

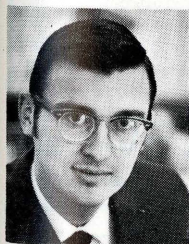
Much faster than drums and less costly than random-access memories, the MOS shift register has become an attractive option for many memory functions. Described here are several means of optimizing access to stored data and maximizing effective storage capacity in CRT display memories and in file memories

Operation and Application of MOS Shift Registers

Marcian E. Hoff, Jr. and Stanley Mazor

Intel Corporation
Mountain View, California

One of the first complex MOS integrated circuits to find extensive acceptance is the MOS shift register. Two properties of MOS IC technology are uniquely compatible with the design of shift registers: the high impedance associated with the gate circuit permits temporary storage of charge on the parasitic capacitances, and MOS technology permits realization of bidirectional transmission gates which have zero dc offset. With the transmission gate, a gate-node may easily be connected to or disconnected from other points in the circuit.



Marcian E. Hoff, Jr. is currently manager of applications research at Intel Corp. Earlier, as a research associate, at Stanford University, he worked in adaptive systems and digital computer applications including memistors and digital circuits, CRT and tape drive computer interfaces, and converter design. He received a BEE from Rensselaer Polytechnic Institute and MS and PhD degrees from Stanford University.



Stanley Mazor is a systems designer and applications engineer at Intel Corp. His previous experience includes scientific, commercial, and systems programming; research on pattern recognition and computer random number generators; development and use of computer-aided design programming systems; design of computer arithmetic and character processors; and studies of hardware/software tradeoffs.

The shift register structure has layout advantages as well. Few interconnections are required, basic shift register stages may be made using little silicon area, and no decoding or other "overhead" circuits need be placed on the chip. These features make the shift register one of the potentially lowest cost semiconductor memories on the market.

Shift register memories already have extensive application in computer display-terminals, where the data to be displayed are recirculated through the shift register memory in synchronism with the CRT display. Also, shift registers are often used in electronic calculators as memory elements. These devices may also be used for small buffer memories where access time may be sufficiently long that the circulation time is not a restriction. For example, in printers, card readers, and other computer peripherals, the MOS shift register may be used as a low-cost buffer memory. By providing address counters and a comparator, the shift register is often used as a low-speed random-access memory.

Operation

There are many variations on the basic shift register. A typical MOS shift register circuit, i.e., half of a dual 100-bit shift register, is shown in Fig. 1. Each bit of the shift register requires six MOS devices. Note that the devices of bit 2 have been labeled Q_{2A} through Q_{2F} . The input to the stage is the charge on the gate of Q_{2A} , deposited by the previous stage. When clock ϕ_2 goes negative (for p-channel devices), devices Q_{2A} and Q_{2B} form an inverter stage. If the charge on the gate of Q_{2A} is sufficiently negative to cause it to conduct strongly, the node common to Q_{2A} , Q_{2B} , and Q_{2C} will approach

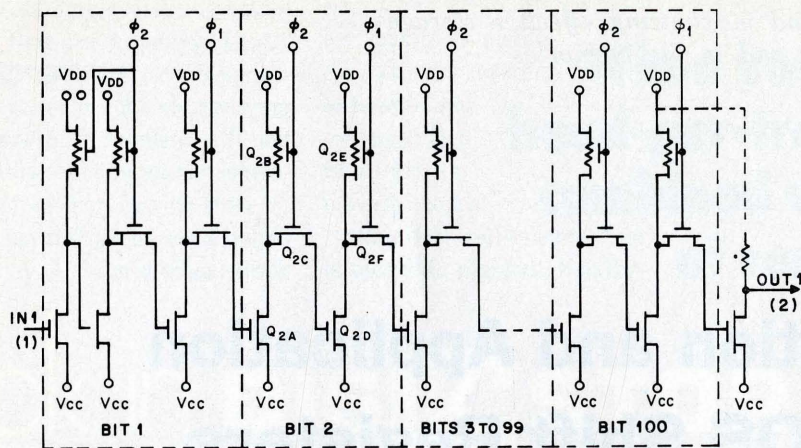


Fig. 1 MOS shift register

V_{CC} (a positive level). However, if the charge at the gate of Q_{2A} is positive enough to leave Q_{2A} cut off, when ϕ_2 becomes negative, this node will approach V_{DD} , the negative bias.

At the same time, when ϕ_2 goes negative, Q_{2C} conducts, charging the parasitic gate capacitance of Q_{2D} to the same potential as the node common to Q_{2A} , Q_{2B} , and Q_{2C} . When ϕ_2 is removed, the gate of Q_{2D} retains its potential. Pulsing ϕ_1 negative then transfers and inverts the data, depositing them at the input to the next stage.

In the circuit shown, an output device is provided so that charge need not be retained on external leads. This output buffer also acts as an inverter. To make input and output levels compatible, an input inverter is provided. In Fig. 1, the data is transferred to the output during clock phase 1. Thus the data is available during clock phase 2. Data at the input must be available prior to and during phase 2. Fig. 2 is an example of a dual 100-bit shift register using this circuit. Only eight connections are required for the dual shift register.

The circuit shown is a dynamic shift register, using only capacitive storage. To retain the data stored in the register, the rate at which the data are circulated (or clocked) through the register must not fall below some minimum, usually approximately 1 kHz at room temperature. Static shift registers may also be manufactured using MOS technology. The static register can retain the data indefinitely if stopped at the proper point in the clock cycle. However, static shift register cells are significantly larger than the dynamic cells and consume much more power. For these reasons, not only are static registers more expensive, but static registers are usually slower than dynamic registers.

Earlier designs of MOS shift registers were realized with standard high-threshold MOS processing. Relatively large voltage swings were required, with typical supply voltages being 15 to 20 V. Clock voltages were usually approximately 30 V. However, the use of new technology, such as the silicon gate process,* permits supply voltages of 10 V and clock amplitudes of only 15 V. More im-

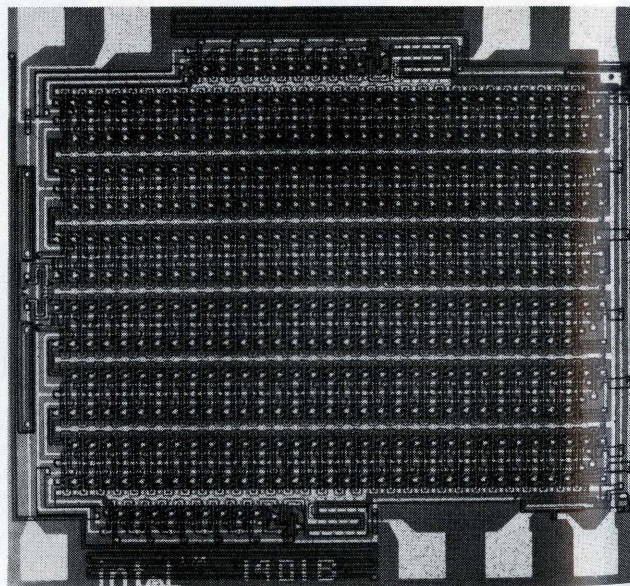


Fig. 2 Microphotograph of dual 100-bit MOS shift register

portant than the supply voltage is the capability for interface with TTL levels that low-threshold MOS offers.

With the shift register of Figs. 1 and 2, when pin V_{CC} is biased at +5 V and pin V_{DD} is biased at -5 V, a TTL gate (biased between ground and V_{CC}) can drive the input pin of the shift register; and by using a suitable resistor on the output terminal, the output can drive a TTL load. The connections are shown in Fig. 3. Here, the clocks swing between +5 and -10 V. The positive swing should approach V_{CC} with a tolerance of +0.3 V, -1.0 V, so that no substrate current flows and no transmission gate remains conductive during the opposite

*L. L. Vadasz, A. S. Grove, T. A. Rowe, and G. E. Moore, "Silicon-Gate Technology," *IEEE Spectrum*, Oct 1969, pp. 28-35

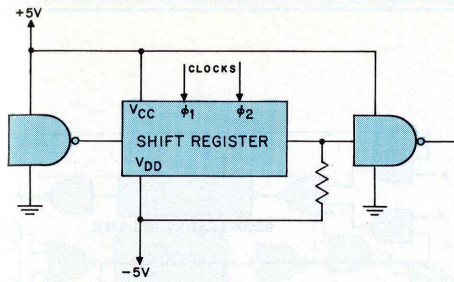


Fig. 3 Input and output connections for low-threshold MOS shift registers

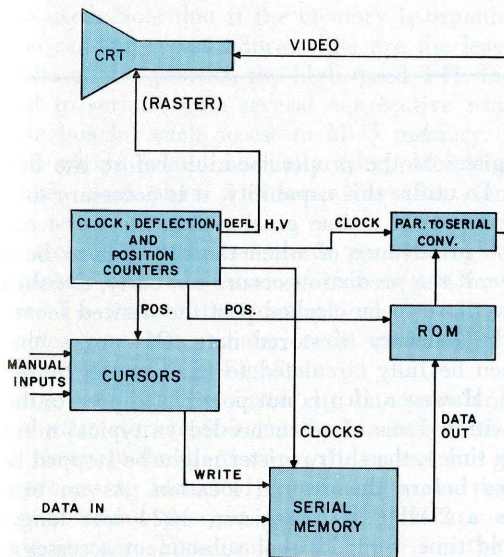


Fig. 4 CRT display terminal

phase of the clock. Either of these conditions can effectively reduce the ability of the gate circuits to store charge.

While some variation in power supply and clock amplitudes may be tolerated, if they are of insufficient magnitude the range of frequency of operation may be reduced at both ends. Excessive supplies may result in problems due to field inversion, i.e., creation of conducting channels under leads carrying the excessive

voltage. These channels correspond to undesired connections between points of the circuit.

The clock signals for driving MOS shift registers must be capable of producing the desired voltage transitions without overlap. That is, the two phases should never simultaneously be more than 1 to 2 V more negative than V_{CC} . The clock drivers must be capable of driving the capacitive loads presented by the shift registers. Even silicon-gate shift registers, which have lower clock capacitances per bit than those realized with any other technology, present capacitances of about 0.1 to 0.2 pF per bit. Thus, for a system of 10,000 bits, a capacitance of 1000 to 2000 pF can be expected. Allowing that a 15-V swing in 150 ns corresponds to 0.1 mA per picofarad, to drive a 10,000-bit memory with 150-ns rise and fall times the clock drivers must provide 200-mA drive currents. (These rise times would limit operation to 1 MHz or less.)

The clock drivers must not only be capable of driving the capacitive load, but must do so without excessive ringing, because ringing might result in violation of the restrictions mentioned previously. The power dissipation associated with the driving of the capacitive load must also be considered. In the example cited above, each clock driver draws 200 mA from the power supply during the 150-ns charging transition. At 1 MHz, the average current is 30 mA, corresponding to a dissipation of 450 mW for a 15-V supply. The average power drawn from the supply is independent of the rise time of the clock drive signal but is proportional to the clocking rate. The power may be dissipated in the drive transistors, in the shift register, or in circuit resistance. Resistors may be placed between the clock driver output and the shift registers to reduce dissipation in the semiconductors and to provide damping. It is possible that inductors might be used to reduce the power dissipation, but the circuits necessary to effect the energy exchange between inductor and shift-register capacitance are difficult to design.

Applications

In many ways, the shift-register memory behaves much like an acoustic delay line memory or a drum memory. Unlike these memories, the shift register can be instantaneously started and stopped. It may remain stopped for approximately 1 ms. Unlike the drum or acoustic delay line, the shift register is easily synchronized with many kinds of equipment. For example, in a CRT display memory, the clock may be stopped during the retrace, then restarted for the next line. In this way, additional buffering can be avoided and yet none of the storage is wasted.

One of the most important uses of shift registers is in the CRT display terminal. Fig. 4 shows a block diagram of the display portion of such a terminal. The serial memory is usually realized with MOS dynamic shift registers and stores several hundred to a few thousand characters. To minimize the amount of memory, the characters are usually stored in a compact code of six to eight bits each. The display on the face of the CRT is usually 5 x 7 or 7 x 9 dot matrix, so that to display the character, the code must be converted from the compact code to the 35- or 63-bit display code. This conversion is performed by the read-only memory.

Several types of CRT raster scans may be used with this type of display. If a standard video scan is used, the serial memory must present the same data for several lines in succession, i.e., for each line on which part of that row of characters appears. Fig. 5 shows one way to organize the memory so that no storage is wasted. One register is used as a recirculating register as well as a continuation of the main memory. In normal mode, control signal Y is true, X is false, and both clock sets are operated in synchronism. When data to be recirculated have been loaded into the recirculation register, Y is made false, X made true, and the clocks to main memory are then recirculated as many times as necessary by applying a sufficient number of clock pulses (ϕ_{1B} and ϕ_{2B}). Note that the length of the recirculating register need not be exactly the same as the number of characters to be displayed—the shift register can be longer if there is sufficient time during retrace to shift the unwanted data past the output port.

Editing of data in the shift register memory can be accomplished by providing an extra stage or two which can be switched in and out—effectively lengthening or shortening the shift register. Thus, when a bit must be inserted between two other bits, the data in the register are shifted until the two bits between which the new data must be inserted are adjacent to the extra stage. The data to be added are placed in the extra stage and the connections changed to include the extra stage in the shift register chain. To maintain the length of the shift register at a constant value, some other character must be deleted. By clocking the data until the character to be deleted is in the extra stage, and then disconnecting the extra stage, the deletion can be effected. Fig. 6 shows one circuit which may be used to provide this feature. When A is false and B and X are true, the extra stage, realized by a D flip-flop, is included. When A is true and B is false, the stage is not included. Inputs Y and D are for loading the D flip-flop with data from an external source. The use of AND-OR-INVERT gates results in the data in the flip-flop being inverted, so the \bar{Q} output is used.

Systems Implications

The MOS shift register promises to offer a low-cost, high-performance memory, that falls, both in access time and price, between the drum memory and the main frame ferrite-core or semiconductor memory. The shift register may be used to realize file memories with performance in this intermediate range. While most drums show access times in excess of several milliseconds, and mainframe computer memory access time is usually under 1 μ s, shift register memories for bulk data storage will probably show access times in the order of 50 to 500 μ s.

Shift-register access time may be reduced below these figures in several ways not available with drum systems. For example, the shift register can be stopped for periods up to 1 ms. If access to the shift register memory can be predicted in advance, it may be possible to bring the

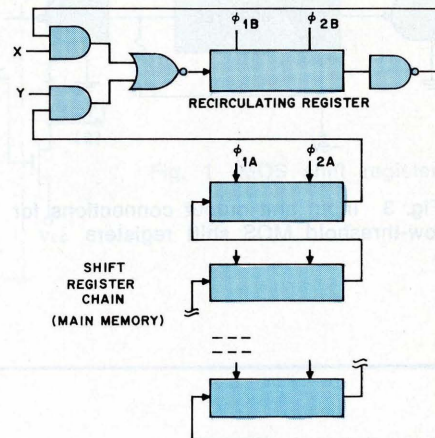


Fig. 5 Serial memory with recirculating loop (1-bit path shown)

shift register to the proper location before the data are needed. To utilize this capability, it is necessary to anticipate the access at a time greater than the worst case access time in advance of when the access is to be made. However, if the prediction occurs too early, the shift register may have to be clocked past the desired location to retain the integrity of stored data. Of course, the data must then be fully circulated to again reach the desired location. However, if it is not possible to predict the next access within 1 ms of when needed (a typical maximum stopping time), the shift register might be stopped several addresses before the desired location. As an example, consider a 2-MHz shift register, 1024 bits long, with 1-ms hold time. Suppose that subsequent accesses to the shift register memory can be predicted as occurring some time between 0.5 and 10 ms in the future. Normal worst case access would be 0.5 ms without prediction. By stopping 10 addresses in advance of the desired location, worst case access is reduced to 5 μ s, and yet the desired location is available within 5 μ s for at least 10 ms.

When the capacity of one shift register is insufficient to store a given string of data bits, several shift registers may be placed in series as shown in Fig. 5. The series cascade, while the least expensive organization, increases the access time to data in the memory.

An alternate structure is shown in Fig. 7. This circuit permits the outputs of several shift registers to be gated selectively onto the data bus. The select and read and select and write signals may be derived by decoding address bits in addition to those used for serial memory address. By using floating collector gates on the register outputs to drive the data bus, the wired-OR connection

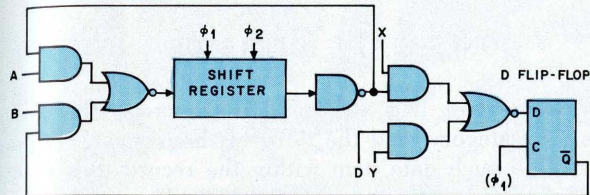


Fig. 6 Circuit for adding and deleting data in shift register memory

may be used. Note that if the memory is organized such that the random-access address bits are the least significant bits of the address, the high-speed TTL logic may be used to serially gate several consecutive words over the data bus for each access to MOS memory. This sequential gating would normally take place during the time when clock phase 2 is active (or after clock phase 1).

The technique described above permits high rates of data transfer and reduces access time to that for a single register. To achieve this connection typically requires four gates per shift register, even when the wired-OR connection is used as in Fig. 7.

In many applications involving data transfer between file and main storage, it is much more important to make a read access to the file than a write access. For example, when processing blocks of data or program, the central processor cannot proceed until the data are available in main storage. Thus, unless the need for the data is anticipated, or multiprogramming is used, the processor is idle until the data are available. However, the processor can continue when data are waiting to be written. The main disadvantage of a long wait is that the more expensive mainframe storage space is occupied by the data that are waiting.

There are many applications in which data are read from a memory file more frequently than they are written into the file. Many data files, tables, and subroutine programs are used primarily in the read mode.

The time for access to data in a shift register memory can be reduced by providing multiple ports. However, each read-write port requires the equivalent of about four gates, as shown in Fig. 7. Even more gates are required if editing must be done when writing; but, if a port provides for reading only, a single gate is sufficient. Fig. 8 shows a shift register memory in which four read-only ports have been used with one read-write port. For this

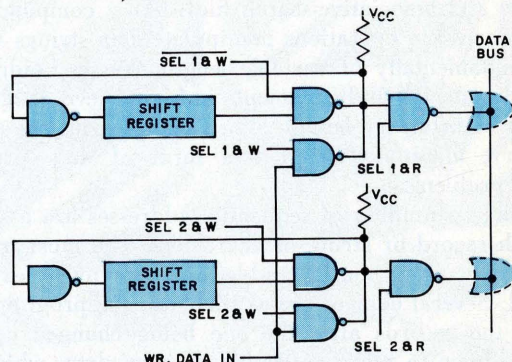


Fig. 7 Combination serial and random-access memory

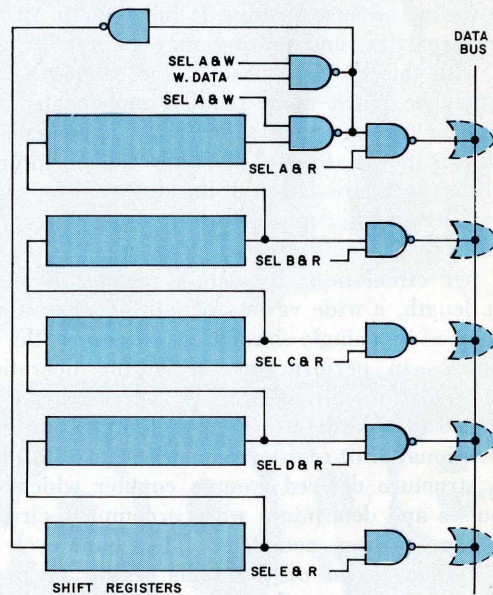


Fig. 8 Shift register memory with multiple read ports and single write port (pull-up resistors not shown)

organization, read-access time is reduced to one fifth of the write access time by using twice as many gates as for the single port version. This structure can also be used for the kind of multiple-access multiplexing described above.

Some less conventional but promising uses of shift register memories are for variable field-length array storage and associative-search files. Many computer and data processing operations manipulate data strings which are fundamentally of variable length. Names, addresses, textual material, and statements in higher level languages are all of arbitrary length. When using a random-access memory, manipulation of this form of data presents severe problems.

Usually a number of sequential addresses are assigned to each record or string of characters. The most trouble occurs when data must be added to a previously existing record. Several choices are available to the programmer: all of the records after the one being changed can be moved down to make space for the new data, which results in a great amount of data manipulation and can be very slow; one or more records can be moved to provide the space, which results in these records being located away from related data; the programmer can provide extra space in each record, which wastes this space; and the changed record can be made to contain a link to another location in memory, which can result in having to make many accesses to memory to find all of the data if extensive editing has been performed.

The bookkeeping overhead and process-time manipulations for these techniques can excessively burden a system. However, because shifting is inherent in MOS shift register memories, and editing may be readily accomplished with this kind of memory (see above), the first choice may be much more readily implemented with a shift register memory than a random access memory.

As shown in Fig. 6, one character can be inserted in a record for each circulation of the shift register memory. By providing an insertion-deletion register of several bits (characters) in length, several bits may be added or deleted per circulation. If such a register is made of variable length, a wide variety of editing operations can be performed in a single circulation. Of course, the single-bit register can perform several editing operations if several circulations are allowed. A source register might be provided to hold data to be used for this editing.

Conventional shift-register memories have an absolute address structure derived from a counter which counts clock pulses and determines when a complete circulation of the data has been completed. The count cycle is, of course, chosen to be of the same length as the shift register chain being used. Such a memory can be organized much like a conventional magnetic drum memory, except with the features of higher operating rates, faster access, and variable clock-rate operation.

Other addressing schemes are also possible. For example, if for each set of data track a marker track is provided, the marker track may be used to indicate the start of each record in the track set. (A typical configuration would use six to eight data tracks in each set, i.e., one for each bit in the code for the characters.) Variable length records may then be addressed by counting marker bits.

An even more powerful scheme permits searches on the contents of data records themselves. One or more

marker tracks might be used to indicate starting points of records, numbers of characters per entry in each record, and the nature or category of the entry. As an example of such a record, consider a possible entry in an employee's record:

JONES, E. A. OPERATOR 8.75

The first item lists the employee's name, the second his job category, and the third his hourly rate. Suppose that each such data item within the record uses at least 6 characters and not more than 63. The marker track can contain six bits per item, stating the number of character positions used by that item. To search a track, a processor reads these codes and adds up the numbers of bits allotted to each item. A second marker track may be used to designate data categories—name, job classification, etc. With this coding scheme, file searches may be performed on a purely content-addressable basis. So that all sets of tracks can be simultaneously searched, a separate processor might be provided for each track. This processor would most likely be a single MOS chip with a comparator and some temporary storage.

Additional features could be added to a file system of this type by adding a master file-processor. One of the functions of such a processor would be moving data in the file so all tracks have some editing space available and so that some empty tracks are available for long records. Other functions which might be performed by such a processor are resolving multiple finds in the content-addressable array and establishing priorities for delivering and writing data. This file organization might be made easily compatible with storage on magnetic tape, so that more rarely used data need not always reside in the file. The variable-speed operation of the shift register permits easy synchronization for input from and output to a magnetic tape transport.

While the system described is purely hypothetical, it is well within the realm of technical feasibility. The cost of the system should be significantly less than a large core storage unit of comparable bit capacity. Access times, even using full associative search, should approach an order of magnitude less than any mechanical drum memory. It is somewhat difficult to evaluate such a system in terms of existing hardware, because it would perform functions not now readily available.

Future Developments

Not only may the cost of shift registers be expected to fall due to improvements in yields and manufacturing methods, but new configurations may reduce the costs of the peripheral circuits. Many of the circuits shown as peripheral in this article may be included on the shift register chip in future designs. Larger and higher speed structures should also be made available as technology improves. When systems designers become familiar with the possibilities of this unique MOS device, even wider ranges of application may be found.