THE IBM 701 SPEEDCODING SYSTEM*

By J. W. BACKUS International Business Machines Corp., New York, N. Y.

The IBM 701 Speedcoding System is a set of instructions which causes the 701 to behave like a three-address floating point calculator. Let us call this the Speedcoding calculator. In addition to operating in floating point, this Speedcoding calculator has extremely convenient means for getting information into the machine and for printing results; it has an extensive set of operations to make the job of programming as easy as possible. Speedcoding also provides automatic address modification, flexible tracing, convenient use of auxiliary storage, and built-in checking. The system was developed by IBM's New York Scientific Computing Service.

Since this floating point Speedcoding calculator is slower than the 701, despite the conveniences that it offers, you might ask: Why go to a lot of trouble to make an extremely fast calculator like the 701 behave like a slower one? There are two principal reasons. The first is that some computing groups are working against time, and the elapsed time for solving a problem may often be reduced by minimizing the time for programming and checking out the problem even though the running time is thereby increased.

The second and most important reason for having a Speedcoding calculator, in addition to the 701, is a matter of economy. Often, the expense of operating a computing installation is almost equally divided between machine costs and personnel cost. Furthermore, machine time spent checking out problems is frequently a very appreciable percentage of the total machine timę.

It is clear, therefore, that programming and testing cost often comprise between 50 and 75% of the total cost of operating a computing installation. Since Speedcoding reduces coding and testing time considerably, and is fairly fast in operation, it will often be the more economical way of solving a problem.

Speedcoding is an interpretive system. I have implied that Speedcoding is a threeaddress system. Actually this is not quite the case. In a floating point system data and instructions have completely different forms and are treated differently. Therefore, it was thought desirable to have separate methods of dealing with each of these two types of information. Thus, each Speedcoding instruction has two operation codes in it called OP1 and OP2. OP1 has three addresses A, B, and C, associated with it and is always an arithmetic or an inputoutput operation. OP2 has one address, D, associated with it and is always a logical operation. OP, deals with floating point numbers, OP, deals with instructions. This arrangement was also adopted because it makes efficient use of the space available for an instruction and because it often speeds up operation by reducing the number of instructions which must be interpreted.

OP1 operations consist of the usual arithmetic operations plus square root, sine, arc tangent, exponential, and logarithm. There are also orders for transferring arbitrary blocks of information between electrostatic storage and tapes, drums or printer. These input-output orders have built-in automatic checks for correct transmission. Accompanying the OP1 operation code is a code to specify that any or all of the three addresses, A, B, C, should be modified during interpretation by the contents of three associated special registers (B tubes) labeled R_A , R_B , R_C . This feature. often enables one to reduce the number of instructions in a loop by a factor of 1/2.

The OP_2 operation in an instruction is executed after the OP_1 . By means of this operation one can obtain conditional or unconditional transfer of control. One can

^{*}Presented at the meeting of the Association, Sept. 9-11, 1953.

initialize the contents of any of the R-registers or one can, in one operation, increment any or all of the R-registers and transfer control. Another OP₂ operation allows one to compare the contents of an R-register with the given D-address and skip the next instruction if they are equal. OP₂ also provides a set of operations for using a fixed point accumulator for computations with addresses and for comparing the contents of this accumulator with the D address. Finally, OP, provides a convenient means of incorporating checking in a problem if desired. This feature consists mainly of two operations, START CHECK, and END CHECK; all instructions between these two orders may be automatically repeated as a block and at the end of the second repetition two separate check sums which have been accumulated during the two cycles are compared and the instruction following the END CHECK skipped if they agree.

Instructions or data may be stored anywhere in electrostatic or auxiliary storage as single Speedcoding words. Average execution times for various Speedcoding operations are as follows:

Add:	4.2 milliseconds
Multiply:	3.5 milliseconds
Read Tape;	14 milliseconds access
	plus 1.6 milliseconds per
	word
Transfer Control:	.77 milliseconds

Electrostatic storage space available is about 700 words.

Let us follow a problem from its coded form on programming sheets and data sheets until it is checked out and ready to run. First the instructions and data are punched on decimal cards whose formats are identical to those of the sheets. If there are any data or instructions which the program requires from tapes or drums, loading control cards are punched (one for each block of information) which will cause the loading system to put this information in the proper places in auxiliary storage. The deck of binary cards for Speedcoding is placed in front of this decimal deck consisting of instruction cards, data cards, and, possibly, loading control cards, and the entire deck is put in the 701 card reader. When the load button is pressed, the information will be stored in electrostatic storage, on tapes or on drums as indicated by locations on the cards. When the last card is read, execution of the program will begin automatically.

In checking out the program, use will be made of a feature of Speedcoding which has not been mentioned yet. Each Speedcoding instruction includes a list code which may be assigned one of three possible values. Associated with each of these values is a switch on the operator's panel of the 701. During execution of a program all instructions will be printed which have list codes corresponding to switches which are on. If one has properly assigned list codes, one may then check out a problem in the following way: One begins execution of the program with all three switches on, after seeing the most repetitive portions of the program printed once or twice, one of the switches is turned off, after which only moderately repetitive parts of the program are listed. Finally, the second switch is turned off and only the least repetitive instructions are seen. If trouble is encountered in the last cycle of a much repeated loop, one can approach this point rapidly with a minimum of printing and just before reaching it one can turn on all three switches and see all details of the program. Each instruction is printed with alphabetic operation codes just as it was originally written on the programming sheet. The floating point numbers at A, B, and C, the contents of the R-registers and the address accumulator, are also printed with each instruction.

Now that we have briefly seen how Speedcoding works, you might be interested to know what our experience has been with it. At present, I believe that five 701 installa-

JOURNAL

tions are using or plan to use Speedcoding. Although many improvements and extensions to Speedcoding are possible, those who have used it report that it is actually easy to use. Just this week, in fact, one of our customers arrived at our computing center in New York with a Speedcoding problem all punched and ready to test. He had programmed the entire problem at his own office using only the Speedcoding manual. He got his results with a minimum of help from us.

We have done problems with Speedcoding which ran for 100 hours, and problems which took three minutes.

Experience has shown that many problems which might require two weeks or more to program in 701 language can be programmed in Speedcoding in a few hours. One reason for this is the small number of instructions required. For example, a matrix multiplication program requires about twelve instructions. There are only two instructions in the principal loop.

Once a problem is coded one can often have it punched, checked out on the 701, and ready to run inside of an hour or two. The speed of operation of Speedcoding makes it an economical system to use. We have solved some problems in Speedcoding for a fraction of the cost on other equipment. Speedcoding is, of course, particularly valuable for small problems and for such problems is often cheaper than 701 language programs, since it reduces the costs of programming and testing.

One other interesting fact which I learned yesterday was that one 701 installation has developed a mechanical procedure for translating their standard CPC routines into Speedcoding programs and have already used these programs quite a lot.

To summarize:

Speedcoding is a floating point threeaddress system which greatly simplifies programming, and checking out a program. Speedcoding provides convenient inputoutput operations, built-in checking, easy loading and printing. Therefore, Speedcoding reduces programming and testing expenses considerably. These expenses are often a large part of the cost of operating a computing installation. Thus Speedcoding is economical as well as convenient to use.