

4 Assembler für die 8051-Controller-Familie

Tabelle 4.1 Wichtige Schreibweisen und Bedeutungen für die Befehle

Abkürzung	Bedeutung
#data	Daten sind in der Instruktion codiert - # = Zahl
data addr	Speicheradresse innerhalb des Chips
Rr	$0 \leq r \leq 7$ Allgemein verwendbares Register der Registerbänke
@Rr	$0 \leq r \leq 1$ indirekt adressierbares Register der Registerbänke
code addr	16-Bit-Adresse decodiert als: volle 16-Bit-Adresse 11-Bit-Seitenadresse 8-Bit relativer Offset
A	Akkumulator
C	Carry Flag
bit addr	Bit-Adresse auf dem Chip
/ bit addr	Komplementierter Inhalt der Bit-Adresse
DPTR	Datenzeiger (Data Pointer)
PC	Programmzähler (Program Counter)
AB	Register Paar

Mnemonic	Register	Incoding Binary	Inc. HEX	Affected Flags	Bytes	Cycles	Function	Funktionsbeschreibung
Transfer Instructions Internal RAM								
mov A,Rr	r=0-7	11101rrr	E8+r	P	1	1	Move register Rr contents to accumulator	Lade Akkumulator mit Inhalt von Register Rr
mov Rr,A	r=0-7	11111rrr	F8+r	-	1	1	Move accumulator contents to register Rr	Lade Register Rr mit Inhalt des Akkumulators
mov A,@Ri	i=0-1	1110011i	E6+i	P	1	1	Move contents of data memory addressed by Ri to accumulator	Lade Akkumulator mit dem Inhalt der internen Datenspeicherzelle, die durch den Inhalt von Ri adressiert ist
mov @Ri,A	i=0-1	1111011i	F6+i	-	1	1	Move accumulator to data memory addressed by Ri	Lade interne Datenspeicherzelle, die durch den Inhalt von Ri adressiert ist, mit dem Inhalt des Akkumulators
mov A,dadr*	-	11100101	E5	P	2	1	Move contents of direct address to accumulator (mov A,ACC is not a valid instruction)	Lade Akkumulator mit Inhalt von dadr (mov A,ACC ist kein gültiger Befehl)
mov dadr,A	-	11110101	F5	-	2	1	Move accumulator contents to direct address	Lade dadr mit Inhalt des Akkumulators
mov A,#const8	-	01110100	74	P	2	1	Move immediate data to accumulator	Lade Akkumulator mit 8 Bit Konstante
mov Rr,#const8	r=0-7	01111rrr	78+r	-	2	1	Move immediate data to register	Lade Register Rr mit 8 Bit Konstante
mov @Ri,#const8	i=0-1	0111011i	76+i	-	2	1	Move immediate data to data memory addressed by Ri	Lade interne Datenspeicherzelle, die durch den Inhalt von Ri adressiert ist, mit 8 Bit Konstante
mov dadr,#const8	-	01110101	75	-	3	2	Move immediate data to data memory	Lade dadr mit 8 Bit Konstante
mov Rr,dadr	r=0-7	10101rrr	A8+r	-	2	2	Load register Rr with contents of direct address	Lade Register Rr mit Inhalt von dadr
mov dadr,Rr	r=0-7	10001rrr	88+r	-	2	2	Load direct address with contents of register	Lade dadr mit Inhalt von Register Rr
mov dadr,@Ri	i=1-0	1000011i	86+i	-	2	2	Move contents of data memory addressed by Ri to direct address	Lade dadr mit Inhalt der Datenspeicherzelle, die durch Ri adressiert ist
mov @Ri,dadr	i=0-1	1010011i	A6+i	-	2	2	Move data memory to contents of data memory addressed by Ri	Lade interne Datenspeicherzelle, die durch Ri adressiert ist, mit Inhalt von dadr
mov dadr1,dadr2	-	10000101	85	-	3	2	Move data memory to data memory	Lade dadr1 mit Inhalt von dadr2
mov DPTR,#const16	-	10010000	90	-	3	2	Move immediate data to data pointer	Lade Daten-Peinter DPTR mit 16 Bit Konstante
Transfer Instructions External RAM								
movx A,@Ri	i=0-1	1110001i	E2+r	P	1	2	Move contents of external data memory addressed by Ri to accumulator	Lade Akkumulator mit dem Inhalt der externen Datenspeicherzelle, die durch Ri adressiert ist
movx @Ri,A	i=0-1	1111001i	F2+i	-	1	2	Move accumulator contents to external data memory, addressed by Ri	Lade externe Datenspeicherzelle, die durch den Inhalt von Ri adressiert ist, mit dem Inhalt des Akkumulators
movx A,@DPTR	-	11100000	E0	P	1	2	Move contents of external data memory addressed by DPTR to accumulator	Lade Akkumulator mit dem Inhalt der externen Datenspeicherzelle, die durch DPTR adressiert ist
movx @DPTR,A	-	11110000	F0	-	1	2	Move accumulator contents to external data memory, addressed by DPTR	Lade externe Datenspeicherzelle, die durch den Inhalt von DPTR adressiert ist, mit dem Inhalt des Akkumulators
Transfer Instructions Internal RAM								
xch A,Rr	r=0-7	11001rrr	C8+r	P	1	1	Exchange accumulator and register contents	Vertausche die Inhalte von Akkumulator und Register Rr
xch A,dadr	-	11000101	C5	P	2	1	Exchange accumulator and data memory contents	Vertausche die Inhalte von Akkumulator und dadr
xch A,@Ri	i=0-1	1100011i	C6+i	P	1	1	Exchange accumulator and data memory addressed by Ri	Vertausche die Inhalte von Akkumulator und der internen Datenspeicherzelle, die durch den Inhalt von Ri adressiert ist
xchd A,@Ri	i=0-1	1101011i	D6+i	P	1	1	Exchange low order nibble of accumulator and data memory	Vertausche die Inhalte der niederwertigen Halbbytes von Akkumulator und der internen Datenspeicherzelle, die durch den Inhalt von Ri adressiert ist
swap A	-	11000100	C4	-	1	1	Exchange nibbles of accumulators	Vertausche die Bytehälften des Akkumulators
push dadr	-	11000000	C0	-	2	2	Increment stack pointer. Push data memory onto stack	Der Stack-Peinter wird um 1 erhöht. Der Inhalt von dadr wird im Stack abgelegt
pop dadr	-	11010000	D0	-	2	2	Pop data memory form stack. Decrement stack pointer	Übertrage den Inhalt der durch den Stack-Peinter adressierten Datenspeicherzelle nach dadr; der Stack-Peinter wird um 1 erniedrigt

4 Assembler für die 8051-Controller-Familie 63

Mnemonic	Register	Incoding Binary	Inc. HEX	Affected Flags	Bytes	Cycles	Function	Funktionsbeschreibung
Transfer Instructions ROM								
movc A,@A+DPTR	-	10010011	93	P	1	2	Move program memory address by accumulator plus DPTR to accumulator	Lade Akkumulator mit dem Inhalt der Programmspeicherzelle, die durch die Summe von DPTR und Akkumulator adressiert ist
movc A,@A+PC	-	10000011	83	P	1	2	Move program memory address by accumulator plus program counter to accumulator	Lade Akkumulator mit dem Inhalt der Programmspeicherzelle, die durch die Summe von Programmzähler und Akkumulator adressiert ist
Boolean Instructions AND								
Mnemonic	Register	Incoding Binary	Inc. HEX	Affected Flags	Bytes	Cycles	Function	Funktionsbeschreibung
anl A,Rr	r=0-7	01011rrr	58+r	P	1	1	Logical AND accumulator with register contents to accumulator	Der Akkumulator wird überschrieben durch das Ergebnis der UND-Verknüpfung von Akkumulator und dem Inhalt von Register Rr
anl A,@Ri	i=0-1	0101011i	56+1	P	1	1	Logical AND accumulator with data memory contents addressed by Ri to accumulator	Der Akkumulator wird überschrieben durch das Ergebnis der UND-Verknüpfung von Akkumulator und dem Inhalt der internen Datenspeicherzelle, die durch den Inhalt von Ri adressiert ist
anl A,dadr	-	01010101	55	P	2	1	Logical AND accumulator with contents of direct address to accumulator	Der Akkumulator wird überschrieben durch das Ergebnis der UND-Verknüpfung von Akkumulator und dem Inhalt von dadr
anl A,#const8	-	01010100	54	P	2	1	Logical AND accumulator with immediate data to accumulator	Der Akkumulator wird überschrieben durch das Ergebnis der UND-Verknüpfung von Akkumulator und der 8 Bit Konstanten
anl dadr,A	-	01010010	52	-	2	1	Logical AND accumulator with contents of direct address to direct address	Der Inhalt von dadr wird überschrieben durch das Ergebnis der UND-Verknüpfung von Akkumulator und dem Inhalt von dadr
anl dadr,#const8	-	01010011	53	-	3	2	Logical AND contents of direct address and immediate data to direct address	Der Inhalt von dadr wird überschrieben durch das Ergebnis der UND-Verknüpfung der 8 Bit Konstanten mit dem Inhalt von dadr
Boolean Instructions OR								
orl A,Rr	r=0-7	01001rrr	48+r	P	1	1	Logical OR accumulator with register contents to accumulator	Der Akkumulator wird überschrieben durch das Ergebnis der ODER-Verknüpfung von Akkumulator und dem Inhalt von Register Rr
orl A,@Ri	i=0-1	0100011i	46+1	P	1	1	Logical OR accumulator with data memory contents addressed by Ri to accumulator	Der Akkumulator wird überschrieben durch das Ergebnis der ODER-Verknüpfung von Akkumulator und dem Inhalt der internen Datenspeicherzelle, die durch den Inhalt von Ri adressiert ist
orl A,dadr	-	01000101	45	P	2	1	Logical OR accumulator with contents of direct address to accumulator	Der Akkumulator wird überschrieben durch das Ergebnis der ODER-Verknüpfung von Akkumulator und dem Inhalt von dadr
orl A,#const8	-	01000100	44	P	2	1	Logical OR accumulator with immediate data to accumulator	Der Akkumulator wird überschrieben durch das Ergebnis der ODER-Verknüpfung von Akkumulator und der 8 Bit Konstanten
orl dadr,A	-	01000010	42	-	2	1	Logical OR accumulator with contents of direct address to direct address	Der Inhalt von dadr wird überschrieben durch das Ergebnis der ODER-Verknüpfung von Akkumulator und dem Inhalt von dadr
orl dadr,#const8	-	01000011	43	-	3	2	Logical OR contents of direct address and immediate data to direct address	Der Inhalt von dadr wird überschrieben durch das Ergebnis der ODER-Verknüpfung der 8 Bit Konstanten und dem Inhalt von dadr
Boolean Instructions EXCLUSIVE OR								
xrl A,Rr	r=0-7	01101rrr	68+r	P	1	1	Logical XOR accumulator with register contents to accumulator	Der Akkumulator wird überschrieben durch das Ergebnis der EXODER-Verknüpfung von Akkumulator und dem Inhalt von Register Rr
xrl A,@Ri	i=0-1	0110011i	66+1	P	1	1	Logical XOR accumulator with data memory contents addressed by Ri to accumulator	Der Akkumulator wird überschrieben durch das Ergebnis der EXODER-Verknüpfung von Akkumulator und dem Inhalt der internen Datenspeicherzelle, die durch den Inhalt von Ri adressiert ist

Mnemonic	Register	Incoding Binary	Inc. HEX	Affected Flags	Bytes	Cycles	Function	Funktionsbeschreibung
xrl A,#const8	-	01100100	64	P	2	1	Logical XOR accumulator with immediate data to accumulator	Der Akkumulator wird überschrieben durch das Ergebnis der EXODER-Verknüpfung von Akkumulator und der 8 Bit Konstanten
xrl dadr,A	-	01100010	62	-	2	1	Logical XOR accumulator with contents of direct address to direct address	Der Inhalt von dadr wird überschrieben durch das Ergebnis der EXODER-Verknüpfung von Akkumulator und dem Inhalt von dadr
xrl dadr,#const8	-	01100011	63	-	3	2	Logical XOR contents of direct address and immediate data to direct address	Der Inhalt von dadr wird überschrieben durch das Ergebnis der EXODER-Verknüpfung der 8 Bit Konstanten und dem Inhalt von dadr

Boolean Instructions COMPLEMENT / CLEAR ACC.

cpl A	-	1110100	F4	P	1	1	Complement accumulator	Komplementiere Akkumulator (Einer Komplement)
clr A	-	11100100	E4	P	1	1	Clear accumulator	Lösche den Akkumulator

Instructions for Bit Manipulation TRANSFER / AND / OR / CLEAR / COMPLEMENT / SET

Mnemonic	Register	Incoding Binary	Inc. HEX	Affected Flags	Bytes	Cycles	Function	Funktionsbeschreibung
mov C,badr	-	10100010	A2	CY	2	1	Move contents of bit adress to carry	Lade Carry mit Inhalt von badr
mov badr,C	-	10010010	92	-	2	2	Move contents of carry to bit adress	Lade badr mit Inhalt des Carry
anl C,badr	-	10000010	82	CY	2	2	Logical AND carry with contents of bit adress	Das Carry wird überschrieben durch das Ergebnis der UND-Verknüpfung von Carry und Inhalt von badr
anl C,/badr	-	10110000	B0	CY	2	2	Logical AND carry with complement of contents of bit adress	Das Carry wird überschrieben durch das Ergebnis der UND-Verknüpfung von Carry und dem invertierten Inhalt von badr
orl C,badr	-	01110010	72	CY	2	2	Logical OR carry with contents of bit adress	Das Carry wird überschrieben durch das Ergebnis der ODER-Verknüpfung von Carry und Inhalt von badr
orl C,/badr	-	10100000	A0	CY	2	2	Logical OR carry with complement of contents of bit adress	Das Carry wird überschrieben durch das Ergebnis der ODER-Verknüpfung von Carry und dem invertierten Inhalt von badr
clr C	-	11000011	C3	CY	1	1	Clear carry	Carry löschen
clr badr	-	11000010	C2	-	2	1	Clear contents of bit adress	Lösche Inhalt von badr
cpl C	-	10110011	B3	CY	1	1	Complement carry	Carry invertieren
cpl badr	-	10110010	B2	-	2	1	Complement contents of bit adress	Invertiere Inhalt von badr
setb C	-	11010011	D3	CY	1	1	Set carry	Carry setzen
setb badr	-	11010010	D2	-	2	1	Set contents of bit adress	Setze Inhalt von badr

Arithmetic Instructions ADD

add A,Rr	r=0-7	00101rrr	28+r	CY,AC,OV,P	1	1	Add register contents to accumulator	Der Akkumulator wird überschrieben durch das Ergebnis der Addition von Akkumulator und dem Inhalt von Register Rr
add A,@Ri	i=0-1	0010011i	26+1	CY,AC,OV,P	1	1	Add contents of data memory addressed by Ri to accumulator	Der Akkumulator wird überschrieben durch das Ergebnis der Addition von Akkumulator und dem Inhalt der internen Datenspeicherszelle, die durch den Inhalt von Ri adressiert ist
add A,dadr	-	00100101	25	CY,AC,OV,P	2	1	Add contents of direct adress to accumulator	Der Akkumulator wird überschrieben durch das Ergebnis der Addition von Akkumulator und dem Inhalt von dadr
add A,#const8	-	00100100	24	CY,AC,OV,P	2	1	Add immediate data to accumulator	Der Akkumulator wird überschrieben durch das Ergebnis der Addition von Akkumulator und der 8 Bit Konstanten
addc A,Rr	r=0-7	00111rrr	38+r	CY,AC,OV,P	1	1	Add carry and register contents to accumulator	Der Akkumulator wird überschrieben durch das Ergebnis der Addition von Akkumulator und dem Inhalt von Register Rr und dem Inhalt des Carry-Flags
addc A,@Ri	i=0-1	0011011i	36+1	CY,AC,OV,P	1	1	Add carry and contents of data memory addressed by Ri to accumulator	Der Akkumulator wird überschrieben durch das Ergebnis der Addition von Akkumulator, dem Inhalt der internen Datenspeicherszelle, die durch den Inhalt von Ri adressiert ist, und dem Inhalt des Carry-Flags
addc A,dadr	-	00110101	35	CY,AC,OV,P	2	1	Add carry and contents of direct adress to accumulator	Der Akkumulator wird überschrieben durch das Ergebnis der Addition von Akkumulator dem Inhalt von dadr und dem Inhalt des Carry-Flags

4 Assembler für die 8051-Controller-Familie 65

Mnemonic	Register	Incoding Binary	Inc. HEX	Affected Flags	Bytes	Cycles	Function	Funktionsbeschreibung
inc A	-	00000100	04	P	1	1	Increment accumulator	Zum Inhalt des Akkumulators wird 1 addiert
inc Rr	r=0-7	00001rrr	08+r	-	1	1	Increment register	Zum Inhalt des Registers Rr wird 1 addiert
inc @Ri	i=0-1	0000011i	06+1	-	1	1	Increment data memory contents addressed by Ri	Zum Inhalt der internen Datenspeicherezelle, die durch Ri adressiert ist, wird 1 addiert
inc dadr	-	00000101	05	-	2	1	Increment contents of direct address	Zum Inhalt von dadr wird 1 addiert
inc DPTR	-	10100011	A3	-	1	2	Increment data pointer	Zum Inhalt des Registers DPTR wird 1 addiert
Arithmetic Instructions DECIMAL ADJUSTING								
da A	-	11010100	D4	CY,P	1	1	Decimal adjust accumulator for addition	Das Ergebnis einer vorangegangenen Addition zweier BCD-Zahlen wird unter Berücksichtigung von Carry und Hilfscarry wieder zu 2 BCD-Ziffern korrigiert
Arithmetic Instructions SUBTRACT								
subb A,Rr	r=0-7	10011rrr	98+r	CY,AC,OV,P	1	1	Subtract sum of register and carry from accumulator	Die Summe der Inhalte von Carry und Register Rr wird vom Akkumulator subtrahiert. Das Ergebnis überschreibt den Akkumulator
subb A,@Ri	i=0-1	1001011i	96+i	CY,AC,OV,P	1	1	Subtract sum of contents of data memory addressed by Ri to accumulator	Die Summe der Inhalte von Carry und der internen Datenspeicherezelle, die durch Ri adressiert ist, wird vom Akkumulator subtrahiert. Das Ergebnis überschreibt den Akkumulator
subb A,dadr	-	10010101	95	CY,AC,OV,P	2	1	Subtract sum of contents of direct adress and carry from accumulator	Die Summe der Inhalte von Carry und der Adresse dadr wird vom Akkumulator subtrahiert. Das Ergebnis überschreibt den Akkumulator
subb A,#const8	-	10010100	94	CY,AC,OV,P	2	1	Subtract sum of immediate data and carry form accumulator	Die Summe der Inhalte von Carry und der 8 BitKonstanten wird vom Akkumulator subtrahiert. Das Ergebnis überschreibt den Akkumulator
dec A	-	00010100	14	P	1	1	Decrement accumulator	Vom Inhalt des Akkumulators wird 1 subtrahiert
dec Rr	r=0-7	00011rrr	18+r	-	1	1	Decrement register	Vom Inhalt des Registers Rr wird 1 subtrahiert
dec @Ri	i=0-1	0001011i	16+1	-	1	1	Decrement data memory contents addressed by Ri	Vom Inhalt der internen Datenspeicherezelle, die durch Ri adressiert ist, wird 1 subtrahiert
dec dadr	-	00010101	15	-	2	1	Decrement contents of direct address	Vom Inhalt von dadr wird 1 subtrahiert
Arithmetic Instructions MULTIPLY / DIVIDE								
mul AB	-	10100100	A4	CY,OV,P	1	4	Multiply accumulator with B-register	Die Inhalte von Akkumulator und B-Register werden multipliziert. Das niederwertige Byte des Produktes überschreibt den Akkumulator, das höherwertige Byte das B-Register. Das Carry wird gelöscht, das OV-Bit wird gesetzt, wenn das Ergebnis im B-Register $\neq 0$ ist
div AB	-	100100100A4		CY,OV,P	1	4	Divide accumulator by B-register	Der Inhalt des Akkumulators wird durch den Inhalt des B-Registers dividiert. Der Quotient überschreibt den Akkumulator, der Rest das B-Register. Das Carry wird gelöscht. Bei Division durch 0 wird das OV-Bit gesetzt
Shift Instructions								
rl A	-	00100011	23	-	1	1	Rotate left without carry	Verschiebe den Inhalt des Akkumulators um 1 Stelle nach links. Die Stelle 2^7 wird zur Stelle 2^0 verschoben
rlc A	-	00110011	33	CY,P	1	1	Rotate left through carry	Verschiebe den Inhalt des Akkumulators um 1 Stelle nach links über das Carry. Das Carry wird an die Stelle 2^0 verschoben
rr A	-	00000011	03	-	1	1	Rotate right without carry	Verschiebe den Inhalt des Akkumulators um 1 Stelle nach rechts. Die Stelle 2^0 wird zur Stelle 2^7 verschoben
rrc A	-	00010011	13	CY,P	1	1	Rotate left through carry	Verschiebe den Inhalt des Akkumulators um 1 Stelle nach rechts. Das Carry wird an die Stelle 2^7 verschoben

Branch Instructions Unconditional Jumps							
Mnemonic	Register	Encoding Binary	Inc. HEX	Affected Flags	Cycles	Function	Funktionsbeschreibung
ljmp adr16	-	00000010	02	-	3 2	Jump to absolute address	Setze das Programm bei der 16 Bit Adresse adr 16 fort
sjmp rel	-	10000010	80	-	2 2	Jump to relative address	Setze das Programm relativ zum Programmzähler fort
ajmp adr11	-	See Table Branch ins	-	-	2 2	Jump to absolute address	Setze das Programm bei der 11 Bit Adresse adr 11 innerhalb der 2K, auf die der Programmzähler zeigt, fort
jmp adr	-	-	-	-	-	Generic Jump	ASM 51 generiert einen sjmp, ajmp, ljmp Befehl in Abhängigkeit der Entfernung des Sprungzieles. (Bei Vorwärts-Referenz wird grundsätzlich LJMP eingesetzt).
jmp @A+DPTR	-	01110011	73	-	1 2	Jump to sum of accumulator and data pointer	Setze das Programm an der Adresse fort, die sich aus der Summe von Akkumulator und DPTR ergibt
Branch Instructions Conditional Jumps							
jc rel	-	01000000	40	-	2 2	Jump if carry is set	Springe relativ zum Programmzähler, wenn Carry = 1 ist
jnc rel	-	01010000	50	-	2 2	Jump if carry is not set	Springe relativ zum Programmzähler, wenn Carry = 0 ist
jb badr,rel	-	00100000	20	-	3 2	Jump if bit is set	Springe relativ zum Programmzähler, wenn der Inhalt von badr = 1 ist
jnb badr,rel	-	00110000	30	-	3 2	Jump if bit is not set	Springe relativ zum Programmzähler, wenn der Inhalt von badr = 0 ist
jnb rel	-	01010000	50	-	3 2	Jump if bit is not set	Springe relativ zum Programmzähler, wenn der Inhalt von badr = 0 ist
jbz badr,rel	-	00010000	10	-	3 2	Jump and clear if bit is set	Springe relativ zum Programmzähler, wenn der Inhalt von badr = 1 ist und lösche den Inhalt von badr
jz rel	-	01100000	60	-	2 2	Jump relative if accumulator is set	Springe relativ zum Programmzähler, wenn der Akkumulator =0 ist
jnz rel	-	01110000	70	-	2 2	Jump relative if accumulator is not zero	Springe relativ zum Programmzähler, wenn der Akkumulator ≠0 ist
cjne A,dadr,rel	-	10110101	B5	CY	3 2	Compare memory to accumulator. Jump if not equal	Springe relativ zum Programmzähler, wenn die Inhalte von Akkumulator und dadr ungleich sind. Das Carry wird gesetzt, wenn dadr größer ist, sonst wird es rückgesetzt
cjne A,#const8,rel	-	10110100	B4	CY	3 2	Compare immediate data to accumulator. Jump if not equal	Springe relativ zum Programmzähler, wenn der Inhalte des Akkumulators ungleich konst8 ist. Das Carry wird gesetzt, wenn konst8 größer ist, sonst wird es rückgesetzt
cjne Rr,#const8,rel	0-7	10111rrr	B8+r	CY	3 2	Compare immediate data to register. Jump if not equal	Springe relativ zum Programmzähler, wenn der Inhalte von Rr ungleich konst8 ist. Das Carry wird gesetzt, wenn konst8 größer ist, sonst wird es rückgesetzt
cjne @Ri,#const8,rel	i=0-1	1011011i	B6+i	CY	3 2	Compare constants of indirect immediate data, Jump if not equal	Springe relativ zum Programmzähler, wenn der Inhalte der Datenspeicherzelle die durch Ri adressiert ist, ungleich konst8 ist. Das Carry wird gesetzt, wenn konst8 größer ist, sonst wird es rückgesetzt
djnz Rr,rel	r=0-7	11011rrr	D8+r	-	2 2	Decrement register and jump if not zero	Der Inhalt von Register Rr wird um 1 erniedrigt. Ist dann der Inhalt ≠ 0, springe relativ zum Programmzähler
djnz dadr,rel	-	11011101	D5	-	3 2	Decrement memory and jump if not zero	Der Inhalt von dadr wird um 1 erniedrigt. Ist dann der Inhalt ≠ 0, springe relativ zum Programmzähler
Subroutine Instructions							
lcall adr16	-	00010010	12	-	3 2	Long call to adr 16	Der Stack-Pointer wird erhöht. Der Inhalt des Programmzählers wird im Stack abgelegt. Das Programm wird bei der Adresse adr 16 fortgesetzt
acall adr11	-	See Table Branch instruction	-	-	2 2	Absolute call within current 2K	Der Stack-Pointer wird erhöht. Der Inhalt des Programmzählers wird im Stack abgelegt. Das Programm wird bei der Adresse adr 11 innerhalb der 2K fortgesetzt, auf die der Programmzähler zeigt
call adr11	-	-	-	-	2 2	Generic Call	ASM51 generiert einen acall, lcall Befehl in Abhängigkeit der Entfernung des Sprungzieles. (Bei Vorwärtsreferenz wird grundsätzlich lcall eingesetzt).
ret	-	00101101	22	-	1 2	Return from subroutine	Die oberen zwei Byte des Stack werden in den Programmzähler geladen. Der Stack-Pointer wird dabei erniedrigt
reti	-	00110010	32	-	1 2	Return from interrupt subroutine	Die oberen zwei Byte des Stack werden in den Programmzähler geladen. Der Stack-Pointer wird dabei erniedrigt. Die aktuelle Interrupt-Ebene wird freigegeben
nop	-	00000000	00	-	1 1	No operation	Leerbefehl