

Summarised **Programming** Information

x! = n + c

x' = n + c

n' = x + c

x' = x + n + c

1900

Series

PLAN

000

001

002

003

004

005

006

007

010

011

012

013

014

015

016

017

020

021

036

037

040

041

042

043

044

045

046

047

DSA

DLA

MPY

MPR

MPA

CDB

DVD

DVR

DVS

CBD

LDX

ADX

NGX

SBX

LDXC

ADXC

NGXC

SBXC

STO

ADS

NGS

SBS

STOC

ADSC

NGSC

SBSC

ANDX

ORX

V $x^{\dagger} = -n - c$ x' = x - n - c٧ C x' = x + n + cC x' = -n - cC C x! = x - n - c

٧

٧

V V V V

n' = n + x + cn! = -x - cn' = n - x - cn' = x + cC C n' = n + x + cC C

n' = -x - cn' = n - x - cx! = x & nx' = x v n $x' = x \neq n$ Obey the instruction in N

 $x' = n_i$ $x' = n_o$ Set C if $n \neq x$ or c = 1Set C if n+c>x

025 LDEX 026 TXU 027 TXL 030 ANDS n' = n & xORS 031 n' = n v x032 **ERS** $n' = n \neq x$ 033 STOZ n' = 0

 $n_a' = x_a$

 $n_{m'} = x_{m}$

x:' = n.x

x:'=n,x+x*

 $x:' = 10.x: + n_i$

024 LDCH

034 DCH $n_i' = x_3$ 035 DEX $n_{\rho}! = x_{\rho}$

x' = n.x rounded, x^* spoiled

 $x^{*} = x:/n, x' = Remainder$

 $x^* = x^*/n$, x' = Remainder

 $x:' = 10.x:, n_i' = \text{Character}$

 $x^{*'} = x:/n$ rounded, x! = Remainder

022 ERX 023 OBEY

V

V

V

V

V

٧

٧

 $N_t = 0$

 $N_t = 1$ $N_t = 2, 3$ $N_t = 0$ 112 $N_t = 1$ $N_t = 2$ $N_{t} = 3$ *114 *116

 $N_t = 0$

 $N_f = 1$

 $N_t = 0$

 $N_t = 1$

 $N_t = 2$

 $N_f = 3$

 $N_f = 2, 3$

110

111

113

*115

*117

LDN ADN NGN SBN LDNC ADNC NGNC SBNC

SLC

SLL

SLA

SRC

SRL

SRA

SRAV

NORM

MVCH

SLC

SLL

SLA

SRC

SRL

SRA

SRAV

NORM

SMO

A remains unchanged. x' = -N - cx' = x - N - c $x^t = N + c$ x' = x + N + c

x' = -N - c

x' = x - N - c

Normalize x

Normalize x:

Transfer N characters

Shift x left N_s places. Circular

Shift x left N_s places. Logical

Shift x left No places. Arithmetic

Shift x right N_s places. Circular

Shift x right No places. Logical

Shift x right N_S places. Special

Shift x: left Ns places. Circular

Shift x: left No places. Logical

Shift x: left N_s places. Arithmetic

Shift x: right N_S places. Circular

Shift x: right N_S places. Logical

Shift x: right Ns places. Special

Shift x: right N_s places. Arithmetic

Supplementary modifier to next instruction

Shift x right N_S places. Arithmetic

Branch to N if FOYR is set x' = N + cx' = x + N + c

Branch to N if $a \neq 0$ Branch to N if $a \ge 0$ Branch to N if a < 0Branch to N if FOVR is clear

Branch unconditionally Branch if V is set Branch if V is set and clear V Branch if V is clear Branch if V is clear or clear V Branch if C is set Branch if C is clear Branch if V is clear and/or invert V

 $x_{m}' = x_{m} + 1$

Subroutine Entry

Link in X

Link in X

Subroutine Exit Conditional Branch to N:-

V ·

C

C

C

 $x_c' = x_c - 1$

 $x_{d}^{\prime} = x_{d} - 1$

 $x_m' = x_m - 1$

Branch to N [if $x_c' \neq 0$]

Branch to N if $x_d \neq 0$

Branch to N if $x_m' \neq 0$

V

٧ V V

Single

length

V

V

V

V

Double

length

BZE

BNZ

BPZ

BNG

BUX

BDX

BCT

070

072

074

*076

BCHX

Branch to N if x = 0

Branch to N if $x \neq 0$

Branch to N if $x \ge 0$

Branch to N if x < 0

Character modify:

 $x_k = 3$

CALL

EXIT

BRN

BVS

BVSR

BVC

BVCR

BCS

BCC

BVCI

BFP

X = 0

X = 1

X = 2

X = 3

X = 4

X = 5

Single word modify: $x_m' = x_m + 1$

Double word modify: $x_m' = x_m + 2$

 $x_k = 0$, 1 or 2 $x_k' = x_k + 1$, $x_m' = x_m$

 $x_k' = 0$

Count least significant 15 bits of X.

050

052

054

056

†060

†062

†064

*066

Test floating point accumulator or floating point overflow, and branch accordingly. Branch to N if a = 0

X = 0

X = 1

X = 2

X = 3

X = 4

X = 5

X = 6

X = 7

120 ANDN	.00			165		
122	120	ANDN	x' = x & N	24-bit ICL 1900 Series word		
No.	121	ORN	x' = x v N		X F A	$Norx_a$
125		ERN	$x' = x \neq N$		3 7 2	2 12
126 MOUE Set are comparation mode Set are comparation mode Set are comparation mode Set are comparation of the set and the set					У Г	
ORDERS 127 SUM X = Sum of N words from address x* x* = Sum of				DDANCH	X F	
130			1		3 6	. 15
130					X F M	$f N_t N_s$
1313	127	SUM	X. = Sum of /v words from addless x	SHIFT		
132 FA	* 130	FLOAT	Convert n: from fixed to floating and store in A		, ,	
132	* 131	FIX	Convert a from floating to fixed and store in N(M) and	(S	
133 FSB				DOLIDIE	1	22
**134 FSP at a -n;					1	23
*136 LFP a' = g): *136 LFP a' = g): *137 SFP n: '= a' = 0, when X = 1 *139 SFP n: '= a' = 0, when X = 1 *150 SUSBY *151 REL *152 DIS *152 DIS *153 Unassigned *154 CONT *155 SUSPP *156 ALLOT *157 SERI *160 NNO *158 DISPIPATION *159 Suspend and dump program on a specified peripheral *150 SUSPY *150 SUSPY *151 SUSPP *152 DIS *153 Unassigned *154 CONT *155 SUSPP *156 ALLOT *157 SERI *160 NNO *157 DEFI *161 INNO *157 DEFI *161 INNO *158 DELTY *162 DELTY *164 DELTY *164 DELTY *164 DELTY *164 DELTY *165 DELTY *165 DELTY *165 DELTY *166 DELTY *166 DELTY *166 DELTY *167 DELTY *167 DELTY *167 DELTY *168 DELTY *169 DELTY *160 DELTY *160 DELTY *160 DELTY *161 DELTY *160 DELTY *			}	FIXED	0	
* 136						22
*136 LFPZ				NOMBER	1	23
* 137 SFP				1	S	
**162 X N(M) SUSA Suspend and specified peripheral is active peripheral of the console typewriter without suspension on the console typewriter without suspension the console typewriter without suspension the console typewriter without suspension on the console typewriter without suspension on the console typewriter without suspension on the console typewriter without suspension the console typewriter without suspension on the console typewriter without su						
Suspend if a specified peripheral is active	* 137	SFPZ	n'' = a, a' = 0, when $X = 1$	FLOATING		23
151 REL Release a specified peripheral 152 DIS Disenages a specified peripheral 153 Unassigned 154 CONT Read more program from a specified peripheral 155 SUSDP 156 ALLOT Assign, or supply information about, a specified peripheral 157 PERI Initiate action on a peripheral according to control area N(M) 157 PERI Unitiate action on a peripheral according to control area N(M) 158 USSTY 160 1 N(M) DISTY 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 believe program and treat message as console directive vibroul suspension on the console typewriter without suspension				POINT	0	x_e
Section Sect				NUMBER	1 14	q
153 Unassigned 154 CONT Read more program from a specified peripheral 155 SUSDP 156 ALLOT Assign, or supply information about, a specified peripheral 157 PERI Initiate action on a peripheral according to control area N(M) 160 O N(M) SUSTY 160 1 N(M) DISTY 160 2 N(M) DELTY 161 0 N(M) SUSSYT 161 1 N(M) DISTY 161 1 N(M) DISP 161 1 N(M) DISP 161 2 N(M) DELTY 162 X N(M) DEL 163 X N(M) AUTO 165 X N(M) AUTO 165 X N(M) AUTO 165 X N(M) GIVE 166 X N(M) GIVE 166 X N(M) GIVE N(M) = 8 Give current member awaiting reactivation, by AUTO. 165 X N(M) GIVE N(M) = 8 Give date in binary in X N(M) = 3 Give current core store allocation to that specified in X N(M) = 8 Give current address mode and branch mode to those specified in X N(M) = 8 Give current address mode and branch mode to those specified in X N(M) = 8 Give current address mode and branch mode to those specified in X N(M) = 8 Give current address mode and branch mode to those specified in X N(M) = 8 Give current address mode and branch mode to those specified in X N(M) = 8 Give current address mode and branch mode to those specified in X N(M) = 8 Give current address mode and branch mode to those specified in X N(M) = 8 Give current address mode and branch mode to those specified in X N(M) = 8 Give current address mode and branch mode to those specified in X N(M) = 8 Give current address mode and branch mode to those specified in X N(M) = 8 Give current address mode and branch mode in X N(M) = 8 Give current address mode and branch mode to those specified in X N(M) = 8 Give current address mode and branch mode in X N(M) = 8 Give current address mode and branch mode in X N(M) = 8 Give current address mode and branch mode in X N(M) = 8 Give current address mode and branch mode in X N(M) = 8 Give current address mode and branch mode in X N(M) = 8 Give current address mode and branch mode in X N(M) = 8 Give current address mode and branch mode in X N(M) = 8 Give current address mode and branch mode in X N(M) = 8 Give current address mode and branch mode in X N(M) = 8 Give c				l l	1	
* 160 0 N(M) SUSTY Supend and dump program on a specified peripheral Suspend and dump program on a specified peripheral Suspend and dump program on a specified peripheral of file. 157 PERI Initiate action on a peripheral according to control area N(M) SUSTY Supend 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 Suspend and type message on console directive Suspend and Suspend		DIS		15AM	Xc	x _m
*165 SUSDP Suspend and dump program on a specified peripheral 156 ALLOT Assign, or supply information about, a specified peripheral 260 N(M) SUSTY PERI Initiate action on a peripheral according to control area N(M) 160 O N(M) SUSTY Suspend and type message on console typewriter without suspension 160 2 N(M) DELTY Type message on console typewriter without suspension 160 2 N(M) DELTY Delete program and treat message as console directive 161 N(M) DISP Type DISPLAY n _g (as two characters) on the console typewriter without suspension 161 2 N(M) DISP Type DISPLAY n _g (as two characters) on the console typewriter bypewriter without suspension 161 2 N(M) DELT Delete program and type DELETED n _g (as two characters) on the console typewriter without suspension 161 2 N(M) DEL Delete program and type DELETED n _g (as two characters) on the console typewriter without suspension 161 2 N(M) DEL Delete program and type DELETED n _g (as two characters) on the console typewriter without suspension 161 2 N(M) DEL Delete program and type DELETED n _g (as two characters) on the console typewriter without suspension 161 2 N(M) DELTED n _g (as two characters) on the console typewriter without suspension 161 2 N(M) DELTED n _g (as two characters) on the console typewriter without suspension 161 2 N(M) DELTED n _g (as two characters) on the console typewriter without suspension 161 2 N(M) DELTED n _g (as two characters) on the console typewriter without suspension 161 2 N(M) DELTED n _g (as two characters) on the console typewriter without suspension 161 2 N(M) DELTED n _g (as two characters) on the console typewriter without suspension 161 2 N(M) DELTED n _g (as two characters) on the console typewriter without suspension 161 2 N(M) DELTED n _g (as two characters) on the console typewriter without suspension 161 2 N(M) DELTED n _g (as two characters) on the console typewriter without suspension 161 2 N(M) DELTED n _g (as two characters) on the console typewriter without suspension 161 2 N(M) DELTED n _g (as two characters) on the		CONT	-	COUNTER-	9	15
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 160 1 N(M) DISTY Type message on console typewriter 161 0 N(M) SUSTY 161 0 N(M) DISTY 161 0 N(M) DISTY 161 1 N(M) DISP 162 X N(M) DELTY 161 1 N(M) DISP 162 X N(M) DEL 163 X N(M) DEL 164 2 N(M) SUSMA 16 m² = 0, make n² y \(\phi \) and n/ = x, and omit next instruction. 164 2 N(M) SUSMA 16 m² = 0, make n² y \(\phi \) and n/ = x, and omit next instruction. 164 1 0 SUSAR 164 1 0 SUSAR 16 S N(M) GIVE N(M) = 0 Give date in binary in X N(M) = 3 Give current member awaiting flag-setting interrupt or AUTO. N(M) = 3 Give current core store allocation to that specified in X. N(M) = 3 Give current core store allocation to that specified in X. N(M) = 3 Give current core store allocation to that specified in X. N(M) = 3 Give current address mode and branch mode to those specified in X. N(M) = 3 Give current address mode and branch mode to those specified in X. N(M) = 3 Give current address mode and branch mode to those specified in X. N(M) = 3 Give current address mode and branch mode to those specified in X. N(M) = 3 Give current address mode and branch mode to those specified in X. N(M) = 3 Give current address mode and branch mode to those specified in X. N(M) = 3 Give current address mode and branch mode to those specified in X. N(M) = 3 Give current address mode and branch mode to those specified in X. N(M) = 3 Give current address mode and branch mode to those specified in X. N(M) = 3 Give current address mode and branch mode to those specified in X. N(M) = 3 Give current address mode and branch mode to those specified in X. N(M) = 3 Give current address mode and branch mode to those specified in X. N(M) = 3 Give current address mode and branch mode to those specified in X. N(M) = 3 Give current address mode and branch mode to those specified in X. N(M) = 3 Give current address mode and branch mode to those specified in X.				MODIFIER		
petipheral or file. Initiate action on a peripheral according to control area 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 ng (as two characters) on the console typewriter without suspension 161 1 N(M) DISP Type DISPLAY ng (as two characters) on the console typewriter without suspension 161 2 N(M) DEL Delete program and type DELETED ng (as two characters) on the console typewriter without suspension 162 X N(M) SUSMA Suspend and type DELETED ng (as two characters) on the console typewriter without suspension 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 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) = 5 Give datails of Executive and central processor N(M) = 8 Give current address mode and branch mode to those specified in X N(M) = 9 Alter address mode and branch mode to those specified in X N(M) = 9 Cape and type message on console typewriter without suspension of the console typewriter without suspension on the console typewriter without suspen				22 A M	0	x _{em}
160 0 N(M) SUSTY Suspend and type message on console typewriter without suspension 160 2 N(M) DISTY Type message on console typewriter without suspension 161 0 N(M) DISTY 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 without suspension 161 1 N(M) DISP Type DISPLAY n_a (as two characters) on the console typewriter without suspension 22AM CHARACTER COUNTER 161 2 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 without suspension 161 2 N(M) DISP Type DISPLAY n_a (as two characters) on the console typewriter without suspension 161 2 N(M) DISP Type DISPLAY n_a (as two characters) on the console typewriter without suspension 161 2 N(M) DISP Type DISPLAY n_a (as two characters) on the console typewriter without suspension 161 2 N(M) DISP Type DISPLAY n_a (as two characters) on the console typewriter without suspension 161 2 N(M) DISP Type DISPLAY n_a (as two characters) on the console typewriter without suspension 161 2 N(M) DISP Type DISPLAY n_a (as two characters) on the console typewriter without suspension 161 2 N(M) DISP Type DISPLAY n_a (as two characters) on the console typewriter without suspension 161 2 N(M) DISP Type DISPLAY n_a (as two characters) on the console typewriter without suspension 161 2 N(M) DISP Type DISPLAY n_a (as two characters) on the console typewriter without suspension 161 2 N(M) DISP Type DISPLAY n_a (as two characters) on the console typewriter without suspension 161 2 N(M) DISP Type DISPLAY n_a (as two characters) on the console typewriter without suspension 161 2 N(M) DISP Type DISPLAY n_a (as two characters) on the console typewriter without suspension 161 2 N(M) DISP Type DISPLAY n_a (as two characters) on the console typewriter the cater on the console typewriter the cater on the console typewriter the cater on th			peripheral or file.	COUNTER-	2	22
160 1 N(M) DISTY 179e message on console typewriter without suspension 160 2 N(M) DELTY 161 0 N(M) SUSWT 161 0 N(M) DISP 179e DISPLAY n _A (as two characters) on the console typewriter without suspension 161 1 N(M) DISP 179e DISPLAY n _A (as two characters) on the console typewriter without suspension 161 2 N(M) DEL 161 2 N(M) DEL 162 N(M) DEL 163 N(M) DEL 164 2 N(M) DEL 165 N(M) DEL 165 N(M) DEL 165 N(M) DEL 166 X N(M) SUSMA 165 N(M) DEL 166 X N(M) SUSMA 166 X N(M) SUSMA 167 N(M) SUSMA 168 N(M) SUSMA 169 N(M) SUSMA 169 N(M) SUSMA 160 N(M)	157	PERI	Initiate action on a peripheral according to control area N(M)	MODIFIER		
160 2 N(M) DELT Delete program and type MALTED n_d (as two characters) on the console typewriter without suspension 161 1 N(M) DISP Type DBPLAY n_d (as two characters) on the console typewriter without suspension 161 2 N(M) DEL Delete program and type DELETED n_d (as two characters) on the console typewriter without suspension 161 2 N(M) DEL Delete program and type DELETED n_d (as two characters) on the console typewriter without suspension 161 2 N(M) DEL Delete program and type DELETED n_d (as two characters) on the console typewriter without suspension 161 2 N(M) SUSMA If $n^* = 0$, make $n^* \neq 0$ and $n^* = x$, and omit next instruction. 16 2 N(M) AUTO Activate member x at N(M). For reactivation, N(M) must be zero. Notes The function codes 140 to 147 are undefined CHARACTER POSITIONS The function samp set the carry register but cannot cause overflow. The 043 order sets C. The carry register C is left clear by any order except 023, 117 and 123, unless that order sets C. The carry register C is left clear by any order except 023, 117 and 123, unless that order sets C. The carry register C is left clear by any order except 023, 117 and 123, unless that order sets C. The carry register C is left clear by any order except 023, 117 and 123, unless that order sets C. The carry register C is left clear by any order except 023, 117 and 123, unless that order sets C. The carry register C is left clear by any order except 023, 117 and 123, unless that order sets C. The carry register C is left clear by any order except 023, 117 and 123, unless that order sets C. The carry register C is left clear by any order except 023, 117 and 123, unless that order sets C. The carry register C is left clear by any order except 023, 117 and 123, unless that order sets C. The carry register C is left clear by any order except 023, 117 and 123, unless that order sets C. The carry register C is left clear by any order except 023, 117 and 123, unless th	160 0 N(M)	SUSTY	Suspend and type message on console typewriter		x_k x_d	x _m _
160 2 N(M) DELTY Delete program and type HALTED n_a (as two characters) on the console typewriter to the console typewriter without suspension on the console typewriter on the console typewriter without suspension on the console typewriter before the possible of typewriter without suspension on the console typewriter before the possible of the p	160 1 N(M)	DISTY	Type message on console typewriter without suspension		2 7 .	15
the console typewriter Type DISPATA** 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 without suspension * 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 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) = 3 Give current core store allocation to that specified 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 to those specified in X. N(M) = 9 Alter address mode and branch mode to those specified in X. N(M) = 8 Give current address mode and branch mode to those specified in X. N(M) = 8 Give current address mode and branch mode to those specified in X. N(M) = 8 Give current address mode and branch mode to those specified in X. N(M) = 8 Give current address mode and branch mode to those specified in X. N(M) = 8 Give current address mode and branch mode to those specified in X. N(M) = 8 Give current address mode and branch mode to those specified in X. N(M) = 8 Give current address mode and branch mode to those specified in X. N(M) = 8 Give current address mode and branch mode to those specified in X. N(M) = 8 Give current address mode and branch mode to those specified in X. N(M) = 8 Give current address mode and branch mode to those specified in X. N(M) = 8 Give current address mode and branch mode to those specified in X. N(M) = 8 Give current address mode and branch mode to those specified in X. N(M) = 8 Give current address mode and branch mode to those specified in X. N(M) = 8 Give current address mode and branch mode to those specified in X. N(M) = 8 Give current address mode and branch mode to those specified in X. N	160 2 N(M)	DELTY	Delete program and treat message as console directive			
161 1 N(M) DISP Type DISPLAY n_d (as two characters) on the console typewriter without suspension 161 2 N(M) DEL Delete program and type DELETED n_d (as two characters) on the console typewriter * 162 X N(M) SUSMA If $n^* \neq 0$, 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) = 2 Give time in characters in XX* N(M) = 3 Give current core store allocation to that specified in X. N(M) = 8 Give current address mode and branch mode in X N(M) = 8 Give current 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) * 180 X = 0 Read request block into store at	161 0 N(M)	SUSWT			x _k	x _{em}
typewriter without suspension 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 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) = 2 Give time in characters in XX* N(M) = 3 Give current core store allocation to that specified in X. N(M) = 5 Give datalis of Executive and central processor N(M) = 8 Give current address mode and branch mode to those specified in X N(M) = 9 Alter address mode and branch mode to those specified in X N(M) = 8 Give current show those specified in X N(M) = 8 Give current show those specified in X N(M) = 8 Give current show those specified in X N(M) = 9 Alter address mode and branch mode to those specified in X N(M) = 8 Give current show those specified in X N(M) = 8 Give current show those specified in X N(M) = 8 Give current show those specified in X N(M) = 8 Give current show those specified in X N(M) = 8 Give current show those specified in X N(M) = 9 Alter address mode and branch mode to those specified in X N(M) = 8 Give current show those specified in X N(M) = 8 Give current show those specified in X N(M) = 8 Give current show those specified in X N(M) = 8 Give current show those specified in X N(M) = 8 Give current show those specified in X N(M) = 8 Give current show those specified in X N(M) = 8 Give current show those specified in X N(M) Show the show	161 1 N(M)	DISP			2	22
Determine program and type BLETED 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. If $n^* \neq 0$, proceed to 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) = 3 Give current 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) * 168 X N(M) RRQ X = 0 Read request block into store at N(M) * 169 X N(M) RRQ X = 0 Read request block into store at N(M) * 160 X N(M) RRQ X = 0 Read request block into store at N(M) * 160 X N(M) RRQ X = 0 Read request block into store at N(M) * 160 X N(M) RRQ X = 0 Read request block into store at N(M) * 160 X N(M) RRQ X = 0 Read request block into store at N(M) * 160 X N(M) RRQ X = 0 Read request block into store at N(M) * 160 X N(M) RRQ X = 0 Read request block into store at N(M) * 160 X N(M) RRQ X = 0 Read request block into store at N(M) * 160 X N(M) RRQ X = 0 Read request block into store at N(M) * 160 X N(M) RRQ X = 0 Read request block into store at N(M) * 160 X N(M) RRQ X = 0 Read request block into store at N(M) * 160 X N(M) RRQ X = 0 Read request block into store at N(M) * 160 X N(M) RRQ X = 0 Read request block into store at N(M) * 160 X N(M) RRQ X = 0 Read request block into store at N(M) * 160 X N(M) RRQ X = 0 Read request block into store at N(M) * 160 X N(M) RRQ X = 0 Read request block into store at N(M) * 160 X N(M) RRQ X = 0 Read request block into store at N(M) * 160 X N(M) RRQ X = 0 Read request block into store at N(M) * 16			typewriter without suspension			10.00
* 162 X N(M) SUSMA If $n^* = 0$, make $n^{**} \neq 0$ and $n' = x$, and omit next instruction. If $n^* \neq 0$, proceed to next instruction. * 163 X N(M) AUTO Activate member X at N(M). For reactivation, N(M) must be zero. * 164 1 0 SUSAR * 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 to that specified in X. N(M) = 8 Give current address mode and branch mode to those specified in X N(M) = 9 Alter address mode and branch mode to those specified in X * 166 X N(M) * 168 X N(M) RRQ X = 0 Read request block into store at N(M) * 168 X N(M) RRQ X = 0 Read request block into store at N(M) * 169 Alter core store allocation to store at N(M) * 160 X N(M) RRQ If $n^* = 0$, make $n^* \neq 0$ and $n' = x$, and omit next instruction. 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 ware overflow. * These instructions with particular machine configurations see the Central Processors manual. * 1 In 22-bit address mode these instructions operate on X_{em} instead of X_{em} and brance 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 after the specification of some instructions subsequent to the date of going to print, so a close watch on User Notices and relevant manual amendments is recommended. * 166 X N(M) RRQ * 2 0 Read request block into store at N(M) * 168 Give Current address mode and branch mode to those specified in X.	161 2 N(M)	DEL	Delete program and type DELETED n_a (as two characters)	CHARACTER	n_0 n_1	n ₂ n ₃
instruction. If n*≠0, proceed to 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 flag-setting interrupt or 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 to that specified in X. N(M) = 8 Give current address mode and branch mode in X N(M) = 8 Give current 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) ** 163 X N(M) RRQ X = 0 Read request block into store at N(M) ** 164 I O SUSAR Suspend current member awaiting flag-setting interrupt or AUTO. The function codes 140 to 147 are undefined The carry register C is left clear by any order except 023, 117 and 123, unless that order ests C. These instructions may cause overflow. The carry register Du to 147 are undefined The carry register Du to 147 are undefined The carry register Du to 143 are carry register but cannot cause overflow. The carry register Du to 2. The carry regis	* 162 X N(M)	SUSMA			6 6	6 6
*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 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 to those specified in X *166 X N(M) RRQ X = 0 Read request block into store at N(M) *168 X N(M) RRQ X = 0 Read request block into store at N(M) *169 X N(M) RRQ X = 0 Read request block into store at N(M) *160 X N(M) R						
*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 to those specified in X N(M) = 9 Alter address mode and branch mode to those specified in X X = 0 Read request block into store at N(M) *166 X N(M) RRQ X = 0 Read request block into store at N(M) *164 X N(M) RRQ SUSSIN Suspend current member awaiting reactivation by AUTO. *165 X N(M) GIVE Suspend current member awaiting flag-setting interrupt or AUTO. *166 X N(M) GIVE N(M) = 0 Give date in binary in X These instructions may cause overflow. *166 X N(M) RRQ 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. *165 X N(M) These instructions may cause overflow. *166 X N(M) = 1 Give date in binary in X These instructions are not available with some machines or with some Executives. *166 X N(M) = 2 Give time 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 to those specified 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) *166 X N(M) RRQ X = 0 Read request block into store at N(M) *168 X N(M) RRQ X = 0 Read request block into store at N(M) *169 X N(M) RRQ X = 0 Read request block into store at N(M) *160 X N(M) RRQ X = 0 Read request block into store a			If $n^* \neq 0$, proceed to next instruction.			
* 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 to those specified 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) * 104 2 0 SUSIN Suspend current member awaiting reactivation by AUTO. The carry register C is left clear by any order except 023, 117 and 123, unless that order sets 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. 1 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 specification of some instructions subsequent to the date of going to print, so a close watch on User Notices and relevant manual amendments is recommended. * 166 X N(M) RRQ X = 0 Read request block into store at N(M) * 10 RRQ X = 0 Read request block into store at N(M) * 10 RRQ X = 0 Read request block into store at N(M)	* 163 X N(M)	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 to those specified in X N(M) = 9 Alter address mode and branch mode to those specified in X * 166 X N(M) * 166 X N(M) * 168 X N(M) * 169 X N(M) * 160 X N(M)	* 164 1 0	SUSAR			set the carry register but canno	t cause overflow. The 043
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) * 18 Q * 2 O Read request block into store at N(M) * 10 Give date in binary in X * These instructions are not available with some machines or with some Executives. * These instructions are not available with some machines or with some Executives. * These instructions are not available with some machines or with some Executives. * These instructions are not available with some machines or with some Executives. * These instructions are not available with some machines or with some Executives. * These instructions are not available with some machines or with some Executives. * These instructions are not available with some machines or with some Executives. * These instructions are not available with some machines or with some Executives. * These 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 specification of some instructions subsequent to the date of going to print, so a close watch on User Notices and relevant manual amendments is recommended. * 10 EX PA				The carry register C is le	eft clear by any order except 02	23, 117 and 123, unless that
* These instructions are not available with some machines or with some Executives. 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) * 1 Req. * 1 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. † 1 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 specification of some instructions subsequent to the date of going to print, so a close watch on User Notices and relevant manual amendments is recommended. * 166 X N(M) * RRQ X = 0 Read request block into store at N(M) * 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. † 18 Can address mode these instructions operate on X_{em} instead of X_{em} 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 Processor manual and the Plan Ref			or AUTO.			
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) * 166 X N(M) * 17 Rea availability of instructions with particular machine configurations see the Central Processors manual. * 1 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 specification of some instructions subsequent to the date of going to print, so a close watch on User Notices and relevant manual amendments is recommended. * 166 X N(M) * 18 Give current address mode and branch mode to those specified in X N(M) = 9 Alter address mode and branch mode to those specified in X * 10 Central Processors manual. * 1 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 specification of some instructions subsequent to the date of going to print, so a close watch on User Notices and relevant manual amendments is recommended. * 10 Expression in X and the Central Processor manual and the Plan Reference Manual. Hardware and software developments may alter the specification of some instructions are defined here in direct branch mode to the central Processor manual. * 10 Expression in X and the Central Processor manual and the Cent	165 X N(M)	GIVE		,		o or with some Everytives
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) * 10 Central Processors manual. 1 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 specification of some instructions subsequent to the date of going to print, so a close watch on User Notices and relevant manual amendments is recommended. * 166 X N(M) RRQ RRQ X = 0 Read request block into store at N(M) Branch instructions are defined here in direct branch mode terms only. For extended				These histractions are in		S Of With Some Executives.
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						inne configurations see the
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 * 106 X N(M) * 107 RRQ * 107 RRQ * 107 RRQ * 107 RRQ * 107 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 specification of some instructions subsequent to the date of going to print, so a close watch on User Notices and relevant manual amendments is recommended. * 106 X N(M) * 107 RRQ * 107 RRQ * 107 RRQ * 108 RRQ				† In 22-bit address mode	these instructions operate on X	em instead of x_m and branch
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 * 166 X N(M)						
Plan Reference Manual. Hardware and software developments may alter the specification of some instructions subsequent to the date of going to print, so a close watch on User Notices and relevant manual amendments is recommended. * 166 X N(M) RRQ X = 0 Read request block into store at N(M) Branch instructions are defined here in direct branch mode terms only. For extended						
those specified in X * 166 X N(M) RRQ * 106 X N(M) * 106 X N(M) * 106 X N(M) RRQ * 107 Some instructions subsequent to the date of going to print, so a close watch on oser Notices and relevant manual amendments is recommended. Branch instructions are defined here in direct branch mode terms only. For extended				Plan Reference Manual. Hard	ware and software development	s may alter the specifications
* 166 X N(M) RRQ X = 0 Read request block into store at N(M) Branch instructions are defined here in direct branch mode terms only. For extended						, so a close watch on User
	* 166 X N(M)	RRQ				erms only. For extended
			X = 1 Replace request block from store at N(M)			
L					-	on the same of the

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

1. 12 4 Idiletion

C is the carry register

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

V is the overflow register.

 \boldsymbol{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

n' is the content of M(M) + 1

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

 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.

 n^{-1} 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.

 $x_{\it e}$ 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_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_{em} is the least significant 22 bits of x (the modifier in extended mode).

 x_k is the most significant 22 bits of x, used in character modifying with end-around-

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

 N_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

LDCM

XcXm N

	INSTRUCTION). OF BASIC STRUCTIONS	
	PLAN 2, 3	3, AND 4				
	LDPL	X	N	x' = N + c (15 bits)	1 + 1 literal	
	PLAN 3 A	ND 4				
	LDX	XX*	N(M)	x:' = n: + c	2	
	ADX	XX*	N(M)	x: t = 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 $x : \langle n_1 : \text{branch to } N_2 \rangle$	3	
	BXGE	X	$N_1(M), N_2$	If $x \ge n_1$ branch to N_2	2	
	BXGE	XX*	$N_1(M), N_2$	If $x \ge n_1$: branch to N_2	3	
	LDSA	X	N(M)	$x' = n_{\alpha}$	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		
1	(BSP	X	- ()	Backspace MTX	1 + 2 literals	
	BTM	X		Move back past tape mark on MTX	1 + 2 literal	
	CLOSE	X		Close MTX	1 + 2 literals	
	FTM	X		Move forward past tape mark on MTX	I + 2 literals	
₹	REW	X		Rewind MTX	1 + 2 literals	
	SCR	X		Open MTX and leave scratch	1 + 2 literal	
	UNL	X		Close file and unload	1 + 2 literals	
	WTM	X		Write tape mark on MTX	1 + 2 literals	
	PLAN 4 ONLY					
	BUX BDX' BCHX	XcXm XcXm XcXm	N N	For handling two-word counter— modifiers in extended data mode	2 2 2	

For loading a two-word counter-modifier

PSEUDO OPERATIONS

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 MACR	O 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		
OVERLA	Y MACRO INSTRUCTIONS			
ENTER	Ascertains which overlay unit contai unit is already in core store, branche otherwise, brings it into store before	s to a specified location in it;		
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			

211221	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