Design considerations for a hardware-refreshed memory card for the MC68000

Though refresh is an essential part of dynamic RAM operation, not all microprocessor units have the appropriate circuitry built in. **S M Said and K R Dimond** introduce refresh hardware for 68000-based systems

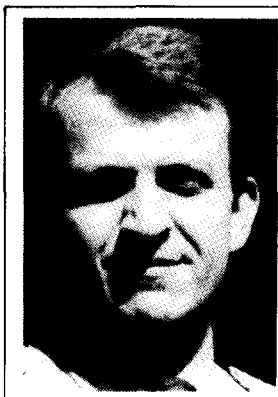
Circuit complexity is the main disadvantage of using hardware rather than software to refresh dynamic RAM. Program overheads, withdrawal of interrupts from use, and increased demands on system components other than the microprocessor militate against the software solution. A hardware-refreshed 256 kbyte dynamic RAM card has been developed for use with 68000-based systems. The design includes a TMS 4500AML controller and a TMS 4164 64k dynamic RAM. The operation of the dynamic RAM is explained. The RAM is interfaced to the 68000 by the system's VME bus. The organization of the memory is discussed, and the steps in a software routine to test blocks of memory are outlined.

microsystems 68000 dynamic RAM refresh

Since the introduction of third-generation microprocessors (eg the Z80) there has been a steady increase in the need for larger RAM for computer systems. Dynamic RAM is often the best overall choice for low-power performance and board density. However, a dynamic RAM needs an extra refresh circuit if one has not been incorporated within the microprocessor unit. Motorola's MC68000 is an example of a microprocessor that lacks such a circuit.

In this paper we describe the hardware and the software to implement and test a hardware-refreshed 256 kbyte dynamic memory card for use in an

Electronics Laboratories, University of Kent at Canterbury, Kent CT2 7NT, UK



Keith Dimond graduated in electrical engineering from the University of Manchester Institute of Science and Technology, UK, where he also undertook research. He was then a scientific officer at Government Communications Headquarters, Cheltenham, UK. Dimond joined the Electronics Laboratory at the University of Kent in October 1971. He became Senior Lecturer there in 1981. His interests are in CAD tools for digital and system design.

S M Said was born in El-Fayom, Egypt, in 1949. He received his BSc in electrical engineering from the Military Technical College of Cairo, Egypt, in 1971. Between 1971 and 1976 he was involved in several communications projects for the Egyptian Army. Then from 1977 to 1981 he was a lecturer at the Military College. Since 1982 he has been working at the University of Kent at Canterbury, UK, for his PhD in the application of 16-bit microprocessors in digital communication.

0141-9331/84/08413-04 $03.00

MC68000 system. Since the MC68000 processor does not support dynamic RAM control directly[1], an external controller is needed. In the design presented here the TMS 4500AML device is used.

HARDWARE DESCRIPTION

In general, refresh techniques fall into two categories, software refresh and hardware refresh.

Software refresh means that the processor must execute a software routine to refresh the dynamic memory. A high-priority interrupt service routine, such as the level seven interrupt service routine on the MC68000, must be dedicated to refresh the memory. A realtime clock is used to generate interrupts. Every time the interrupt is recognized, the processor allows the refresh routine to refresh the memory.

Hardware refresh means that the required circuit must refresh the dynamic RAM cells without affecting the execution of instructions by the processor.

Software refresh has the following disadvantages:

- Because the highest priority interrupt is used for the refresh routine, the user cannot employ this interrupt for any other purpose.
- The user software must include software for the refresh process; this results in more program overhead.
- When a direct memory access controller (DMAC) is used, the MC68000 will be halted. Therefore the DMAC must be able to perform the software refresh instead of the processor.

The only drawback of hardware refresh is the complexity of the circuitry. In the work reported here, hardware refresh was implemented in preference to software refresh because of the latter's shortcomings, mentioned above.

Figure 1 shows a block diagram of the designed dynamic memory card. It uses the TMS 4164 (64k dynamic RAM) and the TMS 4500AML (dynamic RAM system controller) to control and refresh the dynamic memories.

Dynamic memory operation

Two technologies are used to implement RAM: static and dynamic.

The static RAM stores a bit of information within a flipflop. The contents remain stable forever, as long as power is available.

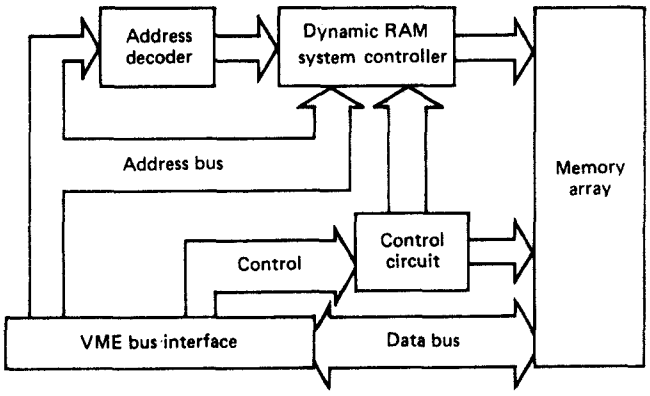


Figure 1. The implemented memory board — block diagram

The dynamic RAM stores a bit of information as a charge. It uses the gate–substrate capacitance of an MOS transistor as an elementary memory cell. The advantage of a dynamic RAM is that the elementary cell is smaller than a static RAM flipflop, resulting in a much higher density. For example, a 64 kbit dynamic RAM resides on the same chip area as a 16 kbit static RAM.

The dynamic RAM stores its data in the form of a charge which, like any charge stored in a capacitor, leaks away. Hence most of the charge is lost within a few milliseconds. For the dynamic memory to retain valid data, this charge must be periodically restored.

The process by which data is restored is called refreshing. A refresh is performed on a row of data each time a read or write cycle is performed on any bit within the given row. This process is defined as the refresh cycle.

The internal memory organization does not correspond to its external appearance (TMS 4164 organization is 65536 × 1 in a 256 row × 256 column matrix). Thus only 256 operations will be needed for the complete refresh of this memory. For the TMS 4164, the refresh operation must be performed[2] at least every 4 ms. During this period each of the 256 rows must be strobed with $\overline{RAS}$ to retain data. Therefore, to refresh all rows within 4 ms, a refresh cycle must be executed every 16 μs (4 ms/256).

The TMS 4164 chip is a 64 kbit dynamic RAM. Thus it needs 16 bits to address all its storage cells. The row and column addresses are strobed into the memory by the two negative-going clocks ($\overline{RAS}$ and $\overline{CAS}$). Hence eight row-address bits are set up on A0–A7 (which represent address lines A0–A7) and latched onto the chip by $\overline{RAS}$. Then eight column-address bits are set up on A0–A7 (which represent address lines A8–A15) and latched onto the chip by $\overline{CAS}$. On a Write cycle, data is written into the device via the (D) input after the $\overline{RAS}$/$\overline{CAS}$ sequence described.

Using the TMS 4164 dynamic RAM, it is necessary to connect eight of these chips in parallel to implement a 64 kbyte memory. Data into (D) and data out (Q) of each chip are connected together to one of the eight data lines (D0–D7 if lower byte or D8–D15 if higher byte).

Dynamic RAM controller

The TMS 4500AML dynamic RAM controller[3] is used to simplify the interface of the dynamic RAMs to the MC68000 microprocessor system. The controller contains eight two-to-one multiplexers. The address lines A0–A7 are connected to one set of inputs, and the eight address lines A8–A15 are connected to the other set of inputs. The outputs of the multiplexers are connected to the eight address lines of the dynamic RAM. The dynamic RAM system controller generates the strobe signals ($\overline{RAS}$ and $\overline{CAS}$) needed by the memory to decode the address as described.

The TMS 4500AML refresh may be initialized internally or externally. This circuit uses the NE555 timer to generate refresh requests externally. The dynamic RAM controller operates from the microprocessor clock ($\overline{SYSCLK}$), and it has an 8-bit refresh counter that generates the 256-row addresses needed for refresh.

Interface via the VME bus

The MC68000 processor is interfaced to the dynamic RAM card via the VME bus[4]. Figure 2 shows the circuit diagram of the designed memory board. This design forms of a page of 128 kword in an eight-page system that can give up to 1 Mword of memory. The memory decoding of this page is done with address lines A18–A23 and the address strobe ($\overline{AS}$).

The memory is divided into two even and two odd 64 kbyte blocks. The odd bytes (D0–D7) and the even bytes (D8–D15) will be selected by $\overline{LDS}$ and $\overline{UDS}$ respectively. If both $\overline{LDS}$ and $\overline{UDS}$ are active low, which means that the microprocessor is requesting 16-bit word access, both odd and even bytes are enabled (D0–D15).

The controller provides a handshake output line (RDY).This indicates that refreshing is in progress. In the circuit shown, the output ($\overline{DTACK}$) signal to the MC68000 processor is a function of $\overline{UDS}$, $\overline{LDS}$, RDY and a time delay. The delay is an integral number of cycles of the system clock $\overline{SYSCLK}$. This prevents the processor from accessing the memory during a refresh cycle or before data is valid.

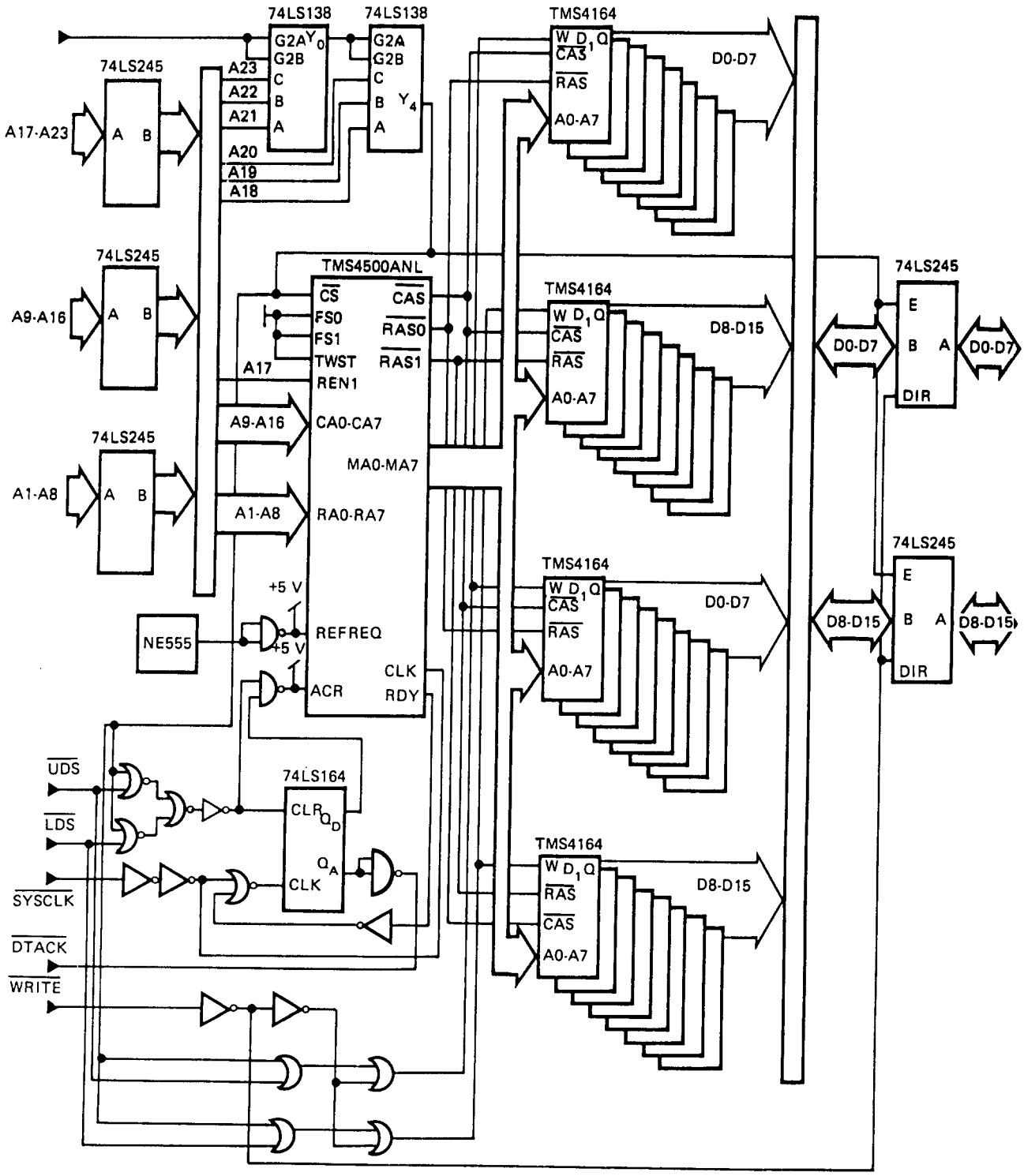


Figure 2. Memory board circuit diagram

SOFTWARE DESCRIPTION

This section outlines a memory diagnosis procedure for testing memory systems such as that described above.

To test the hardware-refresh process of the memory board, which must be performed at least every 4 ms, a delay of 1 s is allowed between the loading of a block of memory and its verification. The memory test routine used in our work is executed in less than 2 min for a 256k × 8 memory.

The program is run to test a block of memory. A complete test cycle consists of the following five tests:

- *Bit-stuck test* Fill the memory block with ones, verify it by reading, and then change all ones to zeros and do the verification again. By this test, all the data bits are checked to be sure that none of them is stuck high or low.
- *Checkerboard test* First fill all the memory block with alternate bits of ones and zeros (55 in hex) and do the verification. Then change the one bits to zeros and the zero bits to ones (AA in hex) and do the verification again. This test complements all bits on every byte, but adjacent bits are always different.
- *Adjacent-bits-shorted test* Fill every eight successive bytes of the memory block with 01, 02, 04, 08, 10, 20, 40 and 80 (in hex), and then do the verification. By this test only two data lines are changed at a time. Thus every two adjacent bytes will always have different values, and the sequence will be repeated every eight memory locations.
- *Walking-left-bit test* Fill every eight successive bytes of the memory with 01, 02, 04, 08, 10, 20, 40 and 80 (in hex), up to the end of the memory block. Then do the verification. This means that, in every byte, only one bit is a one while all the other bits are zeros and every two adjacent bytes always have different values. The test is repeated eight times. Each time, the one bit is shifted to the left inside the byte. Since the data is changed in every byte on all the memory block, the test checks both data and address buses.
- *Walking-right-bit test* This test will examine the memory block in the same way as the previous test. The only difference will be that the one bit is allowed to move only to the right inside the byte.

The above description of the five tests shows that this program exercises the memory in different ways to indicate any possible error. If an error occurs during any of the tests, the program will be terminated, printing an error message that indicates the address of the error, the actual data value, and what the data should have been.

MEMORY PERFORMANCE

As mentioned before, dynamic RAM needs refreshing at constant intervals. Therefore there will be occasions when the CPU requests memory access but the RAM controller delays the access until the current refresh cycle (generated internally by the RAM controller) is terminated. However, this delay only increases the average execution time by 0.07%.

ACKNOWLEDGEMENTS

S M Said would like to acknowledge the Egyptian Government for its financial support, and to thank E Horne and H Twyman for their help and cooperation.

REFERENCES

1 *MC68000 user manual* Motorola, USA (1982)
2 *MOS memory data book* Texas Instruments, USA (1982)
3 *TMS 4500A dynamic RAM controller user manual* Texas Instruments, USA (1983)
4 *VME bus specification manual* United Technologies Mostek, USA (1982)