MACRO INSTRUCTIONS (PLAN 3 ONLY)

NOTATION

N is a core store address or a 12 bit number.

I is an accumulator (registers 0-7).

M is a modifier register (registers 1-3).

Fis a function.

C is the carry register.

c is the content of C (0 or 1).

V is the overflow register.

A is the floating point accumulator.

a is the content of A.

z. m are the contents of I. H respectively.

n is the content of N after modification by m if necessary.

 I^* is the accumulator I + 1 ($I7^* = I0$)

x* is the content of I*.

x', n', a' are the contents of X, N, A after an instruction has been obeyed.

x:, n: are double length numbers in X, X + 1, and H, H + 1. respectively.

S is the sign bit (bit 0).

The most significant bit of the second word of a double length number is always zero.

Subscripts

In general these are applicable to x or n.

x. is the least significant 9 bits of x. The exponent of a floating point number occupies this portion of the second word.

 x_{α} is the least significant 12 bits of x (the N address of an instruction).

 x_n is a 9 bit counter at the most significant end of x.

 x_{m} is the least significant 15 bits of x (the modifier part of an index register).

x, is the most significant 2 bits of x, used in character modifying with end-around - carry to xm-

 x_a is the least significant 7 bits of x_c .

 x_1 is any one of x_0 , x_1 , x_2 , x_3 , the four 6-bit characters

No is the most significant 2 bits of the 12 bit N address.

N. is the least significant 10 bits of the 12 bit N address.

Note: -

* When in extended mode (1906 and 1907 only) the modifier extends to 22 bits, the count being held separately.

24-bit I.C.T. 1900 word								
•.		F	,	N		N o	r x _a	
NORMAL ORDERS	3	7		2		1	2	7
	Z	X F		N				
JUMP ORDERS	3	6		15				7
GLED'	I	F		N	N _t		N's	_
SHIFT ORDERS	3	7		2	.2		10	
DOUBLE LENGTH	S				23			
PIXED POINT NUMBER	0 1 23							
FLOATING	S		· · · · · · · · · · · · · · · · · · ·		23]
POINT NUMBER	0						×e	_
			14	·	,,,	<u></u>	9	
NORMAL		x _c				x _e		
COUNTER- MODIFIER	9			15				
CHARACTER	xk	×a				x _s		_
Counter- Modifier	2	2 7		15				
		0	x ₁			x ₂	x ₃	_
CHARACTER		8	6			6	6	1

LDX XX* N(M) x: '= n: 2 ADX XX* N(M) x: '= x: + n: 2 NGX XX* N(M) x: '= -n: 2 SBX XX* N(M) x: '= x: - n: 2 SBX XX* N(M) n: '= x: 2 ADS XX* N(M) n: '= x: 2 ADS XX* N(M) n: '= n: + x: 2 ADS XX* N(M) n: '= n: + x: 2 SBS XX* N(M) n: '= n: - x: 2 BXU X N ₁ (M), N ₂ If x ≠ n ₁ jump to N ₂ 2 EXU XX* N ₁ (M), N ₂ If x = n ₁ jump to N ₂ 3 BXE XX* N ₁ (M), N ₂ If x: = n ₁ jump to N ₂ 2 BXE XX* N ₁ (M), N ₂ If x: = n ₁ jump to N ₂ 3 EXL X N ₁ (M), N ₂ If x: < n ₁ jump to N ₂ 2 BXL XX* N ₁ (M), N ₂ If x: < n ₁ jump to N ₂ 3 EXCE X N ₁ (M), N ₂ If x: > n ₁ jump to N ₂ 2 BXGE XX* N ₁ (M), N ₂ If x: > n ₁ jump to N ₂ 3 LDSA X N(M) x' = n _a 2 LDSA X N(M) x' = n _a 1 LDSA X N(M) x' = n		INSTR	UCTI	DN ·	EFFECT	NSTRUCTIONS
NGX XX* N(M)		LDX	XX*	N(M)	x; ' = n:	2
SBX XX* N(M) x:' = x: - n: 2 STO XX* N(M) n:' = x: 2 ANS XX* N(M) n:' = n: + x: 2 NGS XX* N(M) n:' = n: - x: 2 SBS XX* N(M) n:' = n: - x: 2 EXU X N ₁ (M), N ₂ If x ≠ n ₁ jump to N ₂ 2 EXU XX* N ₁ (M), N ₂ If x: ≠ n ₁ : jump to N ₂ 3 EXE X N ₁ (M), N ₂ If x: = n ₁ : jump to N ₂ 3 EXE XX* N ₁ (M), N ₂ If x < n ₁ : jump to N ₂ 3 EXI X N ₁ (M), N ₂ If x < n ₁ : jump to N ₂ 3 EXI XX* N ₁ (M), N ₂ If x < n ₁ : jump to N ₂ 3 EXI XX* N ₁ (M), N ₂ If x: ≥ n ₁ : jump to N ₂ 3 EXIS XX* N		ADX	XX*	N(M)	x;' = x; + n;	
STO XX° N(M) $n:'=x:$ 2 ADS XX° N(M) $n:'=n:+x:$ 2 NGS XX° N(M) $n:'=n:+x:$ 2 SGS XX° N(M) $n:'=n:-x:$ 2 SGS XX° N(M), N ₂ If $x:\neq n_1$ jump to N_2 3 SGE X N ₁ (M), N ₂ If $x:\neq n_1$ jump to N_2 3 SGE XX° N ₁ (M), N ₂ If $x:\neq n_1$ jump to N_2 2 SGGE XX° N ₁ (M), N ₂ If $x:\neq n_1$ jump to N_2 3 SGGE XX° N ₁ (M), N ₂ If $x:\neq n_1$ jump to N_2 2 SGGE XX° N ₁ (M), N ₂ If $x:\neq n_1$ jump to N_2 3 SGGE XX° N ₁ (M), N ₂ If $x:\neq n_1$ jump to N_2 3 SGGE XX° N ₁ (M), N ₂ If $x:\neq n_1$ jump to N_2 3 SGGE XX° N ₁ (M), N ₂ If $x:\neq n_1$ jump to N_2 3 SGGE XX° N(M) $x'=n_0$ 2 SGGE XX° N(M) $x'=n_0$ 2 SGGE XX° N(M) $x'=n_0$ 1 SGGE X SGGE MTX 1 SGGE X SGGE X SGGE MTX 1 SGGE X SGGE X SGGE MTX 1 SGGE						
ADS XX^* N(M) $n:'=n:+x:$ 2 NGS XX^* N(M) $n:'=-x:$ 2 SBS XX^* N(M) $n:'=-x:$ 2 BXU X N ₁ (M), N ₂ If $x \neq n_1$ jump to N ₂ 2 BXU XX^* N ₁ (M), N ₂ If $x \neq n_1$ jump to N ₂ 3 BXE X N ₁ (M), N ₂ If $x = n_1$ jump to N ₂ 3 BXE XX^* N ₁ (M), N ₂ If $x = n_1$ jump to N ₂ 3 BXL X N ₁ (M), N ₂ If $x = n_1$; jump to N ₂ 2 BXL XX^* N ₁ (M), N ₂ If $x < n_1$; jump to N ₂ 3 BXGE X N ₁ (M), N ₂ If $x < n_1$; jump to N ₂ 3 BXGE X N ₁ (M), N ₂ If $x > n_1$; jump to N ₂ 2 BXGE XX^* N ₁ (M), N ₂ If $x > n_1$; jump to N ₂ 3 LDSA X N(M) $x' = n_0$ 2 LDSA X N(M) $x' = n_0$ 1 BYM X Rewind MTX 1 BSP X Backspace MTX 1 BYM X Rewind MTX 1 BYM X Move back past tape mark on MTX 1 FYM X Move forward past tape mark on MTX 1 FYM X Move forward past tape mark on MTX 1 CLOSE X Close MTX 1 SCR X OPEN MTX and leave scratch 1		SBX				
NGS XX* N(M) $n:'=-x:$ 2 SBS XX* N(M) $n:'=n:-x:$ 2 BXU X $N_1(M), N_2$ If $x \neq n_1$ jump to N_2 2 BXU XX* $N_1(M), N_2$ If $x:\neq n_1$ jump to N_2 3 BXE X $N_1(M), N_2$ If $x:\neq n_1$ jump to N_2 2 BXE XX* $N_1(M), N_2$ If $x:=n_1$ jump to N_2 3 BXL X $N_1(M), N_2$ If $x:=n_1$: jump to N_2 2 BXL XX* $N_1(M), N_2$ If $x:=n_1$: jump to N_2 3 BXGE X $N_1(M), N_2$ If $x:=n_1$: jump to N_2 2 BXGE XX* $N_1(M), N_2$ If $x:=n_1$: jump to N_2 3 BXGE XX* $N_1(M), N_2$ If $x:=n_1$: jump to N_2 3 LDEA X $N(M)$ $x':=n_0$ 2 LDEA X $N(M)$ $x':=n_0$ 1 BYM X Rewind MTX 1 BSP X Backspace MTX 1 BYM X Rewind MTX 1 FYM X Move back past tape mark on MTX 1 FYM X Move forward past tape mark on MTX 1 FYM X Move forward past tape mark on MTX 1 FYM X Close X Close MTX 1 SCR X OPEN MTX and leave scratch 1		STO	XX.			
BSS XX* N(M) $n: = n: -x:$ 2 BXU X $N_1(M), N_2$ If $x \neq n_1$ jump to N_2 2 BXU XX* $N_1(M), N_2$ If $x: \neq n_1$: jump to N_2 3 BXE X $N_1(M), N_2$ If $x: \neq n_1$: jump to N_2 2 BXE XX* $N_1(M), N_2$ If $x: = n_1$: jump to N_2 3 BXL XX* $N_1(M), N_2$ If $x: = n_1$: jump to N_2 2 BXL XX* $N_1(M), N_2$ If $x: \leq n_1$: jump to N_2 3 BXGE X $N_1(M), N_2$ If $x: \leq n_1$: jump to N_2 2 BXGE XX* $N_1(M), N_2$ If $x: \geq n_1$: jump to N_2 3 LDSA X $N(M)$ $x' = n_a$ 2 LDSA X $N(M)$ $x' = n_a$ 1 LDSA X $N(M)$ $x' = n_a$ 2 LDSA X $N(M)$ $x' = n_a$ 1 LDSA X $N(M)$ $x' = n_a$ 2 LDSA X $N(M)$ $x' = n_a$ 1 LDSA X $N(M)$ $x' = n_a$ 2 LDSA X $N(M)$ $x' = n_a$ 1 LDSA X $N(M)$ $x' = n$		ADS				
BXU X $N_1(M), N_2$ If $x \neq n_1$ jump to N_2 2 BXU XX* $N_1(M), N_2$ If $x : \neq n_1$: jump to N_2 3 BXE X $N_1(M), N_2$ If $x : \neq n_1$: jump to N_2 2 BXE XX* $N_1(M), N_2$ If $x : = n_1$: jump to N_2 3 BXL X $N_1(M), N_2$ If $x : = n_1$: jump to N_2 2 BXL XX* $N_1(M), N_2$ If $x : = n_1$: jump to N_2 3 BXGE X $N_1(M), N_2$ If $x : = n_1$: jump to N_2 3 BXGE X $N_1(M), N_2$ If $x : = n_1$: jump to N_2 3 BXGE XX* $N_1(M), N_2$ If $x : = n_1$: jump to N_2 3 BXGE XX* $N_1(M), N_2$ If $x : = n_1$: jump to N_2 3 LDSA X $N_1(M), N_2$ If $x : = n_1$: jump to		NGS				
BXU XX* N ₁ (M), N ₂ If x : $\neq n_1$: jump to N ₂ 3 BXE X N ₁ (M), N ₂ If $x = n_1$ jump to N ₂ 2 BXE XX* N ₁ (M), N ₂ If x : $= n_1$: jump to N ₂ 3 BXL X N ₁ (M), N ₂ If x : $< n_1$: jump to N ₂ 2 BXL XX* N ₁ (M), N ₂ If x : $< n_1$: jump to N ₂ 3 BXGE X N ₁ (M), N ₂ If x : $< n_1$: jump to N ₂ 2 BXGE X N ₁ (M), N ₂ If x : $< n_1$: jump to N ₂ 2 BXGE XX* N ₁ (M), N ₂ If x : $< n_1$: jump to N ₂ 3 LDSA X N(M) $x' = n_\alpha$ 2 LDSA X N(M) $x' = n_\alpha$ 2 LDLA X N(M) $x' = n_\alpha$ 2 LDLA X N(M) $x' = n_\alpha$ 1 BSP X Rewind MTX 1 BSP X Backspace MTX 1 BYM X Rewind MTX 1 BYM X Move back past tape mark on MTX 1 FYM X Move forward past tape mark on MTX 1 CLOSE X Close MTX 1 SCR X OPEN MTX and leave scratch 1		SBS	XX.	N(N)	n:'=n:-x:	2
BXE X N ₁ (M), N ₂ If $x = n_1$ jump to N_2 2 BXE XX* N ₁ (M), N ₂ If $x := n_1$; jump to N_2 3 BXL X N ₁ (M), N ₂ If $x := n_1$; jump to N_2 2 BXL XX* N ₁ (M), N ₂ If $x := n_1$; jump to N_2 3 EXCE X N ₁ (M), N ₂ If $x := n_1$ jump to N_2 2 BXGE XX* N ₁ (M), N ₂ If $x := n_1$ jump to N_2 3 LDSA X N(M) $x' := n_n$ 2 LDSA X N(M) $x' := n_n$ 2 LDSA X N(M) $x' := n_n$ 2 LDSA X N(M) $x' := n_n$ 1 LDSA X N(M) $x' := n_n$ 1 BYM X Rewind MTX 1 BSP X Backspace MTX 1 BYM X Move back past tape mark on MTX 1 FYM X Move forward past tape mark on MTX 1 CLOSE X Close MTX 1 SCR X OPEN MTX and leave scratch 1		BXU				2
BXE X N ₁ (M), N ₂ If $x = n_1$ jump to N_2 2 BXE XX* N ₁ (M), N ₂ If $x := n_1$; jump to N_2 3 BXL X N ₁ (M), N ₂ If $x := n_1$; jump to N_2 2 BXL XX* N ₁ (M), N ₂ If $x := n_1$; jump to N_2 3 EXCE X N ₁ (M), N ₂ If $x := n_1$ jump to N_2 2 BXGE XX* N ₁ (M), N ₂ If $x := n_1$ jump to N_2 3 LDSA X N(M) $x' := n_n$ 2 LDSA X N(M) $x' := n_n$ 2 LDSA X N(M) $x' := n_n$ 2 LDSA X N(M) $x' := n_n$ 1 LDSA X N(M) $x' := n_n$ 1 BYM X Rewind MTX 1 BSP X Backspace MTX 1 BYM X Move back past tape mark on MTX 1 FYM X Move forward past tape mark on MTX 1 CLOSE X Close MTX 1 SCR X OPEN MTX and leave scratch 1		BXU	XX*	N ₁ (M),N ₂	If $x: \neq n_1$: jump to N_2	
BXL X N ₁ (M), N ₂ If $x < n_1$ jump to N_2 2 BXL XX* N ₁ (M), N ₂ If $x : < n_1$: jump to N_2 3 BXGE X N ₁ (M), N ₂ If $x \ge n_1$ jump to N_2 2 BXGE XX* N ₁ (M), N ₂ If $x \ge n_1$: jump to N_2 3 LDSA X N(M) $x' = n_0$ 2 LDSA X N(M) $x' = n_0$ 2 LDSA X N(M) $x' = n_0$ 1 LDSA X N(M) $x' = n_0$ 1 BSP X Rewind MTX 1 BSP X Rewind MTX 1 BSP X Backspace MTX 1 BTM X Move back past tape mark on MTX 1 FTM X Move forward past tape mark on MTX 1 CLOSE X Close MTX 1 SCR X OPEN MTX and leave scratch 1			X	$N_1(M), N_2$	If $x = n_1$ jump to N_2	
BXL XX* N ₁ (M), N ₂ If $x: < n_1$: jump to N_2 3 BXGE X N ₁ (M), N ₂ If $x \ge n_1$ jump to N_2 2 BXGE XX* N ₁ (M), N ₂ If $x: \ge n_1$: jump to N_2 3 LDEA X N(M) $x' = n_a$ 2 LDEA X N(M) $x' = n_a$ 2 LDEA X N(M) $x' = n_a$ 1 LDEA X N(M) $x' = n_a$ 2 LDEA X N(M) $x' = n_a$ 1 BYTM X Write tape mark on MTX 1 REW X Rewind MTX 1 BSP X Backspace MTX 1 BYTM X Move forward past tape mark on MTX 1 FTM X Move forward past tape mark on MTX 1 CLOSE X Close MTX 1 SCR X OPEN MTX and leave scratch 1		BXE	XX*	$N_1(M),N_2$	If x : = n_1 : jump to N_2	
BXL XX* N ₁ (M), N ₂ If $x: < n_1$: jump to N_2 3 BXGE X N ₁ (M), N ₂ If $x \ge n_1$ jump to N_2 2 BXGE XX* N ₁ (M), N ₂ If $x: \ge n_1$: jump to N_2 3 LDEA X N(M) $x' = n_a$ 2 LDEA X N(M) $x' = n_a$ 2 LDEA X N(M) $x' = n_a$ 1 LDEA X N(M) $x' = n_a$ 2 LDEA X N(M) $x' = n_a$ 1 BYTM X Write tape mark on MTX 1 REW X Rewind MTX 1 BSP X Backspace MTX 1 BYTM X Move forward past tape mark on MTX 1 FTM X Move forward past tape mark on MTX 1 CLOSE X Close MTX 1 SCR X OPEN MTX and leave scratch 1		BXL		$N_1(M),N_2$	If $x \le n_1$ jump to N_2	
BXGE XX* $N_1(M), N_2$ If $x: \ge n_1$: jump to N_2 3 LDEA X $N(M)$ $x' = n_a$ 2 LDEA X $N(M)$ $x' = n_a$ 2 LDPL X $N(M)$ $x' = n_a$ 2 LDPL X $N(M)$ $x' = n_a$ 1 WTM X Write tape mark on MTX 1 REW X Rewind MTX 1 BXP X Backspace MTX 1 BYM X Move back past tape mark on MTX 1 FYM X Move forward past tape mark on MTX 1 CLOSE X Close MTX 1 SCR X OPEN MTX and leave scratch 1		BXL	XX*	$N_1(M), N_2$	If $x: < n_1$: jump to N_2	
LDEA X N(M) $x' = n_a$ 2 LDEA X N(M) $x' = n_a$ 1 WITH X Write tape mark on MIX 1 REW X Rewind MIX 1 BSP X Backspace MIX 1 BTM X Move back past tape mark on MIX 1 FTM X Move forward past tape mark on MIX 1 CLOSE X Close MIX 1 SCR X OPEN MIX and leave scratch 1		BXGE	X	N ₁ (M),N ₂	If $x \ge n_1$ jump to N_2	
LDLA X N(M) x' = n _m 2 LDPL X N x' = N(15 bits) 1 WTM X Write tape mark on MTX 1 REW X Rewind MTX 1 BSP X Backspace MTX 1 BTM X Move back past tape mark on MTX 1 FTM X Move forward past tape mark on MTX 1 CLOSE X Close MTX 1 SCR X OPEN MTX and leave scratch 1		BXGE	XX•	N ₁ (N),N ₂	If $x_1 \ge n_1$: jump to N_2	3
WTM X Write tape mark on MTX 1 REW X Rewind MTX 1 BSP X Backspace MTX 1 BTM X Move back past tape mark on MTX 1 FTM X Move forward past tape mark on MTX 1 CLOSE X Close MTX 1 SCR X OPEN MTX and leave scratch 1		LDGA	X	N(M)	$x' = n_a$	_
WTM X Write tape mark on MTX 1 REW X Rewind MTX 1 BSP X Backspace MTX 1 BTM X Move back past tape mark on MTX 1 FTM X Move forward past tape mark on MTX 1 CLOSE X Close MTX 1 SCR X OPEN MTX and leave scratch 1		LIXLA	X	N(N)		
REW X Rewind MTX 1 BSP X Backspace MTX 1 BTM X Move back past tape mark on MTX 1 FTM X Move forward past tape mark on MTX 1 CLOSE X Close MTX 1 SCR X OPEN MTX and leave scratch 1		LDPL	X	N	x' = N(15 bits)	1
REW X Rewind MTX 1 BSP X Backspace MTX 1 BTM X Move back past tape mark on MTX 1 FTM X Move forward past tape mark on MTX 1 CLOSE X Close MTX 1 SCR X OPEN MTX and leave scratch 1		_				
BSP X Backspace MTX 1 BTM X Move back past tape mark on MTX 1 PTM X Move forward past tape mark on MTX 1 CLOSE X Close MTX 1 SCR X OPEN MTX and leave scratch 1	-1	WIM	X		Write tape mark on MTX	1
BTM X Move back past tape mark on MTX 1 PTM X Move forward past tape mark on MTX 1 CLOSE X Close MTX 1 SCR X OPEN MTX and leave scratch 1	,		X.		Rewind MTX	
PTM X Move forward past tape mark on MTX 1 CLOSE X Close MTX 1 SCR X OPEN MTX and leave scratch 1						-
CLOSE X Close MTX 1 SCR X OPEN MTX and leave scratch 1	١					-
SCR X OPEN MIX and leave scratch 1	1	FTM				
		CLOSE				-
UNL X Close file and unload	•	SCR	X		OPEN MTX and leave scratch	1
f and the second	-	UNL	X		Close file and unload	

NO. OF BASIC

MAJOR DIRECTIVES

POSITIONS

The appearance of any directive in this group cancels the effect of any previous directive in the group.

PROGRAM - introduces a section of program instructions

LOWER - introduces lower data (below location 4096)

UPPER - introduces upper data (not Plan 1)

PERIPHERAL - is followed by specification of peripherals (other than magnetic tapes)

- indicates that a description of a private macro

follows (Plan 3 only)

END - the last statement of a segment; ends compilation

- indicates that this is the last segment to be compiled # PINISH

PLAN 1 only

MACRO

COMPLETE - indicates that the program is to be output in consolidated form.

PROGRAM AREA DIRECTIVES

MACROS

These directives appear in PROGRAM area only.

- gives a label to the following instruction for use by all

ENTRY - makes the following instruction entry point N, where N is written in the operand field

- introduces specification of monitor printing

GENERAL PURPOSE DIRECTIVES

The directives may appear anywhere in the program

SET - used to define a name (may be reset)

DEFINE - used to define a name (may not be redefined)

- used for writing comments

PAGE - causes paper throw on printer.

```
1.DX
                                                                                          070 CALL Subroutine Entry
                                                                                                                                                             * 130 FLOAT Convert n: from fixed to floating
              x' = x + n + c \quad V
                                                                                                      Link in X
                                                                                                                                                             * 131 FIX
                                                                                                                                                                            Convert a from floating to fixed
                                                                                               EXIT
                                                                                                      Subroutine Exit V
                                                                                                                                                            ** 132
                                                                                                                                                                    FAD
                                                                                                                                                                             a^i = a + n
                                                                                                                                                                                            If X = 1. Unrounded
  003
                                                                                                      Link in X
                                                                                                                                                            ** 133
                                                                                                                                                                             a' = a - n:
                                                                                                                                                                     FSR
                                                                                                                                                                                              X = 2, Not normalized
  004
        LDXC x' = n + c
                                                                                                      Conditional Branch to N: -
                                                                                                                                                                            a' = a.n:
                                                                                                                                                            ** 134
                                                                                                                                                                    FMPY
                                                                                                                                                                                               X = 4, Interchange a and n
  005
        ADXC x' = x + n + c
                                                                            X = 0
                                                                                                      Branch unconditionally
                                                                                                                                                            ** 135
                                                                                                                                                                    FDVD
                                                                                                                                                                            a' = a/n
   900
        NGXC x' = -n - c
                                                                            X = 1
                                                                                                RVS
                                                                                                      Branch if V is set
                                                                                                                                                            ** 136
                                                                                                                                                                    LFP
                                                                                                                                                                            a' = n:
                                                                                                                                                                                            If X = 1, a' = 0
   007
        SBXC x' = x - n - c C
                                                                            X = 2
                                                                                                      Branch if V is set and clear V
                                                                                                                                                            ** 137
                                                                                                                                                                    SFP
                                                                                                                                                                             n: ' = a
                                                                                                                                                                                            If X = 1, n : ' = a, a' = 0
                                                                             X = 3
                                                                                                      Branch if V is clear
                                                                                                      Branch if V is clear or clear V
                                                                             X = 4
                                                                                               RVCR
  010 STO
               n' = x + c
                                                                             X = 5
                                                                                               BCS
                                                                                                      Branch if C is set
                                                                                                                                                                             SUSBY Suspend if peripheral N(M), unit X, is active
  011
        ADS
               n' = n + x + c V
                                                                                                                                                             * 150 X N(M)
                                                                             X = 6
                                                                                               RCC
                                                                                                      Branch if C is clear
                                                                                                                                                             * 151 X N(M)
  012
        NGS
              n' = -x - c
                                                                                                                                                                            REI.
                                                                                                                                                                                     Release peripheral N(M), unit X
                                                                            X = 7
                                                                                                      Branch if V is clear and/or invert V V
                                                                                                                                                             * 152 X N(M)
  013
        SBS
               n' = n - x - c V
                                                                                                                                                                                    Disengage peripheral N(M), unit X
                                                                                         1076
                                                                                                      Test floating point accumulator
                                                                                                                                                             * 153 X N(M)
                                                                                                                                                                                     Unassigned
  014 STOC n' = x + c
                                                                                                                                                             * 154 X N(M)
                                                                                                                                                                            CONT
                                                                                                                                                                                    Read more program from peripheral N(M), unit X
  015
        ADSC n' = n + x + c C
                                                                                                                                                             * 155 X N(M)
                                                                                                                                                                            SUSDP
                                                                                                                                                                                    Suspend and dump program on peripheral N(M), unit X
  016
        NGSC n' = -x - c C
                                                                                                    x' = N + c
                                                                                                                                                             # 156 X N(M)
                                                                                                                                                                            ALLOT
                                                                                                                                                                                    Allocate peripheral N(M), unit X, to the program
  017
        SBSC n' = n - x - c C
                                                                                                    x' = x + N + c \quad V
                                                                                         181
                                                                                               ADN
                                                                                                                                                             * 157 X N(M)
                                                                                                                                                                                    Initiate peripheral transfer according to control
                                                                                                                                                                            PERT
                                                                                          102
                                                                                               NGN
                                                                                                    x' = -N - c
                                                                                                                                                                                     area N(M), unit X
        ANDX x' = x & n
  020
                                                                                          103 SBN
                                                                                                    x' = x - N - c
        ORX x' = x v n
  021
                                                                                          104 LDNC x' = N + c
  022
        ERX
             x' = x \neq n
                                                                                          105 ADNC x' = x + N + c C
                                                                                                                                                             . 160 0 N(M)
                                                                                                                                                                            SUSTY Suspend and type message on console typewriter
  023
        OBEY Obey the instruction in N
                                                                                          106 NGNC x' = -H - c C
                                                                                                                                                             * 160 1 N(M)
                                                                                                                                                                                    Type message on console typewriter without suspension
        LDCH x' = n_i
  024
                                                                                         107 SBNC x' = x - N - c C
                                                                                                                                                             * 160 2 N/M)
                                                                                                                                                                                    Delete program and treat message as console directive
  925
        LDEX x' = n_{\theta}
                                                                                                                                                             * 161 0 NN(M)
                                                                                                                                                                            SUSUT
                                                                                                                                                                                    Suspend and type HALTED NN on the console typewriter
  026
        TXU
               Set C if n \neq x or c = 1
                                                                                                                                                             * 161 1 NN(M) DISP
                                                                                                                                                                                    Type DISPLAY NN on the console typewriter without
  027
        TXL
               Set C if n + c > x
                                                                             N_{+} = 0
                                                                                          110 SLC
                                                                                                      Shift x left N. places. Circular
                                                                                                                                                                                     suspension
                                                                            N_t = 1
                                                                                               SLL
                                                                                                      Shift x left N, places. Logical
                                                                                                                                                             * 161 2 NN(M)
                                                                                                                                                                                    Delete program and type DELETED NN on the console
                                                                                                                                                                            DEL.
        ANDS n' = n & x
                                                                             N_t = 2, 3
                                                                                                      Shift x left N_x places. Arithmetic
  930
                                                                                               SLA
                                                                                                                                                                                     typewriter
  831
        ORS n' = n n x
                                                                             N + = 0
                                                                                          112 SRC
                                                                                                      Shift x right No places. Circular
                                                                                                                                                 Single
                                                                                                                                                            +* 162 X 0
                                                                                                                                                                                    Suspend if subprogram X is active
  032
        ERS
             n' = n \neq x
                                                                             N_{t} = 1
                                                                                                      Shift x right No places. Logical
                                                                                               SRI.
                                                                                                                                                length
                                                                                                                                                            †* 163 X N(M)
                                                                                                                                                                            AUTO
                                                                                                                                                                                    Activate and enter supprogram X at N(M)
        STOZ n' = 0
                                                                                                      Shift x right N_3 places. Arithmetic
  033
                                                                                               SRA
                                                                                                                                                            +* 164 0 0
                                                                                                                                                                            SUSAR
                                                                                                                                                                                    De-activate the current subprogram
  034
                                                                                               SRAV
                                                                                                      Shift x right N. places. Special
        DCH
             n_j' = x_3
                                                                                                                                                             * 165 X N(M)
                                                                                                                                                                            GIVE
                                                                                                                                                                                    If N(M) = 0. X will contain date in binary
  035
        DEX
               n_e = x_e
                                                                                        @ 114 NORM
                                                                                                      Normalize x
                                                                                                                                             V
                                                                                                                                                                                    If N(M) = 1. XX* will contain date in character
  036
                                                                                        ‡ 116 MVCH Transfer N characters
        DSA
               na' = xa
                                                                                                                                                                                                 form
  037
              n_{\infty}^{-1} = x_{\infty}
        DLA
                                                                                                                                                                                    If N(M) = 2. XX* will contain time in character
                                                                                                                                                                                                form
                                                                                       0 111 SLC
                                                                                                      Shift x: left No places. Circular
o 040
            x:'=n.x
                                                                             N = 0
                                                                                                                                                                                    If N(M) = 3. X will contain core store allocated
        MPV
                                                                             N_{\pm}^{c} = 1
                                                                                               SEL
                                                                                                      Shift x: left No places. Logical
o 041
        MPR x' = n.x rounded, x^* spoiled
                                                                                                                                                                                                to this program
                                                                                                      Shift x: left N_s places. Arithmetic
                                                                             N_{\pm} = 2,3
        MPA \quad x: ' = n.x + x^*
                                                                                               SLA
0 042
        CDB x: = 10.x: + n_j
                                                                            N_{\pm} = 0
                                                                                       o 113 SRC
                                                                                                      Shift x: right N_s places. Circular
o 043
                                                                                                                                                Double
                                                                            N_t = 1
N_t = 2
                                                                                                      Shift x: right N. places. Logical
        DVD x^{*'} = x; /n, x' = Remainder
                                                                                               SEL
o 044
                                                                                                                                                length
        DVR x^{*'} = x; /n rounded, x' = Remainder
                                                                                               SEA.
                                                                                                      Shift x: right No places. Arithmetic
o 645
                                                                                                                                                                 The function codes 140 to 147 are undefined.
        DVS x^{*'} = x^{*}/n, x^{!} = Remainder
                                                                                               SRAV
                                                                                                      Shift x: right No places. Special
a 048
                                                                                        0 115 NORM
o 047
        CBD x: ' = 10.x:, n_i' = Character
                                                                                                      Normalize x:
                                                                                                                                                                These instructions may set the carry register but cannot cause
                                                                                        $ 117 SMO
                                                                                                      Supplementary modifier to next instruction
  050
               Branch to N if x = 0
        RZE
                                                                                                                                                                 The carry register C is left clear by any order except 023 and 123.
               Branch to N if x \neq 0
   852
        RN7
                                                                                                    x' = x * N
                                                                                                                                                                 unless that order sets C.
   054
        BPZ
               Branch to N if x \ge 0
                                                                                         121 ORN
                                                                                                    x' = x v N
   ARR
               Branch to N if x < 0
                                                                                                                                                                These instructions may cause overflow.
                                                                                          122
                                                                                               EEN
                                                                                                      x' = x \neq x
   060
               Single word modify: x_n' = x_n + 1
                                                    x_c' = x_c - 1
                                                                                         123 NULL No operation
                                                                                                                                                                These instructions are performed on 1902, 1903 by extracode and by
   082
        BDX
               Double word modify: x_n' = x_n + 2
                                                     Branch to N
                                                                                         124 LDCT x_c' = N, x_{\pi'} = 0
                                                                                                                                                                 hardware on the other machines.
  864
               Character modify: x_m' = x_m + .1
        BCHX
                                                    if x_c' \neq 0
                                                                                         125 MGDE Set mode N
               Count least significant 15 bits of \tilde{X}. x_a' = x_a - 1
 ‡ 066
                                                                                       o 126 MOVE Transfer # words from address x to address x*
                                                                                                                                                                These instructions are performed on 1902, 1903, 1904 by extracode and
                                                     Branch to N
                                                                                       o 127 SDE
                                                                                                    x' = Sum of N words from address x*
                                                                                                                                                                 by hardware on the other machines.
                                                     11 x / # 0
                                                                                                                                                                 These instructions are performed by extracode on all machines.
                   INTERNATIONAL COMPUTERS AND TARULATORS LTD.
                                                                                                                                                                 These facilities are not available on 1902, 1903 processors with less
                                 I.C.T. 1900 SERIES
                                                                                                  TECHNICAL SERVICES.
                                                                                                      113 UPPER RICHMOND RD.,
                                                                                                                                                                These instructions are available on 1906, 1907 processors only.
                    PLAN - SUMMABISED PROGRAMMING INFORMATION
                                                                                                           PUTNEY, LONDON, S.W.15.
                                                                                                                                                                           C International Computers and Tabulators Ltd.
FORM 11/129 (1.67)
                                      DATE OF PUBLICATION - MARCH 1965
```