

# How bit-slice families compare: Part 2, sizing up the microcontrollers

by W. Thomas Adams and Scott M. Smith,\*  
*Applied Research Laboratories, University of Texas, Austin, Texas*

□ When they need speed in a microprocessor design, most engineers think first of a bit-slice approach. But comparative data on the various bit-slice chip families has been hard to come by.

This two-part article should help designers pick their way among the maze of options available. Part 1, which appeared in the last issue of *Electronics*, focused on the processor elements of six bit-slice families. Part 2 will cover the equally necessary microcontrollers, as well as assorted special-purpose chips and programming aids.

The microcontroller sequences a bit-slice system's instructions. Models available today include Texas Instruments Inc.'s SN74S482, Advanced Micro Devices Inc.'s Am2909, Am2910, and Am2911, Motorola Inc.'s very fast MC10801, Monolithic Memories Inc.'s 67110, Intel Corp.'s 3001, Signetics Corp.'s 8X20, and Fairchild Camera and Instrument Corp.'s 9408.

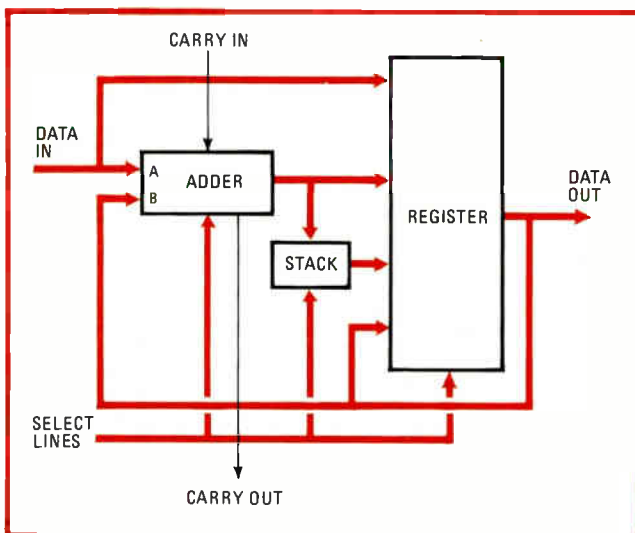
## Alternative philosophies

Microcontroller designs may follow one or other of two distinct philosophies. The first is one in which several parts are cascaded to address a reasonably sized control memory, the part being referred to as a microprogram control element. The second is actually not a bit-slice approach, but one in which a single package, called the microprogram control unit, houses all the circuitry required to address anywhere from 512 to 4,096 control memory words, depending on the device. In the latter approach, the address space may be increased by paging techniques, but not by cascading devices. (A paged memory is one that is divided into blocks, which in this case are equal to the length of microcontroller's address space. The current working block or page is selected by a register set under program control, but not by any of the internal next-address modes of the microcontroller.) A table summarizing the characteristics of the microcontrollers will appear in the next issue.

A simple but extremely flexible microprogram control element is Texas Instruments' SN74S482 (Fig. 1). Each of the 74S482's major components has separate control lines, which the designer may encode to select the next microinstruction. The provision of a full adder rather than a simple incremter allows relative addressing, a feature unavailable in any other microcontroller. Unlike the other microcontrollers, however, the 74S482 lacks three-state outputs on its memory address port. Housed in the 300-mil, 20-pin dual in-line package, the 74S482 occupies relatively little circuit board area.

As indicated in Fig. 2, the Advanced Micro Devices' Am2909 and Am2911 are identical except for the separate register input and the OR inputs that the 2909 offers. Eliminating those features on the 2911 allows it to be packaged in a 300-mil, 20-pin DIP instead of the 2909's 600-mil, 28-pin DIP. Both microprogram control elements, like TI's 74S482, have no inputs for instructions, but rather have a series of control lines for the various internal functions that the designer must encode for the next instruction function.

AMD offers a microprogramming handbook that provides considerable insight into the use of the 2909 and 2911 and some other related devices. However, to be



**1. Simple microcontroller.** Although Texas Instruments' SN74S482 microcontroller element appears simple, it is very flexible. While other sequencers have incrementers to select the next microinstruction, the 74S482 has a full adder, which permits relative addressing.

\*Both authors are now at IBM Corp., Office Products Division, Austin, Texas

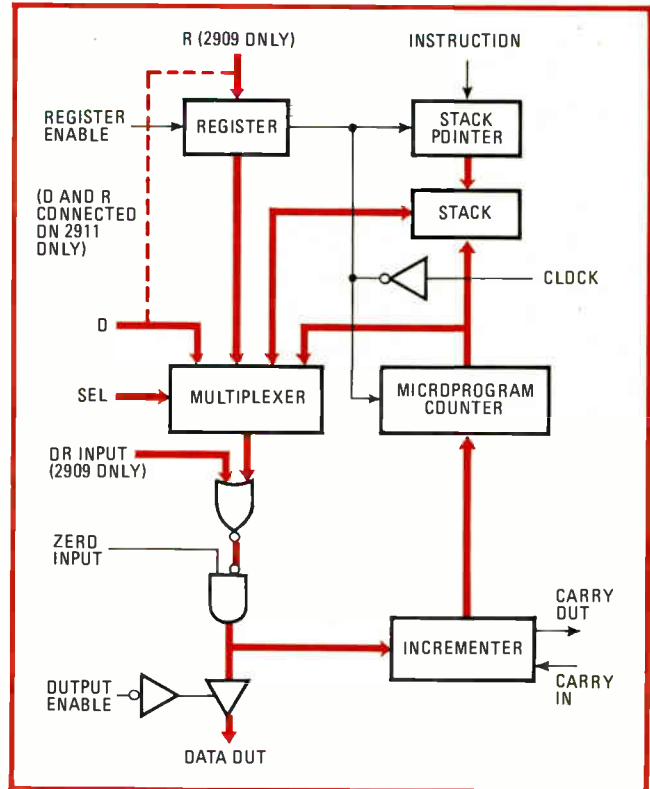
described later is AMD's new microprogram control unit, the Am2910, which makes the 2909 and 2911 practically obsolete for all applications except those requiring control memories in excess of 4,096 words.

**Fastest**

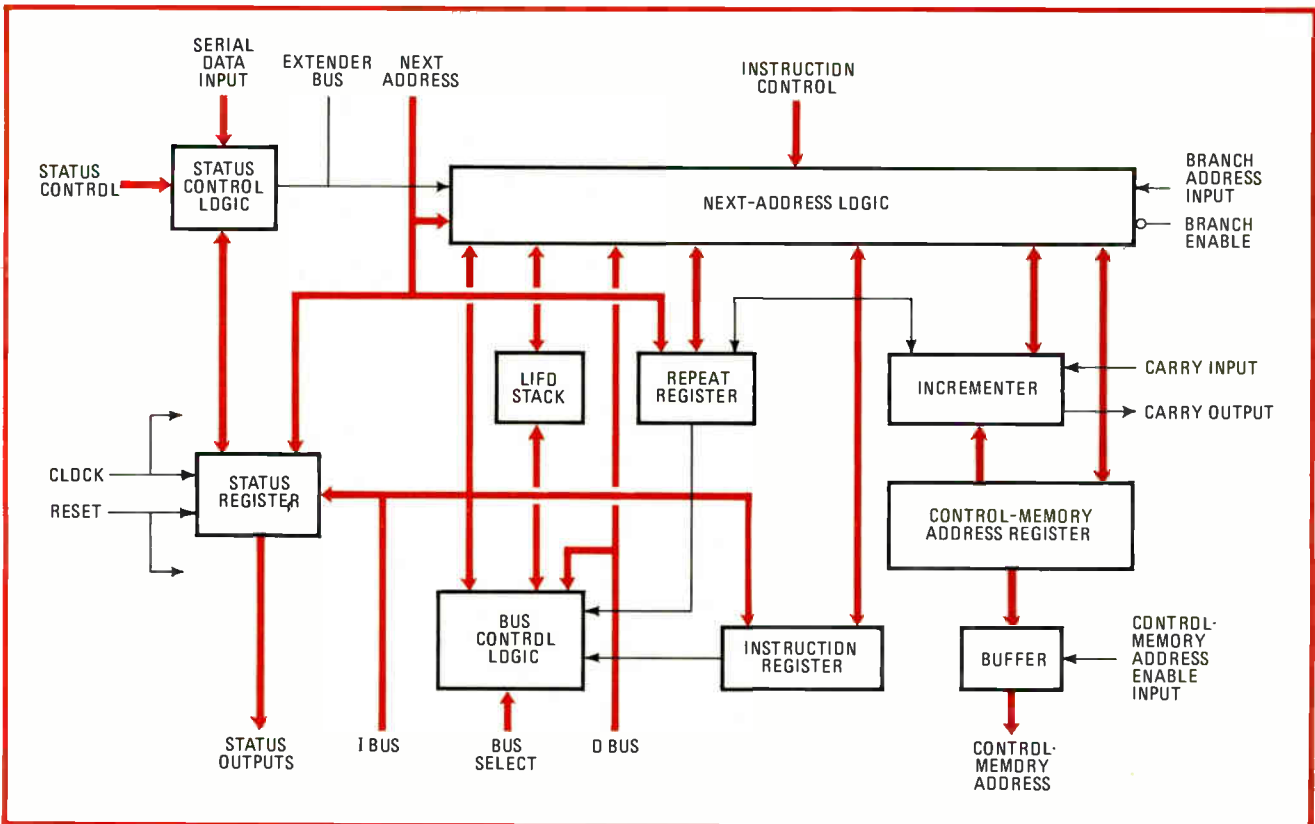
Motorola's MC10801 (Fig. 3) is unique among the microcontrollers for a number of reasons. It is the only emitter-coupled-logic sequencer available, so that in terms of clock frequency alone it is the fastest device available. But the 10801 also has architectural features that further increase its throughput potential. A repeat register allows single instructions or subroutines to be executed a specified number of times without the usual sequence of decrement loop count, test for zero, and branch or increment on test result. Microinstructions are saved, thus increasing throughput, and no register in the processor array is needed for the loop count. The 10801 provides internal storage for status information, which may be used to determine the outcome of conditional instructions. The status bits are also available through an output port for use elsewhere in a system.

An internal instruction register allows the operational code of the next macroinstruction to be loaded during execution of the current one. This register is loaded from one of the two system I/O buses connected to the 10801; these I/O buses may also be used to externally expand the 10801's last-in, first-out subroutine-linkage stack. As can be inferred, the 10801 is a complex, powerful device.

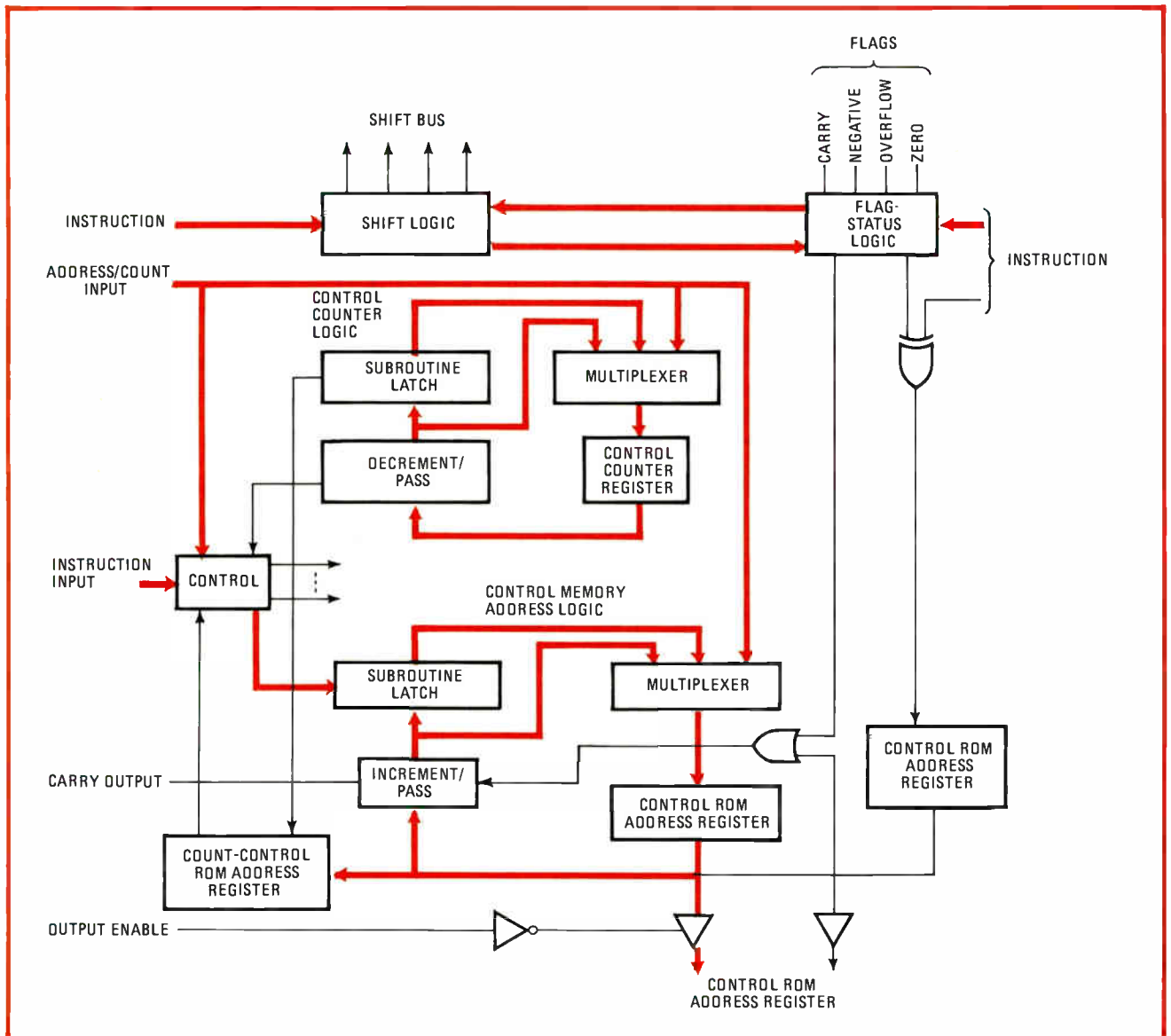
The many options available to the 10801's user are clearly explained in Motorola's excellent data sheet on



**2. Two of a kind.** The only difference between Advanced Micro Devices' Am2909 and Am2911 microcontroller elements is the elimination of separate register and OR inputs on the latter. Both have control-line inputs in lieu of an instruction input.



**3. Fastest sequencer.** Aside from the fact that it is the only all-emitter-coupled-logic microcontroller, Motorola's MC10801 has features that increase its speed potential. One example is the repeat register, which lets subroutines be reiterated without the normal looping sequence.



**4. Not a slice.** Monolithic Memories' 67110 is not cascadable—it is a microcontroller unit with an addressing capability of 512 words. While it is based on a program counter, branching is to even-odd pairs in memory and requires care when microcode is being modified.

the device. Again, as with the 10800, direct mixing of the 10801 and transistor-transistor-logic bit-slice parts is not recommended. The 10801, being a good high-performance part, is recommended for use in high-speed minicomputer and signal-processor applications, though a few controller applications may also require it.

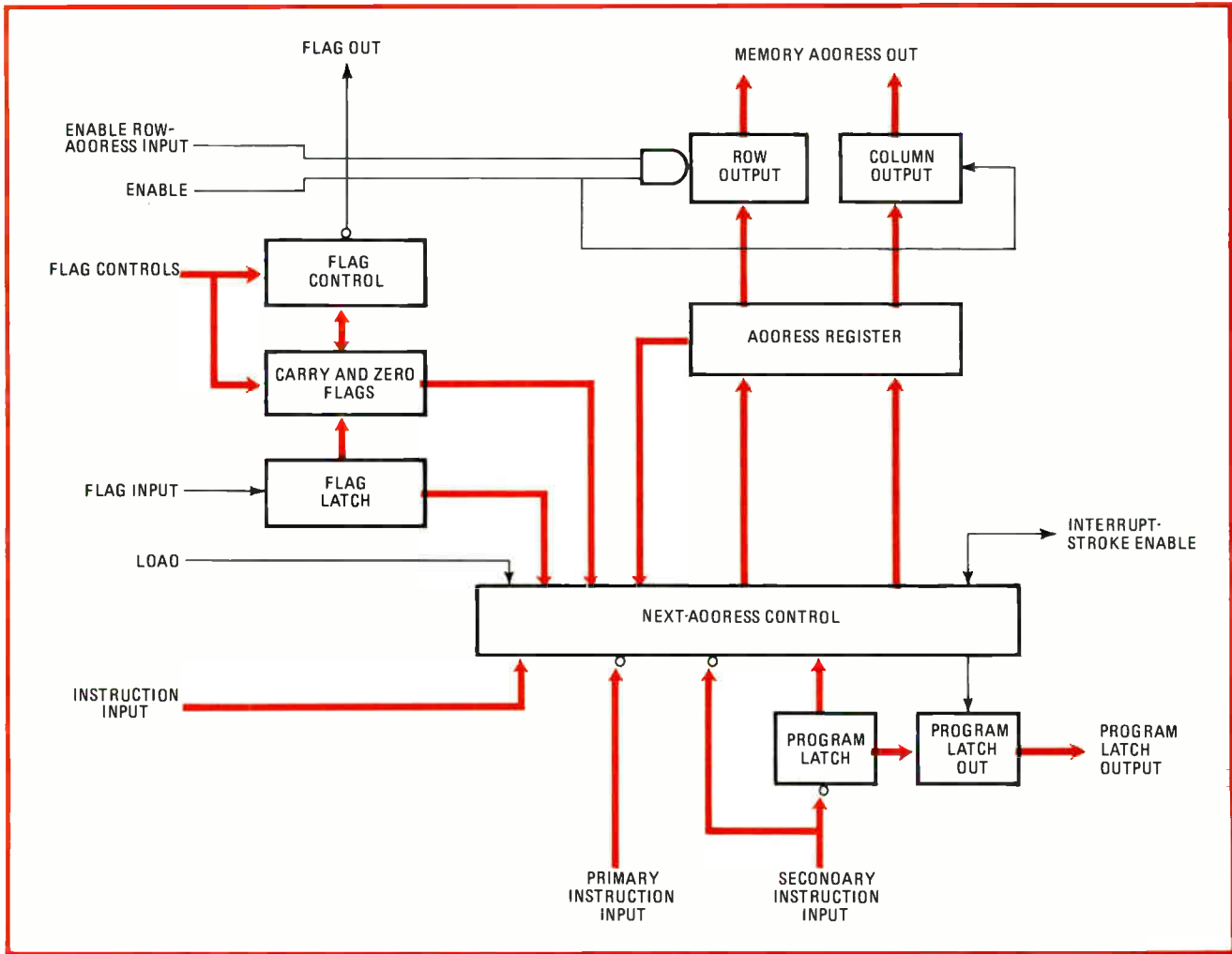
Shown in Fig. 4 is Monolithic Memories' 67110, a microprogram control unit with an addressing capability of 512 words. Its basic addressing scheme is based on a program counter; however, the least significant bit is handled such that a conditional branch to either location of the next even-odd address pair may be executed without the use of the microinstruction next-address field, thus often freeing that field for other uses. When this addressing scheme is employed, note that care must be taken in writing and modifying microcode to ensure that all even-odd branch pairs are in fact stored in even-odd address pairs.

The 67110 offers an internal looping capability not

found in most microcontrollers, which allows it to execute a program loop a specified number of times without the usual sequence of decrementing the counter, checking for zero, and branching or incrementing on test result. The single-level subroutine capability in the 67110 is rather limited, however. Internal storage for various processor-element array flags is provided, as is the ability to condition the next-address selection on the stored or current value of any of those flags. Although the number of control lines seems high, several are required to select the flag bit used in carrying out conditional operations.

Probably more than any other factor, the choice of whether or not to use the 67110 depends upon whether or not the designer likes the pairwise branching scheme.

Another microprogram control unit with a 512-word address capability is the Intel 3001 (Fig. 5). Its addressing scheme, however, is unique. The 512-word address space is defined as a 32-row-by-16-column matrix; a set



**5. Unique addressing.** Intel's 3001 also has a 512-word addressing capability. Only seven instruction control lines are needed; a set of branch instructions allows jumps from a given address to a fixed subset of the address space that is a function of that address.

of branch instructions allows jumps from any address to a fixed subset of the address space that is a function of the given address.

With this technique, only seven instruction-control lines are required. No branch address input is needed other than that needed to obtain macroinstruction starting addresses. While this approach can reduce the width of the control memory, it can also complicate the debugging and modification of microcode.

Undoubtedly, the greatest weakness of the 3001 is its lack of an internal subroutine capability, though the device can be wired externally to provide for single-level subroutines. Like the 67110, the 3001 provides storage for several flags associated with the processor-element array that can be used in conditional branching.

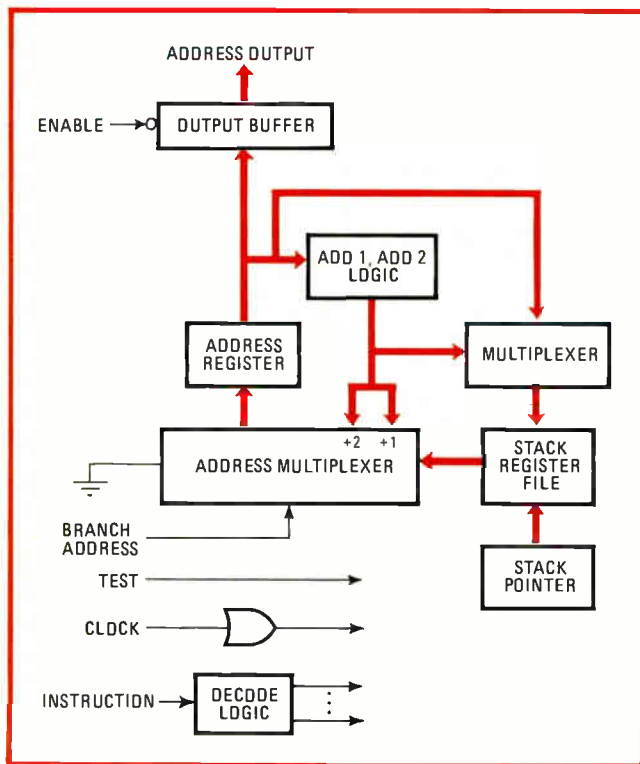
### Simple but capable

The Signetics 8X02 (Fig. 6) is a microprogram control unit with a 1,024-word address space. Though it appears simple, it is a capable device. A program counter is used for the basic addressing mode. Like most microcontrollers that use a program counter, the 8X02 can branch to an address specified on its branch-address input port. However, it provides several instructions that minimize

use of the branch-address input, thus freeing that microinstruction field for other purposes much of the time.

For instance, a skip command allows incrementing the program counter by two rather than one on a true-test input. Also, two other instructions allow program loops to be executed, again without use of the branch-address input—the address of the loop entry point is merely placed on the last-in, first-out stack normally provided for subroutine linkage. (Note that this looping capability is not the same as that referred to in the table, since the number of times the loop is executed must be counted externally in the 8X02.) Housed in a 28-pin dual in-line package, the 8X02 offers more performance per square inch of circuit board area than most other microcontroller devices. It is an excellent choice for all but the highest-performance applications.

Fairchild's 9408 is a 10-bit-wide microprogram control unit that is based on a program counter. Its block diagram (Fig. 7) reveals several useful features. There are four flip-flops for storing input test flags to be used in conditional branches. (The only criticism of this arrangement is that, though there is a separate strobe for the register group, there is no provision for loading one flag without disturbing the other three.) In addition to



**6. Capable.** Signetics' 8X02 microcontroller unit has a 1,024-word address space. Like others that use a program counter, the 8X02 has a branch-address input; however, several commands, like skip and looping commands, save on branch microinstructions.

the basic branch address input, the 9408 control unit has an alternative source for the three least significant bits of the branch address. This alternative source is used in an eight-way branch instruction that can, for example, decode macroinstruction op codes in just a single microinstruction.

The 9408's control-memory address output may be driven either by the program counter for nonpipelined operation or by the program counter's input multiplexer for pipelined operation. Implemented with Fairchild's version of I<sup>2</sup>L, Isoplanar integrated injection logic, the 9408 is a slower device than the TTL microcontrollers. Also, although its power dissipation is relatively low compared to the TTL devices, it lacks the user-programmable speed/power dissipation feature of TI's I<sup>2</sup>L processor element, the SBP0400.

### Three in one

As can be seen by comparing it with the Am2911 (Fig. 2), AMD's Am2910 (Fig. 8) essentially consists of three cascaded 2911s and an instruction-decoding programmed logic array on one chip. The auxiliary register in the 2910 is also a counter with a zero detector, and the last-in, first-out stack is five words deep. In addition to decoding instructions, the PLA provides three output signals that can be used to enable any of three sources to the 2910's D inputs.

The register/counter in the 2910 may be used to store a branch address that is to be used later in a double-address jump or a subroutine call; as a counter, it is used to count down the number of times a program loop has

been executed and to cause a loop exit when the count equals zero. The beginning address of a loop may be pushed onto the LIFO stack as the loop is entered, thus allowing the top-of-loop address to come from the stack rather than from the direct data inputs.

The 2910 has a unique three-way branch instruction that is useful at the end of loops. If the input test condition is true, incrementing the program counter causes an exit from the loop. But if the test condition is false, the loop counter is decremented and the program branches to the top of the loop until the counter is zero, and then branches once more to the address specified on the direct input lines.

The 4,096-word address space in the 2910 is more than enough for the vast majority of applications. The part offers most of the next-address selection modes provided by the other devices and is housed in a single 40-pin DIP.

Two other sequencer-type devices—Fairchild's 9406 program stack and AMD's Am2930 program control unit—might be used as microprogram controllers, but are marketed primarily as macroinstruction sequencers. Both parts are bit-slice designs, and perhaps their single most significant difference from the microcontrollers is the depth of their LIFO stacks—16 or 17 levels compared to the 4 or 5 levels of typical microcontrollers. The 2930 is the more capable of the two devices, since it has in addition to an incrementer a full adder that permits a number of relative-addressing options.

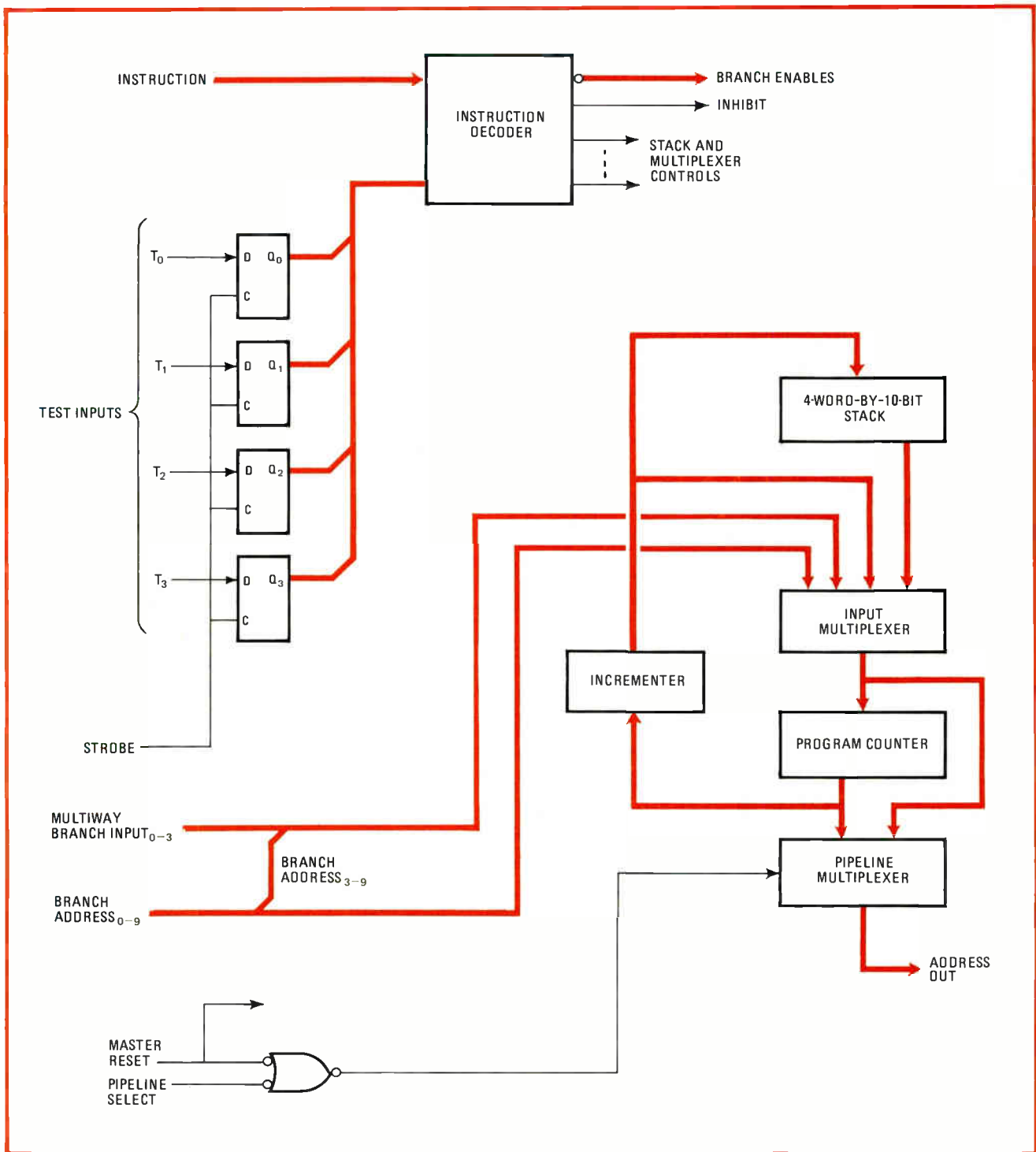
### Special-purpose devices

Several manufacturers provide as part of their bit-slice families special-purpose devices that perform some unique function. Space does not allow us to cover all of the special-function devices available, but here are a couple of the more significant examples.

The addressing and controlling of system memory is one function for which two special purpose-devices have been built. They are Fairchild's 9407 data-access register and Motorola's 10803 memory interface unit, both of which may be used to remove the burden of interfacing with system memory from the processor-element array, thus increasing system throughput. Intel and AMD each offer circuits for vectored priority-interrupt encoding, which aids the microcontroller in interrupt handling. Motorola's MC10802 timing function device supplies 10800-based designs with a number of useful timing features. Among other functions, the 10802 can be programmed to generate from two to four clock phases, any of which may be doubled in length to compensate for a slow data path without slowing down the other phases.

### Design support

One of the more important aspects of designing with bit-slice microprocessors—particularly to new users—is the design support provided by the manufacturer. Such support consists of literature (including data sheets and application notes), direct assistance from the manufacturers' applications engineers, software (primarily microassemblers), and hardware, including kits and development systems. The quality and quantity of literature varies greatly among manufacturers. One reason for



**7. IPL controller.** Fairchild's 9408 microcontroller has a 10-bit-wide address capability and features that include an 8-way branch instruction and flip-flops for storing branch-test flags. Though slower than other controllers, the integrated-injection-logic device requires less power.

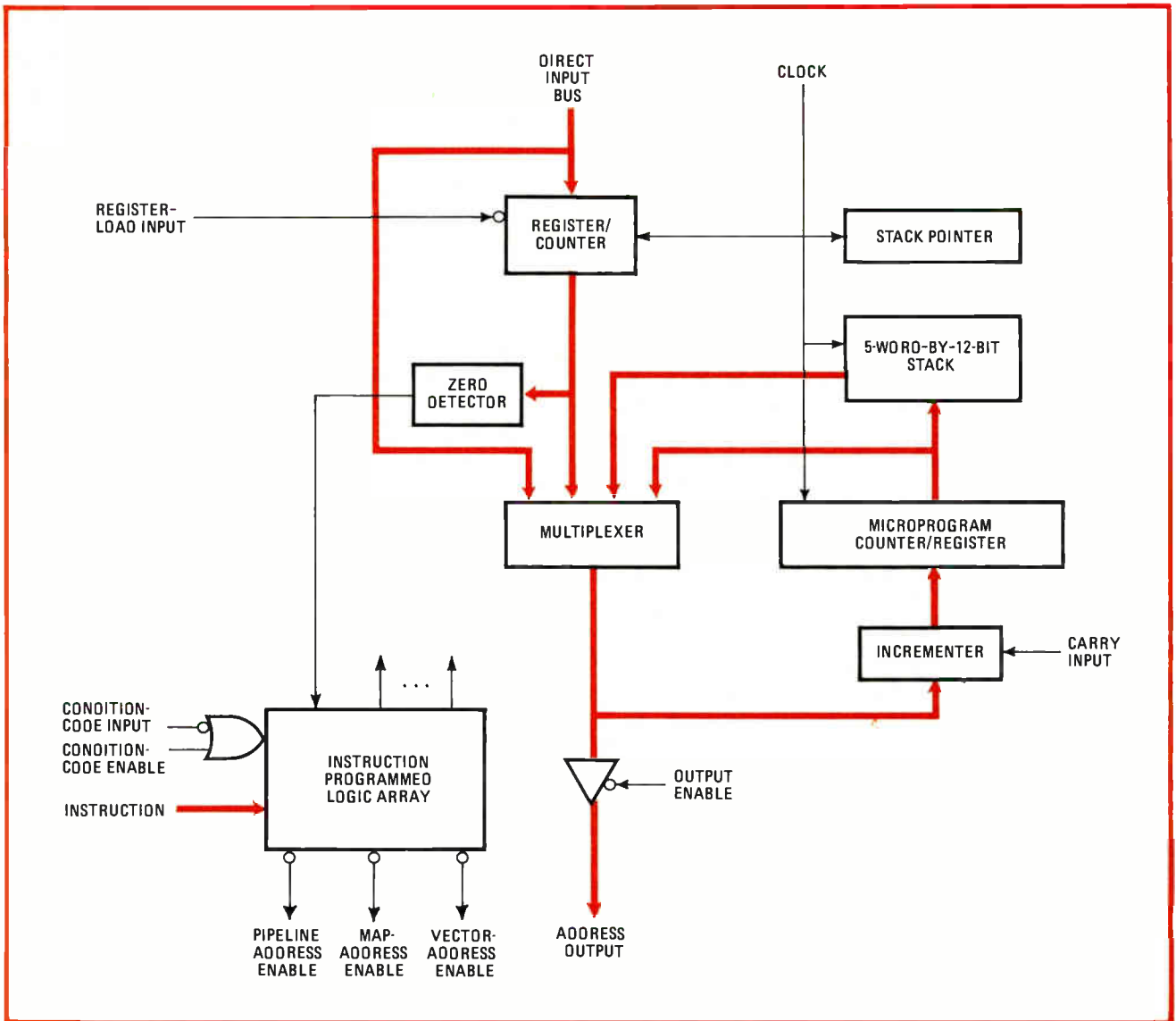
the popularity of AMD's 2900 family has been the excellent literature provided with it. The application notes from AMD have had the most detailed design information. Intel and Motorola also supply excellent applications literature.

The direct assistance provided by the various manufacturers is much harder to evaluate since the quality of advice received is at least as much, if not more, a function of the individual applications engineer than it is

a function of the manufacturer he represents.

Software support for the bit-slice microprocessors consists of operating systems for hardware-development systems and cross-assemblers. Intel, AMD, and Raytheon have offered cross-assemblers for quite a long time. Those programs let the user define an assembler to fit his own application.

Intel provides such a program in Fortran, and AMD and Raytheon, as well as Intel, are supported by



**8. 4-K address space.** Advanced Micro Devices' Am2910 essentially comprises three Am2911s and an instruction-decoding programmed logic array. Control memories larger than the 2910's 4,096-bit address space will require cascadable sequencers like the 2909 or 2911.

programs over various commercial time-sharing networks.

The assembler can be a most cost-effective tool in the development of bit-slice microprocessor designs since the programmer can write, debug, and document his microcode in a much clearer form in the mnemonics of assembly language than in 1s and 0s of machine language. As an alternative, he may define his own assembler on a data-processing system.

### Hardware support

Fairchild and AMD offer hardware support in the form of kits with printed-circuit boards, on which the user can build a demonstration bit-slice microprocessor system. The kits, which sell for a few hundred dollars, aid those with little microprogramming or bit-slice experience.

Intel, AMD, and Motorola, among others, offer some form of hardware development system. The systems basically allow the user to assemble microcode for a bit-slice design in a resident MOS microprocessor and to

load this code into random-access memory (called a read-only memory simulator in this application), which is connected to the system under development in place of its own microcode memory. The hardware development system allows modification of the microcode in RAM and tracing of the execution of microcode and also supplies traps for specified code sequences under microaddress or external hardware control, among other functions. AMD's System 29 also provides a framework for housing the hardware under development. It should be noted, however, that the microcode execution using these development systems is in general at a speed below normal. Moreover, these systems are not inexpensive, so the purchaser should have enough development work to justify their cost.

If a hardware development system is not available, the designer must provide his own scheme for easily altering the contents of the microcode memory during the debug phase. A general-purpose logic analyzer can, of course, be used to trace microcode execution. □