

# **Teorijska pitanja za završni ispit iz predmeta Projektiranje ugradbenih računalnih sustava**

**Moguća pitanja uz prezentaciju PURS01 (poredana su po redoslijedu pojave u prezentaciji)**

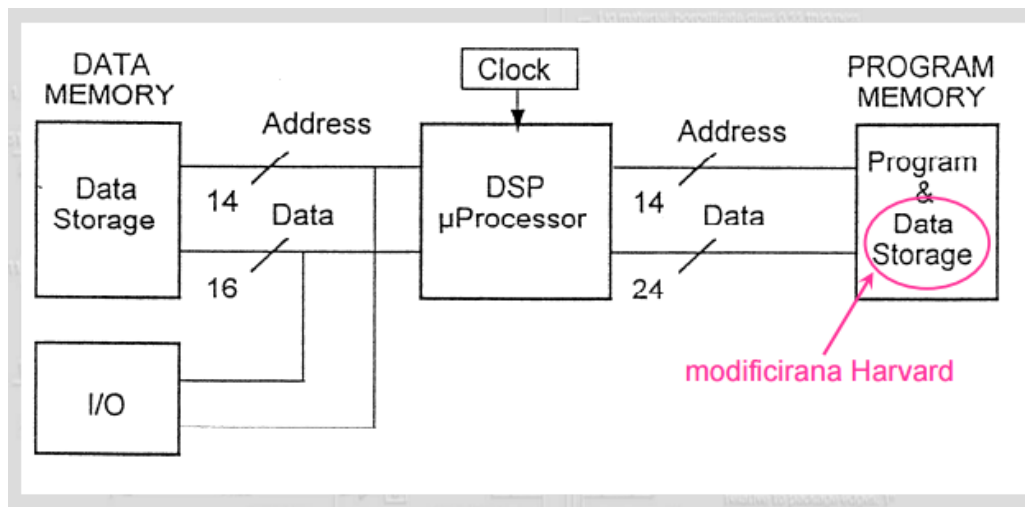
## **1. Opiši ključne razlike u arhitekturi, zadacima i primjenama procesora opće namjene, mikrokontrolera i DSP procesora**

- mikrokontroleri
  - brzina nije kritični zahtjev
  - elementi (ROM, RAM, CPU)
  - prikladni za „toastere“-spori procesi
- procesori opće namjene
  - single chip CPU + vanjsko sklopovlje
  - RISC/CISC
  - nedeterministički odziv, OS, word processing, database management
- DSP procesori
  - CPU optimiran za brzo izvođenje ponovljivih matematičkih algoritama u stvarnom vremenu
  - digital signal processing, motion control, real time execution
  - Brza jedinica za množenje i pribrajanje
  - Arhitektura sa višestrukim memorijskim pristupom

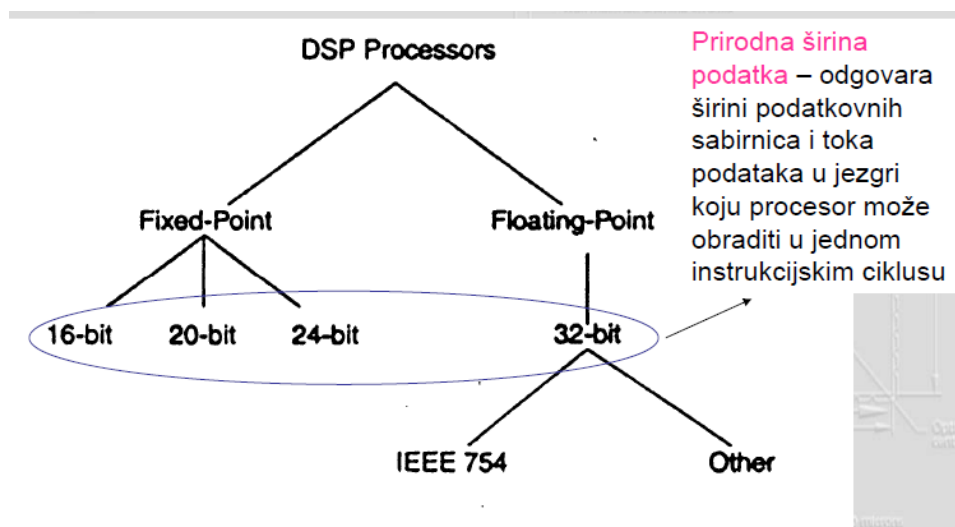
## **2. Nabroji poznatije algoritme i primjene DSP algoritama**

- koderi govora (CELLULAR, PDA)
- kriptografija (CELLULAR, PDA)
- prepoznavanje govora (robotika, automatika, PDA...)
- algoritmi modema (802.11x, cellular..)
- kompresija i dekompresija slika (fotografija, video, igre, PVCR)

### 3. Skiciraj blok dijagram modificirane Harvard arhitekture procesora



### 4. Navedi i kratko opiši moguće reprezentacije brojeva koje se koriste kod DSP procesora



**fixed point** – točka je na fiksnom mjestu

- može biti cjelobrojna (rezultat je prirodni broj) i
- frakcionalna (razlomačka) – (rezultat je decimalni broj) u području -1 do 1 (ili drugi interval, ovisno o broju bitova ispred točke)
- preciznost od 16 / 20 / 24 bita

**floating point** – prikaz pomoću mantise i eksponenta

- Širi dinamički opseg i bolja preciznost
- dsp-jednostavniji, ali skuplji i sporiji

**5. Opiši postupak pretvorbe cjelobrojnog b-bitnog podatka u zapisu dvojnog komplementa u odgovarajuću frakciju 1.(b-1) ili 3.(b-3). Koji je raspon prikaza brojeva za ove dvije frakcije.**

prvo treba provjeriti može li se zadani broj uopće napisati u odgovarajućoj frakciji.  
postupak neznam, nešto muljaš s bitovima.

signed dvojni komplement ide od  $-2^{(b-1)}$  do  $2^{(b-1)+1}$

unsigned dvojni komplement ide od 0 do  $2^b-1$

**frakcija 1.** znači da je vodeći bit jednobitni i da on odlučuje o predznaku. ostalih b-1 bit odlučuje o težini.

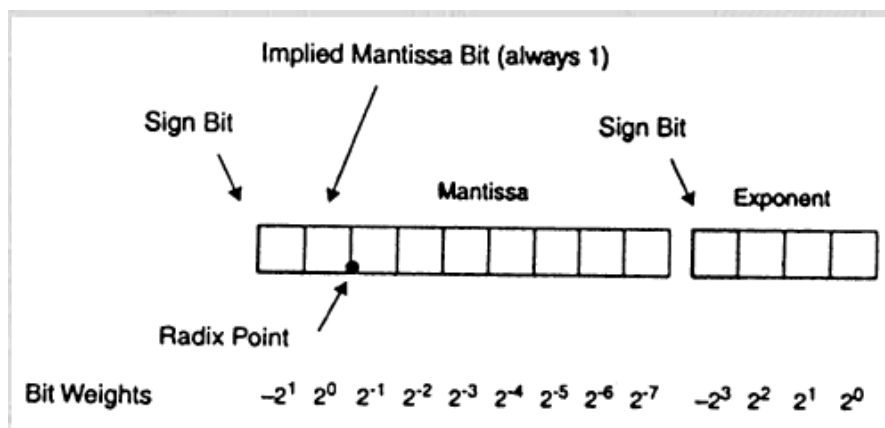
ukupni prikaz je od -1 do  $1-2^{(b-1)}$

**frakcija 3.** znači da je vodeći bit trobitni i da on njegov prvi bit odlučuje o predznaku. ostalih 2 bita odlučuju o vodećoj jedinici, a oni preostali b-3 bitovi odlučuju o težini.

ukupni prikaz je od -3 do  $3-2^{(b-3)}$

**6. Opiši i na jednostavnom primjeru ilustriraj zapis brojeva s pomičnim zarezom. Što je to „pretpostavljeni“ najviši bit mantise?**

tutorial: <http://www.oxfordmathcenter.com/drupal7/node/43>



najviši bit mantise je **predznak**.

prikaz pomoću mantise i eksponenta  
value = mantissa x  $2^{\text{exponent}}$

## 7. Što je aritmetika proširene preciznosti i na koji način je podržava arhitektura DSP procesora?

To je reprezentacija koja osigurava veću točnost u odnosu na prirodnu širinu korištenjem:

- sklopovskih proširenja (accumulator 2x širi)
- slijed instrukcija jednostruke preciznosti

## 8. Što je block-floating-point zapis i kod kojih primjena se često koristi?

– blok podataka sa različitim mantisama ali zajedničkim eksponentom (primjena kod FFTa)

## 9. Opiši moguće izvedbe procesora

- single-chip
- višestruke jezgre na istom chipu
- DSP temeljen na ASIC-u ([Application-specific integrated circuit](#))
- višeprocorski sustavi

## 10. Navedi par glavnih proizvođača DSP procesora i poznatije porodice procesora tog proizvođača

Vendor	Part	Chip, Core, Both
Agere	DSP16xxx	Chip
Analog Devices	AD21xx, AD2106x, AD2116x	Chip
ARM	ARM9	Both
Infineon	Tricore1	Both
Motorola	DSP56xxx	Chip
StarCore	SC140	Core
Texas Instruments	TMS320C5xxx, TMS320C6xxx,	Chip

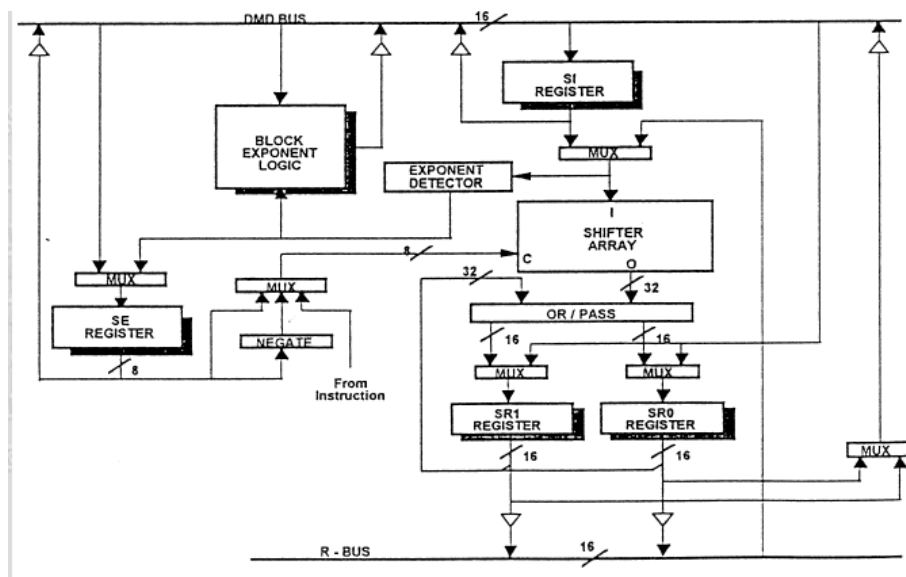
## 11. Nabroji koje su temeljne jedinice arhitekture procesora porodice ADSP21XX i ukratko opiši njihove funkcije.

- aritmetičko logička jedinica – ALU
  - aritmetičke i logičke operacije
  - 16 bitni ulazi i izlaz
  - 6 statusnih zastavica

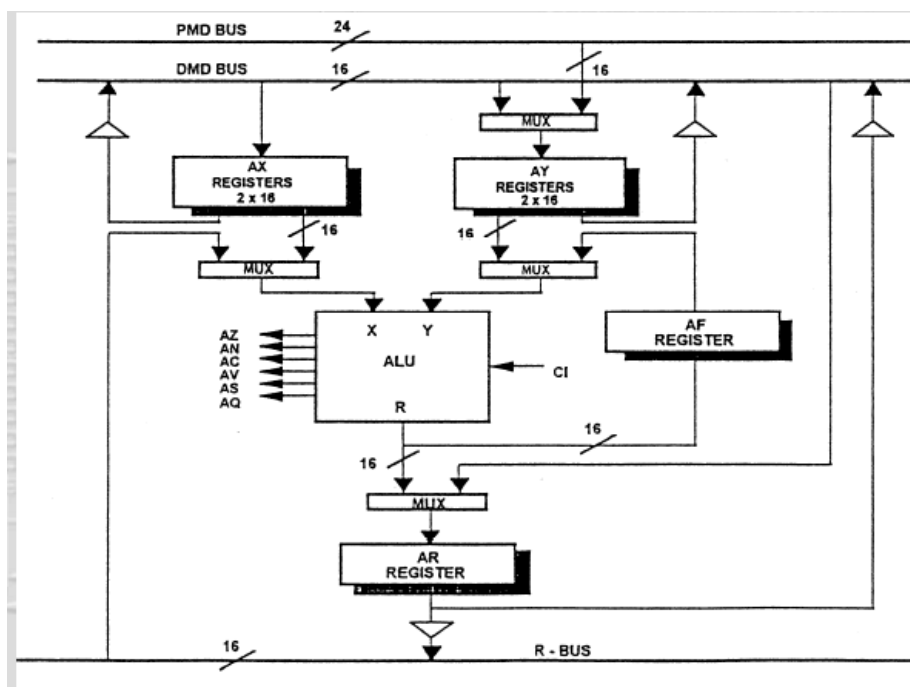
- rad sa zasićenjem (saturation)
- **množačko / pribrojnička jedinica – MAC**
  - provodi množenja
  - 16 bitni ulazi, 32bitni izlaz
  - sklopovsko izvedeno zaokruživanje
- **jedinica za posmak –SHIFTER**
  - aritmetički, logički posmak, normalizacija
  - sve se provodi u jedno koraku – važno !
  - barrel shifter
- **adresni generator – DAG**
  - generatori adresa za istovremeni pristup podatkovnoj i programskoj memoriji
  - DAG1 i DAG2
  - DAG1 adresira podatkovnu memoriju
  - DAG2 oboje (programsku i podatkovnu)
  - adresiranje (direktno, indirektno, modulo-cirkularno)
- **jedinica za kontrolu programskog toka**
  - generira slijed instrukcija i upravlja programskim tokom
  - adresa dolazi iz (PC, PC stack, IR-instruction register, prekidni kontroler)

**12. Skiciraj blok dijagram ALU jedinice procesora ADSP21XX, te označi odgovarajuće registre i sabirnice kao i pripadne širine.**

ma tko će to znat, su oni normalni??!!  
al ajd evo, za nadobudne:



13. Opiši moguće argumente X i Y grane ALU jedinice.



-argumenti su 16 bitni.

-4 ulazna registra (AX0,AX1,AY0,AY1)

14. Opiši i na primjeru ilustriraj mod rada ALU jedinice sa zasićenjem. Zašto i kada je važna primjena ovog moda rada?

**Preljev** –rezultat akumulacije premašuje maksimalni pozitivni ili maksimalni negativni broj odabrane reprezentacije

**Izbjegavanje preljeva** – primjena moda zasićenja (saturacije)

$$\begin{array}{r}
 0100 \\
 + 0101 \\
 \hline
 1001 \dots -7 \\
 0111 \dots 7 \text{ sat. (9)}
 \end{array}$$

prednosti: The result is as numerically close to the true answer as possible

*Wiki: When adding two numbers using two's complement representation, overflow results in a 'wrap-around' phenomenon. The result can be a catastrophic loss in signal-to-noise ratio in a DSP system. Signals in DSP designs are therefore usually either scaled appropriately to avoid overflow for all but the most extreme input vectors, or produced using saturation arithmetic components*

**15. Objasni moguće načine zaokružanja. Kada se koristi zaokruženje? Ilustriraj primjerom rekurzivnog digitalnog filtra 2. reda te označi točke strukture u kojima se provodi ova kvantizacija (zaokruženje).**

**Zaokruženje** –prilikom redukcije preciznosti zapisa rezultata ALU ili MAC operacije s dvostruke na jednostruku preciznost

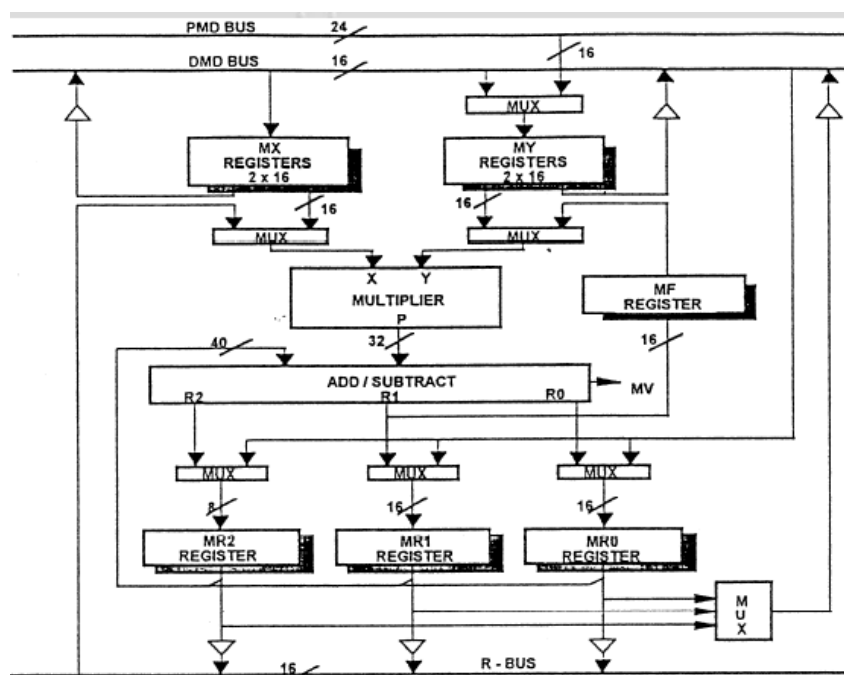
**tipovi:**

- **odsijecanje** – odbaci LSB, zaokruži na prvi manji cijeli broj
- **na najbliži cijeli broj** - (pribroji polovicu LSBa pa onda odsijeci; srednja točka (0.5) se uvijek zaokružuje prema gore)
- **konvergentno zaokruženje** (razlika u odnosu na prethodni je samo u ponašanju za srednju točku)

**Rekurzivni digitalni filter** - ništa to, ako netko zna, neka dopiše

**16. Skiciraj blok dijagram MAC jedinice procesora ADSP21XX, te označi odgovarajuće registre i sabirnice kao i pripadne širine.**

kae ovo fora neka a ?!  
al ajd, evo:



**17. Opiši moguće operacije koje može provoditi MAC jedinica, te diskutiraj širine svih ulaznih operanada i izlaznih registara.**

operacije:

- množenje
- množenje sa kumulativnom pribrajanjem
- množenje sa kumulativnom oduzimanjem + zasićenjem

ulazi: 16 bitni, izlaz: 32 bitni

+izlaz 32 bitni se proslijeđuje na ulaz 40bitnog akumulatora.

**18. Opiši moguće argumente X i Y grane MAC jedinice.**

- 4 ulazna registra (MX0,MX1,MY0,MY1)
- podržava proizvoljni tip ulaznih argumenata (sa ili bez predznaka)

**19. Opiši namjenu posmačnog sklopa (moguće operacije koje provodi) i obrazloži pojam „barrel“ shifter**

- Ulazna 16 bitna vrijednost se posmiče ulijevo ili udesno i pohranjuje u 32-bitnom izlaznom registru
- aritmetički, logički posmak, normalizacija
- sve se provodi u jedno koraku – važno !

A barrel shifter is a [digital circuit](#) that can [shift](#) a [data word](#) by a specified number of [bits](#) in one [clock cycle](#).

**20. Objasni smisao HI/LO reference za posmak i primjenu OR/PASS logike, te ilustriraj primjerom naredbi.**

**HI/LO** označava viši ili nižu riječ (16 bita) koji će se posmicati.

**OR/PASS :**

Ako je odabran **OR**:

- provodi operaciju ILI između (SR0+SR1) i posmačnog polja (SHIFTER ARRAY)

Ako je odabran **PASS**:

- proslijeđuje (SR0+SR1) na BUS / MUX

**21. Što su višefunkcijske instrukcije, te koje naredbe je moguće kombinirati. Objasniti pipe-line podataka, tj. kako se argumenti naredbi prijenosa (čitanje ili spremanje) odnose s argumentima ALU i MAC naredbi**

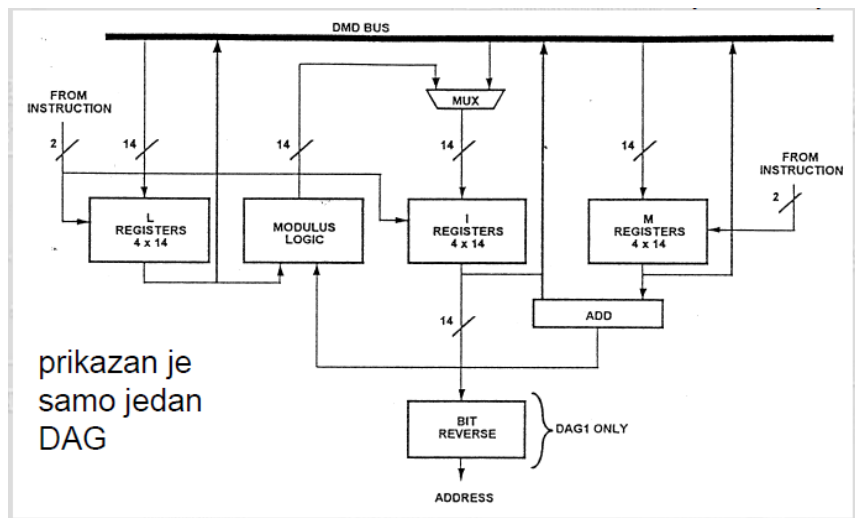
**višefunkcijske instrukcije** - Moguće je kombinirati jednu ALU ili jednu MAC operaciju sa 2 podatkovna dohvata u jednom instrukcijskom ciklusu!

**princip pipelinea:** Registri (memorija) se čitaju na početku instrukcijskog ciklusa a upisuju na kraju istog ciklusa

**22. Skiciraj blok shemu DAG jedinice (samo DAG1), te označi odgovarajuće registre i sabirnice kao i pripadne širine.**



-ma bravo, al ajd možda bi se moglo:



**23. Opiši svrhu svih registara DAG jedinica, te obrazloži mogućnost adresiranja DM i PM memorije pomoću DAG1 i DAG2.**

- 3 register file-a x 4 registra:
  - **I index** (I0-I3 & I4-I7) – početna adresa
  - **M modify** (M0-M3 & M4-M7) – za koliko se promijenila sljedeća adresa u odnosu na prethodnu
  - **L length** (L0-L3 & L4-L7) – koliki je raspon od min do max moguće adrese s obzirom na početnu adresu i M registar. ( detaljnije je objašnjeno na materijalima, 10. zadatak )
- DAG1 adresira podatkovnu memoriju (DM)
- DAG2 adresira obje memorije (podatkovnu DM ili programsku PM)

**24. Opiši moguće načine adresiranja koji se mogu ostvariti pomoću DAG jedinica.**

mogući načini adresiranja:

- **direktno**
- **indirektno**
- **modulo - cirkularno**

**25. Što je to cirkularni (kružni) spremnik. Na koji način se inicijalizira, te kako se koristi. Ilustriraj primjenom za spremnik stanja IIR filtra (FIR).**

Cirkularni spremnik je struktura implementirana u hardveru. Premotavanje indeksa se događa automatski, nije potrebno ispitivati je li indeks prešao max.moguću vrijednost-več se to obavi sklopovski.

---

**Algoritam 1** Inicijaliziraj FIR filter

---

```
1: for  $k = 0$  to  $L - 1$  do
2:    $s[k] \leftarrow 0$ 
3:    $h[k] \leftarrow k$ -ti koeficijent filtra
4: end for
```

---

gdje je; s-cirkularni međuspremnik stanja, h-impulsni odziv filtra

---

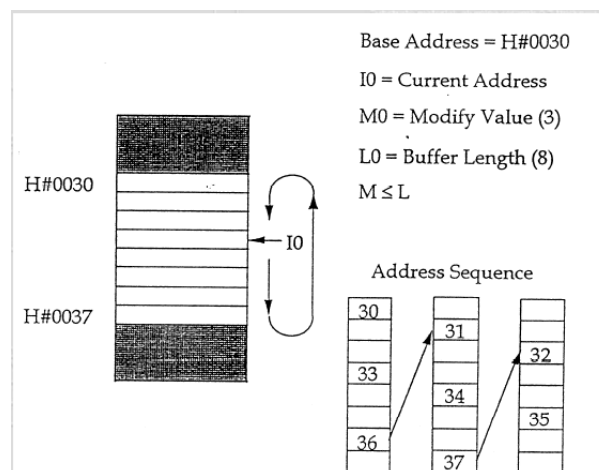
**Algoritam 2** Izračunaj jedan izlazni uzorak za jedan novi ulazni

---

```
1:  $A \leftarrow 0$ 
2:  $s[j] \leftarrow x[n]$ 
3:  $j \leftarrow (j + 1) \bmod L$ 
4: for  $k = 0$  to  $L - 1$  do
5:    $A \leftarrow A + s[j]h[k]$ 
6:    $j \leftarrow (j + 1) \bmod L$ 
7: end for
8:  $y[n] \leftarrow A$ 
```

---

ilitiga grafički:



## 26. Koje su glavne funkcije koje obavlja generator programskog toka?

- Generira slijed adresa instrukcija i upravlja programskim tokom

## 27. Navedi koji su mogući izvori naredne adrese instrukcije.

Adresa slijedeće instrukcije može doći iz jednog od slijedećih izvora:

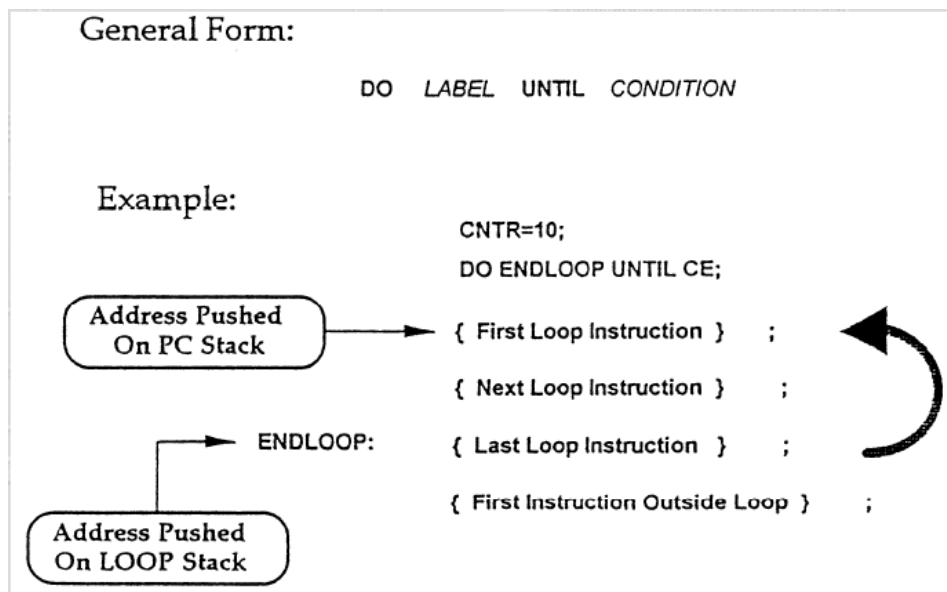
- **PC (programsko brojilo)** – za slijedni tok instrukcija
- **PC stack (programski stog)** – povratak iz potprograma, povratak iz prekida, ili skok na vrh DO UNTIL petlje
- **Instruction register (instrukcijski registar)** – direktni skok na adresu definiranu u instrukciji
- **Interrupt controller (prekidni kontroler)** – prilikom opsluživanja prekida

## 28. Kako je izveden PC stog i za koje naredbe se koristi?

- sklopovski izveden (programski stog) – neka vas to ne zbuni, uglavnom **sklopovski** izveden
- dubine je 16 riječi
- koristi se za programske petlje (DO UNTIL)

## 29. Opiši naredbu za izvođenje DO petlje te objasni koje sklopovlje unutar generatora programskog toka sudjeluje pri njenom izvođenju. U čemu je njen značaj?

1. Kad se uđe u petlju, i zadovoljen je uvjet, stavlja se adresa prve instrukcije na PC stack (To nam je potrebno da bi mogli nakon ENDLOOP labele skočiti natrag na adresu prve instrukcije)
2. Na kraju kod ENDLOOP petlje adresa zadnje instrukcije u toj petlji stavlja se na LOOP stack (To nam služi zato da kad se vratimo gore i provjerimo uvjet, a on nije zadovoljen, skoči na kraj petlje i izađe iz nje.)



**Sklopovlje:** Sklopovska izvedba petlje (brojač i povrat), PC stack, LOOP stack

## 30. Što su prekidi i prekidni vektori, koji su mogući izvori prekida i na koji način procesor posluhuje ove prekide?

**prekid** je vanjski događaj koji izaziva prekid programskog toka i grananje na specijalni blok instrukcija (cijela prekidna rutina ili samo njen poziv)

**Izvori prekida:**

- periferijski sklopovi na chipu
- vanjske prekidne linije
- programski prekidi (traps, exceptions)

Posluživanje prekida koristeći **Interrupt vectors** – lokacije u memoriji pridružene svakom izvoru prekida

### 31. Objasni što je protočna izvedba toka instrukcije.

- **Koristeći pipelining** – razbijanje slijeda operacija u manje dijelove koji se izvršavaju paralelno u vremenu

	Clock (Instruction) Cycle							
	1	2	3	4	5	6	7	8
Instruction Fetch	I1	I2	I3	I4	I5	I6	I7	I8
Decode		I1	I2	I3	I4	I5	I6	I7
Data Read/Write			I1	I2	I3	I4	I5	I6
Execute				I1	I2	I3	I4	I5

- Primjer protočne strukture dubine 4 stupnja.

### 32. Koji periferni sklop se koristi za povezivanje DSP procesora s vanjskim A/D i D/A pretvornicima?

- Paralelni U/I portovi

### 33. Objasni na koji način se kod DSP procesora može smanjiti prosječna potrošnja energije, te koji su tipični taktovi na kojima rade ovi procesori.

smanjenje potrošnje energije:

- smanjenje frekvencije
- smanjene napona

Tipični taktovi:

- **conventional DSPs – 20-50 MHz**
- **midrange DSPs - 100-300 MHz**
- **modern DSPs - up to 1 GHz**

# Teorijska pitanja PURS02

## 1. Opiši prednosti i mane DSP procesora sa stanovišta primjene u ugradbenim računalnim sustavima.

- Jednostavna jezgra sa moćnim izvršnim jedinicama
- Jednostavni memorijski model i Harvard arhitektura
- Skup instrukcija i arhitektura strogo prilagođeni obradi signala
- Izvrstan odnos procesnih mogućnosti i potrošnje
- Točno predvidivo vrijeme izvođenja

## 2. Za što se koristi koncept hijerarhijskog računalnog sustava i koje su zadaće pojedinih njegovih dijelova?

Koristi se zbog mana DPS procesora:

- Mali memorijski prostor
- ne podržava (ili jako teško) OS
- nedostatak instrukcija
- nije OpenSource friendly

Radi se o sustavu sa dva procesora:

- glavni procesor je neka popularna i dobro podržana jezgra klasične arhitekture (npr. RISC) , te se na njemu pokreće OS i glavna aplikacija
- DSP se koristi kao pomoćni procesor koji je upravljan od glavnog i provodi zadatke isključivo iz svoje domene

## 3. Opiši moguće načine povezivanja procesora u hijerarhijskom modelu, te objasni moguće mane takvog koncepta.

Načini povezivanja:

- Dijeljeni zajednički memorijski prostor
- Komunikacija preko tzv. „Host port-a“
- FIFO među-spremnik
- Dvosmjerna serijska veza preko jednog ili više serijskih kanala

Moguće mane:

- Problem učinkovitog raspoređivanja zadataka između dva procesora
- Nužan razvoj dva programa
- Potrebno koristiti dva različita i često nekompatibilna razvojna sustava
- Potreba za dva razvojna tima

- Problem kod intenzivne razmjene podataka između procesora
- Smanjena propusnost kod overhead protokola
- Problem učinkovite sinkronizacije dva procesora
- Povećana potrošnja

#### **4. Koja moderna rješenja se koriste kao alternativa diskretnom hijerarhijskom modelu sustava za obradu signala?**

- System On Chip – integracija oba procesora i pripadnih sustava povezivanja na istom chipu
- Modifikacija procesora opće namjene dodavanjem DSP izvršnih jedinica i pripadnih instrukcija
- Razvoj potpuno nove arhitekture prilagođene takvom dvojnomo konceptu (npr. Blackfin)

#### **5. Opiši ključne značajke arhitekture procesora Blackfin porodice. Odakle izvršne jedinice čitaju argumente operacija i gdje pohranjuju svoja rješenja?**

Ključne značajke:

- Kombinira 32-bitne RISC instrukcije sa 16-bitnim DSP instrukcijama
- JEDINSTVENA JEZGRA, ne dvije paralelne
- Velike procesne mogućnosti i mala potrošnja
- Posjeduje SIMD arhitekturna proširenja za ubrzano izvođenje algoritama obrade slike i video signala

Koristi se univerzalni registarski stog 2x8x16 bita kao izvor i cilj svih ALU i MAC operacija - >ortogonalnost instrukcijskog skupa.

#### **6. Što je pojam ortogonalnosti skupa instrukcija i u kakvom je odnosu sa programiranjem procesora u višim programskim jezicima?**

Rastavljanje izvršnih instrukcija(16/32 bitne) na 8-bitne , te se time ubrzava izvođenje -> SIMD istovremena obrade više malih argumenata. Time se olakšava programiranje i prevođenje viših programskih jezika.

## **7. Na koji način se kod procesora Blackfin porodice ostvaruju uštede u potrošnji i zbog čega je to važno?**

Hibridnom jezrgom se eliminira potreba za posebnim procesorom koji obavlja samo „kontrolerske zadaće“, pa se time ostvaruju uštede u sklopovlju, u potrošnji i razvoju programske podrške.

Mala potrošnja je izuzetno značajna za primjene kao npr. bežične širokopojasne mreže, portabilni audio i video uređaji povezani na internet i mobilne komunikacije.

## **8. Što su to SIMD arhitekturna proširenja? Objasni koje izvršne jedinice i koje instrukcije Blackfin jezgre odgovaraju tom SIMD konceptu.**

Podrška za prirodnu širinu argumenata od 8-bita. Rastava izvršnih jedinica širine 16 ili 32 bita na dvije ili četiri 8-bitne izvršne jedinice koje obrađuju nezavisne ulazne 8-bitne podatke. Dakle, istovremena obrada više malih argumenata.

- Operacije zaokružanja kod DCT transformacije za primjene u JPEG kompresiji
- Instrukcije za izračunavanje sume apsolutnih vrijednosti razlika koja se koristi kod algoritama za procjenu pomaka scene u MPEG2 i MPEG4

## **9. Što je DMA podsustav i za koje namjene se može koristiti DMA podsustav u Blackfin procesorima?**

Višestruki nezavisni DMA kontroleri koji podržavaju automatizirani prijenos podataka uz minimalno opterećenje procesorske jezgre, za npr. prijenos između interne memorije i DMA podržanih periferija ili između periferija i vanjskih sklopova spojenih na vanjsku memorijsku sabirnicu (SDRAM, vanjska asinkrona memorija).

## **10. Kakve instrukcije za obradu slike i video signala posjeduje Blackfin jezgra? Koje su širine ulaznih i izlaznih argumenata ovih operacija? U kojim algoritmima se navedene instrukcije koriste?**

Instrukcije:

- Zaokružanja kod DCT transformacije za primjene u JPEG kompresiji
- Izračunavanje sume apsolutnih vrijednosti razlike koja se koristi kod algoritama za procjene pomaka scene u MPEG2 i MPEG4

Širine su 16 ili 32 bitne koje se rastavljaju na manje 8-bitne argumente. Koriste se u algoritmima za kodiranje audio signala, slike i videa.

### **11. Koja je prednost potpuno programske izvedbe algoritama obrade signala i koja svojstva mora posjedovati DSP jezgra da bi to bilo moguće?**

Omogućava laganu prilagodbu na stalno nadolazeće nove standarde, te mogućnost programske izvedbe algoritama koji su nedavno bili izvedivi isključivo sklopovskim rješenjima. Mora imati SIMD arhitekturna proširenja i DMA podršku.

### **12. Navedi tipične ugrađene periferijske sklopove (kontrolere) u Blackfin procesorima koji su značajni za ostvarenje zadaća iz domene mikrokontrolera.**

10/100 Ethernet MAC, UART, SPI, CAN kontroler, vremenski sklopovi s podrškom za PWM, Watchdog sklop, RTC, potpuni kontroler za asinkronu i sinkronu vanjsku memoriju.

### **13. Što je to hijerarhijski memorijski model i za koje zadaće je on od posebnog značaja?**

Podrška za Level 1 (L1) i Level 2 (L2) internu memoriju.

- L1 memorija je neposredno vezana sa jezgrom i radi na punom taktu, te ostvaruje najviše brzine rada za vremenski kritične programske odsječke
- L2 memorija je veća, nešto nižih brzina rada, ali ipak brža od vanjske memorije

L1 memorija je prilagođena brzom izvođenju algoritama obrade signala, uz zadržavanje maksimalne jednostavnosti programskog modela slično kao kod običnih kontrolera. Podjela L1 memorije na pričuvnu L1 memoriju i/ili SRAM memoriju radi ostvarenja najveće propusnosti uz minimalnu i predvidivu latenciju i istovremeno ubrzavanje izvođenja zadataka sa manje strogim vremenskim uvjetima.

### **14. Koji su mogući načini konfiguriranja L1 memorije kod Blackfin procesora? Za koje namjene se koriste pojedini modovi rada ove memorije?**

Konfiguracija L1:

- Isključivo pričuvna memorija
- Isključivo SRAM memorija direktno mapirana u memorijski prostor jezgre



- Kombinacija pričuvne i SRAM memorije

Namjene u prethodnom pitanju.

### **15. Čemu služi MMU sklop i koji je njegov značaj u računalnom sustavu koji je upravljan od strane operacijskog sustava za rad u stvarnom vremenu?**

MMU je ugrađeni kontroler za zaštitu memorijskih prostora. Nužan je preduvjet za ugradnju operacijskog sustava za rad u stvarnom vremenu (RTOS). MMU osigurava međusobnu izolaciju aplikacija, te siguran, pouzdan i robustan rad operacijskog sustava i samih aplikacija.

### **16. Koja je razlika između nadgledničkog i korisničkog načina rada jezgre?**

U nadgledničkom modu RTOS upravlja korištenjem i raspodjelom resursa između aktivnih programskih zadataka koji se izvode u korisničkom modu.

### **17. Na koji način Blackfin procesor ostvaruje visoku gustoću programskog koda? Kratko opiši način dohvata instrukcija iz programske memorije.**

Procesor podržava miješanje 16 i 32 bitnih instrukcija u paketima od 64 bita radi boljeg iskorištenja programske memorije.

Kod punjenja pričuvne memorije i dohvata instrukcija jezgra koristi punu 64 bitnu širinu, jer nema ograničenja vezana uz poravnanje.

### **18. Opiši alate za razvoj koji se koriste za Blackfin porodicu.**

- IDDE – integrated software development and debugging environment
- VisualDSP++ 5.0
- 

### **19. Koji ključni indikatori performansi su uspoređivani u BDTI analizi DSP procesora?**

- Delayed LMS filter
- 256pt Complex FFT
- Block FIR Filter
- Complex FIR filter
- Max Search

- Brzina po omjeru \$
- Brzina
- Korištena memorija

## **20. Što obuhvaća Open Source podrška Blackfin jezgre?**

- Raspoloživ je potpuni GNU paket besplatnih razvojnih alata za Blackfin
- Linux jezgra uCLinux distribucija već prilagođena za rad na Blackfin arhitekturi
- „Das U-Boot“ – boot-loader korišten za pokretanje uCLinux iz FLASH memorija

by: Žagar & Žagar