

A1.1

DIRECTLY ADDRESSED FETCH & STORE INSTRUCTIONS  
(THREE SYLLABLES)

A1.1

User Code	Effect on Qq			Final State of			Transfer Address
	C	I	M	N1	N2	N3	
Yy	-	-	-	x	a	b	Yy
YyMq	-	-	-	x	a	b	Y(y + Mq)
YyMqQ	Cq-1	-	Mq+Iq	x	a	b	Y(y + Mq)
=Yy	-	-	-	b	c	d	Yy
=YyMq	-	-	-	b	c	d	Y(y + Mq)
=YyMqQ	Cq-1	-	Mq+Iq	b	c	d	Y(y+ Mq)

In any of the above, Yy may be replaced by any of  
YAy, YBy... YZy, Vv, Ww, Ee, Rr, RrPp, RrLl.

A1.2

INDIRECTLY ADDRESSED FETCH & STORE INSTRUCTIONS  
(TWO SYLLABLES)

A1.2

User Code	Effect on Qq			Final State of			Transfer Address
	C	I	M	N1	N2	N3	
Mq' Mq	-	-	-	x	a	b	Mq' + Mq
Mq' MqQ	Cq-1	-	Mq+Iq	x	a	b	Mq' + Mq
Mq' MqN	-	-	-	x	a	b	Mq' + Mq + 1
Mq' MqQN	Cq-1	-	Mq+Iq	x	a	b	Mq' + Mq + 1
Mq' MqH	-	-	-	x	a	b	Mq' + $\frac{1}{2}$ Mq
Mq' MqQH	Cq-1	-	Mq+Iq	x	a	b	Mq' + $\frac{1}{2}$ Mq
Mq' MqHN	-	-	-	x	a	b	Mq' + $\frac{1}{2}$ Mq + 1
Mq' MqQHN	Cq-1	-	Mq+Iq	x	a	b	Mq' + $\frac{1}{2}$ Mq + 1
=Mq' Mq	-	-	-	b	c	d	Mq' + Mq
=Mq' MqQ	Cq-1	-	Mq+Iq	b	c	d	Mq' + Mq
=Mq' MqN	-	-	-	b	c	d	Mq' + Mq + 1
=Mq' MqQN	Cq-1	-	Mq+Iq	b	c	d	Mq' + Mq + 1
=Mq' MqH	-	-	-	b	c	d	Mq' + $\frac{1}{2}$ Mq
=Mq' MqQH	Cq-1	-	Mq+Iq	b	c	d	Mq' + $\frac{1}{2}$ Mq
=Mq' MqHN	-	-	-	b	c	d	Mq' + $\frac{1}{2}$ Mq + 1
=Mq' MqQHN	Cq-1	-	Mq+Iq	b	c	d	Mq' + $\frac{1}{2}$ Mq + 1

Notes:-  $0 \leq q \leq 15$      $0 \leq q' \leq 15$      $C0 = I0 = M0 = 0$

N.B. The symbol + after a quoted operation time indicates the instruction requires a variable time extra for standardisation of results.

User Code	Time μSec.	Final State of			Remarks
		N1	N2	N3	
+	1	x	c	d	$x = b+a$
+F	7+	x	c	d	
+D	3	x	y	e	
+DF	13+	x	y	e	$x, y = c, d + a, b.$
-	1	x	c	d	
-F	7+	x	c	d	$x = b-a$
-D	3	x	y	e	
-DF	13+	x	y	e	$x, y = c, d - a, b$
NEG	1	x	b	c	
NEGF	2+	x	b	c	$x = -a$
NEGD	2	x	y	c	
NEGDF	5+	x	y	c	$x, y = -a, b$
x	16	x	c	d	
xF	19	x	c	d	$x = a.b$
xD	14	x	y	c	
xDF	18	x	y	c	$x, y = a.b$
x+F		x	y	e	$x, y = a.b + c, d$
$\frac{b}{a}$	35+	x	c	d	$x = b/a \quad (-1 \leq x < 1)$
$\frac{b}{a}F$	30+	x	c	d	$x = b/a$
$\frac{b}{a}D$	35+	x	d	e	$x = b, c/a \quad (-1 \leq x < 1)$
$\frac{b}{a}DF$	30+	x	d	e	$x = b, c/a$
$\frac{b}{a}R$	35+	x	r	d	$b, c = a.x + 2^{-47}r \quad (-1 \leq x < 1)$
$\frac{b}{a}I$	35+	r	x	c	$b = a.x + r$
ROUND	1	x	c	d	$x = a, b$ rounded off to
ROUND $\frac{a}{2}$	2+	x	c	d	single-length
ROUND $\frac{a}{4}$		x	b	c	$x = a$ rounded off to
ROUND $\frac{a}{8}$		x	b	c	half-length.
ABS	1	x	b	c	
ABSF	2+	x	b	c	$x = /a/$
STAND	2+	x	b	c	$x = a$ in standard floating form.
FLOAT		x	c	d	$x = b \times 2^a$
FLOAT D		x	y	d	$x, y = b, c \times 2^a$
FIX		x	y	b	$y.2^x = a.$



User Code	Time μSec.	Final State of			Remarks
		N1	N2	N3	
MAX		x	y	c	x is max. of a and b.
MAXF		x	y	c	y is min. of a and b. overflow set if a and b reversed.
STR (stretch)	3	x	y	b	y = a with DO=0; x = 48 copies of sign digit
CONT (contract)	4	x	c	d	x = b with sign digit of a.
SIGN		x	c	d	( x = +1 if b > a; x = 0 if b = a; x = -1 if b < a. overflow not set.
SIGNF		x	c	d	
NOT	1	x	b	c	x = 1's complement of a x = a OR b
OR	2	x	c	d	
AND	2	x	c	d	x = a AND b x = a NOT EQUIVALENT to b.
NEW BITS	1	x	c	d	
	26	x	b	c	x = number of non-zero digits in a.
ERASE	2	b	c	d	Nest up one place, remove a.
ZERO	3	0	a	b	Fetch word of all zeros.
DUP	3	a	a	b	Duplicate a
DUPD	5	a	b	a	Duplicate a,b
REV	2	b	a	c	Reverse a and b
REVD	4	c	d	a	Reverse a, b and c, d.
PERM CAB	2	b	c	a	Interchange abc cyclically. Equivalent to two PERMS.
		c	a	b	
DUMMY		a	b	c	No effect
VR		a	b	c	Clear overflow setting.
=TR		b	c	d	Set test register equal to DO of a.

User Code	Time μSec.	Effect on Nesting Store			Remarks
		N1	N2	N3	
Jr		a	b	c	Jump to ref. r in current level.
JPp		a	b	c	Jump to first instr. of Pp.
JLl		a	b	c	" " " " " Ll.
JrPp		a	b	c	" " instr. labelled r in Pp.
JrPO		a	b	c	" " " " r in main programme.
JrLl		a	b	c	Jump to instr. labelled r in Ll.
JSr		a	b	c	
JSPp		a	b	c	As above, but sending the
JSLl		a	b	c	address of the next instruction
JSrPp		a	b	c	(return address) to the SJNS.
JSrPO		a	b	c	
JSrLl		a	b	c	
EXIT		a	b	c	Leave subroutine. Jump to address in top cell of SJNS.
EXITN		a	b	c	As above, but jumping to address 3 syllables on from that stored in SJNS.
JrV		a	b	c	Jump to ref. r if overflow set. Clear overflow.
JrNV		a	b	c	Jump to ref. r if overflow not set. Clear overflow.
JrTR Clear test register.		a	b	c	Jump to ref. r if test register set.
JrNTR Clear test register.		a	b	c	Jump to ref. r if test register not set.
JrEN		a	b	c	Jump to ref. r if nesting store empty.
JrNEN		a	b	c	Jump to ref. r if nesting store not empty.
JrEJ		a	b	c	Jump to ref. r if SJNS empty.
JrNEJ		a	b	c	Jump to ref. r if SJNS not empty.
JrCqZ		a	b	c	Jump to ref. r if Cq is zero.
JrCqNZ		a	b	c	Jump to ref. r if Cq not zero.
JrCqNZS (Two syllable).		a	b	c	Jump to ref. r if Cq not zero, but as short loop jump.
Jr>Z		b	c	d	Jump to ref. r if $a > 0$ .
Jr≥Z		b	c	d	" " " " " $a \geq 0$ .



JUMP INSTRUCTIONS (THREE-SYLLABLES)

User Code	Time uSec.	Effect on Nesting Store			Remarks
		N1	N2	N3	
Jr = Z		b	c	d	Jump to ref. r if a = 0.
Jr $\neq$ Z		b	c	d	" " " " a $\neq$ 0.
Jr < Z		b	c	d	" " " " a < 0.
Jr $\leq$ Z		b	c	d	" " " " a $\leq$ 0.
Jr =		b	c	d	" " " " a = b
Jr $\neq$		b	c	d	" " " " a $\neq$ b
OUT		a	b	c	End program if a = 0 Return to Director if a $\neq$ 0

Jump Instructions

User Code	Time $\mu$ Sec.	Final State of			Final State of Qq	Remarks
		N1	N2	N3		
Qq		Qq	a	b	No change	Fetch Qq
=Qq		b	c	d	a	Store in Qq
=+Qq		b	c	d	Qq+a	Add to Qq (as fixed pt numbers)
Cq		Cq	a	b	No change	Fetch Cq
=Cq		b	c	d	L(a)/Iq/Mq	Store L(a) in Cq.
=RCq		b	c	d	L(a)/1/0	Reset Cq.
=+Cq		b	c	d	x/Iq/Mq	x = Cq + L(a)
NCq		a	b	c	x/Iq/Mq	x = -Cq.
DCq		a	b	c	x/Iq/Mq	x = Cq - 1
Iq		Iq	a	b	No change	Fetch Iq.
=Iq		b	c	d	Cq/L(a)/Mq	Store L(a) in Iq
=RIq		b	c	d	0/L(a)/0	Reset Iq.
=+Iq		b	c	d	Cq/x/Mq	x = Iq + L(a).
Iq=+1		a	b	c	Cq/+1/Mq	Set Iq = + 1
Iq=+2		a	b	c	Cq/+2/Mq	Set Iq = + 2
Mq		Mq	a	b	No change	Fetch Mq
=Mq		b	c	d	Cq/Iq/L(a)	Store L(a) in Mq
=RMq		b	c	d	0/1/L(a)	Reset Mq
=+Mq		b	c	d	Cq/Iq/x	x = Mq + L(a)
M+Iq		a	b	c	Cq/Iq/x	x = Mq + Iq

User Code	Time $\mu$ Sec.	Final State of			Final State of Qq	Remarks
		N1	N2	N3		
Qq' TOQq					Qq'	Transfer one, two or all three parts of Qq' to corresponding parts of Qq.
Cq' TOQq					Cq'/Iq/Mq	
Iq' TOQq					Cq/Iq'/Mq	
Mq' TOQq					Cq/Iq/Mq'	
IMq' TOQq					Cq/Iq'/Mq'	Qq' is unaltered
CMq' TOQq					Cq'/Iq/Mq'	
CIq' TOQq					Cq'/Iq'/Mq	$1 \leq q \leq 15.$ $0 \leq q' \leq 15.$

Notes:

L(a) means least sig. 16 bits of a.

The three parts of the Q store Cq/Iq/Mq are each signed 16-bit binary numbers. NO SPILL OCCURS FROM ONE PART TO ANOTHER.

$$0 \leq q \leq 15.$$

Note. Q0 = C0 = I0 = M0 = 0



(TOB and FRB are 1 syllable)

User Code	Time $\mu$ Sec.	Final State of			Remarks
		N1	N2	N3	
=LINK		b	c	d	Transfer address in N1 to top cell of SJNS.
LINK		x	a	b	x = address in top cell of SJNS.
TOB		x	c	d	x = a converted from characters to binary using radix system b.
FRB		x	c	d	x = a converted from binary to characters using radix system b.

User Code	Time $\mu$ Sec.	Final State of			Remarks
		N1	N2	N3	
SHA+n		x	b	c	$x = a \cdot 2^{+n}$
SHACq		x	b	c	$x = a \cdot 2^{Cq}$
SHAD+n		x	y	c	$x, y = (a, b) 2^{+n}$ sign changes.
SHADOq		x	y	c	$x, y = (a, b) 2^{Cq}$ sign changes.
SHL+n		x	b	c	x = a or x, y = a, b shifted logically - as a digit pattern with no numerical associations.
SHLCq		x	b	c	
SHLD+n		x	y	c	
SHLDCq		x	y	c	
SHC+n		x	b	c	x = a shifted cyclically - digits leaving at one end appear at the other.
SHCCq		x	b	c	

User Code	Time $\mu$ Sec.	Final State of			Remarks
		N1	N2	N3	
x+		x	y	e	$x, y = a \cdot b + c, d$
x++n		x	y	e	$x, y = (a \cdot b) 2^{+n} + c, d$
x+Cq		x	y	e	$x, y = (a \cdot b) 2^{Cq} + c, d$

$$1 \leq q \leq 15$$

$$-64 \leq n \leq +63$$

<u>Magnetic Tape</u>	33,000 characters/second.
MFRQq	Magnetic Forward Read
MFREQq	Magnetic Forward Read to End of Message (EM)
MBRQq	Magnetic Backward Read
MBREQq	Magnetic Backward Read to EM
MWQq	Magnetic Write (Write is Always Forward)
MWEQq	Magnetic Write to EM
MLWQq	Magnetic Write as "last block"
MLWEQq	Magnetic Write to EM as "last block"
MGAPQq	
MFSKQq	Magnetic Forward Skip specified number of blocks
MBSKQq	Magnetic Backward Skip specified number of blocks
MRWDQq	Magnetic Rewind
MLBQq	Transfer "last block" marker to TR.
METQq	Transfer "End of Tape" marker to TR
MBTQq	Transfer "Beginning of tape" marker to TR
MANUALQq	Set tape unit NOT READY. (For tape changing)
<u>Paper Tape</u>	Input 1000 char./second. Output 110 char./sec.
PRQq	Paper Read
PREQq	Paper Read to EM
PWQq	Paper Write
PWEQq	Paper Write to EM
PWGQq	Paper Write, followed by GAP
PWGEQq	Paper Write to EM, followed by GAP
MANUALQq	Set paper tape device NOT READY
<u>Typewriter</u>	
TRQq	Typewriter read
TWQq	Typewriter write
TREQq	Typewriter Read to EM
TWEQq	Typewriter Write to EM. Change to read if ISS typed
<u>General</u>	
BUSYQq	Transfer "Busy" marker for device given by Qq to TR
INTQq	Interrupt if device given by Qq is busy
PARQq	Send parity marker for device to TR



Device Selection

Fetches actual device number into N1,  
e is programmers' number

EMTe

Magnetic Tape Station

EPRe

Paper Tape Reader

EPPe

Paper Tape Punch

ETWe

Typewriter. (Not necessary for  
interruption typewriter)

Any constant specified is converted to the correct machine code form by the Compiler and placed in its appropriate place on the machine code program tape. When this tape is read into the machine to be obeyed, the constants are already set to their starting values, therefore the instructions specifying the constants do not appear in the machine code program, or occupy any instruction space.

Any constant specified is, therefore, available from the first instruction of any program, but if it is altered by user code instructions, it will not be reset.

#### Numerical Constants

Vv = z/s      Store single-length fixed-point constant.

VvD = z/s      Store double-length fixed-point constant, which occupies constant stores v and v + 1.

Vv = Fz      Store single-length floating constant.

VvD = Fz      Store double-length floating constant, which occupies constant stores v and v + 1.

#### Binary Constants

Vv = Bt/s      Store OCTAL integer t.

#### Address Constants

Vv = AYy      Store the core address of Yy.

#### Q-store Constants

Vv = Q c/i/m      Store the three signed integers c, i, m as separate 16-bit integers. c, i or m (or any combination) may be replaced by AYy.

#### Half-Length Constants

VvU = (specification)      Store constant in D0-23 of constant store v.

VvL = (specification)      Store constant in D24-47 of constant store v.

N.B. (specification) may take any one of the forms given above for single-length, Numerical, Binary, or Address constant.

#### Valid Addresses

In any constant specification where AYy is allowed, any of the following may also appear:-

AYAy, AYBy ... AYZy

AWw



The constant will be the word and syllable address of the label quoted. If required in Q-store constant form, compiler will print a failure indication if syllable number is not zero (i.e. not an asterisk instruction).

User Code	Final State of		Remarks
	N1	N2	
SET n	n	a	n = signed integer -32768 ≤ n ≤ 32767
SET Bt	t	a	t = FIVE digit octal integer
SET AYy	AYy	a	
SET ARr	ARr	a	Must be address of asterisked instruction

Yy may be replaced by - YAy, YBy ... YZy

Ww

Vv

VvPp

VvLl

Rr may be replaced by - RrPp

RrLl

N.B. Compiler will give failure indication if any constant used with SET is not valid.

N.B. In any V-constant setting or SET instruction which sets a core store data address (i.e. AYy or its alternative), the address of a half word - i.e. the address which when in the modifier position of Qq would cause the instructions MOMqH or  $\approx$  MOMqH to fetch or store that particular half-word - may be set by writing U or L after the address. Thus "YyU" would be the address of the more significant half of Yy, YyL of the less significant half. There is an implicit restriction that Yy must not exceed 16,383, otherwise the modifier would be treated as negative.



Dec. Value	Octal Value	Symbol		Remarks
		Case Normal	Case Shift	
0	00	Space	Space	
1	01			
2	02	CR-LF	CR-LF	
3	03			
4	04	Tab	Tab	
5	05	Backspace	Backspace	
6	06	Case Shift	Case Shift	
7	07	Case Normal	Case Normal	
8	10			
9	11			
10	12			
11	13			
12	14			
13	15			
14	16			
15	17	/	:	
16	20	0	;	
17	21	1	{ or $\frac{1}{4}$	
18	22	2	} or "	
19	23	3	< or $\frac{3}{4}$	
20	24	4	> or @	
21	25	5	=	
22	26	6	≤ or $\frac{1}{2}$	
23	27	7	≥	
24	30	8	(	
25	31	9	)	
26	32	x or 10	Suffix 10 or %	
27	33	÷ or 11	£	
28	34	●	●	Item Separator Symbol (ISS)
29	35	+	≠ or &	
30	36	-	*	
31	37	•	,	

KDF 9 CHARACTER CODE (CONTINUED)

Dec. Value	Octal Value	Symbol		Remarks
		Case Normal	Case Shift	
32	40			
33	41	A	a	
34	42	B	b	
35	43	C	c	
36	44	D	d	
37	45	E	e	
38	46	F	f	
39	47	G	g	
40	50	H	h	
41	51	I	i	
42	52	J	j	
43	53	K	k	
44	54	L	l	
45	55	M	m	
46	56	N	n	
47	57	O	o	
48	60	P	p	
49	61	Q	q	
50	62	R	r	
51	63	S	s	
52	64	T	t	
53	65	U	u	
54	66	V	v	
55	67	W	w	
56	70	X	x	
57	71	Y	y	
58	72	Z	z	
59	73			Used for End Data Symbol (ED)
60	74			Used for End File Symbol (EF)
61	75	↑	↑	Used as End Message (EM) or in Algol
62	76			Used if required as Escape character
63	77			<u>Not</u> the same as Delete