Univac LARC Instruction Codes

by
R. Tim Coslet
R.TimCoslet (at) gmail.com

05/08/2019

Computer Instruction Word Format: `TIIAABBMMMMM`

Processor Instruction Word Format: `IINNNNNMMMMM`

`T`	= trace
`II`	= instruction opcode
`AA`	= arithmetic register
`BB`	= index register
`NNNNN`	= memory address, control fields, registers, device/synchronizer selects, etc.
`MMMMM`	= memory address

+, -, ✕, ÷	= fixed point
⊕, ⊗, ⊝	= floating point
'	= double word
| |	= absolute value or magnitude
[ ] Rdd	= rounded result
0500000[ ]	= extend a short number to 12 digits by prepending specified constant digits
00[ ]00000	= extend a short number to 12 digits by appending specified constant digits
{ }	= clarifying comments

Instruction description pseudocode syntax loosely adapted from Ada.

Table colors loosely adapted from "Moore Business Forms, Inc FORM 1488TG SPEEDIREAD WITH EYE TRACK" printer paper, which was used on the Univac 1106 and Univac 1110 that I ran programs on in college.

Computer Instruction[1]		Opcode	Processor Instruction[2]
SK	‖ Skip: (C) + 1 → C {nop}	0 0	Skip: (C) + 1 → C {nop}
AX	‖ (M) + (A) → A; (C) + 1 → C	0 1	Field NNNNN is xxDxx, case D is when 1..4 => If visual-display register D interlock FF set then M → C {jump} else (rP1) → visual-display register D; (C) + 1 → C end if when others => {undefined} end case
A	‖ (M) ⊕ (A) → A; (C) + 1 → C	0 2	Field NNNNN is xxDxx, case D is when 1..4 => (visual-display register D) → rP1; reset alert FF and interlock FF for visual-display register D; (C) + 1 → C when others => {undefined} end case
AM	‖ |(M)| ⊕ (A) → A; (C) + 1 → C	0 3	Field NNNNN is xxDxx, case D is when 1..4 => If visual-display register D interlock FF set then M → C {jump} else (C) + 1 → C end if when others => {undefined} end case
AU	‖ (M) ⊕ (A) → A+1; (C) + 1 → C	0 4	Illegitimate Instruction, Stall Error
AAX	‖ (M') + (A') → A'; (C) + 1 → C	0 5	M → C {jump}
AA	‖ (M') ⊕ (A') → A'; (C) + 1 → C	0 6	Illegitimate Instruction, Stall Error
		0 7	Field NNNNN is nnxxx, case n is when 0..29 => Shift rP1 right circular n places; (C) + 1 → C when others => {undefined} end case

1 Instruction Mnemonics shown for Computer Instructions. See documents: LARC Computing-Unit Instructions, Univac® LARC Programming the Computing Unit, An Introduction to the LARC® Data-Processing System. Behavior of Illegitimate Computer Instructions does not seem to be defined in these documents.
2 Instruction Mnemonics not shown for Processor Instructions. Univac may not have documented them as they provided a standard Processor program that the Computers could request services from as needed. See documents: Univac® LARC Processor Instruction and Function-Signal Analyses, An Introduction to the LARC® Data-Processing System. Illegitimate Processor Instructions result in either a Stall Error or a Timeout Error, stopping the Processor.

Computer Instruction		Opcode	Processor Instruction
		0 8	Field NNNNN is nnxxx, case n is when 0..29 => Shift rP1 & rP2 double right circular n places; (C) + 1 → C when others => {undefined} end case
FV	‖ If interlock FF reset then M → C {jump} else (5DD) → 02650; (02650) → A; reset connect FF and interlock FF; (C) + 1 → C end if	0 9	(rP1) + (rP2) → rP1; (rP1) → M; (C) + 1 → C
		1 0	(N) → rP2; (rP1) + (rP2) → rP1; (C) + 1 → C
NX	‖ -(M) + (A) → A; (C) + 1 → C	1 1	(N) → rP2; (rP1) - (rP2) → rP1; (C) + 1 → C
N	‖ -(M) ⊕ (A) → A; (C) + 1 → C	1 2	(N) → rP2; (C) + 1 → C
		1 3	(rP2) → N; (C) + 1 → C
NU	‖ -(M) ⊕ (A) → A+1; (C) + 1 → C	1 4	05_00000_[(C) + 2] → N; M → C {jump}
NNX	‖ -(M') + (A') → A'; (C) + 1 → C	1 5	(N) → rP1; (C) + 1 → C
NN	‖ -(M') ⊕ (A') → A'; (C) + 1 → C	1 6	(rP1) → N; (C) + 1 → C
		1 7	(N) → rP1; (rP1) + (rP2) → rP1; (rP1) → M; (C) + 1 → C
		1 8	(N) → rP2; If (rP1) = (rP2) then M → C {jump} else (C) + 1 → C end if
FVK	‖If interlock FF reset then M → C {jump} else (12DD) → 02650; (02650) → A; reset connect FF and interlock FF; (C) + 1 → C end if	1 9	(N) → rP2; If (rP1) > (rP2) then M → C {jump} else (C) + 1 → C end if
MXR	‖[(M) x (A)] Rdd → A; (C) + 1 → C	2 0	Extract where digits of (pR2) is even (i.e. +, 0, 2, 4, 6, 8, ., \) corresponding digits of (N) replacing those digits of (pR1); (pR1) → M; (C) + 1 → C
MXE	‖(M) x (A) → A'; (C) + 1 → C	2 1	Set CU 1 interlock FF in the 2500 word

Computer Instruction		Opcode	Processor Instruction
			memory unit containing N; (C) + 1 → C
MR	‖[(M) ⊗ (A)] Rdd → A; (C) + 1 → C	2 2	Set CU 2 interlock FF in the 2500 word memory unit containing N; (C) + 1 → C
M	‖(M) ⊗ (A) → A; (C) + 1 → C	2 3	Reset CU 1 interlock FF and CU 2 interlock FF in the 2500 word memory unit containing N; (C) + 1 → C
MU	‖(M) ⊗ (A) → A+1; (C) + 1 → C	2 4	(N) → rP2; 000000[odd numbered digit positions (i.e. "digit") discarding even numbered digit positions (i.e. "zone") of the alphanumeric characters of (rP2)] → rP1; (C) + 1 → C
ME	‖(M) ⊗ (A) → A'; (C) + 1 → C	2 5	(N) → rP2; If for every 1 in the three quinary bits of every digit of (rP1) there is also a 1 in (rP2) then M → C {jump} else (C) + 1 → C end if
MMX	‖(M') x (A') → A'; (C) + 1 → C	2 6	Illegitimate Instruction, Stall Error
MM	‖(M') ⊗ (A') → A'; (C) + 1 → C	2 7	Illegitimate Instruction, Stall Error
		2 8	Illegitimate Instruction, Stall Error
SV	‖If interlock FF set then M → C {jump} else (A) → 02650; (02650_M) → 5DD; (C) + 1 → C end if	2 9	Illegitimate Instruction, Stall Error
DX	‖(A) ÷ (M) → A; (C) + 1 → C	3 0	Field NNNNN is xxExx, case E is when 0..3 => If any error FFs specified by E set then M → C {jump} else (C) + 1 → C end if when others => {undefined} end case
DXE	‖(A) ÷ (M) → A, Remainder → A+1; (C) + 1 → C	3 1	Field NNNNN is xxxDD, case D is when 1..24 => If action FF of drum D set then M → C {jump}; reset action FF of drum D else (C) + 1 → C

Computer Instruction	Opcode	Processor Instruction
		end if when others => {undefined} end case
DR ‖[(A) ⊕ (M)] Rdd → A; (C) + 1 → C	3 2	Field NNNNN is xxxDD, case D is when 1..24 => Set action FF of drum D; (C) + 1 → C when others => {undefined} end case
	3 3	Illegitimate Instruction, Timeout Error
DUR ‖[(A) ⊕ (M)] Rdd → A+1; (C) + 1 → C	3 4	Illegitimate Instruction, Timeout Error
DDX ‖(A') ÷ (M') → A'; (C) + 1 → C	3 5	If previously selected drum is on local control then M → C {jump} else step head of drum; (C) + 1 → C; about 50ms after step starts set action FF of drum end if
DD ‖(A') ⊕ (M') → A'; (C) + 1 → C	3 6	If previously selected drum step direction is forward then start to reverse step direction relay of drum; M → C {jump}; 10ms after reverse starts set action FF of drum else (C) + 1 → C end if
DSE ‖(A') ⊕ (M) → A'; (C) + 1 → C	3 7	If previously selected drum step direction is backward then start to reverse step direction relay of drum; M → C {jump}; 10ms after reverse starts set action FF of drum else (C) + 1 → C end if
	3 8	If previously selected drum is on local control then M → C {jump} else jog head of drum to high positions; (C) + 1 → C; about 50ms after jog starts set action FF of drum end if
SVK ‖If interlock FF set	3 9	If previously selected drum is on local

Computer Instruction	Opcode	Processor Instruction
then M → C {jump} else (A) → 02650; (02650) → 12DD; (C) + 1 → C end if		control then M → C {jump} else jog head of drum to low positions; (C) + 1 → C; about 50ms after jog starts set action FF of drum end if
S ‖(A) → M; (C) + 1 → C	4 0	Field NNNNN is xxSxx, case S is when 0..4 => If bad-band FF of drum synchronizer S set then M → C {jump}; reset FF else (C) + 1 → C end if when others => {undefined} end case
SN ‖-(A) → M; (C) + 1 → C	4 1	Field NNNNN is xxSxx, case S is when 0..7 => Synchronizer S: 0000000[(MAR)] → rP1; (MAR) are lost; (C) + 1 → C when others => {undefined} end case
SM ‖|(A)| → M; (C) + 1 → C	4 2	Field NNNNN is xxSxx, case S is when 0..7 => Synchronizer S: 0000000[(MAR)] → rP1; (C) + 1 → C when others => {undefined} end case
F ‖(M) → A; (C) + 1 → C	4 3	Field NNNNN is xxxSx, case S is when 1..4 => Tape Synchronizer S: 0000000[(MAR)] → rP1; (C) + 1 → C when others => {undefined} end case
	4 4	Field NNNNN is xxExx, case E^3 is when 0..9 => If Computer error FF specified by E set then M → C {jump}; reset FF else (C) + 1 → C end if

---

3 The 10 Type I Computer errors tested by Instruction 44 interrupt the Processor when set with a forced 14_00000_00001 instruction, and set the Computer error intervention interlock FF preventing further interrupts until the Instruction 83 executes, reseting the Computer error intervention interlock FF.

Computer Instruction		Opcode	Processor Instruction
			end case
SS	$\parallel$(A) → M; (A+1) → M+1; (C) + 1 → C	4 5	Field NNNNN is xxExx, case E[4] is when 1..5 => If Computer error FF specified by E set then M → C {jump}; reset FF else (C) + 1 → C end if when others => {undefined} end case
SSN	$\parallel$-(A) → M; -(A+1) → M+1; (C) + 1 → C	4 6	Field NNNNN is xxExx, case E is when 0..9 => If Synchronizer error FF specified by E set then M → C {jump}; reset FF else (C) + 1 → C end if end case
SSM	$\parallel$|(A)| → M; |(A+1)| → M+1; (C) + 1 → C	4 7	Field NNNNN is xxExx, case E is when 0..9 => If Synchronizer error FF specified by E set then M → C {jump}; reset FF else (C) + 1 → C end if end case
FF	$\parallel$(M) → A; (M+1) → A+1; (C) + 1 → C	4 8	If Synchronizer master error FF set then M → C {jump}; reset FF else (C) + 1 → C end if
		4 9	Field NNNNN is xxSxx, case S is when 0..7 => If Synchronizer S master error FF set then M → C {jump}; reset FF else (C) + 1 → C end if when others => {undefined} end case
CX	$\parallel$(A) - FL → A - FX, M: scale factor; (C) + 1 → C	5 0	Field NNNNN is xxSxx, case S is when 0..4 =>

---

4 The 5 Type II Computer errors tested by Instruction 45 interrupt the Processor when set with a forced 14_00000_00001 instruction, and set the error intervention interlock FF preventing further interrupts until the Instruction 83 executes, reseting the error intervention interlock FF.

Computer Instruction	Opcode	Processor Instruction
		If Drum Synchronizer S sector change FF set then M → C {jump}; reset FF else (C) + 1 → C end if when 5..6 => If Line Printer Synchronizer S action FF set then M → C {jump}; reset FF else (C) + 1 → C end if when 7 => If Electronic Page Recorder Synchronizer action FF set then M → C {jump}; reset FF else (C) + 1 → C end if when others => {undefined} end case
C ‖ (A) - FX → A - FL, M: scale factor; (C) + 1 → C	5 1	Field NNNNN is xxSxx, case S is when 7 => (rP1) → Electronic Page Recorder Synchronizer; (C) + 1 → C when others => {undefined} end case The rP1 format is xxxxxxXXXYY'Y' where XXXYY'Y' is starting coordinate and Y'Y' is alphanumeric character.

PR ‖ (A) x 10^{-M} → A {right shift M places}; (C) + 1 → C	5 2	Field NNNNN is xxSDD, case S is
		when 0..4 => D is range 1..24
		Connect Drum Synchronizer S to drum D; Set in-use FF of Synchronizer S; If drum D is on local control
		then M → C {jump}
		else step head of drum; (C) + 1 → C;
		  After a 7ms delay the Drum Synchronizer is alerted to read a sector band address from the drum; If a sector band address is read within 5ms
		  then it is stored in the sector address register of the Drum Synchronizer; Set sector-change FF
		  else Set both bad-band FF and sector-

Computer Instruction	Opcode	Processor Instruction
		change FF end if end if end if when others => {undefined} end case
PL ‖$(A) \times 10^M \rightarrow A$ {left shift M places}; $(C) + 1 \rightarrow C$	5 3	Field NNNNN is xxSxx, case S is when 0..4 => Start Drum Synchronizer S in mode 1; 5 LSDs (rP1) → Drum Synchronizer MAR when 5..6 => Start Line Printer Synchronizer S; 5 LSDs (rP1) → Line Printer Synchronizer MAR; digit 6 (rP1) → Line Printer Synchronizer Mode Register; Advance paper 1 line; print 1 line in mode m when 7 => Start Electronic Page Recorder Synchronizer; 5 LSDs (rP1) → Electronic Page Recorder Synchronizer MAR; digit 6 (rP1) → Electronic Page Recorder Synchronizer Mode Register; Process 10 words in mode m when others => {undefined} end case; $(C) + 1 \rightarrow C$
	5 4	Field NNNNN is xxSxx, case S is when 0..4 => Reset Drum Synchronizer S in-use FF; If Drum Synchronizer S is a write synchronizer then disconnect it from the drums end if; If Drum Synchronizer S master error FF set then M → C {jump} else $(C) + 1 \rightarrow C$ end if when 5..6 => Reset Line Printer Synchronizer S in-use FF; If Line Printer Synchronizer S master error FF set then M → C {jump} else $(C) + 1 \rightarrow C$ end if when 7 => Set Electronic Page Recorder Synchronizer

Computer Instruction	Opcode	Processor Instruction
		action FF; If Electronic Page Recorder Synchronizer master error FF set then M → C {jump} else (C) + 1 → C end if when others => {undefined} end case
CCX ‖ (A') - FL → A' - FX, M: scale factor; (C) + 1 → C	5 5	Field `NNNNN` is `xxSxx`, case S is when 0..4 => Start Drum Synchronizer S in mode 2; 5 LSDs (rP1) → Drum Synchronizer MAR; (C) + 1 → C when others => {undefined} end case After last word of sector 24 is transferred, Drum Synchronizer MAR is adjusted MOD 2500
CC ‖ (A') - FX → A' - FL, M: scale factor; (C) + 1 → C	5 6	Field `NNNNN` is `xxSxx`, case S is when 0..4 => If Drum Synchronizer S in-use FF reset then M → C {jump} else (C) + 1 → C end if when 5..6 => If Line Printer Synchronizer S in-use FF set then M → C {jump} else (C) + 1 → C end if when 7 => If Electronic Page Recorder Synchronizer in-use FF set then M → C {jump} else (C) + 1 → C end if when 9 => If Tape Positioning Checker in-use FF reset then M → C {jump} else (C) + 1 → C end if when others => {undefined} end case

Computer Instruction	Opcode	Processor Instruction
PPR $\parallel$ (A') × 10^{-M} → A' {right shift M places}; (C) + 1 → C	5 7	Field `NNNNN` is `xxSxx`, case S is when 0..4 => Continue present sequence when Drum Synchronizer S sequence completes; If Drum Synchronizer S was nor reading or writing then do not continue; set both continue error FF and master error FF of Drum Synchronizer S end if when 7 => Continue Electronic Page Recorder Synchronizer; Process next 10 words in previous mode and memory address sequence when others => {undefined} end case; (C) + 1 → C
PPL $\parallel$ (A') × 10^{M} → A' {left shift M places}; (C) + 1 → C	5 8	Field `NNNNN` is `xxSxx`, case S is when 0..4 => For Drum Synchronizer S (sector-band address registers 00[(BB)]0000[(SS)]00) → rP1; (C) + 1 → C when others => {undefined} end case
PPC $\parallel$ (A') × 10^{M} → A' {left circular shift}; (C) + 1 → C	5 9	(tape/drum number register 00[(AB)]00000000) → rP1; (C) + 1 → C
EOP $\parallel$ (M_I) → A_I; (C) + 1 → C	6 0	Field `NNNNN` is `xxxST`, case S is when 1..4 => T^5 is range 0..9 If Synchronizer S availability FF reset then M → C {jump} else (C) + 1 → C end if when others => {undefined} end case
EA $\parallel$ (M_A) → A_A; (C) + 1 → C	6 1	If previously selected Synchronizer read-write availability FF reset then M → C {jump} else (C) + 1 → C end if
EB $\parallel$ (M_B) → A_B; (C) + 1 → C	6 2	If previously selected Synchronizer read bus

---

5   The Instruction 63 following this Instruction 60 selects the specified Uniservo T and connects it to Synchronizer S.

Univac LARC Instruction Codes

Computer Instruction	Opcode	Processor Instruction
		availability FF reset then M → C {jump} else (C) + 1 → C end if
EAB ‖(M_{AB}) → A_{AB}; (C) + 1 → C	6 3	Must be preceded by an Instruction 60 Reset Synchronizer S availability FF; If Synchronizer S tape-control error FF set then M → C {jump} else (C) + 1 → C; Select Uniservo T After 200µs set selection-complete FF end if
EM ‖(M_M) → A_M; (C) + 1 → C	6 4	Field NNNNN is xxxSx, case S is when 1..4 => If Synchronizer S selection-complete FF set then M → C {jump}; reset FF else (C) + 1 → C end if when others => {undefined} end case
EL ‖(A-1) → A (M); (C) + 1 → C	6 5	Connect positioning checker to Synchronizer; set in-use FF of positioning checker; start check-read operation on Uniservo connected to the Synchronizer; If any of tape-direction error FF or tape-control error FF or write-interlock ring FF set then M → C {jump} else (C) + 1 → C end if
EU ‖(A+1) → A (M); (C) + 1 → C	6 6	(rP1) → previously selected tape Synchronizer instruction register xIKLGxxxxxxx I = operation digit, range 1..8 K = translation digit, range 1..2 L = SBB length digit, range 0..1 G = gain digit, range 0..2 If Uniservo tape direction reversal not necessary then M → C {jump} else (C) + 1 → C

Computer Instruction	Opcode	Processor Instruction
		end if
	6 7	Start moving tape on Uniservo; If Uniservo not rewound then M → C {jump} else (C) + 1 → C end if
	6 8	If any of tape direction error or tape control error or tape write interlock ring have occurred then M → C {jump} else Start operation from Instruction 66 on Uniservo connected to previously selected tape Synchronizer; (C) + 1 → C end if
	6 9	Field NNNNN is xxxSx, case S is when 1..4 | 9 => If Synchronizer S ten-word FF set then M → C {jump}; reset FF else (C) + 1 → C end if when others => {undefined} end case

TE ‖If (A) = (A+1)
    then M → C {jump}
    else (C) + 1 → C
    end if
	7 0	If master error FF set
		then M → C {jump}
		else Continue operation from Instruction 66 on Uniservo connected to previously selected tape Synchronizer; (C) + 1 → C
		end if

TG ‖If (A) > (A+1)
    then M → C {jump}
    else (C) + 1 → C
    end if
	7 1	If master error FF set
		then M → C {jump}
		else Continue operation from Instruction 66 beyond SBB on Uniservo connected to previously selected tape Synchronizer; (C) + 1 → C
		end if

TZ ‖If (A) = 0
    then M → C {jump}
    else (C) + 1 → C
    end if
	7 2	If Uniservo availability FF of previously selected tape Synchronizer reset
		then M → C {jump}
		else (C) + 1 → C
		end if

Computer Instruction	Opcode	Processor Instruction
TGZ ‖ If $(A) > 0$ then $M \rightarrow C$ {jump} else $(C) + 1 \rightarrow C$ end if	7 3	If end of block FF of previously selected tape Synchronizer set then $M \rightarrow C$ {jump} else $(C) + 1 \rightarrow C$ end if
TLZ ‖ If $(A) < 0$ then $M \rightarrow C$ {jump} else $(C) + 1 \rightarrow C$ end if	7 4	5 LSDs (rP1) $\rightarrow$ previously selected tape Synchronizer MAR; $(C) + 1 \rightarrow C$
	7 5	Ring warning bell at operator console; $(C) + 1 \rightarrow C$
TTG ‖ If $(A') > (A+2')$ then $M \rightarrow C$ {jump} else $(C) + 1 \rightarrow C$ end if	7 6	If Uniservo interlock FF of previously selected tape Synchronizer reset then $M \rightarrow C$ {jump} else $(C) + 1 \rightarrow C$ end if
	7 7	If tape control error FF of previously selected tape Synchronizer set then $M \rightarrow C$ {jump}; reset FF else $(C) + 1 \rightarrow C$ end if
	7 8	Clear Uniservo control circuits of previously selected tape Synchronizer; $(C) + 1 \rightarrow C$ After 7.5ms set availability FF
	7 9	Illegitimate Instruction, Timeout Error
BIT ‖ $(A_\Delta) + (A_D) \rightarrow A_\Delta$; $(A_N) - 1 \rightarrow A_N$; If $(A_N) \neq 0$ then $M \rightarrow C$ {jump} else $(C) + 1 \rightarrow C$ end if	8 0	Reserved for future use[6] Generate unused FS880 pulse; $(C) + 1 \rightarrow C$
BUT ‖ $(A_\Delta) - (A_D) \rightarrow A_\Delta$; $(A_N) - 1 \rightarrow A_N$; If $(A_N) \neq 0$ then $M \rightarrow C$ {jump} else $(C) + 1 \rightarrow C$ end if	8 1	Field `NNNNN` is `xxDxx`, case D is when 1..2 => Connect Electronic Page Recorder D to Electronic Page Recorder Synchronizer; If Electronic Page Recorder D is out of order then $M \rightarrow C$ {jump} else $(C) + 1 \rightarrow C$

---

6 As of June 1959 (Univac® LARC Processor Instruction and Function-Signal Analyses).

Computer Instruction	Opcode	Processor Instruction
		end if when others => {undefined} end case
BIC ‖ $(A_\Delta) + (A_D) \rightarrow A_\Delta$; $(A_N) - 1 \rightarrow A_N$; If $(A_N) = 0$ then M $\rightarrow$ C {jump} else (C) + 1 $\rightarrow$ C end if	8 2	Field NNNNN is xxDxx, case D is when 1..4 => Turn off console printer D[7]; (C) + 1 $\rightarrow$ C when others => {undefined} end case
BDC ‖ $(A_\Delta) - (A_D) \rightarrow A_\Delta$; $(A_N) - 1 \rightarrow A_N$; If $(A_N) = 0$ then M $\rightarrow$ C {jump} else (C) + 1 $\rightarrow$ C end if	8 3	Reset the Computer error intervention interlock FF; (C) + 1 $\rightarrow$ C
	8 4	Field NNNNN is xxCxx, in a one Computer system C in range 0..4, in a two Computer system C in range 0..9[8] If Processor manual intervention FF C set then M $\rightarrow$ C {jump}; reset FF else (C) + 1 $\rightarrow$ C end if
BI ‖ $(A_\Delta) + (A_D) \rightarrow A_\Delta$; (C) + 1 $\rightarrow$ C	8 5	Turn on console printer connected to console printer Synchronizer; If either console printer error FF or console printer interlock FF set then M $\rightarrow$ C {jump}; reset FF else (C) + 1 $\rightarrow$ C end if
BD ‖ $(A_\Delta) - (A_D) \rightarrow A_\Delta$; (C) + 1 $\rightarrow$ C	8 6	If console printer Synchronizer action FF set then M $\rightarrow$ C {jump}; reset FF else (C) + 1 $\rightarrow$ C end if
	8 7	Field NNNNN is xxSxx, case S is when 5..6 => Advance paper one line on Line Printer on Line Printer connected to Synchronizer S; If mechanical error then M $\rightarrow$ C {jump}

7   The console printer 1 is on Computer 1's engineer control console, the console printer 2 is on Computer 1's operator control console, the console printer 3 is on Computer 2's engineer control console, the console printer 4 is on Computer 2's operator control console (in a two Computer system, no such LARC system was ever built).

8   The manual intervention buttons 0..4 are on Computer 1's operator control console, the manual intervention buttons 5..9 are on Computer 2's operator control console (in a two Computer system, no such LARC system was ever built).

Computer Instruction	Opcode	Processor Instruction
		else (C) + 1 → C end if when 7 => Advance film one frame on Electronic Page Recorder connected to Synchronizer; If Electronic Page Recorder is printing or already advancing film or has a failure then M → C {jump} else (C) + 1 → C end if when others => {undefined} end case
	8 8	Field NNNNN is xxDxx, case D is when 1 => Open shutter of land camera of Electronic Page Recorder connected to Electronic Page Recorder Synchronizer; If shutter was already open then M → C {jump} else (C) + 1 → C end if when 2 => Close shutter of land camera of Electronic Page Recorder connected to Electronic Page Recorder Synchronizer; If shutter was already closed then M → C {jump} else (C) + 1 → C end if when others => {undefined} end case
	8 9	Field NNNNN is xxDxx, case D is when 1..2 => If camera of Electronic Page Recorder D either out of film or film not moved since Instruction 87 then M → C {jump} else (C) + 1 → C end if when others => {undefined} end case
T ‖ M → C {jump}	9 0	Field NNNNN is xxDxx, case D is

Computer Instruction	Opcode	Processor Instruction
		when 1..2 => If camera action FF of Electronic Page Recorder D set then M → C {jump}; reset FF else (C) + 1 → C end if when others => {undefined} end case
TR ‖ 9_90_00_00_[(C)] + 1 → M; M+1 → C {jump}	9 1	Field NNNNN is xxDxx, case D is when 1..2 => If film changed FF of Electronic Page Recorder D set then M → C {jump}; reset FF else (C) + 1 → C end if when others => {undefined} end case
TB ‖ (C) → A$_M$; M → C {jump}	9 2	If any of action FF of Electronic Page Recorder Synchronizer or camera action FF of Electronic Page Recorder D or film changed FF of Electronic Page Recorder D set then M → C {jump} else (C) + 1 → C end if
SLJ ‖ If M is storage location then 9_90_00_00_[(C2)] → M elsif M is fast register then 0_00_00_00_[(C2)] → M {A$_M$} end if; (C) + 1 → C	9 3	Field NNNNN is xxDxx, case D is when 1..4 => Connect console printer D to the console printer synchronizer; If console printer D cannot be connected because either printing is already in progress or console printer D is either on local control or does not exist[9] then M → C {jump} else (C) + 1 → C end if when others => {undefined} end case
	9 4	Illegitimate Instruction, Timeout Error

---

9   The console printer 1 is on Computer 1's engineer control console, the console printer 2 is on Computer 1's operator control console, the console printer 3 is on Computer 2's engineer control console, the console printer 4 is on Computer 2's operator control console (in a two Computer system, no such LARC system was ever built).

Computer Instruction		Opcode	Processor Instruction

TF	‖If FF A
	   then M → C {jump}
	   else (C) + 1 → C
	   end if
		9 5	If realtime clock FF set
			then M → C {jump}; reset FF
			else (C) + 1 → C
			end if
			
			The realtime clock FF is set every 8.33ms

RF	‖Reset FF A; (C) + 1 → C	9 6	Field NNNNN is xxPxx, case P is when 1..2 => Reset Disclosure FF in Computer P; If FF failed to reset then M → C {jump} else (C) + 1 → C end if when 3..4 => Set Processor Contingency FF in Computer P-2; If FF failed to set then M → C {jump) else (C) + 1 → C end if when others => {undefined} end case
SF	‖Set FF A; (C) + 1 → C	9 7	Field NNNNN is xxPxx, case P is when 1..2 => If Disclosure FF in Computer P set then M → C {jump} else (C) + 1 → C end if when 3..4 => If Processor Contingency FF in Computer P-2 reset then M → C {jump} else (C) + 1 → C end if when others => {undefined} end case
		9 8	Field NNNNN is xxBxx, case B^{10} is when 0..4 => If breakpoint stop B on engineer control console is pressed then Stop the Processor;

10 Field designation of Instruction 98 changed from C to B to avoid confusion with C register.

Computer Instruction		Opcode	Processor Instruction

			When operator presses start;
			If force transfer switch set
			then M → C {jump}
			else[11], (C) + 1 → C
			end if;
			Start the Processor
			else (C) + 1 → C
			end if
			when others => {undefined}
			end case

H	‖Stop	9 9	Field NNNNN is xxPxx, case P is when 0 => If any condition specified by P in range 1..9 is true then M → C {jump} else (C) + 1 → C end if when 1 => If sector change FF of any Drum Synchronizer set then M → C {jump} else (C) + 1 → C end if when 2 => If ten word FF of any Tape Synchronizer or the positioning checker set then M → C {jump} else (C) + 1 → C end if when 3 => If realtime clock FF set or action FF of any Line Printer Synchronizer or Electronic Page Recorder Synchronizer set then M → C {jump} else (C) + 1 → C end if when 4 => If action FF of Console Printer Synchronizer or any manual intervention FF set then M → C {jump}

---

11 Behavior in this situation not explicitly specified in Univac® LARC Processor Instruction and Function-Signal Analyses. Logical inference is that if force transfer switch is off Instruction 98 continues after start without jumping.

Computer Instruction	Opcode	Processor Instruction
		else (C) + 1 → C end if when 5 => If action FF of any Drum set, then M → C {jump} else (C) + 1 → C end if when 6 => If action FF of one Drum (7..12) set then M → C {jump} else (C) + 1 → C end if when 7 => If action FF of one Drum (13..18) set then M → C {jump} else (C) + 1 → C end if when 8 => If action FF of one Drum (19..24) set then M → C {jump} else (C) + 1 → C end if when 9 => If selection-complete FF of any Tape Synchronizer set then M → C {jump} else (C) + 1 → C end if