

1900 Series PLAN Summarised Programming Information



050	BZE	Branch to N if $x = 0$
052	BNZ	Branch to N if $x \neq 0$
054	BPZ	Branch to N if $x \geq 0$
056	BNG	Branch to N if $x < 0$
†060	BUX	Single word modify: $x_m' = x_m + 1$ } $x_c' = x_c - 1$
†062	BDX	Double word modify: $x_m' = x_m + 2$ } Branch to N [if $x_c' \neq 0$]
†064	BCHX	Character modify: $x_k = 0, 1$ or 2 } $x_k' = x_k + 1, x_m' = x_m$ } $x_d' = x_d - 1$ $x_k = 3$ } $x_k' = 0, x_m' = x_m + 1$ } Branch to N if $x_d' \neq 0$
*066	BCT	Count least significant 15 bits of X . $x_m' = x_m - 1$ Branch to N if $x_m' \neq 0$

070	CALL	Subroutine Entry
		Link in X
072	EXIT	Subroutine Exit
		Link in X
074		Conditional Branch to N :
$X = 0$	BRN	Branch unconditionally
$X = 1$	BVS	Branch if V is set
$X = 2$	BVSR	Branch if V is set and clear V
$X = 3$	BVC	Branch if V is clear
$X = 4$	BVCR	Branch if V is clear or clear V
$X = 5$	BCS	Branch if C is set
$X = 6$	BCC	Branch if C is clear
$X = 7$	BVCI	Branch if V is clear and/or invert V
*076	BFP	Test floating point accumulator or floating point overflow, and branch accordingly.
$X = 0$		Branch to N if $a = 0$
$X = 1$		Branch to N if $a \neq 0$
$X = 2$		Branch to N if $a \geq 0$
$X = 3$		Branch to N if $a < 0$
$X = 4$		Branch to N if FOVR is clear
$X = 5$		Branch to N if FOVR is set

100	LDN	$x' = N + c$
101	ADN	$x' = x + N + c$
102	NGN	$x' = -N - c$
103	SBN	$x' = x - N - c$
104	LDNC	$x' = N + c$
105	ADNC	$x' = x + N + c$
106	NGNC	$x' = -N - c$
107	SBNC	$x' = x - N - c$

$N_t = 0$	110	SLC	Shift x left N_t places. Circular
$N_t = 1$		SLL	Shift x left N_t places. Logical
$N_t = 2, 3$		SLA	Shift x left N_t places. Arithmetic
$N_t = 0$	112	SRC	Shift x right N_t places. Circular
$N_t = 1$		SRL	Shift x right N_t places. Logical
$N_t = 2$		SRA	Shift x right N_t places. Arithmetic
$N_t = 3$		SRAV	Shift x right N_t places. Special
	*114	NORM	Normalize x
	*116	MVCH	Transfer N characters

$N_t = 0$	111	SLC	Shift x : left N_t places. Circular
$N_t = 1$		SLL	Shift x : left N_t places. Logical
$N_t = 2, 3$		SLA	Shift x : left N_t places. Arithmetic
$N_t = 0$	113	SRC	Shift x : right N_t places. Circular
$N_t = 1$		SRL	Shift x : right N_t places. Logical
$N_t = 2$		SRA	Shift x : right N_t places. Arithmetic
$N_t = 3$		SRAV	Shift x : right N_t places. Special
	*115	NORM	Normalize x :
	*117	SMO	Supplementary modifier to next instruction

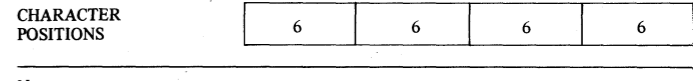
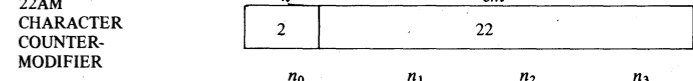
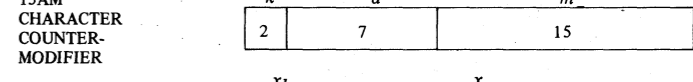
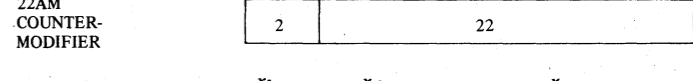
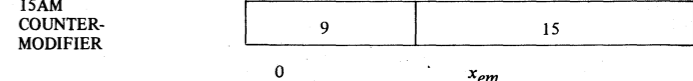
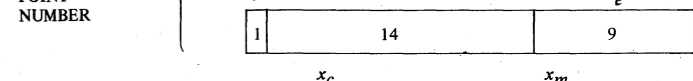
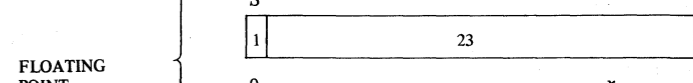
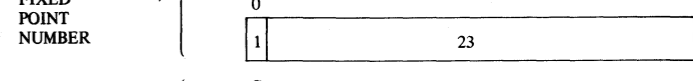
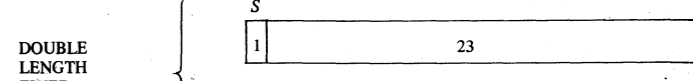
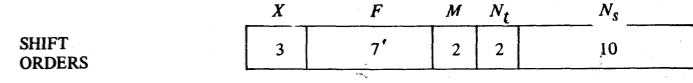
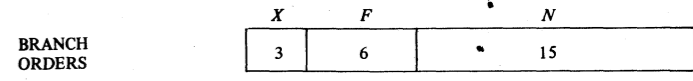
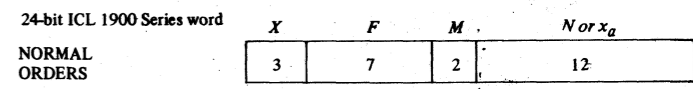
120	ANDN	$x' = x \& N$
121	ORN	$x' = x \vee N$
122	ERN	$x' = x \neq N$
123	NULL	No operation
124	.LDCT	$x_c' = N, n_m' = 0$
125	MODE	Set zero suppression mode
126	MOVE	Transfer N words from address x to address x^*
127	SUM	$x' = \text{Sum of } N \text{ words from address } x^*$

*130	FLOAT	Convert n : from fixed to floating and store in A
*131	FIX	Convert a from floating to fixed and store in $N(M)$ and $N(M) + 1$
*132	FAD	$a' = a + n$:
*133	FSB	$a' = a - n$:
*134	FMPY	$a' = a.n$:
*135	FDVD	$a' = a/n$:
*136	LFP	$a' = n$:
*136	LFPZ	$a' = 0$, when $X = 1$
*137	SFP	$n' = a$
*137	SFPZ	$n' = a, a' = 0$, when $X = 1$

150	SUSBY	Suspend if a specified peripheral is active
151	REL	Release a specified peripheral
152	DIS	Disengage a specified peripheral
153		Unassigned
154	CONT	Read more program from a specified peripheral
*155	SUSDP	Suspend and dump program on a specified peripheral
156	ALLOT	Assign, or supply information about, a specified peripheral or file.
157	PERI	Initiate action on a peripheral according to control area $N(M)$.

160 0 N(M)	SUSTY	Suspend and type message on console typewriter
160 1 N(M)	DISTY	Type message on console typewriter without suspension
160 2 N(M)	DELTY	Delete program and treat message as console directive
161 0 N(M)	SUSWT	Suspend and type HALTED n_a (as two characters) on the console typewriter
161 1 N(M)	DISP	Type DISPLAY n_a (as two characters) on the console typewriter without suspension
161 2 N(M)	DEL	Delete program and type DELETED n_a (as two characters) on the console typewriter
162 X N(M)	SUSMA	If $n^ = 0$, make $n^* \neq 0$ and $n' = x$, and omit next instruction.

*163 X N(M)	AUTO	Activate member X at $N(M)$. For reactivation, $N(M)$ must be zero.
*164 1 0	SUSAR	Suspend current member awaiting reactivation, by AUTO.
*164 2 0	SUSIN	Suspend current member awaiting flag-setting interrupt or AUTO.
165 X N(M)	GIVE	$N(M) = 0$ Give date in binary in X $N(M) = 1$ Give date in characters in XX^* $N(M) = 2$ Give time in characters in XX^* $N(M) = 3$ Give current core store allocation in X . $N(M) = 4$ Alter core store allocation to that specified in X . $N(M) = 5$ Give details of Executive and central processor $N(M) = 8$ Give current address mode and branch mode in X $N(M) = 9$ Alter address mode and branch mode to those specified in X
*166 X N(M)	RRQ	$X = 0$ Read request block into store at $N(M)$ $X = 1$ Replace request block from store at $N(M)$



Notes

The function codes 140 to 147 are undefined

C These instructions may set the carry register but cannot cause overflow. The 043 order may set V or C. The carry register C is left clear by any order except 023, 117 and 123, unless that order sets C.

V These instructions may cause overflow.

* These instructions are not available with some machines or with some Executives. For the availability of instructions with particular machine configurations see the Central Processors manual.

† In 22-bit address mode these instructions operate on X_{em} instead of x_m and branch unconditionally to N .

This card does not in all cases give a complete definition of an instruction. Further information on each instruction may be found in the Central Processors manual and the Plan Reference Manual. Hardware and software developments may alter the specifications of some instructions subsequent to the date of going to print, so a close watch on User Notices and relevant manual amendments is recommended.

Branch instructions are defined here in direct branch mode terms only. For extended branch mode, please refer to the Central Processors manual or the PLAN Reference Manual.

000	LDX	$x' = n + c$	V
001	ADX	$x' = x + n + c$	V
002	NGX	$x' = -n - c$	V
003	SBX	$x' = x - n - c$	V
004	LDXC	$x' = n + c$	C
005	ADXC	$x' = x + n + c$	C
006	NGXC	$x' = -n - c$	C
007	SBXC	$x' = x - n - c$	C

010	STO	$n' = x + c$	V
011	ADS	$n' = n + x + c$	V
012	NGS	$n' = -x - c$	V
013	SBS	$n' = n - x - c$	V
014	STOC	$n' = x + c$	C
015	ADSC	$n' = n + x + c$	C
016	NGSC	$n' = -x - c$	C
017	SBSC	$n' = n - x - c$	C

020	ANDX	$x' = x \& n$
021	ORX	$x' = x \vee n$
022	ERX	$x' = x \neq n$
023	OBEY	Obey the instruction in N
024	LDCH	$x' = n_j$
025	LDEX	$x' = n_e$
026	TXU	Set C if $n \neq x$ or $c = 1$
027	TXL	Set C if $n + c > x$

030	ANDS	$n' = n \& x$
031	ORS	$n' = n \vee x$
032	ERS	$n' = n \neq x$
033	STOZ	$n' = 0$
034	DCH	$n_j' = x_3$
035	DEX	$n_e' = x_e$
036	DSA	$n_a' = x_a$
037	DLA	$n_m' = x_m$

040	MPY	$x' = n.x$	V
041	MPR	$x' = n.x$ rounded, x^* spoiled	V
042	MPA	$x' = n.x + x^*$	V
043	CDB	$x' = 10.x; +n_j$	V
044	DVD	$x^* = x/n, x' = \text{Remainder}$	V
045	DVR	$x^* = x/n$ rounded, $x' = \text{Remainder}$	V
046	DVS	$x^* = x^*/n, x' = \text{Remainder}$	V
047	CBD	$x' = 10.x; n_j' = \text{Character}$	V

NOTATION

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

X is an accumulator (registers 0-7).

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

F is 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 .

x, m are the contents of X, M respectively.

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

n^* is the content of $N(M) + 1$

X^* is the accumulator $X + 1$ ($X7^* = X0$)

x^* is the content of X^* .

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

x^{**} is the content of x^* after an instruction has been obeyed.

n^* is the content of $N(M) + 1$ after an instruction has been obeyed.

x, n are double length numbers in $X, X + 1$, and $N, N + 1$ 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_e is the least significant 9 bits of x . The exponent of a floating point number occupies this portion of the second word.

x_a is the least significant 12 bits of x (the N address of an instruction).

x_v 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_{em} is the least significant 22 bits of x (the modifier in extended mode).

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

x_d is the least significant 7 bits of x_c

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

N_i is the most significant 2 bits of the 12 bit N address.

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

DIRECTIVES

The following lists indicate to which category each PLAN directive belongs and the versions of PLAN to which each is applicable.

MAJOR DIRECTIVES	PLAN
#CMODE	3, 4
#ELASTIC	4
#FINISH	2, 3, 4
#LOWER	1, 2, 3, 4
#MACRO	3, 4
#OMIT	3, 4
#OVERLAY	3, 4
#PERIPHERAL	1, 2, 3, 4
#PERMANENT	3, 4
#PMODE	4
#PROGRAM	1, 2, 3, 4
#STOP	3
#UPPER	2, 3, 4
MINOR DIRECTIVES	PLAN
Program Area Directives	
#COMPLETE	3, 4
#CUE	1, 2, 3, 4
#ENTRY	1, 2, 3, 4
#MONITOR	3, 4
General Purpose Directives	
#DEFINE	1, 2, 3, 4
#ERRORSEG	4
#HMODE	4
#LIBRARY	3, 4
#ORDER	3, 4
#OUST	1
#PAGE	1, 2, 3, 4
#SET	2, 3, 4
#SWITCH	2, 3
# (Comment directive)	1, 2, 3, 4

MACRO INSTRUCTIONS

INSTRUCTION	EFFECT	NO. OF BASIC INSTRUCTIONS
PLAN 2, 3, AND 4		
LDPL X N	$x' = N + c$ (15 bits)	1 + 1 literal
PLAN 3 AND 4		
LDX XX* N(M)	$x' = n + c$	2
ADX XX* N(M)	$x' = x + n + c$	2
NGX XX* N(M)	$x' = -n - c$	2
SBX XX* N(M)	$x' = x - n - c$	2
STO XX* N(M)	$n' = x + c$	2
ADS XX* N(M)	$n' = n + x + c$	2
NGS XX* N(M)	$n' = -x - c$	2
SBS XX* N(M)	$n' = n - x - c$	2
BXU X $N_1(M), N_2$	If $x \neq n_1$ branch to N_2	2
BXU XX* $N_1(M), N_2$	If $x \neq n_1$: branch to N_2	3
BXE X $N_1(M), N_2$	If $x = n_1$ branch to N_2	2
BXE XX* $N_1(M), N_2$	If $x = n_1$: branch to N_2	3
BXL X $N_1(M), N_2$	If $x < n_1$ branch to N_2	2
BXL XX* $N_1(M), N_2$	If $x < n_1$: branch to N_2	3
BXGE X $N_1(M), N_2$	If $x \geq n_1$ branch to N_2	2
BXGE XX* $N_1(M), N_2$	If $x \geq n_1$: branch to N_2	3
LDSA X N(M)	$x' = n_a$	2
LDLA X N(M)	$x' = n_m$	2
ON X N(M)	Set bit N(M) of words 30 to 1	3
OFF X N(M)	Set bit N(M) of word 30 to 0	3
TEST X N(M)	Test the state of bit N(M) of word 30	3
OVER X N(M)	Invert the state of bit N(M) of word 30	4
BSP X	Backspace MTX	1 + 2 literals
BTM X	Move back past tape mark on MTX	1 + 2 literals
CLOSE X	Close MTX	1 + 2 literals
FTM X	Move forward past tape mark on MTX	1 + 2 literals
REW X	Rewind MTX	1 + 2 literals
SCR X	Open MTX and leave scratch	1 + 2 literals
UNL X	Close file and unload	1 + 2 literals
WTM X	Write tape mark on MTX	1 + 2 literals
TAPES MACROS		
PLAN 4 ONLY		
BUX XcXm N	For handling two-word counter-modifiers in extended data mode programs.	2
BDX' XcXm N		2
BCHX XcXm N		2
LDCM XcXm N	For loading a two-word counter-modifier	2

PSEUDO OPERATIONS

INPUT	-	Input a record
INDIS	-	Distribute input records
OUT	-	Distribute and output fields
SDUMP	-	Dump state of program onto a storage device.

SD MACRO INSTRUCTIONS

SD MACRO INSTRUCTIONS	Equivalent MT Macros	
SDBSS	Back to start-of-subfile sentinel	MTBSS
SDBTS	Back to user sentinel	MTBTS
SDBUF	Set up user's buffers	MTBUF
SDCLB	Close bucket/batch early	MTCLB
SDCLS	Close a subfile	MTCLS
SDCRE	Close reel/cassette early	MTCRE
SDDEF	Define a file	MTDEF
SDDEL	Delete record	
SDEND	Close file	MTEND
SDEXT	Extend or contract file	
SDFES	Forward to end-of-subfile sentinel	MTFES
SDFSS	Forward to start-of-subfile sentinel	MTFSS
SDFTS	Forward to user sentinel	MTFTS
SDIND	Search index tables	
SDLAB	Relabel output tape	
SDRD	Read a record	MTRD
SDRDB	Read a bucket/block	MTRDB
SDRDP	Set pointer to start of record	MTRDP
SDRRB	Reverse read a block	MTRRB
SDSUS	Check previous transfer to file	MTSUS
SDWR	Write updated record	MTWR
SDWRB	Write a bucket/block	MTWRB
SDWRI	Insert new record	MTWR
SDWRS	Write user sentinel	MTWRS
SDWRU	Write unchanged record	MTWR
SDWSS	Write start-of-subfile sentinel	MTWSS

OVERLAY MACRO INSTRUCTIONS

ENTER	Ascertains which overlay unit contains the specified cue; if this overlay unit is already in core store, branches to a specified location in it; otherwise, brings it into store before branching.
RECAL	Ascertains which overlay unit contains the specified cue, brings it into core store and branches to a specified location in it.
BRING	Brings an overlay unit into core store (unless it is already there), but does not enter it

International Computers Limited
 Southern Support Department
 85-91 Upper Richmond Road
 London
 SW15