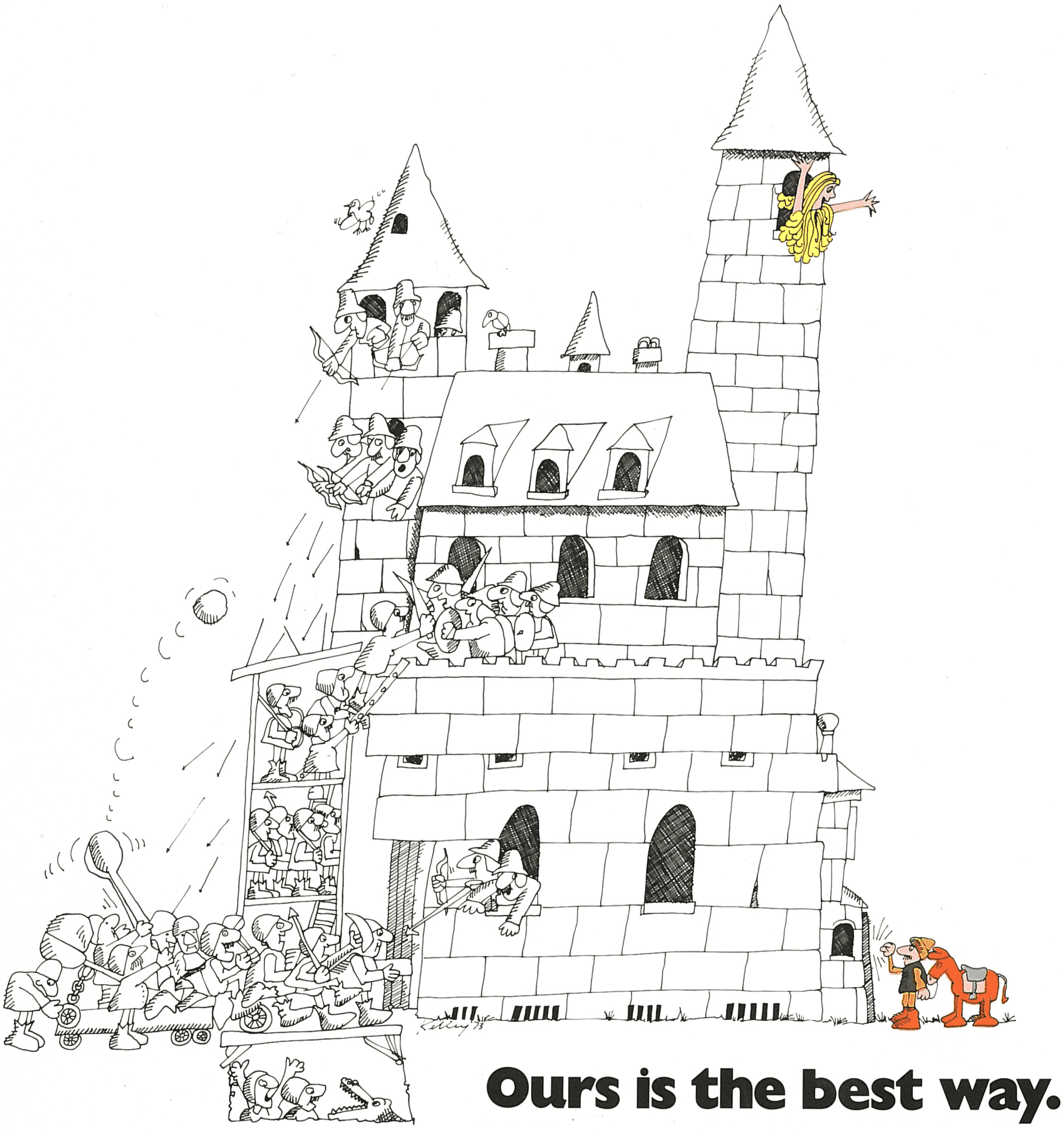
DATA GENERAL
CORPORATION

There are lots of ways to get there...



Ours is the best way. Nova 2.

There are lots of ways to design a computer. A few get the job done right, with a simple straight-forward approach. Others try awfully hard, make a lot of effort, but still miss the mark.

There are lots of ways to get there. Ours is the best way. Nova 2.

It's best because it has to be, to stand up under close scrutiny of people like you. People who have been building, buying and working with computer systems for years. People who know what's good design, and what isn't.

It's best because it's inexpensive. In fact, a Nova 2 with 32K words of memory actually costs less than many comparable computers with 16K words of memory.

It's best because it lets you program in high-level languages. All that extra memory makes the difference between assembly language and languages like Fortran 5. You develop programs in record time, and time is money.

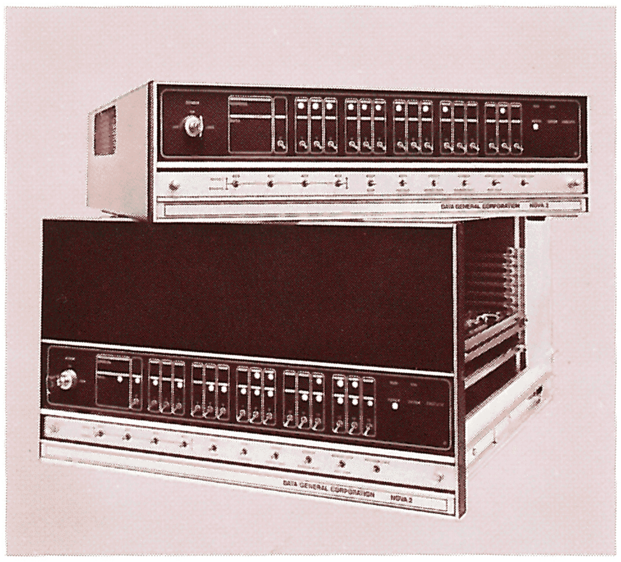
It's best because it makes your programs execute fast. Compilers like Fortran 5 transform source programs into densely packed object code with minimum execution time. And Nova 2's high-performance central processor unit lets assembly language programs execute that much faster. Good for even the most demanding real-time applications.

It's best because it gives you standard and custom computer systems in one package. Nova 2 comes in twenty off-the-shelf configurations. You customize simply by selecting appropriate options, peripherals, interfaces and software.

It's best because it makes use of reliable Nova-line design concepts, peripherals, and software already proven in over 8,000 installations around the world. That's a firm indication of our continuing commitment to our product line, and to you.

With Nova 2, there are no tradeoffs. You get high-level language convenience, fast program development, fast program execution, standard configurations, and custom systems, all at remarkably low cost, from a well-established company that's committed to a continuing compatibility in its product line.

There are lots of ways to design a computer. Ours is the best way. Nova 2.



Best because it's inexpensive

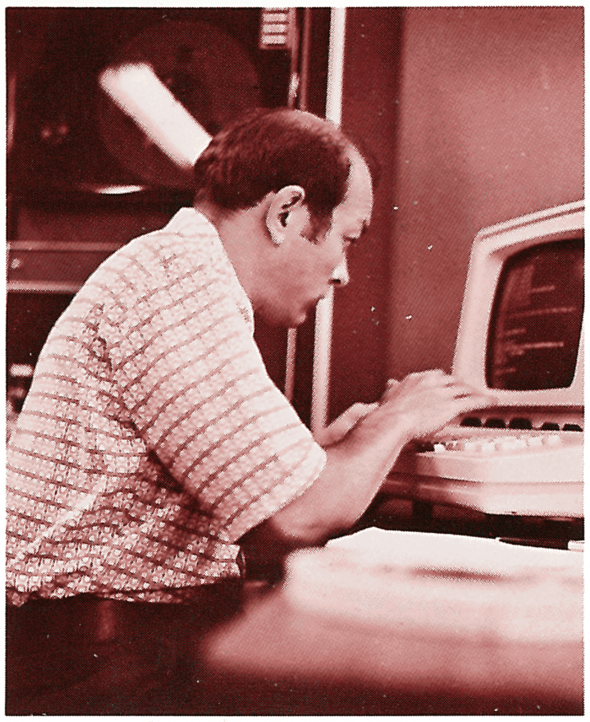
Prices of Nova 2 systems are low. So low that a 16K Nova 2 costs about as much as many comparable computers with only 8K words of memory. At the same time, Nova 2 includes high-performance features, like a multi-accumulator central processor unit, a 16-bit word length, I/O system with programmed data transfer, 16-level programmed priority interrupt, high-speed Direct Memory Access (DMA) data channel, programmer's console, multiple-slot mainframe, and power supply.

The low-cost/high-performance Nova 2 configurations let users do lots of things they simply can't do economically with other computers. For one, it lets users write applications software in high-level languages, and still keep the cost of the hardware low.

For a customer using ten Nova 2 systems a year, the difference between an 8K and a 16K Nova 2 is less than $1,350 per system. For the additional cost, programs can be written with a language processor like Fortran rather than in assembly language. Since the total cost of a programmer is conservatively estimated at about $30,000 a year ($2,500 a month), only two weeks of a programmer's time, per system, have to be saved by using a high-level language processor, to justify the extra memory.

Low-cost Nova 2 systems also make assembly language programming easier. Simply stated, it's usually more economical to buy additional Nova 2 memory to accommodate a program than it is to spend time making a program fit into minimum core. And programs so structured will generally run faster than those optimized for minimum core. You write programs in less time. Your programs take less time to run. And time, in both cases, is money.

No matter how you look at it, Nova 2 offers the best price/performance ratio on the market. Low hardware cost. High-level and assembly-language programming flexibility.



Best because it's fast

Nova 2 is fast enough for even demanding real-time applications.

Efficient Architecture

Its speed starts with a central processor organized around four 16-bit accumulators. The multi-accumulator architecture has large advantages in terms of efficient memory use and programming ease as well as speed. It cuts down on the number of instructions needed to execute a program, reduces data movement, and simplifies programming.

General-Purpose Instruction Set

The efficiency of the Nova 2 architecture is complemented by its 16-bit general-purpose instruction set, suitable for a wide range of applications. The maximum possible number of functions has been built into each instruction, reducing the amount of code which must be written. For example, arithmetic and logical instructions, in addition to their eight basic functions, can modify an operand, shift the result, and/or test the result. Altogether, a total of 256 variations exist for each arithmetic and logical instruction.

High-Speed Input/Output

Nova 2 has as standard equipment a high-speed Direct Memory Access (DMA) data channel which transfers data directly from peripherals to memory. Because the central processor is not involved, transfer takes place at high speeds: 1.25MHz maximum with 800-nanosecond memory; 1.0MHz maximum with 1000-nanosecond memory.

In addition to an extremely high transfer rate, the Nova 2 I/O facility has very short interrupt and data channel latencies. Both of these high-speed I/O features make Nova 2 well-suited for a variety of real-time applications.

Superfast Instruction Times

The facts speak for themselves:

	800-nsec memory (4K, 8K modules)	1000-nsec memory (16K modules)
LDA	1.6 μsec	2.0 μsec
STA	1.6	2.0
ISZ, DSZ	1.9[1]	2.3
[1]For skip add	.1	0
JMP	.8	1.0
JSR	1.1	1.2
Indirect addressing add	.8	1.0
Base register addressing add	0	0
Auto indexing add	.5	.5
COM, NEG, MOV, INC	.8[2]	1.0[2]
ADC, SUB, ADD, AND	.8[2]	1.0[2]
[2]For skip add	.3	.2
IO input	1.4[3]	1.5[3]
NIO	1.6[3]	1.7[3]
IO output	1.6[3]	1.7[3]
[3]For S, C, or P add	.3	.3
IO skips	1.1[4]	1.2[4]
[4]For skip add	.3	.3
INTA	1.4	1.5
MUL[5]	5.5	5.6
DIV[5]	5.8	5.9
interrupt	1.6	2.0
Latency (with multiply/divide)[6]	5.8	5.9
Latency (w/o multiply/divide)[6]	1.9	2.3
Data channel		
Input	2.1[7]	2.2[7]
Output	2.1	2.2
Increment	2.2	2.3
Latency (max.)	2.7	3.1
High-speed data channel		
Input	.8	1.0
Output	1.2	1.3
Increment	1.3	1.4
Latency (max.)	2.7	3.1

[5] With hardware multiple/divide option

[6] Assumes no indirect addressing

[7] Isolated transfer. Sequential transfer, 1.9μsec

Memory Overlapping Saves Time

Another Nova 2 feature which increases speed is memory overlapping. Memory overlapping means that the system can simultaneously read information from one memory module while it rewrites previously accessed data into another memory module. In many computers, the "reading" function cannot proceed until the "rewriting" function is complete. By overlapping functions, Nova 2 is able to complete its work sooner.

Fast Run-Time With Fortran 5

Even with high-level language processors like Fortran 5, program execution is fast. For example, Fortran 5 globally optimizes user source code. Each statement is inspected and, if necessary, modified for efficiency. Then it's compared to all other statements to eliminate redundancy. The result is efficient object code that occupies minimum space and runs at optimum speed.

Final Result: Increased Throughput

The effect of Nova 2's short I/O latency, in conjunction with high-speed data channel transfers, and memory overlapping, is to increase system throughput. That means more features built into your system, and more jobs, per dollar invested.

Best because it's easy to use

Nova 2 is available with a wide range of operating systems and language processors that greatly simplify programming and optimize the hardware capabilities.

Operating Systems for Program Efficiency

The Real-time Disc Operating System (RDOS) is a modular, multitask system operating with 12K or more memory, disc, real-time clock and console terminal. RDOS is both a powerful program development tool and run time executive. It greatly simplifies the transition from program development to system operation. In a program development environment, RDOS supports Data General's Fortran IV, Fortran 5, Extended Algol, Extended Timesharing BASIC, MACRO Assembler, editors, debuggers, Batch monitor, and a library of utility programs. As a run-time executive, RDOS handles file management, monitors access to I/O devices, and schedules tasks in a real-time or Batch environment.

The Real-Time Operating System (RTOS) is a compatible subset of RDOS. That means applications programs written under RDOS can execute under RTOS. It operates with 4K or more memory, console terminal and real-time clock. In applications not requiring program overlay or file management, RTOS is a modular interface to user programs in real-time or off-line environments. Multitasking programs (including Fortran IV and Fortran 5), timer control, and I/O transfers are handled by simple task and system calls to RTOS. Standard Data General peripherals are supported. For peripherals requiring customer-designed interfaces, both RDOS and RTOS readily accommodate user-written software device drivers.

For program development in a stand-alone environment, the Stand-alone Operating System (SOS), a compatible subset of RDOS, allows users to edit, assemble or execute programs stored on paper, magnetic, or cassette tape. It provides buffered service of I/O peripherals on a device-independent basis.

Language Processors to Simplify Programming

Extended Fortran IV gives Nova 2 all the processing facilities of large scale computers. It is full ANSI Fortran IV with important features for real-time applications: ISA real-time extensions, reentrant code, multitasking, easy interfacing to assembly language subroutines, operation under RDOS, RTOS, or SOS. It is also extended Fortran IV, allowing lower as well as upper bounds for array dimensions, mixed mode expressions, strings within quotes, conditional subroutine returns, and recursive subroutines.

Extended Fortran 5 is the most powerful Fortran ever available on a minicomputer. It is a superset of Data General's Fortran IV, ANSI Fortran, IBM Fortran IV (H extended), and Univac Fortran 5. Its superset nature makes conversion from other machines easy. Fortran 5 has all the real-time features of Data General's Fortran IV. Other extensions include static and dynamic storage allocation, PARAMETER declaration, statement functions expanded as in-line code. And it globally optimizes user source programs into object code minimizing execution time. To assist the programmer, the compiler features nearly 250 easily interpreted error messages at compile time.

Extended Algol has all the features found in Algol 60. Extensions allow the use of character strings, pointer and based variables, and subscripted labels. This version of Algol also provides virtually unlimited precision arithmetic, allowing up to 30 digits of precision.

Algol 60 is so powerful that we use it as a systems programming language to write compilers. And at least one of our OEM customers has done the same.

BASIC is the most widely used timesharing language available. It comes in both timesharing and single user versions, both full implementations of the BASIC language developed at Dartmouth College. Data General's BASIC includes string and matrix extensions, and even allows complete matrices to be read or written in a single I/O call. In addition, Extended BASIC allows the accessing of multiple peripheral devices, including disc memories under control of RDOS, in a formatted or unformatted ASCII, or binary I/O mode.

Or, if you'd rather get closer to the program...

We've got lots of assemblers, debuggers and utilities that help you get there.

Our MACRO Assembler, like our relocatable assembler, converts symbolic assembly statements into machine language code. It also includes a number of features that simplify the programming task: an extensive macro library including macros for often-used operations; a powerful macro facility allowing complete recursion and practically no limit on the number of macro definitions; a means of redefining symbol values; and conditional assembly through DO loops and branch statements.

Nova 2 also runs with a completely symbolic debugger. The symbolic debugger can accept from the relocatable loader a symbol table containing local as well as global symbols, so the user can employ symbolic names, rather than absolute addresses, while he is debugging a program. Up to eight breakpoints within a user program can be active at a time. Memory searches and dumps, instruction examination and modification, and program patches can be made using instruction format commands rather than octal commands. Programming is simpler for novices, since there is no need to know octal instruction codes.

Lots of utility programs are available. A two-pass relocatable assembler producing absolute binary, an assembly listing, and relocatable binary. It allows double-precision integers, floating point constants and conditional assembly. A text editor that works on both a character and a line basis letting users correct minor text errors without deleting the entire line. Lots of code conversion programs. Arithmetic programs that manipulate single- and double-precision integers and perform floating point arithmetic.

Three cross assemblers, all written in Fortran, permit assembly of Nova symbolic source code into machine object code for Nova 2, using card input to the IBM 360, CDC 6600, or Univac 1108. Output can be in absolute or relocatable binary suitable as input for Data General's binary or relocatable loader.

Best because it's a complete system

Nova 2 is not just another piece of hardware. It's a series of computer systems that can be as basic or as sophisticated as you need.

Two Models, Twenty Configurations

To start with, Nova 2 is available in two models. The Nova 2/4 is the smaller, with four subassembly slots in a 5¼-inch high chassis. The Nova 2/10 is the larger, with ten subassembly slots in a 10½-inch high chassis. Both models are available in stand-alone table-top versions or in rack-mountable versions which can be embedded easily in a larger system. Both are prepackaged with a central processor and 4K, 8K, 16K, 24K, or 32K words of core memory. Altogether, there are twenty different configurations to choose from.

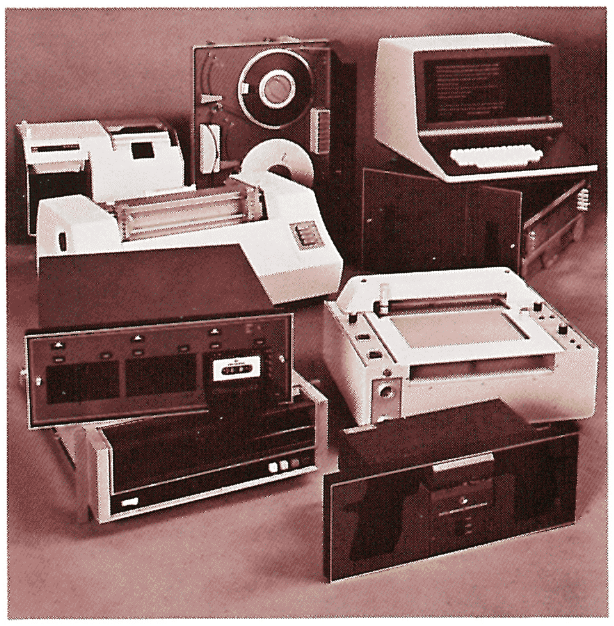
Lots of Options

Numerous options are offered to adapt Nova 2 to specific customer requirements. Integer hardware multiply/divide. A floating point unit that performs single- and double-precision arithmetic at high speed. Complete execution of a double-precision floating multiply takes less than 20usec, with 800-nanosecond memory. A low-cost turnkey console with start, continue, reset and program load functions that's perfect for dedicated applications. Also options that allow unattended operations: power monitor/automatic restart, to sense power failure in time to provide an orderly shut-down, and restart when power is restored. And automatic program load.

Lots of Peripherals

Nova 2 operates with peripheral devices in every major category. Synchronous read/write 7- or 9-track magnetic tape transports from 12.5 to 75 ips. High-speed reel-to-reel Nova Cassettes with an average transfer rate of 1600 characters/second. High-speed fixed-head Novadiscs with 128K to 768K 16-bit word capacity and 8.4msec average access. Two moving-head cartridge drives, one with 1.247 million 16-bit word capacity, one with a 2.494 million 16-bit word capacity. Moving-head disc pack drives, with 3.118 million and 12.472 million 16-bit word capacity. Punched and mark sense card readers, operating at 150 to 1000 cards/minute. Perforated tape reader (400 characters/second) and punch (63.3 characters/second). Teletypewriters. Drum and flatbed incremental plotters, operating at 200 or 300 increments/second. 64-character line printers (245 to 1100 lines/minute) and serial matrix printers operating at 165 characters/second. 24-line, 80-character Novadisplays with variable code structure and baud rate. Real-time clock with AC line or crystal source.

All of these peripherals are completely interfaced and supported as part of a total Nova 2 system.



Lots of Input/Output Devices

Nova 2 is also available with a wide range of input/output devices. High-level and wide-range analog input systems with up to 512 channels. Digital-to-analog converters up to 24 channels. Digital I/O interfaces. An interface that directly connects Nova 2 to IBM 360/370 I/O channels, and lets a Nova 2 system function as a front end processor, data collector, or as any of numerous other devices. A multiprocessor communications adapter that allows up to fifteen Nova-line computers to be connected through their data channel facilities into a processing network.

A complete range of communications options is also available with Nova 2. Asynchronous devices include single- and multi-line (64-line modules) controllers at data rates to 9600 baud. Synchronous devices include single-line controllers at data rates up to 50K baud, and multi-line (up to 64) controllers at data rates up to 9600 baud. A wide range of system redundancy components for data communications is available, featuring dual-access communications controllers and peripherals.

Lots of Software

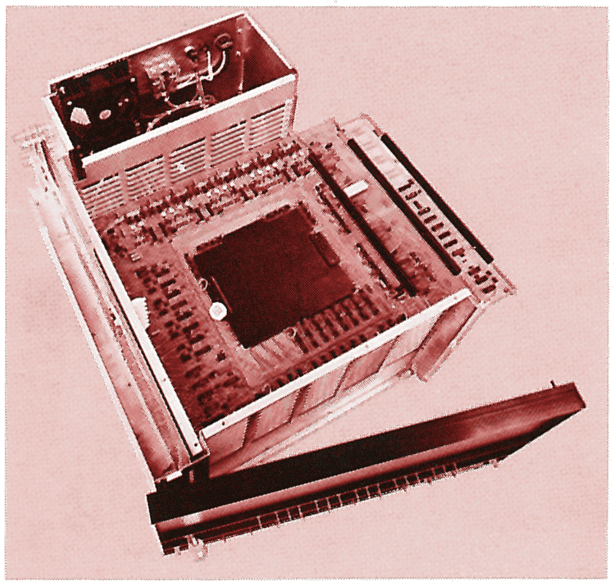
The software available with Nova 2 is as extensive as the available hardware. It includes standard and relocatable assemblers, cross assemblers (for IBM 360, Univac 1108, CDC 6600), diagnostics, and utility programs. Extended Fortran IV, Fortran 5, Extended Algol, and single-user and timesharing BASIC are language processors which can be used with proper hardware configurations. Three compatible operating systems are available: a Real-time Disc Operating System (RDOS), a Real-Time Operating System (RTOS), and a Stand-alone Operating System (SOS).

Best because it's dependable

We Started With Proven Design Concepts...

Reliability was a key factor in the design of Nova 2. To get it, we first turned to our own extensive line of computers. From them we borrowed a number of efficient, reliable design concepts, proven in over 7,000 computer installations around the world.

The Nova 2 chassis is organized around a 15-inch square format. Each major subassembly (including central processor unit and memories) is mounted on a 15-inch square printed circuit board. These boards are inserted in slots in the computer chassis, and plug into a common back panel through which all interboard and external connections are made. The back panel includes convenient, reliable plug-in connectors for commonly specified peripherals, including teletypewriters, paper tape equipment, magnetic tape units, and disc drives. In many configurations, this eliminates the need for wirewrap connections on the back panel. The front panel is a separate assembly, and plugs into the back panel.

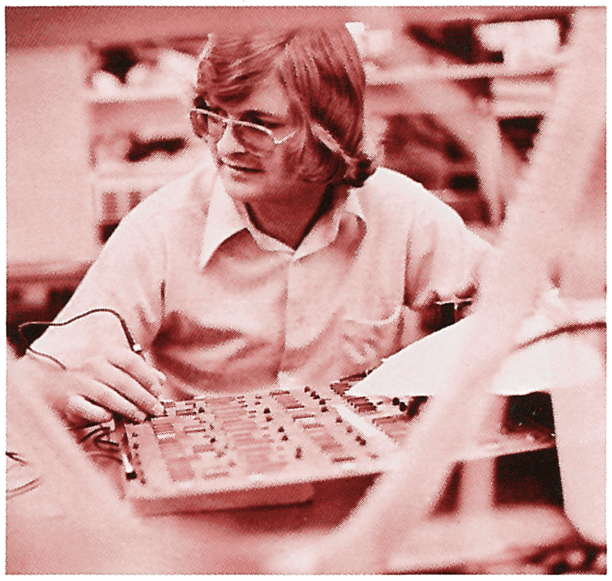


...And Added New Hardware And High-Density Packaging

Nova 2 takes advantage of the highest level of integrated circuitry available (both large- and medium-scale) to achieve very high performance, high reliability, and low cost. The entire central processor unit fits on a single printed circuit board. Each memory (including a 16K version) fits on a single printed circuit board. Because the Nova 2 uses a high degree of integrated circuitry, the parts count is low, component life expectancy is high, and there are few connections, therefore few potential failures.

Testing For Reliability

Before the Nova 2 design was finalized, our quality assurance program was in process. That way, we were able to design reliability right into Nova 2. We inspected numerous potential vendors, and carefully selected those who would supply Nova 2 components.



Incoming Test: The components we buy are stringently tested as they come in. That includes integrated circuits, transformers, peripherals, circuit boards, castings, pieces of hardware and sheet metal.

In-Process Test: We test our product as it's assembled. Boards are tested after they are flow soldered. Power supplies are tested after assembly. Memory circuit boards are tested before the core plane is attached. Memories, central processor units and options get run through diagnostics at room and at elevated temperature.

Final Acceptance: Finally, we double check details from solder joints to revision levels. Then we run mainframes under our own Diagnostic Tape Operating System. It scans the hardware for its configuration, and runs appropriate diagnostics under automatic program sequences. Next we "burn-in" the mainframes with a tough "Exerciser" that runs memory, arithmetic, basic I/O and timing sequences for many hours. If an error occurs at any point after burn-in, we correct the source of the error, then repeat the test. Only when all diagnostics run without a hitch do we approve the system.

But The Proof Is Performance

Granted, our testing for Nova 2 is thorough. But the real proof of reliability is performance. Nova 2/10 has a calculated Mean-Time-Between-Failure figure of 7,255 hours (calculated per MIL-STD-217A, including power supply, console, central processor unit, 8K memory). *That's* reliability.

If a Nova 2 does require maintenance, troubleshooting and service are remarkably simple. Few boards are involved, so a malfunction can usually be isolated very easily to a single board. A spare can be installed rapidly, even in the field, and little operating time is lost. *That's* dependability.

Best because it can be integrated into your system

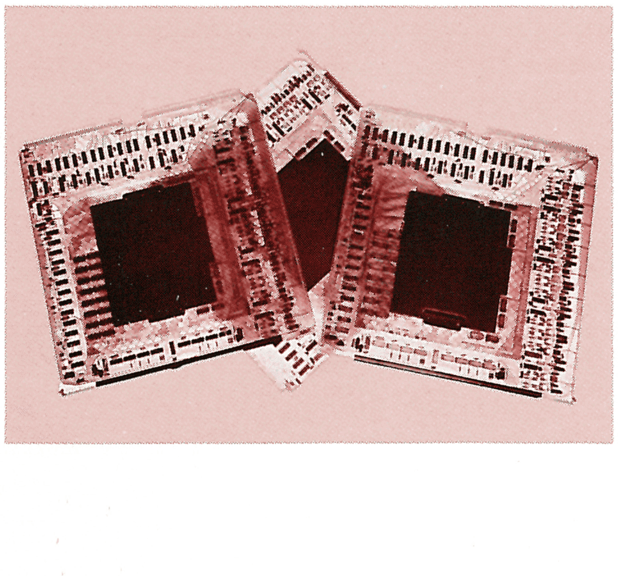
Space-Saving Packaging

In Nova 2, a lot of sophisticated high-density packaging techniques were used to optimize valuable computer space, and to make the computer itself fit into a simple compact package. A high degree of integrated circuitry (large- and medium-scaie) was used in the design of the Nova 2 central processor unit. It fits on a single printed circuit board, and still has space for options such as power monitor/auto restart, and automatic program load. Each memory (including the 16K version) fits on a single printed circuit board and occupies only one slot in the computer chassis.

Even fully loaded with memory, interfaces and options, Nova 2 takes up minimum space. The Nova 2/4 requires only 5¼ inches of rack space, and weighs only 50 pounds. Nova 2/10 is 10½ inches high and weighs only 115 pounds.

Asynchronous Memories: Different Speeds for Different Applications

Nova 2 is available with two types of memories: 800-nanosecond 4K and 8K modules, and 1000-nanosecond 16K modules. The two types of memories can be interchanged as the application requires. If speed is most important, the 800-nanosecond memories can be used. If space economy is a prime consideration, the 1000-nanosecond 16K modules can be used. If both space and speed are important, the memories can be mixed in Nova 2. The effective cycle speed of the computer then depends on which memory module is being accessed.



I/O Facility for High- and Low-Speed Peripherals

Nova 2 has both program-controlled and high-speed data transfer. The program-controlled data transfer is an economical way to service relatively low-speed peripherals such as teletypewriters and card readers. High-speed Direct Memory Access (DMA) is for high-speed peripherals like discs and magnetic tapes. The I/O system also has a 16-bit word length, automatic interrupt source identification, a full 62-device addressing capability, and a a 16-level programmed priority interrupt structure.

Compact, Multi-Purpose Interfaces

Many of the I/O interfaces supplied with Nova 2 are multi-purpose, working with more than one peripheral. For example, a single subassembly can be ordered with any combination of the following: the real-time clock, teletypewriter I/O interface, paper tape reader control and paper tape punch control. Such packaging conserves space, reduces hardware cost and gives users tremendous flexibility in system configuration.

An Easy Way to Design Custom Interfaces...

Our general-purpose interface components let users design and build custom interfaces quickly and easily. The general-purpose interfaces contain basic I/O logic which connects a peripheral to the I/O bus: device selection network, data bus drive and receive logic, BUSY and DONE logic, interrupt request and acknowledge logic, interrupt mask, and I/O signal selection. The board also contains pre-drilled space (or connections) where customer logic can be added to the basic I/O logic. Boards are 15-inches square, and fit in one Nova 2 subassembly slot.

Two wiring board series, without logic, are available. They are widely used for both general-purpose prototyping and fabrication of custom interfaces. Both wiring board series are drilled to readily accept customer logic, and are also available with sockets and/or wirewrap pins.

...Or We'll Do It For You

We can handle those difficult interfacing jobs you'd rather not. Our custom products group will, on request, quote on the design and manufacture of virtually any Nova 2 interface.

Best because it's backed by a strong customer support organization

We realize that good support materials for our customers are as important as the design of our equipment. That's why each customer is offered thorough hardware and software training, complete documentation, and a variety of service agreements.

Field, Factory Service

Our service facilities for the Nova 2 are among the most complete in the industry. Either of two comprehensive contracts is available for a monthly charge. Under one, Data General will service contracted equipment at the customer's site. Under the other, equipment will be repaired at any of several repair centers. Service can also be arranged at the factory or the customer's site on a non-contracted time-and-materials basis.

You Need Us, We'll Travel. And Fix Your Problem Fast.

Contracted or not, if you need us, we will be at your site in minimum time. Our service organization is worldwide, and staffed by a group of highly competent engineers.

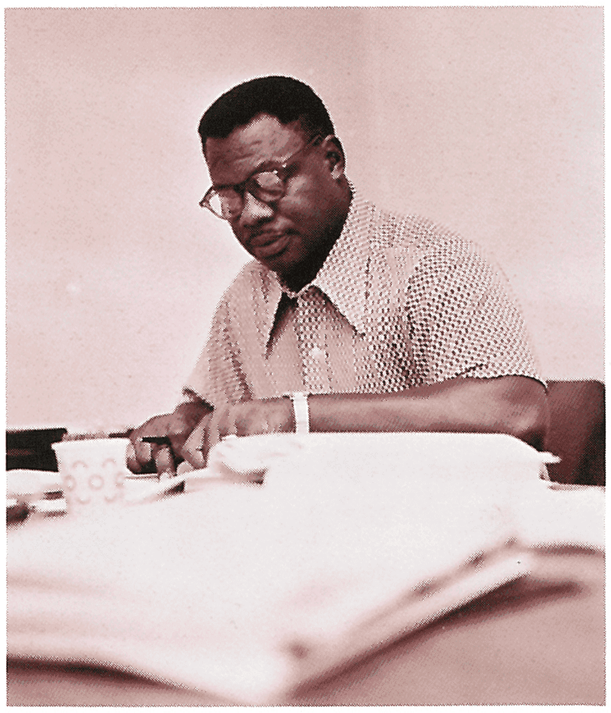
Since the first Nova was installed over 8,000 computers ago, our field service staff has established a reputation for timely, efficient service. That record has been built on the combination of a highly professional service staff and a modular, easily serviced product like Nova 2.

Loaner Boards, To Keep You Running

If you have a subassembly board that has to go back to the factory, your system can stay running. Since Nova 2's subassemblies are packaged on discrete 15-inch square boards, it's a simple matter to locate the defective board and replace it with a loaner board (if you don't have a spare). The result: you're up and running with minimum time lost. Loaner boards are available for a small monthly charge from the Data General factory.

Comprehensive Training

A wide range of training courses is available, and repeated at frequent intervals, both at the Southboro headquarters and at selected field locations. The Nova 2 hardware course is a full five days long. It covers memory and central processor unit operation, logic and flow analysis, instruction set, basic I/O programming, and operating procedures. Software courses are offered for both beginning and advanced programmers. Other courses cover mechanical and electronic aspects of peripherals, options, and controllers.

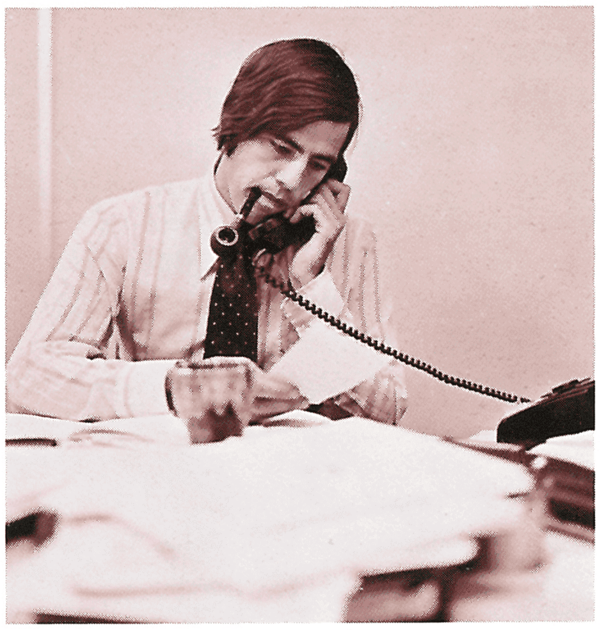


Complete Documentation

Our training program is complemented with a complete documentation package on Nova 2. The package covers use of the computer's software, interfacing, installation, and maintenance.

Application Engineering

We have a large staff of applications engineers whose major responsibility is to help you understand our product line so it can solve your problems. They provide the kind of support that is so crucial during the development of a product or system. And they're as close as your telephone.



A fast wrap-up

Nova 2 is economical. It is available in several low-cost, large memory configurations that economically allow programming in high-level or assembly languages.

Nova 2 is fast. It has a high-speed central processor unit, asynchronous memories, high-speed input/output, and memory overlapping.

Nova 2 is easy to use, whether through high-level languages and operating systems, or with assembly language.

Nova 2 is suitable for a large range of applications. It lets you configure the system, large or small, with or without peripherals, with or without operating systems, with or without options.

Nova 2 is dependable. It uses proven design concepts, and is carefully designed, tested and manufactured.

Nova 2 is easy to customize. It runs with two different-speed memories, handles low- and high-speed peripherals with ease, and is available with hardware that lets you design special-purpose interfaces.

Nova 2 is compact, putting 16K 16-bit words of memory on one board, a high-speed central processor on one board, and up to ten complete subassemblies in a single chassis.

Nova 2 is backed by a worldwide training/customer service organization that guarantees complete documentation and rapid repair for any Nova 2 subsystem, at the factory or at the customer's site.

But the most remarkable fact is that Nova 2 is all of these. You get all the convenience of high-level languages and a low, low system hardware cost.

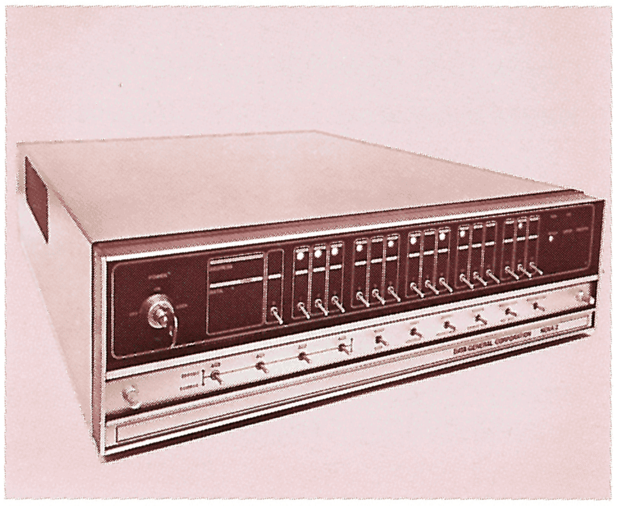
Short program development cycles and, thanks to the efficiencies of language processors like Fortran 5, very fast program execution.

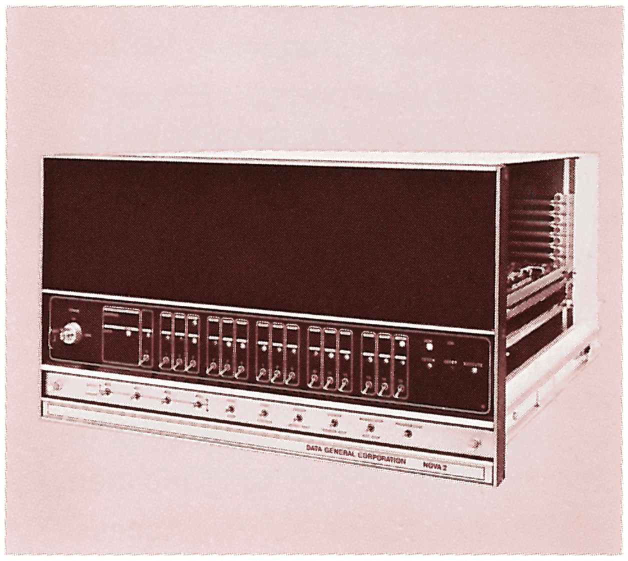
Standard, off-the-shelf configurations and, with the change of a subassembly, front panel or peripheral, custom solutions.

High-density, compact packaging and easily serviced hardware, so modular that its replacement doesn't cost you an arm and a leg if it happens.

Efficient, highly reliable hardware and software. And a complete service/training organization that's there when you need it.

There are lots of ways to design a computer. But ours is the best way. Nova 2.





Nova 2 in depth

Hardware Specifications
Nova 2/4

GENERAL

Word Length: 16 bits.

Hardware Accumulators: 4.

Index Registers: 2 hardware, 16 memory.

Address Modes: Direct addressing of 1024 words, absolute, relative and indexed modes; multi-level indirect addressing of 32,768 words.

Memory Cycle Time: 800-nanosecond and 1000-nanosecond core memory.

Memory Capacity: 32K 16-bit words.

Memory Increments: 4K and 8K 16-bit words, 800-nanosecond memory; 16K 16-bit words, 1000-nanosecond memory.

High-Speed Direct Memory Access Channel: Standard; maximum word transfer rate, 1.25MHz.

Input/Output System: 16-bit word length, 16 priority interrupt levels, 62 devices addressable.

I/O Bus Levels: Ground and +3 volts.

Power Requirements: 115 volts (±20%), 10 amps, or 230 volts (±20%), 5 amps; 47-63Hz, single-phase; max power dissipation, 300 watts.

Heat Generated: 1050 BTU/hr max.

MECHANICAL

Dimensions: 5¼"H x 19"W x 19"D, 4 slots.

Weight: 50 lbs.

Power Cable: 6' long, wired to computer; other end Belden NEMA-type 5-15P molded vinyl grounding plug.

ENVIRONMENTAL

Temperature Range: 0 to +55°C operating; −35°C to +70°C storage.

Relative Humidity Range: to 90% operating; to 95% storage.

Altitude Range: to 10,000' operating; to 50,000' storage.

Hardware Specifications
Nova 2/10

GENERAL

Word Length: 16 bits.

Hardware Accumulators: 4.

Index Registers: 2 hardware, 16 memory.

Address Modes: Direct addressing of 1024 words; absolute, relative and indexed modes; multi-level indirect addressing of 32,768 words.

Memory Cycle Time: 800-nanosecond and 1000-nanosecond core memory.

Memory Capacity: 32K 16-bit words.

Memory Increments: 4K and 8K 16-bit words, 800-nanosecond memory; 16K 16-bit words, 1000-nanosecond memory.

High-Speed Direct Memory Access Channel: Standard; maximum word transfer rate, 1.25MHz.

Input/Output System: 16-bit word length, 16 priority interrupt levels, 62 devices addressable.

I/O Bus Levels: Ground and +3 volts.

Power Requirements: 115 volts (±20%), 10 amps, or 230 volts (±20%), 5 amps; 50Hz (±1Hz) or 60Hz (±1Hz) single-phase; max power dissipation, 725 watts.

Heat Generated: 2500 BTU/hr max.

MECHANICAL

Dimensions: 10½"H x 19"W x 24¼"D, 10 slots.

Weight: 115 lbs.

Power Cable: 6' long, wired to computer; other end Belden NEMA-type 5-15P molded vinyl grounding plug.

ENVIRONMENTAL

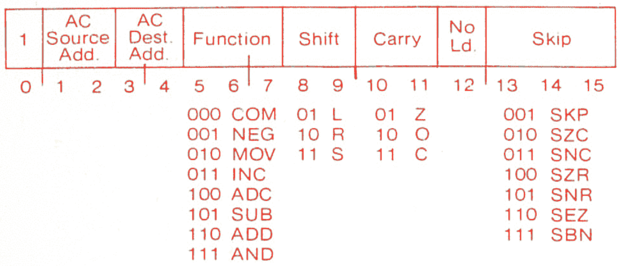
Temperature Range: 0 to +55°C operating; −35°C to +70°C storage.

Relative Humidity Range: to 90% operating; to 95% storage.

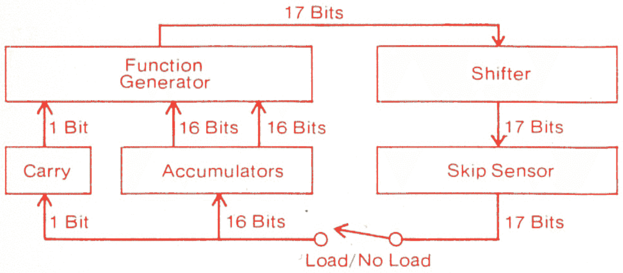
Altitude Range: to 10,000' operating; to 50,000' storage.

Instruction Set

Arithmetic and Logical Instructions The structure of an arithmetic or logical instruction word is shown below.



Each instruction in this class specifies one or two accumulators to supply operands to the function generator, which performs the function specified and produces a carry bit, whose value depends upon a base value specified by the instruction, the function performed, and the result obtained. The base value may be derived from the Carry flag, or the instruction may specify an independent value.



Organization of Arithmetic Unit

The 17-bit output of the function generator, comprising the carry bit and the 16-bit function result, then goes to the shifter. Here the 17-bit result can be rotated one place right or left, or the two 8-bit halves of the 16-bit function result can be swapped without affecting the carry bit. The 17-bit shifter output can then be tested for a skip; the skip sensor can test whether the carry bit or the rest of the 17-bit word is or is not equal to zero. The 17-bit shifted word can be loaded into Carry and one of the accumulators selected by the instruction. However, loading is not necessary: an instruction can perform a complicated arithmetic and shifting operation and test the result for a skip without affecting Carry or any accumulator.

The Carry flag can be used in conjunction with the sign of a result to detect overflow in operations on signed numbers, but its primary use is as a carry out of the most significant bit in operations on unsigned numbers, such as the lower order parts in multiple precision arithmetic. For unsigned numbers, a carry is produced if addition or incrementing increases the number beyond $2^{16} - 1$. In subtraction, the condition is the same if, instead of subtracting, the complement of the subtrahend is added and 1 is added to the result (subtraction is performed by adding the twos complement). In terms of the original operands, the subtraction $A - B$ produces a carry if $A \geq B$.

The arithmetic and logical class includes eight functions: five arithmetic, three logical. In the following descriptions, ACS and ACD refer to the source and destination accumulators.

COM Complement. Place the (logical) complement of the word from ACS in ACD.

NEG Negate. Place the twos complement of the number from ACS into ACD. Complement Carry if ACS contains zero. (Forming the twos complement of zero generates a carry, because complementing zero produces a word containing all 1s, and adding 1 to that produces all zeros again, plus a carry.)

MOV Move. Place the contents of ACS in ACD.

INC Increment. Add 1 to the number from ACS and place it in ACD. If the result is 2^{16}, complement Carry.

ADC Add Complement. Add the (logical) complement of the number from ACS to the number from ACD and place the result in ACD. If the original ACD>ACS, complement Carry.

SUB Subtract. Add the twos complement of the number from ACS to the number from ACD and place the result in ACD. If the original ACD≥ACS, complement Carry.

ADD Add. Add the number from ACS to the number from ACD and place the result in ACD. If the result is $\geq 2^{16}$, complement Carry.

AND And. Place the logical AND function of the word from ACS and the word from ACD in ACD.

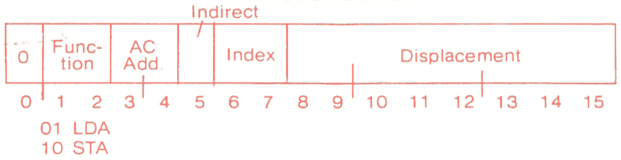
These are the basic forms of the eight arithmetic and logical instructions in which the result is loaded, there is neither shifting nor skipping, and the present state of the Carry flag is used as a base value for carry generation (the Carry flag is complemented if a carry is generated by an arithmetic function, otherwise the original state is retained).

By appending other symbols to the basic mnemonics, the programmer can specify zero, 1, or the complement of the current state of the Carry flag as the base value for carry generation; can shift (rotate) the 17-bit function result with carry one place to the left or right, or swap halves of the 16-bit function result; and he can inhibit the loading of the 17-bit final result into ACD and Carry. He can also test the final result, whether loaded or not, for a skip as follows:

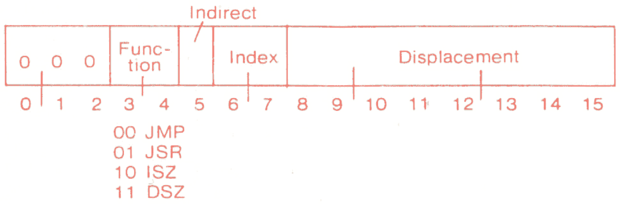
SKP Always Skip.
SZC Skip on Zero Carry.
SNC Skip on Nonzero Carry.
SZR Skip on Zero Result.
SNR Skip on Nonzero Result.
SEZ Skip if Either Carry or Result is Zero.
SBN Skip if Both Carry and Result are Nonzero.

Memory Reference Instructions There are two formats for memory reference instructions, depending on whether an accumulator is specified.

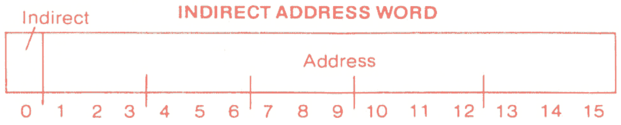
WITH ACCUMULATOR



WITHOUT ACCUMULATOR



INDIRECT ADDRESS WORD



Memory reference instructions must address a memory location. Each instruction word contains information for determining the effective address E, which is the actual address used to fetch or store the operand or alter program flow. The address information comprises an 8-bit displacement, a 2-bit index selection, and a single indirect bit. The displacement can directly address any location in four groups of 256 locations each. It can be an absolute address. That is, it may be used simply to address a location in page zero, the first 256 locations in memory. It can also be taken as a signed number to compute an absolute address by adding it to a 15-bit base address supplied by an index register. The index bits can select AC2 or AC3 as the index register; either of these accumulators can thus be used as an ordinary index register to vary the address computed from a constant displacement, or as a base register for a set of different displacements. The program can also select the program counter as the index register, so any instruction can address 256 words in its own vicinity (relative addressing).

The computed absolute (15-bit) address can be the effective address. However, the instruction can use it as an indirect address. That is, it can specify a location to be used to retrieve another address. The word read from an indirectly addressed location contains an absolute address and an indirect bit; this address can be the effective address, or it can be another indirect address.

The program can make use of an automatic indexing feature by indirectly addressing any memory location from 00020 to 00037 (addresses are always octal numbers). Whenever one of these locations is specified by an indirect address, the processor retrieves its contents, increments or decrements the word retrieved, writes the altered word back into memory, and uses the altered word as the new address, direct or indirect. If the word is taken from locations 00020-00027, it is incremented by 1; if taken from locations 00030-00037, it is decremented by 1.

There are three pairs of memory reference instructions. Two instructions move data between memory and the accumulators; two modify a memory location and test the result for a skip; and two allow the programmer to alter the normal program sequence by jumping to an arbitrary location. The modify-memory instructions are used to count loop iterations or successively modify a word for a series of operations. The jump instructions are especially useful for calling and returning from subroutines. In the following descriptions AC is the accumulator (if any) specified by the instruction, and E represents the effective address calculated from the address information given by the instruction.

LDA Load Accumulator. Load the contents of location E into AC.

STA Store Accumulator. Store the contents of AC in location E.

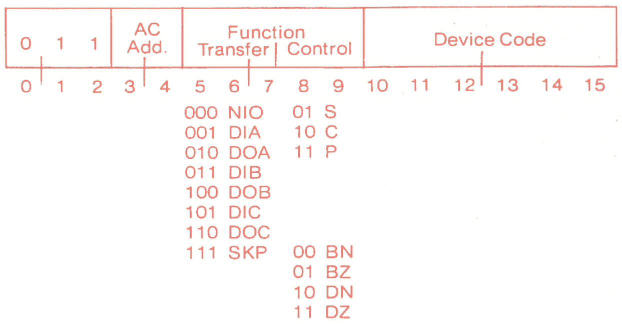
ISZ Increment and Skip if Zero. Add 1 to the contents of location E and place the result back in E. Skip the next instruction in sequence if the result is zero.

DSZ Decrement and Skip if Zero. Subtract 1 from the contents of location E and place the result back in E. Skip the next instruction in sequence if the result is zero.

JMP Jump. Load E into PC. Take the next instruction from location E and continue sequential operation from there.

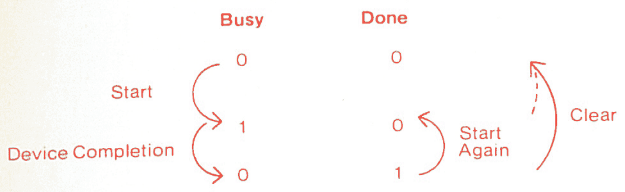
JSR Jump to Subroutine. Load an address one greater than that in PC into AC3 (hence AC3 receives the address of the location following the JSR instruction). Load E into PC. Take the next instruction from location E and continue sequential operation from there.

Input/Output Instructions The format for input/output instructions is shown below.



Input/output instructions govern all transfers of data to and from peripheral equipment, and perform various operations within the processor. An input/output instruction word has six bits for specifying the device. This format allows sixty-four device codes, of which sixty-two can be used to address devices (octal 01-76). The code 00 is not used, and 77 is used for special functions, including reading the console data switches and controlling the program interrupt.

Every device has a 6-bit device selection network, an Interrupt Disable flag, and Busy and Done flags. The selection network decodes the device code part of the instruction so that only the addressed device responds to signals sent by the processor over the I/O bus. The Busy and Done flags together denote the basic state of the device. When both are clear the device is idle. To place the device in operation, the program sets Busy. If the device will be used for output, the program must give a data-out instruction that sends the first unit of data—a word or character depending on how the device handles information. When the device has processed a unit of data, it clears Busy and sets Done to indicate that it is ready to receive new data for output, or that it has data ready for input. In the former case the program would respond with a data-out instruction to send more data; in the latter with a data-in instruction to bring in the data that is ready. If the Interrupt Disable flag is clear, the setting of Done signals the program by requesting an interrupt; if the program has set Interrupt Disable, then it must keep testing Done or Busy to determine when the device is ready.



In all input/output instructions two bits either control or sense Busy and Done. In a skip instruction, the two bits specify the flag and the state on which the skip is to occur. In a transfer instruction, these bits can be used to specify a control function to be performed in addition to the transfer. Control functions are available to start the device by clearing Done and setting Busy; to clear both Busy and Done, idling the device; and to generate a special pulse whose effect, if any, depends on the device.

The overall sequence of Busy and Done states is determined by both the program and the internal operation of the device.

The data-in or data-out instruction that the program gives in response to the setting of Done can also restart the device. When all the data has been transferred the program generally clears Done so the device neither requests further interrupts nor appears to be in use, but this is not necessary. Busy and Done both set is a meaningless situation.

With a single device code the program can address up to three registers in the device. These are referred to simply as the A, B and C buffers. For each buffer there is a pair of transfer instructions, one to move data into an accumulator from the buffer, another to move data from an accumulator out to the buffer. Thus every one of these six transfer instructions must specify a device and an accumulator, and may specify a control function as well.

DIA	Data In A
DOA	Data Out A
DIB	Data In B
DOB	Data Out B
DIC	Data In C
DOC	Data Out C

The amount of data actually supplied or accepted by the device depends on the size of its buffer, its mode of operation, and so forth. The remaining input/output instructions specify a device and either a function or a skip condition.

NIO	No IO Transfer. Perform the specified control function.
SKPBN	Skip if Busy is Nonzero.
SKPBZ	Skip if Busy is Zero.
SKPDN	Skip if Done is Nonzero.
SKPDZ	Skip if Done is Zero.

Input/output instructions with the device code 77 (CPU) perform a number of special functions rather than controlling a specific device. In a transfer instruction there may or may not actually be a transfer, but the start and clear control functions turn the interrupt on and off. The skip instructions sense the Interrupt On and Power Failure flags. In some cases the assembler recognizes a special mnemonic and the CPU device code (these are given in the second column).

NIOS CPU INTEN
Interrupt Enable.

NIOC CPU INTDS
Interrupt Disable.

DIA AC, CPU READS
Read Switches. Read the contents of the console data switches into AC.

DIB AC, CPU INTA
Interrupt Acknowledge. Place in AC the device code of the first device on the bus that is requesting an interrupt ("first" means physically closest to the processor on the bus).

DOB AC, CPU MSKO
Mask Out. Set up the Interrupt Disable flags in the devices according to the mask in AC. For this purpose each device is connected to a given data line, and its flag is set or cleared as the corresponding bit in the mask is 1 or 0.

DIC O, CPU
Clear IO Devices. Clear the control flip-flops, including Busy, Done and Interrupt Disable, in all devices connected to the the Bus.

DICC O, CPU IORST
IO Reset. Clear all IO devices and disable the interrupt.

DOC O, CPU HALT
Halt the Processor.

SKPBN CPU
Skip if Interrupt On is Nonzero.

SKPBZ CPU
Skip if Interrupt On is Zero.

SKPDN CPU
Skip if Power Failure is Nonzero.

SKPDZ CPU
Skip if Power Failure is Zero.

Multiply-Divide Instructions Hardware multiply-divide options are available for all Nova-computers.

MUL Multiply. Multiply the unsigned integers in AC1 and AC2 to generate a double length product; add the product to the unsigned integer in AC0, and place the high and low order parts of the result respectively in AC0 and AC1 (in other words the result left in AC0 and AC1 is AC0+AC1×AC2).

DIV Divide. If the unsigned integer in AC0 is greater than or equal to the unsigned integer in AC2, set Carry and go immediately to the next instruction without affecting the original contents of the accumulators. Otherwise Divide the double length unsigned integer in AC0 and AC1 by the unsigned integer in AC2, producing a single length quotient including leading zeros, and then clear carry. Place the quotient in AC1 and the remainder in AC0.

DATA GENERAL CORPORATION

Southboro, Massachusetts 01772, (617) 485-9100,
TWX (710) 390-0309, TLX 94-8460
Arizona, Phoenix, AZ 85017, (602) 264-3821,
RWX (910) 951-1538
California, El Segundo, CA 90245, (213) 973-0401,
TWX (910) 325-6220
Palo Alto, CA 94303, (415) 965-1010,
TWX (910) 379-6484
San Diego, CA 92117, (714) 276-8450
TWX (910) 335-1211
Colorado, Denver, CO 80222, (303) 758-5080,
TWX (910) 931-0485
Connecticut, Bridgeport, CT 06610, (203) 367-3833,
Vernon, CT 06066, (203) 647-9844
Florida, Orlando, FL 32809, (305) 851-8230,
TWX (810) 850-0519
Georgia, Atlanta, GA 30329, (404) 325-3181,
TWX (810) 751-8356
Illinois, Des Plaines, IL 60018, (312) 297-6310,
TWX (910) 233-5865
Kentucky, Louisville, KY 40218
Maryland, Rockville, MD 20855, (301) 770-2550,
TWX (710) 828-0525
Michigan, Southfield, MI 48075, (313) 357-0006
Minnesota, Minneapolis, MN 55420, (612) 854-7727
Missouri, Clayton, MO 63105, (314) 726-0811
New Jersey, Saddlebrook, NJ 07662, (201) 843-0676,
TWX (710) 990-5061
New Mexico, Albuquerque, NM 87108, (505) 266-5951,
TWX (910) 989-1614
New York, Commack, Long Island, NY 11725,
(516) 864-2700, TWX (510) 226-3741
Rochester, NY 14618, (716) 385-2000
TLX (510) 253-2493
Schenectady, NY (518) 377-1300
Syracuse, NY 13211, (315) 455-1525
North Carolina, Greensboro, NC 27408, (919) 275-8586,
TWX (510) 925-1113
Ohio, Chesterland, OH 44046, (216) 729-1917,
TLX (810) 427-9230
Dayton, OH 45426, (513) 435-1932
Oklahoma, Tulsa, OK 74135, (918) 749-5763,
TWX (910) 845-2285
Pennsylvania, Blue Bell, PA 19422, (215) 643-5515,
TWX (510) 661-0709
Pittsburgh, PA 15220, (412) 922-7584
Texas, Dallas, TX 75240, (214) 233-4496,
TWX (910) 860-5538
Houston, TX 77018, (713) 688-8641
TWX (910) 881-2759
Utah, Salt Lake City, UT 84115, (810) 484-5271
Washington, Renton, WA 98055, (206) 228-5890,
TWX (910) 423-0883

INTERNATIONAL

Australia, Melbourne, (03) 511233, TLX 790-33041
Sydney, 0251-927106, TLX 790-25046
Austria, 1030 Vienna, 0222-72 43 17, 0222-72 42 33,
TLX 841-74559
Canada, Calgary, Alberta, (403) 262-7705,
TLX 038-22712
North Vancouver, British Columbia, (604) 985-9104,
TWX (610) 923-5080
Mississauga, Ontario, (416) 678-2981,
TWX (610) 492-9371
Dorval 760, Quebec, (514) 631-9076,
TWX (610) 422-3049
Hull, Quebec, (819) 770-2030, TWX (610) 564-6752,
TLX 053-3501
Costa Rica, San Jose, 228156
Denmark, DK-2600 Glostrup, 01-96 53 66,
TLX 855-15468
England, Birmingham, 021-742,3117/8
Cheshire, 061-969-3935/6/7/8/9
TLX 851-667903
Greenford, London, 01-6366447, TLX 851-24203
Finland, 00101 Helsinki 10, 550045, TLX 857-12405
France, Boulogne, Paris, 604.91.42, 842-20143
75016 Paris, 504-2344, TLX 842-61289
Hong Kong, Hong Kong, H-754495, TLX 780-3184
Israel, 67132, Tel Aviv, 03-256020, TLX 922-33484
Italy, Baranzate (Mi), 2350 33 33, TLX 843-34074
Japan, Saitama 361, 485-56-8857, TLX 781-2942528
Mexico, Mexico 20 d.f., 524-9195
Netherland, Rijswijk ZH, The Netherlands
070 99 73 96, TLX 844-33545
Scotland, Glasgow G37QF, 041-332-3205
Singapore, Singapore 11, 536122, TLX 786-21249
Spain, Barcelona, 259-90422
Madrid 20, 233 16 01, 831-22404
Sweden, Solna, 8-272880, TLX 854-10089
Switzerland, 1211 Geneva 13, 22-442940,
TLX 845-23359
Zurich
West Germany, 4 Duesseldorf, 0211-622042,
TLX 08-586335
2 Hamburg 54, 0411-562065, TLX 02-212448
7500 Karlsruhe, 0721-571096
8 Munich 22, 0811-223833, TLX 841-524079

The materials contained herein are summary in nature, subject to change, and intended for general information only. Detailed information concerning the use and operation of Data General equipment and software is available in the applicable technical manuals, or through local sales representatives.

□ Please have a salesman call on me.

□ Please add my name to your mailing list.

□ Send me more information on ______________________________.

NAME/TITLE ____

COMPANY ____

STREET ____________________ CITY ________________ STATE __________ ZIP __________

PHONE ____

FIRST
CLASS
PERMIT
No. 14
Southboro
Mass. 01772

BUSINESS REPLY MAIL

NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

Postage will be paid by:

DATA GENERAL CORPORATION

Southboro, Massachusetts 01772