

UNCLASSIFIED
AD 414534

DEFENSE DOCUMENTATION CENTER
FOR
SCIENTIFIC AND TECHNICAL INFORMATION
CAMERON STATION, ALEXANDRIA, VIRGINIA



UNCLASSIFIED

NOTICE: When government or other drawings, specifications or other data are used for any purpose other than in connection with a definitely related government procurement operation, the U. S. Government thereby incurs no responsibility, nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use or sell any patented invention that may in any way be related thereto.

414534

MEMORANDUM REPORT
M63-27-1

UNITED STATES ARMY

CATALOGED BY DDC

AS AD No.

414534

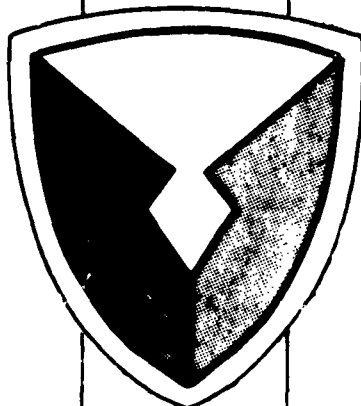
FRANKFORD ARSENAL

PRINT EDIT ROUTINE GENERATOR
(PERGE)

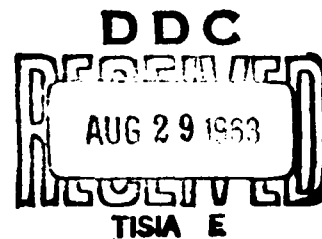
by

D. L. WRIGHT

REPORT M63-27-1



February 1963



PHILADELPHIA 37, PA.

Requests for additional copies are to be made to Defense Documentation Center (DDC), Arlington Hall Station, Arlington 12, Virginia.

The findings in this report are not to be construed as an official Department of the Army position.

Mathematic Branch
Physics Research Laboratory
Research and Development Group
Frankford Arsenal
Philadelphia 19137, Pa.

MEMORANDUM REPORT M63-27-1
February 1963

PRINT EDIT ROUTINE GENERATOR
(PERGE)

Prepared by:

Donald L. Wright

DONALD L. WRIGHT
Mathematician

Reviewed by:

ER Thilo

E. R. THILO
Chief
Physics Research Laboratory

Approved by:

A. L. Jamieson

A. L. JAMIESON
Deputy Director
Pitman-Dunn Institute for Research

ABSTRACT

This report describes the routine (program) developed for the UNIVAC Solid State 90 Computer to edit and print a line of numerical information on the high speed printer. The details of the construction of the routine, as well as instructions for its use, are described. An example of the coding produced is also included.

The routine has proven very useful for work at this arsenal and it is presented in this report to make it available to other users of the USS-90 machine. The general ideas may also find application in the work of other type computers.

TABLE OF CONTENTS

	<u>Page</u>
INTRODUCTION	1
DISCUSSION	1
SPECIFICATION CARDS - INPUT TO PERGE	2
INTERNAL OPERATION	3
Redundent Nature of Instructions to Set Up Interlace	3
Fixed Point Generator	4
Floating Point Generator	7
Efficiency of Generated Routine	7
Output from PERGE	8
Sample Fixed Point Routine	8
Instruction Sets	14
DISTRIBUTION	17

PRINT EDIT ROUTINE GENERATOR (PERGE)

INTRODUCTION

PERGE is a routine programmed for the Solid State 90 Computer to generate coding in X-6 assembly language which will edit and print one line of numerical information. The information to be printed can be one line of fixed point or floating point numbers, but a combination of both types on the same line is not permitted.

DISCUSSION

The UNIVAC Solid State 90 Computer has, as part of its peripheral equipment, an on-line printer capable of printing up to 600 lines per minute, one line at a time, with 130 characters on a line. The 130 characters printed on any one line are determined from the contents of 26 ten-digit locations within any of the 25 bands dividing the magnetic drum.

The format of any printed line can then be varied according to the contents of these 26 locations. These locations must be set up by the programmer writing a routine for each different line of print desired. To merely print sequences of numbers is not difficult, and takes few programming instructions to perform. However, if it is desired to add minus signs, decimal points, or blank spaces in the line, the amount of coding necessary to insert these characters in the proper positions is greatly increased. In the sample fixed point routine (discussed later), 308 lines of coding are used to print nine fixed point numbers with their associated signs and decimal points.

The actual programming of these instructions is not at all difficult, but it is certainly time consuming, tedious, repetitious, and usually boring to the programmer.

PERGE will eliminate the need for writing print routines for every new format that is required for the output of numerical data. A PERGE-generated routine can be included in any program, with very few restrictions placed on the user.

The generated routine will print the numbers found in the locations designated by permanent tags. Each number will have

its associated sign, and three blank spaces will always be inserted between two successive numbers. No matter how many or how few numbers are printed on one line, the first number always begins on the extreme left of the page and continues in this format, with only three spaces between each number. The numbers will not be centered on the page.

This routine has proven very useful for work at this arsenal. Informal exchange of information with other users of the USS 90 machine has indicated a general need for this type of routine. Also, the general ideas may find application in the work of other type computers.

SPECIFICATION CARDS - INPUT TO PERGE

The information necessary to PERGE in order to generate the desired print routine is punched into cards which are read by the PERGE program. These cards designate the type of routine to be generated, fixed or floating point, the tag associated with each desired number, and the sequence in which these numbers are to be printed.

A maximum of eight specifications may be made on one card. For each specification word, a ten-column field is used on the card in the following format.

<u>Specification Word</u>	<u>Card Columns</u>
1	1 - 10
2	11 - 20
3	21 - 30
4	31 - 40
5	46 - 55
6	56 - 65
7	66 - 75
8	76 - 85

The specification word of ten digits has the following format:

NNNNN~~AAA~~ D₁D₂

where NNNNN is the permanent tag assigned to the number to be printed.

D₁ is the number of digits that are to appear to the left of the decimal point.

D_2 is the number of digits that are to appear to the right of the decimal point.

D_1 and D_2 can be any number from 0 to 9. The number 10 is represented by the letter K. If a floating point print routine is desired, then D_1 and D_2 are not numbers, but the letters KK.

For a fixed point routine, a maximum of 16 numbers may be designated to be printed. The size of each of these numbers is variable, but must be no greater than ten digits. Of course, the total number of digits plus the associated signs, decimal points, and spaces, cannot be more than 130. The maximum number of floating point numbers that can be printed is eight.

Since only eight specification words can be placed on one card, the input to PERGE (the specification card) can be either one or two cards. No special indications are required if two cards are necessary. PERGE is programmed to handle either case with no intervention by the user or operator.

The object program which uses a PERGE-generated print routine for fixed point numbers must position those numbers according to the number of digits to be printed. For example, if the fixed point number 123.1234 is to be printed with three digits to the left and two digits to the right of the decimal point, the number must be positioned in its storage location as

0000012312

A print interlace (Pl) must also be specified in a type 4 card prior to assembly with the object program.

INTERNAL OPERATION

Redundant Nature of Instructions to Set Up Interlace

The instructions which must be written to produce a program to print a line of numerical information are, in general, redundant, and can be divided into several sets. Each set of instructions will be similar in nature for each number, but will, of course, usually be different with respect to operands of some instructions and, in some cases, a few instructions have to be added or a few deleted.

Some sets or subsets will always be the same, or at least of the same form, for each output word desired. One example is the set

of three instructions necessary to load one of the registers with the particular output word and the transfer to the general sign determination routine. The set of instructions which is used to make the sign determination is written only once and is generalized so that it may be used for each word. This set of instructions is always the same and can be written first, without consideration of the form, number, or layout of the numbers to be printed.

Floating point and fixed point output instructions have really the same sort of sets of instructions. Floating point, of course, has fewer sets, and these are less variable than those in the fixed point output. Because of this, instructions to print floating point output are much simpler to generate. Bookkeeping and control functions can be kept to a minimum in the generating form.

Fixed Point Generator

The first sets of instructions which are generated, as mentioned previously, are the set to determine the sign of the word and the set to load the register with the first word. Having generated these sets, the remaining sets can be divided into the following general categories.

Set C - Shifting and storing of the actual sign after its determination.

Set D - Shifting a space open for the later insertion of the decimal point.

Set E - Shifting the number to place it in proper position for the print interlace.

Set F - Buffing the signs, decimal points, and possible parts of previous words, and storing in the print interlace.

All four of these sets contain shift instructions which, of course, vary with each number and relative position of that number in the layout of the print line. Set F can vary as to which instructions should be written according to whether or not there is a sign in the print word being processed, one sign, two signs, a decimal point, two decimal points, or none, a part of another number remaining from a previous word, or any combination of these conditions. To provide the necessary control functions and variables, ten tables are generated from the data read from the specification cards.

Two of the tables are merely the untranslated specification words - one table is for the primed and one for the unprimed parts of the words. These tables are used to pick up the tags which designate the names of the numbers to be printed, and the numbers D_1 and D_2 are translated to make up two more tables, one for each D_1 and D_2 . All four of these tables have a maximum of 16 entries each. While these tables are being generated, a check is made to insure that no more than 130 characters will be necessary to print the numbers as specified. If the number of characters required exceeds 130, a stop is provided before any coding is generated.

The table containing D_2 is now a set of variables which can be inserted in a shift instruction in Set D to open a space for the insertion of a decimal point.

A table of 13 entries is now constructed to give a picture of the complete layout of the desired print line. This layout is generated so that the program can scan each entry, or code word, and make a determination as to how many decimal points there are or how many signs and their positions in the particular print word being operated on. The codes used in making up these entries are:

- 0 - Space between numbers
- 1 - Decimal point
- 2 - Sign of a number
- 3 - Digit of a number.

As an example of one of these entries, assume that the first specification word read by PERGE was AAAAF $\Delta\Delta\Delta$ 43. The first entry of the code word table would appear as follows:

2333313330

Using this table of code words, the two erase word tables can be easily manufactured. The erase words are used in the generated routine to insure that the spaces between the printed numbers will be blanks, and not printed zeros. One table contains the unprimed part and the other the primed part of the erase words. These entries are generated from each code word in the following way.

1. A word of all one's is added to the code word.
2. The result is translated to card code.
3. The primed part of this translation is multiplied by 4.
4. This result is translated to card code and the primed part is stored as the primed part of the erase word.

5. This primed part of the erase word is then translated to card code.
6. The unprimed part of this translation is stored as the unprimed part of the erase word.

The table containing the entries used in Set C to position the sign of each number (N1) is made for all 16 possible entries according to the following formula.

$$N1_i = 10 - [(15 - N1_{i-1} + Z_{i-1}) \bmod 10]$$

where $Z_i = D1_i + D2_i$

$$N1_0 = 10$$

$$Z_0 = 6.$$

The table for Set F to position each decimal point (N4) is constructed as follows.

$$N4_i = (N4_{i-1} + D2_{i-1} + D1_i + 5) \bmod 10$$

where $N4_0 = 0$

$$D2_0 = -4.$$

The table for positioning each number in Set E (N3) is simply

$$N3_i = (N4_i + 1) \bmod 10.$$

In addition to the code word table, three other control words are used. Control word one (CW1) contains a record of how many, if any, signs are to be inserted in the print word currently being assembled. The contents of CW1 will cause certain buff and store instructions to be inserted or deleted from Set C.

CW2, similarly, keeps a record of the number of decimal points which must be inserted and it, too, causes certain buff and store instructions to be inserted or deleted from Set E.

CW3 indicates the condition that there may or may not exist a part of a number from a previous print word which could not be wholly contained in that word. This also causes some buff and store instructions to be inserted in Set F.

As information concerning the layout of each print word is extracted from the corresponding entry in the code word table, the codes are erased. The condition of the code word then gives an

indication as to when a print word has been completed and is ready to be stored in the interlace. When the word is not completely erased, and a 2 code (which represents a sign) appears next, this indicates that the print word has not been fully assembled and that a new tag must be picked up from the specification word table.

The end conditions (i.e., when all the desired numbers have been processed) dictate the particular subsets which may be required to complete the generated routine. For example, the last code word may not be zero, and all the specification words may have been used. This indicates that any number of subsets of sets D, E, and F may be necessary before terminating the generation.

Floating Point Generator

The coding to set up and print floating point numbers, in general, is much simpler than that necessary to print fixed point data. The sets required are fewer in number, have fewer variables, and are easier to construct than the fixed point sets.

Because of the format design, there are basically only two sets in addition to sets A and B. If the floating point words to be printed are numbered, starting with the leftmost number as 1, the same instructions for all the odd-numbered and a slightly different set for all the even-numbered floating point numbers can be generated. The only tables required then are the two for the specification words containing the tags assigned.

The erase words necessary then are reduced to four, one of which may or may not be required, depending upon whether the last number was even or odd.

The end conditions are simple in that, again, there may only be two different subsets required to terminate the instruction generation.

Efficiency of Generated Routine

In determining the efficiency of the instructions generated by PERGE, the major consideration was how many instructions were generated which could have been omitted with no harmful effect to the result. All conditions which might appear during actual use of PERGE have obviously not been examined, especially in the fixed point

generation, but it appears that very few conditions will cause unnecessary instructions to be assembled. One example is the case in which there are two decimal points in the same word, one of which happens to be in the same digit position as in the words containing the unprimed and primed parts of the decimal point itself.

Output from PERGE

Output from PERGE is in two forms. The high speed printer will print each instruction as it is generated, and a card is punched for each instruction in the standard X-6 format. A type 7 card precedes the first detail card of the routine generated, and a type 9 card is punched as the last card of the routine.

Sample Fixed Point Routine

The following specification words were supplied to PERGE in order to obtain a routine to print nine fixed point numbers of variable length.

Tag	D1 D2
AAAAF	43
BBBBF	72
CCCCF	21
DDDDF	31
EEEEF	82
FFFFF	62
GGGGF	45
HHHHF	35
IIIIF	61

The following is a copy of the entire routine generated by PERGE to print this set of numbers.

7	PRN001				
8	PRN002	04F	STA	10F	
8	PRN003		LDA	RX	
8	PRN004		CAA		
8	PRN005		CLL		
8	PRN006		TEQ	10F	
8	PRN007		LDA	02F	
8	PRN008		LDL	03F	10F
8	PRN009	02F U			-
8	PRN010	03F P			-
8	PRN011	PRNTF	STA	EXITF	
8	PRN012		LDX	AAAAF	
8	PRN013		LDA		04F
8	PRN014		SHL	0900	
8	PRN015		STA	05F	
8	PRN016		LDA	RL	
8	PRN017		SHL	0900	
8	PRN018		STA	06F	
8	PRN019		LDA	RX	
8	PRN020		CLX		
8	PRN021		SHR	0300	
8	PRN022		SHL	0100	
8	PRN023		BUF	03F	
8	PRN024		SHR	0600	
8	PRN025		SHR	0K00	
8	PRN026		MTC		
8	PRN027		ZUP		
8	PRN028		ERS	31F	
8	PRN029		ATL		
8	PRN030		STX	07F	
8	PRN031		LDA	08F	
8	PRN032		LDX	09F	
8	PRN033		SHR	0500	
8	PRN034		BUF	RL	
8	PRN035		BUF	05F	
8	PRN036		STA	P1U01	
8	PRN037		LDA	RX	
8	PRN038		BUF	07F	
8	PRN039		BUF	06F	
8	PRN040		STA	P1P01	
8	PRN041		LDX	8888F	
8	PRN042		LDA		04F
8	PRN043		SHL	0700	
8	PRN044		STA	05F	
8	PRN045		LDA	RL	
8	PRN046		SHL	0700	
8	PRN047		STA	06F	
8	PRN048		LDA	RX	
8	PRN049		CLX		
8	PRN050		SHR	0200	
8	PRN051		SHL	0100	
8	PRN052		BUF	03F	
8	PRN053		SHR	0100	
8	PRN054		STX	27F	
8	PRN055		MTC		
8	PRN056		ZUP		
8	PRN057		ERS	32F	
8	PRN058		BUF	05F	
8	PRN059		STA	P1U02	
8	PRN060		LDA	RX	

8	PRN061	BUF	06F	
8	PRN062	STA	P1P02	
8	PRN063	LDX	CCCCF	
8	PRN064	LDA		04F
8	PRN065	SHL	0300	
8	PRN066	STA	05F	
8	PRN067	LDA	RL	
8	PRN068	SHL	0300	
8	PRN069	STA	06F	
8	PRN070	LDA	RX	
8	PRN071	CLX		
8	PRN072	SHR	0100	
8	PRN073	SHL	0100	
8	PRN074	BUF	03F	
8	PRN075	BUF	27F	
8	PRN076	STX	27F	
8	PRN077	HTC		
8	PRN078	ZUP		
8	PRN079	ERS	33F	
8	PRN080	ATL		
8	PRN081	STX	07F	
8	PRN082	LDA	08F	
8	PRN083	LDX	09F	
8	PRN084	STA	28F	
8	PRN085	STX	29F	
8	PRN086	LDA	08F	
8	PRN087	LDX	09F	
8	PRN088	SHR	0900	
8	PRN089	BUF	RL	
8	PRN090	BUF	05F	
8	PRN091	BUF	28F	
8	PRN092	STA	P1U03	
8	PRN093	LDA	RX	
8	PRN094	BUF	07F	
8	PRN095	BUF	06F	
8	PRN096	BUF	29F	
8	PRN097	STA	P1P03	
8	PRN098	LOX	0000F	
8	PRN099	LDA		04F
8	PRN100	SHL	0300	
8	PRN101	STA	05F	
8	PRN102	LDA	RL	
8	PRN103	SHL	0300	
8	PRN104	STA	06F	
8	PRN105	LDA	RX	
8	PRN106	CLX		
8	PRN107	SHR	0100	
8	PRN108	SHL	0100	
8	PRN109	BUF	03F	
8	PRN110	SHR	0900	
8	PRN111	SHR	0K00	
8	PRN112	BUF	27F	
8	PRN113	HTC		
8	PRN114	ZUP		
8	PRN115	ERS	34F	
8	PRN116	ATL		
8	PRN117	STX	07F	
8	PRN118	LDA	08F	
8	PRN119	LDX	09F	
8	PRN120	SHR	0800	

8 PRN121	BUF	RL	
8 PRN122	BUF	05F	
8 PRN123	STA	P1U04	
8 PRN124	LDA	RX	
8 PRN125	BUF	07F	
8 PRN126	BUF	06F	
8 PRN127	STA	P1P04	
8 PRN128	LDX	EEEEF	
8 PRN129	LDA		04F
8 PRN130	SHL	0600	
8 PRN131	STA	05F	
8 PRN132	LDA	RL	
8 PRN133	SHL	0600	
8 PRN134	STA	06F	
8 PRN135	LDA	RX	
8 PRN136	CLX		
8 PRN137	SHR	0200	
8 PRN138	SHL	0100	
8 PRN139	BUF	03F	
8 PRN140	SHR	0300	
8 PRN141	STX	27F	
8 PRN142	MTC		
8 PRN143	ZUP		
8 PRN144	ERS	35F	
8 PRN145	BUF	05F	
8 PRN146	STA	P1U05	
8 PRN147	LDA	RX	
8 PRN148	BUF	06F	
8 PRN149	STA	P1P05	
8 PRN150	LDX	FFFFF	
8 PRN151	LDA		04F
8 PRN152	SHL	0100	
8 PRN153	STA	05F	
8 PRN154	LDA	RL	
8 PRN155	SHL	0100	
8 PRN156	STA	06F	
8 PRN157	LDA	RX	
8 PRN158	CLX		
8 PRN159	SHR	0200	
8 PRN160	SHL	0100	
8 PRN161	BUF	03F	
8 PRN162	SHR	0600	
8 PRN163	BUF	27F	
8 PRN164	STX	27F	
8 PRN165	MTC		
8 PRN166	ZUP		
8 PRN167	ERS	36F	
8 PRN168	ATL		
8 PRN169	STX	07F	
8 PRN170	LDA	08F	
8 PRN171	LDX	09F	
8 PRN172	SHR	0200	
8 PRN173	BUF	RL	
8 PRN174	BUF	05F	
8 PRN175	STA	P1U06	
8 PRN176	LDA	RX	
8 PRN177	BUF	07F	
8 PRN178	BUF	06F	
8 PRN179	STA	P1P06	
8 PRN180	LDA	27F	

8	PRN181	HTC		
8	PRN182	ZUP		
8	PRN183	ERS	37F	
8	PRN184	ATL		
8	PRN185	STX	07F	
8	PRN186	LDA	08F	
8	PRN187	LDX	09F	
8	PRN188	SHR	0500	
8	PRN189	BUF	RL	
8	PRN190	STA	P1U07	
8	PRN191	LDA	RX	
8	PRN192	BUF	07F	
8	PRN193	STA	P1P07	
8	PRN194	LDX	6868F	
8	PRN195	LDA		04F
8	PRN196	SHL	0800	
8	PRN197	STA	03F	
8	PRN198	LDA	RL	
8	PRN199	SHL	0800	
8	PRN200	STA	06F	
8	PRN201	LDA	RX	
8	PRN202	CLX		
8	PRN203	SHR	0500	
8	PRN204	SHL	0100	
8	PRN205	BUF	03F	
8	PRN206	SHR	0700	
8	PRN207	SHR	0K00	
8	PRN208	STX	27F	
8	PRN209	HTC		
8	PRN210	ZUP		
8	PRN211	ERS	38F	
8	PRN212	ATL		
8	PRN213	STX	07F	
8	PRN214	LDA	08F	
8	PRN215	LDX	09F	
8	PRN216	SHR	0600	
8	PRN217	BUF	RL	
8	PRN218	BUF	05F	
8	PRN219	STA	P1U08	
8	PRN220	LDA	RX	
8	PRN221	BUF	07F	
8	PRN222	BUF	06F	
8	PRN223	STA	P1P08	
8	PRN224	LDX	HHHHF	
8	PRN225	LDA		04F
8	PRN226	SHL	0400	
8	PRN227	STA	05F	
8	PRN228	LDA	RL	
8	PRN229	SHL	0400	
8	PRN230	STA	06F	
8	PRN231	LDA	RX	
8	PRN232	CLX		
8	PRN233	SHR	0500	
8	PRN234	SHL	0100	
8	PRN235	BUF	03F	
8	PRN236	BUF	27F	
8	PRN237	STX	27F	
8	PRN238	HTC		
8	PRN239	ZUP		
8	PRN240	ERS	39F	

8	PRN241		ATL		
8	PRN242		STX	07F	
8	PRN243		LDA	08F	
8	PRN244		LDX	09F	
8	PRN245		SHR	0900	
8	PRN246		BUF	RL	
8	PRN247		BUF	05F	
8	PRN248		STA	P1U09	
8	PRN249		LDA	RX	
8	PRN250		BUF	07F	
8	PRN251		BUF	06F	
8	PRN252		STA	P1P09	
8	PRN253		LDX	IIIIIF	
8	PRN254		LDA		04F
8	PRN255		SHL	0100	
8	PRN256		STA	05F	
8	PRN257		LDA	RL	
8	PRN258		SHL	0100	
8	PRN259		STA	06F	
8	PRN260		LDA	RX	
8	PRN261		CLX		
8	PRN262		SHR	0100	
8	PRN263		SHL	0100	
8	PRN264		BUF	03F	
8	PRN265		SHR	0600	
8	PRN266		BUF	27F	
8	PRN267		STX	27F	
8	PRN268		HTC		
8	PRN269		ZUP		
8	PRN270		ERS	40F	
8	PRN271		BUF	05F	
8	PRN272		STA	P1U10	
8	PRN273		LDA	RX	
8	PRN274		BUF	06F	
8	PRN275		STA	P1P10	
8	PRN276		LDA	27F	
8	PRN277		HTC		
8	PRN278		ZUP		
8	PRN279		ERS	41F	
8	PRN280		ATL		
8	PRN281		STX	07F	
8	PRN282		LDA	08F	
8	PRN283		LDX	09F	
8	PRN284		SHR	0500	
8	PRN285		BUF	RL	
8	PRN286		STA	P1U11	
8	PRN287		LDA	RX	
8	PRN288		BUF	07F	
8	PRN289		STA	P1P11	
8	PRN290		PRN	P1001	EXITF
8	PRN291	08F	U	.	
8	PRN292	09F	P	.	
8	PRN293	31F		0TTTT	TTTT0
8	PRN294	32F		000TT	TTTTT
8	PRN295	33F		TTT00	00TTT
8	PRN296	34F		T0000	TTTTT
8	PRN297	35F		0000T	TTTTT
8	PRN298	36F		TTTTT	0000T
8	PRN299	37F		TTTTT	TTT00
8	PRN300	38F		00TTT	TTTTT

```

8 PRN301 39F      TT000 OTTTT
8 PRN302 40F      TTTT 0000T
8 PRN303 41F      TTTT TT000
8 PRN304 P1U12 U
8 PRN305 P1P12 P
8 PRN306 P1U13 U
8 PRN307 P1P13 P
9 PRN308

```

The preceding was assembled into machine code along with a sample set of numbers in their respective tags. The following is the line of print obtained from the generated routine.

```

1234.567 -1234567.89 12.3 -123.4 12345678.90 -123456.78 1234.56789 -123.45678 123456.7

```

A second version of PERGE has also been written which will generate coding in the S-4 assembly language. This language is not significantly different from X-6 in its format; thus no major changes to PERGE were necessary. With the S-4 version, the first and last cards are the 'HED C' control to clear the temporary tag table.

It is expected that several additions will shortly be made to both versions of PERGE. An option of choosing a print interlace other than P1 is being added. That is, for any particular generated routine, the number of the interlace, 0-9, will be controlled by the user. A possibility of adding a second print interlace with column headings corresponding to the tags is also under consideration.

Instruction Sets

Set A - Sign Determination

```

04F      STA      10F
          LDA      RX
          CAA
          CLL
          TEQ      10F
          LDA      02F
          LDL      03F      10F
02F U    --
03F P    --

```

Set B - Enter and Pick up New Tag

PRNTF	STA	EXITF	
	LDX	(Tag)	
	LDA		04F

Set C - Shift and Store Sign

SHL	0N00
BUF	05F
STA	05F
LDA	RL
SHL	0N00
BUF	06F
STA	06F

Set D - Open Space for Decimal Point

LDA	RX
CLS	
SHR	0N00
SHL	0100
BUF	03F

Set E - Position Word for Interlace

SHR	0N00
SHR	0K00
BUF	27F
STX	27F

Set F - Store Word in Interlace

MTC	
ZUP	
ERS	nnF
ATL	
STX	07F
LDA	08F
LDX	09F
SHR	0N00
BUF	RL
BUF	05F
STA	PlUnn
LDA	RX
BUF	07F
BUF	06F

STA	P1Pnn
STA	28F
STX	29F
BUF	28F
BUF	29F
LDA	27F
STX	P1Pnn

Set G - Erase Words for Floating Point

31F	OTTOTTTTTT
32F	TT0000TT0T
33F	TTTTTTT000
34F	TT00000000

DISTRIBUTION

- | | |
|--|---|
| 1 - Headquarters
U.S.Army Materiel Command
Attn: AMCRD-RD-CM
Washington 25, D. C. | 1 - Commanding Officer
U.S.Army Ballistic Research
Laboratory
Aberdeen Proving Ground, Md. |
| 1 - Attn: AMCRD-DE-W | 1 - Commanding Officer
Picatinny Arsenal
Attn: Library
Dover, N. J. |
| 1 - Attn: AMCRD-DE-N | |
| 1 - Commanding General
U.S.Army Munitions Command
Attn: Library
Picatinny Arsenal
Dover, N. J. | 1 - Commanding Officer
Rock Island Arsenal
Attn: Library
Rock Island, Ill. |
| 1 - Commanding General
U.S.Army Missile Command
Attn: Library
Redstone Arsenal, Ala. | 1 - Commanding Officer
Watertown Arsenal
Attn: Library
Watertown 72, Mass. |
| 1 - Commanding General
U.S.Army Electronics Command
Fort Monmouth, N. J. | 1 - Commanding Officer
Watervliet Arsenal
Attn: Library
Watervliet, N. Y. |
| 1 - Commanding General
U.S.Army Weapons Command
Attn: Library
Rock Island Arsenal, Ill. 61202 | 1 - Attn: Paul Loatman
1 - Commanding Officer
Springfield Armory
Attn: Library
Springfield, Mass. |
| 1 - Commanding General
U.S.Army Mobility Command
Attn: Library
Warren, Mich. 48090 | 1 - Commanding General
U.S.Army Research Office -
Washington
Washington 25, D. C. |
| 2 - U.S.Army Test and Evaluation
Command
Attn: Technical Library
Aberdeen Proving Ground, Md. | 1 - Commanding Officer
U.S.Army Research Office-
Durham
Box CM, Duke Station
Durham, N. C. |
| 1 - Commanding General
Chemical-Biological-Radiological
Agency
Army Chemical Center, Md. | 1 - Commanding Officer
Detroit Arsenal
Attn: Library
Centerline, Mich. |
| 1 - Commanding Officer
U.S.Army Ordnance Center and School
Aberdeen Proving Ground, Md. | |

- 1 - Commanding General
Attn: Gerald T. Eccles
Fort Detrick, Md.
- 1 - Chemical R & D Laboratories
Attn: Richard Campbell,
Systems Analysis Div
Edgewood Arsenal, Md.
- 1 - U.S. Marine Corps Liaison Officer
Hq, U.S. Army Munitions Command
Attn: AMSMU-LM
Dover, N. J.
- 1 - Combat Development Command
Liaison Officer
Hq, U.S. Army Munitions Command
Attn: AMSMU-LC
Dover, N. J.
- 1 - Commanding Officer
Harry Diamond Laboratories
Attn: AMXDO-TIB
Washington 25, D. C.
- 10 - Defense Documentation Center
Arlington Hall Station
Arlington 12, Va.
- 1 - Erie Electronics Division
Erie Resistor Corporation
645 W. 12th St.
Erie, Pa.
Attn: H. S. Herrick
- 1 - Ethyl Corporation
Research & Development Dept
P.O. Box 341
Baton Rouge 1, La.
Attn: Bernard A. Sobel
- 1 - Ford Instrument Co.,
31-10 Thomson Ave.
Long Island City, N. Y.
Attn: Austin Maher
- 1 - Hewlett-Packard Corporation
1501 Page Mill Rd
Palo Alto, Calif.
Attn: Paul Schmidt
- 1 - Remington Rand Univac
Program Library Services
315 Park Ave., South
New York 10, N. Y.
- 1 - Remington Rand Univac
Blue Bell, Pa.
- 1 - State Road Commission
Planning & Traffic Div
922 Quarrier St
Charleston, West Va.
Attn: Ray N. Shasteen
- 1 - Vickers Incorporated
Division of Sperry Rand Corp
1400 Oakman Blvd
Detroit 32, Mich.
Attn: Frank Rosett

AD- Physics Research Laboratory, Research and Development Gr Frankford Arsenal, Philadelphia, Pa., 19137 PRINT EDIT ROUTINE GENERATOR (PERGE) - by D. L. Wright Unclassified Report PA Rpt M63-27-1, Feb 63; 18 pp This report describes the routine (program) developed for the UNIVAC Solid State 90 Computer to edit and print a line of numerical information on the high speed printer. The details of the construction of the routine, as well as instructions for its use, are described. An example of the coding produced is also included. The routine has proven very useful for work at this arsenal and it is presented in this report to make it available to other users of the USS-90 machine. The general ideas may also find application in the work of other type computers. 1. Mathematics 2. Computers 3. Programming I. Rpt M63-27-1 II. D. L. Wright DISTRIBUTION LIMITATIONS: None; obtain copies from DDC.	AD- Physics Research Laboratory, Research and Development Gr Frankford Arsenal, Philadelphia, Pa., 19137 PRINT EDIT ROUTINE GENERATOR (PERGE) - by D. L. Wright Unclassified Report PA Rpt M63-27-1, Feb 63; 18 pp This report describes the routine (program) developed for the UNIVAC Solid State 90 Computer to edit and print a line of numerical information on the high speed printer. The details of the construction of the routine, as well as instructions for its use, are described. An example of the coding produced is also included. The routine has proven very useful for work at this arsenal and it is presented in this report to make it available to other users of the USS-90 machine. The general ideas may also find application in the work of other type computers. 1. Mathematics 2. Computers 3. Programming I. Rpt M63-27-1 II. D. L. Wright DISTRIBUTION LIMITATIONS: None; obtain copies from DDC.
AD- Physics Research Laboratory, Research and Development Gr Frankford Arsenal, Philadelphia, Pa., 19137 PRINT EDIT ROUTINE GENERATOR (PERGE) - by D. L. Wright Unclassified Report PA Rpt M63-27-1, Feb 63; 18 pp This report describes the routine (program) developed for the UNIVAC Solid State 90 Computer to edit and print a line of numerical information on the high speed printer. The details of the construction of the routine, as well as instructions for its use, are described. An example of the coding produced is also included. The routine has proven very useful for work at this arsenal and it is presented in this report to make it available to other users of the USS-90 machine. The general ideas may also find application in the work of other type computers. 1. Mathematics 2. Computers 3. Programming I. Rpt M63-27-1 II. D. L. Wright DISTRIBUTION LIMITATIONS: None; obtain copies from DDC.	AD- Physics Research Laboratory, Research and Development Gr Frankford Arsenal, Philadelphia, Pa., 19137 PRINT EDIT ROUTINE GENERATOR (PERGE) - by D. L. Wright Unclassified Report PA Rpt M63-27-1, Feb 63; 18 pp This report describes the routine (program) developed for the UNIVAC Solid State 90 Computer to edit and print a line of numerical information on the high speed printer. The details of the construction of the routine, as well as instructions for its use, are described. An example of the coding produced is also included. The routine has proven very useful for work at this arsenal and it is presented in this report to make it available to other users of the USS-90 machine. The general ideas may also find application in the work of other type computers. 1. Mathematics 2. Computers 3. Programming I. Rpt M63-27-1 II. D. L. Wright DISTRIBUTION LIMITATIONS: None; obtain copies from DDC.

AD- Physics Research Laboratory, Research and Development Gr Frankford Arsenal, Philadelphia, Pa., 19137 PRINT EDIT ROUTINE GENERATOR (PENGED) - by D. L. Wright Unclassified Report PA Rpt M43-27-1, Feb 63; 18 pp	ACCESSION NO.	1. Mathematics 2. Computers 3. Programming I. Rpt M43-27-1 II. D. L. Wright DISTRIBUTION LIMITATIONS: None; obtain copies from DDC.	This report describes the routine (program) developed for the UNIVAC Solid State 90 Computer to edit and print a line of numerical information on the high speed printer. The details of the construction of the routine, as well as instructions for its use, are described. An example of the coding produced is also included. The routine has proven very useful for work at this arsenal and it is presented in this report to make it available to other users of the USS-90 machine. The general ideas may also find application in the work of other type computers.	1. Mathematics 2. Computers 3. Programming I. Rpt M43-27-1 II. D. L. Wright DISTRIBUTION LIMITATIONS: None; obtain copies from DDC.	AD- Physics Research Laboratory, Research and Development Gr Frankford Arsenal, Philadelphia, Pa., 19137 PRINT EDIT ROUTINE GENERATOR (PENGED) - by D. L. Wright Unclassified Report PA Rpt M43-27-1, Feb 63; 18 pp ACCESSION NO. This report describes the routine (program) developed for the UNIVAC Solid State 90 Computer to edit and print a line of numerical information on the high speed printer. The details of the construction of the routine, as well as instructions for its use, are described. An example of the coding produced is also included. The routine has proven very useful for work at this arsenal and it is presented in this report to make it available to other users of the USS-90 machine. The general ideas may also find application in the work of other type computers.
--	----------------------	--	--	--	--