PROJEKAT

iz Računarskog projektovanja   
digitalnih integrisanih kola

### TEMA PROJEKTA:

|  |
| --- |
| Multiplekser 32/1 realizovan pomoću multipleksera 8/1, koji je realizovan pomoću multipleksera 4/1 i dodatnih logičkih kola. |

### TEKST PROJEKTA:

|  |
| --- |
| Prikazati funkcionalne tabele multipleksera 4/1, 8/1 i 32/1 i objasniti njihov rad.  Nacrtati kako se realizuje multiplekser 8/1 korišćenjem dva multipleksera 4/1 i potrebnih logičkih kola. Voditi računa da realizovani muliplekser treba da ima svoj signal dozvole, tj. EN signal. Nakon toga nacrtati kako se realizuje multiplekser 32/1 korišćenjem isključivo multipleksera 8/1. Voditi računa da realizovani muliplekser treba da ima svoj signal dozvole, tj. EN signal.  U programskom jeziku VHDL napisati kod kojim se realizuje multiplekser 4/1. (**mux4to1.vhd**)  Korišćenjem multipleksera 4/1 napisati kod za multiplekser 8/1. (**mux8to1.vhd**)  Korišćenjem multipleksera 8/1 napisati kod za multiplekser 32/1. (**mux32to1.vhd**)  Napisati kod za testiranje ispravnosti rada multipleksera 32/1. (**mux32to1\_tb.vhd**)  Testirati i pokazati ispravnost rada multipleksera 32/1.  Korišćenjem alata *Genus Synthesis Solution* programskog paketa *Cadence* projektovati (sintetizovati) šematik multipleksera 32/1u 0.35 µm AMS (C35B4) tehnologiji.  Pomoću alata *Innovus Implementation System* programskog paketa *Cadence* projektovati (generisati) fizičku realizaciju (lejaut) multipleksera 32/1 u 0.35 µm AMS (C35B4) tehnologiji. |

Mentor: Student:

Kristina Nikolić Ognjen Višnjić, EE 217/2020

U Novom Sadu, 23.01.2024.

# 1. Teorijska analiza

Multiplekser je digitalno kolo koje ima više ulaza, ali omogućava prosleđivanje samo jednog od njih na izlazu u zavisnosti od kontrolnog signala. Princip rada multiplekserskog kola se bazira na logičkoj funkciji koja se primenjuje na ulazne signale u skladu sa vrednostima kontrolnog signala. U slučaju multipleksera 32/1, imamo 32 ulaza i koristimo 5-bitni kontrolni signal za odabir željenog ulaza.

Ograničenja multiplekserskih kola uključuju:

* Brzina prenosa podataka: brzina prenosa podataka kroz multiplekser zavisi od unapred definisane brzine rada kola i propagacionog kašnjenja signala kroz njega.
* Veličina kontrolnog signala: broj bitova kontrolnog signala određuje broj ulaza koji se mogu odabrati. Povećanje broja ulaza povećava i širinu kontrolnog signala, što može uticati na brzinu prenosa.
* Osetljivost na šum: kola su osetljiva na elektromagnetni šum i interferenciju, što može uzrokovati netačne rezultate ili gubitak podataka.

Uticaj geometrijskih parametara:

U analizi uticaja geometrijskih parametara, možemo uzeti u obzir širinu i dužinu vodova unutar kola, veličinu tranzistora, kapacitete i otpornike. Na primer, ako analiziramo tranzistor, širina (W) i dužina (L) tranzistora direktno utiču na brzinu prenosa podataka, gde je brzina proporcionalna širini tranzistora i obrnuto proporcionalna dužini tranzistora.

Jednačina koja opisuje ovu zavisnost može izgledati ovako:

(1)

*Vout*​ - izlazni napon,

*V*in​ - ulazni napon,

*V*th​ - prag napona tranzistora.

Ovo je samo jedan primer, a konkretni parametri i jednačine zavise od konkretnih karakteristika kola koje analizirate. Ovaj primer je samo ilustracija kako se može analizirati uticaj geometrijskih parametara na rad kola.

# Multiplekser 4/1:

* Ulazi**: D0 , D1, D2, D3**– 4 ulazna signala.
* Selektor: **S1,S0**- 2-bitni selektorski signal (Sel).
* Izlaz: **Y** – izabrani izlaz

Tabela 1. Funkcionalna tabela multipleksera 4/1

|  |  |
| --- | --- |
| S1S0 | Y |
| 00 | D0 |
| 01 | D1 |
| 10 | D2 |
| 11 | D3 |

Multiplekser 4/1 radi u zavisnosti od vrednosti S1,S0 jedan od ulaza D0, D1, D2, D3 prosleđuje se na izlaz Y.

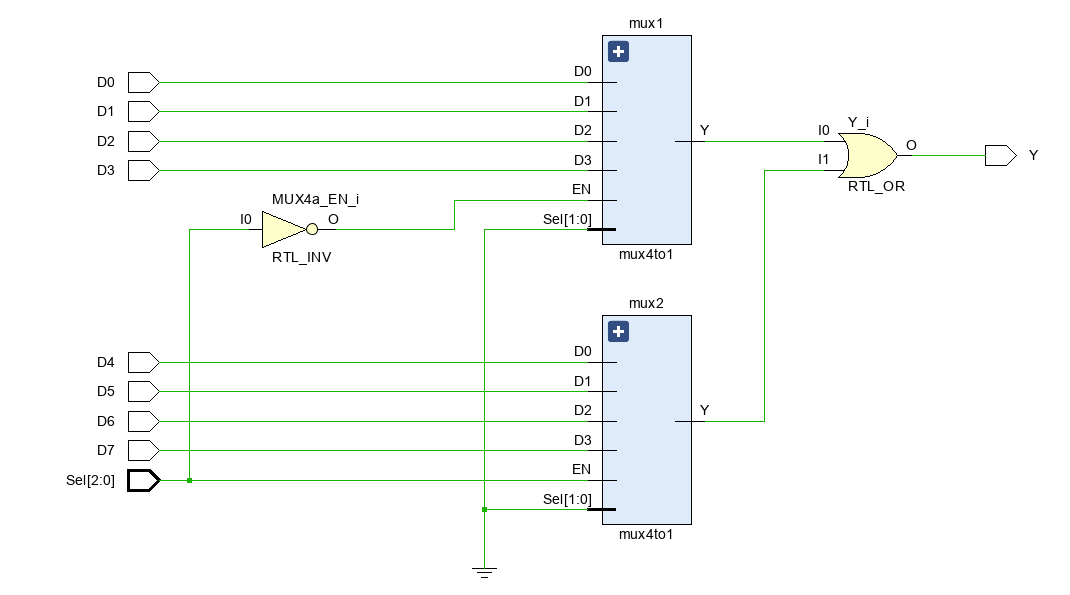
# Multiplekser 8/1:

* Ulazi: **D0, D1, D2, D3,D4, D5, D6, D7**– 8 ulazana signala.
* Selektor: **S2,S1,S0**- 3-bitni selektorski signal (Sel).
* Izlaz: **Y** – izabrani izlaz.

Tabela 2. Funkcionalna tabela multipleksera 8/1

|  |  |
| --- | --- |
| S2S1S0 | Y |
| 000 | D0 |
| 001 | D1 |
| 010 | D2 |
| 011 | D­­3 |
| 100 | D4 |
| 101 | D5 |
| 110 | D6 |
| 111 | D7 |

Multiplekser 8/1 radi u zavisnosti od vrednosti S2,S1,S0 jedan od ulaza D0, D1, D2, D3,D4, D5, D6, D7 bira se kao izlaz Y.



Slika 1. Multiplekser 8/1 realizovan sa samo 2 multipleksera 4/1 i potrebnim logickim kolima,

Sel(2) se koristi za kontrolu koji od multipleksera ce da radi i prosleđuje se na signal dozvole, ukoliko je vrednost 0 na mux2 će biti 0 prosleđena na signal dozvole i on nece raditi, a na mux1 ce zbog invertora biti prosleđena 1 i on će raditi, i obratno. Sel(1 do 0) će biti prosleđeni na standardne Sel ulaze multipleksera 4/1 i imaće normalnu funckionalnost.

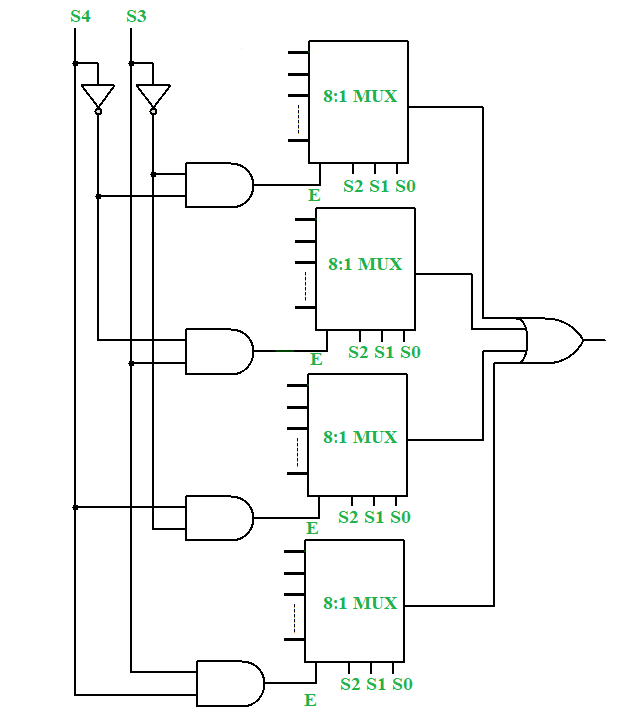
# Multiplekser 32/1:

* Ulazi: **D0, D1, D2, D3,D4, D5, …, D31**– 32 ulazna signala.
* Selektor: **S4,S3,S2,S1,S0**- 5-bitni selektorski signal (Sel).
* Izlaz: **Y** – izabrani izlaz.

Tabela 3. Funkcionalna tabela multipleksera 32/1

|  |  |
| --- | --- |
| S4,S3,S2,S1,S0 | Y |
| 00000 | D0 |
| 00001 | D1 |
| 00010 | D2 |
| 00011 | D3 |
| 00100 | D4 |
| 00101 | D5 |
| 00110 | D6 |
| 00111 | D7 |
| 01000 | D8 |
| 01001 | D9 |
| 01010 | D10 |
| 01011 | D11 |
| 01100 | D12 |
| 01101 | D13 |
| 01110 | D14 |
| 01111 | D15 |
| 10000 | D16 |
| 10001 | D17 |
| 10010 | D18 |
| 10011 | D19 |
| 10100 | D20 |
| 10101 | D21 |
| 10110 | D22 |
| 10111 | D23 |
| 11000 | D24 |
| 11001 | D25 |
| 11010 | D26 |
| 11011 | D27 |
| 11100 | D28 |
| 11101 | D29 |
| 11110 | D30 |
| 11111 | D31 |

Multiplekser 32/1 radi u zavisnosti od vrednosti S4,S3,S2,S1,S0 jedan od ulaza D0, D1, D2, D3,D4, D5, … , D31 bira se kao izlaz Y.

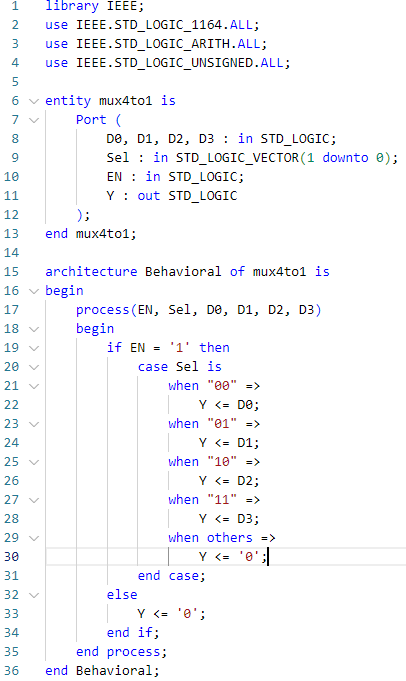


Slika 2. Multiplekser 32/1 realizovan sa multiplekserima 8/1

Sel(4) i Sel(3) se koriste za kontrolu koji od multipleksera će da radi i prosleđuje se na signal dozvole kroz I kolo. Sel(2 do 0) će biti prosleđeni na standardne Sel ulaze multipleksera 8/1 i imaće normalnu funckionalnost.

# 2. Projektovanje digitalnog integrisanog kola

# **Kod za mux 4/1:**



Slika 3. VHDL kod 4/1 multipleksera

Ovaj VHDL kod predstavlja 4-na-1 multiplekser (mux4to1). Sledi detaljno objašnjenje koda.

*Entitet*:

* mux4to1 je naziv entiteta.
* Ulazi (in**): D0**, **D1**, **D2**, **D3** su četiri ulazna signala, **Sel** je 2-bitni selektroski signal, a **EN** je signal dozvole.
* Izlaz (out): **Y** je izlaz multipleksera.

*Arhitektura*:

Ovo je arhitektura nazvana "Behavioral" za entitet mux4to1.

Proces ima osetljivost na promene signala EN, Sel, D0, D1, D2 i D3.

Ako je signal dozvole visok ('1'), proces ulazi u case naredbu na osnovu vrednosti selektorskog signala.

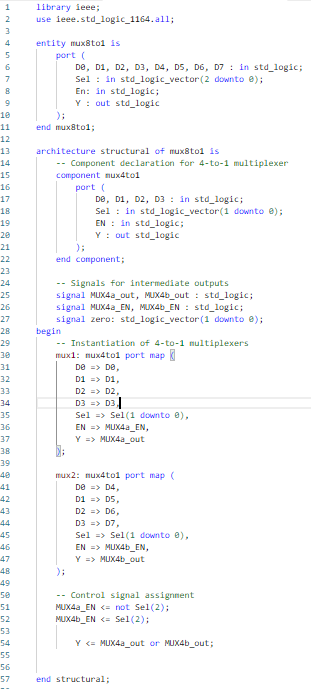
U zavisnosti od vrednosti Sel, izlaz Y se postavlja na odgovarajući ulazni signal.

Ako vrednost Sel ne odgovara nijednoj od navedenih vrednosti, postavlja se izlaz Y na '0'.

Ako je signal dozvole nizak ('0'), izlaz Y se postavlja na '0'.

U suštini, ovaj kod implementira 4-na-1 multiplekser gde se izlaz bira na osnovu 2-bitnog selektorskog signala kada je signal dozvole aktivan ('1'). Ako je signal dozvole neaktivan ('0'), izlaz Y se postavlja na '0'.

# Kod za mux8/1:



Slika 4. VHDL kod 8/1 multipleksera

Ovaj VHDL kod implementira 8-na-1 multiplekser (mux8to1) pomoću dva 4-na-1 multipleksera (mux4to1). Sledi detaljno objašnjenja koda.

*Entitet*:

* mux8to1 je naziv entiteta.
* Ulazi (in): **D0** do **D7** su osam ulaznih signala, **Sel** je 3-bitni selektorski signal, a **EN** je signal dozvole.
* Izlaz (out): **Y** je izlaz multipleksera.

*Arhitektura*:

Ovo je arhitektura nazvana "Structural" za entitet mux8to1.

Mux8to1 koristi dva 4-na-1 multipleksera da bi se ostvario 8-na-1 multiplekser.

Prvi 4-na-1 multiplekser (mux1) obrađuje ulazne signale D0 do D3, dok drugi (mux2) obrađuje ulazne signale D4 do D7.

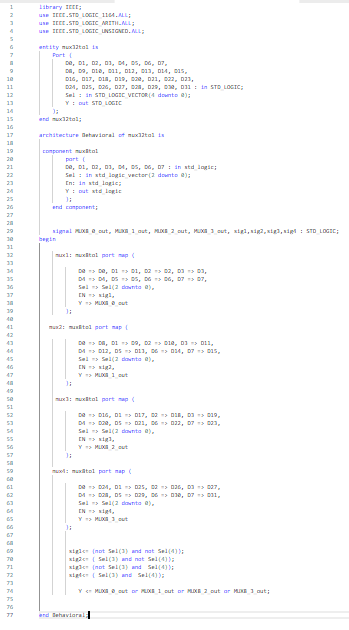
Kontrolni signal Sel(2) se koristi za određivanje koji od ova dva multipleksera će biti aktiviran.

Rezultati izlaza oba multipleksera se kombinuju koristeći logičko "ili" i postavljaju na izlaz Y.

Kontrolni signali MUX4a\_EN i MUX4b\_EN određuju koji od dva multipleksera će biti aktivan, u zavisnosti od vrednosti Sel(2).

Kada je Sel(2) nizak ('0'), aktiviran je prvi multiplekser, a kada je Sel(2) visok ('1'), aktiviran je drugi multiplekser.

# Kod za mux32/1:



Slika 5. VHDL kod 32/1 multipleksera

Ovaj VHDL kod implementira 32-na-1 multiplekser (mux32to1) koristeći četiri 8-na-1 multipleksera (mux8to1). Sledi detaljno objašnjenje koda.

*Entitet*:

* mux32to1 je naziv entiteta.
* Ulazi (in): **D0** do **D31** su 32 ulazna signala, **Sel** 5-bitni selektorski signal .
* Izlaz (out): **Y** je izlaz multipleksera.

*Arhitektura*:

Ovo je arhitektura nazvana "Behavioral" za entitet mux32to1.

U arhitekturi Behavioral koriste se četiri 8-na-1 multipleksera da bi se realizovao 32-na-1 multiplekser.

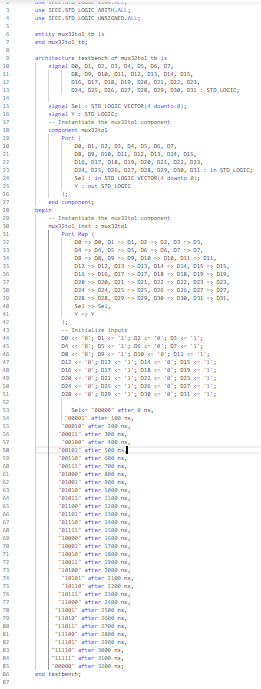
Svaki od četiri multipleksera procesuira osam od ukupno 32 ulazna signala.

Kontrolni signali (Sig1 do Sig4) određuju koji od četiri multipleksera će biti aktiviran.

Izlazi multipleksera se kombinuju logičkim "ili" operacijama i postavljaju na izlaz Y.

# 3. Simulaciona provera rezultata

Na slici se nalazi kod iz testbench kola.



Slika 6. VHDL kod 32/1 multipleksera (testbench)

Ovaj VHDL kod implementira testbench 32-na-1 multipleksera (mux32to1\_tb) koristeći Sledi detaljno objašnjenje koda.

*Entitet*:

* Mux32to1\_tb je naziv entiteta.
* Nemamo konkretno ulaze i izlaze nego inicijalizujemo mux32to1.

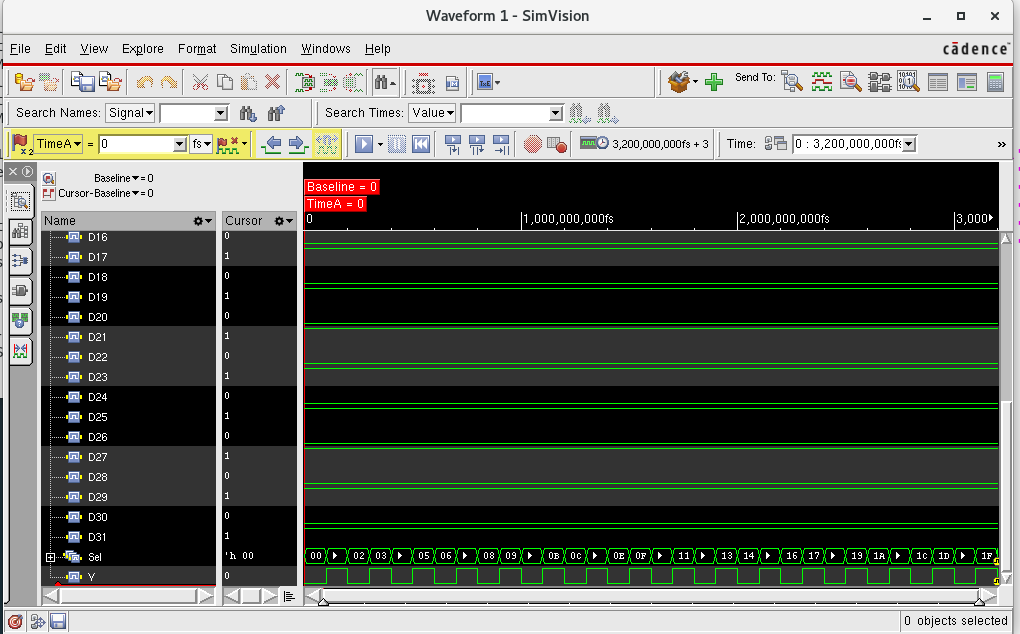
*Arhitektura*:

U arhitekturi testbench pravimo test benč entitet za mux32to1.

Koristimo signale D0 do D31 za ulaze i Sel za selektor.

Instanciramo komponentu mux32to1 i mapiramo ulaze i izlaze.

Inicijalizujemo ulazne signale i simuliramo promene selektora tokom vremena.



Slika 7. Rezultati Simulacije

Ovaj VHDL kod simulira 32-na-1 multiplekser (mux32to1) pomoću testbencha. U nastavku imate detaljno objašnjenje rezultata simulacije korak po korak.

* Inicijalizacija (vreme = 0 ns):

Svi ulazni signali (D0 do D31) inicijalizuju se sa naizmeničnim vrednostima '0' i '1'.

Signal za selekciju (Sel) postavljen je na "00000" (binarno) na vremenu 0 ns. Multiplekser je u početnom stanju, a izlaz (Y) zavisi od početne vrednosti signala za selekciju. U ovom slučaju, izlaz je vrednost D0 (koja je '0').

* Korak 1 (vreme = 100 ns):

Sel se ažurira na "00001".

Izlaz (Y) sada zavisi od vrednosti D1 (koja je '1').

* Koraci 2 do 31 (vreme = 200 ns do 3200 ns):

Signal za selekciju (Sel) nastavlja da se menja pri svakom koraku i uveczava za 1.

Pri svakom koraku, izlaz (Y) odgovara vrednosti izabranog ulaza na osnovu binarne vrednosti Sel.

Sekvenca izabranih ulaza ponavlja se svakih 100 ns (od D0 do D31 redom).

* Korak 32:

Signal za selekciju (Sel) ponovo je postavljen na "00000", čime se izlaz (Y) vraća na vrednost D0.

* Završetak (vreme = 3100 ns):

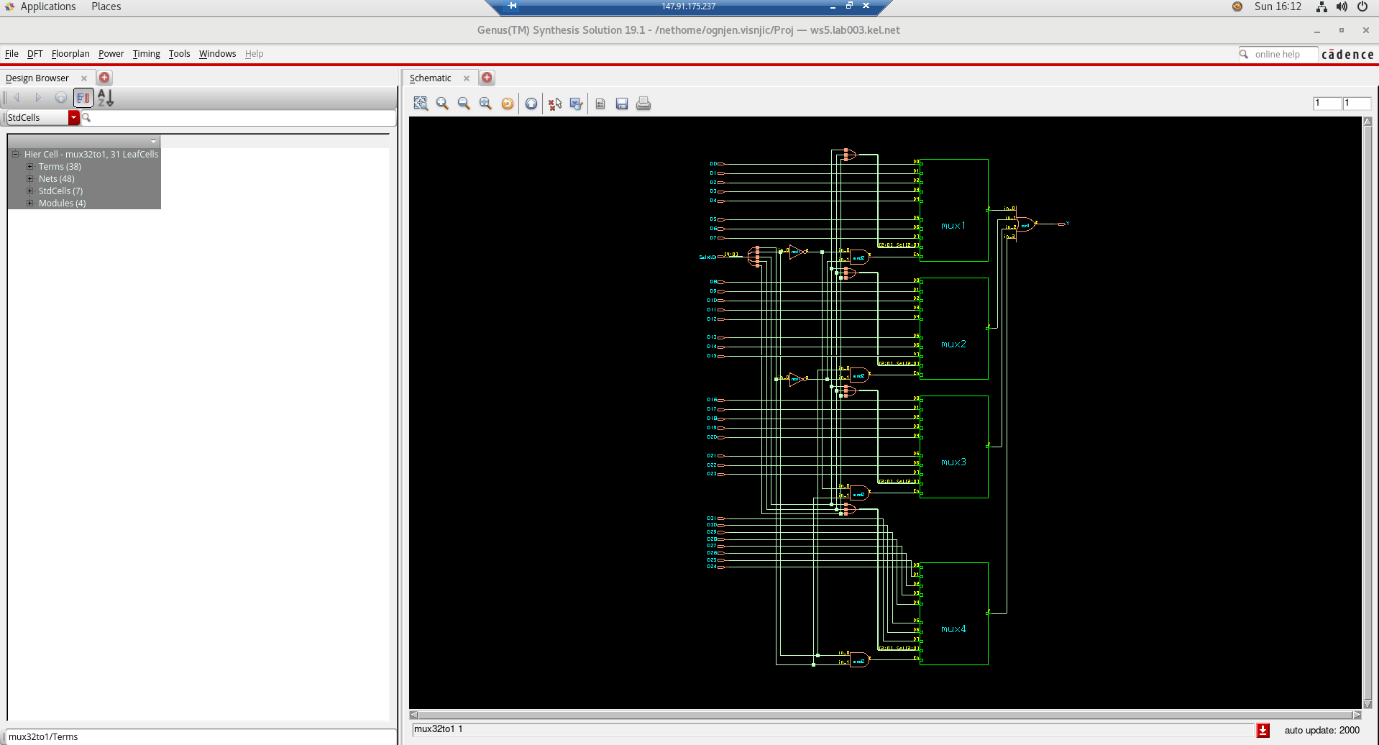
Simulacija se završava nakon 3100 ns.

Ukratko, simulacija prikazuje ponašanje 32-na-1 multipleksera tokom vremena. Signal za selekciju prolazi kroz sve moguće binarne vrednosti od "00000" do "11111", čime se multiplekser podređuje odabranom ulazu pri svakom koraku. Izlaz (Y) odražava vrednost izabranog ulaza. Testbench simulira kompletan ciklus odabira ulaza, vraćajući se na početno stanje pre završetka simulacije.

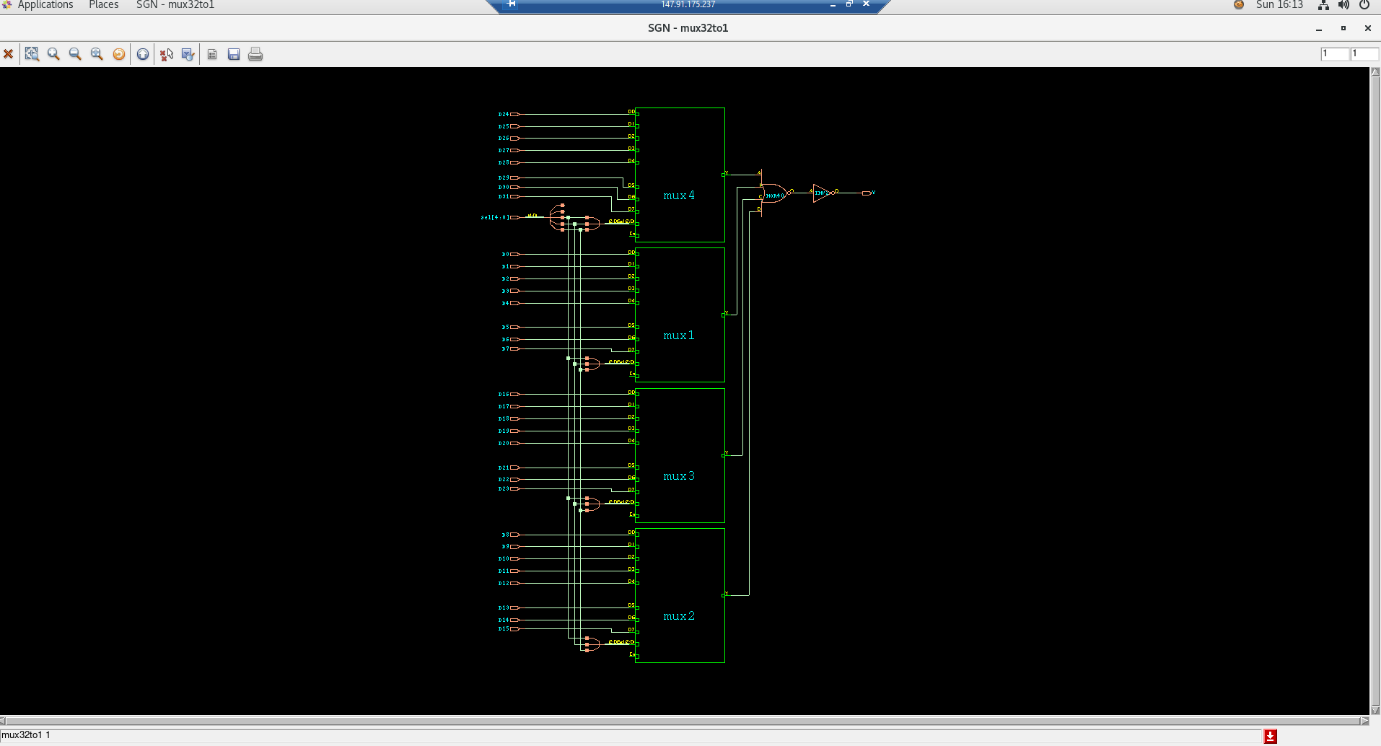
# 4. Lejaut i šematik projektovanog kola

Na slici 8 nalazi nam se šematik koji smo generisali uz pomoć alata *Genus Synthesis Solution*.

Na slici 9 nalazi nam se šematik nakon sinteze koji smo generisali takođe uz pomoć alata *Genus Synthesis Solution*.

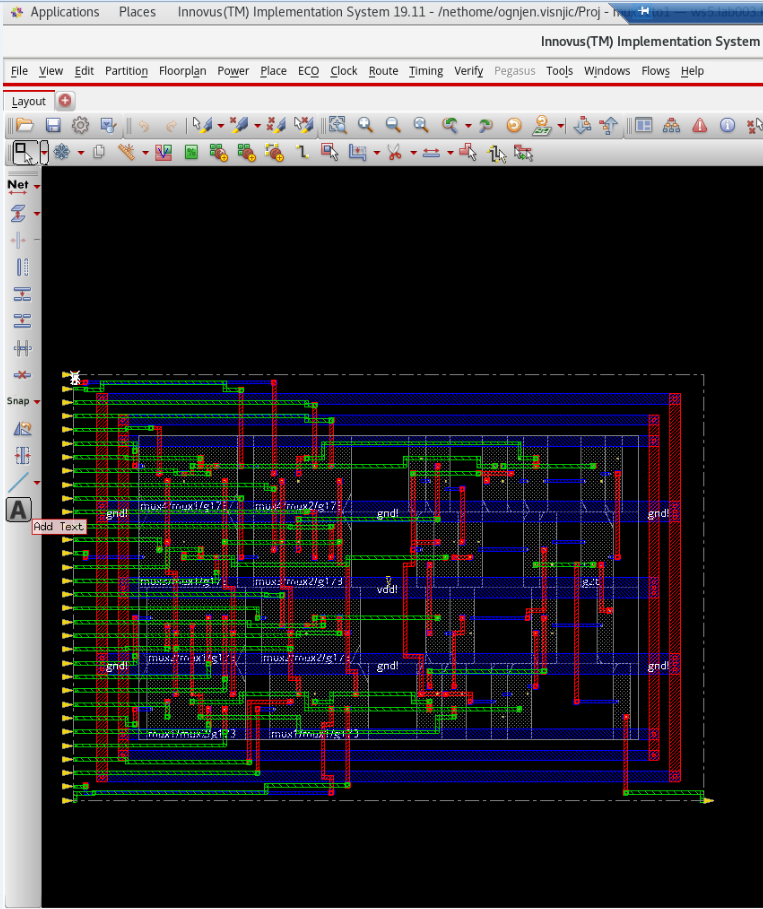


Slika 8. Generisani šematik kola



Slika 9. Generisani šematik kola nakon sinteze

Na slici 10 nalazi nam se lejaut koji smo generisali uz pomoć alata *Innovus Implementation System*.



Slika 10. Generisani lejaut kola

# 5. Zaključak

U okviru ovog projekta, analizirali smo digitalno kolo multipleksera 32/1, razvili VHDL kod za njegovu implementaciju, napisali test-benč za proveru ispravnosti rada, simulirali ga i donosimo sledeće zaključke.

* Funkcionalnost kola:

Multiplekser 32/1 je digitalno kolo koje omogućava izbor jednog od 32 ulaza na osnovu kontrolnog signala.

Implementiran je korišćenjem pet multipleksera 8/1, što omogućava jednostavnu i modularnu strukturu.

* Ispravnost rada:

Test-benč je uspešno prošao testove sa različitim kombinacijama ulaznih signala i kontrolnih signala.

Multiplekser 32/1 je odgovorio očekivano, izlazni signal se pravilno menja u skladu sa selektovanim ulazom.

* Prednosti:

Modularna struktura kola omogućava lakše održavanje i proširivost.

VHDL kod je pisan na način koji omogućava jasno razumevanje strukture i funkcionalnosti.

* Mane:

Višestruki slojevi multipleksera mogu dovesti do povećane zakašnjenja signala, što može uticati na brzinu rada kola.

Potrebno je pažljivo upravljati veličinom kontrolnog signala kako bi se očuvala brzina prenosa podataka.

* Poboljšanja:

Analiza i optimizacija geometrijskih parametara tranzistora mogla bi poboljšati brzinu prenosa podataka.

Dodatna implementacija optimizacija u VHDL kodu može doprineti efikasnosti kola.

* Zaključak:

Multiplekser 32/1 je efikasno implementiran i testiran kroz simulaciju.

Razumevanje geometrijskih parametara tranzistora i njihov uticaj na performanse kola ključno je za dalju optimizaciju.

Projekat predstavlja dobar temelj za dalje istraživanje i poboljšanja u smislu brzine i efikasnosti.