

Univerzitet u Beogradu  
Elektrotehnički Fakultet

Diplomski rad

# **VIZUELNA SIMULACIJA PREKIDAČKIH MREŽA**

Mentor:  
docent dr Boško Nikolić

Kandidat:  
Đorđe Trifunović

Beograd, februar 2010. godine

# Sadržaj

Uvod .....	3
1. Teorijske osnove problema .....	4
1.1 Struktura digitalnog uređaja .....	4
1.2 Realizacija jedinica .....	4
1.2.1 Operaciona jedinica.....	5
1.2.2 Upravljačka jedinica .....	7
1.3 Povezivanje jedinica .....	8
2. Korisnički interfejs softverskog paketa IGoVSoEDS.....	11
1.4 Osnovni prozor softverskog paketa .....	12
1.4.1 Podloga za prikaz elektronske digitalne strukture .....	12
1.4.2 Glavni meni i grafički skup poziva funkcija.....	13
1.4.3 Biblioteke kola i hijerarhijsko stablo modula .....	17
1.4.4 Navigacija i pregled rezultata rada simulatora .....	19
1.4.5 Pregled poruka i sadržaja strukture podataka .....	21
1.4.6 Sistem za pomoć pri korišćenju softverskog paketa .....	22
3. Realizacija simulatora .....	24
1.5 Simulator br. 1.....	24
1.6 Simulator br. 2.....	25
1.7 Simulator br. 4.....	27
1.8 Simulator br. 5.....	29
1.9 Simulator br. 7.....	30
1.10 Simulator br. 17.....	31
1.11 Simulator br. 18.....	33
1.12 Simulator br. 19.....	34
1.13 Simulator br. 21.....	35
1.14 Simulator br. 22.....	35
1.15 Simulator br. 23.....	37
1.16 Simulator br. 24.....	37
1.17 Simulator br. 25.....	38
4. Zaključak .....	39
5. Literatura .....	39

## Uvod

Predmet ovog diplomskog rada je grafička simulacija prekidačkih mreža. Grafička simulacija prekidačkih mreža je izvedena korišćenjem namenskog programskog paketa pod nazivom Interaktivni generator vizuelnih simulatora elektronskih digitalnih struktura (IGoVSoEDS). U okviru ovog rada napravljeno je ukupno dvadeset pet simulatora i oni se oslanjaju na zbirku zadataka iz ORT2 (osnova računarske tehnike 2). Cilj je bio da se za sve prekidačke mreže koje se nalaze u zbirci realizuju simulatori kako bi se vizuelnim prikazom bolje dočarao rad tih prekidačkih mreža u vremenu. Svi simulatori su složene celine koje se sastoje od uređaja koji se projektuje (operaciona i upravljačka jedinica) i uređaja sa kojima se vrši komunikacija (uređaj A, B, procesor, memorija).

U prvoj glavi su date teoretske osnove problema, objašnjena je struktura digitalnih uređaja, opisan je rad operacione i upravljačke jedinice i mogući načini njihovog međusobnog povezivanja.

U drugoj glavi se nalazi detaljno uputstvo softverskog paketa IGoVSoEDS koji je korišćen za realizaciju simulatora.

U trećoj glavi je opisana realizacija simulatora, navedene su modifikacije koje su napravljenije u odnosu na postavku zadataka i svi problemi koji su pratili realizaciju simulatora. Centralni uređaj P u svim simulatorima je realizovan kao u zbirci u koliko u ovom poglavlju nije drugačije naglašeno. Uređaji A, B i memorija su inicijalizovani početnim vrednostima i svaku simulaciju je moguće pokrenuti prostim pritiskom na *forward simulation* dugme. Za potrebe testiranja simulatora moguće je menjanje inicijalnih vrednosti svih uređaja. Veličina šema svih simulatora je prilagođena rezoluciji ekrana od 1280x1024 piksela, kako bi svaki simulator mogao kompletno da stane na jedan ekran.

# 1. Teorijske osnove problema

(preuzeto iz rada profesora Jovana Đorđevića [3])

## 1.1 Struktura digitalnog uređaja

Digitalni uređaj su prekidačke mreže koje realizuju jednu ili više operacija nad podacima koji su predstavljeni u binarnom obliku. Ukoliko uređaj može da realizuje ne jednu već više operacija, onda se i informacije o tome koju operaciju treba realizovati takođe zadaju binarnim vrednostima. Digitalni uređaji su najčešće toliko složene prekidačke mreže da se projektovanje digitalnih uređaja ne može realizovati formalnim postupcima sinteze prekidačkih mreža. Zbog toga postoje samo određeni postupci kojih se treba pridržavati prilikom projektovanja digitalnih uređaja. To dovodi do toga da prilikom projektovanja digitalnih uređaja veliku ulogu ima iskustvo projektanta i da može da bude projektovano više uređaja koji realizuju istu operaciju ili iste operacije a strukturne šeme im se razlikuju.

Jedan od često primenjivanih postupaka je da se izvršavanje operacije ili operacija podeli na logičke celine koje se nazivaju faze i da za izvršavanje svake faze u digitalnom uređaju postoji posebna jedinica (slika 1). Ukoliko je izvršavanje operacije ili operacija podeljeno na N faza, uređaj mora da ima N jedinica. Izvršavanje svake operacije započinje u jedinici 1 na osnovu početnih vrednosti podataka koji se nalaze u registrima i/ili memorijama jedinice 1. Međurezultati faze 1 se prebacuju iz jedinice 1 u jedinicu 2 u kojoj se nastavlja sa izvršavanjem faze 2 operacije. Na sličan način se izvršavaju i ostale faze operacije u ostalim jedinicama uređaja i vrši prebacivanje međurezultata između jedinica. Rezultat operacije se nalazi u registru ili memoriji jedinice N.



**Slika 1** Struktura digitalnog uređaja

Primer je izvršavanje instrukcije računara koje se deli na fazu očitavanje instrukcije, fazu dekodovanje instrukcije i čitanje podataka, fazu izvršavanje operacije i fazu opsluživanje prekida.

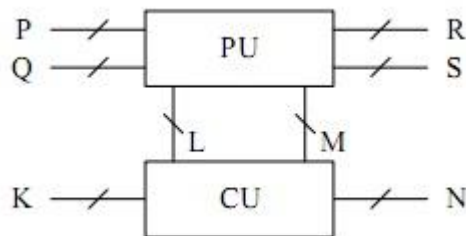
Kod ovog pristupa realizacije uređaja treba rešavati dva pitanja:

- realizacija jedinica i
- povezivanje jedinica.

## 1.2 Realizacija jedinica

Jedinica digitalnog uređaja se sastoji iz (slika 2):

- operacione jedinice PU (Processing Unit) i
- upravljačke jedinice CU (Control Unit).



**Slika 2** Struktura jedinice digitalnog uređaja

### 1.2.1 Operaciona jedinica

Operaciona jedinica se sastoji od kombinacionih i sekvencijalnih prekidačkih mreža koje se prema svojoj funkciji mogu podeliti u tri grupe.

U prvu grupu spadaju sekvencijalne prekidačke mreže koje služe za pamćenje binarnih reči kojima se definišu operacija koju treba realizovati i podaci nad kojima treba da se realizuju operacije, kao i binarnih reči koje predstavljaju međurezultate i rezultate operacija. To su uglavnom registri i memorije.

U drugu grupu spadaju kombinacione i sekvencijalne prekidačke mreže koje realizuju izabrani skup mikrooperacija. To su uglavnom sabirači, aritmetičko logičke jedinice, pomerači, dekoderi, multiplekseri, itd.

U treću grupu spadaju kombinacione i sekvencijalne prekidačke mreže koje realizuju logičke uslove. To su uglavnom komparatori i brojači, ali i posebno konstruisane kombinacione i sekvencijalne prekidačke mreže.

#### Realizacija operacija

Realizacija operacije u opštem slučaju zahteva više taktova.

Operacija se razlaže na niz koraka tako da u svaki korak uđe jedna ili više mikrooperacija.

#### Mikrooperacije

Mikrooperacija je neka operacija koja se realizuje u jednom taktu neke kombinacione ili sekvencijalne mreže.

Mikrooperacije predstavljaju sve operacije koje realizuje aritmetičko logička jedinica, pomerač za jedno ili više mesta ulevo ili udesno, multiplekser, inkrementirajući i dekrementirajući brojač, itd.

Operacija sabiranja dve binarne reči dužine  $n$  bita iz registara A i B i smeštanje rezultata u registar F može da se realizuje i kao

- mikrooperacija i kao
- operacija.

Ukoliko se koristi paralelni sabirač dužine  $n$  razreda, operacija sabiranja se realizuje u jednom taktu i predstavlja mikrooperaciju.

Ukoliko se koristi samo jednorazredni potpuni sabirač, operacija sabiranja se realizuje u  $n$  taktova i predstavlja operaciju.

Za realizaciju operacija se najčešće koriste sledeće mikrooperacije:

- aritmetičke mikrooperacije  $F=A+B$ ,  $F=A-B$ ,  $F=A+1$  i  $F=A-1$ ,
- logičke mikrooperacije  $F=A \vee B$ ,  $F=A \& B$ ,  $F=A \oplus B$  i  $F=\neg A$ ,
- pomeračke mikrooperacije za jedno mesto udesno  $F=I_R.A(n-1:1)$  i jedno mesto ulevo  $F=A(n-2:0).I_L$  i
- mikrooperacije prenosa kroz multiplekser  $F=A$ .

### Broj mikrooperacija u koraku

Mikrooperacije  $F=A+B$  i  $E=C+D$  mogu da se realizuju u jednom koraku, jer nema međuzavisnosti, ali su za to potrebna dva sabirača.

Mikrooperacije  $F=A+B$  i  $E=F+C$  ne mogu da se realizuju u jednom koraku, jer ima međuzavisnosti, i pored toga što postoje dva sabirača.

### Logički uslovi

Redosled koraka u opštem slučaju zavisi i od tzv. logičkih uslova koji pokazuju neke karakteristike binarnih reči nad kojima se realizuje operacija.

### Algoritam operacije

Operacija se razlaže na niz koraka tako da u svaki korak uđe jedna ili više mikrooperacija.

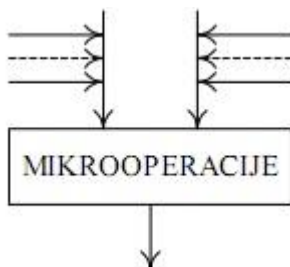
Na redosled koraka utiču i logički uslovi.

Uređeni niz koraka na koje je razložena neka operacija i logički uslovi na osnovu kojih se određuje redosled koraka definišu algoritam operacije.

### Dijagrami toka

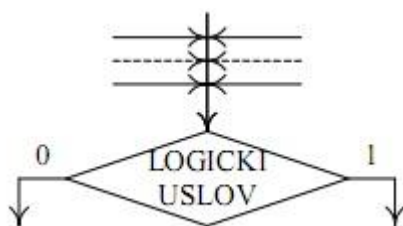
Za predstavljanje algoritama operacija koriste se dijagrami toka koji se crtaju pomoću operacionih i uslovnih blokova.

Operacioni blok ima jedan ili više ulaza i jedan izlaz. U operacioni blok se upisuju mikrooperacije koje se realizuju u jednom koraku algoritma (slika 3).



**Slika 3** Operacioni blok

Uslovni blok ima jedan ili više ulaza i dva izlaza. U uslovni blok se upisuje logički uslov koje definiše grananje algoritma. Pritom 1 označava da je uslov zadovoljen, a 0 da nije (slika 4).



**Slika 4** Uslovni blok

Operacioni i uslovni blokovi se povezuju linijama tako da izlaz nekog bloka vodi na ulaz jednog i samo jednog bloka.

### Registarski prenos (RTL)

Mikrooperacije u dijagramu toka uzimaju binarne vrednosti iz registara, transformišu ih propuštanjem kroz kombinacione mreže i novoformirane vrednosti upisuju u registre.

Zbog toga se kaže da dijagrami toka operacija predstavljaju algoritme operacija na nivou registarskog prenosa (Register Transfer Level).

## **1.2.2 Upravljačka jedinica**

Upravljačka jedinica se sastoji iz sekvencijalnih i kombinacionih prekidačkih mreža koje saglasno algoritmu operacije i vrednostima signala logičkih uslova generišu upravljačke signale za operacionu jedinicu. Ovi signali određuju koje mikrooperacije i po kom redosledu treba izvršavati u operacionoj jedinici da bi se kao rezultat posle određenog broja signala takta realizovala operacija.

### Povezivanje operacione i upravljačke jedinice

Signali L predstavljaju signale logičkih uslova koji dolaze iz PU u CU.

Signali M predstavljaju upravljačke signale koji dolaze iz CU u PU.

### Funkcije operacione i upravljačke jedinice

Funkcije operacione jedinice su definisane skupom registara i memorija, skupom mikrooperacija i skupom signala logičkih uslova.

Funkcije upravljačke jedinice su definisane skupom algoritama operacija za čiju realizaciju jedinica generiše upravljačke signale.

### Projektovanje operacionih i upravljačkih jedinica

Projektovanje operacionih i upravljačkih jedinica se realizuje u dve faze.

U prvoj fazi se paralelno projektuju operaciona jedinica i algoritmi zadatih operacija. Ova faza se zasniva na veštini i iskustvu projektanata, jer formalnih metoda nema.

U drugoj fazi se projektuje upravljačka jedinica na osnovu algoritama operacija. Ova faza se realizuje korišćenjem neke od više formalnih metoda.

## **1.3 Povezivanje jedinica**

### Signali

U jedinicu u kojoj se realizuje i-ta faza operacije dolaze iz jedinice u kojoj se realizuje (i-1)-va faza operacije tri grupe signala:

- signali P koji predstavljaju binarne reči koje dolaze u PU,
- signali Q koji predstavljaju upravljačke signale koji dolaze u PU i
- signali K koji predstavljaju signale logičkih uslova koji dolaze u CU.

Iz jedinice u kojoj se realizuje i-ta faza operacije odlaze iz jedinice u kojoj se realizuje (i+1)-va faza operacije tri grupe signala:

- signali R koji predstavljaju binarne reči koje odlaze iz PU,
- signali S koji predstavljaju signale logičkih uslova koji odlaze iz PU i
- signali N koji predstavljaju upravljačke signale koji odlaze iz CU.

### Prebacivanje binarnih reči

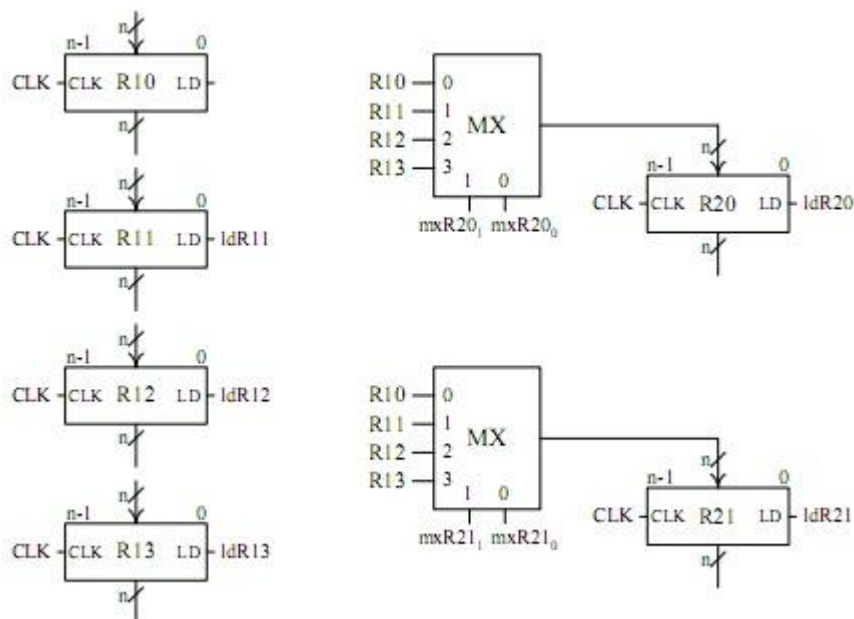
Prebacivanje binarnih reči između uređaja predstavlja pitanje povezivanja izlaza registara iz jedinice iz koje se šalju binarne reči na ulaze registara jedinice u koju se šalje binarna reč.

Povezivanja registara se može realizovati na više načina, pri čemu su ona varijante dva osnovna načina povezivanja:

- povezivanje direktnim vezama i
- povezivanje magistralom.

Kao primer su uzeta četiri registra R10, R11, R12 i R13 iz jedinice 1 čije sadržaje treba prebacivati u registre R20 i R21 jedinice 2.

Povezivanje direktnim vezama



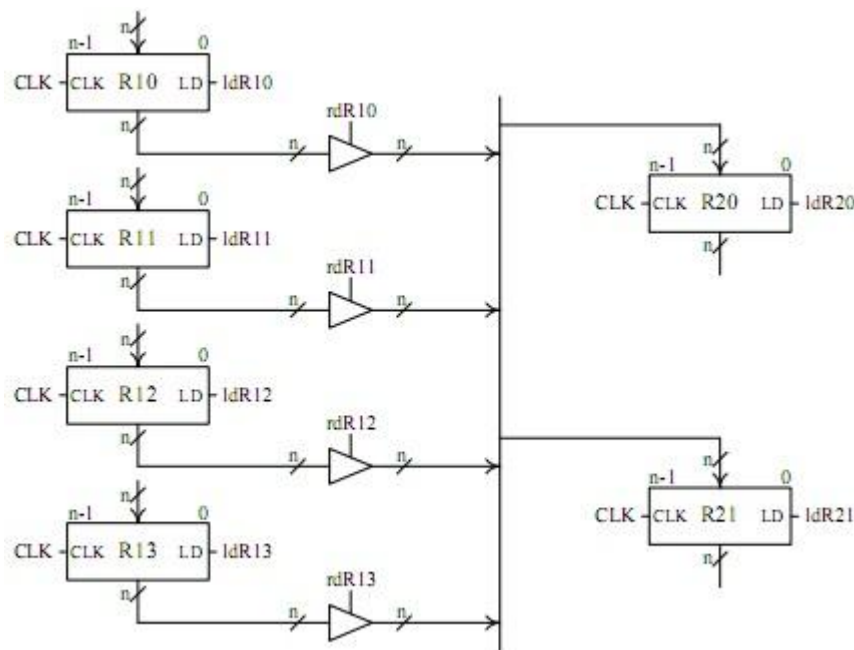
Postoje posebne veze između registara R10, R11, R12 i R13 jedinice 1 i registara R20 i R21 jedinice 2.

Multiplekserima koji su ispred registara R20 i R21 jedinice 2 se selektuje po jedan od registara R10, R11, R12 i R13 jedinice 1. Signalima  $mxR20_1$  i  $mxR20_0$  se selektuje jedan od registara R10, R11, R12 i R13 jedinice 1 i signalom  $ldR20$  upisuje u registar R20 jedinice 2. Signalima  $mxR21_1$  i  $mxR21_0$  se selektuje jedan od registara R10, R11, R12 i R13 jedinice 1 i signalom  $ldR21$  upisuje u registar R21 jedinice 2.

Dobra strana je da se istovremeno može upisivati jedan od registara R10, R11, R12 i R13 jedinice 1 u registar R20 jedinice 2 i jedan od registara R10, R11, R12 i R13 jedinice 1 u registar R21 jedinice 2.

Loša strana je veliki broj linija.

### Povezivanje magistralom



Ne postoje posebne veze između registara R10, R11, R12 i R13 jedinice 1 i registara R20 i R21 jedinice 2.

Preko bafera sa tri stanja se registri R10, R11, R12 i R13 jedinice 1 povezuju na linije magistrale, a linije magistrale se vode na ulaze registara R20 i R21 jedinice 2. Aktivnom vrednošću samo jednog od signala rdR10, rdR11, rdR12 i rdR13 jedinice 1 sadržaj samo jednog od registara R10, R11, R12 i R13 jedinice 1 se pušta na linije magistrale, a aktivnom vrednošću jednog od signala ldR20 ili ldR21 ili oba signala sadržaj sa linija magistrale se upisuje u jedan od registara R20 ili R21 ili u oba registra.

Dobra strana je manji broj linija nego u slučaju povezivanja direktnim vezama.

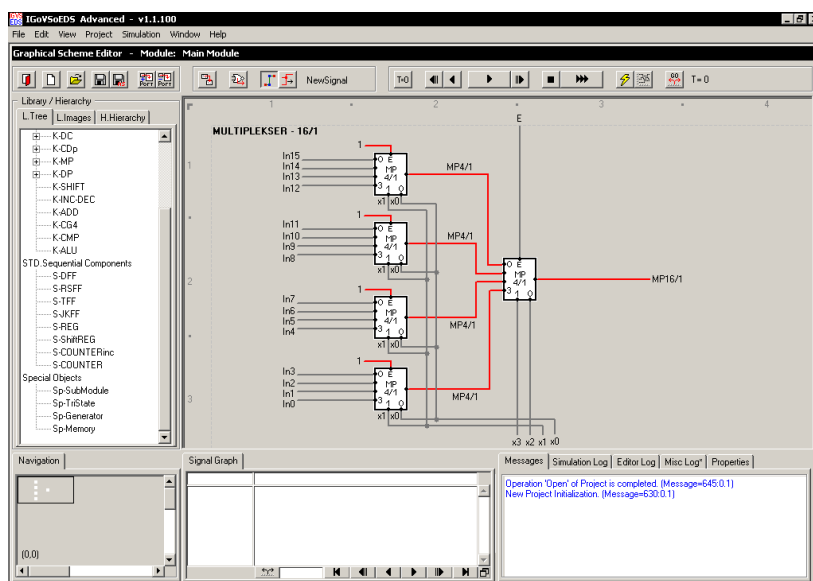
Loša strana je da se istovremeno ne može upisivati jedan od registara R10, R11, R12 i R13 jedinice 1 u registar R20 jedinice 2 i jedan od registara R10, R11, R12 i R13 jedinice 1 u registar R21 jedinice 2, već se to mora uraditi u dva posebna koraka.

## 2. Korisnički interfejs softverskog paketa IGoVSoEDS (preuzeto iz rada Nenada Grbanovića [4])

*Interaktivni generator vizuelnih simulatora elektronskih digitalnih struktura (IGoVSoEDS)* pruža intuitivan korisnički interfejs za kreiranje i modifikaciju elektronskih digitalnih struktura, omogućava detaljno podešavanje značajnih parametara svih elektronskih digitalnih kola i modula, i obezbeđuje jednostavno upravljanje simulacijom i efikasan sistem pregleda rezultata rada simulacije trenutno aktivne elektronske digitalne strukture (u daljem tekstu *projekat*). Na slici 1 prikazan je osnovni prozor softverskog paketa *IGoVSoEDS*, koji se prilagođava trenutnoj rezoluciji radne površine, i može da se koristi za sve rezolucija. Glavne komponente softverskog paketa *IGoVSoEDS* su:

- Osnovni prozor softverskog paketa, čiji su sastavni delovi:
  - Podloga za prikaz elektronske digitalne strukture
  - Glavni meni i grafički skup poziva funkcija
  - Biblioteke kola i hijerarhijsko stablo modula
  - Navigacija i pregled rezultata rada simulatora
  - Pregled poruka i sadržaja strukture podataka
  - Sistem za pomoć pri korišćenju softverskog paketa
- Funkcije grafičkog editora elektronske digitalne strukture
- Rad sa parametrima simulacije i upravljanje simulatorom

Softverski paket je realizovan u tri verzije. *IGoVSoEDS Advanced* ima puni kapacitet strukture podataka za prihvatanje opisa digitalne strukture i na raspolaganju su sve funkcionalnosti, *IGoVSoEDS Basic* ima manji kapacitet i nisu na raspolaganju sve funkcionalnosti, dok verzija *IGoVSoEDS Simulation only* obezbeđuje samo izvršavanje simulacije unapred pripremljenih digitalnih struktura. Za sve verzije softverskog paketa *IGoVSoEDS* na raspolaganju je i odgovarajuća verzija za demonstraciju mogućnosti (*IGoVSoEDS Demo*). U narednim poglavljima detaljno su prikazane mogućnosti svih delova osnovnog prozora softverskog paketa, kao i svih postupaka za upravljanje radom softverskog paketa.



**Slika 1** Osnovni prozor softverskog paketa *IGoVSoEDS*, sa realizacijom multipleksera 16/1

## 1.4 Osnovni prozor softverskog paketa

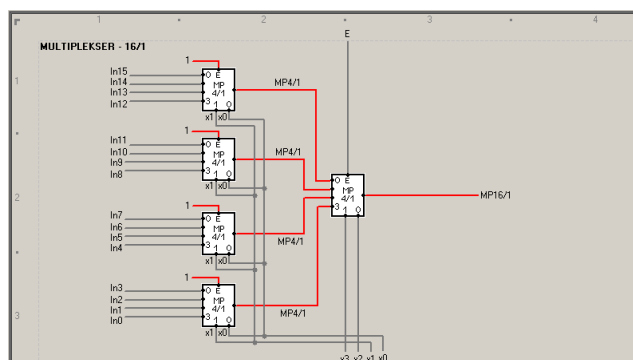
U poglavljima koja slede daje se detaljan pregled svih komponenti osnovnog prozora softverskog paketa *IGoVSoEDS*.

### 1.4.1 Podloga za prikaz elektronske digitalne strukture

Na slici 2 prikazan je deo podloge na kojoj se realizuju topologija šeme elektronskih digitalnih struktura (u daljem tekstu *podloga za prikaz strukture*). Prikazan je deo podloge koji je trenutno vidljiv (slika 1), dok je ostatak podloge dostupan pomeranjem podloge pomoću strelica na tastaturi, u smerovima levo–desno i gore–dole. Na svim rubovima podloge prikazan je lenjir za lakše određivanje pozicije na podlozi. Lenjir ima po 10 podeoka na svim rubovima podloge, i može da se ukloni aktiviranjem odgovarajuće funkcije glavnog menija (*View*, slika 4).

Veličina podloge za prikaz strukture je nepromenljiva i prilagođena je prikazu pri rezoluciji 1280x1024, kada je vidljivo 25% podloge (50% po horizontali i 50% po vertikali). Na slici 2, pri rezoluciji 1024x768, vidljivi deo podloge je 38% po horizontali i 36% po vertikali, što daje vidljivost od oko 14% celokupne raspoložive podloge.

Broj podloga za prikaz strukture ograničen je na 2048 podloga, što daje raspoloživu radnu površinu od oko 300m<sup>2</sup>, pri rezoluciji 1280x1024 i dijagonali monitora od 17". Ovako definisana radna površina može da prihvati do 500.000 D flip–flopova, sa prosečnom pokrivenošću radne površine D flip–flopovima od oko 20%.



**Slika 2** Vidljivi deo podloge na kojoj se realizuje topologija šeme elektronske digitalne strukture

Ugrađene funkcije obezbeđuju:

- Dodavanje, pomeranje i brisanje standardnih sekvencijalnih i kombinacionih kola, kao i specijalnih kola i modula
- Dodavanje, pomeranje i brisanje signala i linija signala
- Dodavanje, pomeranje i brisanje natpisa opšte namene (*General Purpose Labels*), natpisa naziva signala (*Signal Name Labels*) i natpisa logičkih stanja na linijama signala koji su širi od jednog bita (*Signal State Labels*)
- Poravnavanje kola u odnosu na odabrano referentno kolo po levoj, desnoj, gornjoj ili donjoj ivici, kao i poništavanje poslednjeg poravnavanja (funkcija *Alignment*)
- Promenu veličine i izgleda *Port Module*–a i *NonPort Module*–a
- Jednostavnu zamenu *Port Module*–a i *NonPort Module*–a (funkcije *Replace With ALL Connections...* i *Replace Without ANY Connection...*), i slično

Navedene funkcija detaljno se opisuju u poglavljima koja slede.

## 1.4.2 Glavni meni i grafički skup poziva funkcija

Na slici 3 prikazan je gornji deo osnovnog prozora softverskog paketa *IGoVSoEDS*, na kome su informacija o verziji softverskog paketa, glavni meni, naziv modula elektronske digitalne strukture koji je trenutno prikazan i grafički skupovi poziva funkcija (*toolbars*).

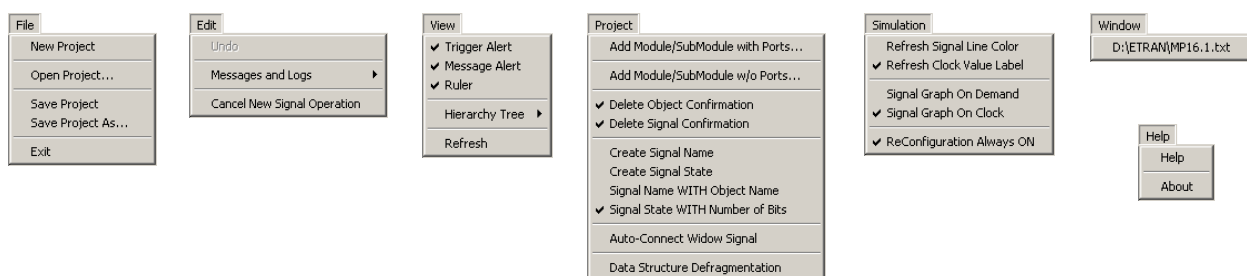


Slika 3 Gornji deo osnovnog prozora softverskog paketa *IGoVSoEDS*

U poglavljima koja slede daju se pregledi glavnog menija i grafičkih skupova poziva funkcija.

### 1.4.2.1 Glavni meni

Na slikama 4 i 5 prikazan je glavni meni softverskog paketa *IGoVSoEDS*.



Slika 4 Glavni skup funkcija softverskog paketa *IGoVSoEDS* – Glavni meni



Slika 5 Skup funkcija *Edit/Messages and Logs* (levo) i *View/Hierarchy Tree* (desno)

Na raspolaganju su grupe funkcija:

- *File* – skup standardnih funkcija za aktiviranje novog projekta, kao i za rad sa datotekama:
  - *New Project* – aktiviranje novog projekta
  - *Open Project...* – dijalog za otvaranje projekta, učitavanjem sadržaja datoteke (slika 6–levo)
  - *Save Project* – snimanje trenutno aktivnog projekta u datoteku
  - *Save Project As...* – dijalog za snimanje trenutno aktivnog projekta u datoteku (slika 6–desno)
  - *Exit* – izlazak iz aplikacije

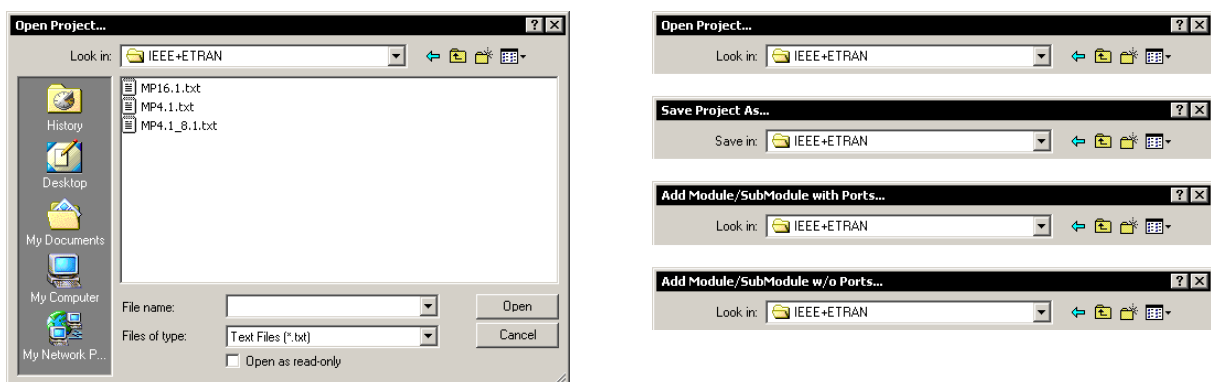
U zavisnosti od tipa funkcije razlikuju se četiri dijaloga za rad sa datotekama (slika 6).

- *Edit* – funkcije za poništavanje akcija korisnika (*Undo* i *Cancel New Signal Operation*), kao i funkcije za rad sa objektima za obaveštavanje korisnika o radu softverskog paketa (skup funkcija *Messages and Logs*, slika 5–levo):
  - *Undo* – funkcija za poništavanje operacija dostupna je samo u dva slučaja:
    - *Undo Alignment* – poništavanje operacije poravnavanja objekata
    - *Undo Signal* – poništavanje operacije kreiranja signalaFunkcija *Undo* može da se iskoristi samo neposredno nakon obavljanja navedenih operacija.
  - *Messages and Logs* – skup funkcija za brisanje dela ili svih poruka u sistemu za informisanje, kao i u delu za arhiviranje i prikaz arhiviranih poruka.

- *Cancel New Signal Operation* – prekidanje operacije kreiranja novog signala i poništavanje svih efekata započete operacije.
- *View* – funkcije za definisanje vrsta i načina prikaza poruka (*Trigger Alert*, *Message Alert* i *Ruler*), i prikaza hijerarhijskog stabla modula (skup funkcija *Hierarchy Tree*, slika 5–desno), kao i funkcija za restauraciju sadržaja podloge za prikaz strukture (*Refresh*):
  - *Trigger Alert* – parametar koji određuje da li nakon zaustavljanja simulacije usled ispunjavanja uslova definisanih nekim od događaja za zaustavljanje simulacije treba prikazati poruku u sistemu za informisanje i arhiviranje poruka, o razlozima za zaustavljanje simulacije.
  - *Message Alert* – parametar koji određuje da li treba da se prikaže kratka animacija prilikom kreiranja poruka u sistemu za informisanje i arhiviranje poruka.
  - *Ruler* – parametar koji određuje da li treba prikazati lenjir na podlozi za prikaz strukture.
  - *Hierarchy Tree* – skup parametara koji određuju na koji način treba da budu prikazani nazivi modula, u prostoru za prikaz hijerarhijske organizacije modula.
  - *Refresh* – funkcija za restauraciju sadržaja podloge za prikaz strukture. Ova funkcija se implicitno poziva nakon svih operacija kojima se menja sadržaj podloge za prikaz strukture, ali nije uvek u potpunosti uspešna, zbog karakteristika softverskog okruženja u kome je realizovan softverski paket. Stoga je u sistemu menija obezbeđena mogućnost eksplicitnog pozivanja ove funkcije.
- *Project* – funkcije za otvaranje dijaloga za učitavanje modula iz datoteka, kao i za definisanje vrsta i načina komunikacije korisnika i softverskog paketa:
  - *Add Module/SubModule with Ports...* – dodavanje modula sa portovima, definisanog sadržajem datoteke, trenutno aktivnom projektu (slika 6–desno). Dodati modul postaje integralni deo trenutno aktivnog projekta. Ukoliko u datoteci nisu definisani portovi, modul se dodaje bez portova.
  - *Add Module/SubModule w/o Ports...* – dodavanje modula bez portovima, definisanog sadržajem datoteke, trenutno aktivnom projektu (slika 6–desno). Dodati modul postaje integralni deo trenutno aktivnog projekta. Ukoliko su u datoteci definisani i portovi, portovi se ne kreiraju, i nije moguće naknadno dodavanje portova. Jedini način za naknadno dodavanje portova je brisanje modula bez portova i dodavanje modula pozivanjem funkcije *Add Module/SubModule with Ports...*
  - Skup parametara kojima se definiše ponašanje softverskog paketa:
    - *Delete Object Confirmation* – parametar koji određuje da li se prilikom brisanja kola i modula prikazuje poruka sa zahtevom da se operacija brisanja još jednom potvrdi.
    - *Delete Signal Confirmation* – parametar koji određuje da li se prilikom brisanja signala ili linije signala prikazuje poruka sa zahtevom da se operacija brisanja još jednom potvrdi.
    - *Create Signal Name* – parametar koji određuje da li se pri crtanju linije signala kreira natpis naziva signala. Ako je parametar obeležen, onda se natpis signala kreira ukoliko je linija signala dovoljno dugačka.
    - *Create Signal State* – parametar koji određuje da li se pri crtanju linije signala kreira natpis stanja signala, za signale šire od jednog bita.
    - *Signal Name WITH Object Name* – parametar koji određuje format kreiranja naziva signala prilikom kreiranja signala. Na raspolaganju su naziv kreiran prema nazivu konektora koji je izvor vrednosti kreiranog signala i naziv kreiran kao kombinacija naziva kola i konektora koji su izvor vrednosti kreiranog signala.
    - *Signal State WITH Number of Bits* – parametar koji određuje format natpisa stanja signala, za signale šire od jednog bita. Ovaj parametar definiše format svih natpisa stanja signala za trenutno aktivan projekat.
    - *Auto-Connect Widow Signal* – parametar koji određuje da li sistem implicitno treba da uspostavi vezu između signala van *NonPort Module*-a i signala unutar *NonPort Module*-a, prilikom povezivanja signala i *NonPort Module*-a, ukoliko je jednoznačno određeno da

postoji samo jedan signal unutar *NonPort Module*–a sa kojim je moguće ostvariti vezu. Ukoliko ne postoji jednoznačno određen signal unutar *NonPort Module*–a, otvara se dijalog za izbor signala sa kojim će se ostvariti veza.

- *Data Structure Defragmentation* – defragmentacija sadržaja strukture podataka *IGoVSoEDS* koja čuva dizajn digitalne strukture koja je trenutno prikazana u softverskom paketu. Ova funkcija implicitno se poziva pri aktiviranju svih funkcija softverskog paketa za koje je neophodno da struktura podataka bude defragmentirana.
- *Simulation* – funkcije za definisanje vrsta i načina promena na podlozi za prikaz strukture i na drugim oblastima za prikaz rezultata rada simulatora, tokom i neposredno nakon završetka rada funkcija simulatora (npr. *Reset Simulation*, *Forward Simulation – Next Clock*, *Forward Simulation – Run Simulation*,...):
  - *Refresh Signal Line Color* – parametar koji određuje da li za svaki simulirani takt treba da se osveže boje linija signala i natpisi stanja signala, prilikom izvršavanja simulacije niza taktova (funkcija simulatora *Forward Simulation – Run Simulation*).
  - *Refresh Clock Value Label* – parametar koji određuje da li za svaki simulirani takt treba da se osveži natpis za prikaz broja simuliranih taktova, prilikom izvršavanja simulacije niza taktova (funkcija simulatora *Forward Simulation – Run Simulation*).
  - *Signal Graph On Demand* – parametar koji određuje da li za svaki simulirani takt treba da se osveži prikaz vremenskih oblika signala, prilikom izvršavanja svih funkcija za simulaciju. Ovaj parametar je stariji od parametra *Signal Graph On Clock*.
  - *Signal Graph On Clock* – parametar koji određuje da li za svaki simulirani takt treba da se osveži prikaz vremenskih oblika signala, prilikom izvršavanja *Forward Simulation – Run Simulation* funkcije simulatora.
  - *ReConfiguration Always ON* – parametar koji određuje da li se sa svakom promenom digitalne strukture ažurira opis digitalne strukture koji se čuva u strukturi podataka. Ovaj parametar je potrebno isključiti kada se obavljaju promene veoma složenih digitalnih struktura.
- *Window* – prikazuje apsolutnu putanju do datoteke u kojoj se čuva potpuni opis trenutno aktivne digitalne strukture. Datoteka sadrži stanje opisa digitalne strukture saglasno stanju prilikom učitavanja sadržaja datoteke, odnosno prilikom poslednjeg poziva funkcija *Save Project* i *Save Project As...*, ukoliko su ove funkcije aktivirane nakon učitavanja datoteke.
- *Help* – pozivi funkcija sistema za pomoć pri korišćenju softverskog sistema (*About* i *Help*).



**Slika 6** Funkcije *File/Open Project...*, *File/Save Project As...*, *Project/Add Module.SubModule with Ports...* i *Project/Add Module.SubModule w/o Ports...*

### 1.4.2.2 Grafički skup poziva funkcija

Grafički skupovi funkcija obezbeđuju prečice ka najznačajnijim funkcijama realizovanim u glavnom meniju softverskog paketa *IGoVSoEDS*, kao i skup funkcija za upravljanje radom simulatora trenutno aktivne digitalne strukture (*Toolbars*, slika 7).



**Slika 7** Grafički skup funkcija za rad sa datotekama (*File Toolbar*), za izbor režima rada grafičkog editora (*Editor Mode Toolbar*) i za rad sa funkcijama simulatora (*Simulation Toolbar*)

Grafički skupovi funkcija obezbeđuju:

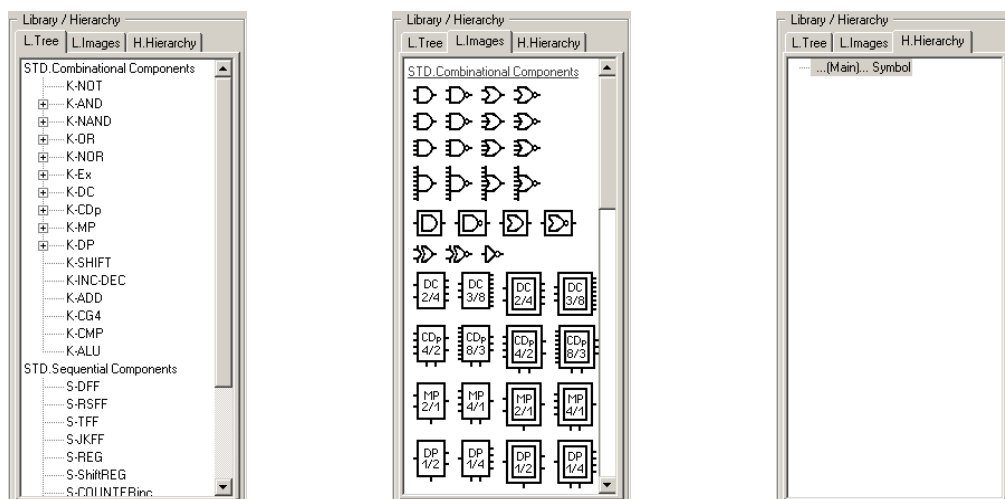
- rad sa datotekama (*File Toolbar*), odnosno pozive najznačajnijih funkcija glavnog menija:
  - *File Menu* (*Exit*, *New Project*, *Open Project...*, *Save Project* i *Save Project As...*)
  - *Project Menu* (*Add Module/SubModule with Ports...* i *Add Module/SubModule w/o Ports...*)
- izbor režima rada grafičkog editora elektronske digitalne strukture (*Editor Mode Toolbar*):
  - *Change Screen* – Ukoliko je aktivan režim rada *Change Screen*, ulazak u modul aktivne elektronske digitalne strukture ostvaruje se dvostrukim aktiviranjem levog tastera miša (*double click*) na objektu koji predstavlja modul. Povratak na prikaz hijerarhijski neposredno nadređenog modula, u odnosu na trenutno prikazani modul, ostvaruje se dvostrukim aktiviranjem levog tastera miša na podlozi trenutno prikazanog modula. Ako nije aktivan režim rada *Change Screen*, akcije ulaska u modul i povratka na nadređeni modul ostvaruju se pozivanjem funkcija *Change Screen INTO* i *Change Screen UP*, u padajućim menijima *Object Menu* i *Canvas Menu*, respektivno.
  - *Move Object* – Ukoliko je aktivan režim rada *Move Object*, omogućeno je pomeranje objekata koji predstavljaju standardna sekvencijalna i kombinaciona kola, kao i podređene module. Moguće je pomeranje i natpisa za prikaz naziva i stanja signala, kao i natpisa opšte namene. Pored osnovne namene, *Move Object* režim rada obezbeđuje i dodatne mogućnosti:
    - *Select* – Grupisanje skupa objekata čija pozicija na podlozi može zajednički da se podešava pozivom funkcija za grupnu promenu položaja (*Special Menu*, *Alignment* funkcija: *Left*, *Right*, *Top* i *Bottom*).
    - *Resize Object* – Promena veličine *NonPort Module*-a, uz ograničenja koja se uvode ukoliko ima linija signala koje su povezane na modul.
  - *New Signal* – Kreiranje signala i linija signala
  - *Move Signal* – Modifikacija postojećih linija signala
  - *Current Mode Indicator* – Indikator koji pokazuje trenutno aktivan režim rada
- skup funkcija za rad sa simulatorom elektronske digitalne strukture (*Simulation Toolbar*). U ovom paragrafu daje se samo pregled funkcija i informativnih polja.
  - *Reset Simulation* – tekuća simulacija dovodi se u početno stanje
  - *Fast Backward Simulation* – postavljanje simulacije na prvi prethodni trenutak memorisanog stanja simulatora
  - *Backward Simulation* – vraćanje simulacije za jedan takt unazad
  - *Forward Simulation*, *Next Clock* – aktiviranje simulacije jednog narednog sistemskog signala takta, ali i svih narednih sistemskih signala takta do trenutka pojavljivanja uzlazne ivice prvog narednog signala takta
  - *Fast Forward Simulation* – postavljanje simulacije na prvi naredni trenutak memorisanog stanja simulatora

- *Stop Simulation* – zaustavljanje simulacije, za sve simulacije koje obuhvataju obradu više sistemskih signala takta
- *Forward Simulation, Run Simulation* – startovanje simulacije, odnosno aktiviranje simulacije za obradu proizvoljnog broja sistemskih signala takta do prvog sledećeg kriterijuma za zaustavljanje simulacije, odnosno do aktiviranja funkcije *Stop Simulation*
- *Triggers Dialog Box* – otvaranje dijaloga za definisanje događaja za zaustavljanje simulacije
- *Signal Graph* – otvaranje dijaloga za rad sa listom signala za pregled promena vremenskih oblika signala
- *Go To Tclk* – funkcija koja stanje simulatora dovodi u stanje jednako postavljenoj vrednosti signala takta u polju *New Tclk Value*. Prvo aktiviranje funkcije *Go To Tclk* kreira polje *New Tclk Value*, a naredno aktiviranje dovodi simulator na vremenski trenutak koji je postavljen u polju *New Tclk Value*, i potom uklanja polje *New Tclk Value* (slike 7–desno–gore i 7–desno–sredina).
- *Clock Info* – prikaz trenutne vrednosti sistemskog signala takta (slika 7–desno–gore)
- *New Tclk Value* – polje za postavljanje parametra funkcije *Go To Tclk* (slika 7–desno–sredina)
- *ReConfiguration is Stopped...* – informacija da struktura za čuvanje opisa digitalne strukture ne sadrži dovoljno informacija za simulaciju rada digitalne strukture (slika 7–desno–dole)

### 1.4.3 Biblioteke kola i hijerarhijsko stablo modula

Na slici 8 prikazana su dva raspoloživa sadržaja dela osnovnog prozora za izbor kola koje treba staviti na podlogu za prikaz strukture (*Library: L.Tree* i *L.Images*, levo i sredina, respektivno), i pregled hijerarhijskog stabla za prikaz hijerarhijske strukture modula trenutno prikazane elektronske digitalne strukture (*Hierarchy: H.Hierarchy*, desno).

Izbor kola koje treba postaviti na podlogu za prikaz strukture obavlja se postavljanjem strelice pokazivača (*mouse pointer*) na odabranu stavku menija *L.Tree* (slika 8–levo) ili sliku menija *L.Images* (slika 8–sredina), i aktiviranjem levog tastera miša izabrano kolo pojavljuje se na podlozi za prikaz strukture.



**Slika 8** Biblioteke za izbor kola, tekstualna *L.Tree* i grafička *L.Images* (levo i sredina), i stablo hijerarhijske organizacije modula elektronske digitalne strukture (desno)

U tabeli 1 daje se pregled svih kola koja su na raspolaganju u biblioteci kola. Na raspolaganju se sledeći tipovi kola i modula:

- Standardna kombinaciona kola, sa označenim brojem ulaznih i izlaznih konektora:  
K–NOT, K–AND<sub>i</sub>, K–NAND<sub>i</sub>, K–OR<sub>i</sub>, K–NOR<sub>i</sub>, K–ExOR2, K–ExNOR2,

K-DC<sub>k-m</sub>, K-CDp<sub>k-m</sub>, K-MP<sub>k-m</sub>, K-DP<sub>k-m</sub>,  
K-SHIFT, K-INC-DEC, K-ADD, K-CG4, K-CMP i K-ALU.

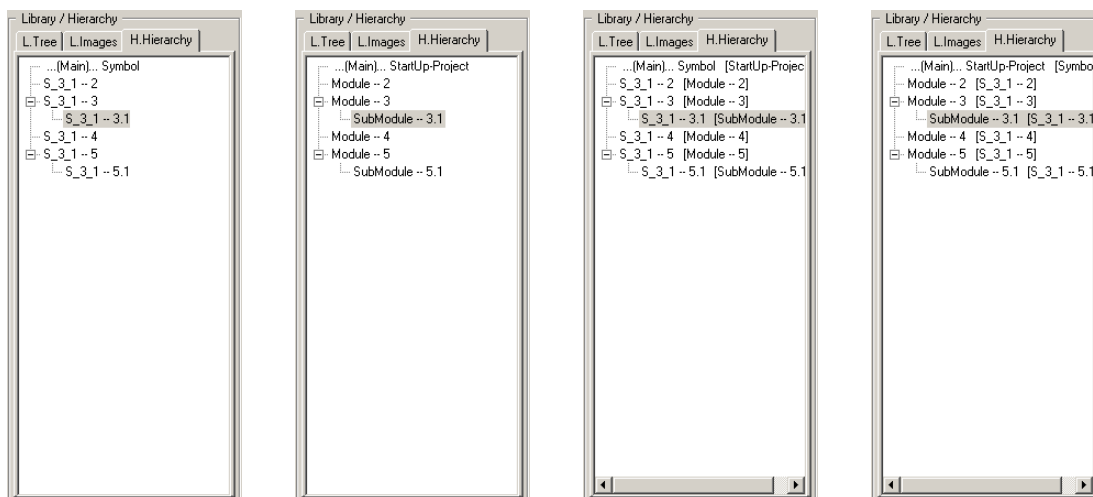
Nakon izbora iz biblioteke, na ulazne konektore ovih kola mogu da se povežu signali širine 1 bit, dok su izlazni signali širine 1 bit. Primenom funkcije *Expand Connectors...* menjaju se širine svih konektora.

- Standardna kombinaciona kola, sa modifikovanim ulaznim konektorima: K-gAND, K-gNAND, K-gOR, K-gNOR, K-gDC<sub>k-m</sub>, K-gCDp<sub>k-m</sub>, K-gMP<sub>k-m</sub> i K-gDP<sub>k-m</sub>. Sva kola iz ove grupe imaju po jedan ulazni konektor na koji može da se poveže signal sa širinom od 1 do 32 bita, dok je izlazni signal uvek širine 1 bit. Nakon izbora iz biblioteke, ulazni konektori ovih kola prihvataju signale širine 16 bita. Primenom funkcije *Expand Connectors...* menjaju se širine samo ulaznih konektora.
- Standardna sekvencijalna kola: S-DFF, S-RSFF, S-TFF, S-JKFF, S-REG, S-ShiftREG, S-COUNTERinc i S-COUNTER. Nakon izbora flip-floпова iz biblioteke kola, na ulazne konektore mogu da se povežu signali širine 1 bit, dok su izlazni signali širine 1 bit. Primenom funkcije *Expand Connectors...* menjaju se širine svih konektora. Nakon izbora registara i brojača iz biblioteke kola, ulazni i izlazni signali za podatke su širine 16 bita, dok su svi kontrolni ulazni i izlazni signali širine 1 bit. Primenom funkcije *Expand Connectors...* menjaju se širine samo ulaznih i izlaznih signala za podatke
- Specijalni moduli:
  - Sp-SubModule – *NonPort Module*.
  - Sp-TriState – trostatički bafer koji omogućava povezivanje signala na magistralu. Isključivo izlazni signal ovog modula može da se poveže na magistralu. Nakon izbora iz biblioteke svi ulazni i izlazni signali su širine 1 bit. Primenom funkcije *Expand Connectors...* menjaju se širine ulaznog i izlaznog signala, dok kontrolni signal zadržava širinu 1 bit.
  - Sp-Generator – generator signala.
  - Sp-Memory – memorijski modul.

K-NOT	K-OR2 K-OR3 K-OR4 K-OR8 K-gOR	K-DC2-4 K-DC3-8 K-DC4-16 K-gDC2-4 K-gDC3-8 K-gDC4-16	K-MP2-1 K-MP4-1 K-MP8-1 K-gMP2-1 K-gMP4-1 K-gMP8-1	K-SHIFT K-INC-DEC K-ADD K-CG4 K-CMP K-ALU	S-DFF S-RSFF S-TFF S-JKFF S-REG S-ShiftREG S-COUNTERinc S-COUNTER
K-AND2 K-AND3 K-AND4 K-AND8 K-gAND	K-NOR2 K-NOR3 K-NOR4 K-NOR8 K-gNOR	K-CDp4-2 K-CDp8-3 K-CDp16-4 K-gCDp4-2 K-gCDp8-3 K-gCDp16-4	K-DP1-2 K-DP1-4 K-DP1-8 K-gDP1-2 K-gDP1-4 K-gDP1-8		Sp-SubModule Sp-TriState Sp-Generator Sp-Memory
K-NAND2 K-NAND3 K-NAND4 K-NAND8 K-gNAND	K-ExOR2 K-ExNOR2				

**Tabela 1** Pregled komponenti koje su na raspolaganju u biblioteci za izbor kola

Na stablu za prikaz hijerarhijske strukture modula elektronske digitalne strukture (slika 8-desno i slika 9) može da se vidi organizacija i raspored raspoloživih modula, a može i da se izabere modul koji će se prikazati na podlozi za prikaz strukture. Izbor modula koji treba da se prikaže obavlja se postavljanjem strelice pokazivača (*mouse pointer*) na odabranu stavku stabla *H.Hierarchy* (slika 8-desno i slika 9), i aktiviranjem levog tastera miša izabrani modul pojavljuje se na podlozi za prikaz strukture. Aktiviranjem odgovarajućih funkcija glavnog menija (slika 4) izgled stabla za prikaz hijerarhijske strukture modula može da se menja, kao što je prikazano na slici 9.



**Slika 9** Stablo hijerarhijske organizacije modula elektronske digitalne strukture – mogući načina prikaza

#### 1.4.4 Navigacija i pregled rezultata rada simulatora

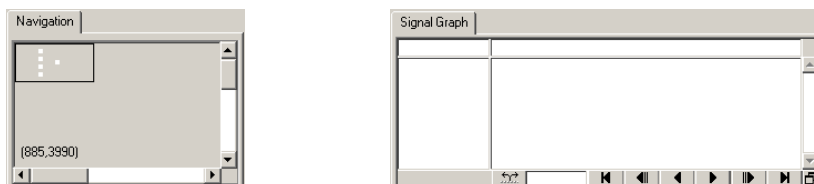
Na slici 10 prikazani su delovi osnovnog prozora za pregled i promenu pozicije na podlozi za prikaz strukture (*Navigation*, levo), kao i pregled rezultata rada simulatora (*Signal Graph*, desno).

U delu za pregled i promenu pozicije na podlozi za prikaz strukture (*Navigation*, slika 10–levo) na raspolaganju su:

- podloga za prikaz trenutnog položaja sa klizačima za promenu položaja (*Scrollbars*)
- trenutni položaj prozora kroz koji se vidi deo podloge za prikaz strukture (pravougaonik na podlozi za prikaz trenutnog položaja)
- stilizovana šema elektronske digitalne strukture sa cele podloge za prikaz strukture (bela polja na podlozi za prikaz trenutnog položaja)
- numeričke vrednosti koordinata trenutne pozicije pokazivača na podlozi za prikaz strukture ( $x$  i  $y$  koordinate unutar zagrada na podlozi za prikaz trenutnog položaja)

Promena položaja na podlozi za prikaz strukture može da se ostvari na dva načina:

- korišćenjem klizača (*Scrollbars*), pozicija se pomera po horizontali i vertikali
- aktiviranjem levog tastera miša, kada je pokazivač (*mouse pointer*) iznad odabrane tačke na podlozi za prikaz trenutnog položaja, pozicija vidljivog dela podloge za prikaz strukture se postavlja na odabrano mesto, srazmerno odnosu veličina podloge za prikaz strukture i podloge za prikaz pozicije na podlozi za prikaz strukture



**Slika 10** Pregled i promena pozicije na podlozi za prikaz strukture (*Navigation*, levo) i pregled rezultata rada simulatora (*Signal Graph*, desno)

Na slikama 10–desno, 11 i 12 prikazan je deo za pregled rezultata rada simulatora, odnosno vremenskih oblika odabranih signala i aktivnih signala takta (u daljem tekstu *vremenski dijagram*). Na slikama 10–desno i 11 vremenski dijagram je prikazan u osnovnom obliku (*minimize mode*), kada se nalazi u donjem srednjem delu osnovnog prozora softverskog paketa (slika 1, strana 11). Ovo mesto je rezervisano za vremenski dijagram, i promene vremenskih

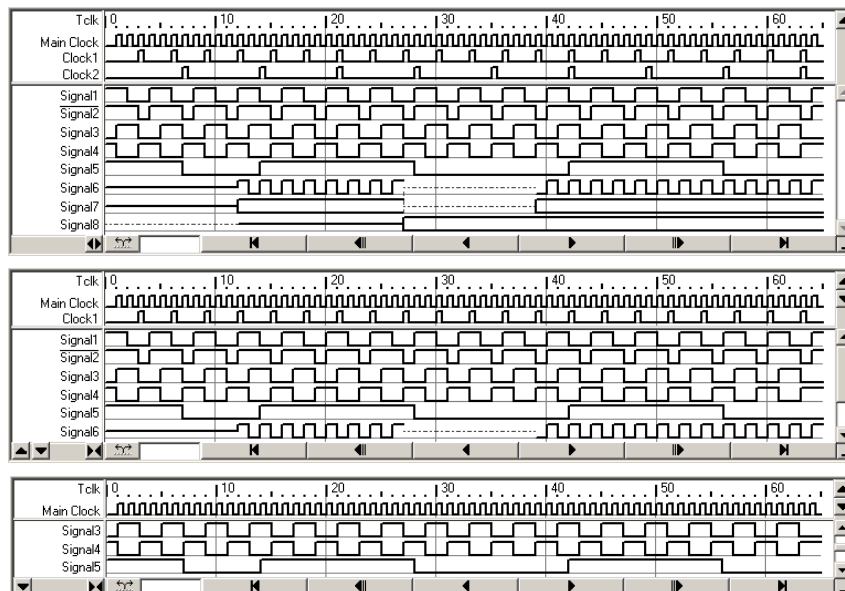
oblika odabranih signala i aktivnih signala takta mogu da se posmatraju i tokom izvršavanja jednog ili više taktova simulacije.



**Slika 11** Pregled rezultata rada simulatora – vremenski oblici odabranih signala (*minimize mode*)

Obezbeđena je i mogućnost šireg prikaza vremenskog dijagrama (*maximize mode*), kada se dijagram prikazuje u prostoru za prikaz strukture, u gornjem levom uglu. I u ovom režimu prikaza vremenskog dijagrama moguće je aktivirati funkcije simulatora, ali veći deo podloge za prikaz strukture nije vidljiv, i nije moguće pratiti sve promene na podlozi za prikaz strukture. Na slici 12 dati su mogući širi prikazi vremenskog dijagrama, kada su vidljivi:

- svi odabrani signali i svi aktivni signali takta (slika 12–gore)
- samo neki od zadatih signala i neki od aktivnih signala takta (slike 12–sredina i 12–dole)



**Slika 12** Pregled rezultata rada simulatora – vremenski oblici odabranih signala (*maximize mode*)

Skup funkcija za rad sa vremenskim oblicima odabranog skupa signala i aktivnih signala takta može da se grupiše na sledeći način:

- podešavanje pozicije vremenskog dijagrama – prelazak iz *minimize mode* u *maximize mode*, i obrnuto, obavlja se aktiviranjem tastera u donjem desnom uglu prostora za prikaz vremenskog dijagrama (slike 11 i 12).
- podešavanje izgleda vremenskog dijagrama – kada je dijagram u *minimize mode* obliku nije moguće da se modifikuje izgled vremenskog dijagrama, dok je u *maximize mode* obliku obezbeđen skup tastera za podešavanje izgleda (slika 12). Tasterima u donjem levom uglu podešava se broj signala koji mogu da se vide, tasterima u gornjem desnom uglu podešava se broj signala takta koji mogu da se vide, dok vertikalnim klizačem (*vertical scroll bar*) može da se odabere skup signala vidljivih u prostoru za prikaz vremenskih oblika signala, kada je odabrano da se prikazuje samo deo skupa signala.
- pregled vremenskih oblika – tasterima koji se nalaze neposredno ispod prostora za prikaz vremenskih oblika signala (slike 11 i 12), obezbeđeno je horizontalno pomeranje prikaza vremenskih oblika signala za sve vremenske trenutke *Tclk* koji su na raspolaganju u

strukturi podataka za čuvanje promena odabranih signala (u daljem tekstu *struktura za čuvanje promena signala*). Krajnji levi taster i polje pored njega koriste se za aktiviranje prikaza proizvoljnog signala takta *Tclk* iz strukture za čuvanje promena signala.

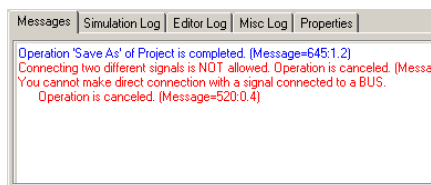
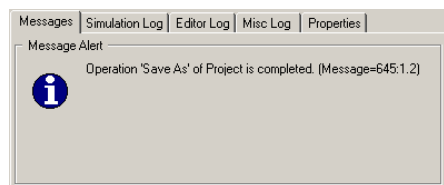
Struktura za čuvanje promena signala može da prihvati promene do 24 odabrana signala za maksimalno 8000 ciklusa pojavljivanja uzlazne ivice glavnog signala takta *Main Clock*. Tasterima za navigaciju prikaza sačuvanih podataka nije moguće da se obezbedi prikaz promena odabranih signala za vremenske trenutke simulacije van opsega koji je sačuvan u strukturi za čuvanje promena signala.

### 1.4.5 Pregled poruka i sadržaja strukture podataka

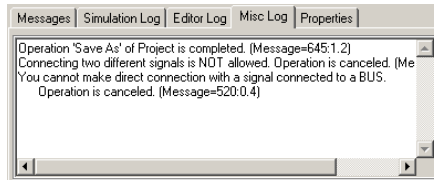
U donjem desnom uglu osnovnog prozora softverskog paketa (slika 1, strana 11) nalazi se prostor rezervisan za sistem informisanja korisnika o porukama koje generiše softverski paket, sistem za arhiviranje poruka, kao i pregled sadržaja strukture podataka softverskog paketa.

#### 1.4.5.1 Pregled poslednjih poruka i arhiviranih poruka

Na slikama 13 i 14 prikazani su izgledi dela za pregled nekoliko poslednjih poruka za informisanje korisnika softverskog paketa (*Messages*), kao i dela za arhiviranje i prikaz arhiviranih poruka (*Misc Log*), respektivno.



Slika 13 Pregled poslednjih poruka za informisanje korisnika softverskog paketa (*Messages*)



Slika 14 Pregled svih poruka za jednu od kategorija arhiviranih poruka (*Misc Log*)

Komunikacija softverskog paketa *IGoVSoEDS* i korisnika obavlja se na dva načina, u zavisnosti od toga da li je neophodan neposredan odgovor korisnika ili odgovor korisnika nije neophodan:

- ukoliko je odgovor obavezan, pojavljuje se standardni prozor za prikaz poruke i odgovarajući skup tastera sa ponuđenim odgovorima (*MS Windows Message Box*). Zaustavlja se rad softverskog paketa i čeka se na odgovor korisnika. Aktiviranjem odgovarajućeg tastera korisnik obrađuje postavljeni zahtev i softverski paket nastavlja sa radom. Odgovarajuća poruka sa opisom postavljenog zahteva i sa opisom akcije korisnika pojavljuje se u polju za prikaz poruka (slika 13–desno).
- ukoliko se ne očekuje neposredan odgovor, ne pojavljuje se prozor za dijalog, već se u polju za prikaz poruka (slika 13–desno) daje odgovarajući opis situacije i akcije koju softverski paket preduzima. Na ovaj način, rad korisnika nije prekinut sa zahtevom da se da odgovor, i ostavlja se korisniku mogućnost da bira stepen pažnje koji će ukazati prikazanoj poruci.

Na vrhu polja za prikaz poruka (slika 13–desno) nalazi se poslednja poruka softverskog paketa, a sve prethodne poruke pomeraju se na dole. Ovo polje nema namenu da arhivira veliki broj poruka, već sadrži samo nekoliko poslednjih poruka. Prikazane poruke imaju identifikatore koji jednoznačno ukazuju na module softverskog paketa koji generišu poruke (slika 13, *Message=645:1.2*), gde identifikator ukazuje na modul, proceduru, i mesto u proceduri iz koje je generisana poruka (slika 13, *600, 45 i 1.2*, respektivno).

Neposredno pre dodavanja poruke u polje za prikaz poruke (slika 13–desno), na trenutak se pojavljuje poruka sa odgovarajućim simbolom koji ukazuje na kategoriju poruke koja se prikazuje (*Message Alert*, slika 13–levo). Raspoložive kategorije su: informativna poruka, upozoravajuća poruka i poruka koja ukazuje na veće probleme u radu softverskog paketa. U zavisnosti od kategorije, poruka se u polju za prikaz poruke ispisuje odgovarajućom bojom (plava, tamno žuta i crvena, respektivno). Ovaj tip poruke ima namenu da upozori korisnika da je poruka generisana, vremenski period u kome je prikazana je ispod jedne sekunde, i njen sadržaj se prenosi u polje za prikaz poruka (slika 13–desno). Za arhiviranje poruka prikazanih u polju za prikaz poruka (slika 13–desno) koriste se tri tekst polja (slika 14, *Simulation Log*, *Editor Log* i *Misc Log*). Sve poruke koje se pojavljuju u polju za prikaz poruka smeštaju se u jedno od raspoloživih polja za arhiviranje poruka, u zavisnosti od tipa poruke. Svako od polja za arhiviranje može da prihvati po nekoliko stotina poruka i ukoliko se ovaj broj prekorači najstarije poruke se brišu.

#### 1.4.5.2 Pregled sadržaja strukture podataka

Struktura podataka u kojoj se čuva opis elektronske digitalne strukture i svi podaci neophodni za uspešnu simulaciju rada strukture (u daljem tekstu *struktura podataka IGoVSoEDS*) je jedna od osnovnih komponenti softverskog paketa *IGoVSoEDS*. Sve što korisnik uradi pri kreiranju ili modifikaciji elektronske digitalne strukture, i svi rezultati rada funkcija softverskog paketa imaju za posledicu kreiranje, modifikaciju ili brisanje slogova strukture podataka *IGoVSoEDS*.

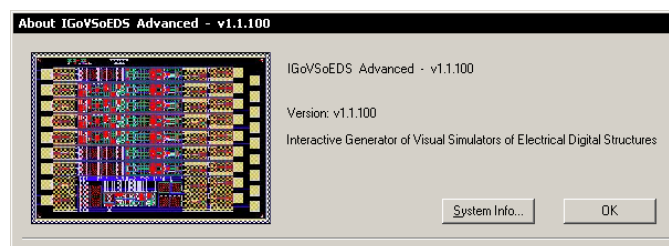
Na slici 15 prikazani su skupovi polja u kojima se daju pregledi sadržaja strukture podataka *IGoVSoEDS* za odabrani objekat (*Module* i *Signal*, respektivno). Pored prikazanih pregleda, na raspolaganju su i pregledi za objekte *Screen*, *SubModule* i *Object*. Promena sadržaja polja u pregledu sadržaja strukture podataka nije dozvoljena. Promena sadržaja strukture podataka može da se obavi samo upotrebom funkcija softverskog paketa *IGoVSoEDS*.



**Slika 15** Pregled sadržaja strukture podataka u kojoj se čuva opis elektronske digitalne strukture za kategorije *Module* i *Object, Signal...*, respektivno

#### 1.4.6 Sistem za pomoć pri korišćenju softverskog paketa

U okviru softverskog paketa *IGoVSoEDS* razvijen je i sistem za pomoć korisniku pri korišćenju softverskog sistema. Na slikama 16 i 17 prikazane su dve osnovne komponente realizovanog sistema za pomoć.



Slika 16 About prozor

*About* prozor (slika 16) prikazuje osnovne informacije o softverskom paketu. Na raspolaganju je i taster za poziv sistemske funkcije za kreiranje izveštaja o konfiguraciji računara na kome se izvršava softverski paket (*System Info...*). *About* prozor se otvara aktiviranjem funkcije *Help/About* koja se nalazi u okviru glavnog menija softverskog paketa. Taster *OK* zatvara *About* prozor.

Sistem za pomoć pri korišćenju softverskog paketa je kreiran kao *HTML* dokument koji sadrži detaljne opise za korišćenje svih komponenti i funkcija softverskog paketa. *HTML* dokument se otvara aktiviranjem funkcije *Help/Help* koja se nalazi u okviru glavnog menija softverskog paketa. Dokument se otvara korišćenjem standardnog *Web* pretraživača koji je prisutan na računaru na kome se izvršava softverski paket.



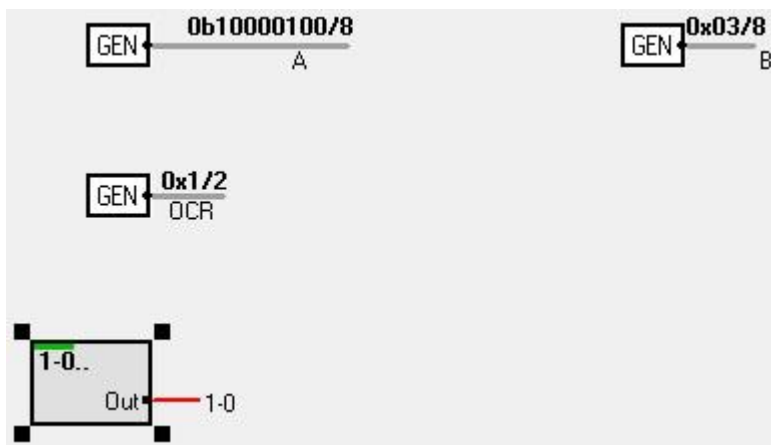
Slika 17 HTML dokument za pomoć pri korišćenju softverskog paketa

### 3. Realizacija simulatora

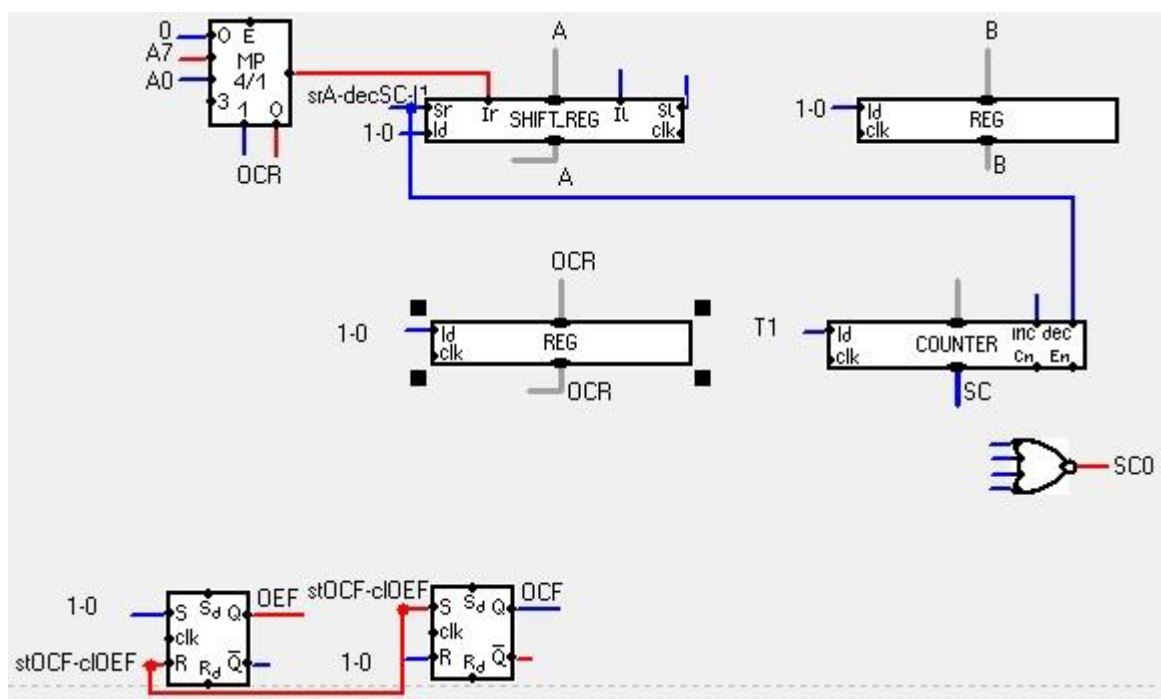
U ovom poglavlju su komentarisane realizacije pojedinih simulatora, problemi koji su se javljali pri realizaciji i izmene koje su napravljene u pojedinim simulatorima u odnosu na zbirku zadataka. Simulatori koji nisu komentarisani su realizovani identično kao u zbirci zadataka.

#### 1.5 Simulator br. 1

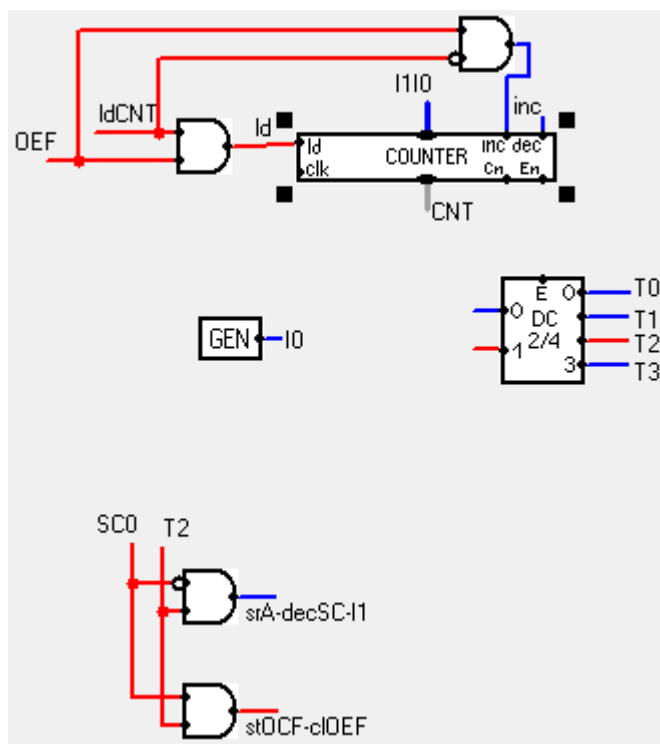
Da bi uređaj P izvršio neku operaciju u njega se najpre upisuje početni sadržaj, a taj upis vrši uređaj A (slika 1). Upis se vrši u registre A, B, OCR, flip-flop OEF se setuje a flip-flop OCF resetuje. Modul na slici 1 označen sa 1-0 samo u prvom taktu na svom izlazu ima jedinicu a u svim narednim taktovima ima nulu. Njegova funkcija je da u početnom trenutku omogući upis podataka u registre A, B, OCR (odgovarajući load signali) i ispravno postavi flip-flobove (S ulaz za OEF i R ulaz za OCF flip-flop). Operaciona i upravljačka jedinica uređaja P su realizovane kao u zbirci (slika 2 i slika 3).



Slika 1 Simulator br. 1, uređaj A



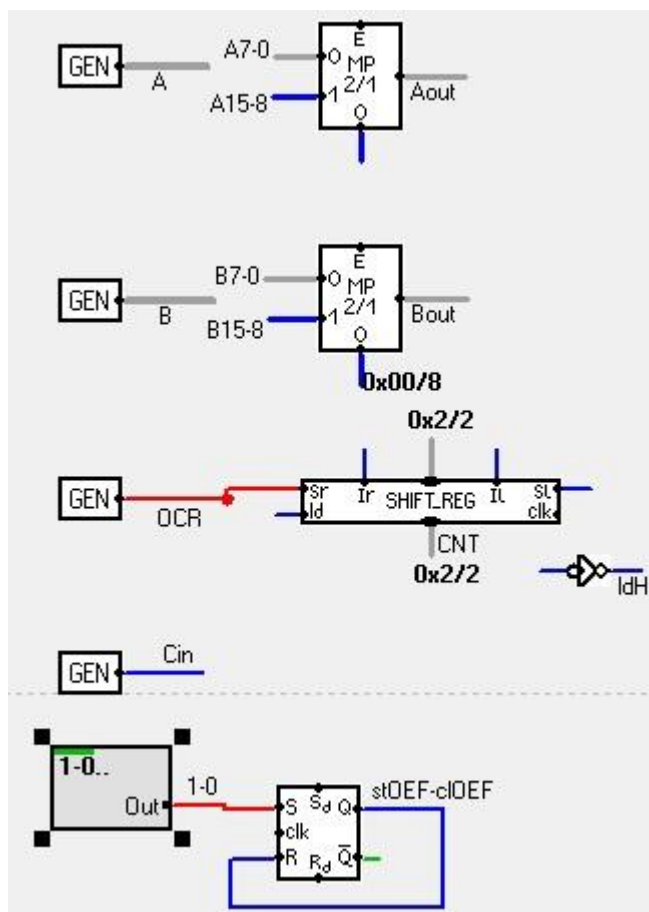
Slika 2 Simulator br. 1, operaciona jedinica



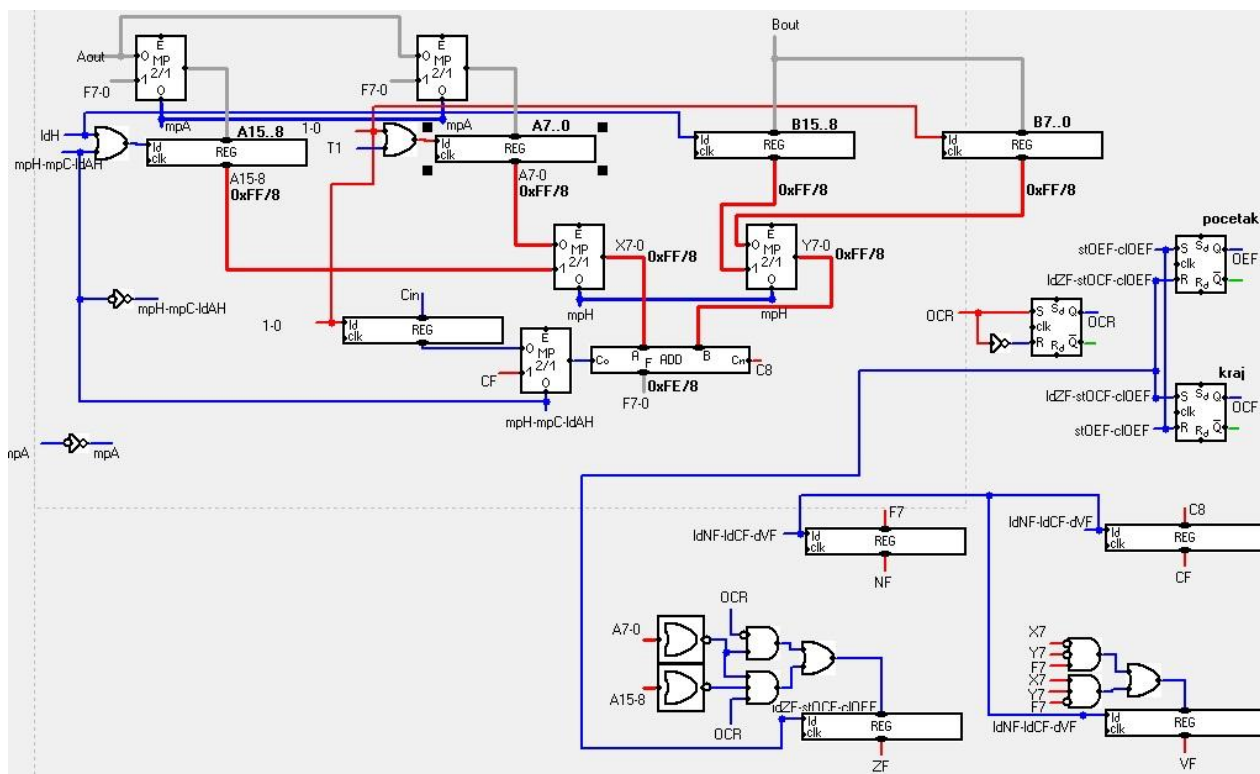
**Slika 3** Simulator br. 1, upravljačka jedinica

## 1.6 Simulator br. 2

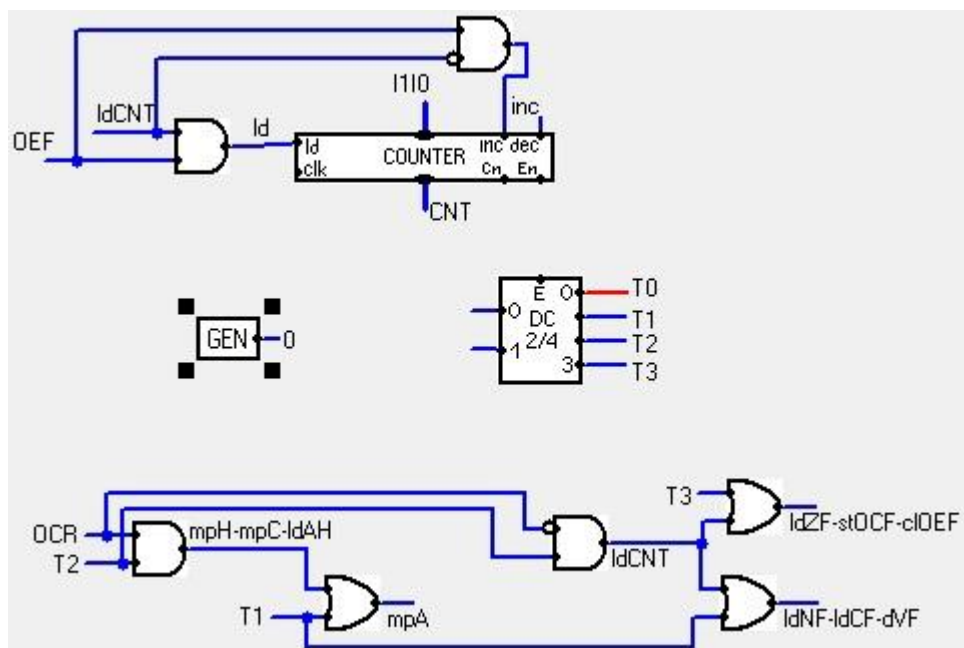
Uređaj A (slika 4) u prvom taktu se vrši upis početnog sadržaja u registre A (nižih 8-bita), B (viših 8-bita), CIN i OCR. U drugom taktu se vrši upis u registre A (viših 8-bita), B (viših 8-bita) i uređaj P startuje upisom vrednosti 1 u flip-flop OEF. Uređaj P je realizovana kao u zbirci (slika 5 i slika 6).



Slika 4 Simulator br. 2, uređaj A



Slika 5 Simulator br. 2, operaciona jedinica

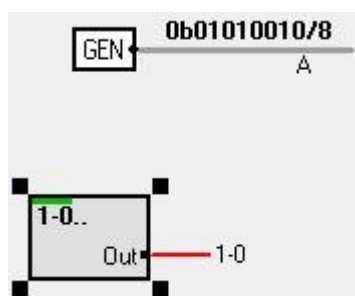


Slika 6 Simulator br. 2, upravljačka jedinica

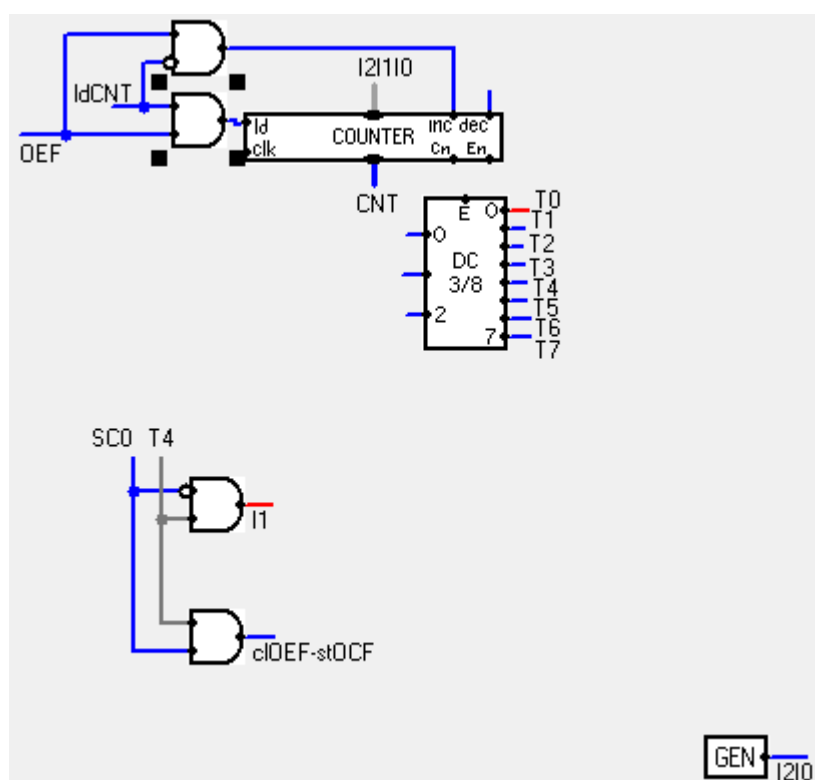
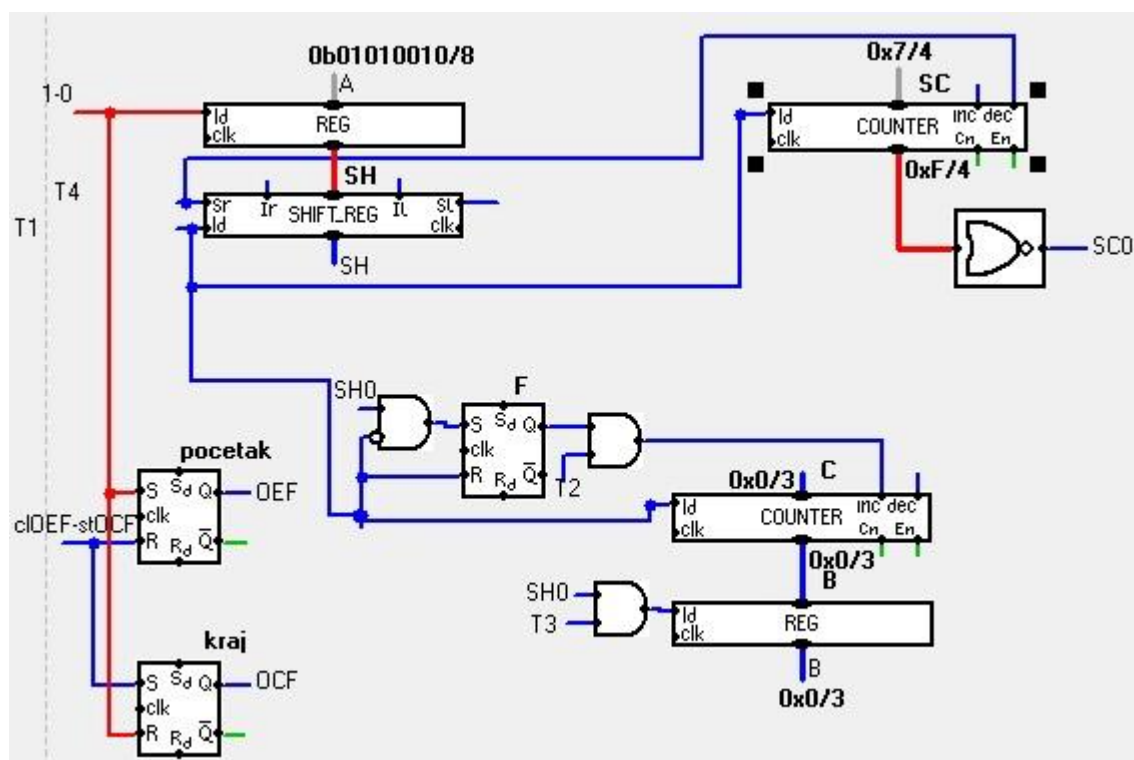
## 1.7 Simulator br. 4

Uređaj A (slika 7) upisuje početni sadržaj u registar A a zatim startuje uređaj P upisivanjem vrednosti 1 u flip-flop OEF.

U uređaju P su napravljene neke izmene. U operacionoj jedinici (slika 8), ulaz u brojač SC je 7 jer je toliko puta potrebno pomeriti sadržaj registra SH. Upravljačka jedinica (slika 9) treba da generiše signale srSH i decSC u taktu T4 a ne u taktu T3. Suviše je rano da se signal srSH pojavi u taktu T3 jer bi se u tom slučaju signal SH<sub>0</sub> sa novom vrednošću pojavio na izlazu flip-flopa F već u taktu T2 naredne iteracije, što bi dovelo do inkrementiranja brojača C prilikom prvog pojavljivanja jedinice. A to nije cilj, cilj je inkrementirati brojač C tek u narednom taktu (prva jedinica se ne broji nego samo broj pozicija posle nje). Signal decSC takođe treba generisati u taktu T4 ako je u SC upisano 7.



Slika 7 Simulator br. 4, uređaj A

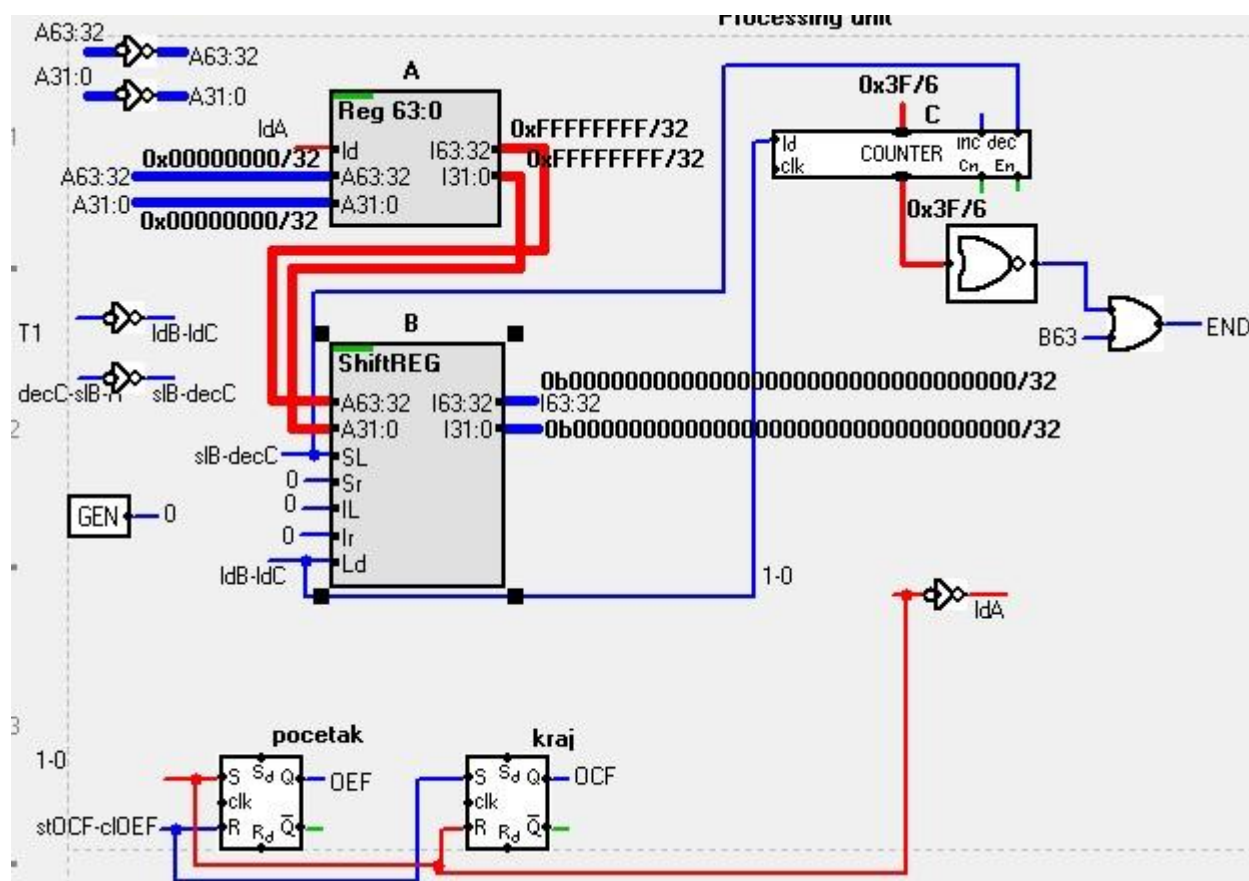


## 1.8 Simulator br. 5

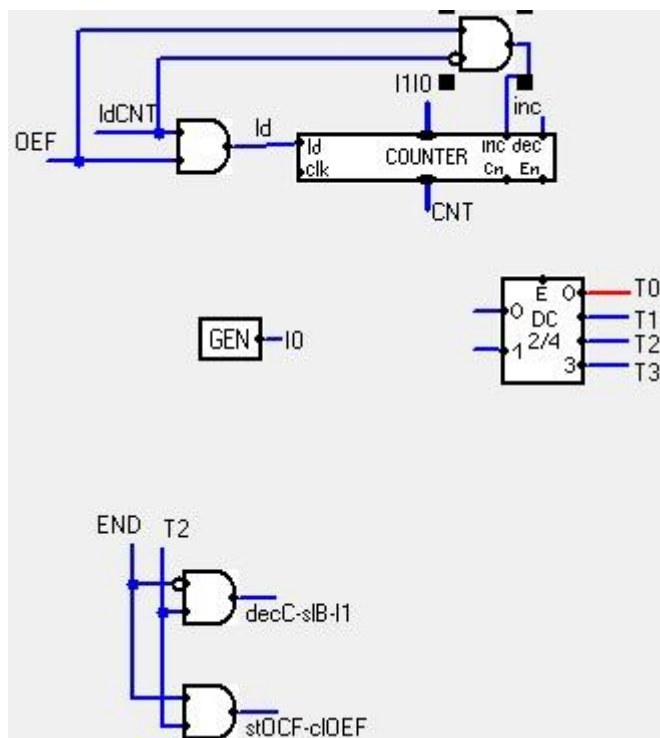
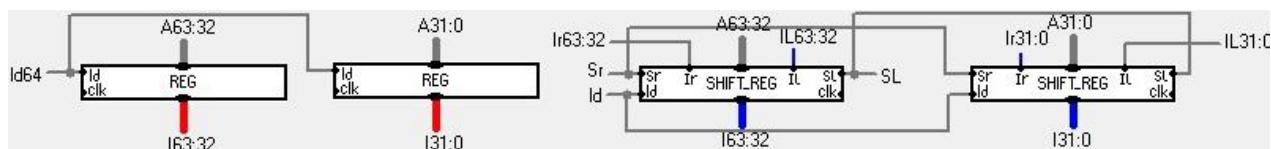
Uređaj A (slika 10) služi za inicijalizaciju i startovanje uređaja P. U softverskom paketu IGoVSoEDS ne postoji podrška za 64-bitne registre, a oni su potrebni za realizaciju operacione jedinice. Zbog toga sam 64-bitni registar i 64-bitni pomerački registar realizovao kao posebne module. U svakom tom modulu se uz pomoć dva 32-bitna registra, odnosno dva 32-bitna pomeračka registra realizuje jedan 64-bitni registar, odnosno 64-bitni pomerački registar (slika 12).



Slika 10 Simulator br. 5, uređaj A

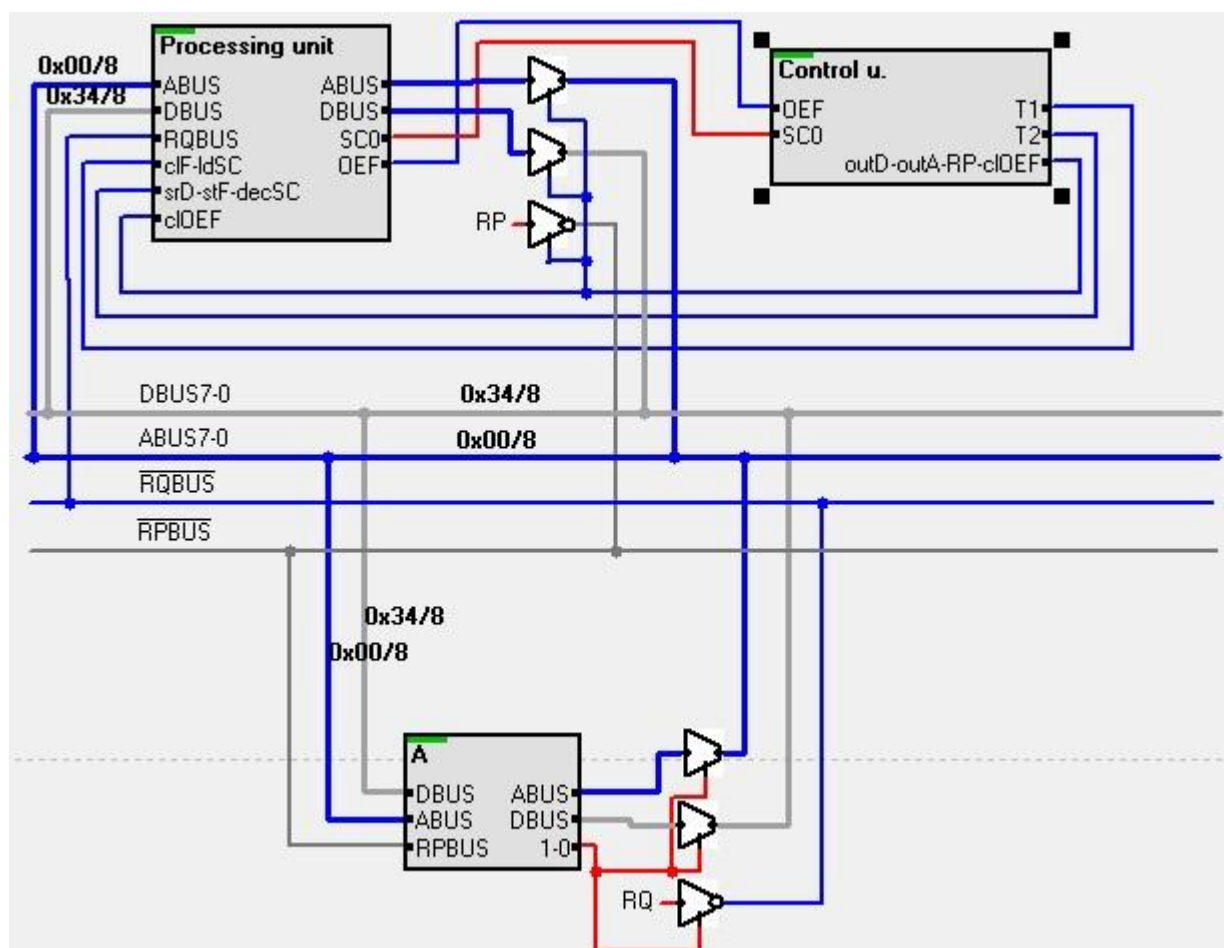


Slika 11 Simulator br. 5, operaciona jedinica



## 1.9 Simulator br. 7

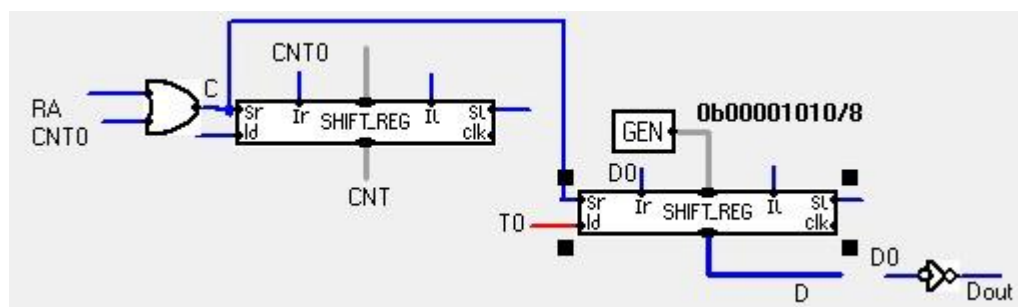
U slučaju simulatora br. 7 komunikacija između uređaja se obavlja preko magistrala A (8-bita), D(8-bita), RP(1 bit), RQ(1 bit). Svi portovi na modulima koji su ulazni mogu se direktno povezati na magistralu. Izlazni signali svih uređaja koji se povezuju na magistralu povezuju se preko trostatičkih bafera (Slika 14).



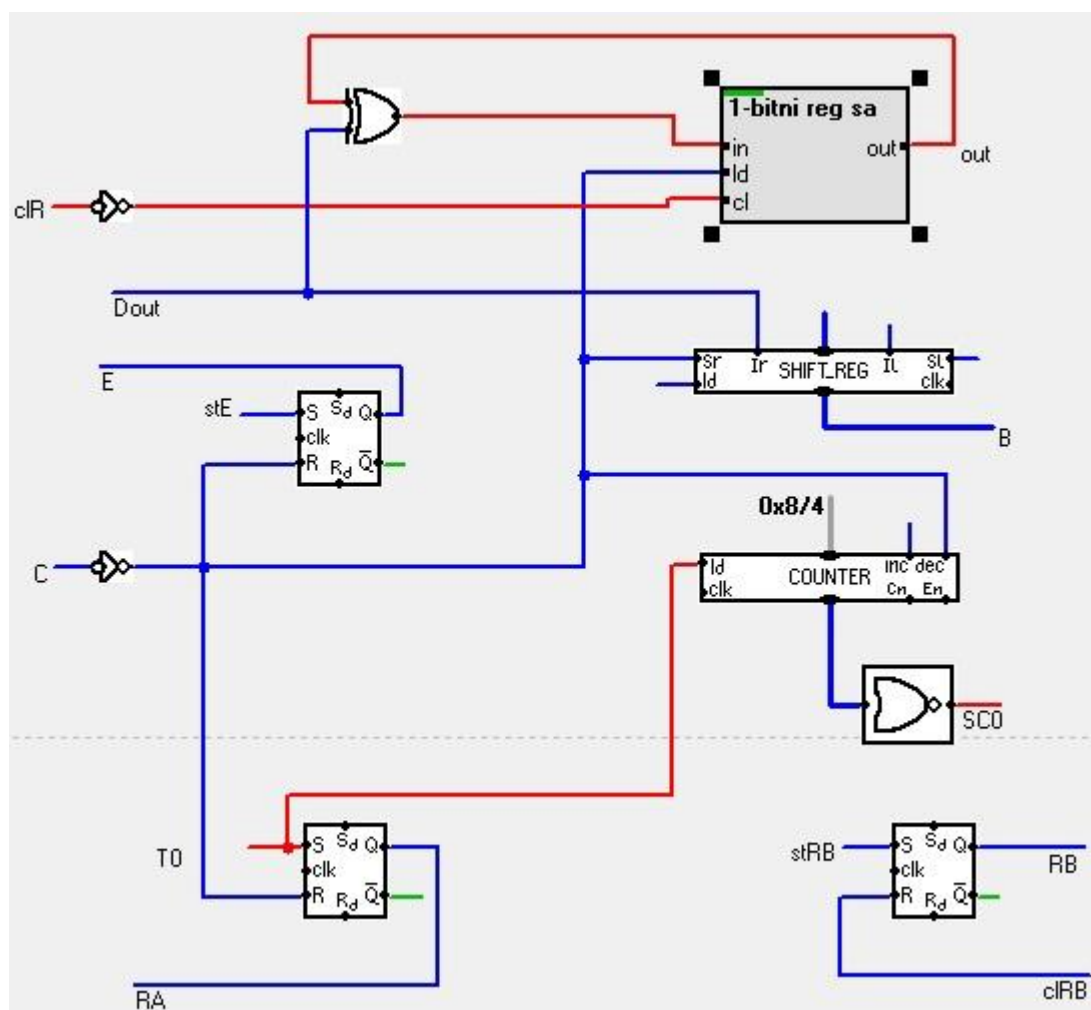
Slika 14 Simulator br. 7

## 1.10 Simulator br. 17

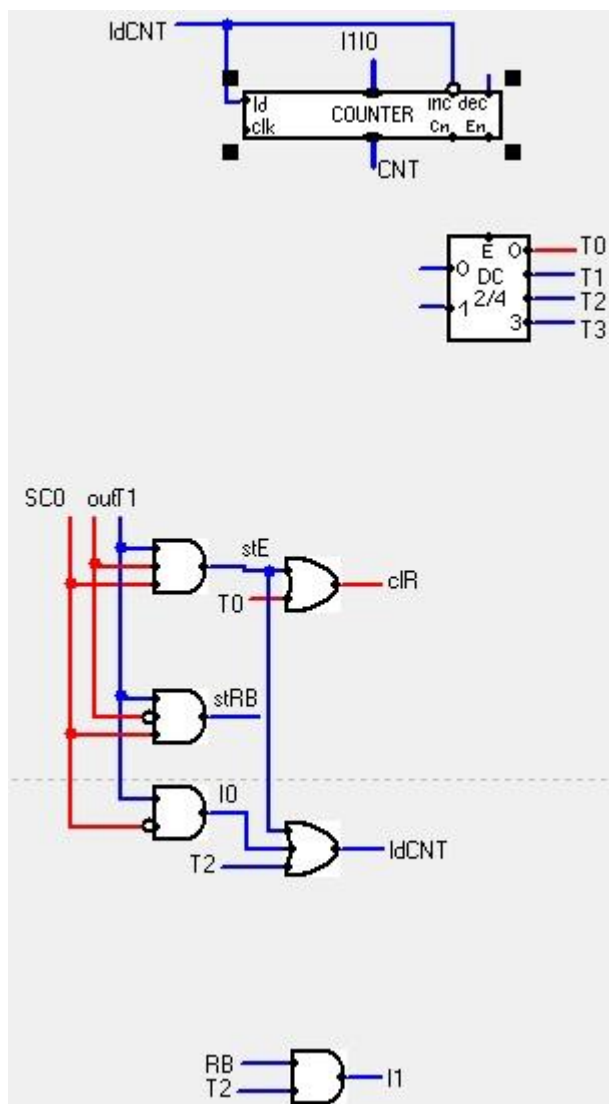
Brojač SC operacione jedinice (Slika 16) je realizovan kao 4-bitni i u njega je upisana vrednost 8, umesto trobitnog brojača sa početnom vrednošću 7 kao što stoji na slici 1.18.a u zbirci zadataka. Zbog nedostatka u standardnoj biblioteci jednobitni registar sa clear ulazom je realizovan kao poseban modul, uz pomoć multipleksera, ILI kola i jednobitnog registra bez clear ulaza. U slučaju da je nastala greška prilikom slanja podatka od uređaja A do uređaja P, R=1, na dijagramima tokova (slika 1.17.b zbirka zadataka) je potrebno vratiti tok na takt T0 umesto na takt T1 kako stoji u zbirci zadataka.



Slika 15 Simulator br. 17, uređaj A



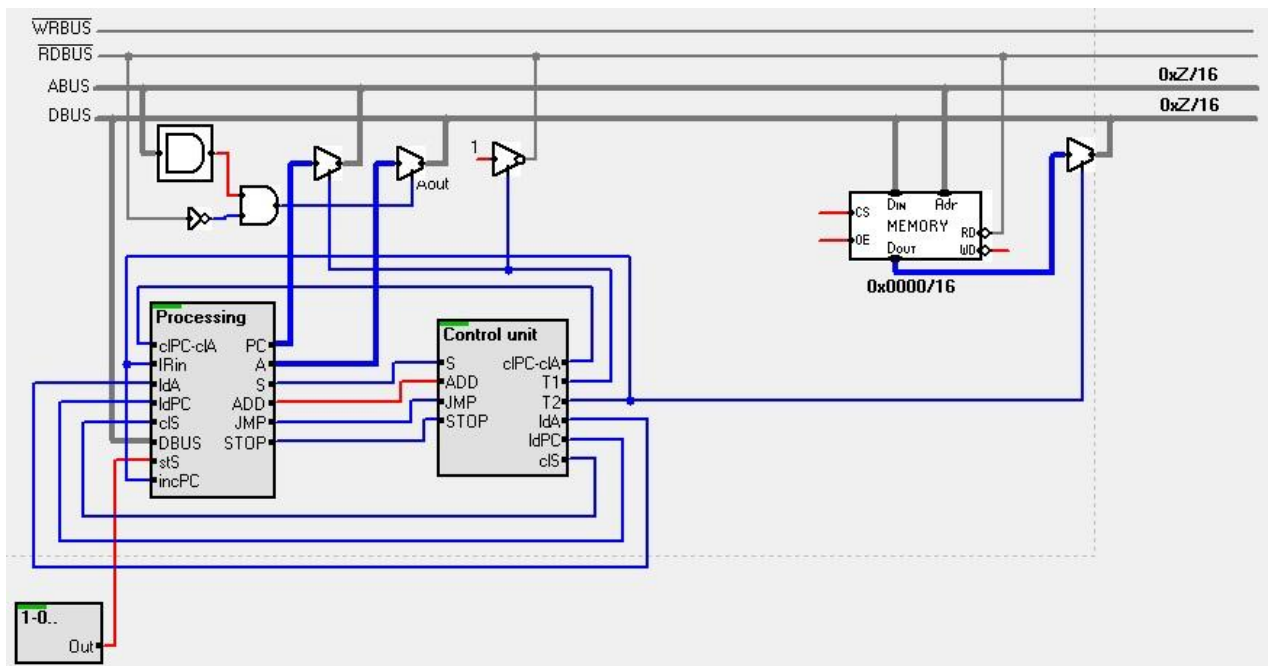
Slika 16 Simulator br. 17, operaciona jedinica



Slika 17 Simulator br. 17, operaciona jedinica

### 1.11 Simulator br. 18

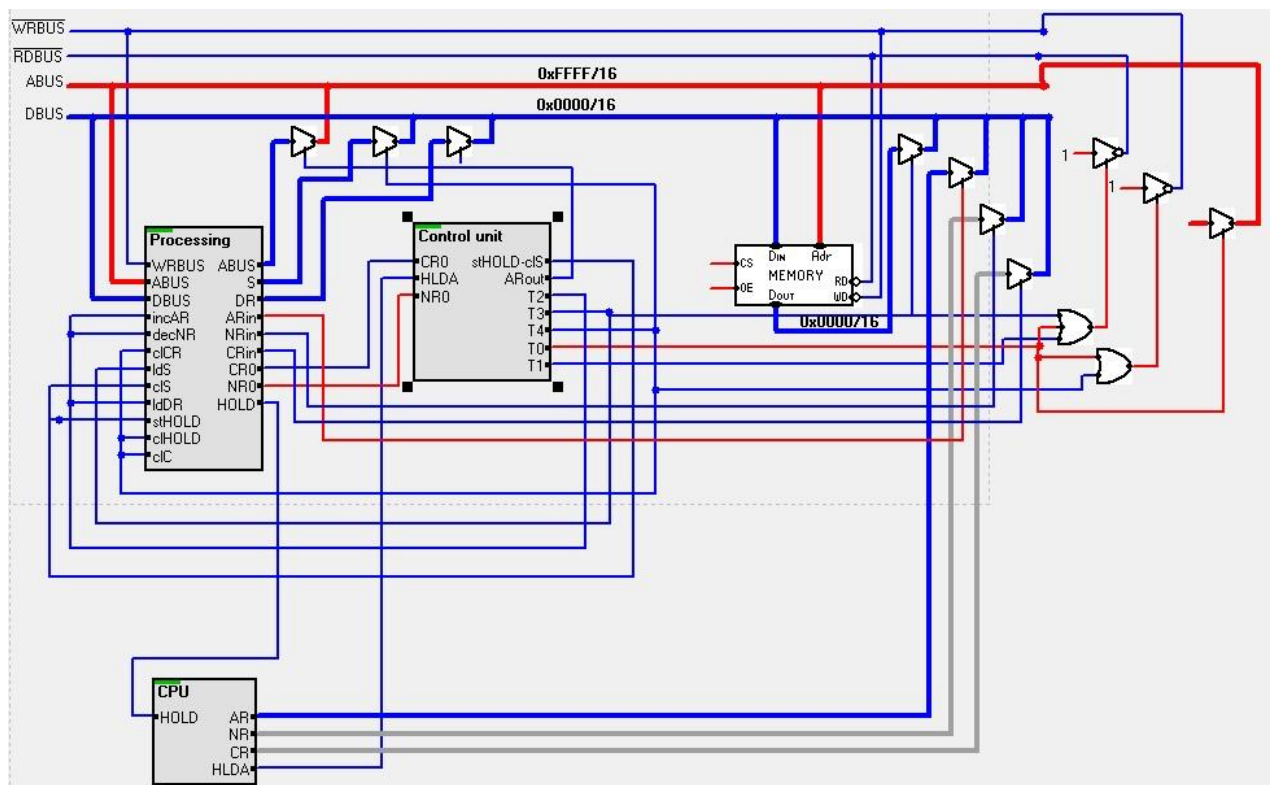
Po uslovima zadatka br. 18 (Slika 18) potrebno je da ciklusi čitanja i upisa u memoriju traju po jednu periodu signala takta. U softverskom paketu IGVSEDES ne postoji memorija sa takvim mogućnostima. Zbog toga je korišćena memorija za koju je potrebno dva signala takta da bi se izvršilo čitanje podatka. U prvom taktu je potrebno postaviti odgovarajuće signale koji će omogućiti čitanje podatka sa željene lokacije, a u sledećem taktu će ti podaci biti dostupni na izlaznim linijama memorije. Zbog korišćenja takve memorije potrebno je izvršiti određene izmene u dijagramima toka operacione i upravljačke jedinice. Neke signale je potrebno prebaciti u sledeći takt a nekada je potrebno i dodati neke taktove, zbog čega dolazi i do izmene upravljačke jedinice koja generiše te signale. U slučaju simulatora br. 18 signali IRin i incPC su prebačeni u takt T2 a svi signali iz takt T2 su kompletno prebačeni u takt T3. Ovim je omogućeno ispravno čitanje podatka iz memorije.



Slika 18 Simulator br. 18

## 1.12 Simulator br. 19

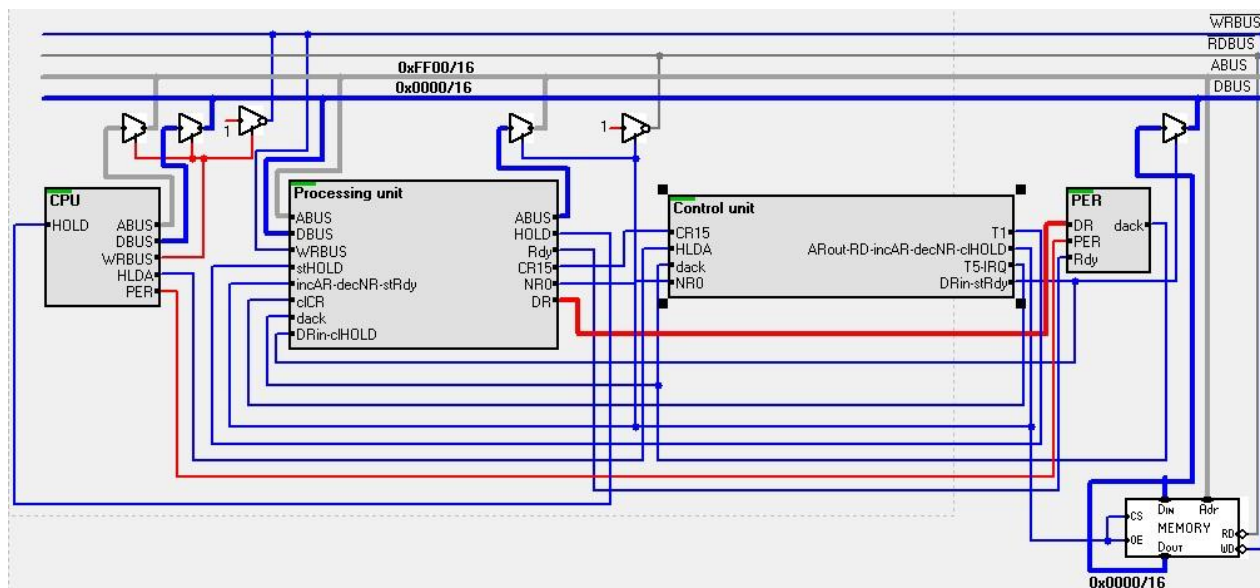
I u ovom slučaju ne postoji odgovarajuća memorija, pa su zbog korišćenja memorije kod koje se čitanje obavlja u dva takta neki signali morali da se generišu ranije. ARout i RD se generišu u taktu T1, signali decNR, incAR i ldDR u taktu T2.



Slika 19 Simulator br. 19

### 1.13 Simulator br. 21

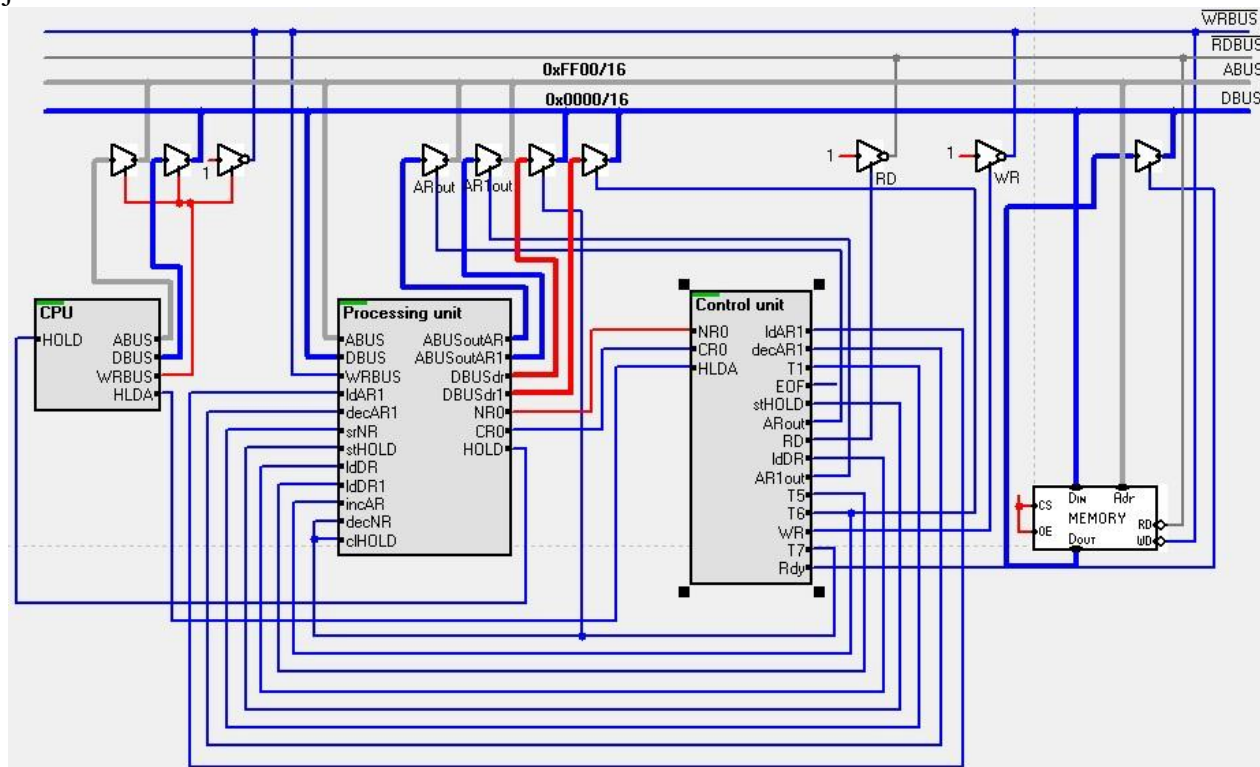
Ovaj simulator je takođe realizovan pomoću memorije kojoj je potrebno dva takta da bi se pročitao podatak i zbog toga je bilo potrebno signale DRin i clHOLD generisati u taktu T3.



Slika 20 Simulator br. 21

### 1.14 Simulator br. 22

I u ovom slučaju postoji problem nedostatka odgovarajuće memorije, pa su zbog korišćenja memorije kod koje se čitanje obavlja u dva takta morale da se naprave neki izmene u upravljačkoj jedinici.

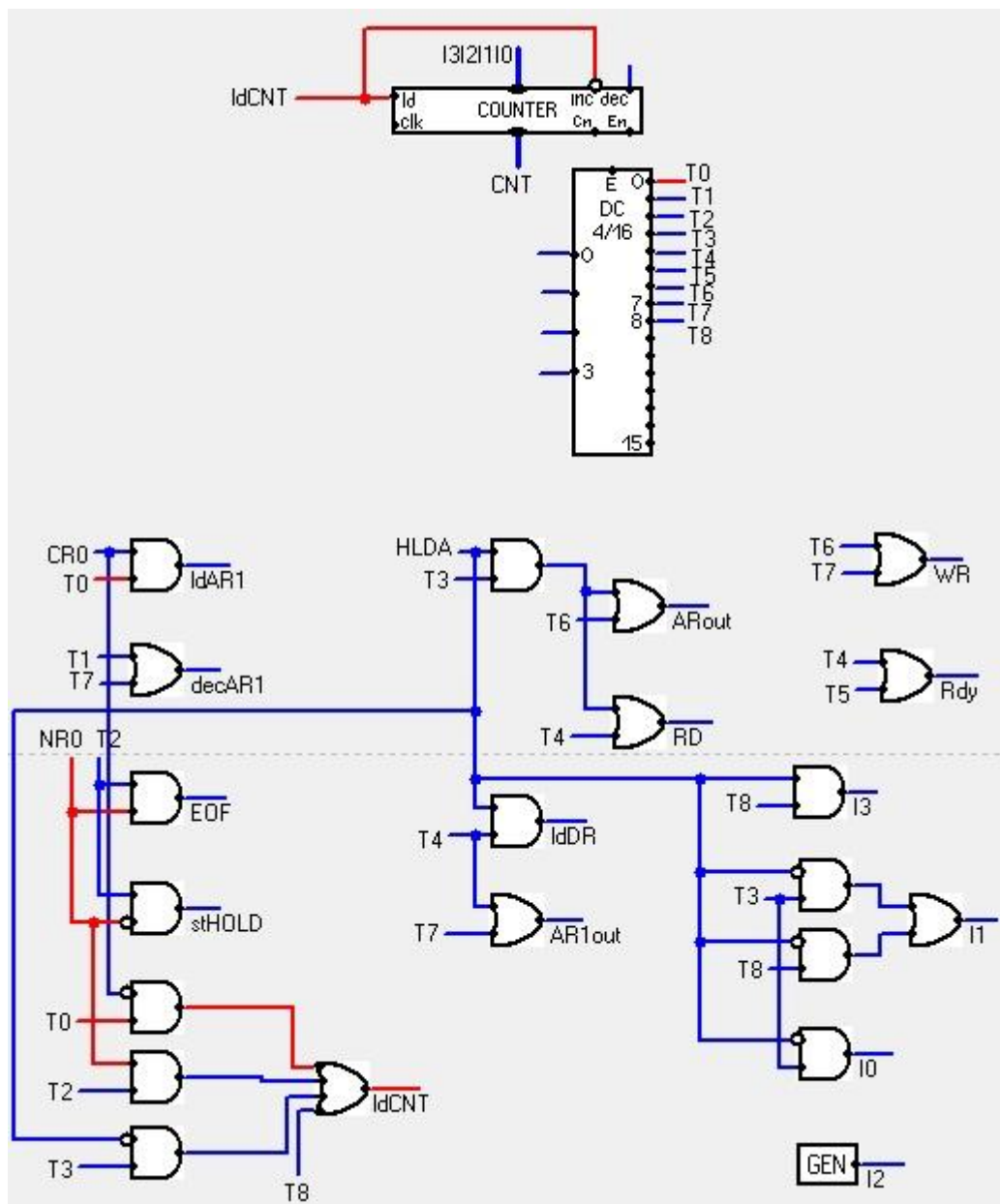


Slika 21 Simulator br. 22

Upravljačka jedinica (Slika 22) je realizovana pomoću brojača i dekodera sa većim brojem ulaza (4 ulaza) u odnosu na to kako je predviđeno postavkom zadatka (3 ulaza) br. 22. Ovo proširenje je bilo potrebno napraviti jer postoji potreba za većim brojem stanja upravljačke jedinice od predviđenih 8. A taj veći broj stanja je potreban je ne postoji mogućnost čitanja podatka iz memorije u jednom taktu.

Signali ldDR i RD iz takta T3 se sada generišu u taktu T4, a signali iz takta T4 ldDR1 i RD se sada generišu u taktu T5. U dijagramu toka upravljačkih signala upravljačke jedinice takt T7 je sada takt T8 i izgleda : ldCNT, if HLDA then I3 else I1.

Sada je došlo i do određene promene prilikom generisanja upravljačkih signala operacione jed.  $decAR1 = T1 + T7$ ,  $ARout = HLDA \cdot T3 + T6$ ,  $ldDR = HLDA \cdot T4$ ,  $AR1out = T4 + T7$ ,  $ldDR1 = T5$ ,  $DR1out = T6$ ,  $WR = T6 + T7$ ,  $incAR = T6$ ,  $DRout = T7$ ,  $decNR = T7$ ,  $clHOLD = T7$ ,  $Rdy = T4 + T5$  i upravljačkih signala upravljačke jedinice  $ldCNT = \overline{CR_0} \cdot T_0 + NR_0 \cdot T_2 + \overline{HLDA} \cdot T_3 + T8$ , I2 nepostoji,  $I1 = \overline{HLDA} \cdot T_3 + \overline{HLDA} \cdot T8$ ,  $I0 = \overline{HLDA} \cdot T3$ ,  $I3 = HLDA \cdot T8$ .

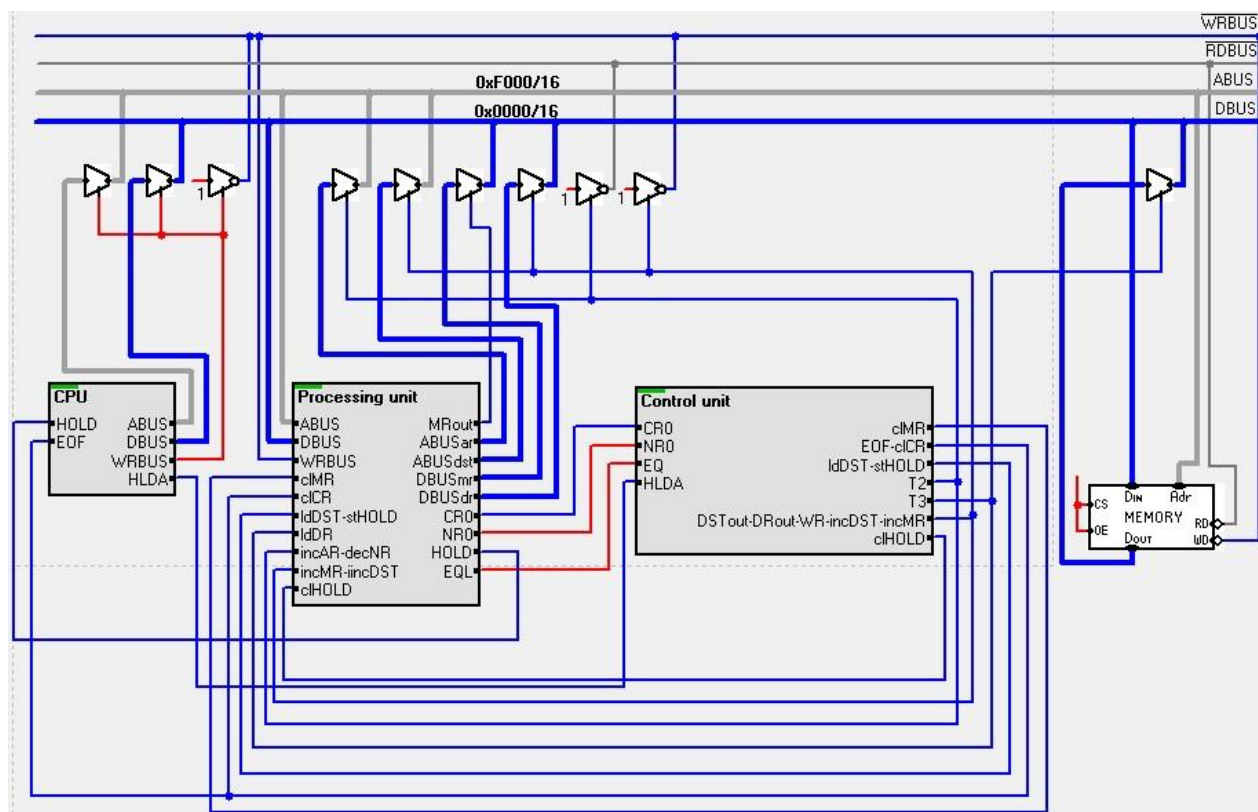


Slika 22 Simulator br. 22, upravljačka jedinica

## 1.15 Simulator br. 23

Zbog korišćenja sporije memorije i u ovom simulatoru su napravljene potrebne izmene. U dijagramu upravljačkih signala upravljačke jedinice dodat je jedan takt posle takta T3. Svi signali iz takta T3 su prebačeni u takt T4 a iz takta T2 u tak T3 je prebačen signal ldDR.

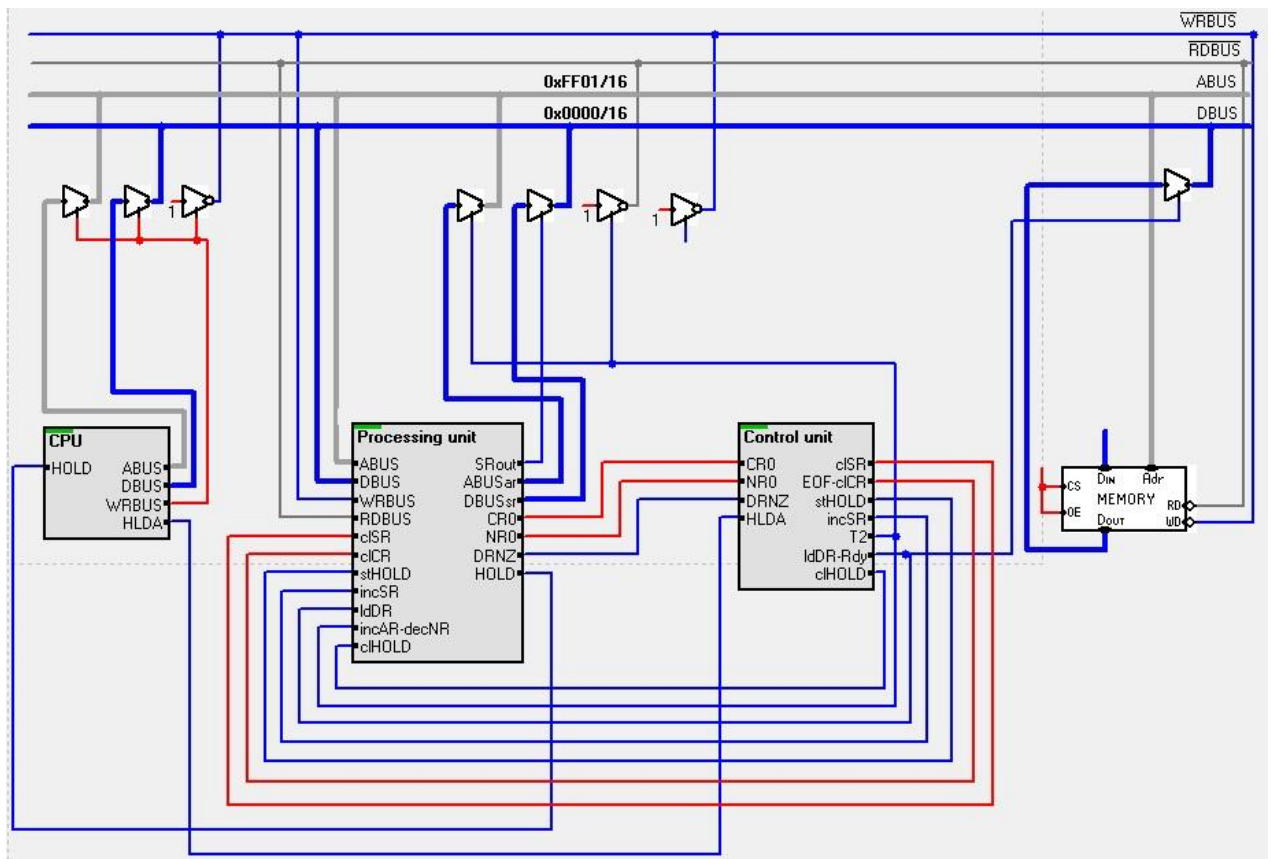
Napravljena je i adekvatna izmena upravljačkih signala upravljačke jedinice:  
 $ldCNT = \overline{CR_0} \cdot T_0 + CR_0 \cdot NR_0 \cdot T_0 + \overline{HLDA} \cdot T_1 + NR_0 \cdot T_4 + T_5$ ,  $I_2 = HLDA \cdot T_5$ ,  $I_1 = NR_0 \cdot T_4$ ,  
 $I_0 = \overline{HLDA} \cdot T_1 + HLDA \cdot T_5$ .



Slika 23 Simulator br. 23

## 1.16 Simulator br. 24

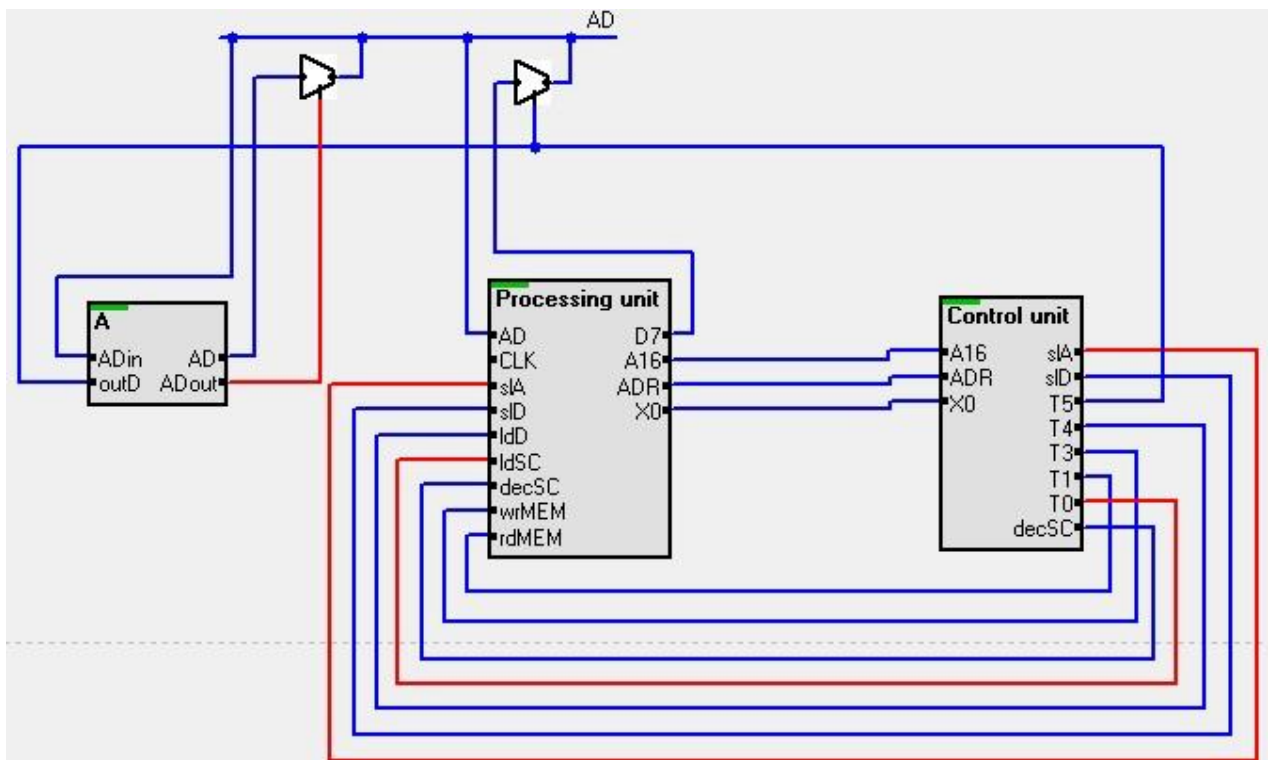
Zbog istih problema sa memorijom u ovom simulatoru (Slika 24) je napravljena slična izmena kao u prethodnom. U dijagramu upravljačkih signala upravljačke jedinice dodat je jedan takt posle takta T2. Signal ldDR iz takta T2 je prebačen u novi takt T3, a stari takt T3 je sada postao takt T4. Sada imamo da je  $ldDR = T_3$ .



Slika 24 Simulator br. 24

## 1.17 Simulator br. 25

Da bi bilo moguće pročitati sadržaj memorije u signalu T4 signal rdMEM je prebačen u takt T1.



Slika 25 Simulator br. 25

## 4. Zaključak

U ovom radu je realizovano dvadeset pet simulatora. Svi su napravljeni u softverskom paketu IGoVSoEDS. Taj softver je obezbedio većinu potrebnih elemenata za realizaciju tih simulatora. Elementi prekidačkih mreža koji su nedostajali su realizovani kao nezavisni moduli i oni su dostupni za neko buduće ponovno korišćenje u nekim novim projektima.

Prema postavci zadataka upravljačke jedinice svih simulatora su realizovane na isti način (korišćenjem brojača koraka i dekodera). Stoga je moguća modifikacija realizovanih simulatora. Modifikacija svakog simulatora drugačijom realizacijom nekog modula. Na primer upravljačku jedinicu je moguće realizovati na neki drugačiji način (pomoću elemenata za kašnjenje, standardnih sekvencijalnih mreža ili mikroprogramski sa jednim i dva tipa mikrooperacija) i jednostavno modifikovati realizovani simulator zamenom postojeće upravljačke jedinice novom. Tom prilikom ne dolazi do bilo kakvih promena u ostatku simulatora, izvrši se samo zamena modula upravljačke jedinice a signali kojima se ona povezuje sa operacionom jedinicom ili magistralom ostaju nepromenjeni.

## 5. Literatura

1. Jovan Đorđević, *“Priručnik iz arhitekture i organizacije računara”*, Beopress štampa, Beograd, 1999.
2. Prof. dr Borivoj Ž. Lazić, *“Logičko projektovanje računara”*, Nauka, Beograd, 1997.
3. Jovan Đorđević, *“Osnovi računarske tehnike II”*, Univerzitet u Beogradu Elektrotehnički fakultet, Beograd, 2006.
4. mr Nenad M. Grbanović, *“Interaktivni generator vizuelnih simulatora elektronskih digitalnih struktura”*, Univerzitet u Beogradu Elektrotehnički fakultet, Beograd, 2007.