

Nova can't lose its instructions

Program can be debugged in main memory and wired into read-only form for fail-safe operation in this very small multiregister machine

By Edson D. de Castro, Henry Burkhardt, and Richard G. Sogge

Data General Corp., Hudson, Mass.

An architectural approach, the key to many large third-generation computers, has been applied for the first time to a small computer. As in the designs of large-scale processors, the principal guidelines here are customer requirements and general trends in hardware technology.

Ideally, the availability of specific components does not influence an architectural design, but it is, of course, a factor in implementing the design. In this case, an economic advantage has been realized by employing integrated circuits. However, since the approach stresses function rather than hardware, the usefulness of the basic design is not limited by the economic life of a particular set of components. The computer has an intrinsic order and utility that are impervious to the quirks of some components.

One of the foremost characteristics of many modern computer designs is a multiaccumulator organization—the system's central processor contains a number of active general-purpose registers. Arithmetic and logic operations are executed more simply by manipulating data in these registers than they could be by transferring the data to and from the computer's memory.

Up until now, such registers have been considered impractical in small general-purpose computers, but medium-scale integration has changed all that. More than half the gates and flip-flops in the new machine are MSI circuits; this serves to reduce the number of interconnections and thereby shaves packaging costs.

Besides embodying an architectural design and featuring multiple registers, the new computer can have a braided read-only memory for program storage. This unit is made of wires preformed into a braid and dropped over U-shaped ferrite cores, which are then capped. It's unique in that it is homogeneous with the alterable main memory—that is, 1,024-word modules of read-only memory are directly interchangeable with main-memory mod-

ules made of conventional toroidal ferrite cores.

The user writes a program, debugs it in the alterable main memory, and orders it wired into read-only modules, which then replace the corresponding portion of alterable memory. From that point on, it is as immune to bugs as a program can be.

Targets

The computer described here is the Data General Corp.'s Nova, one of the growing class of so-called "minicomputers" that sell for under \$10,000. Within the architectural framework, several rather unusual goals were set for the Nova design:

- A price of \$5,000 or less in quantity for a complete system containing processor, power supply, and console, and having a full input-output capability and a memory comprising 4,096 words of 16 bits each.

- An absolute minimum number of interconnections, in the interest of reliability.

- No custom-designed components; all parts used must be produced in volume by a reputable manufacturer.

Besides these, several more or less conventional design objectives were established:

- A full cycle, including a memory cycle and accumulator access or indexing, in less than 3 microseconds.

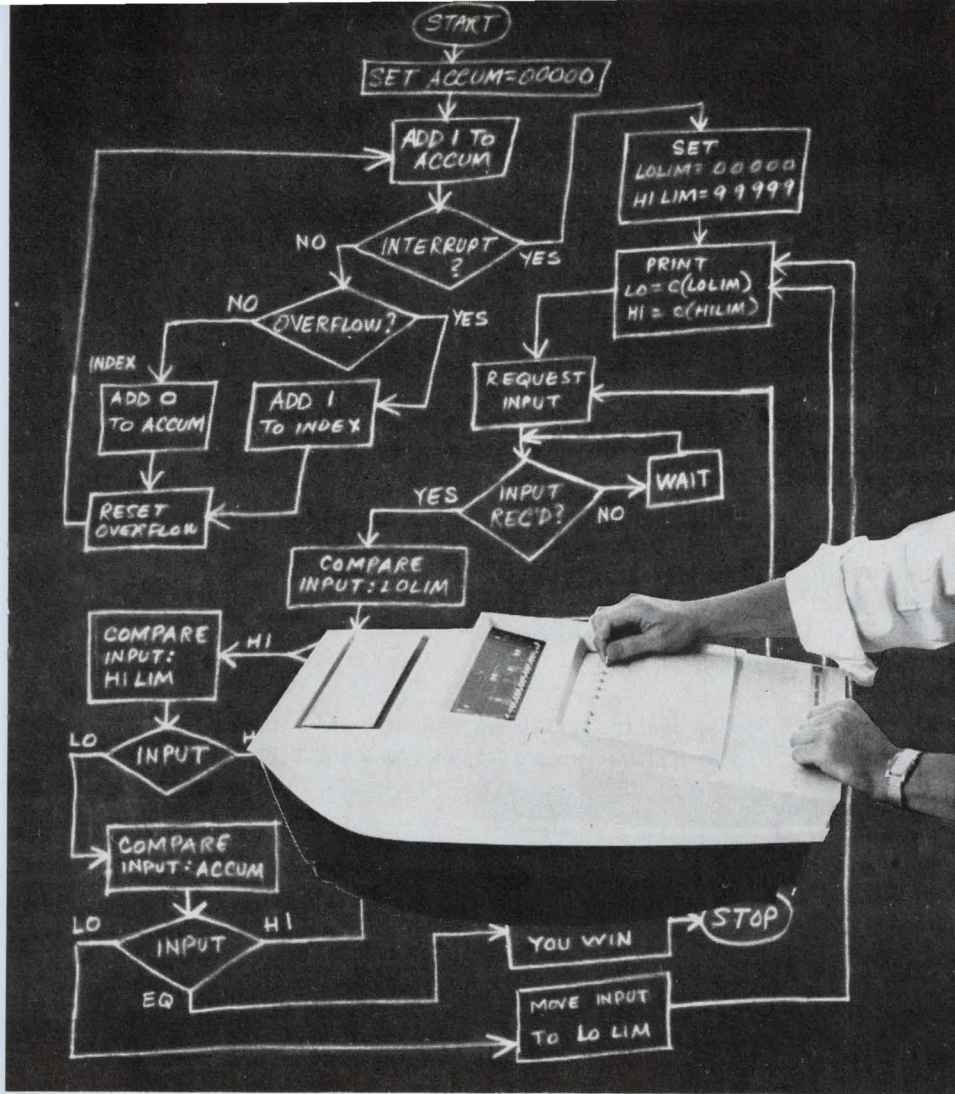
- An input-output facility capable of both single-word transfer under program control and independent direct access to the memory by external devices, as well as a priority interrupt system that identifies the interrupting device.

- Room for expansion of either memory or input-output device controllers.

- Reliable operation over the entire commercial temperature range, 0° to 55°C.

- Minimum use of adjustable components.

All central processors consist of data paths, which are regular and repetitive, and the controls



Mighty midget. Small though it is, the Nova has all the characteristics of a much larger computer, as the flow-chart in the background indicates. The chart itself represents the stored-program concept, the diamond-shaped decision blocks indicate conditional branches, and the operations at the upper left show how the program can modify itself. Strictly speaking, the read-only memory prevents direct modification, and current programming practice frowns on it, but the concept is still valid.

—the irregular and hard-to-describe parts of the machine.

The regular portion consists of registers, adders, multiplexing gates, and the like. Its components, being common to all machines, represent a mass market for integrated-circuit manufacturers. The controls, on the other hand, are usually unique to a specific machine and therefore aren't economically feasible as large integrated chunks unless vast quantities of a single type of computer are being produced.

For instance, 16 flip-flops in a single package are available for use in regular registers, but only two flip-flops can be put in the same package for random control applications. Likewise, a four-bit adder containing 42 gates is available off the shelf, but four gates is the maximum available for individual use. Internal interconnections make the difference.

The Data General designers took into consideration that although regular components are being offered at ever-increasing densities and decreasing costs as technology progresses, the trend is much slower in the case of irregular parts. They therefore structured the Nova computer to utilize as few irregular logic groupings as possible. In practical

terms, this was accomplished by keeping the instruction set straightforward—reducing the amount of irregular random control logic—and by designing the instructions' execution algorithms to operate efficiently in multiple general registers—sometimes at the expense of additional instructions in the set. By using an instruction format in which each bit has its own function, much irregular decoding logic was avoided.

For example, comparing two numbers in many small computers means complementing one of them, adding a 1 to it, adding the other number to it, and then testing the result—which is zero if the two numbers are equal. Any halfway decent computer needs all three of these instructions—complement, add (used twice here), and test. But the Nova achieves the same result with a single instruction—compare—in much less time; the one instruction covers the same four steps, but by whipping the numbers through the registers, it requires very little extra irregular logic.

For another example, one instruction in the Nova can subtract and shift the result to the left. This instruction, which is used in programmed division, necessarily supplements an ordinary subtraction, but the logic to implement it is much less complex

and irregular than a hardware divide would require.

Several such instructions make the Nova's set unusually efficient and easy to use.

Logical choice

To achieve the needed regularity in the processor's design, an IC logic family had to be selected. The designers considered several factors: breadth of line, cost, availability from several sources, noise immunity, and speed. They evaluated four lines and established the ranking diagrammed at the right.

This diagram indicates that emitter-coupled logic is fastest of the four and has the best noise immunity, and that resistor-transistor logic is the least expensive circuit form. Diode-transistor and transistor-transistor logic are quite close to one another in speed and cost—although TTL has a small but distinct advantage in breadth of line and in its availability from several sources.

Several other factors were also considered, especially as they affected cost. For example, the number of new designs appearing in RTL and DTL is declining; thus the production of these lines may drop and prices increase. And because the cost of the printed-circuit board, its connector, and the wiring is often several times the cost of the component itself, the availability of complex functions in a single component could result in real savings in packaging costs.

Speed affects cost because it permits different levels of parallelism or seriality; a given level of performance can be achieved with a smaller number of relatively faster components.

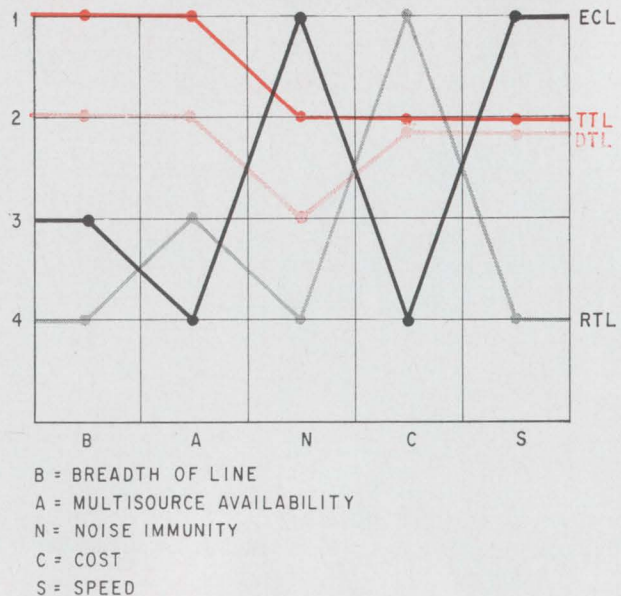
With all these factors in mind, the designers decided on TTL.

Even the most complex functions Data General chose to buy for the Nova were available in off-the-shelf circuits. These circuits have many applications, and the company believes this is a more useful definition of medium-scale integration than any arbitrary number of transistors or gates per chip.

MSI's major contribution is not low cost per function but rather a reduction of interconnections and a consequent improvement in reliability. It is true, though, that cost per function is usually somewhat lower with MSI components than with discrete gates and flip-flops.

Nibbling away

The Nova processes all data in groups of four bits, called "nibbles" because they're larger than a bit and smaller than a byte. This partly parallel organization represents a better choice than either a serial organization with a few fast components or a fully parallel organization with a larger number of slower parts. It's obviously only one of many possible choices; any submultiple of 16 bits—the word length—could have been used, each with its own advantages in cost and performance. But all in all, the four-bit path has more advantages and fewer disadvantages than any of the others.



Optimum choice. TTL circuits have the best combination of characteristics desirable for a small computer.

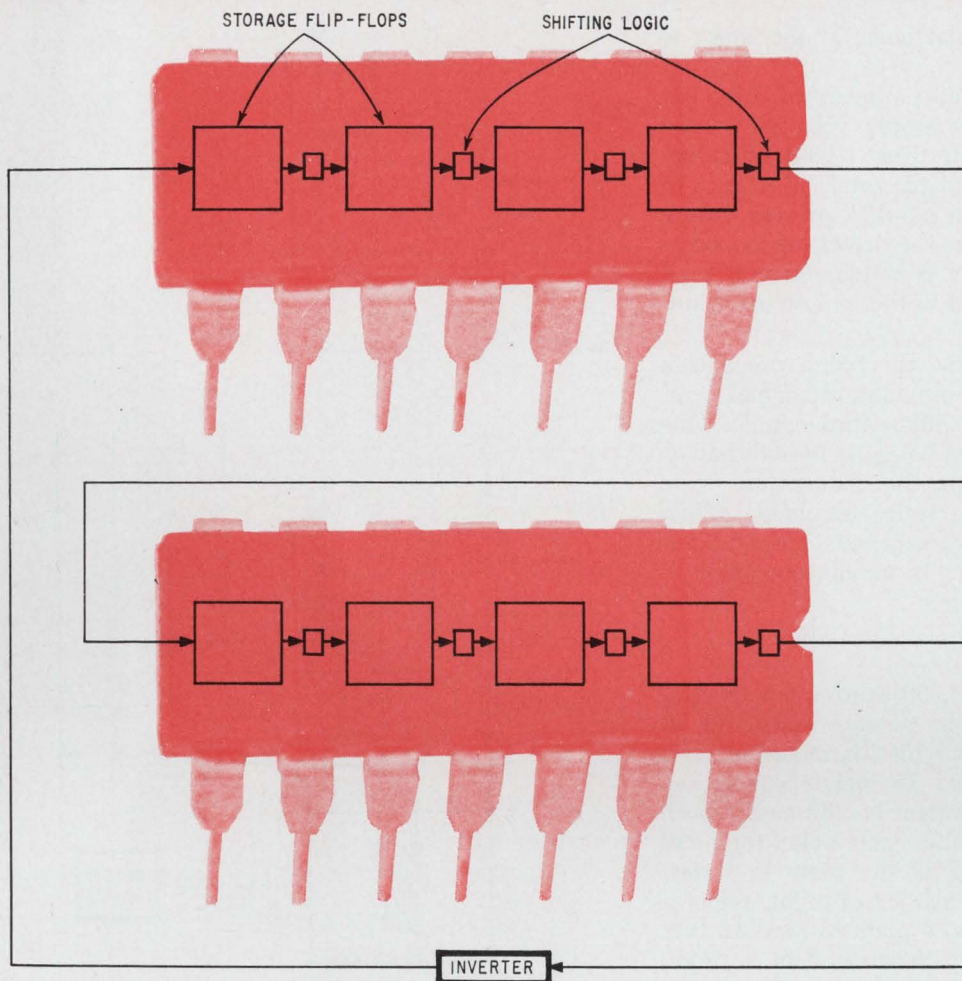
Among the advantages is the availability of standard off-the-shelf MSI circuits for several applications in the Nova. For instance, the accumulators consist of four 16-bit monolithic registers selected and sequenced by two MSI decoders. Each register unit contains 16 flip-flops addressed as a four-by-four matrix. One line from each of two groups of four selects the bit in a particular position in all four registers—four bits, or a nibble, in parallel.

As each register contains four bits from each of the accumulators, one of four control lines selects a particular accumulator. The other four lines are then brought up in sequence to access an entire accumulator four bits at a time. The output of the accumulator is gated into a four-bit adder—another MSI circuit in a single package. A full 16-bit addition is completed in four clock times as data passes through the adder a nibble at a time.

All the other machine registers are MSI four-bit shift registers. These include the memory address, memory buffer, and instruction registers, and the program counter. Data enters these registers from the adder output or passes from them to the adder input. An individual IC shift register will contain bits 1, 5, 9, and 13, for instance, instead of four adjacent bits of a word, so that a single shift operation is equivalent to a shift of four bit positions. Thus, four data entries and four shifts are involved in the transfer of a complete word.

Another interesting use of MSI is in the timing and control section. The entire machine's timing is locked to a single crystal controlled oscillator running at 10 megahertz.

The waveforms driving the memory system come from two MSI shift registers connected as an eight-bit, switched-tail ring counter, that is, as a single



Timing. These two shift registers feeding one another, plus the inverter, generate all the waveforms needed for the memory system in the Nova. The configuration is called a switched-tail ring counter.

shift register with an output that's inverted and fed back to the inputs, as shown above. Every time the counter shifts, one and only one bit changes state, so that the register as a whole has 16 different states.

Connecting various combinations of these eight bits to logic gates generates a wide variety of waveforms; those for the memory system are obtained from two-input AND gates.

Cores and stacks

As with the rest of the Nova design, the approach taken to the core memory was aimed at holding down both cost and physical size. A decision on the core size was easy to make. The required speed could be realized with a 30-mil core, a type now in high-volume production.

The question of stack organization was much thornier. So-called 2½-D arrays have become quite popular because they require only three wires through each core and are therefore inexpensive. But their associated electronic circuits are expensive, especially in small systems. The more common three-dimensional organization employs less electronics, but four wires must be strung through each core, increasing the array's cost.

The three-wire, 3-D scheme offers the best of

both worlds in a small system. In this organization, the sense and inhibit functions share the same wire. In a conventional 3-D memory, the sense line is used for readout only and thus is active during only the read portion of the cycle. Conversely, the inhibit line is functional only during the write half of the cycle. The inhibit line requires a balanced drive and it's important that the sense line not be too noisy, but there's no logical reason why a single wire cannot be time-shared to perform both of these functions.

This sharing requires an unusual sense winding pattern, as shown on page 80. The basic pattern takes the form of a bow tie and gives first order cancellation of noise induced from the parallel selection line. The center-tapped winding is brought out as a twisted triplet; one wire from the center tap is connected to the inhibit driver, and one wire from each end of the bow-tie winding is connected to a transformer in a balun configuration. With this setup, any tendency of the current in one half of the winding to increase causes the impedance in that side of the balun to increase, blocking the current, and causes the impedance on the other side to decrease. The currents in the two halves are thus kept equal. And the maintenance of this approximate balance of current in the winding

serves to reduce differential noise at the input of the sense amplifier.

The circuit not only provides balanced drive but allows the use of a low supply voltage for the memory because the inhibit driver sees the winding in two parts, each with half the total inductance of the array. These halves, in parallel, present only a quarter of the inductance to the driver, which, however, must provide twice as much current as it would if it were connected to the end of the winding.

During a write operation, the balun presents a very low impedance to common-mode signals and a very high impedance to differential signals. This forces the inhibit current to be equal in both halves of the winding and keeps the differential noise small at the input to the sense amplifier. When reading, the balun is effectively out of the circuit because the sense voltage is much too small to forward-bias the two diodes.

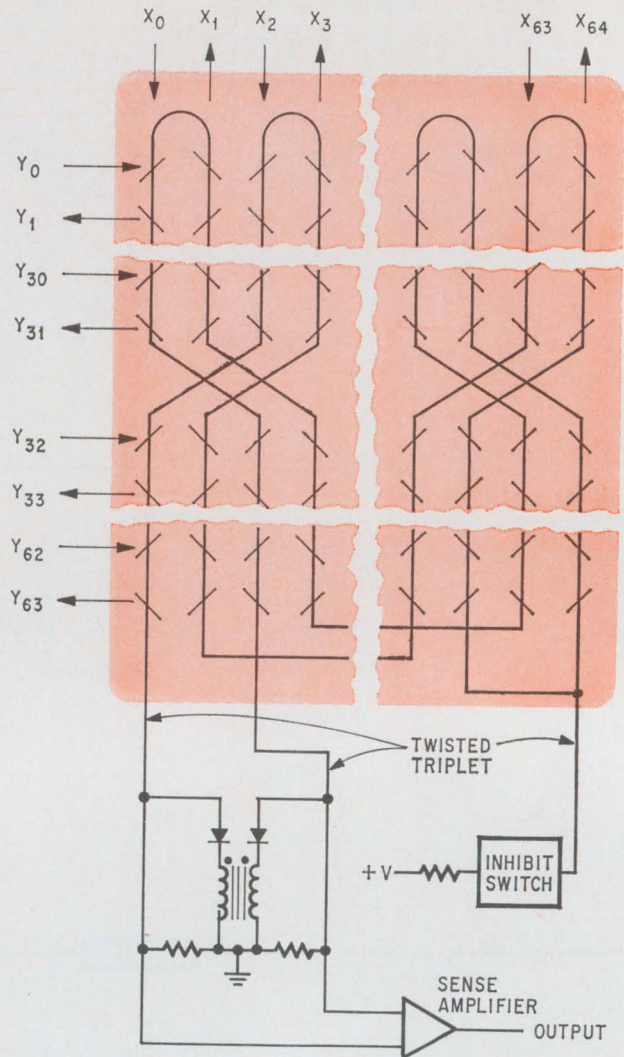
Plane and simple

The Nova's complete 4,096-word memory system, including sixteen 64-by-64 core mats and all the peripheral electronic circuits, is mounted on a single printed-circuit board 15 inches square, as shown on page 81. The system has 32 switches in its drive arrangement—eight at each end of the x and y windings, which thread all the mats in series. Four switches, one from each set of eight, select a single core in each of the 16 mats to read 16 bits in parallel. The core array, fabricated in a single plane, is soldered directly to the board.

Compared with the usual stack of individual planes, this arrangement offers many advantages in terms of cost and reliability. Interconnections are at a minimum because all lines are soldered directly to the main p-c board, all cores are directly accessible and can be repaired without disassembling the array, and lead lengths are short, reducing noise, stray capacitance, and inductance. Best of all, the entire memory is only a half-inch high, leaving plenty of room for more memory or other circuitry in the standard frame, which is 5¼ inches high.

A similar scheme using 32-by-32 mats didn't appear to offer the best packaging arrangement for smaller memories. Of course, smaller arrays cost less, but their use doesn't have much effect on the cost of the associated electronics. For example, chopping the array size from 4,096 to 1,024 words and retaining the 16-bit parallel readout would reduce the number of switches required from 32 to 24—not a very impressive saving.

But the fact that the processor handles bits four at a time affords an opportunity to cut costs by a significant margin. The 1,024-word memory need only be four bits wide instead of 16, and it uses four of the same 64-by-64 mats and the same 32 switches as the larger size. But it has only four inhibit drivers and four sense amplifiers, instead of 16 of each, and this results in a 15% greater saving than the elimination of eight switches and a



Bow tie. This sense winding pattern loads the driver symmetrically, and cancels most noise.

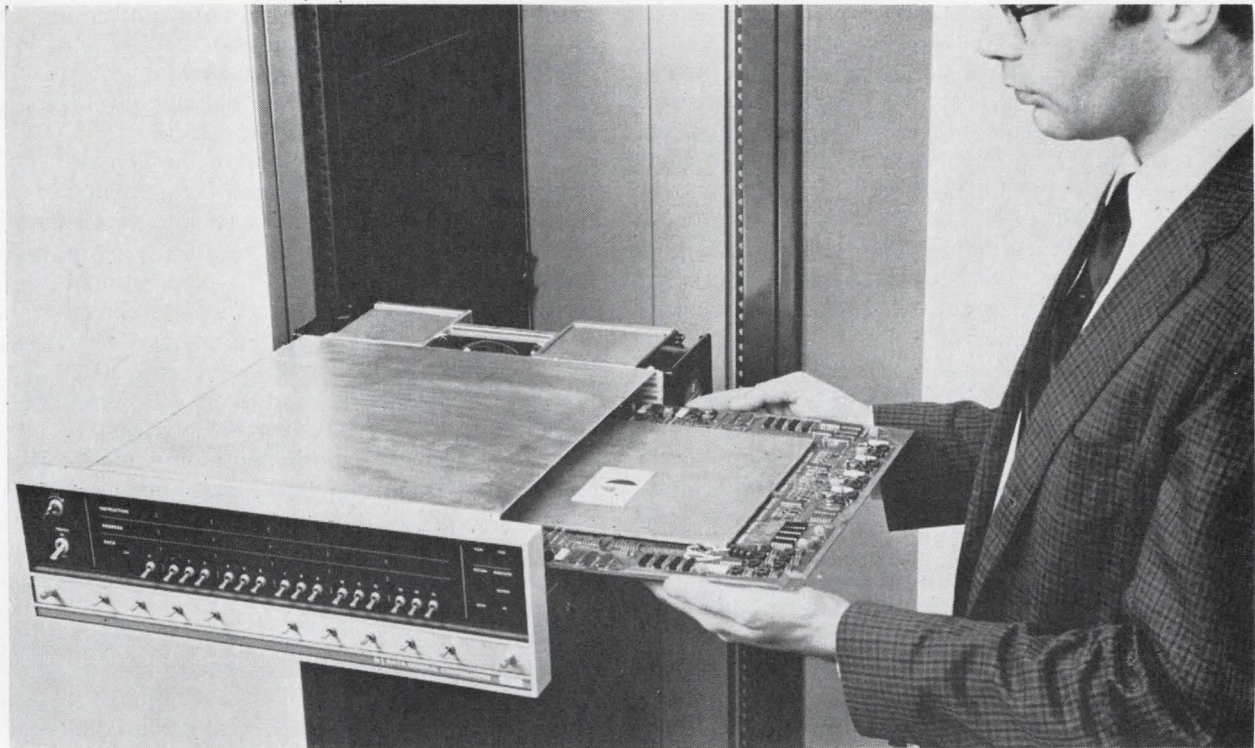
handful of diodes with 32-by-32 mats and a 16-bit-wide memory.

There's a catch, of course. With the larger memory, one memory cycle takes out 16 bits that the processor processes in four cycles. The four-bit-wide memory matches the processor speed cycle for cycle, meaning that equivalent jobs are executed more slowly by the machine with the smaller memory than by one with the larger.

Malleable memory

Because of its unique application, the Nova's read-only memory presented its designers with some requirements not ordinarily associated with this form of storage. Read-only storage has generally been used mainly for microprogram control in fairly complex processors. These memories have to be very fast, but they're also mass-producible because all computers of a given model usually have identical microprograms.

The Nova read-only memory, on the other hand, is a storage medium for the operating program.



Memory board. A complete memory containing 4,096 words and all drivers, decoders, and sense amplifiers, fits on this 15-inch-square board. Design is easily cut back to 1,024 words on a similar layout.

With it, a user can convert a general-purpose processor into a hard-wired control element without having to load new programs. The machine can thus be operated by unskilled workers in electrically noisy areas without affecting the program.

Only a very flexible system could satisfy the needs of this application. Because many of the programs wired into the read-only memory would be one of a kind, high setup or tooling costs couldn't be tolerated. Also, the designers felt that it should be possible to make changes in the field with ordinary tools and techniques. Last but not least, the user should be able to debug his program thoroughly before committing it to the read-only memory.

These requirements are met in the Nova's 1,024-word memory module. It contains 16 of the U-shaped cores with their multiturn sense winding, which functions as the secondary winding of a transformer. A wire for each word is strung through all the cores, passing through a particular core from right to left to store a 1, or from left to right to store a 0. When a current pulse passes through any of these wires, which function as the transformer primary windings, it induces a signal in the secondary winding, the phase of this induced signal depending on the direction of the primary wire through the core. The sense amplifier detects the phase of the output signal to determine whether the bit is a 1 or a 0.

In practice, a user initially loads his program into an alterable core storage and debugs it using

normal procedures. After operationally checking the program's correctness, he simply dumps it on a paper tape and sends it to Data General. This tape serves as the basis for manufacturing the read-only memory and for verifying its contents. The assembled read-only memory can then be plugged into the user's machine in place of that part of the storage that originally contained the program.

If field modifications are required, a technician simply cuts out the wire corresponding to the word to be changed, inserts a new wire in the desired pattern, and solders it to the same parts the old wire was connected to.

Accessibility

There's room for many standard and custom-designed input-output interfaces within the Nova's standard frame. For example, the interfaces for the teleprinter, a high-speed paper-tape reader and punch, and a real-time clock are all mounted on a single board 15 inches square.

Nova's input-output interfaces provide for direct access to the memory, programed data transfers, the status testing of devices, priority interruptions, and identification of the device requesting interrupt service. With the input-output system, which is standard in the basic machine, the program can address up to 64 devices. A single instruction can transfer a word between an accumulator and a device and at the same time control the operation of the device.

A high-speed device such as a magnetic-tape unit

or disk can gain direct access to the memory through a data channel and transfer a great many words following the execution of only one instruction. Likewise, a free-running external device such as a process-control sensor or pulse-height analyzer can store or fetch data in a reserved block of memory at any time without executing any instructions at all. In either case, the program simply pauses as access is made. The data channel logic can transfer data to or from the memory, increment the memory word, or add external data to a word already in the memory.

The interrupt system is particularly useful in process control applications. Several types of interrupt service may be employed by the central processor program depending upon the nature of the data rates and service time required by the input-output devices connected to the system. Each device's interface circuitry includes a flip-flop that enables or disables its interrupt signals.

A single processor instruction called Mask Out can change the flip-flops in all of the device interfaces simultaneously. This instruction, one of the first in a routine to which the computer branches when an external device interrupts it, is accompanied by a 16-bit word on the output data bus. The 1's and 0's of this word turn on or off the flip-flops for the various devices. Since each device's priority is established by the bit in the data bus to which its flip-flop is connected, the computer can enable or disable a particular device's interrupt but cannot change its priority rank.

When a device interrupts the computer, the Mask Out instruction disables the interrupts of all lower-priority devices. If a high-priority interrupt is received while another is being processed, the computer shifts its attention to the higher-priority device but later returns to the lower.

The key to this hierarchical scheme is Mask Out, the kind of instruction usually found only in computers much larger than the Nova.

Another instruction reduces the overhead time required to identify the device causing an interrupt. By transferring the selection code of the interrupting device from the interface to an accumulator, this instruction allows the interrupt service to branch quickly to the routine appropriate to the device.

No weak links

In devising a packaging scheme that minimizes the number of interconnections between components, the Nova's designers sidestepped some cost and reliability problems associated with earlier small computers. In the past, packaging has typically accounted for 40% of the manufacturing cost of small computers. This is true even of some machines using integrated circuits, because their designers simply replaced the discrete components on small printed-circuit boards with IC's, leaving the rest of the system unchanged. In such modules, every circuit connection is made through pins on the back panel; the circuit board functions only as a

carrier for the IC. This technique multiplies the number of interconnections and thus reduces reliability. It also increases the cost of boards, the number of connectors, and the amount of back-panel wiring and cabinet space.

In the case of the Nova, one of the first decisions taken was to use higher-level components to reduce the number of parts. Each integrated circuit in the Nova is of a level of complexity equal to or higher than that found in a conventional logic module circuit card carrying discrete components.

The designers also decided to use only those levels of interconnection that increased the complexity of the system. In other words, they determined to avoid wasteful circuit carriers.

Putting high-level components on printed-circuit boards reduced wire lengths and increased reliability. The size of the printed-circuit board itself wasn't too important. With the conventional module replaced by the IC, boards could contain whole subsystems.

The conductors on the Nova's p-c boards replace almost all conventional back-panel wiring, while what back-panel wiring there is replaces cables. Thus the traditional level of connection complexity for this kind of system has been escalated one degree. Even when small IC modules are required for special device interfaces, connecting them on larger boards is better than accumulating nonfunctional connections in back-panel wiring.

Board size isn't limited by the capacity of existing cameras, copy plates, etchers, solderers, or other production equipment. Even those firms that are using small modules are building big boards and cutting them apart. Nor is yield a problem. Poor yield results from an inability to control process variables; lines are either etched through or short-circuited. The wider the line and the greater the spacing, the less likely are defects, and the Nova's boards have wide lines and wide spaces between them.

If a customer wants to configure his own system from the various memory sizes and peripheral devices available, or wants to interface special devices of his own, he'll find the Nova design accommodating.

The computer's basic rack mount unit holds seven 15-inch-square circuit boards. The central processor takes up two of these cards, so the other five are available for any combination of memory boards and input-output boards.

Interface boards are available for a complete line of standard computer peripherals, including teleprinters, card readers, line printers, displays, and magnetic tape. The customer can build special interfaces by mounting IC's on standard plug-in socket boards purchased from Data General. These boards contain eight interface areas, each with space for 12 standard 14-, 16-, 24-, or 36-pin IC's, either soldered or plugged in.

The user can also buy special circuit boards containing certain kinds of high-current drivers that have to be assembled from discrete components. ■