

Intel 8080 Opcodes

For x=0				Opcode format:
z=0		NOP	No operation	
z=1	q=0	LXI, rp[p]	Load register pair immedate	<div> <div> <div>x x</div> <div>y y</div> <div>y</div> </div> <div> <div>z z</div> <div>z</div> </div> </div> <div> <div>p p</div> <div>q</div> </div>
	q=1	DAD rp[p]	Add register pair to HL	
z=2	q=0	STAX rp[p]	Store accumulator in location pointed by reg	<div> <div>x First octal digit</div> <div>y Second octal digit</div> <div>z Third octal digit</div> <div>p y rightshifted 1</div> <div>q y modulo 2</div> </div>
		SHLD a	Store HL at location	
		STA a	Store accumulator at location	
	q=1	LDAX rp[p]	Load accumulator with value from mem	
		LHLD a	Load HL from location	
		LDA a	Load accumulator with value from mem	
z=3	q=0	INX rp[p]	Increment register pair	<div> <div>Table r[]</div> <div> <div>0</div> <div>B</div> </div> </div>
	q=1	DCX rp[p]	Decrement register pair	
z=4		INR r[y]	Increment register	<div> <div>1</div> <div>C</div> </div>
z=5		DCR r[y]	Decrement register	
z=6		MVI r[y], #	Move immedate to register	<div> <div>2</div> <div>D</div> </div>
z=7	y=0	RLC	Rotate accumulator left with carry	
	y=1	RRC	Rotate accumulator right with carry	<div> <div>3</div> <div>E</div> </div>
	y=2	RAL	Rotate accumulator left	
	y=3	RAR	Rotate accumulator right	<div> <div>4</div> <div>H</div> </div>
	y=4	DAA	Decimal adjust accumulator	
	y=5	CMA	Complement accumulator	<div> <div>5</div> <div>L</div> </div>
	y=6	STC	Set carry bit	
	y=7	CMC	Complement carry bit	<div> <div>6</div> <div>(HL)</div> </div>
				<div> <div>Table rp[]</div> <div> <div>0</div> <div>BC</div> </div> </div>
				<div> <div>1</div> <div>DE</div> </div>
				<div> <div>2</div> <div>HL</div> </div>
				<div> <div>3</div> <div>SP</div> </div>
For x=1				
z=6	y=6	MOV r[y], r[z]	Move one register value to another	
		HALT	Halt cpu	
For x=2				
	y=0	ADD r[z]	Add register to accumulator	<div> <div>Table rp2[]</div> <div> <div>0</div> <div>BC</div> </div> </div>
	y=1	ADC r[z]	Add register to accumulator with carry	
	y=2	SUB r[z]	Subtract register from carry	<div> <div>1</div> <div>DE</div> </div>
	y=3	SBC r[z]	Subtract register from carry with borrow	
	y=4	ANA r[z]	Logical AND between accumulator and register	<div> <div>2</div> <div>HL</div> </div>
	y=5	XRA r[z]	Logical XOR between accumulator and register	
	y=6	ORA r[z]	Logical OR between accumulator and register	<div> <div>3</div> <div>AF</div> </div>
	y=7	CMP r[z]	Compare accumulator and register	
For x=3				
z=0		R cc[y]	Conditional RET	<div> <div>Table cc[]</div> <div> <div>0</div> <div>NZ</div> </div> </div>
z=1	q=0	POP rp2[p]	Pop value from stack to register pair	
	q=1	RET	Return from subroutine	<div> <div>1</div> <div>Z</div> </div>
		PCHL	Store program counter in HL	
		SPHL	Store stack pointer in HL	<div> <div>2</div> <div>NC</div> </div>
z=2		J cc[y]	Conditional JMP	
z=3	q=0	JMP a	Jump to address	<div> <div>3</div> <div>C</div> </div>
		OUT #		
		XTHL		<div> <div>4</div> <div>PO</div> </div>
		DI	Disable interrupts	
	q=1	IN #		<div> <div>5</div> <div>PE</div> </div>
		XCHG		
		EI	Enable interrupts	<div> <div>6</div> <div>P</div> </div>
z=4		C cc[y]	Conditional CALL	
z=5	q=0	PUSH rp2[p]	Push register pair value onto stack	<div> <div>7</div> <div>M</div> </div>
	q=1	CALL	Call subroutine	
z=6	y=0	ADI #	Add immedate to accumulator	
	y=1	ACI	Add immedate to accumulator with carry	
	y=2	SUI	Subtract immedate from accumulator	
	y=3	SBI	Subtract immedate from accumulator with borrow	
	y=4	ANI	Logical AND between accumulator and immedate	
	y=5	XRI	Logical XOR between accumulator and immedate	
	y=6	ORI	Logical OR between accumulator and immedate	
	y=7	CPI	Compare immedate with accumulator	
z=7		RST y*8	Jump to reset vector	