JAMES A. SAXON

Programming the IBM 7090:

A Self-Instructional Programmed Manual

Programming the IBM 7090:

A Self-Instructional Programmed Manual

JAMES A. SAXON

Saxon Research Corporation

PRENTICE-HALL, INC., ENGLEWOOD CLIFFS, N. J.

PRENTICE-HALL INTERNATIONAL, INC., LONDON
PRENTICE-HALL OF AUSTRALIA, PTY., LTD., SYDNEY
PRENTICE-HALL OF CANADA, LTD., TORONTO
PRENTICE-HALL OF INDIA (PRIVATE) LTD., NEW DELHI
PRENTICE-HALL OF JAPAN, INC., TOKYO

Library of Congress Catalog Card Number 63-10543

Printed in the United States of America

Current printing (last digit):
13 12 11 10 9 8 7 6 5 4

73033–C

ACKNOWLEDGEMENT

The technical assistance and constructive criticism given by Dr. George Forsythe and Mr. James Watt, both of Stanford University, and Mr. Ted Medin of General Dynamics, Astronautics, is greatly appreciated by the author.

INTRODUCTORY NOTE

This Self-Instructional Text Book is designed to perform the function of teaching you to program for the IBM 7090 computer.

There will be no formal test at any time throughout the course. You will go through it as fast or as slowly as you desire. It is recommended that study periods should not extend beyond two hours and that no more than two such (two hour) periods be utilized during any one day.

There are large numbers of problems and exercises scattered throughout the book. In every case, the correct answer is given on the back of the page. You are to work each problem in the space allotted to it in the book and then check your answer with the correct answer given. If your answer was incorrect, go back to the previous page for an additional review.

There is nothing to keep you from cheating by looking at the correct answer before you have attempted to work the problem except the realization that you will not learn to program if you do so. The fact that you have this book in front of you indicates that you want to learn to program. If this is true, then please follow all instructions to the letter. Thank you for your cooperation.

Computer manufacturers are constantly making advances and some of the limitations listed in this text will be exceeded, but as long as the 7090 or similar computers are used, the general information and programming methodology will be applicable.

GENERAL INFORMATION

Before getting into the mechanics of programming for the 7090, a certain amount of general information relating to the characteristics and operation of the machine, should be discussed.

The 7090 is a scientific computer. Although it can, and does, do other work, its major function is that of solving complex mathematical problems. Despite complex formulas, every problem can be broken down to the four basic arithmetic operations of addition, subtraction, multiplication and division. This is the method the computer uses in solving its problems. It may have to multiply a set of numbers a thousand times (or a million times), but this poses no problem as each operation is executed in a tiny fraction of a second. The computer is controlled and told what to do by human beings through the use of programs, which are interpreted and executed by the machine.

A program, is a sequence of instructions, stored internally in the machine, which tell the computer exactly what to do with the data to be processed. It must take into account every eventuality and all possibilities. Nothing can be left to chance, because the machine has no capacity for thinking. It can only do what it has been told to do by the program. For example, if an overflow occurs during an arithmetic operation and the programmer has not provided for this possibility in his program, the machine will not be able to handle it.

There are three phases in computer processing: INPUT, COMPUTATION and OUTPUT. The input phase consists of placing the instructions and data to be processed into the computer. Input may be punched cards or magnetic tape although magnetic tape is more commonly used as it is a much faster method.

The computation phase carries out the instructions. It has two functions, that of arithmetic and control. Arithmetic simply carries out those instructions that are concerned with arithmetic operations and control carries out the instructions in a specified order. Normally, the computer carries out instructions sequentially (one after the other), but the programmer may use certain control instructions which may instruct the computer to proceed to any instruction in the program.

The output phase consists of reporting the results of the computer action. This may be in printed form, on punched cards or on tape. It is most economical to produce the output on tape, then if one of the other products is desired, it may be produced off-line (detached from the computer), saving considerable machine operating time.

Tape, card and printer units are connected to the DATA CHANNEL (DC), which is connected to the Central Processing Unit of the computer. The DC allows input and output of information at the same time that computation is taking place. Channels A through H are available.

Each Channel may have up to ten tape units. A printer, card reader and punch may be attached to each Channel. All Channels may operate at the same time, but only one input/output unit per Channel may be in operation at any one time.

The printer writes at the rate of 150 lines per minute. The card reader reads cards at the rate of 250 cards per minute. The Punch can punch cards at the rate of 100 cards per minute. These are high speeds, but they can not be compared to the speed attained by magnetic tape. For this reason, tape is the most commonly used input/output device on the 7090.

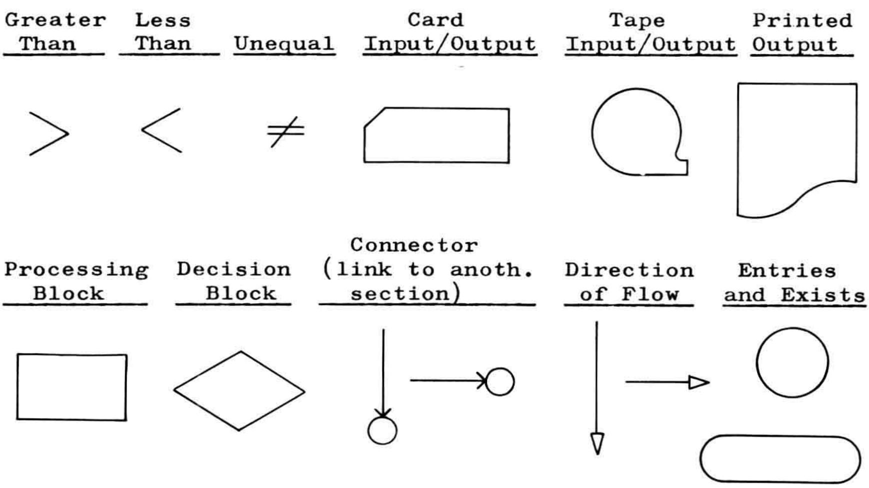
Tape may be operated on either high or low density mode. In low density, 200 characters are packed to each inch of tape. In high density, 556 characters to an inch. Tape may be run at high or low density. Using tape drive, model 729-II, tape passes at the rate of 75 inches per second and using tape drive, model 729-IV, it passes at the rate of 112½ inches per second. A normal tape is about 2400 feet long. In low density mode, about 900,000 machine words may be put on a reel of tape. In high density mode, about 2½ million words will fit on a single reel. This should effectively demonstrate the fantastic speeds attained in the input or output of information utilizing tapes.

PLANNING: After an application to be processed is selected, it must be thoroughly planned. Planning consists of the following steps:

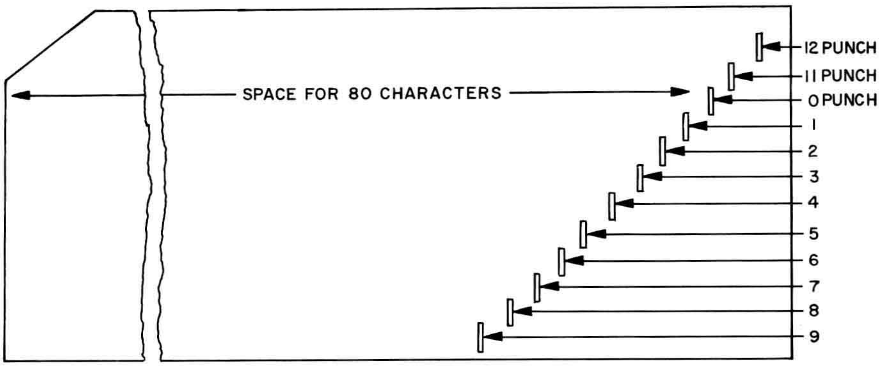
1. Analysis of the application
2. Planning and sequencing steps to be used
3. Writing the instructions
4. Determining which areas of storage will be used for various purposes

<u>FLOW CHARTING</u>: Before writing machine instructions, it is usually advisable to express the necessary steps to be taken in block diagram form. This is called flow charting. A flow chart may be quite general or very detailed, depending on the needs of the programmer. Generally speaking, the larger and more complex the problem, the more detailed the flow chart should become.

A flow chart attempts to cover all aspects of a problem. Every problem contains a multitude of detail which must be analyzed, organized and dealt with each in its own turn, with nothing left out and nothing forgotten. The flow chart is a way of accomplishing this purpose. It is also useful in making modifications and corrections to programs already written. It is advantageous to use a standardized set of symbols so that others may more easily interpret a programmers' flow chart. A few of the more commonly used signs and forms are shown below.



READING A PUNCHED CARD: It is not necessary for a fledgeling programmer to be able to read punches on a card as fluently as he reads English, but it is necessary for him to understand the code used and to be able to decypher the punches if it becomes necessary to do so. A punched card may contain up to 80 characters of information in a horizontal line and it has 12 vertical positions.



The code is as follows:

12 PUNCH - 1 PUNCH together in a column = A, 12-2=B, 12-3=C, 12-4=D, 12-5=E, 12-6=F, 12-7=G, 12-8=H, 12-9=I.

11 PUNCH - 1 PUNCH together in a column = J, 11-2=K, 11-3=L, 11-4=M, 11-5=N, 11-6=Ø, (Slash through 0 indicates it to be alphabetic), 11-7=P, 11-8=Q, 11-9=R.

0 PUNCH - 2 PUNCH together in a column = S, 0-3=T, 0-4=U, 0-5=V, 0-6=W, 0-7=X, 0-8=Y, 0-9=Z.

For numeric 1 through 9, punch only the number, omitting all three of the top columns. Special characters (i.e. comma, period) require special groupings of punches.

COMPUTER-PROGRAMMER INTERACTION: Very briefly, this is how the system works: The programmer is assigned to do a job. He analyzes, flow charts, then programs it on special programming work sheets. These work sheets go to keypunch, where cards are punched from them. This is called the source program. A special program called FAP (Fortran Assembly Program) is loaded into the computer and the source program cards are then fed into the computer. Translation of the cards into language the machine will understand is accomplished automatically by the FAP program.

The new program is then ready for operational use and may be left on cards or put on magnetic tape. When operational data is ready for processing, the program is loaded into the computer before the data is allowed to enter. When data does enter, the program takes over and processes according to the specifications of the job.

INSTRUCTIONS: Approximately one hundred instructions will be covered in detail in this course. Many instructions will not be covered since there is a limit to the size of such a course, but the most important, or useful, ones are covered and the others may be picked up from the reference manual prepared by IBM, entitled, "Reference Manual - 7090 Data Processing System."

COURSE FORMAT: Throughout the course, a small amount of information will be imparted, followed by detailed examples and problems covering the area of information just covered. You are to work the problems in the space provided on the problem page and then check your answers with the correct answers given on the following page.

Pages xiii and xiv will give you an example of how this is done. Work the problems on page xiii to see how much you have retained from your reading of pages vii through x. When you have finished, check your answers with the correct answers given on page xiv.

Each time you pick up the book, it is a good policy to review the portion already covered before starting on the new section. It is difficult to retain everything you read from one learning session to the next and this review will help you keep the knowledge already gained.


IBM
DATA PROCESSING SYSTEM

WORK AREA

Work the problems in this space, then check your answers with the correct answers given on the next page.

PROBLEMS

A. A sequence of instructions, stored internally by the computer is called a_________________.

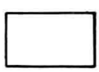
B. The three phases of computer processing are__________, _________________ and ______________________.

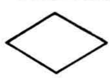
C. How many Channels are available to the 7090?_________.

D. In low density, ________ characters are packed to each inch of tape. In high density, ___________ characters are packed to an inch.

E. What is the length of a normal tape? ________________.

F. Define the following flow-charting symbols:

1.  ______________________________.

2.  ______________________________.

G. Give the alphabetic representation of the following punches in a card:

1. 12 PUNCH 4 _____	8. 0 PUNCH 8 _____
2. 0 PUNCH 4 _____	9. 11 PUNCH 9 _____
3. 11 PUNCH 4 _____	10. 12 PUNCH 1 _____
4. 0 PUNCH 2 _____	11. 11 PUNCH 2 _____
5. 12 PUNCH 6 _____	12. 0 PUNCH 9 _____
6. 12 PUNCH 9 _____	13. 12 PUNCH 2 _____
7. 11 PUNCH 1 _____	14. 11 PUNCH 8 _____

CORRECT ANSWERS

A. Program (see page vii)

B. Input, Computation and Output (see page vii)

C. 8 (see page viii)

D. 200 556 (see page viii)

E. 2400 feet (see page viii)

F. 1. Processing Block (see page ix)

2. Decision Block (see page ix)

G. (see page ix)

1. D	8. Y
2. U	9. R
3. M	10. A
4. S	11. K
5. F	12. Z
6. I	13. B
7. J	14. Q

If you have answered all of these questions correctly, turn the page and start studying Lesson 1.

TABLE OF CONTENTS

PAGE

GENERAL INFORMATION. vii

LESSON 1
Decimal, Octal and Binary Numbering Systems. . 1
Binary Arithmetic. 5
Converting Octal to Decimal. 10
Converting Decimal to Octal. 11

LESSON 2
Machine Words. 15
Registers. 17
Format of Instructions 21

LESSON 3
Fixed Point Numbers, Operations. 27
Format for Writing a Program 32
Binary Point 37
Instr: CLA, ADD (pg. 27). SUB, MPY, DVH, STO (pg. 29). LDQ, STQ, HTR (pg. 30). TZE, TOV (pg. 31).

LESSON 4
Floating Point Numbers 41
Floating Point Arithmetic. 45
Overflow and Underflow 51
Instr: FAD, FSB, FMP, FDH (pg. 46). ALS, ARS, TPL, TMI, XCA (pg. 47). NZT, ZET (pg. 52).

LESSON 5
Symbolic Coding. 55
Symbolic Coding Sheet. 59
Symbolic Language, Use of Asterisk, Plus or Minus. 63
Pseudo Op. Codes: COUNT, END, BSS (pg. 60)

LESSON 6
Additional Instructions. 69
Instr: DVP, RND, DCT (pg. 69). STZ, LLS, LRS, TRA (pg. 73). CAS, NOP (pg. 77).

LESSON 7
Use of Constants and Literals. 83
Pseudo Op. Codes: PZE, EQU, OCT (pg. 81). DEC (pg. 82).
Instr: STA, STD, STT, STP (pg. 89).

CONTENTS continued PAGE

LESSON 8
Use of Index Registers. 93
Use of Two or Three Index Registers 103
Instr: LXA, LXD, TSX, AXT (pg. 94). TIX, TXI, TXL, TXH (pg. 95)

LESSON 9
Quick Reference - Instructions and their Meanings (First Half). 109
Review and Self Test. 111

LESSON 10
Tape - Definitions. 119
Input/Output Instructions and Commands. . . . 120
Flow Chart - Read Tape Routine. 122
Flow Chart - Write Tape Routine 124
Instr: 14 I/O Instructions and Commands (pg. 120)

LESSON 11
Use of Subroutines - Subroutine Linkage . . . 133
Logical Operations (AND-OR) 137
Masking, Packing and Unpacking. 140
Instr: CAL, SLW, ANA, ANS (pg. 138). ORA, ORS, ERA (pg. 139). LGR, LGL (pg. 142).

LESSON 12
Sense Indicator Operations. 147
Sense Lights. 152
Indirect Addressing 156
Pseudo Ops: SWT (pg. 148). SLN, SLF, SLT (pg. 152).
Instr: PAI, PIA, LDI, STI, ONT, OFT, TIO, TIF (pg. 148).

LESSON 13
General Considerations. 161
Trapping. 162
Sorting 164
Program Testing 167
Instr: ETM, LTM, TTR (pg. 163).

LESSON 14
Quick Reference - Instructions and their Meanings (Complete Course) 177
Review and Self Test. 181

LESSON 15
Sample of a Complete Program. 195
Concluding Remarks. 202

INDEX 203

LESSON 1

DECIMAL, OCTAL AND BINARY NUMBERING SYSTEMS: The IBM 7090, and nearly all other large scale computers, operate with the BINARY numbering system. We are all familiar with the DECIMAL system, which utilizes 10 digits as its base, but many people are completely unfamiliar with the other two systems mentioned below. To program for the 7090, it is absolutely essential to become familiar with both BINARY and OCTAL systems.

The BINARY system is a base two system, utilizing only two digits, zero and one. This is most convenient for computers because an electrical current may be "on" or "off" and a magnetic field may be "magnetized" or "not magnetized". These are also base two types of actions. Since computers use BINARY circuits, the internal arithmetic of computers is BINARY in nature.

BINARY numbers tend to be extremely long (roughly 3.3 times longer than a DECIMAL number). For this reason, a shorthand method is used, called the OCTAL system. OCTAL, is a base eight numbering system, from zero through seven (0-7). OCTAL numbers are used when working with the 7090, but it must be remembered that the machine itself works in the BINARY system.

The similarity between OCTAL and BINARY is so close that conversion of numbers from one system to the other may be accomplished quite easily. A very complete set of tables has been designed to convert DECIMAL to OCTAL and OCTAL to DECIMAL numbers, but it is not necessary to depend on these tables as it is fairly simple to make the necessary conversion with pencil and paper. When working with the computer and large volumes of numbers, the conversion tables become very useful.

On the following pages, each of these two new numbering systems will be examined in detail including some simple arithmetic problems. For the time being, we will deal with whole numbers (integers) exclusively. Fractions and decimal fractions will not be discussed at this time. Fraction conversion tables are available in the event that need for them should arise.

BINARY NUMBERING SYSTEM: Counting in the BINARY system is as follows:

DECIMAL	BINARY	DECIMAL	BINARY
0	0	5	101
1	1	6	110
2	10	7	111
3	11	8	1000
4	100	9	1001

Since the BINARY system contains only 0 and 1, it is necessary to take the same "move" at 2, that is taken at 10 in the DECIMAL system. This is to place a "1" to the left and start again with "0". Therefore, a DECIMAL 2 is a BINARY 10, 3=11 and then another shift must be made, adding "1" to the left and starting again with "0".

For convenience, BINARY numbers are usually grouped in threes (001 010 100). Consider the BINARY position to the right as the "ones" position, then double the number for each position to the left (twos, fours, eights, etc.). By using this approach, we can determine the DECIMAL equivalent of any BINARY number.

EXAMPLE:

0	0	1	0	1	0	1	0	1
256	128	64	32	16	8	4	2	1

64 + 16 + 4 + 1 = 85

Add together all numbers that have BINARY "ones". Disregard "0".

A DECIMAL "7" is written as BINARY 111 (4 + 2 + 1 = 7)
A DECIMAL "15" is written as BINARY 001 111 (8 + 4 + 2 + 1 = 15)

Rather than referring to the three systems by name, it is more convenient to designate any number with the system being used, as follows:

DECIMAL 11 will be written 11_{10}

OCTAL 11 will be written 11_8

BINARY 11 will be written 011_2, but it is obvious by inspection if a number is written in BINARY, as it usually consists of a long series of zeros and ones.