Pitanja iz VLSI

200MHz

1. Navesti faze u dizajnu VLSI cipa. U kojoj fazi nastaju razlike kod razlicitih logic symbol dizajn metoda
    - Schematic capture – unosenje logicke seme
    - Logic and timing testing
    - Placement and routing logickih elemenata
    - Tek u Placement And Routing jer slozenost prve dve zavisi od slozenosti seme a ne od primenjen metode projektovanja
2. Navesti 5 nacina gate level design unosenja u kompjuter
    - Graphic entry
    - HDL based entry
    - Logic equation entry
    - State machine entry
    - Direct entry of the net-list, using a test editor
3. Navesti dva output file-a MP2D:
    - artwork file for visual analysis
    - fab file salje se u fabriku mailom
    - u fabrici je svaka standard cell zamenjena sa full-custom ekvivalentom
4. navesti 5 fajlova za opis arhitekture u ENDOT paketu
    - isp prim, isp je jezik za opis hardvera
    - t topology- ima jedan red ako postoji samo jedan isp prim fajl, oznacava veze izmedju isp prim fajlova
    - m meta-micro-velik case za korespondenciju asemblerskog i masinskog jezika, najcesce odnos asemblerskog i masinskog 1:1 ali i ne mora jer moze biti i vise mikroprocesora u jednom multiprocesoru
    - i file information related to linking and loading, opis se odnosi na target a izvrsava na hostu
    - jedan ili vise benchmarka za testiranje(ne mora biti ekstenzija .b)
5. dati red velicine odnosa taktova target:host kod ENDOT i VHDLa. Gde je taj odnos bolji i zasto
    - 1:10^4 ENDOT I 1:10^5 VHDL. Bolji je kod endot zato sto je endot manje komplestan
6. objasniti kom softverskom ili hardveskom sloju ide
    - broj ciklusa takta host masine – ide OS
    - broj ciklusa takta target masine – zovem simulatorski paket
7. prednosti i mane endot i vhdl
    - endot radi bolje odnos target:host, ima statisticku analizu
    - vhdl ima konstrukcije za opis hardvera za multiprocesore koje endot nema
    - silicijumski kompajleri su bolji za vhdl nego za isp prim jer nema drndanja sa potpitanjima a i cip je nesto manje povrsine

8. navesti 6 delova deklarativne sekcije isp prim programa
    - macro
    - port
    - state
    - memory
    - format
    - queue
9. objasniti namenu next and delay naredbi u behavioral sekciji
    - sve izmedju begin I next se izvrsava sekvencijalno(inace izvrsenje ide u paraleli)
    - nije dovoljno samo next, treba I prebrojati takt u alatu za statistiku, za to sluzi clock, ili ako hocemo vise taktova delay(arg) gde arg oznacava broj taktova.
    - Samo clock bez next nema smisla stavljati
    - Delay(1) je isto sto i next+clock
10. sta u .isp fajlu znaci kljucna rec lead
    - prednja ivica
11. Posmatra se Fura RISC procesor koji ima 3 stepena Pipeline. Prikazati izgled coding windowa za 4 razlicite pozcije prozora.
    - Slajd 33/57
12. Navesti 6 parametara koji uticu na brzinu izvrsavanja RISC procesora
    - Pipeline fill-in depth
    - Branching related statistics
    - Branch fill-in function
    - Clock slow down function
    - Technology related statistics
    - Cache impact
13. Oznaciti prozor kod Fura RISC. Koliko je taktova potrebno da ceo prozor Fura RISC kompletno udje u odradi sav posao?
    - Dva takta jer prozor obuhvata IF i IDX iz tekuce instrukcije i LD iz prethodne instrukcije. LD i IDX se izvrse za isti period takta.
14. Kako radi LOAD instrukcija kod Fura RISC? Nacrtati semu.
    - Sema je sa slajda 44/57
    - IDX stepen instrukcije i+1
      ```
      If(op==21){
        pastdst=dst;
        pastval=memory[reg[src]]
      }
      ```
    - LD stepen instrukcije i
      ```
      If(pastop==21)
        reg[pastdst]=pastval
      ```
15. Napisati Main program u isp jeziku za Fura RISC(dovoljno je prikazati ponasanje za LOAD instrukciju). Operandi koji su izvoriste su src1 i src2(za LOAD se koristi samo src2), a jedino odrediste je dst. Nije potrebno voditi racuna o hist nizu. Opcode za LOAD je 21.

- slajd 46/57

16. Navesti dva nacina na koje hw interlock resava branch (sequential) hazard
    - alu suspend
    - rwb(register write unit) suspend, instrukcija se dalje cita ali se ne upisuje rezultat nigde
17. Zasto hardverski interlock usporava procesor
    - vise tranzistora, a svaki mora manje da trosi, pa ih usporavamo, inace kuciste moze da izgori
    - mora da stoji u kriticnoj stazi za clock
    - na vecem cipu imamo vece kasnjenje na zicama
18. Osnovna razlika izmedju synthetic jobs i benchmark
    - Synthetic-program ne radi nista smisleno ima proizvoljnu statistiku naredbi
    - Benchmark - semanticki programi

Introduction to VLSI

1. Za logic symbols navesti koji pristup ima min a koji max
    - *fleksibilnost* max=standard cell jer ima 0 slojeva predfabrikovano
    - *pocetna cena* za pl je najmanja jer treba najmanje inzenjerskog posla
    - *cena* najbrze raste po komadu za PL zato sto je losa iskoriscenost povrsine cipa
    - u pocetnu cenu spadaju cena dizajna i cena inzenjerskog vremena za izradu maski posle dizajna
2. Objasniti kako dolazi do razlicitog stepena iskoriscenosti kod logic symbol
    - standard cell ima logiku iste debljine ali nisu ekvidistanti
    - ga i pl se prave sa kucistima standardne velicine
    - ga gubi i po ivici, a logika je ekvidistanta
    - kod pl macrocell zauzimaju samo 25% povrsine
3. Za sta se koristi FC?
    - Nove familije standard cell
    - Prosirenja postojecih familija
    - Strukture koje se mnogo ponavljaju-npr sistolicka polja
    - Enormne kolicije
    - Jednostavan hardver
4. Objasniti ulogu compare.dat za testiranje u LOGSIM paketu
    - upari se sa gen4.dat
    - proveri se da li je sumulation output file istog sadrzaja kao i compare.dat
5. Navesti razliku izmedju prefab i postfab testiranja
    - u postfab mozemo pristupiti samo pinovima
    - u postfab se radi testiranje strukture,a u prefab funkcije pa je postfab skup testova veci
    - u prefab proveravam da li radi ono sto treba (da li sam u net listi zicu dobro povezao na primer), a u postfab gledama da li se nije neka zica u procesu fabrikacije lose realizovala

- neke stvari vezane za elektroniku nema smisla testirati u prefab

6. razlika behavioral i structural level faults
    - behavioral - kao crnu kutiju ga gledam, da li sklop u celini radi funkciju za koju je namenjen
    - structural - gledam da li svaka komponenta u sklopu radi kao i veze izmedju njih
7. mane SSA pristupa
    - ignorise visestruke greske, ali to nam nije od interesa. Ako cip ne valja bacamo ga
    - neke greske nije u stanju da primeti
8. Objasniti SSA Path sensitisation na kombinacionim mrezama.
    a. Potrebno je staviti nesto na ulaz, da bi se dobilo nesto na izlazu(izlaz mora da zavisi od ulaza)
    b. Input bit se postavi na nulu, da proveri SA-1
    c. To propagira do prve tacke gde se moze testirati
    d. Sve se ponovi za 1
9. Objasniti zasto se i kada koristi Automatic Test Vector Generation
    - Sa malo testova velika pokrivenost, kasnije polako napredujemo (kriva)
    - Nikad ne znas kolika je pokrivenost
    - Moguce samo za kombinacione mreze
10. Dat je Scan path za sekvecijalne mreze. Ako na semi postoji 100FFa
    - Kolika je duzina test vektora? 100
    - Koliko ciklusa je potrebno da test vektor udje i izadje? 201(100 za ulazak,1 za generisanje izlaza, 100 za izlazenje)
    - Koliko ciklusa je potrebno za 2 test vektora? 302(100 za ulazak prvog, 1 za generisanje izlaza prvog, 100 za izlaz prvog i ulaz drugog, 1 za generisanje izlaza prvog, 100 za izlaz drugog test vektora)
11. Objasniti sustinu scan path sema
    slajd 59/69
    - Idemo u test mode, * veze postoje, izbacimo veze FFa sa kombinacionom logikom, unesemo test pattern u shift rgistar
    - Idemo u normal mode, izbacimo veze *, ostvarimo veze sa kombinacionom logikom, FF su ulaz za kombinacione mreze, posle 1 kloka upisujemo vrednosti izracunate u kombinacionoj logici u FF
    - Idemo u test mode, bez veze sa kombinacionom logikom, sa * vezama, sadrzaj shift registra se posalje na izlaz
12. Kako radi Stanford SCAN-PATH design-for-testability?
    - Slajd 60 i 61
    - Postavi se T=1(test mode)
    - Ucita se test pattern u Ffove
    - Test vrednosti na Xi ulazima se postave
    - Postavi se T=0(normal mode)
    - Posle vremena dovoljnog da kombinacione mreze izracunaju izlaz odradimo 1 period signala takta
    - Sadrzaj FF se istisne kroz Zm i proveri se dali je to ocekivan rezultat

- Potrebno je testirati i FFove

13. Navesti 6 tipova greske kod RAMa
    - Stack at
    - Decoder failures
    - Multiple writes
    - Slow sense amplifier recovery
    - Sleeping sicknes
    - Pattern sensitivity
14. Objasniti marching 1s
    - Upisu se sve nule
    - Od pocetka do kraja se procita nula i na njihovo mesto upise 1
    - Idemo u kontra smeru – procitamo 1 i upisemo nulu
15. Objasniti checkerboard
    - Upisemo 0101010101.... pa to i procitamo
    - Upisemo 1010101010... pa to i procitamo
16. Objasniti walking patern
    - slajd 67
    - Memorija se inicijalizuje na sve nule
    - Upise se 1 na prvu lokaciju, proveri se da li su sve nule na ostalim lokacijama, proveri se da li 1 jos uvek stoji na datoj lokaciji
    - Vratiti lokaciju sa 1 na 0
    - Ponoviti isto za walking 0
17. Objasniti Galpat I i Galpat II

RISC

1. Objasniti 8 principa RISC procesora
    a. regularity
    b. orthogonality
    c. composability
    d. one versus all
    e. provide primitives not solutions
    f. addresing principle
    g. environment support principle
    h. deviations principle
2. Navesti 3 osnovne razlike cisc i risc procesora
    a. risc obezbedjuje primitive a cisc resenja koja nisu dovoljno generalna
    b. cisc mora ponovo da izracunava podatke jer dvoadresna arhitektura unisti neke podatke
    c. cisc obezbedjuje vise nacina ali ne sve nacine
3. Da li su kraci ciklusi kod RISC ili CISC
    a. Kod RISC su kraci, jer je najduza operacija kraca. Kod RISC najduza je staza za podatke(Datapath – citanje registara, ALU, upis, regustar) a kod CISC najduza je dekoderska logika

4. Kako funkcionise UCB RISC s obzirom da nema stek?
    a. Register windows
5. Kako se kod RISC II UCB postize ubrznaje u odnosu na RISC I:
    a. RISC II ima 3 stepena
    b. RISC I je imao 3-bus register file
    c. U svakom ciklusu takta non stop rade ALU i 2-bus register file. Radi se citanje dva operanda iz registarske memorije i njihova propagicija kroz ALU(drugi stepen), upisivanje rezultata u registar ili memoriju(treci stepen)
6. Objasniti kako su kod SU-MIPS realizovani npr uslovni skokovi ako „No sticky condition codes”
    a. Jedna naredba i proizvede condition code i iskoristi ga– pakovanje vise naredbi u jednu
    b. OD proizvede fleg, a SX ga koristi
7. Kod SU-MIPS, koliko instrukcija uspe bar da udje u pipeline za 9 perioda signala takta i zasto?
    a. 3 instrukcije. Dok prethodna insturkcija u pipeline-u ne zavrsi ID fazu ne sme se dohvatati sledeca instrukcija
    b. slika sa slajda 26/48
    c. ALU jedinici pristupaju i OD i SX stepeni pipeline-a iste instrukcije
    d. Instrukcijskoj memoriji pristupaju stepeni IF i ID(samo 30% ID vremena se korsti za dekodovanje instrukcije, 70% vremena na pocetku se utrosi na fethcing – fetching se radi u IF i prvih 70% vremena ID)
    e. Memoriji za podatke se pristupa iz OF stepena prethodne instrukcije i SX stepena sledece instrukcije
8. Koraci software-imposed pipeline interlock kod SU-MIPS(Reorganizer)
    a. Izmedju asemblerskog i masinskog jezika se nalazi
    b. Promena redosleda
    c. Ako prethodno nije moguce, ubaciti NOOPs na mesta branch delay
    d. Pakovanje ako je moguce (na primer: neka naredba ne koristi OD ili SX stepen – tu je moguce ubaciti neku od aritmetickih ili logickih operacija)
    e. Asembliranje
9. Navesti 3 osnovne vrste hazarda
    a. Timing(Data)
    b. Sequential(Control)
    c. resource
10. Za svaku od Gross-Henessy sema navesti da li se i kada stedi vreme/prostor
    a. #1 – stedi prostor bezuslovno(nop se zamenjuje korisnim instrukcijama), vreme stedi bezuslovno
    b. #2 – prostor se ne stedi, kopiranje koda; vreme se stedi ako dodje do skoka
    c. #3 – prostor se stedi bezuslovno, radi se move; vreme se stedi ako NE dodje do skoka
    d. za sve tri seme se podrazumeva da je generator koda posle skokova ubacio odgovarajuci broj NOPova
11. Da li je bolje uraditi prvo alokaciju registara ili optimizaciju?
    a. Bolje performanse za optimizaciju pre alokacije, ali su i algoritmi slozeniji

    b. optimizacija posle alokacije – Gross-Henessy
    c. do alokacije registara smatramo da imamo beskonacno mnogo registara
12. Po kojoj semi Henessy-Paterson je radjena optimizacija na slajdu 39/48?
    a. #3 i #2

GaAs

1. Navesti 4 osnovne prednosti GaAs nad Si
    a. Brzi za red velicine pri istoj potrosnji
    b. Integracija elektronike i optike
    c. Sirok opseg radne temprerature -200 do +200
    d. Otporniji na radijaciju
2. Sta znaci efficient fault-tolerance
    a. Pustim gresku pa vidim da li je to na mestu gde mi to smeta
3. Sta su Multiple Chip Moduli i zbog cega se narocito korsite u GaAs
    a. Zbog ogranicenog broja tranzistora pravimo vise cipova stavimo u jedan hibridni, a onda ih povezemo na stampanu plocu(3 nivoa). Sto je cesca komunikacija cipovi postavljamo blize.
4. Za sabirace sa velikim brojem bita preko nekog kriticnog n, od RC i CL odabrati bolji(brzi) za SI i za GaAs
    a. RC je generalno najsporiji (Si), ali ima najmanje tranzistora, nema mnogo medjusobnh veza
    b. CL je generalno najbrzi(Si), ali ima najvise tranzistora i veliki broj grananja(GaAs osetljiv na fan-in i fan-out – njihovim povecanjem direktno se usporava cip)
    c. Sabirac koji je ubacen u 200Mhz je imao 8-bitni RC blokove povezane sa CL logikom
5. Sta je destination swap i zasto nastaje kod GaAs? Kako se izbegava?
    a. Memorija je kod GaAs iz vise modula i razlicito je kasnjenje do svakog.
    b. Inace kod GaAs su veze spore a memorija brza
    c. Zato se moze desiti da dodje podatak koji je kasnije trazen pre onog koji je ranije trazen
    d. Resenje je poslati rgister adress memorijskom cipu. Vraca se nazad zajedno sa sadrzajem memorijske lokacije
6. Skicirati semu Interleaved memory i Memory pipelining. Koji se pristup kad koristi
    a. Slajd 19
    b. Interleaving – memorija spora a off-chip delay mali(brzo dodjemo do memorije)
    c. Memory pipelining – memorija je brza ali sporo dolazimo do nje
7. Zbog cega se trostaticki baferi ne koriste od GaAs i sta se umesto toga koristi?
    a. Fan-in, umesto njih se korsite multiplekseri

VHDL

1. Objasniti temin System-on-a-Chip (SOC)
    a. Specifikacija dizajna na nivo komponenti
    b. Integracija senzora, transmitera I sl
2. Navesti domene strukturnog modelovanja od najviseg ka najnizem stepenu apstrakcije
    a. Processor-memory switch
    b. Register-transfer
    c. Gate
    d. Transistor
3. sta je vhdl I da li je zavistan od tehnologije izrade
    a. kombinuje elemente programiranja i tradicionalnog projektovanja hardverskih sistema
    b. omogucava komplekstan hijerarhijski dizajn
    c. nezavistan od tehnologije izrade
4. sta je ASIC?
    a. Specijalno dizajniran za konkretnu aplikaciju
    b. Opisan u nekom od HDL jezika
    c. Verifikovan koriscenjem simulatora
    d. Preveden u hardver koriscenjem alata za sintezu
    e. Implemetiran u FPGA, standard cell ili gate array tehnologiji
    f. Povezan sa spoljnim okruzenjem preko standardizovanih protokola I elektricnih interfejsa
5. sta je FPGA?
    a. Mogucnost reprogramiranja, moguce ispravljati greske cak I kad je proizvod isporucen
    b. Ali manji kapacitet I brzina u odnosu na integrisana kola
    c. Potrebni sofisticirani alati za mapiranje dizajna u sam cip
6. sta je clb?
    a. LUT i state. U LUT mozemo realizovati bilo koju logicku funkciju.
    b. Configurable logic block
7. Kako se radi sinteza u sistemu sa CLBovima
    a. Fleksibilnost CLBova omogucava lako mapiranje funkcija
    b. Logika se deli na n-ulazne funkcije koji se mapiraju u CLBove
    c. Alati za sintezu mogu imati specijalizovane rutine za neke gradivne blokove
    d. Javlja se problem rasporedjivanja blokova tako da se smanji komunikacija i kasnjenje
8. Da li je moguce svaki pin kod FPGA konfigurasati da radi kao ulazni, izlazni ili bidirekcioni
    a. Jeste pomocu jednog IOB(Input-output block) po pinu FPGA cipa
9. Da li je dozvoljeno vise arhitektura za jedan entitet? A obrnuto?
    a. Jeste, jer razlicite arhitekture mogu da imaju isti skup funkcija(interfejs)
    b. Obrnuto ne. Arhitektura ne moze imati razlicite interfejse
10. Razlika u dodeli promenljivoj i signalu u VHDLu? Kako se treiraju portovi?

a. Posmatracemo proces u kom se nalaze dve naredbe:
   - a=1
   - b=a
b. Ako je sa desne strane promenljiva, na levoj strani se dobija nova vrednost (a je promenljiva, b moze biti i promenljiva i signal, b dobija vrednost 1)
c. Ako je sa desne strane signal(a je signal, b moze biti i promenljiva i signal) b dobija vrednost 0
d. Portovi se tretiraju na isti nacin kao i signali
e. Od poziva do poziva procesa cuvamo i stare vrednosti promenljivih
f. Nije dozvoljeno specificirati kasjenja u dodelama promenljivih.

11. Sta se desava kad se stavi wait bez bilo kakvih argumenata? Sta ako se izostavi wait na kraju procesa?
    a. Ako nema wait statement na kraju procesa implicitno je wait sa signalima ukljucenim u sensitivity list u zaglavlju process statement
    b. Wait bez argumenata znaci da se proces suspenduje da se vise nikad ne pokrene(zgodno prilikom testiranja)
12. Da li je moguce i kako da se kod izmedju begin i end u procesu izvrsava u vise simulacionih ciklusa? Na primer neke naredbe se izvsavaju u jednom ciklusu a neke druge u narednom ciklusu.
    a. Moze ako u telu procesa postoji wait izmedju grupa naredbi
13. U sledecem primeru navesti kad cemo nastaviti sa izvrsavanjem

```
BIN_COMP : process
begin
wait on A, B until CLK = '1';
. . .
end process;
```

The process BIN_COMP is resumed after a change on either A or B signal, but only when the value of the signal CLK is equal to '1'.

14. Objasniti sustinu test bench modela
    a. To je arhitektura koja sadrzi model koji se testira
    b. Test vektori se dovode na ulaz
    c. Prati se stanje izlaznih signala ili koristeci simulator ili pomocu procesa koji kontrolise ispravno izvrsavanje
15. Kakav treba da bude boolean uslov USLOV da bi se ispisala poruka korisniku?

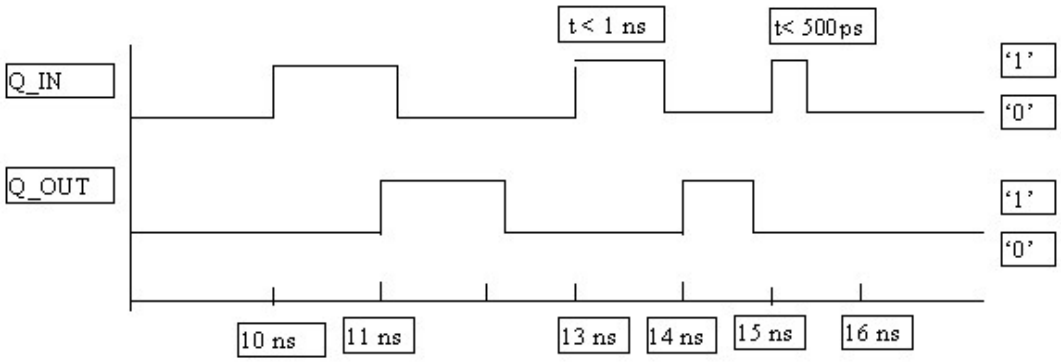
```
assert USLOV
      report "The call was not successful"
         severity WARNING;
```

    Odgovor: FALSE.
16. Da li je dogadjaj svako postavljanje signala(transakcija)?
    a. Ne, dogadjaj(event) je PROMENA vrednosti signala
17. Nabrojati faze u dizajniranju
    a. Analiza
    b. Elaboracija
    c. Simulacija
    d. Sinteza
18. Sta se radi u inicijalizaciji?

a. Svakom signalu se dodeljuje inicijalna vrenost
b. Vreme simulacije se postavlja na 0
c. Svaki proces se aktivira i izvrsava dok ne dodje do wait izraza, kada se suspenduje
d. Izvrsavanje se najcesce nastavlja nekom transakcijom

19. Navesti sta se desava u jednom simulacionom ciklusu
    a. Uveca se vreme simulacije do sledece transakcije
    b. Za svaku transakciju u ovom trenutku azurira se vrednost signala(ako je vrednost promenjena – dogadjaj)
    c. Svaki proces senzitivan na neki od ovih dogadjaja ili oni kod kojih je vreme isteklo se aktivira i izvrsava do wait iskaza
    d. Simulacija se zavrsava kada nema vise transakcija. Moze da se desi i da neko ceka na promenu nekog signala koja se nikad nece desiti (nema transakcija nad tim signalom, svi procesi blokirani) - i tad se sumulacija zavrsava.
20. Sta je delta vreme?
    a. Proces ne vidi efekte promene vrednosti signala sve dok se ne aktivira sledeci put, cak i ako je to u istom simulacionom trenutku
    b. Kasnjenje od 0fs se naziva delta kasnjenje
21. Ako imamo vise procesa koji upisuju u istim trenucima istu vrednost na isti signal, da i je to problem i kako se resava?
    a. Jeste
    b. Resava se na sledeci nacin:
    c. Ili da imamo vise signala pa da formiramo zajednicki signal logickom funkcijom nad signalima
    d. Ili oznacimo port sa bus. Tada ga vise procesa koristi. Potrebna je i bus resolution function. Kada se obradjuje transakcija, poziva se funkcija kojoj se kao parametar prosledjuje niz drajvera
22. Navesti razliku izmedju transportnog I inercijalnog kasnjenja
    a. Samo jedna od mnogih razlika je sledeca
    b. Kod transportnog mehanizma kad zadamo transakciju u neko trenutku 1000 brisu se sve ranije zadate transakcije u 1000. trenutku i posle njega
    c. Pobudnu signal kod inercijalnog koji traje krace od vremena propagacije nece se manifestovati na izlazu
23. Ako je Q_IN dato na slici nacrtati Q_OUT ako je zavisnost po formuli:

```
Q_OUT <= reject 500 ps inertial Q_IN after 1 ns;
```



24. Znacenje pure i impure u funkcijama
    a. Generalno - Funkcije nemaju nikakav uticaj na okruzenje
    b. Pure – za isti ulaz uvek daje isti izlaz (podrazumevano), ne zavisimo od promenljvih i signala deklarisanih izvan tela funkcije
    c. Impure – za isti ulaz ne mora uvek dati isti izlaz
25. Zasto test bench ne sadrzi deklaracije portova I generika
    a. test bench treba da predstavlja kompletno okruzenje u kome se testira Design Under Test
    b. test bench nikada nije sintetizibilan i nema interfejs (sekcija portovi u enititetu je prazna)

SYSTOLICS

1. Prikazati po koracima mnozenje dve matrice 2x2.
2. Prikazati po koracima mnozenje matrice 2x2 i vektora 2x1.