**ARH 1 – 3. Blic**

**ZZV**

**1. Smjer FRISC-ovih priključaka DATA i ADR je :**DATA su dvosmjerni, ADR su izlazni

**2. Izaberite netočnu tvrdnju u nizu rečenica vezanih za "open collector" priključke (tzv. spojeni-I) :**Procesor kojemu je ovaj vod ulazni, mora na vodu držati neaktivno stanje

**3. Za sabirnice vrijedi :**Sabirnica je spojni put koji (električki) povezuje dijelove računala

**4. Koja od navedenih vrsta sabirnice ne pripada među ostale tri :** Dvosmjena

**5. Za sinkrone sabirnice točne su tvrdnje :** Imaju veliku brzinu rada i malu duljinu**;** Prilagođene su za slučaj kad svi uređaji imaju jednaku brzinu**;** Brzina rada ipak im se može prilagoditi

**6. Kod FRISC-ove sabirničke komunikacije WAIT uzrokuje :**Ubacivanje jednog ili više stanja čekanja u cikluse čitanja i pisanja

**7. Redoslijed (nepotpun) kod čitanja iz memorije je :** Nakon čitanja podatka FRISC deaktivira READ; zatim mem. postavi DATA u visoku impedanciju

**8. Odredite koliko ciklusa treba FRISC za izvođenje niza sljedećih naredaba:**

**LOAD, CALL, ADD, POP, MOVE, SUB, CMP, JP, PUSH :** 14

**9. Prednost uvjetnog prijenosa u odnosu na bezuvjetni :** Postoji sinkronizacija procesora s vanjskom jedinicom

**10. Koje su namjene linija za rukovanje u uvjetnoj/prekidnoj VJ :** READY-jem VJ javlja spremnost uređaju, STROBE-om uređaj javlja spremnost v.jedinici

**11. Koja je namjena podatkovnog registra u ulaznoj uvjetnoj VJ :**U njemu se pamti podatak koji VJ primi od vanjskog uređaja/procesa

**12. Koja je tvrdnja istinita za prekide :**VJ može postaviti prekid u bilo kojem trenutku izvođenja glavnog programa

**13. Zašto procesor automatski zabranjuje daljnje prekide kad prihvati zahtjev za prekidom** : Jer bi inače odmah u prekidnom potprogramu došlo do novog prekida od iste VJ

**14. Raspoznavanje koja VJ je izazvala prekid i njeno posluživanje ne može se napraviti :** Programski, čekanjem na spremnost najprioritetnije VJ u prekidnom potprogramu

**15. Što ne treba napraviti u maskirajućem prekidnom potprogramu** : Spremiti registar PC

**16. Za nastavak komunikacije prekdinih i uvjetnih VJ sa vanjskim uređajem vrijedi :** Prekidna nastavlja komunikaciju nakon dojave kraja posluživanja, a uvjetna nakon brisanja bistabila stanja

**17. Što od navedenog nije istina za FRISC-CT :** Nakon što CT postane spreman, sigurno će postaviti zahtjev za prekid

**18. Kod sklopa FRISC-CT, impuls doveden na priključak CNT će** : smanjivati brojilo, ako je tako zadano u upravljačkom registru

**19. Sklop PIO povezuje se s vanjskim procesom pomoću priključaka :** PIOD0-PIOD7, READY i STROBE

**20. U kojem načinu rada se sklopu PIO sigurno ne mora nikada javljati prihvat prekida niti kraj** posluživanja : Načinu postavljanja bitova

**21. Kako bi mogli programski ustanoviti da je FRISC-DMA na adresi PA obavila prijenos :** Ako sa adrese PA+8 pročitamo 0

**22. Izbacite neistinitu tvrdnju za DMA :** Kod blokovskog prijenosa ne koriste se BREQ i BACK

**23. Imamo memorijski blok veličine 2000 podataka na adresi 5000. Prvi podatak iz tog bloka želimo** **kopirati u sve preostale lokacije tog istog bloka pomoću DMA prijenosa. Sklop FRISC-DMA treba** **inicijalizirati upisima sljedećih podataka na navedene adrese** : PA: 5000, PA+4: 5004, PA+8: 1FFF, PA+12: %B0100;;;ono što je zadano na bitovima 2 i 3 tj destination i source ne mora odgovarati onom što je zaista spojeno na DMA

**24. Ako je na PIO na početnu adresu poslan prvo podatak %B 110101, a zatim %B 11110000, onda će** **PIO raditi** : Radit će u izlaznom načinu uz korištenje READY i STROBE i bez postavljanja prekida. %B 11110000 0.bit – 0 ---OCR 1.bit 0 – ne postavlja prekid 3.bit – 0 –izlazni način da ne postavlja prekid, mora koristiit redy i strobe

**25. Želimo prebrajati milijun impulsa korištenjem dva sklopa CT: CT1 i CT2. CT2 nakon milijun** **impulsa treba generirati prekid. CT1 i CT2 spajaju se ovako :** Zajedno se spaja priključak ZC od CT1 i priključak CNT od CT2

**26. BREQ ima sljedeću namjenu :** DMA-jedinica zahtjeva upravljanje nad sabirnicom

**27. BACK ima sljedeću namjenu** : Procesor potvrđuje DMA-jedinici da joj je prepustio upravljanje nad sabirnicom

**28. DMA prijenos kod kojeg se prenosi jedan podatak, nakon čega se sabirnica vraća procesoru zove se** : Krađa ciklusa

**29. Razlika između BREQ i BACK je :** BREQ ulazi u procesor, a BACK izlazi iz procesora

**3.Blic 2010/2011**

**1.Treba prenijeti podatke iz memorije u VJ, tako da se kraj prijenosa detektira pomoću prekida. Upravljačka riječ za FRISC – DMA može biti :** 1001(2);10\_1 – 10 –dest/src, a 1 - prekid

**2.Logička operacija NE na 6 bitnom broju 101100(2) daje rezultat:** 010011(2)

**3.Za sklop FRISC – PIO točne su tvrdnje :** Načini ispitivanja i postavljanja bitova ne koriste priključke za rukovanje; Samo u načinu ispitivanja bitova koristi se maska MQ ;v Ulazni i izlazni načini koriste priključke za rukovanje

**4.Programski UI prijenos djeli se na :** uvjetni, bezuvjetni, prekidni

**5.Glavni nedostatak bezuvjetnog prijenosa je:** Mogućnost gubitaka ili uvišestručenja podataka

**5.1. Glavni nedostatak najbržeg prijenosa je:** Sporost programa za UI; Sporost zbog sinkronizacije s vanjskom jedinicom

**5.2. Glavni nedostatak najjednostavnijeg prijenosa je:** najveća složenost vanjeske jedinice

**6. U jedinici DMA FRISC, adresni registar odredišta se povećava za 4:** Ako se u upravljačku riječ u bit 3 upiše 0

**7.Kod opće prekidne VJ vrijedi:** Jedna od lokacija služi za dozvoljavanje/zabranjivanje postavljanja prekida

**8. Kod sklupa FRISC-CT, za priključke ZC i CNT vrijedi:** ZC je izlazni, CNT je ulazni

**9.Za FRISC-ove priključke INT0-INT2 vrijedi:** Mogu im se odrediti međusobni prioriteti, ali samo programski

**10. Odaberite TOČNE tvrdnje:** WAIT ima ulazni smjer; VCC i GND su potrebni za rad procesora; RESET ima ulazni smjer; READ ima izlazni smjer

**11. Za rekurzivne potprograme TOČNA je tvrdnja**: Parametre prenosimo stogom

**12. Želimo svake sekunde generirati prekid pomoću FRISC \_ CT-a. Na ulaz CNT spojen je signal frekvencije 1 kHz. CT je na adresi PA. CT-u treba poslati:** broje 1000(10){1 kHZ=1000Hz} na adresu PA, te broj 11(2) na adresu PA+4

**13.Na uvjetnoj VJ, za sinkronizaciju s vanjskim uređajem postoje:** Jedan ulazni i jedan izlazni priključak redy i strobe

**14.Koja tvrdnja vijedi za UI sabirnicu:** Može imati duljinu veću od memorijske sabirnice; Ima manju brzinu rada nego memorijska sabirnica; Može spajati veći broj UI jedinica s procesorom; Povezivanje uI jedinice s CPU

**15.Što s podatkom R0, radi programski odsječak ##OR R0, %B 0011, R0 ;; AND R0, 0FF00, R0## :** postavlja 2 najniža bita, pa briše sve osim drugog bajta.

**Koliko koja vrsta naredba traje ciklusa:** 1. Ciklus AL, REG;; 2.Ciklusa MEM, UPR, STORE, LOAD, PUSH, POP, JR, JP, RET, CALL, HALT

**PRIKLJUČCI**-mogu biti aktivni u niskoj razini, tada imaju potez; Aktivni u visokoj razini nemaju potez; Mogu označavati jedno od dva moguća stanja; **Open collector**-više izlaznih priključaka spaja se na jednu sabirnicu i svi njome upravljaju; Ako su svi neaktivni, cijela sabirnica je neaktivna. Ako se jedan aktivira, aktivira se cijela sabirnica; Koriste se za spajanje prekidnih priključaka vanjskih jedinica na jednu sabirnicu. **SABIRNICE**-Prednosti: mala cijena, prilagodljivost prilikom projektiranja i nadogradnje racunala, standardiziranost; Nedostaci: mala propusnost, ogranicena duljina sabirnice, ogranicen broj uredaja koji se mogu spojiti na jednu sabirnicu, problemi zbog uredaja razlicite brzine; **Sinkrone:** sve operacije su sinkronizirane s taktom sustava (tj. clockom), jednostavne su za implementaciju, imaju veliku brzinu rada pa zato i malu duljinu, bolje su prilagodene za slucaj kad svi uredaji imaju jednaku brzinu, imaju mogucnost prilagodbe brzine rada, ali se komunikacija vecinom odvija predvidenom brzinom, češće se koriste za memorijske sabirnice; **Asinkrone:** ne koriste CLOCK za sinkronizaciju, uredaji se sinkroniziraju tzv. rukovanjem (engl. Handshaking protocol), rukovanje je postupak u kojem strane koje komuniciraju prelaze na sljedeci korak komunikacije tek kad obje strane potvrde da je prethodni korak dovršen, složenije su za implementaciju, imaju manju brzinu rada, mogu imati veliku duljinu, bolje su prilagodene za slucaj kad uredaji imaju razlicite brzine, češće se koriste za U-I sabirnice; **PIPELINE**-protočno izvođenje u N razina je za N puta brže od izvođenja korak po korak;FRISC-2 razine-dohvat i izvođenje; Dohvat:dohvat naredbe, dekodiranje naredbe, dohvat operanda; Izvođenje: izvođenje AL operacije, spremanje rezultata; JEDNOCIKLUSNE NAREDBE-AL naredbe, registarske naredbe; DVOCIKLUSNE-upravljačke, memorijske; **HAZARDI**-strukturni, upravljački, podatkovni; Strukturni-procesor ne može izvesti sve faze odjednom- kod FRISC-a-sve memorijske naredbe ga uzrokuju; Upravljački hazard-uzrokuju ga upravljačke naredbe, skače se na neku drugu adresu; **IO**-dva načina komunikacije procesora i UI jedinica: memorijsko UI preslikavanje i izdvojeno UI adresiranje; **Vrste prijenosa podataka:** Programski prijenos i Sklopovski prijenos; Programski:-manja brzina. Vrste: bezuvjetni, uvjetni, prekidni; Sklopovski: veća brzina, izravni pristup memoriji(DMA); **Bezuvjetni:**-najbrži, ne provjerava se jesu li sinkronizirani procesor i VJ – nedostatak; **Uvjetni:** provjerava se je li VJ spremna za prijenos podataka; Redoslijed komunikacije: 0)reg prazan, bist. Stanja=0, Ready=1 – VJ javlja spremnost uređaju; 1)šalje se podatak u VJ, 2) strobe=1 – uređaj javlja VJ da je poslao podatak; 3) b.stanja=1 – VJ spremna za komunikaciju s procesorom, ready=0; 4)strobe=0; 5)procesor čita podatak iz registra VJ; 6) briše se b.stanja, ready=1;**Prekidni**-VJ samostalno dojavljuje procesoru svoju spremnost zahtjevom za prekid; Brži je od uvjetnog, ali sporiji od bezuvjetnog; **PREKIDI**-maskirajući: INT0, INT1, INT2; nemaskirajući: INT3; IACK-potvrda prihvata nemaskirajućeg prekida; **Prihvat nemaskirajućeg:** Aktivira se prikljucak IACK (VJ treba deaktivirati INT3), briše se IIF (zabranjivanje svih daljnjih prekida), Sprema se PC na stog, Deaktivira se IACK, Skok u prekidni potprogram na adresi 12(10); **Prihvat maskirajućeg:** Briše se GIE(zabranjuju se daljnji maskirajući prekidi), sprema se PC na stog, dohvat adrese prekidnog potprograma s memorijske lokacije na adresi 8; RETI- postavlja GIE=1, RETN-postavlja IIF=1;**FRISC-CT-**za brojanje impulsa i mjerenje vremena; CR-upravljački registar kojim se može omogućiti ili onemogućiti postavljanje zahtjeva za prekid, omogućiti ili zaustaviti rad CT-a; LR (Limit Register) je 16-bitni registar granice ili registar vremenske konstante (vrijednost mu se zadaje programski); DC (down counter) je 16-bitno brojilo na dolje (vrijednost upisana u LR automatski se upiše i u DC); Preko CNT-a dolaze impulse koje CT prebraja; Kad CT dođe do nule, generira se impuls na izlazu ZC; Vrijednost iz LR puni se u DC I nastavlja se s brojenjem; Postavlja se stanje spremnosti a prekid se generira ako je omogućen pomoću CR; **FRISC- PIO**-sklop za paralelni prijenos podataka; 4 načina rada: ulazni, ispitivanje bitova, izlazni, postavljanje bitova; ICR-ulazni uprevljački registar, OCR-izlazni; MR-registar maske, svaki bit u 8-bitnom registru maske odgovara jednom PIOD-u, ako je u MR upisana 0, PIOD se ne ispituje, ako je 1, ispituje se; **DMA PRIJENOS**-izravni pristup memoriji; Vrste- zaustavljanje procesora, krađa ciklusa, blokovski, multiplaksirani; DMA zaustavljanjem procesora odvija se ovako:DMA-sklop preuzme upravljanje nad sabirnicom, DMA-sklop prenese sve podatke, Tek tada upravljanje nad sabirnicom se vraća procesoru koji je cijelo vrijeme bio zaustavljen. Dok DMA-sklop upravlja sabirnicom, procesor ne može dohvaćati naredbe iz memorije i potpuno je neaktivan (jedino čeka dojavu od DMA da može nastaviti s radom), Nedostatak: procesor može dulje vrijeme biti neaktivan, Prednost: najbrži prijenos; DMA krađom ciklusa odvija se ovako: DMA-sklop preuzme upravljanje nad sabirnicom, DMA-sklop prenese jedan podatak, Upravljanje nad sabirnicom se vraća procesoru, Gornja tri koraka se ponavljaju dok se ne prenesu svi podatci, Prednost: procesor se usporava, ali ipak izvodi glavni program, Nedostatak: sporije od zaustavljanja procesora; DMA blokovskim prijenosom odvija se ovako: DMA-sklop preuzme upravljanje nad sabirnicom, DMA-sklop prenese nekoliko podataka (tj. blok), Upravljanje nad sabirnicom se vraća procesoru, Gornja tri koraka se ponavljaju dok se ne prenesu svi podatci; Multipleksirani- koriste se trenutci kad procesor ne upravlja sabirnicom;