ICL

Facts

4100
Computer system
Technical information

Contents

1 Instructions and times

The following tables list the available instructions with their execution times for the direct form; where there is no direct form, the time for the literal variant is given. The variants available are listed in the variants column. Their times can be found by adding the appropriate constants given below:

Time in micro-seconds to be added		*Variant*					*With SCU approx % time to be added*
		:S (short)	*:L (literal)*	*:M (modified)*	*:I (indirect)*	*End Around*	
4120	6μs	+0	-4.9	+1.1	+6.0	+8.2	+5
4120	2μs	+0	-1.7	+1.1	+2.8	+4.5	+10
4130	6μs	+0	-6.0	+0.75	+6.0	+7.3	+5
4130	2μs	+0	-1.5	+0.75	+2.25	+3.5	+12

There are a few exceptions to this, but the above will serve as a general guide. For exact details reference must be made to the Technical Manual.

Notation Used

Symbol	*Meaning*
M	Main accumulator (24 bits)
R	Reserve accumulator (24 bits)
S	Sequence Control register (17 bits)
K	Count register (12 bits)
C	Conditions register (24 bits, bits 17-7 not used). Note that C is not always set to the final value of the quantities given in the table. C24 is set according to the true sign of the answer.
I	Normal Interrupt word (12 bits)
A	Attention Interrupt word (12 bits)
FPA	Floating point accumulator (39-bit mantissa, 9-bit exponent). On a 4130, this is a hardware register giving a 48-bit mantissa and 12-bit exponent.
DPA	Double length floating point accumulator (87-bit mantissa, 9-bit exponent).
CPA	Complex floating point accumulator
m,r,k,s	The contents of registers M, R, K, S
fpa,dpa,cpa	The contents of FPA,DPA,CPA
N	Value of address part of instruction
n,n_i	Number held in location with address N, bit i of this number
Q	Operand – either N,n, number held in location with address n, or N+r according to variant (see y)

dQ	The contents of double length floating point operand (held in Q, Q+1, Q+2, and Q+3)
tQ	The contents of triple length operand (held in Q, Q+1 and Q+2)
cQ	The contents of complex operand (held in Q, Q+1, Q+2 and Q+3)
fQ, (Q, Q+1)	The contents of floating point operand, double length number held in Q and Q+1
(r,m)	The contents of double length integer held in R and M
C_i,K_i,m_i,r_i	Bit i of number held in C,K,m or r.
aff	Affected in unspecified manner
a, b, c, d	The 4 6-bit characters which make up one word stored with the character a at the most significant end
-	Condition remains unaltered
y	Represents the 3rd octal digit of long instructions when this can vary. This is even for normal functions, odd for extracodes. y = 0 or 1 for literal, y = 2 or 3 for direct, y = 4 or 5 for modified, y = 6 or 7 for indirect.

Function Code (Octal) short	long	Mnemonic	Effect	Variants	Conditions C_{24} Neg	C_{23} St	C_{22} NZ	C_{21} Ca	C_{20} Of	Time in Microseconds 4120 6µs	4120 2µs	4130 6µs	4130 2µs
00	40y	ADD	m:=m+Q	:S:L:M:I	m	m	m	m	m	12.0	5.6	12.0	4.5
01	41y	SUB	m:=m−Q	:S:L:M:I	m	m	m	m	m	12.0	5.6	12.0	4.5
02	42y	NADD	m:=Q−m	:S:L:M:I	m	m	m	m	m	12.0	5.6	12.0	4.5
03	43y	LD	m:=Q	:S:L:M:I	m	m	m	—	—	12.0	5.6	12.0	4.5
04	44y	LDR	r:=Q	:S:L:M:I	r	r	r	—	—	12.0	5.6	12.0	4.5
05		JIR	†s:=n;C_{24-18}:=n_{24-18}		n_{24}	n_{23}	n_{22}	n_{21}	n_{20}	12.0	5.6	12.0	4.5
	450	J	S:=N		—	—	—	—	—	7.1	3.9	6.0	3.0
	45y	JI	S:=Q	:M:I	—	—	—	—	—	12.0	5.6	12.0	4.5
06	46y	AND	m:=m *and* Q	:S:L:M:I	m	m	m	—	—	12.0	5.6	12.0	4.5
07	47y	ANDN	m:=m *and not* Q	:S:L:M:I	m	m	m	—	—	12.0	5.6	12.0	4.5

† JIR is a short instruction only, although :S is not used. Note therefore that N $\leq$ 63.

Function Code (Octal) short	long	Mnemonic	Effect	Variants	Conditions C_{24} Neg	C_{23} ST	C_{22} NZ	C_{21} Ca	C_{20} Of	Time in Microseconds 4120 6μs	4120 2μs	4130 6μs	4130 2μs
10	50y	ADDR	r:=r+Q	:S:L:M:I	r	r	r	r	r	12.0	5.6	12.0	4.5
11	51y	SUBR	r:=r−Q	:S:L:M:I	r	r	r	r	r	12.0	5.6	12.0	4.5
12	52y	NADR	r:=Q−r	:S:L:M:I	r	r	r	r	r	12.0	5.6	12.0	4.5
	530	JFL*	$0':=C_{24-18}+s;s:=s+N$		—	—	—	—	—	14.2	7.8	14.0	4.8
13	53y	JIL	$0':=C_{24-18}+s;$**s:=Q	:S :M:I	—	—	—	—	—	18.0	8.4	18.0	6.25
14	54y	LDK	k:=Q	:S:L:M:I †	aff	aff	k	—	—	12.0	5.6	12.0	4.5
	55y	COMP	set C_{24-21} from (m−Q)	:L:M:I	m−Q	m−Q	m−Q	m−Q	—	12.0	5.6	12.0	4.5
16	560	JF*	s:=s+N	:S	—	—	—	—	—	7.1	3.9	6.0	3.0
16	56y	JA	s:=s+Q	:S:L:M:I †	—	—	—	—	—	12.0	5.6	12.0	4.5
17	570	JB*	s:=s−N	:S	—	—	—	—	—	7.1	3.9	6.0	3.0
17	57y	JS	s:=s−Q	:S:L:M:I †	—	—	—	—	—	12.0	5.6	12.0	4.5
20	600	JN*	s:=s+N if C_{24}=1	:S ††	—	—	—	—	—	7.1	3.9	6.0	3.0
21	610	JNN*	s:=s+N if C_{24}=0	:S ††	—	—	—	—	—	7.1	3.9	6.0	3.0
22	620	JZ*	s:=s+N if C_{22}=0	:S ††	—	—	—	—	—	7.1	3.9	6.0	3.0
23	630	JNZ*	s:=s+N if C_{22}=1	:S ††	—	—	—	—	—	7.1	3.9	6.0	3.0
24	640	JST*	s:=s+N if C_{23}=1	:S ††	—	—	—	—	—	7.1	3.9	6.0	3.0
25	650	JOF*	s:=s+N if C_{20}=1; then C_{20}:=0	:S ††	—	—	—	—	—	7.1	3.9	6.0	3.0
27	670	DKJN*	If K_{12}=0 then K:=k−1 If K_{12}=1 then s:=s+N (if k=0 initially both conditions will be true)	:S ††	—	—	—	—	—	7.1	3.9	6.0	3.0

* When using NEAT, address phrase is a label.
† The short forms of LDK, JA and JS are literal.
** JIL 0 sets S from location 0 before storing the former value of C_{24-18}+S in location 0.
†† Both the long and short forms of JN, JNN, JZ, JNZ, JST, JOF and DKJN are literal.

Function Code (Octal) short	long	Mnemonic	Effect	Variants	Conditions C_{24} Neg	C_{23} St	C_{22} NZ	C_{21} Ca	C_{20} Of	Time in Microseconds 4120 6μs	4120 2μs	4130 6μs	4130 2μs
30	60y	ST	Q:=m	:S :M:I	Q	Q	Q	–	–	13.1	6.7	12.75	5.25
31	61y	STR	Q:=r	:S :M:I	Q	Q	Q	–	–	13.1	6.7	12.75	5.25
32	62y	NEGS	Q:=-Q	:S :M:I	Q	Q	Q	Q	Q	13.1	6.7	12.75	5.25
33	63y	SUBS	Q:=Q-m	:S :M:I	Q	Q	Q	Q	Q	13.1	6.7	12.75	5.25
34	64y	ADDS	Q:=Q+m	:S :M:I	Q	Q	Q	Q	Q	13.1	6.7	12.75	5.25
35	65y	CLS	Q:=0	:S :M:I	0	0	0	–	–	13.1	6.7	12.75	5.25
36	66y	INCS	Q:=Q+1	:S :M:I	Q	Q	Q	Q	Q	13.1	6.7	12.75	5.25
37	67y	DECS	Q:=Q-1	:S :M:I	Q	Q	Q	Q	Q	13.1	6.7	12.75	5.25
	70y	GET	Q:=Q(bcda); m:=m(abc)Q(a)	:M:I	Q†	Q	Q	–	–	13.1	6.7	12.75	5.25
	71y	PUT	Q:=Q(bcd)m(d)	:M:I	Q†	Q	Q	–	–	13.1	6.7	12.75	5.25
	72y	DIVM	m:=(r,m)/Q;r:= positive remainder	:L:M:I*	m	m	m	–	m	68.1	61.7	29.0	24.0
	73y	MULM	(r,m):=r+mxQ (Q treated as an unsigned integer i.e. $0 \leqslant Q < 2^{24}$)	:L:M:I*	r	r	r	–	–	67.0	60.6	22.0	15.0
	74y	MVE	**m:=Q;r:=r+1;r':=m	:M:I	m	m	m	–	r	19.1	9.5	18.0	6.8
	75y	MVB	**Q:=m:=r';r':=r+1	:M:I	m	m	m	–	r	20.1	10.6	18.5	7.3
	76y	EXC	Q⇌m	:M:I	Q	Q	Q	–	–	14.2	7.8	12.75	5.25
	77y	EXCR	Q⇌r	:M:I	Q	Q	Q	–	–	14.2	7.8	12.75	5.25

* DIVM and MULM destroy the contents of K.

† C_{24} is set by the initial not the final value of Q after GET or PUT.

** In MVB, the meaning of Q depends on the original value of r, not on the new value, r + 1. r' means the contents of the location with address r.

Shift Instructions (function 15)

Function Address Phrase (Octal)		Mnemonic	Effect	Conditions					Times in Microseconds			
									4120		4130	
				C_{24} Neg	C_{23} St	C_{22} NZ	C_{21} Ca	C_{20} Of	6μs	2μs	6μs	2μs
15	0	SRL	Shift r left k places	r	r	r	–	r				
15	1	SRLA	Shift r left around k places	r	r	r	–	–				
15	2	SRR	Shift r right k places	r	r	r	–	–				
15	3	SRLC	Shift r k 6-bit characters left around	r	r	r	–	–				
15	4	SML	Shift m left k places	m	m	m	–	m				
15	5	SMLA	Shift m left around k places	m	m	m	–	–				
15	6	SMR	Shift m right k places	m	m	m	–	–			6.5+	5.5+
15	7	SMLC	Shift m k 6-bit characters left around	m	m	m	–	–	8.2+ 1.1k	5.0+ 1.1k	$0.75\{\frac{k}{4}\}$	$0.75\{\frac{k}{4}\}$
15	12	SRRL	Shift r right logical k places	r	r	r	–	–				
15	16	SMRL	Shift m right logical k places	m	m	m	–	–				
15	20	SRST	Shift r until standardised, or k places, whichever is less	r	r	r	–	–				
15	24	SMST	Shift m until standardised, or k places, whichever is less	m	m	m	–	–				
15	40	SBL	Shift both r and m left k places	r	r	r	–	r				
15	42	SBR	Shift both r and m right k places	r	r	r	–	–			6.5+	5.5+
15	52	SBRL	Shift both r and m right logical k places	r	r	r	–	–	8.2+ 2.2k	5.0+ 2.2k	$0.75\{\frac{k}{4}\}$	$0.75\{\frac{k}{4}\}$
15	60	SBST	Shift both r and m until standardised, or k places, whichever is less.	r	r	r	–	–				

The above shift instructions all exist in short form only. They are written without :S and have no address part. C_{24} is set according to the value of bit 24 immediately before the final bit (or six bits in the case of a character shift) has been shifted on shift left. If k is negative or zero, then the effect on C is undefined. If k is greater than 63, the result of the shift instruction is undefined. If m or r is already standardised, then C is undefined after a shift until standardised. $\{\frac{k}{4}\}$ means the next integer larger than $\frac{k}{4}$. Shift times on the 4130 calculated by the above formula will in any case be only approximate.

Register moves (function 700)

Function Address Phrase (Octal)	Mnemonic	Effect	Conditions					Time in Microseconds			
			C_{24} Neg	C_{23} St	C_{22} NZ	C_{21} Ca	C_{20} Of	4120		4130	
								6μs	2μs	6μs	2μs
700 00020	KTOR	r:=k	0	0	aff	0	—	7.1	3.9	6.0	3.0
700 00402	MTOR	r:=m	r	r	r	0	—				
700 00404	STOR	r:=s	0	0	r	0	—				
700 00441	CAIR	r:=r+1 if carry set	r	r	r	r	r				
700 00541	CADR	r:=r−1 if carry set	r	r	r	r	r				
700 01001	RTOM	m:=r	m	m	m	0	—				
700 01003	MORR	m:=m *or* r	m	m	m	0	—				
700 01004	STOM	m:=s	0	0	m	0	—				
700 01010	CTOM	m:=c	m	m	m	0	—				
700 02001	RTOS	s:=r	—	—	—	—	—				
700 02002	MTOS	s:=m	—	—	—	—	—				
700 04000	CLC*	C_{24-18}:=0	0	0	0	0	0				
700 04002	MTOC	c:=m	m_{24}	m_{23}	m_{22}	m_{21}	m_{20}				
700 06001	RTSP*	s:=r_{17-1}; C_{19-18}:=r_{19-18}	aff	aff	aff	aff	aff				
700 06002	MTSP*	s:=m_{17-1}; C_{19-18}:=m_{19-18}	aff	aff	aff	aff	aff				
700 10001	RTOK	k:=r	r	r	r	0	—				
700 10002	MTOK	k:=m	m	m	m	0	—				
700 10201	RNTK	k:=−r	−r	−r	−r	0	—				
700 21000	ITOM*	m:=interrupt word	m	m	m	0	—				
700 41000	ATOM*	m:=attention word	m	m	m	0	—				

In NEAT no address phrase is used with its register move mnemonics

* Under the commercial NEAT-Assembler-NEATCC-register moves CLC, RTSP, MTSP, ITOM and ATOM are restricted to software programs.

Input/Output

Function Address Phrase (Octal)	*Mnemonic*	*Effect*	*Conditions*		*Times in Microseconds*			
					4120		*4130*	
			C_{24}-C_{22}	C_{21}-C_{20}	*6μs*	*2μs*	*6μs*	*2μs*
740 000 NN	IDPR	Input data packed repetitive	–	–	10.4+ (12.3+ 4D)W	7.2+ (9.6+ 4D)W	13.6+ (12+ 4D)W	6+ (8+4D) W
740 100 NN	ODPR	Output data packed repetitive	–	–				
740 200 NN	IDUR	Input data unpacked repetitive	–	–	10.4+ (6+D) W	7.2+ (3+D) W	13.6+ (6+D) W	6+ (2.5+ +D)W
740 300 NN	ODUR	Output data unpacked repetitive	–	–				
750 000 NN	ISPR	Input status word packed repetitive	–	–	10.4+ (12.3+ 4D)W	7.2+ (9.6+ 4D)W	13.6+ (12+ 4D)W	6+ (8+4D) W
750 100 NN	OCPR	Output control word packed repetitive	–	–				
750 200 NN	ISUR	Input status word unpacked repetitive	–	–	10.4+ (6+D) W	7.2+ (3+D) W	13.6+ (6+D) W	6+ (2.5+ D)W
750 300 NN	OCUR	Output control word unpacked repetitive	–	–				
750 A*	EXEN	Entry to executive mode on a 4130	aff	aff			50	20
750 377 00	EXIT	Exit from executive mode on a 4130	(5'+3)	(5'+3)			56	22
			$_{m}^{24-22}$	$_{m}^{21-20}$				
760 200 NN	IDUM	Input data unpacked single to M	m	–				
760 300 NN	ODUM	Output data unpacked single from M	m	–				
770 200 NN	ISUM	Input status word unpacked single to M	m	–	10.4+D	7.2+D	10+D	6+D
770 300 NN	OCUM	Output control word unpacked single from M	m	–				

D is the device response time
W is the number of words input or output
NN is the channel number as two octal digits

In NEAT, the address phrase is NN (i.e. the channel number) for all input and output instructions except for EXIT (which does not have an address phrase) and EXEN.

*The address phrase of an EXEN instruction is 1BB00 or 3BB00 where BB are two octal digits; in NEAT, bit 13 is not included i.e. 750 10100 is coded as EXEN 64.

Under the commercial NEAT Assembler, NEATCC, all input/output instructions except for EXEN and EXIT are restricted to software programs.

Extracode Functions

The following extracode instructions are all those (apart from the floating point instructions) which are performed by hardware on a 4130. They are performed by software on a 4120.

Function Address Phrase (Octal)	*Mnemonic*	*Effect*	*Variants*	*Conditions* C_{24-22}	*Conditions* C_{21} *Ca*	*Conditions* C_{20} *Of*	*Time in Microseconds* 4120 6μs	4120 2μs	4130 6μs	4130 2μs
411 0	CTLA	Copy to lower address a block with, r=address of first location of block, k=no. of words to be copied, m=distance to be moved. Finally r:=m+r,k:=-1,m unaffected		aff	aff	aff	121+ 53k	60+ 28k	8.25+ 12k	6.25+ 4.5k
411 10000	CTHA	Copy to higher address a block with, r=address of last location of block k=no. of words to be copied, m=distance to be moved. Finally r:=r-m,k:=-1, m unchanged		aff	aff	aff	115+ 52k	58+ 27k	8.25+ 12k	6.25+ 4.5k
50y	MULS	*m:=m x Q	:L:M:I	m unless C_{20}=1	aff	m	262	161	36	20
51y	DIV (or DIVS)	*m:=m/Q; r:=remainder (identical to ALGOL DIV function	:L:M:I	m unless C_{20}=1	aff	m	216	138	33	27.5
52y	BL	(r,m) := (Q,Q+1)	:M:I	aff	aff	–	76	39	19	7.5
53y	WB	(Q,Q+1) := (r,m)	:M:I	aff	aff	–	90	20	20	8.5
54y	JIRX	Jump indirect and restore link	:M:I	(N+1) $_{24-22}$	(N+1) $_{21}$	(N+1) $_{20}$	164	85	40	20
55y	JIX	Jump indirect	:M:I	–	–	–	222	113	40	20
56y	JILX	Jump indirect setting link	:M:I	–	–	–	273	139	66	30
57y	INDEX	**Access chapter item with index Q placing its address in R	L:M:I	aff	aff	aff	282	153	41	21

* K affected in an unspecified manner
** M affected in an unspecified manner

Double Length Floating Point Extracodes

These may be used in scientific operating systems. They are performed by software

Function Address Phrase (Octal)		*Mnemonic*	*Effect*	*Variants*	*Conditions*			*Registers affected in unspecified manner*
					C_{24-22}	C_{21} *Ca*	C_{20} *Of*	
701	0	DN	dpa: = -dpa		aff	aff	aff	FPA,R,M,K
701	2	DCF	dpa: = integer m in double length floating point form		aff	aff	aff	FPA,R,M,K
701	4	DMOD	dpa: = modulus (dpa)		aff	aff	aff	FPA,R,M,K
701	6	DENT	m: = entier (dpa)		aff	aff	m	FPA,R,K
701	10	DSIG	if dpa < 0,m: = -1 if dpa = 0, m: = 0 if dpa > 0, m: = 1	}	m	m	m	FPA,R
701	12	FTOD	dpa: = fpa		aff	aff	aff	FPA,R,M,K
701	14	DTOF	fpa: = dpa		aff	aff	aff	DPA,R,M,K
70y		DL	dpa: = dQ	:M:I	aff	aff	aff	FPA,R,M,K
62y		WD	dQ: = dpa	:M:I	aff	aff	aff	FPA,R,M,K
72y		DA	dpa: = dpa+dQ	:M:I	aff	aff	aff	FPA,R,M,K
73y		DS	dpa: = dpa-dQ	:M:I	aff	aff	aff	FPA,R,M,K
74y		DM	dpa: = dpa x dQ	:M:I	aff	aff	aff	FPA,R,M,K
75y		DD	dpa: = dpa/dQ	:M:I	aff	aff	aff	FPA,R,M,K
76y		DCP	set $C_{22\ 24}$ from (dpa dQ)	:M:I	(dpa-dQ)	aff	aff	FPA,R,M

Floating Point Extracodes

The following instructions are performed by hardware on a 4130 but software on a 4120

Function Address Phrase (Octal)	*Mnemonic*	*Effect*	*Variants*	*Conditions*			*Time in Microseconds*			
				C_{24-22}	C_{21} *Ca*	C_{20} *Of*	*4120*		*4130*	
							6μs	*2μs*	*6μs*	*2μs*
401 0	FN	* fpa := − fpa		aff	aff	–	97	49	6.25	5.25
401 2	FCF	* fpa := integer m in floating point form		aff	aff	–	133-159	70-95	13	12
401 4	FMOD	* fpa := modulus (fpa)		aff	aff	aff	69-116	35-59	6	2.75/5††
401 6	FENT	** m := entier (fpa)		aff	aff	m	125-150	65-90	13	12
401 10	FSIG	** if fpa <0, m :=−1 if fpa =0, m := 0 if fpa >0, m := 1		C_{23} aff C_{24}, C_{22} m	aff	–	84	42	6	3.5
40y	FL	* fpa := fQ	:M:I	aff	aff	–	124	64	19	7.5
41y	WF	* fQ := fpa	:M:I	aff	aff	–	181	92	20	8.5
42y	FA	* fpa := fpa + fQ	:M:I	aff	aff	–	365	199	25	15
43y	FS	* fpa := fpa − fQ	:M:I	aff	aff	–	387	211	25	15
44y	FM	* fpa := fpa X fQ	:M:I	aff	aff	–	619	400	50	40
45y	FD	* fpa := fpa/fQ	:M:I	aff	aff	–	630	411	81	70
46y	FCP	* Set C_{24-22} from (fpa − fQ)	:M:I	C_{23} aff $C_{24,22}$ (fpa−fQ)	aff	–	249-356	125-181	18	10
60y	FLU	† fpa := tQ	:M:I	–	–	–	172	86	26	11.5
61y	WUF	† tQ := fpa	:M:I	–	–	–	165	82	26	11.5

* M, R, K affected in an unspecified manner
** R, K affected in an unspecified manner
† R affected in an unspecified manner
†† The larger figure applies if the operand is negative

Complex Floating Point Extracodes

These may be used in scientific operating systems. They are performed by software. In all cases, the conditions bits in the C register and also FPA, R, M and K are affected in an unspecified manner.

Function				
Direct	*Modified*	*Indirect*	*Mnemonic*	*Effect*
421	441	431	CL	cpa:=cQ
451	471	461	WC	cQ:=cpa
521	541	531	CA	cpa:=cpa+cQ
551	601	561	CS	cpa:=cpa-cQ
711	731	721	CM	cpa:=cpaXcQ
611	631	621	CD	cpa:=cpa/cQ

Other extracodes performed entirely by software in scientific operating systems.

Function	*Mnemonic*	*Effect*	*Location (decimal)*
64y	UXC	as defined by user	104, 105
741	GETA	get address of array } ALGOL	120
761	GETS	get address of string } ALGOL	124
67y	SET	checking and tracing	110, 111
771	TR	extracodes as defined in	126
77y	CH	DETECTIVE or CHAT	127

Extracodes performed entirely by software in commercial operating systems.

Function	*Mnemonic*	*Effect*	*Location (decimal)*
63y	GETX	Get string	103
62y	PUTX	Put string	101
64y	MVX	Copy contents of block of store	105
621	MTIO	Magnetic tape input/output	100
66y	COAX	Call overlay ahead	109
73y	UXC	As defined by user	118, 119
771	TR	Trace	126
77y	CH	Check	127

2 Error messages from NEAT assemblers

The commercial NEAT assembler gives the meanings of its error messages on the listing.

The error numbers given below are output by NEATER and the basic NEAT assemblers NEAT, NEATOUT, NEATC and NEATCOUT. N indicates an error only given by NEATER, B indicates an error only given by basic NEAT assemblers.

	Number	*Error Description*
N	2	An incorrect terminator.
B	3	Symbol contains impermissible character.
	4	An incorrect mnemonic.
	5	Impermissible character or symbol in assumed context.
B	6	Field 1, 2 or 3 incorrectly terminated or P: in wrong field.
	7	Error in floating point constant.
N	8	Block, chapter or program name is missing or not acceptable.
N	9	Field incorrectly non-empty, contents ignored.
N	10	Field incorrectly non-empty, rest of line treated as error.
N	12	Field incorrectly empty.
N	14	LIBRARY director in library item.
N	15	Library item or file cannot be found.
N	57	END director met at program level.
B	58	Store full when forming program in store at end of assembly.
	59	Too many digits in an integer.
	60	Too few digits in an integer or octal string.
	61	Impermissible character in integer or octal string.
	62	Integer overflow.
	63	Impermissible variant of function mnemonic.
	64	Store full.
	65	Address phrase out of range.
	66	Data or constant name already used at this level.
	67	Block or chapter name already used at this level.
	68	Error in C: string or terminator to C: string
B	69	Subscripted data name already used at this level.
	70	Unintroduced identifier in an address phrase.
	71	Integer after B: is 0 or greater than 24.
	72	Separator after B: not : or /
	73	: as two consecutive separators after B:
B	74	Jump to data or constant location under CODE.
B	76	Integer in field 5 not terminated by $, tabulate or newline.
	80	Label name already used (for basic assemblers, also LABEL declaration missing).
	81	Forward reference to non-existent label (given at end of appropriate level).
N	82	Attempt to redeclare a label name under LABEL director.

	Number	*Error Description*
	83	Interchapter jump to a label not declared under LABEL.
N	84	LABEL director met at other than chapter level.
	90	Backward relative jump to non-existent label.
	91	Forward relative jump to a label already introduced.
B	92	Label under CODE as address phrase of relative jump under CONST.
	93	Label name defined more than 63 half locations from a short jump to it.
N	94	Under CONST, a relative jump referring to a data name.

NEATER also gives warning messages for valid but dubious instructions. Warning numbers given are listed below.

Number	*Description*
1	Function and address not compatible e.g. ST V:-1.
2	Address phrase is an end-around constant.
3	Outshift input (it is replaced by an acute).
4	The first chapter is segmented.
5	The first chapter declaration is in a library section.
6	End of source encountered (END director assumed).
7	Floating point constant underflow.
8	Mnemonic (not INDEX) more than 4 characters (First four used).

3 ALGOL

DYNAMIC ROUTINE MODULES

DR0	Basic
DR1	Paper tape dump procedure
DR7	Decimal sterling procedures
DR8	Sterling and fixed length string procedures
DR10	Disc procedures for array input and output
DR11	Disc dump procedure
DR12	Record handling procedures for disc and magnetic tape
DR20	Plotter procedures
DR25	Binary input and output procedures
DR30	Magnetic tape procedures for array input and output
DR31	Magnetic tape dump procedure
DR32	Magnetic tape record handling procedures (T20/30 and T30C)

SUMMARY OF STANDARD PROCEDURES

In the parameters of these procedures

X	represents a real expression	Z	represents a real variable
I, J	represent integer expressions	M	represents an integer variable
Q & R	represent boolean expressions	P	represents a boolean variable
A	represents a real array	B	represents an integer array
S	represents a string		

C & H are integer variables used for channel and handler numbers respectively.

Real result

DR0	ABS(X)	EXP(X)	LN(X)
	SQRT(X)	SIN(X)	COS(X)
	ARCTAN(X)	TAN(X)	ARCSIN(X)
	ARCCOS(X)	CHECKR(X)	
DR8	LSDLIT(X, I, J)		

Integer result

DR0	ENTIER(X)	SIGN(X)	SIZE(A)
	RANGE(A, I)	LOWBOUND(A, I)	STOREMAX
	CHECKI(I)		

Boolean result

DR0	BUFFER(I, S)	CHECKB(Q)

Input (not needing backing store)

DR0	"READ" Z, M, P...	ADVANCE(I)	READER(I)
	INSTRING(B, M)	DECODE(I)	SPECIAL(4)
DR25	BINPUT(I)	INTIL(I, M)	
DR7	LSDCON(Z)	STIN(Z)	
DR8	LSDIN(Z)	FILLSTRING(B, M, I)	PUTSTRING(S, B, I)

Output (not needing backing store)

DR0	"PRINT" X, I, S, Q...	DIGITS(I)	SAMELINE
	OUTSTRING(B, M)	SCALED(I)	PREFIX(S)
	PUNCH(I)	FREEPOINT(I)	LEADZERO(S)
	SPECIAL(I)	ALIGNED(I, J)	GROUPING(I)
	CHECKS(S)	CHECKI(I)	CHECKR(X)
DR25	BOUTPUT(I, M)		
DR7	STOUT(Z, M)		
DR8	LSDOUT(Z, M)	TABSTRING(B,I,J)	

Control

DR0	WAIT	RESTART	STOP

Store dumping

DR1	PTDUMP
DR11	DDUMP(C)
DR31	MTDUMP

Plotter

DR20	SETORIGIN(I,J)	DRAWLINE(I, J)	MOVEPEN(I, J)
	CENCHAR(I)	WAY(I, J)	

Disc array input/output
(P and S represent integers giving protection and security characters)

DR10	DALLOC(C, I, P, S)	DDELETE(C)	DOPEN(C, P, S)
	DCLOSE(C)	DREAD(C, A, I, M)	DWRITE(C, A, I, M)
	DFIND(C, I)	DCHECK(C)	

Disc and magnetic tape record handling

DR12	BINARY	REND(C)	ENDFILE(C)
	RCOND(C)	RSIZE(C, I, Q)	RFIND(C, I)
	RNUMBER(C)	RCOPY(I, C)	REMAINDER(I, C)

Magnetic tape array input/output

DR30	MTSOURCE(H, S)	MTDEST(H, S, Q)	MTREAD(H, A, I, J)
	MTWRITE(H, A, I, J)	MTMARK(H)	MTCLOSE(H, I)
	MTBACK(H)	MTREWIND(H)	MTSEEK(H)
	MTCOND(H)		

DR32	FILE(H)	ENDREC(H)	ENDBLOCK(H)
	FINDREC(H)	FILECOND(H)	
	COPYREC(I, H)	RESTREC(I, H)	

Segments of NEAT Code

Any segment of NEAT code must be written as a basic statement introduced by the symbol "CODE". The lines of code have the form:

"CODE"...

% function $ address phrase
or $$$ function $ address phrase
or $$ label $ function $ address phrase
or $$ label

and the segment is terminated by the following "END", "ELSE" or semicolon.

ALGOL SYNTACTIC ERRORS DURING TRANSLATION

Number	*Error description*
1	Number of impermissible form.
2	Error in basic word.
3	Impermissible beginning to a statement.
4	Procedure declaration not terminated by a semicolon.
5	Name in declaration not terminated by semicolon or comma.
6	Name in call of "LIBRARY" not in library or more than 20 names in library call.
7	Name declared twice in same blockhead.
8	Label occurring twice in same block.
9	Item in a declaration inadmissible.
10	First item in a switch declaration not followed by:=
In a procedure declaration (11 to 21):	
11	Item following a procedure name not ; or (.
12	No ; or) after formal parameter part.
13	List in value or specification part has impermissible form.
14	Specification part occurs before value part.
15	More than 10 parameters are specified as procedures.
16	Too many parameters (more than 24).
17	Parameter not specified.
18	Recursive procedure encountered with real, integer, label or Boolean name parameter.
19	Name in value part not a formal parameter.
20	Name in specification part not a formal parameter.
21	Parameter specified twice.
22	Program name of impermissible form.
In array declaration (23 to 26):	
23	No comma or open bracket after identifier.
24	No colon between upper and lower bounds.

Number	*Error description*
25	No comma or closed bracket after bound pair.
26	Array with too many dimensions (more than 63).
27	No "END" or ; after statement in compound tail.
28	No colon after a label.
29	More than 4095 words required for variables or constants.
30	Left part variable in assignment statement not followed by:=
31	Value assigned to a procedure identifier outside procedure body, or type procedure name used within a NEAT segment.
32	Identifier not declared, or used outside scope of declaration.
33	Inadmissible complex primary in arithmetic expression.
34	Missing arithmetic operand at start of arithmetic expression.
35	Missing operand in arithmetic expression.
36	No close bracket after subscript list.
37	Unmatched brackets.
38	Wrong number of subscripts in subscripted variable.
39	Missing "ELSE".
40	Missing "THEN".
41	Conditional statement or expression after "THEN".
42	Missing or inadmissible operator in arithmetic expression.
43	Non-arithmetic operand in arithmetic expression.
44	Impermissible use of label name.
46	Inadmissible controlled variable in "FOR" statement.
47	Missing operand at start of Boolean expression.
48	Missing relational operator.
49	Missing operand in Boolean expression.
50	Inadmissible complex Boolean primary.
51	Inadmissible operator in Boolean expression.
52	Inadmissible symbol at start of an expression.
During procedure call (53 to 56):	
53	No open bracket following name of procedure with parameters.
54	Actual parameter not followed by comma or close bracket (cf. error 12).
55	Error in parameter delimiter of form)<letter string>:(.
56	No actual switch, procedure or string parameter, where one expected.
58	Controlled variable in "FOR" statement not followed by:=
60	Incorrect designational expression.
61	Arithmetic expression in "FOR" list element not followed by "STEP", "WHILE", "DO", or a comma.
62	Missing "UNTIL".
63	Program too large or complex to be compiled.
64	Occurance of a ' within inner string.
65	"COMMENT" occurs after other than a ; or a "BEGIN".
66	Conditional arithmetic expression on right-hand side of relation.
67	"GO TO" into a "FOR" loop from outside.
68	Amount of code produced for current ALGOL block exceeds 16,383 words.
69	Wrong number of parameters in procedure call.
70	Formal array parameter not replaced by array name.

Number *Error description*

71 Name output parameter replaced by a variable of a different type, a complex expression, or a constant.

72 Error in segment of NEAT code.

73 Jump to one or more non-existent labels in a NEAT code segment.

74 Error in a switch declaration.

75 Own array with variable bounds.

76 Symbol following "OWN" not "REAL", "INTEGER" or "BOOLEAN".

77 (i) In the call of a formal procedure:
The type and class of an actual parameter do not agree with that expected.

(ii) In the call of a procedure having parameters which are specified as procedures:
An actual procedure possesses parameters which are not of the same type or class as those of the corresponding formal procedure.

In the call of a procedure having parameters which are specified as procedures (78 to 80).

78 An actual procedure possesses name output parameters, and these do not correspond to simple variables in all the calls of the corresponding formal procedure (cf. error number 71).

79 An actual procedure does not have the same number of parameters as the corresponding formal procedure.

80 An actual procedure is not of the same type as the corresponding formal procedure, or is a function designator when the corresponding formal procedure is not, or vice-versa.

Any number greater than 1000
This indicates that the compiler cannot continue translation. It usually occurs after a number of other errors.

Other Errors during ALGOL Translation

(1) CARE is displayed if the comment between "END" and the following "END", "ELSE" or semi-colon does *not* have the form of a single legal ALGOL identifier.

(2) NOLABEL is displayed followed by a list of the offending label names if one or more references occur to non-existent labels or to labels outside the scope of their blocks.

(3) MTNOTACC is displayed if on compiling to store the main chapter of the compiled program is above 32K.

(4) If the program is punched on a paper tape, the absence of a halt code at the end of the tape will cause it to shoot through the reader, instead of stopping at the end.

(5) The following errors may cause the ALGOL program to be completely read, without finishing compilation.
(i) Insufficient "END"s to match all the "BEGIN"s in the program.
(ii) No semi-colon after the final "END".
(iii) Missing ' at the end of a string, causing the program statements that follow to be treated as part of the string.

(6) The following errors will cause the end of an ALGOL program to be found prematurely:

(i) Missing "BEGIN".

(ii) "END" or the comment following "END" not followed by "END" "ELSE" or semi-colon, causing a "BEGIN" to be treated as part of a comment.

ALGOL Errors During Running

After the following errors, continuation is automatic

Message	*Meaning*	*Continuation value*
READERR*	(1) Number in impermissible format	Zero
	(2) When obeying INSTRING numeric character found before the first ', or occurrence of a ' in inner string	String terminated
	(3) ' or digits read when reading boolean	"FALSE"
BUFFERR	In a call of BUFFER(I, 'S'), S is not just a single character.	"FALSE"
SQRT ERR	X<0	Zero
SIN ERR	$\|X\| > 8.5 \times 10^{11}$ (approximately)	Zero
COS ERR	$\|X\| > 8.5 \times 10^{11}$ (approximately)	Zero
TAN ERR	$\|X\| > 4.2 \times 10^{11}$ (approximately)	Largest positive number
LOG ERR	X <0 or p↑q with p<0 and q real	Zero
EXP ERR	X >254 $\log_e 2$ (i.e. 176.0)	Largest positive number
ARCS ERR	|X| > 1	Zero
ARCC ERR	|X| > 1	Zero
OUTSERR	The string being output by PREFIX or LEADZERO or OUTSTRING contains an inadmissible character. (The rest of the string is ignored).	–
PRNTERR	Non-zero real number to be output has a mantissa less in absolute value than ¼	Zero output
SWITOFLO	The value of the subscript in a switch designator is outside the range of the switch list. (The program continues with the next statement).	–
NO BINARY NO FILING (H)	A call of BINARY or FILE (H) has occurred outside a "READ" or "PRINT" statement and has been ignored	–

* When inputting from reader 1 or reader 2, after a READERR, the operator should type a fullstop to continue or X. to abandon the run.

Errors after which there is no continuation

Message	*Meaning*
INTOFLO	The result of some integer operation is outside the range -2^{23} to $2^{23}-1$. N.B. This may not be detected immediately it occurs.
SUBOFLO	The subscript of a subscripted variable is outside the declared range or a switch designator used as an actual parameter has subscript out of range.
PARAMERR	An attempt is made to enter a procedure with the wrong number of parameters.
RECN ERR	An attempt is made to enter recursively a procedure having "REAL", "INTEGER", BOOLEAN" or "LABEL" parameters called by name.
ENTOFLO	Integer overflow caused during a call of ENTIER
LOWBDERR	In a call of LOWBOUND, the second parameter is greater than the number of subscripts of the array.
RANGERR	In a call of RANGE, the second parameter is greater than the number of subscripts of the array.
BOUNDERR	An attempt has been made to allocate an array with a row too large, or the lower bound outside the range −256 to +255 or greater than the upper bound.

Note that the systems messages detailed in pages 34 to 36 may also be output.

4 FORTRAN

FORTRAN DYNAMIC ROUTINES (BACKING STORE SYSTEMS)

DRS	basic
COMPLE	complex extracodes
MOTHY	double precision extracodes
FIORBS	disc and magnetic tape input/output
FIORD	disc input/output
FIORM	magnetic tape input/output
DSEG	disc segment loader
MTSEG	magnetic tape segment loader

SUMMARY OF STANDARD FUNCTIONS

In the parameters of the following functions

X and Y	represent real expressions
I and J	represent integer expressions
D and E	represent double precision expressions
C	represents a complex expression

Integer	*Real*	*Double Precision*	*Complex*
	EXP(X)	DEXP(D)	CEXP(C)
	ALOG(X)	DLOG(D)	CLOG(C)
	ALOG10(X)	DLOG10(D)	
	SIN(X)	DSIN(D)	CSIN(C)
	COS(X)	DCOS(D)	CCOS(C)
	TANH(X)		
	SQRT(X)	DSQRT(D)	CSQRT(C)
	ATAN(X)	DATAN(D)	
	ATAN2(X, Y)	DATAN2(D, E)	
IABS(I)	ABS(X)	DABS(D)	
	CABS(C)		
INT(X)	AINT(X)		
IFIX(X)	FLOAT(I)		
IDINT(D)	SINGL(D)	DBLE(X)	
	REAL(C)		
	AIMAG(C)		CMPLX(X, Y)
MOD(I, J)	AMOD(X, Y)	DMOD(D, E)	CONJG(C)
ISIGN(I, J)	SIGN(X, Y)	DSIGN(D, E)	
IDIM(I, J)	DIM(X, Y)		
MAX0(I, J . . .)	AMAX0(I, J . . .)		
MAX1(X, Y . . .)	AMAX1(X, Y . . .)	DMAX1(D, E . . .)	
MIN0(I, J . . .)	AMIN0(I, J . . .)		
MIN1(X, Y . . .)	AMIN1(X, Y . . .)	DMIN1(D, E . . .)	

SUMMARY OF STANDARD SUBPROGRAMS

In the following

E,PAGE,X,Y,D,N,L,C,B,P,S and T are integer variables, constants or expressions

C = channel number, P = protection,S = security (P and S 6-bit characters)

Plotter

ORIGIN(E,PAGE)	set origin E steps from margin. Orientation: PAGE = 0, X-axis E-W; PAGE = 1, N-S.
DRAW(X, Y)	Draw from current position to position X, Y
MOVE(X, Y)	Move from current position to position X, Y
CENCH(N)	Draw graph plotting character (+ ⊡ X ⋈ Y ⅄ ← ↓ → ↑)
WAY(D, L)	Set orientation D and scale size L for character output.

Disc

DALLOC(C, B, P, S)	B = number of blocks
DERASE (C)	Note: if protected, C must be 'DOPENED' before it can be 'DERASED'.
DOPEN(C, P, S)	
DCLOSE(C)	

Magnetic Tape

MOPEN(C, P)
MCLOSE(C)

Record Handling

(I is an integer variable name)

RSIZE(C, N, T)	N = number of words/records, T = 0 unblocked = 1 blocked
RFIND(C, N)	N = record number required (N≥1)
RNUM(C, I)	I becomes current record number
BUFFER(N)	N = number of buffers required (use only in main program)

FORTRAN COMPILE TIME ERROR MESSAGES

Number	*Meaning*
1	Illegal character or parity failure.
2	Statement label on continuation line.
3	Non-alphanumeric character following <. >.
4	Unidentified operator between two <. >s.
5	Format error in real constant.
6	Integer constant outside range.
7	Decimal exponent exceeding two digits.
8	Floating point overflow in real constant.
9	<. identifier> not terminated <.>.

Number	*Meaning*
10	Identifier of more than six characters.
11	Two main programs.
12	Impermissible statement following logical IF or in BLOCK DATA subprogram.
13	(i) Nested subprograms. (ii) BLOCK DATA preceded by SUBROUTINE or FUNCTION statement.
14	Specification statement following statement function, executable statement or data initialisation statement.
15	(i) No executable statement in (non-BLOCK DATA) subprogram. (ii) No GOTO, RETURN etc, before END.
16	Error in a label.
17	Statement type cannot be identified.
18	Impermissible terminator in DIMENSION statement.
19	Impermissible terminator in EXTERNAL statement.
20	Adjustable subscript in array element name has not appeared in dummy argument list of subprogram.
21	Element other than array in DIMENSION statement.
22	Impermissible terminator in TYPE statement.
23	Impermissible element in EXTERNAL statement.
24	Array subscript not terminated by comma or right bracket.
25	Array with more than three dimensions.
26	Array subscript not integer variable or integer constant.
27	No terminator after identifier or array.
28	No identifier where one expected.
29	Two Class IV items with the same name.
30	Illegal terminator in EQUIVALENCE statement.
31	Inconsistency in EQUIVALENCE statement.
32	Two items in COMMON have been equivalenced.
33	Inconsistency between dimension of array in EQUIVALENCE statement and any other specification statement.
34	Dummy argument in EQUIVALENCE statement.
35	Element other than identifier or array in EQUIVALENCE statement.
36	No comma or bracket where one expected in EQUIVALENCE statement.
37	Block name error in COMMON statement.
38	Dummy argument or element other than identifier or array in COMMON statement.
39	(i) Identifier appearing twice in COMMON blocks. (ii) Identifier appearing twice in same COMMON block.
40	Label defined twice, or DO statement terminated by label already defined.
41	Impermissible statement terminating a DO loop.
42	DO loops nested incorrectly.
43	Input/output statements referring to a non-FORMAT statement.
44	DO loops nested too deeply.
45	Impermissible expression in IF statement.
46	Erroneous use of array in format specification.
47	Non-octal string in PAUSE or STOP statement.
48	Non-existent input/output device specified in input/output statement.

Number	*Meaning*
49	Impermissible redefinition of DO loop variable.
50	No label where one expected.
51	Missing delimiter in GOTO, DO or IF statement.
52	Variables in DO or GOTO statements not integer.
53	Impermissible LHS of arithmetic statement.
54	Statement function occurring after executable statement or data initialisation statement.
55	Missing operand in arithmetic expression.
56	Logical operator (i.e. .AND . . OR. or .NOT.) without logical operands.
57	Badly formed complex constant, or superfluous comma after a left bracket.
58	Operator on level zero on left side of assignment statement; or, left bracket or constant first in statement.
59	Missing operator in arithmetic expression.
60	Missing right bracket in arithmetic expression.
61	Missing left bracket after function or array name in a parameter list.
62	Unexpected comma in arithmetic expression, e.g. Function or array with too many arguments.
63	Function or array with too few arguments.
64	Right bracket or comma found on level zero of assignment statement.
65	Assignment to a variable of the wrong type.
66	Array subscript not of type 'integer'.
67	Exponentiation with an impermissible combination of operands.
68	Arithmetic operator with wrong type of operands.
69	Impermissible combination of operands for the comparison operators.
70	Intrinsic or basic external function with operands of impermissible type.
71	(i) Impermissible subprogram name. (ii) Impermissible symbol after subprogram name. (iii) No right bracket after dummy argument list.
72	Error in CALL statement.
73	Dummy variable appears twice in dummy argument list.
74	Impermissible symbol in argument list.
75	Error in input/output list.
76	Undefined label.
77	Impermissible formal parameter list in statement function definition.
78	Equals sign misplaced within an assignment statement.
79	Impermissible use of basic external function, subroutine or function subprogram name in an arithmetic expression.
80	Array subscript in input/output list not of permissible form.
81	No RETURN statement in subprogram.
82	Badly formed variable list in a DATA statement.
83	Badly formed constant list in a DATA statement.
84	A name occurring in a DATA statement is not an array or simple variable.
85	The number of subscripts of an array element in a DATA statement is wrong.
86	Incorrect symbol in constants list of DATA statement.
87	Constant of wrong type in DATA statement
88	Attempt to extend common block backwards by an EQUIVALENCE statement.

Number	*Meaning*
89	Error in a FETCH statement.
90	Badly structured FORMAT statement.
91	Too few characters in Hollerith string of DATA statement.
92	Not two real numbers in a complex constant in a DATA statement.
93	Illegal separator in a complex constant in a DATA statement.
94	Illegal terminator in a complex constant in a DATA statement.
95	More variables than constants in a DATA statement.
96	More constants than variables in a DATA statement.
97	Superflous character at end of DATA statement.
98	Arithmetic expression too deeply nested.
99	Program too large or difficult to compile.
101	Common block greater than 32K words.
102	DO loop limiting control variable greater than 32K.
103	More than 511 words of numeric and/or Hollerith constants in a sub-program.

The following errors do not affect compilation, being really just warnings.

200	Jump to higher level DO loop.
201	Re-definition of array size.
211	Array declarator appears in both Common and Type statements.
226	A formal parameter used as an array dimension specification is apparently real. It will be considered integer.
281	RETURN statement in main program. Considered as STOP.

FORTRAN FILING ERROR MESSAGES

If any disc or magnetic tape files are misused by the operating system, an error message of the form

FILING ERROR <N> CHANNEL <C>

is displayed and the job or run abandoned. The meaning of the error number N is given below.

Number	*Description*
1	File corrupt.
2	Channel not assigned.
3	File wrong type.
4	File not found.
5	File already present.
6	File protected.
7	File overflow.
8	System error.

FORTRAN LOAD TIME ERROR MESSAGES

Number	*Description*
1	Sumcheck failed on input of relocatable binary paper tape.
2	Not enough room in store for program and data.
3	A named common block used by two separate subprograms does not have the same length in each.
4	(i) A subprogram has been included twice. (ii) The type, or class, or the number of parameters of a subprogram is not compatible with an earlier reference.
5	The load time dictionary has overflowed, i.e. there are more than 20 common blocks, or more than 100 subprograms in the current job.
6	Not enough room to allocate store for a particular common block.

FORTRAN RUN TIME ERROR MESSAGES

Unless otherwise specified, the run is terminated.

Number	*Description*
1	Inadequate FORMAT specification found on trying to output a real number. The number is output in FORMAT E 18.11 (or D32.25 in the case of double precision numbers) in the same record as the offending FORMAT descriptor. The program continues running.
2	Inadequate FORMAT specification found on trying to print an integer in I FORMAT. The integer is output in FORMAT I8 and the program continues.
3	An attempt has been made to print, according to F, E, G or D format, a real element which is not in standard floating point form.
4	The format descriptor is not compatible with the type of element to be output.
5	(i) FORMAT descriptor specifies number of digits after decimal point greater than total number of characters. (ii) Number of impermissible format.

Errors in the syntax of a format descriptor

7	Erroneous character following I, L or A.
8	Errors in D, E, F or G descriptor.
9	Erroneous character following descriptor.
10	Erroneous character following separator.
11	Erroneous character following minus sign.
12	Erroneous character following negative integer.
13	Erroneous character following P.
14	Erroneous character following integer.
15	Erroneous character following).
16	Repeat count is zero.
17	There is not a right-hand bracket at end of format statement.
18	Unacceptable character following FORMAT.
19	Descriptor used with storage location of wrong type.
20	There are no descriptors other than nH and nX but there is an I/O list.

Other run time errors

21	There is no room in store to allocate a dynamic array.
22	An illegal extracode has been obeyed.
23	The control variable of a computed GOTO is out of range.
24	The control variable of an assigned GOTO has not been set up by an ASSIGN statement.
25	A program has had more than 10 floating point overflow conditions.
26	Attempt to run a segmented program without the segmentation module.

FORTRAN BACKING STORE ERROR MESSAGES

The error messages for misuse of disc or magnetic tape at run-time are of the form

FIORBS
FIORD } ERR <N> CHANNEL <C>
FIORM

Number	*Meaning*
1	File corrupt.
2	Attempt to read beyond the end of file indicator.
3	Wrong type of file (e.g. not a disc file)
4	File state wrong or file cannot be opened.
5	Channel not assigned.
6	File absent.
7	Attempt to read or write beyond the end of a V-file or the end-of-tape marker on magnetic tape.
8	Attempt to write to a C-file on disc or SBL8 labelled section on magnetic tape that has been opened for reading.
9	Failure of BACKSPACE statement.
10	Failure of RSIZE subprogram.
11	Failure of RFIND subprogram.
12	Failure of RNUM subprogram.
13	Failure of DALLOC subprogram.
14	Failure of DERASE subprogram.
15	Failure of DCLOSE subprogram.
16	Failure of BUFFER subprogram.
17	Not enough room to allocate a buffer for a disc or magnetic tape file.
18	Hardware failure.
19	Failure of DOPEN subprogram.
20	V-file or magnetic tape protected against writing, reading, or deleting.
21	Random access attempted on a V-file or magnetic tape not containing fixed length records.
22	Wrong magnetic tape loaded.
23	Failure of MOPEN subprogram.
24	Failure of MCLOSE subprogram.
99	System error.

No channel number is given with error numbers 16 and 18 as these do not refer to a specific file.

5 Commercial languages

COBOL and the commercial NEAT assembler, NEATCC are run under the commercial operating system. If errors occur when compiling or running, explicit error messages are given.

The compilation mode adopted by COBOL and NEATCC and the program's options are specified by program (or subroutine) control lines input at each compilation.

The following three pages are reduced reproductions of the standard sheets used for specifying control lines.

ICL

4100 Neat compiler program specification sheet

1 5
[L] 0 [T] 0 [T]

9
Enter program name and version number

17
[T] P R O G R A M [T]

25
If program to be relocatable, enter Y.

26
If program has overlay chapters, enter Y

27
If punched source lines to be sorted, enter Y.

28
If source lines with T–numeral ≤ numeral *n* are not to be compiled, enter numeral *n*.

29
If period to be read as field separator, enter Y.

30
If new page/line numbers to be assigned, enter numeric line increment.

33
If program base address not 256, enter base address as six decimal digits > 256

39
If re-compilation, enter T for source master on magnetic tape.

40
Enter source master name and version number.

[T]

49
If listing to be suppressed, enter R to suppress X – ref, M to suppress main listing or B to suppress both

50
If user subroutine listing is required, enter Y.

51
If system subroutine listing is required, enter Y.

52
If main listing is to be double spaced, enter Y.

53
If object program is to be punched, enter P for paper tape, C for cards.

54
If object program is to be written to magnetic tape, enter T.

55
If object program is to be stacked, enter Y.

56
If a new source master is to be created, enter Y

57
Enter name of file to receive output.

1 5
[L] 0 [T] 1 [T]

9
If macro generation is required, enter Y

10
If source lines with T–numeral are to be dropped, enter Y

11
If a list of the magnetic output file is required, enter Y.

17
[T] N E A T [T]

25
Enter "NEXTDO" program name and version number

33
Enter "NEXTDO" $SCC

37
If "NEXTDO" program is on magnetic tape, enter Y if rewind is required before loading.

[L]

[H]
(Only if separate paper tape)

ICL

4100 Neat compiler subroutine specification sheet

T 5
[T] 0 [T] 0 [T]

9
Enter subroutine identifier and name.

17
[T] S R O U T I N [T]

25
Enter C if chapter subroutine, B if block.

26
Enter overlay group number if chapter, [S] if block.

27
If punched source lines to be sorted, enter Y.

28
If source lines with T numeral ≤ numeral *n* are not to be compiled, enter numeral *n*

29
If period to be read as field separator, enter Y.

30
If new page/line numbers to be assigned, enter numeric line increment.

33
[S] [S] [S] [S] [S] [S]

39
If recompilation, enter T for source master on magnetic tape.

40
Enter source master identifier and name.

[T]

49
If listing to be suppressed, enter R to suppress X – ref, M to suppress main listing or B to suppress both.

50
[S] [S]

52
If main listing is to be double spaced, enter Y.

53
If subroutine object is to be punched, enter P for paper tape, C for cards.

54
If subroutine object is to be written to magnetic tape enter T.

55
If object subroutine to be stacked, enter Y.

56
If a new source master is to be created, enter Y.

57
Enter name of file to receive output.

1 5
[L] 0 [T] 1 [T]

9
If macro generation is required, enter Y.

10
If source lines with T–numeral are to be dropped, enter Y.

11
If a list of the magnetic output file is required, enter Y.

17
[T] N E A T [T]

25
Enter octal number of parameters to be coded in address – phrase of call line.

27
Mnemonic type 1st parameter

28
Mnemonic type 2nd parameter

29
Mnemonic type 3rd parameter

0 if NONE
1 if LD
2 if LD L
3 if LDR
4 if LDR L
5 if LDK
6 if LDK L

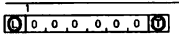
30
Enter 3 – character version number to be printed on listing at call time.

[L]

ICL

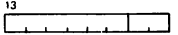
4100 Cobol compiler program specification sheet

1



8

13



Enter program name and version number

21

If source input from source master, enter S
If source input on work tape, enter W
If punched source input, enter P

22

If library text to be included in source program, enter Y

23

If listing to be suppressed, enter Y

24

If object program to be stacked, enter Y

25

If trace facilities to be included, enter Y

26

If load and go compilation, enter Y

27

Object program base address (decimal) - right justified, zero filled

33

Object program file name

45

Enter "NEXT DO" program name and version number

53

Enter "NEXTDO" $SCC

57

If "NEXTDO $SCC" rewind enter Y

6 Systems error messages

EXECUTIVE MESSAGES

Message	*Meaning*
NO DIN	The disc program loader cannot be found in the DES SYSTEMS file on volume 0.
NO MTIN	The magnetic tape program loader cannot be found by searching forward in the file on magnetic tape handler 0.
<program name> ABSENT	The specified program cannot be found in store or in the specified file.
NO CHAP	An attempt has been made to access a chapter not currently in store.
AMOK	A jump has been made to an undefined location between 64 and 255, or an undefined extracode has been obeyed.
X	An incorrect parameter has been given to a program.
TABFUL	The last program input to store filled the NICE table.
FULLUP	An attempt has been made to input a program although the NICE table is already full.
NO ROOM	All available space in main store has been used.
FP OFLO	Floating point overflow.

DISC FILING ERRORS

The following messages are output by DFILE or CFILE

Message	*Meaning*
DF ERROR	Incorrect entry parameter to DFILE.
CF ERROR	Incorrect entry parameter to CFILE.
LINK ERR	A block in a C-file has an incorrect link.

Programs using DFILE or CFILE may output messages of the form DFERR <N> or CFERR <N> where the meaning of N is as follows.

N	*Meaning*
1	The file cannot be opened as the open table is full
2	The channel is already open
3	The file cannot be found
4	There is no room in the VTOC to allocate the file (DFILE) An attempt has been made to delete a C-file being written (CFILE)
5	The file is protected
6	The channel is not assigned

N	*Meaning*
7	The file is not open
8	The file is of the wrong type
9	The filename is already in use
10	The transfer requested was too large (DFILE) The PAD file is full (CFILE)
11	The transfer extends out of range (DFILE) The user identifier cannot be found (CFILE)
12	The sector is outside the file or cannot be found (DFILE)
13	A data check error has occurred (DFILE)

When RECORD is used by a NEAT program or by an ALGOL program using DR12, messages of the form

REC ERR <N> CH <C>

are displayed where C is the channel number of the file and the meaning of N is as follows.

N	*Meaning*
1	The file is corrupt
2	An attempt has been made to read beyond the end of the file or, on magnetic tape, write beyond the end-of-tape without calling RCOND.
3	Wrong type of file used
4	The file cannot be opened
5	The channel is not assigned
6	The V-file is not allocated
7	Attempt to read or write beyond the end of a V-file
10	Misuse of RSIZE
11	Misuse of RFIND
12	Misuse of RNUMBER
15	Misuse of CLOSE
18	Hardware failure
19	Wrong entry parameter
20	Error during output
21	Error during input
22	Misuse of REND
23	Misuse of ENDFILE
24	Misuse of RCOPY
25	Misuse of REMAINDER
26	Misuse of RCOND

GRAPHICAL DISPLAY ERROR MESSAGES

DISMAN errors are of the form DMEROR <X> where X is as follows

X	*Meaning*
A	Corrupt display file or nameword encountered.
B	Not enough room available in display file.
C	Too many anonymous items.
D, E	Parameter error on entry to REMOVE.
F	Corrupt display file or nameword during 'store collapse'.
G	Error in ENDSUB routine.
H	Too much code for size of buffer.
I	Parameter error on entry to MOVE, REMOVE or ALTER.
J	Parameter error on entry to REPLACE.

FRED errors are of the form DEV 12 ERR <A> <B>
EDGAR errors are of the form EDGAR ERROR NO. <A> <B>
In either case, the meaning of A is as follows

A	*Routine*	*Meaning*
1	RESET	Display file too small.
2	INSERT/DRAW	Too many items.
3	INSERT/DRAW	Item not present.
4	DELETE	Item not present.
5	REPL/REPLACE	Item not present.
6	RENAME	Item not present.
7	ENTER	Buffer is full.
8	PLACE/MOVE	Item not present.
9	ALTER	Item not present.
10	DEFSUB	Too many items.
11	SUBPIC	Item not present.
12	SUBPIC	Item not a subpicture.

7 Batch details

CONTROL COMMANDS USED IN T30C AND DES BATCH
N.B. Fields in parentheses are optional.

&ALGOL; (<parameters chosen from CH, L and SC>;)
&ALGSTORE;
* &ASSIGN;<channel number>; DC;<volume number>;<filename>(,<user identifier>);
* &ASSIGN; <channel number>; DV; <volume number>; <filename>;
† &ASSIGN;<channel number>; MT; <handler number>; <filename>; (<label name>;)
* &ASSIGN; <channel number>; <peripheral identifier>;
&BCDCODE;
&EBCDIC;
&END;
** &FILE; <handler number>; <filename>;
&FORTRAN; (L;)
** &FORTRAN; (<program name>;)
&JOB; <job number>; (<job description>;)
** &LIBRARY; <handler number>; <filename>;
&LINES; <maximum number of lines>;
&LIST;
&LOAD; <program name>;
* &LOAD; <program name>; DC; <volume number>; (<language>;)
* &LOAD; <program name >; <user identifier>; DC; <volume number>; (<language>;)
&LOAD; <program name>; MT; <handler number>; <filename>; (<language>;)
** &LOAD; <program name>; <handler number>; <filename>; (<language>;)
* &LOAD; <program name>; PTR1;
&NEAT; <parameters chosen from L, CH, LY, SC, OB and CD>;)
&NORUN;
&OPERATOR; <message>;
&OPTIONS; (<name 1>; . . . . . .<name N>;)
&<program name>; (parameters for program>;)
&RUN; (<program name>; <parameters for program>;)
&TIME; <maximum time>; (<expected time>;)
&UNLIST;
&WAIT; <message>;

* DES BATCH only
** T30C only
† In T30C, only allowed for running FORTRAN programs

CHANNEL NUMBERS DURING COMPILATION (DES BATCH)

Channel Number	*Use*	*Default*
1	Compiler source input	Control stream
2	Edit input	No edit
3	Source code library input (ALGOL and NEAT)	DES ALGOL LIB, volume 0
4	Source code output	None
5	Object code output	WORKFILE 1, volume 0 (store under NEATERD)

At load time, channel 3 is used for FORTRAN subprogram input and defaults to the PAD and DES SYSTEMS files on volume 0. Channel 6 is used at load and run time for the segment file and defaults to the DES WORKSPACE file on volume 0.

Run time channel numbers for standard peripherals are given on page 43.

STATE OF JOB ON COMPLETION

At the end of the job, after the elapsed time has been output, a single character is output to the teleprinter and the lineprinter to show the state of the job on completion. The meaning of the character is as follows:-

Character	*Meaning*
A	The job ran to completion.
B	The run was terminated by a &RUN, &END or &JOB command being read in the data.
C	A control command was encountered in the wrong position, or it is not possible to perform the job as specified by the control commands.
E	The operator abandoned the job or a non-continuable software error condition occurred.
F	A program failed to compile.
G	A program did not run correctly.
H	The programmer's specified time expired.
I	The programmer's specified number of lines was exceeded.
J	An incorrect control command was encountered.
K	A program could not be found in the specified file.
L	A disc filing error occurred.
S	A standard program was not available e.g. ALGOL or NEATERD was absent
T	The job required the FORTRAN system, which was not available.
U	The expected time exceeded the installation maximum time.
V	The installation default time expired.
W	The installation maximum time expired.
X	The installation maximum number of lines was exceeded.
Y	An incorrect magnetic tape or disc was loaded.
Z	A hardware failure occurred (e.g. read failure on magnetic tape).

8 Fixed locations

The first 256 locations of store (or of the slave area when using DES2) are reserved by the operating systems to provide an interface between the user and the standard software. H indicates that the location is used by hardware. Locations not listed are reserved for systems use.

Fixed locations used by scientific operating systems

Octal Address	*Decimal Address*	*Use*
00	0	H Subroutine link set by JFL and JIL
01	1	H Extracode operand address (operand if literal)
02	2	H Extracode link
03	3	H Attention link
04	4	H Interrupt link
05	5	H £CUCHCWP - address of current chapter's codeword
06	6	H £CACHCWP } link set by JILX
07	7	H £CACHLK }
10-12	8-10	Floating point accumulator on a 4120
13-77	11-63	Short workspace used by NEAT, IN, DIN etc.
100-177	64-127	Extracode entries
207	135	£PIN - paper tape reader device routine
210	136	£POUT - paper tape punch device routine
212	138	£EXTYPE - control teleprinter device routine (also £C2TYPE)
213	139	£MTNAME } magnetic tape device routines
214	140	£MTOPEN
215	141	£MTCLOSE
216	142	£MTWAIT
217	143	£MTREAD
220	144	£MTWRITE
221	145	£MTMARK
222	146	£MTSTAT
223	147	£MTCON
224	148	£PTULA - shows which devices are using ATU
226	150	£DISPLAY - graphical display routine
227	151	£PTUBOOK - shows booking of PTUs
230-235	152-157	H Packed Transfer unit workspace
241	161	£CARDIN - card reader device routine
242	162	£LINEOUT - lineprinter device routine
243	163	£PLOT - digital plotter device routine
244	164	£CHOC - card punch device routine
245	165	£STORESIZE
246	166	£DFILE
247	167	£ASSIGN - assign table address
251	169	£CFILE
261	177	£CLOCK

Octal Address	*Decimal Address*		*Use*
263	179		ITABLE address
265	181		£LOAD
272	186		ATABLE address
273	187		£MSTADD - address of program name table
274	188		£ENTER
275	189		£SPECSYS
277	191		£DSEEK
300	192		£DREAD
301	193		£DWRITE
302	194		£DCOND
303	195	H	Attention entry
304	196	H	Interrupt entry
305	197	H	Floating point overflow entry
312	202		£LOWADD
313	203		£TOPADD
316	206		£OUTNAME
317	207		£ASSEMBLE
320	208		£PARAM
323	211		£UPDATE
324	212		£LOOKSUP
325	213		£END
330	216		£MESKEY
336	222		£NOROOM
337	223	H	INDEX check failure
342-377	226-255	H	Unpacked transfer unit workspace

disc device routines

Fixed locations used by Commercial Software

Octal Address	*Decimal Address*	*Use*
00	0	H Link set by JFL and JIL
01	1	H Extracode operand
02	2	H Extracode link
03	3	H Attention link
04	4	H Interrupt link
05	5	H £CUCHCWP - current chapter codeword address
06	6	H £CACHCWP } link signpost set by JILX
07	7	H £CACHLK }
10-12	8-10	Floating point accumulator on a 4120
13-17	11-15	Extracode routines temporary workspace
22-25	18-21	Attention sections }
26-31	22-25	Interrupt sections } Device routines temporary workspace
32-37	26-31	Direct sections }
40-57	32-47	Users temporary workspace
60-77	48-63	Library subroutines temporary workspace
100-177	64-127	Extracode entries
200-201	128-129	£DATEA - actual date in form DD/MM/YY
202-203	130-131	£DATEV - virtual date in form DD/MM/YY
204	132	£YEAR - year in binary
205	133	£DIYR - day-in-year in binary
206	134	Dates hash total
207	135	£ASK - ask routine
210	136	SYS£ASKT - ask table address
211	137	£FIN - finish routine
212	138	SYS£TYPE - type routine
213	139	SYS£PTIN - punched tape read routine
214	140	SYS£PTOT - tape punch routine
215	141	SYS£PRNT - line printer routine
216	142	SYS£PCIN - card reader routine
217	143	SYS£PCOT - card punch routine
225	149	£CRIT - critical interrupt flag
227	151	£RDMPF - rescue dump flag
230-235	152-157	H Packed transfer unit workspace
240	160	£ABAN - abandon routine
241	161	SYS£OLTH - reserved for monitor
242	162	Bootstrap executive entry
243-244	163-164	Executive attention routines workspace
245	165	£HILIM - last free location in store
246-247	166-167	£NXDO - nextdo program name
250	168	$SCC - nextdo device
251	169	SYS£GRAF - graphical display routines
252	170	SYS£APRT - address of program read table
253	171	SYS£DING - dating routine
254	172	SYS£BOOT - bootflag
255-256	173-174	SYS£ENTR - entry signpost to program
261	177	£LOLIM - first free location in store
262-263	178-179	SYS£PROG - current program name
264	180	SYS£XNXT - executive communication (SYSMON)
266	182	SYS£CBCC - Current MT cyclic block count

Octal Address	*Decimal Address*		*Use*
273	187		SYS£BASE - current main chapter base
275	189		SYS£OLAY - overlay program flag
277-302	191-194		Executive communication in SYSMON
303	195	H	Attention entry
304	196	H	Interrupt entry
305	197	H	Floating point overflow entry
337	223	H	INDEX check failure
342-377	226-255	H	Unpacked transfer unit workspace

9 Peripherals

CHANNEL NUMBERS

For configurations without a PTU the standard peripheral channel numbers and the ALGOL and FORTRAN device numbers used to refer to them are as follows. The device numbers in parentheses are the numbers which default to the device in DES BATCH.

Device	*Peripheral Socket*	*ALGOL device number*	*FORTRAN device number*
Control teleprinter/typewriter	1	3 (33)	1 (31)
Paper tape reader 1	2	1 (31)	3 (33)
Paper tape punch 1	3	1 (31)	5 (35)
Paper tape reader 2	4	2 (32)	4 (34)
Paper tape punch 2	5	2 (32)	6 (36)
Digital plotter	6	5	9
Lineprinter	7	4 (34)	2 (32)
Card reader	8	6 (36)	7 (37)
Card punch	9	6	–
Multiple-teleprinter controller	10	–	–
Second Line printer	11	–	–
Magnetic Tape	12	–	10 (40)
Real Time Clock	14	–	–
Graphical display	19	9	12 (32)

When a PTU is fitted, the following peripheral channels may also be used.

Device	*Peripheral Socket*
Magnetic tape	16 or 20 if a graphical display is on channel 19
Disc	16 or 20 if a graphical display is on channel 19 or magnetic tapes are on channel 16 or 24 if a graphical display is on channel 19 and magnetic tapes are on channel 20

Disc pack — Programmers' Statistics

Handlers per Controller	Up to 8	Tracks per cylinder	10
Maximum transfer rate	208 kc/s	Cylinders per pack	100
Average transfer rate	164 kc/s	Revolution time	25ms 2400rpm
Word capacity	1,024,000	Head movement times: 1 cylinder	30ms
Character capacity	4,096,000	20 cylinders	90ms
Words per sector	64	50 cylinders	125ms
Sectors per track	16	100 cylinders	150ms

Magnetic Tape-Programmers' statistics

Tape length (useful)	2,400 ft	Inter-block gap time:	12.5ms
Tape width	0.5 ins		
Tape speed	60 in./sec	Maximum block-length without P.T.U.	2,047 characters (511 complete words)
Rewind time	2 min. (max)		
Inter-block gap length	0.75 in.	Maximum block length with P.T.U.	32,767 characters (8,191 complete words)

	12 kc/s magnetic tapes			33 kc/s magnetic tapes		
Transfer rate	12,000 characters/sec. 3,000 words/sec.			33,360 characters/sec. 8,340 words/sec.		
Packing density	200 characters/in. 50 words/in.			556 characters/in. 139 words/in.		
Time to transfer 2047 characters	171 ms			61 ms		
Time to transfer 32,767 characters	2,732 ms			984 ms		
Average block size	64 words	511 words	8191 words	64 words	511 words	8191 words
Length of block	1.3 in.	10.2 in.	164 in.	0.5 in.	3.7 in.	59.2 in.
Time for one block and gap (at full speed)	34 ms	183 ms	2750 ms	20 ms	74 ms	1010 ms
Blocks per second (max.)	29	5½	3/8	49	13½	1
Effective transfer rate (words/sec)	1,900	2,800	2,978	3,200	6,900	8,190
No. of blocks per reel	14,200	2,620	175	23,800	6,500	482
No. of words per reel (in thousands)	909	1,340	1,439	1,520	3,330	3,950
No. of characters per reel (in thousands)	3,640	5,360	5,756	6,080	13,320	15,800

SUMMARY OF STATUS AND CONTROL WORD BITS

Device	*Control*	*Status*
Paper Tape Reader	1. Inhibit Interrupts 2. Permit Interrupts	1. Busy 2. Manual 3. Unloaded 4. Creep 5. Interrupts inhibited
Paper Tape Punch	1. Inhibit Interrupts 2. Permit Interrupts	1. Busy 2. Manual 3. Paper tape low 5. Interrupts inhibited
Control Teleprinter or Typewriter	1. Inhibit Interrupts 2. Permit Interrupts 3. Set input mode 4. Set output mode	1. Busy 2. Manual 3. Message key depressed 4. Error 5. Interrupts inhibited 6. Input mode
Line Printer	1. Inhibit Interrupts and Attentions 2. Permit Interrupts and Attentions 3. Set manual 4-8. Paper movement control for unbuffered printer	1. Busy 2. Manual 3. Interrupts and Attentions inhibited 4. Mistransfer for unbuffered printers 5. Paper low 6. Error
Magnetic Tape	*Control (A)* 1. Inhibit Interrupts and Attentions 2. Permit Interrupts and Attentions 3. Set handler in manual 4-6. Handler number 7, 8 binary value 0 no operation 1 rewind 2 backspace 3 erase *Control (B)* 1. Set even parity mode 2. Set odd parity mode 3. Clear check bits	*Status (A)* 1. Controller busy 2. Handler in manual 3. Interrupts and Attentions inhibited 4-6. Handler number 7. Parity error 8. Handler in Attention state *Status (B)* 1. Handler busy 2. Write Permit on 3. Short record input 4. Long record input 5. Under 4 characters transferred 6. Beginning-of-tape marker 7. End-of-tape marker 8. Handler in even parity mode.

Device	*Control*	*Status*
Multi Access Teleprinter Controller	1-3. Select this channel number 4. Set input mode on channel 5. Set output mode on channel 7. Inhibit controller Interrupts 8. Permit controller Interrupts	1-3. Currently selected channel number 4. Channel ready for input 5. Channel ready for output 6. Error on channel 7. Message on channel 8. Controller Interrupts inhibited
Digital Plotter	1. Inhibit Interrupts 2. Permit Interrupts	No status word available
Discpack	*Control (A)* 1, 2, 3: 0. No operation; 1. Prime for reading; 2. Prime for writing 3. Prime for checking 4. Seek forwards 5. Seek backwards 6. Return to cylinder 0 7. Set status 4-6. Handler Number 7. Inhibit Interrupts 8. Permit Interrupts *Control (B)* For read, write or check 1-8. Sector Number For seek 1-8. Number of cylinders movement	*Status (A)* 1, 2, 3: 0. No handler selected; 1. Read/write busy; 2. Read/write/check successful 3. Read/write/check failed 4. Seek in progress 5. Seek successful 6. Seek unsuccessful 7. Handler free 4-6. Handler Number 7. Interrupts inhibited 8. Special mode *Status (B)* 1-8. Sector number *Status (C)* 1-8. Cylinder number *Status (D)* 1. Handler not available 2. Handler error 3. Data error 4. Cyclic redundancy check error 5. Check error 6. Address error 7. Operation out of range 8. Writing not permitted

Device	*Control*	*Status*
Card Reader	1. Inhibit Interrupts and Attentions 2. Permit Interrupts and Attentions 3. Set manual 7. Feed a card	1. Busy 2. Manual 3. Interrupts and Attentions inhibited 4. Missed transfer 5. Hopper empty 6. Error 7. Card in flight
Card Punch	1. Inhibit Interrupts and Attentions 2. Permit Interrupts and Attentions 3. Set manual 4. Skip card	1. Busy 2. Manual 3. Interrupts and Attentions inhibited 5. Read check error
Graphical Display	1. Inhibit Interrupts and Attentions 2. Permit Interrupts and Attentions 3. Hold at end of word 4. Hold at end of frame 5–6. Lightpen control 7. Continue after hold	1. Display held 2. Lightpen enable switch depressed

10 Codes

4100 Internal Code

Decimal	*Octal*	*Standard (in-shift)*	*Extended (out-shift)*	*Decimal*	*Octal*	*Standard (in-shift)*	*Extended (out-shift)*
0	00	Space	Null	32	40	@	' (grave)
1	01	Horizontal tabulate		33	41	A	a
2	02	Linefeed		34	42	B	b
3	03	½		35	43	C	c
4	04	$		36	44	D	d
5	05	%		37	45	E	e
6	06	&		38	46	F	f
7	07	' (acute)		39	47	G	g
8	10	(		40	50	H	h
9	11	)		41	51	I	i
10	12	*		42	52	J	j
11	13	+	Vertical tabulate	43	53	K	k
12	14	, (comma)	Formfeed	44	54	L	l
13	15	–	Carriage return	45	55	M	m
14	16	.		46	56	N	n
15	17	/		47	57	O	o
16	20	0		48	60	P	p
17	21	1		49	61	Q	q
18	22	2		50	62	R	r
19	23	3		51	63	S	s
20	24	4	Stop	52	64	T	t
21	25	5		53	65	U	u
22	26	6		54	66	V	v
23	27	7		55	67	W	w
24	30	8		56	70	X	x
25	31	9		57	71	Y	y
26	32	:		58	72	Z	z
27	33	;		59	73	[	"
28	34	<		60	74	£	↑
29	35	=		61	75	]	←
30	36	>	|	62	76	Shift out	Shift out
31	37	10	}	63	77	Shift in	Shift in

Lineprinter and Card Codes

Decimal	Octal	Lineprinter Code A	Lineprinter Code B	Card Punching	Decimal	Octal	Lineprinter Code A	Lineprinter Code B	Card Punching
0	00	Space	Space	Null	32	40	@	@	8-4
1	01	Tabulate	Tabulate		33	41	A	A	12-1
2	02		"	8-7	34	42	B	B	12-2
3	03	½	½	0-8-2	35	43	C	C	12-3
4	04	$	$	11-8-3	36	44	D	D	12-4
5	05	%	%	0-8-4	37	45	E	E	12-5
6	06	&	&	12	38	46	F	F	12-6
7	07	' (apostrophe)	' (acute)	8-5	39	47	G	G	12-7
8	10	(	(	12-8-5	40	50	H	H	12-8
9	11	)	)	11-8-5	41	51	I	I	12-9
10	12	*	*	11-8-4	42	52	J	J	11-1
11	13	+	+	12-8-6	43	53	K	K	11-2
12	14	, (comma)	, (comma)	0-8-3	44	54	L	L	11-3
13	15	-	-	11	45	55	M	M	11-4
14	16	.	.	12-8-3	46	56	N	N	11-5
15	17	/	/	0-1	47	57	O	O	11-6
16	20	0	0	0	48	60	P	P	11-7
17	21	1	1	1	49	61	Q	Q	11-8
18	22	2	2	2	50	62	R	R	11-9
19	23	3	3	3	51	63	S	S	0-2
20	24	4	4	4	52	64	T	T	0-3
21	25	5	5	5	53	65	U	U	0-4
22	26	6	6	6	54	66	V	V	0-5
23	27	7	7	7	55	67	W	W	0-6
24	30	8	8	8	56	70	X	X	0-7
25	31	9	9	9	57	71	Y	Y	0-8
26	32	:	:	8-2	58	72	Z	Z	0-9
27	33	;	;	11-8-6	59	73		[	12-8-2
28	34	<	<	12-8-4	60	74	£	£	8-3
29	35	=	=	8-6	61	75		]	11-8-2
30	36	>	>	0-8-6	62	76		↑	11-8-7
31	37		←	0-8-7	63	77		` (grave)	12-8-7

Paper Tape Codes

Binary	Dec.	Octal	Flexowriter Model T	Teletype Model 33
00000.000	0	000	Runout	Runout
10000.001	129	201		
10000.010	130	202		
00000.011	3	003		
10000.100	132	204		
00000.101	5	005		
00000.110	6	006		
10000.111	135	207		Bell
10001.000	136	210		
00001.001	9	011	Horizontal tabulate	
00001.010	10	012	Newline	Linefeed
10001.011	139	213	(Vertical tabulate)	
00001.100	12	014		
10001.101	141	215		Carriage return
10001.110	142	216		
00001.111	15	017		
10010.000	144	220		
00010.001	17	021		
00010.010	18	022		
10010.011	147	223		
00010.100	20	024	Stop	
10010.101	149	225		
10010.110	150	226		
00010.111	23	027		
00011.000	24	030		
10011.001	153	231		
10011.010	154	232		
00011.011	27	033		
10011.100	156	234		
00011.101	29	035		
00011.110	30	036		
10011.111	159	237		
10100.000	160	240	Space	Space

Binary	Dec.	Octal	Flexowriter Model T	Teletype Model 33
11000.110	198	306	F	F
01000.111	71	107	G	G
01001.000	72	110	H	H
11001.001	201	311	I	I
11001.010	202	312	J	J
01001.011	75	113	K	K
11001.100	204	314	L	L
01001.101	77	115	M	M
01001.110	78	116	N	N
11001.111	207	317	O	O
01010.000	80	120	P	P
11010.001	209	321	Q	Q
11010.010	210	322	R	R
01010.011	83	123	S	S
11010.100	212	324	T	T
01010.101	85	125	U	U
01010.110	86	126	V	V
11010.111	215	327	W	W
11011.000	216	330	X	X
01011.001	89	131	Y	Y
01011.010	90	132	Z	Z
11011.011	219	333	[	[
01011.100	92	134	£	£
11011.101	221	335	]	]
11011.110	222	336	↑	↑
01011.111	95	137		←
01100.000	96	140	@	
11100.001	225	341	a	
11100.010	226	342	b	
01100.011	99	143	c	
11100.100	228	344	d	
01100.101	101	145	e	
01100.110	102	146	f	

00100.001	33	041	‼	‼	11100.111	231	347	g	
00100.010	34	042	"	"	11101.000	232	350	h	
10100.011	163	243	½	½	01101.001	105	151	i	
00100.100	36	044	$	$	01101.010	106	152	j	
10100.101	165	245	%	%	11101.011	235	353	k	
10100.110	166	246	&	&	01101.100	108	154	l	
00100.111	39	047	'(acute)	'(acute)	11101.101	237	355	m	
00101.000	40	050	(	(	11101.110	238	356	n	
10101.001	169	251	)	)	01101.111	111	157	o	
10101.010	170	252	*	*	11110.000	240	360	p	
00101.011	43	053	+	+	01110.001	113	161	q	
10101.100	172	254	, (comma)	, (comma)	01110.010	114	162	r	
00101.101	45	055	-	-	11110.011	243	363	s	
00101.110	46	056	.	.	01110.100	116	164	t	
10101.111	175	257	/	/	11110.101	245	365	u	
00110.000	48	060	0	0	11110.110	246	366	v	
10110.001	177	261	1	1	01110.111	119	167	w	
10110.010	178	262	2	2	01111.000	120	170	x	
00110.011	51	063	3	3	11111.001	249	371	y	
10110.100	180	264	4	4	11111.010	250	372	z	
00110.101	53	065	5	5	01111.011	123	173		
00110.110	54	066	6	6	11111.100	252	374		
10110.111	183	267	7	7	01111.101	125	175		
10111.000	184	270	8	8	01111.110	126	176		
00111.001	57	071	9	9	11111.111	255	377	Delete	Delete
00111.010	58	072	:	:					
10111.011	187	273	;	;					
00111.100	60	074	<	<					
10111.101	189	275	=	=					
10111.110	190	276	>	>					
00111.111	63	077	¿	¿					
11000.000	192	300	`(grave)	`(grave)					
01000.001	65	101	A	A					
01000.010	66	102	B	B					
11000.011	195	303	C	C					
01000.100	68	104	D	D					
11000.101	197	305	E	E					

11 Miscellaneous

Powers of 2 in decimal

2^n	n	2^{-n}
2	1	.5
4	2	.25
8	3	.125
16	4	.062 5
32	5	.031 25
64	6	.015 625
128	7	.007 812 5
256	8	.003 906 25
512	9	.001 953 125
1 024	10	.000 976 562 5
2 048	11	.000 488 281 25
4 096	12	.000 244 140 625
8 192	13	.000 122 070 312 5
16 384	14	.000 061 035 156 25
32 768	15	.000 030 517 578 125
65 536	16	.000 015 258 789 062 5
131 072	17	.000 007 629 394 531 25
262 144	18	.000 003 814 697 265 625
524 288	19	.000 001 907 348 632 812 5
1 048 576	20	.000 000 953 674 316 406 25
2 097 152	21	.000 000 476 837 158 203 125
4 194 304	22	.000 000 238 418 579 101 562 5
8 388 608	23	.000 000 119 209 289 550 781 25
16 777 216	24	.000 000 059 604 644 775 390 625
33 554 432	25	.000 000 029 802 322 387 695 313
67 108 864	26	.000 000 014 901 161 193 847 656
134 217 728	27	.000 000 007 450 580 596 923 828
268 435 456	28	.000 000 003 725 290 298 461 914
536 870 912	29	.000 000 001 862 645 149 230 957
1 073 741 824	30	.000 000 000 931 322 574 615 479
2 147 483 648	31	.000 000 000 465 661 287 307 739
4 294 967 296	32	.000 000 000 232 830 643 653 870
8 589 934 592	33	.000 000 000 116 415 321 826 935
17 179 869 184	34	.000 000 000 058 207 660 913 467
34 359 738 368	35	.000 000 000 029 103 830 456 734
68 719 476 736	36	.000 000 000 014 551 915 228 367
137 438 953 472	37	.000 000 000 007 275 957 614 183
274 877 906 944	38	.000 000 000 003 637 978 807 092
549 755 813 888	39	.000 000 000 001 818 989 403 546
1 099 511 627 776	40	.000 000 000 000 909 494 701 773

Useful Constants

π =	3.141 592 653 590	$1/\pi$ =	0.318 309 886 184
$\log_{10} e$ =	0.434 294 481 903	$\log_e 10$ =	2.302 585 092 994
$\log_{10} 2$ =	0.301 029 995 664	e =	2.718 281 828 459
$\sqrt{2}$ =	1.414 213 562 373	$\sqrt{3}$ =	1.732 050 807 569
1 radian =	57.295 779 513 082°	1° =	0.017 453 292 520 radian

Second Edition October 1970

3505(10.70)

International Computers Limited
ICL House Putney SW15
Telephone 01-788 7272

Printed in Great Britain by
ICL Printing Services Letchworth