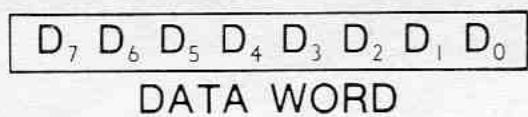


8080 MICROPROCESSOR

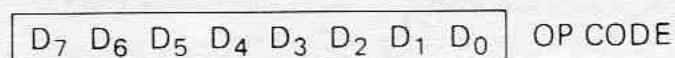
3-2. DATA AND INSTRUCTION FORMATS

Data in the 8080 is stored in the form of 8-bit binary integers. All data transfers to the system data bus will be in the same format.

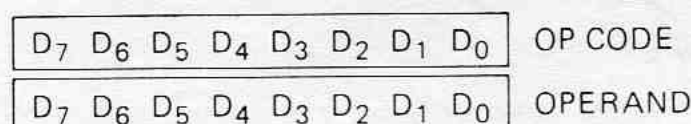


The program instructions may be one, two, or three bytes in length. Multiple byte instructions must be stored in successive words in program memory. The instruction formats then depend on the particular operation executed.

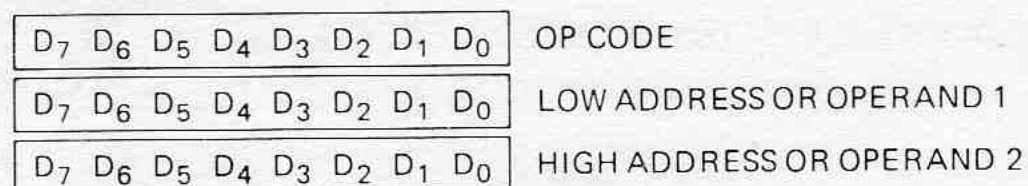
One Byte Instructions



Two Byte Instructions



Three Byte Instructions



For the 8080 a logic "1" is defined as a high level and a logic "0" is defined as a low level.

TYPICAL INSTRUCTIONS

Register to register, memory reference, arithmetic or logical, rotate return, push, pop, enable or disable Interrupt instructions

Immediate mode or I/O instructions

Jump, call or direct load and store instructions

3-3. INSTRUCTION SET

Summary of Processor Instructions

Mnemonic	Description	Instruction Code ⁽¹⁾								Clock ⁽²⁾ Cycles
		D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀	
MOV _{r1,r2}	Move register to register	0	1	0	0	0	S	S	S	5
MOV _{M,r}	Move register to memory	0	1	1	1	0	S	S	S	7
MOV _{r,M}	Move memory to register	0	1	0	0	0	1	1	0	7
HLT	Halt	0	1	1	1	0	1	1	0	7
MVI _r	Move immediate register	0	0	0	0	0	1	1	0	7
MVI _M	Move immediate memory	0	0	1	1	0	1	1	0	10
INR _r	Increment register	0	0	0	0	0	1	0	0	5
DCR _r	Decrement register	0	0	0	0	0	1	0	1	5
INR _M	Increment memory	0	0	1	1	0	1	0	0	10
DCR _M	Decrement memory	0	0	1	1	0	1	0	1	10
ADD _r	Add register to A	1	0	0	0	0	S	S	S	4
ADC _r	Add register to A with carry	1	0	0	0	1	S	S	S	4
SUB _r	Subtract register from A	1	0	0	1	0	S	S	S	4
SBB _r	Subtract register from A with borrow	1	0	0	1	1	S	S	S	4
ANA _r	And register with A	1	0	1	0	0	S	S	S	4
XRA _r	Exclusive Or register with A	1	0	1	0	1	S	S	S	4
ORA _r	Or register with A	1	0	1	1	0	S	S	S	4
CMP _r	Compare register with A	1	0	1	1	1	S	S	S	4
ADD _M	Add memory to A	1	0	0	0	0	1	1	0	7
ADC _M	Add memory to A with carry	1	0	0	0	1	1	1	0	7
SUB _M	Subtract memory from A	1	0	0	1	0	1	1	0	7
SBB _M	Subtract memory from A with borrow	1	0	0	1	1	1	1	0	7
ANA _M	And memory with A	1	0	1	0	0	1	1	0	7
XRA _M	Exclusive Or memory with A	1	0	1	0	1	1	1	0	7
ORA _M	Or memory with A	1	0	1	1	0	1	1	0	7
CMP _M	Compare memory with A	1	0	1	1	1	1	1	0	7
ADI	Add immediate to A	1	1	0	0	0	1	1	0	7
ACI	Add immediate to A with carry	1	1	0	0	1	1	1	0	7
SUI	Subtract immediate from A	1	1	0	1	0	1	1	0	7
SBI	Subtract immediate from A with borrow	1	1	0	1	1	1	1	0	7
ANI	And immediate with A	1	1	1	0	0	1	1	0	7
XRI	Exclusive Or immediate with A	1	1	1	0	1	1	1	0	7
ORI	Or immediate with A	1	1	1	1	0	1	1	0	7
CPI	Compare immediate with A	1	1	1	1	1	1	1	0	7
RLC	Rotate A left	0	0	0	0	0	1	1	1	4
RRC	Rotate A right	0	0	0	0	1	1	1	1	4
RAL	Rotate A left through carry	0	0	0	1	0	1	1	1	4
RAR	Rotate A right through carry	0	0	0	1	1	1	1	1	4
JMP	Jump unconditional	1	1	0	0	0	0	1	1	10
JC	Jump on carry	1	1	0	1	1	0	1	0	10
JNC	Jump on no carry	1	1	0	1	0	0	1	0	10
JZ	Jump on zero	1	1	0	0	1	0	1	0	10
JNZ	Jump on no zero	1	1	0	0	0	0	1	0	10
JP	Jump on positive	1	1	1	1	0	0	1	0	10
JM	Jump on minus	1	1	1	1	1	0	1	0	10
JPE	Jump on parity even	1	1	1	0	1	0	1	0	10
JPO	Jump on parity odd	1	1	1	0	0	0	1	0	10