

This article is Copyright © 1970 by Hewlett-Packard and is used by permission. The article was originally published in the October 1970 issue of the Hewlett-Packard Journal. If errors crept in during the scanning process, please [contact Dave Hicks](#)

More Memory For Desktop Calculators

Extended memory unit adds as many as 3472 program steps.

By Russell Sparks

ADDITIONAL MEMORY FOR THE HP MODEL 9100A/B DESKTOP CALCULATORS is provided by a new, self-contained, rack-mounted unit. By adding 17 times the programmable memory to the Model 9100A, and 8 times the programmable memory to the Model 9100B, this Model 9101A Extended Memory, [Fig. 1](#), greatly extends the range of their problem solving abilities. The calculators are still the controlling and calculating elements of the system; the extended memory is operated by Format (FMT) commands from the calculators. It has a 20,832 bit memory capable of storing up to 3472 program steps, or 248 fourteen-digit registers for data storage.

When used with other peripherals, the Model 9101A becomes the main data and program storage unit. With appropriate interface cards in the Model 2570A Coupler,¹ the calculators are able to feed data to, or accept inputs from teleprinters and other equipment.

To simplify programming, register addressing and program selection are in decimal numbers rather than binary, octal or hexadecimal systems. The programmer need not be concerned with internal addressing. He needs only to remember corresponding program numbers and their functions. Address information important to the programmer is displayed when programs are transferred. Diagnostic codes are shown whenever there is a program error related to the 9101A operation.

Basic Features

The HP 9101A extended memory is divided into 248 fourteen-digit registers (ten digits displayed, two digits overrange, and two digits exponent). The registers of the 9101A are numbered decimally from 0 through 247, and are always addressed from the 9100A/B X-register.

Programs stored in the 9101A are transferred from the 9100A/B, starting at location 00 in the 9100A and +00 in the 9100B; and stopping when an END statement is encountered in the program. To identify the program, a two-digit number (decimal number 00 through 99) is placed in the 9100A/B X-register prior to the transfer. To assist the programmer in making maximum use of the 9101A storage, any new program added in storage is stored starting with the first available register, incrementing registers sequentially (14 program steps per register as in the 9100A/B) until the END statement is reached. This gives maximum program packing in the core memory within a register each time.

Each program is started in a new register. The availability of registers for storage is defined by a protection flag on each register. This protection flag is automatically set when a program is stored and may be moved manually to any desired register by another format command. (A program stripped of its protection is no longer recognized as a program.) Thus, sequential storing of all programs into the 9101A results in the most efficient use of registers, with the maximum area open for data storage. The user need not be concerned with locating each program.

Location of programs is, however, often of interest for future editing, and, in the case of the last program stored, it indicates the boundary where data can be stored. The 9101A returns the last register address used in a given program to the X-register of the 9100A/B whenever a program is stored into or recalled from the 9101A Extended Memory.

Subprograms which are subject to repeated calls from another program are often useful. Since all programs are executed in the 9100A/B Calculators, this requires that the calling program be destroyed in the calculator in order to execute the subprogram and then restored into the 9100A/B after the subprogram has finished its execution. This requires that all programs, including main calling programs, be assigned a number and stored in the 9101A. A return vector is kept in the 9101A which retains the information as to what the calling program number was and remembers where to return in that program to resume executing it. Subprograms can call other subprograms if desired (up to 14 subprograms in one sequence, not including the original calling program). Unless a subprogram is used several times, it is generally more efficient to write one long program and break it into parts which do not require returning to previous parts.

The indirect arithmetic routines allow the programmer to operate on the contents of a 9101 A register, to multiply by, divide by, add to, or subtract from those contents a number which was placed in the 9100A/B Y-register. The address of the 9101 A register desired is placed in the 9100A/B X-register. The original information is retained in the 9100A/B X- and Y-registers after the operation.

Operation of the 9101A memory is from the 9100A/B Calculators by use of the (FMT) key followed by a second key that represents a command code. This second key will not be recognized as an operator, but will output a code only; e.g., FMT PI gives out octal code 56 only and does not put PI in the X-register of the calculator.

Command Codes

FMT Y -> (): Stores 9100A/B Y-register data into 9101A register addressed by the 9100A/B X-register.

FMT PI: Recalls into the 91 OOA/B X-register the data stored in the 9101A register addressed by the number which was originally placed in the 91 OOA/B X register.

FMT (+, -, x, or ÷): Takes the data stored in the 9101A register addressed by the 9100 X-register and operates on it as dictated by the FMT code (+, -, x, or ÷), using the data in the 9100A/B Y-register as the operator.

FMT FMT: Transfers the program in the 9100A/B into the 9101A starting from Step 00 in the 9100A/B (+00 in the 9100B) through the END statement in the program. The number in the X-register prior to the transfer is the number which is assigned to that program. After the transfer is complete, the address of the last register in the 9101A used to store the program is returned to the 9100A/B X-register.

FMT GO TO: Recalls from the 9101A the program identified by the number placed in the 9100A/B Xregister. The program is loaded from 00 in the 9100A/B (+00 in the 9100B) up through the END statement. After the program has been transferred, the address of the last register used in the 9101A to store the program is returned to the 9100A/B Xregister.

FMT END: Executes the return vector at the end of a sub-program to allow return of the calling program, and to return to the proper place in this program. Provision is made here to allow for the case where the calling 'program' was the operator calling it from the 9100A/B keyboard. In this case, the FMT END is recognized as an END and no further execution is required.

FMT SET FLAG: Allows the operator to set the boundary of protection where desired. The boundary is set such that the register address placed in the 9100A/B is the first unprotected register (all addresses numerically less than this are protected).

Electronics

The basic operation of the 9101A is very similar to that of the 9100A/B where two ROM's (Read Only

Memories) are used in control.² The first is a diode ROM used as a basic sequencer. The second controls the sequences involved in operations dictated by the selected FMT codes. This is a bipolar transistor ROM in twelve integrated circuit packages.

A buffer for isolating peripherals, the new HP Model 9102A Buffer, is necessary for multiple connection of calculator peripherals to the 9100A/B. A fairly large system can be assembled using the 9101A. Some systems of this kind become susceptible to noise induced in the interconnecting lines within the system. Several precautions were taken in the 9101A to prevent noise from giving false signals. Among the most important of these are careful ground circuit connections and time sampling of the FMT input signal.

Since the 9101A interfaces with the 9100A/B using -15 volt true logic, many of the circuits used are very similar to the 9100A/B. Integrated circuits have, however, been incorporated into the core memory addressing and into the ROM.

Speed is important in a system where programming requires many iterations in arriving at the solution to a problem. The 9101A has speeds compatible with the operations in the 9100A/B. Thus, any single transfer operation (data or program) has speeds comparable to the functions of the 9100A/B (ranging from a few milliseconds to about 100 milliseconds).

One of the unique techniques used to obtain information from the 9100A/B at high speeds is that of picking the digits from the 9100A/B X, Y, and Z registers using the ROLL key. In addition to high-speed data transfer, this operation also transfers the two guard digits (giving twelve significant digits instead of ten) from these registers.

Programming

Whenever large programs are encountered, an area of core memory is often assigned as data storage. In the 9100A/B/9101A system, large programs must be broken into parts, either functionally (sub-programs) or sequentially as segments of a large program. This requires that some area of memory be set aside that will not be altered by the transferring of program segments. If small amounts of data are common to these segments, then parts of the 9100A/B core memory can be assigned to this task and program segments written short enough that no one segment overlaps into the data area. This is possible since the programs are always loaded into the 9100A/B from the 9101A starting at location 00, and stopping at the END statement. If larger common storage is required then a block of addresses in the 9101A is set aside which can be addressed uniquely from any program.

The FMT commands can be given from a program stored in the 9100A/B and executed much in the same manner as any key code of the 9100A/B; when the function has been executed, the program will continue on to the next step. If any of these FMT commands are given from the 9100A/B keyboard, then execution is halted when the function is complete.

One of the problems with large programs is that of debugging the program after it has been written. The 9101A supplies the user with an error indication and a code to aid him in finding any programming problems related to 9101A operations.

File protection has been provided to prevent accidental destruction of protected programs or data. Additionally, there is a 'FILE PROTECT' switch on the front panel that gives further protection by disallowing the use of FMT SET FLAG, thus inhibiting the function which otherwise could remove protection. It is, however, still possible to store new programs, which are automatically protected, when this switch is on.

Acknowledgments

I would like to thank the members of our design team who ably assisted in the design of the Model 9101A: Hudson Grotzinger--product design; Henry Hetzel--core circuits; Roger Story--ROM circuits; Gene Zeller--power supplies; Lou Dohse for his able leadership. Al Vigil and Virgil Jensen assisted in the transition to production. I would also like to thank the technicians who worked on the project, especially Bill Thayer who spent many hours troubleshooting during the design phase.

References

(1) Gibson F. Anderson, 'A Programmable, Modular, Bidirectional Data Coupler,' Hewlett-Packard Journal, September 1970.

(2) Charles W. Near and Robert E. Watson, 'Read-only memory adds to calculator's repertoire,' Electronics. February 3, 1969, page 70.

Russ Sparka

Russ received his BSEE degree from the University of Michigan in 1963 and continued at Michigan for his MSEE which he received in 1965. He Joined the Hewlett-Packard Loveland Division in 1965 and has been engaged in various projects relating to computational equipment. He was project leader on the HP Model 9101A Extended Memory. Russ is interested in flying and likes backpacking and fishing. When weather limits these activities, Russ does woodworking.

SPECIFICATIONS

HP Model 9101A Extended Memory

DESCRIPTION

The HP 9101A is an extended memory unit offering expanded storage capacity for the HP 9100 family of calculators. When attached to any 9100A/B output connector, the 9101A provides an additional 248 registers giving greater programming and data handling flexibility. These registers are of the same configuration as the calculator registers and useable for storing data or instructions. The execution times for functions in the extended memory are comparable to the times to perform the same function in the calculator alone. A basic system configuration consisting of a 9100A/B calculator and a 9101A extended memory unit requires no interlacing unit. Additional peripherals such as the 9120A printer or 9125A plotter may be added to the basic system using a 9102A buffer unit.

COMMANDS

The 9101A is addressed by the FORMAT (FMT) instruction from the calculator followed by a second instruction to define the desired operation (FMT) (). These commands provide for indirect addressing.

- Data Storage: FMT y -> () and FMT PI are used to store and recall data respectively.
- Program Storage: FMT FMT and FMT GO TO are used to store and recall programs or subprograms. FMT END returns control from a subprogram to the calling program.
- Indirect Arithmetic: FMT +, -, x, ÷ enable operation on and modification of the contents of any of the 248 registers.
- Subprograms: FMT GO TO to call a subprogram and FMT END to terminate execution of a subprogram, returning control to the calling program.

A numeric diagnostic code appears in the 9100A/B X register when an Illegal machine operation has been specified, thus assisting the operator In rapid program debugging.

MEMORY

- Registers: There are 248 registers that may be used to store 248 constants or up to 100 programs (totaling 3,472 program steps), or a combination of constants and programs. Total of 20,832 bits of core memory.
- Protection: FMT SET FLAG protects the contents of consecutive registers specified by the user. Initial program protection is always automatic the protected region expands to include a newly stored program. Register contents remain intact when turning power on or off.

GENERAL

- Temperature: Operating range, 0-45°C.
- Weight: Net 30 lbs (13.6 kg), shipping 40.5 lbs (18.4 kg).
- Power: 115 or 230 V +/-10% (slide switch), 48 to 60 Hz, 95 V/ amps max.
- Dimension: 7 9/32 in high by 16 3/4 in wide by 19 5/16 in deep. (184.9 mm x 425.5 mm x 490.5 mm).

PURCHASE PLAN

- Purchase: HP 9101A, \$3690.00.
- Rent: \$195 per month.
- Lease : \$95 per month.

HP Model 9102A Buffer Box

DESCRIPTION

The HP 9102A buffer box is a necessary unit when using a 9100A Calculator, 9101A extended memory and one or all of the peripherals. The buffer isolates the peripherals in a system and strengthens the output signals. The buffer also allows for a longer cable to be attached to the 9101A. This optional 5 ft cable (Part No. 11162A—\$85) allows the 9101A to be placed farther than the 18 ft standard cable will allow.

PURCHASE PLANS

- Purchase: HP 9102A, \$280.00.
- Rent: \$15.00 per month.
- Lease: \$7.25 per month.
- Service Contract Available

MANUFACTURING DIVISION:

LOVELAND DIVISION
P.O. BOX 301 815 FOURTEENTH STREET S.W.
LOVELAND, COLORADO 80537



[Go back to the HP Journal library](#)



[Go back to the main exhibit hall](#)

