

8085 Instruction Mnemonic Meanings

Data Transfer Group

Instruction			Mnemonic Meaning				Flags			
							Zf	Cf	Pf	Sf
MOV	dreg, sreg	MOVE					.	.	.	.
MVI	reg, byte	MoVe Immediate					.	.	.	.
MOVX	drp, srp	MoVe eXtended-register (pseudo for high & low MOVs)					.	.	.	.
LXI	rp, word	Load eXtended-register Immediate					.	.	.	.
XCHG		eXCHanGe hl with de					.	.	.	.
LDA	addr	LoaD Accumulator direct					.	.	.	.
STA	addr	STore Accumulator direct					.	.	.	.
LDAX	B	LoaD Accumulator indirect via eXtended-register Bc					.	.	.	.
STAX	B	Store Accumulator indirect via eXtended-register Bc					.	.	.	.
LDAX	D	LoaD Accumulator indirect via eXtended-register De					.	.	.	.
STAX	D	Store Accumulator indirect via eXtended-register De					.	.	.	.
LHLD	addr	Load HL Direct					.	.	.	.
SHLD	addr	Store HL Direct					.	.	.	.
LHLI		Load HL Indirect via extended register de					.	.	.	.
SHLI		Store HL Indirect via extended register de					.	.	.	.

Arithmetic Group

Instruction			Mnemonic Meaning				Flags			
							Zf	Cf	Pf	Sf
ADD	reg	ADD					x	x	x	x
ADI	byte	ADd Immediate					x	x	x	x
ADC	reg	ADd with Carry					x	x	x	x
ACI	byte	Add with Carry Immediate					x	x	x	x
SUB	reg	SUBtract					x	x	x	x
SUI	byte	SUBtract Immediate					x	x	x	x
SBB	reg	SuBtract with Borrow					x	x	x	x
SBI	byte	Subtract with Borrow Immediate					x	x	x	x
DAA		Decimal Adjust Accumulator					x	x	x	x
INR	reg	INcrement Register					x	.	x	x
INX	rp	INcrement eXtended-register					.	.	.	.
DCR	reg	DeCrement Register					x	.	x	x
DCX	rp	DeCrement eXtended-register					.	.	.	.
DAD	rp	Dual-register ADD to hl					.	x	.	.
HLMBC		HL Minus BC					x	x	h	x
DEHL	byte	DE from HL plus byte					.	.	.	.
DESP	byte	DE from SP plus byte					.	.	.	.

Logical Group

Instruction			Mnemonic Meaning				Flags			
							Zf	Cf	Pf	Sf
CMP	reg	CoMPare					x	x	x	x
CPI	byte	ComPare Immediate					x	x	x	x
CMA		CoMPlement Accumulator					.	.	.	.
CMC		CoMPlement Carry					.	x	.	.
STC		SeT Carry					.	1	.	.
SHLR		Shift HL Right					.	x	.	.
ANA	reg	ANd Accumulator					x	0	x	x
ANI	byte	ANd Immediate					x	0	x	x
ORA	reg	OR Accumulator					x	0	x	x
ORI	byte	OR Immediate					x	0	x	x
XRA	reg	eXclusive oR Accumulator					x	0	x	x
XRI	byte	eXclusive oR Immediate					x	0	x	x
RAL		Rotate Accumulator Left through carry					.	x	.	.
RAR		Rotate Accumulator Right through carry					.	x	.	.
RLC		Rotate accumulator Left Circular					.	x	.	.
RRC		Rotate accumulator Right Circular					.	x	.	.
RDEL		Rotate DE Left through carry					.	x	.	.

Stack, Input/Output, & Machine Control Group

Instruction			Mnemonic Meaning				Flags			
							Zf	Cf	Pf	Sf
PUSH	rp	PUSH on stack					.	.	.	.
POP	rp	POP off stack					.	.	.	.
SPHL		Stack Pointer from HL					.	.	.	.
XTHL		eXchange Top of stack with HL					.	.	.	.
IN	port	INput from port					.	.	.	.
OUT	port	OUTput to port					.	.	.	.
DI		Disable Interrupts					.	.	.	.
EI		Enable Interrupts					.	.	.	.
RIM		Read Interrupt Mask					.	.	.	.
SIM		Set Interrupt Mask					.	.	.	.
NOP		No OPeration					.	.	.	.
HLT		HaLT					.	.	.	.

Branch Group

Instruction			Mnemonic Meaning				Flags			
							Zf	Cf	Pf	Sf
JMP	label	JuMP unconditional					.	.	.	.
JZ	label	Jump if Zero					.	.	.	.
JNZ	label	Jump if No Zero					.	.	.	.
JP	label	Jump if Positive					.	.	.	.
JM	label	Jump if Minus					.	.	.	.
JC	label	Jump if Carry					.	.	.	.
JNC	label	Jump if No Carry					.	.	.	.
JTM	label	Jump if True sign Minus					.	.	.	.
JTP	label	Jump if True sign Positive					.	.	.	.
JPE	label	Jump if Parity Even					.	.	.	.
JPO	label	Jump if Parity Odd					.	.	.	.
CALL	label	CALL unconditional					.	.	.	.
CZ	label	Call if Zero					.	.	.	.
CNZ	label	Call if No Zero					.	.	.	.
CP	label	Call if Positive					.	.	.	.
CM	label	Call if Minus					.	.	.	.
CC	label	Call if Carry					.	.	.	.
CNC	label	Call if No Carry					.	.	.	.
CPE	label	Call if Parity Even					.	.	.	.
CPO	label	Call if Parity Odd					.	.	.	.
RET		RETurn unconditional					.	.	.	.
RZ		Return if Zero					.	.	.	.
RNZ		Return if No Zero					.	.	.	.
RP		Return if Positive					.	.	.	.
RM		Return if Minus					.	.	.	.
RC		Return if Carry					.	.	.	.
RNC		Return if No Carry					.	.	.	.
RPE		Return if Parity Even					.	.	.	.
RPO		Return if Parity Odd					.	.	.	.
PCHL		Program Counter from HL					.	.	.	.
RST	n	ReSTart					.	.	.	.
RSTV		ReSTart if oVerflow					.	.	.	.

8085 Instruction Actions by Functional Group

Data Transfer Group

Instruction			Mnemonic Meaning				Flags			
							Zf	Cf	Pf	Sf
MOV	dreg, sreg		dreg<=sreg				.	.	.	.
MVI	reg, byte		reg<=byte				.	.	.	.
MVX	drp, srp		drp<=srp (pseudo for high & low MOVs)				.	.	.	.
LXI	rp, word		rp<=word				.	.	.	.
XCHG			HL<=DE while DE<=HL				.	.	.	.
LDA	addr		A<=b[addr]				.	.	.	.
STA	addr		b[addr]<=A				.	.	.	.
LDAX	B		A<=b[BC]				.	.	.	.
STAX	B		b[BC]<=A				.	.	.	.
LDAX	D		A<=b[DE]				.	.	.	.
STAX	D		b[DE]<=A				.	.	.	.
LHLD	addr		HL<=w[addr]				.	.	.	.
SHLD	addr		w[addr]<=HL				.	.	.	.
LHLI			HL<=w[DE]				.	.	.	.
SHLI			w[DE]<=HL				.	.	.	.

Arithmetic Group

Instruction			Mnemonic Meaning				Flags			
							Zf	Cf	Pf	Sf
ADD	reg		A<=A+reg				x	x	x	x
ADI	byte		A<=A+byte				x	x	x	x
ADC	reg		A<=A+reg+Cf				x	x	x	x
ACI	byte		A<=A+byte+Cf				x	x	x	x
SUB	reg		A<=A-reg				x	x	x	x
SUI	byte		A<=A-byte				x	x	x	x
SBB	reg		A<=A-reg-Cf				x	x	x	x
SBI	byte		A<=A-byte-Cf				x	x	x	x
DAA			in A3..A0 and A7..A4: if >9 then +6, carry to next				x	x	x	x
INR	reg		reg<=reg+1				x	.	x	x
INX	rp		rp<=rp+1				.	.	.	.
DCR	reg		reg<=reg-1				x	.	x	x
DCX	rp		rp<=rp-1				.	.	.	.
DAD	rp		HL<=HL+rp				.	x	.	.
HLMB			HL<=HL-BC				x	x	h	x
DEHL	byte		DE<=HL+byte				.	.	.	.
DESP	byte		DE<=SP+byte				.	.	.	.

Branch Group

Instruction			Mnemonic Meaning				Flags			
							Zf	Cf	Pf	Sf
JMP	label		PC<=label				.	.	.	.
JZ	label		if Zf=1 then PC<=label				.	.	.	.
JNZ	label		if Zf=0 then PC<=label				.	.	.	.
JP	label		if Sf=0 then PC<=label				.	.	.	.
JM	label		if Sf=1 then PC<=label				.	.	.	.
JC	label		if Cf=1 then PC<=label				.	.	.	.
JNC	label		if Cf=0 then PC<=label				.	.	.	.
JTM	label		if TSf=1 then PC<=label				.	.	.	.
JTP	label		if TSf=0 then PC<=label				.	.	.	.
JPE	label		if Pf=1 then PC<=label				.	.	.	.
JPO	label		if Pf=0 then PC<=label				.	.	.	.
CALL	label		SP<=SP-2, w[SP]<=PC+3, PC<=label				.	.	.	.
CZ	label		if Zf=1 then SP<=SP-2, w[SP]<=PC+3, PC<=label				.	.	.	.
CNZ	label		if Zf=0 then SP<=SP-2, w[SP]<=PC+3, PC<=label				.	.	.	.
CP	label		if Sf=0 then SP<=SP-2, w[SP]<=PC+3, PC<=label				.	.	.	.
CM	label		if Sf=1 then SP<=SP-2, w[SP]<=PC+3, PC<=label				.	.	.	.
CC	label		if Cf=1 then SP<=SP-2, w[SP]<=PC+3, PC<=label				.	.	.	.
CNC	label		if Cf=0 then SP<=SP-2, w[SP]<=PC+3, PC<=label				.	.	.	.
CPE	label		if Pf=1 then SP<=SP-2, w[SP]<=PC+3, PC<=label				.	.	.	.
CPO	label		if Pf=0 then SP<=SP-2, w[SP]<=PC+3, PC<=label				.	.	.	.
RET			PC<=w[SP], SP<=SP+2				.	.	.	.
RZ			if Zf=1 then PC<=w[SP], SP<=SP+2				.	.	.	.
RNZ			if Zf=0 then PC<=w[SP], SP<=SP+2				.	.	.	.
RP			if Sf=0 then PC<=w[SP], SP<=SP+2				.	.	.	.
RM			if Sf=1 then PC<=w[SP], SP<=SP+2				.	.	.	.
RC			if Cf=1 then PC<=w[SP], SP<=SP+2				.	.	.	.
RNC			if Cf=0 then PC<=w[SP], SP<=SP+2				.	.	.	.
RPE			if Pf=1 then PC<=w[SP], SP<=SP+2				.	.	.	.
RPO			if Pf=0 then PC<=w[SP], SP<=SP+2				.	.	.	.
PCHL			PC<=HL				.	.	.	.
RST	n		SP<=SP-2, w[SP]<=PC+1, PC<=n*8 where n is 0 to 7				.	.	.	.
RSTV			if OVf=1 then SP<=SP-2, w[SP]<=PC+1, PC<=8*8				.	.	.	.

Logical Group

Instruction			Mnemonic Meaning				Flags			
							Zf	Cf	Pf	Sf
CMP	reg		T<=A-reg				x	x	x	x
CPI	byte		T<=A-byte				x	x	x	x
CMA			A<=1's complement of A				.	.	.	.
CMC			Cf<=1's complement of Cf				.	x	.	.
STC			Cf<=1				.	1	.	.
SHLR			HL<=HL/2 while H6<=H7 (extend sign) and Cf<=L0				.	x	.	.
ANA	reg		A<=A AND reg				x	0	x	x
ANI	byte		A<=A AND byte				x	0	x	x
ORA	reg		A<=A Inclusive OR reg				x	0	x	x
ORI	byte		A<=A Inclusive OR byte				x	0	x	x
XRA	reg		A<=A Exclusive OR reg				x	0	x	x
XRI	byte		A<=A Exclusive OR byte				x	0	x	x
RAL			A<=A*2 where Cf<=A7 while A0<=Cf				.	x	.	.
RAR			A<=A/2 where Cf<=A0 while A7<=Cf				.	x	.	.
RLC			A7..A1<=A6..A0 while A0<=A7 and Cf<=A7				.	x	.	.
RRC			A6..A0<=A7..A1 while A7<=A0 and Cf<=A0				.	x	.	.
RDEL			DE<=DE*2 where: Cf<=DE15 while DE00<=Cf				.	x	.	.

Stack, Input/Output, & Machine Control Group

Instruction			Mnemonic Meaning				Flags			
							Zf	Cf	Pf	Sf
PUSH	rp		SP<=SP-2, w[SP]<=rp				.	.	.	.
POP	rp		rp<=w[SP], SP<=SP+2				.	.	.	.
SPHL			SP<=HL				.	.	.	.
XTHL			HL<=w[SP] while w[SP]<=HL				.	.	.	.
IN	port		A<=data from port				.	.	.	.
OUT	port		data to port<=A				.	.	.	.
DI			disable interrupts				.	.	.	.
EI			enable interrupts				.	.	.	.
RIM			A<=interrupt mask				.	.	.	.
SIM			interrupt mask<=A				.	.	.	.
NOP			do nothing				.	.	.	.
HLT			halt 8085 processor				.	.	.	.

8085 Instructions by Mnemonic

Instruction			Mnemonic Meaning				Flags			
							Zf	Cf	Pf	Sf
ACI	byte		Add with Carry Immediate				x	x	x	x
ADC	reg		ADd with Carry				x	x	x	x
ADD	reg		ADD				x	x	x	x
ADI	byte		ADd Immediate				x	x	x	x
ANA	reg		ANd Accumulator				x	0	x	x
ANI	byte		ANd Immediate				x	0	x	x
CALL	label		CALL unconditiona				.	.	.	.
CC	label		Call if Carry				.	.	.	.
CM	label		Call if Minus				.	.	.	.
CMA			CoMplement Accumulator				.	.	.	.
CMC			CoMplement Carry				.	x	.	.
CMP	reg		CoMPare				x	x	x	x
CNC	label		Call if No Carry				.	.	.	.
CNZ	label		Call if No Zero				.	.	.	.
CP	label		Call if Positive				.	.	.	.
CPE	label		Call if Parity Even				.	.	.	.
CPI	byte		CompAre Immediate				x	x	x	x
CPO	label		Call if Parity Odd				.	.	.	.
CZ	label		Call if Zero				.	.	.	.
DAA			Decimal Adjust Accumulator				x	x	x	x
DAD	rp		Dual-register ADd to hl				.	x	.	.
DCR	reg		DeCrement Register				x	.	x	x
DCX	rp		DeCrement eXtended-register				.	.	.	.
DEHL	byte		DE from HL plus byte				.	.	.	.
DESP	byte		DE from SP plus byte				.	.	.	.
DI			Disable Interrupts				.	.	.	.
EI			Enable Interrupts				.	.	.	.
HLMB C			HL Minus BC				x	x	h	x
HLT			HaLT				.	.	.	.
IN	port		INput from port				.	.	.	.
INR	reg		INcrement Register				x	.	x	x
INX	rp		INcrement eXtended-register				.	.	.	.
JC	label		Jump if Carry				.	.	.	.
JM	label		Jump if Minus				.	.	.	.
JMP	label		JuMP unconditional				.	.	.	.
JNC	label		Jump if No Carry				.	.	.	.
JNZ	label		Jump if No Zero				.	.	.	.
JP	label		Jump if Positive				.	.	.	.
JPE	label		Jump if Parity Even				.	.	.	.
JPO	label		Jump if Parity Odd				.	.	.	.
JTM	label		Jump if True sign Minus				.	.	.	.

Instruction			Mnemonic Meaning				Flags			
							Zf	Cf	Pf	Sf
JTP	label		Jump if True sign Positive				.	.	.	.
JZ	label		Jump if Zero				.	.	.	.
LDA	addr		LoaD Accumulator direct				.	.	.	.
LDAX	B		LoaD Accumulator indirect via eXtended-register Bc				.	.	.	.
LDAX	D		LoaD Accumulator indirect via eXtended-register De				.	.	.	.
LHLD	addr		Load HL Direct				.	.	.	.
LHLI			Load HL Indirect via extended register de				.	.	.	.
LXI	rp, word		Load eXtended-register Immediate				.	.	.	.
MOV	dreg, sreg		MOVe							
MVI	reg, byte		MoVe Immediate				.	.	.	.
MOVX	drp, srp		MoVe eXtended-register (pseudo for high & low MOVs)				.	.	.	.
NOP			No OPeration				.	.	.	.
ORA	reg		OR Accumulator				x	0	x	x
ORI	byte		OR Immediate				x	0	x	x
OUT	port		OUTput to port				.	.	.	.
PCHL			Program Counter from HL				.	.	.	.
POP	rp		POP off stack				.	.	.	.
PUSH	rp		PUSH on stack				.	.	.	.
RAL			Rotate Accumulator Left through carry				.	x	.	.
RAR			Rotate Accumulator Right through carry				.	x	.	.
RC			Return if Carry				.	.	.	.
RDEL			Rotate DE Left through carry				.	x	.	.
RET			RETurn unconditional				.	.	.	.
RIM			Read Interrupt Mask				.	.	.	.
RLC			Rotate accumulator Left Circular				.	x	.	.
RM			Return if Minus				.	.	.	.
RNC			Return if No Carry				.	.	.	.
RNZ			Return if No Zero				.	.	.	.
RP			Return if Positive				.	.	.	.
RPE			Return if Parity Even				.	.	.	.
RPO			Return if Parity Odd				.	.	.	.
RRC			Rotate accumulator Right Circular				.	x	.	.
RST	n		ReSTart				.	.	.	.
RSTV			ReSTart if oVerflow				.	.	.	.
RZ			Return if Zero				.	.	.	.
SBB	reg		SuBtract with Borrow				x	x	x	x
SBI	byte		Subtract with Borrow Immediate				x	x	x	x

Instruction			Mnemonic Meaning				Flags			
							Zf	Cf	Pf	Sf
SBI	byte		Subtract with Borrow Immediate				x	x	x	x
SHLD	addr		Store HL Direct				.	.	.	.
SHLI			Store HL Indirect via extended register de				.	.	.	.
SHLR			Shift HL Right				.	x	.	.
SIM			Set Interrupt Mask				.	.	.	.
SPHL			Stack Pointer from HL				.	.	.	.
STA	addr		STore Accumulator direct				.	.	.	.
STAX	B		Store Accumulator indirect via eXtended-register Bc				.	.	.	.
STAX	D		Store Accumulator indirect via eXtended-register De				.	.	.	.
STC			SeT Carry				.	1	.	.
SUB	reg		SUBtract				x	x	x	x
SUI	byte		SUBtract Immediate				x	x	x	x
XCHG			eXCHanGe hl with de				.	.	.	.
XRA	reg		eXclusive oR Accumulator				x	0	x	x
XRI	byte		eXclusive oR Immediate				x	0	x	x
XTHL			eXchange Top of stack with HL				.	.	.	.

8085 Machine Cycles by Functional Group

Data Transfer Group

		Essential Cycles +register M involved or condition Met	EC +MM
Instruction		Mnemonic Meaning	Cycles
MOV	dreg, sreg	MOVe	04 +03
MVI	reg, byte	MOVe Immediate	07 +03
MVX	drp, srp	MOVe eXtended-register (pseudo for high & low MOVs)	--
LXI	rp, word	Load eXtended-register Immediate	10
XCHG		eXCHanGe hl with de	04
LDA	addr	LoaD Accumulator direct	13
STA	addr	STore Accumulator direct	13
LDAX	B	LoaD Accumulator indirect via eXtended-register Bc	07
STAX	B	Store Accumulator indirect via eXtended-register Bc	07
LDAX	D	LoaD Accumulator indirect via eXtended-register De	07
STAX	D	Store Accumulator indirect via eXtended-register De	07
LHLD	addr	Load HL Direct	16
SHLD	addr	Store HL Direct	16
LHLI		Load HL Indirect via extended register de	10
SHLI		Store HL Indirect via extended register de	10

Arithmetic Group

		Essential Cycles +register M involved or condition Met	EC +MM
Instruction		Mnemonic Meaning	Cycles
ADD	reg	ADD	04 +03
ADI	byte	ADd Immediate	07
ADC	reg	ADd with Carry	04 +03
ACI	byte	Add with Carry Immediate	07
SUB	reg	SUBtract	04 +03
SUI	byte	SUBtract Immediate	07
SBB	reg	SuBtract with Borrow	04 +03
SBI	byte	Subtract with Borrow Immediate	07
DAA		Decimal Adjust Accumulator	04
INR	reg	INcrement Register	04 +06
INX	rp	INcrement eXtended-register	06
DCR	reg	DeCrement Register	04 +06
DCX	rp	DeCrement eXtended-register	06
DAD	rp	Dual-register Add to hl	10
HLMBC		HL Minus BC	10
DEHL	byte	DE from HL plus byte	10
DESP	byte	DE from SP plus byte	10

Logical Group

		Essential Cycles +register M involved or condition Met	EC +MM
Instruction		Mnemonic Meaning	Cycles
CMP	reg	CoMPare	04 +03
CPI	byte	ComPare Immediate	07
CMA		CoMplement Accumulator	04
CMC		CoMplement Carry	04
STC		SeT Carry	04
SHLR		Shift HL Right	07
ANA	reg	And Accumulator	04 +03
ANI	byte	AND Immediate	07
ORA	reg	OR Accumulator	04 +03
ORI	byte	OR Immediate	07
XRA	reg	eXclusive oR Accumulator	04 +03
XRI	byte	eXclusive oR Immediate	07
RAL		Rotate Accumulator Left through carry	04
RAR		Rotate Accumulator Right through carry	04
RLC		Rotate accumulator Left Circular	04
RRC		Rotate accumulator Right Circular	04
RDEL		Rotate DE Left through carry	10

Stack, Input/Output, & Machine Control Group

		Essential Cycles +register M involved or condition Met	EC +MM
Instruction		Mnemonic Meaning	Cycles
PUSH	rp	PUSH on stack	12
POP	rp	POP off stack	10
SPHL		Stack Pointer from HL	06
XTHL		eXchange Top of stack with HL	16
IN	port	INput from port	10
OUT	port	OUTput to port	10
DI		Disable Interrupts	04
EI		Enable Interrupts	04
RIM		Read Interrupt Mask	04
SIM		Set Interrupt Mask	04
NOP		No OPeration	04
HLT		HaLT	05

Branch Group

		Essential Cycles +register M involved or condition Met	EC +MM
Instruction		Mnemonic Meaning	Cycles
JMP	label	JuMP unconditional	10
JZ	label	Jump if Zero	07 +03
JNZ	label	Jump if No Zero	07 +03
JP	label	Jump if Positive	07 +03
JM	label	Jump if Minus	07 +03
JC	label	Jump if Carry	07 +03
JNC	label	Jump if No Carry	07 +03
JTM	label	Jump if True sign Minus	07 +03
JTP	label	Jump if True sign Positive	07 +03
JPE	label	Jump if Parity Even	07 +03
JPO	label	Jump if Parity Odd	07 +03
CALL	label	CALL unconditiona	18
CZ	label	Call if Zero	09 +09
CNZ	label	Call if No Zero	09 +09
CP	label	Call if Positive	09 +09
CM	label	Call if Minus	09 +09
CC	label	Call if Carry	09 +09
CNC	label	Call if No Carry	09 +09
CPE	label	Call if Parity Even	09 +09
CPO	label	Call if Parity Odd	09 +09
RET		RETurn unconditional	10
RZ		Return if Zero	06 +06
RNZ		Return if No Zero	06 +06
RP		Return if Positive	06 +06
RM		Return if Minus	06 +06
RC		Return if Carry	06 +06
RNC		Return if No Carry	06 +06
RPE		Return if Parity Even	06 +06
RPO		Return if Parity Odd	06 +06
PCHL		Program Counter from HL	06
RST	n	ReSTart	12
RSTV		ReSTart if oVerflow	06 +06

8085 Instruction Mnemonics by Op-code

	x0h	x1h	x2h	x3h	x4h	x5h	x6h	x7h	x8h	x9h	xAh	xBh	xCh	xDh	xEh	xFh
00h-0Fh	NOP	LXI B,w	STAX B	INX B	INR B	DCR B	MVI B,b	RLC	HLMBC	DAD B	LDAX B	DCX B	INR C	DCR C	MVI C,b	RRC
10h-1Fh	SHLR	LXI D,w	STAX D	INX D	INR D	DCR D	MVI D,b	RAL	RDEL	DAD D	LDAX D	DCX D	INR E	DCR E	MVI E,b	RAR
20h-2Fh	RIM	LXI H,w	SHLD	INX H	INR H	DCR H	MVI H,b	DAA	DEHL b	DAD H	LHLD	DCX H	INR L	DCR L	MVI L,b	CMA
30h-3Fh	SIM	LXI SP,w	STA @	INX SP	INR M	DCR M	MVI M,b	STC	DESP b	DAD SP	LDA @	DCX SP	INR A	DCR A	MVI A,b	CMC
40h-4Fh	MOV B,B	MOV B,C	MOV B,D	MOV B,E	MOV B,H	MOV B,L	MOV B,M	MOV B,A	MOV C,B	MOV C,C	MOV C,D	MOV C,E	MOV C,H	MOV C,L	MOV C,M	MOV C,A
50h-5Fh	MOV D,B	MOV D,C	MOV D,D	MOV D,E	MOV D,H	MOV D,L	MOV D,M	MOV D,A	MOV E,B	MOV E,C	MOV E,D	MOV E,E	MOV E,H	MOV E,L	MOV E,M	MOV E,A
60h-6Fh	MOV H,B	MOV H,C	MOV H,D	MOV H,E	MOV H,H	MOV H,L	MOV H,M	MOV H,A	MOV L,B	MOV L,C	MOV L,D	MOV L,E	MOV L,H	MOV L,L	MOV L,M	MOV L,A
70h-7Fh	MOV M,B	MOV M,C	MOV M,D	MOV M,E	MOV M,H	MOV M,L	HLT	MOV M,A	MOV A,B	MOV A,C	MOV A,D	MOV A,E	MOV A,H	MOV A,L	MOV A,M	MOV A,A
80h-8Fh	ADD B	ADD C	ADD D	ADD E	ADD H	ADD L	ADD M	ADD A	ADC B	ADC C	ADC D	ADC E	ADC H	ADC L	ADC M	ADC A
90h-9Fh	SUB B	SUB C	SUB D	SUB E	SUB H	SUB L	SUB M	SUB A	ANA B	ANA C	ANA D	ANA E	ANA H	ANA L	ANA M	ANA A
A0h-AFh	ANA B	ANA C	ANA D	ANA E	ANA H	ANA L	ANA M	ANA A	XRA B	XRA C	XRA D	XRA E	XRA H	XRA L	XRA M	XRA A
B0h-BFh	ORA B	ORA C	ORA D	ORA E	ORA H	ORA L	ORA M	ORA A	CMP B	CMP C	CMP D	CMP E	CMP H	CMP L	CMP M	CMP A
COh-CFh	RNZ	POP B	JNZ @	JMP @	CNZ @	PUSH B	ADI b	RST 0	RZ	RET	JZ @	RSTV	CZ @	CALL @	ACI b	RST 1
D0h-DFh	RNC	POP D	JNC @	OUT port	CNC @	PUSH D	SUI b	RST 2	RC	SHLI	JC @	IN port	CC @	JTP @	SBI b	RST 3
E0h-EFh	RPO	POP H	JPO @	XTHL	CPO @	PUSH H	ANI b	RST 4	RPE	PCHL	JPE @	XCHG	CPE @	LHLI	XRI b	RST 5
F0h-FFh	RP	POP PSW	JP @	DI	CP @	PUSH PSW	ORI b	RST 6	RM	SPHL	JM @	EI	CM @	JTM @	CPI b	RST 7