

Befehl	Operand	Wirkung
ACI	konstante	addiere zum Akku die Konstante und das Carrybit
ADC	register	addiere zum Akku ein Register und das Carrybit
ADD	register	addiere zum Akku ein Register
ADI	konstante	addiere zum Akku die Konstante
ANA	register	bilde das logische UND des Akkus mit einem Register
ANI	konstante	bilde das logische UND des Akkus mit der Konstanten
CALL	adresse	rufe ein Unterprogramm unbedingt
CC	adresse	rufe Unterprogramm nur, wenn das Carrybit 1 ist
CM	adresse	rufe Unterprogramm nur, wenn das Vorzeichenbit(S) 1 ist
CMA		komplementiere den Akku (Einerkomplement)
CMC		komplementiere das Carrybit
CMP	register	vergleiche den Akku mit dem Register (Testsubtraktion)
CNC	adresse	rufe Unterprogramm nur, wenn das Carrybit 0 ist
CNZ	adresse	rufe Unterprogramm nur, wenn Ergebnis ungleich Null ist
CP	adresse	rufe Unterprogramm nur, wenn Vorzeichenbit (S) 0 ist
CPE	adresse	rufe Unterprogramm nur, wenn Paritätsbit 1 ist
CPI	konstante	vergleiche Akku mit der Konstanten (Testsubtraktion)
CPO	adresse	rufe Unterprogramm nur, wenn Paritätsbit 0 ist
CZ	adresse	rufe Unterprogramm nur, wenn Ergebnis gleich Null ist
DAA		korrigiere den Akku im BCD-Code
DAD	reg.-paar	addiere Registerpaar zum HL-Registerpaar (16 Bit)
DCR	register	vermindere Register um 1
DCX	reg.-paar	vermindere Registerpaar um 1 (16 Bit)
DI		sperre alle Interrupts (Interrupt-Flipflop = 0)
EI		gib alle Interrupts frei (Interrupt-Flipflop = 1)
HLT		anhalten und auf Interrupt warten
IN	port	lade den Akku mit einem Eingabeport
INR	register	erhöhe Register um 1
INX	reg.-paar	erhöhe Registerpaar um 1 (16 Bit)
JC	adresse	springe nur, wenn das Carrybit 1 ist
JM	adresse	springe nur, wenn das Vorzeichenbit (S-Bit) 1 ist
JMP	adresse	springe immer
JNC	adresse	springe nur, wenn das Carrybit 0 ist
JNZ	adresse	springe nur, wenn das Ergebnis ungleich Null ist (Z=0)
JP	adresse	springe nur, wenn das Vorzeichenbit (S-Bit) 0 ist
JPE	adresse	springe nur, wenn das Paritätsbit 1 ist
JPO	adresse	springe nur, wenn das Paritätsbit 0 ist
JZ	adresse	springe nur, wenn das Ergebnis gleich Null ist (Z=1)
LDA	adresse	lade den Akku mit dem Inhalt eines Speicherbytes
LDAX	B oder D	lade den Akku mit Speicherbyte (Adresse in BC oder DE)

Befehl	Operand	Wirkung
LHLD	adresse	lade L mit adressiertem Byte, H mit folgendem Byte
LXI	rp,konst	lade Registerpaar mit einer 16-Bit-Konstanten
MOV	reg1,reg2	lade Register reg1 mit Register reg2
MVI	reg,konst	lade Register mit der Konstanten
NOP		tu nichts (Zeitverzögerung oder Platzhalter)
ORA	register	bilde das logische ODER des Akkus mit einem Register
ORI	konstante	bilde das logische ODER des Akkus mit der Konstanten
OUT	port	speichere den Akku in den Ausgabeport
PCHL		lade den Befehlszähler mit dem HL-Registerpaar (Sprung)
POP	reg.-paar	hole das Registerpaar aus dem Stapel , Stapelzeiger + 2
PUSH	reg.-paar	bringe das Registerpaar in den Stapel , Stapelzeiger -2
RAL		schiebe den Akku mit dem Carrybit zyklisch links
RAR		schiebe den Akku mit dem Carrybit zyklisch rechts
RC		Rücksprung nur, wenn das Carrybit 1 ist
RET		springe immer aus dem Unterprogramm zurück
RIM		lade den Akku mit dem Interruptregister
RLC		schiebe den Akku ohne das Carrybit zyklisch links
RM		Rücksprung nur, wenn Vorzeichenbit (S) 1 ist
RNC		Rücksprung nur, wenn Carrybit 0 ist
RNZ		Rücksprung nur, wenn Ergebnis ungleich Null ist (Z=0)
RP		Rücksprung nur, wenn Vorzeichenbit (S) 0 ist
RPE		Rücksprung nur, wenn Paritätsbit 1 ist
RPO		Rücksprung nur, wenn Paritätsbit 0 ist
RRC		schiebe den Akku ohne Carrybit zyklisch rechts
RST	0 - 7	starte Interruptprogramm , Befehlszähler nach Stapel
RZ		Rücksprung nur, wenn Ergebnis gleich Null (Z=1)
SBB	register	subtrahiere Register und Carrybit vom Akku
SBI	konstante	subtrahiere Konstante und Carrybit vom Akku
SHLD	adresse	speichere L nach adressiertem Byte, H nach folgendem
SIM		speichere den Akku in das Interruptregister
SPHL		lade den Stapelzeiger mit dem HL-Registerpaar
STA	adresse	speichere den Akku in das adressierte Byte
STAX	B oder D	speichere Akku nach Speicherbyte (Adresse in BC oder DE)
STC		setze das Carrybit 1
SUB	register	subtrahiere Register vom Akku
SUI	konstante	subtrahiere Konstante vom Akku
XCHG		vertausche HL-Registerpaar mit dem DE-Registerpaar
XRA	register	bilde das logische EODER des Akkus mit einem Register
XRI	konstante	bilde das logische EODER des Akkus mit der Konstanten
XTHL		vertausche HL-Registerpaar mit den beiden Stapelbytes