Sveučilište J.J. Strossmayera u Osijeku
Odjel za matematiku

GRAĐA RAČUNALA

(predavanja u ak. god. 2005./2006.)

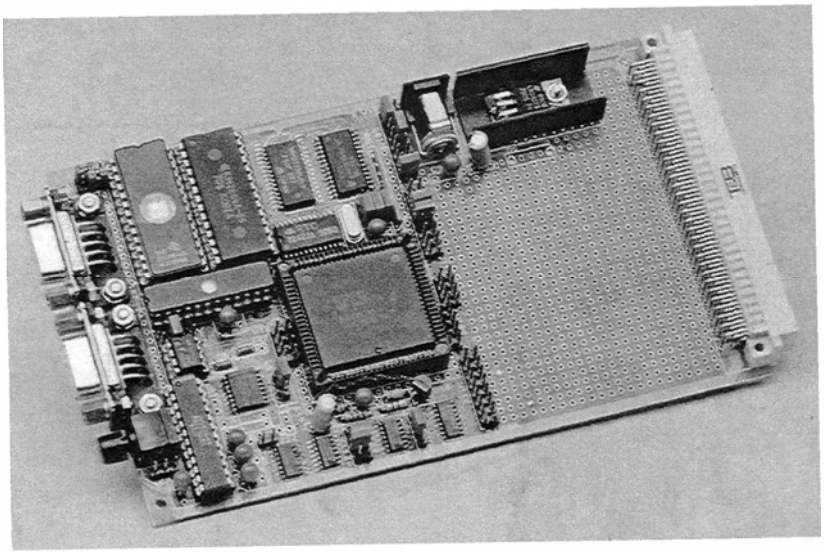
doc.dr.sc. Goran Martinović
www.etfos.hr/~martin
goran.martinovic@etfos.hr
Tel: 031 224-766
Soba: 0-15

Osijek, 2006.

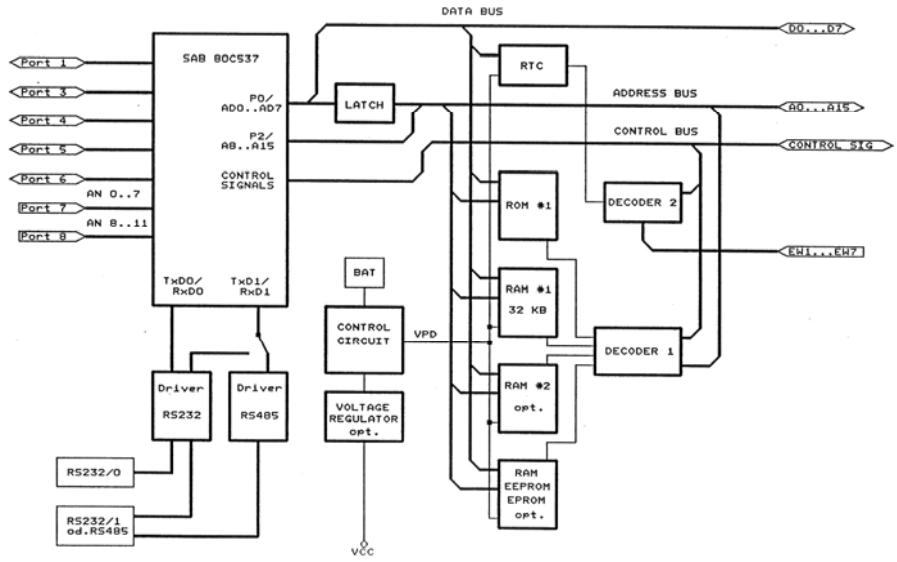
1

MIKROUPRAVLJAČ SAB80C537

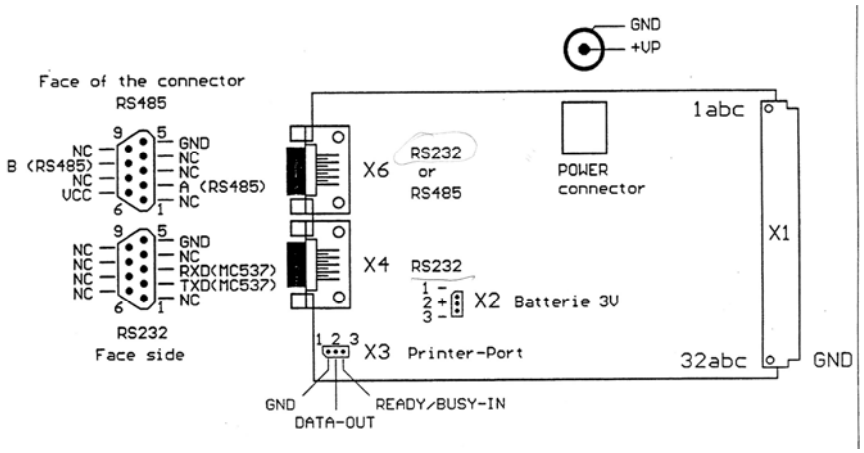
2



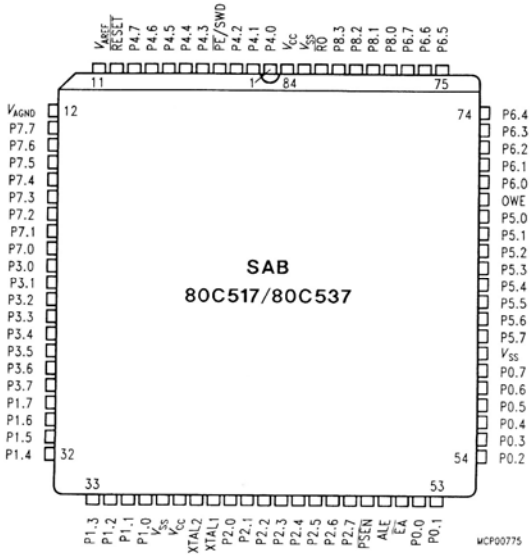
3


DATA BUS
D0...D7
SAB 80C537
Port 1
Port 3
Port 4
Port 5
Port 6
AN 0..7
Port 7
AN 8..11
Port 8
P0/
AD0..AD7
P2/
A8..A15
CONTROL
SIGNALS
TxD0/
RxD0
TxD1/
RxD1
LATCH
RTC
ADDRESS BUS
A0...A15
CONTROL BUS
CONTROL SIG
ROM *1
DECODER 2
EW1...EW7
BAT
CONTROL
CIRCUIT
VPD
RAM *1
32 KB
DECODER 1
RAM *2
opt.
VOLTAGE
REGULATOR
opt.
RAM
EEPROM
EPROM
opt.
Driver
RS232
Driver
RS485
RS232/0
RS232/1
od.RS485
VCC

4

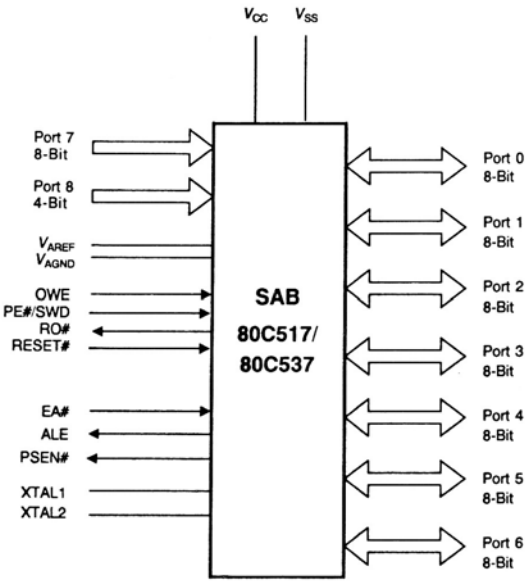

GND
+UP
Face of the connector
RS485
9
5
1
6
NC
B (RS485)
NC
VCC
GND
NC
NC
A (RS485)
NC
X6
RS232
or
RS485
POWER
connector
1abc
X1
NC
NC
NC
NC
GND
NC
RXD(MC537)
TXD(MC537)
NC
X4
RS232
1
2
3
X2
Batterie 3V
RS232
Face side
1 2 3
X3
Printer-Port
32abc
GND
GND
DATA-OUT
READY/BUSY-IN

5


SAB
80C517/80C537
MCP00775

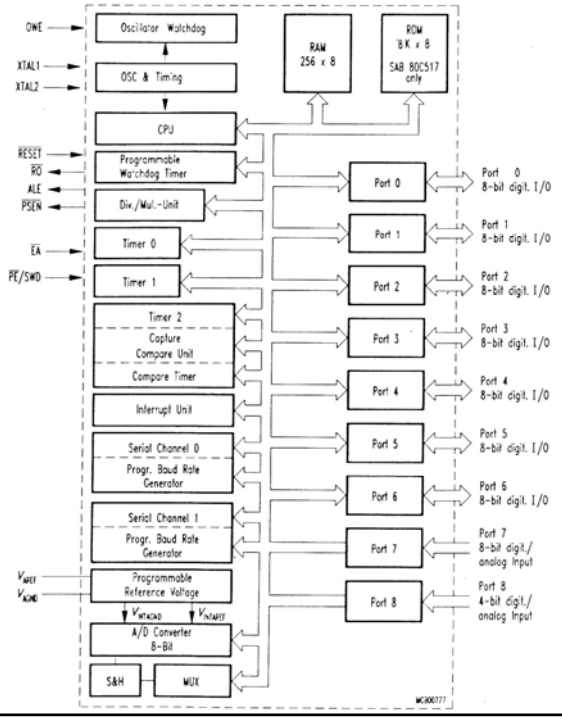
6

Logički simbol



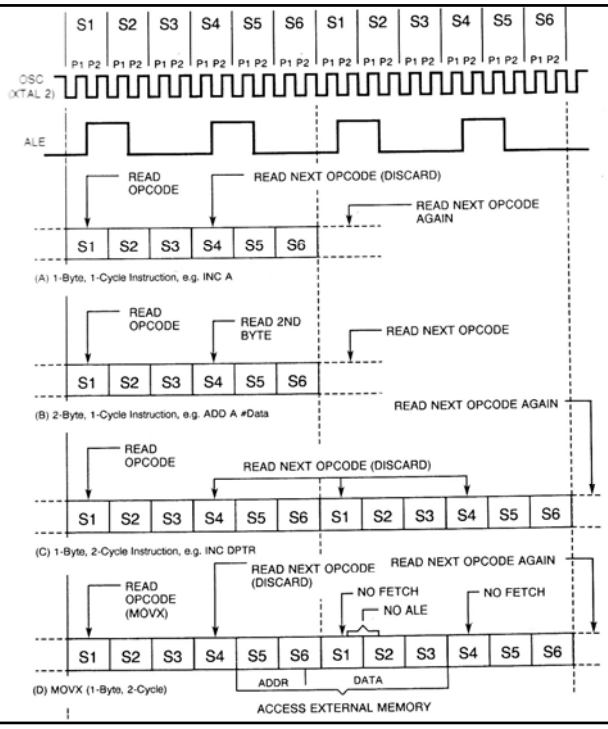
7

Funkcijski blok dijagram



8

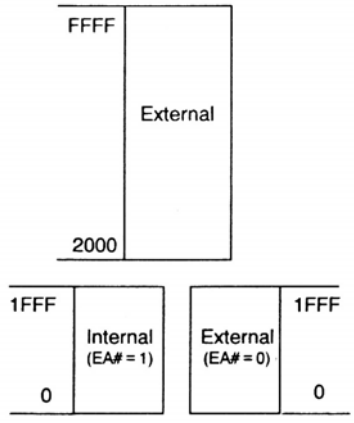
Fetch/execute sekvenca



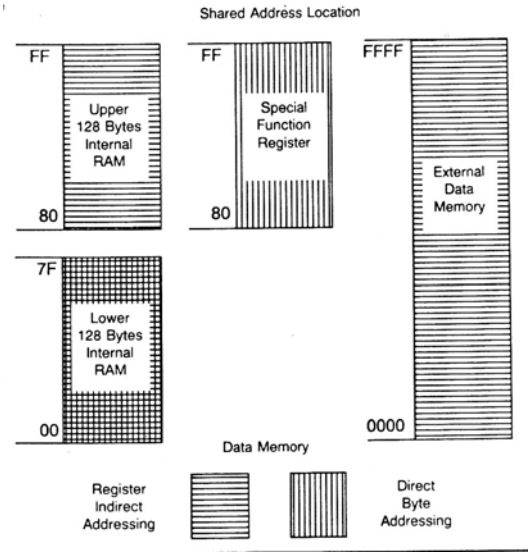
Adresni prostor

Address space	Locations	Addressing mode
Lower 128 bytes of RAM	00H to 7FH	direct/indirect
Upper 128 bytes of RAM	80H to 0FFH	indirect
Special function registers	80H to 0FFH	direct

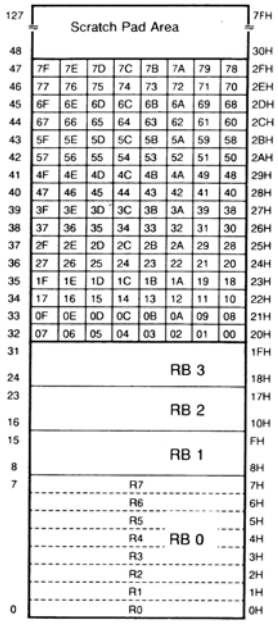
Adresni prostor programske memorije



Podatkovna memorija/
Adresni prostor SFR's



Donji dio unutarnje
podatkovne memorije



SFR's

Block	Symbol	Name	Address	Contents after Reset
CPU	**ACC**	**Accumulator**	**0E0H** 1)	**00H**
	B	**B -Register**	**0F0H** 1)	**00H**
	DPH	Data Pointer, High Byte	83H	
	DPL	Data Pointer, Low Byte	82H	
	DPSEL	Data Pointer Select Register	92H	
	PSW	**Program Status Word Register**	**0D0H** 1)	**00H**
	SP	Stack Pointer	81H	07H
A/D-Converter	**ADCON0**	**A/D Converter Control Register 0**	**0D8H** 1)	**00H**
	ADCON1	A/D Converter Control Register 1	0DCH	XXXX.0000B3)
	ADDAT	A/D Converter Data Register	0D9H	00H
	DAPR	D/A Converter Program Register	0DAH	00H
Interrupt System	**IEN0**	**Interrupt Enable Register 0**	**0A8H** 1)	**00H**
	CTCON 2)	Com. Timer Control Register	0E1H	0XXX.0000B 3)
	IEN1	**Interrupt Enable Register 1**	**0B8H** 1)	00H
	IEN2	Interrupt Enable Register 2	9AH	XXXX.0XX0B 3)
	IP0	Interrupt Priority Register 0	0A9H	00H
	IP1	Interrupt Priority Register 1	0B9H	XX00.0000B 3)
	IRCON	**Interrupt Request Control Register**	**0C0H** 1)	**00H**
	TCON 2)	**Timer Control Register**	**88H** 1)	**00H**
	T2CON 2)	**Timer 2 Control Register**	**0C8H** 1)	**00H**
MUL/DIV Unit	ARCON	Arithmetic Control Register	0EFH	0XXX.XXXXB 3)
	MD0	Multiplication/Division Register 0	0E9H	XXH 3)
	MD1	Multiplication/Division Register 1	0EAH	XXH 3)
	MD2	Multiplication/Division Register 2	0EBH	XXH 3)
	MD3	Multiplication/Division Register 3	0ECH	XXH 3)
	MD4	Multiplication/Division Register 4	0EDH	XXH 3)
	MD5	Multiplication/Division Register 5	0EEH	XXH 3)

12

SFR's

Block	Symbol	Name	Address	Contents after Reset
Compare/ Capture Unit (CCU)	CCEN	Compare/Capture Enable Register	0C1H	00H
	CC4EN	Compare/Capture 4 Enable Register	0C9H	X000.0000B 3)
	CCH1	Compare/Capture Register 1, High Byte	0C3H	00H
	CCH2	Compare/Capture Register 2, High Byte	0C5H	00H
	CCH3	Compare/Capture Register 3, High Byte	0C7H	00H
	CCH4	Compare/Capture Register 4, High Byte	0CFH	00H
	CCL1	Compare/Capture Register 1, Low Byte	0C2H	00H
	CCL2	Compare/Capture Register 2, Low Byte	0C4H	00H
	CCL3	Compare/Capture Register 3, Low Byte	0C6H	00H
	CCL4	Compare/Capture Register 4, Low Byte	0CEH	00H
	CMEN	Compare Enable Register	0F6H	00H
	CMH0	Compare Register 0, High Byte	0D3H	00H
	CMH1	Compare Register 1, High Byte	0D5H	00H
	CMH2	Compare Register 2, High Byte	0D7H	00H
	CMH3	Compare Register 3, High Byte	0E3H	00H
	CMH4	Compare Register 4, High Byte	0E5H	00H
	CMH5	Compare Register 5, High Byte	0E7H	00H
	CMH6	Compare Register 6, High Byte	0F3H	00H
	CMH7	Compare Register 7, High Byte	0F5H	00H
	CML0	Compare Register 0, Low Byte	0D2H	00H
	CML1	Compare Register 1, Low Byte	0D4H	00H
	CML2	Compare Register 2, Low Byte	0D6H	00H
	CML3	Compare Register 3, Low Byte	0E2H	00H
	CML4	Compare Register 4, Low Byte	0E4H	00H
	CML5	Compare Register 5, Low Byte	0E6H	00H
	CML6	Compare Register 6, Low Byte	0F2H	00H
	CML7	Compare Register 7, Low Byte	0F4H	00H
	CMSEL	Compare Input Select	0F7H	00H
	CRCH	Com./Rel./Capt. Register, High Byte	0CBH	00H
	CRCL	Com./Rel./Capt. Register, Low Byte	0CAH	00H
	CTCON	Com. Timer Control Register	0E1H	00H
	CTRELH	Com. Timer Rel. Register, High Byte	0DFH	00H
	CTRELL	Com. Timer Rel. Register, Low Byte	0DEH	00H
	TH2	Timer 2, High Byte	0CDH	00H
	TL2	Timer 2, Low Byte	0CCH	00H
	T2CON	**Timer 2 Control Register**	**0C8H** 1)	**00H**

13

SFR's

Block	Symbol	Name	Address	Contents after Reset
Ports	**P0**	**Port 0**	**80H** 1)	**FFH**
	P1	**Port 1**	**90H** 1)	**FFH**
	P2	**Port 2**	**0A0H** 1)	**FFH**
	P3	**Port 3**	**0B0H** 1)	**FFH**
	P4	**Port 4**	**0E8H** 1)	**FFH**
	P5	**Port 5**	**0F8H** 1)	**FFH**
	P6	Port 6	0FAH	FFH
	P7	Port 7, Analog/Digital Input	0DBH	XXH 3)
	P8	Port 8, Analog/Digital Input, 4-bit	0DDH	XXH 3)
Pow. Sav. M	PCON	Power Control Register	87H	00H
Serial Channels	**ADCON0** 2)	**A/D Converter Control Register**	**0D8H** 1)	**00H**
	PCON 2)	Power Control Register	87H	00H
	S0BUF	Serial Channel 0 Buffer Register	99H	XXH 3)
	S0CON	**Serial Channel 0 Control Register**	**98H** 1)	**00H**
	S1BUF	Serial Channel 1 Buffer Register	9CH	XXH 3)
	S1CON	Serial Channel 1 Control Register	9BH	0X00.0000B 3)
	S1REL	Serial Channel 1 Reload Register	9DH	00H
Timer 0/ Timer 1	**TCON**	**Timer Control Register**	**88H** 1)	**00H**
	TH0	Timer 0, High Byte	8CH	00H
	TH1	Timer 1, High Byte	8DH	00H
	TL0	Timer 0, Low Byte	8AH	00H
	TL1	Timer 1, Low Byte	8BH	00H
	TMOD	Timer Mode Register	89H	00H
Watchdog	**IEN0** 2)	**Interrupt Enable Register 0**	**0A8H** 1)	**00H**
	IEN1 2)	**Interrupt Enable Register 1**	**0B8H** 1)	**00H**
	IP0 2)	Interrupt Priority Register 0	0A9H	00H
	WDTREL	Watchdog Timer Reload Register	86H	00H

14

Akumulator, B, SP, DPTR, P0-P8

Program Status Word Register (PSW), SFR Address 0D0H

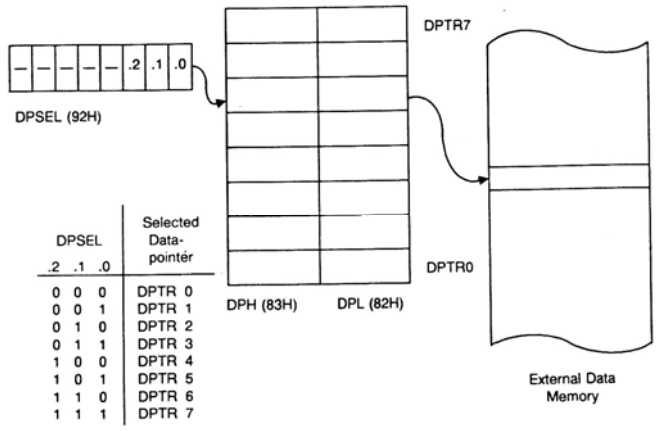
	0D7H	0D6H	0D5H	0D4H	0D3H	0D2H	0D1H	0D0H	
0D0H	CY	AC	F0	RS1	RS0	OV	F1	P	PSW

The PSW register contains program status information.

Bit	Function
CY	Carry flag
AC	Auxiliary carry flag (for BCD operations)
F0	General purpose user flag 0
RS1 RS0 0 0 0 1 1 0 1 1	Register bank select control bits Bank 0 selected, data address 00H-07H Bank 1 selected, data address 08H-0FH Bank 2 selected, data address 10H-17H Bank 3 selected, data address 18H-1FH
OV	Overflow flag
F1	General purpose user flag
P	Parity flag. Set/cleared by hardware each instruction cycle to indicate an odd/even number of "one" bits in the accumulator, i.e. even parity.

15

Vanjsko sučelje prema sabirnici i pokazivači podataka



DPSEL .2 .1 .0	Selected Data-pointer
0 0 0	DPTR 0
0 0 1	DPTR 1
0 1 0	DPTR 2
0 1 1	DPTR 3
1 0 0	DPTR 4
1 0 1	DPTR 5
1 1 0	DPTR 6
1 1 1	DPTR 7

16

Uporaba jednog pokazivača

Action	Code
Initialize shadow_variables with source_pointer	MOV LOW(SRC_PTR),#0FFH MOV HIGH(SRC_PTR),#1FH
Initialize shadow_variables with destination_pointer	MOV LOW(DES_PTR),#0A0H MOV HIGH(DES_PTR),#2FH

17

Primjer rutine

Action	Code
Initialize shadow_variables with source_pointer	MOV LOW(SRC_PTR),#0FFH MOV HIGH(SRC_PTR),#1FH
Initialize shadow_variables with destination_pointer	MOV LOW(DES_PTR),#0A0H MOV HIGH(DES_PTR),#2FH

Table look-up routine under real time conditions

Action	Code	Machine cycles
Save old datapointer	PUSH DPL PUSH DPH	2 2
Load Source Pointer	MOV DPL,LOW(SRC_PTR) MOV DPH,HIGH(SRC_PTR)	2 2
Increment and check for end of table (execution time not relevant for this consideration)	INC DPTR CJNE... ...	
Fetch source data byte from ROM table	MOVC A,@DPTR	2

Save source_pointer and load destination_pointer	MOV LOW(SRC_PTR),DPL
	MOV HIGH(SRC_PTR),DPH
	MOV DPL,LOW(DES_PTR)
	MOV DPH,HIGH(DES_PTR)
		2
		2
		2
		2

Increment destination_pointer (ex. time not relevant)	INC DPTR	
Transfer byte to destination address	MOVX @DPTR,A	2
Save destination_pointer	MOV LOW(DES_PTR),DPL MOV HIGH(DES_PTR),DPH	2 2
Restore old datapointer	POP DPH POP DPL	2 2
Total execution time (machine cycles)		28

Uporaba dva pokazivača

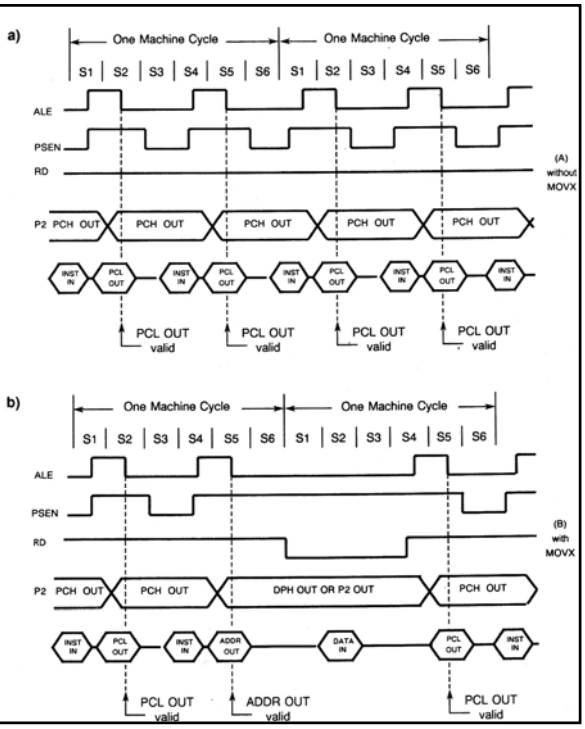
Action	Code
Initialize DPTR6 with source pointer	MOV DPSEL,#06H MOV DPTR,#1FFFH
Initialize DPTR7 with destination pointer	MOV DPSEL,#07H MOV DPTR,#2FA0H

Primjer rutine

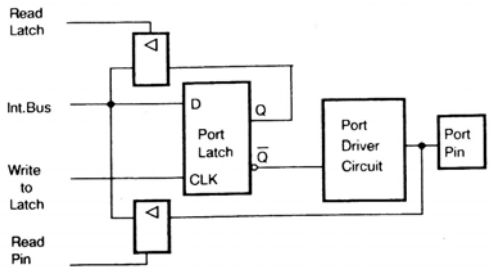
Action	Code	Machine cycles
Save old source pointer	PUSH DPSEL	2
Load source pointer	MOV DPSEL,#06H	2
Increment and check for end of table (execution time not relevant for this consideration)	INC DPTR CJNE... ...	
Fetch source data byte from ROM table	MOVC A,@DPTR	2
Save source__pointer and load destination__pointer	MOV DPSEL,#07H	2
Transfer byte to destination address	MOVX @DPTR,A	2
Save destination pointer and restore old datapointer	POP DPSEL	2
Total execution time (machine cycles)		12

19

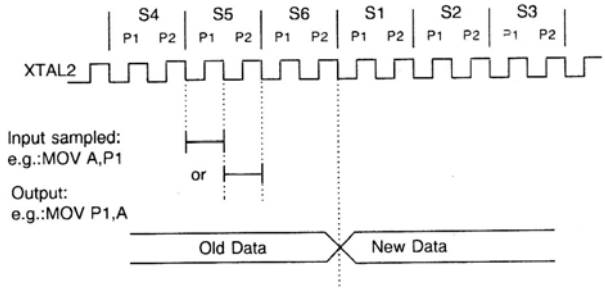
Izvođenje programa iz vanjske memorije



Pristupi



Ports 1-6 output driver



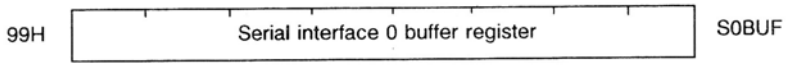
Vremenski dijagram pristupa

Serijsko sučelje

Serijsko sučelje 0

SM0	SM1	Mode	Description	Baud rate
0	0	0	Shift register	$f_{osc}/12$
0	1	1	8-bit UART	Variable
1	0	2	9-bit UART	$f_{osc}/64$ or $f_{osc}/32$
1	1	3	9-bit UART	Variable

Special Function Register S0BUF (Address 99H)



Serijsko sučelje 0

Special Function Register S0CON (Address 98H)

	9FH	9EH	9DH	9CH	9BH	9AH	99H	98H	
98H	SM0	SM1	SM20	REN0	TB80	RB80	TI0	RI0	S0CON

Bit		Function
SM0	SM1	
0	0	Serial mode 0: Shift register mode, fixed baud rate
0	1	Serial mode 1: 8-bit UART, variable baud rate
1	0	Serial mode 2: 9-bit UART, fixed baud rate
1	1	Serial mode 3: 9-bit UART, variable baud rate
SM20		Enables the multiprocessor communication feature in modes 2 and 3. In mode 2 or 3 and SM20 being set to 1, RI0 will not be activated if the received 9th data bit (RB80) is 0. In mode 1 and SM20 = 1, RI0 will not be activated if a valid stop bit has not been received. In mode 0, SM20 should be 0.
REN0		Receiver enable. Enables serial reception. Set by software to enable reception. Cleared by software to disable reception.
TB80		Transmitter bit 8. Is the 9th data bit that will be transmitted in modes 2 and 3. Set or cleared by software as desired.
RB80		Receiver bit 8. In modes 2 and 3 it is the 9th bit that was received. In mode 1, if SM20 = 0, RB80 is the stop bit that was received. In mode 0, RB80 is not used.
TI0		Transmitter interrupt. Is the transmit interrupt flag. 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. Must be cleared by software.
RI0		Receiver interrupt. Is the receive interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or during the stop bit time in the other modes, in any serial reception. Must be cleared by software.

23

Mode 0

The baud rate in mode 0 is fixed:

$$\text{Mode 0 baud rate} = \frac{\text{oscillator frequency}}{12}$$

Mode 2

The baud rate in mode 2 depends on the value of bit SMOD in special function register PCON (see figure 7-9). If SMOD = 0 (which is the value after reset), the baud rate is 1/64 of the oscillator frequency. If SMOD = 1, the baud rate is 1/32 of the oscillator frequency.

$$\text{Mode 2 baud rate} = \frac{2^{SMOD}}{64} * \text{oscillator frequency}$$

24

Modes 1 and 3

In these modes the baud rate is variable and can be generated alternatively by a dedicated baud rate generator or by timer 1.

Using the baud rate generator:

In modes 1 and 3, the SAB 80C517 can use the internal baud rate generator for serial interface 0. To enable this feature, bit BD (bit 7 of special function register ADCON0) must be set (see figure 7-10). This baud rate generator divides the oscillator frequency by 2500. Bit SMOD (PCON.7) also can be used to enable a multiply-by-two prescaler (see figure 7-9). At 12 MHz oscillator frequency, the commonly used baud rates 4800 baud (SMOD = 0) and 9600 baud (SMOD = 1) are available. The baud rate is determined by SMOD and the oscillator frequency as follows:

$$\text{Mode 1, 3 baud rate} = \frac{2^{SMOD}}{2496} * \text{oscillator frequency}$$

Special Function Register ADCON0 (Address 0D8H)

	0DFH	0DEH	0DDH	0DCH	0DBH	0DAH	0D9H	0D8H	
0D8H	BD	CLK	ADEX	BSY	ADM	MX2	MX1	MX0	ADCON

▨ These bits are not used in controlling serial interface 0.

Bit	Function
BD	Baud rate enable. When set, the baud rate in modes 1 and 3 of serial interface 0 is taken from a dedicated prescaler. Standard baud rates 4800 and 9600 baud at 12 MHz oscillator frequency can be achieved.

Using timer 1 to generate baud rates:

In mode 1 and 3 of serial channel 0 timer 1 can be used for generating baud rates. Then the baud rate is determined by the timer 1 overflow rate and the value of SMOD as follows:

$$\text{Mode 1, 3 baud rate} = \frac{2^{SMOD}}{32} \cdot \text{(Timer 1 OV-rate)}$$

The timer 1 interrupt is usually disabled in this application. The timer itself can be configured for either "timer" or "counter" operation, and in any of its operating modes. In the most typical applications, it is configured for "timer" operation in the auto-reload mode (high nibble of TMOD = 0010B). In that case, the baud rate is given by the formula:

$$\text{Mode 1, 3 baud rate} = \frac{2^{SMOD} \cdot \text{oscillator frequency}}{32 \cdot 12 \cdot (256 - (TH1))}$$

One can achieve very low baud rates with timer 1 by leaving the timer 1 interrupt enabled, configuring the timer to run as 16-bit timer (high nibble of TMOD = 0001B), and using the timer 1 interrupt for a 16-bit software reload.

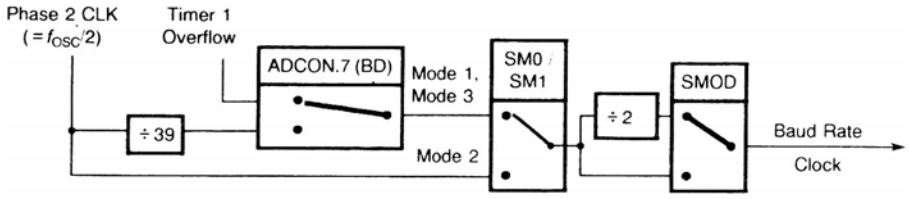
Table 7-4 lists various commonly used baud rates and shows how they can be obtained from timer 1.

27

Timer 1 Generated Commonly Used Baud Rates

Baud rate		f_{OSC} (MHz)	SMOD	Timer 1		
				C/T	Mode	Reload Value
Mode 1,3:	62.5 Kbaud	12.0	1	0	2	FFH
	19.2 Kbaud	11.059	1	0	2	FDH
	9.6 Kbaud	11.059	0	0	2	FDH
	4.8 Kbaud	11.059	0	0	2	FAH
	2.4 Kbaud	11.059	0	0	2	F4H
	1.2 Kbaud	11.059	0	0	2	E8H
	110 Baud	6.0	0	0	2	72H
	110 Baud	12.0	0	0	1	FEEBH

Generiranje brzine serijskog sučelja 0



Baud rate derived from	Interface mode	Baud rate
Timer 1 in mode 1 (see table 7-4)	1, 3	$\frac{2^{SMOD}}{2} \cdot \frac{1}{16} \cdot$ (timer 1 overflow rate)
Timer 1 in mode 2 (see table 7-4)	1, 3	$\frac{2^{SMOD}}{2} \cdot \frac{1}{16} \cdot \frac{f_{OSC}}{12 \cdot (256 - (TH1))}$
Oscillator	2	$\frac{2^{SMOD}}{2} \cdot \frac{1}{16} \cdot \frac{f_{OSC}}{2}$
BD	1, 3	$\frac{2^{SMOD}}{2} \cdot \frac{f_{OSC}}{1250}$

29

Serijsko sučelje 1

```
LOOP:   MOV  A,S1CON
        JNB  ACC.0,LOOP    ;Testing of RI1
        ANL  S1CON,#0FEH   ;Resetting of RI1
```

30

Special Function Register S1CON (Address 9BH)

9BH	SM	–	SM21	REN1	TB81	RB81	TI1	RI1	S1CON

Bit	Function
SM	SM = 0: serial mode A; 9-bit UART SM = 1: serial mode B; 8-bit UART
SM21	Enables the multiprocessor communication feature in mode A. If SM21 is set to 1, RI1 will not be activated if the received 9th data bit (RB81) is 0. In mode B, if SM21 = 1, RI1 will not be activated if a valid stop bit was not received.
REN1	Receiver enable of interface 1. Enables serial reception. Set by software to enable reception. Cleared by software to disable reception.
TB81	Transmitter bit 8 of interface 1. Is the 9th data bit that will be transmitted in mode A. Set or cleared by software as desired.
RB81	Receiver bit 8 of interface 1. Is the 9th data bit that was received in mode A. In mode B, if SM21 = 0, RB81 is the stop bit that was received.
TI1	Transmitter interrupt of interface 1. Is the transmit interrupt flag. Set by hardware at the beginning of the stop bit in any serial transmission. Must be cleared by software.
RI1	Receiver interrupt of interface 1. Is the receive interrupt flag. Set by hardware at the halfway through the stop bit time in any serial reception. Must be cleared by software.

31

Special Function Register S1CON (Address 9BH)

9BH	SM	–	SM21	REN1	TB81	RB81	TI1	RI1	S1CON

Bit	Function
SM	SM = 0: serial mode A; 9-bit UART SM = 1: serial mode B; 8-bit UART
SM21	Enables the multiprocessor communication feature in mode A. If SM21 is set to 1, RI1 will not be activated if the received 9th data bit (RB81) is 0. In mode B, if SM21 = 1, RI1 will not be activated if a valid stop bit was not received.
REN1	Receiver enable of interface 1. Enables serial reception. Set by software to enable reception. Cleared by software to disable reception.
TB81	Transmitter bit 8 of interface 1. Is the 9th data bit that will be transmitted in mode A. Set or cleared by software as desired.
RB81	Receiver bit 8 of interface 1. Is the 9th data bit that was received in mode A. In mode B, if SM21 = 0, RB81 is the stop bit that was received.
TI1	Transmitter interrupt of interface 1. Is the transmit interrupt flag. Set by hardware at the beginning of the stop bit in any serial transmission. Must be cleared by software.
RI1	Receiver interrupt of interface 1. Is the receive interrupt flag. Set by hardware at the halfway through the stop bit time in any serial reception. Must be cleared by software.

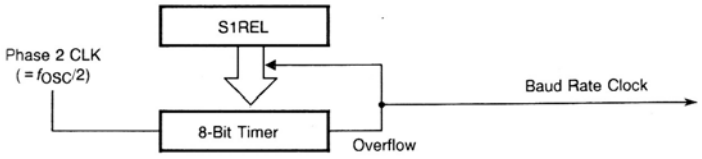
32

Special Function Register S1BUF (Address 9CH)

9CH	Serial interface 1 buffer register	S1BUF

$$\text{Mode A, B baud rate} = \frac{\text{oscillator frequency}}{32 * (256 - \text{S1REL})}$$

Generiranje brzine serijskog sučelja 1



Special Function Register S1REL (Address 9DH)

9DH	Serial interface 1 reload register	S1REL

33

Vremenski sklopovi

Special Function Register TCON (Address 88H)

	8FH	8EH	8DH	8CH	8BH	8AH	89H	88H	
88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	TCON

▨ These bits are not used in controlling timer/counter 0 and 1.

Bit	Function
TR0	Timer 0 run control bit. Set/cleared by software to turn timer/counter 0 on/off.
TF0	Timer 0 overflow flag. Set by hardware on timer/counter overflow. Cleared by hardware when processor vectors to interrupt routine.
TR1	Timer 1 run control bit. Set/cleared by software to turn timer/counter 1 on/off.
TF1	Timer 1 overflow flag. Set by hardware on timer/counter overflow. Cleared by hardware when processor vectors to interrupt routine.

34

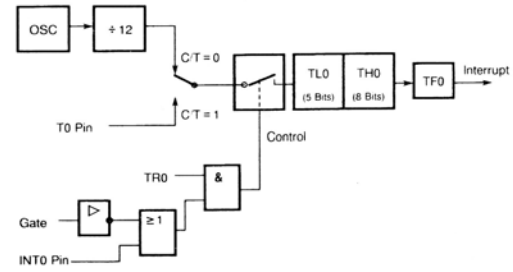
Special Function Register TMOD (Address 89H)

89H	GATE	C/T#	M1	M0	GATE	C/T#	M1	M0	TMOD
	Timer 1				Timer 0				

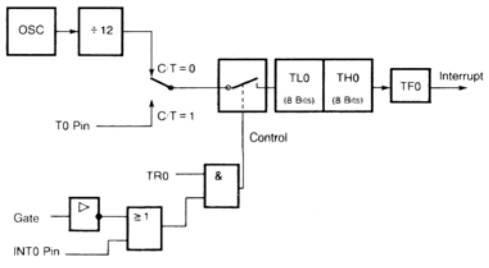
Timer/counter 0/1 mode control register.

Bit		Function
Gate		Gating control. When set, timer/counter "x" is enabled only while "INTx#" pin is high and "TRx" control bit is set. When cleared timer "x" is enabled whenever "TRx" control bit is set.
C/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	
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. TH00 is an 8-bit timer only controlled by timer 1 control bits.
1	1	Timer 1: Timer/counter 1 stops

35

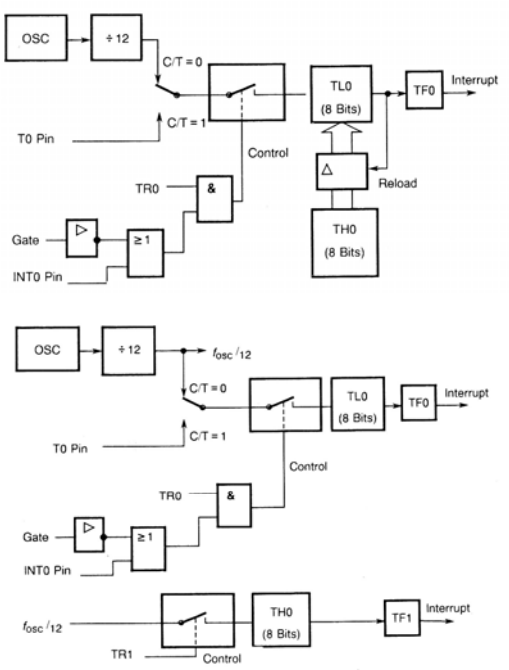


Načini rada 0 i 1

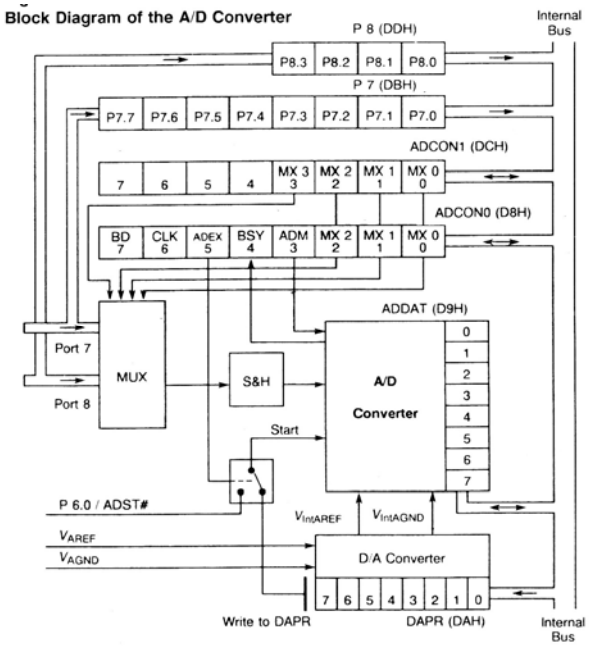


36

Načini rada 2 i 3



A/D pretvorba



Special Function Register ADCON0 (Address 0D8H)

	0DFH	0DEH	0DDH	0DCH	0DBH	0DAH	0D9H	0D8H	
0D8H	BD	CLK	ADEX	BSY	ADM	MX2	MX1	MX0	ADCON0

▨ These bits are not used in controlling A/D converter functions.

Bit	Function
MX0 **MX1** **MX2** **MX3**	Select 12 input channels of the A/D converter.
ADM	A/D conversion mode. When set, a continuous conversion is selected. If ADM = 0, the converter stops after one conversion.
BSY	Busy flag. This flag indicates whether a conversion is in progress (BSY = 1). The flag is cleared by hardware when the conversion is completed.
ADEX	Internal/external start of conversion. When set, the external start of conversion by P6.0/ADST# is enabled.

39

Special Function Register ADCON1 (Address 0DCH)

0DCH	–	–	–	–	MX3	MX2	MX1	MX0	ADCON1

Selection of the Analog Input Channels

MX3	MX2	MX1	MX0	Selected channel	Pin
0	0	0	0	Analog input 0	P7.0
0	0	0	1	Analog input 1	P7.1
0	0	1	0	Analog input 2	P7.2
0	0	1	1	Analog input 3	P7.3
0	1	0	0	Analog input 4	P7.4
0	1	0	1	Analog input 5	P7.5
0	1	1	0	Analog input 6	P7.6
0	1	1	1	Analog input 7	P7.7
1	X *)	0	0	Analog input 8	P8.0
1	X	0	1	Analog input 9	P8.1
1	X	1	0	Analog input 10	P8.2
1	X	1	1	Analog input 11	P8.3

*) X means that the value may be 1 or 0.

40

Special Function Register ADDAT (Address 0D9H)

0D9H	Conversion result	ADDAT

$$V_{IntAGND} = V_{AGND} + \frac{DAPR\ (.3\text{-}.0)}{16} (V_{AREF} - V_{AGND})$$

with DAPR (.3-.0) < CH;

$$V_{IntAREF} = V_{AGND} + \frac{DAPR\ (.7\text{-}.4)}{16} (V_{AREF} - V_{AGND})$$

with DAPR (.7-.4) > 3H;

41

Special Function Register DAPR (Address DAH)

0DAH	Programming of $V_{IntAREF}$	Programming of $V_{IntAGND}$	DAPR

D/A converter program register. Each 4-bit nibble is used to program the internal reference voltages. Write-access to DAPR starts conversion.

$$V_{IntAGND} = V_{AGND} + \frac{DAPR\ (.3\text{-}.0)}{16} (V_{AREF} - V_{AGND})$$

with DAPR (.3-.0) < 13;

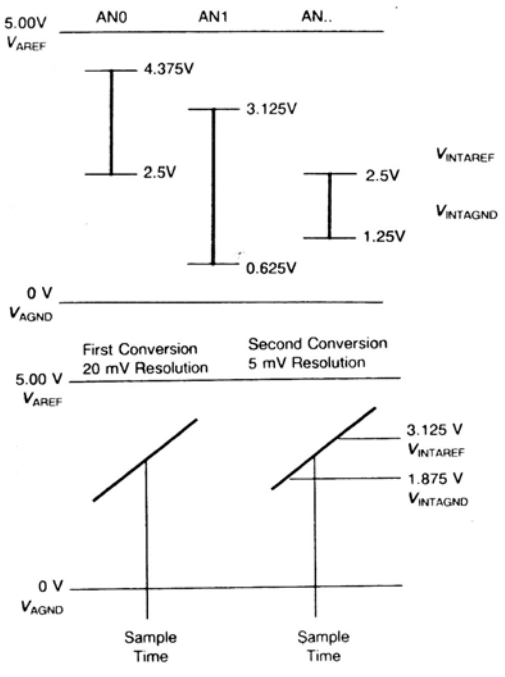
$$V_{IntAREF} = V_{AGND} + \frac{DAPR\ (.7\text{-}.4)}{16} (V_{AREF} - V_{AGND})$$

with DAPR (.7-.4) > 3;

42

Step	DAPR (.3-.0) DAPR (.7-.4)	$V_{IntAGND}$	$V_{IntAREF}$
0	0000	0.0	5.0
1	0001	0.3125	–
2	0010	0.625	–
3	0011	0.9375	–
4	0100	1.25	1.25
5	0101	1.5625	1.5625
6	0110	1.875	1.875
7	0111	2.1875	2.1875
8	1000	2.5	2.5
9	1001	2.8125	2.8125
10	1010	3.125	3.125
11	1011	3.4375	3.4375
12	1100	3.75	3.75
13	1101	–	4.0625
14	1110	–	4.375
15	1111	–	4.68754

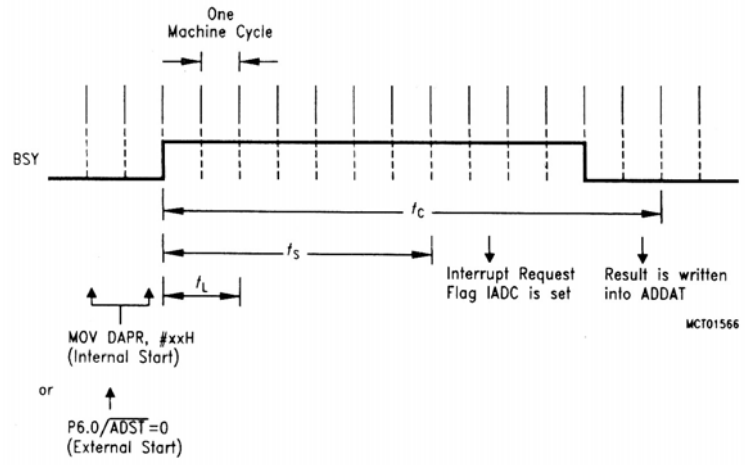
43



Podešavanje referentnog napona i povećanje razlučivosti drugom pretvorbom

44

Vremenski dijagram A/D pretvorbe



45

Sigurnosni načini rada (idle)

Special Function Register PCON (Address 87H)

87H	SMOD	PDS	IDLS	SD	GF1	GF0	PDE	IDLE	PCON

▨ These bits are not used in controlling the power saving modes.

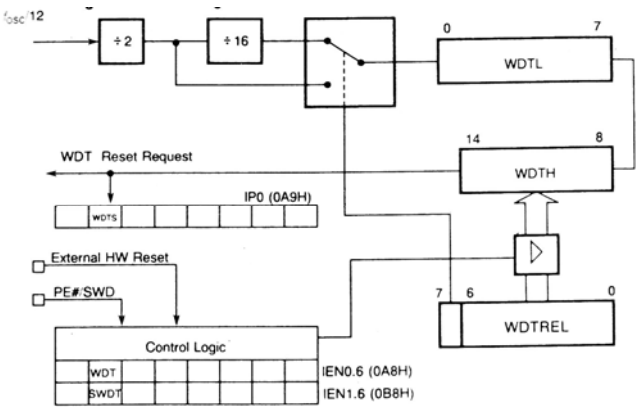
Bit	Function
PDS	Power-down start bit. The instruction that sets the PDS flag bit is the last instruction before entering the power-down mode.
IDLS	IDLE start bit. The instruction that sets the IDLS flag bit is the last instruction before entering the idle mode.
SD	When set, the slow-down mode is enabled.
GF1	General purpose flag
GF0	General purpose flag
PDE	Power-down enable bit. When set, starting the power-down mode is enabled.
IDLE	Idle mode enable bit. When set, starting the idle mode is enabled.

Sigurnosni načini rada (power down i slow down)

46

Sigurnosni načini rada (budilnički sklop)

WDTREL =	Time-out period	Comments
00 H	65.535 ms	This is the default value and coincides with the watchdog period of the SAB 80515
80 H	1.1 s	Maximum time period
7F H	512 µs	Minimum time period



47

Special Function Register WDTREL

086H | Watchdog timer reload register | WDTREL

Bit	Function
WDTREL.7	Prescaler select bit. When set, the watchdog is clocked through an additional divide-by-16 prescaler (see figure 7-58).
WDTREL.6 to WDTREL.0	Seven bit reload value for the high-byte of the watchdog timer. This value is loaded to the WDT when a refresh is triggered by a consecutive setting of bits WDT and SWDT.

48

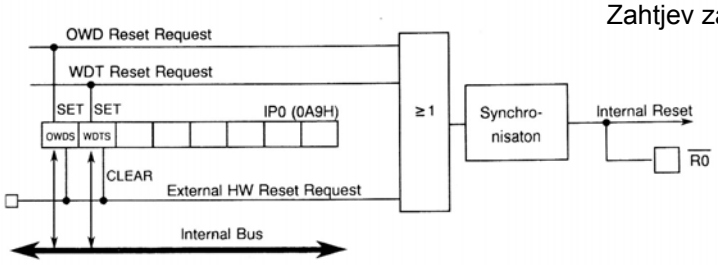
Special Function Register IEN0

	0AFH	0AEH	0ADH	0ACH	0ABH	0AAH	0A9H	0A8H	
0A8H	EAL	WDT	ET2	ES0	ET1	EX1	ET0	EX0	IEN0

▨ These bits are not used in controlling the fail-safe mechanisms.

Bit	Function
WDT	Watchdog timer refresh flag. Set to initiate a refresh of the watchdog timer. Must be set directly before SWDT is set to prevent an unintentional refresh of the watchdog timer.

Zahtjev za resetiranje



49

Special Function Register IEN1

	0BFH	0BEH	0BDH	0BCH	0BBH	0BAH	0B9H	0B8H	
0B8H	EXEN2	SWDT	EX6	EX5	EX4	EX3	EX2	EADC	IEN1

▨ These bits are not used in controlling the fail-safe mechanisms.

Bit	Function
SWDT	Watchdog timer start flag. Set to activate the watchdog timer. When directly set after setting WDT, a watchdog timer refresh is performed.

Special Function Register IP0

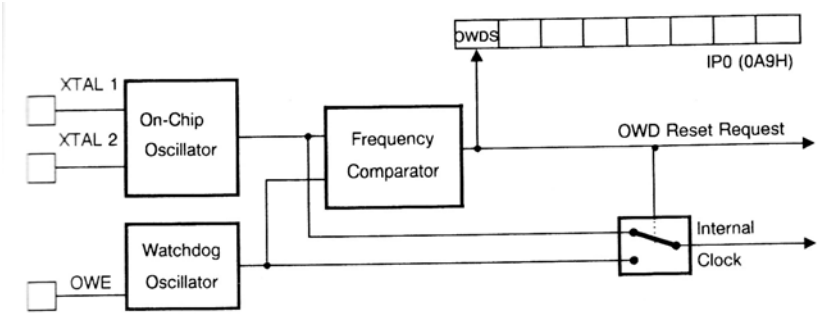
0A9H	OWDS	WDTS	IP0.5	IP0.4	IP0.3	IP0.2	IP0.1	IP0.0	IP0

▨ These bits are not used in controlling the fail-safe mechanisms.

Bit	Function
OWDS	Oscillator watchdog timer status flag. Set by hardware when an oscillator watchdog reset occurred. Can be cleared or set by software.
WDTS	Watchdog timer status flag. Set by hardware when a watchdog timer reset occurred. Can be cleared or set by software.

50

Budilnički oscilator



51

Instrukcijski skup

Načini adresiranja

Addressing modes	Associated memory spaces
Register addressing	R0 through R7 of selected register bank, ACC, B, CY (Bit), DPTR
Direct addressing	Lower 128 bytes of internal RAM, special function registers
Immediate addressing	Program memory
Register indirect addressing	Internal RAM (@R1, @R0, SP), external data memory (@R1, @R0, @DPTR)
Base register plus index register addressing	Program memory (@DPTR + A, @PC + A)

52

Instrukcijski skup

Djelovanje instrukcija na zastavice

Instruction	Flag			Instruction	Flag		
	CY	OV	AC		CY	OV	AC
ADD	X	X	X	SETB C	1		
ADDC	X	X	X	CLR C	0		
SUBB	X	X	X	CPL C	X		
MUL	0	X		ANL C,bit	X		
DIV	0	X		ANL C,/bit	X		
DA	X			ORL C,bit	X		
RRC	X			ORL C,/bit	X		
RLC	X			MOV C,bit	X		
CJNE	X						

53

Notes on Data Addressing Modes

Rn	-	Working register R0-R7
direct	-	128 internal RAM locations, any I/O port, control or status register
@Ri	-	Indirect internal or external RAM location addressed by register R0 or R1
#data	-	8-bit constant included in instruction
#data 16	-	16-bit constant included as bytes 2 and 3 of instruction
bit	-	128 software flags, any bitaddressable I/O pin, control or status bit
A	-	Accumulator

54

Notes on Program Addressing Modes

addr 16 - Destination address for LCALL and LJMP may be anywhere within the 64-Kbyte program memory address space.

addr11 - Destination address for ACALL and AJMP will be within the same 2-Kbyte page of program memory as the first byte of the following instruction.

rel - SJMP and all conditional jumps include an 8 bit offset byte. Range is +127/-128 bytes relative to the first byte of the following instruction.

55

Mnemonic	Description	Byte	Cycle
Arithmetic operations			
ADD A,Rn	Add register to accumulator	1	1
ADD A,direct	Add direct byte to accumulator	2	1
ADD A, @Ri	Add indirect RAM to accumulator	1	1
ADD A,#data	Add immediate data to accumulator	2	1
ADDC A,Rn	Add register to accumulator with carry flag	1	1
ADDC A,direct	Add direct byte to A with carry flag	2	1
ADDC A, @Ri	Add indirect RAM to A with carry flag	1	1
ADDC A, #data	Add immediate data to A with carry flag	2	1
SUBB A,Rn	Subtract register from A with borrow	1	1
SUBB A,direct	Subtract direct byte from A with borrow	2	1
SUBB A,@Ri	Subtract indirect RAM from A with borrow	1	1
SUBB A,#data	Subtract immediate data from A with borrow	2	1
INC A	Increment accumulator	1	1
INC Rn	Increment register	1	1
INC direct	Increment direct byte	2	1
INC @Ri	Increment indirect RAM	1	1
DEC A	Decrement accumulator	1	1
DEC Rn	Decrement register	1	1
DEC direct	Decrement direct byte	2	1
DEC @Ri	Decrement indirect RAM	1	1
INC DPTR	Increment data pointer	1	2
MUL AB	Multiply A and B	1	4
DIV AB	Divide A by B	1	4
DA A	Decimal adjust accumulator	1	1

56

Mnemonic		Description	Byte	Cycle
Logic operations				
ANL	A,Rn	AND register to accumulator	1	1
ANL	A,direct	AND direct byte to accumulator	2	1
ANL	A,@Ri	AND indirect RAM to accumulator	1	1
ANL	A,#data	AND immediate data to accumulator	2	1
ANL	direct,A	AND accumulator to direct byte	2	1
ANL	direct,#data	AND immediate data to direct byte	3	2
ORL	A,Rn	OR register to accumulator	1	1
ORL	A,direct	OR direct byte to accumulator	2	1
ORL	A,@Ri	OR indirect RAM to accumulator	1	1
ORL	A,#data	OR immediate data to accumulator	2	1
ORL	direct,A	OR accumulator to direct byte	2	1
ORL	direct,#data	OR immediate data to direct byte	3	2
XRL	A,Rn	Exclusive OR register to accumulator	1	1
XRL	A,direct	Exclusive OR direct byte to accumulator	2	1
XRL	A,@Ri	Exclusive OR indirect RAM to accumulator	1	1
XRL	A,#data	Exclusive OR immediate data to accumulator	2	1
XRL	direct,A	Exclusive OR accumulator to direct byte	2	1
XRL	direct,#data	Exclusive OR immediate data to direct byte	3	2
CLR	A	Clear accumulator	1	1
CPL	A	Complement accumulator	1	1
RL	A	Rotate accumulator left	1	1
RLC	A	Rotate accumulator left through carry	1	1
RR	A	Rotate accumulator right	1	1
RRC	A	Rotate accumulator right through carry	1	1
SWAP	A	Swap nibbles within the accumulator	1	1

57

Mnemonic		Description	Byte	Cycle
Data transfer				
MOV	A,Rn	Move register to accumulator	1	1
MOV	A,direct	Move direct byte to accumulator	2	1
MOV	A,@Ri	Move indirect RAM to accumulator	1	1
MOV	A,#data	Move immediate data to accumulator	2	1
MOV	Rn,A	Move accumulator to register	1	1
MOV	Rn,direct	Move direct byte to register	2	2
MOV	Rn,#data	Move immediate data to register	2	1
MOV	direct,A	Move accumulator to direct byte	2	1
MOV	direct,Rn	Move register to direct byte	2	2
MOV	direct,direct	Move direct byte to direct byte	3	2
MOV	direct,@Ri	Move indirect RAM to direct byte	2	2
MOV	direct,#data	Move immediate data to direct byte	3	2
MOV	@Ri,A	Move accumulator to indirect RAM	1	1
MOV	@Ri,direct	Move direct byte to indirect RAM	2	2
MOV	@Ri, #data	Move immediate data to indirect RAM	2	1
MOV	DPTR, #data16	Load data pointer with a 16-bit constant	3	2
MOVC	A,@A + DPTR	Move code byte relative to DPTR to accumulator	1	2
MOVC	A,@A + PC	Move code byte relative to PC to accumulator	1	2
MOVX	A,@Ri	Move external RAM (8-bit addr.) to A	1	2
MOVX	A,@DPTR	Move external RAM (16-bit addr.) to A	1	2
MOVX	@Ri,A	Move A to external RAM (8-bit addr.)	1	2
MOVX	@DPTR,A	Move A to external RAM (16-bit addr.)	1	2
PUSH	direct	Push direct byte onto stack	2	2
POP	direct	Pop direct byte from stack	2	2
XCH	A,Rn	Exchange register with accumulator	1	1
XCH	A,direct	Exchange direct byte with accumulator	2	1
XCH	A,@Ri	Exchange indirect RAM with accumulator	1	1
XCHD	A,@Ri	Exchange low-order nibble indir. RAM with A	1	1

58

Mnemonic	Description	Byte	Cycle
Boolean variable manipulation			
CLR C	Clear carry flag	1	1
CLR bit	Clear direct bit	2	1
SETB C	Set carry flag	1	1
SETB bit	Set direct bit	2	1
CPL C	Complement carry flag	1	1
CPL bit	Complement direct bit	2	1
ANL C,bit	AND direct bit to carry flag	2	2
ANL C,/bit	AND complement of direct bit to carry	2	2
ORL C,bit	OR direct bit to carry flag	2	2
ORL C,/bit	OR complement of direct bit to carry	2	2
MOV C,bit	Move direct bit to carry flag	2	1
MOV bit,C	Move carry flag to direct bit	2	2
Program and machine control			
ACALL addr11	Absolute subroutine call	2	2
LCALL addr16	Long subroutine call	3	2
RET	Return from subroutine	1	2
RETI	Return from interrupt	1	2
AJMP addr11	Absolute jump	2	2
LJMP addr16	Long jump	3	2
SJMP rel	Short jump (relative addr.)	2	2
JMP @A + DPTR	Jump indirect relative to the DPTR	1	2
JZ rel	Jump if accumulator is zero	2	2
JNZ rel	Jump if accumulator is not zero	2	2
JC rel	Jump if carry flag is set	2	2
JNC rel	Jump if carry flag is not set	2	2
JB bit,rel	Jump if direct bit is set	3	2
JNB bit,rel	Jump if direct bit is not set	3	2
JBC bit,rel	Jump if direct bit is set and clear bit	3	2

59

Mnemonic	Description	Byte	Cycle
Program and machine control (cont'd)			
CJNE A,direct,rel	Compare direct byte to A and jump if not equal	3	2
CJNE A,#data,rel	Compare immediate to A and jump if not equal	3	2
CJNE Rn,#data,rel	Compare immed. to reg. and jump if not equal	3	2
CJNE @Ri,#data,rel	Compare immed. to ind. and jump if not equal	3	2
DJNZ Rn,rel	Decrement register and jump if not zero	2	2
DJNZ direct,rel	Decrement direct byte and jump if not zero	3	2
NOP	No operation	1	1

60

Primjeri programiranja

```
; A/D pretvorba
        ORG     8000H
ADPR:   MOV     ADCON,#80H      ;0.kanal, pojedinacno, za kont. je #18H
        MOV     DAPR,#00H       ;ref. napon je od 0-5 V
WTBSY:  JB      BSY,WTBSY       ;je li pretvorba obavljena
        MOV     A,ADDAT                 ;uzmi rezultat
        CPL     A               ;napravi obradu rezultata
        MOV     P1,A            ;prikazi rezultat na izl. prilazu
        JMP     ADPR            ;nastavi pretvorbu
        END
```

61

Primjeri programiranja

```
; Prema postojecem programu za A/D pretvorbu napisati program koji ovisno o
; postavljenoj preklopkici na ulaznom portu uzorkuje analogni signal sa nultog
; odnosno sa sa prvog kanala A/D pretvornika ( port P5 ).
;
        ORG     8000H ; ako program pocinje iznad 8000H mozemo ga izvesti na PC
ADDPR:  MOV     ADCON,#80H; definiramo nulti kanal pojedinacno
        MOV     ADCON,#81H; def. 1 kanal
        MOV     DAPR,#40H ; ref. napon 0-5V
PROV:   MOV     A,P1
        JNZ     ACC.0,WTBSY
        JZ      ACC.1,WTBSYY
WTBSY:  JB      BSY,WTBSY ; je li pretvorba gotova
        MOV     A,ADDAT   ; uzmi mi rezultat
        MOV     P5,A
        JMP     ADDPR     ; nastavi
WTBSYY:JB       BSY,WTBSYY
        MOV     A,ADDAT
        MOV     P5,A
        JMP     ADDPR
        END
```

62

Primjeri programiranja

```
;Program podrzava serijsku vezu po standardu RS-232
S1CON   EQU     09BH    ;osobine prijenosa
S1BUF   EQU     09CH    ;spremnik podataka veze
S1REL   EQU     09DH    ;reguliranje brzine prijenosa
KON1    EQU     041H    ;znak po volji
KON2    EQU     042H    ;znak po volji
        ORG     8000H
        MOV     KON1,#0F0H
        CALL    PREDAJA
        CALL    PRIJEM
        MOV     P1,KON2
PRIJEM:         MOV     A,S1CON
        JNB     ACC.0,PRIJEM  ;cekanje na prijem znaka
        ANL     S1CON,#0FEH   ;reset RI1
        MOV     A,S1BUF
        MOV     KON2,A
        RET
PREDAJA:PUSH    ACC
CEK:    MOV     A,S1CON
        JNB     ACC.1,PRIJEM  ;cekanje na slanje znaka
        ANL     S1CON,#0FDH   ;reset TI1
        POP     ACC
        MOV     A,KON1
        MOV     S1BUF,A
        RET
        END
```

63

Primjeri programiranja

```
; Postaviti zeljenu rijec na P1 poslati je na drugi mikro-upravljac sa drugog
; mikro-upravljaca odgovoriti slanjem te komplementirane rijeci na prvi mikro-
; Na kraju prijema neka svijetli jedna dioda po izboru, a na kraju slanja neka
; druga LED. Komunikacija je po RS485 (Sucelje 1).
; Program je za MASTER sudionika a SLAVE ima obrnut redoslijed

S1CON   EQU     09BH
S1BUF   EQU     09CH
S1REL   EQU     09DH
KON1    EQU     041H
KON2    EQU     042H
        ORG     8000H
SERINI:MOV      S1REL,#0FEH ;programiramo brzinu od 100 kbita u sekundi
        MOV     S1CON,#092H
        ANL     S1CON,#0FEH
SLANJE:MOV      A,S1CON
        JNB     ACC.1,SLANJE
        ANL     S1CON,#0FDH
        ANL     S1CON,#0FEH
        SETB    P1.0
        CLR     P1.0
        MOV     S1BUF,KON1
        ANL     S1CON,#0FDH
        MOV     R0,#064H
        SETB    P1.7
ODBR:   DJNZ    R0,ODBR
TST2:   MOV     A,S1CON
        CLR     P1.0
PRIJEM:SETB     P5.1
        CLR     P5.0
        ANL     S1CON,#0FEH
        SETB    P1.6
REI2:   MOV     A,S1CON
        JB      ACC.0,REIOK
        SJMP    REI2
REIOK:  MOV     A,S1BUF
        MOV     KON2,A
        ANL     S1CON,#0FEH
        CLR     P1.1
        END
```

64

Primjeri programiranja

```
;DSMCWDT.ASM
; Aktiviranje budilnika
; Osvjeľavanje se obavlja pozivanjem ovog potprograma nakon odreÅenog
; vremena izvoÅenja
WDTREL EQU    086H          ;registar upravljanja budilnikom
       ORG    8000H
       MOV    WDTREL,#080H  ;vrijednost za 1.1 s
       SETB   WDT           ;osvjeľavanje
       SETB   SWDT          ;pokretanje
       END
```

65