HEATHKIT H-8 DIGITAL COMPUTER

Op Code Table

	0	1	2	3	4	5	6	7
00	NOP	** LXI B ++	STAX B	INX B	* INR B	* DCR B	MVI B +	** RLC
01	–	DAD B	LDAX B	DCX B	* INR C	* DCR C	MVI C +	** RRC
02	–	** LXI D ++	STAX D	INX D	* INR D	* DCR D	MVI D +	** RAL
03	—	DAD D	LDAX D	DCX D	* INR E	* DCR E	MVI E +	** RAR
04	—	** LXI H ++	SHLD ++	INX H	* INR H	* DCR H	MVI H +	DAA
05	—	DAD H	LHLD ++	DCX H	* INR L	* DCR L	MVI L +	CMA
06	—	** LXI SP++	STA ++	INX SP	* INR M	* DCR M	MVI M +	** STC
07	—	DAD SP	LDA ++	DCX SP	* INR A	* DCR A	MVI A +	** CMC

	0	1	2	3	4	5	6	7
10	MOV B,B	MOV B,C	MOV B,D	MOV B,E	MOV B,H	MOV B,L	MOV B,M	MOV B,A
11	MOV C,B	MOV C,C	MOV C,D	MOV C,E	MOV C,H	MOV C,L	MOV C,M	MOV C,A
12	MOV D,B	MOV D,C	MOV D,D	MOV D,E	MOV D,H	MOV D,L	MOV D,M	MOV D,A
13	MOV E,B	MOV E,C	MOV E,D	MOV E,E	MOV E,H	MOV E,L	MOV E,M	MOV E,A
14	MOV H,B	MOV H,C	MOV H,D	MOV H,E	MOV H,H	MOV H,L	MOV H.M	MOV H,A
15	MOV L,B	MOV L,C	MOV L,D	MOV L,E	MOV L,H	MOV L,L	MOV L,M	MOV L,A
16	MOV M,B	MOV M,C	MOV M,D	MOV M,E	MOV M,H	MOV M,L	HLT	MOV M,A
17	MOV A,B	MOV A,C	MOV A,D	MOV A,E	MOV A,H	MOV A,L	MOV A,M	MOV A,A

	0	1	2	3	4	5	6	7
20	ADD B	ADD C	ADD D	ADD E	ADD H	ADD L	ADD M	ADD A
21	ADC B	ADC C	ADC D	ADC E	ADC H	ADC L	ADC M	ADC A
22	SUB B	SUB C	SUB D	SUB E	SUB H	SUB L	SUB M	SUB A
23	SBB B	SBB C	SBB D	SBB E	SBB H	SBB L	SBB M	SBB A
24	ANA B	ANA C	ANA D	ANA E	ANA H	ANA L	ANA M	ANA A
25	XRA B	XRA C	XRA D	XRA E	XRA H	XRA L	XRA M	XRA A
26	ORA B	ORA C	ORA D	ORA E	ORA H	ORA L	ORA M	ORA A
27	CMP B	CMP C	CMP D	CMP E	CMP H	CMP L	CMP M	CMP A

	0	1	2	3	4	5	6	7
30	RNZ	POP B	JNZ ++	JMP ++	CNZ ++	PUSH B	ADI +	RST 0
31	RZ	RET	JZ ++	—	CZ ++	CALL ++	ACI +	RST 1
32	RNC	POP D	JNC ++	OUT +	CNC ++	PUSH D	SUI +	RST 2
33	RC	—	JC ++	IN +	CC ++	—	SBI +	RST 3
34	RPO	POP H	JPO ++	XTHL	CPO ++	PUSH H	ANI +	RST 4
35	RPE	PCHL	JPE ++	XCHG	CPE ++	—	XRI +	RST 5
36	RP	POP PSW	JP ++	DI	CP ++	PUSH PSW	ORI +	RST 6
37	RM	SPHL	JM ++	EI	CM ++	—	CPI +	RST 7

Red Op Code indicates all flags affected

Black Op Code Indicates No Flags Affected

(each + indicates an extra instruction byte)

All Flags Except Carry Affected

****Only Carry Flags Affected***

MEMORY BLOCKS

DEC	OCTAL	DEC	OCTAL
8K	040	40K	240
16K	100	48K	300
24K	140	56K	340
32K	200		

FLAG REGISTER

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
S	Z	0	AC	0	P	1	C

D_7 D_6	D_5 D_4 D_3	D_2 D_1 D_0
0 = $\bar{S}, \bar{Z}$	0 = $\overline{AC}$	2 = $\bar{P}, \bar{C}$
1 = Z	2 = AC	3 = C
2 = S		6 = P
3 = S, Z		7 = P, C

AC = AUX CARRY
S = SIGN
Z = ZERO
C = CARRY
P = PARITY

INSTRUCTION SET

Mnemonic	Description
ACI	Add immediate to A with carry
ADC M	Add memory to A with carry
ADC r	Add register to A with carry
ADD M	Add memory to A
ADD r	Add register to A
ADI	Add immediate to A
ANA M	And memory with A
ANA r	And register with A
ANI	And immediate with A
CALL	Call unconditional
CC	Call on carry
CM	Call on minus
CMA	Complement A
CMC	Complement carry
CMP M	Compare memory with A
CMP r	Compare register with A
CNC	Call on no carry
CNZ	Call on no zero
CP	Call on positive
CPE	Call on parity even
CPI	Compare immediate with A
CPO	Call on parity odd
CZ	Call on zero
DAA	Decimal adjust A
DAD B	Add B & C to H & L
DAD D	Add D & E to H & L
DAD H	Add H & L to H & L
DAD SP	Add stack pointer to H & L
DCR M	Decrement memory
DCR r	Decrement register
DCX B	Decrement B & C
DCX D	Decrement D & E
DCX H	Decrement H & L
DCX SP	Decrement stack pointer
DI	Disable Interrupt
EI	Enable Interrupts
HLT	Halt
IN	Input
INR M	Increment memory
INR r	Increment register
INX B	Increment B & C registers
INX D	Increment D & E registers
INX H	Increment H & L registers
INX SP	Increment stack pointer
JC	Jump on carry
JM	Jump on minus
JMP	Jump unconditional
JNC	Jump on no carry
JNZ	Jump on no zero
JP	Jump on positive
JPE	Jump on parity even
JPO	Jump on parity odd
JZ	Jump on zero
LDA	Load A direct
LDAX B	Load A indirect
LDAX D	Load A indirect
LHLD	Load H & L direct
LXI B	Load immediate register Pair B & C
LXI D	Load immediate register Pair D & E
LXI H	Load immediate register Pair H & L
LXI SP	Load immediate stack pointer

Mnemonic	Description
MVI M	Move immediate memory
MVI r	Move immediate register
MOV M, r	Move register to memory
MOV r, M	Move memory to register
MOV r1, r2	Move register to register
NOP	No-operation
ORA M	Or memory with A
ORA r	Or register with A
ORI	Or immediate with A
OUT	Output
PCHL	H & L to program counter
POP B	Pop register pair B & C off stack
POP D	Pop register pair D & E off stack
POP H	Pop register pair H & L off stack
POP PSW	Pop A and Flags off stack
PUSH B	Push register Pair B & C on stack
PUSH D	Push register Pair D & E on stack
PUSH H	Push register Pair H & L on stack
PUSH PSW	Push A and Flags on stack
RAL	Rotate A left through carry
RAR	Rotate A right through carry
RC	Return on carry
RET	Return
RLC	Rotate A left
RM	Return on minus
RNC	Return on no carry
RNZ	Return on no zero
RP	Return on positive
RPE	Return on parity even
RPO	Return on parity odd
RRC	Rotate A right
RST	Restart
RZ	Return on zero
SBB M	Subtract memory from A with borrow
SBB r	Subtract register from A with borrow
SBI	Subtract immediate from A with borrow
SHLD	Store H & L direct
SPHL	H & L to stack pointer
STA	Store A direct
STAX B	Store A indirect
STAX D	Store A indirect
STC	Set carry
SUB M	Subtract memory from A
SUB r	Subtract register from A
SUI	Subtract immediate from A
XCHG	Exchange D & E, H & L Registers
XRA M	Exclusive Or memory with A
XRA r	Exclusive Or register with A
XRI	Exclusive Or immediate with A
XTHL	Exchange top of stack, H & L