Operating Characteristics of the National Cash Register Company's Decimal Computer, The CRC 102-D

R. M. HAYES

THE *102-D* is a general-purpose computer using a magnetic drum for its internal memory. This drum has a storage capacity of 1,024 words of nonvolatile, permanent storage, each word containing the equivalent of ten decimal digits. Any word of the internal memory can be used for storage of either instructions or data. When used for an instruction, a word is treated as 14 octal digits; when used for data in the arithmetical operations, a word is nine binary-coded decimal digits plus a 6-bit section containing a sign bit and an overflow bit.

Built into the computer is the ability to use magnetic tape for auxiliary storage. When in use, this magnetic tape is stored in a magnetic-tape unit, the *CRC 126*. This magnetic-tape unit has some independent abilities—in particular, the ability to search independently for desired information. A single such magnetic-tape unit, with a 1,200-foot reel of tape, has a storage capacity of well over 1,000,000 decimal digits. These data are stored in blocks of 80 decimal digits, each such block identified by a so-called block address. The searching ability of the tape unit involves the independent search for a specified block address.

There is, of course, further input-output and external control equipment. In particular, the operation of the computer can be monitored and controlled from the control console, a desk with an associated Flexowriter and various switches, buttons, and lights. There is available an auxiliary high-speed photoelectric punched-paper-tape reader operating at 200 characters per second. Similarly for output, a high-speed paper-tape punch operating at 60 characters per second is available. Finally, punched-card input and output is possible involving only the addition of the necessary punched-card equipment.

The most satisfactory means of describing the operating characteristics of the machine is in terms of the command code. Each command will be considered in turn and significant features discussed. First, however, some remarks concerning general command structure should be made. The instructions to the machine are in general three address instructions. In order to use efficiently the storage capacity of a single word, each address in a command is specified by four octal digits. There are 27 distinct and different operations available, and thus the particular command specifies the operation by a 2-octal-digit code.

R. M. HAYES is with the National Cash Register Company, Hawthorne, Calif.

Arithmetic Operations

1. *Add decimally:* This operation will take two 9-decimal-digit numbers, add them, and then store the resulting 9-decimal-digit number in the designated place. Any overflow occurring will be recorded in the result.

2. *Subtract decimally:* Similar to add decimally.

3. *Multiply decimally and round:* This operation will take two 9-digit numbers, multiply them, round the result to nine significant decimal digits, and then store the result in the designated place.

4. *Multiply double length decimally:* This operation will take two 9-decimal-digit numbers, multiply them, and store the resulting 18 decimal digits in the specified two adjacent 9-decimal-digit words. Each half of the result is given the sign of the result itself.

5. *Divide decimally and round:* This operation will take two 9-decimal-digit numbers, divide one into the other, rounding the result to nine decimal places, and store the result in the designated place. The division operates as if the numbers were fractions. Any overflow occurring is indicated in the result.

6. *Divide and save remainder decimally:* This operation will take two 9-decimal-digit numbers, divide one into the other, and consider them as integers while doing so. The result is recorded as a 9-decimal-digit integer quotient and a 9-decimal-digit integer remainder.

7. *Scale factor decimally:* This operation will "normalize" a specified number by shifting it to the left the number of times necessary to bring the most significant digit up to the decimal point. The number of decimal shifts is recorded in a second word.

8. *Add binary:* Take two 36-bit numbers or instructions and add them binarily, storing the resulting 36-bit number in the designated place. Overflows are indicated in the result.

9. *Subtract binary:* Similar to add binary.

10. *Shift logical:* Takes an entire 42-bit word and shifts it, as specified, either to right or left any desired number of places. The result is stored in the designated place.

11. *Shift magnitude:* Similar to shift logical, except that only the magnitude portion, 36 bits, is affected.

Logical and Transfer Operations

1. *Extract:* Extracts designated bits or digits from a specified word and inserts them into some other specified word.

2. *Test magnitude:* Compares the absolute value or the magnitude of two designated words and, on the basis of that comparison, chooses one of two alternatives.

3. *Test algebraically:* Similar to test magnitude, except that the signs of the two designated words are also considered.

4. *Test bit:* Examines a designated word in selected bits. If ones occur in all of those bits, one choice is made; if any of those bits are not one, another choice is made.

5. *Test search:* Determines whether any tape unit is presently searching for data.

6. *Test switch:* Determines whether a specified switch on the control console has been thrown one way or another.

7. *Buffer load:* Transfers, en masse, a designated set of eight words from the main memory to an 8-word buffer register.

8. *Buffer out:* Transfers, en masse, the contents of the 8-word buffer register into a designated set of words in the main memory.

9. *Halt:* Puts the machine in a state ready to receive new instructions from operator.

Magnetic Tape

1. *Block search:* An instruction to a designated tape unit to search for a specified block address. The operation of searching is initiated by the computer and then continues independently.

2. *Read tape:* Reads from the magnetic tape into the main memory any specified number of words.

3. *Write tape:* Similarly reads from the main memory onto magnetic tape any specified number of words.

Input-Output Commands

1. *Fill:* Calls for input, from either the console Flexowriter or the high-speed photoelectric reader, of any designated number of words.

2. *Print:* Calls for output to either the console Flexowriter or the high-speed paper-tape punch of any designated number of words.

3. *Read card:* Calls for input from punched cards.

4. *Punch cards:* Calls for output to punched cards.

Certain aspects of this command listing are of some interest. The machine has been designed so that both decimal and binary operations are possible. Most arithmetic operations will, of course, be on decimal data. However, efficient use of storage space will frequently make it desirable to store decision information, identification data, and the like in a binary form. This machine deals conveniently with both types of information.

The searching operation has been increased in flexibility by allowing for a "back-up" mode. Upon command from the computer, the tape unit will back-up the magnetic tape, a block at a time. The actual operation is merely initiated by the computer and then continues independently.

The reading and writing operations with magnetic tape have been increased in flexibility by allowing any specified number of words to be read or written. This is accomplished by a single command which specifies the tape unit involved, the first word to be written, and the last word to be written; all else is automatic.

The decision operations have been increased in flexibility by use of the "test bit" command. This is extremely useful for determining the status of an item with respect to certain categorization. One particular specialization of this command is as a test for overflow.

The input-output has been increased in flexibility by allowing for high-speed paper-tape input and output. In addition, there is complete alpha-numeric input and output, and the representation used is such that programmed internal sorting of alpha-numeric data is possible.

Discussion

Milton Adams (Stanford Research Institute): Is the magnetic tape erased while backing up or is rewriting done on top of the old data?

R. M. Hayes: No, the information is not erased while the magnetic tape is backing up. The data are rewritten on top of the old data. There is no reason why the data would necessarily be erased. It might well be that you have recorded data, backed up, and want to read those data again to check the accuracy of what you have recorded. This is one way of controlling accuracy of information.

I might mention this, that when the magnetic-tape unit is at searching, the block searching speed is 90 inches per second. The reading and writing speed is 15 inches per second. The backup speed is also 15 inches per second.

Gilbert Clotar (American Optical Company): Would you give us some idea of the speed of operation of the *102-D* in terms of orders per second, such as plus, minus, times?

R. M. Hayes: The magnetic drum rotates at the rate of 40 drum revolutions per second. An add-type operation will normally take about half a drum revolution with relatively minimum access programming. If you have any bad distribution of the numbers, the access time for the data will therefore be added to that.

The multiply-type operation, such as the multiply or divide either round or double length, will take about a drum revolution and a half, or a little less than that.

William Miehle (Burroughs Corporation): Would you explain category search again?

R. M. Hayes: I would be happy to, and I will also explain the test bit commands, because they are very similar in their nature.

To give you an example which is perhaps foreign to many of you (as it was foreign to me), suppose that you are a bank and you have made loans and you want to determine whether a given loan is delinquent or not. When a payment is made you post the fact that that payment has been made. Actually you post it by cancelling out a one that occurs there and putting a zero. So you tell the machine at the end of the delinquency period, when you are going to send out your delinquent notices, "Search the magnetic tape for the first account which is delinquent." In other words, "Look for a one in this position where I should have erased it." If I did not erase it, then the account is delinquent. And so you specify the bit that you are searching for.

This search is actually over the first word in the block—actually the first two words in the block—so it will search down the tape until it finds an item which has a one in that position and then will halt and you are now ready to read that item in. In effect it does your decision operation for you—and independent of the computer itself.

The test bit command works in exactly the same way. You don't have to have a single bit; you can have a combination of bits and it will examine all of them simultaneously and if a one occurs in every position the tape unit will halt, ready to receive a read instruction, or with the test bit command the decision will be made one way or the other.

Mr. Casey (General Electric AGT): What have you found are the most valuable uses of the extract operation? Is it the 3-address instruction?

R. M. Hayes: As far as the uses of the extract operation are concerned, or even the most valuable ones, I am afraid that I have never evaluated the relative uses of the extract. It is certainly extremely useful in actual command flow in determining in an iterative operation, for example, where you ended the operation. It certainly is extremely necessary if you are packing information, as for example packing information on a drum. You will have to extract to obtain the relevant parts and be able to operate with them. It would be extremely hard to evaluate.

Is it a 3-address instruction? Yes. The second address contains the extractor. That is, you specify the locations that you are taking information from. The first word is the extractee, namely, the address from which you are taking information. You have 42 bits and you want to extract the fifth, sixth, and seventh, say, of a given word. You specify that word and you specify the fifth, sixth, and seventh bits. Then these are extracted into the third address into again the fifth, sixth, and seventh bit positions.

J. Belzer (Battelle Memorial Institute) and **R. J. Pfaff** (International Business Machines Corporation): How many *102-D*'s will be produced and, roughly, what would you quote as the cost of a minimum unit?

R. M. Hayes: As far as quoting the cost of the minimum unit is concerned, I will be glad to discuss that with anyone who is interested. I do not think this is the place to do that.

How many *102-D*'s will be produced? Approximately somewhere in the order of 20 will be produced.