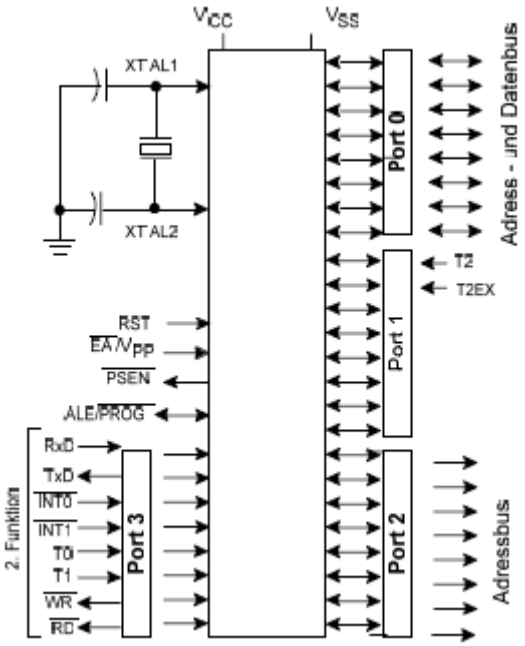
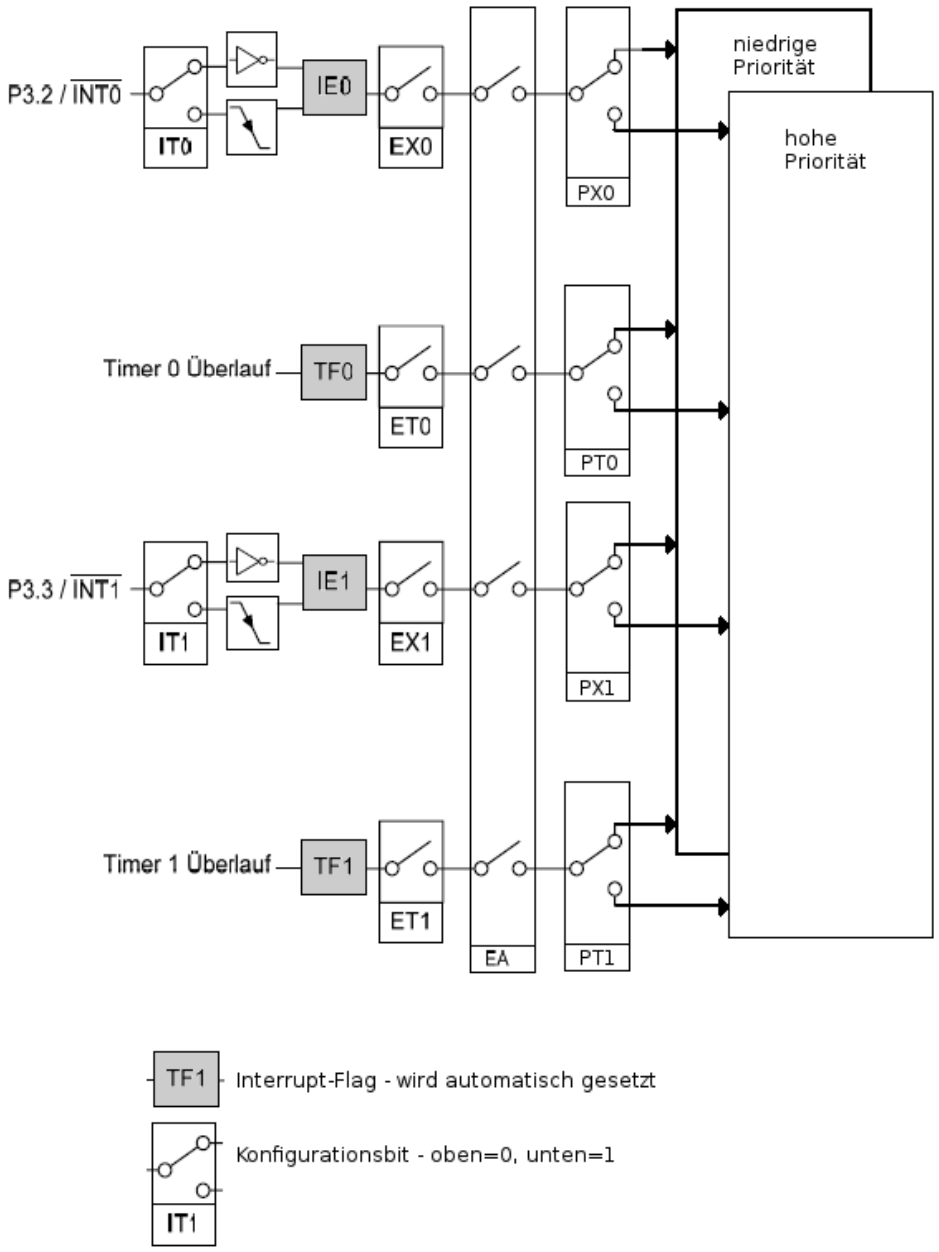
Inhalt



Interruptstruktur des 8051-Mikrocontroller



Interrupt	Bezeichnung	Einsprungadresse	Interrupt-Flag
0	Externer Interrupt 0	0x0003	IE0
1	Timer-Interrupt 0	0x000B	TF0
2	Externer Interrupt 1	0x0013	IE1
3	Timer-Interrupt 1	0x001B	TF1
4	Serieller Schnittstellen-Interrupt	0x0023	RI oder TI

Register zur Interrupt- und Timerkonfiguration

Bezeichnung: Interrupt-Enable				**Name: IEN0**		**Adresse: 0xA8**	
0xAF	0xAE	0xAD	0xAC	0xAB	0xAA	0xA9	0xA8
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EA	**-**	**-**	**-**	**ET1**	**EX1**	**ET0**	**EX0**

(Register sind über Name bzw. Adresse schreib- und lesbar. Bei bitadressierbaren Registern können auch einzelne Bits über die Namen in der untersten oder die Adressen in der zweiten Zeile gesetzt und gelöscht werden.)

Bezeichnung: Timermodus				**Name: TMOD**		**Adresse: 0x89**	
nicht bitadressierbar!							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Gate(1)	C/T(1)	M1(1)	M0(1)	Gate(0)	C/T(0)	M1(0)	M0(0)

Gate	C/T	M1	M0	
		0	0	Modus 0: 13 Bit mit Prescaler
		0	1	Modus 1: 16 Bit ohne Nachladen (THx + TLx)
		1	0	Modus 2: 8 Bit mit Auto-Reload (THx → TLx)
		1	1	Modus 3: 2 Stück 8-Bit-Timer (THx / TLx)
	0	Timer – Betrieb		
	1	Zähler – Betrieb		
0	Timer nur durch TR-Bit ein- und ausschalten			
1	Timer mit TR-Bit und Portpin ein- und ausschalten			

Bezeichnung: Timer- / Interruptkontrolle				**Name: TCON**		**Adresse: 0x88**	
0x8F	0x8E	0x8D	0x8C	0x8B	0x8A	0x89	0x88
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TF1	**TR1**	**TF0**	**TR0**	**IE1**	**IT1**	**IE0**	**IT0**

IEx	ITx	
	0	Lowpegel
	1	fallende Flanke
Interruptflag, i.d.R. von CPU gesetzt		

TFx	TRx	
	0	Timer x stopp
	1	Timer x läuft
Interruptflag, bei Timerüberlauf gesetzt		

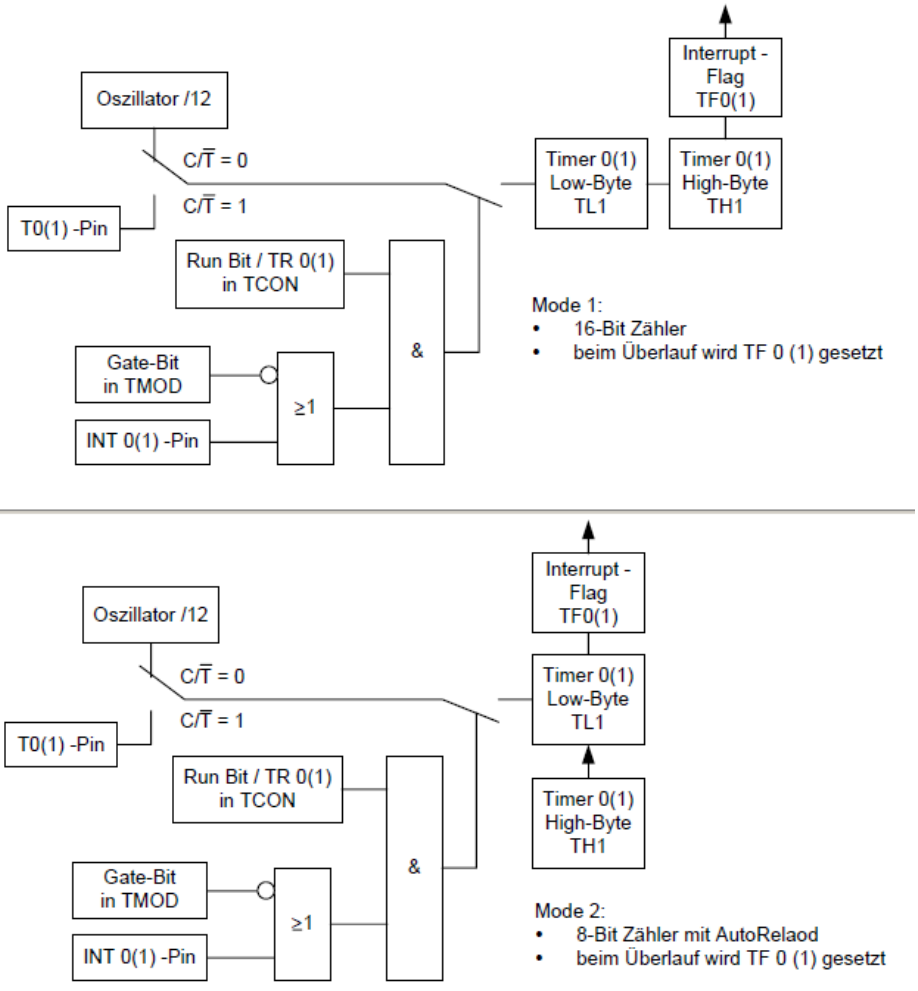
Bezeichnung: Interrupt-Prioritäten				**Name: IP**		**Adresse: 0xB8**	
0xBF	0xBE	0xBD	0xBC	0xBB	0xBA	0xB9	0xB8
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	-	-	**PS**	**PT1**	**PX1**	**PT0**	**PX0**

0: niedrige Priorität, 1: hohe Priorität

- Eine laufende ISR mit niedriger Priorität wird bei Auftreten eines Interrupts mit hoher Priorität unterbrochen.
- Eine laufende ISR mit gleicher oder höherer Priorität wird nicht unterbrochen.
- Treten zwei Interrupts gleichzeitig auf wird ein Interrupt mit hoher Priorität vor einem mit niedriger Priorität bearbeitet, bei gleicher Priorität gilt die Reihenfolge

  ExtInt0 → Timer 0 → ExtInt1 → Timer 1 → serieller Interrupt
- Wird das Interruptflag gelöscht bevor die ISR aufgerufen wurde (z.B. in einer ISR mit höherer Priorität), wird diese NICHT ausgeführt.

Prinzipschaltbild Timer 0 / Timer 1



Befehlssatz

Verwendete Abkürzungen / Operanden

Operand	Bedeutung (Adresse)
A	Akkumulator (E0)
dadr	8 Bit – Adresse im internen RAM oder im SFR-Bereich
adr11	11 Bit - Adresse
adr16	16 Bit - Adresse
AC	Auxiliary Carry (Hilfsübertrag-Flag) (D6)
B	Register B
badr	Bitadresse im internen RAM (20-2F) oder im SFR-Bereich(80-FF)
/badr	Invertierter Inhalt der Bitadresse (Komplement)
CY	Carry-Flag (D7)
#c8	8 Bit - Konstante
#c16	16 Bit - Konstante
D	Kennzeichnung für ein 4 Bit - Digit (Nibble)
DPTR	Datenpointerregister
HB	Highbyte eines Datenwortes
I	Interrupt
LB	Lowbyte eines Datenwortes
LSB	Bit 0 eines Bytes
MSB	Bit 7 eines Bytes
MZ	Maschinenzyklen
OV	Overflow (Überlauf)- Flag (D2)
P	Port
PC	Programmzähler
PSW	Programmstatuswort
rel	Signiertes 8 Bit - Offset für Sprungbefehle
@Ri	Adressregister für internes und externes RAM
Rn	Register 0 bis 7 der aktuellen Registerbank
SFR	Spezialfunktionsregister
SP	Stackpointer
T	Timer
TF	Timer - Flag
■	Flag wird beeinflusst
--	Flag wird nicht beeinflusst
WB	Wortbreite des Befehls in Bytes (Befehl + Argumente)

Datentransportbefehle

Befehl	Argumente	HEX-Code	WB	MZ	CY	OV	AC	Kurzbeschreibung
MOV	A,#c8	74	2	1	--	--	--	Akku direkt mit Konstante laden
MOV	Rn,#c8	78-7F	2	1	--	--	--	Direktes Laden des Registers mit einer Konstanten
MOV	dadr,#c8	75	3	2	--	--	--	Internen Speicher mit Konstante laden
MOV	A,Rn	E8-EF	1	1	--	--	--	Kopieren Registerinhalt in den Akku
MOV	Rn,A	F8-FF	1	1	--	--	--	Kopieren des Akkuinhaltes in ein Register
MOV	A,dadr	E5	2	1	--	--	--	Inhalt interner Speicherplatz in den Akku kopieren
MOV	dadr,A	F5	2	1	--	--	--	Inhalt Akku in internen Speicherplatz kopieren
MOV	Rn,dadr	A8-AF	2	2	--	--	--	internen Speicherplatz in ein Register kopieren
MOV	dadr,Rn	88-8F	2	2	--	--	--	Registerinhalt in internen Speicherplatz kopieren
MOV	dadr,dadr	85	3	2	--	--	--	Inhalt interner Speicherplatz in einen anderen kopieren
MOV	A,@R0	E6	1	1	--	--	--	Speicherinhalt internes RAM in Akku kopieren R0 bzw. R1 enthält die Adresse
MOV	A,@R1	E7	1	1	--	--	--	
MOV	@R0,A	F6	1	1	--	--	--	Akkuinhalt in Speicherplatz des internen RAM kopieren R0 bzw. R1 enthält die Adresse
MOV	@R1,A	F7	1	1	--	--	--	
MOV	dadr,@R0	86	2	2	--	--	--	Inhalt eines internen Speicherplatz in einen anderen kopieren R0 bzw. R1 enthält die Adresse
MOV	dadr,@R1	87	2	2	--	--	--	
MOV	@R0,dadr	A6	2	2	--	--	--	Inhalt eines internen Speicherplatzes in einen anderen kopieren R0 bzw. R1 enthält die Adresse
MOV	@R1,dadr	A7	2	2	--	--	--	
MOV	@R0,#c8	76	2	1	--	--	--	Konstante in internen RAM laden R0 bzw. R1 enthält die Zieladresse
MOV	@R1,#c8	77	2	1	--	--	--	
POP	dadr	D0	2	2	--	--	--	Speicherinhalt vom Stack holen
PUSH	dadr	C0	2	2	--	--	--	Speicherinhalt auf den Stack schreiben
MOV	badr,C	92	2	2	--	--	--	Carry-Inhalt in angegebene Bitadresse kopieren
MOV	C,badr	A2	2	1	■	--	--	Der Inhalt Bitadresse in das Carry kopieren
MOV	DPTR,#c16	90	3	2	--	--	--	16Bit-Konstante in Datenpointer laden
XCH	A,Rn	C8-CF	1	1	--	--	--	Akku- und Registerinhalt austauschen
XCH	A,dadr	C5	2	1	--	--	--	Internen Speicher mit dem Akkuinhalt tauschen
XCH	A,@R0	C6	1	1	--	--	--	Inhalt interner Speicherplatzes Akku austauschen (R0 bzw. R1 enthält die Zieladresse
XCH	A,@R1	C7	1	1	--	--	--	
XCHD	A,@R0	D6	1	1	--	--	--	Das LOW-Nibble eines Speicherplatzes im internen RAM gegen das LOW-Nibble des Akkus austauschen. Die HIGH-Nibble beider Speicher werden nicht verändert. (R0 bzw. R1 enthält Adresse)
XCHD	A,@R1	D7	1	1	--	--	--	
MOVX	A,@R0	E2	1	2	--	--	--	Inhalt externer Speicherplatzes in Akku kopieren, R0 bzw. R1 enthält Adresse
MOVX	A,@R1	E3	1	2	--	--	--	
MOVX	@R0,A	F2	1	2	--	--	--	Inhalt Akku in externen Speicherplatz kopieren, R0 bzw. R1 enthält Adresse.
MOVX	@R1,A	F3	1	2	--	--	--	
MOVX	A,@DPTR	E0	1	2	--	--	--	Inhalt externer Speicherplatz in Akku kopieren
MOVX	@DPTR,A	F0	1	2	--	--	--	Inhalt Akku in externen Speicherplatz kopieren
MOVC	A,@A+DPTR	93	1	2	--	--	--	Hole Konstante aus einer Tabelle im EEPROM.
MOVC	A,@A+PC	83	1	2	--	--	--	Hole Konstante aus einer Tabelle im EEPROM.

Arithmetische Operationen

Befehl	Argumente	HEX-Code	WB	MZ	CY	OV	AC	Kurzbeschreibung
CLR	A	E4	1	1	--	--	--	Löschen des Akku-Inhaltes
CPL	A	F4	1	1	--	--	--	Komplementieren des Akku-Inhaltes
INC	A	04	1	1	--	--	--	Inhalt des Akku um „1“ erhöhen
INC	Rn	08-0F	1	1	--	--	--	Inhalt des Registers um „1“ erhöhen
INC	dadr	05	2	1	--	--	--	Inhalt intern. Speicherstelle um „1“ erhöhen
INC	DPTR	A3	1	2	--	--	--	Inhalt des Datenpointers um „1“ erhöhen
INC	@R0	06	1	1	--	--	--	Inhalt einer Speicherstelle im internen
INC	@R1	07	1	1	--	--	--	RAM um „1“ erhöhen
DEC	A	14	1	1	--	--	--	Inhalt des Akku um „1“ vermindern
DEC	Rn	18-1F	1	1	--	--	--	Inhalt des Registers um „1“ vermindern
DEC	dadr	15	2	1	--	--	--	Inhalt interne Speicherst. um „1“ vermindern
DEC	@R0	16	1	1	--	--	--	Inhalt int. Speicherstelle um „1“ vermindern
DEC	@R1	17	1	1	--	--	--	Inhalt int. Speicherstelle um „1“ vermindern
ADD	A,#c8	24	2	1	■	■	■	Addition einer Konstante zum Akkuinhalt
ADDC	A,#c8	34	2	1	■	■	■	Addition einer Konstante plus Carry
ADD	A,Rn	28-2F	1	1	■	■	■	Addition eines Registerinhaltes zum Akkuinhalt
ADDC	A,Rn	38-3F	1	1	■	■	■	Add. eines Registerinh. plus Übertrag zum Akkuinhalt
ADD	A,dadr	25	2	1	■	■	■	Inhalt int. Speicherstelle zum Akkus addieren
ADDC	A,dadr	35	2	1	■	■	■	Inhalt int. Speicherstelle plus CY zum Akku addieren
ADD	A,@R0	26	1	1	■	■	■	Inhalt einer Speicherstelle im internen
ADD	A,@R1	27	1	1	■	■	■	RAM zum Inhalt des Akkus addieren
ADDC	A,@R0	36	1	1	■	■	■	Inhalt einer Speicherstelle im internen RAM
ADDC	A,@R1	37	1	1	■	■	■	plus CY zum Akku addieren
DA	A	D4	1	1	■	--	--	Dezimalkorrektur des Akku ***nur*** nach einer **BCD-Addition**
SUBB	A,#c8	94	2	1	■	■	■	Subtraktion Konstante plus Carry vom Akku
SUBB	A,dadr	95	2	1	■	■	■	Subtrakt. Int. Speicherinhalt plus Carry vom Akku
SUBB	A,Rn	98-9F	1	1	■	■	■	Subtrakt. eines Registers plus Carry vom Akku
SUBB	A,@R0	96	1	1	■	■	■	Subtraktion eines Speicherinhaltes des
SUBB	A,@R1	97	1	1	■	■	■	internen RAM plus Carry vom Akkuinhalt
SWAP	A	C4	1	1	--	--	--	Vertausche die Nibbles des Akkus
MUL	AB	A4	1	4	■	■	--	Multipliziere den Akku B- Register
DIV	AB	84	1	4	■	■	--	Teile Akkuinhalt durch den B-Registerinhalt
RL	A	23	1	1	--	--	--	Rotiere Akku-Inhalt eine Stelle nach links
RLC	A	33	1	1	■	--	--	Rotiere Akku-Inhalt durch Carry nach links
RR	A	03	1	1	--	--	--	Rotiere Akku-Inhalt eine Stelle nach rechts
RRC	A	13	1	1	■	--	--	Rotiere Akku-Inhalt durch Carry nach rechts
SETB	C	D3	1	1	■	--	--	Setze das CY-Bit auf „1“
CLR	C	C3	1	1	■	--	--	Setze das CY-Bit auf „0“
CPL	C	B3	1	1	■	--	--	Komplementiere das CY-Bit
SETB	badr	D2	2	1	--	--	--	Setze das adressierte Bit auf „1“
CLR	badr	C2	2	1	--	--	--	Setze das adressierte Bit auf „0“
CPL	badr	B2	2	1	--	--	--	Komplementiere das adressierte Bit

Logische Operationen

Befehl	Argumente	HEX-Code	WB	MZ	CY	OV	AC	Kurzbeschreibung
ANL	A,#c8	54	2	1	--	--	--	Bitweise UND-Verknüpfung Akku und Konstante
ANL	A,Rn	58-5F	1	1	--	--	--	Bitweise UND-Verknüpfung zwischen Akku und Register.
ANL	A,dadr	55	2	1	--	--	--	Bitweise UND-Verknüpfung zwischen Akku Speicherstelle im internen RAM
ANL	dadr,#c8	53	3	2	--	--	--	Bitweise UND-Verknüpfung Konstante und Speicherstelle im internen RAM
ANL	dadr,A	52	2	1	--	--	--	Bitweise UND-Verknüpfung zwischen Akku und Speicherstelle im internen RAM
ANL	C,badr	82	2	2	■	--	--	UND-Verknüpfung zwischen Carry und Bit
ANL	C,/badr	B0	2	2	■	--	--	UND-Verknüpfung zwischen Carry und invertiertem Bit.
ANL	A,@R0	56	1	1	--	--	--	Bitweise UND-Verknüpfung Akku und RAM-Inhalt R0 bzw. R1 enthält die Adresse
ANL	A,@R1	57	1	1	--	--	--	
ORL	A,#c8	44	2	1	--	--	--	Bitweise ODER-Verknüpfung Akku und Konstante
ORL	dadr,#c8	43	3	2	--	--	--	Bitweise ODER-Verknüpfung Konstante und Speicherstelle im internen RAM
ORL	A,Rn	48-4F	1	1	--	--	--	Bitweise ODER-Verknüpfung zwischen Akku und Register.
ORL	A,dadr	45	2	1	--	--	--	Bitweise ODER-Verknüpfung zwischen Akku und Speicherstelle im internen RAM
ORL	dadr,A	42	2	1	--	--	--	Bitweise ODER-Verknüpfung Akku und Speicherstelle im internen RAM
ORL	C,badr	72	2	2	■	--	--	ODER-Verknüpfung zwischen Carry Bit
ORL	C,/badr	A0	2	2	■	--	--	ODER-Verknüpfung zwischen Carry invertiertem Bit.
ORL	A,@R0	46	1	1	--	--	--	Bitweise ODER-Verknüpfung Akku und RAM-Inhalt R0 bzw. R1 enthält die Adresse
ORL	A,@R1	47	1	1	--	--	--	
XRL	A,#c8	64	2	1	--	--	--	Bitweise EXKLUSIV-ODER-Verknüpfung Akku und Konstante
XRL	dadr,#c8	63	3	2	--	--	--	Bitweise EXKLUSIV-ODER-Verknüpfung Konstante und Speicherstelle im internen RAM
XRL	A,Rn	68-6F	1	1	--	--	--	Bitweise EXKLUSIV-ODER-Verknüpfung Akku und Register.
XRL	A,dadr	65	2	1	--	--	--	Bitweise EXKLUSIV-ODER-Verknüpfung Akku und Speicherstelle im internen RAM
XRL	dadr,A	62	2	1	--	--	--	Bitweise EXKLUSIV-ODER-Verknüpfung Akku und Speicherstelle im internen RAM
XRL	A,@R0	66	1	1	--	--	--	Bitweise EXKLUSIV-ODER-Verknüpfung Akku und RAM-Inhalt R0 bzw. R1 enthält die Adresse
XRL	A,@R1	67	1	1	--	--	--	

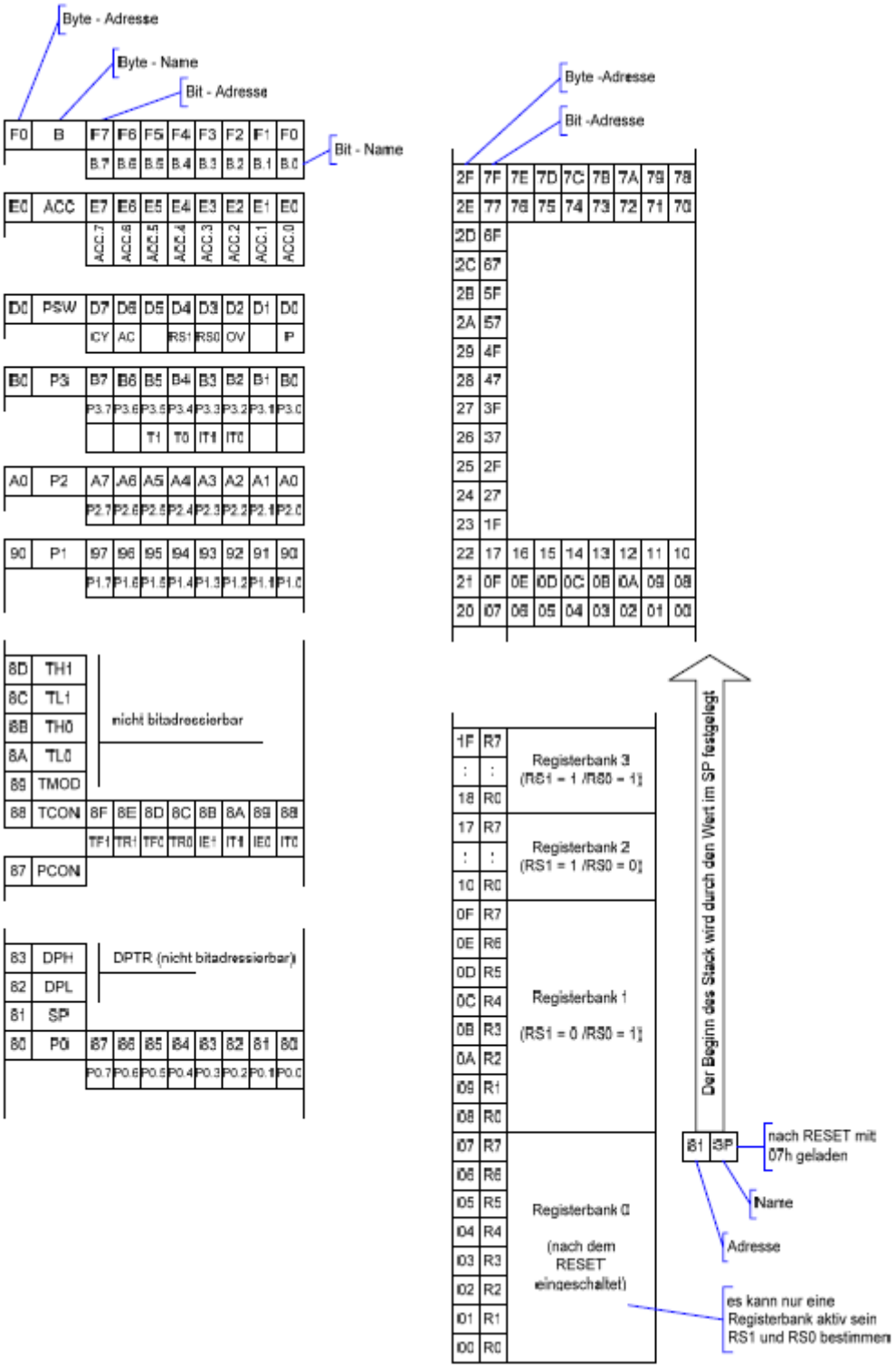
Hinweis: Das vordere Argument gibt an, wo das Ergebnis steht:

ANL A, dadr → Ergebnis steht im Akku
ANL dadr, A → Ergebnis steht im internen RAM

Sprungbefehle

Befehl	Argumente	HEX-Code	WB	MZ	CY	OV	AC	Kurzbeschreibung
LJMP	adr16	02	3	2	--	--	--	Programmsprung im 64K-Block
SJMP (JMP)	rel	80	2	2	--	--	--	relativer Programmsprung im Bereich -128 bis +127 zur nachfolgenden Befehlsadresse
AJMP	adr11	01-E1	2	2	--	--	--	Sprung im 2k-Block
JMP	@A+DPTR	73	1	2	--	--	--	Springe zur Adresse, die aus Akku- und Datenpointerinhalt gebildet wird.
JBC	badr,rel	10	3	2	--	--	--	Springe bei gesetztem Bit und lösche es
JB	badr,rel	20	3	2	--	--	--	Springe bei gesetztem Bit
JNB	badr,rel	30	3	2	--	--	--	Springe bei gelöschtem Bit (=0)
JC	rel	40	2	2	--	--	--	Springe bei gesetztem Carry-Bit
JNC	rel	50	2	2	--	--	--	Springe bei gelöschtem Carry-Bit (=0)
JZ	rel	60	2	2	--	--	--	Springe, wenn Akkuinhalt gleich Null
JNZ	rel	70	2	2	--	--	--	Springe, wenn Akkuinhalt ungleich Null
DJNZ	Rn,rel	D8-DF	2	2	--	--	--	Vermindere Register um Eins und springe, wenn Rest ungleich Null
DJNZ	dadr,rel	D5	3	2	--	--	--	Vermindere den Speicherinhalt im internen RAM um Eins und springe, wenn Rest ungleich Null
CJNE	A,#c8,rel	B4	3	2	■	--	--	Vergleiche Akku mit Konstante und springe bei Ungleichheit
CJNE	Rn,#c8,rel	B8-BF	3	2	■	--	--	Vergleiche Register mit Konstante und springe bei Ungleichheit
CJNE	A,dadr,rel	B5	3	2	■	--	--	Vergleiche Akku- und Speicherinhalt und springe bei Ungleichheit.
CJNE	@R0,#c8,rel	B6	3	2	■	--	--	Vergleiche Inhalt interne Speicherstelle mit Konstante und springe bei Ungleichheit. R0 bzw. R1 enthält die Adresse
CJNE	@R1,#c8,rel	B7	3	2	■	--	--	
LCALL	Adr16	12	3	2	--	--	--	Unterprogrammaufruf im 64k-Block
ACALL	adr11	11-F1	2	2	--	--	--	Unterprogrammaufruf im 2k-Block
RET		22	1	2	--	--	--	Ende Unterprogramm
RETI		32	1	2	--	--	--	Ende UP plus löschen des INT-Flags
NOP		00	1	1	--	--	--	Keine Aktivität

Aufteilung und Adressen internes RAM



Blockkschaltbild 8051