SPERRY RAND

UNIVAC® 1005

EXTENDED SYSTEM

PROGRAMMERS REFERENCE MANUAL

UNIVAC

FEDERAL SYSTEMS DIVISION

CHAPTER 1
1005 SYSTEM

		Page
I	INTRODUCTION	1/1
II	PROCESSOR	1/2
	A. Program Logic	1/2
	B. Operational Registers	1/2-3
	C. Transient Registers	1/3-4
	D. Program Control	1/4-5
	E. Core Memory	1/5-11
	1. Memory Allocation	1/9-10
	2. I/O Buffers	1/11
	3. Addressing	1/11

CHAPTER 2
SAAL ASSEMBLY SYSTEM

I	INTRODUCTION	2/1
II	GENERAL DESCRIPTION	2/1-2
III	INSTRUCTION FORMAT	2/2-5
	A. Symbolic Coding Format	2/2
	1. Label Field	2/3
	2. Operation Field	2/3
	3. Operand Field	2/4-5
	4. Comments	2/5
IV	PROGRAM ORGANIZATION	2/6-14
	A. BEG Directive	2/6
	B. CRD Directive	2/6-7
	C. PRT Directive	2/7
	D. PCH Directive	2/7-8
	E. BF1 Directive	2/8
	F. BF2 Directive	2/9
	G. BF3 Directive	2/9
	H. BF4 Directive	2/10
	I. ORG Directive	2/10-11
	J. Literals	2/11
	K. Comments Card	2/11-12
	L. STA Directive	2/12-13
	M. END Directive	2/13
V	INSTRUCTION REPERTOIRE	2/13-96
	A. Instruction Repertoire - Central Processor	2/14-62
	Load Ascending	2/15-16

Page

V INSTRUCTION REPERTOIRE (continued)

Load Descending ____ 2/17-18
Load Print ____ 2/19
Store Ascending ____ 2/20-21
Store Descending ____ 2/22-23
Store Print ____ 2/24
Shift Right ____ 2/25-26
Shift Left ____ 2/27-29
Clear ____ 2/30
Compare Alpha/Numeric ____ 2/31-32
Compare Numeric ____ 2/33-34
Increment Compare ____ 2/35
Jump Unconditional ____ 2/36
Jump Greater ____ 2/36
Jump Less ____ 2/36
Jump Equal ____ 2/36
Jump Equal Alpha/Numeric ____ 2/37
Jump Unequal Alpha/Numeric ____ 2/37
Jump Positive ____ 2/38
Jump Negative ____ 2/38
Jump Zero ____ 2/38
Jump Return ____ 2/39
Jump Exit ____ 2/40
Add to Memory ____ 2/41-42
Add to Register ____ 2/43-44
Subtract from Memory ____ 2/45-46
Subtract from Register ____ 2/47
Multiply ____ 2/48
Divide ____ 2/49
Translate ____ 2/50-55
Store With Zero Suppress ____ 2/56
Load With Sign ____ 2/57-58
Load Numerics ____ 2/59-60
Store Edited ____ 2/61
Punch Test ____ 2/62

B. Instruction Repertoire - Card System External Functions 2/62-80
Read ____ 2/63
Print-Space 1/Space 2 ____ 2/64
Print - Skip 7 ____ 2/65
Punch ____ 2/66
Read-Print-Space 1 ____ 2/67
Read-Print-Space 2 ____ 2/68
Read-Punch ____ 2/69
Read-Print-Space 1-Punch ____ 2/70
Skip 2 ____ 2/71
Skip 4 ____ 2/71

Page

V INSTRUCTION REPERTOIRE (continued)

Skip 7 ____ 2/71
Read Code Image ____ 2/72
Punch Code Image ____ 2/73
Read Auxiliary Code Image Stacker Select 1 ____ 2/74
Read Auxiliary Stacker Select 1 ____ 2/75
Read Auxiliary Stacker Select 2 ____ 2/75
Read Auxiliary Stacker Select 3 ____ 2/75
Punch with Stacker Select ____ 2/76
Read/Read Punch ____ 2/77
Read/Read Punch with Stacker Select ____ 2/78
Read/Read Punch Code Image ____ 2/79
Halt ____ 2/80

C. Instruction Repertoire - Paper Tape External Functions and Conditional Tests ____ 2/81-86

1. Paper Tape External Functions ____ 2/81-84
   Read Paper Tape 1 Frame ____ 2/82
   Read Paper Tape 80 Frames ____ 2/82
   Read Paper Tape Through Sentinel ____ 2/82
   Punch Paper Tape Without Parity 1 Frame ____ 2/83
   Punch Paper Tape Without Parity to Sentinel ____ 2/83
   Punch Paper Tape With Parity 1 Frame ____ 2/84
   Punch Paper Tape With Parity to Sentinel ____ 2/84

2. Paper Tape Conditional Tests ____ 2/85-86
   Jump Parity Error ____ 2/86
   Jump Channel 8 ____ 2/86

D. Instruction Repertoire - Magnetic Tape External Functions and Conditional Tests ____ 2/87-96

1. Magnetic Tape External Functions ____ 2/87-92
   Read Tape Servo 1 Normal Gain ____ 2/88
   Read Tape Servo 2 Normal Gain ____ 2/88
   Read Tape Servo 1 High Gain ____ 2/88
   Read Tape Servo 2 High Gain ____ 2/88
   Write Tape Servo 1 ____ 2/89
   Write Tape Servo 2 ____ 2/89
   Erase Before Write Servo 1 ____ 2/90
   Erase Before Write Servo 2 ____ 2/90
   Backspace Servo 1 ____ 2/91
   Backspace Servo 2 ____ 2/91
   Rewind Servo 1 ____ 2/92
   Rewind Servo 2 ____ 2/92
2. Magnetic Tape Conditional Tests ____ 2/93-96
   Jump Parity Error ____ 2/94
   Jump End of Tape ____ 2/94

Page

V INSTRUCTION REPERTOIRE (continued)

Example Parity Error Recovery on Read Tape Function - - - - 2/95
Example Parity Error Recovery on Write Tape Function - - - - 2/96

E. Instruction Repertoire - Advanced Programming - - - - 2/97-105
Jump Alternate Switch 3 - - - - 2/97
Jump Arithmetic Overflow - - - - 2/98
Compare Character Alpha/Numeric - - - - 2/99
Store Character - - - - 2/100
Logical And - - - - 2/101-103
Logical Or - - - - 2/103-104
Bit Shift - - - - 2/105

F. Instruction Repertoire - External Function Combinations - - - - 2/106-110

G. Instruction Repertoire - 1005 Data Line Terminal-3 External Functions and Conditional Tests - - - - 2/111-119

1. DLT-3 External Functions - - - - 2/111
2. General - - - - 2/111-112
3. Transmitting - - - - 2/112-114
4. Receiving - - - - 2/114
5. Error Conditions - - - - 2/114-115
6. Instruction Formats External Functions - - - - 2/115-117
   Send DLT 80 Characters - - - - 2/116
   Send DLT Through Sentinel - - - - 2/116
   Receive DLT To End of Message - - - - 2/117
7. Instruction Formats Conditional Tests - - - - 2/118-119
   Pause Test - - - - 2/118
   Jump End of Time - - - - 2/119
   Jump Parity Error - - - - 2/119

CHAPTER 3

UNIVAC 1005 SOFTWARE

		Page
I	THE UNIVAC 1005 SINGLE ADDRESS ASSEMBLY SYSTEM	3/1-6
	A. SAAL 1 (Illustration 1) Trial Balance Sample Program P2-4	3/1-3
	B. SAAL 2 (Illustration 2) Trial Balance Sample Program P2-4	3/3-5
	C. Trial Balance Sample Report (Illustration 3)	3/5-6
II	THE UNIVAC 1005 SINGLE ADDRESS REPORT GENERATOR	3/6-8
	A. SARGE 1 - Trial Balance Sample Program P2-4	3/7
	B. SARGE 2 (Illustration 4) Trial Balance Sample Program P2-4	3/8
III	THE UNIVAC 1005 UTILITY ROUTINES	3/8-12
	A. CONDENSE	3/8-9
	B. MEMORY DUMP (Illustration 5)	3/10
	C. READ-PRINT-PUNCH	3/10
	D. NUMBER IT	3/10
	E. DUPLICATE	3/11-12
	F. CLEAR	3/12
IV	ILLUSTRATIONS	

CHAPTER 4

UNIVAC 1005 SOFTWARE OPERATING PROCEDURES

		Page
I	ALTERNATE SWITCHES OPERATING PROCEDURES	4/1
II	SOFTWARE OPERATING PROCEDURES	4/1-6
	A. SAAL 1 -- First pass of the assembly program	4/1-2
	B. SAAL 2 -- Second pass of the assembly program	4/2-3
	C. Condense Program	4/3-4
	D. Memory Dump	4/4
	E. Read -- Print -- Punch	4/4-5
	F. Number It	4/5
	G. Duplicate	4/5-6
	H. Clear	4/6

CHAPTER 5

UNIVAC 1005 HARDWARE MACHINE TESTING and OPERATING PROCEDURES

	Page
I. MANUAL ALTERNATE SWITCHES	5-1
A. Mode of Operation	5-1
B. Automatic Form Overflow Mode	5-1
C. Trace Mode	5-2
D. Single Instructions Mode	5-2
1. Reading PAK	5-3
II. TEST SWITCH PANEL	5-4
A. Program Step Counter Switches	5-4
III. DISPLAY MASKS	5-6
A. Display Mask 4	5-6
B. Display Mask 6	5-16
C. Display Mask 8	5-19
D. Display Mask 9	5-21

CHAPTER 1

THE UNIVAC 1005 CARD PROCESSING SYSTEM

I. INTRODUCTION

The UNIVAC 1005 Card Processing system is a powerful, high performance system, which combines into a low-cost consolidated card processor features usually found only in more complex, higher priced systems. This small-scale data processing system has been designed around a single address, internally programmed processor, the UNIVAC 1005 Card Processor, and includes, as secondary units, a hardware integrated card reader, an optional, free-standing, high-speed card reader, and a free-standing card punch.

The standard card reader, which is located to the immediate right of the card processor, and which is an integral part of the hardware of the card processor, operates by means of photo-electric cells at speeds up to 600 cards per minute. The input hopper has a 1,000 card capacity, while the output stacker has a 1,500 card capacity.

The optional card reader, like the card punch, is cable connected to the central processor, and has an input hopper capacity of 1,000 cards, and an output stacker with a capacity of 1,000 cards. It features an increase in card reading speed to a maximum of 800 cards per minute.

The card processor, the central unit in the system, contains, in a single hardware unit, a high-speed printer, which prints a maximum of 132 print positions per line, and up to 600 lines of alphanumeric data per minute, the core memory, and all logic and control circuitry for the entire system. The standard configuration also includes the card reader.

The card punch is capable of punching up to 250 cards per minute, and like the free-standing card reader is cable connected to the card processor. This feature permits maximum flexibility in satisfying individual installation requirements as well as enabling maximum consideration to be given to operational preferences.

By consolidating all these components into a single, well-designed unit, the UNIVAC 1005 Card Processing System minimizes installation operational problems and maximizes supervisory and operator efficiency.

Additional detailed information on the various components available with the UNIVAC 1005 Card Processor is contained in the General Description Manual for the 1005 Card Processor.

The following section discusses the logic and control circuitries contained in the processor itself, while subsequent chapters of this manual are concerned with detailed software considerations.

II. PROCESSOR

The processor contains the systems control, arithmetic and logic circuitry, as well as core memory, and is located to the rear and left of the card reader.

The standard 6.5 microsecond core memory of 1024 characters (32 x 32 matrix plane) is expandable in increments of 1024 characters.

Complete solid-state components, ribbon cabling and wire-wrap terminals assure high operational reliability.

Logic Characteristics.

A. Program Logic

UNIVAC 1005 logic is organized around a single address fixed word logic.

B. Operational Registers.

PAK Register	The PAK Register is the Program Address Counter. This 2-character register holds the address of the instruction being executed. It occupies two memory locations. During the final execution phase of the instruction, the contents of the PAK Register are normally incremented by five to give the address of the next instruction. Certain instructions will cause the address in the PAK Register to be replaced with a new address from the instruction word, e.g., jump instructions.

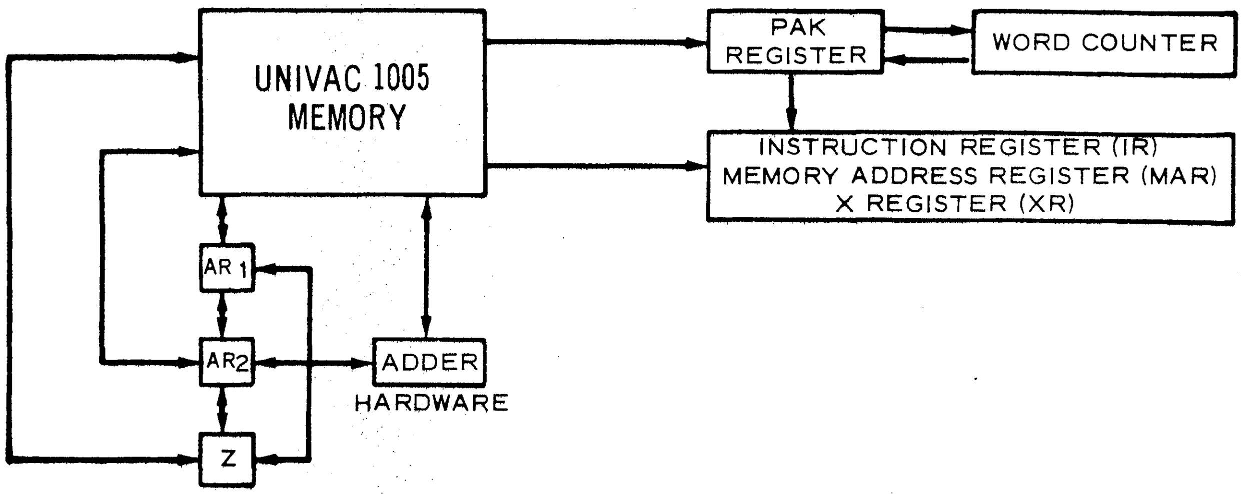


Figure 1. - Diagram of System Logic

IR Register

The IR Register is the Instruction Decoder Register. It is used to contain the operation code of the current instruction and is loaded during the instruction access cycle. The IR Register occupies one memory Location.

MAR Register

The MAR Register is the Memory Address Register. This is used to contain the address portion of the instruction. It defines the memory locations to or from which data is to be transferred. It occupies four memory locations.

C. Transient Registers.

Lengths and Uses

Two programmable transient registers are available. The registers are designated Register AR_1, Register AR_2. Register AR_1 is 10 characters in length; Register AR_2 is 21 characters in length.

Any register may be used for memory transfers. Registers 1 and 2 are the arithmetic registers. All adds, subtracts and compares are executed from these two registers. Multiply and divide operations use both arithmetic registers and the auxiliary Z register. The quotient or product is stored in registers 1

Lengths and Uses (cont'd)

and 2 (See Figure 2). Jump Return and Jump Exit operations use the auxiliary X Register.

Indicator Unit

The Indicator Unit contains the program testable indicators described below. When the indicator tested is found to be reset, the next instruction in sequence is accessed. When the indicator tested is found to be set, control is transferred to the address specified by the instruction.

1. Comparison Indicators. There are three numeric comparison indicators--greater than, less than and equal to. There are two alphanumeric comparison indicators--equal and unequal.

2. Sign Indicators. There are three sign indicators--positive, negative, zero. The contents of the arithmetic registers may be tested by the program for positive, negative or zero.

3. I/O Indicators. These additional indicators are explained in detail under their respective Input/Output Sections.

D. Program Control

The activity of the Program Control Section is divided into a series of logical machine sequences. All of these sequences are fixed in nature and occur with every instruction being processed.

Basic Machine Sequences.

(P) Program Control--Extract the program instruction address from the Program Address Counter (PAK). Store this value in the Instruction Register (IR).

(I) Instruction Access--Extract the instruction referenced by the previous P sequence. Test the operation code and generate the function signal necessary to execute instruction.

(A) Address Access--Extract the operand portion of the instruction from memory and store in the Memory Address Register (MAR).

(P+5)	Program Control Plus Five--Update the program address counter by five unless a jump instruction has been detected. In that case, this sequence will be updated by the address in the MAR Register.
(E)	Execution--Execution phase; perform operation specified.

E. Core Memory.

The UNIVAC 1005 Card Processor employs magnetic core storage modules with a capacity of 1024 characters each. The UNIVAC 1005 can be expanded to meet increased processing requirements in increments of 1024 characters to a maximum of 4096. Internal representation of each character in storage is by means of an internal binary code called XS3.

Data Representation. Excess three (XS3) is a method of notation that is used by the UNIVAC 1005 System. It establishes some measure of compatibility with the data formats of the other UNIVAC Computing Systems. The zone position is specified by the two high order bits, the numeric portion by low order four bits as in binary coded decimal notation. The difference exists in the numeric portion where each binary specification is a value that is three greater than its decimal equivalent. For example, the number 8 is represented in XS3 as:

ZONE	NUMERIC
00	1011

Note that the numeric portion, weighted with positional values of 8, 4, 2, and 1 from left to right, is actually equal to 11. Similarly, the number 6 is represented as:

ZONE	NUMERIC
00	1001

Here the numeric portion is specified as 9 or three greater than the decimal digit it represents.

There are several reasons for utilizing this method of notation in certain UNIVAC Systems. Some of these reasons are:

It allows three quantities to test less than 0.

It facilitates complementation.

It permits the carry to occur as in decimal notation.

An involved discussion of these and other reasons for the utilization of XS3 notation is beyond the scope of this manual. It is sufficient that the programmer is aware of the basic format and that this provides in the UNIVAC 1005 Computer a factor of data compatibility with other UNIVAC Systems. Figure 3 gives a listing of the XS3 code configurations.

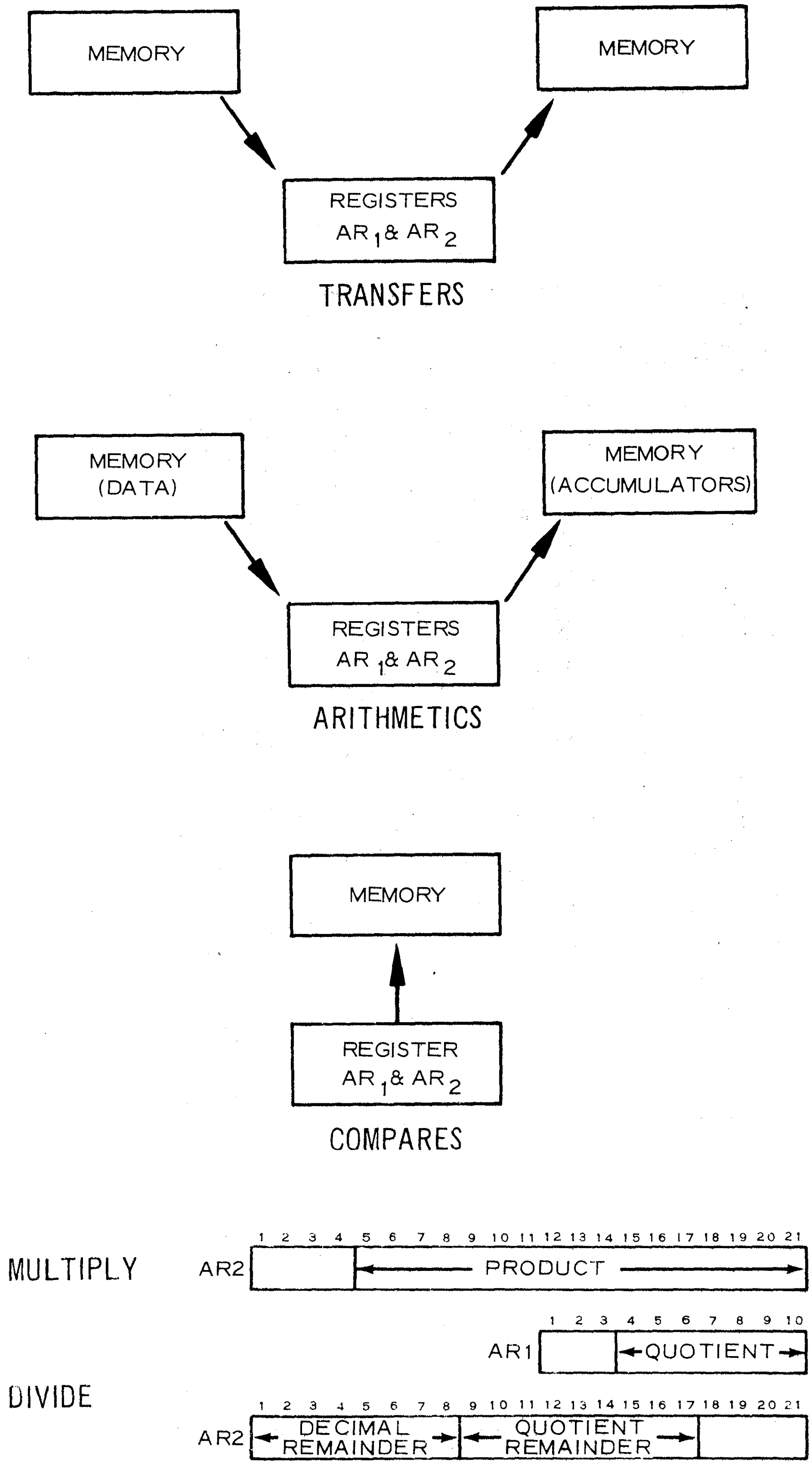


Figure 2. - Operation of Transient Registers

The alphabetic, numeric, and special characters utilized in the UNIVAC 1005 System.

80-COLUMN CODE

80-Col. Card Code	Printable Characters	XS-3 Code	80-Col. Card Code	Printable Characters	XS-3 Code
12–1	A	01 0100	7	7	00 1010
12–2	B	01 0101	8	8	00 1011
12–3	C	01 0110	9	9	00 1100
12–4	D	01 0111	12	&	01 0000
12–5	E	01 1000	11	- (minus)	00 0010
12–6	F	01 1001	12–0	?	01 0011
12–7	G	01 1010	11–0	! (exclam.)	10 0011
12–8	H	01 1011	0–1	/	11 0100
12–9	I	01 1100	2–8	+	11 0011
11–1	J	10 0100	3–8	#	01 1101
11–2	K	10 0101	4–8	@	10 1110
11–3	L	10 0110	5–8	: (colon)	01 0001
11–4	M	10 0111	6–8	>	11 1110
11–5	N	10 1000	7–8	' (apos.)	10 0000
11–6	O	10 1001	12–3–8	. (period)	01 0010
11–7	P	10 1010	12–4–8	⌑	11 1101
11–8	Q	10 1011	12–5–8	[	00 1111
11–9	R	10 1100	12–6–8	<	01 1110
0–2	S	11 0101	12–7–8	=	01 1111
0–3	T	11 0110	11–3–8	$	10 0010
0–4	U	11 0111	11–4–8	*	10 0001
0–5	V	11 1000	11–5–8	]	00 0001
0–6	W	11 1001	11–6–8	; (semi-col)	00 1110
0–7	X	11 1010	11–7–8	Δ	10 1111
0–8	Y	11 1011	0–2–8	≠	11 0000
0–9	Z	11 1100	0–3–8	, (comma)	11 0010
0	0	00 0011	0–4–8	%	11 0001
1	1	00 0100	0–5–8	(	10 1101
2	2	00 0101	0–6–8	\	00 1101
3	3	00 0110	0–7–8	)	11 1111
4	4	00 0111	Blank	Space N.P.	00 0000
5	5	00 1000			
6	6	00 1001			

Figure 3. - 80-Column Codes and UNIVAC XS3 Codes for 63 Printable Characters

1. Memory Allocation.

As previously stated, core memory is expandable, to meet increased processing loads, in increments of 1024 characters.

A portion of the 1024 character core memory is allocated to each of the input/output functions of the system--such as reading, punching and printing. The remaining portion of core memory is available for use by working programs. Under certain program conditions, part or all of the input/output memory areas may be used as expanded working core memory. For example, if a punch operation is not required for a particular program, the preassigned portion of core memory allocated to punching could be used as working storage. The 1005 Card Processor Control logic is such that "time-sharing" can be affected, allowing, simultaneous printing and punching, or punching and processing. (Reference Figure 4).

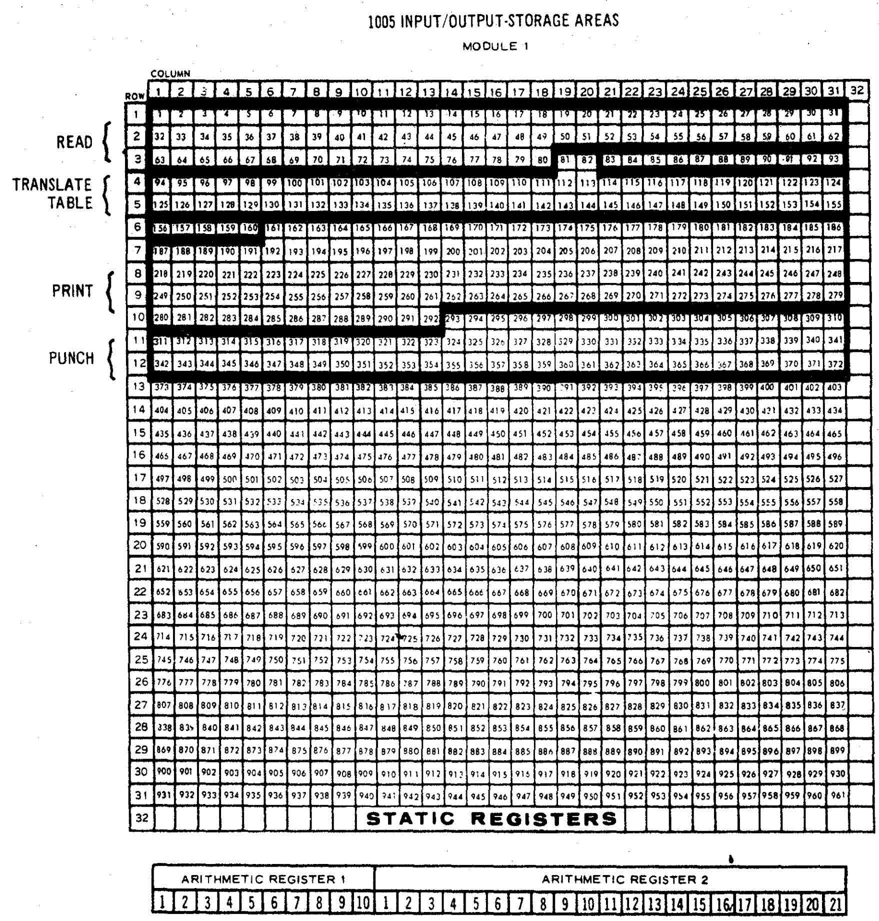


Figure 4. - 1005 Input/Output-Storage Areas - Module 1

2. Input/Output Buffer Areas

The three preassigned Input/Output buffers in the first module of the UNIVAC 1005 Card Processor are as follows.

Read Buffer Area. The read area is assigned the first 80 positions in core memory. Hence, the numeric addresses of the read area is ØØØ1 to ØØ8Ø. When ever the programmer gives an instruction to read a card, the card is read into this area. Column one of the input card is stored in the first position of the read buffer (ØØØ1), column two being stored in the second position (ØØØ2) and so on.

Print Buffer Area. There are 132 positions of core memory corresponding to the 132 print positions of the UNIVAC 1005 printer. When the programmer gives a print command, all 132 positions of the print buffer area are printed, the buffer is cleared to spaces, and the printer form is advanced. The core memory positions assigned to the print buffer are Ø161 to Ø292. The first character of the print buffer area (Ø161) corresponds to print position one, the second character (Ø162) corresponds to print position two, and so on.

Punch Buffer Area. There are 80 positions of core memory corresponding to the 80 columns of a punched card. The numeric addresses assigned to the punch buffer area are Ø293 to Ø372. When a punch command is executed, the first character of the punch buffer area is punched in card column one, the second character is punched in card column two, and so on.

The punch buffer area is not cleared during the punch cycle and the data remains the same in core memory.

Optional Buffer Areas. These additional buffer areas are explained in detail under their respective Input/Output Sections.

3. Memory Addressing.

Each character in the UNIVAC 1005 core memory is directly addressable by its numeric address. For example, the first character of the punch buffer area can be referenced by its numerical address Ø293, the second by Ø294 and so on.

CHAPTER 2

THE UNIVAC 1005 SINGLE ADDRESS ASSEMBLY SYSTEM

I. INTRODUCTION

To solve a problem, a computer must have a series of instructions which determine how the computer is to operate. In addition, the computer must be given one or more sets of data upon which to operate. This combination of instructions and data is called a program. A program must define, in complete detail, exactly what the computer is to do, under every conceivable combination of circumstances, with the data which is read into or processed by the computer. The number of instructions required for the complete solution of a problem may be a few hundred or many thousands, depending on the problem. The computer may refer to these instructions one after another, or it may repeat, skip, or modify over certain instructions, depending upon immediate results or circumstances.

These instructions are understood by the computer in a form known as Machine Language, a form which is difficult for the programmer to encode. In order to facilitate coding, considerable time and effort has been expended in developing programming systems that allow the programmer to write in a symbolic language more easily comprehensive to him than machine language.

Associated with a programming system is a machine language program called an Assembler. The assembler accepts a program written in symbolic language (source program) and converts it into machine language (object program).

II. GENERAL DESCRIPTION

The symbolic language used by the UNIVAC 1005 Card Processing System is single address in design and is intended to provide an easy to learn, easy to use tool whereby data processing requirements can be translated into machine coded instructions.

The machine language program or assembly system associated with the UNIVAC 1005 symbolic language is called SAAL (Single Address Assembly Language). This assembly system consists of two passes, SAAL 1 and SAAL 2.

The first pass, SAAL 1 relates each symbolic reference (label) in the symbolic program (source program) with its appropriate position in core memory. This relationship between symbolic labels in the source program and core memory position is retained in memory and utilized in SAAL 2. This noted relationship is commonly referred to as the "TAG" or "Label" Table.

The second pass, SAAL 2, interprets each operand field in the source program, determines its length and core position using the "LABEL" Table generated by SAAL 1, and produces a UNIVAC 1005 machine code object program deck. In addition, a one for one listing is prepared equating each symbolic line of coding in the source program with the generated machine code.

III. INSTRUCTION FORMAT

The UNIVAC 1005 Machine Code instruction consists of five characters. The format of the instruction characters on this basis is illustrated below.

1	2 3	4 5
OP	M	L

OP - Indicates the operation to be performed.

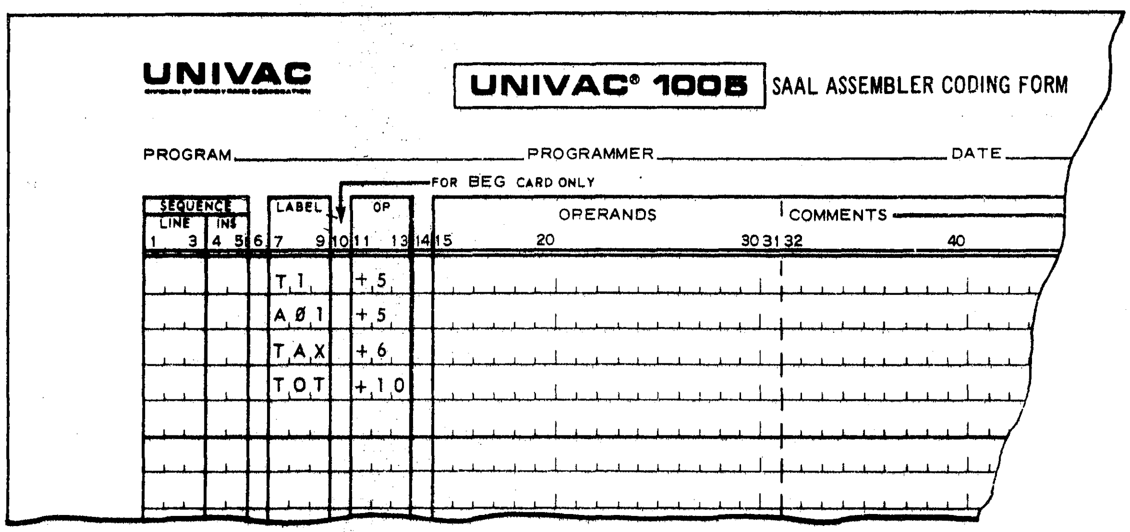
M - Indicates most significant location.

L - Indicates least significant location.

A. SYMBOLIC CODING FORMAT

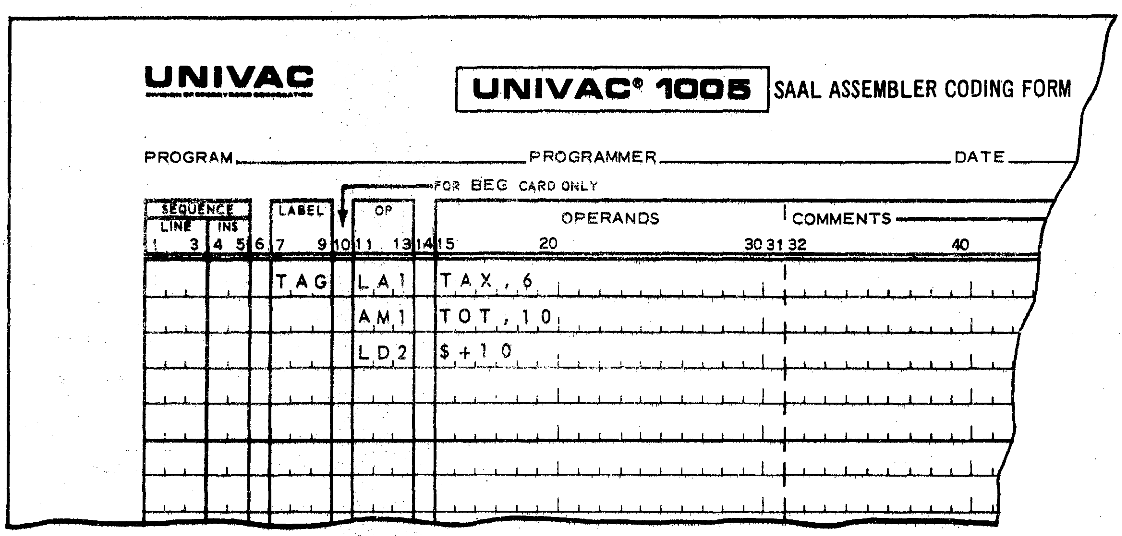
In writing a program in SAAL symbolic language, the programmer is primarily concerned with three fields: Label field, Operation field, and Operand field. In addition, it is possible to annotate the symbolic language at the time it is written through the use of comments which will provide clarity for the programmer and relate coding to its associated flow chart.

1. Label Field. A label is a method of identifying either a symbolic line of coding or a word of data. In writing a label in the assembly language SAAL, the programmer may use any meaningful combination of one to three characters. Of these three characters, the first may be any alpha character, including special characters, except the dollar sign, asterisk, plus, minus, or comma. The second and third position of the label field, if present, may be either alphabetic or numeric or special characters, including the dollar sign but excluding the asterisk, plus, minus, and comma. In writing a label in the label field of a symbolic line, the first character of the label must appear in the leftmost position of the label field. The following are examples of acceptable labels.


UNIVAC
UNIVAC® 1005 SAAL ASSEMBLER CODING FORM
PROGRAM PROGRAMMER DATE
FOR BEG CARD ONLY

SEQUENCE LINE 1-3	INS 4-5	6	LABEL 7-9	10	OP 11-13	14	OPERANDS 15-30	COMMENTS 31-
			T1		+5			
			AØ1		+5			
			TAX		+6			
			TOT		+10			

2. Operation Field. In the operation field, the programmer places a symbolic code indicating the machine function that is to be performed. These function codes are explained subsequently. An example of acceptable operation codes is shown below.


UNIVAC
UNIVAC® 1005 SAAL ASSEMBLER CODING FORM
PROGRAM PROGRAMMER DATE
FOR BEG CARD ONLY

SEQUENCE LINE 1-3	INS 4-5	6	LABEL 7-9	10	OP 11-13	14	OPERANDS 15-30	COMMENTS 31-
			TAG		LA1		TAX,6	
					AM1		TOT,10	
					LD2		$+10	

3. Operand Field. The operand field of a symbolic program follows the operation field, and it is used to inform the assembler which location is to be addressed in conjunction with the operation to be performed. For example, if the programmer called for data to be added in the Arithmetic Register 1, the operand field would tell the processor where to go for the data to be added. Also, the operand field would tell the assembler how many positions of memory to accumulate in Arithmetic Register 1.

The following example depicts the instructions required to add a five digit numeric field to Arithmetic Register one, and store the result back into core memory.

UNIVAC — UNIVAC® 1005 SAAL ASSEMBLER CODING FORM

PROGRAM ____ PROGRAMMER ____ DATE ____

SEQUENCE LINE 1–3	INS 4–5	6	LABEL 7–9	10 (FOR BEG CARD ONLY)	OP 11–13	14	OPERANDS 15–30	COMMENTS 31–40
					AR1		T1,5	
					SA1		AØ1,5	

In addition the M position of the operand may be incremented or decremented in order to provide increased flexibility in addressing.

In the following example the two least significant characters of a ten character field called FD1 are to be loaded into Arithmetic Register 1. In order to address these characters an increment of eight is added to the base address of the field thereby obtaining the desired result.

UNIVAC — UNIVAC® 1005 SAAL ASSEMBLER CODING FORM

PROGRAM ____ PROGRAMMER ____ DATE ____

SEQUENCE LINE 1–3	INS 4–5	6	LABEL 7–9	10 (FOR BEG CARD ONLY)	OP 11–13	14	OPERANDS 15–30	COMMENTS 31–40
					LA1		FD1+8,2	
					CA1		AR2+19,2	
					JEA		$+10	

If field FD_1 were decremented by eight, the seventh and eighth characters immediately to the left of the most significant character of FD_1

would be loaded into Arithmetic Register one. When incrementing or decrementing an address, the programmer may use one, two or three characters. The programmer can increment or decrement from 1 to 999 positions in memory; however, an operand may not be split between memory modules.

NOTES:
1) In the above example the second instruction references Arithmetic Register two in the operand field. Arithmetic Register 1 and Arithmetic Register 2 are predefined labels (AR1 and AR2) and can be referenced as operands in the same manner as labels.

2) In the above illustration the third instruction references $ in the operand field. $ represents the current value of the location counter which may be modified (+ or -) in increments of five (5). Thus, in the illustration, if an equal condition is met, control will bypass the next sequential instruction.

3) When modifying an instruction within the program with another instruction, both the instruction being modified and the modifier should be labeled.

4) If the length is not specified, the assembler assumes an operand of 5 characters.

4. Comments. Comments are coded starting in column 32 of the code sheet. The comments written here by the programmer are not looked at by the assembler. However, they do appear on the printout from SAAL 2; they are put into the code sheet for reference only. Any character may be used in the Comments section of the sheet.

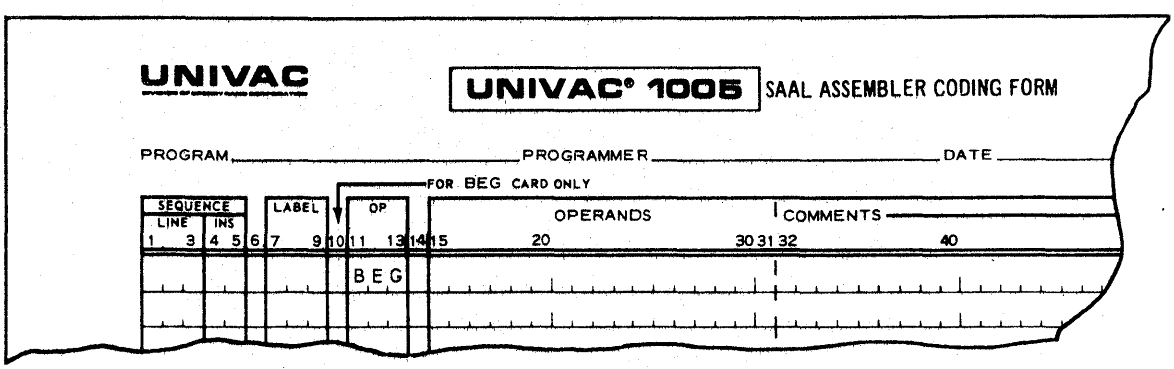
IV. PROGRAM ORGANIZATION

Certain required parameter cards must be supplied to the assembler in order to properly position constants, headers, or any data the programmer wishes to store in memory. These parameter cards are called directives. They direct the assembly in the allocation of core memory for the various divisions of a symbolic program. They are described below.

A. BEG DIRECTIVE

The first card of every symbolic program written in the assembly language SAAL must have BEG card or directive. This card initiates the assembly process.

For example:



B. CRD DIRECTIVE

CRD Card is used to call the assembler's attention to the Read Area in core memory. CRD is punched in the operation field of the card format. Labels are then used to define areas within the Read Area. The label for each field is placed in the label field on the card. In the operation field, punch a minus (-) in column 11. In column 15 punch the position in the read area the program wishes to designate.

For example:

UNIVAC

UNIVAC® 1005 SAAL ASSEMBLER CODING FORM

PROGRAM ______ PROGRAMMER ______ DATE ______

SEQUENCE LINE (1–3)	SEQUENCE INS (4–5)	6	LABEL (7–9)	10	OP (11–13)	14	OPERANDS (15–)	COMMENTS
					BEG			
					CRD			
			FSN		-		1	
			NOM		-		16	
			CAT		-		38	
			AMT		-		56	
			QTY		-		71	

C. PRT DIRECTIVE

This card is used to direct the assembler's attention to the print area in core memory. Like the Read Area, the Print Area may be labeled. The format for doing this is the same as for the Read Area.

For example:

UNIVAC

UNIVAC® 1005 SAAL ASSEMBLER CODING FORM

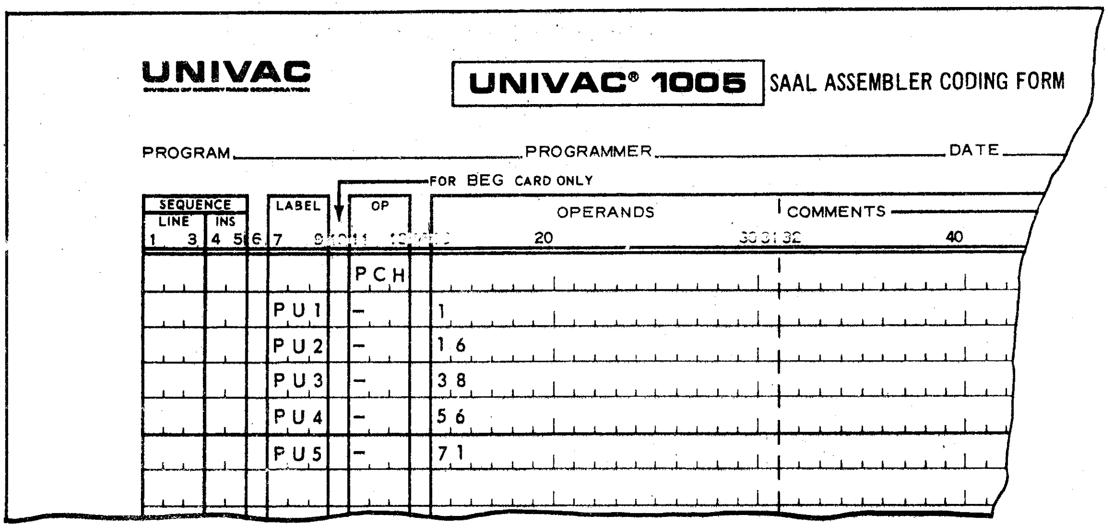
PROGRAM ______ PROGRAMMER ______ DATE ______

SEQUENCE LINE (1–3)	SEQUENCE INS (4–5)	6	LABEL (7–9)	10	OP (11–13)	14	OPERANDS (15–)	COMMENTS
					PRT			
			PT1		-		1	
			PT2		-		49	
			PT3		-		87	
			PT4		-		109	

D. PCH DIRECTIVE

As in the Read and Print Areas, subdivision of the Punch Area is possible. The format is the same as described for the CRD directive.

For example:



UNIVAC — UNIVAC® 1005 SAAL ASSEMBLER CODING FORM

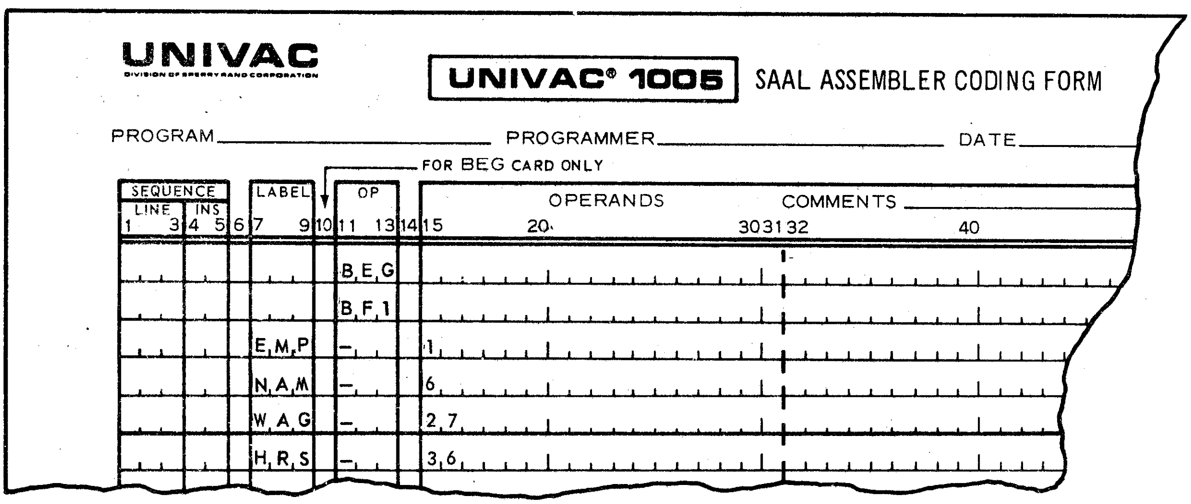
PROGRAM ________ PROGRAMMER ________ DATE ________

SEQUENCE LINE (1–3)	INS (4–5)	6	LABEL (7–9)	10	OP (11–13)	14	OPERANDS (15–)	COMMENTS
					PCH			
			PU1		–		1	
			PU2		–		16	
			PU3		–		38	
			PU4		–		56	
			PU5		–		71	

E. BF1 DIRECTIVE (Buffer 1)

BF1 card is used to call the assembler's attention to the 1st core position of Bank 1. In this regard, it is similar to the CRD directive. Its primary use is to define areas for peripheral devices, i.e. paper tape. BF1 is punched in the operation field of the card format. Labels are then used to define areas. The label for each field is placed in the label field on the card. In the operation field, punch a minus (-) in Column 11. In Column 15, punch the position in the buffer area the program wishes to designate.

For example:



UNIVAC — UNIVAC® 1005 SAAL ASSEMBLER CODING FORM

PROGRAM ________ PROGRAMMER ________ DATE ________

SEQUENCE LINE (1–3)	INS (4–5)	6	LABEL (7–9)	10	OP (11–13)	14	OPERANDS (15–)	COMMENTS
					BEG			
					BF1			
			EMP		–		1	
			NAM		–		6	
			WAG		–		27	
			HRS		–		36	

EMP would be assigned the location starting at 0001, NAM at 0006 and so forth.

F. BF2 DIRECTIVE (Buffer 2)

BF2 card is used to call the assembler's attention to the 1st core position of Bank 2. Its primary use is to define areas for peripheral devices, i.e. magnetic tape. As in BF1, buffer 2 may be labeled. The format for doing this is the same as described for BF1.

For example:

UNIVAC

UNIVAC® 1005 SAAL ASSEMBLER CODING FORM

PROGRAM ______ PROGRAMMER ______ DATE ______

FOR BEG CARD ONLY

SEQUENCE LINE 1 3	INS 4 5	6	LABEL 7 9	10	OP 11 13	14	OPERANDS 15 20 30	COMMENTS 31 32 40
					BF2			
			FSN		-		1	
			NOM		-		16	
			CAT		-		34	
			VAL		-		52	
			QTY		-		67	

FSN would be assigned the location starting at 0962, NOM at 0977 and so forth.

G. BF3 DIRECTIVE (Buffer 3)

BF3 card is used to call the assembler's attention to the 1st core position of Bank 3. Its primary use is to define areas for peripheral devices. As in BF1, buffer 3 may be labeled. The format for doing this is the same as described for BF1.

For example:

UNIVAC

UNIVAC® 1005 SAAL ASSEMBLER CODING FORM

PROGRAM ______ PROGRAMMER ______ DATE ______

FOR BEG CARD ONLY

SEQUENCE LINE 1 3	INS 4 5	6	LABEL 7 9	10	OP 11 13	14	OPERANDS 15 20 30	COMMENTS 31 32 40
					BF3			
			FD1		-		1	
			FD2		-		67	
			FD3		-		150	
			FD4		-		455	

FD1 would be assigned the location starting at 1923, FD2 at 1989 and so forth.

H. BF4 DIRECTIVE (Buffer 4)

BF4 is used to call the assembler's attention to the 1st core position of Bank 4. Its primary use is to define areas for peripheral devices. As in BF1, buffer 4 may be labeled. The format for doing this is the same as described for BF1.

For example:

UNIVAC
DIVISION OF SPERRY RAND CORPORATION

UNIVAC® 1005 SAAL ASSEMBLER CODING FORM

PROGRAM ______ PROGRAMMER ______ DATE ______

FOR BEG CARD ONLY

SEQUENCE LINE 1–3	SEQUENCE INS 4–5	6	LABEL 7–9	10	OP 11–13	14	OPERANDS 15–20 … 30	COMMENTS 31 32 … 40
					BF4			
			TAX		–		1	
			TDT		–		26	
			QTY		–		58	
			AL		–		127	
			G		–		162	

TAX would be assigned the location starting at 2884, TDT at 2909 and so forth.

I. ORG DIRECTIVE

The ORG Directive informs the assembler that the programmer wished to adjust the assembly address counter to the numeric value contained in the operand field. For example, if the programmer wishes to start storing at one particular place in memory, he specifies this by placing the numeric address in the operand field. This numeric address must be four characters.

The following example would origin the next instruction, constant, or work area in position Ø373 of core memory.

UNIVAC
DIVISION OF SPERRY RAND CORPORATION

UNIVAC® 1005 SAAL ASSEMBLER CODING FORM

PROGRAM ______ PROGRAMMER ______ DATE ______

FOR BEG CARD ONLY

SEQUENCE LINE 1–3	SEQUENCE INS 4–5	6	LABEL 7–9	10	OP 11–13	14	OPERANDS 15–20 … 30	COMMENTS 31 32 … 40
					ORG		0373	

The programmer may use an ORG statement anywhere in a program, provided he complies with the following rules.

1. The operand value must be a four digit decimal number.

2. If the ORG directive is employed within the procedure division (after the STA directive) the new assembly address must be a multiple of thirty-one (31) plus one (1), beginning with 1, 32, 63, and so on.

3. The ORG directive must be employed before the 1st literal instruction.

J. LITERALS

The use of literal instructions enables the assembler to move the number of characters specified by the operation code from the operand field to an equal number sequential core locations, beginning at the address specified by the preceding ORG directive.

With literal instructions, the programmer is able to store headers, constants, or set aside storage for work areas.

The literal instruction consists of a label in the label field of the symbolic deck, a plus sign (+) in column 11 of the operation field followed by the number of positions to be set aside. The operand portion of the card contains the constant or literal to be stored. The maximum for one line is 34 positions, however this line may not be split between memory modules. For example:

PROGRAM ______ PROGRAMMER ______ DATE ______

FOR BEG CARD ONLY

SEQUENCE LINE 1–3	INS 4–5	6	LABEL 7–9	10	OP 11–13	14	OPERANDS 15–30	COMMENTS 31–40
					ORG		0373	
			HD1		+10		END OF JOB	
			K2		+2		10	
			WS		+20			

In the first example, HD1, the constant "END OF JOB" is stored in 10 positions of memory, which can be referred to by HD1.

In the second example, K2, the constant "10" is stored in 2 positions of memory. To refer to this constant, the label K2 need only be called.

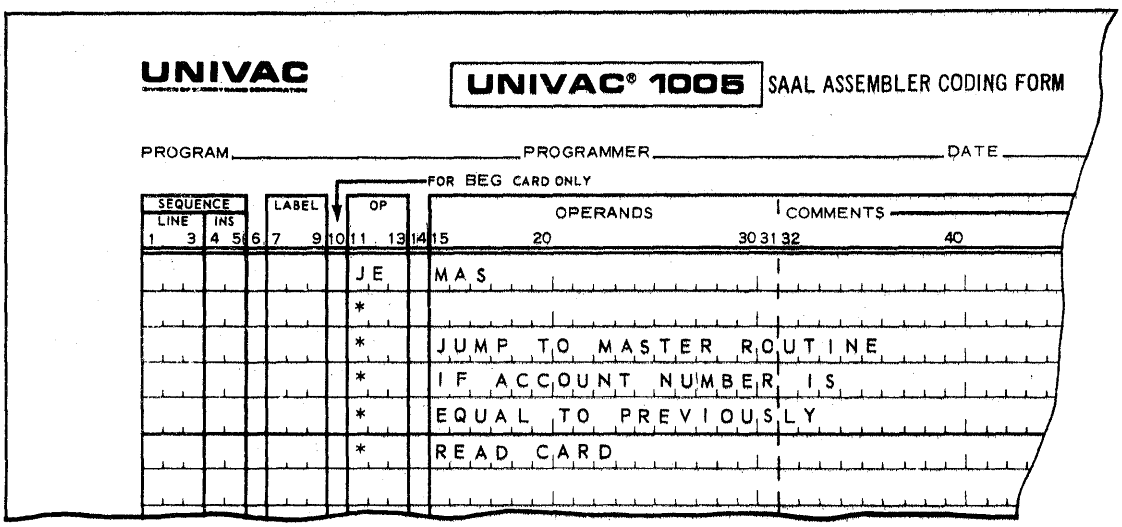
The third example, WS, a work area of 20 blank positions is set aside, that is labeled WS for programming reference.

K. * COMMENTS CARD

An asterisk punched in the operation field (Col. 11) indicates a comments card, and is listed 80/80 on the assembly printout. This card is used by the programmer to facilitate reference to the assembly printout, and/or to explain certain portions of his program.

A Comments Card may be used anywhere within a program. The programmer is not limited by the number of the cards he may use.

For example:



UNIVAC | UNIVAC® 1005 SAAL ASSEMBLER CODING FORM

PROGRAM ______ PROGRAMMER ______ DATE ______

FOR BEG CARD ONLY

SEQUENCE LINE 1 3	INS 4 5	6	LABEL 7 9	10	OP 11 13	14	OPERANDS 15 20 30 31	COMMENTS 32 40
					JE		MAS	
					*			
					*		JUMP TO MASTER RO	UTINE
					*		IF ACCOUNT NUMBER	IS
					*		EQUAL TO PREVIOUS	LY
					*		READ CARD	

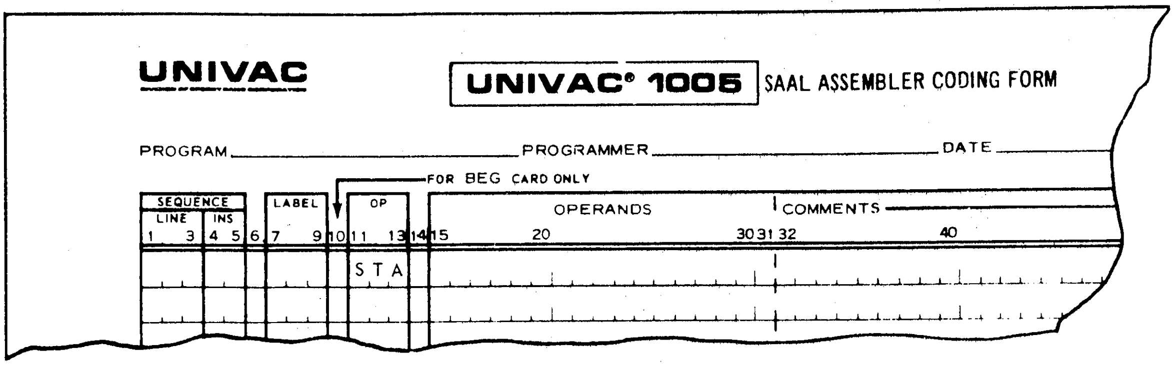
In this example, the programmer has used five comments cards to break into the printout format. The assembler would only interpret the jump instruction, and the Comments Cards would be listed as they appear on the coding form.

L. STA DIRECTIVE

This directive terminates the DATA DIVISION and marks the beginning of the PROCEDURE DIVISION of the program. The assembler, upon decoding this card, advances the assembly address counter to the next row of core memory, and assigns the addresses to the instructions of the program from that point. The PROCEDURE DIVISION of every program must be indicated by this directive.

Note: All labels used in the 1005 program, with the exception of instruction labels, must be defined before the STA card either in the I/O sections or as a literal.

For example:



UNIVAC — UNIVAC® 1005 SAAL ASSEMBLER CODING FORM

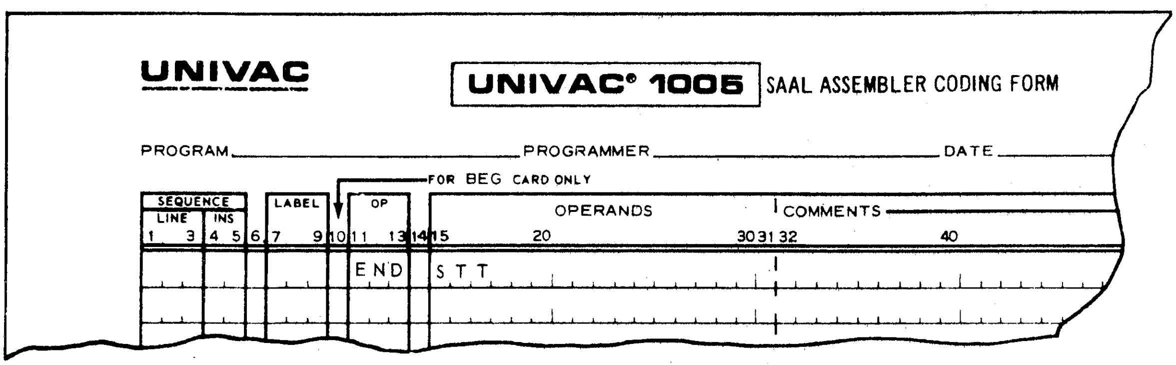
PROGRAM ____ PROGRAMMER ____ DATE ____

SEQUENCE LINE 1–3	INS 4–5	6	LABEL 7–9	10	OP 11–13	14	OPERANDS 15–30	COMMENTS 32–40
					STA			

M. END DIRECTIVE

The END directive is the last card of the source deck. This card must always be present. The purpose of this card is to inform the assembler that all card instructions used in the program have been inserted and to terminate the assembly. The operand field must have the tag of the first instruction.

For example:



UNIVAC — UNIVAC® 1005 SAAL ASSEMBLER CODING FORM

PROGRAM ____ PROGRAMMER ____ DATE ____

SEQUENCE LINE 1–3	INS 4–5	6	LABEL 7–9	10	OP 11–13	14	OPERANDS 15–30	COMMENTS 32–40
					END		STT	

V. INSTRUCTION REPERTOIRE

Each instruction in the UNIVAC 1005 consists of five character positions, and are sequentially numbered in increments of five, beginning with the first character of a row. The last character of a row is utilized by the U1005 logic to designate at which row the next sequential instruction is located.

There are four general classes of instructions varying slightly in format.

Class I: Class I instructions contain an "M" address and an "L" modifier. The "M" portion defines the most significant position of a field, where the "L" portion defines the length of the field. All Arithmetic and Transfer instructions are Class I.

Class II: Class II instructions contain only an "M" address indicating the most significant character of an instruction. This format is employed exclusively by Jump or Branching instructions.

Class III: Class III instructions are Input/Output or External Function Commands, and contain a mnemonic code in the "M" portion of an instruction indicating the I/O device or devices to be initiated.

Class IV: Class IV instructions are Input/Output or External Function Commands, and contain a mnemonic code, Buffer (BF_n), and length in the "M" portion of an instruction indicating the I/O device, memory bank, and length of operand to be initiated.

A. INSTRUCTION REPERTOIRE -- CENTRAL PROCESSOR

The Central Processor instructions pertain to Class I and Class II and are explained in detail on the following pages.

LOAD ASCENDING: LAr M,L

Function: Load ascending L most significant characters from the field specified by M, into the L least significant character positions of AR1 or 2.

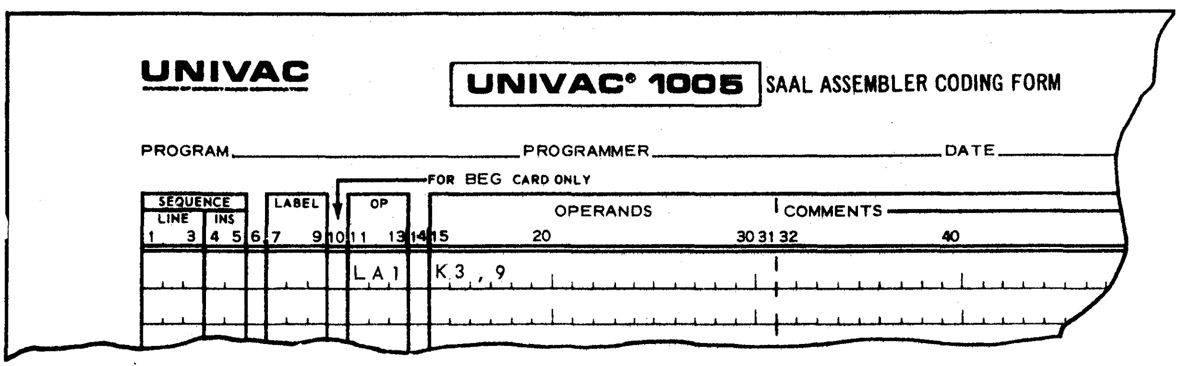
Notes: a.) L must be decimal number.

b.) L most significant characters of the field specified by M, are transferred in ascending order to the L least significant positions of the specified register.

c.) When L is less than the capacity of the register the remaining positions of the register will be space filled.

d.) When L is greater than the capacity of the register truncation will occur and the most significant characters of the field will be deleted.

Example: Load Arithmetic Register 1 with a nine character constant.


UNIVAC

UNIVAC® 1005 SAAL ASSEMBLER CODING FORM

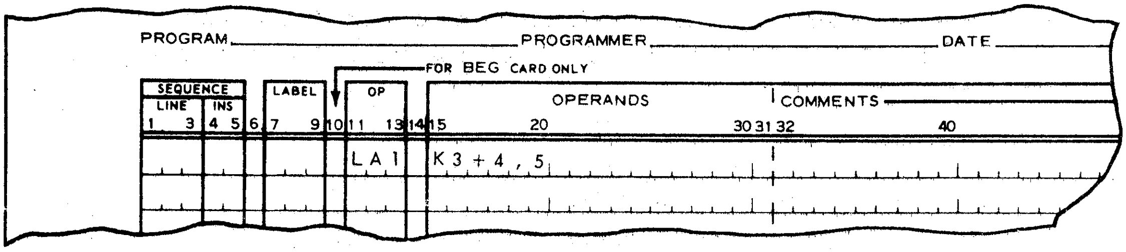
PROGRAM ______ PROGRAMMER ______ DATE ______

FOR BEG CARD ONLY

SEQUENCE LINE 1 3	INS 4 5	6	LABEL 7 9	10	OP 11 13	14	OPERANDS 15 20 30 31 32	COMMENTS 40
					LA1		K3,9	

*AR1 (before)	=	7 9 2 4 6 5 1 3 6 4
K3	=	S U B Δ T O T A L
AR1 (after)	=	Δ S U B Δ T O T A L

Load Register 1 with a five character constant.

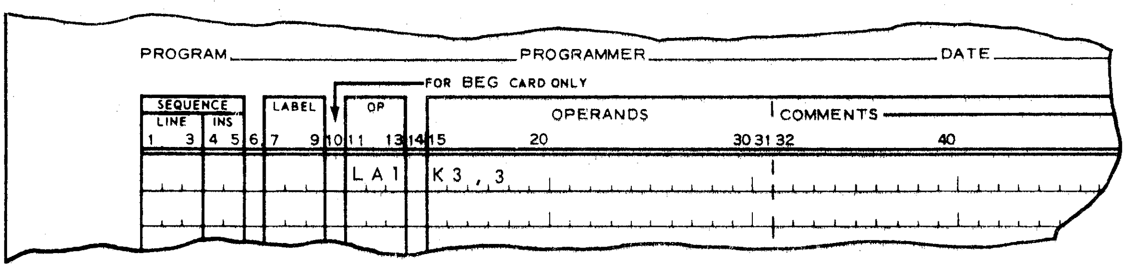

PROGRAM PROGRAMMER DATE

FOR BEG CARD ONLY

SEQUENCE LINE 1 3	INS 4 5	6	LABEL 7 9	10	OP 11 13	14	OPERANDS 15 20 30 31	COMMENTS 32 40
					LA1		K3+4,5	

*AR1 (before) = 7 9 2 4 6 5 1 3 6 4
K3 = S U B Δ T O T A L
AR1 (after) = Δ Δ Δ Δ Δ T O T A L

Load Register 1 with a three character constant.


PROGRAM PROGRAMMER DATE

FOR BEG CARD ONLY

SEQUENCE LINE 1 3	INS 4 5	6	LABEL 7 9	10	OP 11 13	14	OPERANDS 15 20 30 31	COMMENTS 32 40
					LA1		K3,3	

*AR1 (before) = 7 9 2 4 6 5 1 3 6 4
K3 = S U B Δ T O T A L
AR1 (after) = Δ Δ Δ Δ Δ Δ Δ S U B

*The functions indicated are identical for AR2 with the exception that larger fields can be manipulated.

LOAD DESCENDING: LDr M,L

Function: Load Descending L consecutive characters whose most significant character is at M, into the L most significant positions of AR1 or 2.

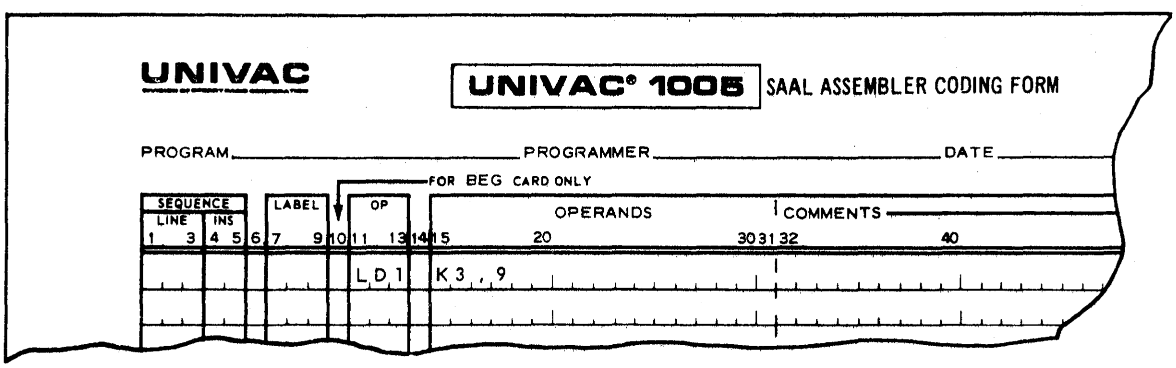
Notes: a.) L must be a decimal number.

b.) L characters of the field specified by M are transferred to the register.

c.) When L is less than the capacity of the register the remaining positions of the register will be space filled.

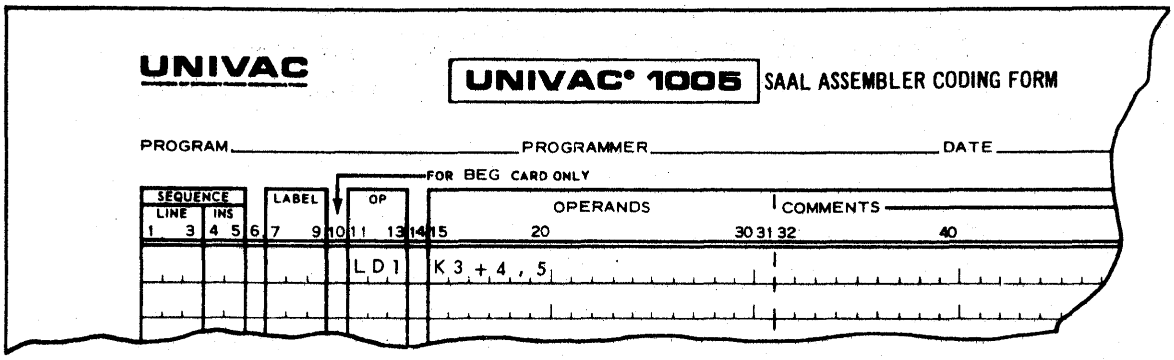
d.) When L is greater than the capacity of the register truncation will occur and the least significant characters of the field will be deleted.

Example: Load Arithmetic Register 1 with a nine character constant called K3.



*AR1	(before)	=	7	9	2	4	6	5	1	3	6	4
K3		=	S	U	B	Δ	T	O	T	A	L	
AR1	(after)	=	S	U	B	Δ	T	O	T	A	L	Δ

Load Arithmetic Register 1 with a five character constant called K3.


UNIVAC

UNIVAC® 1005 SAAL ASSEMBLER CODING FORM

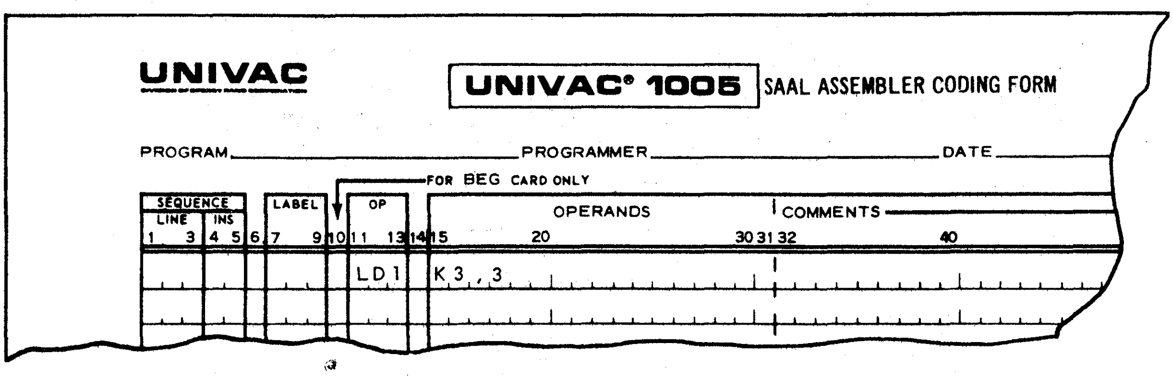
PROGRAM ________ PROGRAMMER ________ DATE ________

FOR BEG CARD ONLY

SEQUENCE LINE 1 3	INS 4 5	6	LABEL 7 9	10	OP 11 13	14	OPERANDS 15 20 30	COMMENTS 31 32 40
					LD1		K3+4,5	

*AR1 (before) = 7 9 2 4 6 5 1 3 6 4
K3 = S U B Δ T O T A L
AR1 = T O T A L Δ Δ Δ Δ Δ

Load Arithmetic Register 1 with a three character constant called K3.


UNIVAC

UNIVAC® 1005 SAAL ASSEMBLER CODING FORM

PROGRAM ________ PROGRAMMER ________ DATE ________

FOR BEG CARD ONLY

SEQUENCE LINE 1 3	INS 4 5	6	LABEL 7 9	10	OP 11 13	14	OPERANDS 15 20 30	COMMENTS 31 32 40
					LD1		K3,3	

*AR1 (before) = 7 9 2 4 6 5 1 3 6 4
K3 = S U B Δ T O T A L
AR2 = S U B Δ Δ Δ Δ Δ Δ Δ

*The functions indicated are identical for AR2 with the exception that larger fields can be manipulated.

LOAD PRINT: LPR M,L

Function: Load descending L consecutive characters whose most significant character is a M, into the L most significant positions of the print buffer.

Note:

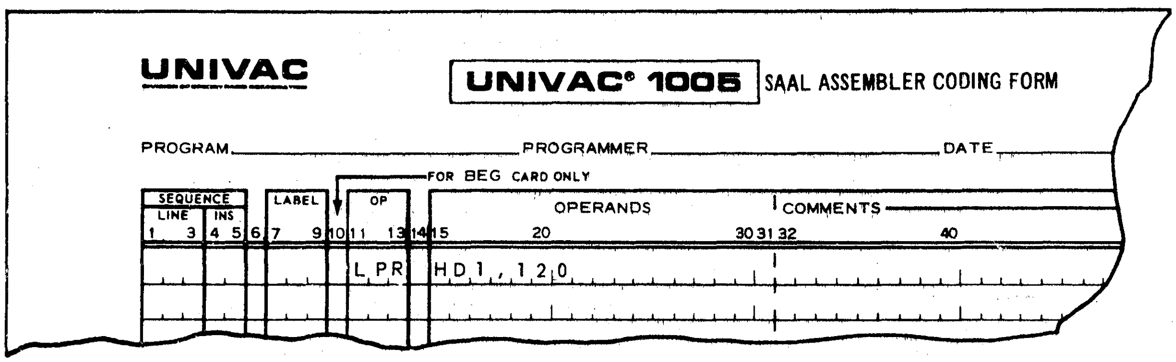
a.) L must be a decimal number, and should range from 1 to 132.

b.) L characters of the field specified by M are transferred to the most significant positions of the print buffer.

c.) When L is less than the capacity of the print buffer the remaining positions of the buffer are space filled.

d.) When L is greater than the capacity of the print buffer the least significant characters of the sending field will be truncated.

Example: Load the Print Buffer with the first header line labeled HD1.



STORE ASCENDING: SAr M,L

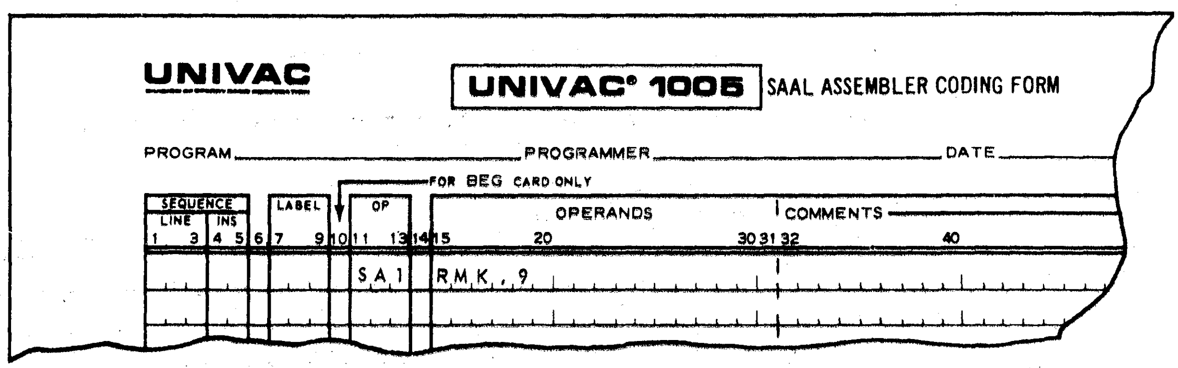
Function: Store ascending L least significant characters from AR1 or 2, into the L most significant positions of the field specified by M.

Notes: a.) L must be a decimal number.

b.) L characters are transferred in ascending order (least to most) from AR1 or 2 to the most significant positions of the field specified by M.

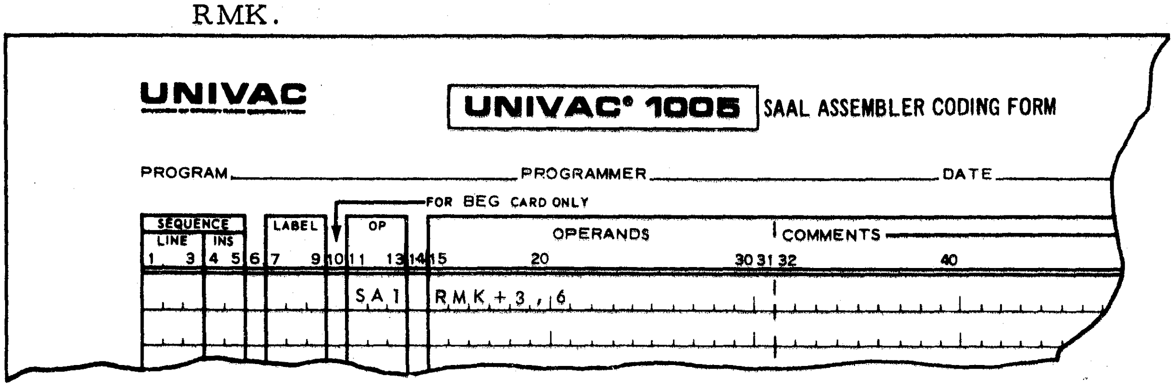
c.) When L is greater than the capacity of the register the receiving field will be space filled.

Example: . Store the nine least significant characters of AR1 into the field labeled RMK.



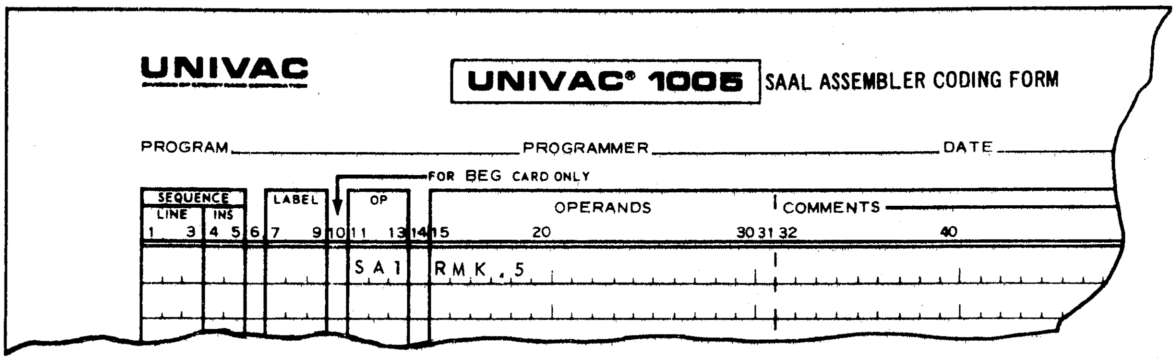
RMK (before) = Δ Δ Δ $ 1 0 . 1 5
*AR1 = Δ S U B Δ T O T A L
RMK (after) = S U B Δ T O T A L

. Store the six least significant characters of AR1 into the six least significant character positions of the field labeled RMK.



RMK (before) = Δ Δ Δ $ 1 0 . 1 5
*AR1 = Δ S U B Δ T O T A L
RMK (after) = Δ Δ Δ Δ T O T A L

. Store the five least significant characters of AR1 into the five most significant character positions of the field labeled RMK.



RMK	(before)	=	Δ Δ Δ Δ Δ Δ 1 1 5
*AR1		=	Δ S U B Δ T O T A L
RMK	(after)	=	T O T A L Δ 1 1 5

*The functions indicated are identical for AR2 with the exception that larger fields can be manipulated.

STORE DESCENDING: SDr M,L

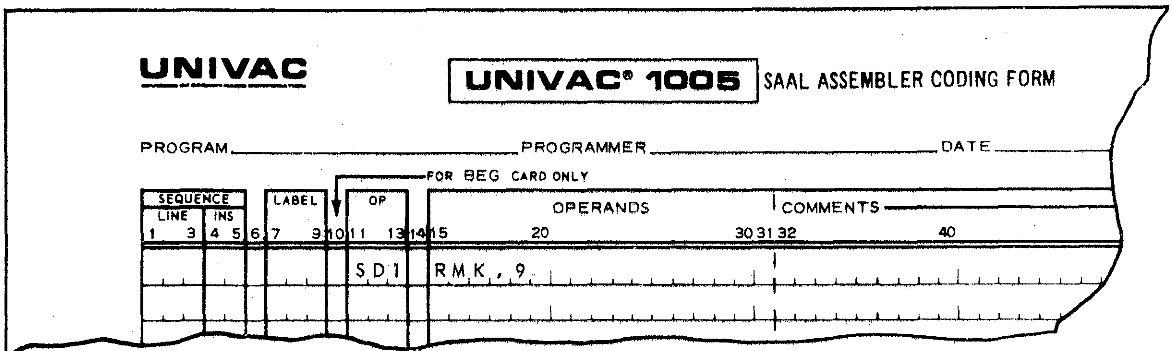
Function: Store descending L most significant characters from AR1 or 2 into the L most significant positions of the field specified by M.

Notes: a.) L must be a decimal number.

b.) L characters are transferred from AR1 or 2 to the most significant positions of the field specified by M.

c.) When L is greater than the capacity of the register the receiving field will be space filled.

Example: . Store the nine most significant characters of AR1 into the field labeled RMK.


UNIVAC — UNIVAC® 1005 SAAL ASSEMBLER CODING FORM

PROGRAM ____ PROGRAMMER ____ DATE ____

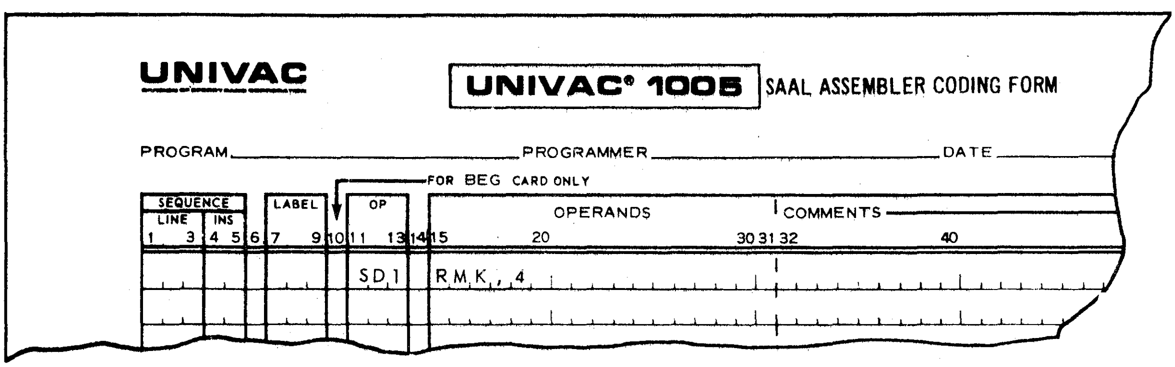
SEQUENCE LINE 1-3	INS 4-5	6	LABEL 7-9	10	OP 11-13	14	OPERANDS 15	COMMENTS 31 32
					SD1		RMK,9	

RMK (before) = Δ Δ Δ $ 1 0 . 1 5

*AR1 = S U B Δ T O T A L Δ

RMK (after) = S U B Δ T O T A L

. Store the four most significant characters of AR1 into the four most significant positions of the field labeled RMK.


UNIVAC — UNIVAC® 1005 SAAL ASSEMBLER CODING FORM

PROGRAM ____ PROGRAMMER ____ DATE ____

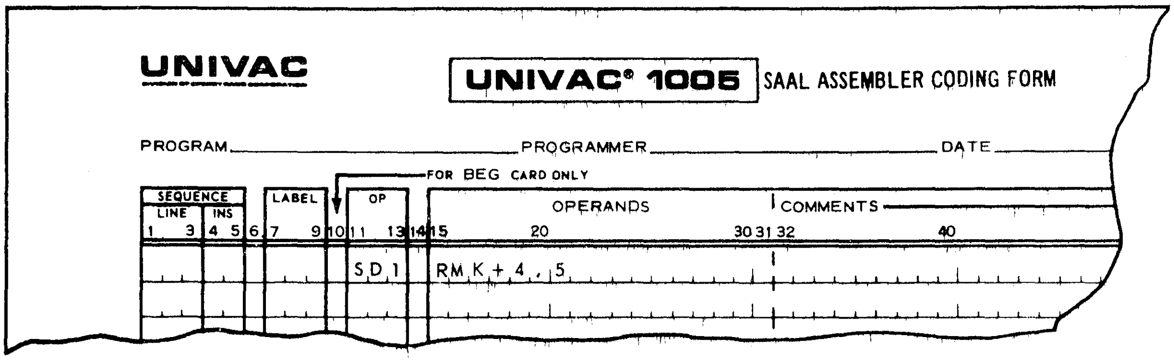
SEQUENCE LINE 1-3	INS 4-5	6	LABEL 7-9	10	OP 11-13	14	OPERANDS 15	COMMENTS 31 32
					SD1		RMK,4	

RMK (before) = Δ Δ Δ $ 1 0 . 1 5

AR1 = S U B Δ T O T A L Δ

RMK (after) = S U B Δ 1 0 . 1 5

. Store the five most significant characters of AR1 into the five least significant positions of the field labeled RMK.


UNIVAC

UNIVAC® 1005 SAAL ASSEMBLER CODING FORM

PROGRAM ____ PROGRAMMER ____ DATE ____

FOR BEG CARD ONLY

SEQUENCE LINE 1 3	INS 4 5	6	LABEL 7 9	10	OP 11 13	14	OPERANDS 15 20 30 31	COMMENTS 32 40
					SD1		RMK+4,5	

RMK (before)	=	1 1 5 Δ Δ Δ Δ Δ Δ
*AR1	=	S U B Δ T O T A L Δ
RMK	=	1 1 5 Δ S U B Δ T

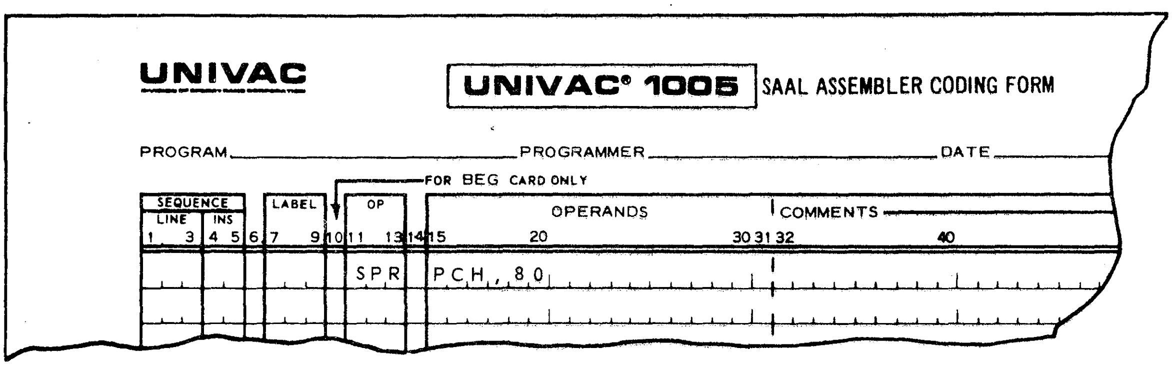
*The functions indicated are identical for AR2 with the exception that larger fields can be manipulated.

STORE PRINT: SPR M,L

Function: Store descending L most significant characters from the Print Buffer into the L most significant positions of the field specified by M.

Notes:
a.) L must be a decimal number.
b.) L characters are transferred from the Print Buffer to the most significant positions of the field specified by M.
c.) When L is greater than the capacity of Print Buffer (L > 132) the receiving field will be space filled.

Example: . Store the eighty most significant characters of the Print Buffer into the punch buffer.



PCH is the tag assigned to the most significant position of the punch buffer.

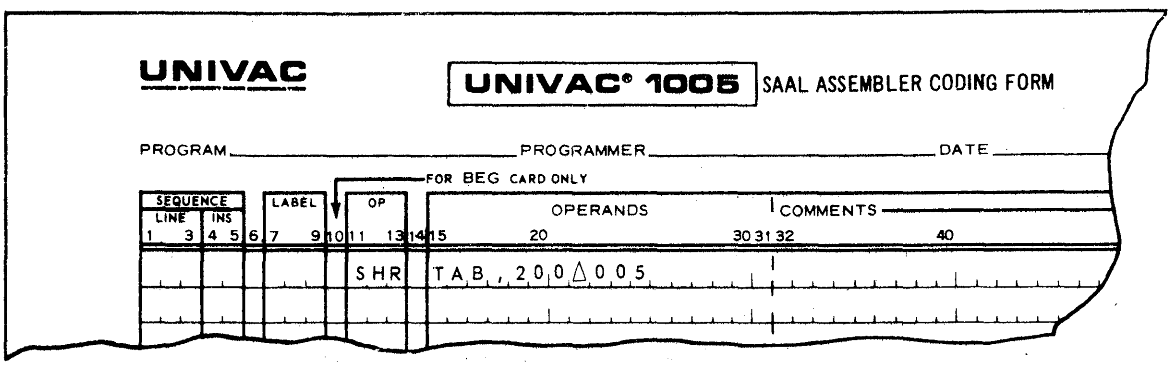
SHIFT RIGHT: SHR M,LΔS

Function: Shift the area in memory specified by M and L,S character positions Right.

Notes:

a.) L must be a decimal number less than 961 and wholly contained in one memory bank.
b.) The S least significant characters of the area are lost during the shift operation.
c.) The shift count S must be preceded by a space and must be a three digit decimal value, equal to or less than 30.
d.) Spaces will be stored in the S most significant character positions of the shift area.
e.) The memory location assigned to the least significant character of the area to be shifted must be a multiple of 31. In other words, it must terminate at the end of a row, i.e. 31, 62, 93 and so forth.

Example 1: Shift right an area of 200 characters labeled TAB five (5) characters or positions.


UNIVAC UNIVAC® 1005 SAAL ASSEMBLER CODING FORM

PROGRAM ______ PROGRAMMER ______ DATE ______

FOR BEG CARD ONLY

SEQUENCE LINE 1 3	INS 4 5	6	LABEL 7 9	10	OP 11 13	14	OPERANDS 15 20 30	COMMENTS 31 32 40
					SHR		TAB,200Δ005	

Example 2: Shift right an area of 63 characters labeled TAB three (3) characters or positions. The table contains 21 three character fields terminating in core location 0713.

MEMORY LAYOUT OF TABLE

0620																															A	21
0651	A	A	B	B	B	C	C	C	D	D	D	E	E	E	F	F	F	G	G	G	H	H	H	I	I	I	J	J	J	K	K	22
0682	K	L	L	L	M	M	M	N	N	N	O	O	O	P	P	P	Q	Q	Q	R	R	R	S	S	S	T	T	T	U	U	U	23
0713																																24

A three character field in the card labeled FDI is compared successively to each field in the table.

FOR BEG CARD ONLY

SEQUENCE LINE 1-3	INS 4-5	6	LABEL 7-9	10	OP 11-13	14	OPERANDS 15-31	COMMENTS 32-50
							IN DATA DIVISION	-LITERAL
			CTR		+5		21001	COUNTER
							IN PROCEDURE DIVI	SION
			ROU		CLR		CTR+2,2	CLEAR CTR
					LA1		TAB+60,3	TAB FIELD TO AR1
					CA1		FD1,3	COMP TO INPUT
					JEA		FIN	FIND IN TABLE
					IC		CTR	INCR CTR
					JE		ERR	NO FIND IN TABLE
					SHR		TAB,63 003	SHIFT TAB 3 POS
					SA1		TAB,3	STORE AT BEG
					J		ROU+5	REPEAT

SEQ. NO:

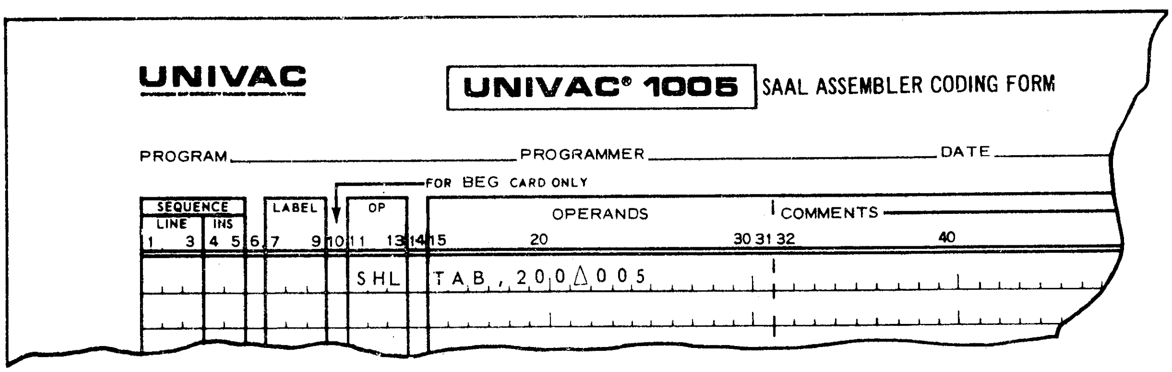
001 - The table counter is cleared
002 - Last field of table is loaded into AR1
003 - Compare AR1 to field in the card
004 - Jump equal to <u>FIN</u>
005 - Increment the table counter (21011)
006 - Jump equal to <u>ERR</u>
007 - Shift the table 3 positions clearing last field
008 - Restore last field at the beginning of table
009 - Jump to repeat routine (seq. No. 002)

SHIFT LEFT: SHL M,LΔS

Function: Shift the area in memory specified by M and L,S character positions left

Notes:
a.) L must be a decimal number less than 961 and wholly contained in one memory bank.
b.) The S most significant characters of the area are lost during the shift operation.
c.) The shift count S must be preceded by a space and must be a three digit decimal value, equal to or less than 30.
d.) Spaces will be stored in the S least significant character positions of the shift area.
e.) The memory location assigned to the most significant character of the area to be shifted must be a multiple of 31, plus 1. In other words, it must start at the beginning of a row, i.e. 32, 63, 94 and so forth.

Example 1: Shift right an area of 200 characters labeled TAB five (5) characters or positions.



Example 2: Shift left an area of 63 characters labeled TAB three (3) characters or positions. The table contains 21 three character fields starting in core position 0621.

MEMORY LAYOUT OF TABLE

0589																																20
0620	A	A	A	B	B	B	C	C	C	D	D	D	E	E	E	F	F	F	G	G	G	H	H	H	I	I	I	J	J	J	K	21
0651	K	K	L	L	L	M	M	M	N	N	N	O	O	O	P	P	P	Q	Q	Q	R	R	R	S	S	S	T	T	T	U	U	22
0682	U																															23
0713																																24

A three character field in the card labeled FDI is compared successively to each field in the table.

FOR BEG CARD ONLY

SEQUENCE LINE (1–3)	INS (4–5)	LABEL (7–9)	OP (11–13)	OPERANDS (15–31)	COMMENTS (32–)
				IN DATA DIVISION	-LITERAL
		CTR	+5	21001	COUNTER
				IN PROCEDURE DIVISION	
001		ROU	CLR	CTR+2,2	CLEAR CTR
002			LA1	TAB,3	TAB FIELD TO AR1
003			CA1	FDI,3	COMP TO INPUT
004			JEA	FIN	FIND IN TABLE
005			IC	CTR	INCR CTR
006			JE	ERR	NO FIND IN TABLE
007			SHL	TAB,63 003	SHIFT TAB 3 POS
008			SA1	TAB+60,3	STORE AT END
009			J	ROU+5	REPEAT

SEQ NO:

- 001 - The table counter is cleared
- 002 - 1st field of table is loaded into AR1
- 003 - Compare AR1 to the field in the card
- 004 - Jump equal to FIN
- 005 - Increment the table counter (21011)
- 006 - Jump equal to ERR
- 007 - Shift the table 3 positions, clearing 1st field
- 008 - Restore 1st field at end of table
- 009 - Jump to repeat routine (Seq. No. 002)

Example 3: Shift left an area of 63 characters labeled TAB twenty-one (21) characters or positions. The table contains 21 three character fields starting in core position 0621. A third of the table will be transferred to AR2 and the register will be shifted 7 times before the table is shifted in memory. The execution time will be reduced, but the number of instructions will increase from example 2.

FOR BEG CARD ONLY

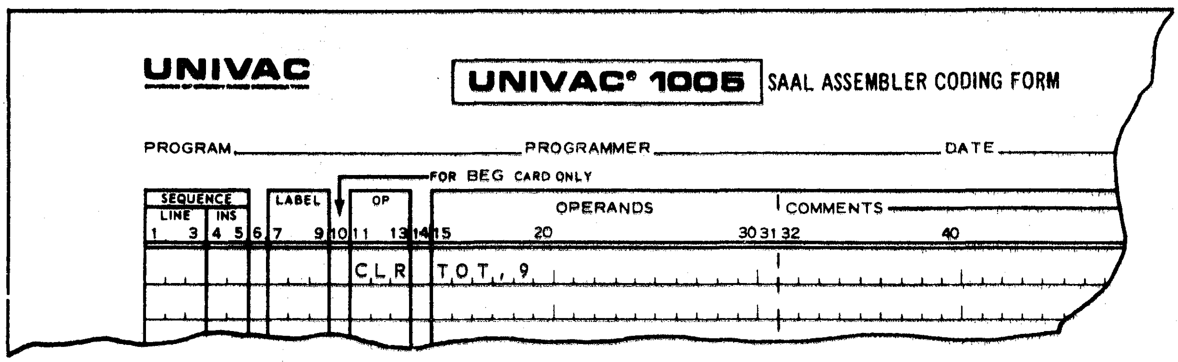
SEQUENCE LINE 1–3	INS 4–5	6	LABEL 7–9	10	OP 11–13	14	OPERANDS 15–30	COMMENTS 31–50
							IN DATA DIVISION	- LITERALS
			CT1		+5		04001	COUNTER 1
			CT2		+5		07001	COUNTER 2
							IN PROCEDURE DIVI	SION
001			ROU		LA1		FD1,3	INPUT FIELD TO AR1
002					CLR		CT1+2,2	CLEAR CT1
003					LA2		TAB,21	TAB FIELDS TO AR2
004					SHL		TAB,63 021	SHIFT TAB 21 POS
005					SA2		TAB+42,21	STORE AT END
006					IC		CT1	INCR CT1
007					JE		ERR	NO FIND IN TAB
008					CLR		CT2+2,2	CLEAR CTR 2
009			SUB		CA1		AR2,3	COMP TO TAB
010					JEA		FIN	FIND IN TAB
011					IC		CT2	INCR CT2
012					JE		ROU+10	REPEAT ROUTINE
013					LD2		AR2+3,18	SHIFT AR2 3 POS
014					J		SUB	REPEAT SUB

CLEAR: CLR M,L

Function: Clear L most significant positions of the field whose most significant character is at M.

Note: L must be a decimal number.

Example: Clear the first nine character positions of the accumulator called TOT.


UNIVAC

UNIVAC® 1005 SAAL ASSEMBLER CODING FORM

PROGRAM ____ PROGRAMMER ____ DATE ____

FOR BEG CARD ONLY

SEQUENCE LINE 1 3	INS 4 5	6	LABEL 7 9	10	OP 11 13	14	OPERANDS 15 20 30 31 32	COMMENTS 40
					CLR		TOT,,9	

TOT (before) = \$ Δ 1 0 0 , 0 0 -
TOT (after) = Δ Δ Δ Δ Δ Δ Δ Δ Δ

COMPARE ALPHA/NUMERIC: CA_r M,L

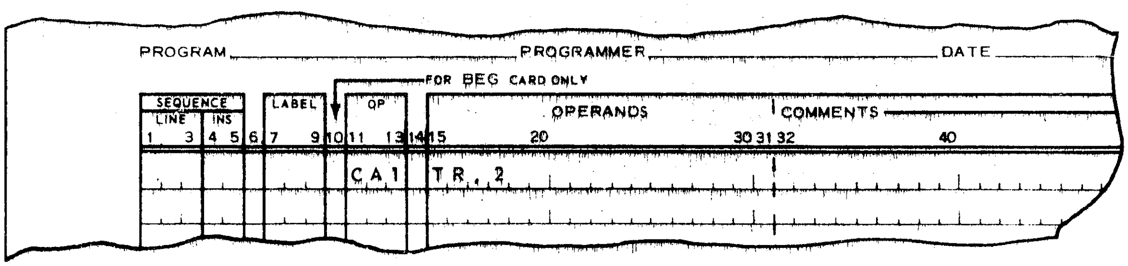
Function: Compare for equality L least significant character positions of AR1 or 2, to the L most significant characters of the field specified by M.

Notes:
a.) This is a binary comparison and all data bits are considered.
b.) L specifies the number of six (6) bit characters that will be compared.
c.) A maximum of 10 or 21 characters can be compared in AR1 and AR2 respectfully.
d.) The result of the comparison is recorded in testable indicators as follows:
Result of Comparison:

	JUA (UNEQUAL)	JEA (EQUAL)
(ARr) = (MEM)		SET
(ARr) ≠ (MEM)	SET	

Example:

. Compare the two least significant characters of AR1 against the two most significant characters of the field called TR.



*AR1	(before)	=	0	Δ	?	1	6	5	B	C	A	B
TR		=							A	B	C	D
AR1	(after)	=	0	Δ	?	1	6	5	B	C	A	B

Result: JEA (equal) indicator set.

. Compare the two least significant characters of AR1 against the two least significant characters of the field labeled TR.

UNIVAC   UNIVAC® 1005 SAAL ASSEMBLER CODING FORM

PROGRAM ______ PROGRAMMER ______ DATE ______

SEQUENCE LINE 1 3	INS 4 5	6	LABEL 7 9	10	OP 11 13	14	OPERANDS 15 20 30	COMMENTS 31 32 40
					CA1		TR+2,2	

*AR1 (before) = 0 Δ ? 1 6 5 B C A B
TR = A B C D
AR1 (after) = 0 Δ ? 1 6 5 B C A B

Result: JUA (unequal) indicator set.

. Compare the two least significant characters of AR1 against the 2nd and 3rd character of the field labeled TR.

PROGRAM ______ PROGRAMMER ______ DATE ______

SEQUENCE LINE 1 3	INS 4 5	6	LABEL 7 9	10	OP 11 13	14	OPERANDS 15 20 30	COMMENTS 31 32 40
					CA1		TR+1,2	

*AR1 (before) = Δ Δ 0 Δ ? 1 6 5 B C
TR = A B C D
AR1 (after) = Δ Δ 0 Δ ? 1 6 5 B C

Result: JEA (equal) indicator set.

*The functions indicated are identical for AR2 with the exception that larger fields can be compared.

COMPARE NUMERIC: CNr M,L

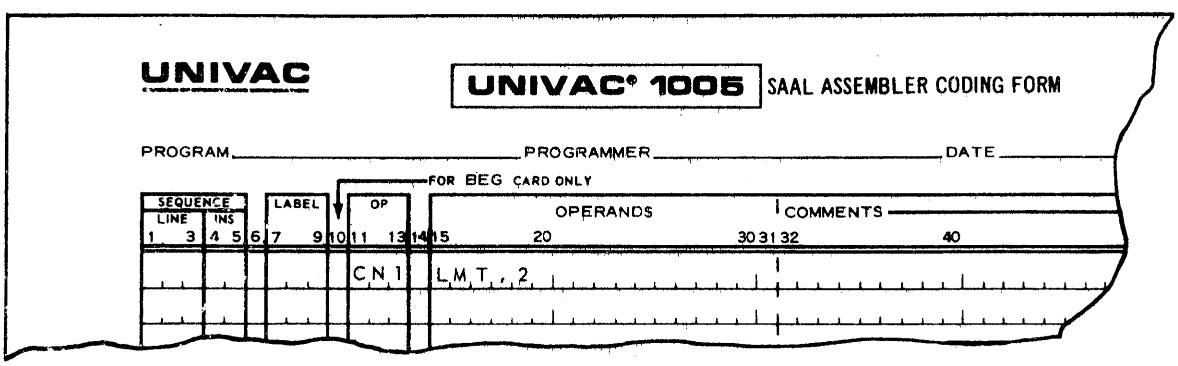
Function: Compare algebraically L least significant characters of a signed number in AR1 or 2, to the L most significant characters of a signed numeric field specified by M.

Notes:
a.) If the two fields have unlike signs, the comparison is terminated immediately and the proper indicator set.
b.) If L is greater than the capacity of the register spaces are assumed in the implied high order positions of the register.
c.) The comparison terminates when all L characters at M have been compared.
d.) Only the numeric bits are compared.
e.) The results of the algebraic comparison is stored in testable indicators as follows:

Results of Comparison:

	JE (Equal)	JG (Greater)	JL (Less)
(ARr) > (MEM)		SET	
(ARr) < (MEM)			SET
(ARr) = (MEM)	SET		

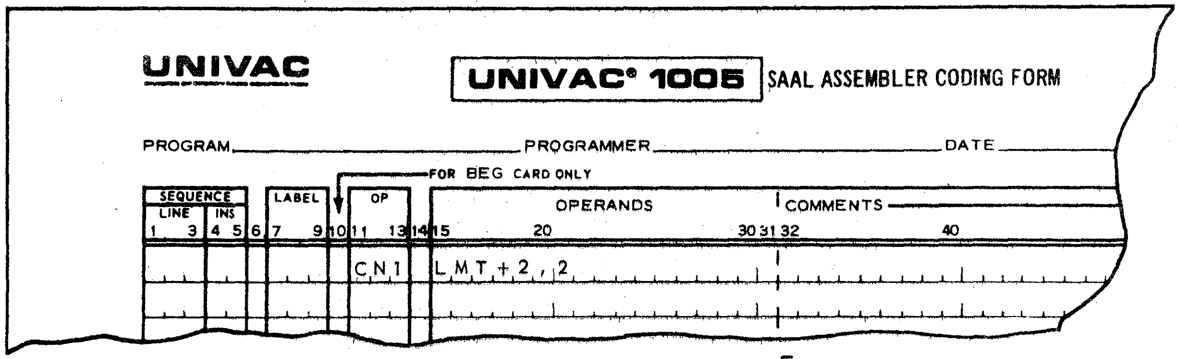
Example: Compare the two least significant characters of AR1 against the two most significant characters of the field called LMT.



$$
\begin{array}{llcccccccccc}
\text{*AR1 (before)} & = & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \bar{0} \\
\text{LMT} & = & & & & & & & 0 & 0 & 1 & \bar{0} \\
\text{AR1 (after)} & = & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \bar{0}
\end{array}
$$

Result: JL (less than) indicator set

. Compare the two least significant characters of AR1 against the two least significant characters of the field called LMT.


UNIVAC | UNIVAC® 1005 | SAAL ASSEMBLER CODING FORM

PROGRAM ____ PROGRAMMER ____ DATE ____

FOR BEG CARD ONLY

SEQUENCE LINE 1 3	INS 4 5	6	LABEL 7 9	10	OP 11 13	14	OPERANDS 15 20 30 31	COMMENTS 32 40
					CN1		LMT+2,2	

*AR1 (before) = 0 0 0 0 0 0 0 0 1 $\bar{0}$
LMT = 0 0 1 $\bar{0}$
AR1 = 0 0 0 0 0 0 0 0 1 $\bar{0}$

Result: JE (equal) indicator set

*The functions indicated are identical for AR2 with the exception that larger fields can be compared.

INCREMENT AND COMPARE: IC M

Function: Increment a two digit (2) counter whose most significant character is at M+2 by a decimal value store at M+4. Compare the result to a two digit limit whose most significant character is at M.

Notes:
a.) The field specified by M must be five characters in length.
b.) The two most significant positions of the field specified by M contain the limit, the next two positions contain the count and the last position contains the increment.
c.) The sub-functions of the instruction are as follows:
1. The increment stored at M+4 is added to the count stored at M+2 and M+3.
2. The result is compared numerically against the predetermined limit stored at M and M+1.
3. The results of the comparison are recorded in the testable indicators.

Example: Determine by means of the IC instruction if the page line counter labeled CTR has been incremented fifty four times. If the condition is present branch to a sub-routine labeled OFL for page compensation.

PROGRAM ______ PROGRAMMER ______ DATE ______

FOR BEG CARD ONLY

SEQUENCE LINE (1-3)	INS (4-5)	6	LABEL (7-9)	10	OP (11-13)	14	OPERANDS (15-31)	COMMENTS (32-40)
					IC		CTR	
					JE		OFL	
							(MAIN PROGRAM)	
			OFL		CLR		CTR+2,2	

The first increment of the counter:

CTR (before) = 5 4 0 0 1
CTR (after) = 5 4 0 1 1

The fifty-fourth increment of the counter:

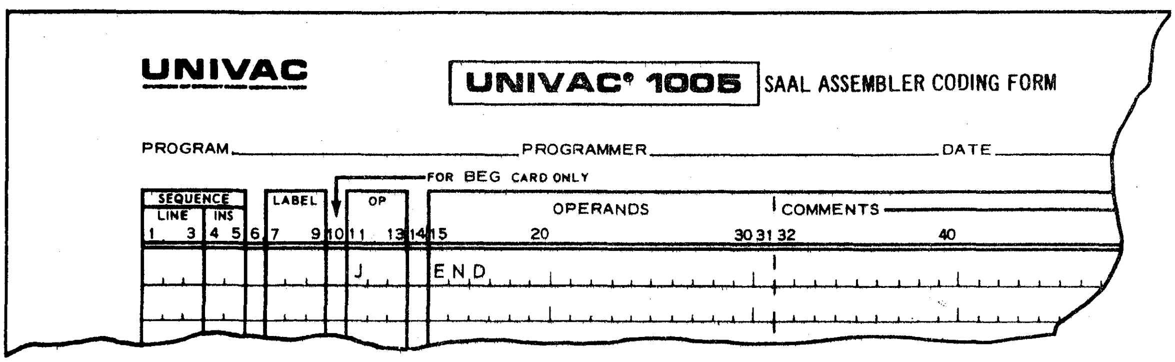
CTR (before) = 5 4 5 3 1
CTR (after) = 5 4 5 4 1

Control is then transferred to the routine labeled 'OFL' where the increment counter is cleared and page compensation is performed by the programmer.

JUMP: J M

Function: Transfer program control to the instruction stored at M.

Example: . Transfer program control to the routine labeled END.



JUMP IF GREATER: JG M

JUMP IF LESS: JL M

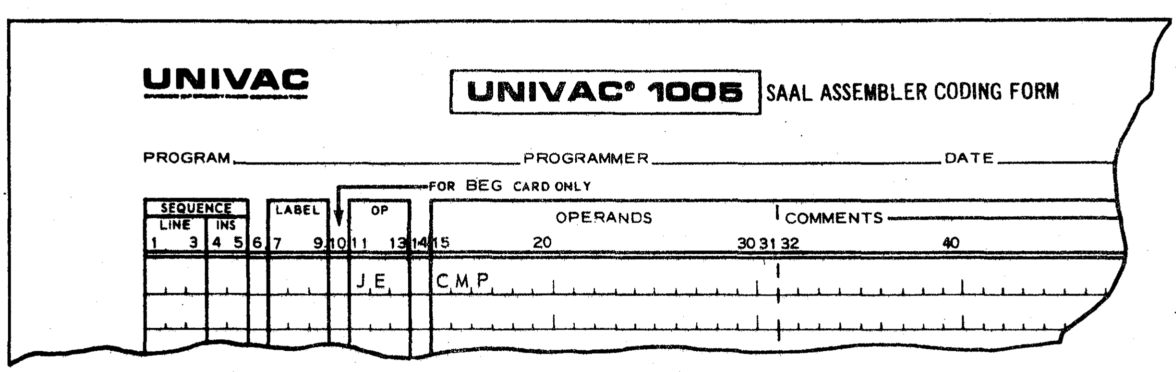
JUMP IF EQUAL: JE M

Function: Transfer program control to the instruction stored at M if the numeric comparison indicator specified by the operation is set.

Notes: a.) These instructions are used to test the result of a numeric comparison, (CNr).

b.) If the condition tested is not present, control will not be transferred and the next instruction in the testing sequence will be executed.

Example: A numeric comparison instruction has been executed. If the equal indicator is set transfer control to the routine labeled CMP.



JUMP EQUAL (ALPHA/NUMERIC): JEA M

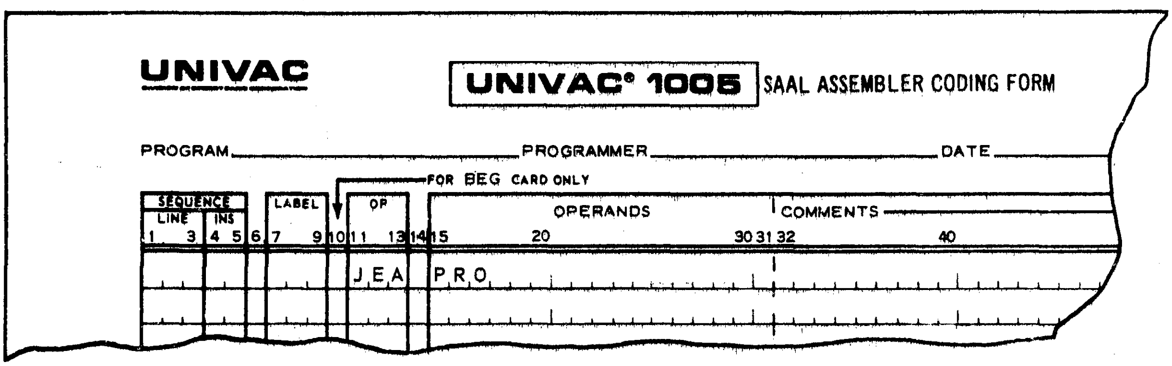
JUMP UNEQUAL (ALPHA/NUMERIC): JUA M

Function: Transfer program control to the instruction stored at M if the comparison indicator specified by the operation code is set.

Notes: a.) These instructions are used to test the results of an alpha/numeric comparison. (CAr)

b.) If the condition tested is not present control will not be transferred and the next instruction in the testing sequence will be executed.

Example: Test the alpha/numeric indicators in order to determine the results of a previous alpha/numeric compare. If the arguments were equal transfer control to the routine labeled PRO.


UNIVAC

UNIVAC® 1005 SAAL ASSEMBLER CODING FORM

PROGRAM PROGRAMMER DATE

FOR BEG CARD ONLY

SEQUENCE LINE 1 3	INS 4 5	6	LABEL 7 9	10	OP 11 13	14	OPERANDS 15 20 30	COMMENTS 31 32 40
					JEA		PRO	

JUMP POSITIVE: JP M

JUMP NEGATIVE: JN M

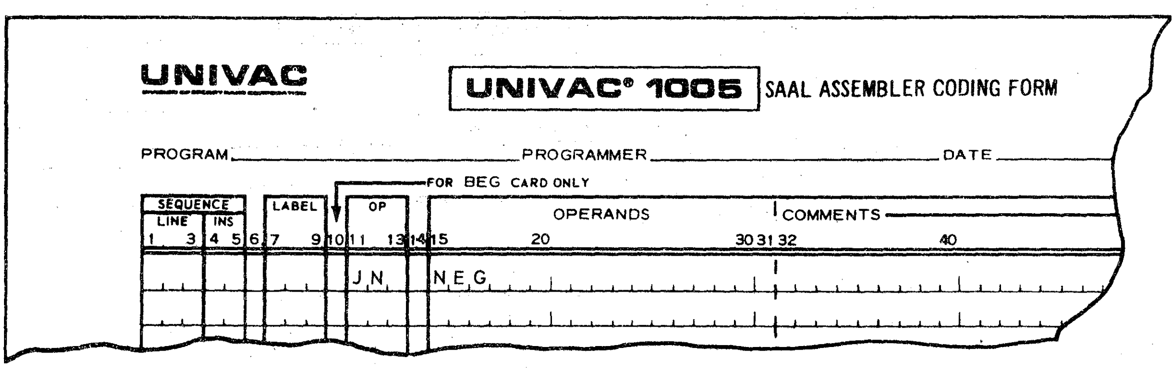
JUMP ZERO: JZ M

Function: Transfer program control to the instruction stored at M if the arithmetic indicator specified by the operation code is set.

Notes: a.) These instructions are used to test the resultant sign of an arithmetic operation (AMr, ARr, SMr, SRr).

b.) If the condition tested is not present control will not be transferred and the next instruction in the testing sequence will be executed.

Example: . Test arithmetic indicators in order to determine if the result of a previous arithmetic operation was negative. If the condition is true, transfer control to the routine labeled NEG.



JUMP RETURN: JR M

Function: This instruction stores the address of the next sequential instruction in the X register and transfers program control to the instruction stored at M.

Notes:

a.) This instruction provides the programmer with the facility of breaking program sequence and executing a subroutine; and then returning program control to the instruction immediately following the JR instruction.

b.) The subroutine at M must contain a JX instruction so that the return line to the main program can be established.

Example: Transfer program control to an initialized sub-routine called INT, perform those functions required, and return control to the main program.

PROGRAM ______ PROGRAMMER ______ DATE ______

FOR BEG CARD ONLY

SEQUENCE LINE (1–3)	INS (4–5)	6	LABEL (7–9)	10	OP (11–13)	14	OPERANDS (15–30)	COMMENTS (31–40)
			TAG		JR		INT	
					CLR		TOT,35	

PROGRAM ______ PROGRAMMER ______ DATE ______

FOR BEG CARD ONLY

SEQUENCE LINE (1–3)	INS (4–5)	6	LABEL (7–9)	10	OP (11–13)	14	OPERANDS (15–30)	COMMENTS (31–40)
			INT		JX		EX	

PROGRAM ______ PROGRAMMER ______ DATE ______

FOR BEG CARD ONLY

SEQUENCE LINE (1–3)	INS (4–5)	6	LABEL (7–9)	10	OP (11–13)	14	OPERANDS (15–30)	COMMENTS (31–40)
			EX		J		TAG+5	

Note: Reference function of JX instruction.

JUMP RETURN EXIT: JX M

Function: This instruction creates a jump instruction to the address specified by the X Register and stores it at M.

Notes: a.) This instruction is used in conjunction with the Jump Return (JR) instruction in order to establish the return link to the main program from a given sub-routine.

b.) This instruction is normally executed as the first instruction in a called sub-routine.

Example: . Establish the exit line back to the main program for an initialize sub-routine called INT.

LABEL	OP	OPERANDS
TAG	JR	INT
	CLR	TOT,35

LABEL	OP	OPERANDS
INT	JX	EX

LABEL	OP	OPERANDS
EX	J	TAG+5

ADD TO MEMORY: AM_r M,L

Function: Adds algebraically L least significant characters of AR1 or 2, to the L most significant characters of the field specified by M.

Notes: a.) If the length of the Register is equal to or greater than L, the instruction is terminated when L characters have been added to memory.

b.) If the length of Register is less than L, decimal zeroes are added to memory.

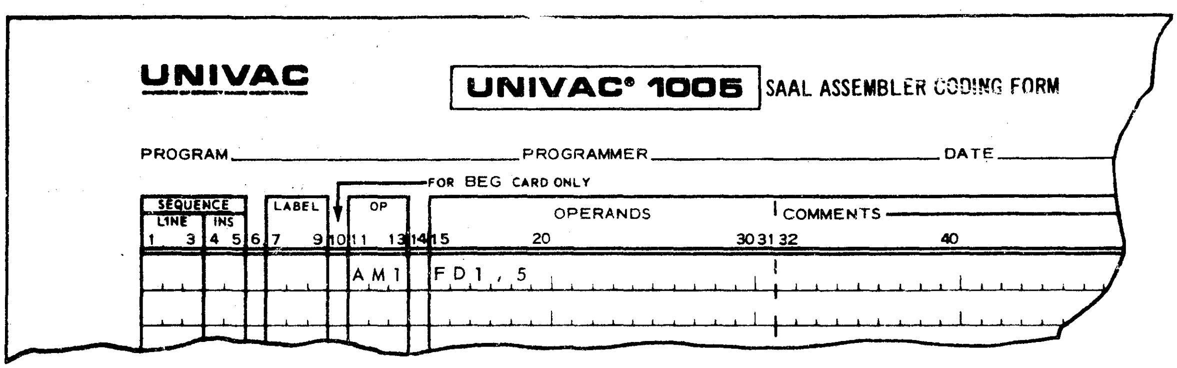
c.) Except for the sign bit, zone bits are ignored in the Register.

d.) The results of an Arithmetic instruction are recorded in testable indicators as follows:

If the sum is plus (+), the positive indicator is set.

If the sum is negative (-), the negative indicator is set.

Examples: . Add the 5 least significant characters of Arithmetic Register one (AR1) to the field labeled FDI.


UNIVAC

UNIVAC® 1005 SAAL ASSEMBLER CODING FORM

PROGRAM ______ PROGRAMMER ______ DATE ______

FOR BEG CARD ONLY

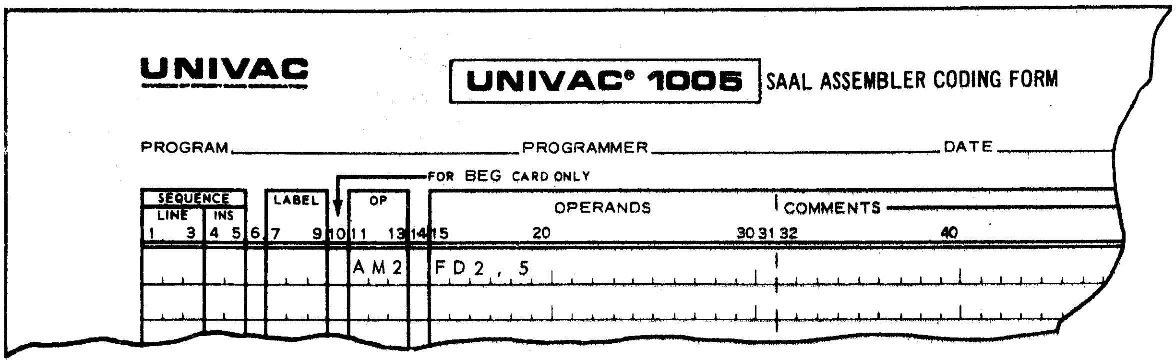
SEQUENCE LINE 1 3	INS 4 5	6	LABEL 7 9	10	OP 11 13	14	OPERANDS 15 20 30	COMMENTS 31 32 40
					AM1		FD1, 5	

ARI (before & after) = 1 2 3 0 0 0 4 7 1 6

FDI (before) = 5 2 3 0 1

FDI (after) = 5 7 0 1 7

Add the 5 least significant characters of arithmetic register 2 to the field labeled FD2.



AR2 (before & after) = 0 ← 0320

FD2 (before) = ΔΔ472

FD2 (after) = 00792

Special consideration should be given on all arithmetic processes (AR, AM, SR, SM) to the fact that when a negative result is developed the sign indications (X bits) will be generated in both the most and least significant locations of the resultant field. When a zero result is developed the zero balance indicator (Y bit) will be generated in the most significant location of the resultant field. A zero balance cannot be tested for sign (+ or -) through the use of testable indicators. All testable indicators remain set until another compare, add, subtract or print (if alt switch two is on/illuminated).

ADD TO REGISTER: AR_r, M,L

Function: Adds algebraically L most significant characters of the field specified by M, to the L least significant characters of AR1 or 2.

Notes: a.) If the length of the Register is greater than L, decimal zeroes are added to the Register.

b.) If the length of the Register is equal to or less than L, the instruction is terminated when L characters have been added to the Register.

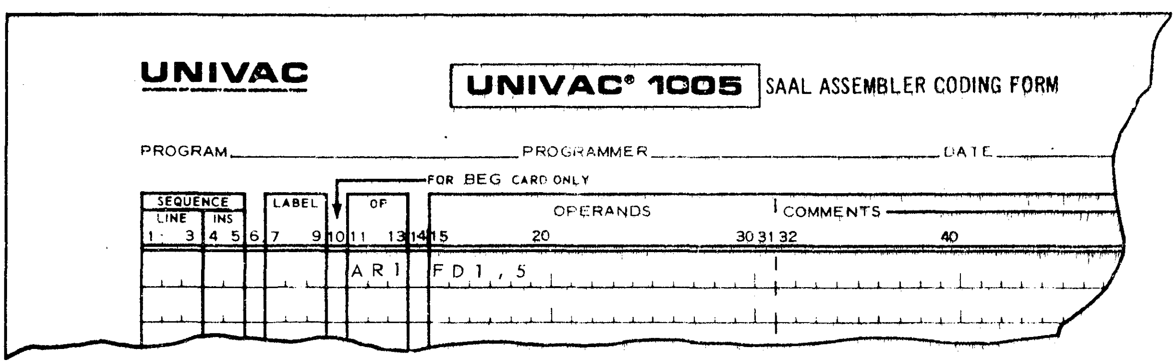
c.) Except for the sign bit, zone bits are ignored in memory.

d.) The results of an Arithmetic instruction are recorded in testable indicators as follows:

If the sum is plus (+), the positive indicator is set.

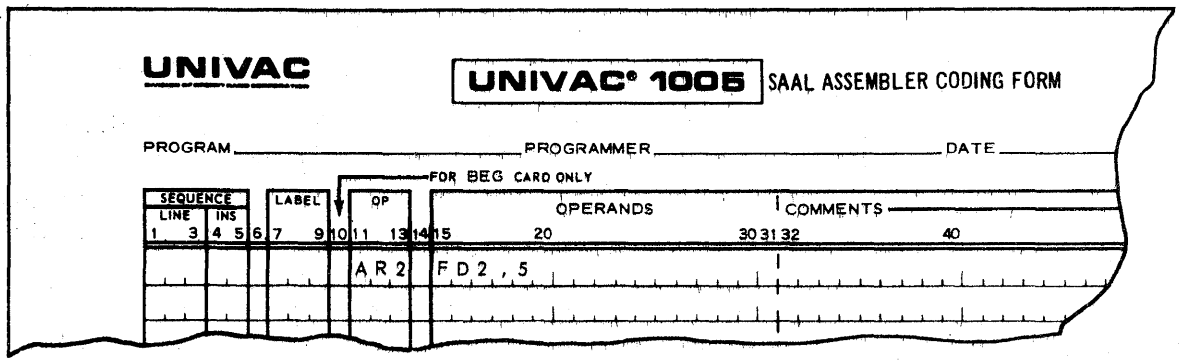
If the sum is negative (-), the negative indicator is set.

Examples: . Add the five digit field labeled FD1 to Arithmetic Register One (AR1).



FD1 (before & after)	=					0	0	2	5	3
AR1 (before)	=	Δ	Δ	Δ	Δ	Δ	0	5	6	2	3
AR1 (after)	=	0	0	0	0	0	0	5	8	7	6

. Add the five digit negative field labeled FD2 to arithmetic register 2.


UNIVAC

UNIVAC® 1005 SAAL ASSEMBLER CODING FORM

PROGRAM ____ PROGRAMMER ____ DATE ____

SEQUENCE LINE 1 3	INS 4 5	6	LABEL 7 9	10	OP 11 13	14	OPERANDS 15 20 30 31	COMMENTS 32 40
					AR2		FD2,5	

FD2 (before & after) = $\bar{0}$ 0 1 2 $\bar{7}$

AR2 (before) = Δ ⟵ Δ 7 6 9 5 4 6

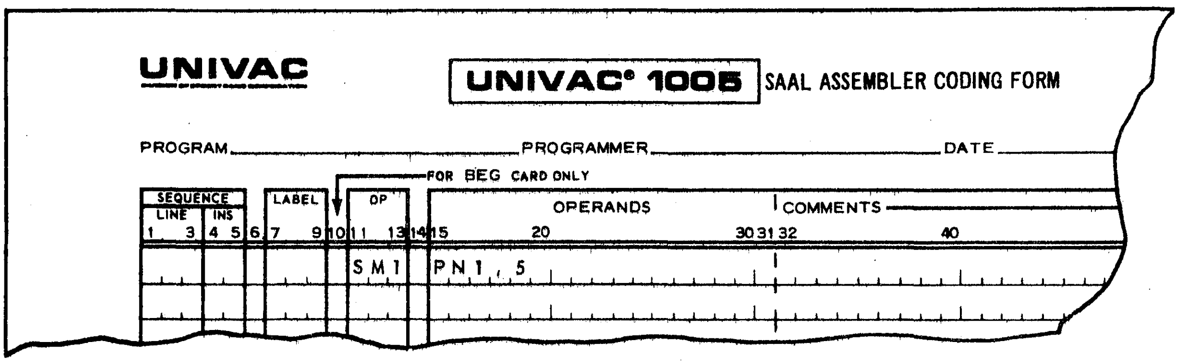
AR2 (after) = 0 ⟵ 0 7 6 9 4 1 9

SUBTRACT FROM MEMORY: SM_r M,L

Function: Subtracts algebraically L least significant characters of AR1 or 2, from the L most significant characters of the field specified by M.

Note: This instruction operates identically to the AM instruction, except that the operation is subtraction. Otherwise the notes under the AM instruction apply.

Examples: . Subtract the 5 least significant characters of AR1 from the field labeled PN1.

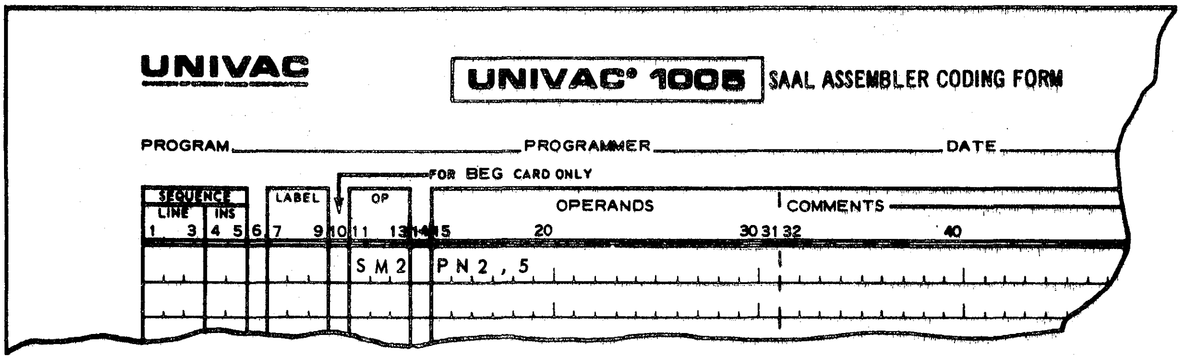


AR1 (before & after) = Δ Δ Δ Δ Δ Δ 1 9 7 6

PN1 (before) = 3 9 8 7 8

PN1 (after) = 3 7 9 0 2

. Subtract the 5 least significant characters of AR 2 from the field labeled PN2.


UNIVAC

UNIVAC® 1005 SAAL ASSEMBLER CODING FORM

PROGRAM ____ PROGRAMMER ____ DATE ____

FOR BEG CARD ONLY

SEQUENCE LINE 1 3	INS 4 5	6	LABEL 7 9	10	OP 11 13	14	OPERANDS 15 20 30 31	COMMENTS 32 40
					S M 2		P N 2 , 5	

AR2 (before & after) = Δ ← Δ 6 9 3 7

PN2 (before) = 0 6 0 0 0

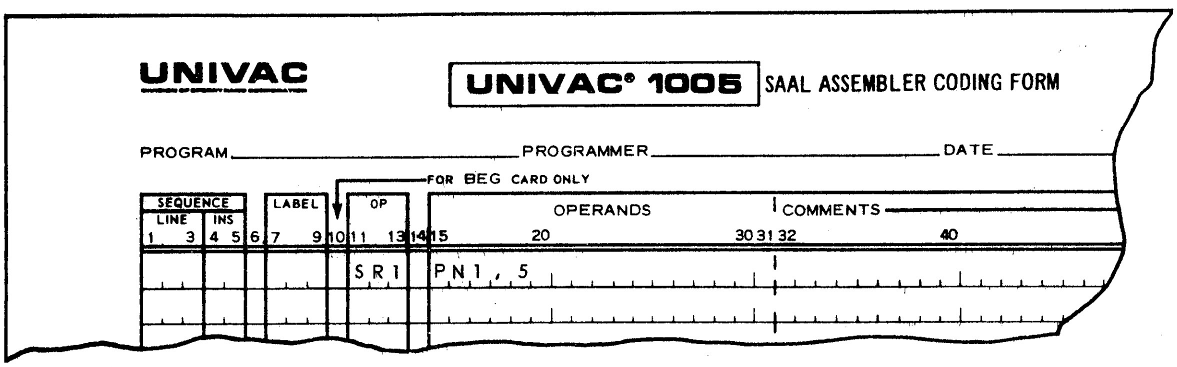
PN2 (after) = $\bar{0}$ 0 9 3 $\bar{7}$

SUBTRACT FROM REGISTER: SR_r M,L

Function: Subtracts algebraically L most significant characters of the field specified by M, from the L least significant characters of AR1 or 2.

Note: This instruction operates identically to the AR instruction, with the sole exception that the operation is a subtraction. Otherwise the notes under the AR instruction apply.

Examples: . Subtract the 5 digit field labeled PN1 from Arithmetic Register one (AR1).


UNIVAC
UNIVAC® 1005 SAAL ASSEMBLER CODING FORM
PROGRAM ____ PROGRAMMER ____ DATE ____
FOR BEG CARD ONLY

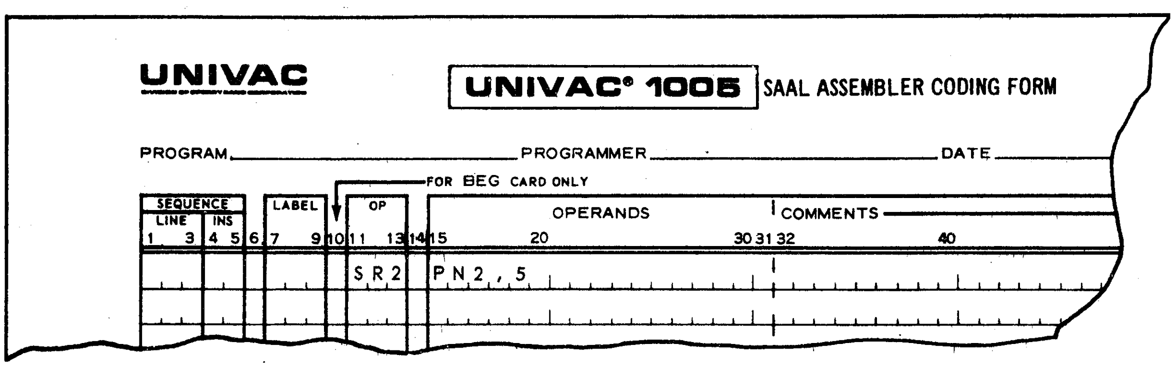
SEQUENCE LINE 1 3	INS 4 5	6	LABEL 7 9	10	OP 11 13	14	OPERANDS 15 20 30	COMMENTS 31 32 40
					S R 1		P N 1 , 5	

PN1 (before & after) = 0 0 6 5 3

AR1 (before) = Δ Δ Δ Δ Δ Δ Δ 9 5 7

AR1 (after) = 0 0 0 0 0 0 0 3 0 4

. Subtract the 5 digit field labeled PN2 from arithmetic register 2.


UNIVAC
UNIVAC® 1005 SAAL ASSEMBLER CODING FORM
PROGRAM ____ PROGRAMMER ____ DATE ____
FOR BEG CARD ONLY

SEQUENCE LINE 1 3	INS 4 5	6	LABEL 7 9	10	OP 11 13	14	OPERANDS 15 20 30	COMMENTS 31 32 40
					S R 2		P N 2 , 5	

PN2 (before & after) = 7 6 5 6 0

AR2 (before) = Δ ← Δ 7 6 0 6 0

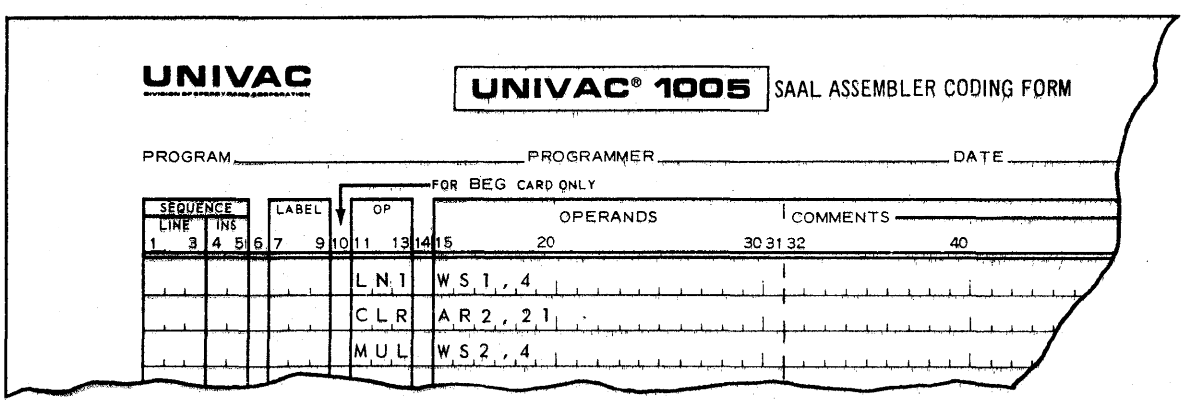
AR2 (after) = $\overline{0}$ ← 0 5 0 $\overline{0}$

MULTIPLICATION: MUL M,L

Function: Multiply L most significant characters of the field specified by M by the value previously stored in AR1 and place the product in AR2.

Notes:
a.) L must be a decimal number ranging from one to eight.
b.) The multiplier must be previously stored in AR1 and must be less than ten digits in length and have sign deleted.
c.) AR2 must be cleared to spaces before the Multiplication instruction is executed.
d.) Both the Multiplier (AR1) and the Multiplicand (MEM) must be positive values.
e.) A maximum product of 17 decimal digits can be developed.
f.) The result is formed in AR2 and is right justified with zero fill.
g.) Testable indicators are not set or affected by this instruction.

Example: Multiply two four digit numbers labeled WS1 and WS2.



WS1	=	0 1 6 5
AR1 (before)	=	Δ ← Δ 0 1 6 5
AR2 (before)	=	Δ ← → Δ
WS2	=	1 0 2 5
AR2 (after)	=	0 ← 0 1 6 9 1 2 5

DIVISION: DIV M,L

Function: Divide AR2 by the L most significant characters of the field specified by M and place the results in AR1 and 2.

Notes:
a.) L must be a decimal number ranging from one to seven.
b.) The dividend must be previously stored in AR2 and must be less than thirteen digits in length. If signed, sign must be deleted.
c.) AR1 must be cleared to spaces before the Division instruction is executed.
d.) Both the divisor (MEM) and the dividend (AR2) must be positive values, subsequently testable indicators are not set or affected by this instruction.
e.) Seven whole numbers are developed as the quotient and will appear in AR1 right justified. That is if the length of the dividend is greater than 7,there must be less than 9,999,999 difference in the absolute values of the dividend and the divisor.
f.) Eight decimal and nine remainder of the quotient are developed and will appear in AR2 left justified.
g.) If the divisor is zero, the result will be blank.

Example: Divide WS1 3 digits into WS2 (5 digits).

UNIVAC

UNIVAC® 1005 SAAL ASSEMBLER CODING FORM

PROGRAM ______ PROGRAMMER ______ DATE ______

SEQUENCE LINE 1–3	INS 4–5	6	LABEL 7–9	10	OP 11–13	14	OPERANDS 15	COMMENTS
					LN2		WS2,5	
					CLR		AR1,1Ø	
					DIV		WS1,3	

WS1	=	1 2 6
WS2	=	5 5 3 1 6
AR1 (before)	=	← Δ
AR2 (before)	=	0 ← 0 5 5 3 1 6
AR1 (after)	=	0 0 0 0 0 0 0 4 3 9
AR2 (after)	=	0 1 5 8 7 3 0 1 0 0 0 0 0 0 0 7 4 0 0 0 0

$\underbrace{0\ 1\ 5\ 8\ 7\ 3\ 0\ 1}_{\text{Decimal Remainder}}\ \underbrace{0\ 0\ 0\ 0\ 0\ 0\ 0\ 7\ 4}_{\text{Quotient Remainder}}\ 0\ 0\ 0\ 0$

TRANSLATE INTRODUCTION

The Translate Process for the UNIVAC 1005 permits the translation of an entire record to be accomplished by a single instruction.

The Translate Instruction functions, quite simply:

All of the characters of the translated code are entered into Core Storage in the form of a reference table (Translate Table) at or before the start of a run.

The bits of each character of the code to be translated, acting as address codes, call the translated character code out of the Translate Table during the Translate Instruction.

The translated codes substitute themselves for the codes to be translated in the M (Operand) Address of the Translate Operation. This leaves a fully translated record in the M Address locations at the end of the operation.

The UNIVAC 1005 uses a code when addressing its Core Storage. The Address Control recognizes the code for the original character and relates this with a specific storage location containing the translate character.

With practically all of the codes used in data processing, be they 5-, 6-, 7-, or 8-Track, a maximum of six tracks are valid or significant as far as character code formation is concerned. The other tracks serve for parity or functional control purposes.

By using six significant tracks (or levels) of the code to be translated for address control, one level for Row Address control and the other levels for Column Address control, the UNIVAC 1005 Translate Process is practically universal in its application to code translation.

To change from one translation to another can require nothing more than changing the translation table in the storage.

The Translate Process combines simplicity of programming with efficiency of operation to obtain a wide scope of translating abilities.

GENERAL DESCRIPTION OF THE TRANSLATION TABLE

Figure 1 illustrates the required format of the Translation Table insofar as it is determined by the 1005 circuitry, and is intended to give a correct approach to the planning of the table. Figure 1-A is a sample chart,

ORIG. CHAR.		X	Y	8	4	2	1	MOD 1 MEM. LOC.		NEW CHAR.
	→	0	1	1	1	1	1	0081	→	
	→	1	1	1	1	1	1	0082	→	
	→	0	0	0	0	0	0	0094	→	
	→	0	0	0	0	0	1	0095	→	
	→	0	0	0	0	1	1	0096	→	
	→	0	0	0	1	1	1	0097	→	
	→	0	0	1	1	1	0	0098	→	
	→	0	1	1	1	0	0	0099	→	
	→	0	1	1	0	0	1	0100	→	
	→	0	1	0	0	1	0	0101	→	
	→	0	0	0	1	0	0	0102	→	
	→	0	0	1	0	0	0	0103	→	
	→	0	1	0	0	0	1	0104	→	
	→	0	0	0	0	1	0	0105	→	
	→	0	0	0	1	0	1	0106	→	
	→	0	0	1	0	1	0	0107	→	
	→	0	1	0	1	0	1	0108	→	
	→	0	0	1	0	1	1	0109	→	
	→	0	1	0	1	1	1	0110	→	
	→	0	0	1	1	1	1	0111	→	
	→	0	1	1	1	1	0	0112	→	
	→	0	1	1	1	0	1	0113	→	
	→	0	1	1	0	1	1	0114	→	
	→	0	1	0	1	1	0	0115	→	
	→	0	0	1	1	0	1	0116	→	
	→	0	1	1	0	1	0	0117	→	
	→	0	1	0	1	0	0	0118	→	
	→	0	0	1	0	0	1	0119	→	
	→	0	1	0	0	1	1	0120	→	
	→	0	0	0	1	1	0	0121	→	
	→	0	0	1	1	0	0	0122	→	
	→	0	1	1	0	0	0	0123	→	
	→	0	1	0	0	0	0	0124	→	

ORIG. CHAR.		X	Y	8	4	2	1	MOD 1 MEM. LOC.		NEW CHAR.
	→	1	0	0	0	0	0	0125	→	
	→	1	0	0	0	0	1	0126	→	
	→	1	0	0	0	1	1	0127	→	
	→	1	0	0	1	1	1	0128	→	
	→	1	0	1	1	1	0	0129	→	
	→	1	1	1	1	0	0	0130	→	
	→	1	1	1	0	0	1	0131	→	
	→	1	1	0	0	1	0	0132	→	
	→	1	0	0	1	0	0	0133	→	
	→	1	0	1	0	0	0	0134	→	
	→	1	1	0	0	0	1	0135	→	
	→	1	0	0	0	1	0	0136	→	
	→	1	0	0	1	0	1	0137	→	
	→	1	0	1	0	1	0	0138	→	
	→	1	1	0	1	0	1	0139	→	
	→	1	0	1	0	1	1	0140	→	
	→	1	1	0	1	1	1	0141	→	
	→	1	0	1	1	1	1	0142	→	
	→	1	1	1	1	1	0	0143	→	
	→	1	1	1	1	0	1	0144	→	
	→	1	1	1	0	1	1	0145	→	
	→	1	1	0	1	1	0	0146	→	
	→	1	0	1	1	0	1	0147	→	
	→	1	1	1	0	1	0	0148	→	
	→	1	1	0	1	0	0	0149	→	
	→	1	0	1	0	0	1	0150	→	
	→	1	1	0	0	1	1	0151	→	
	→	1	0	0	1	1	0	0152	→	
	→	1	0	1	1	0	0	0153	→	
	→	1	1	1	0	0	0	0154	→	
	→	1	1	0	0	0	0	0155	→	

FIGURE 1.

ORIG. CHAR.		X	Y	8	4	2	1	MOD 1 MEM. LOC.		NEW CHAR.
=	→	0	1	1	1	1	1	0081	→	”
)	→	1	1	1	1	1	1	0082	→	ǂ
	→	0	0	0	0	0	0	0094	→	
]	→	0	0	0	0	0	1	0095	→	1

FIGURE 1-A

filled in, to illustrate a possible input translation to the 1005.

Fig. 1 represents the sixty-four (64) characters that are recognized by the 1005. These characters are shown in the table by bit configurations. Zero represents a bit absent and 1 represents a bit present. Therefore, the programmer must have a six level code showing the bit configuration for each letter, number or special character:

X	Y	8	4	2	1	
Bit Abs	Bit Pres	Bit Abs	Bit Pres	Bit Abs	Bit Abs	= A (in XS-3, 80 col.)

In the context of the translation instruction, this pattern has two meanings:

Meaning 1: It is the pattern of a character in the original code as it appears in 1005 storage before translation.

Meaning 2: This is the code that Address Control recognizes to relate to a specific storage location containing the translate character.

Since the bit patterns are arranged by the sequence as addresses, they are in no meaningful sequence as original code characters.

The Original Character box will contain the character that is equal to the bit configuration shown directly to the right on the same line:

	Orig. Char.										
(BCD)	$\boxed{J}$ →	1	0	0	0	0	1	$\boxed{0126}$ →	$\boxed{J}$	(XS3)

The Mod. 1 Mem. Loc. box refers to the location in memory that will contain the new character. Note that the translate table is a fixed area in Module 1 with two characters at location 0081 and 0082 and sixty-two (62) more characters starting at location 0094 and continuing to 0155. This corresponds with the layout of the translation tables in that entry 126 of the table, ($\boxed{0126}$ → $\boxed{J}$) a J will be entered in position 0126 of memory.

PLANNING THE TRANSLATION TABLE (Ref. Figure 1-A)

To construct the translation table, the first step is to examine the bit patterns of the character to be translated. Having found the bit configuration in the table (under Bit Configuration of Orig. Character) write the character to be translated in the small box at the left. Next, fill in the corresponding small box on the right (under New Char.) with the resultant UNIVAC 1005 character desired.

Loading the translate table into memory is easily accomplished in the data division of the program. Recommended procedure is to define the areas in CRD, PRT, PCH. Immediately follow this with ORG 0081 to set the Address Control to the beginning of the translate table. Next, code a literal instruction with +2 in the operation field and two characters in the operand field. These two characters will be the first two entries under NEW CHAR. corresponding to 0081 and 0082. Note: The use of the literal instruction directs the assembler to move the number of characters specified in the Op field from the operand field to sequential core locations starting at 0081. It is now necessary to reposition Address Control to the next position of the translate table. This is accomplished with an ORG 0094. Next, code a literal instruction with +31 in the operation field and 31 characters in the operand field. These characters are found under NEW CHAR. corresponding to 0094 thru 0124. Next, code another literal instruction with +31 in the operation field and 31 characters in the operand field. These characters are found under NEW CHAR. corresponding to 0125 thru 0155. This completes the coding necessary and upon execution of loading the program the translate table will be properly positioned in memory. Following is the data division of a sample program showing the necessary coding for a translation from BCD to XS-3.

```
        Beg
        CRD
FD1     -       1
FD2     -       7
        PRT
PR1     -       1
PR2     -       7
        PCH
PC1     -       1
PC2     -       7
        ORG     0081
        +2      ; (
        ORG     0094
        +31     137⊐%ZS48/25ϕV#X:⊏<,W\≠U9T6@Y
        +31     -JLP!⌑IBMQAKNΔE$G≥+?.F=)DRCO*H&
        ORG     0373
        STA
```

This chart and its explanation cover the needs of translation into BCD. It is simple to punch the translation characters into a card and load it into the 1005 table area. For translating into foreign codes, it is necessary to load the bit patterns of the foreign code into the table. Further planning is needed to determine the proper card punching to obtain these bit patterns.

TRANSLATE: TRL M,L

Function: Replace L most significant characters of the field specified by M with their positional equivalent as dictated by the translate table.

Notes: a.) L must be a decimal number less than 961. The entire operand must be located in the 1st bank.

b.) Any combination of 64 possible U1005 6 bit characters can appear in the translation table.

c.) Prior to executing the translate instruction the translate table is stored in memory locations 0081, 0082, 0094 - 0155.

d.) The M expression specifies the most significant location of the field to be translated. The conversion proceeds from the most significant character to the least for L characters.

e.) The TRL instruction replaces each character in the field to be translated with a character selected from the translate table. The basis for selecting the replacement character is the Boolean value of the character to be replaced.

f.) The contents of the translate table are not altered by the instruction, unless the translate table itself is translated.

Example: A three character field containing three 6 bits configurations 110001 110010 110011 is labeled FD1. Those 6 bit configurations are the BCD (Binary Coded Decimal) codes for the characters ABC. FD1 is to be printed on the U1005 and must be translated from BCD to UNIVAC 1005 XS-3 code. The first four instructions load the new translate table.

UNIVAC

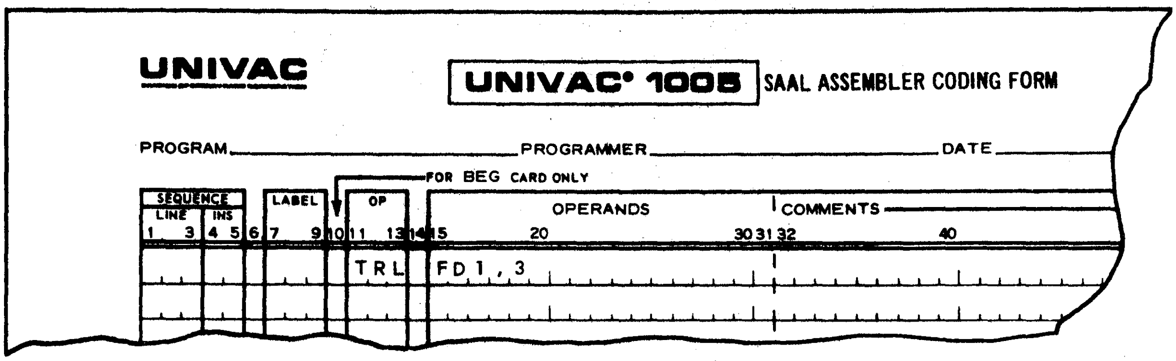
UNIVAC® 1005 SAAL ASSEMBLER CODING FORM

PROGRAM ______ PROGRAMMER ______ DATE ______

FOR BEG CARD ONLY

SEQUENCE LINE 1 3	INS 4 5	6	LABEL 7 9	10	OP 11 13	14	OPERANDS 15 20 30	COMMENTS 31 32 40
					LPR		K1,62	
					SPR		TR1,62	
					LA1		K2,2	
					SA1		TR2,2	

The translate function is now executed.


UNIVAC

UNIVAC® 1005 SAAL ASSEMBLER CODING FORM

PROGRAM ________ PROGRAMMER ________ DATE ________

FOR BEG CARD ONLY

SEQUENCE LINE 1–3	INS 4–5	6	LABEL 7–9	10	OP 11–13	14	OPERANDS 15–30	COMMENTS 31–
					TRL		FD1,3	

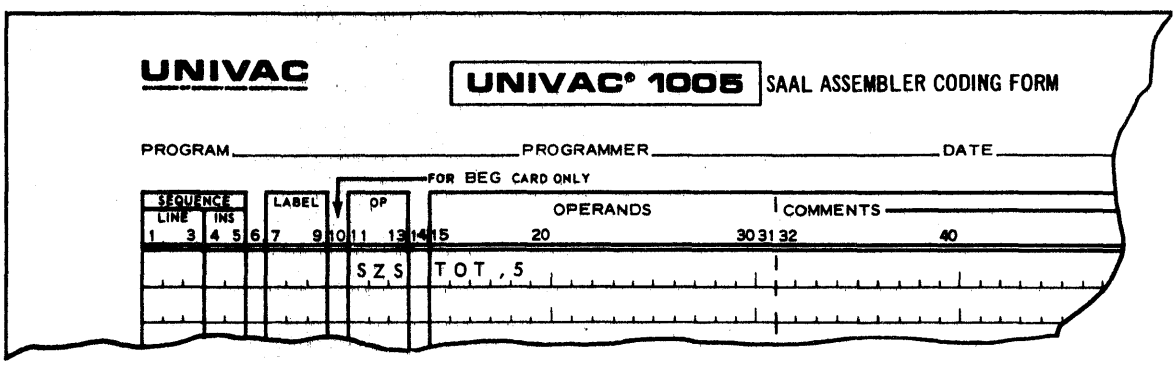
The resultant characters stored in FD1 are the XS-3 equivalent for the alpha character ABC.

STORE ZERO SUPPRESSED: SZS M,L

Function: Store ascending L least significant characters from AR2 into the L most significant characters of the field specified by M suppressing all leading zeroes.

Notes:
a.) L must be a decimal number.
b.) L characters are transferred in ascending order (least to most) from AR2 to the most significant positions of the field specified by M.
c.) Zero suppressing will continue until some character other than a zero or space is decoded.
d.) When L is greater than the capacity of AR2 the receiving field will be space filled.

Example: Store the five least significant positions of AR2, suppressing all leading zeros, into the field labeled TOT.



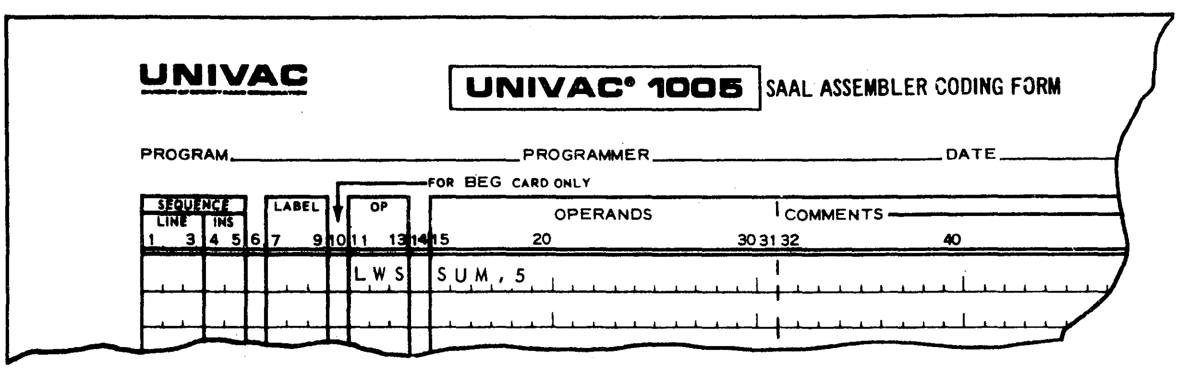
AR2 (before) = 0 ⟵ 0 0 0 1 5
TOT (after) = Δ Δ Δ 1 5
AR2 (after) = Δ ⟵ Δ Δ Δ 1 5

LOAD WITH SIGN: LWS M,L

Function: Load ascending L most significant numeric characters from the field specified by M, into the L least significant character positions of AR2.
Insert a sign in the LSL position of AR2 on the basis of the low-order "X".

Notes:
a.) L must be a decimal number.
b.) L most significant characters of the field specified by M are transferred in ascending order to the L least significant positions of AR2.
c.) The LSL position of AR2 is examined and a sign is inserted. If the value in AR2 is positive it is left shifted one position and a space (plus sign) is inserted in the least significant character of AR2. If the value in AR2 is negative it is left shifted one position and a minus (negative sign) is inserted in the least significant character of AR2.
d.) When L is less than the capacity of AR2 the remaining positions of the register are space filled.
e.) When L is greater than the capacity of the register truncation will occur and the most significant characters of the field will be deleted.
f.) All non-numeric bits are deleted by this instruction.

Example: . Load a five digit negative field called SUM into AR2 inserting a sign in the LSL character of AR2 based on the presence or absence of the low - order "X" bit.


UNIVAC

UNIVAC® 1005 SAAL ASSEMBLER CODING FORM

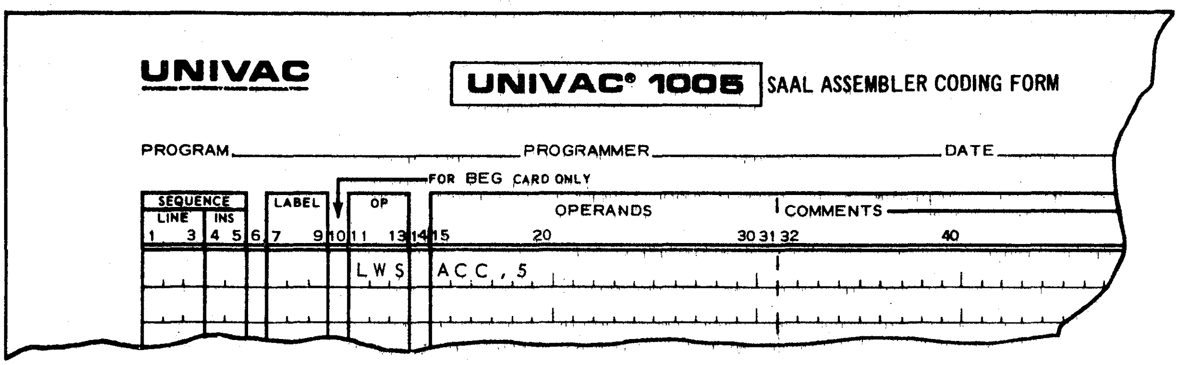
PROGRAM ________ PROGRAMMER ________ DATE ________

FOR BEG CARD ONLY

SEQUENCE LINE 1 3	INS 4 5	6	LABEL 7 9	10	OP 11 13	14	OPERANDS 15 20 30	COMMENTS 31 32 40
					L W S		S U M , 5	

$$\begin{array}{lcl} \text{SUM} & = & \bar{0}\;6\;0\;1\;\bar{5} \\ \text{AR2} & = & \Delta \longleftarrow \Delta\;0\;6\;0\;1\;5\;- \end{array}$$

. Load a five digit positive field call ACC into AR2 inserting a sign in the LSL character of AR2 based on the presence or absence of the Low - order "X" bit.


UNIVAC

UNIVAC® 1005 SAAL ASSEMBLER CODING FORM

PROGRAM ______ PROGRAMMER ______ DATE ______

FOR BEG CARD ONLY

SEQUENCE LINE 1 3	INS 4 5	6	LABEL 7 9	10	OP 11 13	14	OPERANDS 15 20 30	COMMENTS 31 32 40
					LWS		ACC,5	

ACC = 0 5 0 1 5

AR2 = Δ ← Δ 0 5 0 1 5 Δ

LOAD NUMERICS: LN_r M,L

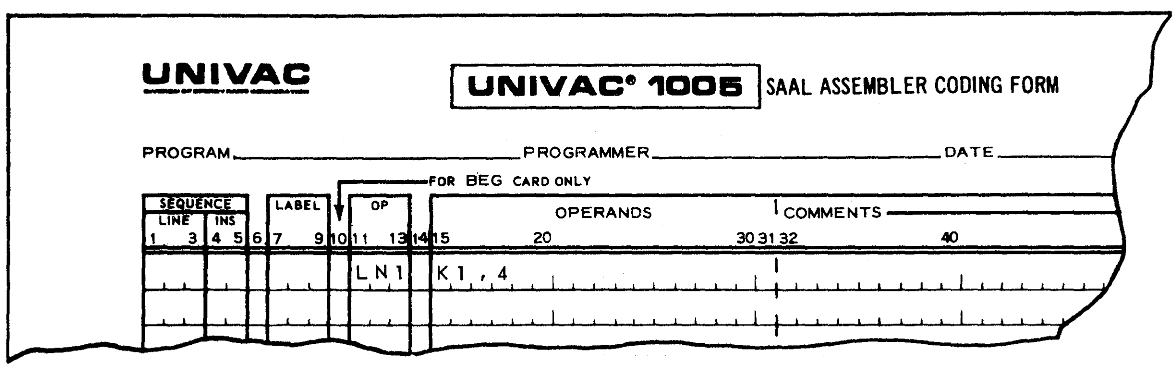
Function: Load ascending L most significant characters from the field specified by M, into the L least significant characters of AR1 or 2. During the transfer all zone bits are changed to binary zeroes.

Notes: a.) L can be a decimal number ranging from 1 to 21 depending upon which AR has been specified by the operation code.

b.) If a field contains less characters than the register capacity the remaining positions of the register will be space filled.

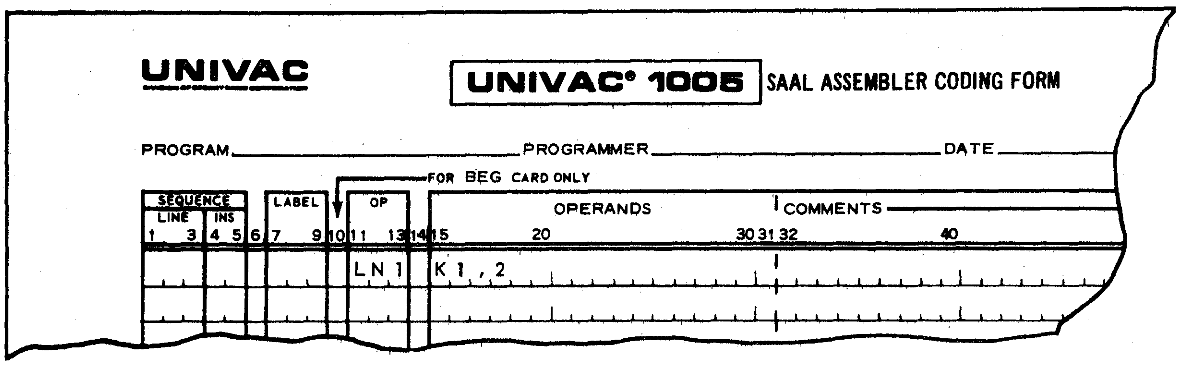
c.) If a field contains more characters than register capacity the surplus positions will be truncated.

Examples: Transfer a four character constant K1 into the four least significant positions of AR1.



*AR1 (before)	=	Δ	Δ	0	0	1	3	4	5	6	7
K1								A	B	C	D
AR1 (after)	=	Δ	Δ	Δ	Δ	Δ	Δ	1	2	3	4

Transfer a two character constant from K1 into the two least significant positions of AR1.


UNIVAC

UNIVAC® 1005 SAAL ASSEMBLER CODING FORM

PROGRAM PROGRAMMER DATE

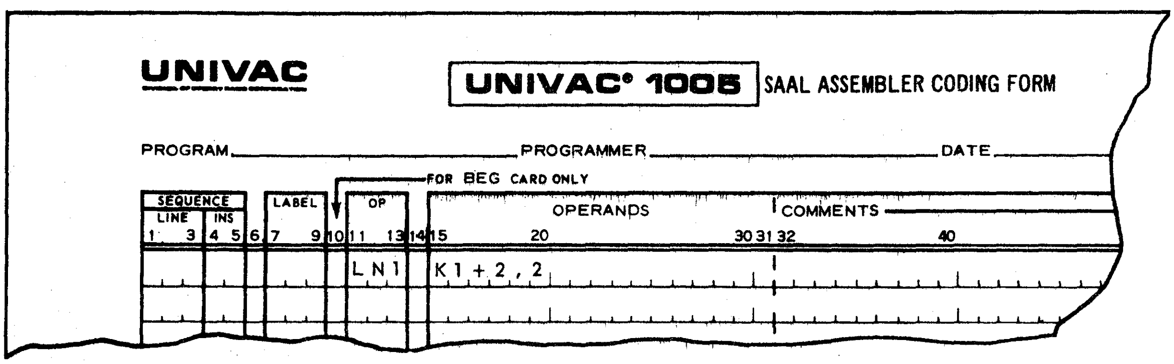
FOR BEG CARD ONLY

SEQUENCE LINE 1 3	INS 4 5	6	LABEL 7 9	10	OP 11 13	14	OPERANDS 15 20 30	COMMENTS 31 32 40
					LN1		K1,2	

*AR1 (before)	=	Δ	Δ	0	0	1	3	4	5	6	7
K1	=							A	B	C	D
AR1 (after)	=	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	1	2

Since the most significant position of the field is the character specified by the address M, the two most significant characters of K1 were transferred.

Transfer a two character constant from K1 beginning with LSL character into the two least significant positions of AR1.


UNIVAC

UNIVAC® 1005 SAAL ASSEMBLER CODING FORM

PROGRAM PROGRAMMER DATE

FOR BEG CARD ONLY

SEQUENCE LINE 1 3	INS 4 5	6	LABEL 7 9	10	OP 11 13	14	OPERANDS 15 20 30	COMMENTS 31 32 40
					LN1		K1+2,2	

*AR1 (before)	=	Δ	Δ	0	0	1	3	4	5	6	7
K1	=							A	B	C	D
AR1 (after)	=	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	3	4

Since the most significant position of the field is the character specified by the address M + 2, the two least significant characters of K1 were transferred.

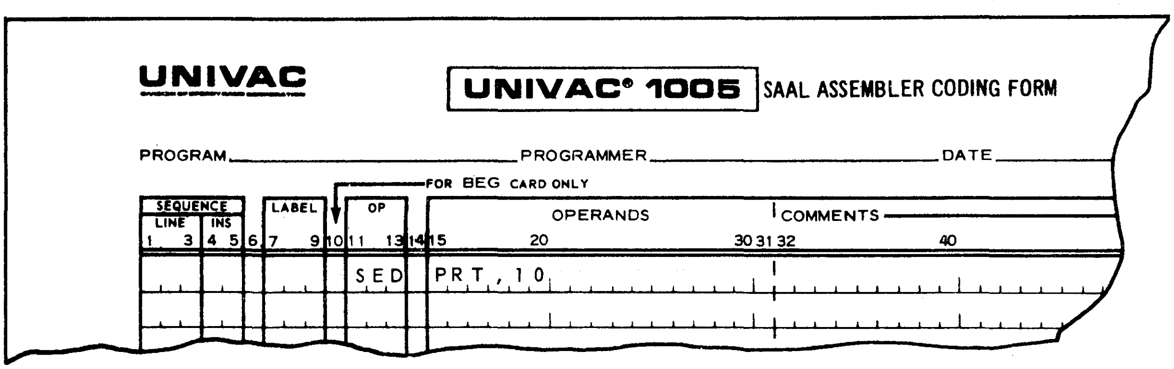
*The functions indicated are identical for AR2 with the exception that larger fields can be manipulated.

STORE EDITED: SED M,L

Function: Store ascending L least significant characters from AR2 into the L most significant positions of the field specified by M. Suppress all leading zeroes and edit the field according to a fixed pattern.

Notes:
- a.) L must be a decimal number.
- b.) L characters are transferred in ascending order (least to most) from AR2 to the L most significant positions of the field specified by M.
- c.) The field will be zero suppressed until some character other than a zero or space is decoded.
- d.) A period is inserted in the fourth least significant position of AR2.
- e.) Commas are inserted for separating significant values when they exist. If the integer value of the field is less than 1,000 commas will not be inserted.
- f.) The rules for truncation and space fill are the same as for store ascending.

Example: Store AR2 edited into the print-buffer.


UNIVAC

UNIVAC® 1005 SAAL ASSEMBLER CODING FORM

PROGRAM ______ PROGRAMMER ______ DATE ______

SEQUENCE LINE 1 3	INS 4 5	6	LABEL 7 9	10 (FOR BEG CARD ONLY)	OP 11 13	14	OPERANDS 15 20	COMMENTS 30 31 32 40
					SED		PRT,10	

AR2 (before)	=	Δ←——Δ	0	1	4	0	0	1	5	-	
PRT (before)	=	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ
AR2 (after)	=	Δ←—Δ	1	,	4	0	0	.	1	5	-
PRT (after)	=	Δ	1	,	4	0	0	.	1	5	-

PUNCH TEST: PTE

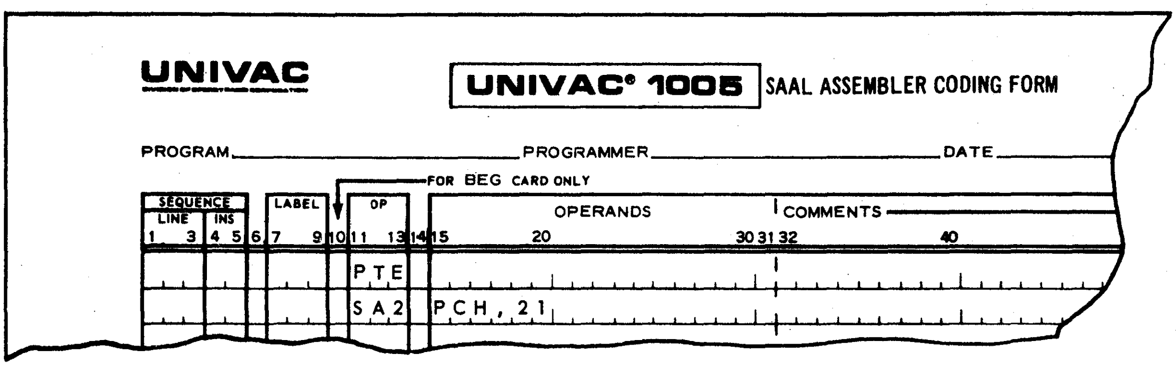
Function: This instruction tests the ready status of the Punch Unit. Control will not be transferred to the next instruction in sequence if the Punch Unit is still active.

Notes: a.) This instruction is normally given following a Punch command (XF PUN) and prior to the first transfer of new data into the Punch-buffer.

b.) This instruction insures that information will not be transferred into the Punch-buffer while it is in the process of unloading.

c.) Optimum utilization of the Punch-Test instruction will provide the maximum overlap of processing with punching.

Example: Test the Punch before storing AR2 in the Punch-buffer.


UNIVAC

UNIVAC® 1005 SAAL ASSEMBLER CODING FORM

PROGRAM PROGRAMMER DATE

FOR BEG CARD ONLY

SEQUENCE LINE 1 3	INS 4 5	6	LABEL 7 9	10	OP 11 13	14	OPERANDS 15 20 30	COMMENTS 31 32 40
					PTE			
					SA2		PCH, 21	

B. INSTRUCTION REPERTOIRE -- CARD SYSTEM EXTERNAL FUNCTIONS

The UNIVAC 1005 Card Processing system has been designed around a single address, internal programmed processor and includes as secondary units the following:

- Integrated High Speed Printer
- Integrated or free standing Card Reader
- Free standing Punch Unit or Read/Punch Unit
- Optional free standing Auxiliary Reader

The Card System External Function instructions pertain to Class III and are explained in detail on the following pages.

Class III: Class III instructions are Input/Output or External Function Commands, and contain a mnemonic code in the "M" portion of an instruction indicating the I/O device or devices to be initiated.

READ CARD: XF ΔREA

Function: This instruction reads a full 80 column card into the U1005 input Card-buffer.

Notes: a.) The input Card-buffer area is 80 locations in length, beginning with memory location 1 through memory location 80.

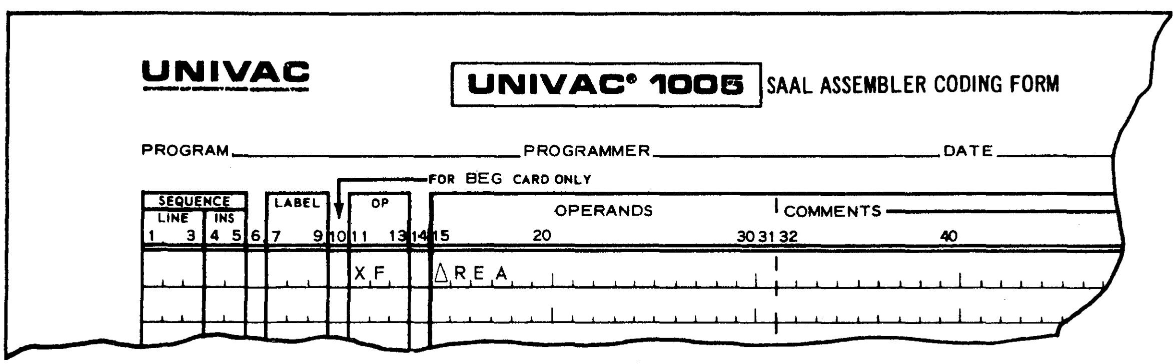
b.) Input Card-buffer locations correspond to card columns, thus a character punched in column 1 will be stored in location 1, a character punched in column 2 will be stored in location 2 and so on.

c.) As each column is read it is automatically translated from Hollerith card code to XS - 3.

d.) The mnemonic operand field must be preceded by a space.

(For illustration purposes this space will be indicated by a Δ for all XF instructions)

Example: Read a card from the Main Reader.



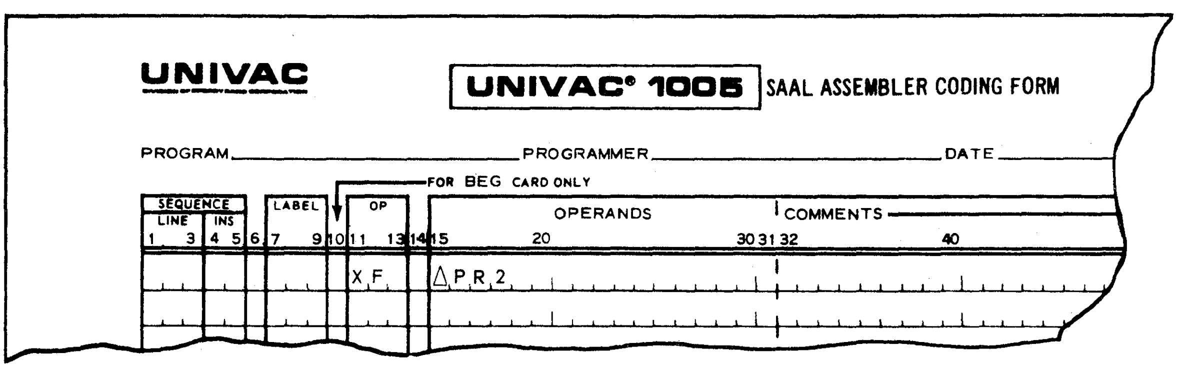
PRINT-SPACE ONE XF ΔPR1

TWO XF ΔPR2

Function: This instruction prints the contents at the Print-buffer and spaces the paper one or two lines depending on the numeric modifier specified.

Notes:
a.) The Print-buffer area is 132 locations in length, beginning with memory location 161 through memory location 292.
b.) Print-buffer locations correspond to printing positions, thus, a character stored in memory location 161 will be printed at print position one, a character stored in memory location 162 will be printed at printed position two; and so forth.
c.) The Print-buffer area is automatically cleared to spaces following the execution of each Print command.
d.) All Printer spacing occurs subsequent to printing, or in other words the contents of the Print-buffer is printed, the Print-buffer is cleared and then the printer form is advanced.
e.) The mnemonic operand field must be preceded by a space.

Example: Print the contents at the Print-buffer and advance the form two lines.


UNIVAC

UNIVAC® 1005 SAAL ASSEMBLER CODING FORM

PROGRAM ____ PROGRAMMER ____ DATE ____

FOR BEG CARD ONLY

SEQUENCE LINE 1 3	INS 4 5	6	LABEL 7 9	10	OP 11 13	14	OPERANDS 15 20 30	COMMENTS 31 32 40
					XF		ΔPR2	

With Alt Switch 2 on/illuminated on all print commands an automatic ejection (skip 7) occurs when a one (1) punch is detected in the forms control tape. This condition is testable. A JG condition is set if the one (1) punch has not been detected. A JL condition is set when the one (1) punch has been detected. These settings remain testable until another card, print or paper tape I/O command, compare, add or subtract instruction is executed.

PRINT - ADVANCE 7 XF ΔPR7

Function: This instruction prints the contents at the Print-buffer and advances the paper until a one, two, four, punch is detected in the control loop.

Notes: a.) The Print-buffer area is 132 locations in length, beginning with memory location 161 through memory location 292.

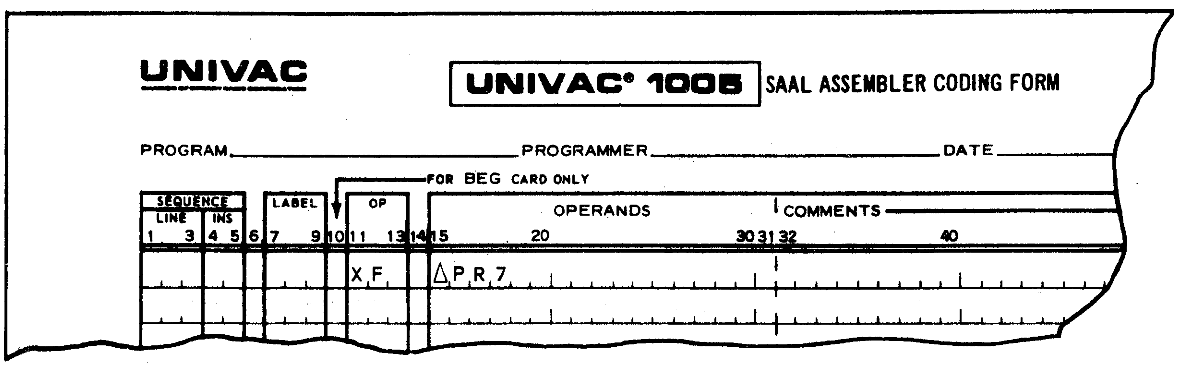
b.) Print-buffer locations correspond to printing positions, thus a character stored in memory location 161 will be printed at print position one, a character stored in memory location 162 will be printed at print position two, and so forth.

c.) The Print-buffer area is automatically cleared to spaces following the execution of the print command.

d.) Once the forms advance function of the PR7 instruction is initiated, control is returned to the next instruction in sequence and further processing is overlapped during the actual form advancing.

e.) The first line of a form is normally indicated by a control punch in all channels of the printer control loop. Hence, an advance 7 would mean advance the form to the 1st line of the next page.

f.) The mnemonic operand field must be preceded by a space.



PUNCH: **XF ΔPUN**

Function: This instruction punches the 80 column card image in the Punch-buffer.

Notes:

a.) The Punch-buffer area is 80 locations in length, beginning with memory location 293 through memory location 372.

b.) Punch-buffer locations correspond to card columns, thus a character stored in location 293 will be punched in card column 1, a character stored in 294 will be punched in card column 2 and so on.

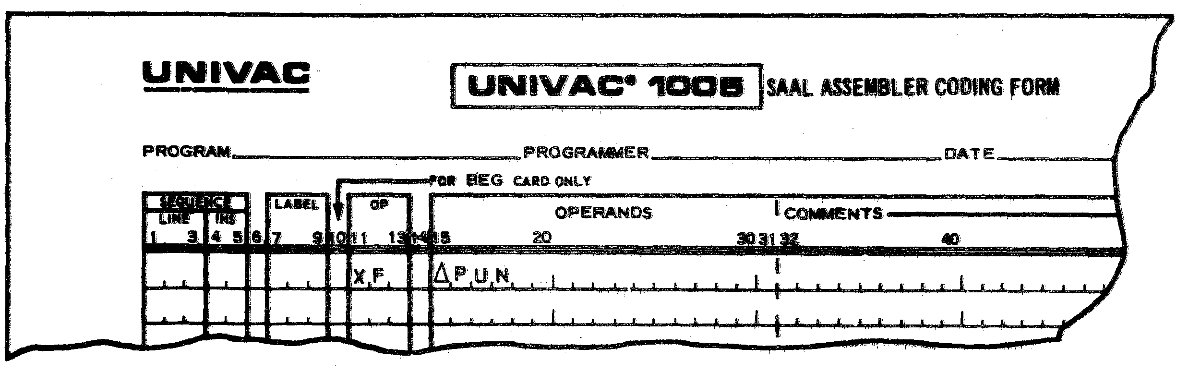
c.) The Punch-buffer is not cleared following the execution of the punch instruction.

d.) Once the punch cycle has been initiated, control is returned to the next instruction in sequence and further processing is overlapped during the punch-cycle.

e.) As each column is punched it is automatically translated from XS-3 code to Hollerith card code.

f.) The mnemonic operand field must be preceded by a space.

Example: Punch the card image stored in the Punch-buffer.



READ - PRINT - SPACE ONE: XF ΔRPR

Function: This instruction reads a full 80 column card into the U1005 input Card-buffer, prints the contents of the Print-buffer and advances the printer form one line.

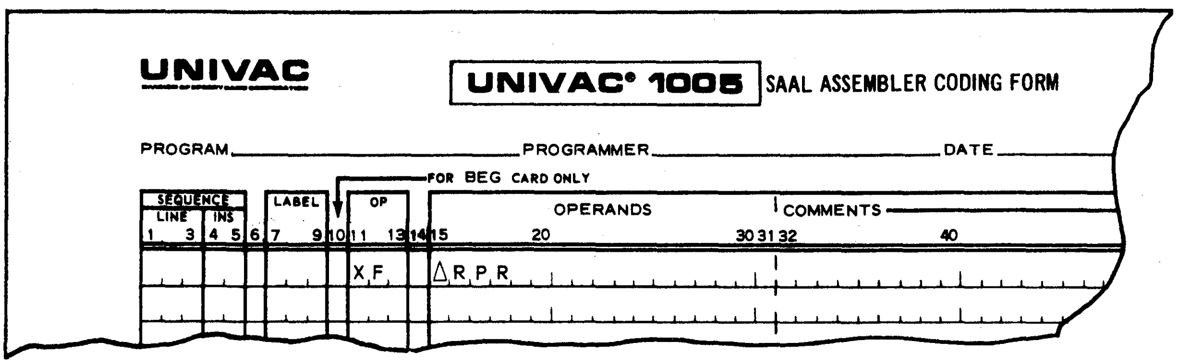
Notes: a.) The Read-Print instruction is a combination of the Read Card (XF REA) and the Print (XF PR1) instructions. All notes pertaining to these instructions are applicable to the Read-Print instructions.

b.) During the Read-Print execution cycle both I/O devices will function concurrently, with the execution time of the faster peripherial being overlapped by the slower one.

For example, in the case of a 400 CPM reader and a 600 LPM printer, the execution time required to read a card is sufficient so that the print cycle can be completed concurrently.

c.) The mnemonic operand field must be preceded by a space.

Example: Read the next card into the Card-buffer, print the contents of the Print-buffer and advance the printer form one line.



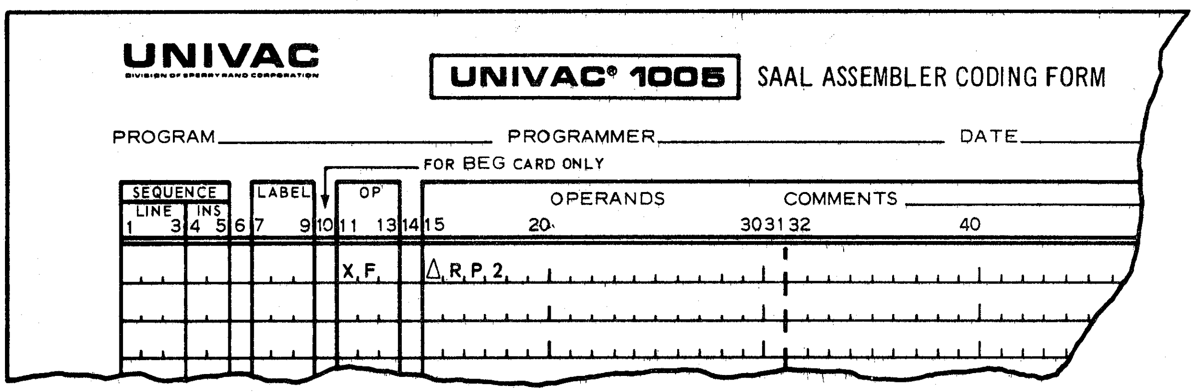
READ - PRINT - SPACE TWO: XF ΔRP2

Function: This instruction reads a full 80 column card into the U1005 input Card-buffer, prints the contents of the Print-buffer and advances the printer form two lines.

Notes: a.) All notes pertaining to the READ-PRINT-SPACE ONE (XF RPR) instruction are applicable to the READ-PRINT-SPACE TWO instruction.

b.) The mnemonic operand field must be preceded by a space.

Example: Read the next card into the Card-buffer, print the contents of the Print-buffer and advance the printer form two lines.


UNIVAC
DIVISION OF SPERRY RAND CORPORATION
UNIVAC® 1005 SAAL ASSEMBLER CODING FORM
PROGRAM PROGRAMMER DATE
FOR BEG CARD ONLY

SEQUENCE LINE 1 3	INS 4 5	6	LABEL 7 9	10	OP 11 13	14	OPERANDS 15 20 30	COMMENTS 31 32 40
					XF		ΔRP2	

READ - PUNCH: XF ΔRPH

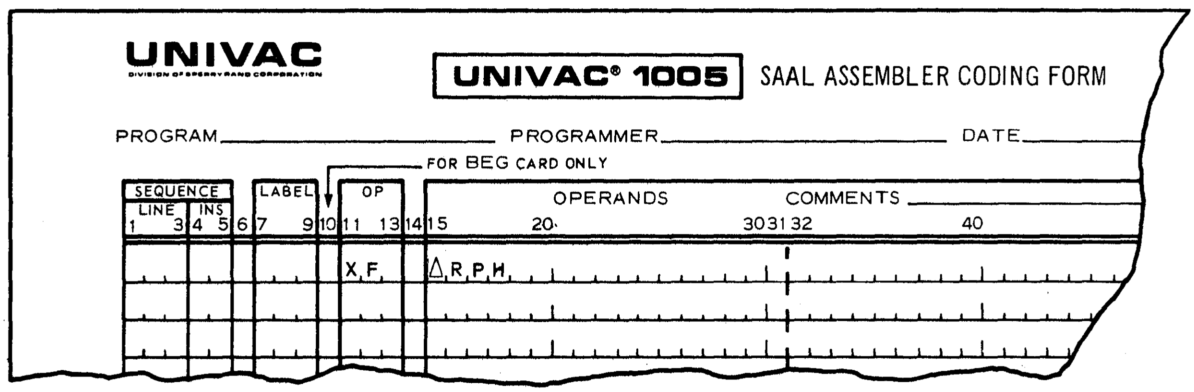
Function: This instruction reads a full 80 column card into the U1005 input Card-buffer and punches the 80 column card image in the Punch-buffer.

Notes: a.) The READ-PUNCH is a combination of the Read Card (XF REA) and the Punch (XF PUN) instructions. All notes pertaining to these instructions are applicable to the READ-PUNCH instruction.

b.) During the READ-PUNCH execution cycle, I/O devices will function concurrently with the execution time of the faster peripheral being overlapped by the slower one.

c.) The mnemonic operand field must be preceded by a space.

Example: Read the next card into the Card-buffer and punch the contents of the Punch-buffer.



READ-PRINT-PUNCH: XF ΔRPP

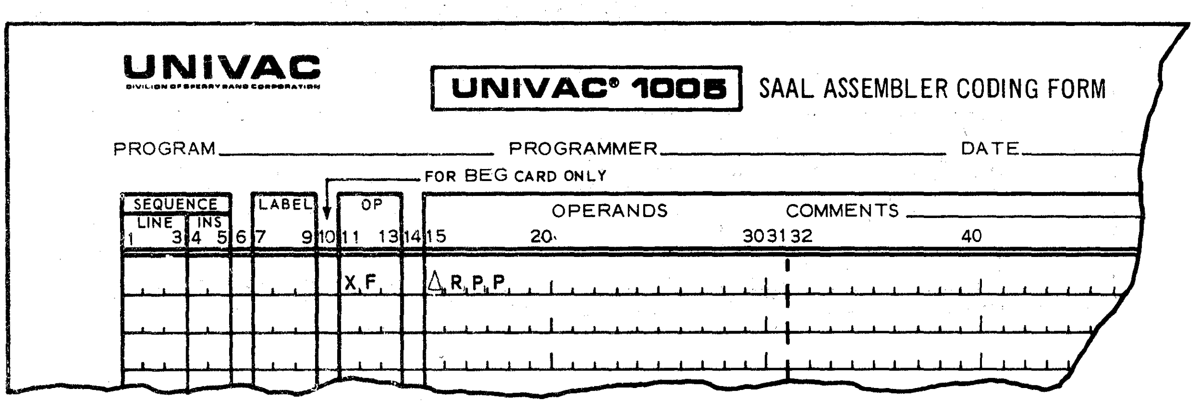
Function: This instruction reads a full 80 column card into the U1005 input Card-buffer, prints the contents of the Print-buffer, advances the printer form one line, and punches the contents of the Punch-buffer.

Notes: a.) The Read-Print-Punch instruction is a combination of the Read Card (XF REA), the Print (XF PR1), and the Punch Card (XF PUN) instructions. All notes pertaining to these instructions are applicable to the Read-Print-Punch instruction.

b.) During the Read-Print-Punch execution cycle, all three I/O devices will function concurrently, with the execution time of the faster peripherial being overlapped by the slower one. Ref. Read-Print Inst.

c.) The mnemonic operand field must be preceded by a space.

Example: Read the next card into the input Card-buffer, Print the contents of the Print-buffer, space the printer form one line, and punch the contents of the Punch-buffer.


UNIVAC
UNIVAC® 1005 SAAL ASSEMBLER CODING FORM
PROGRAM PROGRAMMER DATE
FOR BEG CARD ONLY

SEQUENCE LINE 1 3	INS 4 5	6	LABEL 7 9	10	OP 11 13	14	OPERANDS 15 20	COMMENTS 30 31 32 40
					XF		ΔRPP	

FORMS ADVANCE: XF ΔSK2

XF ΔSK4

XF ΔSK7

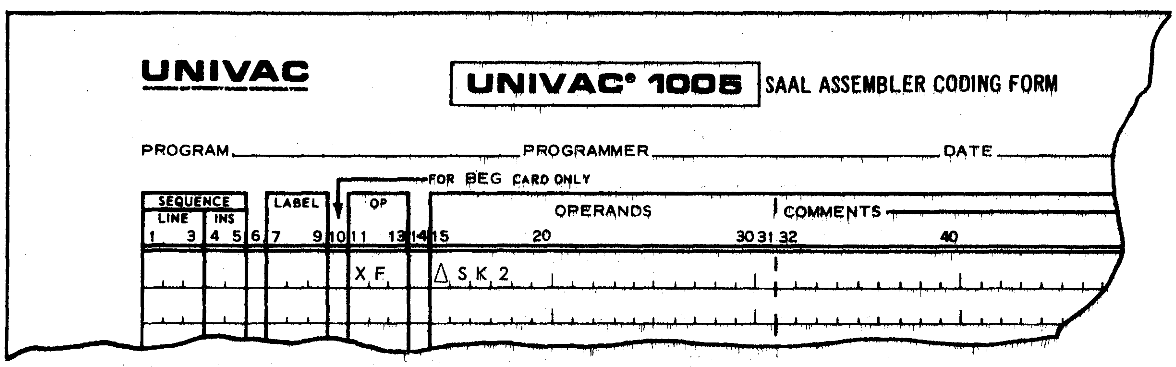
Functions: These instructions will advance the printed form as indicated by the forms control-loop.

Notes: a.) The Print-buffer area is not cleared following the execution of a skip command.

b.) Once the forms advance command has been initiated control is returned to the next instruction in sequence and further processing is overlapped during the actual form advancing.

c.) The mnemonic operand field must be preceded by a space.

Example: Advance the printer form until a channel two punch is detected in the control loop.



READ CODE IMAGE: XF ΔRCI

Function: This instruction reads a full 80 column card into the U1005 Card-buffer. The capacity of an 80 column card is expanded by allowing two columns of data to be obtained from what would ordinarily be <u>one</u> card column. At the same time, automatic code translation is suspended. Subsequently, the U1005 Card-buffer is incremented by 80 positions.

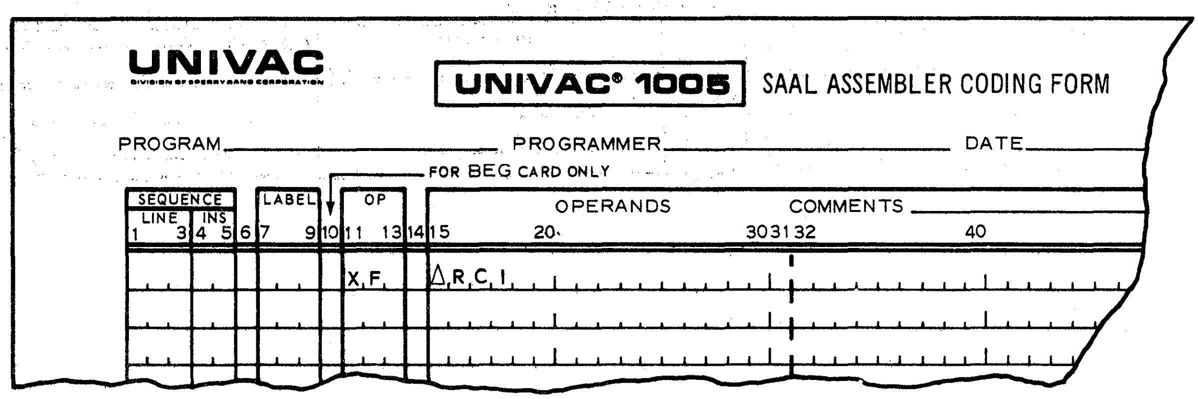
Notes: a.) The input Card-buffer area is 160 locations in length, beginning with memory location 1 through memory location 160.

b.) Input Card-buffer locations correspond sequentially to card columns. Thus, a configuration punched in card column 1 will be stored in memory locations 1 and 2, a configuration punched in card column 2 will be stored in memory locations 3 and 4 and so on.

c.) This instruction increases the data handling capacity of the U1005 in that the primary design is to combine in one card form the compact 6-position UNIVAC XS-3 code with the 12-position 80 column punched card code.

d.) The mnemonic operand field must be preceded by a space.

Example: Read a card from the Main Reader in Code Image mode.


UNIVAC

UNIVAC® 1005 SAAL ASSEMBLER CODING FORM

PROGRAM PROGRAMMER DATE

FOR BEG CARD ONLY

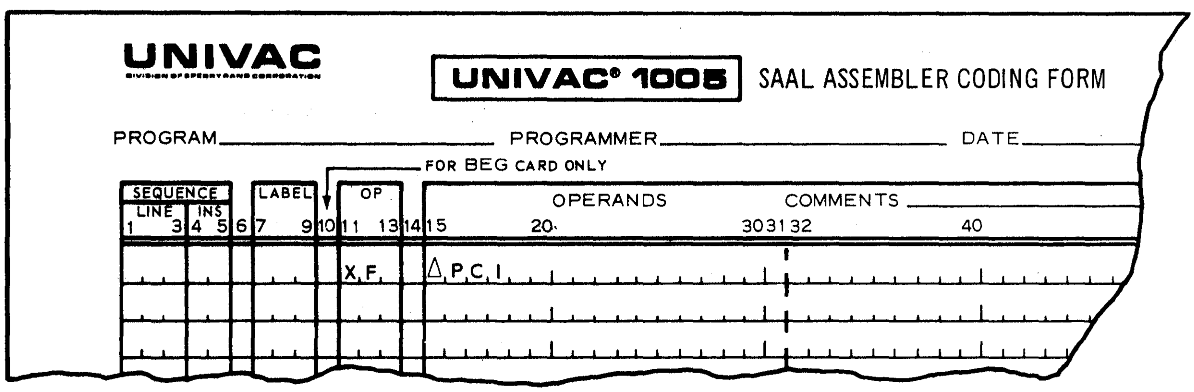
SEQUENCE LINE 1 3	INS 4 5	6	LABEL 7 9	10	OP 11 13	14	OPERANDS 15 20 30	COMMENTS 31 32 40
					XF		ΔRCI	

PUNCH CODE IMAGE: XF ΔPCI

Function: This instruction punches the card image located in the Code Image Punch-buffer into an 80 column card.

Notes:
a.) The Code Image Punch-buffer is 160 locations in length beginning with memory location 293 through memory location 452.
b.) Code Image Punch-buffer locations chronologically correspond to card columns. Thus, the data stored in locations 293 and 294 will be punched in card column 1, data stored in locations 295 and 296 will be punched in column 2 and so on.
c.) The Code Image Punch-buffer is not cleared following the execution of the PUNCH CODE IMAGE instruction.
d.) Once the punch cycle has been initiated, control is returned to the next instruction in sequence and further processing is overlapped during the punch cycle.
e.) The automatic XS-3 to 80 column code is suspended.
f.) The mnemonic operand field must be preceded by a space.

Example: Punch the card image stored in the Code Image Punch-buffer.


UNIVAC

UNIVAC® 1005 SAAL ASSEMBLER CODING FORM

PROGRAM ______ PROGRAMMER ______ DATE ______

FOR BEG CARD ONLY

SEQUENCE LINE 1 3	INS 4 5	6	LABEL 7 9	10	OP 11 13	14	OPERANDS 15 20 30	COMMENTS 31 32 40
					XF		ΔPCI	

READ AUXILIARY CODE IMAGE: XF ΔRXC

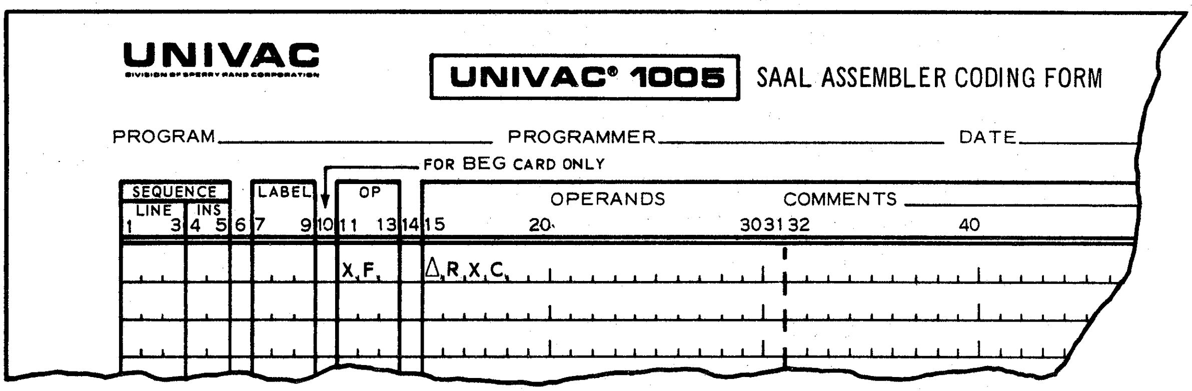
Function: Read a card from the Auxiliary Reader in Code Image mode.

Notes: a.) The READ AUXILIARY code image instruction places the prior card read in output stacker No. 1.

b.) All notes pertaining to the Read Code Image instruction (XF RCI) are applicable to the Read Auxiliary Code Image function.

c.) The mnemonic operand field must be preceded by a space.

Example: Read a card from the Auxiliary Reader in Code Image Code.



READ AUXILIARY WITH STACKER SELECT:

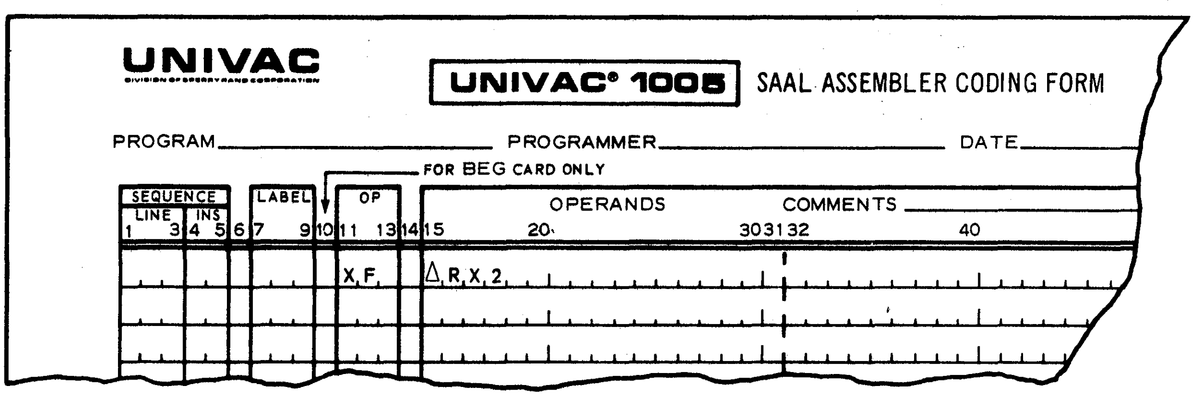
ONE	XF	ΔRX1
TWO	XF	ΔRX2
THREE	XF	ΔRX3

Function: This instruction reads a full 80 column card from the Auxiliary Reader into the U1005 input Card-buffer and places the prior card read in output stacker 1, 2 or 3 as designated by the numeric digit in the third position of the mnemonic operand field.

Notes: a.) All notes pertaining to the Read Card instruction (XF REA) are applicable to the READ AUXILIARY instruction.

b.) The mnemonic operand field must be preceded by a space.

Example: Read a card from the Auxiliary Reader and place the prior card read in Stacker 2.



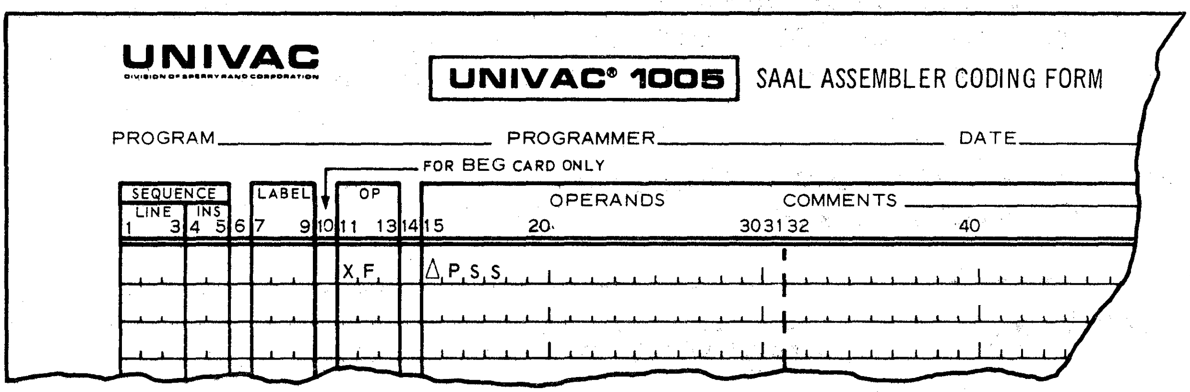
PUNCH WITH STACKER SELECT: XF ΔPSS

Function: This instruction punches the 80 column card image in the Punch-buffer and places the card being punched in the select stacker.

Notes: a.) All notes pertaining to the PUNCH instruction (XF PUN) are applicable to the PUNCH SELECT STACKER command.

b.) The mnemonic operand field must be preceded by a space.

Example: Punch the card image stored in the Punch-buffer and place that card in the select stacker.


UNIVAC
DIVISION OF SPERRY RAND CORPORATION
UNIVAC® 1005 SAAL ASSEMBLER CODING FORM
PROGRAM ______ PROGRAMMER ______ DATE ______
FOR BEG CARD ONLY

SEQUENCE LINE 1 3	INS 4 5	6	LABEL 7 9	10	OP 11 13	14	OPERANDS 15 20 30 31 32 COMMENTS 40
					XF		ΔPSS

READ/READ PUNCH: XF ΔRRP

Function: This instruction reads a full 80 column card from the punch unit into the 1005 input Read/Punch Card-buffer and punches a full 80 columns from the output Read/Punch Card-buffer into the second prior card read.

Notes:

a.) The input Read/Punch Card-buffer area is 80 locations in length, beginning with memory location 293 through memory location 372.

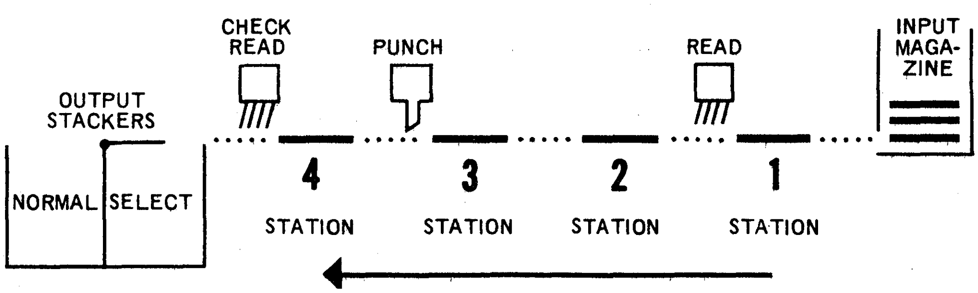
b.) Read/Punch Input Card-buffer locations correspond to card columns, thus a character punched in column 1 will be stored in location 293, a character punched in column 2 will be stored in location 294 and so on.

c.) Since the Read/Punch Input Card-buffer locations constitute the area normally reserved for the Punch-buffer, memory locations 373 through 452 are used for punching. Subsequently, any data in these locations during execution of the RRP instruction will be punched into the second previous card read.

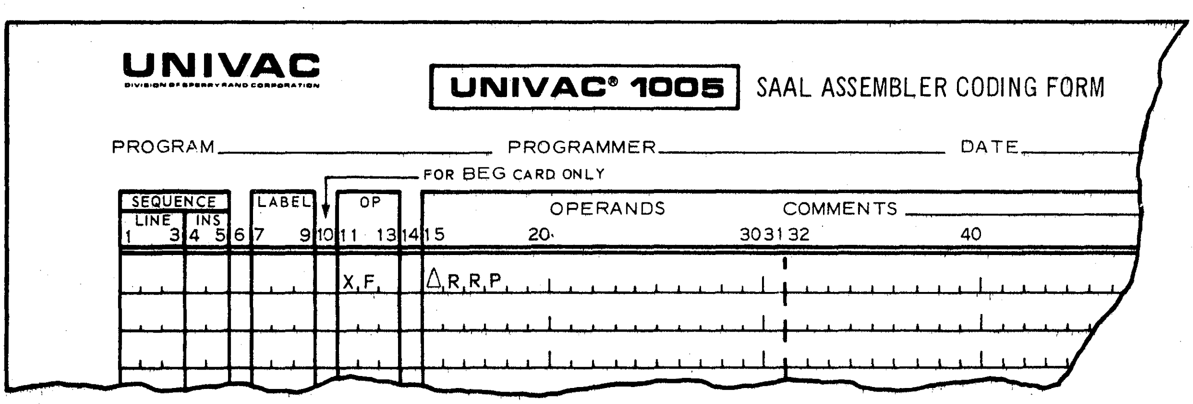
d.) As each column is read, it is automatically translated from Hollerith card code to XS-3.

e.) The mnemonic operand field must be preceded by a space.

Example: Read A card from the Read/Punch Unit Station 1 and punch the card in Station 3.



CARD PATH THROUGH READ/PUNCH



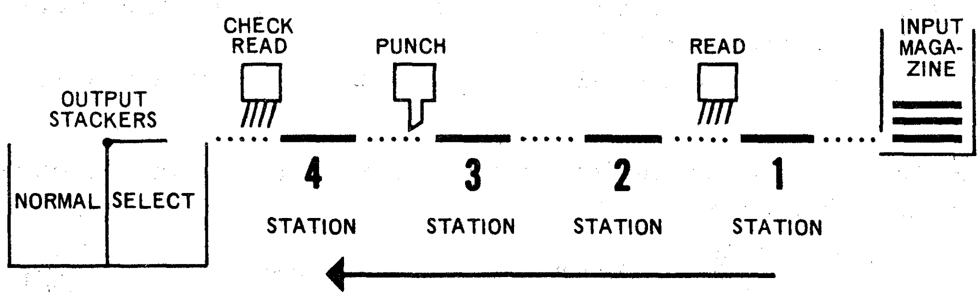
READ/READ PUNCH WITH STACKER SELECT: XF ΔRRS

Function: This instruction reads a full 80 column card from the Read/Punch into the U1005 Read/Punch Card-buffer and punches a full 80 columns from the output Read/Punch Card-buffer into the second prior card read, placing that card in the selected output stacker.

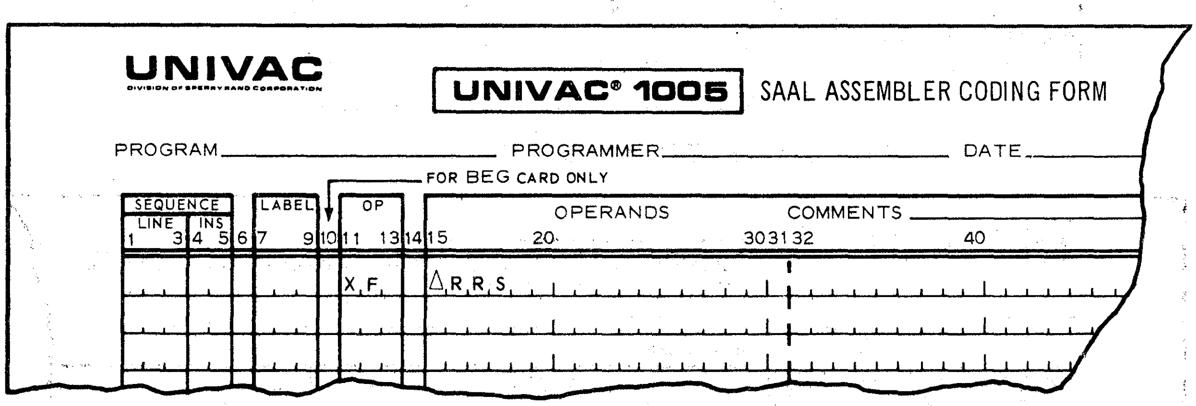
Notes: a.) The READ/PUNCH READ STACKER SELECT instruction is an offset of the READ/PUNCH READ instruction (XF RRP). All notes pertaining to the Read/Read Punch instruction (RRP) are applicable to the READ/PUNCH READ STACKER SELECT instruction.

b.) The mnemonic operand field must be preceded by a space.

Example: Read a card from the Read/Punch Unit Station 1 and punch and stacker select the card in Station 3.



CARD PATH THROUGH READ/PUNCH

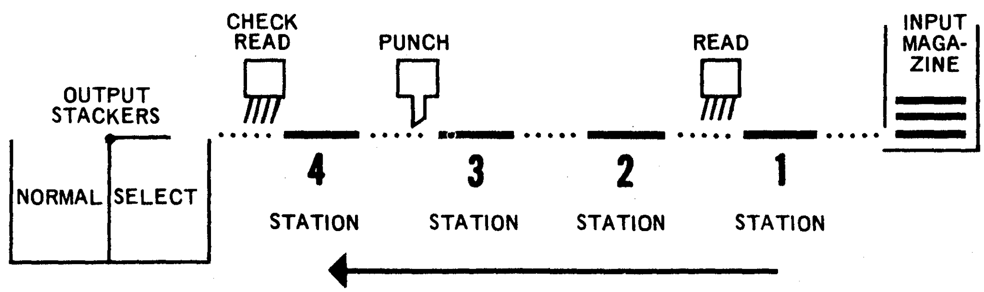


READ/READ PUNCH CODE IMAGE: XF ΔRRC

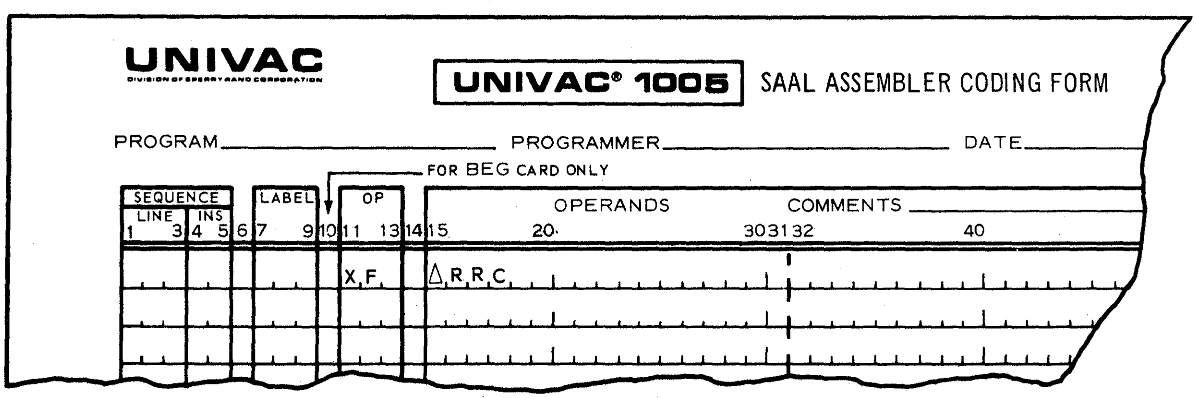
Function: This instruction reads a full 80 column card from the Read/Punch unit into the U1005 Read/Punch Card-buffer in Code Image mode and punches a full 80 columns from the output Read/Punch Card-buffer into the second prior card read in Code Image mode.

Notes:
a.) All notes pertaining to the READ CODE IMAGE instruction (XF RCI) are applicable to the READ/READ PUNCH CODE IMAGE instruction.
b.) The input buffer is 160 locations in length beginning with memory location 293 through memory location 452.
c.) Since the input buffer locations constitute the area normally reserved for the Punch-buffer, memory locations 453 through 612 are used for punching. Subsequently, any data in these locations during execution of the RRC instruction will be punched into the previous card read.
d.) The mnemonic operand field must be preceded by a space.

Example: Read a card from the Read/Punch Unit Station 1 in code image mode and punch the card in Station 3 in code image mode.



CARD PATH THROUGH READ/PUNCH

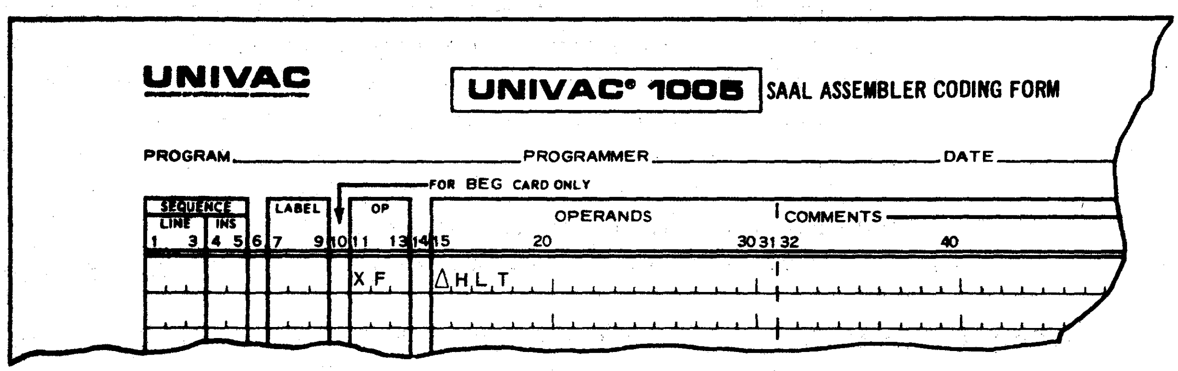


HALT: **XF ΔHLT**

Function: This instruction brings the computer to an orderly halt.

Notes:
a.) All I/O functions in processes will be completed before the halt will be effective.
b.) If the U1005 is restarted following a HALT the next instruction in sequence will be executed.
c.) The mnemonic operand field must be preceded by a space.

Example: Halt the computer


UNIVAC
UNIVAC® 1005 SAAL ASSEMBLER CODING FORM
PROGRAM ______ PROGRAMMER ______ DATE ______
FOR BEG CARD ONLY

SEQUENCE LINE 1 3	INS 4 5	6	LABEL 7 9	10	OP 11 13	14	OPERANDS 15 20 30 31	COMMENTS 32 40
					XF		ΔHLT	

C. INSTRUCTION REPERTOIRE - PAPER TAPE EXTERNAL FUNCTIONS AND CONDITIONAL TESTS

1. PAPER TAPE EXTERNAL FUNCTIONS

The Paper Tape Reader and Paper Tape Punch provide the UNIVAC 1005 with the ability to use paper tape as a direct input media and paper tape punch as a direct output media. The reader will accept any form of 5-, 6-, 7- or 8- track tape providing odd-parity checking when desired. The punch will perforate the aforementioned track tape codes providing odd-parity perforating if desired.

Paper tape reading and punching operations are controlled by the program. The input area starts with the first position of memory module one and will extend for the Tape Block length. Output area is designated to start at 0293 and extend for the Tape Block length. So that a wide variety of tape codes can be handled, the Paper Tape Reader and Punch functions to transmit or perforate an exact image of all or part of each tape frame. This selection is through program control which specifies 80 column read mode for 6 data track reading and punching, Code Image mode for 8 tape track reading and punching. In the above two modes, the 7th track is available for parity checking and the 8th track for special control. For data processing, the information recorded in paper tape can be entered one character at a time, 80 characters at a time, or a variable length block ended by a configuration of all bits present. For further assistance in data processing, the Paper Tape Reader permits printing and punching of end results directly from paper tape without intermediate tape-to-card conversion.

The format of the Paper Tape External Functions requires only the mode of punching or reading (80 Column or Code Image).

The Paper Tape External Function instructions pertain to Class III and are explained in detail on the following pages.

Class III: Class III instructions are Input/Output or External Function Commands and contain a mnemonic code in the "M" portion of an instruction indicating the I/O device or devices to be initiated.

READ PAPER TAPE:	XF	ΔRP1	Read 1 Frame
	XF	ΔRP8	Read 80 Frames
	XF	ΔRPS	Read through Sentinel

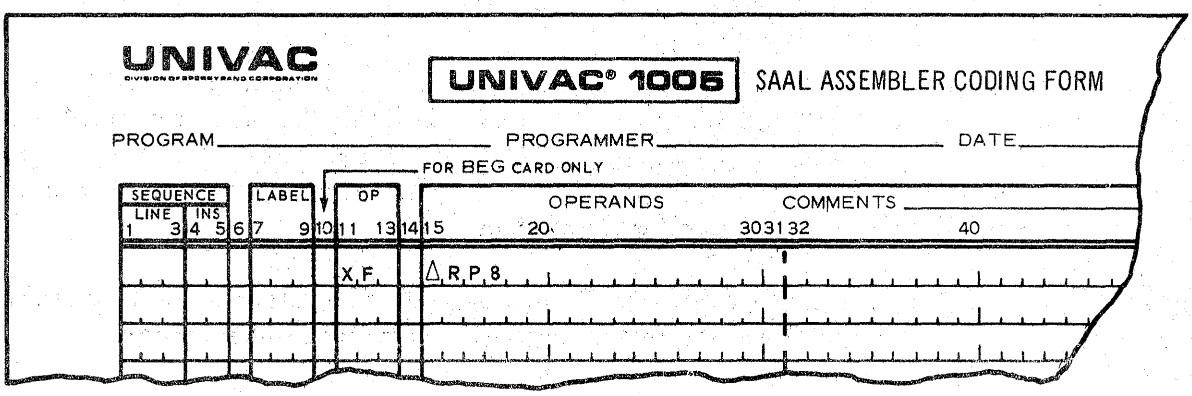
Function: This instruction reads a block of tape into the U1005 Card Read-buffer. The variable length of the block is determined by the 3rd character of the mnemonic field. Specifically, RP1 designates a 1 character block, RP8 designates an 80 character block, RPS designates a variable length block ended by a configuration of all bits present.

Notes: a.) Substituting a frame in paper tape for a column in the card, all notes pertaining to the Read instruction (XF ΔREA) are applicable to the Read Paper Tape instruction.

b.) On a RPS instruction, the all bit present character is read.

c.) The mnemonic operand field must be preceded by a space.

Example: Read a block of paper tape 80 characters in length.


UNIVAC
DIVISION OF SPERRY RAND CORPORATION
UNIVAC® 1005 SAAL ASSEMBLER CODING FORM

PROGRAM ______ PROGRAMMER ______ DATE ______

FOR BEG CARD ONLY

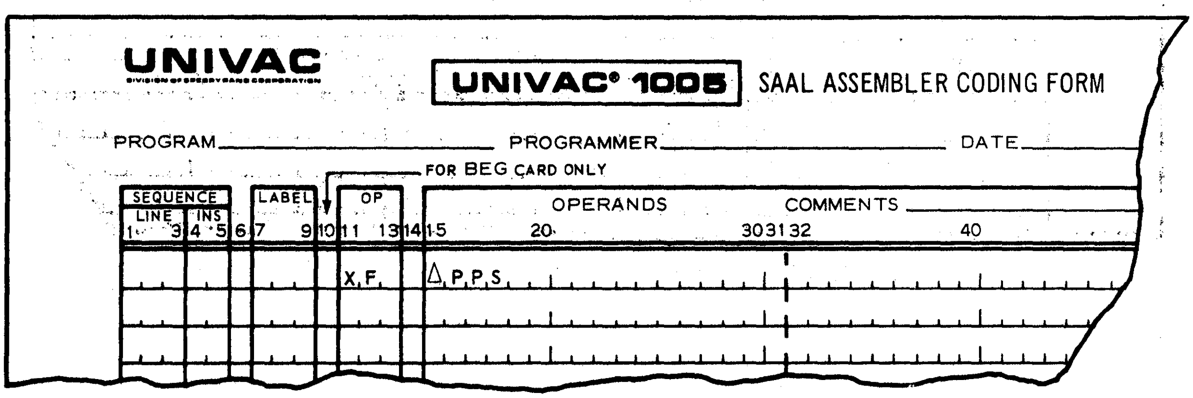
SEQUENCE LINE 1 3	INS 4 5	6	LABEL 7 9	10	OP 11 13	14	OPERANDS 15 20 30	COMMENTS 31 32 40
					XF		ΔRP8	

PUNCH PAPER TAPE WITHOUT PARITY: XF ΔPP1 Punch 1 Frame
XF ΔPPS Punch to Sentinel

Function: This instruction punches a block of tape from the U1005 Card Punch-buffer. The variable length of the block is determined by the 3rd character of the mnemonic operand field. Specifically, PP1 designates a 1 character block, PPS designates a variable - length block ended by a configuration of all bits present.

Notes:
a.) Substituting a frame in paper tape for a column in the card, all notes pertaining to the PUNCH instruction (XF PUN) instruction are applicable to the PUNCH PAPER TAPE instruction.
b.) On a PPS instruction, the all bit present character is not punched.
c.) The mnemonic operand field must be preceded by a space.

Example: Punch a block of paper tape up to but not including the sentinel (all bits).


UNIVAC
UNIVAC® 1005 SAAL ASSEMBLER CODING FORM
PROGRAM PROGRAMMER DATE
FOR BEG CARD ONLY

SEQUENCE LINE 1 3	INS 4 5	6	LABEL 7 9	10	OP 11 13	14	OPERANDS 15 20 30	COMMENTS 31 32 40
					XF		ΔPPS	

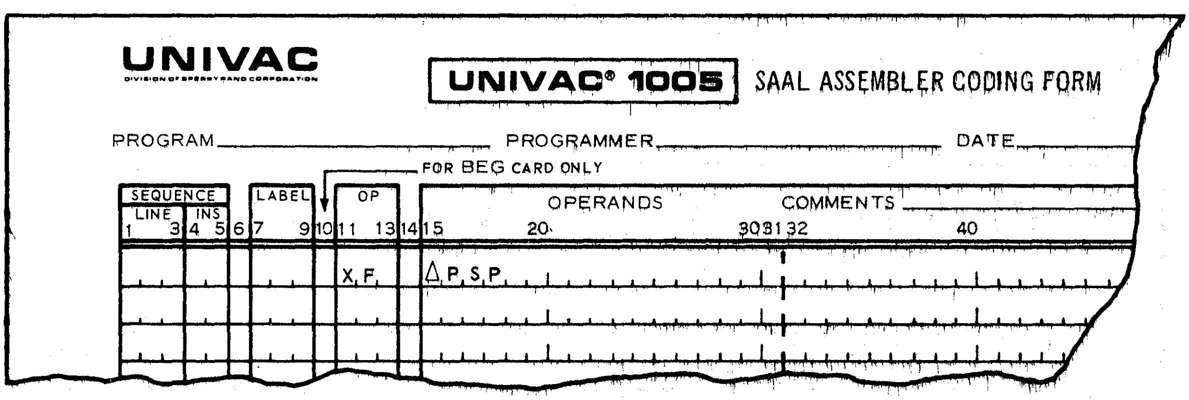
PUNCH PAPER TAPE WITH PARITY: XF ΔP1P Punch 1 Frame
XF ΔPSP Punch to Sentinel

Function: This instruction punches a block of tape with odd-parity from the U1005 Card Punch-buffer. The variable length of the block is defined by the second character of the mnemonic operand field. When punching to (but not including) sentinel, all bits constitute the sentinel configuration.

Note: a.) All notes pertaining to the PUNCH PAPER TAPE instruction are applicable to the above instructions.

b.) The mnemonic operand field must be preceded by a space.

Example: Punch a block of paper tape with odd-parity up to but not including the sentinel (all bits).


UNIVAC
DIVISION OF SPERRY RAND CORPORATION
UNIVAC® 1005 SAAL ASSEMBLER CODING FORM
PROGRAM PROGRAMMER DATE
FOR BEG CARD ONLY

SEQUENCE LINE 1 3	INS 4 5	6	LABEL 7 9	10	OP 11 13	14	OPERANDS 15 20 30 31 32 COMMENTS 40
					XF		ΔPSP

2. PAPER TAPE CONDITIONAL TESTS

Associated with the UNIVAC 1005 Paper Tape System are two (2) Conditional instructions which allow the programmer to test for parity error and channel 8 conditions.

The Paper Tape Conditional Test instructions pertain to Class II and are explained in detail on the following pages.

Class II: Class II instructions contain only an "M" address indicating the most significant character of an instruction. This format is employed exclusively by Jump or Branching instructions.

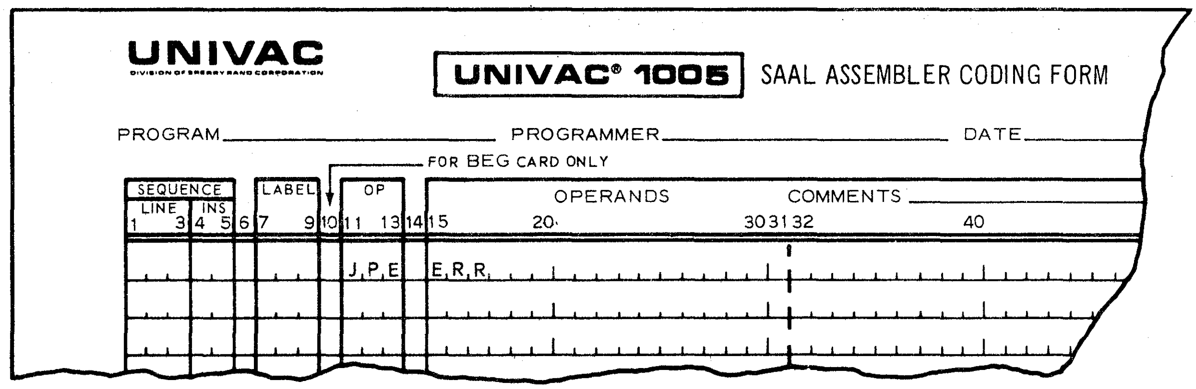
PAPER TAPE CONDITIONAL TESTS: Jump Parity Error: JPE M
Jump Channel 8: JC8 M

Function: Transfer program control to the instruction stored at M if the condition specified by the operation code is present.

Notes: a.) These instructions are used to test the status of paper tape instructions after execution.

b.) If the condition tested is not present, control will not be transferred and the next instruction in the testing sequence will be executed.

Example: Test results of a previous paper tape read instruction. If the condition is true, transfer control to the routine labeled ERR.


UNIVAC
DIVISION OF SPERRY RAND CORPORATION
UNIVAC® 1005 SAAL ASSEMBLER CODING FORM
PROGRAM ______ PROGRAMMER ______ DATE ______
FOR BEG CARD ONLY

SEQUENCE LINE 1 3	INS 4 5	6	LABEL 7 9	10	OP 11 13	14	OPERANDS 15 20 30 31 32	COMMENTS 40
					JPE		ERR	

D. INSTRUCTION REPERTOIRE - MAGNETIC TAPE EXTERNAL FUNCTIONS AND CONDITIONAL TESTS

1. MAGNETIC TAPE EXTERNAL FUNCTIONS

The UNISERVO VI C Magnetic Tape Units provide the UNIVAC 1005 with the capability of reading and writing IBM compatible tapes at densities of 200, 556 and 800 Characters Per Inch (CPI). When using more than one unit, it is possible to read or write any six level code at a given density on one or more units, and another code at a different density on one or more other units. Seven tape tracks are read and written; one parity and six data tracks.

Magnetic tape reading and writing operations are controlled by the program. Input/Output areas may be the 1st core position of any memory bank designated by the programmer. Data checking includes character parity, automatically performed by all tape units. In addition to Read and Write instructions, the 1005 features the Backspace one block, Erase before write, Read at high gain and Rewind functions. The programmer has an option of using odd or even parity. The UNIVAC 1005 is capable of handling up to 2 Magnetic Tape Units.

The format of the Magnetic Tape External Functions is slightly different in that a Buffer Directive (See Assembler Directives) and a length (of block) must be employed. The length, which designates the number of characters to be read or written, can be any number from 1 to 961. However, on a write instruction the length must be 5 characters greater than the number of characters to be written. When reading variable length records, the length must be the largest number of characters to be read. Reading terminates when an inter-block gap is encountered or when the designated length is read, whichever occurs last. When the block length is shorter than the maximum length, the remainder will be space filled.

The Magnetic Tape External Function instructions pertain to Class IV and are explained in detail on the following pages.

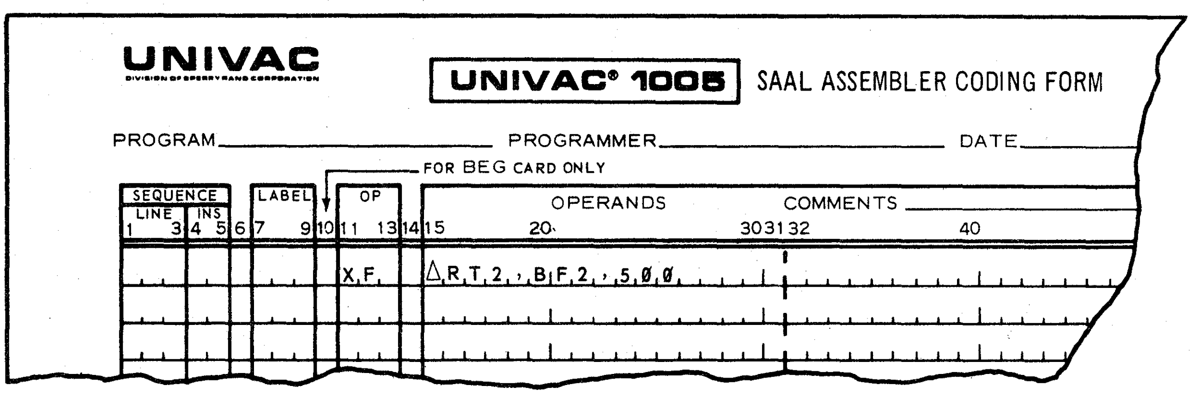
Class IV: Class IV instructions are Input/Output or External Function Commands, and contain a mnemonic code, Buffer (BF_n), and length in the "M" portion of an instruction indicating the I/O device, memory bank, and length of operand to be initiated.

READ TAPE:	Servo One Normal Gain	XF	ΔRT1, BF_n, L
	Servo Two Normal Gain	XF	ΔRT2, BF_n, L
	Servo One High Gain	XF	ΔRT5, BF_n, L
	Servo Two High Gain	XF	ΔRT6, BF_n, L

Function: This instruction reads a block of magnetic tape into the U1005 memory.

Notes:
a.) The number of the Servo from which the data is to be read is designated by the 3rd character of the mnemonic operand field.
b.) The BF_n mnemonic designates the bank of memory in which the data is to be read. (See Assembler Directives.) Reading starts in the first memory location of the designated bank.
c.) The L mnemonic is a number from 1 to 961 and is used to determine the length of the block being read.
d.) Normal tape operations are in odd parity. An asterisk (*) is placed in card column 15 to designate an even parity operation.
e.) To indicate a High Gain Read function, the third character of the mnemonic operand field (Servo number) is incremented by 4.
f.) The mnemonic operand field must be preceded by a space (except for even parity).

Example: Read a block of tape from Servo 2, odd parity, normal gain and store data into core positions 0962 - 1461.



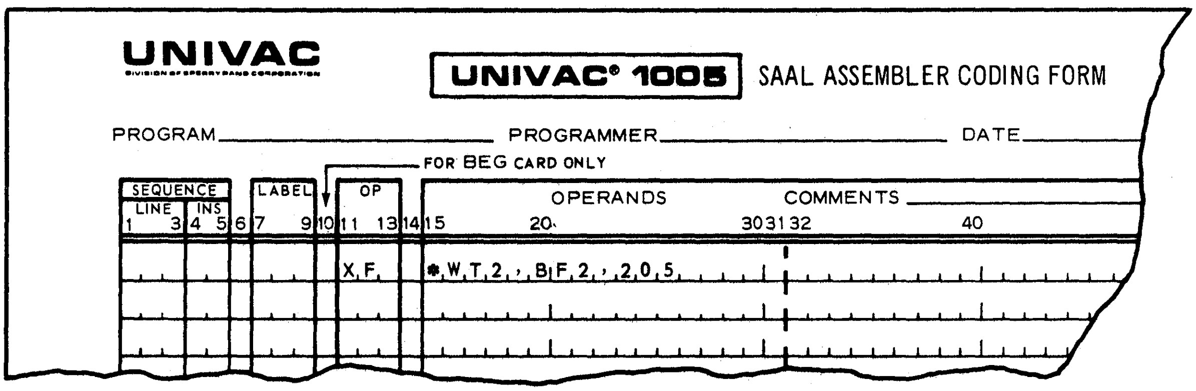
WRITE TAPE: Servo One XF ΔWT1, BF_n, L
Servo Two XF ΔWT2, BF_n, L

Function: This instruction writes a block of data from the U1005 memory onto magnetic tape.

Notes: a.) The L mnemonic is the number used to determine the length of the block to be written. This number must be 5 greater than the actual number of characters to be written.

b.) All other notes pertaining to the READ TAPE instruction are applicable to the WRITE TAPE function.

Example: Write a block of tape on Servo 2, even parity, from core positions 1923 - 2122.


UNIVAC
DIVISION OF SPERRY RAND CORPORATION
UNIVAC® 1005 SAAL ASSEMBLER CODING FORM

PROGRAM ______ PROGRAMMER ______ DATE ______

FOR BEG CARD ONLY

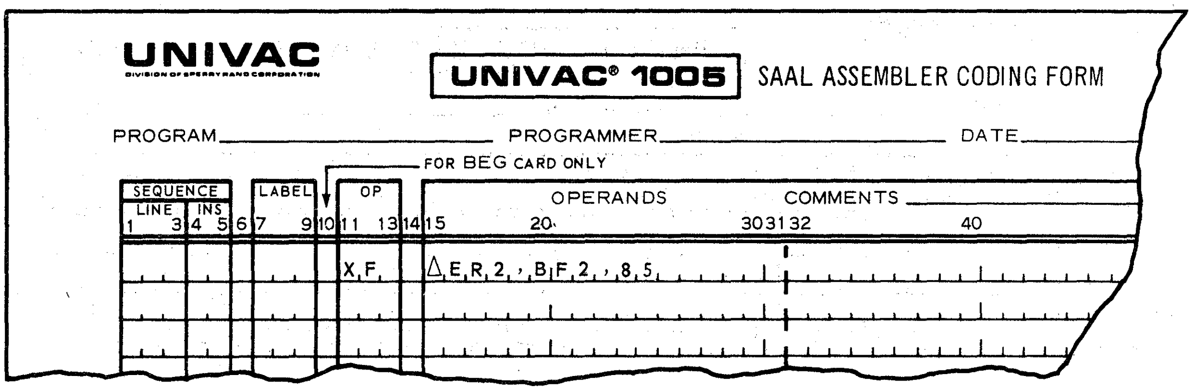
SEQUENCE LINE 1 3	INS 4 5	6	LABEL 7 9	10	OP 11 13	14	OPERANDS 15 20 30	COMMENTS 31 32 40
					XF		*WT2,BF2,205	

ERASE BEFORE WRITE: Servo One XF ΔER1, BF_n, L
Servo Two XF ΔER2, BF_n, L

Function: This instruction is used to delay the writing of a block on tape, to insure that a portion of tape is erased before writing on it. This instruction can be used to continue an old file or by-pass a bad spot by backspacing and then writing again with the ERASE BEFORE WRITE instruction (See conditional test - parity error recovery example).

Note: a.) All notes pertaining to the WRITE TAPE instruction are applicable to the ERASE BEFORE WRITE function.

Example: Erase before write a block of tape on Servo 2, odd parity, from core positions 1923-2002.

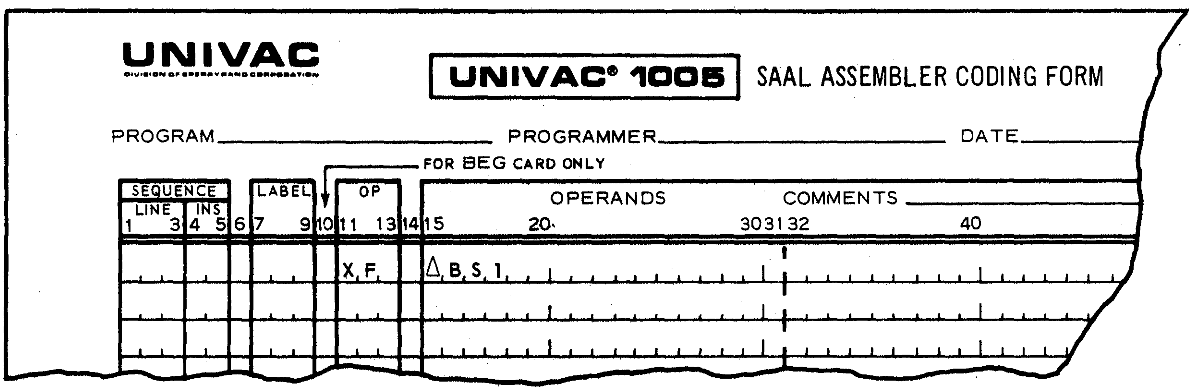


BACKSPACE: Servo One XF ΔBS1
Servo Two XF ΔBS2

Function: This instruction generates the backspace of one magnetic tape block (See conditional test-parity error recovery example).

Notes:
a.) The third character of the mnemonic operand field designates the Magnetic Tape Servo on which the backspace is to occur.
b.) BF_n, L is not to be used with this instruction.
c.) The mnemonic operand field must be preceded by a space.

Example: Backspace a block of tape on Servo 1.


UNIVAC
UNIVAC® 1005 SAAL ASSEMBLER CODING FORM
PROGRAM PROGRAMMER DATE
FOR BEG CARD ONLY
SEQUENCE LINE 1 3 INS 4 5 | 6 | LABEL 7 9 | 10 | OP 11 13 | 14 | OPERANDS 15 20 30 | COMMENTS 31 32 40
XF ΔBS1

REWIND: Servo One XF ΔRW1
Servo Two XF ΔRW2

Function: This instruction causes the tape to rewind to a point past the load point. Depression of the LOAD POINT switch, following the REWIND instruction, causes the tape to advance to the load point.

Notes: a.) The third character of the mnemonic operand field designates which Magnetic Tape Servo is to be rewound.
b.) BF_n, L is not to be used with this instruction.
c.) The mnemonic operand field must be preceded by a space.

Example: Rewind Servos 1 and 2.

UNIVAC

UNIVAC® 1005 SAAL ASSEMBLER CODING FORM

PROGRAM ________ PROGRAMMER ________ DATE ________

FOR BEG CARD ONLY

SEQUENCE LINE 1–3	INS 4–5	6	LABEL 7–9	10	OP 11–13	14	OPERANDS 15 … 20 … 30	COMMENTS 31 32 … 40
					XF		ΔRW1	
					XF		ΔRW2	

2. MAGNETIC TAPE CONDITIONAL TESTS

Associated with the UNIVAC 1005 Magnetic Tape System are two (2) Conditional Tape instructions which allow the programmer to test for parity error and end of tape conditions.

The Magnetic Tape Conditional Test instructions pertain to Class II and are explained in detail on the following pages.

Class II: Class II instructions contain only an "M" address indicating the most significant character of an instruction. This format is employed exclusively by Jump or Branching instructions.

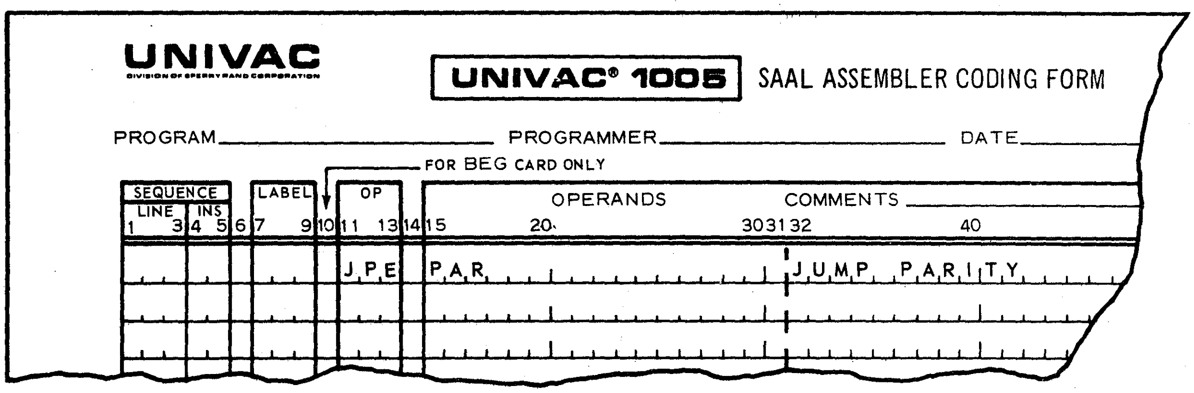
MAGNETIC TAPE CONDITIONAL TESTS: Jump Parity Error: JPE M
Jump End of Tape: JET M

Function: Transfer program control to the instruction stored at M if the condition specified by the operation code is present.

Notes: a.) These instructions are used to test the status of magnetic tape instructions after execution.

b.) If the condition tested is not present control will not be transferred and the next instruction in the testing sequence will be executed.

Example: Test results of a previous magnetic tape read or write instruction. If the condition is true, transfer control to the routine labeled PAR.


UNIVAC
DIVISION OF SPERRY RAND CORPORATION
UNIVAC® 1005 SAAL ASSEMBLER CODING FORM
PROGRAM PROGRAMMER DATE
FOR BEG CARD ONLY

SEQUENCE LINE 1 3	INS 4 5	6	LABEL 7 9	10	OP 11 13	14	OPERANDS 15 20	30	COMMENTS 31 32 40
					JPE		PAR		JUMP PARITY

MAGNETIC TAPE CONDITIONAL TESTS

One method of handling parity errors is as follows:

Example: Parity on Read Function

UNIVAC® 1005 SAAL ASSEMBLER CODING FORM

PROGRAM ____________ PROGRAMMER ____________ DATE ________

← FOR BEG CARD ONLY (column 10)

SEQUENCE LINE (1–3)	INS (4–5)	6	LABEL (7–9)	10	OP (11–13)	14	OPERANDS (15–31)	COMMENTS (32–)
							IN DATA DIVISION	-LITERAL
			CT2		+5		04001	COUNTER
							IN PROCEDURE DIVI	SION
001					CLR		CT2+2,2	
002			TRD		XF		RT1,BF2,350	READ SERVO 1 NORM
003			PTE		JPE		RPE	TEST FOR PARITY
004					JET		RTE	
005			RPE		IC		CT2	INCREMENT CTR
006					JL		$+15	REPEAT 3 TIMES
007					XF		HLT	HALT CLEAN HEAD
008					CLR		CT2+2,2	REPEAT AGAIN
009					XF		BS1	BACK SPACE SERVO1
010					XF		RT5,BF2,350	READ SERVO 1 HIGH
011					JPE		$+10	REPEAT NORMAL READ
012					J		PTE+5	CORRECT
013					XF		BS1	BACKSPACE SERVO 1
014					J		TRD	TO TAPE READ

SEQNO 001 - Clear Read Parity Error Counter
002 - Read One Block of Tape from Servo 1, Normal Gain, Odd Parity
003 - Test for Parity Error
004 - Test for End of Tape
005 - Increment the Read Parity Error Counter
006 - Jump Less to 009
007 - Counter Equals 4, Halt and Clean Servo Head
008 - Clear Counter and Repeat
009 - Backspace Servo 1
010 - Read One Block from Servo 1, High Gain, Odd Parity
011 - Test for Parity Error
012 - Correct, Jump to Seq. No. 004
013 - Error, Backspace Servo 1
014 - To Seq. No. 002

MAGNETIC TAPE CONDITIONAL TESTS

Example: Parity on Write Function

UNIVAC® 1005 SAAL ASSEMBLER CODING FORM

PROGRAM ________________ PROGRAMMER ________________ DATE ________

FOR BEG CARD ONLY (column 10)

SEQUENCE LINE 1–3	INS 4–5	6	LABEL 7–9	10	OP 11–13	14	OPERANDS 15–31	COMMENTS 32–40
							IN DATA DIVISION	LITERAL
			CT1		+5		07001	COUNTER
							IN PROCEDURE DIVISION	
001					CLR		CT1+2,2	CLEAR CTR
002			TWR		XF		WT2,BF2,100	WRITE SERVO 2
003			TPE		JPE		WPE	TEST FOR PARITY
004					JET		WOT	TO END OF TAPE
005			WPE		IC		CT1	INCR CTR
006					JL		$+15	REPEAT 6 TIMES
007					XF		HLT	HALT CLEAN HEAD
008					CLR		CT1+2,2	REPEAT AGAIN
009					XF		BS2	BACK SPACE SERVO 2
010					XF		ER2,BF2,100	ERASE BEFORE WRITE
011					J		TPE	REPEAT AGAIN

SEQ NO 001 - Clear Write Parity Error Counter
002 - Write One Block of Tape on Servo 2, Odd Parity
003 - Test for Parity Error
004 - Test for End of Tape
005 - Increment the Write Parity Error Counter (07001)
006 - Jump Less to SEQ NO 009
007 - Counter Equals 7, Halt and Clean Servo Head
008 - Clear Counter and Repeat
009 - Backspace Servo 2
010 - Erase Before Write, Odd Parity
011 - Jump to SEQ NO 003

E. INSTRUCTION REPERTOIRE - ADVANCED PROGRAMMING

The advanced programming instructions are applicable only to an Extended 1005 System and a program which utilizes these instructions can not be executed on a 2K 1005 System.

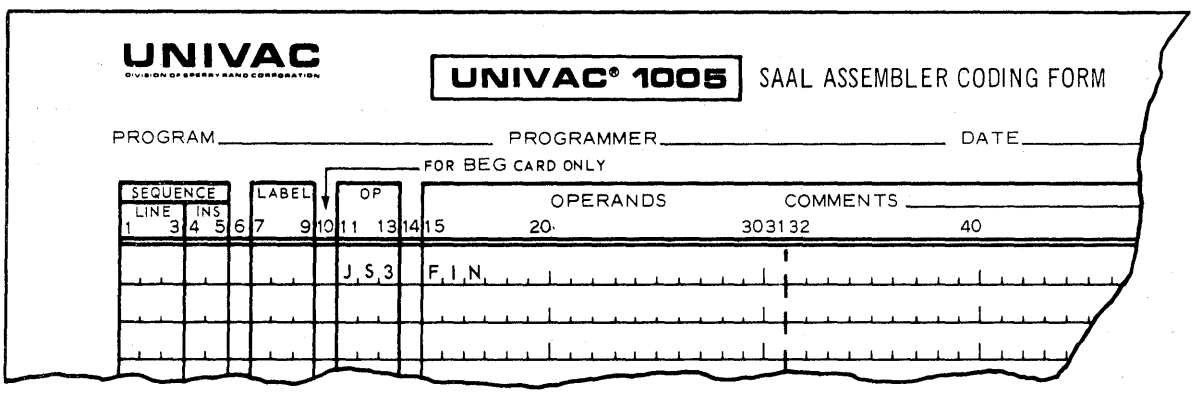
NOTE: CCA, SC, LAN, LOR instructions require a symbolic tag in the operand field.

JUMP ALTERNATE SWITCH 3: JS3 M

Function: Transfer program control to the instruction stored at M if Alternate switch 3 is on/illuminated.

Note: If the condition tested is not present, control will not be transferred and the next instruction in sequence will be executed.

Example: Transfer control to the routine labeled FIN if alternate switch 3 is illuminated.


UNIVAC
DIVISION OF SPERRY RAND CORPORATION

UNIVAC® 1005 SAAL ASSEMBLER CODING FORM

PROGRAM ______ PROGRAMMER ______ DATE ______

FOR BEG CARD ONLY

SEQUENCE LINE 1 3	INS 4 5	6	LABEL 7 9	10	OP 11 13	14	OPERANDS 15 20 30	COMMENTS 31 32 40
					JS3		FIN	

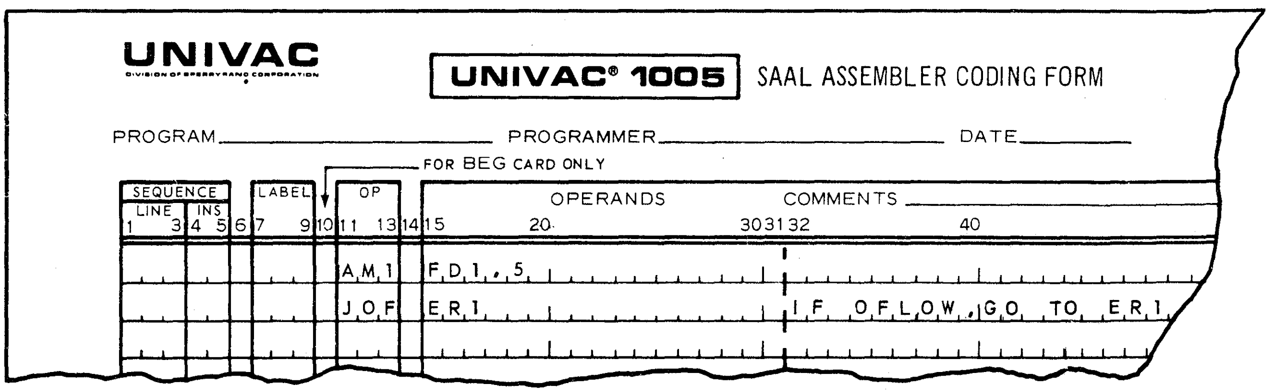
JUMP ARITHMETIC OVERFLOW: JOF M

Function: Transfer program control to the instruction stored at M if the Arithmetic overflow indicator is set.

a.) This instruction is used to test the results of an arithmetic operation.

b.) If the condition tested is not present, control will not be transferred and the next instruction in sequence will be executed.

Example: Add the 5 least significant characters of Arithmetic Register one (AR1) to the field FD1 and test the result for Arithmetic overflow.


UNIVAC
DIVISION OF SPERRY RAND CORPORATION
UNIVAC® 1005 SAAL ASSEMBLER CODING FORM
PROGRAM PROGRAMMER DATE
FOR BEG CARD ONLY

SEQUENCE LINE 1–3	INS 4–5	6	LABEL 7–9	10	OP 11–13	14	OPERANDS 15–30	COMMENTS 32–…
					AM1		FD1,5	
					JOF		ER1	IF OFLOW, GO TO ER1

AR1 (before and after) = 0000056982
FD1 (before) = 55692
FD1 (after) = 12674

In the above example, the Arithmetic overflow indicator is set and control is transferred to the routine labeled ER1.

COMPARE CHARACTER ALPHA/NUMERIC: CCA M, L Δ C

Function: Compare for equality the least significant location of the field specified by M and L, to the character specified by C.

Notes:

a.) L specifies the length and should equal 1. If L is unequal to 1, the least significant location of M will be compared to the character specified by C.

b.) C specifies the character M will be compared to and may be any one of the 63 valid UNIVAC 1005 characters. If no character is specified, M will be compared to a space.

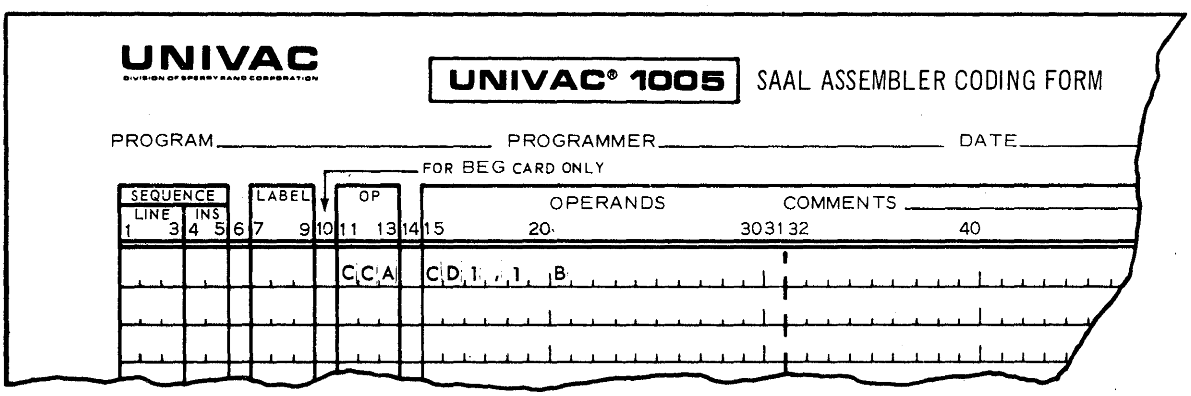
c.) The C character must be preceded by a space.

d.) This is a binary comparison and all data bits are considered.

e.) The results of the comparison is recorded in testable indicators as follows:

	JUA (Unequal)	JEA (Equal)
(MEM) = C		Set
(MEM) ≠ C	Set	

Example: Compare the one character field CD1 against the character B.


UNIVAC
DIVISION OF SPERRY RAND CORPORATION
UNIVAC® 1005 SAAL ASSEMBLER CODING FORM
PROGRAM ______ PROGRAMMER ______ DATE ______
FOR BEG CARD ONLY

SEQUENCE LINE 1 3	INS 4 5	6	LABEL 7 9	10	OP 11 13	14	OPERANDS 15 20 30 31 32 COMMENTS 40
					CCA		CD1,1 B

In the above example, if the contents of CD1 contained a B, the JEA (equal) indicator will be set. If it did not contain a B, the JUA (unequal) indicator will be set.

STORE CHARACTER: SC M, L Δ C

Function: Store the character specified by C into the least significant location of the field specified by M and L.

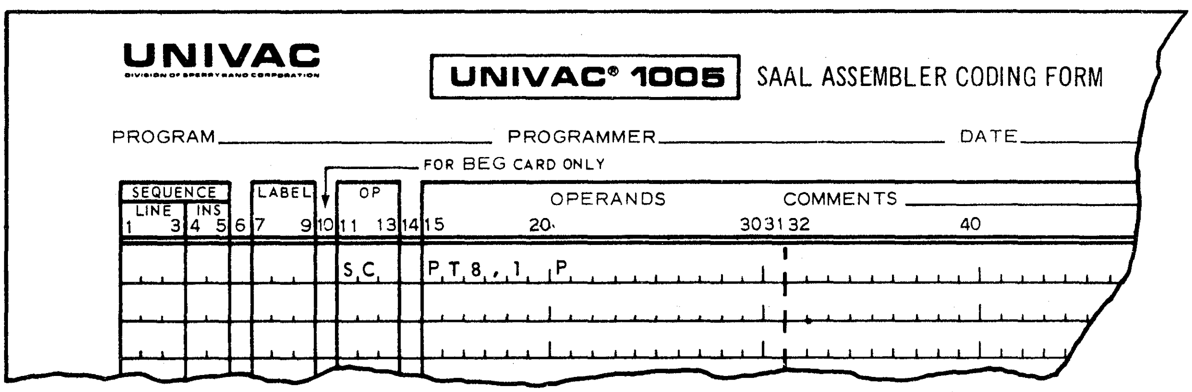
Notes:

a.) L specifies the length and should equal 1. If L is unequal to 1, the character will be stored in the least significant location of M.

b.) C specifies the character to be stored in M and may be any one of the 63 valid UNIVAC 1005 characters. If no character is specified, a space will be stored in M.

c.) The character must be preceded by a space.

Example: Store the character P into the one character field PT8.



LOGICAL AND: LAN M, L Δ C

Function: Compute the logical product of the character specified by C and the least significant location of the field specified by M and L. The result replaces the least significant location of the field specified by M and L.

Notes:

a.) L specifies the length and should equal 1. If L is unequal to 1, the least significant location of M will be used to compute the logical product.

b.) C specifies the character used to compute the logical product and may be any one of the 63 valid UNIVAC 1005 characters. If no character is specified, a space will be used to compute the logical product.

c.) The C character must be preceded by a space.

d.) For each zero bit in the C character the corresponding bit position in M is cleared to zero. For each one bit in the C character the corresponding bit in M is retained.

The logical product is formed based on the following truth table:

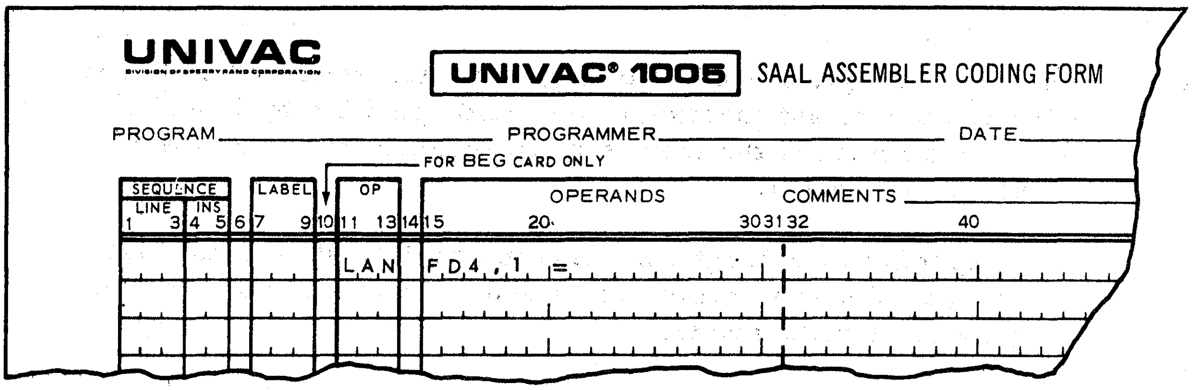
AND	0	1
0	0	0
1	0	1

i.e.,

C	θ*	(M)	⟶	(M)
0	θ	0	=	0
0	θ	1	=	0
1	θ	0	=	0
1	θ	1	=	1

* 0 represents the logical product

Example: Compute and store the logical product of the character = and the one character field labeled FD4.


UNIVAC

UNIVAC® 1005 SAAL ASSEMBLER CODING FORM

PROGRAM ____ PROGRAMMER ____ DATE ____

FOR BEG CARD ONLY

SEQUENCE LINE 1 3	INS 4 5	6	LABEL 7 9	10	OP 11 13	14	OPERANDS 15 20 30 31 32 COMMENTS 40
					LAN		FD4,1 =

C (before and after) 011111 equals =
FD4 (before) 100100 equals -1
FD4 (after) 000100 equals +1

In the above example, the C character is used to remove the "X" bit of FD4.

LOGICAL OR: LOR M, L Δ C

Function: Compute the logical sum of the character specified by C and the least significant location of the field specified by M and L. The result replaces the least significant location of the field specified by M and L.

Notes:

a.) L specifies the length and should equal 1. If L is unequal to 1, the least significant location of M will be used to compute the logical sum.

b.) C specifies the character used to compute the logical sum and may be any one of the 63 valid UNIVAC 1005 characters. If no character is specified, a space will be used to compute the logical sum.

c.) The C character must be preceded by a space.

d.) For each one bit in the C character the corresponding bit position in M is set to one. For each zero bit in the C character the corresponding bit in M is retained.

The logical sum is formed based on the following truth table:

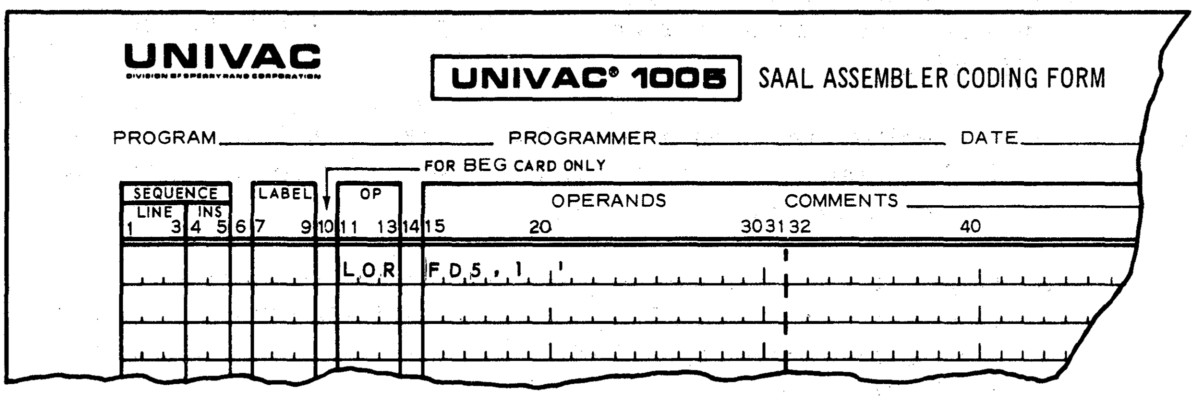
OR	0	1
0	0	1
1	1	1

i.e.,

C	$\oplus$*	(M)	$\rightarrow$	(M)
0	$\oplus$	0	=	0
0	$\oplus$	1	=	1
1	$\oplus$	0	=	1
1	$\oplus$	1	=	1

* $\oplus$ represents the logical sum

Example: Compute and store the logical sum of the character ' (apostrophe) and the one character field labeled FD5.



C (before and after)	100000	equals ' (apostrophe)
FD5 (before)	000110	equals +3
FD5 (after)	100110	equals –3

In the above example, the C character is used to add the "X" bit to FD5.

BIT SHIFT: BSH M, L

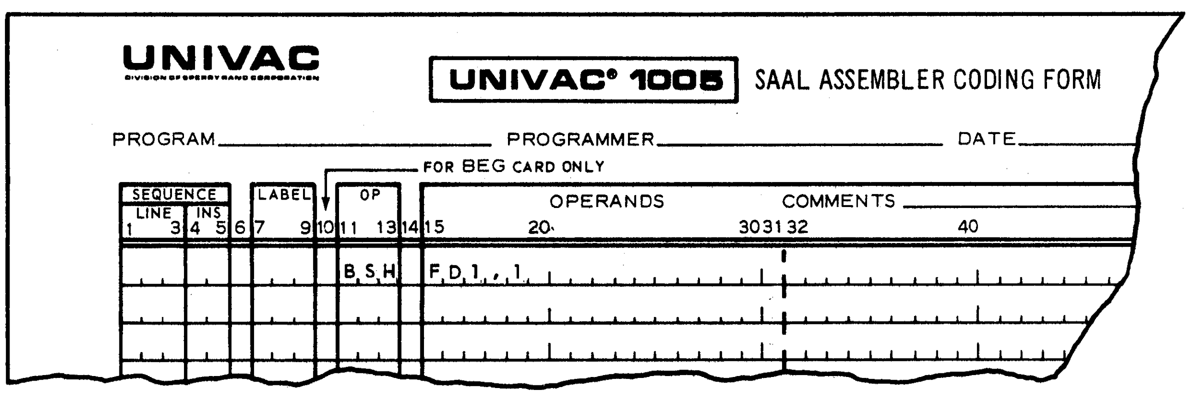
Function: Shift circularly one bit, the least significant location of the field specified by M and L.

Notes: a.) L specifies the length and should equal 1. If L is unequal to 1, the least significant character of M will be shifted.

b.) This is a binary circular shift and all data bits are considered. The "X" bit is shifted to the "1" bit, the "Y" bit is shifted to the "X" bit and so forth.

Original Bit	Shifted to Bit
X	1
Y	X
8	Y
4	8
2	4
1	2

Example: Shift circular one bit, the one character field FD1.



FD 1 (before) 011110 equals <
FD 1 (after) 111100 equals Z

F. INSTRUCTION REPERTOIRE - EXTERNAL FUNCTION COMBINATIONS

To provide a greater degree of flexibility, the External Function Combination instruction (XFC) augments the individual External Function (XF) instructions. In using this instruction, the programmer assigns the necessary machine codes for desired Input/Output combinations. This provides for Concurrent execution on the Reader or Auxiliary Reader, Printer, Punch or Read/Punch, Paper Movement and Program Halt.

The Card System External Function Combination instructions are explained in detail on the following pages. The instruction format depicts the bits absent necessary to perform Read, Print and Punch operations.

INSTRUCTION FORMAT XFC

COL. 16	COL. 17	COL. 18	COL. 19
X Y 8 4 2 1 B B B B B B 1 2 3 4 5 6	X Y 8 4 2 1 B B B B B B 1 2 3 4 5 6	X Y 8 4 2 1 B B B B B B 1 2 3 4 5 6	X Y 8 4 2 1 B B B B B B 1 2 3 4 5 6

COLUMN 16	'X'	B1	Always Present	Not Used
	'Y'	B2	Absent	Print Space 1
	'8'	B3	Absent	Print Space 2
	'4'	B4	Absent	Skip 1
	'2'	B5	Absent	Skip 2
	'1'	B6	Absent	Skip 4

COLUMN 17	'X'	B1	Always Present	Not Used
	'Y'	B2	Absent	Read
	'8'	B3	Absent	Read/Auxiliary
	'4'	B4	Absent	Read/Read Punch
	'2'	B5	Absent	Punch
	'1'	B6	Absent	Halt

COLUMN 18	'X'	B1	Absent	Stacker Select 2 - Aux. Reader
	'Y'	B2	Absent	Stacker Select 3 - Aux. Reader
	'8'	B3	Absent	1. Stacker Select - Punch 2. Paper Tape Parity Punch
	'4'	B4	Absent	Paper Tape Read 1 Frame
	'2'	B5	Absent	Paper Tape Read Through Sentinel
	'1'	B6	Absent	Paper Tape Read 80 Frames

COLUMN 19	'X'	B1	Absent	Paper Tape Punch 1 Frame
	'Y'	B2	Always Present	Not Used
	'8'	B3	Absent	Read Code Image
	'4'	B4	Absent	Punch Code Image
	'2'	B5	Absent	Paper Tape Punch To Sentinel
	'1'	B6	Absent	Paper Tape Punch Channel 8

A table to determine the codes necessary for many combinations follows:

	Function	CARD COL. 16	CARD COL. 17	CARD COL. 18	CARD COL. 19
Group 1	Print and Space 1	Δ	)	)	)
	Print and Space 2	U	)	)	)
	Skip 1	Y	)	)	)
	Skip 2	⌑	)	)	)
	Skip 3	W	)	)	)
	Skip 4	>	)	)	)
	Skip 5	X	)	)	)
	Skip 6	Z	)	)	)
	Skip 7	V	)	)	)
	Print and Skip 1	Q	)	)	)
	Print and Skip 2	(	)	)	)
	Print and Skip 3	⊙	)	)	)
	Print and Skip 4	@	)	)	)
	Print and Skip 5	P	)	)	)
	Print and Skip 6	R	)	)	)
	Print and Skip 7	N	)	)	)
Group 2	Read	)	Δ	)	)
	Read Code Image	)	Δ	)	U
	Read Auxiliary Stacker Select 1	)	U	)	)
	Read Auxiliary Stacker Select 2	)	U	=	)
	Read Auxiliary Stacker Select 3	)	U	Δ	)
	Read Auxiliary Code Image Stacker Select 1	)	U	)	U
	Read/Read Punch	)	W	)	)
	Read/Read Punch Stacker Select	)	W	U	)
	Read/Read Punch Code Image	)	W	)	Y
	Punch	)	⌑	)	)
	Punch and Stacker Select	)	⌑	U	)
	Punch Code Image	)	⌑	)	Y
	Halt	)	>	)	)
	Read and Punch	)	(	)	)
	Read and Halt	)	@	)	)

	Function	CARD COL. 16	CARD COL. 17	CARD COL. 18	CARD COL. 19
Group 2 (cont'd.)	Read, Punch and Halt	)	R	)	)
	Punch and Halt	)	Z	)	)
Group 3	Read Paper Tape 1 Frame	)	)	Y	)
	Read Paper Tape 1 Frame Code Image	)	)	Y	U
	Read Paper Tape 80 Frames	)	)	>	)
	Read Paper Tape 80 Frames Code Image	)	)	>	U
	Read Paper Tape through Sentinel	)	)	⌑	)
	Read Paper Tape through Sentinel Code Image	)	)	⌑	U
Group 4	Punch Paper Tape 1 Frame	)	⌑	)	=
	Punch Paper Tape 1 Frame with Parity	)	⌑	U	=
	Punch Paper Tape to Sentinel	)	⌑	)	⌑
	Punch Paper Tape to Sentinel with Parity	)	⌑	U	⌑

EXTERNAL FUNCTION COMBINATIONS: XFC nnnn

Function: This instruction augments the individual External Function Instructions. In using this instruction, the programmer assigns the necessary machine codes for desired Input/Output combinations.

Notes:

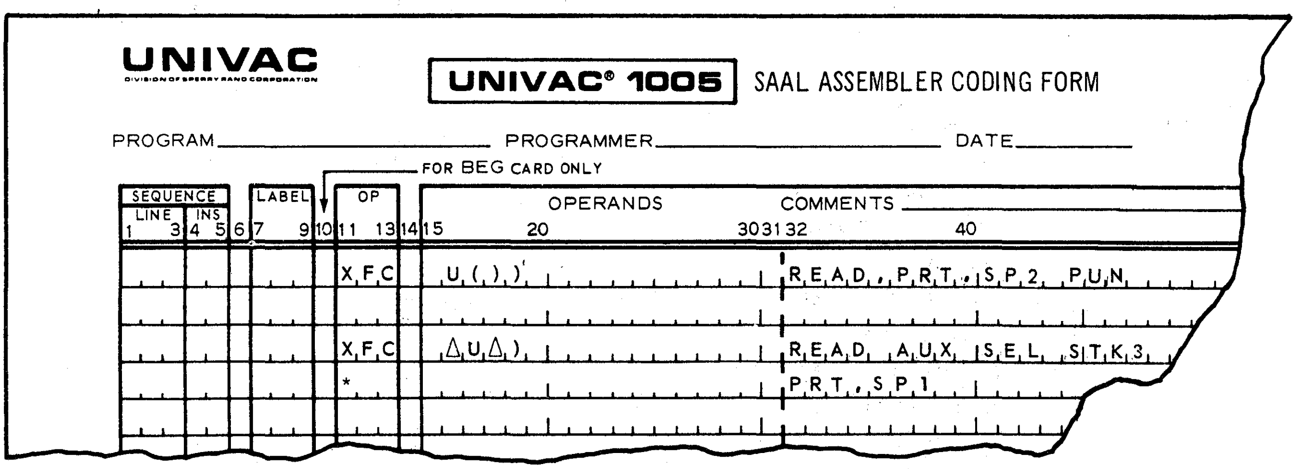
a.) XFC is the mnemonic operation entered in card columns 11-13.

b.) The machine code operand field must be preceded by a space in card column 15.

c.) The applicable I/O function codes are entered in card columns 16-19.

To use the table, select all applicable I/O functions to be performed upon execution of the XFC instruction.

Example:


UNIVAC

UNIVAC® 1005 SAAL ASSEMBLER CODING FORM

PROGRAM ______ PROGRAMMER ______ DATE ______

FOR BEG CARD ONLY

SEQUENCE LINE 1 3	INS 4 5	6	LABEL 7 9	10	OP 11 13	14	OPERANDS 15 20 30 31	COMMENTS 32 40
					XFC		U())	READ, PRT, SP2 PUN
					XFC		ΔUΔ)	READ AUX SEL STK3
					*			PRT, SP1

G. INSTRUCTION REPERTOIRE - 1005 DATA LINE TERMINAL-3 EXTERNAL FUNCTIONS and CONDITIONAL TESTS

1. DLT-3 External Functions

The Data Line Terminal-3 is an optional feature to the 1005 that enables the 1005 to communicate via telephone circuits while processing. This ability is provided by utilizing independent control and buffering circuitry. Data is transmitted at the rate governed by the modem employed. DLT-3 used by the 1005 may communicate with a 1004 having either a DLT-1 or a DLT-3, another 1005 with DLT-3 and any other compatible device.

The 1005, with this feature, will process data and transmit or receive data simultaneously.

> Note: Input/Output operations are specifically excluded from overlap, i.e., do not execute any XF functions between the Send or Receive instruction and the Pause Test instruction.

The same principle of simultaneous execution and time-sharing of storage applies to DLT operations as it does to reading, printing and punching, except that DLT-3 is not instruction dependent. Whereas reading and printing are preformed entirely during a single instruction execution, DLT operation can occur throughout many instructions, as does the punching operation. A PTE instruction (Pause Test) serves to interlock the processor if the DLT is transmitting or receiving.

2. General

Both equipments, to communicate, must have the DLT option. Assuming they are both 1005's, and have DLT-3, they must both be using the same type of data set. The data sets are used in the half-duplex mode, i.e., communication can be in one direction only, at one time. Both the transmitting and receiving functions may take place independently of, and concurrently with data processing functions. The maximum rates of data transmission are: the 201A Data Set - 2000 bits per second; the 201B Data Set - 2400 bits per second. The DLT circuits use a 7-bit character - 6 data bits and 1 parity bit.

The DLT-3 storage area is simi-fixed, and of variable length. The beginning location is Module 1 position 0435. The ending location may be Module 1 position 0434 with automatic wrap around from 0961 to 0001, i.e., transmission is fixed to 961 characters. The transfer from DLT storage to the Data Set will be descending in a continuous sequence. The message length is controlled by the program when transmitting. When receiving, the End of Message character received will halt the descending locations. The send/receive buffers, may be used for internal processing. Precaution should be observed to prevent internal processing from prematurely changing the data to be transmitted (or the Data received).

A prescribed transmission format must be used in all communications. The message (useable data) must be preceded by a least four synchronization characters (the letter S in UNIVAC XS-3 code); and one character of no bits. The Send 80 message must be followed by an End of Message character (the letter B in UNIVAC XS-3 code); and one character of no bits.

The Send through Sentinel message must be followed by a sentinel character, (the character ")" in UNIVAC XS-3 code), an EOM character and one character of no bits.

The storing of these characters is the responsibility of the programmer. All of this information must be in the storage area beginning at Module 1 position 0435 during each transmission. When receiving an 80 character transmission from another 1005, only the message (useable data), the EOM character, and the Longitudinal Parity character will be stored in the sequentially allocated DLT storage area beginning with Module 1 position 0435. When receiving more than 80 characters from another 1005, the message, the sentinel character, the EOM character and the LP character will be stored sequentially. The LPC is automatically placed in the no bits position following the EOM character by the transmitting 1005 and will vary depending upon the total bit content of the message. Receiving will terminate automatically when the EOM and the LPC characters are stored.

Error detection is provided in the form of transverse parity, longitudinal parity, and incomplete-message checking. In the event of abnormalities, an error signal is provided for the program to test or ignore. The error instructions should be used to alter the program sequence to effect corrective action.

3. Transmitting

Before each transmission, the message data is assembled in DLT storage:

1) The program must place four synchronization characters (letter S UNIVAC XS-3 code) initiated in the data division in Module 1 positions 0435 through 0438.

2) The program must place a no-bits character (Space, UNIVAC XS-3 code) in Module 1 position 0439.

3) To send 80 characters, the program must place the message (useable data) from Module 1 positions 0440 to 0519. No Sentinel is required and the character ")" is permissible within the message.

4) To send other than exactly 80 characters, the program must place the message from Module 1 position 0440 to any length less than 955 positions with a Sentinel immediately following the last character of useable data. The character ")" is not permissible within the useable data.

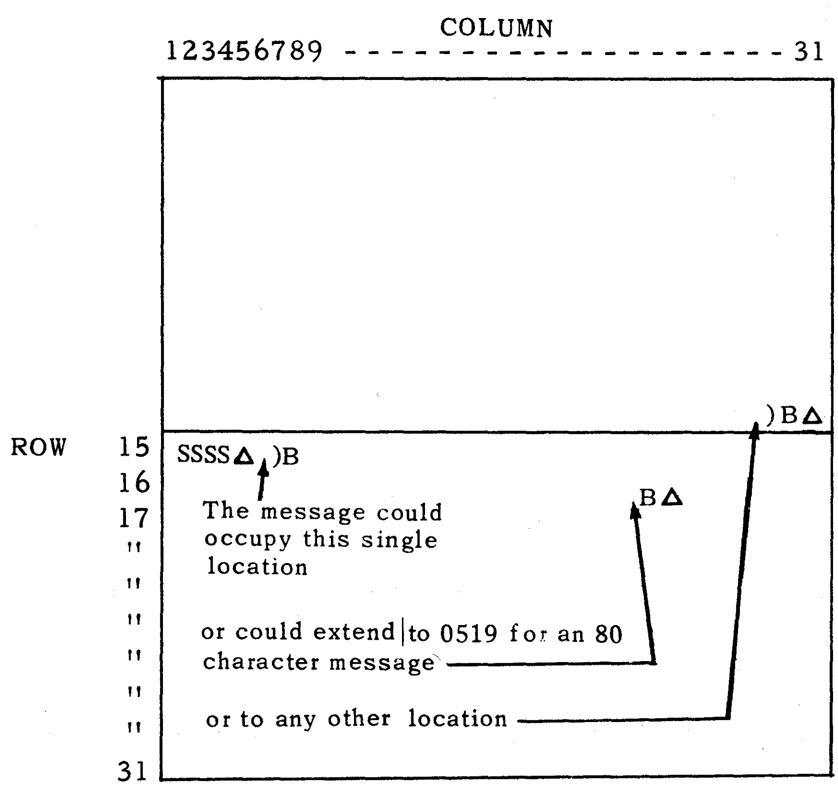
5) The program must place an End of Message character, (letter B UNIVAC XS-3 code) initiated in the data division, immediately following the last character of useable data in an 80 character message and immediately following the Sentinal character in all other messages.

6) The program must place a no-bits character (Space, UNIVAC XS-3 code) immediately following the End of Message character.

The 80 character message area per transmission is therefore at least six locations greater than the message length and all other are seven greater.

Illustrated in Figure 1 is the format of a DLT-3 message and the allocation of DLT-3 storage.

FIGURE 1



After assembly of all information based on the above recommendations, utilization of the transmit instruction may be effected.

4. Receiving

No receiving format is required and any information in the receive area will be overlaid by the incoming message. The first character to enter storage in the receiving 1005 will be the first message character. The synchronization characters and the Start of Message space, initially transmitted by the other machine, will not enter storage. The first message character will enter Module 1 position 0435; all remaining message characters will be stored in a continuous descending sequence. The Sentinel or End of Message character will enter the location following the last message character. The Longitudinal Parity Character will follow the EOM character in storage.

A Receive operation is accomplished by the Receive DLT to EOM instruction. Once the receive operation is initiated in this manner, the 1005 may proceed to succeeding instructions. The DLT circuits will wait for the first character and then store the message as it is received. When the LPC is received, this character is automatically compared with an LPC that is generated by the receiving 1005. Regardless of the results of this comparison, the LPC enters receive storage in the location following the EOM character. Upon entry of this character, the receive operation terminates.

5. Error Conditions

An error signal is available for testing should any of the following occur during a Receive operation:

1) One of the message characters is of even parity, and is not the EOM character.

2) The Receiving DLT does not synchronize on any of the synchronization characters.

3) The Receiving DLT does not complete the Receive order within 15 seconds.

4) The received LPC does not agree with the generated LPC.

5) The EOM character is not detected, or is incorrect.

Of the above five error conditions, the first one will result in less than expected storage used, with the properly received message characters in their respective locations, followed by the improper character. The second error type will result in nothing being entered into Receive storage; after 15 seconds the Receive operation will terminate. The third condition can be caused by no transmission, and will result in nothing being entered into Receive storage. The fourth condition will result in all expected Receive storage being filled, and an improper LPC. The fifth error type might result in more than the allocated storage being used. If the EOM character is received as an odd parity B, due to loss of the parity bit, it will be transferred to memory and the DLT will continue to look for more data. If the LPC also happens to be of an odd bit configuration, this too will enter Receive storage. There should be no further data reception, but noise in the transmission system might result in the reception of another erroneous character, which will be entered into storage. Thus, one location more than expected may be used.

6. Instruction Formats External Functions

SEND DLT 80 CHARACTERS: XF ΔSN8

Function: This instruction sends 80 characters from the DLT buffer via telephone circuits to any other compatible device.

Notes: a.) The message format must be completed prior to this instruction.
b.) No operand is specified.
c.) The mnemonic operand field must be preceded by a space.

Example: Format the message and transmit 80 characters.

SEQUENCE LINE (1–3)	INS (4–5)	6	LABEL (7–9)	10	OP (11–13)	14	OPERANDS (15–31)	COMMENTS (32–50)
							IN DATA DIVISION	
			SYN		+4		SSSS	SYNCHRONIZATION CHRS
							IN PROCEDURE DIVI	SION
					LA1		SYN,4	SSSS TO SEND AREA
					PTE			PAUSE TEST
					SA1		BUF,4	BUF STARTS AT 0435
					SC		BUF+4,1	SPACE TO SEND AREA
					LPR		CRD+80	MOVE MESSAGE TO
					SPR		BUF+5,80	SEND AREA
					SC		BUF+85,1 B	EOM CHARACTER ADDED
					SC		BUF+86,1	SPACE AFTER EOM
					XF		SN8	TRANSMIT 80 CHAR.

SEND DLT THROUGH SENTINEL: XF ΔSNS

Function: This instruction sends from 1 to 953 characters from the DLT buffer via telephone circuits to any other compatible device.

Notes:
a.) The message format must be completed prior to this instruction.
b.) The XS-3 character ")" must immediately follow the message and is not a permissible character within the useable data.
c.) No operand is specified.
d.) The mnemonic operand field must be preceded by a space.

Example: Format the message and transmit 132 characters.

SEQUENCE LINE (1–3)	INS (4–5)	6	LABEL (7–9)	10	OP (11–13)	14	OPERANDS (15–31)	COMMENTS (32–)
							IN DATA DIVISION	
			SYN		4		SSSS	SYNCHRONIZATION CHRS
							IN PROCEDURE DIVISION	
					LA1		SYN,4	SSSS TO SEND AREA
					PTE			PAUSE TEST
					SA1		BUF,4	BUF STARTS AT 0435
					SC		BUF+4,1	SPACE TO SEND AREA
					LPR		MES,132	MOVE MESSAGE TO
					SPR		BUF+5,132	SEND AREA
					SC		BUF+137,1)	SENTINEL AFTER MESSAG
					SC		BUF+138,1 B	EOM CHARACTER
					SC		BUF+139,1	SPACE AFTER EOM
					XF		SNS	TRANSMIT 132 CHAR.

RECEIVE DLT TO EOM: **XF** ΔRCD

Function: This instruction receives data from the Data Line Terminal.

Notes:
a.) The first message character will enter Module 1 position 0435.
b.) Message characters will be stored in a continuous descending sequence.
c.) No operand is specified.
d.) The mnemonic operand field must be proceded by a space.

Example: Receive to end of message.

SEQUENCE LINE 1–3	SEQUENCE INS 4–5	6	LABEL 7–9	10	OP 11–13	14	OPERANDS 15–31	COMMENTS 32–
					XF		RCD	RECEIVE TO EOM

In the example above, the 1005 could receive from 1 to 953 characters.

7. Instruction Formats Conditional Tests

Associated with the DLT-3 system are three (3) conditional instructions which allow the programmer to test for ready, interlocked and error conditions.

The 1005 DLT-3 Conditional Test instructions pertain to Class II and are explained in detail.

a) Pause Test: PTE

b) Function: This instruction tests the ready status of the DLT-3. Control will not be transferred to the next instruction in sequence if the DLT-3 is still active.

Notes: a.) This instruction is given following a transmit or receive command and prior to the first transfer of new data into the DLT buffer.

b.) This instruction insures that information will not be transferred into the DLT buffer while it is in the process of transmitting or receiving.

c.) Optimum utilization of the Pause Test instruction will provide the maximum overlap of processing with DLT operations.

Example: Test the DLT buffer before moving the incoming message to print area.

SEQUENCE LINE 1–3	INS 4–5	6	LABEL 7–9	10	OP 11–13	14	OPERANDS 15–31	COMMENTS 32–
					PTE			PAUSE TEST
					LPR		BUF,80	MESSAGE TO PRINT

JUMP END OF TIME: JET M

JUMP PARITY ERROR: JPE M

Function: Transfer program control to the instruction stored at M if the condition specified by the operation code is present.

Notes: a.) These instructions are used to test the status of the DLT-3 after execution of send or receive.

b.) If the condition tested is not present, control will not be transferred and the next instruction in sequence will be executed.

c.) Do not issue any Input/Output instructions between the receive instruction and the JPE instruction.

Example: Test the status of a previously executed Send or Receive instruction. If there was an error in the message or no message received in 15 seconds, transfer control to the routine labeled ERR.

SEQUENCE LINE 1–3	SEQUENCE INS 4–5	6	LABEL 7–9	10	OP 11–13	14	OPERANDS 15–31	COMMENTS 32–40
					JET		ERR	
					JPE		ERR	

CHAPTER 3
1005 SOFTWARE

I. THE UNIVAC 1005 SINGLE ADDRESS ASSEMBLY SYSTEM

Associated with a programming system is a machine language program called an Assembler. The Assembler accepts a program written in symbolic language (source program) and converts it into machine language (object program).

The symbolic language used by the UNIVAC 1005 Card Processing System is single address in design and is intended to provide an easy to learn, easy to use tool whereby data processing requirements can be translated into machine coded instructions.

The machine language program or assembly system associated with the UNIVAC 1005 symbolic language is called SAAL (Single Address Assembly Language). This assembly system consists of two passes, SAAL 1 and SAAL 2.

A. SAAL 1 (Illustration 1) Trial Balance Sample Program P2-4

The first pass, SAAL 1, relates each symbolic reference (label) in the symbolic program (source program) with its appropriate position in core memory. This relationship between symbolic labels in the source program and core memory position is retained in memory and utilized in SAAL 2. This relationship is commonly referred to as the "TAG" or "LABEL" Table.

1. Card Input - Original Symbolic Program

   The Symbolic Input Card format is as follows:

Card Columns	Description
1- 3	Sequence number
4- 5	Sequence number (insert)
* 7- 9	Label
11-13	Operation
**15-31	Operand
**32-48	Comments
62-65	Program I. D.

   * Two labels are prestored, AR1 and AR2. The programmer can reference these labels without prior definition.
   ** Literal instructions use columns 15-48 to generate constants.

2. Output

   a. Punched Card - None
   b. Printer - Listing of the label table relating each symbolic reference (label) in the symbolic program (source program) with its appropriate position in core memory.

The Label Table Listing format is as follows:

Description of Fields

SEQ #	- From source program
LBL	- From source program
LOC	- Assigned location of the label in memory
ERR	- Assigned error codes

NOTES - Possible errors are as follows:

1) ERR NO BEG CRD is printed, paper is advanced to the next page and the program halts - Indicates the BEG card does not precede the source program.

2) ERR OP IN DATA DIV is printed, paper is advanced to the next page and the program halts - Indicates an illegal directive, data description, literal or comment punched in the operation field.

3) DUP printed under ERROR heading - Indicates a duplicate label and is not stored in the label table.

4) >148 printed under ERROR heading - Indicates the maximum number of labels has been exceeded (148 labels).

5) OVM printed under ERROR heading - Indicates the maximum memory has been exceeded (3844 positions).

3. LABEL RESERVATIONS - The following labels are used by the SAAL Assembly System to define specific I/O functions. The programmer should exercise care that labels referenced as an external function (referenced in an XF instruction) are not duplicated as a line reference point or operand.

SK2	RPH	RPS	WT1	SN8
SK4	RCI	RP8	WT2	SNS
SK7	PCI	PP1	ER1	RCD
REA	RX1	PPS	ER2	
RPR	RXC		RW1	
RP2	RX2	P1P	RW2	
RPP	RX3	PSP	BS1	
PR1	PSS		BS2	
PR2	RRP	RT1	SI1	
PR7	RRC	RT2	SI2	
PUN	RRS	RT5	RI1	
HLT	RP1	RT6	RI2	

Example: The following coding will cause a duplicate label.

```
       XF    REA
REA    LA1   FD1
```

B. SAAL 2 (ILLUSTRATION 2) TRIAL BALANCE SAMPLE PROGRAM P2-4

The second pass, SAAL 2, interprets each operand field in the source program, determines its length and core position using the "LABEL" Table generated by SAAL 1, and produces a UNIVAC 1005 machine code object program deck. In addition, a one for one listing is prepared equating each symbolic line of coding in the source program with the generated machine code.

1. Card Input - Original symbolic cards.

2. Output.

a. Punched card - A one for one object deck which contains the original symbolic coding with generated pseudo-machine code and the UNIVAC 1005 machine code. Preceding this deck one load card is punched.

Card Columns	Description
1-48	Duplicated from input card
49-51	Card Code - Machine coded card column relating to the storage of data from the card.
52-57	Instruction - Machine coded instruction. The first position is the operation code and the next four are the operand. After every six instructions an additional character is assigned to indicate the next row.
58-61	Instruction address - Machine coded instruction address for each literal and instruction.
62-65	Duplicated from input card.

b. Printer - A one for one listing of each instruction written, in three different formats, the symbolic (original instruction), mnemonic (actual instruction), and machine (coded instruction) language.

The Machine Coded Listing format is as follows:

Description of Fields

SEQ#	- From source program
LBL	- From source program
OP	- From source program
OPERAND	- From source program
COMMENTS	- From source program
IDENT	- From source program
LOC	- Assigned pseudo address for each literal and instruction.
OPERAND	- Assigned pseudo address for the beginning and ending locations of each operand.
ERROR	- Assigned error codes
C/C	- Machine coded card column relating to the storage of data from the card.
INSTR	- Machine coded instruction. The first position is the operation code and the next four are the operand. After every six instructions, an additional character is assigned to indicate the next row.

Description of Fields

LOC - Assigned machine coded instruction address for each literal and instruction.

NOTES - Possible errors are as follows:

1) Program Halts after first card is read - Indicates BEG card does not precede source program.
2) ' O ' printed under 1st position of ERROR heading - Indicates an illegal operation code.
3) ' E ' printed under 2nd position of ERROR heading - Indicates an expression error, i.e. operand which is less than 0001 or greater than 3875. The most frequent cause of error is an undefined label. This type of error will print 6530 under the OPERAND heading.
4) ' P ' printed under 3rd position of ERROR heading - Indicates a precautionary warning, i.e. an instruction greater than 10 or 21 characters utilizing AR1 or AR2 respectively.
5) ' S ' printed under the 4th position of ERROR heading - Indicates a sequence number error.

C. Trial Balance Sample Report P2-4 (Illustration 4)

This program prepares a Trial Balance Tabulation and punches Trial Balance cards utilizing sorted General Ledger Account cards.

1. Card Input - Sorted General Ledger Account cards.

The Input Card format is as follows:

Card Columns	Description	Remarks
1	Type	Determine card columns of amount field. 1/1 indicates amount in Cols. 59-68; 2/1 indicates amount in Cols. 70-79.
6- 7	Program Number	Major control for this report. Each control break prints the amount accumulated and is reset prior to the next total being accumulated. Card Col. 7 is not printed.
55-58	Account Number (Note 1)	Minor control for this report. A Trial Balance Summary card is punched for each Account Number.
59-68	Account "1" (Note 2)	This amount is accumulated if the card contains a "1" in Col. 1.

Card Columns	Description	Remarks
70-79	Amount "2" (Note 2)	This amount is accumulated if the card contains a "2" in Column 1.

NOTE 1 - An "X" overpunch in Col. 55 indicates a credit account and the amount is accumulated in the credit field.

NOTE 2 - An "X" overpunch in Col. 55 or 70 indicates a credit amount and is accumulated as such in either the debit or credit account field.

2. Output

a. Punched card - A Trial Balance Summary Card is punched for each Account Number within Program Number.

Card Column	Description
2- 5	Julian date
6- 7	Program Number
55-58	Account Number
59-68	Amount

b. Printer - Trial Balance Tabulation

The Trial Balance Tabulation format is as follows:

	Description of Fields
P #	- From input
ACCT #	- From input
Debit	- Accumulated and printed on control break
Credit	- Accumulated and printed on control break
Cumulative Balance	- Accumulated and printed on control break

II. The UNIVAC 1005 Single Address Report Generator

SARGE, a problem oriented programming system and report program generator, is designed to reduce substantially the time and effort necessary to translate general data processing and reporting requirements into detailed computer instructions. It demands little knowledge of computer coding or instructions other than the basic rules for writing in the simplest form of the SAAL assembly language. Essentially, the SARGE report program generator is a program which, on the basis of a series of statements provided to it, produces another program which will produce a report or other output of the desired kind. These statements, written on the standard SAAL coding form and then keypunched into cards, provide the formats of the input card files (these contain the information from which the report is to be prepared), the format of the output to be produced (this may be a printed document, a series of summary cards, or both), and the operations to be performed (arithmetic operations, data movement and editing, control, input/output operations). The input and output format descriptions and processing statements will, in conjunction with SAAL, produce an efficient ready to run object program. Also provided is a listing of source input and the object coding generated. Sections of programmer's own code may be included as necessary.

A. SARGE 1

On the first pass SARGE 1 reproduces the symbolic program (source program) as comments cards. For each reproduced comments card, one or more SAAL statements are generated. Any card not recognized as a SARGE statement is reproduced without change.

1. Card Input - Original symbolic program

   The symbolic input card format is as follows:

Card Columns	Description
1-3	Sequence number
4-5	Sequence number (insert)
*7-9	Label
11-13	Operation
15-48	Operand
32-48	Comments
62-65	Program identification

*The following labels are reserved for the generator and may not be used by the programmer:

AR1	REA
AR2	RPP
HLT	RPR
PR1	SK2
PR2	SK4
PR7	SK7
PUN	XXX
	XØ1 thru X99

2. Output

   a. Punched Card - SARGE input reproduced as comments cards with associated SAAL statements.

   b. Printer - None

B. SARGE 2 (Illustration 4) Trial Balance Sample Program P2-4

The second pass, SARGE 2, produces the pseudo-machine code for all labels describing the input/output buffer areas. The length is added to all labels describing constants and working storage.

1. Card Input - Output cards from SARGE 1

2. Output

   a. Punched card - A complete program deck ready for the SAAL assembly.

Card Columns	Description
1-5	Sequence number beginning with 5ØØØØ
7-9	Label
11-13	Operation
15-48	Operand
32-48	Comments
62-65	Program identification

   b. Printer - A listing of the source input preceded by an asterisk and the object coding generated.

Print Positions	Description
1-5	Sequence number beginning with 5ØØØØ
7-9	Label
11-13	Operation
15-48	Operand
32-48	Comments
62-65	Program identification

NOTES - Possible errors are as follows:

1) An E (print position 85) printed to the right of an input/output label definition indicates that the maximum of 68 input/output labels has been exceeded.

2) An E (print position 85) printed to the right of a constant or working storage definition indicates that the maximum of 5Ø labels has been exceeded.

III. UTILITY ROUTINES

A. CONDENSE

Condenses object programs produced by SAAL 3, consolidating 6 instructions to a card. All literal instructions are punched one for one.

1. Card Input - Object program produced by SAAL 2 in the same sequence.

2. Output

   a. Punch Card - Consolidated object program

Card Columns	Description
1 - 3	Sequence number
15 - 48	Consolidated instructions or literal
49 - 61	Machine Code
62 - 65	Program I. D.

   b. Printer

      1) Successful termination - END OF PROGRAM is printed, paper is advanced to next page and the program halts.

      2) Possible errors are as follows:

      ERROR NO BEG CARD is printed, paper is advanced to next page and the program halts. This error indicates the BEG card does not precede all object cards or does not immediately follow the load card produced from SAAL 2 (2nd object card).

      ERROR INCORRECT INSTR CODE is printed, paper is advanced to next page and the program halts. This error indicates an instruction stored in an invalid location. All instructions must be stored beginning in Columns 1, 6, 11, 16, 21 or 26. The most frequent cause of this type of error is incorrectly repunching an object program card.

      Notes:

      1. The Program I. D. from the BEG card is gang punched in all succeeding cards.

      2. All condensed cards are numbered successively beginning with 001.

      3. The cards to be condensed must be in the correct sequence.

B. MEMORY DUMP (Illustration 5)

Each row of core memory is printed in sequence with a row and bank identification annotated.

1. Card Input - Memory dump object program

2. Output

   a. Punched card - None
   b. Printer - Memory listing

      NOTE - Data in the print buffer will be printed as the first line across the page and data in the read buffer will be lost. The only memory that will be printed is the memory addressable by the programmer.

C. READ-PRINT-PUNCH

Produces and prints each card, column for column, in the first 80 positions of the printer.

1. Card Input - Any data cards

2. Output

   a. Punched card - Reproduced data cards.
   b. Printer - 80/80 listing of data cards.

      NOTE - Punching will be suppressed when alternate switch 4 is on.

D. NUMBER IT

Re-numbers program cards with option of gang punching new program identification.

1. Card Input - Source or object program cards.

2. Output

   a. Punched card - Duplicate input cards re-numbering them starting with 001 (Cols. 1-3)
   b. Printer - None

      NOTE - To reidentify a program, precede the program cards with a header card punched as follows:

Card Columns 11-13	***
Card Columns 62-65	New Program I. D.

E. DUPLICATE

Reformates and prints any 80 columns of information in any other 80 columns with or without gang punching.

1. Card Input - Any data cards preceded by four header cards (see notes).

2. Output

   a. Punched card - Reformatted data cards
   b. Printer - 80/80 listing of reformatted data cards

NOTES:

1. The first header card contains information that is desired in all the following cards. If gang punching is not desired, this card must be blank.

2. The second and third header cards are divided into eighty sequentially numbered fields of two columns each. These cards describe the output card by indicating the column from which the input will be transferred.

   For example:

Card Column	Punch With
1- 2	01
3- 4	Blank
5- 6	05
7- 8	04
9-10	03
11-12	06
↓	↓
159-160	80

   Will reproduce the card identically to the original except that Cols 3 and 5 will be punched into Cols. 5 and 3 and card column 2 will be blank.

3. The fourth header card is literally a duplicate of the card that will be recognized as a sentinel. For example if a blank card were introduced as the fourth header the program would terminate when a second blank card was read.

4. Printing may be eliminated by changing the Duplicate object program. Column 16 of card number 43 (Cols. 4-5) may be changed from Δ to) and Column 31 of card number 45 may be changed from E to).

F. CLEAR

Clears Bank 1 thru 4 core to spaces

1. Card Input - Clear object program

2. Output - None

SAAL 1
1st PASS OF ASSEMBLY SYSTEM

ILLUSTRATION 1-1
REFER TO CHAPTER 3-I-A

P2-4

SEQ #	LBL	LOC	ERR	SAAL1
004	FD1	0001		
005	FD3	0006		
006	FD4	0055		
007	FD5	0059		
008	FD6	0070		
010	PN0	0161		
011	ACT	0164		
012	DEB	0172		
013	CRE	0188		
014	BAL	0236		
016	PUN	0293		
017	DTE	0294		
018	PNU	0298		
019	ACN	0347		
020	AMT	0351		
022	DAT	0081		
023	CNT	0085		
024	DLR	0090		
025	ASK	0091		
026	HLD	0092		
027	PV	0096		
028	ID1	0098		
029	ID0	0099		
030	IND	0100		
031	CRT	0101		
038	HD1	0373		
039	HD2	0380		
040	HD3	0390		
043	AC1	0477		
044	AC2	0487		
045	AC3	0497		
046	AC4	0507		
047	AC6	0517		
048	ST0	0527		
050	ST1	0528		
054	FST	0548		
056	RT2	0559		
064	ON	0600		
066	MO1	0610		
075	NX1	0657		
078	MO2	0672		
083	ALT	0698		
085	MO3	0708		
089	CON	0729		
104	ZZ	0807		
110	WW	0838		
116	MO4	0869		
132	RTN	0951		
133	MO5	0956		
138	BK1	0982		
153	YY	1060		
159	RT4	1091		
170	BK2	1148		
176	OF2	1179		
183	TAG	1215		
184	OFL	1220		
193	LST	1266		

SAAL 2
2nd PASS OF ASSEMBLY SYSTEM

ILLUSTRATION 2-1
REFER TO CHAPTER 3-I-B

```
SEQ # LBL OP  OPERAND              COMMENTS          IDEN LOC  OPERAND        ERR   C/C   INSTR   LOC     SAAL 3
001       BEG                                        P2-4 152HA                     &
002       CRD                                        P2-4 152HA                     &
003   FD1 -   1                                      P2-4 152HA                     &
004   FD3 -   6                                      P2-4 152HA                     &
005   FD4 -   55                                     P2-4 152HA                     &
006   FD5 -   59                                     P2-4 152HA                     &
007   FD6 -   70                                     P2-4 152HA                     &
008       PRT                                        P2-4 152HA                     &
009   PNO -   1                                      P2-4 152HA                     &
010   ACT -   4                                      P2-4 152HA                     &
011   DEB -   12                                     P2-4 152HA                     &
012   CRE -   28                                     P2-4 152HA                     &
013   BAL -   76                                     P2-4 152HA                     &
014       PCH                                        P2-4 152HA                     &
015   PUN -   1                                      P2-4 152HA                     &
016   DTE -   2                                      P2-4 152HA                     &
017   PNO -   6                                      P2-4 152HA                     &
018   ACN -   55                                     P2-4 152HA                     &
019   AMT -   59                                     P2-4 152HA                     &
020       ORG 0081                                   P2-4 152HA                     &
021   DAT +4  0865                                   P2-4 0081A                     & B           0<0C
022   CNT +5  52  1                                  P2-4 0085A                     & B           0\0?
023   DLR +1  $                                      P2-4 0090A                     & B           0303
024   ASK +1  *                                      P2-4 0091A                     & B           0909
025   HLD +4                      ACCT NO            P2-4 0092A                     & B           0E4]
026   PV  +2                      COLS 6 & 7         P2-4 0096A                     & B           4044
028   ID1 +1  1                                      P2-4 0098A                     & B           4;4;
029   ID0 +1  0                                      P2-4 0099A                     & B           4I4I
030   IND +1  0                   PRINTED ACCT BAL IND P2-4 0100A                   & B           4F4F
031   CRT +2  CR                                     P2-4 0101A                     & B           4.41
032       ORG 0161                                   P2-4 0101A                     &
0321      +34                                        P2-4 0161A                     & B           IIF.
033       +34            TRIAL BALANCE               P2-4 0195A                     & B           F1.:
0331      +34                                        P2-4 0229A                     & B           .-17
0332      +30                                        P2-4 0263A                     & B           1B52
034       ORG 0373                                   P2-4 0263A                     &

035   HD1 +7  P  ACCT                                P2-4 0373A                     & B           ? ?F
036   HD2 +10 CUMULATIVE                             P2-4 0380A                     & B           ?.?D
037   HD3 +34 #    #              DEBIT              P2-4 0390A                     & B           ?[7#
038       +34  CREDIT                                P2-4 0424A                     & B           7HB\
039       +19             BALANCE                    P2-4 0458A                     & B           BG8:
040   AC1 +10                                        P2-4 0477A                     & B           8-8H
041   AC2 +10                                        P2-4 0487A                     & B           8C8&
042   AC3 +10                                        P2-4 0497A                     & B           D D5
043   AC4 +10                                        P2-4 0507A                     & B           D:D#
044   AC6 +10                                        P2-4 0517A                     & B           DHDE
045   STO +1                                         P2-4 0527A                     & B           D&D&
046       STA                SAAL                    P2-4 0527A                     &
047   STT XF   PR2           TITLE                   P2-4 0528A PR2                 &]H   &D)))   [ [;
048       CLR PUN,80                                 P2-4 0533A0293 0372            &]H   157-&   [I[5
049       JR  OF2            TO COLUMN HEADINGS SUBRTN P2-4 0538A1179 1183          &]H   D, ,;   [:[B
050       XF   REA           READ FIRST CARD         P2-4 0543A REA                 &]H   &)Δ))   [8[#
051   FST LD1 FD3,2          STORE COLS 6 & 7        P2-4 0548A0006 0007            &]H   ] I F   [H[A
052       SD1 PV,2                                   P2-4 0553A0096 0097            &]H   ;4044<  [6[&
053   RT2 LD1 FD4,4          STORE ACCT NO           P2-4 0559A0055 0058            &]H   ]]G]?   < <;
054       SD1 HLD,4                                  P2-4 0564A0092 0095            &]H   ;0E4]   <I<5
055       LA1 ID1,1          COMPARE COL 1 TO ONE    P2-4 0569A0098 0098            &]H    4;4;   <:<B
056       CN1 FD1,1                                  P2-4 0574A0001 0001            &]H   :       <8<#
058       JL  ON             IF 2                    P2-4 0579A0600 0604            &]H   7#:#B   <H<A
059       LA2 FD5,10         PICK UP AMT COLS 59-68  P2-4 0584A0059 0068            &]H   ']30I#  <6<&
060       LD1 FD5,1                                  P2-4 0590A0059 0059            &]H   ]]3]3   # #;
061       J   MO1                                    P2-4 0595A0610 0614            &]H   2#H#A   #I#5
062   ON  LA2 FD6,10         PICK UP AMT COLS 70-79  P2-4 0600A0070 0079            &]H   '0.0D   #:#B
063       LD1 FD6,1                                  P2-4 0605A0070 0070            &]H   ]0.0.   #8##
064   MO1 SD1 STO,1          STORE MSL OF AMT        P2-4 0610A0527 0527            &]H   ;D&D&   #H#A
065       LA1 ID0,1                                  P2-4 0615A0099 0099            &]H    4I4IH  #6#&
066       CN1 HLD,1          CHECK CR-DEB ACCT       P2-4 0621A0092 0092            &]H   :0E0E   H H;
067       JL  MO2            IF DEBIT                P2-4 0626A0672 0676            &]H   7CHCA   HIH5
068       CN1 STO,1          CHECK CR-DEB AMT        P2-4 0631A0527 0527            &]H   :D&D&   H:HB
069       JG  NXT                                    P2-4 0636A0657 0661            &]H   BCIC5   H8H#
070       AM2 AC3,10         DEBIT AMT, CREDIT ACCUM P2-4 0641A0497 0506            &]H   >D D5   HHHA
071       AM2 AC4,10                                 P2-4 0646A0507 0516            &]H   >D:D#C  H6H&
072       J   MO3                                    P2-4 0652A0708 0712            &]H   2\6\E   C C;
```

ILLUSTRATION 2-2

```
073   NX1 SM2 AC3,10       CREDIT AMT, CREDIT ACCUM         P2-4 0657A0497 0506          &JH   YD D5     CIC5
074       SM2 AC4,10                                        P2-4 0662A0507 0516          &JH   YD:D#     C:CB
075       J   MO3                                           P2-4 0667A0708 0712          &JH   2\6\E     C8C#
076   MO2 CN1 STO,1        CHECK CR-DEB AMT                 P2-4 0672A0527 0527          &JH   :D&D&     CHCA
077       JG  ALT                                           P2-4 0677A0698 0702          &JH   B\8\#\    C6C&
078       AM2 AC1,10       DEBIT AMT, DEBIT ACCUM           P2-4 0683A0477 0486          &JH   >8-8H     \ \;
079       AM2 AC2,10                                        P2-4 0688A0487 0496          &JH   >8C8&     \I\5
080       J   MO3                                           P2-4 0693A0708 0712          &JH   2\6\E     \:\B
081   ALT SM2 AC1,10       CREDIT AMT, DEBIT ACCUM          P2-4 0698A0477 0486          &JH   Y8-8H     \8\#
082       SM2 AC2,10                                        P2-4 0703A0487 0496          &JH   Y8C8&     \H\A
083   MO3 XF   REA         READ NEXT CARD                   P2-4 0708A REA               &JH   &)Δ))G    \6\&
084       LA1 FD4,4                                         P2-4 0714A0055 0058          &JH    ]G]?     G G;
085       CN1 HLD,4        COMPARE NEW CARD ACCT NO         P2-4 0719A0092 0095          &JH   :0E4]     GIG5
086       JE  RT2                                           P2-4 0724A0559 0563          &JH   8< <;     G:GB
087   CON LD2 PV,2         IF BREAK, INFO TO PRINT & PUN    P2-4 0729A0096 0097          &JH   *4044     G8G#
0871      PTE                                               P2-4 0734A                   &JH   E         GHGA
088       SD2 PNO,1                                         P2-4 0739A0161 0161          &JH   @IIIIA    G6G&
089       SD2 PNO,2                                         P2-4 0745A0298 0299          &JH   @5<5#     A A;
0891      LD1 DAT,4                                         P2-4 0750A0081 0084          &JH   ]0<0C     AIA5
0892      SD1 DTE,4                                         P2-4 0755A0294 0297          &JH   ;5B5[     A:AB
090       LN1 HLD,4                                         P2-4 0760A0092 0095          &JH   30E4]     A8A#
091       SA1 ACT,4                                         P2-4 0765A0164 0167          &JH   4I1I-     AHAA
092       SA1 ACN,4                                         P2-4 0770A0347 0350          &JH   4-I-16    A6A&
0921      CLR AR2,21                                        P2-4 0776A1933 1953          &JH   1):)&     6 6;
093       LN1 AC1,10                                        P2-4 0781A0477 0486          &JH   38-8H     6I65
0931      CN2 AR1,10                                        P2-4 0786A1923 1932          &JH   %) )5     6:6B
094       JE  ZZ                                            P2-4 0791A0807 0811          &JH   8? ?;     686#
095       LWS AC1,10                                        P2-4 0796A0477 0486          &JH   ?8-8H     6H6A
096       SED DEB+1,14                                      P2-4 0801A0173 0186          &JH   RI[I&?    666&
097   ZZ  CLR AR2,21                                        P2-4 0807A1933 1953          &JH   1):)&     ? ?;
098       LN1 AC3,10                                        P2-4 0812A0497 0506          &JH   3D D5     ?I?5
0981      CN2 AR1,10                                        P2-4 0817A1923 1932          &JH   %) )5     ?:?B
099       JE  WW                                            P2-4 0822A0838 0842          &JH   83 3;     ?8?#
100       LWS AC3,10                                        P2-4 0827A0497 0506          &JH   ?D D5     ?H?A
101       SED CRE,15                                        P2-4 0832A0188 0202          &JH   RF]F83    ?6?&
102   WW  LA1 AC1,10       DEB-CRED = BAL OF ACCT NO        P2-4 0838A0477 0486          &JH    8-8H     3 3;
103       SR1 AC3,10                                        P2-4 0843A0497 0506          &JH   CD D5     3I35
```

```
104       SA1 AC6,10                                        P2-4 0848A0517 0526          &JH   4DHDE     3:3B
105       SA1 AMT,10                                        P2-4 0853A0351 0360          &JH   4-5-<     383#
106       JR  MO4                                           P2-4 0858A0869 0873          &JH   D9 9;     3H3A
107       J   MO5                                           P2-4 0863A0956 0960          &JH   2&6&E9    363&
108   MO4 JX  RTN          EDIT $, CR                       P2-4 0869A0951 0955          &JH   [&H&A     9 9;
109       LA1 DLR,1                                         P2-4 0874A0090 0090          &JH    0303     9I95
110       SA1 DEB,1                                         P2-4 0879A0172 0172          &JH   4IDID     9:9B
111       SA1 CRE,1                                         P2-4 0884A0188 0188          &JH   4F]F]     989#
112       SA1 BAL,1                                         P2-4 0889A0236 0236          &JH   4.<.<     9H9A
113       CLR AR2,21                                        P2-4 0894A1933 1953          &JH   1):)&F    969&
114       LN1 AC6,10                                        P2-4 0900A0517 0526          &JH   3DHDE     F F;
115       CN2 AR1,10                                        P2-4 0905A1923 1932          &JH   %) )5     FIF5
116       JE  RTN                                           P2-4 0910A0951 0955          &JH   8&H&A     F:FB
117       LWS AC6,10                                        P2-4 0915A0517 0526          &JH   ?DHDE     F8F#
118       SED BAL+1,14                                      P2-4 0920A0237 0250          &JH   R.#1]     FHFA
119       LN2 AC6,10                                        P2-4 0925A0517 0526          &JH   LDHDE&    F6F&
1191      CA2 AC6,10                                        P2-4 0931A0517 0526          &JH   NDHDE     & &;
1192      JEA RTN                                           P2-4 0936A0951 0955          &JH   8&H&A     &I&5
120       LA1 CRT,2                                         P2-4 0941A0101 0102          &JH    4.41     &:&B
121       SA1 BAL+14,2                                      P2-4 0946A0250 0251          &JH   41]10     &8&#
122   RTN J   $                                             P2-4 0951A0951 0955          &JH   2&H&A     &H&A
123   MO5 XF   PR1         PRINT ACCT TOTALS                P2-4 0956A PR1               &JH   &A)))'    &6&&
124       XF   PUN         PUNCH                            P2-4 0962A PUN               &JH   &)#))     ' ';
125       1C  CNT                                           P2-4 0967A0085 0089          &JH   -0\0?     'I'5
126       CLR IND,1                                         P2-4 0972A0100 0100          &JH   14F4F     ':'B
127       JE  OFL                                           P2-4 0977A1220 1224          &JH   8J:JB     '8'#
128   BK1 CLR AC1,10       CLR ACCT ACCUMS                  P2-4 0982A0477 0486          &JH   18-8H     'H'A
129       CLR AC3,10                                        P2-4 0987A0497 0506          &JH   1D D5*    '6'&
130       LA1 PV,2         COMPARE  COLS 6 & 7              P2-4 0993A0096 0097          &JH    4044     * *;
131       CN1 FD3,2                                         P2-4 0998A0006 0007          &JH   : I F     *I*5
132       JE  RT2                                           P2-4 1003A0559 0563          &JH   8< <;     *:*B
133       LA2 AC2,10       IF BREAK, INFO TO PRINT          P2-4 1008A0487 0496          &JH   '8C8&     *8*#
134       SR2 AC4,10       DEB-CRED = SECTION BAL           P2-4 1013A0507 0516          &JH   TD:D#     *H*A
135       SA2 AC6,10                                        P2-4 1018A0517 0526          &JH   MDHDE!    *6*&
136       JR  MO4                                           P2-4 1024A0869 0873          &JH   D9 9;     ! !;
137       CLR AR2,21                                        P2-4 1029A1933 1953          &JH   1):)&     !I!5
138       LN1 AC2,10                                        P2-4 1034A0487 0496          &JH   38C8&     !:!B
```

ILLUSTRATION 2-3

```
139       CN2 AR1,10                                P2-4 1039A1923 1932        &]H  %) )5     !8!#
140       JE  YY                                    P2-4 1044A1060 1064        &]H  8MIM5     !H!A
141       LWS AC2,10                                P2-4 1049A0487 0496        &]H  ?ACA&M    !6!&
142       SED DEB+1,14                              P2-4 1055A0173 0186        &]H  RI[I&     M M;
143   YY  CLR AR2,21                                P2-4 1060A1933 1953        &]H  1):)&     MIM5
144       LN1 AC4,10                                P2-4 1065A0507 0516        &]H  3D:D#     M:MB
145       CN2 AR1,10                                P2-4 1070A1923 1932        &]H  %) )5     M8M#
146       JE  RT4                                   P2-4 1075A1091 1095        &]H  8QIQ5     MHMA
147       LWS AC4,10                                P2-4 1080A0507 0516        &]H  ?D:D#Q    M6M&
148       SED CRE+1,14                              P2-4 1086A0189 0202        &]H  RFOF8     Q Q;
149   RT4 LA1 ASK,1          EDIT *                 P2-4 1091A0091 0091        &]H   0909     QIQ5
150       SA1 DEB+15,1                              P2-4 1096A0187 0187        &]H  4F F      Q:Q8
151       SA1 CRE+15,1                              P2-4 1101A0203 0203        &]H  4FDFD     Q8Q#
152       SA1 BAL+16,1                              P2-4 1106A0252 0252        &]H  41414     QHQA
153       XF   PR2           PRINT SECTION TOTALS   P2-4 1111A PR2             &]H  8D)))7    Q6Q&
154       LA1 ID1,1                                 P2-4 1117A0098 0098        &]H   4;4;     Z Z;
155       AM1 IND,1                                 P2-4 1122A0100 0100        &]H  <4F4F     ZIZ5
156       IC  CNT                                   P2-4 1127A0085 0089        &]H  -0\0?     Z:ZB
1561      JE  OFL                                   P2-4 1132A1220 1224        &]H  8J:JB     Z8Z#
157       IC  CNT                                   P2-4 1137A0085 0089        &]H  -0\0?     ZHZA
158       JE  OFL                                   P2-4 1142A1220 1224        &]H  8J:JBW    Z6Z&
159   BK2 CLR AC2,10         CLEAR ACCUMS           P2-4 1148A0487 0496        &]H  1ACA&     W W;
160       CLR AC4,10                                P2-4 1153A0507 0516        &]H  1D:D#     WIW5
161       CLR AR1,10         COMPARE FOR LAST CARD  P2-4 1158A1923 1932        &]H  1) )5     W:WB
162       CN1 FD4,4                                 P2-4 1163A0055 0058        &]H  :]G]?     W8W#
163       JE  LST                                   P2-4 1168A1266 1270        &]H  8N6NE     WHWA
164       J   FST                                   P2-4 1173A0548 0552        &]H  2[H[A,    W6W&
165   OF2 JX  TAG            PRINT COL HEADINGS     P2-4 1179A1215 1219        &]H  [JIJ5     , ,;
166       LPR HD1,7                                 P2-4 1184A0373 0379        &]H  02 2F     ,I,5
167       LD1 HD2,10                                P2-4 1189A0380 0389        &]H  ]2.2D     ,:,B
168       SD1 BAL+4,10                              P2-4 1194A0240 0249        &]H  ;.\1      ,8,#
169       XF   PR1                                  P2-4 1199A PR1             &]H  8A)))     ,H,A
171       LPR HD3,87                                P2-4 1204A0390 0476        &]H  02[8:J    ,6,&
172       XF   PR2                                  P2-4 1210A PR2             &]H  8D)))     J J;
174   TAG J   $                                     P2-4 1215A1215 1219        &]H  2JIJ5     JIJ5
175   OFL XF   SK7           START NEXT PAGE        P2-4 1220A SK7             &]H  8E)))     J:JB
176       LA1 ID1,1                                 P2-4 1225A0098 0098        &]H   4;4;     J8J#
```

```
177       SA1 CNT+2,3                               P2-4 1230A0087 0089        &]H  40A0?     JHJA
178       JR  OF2                                   P2-4 1235A1179 1183        &]H  D, ,;N    J6J&
179       LA1 ID0,1                                 P2-4 1241A0099 0099        &]H   4I4I     N N;
180       CN1 IND,1                                 P2-4 1246A0100 0100        &]H  :4F4F     NIN5
181       JE  BK1                                   P2-4 1251A0982 0986        &]H  8'H'A     N:NB
182       SA1 IND,1                                 P2-4 1256A0100 0100        &]H  44F4F     N8N#
183       J   BK2                                   P2-4 1261A1148 1152        &]H  2W W;     NHNA
184   LST XFC  E<))          SKIP 7, HALT           P2-4 1266A E<))            &]H  8E<))%    N6N&
185       END STI                                   P2-4 1272A0528 0532        &]H  2[ [;     =7=[
```

TRIAL BALANCE SAMPLE REPORT

ILLUSTRATION 3-1
REFER TO CHAPTER 3-I-C

TRIAL BALANCE

P #	ACCT #	DEBIT	CREDIT	CUMULATIVE BALANCE
1	1000	$	$ 12,645.07	$ 12,645.07CR
1	2100	$ 12,445.07	$ 31.88	$ 12,413.19
1	2221	$ 200.00	$ 8.00-	$ 208.00
1	3012	$	$ 12,645.07	$ 12,645.07CR
1	4501	$ 23.88	$	$ 23.88
1	4803	$ 12,645.07	$ 23.88	$ 12,621.19
1	7199	$ 3.12-	$	$ 3.12CR
1	7953	$ 27.00	$	$ 27.00
		$ 25,337.90 *	$ 25,337.90 *	$ *
1	2100	$	$ 989.98	$ 989.98CR
1	2221	$	$ 251.30	$ 251.30CR
1	4501	$ 395.45	$	$ 395.45
1	4601	$ 859.18	$ 144.15	$ 715.03
1	4802	$ 498.09	$ 367.29	$ 130.80
1	4803	$	$ 1,241.28	$ 1,241.28CR
1	7199	$ 845.83	$	$ 845.83
1	7952	$ 31.94	$	$ 31.94
1	7953	$ 363.51	$	$ 363.51
		$ 2,994.00 *	$ 2,994.00 *	$ *
2	1000	$	$ 2,450.94	$ 2,450.94CR
2	3012	$	$ 2,450.94	$ 2,450.94CR
2	4801	$ 2,450.94	$	$ 2,450.94
2	7199	$ 2,450.94	$	$ 2,450.94
		$ 4,901.88 *	$ 4,901.88 *	$ *
2	4501	$ 8,300.00	$	$ 8,300.00
2	4801	$	$ 8,300.00	$ 8,300.00CR
		$ 8,300.00 *	$ 8,300.00 *	$ *
3	1000	$	$ 724.25-	$ 724.25
3	1020	$ 497.83-	$ 58.35	$ 556.18CR
3	1152	$ 11.33-	$	$ 11.33CR
3	1401	$ 160.00-	$	$ 160.00CR
3	2400	$ 18.37-	$ 1.03	$ 19.40CR
3	2520	$ 19.40	$	$ 19.40
3	3013	$	$ 467.53-	$ 467.53
3	3018	$	$ 10.30-	$ 10.30
3	4501	$ 467.53-	$	$ 467.53CR
3	4702	$	$ 467.53-	$ 467.53
3	6051	$	$ 18.37	$ 18.37CR
3	6799	$ 456.20-	$	$ 456.20CR
		$ 1,591.86-*	$ 1,591.86-*	$ *
3	1000	$	$ 12,511.77	$ 12,511.77CR
3	1020	$ 12,499.17	$ 12.60-	$ 12,511.77
3	3012	$	$ 570.44	$ 570.44CR
3	7199	$ 570.44	$	$ 570.44
		$ 13,069.61 *	$ 13,069.61 *	$ *
3	1020	$ 67,286.60	$ 67,286.60	$

P #	ACCT #	DEBIT	CREDIT	CUMULATIVE BALANCE
		$ 67,286.60 *	$ 67,286.60 *	$ *
3	1102	$ 3,418.00	$	$ 3,418.00
3	1152	$ 95.00	$	$ 95.00
3	2520	$ 18.37-	$	$ 18.37CR
3	4731	$	$ 3,444.00	$ 3,444.00CR
3	4732	$ 3,444.00	$ 3,444.00	$
3	4733	$ 3,444.00	$	$ 3,444.00
3	6051	$	$ 3,425.63	$ 3,425.63CR
3	6799	$	$ 69.00	$ 69.00CR
		$ 10,382.63 *	$ 10,382.63 *	$ *
5	2221	$	$ 104.30	$ 104.30CR
5	4501	$ 104.30	$	$ 104.30
5	4803	$	$ 104.30	$ 104.30CR
5	7953	$ 104.30	$	$ 104.30
		$ 208.60 *	$ 208.60 *	$ *
8	1000	$	$ 174.84	$ 174.84CR
8	2221	$ 174.84	$ 12.84	$ 162.00
8	3012	$	$ 174.84	$ 174.84CR
8	4501	$ 12.84	$	$ 12.84
8	4803	$ 174.84	$ 12.84	$ 162.00
8	7953	$ 12.84	$	$ 12.84
		$ 375.36 *	$ 375.36 *	$ *

SARGE 2
2nd PASS OF REPORT GENERATOR

ILLUSTRATION 4-1
REFER TO CHAPTER 3-II-B

```
50000       BEG                                                     P2-4
50010       CRD                                                     P2-4
50020       -    1,1                                                P2-4
50030       -    6,2                                                P2-4
50040       -    55,4                                               P2-4
50050       -    59,10                                              P2-4
50060       -    70,10                                              P2-4
50070       PRT                                                     P2-4
50080       -    1,132                                              P2-4
50090       -    1,1                                                P2-4
50100       -    4,5                                                P2-4
50110       -    12,1                                               P2-4
50120       -    13,14                                              P2-4
50130       -    27,1                                               P2-4
50140       -    28,1                                               P2-4
50150       -    29,14                                              P2-4
50160       -    43,1                                               P2-4
50170       -    42,13                                              P2-4
50180       -    76,1                                               P2-4
50190       -    77,14                                              P2-4
50200       -    90,2                                               P2-4
50210       -    92,1                                               P2-4
50220       PCH                                                     P2-4
50230       -    1,80                                               P2-4
50240       -    2,4                                                P2-4
50250       -    6,1                                                P2-4
50260       -    6,2                                                P2-4
50270       -    55,5                                               P2-4
50280       -    59,10                                              P2-4
            *     029        E10                                                P2-4
50290       ORG 0373                                                P2-4
50300 HD0 +13 TRIAL BALANCE                                         P2-4
50310 HD1 +7  P  ACCT                                               P2-4
50320 HD2 +11 CUMULATIVE                                            P2-4
50330 HD3 +25 #     #                DEBIT                          P2-4
50340 HD4 +7  CREDIT                                                P2-4
50350 HD5 +10 BALANCE                                               P2-4

50360 AC1 +10                                                       P2-4
50370 AC2 +10                                                       P2-4
50380 AC3 +10                                                       P2-4
50390 AC4 +10                                                       P2-4
50400 AC6 +10                                                       P2-4
50410 PV  +2                 COLS 6 & 7                             P2-4
50420 HLD +4                 ACCT NO                                P2-4
50430 ID0 +1  0                                                     P2-4
50440 ID1 +1  1                                                     P2-4
50450 ID2 +1  2                                                     P2-4
50460 SPA +10                                                       P2-4
50470 SPN +10 0000000000!
50480 ST0 +1                                                        P2-4
50490 ST1 +1                                                        P2-4
50500 DAT +4  0865                                                  P2-4
50510 DLR +1  $                                                     P2-4
50520 CRT +2  CR                                                    P2-4
50530 IND +1                 PRID ACCT BAL IND                      P2-4
50540 ASK +1  *                                                     P2-4
50550 CNT +2                                                        P2-4
50560 KON +2  51                                                    P2-4
50570     STA                                                       P2-4
          *    056    STT RES PRT                                           P2-4
50580 STT CLR 0161 0292                                             P2-4
          *    057       MOV HD0,PR0          TITLE                         P2-4
50590     LA2 HD0,13                                                P2-4
50600     SA2 0202 0214                                             P2-4
          *    058       PRN SP2                                            P2-4
50610     XF   PR2                                                  P2-4
          *    059       RES PCH                                            P2-4
50620     CLR 0293 0372                                             P2-4
          *    060       ORT OF2              TO COL HDGS SUBRT             P2-4
50630     JR  OF2                                                   P2-4
          *    061       REA                  READ FIRST CARD               P2-4
50640     XF   REA                                                  P2-4
          *    062    FST MOV FD3,PV          STORE COLS 6 & 7              P2-4
50650 FST LA2 0006 0007                                             P2-4
```

ILLUSTRATION 4-2

```
50660       SA2 PV,2                                             P2-4
            *    063    R12 SEN FD4,HLD,ST1      STORE ACCT NO               P2-4
50670 RT2  LPR 0055 0058                                         P2-4
50680       SPR HLD,4                                            P2-4
50690       SPR ST1,1                                            P2-4
            *    064        IFD ID1,L,FD1,ON     COMP COL 1 TO ONE           P2-4
50700       LA2 ID1,1                                            P2-4
50710       CN2 0001 0001                                        P2-4
50720       JL   ON                                              P2-4
            *    065        SEN FD5,STO          STORE MSL OF AMT            P2-4
50730       LPR 0059 0068                                        P2-4
50740       SPR STO,1                                            P2-4
            *    066        JMP M01                                          P2-4
50750       J    M01                                             P2-4
            *    067    ON  SEN FD6,STO          STORE MSL OF AMT            P2-4
50760 ON   LPR 0070 0079                                         P2-4
50770       SPR STO,1                                            P2-4
            *    068    M01 IFD ST1,G,ID0,M02    CHECK CR-DEB ACCT           P2-4
50780 M01  LA2 ST1,1                                             P2-4
50790       CN2 ID0,1                                            P2-4
50800       JG   M02                                             P2-4
            *    069        IFD ID0,G,STO,NXT    CHECK CR-DEB AMT            P2-4
50810       LA2 ID0,1                                            P2-4
50820       CN2 STO,1                                            P2-4
50830       JG   NXT                                             P2-4
            *    070        IFD ID1,L,FD1,ON1                                P2-4
50840       LA2 ID1,1                                            P2-4
50850       CN2 0001 0001                                        P2-4
50860       JL   ON1                                             P2-4
            *    071        ADD FD5,AC3,AC4      DEB AMT, CR ACCUM           P2-4
50870       LA2 0059 0068                                        P2-4
50880       AM2 AC3,10                                           P2-4
50890       AM2 AC4,10                                           P2-4
            *    072        JMP M03                                          P2-4
50900       J    M03                                             P2-4
            *    073    ON1 ADD FD6,AC3,AC4                                  P2-4
50910 ON1  LA2 0070 0079                                         P2-4



50920       AM2 AC3,10                                           P2-4
50930       AM2 AC4,10                                           P2-4
            *    074        JMP M03                                          P2-4
50940       J    M03                                             P2-4
            *    075    NXT IFD ID1,L,FD1,ONA                                P2-4
50950 NXT  LA2 ID1,1                                             P2-4
50960       CN2 0001 0001                                        P2-4
50970       JL   ONA                                             P2-4
            *    076        SUB FD5,AC3,AC4      CR AMT, CR ACCUM            P2-4
50980       LA2 0059 0068                                        P2-4
50990       SM2 AC3,10                                           P2-4
51000       SM2 AC4,10                                           P2-4
            *    077        JMP M03                                          P2-4
51010       J    M03                                             P2-4
            *    077    ONA SUB FD6,AC3,AC4                                  P2-4
51020 ONA  LA2 0070 0079                                         P2-4
51030       SM2 AC3,10                                           P2-4
51040       SM2 AC4,10                                           P2-4
            *    079        JMP M03                                          P2-4
51050       J    M03                                             P2-4
            *    080    M02 IFD ID0,G,STO,ALT    DEB,CK CR-DEB AMT           P2-4
51060 M02  LA2 ID0,1                                             P2-4
51070       CN2 STO,1                                            P2-4
51080       JG   ALT                                             P2-4
            *    081        IFD ID1,L,FD1,ONN                                P2-4
51090       LA2 ID1,1                                            P2-4
51100       CN2 0001 0001                                        P2-4
51110       JL   ONN                                             P2-4
            *    082        ADD FD5,AC1,AC2      DEB AMT,DEB ACCUM           P2-4
51120       LA2 0059 0068                                        P2-4
51130       AM2 AC1,10                                           P2-4
51140       AM2 AC2,10                                           P2-4
            *    083        JMP M03                                          P2-4
51150       J    M03                                             P2-4
            *    084    ONN ADD FD6,AC1,AC2                                  P2-4
51160 ONN  LA2 0070 0079                                         P2-4
51170       AM2 AC1,10                                           P2-4
```

ILLUSTRATION 4-3

```
51180      AM2 AC2,10                                        P2-4
           *    085        JMP M03                                      P2-4
51190      J    M03                                          P2-4
           *    086   ALT  IFD ID1,L,FD1,ONV    CR AMT, DEB ACCUM       P2-4
51200 ALT  LA2  ID1,1                                        P2-4
51210      CN2  0001 0001                                    P2-4
51220      JL   ONV                                          P2-4
           *    087        SUB FD5,AC1,AC2                              P2-4
51230      LA2  0059 0068                                    P2-4
51240      SM2  AC1,10                                       P2-4
51250      SM2  AC2,10                                       P2-4
           *    088        JMP M03                                      P2-4
51260      J    M03                                          P2-4
           *    089   ONV  SUB FD6,AC1,AC2                              P2-4
51270 ONV  LA2  0070 0079                                    P2-4
51280      SM2  AC1,10                                       P2-4
51290      SM2  AC2,10                                       P2-4
           *    090   M03  REA                  READ NEXT CARD          P2-4
51300 M03  XF   REA                                          P2-4
           *    091        IFD FD4,E,HLD,RT2    COMP NEW ACCT NO        P2-4
51310      LA2  0055 0058                                    P2-4
51320      CN2  HLD,4                                        P2-4
51330      JE   RT2                                          P2-4
           *    092   CON  SEN PV,PNO          IF BK,INFO TO PPR        P2-4
51340 CON  LPR  PV,2                                         P2-4
51350      SPR  0298 0299                                    P2-4
           *    09250      SEN PN1,PNO                                  P2-4
51360      LPR  0298 0298                                    P2-4
51370      SPR  0161 0161                                    P2-4
           *    093        MOV DAT,DTE                                  P2-4
51380      LA2  DAT,4                                        P2-4
51390      SA2  0294 0297                                    P2-4
           *    094        MOV HLD,ZS,ACT,ACN                           P2-4
51400      LWS  HLD,4                                        P2-4
51410      SZS  0164 0168                                    P2-4
51420      SZS  0347 0351                                    P2-4
           *    095        IFD AC1,E,SPA,ZZ                             P2-4


51430      LA2  AC1,10                                       P2-4
51440      CN2  SPA,10                                       P2-4
51450      JE   ZZ                                           P2-4
           *    096        MOV AC1,ED,DB2                               P2-4
51460      LWS  AC1,10                                       P2-4
51470      SED  0173 0186                                    P2-4
           *    097   ZZ   IFD AC3,E,SPA,WW                             P2-4
51480 ZZ   LA2  AC3,10                                       P2-4
51490      CN2  SPA,10                                       P2-4
51500      JE   WW                                           P2-4
           *    098        MOV AC3,ED,CR2                               P2-4
51510      LWS  AC3,10                                       P2-4
51520      SED  0189 0202                                    P2-4
           *    099   WW   MOV AC1,AC6,AMT      DB-CR=BAL ACCT NO       P2-4
51530 WW   LA2  AC1,10                                       P2-4
51540      SA2  AC6,10                                       P2-4
51550      SA2  0351 0360                                    P2-4
           *    100        SUB  AC3,AC6,AMT                             P2-4
51560      LA2  AC3,10                                       P2-4
51570      SM2  AC6,10                                       P2-4
51580      SM2  0351 0360                                    P2-4
           *    101        DRT M04                                      P2-4
51590      JR   M04                                          P2-4
           *    102        JMP M05                                      P2-4
51600      J    M05                                          P2-4
           *    103   M04  ERT RTN              EDIT $, CR              P2-4
51610 M04  JX   RTN                                          P2-4
           *    104        MOV DLR,DB1,CR1,BL1                          P2-4
51620      LA2  DLR,1                                        P2-4
51630      SA2  0172 0172                                    P2-4
51640      SA2  0188 0188                                    P2-4
51650      SA2  0236 0236                                    P2-4
           *    105        IFD AC6,E,SPA,RTN                            P2-4
51660      LA2  AC6,10                                       P2-4
51670      CN2  SPA,10                                       P2-4
51680      JE   RTN                                          P2-4
           *    10550      IFD AC6,E,SPN,RTN
```

ILLUSTRATION 4–4

```
51690      LA2 AC6,10
51700      CN2 SPN,10
51710      JE  RTN
           *    106        MOV AC6,ED,BL2                                      P2-4
51720      LWS AC6,10                                         P2-4
51730      SED 0237 0250                                      P2-4
           *    107        IFD AC6,L,SPA,RT,RTN                                P2-4
51740      LA2 AC6,10                                         P2-4
51750      CN2 SPA,10                                         P2-4
51760      JL  RT                                             P2-4
51770      J   RTN                                            P2-4
           *    108    RT  MOV CRT,BL3                                         P2-4
51780 RT   LA2 CRT,2                                          P2-4
51790      SA2 0250 0251                                      P2-4
           *    109    RTN XRT M04                                             P2-4
51800 RTN  J   M04                                            P2-4
           *    110    M05 PRN SP1           PRINT ACCT TOTALS                 P2-4
51810 M05  XF  PR1                                            P2-4
           *    111        PUN               PUNCH                             P2-4
51820      XF  PUN                                            P2-4
51830      PTE                                                P2-4
           *    112        RES IND                                             P2-4
51840      CLR IND,1                                          P2-4
           *    113        ADD ID1,CNT                                         P2-4
51850      LA2 ID1,1                                          P2-4
51860      AM2 CNT,2                                          P2-4
           *    114        IFD CNT,G,KON,OFL                                   P2-4
51870      LA2 CNT,2                                          P2-4
51880      CN2 KON,2                                          P2-4
51890      JG  OFL                                            P2-4
           *    115    BK1 RES AC1,AC3       CLEAR ACCT ACCUMS                 P2-4
51900 BK1  CLR AC1,10                                         P2-4
51910      CLR AC3,10                                         P2-4
           *    116        IFD PV,E,FD3,RT2  COMPARE COL 6 & 7                 P2-4
51920      LA2 PV,2                                           P2-4
51930      CN2 0006 0007                                      P2-4
51940      JE  RT2                                            P2-4
```

```
           *    117        MOV AC2,AC6       IF BK,INFO TO PRN                 P2-4
51950      LA2 AC2,10                                         P2-4
51960      SA2 AC6,10                                         P2-4
           *    118        SUB AC4,AC6                                         P2-4
51970      LA2 AC4,10                                         P2-4
51980      SM2 AC6,10                                         P2-4
           *    119        DRT M04                                             P2-4
51990      JR  M04                                            P2-4
           *    120        IFD AC2,E,SPA,YY                                    P2-4
52000      LA2 AC2,10                                         P2-4
52010      CN2 SPA,10                                         P2-4
52020      JE  YY                                             P2-4
           *    121        MOV AC2,ED,DB2                                      P2-4
52030      LWS AC2,10                                         P2-4
52040      SED 0173 0186                                      P2-4
           *    122    YY  IFD AC4,E,SPA,RT4                                   P2-4
52050 YY   LA2 AC4,10                                         P2-4
52060      CN2 SPA,10                                         P2-4
52070      JE  RT4                                            P2-4
           *    123        MOV AC4,ED,CR2                                      P2-4
52080      LWS AC4,10                                         P2-4
52090      SED 0189 0202                                      P2-4
           *    124    RT4 MOV ASK,DB3,CR3,BL4  EDIT *                         P2-4
52100 RT4  LA2 ASK,1                                          P2-4
52110      SA2 0187 0187                                      P2-4
52120      SA2 0203 0203                                      P2-4
52130      SA2 0252 0252                                      P2-4
           *    125        PRN SP2           PRINT SECTION TOT                 P2-4
52140      XF  PR2                                            P2-4
           *    126        MOV ID1,IND                                         P2-4
52150      LA2 ID1,1                                          P2-4
52160      SA2 IND,1                                          P2-4
           *    127        ADD ID2,CNT                                         P2-4
52170      LA2 ID2,1                                          P2-4
52180      AM2 CNT,2                                          P2-4
           *    128        IFD CNT,G,KON,OFL                                   P2-4
52190      LA2 CNT,2                                          P2-4
```

```
52200         CN2 KON,2                                        P2-4
52210         JG  OFL                                          P2-4
              *    129    BK2 RES AC2,AC4      CLEAR ACCUMS                P2-4
52220 BK2 CLR AC2,10                                          P2-4
52230         CLR AC4,10                                       P2-4
              *    130        IFD FD4,E,SPA,LST    COMP FOR LAST CRD           P2-4
52240         LA2 0055 0058                                    P2-4
52250         CN2 SPA,10                                       P2-4
52260         JE  LST                                          P2-4
              *    131        JMP FST                                      P2-4
52270         J   FST                                          P2-4
              *    132    OF2 EXT TAG          PRN COL HEADINGS            P2-4
52280 OF2 JX  TAG                                              P2-4
              *    133        SEN HD1,PRT                                  P2-4
52290         LPR HD1,7                                        P2-4
52300         SPR 0161 0292                                    P2-4
              *    134        MOV HD2,BL2                                  P2-4
52310         LA2 HD2,11                                       P2-4
52320         SA2 0237 0250                                    P2-4
              *    135        PRN SP1                                      P2-4
52330         XF  PR1                                          P2-4
              *    136        SEN HD3,PRT                                  P2-4
52340         LPR HD3,25                                       P2-4
52350         SPR 0161 0292                                    P2-4
              *    137        MOV HD4,CR2                                  P2-4
52360         LA2 HD4,7                                        P2-4
52370         SA2 0189 0202                                    P2-4
              *    138        MOV HD5,BL2                                  P2-4
52380         LA2 HD5,10                                       P2-4
52390         SA2 0237 0250                                    P2-4
              *    139        PRN SP2                                      P2-4
52400         XF  PR2                                          P2-4
              *    140    TAG XRT OF2                                      P2-4
52410 TAG J   OF2                                              P2-4
              *    141    OFL RES PRT,CNT      START NEXT PAGE             P2-4
52420 OFL CLR 0161 0292                                        P2-4
52430         CLR CNT,2                                        P2-4
```

```
              *    142        PRN SK7                                      P2-4
52440         XF  PR7                                          P2-4
              *    143        DRT OF2                                      P2-4
52450         JR  OF2                                          P2-4
              *    144        IFD IND,E,SPA,BK1                            P2-4
52460         LA2 IND,1                                        P2-4
52470         CN2 SPA,10                                       P2-4
52480         JE  BK1                                          P2-4
              *    145        RES IND                                      P2-4
52490         CLR IND,1                                        P2-4
              *    146        JMP BK2                                      P2-4
52500         J   BK2                                          P2-4
              *    147    LST RES PRT,PCH                                  P2-4
52510 LST CLR 0161 0292                                        P2-4
52520         CLR 0293 0372                                    P2-4
              *    148        PRN SK7                                      P2-4
52530         XF  PR7                                          P2-4
              *    149        PUN                                          P2-4
52540         XF  PUN                                          P2-4
52550         PTE                                              P2-4
              *    150    STP HLT STP                                      P2-4
52560 STP XF  HLT                                              P2-4
52570         J   STP                                          P2-4
52580         END STT                                          P2-4
```

MEMORY DUMP

ILLUSTRATION 5-1
REFER TO CHAPTER 3-III-B

```
                                                       TRIAL BALANCE
                    U86552  1$*     03-1
    1OUCK                           04-1
        '+0962!52\1 ;    %;$ RJH= 05-1
7=L35                               06-1
                                    07-1
                                    08-1
                                    09-1
              U51            2C  L 10-1
                                 & 11-1
 6        =7=[P2-4                  12-1
P  ACCTCUMULATIVE#     #            13-1
     DEBIT            CREDIT        14-1
                                    15-1
    BALANCE                         16-1
                                    17-1
&D)))157-&D,',W&)Δ)) J 1 F;4044< 18-1
JJGJ?;0E4J 4;4;:     7#:#6'J3OI# 19-1
JJ5J52#H#A'0.0UJU.U.;D&D& 4I4IH 20-1
:0E0E7CHCA:D&D&BCIC5>D D5>D:D#C 21-1
2\6\EYU D5YU:D#2\6\E:D&D&B\8\#\ 22-1
>8-8H>8C8&2\6\EY8-8HY8C8&&)Δ))G 23-1
 JGJ?:UE4J8< <;*4044E    @1I1IA 24-1
W5<5#JU<UC;5B5L3UE4J4I1I-4-I-16 25-1
1)%)#38-6H%)')N8? ?;?8-8HR1[18? 26-1
1)%)#3U U5%)')N83 3;?D D5RF JF83 27-1
 8-8HCD U54DHDE4-5-<D9 9;2&6&E9 28-1
L8H&A U3U34IUIU4F JFJ4.<.<1)%)#E 29-1
3DHDE%)')N8&H&A?DHDEK.#1 JLDHDF& 30-1
NDHDE8&H&A 4.4141J10Z&H&A&Δ)))' 31-1
```

CHAPTER 4

UNIVAC 1005 SOFTWARE OPERATING PROCEDURES

I. ALTERNATE SWITCHES OPERATING PROCEDURES

1. Loading program into Core Memory.

   Alt. Switch 1 on/illuminated.
   Alt. Switch 2 off/extinguished.

2. Normal running.

   Alt. Switch 1 off/extinguished.
   Alt. Switch 2 on/illuminated (if automatic forms overflow desired).

3. Testing programs (debugging).

   Alt. Switch 1 on/illuminated.
   Alt. Switch 2 on/illuminated.

During testing the programmer is able to step instruction by instruction through a program.

4. Note: ALT Switch 4 on/illuminated suppresses punching

II. SOFTWARE OPERATING PROCEDURES

Single Address Assembly Language (SAAL)

A. SAAL 1 - this is the first pass of the assembly program (S41).

(1) Operating Instructions:

(a) Reader - load cards into input hopper (SAAL 1 object program, followed by source program, followed by one blank card).

(b) Console

1. Depress START and CLEAR BUTTON.
2. Alternate Switch 1 on/illuminated, all others off/extinguished.
3. Depress FEED BUTTON.
4. Depress RUN BUTTON.

When processor HALTS, SAAL 1 is loaded.

5. Depress Alternate Switch 1 off/extinguished.
6. Depress Alternate Switch 2 on/illuminated (if automatic forms overflow is desired).
7. Depress START and CLEAR BUTTONS.

8. Depress FEED BUTTON.
9. Depress RUN BUTTON.

(2) Output

(a) PUNCH - no punched output in SAAL 1.

(b) PRINTOUT - listing of the label table relating each symbolic reference (label) in the symbolic program (source program) with its appropriate position in Core Memory.

(3) Errors

(a) ERR NO BEG CRD is printed, paper is advanced to the next page and the program halts - Indicates the BEG card does not precede the source program.

(b) ERR OP IN DATA DIV is printed to the right of the card in error, paper is advanced to the next page and the program halts. This type of error indicates an illegal code in the operation field (Cols. 11-13). No recovery is possible. The last card in the output stacker is the card in error. Correct card and restart.

(c) DUP printed under ERROR heading - Indicates a duplicate label.

(d) >148 printed under ERROR heading - Indicates the maximum number of labels has been exceeded (148 labels).

(e) OVM printed under ERROR heading - Indicates the maximum memory has been exceeded (3844 positions).

B. SAAL 2 - second pass of the Assembler - (S42)

(1) Operating Instructions:

(a) Reader - load cards into input hopper (SAAL 2 object program followed by source program, followed by one blank card).

(b) Punch - clear punch and fill hopper with blank cards.

(c) Console

1. Depress Alternate Switch 1 on/illuminated - all other switches off.
2. Depress START and CLEAR BUTTONS.
3. Depress FEED BUTTON.
4. Depress RUN BUTTON.

When processor HALTS, SAAL 2 is loaded.

5. Depress Alternate Switch 1 off/extinguished.
6. Depress Alternate Switch 2 on/illuminated (if automatic forms overflow is desired).
7. Depress START and CLEAR BUTTONS.
8. Depress FEED BUTTON.
9. Depress RUN BUTTON.

(2) Output

(a) Punch - a card for card output with the pseudo-machine code punched in the cards.

(b) Printout - a listing of each card equating each symbolic line of coding in the source program with the generated machine code.

(3) Errors

(a) Program halts after first card is read - Indicates BEG card does not precede source program.

(b) ' O ' printed under 1st position of ERROR heading - Indicates an illegal operation code.

(c) ' E ' printed under 2nd position of ERROR heading - Indicates an expression error, i.e. operand which is less than 0001 or greater than 3875. The most frequent cause of error is an undefined label. This type of error will print 6530 under the OPERAND heading.

(d) ' P ' printed under 3rd position of ERROR heading - Indicates a precautionary warning, i.e. an instruction greater than 10 or 21 characters utilizing AR1 or AR2 respectively.

(e) ' S ' printed under the 4th position of ERROR heading - Indicates a sequence number error.

C. Condense Program (CD4)

(1) Operating Instructions

(a) Reader - load cards into input hopper (condense object program followed by output of SAAL 2, followed by one blank card).

(b) Punch - clear punch unit and fill hopper with blank cards.

(c) Console

1. Depress Alternate Switch 1 on/illuminated.
2. Depress START and CLEAR BUTTONS.
3. Depress FEED BUTTON.
4. Depress RUN BUTTON.

When processor HALTS, condense is loaded.

5. Depress Alternate Switch 1 off/extinguished.
6. Depress START and CLEAR BUTTONS.
7. Depress FEED BUTTON.
8. Depress RUN BUTTON.

D. Memory Dump (DMP)

(1) Operating Instructions:

(a) Reader - load input hopper with memory dump object program.

(b) Punch - no punch output.

(c) Console

1. Depress Alternate Switch 1 on/illuminated.
2. Depress START and CLEAR BUTTONS.
3. Depress FEED BUTTON.
4. Depress RUN BUTTON.

When processor HALTS

5. Depress Alternate Switch 1 off/extinguished.
6. Depress START and CLEAR BUTTONS.
7. Depress FEED BUTTON.
8. Depress RUN BUTTON.

E. READ - PRINT - PUNCH (RPX)

(1) Operating Instructions:

(a) Reader - load input hopper with RPX object program, followed by data cards, followed by one blank card.

(b) Punch - clear punch unit and fill hopper with blank cards.

(c) Console

1. Depress Alternate Switch 1 on/illuminated.
2. Depress START and CLEAR BUTTONS.
3. Depress FEED BUTTON.
4. Depress RUN BUTTON.

When processor HALTS

5. Depress Alternate Switch 1 off/extinguished.
6. Depress Alternate Switch 2 on/illuminated (if automatic forms overflow is desired).
7. Depress START and CLEAR BUTTONS.
8. Depress FEED BUTTON.
9. Depress RUN BUTTON.

F. NUMBER IT (NIT)

(1) Operating Instructions:

(a) Reader - load cards into input hopper (NITA followed by data cards, followed by one blank card).

(b) Punch - clear punch unit and fill input hopper with blank cards.

(c) Console

1. Depress Alternate Switch 1 on/illuminated.
2. Depress START and CLEAR BUTTONS.
3. Depress FEED BUTTON.
4. Depress RUN BUTTON.

When processor HALTS, Number it is loaded.

5. Depress Alternate Switch 1 off/extinguished.
6. Depress START and CLEAR BUTTONS.
7. Depress FEED BUTTON.
8. Depress RUN BUTTON.

(2) Output

(a) Punch - a card for card punched deck with all cards sequence punched in columns 1-3 starting with ØØ1, and new program ID inserted in columns 62-65 if header was used.

(b) Printer - an 80/80 listing of each card punched.

G. DUPLICATE (DUP)

(1) Operating Instructions:

(a) Reader - load cards into input hopper (DUPA followed by four header cards, followed by data cards, followed by a sentinal and a blank card.

(b) Punch - clear punch unit and fill input hopper with blank cards.

(c) Processor

1. Depress Alternate Switch 1 on/illuminated.
2. Depress START and CLEAR BUTTONS.
3. Depress FEED BUTTON.
4. Depress RUN BUTTON.

When processor HALTS

5. Depress Alternate Switch 1 off/extinguished.
6. Depress START and CLEAR BUTTONS.
7. Depress FEED BUTTON.
8. Depress RUN BUTTON.

H. CLEAR (CLR)

(1) Operating Instructions:

Clear cards are normally placed before object cards for the purpose of clearing memory prior to loading a new program.

CHAPTER 5

UNIVAC 1005 HARDWARE MACHINE TESTING and OPERATING PROCEDURES

I. MANUAL ALTERNATE SWITCHES.

A. Mode of Operation Table.

The following table shows the mode for the sixteen possible switch combinations:

MODE	Punch Inhibited[1]	JS3 Instruction[2]	SWITCH ONE	SWITCH TWO	SWITCH THREE	SWITCH FOUR
Normal Operation	No	NI[3]	OFF	OFF	OFF	OFF
"	Yes	NI	OFF	OFF	OFF	ON
"	No	JUMP	OFF	OFF	ON	OFF
"	Yes	JUMP	OFF	OFF	ON	ON
Normal Auto Form Overflow	No	NI	OFF	ON	OFF	OFF
"	Yes	NI	OFF	ON	OFF	ON
"	No	JUMP	OFF	ON	ON	OFF
"	Yes	JUMP	OFF	ON	ON	ON
LOAD	No	NI	ON	OFF	OFF	OFF
TRACE	Yes	NI	ON	OFF	OFF	ON
RESERVED	No	JUMP	ON	OFF	ON	OFF
TRACE	Yes	JUMP	ON	OFF	ON	ON
Single Instruction	No	NI	ON	ON	OFF	OFF
" " W TRACE	Yes	NI	ON	ON	OFF	ON
" "	No	JUMP	ON	ON	ON	OFF
" " W TRACE	Yes	JUMP	ON	ON	ON	ON

Notes: 1. When switch four is "on", punch and PTE orders will be ignored.

2. Switch three sets an indicator that is program testable by the JS3 instruction. If alternate switch 3 is "on", control will be transferred "M"; if "off", the next instruction in sequence will be executed

3. NI means Next Instruction.

B. Automatic Form Overflow Mode. Normal auto form overflow does the following during XF print orders:

1. If a "1" punch only on the printer form loop is detected during a prior print, the form will be advanced to the next line of the form loop on which there are 1, 2 and 4 punches on the next print instruction.

2. If a form overflow occurs the compare indicator is set to a less than condition.

3. If no form overflow occurs the compare indicator is set to a greater than condition.

4. All card or paper tape XF's affect the comparator. If there is no print on the XF the comparator will be set to greater.

C. Trace Mode. This prints the static registers between the update of the program address counter and the execution of an instruction. It destroys print storage.

The following table shows the registers traced and their print positions:

Description	Print Position
Z Register	81-90
Instructions Register	91-95
Blank	96-96
Program Address Counter PAK (address of next instruction in memory)	97-98
Machine Constants	99-107
X Register	108-109
Machine Constants	110-111

D. Single Instructions Mode. This permits the programmer to cycle through his program. During this mode, the processor Halts at the end of the first internal cycle of each instruction executed. In single instruction mode trace may or may not be used depending on the setting of Manual Alternate Switch 4 (on for trace).

Each 1005 instruction consists of 5 "6 bit" characters. During single instruction mode, the entire instruction is readable from masks.

Mask 6 - Operation code (instruction Character 1)

Mask 8 - Operand (instruction Character 2-5)

Mask 9 - Operation register and operand bank designation.

When executing a conditional jump, the indication of the condition may be seen on Mask 9. If indicator light 1 is lit, the condition is not met and the next instruction in sequence will be executed. If indicator light 2 is lit, the condition is met and control will be transferred to the "M" address.

In single instruction mode, the following instructions show on Mask 6 as multiple instructions.

a) Conditional Jump Instructions - When the condition is met, an unconditional jump instruction cycle is generated.

b) Store Zero Suppress (SZS) and Store Edit (SED) - These instructions generate a SA2 (Store Ascending Register 2) instruction cycle.

1. Reading PAK

a) Set processor to single instruction mode to stop after the execution of the previous instruction.

b) Set the processor MODE switch to STEP.

c) Depress run button until Step 1 lights on Mask 5.

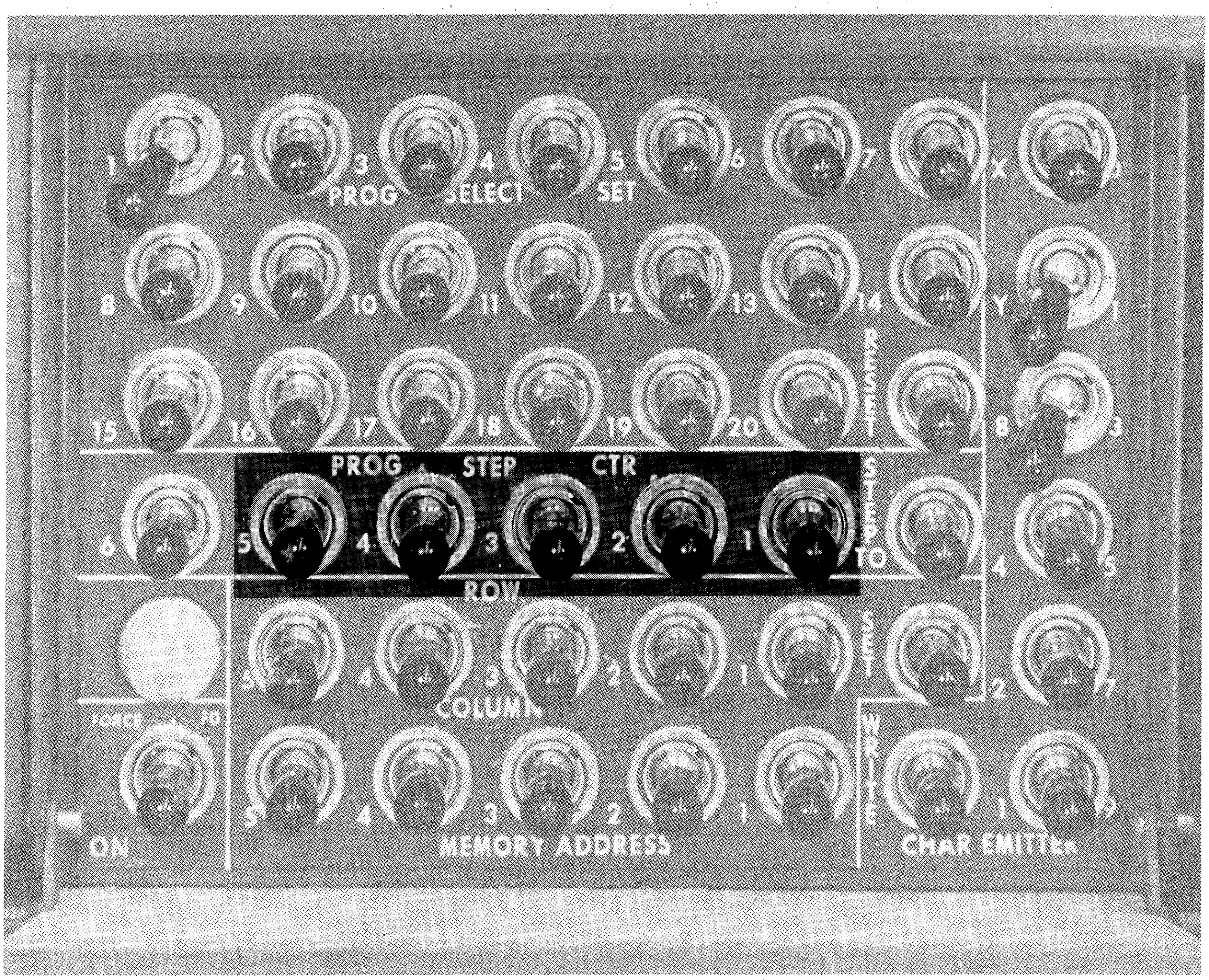
d) PAK is displayed:

1) Mask 8 indicators 11-15 (Row) 16-20 (Column).

2) Mask 9 indicators 20-21 (Bank Designation).

Reference description of masks for details.

II. TEST SWITCH PANEL.



The Test Switch Panel for the UNIVAC 1005 Card Processor is located on the upper front of the Processor just to the left of the Card Stacker. The Test Switch Panel occupies the lower half of this panel area.

The Test Switches are beneath a cover which is hinged at the bottom. Access to the switches is obtained by swinging this cover down. There are 47 toggle switches in the area; 6 rows of 8 switches each with one blank position.

A. Program Step Counter Switches

The following 5 switches, located near the center of the panel are used to stop the program on a given type of instruction.

SWITCHES 1 - 5 - These five switches are used to set up the instruction number desired according to the binary code printed on Display Panel 6. Each of these five switches is set in one of two positions according to whether the related code position calls for a 1 or 0:

Off (Up) for a 1
On (Down) for a 0

By keying instructions to switches and running the processor in a continuous mode, the machine will come to a halt after executing the first cycle of the keyed instruction. Using this procedure, the programmer may let his program run until it gets to a particular instruction and then step through that particular routine in single instruction mode.

The remaining switches are primarily used for engineering maintenance.

III. DISPLAY MASKS.

A. Display Mask 4.

1	2	3	4	5	6	7	8
HOPPER	FEED	RD JAM	TSP JAM	STACKR	FORM	ADV ✓	PUNCH
00000	00001	00011	00111	01110	11100	11001	10010

9	10	11	12	13	14	15	16
HALT	IND 1	IND 2	IND 3	IND 4	RD I/O	PR I/O	PCH I/O
00100	01000	10001	00010	00101	01010	10101	01011

17	18	19	20	21	22	23	24
SP 1	SP 2	SK 1	SK 2	SK 4	END RD	END PR	R/P EX
10111	01111	11110	11101	11011	10110	01101	11010

25	26	27	28	29	30	31	32
HOPPER	FEED	RD JAM	TSP JAM	WAIT JAM	FORM O'F	PCH HLD	PCH CLR
10100	01001	10011	00110	01100	11000	10000	11111

Indicators 1 - 13 are of interest during continuous operation to signify a reason for Processor stopping. Indicators 14-21, 24, & 30 - 31 are for program analysis with regard to Input/Output. Indicators 25 - 29 apply when an Auxiliary Card Reader is used.

Display Mask 4 should be displayed when the Processor is in Continuous operation.

IMPORTANT: -- If the Processor stops during a run, the operator must always consult Display Mask 4 to determine the reason for stopping before pressing any of the operating controls.

By noting the indication on this Display Mask, the proper action can be taken. The Processor operation can then be resumed properly.

Card Feeding (1 - 5)

All areas of the card feeding mechanism from the Magazine to the Stacker are covered by controls to stop the Processor in the event of mis-feeding.

HOPPER (1) - Input Magazine

This indicator will be lit whenever the Input Magazine is empty and the Feed indicator is lit. The Hopper indicator cannot be on alone.

During operation, this indicator will light after the last card is read.

The Processor will stop after the read order is executed with the last card in the Card Stacker.

Processor operation is resumed by:

Pressing the Stop switch.

Placing cards in the Magazine.

Pressing the Feed switch once to feed a card from the Magazine into the Wait Station; the Hopper and Feed indicators will turn off.

Pressing the Run switch once to resume the Processor operation.

FEED (2) - Wait Station

This indicator will be lit by pressing the Clear switch or by a card cycle if there is no card fed to the Wait Station.

Should this indicator light during operation, a card has failed to feed from the Magazine. If there are cards in the Magazine, the Processor will stop on the next read order with the Feed indicator lit and the Read not executed.

Processor operation is resumed by:

Pressing the Stop switch.

Removing the cards from the Magazine.

Examining the cards on the bottom of the stack to determine the reason for the failure to feed.

Correcting these cards and returning all cards to the Magazine.

Pressing the Feed switch once to feed a card from the Magazine to the Wait Station; the Feed indicator will turn off.

Pressing the Run switch once to resume the Processor operation.

The Hopper and Feed indicators will be lit when the last card has been fed from the Wait Station to the Card Stacker. The Processor will stop at the completion of the current Read. If additional cards are to be processed; press the Stop switch, place the cards in the Magazine, press the Feed and Run switches.

RD JAM (3) - Read Jam

Should the Processor stop during operation with this indicator lit, either one of the following has occurred:

1. A card from the Wait Station may have failed to feed to the Read Photoelectric Diodes.

2. The Read Photoelectric Diodes may have failed the "light-dark" test.

   Before reading the first card and between the reading of each following card, the photo-diodes are in a "light" condition. When the leading end of a card enters the photo-diode area, a "dark" condition occurs.

   This light-dark change must be executed properly to assure correct reading; if it is not, the Processor will stop.

If the stoppage is due to a card jam before the photo-diodes, the Read-Execute signal is retained in the Processor; the jammed card was not read. The following procedure will return the Processor to operation without loss of data:

1. Press the Stop switch.

2. Remove all cards from the Magazine and Wait Station.

3. Press the Feed switch once while the Magazine is empty. The Feed indicator will light.

4. Remake the damaged cards, if necessary, and replace them in their proper sequence at the bottom of the stack in the Magazine.

5. Press the Feed switch once to feed a card from the Magazine to the Wait Station.

6. Press the Run switch once to resume the Processor operation.

If there is no card jam when the Processor stops with the RD JAM indicator lit, a light-dark test failure is signified. In this case:

The Read-Execute signal is retained in the Processor; card reading did not take place, only card feeding.

The last card in the Stacker has not been read.

The following procedure should be followed to restore the Processor to operation in the event the light-dark test failure was only momentary:

1. Remove all cards from the Magazine. Remove the last card from the Stacker and the card from the Wait Station.

2. Follow steps 3 through 6 above. The card from the Stacker should be first in sequence when replacing the cards in the Magazine.

Should the RD JAM indicator light, try the procedure again. If the same indication persists, remake the card and try again. If failure continues, have the field engineer check the photodiode operation.

TSP JAM (4) - Transport Jam (Photo-Diodes to Stacker)

This indicator will light in the event of a jam as the card is delivered to the Stacker.

The Processor will stop.

To resume the Processor operation without loss of data:

Press the Stop switch.

Remove the mis-fed card or cards.

Press the Run switch.

STACKR (5) - Stacker

This indicator will light to indicate a full Card Stacker. The Processor operation will stop after a Read Order.

To resume the Processor operation without loss of data:

Press the Stop switch.

Remove the cards from the Stacker.

Press the Run switch.

Form Feeding (6 & 7)

FORM (6)

This indicator will light to signify that the supply of forms to be fed is exhausted or that there is a break in the perforation between forms.

The Processor operation will stop when form feeding occurs to or through the next Home position so that the operator can replenish the form supply.

When a new form is installed in the proper position, the operation is resumed by pressing the Run switch.

ADV ✓ (7) - Form Advance Check

Should the form be fed in one skip beyond the permissible maximum (22"), this indicator will light to signify a form "run-away". This would be an uncontrolled skip.

The Processor operation stops automatically within a very short interval.

This stoppage is due to an error in the punching of the Form Control Tape.

After the proper correction has been made to the control and to the form alignment, the operation is resumed by pressing the Run switch.

Card Punching

PUNCH (8)

This indicator will light and the Processor operation will stop in the event of an abnormal condition in the Punch when a Punch function is given.

The Punch Control Panel will indicate the reason for the Processor stoppage at this time.

The lighting of this PUNCH indicator can designate any of the following Punch conditions:

The power cord of the Punch is not connected. The AC and DC indicators will not turn on.

The Punch power switch is not turned on. The AC and DC indicators will not be lit.

A fuse is blown in the Punch. The AC and DC indicators or the DC indicator only will not light.

The Punch covers are not in place. The Interlock (INTL) indicator will be lit.

The punching mechanism in the head of the Punch has been raised and has not been lowered and locked in its proper position. The Interlock (INTL) indicator will be lit.

The Punch reading brushes have been unlocked or removed and have not been reseated and locked in their proper position. The Interlock (INTL) indicator will be lit.

The Input Magazine of the Punch is empty. The HOPPER indicator will be lit.

A Card Stacker of the Punch is full. The STACKER FULL indicator will be lit.

There is a card jam in the Punch. The FEED A JAM or B JAM or the STACKER JAM indicator will be lit.

The Chip Drawer of the Punch is full or is not in place. The CHIPS indicator will be lit and/or the READY Light will be extinguished.

The Punch Check is set to stop the Processor operation when the hole count does not agree.

The Processor operation is resumed, after correcting the Punch condition, by pressing the Run switch.

HALT (9)

There are three conditions under which HALT may light.

1) When last card of Object Deck has been loaded.

2) When machine is running in Single Instruction mode.

3) When an XF HLT instruction is executed.

Auxiliary Card Reader (25 - 29)

These five indicators function when an Auxiliary Card Reader is being used. All areas of the card feeding mechanism of the Auxiliary Card Reader from the Magazine to the Stackers are covered by controls to stop the Processor in the event of mis-feeding. These indicators apply only to the Auxiliary Card Reader, they are not related to the similar indicators 1 - 4 above. The STACKR (5) applies to both Card Readers.

HOPPER (25) - Input Magazine

This indicator will be lit whenever the Input Magazine is empty and the Feed indicator (26) is lit. The Hopper indicator cannot be on alone.

During operation, this indicator will light after the last card is read. The Processor will stop with the last card in Wait Station 2 after the auxiliary read order is executed.

Processor operation is resumed by:

Pressing the Stop switch.

Placing cards in the Magazine.

Pressing the Feed switch of the Auxiliary Card Reader once to feed a card from the Magazine into Wait Station 1; the Hopper and Feed indicators will turn off.

Pressing the Processor Run switch once to resume the operation.

FEED (26) - Wait Station 1

This indicator will be lit by pressing the Clear switch on the Processor Central Control Panel or by a card cycle if there is no card fed to Wait Station 1.

Should this indicator light during operation, a card has failed to feed from the Magazine. If there are cards in the Magazine, the Processor will stop on the next Auxiliary Read order with the Feed indicator lit and the Read not executed.

Processor operation is resumed by:

Pressing the Stop switch.

Removing the cards from the Magazine.

Examining the cards on the bottom of the stack to determine the reason for the failure to feed.

Correcting these cards and returning all cards to the Magazine.

Pressing the Feed switch of the Auxiliary Card Reader once to feed a card from the Magazine to Wait Station 1; the Feed indicator will turn off.

Pressing the Processor Run switch once to resume the operation.

The Hopper and Feed indicators will be lit when the last card has been fed from Wait Station 1 to the Card Stackers. The Processor will stop at the completion of the current Read. If additional cards are to be processed; press the Stop switch, place the cards in the Magazine, press the Auxiliary Card Reader Feed switch and the Processor Run switch.

RD JAM (27) - Read Jam

Should the Processor stop during operation with this indicator lit, either one of the following has occurred:

1. A card from Wait Station 1 may have failed to feed to the Read Photoelectric Diodes.

2. The Read Photoelectric Diodes may have failed the "light-dark" test.

   Before reading the first card and between the reading of each following card, the photo-diodes are in a "light" condition.

When the leading end of a card enters the photo-diode area, a "dark" condition occurs.

This light-dark change must be executed properly to assure correct reading; if it is not, the Processor will stop.

If the stoppage is due to a card jam before the photo-diodes, the Read 2-Execute signal is retained by the Processor; the jammed card was not read. The following procedure will return the Processor to operation without loss of data:

1. Press the Stop switch.

2. Remove all cards from the Magazine and Wait Station 1.

3. Press the Feed switch of the Auxiliary Card Reader once while the Magazine is empty. The Feed indicator will light.

4. Remake the damaged cards, if necessary, and replace them in their proper sequence at the bottom of the stack in the Magazine.

5. Press the Feed switch of the Auxiliary Card Reader once to feed a card from the Magazine to Wait Station 1.

6. Press the Processor Run switch once to resume the operation.

If there is no card jam when the Processor stops with the RD JAM indicator lit, a light-dark test failure is signified. In this case:

The Read 2-Execute signal is retained in the Processor; card reading did not take place, only card feeding.

The card in Wait Station 2 has not been read.

The following procedure should be followed to restore the Processor to operation in the event the light-dark test failure was only momentary:

1. Remove all cards from the Magazine. Remove the card from Wait Station 1. Press the Run Out switch of the Auxiliary Card Reader to feed the card in Wait Station 2 to the Stackers.

2. Follow steps 3 through 6 above. The card from Wait Station 2 should be first in sequence when replacing the cards in the Magazine.

Should the RD JAM indicator light, try the procedure again. If the same indication persists, remake card and try again. If failure continues have the field engineer check the photodiode operation.

WAIT JAM (29) - Wait Station 2 Jam (Photo-Diodes to Wait Station 2)

This indicator will light to indicate the failure of a card to feed to or from Wait Station 2.

To resume the Processor operation without loss of data:

Press the Stop switch.

Remove the mis-fed card or cards.

Press the Clear switch on the Control Panel of the Auxiliary Card Reader.

Press the Processor Run switch.

TSP JAM (28) - Transport Jam (Wait Station 2 to Stackers)

This indicator will light in the event of a jam as the card is delivered to the Stackers.

The Processor will stop.

To resume the Processor operation without loss of data:

Press the Stop switch.

Remove the mis-fed card or cards.

Press the Processor Run switch.

STACKR (5) - Stacker

This indicator will light to indicate a full Card Stacker in the Auxiliary Card Reader as well as in the Card Reader. The Processor operation will stop after an auxiliary read order.

To resume the Processor operation without loss of data:

Press the Stop switch.

Remove the cards from the full Stacker.

Press the Processor Run switch.

B. Display Mask 6.

1	2	3	4	5	6	7	8
LAr 32	LDr 33	LPR 34	SAr 35	SDr 36	SPR 37	SHR 38	SHL 39
00000	00001	00011	00111	01110	11100	11001	10010

9	10	11	12	13	14	15	16
CLR 40	CAr 41	CNr 42	IC 43	J 44	JL 45	JG 46	JE 47
00100	01000	10001	00010	00101	01010	10101	01011

17	18	19	20	21	22	23	24
JR 48	JX 49	AMr 50	ARr 51	SMr 52	SRr 53	MUL 54	DIV 55
10111	01111	11110	11101	11011	10110	01101	11010

25	26	27	28	29	30	31	32
TRL 56	SZS 57	LWS 58	LNr 59	SED 60	PTE 61	XF 62	
10100	01001	10011	00110	01100	11000	10000	11111

Note: JS3, JET, JPE, JC8, JOF, JAL, JI1, and XF functions SI1, RI1, RCD, SNS, SN8, Light the Indicator marked PTE. SC, LOR, LAN, BSH, CCA, XFC Light the indicator marked XF.

Mask 6 is used to determine the operation being executed during single instruction mode. For register designation, refer to Mask 9.

Indicator	1 = LA_r	Load Ascending AR1 or 2
	2 = LD_r	Load Descending AR1 or 2
	3 = LPR	Load Print Descending
	4 = SA_r	Store Ascending AR1 or 2
	5 = SD_r	Store Descending AR1 or 2
	6 = SPR	Store Print Descending
	7 = SHR	Shift Right
	8 = SHL	Shift Left
	9 = CLR	Clear Area to Spaces
	10 = CA_r	Compart Alpha AR1 or 2
	11 = CN_r	Compare Numeric AR1 or 2
	12 = IC	Increment and Compare
	13 = J	Jump Unconditional
	14 = JL	Jump Less (Numeric)
	15 = JG	Jump Greater (Numeric)
	16 = JE	Jump Equal (Numeric)
	17 = JR	Jump Return (Store PAK in X Register)
	18 = JX	Store X Register in M
	19 = AM_r	Add Algebraic AR1 or 2 to M
	20 = AR_r	Add Algebraic M to AR1 or 2
	21 = SM_r	Subtract Algebraic AR1 or 2 from M
	22 = SR_r	Subtract M from AR1 or 2

23 = MUL	Multiply
24 = DIV	Divide
25 = TRL	Translate
26 = SZS	Ø Suppress AR2 and Store Ascending
27 = LWS	Load AR2 with Sign and Zone Delete
28 = LN_r	Zone Delete AR1 and AR2
29 = SED	Edit , , . AR2 and Store Ascending
30 = PTE	Punch Text (See Note 1)
31 = XF	External Functions (See Note 2)

NOTE 1: JS3, JET, JPE, JC8, JOF, JAL, JI1 and XF Functions SI1, RI1, RCD, SNS, SN8 light the indicator marked PTE.

NOTE 2: SC, LOR, LAN, BSH, CCA, XFC light the indicator marked XF.

C. Display Mask 8.

1	2	3	4	5	6	7	8
1MSR5	2MSR4	3MSR3	4MSR2	5MSR1	6MSC5	7MSC4	8MSC3
00000	00001	00011	00111	01110	11100	11001	10010

9	10	11	12	13	14	15	16
9MSC2	10MSC1	11LSR5	12LSR4	13LSR3	14LSR2	15LSR1	16LSC5
00100	01000	10001	00010	00101	01010	10101	01011

17	18	19	20	21	22	23	24
17LSC4	18LSC3	19LSC2	20LSC1	CCB	CCA	CBB	CBA
10111	01111	11110	11101	11011	10110	01101	11010

25	26	27	28	29	30	31	32
CAB	CAA	ICA	ICB	ICC	ICD	OCA	OCB
10100	01001	10011	00110	01100	11000	10000	11111

MSR = Most Significant Row
MSC = Most Significant Column
LSR = Least Significant Row
LSC = Least Significant Column

Mask 8 displays the operand of the instruction being executed during single instruction mode. For operand bank designation, refer to Mask 9.

INDICATORS 1-5 represents all but the "X" bit of instruction character 2. (Most significant row)

IND. 1 = Y bit
2 = 8 bit
3 = 4 bit
4 = 2 bit
5 = 1 bit

INDICATORS 6-10 represents all but the "X" bit of instruction character 3. (Most significant column)

IND. 6 = Y bit
7 = 8 bit
8 = 4 bit
9 = 2 bit
10 = 1 bit

INDICATORS 11-15 represents all but the "X" bit of instruction character 4. (Least significant row)

IND. 11 = Y bit
12 = 8 bit
13 = 4 bit
14 = 2 bit
15 = 1 bit

INDICATORS 16-20 represents all but the "X" bit of instruction character 5. (Least significant column)

IND. 16 = Y bit
17 = 8 bit
18 = 4 bit
19 = 2 bit
20 = 1 bit

INDICATORS 21-32 reference internal maching cycles and is primarily used for engineering maintenance.

D. Display Mask 9

1	2	3	4	5	6	7	8
C1 > −	C1 ≠<+	C1 = Ø	C2 > −	C2 ≠<+	C2 = Ø	C3 > −	C3 ≠<+
00000	00001	00011	00111	01110	11100	11001	10010
FALL	JMP						

9	10	11	12	13	14	15	16
C3 = Ø	C4 > −	C4 ≠<+	C4 = Ø	C5 > −	C5 ≠<+	C5 = Ø	C6 > −
00100	01000	10001	00010	00101	01010	10101	01011
							ch8

17	18	19	20	21	22	23	24
C6 ≠<+	C6 = Ø	C7 > −	C7 ≠<+	C7 = Ø	C8 > −	C8 ≠<+	C8 = Ø
10111	01111	11110	11101	11011	10110	01101	11010
IC1X	IC2X	IC3X	IC4X	IC5X	PE	EOT	

25	26	27	28	29	30	31	32
C9 > −	C9 ≠<+	C9 = Ø	C10 > −	C10 ≠<+	C10 = Ø	MAINT B	MAINT C
10100	01001	10011	00110	01100	11000	10000	11111

Mask 9 displays various indicators and registers in the 1005. Of interest to the programmer are the following:

INDICATOR

1. If this indicator is lit on a conditional jump, the condition is not met.

2. If this indicator is lit on a conditional jump, the condition is met.

16. A paper tape channel eight punch has been sensed.

17. Instruction character One "X" bit present.

18. Instruction character Two "X" bit present.

19. Instruction character Three "X" bit present.

20. Instruction character Four "X" bit present.

21. Instruction character Five "X" bit present.

NOTE 1: Instruction character one "X" bit determines the register (when applicable) the instruction will use.

"X" bit absent = Register 1

"X" bit present = Register 2

NOTE 2: Instruction characters four and five determine the bank designation. The following table of bits illustrate bank addressing:

"X" Bit Char. 4	"X" Bit Char. 5	Bank Designation
Absent	Absent	1
Present	Absent	2
Absent	Present	3
Present	Present	4

22. Paper tape parity error, magnetic tape parity error, DLT Mod Error, or invalid card code has been detected.

23. End of magnetic tape has been sensed.

UNIVAC

FEDERAL SYSTEMS DIVISION

FSD 1089.1 APRIL 1968