Adder on a chip: LSI helps reduce cost of small machine

A complete 8-bit arithmetic unit on a single chip of silicon reduces need for random logic and, by subtracting without complementing, can be doubled up for words of any length

By Lee L. Boysel
Fairchild Semiconductor Division, Mountain View, Calif.

Large-scale integration becomes feasible for small, slow computers when applied to the computer's arithmetic unit. Previously the technique has been considered only for large machines with many identical circuits, because the main LSI effort has been aimed at such circuits—memories, registers, and data paths. But now small computers—for applications such as process control, data acquisition, testing, or education, where cost is a vital factor—can take advantage of LSI. Their entree is an eight-bit parallel arithmetic unit, containing 200 metal oxide semiconductor gates interconnected on a single chip of silicon.

Large-scale integration is especially valuable in an arithmetic unit, which is made of several identical parts that operate in parallel. Virtually every computer operation involves the unit, even if only to pass a number from one register to another.

However, to achieve the necessary low cost of a small, slow machine and still realize the benefits of LSI, only 10% or less of the total circuits should be random logic—circuits that perform specific functions, such as sign control, division stop, overflow error, and comparisons. Only one of each of these random logic functions, which contrast with repetitive data-flow logic functions, is required in any computer.

Previous approaches to LSI were effective on only about 25% of the system—the remaining 75% being composed of random logic, which was either unsuitable for LSI or too costly for it. But a large-scale integrated arithmetic unit and a read-only memory can reduce the random logic to well below 10% in a small machine. Working together, they perform almost all the needed functions.

With the arithmetic unit's unusual algorithm for addition and subtraction, much of the need for control by random logic was eliminated. The read-only memory approach to control cuts random logic significantly, because it can be used for combinational and sequential logic, which in turn can be applied to control functions.[1] Thus a slow, fully parallel machine can be built that is, curiously, less expensive than a serial machine, because its proportionate amount of control is substantially less. Traditionally, because parallel machines contain more circuits, they are more expensive—their cost being justified by their higher performance. Serial machines usually sacrifice speed for low cost.

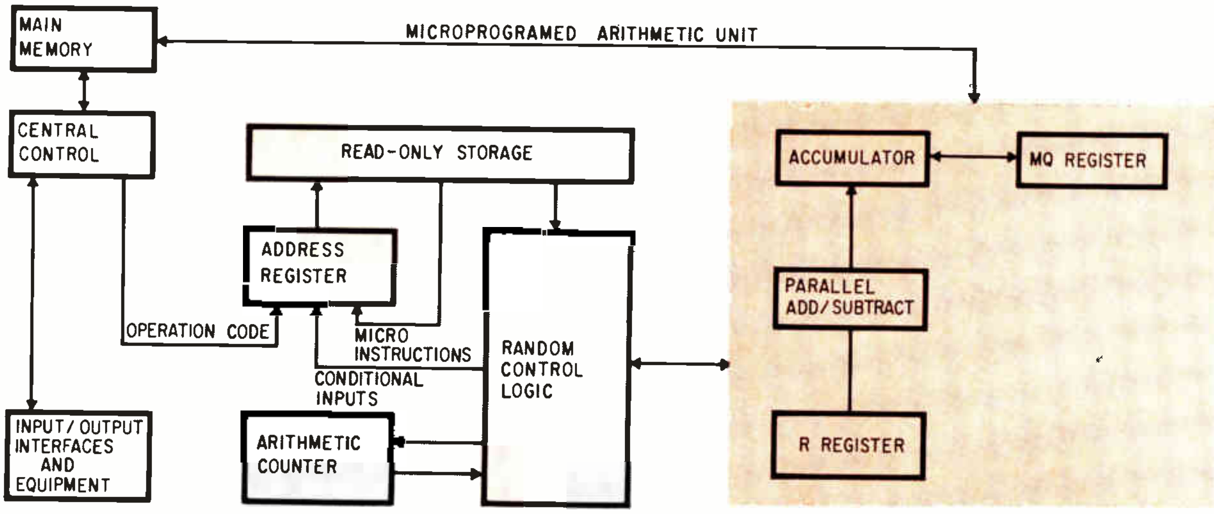
A 16-bit parallel system has been constructed with the new arithmetic unit and a monolithic read-only memory; it has only about 1% random logic.

Ideal repetition

The arithmetic unit of any computer can be designed as a series of identical blocks, each of which performs a complete arithmetic function on a single bit. Several blocks can be fabricated inexpensively on a single chip with LSI techniques. This unit can become a subsystem of a larger system with any of several word lengths.

The author

Lee L. Boysel is the supervisor of Fairchild Semiconductor's LSI-MOS subsystem section. This is his second article in Electronics; last year he described Fairchild's monolithic read-only memory.



Typical small computer. Its design requires only a monolithic arithmetic unit (dark tint), monolithic read-only memory (light tint), small amount of random control logic, and conventional input/output and memory hardware.

A computer organization that includes a fully parallel arithmetic unit can be used in either small or large machines. This organization, whose block diagram is shown above, also includes a fixed-program subroutine and conditional input control capability. It has a minimum number of random gates.

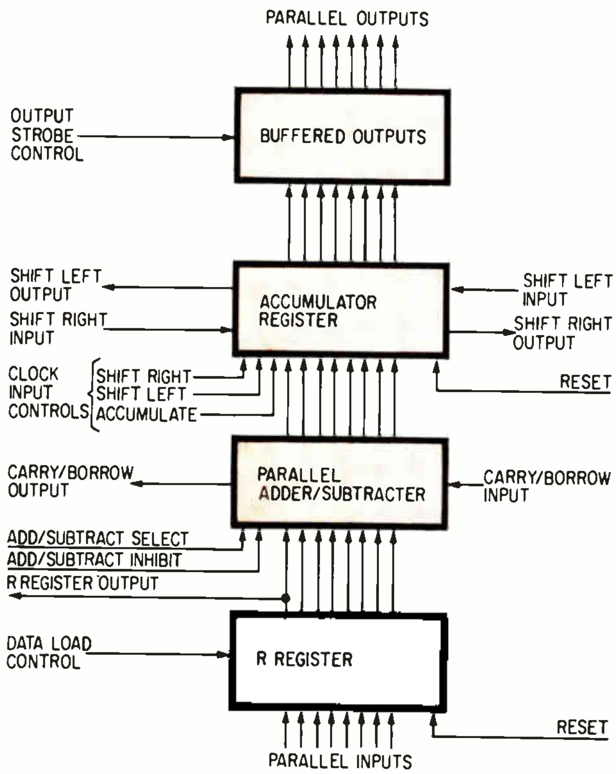
Most control functions are executed by large MOS read-only storage, using currently available hardware.[2] The standard operations include addition, subtraction, multiplication, and division, extracting square roots, and doing simple logic manipulations. More complex operations include calculating logarithms and trigonometric functions, converting between binary and decimal, doing floating-point arithmetic, and setting up subroutines for solving complex mathematical functions.

In action, the central control unit simultaneously enters an operation code into part of the arithmetic unit's instruction address register and calls required operands from the main memory. This causes the read-only memory to begin a micro-programed sequence—a sequence of microinstructions that control the flow of data into and out of the arithmetic unit, and manipulate it during multiple-step operations such as multiplication and division.

During each cycle, the read-only memory feeds the next microinstruction address back to part of its own address register and supplies a control word to the decoder. The decoder converts these microinstructions into signals that control the sequence.

During these sequences, events may occur that require modifications. For this, conditional branch or jump operations may be implemented by including conditional inputs as part of the address register input. Thus, when a data-dependent decision is made—divide stop, for example—the microprogram address is modified by the conditional inputs and the program jumps automatically to another subroutine that takes appropriate action. After that operation is completed, control returns to the central unit and the result is sent to the memory.

While the arithmetic unit is functioning, the central control section is free to operate on the main memory, input-output terminals, or other arithmetic units, thereby effectively increasing machine speed.

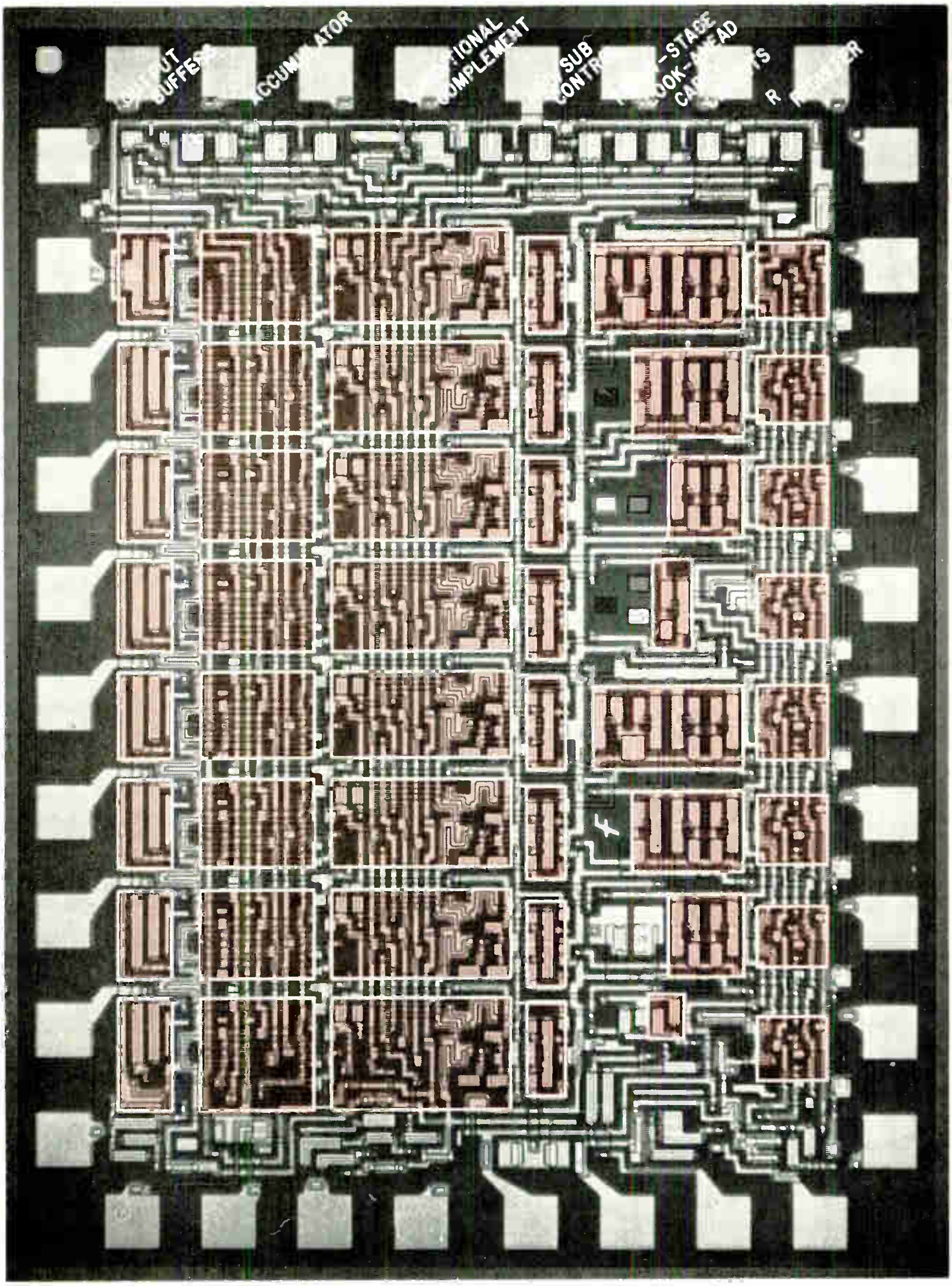


Four-in-one. Monolithic arithmetic unit comprises input register, adder-subtracter, accumulator, and output buffer, all for eight bits in parallel, plus connections to outside world or similar adjacent units.

Four sliced sections

An eight-bit integrated MOS arithmetic unit, measuring 86 by 116 mils and mounted in a 36-pin dual-in-line package, will be introduced soon by Fairchild Semiconductor under the designation 3800. The unit combines four major sections on one chip: an input register, an adder-subtracter, an ac-

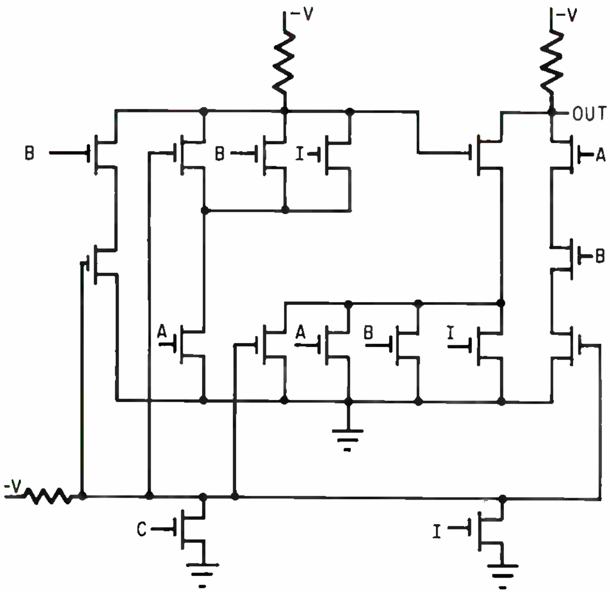
Largest production IC



Tightly packed. This silicon chip is a metal oxide semiconductor arithmetic unit with 200 gates, capable of processing eight bits at once, in an area only 86 by 116 mils. The carry-borrow look-ahead logic consists of two groups of ten circuits, each of which generates a cumulative carry or borrow for four bits at a time. All other circuitry on the chip forms eight identical parallel channels.

Add-subtract truth table

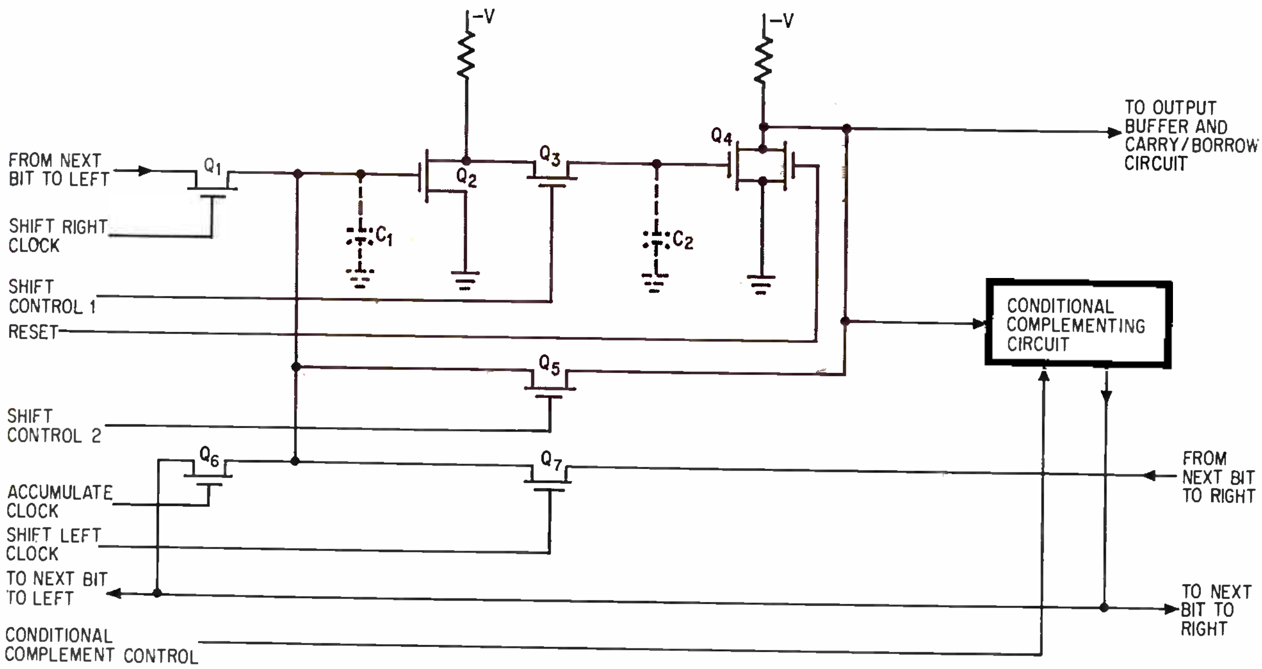
Accumulator	Addend-subtrahend	Input carry or borrow	Output carry	Sum	Output borrow	Difference
0	0	0	0	0	0	0
0	0	1	0	1	1	1
0	1	0	0	1	1	1
0	1	1	1	0	1	0
1	0	0	0	1	0	1
1	0	1	1	0	0	0
1	1	0	1	0	0	0
1	1	1	1	1	1	1



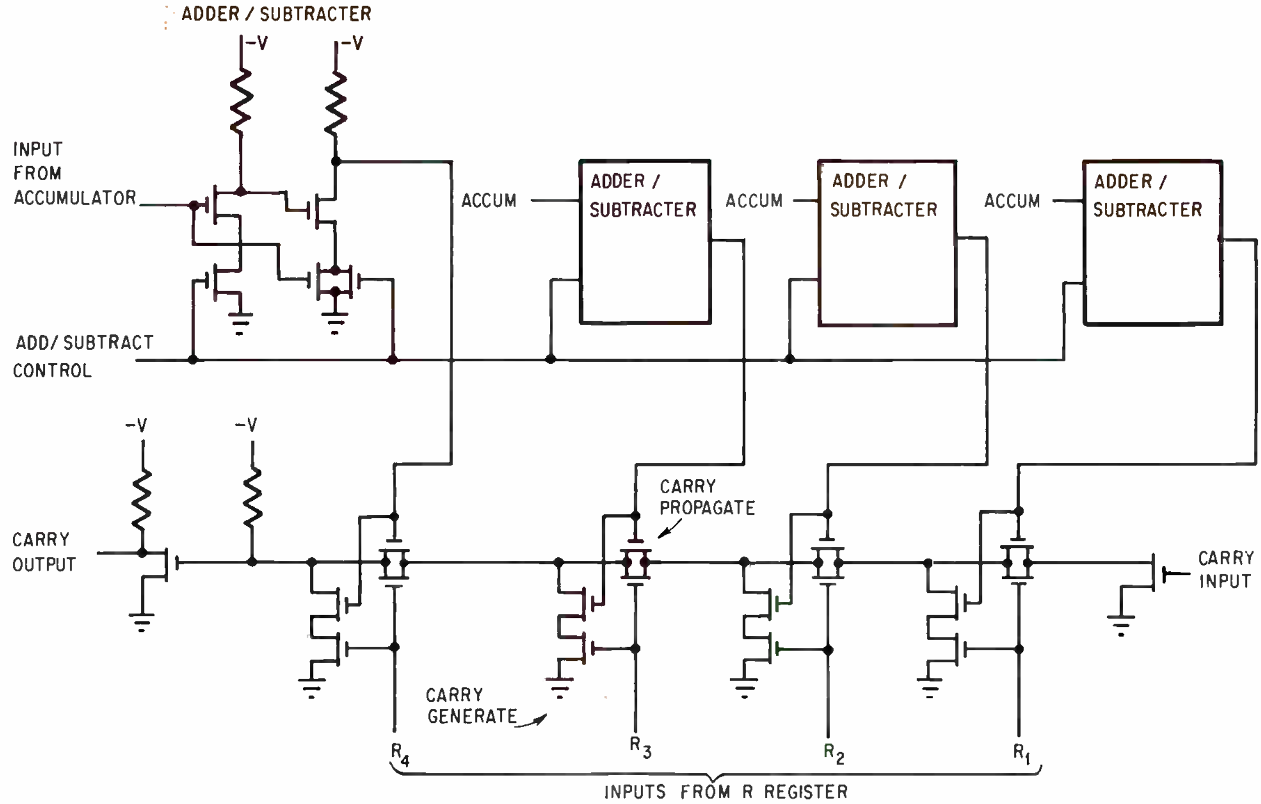
Iffy inverter. Conditional complementing circuit inverts the accumulator output if the input register and the carry bits are different and if the inhibit signal is off.

cumulator register, and an output buffer. All are made from p-channel enhancement-type MOS transistors. The four sections are shown on page 120.

- The R register temporarily stores operands for arithmetical operations. All eight positions may be loaded, bypassed, or reset at once. The leftmost bit is available for sign control.
- The parallel adder performs both addition and subtraction on either positive or negative numbers directly, without complementation. The eight-bit circuit can both accept and generate carries and borrows where more than eight bits must be processed. Look-ahead carry circuits reduce the worst-case time to carry through eight stages to approximately 2 μsec.
- The accumulator register shifts either right or left and accumulates partial sums. This shift-and-add capability much improves multiplication and division times.
- All outputs are push-pull MOS buffers. Each stage has one MOS device connected to the power supply line to pull the output up, and another connected to ground to pull it down.



Shift mechanics. Accmulator bit position contains cross-connected MOS NOR gates, shifting controls, and conditional complementing circuit that controls output for adding or subtracting. Stray capacitance in cross-connected gates retains stored data temporarily during shifting.



Add or subtract. Carry look-ahead circuit generates a carry at any bit position if both the accumulator and the input register contain a 1, and propagates an input carry if either is 1. Borrows are similarly treated during subtraction, when the accumulator bit is inverted.

Difference in subtracting

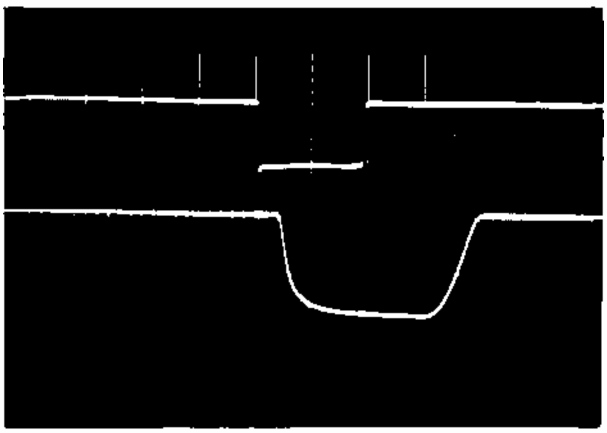
An eight-bit design was chosen for the arithmetic unit because it was easily manufactured and because it was a submultiple of frequently used word lengths such as 8, 16 or 24 bits. However, this standard block must also be usable in machines with lengths such as 12, 15, 18 or 20 bits. So the conventional subtraction algorithm was not suitable.

Ordinarily, arithmetic units subtract by a process called complement-add. First, the subtrahend—the number to be subtracted—is complemented, which involves inverting all 0 bits by 1 bits and vice versa, and adding the number 1 to the inverted number. The complemented subtrahend is added to the minuend—the number from which the subtrahend was to be subtracted. That gives the difference, except when another 1 may have to be added to the result, depending on whether an overflow carry from the left-hand end of the complement-add has occurred. This extra operation—called end-around-carry—is impossible in a standard accumulator used in systems of different word lengths, because either end of the accumulator may or may not be at the corresponding end of the system.

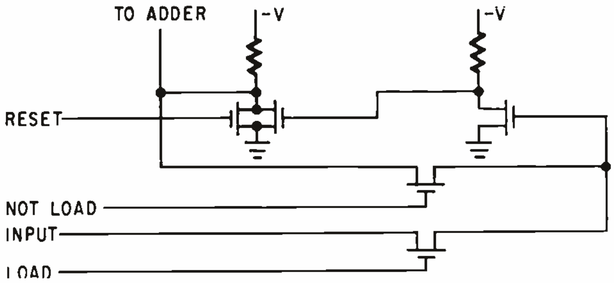
So direct subtraction circuits were included in the 3800, eliminating complementing and end-around-carry, and making the circuit a truly variable-word-length device. This additional circuitry, combined with look-ahead carry circuits to increase operating speed, made the logic techniques used with conventional bipolar circuits inapplicable to the MOS circuit. An unusual method of adding, subtracting and shifting was therefore chosen to fit the MOS technology, making possible a substantial reduction in the number of needed devices.

Logically similar

The same logic can generate either the sum or difference in the adder-subtracter without modifica-



Brief delay. Propagation delay for worst-case conditions is only about two microseconds, when both the accumulator and the input register contain zeros and a borrow signal is applied to the right-hand bit. The upper trace is the borrow input, and the lower trace is the output borrow at the left-hand bit. Scales are 10 v/div. vertically and 1 μsec/div. horizontally.



Pieces of eight. Input register position contains a pair of cross-connected transistors, loading controls, and provision for reset. The arithmetic unit contains eight of these circuits.

tion. An output to the next bit to the left—carry or borrow—can be generated by the same logic modified only by inverting the accumulator input during a subtraction.

The accumulator always contains one of the two numbers involved. An addend or subtrahend in the R register is combined with the contents of the accumulator. The sum or difference then modifies the previous contents bit by bit.

When adding, the conditional complementing circuit complements a particular bit of the accumulator if either the corresponding bit of the addend or the carry from the next bit to the right is 1, but not if both are 1. The same is true in subtraction, except that the carry signal is called a borrow. This is shown in the truth table, top of page 122, where the bits in the columns headed Sum and Difference are the same.

The table also shows that a carry to the next stage is generated if both the accumulator and the addend have a 1 in a particular bit position; and an input carry is not propagated beyond that position if both bits are 0. Otherwise, the output carry equals the input carry. On the other hand, a borrow is generated at any position if the accumulator bit is 0 and the subtrahend is 1, and an incoming borrow is killed if the accumulator is 1 and the subtrahend is 0. Otherwise, the output borrow equals the incoming borrow.

Circuit design

The accumulator register is a standard one whose contents can be shifted left or right. In either shifting mode, a particular bit can be complemented while being shifted. Any bit can also be complemented and restored to its original position, without disturbing the bit on either side.

In its static or d-c state, each bit of the accumulator—diagram, bottom of page 122—is represented by transistors Q_2 and Q_4, connected as a pair of NOR gates. The logic transistors are coupled through shift-control transistors, both of which are turned on by negative levels on their gates, forming a flip-flop under d-c conditions. The three clock-input transistors are turned off in the stable condition.

When a shift left is to occur, first the shift-control transistors are turned off and the state of the flip-flop retained temporarily by the charge on the stray capacitance of one side of the flip-flop. Secondly, a shift-left signal connects the next stage on the right to the stray capacitance on the other side of the flip-flop. Next, the shift-left signal is turned off and the shift-control transistors are turned on again, transferring the stored charges into the flip-flop. The previous output of the next stage to the right now appears on the output stage, having been inverted twice. Shift-right operations are similar.

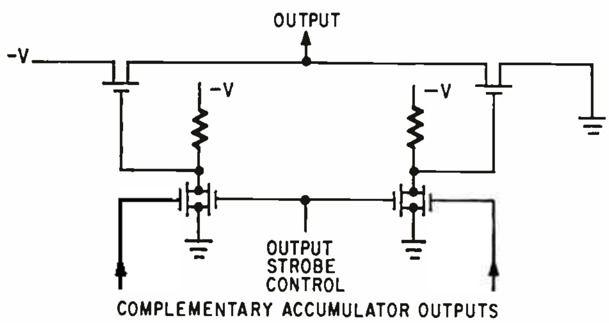
Connections to the output buffer are from the drain and gate of one side of the flip-flop. This maintains the old output value at the buffer while a new bit is shifting in. The alternative—connections from the drains of both sides—would render the output uncertain during the shift.

The unusual feature of this register is its conditional complementing circuit for each bit output. It is a function of the second operand and the input carry to each stage. The complete conditional toggle circuit, including an inhibit "I" for conventional shifting operations, is shown on page 122, center.

A four-bit binary look-ahead carry circuit, as on page 123, top, minimizes the time for addition carries to propagate through a parallel arithmetic unit, but adds little circuitry. Each of the four stages generates a carry if both the corresponding accumulator and R register bits are 1, and propagates an input carry if either bit is 1. The total carry propagation delay through the carry circuits of an eight-bit accumulator slice is about two microseconds, as in the oscilloscope photos on page 123.

A complementing circuit functionally identical to the one described, but controlled by the bit line and the add-subtract control line, appears in one of the four positions of the carry look-ahead diagrams.

The remaining circuits are straightforward. A single-input flip-flop is used in the R register as shown above; the output buffer has a simple push-pull drive, as shown at left. The strobe control for the output admits data to a common parallel bus system without adding gating or propagation delays.



Plus and minus. Output buffer connects both positive- and negative-going signals to proper references, and has a strobe input to disconnect both levels.

References

1. John L. Nichols, "Construction of complex sequential circuits with read-only memories," Electronics, June 12, 1967, p. 111.

2. Lee L. Boysel, "Memory on a chip: a step toward large-scale integration," Electronics, Feb. 6, 1967, p. 92.