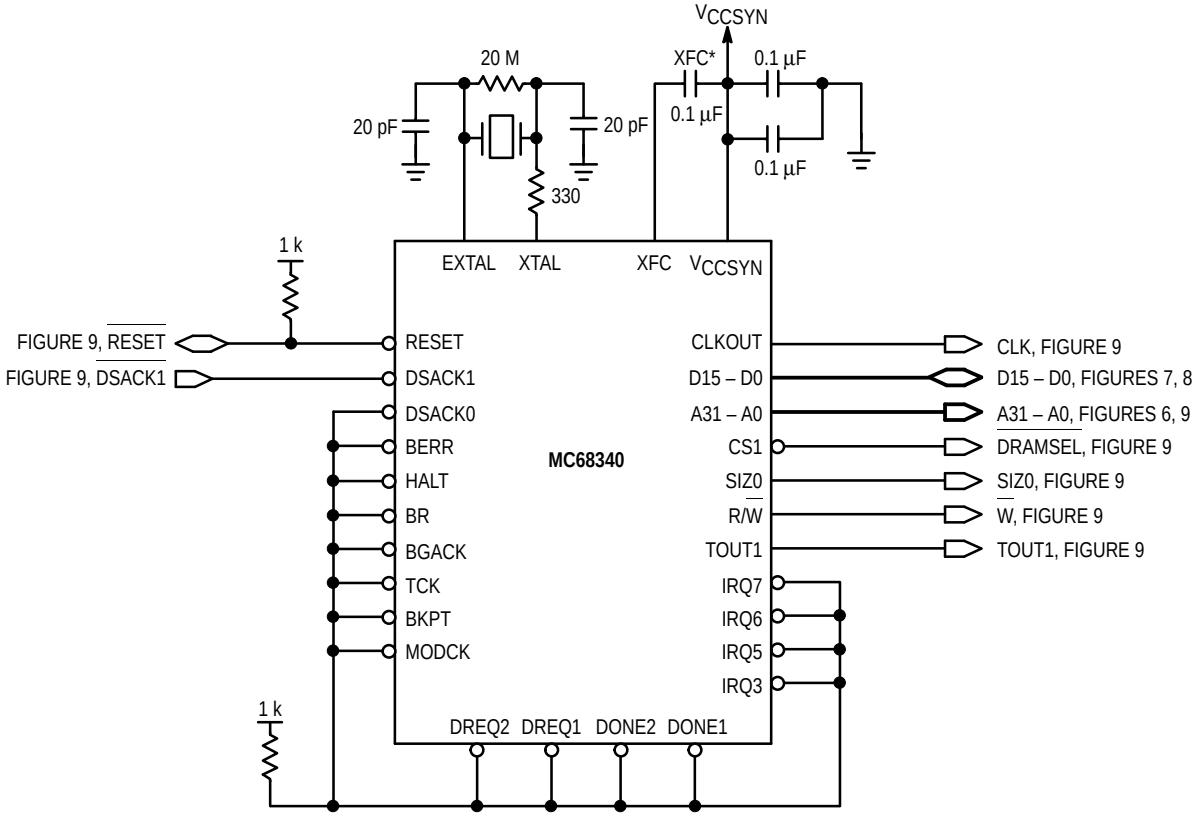


Figure 4. 16.78 MHz Refresh Cycle Timing Diagram

PROCESSOR

Figure 5 shows the connections required for the MC68340. MC68340 inputs that are not used in this design but cannot be left floating, are tied to a pullup resistor. Since the clock generation capability of the MC68340 is used to generate a 16.78 MHz clock, a 32.768 kHz reference crystal and some circuitry must be supplied. This circuitry, which has been copied from the MC68340UM/D, *MC68340 User's Manual*, is shown between EXTAL and XTAL and between XFC and V_{CCSYN}.



*Must be low-leakage capacitor.

Figure 5. MC68340 Connections

ADDRESS MULTIPLEXING

The address multiplexing scheme is shown in Figure 6. The multiplexing is performed by three 74F258 two-input multiplexers. Address lines A20 – A1 are used to form the multiplexed address bus (MA9 – MA0).

Address line A0 is not used in the multiplexing because the size of the memory port is 16 bits. The multiplexer select (MUXSEL) signal is connected to the select input of each multiplexer. When asserted, MUXSEL selects the row address (A20 – A11); when negated, it selects the column address (A10 – A1).

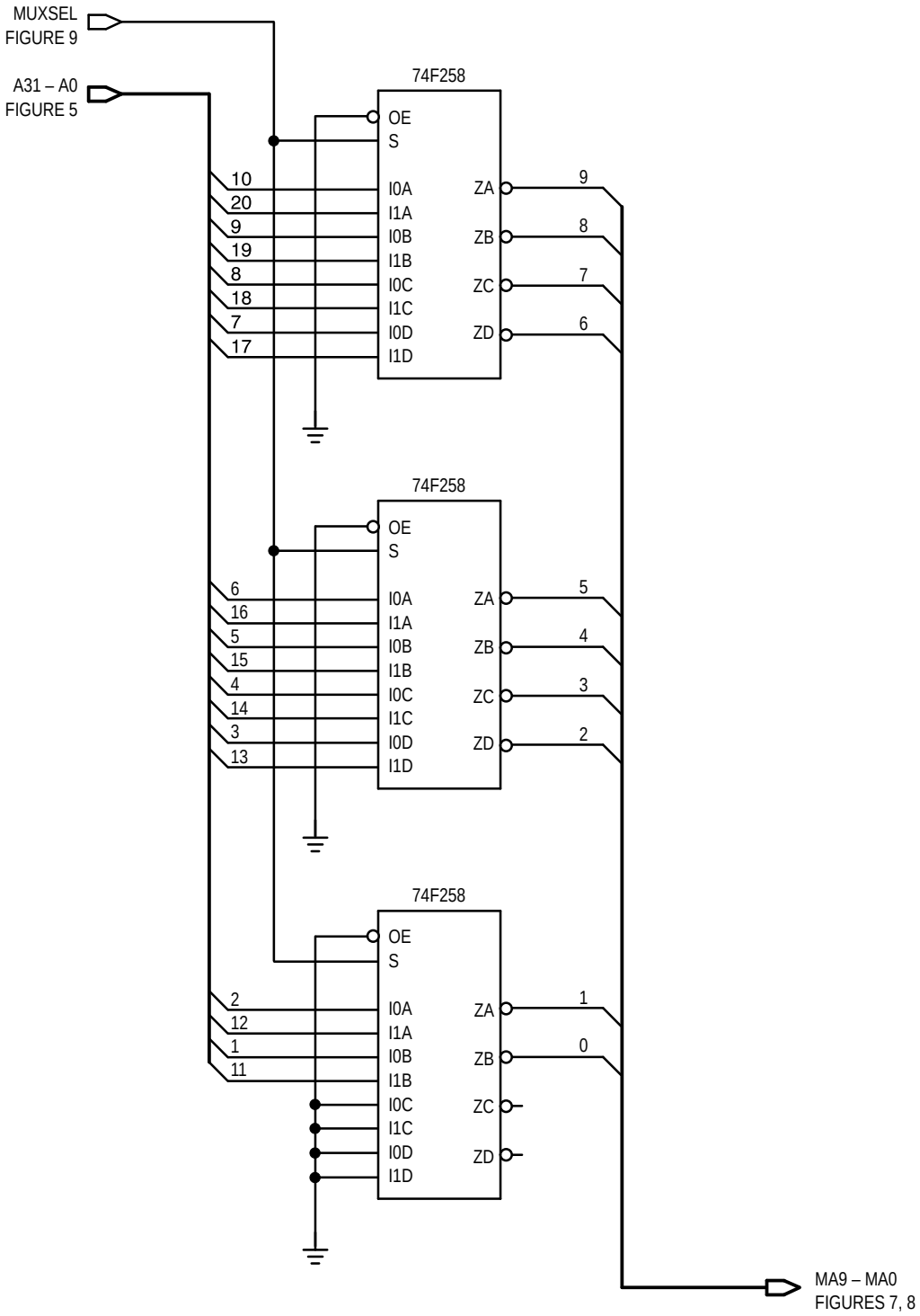


Figure 6. Address Multiplexing

DRAM ARRAY

Figures 7 and 8 show the DRAM array. The array is made up of 16, 100 ns, 1M x 4 bit DRAMs (MCM514400-10). These DRAMs require a 1024 cycle refresh every 16 ms. Since four DRAM chips are capable of supplying 16 bits of data, the array is organized into four columns of four chips each, with only one column active at a time, except during refresh cycles in which all four columns are active. The active column is determined by address bits A22 and A21, which select which of the four sets of RAS and CAS signals are pulsed.

Separate write enable signals are used for the most significant byte (WE0) and the least significant byte (WE1). WE0 is asserted during word (16-bit) or even-byte accesses. WE1 is asserted during word or odd-byte accesses.

The use of 1M x 4 bit DRAMs has two major advantages over the use of 4M x 1 bit DRAMs. The first advantage is lower power consumption, which results from having only four of the 16 chips active during read or write cycles. The second advantage involves the ease with which the design can be altered to provide 6, 4, or 2 Mbytes of memory. For example, to reduce the memory size from the 8 Mbytes given in this design to 6 Mbytes, only two steps are required. First, remove the fourth column of the DRAM array along with the two NAND gates that generate RAS3 and CAS3. Second, reprogram the MC68340 so that DRAMSEL is asserted only during accesses to the low 6 Mbytes of memory instead of the low 8 Mbytes. Similarly, two columns are removed to provide 4 Mbytes, and three columns are removed to provide 2 Mbytes.

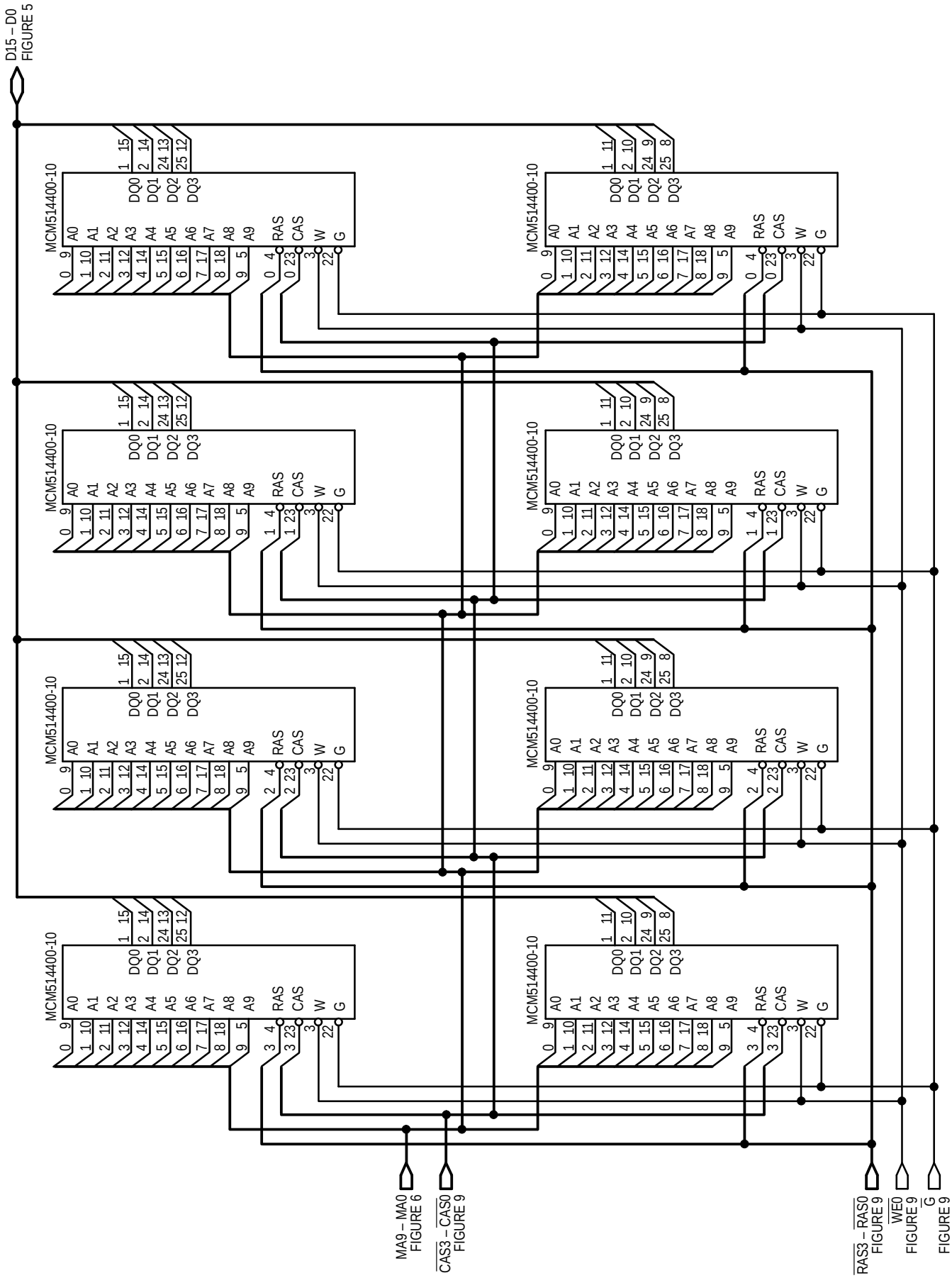


Figure 7. DRAM Arrays — High Order Bits

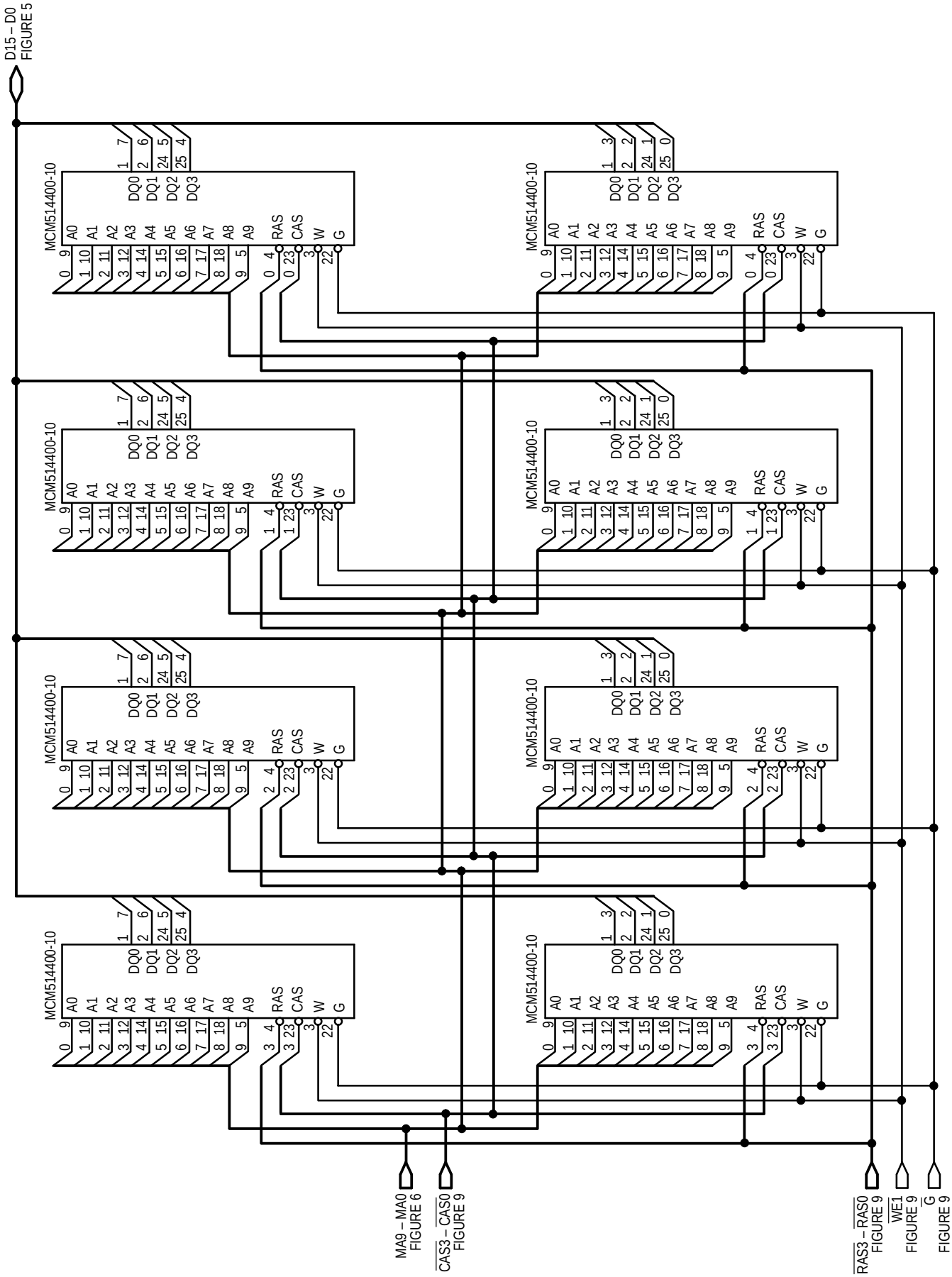


Figure 8. DRAM Arrays — Low Order Bits

DRAM CONTROL LOGIC

The DRAM control logic is shown in Figure 9. The PLAs used include one 10 ns PAL16R4 and one 25 ns PAL16L8. The logic equations for these devices provide more detailed information about the individual signals (see **LIST OF LOGIC EQUATIONS** at conclusion of text).

The PAL16R4 contains the gray-coded control state machine, which has eight possible states. This PAL is clocked by CLKB (the inverted system clock), which causes it to change state on falling edges of the system clock. State T0 is the idle state, states T1 – T3 are the states for a read or write cycle, and states T4 – T7 are the states for a refresh cycle.

An asserted DRAMSEL causes a read or write cycle to occur; an asserted RFQ causes a refresh cycle to occur. If TOUT1 is asserted during a read or write cycle, it is latched to generate the RFQ signal, and the refresh is performed after the current cycle is complete. If DRAMSEL is asserted during a refresh cycle, the read or write cycle is performed after the refresh cycle is complete. The MC68340 automatically inserts extra wait states in this situation as a result of the DRAM controller not asserting DSACK1.

The PAL16L8 receives information about the present state from the PAL16R4. This information, along with information received directly from the MC68340 is used to generate four select signals (SEL3 – SEL0) and two write enable signals (WE1 and WE0). The four select signals choose which of the four DRAM array columns are active during an access. During a read or write cycle, only one of these select signals is asserted, depending on address bits A22 and A21. During a refresh cycle, all four select signals are asserted because all the DRAMs need to be refreshed. WE0 is asserted only during a write cycle in which the most significant byte must be written, and WE1 is asserted only during a write cycle in which the least significant byte must be written. These write enables are always negated during read cycles or refresh cycles.

The RAS and CAS signals generated by the PAL16R4 are NANDed with the SEL3 – SEL0 signals from the PAL16L8 to generate the RAS3 – RAS0 and CAS3 – CAS0 signals. Thus, if SELn is asserted during an access, then RASn and CASn will pulse during the access, causing column n of the DRAM array to be active. The output enable signal (G) is generated by inverting R/W, causing DRAM outputs to be enabled only during read cycles.

REFRESH COUNTER VALUE

The following paragraphs describe the calculations that give the value to load into the refresh counter. The MC68340 timers operate at 8.39 MHz, which is one half the system frequency of 16.78 MHz. Since the 1M x 4 bit DRAM refresh specification is 1024 cycles per 16 ms, the number of timer clock periods between refresh cycles is given by the following equation:

$$RC = 8.39\ \text{MHz} \div (1024\ \text{cycles} \div 16\ \text{ms}) = 131$$

where RC equals the refresh count, which is rounded down.

Any delay from the time that the refresh is requested to the time it actually occurs must be subtracted from the refresh count. If a refresh request occurs during the first clock of a read or write cycle, it must wait three more system clocks for the cycle to finish. This translates to 1.5 timer clocks, which will be rounded up to two.

The interval between refresh requests is given by the refresh count minus the maximum refresh delay. The value to be loaded into the counter is this result decremented by one, since the counter goes down to zero. This value is 128, which translates to 80 in hexadecimal.

MC68340 REGISTER SPECIFICATIONS

The following paragraphs describe the values to be loaded into the MC68340 system integration module (SIM) for the MC68340 to function as required. The module configuration register (MCR) should be programmed so that the four external chip selects are chosen. The clock synthesizer control register (SYNCR) should be programmed so that the operating frequency is 16.78 MHz when referenced to a 32.768 kHz crystal. The base address registers and the address mask registers should be programmed so that CS1 is asserted during accesses to the DRAM array and no internal DSACKx generation occurs.

Table1 lists the hexadecimal values to be loaded into each register and gives a brief description of each value.

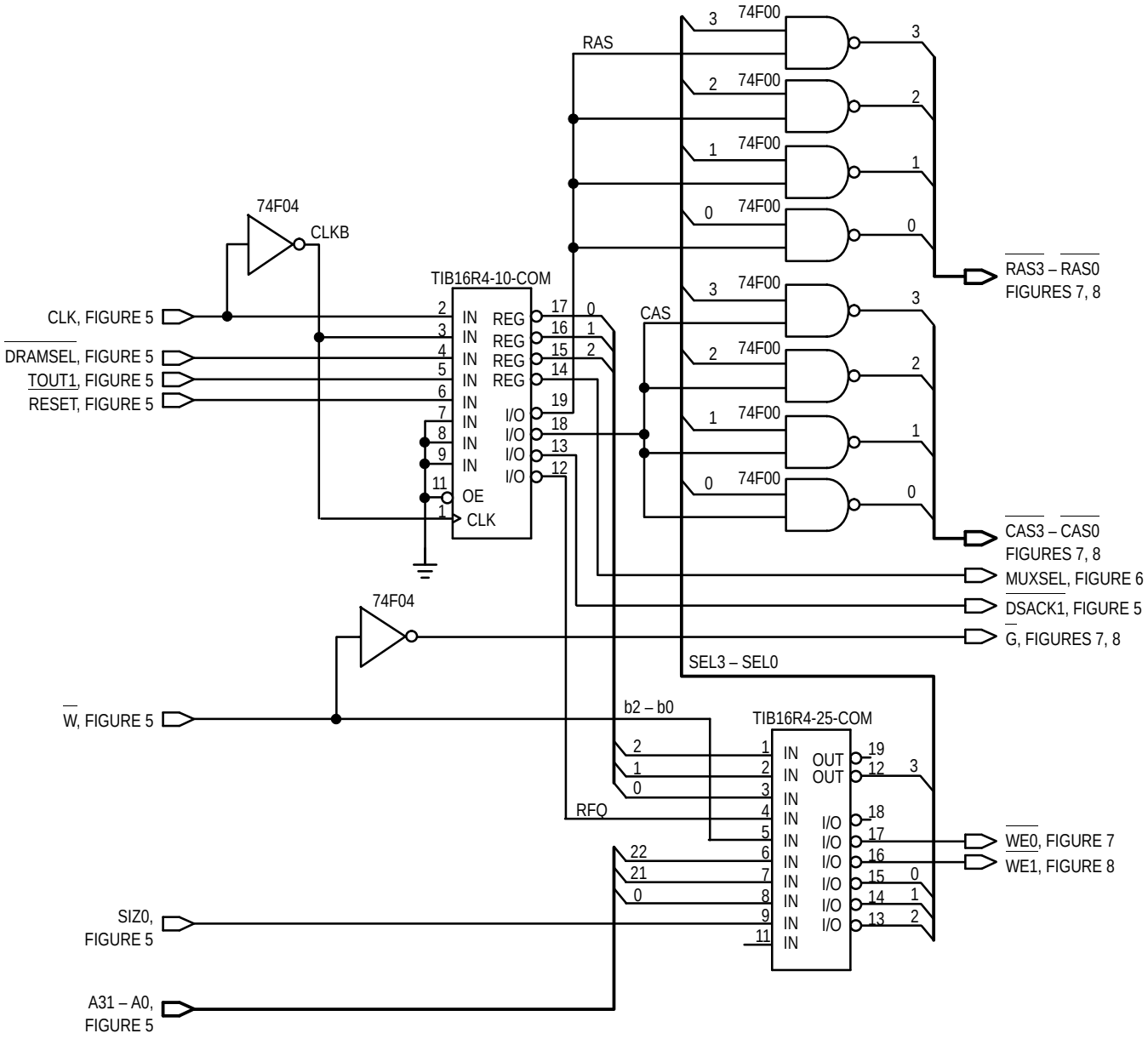


Figure 9. DRAM Control Logic

Table 1. Register Hexadecimal Values

Address	Register Name	Hexadecimal Value	Description
$000	MCR	$0082	Choose chip selects, SIM registers restricted to supervisor accesses, interrupt arbitration level 2
$004	SYNCR	$CF00	Choose 16.78 MHz clock, given a 32.768 kHz crystal
$04C $04E	Base Address 1 CS1 Base Address 2 CS1	$0000 $0001	Set base address of 8 Mbytes to zero, set valid bit
$048 $04A	Address Mask 1 CS1 Address Mask 2 CS1	$007F $FF33	Bits A22 – A0 are don't care to allow selection of addresses $0000 0000 to $007F FFFF, no chip select on supervisor accesses, no internal DSACKx generation

MC68340 TIMER1 REGISTER SPECIFICATIONS

For this design, the timer operates so that TOUT1 is asserted for one counter clock cycle when a refresh is necessary. Since the counter clock operates at one-half the system clock frequency, TOUT1 is asserted for two system clocks. This operation is achieved by operating the timer in input capture/output compare mode. When the timer counts down to zero, TOUT1 is asserted. On the next clock cycle, the timer reloads itself from the preload register, and TOUT1 is negated.

To program timer 1, the registers must be written as follows. The timer is ensured to be in normal operating mode by programming the module configuration register (MCR). The control register (CR) and status register (SR) are then written to perform a software reset. The preload 1 register (PREL1) and the compare register (COM) are loaded with the count value, and CR is written to start the timer.

Table 2 lists the hexadecimal values to be loaded into each register, the order in which they should be written, and a brief description of each.

Table 2. Register Value Order

Address	Register Name	Hexadecimal Value	Description
$600	TIMER1 MCR	$0081	Ensure timer is not in stop mode, interrupt arbitration level 1
$606	TIMER1 CR	$0000	Software reset
$608	TIMER1 SR	$0000	Clear TC, TG, TO bits
$60C	TIMER1 PREL1	$0080	Value that will be preloaded into counter
$610	TIMER1 COM	$0080	Compare value
$606	TIMER1 CR	$8203	Start timer, no interrupts enabled, do not use prescalar, clock enabled, clock is one-half system frequency, input capture/ output compare mode, one mode

POWER-UP INITIALIZATION REQUIREMENTS

Certain initialization steps required by the DRAMs should be performed in software on power-up. These steps should follow the assertion of RESET as required by the MC68340. The initialization steps required by the DRAMs are a 200 µs pause, followed by a minimum of eight active cycles of the row address strobe. At a 16.78 MHz frequency, 200 µs translates to 3356 clocks. Thus, the first initialization step could be performed by polling an MC68340 timer until these 3356 clock cycles have passed. Alternately, a loop could be written that took 3356 clock cyles to execute, and the MC68340 could execute this loop. The second step is accomplished by performing eight 16-bit write cycles to each of the four columns of the DRAM array. For instance, eight writes could be performed to each of the following hexadecimal addresses: $0000 0000, $0020 0000, $0040 0000, and $0060 0000.

CONCLUSION

The memory system presented in this application note provides high performance at a reasonably low cost. This low cost is partially due to the integration of the MC68340, which contains timers, programmable chip-select signals, and clock generation circuitry. Table 3 lists the parts required for the DRAM memory. The design provides 8 Mbytes of memory but is easily modified to provide smaller amounts of memory.

Table 3. List of Parts

Part Number	Description	Quantity
74F258	Two-Input Multiplexer	3
PAL16R4	10 ns, Registered PAL	1
PAL16L8	25 ns, PAL	1
74F04	Hex Inverter	1
74F00	Two-Input NAND	2
MCM514400-10	100 ns 4 M x 1 DRAM	16

LIST OF LOGIC EQUATIONS

PAL16R4 Original Equations

```
module Dram_Ctl_Pal;
title '68340 Dram Control Pal'

per_state device 'p16r4';     "10 ns

" Inputs
CLKB                                    pin 1;
CLK                                     pin 2;
CLKB2                                   pin 3;
~DRAMSEL                                pin 4;
TOUT                                    pin 5;
~RESET                                  pin 6;
" Outputs
b2,b1,b0                                pin 15,16,17;
MUXSEL                                  pin 14;
RAS                                     pin 19;
CAS                                     pin 18;
RFQ                                     pin 12;
~DSACK1                                 pin 13;
~OE                                     pin 11;

" Set assignments for state machine
state_bits  = [b2,b1,b0];

" Constant assignments
x,c = .x.,.c.;

" Master State Values
T0 = (state_bits == ^b000); t0=^b000;
T1 = (state_bits == ^b001); t1=^b001;
T2 = (state_bits == ^b011); t2=^b011;
T3 = (state_bits == ^b010); t3=^b010;
T4 = (state_bits == ^b100); t4=^b100;
T5 = (state_bits == ^b101); t5=^b101;
T6 = (state_bits == ^b111); t6=^b111;
T7 = (state_bits == ^b110); t7=^b110;

state_diagram state_bits
    State t0:
         IF (!~RESET) THEN t0 WITH MUXSEL := 1;
         ELSE IF (RFQ) THEN t4 WITH MUXSEL := 1;
         ELSE IF (RAS) THEN t1 WITH MUXSEL := 0;
         ELSE t0 WITH MUXSEL := 1;
    State t1:
         IF (!~RESET) THEN t0 WITH MUXSEL := 1;
         ELSE t2 WITH MUXSEL := 0;
    State t2:
         IF (!~RESET) THEN t0 WITH MUXSEL := 1;
         ELSE t3 WITH MUXSEL := 1;
    State t3:
         GOTO t0 WITH MUXSEL := 1;
    State t4:
         IF (!~RESET) THEN t0 WITH MUXSEL := 1;
         ELSE t5 WITH MUXSEL := 1;
    State t5:
         IF (!~RESET) THEN t0 WITH MUXSEL := 1;
         ELSE t6 WITH MUXSEL := 1;
    State t6:
         IF (!~RESET) THEN t0 WITH MUXSEL := 1;
         ELSE t7 WITH MUXSEL := 1;
    State t7:
         GOTO t0 WITH MUXSEL := 1;
```

```
equations
RAS = ( T0 & !~DRAMSEL & CLK & !RFQ
    # T0 & RAS                      ”Will prevent glitch between T0 and T1
    # T1
    # T2 & !CLK & RAS               ”Will prevent glitch between T2 and T3
    # T4
    # T5
    # T6 & !CLK & RAS)              ”Will prevent glitch between T6 and T0
    & ~RESET;
CAS =
    ( T0 & RFQ
    # T4
    # T1 & CLK
    # T1 & CAS                      Will prevent glitch between T1 and T2
    # T2
    # T3 & !CLK & CAS)              ”Will prevent glitch between T3 and T0
    & ~RESET;
RFQ =
    ( TOUT & CLKB2 & !CLK           ” RFQ not asserted until CLK2 low and
                                    ”  inverter has caused CLKB high
    # RFQ & !T6)                    ” RFQ goes off in state T6
    & ~RESET;
!~DSACK1 =
    ( T1 & CLK
    # T1 & !~DSACK1                 ”Will prevent glitch between T1 and T2
    # T2
    # T3 & !CLK & !~DSACK1)         ”Will prevent glitch between T3 and T0
    & ~RESET;
end Dram_Ctl_Pal;
```

PAL16R4 Reduced Equations

```
MUXSEL := !(b0 & !b1 & !b2 & ~RESET # RAS & !RFQ & !b1 & !b2 & ~RESET);
b2 := !(b0 & !b1 & !b2 & ~RESET
  # RAS & !RFQ & !b1 & !b2 & ~RESET
  # !b0 & b1
  # b0 & b1 & !b2 & ~RESET
  # !RAS & !RFQ & !b0 & !b2
  # !~RESET);
b1 := !(!~RESET # !b0);
b0 := !(!RAS & !RFQ & !b0 & !b2 # !~RESET # RFQ & !b0 & !b2 # b1);
RAS = !(!~RESET
  # !RAS & RFQ & !b0 & !b2
  # !CLK & !RAS & !b0 & !b2
  # !RAS & !b0 & !b2 & ~DRAMSEL
  # !RAS & b1
  # CLK & b1
  # !b0 & b1);
CAS = !(!~RESET
  # !CAS & !b0 & b1
  # !CAS & !CLK & b0 & !b1
  # CLK & !b0 & b1
  # b0 & b2
  # !RFQ & !b0 & !b1 & !b2
  # b1 & b2);
RFQ = !(!~RESET
  # CLK & b0 & b1 & b2
  # !CLKB2 & b0 & b1 & b2
  # !TOUT & b0 & b1 & b2
  # CLK & !RFQ
  # !CLKB2 & !RFQ
```

```
    # !RFQ & !TOUT);
  ~DSACK1 = !(b0 & b1 & !b2 & ~RESET
    # !CLK & b1 & !b2 & !~DSACK1 & ~RESET
    # b0 & !b2 & !~DSACK1 & ~RESET
    # CLK & b0 & !b2 & ~RESET);
```

PAL16L8 Original Equations

```
module Dram_Ctl_Pal2;
title '68340 Dram Control Pal 2'

per_state device 'p16l8';        "25 ns

" Inputs
b2,b1,b0                                    pin 1,2,3;
RFQ                                         pin 4;
~W                                          pin 5;
A22                                         pin 6;
A21                                         pin 7;
A0                                          pin 8;
SIZ0                                        pin 9;

" Outputs
SEL3,SEL2,SEL1,SEL0                         pin 12,13,14,15;
~WE1,~WE0                                   pin 16,17;

" Group bits together for readability
state_bits  = [b2,b1,b0];
Chip_Select = [A22,A21];
SELn        = [SEL3,SEL2,SEL1,SEL0];

" Constant assignments
x,c = .x.,.c.;

" Master State Values
T0 = (state_bits == ^b000); t0=^b000;
T1 = (state_bits == ^b001); t1=^b001;
T2 = (state_bits == ^b011); t2=^b011;
T3 = (state_bits == ^b010); t3=^b010;
T4 = (state_bits == ^b100); t4=^b100;
T5 = (state_bits == ^b101); t5=^b101;
T6 = (state_bits == ^b111); t6=^b111;
T7 = (state_bits == ^b110); t7=^b111;

CS0 = (Chip_Select == 0); cs0 = 0;
CS1 = (Chip_Select == 1); cs1 = 1;
CS2 = (Chip_Select == 2); cs2 = 2;
CS3 = (Chip_Select == 3); cs3 = 3;

sel4 = ^b1111;
sel3 = ^b1000;
sel2 = ^b0100;
sel1 = ^b0010;
sel0 = ^b0001;

RFSH = (RFQ & T0) # T4 # T5 # T6 # T7;

equations

SEL0 =      CS0 # RFSH;

SEL1 =      CS1 # RFSH;

SEL2 =      CS2 # RFSH;

SEL3 =      CS3 # RFSH;

!~WE0  =  ( !RFSH & !~W )          A~WE0 not asserted during refresh
     & ( !SIZ0 # !A0) ;            " ~WE0 asserted only if word, longword, or even byte

!~WE1  =  ( !RFSH & !~W )          " ~WE1 not asserted during refresh
     & ( !SIZ0 # A0) ;             " ~WE1 asserted only if word, longword, or odd byte

end Dram_Ctl_Pal2;
```

PAL16L8 Reduced Equations

```
SEL0 = !(A21 & b0 & !b2
  # A22 & b0 & !b2
 # A21 & b1 & !b2
 # A22 & b1 & !b2
 # A21 & !RFQ & !b2
 # A22 & !RFQ & !b2);
SEL1 = !(A22 & b0 & !b2
 # A22 & b1 & !b2
 # A22 & !RFQ & !b2
 # !A21 & b0 & !b2
 # !A21 & b1 & !b2
 # !A21 & !RFQ & !b2);
SEL2 = !(A21 & b0 & !b2
 # A21 & b1 & !b2
 # A21 & !RFQ & !b2
 # !A22 & b0 & !b2
 # !A22 & b1 & !b2
 # !A22 & !RFQ & !b2);
SEL3 = !(!A21 & b0 & !b2
 # !A21 & b1 & !b2
 # !A21 & !RFQ & !b2
 # !A22 & b0 & !b2
 # !A22 & b1 & !b2
 # !A22 & !RFQ & !b2);
~WE0 = !(!A0 & b0 & !b2 & !~W
 # !A0 & b1 & !b2 & !~W
 # !A0 & !RFQ & !b2 & !~W
 # !SIZ0 & b0 & !b2 & !~W
 # !SIZ0 & b1 & !b2 & !~W
 # !RFQ & !SIZ0 & !b2 & !~W);
~WE1 = !(!SIZ0 & b0 & !b2 & !~W
 # !SIZ0 & b1 & !b2 & !~W
 # !RFQ & !SIZ0 & !b2 & !~W
 # A0 & b0 & !b2 & !~W
 # A0 & b1 & !b2 & !~W
    # A0 & !RFQ & !b2 & !~W);
```

Literature Distribution Centers:
USA: Motorola Literature Distribution; P.O. Box 20912; Phoenix, Arizona 85036.
EUROPE: Motorola Ltd.; European Literature Centre; 88 Tanners Drive, Blakelands, Milton Keynes, MK14 5BP, England.
JAPAN: Nippon Motorola Ltd.; 4-32-1, Nishi-Gotanda, Shinagawa-ku, Tokyo 141, Japan.
ASIA PACIFIC: Motorola Semiconductors H.K. Ltd.; Silicon Harbour Center, No. 2 Dai King Street, Tai Po Industrial Estate, Tai Po, N.T., Hong Kong.