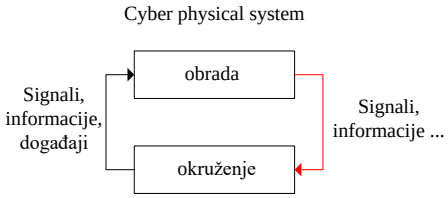
Brojači

1

Da se podsetimo



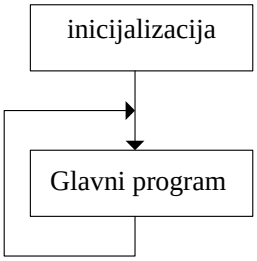
Okruženje – fizički svet iz kojeg je potrebno dobiti informacije a po potrebni i delovati na njega.

Obrada – naš embedded sistem

2

Kako izgleda na globalnom nivou tipičan program koji izvodi naš namenski sistem?

Tipičan program

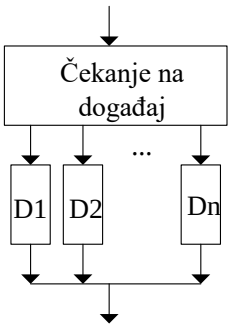


Inicijalizacija – po resetu sistema se inicijalizuju po potrebi svi delovi sistema
Na primer: već sretali neophodnost inicijalizacije kod paralelnih portova

Glavni program – sastoji se od više programskih celina u kojima se izvode neophodne programske rutine za pravilan i očekivan rad našeg sistema

3

Glavni program



U opštem slučaju glavni program se izvršava u petlji u kojoj se registruju događaji, promene, u sistemu

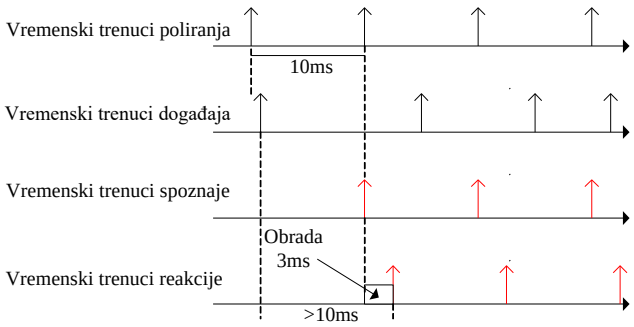
Da li se desio događaj, promena, glavni program saznaje
-poliranjem
-prekidima

4

Poliranje da se podsetimo

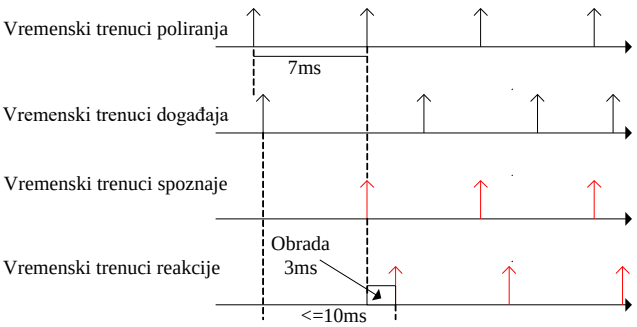
Ako na neki asinhroni događaj mora da se reaguje u vremenskom periodu od 10ms i ako obrada tog događaja traje 3 ms koliko često mora da se polira?

Ako poliramo svakih 10ms



5

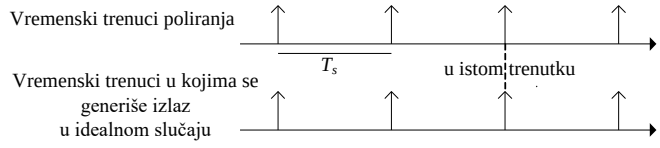
Ako poliramo svakih 7ms (10-3)

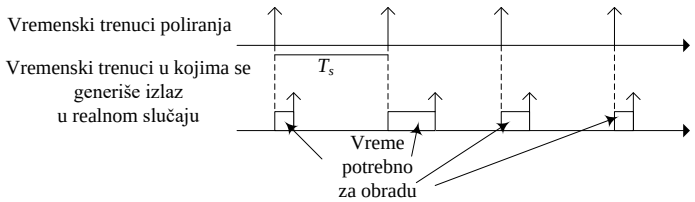


U praksi nećemo raditi na “ivici”. Verovatno ćemo izabrati poliranje na 5ms.

6

Rad u realnom vremenu – na događaj u sistemu mora da se odgovori u zadatom vremenskom intervalu – teorema odabiranja …





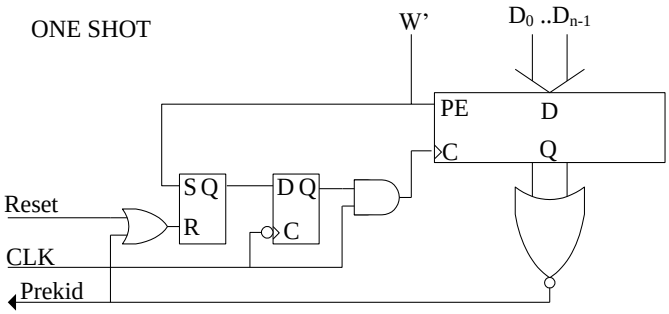
Treba nam element, periferija, u sistemu koji će dati informaciju da je došao trenutak poliranja, odabiranja.

7

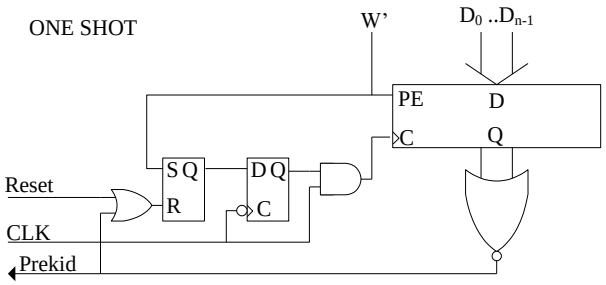
Odgovor: Brojači koji broje sa definisanim, poznatim, taktnim signalom.

Pošto je signal takta poznat i vremenski definisan onda se ovakvi brojači nazivaju i tajmerima – meračima vremena.

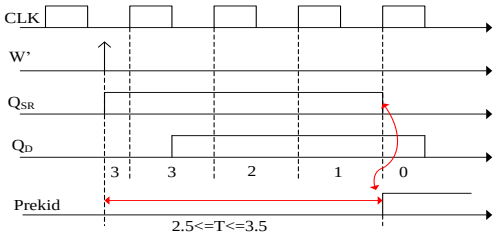
Primer: Takt 1ms - brojač odbrojao do 3 – prošlo 3ms.

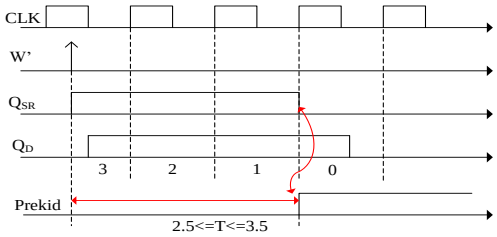


8



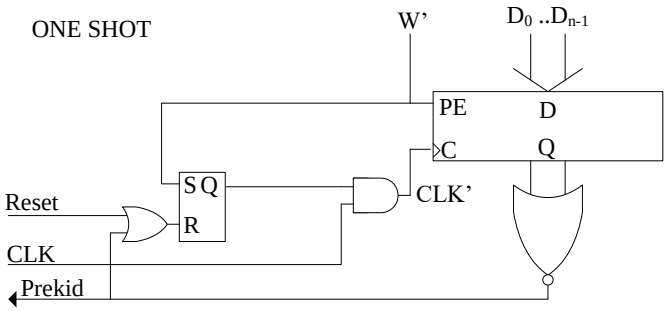
Hoćemo da izmerimo 3 Tclk. Upišemo 3.

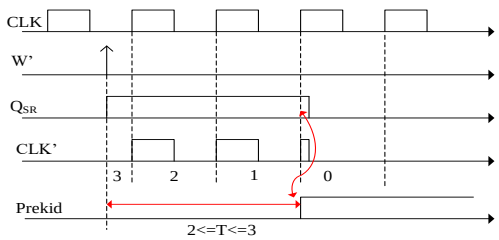


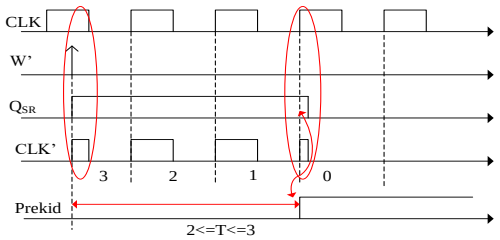


9

Ništa se ne dobija uklanjanjem sinhronizacije, odnosno D flipflopa.

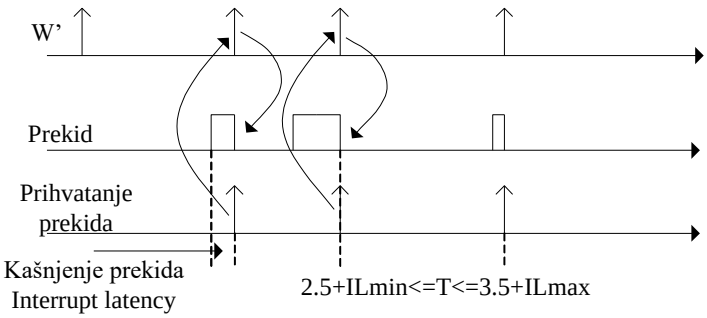






10

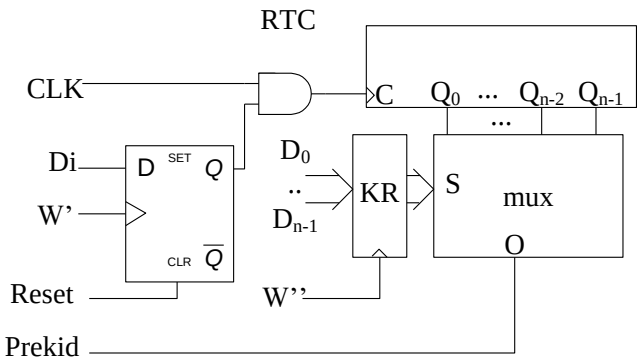
Ako ovakav tajmer koristimo za generisanje realnog vremena



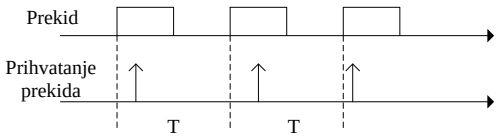
1. Pojavljuje se sistemska greška
2. Pojavljuje se akumulacija greške

Ne može kao RTC – real time clock
Pitanje koliko je dobro i za teoremu odabiranja

Najčešća realizacija brojača
koji mogu da posluže kao RTC
pa se tako i najčešće zovu.

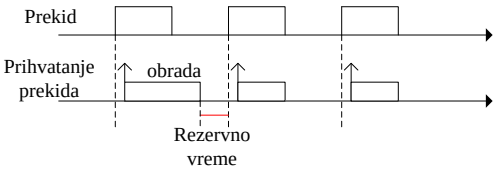


Ako ovakav brojač koristimo za generisanje realnog vremena



1. Sistemska greška ne postoji
2. Akumulacija greške ne postoji

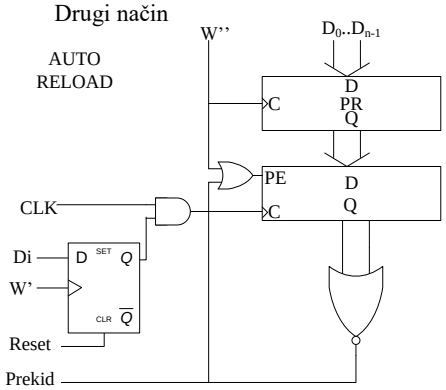
Trenutci prihvatanja prekida imaju jitter – podrhtavanje. Može se izbeći

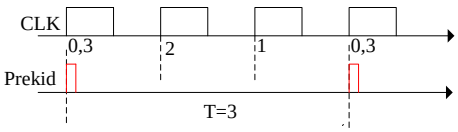


U „rezervnom vremenu“ procesor ulazi u režim u kojem miruje i čeka prekid - „uvek isto“ vreme kašnenja prekida

13

Drugi način



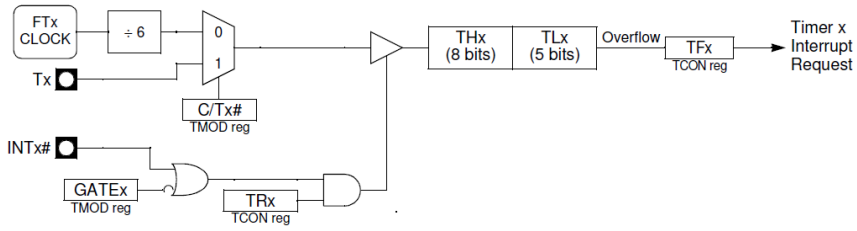


Mana:
Jedan interval = jedan brojač
Brojači – veliki potrošači energije
U okviru SoCa često više ovakvih konfiguracija.
Prednost: nema potrebne intervencije programa osim u inicijalizaciji

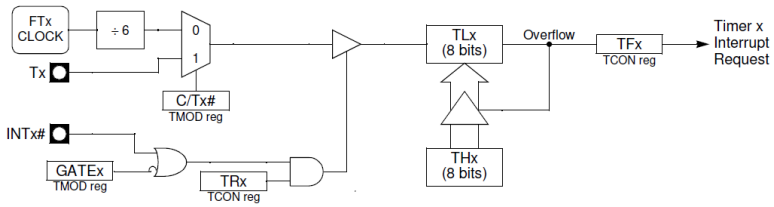
Koriste se
Za generisanje vremenskih intervala ali i kao programabilni delitelji učestanosti $\frac{Fclk}{k}$

14

Kako to izgleda u 89c51



RTC
TL - preskaler



Autoreload

15

TMOD (S:89h)
Timer/Counter Mode Control Register

7	6	5	4	3	2	1	0
GATE1	C/T1#	M11	M01	GATE0	C/T0#	M10	M00

Bit Number	Bit Mnemonic	Description
7	GATE1	**Timer 1 Gating Control Bit** Clear to enable Timer 1 whenever TR1 bit is set. Set to enable Timer 1 only while INT1# pin is high and TR1 bit is set.
6	C/T1#	**Timer 1 Counter/Timer Select Bit** Clear for Timer operation: Timer 1 counts the divided-down system clock. Set for Counter operation: Timer 1 counts negative transitions on external pin T1.
5	M11	**Timer 1 Mode Select Bits** M11 M01 Operating mode 0 0 Mode 0: 8-bit Timer/Counter (TH1) with 5-bit prescaler (TL1). 0 1 Mode 1: 16-bit Timer/Counter. 1 0 Mode 2: 8-bit auto-reload Timer/Counter (TL1) (1) 1 1 Mode 3: Timer 1 halted. Retains count
4	M01	
3	GATE0	**Timer 0 Gating Control Bit** Clear to enable Timer 0 whenever TR0 bit is set. Set to enable Timer/Counter 0 only while INT0# pin is high and TR0 bit is set.
2	C/T0#	**Timer 0 Counter/Timer Select Bit** Clear for Timer operation: Timer 0 counts the divided-down system clock. Set for Counter operation: Timer 0 counts negative transitions on external pin T0.
1	M10	**Timer 0 Mode Select Bit** M10 M00 Operating mode 0 0 Mode 0: 8-bit Timer/Counter (TH0) with 5-bit prescaler (TL0). 0 1 Mode 1: 16-bit Timer/Counter. 1 0 Mode 2: 8-bit auto-reload Timer/Counter (TL0) (2) 1 1 Mode 3: TL0 is an 8-bit Timer/Counter TH0 is an 8-bit Timer using Timer 1's TR0 and TF0 bits.
0	M00	

1. Reloaded from TH1 at overflow.
2. Reloaded from TH0 at overflow.

Reset Value = 0000 0000b

TCON (S:88h)
Timer/Counter Control Register

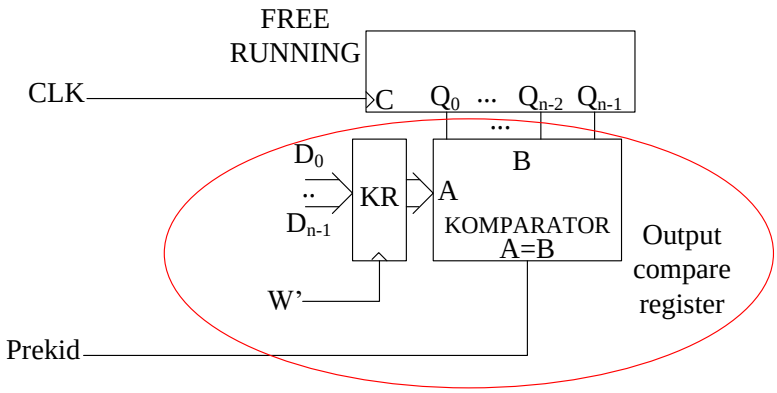
7	6	5	4	3	2	1	0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

Bit Number	Bit Mnemonic	Description
7	TF1	**Timer 1 Overflow Flag** Cleared by hardware when processor vectors to interrupt routine. Set by hardware on Timer/Counter overflow, when Timer 1 register overflows.
6	TR1	**Timer 1 Run Control Bit** Clear to turn off Timer/Counter 1. Set to turn on Timer/Counter 1.
5	TF0	**Timer 0 Overflow Flag** Cleared by hardware when processor vectors to interrupt routine. Set by hardware on Timer/Counter overflow, when Timer 0 register overflows.
4	TR0	**Timer 0 Run Control Bit** Clear to turn off Timer/Counter 0. Set to turn on Timer/Counter 0.
3	IE1	**Interrupt 1 Edge Flag** Cleared by hardware when interrupt is processed if edge-triggered (see IT1). Set by hardware when external interrupt is detected on INT1# pin.
2	IT1	**Interrupt 1 Type Control Bit** Clear to select low level active (level triggered) for external interrupt 1 (INT1#). Set to select falling edge active (edge triggered) for external interrupt 1.
1	IE0	**Interrupt 0 Edge Flag** Cleared by hardware when interrupt is processed if edge-triggered (see IT0). Set by hardware when external interrupt is detected on INT0# pin.
0	IT0	**Interrupt 0 Type Control Bit** Clear to select low level active (level triggered) for external interrupt 0 (INT0#). Set to select falling edge active (edge triggered) for external interrupt 0.

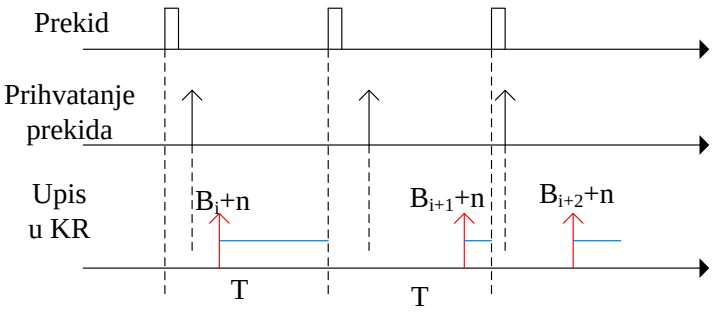
Reset Value = 0000 0000b

16

Treći način:
FREE RUNNING + OUTPUT COMPARE REGISTER



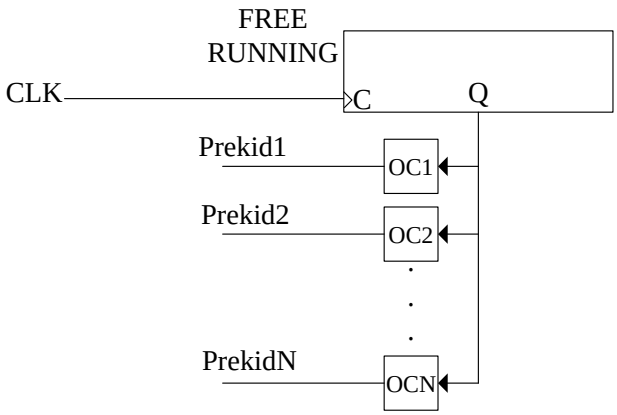
Treći način:
FREE RUNNING + OUTPUT COMPARE REGISTER



$B_{i+1}+n=B_i+2n$

$B_{i+2}+n=B_{i+1}+2n=B_i+3n$

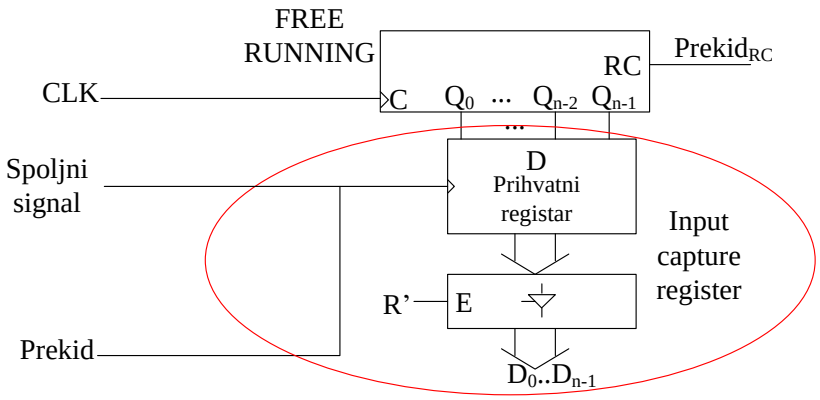
Jedan brojač – više intervala



Zahteva intervenciju programa

19

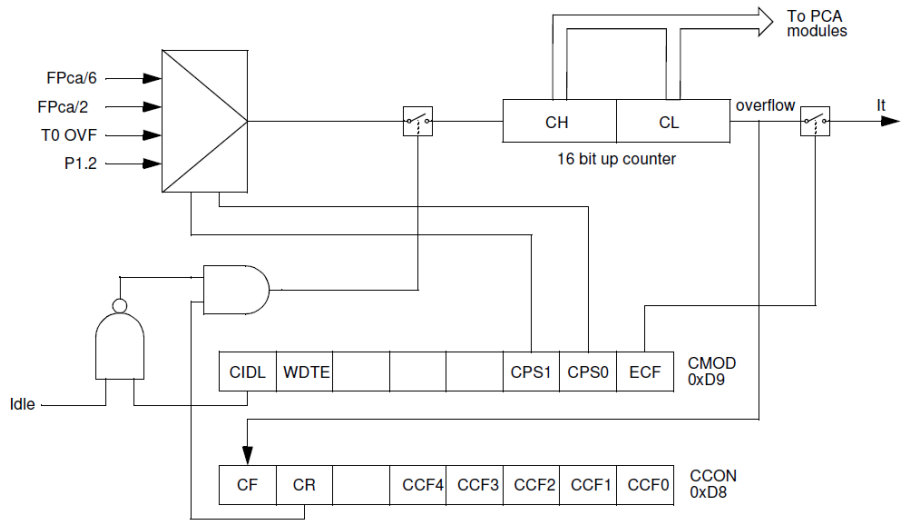
Merenje spoljnih intervala INPUT CAPTURE REGISTER



$$T=(B_{i+1}-B_i)T_{CLK}$$

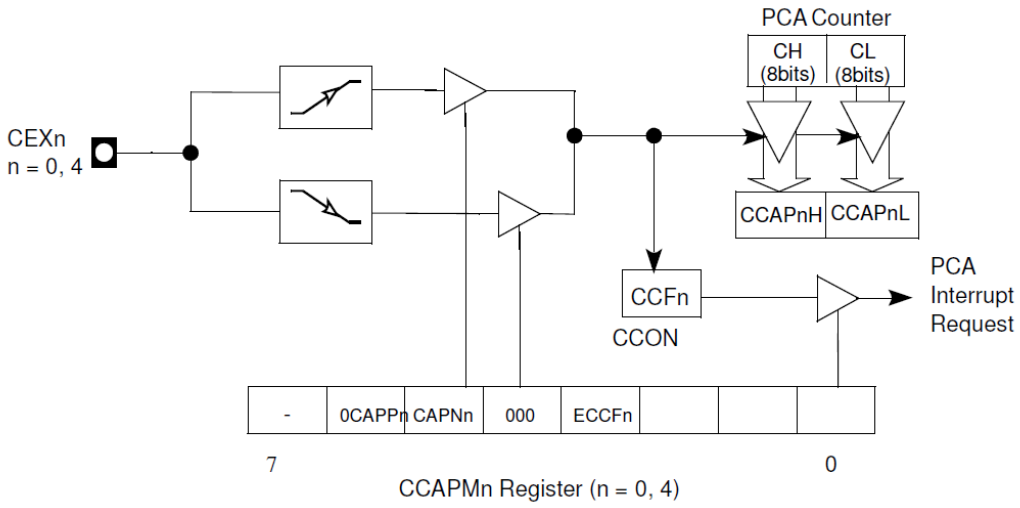
20

89c51 free running



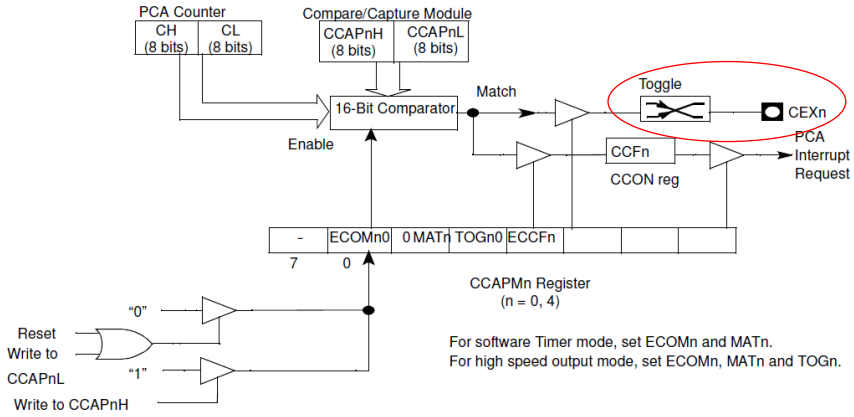
21

89c51 input capture



22

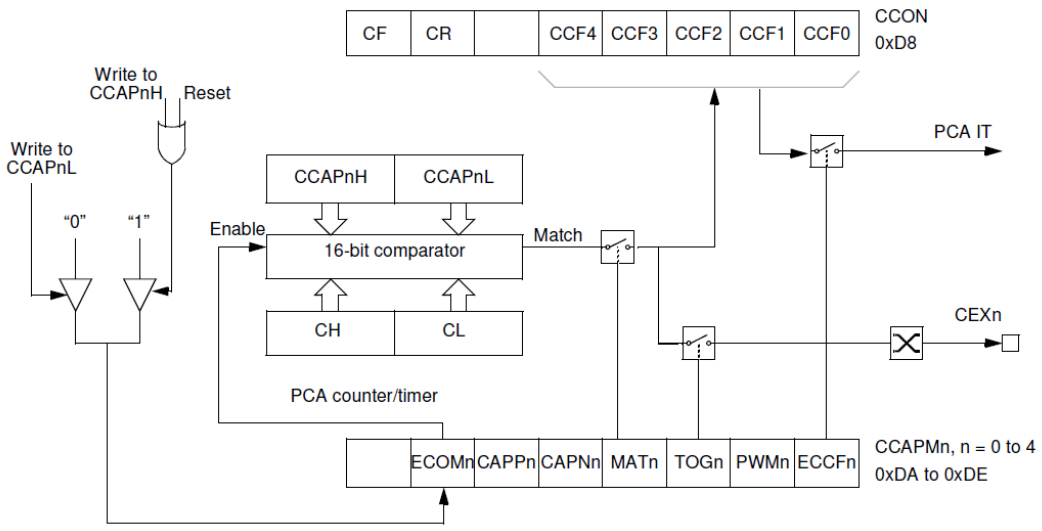
89c51 output compare



PROMENA STANJA PORTA BEZ INTERVENCIJE PROGRAMA!

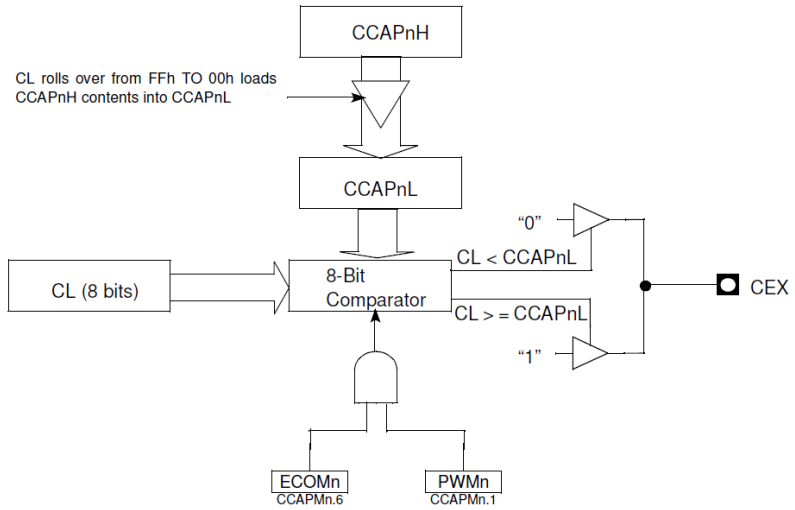
23

Detaljnije



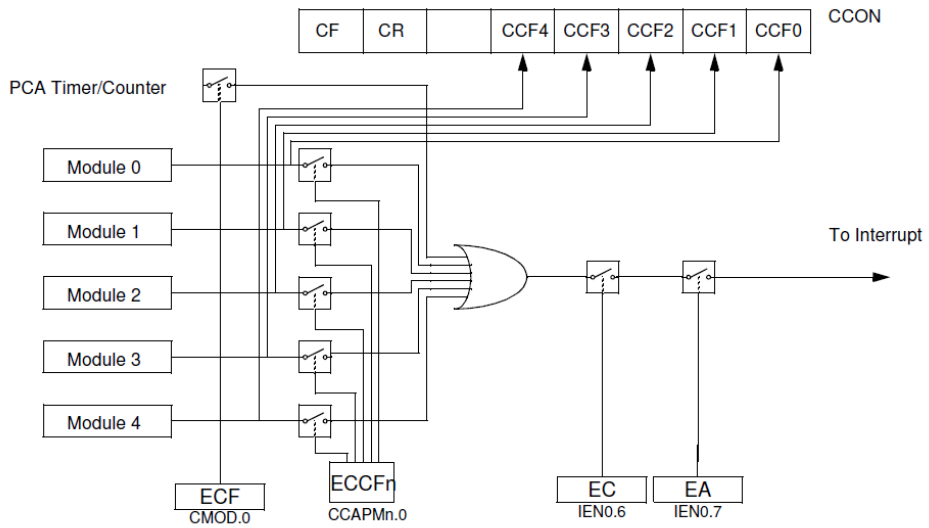
24

Različito trajanje impulsa i pauze izlaznog signala



25

Prekid + poliranje



26

Zaključak

SoC ili mikrokontroler
Više brojača
Nekoliko autoreload brojača
Free running sa više OC i IC registara

RTC – generički

Kompletan digitalni sistem koji vodi računa o sekundama, minutima, …
Letnjem zimskom računanju vremena
Prestupnim godinama

Specijalizovani brojači najčešće predviđeni za baterijsko napajanje.

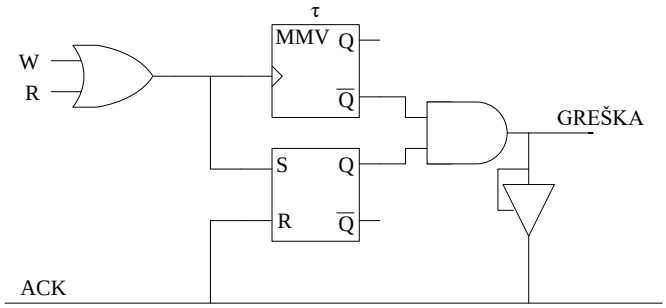
27

Timeout logika

28

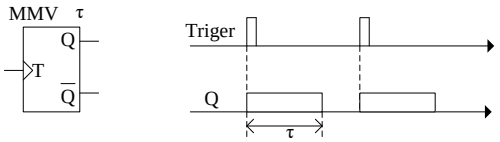
Timeout logika - „ISPRAVNOST“ rada periferija

Na prozivanje periferija mora da odgovori u zadatom intervalu – ne kasnije od …



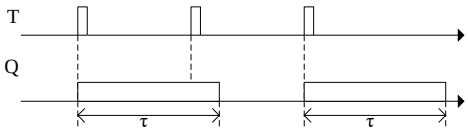
29

Merenje vremena - Monostabilni multivibrator

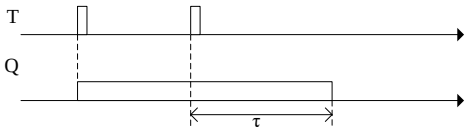


Stabilno stanje
Kvazistabilno stanje

neretrigerabilni

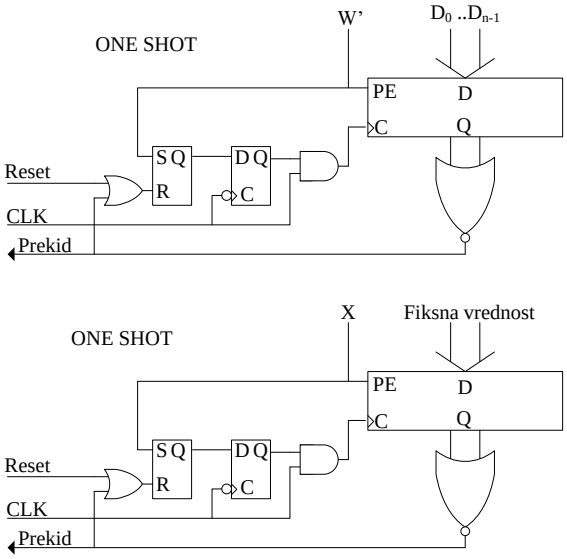


retrigerabilni



30

Monostabilni multivibrator u digitalnom obliku – sa brojačem - retrigerabilni



Napravite neretrigerabilni

31

Zaključak

Timeout logika - „ISPRAVNOST“ rada periferija

Na prozivanje u magistralnom ciklusu periferija mora da odgovori u zadatom intervalu – ne kasnije od …

Ali i u komunikacijama

Kada se traži nesto od periferije (R) mora da odgovori u zadatom intervalu

Kada se šalje nešto periferiji (W) mora da odgovori u zadatom intervalu???
Kako?

EHO protokol

ACK protokol

32

Watch dog

33

Watch dog tajmer

„ISPRAVNOST“ rada računara - procesora

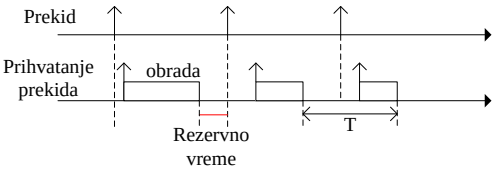
Preslikano iz drugi oblasti DEAD MAN SWITCH

dead-man's vigilance device - budnost

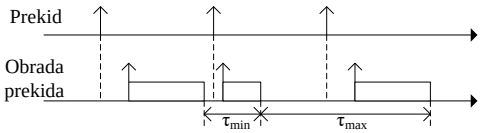
Driver Safety Device (DSD)

34

Uobičajen način rada računara



Šta može da se uoči, izračuna, izmeri



$T > \tau_{max}$
$T < \tau_{min}$ Računar sigurno ne radi kako treba

Pitanje

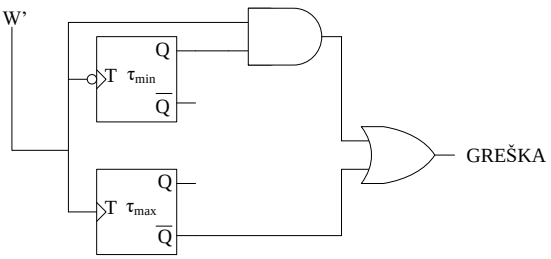
$\tau_{min} \leq T \leq \tau_{max}$

1. Računar sigurno radi kako treba
2. Računar možda radi kako treba

35

Realizacija ideje

Dummy write

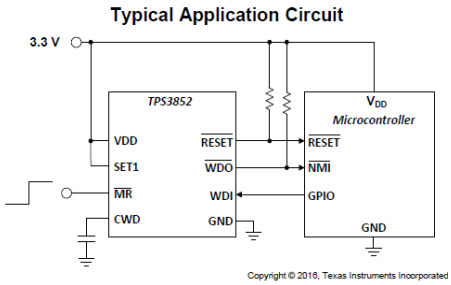


GREŠKA = RESET SISTEMA, HARD, SOFT …

GREŠKA = PREKID, NMI

36

Zajedno sa resetnim kolom
Supervizorsko kolo



CWD – priključak za kapacitivnost koja određuje vreme

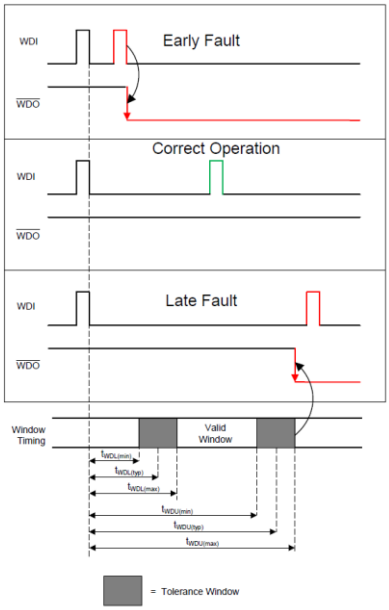
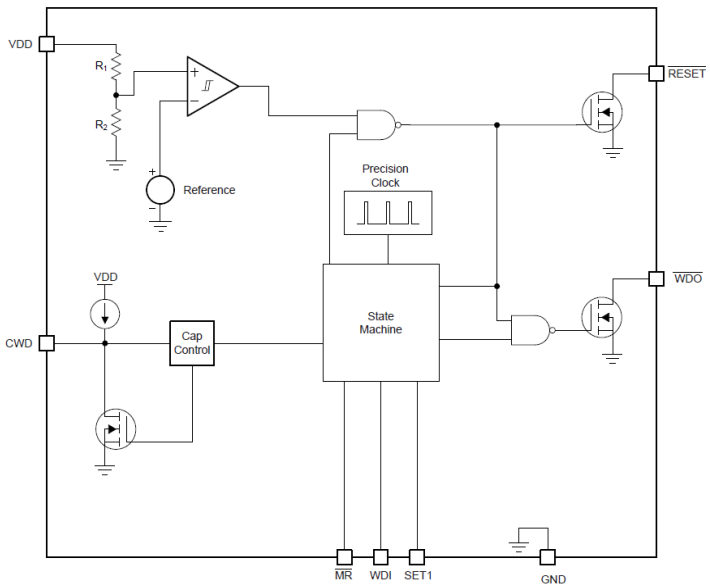


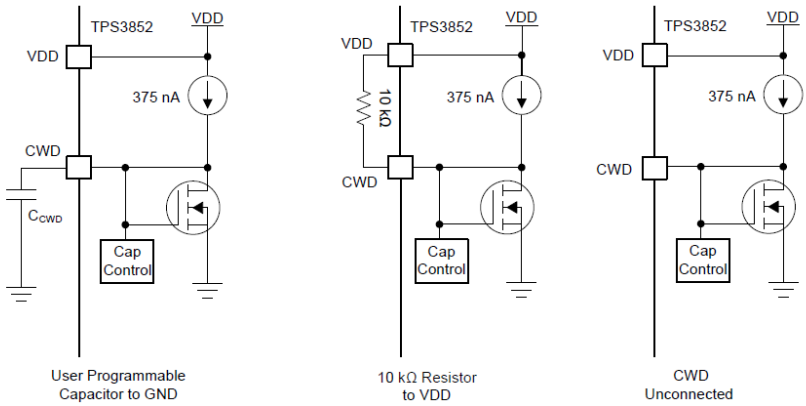
Figure 2. TPS3852 Window Watchdog Timing

37

Supervizorsko kolo



38



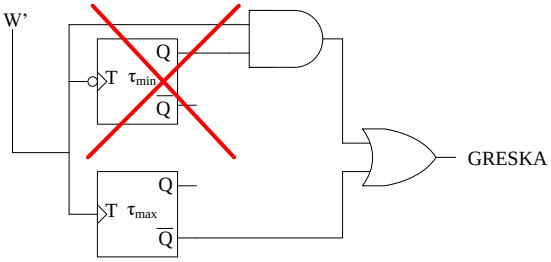
39

INPUT		WATCHDOG LOWER BOUNDARY (t_{WDL})			WATCHDOG UPPER BOUNDARY (t_{WDU})			UNIT
CWD	SET1	MIN	TYP	MAX	MIN	TYP	MAX	
NC	0	Watchdog disabled			Watchdog disabled			
	1	680	800	920	1360	1600	1840	ms
10 kΩ to VDD	0	Watchdog disabled			Watchdog disabled			
	1	1.5	1.85	2.2	8.8	11.0	13.2	ms

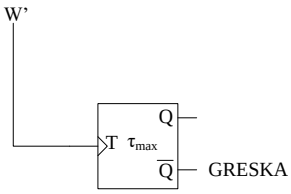
C_{CWD}	WATCHDOG UPPER BOUNDARY (t_{WDU})			UNIT
	MIN(1)	TYP	MAX(1)	
100 pF	53.32	62.74	72.15	ms
1 nF	112.5	132.4	152.2	ms
10 nF	704	829	953	ms
100 nF	6625	7795	8964	ms
1 µF	65836	77455	89073	ms

40

Najčešća realizacija



Meri se samo maksimalno vreme



41

Typical Application Schematic

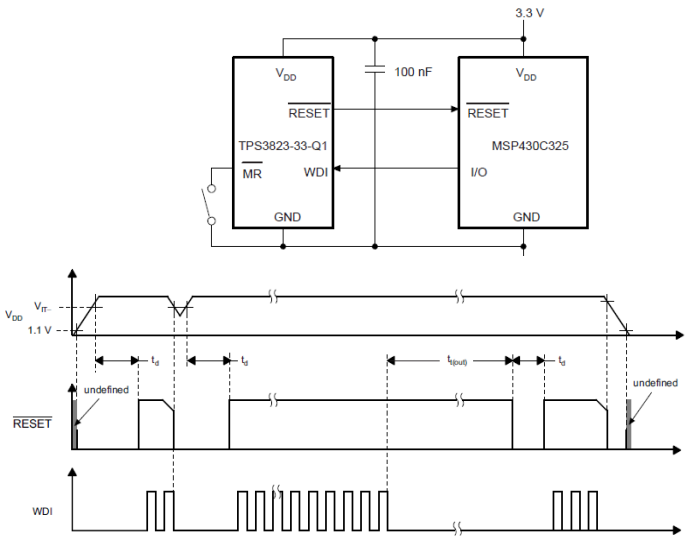
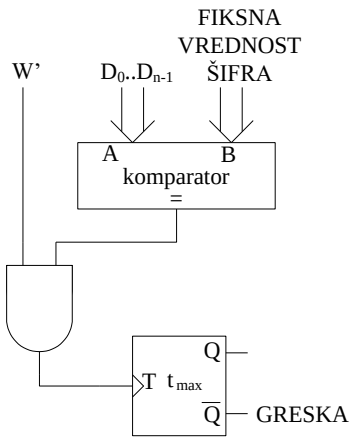


Figure 1. Delay and Time Out Timing Diagram

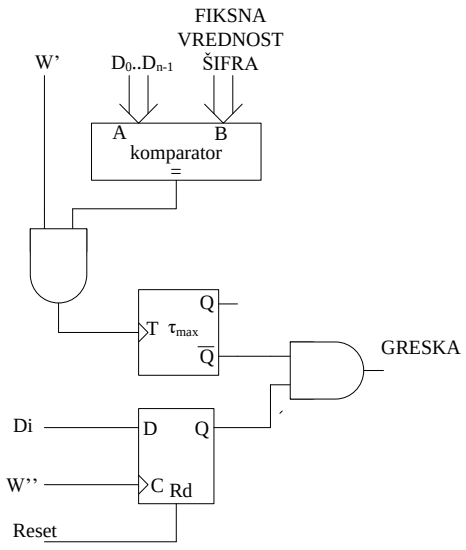
42

Povećanje pouzdanosti rada – pristup, reset, sa šifrom



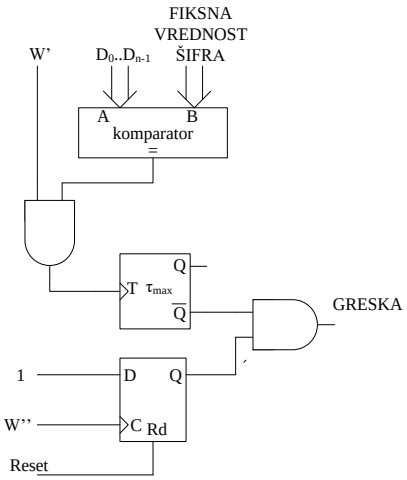
43

Može da se pokrene i zaustavi



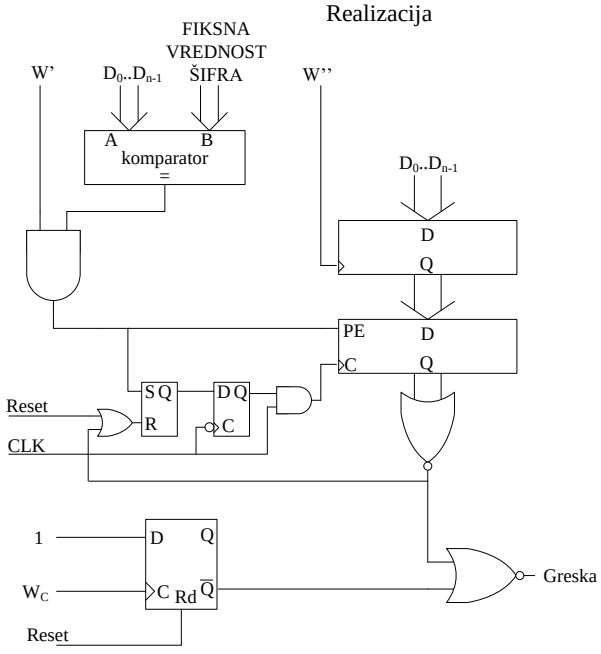
44

Kada se pokrene nema zaustavljanja



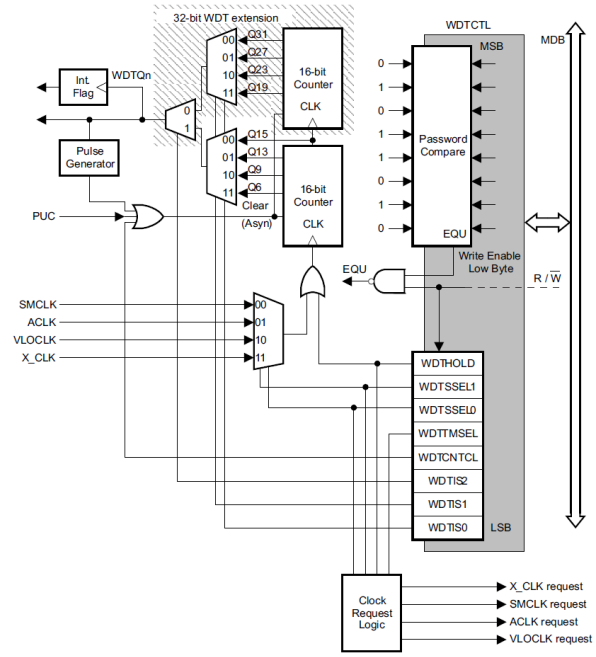
45

Realizacija



46

MSP430



47

The primary function of the watchdog timer (WDT_A) module is to perform a controlled system restart after a software problem occurs. If the selected time interval expires, a system reset is generated. If the watchdog function is not needed in an application, the module can be configured as an interval timer and can generate interrupts at selected time intervals.

Features of the watchdog timer module include:

- Eight software-selectable time intervals
- Watchdog mode
- Interval mode
- Password-protected access to Watchdog Timer Control (WDTCTL) register
- Selectable clock source
- Can be stopped to conserve power
- Clock fail-safe feature

NOTE: Watchdog timer powers up active.

After a PUC, the WDT_A module is automatically configured in the watchdog mode with an initial ~32-ms reset interval using the SMCLK. The user must setup or halt the WDT_A prior to the expiration of the initial reset interval.

48

16.2 WDT_A Operation

The watchdog timer module can be configured as either a watchdog or interval timer with the WDTCTL register. WDTCTL is a 16-bit password-protected read/write register. Any read or write access must use word instructions and write accesses must include the write password 05Ah in the upper byte. Any write to WDTCTL with any value other than 05Ah in the upper byte is a password violation and triggers a PUC system reset, regardless of timer mode. Any read of WDTCTL reads 069h in the upper byte. Byte reads on WDTCTL high or low part result in the value of the low byte. Writing byte wide to upper or lower parts of WDTCTL results in a PUC.

16.2.5 Clock Fail-Safe Feature

The WDT_A provides a fail-safe clocking feature, ensuring the clock to the WDT_A cannot be disabled while in watchdog mode. This means that the low-power modes may be affected by the choice for the WDT_A clock.

If SMCLK or ACLK fails as the WDT_A clock source, VLOCLK is automatically selected as the WDT_A clock source.

49

Zaključak

WDT – ispravnost rada računara

Naredba za reset WDT na što manje mesta u programu.
Da bi se smanjila verovatnoća da u slučaju pogrešnog rada program “slučajno natrči“ na nju.

Naredba za reset WDT po pravilu samo na jednom mestu u programu i to u glavnom.

Izbegavati naredbu za reset WDT u prekidnim rutinama.

Smanjivanje verovatnoće da u slučaju pogrešnog rada to WDT ne detektuje.

Ako WDT nije resetovao sistem to ne znači da sistem radi ispravno

Fault tolerant sistemi – Sistemi otporni na otkaze

50