i8080 Instruction set

F E I S T U

Data Transfer Group

Code	Comment	Symbolic	FLAGS S	Z	A	P	C	Bytov	Cyklov
MOV r1, r2	Move the content of the one register to another	r1 ← r2	-	-	-	-	-	1	5
MOV r, M	Move the content of memory to the register	R ← [HL]	-	-	-	-	-	1	7
MOV M, r	Move the content of register to the memory	[HL] ← r	-	-	-	-	-	1	7
MVI r, data8	Move immediate data to register	r ← data8	-	-	-	-	-	2	7
MVI M, data8	Move immediate data to memory	[HL] ← data8	-	-	-	-	-	2	7
LXI rp, data16	Load register pair with immediate deta	rp ← data16, rh ← 8 LSB	-	-	-	-	-	3	10
LDA addr16	Load Accumulator from direct address	A ← [addr]	-	-	-	-	-	3	13
STA addr16	Store accumulator directly to address	[addr] ← A	-	-	-	-	-	3	13

Arithmetic Group

Code	Comment	Symbolic	FLAGS S	Z	A	P	C	Bytov	Cyklov
ADD r	Add register to accumulator	[A] ← [A] + [r]	S	Z	A	P	C	1	4
ADD M	Add memory to accumulator	[A] ← [A] + [[H-L]]	S	Z	A	P	C	1	7
ADC r	Add register with carry to accumulator	[A] ← [A] + [r] + [CS]	S	Z	A	P	C	1	4
ADC M	Add memory with carry to accumulator	[A] ← [A] + [[H-L]] [CS]	S	Z	A	P	C	1	7
SUB r	Subtract register from accumulator	[A] ← [A] – [r]	S	Z	A	P	C	1	4
SUB M	Subtract memory from accumulator	[A] ← [A] – [[H-L]]	S	Z	A	P	C	1	7
INR r	Increment register content	[r] ← [r] +1	S	Z	A	P		1	5
INR M	Increment memory content	[[H-L]] ← [[H-L]] + 1	S	Z	A	P		1	10
DCR r	Decrement register content	[r] ← [r] – 1	S	Z	A	P		1	5
DCR M	Decrement memory content	[[H-L]] ← [[H-L]] – 1	S	Z	A	P		1	10
INX rp	Increment register pair	[rp] ← [rp] – 1						1	5
DCX rp	Decrement register pair	[rp] ← [rp] -1						1	5

Logical Group

Code	Comment	Symbolic	FLAGS S	Z	A	P	C	Bytov	Cyklov
ANA r	AND register with accumulator	[A] ← [A] ^ [r]	S	Z	A	P	C	1	4
ORA r	OR register with accumulator	[A] ← [A] v [r]	S	Z	A	P	C	1	4
XRA r	EX–Or register with accumulator	[A] ← [A] v [r]	S	Z	A	P	C	1	4
CMA	Complement the accumulator	[A] ← [A]						1	4
CMP r	Compare register with accumulator	[A] – [r]	S	Z	A	P	C	1	7
RLC	Rotate accumulator left	[An+1] ← [An], [A0] ← [A7],[CS] ← [A7]					C	1	4
RRC	Rotate accumulator right	[A7] ← [A0], [CS] ← [A0], [An] ← [An+1]					C	1	4
RAL	Rotate accumulator left through carry	[An+1] ← [An], [CS] ← [A7], [A0] ← [CS]					C	1	4
RAR	Rotate accumulator right through carry	[An] ← [An+1], [CS] ← [A0], [A7] ← [CS]					C	1	4

Branch Group

			FLAGS						
JMP addr16	Unconditional jump to the specified address	[PC] ← addr16						3	10
JZ addr16	Jump if the result is zero	[PC] ← addr16 / [PC] ← [PC]+3						3	10
JNZ addr16	Jump if the result is not zero	[PC] ← addr16 / [PC] ← [PC]+3						3	10
JC addr16	Jump if there is a carry	[PC] ← addr16 / [PC] ← [PC]+3						3	10
JNC addr16	Jump if there is no carry	[PC] ← addr16 / [PC] ← [PC]+3						3	10

Stack, I/O and Machine Control Group

			FLAGS						
IN addr8	Input to accumulator from I/O port	[A] ← [Port]	-	-	-	-	-	2	10
OUT addr8	Output from accumulator to I/O port	[Port] ← [A]	-	-	-	-	-	2	10
HLT	Halt processor		-	-	-	-	-	1	7
NOP	No Operation		-	-	-	-	-	1	4

Registers

15 8	7 0	
A (accumulator)	F (flags)	← PSW
B	C	← B
D	E	← D
H	L	← H

15 0
SP (stack pointer)
PC (program counter)

Flag register (F) bits:

FLAGS 7	6	5	4	3	2	1	0
S	Z	0	A	0	P	1	C

S - Sign Flag
Z - Zero Flag
0 - Not used, always zero
A - also called AC, Auxiliary Carry Flag
0 - Not used, always zero
P - Parity Flag
1 - Not used, always one
C - Carry Flag