IBM Customer Engineering

Manual of Instruction

7090 System Fundamentals

Issued to:_____

Department or Branch Office_____ Telephone Number_____

Address_____ City_____ State_____

Home Address_____ City_____ State_____

If this manual is mislaid, please notify the above address.

Printed in U.S.A.
Form 223-6860

CONTENTS

1.0.00 INTRODUCTION

BUSINESS AND SCIENTIFIC paper work becomes more complex day-by-day; there is an ever-increasing need for faster and more versatile machines to automate this work. To meet this need, IBM has produced the 7090 system.

The IBM 7090 is a transistorized computer that is six times faster than its tube predecessor, the 709. The 7090 can add 208,000 twelve-digit numbers every second. Functioning at such a speed, it can duplicate 800 years of pencil and paper calculations in only 5 minutes. In addition to being fast, the 7090 is also versatile. More than 200 different instructions may be executed by the system.

The 7090 will be used in such fields as airplane, rocket, and missile design, atomic research, weather, and missile tracking. Using the computer, calculations in these fields may be done in much more detail than ever before. High-speed computers will also eliminate much final testing; rapid, detailed calculations during design can forsee many troubles that, formerly, only testing an assembled product would indicate.

Science is not the only area in which the 7090 is advantageous. Business paper work such as payroll, billing, and sales analysis may also be automated. Management, with the aid of such a computer, can base business decisions on more current information than was ever before possible.

1.1.00 GENERAL SYSTEM OPERATION

The 7090, like a card machine, must be instructed. In a card machine, instructions and control come from the control panel, carriage tape, and program cards. In the 7090 system, the instructions and control come from a stored program.

A stored program is an ordered list of instructions which cause the computer to perform the desired calculations. There are instructions to make the computer add, subtract, multiply, and divide. There are instructions to cause the system to print, punch cards, write or read tape, and perform other logical operations. The instructions and their sequence are determined by the type of problem and the person writing the program.

Once the program is written, it is translated into machine language by the machine and stored in proper order in the systems storage device. The computer then uses these instructions to solve the problem. The computer calls for one instruction at a time from storage. The instruction causes the computer to perform a specific function. When that function has been completed, the computer signals for another instruction. Executing and calling for instructions continues until all program steps are completed.

The modern high-speed computer has 5 distinct functional parts:

1. Input
2. Storage
3. Arithmetic
4. Control
5. Output

The general interconnection of the five sections is shown in Figure 1.1-1. Note that storage is the distribution center for information to the rest of the system. The control section coordinates all the sections. System operation is similar to a 407-514 summary punch operation. The 514 should not punch a card until instructed to do so by the 407. Before the 514 punches the card, the information to be punched must be put into storage, either counters or the mechanical storage unit in the 407. The 407 cannot change the information in its storage until the 514 signals that it has punched the information in a card. Comparable conditions exist in the 7090 system so control must be maintained.

1.2.00 FUNCTIONAL PARTS OF A COMPUTER SYSTEM

1.2.01 Input

The input section of a system takes information from an external source and puts it into storage. The information may come from punched cards, magnetic tape, or manually operated keys. The information may be instructions to tell the system what to do, numbers for arithmetic calculations, or alphabetic characters for printing page headings, comments, etc.

1.2.02 Storage

The storage unit accepts and distributes information. Information coming from the input section is placed in storage. As this information is needed, it is sent to the section of the system which calls for it. Because all information is at one time or another in storage, computer speed is dependent on the speed of storage. The storage device used on the 7090 is a 2-microsecond, random access, storage unit. Any piece of information may be taken from storage or placed into storage in 2 microseconds. Random access means that the computer may directly locate a piece of information in storage without searching other locations.

1.2.03 Arithmetic

The arithmetic section of the system is the problem solving section of the computer. Generally this is the only section of the system in which information can be transformed, combined, or altered. The 7090 can perform arithmetic on instructions as well as on data. That is, instructions may be altered as determined by intermediate steps of a program. The programmer can use this feature to build a decision making or evaluating routine into his program.

1.2.04 Control

The control section directs the other sections. It tells them what to do and when to do it. Direction is initiated by instructions the control section receives from storage. The control section usually includes an instruction sequencer which locates

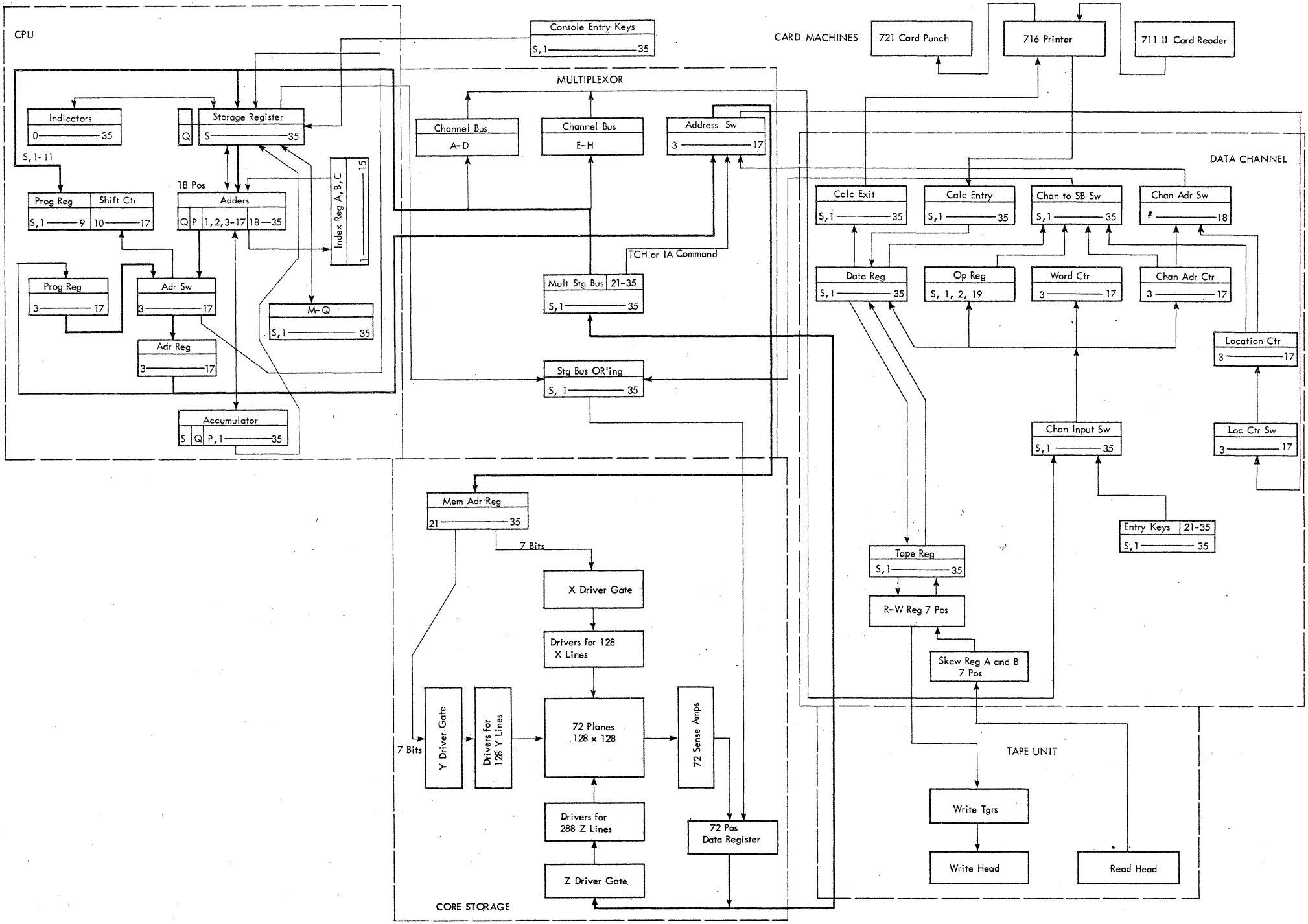

FIGURE 1.1-1. 7090 SYSTEM

the instructions to be executed. In the 7090 system this sequence may be altered at any time to transfer control to a new group of instructions. The ability to perform these transfers greatly expands the logical ability of the system.

1.2.05 Output

The output section can take information from storage and put it at the operator's disposal in such forms as magnetic tape, punched cards, printed forms, or indicator lights.

1.3.00 7090 SYSTEM MAKEUP

The 7090 system consists of the 5 sections previously described. The only variation is in the control section which is actually 2 separate controls. One section controls information flow between storage and the input-output (I-O) units; the other section controls the remainder of the system. The control is split because the I-O equipment is slower than the rest of the system. Through the divided control, I-O can operate independently from the rest of the system. This asynchronous operation allows the arithmetic and logical operations to be executed at high speed while I-O is moving information at a slower speed.

The control section allows the system to do calculations, read new information, and take results out of storage, all at the same time. Figure 1.3-1 is a block representation of the units that make up the 7090 system.

1.3.01 Central Processing Unit (CPU)

The IBM 7100 CPU is the heart of the 7090 system. All arithmetic and logical circuits, plus part of the control is located in the CPU. The CPU receives informa-

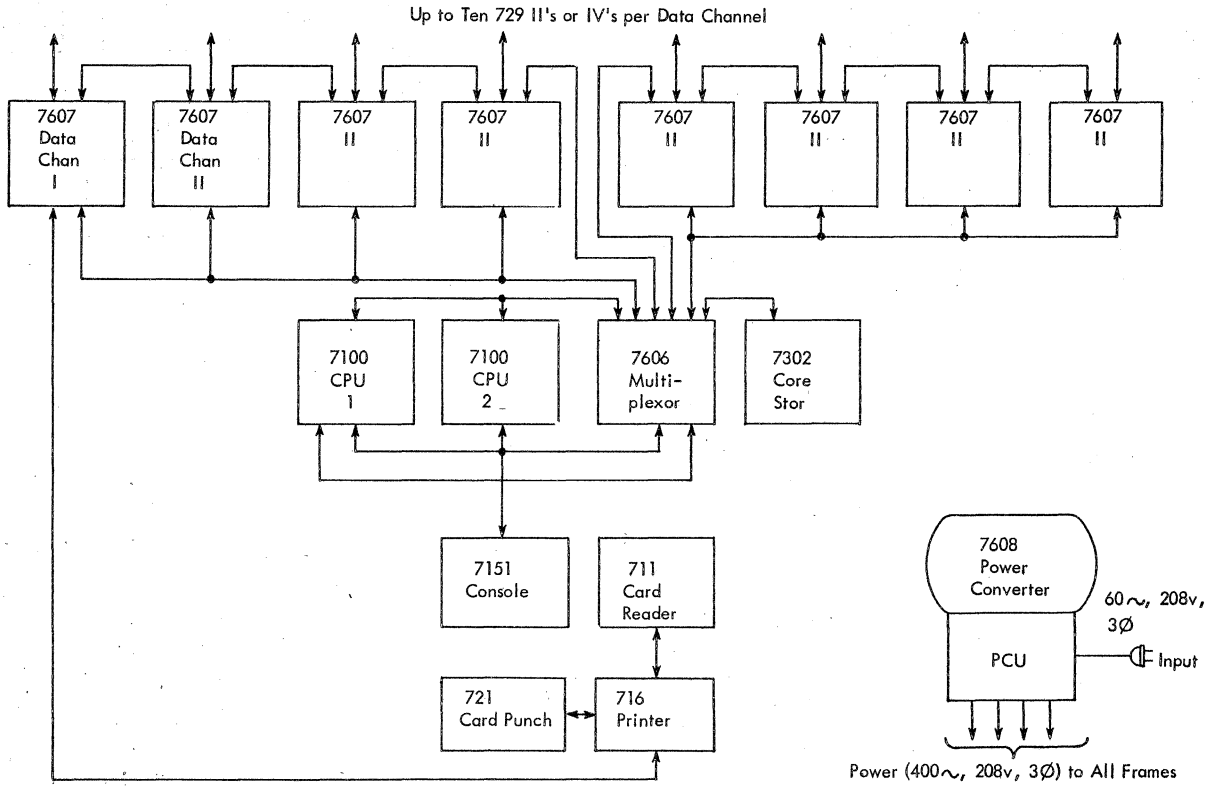

FIGURE 1.3-1. BLOCK REPRESENTATION OF 7090 SYSTEM

tion from storage, decodes it, and performs the necessary operation. Even though I-O is an independently functioning section, its operation must be initiated from the CPU.

1.3.02 Magnetic Tape Unit

The IBM 729 II and IV Magnetic Tape Units write information on magnetic tape or read information from magnetic tape. The two models, the II and IV, perform identical functions but the 729 IV is faster than the 729 II.

1.3.03 Card Reader

The IBM 711 Card Reader reads information from punched cards at 250 cards per minute.

1.3.04 Printer

The IBM 716 printer prints information from core storage at 150 lines per minute. Information may be printed with or without echo checking.

1.3.05 Card Punch

The IBM 721 Card Punch punches 100 cards per minute with information taken from storage.

1.3.06 Data Channel

The IBM 7607 I and II Data Channels control the information flow from storage to the I-O equipment. A 7607 I can control any combination of ten 729 II's or 729 IV's, and one each of the card machines. The 7607 II can control the 10 tape units but not the card machines. The 7090 system may include up to eight data channels.

Each data channel may be regarded as a separate computer. The data channel receives and decodes commands to effect information transfer between core storage and the I-O equipment.

The CPU handles instructions that select a particular channel and the I-O device on the channel. CPU then initiates the loading of channel with an I-O command. The channel uses the I-O command and any succeeding commands to move information from I-O to storage or vice versa. Once the channel has started operation, it operates independently of CPU until the I-O operation has been completed.

1.3.07 Core Storage

The IBM 7302, Core Storage Unit is a fast, random access storage. Information may be read into or read out of storage in only 2 microseconds. The storage unit is a nondestructive read out unit. That is, information may be read out of storage but still remain intact in the storage unit.

Storage serves both the CPU and data channel. The only restriction is that both cannot be using storage at exactly the same time. If data channel calls for storage while the CPU is using storage, the channel will wait until CPU has completed its storage cycle.

1.3.08 Multiplexor

The IBM 7606 Multiplexor is a time sharing and switching device. The 7606 provides a path to and from storage for the CPU and data channel. The multiplexor also performs certain "look ahead" functions to speed computer operation.

1.3.09 Console

The IBM 7151 Console provides the means to manually control the system and to investigate the contents of registers or storage. Several registers are continually displayed as indicator lamps. Other registers may be displayed by depressing certain keys. In addition to the operating features, the console also houses a customer engineer test panel and a marginal check panel.

1.4.00 7090 GENERAL LOGIC

To explain information flow in a one-channel 7090 system, the following problem is set up and developed as the 7090 would handle it:

Problem: Solve A + B = C and print C.

Given: The quantities A and B punched in a card.

The control panels for the card reader and printer must be put in place to select the proper card columns and print wheels. These panels usually are prewired to accommodate almost any job.

In addition, two commands must be in storage to instruct the channel to transmit information. One command must cause information to move from the card reader to storage; another command must move information from storage to the printer.

The 7090 will accomplish the following steps by decoding the instructions in the program:

1. Instruct the card reader to operate.
2. Put into the data channel the I-O command that will move information from the card reader to storage.
3. Instruct the CPU not to go on with step 4 until step 2 is complete.
4. Sample A in storage and move it to the CPU.
5. Sample B in storage and add it to A in the CPU.
6. Move the result, C, from CPU to storage.
7. Instruct the printer to operate.
8. Place the command in the data channel that will move C from storage to the printer.

The program to solve the above problem requires eight CPU instructions and two channel commands.

Before building the program and investigating the inner workings of the 7090, an understanding of machine language is necessary. The number system used in the 7090 is explained in Appendix A of 709-7090 Data Processing System Reference Manual (Form A22-6503).

1.5.00 7090 WORDS AND INSTRUCTIONS

A 7090 word may be a fixed-point number, a floating-point number, an instruction, or any pattern of 36 bits desired by the programmer for any reason.

Most 7090 instructions have an address part which is used to denote the location in core storage which is to be subjected to some arithmetic or logical operation. This address part or field always occupies bit positions 21 through 35 (Figure 1.5-1). The 15-bit address field is just large enough to hold the number 32,767, the highest address in core storage.

This number, expressed in binary, is simply 15 consecutive ones (Figure 1.5-2). On a 7090 with 8192 words of core storage the largest address (8191) is contained in only 13 bit positions and the contents of the two left-most positions of the address field (21 and 22) are ignored during the execution of an instruction. Similarly, a machine with 4096 words of core storage uses only bit positions 24 through 35.

The operation part of an instruction, in general, is not fixed in length but may vary from one instruction to another. Figure 1.5-3 shows the bit pattern for an addition instruction specifying core location 1.

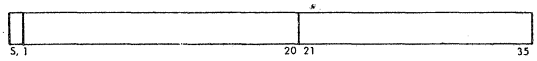

FIGURE 1.5-1. ADDRESS PART OF INSTRUCTION WORD

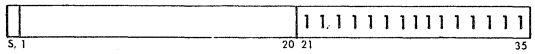

FIGURE 1.5-2. ADDRESS PART OF INSTRUCTION WORD SHOWING MAXIMUM ADDRESS

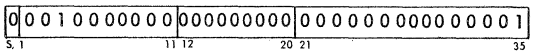

FIGURE 1.5-3. ADD INSTRUCTION

1.6.00 EXERCISES

The differences between decimal and binary number systems are explained in Appendix A of 709-7090 Data Processing System Reference Manual (Form A22-6503). To determine if you understand how to convert from one base to another, do the following conversions to see if your answers agree with those given. (Decimal and octal fractions are rounded to three places and binary fractions are rounded to nine places.

Whole Numbers

1.	5_{10}	=	5_8	=	101_2
2.	1125_{10}	=	2145_8	=	$10\ 001\ 100\ 101_2$
3.	177_8	=	$1\ 111\ 111_2$	=	127_{10}
4.	1325_8	=	$1\ 011\ 010\ 101_2$	=	725_{10}
5.	7777_8	=	$111\ 111\ 111\ 111_2$	=	4095_{10}
6.	$1\ 101\ 010_2$	=	152_8	=	106_{10}
7.	$1\ 111\ 111_2$	=	177_8	=	127_{10}
8.	$1\ 010\ 101_2$	=	125_8	=	85_{10}
9.	4096_{10}	=	10000_8	=	$1\ 000\ 000\ 000\ 000_2$
10.	3333_{10}	=	6405_8	=	$110\ 100\ 000\ 101_2$

Fractions

1.	0.7_{10}	=	0.546_8	=	$0.101\ 100\ 101_2$
2.	0.25_{10}	=	0.2_8	=	0.01_2
3.	0.33_{10}	=	0.251_8	=	$0.010\ 101\ 001_2$
4.	0.145_{10}	=	0.112_8	=	$0.001\ 001\ 01_2$
5.	0.915_{10}	=	0.724_8	=	$0.111\ 010\ 1_2$
6.	0.4_8	=	0.1_2	=	0.5_{10}
7.	0.57_8	=	$0.101\ 111_2$	=	0.734_{10}
8.	0.715_8	=	$0.111\ 001\ 101_2$	=	0.9_{10}
9.	0.101_2	=	0.5_8	=	0.625_{10}
10.	$0.101\ 010_2$	=	0.52_8	=	0.656_{10}

Improper Fractions

1.	17.05_{10}	=	21.032_{8}	=	$10\ 001.000\ 011\ 01_{2}$
2.	40.96_{10}	=	50.753_{8}	=	$101000.111\ 101\ 011_{2}$
3.	17.05_{8}	=	$1\ 111.000\ 101_{2}$	=	15.078_{10}
4.	77.77_{8}	=	$111\ 111\ .\ 111\ 111_{2}$	=	63.984_{10}
5.	11.11_{2}	=	3.3_{8}	=	3.375_{10}
6.	10.01_{2}	=	2.2_{8}	=	2.25_{10}

2.0.00 COMPUTER OPERATIONS

TO INTRODUCE YOU to the inner workings of the 7090 system, this section follows the solution of the problem, A + B = C, as the system would handle it. The explanation traces the problem from unit to unit and from register to register within each unit.

2.1.00 REGISTERS

Some of the registers and their functions are:

Program Counter (PC)

The program counter keeps track of where in a program the computer is operating. It does so by keeping count of where in storage an instruction is coming from. This counter is always one step ahead of the address of the instruction being executed. The program counter is initially set to zero.

Memory Address Register (MAR)

The MAR decodes the address and locates the information in the core array.

Memory Data Register (MDR)

The MDR accepts information coming from the core array, CPU, or a data channel. From the MDR it may be read into the core array, or sent to the CPU or the data channel.

Storage Register (SR)

The SR routes information coming to or leaving the CPU. All such information must pass through the SR.

Program Register (PR)

The PR decodes positions S-11 of an instruction coming from storage. Positions S-11 form the operation code of the instruction.

In tracing the problem through the system, the write up will follow the system flow chart, Figure 1.1-1.

2.2.00 INFORMATION FLOW

Assume that A and B have been punched in a card and placed in the card reader. Also assume that the program necessary to solve the problem and print the results has been entered into core storage. The program will appear in core storage as follows:

Location	Instruction
00000	Select the card reader.
00001	Place a command in the data channel to read two cards and store the information in storage locations 20 and 21.
00002	Hold up CPU operation until the two cards have been read; then proceed to the next instruction.
00003	Bring A to the CPU
00004	Add A to B
00005	Save C in location 00010
00006	Select the printer.
00007	Place a command in the channel to print C.
00010	All zeros
00011	All zeros
00012	Read A and B.

To start computer operation, the machine is placed in automatic status and the start key is depressed. Depression of the start key routes the contents of the program counter, 00000, to the CPU address switches. The address switches switch the address to the CPU address register. From this register the address is sent to the multiplexor address switches and then routed to the memory address register.

The MAR decodes the address and locates it in the core array. From the array the information at the address is taken to the memory data register. From the MDR the information goes to the multiplexor storage bus and also back into the core array at the same address it was taken from. In this manner, the system has accomplished a nondestructive read out from storage.

The contents of the multiplexor storage bus are then sent to the CPU storage register. Also, positions S-11 of the word are routed to the program register. In the program register, positions S-11 are decoded as the operation code of the instruction. The decoding of the operation code brings up the control lines necessary to perform the instruction. The system now recognizes that this is a read select instruction.

To determine which I-O unit to read, positions 21-35 of the storage register are routed through the adders, positions 3-17, to the CPU address switches. The switches route the information to the shift counter. The bits in the shift counter are decoded to determine the proper I-O unit to read. The card reader is now ready to read a card.

Execution of the first instruction in our program is complete, and a signal is given to the system signifying that it is ready for the next instruction. By this time the program counter has been stepped to 1, the location of our next instruction.

The program counter, now at 00001, is again routed to the CPU address switches, through the switches to the address register, and from there to the multiplexor address switches, The address is switched to the memory address register where it is decoded. Address 00001 is located in the array and the information at that address is brought out to the MAR. From the MAR the information goes to the multiplexor storage bus and is also read back into storage at location 00001. The multiplexor recognizes this as a load channel instruction and routes positions 21-35 to the channel location counter. Also, the entire word S-35 is routed to the storage register. Positions S-11 go to the program register to be decoded.

On decoding the instruction the system is signaled that this is an instruction to place a command in the data channel. The system must determine where to get the command. This information is in the address portion of the instruction, positions 21-35 of the SR. The address, 00012, is the location of the first channel command.

The address is now routed from the SR (21-35) to the adders (3-17) which, besides performing addition for the CPU, also route information to different registers in the CPU. Adder positions 3-17 are routed through the address switches to the CPU address register, then to the multiplexor address switches. The address is sent to the memory address register. Next the MAR will be decoded to locate the command in the core array. When the command is located, it will be brought out to the memory data register. From the MDR it will be read back into the core array and also taken to the multiplexor storage bus.

The multiplexor storage bus is now routed through the channel bus A-D. (We will consider all I-O operation to take place on channel A.) The channel bus routes the command to the channel input switches. The command must signal the channel that this is a read operation, signal when to stop reading, and tell the channel where to store the information read.

Positions S, 1, 2, and 19 are sent to the channel operation register. These positions form the operation code of the command. The bits are decoded in the operation register and, in this instance, signal the channel that it is a read operation.

Positions 3-17 of the command are sent to the word counter. These positions will not be used in this particular command. In a command that specifies a given number of words to be read or written, this counter is used to signal the system that the given number has been read or written.

Positions 21-35 of the command are sent to the channel address switches. This is the address where the system will store the information read. If several words were to be read they would be stored in sequential locations starting at this address.

The channel now contains all of the information necessary to read a card. First, A will be sent from the card reader to the printer and then to calc entry. From calc entry, A will be routed to the data register and then to the "channel to storage bus switches." The channel address counter will be routed to the channel address switches and then to the multiplexor address switches. From the multiplexor the address is sent to the memory address register to set up the address at which A will be stored.

The channel to storage bus switches are sent to storage bus OR'ing and then to the memory data register. From the MDR the information is stored in the address set up in the memory address register.

After the channel address counter was routed to core to set up the address for A, the address counter was advanced one. Now the process can be repeated to store B, and B will be stored in the next sequential location from A.

While the channel was decoding the command and preparing to read the card, the CPU program counter, which had been stepped and is setting at 00002, was taken to the memory address register. The address was located and the information routed to the CPU storage register. The information at location 00002 is an instruction to signal the CPU to hold up operation until A and B have been read into storage.

As soon as a signal is received that A and B have been read in, the channel will go out of operation and CPU will once again begin operation.

The instruction that was holding up CPU operation stipulated that upon resuming operation the next sequential instruction would be used. Therefore, 00003 is now routed from the program counter to the CPU address switches. From the address switches the information is once again routed to the memory address register. Location 00003 is placed in the memory data register and routed to the CPU storage register and to the program register. In the PR the instruction is decoded as a clear and add instruction. A clear and add instruction resets the contents of the accumulator and places in it the information located at the specified address. The accumulator accumulates the results of our additions and subtractions.

The information to be placed in the accumulator is located at the address specified in positions 21-35 of the instruction. To locate this address, 21-35 of the storage register, which contains the instruction, are routed through the adders to the address switches and then to the memory address register, along the same path we have been using.

The information termed A is now sent from the memory data register to the storage register in the CPU. From the SR the information goes through the adders and is set in the accumulator.

During the latter part of the clear and add instruction the program counter was advanced to 00004. On receiving a signal that the clear and add were completed, the PC is once again routed to the memory address register and a new instruction is sent to the CPU.

In the CPU the new instruction will be decoded as an add instruction. The information to be added will be added to the data in the accumulator. The data to be added are located at the address specified in positions 21-35 of the instruction. To locate this data, positions 21-35 of the SR are again routed to the MAR. The data are located and sent to the SR in the CPU.

The SR is then routed to the adders and, at the same time, the accumulator is routed to the adders. The two pieces of data are combined in the adders and the results are routed back to the accumulator. The accumulator now contains A + B.

The PC, having been stepped to 00005, is again routed to the MAR. The next instruction is located and sent to the CPU. In the CPU it is decoded as a store operation. The store operation will place the contents of the accumulator in the address specified in positions 21-35 of the instruction.

Positions 21-35 of the SR are sent to the MAR to set up the address lines to perform the store operation. The accumulator is sent to the SR. From the SR the information is sent to storage bus OR'ing in the multiplexor. From storage bus OR'ing it is routed to the MDR. From the MDR it is read into the core array at the address previously set up by the MAR.

On completing the store operation, the PC, now setting at 00006, will be sent to the MAR to locate the next instruction. The instruction is sent to the CPU and decoded as a write select instruction. To determine which I-O unit to select, positions 21-35 of the SR are routed through the adders to the address switches and then to the shift counter. The outputs of the shift counter are decoded to determine that the printer is the I-O unit to be selected. The printer is then prepared to print.

The next instruction decoded by the CPU is an instruction that places a command in the data channel to instruct the channel where in storage the word to be printed may be found.

The command, once located in storage, is taken to the multiplexor storage bus. From the multiplexor storage bus it is routed through the channel A-D bus and to the channel input switches. From the switches the information is sent to the operation register, word counter, and channel address counter.

The operation register signals that it is a write operation. The word counter signals that the system is to write one word. The channel address counter specifies the address this word is to come from.

To locate the word, the channel address counter is sent to the channel address switches and from there to the multiplexor address switches. The multiplexor address switches are then sent to the memory address register and the address is located in the core array. From the array the information is placed in the MDR and from there, routed through the channel A-D bus to the channel input switches. From the channel input switches, the information goes to the data register. From the data register the information is routed through the calc exit to the printer. After the information is printed, the computer stops and the program has been completed.