AADSP2 teorija

ELIMINACIONI

1. **Šta podrazumeva Harvard arhitektura?**

POGRESNO a) postojanje MAC instrukcije

POGRESNO b) protocnu arhitekturu

POGRESNO c) arhitekturu sa veoma dugom instrucijksom reci

**TACNO d) harvard arhitektura znači odvojenu memoriju za podatke i instrukcije**

**2. Zaokruziti sprege karakteristicne za komunikaciju izmedju glavne procesorske jedinice(mikrokontolera)…**

POGRESNO a) Ethernet

**TACNO b) I2C**

POGRESNO c) I2S

POGRESNO d) HDMI

**TACNO e) SPI**

**3. Navedi bar četiri bita iz registra stanja?**

zero (ALJ nula),

carry(ALJ prenos)

sign(znak)

overflow(ALJ prekoracenje opsega),

negative(ALJ negativna vrednost),

extension (prosirenje)

**4. Koliki je dinamički opseg u slučaju 32-bitnih vrednosti u nepokretnom (fixnom) zarezu?**

POGRESNO 96db

**TACNO SNR(dB) = 187dB ~ 31bit x 6.02dB**

POGRESNO 1535db

**5. Koliki je dinamički opseg u slučaju 32-bitnih vrednosti u pokretnom zarezu?**

SNR(dB) =1535dB

**6. Koja funkcinalna jedinica je odgovorna za aritmetiku zasićenja?**

SRS (shift round saturate)

**7. Binarni zapis 1010110100 je broj u nepokretnom zarezu u formatu s.9, kolika je njegova vrednost?**

POGRESNO a) -0.3515625

POGRESNO b) -0.32421875

**TACNO c) -0.6484375**

POGRESNO d) -1.296875

**8. Dat ti je binarni broj tipa 1101101000 u formatu <s.9> treba odrediti koja je njegova decimalna vrednost?**

**TACNO a) -0.296875**

POGRESNO b) -1.703125

POGRESNO c) 0.1296875

POGRESNO d) 1.703125

**9. Navedi bar 3 izvora prekida kod DSP procesora?**

a) prekid od strane neke od periferija

b) prekid od strane hosta

c) prekid izazvan programom

**10. Nabrojati tri izvora prekida.**

a) periferije u okviru integrisanog kola  
b) spoljne linije prekida  
c) softverski prekidi – zamke ili izuzeci

**11. Za šta služe akumulatorski registri?**  
 Služi da se čuvaju krajnje i međurezultate MAC instrukcije i drugih aritmetičkih operacija

**12. Za šta služe registri stanja?**  
 Služi za smeštanje informacije o rezultatu dobijenom u toku aritmetičkih operacija DSP.

**13. Cemu je namenjen DMA?**  
Direktni memorijski pristup je tehnika kojom se podaci mogu prenositi iz/u memoriju bez uključenja procesora u taj proces.

**14. Zaokruziti tacne odgovore za registarsko indirektno adresiranje sa modulo adresnom aritmetikom?** POGRESNO a) adresirani podaci su u registru

**TACNO b) adresirani podaci su u memoriji**

POGRESNO c) adresa se racuna u ALJ

**TACNO d) adresa se racuna u adresnoj jedinici**

**15. Definisati MAC jedinicu?** a=x\*y+a

**16. Napisati MAC instrukciju?**

A = A + C\*D

ISPITNI

**1. Navedi osnovne elemente jezgra DSP procesora?**

Skup registara (A0-A3), skup registara (B0-B3), akumulatori, privremeni registri , registar upravljanja rezimom rada, registar stanja, dekodiranje instrukcije i distribucija signala, mnozac, multiplekseri, ALJ,sabirac.

**2. Koliko MAC jedinica sadrzi jedno jezgro Currus Logic C49x DSP procesora?**

Dve MAC instrukcije

**3. Navedi 5 tipova instrukcija DSP procesora?**

Aritmeticke operacije i mnozenje, logicke operacije, pomeranje, rotacija, poredjenje, petlje, grananje poziv potprograma i povratak, grananje, uslovno izvrsavanje instrukcija, instrukcije za manipulaciju sa bitima, podrska peralelnom premestanju, ortogonalnost, upravljanje izvrsenjem, hardverska petlja, prekidi.

**4. Navedi barem 3 nacina adresiranja podrzanih od strane adresnog generatora kod C49x?**

Neposredno, posredno, modulo adresiranje, linearni rezim adresiranja, posredno adresiranje sa bitobrnutim redosledom.

**5. Navesti 9 rutina koje su definisane unutar MCT tabele bilo kog modula?**

MCT tabela je niz od devet elemenata – pokazivaca na osnovne javne (public) rutine. Redosled

elemenata u tabeli je unapred definisan, a ukoliko neka od rutina nije definisana za dati modul, na mestu njenog pokazivaca se nalazi nula. Ove rutine poziva OS kao odgovor na pojavu odgovarajucih dogadjaja u sistemu.

PreKickstart, PostKickstart, Timer, Frame, Bock, AFAP, Background, PostMalloc, PreMalloc.

**6. Kada se pozivaju Block i Frame rutina od strane programskog okruzenja?**

Block nakon svakog stizanja bloka odbiraka, a Frame nakon jednog frame-a kojeg cine djuture pristigli blokovi.

**7. Sta je prednost konvergentnog zaokruzivanja?**

Ne unosi pomeraj u vrednost ulaznog signala.

**8. Koji registri su neophodni da bi bila podržana hardverska petlja? (proveri!!!)**

START, STOP, brojac ciklusa

**9. Koja je funkcija komandne datoteke povezivaca?**

Njihova funkcija je smestanje C sekcija u memoriju.

**10. Objasni funkciju adresnog generatora?**

Adresni generator se sastoji iz skupa od 12 parova registara – 16bitnih indeksnih registara (i0-i11) i 16- bitnih modulo-ofset registara (nm0-nm11). NM registri sadrze 4-bitni moduo deo (biti[15:12]) i 12-bitni ofset deo. Ofset deo azurira sadrzaj indeksnog registra a modulo deo sluzi za specificiranje jednog od tri tipa adresiranja: linearnog, reverznog binarnog ili modulo adresiranja. Ofset deo se tretira kao 12-bitni oznacen broj, i kao takav moze azurirati adresu u odgovarajucem indeksnom registru za vrednost u opsegu od -2048 do 2047. Adresni generator je u stanju da obavi izvestan broj aritmetickih operacija i da generise adresu na kojoj se nalazi podatak.

**11.Namena akumulatorskih registara kod procesora koji koriste aritmetiku u nepokretnom zarezu je:**

Skladistenje rezultata obrade

**12.Cemu sluzi MCV tabela i cemu sluzi kod radnog okruzenja C49x DSP procesora?**

MCV tabela predstavlja niz javno dostupnih konfiguracionih parametara datog modula, i ona omogucava konfigurisanje modula od strane glavnog kontrolera uredaja (host). Struktura ove tabele nema neku unapred definisanu formu i programeru je prepušteno da formira njen sadržaj i strukturu.

**13. Kada se poziva Background rutina od strane programsok okruzenja Cirrus C49x DSP procesora?**

Izuzetak je jedino Background rutina koja je prekidiva. Ona je najnižeg prioriteta. Background rutina se izvršava periodicno u pozadinskoj niti (Background thread), koja ima niži prioritet od Foreground niti. Njeno izvršavanje može da bude prekidano drugim rutinama višeg prioriteta (npr. Timer, Frame, Block ili AFAP rutinama) pri cemu se OS brine o ocuvanju zatecenog stanja rutine – konteksta.

**14. Sta je horizontalno--vertikalna (criss-cross) sema skladistenja podataka, i koja je njena prednost u odnosu na dvostruko skladistenje (ping-pong)?**

Osnovna karakteristika Criss-Cross šeme skladištenja odbiraka jeste velika ušteda memorijskog prostora. Jedan od primera koji ovo pokazuje je uporedivanje Ping-Pong i Criss-Cross metoda za obradu podataka u audio dekoderima, koji dekodovanjem minimalne jedinice generišu N=512 odbiraka po kanalu. U ovom slucaju Ping-Pong buffer ce za jedan kanal zahtevati 2x512 memorijskih lokacija.

**15. Čemu služi Block rutina , a cemu Frame rutina?**

Izvršavanje Block i Frame rutine je upravljano ulaznim tokom podataka pri cemu se block rutina izvršava pri prijemu svakih 16 PCM odbiraka u U/I nizu, dok se frame rutina izvršava na svakih N blokova. Ovaj broj blokova zavisi od dekoderskog modula u sistemu, odnosno od njegove jedinice dekodovanja (decoding frame).

**16. Koja je prednost hardverske petlje u odnosu na softwersku?**

Prednost hardverskih petlji jeste ušteda procesorskih ciklusa koji se troše na instrukcije za uvecanje indeksa, proveru uslova i instrukcije skoka, brzina.

**17. Objasniti koncept deljene aritmeticko logicke jedinice?**

Povecanje performanse u putu podataka je moguce i deljenjem ALJ koncepta koji se odnosi na paralelizam. Ovaj koncept, koji se naziva i paralelizam podreci, obuhvata obradu nekoliko elemenata podataka manje tacnosti na jednoj ALJ vece duzine reci, paralelno.

(Uz male modifikacije fizicke arhitekture moguce je realizovati operaciju sabiranja u jednom taktu.)

**18. Navedi tri adresna rezima koja su podrzana od strane Cirrus C49x DSP procesora?**

a) Linearni adresiranje,

b) Moduo adresiranje,

c) Bit inverzno adresiranje.

**19. Navesti cetiri rezima adresiranja :**

pretpostavljeno adresiranje, direktni podaci neposredna adresa, memorijsko – direktno adresiranje,

registarsko – direktno adresiranje, registarsko – indirektno adresiranje, registarsko – indirektno adresiranje sa indeksacijom, registarsko – indirektno adresiranje sa modulo adresnom aritmetikom, registarsko – indirektno adresiranje sa bit obrnutim redosledom

**20. Navesti četiri konfiguraciona parametra kod DMA:**

Startna memorijska adresa

Duzina podataka za prenos

Smer prenosa

Odredisna ili polazna periferna jedinica

**21. Navesti cetiri znacenja registra upravljanja:**

zero (ALJ nula), carry(ALJ prenos), overflow(ALJ prekoracenje opsega), negative(ALJ negativna vrednost), extension (prosirenje)

**22. Standardne DSP biblioteke algoritama?**

Standardna DSP biblioteka

Biblioteka za obradu slike

Biblioteka za podrsku u integrisanom kolu

**23. Navedi bar 4 znacenja koja mogu imati biti registra stanja.**

Carry (C), Zero (Z), Overflow(OV), Negative (N), M Carry (MC), M Overflow (MO)

**24. Navesti barem tri nacina za proveru ispravnosti rada tipicne DSP aplikacije?**

Slusni testovi, poredjenje datoteka na nivou bita, poredjenje spektralnih slika.

**25. Sta znaci kada za neki procesor kazemo daima protocnu strukturu?**

Protocnost je tehnika uvecanja performansi CPU-a deljenjem niza operacija na manje i izvrsavanje u paraleli ako je to moguce.

**26. Objasni kako funkcionise i gde se primenjuje registarsko indirektno adresiranje sa moduo adresnom aritmetikom?**

Adresiranje po modulu je veoma efikasno kod kruznog bafera. Koristi FIFO strukturu, i ima dva pokazivaca, pokazivac citanja ukazuje na memorijsku lokaciju koja sadrzi sledecu vrednost podatka koji ce biti ucitan. Pokazivac upisivanja koji sadrzi memorijsku lokaciju sledeceg pristigle vrednosti koja ce biti upisana.

**27. Objasni kako funkcionise i gde se primenjuje adresiranje sa bit obrnutim redosledom?**

Koristi se u specijalnim okolnostima. Ako je izlazna vrednost iz binarnog brojaca napisana u obrnutom redosledu, rezultirajuca sekvenca izlaza vrednosti brojaca poklapa se sa skremlonovanim nizom FFT izlaza.

**28. Navedi prednosti i mane koriscenja asemblerskog jezika za programiranje DSP procesora u odnosu na jezike viseg nivoa?**

Prednosti: prostiji od samih kompajlera visih jezika, 1-1 preslikavanje, najblizi masinskom jeziku

Mane: nije portabilan, zavisi od arhitekture I malo teza sintaksa

**29. Kako je realizovana sprega modula i operativnog sistema u okviru Cirrus Logic radnog**

**okruzenja?**

Uz pomoc MIF tabela.

**30. Nacrtaj blok dijagram generickog audio kodera sa gubicima.**

**31. Dat je primer procesora kod koga je velicina registra za podatke 8 bita i format u nepokretnom zarezu s.7, a velicina akumulatorskog registra 24 bita, a preciznost s8.15. Prilikom prenosa vrednosti iz akumulatorskog registra u registar za podatke, podaci prolaze kroz SRS jedinicu. Sadrzaj akumulatorskog registra a0 je: 00000000 00101011 01000000**

a) Zapisati vrednost registra a0 u decimalnom formatu: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

b) Aktivan rezim SRS jedinice je “Pomeraj u desno za 1” i “Konvergentno zaokruzivanje”. Kolika ce biti izlazna vrednost te SRS jedinice ukoliko se na ulaz dovede a0? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

c) Aktivan rezim SRS jedinice je “Pomeraj u levo za 1” i “Aritmeticko zaokruzivanje”. Kolika ce biti izlazna jedinica vrednost iz SRS jedinice ukoliko se na ulaz dovede a0? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

d) Aktivan rezim SRS jedinice je “Pomeraj u levo za 2” i “Zaokruzivanje ka nuli”,. Kolika ce biti izlazna vrednost SRS jedinice ukoliko se na ulaz dovede a0? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Koje informacije o digitalnom audio signalu su kljucne radi njegove reprodukcije?

Vremenska duzina audio signala

**Bitska dubina kvantizacije**

**Frekvencija odabiranja**

**Broj audio kanala**

2. Pretpostaviti sta bi se dogodilo ako bi u (softverskom) audio plejeru pokusali da pustimo .pcm fajl

Reprodukovao bi se .pcm ali sa velikim nivoom distorzije

**Ne bi se dogodila reprodukcija signala jer .pcm fajl ne sadrži osnovne informacije o signalu potrebne za reprodukciju.**

Normalno bi reprodukovao taj .pcm audio fajl.

3. Stereo audio fajl sadrži koliko kanala? **2**

4. Povezati tip CLIDE projekta sa njegovom namenom.

Standalone -> Razvijanje DSP algoritma/modula izolovano od Cirrus OS-a

DSP application -> Integracija vise DSP modula i funkcionalnih jedinica unutar Cirrus OS

5. Povezati mem. elemenete arhitekture CS497xx sa odgovarajućim opsezima koje ti elementi podržavaju.

Memorija [-1, 1)

Data registri [-1, 1)

Akumulatorski registri [-256, 256)

6. Povezati mem. elemente arhitekture CS497xx sa odgovarajućim fixed-point fomatima koje ti elementi podržavaju.

Memorija -> s1.31

Data registri -> s1.31

Akumulatorski registri -> s9.63

7. Koja od sledećih linija koda će se uspešno prevesti za arhitekturu CS497xx?

a0 = x0 + y0

a0 = b0 + y0

a0 = a0 + 1

**a0 = a0 + b0**

8. Koja direktiva se koristi za zauzimanje memorijskih lokaciju na CS497?

.prc .wav .brm **.bss** .pcm

9. Koja hardverska jedinica arh. CS497xx se implicitno aktivira sledećom linijom koda:

b0 = ymem[i3]

ALUAGU **SRS** MAC

10. Koji izraz je validan kao uslov za if?

b0 == 0 a0 == 1 **a != 0** b0 > b0 **b > 0** a0l > b0 **a > 0**

11. Označiti pravilno definisane labele.

**%ime\_labele**

1meLabele

**X\_S\_mojaLabela**

ispred\_labele\_ima\_par\_spejsova

**ime\_labele**

**nijePravilnoDefinisanaLabela**

12. Povezati "karakteristike" sistema sa odgovarajućim tipom obrade: standalone (non real-time) i real-time.

Akcenat je na verifikaciji rada DSP algoritma -> Standalone

Izvršavanje DSP algoritma može trajati beskonačno dugo -> Standalone

Izvršavanje DSP algoritma mora biti vremenski usklađeno -> Real-time

Semplovi se direktno preuzimaju iz i pišu u (wav) fajlove -> Standalone

Semplovi pristižu u ekvidistantnim momentima od spoljnjeg uređaja -> Real-time

DSP algoritam je "podređen" spoljnim okolnostima -> Real-time

DSP algoritam je "nadređen" u odnosu na okolinu -> Standalone

13. Povezati spoljnji događaj sa promenom u real-time DSP sistemu koju uzrokuje.

N pristiglih semplova/podataka u sistem -> Block/Brick prekid

Zavrsena inicijalizacija sistema -> Post-kick start prekid

Ažuriranje/promena korisničkih kontrola -> Background nit

14. Kako se ostvaruje sprega sa IO baferima u kojima se nalaze podaci DSP sistema spremni za obradu?

**OS pokazivačima \_\_X\_BY\_IOBUFFER\_PTRS**

OS rutinom Post-kick start

OS pokazivacem BRICK\_SIZE

MCV kontrolama

15. Kako se dodaju MCV/korisničke kontrole za modul koji razvijamo?

Kroz DSP pipeline

Nije moguce dodati korisnicke kontrole u modul

Rucnim pravljenjem MCV strukture

**Kroz graficki uredjivac MCV tabele**

16. Povezati referentni Model 0 razvoja DSP aplikacija sa njegovom namenom.

Generisanje referentnih audio izlaza

Razvoj DSP algoritma nezavisno od ciljne platforme

Jednostavno debagovanje i skaliranje DSP algoritma

17. Povezati Model 1 razvoja DSP aplikacija sa njegovom namenom.

Izmene prema hardverskim prosirenjima ciljne DSP platforme

18. Zbog čega se smanjuje broj argumenata i lokalnih promenljivih u funkciji obrade u modelu 1?

Da bi smestili vise podataka na stek i registre ciljne DSP platforme

**Da bi se rasteretili stek i registri ciljne DSP platforme**

Memorija ciljne platforme ne podzava vise od 4 argumenta za funkciju

19. U kojoj od sledećih C linija koda je uspešno zamenjeno indeksiranje sa pristupom elementu preko pokazivača?

**\*pInBuffer++ = x \* y;**

 pInBuffer[i] = x \* y; i++;

\*(pInBuffer+i) = x \* y; i++;

\*\*(pInBuffer + i) = x \* y; i++;

 20. Povezati C++ emulatorske klase sa odgovarajućim formatima DSP CS497xx arhitekture.

long\_accum -> akumulatorski registri

fract -> X i Y memorija i registri za podatke

long\_fract -> L memorija

21. Ako koeficijent pojačanja iznosi 4.6777, sa kojom vrednošću je potrebno skalirati ga ako koeficijent smeštamo u memoriju CS497xx arhitektue?

8

22. Ako je koeficijent u memoriji skaliran sa 4, prilikom izvršavanja ALU operacija moguće je dobiti traženu originalnu vrednost. Šta je potrebno uraditi u akumulatorskom domenu u tom slučaju?

Re-skalirati vrednost pomeranjem u levo za 2 bita

 23. U modelu 1 vršimo sledeću MAC operaciju:

\*pOut++ += \*pCoef++ \* \*pIn++,

pri čemu je vrednost na pIn adresi 0.0867, dok je vrednost na pCoef adresi 1.1112, izračunata vrednost na pOut adresi biće 0.09634104 što nam spada u dozvoljeni opseg naše memorije.

pCoef pokazuje na memoriju za podatke i očigledno sadrži vrednost koja izlazi iz opsega, ali finalna vrednost staje u opseg. Šta je potrebno uraditi da bi se na kraju ipak dobila odgovarajuća vrednost?

skalirati vrednost na pCoef adresi sa 2, a nakon MAC opeacije dodati re-skaliranje za isti taj pomeraj (1 bit ulevo)

 24. Povezati modele 1 i 2 sa razlikama u bitima dozvoljenim u odnosu na izlaze modela 0.

Model 1 -> 0 bita

Model 2 -> 1-2 bita

25. Povezati oznaku konfiguracije sa odgovarajućom vrstom kanala koju predstavlja.

2\_0\_0 -> Stereo (L i R)

2\_2\_0 -> L, R, Ls, Rs

1\_2\_0 -> C, Ls, Rs

3\_2\_1 -> L, R, C, Ls, Rs, LFE

3\_0\_1 -> L, R, C, LFE

26. U slučaju da je po specifikaciji DSP modula potrebno implementirati promenljivi gain koeficijent (koji je moguće menjati u realnom vremenu), koje sve vrednosti gain koeficijenta je potrebno verifikovati u modelima 0, 1 i 2?

Difolt vrednosti i granične vrednosti koeficijenata

27. Koji od sledećih registara su slobodni za korišćenje unutar ASM pozvane funkcije, ako je pozivamo iz C-a?

a0 a1 b0 b1 i0 i1 i4 i5

28. Povezati C imena funkcija i promenljivih sa odgovarajućim ASM imenima na koje ih CCC2 prevodilac mapira.

myProcFoo() -> \_myProcFoo

myVariable -> \_myVariable

thisIsASMSymbol -> \_thisIsASMSymbol

29. U modelu 3 potrebne su nam C++ emulacione klase za fract i accum tipove koje se koriste u modelu 2 (fixed\_point\_math, stdfix\_emu).

False

30. Ako su funkcija ili promenljiva definisani u ASM fajlu a koristimo ih u C fajlu, da bi bili vidljivi C domenu, potrebno je dodati koji kvalifikator u ispred svake promenljive/funkcije (u ASM fajlu)?

.public

31. Ako su funkcija ili promenljiva definisani u Cfajlu a koristimo ih u ASM fajlu, da bi bili vidljivi ASM domenu, potrebno je dodati koji kvalifikator ispred svake promenljive/funkcije (u ASM fajlu)?

.extern

32. Povezati MCV strukture koje imamo u finalnom modelu sa odgovarajućom namenom.

example\_module\_HOST\_MCV -> Os struktura u kojoj se direktno mapiraju kontrolne vrednosti izmenjene u realnom vremenu

example\_module\_MCV -> Lokalna struktura DSP modula koja sluzi kao kopija poslednjih kontrolnih vredosti izmenjenih u realnom vremenu

33. Povezati OS rutinu sa namenom.

Brick -> obrada U/I podataka (PCM semplova)

Background -> azuriranje MCV vrednosti

PostKickStart -> inicijalizacija modula

34. Da bi u processing fajlu bila vidljiva MCV struktura koja služi kao kopija originalnih MCV vrednosti, kao i da bi uspešno preuzimali MCV vrednosti iz nje, potrebno je:

dodati deklaraciju te MCV strukture (sa .extern)

uključiti generisani mcv header sa MCV\_T definicijom

podesiti lokalni pokazivač na odgovarajuća polja te MCV strukture

35. Ako se dogodi da redosled kanala u OS baferima nije raspoređen prema implementaciji obrade koja je verifikovana u Modelu 3, kako je moguće rešiti taj problem?

Izmenom redosleda kanala u Input Remapu DSP aplikacije

1. Jedna od metodi za potencijalnu optimizaciju brzine CS48x aplikacije napisane u C kodu podrazumeva smanjivanje broja argumenata kod funkcija obrade. Zbog cega?

Zato sto time smanjujemo broj promenljivih koje ce biti smestene u registre procesora i na sam stek, cime uticemo na broj instrukcija (smestanje na stek iziskuje vise instrukcija)

2. Odaberi tacne odgovore

I2C sprega se koristi za povezivanje DSP i FLASH memorije

Prednost I2C sprege je velika brzina komunikacije, pa se koristi za prenos nekomprimovanih audio podataka

**Prednost I2C sprege je jednostavna implementacija na fizickom nivou**

3. Odredi interfejs za povezivanje komponenti

DSP - HDMI Rx -> I2S

DSP - uC -> I2C

DSP - DAC -> I2S

DSP - FLASH -> SPI

4. Pod sekundarnim (dodatnim) podacima kod Super Harvard arhitekture podrazumevano podatke koji se nalaze u okviru

Programske memorije

5. Koji elementi ne spadaju u centralno jezgro digitalnog signal procesora

ALU

Dekoder instrukcija

**Kolo za generisanje takta**

Registar upravljanja rezimom rada procesora

**Serijski interfejs**

6. Registri opste namene:

podrzavaju prosirenu preciznost pri obradi podataka

sluze za cuvanje medjurezultata obrade

**sluze kao ulazi u MAC**

**se organizuju u parove skupova (X i Y)**

imaju duplo duzu sirinu nego memorijska rec procesora

7. Povezi rutinu sa uzrokom njenog pozivanja od strane sistema/kernela arhitekture CS48x

Post-Kick start -> zavrsetak inicijalizacione faze

Timer -> Periodicno pozivanje na svakih N milisekundi

Background -> Periodicno izvrsavanje kada se nista ne poziva od FG rutina

Block, Brick -> Prijem 16 ulaznih podataka

Frame -> prijem 256 ulaznih podataka

Pre-Kick start -> Pocetna inicijalizacija modula (pre svega MCV elemenata)

8. Pristup sistemskim ulazno-izlaznim podacima (samim odbircima) moguce je izvrsiti preko kojih simbola operativnog sistema za CS48x?

\_\_X\_BY\_IOBUFFER\_PTRS

9. Sta ce se desiti nakon pokretanja sl. koda u modelu 3 projektta za arhitekturu CS48x?

\_\_memX int data2 = 2;

int main(void)

{

\_\_memY int data1;

\_\_memY int\* localP1 = &data1;

int\* localP2 = &data2;

return \*localP1 + \*localP2;

}

Kod se uopste ni nece prevesti - CCC2 kompajler prijavice gresku zbog zadavanja kvalifikatora memorijske zone prilikom definisanja lokalne promenljive data1

10. Povezi zastavice sa odgovarajucim registrima:

Maske prekida -> Registar upravljanja rezimom rada procesora

Mnozac prenos (carry) -> Registar stanja

Prekoracenje opsega -> Registar stanja

ALU nula (zero) -> Registar stanja

Rezim redosleda bajta -> Registar upravljanja rezimom rada procesora

Numericki rezim rada -> Registar upravljanja rezimom rada procesora

11. Koliko instrukcijskih ciklusa se potrosi na izvrsavanje sl. koda (racunati da “do” instrukcija trosi 2 ciklusa)?

31

12. Povezi elemente sa odgovarajucim tokom

Pomerac -> Tok obrade podataka

Dekoder instrukcija -> Tok obrade instrukcija

Jedinica za generisanje adresa podataka -> Tok obrade podataka

Programska memorija -> Tok obrade instrukcija

Sekvencer instrukcija -> Tok obrade instrukcija

Memorija podataka -> Tok obrade podataka

Instrukcioni registar -> Tok obrade instrukcija

Aritmeticko logicka jedinica -> Tok obrade podataka

13. Koliko reci programske memorije je potrebno za smestanje sl. koda?

12

14. Izaberi sve tacne odgovore

**Digitalni sistemi su kompleksniji, ali se podaci lako skladiste**

Analogni sistemi obezbedjuju vecu sigurnost i korekciju greske signala pri prenosu

Analogni sistemi su kompleksniji od digitalnih sistema

Analogni signali se lako skladiste i jednostavno obradjuju

**Digitalni sistemi omogucavaju ponovljivost i predikciju ponasanja sistema**

Digitalni sistemi koriste manji propusni opseg od analognih sistema

**Analogni sistemi su osetljivi na tolerancije komponenata, za razliku od digitalnih**

15. Velicina bajta na arhitekturi CS48x je?

32b

16. Povezati tip registara sa odgovarajucim podrzanim decimalnim opsegom

Registar opste namene (data registar) -> [-1, 1)

Akumulatorski registar -> [-256, 256)

17. SPDIF interfejs je namenjen prenosu

PCM odbiraka i komprimovanog bitskog toka

18. Zaokruzi tacne odgovore

**U tabeli vektora prekida su smesteni pokazivac na rutine za obradu prekida, ili kratke sekvence koda za obradu prekida**

Tabela vektora prekida se nalazi na pocetku memorije podataka

Vektori prekida su pridruzeni fizickim prekidima, dok su softverski prekidi definisani u okviru glavnog programa

19. Povezati sistemske simbole CS48x operativnog sistema sa njihovom namenom

\_\_X\_VY\_BLOCK\_SIZE -> Definisanje broja odbiraka po kanalu u okviru jednog bloka

\_\_X\_BY\_IOBUFFER\_PRTS -> Pristup sistemskim ulazno-izlaznim podacima

\_\_X\_VX\_PPM\_VALID\_CHANNELS -> Kontrola aktivnih kanala na izlazu iz prethodnog modula u lancu izvrsenja

\_\_X\_VX\_PPM\_OUTPUT\_CHANNELS -> Kontrola aktivnih audio kanala na izlazu iz citavog sistema