

MOP: Befehlsliste für den Mikrocontroller 8051

Erläuterung der Operanden

Operand	Erklärung
A	Akkumulator
addr11	Adresse innerhalb eines 2 Kilobyte-Programmspeicherblocks
addr16	Adresse irgendwo im 64 Kilobyte-Programmspeicherraum
bit	Bitadresse im internen RAM
/bit	Inhalt der Bitadresse wird invertiert
C	Carry-Bit
#data	8-Bit-Konstante
#data16	16-Bit-Konstante
direct	Adresse eines Platzes im internen RAM
DPTR	Datenpointer
PC	Programmzähler (program counter)
Ri	Indirekte Adressierung eines Speicherplatzes im RAM durch R1 oder R2
Rn	Register R0 bis R7 der gerade aktuellen Registerbank
rel	Signiertes 8-Bit-Offset für Sprungbefehle (relativer Sprung um max. 128 Byte nach oben oder unten)

Arithmetische Befehle

Diese Befehle umfassen die Addition, Subtraktion, Multiplikation, Division und die Dezimalkorrektur (für BCD-Zahlen). Außerdem das Inkrementieren (= Erhöhen eines Registerinhaltes um 1) und das Dekrementieren (Erniedrigen eines Registerinhaltes um 1).

Hex-Code	Mnemonik	Beschreibung	Byte	Masch.- Zyklen
28.....2F	ADD A, Rn	Addiere den Inhalt des ausgewählten Registers (R0.....R7) zum Akkumulator	1	1
25	ADD A, direct	Addiere den Inhalt des direkt adressierten RAM-Platzes zum Akku	2	1
26	ADD A,@R0	Addiere den Inhalt des indirekt durch R0 adressierten RAM-Platzes zum Akku	1	1
27	ADD A,@R1	Addiere den Inhalt des indirekt durch R1 adressierten RAM-Platzes zum Akku	1	1
24	ADD A,#data	Addiere die angegebene Konstante zum Akku	2	1
38.....3F	ADDC A,Rn	Addiere den Inhalt des ausgewählten Registers (R0.....R7) zum Akku inkl. Carry	1	1
35	ADDC A,direct	Addiere den Inhalt des direkt adressierten RAM-Platzes zum Akku inkl. Carry	2	1
36	ADDC A,@R0	Addiere den Inhalt des indirekt durch R0 adressierten RAM-Platzes zum Akku inkl. Carry	1	1
37	ADDC A,@R1	Addiere den Inhalt des indirekt durch R1 adressierten RAM-Platzes zum Akku inkl. Carry	1	1

34	ADDC	A,#data	Addiere die angegebene Konstante zum Akku inkl. Carry	2	1
98.....9F	SUBB	A,Rn	Subtrahiere den Inhalt des ausgewählten Registers (R0.....R7) vom Akku	1	1
95	SUBB	A,direct	Subtrahiere den Inhalt des direkt adressierten RAM-Platzes vom Akku	2	1
96	SUBB	A,@R0	Subtrahiere den Inhalt des indirekt durch R0 adressierten RAM-Platzes vom Akku	1	1
97	SUBB	A,@R1	Subtrahiere den Inhalt des indirekt durch R1 adressierten RAM-Platzes vom Akku	1	1
94	SUBB	A,#data	Subtrahiere die angegebene Konstante vom Akku	2	1
04	INC	A	Erhöhe (= inkrementiere) den Akkuinhalt um 1	1	1
08.....0F	INC	Rn	Inkrementiere das angegebene Register (R0.....R7)	1	1
05	INC	direct	Inkrementiere den direkt adressierten RAM-Platz	2	1
06	INC	@R0	Inkrementiere den durch R0 indirekt adressierten RAM-Platz	1	1
07	INC	@R1	Inkrementiere den durch R1 indirekt adressierten RAM-Platz	1	1
14	DEC	A	Vermindere (= dekrementiere) den Akku um 1	1	1
18.....1F	DEC	Rn	Dekrementiere das angegebene Register (R0.....R7)	1	1
15	DEC	direct	Dekrementiere den direkt adressierten RAM-Platz	1	1
16	DEC	@R0	Dekrementiere den durch R0 indirekt adressierten RAM-Platz	1	1
17	DEC	@R1	Dekrementiere den durch R1 indirekt adressierten RAM-Platz	1	1
A3	INC	DPTR	Inkrementiere den Datenpointer	1	2
A4	MUL	AB	Multipliziere Akku und B-Register	1	4
84	DIV	AB	Dividiere den Akku durch das B-Register	1	4
D4	DA	A	Dezimalkorrektur des Akkus bei BCD-Operationen	1	1

Sprungbefehle

Man muss zunächst zwischen

SJMP (= short jump = Sprung um max. 127 nach oben oder unten)

AJMP (= absolute jump = Sprung um max. 2 Kilobyte nach oben oder unten) und

LJMP (= long jump = Sprung innerhalb des kompletten 64 Kilobyte-Adressraumes)

unterscheiden. Die Unterschiede liegen sowohl in der Ausführungszeit, als auch im Speicherplatzbedarf. Außerdem finden sich hier viele bedingte Sprungbefehle (z. B. Sprung, wenn Akku leer usw.)

Zu dieser Gruppe gehören auch die Aufruf-Befehle für Unterprogramme (ACALL, LCALL), sowie das Rückkehrkommando (RET) und der Befehl NOP (no operation = tu gar nix).

Hinweis: Je nach verwendetem Assembler mag es ausreichen, die Befehle JMP oder CALL im Programm zu verwenden. Die geeigneten Jump- oder Call-Befehl werden dann automatisch eingesetzt.

Hex-Code	Mnemonik	Beschreibung	Byte	Masch.- Zyklen
11 oder F1	ACALL addr11	Subroutinenaufruf innerhalb eines 2 Kilobyte Programmspeicherblocks	2	2
12	LCALL addr15	Subroutinenaufruf innerhalb des 64 Kilobyte-	3	2

Programmspeicher-Raumes					
22	RET		Zurück aus der Subroutine	1	2
32	RETI		Zurück aus der Interrupt-Routine	1	2
01 oder E1	AJMP	addr11	Sprung innerhalb eines 2 Kilobyte-Blockes	2	2
02	LJMP	addr16	Sprung innerhalb des 64 Kilobyte - Adressraumes	3	2
80	SJMP	rel	relativer Sprung um max. 128 Byte nach oben oder nach unten	2	2
73	JMP	@A + DPTR	Indirekter Sprung relativ zum Datenpointer	1	2
60	JZ	rel	Relativer Sprung um max. 128 Byte, wenn der Akku leer ist	2	2
70	JNZ	rel	Relativer Sprung um max. 128 Byte, wenn der Akku nicht leer ist	2	2
B5	CJNE	A,direct,rel	Springe, wenn Akku und direkt adressierter RAM-Platz ungleiche Inhalte haben, um max. 128 Byte	3	2
B4	CJNE	A,#data,rel	Springe, wenn der Akku-Inhalt und die vorgegebene Konstante ungleich sind, um max. 128 Byte	3	2
B8 bis BF	CJNE	Rn,#data,rel	Springe, wenn der Inhalt des gewählten Registers (R0 bis R7) nicht mit der vorgegebenen Konstante übereinstimmt, um max. 128 Byte	3	2
B6	CJNE	@R0,#data,rel	Springe, wenn der Inhalt der indirekt durch R0 adressierten Speicherzelle nicht mit der vorgegebenen Konstante übereinstimmt, um max. 128 Byte	3	2
B7	CJNE	@R1,#data,rel	Springe, wenn der Inhalt der indirekt durch R1 adressierten Speicherzelle nicht mit der vorgegebenen Konstante übereinstimmt, um max. 128 Byte	3	2
D8 bis DF	DJNZ	Rn,rel	Dekrementiere das gewählte Register (R0 bis R7) und springe um max. 128 Byte, wenn anschließend noch keine Null im Register steht	2	2
D5	DJNZ	direct,rel	Dekrementiere den angegebenen RAM-Speicherplatz und springe um max. 128 Byte, wenn anschließend der Inhalt noch größer als Null ist	3	2
40	JC	rel	Springe um max. 128 Byte, wenn das Carry-Bit auf HIGH steht (also gesetzt ist)	2	2
50	JNC	rel	Springe um max. 128 Byte, wenn das Carry-Bit auf LOW steht (also gelöscht ist)	2	2
20	JB	bit,rel	Springe um max. 128 Byte, wenn das angegebene Bit auf HIGH steht	3	2
30	JNB	bit,rel	Springe um max. 128 Byte, wenn das angegebene Bit auf LOW steht	3	2
10	JBC	bit,rel	Springe um max. 128 Byte, wenn das angegebene Bit auf HIGH steht und lösche anschließend dieses Bit	3	2
00	NOP		Keine Wirkung (ergibt nur Zeitverzögerung um 1 Maschinenzklus, z.B. 1 Mikrosekunde)	1	1

Logische Befehle

Die Gruppe umfasst die Befehle UND (ANL), ODER (ORL), EXKLUSIV-ODER (XRL), und INVERTIERUNG (CPL). Ebenso fallen darunter elementare Operationen wie „CLR A“ (= lösche den Akku), Rotations- und Vertauschungsbefehle.

Hex-Code	Mnemonik		Beschreibung	Byte	Masch.- Zyklen
58 bis 5F	ANL	A,Rn	Akku UND gewähltes Register (R0 bis R7)	1	1
55	ANL	A,direct	Akku UND direkt angegebener RAM-Speicherplatz (Ergebnis im Akku)	2	1
56	ANL	A,@R0	Akku UND indirekt durch R0 adressierter Speicherplatz	1	1
57	ANL	A,@R1	Akku UND indirekt durch R1 adressierter Speicherplatz	1	1
54	ANL	A,#data	Akku UND vorgegebene Konstante	2	1
52	ANL	direct,A	Direkt angewählter RAM-Platz UND Akku (Ergebnis im RAM-Speicherplatz)	2	1
53	ANL	direct,#data	Direkt angewählter RAM-Platz UND Konstante (Ergebnis im RAM-Speicherplatz)	3	2
48 bis 4F	ORL	A,Rn	Akku ODER gewähltes Register	1	1
45	ORL	A,direct	Akku ODER direkt angegebener RAM-Speicherplatz (Ergebnis im Akku)	2	1
46	ORL	A,@R0	Akku ODER indirekt durch R0 adressierter Speicherplatz	1	1
47	ORL	A,@R1	Akku ODER indirekt durch R1 adressierter Speicherplatz	1	1
44	ORL	A,#data	Akku ODER vorgegebene Konstante	2	1
42	ORL	direct,A	Direkt angewählter RAM-Platz ODER Akku (Ergebnis im RAM-Speicherplatz)	2	1
43	ORL	direct,#data	Direkt angewählter RAM-Platz ODER Konstante (Ergebnis im RAM-Speicherplatz)	3	2
68 bis 6F	XRL	A,Rn	Akku EXKLUSIV ODER gewähltes Register (R0 bis R7)	1	1
65	XRL	A,direct	Akku EXKLUSIV ODER direkt angegebener RAM-Speicherplatz (Ergebnis im Akku)	2	1
66	XRL	A,@R0	Akku EXKLUSIV ODER indirekt durch R0 adressierter Speicherplatz	1	1
67	XRL	A,@R1	Akku EXKLUSIV ODER indirekt durch R1 adressierter Speicherplatz	1	1
64	XRL	A,#data	Akku EXKLUSIV ODER vorgegebene Konstante	2	1
62	XRL	direct,A	Direkt angewählter RAM-Speicherplatz EXKLUSIV ODER Akku (Ergebnis im RAM-Speicherplatz)	2	1
63	XRL	direct,#data	Direkt angewählter RAM-Speicherplatz EXKLUSIV ODER vorgegebene Konstante (Ergebnis im RAM-Speicherplatz)	3	2
E4	CLR	A	Lösche den Akkumulatorinhalt	1	1
F4	CPL	A	Komplementiere den Akkuinhalt	1	1
23	RL	A	Rotiere den Akkuinhalt (um 1 Schritt) nach links	1	1
33	RLC	A	Rotiere den Akkuinhalt (um 1 Schritt) über das	1	1

			Carry-Bit nach links		
03	RR	A	Rotiere den Akkuinhalt (um 1 Schritt) nach rechts	1	1
13	RRC	A	Rotiere den Akkuinhalt (um 1 Schritt) über das Carry-Bit nach rechts	1	1
C4	SWAP	A	Vertausche die beiden Nibbles des Akkumulators	1	1

Datentransfer-Befehle

Diese Gruppe enthält alle MOV-Varianten (sowie die Ableger MOVX und MOVC).

Es gilt: 1) Mit MOV werden Registerinhalte irgendwo hinbewegt.

2) Mit MOVX wird aus dem externen Datenspeicher (RAM) gelesen oder in ihn etwas hineingeschrieben

3) Mit MOVC wird ein Code-Byte aus dem Programmspeicher in den Akku geholt und dort wie eine normale Hex-Zahl behandelt oder verarbeitet (Anwendung: Einsatz von gespeicherten Texten oder Tabellen).

Mit den Befehlen PUSH und POP lassen sich Registerinhalte auf den „Stack“ retten bzw. wieder von dort holen. Mit XCH können Registerinhalte vertauscht werden

Hex-Code	Mnemonik		Beschreibung	Byte	Masch.- Zyklen
E8 bis EF	MOV	A,Rn	Hole den Registerinhalt (R0.....R7) in den Akku	1	1
E5	MOV	A,direct	Hole den Inhalt des direkt adressierten RAM-Platzes in den Akku	2	1
E6	MOV	A,@R0	Hole den Inhalt der indirekt durch R0 adressierten Speicherzelle in den Akku	1	1
E7	MOV	A,@R1	Hole den Inhalt der indirekt durch R1 adressierten Speicherzelle in den Akku	1	1
74	MOV	A,#data	Lade den Akku mit der gewünschten Konstante	2	1
F8 bis FF	MOV	Rn,A	Speichere den Akkuinhalt im ausgewählten Register (R0.....R7)	1	2
A8 bis AF	MOV	Rn,direct	Lade das ausgewählte Register (R0.....R7) mit dem Inhalt des direkt adressierten RAM-Speicherplatzes	2	2
78 bis 7F	MOV	Rn,#data	Lade das gewählte Register (R0.....R7) mit der gewünschten Konstante	2	1
F5	MOV	direct,A	Speichere den Akkuinhalt im direkt adressierten RAM-Speicherplatz	2	1
88 bis 8F	MOV	direct,Rn	Speichere den ausgewählten Registerinhalt (R0.....R7) im direkt adressierten RAM-Speicherplatz	2	2
85	MOV	direct,direct	Speichere den Inhalt eines direkt adressierten RAM-Platzes in einem anderen direkt adressierbaren RAM-Platz	3	2
86	MOV	direct,@R0	Speichere den Inhalt der indirekt durch R0 adressierten Speicherzelle im direkt adressierten RAM-Platz	2	2
87	MOV	direct,@R1	Speichere den Inhalt der indirekt durch R1 adressierten Speicherzelle im direkt adressierten RAM-Platz	2	2
75	MOV	direct,#data	Lade den direkt adressierten RAM-Platz mit der vorgegebenen Konstanten	3	2
F6	MOV	@R0,A	Speichere den Akkuinhalt in der indirekt durch R0 adressierten Speicherzelle	1	1

F7	MOV	@R1,A	Speichere den Akkuinhalt in der indirekt durch R1 adressierten Speicherzelle	1	1
A6	MOV	@R0,direct	Speichere den Inhalt des direkt adressierten RAM-Platzes in die indirekt durch R0 adressierte Speicherzelle	2	2
A7	MOV	@R1,direct	Speichere den Inhalt des direkt adressierten RAM-Platzes in die indirekt durch R1 adressierte Speicherzelle	2	2
76	MOV	@R0,#data	Lade die indirekt durch R0 adressierte Speicherzelle mit der vorgegebenen Konstanten	2	1
77	MOV	@R1,#data	Lade die indirekt durch R1 adressierte Speicherzelle mit der vorgegebenen Konstanten	2	1
90	MOV	DPTR,#data16	Lade den Datenpointer mit einer 16 Bit-Konstante	3	2
93	MOVC	A,@A + DPTR	Lade den Akku mit einem Programmcode (z. B. abgespeicherter Tabellenwert). Die Code-Adresse erhält man als Summe aus Akkuinhalt und DPTR-Inhalt	1	2
83	MOVC	A,@A + PC	Lade den Akku mit einem Programmcode (z. B. abgespeicherter Tabellenwert). Die Code-Adresse erhält man als Summe aus Akkuinhalt und PC-Inhalt	1	2
E2	MOVX	A,@R0	Hole ein Byte aus dem externen Datenspeicher (RAM).Die Adresse steht in Register R0	1	2
E3	MOVX	A,@R1	Hole ein Byte aus dem externen Datenspeicher (RAM).Die Adresse steht in Register R1	1	2
E0	MOVX	A,@DPTR	Hole ein Byte aus dem externen Datenspeicher (RAM).Die Adresse steht im Datenpointer	1	2
F2	MOVX	@R0,A	Speichere den Akkuinhalt im externen Datenspeicher (RAM). Die Adresse steht im Register R0	1	2
F3	MOVX	@R1,A	Speichere den Akkuinhalt im externen Datenspeicher (RAM). Die Adresse steht im Register R1	1	2
F0	MOVX	@DPTR,A	Speichere den Akkuinhalt im externen Datenspeicher (RAM). Die Adresse steht im DPTR	1	2
C0	PUSH	direct	Lege den Inhalt des direkt adressierten RAM-Platzes auf dem Stack ab	2	2
D0	POP	direct	Schreibe vom Stack zurück in direkt adressierten RAM-Platz	2	2
C8 bis CF	XCH	A,Rn	Vertausche die Inhalte von Akku und gewähltem Register (R0.....R7)	1	1
C5	XCH	A,direct	Vertausche die Inhalte von Akku und direkt adressiertem RAM-Platz	2	1
C6	XCH	A,@R0	Vertausche die Inhalte von Akku und indirekt durch R0 adressierter RAM-Speicherzelle	1	1
C7	XCH	A,@R1	Vertausche die Inhalte von Akku und indirekt durch R1 adressierter RAM-Speicherzelle	1	1
D6	XCHD	A,@R0	Vertausche die Low-Nibbles von Akku und der indirekt durch R0 adressierten RAM-Speicherzelle	1	1
D7	XCHD	A,@R1	Vertausche die Low-Nibbles von Akku und der indirekt durch R1 adressierten RAM-Speicherzelle	1	1

Bitmanipulations-Befehle

Mit diesen Befehlen können einzelne Bits in den SFR (Special Function Registers) bzw. im internen, bitmanipulierbaren Bereich von RAM 1 auf dem Chip (Adressen 20h bis 2Fh) gesetzt, gelöscht, invertiert oder logisch verknüpft werden.

Hex-Code	Mnemonik		Beschreibung	Byte	Masch.- Zyklen
C3	CLR	C	Lösche das Carry-Bit	1	1
C2	CLR	bit	Lösche das direkt angesprochene Bit	2	1
D3	SETB	C	Setze das Carry-Bit	1	1
D2	SETB	bit	Setze das direkt angesprochene Bit	2	1
B3	CPL	C	Komplementiere das Carry-Bit	1	1
B2	CPL	bit	Komplementiere das direkt angesprochene Bit	2	1
82	ANL	C,bit	Bilde: Carry UND direkt angesprochenes Bit (Ergebnis: als neues Carry-Bit)	2	2
B0	ANL	C,/bit	Bilde: Carry UND invertiertes, direkt angesprochenes, Bit. (Ergebnis: als neues Carry-Bit)	2	2
72	ORL	C,bit	Bilde: Carry ODER direkt angesprochenes Bit (Ergebnis: als neues Carry-Bit)	2	2
A0	ORL	C,/bit	Bilde: Carry ODER invertiertes, direkt angesprochenes Bit. (Ergebnis: als neues Carry-Bit)	2	2
A2	MOV	C,bit	Benütze das direkt angesprochene Bit als neues Carry-Bit	2	1
92	MOV	bit,C	Schreibe das Carry-Bit in den Platz des direkt angesprochenen Bits	2	2