

## Befehlsliste der 8051-Familie / Arithmetik-Befehle

Mnemonic	Funktion	Bytes	MZ	Flags
<b>ADD A,Rr</b>	Der Akkumulator wird überschrieben durch das Ergebnis der Addition von Akku und dem Inhalt von Rr.	1	1	CY,AC,OV,P
<b>ADD A,dadr</b>	Der Akkumulator wird überschrieben durch das Ergebnis der Addition von Akku und dem Inhalt von dadr.	2	1	CY,AC,OV,P
<b>ADD A,@Ri</b>	Der Akkumulator wird überschrieben durch das Ergebnis der Addition von Akku und dem Inhalt der internen Datenspeicherzelle, die durch Ri adressiert wird.	1	1	CY,AC,OV,P
<b>ADD A,#const8</b>	Der Akkumulator wird überschrieben durch das Ergebnis der Addition von Akku und dem Inhalt der 8-Bit-Konstanten.	2	1	CY,AC,OV,P
<b>ADDC A,Rr</b>	Der Akkumulator wird überschrieben durch das Ergebnis der Addition von Akku, Rr und dem Inhalt des Carry-Flag.	1	1	CY,AC,OV,P
<b>ADDC A, dadr</b>	Der Akkumulator wird überschrieben durch das Ergebnis der Addition von Akku, dadr und dem Inhalt des Carry-Flag.	2	1	CY,AC,OV,P
<b>ADDC A, @Ri</b>	Der Akkumulator wird überschrieben durch das Ergebnis der Addition von Akku, dem Inhalt der internen Datenspeicherzelle, die durch Ri adressiert wird und dem Inhalt des Carry-Flag.	1	1	CY,AC,OV,P
<b>ADDC A, #const8</b>	Der Akkumulator wird überschrieben durch das Ergebnis der Addition von Akku, dem Inhalt der 8-Bit-Konstanten und dem Inhalt des Carry-Flag.	2	1	CY,AC,OV,P
<b>SUBB A,Rr</b>	Der Akkumulator wird überschrieben durch das Ergebnis der Subtraktion von Akku und der Summe der Inhalte von Rr und Carry-Flag.	1	1	CY,AC,OV,P
<b>SUBB A,dadr</b>	Der Akkumulator wird überschrieben durch das Ergebnis der Subtraktion von Akku und der Summe der Inhalte von dadr und Carry-Flag.	2	1	CY,AC,OV,P
<b>SUBB A,@Ri</b>	Der Akkumulator wird überschrieben durch das Ergebnis der Subtraktion von Akku und der Summe der Inhalte von Carry-Flag und dem Inhalt der internen Datenspeicherzelle, die durch Ri adressiert wird.	1	1	CY,AC,OV,P
<b>SUBB A,#const8</b>	Der Akkumulator wird überschrieben durch das Ergebnis der Subtraktion von Akku und der Summe der Inhalte von Carry-Flag und der 8-Bit-Konstanten.	2	1	CY,AC,OV,P
<b>INC A</b>	Zum Inhalt des Akkumulators wird inkrementiert.	1	1	P
<b>INC Rr</b>	Der Inhalt des Registers Rr wird inkrementiert.	1	1	-
<b>INC dadr</b>	Der Inhalt der Speicherzelle dadr wird inkrementiert.	2	1	-
<b>INC @Ri</b>	Der Inhalt der Speicherzellen des internen Datenspeichers, der durch Ri adressiert ist, wird inkrementiert.	1	1	-
<b>INC DPTR</b>	Der Inhalt des Datenpointers wird inkrementiert.	1	2	-
<b>DEC</b>	Der Inhalt des Akkumulators wird dekrementiert	1	1	P
<b>DEC Rr</b>	Der Inhalt des Registers Rr wird dekrementiert.	1	1	-

<b>DEC dadr</b>	Der Inhalt der Speicherzelle dadr wird dekrementiert.	2	1	-
<b>DEC @Ri</b>	Der Inhalt der Speicherzelle des internen Datenspeichers, die durch Ri adressiert ist, wird dekrementiert.	1	1	-
<b>MUL AB</b>	Die Inhalte von Akkumulator und Register B werden multipliziert. Das niederwertige Byte des Produktes steht im Akku, das höherwertige Byte im Register B. Das Carry-Flag wird gelöscht und das OV-Flag gesetzt, wenn das Ergebnis im Reg. B ungleich Null ist.	1	4	CY,OV,P
<b>DIV AB</b>	Die Inhalte von Akkumulator und Register B werden dividiert. Der Quotienten steht im Akku, der Divisionsrest im Register B. Das Carry-Flag wird gelöscht. Bei einer Division durch Null wird das OV-Flag gesetzt.	1	4	CY,OV,P
<b>DA A</b>	Das Ergebnis einer vorangegangenen Addition zweier BCD-Zahlen steht im Akku und wird unter Berücksichtigung von Carry und Hilfsarry korrigiert.	1	1	CY,P

### Befehlsliste der 8051-Familie / Logik-Befehle

<b>Mnemonic</b>	<b>Funktion</b>	<b>Bytes</b>	<b>MZ</b>	<b>Flags</b>
<b>ANL A,Rr</b>	Der Akkumulator wird überschrieben durch das Ergebnis der UND-Verknüpfung von Akku und dem Inhalt von Rr.	1	1	P
<b>ANL A,dadr</b>	Der Akkumulator wird überschrieben durch das Ergebnis der UND-Verknüpfung von Akku und dem Inhalt von dadr.	2	1	P
<b>ANL A,@Ri</b>	Der Akkumulator wird überschrieben durch das Ergebnis der UND-Verknüpfung von Akku und dem Inhalt der internen Datenspeicherzelle, die durch den Inhalt von Ri adressiert wird.	1	1	P
<b>ANL A,#const8</b>	Der Akkumulator wird überschrieben durch die UND-Verknüpfung von Akku und der 8-Bit-Konstanten.	2	1	P
<b>ANL dadr,A</b>	Der Inhalt von dadr wird überschrieben durch die UND-Verknüpfung von Akku und dem Inhalt von dadr.	2	1	-
<b>ANL dadr,#const8</b>	Der Inhalt von dadr wird überschrieben durch die UND-Verknüpfung der 8-Bit-Konstanten und dem Inhalt von dadr.	3	2	-
<b>ORL A,Rr</b>	Der Akkumulator wird überschrieben durch die ODER-Verknüpfung von Akku und dem Inhalt von Rr.	1	1	P
<b>ORL A,dadr</b>	Der Akkumulator wird überschrieben durch die ODER-Verknüpfung von Akku und dem Inhalt von dadr.	2	1	P
<b>ORL A,@Ri</b>	Der Akkumulator wird überschrieben durch die ODER-Verknüpfung von Akku und dem Inhalt der internen Datenspeicherzelle, die durch den Inhalt von Ri adressiert wird.	1	1	P
<b>ORL A,#const8</b>	Der Akkumulator wird überschrieben durch das Ergebnis der ODER-Verknüpfung von Akku und der 8-Bit-Konstanten.	2	1	P
<b>ORL dadr,A</b>	Der Inhalt von dadr wird überschrieben durch die ODER-Verknüpfung von Akkumulator und dem Inhalt von dadr.	2	1	-

<b>ORL dadr,#const8</b>	Der Inhalt von dadr wird überschrieben durch die ODER-Verknüpfung der 8-Bit-Konstanten und dem Inhalt von dadr.	3	2	-
<b>XRL A,Rr</b>	Der Akkumulator wird überschrieben durch das Ergebnis der EXCLUSIVE-ODER-Verknüpfung von Akku und dem Inhalt von Rr.	1	1	P
<b>XRL A,dadr</b>	Der Akkumulator wird überschrieben durch das Ergebnis der EXCLUSIVE-ODER-Verknüpfung von Akku und dem Inhalt von dadr.	2	1	P
<b>XRL A,@Ri</b>	Der Akkumulator wird überschrieben durch die EXCLUSIVE-ODER-Verknüpfung von Akku und dem Inhalt der internen Datenspeicherzelle, die durch den Inhalt von Ri adressiert wird.	1	1	P
<b>XRL A,#const8</b>	Der Akkumulator wird überschrieben durch das Ergebnis der EXCLUSIVE-ODER-Verknüpfung von Akku und der 8-Bit-Konstanten.	2	1	P
<b>XRL dadr,A</b>	Der Inhalt von dadr wird überschrieben durch die EXCLUSIVE-ODER-Verknüpfung von Akkumulator und dem Inhalt von dadr.	2	1	-
<b>XRL dadr,#const8</b>	Der Inhalt von dadr wird überschrieben durch die EXCLUSIVE-ODER-Verknüpfung der 8-Bit-Konstanten und dem Inhalt von dadr.	3	2	-
<b>CLR A</b>	Der Akkumulator wird gelöscht.	1	1	P
<b>CPL A</b>	Der Inhalt des Akkumulators wird invertiert.	1	1	P

### Befehlsliste der 8051-Familie / Transport-Befehle

<b>Mnemonic</b>	<b>Funktion</b>	<b>Bytes</b>	<b>MZ</b>	<b>Flags</b>
<b>MOV A,Rr</b>	Lade den Akkumulator mit dem Inhalt von Rr.	1	1	P
<b>MOV A,dadr</b>	Lade den Akkumulator mit dem Inhalt von dadr.	2	1	P
<b>MOV A,@Ri</b>	Lade den Akkumulator mit dem Inhalt der internen Speicherzelle, die durch Ri adressiert wird.	1	1	P
<b>MOV A,#konst8</b>	Lade den Akkumulator mit der 8-Bit-Konstanten.	2	1	P
<b>MOV Rr,A</b>	Lade Register Rr mit dem Inhalt des Akkumulators.	1	1	-
<b>MOV Rr,dadr</b>	Lade Register Rr mit dem Inhalt von dadr.	2	2	-
<b>MOV Rr,#konst8</b>	Lade Register Rr mit der 8-Bit-Konstanten.	2	1	-
<b>MOV dadr,A</b>	Lade die interne Datenspeicherzelle mit dem Inhalt des Akkumulators.	2	1	-
<b>MOV dadr,Rr</b>	Lade die interne Datenspeicherzelle mit dem Inhalt des Registers Rr.	2	2	-
<b>MOV dadr,dadr</b>	Lade die interne Datenspeicherzelle dadr mit dem Inhalt der internen Speicherzelle dadr.	3	2	-
<b>MOV dadr,@Ri</b>	Lade die interne Datenspeicherzelle dadr mit dem Inhalt der internen Speicherzelle, die durch Ri adressiert ist.	2	2	-

<b>MOV dadr,#konst8</b>	Lade die interne Datenspeicherzelle dadr mit der 8-Bit-Konstanten.	3	2	-
<b>MOV @Ri,A</b>	Lade in die interne Datenspeicherzelle, die durch Ri adressiert wird, den Inhalt des Akkumulators.	1	1	-
<b>MOV @Ri,dadr</b>	Lade in die interne Datenspeicherzelle, die durch Ri adressiert wird, den Inhalt von dadr.	2	2	-
<b>MOV @Ri,#konst8</b>	Lade in die interne Datenspeicherzelle, die durch Ri adressiert wird, die 8-Bit-Konstante.	2	1	-
<b>MOV DPTR,#konst16</b>	Lade den Daten-Pointer mit der 16-Bit-Konstanten.	3	2	-
<b>MOVC A,@A+DPTR</b>	Lade den Akkumulator mit dem Inhalt der Programmspeicherzelle, die durch die Summe von Daten-Pointer und Akkumulator adressiert wird.	1	2	-
<b>MOVC A,@A+PC</b>	Lade den Akkumulator mit dem Inhalt der Programmspeicherzelle, die durch die Summe von Program-Counter und Akkumulator adressiert wird.	1	2	P
<b>MOVX A,@Ri</b>	Lade den Akkumulator mit dem Inhalt der externen Speicherzelle, die durch Ri adressiert wird.	1	2	P
<b>MOVX A,@DPTR</b>	Lade den Akku mit dem Inhalt der externen Datenspeicherzelle, die durch den Daten-Pointer adressiert wird.	1	2	P
<b>MOVX @Ri,A</b>	Lade die externe Datenspeicherzelle, die durch Ri adressiert wird, mit dem Inhalt des Akkumulators.	1	2	-
<b>MOVX @DPTR,A</b>	Lade die ext. Datenspeicherzelle, die durch den Daten-Pointer adressiert ist, mit dem Inhalt des Akkus.	1	2	-
<b>PUSH dadr</b>	Der Stack-Pointer wird um 1 erhöht und der Inhalt von dadr im Stack abgelegt.	2	2	-
<b>POP dadr</b>	Der Inhalt der durch den Stack-Pointer adressierten Adresse wird nach dadr transferiert und der Stack-Pointer um 1 erniedrigt.	2	2	-
<b>XCH A,Rr</b>	Vertausche die Inhalte des Akkumulators und des Registers Rr.	1	1	P
<b>XCH A,dadr</b>	Vertausche die Inhalte des Akkumulators und der internen Datenspeicherzelle dadr.	2	1	P
<b>XCH A,@Ri</b>	Vertausche die Inhalte des Akkus und der internen Datenspeicherzelle, die durch Ri adressiert wird.	1	1	P
<b>XCHD A,@Ri</b>	Vertausche die Inhalte der niederwertigen Halbbytes von Akku und der internen Datenspeicherzelle, die durch Ri adressiert wird.	1	1	P
<b>SWAP A</b>	Vertausche die Bytehälften des Akkumulators	1	1	-
<b>NOP</b>	Lehrbefehl	1	1	-

## Befehlsliste der 8051-Familie / Bitverarbeitungs-Befehle

Mnemonic	Funktion	Bytes	MZ	Flags
----------	----------	-------	----	-------

<b>CLR C</b>	Lösche das Carry-Flag.	1	1	CY
<b>CLR badr</b>	Lösche den Inhalt von badr.	2	1	-
<b>SETB C</b>	Setze das Carry-Flag.	1	1	CY
<b>SETB badr</b>	Setze den Inhalt von badr.	2	1	-
<b>CPL C</b>	Invertiere den Inhalt des C-Flag.	1	1	CY
<b>CPL badr</b>	Invertiere den Inhalt von badr.	2	1	-
<b>ANL C,badr</b>	Das Carry-Flag wird überschrieben durch das Ergebnis der UND-Verknüpfung von Carry-Bit und dem Inhalt von badr.	2	2	CY
<b>ANL C,/badr</b>	Das Carry-Flag wird überschrieben durch das Ergebnis der UND-Verknüpfung von Carry-Bit und dem invertierten Inhalt von badr.	2	2	CY
<b>ORL C,badr</b>	Das Carry-Flag wird überschrieben durch das Ergebnis der ODER-Verknüpfung von Carry-Bit und dem Inhalt von badr.	2	2	CY
<b>ORL C,/badr</b>	Das Carry-Flag wird überschrieben durch das Ergebnis der ODER-Verknüpfung von Carry-Bit und dem invertierten Inhalt von badr.	2	2	CY
<b>MOV C,badr</b>	Lade in das Carry-Bit den Inhalt von badr.	2	2	CY
<b>MOV badr,C</b>	Lade in die badr den Inhalt des Carry-Bit.	2	2	-

### Befehlsliste der 8051-Familie / Schiebe-Befehle

Mnemonic	Funktion	Bytes	MZ	Flags
<b>RL A</b>	Verschiebe den Inhalt des Akkus um eine Stelle nach links. Die höchstwertige Bitstelle wird in die niederwertigste Bitstelle verschoben.	1	1	-
<b>RLC A</b>	Verschiebe den Inhalt des Akkus um eine Stelle nach links über das Carry-Flag. Der Inhalt des Carry-Flag wird in die niederwertigste Bitstelle verschoben.	1	1	CY,P
<b>RR A</b>	Verschiebe den Inhalt des Akkus um eine Stelle nach rechts. Die niederwertigste Bitstelle wird in die höchstwertigste Bitstelle verschoben.	1	1	-
<b>RRC A</b>	Verschiebe den Inhalt des Akkus um eine Stelle nach rechts über das Carry-Flag. Die niederwertigste Bitstelle wird in das Carry-Flag verschoben.	1	1	CY,P

### Befehlsliste der 8051-Familie / Unterprogramm-Befehle

Mnemonic	Funktion	Bytes	MZ	Flags
<b>ACALL adr11</b>	Unterprogrammaufruf innerhalb einer 2 k-Byte-Seite.	3	2	-
<b>LCALL adr16</b>	Unterprogrammaufruf .	2	2	-
<b>RET</b>	Rücksprung aus einem Unterprogramm.	1	2	-

<b>RETI</b>	Rücksprung aus einer Interrupt-Service-Routine.	1	2	-
-------------	---	---	---	---

## Befehlsliste der 8051-Familie / Sprung-Befehle

<b>Mnemonic</b>	<b>Funktion</b>	<b>Bytes</b>	<b>MZ</b>	<b>Flags</b>
<b>AJMP adr11</b>	Setze das Programm bei adr11 innerhalb der 2 kByte-Seite fort.	2	2	-
<b>LJMP adr16</b>	Setze das Programm bei adr16 fort.	3	2	-
<b>SJMP rel</b>	Setze das Programm bei rel fort (relativ zum Programm-Counter)	2	2	-
<b>JMP @A+DPTR</b>	Setze das Programm an der Stelle fort, die sich aus der Summe von Akkumulator und DPTR ergibt.	1	2	-
<b>JZ rel</b>	Springe relativ zur Adresse rel, wenn der Inhalt des Akkus gleich null ist.	2	2	-
<b>JNZ rel</b>	Springe relativ zur Adresse rel, wenn der Inhalt des Akkus ungleich null ist.	2	2	-
<b>JC rel</b>	Springe relativ zur Adresse rel, wenn der Inhalt des Carry-Flag gesetzt ist.	2	2	-
<b>JNC rel</b>	Springe relativ zur Adresse rel, wenn der Inhalt des Carry-Flag nicht gesetzt ist .	2	2	-
<b>JB badr, rel</b>	Springe relativ zur Adresse rel, wenn der Inhalt von badr gleich eins ist.	3	2	-
<b>JNB badr,rel</b>	Springe relativ zur Adresse rel, wenn der Inhalt von badr gleich null ist.	3	2	-
<b>JBC badr,rel</b>	Springe relativ zur Adresse rel, wenn der Inhalt von badr gleich eins ist und lösche den Inhalt von badr.	3	2	-
<b>CJNE A,dadr,rel</b>	Springe relativ zur Adresse rel, wenn die Inhalte von Akkumulator und dadr ungleich sind.	3	2	CY
<b>CJNE A,#konst8,rel</b>	Springe relativ zur Adresse rel, wenn der Inhalt des Akkus ungleich der 8-Bit-Konstanten ist.	3	2	CY
<b>CJNE Rr,#konst8,rel</b>	Springe relativ zur Adresse rel, wenn der Inhalt des Registers Rr ungleich der 8-Bit-Konstanten ist.	3	2	CY
<b>CJNE @Ri,#konst8,rel</b>	Springe relativ zur Adresse rel, wenn der Inhalt der internen Datenspeicherezelle, die durch Ri adressiert wird, ungleich der 8-Bit-Konstanten ist.	3	2	CY
<b>DJNZ Rr,rel</b>	Der Inhalt von Register Rr wird um eins erniedrigt. Ist dann der Inhalt ungleich null, springe relativ zur Adresse rel.	3	2	-
<b>DJNZ dadr,rel</b>	Der Inhalt von dadr wird um eins erniedrigt. Ist dann der Inhalt ungleich null, springe relativ zur Adresse rel.	3	2	-