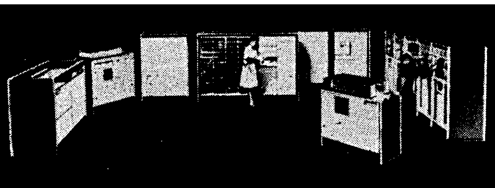


UNIVAC®

SOLID-STATE SYSTEMS

GENERAL MANUAL

S4

ASSEMBLY

80 CARD

80 CORE

90 CORE

80 TAPE

90 TAPE

UP-1774.6 Rev. 1

P R E F A C E

This manual is intended as a programmer's reference manual to familiarize the programmer with the basic elements of the S-4 80 Card, 80/90 Tape-Core Assembly System. The S-4 assembler is designed to facilitate the coding of data-processing applications by reducing both coding time and error frequency.

S-4 has been designed for use with the UNIVAC® Solid-State computing systems and a knowledge of these systems is presumed. Since this manual is a comprehensive discussion of S-4, options pertaining only to a specific computing system will be so noted. References to input-card format in this manual will apply to both 80- and 90-column cards. References to 80-column cards will be enclosed in parentheses. For example, the notation 51 (46) indicates a reference to column 51 on the 90-column card and 46 on the 80-column card.

The 80 system is issued in two forms: a multi-load card instruction deck with card overlay sections and a scatter format tape load also with card overlays. The 90 system is available only in scatter-format tape.

TABLE OF CONTENTS

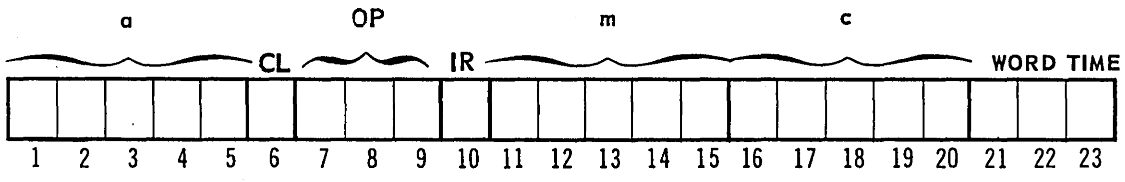
I. INTRODUCTION

The S-4 Assembly System is a one-pass assembly-language translator. Input is a source program coded in S-4 assembly language. Output is an object program in UNIVAC Solid-State machine code acceptable either to a standard loading-routine format or a self-loading program on magnetic tape. For documentation and desk-checking purposes, printed listings of both the source and object programs, and printouts of the various tables of the assembly system are provided.

The source program for an S-4 assembly is written on the coding form applicable to the computer on which the assembly will be made (see Appendix A). Each line consists of 49 (44) character positions, and is equated to a particular columnar position on the punched card. The four leftmost character positions are primarily for the convenience of the programmer and are used to specify program line-sequence. This information, though punched on the card, is ignored by the assembler since it sets up its own line or card sequence.

A. INSTRUCTION FORMAT

The basic S-4 instruction is specified in the following format:



It consists of a symbolic a, m, and c field; a symbolic OP code field; a class designation, an index register specification and a word-time field. The instruction is contained in columnar positions 46-68 (41-63). Explanation of each element in the instruction format follows.

1. Symbolic "a" Field

   This field, specified in columns 46-50 (41-45), contains the address of an instruction or constant and may be any one of the following:

   a. Absolute machine address

   b. Permanent tag

   c. Temporary tag

   d. Local reference point

   e. Blank address

   f. Interlace address

   g. Overflow address

   h. Register address

   i. Region address

   The various types of specifications listed above are discussed in chapter 3.

2. Class Field

   This field, specified in column 51 (46), is used when the information in the remaining fields, columns 52-90 (47-80), is to be treated in a special way by the assembly program. Any one of ten specifications may be made in this field. The following is a list of the specifications and the actions they cause the assembler to perform:

<u>Character</u>	<u>Assembler Action</u>
C	The data in columns 52-90 (47-80) is to be treated as a comment and therefore, carried over to final output unchanged.
D	If the line represents a constant, the duo-primed portion of the contents of columns 56-65 (51-60) will be stored when the object program is run (Solid-State 80 only).

P	If the line represents a constant, the primed portion of the contents of columns 56-65 (51-60) will be stored in the object program.
U	If the line represents a constant, the unprimed portion of the contents of columns 56-65 (51-60) will be stored in the object program.
N	If the line represents a constant, the contents of columns 56-65 (51-60) will be translated and the numeric portion stored in the object program.
Z	If the line represents a constant, the contents of columns 56-65 (51-60) will be translated and the zone portion stored in the object program.
I	The tens complement of the data produced from 56-65 (51-60) will be stored in the object program.
2	Normal processing occurs, but the sign of the result will contain a 4 bit to designate: a. that the second synchronizer will be used if a tape handling instruction is specified. b. that special alphabetic processing is to take place.
Δ	No special action imposed.
*	The line is treated as a title line and is printed on the assembly record at the beginning of a new page.

3. Symbolic "OP" Field

The 3-digit symbolic operation-code field, columns 52-54 (47-49) may contain any of the following:

a. ΔΔΔ (blanks) to indicate that the content of the m and c fields, columns 56-65 (51-60) is a constant and should be treated as column 51 (46) directs.

b. A 3-digit symbolic OP code or an S-4 control operator. If a symbolic OP code, it will be translated to its machine code equivalent during assembly.

c. A 2-digit machine coded instruction in form Δnn. It will appear in final output unchanged. If this option is used, the word-time field must also be employed.

4. Index Register Field

The index register field, column 55 (50), contains a numeric specification when index-register modification is indicated for the m address of the associated instruction:

Entry	Meaning
1	Use index register 1.
2	Use index register 2; load negative.
3	Use index register 3.
4-9	Use index registers 4 through 9. These registers are employed only for the USS-II computers with nine index registers.

5. Symbolic "m" Field

The symbolic m field, columns 56-60 (51-55), may contain any of the entries specified for the symbolic a field. In addition, it may also contain a shift counter or a stacker-selection specification. The field may also be part of a constant entry.

6. Symbolic "c" Field

The symbolic c field, columns 61-65 (56-60), specifies the next instruction and may contain any of the entries indicated for the symbolic a field. It may also be part of a constant entry.

7. Word-Time Field

The word-time field, columns 66-68 (61-63), is analyzed during assembly. If it contains a numeric specification, the numeric value will be assigned as the maximum number of word-times for the operation specified in the OP field. If the field is not entirely numeric, it is treated as part of the remarks field. When the entry in the symbolic Op field is a machine coded instruction, the number of word-times from a to c is specified by the programmer in the word-time field. With this, an additional specification is entered in the most significant position of the field, column 66 (61), to direct the assembly of the instruction line. The format of the entry in the word-time field is;

xnn

where; x is a Δ or 5 bit if c specifies the next instruction.

4 bit if m specifies the next instruction.

7 bit when m is not to be used to update the latency counter (the clock).

1 bit if x is considered part of nn field.

nn is the number of word-times between a and c.

If three digits are needed to express the number of word-times, the 1 bit in the x position may be employed; however, the above mentioned assembly-directing code bits,if not equal to zero,will also be buffed onto this position.

B. REMARKS

The remarks field, columns 66-90 (61-80) if the word-times field is not used, contains comments concerning the source program. All data in this field will be printed as part of the side-by-side object and source-code listing printed by the assembler.

C. DATA CONSTANT CODING

The following considerations apply when data constants are specified in the source program:

1. The OP portion of the instruction will contain blanks (ΔΔΔ).

2. The IR field indicates the sign of the constant;

   Δ if positive

   - if negative

3. The constant is entered in the symbolic m and c fields.

4. If the constant contains undigits, the class field must be blank (Δ).

5. A blank in the constant field will be interpreted as a zero. If a space is required in the output constant field, a B is entered in the space position.

6. The class field may contain any of the values I, N, Z, 2, U, P, D, or Δ.

7. The specifications U, P, or D produce card-coded constants.

II. S-4 A D R E S S I N G

The S-4 Assembly System provides for many types of addressing to ensure a high degree of programming flexibility. That is, programmers are able to select, from a variety of addressing forms, that format most suited to a particular need in a computer application. This capability increases the power of the system as a programming tool and provides a versatility not present in ordinary machine coding. The following is a listing of the various addressing forms provided by the S-4 System:

1. Blank Addressing
2. Tag Addressing
3. Absolute Addressing
4. Regional Addressing
5. Interlace Addressing

A. BLANK ADDRESSING

If the generation of absolute addresses in the object program is to be left to the assembler, the a, m, or c portion of the instruction involved may be left blank. If either the m or c field of an instruction is left blank, it will be an indication to the assembler that a reference is being made to the next consecutive line of coding; therefore, the a field of the next instruction must also be left blank to allow the assembler to assign the same absolute address as the m or c of the previous instruction which made the reference. For example:

a	c	OP	IR	m	c
X		LDA			Y
				00000	00005
Y		STA		Z	
		LDL		P	Q

Assuming that X, Y, Z, P, and Q are some form of S-4 address specification, the converted machine-coded version might appear in final output as;

a	OP	m	c
0404	25	0406	0408
0406	00	0000	0005
0408	60	0410	0412
0412	30	0414	0557

When both the m and c symbolic address fields are left blank, the symbolic a fields of the next two instructions in sequence must also be left blank. The blank m field will reference the next line in sequence; the blank c address will reference the second line down. For example;

a	C	OP	IR	m	c
X		LDA			
				00000	00001
		ADD		Y	Z
Z		STA			

Assuming that X, Y, and Z are some form of S-4 address specification, the converted machine-coded version might appear in the final output as;

a	OP	m	c
0200	25	0202	0204
0202	00	0000	0000
0204	70	0207	0209
0209	60	....	....

It should be noted that absolute addresses will be assigned in the object program only if the blank m or c portion is normally specified in the machine-coded instruction. That is, certain instructions require no specification in either the m or c field; for example a 26 (CLA) or 06 (CLX) in which the c portion is not specified, or the 77 (ATL) in which the m address is not specified.

B. TAG ADDRESSING

A tag is symbolic specification or address that relates nonconsecutive lines of coding. Tags provide connecting links between operations by relating the m or c portion of an instruction with the a portion of another instruction that has been, or is yet to be specified. They may be used to denote the entrance and exit lines of a common subroutine; to transfer from one operation to another; to reference lines that are to be modified; or to transfer control to a common line at the end of a branching chain of instructions.

Provision is made in S-4 for three types of tag specification:

1. Permanent tags

2. Temporary tags

3. LRP (Local Reference Point) tags

Permanent tags are employed to preserve relationships that will be maintained throughout the program. That is, since programs are normally subdivided into logical units or sections, the permanent tag provides a method of referencing either across or within these program sections. Temporary tags are generally employed to establish relationships between lines of coding within a logical section of the program and are generally not referenced by lines of coding from another section. The LRP tag is a special form of temporary tag. It is generally used within comparatively short coding segments and allows a relationship to be established without exhausting the combined total of 300 temporary and permanent tags permitted in a program.

As each tag is specified, it is entered in a tag table along with its assigned absolute address. Temporary tags may be cleared from the tag table at any time (but usually at the end of a logical program section) to permit their reassignment in another portion of the source program. Permanent tag entries are maintained in the tag table throughout the program. However, should a permanent tag become inactive (that is, no reference made to it during the remainder of the program), its assigned absolute address may be cleared from the table and a new address assigned. Clearing of the tag table will be discussed when the S-4 control operators are considered.

Program subdivision is left entirely to the discretion of the programmer since no formal method is provided by the S-4 System. It should be noted, however, that sections assembled first will receive preferential treatment as far as optimization is concerned. Therefore, it is in the interest of the programmer to assemble the most important sections first.

1. Permanent tags

The permanent tag is specified in the a, m, or c field in the following format:

x n n n m

Here; x is any alphabetic or special character.

nnn is any combination of alphabetic, alphanumeric and/or special characters. This specification must not be entirely numeric.

m is the area in storage to which the tag is to be assigned and should contain one of the following:

a. A blank for standard-access memory assignment

b. Any character for high-speed-access memory assignment except a C, O, P, Q, or R.

c. A C for core memory assignment.

d. An O, P, Q, or R for overflow (c+1) condition. (See "Overflow Addressing.")

Ideally, a permanent tag specification should, in some way, be indicative of the function performed by the tagged procedure or should conform to some meaningful tag coding scheme. For example, the tag

E D I T C

might be the tag specification for the entrance line of an editing subroutine that is to be stored in core memory; or the permanent tag specification

G R O S S

might specify the location at which the result of a gross pay compution is stored in high-speed-access memory.

2. Temporary Tags

A temporary tag is specified in the a, m, or c field in the following format:

x n n b y

Here; x n n is the tag identifier

x may be any alphabetic, numeric or special character except Δ.

nn may be any two-digit numeric if the tag is to be assigned to standard-access memory; it must be blank if the tag is to be assigned to high-speed-access memory.

b is blank if fast-access memory is to be assigned; it is numeric if high-speed-access memory is to be assigned.

y must be a numeric if b is numeric. If b is blank, y may be one of the following:

C for core storage assignment

O, P, Q, or R for an overflow condition.

When employing temporary tag specifications, the following should be observed:

a. Absolute locations may be assigned to temporary tags by the programmer.[1]

b. Individual tags may be cleared from the tag table (released for reassignment) at any time during an assembly.[2]

c. The tag table may be entirely cleared of temporary tags at any time during an assembly and new temporary tags initiated.[3]

d. Once a tag has been cleared from the table, any further reference to the tag is treated as if no previous reference had appeared, consequently, a new absolute address will be assigned.

---

[1] See SYN Control Operator.

[2] See EQU Control Operator.

[3] See HED C Control Operator.

3. Overflow (c + 1) Addressing

Overflow and c + 1 conditions can result from either an arithmetic operation or an abnormal condition in an input or output unit. In an arithmetic operation, it is caused by the generation of a numerical quantity beyond the digit capacity of the register that is to receive it. In an input or output unit, it might be the result of any of a number of mechanical conditions (Printer out of paper, RPU card jam, for example). In either case, the instruction that will be executed next is determined by the addition of 1 to the c address of the instruction in which the overflow or c+1 condition occurs or is detected.

There are eight instruction codes that can result in overflow or c + 1 conditions:

S-4 Coding	Machine Code
ADD	70
SUB	75
DIV	55
PRN	11
PFD	16
HCC	72
RCC	81
TBU	F6

Whenever one of these codes is used, a subroutine should be coded that will handle the possible overflow or c + 1 condition. In S-4 coding this is accomplished by the use of temporary or permanent tags with an O, P, Q, or R in the LSD of the tag.

If there is no overflow, control will be sent to the instruction coded with the O or Q tag in the a address portion. If overflow occurs, control is sent to the instruction containing the P or R tag in the a address portion.

The O and P tags are used when drum storage is to be used. The Q and R tags are used when core storage is to be used. This is necessary because drum addresses are modulo 200 and core storage addresses are modulo 1000. Thus, in those cases where drum storage is at a 199 band level for an O tag, the P tag will be assigned at level 000 in the same band. When core storage for a Q tag is B999, B000 will be assigned to the R tag line.[4]

[4] The Core Storage addresses G000 through G279 will not be used for overflow addressing.

When coding for overflow and c + 1 conditions, the following should be observed:

a. The c address portion of the line in which overflow may occur must be in the O or Q form of a permanent or temporary tag.

b. O, P, Q, and R tags do not have to follow the line in which overflow may occur but may be placed at any point during the assembly. The only restriction is that when temporary tag form is used, all program references must be made before a HED C pseudo operator is introduced.

c. Overflow tags must be counted as part of the tag limits. Each set (O and P; Q and R) is counted as one tag.

d. Unless a HED F control operator is in effect, O and P tags will be assigned to fast-access memory.

Examples of overflow (c + 1) coding:

a	OP	M	c
	LDA	X12ΔΔ	
	ADD	X13ΔΔ	L01ΔO
L01ΔO	STA	X14ΔΔ	N1ΔΔΔ
L01ΔP	JMP		

The temporary tag L01 is labeled O on the LSD for its normal a address. The c + 1 tag contains a P in the LSD.

a	OP	m	c
	LDA	X12	
	ADD	X13ΔΔ	L01ΔQ
L01ΔQ	STA	X14ΔΔ	N1
L01ΔR	JMP	ERROR	

The tag for the normal a address has a Q in the LSD position. The c + 1 address tag contains an R in the LSD.

4. LRP (Local Reference Point) Tags

LRP tags permit relationships to be established between nonconsecutive lines of coding without using permanent or temporary tags. The LRP tags a line with which communication is to be made, through an m or c portion of a prior, a succeeding, or the same instruction.

The LRP tag is specified in the following format:

nΔΔΔx

Here; n is the LRP identifier and must be a numeric in the range 0-9.

ΔΔΔ is always blank.

x is the storage-allocation position and may be:

Δ (blank) if standard-access memory is to be assigned.

H if high-speed-access memory is to be assigned.

An LRP tag is referenced in the following manner. Note that the reference must indicate the direction or relation of the tagged line to the line referencing it; that is, whether the tagged line is a previous, a succeeding, or the same instruction. The format is:

nyΔΔx

Here; n is the identifier assigned to the LRP tag (0-9)

y is the direction indicator when the LRP being referenced is assigned by the assembler to <u>standard</u>-<u>access</u> storage; otherwise, this position is blank (Δ).

x is the direction indicator when the LRP being referenced is assigned by the assembler to high-speed access storage; otherwise, this position is blank (Δ).

ΔΔ is always blank.

Note that either x or y may be specified at one time and never both. One or the other will always be blank depending on the storage assignment of the LRP. The following may be entered in either x or y:

B if the LRP-tagged line, with which communication is made, exists in a backward direction from the referencing line. B may be either in y or x and is the only reference not dependent on storage assignment.

F if the LRP-tagged line, with which communication is made, exists in a forward direction from the referencing line.

H if the LRP-tagged line, with which communication is made, is the same as the line in which the reference occurs. If this specification is employed but there is no LRP tag in the a address, the assigned address of the current entry in the a field is established in the LRP table as the address of that LRP tag.

LRP tags are assigned in a two-part tag table, based on their order of specification. The two parts of the table are designated B(backward) and F(forward). When an LRP is specified in a field, its absolute address is entered in B. For example, the LRP 1Δ␣␣H indicates that a high-speed storage assignment is to be made for LRP 1. The table entry is made in the following manner:

LRP	B	F
0		
1	4211	
2		
3		
4		
5		
6		
7		
8		
9		

From this point on, all backward references to LRP 1 will address 4211. The address will remain valid until a new LRP 1 appears in a succeeding instruction. When a new LRP 1 appears, 4211 will be cleared from the table and a new assignment will be made.

If a forward reference to an LRP is made from an m or c field, its absolute value is assigned in F. For example, 1FΔΔ indicates that the next LRP 1 encountered is in standard-access memory. The table entry is then made in the following manner.

LRP	B	F
0		
1		1342
2		
3		
4		
5		
6		
7		
8		
9		

When the LRP 1 is encountered in some succeeding instruction, it shifts from F to its corresponding position in B clearing any entry in that position. Until this shift occurs, any entry made for a previous LRP 1 in the B column will remain valid.

LRP	B	F
0		
1	1342	
2		
3		
4		
5		
6		
7		
8		
9		

Examples of LRP coding:

a.

a	OP	m	c
1	LDA	GIN	SIN

LRP 1 is assigned in standard access storage.

b.

GIN	LDA	1B	SIN

1B refers to previous LRP 1.

c.

GIN	STA	SIN	2F

Next LRP 2 encountered is in standard-access storage.

d.

4 H	LDL	GIN	SIN

LRP 4 is assigned in high-speed access storage.

e.

GIN	LDX	4F	SIN

Next LRP 4 encountered is in standard-access storage.

f.

SIN	LDX	5H	GIN

LRP 5 is assigned the address given to SIN.

g.

GIN	LDL	SIN	2 B

2 B refers to previous LRP 2.

h.

GIN	LDA	2 F	SIN

Next LRP 2 encountered is in high-speed access storage.

C. ABSOLUTE ADDRESSING

Fixed computer locations (absolute addresses) are referenced in the following manner:

Δ n n n n

Here; Δ must be blank.

nnnn must be the specific memory address and must be within the following limits:

Drum Addresses	Δ0000	through	Δ9199
Core Storage Addresses	ΔB000	through	ΔB999
Core Storage Addresses	ΔG000	through	ΔG279

The following table illustrates the absolute address to which the G series is converted:

G Address	Machine Address
G000 - G099	B00A - B99A
G100 - G199	B00F - B99F
G200 - G209	B0FH - B9FH
G210 - G219	B0FG - B9FG
G220 - G229	B0FC - B9FC
G230 - G239	B0FB - B9FB
G240 - G249	B0AH - B9AH
G250 - G259	B0AG - B9AG
G260 - G269	B0AC - B9AC
G270 - G279	B0AB - B9AB

An address coded in the above manner will not be modified in any way. When absolute addressing is used, the locations specified must be restricted from assignment before the source program instruction lines are assembled. This is done by specifying such locations, or groups of locations with appropriate control operators.

D. REGISTER ADDRESSING

When it is necessary to address a register, the address is coded by using the two most significant digits of the desired symbolic address field.

R n Δ Δ Δ

Here; R must be R.

n may be A, X, or L depending upon the register to be referenced.

ΔΔΔ must be blank.

Whenever program control passes to a register address, the contents of the register should be displayed on the next line with the appropriate register address in the symbolic a field.

Register addresses always produce the following absolute addresses:

RAΔΔΔ	000A
RXΔΔΔ	000C
RLΔΔΔ	000B

E. REGIONAL ADDRESSING

Data, or instructions stored as data, are placed in reserved areas of memory known as regions.[5] An entry in any of these areas may be referenced by a regional address. The format of a regional address is:

a n n n n

Here; a is any alphabetic or non-blank special character.

nnnn is the entry number within the region area. This will be in the range 0001 through the highest entry number reserved. Thus, if 200 locations, 0200 through 0399, have been reserved for region B. The B regional addresses would be B0001 through B0200. B0001 would be location 0200; B0009 would be 0208, etc. If 50 locations four word-times apart have been reserved in bank 10 for region C, the C regional addresses would be C0001 through C0050. C0001 would be location 1000; C0002, 1004, etc.

[5] See REG Control Operator.

F. INTERLACE ADDRESSING

Reference to an input or an output interlace is accomplished by use of symbolic interlace address.[6] The format of this address is:

k i m n z

Here; k represents the input/output device and must be one of the following:

H - High-Speed Reader Interlace.
R - Read-Punch Unit Read Interlace.
O - Read-Punch Unit Output Interlace.
P - High-Speed Printer Interlace.
T - Tape Interlace.
D - RANDEX Drum Interlace. (Also for first or second tape interlace if desired).

i represents the number of the interlace and must be a numeric in the range 0 through 9. S-4 provides ten interlace patterns for each input/output unit.

m refers to the particular level of an interlace word or to the band of the interlace. It must be one of the following:

U - Unprimed portion of a word
P - Primed portion of a word
D - Duo-Primed portion of a word (USS 80 only)
N - Numeric portion of a word
Z - Zone portion of a word
B - Entire interlace (used with buffer load and unload instructions)

nz specifies the word referenced. This specification depends upon the input or output unit designated by k. When k is H, R, or O.

n is 1 when the word being addressed is located in the <u>first read interlace</u> of the Card Reader or RPU; or the <u>punch interlace</u> of the RPU.

[6] See INT Control Operator.

n is 2 when the word being address is
located in <u>second read interlace</u> of the
Card Reader or RPU.

z is the card word; 0-9 (0-7).

When k is T;

nz is a numeric in the range 00 through 99.

When k is P;

nz is a numeric in the range 01 through 13.

When k is D;

nz is a numeric in the range 00 through
47. If a number greater than 47 is
specified, it will be treated as a
tape interlace and a note will be
printed on the printer listing. This
capability permits 20 interlace pat-
terns to be specified for tape as
opposed to ten for all other units.

When an entire interlace is specified (m is B)

nz is 00 for the untranslated interlace.

01 for the translated interlace.

nz is the number of lines to advance on
the printer if k is P.

III. S-4 CONTROL OPERATORS

Command of the S-4 Assembly process is exercised by the use of Control Operators. Their function is to provide:

1. Space reservation.
2. Memory allocation controls.
3. Tag input/output controlling commands.
4. Access to, and control of, tag table content.
5. The use of a Constant Library.
6. Development of program testing aids and their inclusion in an object program.

Control operators are coded in the Symbolic Operation Field as three digit mnemonics.

A. ASSEMBLY CONTROL OPERATORS

1. RST - Initialize For Assembly

The RST Operator sets conditions for an assembly, By using the RST Operator between programs, a series of programs can be assembled in one computer run. The functions performed by the RST Operator are:

a. Sets Card Reader listing page number to 1.

b. Clears the Availability Table.

c. Clears the Symbol Table.

d. Clears the Interlace Table.

e. Clears the Region Table.

f. Stores the new program title from the RST Operator (see RST Operator format, below).

g. Clears the Card Number Counter.

h. Clears the Word Time Clock.

i. Initializes the assembly program to non-forward search mode.

j. Initializes modes of some Control Operators

HED D

HED K

HED N

HED P (80CTC), HED M (90 TC)

HED Y

The RST format is:

OP	m	c
RST	ppppp	ppppp

RST is the mnemonic used in the symbolic Op field.

pppppppppp is ten digit position for the alphanumeric and/or special characters that identify the source program to be assembled. Unused positions are coded as blanks (Δ). This specification is punched in columns 1 - 10 of the output cards, goes into word 1 of the output tape and is printed in columns 121 - 130 of the page header-line of the printed listing.

2. END - End Card Output Assembly

The End operator will punch a sentinel card in PTA format when used as the last card of a card output assembly.

OP	m	c
END	sssss	ΔΔΔΔΔ

sssss = Symbolic tag used to denote start of output program. The output created by the END line is an instruction of the form

00 mmmm 0000

where mmmm is the absolute address derived from sssss.

B. STORAGE ALLOCATION OPERATORS

In any program, certain areas and/or locations must be restricted from assignment during the assembly process (data storage locations, interlaces, tables, packaged subroutine locations, etc.). In S-4 coding, this is accomplished by the use of certain Control Operators.

1. BLR - Block Reservation

The BLR Operator is used to reserve a given number of locations at a fixed increment from each other beginning at a specific address and ending at a specific address.

The format is:

OP	m	c	w/t
BLR	bbbbb	eeeee	iii

BLR is the mnemonic for Block Reservation.

bbbbb is the absolute address or defined symbol at which block reservation is to begin.

eeeee is the absolute address or defined symbol at which reservation is to end.

iii is the increment between locations. If this field is blank or 000 the increment is considered to be 001. Increments less than 200 are modulo drum size or core. Increments greater than 200 are modulo 200 in drum memory and will remain within the band specified.

Examples of BLR coding:

a.

OP	m	c	w/t
BLR	0403	0793	005

This would reserve every fifth location, beginning with 0403, through 0793.

b. BLR 4400 4599 257

This would reserve every fifty-seventh location within band 44.

c. BLR GET START

This would reserve every location between the previously defined tags GET and START.

d. BLR B201 B400

This would reserve every location between B201 and B400 in core memory.

2. BLA - Block Availability

The BLA operator makes available a given number of locations at a fixed increment from each other beginning at a specific address and ending at a specific address. It is the reverse of the BLR Operator.

The BLA coding is in the same format as that of the BLR.

Examples of BLA coding:

a.	BLA	0403	0798	005

This would make available for S-4 Assembly assignment every fifth location, beginning with 0403, through 0798.

b.	BLA	4400	4599	257

This would make available for S-4 Assembly assignment every fifty-seventh location within band 44.

3. REG - Regional Specification

The REG Operator defines a region composed of a specified number of elements beginning at a certain location and separated by a given increment. REG coding is in the following format:

OP	m	m	w/t
REG	xnnnn	yyyyy	iii

REG is the mnemonic for Regional Specification.

x is an alphabetic or non-blank special character.

nnnn is the absolute address at which the region is to begin.

yyyyy is the absolute address or defined symbol at which the region is to end. If yyyyy is blank, the region will be defined but the elements of the region will not be restricted in the memory table.

iii as defined under BLR

Examples of REG coding:

a. REG A1700 Δ1842 ΔΔΔ

This would reserve every location from 1700 through 1842 for Region A. ΔΔΔ could also have been coded 000 or 001.

b. REG B1200 Δ1350 010

This would reserve every tenth location from 1200 through 1350 for Region B.

c. REG S4600 ΔΔΔΔΔ 203

This would set up every third location within band 46 (modulo 200) as region S. The region will not be restricted because yyyyy is blank.

4. INT - Interlace Pattern Reserve

The INT Operator reserves an interlace for the input/output unit specified in the m field. This interlace will be located in the memory area specified in the symbolic c field:

OP	m	c
INT	xyΔΔz	Δnnnn

INT is the mnemonic for Interlace Pattern Reservation.

x is the input/output unit:

H - Card Reader Interlace.

R - Read-Punch Unit Read Interlace.

O - Read-Punch Unit Output (Punch) Interlace.

P - High-Speed Printer Output Interlace.

T - Tape-Synchronizer Interlace.[7]

D - RANDEX Drum Interlace (also for first or Second Tape Synchronizer Interlace if desired).*

[7]See Tape Interlace, Page 4-7.

y is the number of the interlace and must be a decimal digit in the range 0 through 9. This allows up to ten interlaces for each input/output unit (twenty for tape since both T and D may be used).

ΔΔ these digits are always blank.

z is 0 if automatic translation is not to be used (unless x = D).

is 1 if automatic translation is to be used (unless x = D).

is Δ(blank) if the input/output unit involved does not use translation (such as Tape and RANDEX units).

If x = d, z = 0 if a RANDEX input interlace is desired (200 locations). z = 1 if a RANDEX output interlace is desired (48 locations).

Δnnnn is space followed by the memory area in which the interlace is to be reserved. When interlace reservation is to be made on the drum, nnnn must be an even band number (0200, 0400, 1000, 4200, etc.).

Tape interlaces defined in core memory with an INT Operator must begin at the 01 level (e.g., B001, B401, B701, etc.). Two hundred consecutive locations are restricted for a core interlace.

It should be noted that while overlapping of interlaces is permissible, the condition must be kept in mind when coding the source program.

Examples of INT coding:

a. INT H1ΔΔ1 2000

This would reserve the locations for a Card Reader translated interlace on band 20.

b. INT D1ΔΔ0 0800

This would reserve the locations for a RANDEX input interlace on band 08.

5. SYN - Synonym

The SYN Operator will reserve a single location. It may be used for the following purposes:

a. To equate a tag to specific memory location.

b. To provide a time relationship between two tags.

The coding of an SYN Operator is:

OP	m	c	w/t
SYN	xxxxx	yyyyy	iii

SYN is the mnemonic for Synonym

xxxxx is the symbolic address which is to be equated to the content of the symbolic c field.

yyyyy may be 1) a previously defined symbol, the location of which is assigned.

2) an absolute address to be assigned.

If yyyyy is an undefined symbol the SYN Operator will be bypassed and an error note 5 will be printed.

iii is the word-time increment to be added to yyyyy before assigning the xxxxx address.

Examples of SYN coding:

a. SYN STOP△ △0674

This will cause the tag STOP to be assigned 0674 as its address. If tag STOP had already been assigned an address, this would establish 0674 as the address.

b. SYN JOE△△ SAM△△ 015

The tag JOE will be assigned an address fifteen word-times greater than that assigned to tag SAM.

c. SYN ENT△△ EXIT△ 030

The tag ENT will be assigned an address thirty word-times greater than that assigned to tag EXIT.

C. ALLOCATION-CONTROL OPERATORS

Certain control operators included in the S-4 Assembly System permit control of the allocation processes governing the address assignments of instructions, constants, and tag addresses without source program revision. Thus, latency needs, discernable only from an over-all understanding of a source program, can be met.

These Allocation Control Operators are coded in the symbolic OP field and the most significant digit position of the symbolic m field:

OP	m
HED	a△△△△

HED is the mnemonic used in the Symbolic Op Field.

a△△△△ is the alphabetic designating the desired Allocation Control Operator followed by four blanks.

The Allocation Control Operators are:

1. HED B - Initiate Forward Search.

   The use of a HED B Operator will cause a scanning of lines ahead of the line which an undefined symbol is encountered. This scanning will proceed until:

   a. A "C" is found in column 51 (46).

   b. A constant is detected.

   c. Ten lines have been scanned.

   d. A HED A operator is encountered (this stops it permanently).

   e. Any operator is encountered (this stops it temporarily).

When any one of these conditions is met, forward search is terminated for that sequence and allocation is made on a reverse direction; that is, from the line on which forward search is stopped, back to the line in which it was begun. When forward search has been initiated, it will continue to operate until a HED A control operator is encountered. That is, when any of conditions a through d has been met, allocation is made. Normal allocation is then resumed until an undefined symbol is encountered; forward search again takes effect. This process will continue until a HED A control operator is found.

Under certain conditions, a HED B operator will have no effect:

a. the memory table is filled.

b. the symbol table is filled.

2. HED A - End Forward Search.

The HED A Control Operator terminates the forward search initiated by a HED B Control operator, if any.

3. HED D - Extend to High-Speed Memory (when necessary).

The HED D Control Operator, in effect, extends allocation from standard to high-speed memory for minimum latency address assignment. That is, if a HED D control operator is in effect, and an unassigned address cannot be optimally assigned in standard memory, high-speed-access memory is examined for an optimum location. If such a location is found, the assignment is made. If no such assignment is possible in the high-speed area, the standard area is searched for the next best location. If not found, high-speed memory is searched. This process continues until assignment is made.

4. HED H - Extend to Core Storage (when necessary).

The HED H Control Operator operates in the same manner as the HED D except that assignment will be made in core storage if optimum assignment cannot be made in standard or high-speed-access memory.

5. HED E - Terminate HED D And HED H Functions.

The HED E Control Operator eliminates the allocation modes initiated by either a HED D or HED H Control Operator.

6. HED F - Assign High-Speed Storage.

The HED F operator will cause all succeeding unassigned symbolic and blank addresses to be assigned in high-speed access memory. If core storage memory has been specified in the final digit of the symbolic address, however, core storage assignment will be made.[8]

7. HED G - Assign Core Storage.

The HED G Operator will cause all succeeding unassigned symbolic and blank address to core storage.[8]

8. HED J - High-Speed Tags to Core Storage.

The HED J operator will cause all succeeding unassigned high-speed tags to be assigned in core storage. Standard tags will continue to be assigned to standard memory.[8]

9. HED N - Resume Normal Allocation.

The HED N operator eliminates the assembly modes initiated by HED F, HED G, and HED J control operators.

10. HED Z - Allocate in Standard; Execute In High-Speed Storage.

The HED Z operator is only applicable for source programs that will operate on tape configurations. The instruction lines following HED Z operator will have addresses allocated in standard memory. The m and c addresses, however, will be allocated in high-speed memory. The address assignment is such that when a band-to-band transfer through the tape buffer is made the instructions will occupy the correct locations for minimum latency.

Before a HED Z operator is used, all of memory must be reserved except the area in which the instructions are to be executed. Besides this memory reservation, the m and c fields of the HED Z line must contain the band specifications of the standard band in

[8]When storage assigned has been exhausted, the assembler will step to the next memory level and continue to assign from that area.

which allocation is to begin and the high-speed band in which execution will take place.

For example:

OP	m	c
HED	Z2000	Δ4600

This would cause succeeding instructions to be allocated to locations beginning in band 20 for execution in band 46.

11. HED Y - Terminate HED Z Control.

The HED Y operator returns the assembly process to the normal allocation mode.

12. WDT - Word-Time Control.

The WDT Control Operator is used to modify the word-time clock; setting, resetting, and/or adding to it for the next instruction or a portion of the next instruction. The information concerning the desired modification to the word-time clock is coded in the m, c and word time fields:

OP	m	c	w/t
WDT	sssss	ΔΔxyz	iii

WDT	is the mnemonic for word-time clock control operator.
sssss	is a tag or a word-time level or an absolute address if x is S or blank.
ΔΔ	these columns are always blank.
x	is A if an increment is to be added to the word-time clock. is S if the word time clock is to be set to a particular level.
y	is A if next a address is to be modified. is M if next m address is to be modified. is C if next c address is to be modified.

z is Δ (blank) if the word-time clock is not to be reset after the action specified by x.

is R if the word-time clock is to be reset to its previous level plus normal incrementation.

iii is a three-digit numeric increment to be added to the word-time clock specified by sssss if x = S.

is a three-digit numeric increment to be added to the word-time clock <u>before</u> assigning the address specified if x = A.

Examples of WDT coding:

a. WDT SAMΔΔ ΔΔΔΔΔ 015

Would result in the word-time clock being set to the level of SAM plus 15 for the next a address.

b. WDT Δ0013 ΔΔΔSAΔ 000

Would set the a address of the next line to be assembled to level 013.

c. WDT SAMΔΔ ΔΔSAR 010

Would set the next assembled a address to the level of SAM plus 10 then reset the word-time clock to its prior setting plus normal incrementatio

d. WDT ΔΔΔΔΔ ΔΔAAΔ 015

Would add 15 to the word-time clock before assigning the next a address.

e. WDT ΔΔΔΔΔ ΔΔAMR 015

Would add 15 plus normal incrementation to the word-time clock before assigning the next m address and then reset the word-time clock to its previous setting plus normal incrementation.

f. WDT Δ1345 ΔΔSCΔ 000

Would set the word-time clock to level 145 (band relative address of 1345) before assigning the next c address.

g. WDT Δ0015 ΔΔAMΔ 025

Would add 25 plus normal incrementation to the word-time clock before assigning the next m address and 15 plus normal incrementation after assigning the next m address.

h. WDT △0006 △△AAR 007

Would add 7 to the word-time clock before assigning the next a address. The clock would then be reset to its reading before the a address assignment and 6 in addition to normal incrementation added to it before assigning the next m address.

D. TAG TABLE CONTROL OPERATORS

Specific control of the Tag Table content is provided through the use of two control operators:

1. EQU - Equivalence.

The EQU Control Operator can equate a tag to a specific value or location or clear the symbolic tag from the tag table so that the tag may be reused. It is similar, though not identical to the SYN Control Operator (see page 3-7). An EQU will not restrict a location in the memory table.

The coding of an EQU Operator is:

OP	m	c
EQU	xxxxx	yyyyy

EQU is the mnemonic for Equivalence.

xxxxx is the symbolic address to be equated to yyyyy.

yyyyy is the defined actual value or symbolic address or spaces if xxxxx is to be erased from the tag table.

Examples of EQU coding:

a. EQU CAA△△ A12△△

It is assumed that A12 has been defined in the assembly process. This EQU will cause CAA to be permanently stored in the symbol table whereas A12 will be erased by the next HED C card.

b. EQU INCRA △0002

This will relate INCRA to an increment of 0002 when used in the symbolic m field of a LIR or IIR instruction line.

c. EQU GROSS H1N21

H1N21 was previously defined by an INT entry. The processing of this EQU will equate GROSS to card word 1, numeric, Card Reader second read station.

d. EQU BED△△ △△△△△

Since the Symbolic c Field is blank, permanent tag BED is undefined (erased from the tag table) if it were previously defined and is not available for redefinition.

2. HED C - Clear Temporary Tag Table.

The HED C Control Operator clears the temporary tag table. It is usually used to mark the end of a particular section or segment of coding within the source program.

E. TAPE ASSEMBLY OPERATORS

Three operators are provided in the S-4 Assembly System for control when the input and/or output is wholly or partially on magnetic tapes.

1. HED T - Accept Tape Input.

The HED T Control Operator specifies that a certain number of entries are to be accepted from an input tape or tapes; these items must have been placed on tape by a previous assembly run, thus having "card" or item numbers assigned to them. The m field will specify the first item to be accepted from the tape. The symbolic c field will specify the last item to be accepted from the tape. After consecutively accepting and assembling the items specified (from the first through the last item) further input will be called for from the Card Reader.

All input is assumed to be from servo 3. This may be modified by placing an increment in the least significant digit of the word-time field. The increment, when added to the assumed 3 will result in a new input servo assignment. Thus, an increment of 2 would mean servo 5; an increment of 8 would mean servo 1. Care must be taken so that incrementation does not cause the assignment of servo 2. All output is assumed to be on servo 2.[9]

[9] HED T is never written out on tape or assigned an item number.

Examples of HED T coding:

```
a.  HED     T0015     Δ0035     ΔΔΔ
```

This would cause the next input source items to be read from the tape mounted on servo 3. The first item accepted will be number 15. All items from 15 through 35 will be accepted and assembled in order.

```
b.  HED     T0082     Δ0096     008
```

This would cause the acceptance of items 82 through 96 from the tape on servo 1 as source input.

```
c.  HED     T0001     Δ0022     001
```

This would cause the acceptance of items 1 through 22 from the tape mounted on servo 4 as source input.

2. HED I - Rewind Input Servo.

The HED I Control Operator will cause the input servo designated by the last HED T operator to be rewound. Initialization will be performed for a new input reel to be specified by the next HED T operator.[10]

3. HED O - Rewind Output Servo.

The HED O Control Operator will cause the output tape on servo 2 to be rewound. Before rewinding, an End-of-File Sentinel will be written on the tape and the output block counter will be reset. Initialization for a new output tape on logical servo 2 will be performed.[11]

---

[10] The HED I operator, when orignating on an input card, will not be placed on the output tape or assigned an item number.

[11] The HED O operator may originate on an input card or may be fabricated by the system as a result of a block-limit test.

F. CONSTANT LIBRARY OPERATORS

A constant library may be established for an installation using the S-4 Assembly System. This is done by assigning each constant a tag. Thus, a source program may be prepared using the specific tags for the constants that are desired. Through the use of HED L and HED K control operators, the entire constant library may be introduced as part of the source program. Only those constants whose tags have been referenced during the previous assembly of the source program coding will be used. Constants not referenced will be ignored.

1. HED L - Process Constant Library

The HED L Control Operator is used just before the constant library deck is introduced. It will cause only those constants whose tags have been referenced by the previously assembled source program to be accepted from the constant library.

2. HED K - End Constant Library Processing

The HED K Control Operator indicates that the processing of the constant library is finished. Normal assembly mode will be resumed.

G. PROGRAM TESTING OPERATORS

The S-4 Assembly System includes a number of control operators designed to aid in program testing and to permit assembly around previously allocated portions of memory.

1. HED X - Printer Output.

The HED X Control Operator eliminates the assembly output on the RPU and magnetic tape. Only Printer output continues.

To eliminate the assembly mode initiated by a HED X control operator, a HED M or HED P control operator must be used.

2. HED M - Tape Output.

The HED M Control Operator causes assembly output to be produced on magnetic tape and the Printer. If tape output is desired, the HED M Control Operator must be used.

3. HED P - Resume RPU Output (only in 80 CTC).

   The HED P Control Operator causes assembly output to be resumed on the Read-Punch Unit. Printer output continues.

4. PPA - Print and Punch Availability Table (Overlay for 80 CTC S-4 Assemblies).

   The PPA Control Operator provides a printed listing of locations not used and a punched card deck as a reloadable record of the memory availability table. The deck produced is in multiple word-per-card format (8 words per 90-column card; 7 words per 80-column card). By loading such a deck in another assembly run, the memory availability table would be restored and further assembly could be initiated without any overlay of previous assembly allocation.

   The format of the printed listing caused by a PPA Control Operator is:

```
0000 nnnnnnnnnn 2000 nnnnnnnnnn 4000 nnnnnnnnnn etc.
0001 nnnnnnnnnn 2001 nnnnnnnnnn 4001 nnnnnnnnnn
0002 nnnnnnnnnn 2002 nnnnnnnnnn 4002 nnnnnnnnnn
  "      "        "      "        "      "
  "      "        "      "        "      "
  "      "        "      "        "      "
0199 nnnnnnnnnn 2199 nnnnnnnnnn 4199 nnnnnnnnnn etc.
```

   The first n after 0000 refers to location 0000; the second n after 0000 refers to location 0200; the third n to 0400; the fourth n to 0600; the fifth n to 0800; and so on for the balance of the printout.

   Each n will contain a utilization key the interpretation of which is:

   1 for an unused location.

   0 for an allocated location or for a location not available for the object program computer.

5. PAT - Print Availability Table (Overlay for 80 CTC S-4 Assemblers).

The PAT Control Operator will cause the Printer listing of the Memory Availability Table. No punched card deck will be produced.

6. SYP - Print Symbol Table (Overlay for 80 CTC S-4 Assemblers).

The contents of the Symbol Table may be listed on the Printer at any time during the S-4 Assembly by use of an SYP Control Operator. Each line of the listing consists of five symbol sets each of which is in the following format:

```
ttttt   f   aaaa
```

ttttt is the tag mnemonic.

f is the utilization key (the interpretation is the same as for the PPA utilization keys). It is 0 in 80 CTC and 90 TC.

aaaa is the address allocated to the tag.

When aaaa appears as a three-digit numeric quantity (Δnnn) a B core address has been assigned; when a three-digit number is preceded by a semi-colon (;nnn) a G core address has been assigned.

A. CONFIGURATION ADAPTABILITY

By altering two locations within the 80 CTC or 90 TC assembly program, it is possible to assemble for a minimum STEP system, a USS I or a USS II. The patterns are shown below for the two erase patterns which govern memory allocation.

	Bit											
Location 3502	5	60	62	64	66	68	70	72	74	76	78	
	4	40	42	44	46	48	50	52	54	56	58	
	2	20	22	24	26	28	30	32	34	36	38	
	1	00	02	04	06	08	10	12	14	16	18	
Location 3503	5	*	*	*	*	*	*	*	*	*	*	
	4	*	*	*	*	*	*	*	*	*	*	
	2	00	02	04	06	08	10	12	*	*	*	(CORE LOCATIONS)
	1	80	82	84	86	88	90	*	*	*	*	

There should be a bit present in each position corresponding to a band present on the system for which assembly is desired. The "2" level of location 3503 applies to USS II Core Memory locations. Wherever there is an asterisk in word 3503 there must be a zero bit present.

B. ASSEMBLY OUTPUT

1. Punched-Card Output

The output deck of an S-4 Assembly is the machine coded object program in a one-instruction-per-card format plus the symbolic coding and remarks. This format is acceptable to the standard loading routine of the object computer.

2. Printer Output

The S-4 Assembly Printer output is a side-by-side listing of the assembled and the symbolically coded lines (including the Word-Time and Remarks Field) and codes to indicate error detected during the assembly processing. The latter, when necessary, are printed on the extreme right of the listing.

3. Printer Error Codes

The error codes that may appear on the Printer listing are:

Code	Error
Δ (Blank)	No error detected.
1	Region or interlace not defined.
2	A non USS form of machine (absolute) address was encountered.
3	An input error in blank-address linkage was discovered.
4	Incorrect Class or Symbolic Op code was used.
5	This line was bypassed because of an error condition. Error condition was one of the following: 1. SYN operator - a. c address symbol undefined. b. an address is unavailable. 2. EQU operator - Spaces in m and c. 3. HED operator - Invalid HED designation. 4. INT operator - Invalid INT designation. 5. BLR and BLA operators - Invalid address in m. 6. REG operator - a. Invalid address in m. b. Invalid REG designation.

Code	Error
6	Previously defined symbolic address.
7	SYP, PAT, and PPA operators - In 80 CTC when Forward Search coding has overlaid coding for these operators. Two consecutive locations are not available for assignment to an overflow (c+1) set.
8	Memory full indicator. May be result of one of the following: 1. All memory was depleted and no assignment could be made. 2. Memory area step down has occurred. The area being searched (high-speed or core) as a result of a tag designator or a HED operator (F, J, or G) has been depleted and assignment has been made from the next memory area. (fast or high-speed) 3. A band-relative address required by a SYN operator was unavailable. The next best address level was assigned.
9	The symbol table is full and a symbolic tag could not be stored in the table nor assigned an address.

If several errors appear on a line they will be noted in the same order as detected (class, OP, a, m, c).

For example, if both the class and OP were erroneous and the m address was a defined regional reference, 4, 4, 1 would appear as the error note.

C. PREPARATION FOR ASSEMBLY

After the program has been coded it is punched, one instruction line per card. The resultant deck is the input to the S-4 Assembly program. The sequence of the input cards is:

a. RST Card: The RST card contains the name of the program in the m and c field of the card.

b. Specials: If desired, PPA cards may be entered at this point, in order to restore the memory allocation as it was at the time of punching the PPA cards so that further assembly will be an areas other than used in the prior program. In addition cards controlling package routine changes, interlace changes and so on should be entered at this point.

c. BLR: Block Reservation cards.

d. REGion: Regional Reservation cards.

e. INTerlace: Interlace Reservation cards.

f. SYNonym: Synonym Cards.

g. EQUivalence: Equivalence Cards.

h. HED: HED cards as needed prior to initiation of main program detail card assembly.

i. Detail: Main program detail cards plus Control Operator cards as required.

j. HED L: If the constant assembly option is utilized, a HED L card followed by the constant pool followed by a HED K card is required at this point.

k. SYP: Symbol Table printout (recommended but not required).

l. PPA/PAT: Print-Availability Table or Print and Punch-Availability Table.

m. HED O: Output tape rewind if tape output.

n. HED I: Input tape rewind if tape input.

o. END: To create PTA sentinel card.

p. RESet: If further compilation to reinitialize.

D. THE FORWARD SEARCH OPERATION

The Forward Search function of S-4 is primarily intended to optimize latency where a branching operation is involved. This function enables the assembler to scan up to 10 lines ahead when a previously undefined symbolic address is encountered and then make assignments in a backward direction.

In 80 CTC the Forward Search coding overlays that portion of the program that contains the coding for SYP, PAT, and PPA. Therefore, to utilize Forward Search on the 80 system, the special F. S. Overlay must be placed in memory. Each card of this deck contains a loading sentinel which indicates to S-4 that the data on the card is not to be processed, but rather to be loaded directly onto the drum. From this point on, SYP, PAT, and PPA operators will be bypassed with a note 5 on the printer listing unless the SYP-PAT-PPA coding (overlay deck) is restored. With the latter overlay in memory, a HED B card (See below) is simply ignored.

The 90 TC program does not require the use of overlays; both the coding for Forward Search and SYP and PAT are contained in the main program.

With Forward Search in memory, a HED B control operator will initiate the function. HED B will set a switch that causes S-4 to enter Phase 1 (Forward Scanning) of Forward Search when the first undefined symbolic Tag (LRP not included) is encountered. The data on this card and on subsequent cards will be stored in a special table that will be accessed when Backward Assignment (Phase 2) commences.

Forward Scanning will normally continue for up to 10 lines. However, if certain conditions are encountered, the assembler will enter Phase 2 prior to encountering this limit. Further, depending upon the condition, Backward assignment will commence either with the line in which the condition occurs or with the line preceding. Upon execution of Phase 2, the normal processing mode resumes, but only until another undefined symbolic tag is encountered, at which point the Forward Search cycle will be repeated.

If it is desired to permanently eliminate the Forward Search mode, a HED A card must be introduced. If, at this point, Phase 1 is in effect, Phase 2 of the current cycle will be executed before termination.

The conditions that suspend Phase 1 of Forward Search are as follows:

1. Where backward assignment will begin with the preceding line:

    a. A constant has been encountered.

    b. The card contains a "C" in the class field.

    c. Any pseudo operator.

    d. Any LRP if the suffix is:

        (1) blank.

        (2) H and the A address did not contain this LRP.

2. Where backward assignment will begin with the same line:

    a. The 10th line has been scanned.

The "C" class field comments cards is used as a control for Forward Search processing. Those sections of coding containing branches to a common exit point should be preceded and followed by "C" cards. The HED A and HED B are also required to initiate and terminate the function.

E. LISTER OVERLAY (80 Only)

The lister overlay modifies the assembly process so that a copy of an S-4 tape or card program and a printer listing can be produced when the object program is introduced as input to the system. The assembler input and output areas are linked with minor parts of the processing areas to provide the listing function.

As the memory availability table is updated during listing, all control operators which affect the memory table will operate. All m addresses encountered in the assembled output fields which have a tag form in the corresponding symbolic fields will be made unavailable. All a addresses will be restricted. A PAT or PPA, introduced after the completion of the listing function, will accurately reflect the condition of the memory table.

In addition to copying a program, the lister, if desired, will produce only the Printer listing. Merging patches to a previously assembled program can be accomplished while the listing is being produced. These functions can be performed by employing the S-4 control operators: HED X, HED T and HED M.

Special operating procedures for use with the lister overlay feature are provided at the end of the OPERATING PROCEDURES section of this manual.

F. TAPE INTERLACES

Tape interlaces defined by INT control cards are set up according to the hardware interlace format. Consequently, when these interlaces are addressed the interlace position produced by the assembler will be standard hardware position (numeric = BB00 + 001 + 58 N (Mod 200); zone = numeric + 5).

Those data tape interlaces which are arbitrarily defined by the programmer in accordance with a data item layout required by software packages or own programming design, should be defined and addressed in a different manner. The control operators, BLR, REG, and EQU are most appropriate for this purpose. The BLR restricts the necessary locations, REG sets up the required item layout as a region, and EQU defines the inter-item span and the span to the last item (for end-of-block testing).

The recommended item formats are different for the USS I and USS II. However, in each case a region is defined so that the first location of the region will refer to the first word of the first item, the second to the second word of the first item, and so forth. Words of subsequent items are referenced by IR incrementations as defined by the EQU card.

USS I Example

A data interlace of twenty-five 4-word items in band 14 is defined by the following:

```
BLR         1400        1599
REG         U1401       ΔΔΔ␣␣       250
REG         P1406       Δ␣␣Δ␣       250
EQU         UINCR       0002
EQU         ULAST       0050
```

(Testing for end-of-block is accomplished by)

```
LDL
 00         ULAST       0000
IIR         UINCR
TEQ          FINI       CONT.
```

A data interlace of eight 12-word items in band 42 is defined by the following:

```
BLR         4200        4399
REG         N4201                    216
REG         Z4206                    216
EQU         NINCR       0002
EQU         NLAST       0016
```

USS II Example

A data interlace of eight 12-word items in core storage is defined by the following:

```
BLR          B201       B400
REG         NB201                    002
REG         ZB202                    002
EQU         NINCR       0024
EQU         NLAST       0192
```

V. OPERATING PROCEDURES

A. LOADING S-4 ASSEMBLY

1. Read-Punch Unit

   a. Fill input hopper with blank cards.

   b. Depress FEED ONE CARD button three times.

   c. Depress RESET button.

2. Printer

   a. Determine adequacy of paper supply.

   b. Advance paper until six holes are above sprocket.

3. Tape Units

   a. Mount scratch tape on servo number 2.

   b. Connect servo logically to output 2.

   c. Depress READY button.

4. Console and Card Reader (USS-80 Only) [12]

   a. Place S-4 self-loading deck in Reader input stacker.

   b. Depress GENERAL CLEAR.

   c. Depress ONE INSTRUCTION button.

   d. Key 72 0000 0000 into register C.

   e. Depress RUN button (Reader should feed one card).

   f. Key 96 0001 0118 into register C.

   g. Depress CONTINUOUS and RUN buttons.

      1. If loading halts on a 67 0H9F 0H6H or 67 0H99 0H1H, a read-check during preload has failed. Correct Reader trouble and restart at 4a. above.

[12] The USS 90 version is being released as a tape load program in standard scatter format. Operating instructions will be the same as those applicable for any MASCOT-generated locator and load blocks.

2. If loading halts on a 67 0xxx 0022 it indicates that a card is missing or that the read-check was not successful. 0xxx is the card sequence number which is punched in card code in columns 1-4 and in machine code in the duoprimed portion of columns 73-76. Check the last 5 cards in stacker 1. Correct their sequence if necessary. Re-insert cards beginning at xxx in Reader. Depress GENERAL CLEAR and RUN buttons.

h. Successful Load: When the computer stops on 67 HHHH 000H, the sentinel card has been detected and a successful load is indicated. Register X contains the starting address of the S-4 program.

i. Parameter changes: Certain parameters can be changed at the discretion of the user. The sentinel card, number 523, contains the three constants which control memory size and the output tape block limit.

1. Memory size is controlled by locations 3502 and 3503. Memory is set to full 8,800 words plus core.

2. Output limit is determined by the contents of location 4001 (00 0000 nnnn).

B. EXECUTING S-4 ASSEMBLY

1. Place program deck in the input magazine of the Card Reader.

2. Release 96 button.

3. Depress GENERAL CLEAR and RUN buttons.

C. ASSEMBLY STOPS AND CORRECTIVE ACTION

67 0111

Reader malfunction (c+1):
Take necessary remedial action; depress GENERAL CLEAR and RUN buttons.

67 0222

Printer malfunction (c+1):
Correct paper condition; depress GENERAL CLEAR and RUN buttons.

67 3300

Error on tape write instruction:
Correct error condition; depress GENERAL CLEAR and RUN buttons. If same stop occurs begin new assembly.

67 33AA

Correctable error on tape read instruction. The instruction has been executed at normal, low, and high gain with no success.

To skip over one or more blocks, create from input tape listing one or more cards and a HED T card reflecting new tape input position. Insert cards in HSR input stacker, depress "m", GENERAL CLEAR and RUN buttons.

67 33BB

c+1 on tape buffer unload instruction. To re-execute depress "c", CLEAR and RUN buttons. If several attempts are unsuccessful and/or machine error cannot be corrected, manually read input servo backward 1 block and go to location 4438 to re-execute read tape commands.

67 33GG

Output block counter has reached limit as defined in location 4001:
To write sentinel block on output tape, clear block counter and rewind, depress "c" and RUN buttons. Mount new output tape and if input is from tape, create a HED T card to define next series of tape blocks to be assembled.

To override output block limit, depress "m" and RUN buttons.

67 33HH

Uncorrectable error on tape read instruction:

If error is a result of servo "off" condition or synchro-servo mis-wiring, and/or tape is in first block condition, correct error condition and begin assembly again.

If error is a result of a tape misread, error-recovery procedure is as listed under stop 67 33AA. To reach this stop, depress GENERAL CLEAR and RUN buttons.

67 0444 (Only in 80 CTC)

Punch malfunction (c+1):
If card jam occurs, remove cards from output hopper. Clean out punch, reposition cards, depress ONE CARD three times, depress RESET, GENERAL CLEAR, and RUN buttons.

If card is not read, reposition cards in input hopper after remove damaged card, depress ONE CARD three times, RESET, GENERAL CLEAR and RUN buttons.

67 0888 (Only in 80 CTC)

Malfunction in RPU:
Six attempts have been made to punch read check card with no success. Repair punch unit. To omit bad card and continue, depress RUN button (manually create lost card from printer listing).

67 0999 (Only in 80 CTC)

An END operator has been detected and the sentinel has been punched. A card output assembly is now completed. To begin new assembly depress RUN button.

D. LOADING OBJECT PROGRAM FROM S-4 OUTPUT TAPE

1. Mount tape on servo 2.

2. Key into rC G2 0200 000A
   rA F6 0000 0029

3. Depress CONTINUOUS, GENERAL CLEAR, and RUN buttons. Location 0029 is entrance to memory fill section which fills memory to 67 nnnn 0145 stop orders.

4. If program is on two or more tapes load additional tapes with:

   G2 0200 000A
   F6 0000 0045

Location 0045 is entrance to load tape section.

5. Tape Load Stop

67 BB33 0045	An attempt has been made to correct a correctable read without success. To continue, depress RUN button. The next read should halt at out-of-sequence stop.
67 HH33 0070	A non-correctable read has been encountered. Examine error flip flops and correct mechanical malfunction, if any. Check contents of 0070 for type of read. If backward read, execute G2 0200 0045 from register C. If a forward read, execute 00 0045 0045 and run on continuous. The next halt should read at out-of-sequence stop.
67 0555 0043	Out-of-sequence stop. Check contents of 0003 for number of block now being loaded. Subtract one for number of block not successfully loaded. To continue depress "c" and RUN button. Manually load lost instruction from printer listing after tape loading is completed.
67 0666 0189	The sentinel block has been encountered. Depress RUN button to rewind tape.
90 0161 0007	An attempt has been made to load a sign-digit key. USS Model 1 does not have this capability. Check block number in location 0003. Execute 00 0007 0007 and run. When tape load is completed, check printer listing, correct instruction, and load manually.

If tape is not moving, push one instruction button, depress GENERAL CLEAR, and execute a 00 0070 0070.

E. LOADING S-4 OUTPUT CARDS

1. On USS Model II

Use PTA21-8T00 (Core)

2. On USS Model I

Use PTAS1-8C00

F. LISTER OVERLAY

1. To produce a Printer listing and updated card deck or tape;

   a. Use HED X if only the Printer listing is desired and HED M if output is to be on tape. HED specification is not required if output is to be punched card.

   b. Insert change cards in the input deck and place on top of lister overlay in card reader input stacker. If input is on tape, merge the change cards with HED T control operators.

      Change cards should contain both the machine form and the symbolic form of the instruction, i.e., they have been produced by an assembly or fabricated to appear the same as assembly output. Control operators must contain zero punches in columns 23 through 26.

   c. Key into register C 00 3500 3500, press CONTINUOUS and RUN buttons. Stop and restart points are the same as those used during the normal assembly process.[13]

2. To restart the normal assembly process after listing to a specific point;

   a. Prepare SYN control operator cards for all defined tags which will be encountered in the segment to be assembled.

   b. List the program up to the assembly restart point as outlined in (1) above.

   c. When restart point is reached, restore the SYP, PAT, PPA or Forward Search overlay.

   d. Introduce the segment to be assembled preceded by the SYN cards.

   e. Continue with normal assembly.

3. To obtain a PPA of a program for restricting purposes;

   a. If a new deck or tape is not desired, a HED X should be used.

   b. Introduce the program deck in PTA format. Control operators which restrict required memory areas, e.g., INT, BLR, and REG, should precede the input dec

[13]As the lister function cannot fabricate a block sentinel when a HED 0 operator is encountered, it is necessary to restore the system to normal processing to obtain this feature.

Temporary storage locations referenced only in the "m" address should be restricted with control operators. All "a" addresses will be restricted by the lister.

c. At completion of the listing function replace the SYP, PAT, PPA processing and insert a PPA card.

4. Lister Options;

As during the normal assembly process the lister begins the output page and item count at 1. If a page and/or item number other than 1 is desired, it may be set before the listing function begins.

To change either, it is necessary to initialize the function, i.e., run on 00 3500 3500 to load the overlay, without any data or control cards in the Reader. When the stop 67 0111 4570 is encountered, the changes listed below are made. Data cards are then placed in the Reader and program control is returned to 4570.

a. If a specific card or block number, other than 1, is desired on the output list, key the number minus 1 into location 4486.

b. If a specific page number on the list is desired, key the page number minus 1 into the LSD portion of locations 4487 and 4815, and zeros into location 4036. Space the paper 6 holes above the next page.

A P P E N D I X   A - C O D I N G   F O R M S

The two coding forms shown on the following pages can be used for any S-4 80C or 80/90 TC Assembly.

Remington Rand Univac
DIVISION OF SPERRY RAND CORPORATION

S4 ASSEMBLY FORM
UNIVAC SOLID-STATE SYMBOLIC SYSTEM

80

PROGRAM: DATE: PAGE: OF

A-2

L I N E	LINE NO.	SYMBOLIC "A"	OPERATION		IR	SYMBOLIC "M"	SYMBOLIC "C"	REMARKS:	
			CL	Symbolic				WORD TIME	
	12 15	41 42 43 44 45	46	47 48 49	50	51 52 53 54 55	56 57 58 59 60	61 62 63	64 65 70 75 80
1									
2									
3									
4									
5									
6									
7									
8									
9									
10									
11									
12									
13									
14									
15									
16									
17									
18									
19									
20									
21									
22									
23									
24									

UP 1774.6 Rev. 1

Remington Rand Univac
DIVISION OF SPERRY RAND CORPORATION

ASSEMBLY FORM
UNIVAC SOLID-STATE SYMBOLIC SYSTEM

90

PROGRAM: ________ DATE: ________ PAGE: ____ OF ____

LINE	LINE NO.	SYMBOLIC "A"	OPERATION		IR	SYMBOLIC "M"	SYMBOLIC "C"	REMARKS	
			CL	Symbolic				WORD TIME	
	13 14 15 16	46 47 48 49 50	51	52 53 54	55	56 57 58 59 60	61 62 63 64 65	66 67 68	69 70 75 80 85 90
1									
2									
3									
4									
5									
6									
7									
8									
9									
10									
11									
12									
13									
14									
15									
16									
17									
18									
19									
20									
21									
22									
23									
24									
25									

UT 2521

S-4 ASSEMBLER	OP CD	m	c	SIGN	DESCRIPTION	WORD TIMES
ARITHMETIC						
ADD	70	m	c		(m) + (rA) → rA, if Overflow c=c + 1.	5
SUB	75	m	c		(rA) - (m) → rA, if Overflow c=c + 1.	5
MUL	85	m	c		(rL) × (m) → rA MSD, rX LSD.	5 + ND + SD
DIV	55	m	c		(m) ÷ (rL) → rA Quot, rX Rem. if Overflow c=c + 1.	20 + SOD + STCED
TRANSFER						
LDA	25	m	c		(m) → rA	4
LDX	05	m	c		(m) → rX	4
LSX	05	m	c	4	(1 & 2 Bits of m) → Sub-Registers 1 & 2 and Sign rX unchanged.	4
LDL	30	m	c		(m) → rL	4
STA	60	m	c		(rA) → m; m may not be a register	4
STX	65	m	c		(rX) → m; m may not be a register	4
SSX	65	m	c	4	(Sub-Reg. 3 & 4 of rX) → 1 & 2 Bits of m; Zero Bits 3 & 4 of m & Sign +. may not be a Register.	4
STL	50	m	c		(rL) → m; m may not be a register	4
SML	90	m	c		MSD of m → Sign of rL; Balance Unchanged	4
TDC	B0	m	c		(DRUM) → Core	4 + N
TCD	B8	m	c		(CORE) → DRUM	4 + N
SMA	F0	m	c		Sign of m → MSD of rA; Balance Unchanged	4
ATL	77	-	c		(rA) → rL	3
CT4	23	m	-		(rC) → rA	3
CLX	06	m	c^2		Zeros → rX; sign +	3
CLA	26	m	c^2		Zeros → rA; sign +	3
ZSX	06	m	c^2	4	Zeros → Sub-Registers 3 & 4 of rX; sign +; balance rX unchanged.	3
CLL	31	m	c^2		Zeros → rL; sign +	3
CAA	36	m	c^2		Zeros → rA; retain original sign	3
LOGICAL						
CAX	86	m	-		Zeros → rA, rX; sign of rL → rA, rX	14
BUF	20	m	c		Superimpose (m) on (rA) → rA	4
ERS	35	m	c		Extract (m) from (rA) → rA	4
SHR	32	0n00	c		Shift right n places. (rA) → (rX) → rA	3 + n
SHL	37	0n00	c		Shift left n places. Zeros → rA LSD	3 + n
ZUP	62	-	c		Suppress Zeros, Commas, Mc-6 in rA, rX.	4
JMP	00	m	-		Skip	2
HLT	67	-	-		Stop	-

S-4 ASSEMBLER	OP CD	m	c	SIGN	DESCRIPTION	WORD TIMES
COMPARISON						
TEQ	82	=	≠		(rA) : (rL)	3
TEA	82	=	≠	4	(rA) & (Sub-Registers 1 & 2 of rX) : (rL) & (Sub-Registers 3 & 4 of rX)	3
TGR	87	>	≤		(rA) : (rL)	3
TGA	87	>	≤	4	(rA) & (Sub-Registers 1 & 2 of rX) : (rL) & (Sub-Registers 3 & 4 of rX)	3
TRANSLATE						
CTM	12	-	c		CC (rA, rL, rX) → MC-6 (rA, rX). Zeros - rL	3
MTC	17	-	c		MC-6 (rA, rX) → CC (rA, rL, rX)	3
MTX	C1	-	c		MC-4 (rA) → XS-3 (rA)	3
XTM	C3	-	c		XS-3 (rA) → MC-4 (rA)	3
INDEX REGISTER						
LIR	02	m	c	*	m of instruction word → IRi	3
IIR	07	m	c	*	m of instruction word + (IRi) → IRi, and m of rA. Zeros → balance of rA	4
PRINTER						
PRN[3]	11	bbnn	c		Advance nn lines, print bb band. (rA), (rX) destroyed L197 : N1 189	592
PFD[3]	16	00nn	c		Advance nn lines	4
PBT	27	Yes	No		Printer Test: Yes (rC) → rA	No=3 Yes=4
CARD READER						
HBT	42	Yes	No		Buffer Test: Yes (rC) → rA	No=3 Yes=4
HBU	96	bb00	c		(B) → J interlace on bb and: L 198; NI 001	203
HBU	96	bb01	c		(B) → MC-6 → J_T interlace on bb band. L198; NI 013	215
HCC[3]	72	m	c		Card Cycle. Interlock (rC) → rA. NI → m	4 if c; 4 if
HSS	47	0n00	c		Select Stacker n (n = 0, 1, 2)	3
READ-PUNCH						
RBT	22	Yes	No		Buffer Test: Yes, (rC) → rA	No=3 Yes=4
RBU	46	bboo	c		(B) → Ir interlace on bb band. L 098; NI 101	203
RBU	46	bb01	c		(B) → MC-6 → Ir interlace on bb band. L 098; NI 113	215
RCC[3]	81	bb00	c		Card Cycle. 0 interlace on bb band → B. L 098; NI 001	103
RCC[3]	81	bb01	c		Card Cycle. MC-6 in Or interlace on bb band → CC → B. L 098; N1 108	210
RSS	57	-	c		Select Stacker 1 (sort)	3
MAGNETIC TAPE						
TST	C2	Yes	No		Synchronizer Test: Yes, (rC) → rA	No=3 Yes=4
TBL	C6	m[4]	c		Tape Interlace on m band → B. L 048; NI 053	205
TBT	C7	Yes	No		Buffer Test: Yes, (rC) → rA, Error FF → rL	No=3 Yes=5
TRW	F2	Oxy0	c		Rewind UNISERVO X. (X=0-9) y=0, no interlock. y=2, interlock	600 ms.
TBU[3]	F6	m[4]	c		(B) → Tape Interlace on n band. L198; NI 003. If AOT (rC) → rA, NI in C+1.	205
TRD	G2	Oxyz	c		Read 1 blk. from tape → B	17
TWR	H2	Oxy0	c		Write 1 blk. from B → Tape	17

S-4 ASSEMBLER	OP CD	m	c	SIGN	DESCRIPTION	WORD TIMES
RANDEX						
LSR	40	m	c		(m) → SIR	4
DPT	43	OnOO	c		Test Unit N; if head in position set H.P.F.F.[5]	No=3 Yes=4
DBT	92	Yes	No		Test H.P.F.F.[5]; if set (rC) → rA; N1 → m	3
TBU[3]	F6	m[4]	c		(B) → Tape Interlace on n band. L 198; N1 003. If AOR (rC) → rA, N1 in C+1.	205
TBL	C6	m[4]	c		Tape Interlace on m band. B. L048; N1 053	205
TBT	C7	Yes	No		Buffer Test: Yes, (rC) → rA, Error FF → rL	No=3 Yes=4
TST	C2	Yes	No		Synchronizer Test: Yes, (rC) → rA	No=3 Yes=4
PDH	18	OUDSSTTB[6]			Position Read-Write Head. 125-550 ms.	No Processor Time Used
DWT	28	OUDSSTTB[6]			Write RANDEX Blk (B) → Blk. specified	
DRD	38	OUDSSTTB[6]			Read RANDEX Blk. Blk specified → B	
DWC	48	OUDSSTTB[6]			Write/Check RANDEX Blk. (B) → Blk, specified and check.	
DSW	58	OUDSSTTB[6]			Search Write. (B) → Blk. identified by Search.	
DSR	68	OUDSSTTB[6]			Search Read. (Blk.) identified by Search → B	
PAPER TAPE						
RPT	A1	m	c		Read paper tape. If interlock rC → rA; NI at m.	3 if c; 4 if m
PBU	A2	0000	c		(B) → rA & rX. Numeric → rA; Zone → rX. If parity error c=c + 1.	3
TTR	A3	m	c		Input Buffer-Loaded Test: Yes, (rC) → rA. NI at m	3 if c; 4 if m
PPT	A7	-	c		(rA) & (rX) → B: Initiate output punching.	3
TPB	A8	m	c		Output Buffer Free: Yes, (rC) → rA. NI at m.	3 if c; 4 if m

[1]Add 1 Word Time to instructions employing IR modification.

[2]If the next instruction is to be found in core, then the "m" & "c" must be the same address. The 26 instruction may not be index register modified.

[3]If not executed, (rC) → rA, next instruction → c+1.

[4]m=bb00 if drum: where bb is band address
m=BXXX if core: where BXXX is beginning word address in core.

[5]H.P.F.F. = Head Position Flip Flop.

[5]Instruction executed in SIR. O=unused digit position; U=RANDEX Unit D=Drum Half; SS=Sector; T=Track; B=Block.

S 4 ASSEMBLY

UNIVAC

DIVISION OF SPERRY RAND CORPORATION

UP-1774.6 Rev. 1