

1. Sistemski takt procesora CCLK je učestanosti 10 MHz (perioda 0,1 μ s). Delitelj za dobijanje takta za periferije (PCLK) je podešen na 1.

Potrebno je na tri izlazne nožice generisati tri signala (**A**, **B** i **C**) učestanosti 1 kHz (periode 1 ms) koji imaju zajedničku rastuću ivicu, a trajanje impulsa (logička jedinica) signala **A** iznosi 100 μ s, trajanje impulsa signala **B** je 256 μ s, a impulsa signala **C**, 750 μ s. Ostatak do 1 ms je pauza (logička nula).

Definisati nožice procesora na kojima će se pojaviti signali i opisati šta je sve potrebno upisati i u koje registre da bi se to obezbedilo.....(4)

Koje registre je potrebno inicijalizovati i na koje vrednosti, da bi se dobili opisani talasni oblici na odabranim nožicama?(6)

Ako korisnik želi da promeni trajanje signala **B** i **C** na 100 μ s, a da ne poremeti ni jednu celu periodu, u koje registre i koje podatke treba da upiše?(5)

2. Napisati deo c-kôda kojim se obezbeđuje da se na izlazu DA konvertora dobije signal periode 2 ms pri čemu je 1 ms napon na izlazu 1,50 V, a 1 ms 3,00V. Referentni napon je 3,00V. Pretpostaviti da postoji funkcija `void pauza_1ms(void)`. Nije neophodno da sintaksa bude tačna.... (15)

3. Koji podatak treba upisati (i u koji registar) da bi se odmah startovala jedna desetobitna AD konverzija sa kanala AD1.0. Učestanost perifernog takta PCLK je 4MHz. Obrazložiti rešenje.....(6)

Kako bi izgledao kôd kojim bi se u petlji sačekao kraj pokrenute konverzije. Koji registar treba očitati i kako prepoznati kraj konverzije?(4)

4. Koje sve registre treba inicijalizovati, i na koje vrednosti, da bi se ostvarila šema prekida u kojoj postoje dva prekida tipa IRQ od sledećih periferija: TIMER1 višeg i TIMER0, nižeg prioriteta i jedan prekid tipa FIQ od ADC0 periferije.....(6)

Opisati razliku i namenu vektorizovanih IRQ, nevektorizovanih IRQ i prekida tipa FIQ(4)

VAŽNO:

Odgovori treba da sadrže tačne oznake registara i konkretne vrednosti koje treba upisati u njih, bilo kao decimalni, heksadecimalni ili binarni broj! Opisni odgovori nisu prihvatljivi, osim za teorijska pitanja.

Ispit traje 90 minuta.

Dozvoljena je upotreba literature.

Nije dozvoljeno pozajmljivanje literature.

1. Sistemski takt procesora CCLK je učestanosti 20 MHz (perioda 0,05 μ s). Delitelj za dobijanje takta za periferije (PCLK) je podešen na 2.

Potrebno je na tri izlazne nožice generisati tri signala (**A**, **B** i **C**) učestanosti 1 kHz (periode 1 ms) koji imaju zajedničku rastuću ivicu, a trajanje impulsa (logička jedinica) signala **A** iznosi 100 μ s, trajanje impulsa signala **B** je 366 μ s, a impulsa signala **C**, 450 μ s. Ostatak do 1 ms je pauza (logička nula).

Definisati nožice procesora na kojima će se pojaviti signali i opisati šta je sve potrebno upisati, i u koje registre, da bi se to obezbedilo.....(4)

Koje registre je potrebno inicijalizovati, i na koje vrednosti, da bi se dobili opisani talasni oblici na odabranim nožicama?(6)

Ako korisnik želi da promeni trajanje signala **B** i **C** na 100 μ s, a da ne poremeti ni jednu celu periodu, u koje registre, i koje podatke, treba da upiše?(5)

2. Napisati deo c-kôda (samo petlju) kojim se obezbeđuje da se na izlazu DA konvertora dobije signal periode 2 ms pri čemu je 1 ms napon na izlazu 0 V, a 1 ms 1,50 V. Referentni napon je 3,00 V. Pretpostaviti da postoji funkcija `void pauza_1ms(void)`. Nije neophodno da sintaksa bude tačna.(15)

3. Koji podatak treba upisati (i u koji registar) da bi se odmah startovala jedna desetobitna AD konverzija sa kanala AD0.4. Učestanost perifernog takta PCLK je 4MHz. Obrazložiti rešenje.....(6)

Kako bi izgledao kôd kojim bi se u petlji sačekao kraj pokrenute konverzije. (koji registar treba očitati i kako prepoznati kraj konverzije)?(4)

4. Koje sve registre treba inicijalizovati, i na koje vrednosti, da bi se ostvarila šema prekida u kojoj postoje dva prekida tipa IRQ od sledećih periferija: TIMER0 višeg i TIMER1, nižeg prioriteta i jedan prekid tipa FIQ od ADC1 periferije.....(6)

Opisati razliku i namenu vektorizovanih IRQ, nevektorizovanih IRQ i prekida tipa FIQ(4)

VAŽNO:

Odgovori treba da sadrže tačne oznake registara i konkretne vrednosti koje treba upisati u njih, bilo kao decimalni, heksadecimalni ili binarni broj! Opisni odgovori nisu prihvatljivi, osim za teorijska pitanja.

Ispit traje 90 minuta.

Dozvoljena je upotreba literature.

Nije dozvoljeno pozajmljivanje literature.

REŠENJE - studija (1. grupa)

1. zadatak:

Analiza:

1. Iz trajanja impulsa (100 μ s, 256 μ s i 750 μ s) može se zaključiti da je dovoljna rezolucija od 1 μ s (svakako, može i bilo koja bolja rezolucija $<1 \mu$ s). Ako usvojimo ovoliku rezoluciju (1 μ s) potreban nam je predelitelj koji bi takt periode 0,1 μ s podelio sa 10. Tada je perioda signala 1000, a širine impulsa 100, 256 i 750. Da smo usvojili drugu rezoluciju, ove vrednosti bi bile drugačije. Na primer, za maksimalnu rezoluciju od 0,1 μ s (predelitelj deli ulazni takt sa 1), perioda bi bila 10000, a širine impulsa 1000, 2560 i 7500.

2. Potrebna su nam tri PWM izlaza sa kontrolom jedne ivice, usvojimo da su to PWM1, PWM2 i PWM3. PWM1 deli pin sa P0.0; PWM2 deli pin sa P0.7; PWM3 deli pin sa P0.1 (strana 74 priručnika). Potrebno je odabrati da se ova tri pina definišu kao izlazi PWM periferije.

3. Periodu ćemo definisati u registru PWMMR0, a trajanje impulsa na izlazima A, B i C u registrima PWMMR1, PWMMR2 i PWMMR3 (pog. 15.4.6, strana 189 priručnika).

4. Promena dužine impulsa bez remećenja signala vrši se korišćenjem PWMLER registra (pog. 15.4.9, strana 191 priručnika).

Rešenje:

Definicija pinova (tačka 2):

PINSEL0.1 = 1 PINSEL0.0 = 0 da bi se odabrao PWM1 umesto P0.0

PINSEL0.3 = 1 PINSEL0.2 = 0 da bi se odabrao PWM3 umesto P0.1

PINSEL0.15 = 1 PINSEL0.14 = 0 da bi se odabrao PWM2 umesto P0.7

```
//          3          2          1          0
//bit broj 10987654321098765432109876543210
PINSEL0 = xxxxxxxxxxxxxxxxxxx10xxxxxxxxxxx10101 (binarno)
```

Pre upisa vrednosti treba zabraniti brojanje PWM brojaču i resetovati ga. Da bi vrednosti upisane u registre PWMMR0 do PWMMR3 odmah postale „važeće”, bit 3, koji uključuje registre u senci (PWM režim), mora biti 0 na početku. U suprotnom, vrednosti postaju „važeće” tek posle poklapanja sa PWMMR0. Ovo je naročito važno zbog upisa u „radni” PWMMR0 koji je u PWM-režimu (bit 3 ovog registra na jedinici) uslovljen trenutkom početka periode, a perioda ne može da se odredi dok se ne upiše trajanje periode u PWMMR0.

Kada PWM brojač radi u PWM-režimu (PWMTCR.3=1), svi registri poklapanja su duplirani. Pored „radnog”, to jest onog sa kojim se poredi vrednost brojača, postoji i registar u senci. Upis podatka u bilo koji registar poklapanja je ustvari upis u njegov registar u senci, a hardver sâm prebaci² te podatke u radni registar u trenutku početka nove periode (kada dođe do poklapanja vrednosti PWMTTC sa PWMMR0) i tek

¹ Podebljana nula ili jedinica je vrednost koju je nužno upisati da bi se dobio željeni efekat. Nula koja nije podebljana je vrednost koju zahteva kontroler, x može biti nula ili jedinica, zavisno od drugih potreba (ne utiče na ono što je tema zadatka).

² Prebacivanje podataka u radne registre se može kontrolisati i PWMLER registrom.

tada novoupisani podaci postaju važeći. Ovakav princip je bio nužan da bi se svi upisani podaci ažurirali istovremeno i u trenutku kada ne mogu da poremete periodu koja je već otpočela.

Kada PWM brojač radi u režimu običnog brojača (PWMTCCR.3=1), svaki upis u registre poklapanja ide direktno u „radni” registar (registri u senci ne postoje), i odmah postaje „važeći” (pog. 15.4.2, strana 188 priručnika):

```
//bit broj      76543210
    PWMTCCR = 00000010    (binarno)
```

Definicija predelitelja (tačka 1):

```
    PWMPR = 9;    /* predelitelj deli sa 10 (10=9+1) */
```

Kontrolni registar poklapanja (PWMMCR) treba inicijalizovati tako da poklapanje 0 resetuje PWMTC, ne postavlja zahtev za prekid i ne zaustavlja brojač. Poklapanja 1, 2 i 3 ne treba ni da resetuju PWMTC niti da zahtevaju prekid niti da zaustavljaju brojač. Ostala poklapanja se ne koriste. U registru PWMMCR biti 0, 1 i 2 se odnose na to šta će se desiti u slučaju poklapanja vrednosti brojača PWMTC sa registrom poklapanja PWMMR0. Sledeća grupa od tri bita odnosi se na poklapanje sa PWMMR1 i tako dalje. U grupi od tri bita poslednji (najniže pozicione vrednosti) određuje da li će se u slučaju poklapanja postaviti zahtev za prekid, sledeći bit određuje da li će se resetovati brojač i bit najviše pozicione vrednosti da li će se u slučaju poklapanja zaustaviti brojač. (pog. 15.4.7, strana 189 priručnika):

```
//          3          2          1          0
//bit broj  10987654321098765432109876543210
    PWMMCR = 000000000000xxxxxxx000000000010
//
//          Pok. sa MR3  Pok. sa MR2  Pok. sa MR1  |||_ bez prekida
//          |           |           |           ||__ reset PWMTC
//          |           |           |           |___ bez zaustav. PWMTC
```

Pomoću kontrolnog registar širinskog modulatora (PWMPCCR) se određuje da PWM1, PWM2 i PWM3 rade u režimu kontrole jedne ivice (tačka 1) i aktiviraju se ova tri modulatora (pog. 15.4.8, strana 190 priručnika):

```
//          3          2          1          0
//bit broj  10987654321098765432109876543210
    PWMPCCR = 000000000000000000xxx11100xxx0000
//
//          \||/          ||_ PWM2 rezim jedne ivice
//          |           |___ PWM3 rezim jedne ivice
//          |_Uklj. sir. mod. PWM1, PWM2 i PWM3
```

Zadavanje vrednosti registrima poklapanja (tačka 1):

Upisane vrednosti odmah postaju aktivne pošto je PWM režim isključen (još uvek je PWMTCCR.3=0) tako da radni registar PWMMR0 koji definiše periodu ima upisanu vrednost 1000 i kada brojač PWMTC odbroji do 1000, doći će do poklapanja sa PWMMR0 i uslova za počinjanje nove periode.

```

PWMMR0 = 1000; /* Perioda signala A, B, i C */
PWMMR1 = 100; /* Širina impulsa signala A */
PWMMR2 = 256; /* Širina impulsa signala B */
PWMMR3 = 750; /* Širina impulsa signala C */

```

Registri poklapanja su spremni, možemo „pustiti“ brojač da broji i prebaciti ga da radi u PWM režimu (da upis u registre poklapanja ide preko registara u senci) ([pog. 15.4.2, strana 188 priručnika](#)):

```

//bit broj      76543210
PWMTCCR = 00001001    (binarno)

```

Upis novih vrednosti trajanja impulsa signala B i C (tačka 4):

```

PWMMR2 = 100; /* Nova širina impulsa signala B */
PWMMR3 = 100; /* Nova širina impulsa signala C */

```

Nove vrednosti u registrima poklapanja nemaju nikakav efekat jer su upisane samo u registre u senci (sada brojač radi u PWM režimu) i na izlazu će se i dalje generisati impulsi širine 256 μ s, odnosno 750 μ s sve dok se ne ispune dva uslova: Prvo, mi moramo da kažemo da želimo da promene postanu aktivne (upisom u PWMLER registar), i drugo, kada to uradimo, mora da se završi tekuća periodia i tek po njenom završetku, u sledećoj periodi, trajanje impulsa će biti 100 μ s ([pog. 15.4.9, strana 191 priručnika](#))

```

//bit broj      76543210
PWMLER = xxxx1100 /* Po završetku tekuće periode, počni novu
                    periodu sa novim vrednostima PWMMR2 i PWMMR3 */

```

Jedno od mogućih kompletnih rešenja zadatka bi bilo:

```

//bit broj      10987654321098765432109876543210
PINSEL0= 0x0000800A;
PWMTCCR = 0x02;      /* režim obicnog brojaca, reset, stop */
PWMPR   = 9;         /* predelitelj deli sa 10 (10=9+1)*/
PWMMCR  = 0x00000002 /* reset TC pri poklapanju MR0 */
PWMPCCR = 0x00000E00;
PWMMR0  = 1000;      /* Perioda signala A, B, i C */
PWMMR1  = 100;       /* Širina impulsa signala A */
PWMMR2  = 256;       /* Širina impulsa signala B */
PWMMR3  = 750;       /* Širina impulsa signala C */
PWMTCCR = 0x0A       /* PWM-režim , počni brojanje */

PWMMR2  = 100; /* Nova širina impulsa signala B */
PWMMR3  = 100; /* Nova širina impulsa signala C */
PWMLER  = 0x0C; /* Po završetku tekuće periode, počni novu
                  periodu sa novim vrednostima PWMMR2 i PWMMR3 */

```

2. zadatak:

Vrednost koju treba upisati u polje VALUE registra DACR ([pog. 17.3, strana 198 priručnika](#)) da bi se na izlazu dobio napon 1,50 V, odnosno polovina referentnog napona, je 10 0000 0000 binarno (bit najveće težine nosi 50% vrednosti referentnog napona odnosno $1/2^1$, sledeći bit nosi 25%, odnosno $1/2^2$, naredni $1/2^3$ i tako dalje).

Vrednost koju treba upisati u polje VALUE registra DACR da bi se dobio pun napon na izlazu (najbliži referentnom naponu) je 11 1111 1111 binarno. Preciznije, napon koji se u tom slučaju dobije je umanjen za oko 3mV ($3V/1024$) u odnosu na referentni, odnosno oko 2,997V, ali to je najveća vrednost koja se može dobiti sa referentnim naponom 3,00V.

Vrednost koju treba upisati u BIAS polje je proizvoljna . Ostali biti DACR

```
//          3          2          1          0
//bit broj  10987654321098765432109876543210
DACR =      0000000000000000x1000000000000000 /* Vout=1,500V */
DACR =      0000000000000000x1111111111000000 /* Vout=2,997V */
```

Petlja u c-kodu, tražena zadatkom bi izgledala:

```
While (1)
{
    DACR = 0x00008000;
    pauza_1ms ();
    DACR = 0x0000ffc0;
    pauza_1ms ();
}
```

3. zadatak:

Da bismo pokrenuli AD konverziju sa kanala AD1.0, treba u polje START registra AD1CR ([pog. 16.4.1, strana 194 priručnika](#)) upisati 001.

U polje SEL registra AD1CR treba upisati 00000001 binarno. U ovom polju, svaki bit pojedinačno određuje koji od osam ulaznih kanala se koristi. Osim u „BURST” režimu, u ovom polju ne bi trebalo da se nađe više od jedne jedinice.

U polje CLKDIV registra AD1CR treba upisati 00000000 binarno pošto je učestanost takta sa kojom rade periferije 4 MHz, a AD konvertor može da radi sa taktom do 4,5 MHz, to ova periferija može koristiti isti takt kao ostale periferije. Inače, AD konvertor radi sa taktom čija se učestanost dobija deljenjem periferijskog takta PCLK vrednošću upisanom u CLKDIV polje, uvećanom za 1. Kada bi u polju upisali, na primer, 00000011 što iznosi 3 kao decimalni broj, učestanost periferijskog takta bi se delila sa 4, ($4=3+1$) i AD konvertor bi radio sa taktom učestanosti 1 MHz (a može da radi do 4,5 MHz).

U polje BURST(bit 16), registra AD1CR, treba upisati nulu jer se ovaj režim ne koristi za konverziju sa jednog kanala.

U polje PDN (bit 21), registra AD1CR treba upisati jedinicu jer se time uključuje napajanje AD periferiji. U suprotnom, AD konvertor je isključen (bez napajanja).

Ostala polja nisu značajna za ovaj zadatak.

```
//          3          2          1          0
//bit broj 10987654321098765432109876543210
DACR = 0000x0010010xxx0000000000000001
```