INTRODUCTION

Generally stated, programming is the process by which problems are put into a form which a computer can handle. Since the computer can only calculate numerical answers to numerical problems, the programmer has to formulate all problems in this form and replace non-numerical problems with equivalent numerical ones.

Calculations on numerically-stated problems involve the use of basic arithmetic operations; i. e . , addition, subtraction, etc. These operations are initiated by a set of commands which are easily remembered as they bear a mnemonic relationship to familiar operations, such as "A" signifying Add, "D" Divide, etc. In all, the LGP-21 responds to 23 basic commands or orders concerning arithmetic, logical, manipulative, and input/output operations.

It was mentioned before that the LGP-21 memory disc has 4096 sectors in which information may be stored. The information unit which is stored in a sector is called a computer word and may consist of data or an instruction (Figure 2.1).

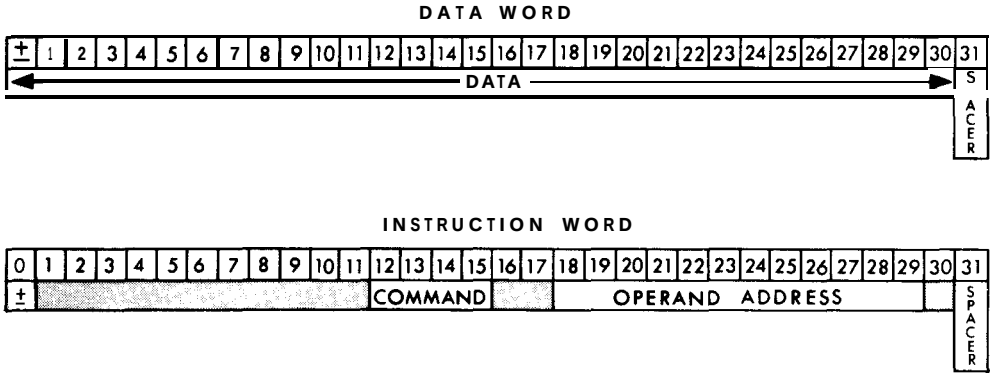


FIGURE 2.1 Word Structure

An LGP-21 word consists of 32 binary digits or "bits" which are used to represent decimal numbers or alphabetic symbols as combinations of l's and 0's. The presence of a bit in a computer word is represented as a 1, the absence of a value as 0. Since the computer performs all internal information manipulation in binary form, the programmer must acquire some familiarity with binary arithmetic. However, this chapter will be concerned only with the decimal representation of instruction words, and with a basic understanding of their functions. The binary number system will be discussed in Chapter 4.

LGP-21 INSTRUCTIONS

The programmer uses an instruction to tell the computer what operation it must perform. Each instruction is composed of two significant parts which identify the instruction as such to the computer: the command part which specifies the type of operation (add, multiply, etc.), and the address of the operand in track-and-sector notation.

Each command is assigned one of sixteen alphabetic characters. A few of these characters are used to represent two different functions. As a means of distinguishing between these alternate functions, the character is preceded by a minus sign for one operation, and by no sign for the other. When such an instruction is entered into the computer, however, it is recorded as follows: a minus instruction is recorded with a 1 in bit position zero; a non-minus instruction is recorded with a 0 in bit position zero of the word.

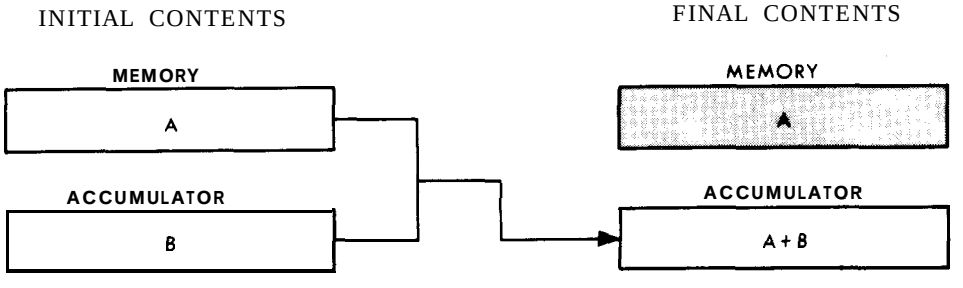
The two significant parts of an instruction word are recorded in the following positions: the command in bit positions 12 through 15, the operand address in bit positions 18 through 29. The operand address, furthermore, consists of a track address (bits 18 through 23), and the sector address (bits 24 through 29). For instructions whose command portion calls for a transfer of data, the operand address specifies the memory location from which the data is brought to the Accumulator, or in which the data is stored from the Accumulator. For arithmetic operations this address specifies the memory location of the second operand (the first operand must be in the Accumulator).

A typical instruction would be A 1532; that is, the instruction to the computer to ADD the contents of Location 1532 to whatever is in the Accumulator at the time the instruction is issued. This would be stored as follows: a 0 bit in the zero position of the computer word, to indicate that this is not a negative instruction; the binary equivalent of "A" (add) in bit positions 12 through 15; and the binary equivalent of the address 1532 in positions 18 through 29 of the instruction word. The unfilled bit positions 1 through 11, 16 and 17, and 30 and 31 are ignored by the computer when it executes an instruction. Actually, bit position 31 is always recorded in memory as a "0" as it serves to separate computer words. It is called the "spacer bit".

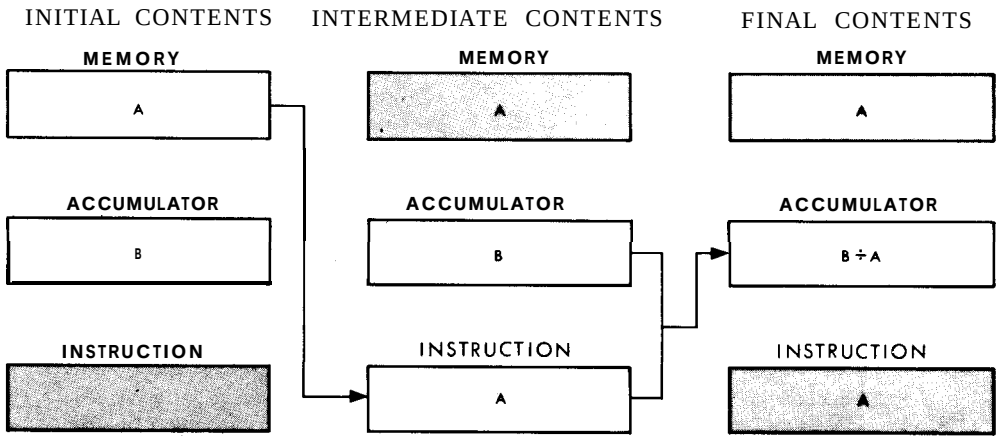
The four groups of instructions — arithmetic, logical, manipulative, and input/output — are summarized below in the following manner: the first column, headed "Order, " gives the alphabetic designation of the command; the "Address" column contains an "m" or "n", where "m" represents any one of the 4096 memory locations and "n" represents a value rather than an address. The "Interpretation" column explains the function of each instruction.

Arithmetic Instructions

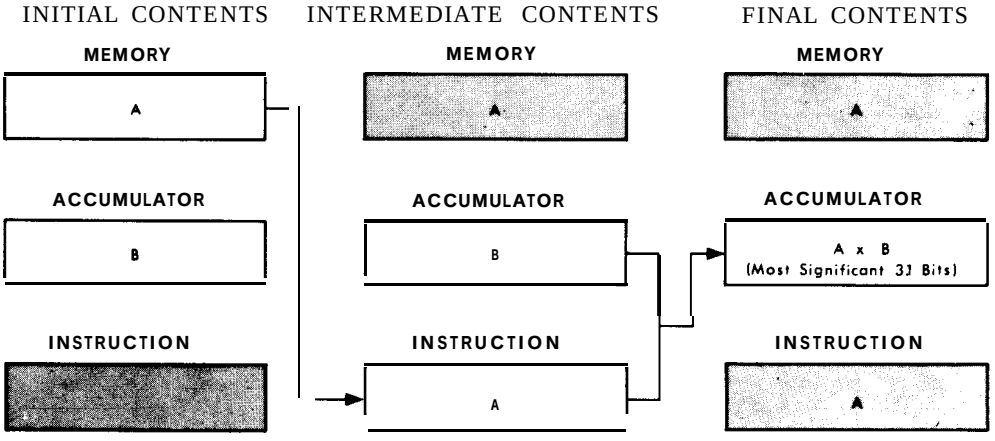
	Order	Address	Interpretation
A ADD	A	m	ADD--Add the contents of location m to the contents of the Accumulator. The sum replaces the contents of the Accumulator. If an addition results in a number beyond the limits of the Accumulator, overflow will occur. The contents of m remains unaltered.



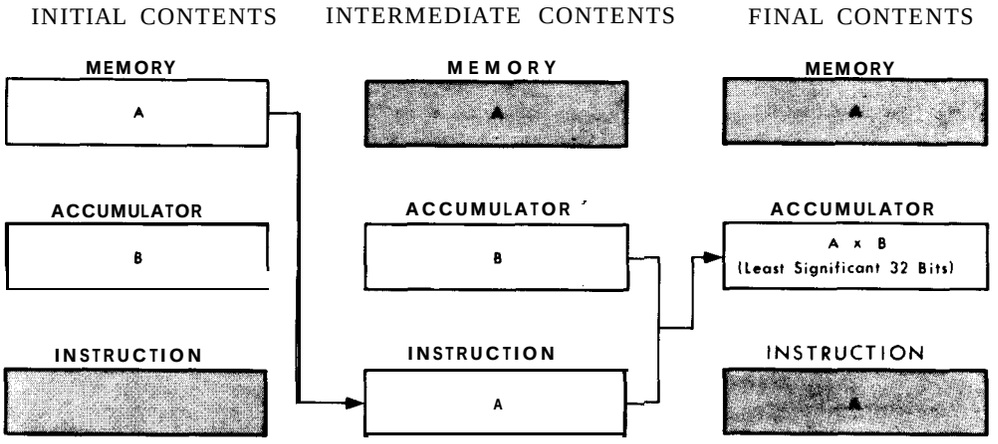
		Order	Address	Interpretation
D	**DIVIDE**	D	m	DIVIDE--Divide the number in the Accumulator by the number in location m, retaining the quotient, rounded to 30 bits, in the Accumulator. The absolute value of the contents of m must be greater than the absolute value of the contents of the Accumulator, or overflow will occur. During the divide operation the Instruction Register holds the divisor. m remains unaltered.



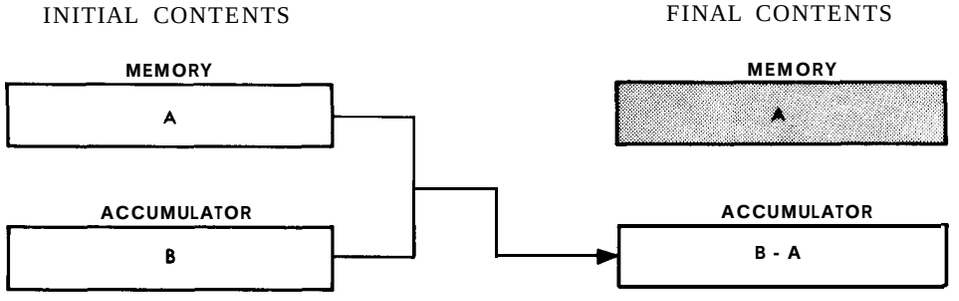
		Order	Address	Interpretation
M	**MULTIPLY**	M	m	MULTIPLY--Multiply the contents of the Accumulator by the contents of location m, forming a 62-bit product of which 31 bits are retained: the sign and the most significant 30 bits of the product replace the contents of the Accumulator. The Instruction Register holds the multiplicand during the multiply operation. Memory remains unaltered.



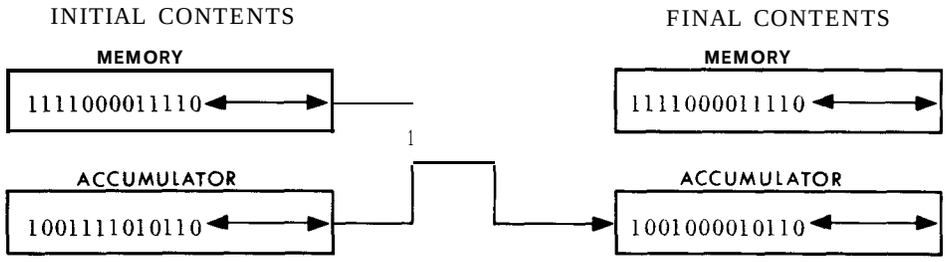
		Order	Address	Interpretation
N	**MULTIPLY**	N	m	MULTIPLY--Multiply the contents of the Accumulator by the contents of location m, forming a 62-bit product of which 31 bits are retained: the least significant 31 bits replace the contents of the Accumulator, occupying bit positions 0 through 30. Loss of any of the most significant bits does not cause overflow. During the multiply operation, the Instruction Register holds the multiplicand. Memory remains unaltered.



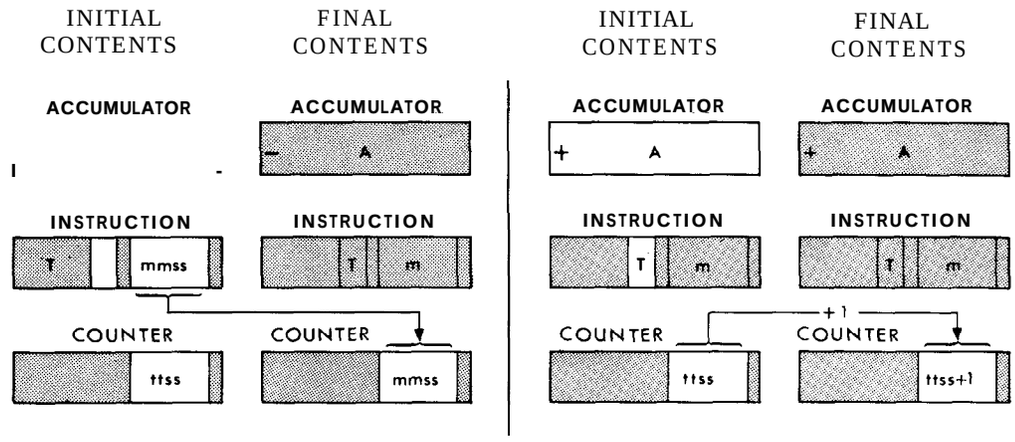
		Order	Address	Interpretation
S	**SUBTRACT**	S	m	SUBTRACT--Subtract the contents of location m from the contents of the Accumulator and retain the difference in the Accumulator. If a subtraction results in a number beyond the limits of the Accumulator, overflow will occur. Memory remains unaltered.



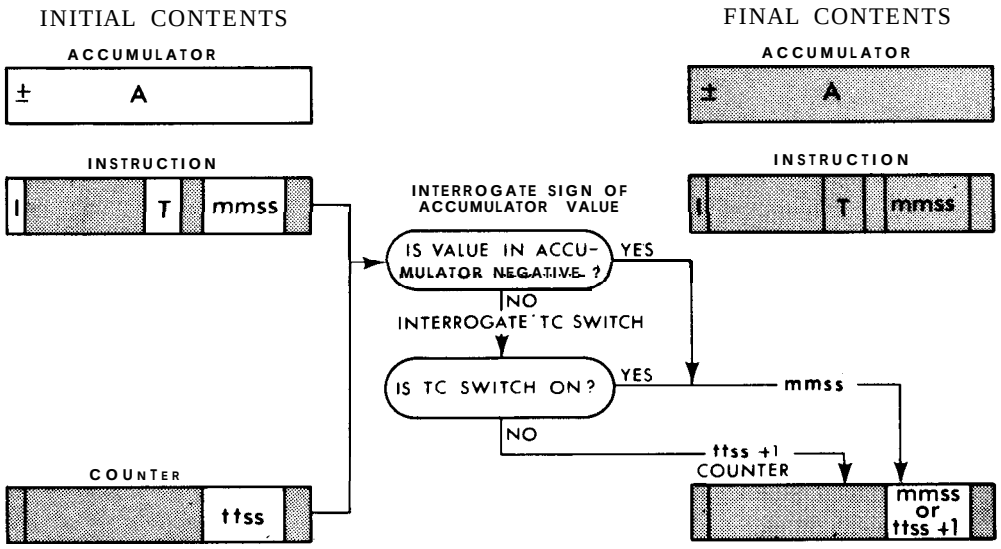
		Order	Address	Interpretation
E	**EXTRACT**	E	m	EXTRACT--Where "1" bits are in location m, retain the value of the corresponding bit positions in the Accumulator; where "0" bits are in m, place 0 bits in the corresponding positions in the Accumulator. The word in location m is called the "mask" and remains unaltered.



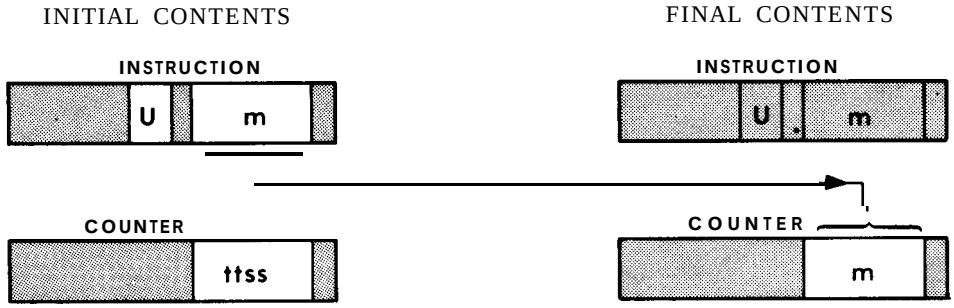
		Order	Address	Interpretation
T	**CONDITIONAL TRANSFER**	T	m	CONDITIONAL TRANSFER--If the contents of the Accumulator is negative (1 in the sign position), replace the contents of the address portion of the Counter Register with m and get the next instruction from location m. If the contents of the Accumulator is positive, continue to the next instruction in sequence without altering the Counter.



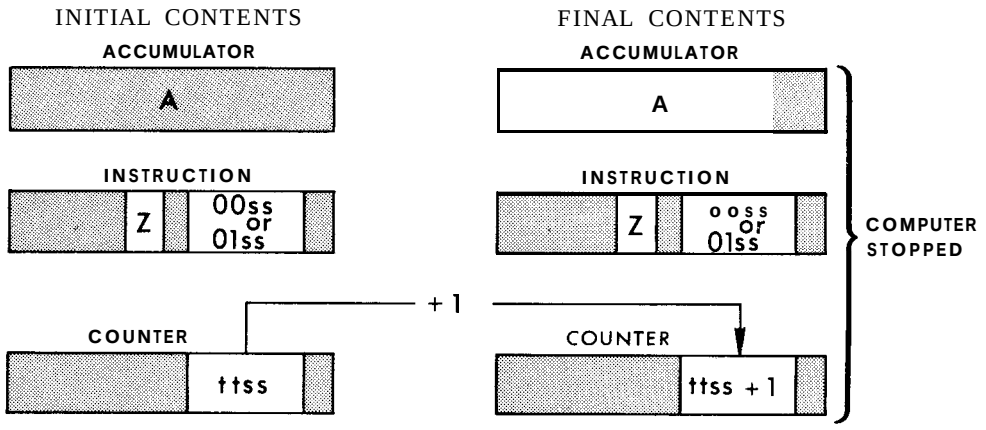
		Order	Address	Interpretation
-T	**TRANSFER CONTROL**	-T	m	TRANSFER CONTROL-If the contents of the Accumulator is negative, or if the TC switch on the console is ON, replace the contents of the address portion of the Counter Register with m and get the next instruction from location m.



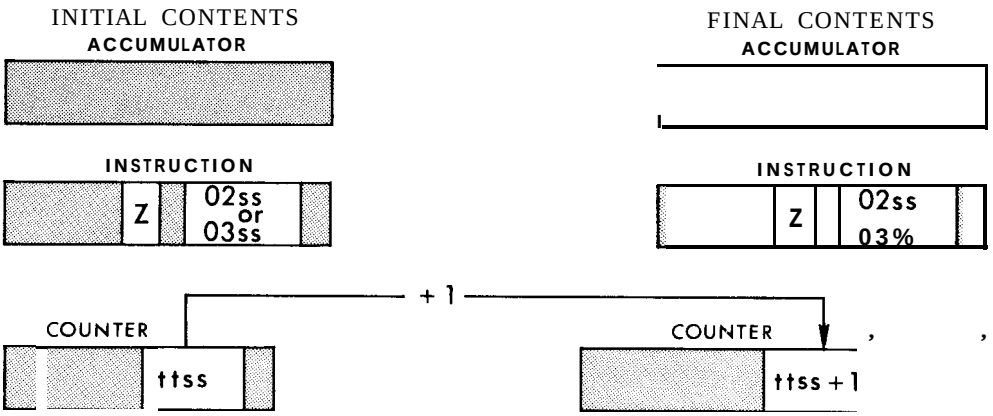
		Order	Address	Interpretation
U	**UNCONDITIONAL TRANSFER**	u	m	UNCONDITIONAL TRANSFER--Replace the contents of the address portion of the Counter Register with m and get the next instruction from location m.



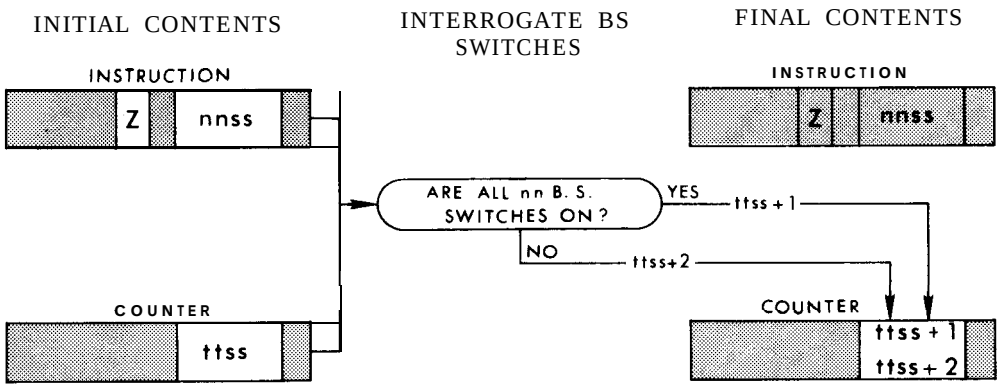
Order	Address	Interpretation
STOP	**Z**	Z n STOP--When n = 0000 or 0100, halt computation.



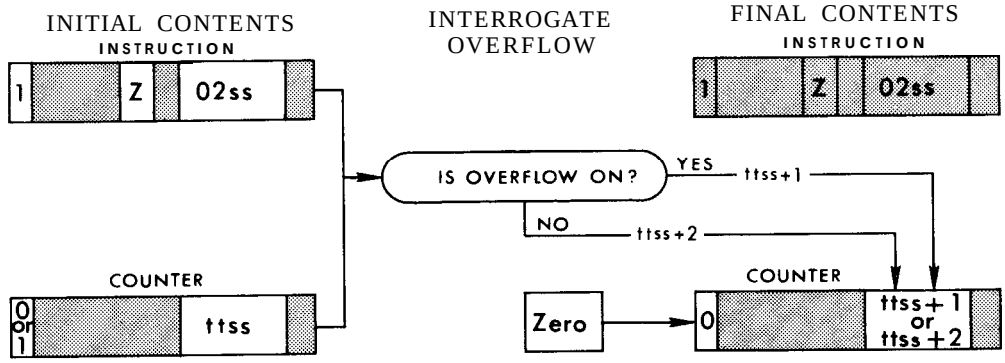
When n = 0200 or 0300, no operation occurs; i.e., the computer does not halt, the contents of the Accumulator remains unchanged, and nothing in memory is altered.



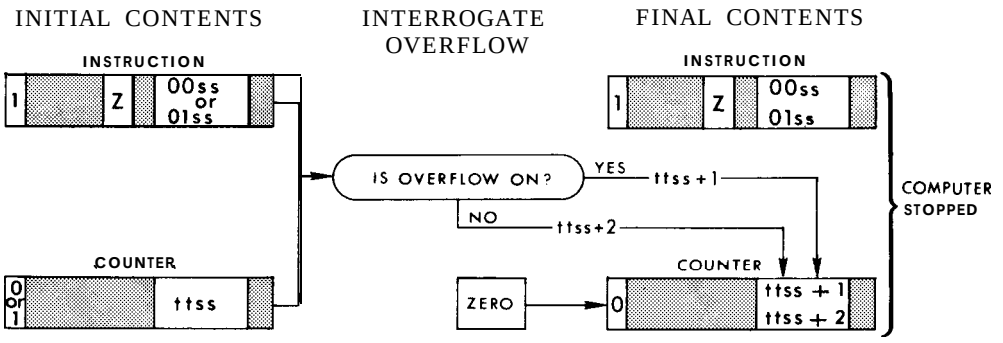
		Order	Address	Interpretation
Z	**SENSE BS AND TRANSFER**	Z	n	SENSE BS AND TRANSFER--Interrogate the Branch Switches specified by the track portion of n ($3<n\leq63$). If all of the specified Branch Switches are ON, the next sequential instruction will be executed. If any of them is OFF, the next instruction will be skipped. The Branch Switches are numbered 4, 8, 16 and 32.



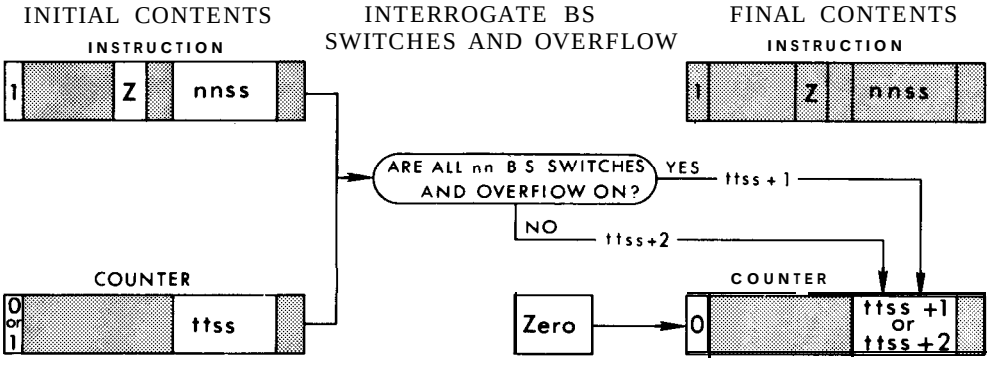
		Order	Address	Interpretation
- z	**SENSE OVERFLOW AND TRANSFER**	- Z	n	SENSE OVERFLOW AND TRANSFER--If overflow is OFF (0 in the sign position of the Counter Register), skip the next instruction in sequence. If overflow is ON (1 in the sign position of the Counter), reset the overflow bit to zero; then execute the next instruction. The track portion of n designates which, if any, Branch Switches are also to be interrogated.



If Sense Overflow is combined with Stop (-ZOOOO), the skip or no skip is deferred until after the stop. If no Branch Switches are to be tested and no stop is desired, the track address can be 02 or 03.

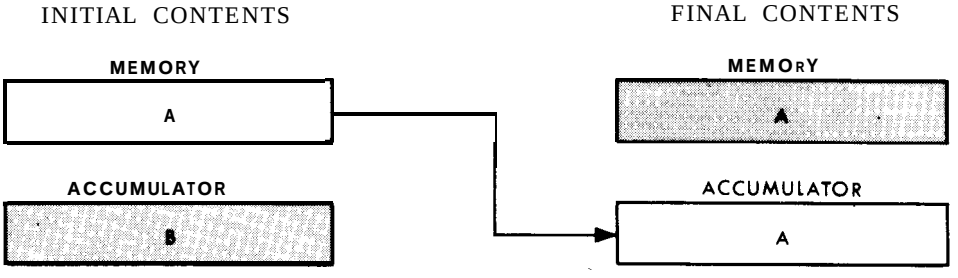


Overflow and/or any combination of Branch Switches can be interrogated with one Sense and Transfer instruction.

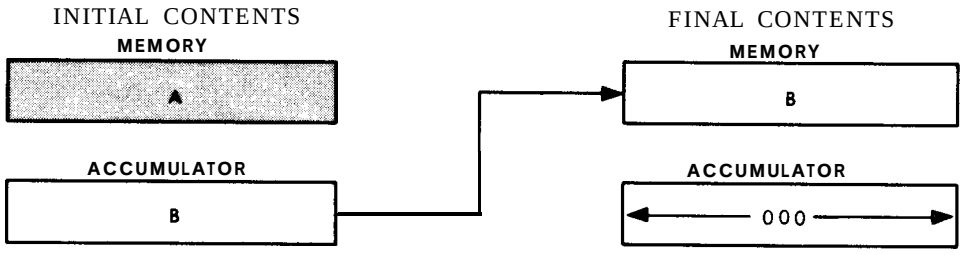


Manipulative Instructions

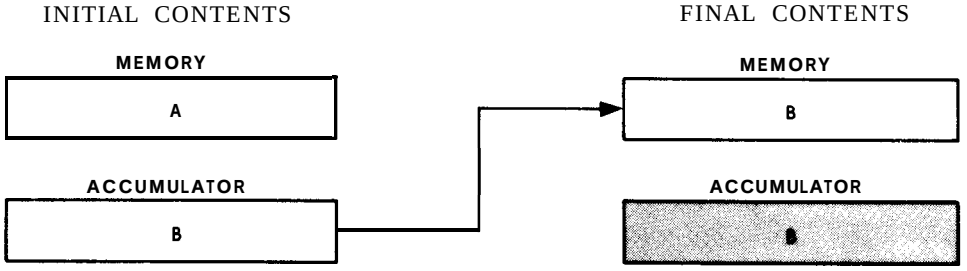
		Order	Address	Interpretation
B	BRING	B	m	BRING--Bring the contents of location m into the Accumulator, replacing its contents. Memory remains unchanged.



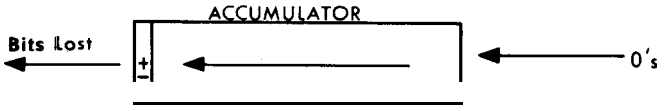
		Order	Address	Interpretation
C	**CLEAR**	C	m	CLEAR--Store the contents of the Accumulator into memory location m; then clear the Accumulator to zero.



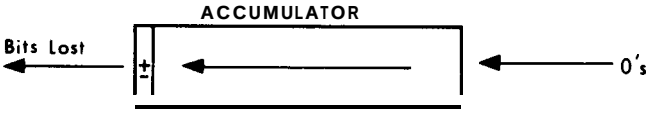
		Order	Address	Interpretation
H	**HOLD**	H	m	HOLD--Store the contents of the Accumulator into location m, without altering the contents of the Accumulator.



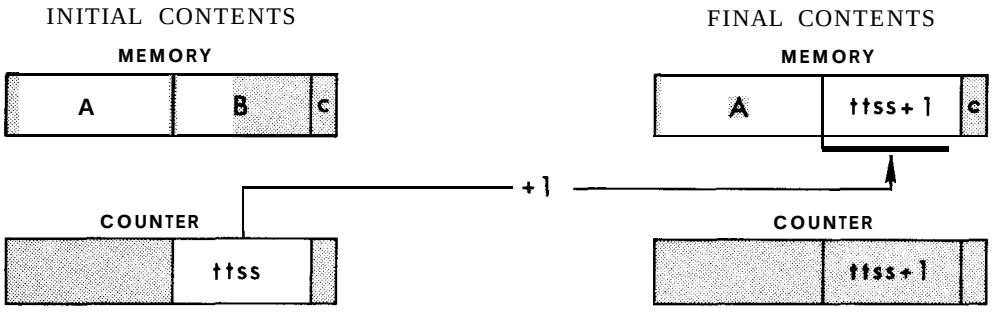
		Order	Address	Interpretation
I	**6-Bit SHIFT**	I	n	6-BIT SHIFT--When n-6200, shift the contents of the Accumulator left 6 places, inserting zeros at the right.



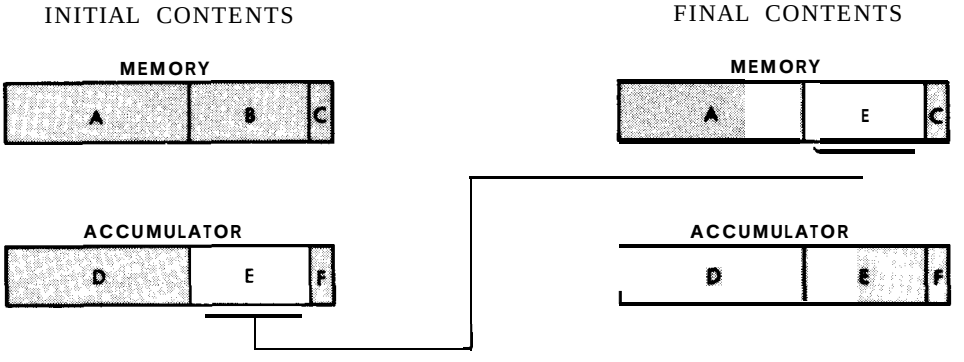
		Order	Address	Interpretation
-I	**4-Bit SHIFT**	-1	n	4-BIT SHIFT--When n=6200, shift the contents of the Accumulator left 4 places, inserting zeros at the right.



		Order	Address	Interpretation
R	**SET RETURN ADDRESS**	R	m	SET RETURN ADDRESS--In the address portion of location m, record the address which is 2 greater than the location of the I instruction being executed (i. e. , the contents of the Counter Register plus 1).

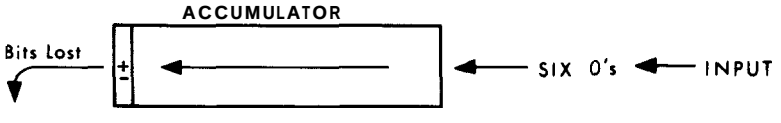


		Order	Address	Interpretation
Y	**STORE ADDRESS**	Y	m	STORE ADDRESS--Replace the address portion of the word in location m with the address portion of the word in the Accumulator, leaving the rest of m and all of the Accumulator undisturbed.

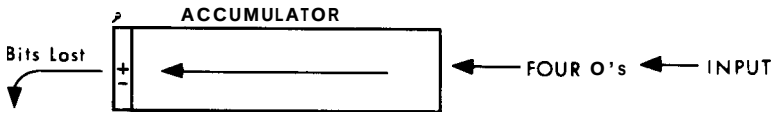


Input/Output Instructions

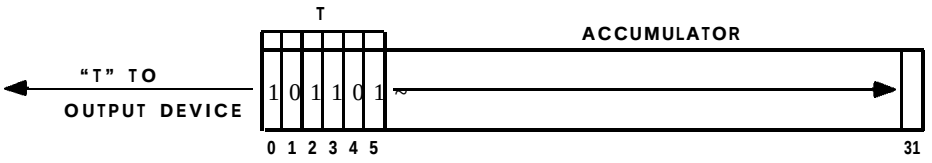
		Order	Address	Interpretation
I	**6-BIT INPUT**	I	n	6-BIT INPUT--Shift the contents of the Accumulator left 6 places, inserting zeros at the right. Then give a start read signal, allowing 6 bits of each character read by the input device specified by n to enter the Accumulator. A character enters the low-order (right) end of the Accumulator, shifting the previous contents of the register toward the high-order end. Once input is initiated, characters will be shifted into the Accumulator (and out the left end if too many are entered) until input is terminated.



		Order	Address	Interpretation
-I 4-Bit INPUT		-I	n	4-BIT INPUT--Shift the contents of the Accumulator left 4 places, inserting zeros at the right. Then give a start read signal, allowing the 4 bits of each character read by the input device specified by n to enter the Accumulator. A character enters the low-order (right) end of the Accumulator, shifting the previous contents of the register toward the high-order end. Once input is initiated, characters will be shifted into the Accumulator (and out the left end if too many are entered) until input is terminated.



		Order	Address	Interpretation
P 6-Bit PRINT		P	n	6-BIT PRINT--Transmit the character represented by bits 0 through 5 of the Accumulator to the output device specified by n. The contents of the Accumulator remains unaltered.



		Order	Address	Interpretation
-P 4-Bit PRINT		-P	n	4-BIT PRINT--Combine "1" for channel 5 and "0" for channel 6 with bits 0 through 3 from the Accumulator, then transfer this character to the output device specified by n. The contents of the Accumulator remains unaltered.