SPERRY RAND

UNIVAC

418-III

REAL-TIME SYSTEM

PROGRAMMERS HARDWARE AND SOFTWARE REFERENCE

	ASSEMBLER CODE	OCTAL CODE	TYPE	SEE ②	OPERATION	TIMING IN μSEC
ARITHMETIC	AL	14	Y	I,S	(AL) + (Y) → AL	1.5
	AA	20	Y	I,S	(A) + (Y−1,Y) → A	3.0
	ALK	71	Z	– –	(AL) + Z → AL	1.0
	ANL	16	Y	I,S	(AL) − (Y) → AL	1.5
	ANA	22	Y	I,S	(A) − (Y−1,Y) → A	3.0
	M	24	Y	I,S	(AL) * (Y) → A	6.5
	D	26	Y	I,S	(A) ÷ (Y) → AL; REMAINDER → AU	6.5
	RND	5060	--	——	IF (A) POSITIVE AND (AL_{17}) = 1, (AU) + 1 → AL	1.75
					IF (A) NEGATIVE AND (AL_{17}) 0, (AU) − 1 → AL, OTHERWISE (AU) → AL	
	① FA	5002	Y	I,S	(A) + (Y−1,Y) → A	4 + X'8
	① FS	5003	Y	I,S	(A) − (Y−1,Y) → A	4 + X'8
	① FM	5004	Y	I,S	(A) * (Y−1,Y) → A	11.0
	① FD	5005	Y	I,S	(A) ÷ (Y−1,Y) → A	11.0
LOGICAL	AND	52	Y	——	(AL) **AND** (Y) → AL [SET AL_N = 0 FOR (Y_N = 0)]	1.5
	OR	51	Y	——	(AL) **OR** (Y) → AL [SET AL_N = 1 FOR (Y_N = 1)]	1.5
	XOR	53	Y	——	(AL) **XOR** (Y) → AL [COMPLEMENT AL_N FOR (Y_N = 1)]	1.5
	CPL	5061	–	——	−(AL) → AL [1'S COMPLEMENT → AL]	1.0
	CPU	5062	–	——	−(AU) → AU [1'S COMPLEMENT → AU]	1.0
	CPA	5063	–	——	−(A) → A [1'S COMPLEMENT → A]	1.75
TRANSFER	LU	10	Y	I,S	(Y) → AU	1.5
	LL	12	Y	I,S	(Y) → AL	1.5
	SU	46	Y	I,S	(AU) → Y	1.5
	SL	44	Y	I,S	(AL) → Y	1.5
	LLK	70	Z	——	(Z) → AL [Z IS SIGN EXTENDED TO AL]	1.0
	MSL	04	Y	I,S	(Y_N) → AL_N FOR (AU_N) = 1	1.5
	SAD	74	Y	——	($AL_{11\text{-}0}$) → $Y_{11\text{-}0}$	2.5
	LB	32	Y	I,S	(Y) → IR	1.5
	SB	42	Y	I,S	(IR) → Y	.75
	LBK	36	Z	I	Z → IR	.75
	LIR	5072	K	——	$K_{2\text{-}0}$ → IRP	2.5
	SIR	72	Y	——	$IRP_{3\text{-}0}$ → $Y_{3\text{-}0}$; 0 → $Y_{5,4}$ IF (IRP) = 0, Y_3 = 1 IF (IRP) ≠ 0, Y_3 = 0	2.5
	LSR	5073	K	——	$K_{5\text{-}0}$ → $SR_{5\text{-}0}$	1.0
	LSD	5020	—		((IAR) + $1)_{5\text{-}0}$ → $SR_{5\text{-}0}$((IAR) + $1)_{10}$ → OVERFLOW DESIGNATOR	2.5
					((IAR) + $1)_{11}$ → BORROW DESIGNATOR	
	SSR	75	Y	——	(SR) → $Y_{5\text{-}0}$; 0 → SR_4	2.5
	SSD	5017			(SR) → ((IAR) + $1)_{5\text{-}0}$ OVERFLOW DESIGNATOR → ((IAR) + $1)_{10}$	
					BORROW DESIGNATOR → ((IAR) + $1)_{11}$; SR_4, BORROW, OVERFLOW ARE SET TO 0	2.5
	CY	40	Y	I,S	0 → Y	1.5
	① FP	5006	Y	I,S	(A) NORMALIZED (MANTISSA) → $A_{35}A_{26\text{-}0}$	3 + X'8
					IF A_{35}=0, ($Y_{7\text{-}0}$) → $A_{34\text{-}27}$ IF A_{35}=1, ($Y_{7\text{-}0}$) → $A_{34\text{-}27}$	
	① FU	5007	Y	I,S	IF A_{35} = 0, ($A_{34\text{-}27}$) → $Y_{7\text{-}0}$ 0's → $Y_{17\text{-}8}$; A_{35} → $A_{34\text{-}27}$	3.0
					IF A_{35} = 1, ($A_{34\text{-}27}$) → $Y_{7\text{-}0}$, 0's → $Y_{17\text{-}8}$; A_{35} → $A_{34\text{-}27}$	
	① DB	5074	–	——	($AU_{15\text{-}12,\ 9\text{-}6,\ 3\text{-}0}$) DECIMAL VALUE → $AL_{9\text{-}0}$ AS A BINARY VALUE	9.0
	① BD	5075	–	——	$(AL)_{9\text{-}0}$ BINARY → $AU_{15\text{-}12,\ 9\text{-}6,\ 3\text{-}0}$ AS DECIMAL VALUE	9.0
	BT	5070	K	——	(Y_{AU}) → (Y_{AL}), (AU) + 1 → AU, (AL) + 1 → AL; REPEAT K TIMES	1 + 1.5K
MISC.	RS	5010	Y	I,S	(Y) → AL, 1 → Y_{17} [FINAL (AL) = INITIAL (Y)]	2.5
	NOP	5026	Y	——	(IAR) + 1 → IAR	1.0
COMP.	CL	02	Y	I,S	ACTIVATE CD; (AL) < (Y), SET CD LESS; (AL) = (Y), SET CD EQUAL; (AL) > (Y)	1.5
					CD NOT SET EQUAL OR LESS	
	CLM	06	Y	I,S	(AU) **AND** (AL) : (AU) **AND** (Y) SET CD SAME AS THE COMPARE LOWER	1.75
COND. JUMP CD ACTIVE	JE	60,61	Y	——	IF CD SET TO = CONDITION Y → IAR [(AL) WAS = (Y)]	.75
	JNE	62,63	Y	——	IF CD NOT SET TO = CONDITION Y → IAR [(AL) WAS ≠ (Y)]	.75
	JLS	66,67	Y	——	IF CD SET TO < CONDITION Y → IAR [(AL) WAS < (Y)]	.75
	JNLS	64,65	Y	——	IF CD NOT SET TO < CONDITION Y → IAR [(AL) WAS ≥ (Y)]	.75
COND. JUMP CD NOT ACTIVE	JUZ	60	Y	——	IF (AU) = 0 } Y → IAR	.75
	JUNZ	62	Y	——	IF (AU) ≠ 0 } Y → IAR	.75
	JUP	64	Y	——	IF (AU) IS POSITIVE } Y → IAR	75
	JUN	66	Y	——	IF (AU) IS NEGATIVE } Y → IAR	.75
	JLZ	61	Y	——	IF (A) IS = 0 } Y → IAR	75
	JLNZ	63	Y	——	IF (AL) ≠ 0 } Y → IAR	.75
	JLP	65	Y	——	IF (AL) IS POSITIVE } Y → IAR	.75
	JLN	67	Y	——	IF (AL) IS NEGATIVE } Y → IAR	.75

	ASSEMBLER CODE	OCTAL CODE	TYPE	SEE ②	OPERATION	TIMING IN μSEC
UNCOND. JUMP	J	34	Y	I–	Y → IAR	.75
	JI	55	Y	––	$(Y_{16\text{-}0})$ → IAR	1.5
	EJI	54	Y	––	ENABLE INTERRUPTS; $(Y_{16\text{-}0})$ → IAR	1.5
	SLJ	76	Y	––	(IAR) + 1 → Y; Y + 1 → IAR	1.75
	SLJI	30	Y	I –	(IAR) + 1 → (Y); $(Y)_{16\text{-}0}$ + 1 → IAR	2.25
SKIP	TK	5050	K	––	IF KEYS DESIGNATED BY K ARE SET, (IAR) + 2 → IAR	1.0
	TNB	5051	–	––	IF BORROW INDICATOR OFF	1.0
	TOF	5052	–	––	IF OVERFLOW INDICATOR ON	1.0
	TNO	5053	–	––	IF OVERFLOW INDICATOR OFF } (IAR) + 2 → IAR	1.0
	TOP	5054	–	––	IF SUM OF 1'S IN (AU) **AND** (AL) IS ODD	2.0
	TEP	5055	–	––	IF SUM OF 1'S IN (AU) **AND** (AL) IS EVEN	2.0
SHIFT	SRU	5041	K	––	SHIFT RIGHT AU	1.25 + X/8
	SRL	5042	K	––	SHIFT RIGHT AL } K BIT POSITIONS; END-OFF	1.25 + X/8
	SRA	5043	K	––	SHIFT RIGHT A	1.25 + X/8
	SLU	5045	K	––	SHIFT LEFT AU	1.25 + X/8
	SLL	5046	K	––	SHIFT LEFT AL } K BIT POSITIONS; END-AROUND	1.25 + X/8
	SLA	5047	K	––	SHIFT LEFT A	1.25 + X/8
	SCA	5044	K	––	SHIFT LEFT (END-AROUND) K BIT POSITIONS OR UNTIL NORMALIZED; K LESS ACTUAL SHIFT 000017_8	2.0 + X/8
LOOP CONTROL	JBNZ	73	Y	––	IF (IR) ≠ 0, (IR) − 1 → IR, Y → IAR; IF (IR) = 0, (IAR) + 1 → IAR	1.75
	TB	56	Y	––	IF (IR) = (Y), (IAR) + 2 → IAR; IF (IR) ≠ (Y), (IR) + 1 → IR, (IAR) + 1 → IAR	2.25
	TZ	57	Y	––	IF (Y) = 0, (IAR) + 2 → IAR; IF (Y) ≠ 0, (Y) − 1 → Y (IAR) + 1 → IAR	2.25
INTERRUPT CONTROL	PAI	5034 5035	–	––	PREVENT ALL IOM GENERATED INTERRUPTS	1.0
	AAI	5030 5031	–	––	REMOVE LOCKOUT ON ALL IOM GENERATED INTERRUPTS	1.0
PRIVILEGED	LGM	5065	–	––	$((IAR) + 1)_{17-9}$ → UPPER STORAGE LIMIT $((IAR) + 1)_{8-0}$ → LOWER STORAGE LIMIT, GUARD MODE IS SET AND (IAR) + 2 → IAR	1.75
	LIC	5011	K	––	LOAD I/O CHANNEL K FROM (IAR) + 1 AND (IAR) + 2, INITIATE INPUT, (IAR) + 1 → IAR	4.0
	LCC	5012	K	––	LOAD I/O CHANNEL K FROM (IAR) + 1 AND (IAR) + 2, INITIATE OUTPUT, (IAR) + 3 → IAR	4.0
	LFC	5013	K	––	LOAD I/O CHANNEL K FROM (IAR) + 1 AND (IAR) + 2, INITIATE EXTERNAL FUNCTION, (IAR) + 3 → IAR	4.0
	STIC	5015	K	––	STOP INPUT ON CHANNEL K	1.75
	STOC	5016	K	––	STOP OUTPUT ON CHANNEL K	1.75
	TIC	5021	K	––	TEST INPUT CHANNEL K FOR IDLE, IF IDLE SKIP NEXT INSTRUCTION	1.0
	TOC	5022	K	––	TEST OUTPUT CHANNEL K FOR IDLE, IF IDLE SKIP NEXT INSTRUCTION	1.0
	TFC	5023	K	––	TEST EXTERNAL FUNCTION CHANNEL K FOR IDLE, IF IDLE SKIP NEXT INSTRUCTION	1.0
	WFI	5024			STOP C/A (BUT NOT I/O) UNTIL INTERRUPT	1.0
		5025				1.0
	EEI	5067	K	––	ENABLE ESI INTERRUPT IN IOM #0 IF K_4 = 0 IN IOM #2 IF K_4 = 1	1.0
	SK	5056	K	––	IF IN GUARD MODE, (IAR) + 1 → IAR, OTHERWISE STOP IF KEYS SPECIFIED BY K SET	1.0
	SAA	5066	–	––	SET AUDIBLE ALARM	

AL LOWER ACCUMULATOR
AU UPPER ACCUMULATOR
A UPPER & LOWER ACCUMULATORS
IR THE ACTIVE INDEX REGISTER
B SAME AS IR
IAR INSTRUCTION ADDRESS REGISTER
CD COMPARE DESIGNATOR
IRP INDEX REGISTER POINTER, INDICATES B REGISTER
K THE LOW-ORDER 6 BITS OF THE INSTRUCTION
* MULTIPLICATION OPERATOR
( ) CONTENTS OF THE LOCATION SPECIFIED IN THE PARENTHESES
Z THE LOW-ORDER 12 BITS OF THE INSTRUCTION SIGN EXTENDED TO 18 BITS
Y THE 17-BIT STORAGE ADDRESS, CONSISTS OF 12 LOW-ORDER BITS FROM U IN THE INSTRUCTION AND 5 HIGH-ORDER BITS FROM $IAR_{16\text{-}12}$ OR $SR_{5,3-0}$
→ REPLACES THE CONTENTS OF
: COMPARE ALGEBRAICALLY
AND LOGICAL AND
OR INCLUSIVE OR
XOR EXCLUSIVE OR
X NUMBER OF SHIFTS
{ } = ADDITIONAL DESCRIPTION

① OPTIONAL HARDWARE
② –– Y = $(IAR_{16\text{-}12}), (U_{11\text{-}0})$
I = INDEXABLE
Y = $(IAR_{16\text{-}12}), (U_{11\text{-}0}) + (IR_{16\text{-}0})$
S = SR SENSITIVE
I,S Y = $(SR_{5,3\text{-}0}), (U_{11\text{-}0}) + (IR_{16\text{-}0})$

RTOS RUN CONTROL CARDS

GENERAL FORMAT: ∇FUNCTION, OPTIONS $E_1,E_2,E_3,\ldots,E_n$

NOTE: Column one must contain a 7/8 multipunch

FUNCTION	OPTIONS	LIST	DESCRIPTION
RUN	None	E_1 Six character run identification. E_2 Six character account number.	First card of a RUN (mandatory).
XQT	A-Ignore undefined references	Six character program name.	Load and execute main program relocatable element NAME from program library.
FIN	None	None	Indicates end of the RUN (mandatory).
EOF	None	None	May be used at user's discretion to seperate data card files.
EOD	None	None	Indicates end of card data. Follows last run file in job.
ASG	None	Indeterminate (E_1, E_2, . . . E_n) E_i has the following form: LU = NAME, LU = LENGTH, or LU	Assign tape file NAME to RUN referenced as unit LU. Assign temporary mass storage file of length LNGTH to RUN referenced as file number LU. Assign scratch tape (Label '*') to LU. Magnetic Tape = 0–15 High Speed Drum = 20–35 FASTRAND mass storage = 40–45
CAT	D – Delete NAME from Master File Directory V – Variable length records T – Magnetic Tape file R – Random file S – Standard labels M – Multi-reel file	E_1 Six character file NAME. E_2 Six character account number. E_3 Number of SAVE days. E_4 Number of FASTRAND tracks to be allocated (may be blank). E_5 Number of words per record (NWPR). E_6 Number of records per block (BF) may be blank.	Catalogue NAME file in the Master File Directory under ACCT account number. Remove NAME file from catalogue automatically after SAVE days. Allocate L tracks of FASTRAND mass storage for FILE. If L is omitted, file is magnetic tape. File format is fixed length of NWPR and BF. If BF is omitted, format is NWPR words per block.
ELT	S – Symbolic element R – Relocatable element M – Map element P – Procedure element	Six character element identification.	Enter element into users program library from RUN file.
IN	L – List element and corrections N – Do not list corrections	E_1 Logical tape unit number of input tape (0–15). E_2 Element type (T) may be blank. R – relocatable S – symbolic P – procedure M – map E_3 Library or element NAME (may be blank). E_4 through E_7: Logical tape unit numbers for output (0–15).	Enter NAME and element of type T into user program library from input logical tape unit E_1. Output the element to E_4,E_5,E_6, and E_7 if present. If T is blank all elements NAME are entered. If NAME is blank, all elements T are entered. If both T and NAME are blank, all elements are entered into the user program library.

FUNCTION	OPTIONS	LIST	DESCRIPTION
OUT	L – List element If option omitted, list control cards only.	E_1 Logical tape unit number of output tape (0–15). E_2 Element type (T): same as for IN card. E_3 Library or element NAME.	Output NAME element of type T from users program library to logical tape unit LU file. If type is omitted, all types of NAME are output. If NAME is omitted, all elements in the users program library are output.
COPY	L – List symbolic element and corrections N – Do not list corrections	E_1 Logical tape unit number of input tape (0–15). E_2 Element type (T): same as for IN card. E_3 Library or element NAME. E_4through E_7: Logical tape unit numbers for output (0–15).	Copy NAME element of type T from logical tape unit E_1 file to E_4,E_5, E_6, or E_7 files. If NAME is omitted, copy elements of type T. If element type is omitted, copy elements of all types. If both NAME and T are omitted, copy the entire file
DEL	None	E_1 Element type (T): Same as for IN card. E_2 Library or element NAME.	Delete NAME element of type T from users program library. If T is omitted, delete all NAME elements. If NAME is omitted, delete all type T. If both NAME and T are omitted, clear the users program library.
TRW	None	E_1through E_4: Logical tape unit to be rewound (0–15).	Rewind E_1 through E_4. If tape file has been referenced as an output file on an IN, OUT, or COPY card; write tape end-of-file before rewind. Release the tape unit.
TRI	None	E_1through E_4: Logical tape unit to be rewound with interlock (0–15).	Rewind with interlock E_1 through E_4. If tape file has been referenced as an output file on an IN, OUT, or COPY card; write tape end-of-file before rewind with interlock. Release the tape unit.
ASM	* – Symbolic element NAME is the users program library N – No listing desired. Errors and external references are listed. P – Punch a relocatable card element. If the * is omitted, element NAME follows in the RUN file.	Library or element NAME.	Assemble source element NAME and write relocatable output element NAME into users program library replacing NAME element if present.
FOR	Same as for ASM card with addition of: L – List source and object code	Library or element NAME.	Compile source element NAME and write relocatable output element NAME into users program library replacing NAME element if present.
COB	Same as for ASM card with addition of: L – List source and object code	Library or element NAME.	Compile source element NAME and write relocatable output element NAME into users program library replacing NAME element if present.

RTOS DISPATCHER REQUESTS

CALLING SEQUENCE		CODE GENERATED	DESCRIPTION
EXRET$		+0770000	Release control to RTOS. No return. Program must ensure that control will return where required.
TOD$		+0770100	RTOS returns control to CALL +1 with environment saved (except AU and AL). (AU) = time of day in seconds. (AL) = julian date in decimal. (AL$_{15-12}$) = Y. (AL$_{11-0}$) = DDD.
LOAD$	To be supplied.		
EOJ$		+0770300	Normal job termination request.
ERROR$		+0770301	Terminate job with postmortem dump.
ABORT$		+0770302	Terminate the run.
GETB$	NO,T	+0770400 + T +NO	Allocate NO contiguous blocks of storage. Block is 256 words. Start search at bay T. Status; (AL)=0, core not available. If NO$\geq$ 16 storage will be allocated on bay boundary. If NO$<$16 storage will be allocated within a bay. Return to CALL+ 2 with environment saved (except AL).
RELB$	ADDR,T	+0770500 + T +ADDR	Block is 256 words. First block released includes ADDR. If ADDR is zero generated, all blocks assigned by GETB$ are released. T = 0: entire buffer released; T = 1: buffer from block following ADDR to end released; T = 2: buffer from start to block following ADDR released. Return to CALL+2 with environment saved.
PTG$	ADDR,MID,IRL	+0770600 + MID * 16 + IRL + ADDR	ADDR is scheduled as a place-to-go at MID priority.
WPTG$	ADDR,MID,IRL	+0770700 + MID * 16 + IRL +ADDR	ADDR is scheduled as a place-to-go at MID priority, but will be bypassed once by RTOS.
UPTG$	ADDR,MID,IRL	+0771000 + MID * 16 + IRL + ADDR	ADDR is scheduled as a place-to-go at MID priority by an unsolicited console type-in. AU, AL will then contain USTI parameter.
RTPTG$	ADDR,MID,IRL H,M,S	+0771100 + MID * 16 + IRL +ADDR +H * 360 + M * 60 + S	Address is scheduled as a place-to-go at MID priority at H.M.S. time of day.

CALLING SEQUENCE	CODE GENERATED	DESCRIPTION
ETPTG$ ADDR,MID,IRL SEC;mSEC	+0771200 + MID * 16 + IRL +ADDR ③ +(SEC > 26) * (0400000 + SEC) +(SEC < 27) * (SEC*5000 + mSEC*5)	ADDR is scheduled as a place-to-go at MID priority after SEC, mSEC has elapsed. If (CALL + 2) ≥ 0, delay is +(CALL + 2)/5 mSEC. If (CALL + 2) < 0, delay is (CALL + 2)**0377777 SEC.
SPTG$ ADDR,T	+0771300 + T +ADDR	Strike one previous request to schedule ADDR as a place-to-go. I = 1 (AL) also compared. Sequence: UPTG$, PTG$, WPTG$, ETPG$, and RTPTG$. Return to CALL + 2, with environment saved.
TREQ$ LU,NAM,E	+0771400 + LU +'NAM' +'E '	Assign magnetic tape file identified by NAME to logical tape unit LU. Status (AL = 0), assignment not possible. (AL ≠ 0), the requested assignment is completed. Return to CALL + 2 with environment saved. 0 ≤ LU ≤ 15.
TREL$ LU	+0771500 + LU	Release magnetic tape file assigned to LU after rewind with interlock. Return to CALL + 1 with environment saved.
TSW$ LU1,LU2	+0771600 +LU1 +LU2	Assign magnetic tape file LU1 to LU2. Assign magnetic tape file LU2 to LU1.
MOVE$ ADDR1,ADDR2,NO	(AU) = ADDR1 (AL) = ADDR2 (B1) = NO +0772100	For each parameter present, the assembly procedure loads the corresponding register. If NO is present, B1 is set active. Return to CALL + 1, AU = (AU) + (B1), AL: (AL) + (B1), (B1) = 0 and remainder of environment saved.
SNAP$ ADDR1,ADDR2,CODE	+0772200 + CODE +ADDR1 +ADDR2	Print environment and memory from ADDR1 to ADDR2. If code is zero, octal conversion is assumed. If ADDR1 and ADDR2 are zero, environment is dumped only. If ADDR1 > 0 and ADDR2 = 0, all storage space assigned to program is dumped.
DGET$ To be supplied.		
DREL$ To be supplied.		
CEPTG$ To be supplied.		

③ ONE CONTINUOUS LINE

UNIVAC
418-III

**HARDWARE AND SOFTWARE
REFERENCE
UP-7659**

UNIVAC SYSTEMS PROGRAMMING LIBRARY SERVICES **RELEASE**

UNIVAC 418-III Real-Time System Library Memo 2 announces the release and availability of "UNIVAC 418-III Real-Time System Programmers Hardware and Software Reference," UP-7659, 6 pages plus covers. This is a Standard Library Item (SLI).

This document is an easily referenced summary of the repertoire, control cards, and dispatcher requests. The information is arranged in a columnar format. Missing and additional information will be added.

Distribution of this booklet, UP-7659, is being made as indicated below. Additional copies of UP-7659 may be requisitioned from Holyoke, Massachusetts, via a Sales Help Requisition through your local Univac Manager.

The following is a recap of current items for the 418-III System:

UP Number		Release Date
UP-7612	418-III System Binder	June 7, 1968
UDI-723 (Rev.12-67)	Set - Template and Instruction Booklet	March 29, 1968
UDI-1604	Vocabulary for Information Processing	October 24, 1968
UP-3910.5	Software System Field Report Procedures	May 11, 1964
UP-3910.10	General Software Category Report	March 29, 1968
UP-7503 SERIES	Set of 18 Tabs for COBOL Series	June 16, 1967
UP-7503 Rev. 1	Fundamentals of COBOL - Contents Section	December 23, 1968
UP-7503.1 Rev. 1	Fundamentals of COBOL - Language	December 23, 1968
UP-7503.2 Rev. 1	Fundamentals of COBOL - Table Handling	December 23, 1968
UP-7503.3 Rev. 1	Fundamentals of COBOL - Sorting	December 23, 1968
UP-7513	Binder for COBOL	June 16, 1967
UP-7515	Binder for FORTRAN	January 12, 1968
UP-7536	Fundamentals of FORTRAN (with 10 Index Tabs)	January 12, 1968
UP-3910.11	Update "A" to Fundamentals of FORTRAN, UP-7536	September 20. 1968
UP-3910.12	Update "B" to Fundamentals of FORTRAN, UP-7536	October 14, 1968
UP-7641	Assembler Programming Form	October 4, 1968
UP-7576	System Description	June 4, 1968
UP-7599	Assembler Programmers Reference	June 24, 1968
UP-7659	Programmers Hardware and Software Reference	

MANAGER
Documentation and Library Services

TO LISTS:	ATTACHMENTS: UP-7659 plus Library Memo 2 to Lists 10U, 217, 630, 692 and S.P.L.S. Lists 57 and 58.	UNIVAC 418-III Real-Time System Library Memo 2 December 20, 1968

UNIVAC

SPERRY RAND