

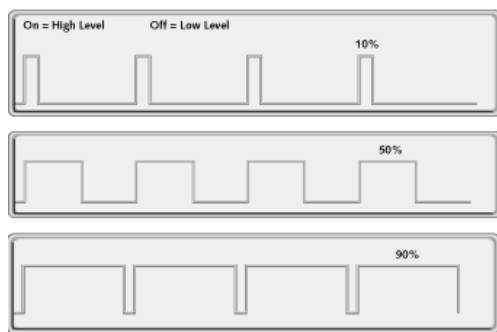
# FRISC-MC

Zavod za automatiku i procesno računarstvo  
grupa RASIP

## FRISC-MC

- Jedinica za pulsno-širinsku modulaciju (PWM)
- Primjeri

## Pulsno-širinska modulacija



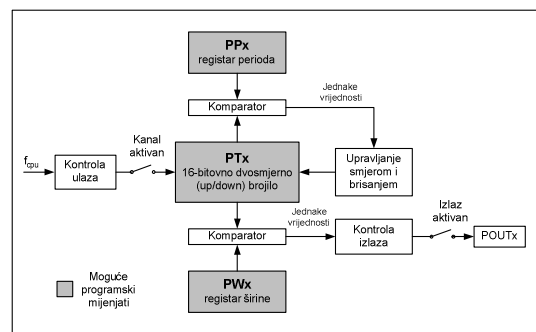
## Jedinica za pulsno-širinsku modulaciju

- Spojena na priključke mikrokontrolera (Port 0)
- Omogućava generiranje pravokutnog signala željenog faktora popunjenosti
- Sastoji se od 4 kanala koji mogu raditi nezavisno ili se kombinirati
  
- Zauzima 16 lokacija u memoriji kao vanjska jedinica
- Može izazvati prekid (PWMIR)

## PWM jedinica

- Svaki kanal sadrži:
  - 16-bitni dvosmjerni timer (PTx)
  - 16-bitni registar širine impulsa (PWx)
  - 16-bitni registar perioda signala (PPx)
  - Dva komparatora
  - Dodatnu upravljačku logiku
- Frekvencija generiranog signala ovisi o:
  - Taktu na kojem procesor radi  $f_{CPU}$
  - Odabranoj razlučivosti brojača ( $f_{CPU}/1$  ili  $f_{CPU}/64$ )
  - Odabranom načinu rada – poravnanje po rubu ili po sredini
  - Vremenskoj konstanti brojača

## PWM jedinica – shema sklopa



## Načini rada

- Moguće je generirati:
  - Standardni PWM signal (asimetrični, *edge-aligned*)
  - Simetrični signal (*center-aligned*)
- Kombinirani (*burst*) signal – kombinaciju dvaju kanala

## Registri PWM jedinice

- PWMCON0 – kontrola rada jedinice, ulaza, prekida
- PWMCON1 – kontrola izlaza, načina rada
- PWMIC – kontrola prekida (jedan za sve kanale)
- PTx – brojlara za svaki kanal (PT0..PT3)
- PPx – registar perioda za svaki kanal (PP0..PP3)
- PWx – registar širine impulsa (PW0..PW3)

## Registri PWM jedinice – PWMCON0

- PWMCON0 – kontrola rada, ulaza, prekida

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIR3	PIR2	PIR1	PIR0	PIE3	PIE2	PIE1	PIE0	PTI3	PTI2	PTI1	PTI0	PTR3	PTR2	PTR1	PTR0

- PTRx – pokretanje pojedinih kanala
- PTIx – odabir ulaznih impulsa
- PIEx – kontrola prekida
- PIRx – zastavica koja označava da je prekid aktivan

## Registri PWM jedinice – PWMCON0

- PWMCON0 – kontrola rada, ulaza, prekida

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIR3	PIR2	PIR1	PIR0	PIE3	PIE2	PIE1	PIE0	PTI3	PTI2	PTI1	PTI0	PTR3	PTR2	PTR1	PTR0

- PTRx – PWM timer X Run control bit
  - 0 – timer x nije spojen na ulazni signal (neaktivan je)
  - 1 – timer x je aktivan

## Registri PWM jedinice – PWMCON0

- PWMCON0 – kontrola rada, ulaza, prekida

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIR3	PIR2	PIR1	PIR0	PIE3	PIE2	PIE1	PIE0	PTI3	PTI2	PTI1	PTI0	PTR3	PTR2	PTR1	PTR0

- PTIx – PWM Timer X Input clock selection
  - 0 – ulazni impulsi su  $f_{cpu}$
  - 1 – ulazni impulsi su  $f_{cpu}/64$  (predjelilo)

## Registri PWM jedinice – PWMCON0

- PWMCON0 – kontrola rada, ulaza, prekida

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIR3	PIR2	PIR1	PIR0	PIE3	PIE2	PIE1	PIE0	PTI3	PTI2	PTI1	PTI0	PTR3	PTR2	PTR1	PTR0

- PIEx – PWM channel X Interrupt Enable zastavica
  - 0 – kanal x ne izaziva prekide
  - 1 – kanal x izaziva prekide

## Registri PWM jedinice – PWMCON0

- PWMCON0 – kontrola rada, ulaza, prekida

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIR3	PIR2	PIR1	PIR0	PIE3	PIE2	PIE1	PIE0	PTI3	PTI2	PTI1	PTI0	PTR3	PTR2	PTR1	PTR0

- PIRx – PWM channel X Interrupt Request zastavica
  - Postavlja se kad je aktivan prekid za neki kanal (potrebno je programski ispitati koji kanal je izazvao prekid jer svi PWM kanali dijele isti zahtjev za prekid; nakon obrade prekida potrebno je programski obrisati zastavicu)
  - 0 – kanal x nije izazvao prekid
  - 1 – kanal x je izazvao prekid

## Registri PWM jedinice – PWMCON1

- PWMCON1 – kontrola izlaza, načina rada

9	8	7	6	5	4	3	2	1	0
PB23	PB01	PM3	PM2	PM1	PM0	PEN3	PEN2	PEN1	PEN0

- PENx – kontrola izlaza
- PMx – kontrola načina rada
- PBxy – aktiviranje *burst* načina rada

## Registri PWM jedinice – PWMCON1

- PWMCON1 – kontrola izlaza, načina rada

9	8	7	6	5	4	3	2	1	0
PB23	PB01	PM3	PM2	PM1	PM0	PEN3	PEN2	PEN1	PEN0

- PENx – PWM channel X output ENable bit
  - 0 – izlaz kanala x je isključen, moguće generiranje prekida
  - 1 – izlaz kanala x je uključen

## Registri PWM jedinice – PWMCON1

- PWMCON1 – kontrola izlaza, načina rada

9	8	7	6	5	4	3	2	1	0
PB23	PB01	PM3	PM2	PM1	PM0	PEN3	PEN2	PEN1	PEN0

- PMx – PWM channel X Mode control bit
  - 0 – kanal x radi u asimetričnom načinu rada (*edge-aligned*)
  - 1 – kanal x radi u simetričnom načinu rada (*center-aligned*)

## Registri PWM jedinice – PWMCON1

- PWMCON1 – kontrola izlaza, načina rada

9	8	7	6	5	4	3	2	1	0
PB23	PB01	PM3	PM2	PM1	PM0	PEN3	PEN2	PEN1	PEN0

- PB01 – PWM channel 0/1 Burst mode control bit
  - 0 – kanali 0 i 1 rade neovisno u svojim načinima rada
  - 1 – izlaz kanala 0 i 1 kombiniran je logičkim AND na POUT0
- PB23 – PWM channel 2/3 Burst mode control bit
  - 0 – kanali 2 i 3 rade neovisno u svojim načinima rada
  - 1 – izlaz kanala 2 i 3 kombiniran je logičkim AND na POUT2

## Registri PWM jedinice – PWMIC

- PWMIC – kontrola prekida

7	6	5	4	3	2	1	0
PWMIR	PWMIE	-	-	-	ILVL		

- ILVL – određuje razinu prekida (0 – 7), razina 0 ne izaziva prekid
- PWMIE – *PWM interrupt enable* – uključuje postavljanje zahtjeva za prekid (potrebno je ručno ispitati koji kanal je izazvao prekid)
- PWMIR – *PWM interrupt request* – označava postavljen zahtjev za prekid, automatski se briše nakon prihvatanja prekida

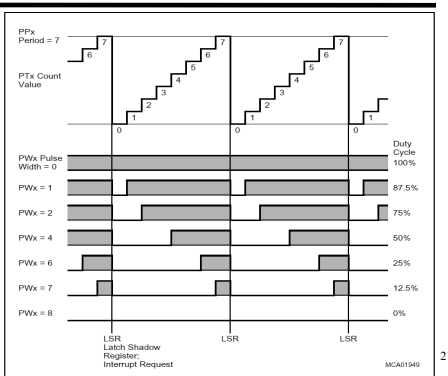
## Registri PWM jedinice – PTx, PPx, PWx

- Koriste se kao 'obični' 16-bitni registri, nemaju posebnih polja
- PTx – registar dvosmjernog brojila vremenskog sklopa
  - Pisanje – postavlja vrijednost brojila kanala x
  - Čitanje – vraća trenutnu vrijednost brojila kanala x
- PPx – registar perioda kanala x
- PWx – registar širine impulsa kanala x

## PWM - načini rada

- **Način rada 0 – standardni PWM signal**
  - U ovom načinu rada vremenski sklop PTx odgovarajućeg PWM kanala uvijek broji prema gore dok ne dosegne vrijednost zapisanu u PPx registru
  - Nakon dosezanja vrijednosti vremenski sklop počinje ponovo brojati od nule
  - PWM signal je na visokoj razini kada je sadržaj brojila vremenskog sklopa PTx jednak ili veći od vrijednosti pomoćnog registra koji određuje širinu impulsa – PWx, a vraća se u nisku razinu nakon resetiranja brojila na nulu
  - Period generiranog signala je:  $PWM\_Period_{mode0} = PPx + 1$
  - Popunjenost signala određena je vrijednošću PWx registra i kreće se od 0% za  $PWx = PWx > PPx$  do 100% za  $PWx = 0$
  - Zamka: registar PWx u stvari odražava *ispražnjenost* a ne popunjenost signala!

## PWM – način rada 0

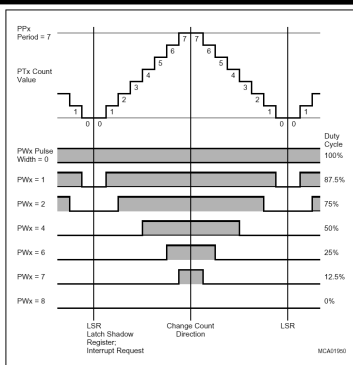


## PWM - načini rada

- **Način rada 1 – simetrični PWM signal**
  - Vremenski sklop PTx odgovarajućeg kanala broji prema gore dok ne dosegne vrijednost PPx registra
  - Sljedeći impuls mijenja smjer brojenja
  - PTx počinje brojati prema nuli
  - Nakon dosezanja nule smjer brojanja ponovo se mijenja i ciklus se ponavlja
  - Signal je u visokoj razini kada je  $PTx \geq PWx$
  - Zamka: registar PWx u stvari odražava *ispražnjenost* a ne popunjenost signala!
  - Period generiranog signala dan je sljedećim izrazom:  

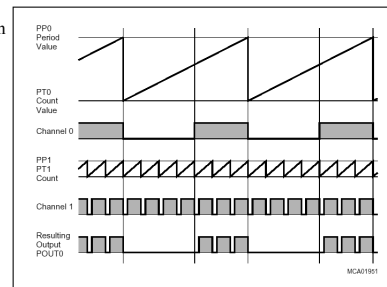
$$PWM\_Period_{model1} = 2 * (PPx + 1)$$

## PWM – način rada 1



## PWM – načini rada

- **Kombinirani (burst) način rada**
  - Kombiniraju se izlazi kanala 0 i 1 odn. 2 i 3
  - Rezultat se dovodi na izlaz kanala 0 odn. 2 – visoka razina postiže se kad je na oba kanala visoka razina => operacija **logičko I**
  - Svaki od kanala može raditi u standardnom ili simetričnom načinu



## Programski model PWM jedinice

- PWM jedinica povezana je sa portom P0
  - izlazi POUT0..3 spojeni su na linije P0.0..P0.3
- Može raditi kao bezuvjetna jedinica
- Može raditi kao prekidna jedinica

```
`ORG 3
PWMIR `DW 20 ; prekidni vektor PWM
```

## Programski model PWM jedinice

- Registri u memoriji

```
PP0 na adresi %D224 ; period kanala 0
PW0 na adresi %D225 ; širina kanala 0
PT0 na adresi %D226 ; brojilo kanala 0
PP1 na adresi %D227 ; period kanala 1
PW1 na adresi %D228 ; širina kanala 1
...
PT3 na adresi %D235 ; brojilo kanala 3
```

## Programski model PWM jedinice

- Registri u memoriji

```
PWMCON0 na adresi %D236 ; kontrolni reg. 0
PWMCON1 na adresi %D237 ; kontrolni reg. 1
PWMIC na adresi %D238 ; kontrola prekida
```

## Primjer programiranja PWM

- **Zadatak:**

Promatra se PWM sklop FRISC-MC sa  $f_{CPU}=20MHz$ .

- Koja je najveća a koja najmanja frekvencija izlaznog signala koju je moguće generirati u zadanom slučaju ako radi u načinu 0?
- O čemu ovisi rezolucija regulacije faktora popunjenosti signala u zadanom slučaju?
- Programirajte PWM kanal 0 tako da generira signal frekvencije 10[kHz] uz faktor popunjenosti 87%. Neka PWM0 radi u načinu 0 uz onemogućene prekide. Nacrtati vremenske dijagrame promjene stanja brojača PWM0 sklopa u vremenu, u ovisnosti o ostalim registrima. Kolika je rezolucija regulacije popunjenosti u ovom slučaju?

## Primjer programiranja PWM.

- Ako je  $f_{PWM}=f_{CPU}$ , a sadržaj registra PPx=1, tada je najveća moguća frekvencija 10[MHz] u načinu rada 0. Ako je  $f_{PWM}=f_{CPU}$ , a sadržaj registra PPx=65535, tada je izlazna frekvencija najmanja moguća i to 4.8[Hz].

Način rada 1 (kao podsjetnik)

## Primjer programiranja PWM..

- Rezolucija faktora popunjenosti direktno je ovisna o vrijednosti registra PPx. Broj koraka unutar kojih se može mijenjati faktor popunjenosti između 0% i 100% upravo je jednak PPx.
- Zadani se oblik izlaznog signala može se postići programiranjem PWM0 sklopa na dva načina, ovisno o faktoru dijeljenja ulaznog signala brojača PT0. Frekvencija rada PT0 može biti  $f_{PWM}=f_{CPU}$  ili  $f_{PWM}=f_{CPU}/64$ .

## Primjer programiranja PWM...

- Za ova dva slučaja se dobiju sljedeće vrijednosti PP0:  
 $f_{IzL} = 10[\text{kHz}]$ ,  $f_{CPU} = 20[\text{MHz}]$

## Primjer programiranja PWM....

- S obzirom na ranije razmatranje o rezoluciji promjene faktora popunjenosti, očito je prvo rješenje povoljnije u tom smislu. Iz toga prolazi i činjenica da će traženi faktor popunjenosti biti moguće preciznije postaviti.
- Ako se odabere  $PP0=1999$ , može se izračunati i potrebna vrijednost  $PW0$  za traženi faktor popunjenosti. Ako se uzme u obzir da promjena faktora popunjenosti iznosi 0.05% kada se  $PW0$  promijeni za 1, može se iz omjera periode i poluperiode, te poznate vrijednosti  $PP0$ , izračunati potrebna vrijednost  $PW0=260$ .

## Postavljanje PWx

- $PWx$  registar određuje graničnu vrijednost koja utječe na postavljanje izlaza (popunjenost signala)
- $PWx$  je *ispražnjenost* signala

## Primjer programiranja PWM.....

## Programiranje PWM0

```
\ORG 0
LOADL R0, %D260      ; faktor popunjenosti
STORE R0, (PW0)
LOADL R0, %D1999    ; period (frekv.) signala
STORE R0, (PP0)
CLEAR R0             ; R0=0
STORE R0, (PWMIC)   ; ne koriste se prekidi
INC R0               ; R0=1
STORE R0, (PWMCON0) ; pokreni PWM0
STORE R0, (PWMCON1) ; omoguci izlaz iz PWM0
IDLE

                        (10 naredbi)
```

## Korištenje prekida

### • Zadatak:

Proširiti program iz prošlog zadatka tako da generira točno  $1000_{10}$  impulsa na izlazu.

Iskoristiti mogućnost da PWM generira prekid nakon svakog ciklusa kanala PWM0 za brojanje impulsa.

Nakon generiranja traženog broja impulsa potrebno je to dojaviti impulsom na liniji P1.0 i ugastiti mikrokontroler.

## Inicijalizacija PWM0

```
ORG 0
JP INIT
ORG 3
DW 30 ; prekidni vektor PWMIR
INIT LOADL R7, %HDF ; inicijalizacija stoga
LOADL R0, %D260 ; faktor popunjenosti
STORE R0, (PWO)
LOADL R0, %D1999 ; period (frekv.) signala
STORE R0, (PFO)
CLEAR R0
STORE R0, (%D128) ; inicijalizacija brojac
LOADL R0, %B1000001 ; ILVL=1, PWMIE=1
STORE R0, (PWMIC) ; inicijaliziraj prekid
WRST %H80 ; omogući prekide
LOADL R0, %H101 ; uključi prekide i ..
STORE R0, (PWMCON1) ; ..omogući izlaz iz PWM0
LOADL R0, 1 ; R0=1
STORE R0, (PWMCON0) ; pokreni PWM0
IDLE
```

(c) Marin Otkić, APR, grupa RASIP

(17 naredbi)

37

## Prekidni potprogram

```
ORG 30
PWMIRQ FUSH R0 ; pohrani kontekst
RDST R0
FUSH R0
LOAD R0, (PWMCON0) ; učitaj PWMCON0
MOVE R0, R1
AND %H1000, R0, R0 ; provjeri je li ovo prekid za PWM0
JE_Z ENDIRQ ; ako nije, izadji
LOAD R0, (%D128) ; brojac impulsa
INC R0 ; povecaj brojac
CMP %D1000, R0 ; dosli do kraja?
CALL_UGE IMPULS ; posalji impuls
STORE R0, (%D128) ; pohrani brojac

ENDIRQ AND R1, %EFFF ; obriši bit za prekid PWM0
STORE R1, (PWMCON0)
POP R0
WRST R0
POP R0
```

(c) Marin Otkić, APR, grupa RASIP

(18 naredbi)

38

## Pomoćni potprogram

; ne pohranjujemo kontekst jer potprogram gasi mikrokontroler

```
IMPULS LOADL R0, 1 ; bit P1.0
STORE R0, (DP1) ; postavi smjer na izlazni

PSET P1, 1 ; postavi P1.0
PCLEAR P1, 1 ; brisi P1.0

PWRDN ; gasi mikrokontroler

; nema povratka iz potprograma
```

(5 naredbi)

(c) Marin Otkić, APR, grupa RASIP

39

## Zaključak

- Inicijalizacija – 17 naredbi
- Prekidni potprogram – 18 naredbi
- Pomoćni potprogram – 5 naredbi
  
- Ukupno:  $17+18+5 = 40$  naredbi (lokacija)
- Zauzeće memorije:
  - ROM: 31%
  - RAM: 1 lokacija + stog

(c) Marin Otkić, APR, grupa RASIP

40