

8085 Instruction Set

Instructions		Operation	Cycles	Bytes	Flag	Description
Mnemonics	Arguments					
ACI	n	$A=A+n+CY$	7	2	SZAPC	Add with Carry Immediate
ADC	r	$A=A+r+CY(21X)$	4	1	SZAPC	Add with Carry
ADC	M	$A=A+[HL]+CY$	7	1	SZAPC	Add with Carry to Memory
ADD	r	$A=A+r (20X)$	4	1	SZAPC	Add
ADD	M	$A=A+[HL]$	7	1	SZAPC	Add to Memory
ADI	n	$A=A+n$	7	2	SZAPC	Add Immediate

DAD	B	$HL=HL+BC$	10	1	SZAPC	Double Add BC to HL
DAD	D	$HL=HL+DE$	10	1	SZAPC	Double Add DE to HL
DAD	H	$HL=HL+HL$	10	1	SZAPC	Double Add HL to HL
DAD	SP	$HL=HL+SP$	10	1	SZAPC	Double Add SP to HL

DCR	r	$r=r-1$	4	1	SZAPC	Decrement
DCR	M	$[HL]=[HL]-1$	4	1	SZAPC	Decrement Memory
DCX	B	$BC=BC-1$	6	1		Decrement BC
DCX	D	$DE=DE-1$	6	1		Decrement DE
DCX	H	$HL=HL-1$	6	1		Decrement HL
DCX	SP	$SP=SP-1$	6	1		Decrement Stack Pointer

INR	r	$r=r+1 (0X4)$	4	1	SZAPC	Increment
INR	M	$[HL]=[HL]+1$	4	1	SZAPC	Increment Memory
INX	B	$BC=BC+1$	6	1		Increment BC
INX	D	$DE=DE+1$	6	1		Increment DE
INX	H	$HL=HL+1$	6	1		Increment HL
INX	SP	$SP=SP+1$	6	1		Increment Stack Pointer

SBB	r	$A=A-r-CY$	4	1	SZAPC	Subtract with Borrow
SBB	M	$A=A-[HL]-CY$	7	1	SZAPC	Subtract with Borrow
SBI	n	$A=A-n-CY$	7	2	SZAPC	Subtract with Borrow Immed

SUB	r	$A=A-r (22X)$	4	1	SZAPC	Subtract
SUB	M	$A=A-[HL]$	7	1	SZAPC	Subtract Memory
SUI	n	$A=A-n$	7	2	SZAPC	Subtract Immediate

Instructions		Operation	Cycles	Bytes	Flag	Description
Mnemonics	Arguments					
ANA	r	$A=A \& r$	4	1	<u>SZAP</u> <u>O</u>	AND Accumulator
ANA	M	$A=A \& [HL]$	4	1	<u>SZAP</u> <u>O</u>	AND Accumulator and Memory
ANI	n	$A=A \& n$	7	2	<u>SZOP</u> <u>O</u>	AND Immediate
CMA		$A=\sim A$	4	1	SZAPC	Complement Accumulator
ORA	r	$A=A r$	4	1	<u>SZOP</u> <u>O</u>	Inclusive OR Accumulator
ORA	M	$A=A [HL]$	7	1	<u>SZOP</u> <u>O</u>	Inclusive OR Accumulator
ORI	n	$A=A n$	7	2	<u>SZOP</u> <u>O</u>	Inclusive OR Immediate
XRA	r	$A=A \oplus r$	4	1	<u>SZOP</u> <u>O</u>	Exclusive OR Accumulator
XRA	M	$A=A \oplus [HL]$	7	1	<u>SZOP</u> <u>O</u>	Exclusive OR Accumulator
XRI	n	$A=A \oplus n$	7	2	<u>SZOP</u> <u>O</u>	Exclusive OR Immediate
RAL		$A=\{CY,A\} \ll 1$	4	1	SZAP <u>C</u>	Rotate Accumulator Left
RAR		$A=\gg \{CY,A\}$	4	1	SZAP <u>C</u>	Rotate Accumulator Right
RLC		$A=A \ll 1$	4	1	SZAP <u>C</u>	Rotate Left Circular
RRC		$A=\gg A$	4	1	SZAP <u>C</u>	Rotate Right Circular
CMP	r	$A-r$	4	1	<u>SZAP</u> <u>C</u>	Compare
CMP	M	$A-[HL]$	7	1	<u>SZAP</u> <u>C</u>	Compare with Memory
CPI	n	$A-n$	7	2	<u>SZAP</u> <u>C</u>	Compare Immediate

Instructions		Operation	Cycles	Bytes	Flag	Description
Mnemonics	Arguments					
MOV	r1,r2	r1=r2	4	1	-----	Move register to register
MOV	M,r	[HL]=r	7	1	-----	Move register to Memory
MOV	r,M	r=[HL]	7	1	-----	Move Memory to register
MVI	r,n	r=n	7	2	-----	Move Immediate
MVI	M,n	[HL]=n	10	2	-----	Move Immediate to Memory

LDA	a	A=[a]	13	3	-----	Load Accumulator direct
LDAX	B	A=[BC]	7	1	-----	Load Accumulator indirect
LDAX	D	A=[DE]	7	1	-----	Load Accumulator indirect
LHLD	a	HL=[a]	16	3	-----	Load HL Direct
LXI	B,nn	BC=nn	10	3	-----	Load Immediate BC
LXI	D,nn	DE=nn	10	3	-----	Load Immediate DE
LXI	H,nn	HL=nn	10	3	-----	Load Immediate HL
LXI	SP,nn	SP=nn	10	3	-----	Load Immediate Stack Ptr

STA	a	[a]=A	13	3	-----	Store Accumulator
STAX	B	[BC]=A	7	1	-----	Store Accumulator indirect
STAX	D	[DE]=A	7	1	-----	Store Accumulator indirect

SHLD	a	[a]=HL	16	3	-----	Store HL Direct
SPHL		SP=HL	6	1	-----	Move HL to SP
XCHG		HL<->DE	4	1	-----	Exchange HL with DE
XTHL		[SP]<->HL	16	1	-----	Exchange stack Top with HL

IN	p	A=[p]	10	2	-----	Input
OUT	p	[p]=A	10	2	-----	Output

Instructions		Operation	Cycles	Bytes	Flag	Description
Mnemonics	Arguments					
JMP	a	PC=a	7	3	-----	Jump unconditional
JM	a	If S=1	7/(10~s)	3	-----	Jump on Minus
JP	a	If S=0	7/(10~s)	3	-----	Jump on Plus
JC	a	If CY=1	7/(10~s)	3	-----	Jump on Carry
JNC	a	If CY=0	7/(10~s)	3	-----	Jump on No Carry
JZ	a	If Z=1	7/(10~s)	3	-----	Jump on Zero
JNZ	a	If Z=0	7/(10~s)	3	-----	Jump on No Zero
JPE	a	If P=1	7/(10~s)	3	-----	Jump on Parity Even
JPO	a	If P=0	7/(10~s)	3	-----	Jump on Parity Odd

PCHL		PC=[HL]	6	1	-----	Jump HL indirect
------	--	---------	---	---	-------	------------------

CALL	a	-[SP]=PC,PC=a	18	3	-----	Call unconditional
CM	a	If S=1	9/(18~s)	3	-----	Call on Minus
CP	a	If S=0	9/(18~s)	3	-----	Call on Plus
CC	a	If CY=1	9/(18~s)	3	-----	Call on Carry
CNC	a	If CY=0	9/(18~s)	3	-----	Call on No Carry
CZ	a	If Z=1	9/(18~s)	3	-----	Call on Zero
CNZ	a	If Z=0	9/(18~s)	3	-----	Call on No Zero
CPE	a	If P=1	9/(18~s)	3	-----	Call on Parity Even
CPO	a	If P=0	9/(18~s)	3	-----	Call on Parity Odd

RET		PC=[SP]+	10	1	-----	Return
RM		If S=1	6/(12~s)	1	-----	Return on Minus
RP		If S=0	6/(12~s)	1	-----	Return on Plus
RC		If CY=1	6/(12~s)	1	-----	Return on Carry
RNC		If CY=0	6/(12~s)	1	-----	Return on No Carry
RZ		If Z=1	6/(12~s)	1	-----	Return on Zero
RNZ		If Z=0	6/(12~s)	1	-----	Return on No Zero
RPE		If P=1	6/(12~s)	1	-----	Return on Parity Even
RPO		If P=0	6/(12~s)	1	-----	Return on Parity Odd

Instructions		Operation	Cycles	Bytes	Flag	Description
Mnemonics	Arguments					
CMC		CY= \sim CY	4	1	SZAPC	Complement Carry
STC		CY=1	4	1	1	Set Carry

NOP			4	1	-----	No Operation
RST	z	-[SP]=PC,PC=z	12	1	-----	Restart (3X7)
HLT			5	1		Halt
DAA		A=BCD format	4	1	SZAPC	Decimal Adjust Accumulator

SIM		mask=A	4	1	-----	Set Interrupt Mask
RIM		A=mask	4	1	-----	Read Interrupt Mask
DI			4	1	-----	Disable Interrupts
EI			4	1	-----	Enable Interrupts

POP	B	BC=[SP]+	10	1	-----	Pop BC
POP	D	DE=[SP]+	10	1	-----	Pop DE
POP	H	HL=[SP]+	10	1	-----	Pop HL
POP	PSW	{PSW,A}=[SP]+	10	1	-----	Pop Processor Status Word
PUSH	B	-[SP]=BC	12	1	-----	Push BC
PUSH	D	-[SP]=DE	12	1	-----	Push DE
PUSH	H	-[SP]=HL	12	1	-----	Push HL
PUSH	PSW	-[SP]={PSW,A}	12	1	-----	Push Processor Status Word