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

Mikrorechnertechnik SS 2007

Mikrocontrollerprogrammierung Übung 4 Lösungen

Seite: 1/6

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
```

<u>Aufgabe 2: Mikrocontrollerprogrammierung mit dem 80C51</u>

```
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
```



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

Mikrorechnertechnik SS 2007

Mikrocontrollerprogrammierung Übung 4 Lösungen

Seite: 2/6

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
	Mod	de 1	(für Mode 2, 3)	receive enable	(für Mo	ode 2, 3)	Interrup	ot-Flags

	B ₁ t 7							Bit 0
TMOD	0/1/X	0	1	0	X	X	X	X
	Start	Timer	81	bit		(für Ti	mer 0)	
	durch		Autor	reload				
	TR1							

b)
Autoreloadwert: EEh → Anzahl Zählerschritte = 256 – 238 = 18
Frequenz zur Baudratengenrierung: 18 · 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 · 12 · 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

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

Mikrorechnertechnik SS 2007

Mikrocontrollerprogrammierung Übung 4 Lösungen

Seite: 3/6

Aufgabe 4: Timer/Counter des Mikrocontrollers 80C52

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

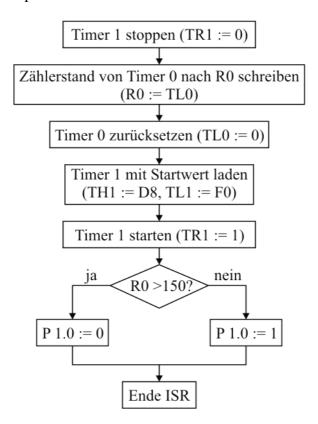
Dafür benötigte Zählschritte für Timer 1:
$$n = \frac{12 \text{ MHz}}{12} \cdot 10 \text{ ms} = 10.000$$

Timer 1 muss als 16 bit-Timer betrieben werden (ohne Autoreload) \Rightarrow 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
ΙΕ	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



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

Mikrorechnertechnik SS 2007

Mikrocontrollerprogrammierung Übung 4 Lösungen

Seite: 4/6

Timer 0/1 des 80C51

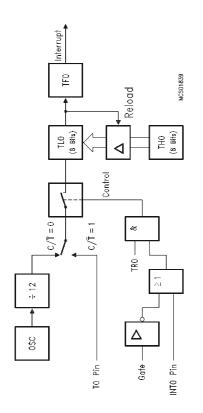


Figure 7-23 Timer/Counter 0/1, Mode 2: 8-Bit Timer/Counter with Auto-Reload

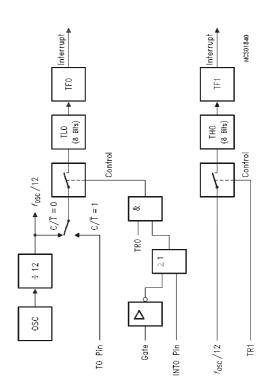
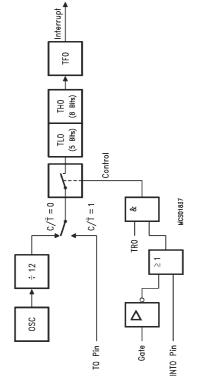
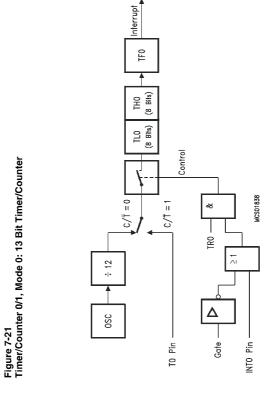


Figure 7-24 Timer/Counter 0, Mode 3: Two 8-Bit Timer/Counter

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







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

Mikrorechnertechnik SS 2007

Mikrocontroller programmierung Übung 4 Lösungen

Seite: 5/6

Spezialfunktionsregister des 80C51

Special	Functi	Special Function Register TMOD (Address 89 _H)	er TMOD	(Addi	H68 ssə.	~			Rese	Reset Value : 00 _H
ш	Bit No.	MSB 7	9	2	4	ო	Ø	-	LSB 0	
	H ₆₈	Gate	C∕⊒	2	MO	Gate	Ľ/O	M	MO	TMOD
			Timer 1 Control	control			Timer 0 Control	Control		
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.	ntrol timer/cou is set. red timer	inter ">	t" is enab	oled only v	while "IN	T x" pin i	is high a is set.	ind "TRx"
C/ <u></u>		Counter or timer select bit Set for counter operation (input from "Tx" input pin). Cleared for timer operation (input from internal system clock).	timer sel Inter oper r timer op	ect bit ation (eratior	input fror (input fr	n "Tx" ing om intern	out pin). ial syster	m clock).		
E S		Mode select bits	ct bits							
2		M1	MO	교	Function					
	,	0	0	 	8-bit timer/counter: "THx" operates as "TLx" serves as 5-k	8-bit timer/counter: "THx" operates as 8-bit timer/counter "TLx" serves as 5-bit prescaler	bit timer/ prescale	'counter er		
	•	0	-	16. T.	16-bit timer/counter. "THx" and "TLx" are	16-bit timer/counter. "THx" and "TLx" are cascaded; there is no prescaler	cascade	d; there i	s no pre	scaler
		-	0	# ± #	8-bit auto-reload "THx" holds a val time it overflows	8-bit auto-reload timer/counter. "THx" holds a value which is to t time it overflows	er/counte /hich is to	ır. o be reloa	aded into	8-bit auto-reload timer/counter. "THx" holds a value which is to be reloaded into "TLx" each time it overflows
		-	-	투투	Timer 0 : TL0 is an 8- timer 0 cont	Timer 0: TLO is an 8-bit timer/counter controlled by the standard timer 0 control bits. TH0 is an 8-bit timer only controlled	counter o	controlled 8-bit time	d by the	Timer 0 : TLO is an 8-bit timer/counter controlled by the standard timer 0 control bits. TH0 is an 8-bit timer only controlled by
				를 Ë i	timer 1 control bits. Timer 1:	irol bits.				
				Ë	ner/count	Timer/counter 1 stops	S			

Special Function Register SCON (Address 98 _H) Special Function Register SBUF (Address 99 _H)	tion Reg	jister SC Jister SB	ON (Ade	dress 98 Iress 99 ₁	Ĵ.			Res	Reset Value : 00 _H Reset Value : XX _H	Special
Bit No.	MSB							LSB		В
	9F _H		9Ен 9Dн 9Сн 9Вн 9Ан	9CH	9B _H	9A _H	H ₆₆	98 _H		
H ₈₆	SM0	SM1	SM2	REN	TB8	88H	F	굔	SCON	
	7	9	5	4		2	-	0		
H ₆₆			Serial In	Serial Interface 0 Buffer Register	Buffer F	Register	_		SBUF	

Bit	Function	5	
SMO	Serial p	ort 0 ope	Serial port 0 operating mode selection bits
LIMIO	SMO	SM1	Selected operating mode
	0	0	Serial mode 0 : Shift register, fixed baud rate (fosc/12)
	0	-	Serial mode 1 : 8-bit UART, variable baud rate
	-	0	Serial mode 2 : 9-bit UART, fixed baud rate ($f_{\rm osc}/32~{\rm or}f_{\rm osc}/64$)
	-	-	Serial mode 3: 9-bit UART, variable baud rate
SM2	Enable In mode	serial por	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 stop bit	(RB8) is was not	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 p Enables software	ort receiv s serial re e to disak	Serial port receiver enable Enables serial reception. Set by software to enable serial reception. Cleared by software to disable serial reception.
ТВ8	Serial p TB8 is t software	Serial port transmitte TB8 is the 9th data b software as desired.	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 p In mode RB8 is t	Serial port receiver bit 9 In modes 2 and 3, RB8 is RB8 is the stop bit that v	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.
F	Serial port tr TI is set by h of the stop b by software.	ort transr t by hardv top bit in vare.	Serial port transmitter interrupt flag It 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. It must be cleared by software.
<u> </u>	Serial p RI is ser the stop RI must	ort receiv t by hard b bit time t be clear	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.

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

Mikrorechnertechnik SS 2007

Mikrocontroller programmierung Übung 4 Lösungen

Seite: 6/6

Serielle Schnittstelle des 80C51

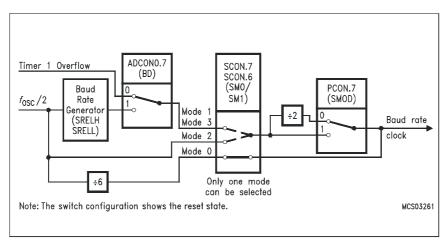


Figure 6-23
Baud Rate Generation for the Serial Port

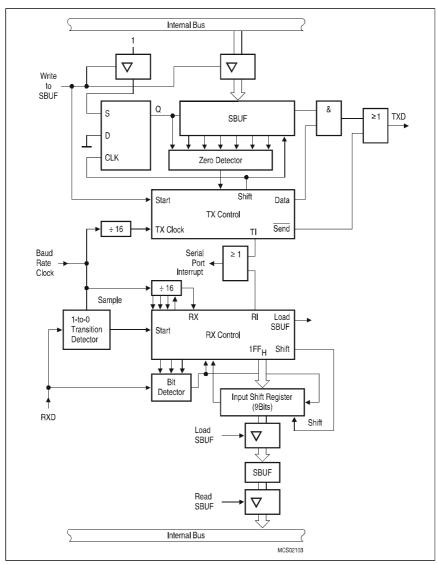


Figure 6-27 Serial Interface, Mode 1, Functional Diagram