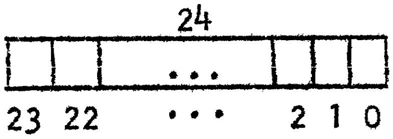
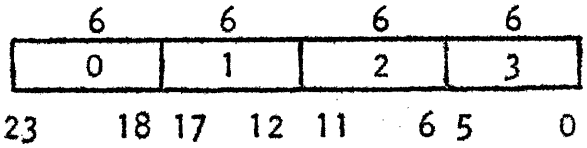
CDC 3300

The CDC 3300 is a fast, medium-scale digital computer. It is basically a binary machine. There is an optional Business Data Processing unit (BDP) which enables the machine to do binary-coded decimal arithmetic and various search, move, and edit operations. We shall not describe this unit here. We shall describe the basic instruction set, together with the double precision and floating point instructions which are provided by another optional unit. We shall also discuss the Comprehensive Assembler (COMPASS).

Storage. In the CDC 3300, a word is a storage element composed of 24 bits (8 octal digits).

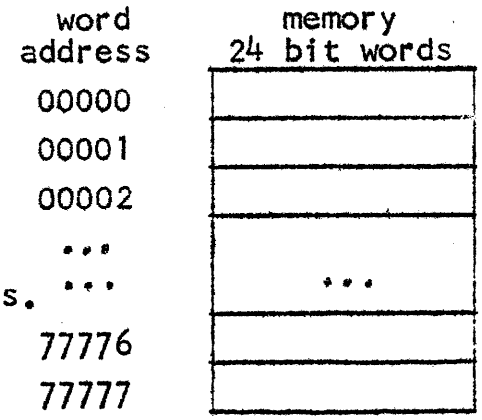


A character is a six bit storage element (2 octal digits). There are 4 characters per word, numbered from left to right.



Most of the instructions that refer to memory, refer to words or to specific portions of words. A few instructions refer to individual characters.

The memory of the CDC 3300 consists of magnetic core storage with a 1.25 microsecond cycle time. There are 2^{15}= 32,768 words of 24 bits each, or 2^{17}= 131,072 characters of 6 bits each. The memory can be expanded to as much as 2^{18}= 262,144 words, or 2^{20}characters.



A word address is a 15 bit number (5 octal digits) which denotes the location of a word in memory. Word addresses range from 00000$_8$ to 77777$_8$. A character address is a 17 bit number which denotes the location of a character in memory. The leftmost 15 bits determine the word and the rightmost 2 bits select the character within the word (numbered from left to right, as shown above).

Registers. A register is a storage element which has special purposes. The CDC 3300 has the following registers:

Name	No. of bits	Purposes
A	24	Accumulator. Arithmetic, logical operations, shifting, etc.
Q	24	Auxiliary accumulator. Extension of A in some operations.
E	48	Double length accumulator. Double precision and floating point operations.
B1	15	Index register. For counting loops, modifying addresses, etc.
B2	15	Index register.
B3	15	Index register.
P	15	Program counter. Holds memory address from which current instruction was fetched.
F	24	Function register. Holds instruction being performed.

Register file. This is a set of 64 words with faster access than the main memory. Some of them have special uses such as control of input-output operations, search, move, real-time, clock, etc. Others, particularly those with addresses 40_8 to 77_8, are available for temporary storage.

address	24 bits
00	
01	
...	
76	
77	

Arithmetic. All binary arithmetic in the CDC 3300 is done in one's complement form. (This even includes address modification and the incrementing of P.) In one's complement notation, the left-most bit determines the sign of a number. If the left-most bit is 0, the number is positive; if 1, negative. To change the sign of a number, all bits are changed (complemented). All zeroes (0000) denotes +0; all ones (1111) denotes -0. A number can be lengthened by extending its sign bit. For example, +5 = 0101 = 00000101 ; -5 = 1010 = 11111010 . Addition is performed by simply adding two binary numbers, and if a carry occurs from the left, adding one to the right-most bit (end-around carry).

In the CDC 3300, if the result of an arithmetic operation is -0, the answer is changed to +0 . See example at right.

	111110	(-1)
	+ 000001	(+1)
	111111	(-0)
changed to:	000000	(+0)

There are three modes of binary aritmetic in the CDC 3300. The numbers in parentheses below indicate the number of bits in the operands and results.

Single precision fixed point (integer) arithmetic.

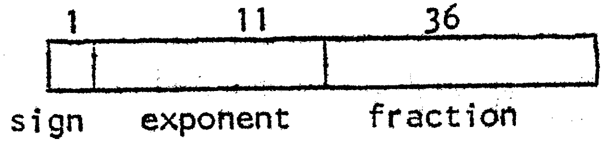
addition	$(24) + (24) \rightarrow (24)$
subtraction	$(24) - (24) \rightarrow (24)$
multiplication	$(24) * (24) \rightarrow (48)$
division	$(48) / (24) \rightarrow (24)$ quotient, (24) remainder

Double precision fixed point (integer) arithmetic.

addition	$(48) + (48) \rightarrow (48)$
subtraction	$(48) - (48) \rightarrow (48)$
multiplication	$(48) * (48) \rightarrow (96)$
division	$(96) / (48) \rightarrow (48)$ quotient, (48) remainder

Floating point arithmetic.

Addition, subtraction, multiplication, and division operate on 48 bit floating point numbers and yield 48 bit floating point numbers as results. The format for floating point numbers is shown below.



Instruction formats and sequencing. Most CDC 3300 instructions occupy one word, but some occupy 2 or 3 consecutive words. We shall be concerned principally with the one-word instructions. These instructions are usually of the "one-address" format. That is, an instruction contains one address which specifies the location of an operand in memory, together with an operation code to specify which operation is to be done. For example, the ADA (Add to A) instruction specifies the location in memory of a number to be added to the number in the A register (the sum replaces the original number in A).

The CDC 3300 follows the usual instruction sequencing scheme for one-address computers, in which instructions are taken from successive locations in memory until a jump occurs. This scheme works as follows: The control unit transmits the contents of P to memory control, which fetches the contents of the word at the address specified by P. The quantity obtained from memory is placed in F and the control unit interprets it as an instruction and performs the operation indicated. Then, if the instruction did not cause a jump (by placing a new address in P), the content of P is incremented by 1. The control unit fetches another instruction from the new address specified by P and performs it. This process continues at a high rate of speed (about 500,000 instructions per second) until the computer is stopped by a halt instruction or until the operator presses the "STOP" button on the control console.

In all instructions, the left-most 6 bits specify the basic operation code (function). The meaning of the other bits depends on the operation code. There are several different instruction formats. We shall list most of the CDC 3300 instructions, in groups according to various formats. (We do not include the BDP instructions.)

Memory reference group 1.

6	1	2	15
f	a	b	m
23 18	17	16 15	14 0

f (bits 23-18) is the function code.

a (bit 17) specifies addressing mode: 0 = direct, 1 = indirect.

b (bits 16-15) selects a B register: 0 = none, 1 = B1, 2 = B2, 3 = B3.

m (bits 14-0) is the basic word address.

All instructions in this group except LDI and STI interpret the a, b, and m fields as follows: After the instruction has been fetched to the F register, an effective address M is computed by adding the selected B register to m: $M = m + (B^b)$. (If b = 0, M = m.) Then a is examined. If a = 1, the contents of memory word M are fetched and bits 17-0 of this word are placed in bits 17-0 of F (the function code remains unchanged). Then this whole process is repeated. If a = 0, M is the effective address of the instruction. The operation may use the contents of M (denoted by (M)) as an operand, or may store information into M. Some instructions use two words in memory (M and M + 1).

The instructions LDI and STI use the b field of the original instruction to determine which B register is to be loaded or stored. The a field specifies indirect addressing as above. LDI with a b field equal to 0 is a "no operation"; STI with b = 0 stores 0 in bits 14-0 of word M.

The COMPASS language format for instructions in this group is:

```
OPC,I      M,B
```

OPC is the mnemonic operation code (such as ADA). If indirect addressing is desired ($a=1$), the op code is followed by a comma and the letter I. Omitting this modifier specifies direct addressing ($a=0$). M denotes the basic memory address, which may be an expression, usually involving symbolic addresses. If a B register is to be specified, the address field is followed by a comma and an expression whose value is 1, 2, or 3. Usually we simply write a 1, 2, or 3, but one can use a symbol whose value has been defined as 1, 2, or 3. The value of B is placed in the b field of the instruction. If the B field is omitted, 0 will be placed in the b field.

The instructions which use this format are listed below. f is given as 2 octal digits.

f	mnemonic op code	Name	Operation
30	ADA	Add to A	$(A) + (M) \rightarrow (A)$
32	ADAQ	Add to AQ	$(AQ) + (M,M+1) \rightarrow (AQ)$
52	CPR	Compare	If $(M) > (A)$, take next instruction in sequence. If $(Q) > (M)$, skip one instruction. If $(A) \geq (M) \geq (Q)$, skip two instructions.
51	DVA	Divide A	$(AQ) / (M) \rightarrow (A)$, remainder $\rightarrow (Q)$
57	DVAQ	Divide AQ	$(AQE) / (M,M+1) \rightarrow (AQ)$, remainder $\rightarrow (E)$
60	FAD	Floating add	$(AQ) + (M,M+1) \rightarrow (AQ)$ (Floating point)
63	FDV	Floating divide	$(AQ)/(M,M+1) \rightarrow (AQ)$ (Floating point)
62	FMU	Floating multiply	$(AQ)*(M,M+1) \rightarrow (AQ)$ (Floa ing point)
61	FSB	Floating subtract	$(AQ)-(M,M+1) \rightarrow (AQ)$ (Floating point)
24	LCA	Load complement A	$(\overline{M}) \rightarrow (A)$ (one's complement of M)
26	LCAQ	Load complement AQ	$(\overline{M},\overline{M+1}) \rightarrow (AQ)$
20	LDA	Load A	$(M) \rightarrow (A)$
25	LDAQ	Load AQ	$(M,M+1) \rightarrow (AQ)$
54	LDI	Load index	$(M_{14-0}) \longrightarrow (B^b)$
27	LDL	Load logical	$(Q) \wedge (M) \rightarrow (A)$
21	LDQ	Load Q	$(M) \rightarrow (Q)$
37	LPA	Logical product to A	$(A) \wedge (M) \rightarrow (A)$ ("and" operation)
50	MUA	Multiply A	$(A)*(M) \rightarrow (QA)$ (yes, it's QA !)
56	MUAQ	Multiply AQ	$(AQ)*(M,M+1) \rightarrow (AQE)$
34	RAD	Replace add	$(M)+(A) \rightarrow (M)$ (A) unchanged

31	SBA	Subtract from A	$(A)-(M)\longrightarrow(A)$
33	SBAQ	Subtract from AQ	$(AQ)-(M,M+1)\longrightarrow(AQ)$
36	SCA	Selectively complement A	$(A)\veebar(M)\longrightarrow(A)$ (exclusive "or" operation)
46	SCHA	Store character address	$(A_{16-0})\longrightarrow(M_{16-0})$
35	SSA	Selectively set A	$(A)\vee(M)\longrightarrow(A)$ (inclusive "or" operation)
40	STA	Store A	$(A)\longrightarrow(M)$
45	STAQ	Store AQ	$(AQ)\longrightarrow(M,M+1)$
47	STI	Store index	$(B^b)\longrightarrow(M_{14-0})$
41	STQ	Store Q	$(Q)\longrightarrow(M)$
44	SWA	Store word address	$(A_{14-0})\longrightarrow(M_{14-0})$
01	UJP	Unconditional jump	$M\longrightarrow(P)$ (Take next instruction from location M.)

<u>Memory reference group 2.</u>

6	3	15
f	g	m
23 18	17 15	14 0

f (bits 23-18) is function code.

g (bits 17-15) is sub-operation.

m (bits 14-0) is word address.

There is no B register selection and no indirect addressing for instructions in this group. Fields f and g together specify the operation (denoted in the list below by 3 octal digits with a point after the first 2). Field m supplies a memory address.

The COMPASS format is :

```
                OPC,MOD    M
```

OPC is the operation code, which may have a required modifier (as shown below) to specify a jump condition. M is the address field, which may be an expression involving symbolic addresses.

f,g	mnemonic op code	Name	Operation
03.4	AQJ,EQ	AQ jump, equal	If $(A)=(Q)$, jump to m.
03.6	AQJ,GE	" , greater or equal	If $(A)\geq(Q)$, jump to m.
03.7	AQJ,LT	" , less than	If $(A)<(Q)$, jump to m.
03.5	AQJ,NE	" , not equal	If $(A)\neq(Q)$, jump to m.
03.0	AZJ,EQ	A zero jump, equal	If $(A)=0$, jump to m.
03.2	AZJ,GE	" , greater or equal	If $(A)\geq+0$, jump to m.

03.3	AZJ,LT	A zero jump, less than	If $(A)<+0$, jump to m.
03.1	AZJ,NE	" , not equal	If $(A)\neq 0$, jump to m.
00.0	HLT	Halt	ABNORMAL STOP. Jump to m upon restarting.
00.7	RTJ	Return jump	$(P)+1 \longrightarrow (m_{14-0})$, jump to m+1.
00.1	SJ1	Select jump 1	If jump key 1 is set, jump to m.
00.2	SJ2	Select jump 2	If jump key 2 is set, jump to m.
00.3	SJ3	Select jump 3	If jump key 3 is set, jump to m.
00.4	SJ4	Select jump 4	If jump key 4 is set, jump to m.
00.5	SJ5	Select jump 5	If jump key 5 is set, jump to m.
00.6	SJ6	Select jump 6	If jump key 6 is set, jump to m.
10.0	SSH	Storage shift	Test (m_{23}). Shift (m) left end-around one place, store back in (m). Then, if (m_{23}) was 1, skip one instruction.

Note: on tests for equality (=) or for non-equality (≠), +0 and -0 are treated as being equal. On tests for greater or equal (≥) or for less than (<), +0 is considered to be greater than -0.

Index jumps.

6	1	2	15
f	g	b	m
23 18	17	16 15	14 0

f (bits 23-18) is function code.
g (bit 17) is sub-operation.
b (bits 16-15) selects a B register.
m (bits 14-0) is a word address (for jumping).

The b field specifies which B register is to be tested and incremented or decremented. The g field specifies whether to increment or decrement. The COMPASS format is

```
OPC     M,B
```

f,gb	mnemonic op code	Name	Operation
02.(4+b)	IJD	Index jump decremental	If $(B^b)\neq 0$, $(B^b)-1\rightarrow(B^b)$ and jump to m. If $(B^b)=0$, take next instruction in sequence.
02.b	IJI	Index jump incremental	If $(B^b)\neq 0$, $(B^b)+1\rightarrow(B^b)$ and jump to m. If $(B^b)=0$, take next instruction in sequence.

Masked search.

6	3	15
f	i	m
23 18	17 15	14 0

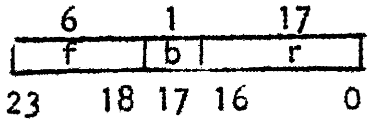
i (bits 17-15) is interval for search. i = 0 denotes an interval of 8; i = 1,2,...,7 denote intervals of 1,2,...,7 respectively. The COMPASS format is:

OPC M,I

I denotes the interval (1,2...,8). A value of I = 8 causes 0 to be placed in the i field.

f	mnemonic op code	Name	Operation
06	MEQ	Masked equality search	$(B^1)-i\rightarrow(B^1)$; if (B^1) changed sign from positive to negative, take next instruction in sequence. If not, test $(A)=(Q)\wedge(M)$. If true, skip next instruction; if false, repeat this sequence. Note: $M=m+(B^1)$.
07	MTH	Masked threshold search	$(B^2)-i\rightarrow(B^2)$; if (B^2) changed sign from positive to negative, take next instruction in sequence. If not, test $(A)\geq(Q)\wedge(M)$. If true, skip next instruction; if false, repeat this sequence. Note: $M=m+(B^2)$.

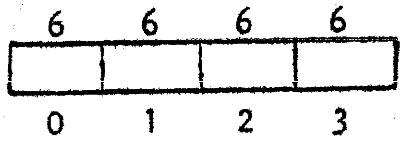
Character reference.



b (bit 17) specifies address modification. If b=0, no modification. If b=1, (B^1) or (B^2) is selected, depending on instruction.

r (bits 16-0) is basic character address.

The b and r fields are interpreted as follows: An effective character address R is computed by adding the selected B register (with sign extended) to r: $R=r+(B^b)$. If b=0, R=r. The result is a 17 bit address in which the leftmost 15 bits select a word in memory. The rightmost 2 bits determine the character in the word which is to be referenced. Characters are numbered left to right:



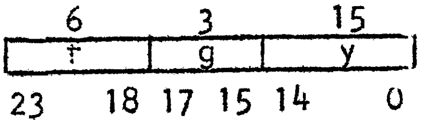
The COMPASS format is

OPC R,B

R denotes the basic character address, which may be an expression, usually involving symbolic addresses. If address modification is desired, the address field is followed by a comma and an expression whose value is 1 (for LACH or SQCH) or 2 (for SACH or LQCH). In this case, a 1 is placed in the b field. If the B field is omitted, a 0 is placed in the b field.

f	mnemonic op code	address field	Name	Operation
22	LACH	r ,1	Load A character	$0 \rightarrow (A)$; $(R) \rightarrow (A_{5-0})$
23	LQCH	r ,2	Load Q character	$0 \rightarrow (Q)$; $(R) \rightarrow (Q_{5-0})$
42	SACH	r ,2	Store A character	$(A_{5-0}) \rightarrow (R)$
43	SQCH	r ,1	Store Q character	$(Q_{5-0}) \rightarrow (R)$

Operand-in-instruction.



g (bits 17-15) selects which register is to be used and determines whether or not the sign is to be extended. The selection is as follows:

g	Register selected	g	Register selected
0	None	4	A, with sign extension on y
1	B1	5	Q, with sign extension on y
2	B2	6	A, no sign extension
3	B3	7	Q, no sign extension

y (bits 14-0) is a 15 bit operand. In operations with A or Q, y is extended to a 24 bit operand by placing 0 bits at the left (if no sign extension is selected), or by placing copies of the sign bit (bit 14) at the left (if sign extension is selected).

The COMPASS formats are

OPC	Y,B	for instructions that select a B register
OPC	Y	for instructions that select A or Q without sign ext.
OPC,S	Y	for instructions that select A or Q with sign ext.

Y is a number or expression that specifies the desired operand. [Sy in the table below denotes y with sign extension.]

f.g	mnemonic op code	address field	Name	Operation
17.6	ANA	Y	And to A	$(A) \wedge y \rightarrow (A)$
17.4	ANA,S	Y	" , sign extended	$(A) \wedge Sy \rightarrow (A)$
17.b	ANI	Y,B	And to index	$(B^b) \wedge y \rightarrow (B^b)$
17.7	ANQ	Y	And to Q	$(Q) \wedge y \rightarrow (Q)$
17.5	ANQ,S	Y	" , sign extended	$(Q) \wedge Sy \rightarrow (Q)$
04.6	ASE	Y	A skip if equal	Skip next instruction if $(A_{14-0})=y$
04.4	ASE,S	Y	" , sign extended	Skip if $(A)=Sy$
05.6	ASG	Y	A skip, greater or equal	Skip if $(A_{14-0}) \geq y$
05.4	ASG,S	Y	" , sign extended	Skip if $(A) \geq Sy$
14.6	ENA	Y	Enter A	$0 \rightarrow (A)$, then $y \rightarrow (A_{14-0})$
14.4	ENA,S	Y	" , sign extended	$Sy \rightarrow (A)$
14.b	ENI	Y,B	Enter index	$y \rightarrow (B^b)$

f.g	mnemonic op code	address field	Name	Operation
14.7	ENQ	Y	Enter Q	$0 \rightarrow (Q)$, then $y \rightarrow (Q_{14-0})$
14.5	ENQ,S	Y	" , sign extended	$Sy \rightarrow (Q)$
15.6	INA	Y	Increase A	$(A) + y \rightarrow (A)$
15.4	INA,S	Y	" , sign extended	$(A) + Sy \rightarrow (A)$
15.b	INI	Y,B	Increase index	$(B^b) + y \rightarrow (B^b)$
15.7	INQ	Y	Increase Q	$(Q) + y \rightarrow (Q)$
15.5	INQ,S	Y	" , sign extended	$(Q) + Sy \rightarrow (Q)$
04.0	ISE	Y	Index skip if equal	Skip if $0 = y$
04.b	ISE	Y,B	"	Skip if $(B^b) = y$
05.0	ISG	Y	Index skip, greater or equal	Skip if $0 \geq y$
05.b	ISG	Y,B	"	Skip if $(B^b) \geq y$
04.7	QSE	Y	Q skip if equal	Skip if $(Q_{14-0}) = y$
04.5	QSE,S	Y	" , sign extended	Skip if $(Q) = Sy$
05.7	QSG	Y	Q skip, greater or equal	Skip if $(Q_{14-0}) \geq y$
05.5	QSG,S	Y	" , sign extended	Skip if $(Q) \geq Sy$
16.6	XOA	Y	Exclusive or to A	$(A) \not\vee y \rightarrow (A)$
16.4	XOA,S	Y	" , sign extended	$(A) \not\vee Sy \rightarrow (A)$
16.b	XOI	Y,B	Exclusive or index	$(B^b) \not\vee y \rightarrow (B^b)$
16.7	XOQ	Y	Exclusive or to Q	$(Q) \not\vee y \rightarrow (Q)$
16.5	XOQ,S	Y	" , sign extended	$(Q) \not\vee Sy \rightarrow (Q)$

Index skips.

6	1	2	15
f	g	b	y
23 18	17	16 15	14 0

g (bit 17) is sub-operation. Determines whether to increment or decrement B register selected by b (bits 16-15).
y (bits 14-0) is a 15 bit operand. The selected B register is compared with y.

The COMPASS format is OPC Y,B

f.gb	mnemonic op code	address field	Name	Operation
10.(4+b)	ISD	Y,B	Index skip decremental	If $(B^b) \neq y$, $(B^b)-1 \rightarrow (B^b)$. If $(B^b) = y$, $0 \rightarrow (B^b)$ and skip next instruction.
10.b	ISI	Y,B	Index skip incremental	If $(B^b) \neq y$, $(B^b)+1 \rightarrow (B^b)$. If $(B^b) = y$, $0 \rightarrow (B^b)$ and skip.

Note: 10.0 is SSH (storage shift) instruction.

17-bit operand-in-instruction.

6	1	17
f	g	r
23 18	17	16 0

g (bit 17) specifies sign extension (g=1) or no sign extension (g=0).

r (bits 16-0) is a 17 bit operand.

The COMPASS format is OPC,S R
Sign extension (comma, S) is optional. R is an expression denoting either a 17 bit quantity or a symbolic character address.

f.g	mnemonic op code	Name	Operation
11.0	ECHA	Enter character address	$0 \rightarrow (A)$, then $r \rightarrow (A_{16-0})$
11.4	ECHA,S	" , sign extended	$Sr \rightarrow (A)$

Shift instructions.

6	1	2	15
f	g	b	k
23 18	17	16 15	14 0

g (bit 17) is sub-operation.
b (bits 16-15) selects a B register in usual way.
k (bits 14-0) is shift parameter.

On all shift instructions except SCAQ, a modified shift parameter K is computed by adding the selected B register to k : $K = k + (B^b)$. If b=0, K=k. The value of K determines how many places to shift, and also which direction to shift. If $K \geq +0$, the register is shifted left end-around K places. If $K \leq -0$, the register is shifted right -K places, with the left-most bit (sign bit) being extended. Only the rightmost 6 bits of K are examined to determine how many places to shift.

The COMPASS format is OPC K,B

K is an expression denoting the basic shift parameter. The B field is optional; if present, it specifies the B register to be used.

f.gb	mnemonic op code	Name	Operation
13.(4+b)	SCAQ	Scale AQ	Shift (AQ left end-around until the leftmost two bits of (A) are not equal ($A_{23} \neq A_{22}$). (If (AQ)=+0 or -0, 48 shifts are made.) The residue K=k-(number of places shifted). $K \rightarrow (B^b)$. If b=0, K is discarded.

f.gb	Mnemonic op code	Name	Operation
12.b	SHA	Shift A	(A) is shifted K places. (See above)
13.b	SHAQ	Shift AQ	(AQ) is shifted K places.
12.(4+b)	SHQ	Shift Q	(Q) is shifted K places.

Inter-register transfers group 1.

6	3	15
f	g	u
23 18	17 15	14 0

g (bits 17-15) selects registers to be transfered.
u (bits 14-0) is unused.

In the following descriptions, E_U denoted the upper 24 bits of the 48 bit E register, and E_L denotes the lower 24 bits.

The COMPASS format is OPC

There is no address field.

f.g	Mnemonic op code	Name	Operation
55.6	AEU	A to E upper	(A) → (E_U)
55.7	AQE	AQ to E	(AQ) → (E)
55.3	EAQ	E to AQ	(E) → (AQ)
55.1	ELQ	E lower to Q	(E_L) → (Q)
55.2	EUA	E upper to A	(E_U) → (A)
55.5	QEL	Q to E lower	(Q) → (E_L)

Inter-register transfers group 2.

6	1	2	3	6	6
f	d	b	g	u	v
23 18	17	16 15	14 12	11 6	5 0

d (bit 17) determines the direction of the transfer.
b (bits 16-15) selects a B register, if one of them is used in the transfer.
g (bits 14-12) is sub-operation. Determines which registers are to be used.
u (bits 11-6) is unused.
v (bits 5-0) selects one of the 64 registers in the register file, if one of them is to be used in the transfer.

The COMPASS formats are:

```
OPC
OPC    B
OPC    V
OPC    V,B
```

B denotes B register selection; V is an expression whose value is in the range 0 to 63_{10} (0 to 77_8), which selects a register in the register file. [M in mnemonics below refers to register file.]

f.dbg	mnemonic op code	address field	Name	Operation
53.b4	AIA	B	A plus index to A	$(A)+(B^b) \rightarrow (A)$, sign extended on (B^b).
53.04	AQA		A plus Q to A	$(A)+(Q) \rightarrow (A)$
53.(4+b)4	IAI	B	Index plus A to index	$(B^b)+(A) \rightarrow (B^b)$, sign extended on (B^b) before adding.
53.(4+b)0	TAI	B	Transfer A to index	$(A_{14-0}) \rightarrow (B^b)$.
53.42	TAM	V	Transfer A to M	$(A) \rightarrow (v)$
53.b0	TIA	B	Transfer index to A	$0 \rightarrow (A)$; $(B^b) \rightarrow (A_{14-0})$
53.(4+b)3	TIM	V,B	Transfer index to M	$0 \rightarrow (v)$; $(B^b) \rightarrow (v_{14-0})$
53.02	TMA	V	Transfer M to A	$(v) \rightarrow (A)$
53.b3	TMI	V,B	Transfer M to index	$(v_{14-0}) \rightarrow (B^b)$
53.01	TMQ	V	Transfer M to Q	$(v) \rightarrow (Q)$
53.41	TQM	V	Transfer Q to M	$(Q) \rightarrow (v)$

No-address instructions.

6	6	12
f	g	u
23 18	17 12	11 0

g (bits 17-12) is sub-operation. (In NOP, g is bits 17-15)
u (bits 11-0) is unused.

The COMPASS format is OPC . There is no address field.

f.g	Mnemonic op code	Name	
77.75	CTI	Console Teletype Input	0 → A Read (ASCII) from TTY $A_{u-0} \to A_7 = 1$ Other registers not affected
77.76	CTO	Console teletype Output	$A_{7-0} \to$ Output TTY (ASCII) TTY Input Buffer cleared registers not affected.
77.73	DINT	Disable interrupts	
55.0	RIS	Relocate with instruction state mode	Operands are fetched from same memory bank
55.4	ROS	Relocate with operand state mode	Operands are fetched from other bank memory
77.620	SBJP	Set boundary jump	Normal Termination of User's program.
77.70	SLS	Selective Stop	Abnormal termination of program
77.0	m	Jump	Jump to M (16 bit address) If bit 15 = 0, lower memory If bit 15 = 1, upper memory

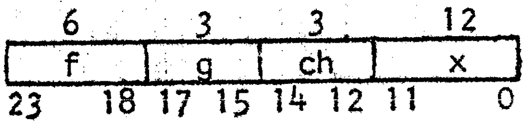
6	2	16
f	g	m
23 18	17 16	15 0

g = 00

77.54	ACI	ASCII code to imput string	Place A_{7-0} (ASCII Character code) at end of teletype input string.

f.g	mnemonic op code	Name and Operation
77.74	EINT	Enable Interrupts
77.57	IAPR	Interrupt associated processor
14.0	NOP	No operation
77.72	SBCD	Set BCD fault indicator
77.71	SFPF	Set floating point fault indicator
77.70	SLS	Selective stop. Stop if Select Stop switch is on; when restarted, take next instruction in sequence.
77.77	UCS	Unconditional stop. Stop; when restarted, take next instruction in sequence.

Interrupt and input/output control:



g (bits 17-15) is sub-operation.
ch (bits 14-12) selects an input/output channel (0-7) in some instructions below; in those with g = 5, it is a sub-operation specifier.
x (bits 11-0) is a 12 bit code or mask in which the individual bits select various conditions or devices.

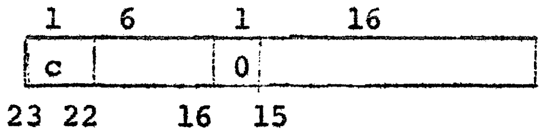
The COMPASS formats are OPC CH
OPC X,CH

CH is channel designator (0 to 7). X is an expression or number denoting the 12 bit mask desired.

f.g	mnemonic op code	address field	Name	Operation
77.3	CINS	CH	Copy internal status	Internal status code → (A_{11-0}); (interrupt mask register) → (A_{23-12}). (x = 0)
77.0	CON	X,CH	Connect	If channel CH is busy, take next instruction. If not, send 12 bit connect code (x) on channel CH, skip next instruction.

f.g	mnemonic op code	address field	Name	Operation
77.2	COPY	CH	Copy external status	External status code from channel CH $\rightarrow$ (A_{11-0}); (interrupt mask register) $\rightarrow$ (A_{23-12}). ($x = 0$)
77.2	EXS	X,CH	Sense external status	If any external status line selected by X is 1, take next instruction; if not, skip next instruction. ($x \neq 0$)
77.50	INCL	X	Clear interrupt	Clear internal interrupt faults selected by X.
77.3	INS	X,CH	Sense internal status	If any internal status line selected by X is a 1, take next instruction; if not, skip. ($x \neq 0$)
77.4	INTS	X,CH	Sense interrupt	If any interrupt condition selected by X is a 1, take next instruction; if not, skip.
77.51	IOCL	X	Clear I/O, etc.	Clear I/O channels, typewriter, or search/move control as defined by X.
77.60	PAUS	X	Pause	Sense busy lines. If a 1 appears on any line selected by X, pause. If 40 msec elapses, take next instruction. If no selected line is a 1, or all selected lines become 0 during pause, skip next instruction.
77.53	SCIM	X	Selectively clear interrupt mask	Clear to 0 the bits in the interrupt mask register corresponding to 1 bits in X.
77.1	SEL	X,CH	Select function	If channel CH is busy, etc., take next instruction. If not, send 12 bit function code X to unit presently connected to channel CH and skip next instruction.

INTERRUPT stores word in m and jumps to m + 1



← Address of instruction to be executed next.

C = 1 RIS d = INTERNAL STATUS

0 ROS

m	Condition
2	BCD FAULT
4	DIVIDE
6	ARITHMETIC OVERFLOW
10_8	EXPONENTIAL FAULT
12_8	MANUAL INTERRUPT

INTERNAL STATUS

BIT	MASK	CONDITION
11	4000	BCD FAULT
10	2000	DIVIDE
9	1000	ARITHMETIC OVERFLOW
8	0400	EXPONENTIAL FAULT

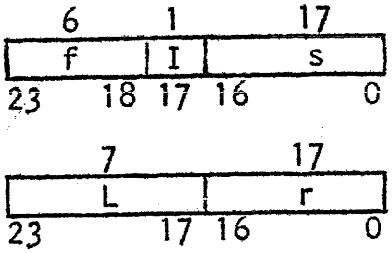
INTERRUPT MASK REGISTER

9	1000	EXPONENTIAL AND BCD FAULT
10	2000	DIVIDE AND ARITHMETIC OVERFLOW

SCIM AND SSIM SETS OR CLEARS INTERRUPT MASK REGISTER.

f.g	mnemonic op code	address field	Name	Operation
77.52	SSIM	X	Selectively set interrupt mask	Set to 1 the bits in the interrupt mask register corresponding to 1 bits in X.

Move instruction.



I (bit 17 of 1st word) is interrupt indicator. If I = 1, interrupt on completion of operation.

r (bits 16-0 of 2nd word) is address of first character in source block.

s (bits 16-0 of 1st word) is address of first character in destination block.

L (bits 23-17 of 2nd word) is length of field (number of characters to be moved). If L = 0, 128 characters are moved.

The COMPASS format is OPC,INT L,R,S

If the op code is followed by (comma, INT), a 1 is placed in the I field. L is a number in the range 1 to 128, denoting the number of characters to be moved. R and S are character addresses (may be expressions) denoting the first characters of the blocks to be moved: from (R) to (S).

f	mnemonic op code	Name	Operation
72	MOVE	Move	If search/move control is busy, take next instruction. If not, start copying characters from R to S, R+1 to S+1, etc., until L characters have been moved (if L=0, move 128 characters), and skip next instruction. If I=1, interrupt when move is completed.

Search instructions.

6	1	17
f	I	s
23 18	17	16 0

6	1	17
c	g	r
23 18	17	16 0

I (bit 17 of 1^{st} word) is interrupt indicator. If I = 1, interrupt on completion of operation.

r (bits 16-0 of 2^{nd} word) is address of first character in block to be searched.

s (bits 16-0 of 1^{st} word) is address of last character, plus 1, in search block.

c (bits 23-18 of 2^{nd} word) is BCD code of search character.

g (bit 17 of 2^{nd} word) is sub-operation. g = 0: search for equality; g = 1: search for inequality.

The COMPASS format is OPC,INT C,R,S

If the op code is followed by (comma, INT), a 1 is placed in the I field. C is an expression whose value is in the range 0 to 77_8, denoting the BCD code of the character to be searched for. R is the character address of the first character to be examined and S is the character address, plus 1, of the last character to be examined.

f.g	mnemonic op code	Name	Operation
71.0	SRCE	Search for character equality	If search/move control is busy, take next instruction. If not, start comparing character C with characters (R), (R+1), ..., (S-1), and skip next instruction. If a match occurs, stop search. Address of matching character is in register 20_8. When operation complete, interrupt if I = 1.
71.1	SRCN	Search for character inequality	Same as SRCE, except terminate search when no match is found.

Character-addressed I/O with storage.

6	1	17
f	g	s
23 18	17	16 0

3	1	1	1	1	17
ch	G	B	H	I	r
23 21	20	19	18	17	16 0

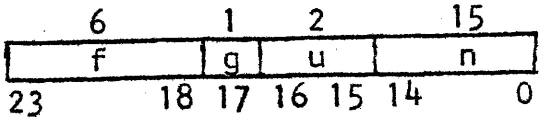
g is sub-operation. g = 0 for I/O operations with storage.
r is address of first character of data block.
s is address of last character of data block, plus one (minus one for backward operation).
ch is I/O channel designator (0 to 7).
I = 1 for interrupt on completion.
H = 1 for half word assembly/disassembly.
B = 1 for backward storage operation.
G = 1 for word count control.

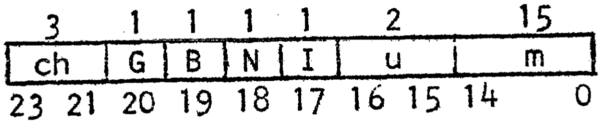
The COMPASS format is OPC,INT,B,H CH,R,S

The optional modifiers (INT, B, and H) cause 1 bits to be placed in the corresponding fields in the instruction, if they are present. CH is an expression whose value is 0 to 7, denoting the I/O channel. R and S are character address expressions.

f.g	mnemonic op code	Name	Operation
73.0	INPC	Input characters to storage	If channel CH is busy, take next instruction. If not, start reading characters from I/O device connected to channel CH, storing them in memory in characters (R), (R+1), ..., (S-1), and skip next instruction. If I = 1, interrupt on completion.
75.0	OUTC	Output characters from storage	If channel CH is busy, take next instruction. If not, start sending characters from storage locations (R), (R+1), ..., (S-1) to I/O device connected to channel CH, and skip next instruction. If I = 1, interrupt on completion.

Word-addressed I/O with storage.





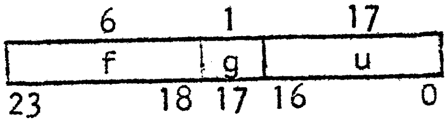
m is address of first word of data block.
n is address of last word of data block, plus one (minus one for backward operation).
N = 1 for no assembly.
u is unused.
Other fields have same meaning as in character-addressed I/O operations (see page 19).

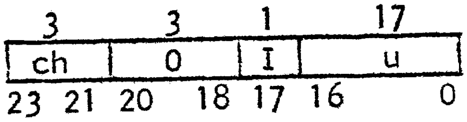
The COMPASS format is OPC,INT,B,N CH,M,N

M and N are word address expressions.

f.g	mnemonic op code	Name	Operation
74.0	INPW	Input words to storage	If channel CH is busy, take next instruction. If not, start reading words from I/O device connected to channel CH, storing them in words (M), (M+1), ..., (n-1), and skip next instruction. If I = 1, interrupt on completion.
76.0	OUTW	Output words from storage	If channel CH is busy, take next instruction. If not, start sending words from storage locations (M), (M+1), ..., (N-1), to I/O device connected to channel CH, and skip next instruction. If I = 1, interrupt on completion.

Input/output with A.





g is sub-operation. g = 1 for I/O operations with A.
ch is I/O channel (0 to 7).
I = 1 for interrupt on completion.
0 (bits 20-18 of 2nd word) is zero.
u is unused.

The COMPASS format is OPC,INT CH

If the optional modifier INT is present, a 1 is placed in the I field. CH denotes the I/O channel (0 to 7).

f.g	mnemonic op code	Name	Operation
73.1	INAC	Input character to A	If channel CH is busy, take next instruction. If not, clear A and input a character from I/O device on channel CH to (A_{5-0}), skip next instruction. If I = 1, interrupt.
74.1	INAW	Input word to A	Same as INAC, except clear A and input a word to (A_{11-0}) or (A_{23-0}).
75.1	OTAC	Output character from A	If channel CH is busy, take next instruction. If not, transfer character from (A_{5-0}) to I/O device on channel CH, skip next instruction. If I = 1, interrupt.
76.1	OTAW	Output word from A	Same as OTAC, except output a word from (A_{11-0}) or (A_{23-0}).

INPUT/OUTPUT

6	1	2	15
f	a	b	L

$L = \ell + (B^b)$

if a = 1 fetch bit 17 - 0 from memory location L and repeat
if a = 0 L specifies the logical unit no (lun) $0 \leq \text{lun} \leq 99$

f	Mnemonic op code	Address field	Operation
72	Control CNTL,I	L,b	Issue control code in lower 15 bits of q_{14-0} to Lun. After operation status returned in A other registers unchanged

CONTROL CODES

SYMBOLIC NAME	OCTAL CODE	
STATUS	0	copy status of L to A
CLEAR	1	clear status of L (file mark & binary bits
WFM	2	write file mark
RELEASE	3	release file or output unit
REWING	4	rewind to load pt
SFPFM	5	search forward past file mark
SBPFM	6	search backward past file mark
BKSPACE	7	backspace one record
FWDSPACE	10	space forward one record

Status of device in A

9	15
STATUS	TYPE
24 15	14 0

STATUS

BIT	MASK	CONDITION
23	40000000	READ ONLY (FILE PROTECT)
22	20000000	LOAD PT (BEGINNING OF FILE)
21	10000000	END OF DATA
20	04000000	FM just processed
19	02000000	NOT USED

TYPE

0 = UNIT NOT EQUIPPED
1 = FILE
2 = LINE PRINTER
3 = CARD PUNCH
4 = CARD READER

STATUS

BIT	MASK	CONDITION
18	01000000	BINARY RECORD just processed
17	00400000	ABNORMAL/UNAVAILABLE
16	00200000	ADRESS ERROR Locate
15	00100000	SAVED FILE

f.g	Mnemonic op code	Address field
74	READ,I	L,B
76	WRITE,I	L,B

TYPE

5 = MAGNETIC TAPE
6 = TELETYPE
7 = PLOTTER
8 = NULL (ABSORBS ALL OUTPUT)
9 = CRT (DISPLAY CONSOLE)
10 = RANDOM ACCESS FILE
11 = TASK (REMOTE BATCHJOB)

		1	15
A		c	Address to start at

c = 0 start in same bank
= 1 start in other bank

		1	2	16
Q		A	B	no of words to read or write
	~~24~~ 23 20	19 18	17 16	

A = 0 BCD MODE (bit 18)
= 1 BINARY
B = 00 (bit 17,16)

AFTER READ OR WRITE STATUS IN A
AFTER READ Q IS SET EQUAL TO ORIGINAL VALUE IN LOWER 16 BITS - NO OF WORDS IN THE RECORD

f.g	Mnemonic op code	Address field	Name
71	XREQ,I	L,B	Executive Request see below.

EXECUTIVE REQUEST instruction

This is a simulated instruction (under OS-3) which is used to equip logical units, save files, call library programs, etc. The registers, etc., are used as follows:

AQ Name of file or library program, in BCD character codes. Is left unchanged except in case of EQUIP request.

B1 Code for action desired. After operation, contains an error code (0 means no error).

effective address (L): logical unit number (0 to 99) or memory page number (0 to 37_8), or not used.

CODE in B1	Request	Action and error conditions
0	DELETE	Delete file name (in AQ) from file directory. File must be equipped as the specified logical unit. ERROR 1 unit is not equipped 2 file is protected 3 unit is not a saved file 4 name in AQ does not agree with name of file 5 not enough scratch file space for file
1	SAVE	Save specified logical unit under name in AQ (put name in file directory). ERROR 1 unit is not equipped 2 there already exists a file with the name given in AQ 3 unit is already a saved file 4 unit is not a file 5 not enough saved file space for file 6 name is illegal (such as FILE, PUN, etc.)
2	UNEQUIP	Unequip the specified logical unit. AQ is ignored. ERROR 1 unit is not equipped 2 unit is a file which is protected and not saved
3	EQUIP	If A ≠ 0, equip specified unit as the saved file whose name is in AQ, or as the hardware unit whose name is in AQ. See list below. If A = 0, equip specified unit as equivalent to the unit number specified is Q(0 to 100). If no error occurs, A contains the status of the unit after equipping it. HARDWARE NAMES (in AQ): FILE create an empty scratch file LP line printer PR same as LP PUN card punch PLOT plotter

XREQ (con't)

RAF create an empty random access file
TASK remote batch job
NULL null (device absorbs and discards all outputs to it)
MT (in A) magnetic tape
no. (in Q) reel number in Q.

<u>ERROR</u>

1 unit is already equipped
2 there is no saved file with the name given in AQ (or, perhaps, file is busy)
3 unit number (in Q) is not equipped
4 saved file with name in AQ is busy (it is not protected and some other user has it equipped).
5 not enough tape drives available
6 illegal tape number or density
7 not enough hardware available

4 RFP Remove file protection from specified unit. If it is saved, its name must be given in AQ. If it is not saved, AQ is ignored.

<u>ERROR</u>

1 unit is not equipped
2 file is busy (some other user is equipped to it, too).
3 name in AQ does not agree with the name of the file
4 file is public (first character of name is an asterisk) and it belongs to some other user.

5 FP Protect the file which is equipped to the specified unit AQ is ignored.

<u>ERROR</u>

1 unit is not equipped
2 unit is not a file

6 unused

7 ZEROPAGE Effective address (0 to 37$_8$) specifies a page in memory, which is set to all zero. A page is 2048 words, and they are numbered from the low end of memory upward. Thus, page 0 extends from address 000000 to 003777, page 13 extends from 054000 to 057777, etc. AQ is ignored.

XREQ (con't)

8 LIBCALL

Call the library program whose name is in AQ. This involves copying the program into the pages of memory which it is to occupy, and transfering control to it. If the library program needs a parameter string, this must have been defined previously by storing ASCII characters, using the ACI instruction. The effective address is ignored.

ERROR

If there is no library program by the name given in AQ, nothing is done and the computer executes the next instruction after the XREQ.

	Mnemonic op code	Address field	Page		Mnemonic op code	Address field	Page
$	ACI		14½				
	ADA,I	M,B	5		ELQ		13
	ADAQ,I	M,B	5		ENA	Y	10
	AEU		13		ENA,S	Y	10
	AIA	B	14		ENI	Y,B	10
	ANA	Y	10		ENQ	Y	11
	ANA,S	Y	10		ENQ,S	Y	11
	ANI	Y,B	10		EUA		13
	ANQ	Y	10	*	EXS	X,CH	16
	ANQ,S	Y	10		FAD,I	M,B	5
	AQA		14		FDV,I	M,B	5
	AQE		13		FMU,I	M,B	5
	AQJ,EQ	M	6		FSB,I	M,B	5
	AQJ,GE	M	6	$	HLT	M	7
	AQJ,LT	M	6		IAI	B	14
	AQJ,NE	M	6	*	IAPR		15
	ASE	Y	10		IJD	M,B	8
	ASE,S	Y	10		IJI	M,B	8
	ASG	Y	10		INA	Y	11
	ASG,S	Y	10		INA,S	Y	11
	AZJ,EQ	M	6	*	INAC,INT	CH	21
	AZJ,GE	M	6	*	INAW,INT	CH	21
	AZJ,LT	M	7	*	INCL	X	16
	AZJ,NE	M	7		INI	Y,B	11
$	CINS	CH	15	*	INPC,INT,B,H	CH,R,S	19
$	CNTL	M,B	22				
~~$~~	~~CONTROL,I~~	M,B	22	*	INPW,INT,B,N	CH,M,N	20
*	CON	X,CH	15		INQ	Y	11
*	COPY	CH	16		INQ,S	Y	11
	CPR,I	M,B	5	$	INS	X,CH	16
$	CTI		14½				
$	CTO		14½	*	INTS	X,CH	16
*	DINT		14½	*	IOCL	X	16
	DVA,I	M,B	5		ISD	Y,B	11
	DVAQ,I	M,B	5		ISE	Y,B	11
	EAQ		13		ISG	Y,B	11
	ECHA	R	12		ISI	Y,B	11
	ECHA,S	R	12		LACH	R,1	9
*	EINT		15		LCA,I	M,B	5

$ Simulated under OS-3
* Not usable by user program under OS-3

Mnemonic op code	Address field	Page
LCAQ,I	M,B	5
LDA,I	M,B	5
LDAQ,I	M,B	5
LDI,I	M,B	5
LDL,I	M,B	5
LDQ,I	M,B	5
LPA,I	M,B	5
LQCH	R,2	9
MEQ	M,I	8
* MOVE,INT	L,R,S	17
MTH	M,I	8
MUA,I	M,B	5
MUAQ,I	M,B	5
NOP		15
* OTAC,INT	CH	21
* OTAW,INT	CH	21
* OUTC,INT,B,H	CH,R,S	19
* OUTW,INT,B,N	CH,M,N	20
* PAUS	X	16
QEL		13
QSE	Y	11
QSE,S	Y	11
OSG	Y	11
QSG,S	Y	11
RAD,I	M,B	5
$ READ,I	M,B	23
RIS	H	14½
ROS	H	14½
RTJ	M	7
SACH	R,2	9
SBA,I	M,B	6
SBAQ,I	M,B	6
SBCD		15
$ SBJP		14½
SCA,I	M,B	6
SCAQ	K,B	12
SCHA,I	M,B	6
$ SCIM	X	16
* SEL	X,CH	16
SFPF		15
* SLS		14½
SHA	K,B	13
SHAQ	K,B	13
SHQ	K,B	13
SJ1	M	7
SJ2	M	7
SJ3	M	7
SJ4	M	7
SJ5	M	7
SJ6	M	7
* SLS		15,14½
SQCH	R,1	9
* SRCE,INT	C,R,S	18
* SRCN,INT	C,R,S	18
SSA,I	M,B	6
SSH	M	7
$ SSIM	X	17
STA,I	M,B	6
STAQ,I	M,B	6
STI,I	M,B	6
STQ,I	M,B	6
SWA,I	M,B	6
TAI	B	14
TAM	V	14
TIA	B	14
TIM	V	14
TMA	V,B	14
TMI	V,B	14
TMQ	V	14
TQM	V	14
* UCS		15
UJP,I	M,B	6
$ WRITE,I	M,B	23
XOA	Y,B	11
XOA,S	Y	11
XOI	Y	11
XOQ	Y	11
XOQ,S	Y	11
$ XREQ,I	M,B	24
$ 77		14½

CDC3300 CHARACTER CODES

Appendix B

BCD Code	Card Code	Key Punch	Line Prntr	Tele-type	ASCII Code	BCD Code	Card Code	Key Punch	Line Prntr	Tele-type	ASCII Code
00	0	0	0	0	260	40	11	-	-	-	255
01	1	1	1	1	261	41	11,1	J	J	J	312
02	2	2	2	2	262	42	11,2	K	K	K	313
03	3	3	3	3	263	43	11,3	L	L	L	314
04	4	4	4	4	264	44	11,4	M	M	M	315
05	5	5	5	5	265	45	11,5	N	N	N	316
06	6	6	6	6	266	46	11,6	O	O	O	317
07	7	7	7	7	267	47	11,7	P	P	P	320
10	8	8	8	8	270	50	11,8	Q	Q	Q	321
11	9	9	9	9	271	51	11,9	R	R	R	322
12	2,8		:	:	272	52	11,0		∨	!	241
13	3,8	=	=	=	275	53	11,3,8	$	$	$	244
14	4,8	'	≠	'	247	54	11,4,8	*	*	*	252
15	5,8		<	&	246	55	11,5,8		↑	↑	336
16	6,8		%̄	%	245	56	11,6,8		↓	@	300
17	7,8		[	[	333	57	11,7,8		>	>	276
20	12	+	+	+	253	60	Blank	Blk	Blk	Sp	240
21	12,1	A	A	A	301	61	0,1	/	/	/	257
22	12,2	B	B	B	302	62	0,2	S	S	S	323
23	12,3	C	C	C	303	63	0,3	T	T	T	324
24	12,4	D	D	D	304	64	0,4	U	U	U	325
25	12,5	E	E	E	305	65	0,5	V	V	V	326
26	12,6	F	F	F	306	66	0,6	W	W	W	327
27	12,7	G	G	G	307	67	0,7	X	X	X	330
30	12,8	H	H	H	310	70	0,8	Y	Y	Y	331
31	12,9	I	I	I	311	71	0,9	Z	Z	Z	332
32	12,0		<	<	274	72	0,2,8		]	]	335
33	12,3,8	.	.	.	256	73	0,3,8	,	,	,	254
34	12,4,8	)	)	)	251	74	0,4,8	(	(	(	250
35	12,5,8		≥	#	243	75	0,5,8		↱	\	334
36	12,6,8		¬	"	242	76	0,6,8		≡	←	337
37	12,7,8		;	;	273	77	0,7,8		∧	?	277

Other teletype characters & their ASCII codes:

Bell	207	Horizontal Tab	211
Line Feed	212	Vertical Tab	213
Return	215	Form Feed	214
Rubout	377	Alt Mode & Escape	233, 374,375,376

Display unit uses BCD codes, with line printer characters, except:

36 - (carriage return)
37 ▲ (send)
75 ■ (parity error)
76 ' (print)

INPUT/OUTPUT

f	a	b	L

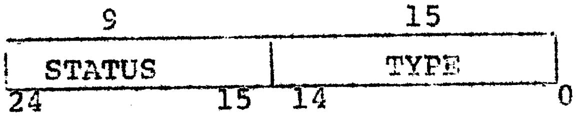
$L = \ell + (B^b)$

if a = 1 fetch bit 17-0 from memory location L and repeat
if a = 0 L specifies the logical unit no (lun) $0 \leq \text{lun} \leq 99$

f.g	Mnemonic op code	Address field	Operation
72	Control CNTL,I	L,b	Issue control code in lower 15 bits of q_{14-0} to Lun. After operation status returned in A other registers unchanged

SYMBOLIC NAME	OCTAL CODE	
STATUS	0	copy status of L to A
CLEAR	1	clear status of L
WFM	2	write file mark
RELEASE	3	release file or output unit
REWIND	4	rewind to load pt
SFPFM	5	search forward past file mark
SBPFM	6	search backward past file mark
BK SPACE	7	backspace one record
FWD SPACE	10	space forward one record

Status of device in A



BIT	STATUS MASK	CONDITION	TYPE
23	40000000	READ ONLY (FILE PROTECT)	0 = UNIT NOT EQUIPPED
22	20000000	LOAD PT (BEGINNING OF FILE)	1 = FILE
21	10000000	END OF DATA	2 = LINE PRINTER
20	04000000	FM just processed	3 = CARD PUNCH
19	02000000	NOT USED	4 = CARD READER

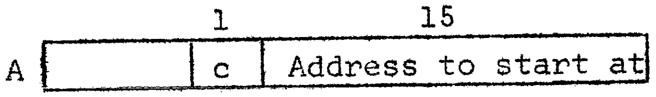
STATUS

BIT	MASK	CONDITION
18	01000000	BINARY RECORD just processed
17	00400000	ABNORMAL/UNAVAILABLE
16	00200000	~~UNUSED~~ LOCATE [illegible]
15	00100000	SAVED FILE

TYPE

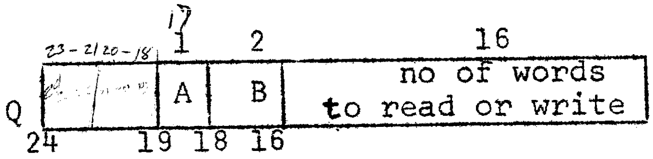
5 = MAGNETIC TAPE
6 = TELETYPE
7 = PLOTTER
8 = NULL (ABSORBS ALL OUTPUT)

f.g	Mnemonic op code	Address field
74	READ,I	M,B



c = 0 start in same bank
= 1 start in other bank

76	WRITE,I	M,B



A = 0 BCD MODE
= 1 BINARY
6.+ 17
00400000 B
B = 00

AFTER READ OR WRITE STATUS IN A
AFTER READ Q IS SET EQUAL TO ORIGINAL VALUE IN LOWER 16 BITS - NO OF WORDS IN THE RECORD

f.g	Mnemonic op code	Address field	Name
71	XREQ,I	M,B	Executive Request see below.

EXECUTIVE REQUEST instruction.

This is a simulated instruction (under OS-3) which is used to equip logical units, save files, call library programs, etc. The registers, etc, are used as follows:

AQ Name of file or library program, in BCD charactetr codes. Is left unchanged except in case of EQUIP request.

B1 Code for action desired. After operation, contains an error code (0 means no error).

effective address: logical unit number (0 to 99) or memory page number (0 to 37_8), or not used.

CODE in B 1	Request	Action and error conditions
0	DELETE	Delete file name (in AQ) from file directory. File must be equipped as the specified logical unit. ERROR 1 unit is not equipped 2 file is protected 3 unit is not a file 4 name in AQ does not agree with name of file 5 not enough scratch file space for file
1	SAVE	Save specified logical unit under name in AQ (put name in file directory). ERROR 1 unit is not equipped 2 there already exists a file with the name given in AQ 3 unit is already a saved file 4 unit is not a file 5 not enough saved file space for file 6 name is illegal (such as FILE, PUN, etc.)
2	UNEQUIP	Unequip the specified logical unit. AQ is ignored. ERROR 1 unit is not equipped 2 unit is a file which is protected and not saved
3	EQUIP	If A≠0, equip specified unit as the saved file whose name is in AQ, or as the hardware unit whose name is in AQ. See list below. If A = 0, equip specified unit as equivalent to the unit number specified in Q(0 to 100). If no error occurs, A contains the status of the unit after equipping it. HARDWARE NAMES (in AQ): FILE create an empty scratch file LP line printer PR same as LP PUN card punch PLOT plotter

NULL null (device absorbs and discards all outputs to it)
MT (in A) magnetic tape
no. (in Q) reel number in Q.

ERROR

1 unit is already equipped
2 there is no saved file with the name given in AQ (or, perhaps, filer busy)
3 unit number (in Q) is not equipped
4 saved file with name in AQ is busy (it is not protected and some other user has it equipped).

4 RFP Remove file protection from specified unit. If is saved, its name must be given in AQ. If it is not saved, AQ is ignored.

ERROR

1 unit is not equipped
2 file is busy (some other user is equipped to it, too).
3 name in AQ does not agree with the name of the file
4 file is public (first character of name is an asterisk) and it belongs to some other user.

5 FP Protect the filw which is equipped to the specified unit. AQ is ignored.

ERROR

1 unit is not equipped
2 unit is not a file

6 unused

7 ZEROPAGE Effective address (0 to 37_8) specifies a page in memory, which is set to all zero. A page is 2048 words, and they are numbered from the low end of memory upward. Thus, page 0 extends from addres 000000 to 003777, page 13 extends from 054000 to 057777, etc. AQ is ignored.

8 LIBCALL Call the library program whose name is in AQ. This involves copying the program into the pages

of memory which it is to occupy, and transfering control to it. If the library program needs a parameter string, this must have been defined previously by storing ASC II characters, using the ACI instruction. The effective address is ignored.

ERROR

If there is no library program by the name given in AQ, nothing is done and the computer executes the next instruction after the XREQ.

CIPHER & CDC 3300 Character Codes

Clipped ASCII	Line printer	Card code	Key punch	Tele type	ASCII	BCD
00	↓	11,6,8		@	300	56
01	A	12,1	A	A	301	21
02	B	12,2	B	B	302	22
03	C	12,3	C	C	303	23
04	D	12,4	D	D	304	24
05	E	12,5	E	E	305	25
06	F	12,6	F	F	306	26
07	G	12,7	G	G	307	27
10	H	12,8	H	H	310	30
11	I	12,9	I	I	311	31
12	J	11,1	J	J	312	41
13	K	11,2	K	K	313	42
14	L	11,3	L	L	314	43
15	M	11,4	M	M	315	44
16	N	11,5	N	N	316	45
17	O	11,6	O	O	317	46
20	P	11,7	P	P	320	47
21	Q	11,8	Q	Q	321	50
22	R	11,9	R	R	322	51
23	S	0,2	S	S	323	62
24	T	0,3	T	T	324	63
25	U	0,4	U	U	325	64
26	V	0,5	V	V	326	65
27	W	0,6	W	W	327	66
30	X	0,7	X	X	330	67
31	Y	0,8	Y	Y	331	70
32	Z	0,9	Z	Z	332	71
33	[	7,8		[	333	17
34	↱	0,5,8		\	334	75
35	]	0,2,8		]	335	72
36	↑	11,5,8		↑	336	55
37	≡	0,6,8		←	337	76
40	space	blank	space	space	240	60
41	∨	11,0		!	241	52
42	¬	12,6,8		"	242	36
43	≥	12,5,8		#	243	35
44	$	11,3,8	$	$	244	53
45	%	6,8		%	245	16
46	≤	5,8		&	246	15
47	≠	4,8	'	'	247	14
50	(	0,4,8	(	(	250	74
51	)	12,4,8	)	)	251	34
52	*	11,4,8	*	*	252	54
53	+	12	+	+	253	20
54	,	0,3,8	,	,	254	73
55	-	11	-	-	255	40
56	.	12,3,8	.	.	256	33
57	/	0,1	/	/	257	61
60	0	0	0	0	260	00
61	1	1	1	1	261	01
62	2	2	2	2	262	02
63	3	3	3	3	263	03
64	4	4	4	4	264	04
65	5	5	5	5	265	05
66	6	6	6	6	266	06
67	7	7	7	7	267	07
70	8	8	8	8	270	10
71	9	9	9	9	271	11
72	:	2,8		:	272	12
73	;	12,7,8		;	273	37
74	<	12,0		<	274	32
75	=	3,8	=	=	275	13
76	>	11,7,8		>	276	57
77	∧	0,7,8		?	277	77