Aufgabe 1: ***Entwurf von Assemblerprogrammen***

a) Vorteile von Assembler gegenüber Maschinensprache:

- Befehle werden nicht numerisch sondern symbolisch notiert, d.h. statt 0-en und 1-en enthält jeder Operationscode einen festen symbolischen Namen („mnemonisch“), der an die Semantik der Operation erinnert (ADD, STORE, LOAD, …)
- Operandenadressen können Namen zugeordnet werden -> Adressierung erfolgt dann über diesen Namen
- Befehle (d.h. die Befehlsadressen) können durch Marken gekennzeichnet werden (etwa zur Festlegung von Sprungzielen)
- Zusätzliche Pseudobefehle definierbar, z.B. Reservierung von Speicherplatz für Variablen, Festlegung von Programmanfangsadresse, …

b)

```
Add48:
     mov A,R0
     add A,#48
     mov R0,A
ret
```

c)

```
Divider:
     mov A,R1
     rrc A
     mov R1,A
     mov A,R0
     rrc A
     mov R0,A
ret
```

Aufgabe 2: ***Mikrocontrollerprogrammierung mit dem 80C51***

a) Unterprogramm

```
START_SUBROUTINE:
     MOV A, R2
     MOV R1, A
     MOV A, R0
LOOP JZ FERTIG
     MUL A, R1
     MOV R1, A
     MOV A, R0
     SUBB A, R2
     MOV R0, A
     JMP LOOP
FERTIG:
     RET
```

b) Registerinhalte

Vorher: R0: 6_d, R1: 4_d, R2: 3_d

Nachher: R0: $0000\ 0000_b$,
R1: $0011\ 0110_b$,
R2: $0000\ 0011_b$.

c) Das Ergebnis steht in R1 und entspricht der Berechnung der Fakultät der Zahl, die sich vor dem Programmstart in R0 befand.

Aufgabe 3: ***Baudratengenerierung mit dem 80C51***

a)

	Bit 7							Bit 0
SCON	0	1	X	1	X	X	0	0
	Mode 1		(für Mode 2, 3)	receive enable	(für Mode 2, 3)		Interrupt-Flags	

	Bit 7							Bit 0
TMOD	0/1/X	0	1	0	X	X	X	X
	Start durch TR1	Timer	8 bit Autoreload			(für Timer 0)		

b)

Autoreloadwert: EEh → Anzahl Zählerschritte = 256 – 238 = 18
Frequenz zur Baudratengenrierung: $18 \cdot 4800 = 86400$ kHz
Eingangsfrequenz der seriellen Schnittstelle: Timer 1-Überläufe dividiert durch 16
Eingangsfrequenz für Timer 1: $f_{osc}/12$
damit folgt für die Frequenz f_{osc}: $f = 86,4kHz \cdot 12 \cdot 16 = 16,5888MHz$

c) Exakte Baudrate: $\frac{16\text{MHz}}{12 \cdot 16 \cdot 18} = 4629,629$ baud

Fehler: $\frac{4800 - 4629,629}{4629,629} = 0,0368$

d) Für die maximale Baudrate muss die Zahl der Zählerschritte minimal werden, d. h. Zählerüberlauf nach einem Schritt.

damit folgt für die zugehörige Baudrate: $\frac{12MHz}{12 \cdot 16} = 62.5$ kbaud

Aufgabe 4: ***Timer/Counter des Mikrocontrollers 80C52***

a) Der Aufruf der Interruptroutine soll alle 10 ms erfolgen.

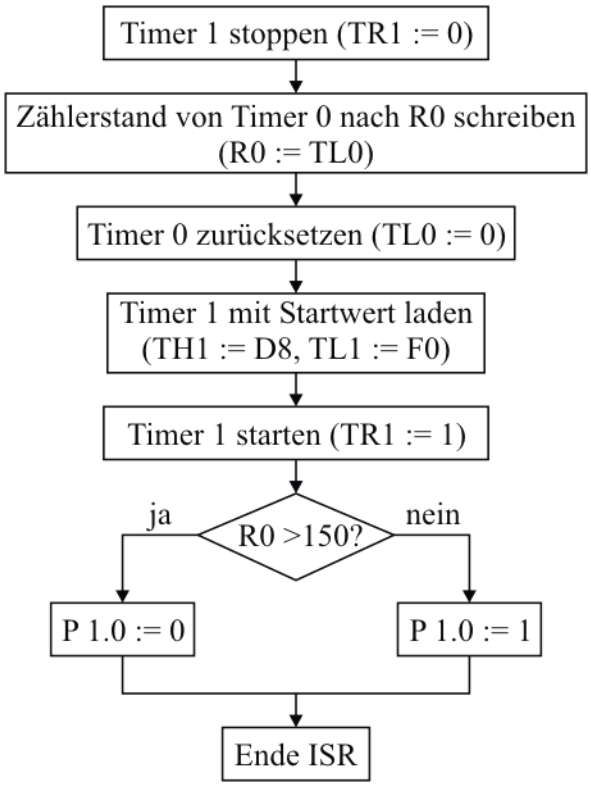
Dafür benötigte Zählschritte für Timer 1: $n = \frac{12\ \text{MHz}}{12} \cdot 10\ \text{ms} = 10.000$

Timer 1 muss als 16 bit-Timer betrieben werden (ohne Autoreload) ⇒ Mode 1
Reloadwert: 65.536 – 10.000 = 55.536 = D8F0 h

b) In 10 ms werden durchschnittlich maximal 200 Impulse gezählt
⇒ Timer 0 als 13 bit-Zähler (Mode 0) oder 16 bit-Zähler (Mode 1), es werden aber nur 8 bit zum Zählen benötigt.

Bit	7	6	5	4	3	2	1	0
IE	**1**	**X**	**X**	**X**	**1**	**X**	**0/X**	**X**
TMOD	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**0/1**
TCON	**0**	**1**	**0**	**1**	**X**	**X**	**X**	**X**

c) Flussdiagram der Interruptserviceroutine



Timer 0/1 des 80C51

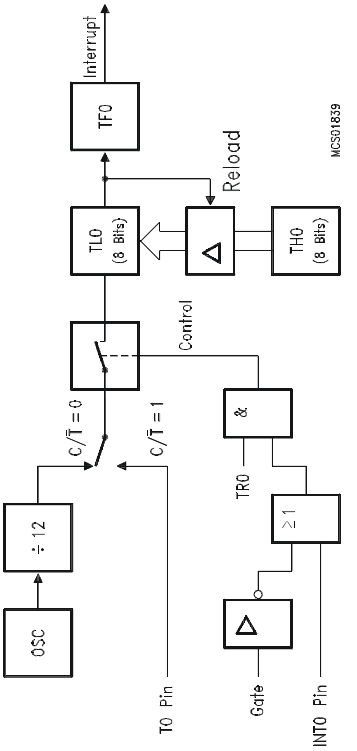


Figure 7-23
Timer/Counter 0/1, Mode 2: 8-Bit Timer/Counter with Auto-Reload

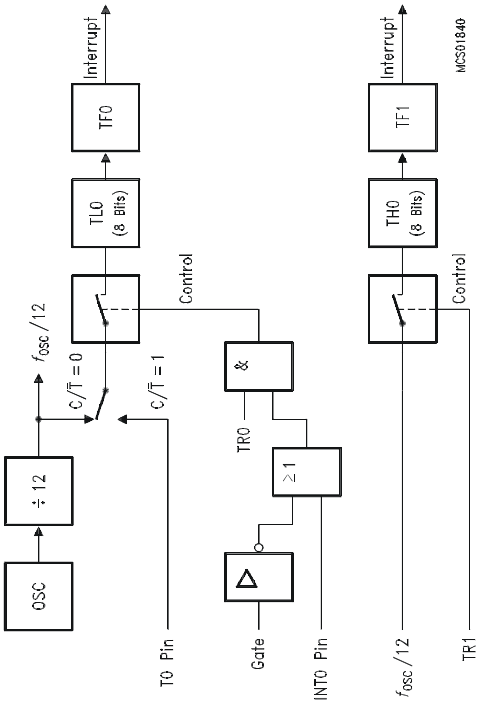


Figure 7-24
Timer/Counter 0, Mode 3: Two 8-Bit Timer/Counter

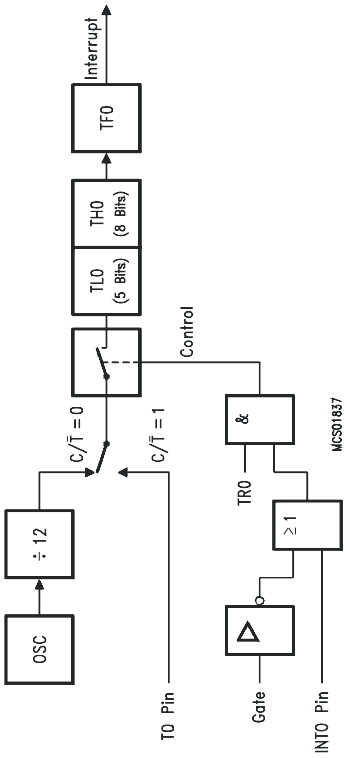


Figure 7-21
Timer/Counter 0/1, Mode 0: 13 Bit Timer/Counter

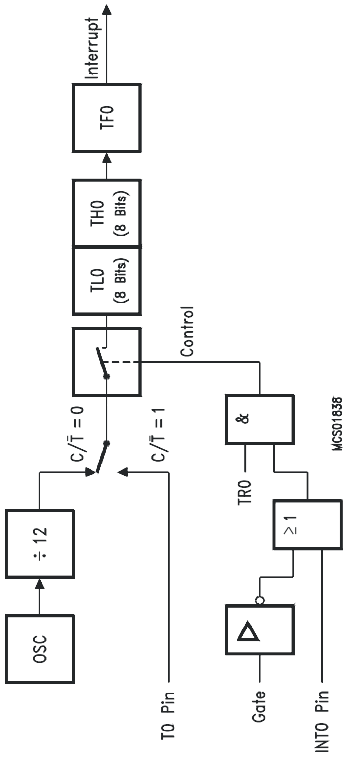
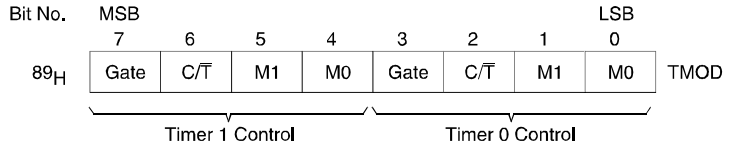


Figure 7-22
Timer/Counter 0/1, Mode 1: 16-Bit Timer/Counter

Institut für Industrielle Informationstechnik
Prof. Dr.-Ing. K. Dostert
Hertzstr. 16/Bau 35, Tel. 0721/608-4597

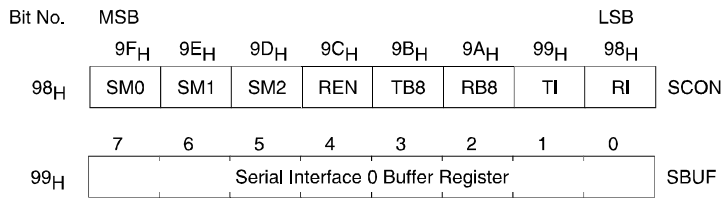
Special Function Register TMOD (Address 89_H) **Reset Value : 00_H**



Bit	Function
GATE	Gating control When set, timer/counter "x" is enabled only while "INT x" pin is high and "TRx" control bit is set. When cleared timer "x" is enabled whenever "TRx" control bit is set.
C/$\overline{T}$	Counter or timer select bit Set for counter operation (input from "Tx" input pin). Cleared for timer operation (input from internal system clock).
M1 M0	Mode select bits

M1	M0	Function
0	0	8-bit timer/counter: "THx" operates as 8-bit timer/counter "TLx" serves as 5-bit prescaler
0	1	16-bit timer/counter. "THx" and "TLx" are cascaded; there is no prescaler
1	0	8-bit auto-reload timer/counter. "THx" holds a value which is to be reloaded into "TLx" each time it overflows
1	1	Timer 0 : TL0 is an 8-bit timer/counter controlled by the standard timer 0 control bits. TH0 is an 8-bit timer only controlled by timer 1 control bits. Timer 1 : Timer/counter 1 stops

Special Function Register SCON (Address 98_H) **Reset Value : 00_H**
Special Function Register SBUF (Address 99_H) **Reset Value : XX_H**



Bit	Function
SM0 SM1	Serial port 0 operating mode selection bits

SM0	SM1	Selected operating mode
0	0	Serial mode 0 : Shift register, fixed baud rate (f_{osc}/12)
0	1	Serial mode 1 : 8-bit UART, variable baud rate
1	0	Serial mode 2 : 9-bit UART, fixed baud rate (f_{osc}/32 or f_{osc}/64)
1	1	Serial mode 3 : 9-bit UART, variable baud rate

SM2	Enable serial port multiprocessor communication in modes 2 and 3 In mode 2 or 3, if SM2 is set to 1 then RI0 will not be activated if the received 9th data bit (RB8) is 0. In mode 1, if SM2 = 1 then RI will not be activated if a valid stop bit was not received. In mode 0, SM2 should be 0.
REN	Serial port receiver enable Enables serial reception. Set by software to enable serial reception. Cleared by software to disable serial reception.
TB8	Serial port transmitter bit 9 TB8 is the 9th data bit that will be transmitted in modes 2 and 3. Set or cleared by software as desired.
RB8	Serial port receiver bit 9 In modes 2 and 3, RB8 is the 9th data bit that was received. In mode 1, if SM2 = 0, RB8 is the stop bit that was received. In mode 0, RB8 is not used.
TI	Serial port transmitter interrupt flag TI is set by hardware at the end of the 8th bit time in mode 0, or at the beginning of the stop bit in the other modes, in any serial transmission. TI must be cleared by software.
RI	Serial port receiver interrupt flag RI is set by hardware at the end of the 8th bit time in mode 0, or halfway through the stop bit time in the other modes, in any serial reception (exception see SM2). RI must be cleared by software.

Serielle Schnittstelle des 80C51

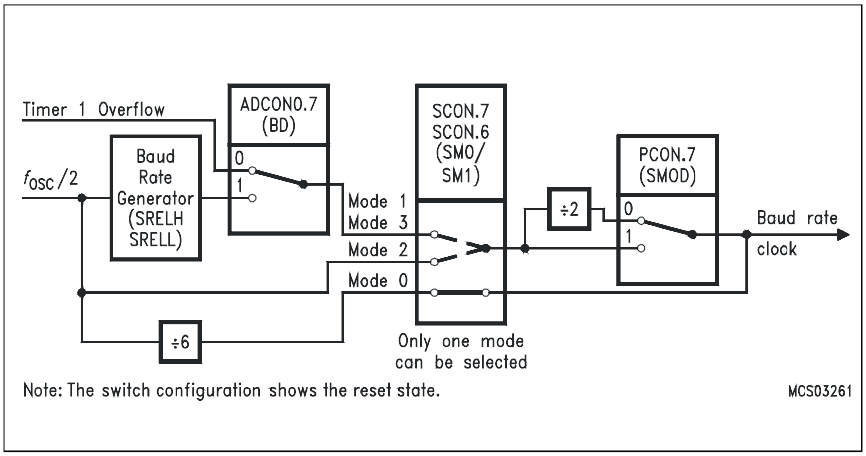


Figure 6-23
Baud Rate Generation for the Serial Port

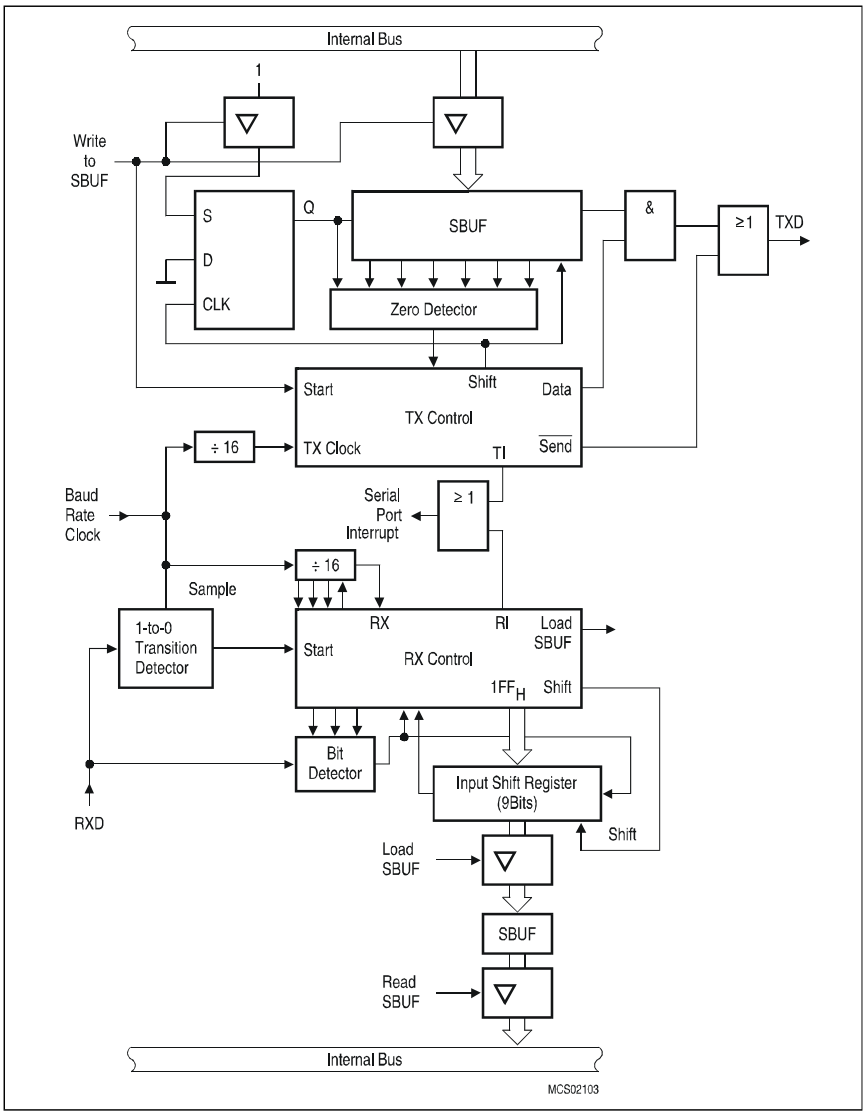


Figure 6-27
Serial Interface, Mode 1, Functional Diagram