

**UNIVAC<sup>®</sup>**  
**SOLID-STATE**  
**90**

Off-Line Program Testing



*Remington Rand Univac*  
DIVISION OF SPERRY RANK CORPORATION

© 1968 - SPERRY RANK CORPORATION

## CONTENTS

PURPOSE. . . . .	1
CONVENTIONS. . . . .	1
PROCEDURES . . . . .	5
Verify Key Punching. . . . .	5
Verify Legality of "a" Addresses In All Program Cards. . . . .	6
Verify Legality of Operation Codes. . . . .	6
Verify Legality of "m" Addresses In Instruction Cards. . . . .	7
Verify Legality of "c" Addresses In Instruction Cards. . . . .	8
Verify Continuity of "c" Address to "a" Address. . . . .	9
Verify Insertion of 22 Tests of HSR Buffer . . . . .	9
CONSIDERATIONS . . . . .	10

## I. PURPOSE

- A. To suggest methods of analyzing the clerical accuracy of machine coded problems for the UNIVAC Solid-State 90 using conventional Punched-Card equipment prior to program testing on the computer.
- B. To establish standards to aid in program testing at the computer and to assure efficient use of time at program testing facilities.

## II. CONVENTIONS

- A. To aid in developing a uniform testing procedure, the illustrated Coding Chart (U1711), Memory Chart (U1710) and Program Card form are recommended.

Coding Chart		Card Columns	Purpose
Field	Name		
A	Page No.	11 - 13	A sequential number to identify the page of coding.
B	Line No.	14 - 15	To identify the line of coding within the page.
C	Suffix	16	Optional field. May be used to designate an insertion.
D	Instruction Location (a)	17 - 20	Designates the memory location where the instruction/constant is to be stored.
E	Operation Code	21 - 22	Designates the machine operation to be performed. (2 MSD of a constant)
F	Data Location (m)	23 - 26	Designates the memory location of the data to be handled. (4 next MSD of a constant)
G	Next Instruction (c)	27 - 30	Designates the memory location of the next instruction to be executed. (4 LSD of a constant)
H.	Key	31	Designates how the instruction/constant is to be stored.

**Univac<sup>®</sup> SOLID-STATE COMPUTER - CODING CHART**

CARD	ROUTINE
III	IV

APPLICATION:           I           PROGRAM:           II           PAGE:                     

REFERENCE			INSTRUCTION				KEY	NOTES			TIME
SEQUENCE		N	LOCATION	WORD				rA	rX	rL	
PAGE	LINE		a	Op	m	c					
VI		1									
	VII	2									
		3									
		4									
		5									
		6									
		7									
		8									
		9									
		10									
		11									
		12									
		13									
		14									
		15									
		16									
		17									
		18									
		19									
		20									

I. THE NAME OF THE APPLICATION

II. THE NAME OF THE PROGRAM

III. CARD NUMBER (for use in assembly systems)

IV. ROUTINE DESIGNATION (for use in assembly systems)

V. SEQUENCE NUMBER (for use in assembly systems and for those who prefer to use a straight sequence number rather than a page and line reference)

VI. PAGE NUMBER (used to designate the page where reference is made to this instruction)

VII. LINE NUMBER (used with Page Number to designate a specific line on which reference is made to this instruction)

CARD	ROUTINE

APPLICATION: \_\_\_\_\_ PROGRAM: \_\_\_\_\_ PAGE: **A**

REFERENCE		INSTRUCTION				KEY	NOTES				TIME
SEQUENCE	PAGE LINE	LOCATION	WORD				TA	IX	IL		
		a	Op	m	c						
1		<b>D</b>	<b>E</b>	<b>F</b>	<b>G</b>	<b>H</b>					
2											
3											
4											
5											
6											
7											
8											
9											
10											
11											
12											
13											
14											
15											
16											
17											
18											
19											
20											

  

Printed in U. S. A. REMINGTON RAND	12	IDENTIFICATION	ROUTINE	PAGE	LINE	INST. LOC.	OP.	DATA LOC.	NEXT INST.	K	12																																																																																
	34	REFERENCE		INSTRUCTION							34																																																																																
	56						WORD			<b>H</b>	56																																																																																
	78			<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>	<b>F</b>	<b>G</b>		78																																																																																
						a		m	c																																																																																		
		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	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	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	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90
		<b>UNIVAC SOLID-STATE 90</b>																																																																																									
		<b>PROGRAM CARD</b>																																																																																									

  

Note:

Some programmers may prefer to use a straight Sequence number rather than a Page and Line reference. In this event the actual Page (A) and preprinted Line (B) numbers are ignored and the Sequence number is written in the reference area under the captions Page and Line. Sequence number is punched in card columns 11 through 15.

**KEY:** 1. TRANSLATE AND STORE + 2. TRANSLATE AND STORE - 3. STORE UNPRIMED + 4. STORE UNPRIMED - 5. STORE PRIMED+ 6. STORE PRIMED -

# NEW UNIVAC<sup>®</sup> MAGNETIC AMPLIFIER, SOLID-STATE COMPUTER - MEMORY CHART

CUSTOMER: \_\_\_\_\_ PROGRAM: \_\_\_\_\_

APPLICATION: \_\_\_\_\_ PAGE: \_\_\_\_\_

BAND:			
__00 TO __99		__00 TO __99	
00	50	00	50
01	51	01	51
02	52	02	52
03	53	03	53
04	54	04	54
05	55	05	55
06	56	06	56
07	57	07	57
08	58	08	58
09	59	09	59
10	60	10	60
11	61	11	61
12	62	12	62
13	63	13	63
14	64	14	64
15	65		
16	66		
17	67		
18	68		
19	69		
20	70		
21	71		
22	72		
23	73	23	73
24	74	24	74
25	75	25	75
26	76	26	76
27	77	27	77
28	78	28	78
29	79	29	79
30	80	30	80
31	81	31	81
32	82	32	82
33	83	33	83
34	84	34	84
35	85	35	85
36	86	36	86
37	87	37	87
38	88	38	88
39	89	39	89
40	90	40	90
41	91	41	91
42	92	42	92
43	93	43	93
44	94	44	94
45	95	45	95
46	96	46	96
47	97	47	97
48	98	48	98
49	99	49	99

BAND:			
__00 TO __99		__00 TO __99	
00	50	00	50
01	51	01	51
02	52	02	52
03	53	03	53
04	54	04	54
05	55	05	55
06	56	06	56
07	57	07	57
08	58	08	58
09	59	09	59
10	60	10	60
11	61	11	61
12	62	12	62
13	63	13	63
14	64	14	64
15	65		
16	66		
17	67		
18	68		
19	69		
20	70		
21	71		
22	72		
23	73	23	73
24	74	24	74
25	75	25	75
26	76	26	76
27	77	27	77
28	78	28	78
29	79	29	79
30	80	30	80
31	81	31	81
32	82	32	82
33	83	33	83
34	84	34	84
35	85	35	85
36	86	36	86
37	87	37	87
38	88	38	88
39	89	39	89
40	90	40	90
41	91	41	91
42	92	42	92
43	93	43	93
44	94	44	94
45	95	45	95
46	96	46	96
47	97	47	97
48	98	48	98
49	99	49	99

Each time an Instruction or Constant is assigned a memory location write the reference, either Page/Line or Sequence number, opposite the word designation on the memory chart. This will eliminate the possibility of assigning a memory location to more than one instruction/constant and aid the programmer in selecting unused locations.

- B. To facilitate automatic sorting of instruction and constant words it is suggested that blocks of Page and Line or Sequence numbers be assigned as follows:
  - 1. A major block for all instruction words.
  - 2. A minor block for all constants which are instruction words.
  - 3. A second minor block for all constants which are not instruction words.
- C. Consideration should be given to the advantages of punching a one (1) in some unique position of all program cards to establish a card count control.
- D. To aid in the automatic analysis it will be helpful to prepare and verify Program Testing Decks. (Note that Band 1, words 0000 through 0199, is presently reserved for control routines) Three program testing decks will be required:
  - 1. PT-1 - 4800 cards with memory locations 0200 through 4999 punched in the "a" location, card columns 17 - 20.
  - 2. PT-2 - 4800 cards with memory locations 0200 through 4999 punched in the "m" location, card columns 23 - 26.
  - 3. PT-3 - 4800 cards with memory locations 0200 through 4999 punched in the "c" location, card columns 27 - 30.

### III. PROCEDURES

#### A. Verify Key Punching

- 1. List all the Program Cards on the tabulator in the same order as the coding appears on the coding charts. Set sub-total control on Page number (columns 11 - 13) to obtain space between pages.
- 2. Verify to original coding charts. Save this list and manually post all corrections made as a result of subsequent analysis.

B. Verify legality of "a" addresses in all program cards.

1. Sort all program cards, instructions/constants, to "a" address, card columns 17 - 20.
2. Merge program testing deck # 1 into the program deck. Segregate duplicates and non-matching cards from the program deck. A 315 or 319 Collator will:
  - (a) Segregate program cards if more than one instruction/constant was assigned to a memory location.
  - (b) Segregate invalid "a" addresses.
  - (c) Interfile a blank "a" address card into the program deck for each location not used.
3. List on tabulator, print all data. This listing will serve to replace Memory Chart (U1710) by providing complete information relative to the line of coding and provide a space opposite unused words for noting additional instructions, if necessary.

C. Verify legality of Operation Codes

1. Sort instruction and constant cards which represent instructions to Operation Code, card columns 21 - 22.
2. List on tabulator, print all data. Set sub-total controls on Operation Code to obtain spacing between codes.
  - (a) Scan list visually for illegal Operation Codes.
  - (b) Check Operation Codes which should be blank in "m" or "c" locations.
  - (c) Check for correct stacker location in 47 and 57 codes.
  - (d) In the 81,96,11 and 46 codes check to be sure that the "c" address is no earlier than the recommended location for minimum latency.
  - (e) In the same 81,96,11 and 46 codes check to be sure that the instruction address is no later than that recommended for minimum latency.
  - (f) Scan shift codes for correct location of "n"



- (g) Optional: if word times have been punched in the cards and if a 330 Calculating Punch, UNIVAC 60/120 or UNIVAC Solid-State 90 is available, calculations may be made to determine the word times between instruction address and the "c" address to check the word times determined by the programmer.
- (h) If a UNIVAC 60/120 or UNIVAC Solid-State 90 is available, it may also be desirable to program to determine whether the "c" is spaced from the "m" address by at least the execution word times.

D. Verify legality of "m" addresses in instruction cards

1. Sort instruction and constant cards which represent instructions to Page and Line, card columns 11 - 15, (minor) then to "m" address, card columns 23 - 26 (major).
2. Merge program testing deck # 2 into the program deck. Segregate non-matching cards from the program deck. A 315 or 319 Collator will:
  - (a) Segregate invalid "m" addresses.
  - (b) Interfile a blank "m" address card into the program deck for each memory location not used as "m".
3. List on tabulator, print all data. Set sub-total controls on two most significant digits of "m" to obtain spacing. This listing will group all "m" references to memory by Page and Line or Sequence number and provide a single blank space opposite each memory location not used.
4. List on tabulator all program cards which were segregated on the merge operation. These cards will be:
  - register addresses
  - shifts of 10 positions
  - addresses in machine code.
  - (a) Check carefully all instances where the same "m" address is in more than one card. Duplicates may be for legal multiple use of the same operand or may indicate erroneous coding. Use Page and Line or Sequence number to refer to coding charts.

- (b) Check "m" addresses in interlace positions to be sure that correct addresses have been used for first read, second read, print and punch interlace words.

E. Verify legality of "c" addresses in instruction cards

1. Sort instruction and constant cards which represent instructions to Page and Line, card columns 11 - 55, (minor) then to "c" address, card columns 27 - 30 (major).
2. Merge program testing deck # 3 into the program deck. Segregate non-matching cards from the program deck. A 315 or 319 Collator will:
  - (a) Segregate invalid "c" addresses.
  - (b) Interfile a blank "c" address card into the program deck for each memory location not used as "c".
3. List on tabulator, print all data. Set sub-total controls on the two most significant digits of "c" to obtain spacing. This listing will group all "c" references to memory by Page and Line or Sequence number and provide a single blank space opposite each memory location not used.
4. List on tabulator all program cards which were segregated on the merge operation. These cards will be register addresses.
5. Check carefully all instances where the same "c" address is in more than one card. Duplicates may indicate a legal condition, as where the "c" address concerned is the point to which a number of instructions jump, or may indicate erroneous use of same "c" address in two or more instructions. Use Page and Line or Sequence numbers to refer to coding charts.
6. Check "c" addresses for all words which are in bands used as interlaces in the program to insure that the correct location has been used for first read, second read, print and punch data. For example, if an instruction refers to a "c" address which is supposed to be in first read words, determine that the address actually is in first read interlace position.

F. Verify continuity from "c" address to "a" address

1. Remove all PT-3 cards from program cards used in the "c" verification above.
2. Keeping the program cards in "c" address order, reproduce the instruction cards gang punching into the reproduced cards a carriage line space control hole and some printable position to identify each card as being part of the reproduced deck.
3. Sort reproduced deck (PT-4) on "a" address, card columns 17 - 20.
4. On a 319 Collator merge the program cards ahead of matching cards in the reproduced deck. Compare the "c" address in the program deck, card columns 27 - 30 to the "a" address in the reproduced deck, card columns 17 - 20. Segregate non-matching cards from both decks.
5. List all cards on the tabulator, print all data. Set carriage line spacing to 1/1.
6. For nearly every address "c" of a true instruction card there should be a matching instruction address, ("a"), in the reproduced card listed on the line below. The carriage line space control hole separates the related pair for reading convenience.
  - (a) Check all cases where a single line appears for either the original deck or reproduced deck. Be sure to print the identification of the reproduced deck.
  - (b) Single lines may indicate constants, or may indicate a "c" address reference to an instruction address which is not represented by a card, or may indicate a register which is used to contain the next instruction, or may indicate one of the addresses involved in a comparison operation.

G. Verify insertion of 22 test of HSR buffer

1. Return to original coding charts. Be sure that all changes from preceding procedures have been made on the chart and in the program cards.

2. For each type of input card and every branching condition possible from the card, add up the word times on a Remington adding machine or "99" calculator from each 42 test in the instruction sequence to the next applicable 42 test. Check accumulated word times to insure that the interval from one 42 test (including executing that test) up to the next 42 test does not exceed 1000 word times (5 drum revolutions). Save the printed copy to file with the coding papers, identified by input condition tested. Note that the instruction address can be entered at the left of the word times, allowing four total positions for the word times, thus providing reference to the instruction address related to the word times figure.

#### IV. CONSIDERATIONS

- A. When frequent use is made of the standard High-Speed Reader Reserve Storage Routine and Read-Punch Unit Backup Punching Routine, it may be desirable to include the complete coding in the three program testing decks.
- B. Some programmers will find it advantageous to punch a designating position into each of the program testing decks to identify each memory location used in an interlace.
- C. It will always be desirable to rerun each of the lists described above when final corrections have been made. These lists should become part of the Run Book. Corrected lists will insure efficient use of time when debugging at the computer.



**UNIVAC<sup>®</sup>—The FIRST Name in Electronic Computing Systems**