

Računala i procesi

FRISC-MC

Zavod za automatiku i računalno inženjerstvo
grupa RASIP

Uvod u FRISC-MC

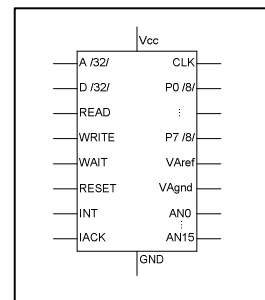
- Osnovna građa mikrokontrolera
- Priključci
- Registri
- Memorija
- Naredbe
- Obrada prekida
- Ugrađene jedinice

Osnovna građa mikrokontrolera

- Baziran na FRISC jezgri (Frisic1)
- U istom kućištu ugrađena jezgra, memorija i vanjske jedinice
 - Promjene priključaka
- Modifikacije u načinu rada jezgre – pretvaranje u mikrokontroler
 - Dodatne instrukcije
 - Promjena obrade prekida

Priključci

- Podrška za vanjsku memoriju
- Podrška za vanjski NMI
- Ne postoje BACK i BREQ
- Osmam 8-bitovnih portova P0-P7
- Analogni ulazi AN0-AN15
- Napajanje AD pretvornika



Načini rada

- RUN način rada:
 - Klasični način rada, kao kod FRISC-a
 - Nakon pokretanja procesor ulazi u ovaj način rada
 - Procesor izvršava program i prihvaća prekide od VJ
- IDLE način rada:
 - Program se ne izvršava (slično NOP), ali se prihvaćaju prekidi
 - Tipičan način rada za mikrokontroler
 - Procesor prelazi u IDLE način izvršavanjem naredbe **IDLE**
- POWER DOWN način rada:
 - Procesor i vanjske jedinice se gase
 - Ponovno pokretanje na RESET
 - Procesor prelazi u POWER DOWN način izvršavanjem naredbe **PWRDN**

Registri

- Isti registri kao kod FRISC-a:
 - R0 - R7
 - PC
 - AR, DR, IR, ...
- Promjene u statusnom registru SR
 - Dodan ILVL (*interrupt level*) umjesto EINTx
 - Nema virtualnih bitova



Memorija

- Ugrađena (on-chip) memorija – ROM i RAM
- Memorijske riječi 32-bitovne, kao i procesorske
 - ROM – 128 riječi
 - RAM – 128 riječi (sadržava i ugrađene vanjske jedinice)
- Vanjska memorija po potrebi
- Brzina komunikacije s internim memorijama i VJ-ma je jedan takt
- S vanjskom memorijom komunikacija traje s jednim stanjem čekanja (2 takta clocka)

Memorija – raspored adresa

- Raspored adresa takav da se ugrađena (on-chip) memorija i vanjska memorija (kao i eventualne VJ koje su spojene *izvana*) iz programa vide na jednak način
- Raspon adresa:

Adresa (dec)	Adresa (hex)	Tip memorije
0 – 127	0x0 – 0x7F	ROM
128 – 255	0x80 – 0xFF	RAM
256 – 2 ³² -1	0x100 – 0xFFFFFFFF	Vanjski RAM i vanjske VJ

Memorija – raspored adresa (2)

- Ugrađene vanjske jedinice zauzimaju dio adresa RAM-a:
 - PIO – 8 adresa
 - Timer – 4 adrese
 - ADC – 4 adrese
 - PWM – 16 adresa
 - Detalji uz opis svake vanjske jedinice
- Slobodno područje RAM-a (stog) počinje od 223, odnosno od 0xDF (SP se inicijalizira na 0xE0, može i 0xDF)

Portovi

- Specifičnost za mikrokontrolere
- Ulazno-izlazne linije (digitalne) s programski podesivim smjerom (ulaz/izlaz)
- Ne vide se u memoriji kao vanjske jedinice
- Koristi se izolirani U/I (*isolated I/O*)
 - Posebne naredbe za rad s portovima
 - Pristupa im se preko oznaka P0-P7
- Na svaki fizički port spojena je PIO jedinica i eventualno još jedna VJ (spojena na sve ili samo neke bitove porta)
- Smjer porta određuje se putem PIO jedinice

Portovi (2)

- VJ spojena na port dijeli port sa PIO jedinicom
- VJ ima prioritet nad PIO – ako se aktivira VJ zauzme odgovarajuće bitove (linije) porta
- Dohvaćanje portova posebnim naredbama (*isolated I/O*):

```
PIN R0, P0 ; citaj vrijednost s P0
POUT R1, P4 ; pisi vrijednost na P4
```

Portovi (3)

- Dijeljenje porta između VJ i PIO:
 - Čitanje:
 - Linija **ulazna** – čita se stanje priključka
 - Nešto **mora** biti spojeno (moguća "bus in high-Z" greška)
 - Linija **izlazna** – čita se stanje koje je postavljeno na priključke
 - Stanje može postaviti VJ ili je prethodno upisano na port
 - Pisanje:
 - Linija **ulazna** – bit se ne postavlja na priključak
 - Linija **izlazna** – ako priključak nije zauzela VJ, bit se postavlja na priključak
- Svi bitovi koji su poslani PIO sklopu za pisanje se pamte, čak i ako ne utječu na stanje priključaka (iako mogu utjecati ako se promijeni smjer priključaka ili PIO preuzme upravljanje portom).

Naredbe

- Jedno- i dvo-ciklusne kao kod FRISC-a
- Preuzete od FRISC-a:
 - `ADD, ADC, SUB, SBC, AND, OR, XOR, CMP, ROTR, ROTL, LOAD, LOADL, LOADH, STORE, PUSH, POP, JP, JR, CALL, RET, RETI, RETN, WRST, RDST`
 - `NOP, CLEAR, COMPL, NEG, SUBR, INC, DEC, TEST, MOVE`
- Nema naredbe **HALT**
- Dodatne naredbe:
 - Načini rada: `PWRDN, IDLE`
 - Rad s portovima: `PIN, POUT, PCLEAR, PSET, PCOMPL, PTEST`

(c) Marin Ošić, ZARI, grupa RASIP

13

Naredbe – PWRDN

- Prebacuje procesor u POWER DOWN način rada
- Prekida simulaciju (kao **HALT** kod FRISC-a)

- Korištenje:

PWRDN ; kraj simulacije

(c) Marin Ošić, ZARI, grupa RASIP

14

Naredbe – IDLE

- Prebacuje procesor u IDLE način rada
- Izvršavanje programa staje, ali se prihvaćaju prekidi (simulacija se nastavlja)
- U stvari pomoćna naredba umjesto `JR .+0`
- Korištenje:

IDLE ; kraj programa, prekidi

(c) Marin Ošić, ZARI, grupa RASIP

15

Naredbe – PIN (port input)

- Čita stanje s porta Px (P0-P7) u registar Rx (R0-R7)
- Portovi su 8-bitovni, u određinom registru se postavlja samo donjih 8 bita

- Korištenje:

PIN R0, P1 ; P1 → R0

(c) Marin Ošić, ZARI, grupa RASIP

16

Naredbe – POUT (port output)

- Upisuje *vrijednost* na port Px (P0-P7)
- Vrijednost se može zadati kao 8-bitovna konstanta (*pod8*) ili putem registra R0-R7 (koristi se samo donjih 8 bita)
- Korištenje:

POUT 2, P7 ; 00000010 → P7
POUT R1, P5 ; R1[0..7] → P5

(c) Marin Ošić, ZARI, grupa RASIP

17

Naredbe – PCLEAR (port clear)

- Briše bitove porta Px (P0-P7) zadane maskom
- Maska se može zadati kao 8-bitovna konstanta (*pod8*) ili putem registra R0-R7 (koristi se samo donjih 8 bita)
- Jedinice u maski označavaju bitove koji se brišu
- Korištenje:

PCLEAR P6, 3 ; 0 → P6[0,1]
PCLEAR P3, R1 ; ~R1&P3 → P3

(c) Marin Ošić, ZARI, grupa RASIP

18

Naredbe – PSET (port set)

- Postavlja bitove porta Px (P0-P7) zadane maskom
- Maska se može zadati kao 8-bitovna konstanta (*pod8*) ili putem registra R0-R7 (koristi se samo donjih 8 bita)
- Jedinice u maski označavaju bitove koji se postavljaju
- Korištenje:

```
PSET P6, 2 ; 1 → P6[1]
PSET P6, R4 ; R4|P6 → P6
```

Naredbe – PCOMPL (port complement)

- Komplementira bitove porta Px (P0-P7) zadane maskom
- Maska se može zadati kao 8-bitovna konstanta (*pod8*) ili putem registra R0-R7 (koristi se samo donjih 8 bita)
- Jedinice u maski označavaju bitove koji se komplementiraju
- Korištenje:

```
PCOMPL P4, 3 ; ~P4[0,1]
PCOMPL P4, R1 ; R1^P4 → P4
```

Naredbe – PTEST (port test)

- Ispituje bitove porta Px (P0-P7) zadane maskom
- Maska se može zadati kao 8-bitovna konstanta (*pod8*) ili putem registra R0-R7 (koristi se samo donjih 8 bita)
- Jedinice u maski označavaju bitove koji se ispituju
- Zastavica Z se postavlja ako su svi ispitani bitovi 0 (uvjet _PZ)
- Zastavica C se postavlja ako su svi ispitani bitovi 1 (uvjet _PS)
- Ako postoje i 1 i 0 na portu pobrisane su i Z i C (uvjet _PX)
- Korištenje:

```
PTEST P0, %B11 ; ispituje P0[0,1]
npr: P0 = 00100100, nakon PTEST : Z=1, C=0
```

Prekidi

- Najveće promjene u odnosu na FRISC
- Osim NMI i RESET, prekidi dolaze isključivo od ugrađenih jedinica (koriste se interni priključci)
- Postoji 7 razina prekida, svakoj jedinici moguće je pridružiti razinu (ako je 0, jedinica ne izaziva prekid)
- Jedinice imaju fiksno pridružene prekidne vektore
- Inicijalna razina prekida je 0, a prekidi nisu dozvoljeni (ILVL=0, GIE=0)

Razine prekida

- Statusni registar SR sadrži polje bitova ILVL koje određuje razinu prekida procesora
- Najniža razina – prekid razine 0 (*nema prekida*)
- Najviša razina – prekid razine 7 (*NMI*)
- Prihvaćaju se samo prekidi *veće* (*strogo veće*) razine od trenutne – postavljene u ILVL bitovima statusnog registra

Prihvaćanje prekida

- Svakoj vanjskoj jedinici se može zadati njena razina prekida
- Svaka jedinica ima pridružen prekidni vektor (fiksno)
- Oprez: ako se ista razina zada dvjema jedinicama, može doći do greške!
Dužnost programera je da zada različite razine

Obrada prekida

- Prihvaćanje prekida
 - Automatski se pohranjuje PC na stog (SR treba ručno)
 - Prihvaćeni nivo prekida sprema se u ILVL bitove (GIE=0)
 - Skače se na pripadni vektor (adrese 1 do 6) od VJ
 - Automatski se potvrđuje prihvat prekida
 - Povratak iz prekidnog potprograma sa **RETI** / **RETN**
- Prekidni **vektori** na adresama 1-6 za VJ
- Na adresama 0 (RESET) i 7 (NMI) nalazi se **naredba**, a ne vektor

Rad s FRISC-MC mikrokontrolerom

- Procesor:
 - Program – inicijalizacija i obrada
 - Prekidi – prijenos podataka
 - Kraj - **HALT**
- Mikrokontroler:
 - Program – inicijalizacija vanjskih jedinica
 - Prelazak u **IDLE** način
 - Prekidi – obrada i prijenos podataka