UNIVAC Sperry Rand

UNIVAC® 1107 THIN-FILM MEMORY COMPUTER

ANSWERS THE COMPLEX COMPUTATION NEEDS OF SCIENCE AND INDUSTRY

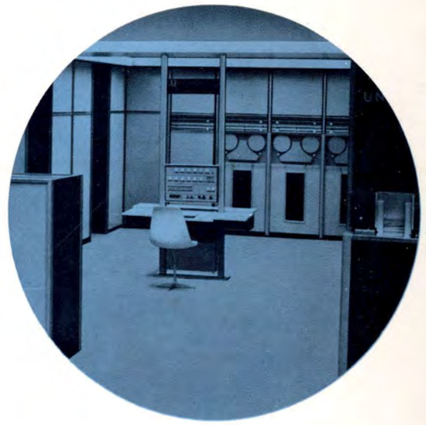
UNIVAC 1107 PROVIDES RELIABLE, HIGH-SPEED PERFORMANCE...

The UNIVAC 1107 Thin-Film Memory Computer represents the most significant departure from conventional data-processing systems since the introduction of solid-state circuitry. For the first time, a thin film memory is available in a commercial computing system.

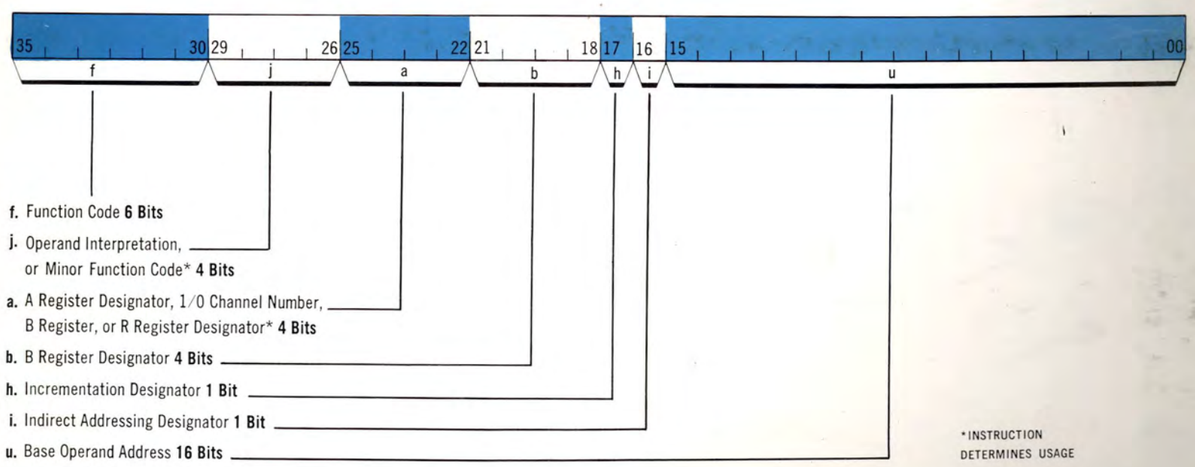
UNIVAC 1107 is a multi-purpose system which operates at extremely high computing speeds. So fast, in fact, that access time is measured in nanoseconds—billionths of a second. The Thin-Film Memory access time is only 300 nanoseconds!

UNIVAC Thin-Film Memory is more than simply a major technological advance. For with it comes a large number of major logical design features. Here are just a few: ▪ Automatic incrementation of 15 index registers. ▪ Automatic indexing. ▪ Multiple (16) arithmetic registers. ▪ Real-Time Clock. ▪ Indirect Addressing. ▪ Partial word transfers, adds, subtracts, and compares. ▪ Multiple adds and subtracts.

Featuring 16 input and 16 output channels, the UNIVAC 1107 provides for efficient bi-directional data transfers. These channels facilitate direct communication between peripheral equipment and internal memory.



THE INSTRUCTION WORD



GLOSSARY OF SYMBOLS AND TERMS

( ) Contents of.

u Address, or base address.

$U = u + (B_Q)$
Address of the operand to be used.

A_{n}, $(U)_n$ Subscript **n** indicates the bit number under discussion.

$( )_{17\text{-}00}$ Subscript numbers represent the range of bit positions.

B_Q Modifier portion of index register, $B_{17\text{-}00}$.

B_Δ Increment portion of index register, $B_{35\text{-}18}$, used to increment the modifier B_Q.

b b-designator (bits 21-18 of the instruction word).

a a-designator (bits 25-22 of the instruction word). In arithmetic instructions, **a** designates one of the A-registers; in input-output instructions, **a** desiginates an input or output channel; in certain other instructions, **a** designates a B-register.

R Refers to a group of special registers.

M Mask register (an R-register).

( )′ Prime on a quantity represents the one's complement of that quantity.

NI Next Instruction.

P Program Address count in the P-register.

→ Transfer the word (or words) shown at the left of the arrow to the address (or addresses) shown at the right of the arrow.

( )⊙( ) The logical product, logical AND etc., is defined by the table:

	01
0	00
1	01

( )⊕( ) The logical sum, also called the Inclusive OR:

	01
0	01
1	11

( )$\bar{\oplus}$( ) The logical difference, controlled complement, add-without-carry, Exclusive OR:

	01
0	01
1	10

In the mnemonic column:
Normal J designator

Sixths	S1	S2	S3	S4	S5	S6
Thirds	XT1		XT2		XT3	
Halves	H1		H2			
Whole	W					

An X placed before H1 or H2 extends the sign bit when transferring **to** arithmetic, halfwords being the only section of a word for which the sign extension may be specified. Thirds are always extended and sixths never extended.

In transfers **from** arithmetic, the signs are never extended. In this case T1, T2, and T3 must be used for the one-third word transfer.

In addition, one may transfer the lower half of the current instruction **to** arithmetic either extended or not. These are indicated by UOP and XUOP. The **h** and **i** designators are not interpreted in the usual way in this case; they are merely bits of the number transferred.

The execution of an arithmetic instruction includes the following steps:

1. Read out (B_b) from film memory specified by the **b** designator of the current instruction.
2. Store the result of the previous instruction at A_a of thin-film memory specified by **a** of the previous instruction.
3. Add $(B)_Q$ to **u** where $b \neq 0$ for current instruction.
4. Store the second part of the result of the previous instruction at $A_a + 1$ in film memory at the address one higher than that used in Step 2 if applicable.
5. Initiate access to the operand with the effective address obtained from Step 3. The storage reference may be made to film memory or core memory.
6. Initiate access to NI. This storage reference is made to core memory.
7. Read out (A_a) from film memory specified by **a** of the current instruction.
8. Add $(B)_\Delta$ to $(B)_Q$ if specified by **h** of the current instruction.
9. Store index register word at the film memory address used in Step 1.
10. Initiate arithmetic operation.
11. Initiate input-output word transfers if requested and then proceed to Step 1.

INSTRUCTION REPERTOIRE

f	j	NAME	DESCRIPTION		EXECUTION TIME IN μ SEC. Alternate Core Banks	EXECUTION TIME IN μ SEC. Same Core Bank	MNEMONIC CODE
01	0-17	Store Positive	$(A) \rightarrow U$		4.0	8.0	STP
02		Store Negative	$-(A) \rightarrow U$		4.0	8.0	STN
03		Store Magnitude	$\lvert(A)\rvert \rightarrow U$		4.0	8.0	STM
04		Store R_a	$(R_a) \rightarrow U$		4.0	8.0	STR
05		Store Zero	$0 \rightarrow U$ (Clear U)		4.0	8.0	STZ
06		Store B_a	$(B_a) \rightarrow U$		4.0	8.0	STB
10		Load Positive	$(U) \rightarrow A$		4.0	8.0	LDP
11		Load Negative	$-(U) \rightarrow A$		4.0	8.0	LDN
12		Load Positive Magnitude	$\lvert(U)\rvert \rightarrow A$		4.0	8.0	LDM
13		Load Negative Magnitude	$-\lvert(U)\rvert \rightarrow A$		4.0	8.0	LNM
14		Add	$(A) + (U) \rightarrow A$		4.0	8.0	ADD
15		Subtract	$(A) - (U) \rightarrow A$		4.0	8.0	SUB
16		Add Magnitude	$(A) + \lvert(U)\rvert \rightarrow A$		4.0	8.0	ADM
17		Subtract Magnitude	$(A) - \lvert(U)\rvert \rightarrow A$		4.0	8.0	SBM
20		Add and Load	$(A) + (U) \rightarrow A + 1$		4.0	8.0	ADL
21		Subtract and Load	$(A) - (U) \rightarrow A + 1$		4.0	8.0	SBL
22†		Block Transfer	$(V_1)_i \rightarrow (V_2)_i$ repeated k times. Initial V_1 address is $u + (B_b)_{17\text{-}0}$, and subsequent addresses are formed by incrementation by $(B_b)_{35\text{-}18}$. Similarly, V_2 addresses are $u + (B_a)_{17\text{-}0}$ incremented by $(B_a)_{35\text{-}18}$.		8.0	8.0	BTR
23		Load R_a	$(U) \rightarrow R_a$		4.0	8.0	LDR
24		Add to B_a	$(B_a) + (U) \rightarrow B_a$		4.0	8.0	ADB
25		Subtract from B_a	$(B_a) - (U) \rightarrow B_a$		4.0	8.0	SBB
26		Load B_a Modifier Only	$(U) \rightarrow B_{a17\text{-}0}$		4.0	8.0	LBM
27		Load B_a	$(U) \rightarrow B_a$		4.0	8.0	LDB
30		Multiply Integer	$(A) \cdot (U) \rightarrow A, A + 1$		12.7	16.7	MPI
31		Multiply Single (Integer)	$(A) \cdot (U) \rightarrow A$		12.0	16.0	MPS
32		Multiply Fractional	$(A) \cdot (U) \rightarrow A, A + 1$		13.3	17.3	MPF
34		Divide Integer	$(A, A + 1) \div (U)$; Quotient $\rightarrow A$ Remainder $\rightarrow A + 1$		31.0	35.0	DVI
35		Divide Single and Load (Fractional)	$(A) \div (U)$; Quotient $\rightarrow A + 1$ No Remainder		31.0	35.0	DVL
36		Divide Fractional	$(A, A + 1) \div (U)$; Quotient $\rightarrow A$ Remainder $\rightarrow A + 1$		31.0	35.0	DVF
40		Selective Set	$(A) \rightarrow A + 1$. Then set $(A + 1)_n$ for $(U)_n = 1$ i.e., $(A) \oplus (U) \rightarrow A + 1$		4.0	8.0	SSE
41		Selective Complement	$(A) \rightarrow A + 1$. Then complement $(A + 1)_n$ for $(U)_n = 1$ i.e., $(A) \bar{\oplus} (U) \rightarrow A + 1$		4.0	8.0	SCP
42		Selective Clear	$(A) \rightarrow A + 1$. Then clear $(A + 1)_n$ for $(U)_n = 0$ i.e., $(A) \odot (U) \rightarrow A + 1$		4.0	8.0	SCL
43		Selective Substitute	$(A) \rightarrow A + 1$. Then $(U)_n \rightarrow (A + 1)_n$ for $(M)_n = 1$ i.e., $(A) \odot (M)' + (U) \odot (M) \rightarrow A + 1$		4.7	8.7	SSU
44		Selective Even Parity Test	If $[(A) \odot (U)]$ is even parity, Skip NI	No Skip	6.0	10.0	SEP
				Skip	10.0	14.0	
45		Selective Odd Parity Test	If $[(A) \odot (U)]$ is odd parity, Skip NI	No Skip	6.0	10.0	SOP
				Skip	10.0	14.0	
47		Test Modifier	If $(B_a)_{17\text{-}0} < (U)$, take NI; If $(B_a)_{17\text{-}0} \geq (U)$, Skip. In either case, $(B_a)_{17\text{-}0} + (B_a)_{35\text{-}18} \rightarrow B_{a17\text{-}0}$	No Skip	4.7	8.7	TMO
				Skip	8.7	12.7	
50		Test Zero	Skip NI if $(U) = 0$	No Skip	4.0	8.0	TZR
				Skip	8.0	12.0	
51		Test Not Zero	Skip NI if $(U) \neq 0$	No Skip	4.0	8.0	TNZ
				Skip	8.0	12.0	
52		Test Equal	Skip NI if $(U) = (A)$	No Skip	4.0	8.0	TEQ
				Skip	8.0	12.0	
53		Test Not Equal	Skip NI if $(U) \neq (A)$	No Skip	4.0	8.0	TNE
				Skip	8.0	12.0	
54		Test Less Than or Equal	Skip NI if $(U) \leq (A)$	No Skip	4.0	8.0	TLE
				Skip	8.0	12.0	
55		Test Greater Than	Skip NI if $(U) > (A)$	No Skip	4.0	8.0	TGR
				Skip	8.0	12.0	
56		Test Within Limits	Skip NI if $(A) < (U) \leq (A + 1)$ (Note: $(A) < (A + 1)$)	No Skip	4.7	8.7	TWL
				Skip	8.7	12.7	
57		Test Outside Limits	Skip NI if $(U) \leq (A)$ or $(U) > (A + 1)$ (Note: $(A) \leq (A + 1)$)	No Skip	4.7	8.7	TOL
				Skip	8.7	12.7	

† All repeat operations (22, 62-67, 71) take 16 μ sec. combined setup and termination time.

INSTRUCTION REPERTOIRE

f	j	NAME	DESCRIPTION		EXECUTION TIME IN μ SEC. Alternate Core Banks	EXECUTION TIME IN μ SEC. Same Core Bank	MNEMONIC CODE
60	0-17	Test Positive	Skip NI if (U) $\geq$ 0	No Skip	4.0	8.0	TPO
				Skip	8.0	12.0	
61		Test Negative	Skip NI if (U) $<$ 0	No Skip	4.0	8.0	TNG
				Skip	8.0	12.0	
62†		Search Equal	Skip NI if $(U)_i = (A)$	No Skip	4.0	4.0	SEQ
			Repeated k times	Skip	4.0	4.0	
63†		Search Not Equal	Skip NI if $(U)_i \neq (A)$	No Skip	4.0	4.0	SNE
			Repeated k times	Skip	4.0	4.0	
64†		Search Less Than or Equal	Skip NI if $(U)_i \leq (A)$	No Skip	4.0	4.0	SLE
			Repeated k times	Skip	4.0	4.0	
65†		Search Greater Than	Skip NI if $(U)_i > (A)$	No Skip	4.0	4.0	SGR
				Skip	4.0	4.0	
66†		Search Within Limits	Skip NI if $(A) < (U)_i \leq (A+1)$	No Skip	4.7	4.7	SWL
			(Note: $(A) < (A+1)$)	Skip	4.7	4.7	
67†		Search Outside Limits	Skip NI if $(U)_i \leq (A)$ or $(U)_i > (A+1)$	No Skip	4.7	4.7	SOL
			(Note: $(A) < (A+1)$)	Skip	4.7	4.7	
70		Index Jump	If $(CM)_{ja} > 0$, Jump to U	No Jump	8.0	8.0	IXJP
			$(CM)_{ja} < 0$, Take NI	Jump	4.0	4.0	
			Then $(CM)_{ja} - 1 \rightarrow CM_{ja}$				
			NOTE: j in this instruction serves with the a-designator to specify any one of the 128 words of Control Memory.				
71†	*						
	00	Masked Search Equal	Skip NI if $(U)_i \odot (M) = (A) \odot (M)$	No Skip	4.0	4.0	MSEQ
			Repeated k times	Skip	4.0	4.0	
	01	Masked Search Not Equal	Skip NI if $(U)_i \odot (M) \neq (A) \odot (M)$	No Skip	4.0	4.0	MSNE
			Repeated k times	Skip	4.0	4.0	
	02	Masked Search Less Than or Equal	Skip NI if $(U)_i \odot (M) \leq (A) \odot (M)$	No Skip	4.0	4.0	MSLE
			Repeated k times	Skip	4.0	4.0	
	03	Masked Search Greater Than	Skip NI if $(U)_i \odot (M) > (A) \odot (M)$	No Skip	4.0	4.0	MSGR
			Repeated k times	Skip	4.0	4.0	
	04	Masked Search Within Limits	Skip NI if $(A) \odot (M) < (U)_i \odot (M) \leq (A+1) \odot (M)$	No Skip	4.7	4.7	MSWL
			(Note: $(A) \odot (M) < (A+1) \odot (M)$)	Skip	4.7	4.7	
			Repeated k times				
	05	Masked Search Outside Limits	Skip NI if $(U)_i \odot (M) \leq (A)$ or $(U)_i \odot (M) > (A+1)$	No Skip	4.7	4.7	MSOL
			(Note: $(A) \odot (M) < (A+1) \odot (M)$)	Skip	4.7	4.7	
			Repeated k times				
72	*						
	00	Wait for Interrupt	The computer program sequence stops (i.e., P is not advanced). The wait condition is removed by an interrupt.		4.0		WAIT
	01	Return Jump	$(P) \rightarrow U_{17-0}$ and Jump to U + 1		8.0	8.0	RTJP
	02	Positive Bit Control Jump	If $(A)_{35} = 0$, Jump to U	No Jump	4.0	4.0	PBJP
			Shift (A) left one in either case	Jump	8.0	8.0	
	03	Negative Bit Control Jump	If $(A)_{35} = 1$, Jump to U	No Jump	4.0	4.0	NBJP
			Shift (A) left one in either case	Jump	8.0	8.0	
	04	Add Halves	$(A)_{17-0} + (U)_{17-0} \rightarrow A_{17-0}$		4.0	8.0	ADDH
			$(A)_{35-18} + (U)_{35-18} \rightarrow A_{35-18}$				
	05	Subtract Halves	$(A)_{17-0} - (U)_{17-0} \rightarrow A_{17-0}$		4.0	8.0	SUBH
			$(A)_{35-18} - (U)_{35-18} \rightarrow A_{35-18}$				
	06	Add Thirds	$(A)_{35-24} + (U)_{35-24} \rightarrow A_{35-24}$		4.0	8.0	ADDT
			$(A)_{23-12} + (U)_{23-12} \rightarrow A_{23-12}$				
			$(A)_{11-0} + (U)_{11-0} \rightarrow A_{11-0}$				
	07	Subtract Thirds	$(A)_{35-24} - (U)_{35-24} \rightarrow A_{35-24}$		4.0	8.0	SUBT
			$(A)_{23-12} - (U)_{23-12} \rightarrow A_{23-12}$				
			$(A)_{11-0} - (U)_{11-0} \rightarrow A_{11-0}$				
	10	Execute Remote Instruction	Execute the Instruction at U		4.0	— + Execution Time	EXRI
	11	Load Memory Lockout Register	$U_{5-0} \rightarrow MLR$		4.0	—	LMLR
			For $U_0 = 1$ lockout 0—4095				
			$U_1 = 1$ lockout 4096—8191				
			$U_2 = 1$ lockout 8192—16383				
			$U_3 = 1$ lockout 16384—32767				
			$U_4 = 1$ lockout applies to 1st BANK				
			$U_5 = 1$ lockout applies to 2nd BANK				
73	*						
	00	Single Right Circular Shift‡	Shift (A) right U places circularly		4.0		SCSH
	01	Double Right Circular Shift	Shift (A, A + 1) right U places circularly		4.0		DCSH
	02	Single Right Logical Shift	Shift (A) right U places, end off; fill with zeros (Max. Shift = 36)		4.0		SLSH

*j serves as part of the Function Code

† All repeat operations (22, 62-67, 71) take 16 μ sec. combined setup and termination time.

‡Instruction execution time is independent of the number of shifts performed (e.g. a shift of 72 takes 4 microseconds). There are no memory references in the first six shift instructions, 73 00 — 73 05; consequently, the distinction between **alternate** core banks and the **same** core bank is irrelevant.

INSTRUCTION REPERTOIRE

f	j	NAME	DESCRIPTION		EXECUTION TIME IN μ SEC. Alternate Core Banks	EXECUTION TIME IN μ SEC. Same Core Bank	MNEMONIC CODE
	03	Double Right Logical Shift	Shift (A, A + 1) right U places, end off; fill with zeros. (Max. Shift = 72)		4.0		DLSH
	04	Single Right Arithmetic Shift	Shift (A) right U places, end off; fill with sign bits.		4.0		SASH
	05	Double Right Arithmetic Shift	Shift (A, A + 1) right U places, end off; fill with sign bits. (Max. Shift = 72)		4.0		DASH
	06	Scale Factor Shift	(U) → A, shift A left circularly until $A_{35} \neq A_{34}$ or until A has been shifted 36 times. Store the scaled quantity in A and the number of shifts that occurred in A + 1.		6.0	10.0	SFSH
74	*						
	00	Zero Jump	Jump to U if (A) = 0	No Jump	4.0	4.0	ZRJP
				Jump	8.0	8.0	
	01	Non-zero Jump	Jump to U if (A) $\neq$ 0	No Jump	4.0	4.0	NZJP
				Jump	8.0	8.0	
	02	Positive Jump	Jump to U if (A) $\geq$ 0	No Jump	4.0	4.0	POJP
				Jump	8.0	8.0	
	03	Negative Jump	Jump to U if (A) $<$ 0	No Jump	4.0	4.0	NGJP
				Jump	8.0	8.0	
	04	Console Selective Jump	Jump to U if A = key setting on console (1 of 15)		4.0	4.0	CSJP
	05	Selective Stop Jump	Stop if A = stop key setting on console (1 of 4), always jump to U		4.0	4.0	SSJP
	06	No Operation	Do Nothing; continue with NI		4.0	4.0	NOOP
	07	Enable All Input—Output Interrupts and Jump	Jump to U and permit interrupts to occur		4.0	4.0	EIJP
	10	Even Jump	Jump to U if $(A)_0 = 0$	No Jump	4.0	4.0	EVJP
				Jump	8.0	8.0	
	11	Odd Jump	Jump to U if $(A)_0 = 1$	No Jump	4.0	4.0	ODJP
				Jump	8.0	8.0	
	12	Modifier Jump	If $(B_a)_{17-0} > 0$, Jump to U	No Jump	4.0	4.0	MOJP
			If $(B_a)_{17-0} \leq 0$, Take NI	Jump	8.0	8.0	
			In either case $(B_a)_{17-0} + (B_a)_{35-18} \rightarrow B_{a17-0}$				
	13	Load Modifier and Jump	$(P) \rightarrow (B_a)_{17-0}$ and Jump to U		4.0	4.0	LMJP
	14	Overflow Jump	Jump to U if overflow cond. is set		4.0	4.0	OVJP
	15	No-Overflow Jump	Jump to U if overflow cond. is not set		4.0	4.0	NOJP
	16	Carry Jump	Jump to U if carry cond. is set		4.0	4.0	CYJP
	17	No-Carry Jump	Jump to U if carry cond. is not set		4.0	4.0	NCJP
75	*						
	00	Initiate Input Mode	(U) → input control word a, and initiate input mode on channel a.		4.0	8.0	IIPM
	01	Initiate Monitored Input Mode	(U) → input control word a, and initiate input mode on channel a with monitor.		4.0	8.0	IMIM
	02	Input Mode Jump	Jump to U if channel a is in the input mode.		4.0	4.0	IMJP
	03	Terminate Input Mode	Terminate input mode on channel a.		4.0	4.0	TIPM
	04	Initiate Output Mode	(U) → output control word a, and initiate output mode on channel a.		4.0	8.0	IOPM
	05	Initiate Monitored Output Mode	(U) → output control word a, and initiate output mode on channel a with monitor.		4.0	8.0	IMOM
	06	Output Mode Jump	Jump to U if channel a is in the output mode.		4.0	4.0	OMJP
	07	Terminate Output Mode	Terminate output mode on channel a.		4.0	4.0	TOPM
	10	Initiate Function Mode	(U) → output control word a, and initiate function mode on channel a.		4.0	8.0	IFNM
	11	Initiate Monitored Function Mode	(U) → output control word a, and initiate function mode on channel a with monitor.		4.0	8.0	IMFM
	12	Function Mode Jump	Jump to U if channel a is in the function mode.		4.0	4.0	FMJP
	13	Force External Transfer	Request external function or output word on channel a.		4.0	4.0	FEXT
	14	Enable All External Interrupts	All external interrupts are permitted to occur.		4.0	4.0	EAEI
	15	Disable All External Interrupts	All external interrupts are prevented from occurring.		4.0	4.0	DAEI
	16	Enable Single External Interrupt	An external interrupt on channel a is permitted to occur.		4.0	4.0	ESEI
	17	Disable Single External Interrupt	An external interrupt on channel a is prevented from occurring.		4.0	4.0	DSEI
76	*						
	00	Floating Add	(A) + (U) → A, A + 1		14.0	18.0	FLAD
	01	Floating Subtract	(A) − (U) → A, A + 1		14.0	18.0	FLSB
	02	Floating Multiply	(A) • (U) → A, A + 1		12.7	16.7	FLMP
	03	Floating Divide	(A) ÷ (U); Quotient → A		26.0	30.0	FLDV
			Remainder → A + 1				
	04	Floating Point Unpack	Unpack (U), store mantissa in A + 1 and store the biased characteristic in A		4.7	8.7	FLUP
	05	Floating Point Normalize Pack	Form the packed normalized number from mantissa stored in U and from biased characteristic in A, and store at A + 1		6.7	10.7	FLNP
	06	Floating Characteristic Difference Magnitude	Absolute value of $\lvert(A)_{34-27}\rvert - \lvert(U)_{34-27}\rvert \rightarrow A+1$		4.0	8.0	FLCM
	07	Floating Characteristic Difference	$\lvert(A)_{34-27}\rvert - \lvert(U)_{34-27}\rvert \rightarrow A+1$		4.0	8.0	FLCD

*j serves as part of the Function Code

...AND DELIVERS 8 ADVANCE FEATURES FOR A WIDE VARIETY OF APPLICATIONS

1. High-Speed Thin-Film Memory
   - 128 36-Bit Words of Control Memory
   - 600 Nanoseconds Cycle Time
2. Modular General-Purpose Computer
3. Compact Solid-State Circuitry
4. Large Core Memory
   - 16,384 Words to 65,536 Words
   - 2 Microseconds Effective Cycle Time
5. Powerful Instruction Repertoire
6. Versatile 36-Bit Word Length
7. Fixed (Integral & Fractional) and Floating-Point Arithmetic
8. Proven Peripheral Equipment
   - Magnetic Drums
   - Magnetic Tapes
   - Mass Storage
   - Punched Card
   - High-Speed Printer
   - Paper Tape
   - Displays And Plotters
   - Teletypewriters
   - Real-Time Equipment

UNIVAC 1107 Thin-Film Memory Computer offers ALGOL (Algorithmic Language) with a FORTRAN Translator and COBOL (Common Business Oriented Language). These programs permit you to instruct the computer in simplified algebraic or English language. ALGOL and COBOL reduce programming time and give a common ground for programming and communication between all EDP users – businessmen, engineers, scientists.

UNIVAC 1107 ...A FEW TYPICAL APPLICATIONS

SCIENTIFIC COMPUTATIONS

DATA REDUCTION AND ANALYSIS

DIGITAL COMMUNICATIONS AND SWITCHING SYSTEMS

TACTICAL DATA AND CONTROL SYSTEMS

SIMULATION APPLICATIONS

LOGISTICS AND INTELLIGENCE SYSTEMS

TRAFFIC CONTROL

RESERVATION SYSTEMS

INVENTORY & SCHEDULING SYSTEMS

Remington Rand Univac
DIVISION OF SPERRY RAND CORPORATION