DDP-24 CHARACTER CODES

OCTAL CODE	TYPEWRITER L/C	TYPEWRITER U/C	PAPER TAPE 8	7	6	P	4	S	3	2	1
00	Ø	b				○		•			
01	1							•			○
02	2							•		○	
03	3					○		•		○	○
04	4	:						•	○		
05	5	@				○		•	○		○
06	6	√				○		•	○	○	○
07	7	>						•	○	○	○
10	8						○	•			
11	9					○	○	•			○
13	#	-					○	•		○	○
20	*	¢			○			•			
21	/				○	○		•		○	
22	S				○	○		•		○	
23	T				○			•		○	○
24	U	=			○	○		•	○		
25	V	%			○			•	○		○
26	W	"			○			•	○	○	
27	X	'			○	○		•	○	○	○
30	Y				○	○	○	•			
31	Z				○		○	•			○
33	,				○	○	○	•	○	○	
36	tab				○	○	○	•	○	○	
40	-	-		○				•			
41	J			○		○		•			○
42	K			○		○		•		○	
43	L			○				•		○	○
44	M	)		○		○		•	○		
45	N	*		○				•	○		○
46	O	Δ		○				•	○	○	
47	P	;		○		○		•	○	○	○
50	Q			○		○	○	•			
51	R			○			○	•			○
53	S			○		○	○	•		○	○
54	backspace*			○			○	•	○			**
56	space			○		○	○	•	○	○	
60	&	&			○	○	○	•			
61	A				○	○		•			○
62	B				○	○		•		○	
63	C			○	○	○		•		○	○
64	D	(		○	○			•	○		
65	E	□		○	○	○		•	○		○
66	F	/		○	○	○		•	○	○	
67	G	<		○	○			•	○	○	○
70	H			○	○		○	•			
71	I			○	○	○	○	•			○
73	.	∨		○	○		○	•		○	○
74	lower shift			○	○	○	○	•	○		
75	upper shift			○	○		○	•	○		○
76	car. return			○	○		○	•	○	○	
77	line feed			○	○	○	○	•	○	○	○
stop	backspace*		○					•			

**This code can only be punched when the computer is punching the tape under program control.

*If the backspace key is depressed either on-line or off-line, a stop code will be generated; if the typewriter receives the code (54_8) either on-line or off-line, the carriage will be backspaced.

NOTE: 1. The assembly program (DAP) will recognize the & (60_8) as a +
2. The assembly program (DAP) will recognize the line feed (77_8) as a delete

COMMONLY USED OCP ADDRESS CODES FOR STANDARD EQUIPMENT

00000	Enable both character channels (Input and Output)
00001	Enable input word channel
00002	Enable output word channel
01000	Punch stop code
02000	Typewriter input select (keyboard enabled)
02010	Typewriter output select (keyboard inhibited)
02070	Disconnect standard I/O devices
02100	Paper tape reader select
02200	Paper tape punch select

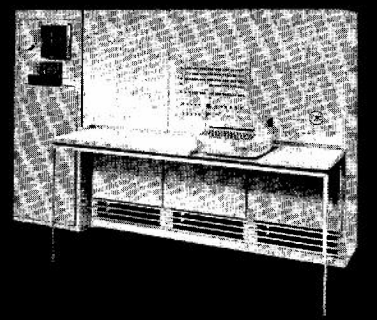
COMMONLY USED SKS ADDRESS CODES FOR STANDARD EQUIPMENT

00001	Sense switch #1
00002	Sense switch #2
00004	Sense switch #3
00010	Sense switch #4
00020	Sense switch #5
00040	Sense switch #6
00100	Parity error
00200	Improper divide indicator
00400	Overflow indicator
01000	Stop code
11000	Word output channel ready
12000	Word input channel ready
14000	Character I/O channel ready

SUMMARY OF DAP PSEUDO-OPERATIONS

PSEUDO-OP	DESCRIPTION	SYMBOL IN LOC	ADDRESS SYMBOL	ADDRESS OCTAL	ADDRESS DECIMAL	ADDRESS COMPOUND
ABS	Absolute					
BCI	Binary Coded Information	X				
BES	Block Ending Symbol	X	X	X	X	X
BSS	Block Starting Symbol	X	X	X	X	X
CALL	Subroutine Linkage	X	X			
DEC	Decimal constant	X			X	
ENBI	ITC with bit 10 = 1	X	X	X	X	X
END	End of program					
ENDM	End of macro					
EQU	Equals	X	X	X	X	X
INAM	INA with bit 10 = 1	X	X	X	X	X
INHI	ITC with bit 10 = 0	X	X	X	X	X
LIST	Produce Listing					
MAC	Macro definition	X	X	X	X	
MOR	More tape					
MZE	Minus Zero	X	X	X	X	X
NLST	Do Not Produce Listing					
NTRY	Subroutine Entry		X			
OCT	Octal Constant	X		X		
ORG	Program Origin	X	X	X	X	X
OTAM	OTA with bit 10 = 1	X	X	X	X	X
PZE	Plus Zero	X	X	X	X	X
REL	Relocatable					
RTRN	Subroutine Exit	X	X			X
SWT	Sense Switch Test	X	X	X	X	X

DDP-24 digital data processor



PROGRAMMER'S REFERENCE CARD

3C

OP-CODE OCTAL	OP-CODE MNEMONIC	INSTRUCTION	EXECUTION TIME (μs)
00	HLT	Halt	5
02	XEC	Execute	5+
03	STB	Store B	10
04	STC	Store Command Portion of A	10
05	STA	Store A	10
06	STD	Store Address Portion of A	10
07	INM	Input to Memory	10
10	ADD	Add	10
11	SUB	Subtract	10
12	SKG	Skip if A Greater	10-12
13	SKN	Skip if A Not Equal	10-12
15	ANA	AND to A	10
16	ORA	OR to A	10
17	ERA	Exclusive OR to A	10
20	ADM	Add Magnitude	10
21	SBM	Subtract Magnitude	10
22	OTM	Output from Memory	10
23	LDB	Load B	10
24	LDA	Load A	10
25	JRT	Jump Return	10
27	JST	Jump and Store Location	10
30	SMP	Step Multiple Precision	10
31	FMB	Fill Memory Block	variable
32	DMB	Dump Memory Block	variable
34	MPY	Multiply	31
35	DIV	Divide	33
36*	BCD*	BCD to Binary Conversion*	33
37*	BIN*	Binary to BCD Conversion*	33
40	ARS	A Right Shift	5+n
41	ALS	A Left Shift	5+n
42	LRR	Long Right Rotate	5+n
43	LLR	Long Left Rotate	5+n
44	LRS	Long Right Shift	5+n
45	LLS	Long Left Shift	5+n
46	NRM	Normalize	5+n
47	LGL	Logical Left Shift	5+n
50	OTA	Output from A	5
51	ITC	Interrupt Control	5
52	INA	Input to A	5
53	OCP	Output Control Pulse	5
54	ADX	Add to Index	5
55	TAB	Transfer A to B	5
56	LDX	Load Index	5
57	IAB	Interchange A and B	10
60	CRA	Clear A	5
61	SKS	Skip if Sense Line Not Set	5
62	RND	Round A	6
63	TAX	Transfer A to Index	5
64	SCR	Scale Right	5+n
65	SCL	Scale Left	5+n
66	STX	Store Index	10
67	IRX	Increment, Replace, and Load Index	14
70	JPL	Jump if A Plus	6
71	JZE	Jump if A Zero	5
72	JIX	Jump on Index	5
73	JOF	Jump on Overflow	5
74	JMP	Unconditional Jump	5
75	JXI	Jump on Index Incremented	7
77	NOP	No Operation	5

*OPTIONAL

	OP-CODE MNEMONIC	OP-CODE OCTAL	FUNCTION	X	I	O'F
LOAD & STORE	CRA	60	0 → (A)			
	IAB	57	(A) ↔ (B)			
	LDA	24	(EA) → (A)	P	P	
	LDB	23	(EA) → (B)	P	P	
	STA	05	(A) → (EA)	P	P	
	STB	03	(B) → (EA)	P	P	
	STC	04	$(A)_{1-9}$ → $(EA)_{1-9}$	P	P	
	STD	06	$(A)_{10-24}$ → $(EA)_{10-24}$	P	P	
	TAB	55	(A) → (B)			
ARITHMETIC	ADD	10	(A) + (EA) → (A)	P	P	P
	ADM	20	(A) + |(EA)| → (A)	P	P	P
	BCD*	36	(EA) BCD → (A) Binary	P	P	
	BIN*	37	(EA) Binary → (B) BCD	P	P	P
	DIV	35	(A, B) / (EA) → (quotient to B, remainder to A)	P	P	P
	MPY	34	(B) X (EA) → (A, B)	P	P	
	RND	62	(A) + 1 → (A), if $(B)_2$=1			P
	SBM	21	(A) - |(EA)| → (A)	P	P	P
	SMP	30	See Manual	P	P	P
	SUB	11	(A) - (EA) → (A)	P	P	P
LOGICAL	ANA	15	(A) AND (EA) → (A)	P	P	
	ERA	17	(A) Exclusive OR (EA) → (A)	P	P	
	ORA	16	(A) OR (EA) → (A)	P	P	
SHIFT	ALS**	41	Shift $(A)_{2-24}$ left, positions specified by $(EA)_{19-24}$	P	P	
	ARS**	40	Shift $(A)_{2-24}$ right, positions specified by $(EA)_{19-24}$	P	P	
	LGL**	47	Shift $(A)_{1-24}$ left, positions specified by $(EA)_{19-24}$	P	P	
	LLR**	43	Rotate $(A, B)_{1-24}$ left, positions specified by $(EA)_{19-24}$	P	P	
	LLS**	45	Shift $(A, B)_{2-24}$ left, positions specified by $(EA)_{19-24}$	P	P	
	LRR*	42	Rotate $(A,B)_{1-24}$ right, positions specified by $(EA)_{19-24}$	P	P	
	LRS**	44	Shift $(A, B)_{2-24}$ right, positions specified by $(EA)_{19-24}$	P	P	
	NRM	46	Shift $(A, B)_{2-24}$ left until $(A)_2$=1, See Manual	P		
	SCL**	65	Shift $(A, B)_{2-24}$ left and decrement index register, positions specified by $(EA)_{19-24}$	R	P	
	SCR**	64	Shift $(A, B)_{2-24}$ right and increment index register, positions specified by $(EA)_{19-24}$	R	P	

X = Indexable

I = Indirectly Addressable

O'F = Overflow possible

* Optional

** If indirect address not specified (I = 0), address portion of instruction is effective operand

P = Possible

R = Required

(P) = Improper divide possible

	OP-CODE MNEMONIC	OP-CODE OCTAL	FUNCTION	X	I	O'F
JUMP	JMP	74	Jump to EA	P	P	
	JOF	73	Jump to EA, if overflow indicator set; reset overflow indicator	P	P	
	JPL	70	Jump to EA, if sign of (A) is +	P	P	
	JRT	25	Jump to location specified by $(EA)_{10-24}$ restore interrupt	P	P	
	JST	27	Jump to EA + 1 and store location in $(EA)_{10-24}$	P	P	
	JZE	71	Jump to EA, if (A) = ±0	P	P	
	SKG	12	Skip next instruction, if (A) > (EA)	P	P	
	SKN	13	Skip next instruction, if (A) ≠ (EA)	P	P	
INDEX	ADX**	54	(X) + $(EA)_{10-24}$ → (X)	R	P	
	IRX	67	$(EA)_{10-24}$ + 1 → $(EA)_{10-24}$ and (X), See Manual	P	P	
	JIX	72	Jump to EA, if (X) ≠ 0	R	P	
	JXI	75	(X) + 1 → (X), Jump to EA, if resultant (X) ≠ 0	R	P	
	LDX**	56	$(EA)_{10-24}$ → (X)	R	P	
	STX	66	(X) → $(EA)_{10-24}$	R	P	
	TAX	63	$(A)_{10-24}$ → (X)	R		
INPUT/OUTPUT	DMB	32	Dump memory starting at EA, See Manual	R	P	
	FMB	31	Load memory starting at EA, See Manual	R	P	
	INA**	52	Input → (A), according to mask in (EA), See Manual	P	P	
	INM	07	Input → (EA)	P	P	
	ITC**	51	Inhibit or enable interrupt, according to mask in (EA)	P	P	
	OCP**	53	Select I/O, according to mask in (EA)	P	P	
	OTA**	50	(A) → Output, according to mask in (EA), See Manual	P	P	
	OTM	22	(EA) → Output	P	P	
CONTROL	SKS**	61	Skip next instruction, if sense line not set; sense line specified by (EA)	P	P	
	HLT	00	Stop computer operation until start button pressed			
	NOP	77	Perform no operation			
	XEC	02	Execute instruction at EA	P	P	

**If indirect address not specified (I = 0), address portion of instruction is effective operand.

WORD FORMATS