SOFTWARE REFRESHED MEMORY CARD FOR THE MC68000

Prepared by:
Duane Graden
Microprocessor Applications Engineer

This application describes the hardware and software to implement a software-refreshed, dynamic memory card for use in an eight megahertz MC68000 system. This refresh approach consumes less than five percent of processor time. The MCM4116 16K RAM was chosen for this design, but the techniques discussed are applicable to the MCM6664 64K RAM as well.

Refresh techniques fall into two categories, hardware and software. Hardware refresh is more component intensive with little or no overhead in program time, while software refresh has less hardware and more program overhead.

Hardware refresh means that the required circuitry must refresh the dynamic RAM cell with little or no impact on execution of instructions by the processor. Normally, this means accessing the address bus during a dead part of the cycle. Another drawback is the complex circuitry, usually requiring the use of expensive delay lines.

Software refresh means that the processor must execute a software routine to refresh dynamic memory. To accomplish this, an interrupt service routine, such as the level seven interrupt service routine on the MC68000, can be dedicated to refresh the memory. Every time the interrupt is recognized, a hardware enable allows the refresh routine to refresh the dynamic RAM.

TIMING SIGNALS

Timing requirements of MCM4116 RAMs and the MC68000 are easy to match because of the asynchronous nature of the MC68000 bus structure. The MC68000 can wait for the slowest RAM through the use of the data transfer acknowledge ($\overline{DTACK}$) signal. As long as $\overline{DTACK}$ is asserted a setup time before the falling edge of any clock state (S4 or later), it will be recognized during that state. Termination of the access is 1½ clock periods later. Figure 1 is a timing diagram for a read, write, and refresh operation.

The $\overline{RAS}$ and $\overline{CAS}$ signals are the row address and column address multiplex control inputs, respectively, for the seven memory address lines A1 through A7. Since no chip select inputs are present with this dynamic memory, $\overline{RAS}$ is the active low signal that starts a memory access cycle. When $\overline{RAS}$ falls, the row address of the location to be accessed is latched into memory. Similarly, the falling edge of $\overline{CAS}$ latches the column address into memory.

The refresh cycle shown in Figure 1, is known as $\overline{RAS}$-only refresh. Row address select is low, $\overline{CAS}$ is high, R/$\overline{W}$ does not matter, and the row address of the row to be refreshed is present on the seven address lines. Each row of memory requires a refresh cycle to be performed every two milliseconds for data to be retained. For the MCM4116 memory, there are 128 rows and, therefore 128 refresh cycles required every two milliseconds.

HARDWARE DESCRIPTION

Figure 2 is the schematic diagram for a dynamic memory card using MCM4116 memories. This card, when used with a MC68000 system, provides 64K bytes of memory from 32K to 96K of the physical address map.

Memory decoding is done with the upper and lower data strobes and address lines A15 and A16. The data strobes divide the memory into even and odd blocks, respectively. The upper data strobe chip selects even bytes from 32K to 96K by activating a row address select upper (RASU) signal. Address lines A15 and A16, through decoder U2 and gate U4, decode whichever of the two banks of even memory (RAS1U or RAS2U) is selected. Similarly, the lower data strobe activates a row address select lower (RASL) signal.

Column address select ($\overline{CAS}$) is activated on the second falling edge of the eight megahertz clock after $\overline{RAS}$ is asserted by flip flop U9. Both $\overline{RAS}$ and $\overline{CAS}$ are turned off when the data strobes are inactive.

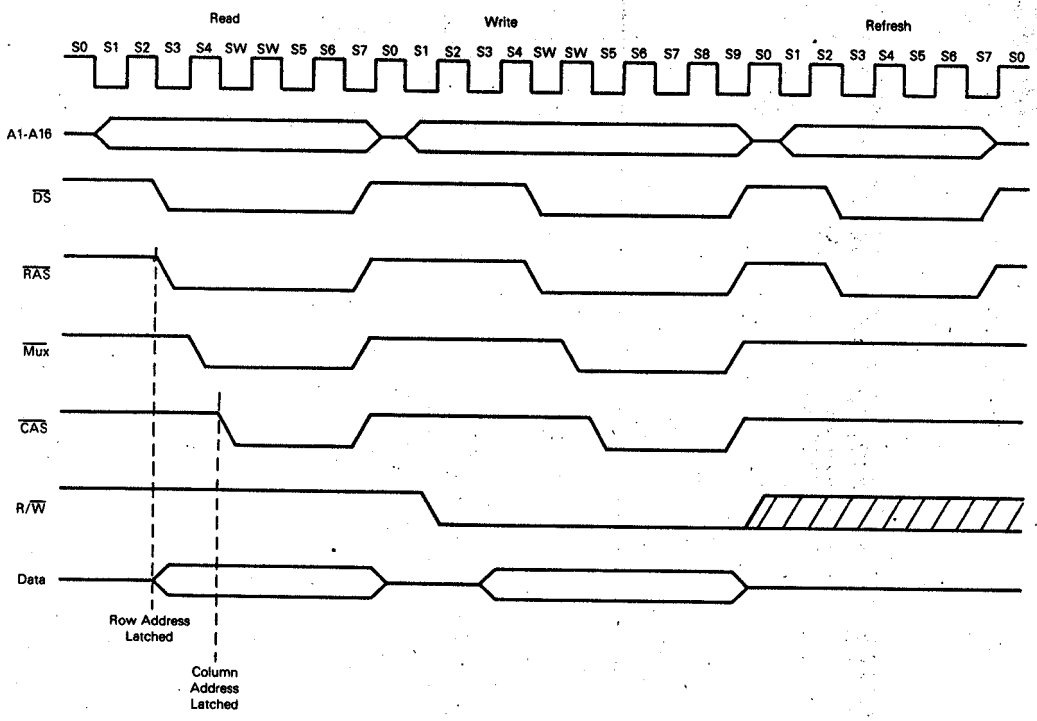


Figure 1. Read, Write, and Refresh Timing Diagrams

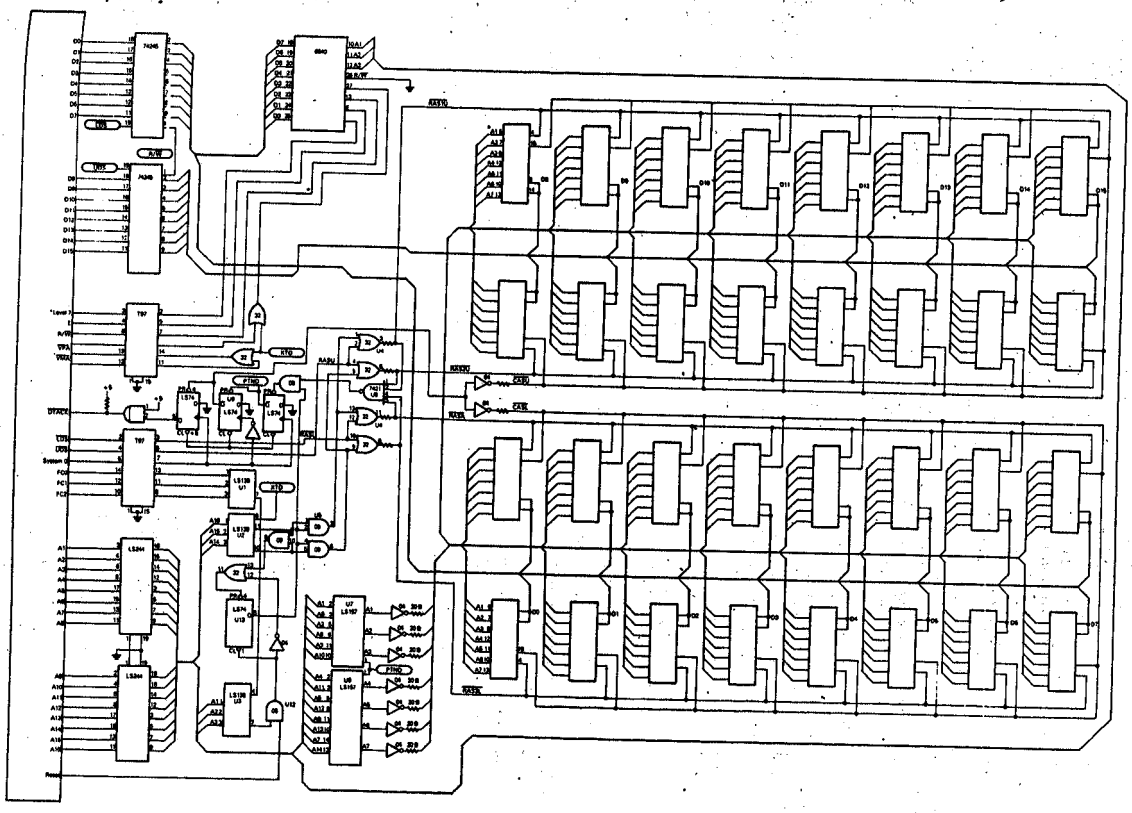


Figure 2. Dynamic Memory Card — Schematic Diagram

Multiplexed addresses for the dynamic memory are supplied by multiplexers U7 and U8. The row address on address lines A1 through A7 is present on the memory address lines until $\overline{RAS}$ is asserted. On the next rising edge of the eight megahertz clock, the column address on address lines A8 through A14 is on the memory address lines. The multiplexed address is valid only when $\overline{RAS}$ or $\overline{CAS}$ is present, making an enable for the multiplexers unnecessary.

Memory refresh is controlled by U11, a MC6840 programmable timer module (PTM). Once programmed, the PTM timer used (the PTM has three timers) causes a level seven interrupt every 1.9 milliseconds (2 milliseconds — routine execution time). This interrupt enables all four banks of memory for simultaneous refresh.

Interrupts with M6800 type peripherals are handled with a reference to the internal vector table. Figure 3 is a schematic of the hardware used with the MC68000 to create a vectored interrupt (level one to level seven). The level present on the $\overline{IPL0}$, $\overline{IPL1}$, and $\overline{IPL2}$ lines is checked against the interrupt level of the processor. If it is higher than the internal level, an interrupt sequence is started. The function code outputs will be high and address lines A1, A2, and A3 will be the vector number of the interrupt being serviced (in this case, all high). Now decoders U1 and U3 (Figure 1) decode the level seven interrupt and generate valid peripheral address ($\overline{VPA}$) to the MC68000 through U13 and U9. The assertion of valid peripheral address causes the internal vector table entry for level seven to be fetched and used as the starting location of the service routine. At the same time, U12 and U13 enable all $\overline{RAS}$ signals and disable $\overline{CAS}$ for refresh of the memories.

OPTIONAL HARDWARE

One situation may occur with the memory card where data might be lost. If the reset button is held closed too long, data could be lost. To prevent this, the circuitry shown in Figure 4 can be added. This provides for a single E cycle reset which will retain the integrity of the stored data.

When power is initially applied to the MC68000, a reset must occur for at least 100 milliseconds after the supply voltage has reached 4.75 volts for proper power-up reset. This means that a one shot or a resistor-capacitor combination should be used to hold the clear pin of the flip flops at or below the logic low level (0.8 volts) for the required time. The E signal will clock the 2-bit counter twice. This presets flip flop U3, removing the system reset. On a non-powerup reset, the reset switch is closed, clocking a low into flip flop U3. Gate U4 provides debounce of the reset switch, allowing only one clock pulse into flip flop U3. Again, E will clock the counter removing reset.

SOFTWARE

Row address select-only refresh is the refresh method used in this application. It is accomplished by a hardware enable (level seven interrupt) and 128 NOPs for the service routine. The level seven interrupt being low enables all four $\overline{RAS}$ signals and disables $\overline{CAS}$. Each NOP increments the address bus to provide the 128 row addresses (0 to 127) needed for refreshing all four banks of memory. Incrementing the address bus accesses and refreshes that row.

However, this has one problem — reset. If a reset occurs just prior to an interrupt for software refresh, data could be lost due to a late or missing refresh cycle. This problem is solved by locating the software refresh routine at the beginning of the reset code. A hardware enable for reset refresh enables $\overline{RAS}$-only refresh in the same way that the level seven interrupt signal did for a normal refresh. In addition, the refresh at reset must load the stack with a valid return address, to return to when the return from interrupt (RTI) instruction is executed at the end of the refresh routine. Figure 5 is a listing of this software with comments to document the reset refresh.

Refresh is enabled at restart by U10 and U13. All $\overline{RAS}$ signals are on and all $\overline{CAS}$ signals are off. Like a normal refresh operation, $\overline{CAS}$ is enabled by the first access to memory after the refresh routine. Software refresh with the MC68000 is an efficient option to implement dynamic RAM without costly delay lines. The application presented here has only a five percent program time overhead.

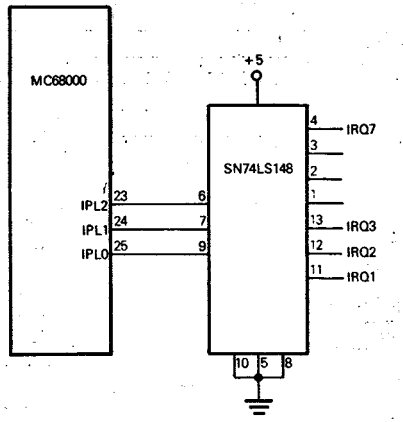
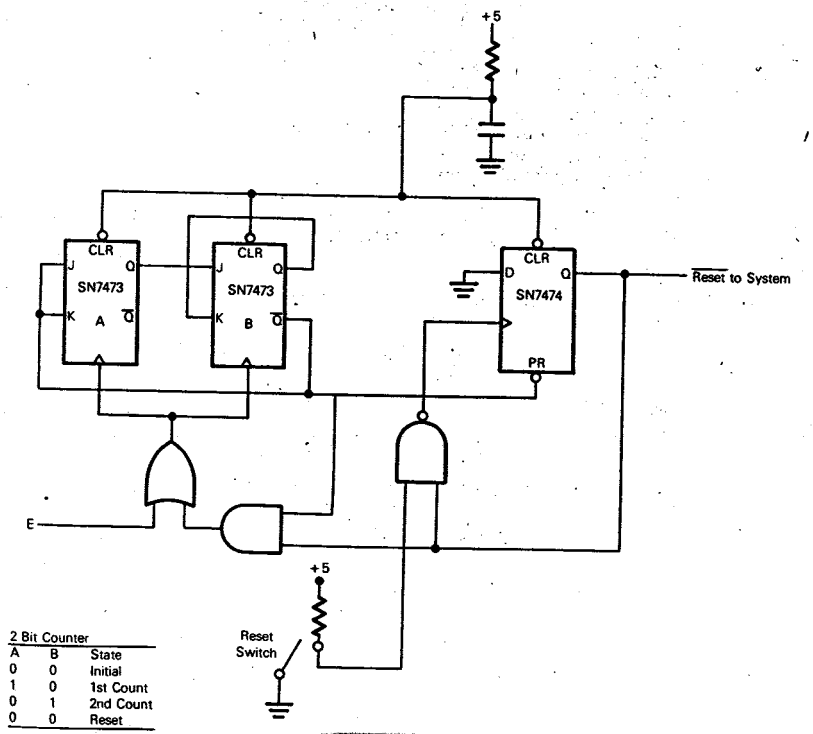


Figure 3. Single-Cycle Reset Circuit — Schematic Diagram



2 Bit Counter

A	B	State
0	0	Initial
1	0	1st Count
0	1	2nd Count
0	0	Reset

Figure 4. Vectored Interrupt Circuit — Schematic Diagram

```
PAGE 001  S68K     .SA:0

DATAR1 EQU $18005
DATAR  EQU $18007
CTR1   EQU $18001
CTR2   EQU $18003
 PEA FIREUP          ****
*                       * LOAD STACK WITH
*                       * USER INITIALIZATION
*                       * AND STATUS REGISTER
 MOVE SR,D           ****
 MOVE D1,-(SP)       *****
 MOVE,B #$FE,CTR2        *
 MOVE,B #$09,DATAR1      * INITIALIZE PIM TIMER
 MOVE,B #$47,DATAR       *
  MOVE,B #$7D,CTR1 *****
*
*                           * LEVEL 7 INTERRUPT
*                           * ENTRY POINT
*
 NOP                 ******
 NOP                     *
 NOP                     *
                         *
                         * 128 NOP'S FOR REFRESH
                         *
 NOP                     *
 NOP                     *
 NOP                 ******
 RTE
FIREUP ******* BEGINING OF USER RESET INITIALIZATION
```

Figure 5. Program Listing