

Univerzitet u Nišu
Elektronski Fakultet

Predrag Petković
Miona Andrejević Stošović
Miljana Milić
Dejan Mirković

**Praktikum laboratorijskih vežbi iz
predmeta Projektovanje elektronskih kola
i Projektovanje digitalnih integrisanih
kola**



Education and Culture DG

TEMPUS JEP 41107 - 2006

Niš, Februar 2010.

PRAKTIKUM LABORATORIJSKIH VEŽBI IZ PREDMETA
PROJEKTOVANJE ELEKTRONSKIH KOLA I PROJEKTOVANJE
DIGITALNIH INTEGRISANIH KOLA

Izdavač: Elektronski fakultet u Nišu

P. fah 73, 18000 Niš

<http://www.elfak.ni.ac.rs>

Glavni i odgovorni urednik: Prof. Dr Zoran Perić

CIP - Каталогизација у публикацији
Народна библиотека Србије, Београд

621.3.049(075.8)(076)

004.42OrCAD(075.8)(076)

PRAKTIKUM laboratorijskih vežbi iz
predmeta Projektovanje elektronskih kola i
Projektovanje digitalnih integrisanih kola /
Predrag Petković ... [et al.]. - Niš :
Elektronski fakultet, 2010 (Niš : Unigraf). -
225 str. : ilustr. ; 25 cm

Na vrhu nasl. str.: Univerzitet u Nišu. -
Tiraž 100. - Bibliografija: str. 225.

ISBN 978-86-6125-019-4

1. Петковић, Предраг М., 1954- [аутор]

а) Електрична кола - Пројектовање - Вежбе

б) Апликативни програм "OrCAD" - Вежбе

COBISS.SR-ID 179325708

Sadržaj

I Projektovanje elektronskih kola- Uvod.....	1
1. Korišćenje <i>OrCAD Capture</i> alata za crtanje električne šeme.....	3
2. Analiza analognih elektronskih kola u jednosmernom režimu rada programom <i>PSpice</i>	29
3. Analiza analognih elektronskih kola u frekvencijskom i vremenskom domenu programom <i>PSpice</i>	55
4. Logička simulacija programom <i>PSpice</i>	85
5. Simulacija kola sa mešovitim signalima programom <i>PSpice</i>	111
6. Optimizacija vrednosti parametara elektronskog kola.....	129
II Projektovanje digitalnih integrisanih kola- Uvod.....	143
7. Crtanje električne šeme invertora korišćenjem alata <i>DA-IC</i>	149
8. Simulacija elektronskih kola korišćenjem alata <i>IC studio</i>	165
9. Crtanje lejauta invertora korišćenjem alata <i>Design Architect-IC</i>	175
10. Projektovanje ASIC kola na bazi standardnih ćelija upotrebom ADK alata.....	195
Prilog 1.....	213
Prilog 2.....	214
Prilog 3.....	224
Literatura.....	225

I Praktikum laboratorijskih vežbi iz predmeta

Projektovanje elektronskih kola

I.1 Uvod

Prvi deo praktikuma sadrži vežbe predviđene programom predmeta **Projektovanje elektronskih kola** (PEK). U okviru ovog kursa očekuje se da studenti nauče kako da koriste programe za analizu, optimizaciju i simulaciju elektronskih kola. Očekuje se da se studenti osposobe da:

1. nacrtaju električnu šemu analognog elektronskog kola uz primenu šematskog editora;
2. kreiraju netlistu kola čiju su šemu nacrtali;
3. koriste program za analizu kola u
 - a. jednosmernom domenu (DC),
 - b. naizmeničnom (frekvencijskom) domenu (AC),
 - c. vremenskom domenu (TR);
4. analiziraju digitalna elektronska kola;
5. analiziraju kola sa mešovitim signalima (analogni/digitalni);
6. koriste program za optimizaciju parametara (izračunavanje optimalnih vrednosti elemenata) kako bi se zadovoljili traženi kriterijumi (odziv).

Vežbe će se izvoditi na platformi baziranoj na studentskoj verziji programa *OrCAD*.

Način pozivanja pojedinih alata i njihova namena opisani su u vežbama koje slede i to:

Vežba 1. opisuje primenu alata *OrCAD Capture* za crtanje električne šeme.

Vežba 2. upućuje na semantiku i sintaksu jezika za opis hardvera u okviru netliste koju koristi program *PSpice* za simulaciju elektronskih kola u jednosmernom režimu.

Vežba 3. opisuje primenu programa *PSpice* za simulaciju elektronskih kola u AC i TR režimu.

Vežba 4. opisuje primenu programa *PSpice* za simulaciju digitalnih elektronskih kola.

Vežba 5. opisuje primenu programa *PSpice* za simulaciju elektronskih kola sa mešovitim signalima.

Vežba 6. opisuje primenu programa za optimizaciju vrednosti parametara kola *PSpice Optimizer*.

Svi navedeni alati deo su studentske verzije OrCAD paketa poznate pod nazivom *OrCAD family Release 9.2 Lite*, a dostupni su besplatno.

Vežba broj 1

Korišćenje *OrCAD Capture* alata za crtanje električne šeme

1.1 Cilj vežbe

Upoznati se sa načinom opisa kola u vidu električne šeme primenom šematskog editora iz paketa *OrCAD*. Na primeru pojačavača proučiti postupak crtanja električne šeme i svih pratećih aktivnosti.

1.2 Teorijska postavka vežbe

Električna šema predstavlja najčešći oblik opisa projekata u elektronici. Osim grafički, projekat može da se opiše u obliku teksta nekim od jezika za opis hardvera. Pre nego što su se razvile grafičke mogućnosti računara da prihvate i prikažu električnu šemu, opis se svodio na specificiranje liste komponenata. Uobičajeni naziv za ovu listu jeste *netlista*. Prvobitna funkcija netliste jeste da opiše izgled elektronskog kola kako bi bio razumljiv projektantima, sa jedne i računaru, sa druge strane. Naime, od računara se zahtevalo samo da analizira rad elektronskog kola pod određenim uslovima. U tom cilju trebalo je uneti podatke o tipu elementa, njegovom položaju u kolu i vrednostima parametara modela elementa (otpornost, kapacitivnost, induktivnost, ...). Prvobitnom zadatku računara

u procesu projektovanja biće posvećeno narednih pet vežbi u ovom praktikumu. Za sada ćemo se pozabaviti drugom funkcijom opisa projekta a to je dokumentovanje, odnosno kreiranje zapisa o izgledu kola. Kao što je rečeno, najlogičniji oblik predstavljanja nekog kola jeste grafički prikaz u obliku električne šeme. Koji su osnovni zahtevi koje mora da ispuni editor električne šeme?

Najpre mora da raspolaže skupom simbola na osnovu kojih se prepoznaju različiti tipovi komponenata (otpornik, kondenzator, kalem, tranzistor,...). Najčešće se skup elemenata koje editor prepoznaje nalazi u posebnim bibliotekama. Radi lakšeg snalaženja, biblioteke su organizovane po određenim pravilima koja ukazuju na pojedine tipove komponenata (analogne, bipolarni tranzistori, MOS tranzistori, prekidači,...). Obično svaki editor uz instalaciju sadrži određeni broj biblioteka. Pored toga, pojedini proizvođači komponenata nude biblioteke sa modelima sopstvenih proizvoda. Tako se mogu naći biblioteke sa nazivima pojedinih proizvođača.

Pozicioniranje simbola komponenata na radnoj površini može da se obavlja slobodno ili u diskretnim tačkama koje se nalaze u preseku virtuelne horizontalne i vertikalne rešetke (*grid* opcija). (Izbor ove opcije nalazi se kod najvećeg broja editora, pa i u *OrCAD Capture* u *View* meniju). Osim toga, editor mora da obezbedi mogućnost pomeranja, okretanja, kopiranja i brisanja objekta. (Ove opcije nalaze se najčešće ispod *Edit* menija). Ove operacije efikasnije mogu da se obave pritiskom na pojedina slova sa tastature. O tome će više reči biti u tekstu koji opisuje konkretne aktivnosti u primeru.

Naravno nije dovoljno samo smestiti simbol komponente na radnu površinu. Potrebno je svaku komponentu karakterisati sa vrednostima parametara. Kada su u pitanju pasivne R, C i L komponente, osnovni parametar modela predstavlja vrednosti otpornosti za otpornik, kapacitivnosti za kondenzator i induktivnosti za kalem. U slučaju tranzistora spisak parametara koji ih karakterišu mnogo je veći, ali sa stanovišta dokumentovanja dovoljno je uneti katalošku oznaku tranzistora (BC107, 2N3055,...). Ovo važi i za ostale komponente.

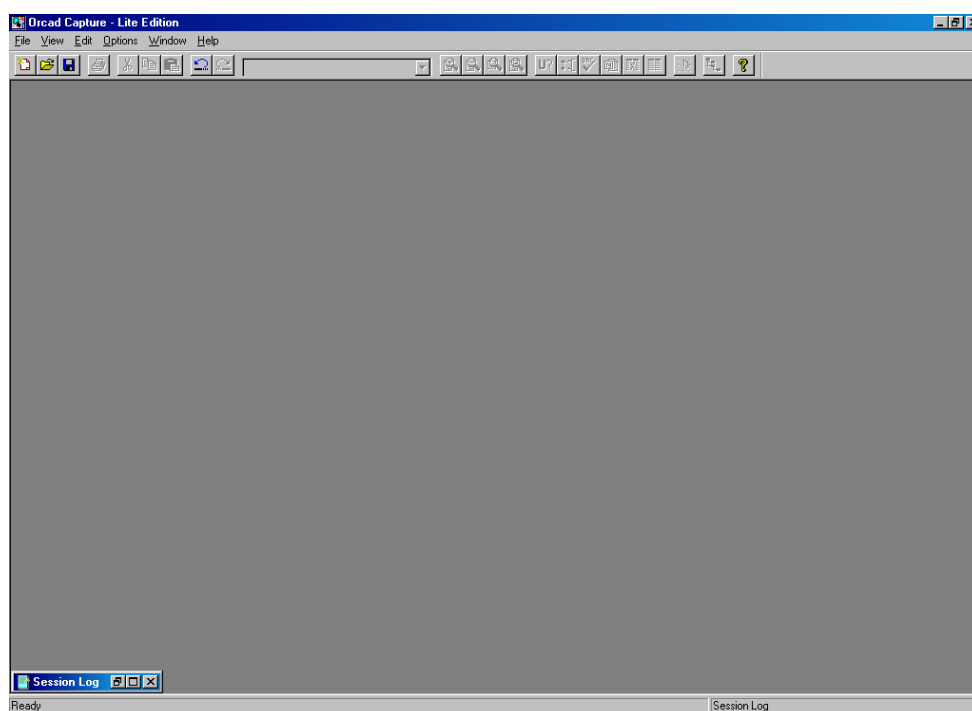
Pored toga editor treba da omogući međusobno povezivanje komponenata. Svaka komponenta povezuje se sa ostalima preko portova. Editor treba da obezbedi lako pronalaženje portova i spreči crtanje veze van portova.

Kada je reč o crtanju električne šeme koja služi za dokumentovanje projekta editor mora da obezbedi i prostor za unošenje podataka o kolu, autoru, vlasniku intelektualne svojine, datumu i sl.

1.3 Korišćenje šematskog editora *OrCAD Capture*

OrCAD Capture [-98] je program u okviru programskog paketa *OrCAD* koji se koristi za šematski prikaz elektronskih kola koja projektujemo.

Da bi se pokrenuo *OrCAD Capture* program, treba iz *Windows Start* menija izabrati *Orcad Family Release 9.2* programski paket, i iz njega pokrenuti *Capture CIS*. Izgled osnovnog prozora prikazan je na slici 1.1.

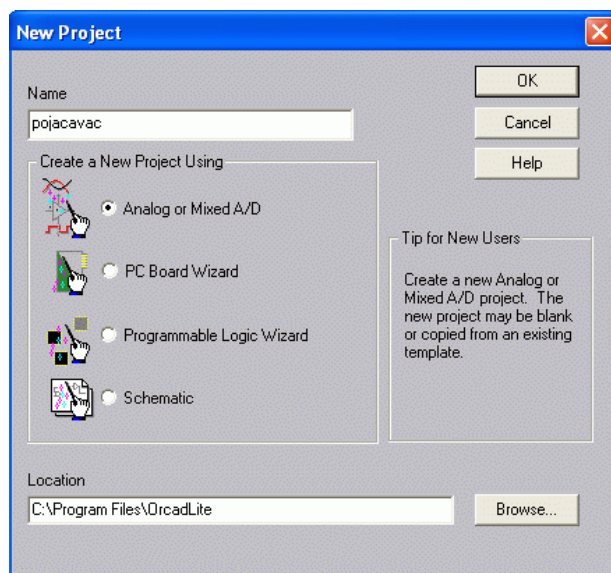


Slika 1.1: Izgled osnovnog prozora u programu *Orcad Capture*

U *Orcad Capture* prozoru:

- kliknuti **File -> New ->Project**

Otvara se novi prozor **New Project** čiji je izgled prikazan na slici 1.2.



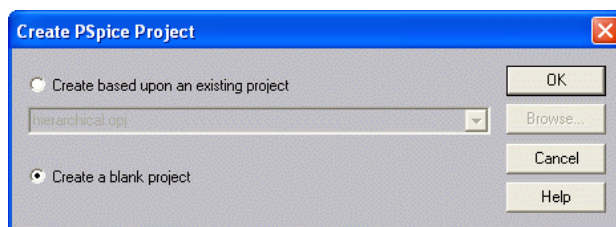
Slika 1.2: Dijalog kada se otvara novi projekat

U polje za dijalog *Name* treba upisati ime novog projekta (na primer *pojacavac*, na slici 1.2). Korisniku se nude 4 opcije koje bira zavisno od svrhe projekta. Prva opcija **Analog or Mixed A/D** omogućava da se električna šema poveže sa programom za analizu analognih i kola sa mešovitim analogno/digitalnim signalima (ovo praktično znači da sem simbola komponenta treba projektu priključiti i modele komponenta). Opcija **PC Board Wizard** omogućava da se posle crtanja električne šeme uđe u program za projektovanje štampanih kola (projektu se priključuju podaci o fizičkom izgledu komponenta). **Programmable Logic Wizard** uvodi projektanta u program za programiranje CPLD i FPGA komponenta (omogućava kreiranje VHDL opisa na osnovu električne šeme).

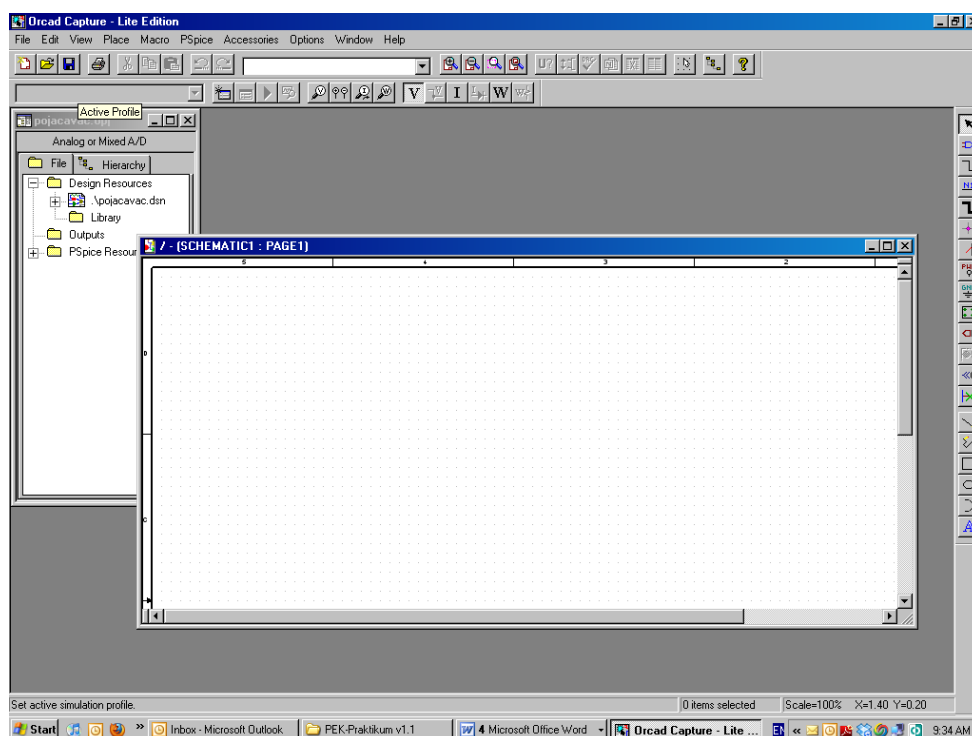
Opcija **Schematic** omogućava samo crtanje električne šeme. S obzirom da svaki novi projekat treba verifikovati simulacijom, opredeljujemo se za opciju **Analog or Mixed A/D**, kao na slici 1.2. Kliknuti **OK**.

Otvoriće se novi prozor za dijalog, prikazan na slici 1.3, gde treba izabrati opciju **Create a blank project** i kliknuti **OK**.

Otvara se radna površina prikazana na slici 1.4 u koju se unosi nova šema.

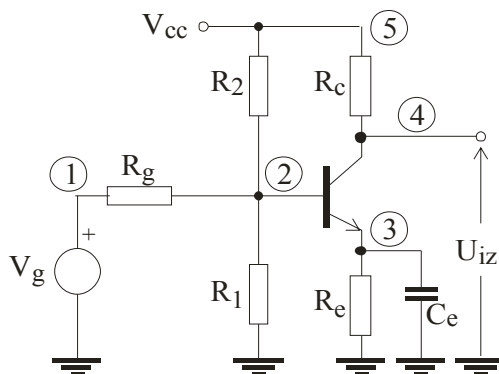


Slika 1.3: Dijalog kada se otvara novi projekat



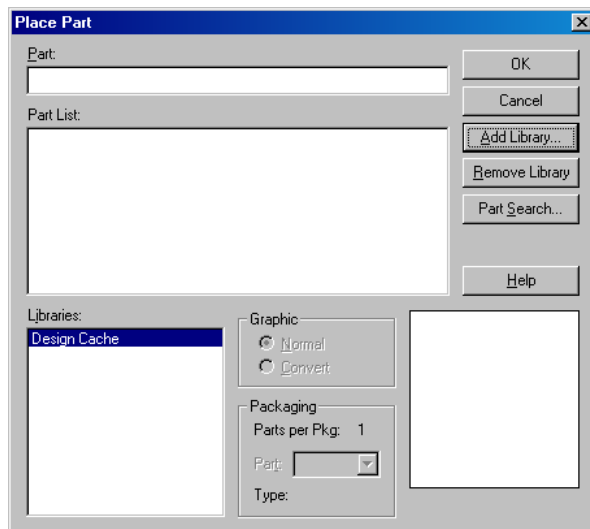
Slika 1.4: Radna površina editora električne šeme OrCAD Capture

U ovoj vežbi biće opisan postupak kreiranja šeme pojačavača sa slike 1.5. Vrednosti elemenata su: $V_{cc}=12V$; $R_1=10k\Omega$; $R_2=100k\Omega$; $R_c=3k\Omega$; $R_e=570\Omega$; $R_g=50\Omega$; $C_e=100\mu F$; $V_g=2V$. Za tranzistor koristiti model tranzistora BC107A.



Slika 1.5: Pojačavač sa zajedničkim emitorom

U *OrCAD Capture* prozoru kliknuti **Place** -> **Part**. Otvara se *Place Part* dijalog sa slike 1.6.

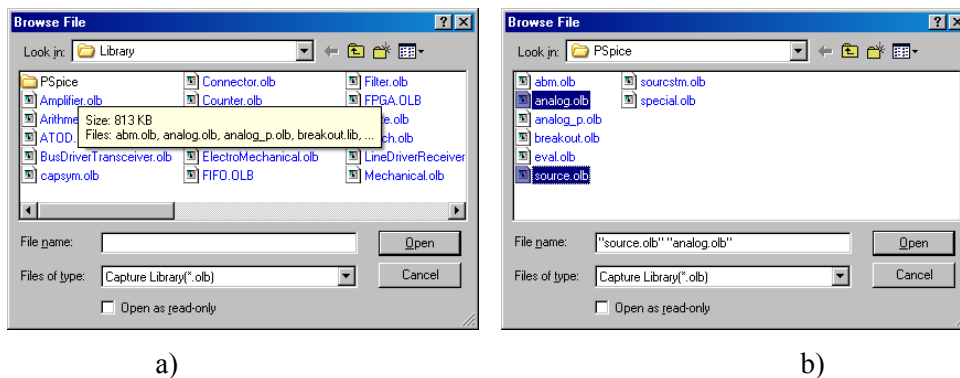


Slika 1.6: Dijalog za unos komponenti

Klikom na **Add Library** otvara se novi prozor u kome je prikazan spisak raspoloživih biblioteka. Ako znamo u kojim bibliotekama se nalaze, nama potrebni elementi, biramo samo te biblioteke, a možemo i da ih uvezemo sve, tako da nam je na raspolaganju veliki broj elemenata. Međutim, treba reći da unošenje svih biblioteka u radno okruženje korisnika otežava pretragu za potrebnom komponentom. Ovo važi za korisnika koji je instalirao program prvi put, svaki

naredni put standardne biblioteke su već povezane, i ovaj korak se preskače, a eventualno se dodaje neka nova potrebna biblioteka.

Pretpostavićemo da je korisnik upoznat sa sadržajem biblioteka. Za sada ćemo reći da se podaci o pasivnim analognim komponentama nalaze u biblioteci **analog.olb**, a o pobudnim generatorima i izvorima napajanja u biblioteci **source.olb** u direktorijumu PSpice, dok se o komponenti BC107A nalaze se u biblioteci **Transistor.olb** u direktorijumu Library. Zato iz spiska biblioteka sa slike 1.7.a treba izabrati biblioteke: **Transistor.olb** iz direktorijuma **Library**, kao i **analog.olb** i **source.olb** koje se nalaze u direktorijumu **PSpice** (slika 1.7.b).



Slika 1.7: Dijalog za unos biblioteka

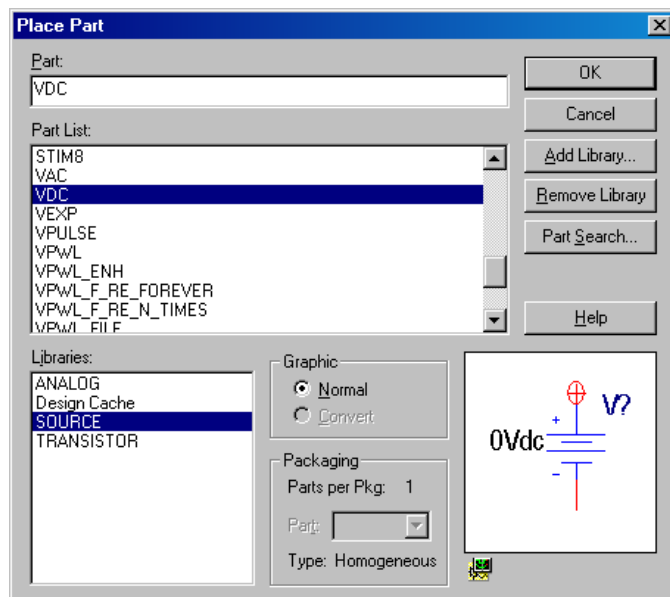
Unos simbola pojedinih komponenata počecemo od selektovanja izvora jednosmernog napona. Ova komponenta nalazi se u biblioteci SOURCE a prepoznaje se po nazivu VDC i simbolu u donjem desnom delu prozora **Place Part**. Zatim treba kliknuti **OK** (slika 1.8). Time se simbol izabrane komponente pojavljuje na radnoj površini i „klizi“ prateći pomeranje kursora.

Kursorom se pomerimo do mesta na radnoj površini na koje želimo da postavimo VDC, i kliknemo da bismo ga pozicionirali. Kliknemo desnim klikom i izaberemo **End Mode** da bismo izašli iz moda postavljanja komponenti. Alternativno, može se na tastaturi pritisnuti „Esc“ taster. Pritiskom tastera „R“ element se rotira za po 90° da bi se postigla željena orijentacija. Pritiskom tastera „H“ i „V“ komponenta se ogledalski preslikava po horizontali i vertikali, respektivno.

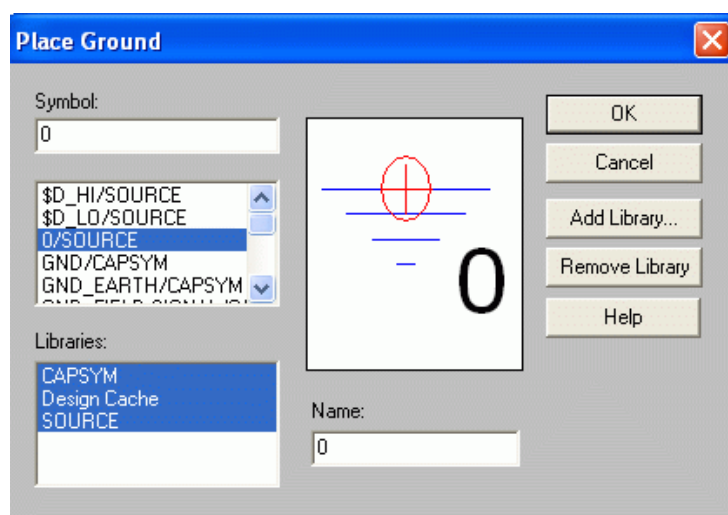
Istim postupkom kao za naponski izvor VDC, biraju se i ostale komponente i smestaju na radnu površinu. Otpornici se prepoznaju po oznaci R, kondenzatori po oznaci C, a naponski pobudni generator po oznaci VSCR. Tranzistor se bira iz biblioteke TRANSISTOR tako što selektujemo željenu komponentu, BC107A.

Najzad, treba dodati masu. U *OrCAD Capture* prozoru kliknuti **Place** -> **Ground** i izabrati masu 0 iz biblioteke *SOURCE (0/SOURCE)*, slika 1.9.

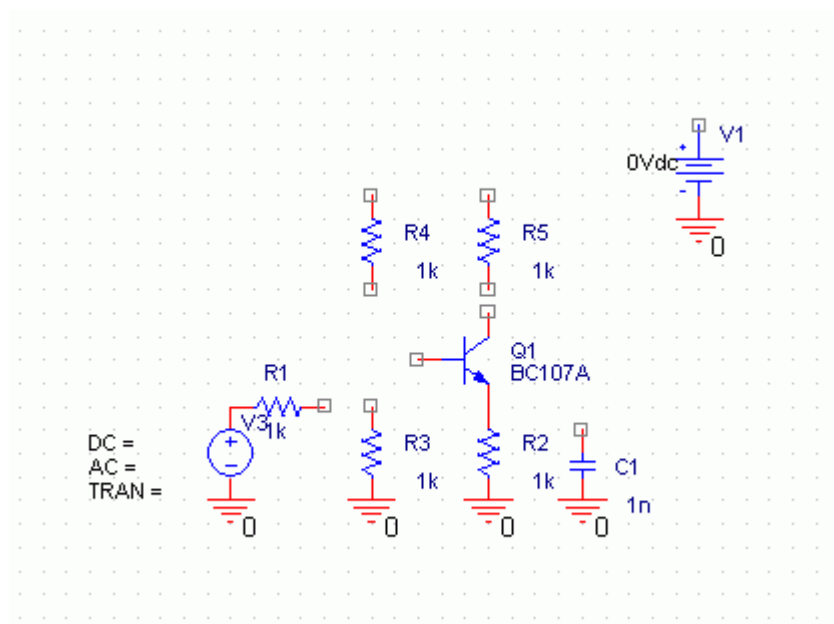
Radna površina treba da izgleda kao na slici 1.10.



Slika 1.8: Dijalog za izbor određene komponente (VDC)



Slika 1.9: Dijalog za izbor mase

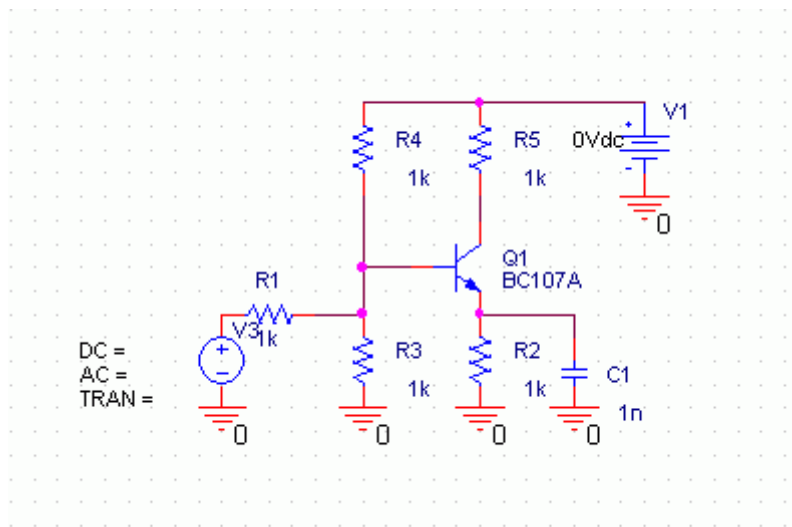


Slika 1.10: Izgled radne površine nakon unošenja elemenata

Sledeći korak je povezivanje ovih elemenata, da bismo dobili šemu kao na slici 1.4. Klikom na **Place -> Wire**, kursor dobija oblik znaka +, kliknemo na početak veze, vučemo kursorom do krajnje tačke, i kliknemo ponovo na mesto gde hoćemo da se veza završi. Treba povezati elemente da bi šema izgledala kao na slici 1.11. Kada se uspostavi kontakt između veze i elementa ili između dve veze, tačka kontakta se na kratko poveća, a u šemu se, automatski unosi kružić. Važno je napomenuti da, ukoliko se to ne desi, veza nije uspostavljena, mada optički može da se stekne takav utisak.

Već raspoređene komponente i veze mogu se selektovati dovođenjem kursora i klikom miša pojedinačno ili grupno uz držanje „Ctrl“ tastera. Pored toga, može se selektovati i više objekata koji se nalaze unutar pravougaonika definisanog položajem dijagonalinih uglova (povlačenje miša uz stisnuti levi taster). Svi selektovani elementi menjaju boju. Nad njima se mogu obavljati operacije brisanja („Delete“), rotiranja („R“), pomeranja (pomeranje miša sa pritisnutim levim tasterom), kopiranja („Ctrl“+“C“), vraćanja na radnu površinu (*paste* „Ctrl“+“V“), „sečenja“ *cut* („Ctrl“+“X“).

Kopirani elementi (ili cela šema) automatski su prebačeni u klipbord, tako da se mogu importovati u ostale dokumente.



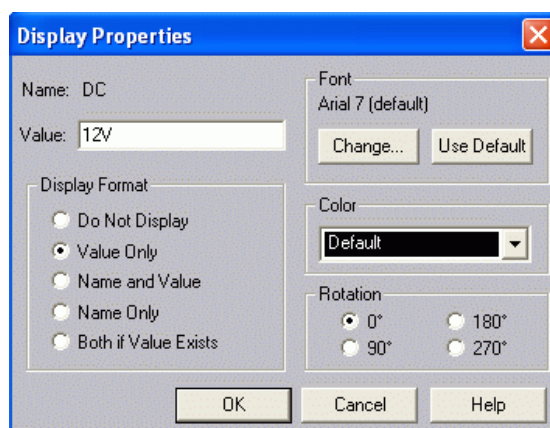
Slika 1.11: Izgled radne površine nakon povezivanja elemenata

Da bismo promenili ime nekog elementa, kliknemo na element, na primer V1, i otvara se dijalog čiji je deo prikazan na slici 1.12. U polje **Part Reference**, u kome je pisalo V1, unesemo novo ime elementa, na primer, Vcc.

Implementation Type	Name	Part Reference	PCB Footprint	Power Pins Visible
PSpice Model	/000003	Vcc		☐

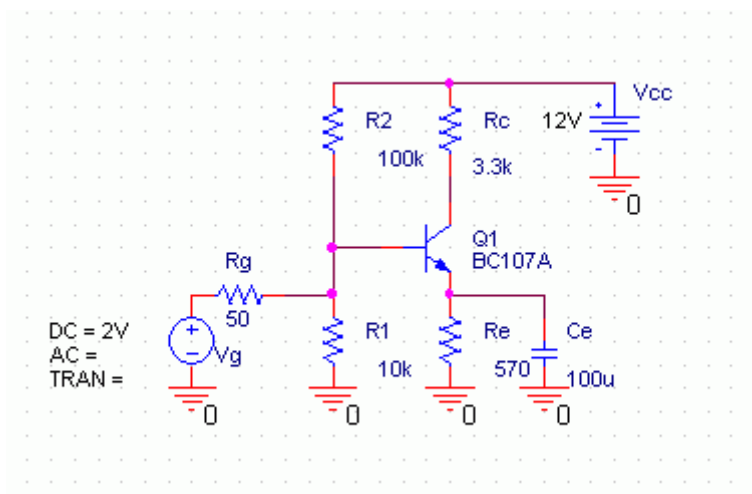
Slika 1.12: Dijalog za promenu imena elementa

Da bismo promenili vrednost elementa, na primer Vcc, kliknemo na labelu koja pokazuje vrednost (0V), i otvara se dijalog kao na slici 1.13. U polje *Value* unesemo željenu vrednost elementa (12V).



Slika 1.13: Dijalog za promenu vrednosti elementa

Kada se svim elementima promene vrednosti i imena, dobija se šema kao na slici 1.14.



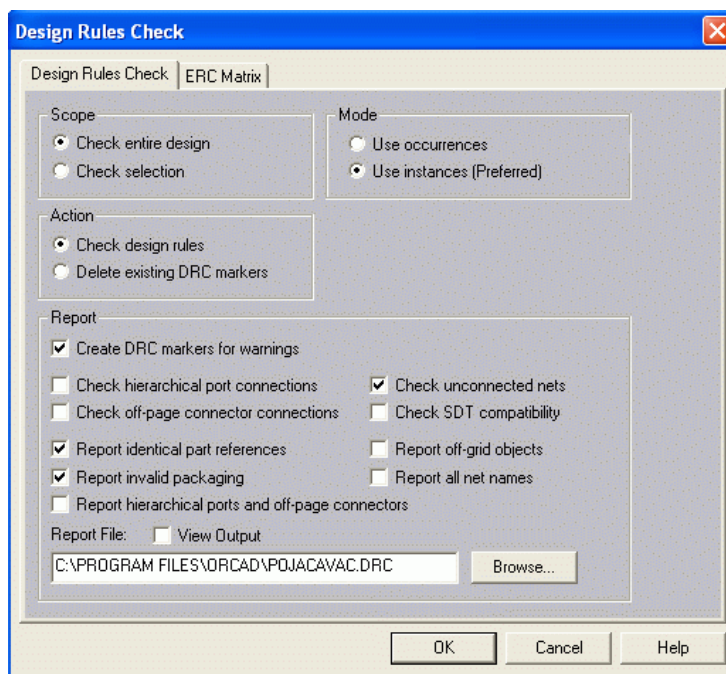
Slika 1.14: Konačni izgled šeme

Na kraju je potrebno snimiti šemu, **File -> Save**.

Po završetku crtanja električne šeme dobro je proveriti da li su svi elementi povezani i da li su ispoštovana sva pravila projektovanja. Ova faza provere šeme zove se *Design Rule Check*, a njome se proverava:

- da li su svi pinovi komponenta povezani;
- električna povezanost *Electric Rule Check, ERC*;
- da li su svi čvorovi međusobno povezani;
- da li nazivi komponenta odgovaraju predefinisanim vrednostima;
- dupliranje naziva komponenta;

Da bi se pokrenula, treba u programu za navigaciju selektovati **pojacavac.dsn**. Zatim izabrati **Tools-> Design Rule Check**, time se otvara prozor u kome treba potvrditi (čekirati) opcije kao na slici 1.15.

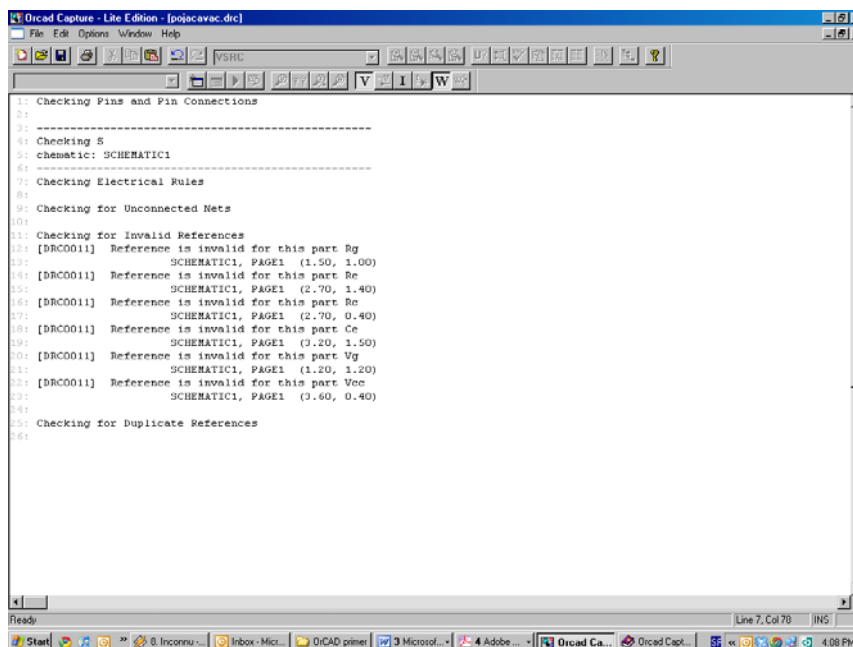


Slika 1.15: Prozor u kome se priprema fajl za proveru pravila projektovanja

Kada se pritisne OK, generiše se fajl **pojacavac.drc** koji se automatski priključi spisku fajlova u direktorijumu **Outputs** u navigacionom prozoru.

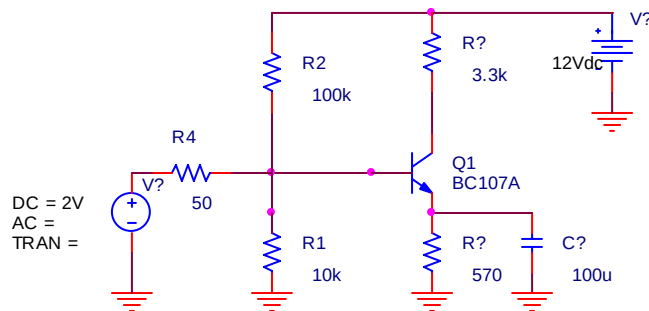
Ukoliko nisu registrovane greške, generiše se fajl čiji je sadržaj prikazan na slici 1.16.

Očigledno je da ne postoji greška u povezivanju, ali su evidentne greške koje se odnose na označavanje pojedinih komponenata. Ovo nisu katastrofalne greške tipa Error, nego se radi samo o načinu označavanja komponenata. Da je u prethodnom prozoru bila isključena opcija „Report invalid packaging“ ovaj tip greške ne bi bio prikazan.



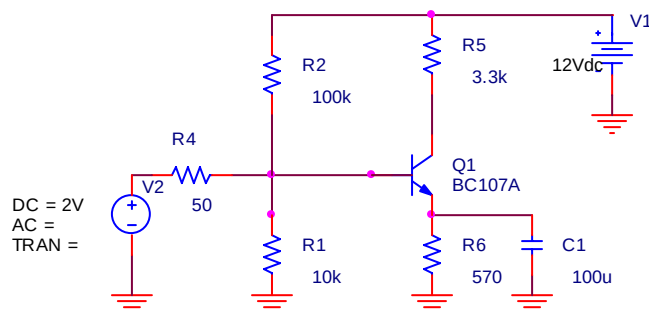
Slika 1.16: Izveštaj štampan u fajlu **pojacavac.drc**

Da bi se izbeglo prikazivanje prethodnog tipa grešaka, moguće je da se isključi opcija „Report invalid packaging“ ili da se svim elementima kojima je promenjena oznaka iz broja u slovo (recimo, u Rc, Vcc,...) umesto oznake unese simbol „?“ (videti sliku 1.17), a da se, potom, uključi opcija za automatsko označavanje elemenata. U ovu opciju ulazi se selektovanjem **pojacavac.dsn** u navigacionom prozoru i **Tools->Annotate**. Time se automatski ažurira označavanje naziva komponenata.



Slika 1.17: Zamena naziva komponenata sa znakom „?“

Posle ove opcije, elementi sa slike 1.17 dobijaju oznake kao na slici 1.18.

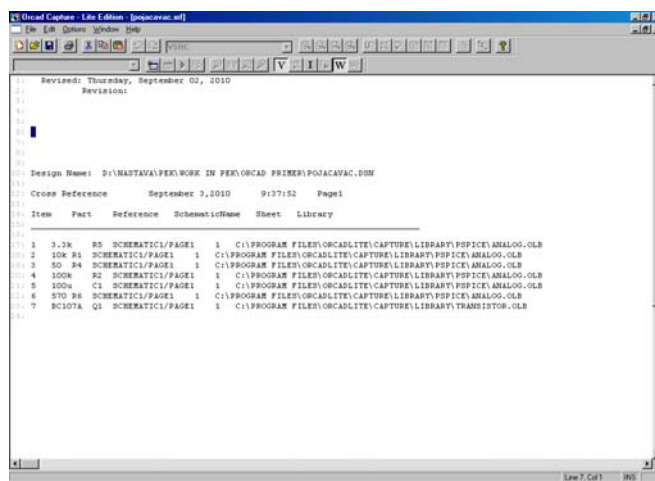


Slika 1.18: Efekat operacije *Annotate* na šemu sa slike 1.17

Ukoliko se ponovo pozove opcija za proveru pravila projektovanja, pod „Checking for Invalid References“ neće biti neispravnih komponenata. Treba napomenuti da je moguće da se u posebnom fajlu zapamte imena komponenata iz koga bi se uz opciju **Tools->Back Annotate** vratili raniji nazivi komponenata.

Kada se proverí ispravnost električne šeme mogu se iz šematskog editora kreirati lista komponenata (**Cross Reference**), lista za nabavku materijala (**Bill of Materials**), kao i potrebne netliste. Svim ovim opcijama pristupa se iz menija **Tools** kada je selektovan **poj.acavac.dsn** u navigacionom prozoru.

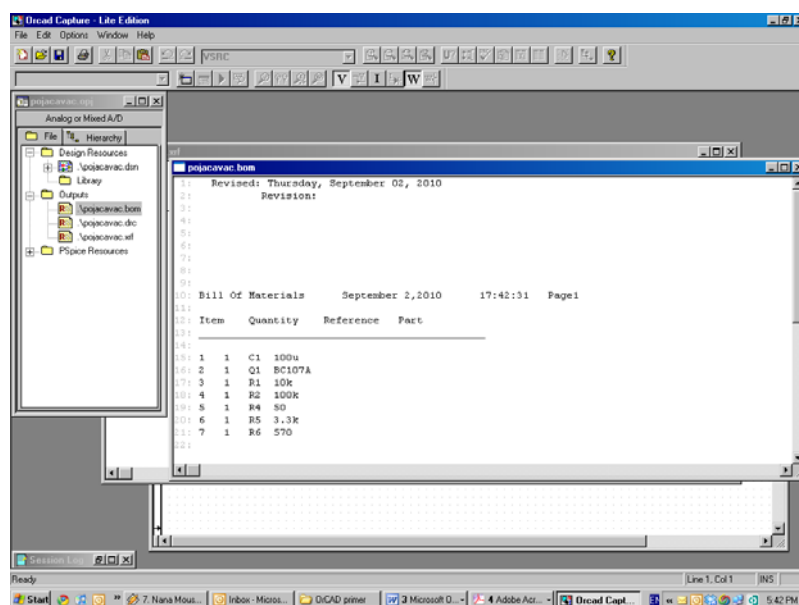
Posle izbora opcije **Tools->Cross Reference** generiše se fajl **pojacavac.xrf**. Njegov sadržaj ilustrovan je na slici 1.19.



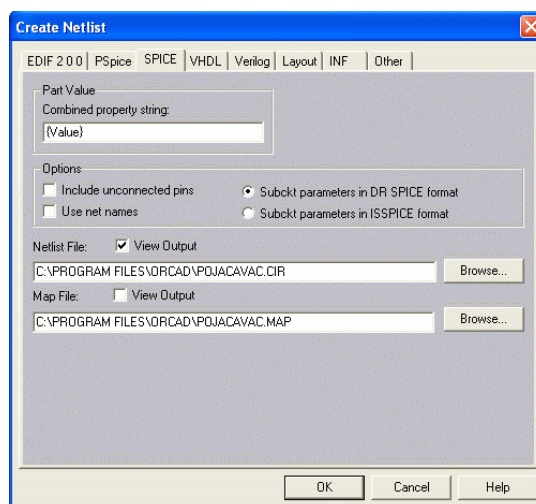
Item	Part	Reference	SchematicName	Sheet	Library
1	3.2k	R5	SCHEMATIC1/PAGE1	1	C:\PROGRAM FILES\ORCAD\LIBRARY\PSPIPICE\ANALOG.OLB
2	10k	R1	SCHEMATIC1/PAGE1	1	C:\PROGRAM FILES\ORCAD\LIBRARY\PSPIPICE\ANALOG.OLB
3	50	R4	SCHEMATIC1/PAGE1	1	C:\PROGRAM FILES\ORCAD\LIBRARY\PSPIPICE\ANALOG.OLB
4	100k	R2	SCHEMATIC1/PAGE1	1	C:\PROGRAM FILES\ORCAD\LIBRARY\PSPIPICE\ANALOG.OLB
5	100k	C1	SCHEMATIC1/PAGE1	1	C:\PROGRAM FILES\ORCAD\LIBRARY\PSPIPICE\ANALOG.OLB
6	570	R6	SCHEMATIC1/PAGE1	1	C:\PROGRAM FILES\ORCAD\LIBRARY\PSPIPICE\ANALOG.OLB
7	BC107A	Q1	SCHEMATIC1/PAGE1	1	C:\PROGRAM FILES\ORCAD\LIBRARY\PSPIPICE\TRANSISTOR.OLB

Slika 1.19: Sadržaj fajla **pojacavac.xrf** za pojačavač sa slike 1.18

Izborom opcije **Tools->Bill of Materials** (pri selektovanom **pojacavac.dsn** u navigacionom prozoru) generiše se fajl **pojacavac.bom** koji se priključuje direktorijumu **Outputs** u navigacionom prozoru. Njegov sadržaj u slučaju kola sa slike 1.18 dat je na slici 1.20.



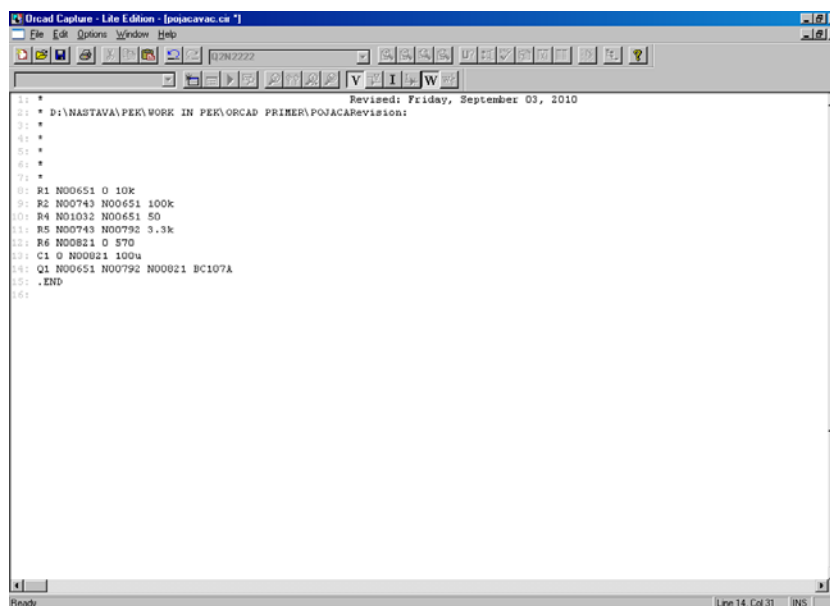
Slika 1.20: Sadržaj fajla **pojacavac.bom** za pojačavač sa slike 1.18



Slika 1.21: Izgled menija Create Netlist

Izborom opcije **Tools->Create netlist** (pri selektovanom **pojacavac.dsn** u navigacionom prozoru) otvara se prozor prikazan na slici 1.21 iz koga se biraju različiti formati netliste za kolo pojačavača.

Moguće je kreirati netliste pogodne za izradu štampanih ploča, EDIF, analizu programima PSpice, SPICE ili u formatima jezika za opis hardvera VHDL, Verilog, kao i u formatima Layout, INF i generiše se fajl **pojacavac.cir** koji se priključuje direktorijumu **Outputs** u navigacionom prozoru. Njegov sadržaj u slučaju kola sa slike 1.18 dat je na slici 1.22.



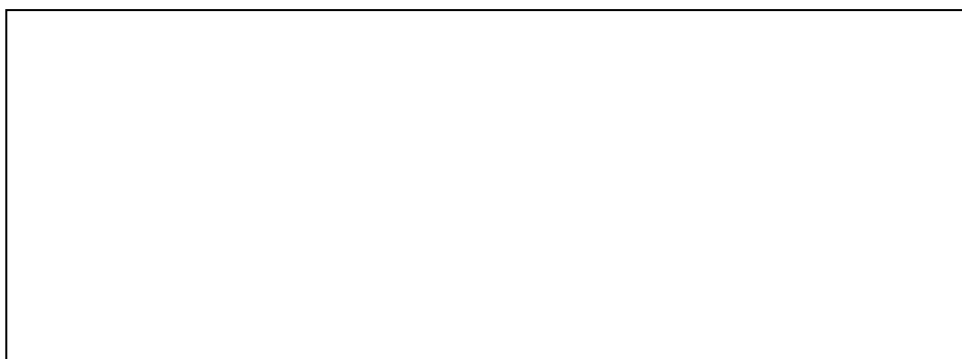
```
1: *
2: * Revised: Friday, September 03, 2010
3: *
4: *
5: *
6: *
7: *
8: R1 N00651 0 10k
9: R2 N00743 N00651 100k
10: R4 N01032 N00651 50
11: R5 N00743 N00792 3.3k
12: R6 N00821 0 570
13: C1 0 N00821 100u
14: Q1 N00651 N00792 N00821 BC107A
15: .END
16:
```

Slika 1.22: SPICE netlista za kolo sa slike 1.18

Z A D A T A K

ZADATAK: Štampanje električne šeme

1. Električnu šemu pojačavača sa slike 1.14 dopuniti otpornikom potrošača $R_p = 1k$ koji je preko $C_s = 1\mu F$ vezan za izlaz. Za aktivni element koristiti tranzistor 2N2222. Dobijenu šemu uneti u izveštaj sa Copy-Paste opcijom.



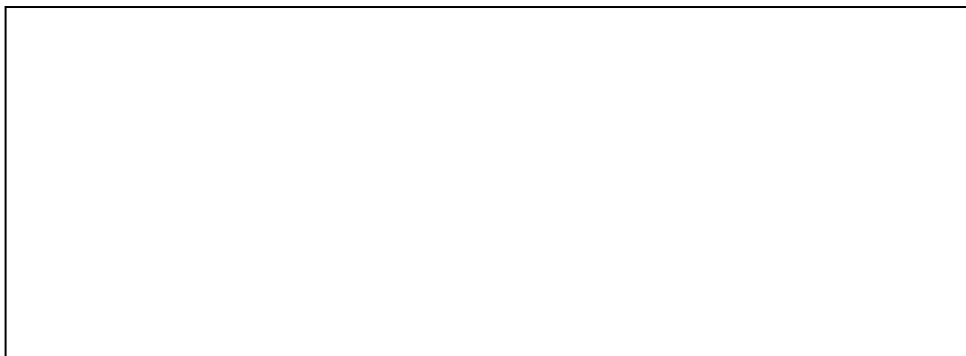
Slika 1.23: Električna šema pojačavača sa slike 1.14

2. Generisati PSpice netlistu pojačavača i priložiti je u izveštaj. Uporediti sa netlistom sa slike 1.22.



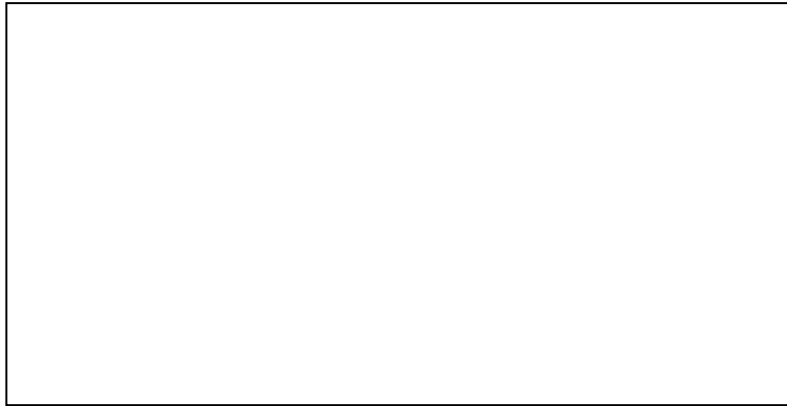
Slika 1.24: PSpice netlista za kolo sa slike 1.23

3. Primenom opcije „ogledalske simetrije“ tranzistor sa prethodne slike okrenuti tako da priključak baze bude sa desne strane slike. Rasporediti ostale elemente sa slike tako da pobuda bude sa desne, a potrošač sa leve strane slike. Generisati SPICE i PSpice netliste. Priložiti električnu šemu kola i obe netliste.



Slika 1.25: Električna šema pojačavača iz Zadatka 3





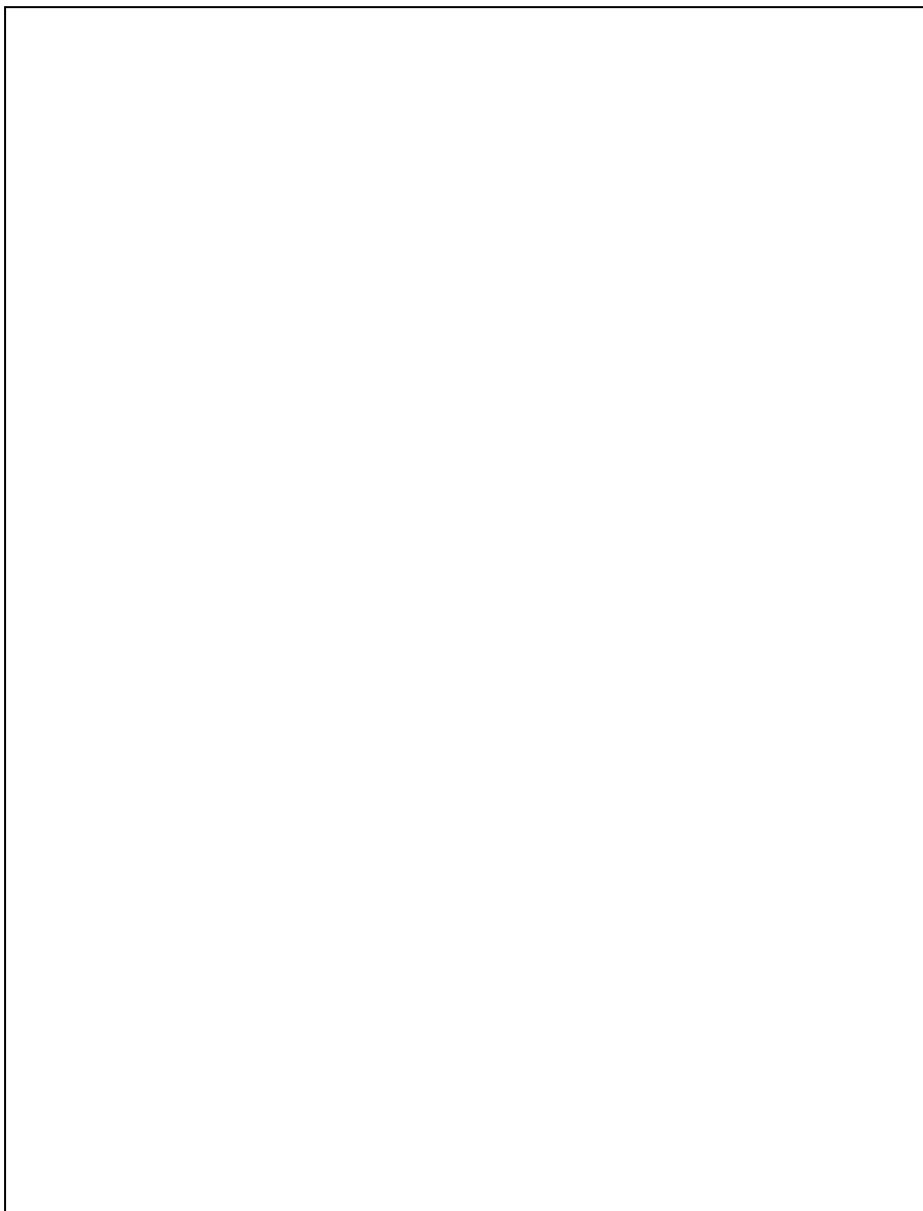
Slika 1.26: SPICE i PSpice netliste za kolo sa slike 1.25

4. Kopirati pojačavač sa slike 1.23 (sem potrošača, pobude i napajanja) i kreirati dvostepeni pojačavač sa dva identična stepena koji je na izlazu opterećen otpornikom R_p , a pobuđuje ga isti generator (sa R_g) kao onaj sa slike 1.23. Za oba stepena koristiti napajanje prvog stepena.



Slika 1.27: Električna šema pojačavača iz zadatka 4

5. Uraditi DRC proveru pojačavača iz zadatka 4. Priložiti dobijeni izveštaj.



Slika 1.28: DRC izveštaj za pojačavač iz zadatka 4

6. Otkloniti greške. Za renumeraciju elemenata koristiti opciju *Annotate*. Priložiti električnu šemu korigovanog kola i izveštaj DRC.

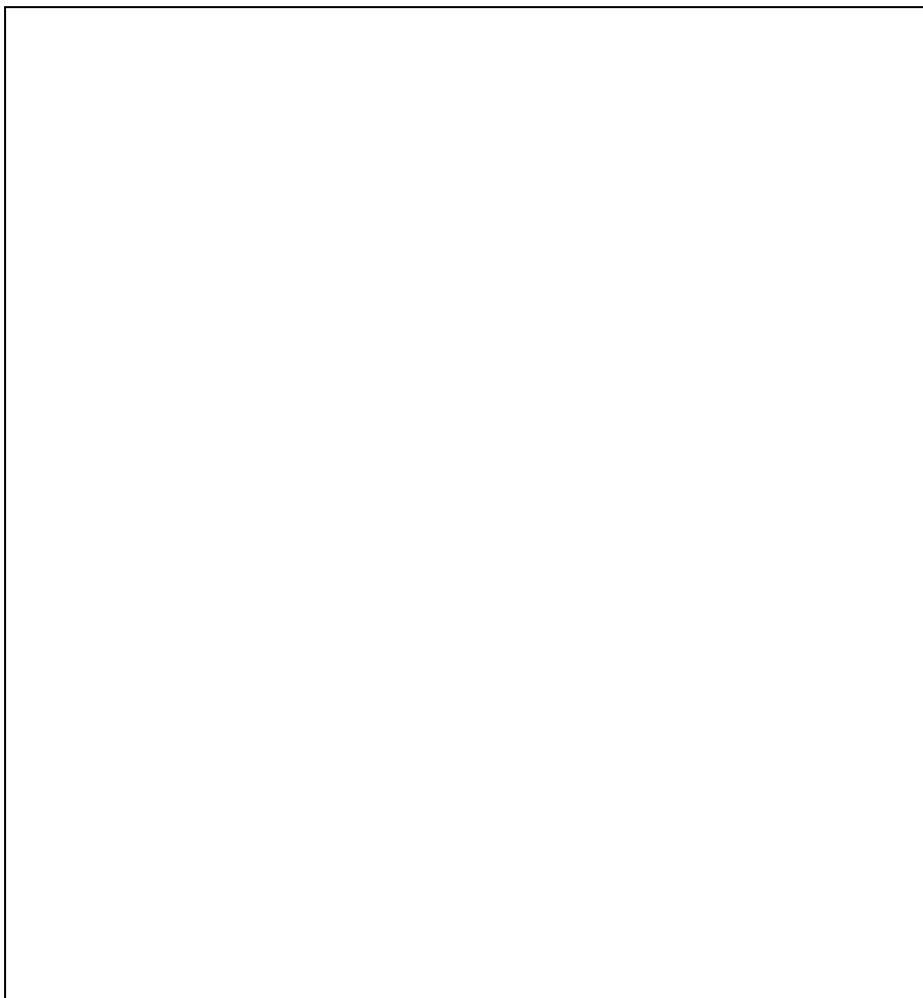


Slika 1.29: Električna šema sa korigovanim vrednostima elemenata

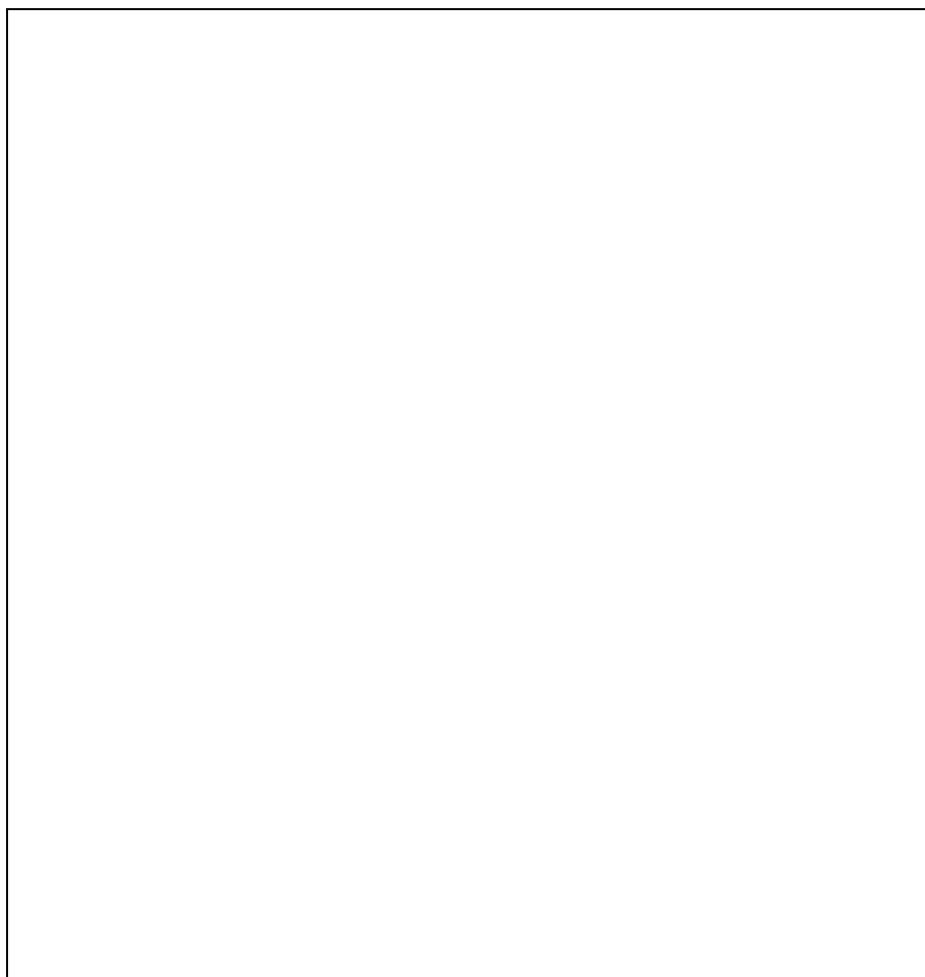


Slika 1.30: DRC izveštaj za pojačavač iz zadatka 6

7. Kreirati listu svih komponenata (Cross Reference) sortiranu po
 - a. vrednostima komponenata i
 - b. oznakama komponenata



Slika 1.31: Lista komponenata sortirana po vrednostima



Slika 1.32: Lista komponenata sortirana po oznakama

8. Kreirati listu za nabavku materijala (Bill of materials)



Slika 1.33: Lista za nabavku materijala

Datum : _____

Student : _____

Overava : _____

Vežba broj 2

Analiza analognih elektronskih kola u jednosmernom režimu rada programom *PSpice*

2.1 Cilj vežbe

Upoznati se sa osnovnim principima analize elektronskih kola u jednosmernom režimu rada primenom programa *PSpice*. Posebno proučiti način generisanja statičkih karakteristika aktivnih elemenata, određivanje mirne radne tačke pojačavača, izračunavanje dinamičkih parametara pojačavača, određivanje osetljivosti na promene vrednosti elemenata kola i uticaja temperature na statičke karakteristike.

2.2 Teorijska postavka vežbe

2.2.1 Električna analiza jednosmernih režima

Pod električnom analizom podrazumeva se postupak pomoću koga se može izračunati odziv električnog kola čija je pobuda poznata. Pri tome mogu da se utvrde i druge performanse kola kao što su potrošnja, temperaturske zavisnosti, šum, osetljivosti i slično. Analizu iole složenijih kola nemoguće je zamisliti bez upotrebe računara. Programi za analizu kola pomažu projektantima da provere ispravnost projektovanog kola pre realizacije. Da bi neko kolo moglo da se

analizira pomoću računara, neophodno je opisati njegovo ponašanje odgovarajućim sistemom jednačina. Pri tome, svaki element kola doprinosi sistemu jednačina saglasno načinu na koji je matematički opisan njegov uticaj na rad kola, odnosno načinu na koji je definisan *električni model* tog elementa.

Iz teorije električnih kola poznato je da se isto kolo različito ponaša zavisno od režima u kome radi. Shodno tome, modeli koji opisuju ponašanje kola razlikuju se u zavisnosti od režima rada kola, što unosi specifičnosti u opisivanju kola i postupku njegove analize. Iz toga proizilaze tri osnovna tipa analize kola:

- analiza jednosmernih režima (DC)
- analiza naizmjeničnih režima (AC, analiza u frekvencijskom domenu) i
- analiza u vremenskom domenu (TRAN, analiza prelaznih režima).

Ponašanje linearnih kola u DC režimu opisuje se skupom linearnih algebarskih jednačina. Nelinearna kola opisuju se sistemom nelinearnih jednačina. Analiza u frekvencijskom režimu (AC) podrazumeva rešavanje sistema linearnih jednačina u kome su sve promenljive kompleksne veličine. Ponašanje kola u vremenskom domenu (TRAN) opisuje se sistemom diferencijalnih jednačina koji, u zavisnosti od toga da li je kolo linearno ili nelinearno, može biti linearan ili nelinearan.

Sistem jednačina koji opisuje ponašanje kola rešava se numerički. Rešavanje sistema nelinearnih jednačina svodi se na iterativni postupak u okviru koga se više puta rešava sistem linearnih jednačina. U programima za analizu kola najčešće se za linearizaciju nelinearnih jednačina koristi metod *Newton-Kantorovich-a* [Mi88]. Sistem diferencijalnih jednačina rešava se diskretizacijom vremenske ose i aproksimacijom izvoda količnikom razlike vrednosti promenljivih i razlike vremena u susednim tačkama na vremenskoj osi. Ovaj postupak naziva se numeričkom integracijom sistema diferencijalnih jednačina. U programima za analizu kola u ovu svrhu najčešće se koriste višekoračni metodi [Lit00].

Jednosmerni režim rada linearnog kola opisuje se sistemom linearnih jednačina čije su promenljive naponi čvorova i struje grana u kolu. Pod analizom jednosmernog režima podrazumeva se rešavanje ovog sistema jednačina. Analiza nelinearnog kola u DC režimu svodi se na iterativno rešavanje sistema linearnih jednačina da bi se došlo do rešenja. Najčešće se do rešenja dolazi za manje od 10 iteracija. Jednosmerna analiza može se ponoviti za različite vrednosti nekog parametra kola (*parametarska analiza*) ili za različite vrednosti nekog pobudnog generatora (*generisanje statičkih karakteristika kola*). DC analiza neophodna je za određivanje dinamičkih parametara nelinearnih komponenata kao i dinamičkih parametara kola (kao što su pojačanje, ulazna i izlazna otpornost u mirnoj radnoj tački). Tek na osnovu tako dobijenih rezultata mogu se odrediti linearni modeli

komponentata neophodni u AC analizi kola. Takođe, granične vrednosti napona/struja na/kroz reaktivne komponente za $t=0$ određuju se DC analizom. Zbog toga DC analiza prethodi svakoj analizi u vremenskom i frekvencijskom domenu. Takođe, ponavljanjem jednosmerne analize moguće je odrediti osetljivost neke promenljive na parametre kola (*analiza osetljivosti*) ili neki parametar okoline kao što je recimo temperatura (*temperaturska analiza*).

2.2.2 Simulator PSpice

PSpice [-91] je simulator elektronskih kola sa analognim, digitalnim i mešovitim signalima. Dakle, može se koristiti za analizu elektronskih kola opisanih na nivou električne šeme (analogna kola) ili za simulaciju kola na nivou logičke šeme (digitalna kola) ili za simulaciju kola u kojima se javljaju signali oba tipa. Prilagođen je radu na personalnim računarima, što ga čini veoma rasprostranjenim i popularnim među projektantima elektronskih kola.

Četiri vežbe u ovoj knjizi (od Vežbe broj 2 do Vežbe broj 5) sadrže opis osnovne sintakse ulaznog jezika programa *PSpice*. Kompletan pregled sintakse ulaznog jezika simulatora *PSpice* dat je u prilogu 2.

2.2.3 Sintaksa ulaznog jezika simulatora PSpice

Analiza kola programom *PSpice* zahteva jednu ulaznu datoteku (nazivaćemo je jednostavno *ulazna datoteka*). *PSpice* uvek generiše jednu izlaznu datoteku, koju ćemo nazivati *tekstualna* izlazna datoteka. Ako se posebnom komandom zahteva upotreba grafičkog postprocesora *Probe*, *PSpice* generiše još jednu izlaznu datoteku u koju se smeštaju izlazni podaci u odgovarajućem grafičkom formatu. Ovu datoteku ćemo nazivati *grafička* izlazna datoteka. Osim ovih datoteka, *PSpice* može koristiti *bibliotečke* ulazne datoteke i *priključene* ulazne datoteke.

Da bi se kolo analiziralo, mora se kreirati ulazna datoteka koja sadrži listu podataka o strukturi kola i zahtevanim tipovima analiza. Ova lista naziva se *lista opisa kola*, odnosno *net-lista*. Ulazna datoteka treba da ima ekstenziju **.cir**. Ona može sadržati opis jednog ili više kola za simulaciju. Ako je u ulaznoj datoteci zadato više opisa kola, *PSpice* ih simulira sekvencijalno, jedno za drugim.

Rezultati analize smeštaju se u tekstualnu izlaznu datoteku koja ima isto ime kao i ulazna datoteka, ali sa ekstenzijom **.out**. *PSpice* može predati rezultate analize grafičkom postprocesoru *Probe* preko izlazne datoteke sa ekstenzijom **.dat**, što se zahteva komandom **.PROBE** u listi opisa kola.

Sve komande u ulaznoj datoteci pišu se od prve kolone (prvog znaka u liniji). Prazni redovi su dozvoljeni.

Prva linija opisa kola je takozvana *naslovna linija*. Ovu liniju *PSpice* tumači kao komentar o analizi, a ne kao sastavni deo opisa kola. Sadržaj naslovne linije upisuje se u izlazne datoteke (tekstualnu i grafičku) uz rezultate svake analize.

Lista opisa kola završava se komandom `.END` iza koje može da se navede opis narednog kola.

Između naslovne linije i komande `.END` nalaze se:

- komande za opis topologije kola i
- komande za kontrolu toka simulacije i za prikaz dobijenih rezultata.

Komande za opis topologije kola počinju slovom, a komande za kontrolu toka analize počinju tačkom.

PSpice ne pravi razliku između velikih i malih slova. Tako su imena `Vin`, `VIN`, `ViN` i `vIN` ekvivalentna.

Format za pisanje brojnih vrednosti parametara analize odgovara onom kod viših programskih jezika, s tom razlikom što *PSpice* dozvoljava korišćenje *numeričkih sufiksa*. Numerički sufiks je jedno ili više slova koja se pišu bez razmaka (razmak ćemo zvati znak *blanko*) iza broja i menjaju mu vrednost. Na primer, `1.2nsec` predstavlja brojnu vrednost $1.2 \cdot 10^{-9}$, jer se sufiks `n` tumači kao `nano` = 10^{-9} , a tekst iza numeričkog sufiksa do prvog blanko znaka smatra se komentarom. Spisak svih numeričkih sufiksa koje *PSpice* prepoznaje sa njihovim tumačenjima dat je u prilogu 1. Tekst iza brojne vrednosti do prvog blanko znaka koji nije prepoznat kao sufiks, takođe se smatra komentarom.

Linija koja počinje zvezdicom predstavlja komentar i *PSpice* je ignoriše. Druga vrsta komentara su takozvani linijski komentari koji mogu da se nalaze u svakoj liniji posle komande. Linijski komentar počinje znakom `;` a tekst iza njega do kraja linije tretira se kao komentar.

*** Ovo je komentar.**

.END ; Ovo je linijski komentar.

U tekstu naredbi dozvoljena je proizvoljna upotreba blanko znakova, tabulatora i zareza za međusobno razdvajanje pojedinih podataka radi poboljšanja čitljivosti. *PSpice* ignoriše ove znake pri učitavanju ulazne datoteke.

Jedna naredba može da se opiše u više redova. Red u kome se nastavlja opis naredbe mora započeti (u prvoj koloni) znakom `+`.

2.2.4 Analogni elementi koje prepoznaje PSpice

PSpice prepoznaje dvadeset i dva tipa različitih elemenata kola. Ovi elementi nabrojani su u tabeli 2.1.

Komanda za opis topologije kola sadrži tri grupe podataka koje se moraju navesti u sledećem redosledu:

- 1) oznaka elementa
- 2) topološka pozicija elementa u kolu
- 3) vrednosti parametara modela elementa.

Oznaka elementa počinje *karakterističnim slovom* koje predstavlja globalnu oznaku tipa elementa (R za otpornik, C za kondenzator, D za diodu, ...), a nastavlja se nizom karaktera koji određuju ime elementa. Karakteristična slova svih elemenata data su u tabeli 2.1. Tako, na primer, otpornik vezan za kolektor osmog tranzistora u kolu može da se označi sa Rc8, ili Rcol8, ili R93, ukoliko je to 93. otpornik u kolu. Obavezno je da oznaka počne karakterističnim slovom R, dok je ime elementa proizvoljan niz od najviše 131 karaktera. Naravno, preporučuje se da ime ne bude duže od 8 karaktera. Osim slova i brojeva, ime elementa može da sadrži i sledeće specijalne karaktere: \$, _, % i /. Nije dozvoljeno da dva elementa u kolu imaju ista imena.

Topološku poziciju elementa u kolu određuju imena čvorova za koje je taj element vezan i oznake onih grana čije struje kontrolišu stanje na izlazu elementa. Čvorovi u kolu mogu se označavati brojevima ili nizom karaktera. Oznaka 0, 00, 000, ... (niz nula proizvoljne dužine) rezervisana je za referentni čvor (masu). Oznake kontrolišućih grana identične su sa oznakom elementa koji se nalazi u toj grani (kontrolišuće grane javljaju se u opisu strujom kontrolisanih izvora i spregnutih induktivnosti).

Vrednosti parametara matematičkog modela elemenata u kolu mogu da se navedu na dva načina, u zavisnosti od složenosti modela. Ukoliko se radi o elementu koji se opisuje samo jednim parametrom, tada se navodi samo brojna vrednost tog parametra. U slučaju da model elementa kola sadrži veći broj parametara (na primer *Gummel-Poon*-ov model bipolarnog tranzistora opisuje se sa 55 parametara), ovaj način nije pogodan. Zato je uvedena mogućnost zadavanja vrednosti parametara preko *modelske kartice*. Modelska kartica definiše se posebnom komandom u kojoj se specificira njeno ime i vrednosti parametara modela. Vrednosti parametara modela pridružuju se opisu elemenata navođenjem

imena odgovarajuće modelske kartice. Ista modelska kartica može poslužiti za opis više elemenata koji imaju ekvivalentni matematički model.

Tabela 2.1: Elementi elektronskih kola koje prepoznaje simulator *PSpice*

Karakteristično slovo	Tip elementa
B	GaAsFET
C	kondenzator
D	dioda
E	NGKN
F	SGKS
G	SGKN
H	NGKS
I	nezavisni strujni generator za pobudu i napajanje
J	JFET
K	spregnute induktivnosti (transformator)
L	kalem
M	MOSFET
N	digitalni ulaz
O	digitalni izlaz
Q	bipolarni tranzistor
R	otpornik
S	naponom kontrolisani prekidač
T	vod
U	digitalna komponenta
V	nezavisni naponski generator za pobudu i napajanje
W	strujom kontrolisani prekidač
X	poziv potkola

Za definisanje modelskih kartica koristi se komanda `.MODEL`. S obzirom da se modeli različitih elemenata suštinski razlikuju, ima i više *tipova modelskih kartica*. Kod aktivnih komponenti isti element kola može imati više različitih tipova modelske kartice, zavisno od izabranog modela. Recimo, bipolarni tranzistor (element sa karakterističnim slovom Q) ima tri tipa modelske kartice: NPN, PNP i LPNP (lateralni PNP tranzistor).

Opisaćemo sintaksna pravila ulaznog jezika simulatora *PSpice* za opis samo nekih analognih elemenata (onih koji su nam potrebni u ovoj vežbi) i njihovih modelskih kartica, kao i sintaksu potrebnih komandi za kontrolu toka simulacije. Kompletne sintakse ulaznog jezika simulatora *PSpice* data je u prilogu 2. Na početku priloga 2 objašnjene su oznake koje će biti korišćenje pri navođenju sintakasnih pravila.

2.2.5 Otpornik

Sintaksa za opis otpornika je sledeća:

```
R<ime>      <(+) čvor> <(-) čvor> [ime modelske kartice]
+          <vrednost> [ TC = <TC1> [ , <TC2> ] ]
.MODEL      <ime modelske kartice> RES [parametri modela]
```

Struja kroz otpornik teče od čvora <(+) čvor> ka čvoru <(-) čvor>. Otpornik može opcionalno imati modelsku karticu tipa RES. Modelom se definiše linearna, kvadratna ili eksponencijalna zavisnost otpornosti od temperature radne okoline. Parametri modela otpornika opisani su u tabeli 2.2.

Ako nije specificirana vrednost parametra TCE, vrednost otpornosti u toku simulacije određuje se prema sledećem obrascu:

$$\text{vrednost} \cdot R \cdot [1 + TC1 \cdot (T - T_{nom}) + TC2 \cdot (T - T_{nom})^2]$$

gde je T tekuća vrednost temperature, a T_{nom} nominalna vrednost temperature radne okoline.

Tabela 2.2: Parametri modela otpornika

Ime parametra	Značenje	Jedinice	Podraz. vrednost
R	koeficijent množenja otpornosti	-	1
TC1	linearni temp. koeficijent	$^{\circ}\text{C}^{-1}$	0
TC2	kvadratni temp. koeficijent	$^{\circ}\text{C}^{-2}$	0
TCE	eksponencijalni temp. koeficijent	%/C	0

Ako je specificirana vrednost parametra TCE, vrednost otpornosti u toku simulacije određuje se prema sledećem obrascu:

$$\text{<vrednost>} \cdot R \cdot 1.01 TCE \cdot (T - T_{nom})$$

Vrednost Tnom definiše se komandom .OPTIONS (videti 2.2.15), a vrednost T može se menjati u parametarskim analizama komandama .DC (2.2.17), .TEMP (2.2.19) i .STEP (3.2.14).

2.2.6 Kondenzator

Sintaksa za opis kondenzatora je sledeća:

C<ime> <(+) čvor> <(-) čvor> [ime modelske kartice]
 + <vrednost> [IC = <početna vrednost napona>]
 .MODEL <ime modelske kartice> CAP [parametri modela]

Tabela 2.3: Parametri modela kondenzatora

Ime parametra	Značenje	Jedinice	Podraz. vrednost
C	koef. množenja kapacitivnosti	-	1
VC1	linearni naponski koeficijent	V ⁻¹	0
VC2	kvadratni naponski koeficijent	V ⁻²	0
TC1	linearni temp. koeficijent	°C ⁻¹	0
TC2	kvadratni temp. koeficijent	°C ⁻²	0

Kondenzator može opciono imati modelsku karticu kojom se definiše nelinearna zavisnost kapacitivnosti od napona na kondenzatoru i nelinearna zavisnost kapacitivnosti od temperature. Parametri modela kondenzatora dati su u tabeli 2.3. Vrednost kapacitivnosti u toku simulacije izračunava se po sledećem obrascu:

$$\text{<vrednost>} \cdot C \cdot (1 + VC1 \cdot V + VC2 \cdot V^2) \cdot [1 + TC1 \cdot (T - T_{nom}) + TC2 \cdot (T - T_{nom})^2]$$

2.2.7 Kalem

Sintaksa za opis kalema je sledeća:

L<ime> <(+) čvor> <(-) čvor> [ime modelske kartice]
 + <vrednost> [IC = <početna vrednost struje>]
 .MODEL <ime modelske kartice> IND [parametri modela]

Kalem može opciono imati modelsku karticu kojom se definiše nelinearna zavisnost induktivnosti od struje kroz kalem i nelinearna zavisnost induktivnosti od temperature. Parametri modela kalema dati su u tabeli 2.4.

Tabela 2.4: Parametri modela kalema

Ime parametra	Značenje	Jedinice	Podraz. vrednost
L	koef. množenja induktivnosti	-	1
IL1	linearni strujni koeficijent	A ⁻¹	0
IL2	kvadratni strujni koeficijent	A ⁻²	0
TC1	linearni temperaturni koeficijent	°C ⁻¹	0
TC2	kvadratni temp. koeficijent	°C ⁻²	0

Vrednost induktivnosti u toku simulacije izračunava se po sledećem obrascu:

$$\langle \text{vrednost} \rangle \cdot L \cdot (1 + IL1 \cdot I + IL2 \cdot I^2) \cdot [1 + TC1 \cdot (T - T_{nom}) + TC2 \cdot (T - T_{nom})^2]$$

2.2.8 Bipolarni tranzistor

Sintaksa za opis bipolarnog tranzistora je sledeća:

```
Q<ime>      <(kolektor) čvor> <(baza) čvor> <(emitor) čvor>
+           [(supstrat) čvor] <ime modelske kartice>
+           [vrednost parametra area]
.MODEL      <ime modelske kartice> NPN [parametri modela]
.MODEL      <ime modelske kartice> PNP [parametri modela]
.MODEL      <ime modelske kartice> LPNP [parametri modela]
```

U PSpice je ugrađen modifikovani Gummel-Poon-ov [Lit06 ili Lit09] model bipolarnog tranzistora, koji se svodi na Ebers-Moll-ov model [Lit06 ili Lit09] ako se vrednosti nekih parametara modela zanemare. Električna šema ovog modela prikazana je na slici 2.1. Modelska kartica se može definisati za tri različita tipa tranzistora: npn tranzistor, pnp tranzistor i lateralni pnp tranzistor. Parametri modelske kartice opisani su u tabeli 2.5. Uz imena nekih parametara navedene su u zagradama oznake koje se ravnopravno mogu koristiti pri definisanju vrednosti ovih parametara. Vrednosti elemenata električne šeme sa slike 2.1 izračunavaju se na osnovu 55 para metara modela (iz tabele 2.5) prema jednačinama koje se mogu naći u [91], [An88]. Parametar area, koji se opciono navodi iza imena modelske kartice, predstavlja relativnu površinu tranzistora na čipu i figuriše u jednačinama koje određuju vrednosti elemenata električne šeme sa slike 2.1. Na primer, otpornost Rc sa slike 2.1 ima vrednost RC/area, otpornost Re ima vrednost RE/area itd. Podrazumevana vrednost ovog parametra je 1.

ISC (C4)	struja curenja spoja baza-kolektor	A	0
NC	koef. emisije struje curenja spoja BC	-	2
NK	koef. opadanja k-ka pri visokim str.	-	0.5
ISS	struja zasićenja spoja supstrata	A	0
NS	koeficijent emisije struje supstrata pri nultoj polarizaciji	-	1
RE	otpornost emitora	Ω	0
RB	maksimalna otpornost baze pri nultoj polarizaciji	Ω	0
RBM	minimalna otpornost baze (pri velikoj struji)	Ω	RB
IRB	struja pri kojoj otpornost baze opada na (RB-RBM)/2.0	A	∞
RC	otpornost kolektora	Ω	0
CJE	kapacitivnost osiromašene oblasti emitorskog spoja pri nultoj polarizaciji	F	0
VJE (PE)	potencijalna barijera emitorskog spoja	V	0.75
MJE (ME)	koeficijent gradijenta spoja BE	-	0.33
CJC	kapacitivnost osirom. obl. kolektorskog spoja pri nultoj polarizaciji	F	0
VJC (PC)	potencijalna barijera kolektorskog spoja	V	0.75
MJC (MC)	koeficijent gradijenta spoja BC	-	0.33
XCJC	deo kapacitivnosti CJC povezan za unutrašnji priključak baze	-	1
CJS (CCS)	kapac. osirom. obl. spoja kolektor-supstrat pri nultoj polarizaciji	F	0
VJS (PS)	potencijalna barijera supstratskog spoja	V	0.75
MJS (MS)	koeficijent gradijenta supstratskog spoja	-	0
FC	koeficijent difuzione kapacitivnosti pri direktnoj polarizaciji	-	0.5
TF	idealno direktno vreme preleta	s	0
XTF	koeficijent zavisnosti TF-a od napona napajanja	-	0
VTF	zavisnost TF-a od napona B-C	V	∞
ITF	zavisnost TF-a od kolektorske struje	A	0
PTF	dodatni fazni pomerač na $1/2\pi TF$	0	0
TR	idealno inverzno vreme preleta	s	0

QCO	naelektrisanje epitaksijalne oblasti	C	0
RCO	otpornost epitaksijalne oblasti	Ω	0
VO	prelomni napon krive pokretljivosti nosilaca	V	10
GAMMA	faktor dopiranja epitaksijalne oblasti	-	1E-11
EG	potencijalna barijera P-N spoja	eV	1.11
XTB	temperaturni koeficijent za BF i BR	-	0
XTI (PT)	temperaturni koeficijent za IS	-	3
TRE1	linearni temperaturni koeficijent za RE	$^{\circ}\text{C}^{-1}$	0
TRE2	kvadratni temperaturni koeficijent za RE	$^{\circ}\text{C}^{-2}$	0
TRB1	linearni temperaturni koeficijent za RB	$^{\circ}\text{C}^{-1}$	0
TRB2	kvadratni temperaturni koeficijent za RB	$^{\circ}\text{C}^{-2}$	0
TRM1	linearni temperaturni koeficijent za RBM	$^{\circ}\text{C}^{-1}$	0
TRM2	kvadratni temperaturni koeficijent za RBM	$^{\circ}\text{C}^{-2}$	0
TRC1	linearni temperaturni koeficijent za RC	$^{\circ}\text{C}^{-1}$	0
TRC2	kvadratni temperaturni koeficijent za RC	$^{\circ}\text{C}^{-2}$	0
KF	koeficijent množenja za fliker šum	-	0
AF	koeficijent stepenovanja za fliker šum	-	1

2.2.9 Nezavisni pobudni generatori

PSpice prepoznaje strujni i naponski nezavisni pobudni generator. Strujni generator označava se karakterističnim slovom I, a naponski generator slovom V. Nezavisni generatori koriste se kao izvori jednosmernog napajanja ili kao pobudni generatori različitih talasnih oblika. S obzirom da su struje svih idealnih naponskih generatora izlazne veličine (prema modifikovanoj metodi nezavisnih napona), to se i struja kroz nezavisni pobudni naponski generator nalazi u spisku promenljivih u sistemu jednačina. Ova osobina može se iskoristiti da bi se analizom kola dobila vrednost neke struje. Ako je potrebno da struja kroz neku granu kola bude izlazna veličina, u tu granu kola umeće se nezavisni naponski generator nulte vrednosti.

Nezavisni generatori su jedine komponente u opisu kola koje mogu imati različite parametre za jednosmerni, naizmenični i vremenski domen rada. Generator može imati specifikaciju samo za jedan, za dva ili za sva tri režima. Opis komponente V ima sledeću formu:

```
V<ime>      <(+) čvor> <(-) čvor> [[DC] <vrednost>]
+           [AC <amplituda> [faza]]
+           [talasni oblik u vremenskom domenu]
```

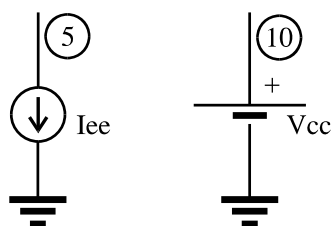
Opis komponente I ima sledeću formu:

```
I<ime>          <(+) čvor> <(-) čvor> [[DC] <vrednost>]
+               [AC <amplituda> [faza]]
+               [talasni oblik u vremenskom domenu]
```

Po konvenciji, kod opisa naponskog generatora prvo se navodi čvor generatora koji je na višem potencijalu. Kod strujnog generatora struja teče kroz generator od prvog navedenog čvora (<(+) čvor>) prema drugom navedenom čvoru (<(-) čvor>).

Prvo se navodi jednosmerna vrednost generatora. Oznaka DC nije neophodna ako se definiše samo jednosmerna vrednost generatora, to jest ako je u pitanju generator jednosmernog napajanja. Generatori prikazani na slici 2.2 opisuju se na sledeći način:

```
Vcc  10,0    DC 12V
Iee   5,0    10mA ; DC se podrazumeva
```



Slika 2.2: Jednosmerni nezavisni generatori

O načinu specificiranja talasnog oblika nezavisnih pobudnih generatora u frekvencijskom i vremenskom domenu biće reči u vežbi 3 (odeljci 3.2.3 do 3.2.8).

2.2.10 Komande za kontrolu toka i prikaz rezultata simulacije

Po konvenciji sve komande za kontrolu toka simulacije i prikaz rezultata simulacije počinju tačkom. Koriste se za:

- definisanje globalnih parametara simulacije (.OPTIONS, .PARAM, ...);
- zadavanje parametara različitih tipova analiza (.AC, .TRAN, .SENS, .OP, .DC, .STEP, .TF, .TEMP, ...);
- zadavanje formata u kome treba korisniku prezentirati rezultate simulacije (.PRINT, .PLOT, .PROBE, .WATCH, .WIDTH, ...);

- definisanje korisničkih funkcija (.FUNC), potkola (.SUBCKT, .ENDS), modelskih kartica (.MODEL);
- navođenje imena bibliotečkih i priključenih datoteka (.LIB, .INC) i drugo.

Komande koje se koriste za definisanje opštih uslova analize (.OPTIONS, .PROBE) kao i komande koje su karakteristične za specificiranje DC analize (.DC, .OP, .TF) biće opisane u nastavku, dok će ostale biti opisane u teorijskoj postavci treće vežbe.

2.2.11 Komanda .OPTIONS

Komandom .OPTIONS zadaju se vrednosti kontrolnih parametara simulacije i zahtevaju od simulatora određene akcije u toku simulacije. Recimo, menja se nominalna temperatura radne okoline, zadaje se tačnost rezultata analiza i slično. Sintaksa komande .OPTIONS je sledeća:

```
.OPTIONS [parametar]* [<parametar> = <vrednost>]*
```

U tabeli 2.6 navedeni su neki od najznačajnijih parametara komande .OPTIONS i njihovo značenje.

Tabela 2.6: Neki od parametara komande .OPTIONS

Parametar	Značenje	Jedinica	Podraz. vred.
ACCT	štampanje se statističke informacije o toku simulacije u tekstualnoj izlaznoj datoteci		
EXPAND	štampanje se opisi svih korišćenih potkola u tekstualnoj izlaznoj datoteci		
ITL1	maksimalni broj iteracija pri DC analizi		40
LIBRARY	štampanje se korišćene linije iz bibliotečkih datoteka u tekstualnoj izlaznoj datoteci		
NOECHO	sprečava štampanje listinga ulazne datoteke u tekstualnoj izlaznoj datoteci		
NOBIAS	sprečava štampanje vrednosti jednosm. napona u tekstualnoj izlaznoj datoteci		
ABSTOL	apsolutna tolerancija struja	A	1E-12
CHGTOL	apsolutna tolerancija naelektrisanja	C	1E-14

DIGFREQ	minimalni korak logičkog simulatora iznosi 1/DIGFREQ	Hz	10E+9
DIGDRVF	minimalna izlazna otpornost drajvera	Ω	2
DIGDRVZ	maksimalna izlazna otpornost drajvera	Ω	20000
DIGOVRDRV	parametar koji definiše uslov dominacije jednog drajvera nad drugim	-	
DIGMNTYMX	selekcija tipa kašnjenja za logičku simulaciju: 1-min; 2-tipično; 3-max	-	2
GMIN	minimalna dozvoljena provodnost grane	1/ Ω	1E-12
RELTOL	relativna tolerancija napona i struja	-	0.001
TNOM	nominalna temperatura radne okoline	$^{\circ}\text{C}$	300
VNTOL	apsolutna tolerancija napona	V	1E-6

2.2.12 Komanda .OP

Komandom .OP zahteva se štampanje kompletne informacije o jednosmernoj analizi u tekstualnoj izlaznoj datoteci. To obuhvata štampanje:

- vrednosti svih jednosmernih napona u kolu,
- vrednosti jednosmernih struja kroz idealne naponske generatore i ukupnu snagu kola,
- vrednosti linearizovanih parametara modela nelinearnih komponenti u jednosmernoj radnoj tački.

Ako komanda .OP nije navedena, u izlaznoj datoteci štampaće se samo vrednosti jednosmernih napona.

Komanda .OP nema parametara, sintaksa je sledeća:

.OP

2.2.13 Komanda .DC

Komanda .DC služi za zadavanje uslova analize rada kola u prisustvu sporo promenljivih signala (jednosmerni režim rada). Njome se bira parametar kola za čiji zadati skup vrednosti treba ponoviti jednosmernu analizu. Ovaj parametar može biti:

- vrednost nezavisnog naponskog/strujnog generatora (komponente V i I),

- temperatura radne okoline (globalni parametar TEMP),
- vrednost *globalnog parametra* definisanog komandom .PARAM ili
- vrednost parametra modela neke komponente.

Sintaksa komande .DC je sledeća:

```
.DC [LIN] <ime varirane promenljive>
+   <početna vrednost> <završna vrednost>
+   <vrednost koraka>
+   [višestruka specifikacija]
.DC [OCT] [DEC] <ime varirane promenljive>
+   <početna vrednost> <završna vrednost>
+   <vrednost broja tačaka>
+   [višestruka specifikacija]
.DC <ime varirane promenljive> LIST <vrednost>*
+   [višestruka specifikacija]
```

Prvi oblik komande koristi se za zadavanje linearne promene vrednosti. Specifikacija LIN se podrazumeva, pa se može izostaviti. Zadaje se apsolutna vrednost koraka promene, i ona mora biti različita od nule. Ako se zada veća početna vrednost od završne vrednosti, simulator će automatski koristiti negativan korak.

Drugi oblik komande koristi se za zadavanje logaritamske promene vrednosti. Kolo se analizira u zadatom broju tačaka po oktavi (specifikacija OCT) ili dekadi (specifikacija DEC).

Treći oblik naredbe koristi se za zadavanje neuniformnog skupa vrednosti varirane promenljive. Vrednosti za koje treba obaviti jednosmernu analizu zadaju se nabiranjem u obliku liste i ne moraju biti u rastućem ili opadajućem poretku.

Moguće je u istoj .DC komandi zadati višestruku parametarsku analizu, tako da se za svaku vrednost prve promenljive kolo analizira za više zadatih vrednosti druge promenljive (a može biti i više od dve varirane promenljive), što je označeno sa [višestruka specifikacija].

Primeri:

```
.DC LIN Vgen -0.25 0.25 0.05 ; primer 1
.DC Igen 5mA -2mA 0.1mA ; primer 2
.DC Vce 0V 10V 0.5V Ib 0mA 1mA 50uA ; primer 3
.DC RES rmodel(R) 0.9k 1.1k 50ohm ; primer 4
.DC DEC NPN qfast(IS) 1E-18A 1E-14A 5 ; primer 5
.DC TEMP LIST 0, 20, 40, 125, 27, -25 ; primer 6
.DC PARAM Vcc 4.5V 5.5V 0.1V ; primer 7
```

U prvom primeru linearno se menja vrednost generatora Vgen od -0.25V do 0.25V sa korakom 0.05V. U drugom primeru menja se vrednost generatora Igen od 5mA do -2mA, sa korakom promene -0.1mA. Specifikacija LIN može se

navesti, a ako se ne navede podrazumeva se. Treći primer ilustruje višestruku specifikaciju .DC analize. Vrednost generatora V_{ce} menja se od 0V do 10V sa korakom 0.5V, a za svaku od ovih vrednosti menja se vrednost generatora I_b od 0mA do 1mA sa korakom 50 μ A. U četvrtom primeru specificirana je promena parametra R modela otpornika (tip modela je RES) čije je ime `rmodel` od 0.9k Ω do 1.1k Ω sa korakom 50 Ω . U petom primeru zahteva se logaritamska promena vrednosti parametra I_S u modelskoj kartici `qfast` sa 5 tačaka po dekadi od 10^{-18} A do 10^{-14} A. U šestom primeru vrednost temperature uzima vrednosti iz zadate liste. U sedmom primeru menja se vrednost parametra V_{CC} definisanog komandom .PARAM od 4.5V do 5.5V sa korakom 0.1V.

2.2.14 Komanda .TF

Komandom .TF zahteva se izračunavanje pojačanja od pobudnog (naponskog ili strujnog) generatora do izlazne promenljive (napona ili struje), ulazne otpornosti i izlazne otpornosti. Za potrebe određivanja ovih veličina kolo se linearizuje u okolini jednosmerne radne tačke. Izračunati rezultati smeštaju se u tekstualnu izlaznu datoteku.

Sintaksa komande .TF je sledeća:

```
.TF <ime izlazne promenljive> <ime pobudnog generatora>
```

Parametar <ime izlazne promenljive> mora biti napon u kolu ili struja kroz idealni naponski generator. Parametar <ime pobudnog generatora> je ime pobudnog naponskog ili strujnog generatora.

Primeri:

```
.TF V(5) Vgen ; Av = V(5)/Vgen  
.TF I(Vdrive) Igen ; Ai = I(Vdrive)/Igen
```

2.2.15 Komanda .TEMP

Komandom .TEMP definiše se temperatura radne okoline. Ako se zada više od jedne temperature, sve specificirane analize (.DC, .AC, .TRAN, ...) obaviće se za svaku od zadatih vrednosti temperature. Temperatura se zadaje u $^{\circ}$ C. Sintaksa komande .TEMP je sledeća:

```
.TEMP <vrednost temperature>*
```

Primeri:

```
.TEMP 125
```

```
.TEMP 0, 27, 125
```

Ako ova komanda nije zadata, analiza se radi za sobnu temperaturu, tj. za 27°C.

2.2.16 Komanda .SENS

Komandom `.SENS` zahteva se analiza osetljivosti specificirane izlazne veličine u kolu (napon ili struja) na:

- vrednosti parametara kola (otpornosti, kapacitivnosti, induktivnosti, ...)
- vrednosti nezavisnih naponskih/strujnih generatora,
- parametre modela prekidača,
- parametre modela dioda i
- parametre modela bipolarnih tranzistora.

Sintaksa komande `.SENS` je sledeća:

```
.SENS <ime promenljive>*
```

Promenljiva čija se osetljivost izračunava može biti napon ili struja u kolu. Ako je struja, to mora biti struja kroz naponski generator (po potrebi, mogu se koristiti fiktivni naponski generatori nulte vrednosti). Rezultat komande `.SENS` smešta se u tekstualnu izlaznu datoteku.

2.2.17 Komande .INC i .LIB

Ulaznoj datoteci (ekstenzija **.cir**) može se priključiti proizvoljna datoteka korišćenjem komande `.INC` (include). Sintaksa ove komande data je u prilogu 2. Celokupni sadržaj navedene datoteke smešta se na mesto `.INC` komande pre kompajliranja. Priključena datoteka može sadržati sve *PSpice* komande osim naslovne linije i `.END` komande. Priključena datoteka može sadržati `.INC` komandu, s tim da je dubina hijerarhije priključivanja ograničena na 4 nivoa.

Definicije potkola i modelskih kartica smeštaju se u bibliotečke datoteke. Uobičajeno je da ove datoteke imaju ekstenziju **.lib**. Korišćenjem komande `.LIB` u opisu kola mogu se navesti biblioteke u kojima će *PSpice* tražiti objekte koji nisu definisani u ulaznoj datoteci. Pogodnost ovakvog pristupa je u tome što se biblioteka datoteka ne učitava cela (kao kod komande `.INC`), već se u njoj pronalaze samo potrebni objekti i oni priključuju opisu kola. Tako biblioteka datoteka može biti proizvoljno velikih dimenzija, a da to ne opterećuje rad simulatora. Sintaksa komande `.LIB` data je u prilogu 2. Treba uočiti da u jednoj `.LIB` komandi može da se navede samo jedno ime datoteke. Ako je potrebno koristiti više različitih bibliotekskih datoteka, navodi se više `.LIB` komandi. Isto

pravilo važi za komandu `.INC`. Ako u opisu kola nije navedeno ime nijedne bibliotečke datoteke, podrazumeva se datoteka **nom.lib** koja je deo standardne instalacije programskog paketa *PSpice*.

Primeri:

```
.LIB      analog.lib
.INC      funkcije
```

2.2.18 Komanda `.PROBE`

Probe je grafički postprocesor simulatora *PSpice*. To je poseban program koji služi za obradu i grafički prikaz rezultata simulacije kako na ekranu računara, tako i na papiru (korišćenjem štampača ili plotera). Komandom `.PROBE` naređuje se simulatoru da formira grafičku izlaznu datoteku sa ekstenzijom **.dat** koju kasnije obrađuje program *Probe*. Postprocesiranje je moguće samo za rezultate analiza zadatih komandama `.DC`, `.AC` (vidi odeljak 2.2.16) i `.TRAN` (vidi odeljak 2.2.17). Ostali rezultati simulacije smeštaju se u tekstualnu izlaznu datoteku. Sintaksa komande je sledeća:

```
.PROBE      [izlazna promenljiva]*
```

Ako se ne navede spisak izlaznih promenljivih, *PSpice* podrazumeva **kompletne** rezultate simulacije (vrednosti svih promenljivih iz svih obavljenih analiza). Ako se navedu izlazne promenljive, onda se u grafičku izlaznu datoteku smeštaju samo informacije o navedenim promenljivim, tako da se samo one mogu grafički prikazati.

2.2.19 Najčešće greške u opisu kola

Kako sintaksa ulaznog jezika simulatora *PSpice* nije fleksibilna, pri formiranju ulazne datoteke često se prave greške. Najčešće greške, kao i način za njihovo otklanjanje, nabrojane su u daljem tekstu.

- Izostavljanje naslovne linije. Simulator smatra naslovnom linijom prvu liniju opisa kola, a kako je to u ovom slučaju neki od elemenata kola, opis kola postaje nepotpun. Ovo može izazvati nerazumljive poruke o grešci.

- Nepotpuna ili loše zadata specifikacija biblioteke. Simulator prijavljuje da ne može da pronađe biblioteku.

- Razdvajanje broja i njegovog numeričkog sufiksa. Simulator detektuje grešku u opisu kola i javlja različite poruke, zavisno od mesta gde je greška načinjena.

Najčešće se numerički sufiks nađe na mestu imena modelske kartice i simulator prijavljuje da modelska kartica sa tim imenom nije definisana.

- Više elemenata istog tipa imaju isto ime. Simulator detektuje dupliranje imena i prijavljuje grešku.

- Nezavisni naponski generator definisan kao komponenta U (recimo U_g) umesto V . Simulator prijavljuje grešku.

- Manje od dva elementa vezano za neki čvor u kolu. Dešava se usled izostavljanja opisa neke komponente kola. Simulator prijavljuje grešku.

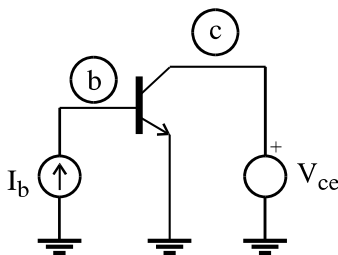
- Posle naredbe `.END` u ulaznoj datoteci postoji još nekoliko praznih redova. Simulator posle završene analize pokušava da čita opis sledećeg kola iz ulazne datoteke i prijavljuje da u kolu nema komponenti.

- Greška u redosledu navođenja čvorova pri specificiranju komponente. Ovo je najteži oblik greške, jer simulator ne detektuje grešku i analiza se izvodi do kraja (što može oduzeti dosta vremena). Greška se detektuje analizom rezultata simulacije.

Z A D A T A K

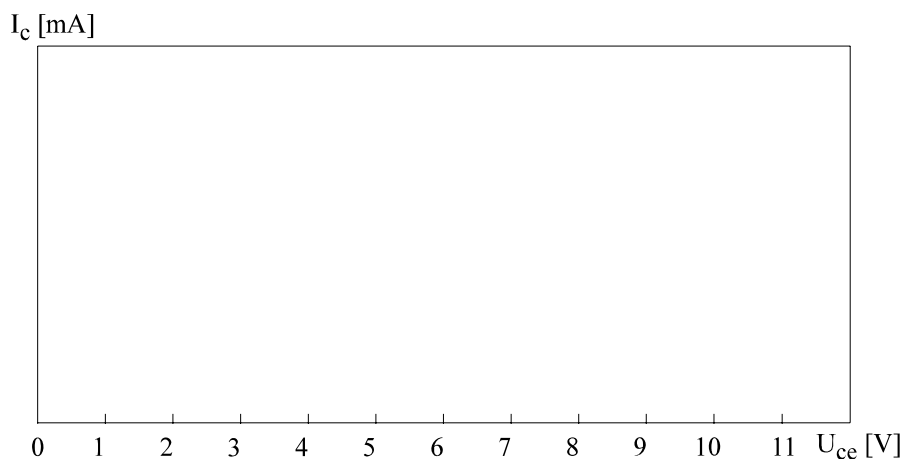
ZADATAK A: Karakterizacija tranzistora

A1. Povezati kolo za dobijanje izlaznih karakteristika tranzistora u sprezi sa zajedničkim emitorom prema slici 2.3. Koristiti model tranzistora BC107AX iz biblioteke **europa.lib** koja se nalazi na tekućem direktorijumu. Promene jednosmerne vrednosti generatora I_b i V_{ce} zadati komandom `.DC`. Struja baze tranzistora treba da uzima vrednosti od 0A do 15 μ A sa korakom 3 μ A. Napon između kolektora i emitora tranzistora treba da uzima vrednosti od 0V do 12V sa korakom 0.1V.



Slika 2.3: Šema kola za snimanje izlaznih karakteristika tranzistora

A2. Izlazne karakteristike tranzistora nacrtati na ekranu primenom grafičkog postprocesora **Probe** i skicirati ih u priloženom koordinatnom sistemu.

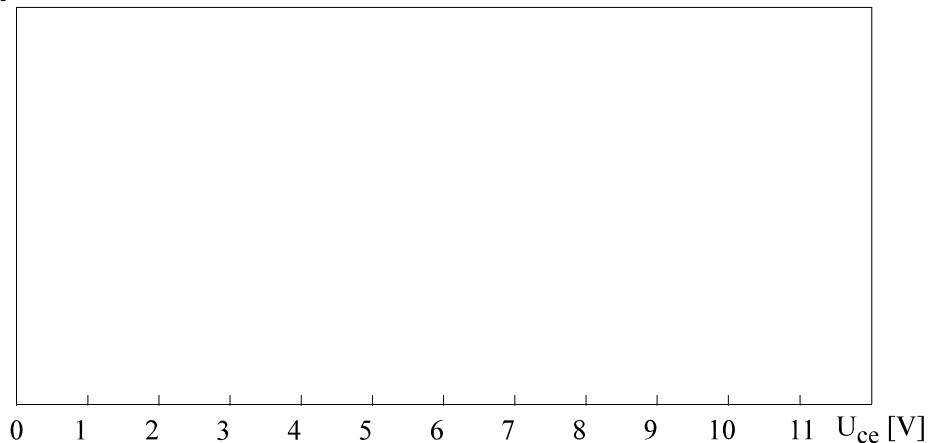


A3. Korišćenjem postprocesora **Probe** u izlazne karakteristike tranzistora ucrtati radnu pravu za slučaj da se u kolektor tranzistora veže otpornik vrednosti $3.9k\Omega$, a napajanje ima vrednost $V_{cc}=12V$. Izabrati radnu tačku tranzistora tako da tranzistor bude u linearnoj oblasti rada sa maksimalnom mogućom amplitudom signala na ulazu.

$$I_{b0} = \text{_____} \quad U_{ce0} = \text{_____} \quad I_{c0} = \text{_____}$$

A4. Analizirati uticaj promene temperature na izlazne karakteristike tranzistora. Komandom .TEMP zadati da se izvrši analiza za temperature $25^{\circ}C$ i $50^{\circ}C$. U priloženom koordinatnom sistemu skicirati sa ekrana izlazne karakteristike tranzistora na ove dve temperature.

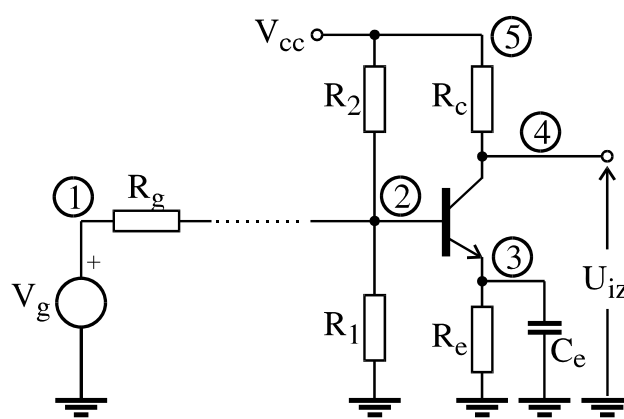
I_c [mA]



A5. Priložiti listing ulazne datoteke.

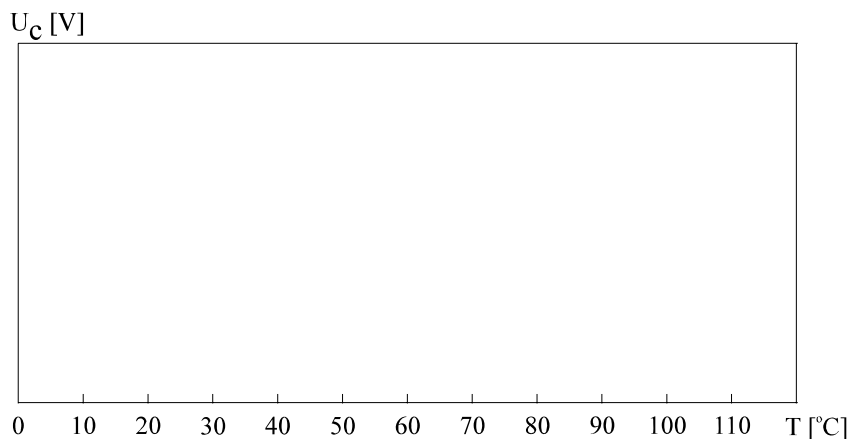
ZADATAK B: Analiza pojačavača

B1. Opisati desni deo kola sa slike 2.4 (bez generatora V_g i otpornika R_g) za simulaciju programom **PSpice**. Elementi kola sa slike 2.4 su: $V_{cc}=12V$; $R_1=10k\Omega$; $R_2=100k\Omega$; $R_c=3k\Omega$; $R_e=570\Omega$; $C_e=100\mu F$. Za tranzistor koristiti model sa imenom BC107AX iz biblioteke **europa.lib**.



Slika 2.4: Pojačavač sa zajedničkim emitorom

B2. Analizirati zavisnost jednosmernog napona na kolektoru tranzistora od temperature u opsegu od $0^{\circ}C$ do $120^{\circ}C$ sa korakom $5^{\circ}C$. Za prikaz rezultata simulacije koristiti grafički postprocesor **Probe**. Skicirati traženu zavisnost sa ekrana računara u priloženom koordinatnom sistemu.



B3. Sa grafika odrediti ukupnu promenu napona na kolektoru za ovaj temperaturni opseg i osetljivost napona na kolektoru na promenu temperature radne okoline.

$$\Delta U_c = \underline{\hspace{2cm}} \quad S_T = dU_c/dT = \underline{\hspace{2cm}}$$

B4. Odrediti osetljivost napona na kolektoru tranzistora na promenu parametara kola (komanda .SENS).

	apsolutna osetljivost	relativna osetljivost
$dU_c/dR_1 =$	<u> </u>	<u> </u>
$dU_c/dR_2 =$	<u> </u>	<u> </u>
$dU_c/dR_c =$	<u> </u>	<u> </u>
$dU_c/dR_e =$	<u> </u>	<u> </u>
$dU_c/dV_{cc} =$	<u> </u>	<u> </u>

B5. Dopuniti opis kola generatorom V_g i otpornikom R_g prema slici 2.4. Vrednost unutrašnje otpornosti generatora je $R_g=50\Omega$, a jednosmerna vrednost generatora V_g iznosi 2V. Analizirati jednosmerni režim rada kola korišćenjem komande .OP. Iz tekstualne izlazne datoteke očitati vrednosti napona čvorova, struja kroz generatore i napona i struja tranzistora.

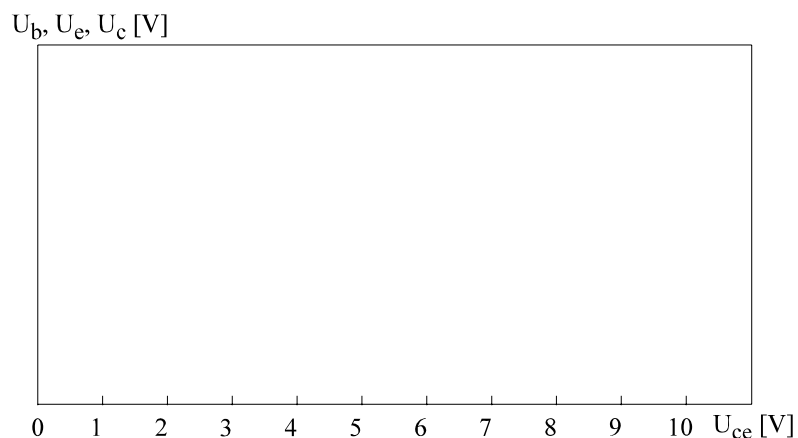
$$\begin{array}{lll} V(1) = \underline{\hspace{2cm}} & V(2) = \underline{\hspace{2cm}} & V(3) = \underline{\hspace{2cm}} \\ V(4) = \underline{\hspace{2cm}} & V(5) = \underline{\hspace{2cm}} & I(V_{cc}) = \underline{\hspace{2cm}} \end{array}$$

$I(V_g) =$ _____ $I_b =$ _____ $I_c =$ _____

$V_{be} =$ _____ $V_{bc} =$ _____ $V_{ce} =$ _____

U kom režimu radi tranzistor? _____

B6. Analizirati kolo u jednosmernom režimu kad se vrednost generatora V_g menja od 0 do 10V sa korakom 0.1V. Koristeći postprocesor **Probe** na ekranu nacrtati zavisnost napona na bazi, emitoru i kolektoru tranzistora od jednosmerne vrednosti generatora V_g . Očitati vrednosti napona u karakterističnim tačkama i skicirati grafik sa ekrana u priloženom koordinatnom sistemu. Na dobijenom dijagramu označiti režime rada tranzistora (zakočenje, aktivni režim i zasićenje).



B7. Odrediti pojačanje kola, ulaznu i izlaznu otpornost kola za male signale.

$A = U_{iz} / V_g =$ _____ $R_{ul} =$ _____ $R_{iz} =$ _____

B8. Priložiti listing ulazne datoteke.

Datum : _____

Student : _____

Overava : _____

Vežba broj 3

Analiza analognih elektronskih kola u frekvencijskom i vremenskom domenu programom PSpice

3.1 Cilj vežbe

Upoznati se sa osnovnim principima analize elektronskih kola u frekvencijskom i vremenskom domenu pomoću računara. U tu svrhu koristiti hibridni simulator *PSpice*. Proučiti mogućnosti parametarske analize, analize harmonijskih izobličenja, definisanje potkola i upotrebu kontrolisanih generatora za modelovanje specifičnih elektronskih komponenti.

3.2 Teorijska postavka vežbe

3.2.1 Električna analiza kola u frekvencijskom domenu

Analiza kola u frekvencijskom domenu podrazumeva da se kolo pobuđuje pomoću jednog ili više generatora (strujnih ili naponskih) sinusnog talasnog oblika iste frekvencije. Generatorima treba definisati amplitudu i početni fazni stav. Podrazumeva se da su amplitude generatora dovoljno male, te da se sve nelinearne komponente kola mogu smatrati linearnim u okolini mirne radne tačke. Nelinearne komponente zamenjuju se odgovarajućim linearnim modelima. Da bi se odredili parametri ovih modela potrebno je izvršiti DC analizu kola. Dakle, pre početka AC analize automatski se obavi DC analiza bez intervencije korisnika.

Pri analizi u AC režimu traži se odziv kola za više različitih frekvencija kako bi se skicirale amplitudске i fazne frekvencijske karakteristike. Očigledno je zato da ovaj tip analize zahteva višestruko rešavanje sistema linearnih jednačina. Pri tome treba imati na umu da su promenljive kompleksne veličine i da se tokom analize mogu javiti problemi vezani za pojavu lokalnih rezonansi u nekim internim čvorovima [Lit00].

3.2.2 Električna analiza kola u vremenskom domenu

Kod analize kola u vremenskom domenu traži se odziv kola na zadate talasne oblike pobudnih generatora. Kao što je rečeno u poglavlju 2.2.1, kolo se u vremenskom domenu opisuje sistemom diferencijalnih jednačina koji može biti linearan ili nelinearan u zavisnosti od toga da li je kolo linearno ili nelinearno. Sistem se rešava tako što se željeni vremenski interval za koji se traži odziv kola diskretizuje na manje intervale shodno koraku numeričke integracije. Zatim se u svakoj tački na vremenskoj osi rešava sistem jednačina. Rešavanje nelinearnog sistema jednačina svodi se na iterativni postupak u okviru koga se potreban broj puta ponavlja rešavanje sistema linearnih jednačina. Granične vrednosti za $t=0$ određuju se automatski DC analizom kola.

Ilustrujmo kompleksnost analize u vremenskom domenu na jednom primeru. Za analizu nekog kola u intervalu od 0 do 1ms, sa prosečnim korakom integracije od 1 μ s, potrebno je 1000 puta rešiti sistem jednačina. Ukoliko je kolo nelinearno, u svakom koraku analize (tački na vremenskoj osi) treba iterativno rešiti sistem nelinearnih jednačina. Uz pretpostavku da je potrebno prosečno samo 5 iteracija za dobijanje rešenja, sledi da treba rešiti sistem linearnih jednačina 5000 puta. Imajući ovo u vidu, očigledno je da analiza nelinearnih kola u vremenskom domenu može da zahteva dosta računarskog vremena.

Uz analizu u vremenskom domenu može se izvršiti i Fourier-ova analiza signala u kolu i sagledati harmonijska izobličenja signala [Lit06].

U nastavku će biti reči o delu sintakse ulaznog jezika simulatora **PSpice** koja je potrebna za simulaciju električnih kola u frekvencijskom i vremenskom domenu.

3.2.3 Nezavisni pobudni generatori u frekvencijskom i vremenskom domenu

Kao što smo rekli u poglavlju 2.2.9, posle specifikacije jednosmerne vrednosti (koja nije obavezna) definiše se ponašanje generatora u frekvencijskom domenu. Iza oznake AC navode se dva parametra talasnog oblika generatora: amplituda (u V ili u A) i početna faza (zadaje se u stepenima), kao u sledećim primerima:

```
Vgen 10, 0 DC 12V AC 0.1V, 45degree
```

Ig 3, 0 AC 10mA ; podrazumevana vrednost faze je 0 stepeni

Zatim sledi specifikacija talasnog oblika pobudnog generatora za analizu u vremenskom domenu. Na raspolaganju je pet različitih talasnih oblika koji se prepoznaju po sledećim oznakama:

- PULSE - trapezni (impulsni) talasni oblik
- SIN - sinusni talasni oblik
- PWL - segmentno linearni talasni oblik
- EXP - eksponencijalni talasni oblik
- SFFM - frekvencijski modulisan talasni oblik

U nastavku će biti reči o načinu specificiranja pobudnih generatora sa ovim talasnim oblicima na izlazu.

3.2.4 Generator trapeznog talasnog oblika

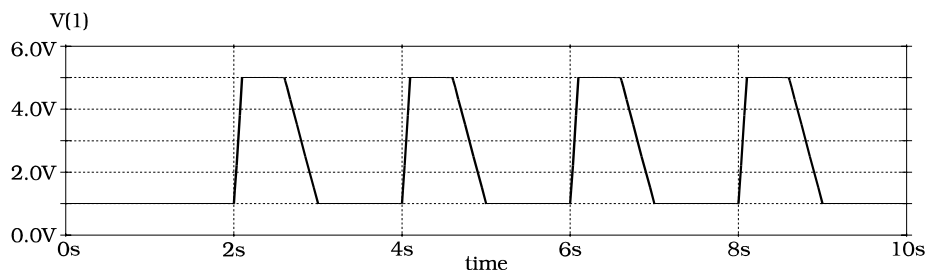
Trapezni talasni oblik opisuje se pomoću sedam parametara: prvim naponskim nivoom v_1 , drugim naponskim nivoom v_2 , kašnjenjem početka generisanja impulsa t_d , trajanjem rastuće ivice t_r , trajanjem opadajuće ivice t_f , trajanjem pozitivnog impulsa p_w i periodom ponavljanja p_{er} . Parametrima t_r, t_f, p_w i p_{er} ne sme se zadati vrednost 0s. Početna vrednost napona generatora je v_1 . Neka je $v_1 < v_2$ (moguće je i $v_1 > v_2$). Tada t_r predstavlja vreme potrebno da napon generatora linearno promeni vrednost od v_1 do v_2 , a t_f predstavlja vreme potrebno da napon generatora linearno promeni vrednost od v_2 do v_1 . Parametar p_w predstavlja vremenski period u toku koga generator ima konstantnu vrednost v_2 . Sintaksa za opis trapezoidnog talasnog oblika je sledeća:

PULSE(< v_1 > < v_2 > < t_d > < t_r > < t_f > < p_w > < p_{er} >)

Primer:

Vgpulse 1,0 PULSE(1V, 5V, 2s, 0.1s, 0.4s, 0.5s, 2s)

Talasni oblik koji generiše generator Vgpulse prikazan je na slici 3.1.



Slika 3.1: Talasni oblik na izlazu trapeznog generatora Vgpulse

3.2.5 Generator sinusnog talasnog oblika

Sinusni talasni oblik opisuje se na sledeći način:

`SIN (<voff> <vamp1> <freq> <td> <df> <phase>)`

Parametar `voff` definiše napon jednosmernog ofseta, `vamp1` je amplituda sinusoide, `freq` frekvencija, `td` je kašnjenje početka generisanja sinusoide, `df` je faktor slabljenja amplitude i `phase` je početni fazni stav generatora (zadaje se u stepenima). Vrednosti parametara `voff` i `vamp1` moraju se zadati, podrazumevana vrednost frekvencije je $1/tstop$, gde je `tstop` trajanje simulacije zadato u komandi `.TRAN`, a za parametre `td`, `df` i `phase` podrazumeva se vrednost nula, ako im vrednost nije zadata. Sinusni talasni oblik može se opisati sledećim jednačinama:

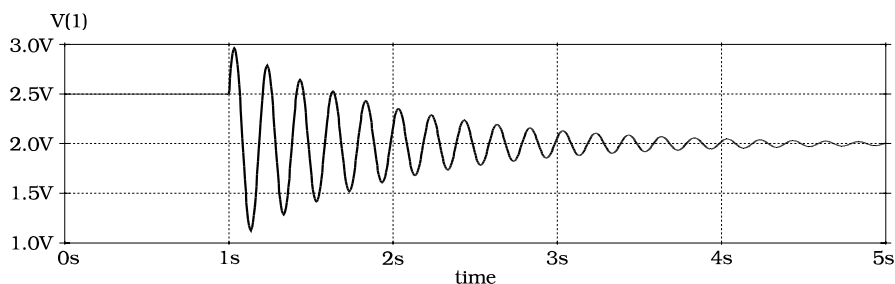
$$V_{iz} = v_{off} + v_{amp1} \cdot \sin(2\pi \cdot \text{phase}/360^\circ) \quad \text{za } t \leq t_d$$

$$V_{iz} = v_{off} + v_{amp1} \cdot e^{-(t-t_d)} \cdot df \cdot \sin[2\pi \cdot (\text{freq} \cdot (t-t_d) + \text{phase}/360^\circ)] \quad \text{za } t \geq t_d$$

Primer:

`Vgsin 1, 0 SIN(2V, 1V, 5Hz, 1s, 1, 30degree)`

Talasni oblik koji generiše generator Vgsin prikazan je na slici 3.2.



Slika 3.2: Talasni oblik na izlazu sinusnog generatora Vgsin

3.2.6 Generator eksponencijalnog talasnog oblika

Generator eksponencijalnog talasnog oblika generiše impuls sa eksponencijalnom rastućom i opadajućom ivicom. Sintaksa za opis eksponencijalnog generatora je sledeća:

EXP (<v1> <v2> <td1> <tc1> <td2> <tc2>)

Parametar v_1 je početni napon, v_2 je napon za vreme trajanja impulsa, t_{d1} je kašnjenje početka generisanja impulsa, t_{c1} je vremenska konstanta rastuće ivice, t_{d2} je vreme završetka rastuće ivice, odnosno početka opadajuće ivice i t_{c2} je vremenska konstanta opadajuće ivice. Generator na izlazu drži napon v_1 do trenutka t_{d1} , zatim napon eksponencijalno raste prema v_2 sa vremenskom konstantom t_{c1} sve do trenutka t_{d2} , kad počinje eksponencijalno opadanje ka naponu v_1 sa vremenskom konstantom t_{c2} . Opadanje traje do kraja simulacije, odnosno do t_{stop} . Generiše se samo jedan, pozitivni ili negativni (zavisno od vrednosti napona v_1 i v_2) impuls. Talasni oblik koji daje generator eksponencijalnog impulsa može se opisati jednačinama:

$$V_{iz} = v_1 \quad \text{za } t \leq t_{d1}$$

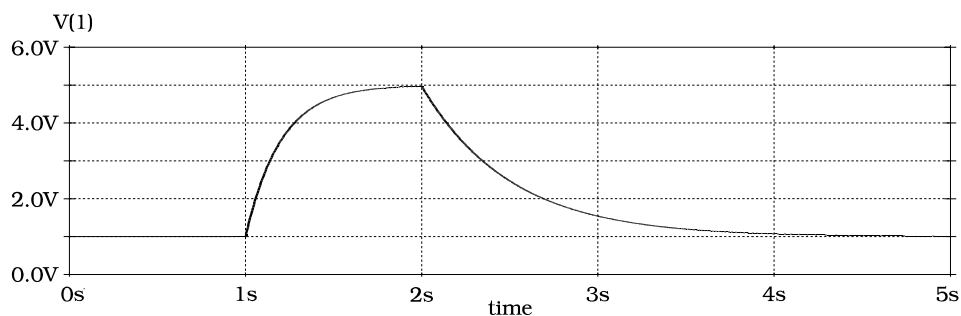
$$V_{iz} = v_1 + (v_2 - v_1) \cdot [1 - e^{-(t - t_{d1})/t_{c1}}] \quad \text{za } t_{d1} \leq t \leq t_{d2}$$

$$V_{iz} = v_1 + (v_2 - v_1) \cdot [(1 - e^{-(t - t_{d1})/t_{c1}}) - (1 - e^{-(t - t_{d2})/t_{c2}})] \quad \text{za } t_{d1} \leq t \leq t_{d2}$$

Primer:

Vgexp 1, 0 EXP (1V, 5V, 1s, 0.2s, 2s, 0.5s)

Talasni oblik koji generiše generator Vgexp prikazan je na slici 3.3.



Slika 3.3: Talasni oblik na izlazu generatora eksponencijalnog impulsa Vgexp

3.2.7 Generator frekvencijski modulisanog talasnog oblika

Frekvencijski modulisani talasni oblik opisuje se na sledeći način:

SFFM (<voff> <vamp1> <fc> <mod> <fm>)

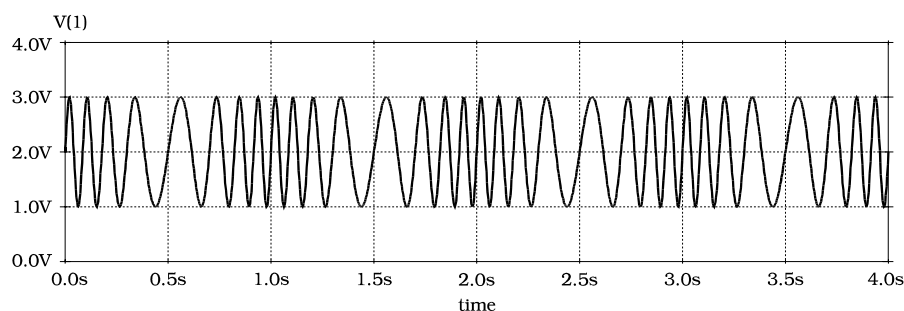
Parametar *voff* definiše napon jednosmernog ofseta, *vamp1* je amplituda nosećeg sinusnog signala, *fc* je frekvencija nosećeg signala, *mod* je indeks modulacije, a *fm* frekvencija modulišućeg sinusnog niskofrekventnog signala. Talasni oblik koji daje generator frekvencijski modulisanog signala definisan je sledećom jednačinom:

$$V_{iz} = v_{off} + v_{amp1} \cdot \sin(2\pi \cdot f_c \cdot t + \text{mod} \cdot \sin(2\pi \cdot f_m \cdot t))$$

Primer:

Vgsffm 1, 0 SFFM (2V, 1V, 8Hz, 4, 1Hz)

Talasni oblik koji daje generator Vgsffm prikazan je na slici 3.4.



Slika 3.4: Izlaz generatora frekvencijski modulisanog talasnog oblika Vgsffm

3.2.8 Generator segmentno-linearnog talasnog oblika

Segmentno-linearni talasni oblik opisuje se na sledeći način:

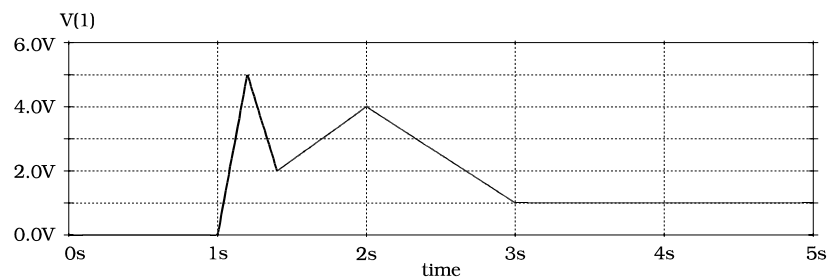
PWL (<t1> <v1> <t2> <v2> <t3> <v3> ... <tn> <vn>)

Talasni oblik koji daje ovakav generator sastoji se od linearnih segmenata između zadatih tačaka. Svaka tačka zadaje se parom vrednosti: (*vreme*, *napon*). U definiciji generatora tačke su poredane po rastućim vremenima. Nije dozvoljeno da dve tačke imaju isto vreme (vertikalna ivica). Prva tačka mora definisati vrednost u nultom trenutku simulacije. Od poslednje definisane tačke, pa do kraja simulacije napon je konstantan, isti kao u poslednjoj definisanoj tački.

Primer:

```
Vgpgwl 1,0 PWL (0s, 0V 1s, 0V 1.2s, 5V
+          1.4s, 2V 2s, 4V 3s, 1V)
```

Talasni oblik koji daje generator Vgpgwl prikazan je na slici 3.5.



Slika 3.5: Izlaz generatora segmentno-linearnog talasnog oblika Vgpgwl

3.2.9 Kontrolisani generatori

PSpice prepoznaje četiri tipa kontrolisanih generatora. To su elementi:

- E** - naponski generator kontrolisan naponom (NGKN)
- F** - strujni generator kontrolisan strujom (SGKS)
- G** - strujni generator kontrolisan naponom (SGKN)
- H** - naponski generator kontrolisan strujom (NGKS)

3.2.10 NGKN i SGKN

Opšti oblik opisa NGKN je sledeći:

```
E<ime> <(+) čvor> <(-) čvor> <(+) kontrolišući čvor>
+ <(-) kontrolišući čvor> <pojačanje>
E<ime> <(+) čvor> <(-) čvor> POLY(<vrednost>)
+ <<(+) kontrolišući čvor> <(-) kontrolišući čvor>>*
+ <vrednost koeficijenta polinoma>*
E<ime> <(+) čvor> <(-) čvor> VALUE = {<izraz> }
E<ime> <(+) čvor> <(-) čvor> TABLE { <izraz> } =
+ < <vrednost ulaza> , <vrednost izlaza> >*
E<ime> <(+) čvor> <(-) čvor> LAPLACE { <izraz> } =
+ { <transformacija> }
E<ime> <(+) čvor> <(-) čvor> FREQ { <izraz> } =
+ < <frekvencija> , <amplituda> , <faza> >*
```

Od šest mogućih oblika ovde ćemo opisati samo prva dva. Prvi oblik predstavlja specijalni slučaj - linearni naponski generator kontrolisan jednim naponom. Ovakav generator ima samo jedan parametar - <pojačanje>. Sledeći

primer predstavlja naponski generator vezan između čvorova 7 i 3 kontrolisan naponom između čvorova 1 i 2, sa pojačanjem 3.45 puta.

$$\text{Ectrl } 7,3 \quad 1,2 \quad 3.45 \quad ; \quad V(7,3) = 3.45 \, V(1,2)$$

Drugi oblik koristi se za definisanje nelinearnog naponskog generatora kontrolisanog proizvoljnim brojem kontrolišućih napona. Nelinearna zavisnost definiše se u polinomskom obliku. Parametar <vrednost> određuje broj kontrolišućih napona. Parovi čvorova kojima se specificiraju kontrolišući naponi mogu se smestiti i u zagrade. Vrednost generatora kontrolisanog naponima $V_1, V_2, \dots, V_n$ opisuje se u polinomskom obliku na sledeći način:

$$\begin{aligned} V_{\text{out}}(V_1, V_2, \dots, V_n) = & \\ & p_0 + \\ & + p_1 V_1 + p_2 V_2 + \dots + p_n V_n + \\ & + p_{n+1} V_1 V_1 + p_{n+2} V_1 V_2 + \dots + p_{n+n} V_1 V_n + \\ & + p_{2n+1} V_2 V_2 + p_{2n+2} V_2 V_3 + \dots + p_{2n+n-1} V_2 V_n + \\ & + \dots + \\ & + p_n!/(2(n-2)!)+2n V_n V_n + \\ & + p_n!/(2(n-2)!)+2n+1 V_1^2 V_1 + p_n!/(2(n-2)!)+2n+1 V_1^2 V_2 + \dots \end{aligned}$$

Za slučaj naponskog generatora kontrolisanog jednim naponom ovaj izraz se svodi na sledeći oblik:

$$V_{\text{out}}(V) = p_0 + p_1 V + p_2 V^2 + \dots + p_n V^n + \dots$$

Talasni oblik napona kontrolisanog generatora definiše se zadavanjem vrednosti koeficijenata polinoma p_i , $i=0,1,\dots$. Podrazumevane vrednosti svih koeficijenata p_i su 0. Svi koeficijenti se ne moraju navesti, ali se ni jedan ne sme preskočiti. Na primer, ranije definisani generator Ectrl može se definisati u sledećem obliku:

$$\begin{aligned} \text{Ectrl } 7,3 \text{ POLY}(1) \quad 1,2 \quad 0.0 \quad 3.45 \quad ; \\ V(7,3) = 0.0 + 3.45 V(1,2) \end{aligned}$$

Evo još nekoliko primera.

Četvoroulazni naponski sabirač vezan između čvorova sa oznakama 100 i 101, definisan funkcijom

$$V(100,101) = V(1) + V(2) + V(3) + V(4)$$

može se opisati na sledeći način:

$$\begin{aligned} \text{Esum4 } 100,101 \text{ POLY}(4) \quad (1,0) \quad (2,0) \quad (3,0) \quad (4,0) \\ + \quad 0.0, \quad 1.0, \quad 1.0, \quad 1.0, \quad 1.0 \end{aligned}$$

Dvoulazni množač napona vezan između čvorova sa oznakama 100 i 101, definisan funkcijom

$$V(100, 101) = V(1) V(2)$$

može se opisati na sledeći način:

```
Emult2 100,101 POLY(2) (1,0) (2,0) 0.0, 0.0, 0.0, 0.0, 1.0
```

Kvadrator napona u čvoru 1 vezan između čvorova 100 i 101, definisan funkcijom

$$V(100, 101) = (V(1))^2$$

opisuje se na sledeći način:

```
Esquare 100,101 POLY(1) (1,0) 0.0, 0.0, 1.0
```

Sintaksa za opis strujnog generatora kontrolisanog naponom (element G) slična je, pa je nećemo posebno opisivati (videti prilog 2).

3.2.11 SGKS i NGKS

Opšti oblik opisa SGKS je sledeći:

```
F<ime> <(<+> čvor> <(<-> čvor>  
+ <ime kontrolišućeg V elementa> <pojačanje>  
F<ime> <(<+> čvor> <(<-> čvor> POLY(<vrednost>)  
+ <ime kontrolišućeg V elementa>*<  
+ <vrednost koeficijenta polinoma>*<
```

Prvi oblik predstavlja linearni SGKS, a drugi predstavlja generalni nelinearni SGKS. Da bi kontrolišuće struje bile promenljive u sistemu jednačina, **PSpice** postavlja uslov da to budu struje kroz idealne naponske generatore konstantne vrednosti (element V). Ovi generatori mogu imati i nultu vrednost. Pozitivna struja kontrolisanog strujnog generatora ističe iz čvora <(<+> čvor>, teče **kroz generator** i utiče u <(<-> čvor>.

Sledeća naredba opisuje linearni SGKS vezan između čvorova 6 i 4 (smer struje je od čvora 6 kroz generator prema čvoru 4) koji 5.0 puta pojačava struju kroz naponski generator Vg:

```
Fctrl 6,4 Vg 5.0 ; Iout = 5.0 I(Vg)
```

Isti SGKS može se opisati korišćenjem POLY notacije na sledeći način:

```
Fctrl 6,4 POLY(1) Vg 0.0, 5.0 ; Iout=0.0 + 5.0  
I(Vg)
```

Sintaksa za opis naponskog generatora kontrolisanog strujom (element H) slična je, pa je nećemo posebno opisivati (videti prilog 2).

3.2.12 Komanda .PARAM

Sintaksa komande .PARAM je sledeća:

```
.PARAM < <ime> = <vrednost> >*<  
.PARAM < <ime> = { <izraz> } >*<
```

Komandom `.PARAM` definišu se globalni parametri simulacije. Dva su osnovna motiva za definisanje globalnog parametra. Prvo, to može biti neka konstanta koja se pri opisivanju simuliranog kola često pojavljuje u izrazima, pa je pogodno skratiti pisanje. Na primer:

```
.PARAM pi=3.14159, pi2={2.0 * pi}
```

Druga primena globalnih parametara odnosi se na zadavanje parametarske analize komandom `.STEP` (odjeljak 3.2.14). Na primer:

```
.PARAM Vcc=12V
.STEP LIN PARAM Vcc 10V, 14V, 0.1V
```

Globalni parametar može figurisati u izrazima, a takođe se može definisati korišćenjem izraza. Izraz se uvek smešta u vitičaste zagrade `{ ... }`. Osim globalnih parametara, realnih brojeva i operacija, u izrazu se mogu javiti i ugrađene standardne funkcije, kao što su: `SIN`, `COS`, `LOG`, `LOG10`, `EXP`, `ABS` (videti prilog 3). Izraz ne sme biti duži od jedne linije. U slučaju da izraz mora biti duži od jedne linije, neophodno je koristiti komandu `.FUNC` (odjeljak 3.2.13) kojom se delovi izraza definišu kao nove funkcije i tako se skraćuje njihovo pisanje.

3.2.13 Komanda `.FUNC`

Sintaksa komande `.FUNC` je sledeća:

```
.FUNC <ime> ( [<argument>*] ) <telo funkcije>
```

Ovom komandom definišu se nove funkcije potrebne za modelovanje sistema. Funkcija ne mora imati argumente, ali su zagrade obavezne. Maksimalan broj argumenata je 10. Komanda `.FUNC` u kojoj je definisana funkcija mora prethoditi prvoj upotrebi te funkcije. Definicija funkcije ne sme biti duža od jedne linije, ali može sadržati pozive drugih, ranije definisanih funkcija. **U telu funkcija ne smeju se koristiti linijski komentari.** Ne smeju se definisati dve funkcije sa istim imenom. Ne može se definisati nova funkcija sa istim imenom kao neka od standardnih funkcija. Standardne funkcije koje prepoznaje **PSpice** nabrojane su u prilogu 3.

Primeri:

```
.FUNC e(x)          EXP(x)
.FUNC sinh(x)       (e(x)-e(-x)) / 2.0
.FUNC min(a,b)      (a+b-ABS(a-b)) / 2
.FUNC F()           1/(6.28*SQRT(L*C))
```

3.2.14 Komanda .STEP

Komanda .STEP izaziva ponavljanje svih navedenih analiza (komande .DC, .AC, .TRAN, ...) za zadati skup vrednosti promenljivog parametra. Rezultati svih analiza biće smešteni u istu izlaznu datoteku, jedan za drugim. Grafički postprocesor **Probe** prihvata rezultate parametarske analize i omogućava grafičko prikazivanje familija krivih.

Sintaksa komande .STEP je sledeća:

```
.STEP [LIN] <ime varirane promenljive>
+      <početna vrednost> <završna vrednost>
+      <vrednost koraka>

.STEP [OCT] [DEC] <ime varirane promenljive>
+      <početna vrednost> <završna vrednost> <vrednost>

.STEP <ime varirane promenljive> LIST <vrednost>*
```

Parametar <ime varirane promenljive> može biti:

- ime nezavisnog pobudnog generatora (komponente V i I),
- parametar modela komponente (svi parametri osim L i W za MOSFET i temperaturnih koeficijenata TC1 i TC2 za komponente kod kojih ovi koeficijenti mogu da se definišu),
- temperatura (promenljiva TEMP) ili
- globalni parametar (definisan komandom .PARAM).

Argumentima <početna vrednost> i <završna vrednost> zadaje se početna i krajnja vrednost promenljivog parametra. Postoje tri načina promene vrednosti parametra: linearna promena, logaritamska promena i korišćenje liste vrednosti. Kod linearne promene (vidi prvi oblik naredbe) vrednost parametra se u svakom narednom prolazu uvećava za fiksni korak <vrednost koraka>. Kod logaritamske promene (drugi oblik naredbe) analiza se obavlja u <vrednost> tačaka po oktavi (OCT) ili dekadi (DEC). Kod zadavanja liste vrednosti parametra (treći oblik naredbe) analiza će biti ponovljena za sve vrednosti parametra zadate u listi.

Primeri:

```
.STEP Vg 0V 10V .5V
.STEP LIN Ig 5mA -2mA 0.1mA ; negativan korak -0.1mA
.STEP DEC NPN qfast(IS) 1E-18 1E-14 5
.STEP TEMP LIST 0, 20, 30, 40, 75, 125, -50
.STEP PARAM eta 0.2 1.6 0.1
.STEP RES rmod(R) 1k 5k 0.5k
```

Postoje dva načina zadavanja parametarske analize sa promenom vrednosti otpornosti. Prvi način zahteva da se definiše model otpornika, pa da se menja vrednost parametra modela, kao u sledećem primeru:

```
Rp 1,2 rmodel
.MODEL rmodel RES (R=1)
.STEP RES rmodel(R) 30ohm, 50ohm, 2ohm
```

Drugi način je da se vrednost otpornosti definiše .PARAM naredbom, pa da se menja kao globalni parametar:

```
.PARAM Rvalue=1
Rp 1,2 {Rvalue}
.STEP PARAM Rvalue 30ohm, 50ohm, 2ohm
```

3.2.15 Komanda .SUBCKT

Elektronsko kolo koje treba analizirati može biti složene strukture u kojoj se pojedine grupe elemenata više puta ponavljaju. Da bi se pojednostavio opis takvog kola, **PSpice** dozvoljava grupisanje elemenata u gradivne blokove koji se nazivaju *potkola* (koristi se i termin *makro*). Potkolo može sadržati osnovne *PSpice* elemente i/ili druga potkola. Na taj način može se generisati hijerarhijska organizacija opisa strukture kola proizvoljne dubine. Komandom .SUBCKT definiše se struktura potkola, a jednom definisano potkolo poziva se kao element sa oznakom X. Potkolo mora biti definisano u ulaznoj datoteci pre poziva. Definicija potkola može se nalaziti u okviru opisa kola (recimo na početku) ili u bibliotčkoj datoteci. Potkola smeštena u biblioteke moraju se učiniti dostupnim korišćenjem komande .LIB.

Opšti oblik definicije potkola je sledeći:

```
.SUBCKT <ime potkola> [čvor]*
+ [ OPTIONAL: <<čvor>=<podrazumevana vrednost>>* ]
+ [ PARAMS: < <ime> = <vrednost> >* ]
+ [ TEXT: < <ime> = <tekstualna vrednost> >* ]
... ; struktura potkola
.ENDS [ime potkola]
```

Opšti oblik poziva potkola je sledeći:

```
X<ime> [čvor]* <ime potkola>
+ [ PARAMS: < <ime> = <vrednost> >* ]
+ [ TEXT: < <ime> = <tekstualna vrednost> >* ]
```

Opis potkola počinje komandom .SUBCKT, a završava se komandom .ENDS. Između ove dve komande smeju se nalaziti samo definicije elemenata i/ili pozivi drugih potkola. Oznake čvorova koje se koriste u definiciji potkola formalne su i

pri pozivu potkola biće zamenjene zadatim aktuelnim čvorovima. Ne sme se koristiti čvor sa oznakom 0, jer je ova oznaka rezervisana za globalni čvor uzemljenja.

Iza ključne reči `OPTIONAL` navode se opcioni čvorovi, sa svojim predefinisanim imenima. Ako se pri pozivu potkola ovi čvorovi ne navedu u listi čvorova, biće korišćeni predefinisani čvorovi. Na primer, potkolo 74LS00 definisano na sledeći način:

```
.SUBCKT 74LS00 a, b, y OPTIONAL: dpwr=$g_dpwr,
dgnd=$g_dgnd
..; u opisu potkola koriste se čvorovi a, b, y, dpwr i
dgnd
.ENDS 74LS00
```

može se pozvati kao:

```
X1 in1, in2, out 74LS00
```

pri čemu se koriste globalni digitalni čvorovi napajanja `$g_dpwr` i `$g_dgnd`, (videti 4.2.16) ili kao:

```
X1 in1, in2, out, newpwr, newgnd 74LS00
```

pri čemu se spolja definišu čvorovi `newpwr` i `newgnd` kojima će biti zamenjeni interni čvorovi `dpwr` i `dgnd` pri pozivu potkola.

Opis potkola može se generalizovati tako što će se u potkolo preneti vrednosti parametara nabrojane iza ključne reči `PARAMS`. Imena parametara iza reči `PARAMS` mogu se proizvoljno koristiti u izrazima unutar definicije potkola. U sledećem primeru definisano je potkolo sa imenom `oppoj` koje predstavlja operacioni pojačavač sa generalizovanim vrednostima ulazne i izlazne otpornosti:

```
.SUBCKT oppoj plus,minus,out PARAMS: riz=100ohm,
rul=2megohm
Rin plus,minus {rul}
Rout out1,out {riz}
Eout out1,0 plus,minus 10000 ; pojačanje
.ENDS oppoj
```

Ovako definisano potkolo `oppoj` može se pozvati sa različitim vrednostima parametara `Riz` i `Rul`:

```
X1 10, 12, 15 oppoj params: riz=50, rul=200k
X2 15, 16, 25 oppoj params: rul=500k
```

Kod potkola `x2` u gornjem primeru ulazna otpornost će biti 500kΩ, dok će izlazna otpornost imati podrazumevanu vrednost 100Ω.

Iza ključne reči `TEXT` mogu se navesti imena i podrazumevane vrednosti tekstualnih parametara koji se mogu koristiti u tekstualnim izrazima unutar definicije potkola.

Između ključnih reči `OPTIONAL`, `PARAMS` i `TEXT` i znaka dve tačke (:) ne sme biti razmaka.

U izlaznoj datoteci štampa se samo ime potkola, osim ako je u `.OPTIONS` naredbi korišćena opcija `EXPAND` koja prouzrokuje da se u izlaznoj datoteci štampaju opisi svih pozivanih potkola u potpunom obliku.

3.2.16 Komanda `.AC`

Komandom `.AC` zahteva se analiza u frekvencijskom domenu i zadaju potrebni parametri analize. Sintaksa komande je sledeća:

```
.AC [LIN] [OCT] [DEC] <vrednost>  
+ <početna vrednost frekvencije>  
+ <završna vrednost frekvencije>
```

Mora se izabrati jedan od sledećih načina promene frekvencije: linearna (`LIN`), logaritamska po oktavi (`OCT`) ili logaritamska po dekad (decadi) (`DEC`). Parametar `<vrednost>` definiše u koliko tačaka treba analizirati kolo. Uz `LIN` specifikaciju to je ukupan broj tačaka na frekvencijskoj osi u kojima će kolo biti analizirano, a kod `OCT` i `DEC` specifikacija to je broj tačaka po oktavi, odnosno dekadi.

Rezultati AC analize mogu se pregledati korišćenjem komandi `.PRINT`, `.PLOT` i `.PROBE`. Ako je grupno kašnjenje izlazna veličina simulacije, treba obezbediti da uzastopne frekvencije budu dovoljno bliske, jer *PSpice* grupno kašnjenje (izvod faze po frekvenciji) računa oduzimanjem faze u dve uzastopne tačke i deljenjem razlikom frekvencija.

Ponovo podsećamo da svakoj AC analizi prethodi DC analiza kako bi se odredile vrednosti dinamičkih parametara nelinearnih elemenata u radnoj tački.

3.2.17 Komanda `.TRAN`

Komandom `.TRAN` zahteva se analiza u vremenskom domenu i zadaju se njeni parametri. Sintaksa komande je sledeća:

```
.TRAN[/OP]<vrednost koraka štampanja>  
+ <vreme završetka> [ (vreme početka štampanja)  
+ [maksimalna vrednost koraka] ] [UIC]
```

Kao što je ranije rečeno, pre početka analize u vremenskom domenu obavezno se obavlja DC analiza da bi se odredili granični uslovi. Ako želimo da se rezultati ove analize prikažu u potpunom obliku u tekstualnoj izlaznoj datoteci, koristićemo komandu `.TRAN/OP`, čiji je efekat ekvivalentan upotrebi komandi `.OP` i `.TRAN` zajedno.

Parametar `<vrednost koraka štampanja>` je vremenski interval štampanja rezultata simulacije (interval štampanja). Za analizu u vremenskom domenu *PSpice* koristi promenljivi vremenski korak koji se ne prilagođava

intervalu štampanja. Zato se vrednost izlaznih promenljivih u trenucima koje određuje interval štampanja izračunava polinomskom interpolacijom drugog reda.

Simulacija u vremenskom domenu počinje u trenutku 0.0s. Parametar <vreme završetka> definiše vreme završetka simulacije **t_{stop}**.

Parametar <vreme početka štampanja> definiše koliko dugo od početka simulacije ne treba štampati rezultate simulacije. Štampanje počinje u trenutku <vreme početka štampanja>, a završava se u <vreme završetka>.

S obzirom da mehanizam automatske kontrole vrednosti koraka simulacije omogućava simulatoru da uvećava korak simulacije, korisniku je ostavljena mogućnost da parametrom <maksimalna vrednost koraka> definiše maksimalnu dozvoljenu vrednost koraka simulacije.

Specifikacija UIC nalaže simulatoru da pri DC analizi kojom se određuju granični uslovi za analizu u vremenskom domenu koristi zadate vrednosti napona na kondenzatorima i struja kroz kalemове u kolu.

3.2.18 Komanda .NOISE

Komandom .NOISE zahteva se analiza šuma u kolu. Pošto šumovi u kolu zavise od frekvencije pobudnog generatora, analiza šuma obavlja se zajedno sa AC analizom. Zato komanda .NOISE ne može da se zada bez komande .AC u kojoj su specificirane frekvencije na kojima treba analizirati kolo. Izvori šumova u kolu su otpornici i poluprovodničke komponente. Rezultat analize šuma je vrednost pojačanja kola za male signale, vrednost ekvivalentnog napona šuma na izlazu kola (promenljiva ONOISE) i vrednost ekvivalentnog generatora šuma na ulazu (promenljiva INOISE). Ekvivalentni napon šuma na izlazu predstavlja napon koji se javlja na izlazu kola samo usled šumova koje stvaraju elementi kola (kada svi pobudni generatori u kolu imaju vrednost nula). Izražava se u jedinicama $V/\sqrt{\text{Hz}}$. Vrednost ekvivalentnog generatora šuma jeste vrednost koju treba da ima strujni ili naponski generator na izabranom ulazu kola, tako da se na izlazu idealnog kola (u kome komponente ne generišu šum) dobije napon jednak ekvivalentnom naponu šuma koji generiše realno kolo bez pobude. Izražava se u $V/\sqrt{\text{Hz}}$ ili $A/\sqrt{\text{Hz}}$ u zavisnosti od toga da li je pobudni generator naponski ili strujni. Rezultati analize šuma dobijaju se u tekstualnoj izlaznoj datoteci. Sintaksa komande je sledeća:

```
.NOISE V(<čvor> [, <čvor>]) <ime> [<interval štampanja>]
```

Komandu .NOISE objasnićemo najlakše pomoću primera:

```
.NOISE V(5) Vgen 10 ; primer 1
```

```
.NOISE V(4, 2) Ig ; primer 2
```

Prvi parametar komande .NOISE definiše izlaz kola. Izlaz se zadaje kao napon nekog čvora u kolu (V(5) u primeru 1, izlaz je između čvora 5 i mase) ili kao napon između proizvoljne dve tačke u kolu (V(4, 2) u drugom primeru).

Drugi parametar je ime nezavisnog pobudnog generatora (komponente V i I) na čijem mestu treba sračunati vrednost ekvivalentnog pobudnog generatora šuma. U primeru 1 to je naponski generator V_{gen} , a u primeru 2 strujni generator I_g .

Treći parametar komande .NOISE je interval štampanja. U primeru 1 on ima vrednost 10 što znači da u svakoj 10-toj tački AC analize treba štampati rezultate analize šuma. Ako se interval štampanja izostavi, neće se štampati rezultati analize šuma u toku analize, ali se oni mogu štampati korišćenjem komande .PRINT (videti odeljak 3.2.20).

3.2.19 Komanda .FOUR

Komandom .FOUR zahteva se Fourier-ova analiza talasnog oblika nekih promenljivih i izračunavanje koeficijenata harmonijskih izobličenja. Za svaku zahtevanu promenljivu simulator će izračunati i odštampati u tekstualnoj izlaznoj datoteci vrednost jednosmerne komponente, amplitudu osnovnog harmonika i amplitude drugog do devetog harmonika. Takođe, simulator računa i štampa klir faktore pojedinih harmonika i ukupni klir faktor.

Sintaksa komande je sledeća:

```
.FOUR <vrednost frekvencije osnovnog harmonika>  
<promenljiva>*
```

Parametri komande su frekvencija osnovnog harmonika i niz promenljivih koje se zadaju kao u komandi .PRINT. Za Fourier-ovu analizu neophodno je da se raspolaze rezultatima analize u vremenskom domenu. Zato u ulaznoj datoteci koja sadrži komandu .FOUR mora da postoji i komanda .TRAN da bi Fourier-ova analiza bila moguća.

Fourier-ova analiza obavlja se posle završetka analize u vremenskom domenu koristeći talasne oblike potrebnih promenljivih. Za Fourier-ovu analizu koristi se samo vremenski interval od kraja analize pa unazad za $1/\text{<vrednost frekvencije osnovnog harmonika>}$ i podrazumeva se da se ovaj period beskonačno ponavlja. Da bi Fourier-ova analiza bila validna potrebno je da ukupno trajanje analize ne bude kraće od ovog perioda i da to bude perioda ponavljanja signala u kolu.

Primer:

```
.FOUR 10kHz V(5) V(4, 3) I(Vsens)
```

3.2.20 Komanda .PRINT

Komandom .PRINT zadaje se lista izlaznih veličina čije vrednosti u toku analize treba štampati u tekstualnu izlaznu datoteku. Sintaksa komande je sledeća:

```
.PRINT[/DGTLCHG] [DC] [AC] [NOISE] [TRAN] [izlazna  
promenljiva]*
```

Komanda .PRINT štampa vrednosti specificiranih promenljivih (bilo analognih, bilo digitalnih) sa zadatim korakom štampanja (kod analize u vremenskom domenu to je <vrednost koraka štampanja>, kod AC analize štampa se rezultat za svaku frekvenciju, pri parametarskoj analizi štampa se rezultat za svaku novu vrednost parametra i slično). U ulaznoj datoteci može se naći više od jedne .PRINT naredbe, pri čemu će se štampati sve zahtevane promenljive iz svih naredbi.

Kod .PRINT komande obavezno se navodi samo jedan tip analize na koju se komanda odnosi. Komandom .PRINT moguće je štampati sledeće kategorije izlaznih veličina:

- Napon u čvoru. Na primer: V(1), V(a_node).
- Napon između dva čvora. Na primer: V(3,4), V(1,a_node).
- Napon na komponenti sa dva priključka (otpornik, kondenzator, ...). Na primer: V(R1), V(Cs), V(D2).
- Napon na priključku elementa sa više pristupa (tranzistori). Za razlikovanje priključaka komponenta postoje odgovarajući sufiksi. Recimo, sufiksi za pristupne krajeve bipolarnog tranzistora su B, C, E i S (supstrat), sufiksi za priključke MOSFET-a i JFET-a su G, D, S i B (bulk). Na primer: VB(Q3) je napon na bazi bipolarnog tranzistora Q3, dok je VD(J23) napon na drejnu JFET-a J23.
- Napon između dva priključka elementa sa više pristupa. Na primer: VBE(Q3) je napon između baze i emitora tranzistora Q3, VDS(J23) je napon između drejna i sorsa JFET-a J23.
- Struja kroz element sa dva pristupa. Na primer: I(Rc1) je struja kroz otpornik Rc1, I(Cs) je struja kroz kondenzator Cs. Kod načina označavanja pristupnih krajeva elemenata sa dva pristupa usvojenog u prilogu 2 smer struje je od <(+) čvor> kroz element prema <(-) čvor>.
- Struja koja utiče u priključak elementa sa više od dva pristupa. Priključak se specificira navođenjem odgovarajućeg sufiksa. Na primer: IB(Q3) je struja koja utiče u bazu tranzistora Q3, ID(J23) je struja koja utiče u drejn JFET-a J23.

Kod AC analize rezultat su kompleksni brojevi, pa je moguće štampati sledeće veličine:

- Amplituda (moduo) napona ili struje. Sufiks je M. Ako nije zadat drugi sufiks, za rezultate AC analize se podrazumeva sufiks M, pa može da se izostavi. Dozvoljene su i kombinacije sa ranije pomenutim sufiksima. Na primer: VM(1) je moduo napona u čvoru 1, VCEM(Q3) je moduo napona između kolektora i emitora bipolarnog tranzistora Q3, ekvivalentna oznaka je VCE(Q3), IM(Rb) je moduo struje kroz otpornik Rb.

- Amplituda (moduo) napona ili struje u dB. Sufiks je DB. Na primer: VDB(1) je vrednost modula napona V(1) izražena u dB ($VDB(1) = 20\log_{10}VM(1)$), VGSDB(M2) je vrednost napona između gejta i sorsa MOSFET-a M2 izražena u dB, IDB(Rc) je vrednost struje kroz otpornik Rc izražena u dB.

- Faza napona ili struje. Sufiks je P. Na primer: VP(1) je faza napona u čvoru 1, VEP(Q2) je faza napona na emitoru tranzistora Q2, IP(R1) je faza struje kroz otpornik R1.

- Grupno kašnjenje (osetljivost faze na promenu frekvencije). Sufiks je G. Na primer: VG(1) je grupno kašnjenje napona u čvoru 1.

- Realni deo napona ili struje. Sufiks je R. Na primer: VR(Rc2) je realni deo napona na otporniku Rc2, VR(3,5) je realni deo napona između čvorova 3 i 5, ICR(Q3) je realni deo kolektorske struje tranzistora Q3.

- Imaginarni deo napona ili struje. Sufiks je I. Na primer: VI(Rc2) predstavlja imaginarni deo napona na otporniku Rc2, VI(3,5) je imaginarni deo napona V(3,5), II(Vg) je imaginarni deo struje kroz naponski generator Vg.

Rezultat analize šuma (komanda .NOISE) su dve promenljive čije je vrednosti moguće štampati u izlaznoj datoteci:

- Promenljive ONOISE (ekvivalentni napon šuma na izlazu) i INOISE (vrednost ekvivalentnog generatora šuma na ulazu). Ove promenljive mogu se štampati i u dB, pri čemu je notacija DB(<ime promenljive>).

Primeri:

```
.PRINT DC V(3),V(2,3),V(R1),I(Vin),I(R2),IB(Q2),VBE(Q41)
.PRINT TRAN Vbe(Q2), Ic(Q2)
.PRINT AC V(3) VP(3) VR(3) VI(3) VG(5) IR(Vgen) II(Vgen)
.PRINT NOISE ONOISE INOISE DB(ONOISE) DB(INOISE)
```

3.2.21 Komanda .PLOT

Komandom .PLOT moguće je zahtevati od simulatora da u tekstualnoj izlaznoj datoteci "nacrt" talasne oblike zahtevanih promenljivih uz pomoć ASCII karaktera. Ova komanda potiče iz vremena kad nije postojao grafički postprocesor **Probe** i omogućava grubi vizuelni pregled talasnih oblika, ali je za iole ozbiljniju

grafičku obradu izlaznih rezultata neophodan grafički postprocesor. Sintaksa komande `.PLOT` je sledeća:

```
.PLOT [DC] [AC] [NOISE] [TRAN] [izlazna promenljiva]*  
+    ( [<vrednost donje granice>,  
+    <vrednost gornje granice>] )*
```

U jednoj `.PLOT` naredbi specificira se samo jedan tip analize, jedna ili više izlaznih promenljivih i, opciono, opseg vrednosti ovih promenljivih na y-osi. Opseg se zadaje posle svake navedene promenljive ili posle grupe promenljivih za koju važi. Ako opseg nije zadat, y-osa se automatski skalira shodno minimalnoj i maksimalnoj vrednosti promenljivih u listi. Može biti više od jedne `.PLOT` naredbe u listi opisa kola.

Primeri:

```
.PLOT DC V(3) V(2,3) V(R1) I(Vin) I(R2) IB(Q2) VBE(Q41)  
.PLOT AC V(3) VP(3) VR(3) VI(3) VG(5) IR(Vgen) II(Vgen)  
.PLOT TRAN Vce(Q2) (0V, 12V) Ic(Q2), I(Vcc) (-50mA, 50mA)  
.PLOT NOISE INOISE, ONOISE, DB(INOISE), DB(ONOISE)
```

3.2.22 Komanda `.WATCH`

Komanda `.WATCH` omogućava da se rezultati DC, AC i TRAN analiza posmatraju na ekranu **u toku analize**. Sintaksa komande je sledeća:

```
.WATCH [DC] [AC] [TRAN] [ <promenljiva>  
+ [(<vrednost donje granice>,  
+ <vrednost gornje granice>)] ]*
```

Prvi parametar komande specificira tip analize i mora biti jedna od vrednosti DC, AC ili TRAN. U ulaznoj datoteci mogu se naći tri `.WATCH` komande, po jedna za svaki tip analize. Drugi parametar komande je ime promenljive. Ono se zadaje u istom formatu kao kod `.PRINT` komande uz ograničenje da ne mogu da se zadaju digitalne promenljive i grupno kašnjenje. Iza imena promenljive može opciono da sledi opseg u kome mora da se nalazi vrednost promenljive u toku analize. Ukoliko opseg nije naveden, podrazumeva se $(-\infty, +\infty)$. U toku simulacije simulator na ekranu ispisuje trenutnu vrednost zahtevane promenljive. U slučaju da vrednost promenljive izađe izvan zadatog opsega simulator zaustavlja simulaciju i postavlja korisniku pitanje da li simulaciju treba nastaviti ili prekinuti. Ako se simulacija nastavi, narušena granica opsega promenljive se ne proverava u nastavku analize. U komandi `.WATCH` može se navesti do osam promenljivih sa odgovarajućim opsezima. Svi zadati opsezi će biti proveravani u toku simulacije, ali će se na ekranu prikazivati vrednosti samo prve tri promenljive.

Primeri:

```
.WATCH DC V(3) (-1V, 5V) V(2, 3) VC(Q1)
```

```
.WATCH AC VM(2) VP(2) VR(2) VI(2)
```

3.2.23 Komande .IC i .NODESET

Kao što je rečeno, analiza u vremenskom domenu predstavlja numeričku integraciju diferencijalnih jednačina. Numerička integracija zahteva zadavanje početnih uslova, to jest početnih vrednosti promenljivih u sistemu jednačina. Simulator pokušava da početne uslove odredi samostalno - DC analizom kola. Ponekad je ovaj pokušaj neuspešan, pa je potrebno simulatoru "pomoći" da dođe do početnih uslova. Ili početni uslovi mogu biti nejednoznačni, pa u tom slučaju treba izabrati jedno od mogućih rešenja. Zadavanje početnih uslova može se izvesti na tri načina:

- komandom .IC
- komandom .NODESET ili
- zadavanjem vrednosti parametra IC za kondenzatore i kalemове.

Komande .IC i .NODESET imaju sledeću sintaksu:

```
.IC < V(<čvor> [, <čvor>]) = <vrednost> >*
.NODESET < V(<čvor> [, <čvor>]) = <vrednost> >*
```

Komandama .IC i .NODESET mogu se zadati naponi u nekim ili svim čvorovima u kolu ili naponi između nekih čvorova u kolu. Komanda .IC **nameće** početnu vrednost. Između mase i čvora za koji je komandom .IC zadat početni napon (ili između dva čvora) simulator dodaje rednu vezu naponskog generatora zadate vrednosti i otpornika vrednosti 0.002Ω . Ovi elementi ostaju u kolu dok se obavi DC analiza kojom se određuju početni uslovi.

Komanda .NODESET **pomaže** simulatoru da pronađe početna rešenja. Postupak za određivanje početnih vrednosti promenljivih je isti kao kod .IC komande, ali se posle određivanja nametnutih rešenja dodatni generatori i otpornici odstranjuju iz kola, pa se DC analiza ponavlja startujući od nametnutih početnih rešenja. U ovoj dodatnoj DC analizi promenljive mogu da promene vrednosti, tako da konačna početna rešenja ne moraju biti jednaka onima koja su zadata u .NODESET komandi. Ako se u opisu kola zadaju obe komande, komanda .NODESET se ignoriše. Komanda .NODESET ima efekta i kod analize statičkih karakteristika kola (zadate komandom .DC) gde se koristi kao pomoć za izračunavanje prve tačke, a u daljoj analizi se ignoriše.

Primeri:

```
.IC V(5)=12V V(3, 7)=2.5V
.NODESET V(IN1, IN2)=1.0E-3 V(103, 0)=5V
```

Pri definiciji kondenzatora i kalemova mogu se specificirati granični uslovi: napon na kondenzatoru i struja kroz kalem. Ako se u komandi .TRAN navede

parametar `UIC`, onda se analiza u vremenskom domenu startuje koristeći samo ove početne uslove (bez DC analize).

3.2.24 Najčešće greške u opisu kola

Najčešće greške, kao i načini za njihovo otklanjanje, nabrojane su u daljem tekstu.

- Zadavanje prekratkog trajanja analize u vremenskom domenu. Analiza u vremenskom domenu obično zahteva da neki talasni oblik iz generatora "prođe" kroz kolo i generiše odziv na izlazu. Dakle, trajanje analize treba da bude dovoljno dugo da bi se odziv pojavio na izlazu.

- Pogrešno zadate granične vrednosti u komandi `.IC`. Ovo narušava validnost analize. Ako nismo sigurni u vrednosti nekih graničnih vrednosti, bolje je zadati ih komandom `.NODESET`.

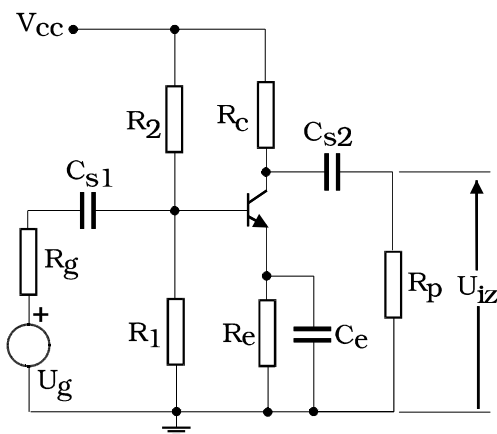
- Zadavanje linearnog inkrementiranja frekvencije (specifikacija `LIN`) u vrlo širokom opsegu frekvencije u `.AC` komandi. Ova greška ima za posledicu da se na niskim frekvencijama kolo analizira sa nesrazmerno velikim korakom što analizu čini besmislenom. Kada se traži AC analiza za frekvencijski opseg od nekoliko dekada, mora se zadati logaritamski inkrement frekvencije (specifikacije `DEC` ili `OCT`).

- U komandi `.NOISE` ne navede se interval štampanja rezultata, a ne zahteva se štampanje rezultata analize šuma ni pomoću komandi `.PRINT` i `.PLOT`. U ovom slučaju rezultat analize šuma se nigde ne beleži i njeno izvršenje je besmisleno.

Z A D A T A K

ZADATAK A: Analiza pojačavača

A1. Elementi kola sa slike 3.6 su: $V_{cc}=12V$; $R_1=10k\Omega$; $R_2=100k\Omega$; $R_c=3k\Omega$; $R_e=570\Omega$; $R_p=5k\Omega$; $R_g=200\Omega$; $C_{s1}=C_{s2}=1000\mu F$; $C_e=100\mu F$. Opisati kolo za analizu simulatorom **PSpice**. Koristiti model tranzistora BC107AX iz biblioteke **europa.lib** koja se nalazi na tekućem direktorijumu.



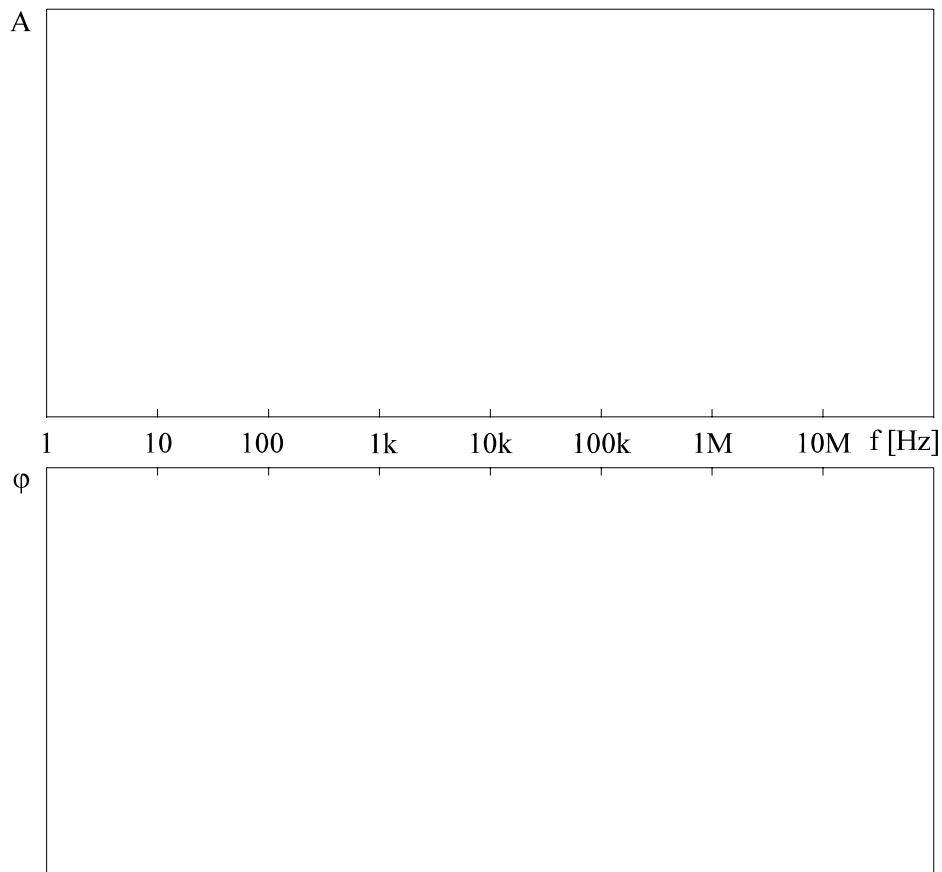
Slika 3.6: Pojačavač sa zajedničkim emitorom

A2. Analizirati kolo sa slike 3.6 u AC domenu za opseg frekvencija od 1Hz do 100MHz. Amplituda pobudnog generatora ima vrednost 0.01V. Izvršiti analizu šumova u kolu u jednoj tački po dekadi. Kolika je vrednost ekvivalentnog napona šuma na izlazu i ekvivalentnog generatora šuma na ulazu pri frekvenciji 1kHz?

$$ONOISE(1kHz) = \underline{\hspace{2cm}} \quad INOISE(1kHz) = \underline{\hspace{2cm}}$$

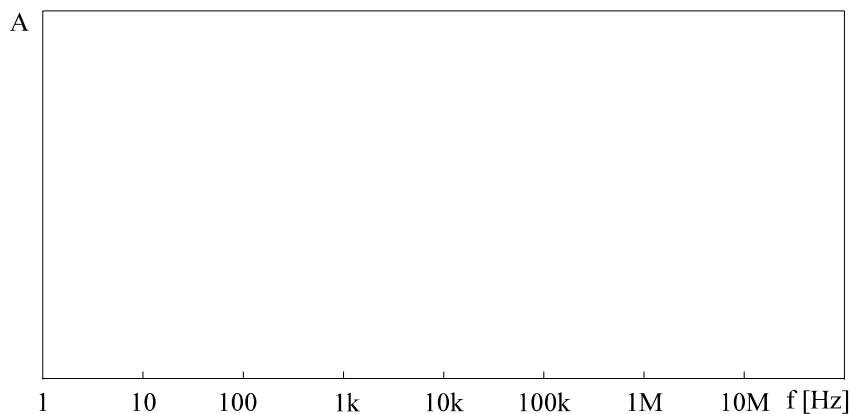
A3. Nacrtati na ekranu računara amplitudsku i faznu frekvencijsku karakteristiku korišćenjem grafičkog postprocesora **Probe**. Skicirati tražene karakteristike u priloženom koordinatnom sistemu. Odrediti maksimalno pojačanje u propusnom opsegu i granične frekvencije pojačavača.

$$A = \underline{\hspace{2cm}} \quad f_d = \underline{\hspace{2cm}} \quad f_g = \underline{\hspace{2cm}}$$



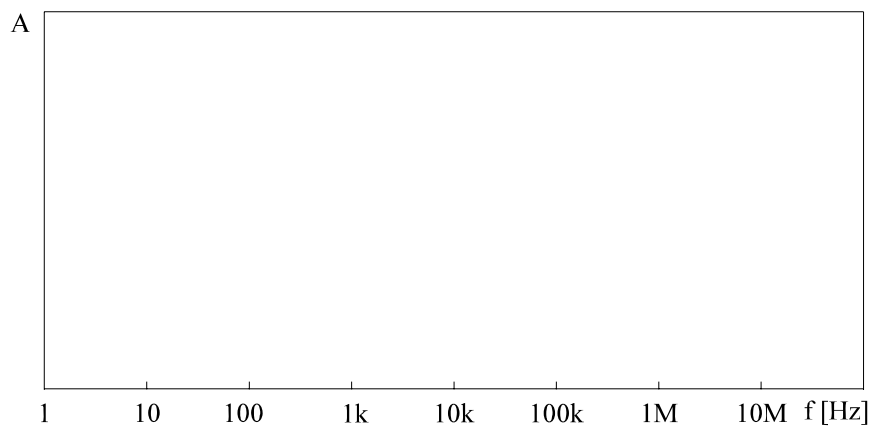
A4. Analizirati kolo u frekvencijskom domenu kada otpornik R_c uzima vrednosti: $3k\Omega$, $3.3k\Omega$, $4k\Omega$. U priloženom koordinatnom sistemu skicirati amplitudsku frekvencijsku karakteristiku kola za ove tri vrednosti otpornika R_c . Kolika je maksimalna vrednost pojačanja u ova tri slučaja?

$A(3k) = \underline{\hspace{2cm}}$ $A(3.3k) = \underline{\hspace{2cm}}$ $A(4k) = \underline{\hspace{2cm}}$



A5. Ponoviti analizu u AC domenu kada kondenzator C_e uzima vrednosti: $50\mu\text{F}$, $100\mu\text{F}$ i $200\mu\text{F}$. Skicirati amplitudne frekvencijske karakteristike kola za ove tri vrednosti kondenzatora C_e u priloženom koordinatnom sistemu. Kolika je vrednost donje granične frekvencije u ova tri slučaja?

$$f_d(50\mu\text{F}) = \underline{\hspace{2cm}} \quad f_d(100\mu\text{F}) = \underline{\hspace{2cm}} \quad f_d(200\mu\text{F}) = \underline{\hspace{2cm}}$$



A6. Analizirati ponašanje kola sa slike 3.6 u vremenskom domenu kada se kolo pobudi sinusnim generatorom amplitude 0.01V i frekvencije 100Hz . Analizu završiti posle 30ms , a korak analize ograničiti na 0.01ms (interval štampanja rezultata može biti i veći). Zadati da se izvrši Fourier-ova analiza talasnog oblika

napona na potrošaču za frekvenciju osnovnog harmonika jednaku frekvenciji pobudnog generatora.

Koliko iznosi:

Jednosmerna komponenta izlaznog napona: _____

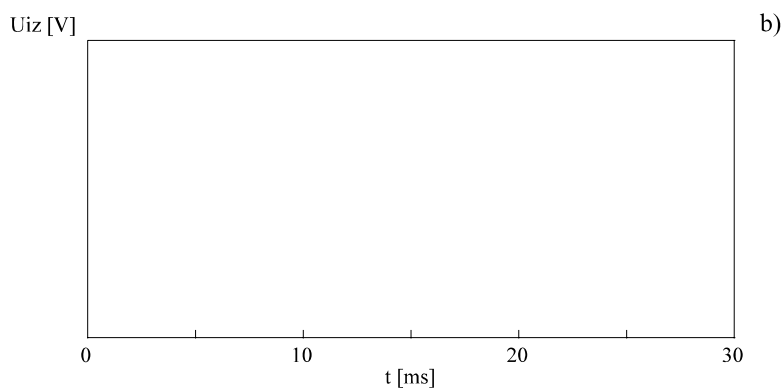
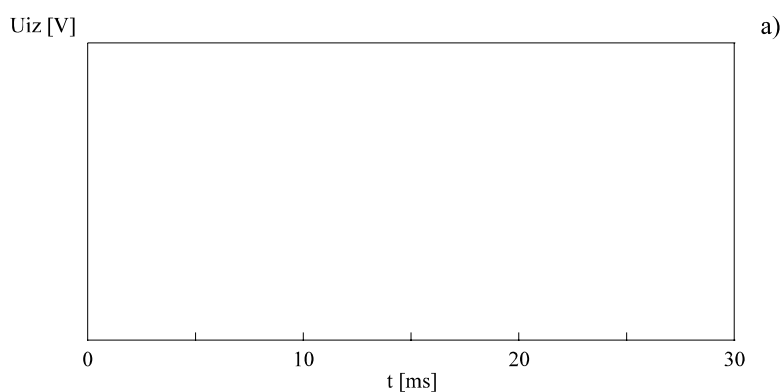
Amplituda 1. harmonika: _____

Amplituda 2. harmonika: _____ Klir faktor 2. harmonika: _____

Amplituda 3. harmonika: _____ Klir faktor 3. harmonika: _____

Ukupni klir faktor: _____

A7. Uporediti talasne oblike napona na generatoru i napona na izlazu korišćenjem grafičkog postprocesora **Probe**. U priloženom koordinatnom sistemu (grafik označen sa a) skicirati talasni oblik izlaznog napona.



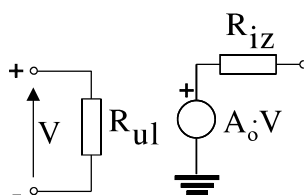
A8. Ponoviti analizu sa deset puta većom vrednošću amplitude sinusnog generatora (0.1V). U priloženom koordinatnom sistemu skicirati talasni oblik izlaznog napona (grafik označen sa b). Komentarisati razlike u talasnim oblicima na graficima a) i b).

Komentar:

A9. Priložiti listing ulazne datoteke.

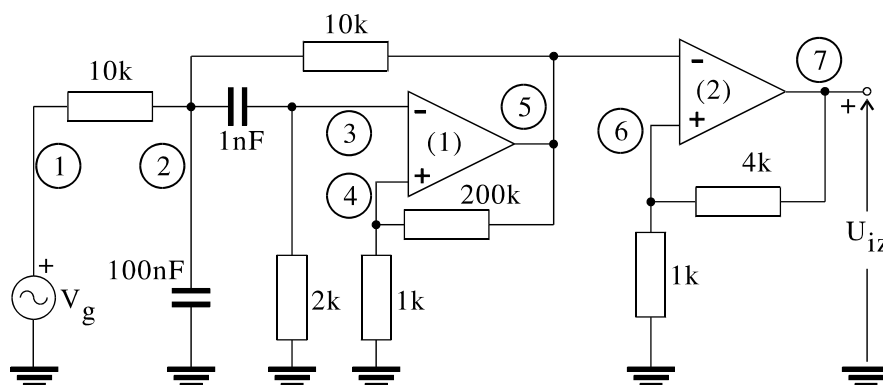
ZADATAK B: Analiza filtra

B1. Model operacionog pojačavača prikazan na slici 3.7 opisati kao potkolo. Vrednosti elemenata modela su: $R_{ul}=2\text{M}\Omega$, $R_{iz}=100\Omega$ i $A_o=10000$.



Slika 3.7: Model operacionog pojačavača

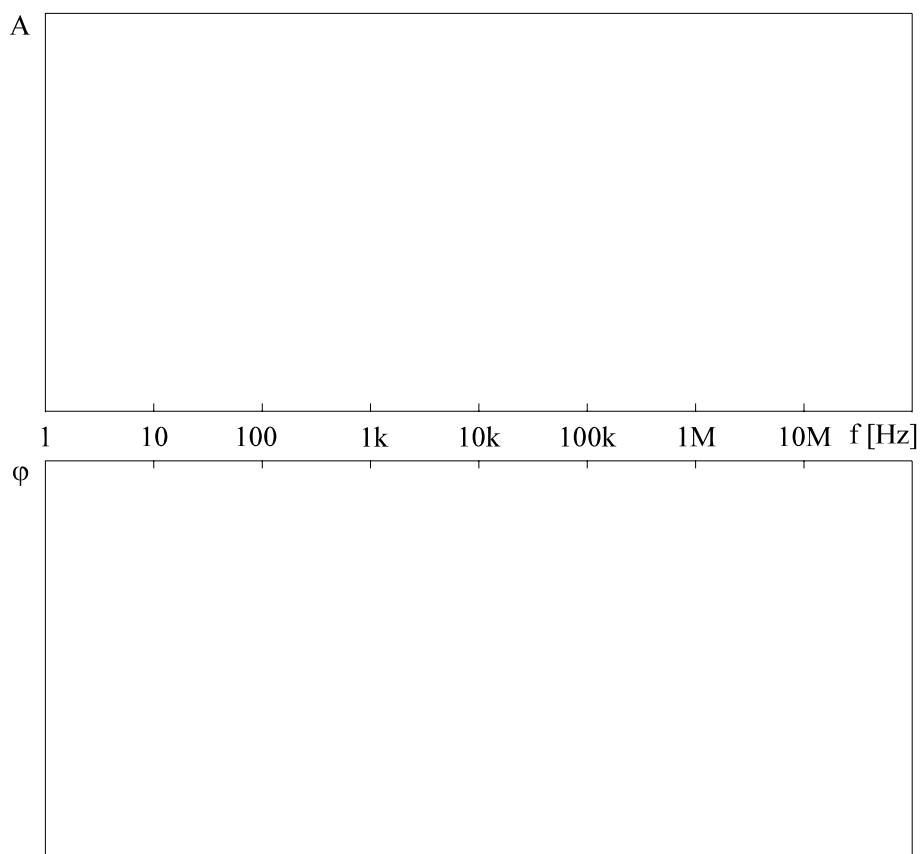
B2. Korišćenjem definisanog potkola opisati kolo filtra sa slike 3.8 za analizu programom **PSpice**. Vrednosti elemenata kola date su na slici.



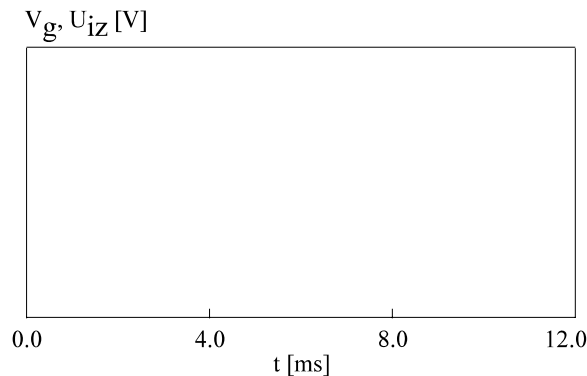
Slika 3.8: Filtar propusnik opsega frekvencija

B3. Analizirati filter u frekvencijskom domenu za opseg frekvencija od 1Hz do 100MHz u 10 tačaka po dekadi. Pomoću postprocesora **Probe** iscrtati na ekranu računara amplitudsku i faznu frekvencijsku karakteristiku i skicirati ih u priloženom koordinatnom sistemu. Odrediti vrednosti maksimalnog pojačanja i graničnih frekvencija filtra.

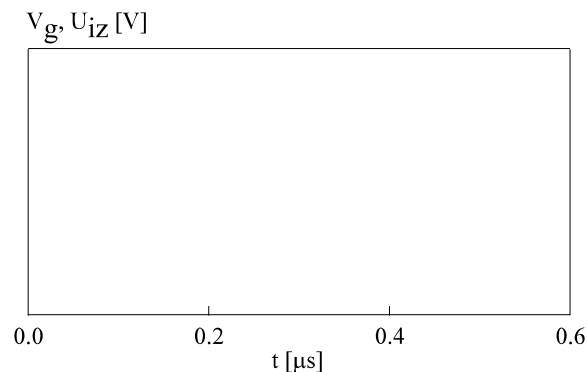
$$A = \underline{\hspace{2cm}} \quad f_d = \underline{\hspace{2cm}} \quad f_g = \underline{\hspace{2cm}}$$



B4. Pobuditi filter signalom testerastog talasnog oblika frekvencije 250Hz. Trougaoni signal formirati pomoću generatora trapeznog talasnog oblika sa sledećim parametrima: $v1=-0.1V$, $v2=0.1V$, $td=0.0s$, $tr=tf=2ms$, $pw=1ps$, $per=4ms$. Analizirati kolo u vremenskom domenu ako je korak štampanja rezultata $10\mu s$, a trajanje analize 12ms. Talasne oblike napona na generatoru V_g i izlaznog napona U_{iz} skicirati u priloženom koordinatnom sistemu. Kolo se ponaša kao

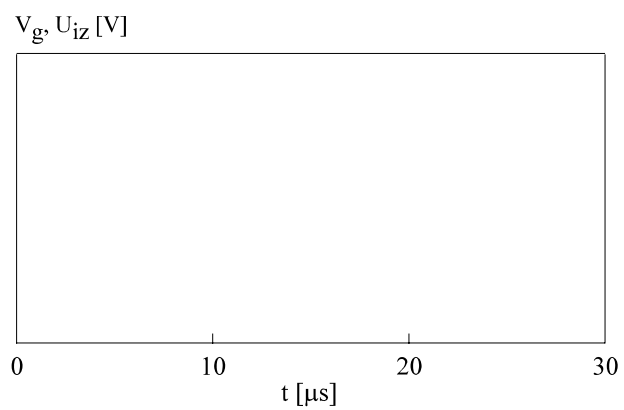


B5. Filtar pobuditi signalom testerastog talasnog oblika frekvencije 5kHz. Trougaoni signal formirati pomoću generatora trapeznog talasnog oblika sa sledećim parametrima: $v_1 = -0.1\text{V}$, $v_2 = 0.1\text{V}$, $t_d = 0.0\text{s}$, $t_r = t_f = 0.1\text{ms}$, $p_w = 1\text{ps}$, $p_{er} = 0.2\text{ms}$. Analizirati kolo u vremenskom domenu ako je korak štampanja rezultata $5\mu\text{s}$, a trajanje analize 0.6ms . Nametnuti kolu nulte početne napone na kondenzatorima korišćenjem specifikacije UIC u komandi .TRAN. Talasne oblike napona na generatoru V_g i izlaznog napona U_{iz} skicirati u priloženom koordinatnom sistemu. Kolo se ponaša kao _____.



B6. Filtar pobuditi signalom pravougaonog talasnog oblika frekvencije 100kHz. Pravougaoni signal formirati korišćenjem generatora trapezoidnog talasnog oblika sa sledećim parametrima: $v_1 = -0.1\text{V}$, $v_2 = 0.1\text{V}$, $t_d = 0.0\text{s}$, $t_r = t_f = 0.1\mu\text{s}$, $p_w = 5\mu\text{s}$, $p_{er} = 10\mu\text{s}$. Analizirati kolo u vremenskom domenu ako je korak štampanja rezultata $0.01\mu\text{s}$, a trajanje analize $30\mu\text{s}$. Korišćenjem komande .IC sistemu jednačina nametnuti granične uslove $V(2) = 4.5\text{mV}$ i $V(2,3) = 4.7\text{mV}$. Talasne oblike

napona na generatoru V_g i izlaznog napona U_{iz} skicirati u priloženom koordinatnom sistemu. Kolo se ponaša kao _____.



B7. Priložiti listing ulazne datoteke.

Datum : _____

Student : _____

Overava : _____

Vežba broj 4

Logička simulacija programom PSpice

4.1 Cilj vežbe

Upoznati se sa osnovnim principima logičke simulacije. Proučiti upotrebu programa *PSpice* za simulaciju digitalnih elektronskih kola. Posebno analizirati problem rezolucije na magistrali bez i sa pull-up/pull-down otpornikom kao i problem propagacije kratkotrajnih impulsa kroz digitalno kolo.

4.2 Teorijska postavka vežbe

4.2.1 Logička simulacija

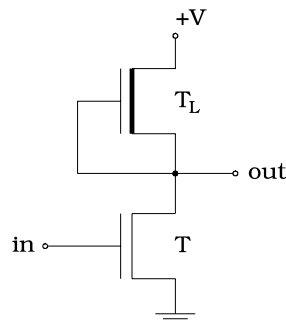
Logičkom simulacijom naziva se analiza digitalnih kola i sistema kod koje se (i) analogne vrednosti napona zamenjuju konačnim brojem diskretnih logičkih stanja (0, 1, Z, X,...) (ii) vremenska osa diskretizuje (iii) električni modeli komponenata zamenjuju logičkim [Lit00]. Za datu pobudu, logičkom simulacijom određuje se odziv sistema u obliku tabele stanja na izlazu ili izlazima. Prvu kolonu tabele čini diskretizovana vremenska osa. Tako se, poređenjem dobijenog i željenog odziva, može rasuđivati o ispravnosti projekta. Kao i kod električne analize, u osnovi simulacije stoji model koji je imitacija realnog stanja u kolu. Da

bi simulacija bila verna, model treba da bude dovoljno tačan, a da bi se ubrzao proces simulacije, model treba da bude što jednostavniji. Primenom *principa selekcije puteva* i *principa narednog događaja* [Lit00] postiže se smanjenje memorijskih zahteva i ubrzanje procesa simulacije. Zahvaljujući tome, moguće je simulirati digitalna elektronska kola velike složenosti.

Najvažnija karakteristika logičkog simulatora je skup dozvoljenih logičkih stanja. Osnovna logička stanja su logička nula '0' i logička jedinica '1'. Ova stanja odgovaraju stabilnim naponskim (ili strujnim) nivoima u kolu. Osim ovih stanja, za normalan rad logičkog simulatora neophodno je i neodređeno/nepoznato stanje 'X'. Ovo stanje koristi se za inicijalizaciju čvorova u kolu u početnom trenutku simulacije kada je zbog povratnih veza teško odrediti stanja čvorova. Takođe, stanje 'X' se koristi u slučaju konfliktnih situacija kod kojih nije moguće sa sigurnošću odrediti stanje na izlazu logičkog elementa.

Da bi se omogućila simulacija trostatičkih digitalnih kola uvodi se stanje visoke impedanse 'Z'. Treba, međutim, primetiti da stanje visoke impedanse ne treba smatrati stanjem u pravom smislu te reči. Kada je izlaz trostatičkog kola u stanju visoke impedanse, to ne znači da je na izlazu uspostavljen neki određeni naponski nivo, već samo označava da je izlazna otpornost kola postala vrlo velika. Pri tome, ako je kapacitivnost izlazne veze velika, na njoj se može još izvesno vreme zadržati prethodno stanje izlaza. Ako ova veza povezuje izlaze više trostatičkih kola, onda neko od njih, koje nije u stanju visoke impedanse, može da odredi stanje na vezi. Kratkospajanjem izlaza više trostatičkih logičkih kola formira se *magistrala* (uobičajen je i naziv *bus*), a kratkospajanje izlaza više standardnih logičkih kola naziva se *žičana logika*.

Kada se više od jednog izlaza vezanih na magistralu nalazi u stanju različitom od 'Z' na magistrali postoji konflikt. Konfliktom aktivnosti naziva se situacija kada je verovatno ili sigurno da oba izlaza vode magistralu u isto stanje, a konfliktom stanja kada je verovatno ili sigurno da izlazi vode magistralu u različita stanja. U slučaju konflikta, potrebno je definisati stanje na magistrali - razrešiti magistralu (takozvana rezolucija magistrale).



Slika 4.1: NMOS invertor. Tranzistor T ima manju otpornost od tranzistora T_L , pa je logička nula dominantna u odnosu na logičku jedinicu pri žičanoj logici.

Rezolucija magistrale i žičane logike modeluje se na različite načine kod različitih logičkih simulatora. Obično se pored skupa dozvoljenih logičkih stanja usvaja i skup jačina stanja na izlazu logičkog elementa. Jačinu stanja na izlazu logičkog elementa određuje izlazna otpornost elementa kada mu je izlaz u tom stanju. Recimo, kod logičkog kola realizovanog u NMOS tehnologiji (slika 4.1) logička nula ima veću jačinu od logičke jedinice, jer tranzistor T ima manju otpornost kanala kada vodi nego tranzistor T_L . Sistem stanja logičkog simulatora sastoji se od svih kombinacija stanja i jačina stanja. Tako, na primer, u sistemu stanja može da postoji "slaba" i "jaka" logička jedinica. Drugačiji pristup razrešenju problema rezolucije je da se uz informaciju o stanju na izlazu logičkog elementa memoriše i vrednost njegove izlazne otpornosti. Konflikt na magistrali razrešava se analizom izlaznih otpornosti svih drajvera magistrale. Ovaj pristup primenjen je kod simulatora *PSpice*.

Da bi se digitalno kolo simuliralo, njegova struktura mora se opisati pomoću ulaznog jezika simulatora. Pri tome se kolo razlaže do osnovnih logičkih elemenata (logičkih primitiva). Opis kola obično je hijerarhijski: od osnovnih elemenata formiraju se složenije celine - potkola (makroi), a od ovih još složenije celine na višem nivou hijerarhije, sve dok se ne opiše celo kolo.

U daljem tekstu biće objašnjeni mehanizmi logičke simulacije ugrađeni u hibridni simulator *PSpice* i opisane konstrukcije ulaznog jezika koje su u vezi sa logičkom simulacijom. Kompletan pregled sintakse ulaznog jezika simulatora *PSpice* dat je u prilogu 2. Specifične oznake korišćene u opisima sintakse objašnjene su na početku istog priloga.

4.2.2 Logička simulacija simulatorom *PSpice*

Iako je u osnovi *PSpice* namenjen za analizu analognih kola on podržava i simulaciju digitalnih kola. Digitalna simulacija "dograđena" je na osnovni, analogni, simulator pomoću komponente **U**. Ova komponenta objedinjuje sve logičke primitive (osnovne logičke elemente) i logičke pobudne generatore.

4.2.3 Sistem stanja i jačine stanja

Simulator *PSpice* za logičku simulaciju koristi 5 logičkih stanja koja su opisana u tabeli 4.1.

Osim logičkog stanja, izlaz logičkog elementa karakteriše se i izlaznom otpornošću. Na osnovu opsega izlaznih otpornosti logičkih elemenata u kolu, *PSpice* formira 64 nivoa jačine stanja. Raspon vrednosti izlaznih otpornosti u simuliranom kolu definiše se pomoću kontrolnih parametara **DIGDRVZ** i **DIGDRVF** u **.OPTIONS** naredbi. **DIGDRVZ** predstavlja maksimalnu izlaznu otpornost logičkog elementa u simuliranom kolu (najslabiji drajver), a **DIGDRVF** predstavlja minimalnu izlaznu otpornost u kolu (najjači drajver). Ako korisnik ne zada druge vrednosti, *PSpice* podrazumeva vrednosti $20\text{k}\Omega$ i 2Ω , respektivno. Između ove dve vrednosti simulator formira logaritamsku skalu sa 64 intervala otpornosti. Jačina stanja na izlazu konkretnog logičkog elementa definisana je celim brojem između 0 i 63 koji definiše u kome se od ovih opsega nalazi vrednost izlazne otpornosti tog logičkog elementa. Drajver sa izlaznom otpornošću **DIGDRVZ** ima jačinu 0 (s obzirom da je ovo jačina stanja visoke impedanse, zvaćemo je i jačina 'Z'), a drajver sa izlaznom otpornošću **DIGDRVF** ima jačinu 63. Vrednost izlazne otpornosti pridružuje se svakom logičkom elementu pomoću parametara **DRVH** (za stanje logičke jedinice) i **DRVL** (za stanje logičke nule) u U/I modelu. Ako su vrednosti **DRVH** i **DRVL** različite, jačine stanja logičke jedinice i logičke nule se razlikuju.

Tabela 4.1: Logička stanja kod simulatora *PSpice*

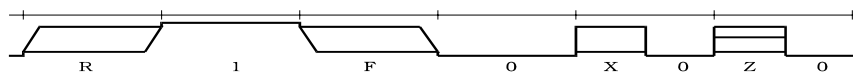
Oznaka stanja	Značenje
0	logička nula
1	logička jedinica
X	nepoznato/neodređeno stanje
R	stanje prelaska sa '0' na '1'
F	stanje prelaska sa '1' na '0'

Pri vezivanju izlaza više logičkih elemenata u isti čvor (magistralu), stanje čvora određuje drajver sa minimalnom izlaznom otpornošću, odnosno sa

maksimalnom jačinom. Može se, međutim, desiti da više drajvera imaju približno iste jačine, tako da se nijedan ne može smatrati dominantnim. Kontrolni parametar DIGOVRDRV u komandi .OPTIONS definiše koliko nivoa (na skali od 0 do 63) mora drajver biti jači od svih ostalih drajvera da bi bio dominantan u odnosu na njih. Podrazumevana vrednost ovog parametra je 3.

Osim pet logičkih stanja navedenih u tabeli 4.1, specijalno, pri opisu talasnih oblika na izlazima logičkih pobudnih generatora, može se koristiti i stanje 'Z'. U tom slučaju, ovo stanje tumači se kao nepoznato stanje 'X' sa jačinom 'Z'.

Slika 4.2 ilustruje predstavljanje logičkih stanja u grafičkom postprocesoru Probe pri njegovom korišćenju za prikaz rezultata logičke simulacije. Sva stanja koja imaju jačinu 'Z' prikazuju se trostrukom linijom, dok se ostale jačine ne mogu grafički predstaviti.



Slika 4.2: Grafička reprezentacija stanja u postprocesoru Probe

4.2.4 Osnovni logički elementi

Format za opis logičkih elemenata sličan je formatu za opis analognih elemenata. Najbitnija razlika je u tome što se kod opisa logičkog elementa koriste dva modela (modelske kartice). To su *vremenski model* i *ulazno/izlazni model*. Razlog za postojanje dva modela je u tome što su vremenske karakteristike (vremena kašnjenja, vremena postavljanja i držanja, i druge) različite za pojedine tipove logičkih elemenata, dok su U/I karakteristike (ulazne otpornosti i kapacitivnosti, pragovi promene stanja, i druge) zajedničke za celu familiju logičkih elemenata proizvedenih istim tehnološkim postupkom. Dakle, mnogi elementi iz iste familije mogu imati isti U/I model, a različite vremenske modele.

PSpice prepoznaje sledeće grupe logičkih elemenata:

- gejtovi (standardni i trostatički)
- flip-flopovi
- višebitni A/D i D/A konvertori
- linija za kašnjenje
- pull-up i pull-down otpornik
- PLD strukture
- memorije tipa ROM i RAM
- elementi vremenske provere

4.2.5 Format opisa logičkog elementa

Opšti format opisa logičkog elementa je sledeći:

```
U<ime> <tip elementa> [( <vrednost parametra>* )]
+   <čvor digitalnog napajanja> <čvor digitalne mase>
+   <čvor>* <ime vremenskog modela> <ime U/I modela>
+   [MNTYMXDLY = <vrednost selektora kašnjenja>]
+   [IO_LEVEL = <vrednost selektora interfejsa>]
```

Parametrom <tip elementa> korisnik određuje koji tip logičkog elementa je u pitanju (na primer: INV, AND, NAND, DFF, ROM, PLA i drugi). Zavisno od tipa elementa, može biti potrebno definisati dodatne parametre - broj ulaza u element ili broj elemenata u nizu.

Zatim se navode čvorovi digitalnog napajanja. Za ove čvorove vezuju se interfejsna potkola za konverziju signala (A/D i D/A konvertori koje simulator automatski ubacuje na mestima sprežanja analognih i logičkih elemenata pri hibridnoj simulaciji). Kod simulacije čisto digitalnih kola ovi čvorovi nemaju nikakvu ulogu, **ali se moraju navesti**. Uobičajeno je da se koriste globalni čvorovi digitalnog napajanja \$G_DPWR i \$G_DGND.

Nakon toga, navode se ulazni, a zatim izlazni čvorovi logičkog elementa.

Sledi ime vremenskog modela i ime U/I modela. Parametri vremenskog modela zadaju se minimalnom, tipičnom i maksimalnom vrednošću, a pri simulaciji se koristi samo jedna od njih. Za izbor vrednosti kašnjenja kod pojedinačnih logičkih elemenata služi parametar MNTYMXDLY u opisu elementa, a izbor kašnjenja za sve logičke elemente odjednom vrši se kontrolnim parametrom DIGMNTYMX koji se definiše u okviru .OPTIONS naredbe. Vrednost parametra DIGMNTYMX može biti 1, 2 ili 3. Podrazumevana vrednost je 2. Vrednost parametra MNTYMXDLY može biti 0, 1, 2 ili 3. Podrazumevana vrednost je 0. Tumačenja ovih vrednosti data su u tabeli 4.2.

Tabela 4.2: Tumačenje vrednosti parametara DIGMNTYMX i MNTYMXDLY

Vrednost	Značenje
0	/samo za parametar MNTYMXDLY/ koristi se kašnjenje zadato parametrom DIGMNTYMX u .OPTIONS naredbi
1	koristi se minimalno kašnjenje
2	koristi se tipično kašnjenje
3	koristi se maksimalno kašnjenje

Parametrom `IO_LEVEL` bira se jedan od četiri para interfejsnih potkola iz U/I modela elementa. U komandi `.OPTIONS` postoji kontrolni parametar `DIGIOLVL` koji ima istu ulogu - izbor interfejsnih potkola za hibridnu simulaciju - ali za sve logičke elemente u kolu odjednom. Parametar `DIGIOLVL` može imati vrednost 1, 2, 3 ili 4, a parametar `IO_LEVEL` može imati vrednost 0, 1, 2, 3 ili 4. Tumačenje ovih vrednosti dato je u tabeli 4.3.

Tabela 4.3: Tumačenje vrednosti kontrolnih parametara `DIGIOLVL` i `IO_LEVEL`

Vrednost	Značenje
0	/samo za parametar <code>IO_LEVEL</code> / koristi se interfejsno potkolo zadato parametrom <code>DIGIOLVL</code> u <code>.OPTIONS</code> naredbi
1	koriste se potkola AtoD1 i DtoA1
2	koriste se potkola AtoD2 i DtoA2
3	koriste se potkola AtoD3 i DtoA3
4	koriste se potkola AtoD4 i DtoA4

Primeri:

```

U1 NAND(2) $G_DPWR $G_DGND 1 2 10 D_GATE IO_DFT
U2 JKFF(1) $G_DPWR $G_DGND 3 5 200 13 13 10 11
+      D_293ASTD IO_STD
U3 INV $G_DPWR $G_DGND IN OUT D_INV IO_INV MNTYMXDLY=3
+      IO_LEVEL=2

```

U prvom primeru opisano je dvoulazno NI kolo čiji su ulazi vezani za čvorove 1 i 2, a izlaz za čvor 10. Vremenski model NI kola ima ime `D_GATE`, a U/I model ima ime `IO_DFT`. U drugom primeru opisan je jedan JK flip-flop čiji su ulazi reset, set, takt, J, K, Q i NQ vezani za čvorove sa oznakama 3, 5, 200, 13, 13, 10 i 11, respektivno. Vremenski model je `D293ASTD`, a U/I model je `IO_STD`. U trećem primeru opisan je inverter vezan između čvorova `IN` i `OUT` kod koga se koriste maksimalne vrednosti kašnjenja definisane u vremenskoj modelskoj kartici `D_INV` i interfejsna potkola `AtoD2` i `DtoA2` iz U/I modelske kartice `IO_INV`.

4.2.6 U/I model logičkog elementa

U/I model logičkog elementa sadrži:

- vrednosti ulazne i izlazne kapacitivnosti logičkog elementa
- vrednosti izlaznih otpornosti elementa u stanjima logičke jedinice i logičke nule
- imena četiri A/D i četiri D/A interfejsnih potkola koja **PSpice** automatski poziva kada prepozna hibridni čvor na ulazu ili na izlazu logičkog elementa

U/I model definiše se u sledećem formatu:

```
.MODEL <ime modelske kartice> UIO [parametri modela]
```

Parametri modela objašnjeni su u tabeli 4.4.

Tabela 4.4: Parametri U/I modela logičkog elementa

Ime parametra	Značenje	Jedinica	Podraz. vrednost
INLD	ulazna kapacitivnost	F	0
OUTLD	izlazna kapacitivnost	F	0
DRVH	izlazna otpornost u stanju '1'	Ω	50
DRVL	izlazna otpornost u stanju '0'	Ω	50
AtoD1	ime A/D interf. potkola (nivo 1)		AtoDDefault
DtoA1	ime D/A interf. potkola (nivo 1)		DtoADefault
AtoD2	ime A/D interf. potkola (nivo 2)		AtoDDefault
DtoA2	ime D/A interf. potkola (nivo 2)		DtoADefault
AtoD3	ime A/D interf. potkola (nivo 3)		AtoDDefault
DtoA3	ime D/A interf. potkola (nivo 3)		DtoADefault
AtoD4	ime A/D interf. potkola (nivo 4)		AtoDDefault
DtoA4	ime D/A interf. potkola (nivo 4)		DtoADefault

Primeri:

```
.MODEL io_std UIO ; difoltni model
.MODEL io_nand2 UIO (DRVL=2kohm, DRVH=100ohm,
+ INLD=3.5pF, OUTLD=1pF)
```

4.2.7 Model kašnjenja logičkog elementa

Vremenske karakteristike logičkog elementa određene su opisima u vremenskom modelu i U/I modelu. Vremenski model specificira propagaciona kašnjenja (recimo minimalno, tipično i maksimalno kašnjenje prednje i zadnje ivice) i vremenska ograničenja (recimo vremena postavljanja i držanja kod flip-flopova), a U/I model specificira ulaznu i izlaznu kapacitivnost i izlazne otpornosti u stanju logičke nule i u stanju logičke jedinice. Ukupno kašnjenje kroz logički element izračunava se sabiranjem kašnjenja koje je specificirano u vremenskom modelu i kašnjenja koje je posledica postojanja kapacitivnosti i otpornosti na izlazu elementa. Ovo dodatno kašnjenje izračunava se pre početka simulacije na osnovu ukupne kapacitivnosti na izlazu i izlazne otpornosti, i to za svaki izlazni čvor svakog logičkog elementa. Ukupna kapacitivnost dobija se sabiranjem

kapacitivnosti (parametri OUTLD i INLD iz U/I modela) svih elemenata koji su vezani za taj izlaz. Iz izračunate ukupne kapacitivnosti i izlazne otpornosti digitalnog elementa (parametri DRVL i DRVH iz U/I modela) dodatno kašnjenje određuje se kao:

$$\text{DODATNO_KAŠNJENJE} = R_{\text{elementa}} C_{\text{ukupno}} \ln 2$$

4.2.8 Gejtovi

Grupa logičkih elemenata poznata pod nazivom gejtovi deli se na dve podgrupe: standardni gejtovi i trostatički gejtovi. Kod trostatičkih gejtova stanje izlaza kontroliše se ulazom *enable*. Kada je ulaz *enable* u stanju logičke nule, na izlazu gejta je stanje visoke impedanse definisano kao 'X' stanje sa jačinom 'Z'.

Gejtovski logički elementi pojavljuju se u formi običnog gejta i u formi gejtovskog niza. Gejtovski niz sadrži jedan ili više običnih gejtova u jednom digitalnom elementu. Na taj način olakšan je rad sa poluprovodničkim komponentama koje sadrže više gejtova u jednom pakovanju.

4.2.9 Standardni gejtovi

U tabeli 4.5 navedeni su svi standardni gejtovi koje prepoznaje PSpice.

Kod pojedinačnih gejtova prvo se navode ulazni čvorovi, pa zatim izlazni čvor. Kod gejtovskog niza prvo se navode svi ulazni čvorovi (za prvi gejt, za drugi gejt, ...), a zatim svi izlazni čvorovi (u istom redosledu). Ukupan broj ulaznih čvorova jednak je proizvodu broja ulaza i broja gejtova, a broj izlaznih čvorova jednak je broju gejtova.

Poslednja četiri elementa u tabeli 4.5 predstavljaju dvonivoska kola sa proizvoljnim brojem gejtova u prvom nivou i samo jednim gejtom u drugom nivou. Tako, na primer, element AOI ima <#gejtova> I logičkih kola sa <#ulaza> ulaza u prvom nivou. Izlazi svih ovih I kola predstavljaju ulaze jednog NILI kola u drugom nivou, čiji izlaz je istovremeno izlaz elementa AOI. Tako element AOI ima <#ulaza>·<#gejtova> ulaza, a samo jedan izlaz. Redosled navođenja čvorova je sledeći: svi ulazi prvog gejta iz prvog nivoa, svi ulazi drugog gejta iz prvog nivoa, ..., svi ulazi poslednjeg gejta iz prvog nivoa, izlaz gejta iz drugog nivoa.

Tabela 4.5: Standardni gejtovi (* označava da može biti više od jednog čvora)

Tip gejta	Parametri	Čvorovi	Opis
BUF		in, out	bafer
INV		in, out	invertor
AND	(<#ulaza>)	in*, out	I kolo

NAND	(<#ulaza>)	in*, out	NI kolo
OR	(<#ulaza>)	in*, out	ILI kolo
NOR	(<#ulaza>)	in*, out	NILI kolo
XOR		in1, in2, out	ekskl. ILI kolo
NXOR		in1, in2, out	ekskl. NILI kolo
BUFA	(<#ulaza>)	in*, out*	niz bafera
INVA	(<#ulaza>)	in*, out*	niz invertora
AND	(<#ulaza>, <#gejtova>)	in*, out*	niz I kola
NAND	(<#ulaza>, <#gejtova>)	in*, out*	niz NI kola
ORA	(<#ulaza>, <#gejtova>)	in*, out*	niz ILI kola
NORA	(<#ulaza>, <#gejtova>)	in*, out*	niz NILI kola
XORA	(<#gejtova>)	in*, out*	niz ekskl. ILI kola
NXORA	(<#gejtova>)	in*, out*	niz ekskl. NILI kola
AO	(<#ulaza>, <#gejtova>)	in*, out	I-ILI kombinacija
OA	(<#ulaza>, <#gejtova>)	in*, out	ILI-I kombinacija
AOI	(<#ulaza>, <#gejtova>)	in*, out	I-NILI kombinacija
OAI	(<#ulaza>, <#gejtova>)	in*, out	ILI-NI kombinacija

Vremenski model standardnog gejta definiše se na sledeći način:

```
.MODEL <ime modelske kartice> UGATE [ parametri modela ]
```

Parametri vremenskog modela standardnog gejta objašnjeni su u tabeli 4.6.

Tabela 4.6: Parametri vremenskog modela standardnog gejta

Ime parametra	Opis	Jedinica	Podraz. vrednost
TPLHMN	minimalno kašnjenje '0'-'1'	s	0
TPLHTY	tipično kašnjenje '0'-'1'	s	0
TPLHMX	maksimalno kašnjenje '0'-'1'	s	0
TPHLMN	minimalno kašnjenje '1'-'0'	s	0
TPHLTY	tipično kašnjenje '1'-'0'	s	0
TPHLMX	maksimalno kašnjenje '1'-'0'	s	0

Primeri:

* dvoulazno I kolo:

```
U5 AND(2) $G_DPWR, $G_DGND, IN1, IN2, OUT, TM_AND2, IO_STD
.MODEL IO_STD UIO ; difoltni U/I model
.MODEL TM_AND2 (TPLHTY=2ns, TPHLTY=1.5ns) ; vremenski model
```

* četiri dvoulazna NI kola (A, B, C i D):

```
U13 NANDA(2, 4) $G_DPWR, $G_DGND,
```

```

+   INA1, INA2, INB1, INB2, INC1, INC2, IND1, IND2,
+   OUTA, OUTB, OUTC, OUTD,
+   TM NANDA, IO STD
.MODEL TM_NANDA (TPLHMN=1.5ns, TPLHTY=2.2ns, TPLHMX=3.4ns,
+   TPHLMN=.9ns, TPHLTY=1.2ns, TPHLMX=3ns)

```

4.2.10 Trostatički gejtovi

U tabeli 4.7 navedeni su svi trostatički gejtovi koje prepoznaje PSpice.

Pri opisivanju trostatičkih logičkih elemenata, kod pojedinačnih gejtova redosled navođenja čvorova je: ulazni čvorovi, čvor *enable*, izlazni čvor. Kod gejtovskog niza prvo se navode svi ulazni čvorovi (za prvi gejt, za drugi gejt, ...), zatim zajednički čvor *enable* i na kraju svi izlazni čvorovi. Ukupan broj ulaznih čvorova za jedan je veći od proizvoda broja ulaza i broja gejtova, a broj izlaznih čvorova jednak je broju gejtova.

Vremenski model trostatičkog gejta ima sledeći oblik:

```
.MODEL <ime modelske kartice> UTGATE [ parametri modela ]
```

Parametri modela objašnjeni su u tabeli 4.8.

Primer:

```

* trostatički inverter:
Uinv   INV3   $g_dpwr, $g_dgnd, input, enable, output
+       t_inv3 io_std
.MODEL  t_inv3 (tplhty=2ns, tphlty=1.5ns, tplzty=3ns,
+               tpzltty=1.4ns, tphzty=4ns, tpzhty=2ns)

```

Tabela 4.7: Trostatički gejtovi (* označava da može biti više od jednog čvora)

Tip gejta	Parametri	Redosled čvorova	Opis
BUF3		in, en, out	bafer
INV3		in, en, out	inverter
AND3	(<#ulaza>)	in*, en, out	I kolo
NAND3	(<#ulaza>)	in*, en, out	NI kolo
OR3	(<#ulaza>)	in*, en, out	ILI kolo
NOR3	(<#ulaza>)	in*, en, out	NILI kolo
XOR3		in1, in2, en, out	ekskluzivno ILI kolo
NXOR3		in1, in2, en, out	ekskluzivno NILI kolo

BUF3A	(<#gejtova>)	in*, en, out*	niz bafera
INV3A	(<#gejtova>)	in*, en, out*	niz invertora
AND3A	(<#ulaza>,<#gejtova>)	in*, en, out*	niz I kola
NAND3A	(<#ulaza>,<#gejtova>)	in*, en, out*	niz NI kola
OR3A	(<#ulaza>,<#gejtova>)	in*, en, out*	niz ILI kola
NOR3A	(<#ulaza>,<#gejtova>)	in*, en, out*	niz NILI kola
XOR3A	(<#gejtova>)	in*, en, out*	niz ekskl. ILI kola
NXOR3A	(<#gejtova>)	in*, en, out*	niz eksk. NILI kola

4.2.11 Pull-up i pull-down otpornici

Pull-up (pull-down) otpornik dovodi magistralu u stanje '1' ('0') kada se svi dražveri magistrale nalaze u stanju visoke impedanse, odnosno kada magistrala ostane bez aktivnog dražvera. Elementi PULLUP i PULLDN predstavljaju niz pull-up, odnosno niz pull-down otpornika. Ovi elementi nemaju ulaze, već samo jedan ili više izlaza (onoliko koliko je otpornika u nizu). Nemaju ni vremenski model, a jačina logičke jedinice na izlazima elementa PULLUP, odnosno logičke nule na izlazima elementa PULLDN, definiše se u okviru U/I modela. Format opisa ovih elemenata je sledeći:

```
U<ime>      <tip otpornika> ( <broj otpornika u nizu> )
+   <čvor digitalnog napajanja> <čvor digitalne mase>
+   <izlazni čvor>* <ime U/I modela>
+   [IO_LEVEL = <vrednost selektora interfejsa>]
```

Primeri:

```
Upullup1 PULLUP(4) $G_DPWR $G_DGND out1 out2 out3 out4 R1k
Upulldn1 PULLDN(1) $G_DPWR $G_DGND out R500
.MODEL R1k UIO (DRVH=1kohm)
.MODEL R500 UIO (DRVL=500)
```

Tabela 4.8: Parametri vremenskog modela trostatičkog gejt

Ime parametra	Opis	Jedinica	Podraz. vrednost
TPLHMN	minimalno kašnjenje '0'-'1'	s	0
TPLHTY	tipično kašnjenje '0'-'1'	s	0
TPLHMX	maksimalno kašnjenje '0'-'1'	s	0
TPHLMN	minimalno kašnjenje '1'-'0'	s	0
TPHLTY	tipično kašnjenje '1'-'0'	s	0
TPHLMX	maksimalno kašnjenje '1'-'0'	s	0

TPLZMN	minimalno kašnjenje '0'-'>'Z'	s	0
TPLZTY	tipično kašnjenje '0'-'>'Z'	s	0
TPLZMX	maksimalno kašnjenje '0'-'>'Z'	s	0
TPHZMN	minimalno kašnjenje '1'-'>'Z'	s	0
TPHZTY	tipično kašnjenje '1'-'>'Z'	s	0
TPHZMX	maksimalno kašnjenje '1'-'>'Z'	s	0
TPZLMN	minimalno kašnjenje 'Z'-'>'0'	s	0
TPZLTY	tipično kašnjenje 'Z'-'>'0'	s	0
TPZLMX	maksimalno kašnjenje 'Z'-'>'0'	s	0
TPZHMN	minimalno kašnjenje 'Z'-'>'1'	s	0
TPZH TY	tipično kašnjenje 'Z'-'>'1'	s	0
TPZHM X	maksimalno kašnjenje 'Z'-'>'1'	s	0

4.2.12 Linija za kašnjenje

Linija za kašnjenje, komponenta DLYLINE, prenosi stanje sa ulaza na izlaz posle propagacionog kašnjenja koje je specificirano u vremenskom modelu. Linija za kašnjenje modeluje ponašanje **veze** u digitalnom kolu. Impuls bilo koje širine može da se prenese kroz liniju za kašnjenje, dok gejtovi ne propuštaju impulse kraće od propagacionog kašnjenja.

Format opisa linije za kašnjenje je sledeći:

```
U<ime> DLYLINE <čvor digitalnog napajanja>
+ <čvor digitalne mase> <ulazni čvor> <izlazni čvor>
+ <ime vremenskog modela> <ime U/I modela>
+ [MNTYMXDLY = <vrednost selektora kašnjenja>]
+ [IO_LEVEL = <vrednost selektora interfejsa>]
```

Definicija vremenskog modela ima sledeću sintaksu:

```
.MODEL <ime modelske kartice> UDLY [ parametri modela ]
```

Parametri vremenskog modela objašnjeni su u tabeli 4.9.

Tabela 4.9: Parametri vremenskog modela elementa DLYLINE

Ime parametra	Opis	Jedinica	Podraz. vrednost
DLYMN	minimalno kašnjenje	s	0
DLYTY	tipično kašnjenje	s	0
DLYMX	maksimalno kašnjenje	s	0

Primer:

```
U5 DLYLINE $G_DPWR $G_DGND line_in line_out dly20ns io_std
.MODEL dly20ns UDLY (DLYTY=20ns)
```

4.2.13 Elementi vremenske provere

U ovu grupu logičkih elemenata spadaju dve specifične komponente koje služe za proveru ispravnosti rada simuliranog kola u odnosu na zadata vremenska ograničenja. Element WDTCHK proverava trajanje impulsa (detektuje visokofrekventne impulse), a element SUHDCK proverava vremena postavljanja i držanja signala u odnosu na zadati taktni signal.

Element WDTCHK predstavlja niz elemenata vremenske provere od kojih svaki ima jedan ulazni čvor (na kome se testiraju trajanja impulsa) i dva izlazna čvora (koji se postavljaju u stanje logičke jedinice kada se detektuje impuls kraći od zadatog vremena). Prvi izlazni čvor signalizira detekciju pika, a drugi detekciju dipa. Format opisa elementa WDTCHK je sledeći:

```
U<ime> WDTCHK ( <broj elemenata provere> )
+ <čvor digitalnog napajanja> <čvor digitalne mase>
+ <ulazni čvor>*
+ <čvor za detekciju pika>* <čvor za detekciju dipa>*
+ <ime vremenskog modela> <ime U/I modela>
+ [MNTYMXDLY = <vrednost selektora kašnjenja>]
+ [IO_LEVEL = <vrednost selektora interfejsa>]
```

Čvor u kome se signalizira prekratak impuls logičke jedinice postavlja se u stanje '1' pri pojavi opadajuće ivice na ulaznom čvoru, a resetuje se ponovo na nulu pri sledećoj rastućoj ivici. Čvor koji signalizira prekratak impuls logičke nule postavlja se u stanje '1' pri pojavi rastuće ivice na ulaznom čvoru, a resetuje se na nulu pri sledećoj opadajućoj ivici.

U vremenskom modelu elementa WDTCHK definišu se minimalne dozvoljene širine impulsa logičke jedinice i impulsa logičke nule njihovim minimalnim, tipičnim i maksimalnim vrednostima. Sintaksa za definisanje modela je sledeća:

```
.MODEL <ime modelske kartice> UWDTH [ parametri modela ]
```

Parametri vremenskog modela nabrojani su u tabeli 4.10.

Tabela 4.10: Parametri vremenskog modela elementa WDTCHK

Ime parametra	Opis	Jedinica	Podraz. vrednost
TWHMN	min vrednost minimalne širine pika	s	0

TWHTY	tipična vrednost minimalne širine pika	s	0
TWHMX	max vrednost minimalne širine pika	s	0
TWLMN	min vrednost minimalne širine dipa	s	0
TWLTY	tipična vrednost minimalne širine dipa	s	0
TWLMX	max vrednost minimalne širine dipa	s	0

Primer:

```
U54 WDTCK(1) $G_DPWR, $G_DGND, in, higherr, lowerr,
+          tm_chk, io_std
.MODEL tm_chk UWIDTH (TWHTY=10ns, TWLTY=5ns)
```

Element SUHDCK predstavlja niz elemenata vremenske provere kojima se testiraju vremena postavljanja i držanja signala u odnosu na jedan zajednički *clock* signal. Svaki element provere ima jedan ulazni čvor i dva izlazna čvora. Format opisa elementa SUHDCK je sledeći:

```
U<ime> SUHDCK ( <broj elemenata provere> )
+ <čvor digitalnog napajanja> <čvor digitalne mase>
+ <clock čvor> <ulazni čvor>*
+ <čvor za detekciju prekršaja setup-a>*
+ <čvor za detekciju prekršaja hold-a>*
+ <ime vremenskog modela> <ime U/I modela>
+ [MNTYMXDLY = <vrednost selektora kašnjenja>]
+ [IO_LEVEL = <vrednost selektora interfejsa>]
```

Vreme postavljanja i vreme držanja mere se u odnosu na uzlaznu ivicu signala takta (*clock*). Prekršaj vremena postavljanja detektuje se pri pojavi uzlazne ivice signala takta. Ako je došlo do prekršaja, u tom trenutku se čvor <čvor za detekciju prekršaja setup-a> postavlja u stanje '1' i ostaje u tom stanju sve do sledeće uzlazne ivice *clock* signala, a zatim se vraća na nulu (osim ako je ponovo došlo do narušavanja vremena postavljanja). Prekršaj vremena držanja detektuje se pri promeni stanja signala podataka (ulazni čvor). Ako je došlo do prekršaja vremena držanja, u tom trenutku se čvor <čvor za detekciju prekršaja hold-a> postavlja u stanje '1' i ostaje u tom stanju sve do prve naredne uzlazne ivice signala takta.

Vremenski model elementa SUHDCK definiše se na sledeći način:

```
.MODEL <ime modelske kartice> USUHD [ parametri modela ]
```

Parametri modela navedeni su u tabeli 4.11.

Tabela 4.11: Parametri vremenskog modela elementa SUHDCK

Ime parametra	Opis	Jedinica	Podr. vrednost
TSUMN	min vrednost vremena postavljanja	s	0

TSUTY	tipična vrednost vremena postavljanja	s	0
TSUMX	max vrednost vremena postavljanja	s	0
THDMN	min vrednost vremena držanja	s	0
THDTY	tipična vrednost vremena držanja	s	0
THDMX	max vrednost vremena držanja	s	0

Primer:

```

Uchk1 SUHD(3) $G_DPWR $G_DGND CLK; tri setup/hold čekera
+   IN0, IN1, IN2,
+   SUERR0, SUERR1, SUERR2,
+   HDERR0, HDERR1, HDERR2,
+   FFCHK IO STD
.MODEL FFCHK USUHD (TSUTY=5NS, THDTY=4NS)

```

4.2.14 Logički pobudni generator

Uloga logičkog pobudnog generatora je da obezbedi odgovarajući talasni oblik digitalnih signala na digitalnim ulazima kola koje se simulira. Logički pobudni generator nema vremenski model. Opšti format opisa logičkog pobudnog generatora je sledeći:

```

U<ime>      STIM ( <broj izlaza>, <format> )
+          <čvor digitalnog napajanja> <čvor digitalne mase>
+          <čvor>* <ime U/I modela>
+          [IO_LEVEL = <vrednost selektora interfejsa>]
+          [TIMESTEP = <vrednost koraka>] <komanda>*

```

Pobudni generatori mogu imati jedan ili više izlaza. Parametrom <format> specificira se način zadavanja promene stanja u izlaznim čvorovima. Dozvoljeno je da se binarna reč kojom se opisuju stanja na izlazima predstavi u binarnom (oznaka 1), oktalnom (oznaka 3) i heksadecimalnom (oznaka 4) zapisu.

U komandama kojima se definišu talasni oblici na izlazima (<komanda>), vreme se može izraziti u sekundama (sufiks s) ili u periodama taktnog signala (sufiks c). Ako se vreme izražava u periodama takta, potrebno je zadati vrednost periode. Parametar TIMESTEP predstavlja trajanje periode signala takta.

Komande za definisanje talasnog oblika na izlazima mogu imati sledeće oblike:

```

<vreme> <vrednost>
LABEL = <ime labela>
<vreme> GOTO <ime labela> <n> TIMES
<vreme> INCR BY <vrednost>
<vreme> DECR BY <vrednost>
<vreme> GOTO <ime labela> UNTIL GT <vrednost>

```



```

<vreme> GOTO <ime labela> UNTIL GE <vrednost>
<vreme> GOTO <ime labela> UNTIL LT <vrednost>
<vreme> GOTO <ime labela> UNTIL LE <vrednost>

```

Parametar <vreme> predstavlja trenutak aktiviranja komande i **mora imati ili sufix s ili sufix c**. Vreme se može zadati apsolutno (na primer 45NS, 5C) ili relativno, u odnosu na vreme korišćeno u prethodnoj komandi. Za specificiranje relativnog vremena koristi se prefiks + (na primer +5NS, +10C).

Parametar <vrednost> predstavlja stanje na izlazu i može biti: 0, 1, X, R, F ili Z (neodređeno stanje sa jačinom 'Z').

Promene koje su naznačene u apsolutnom vremenu unutar GOTO petlje biće pretvorene u relativno vreme na osnovu prethodnog vremena pri prvom ulasku u petlju i brojača petlje. Naredbi GOTO mora prethoditi definicija odgovarajuće labela. Vremena promene stanja na izlazima moraju biti u striktno rastućem redosledu, osim jednog izuzetka: prva naredba posle GOTO skoka **uvek se izvršava bez kašnjenja**. Ovo pravilo sprečava dupliranje kašnjenja, jer je kašnjenje sadržano i u samoj GOTO naredbi.

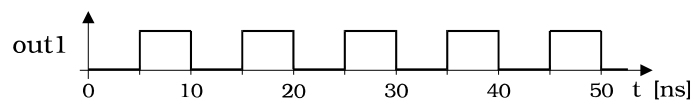
Primer 1:

```

U1 STIM(1, 1) $G_DPWR G_DGND OUT1 IO_STIM
+ 0S 0 ; u t=0s čvor OUT1 inicijalizuje se stanjem '0'
+ LABEL = STARTLOOP
+ +5NS 1 ; 5ns kasnije OUT1 se postavlja u stanje '1'
+ +5NS 0 ; 5ns kasnije OUT1 se postavlja u stanje '0'
+ +5NS GOTO STARTLOOP -1 TIMES
      ; 5ns kasnije prelazi se na labelu STARTLOOP.
      ; Prva naredba u petlji (naredba iza LABEL
      ; deklaracije) izvršava se uvek bez kašnjenja,
      ; kašnjenje 5ns obezbeđeno je GOTO naredbom).
      ; Specifikacija -1 TIMES prouzrokuje da se
      ; petlja izvršava beskonačan broj puta.

```

Talasni oblik generisan na izlazu OUT1 do trenutka 50ns prikazan je na slici 4.3.



Slika 4.3: Talasni oblik na izlazu pobudnog generatora U1

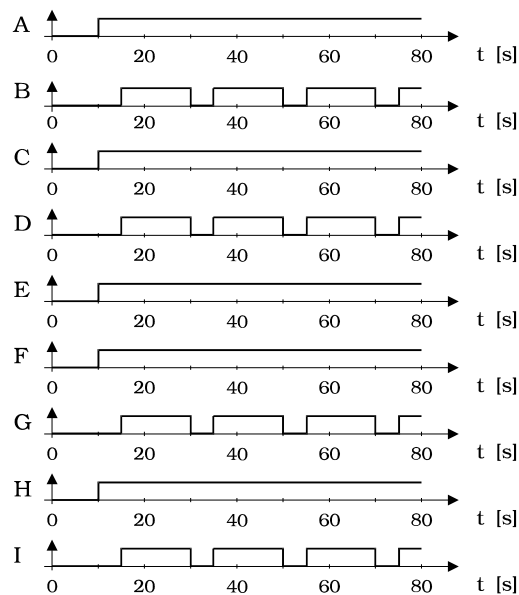
Primer 2:

```

U2 STIM(9, 1134) $G_DPWR G_DGND
+      A, B, C, D, E, F, G, H, I   IO_STIM
+    0S   0000
+ LABEL=LOOP
+  10S  105A      ; A=1, B=0, CDE=101, FGHI=1010
+  15S  117F      ; A=1, B=1, CDE=111, FGHI=1111
+  30S  GOTO LOOP 3 TIMES
          ; specifikacija 3 TIMES uzrokuje ponavljanje
          ; komandi unutar petlje 1+3=4 puta

```

Talasni oblici na izlazima A, ..., I generatora U2 prikazani su na slici 4.4.



Slika 4.4: Talasni oblici na izlazima pobudnog generatora U2

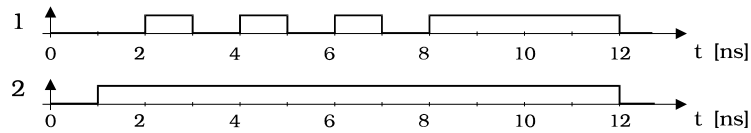
Primer 3:

```

U3 STIM(2, 11) $G_DPWR, $G_DGND, 1, 2 IO_STIM TIMESTEP=1NS
+ 0C 00
+ LABEL=LOOP
+ 1C 01      ; relativni pomak je +1C
+ 2C 11      ; relativni pomak je +1C
+ 3C GOTO LOOP 3 TIMES ; relativni pomak je +1C
+ 12C 00

```

Talasni oblici u izlaznim čvorovima 1 i 2 generatora U3 prikazani su na slici 4.5.



Slika 4.5: Talasni oblici na izlazima pobudnog generatora U3

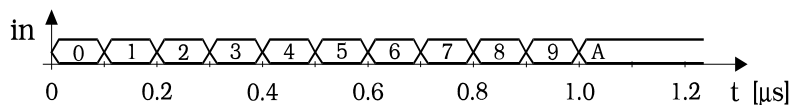
Primer 4:

```

U4    STIM(16, 4444) $G_DPWR G_DGND    in0, in1, in2, in3,
+      in4, in5, in6, in7, in8, in9, in10, in11,
+      in12, in13, in14, in15    io_std
+    TIMESTEP=1ns
+    0ns 0000
+    LABEL=LOOP
+    10c INCR BY 0001
+    20c GOTO LOOP UNTIL GE 000A

```

Talasni oblici u izlaznim čvorovima in0, ... , in15 generatora U4 (šesnaestobitna magistrala) prikazani su na slici 4.6 u simboličkom obliku.



Slika 4.6: Talasni oblici na izlazima pobudnog generatora U3

4.2.15 Globalni digitalni čvorovi

Globalni digitalni čvorovi su čvorovi koji se nalaze na fiksnom logičkom stanju tokom cele simulacije. *PSpice* ima predefinisana tri globalna digitalna čvora: \$D_HI, \$D_LO i \$D_X, za stanja '1', '0' i 'X', respektivno (tabela 4.12).

Ako se ulazni priključak logičkog elementa ili potkola veže za globalni digitalni čvor, stanje tog priključka biće određeno stanjem globalnog čvora i neće se menjati u toku cele simulacije.

Tabela 4.12: Globalni digitalni čvorovi

Oznaka čvora	Stanje čvora
\$D_HI	1
\$D_LO	0
\$D_X	X

4.2.16 Digitalno napajanje

Svi logički elementi u svom opisu moraju imati digitalni čvor napajanja i digitalni čvor uzemljenja. Po konvenciji, to su uvek prva dva čvora kod logičkih elemenata (U), a poslednja dva čvora kod potkola (X). Definisane napajanja i uzemljenja digitalnih elemenata ima primenu pri hibridnoj simulaciji, jer se za ove čvorove vezuju interfejsna potkola. Kod simulacije čisto digitalnih kola ovi čvorovi ne igraju nikakvu ulogu, ali se moraju navesti. U standardnoj biblioteci modela **nom.lib**, koja se instalira uz program **PSpice**, definisane su dva globalna digitalna čvora napajanja - \$G_DPWR i \$G_DGND koji se mogu koristiti u ovu svrhu.

Tabela 4.13: Globalni digitalni čvorovi napajanja

Oznaka čvora	Značenje
\$G_DPWR	pozitivni kraj digitalnog napajanja
\$G_DGND	negativni kraj digitalnog napajanja (masa)

Kako su imena ovih čvorova relativno duga, moguće je skratiti ih korišćenjem .TEXT naredbe na sledeći način:

```
.TEXT DP = "$G_DPWR"   DG = "$G_DGND"
```

Na svim mestima gde je potrebno koristiti globalne čvorove napajanja, oni se sada mogu pisati novim, kraćim imenima: DP i DG. Ova naredba može se smestiti i u neku *priključenu datoteku* čije se ime navodi u opisu kola korišćenjem naredbe .INC.

4.2.17 Komande za kontrolu toka i prikaz rezultata simulacije

Logička simulacija po svojoj prirodi predstavlja analizu rada kola u vremenskom domenu. Dakle, za zadavanje parametara simulacije koristićemo komandu .TRAN. Kako se logička simulacija odvija po principu narednog događaja, zadavanje koraka simulacije nije od značaja, već je potrebno zadati samo vreme završetka simulacije. Vreme završetka simulacije je drugi parametar .TRAN komande, pa se mora zadati i prvi parametar - interval štampanja. Njegova vrednost je proizvoljna.

Primer:

```
.TRAN 10ns 500ns ; int. štamp. 10ns, trajanje simul. 500ns
```

Kada kolo sadrži samo digitalne elemente, tada je moguća samo analiza u vremenskom domenu, pa simulator ignoriše komande za druge analize koje korisnik eventualno može da zada (.DC, .AC, ...).

Za prikaz rezultata logičke simulacije najpogodnije je koristiti grafički postprocesor **Probe**. U komandi .PROBE nije neophodno naglašavati da su izlazni čvorovi digitalni.

Primeri:

```
.PROBE  
.PROBE a, b, a2, in1, in2, out, 12, 13, 14
```

Osim komande .PROBE koja obezbeđuje grafički prikaz rezultata simulacije, rezultati se mogu dobiti u obliku liste u tekstualnoj izlaznoj datoteci. Za ovu svrhu koristi se komanda .PRINT. U komandi .PRINT digitalni čvorovi se navode u obliku D(<ime digitalnog čvora>), slično naponima i strujama kod analogne simulacije. Ako se .PRINT komanda koristi za štampanje samo digitalnih promenljivih, a to je kod simulacije čisto digitalnog kola uvek slučaj, može se koristiti komanda .PRINT/DGTLCHG. Tada se imena digitalnih čvorova navode direktno.

Primeri:

```
.PRINT TRAN D(a), D(b), D(in1), D(in2), D(out)  
.PRINT/DGTLCHG TRAN a, b, in1, in2, out
```

4.2.18 Najčešće greške u opisu kola

Ulazna datoteka sadrži opise digitalnih elemenata od kojih se kolo sastoji, opise pobudnih generatora, komandu .TRAN i komande za specificiranje načina prikaza rezultata simulacije (.PROBE i/ili .PRINT). U opisu kola mogu se pojaviti i druge komande, kao što su .PARAM, .INC, .LIB, .TEXT, .SUBCKT itd.

Pri formiranju ulazne datoteke često se prave greške. Najčešće greške nabrojane su u daljem tekstu.

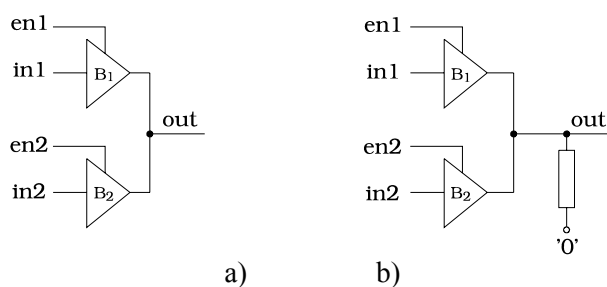
- Izostavljanje naslovne linije. Simulator smatra naslovnom linijom prvu liniju opisa kola, a kako je to u ovom slučaju neki od elemenata kola, opis kola postaje nepotpun. Ovo može izazvati nerazumljive poruke o grešci.

- Izostavljanje čvorova digitalnog napajanja u listi ulazno/izlaznih čvorova logičkog elementa. Simulator prijavljuje nedovoljan broj pristupnih čvorova ili nepostojanje vremenskog i U/I modela (jer se njihova imena smatraju pristupnim čvorovima elementa).

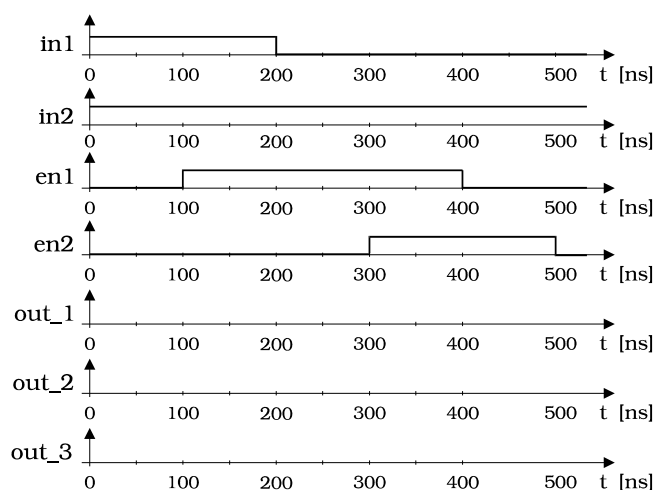
- Izostavljanje vremenskog sufiksa (s ili c) u specifikaciji logičkog pobudnog generatora. Simulator prijavljuje grešku.

Z A D A T A K

1. Opisati kolo sa slike 4.7a za simulaciju programom **PSpice**. Za trostatičke bafere koristiti nulto propagaciono kašnjenje. Na ulaze kola dovesti pobudne signale prikazane na slici 4.8 iz pobudnog generatora sa četiri izlaza. Trostatički baferi sa slike 4.7a imaju jednake izlazne otpornosti u stanju logičke jedinice i u stanju logičke nule - 100Ω . Ulazne i izlazne kapacitivnosti su zanemarive. Simulaciju završiti posle 1000ns. Za prikaz rezultata simulacije koristiti komande: `.PROBE` i `.PRINT/DGTLCHG` (ili `.PRINT`).



Slika 4.7: a) Magistrala sa dva drajvera bez i b) sa pull-down otpornikom

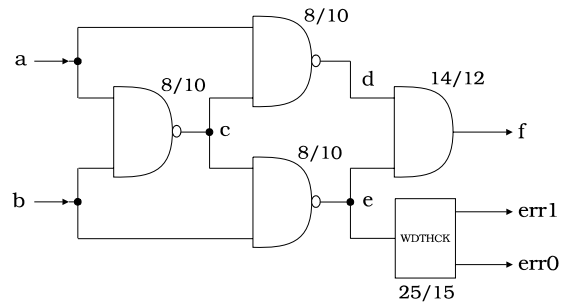


Slika 4.8: Pobuda za kolo sa slike 4.7. Na grafike označene sa out_1, out_2 i out_3 treba ucrtati rezultate simulacije.

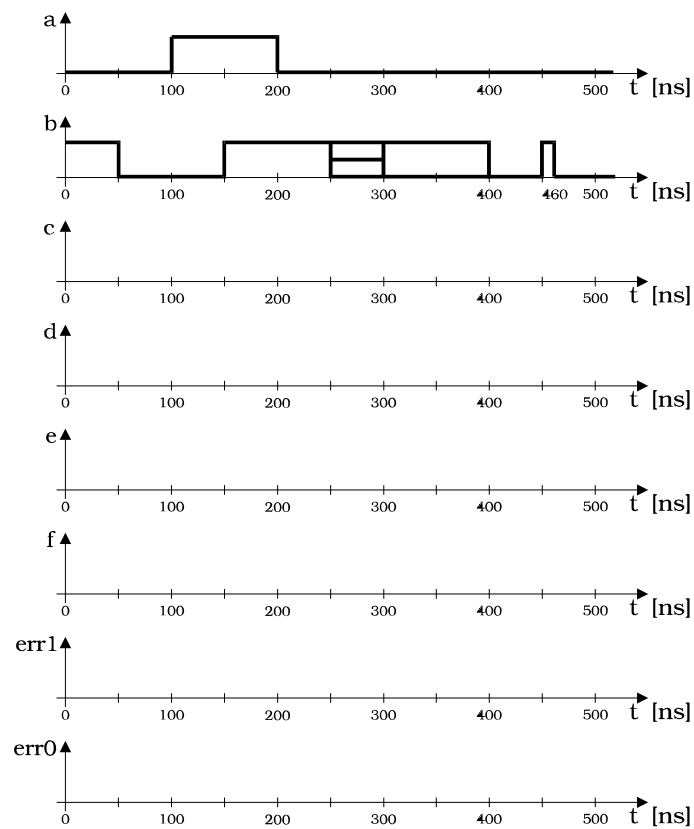
Na slici 4.8 skicirati talasni oblik signala u izlaznom čvoru out (grafik sa oznakom out_1). Priložiti listing ulazne datoteke.

2. Kolu dodati pull-down otpornik vrednosti $5k\Omega$, kao što je prikazano na slici 4.7b. Simulirati kolo sa istom pobudom kao u tački 1. Na slici 4.8 skicirati talasni oblik signala u izlaznom čvoru out (grafik sa oznakom out_2). Komentarisati rezultat simulacije.

3. Simulirati kolo sa slike 4.7b, ali tako da bafer B_1 ima izlazne otpornosti 100Ω , a bafer B_2 izlazne otpornosti $1k\Omega$ u oba logička stanja. Na ulaze kola dovesti istu pobudu kao u tačkama 1 i 2. Na slici 4.8 skicirati talasni oblik signala u izlaznom čvoru out (grafik sa oznakom out_3). Komentarisati rezultat simulacije.



Slika 4.9: Jedna realizacija kola ekvivalencije



Slika 4.10: Pobuda za kolo sa slike 4.9. U grafike označene sa c, d, e, f, err1 i err0 treba ucrtati rezultate simulacije.

4. Opisati kolo sa slike 4.9 za simulaciju programom **PSpice**. Vremena kašnjenja na slici data su u ns. Na ulaze kola dovesti pobudne signale prikazane na slici 4.10. Elementom vremenske provere `WDTHCK` ispitati trajanja impulsa logičke jedinice i logičke nule u čvoru `e`. Trajanje impulsa logičke jedinice ne sme biti kraće od 25ns, a trajanje impulsa logičke nule ne sme biti kraće od 15ns. Čvorovi `err1` i `err0` služe za detekciju nepoštovanja minimalnog dozvoljenog trajanja impulsa logičke jedinice i logičke nule, respektivno. Simulaciju završiti posle 500ns. Za prikaz rezultata simulacije koristiti komande: `.PROBE` i `.PRINT/DGTLCHG`.

Rezultate simulacije dočrtati na slici 4.10. Priložiti listing ulazne datoteke.

Datum : _____

Student : _____

Overa : _____

Vežba broj 5

Simulacija kola sa mešovitim signalima programom PSpice

5.1 Cilj vežbe

Upoznati se sa osnovnim principima analize kola sa mešovitim analognim i digitalnim signalima. Proučiti upotrebu programa *PSpice* za simulaciju elektronskih kola sastavljenih od analognih i digitalnih komponenata.

5.2 Teorijska postavka vežbe

5.2.1 Osnovni pojmovi o analizi kola sa mešovitim signalima

Savremena elektronska kola često obavljaju veoma složene funkcije za čiju realizaciju su neophodne i analogne i digitalne komponente. Za verifikaciju rada takvih kola nije racionalno koristiti isključivo električnu analizu jer bi tada i digitalni deo kola morao da se analizira na nivou kontinualnih signala, a ne na nivou logičkih stanja. S druge strane, primena logičkih simulatora za analizu ovih kola *sa mešovitim signalima* nije moguća jer je nemoguće modelovati analogne elemente logičkim funkcijama. Jedino ispravno rešenje svodi se na to da se analiza

analognog dela kola obavlja na nivou analognih signala, električnom analizom, a da se uvid u rad digitalnog dela kola dobije logičkom simulacijom.

Osnovne specifičnosti koje se javljaju kod analize analognih i kola sa mešovitim signalima (AMS kola) posledica su potrebe da se istovremeno manipuliše analognim i digitalnim signalima, a da se ne narušavaju njihove pojedinačne prednosti. Pri tome treba razrešiti sledeće probleme:

- sprega između analognih i digitalnih signala
- kontrola vremenskog koraka simulacije
- inicijalizacija simulacije.

Ukoliko se signalom iz analognog dela kola pobuđuje digitalni deo kola, kaže se da postoji A/D sprega, u suprotnom, radi se o D/A sprezi. Naravno, postoje slučajevi kada se signalima jednog tipa pobuđuju i analogni i digitalni deo kola. U svim takvim slučajevima radi se o takozvanim AMS čvorovima. Problem simulacije u takvim čvorovima rešava se ubacivanjem odgovarajućih kola za konverziju jednih signala u druge. Ova kola ćemo zvati *kola za spregu*. Model A/D kola za spregu realizuje se relativno jednostavno. Naime, model mora da sadrži definisane vrednosti pragova analognog signala između kojih se dodeljuju odgovarajuća logička stanja digitalnom signalu. Pored toga, u model mora da se uključi i uticaj ulazne kapacitivnosti digitalnog dela kola na analogni signal. D/A sprega je po svojoj prirodi mnogo kompleksnija. Model ovog kola mora, najpre, da sadrži preslikavanje logičkih stanja, definisanih simbolima koji označavaju relativni odnos njihovih nivoa, u vrednosti napona koje se prosleđuju programu za analizu analognog dela. Zbog toga je neophodno da se specificiraju vrednosti napona napajanja digitalnog dela kola jer one definišu vrednosti napona koje odgovaraju logičkoj nuli, odnosno logičkoj jedinici. Kada postoji više logičkih stanja, neophodno je da se uspostavi korespodencija između logičkih nivoa svih stanja sa odgovarajućim vrednostima napona. Pored toga, model mora da sadrži i informaciju o funkciji kojom se definiše prelazak između susednih naponskih nivoa. Naravno, neophodno je uzeti u obzir i uticaj izlazne impedanse digitalne komponente na signal koji pobuđuje analogni deo kola.

Kontrola vremenskog koraka simulacije ima ključni uticaj na ceo tok analize. Pri tome treba imati na umu da se paralelno odvijaju dve analize sa različitim veličinama vremenskog koraka. Električna analiza napreduje sa korakom koji zavisi od dinamike promena napona/struja u kolu i veličine najmanje lokalne vremenske konstante u kolu, dok logička simulacija ima vremenski korak vođen narednim događajem. Tokom hibridne simulacije, obrađuje se deo kola u kome se u tekućem trenutku očekuje promena. Naravno, u najvećem delu to je analiza

analognog dela kola. Kada se dođe do trenutka u kome se u tabeli budućih događaja otkrije promena u digitalnom kolu, ažuriraju se neophodne tabele. Konflikti mogu da nastanu u situacijama kada promena signala na digitalnom izlazu izaziva trenutnu promenu signala u analognom delu kola koja bi preko povratne veze mogla da poremeti vrednosti prethodno određenih veličina. U takvim situacijama neophodno je da se vreme 'vrati' i da se provere rezultati analize analognog dela kola pod novonastalim uslovima. S obzirom da se tokom logičke simulacije ide daleko brže po vremenskoj osi, ukupno vreme potrebno za hibridnu simulaciju diktira trajanje električne analize uz dodatak vremena neophodnog za razrešavanje konfliktnih situacija.

Inicijalizacija kola sa mešovitim signalima nešto je složenija nego što je to slučaj sa čisto analognim i digitalnim kolima. Naime, najjednostavnija inicijalizacija digitalnih kola dozvoljava da se u početku svim signalima dodeli neodređeno stanje 'X'. Međutim, u mešovitim D/A čvorovima, nemoguće je u tom slučaju izvršiti pravilnu konverziju digitalnog signala u analogni. Takođe, kada nije poznata vrednost analognog signala, on se ne može konvertovati u digitalni. Međusobni uticaj digitalnog i analognog dela kola u trenutku $t=0$ zahteva da se problem automatske inicijalizacije reši iterativno. Obavi se analiza logičkog dela kola sa nultim kašnjenjem u trenutku $t=0$, tako što se u A/D čvorovima postavi 'X' stanje. Na osnovu dobijenih rezultata obavi se analiza analognog dela kola sa konverzijom u D/A čvorovima od '0' $\rightarrow$ 'V', '1' $\rightarrow$ 'V⁺' i 'X' $\rightarrow$ $(V^+ + V^-)/2$, gde 'V⁺' označava pozitivnu, a 'V⁻' negativnu vrednost napona napajanja, respektivno. Zatim se ponovo simulira digitalni deo. Ovaj ciklus se ponavlja dok se ne dobiju dva uzastopno jednaka odziva digitalnog dela kola. Postoje slučajevi u kojima automatska inicijalizacija ne dovodi do rešenja. Tada je neophodno da projektant sam postavi granične uslove.

5.2.2 Analiza kola sa mešovitim signalima programom PSpice

PSpice je nastao iz jezgra koje čini program za analizu analognih elektronskih kola, dok je logički simulator ugrađen u njega naknadno. Iz tog razloga je u ovom programskom paketu zastupljen pristup odvojenih analiza analognog i digitalnog dela kola. Zato sve ono što je do sada rečeno o analizi analognih kola (vežbe 2 i 3), a takođe i sve ono što se tiče logičke simulacije (vežba 4) i dalje važi. Naravno, program vodi računa o prisustvu digitalnih elemenata prilikom DC i AC analize, kao i o uticaju analognih elemenata na rezultat logičke simulacije.

Kada kolo sadrži samo digitalne elemente tada je moguća samo vremenska analiza, dok se druge analize (.DC i .AC) ignorišu jer nemaju smisla. Analogna i kola sa mešovitim signalima mogu da se analiziraju u sva tri režima.

Jednosmerna analiza (.DC, .OP, .TF, .SENS) kola sa mešovitim signalima podrazumeva da se usvajaju nulta propagaciona kašnjenja svih digitalnih komponenata i da se svi digitalni pobudni generatori inicijaliziraju stanjem u nultom vremenskom trenutku. Naravno, sve kapacitivnosti u kolu tretiraju se kao otvoreni prekidač, a sve induktivnosti kao kratak spoj (zatvoreni prekidač), dok se u obzir uzimaju samo jednosmerne vrednosti strujnih i naponskih izvora.

Celokupni postupak DC analize kola sa mešovitim signalima suočava se sa problemima koji su pomenuti prilikom opisa inicijalizacije u prethodnom odeljku. Otuda je neophodno da se odvija iterativno.

Posle svake iteracije određuju se stanja na digitalnom delu mešovityh čvorova. Ukoliko posle ITL1 iteracija (ITL1 parametar u naredbi .OPTIONS definiše maksimalni broj iteracija) ne dođe do konvergencije iterativnog procesa, digitalni deo A/D čvora postavlja se u neodređeno stanje 'X'.

Pri simulaciji digitalnog dela kola ne može doći do divergencije, s obzirom da je stanje na izlazu uvek definisano, čak i u slučaju da je to neodređeno, 'X', stanje.

Ako digitalni element ne može doći u neko poznato stanje za vreme DC analize (npr. flip-flop čiji je takti ulaz u logičkom stanju '0') tada će se njegovom izlazu pridružiti logičko stanje 'X'.

AC analiza (.AC, .NOISE) kola sa mešovitim signalima uključuje uticaj linearnih ekvivalenata analognog dela potkola za spregu.

Vremenskoj analizi (.TRAN) kola sa mešovitim signalima obavezno prethodi inicijalizacija koja se dobija kao rezultat DC analize za $t=0$. Granične uslove analognih promenljivih čine vrednosti napona/struja u analognim čvorovima, dok su početna stanja digitalnih elemenata logička stanja u digitalnim čvorovima. Početna logička stanja mogu biti poznata, '0' i '1', a mogu biti i nepoznata, 'X'.

Za vreme analize, propagaciona kašnjenja iz vremenskog modela koriste se za formiranje tabela budućih događaja na izlazu. Pri tome, ignorišu se svi impulsi koji su kraći od zadatih vremena uspostavljanja i propagacionih kašnjenja. Ovo pravilo ne važi za liniju za kašnjenje.

Pri analizi AMS kola, postoje dva vremenska koraka. Jedan za kontrolu analogne, a drugi za kontrolu digitalne simulacije. To je neophodno zbog povećanja efikasnosti analize, jer se analogne i digitalne vremenske konstante obično razlikuju za nekoliko redova veličine. Vremenski korak koji se prikazuje na ekranu tokom analize odgovara analognom vremenskom koraku.

Digitalni vremenski korak određen je aktivnošću digitalnih elemenata i njihovim kašnjenjem. Zbog efikasnosti, vreme koje se koristi za procesiranje logičkog algoritma, kao i sva kašnjenja, normalizuju se sa minimalnim vremenskim korakom. Minimalni vremenski korak računa se po formuli $1/\text{DIGFREQ}$, gde je DIGFREQ parametar čija se vrednost definiše u .OPTIONS naredbi. Podrazumevana vrednost parametra DIGFREQ je 10GHz. Kašnjenja kraća od vremenskog koraka tretiraju se kao da su iste dužine kao i vremenski korak.

Specifičnosti analize kola sa mešovitim signalima nalažu da se pored pojmova vezanih za analognu i digitalnu simulaciju uvedu i neki novi pojmovi. Tako se uvodi pojam *tipa čvora*. Tip čvora određen je tipom elemenata koji su vezani za njega. Ako su svi ti elementi analogni, odnosno digitalni, onda se radi o analognom, odnosno digitalnom čvoru. Ako su za isti čvor vezani analogni i digitalni elementi, tada se radi o mešovitom čvoru, MČ. Po detektovanju takvog čvora program automatski dodaje A/D ili D/A potkola za spregu između njega i svakog digitalnog elementa vezanog za MČ. Na taj način originalni MČ se razbija u jedan analogni i jedan ili više digitalnih čvorova koji su razdvojeni potkolima za spregu. Ova aktivnost je neophodna da bi se logička stanja prevela u analogne signale i obrnuto. Pri tome, pojmove A/D i D/A potkola za spregu ne treba mešati sa pojmovima A/D i D/A konvertora koji su legalni digitalni elementi tipa U.

5.2.3 Vrste potkola za spregu

Potkola za spregu imaju zadatak da prevedu analogni napon i impedansu u digitalno stanje i obrnuto. Osnovne komponente potkola za spregu su N element (digitalni ulaz: DtoA) i O element (digitalni izlaz: AtoD). Ime potkola za spregu koje PSpice automatski dodaje sadržano je u U/I modelu logičkog elementa. U/I model prepoznaje imena četiri A/D i četiri D/A potkola za spregu (videti tabele 4.3 i 4.4). Potkolo za spregu specificirano u U/I modelu može da se nađe u ulaznoj datoteci opisa kola (sa ekstenzijom .CIR) ili u biblioteci koja mu je na raspolaganju. Postojanje više nivoa sprege omogućava izbor tačnosti digitalne simulacije. Četiri nivoa sprege omogućavaju izbor između jednostavnog i složenijeg modela, zavisno od željene tačnosti digitalne simulacije. Modeli složenijih potkola za spregu sadrže nelinearne komponente, tačnost im je bolja, ali usporavaju tok simulacije. Najčešće, primena jednostavnijih modela potkola za spregu daje zadovoljavajuće rezultate. Po konvenciji nivoi 1 i 2 označavaju jednostavnu, a nivoi 3 i 4 složenu funkciju prelaza između dva stanja. Nivoi 2 i 4 označavaju direktan prelaz iz stanja u stanje, bez prolaska kroz stanja 'R', 'F', 'X'. Imena potkola za spregu data su u tabeli 5.1.

Tabela 5.1: Opis značenja nivoa složenosti pojedinih tipova potkola za spregu

Nivo	Potkolo	Opis
1	AtoD1 i DtoA1	Linearni model sa 'X', 'R', 'F' stanjima
2	AtoD2 i DtoA2	Linearni model bez 'X', 'R', 'F' stanja
3	AtoD3 i DtoA3	Nelinearni model sa 'X', 'R', 'F' stanjima
4	AtoD4 i DtoA4	Nelinearni model bez 'X', 'R', 'F' stanja

5.2.4 Izbor potkola za spregu

Postoje dva načina izbora potkola za spregu koja će *PSpice* automatski dodavati mešovitim čvorovima. Jedan način je definisanjem vrednosti parametra DIGIOLVL u okviru naredbe .OPTIONS, na nivou celog kola. Podrazumevana vrednost parametra DIGIOLVL odgovara nivou 1.

Primer:

```
.OPTIONS DIGIOLVL=3      ;svim hibridnim cvorovima dodaje
se                        ;nelinearno potkolo za spregu sa
                          ;X, R, F stanjima
```

Drugi način predstavlja definisanje nivoa potkola za spregu za svaki digitalni element u kolu posebno, korišćenjem IO_LEVEL parametra u opisu U elementa ili IO_LEVEL parametra pri pozivu potkola definisanog .SUBCKT naredbom. Vrednost definisana sa IO_LEVEL poništiće vrednost DIGIOLVL za taj digitalni element ili potkolo. Podrazumevana vrednost parametra IO_LEVEL je 0, što znači da će biti korišćena vrednost parametra DIGIOLVL.

Primer:

```
X1 in out 74LS04 PARAMS:IO_LEVEL=2 ;mesovitim cvorovima za
                                   ;koje je vezan element X1 dodaje
                                   ;se linearno potkolo za
                                   ;spregu bez X, R, F stanja.
```

5.2.5 Modelovanje potkola za spregu i imena čvorova

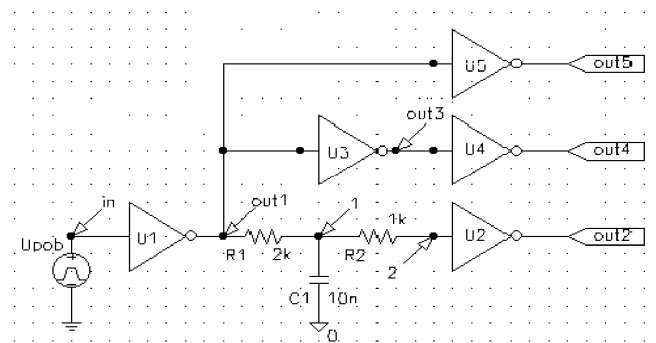
Kao što je već ranije rečeno, *PSpice* rešava problem mešovitih čvorova automatskim dodavanjem potkola za spregu. Kada se prepozna mešoviti čvor, poziva se potkolo za spregu za svaki digitalni element vezan za taj čvor. Na taj način MČ ostaje analogni za sve ostale analogne elemente vezane za taj čvor, uključujući i analogni deo potkola za spregu. Da bi se kompletirala originalna

veza, kreira se novi digitalni čvor između digitalnog priključka potkola za spregu i priključka digitalnog elementa. S obzirom da se novi čvor kreira automatski, mora mu se dodeliti jedinstveno ime. Da bi se posmatrali talasni oblici odziva u mešovitim čvorovima primenom naredbe .PROBE, potrebno je znati pravila koja koristi PSpice pri automatskom dodeljivanju imena čvorova.

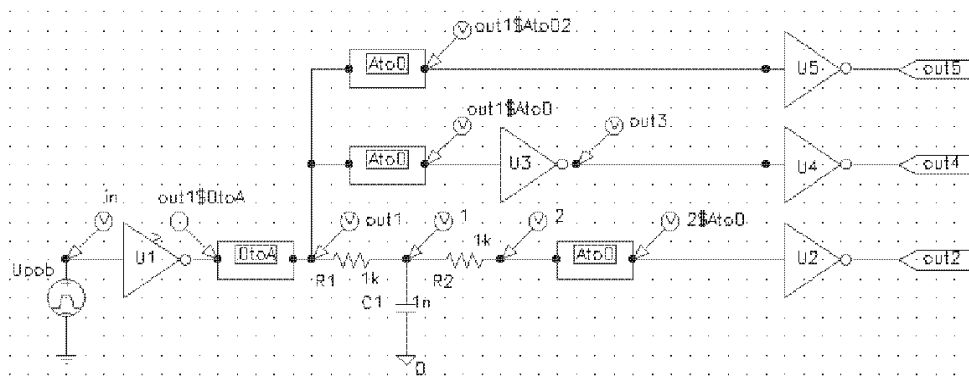
- Analogni čvor zadržava ime originalnog hibridnog čvora.
- Ime digitalnog dela hibridnog čvora označava se tako što se originalnom imenu hibridnog čvora dodaje sufiks "\$AtoD" za A/D potkolo za spregu, odnosno "\$DtoA" za D/A potkolo za spregu.
- Ako je više digitalnih elemenata vezano za isti hibridni čvor, dodaju se sufiksi "\$AtoD", "\$AtoD2", "\$AtoD3",... ili "\$DtoA", "\$DtoA2", "\$DtoA3",...

Primer:

Na slici 5.1a prikazano je originalno kolo kakvo je opisano u ulaznoj datoteci (ime.CIR), dok je na slici 5.1b prikazano kolo prošireno potkolima za spregu u mešovitim čvorovima. Postoje dva MČ: out1 i 2. Za čvor out1 vezana su tri digitalna elementa. Svaki od njih spreže se sa analognim delom kola preko odgovarajućeg potkola za spregu. Za izlaz invertora U1 vezuje se potkolo DtoA tipa, a digitalni deo čvora dobija ime out1\$DtoA. Između analognog dela čvora out1 koji zadržava originalno ime i ulaza invertora U3 i U5 vezuju se potkola AtoD tipa. Novouvedenim digitalnim čvorovima dodeljuju se imena out1\$AtoD i out1\$AtoD2 elemenata. Za čvor 2 vezan je ulaz invertora U2, tako da se ubacuje potkolo za spregu tipa AtoD. Novouvedeni digitalni čvor dobija ime 2\$AtoD.



Slika 5.1a: Originalne oznake čvorova u obliku koji je zadao projektant



Slika 5.1b: Novouvedeni čvorovi posle ubacivanja potkola za spregu u mešovitim čvorovima kola sa slike 5.1a

5.2.6 Grafičko prikazivanje odziva u mešovitim čvorovima

Za grafičko prikazivanje rezultata simulacije koristi se naredba `.PROBE`. Prostim navođenjem naredbe `.PROBE` omogućava se prikazivanje talasnih oblika odziva u svim čvorovima kola. Na ekranu se, tada, mogu istovremeno posmatrati svi analogni i digitalni rezultati simulacije. Ne treba gubiti iz vida da su svim digitalnim delovima hibridnih čvorova automatski dodeljena nova imena.

Ako se traži talasni oblik odziva u određenom mešovitom čvoru, tada se za posmatranje odziva u analognom delu čvora navodi oznaka

`V<(originalno ime čvora)>.`

Prilikom posmatranja odziva u digitalnom delu hibridnog čvora navodi se ime hibridnog čvora sa odgovarajućim sufiksom.

5.2.7 Format opisa potkola za spregu

Osim potkola za spregu koja su ugrađena u biblioteke programa *PSpice*, A/D i D/A potkola za spregu može da kreira i korisnik. S obzirom da su ona kreirana naredbom `.SUBCKT`, pri kreiranju sopstvenih A/D i D/A potkola za spregu, neophodno je da se koristi sledeća sintaksa:

```
.SUBCKT AtoD<ime> A, D, $G_DPWR, $G_DGND,
+      PARAMS: CAPACITANCE=<vrednost>
*
```

```

* definisanje potkola: digitalni izlaz, kapacitivnost
itd...
*
.ENDS

```

Opis A/D potkola mora da sadrži specifično ime sa prefiksom `AtoD`, četiri čvora po navedenom redosledu (gde A označava analogni ulazni čvor, D - digitalni izlazni čvor, `$G_DPWR` - čvor digitalnog napajanja i `$G_DGND` - čvor digitalne mase) i parametar `CAPACITANCE`, koji odgovara ulaznoj kapacitivnosti digitalnog elementa. Pri pozivu ovog potkola parametru `CAPACITANCE` pridružuje se vrednost `INLD` parametra iz U/I modela.

D/A potkolo za spregu ima sličnu sintaksu:

```

.SUBCKT DtoA<ime> D A $G_DPWR $G_DGND
+ PARAMS:DRVH=<vrednost> DRVH=<vrednost>
+ CAPACITANCE=<vrednost>
*
* definisanje potkola: digitalni ulaz, kapacitivnost
itd..
*
.ENDS

```

D/A potkolo za spregu mora da sadrži originalno ime sa prefiksom `DtoA`, četiri čvora koja se navode po navedenom redosledu (D - digitalni ulazni čvor, A - analogni izlazni čvor, `$G_DPWR` - čvor digitalnog napajanja i `$G_DGND` - čvor digitalne mase) i tri parametara: `DRVH` - izlazna otpornost logičkog stanja 0, `DRVH` - izlazna otpornost logičkog stanja 1, `CAPACITANCE` - izlazna kapacitivnost. *PSpice* će pri pozivu ovog potkola tim parametrima pridružiti vrednosti `DRVH`, `DRVH` i `INLD` parametara iz U/I modela.

5.2.8 Svrha digitalnog napajanja

Kao što je rečeno u poglavlju 4.2.16, opis svih digitalnih komponenta (logički elementi i logički pobudni generatori) mora da sadrži digitalni čvor napajanja i digitalni čvor uzemljenja. Po konvenciji, to su uvek prva dva čvora kod svih U elemenata i poslednja dva čvora kod digitalnih X elemenata (potkola).

Definisanje napajanja i uzemljenja digitalnih elemenata dobija praktičnu primenu tek kod analize kola sa mešovitim signalima prilikom definisanja A/D i D/A potkola za spregu. U slučajevima kada je digitalni element vezan za digitalne čvorove na svim pristupima, ovi čvorovi ne igraju nikakvu ulogu, ali se moraju navesti. Kada program *PSpice* pozove A/D ili D/A potkolo za spregu specificirano u I/O modelu, on mu tada pridružuje imena čvorova digitalnog napajanja i uzemljenja koja su naznačena u opisu digitalnog elementa.

Na primer, ako su u opisu U elementa digitalni čvorovi napajanja i uzemljenja označeni sa DigPwr i DigGnd, respektivno :

```
U1 BUF DigPwr DigGnd IN OUT BUF_TMod BUF_IOM
```

i ukoliko je čvor IN hibridnog tipa, *PSpice* će automatski za taj čvor vezati odgovarajuće AtoD potkolo za spregu (recimo AtoD1) pridružujući mu digitalne čvorove napajanja date u opisu U elementa:

```
X1$IN_AtoD1 IN IN$AtoD DigPwr DigGnd AtoD1
+                                     PARAMS:CAPACITANCE=0
```

Treba uočiti da poziv ovog potkola ostaje 'nevidljiv' za korisnika sve do završetka analize. Na kraju analize, novi digitalni deo hibridnog čvora IN postaje 'vidljiv' pod nazivom IN\$AtoD.

5.2.9 Potkolo digitalnog napajanja

U pridruženoj biblioteci **Dig_io.LIB** koja je obično priključena uz **Digital.LIB**, odnosno **NOM.LIB** nalazi se potkolo sa imenom DIGIFPWR u kome su specificirane podrazumevane vrednosti globalnog digitalnog čvora napajanja, označenog sa \$G_DPWR, od 5.0 V i čvora digitalnog uzemljenja, označenog sa \$G_DGND, od 0 V. Potkolo digitalnog napajanja DIGIFPWR definisano je skupom naredbi:

```
.subckt DIGIFPWR AGND
+           optional: DPWR=$G_DPWR DGND=$G_DGND
+           params:   VOLTAGE=5.0v REFERENCE=0v
*
VDPWR  DPWR DGND {VOLTAGE}
R1      DPWR AGND 1MEG
VDGND  DGND AGND {REFERENCE}
R2      DGND AGND 1MEG
X1      AGND      ECL_10K_PWR      ;napajanje za ECL kola
X2      AGND      ECL_100K_PWR     ;napajanje za ECL kola

.ends
```

U opštem slučaju ova dva globalna čvora za digitalno napajanje mogu se koristiti kod U elemenata i kod digitalnih potkola.

Potkola ECL_10K_PWR i ECL_100K_PWR koriste se za definisanje digitalnog

napajanja u ECL kolima kod kojih se napajanje realizuje na drugačiji način u odnosu na ostala digitalna kola. Detalji o tome mogu se naći u [-91].

5.2.10 Definisane sopstvenog digitalnog napajanja

Kada se ukaže potreba za digitalnim napajanjem koje je različito od 5V, moguće je definisati sopstveno napajanje promenom vrednosti parametra VOLTAGE pri pozivu potkola DIGIFPWR.

Primer:

Ako se želi napajanje od +2.5V na čvoru MyPwr i -2.5V na čvoru MyGnd potkolo DIGIFPWR treba pozvati na sledeći način:

```
XMyPower MyPwr MyGnd DIGIFPWR PARAMS: VOLTAGE=2.5V  
+ REFERENCE=-2.5V
```

Da bi se koristilo ovako definisano digitalno napajanje za digitalna potkola koja se pozivaju iz biblioteke, potrebno je naznačiti samo imena čvorova za digitalno napajanje (MyPwr i MyGnd) pri pozivu tog potkola:

```
XCD4011A In1 In2 Out MyPwr MyGnd CD4011A
```

Naznačeni čvorovi prekrće ranije podrazumevane vrednosti digitalnog napajanja.

Novokreirano digitalno napajanje može se koristiti ravnopravno i kod U elemenata. Kod njih su, po konvenciji, čvorovi za digitalno napajanje prva dva navedena čvora:

```
U1 NAND(2) MyPwr MyGnd In1 In2 Out TimingModel IOModel
```

Kako se digitalno napajanje koristi samo za A/D i D/A potkola za spregu, to nema potrebe da se u čisto digitalnim kolima definiše sopstveno digitalno napajanje. U takvim kolima koriste se globalni digitalni čvorovi napajanja, \$G_DPWR, i uzemljenja, \$G_DGND. Međutim ako se, ipak, definiše sopstveno digitalno napajanje, to će usporiti simulaciju [-91].

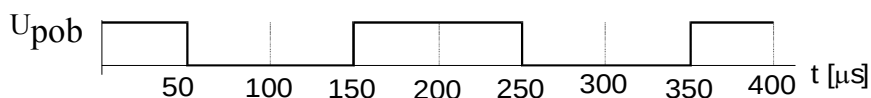
5.2.11 Najčešće greške u opisu kola

- Poštovati napomene iz poglavlja 4.2.18.
- Program ne prepoznaje potkola za spregu. Ovo je posledica ne priključivanja biblioteke **Dig_io.LIB** opisu kola. Greška se eliminiše navođenjem imena biblioteke u .LIB komandi.

- Zamenjen redosled navođenja čvorova. Ova greška nastaje kao posledica zanemarivanja činjenice da se redosled navođenja čvorova digitalnog napajanja i ulazno/izlaznih čvorova u digitalnim U elementima i potkolima (X elementi) razlikuje (videti 5.2.10).

5.2.12 Ilustrativni primer

Kao primer hibridne analize razmotrićemo kolo sa slike 5.1a. Treba odrediti talasne oblike napona i struja u kolu pod pretpostavkom da je kolo pobuđeno povorkom impulsa prikazanom na slici 5.2. Analizu završiti posle vremena $T=400\mu\text{s}$. Opis ulazne liste prikazan je na slici 5.3, a deo rezultata simulacije na slici 5.4.



Slika 5.2: Pobudna povorka impulsa za kolo sa slike 5.1a

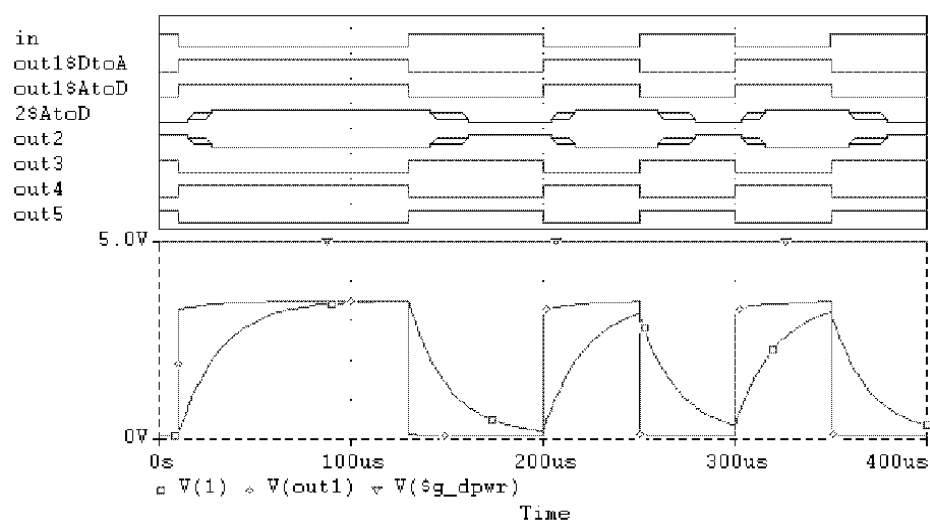
```

Ilustrativni primer - kolo sa slike 5.1a
.lib "Dig_io.lib"
r1 out1 1 2k
*
r2 1 2 1k
*
c1 1 0 10n
u1 inv $g_dpwr $g_dgnd in out1 tminv io_std
u2 inv $g_dpwr $g_dgnd 2 out2 tminv io_std
u3 inv $g_dpwr $g_dgnd out1 out3 tminv io_std
u4 inv $g_dpwr $g_dgnd out3 out4 tminv io_std
u5 inv $g_dpwr $g_dgnd out1 out5 tminv io_std
.model tminv ugate (tphlmn=2ns tplhty=3ns tplhmx=4ns
+                    tphlmn=4ns tplhty=5ns tplhmx=7ns)
.model io_std UIO()
upob stim(1, 1) $g_dpwr $g_dgnd in io_std
+      0us 1
+      50us 0
+      150us 1
+      250us 0
+      350us 1
.tran 1us 400us
.probe

```

```
.end
```

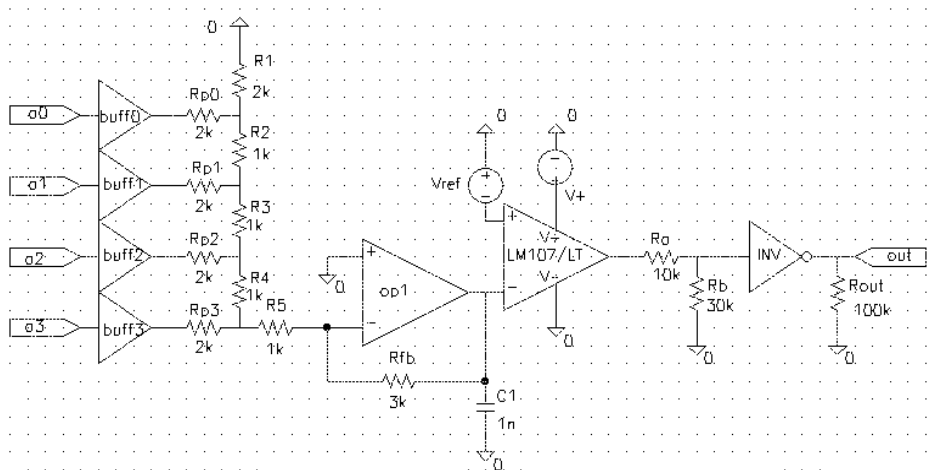
Slika 5.3: Ulazna lista za simulaciju kola sa slike 5.1a



Slika 5.4: Deo rezultata simulacije kola sa slike 5.1a

Z A D A T A K

1. Primenom programa *PSpice* analizirati kolo sa slike 5.5. Na slici označiti imena čvorova koja će biti korišćena u opisu kola. Za opis ulaznih bafera koristiti U element tipa `BUFFA`, niz bafera. Operacioni pojačavač opisati modelom definisanim kao potkolo iz vežbe broj 3 sa $R_{ul}=1M\Omega$, $C_{ul}=2pF$, $R_{iz}=50\Omega$ i $A=100dB$. Model komparatora LM107/LT definisan je u biblioteci **Lin_tech.LIB** potkolom koje se zove LM107/LT, a redosled navođenja priključaka je +ulaz, -ulaz, V+, V-, izlaz. Poznato je $V_{ref}=2.5V$, $V+=5V$.



Slika 5.5: Četvorobitni D/A konvertor

2. Na slici 5.6 ucrtati odgovarajuća potkola za spregu koja se automatski ubacuju na mestima hibridnih čvorova i naznačiti imena digitalnih i analognih delova ovih čvorova.

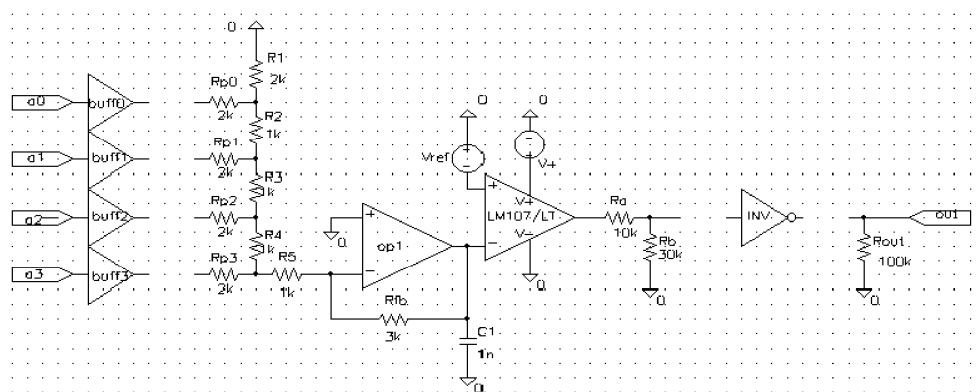
3. Primenom naredbe `.DC` analizirati ponašanje kola ako se vrednost napona V_{ref} menja od $-3V$ do $+3V$ sa korakom od $0.2V$, a kolo na ulazima $a3$, $a2$, $a1$ i $a0$ pobuđuje heksadecimalna reč 0 (svi ulazi u stanju 0). Postprocesorom **Probe** posmatrati napone na:

- ulazima komparatora
- izlazu komparatora
- potrošaču

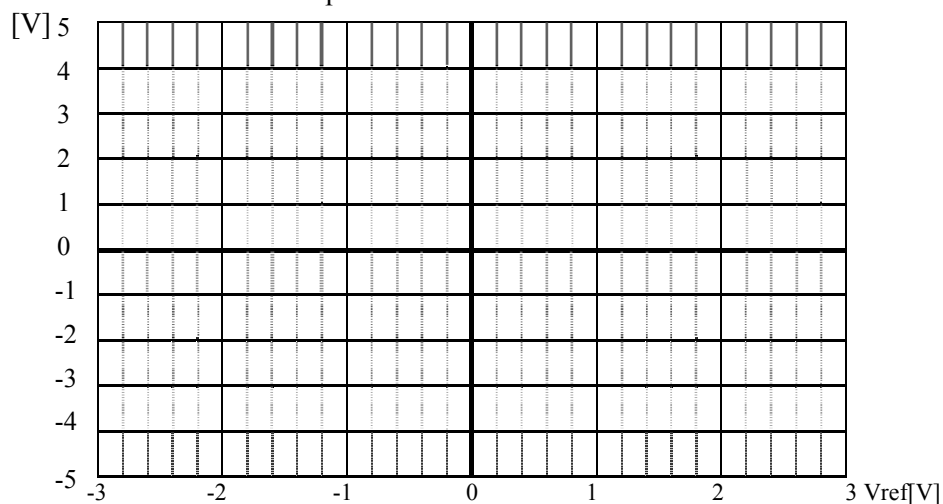
Dobijene rezultate skicirati u koordinatnom sistemu sa slike 5.7.

4. Ponoviti analizu iz tačke 3 za ulaznu reč kojoj odgovara heksadecimalna vrednost 5 (ulazu a0 odgovara bit najmanje težine).

Da li i kako digitalni elementi utiču na rezultat analize u stacionarnom režimu?



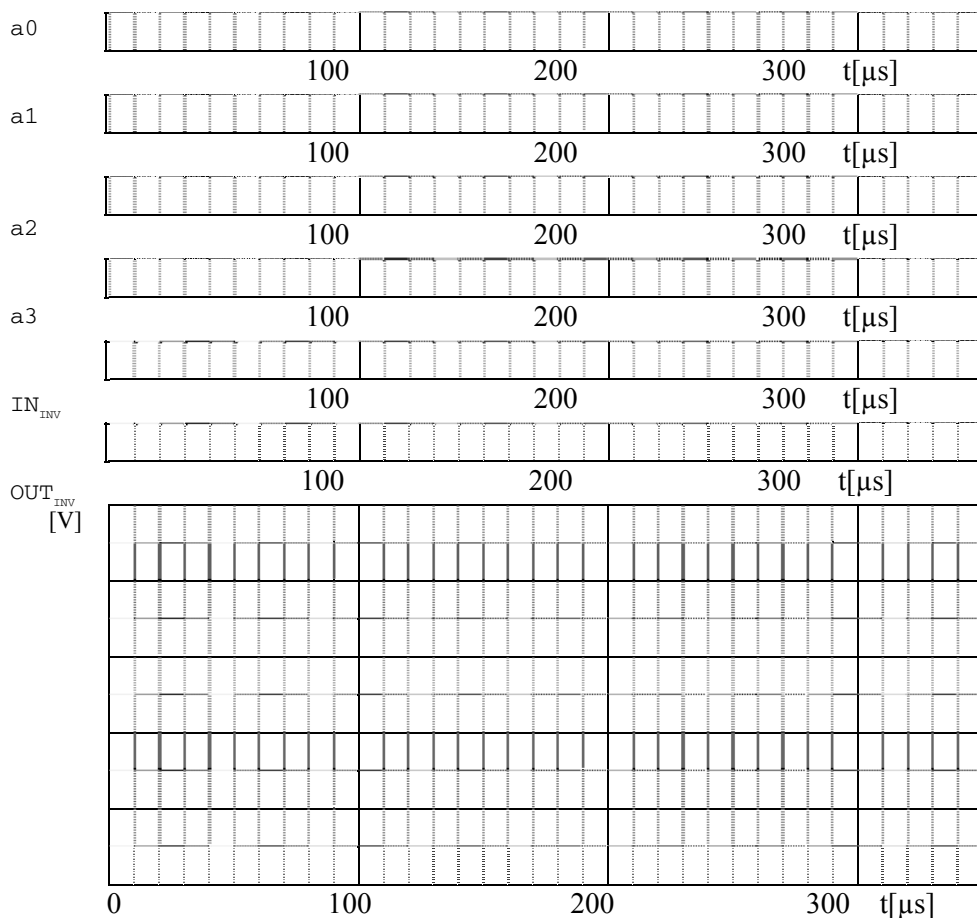
Slika 5.6: Ucertati blokove sa odgovarajućim tipovima potkola za spregu i upisati nazive svih čvorova



Slika 5.7: Rezultati dobijeni analizom kola sa slike 5.5 u stacionarnom stanju

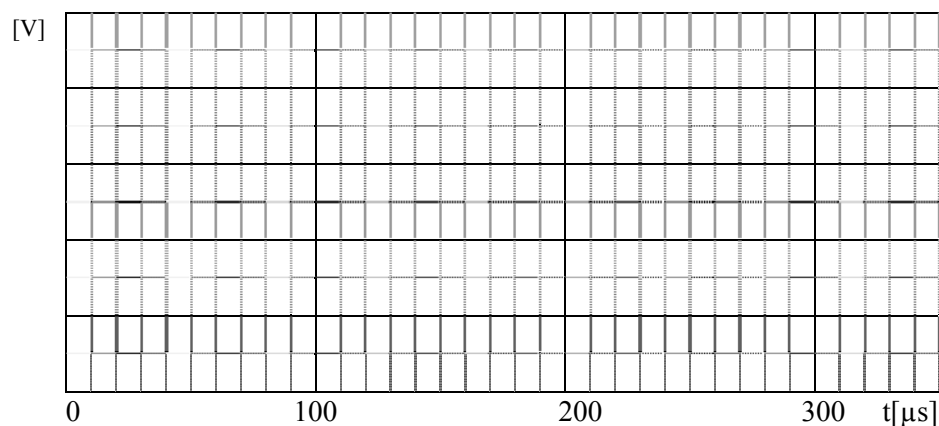
5. Analizirati kolo sa slike 5.5 u vremenskom domenu kada se kolo pobudi signalima kojima odgovara promena ulazne reči od 0 do F sa korakom 1.

Inkrementirati stanje na ulazu impulsima sa podjednakim dužinama trajanja logičke nule i jedinice od po $10\mu\text{s}$. U priloženim koordinatnim sistemima na slici 5.8 skicirati talasne oblike signala dobijenih postprocesorom **Probe**. Prikazati digitalne signale na ulaznim čvorovima (a_0 , a_1 , a_2 , a_3), na ulazu i izlazu invertora, kao i analogne signale na ulazima i izlazu komparatora i na otpornika R_{out} .



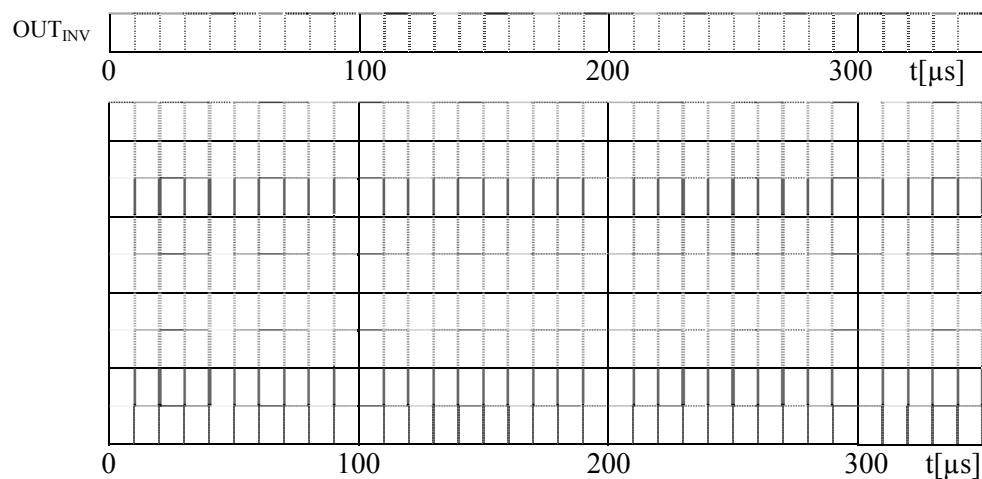
Slika 5.8: Rezultati dobijeni analizom kola definisanom u tački 5

6. Komandom `.STEP` menjati vrednost otpornika R_{fb} od 1k do 3k sa korakom 1k. Ponoviti analizu iz tačke 5. Posmatrati talasne oblike napona na ulazima i izlazu komparatora. Dobijene rezultate skicirati u priloženi koordinatni sistem sa slike 5.9. Objasniti ulogu otpornika R_{fb} .



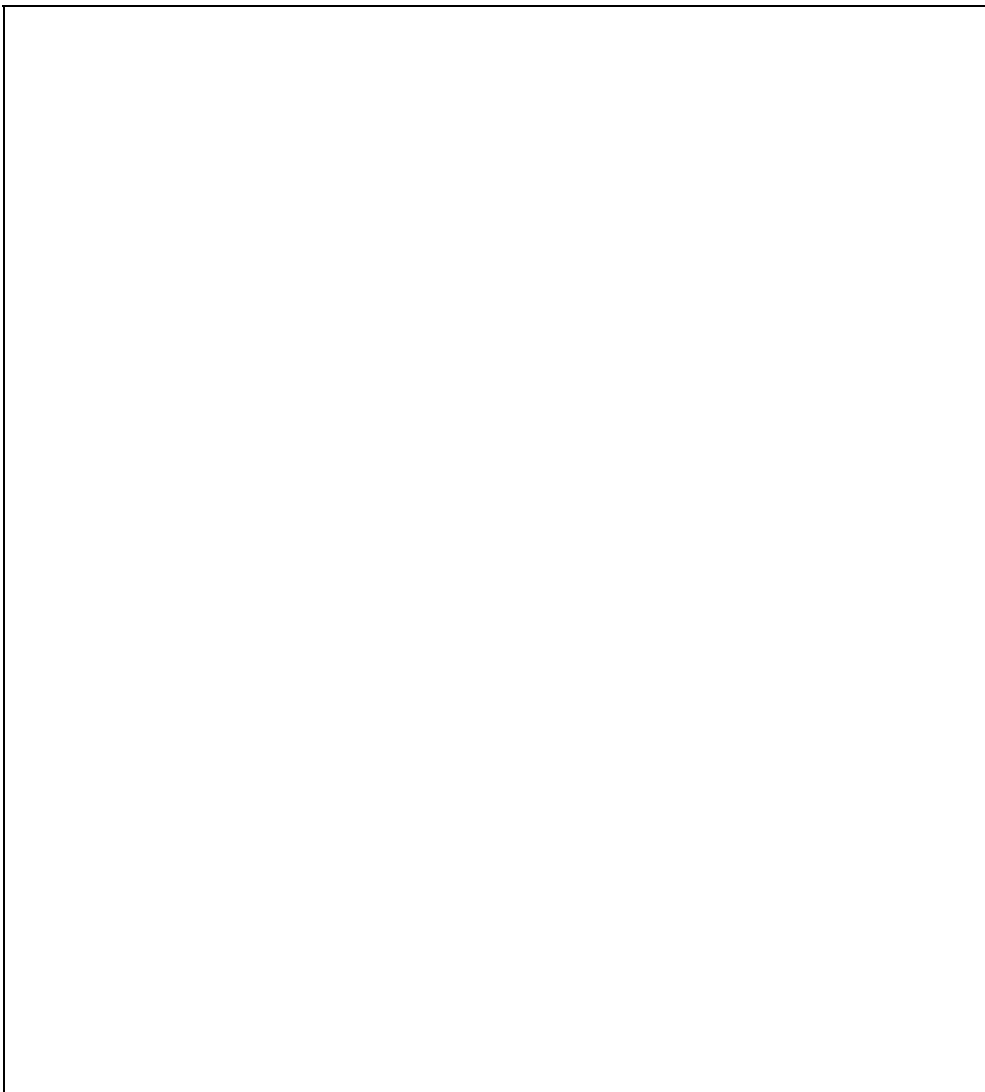
Slika 5.9: Rezultati dobijeni analizom kola definisanom u tački 6

7. Ponoviti analizu kola sa slike 5.5 definisanu u tački 5 kada se umesto bafera na ulaze kola vežu invertori. Za ovu svrhu koristiti U element tipa INVA, niz invertora. U koordinatni sistem sa slike 5.10 ucrtati talasni oblik digitalnog signala na izlazu i analognih signala na krajevima komparatora.



Slika 5.10: Rezultati dobijeni analizom kola definisanom u tački 7

8. Priložiti listing ulazne datoteke.



Datum : _____

Student : _____

Overa : _____

Vežba broj 6

Optimizacija vrednosti parametara elektronskog kola

6.1 Cilj vežbe

Proučiti upotrebu programa *PSpice Optimizer (PSO)* za optimizaciju parametara elektronskog kola.

6.2 Teorijska postavka vežbe

PSO je program za optimizaciju parametara analognih i elektronskih kola sa mešovitim signalima.

- Optimizacija predstavlja iterativni postupak koji ima za cilj da se odrede nepoznate vrednosti parametara kola pomoću kojih će se dobiti željeni odziv uz određena ograničenja. Ograničenja mogu da budu zadata u obliku opsega dozvoljenih vrednosti parametara i/ili funkcija koje zavise od parametara kola. U svakom koraku optimizacije kolo se analizira programom *PSpice* kako bi se odredila vrednost odziva za tekuće vrednosti parametara kola. Istovremeno, računaju se koeficijenti osetljivosti odziva na svaki od parametara kola.

- Da bi se proverilo odstupanje dobijene tekuće vrednosti odziva od željene, program obezbeđuje grafički prikaz odziva za svaki novi set vrednosti parametara.

- Određivanje funkcije greške predstavlja sastavni deo postupka optimizacije:

Kada se unesu nove vrednosti za parametre projektovanja, *PSO* obezbeđuje povratnu informaciju tako što grafički prikaže nove performanse odziva. Mogu se menjati i ograničenja da bi se ispitala promene vrednosti parametara.

- Podešavanje parametara modela: Kada je dat parametrizovani model, skup merenih podataka i dobra početna vrednost parametra, *PSO* podešava tačniji model.

6.2.1 Slučajevi koje je moguće optimizovati

Optimizovanje parametara kola moguće je samo ukoliko su ispunjeni sledeći kriterijumi:

- Analiza kola je moguća: svi parametri potrebni za analizu programom *PSpice* uneti su ispravno, tako da se analiza kola obavlja do kraja, a rezultati analize daju očekivane rezultate (npr. tranzistori rade u željenom režimu).
- Željeni odziv zavisi od svih parametara kola čije se vrednosti određuju.
- Postoji algoritam kojim se meri odstupanje odziva od željene vrednosti (definisana funkcija cilja).

Svi primeri kod kojih se može registrovati promena odziva sa promenom vrednosti parametara koji se optimizuju predstavljaju dobre kandidate za primenu programa *PSO*.

6.2.2 Slučajevi koje se ne mogu optimizovati

Program *PSO* se ne koristi za:

- Inicijalno određivanje parametara kola (podrazumeva se da su poznate početne vrednosti parametara koje obezbeđuju da se tranzistori nalaze u željenom radnom režimu).
- Optimizaciju u kolima kod kojih promena jednog ili više parametara utiče na promenu režima rada kola (npr. za malu promenu nekog parametra flipflop menja stanje).

6.2.3 Osnovni termini

Parametar

Parametar predstavlja veličinu čija se vrednost traži kako bi odziv imao željenu vrednost ili se našao u okviru specificiranih granica. Parametar može da predstavlja:

- vrednost komponente (kao što je otpornost, *R*, za otpornik, kapacitivnost kondenzatora, ...)
- vrednosti atributa komponente
- deo izraza koji se koriste za definisanje vrednosti komponente ili

vrednosti atributa komponente.

Program podržava optimizaciju ne više od osam parametara kola.

Specifikacija

Specifikacija opisuje željeno ponašanje odziva kola i mogućih ograničenja. (Na primer, očekuje se da: pojačanje bude $20\text{ dB} \pm 1\text{ dB}$ ili vreme uspostavljanja prednje ivice izlaznog napona mora da bude manje od $1\mu\text{s}$).

Specifikacije se mogu zadati u dva formata:

- interni
- eksterni.

Interna specifikacija se sastoji od ciljeva i ograničenja definisanih kao željene vrednosti odziva i odgovarajućih opsega. Oni se unose putem grafičkog interfejsa (prozora za dijalog).

Eksterna specifikacija se sastoji od seta (merenih) podataka. Oni su definisani u tekstualnom fajlu koji program *PSO* učitava.

Željena vrednost odziva

Željena vrednost predstavlja idealnu vrednost odziva koja se definiše specifikacijom.

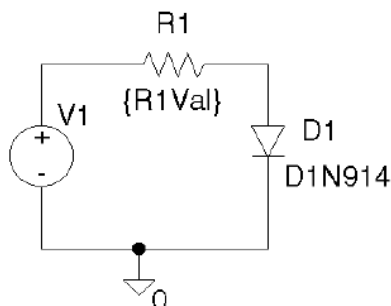
Cilj

Cilj definiše nivo koji projekat treba da dostigne. Specifikacija cilja uključuje:

- naziv cilja
- željenu vrednost i prihvatljivi opseg
- fajl sa opisom kola koje se simulira
- tip analize koja se koristi u simulaciji kola

6.2.4 Optimizacija vrednosti parametara kola za polarizaciju diode

Pretpostavimo da je potrebno odrediti optimalnu vrednost otpornosti otpornika $R1$ u kolu sa slike 6.1 koja će obezbediti da kroz diodu (D1N914) protiče struja od 1mA ($\pm 5\mu\text{A}$) kada se kolo napaja iz jednosmernog naponskog generatora naponom od 5V .



Slika 6.1: Kolo za polarizaciju diode

Iz postavljenog uslova zaključuje se da treba pronaći takvu vrednost otpornosti $R1$ koja obezbeđuje da struja kroz diodu $D1$ bude u opsegu od 0.995mA do 1.005mA .

6.2.5 Čemu služi program za optimizaciju?

Da bi se prethodni zadatak rešio moguće je dodeliti proizvoljnu vrednost otpornosti $R1$, izračunati struju i ukoliko se ne poklapa sa željenom veličinom, pogađati vrednost otpornosti sve dok se ne dobije zadovoljavajuća vrednost struje kroz diodu. Alternativno, može se raditi DC analiza kod koje je otpornost $R1$ uzeta kao parametar zadat svojim opsegom i korakom promene.

Ova dva metoda imaju sledeće nedostatke:

- S obzirom da je dioda nelinearni element, izračunavanje vrednosti struje sa proizvoljnim vrednostima otpornosti $R1$ može da bude veoma sporo.
- Simulacija sa parametrizovanom vrednošću otpornosti može da traje veoma dugo, naročito ako se izabere preširok opseg parametra i premali korak promene tj. inkrement.

Za razliku od navedenih slučajeva program za optimizaciju zasnovan je na primeni algoritma kojim se, na osnovu rezultata analize i vrednosti koeficijenata osetljivosti odziva, izračunava nova vrednost otpornosti $R1$. Zato se, umesto nasumične promene vrednosti otpornosti, izračunava nova vrednost koja u narednom koraku dovodi do smanjenja razlike između željenog i dobijenog odziva (u ovom slučaju struje kroz diodu). Na taj način eliminišu se nepotrebne iteracije, što dovodi do bržeg nalaženja rešenja i sa manje napora.

Kada se pronađe zadovoljavajuće rešenje, ono se može i dodatno popravljati.

Iterativno traženje najboljeg rešenja računskim putem može da bude vrlo težak i frustrirajući proces. Korišćenjem programa za optimizaciju, sa svakom promenom vrednosti parametara, može se pratiti uticaj na specificirani odziv. Takođe se mogu menjati specificirane vrednosti i istovremeno pratiti njihov uticaj na promenu vrednosti parametara. Ukoliko korisnik nije zadovoljan rezultatima nakon bilo koje promene, uvek se može vratiti poslednjoj dobijenoj vrednosti parametara koja daje zadovoljavajuće rezultate.

6.2.6 Tok projektovanja

S obzirom da je analiza kola sastavni deo optimizacionog algoritma, neophodno je da se kolo pravilno opiše. U tu svrhu najčešće se koristi šematski editor (*Orcad Capture*).

Da bi opisalo kolo za polarizaciju diode, treba proći kroz sledeće korake:

1. U Capture (CIS) Project Manager-u pod *File* menijem izabrati *New Project*.
2. Kao ime projekta izabrati npr. *MyDiode*.
3. Da bi se omogućila simulacija ovog projekta izabrati *Analog or Mixed signal Circuit Wizard*.
4. Kliknuti na *OK*, a zatim na *Finish*. Otvoriće se novi prazan radni prostor za unos šeme.
5. Iz *Place* menija, izabrati *Part* opciju da bi se sledeće komponente unele u šemu kola za polarizaciju diode:
 - R otpornik R1
 - D1N914 dioda D1
 - VSRC naponski izvor V1
6. Izabrati opciju *Place Ground* da bi se u šemu unela i analogna masa
 - 0 analogna masa 0
7. Izabrati iz *Place* menija opciju *Wire* za povezivanje unetih komponenata kao što je to prikazano na slici 6.1.
8. Selektovati *VSRC* komponentu (V1).
9. Dva puta kliknuti na labelu *DC* koja se nalazi pored generatora V1, i podesiti jednosmernu vrednost napona V1 na 5V (*DC 5V*); opcija *show name and value*.
10. Iz *File* menija izabrati opciju *Save*.

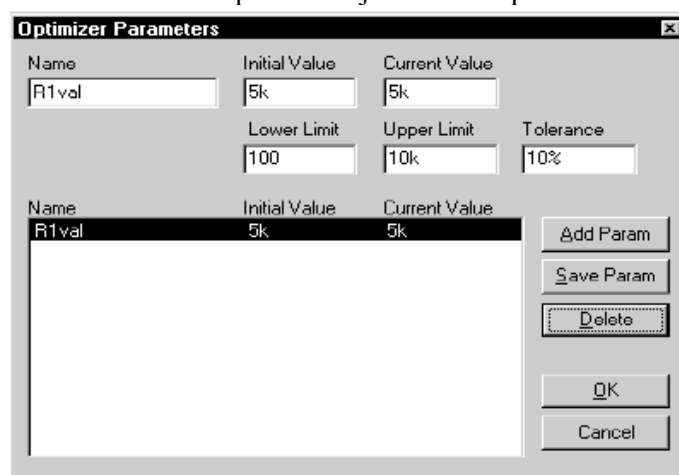
6.2.7 Zadavanje parametara optimizacije

Kada je kolo opisano, potrebno je definisati parametre i ciljeve optimizacije. Da bi se kolo za polarizaciju diode pripremilo za optimizaciju, potrebno je da se vrednost R1 parametrizuje. Potrebno je dakle uraditi sledeće:

1. Iz *Capture PSpice* menija izabrati opciju *Place Optimizer Parameters* da bi se unela instanca *OPTPARAM* (koja se nalazi u *SPECIAL.OLB* biblioteci).
2. Dva puta kliknuti na *OPTPARAM* instancu, a zatim na opciju *User Properties*.
3. Specificirati osobine kao na slici 6.2, tako da se dobije prozor sa slike 6.3.

Name= R1Val
 Initial Value= 5k
 Current Value= 5k
 Lower Limit= 100
 Upper Limit= 10k
 Tolerance = 10%

Slika 6.2: Specificiranje osobina otpornika



Slika 6.3: Prozor u kome se specificiraju osobine otpornika

4. Nakon podešavanja zatvoriti prozor klikom na *OK*.
5. Dva puta kliknuti na labelu 1k koja se nalazi na šemi pored otpornika R1 i uneti umesto toga {R1val}. Time smo vrednost otpornosti R1 parametrizovali.
6. Kliknuti *OK*.

6.2.8 Zadavanje ciljeva optimizacije

Pre nego što se krene sa poboljšanjem karakteristika kola koje se projektuje, mora se najpre odgovoriti na dva pitanja:

- koju karakteristiku kola treba posmatrati i meriti, i
- u kom domenu se ona najbolje ispoljava.

Prilikom postavljanja svakog cilja optimizacije, mora se definisati i tip analize (ona može da bude DC, AC ili vremenska). Program će izvršavati analizu koju izaberemo i u tom domenu će prikazivati vrednosti odziva (izlaznih veličina koje želimo da postignemo optimizacijom). U kolu za polarizaciju diode veličina koju želimo da merimo i posmatramo je struja kroz diodu I(D1), pri čemu je ulazni jednosmerni napon konstantan i iznosi 5V, dok se parametar optimizacije, R1val menja. Dakle, radi se optimizacija jednog parametra kola da bi se ostvario jedan cilj. Da bi se izabrao tip analize za koju optimizujemo kolo potrebno je uraditi sledeće korake:

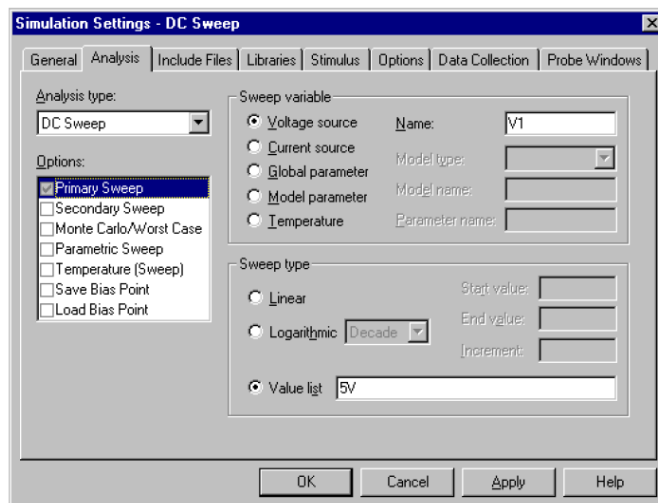
1. Iz *Capture PSpice* menija, izabrati *New Simulation Profile*, u koji se unese *DC sweep* kao ime profila. Pojavljuje se prozor za podešavanje parametara optimizacije.
2. U okviru *Analysis type*, izabrati *DC sweep*.
3. Izabrati parametre simulacije sa slike 6.4, tako da se dobije prozor kao na slici 6.5.

The DC Sweep settings are:
Swept Var Type = Voltage Source
Sweep Type= Value List
Name = V1
Values= 5v

Slika 6.4: Podešavanje parametara simulacije

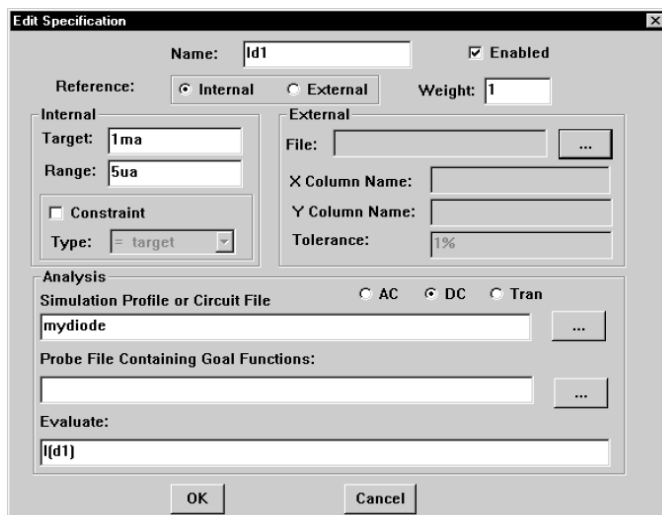
4. Kliknuti na *OK* da bi se sačuvala podešavanja.

Konačna faza pripreme optimizacije je zadavanje ciljeva optimizacije. Do sada su svi potrebni koraci rađeni u *Capture* alatu. Završne korake optimizacije treba raditi u okviru *PSpice Optimizer* alata. Da bi se u našem primeru kao cilj optimizacije izabrala struja kroz diodu I(D1), potrebno je uraditi sledeće korake:



Slika 6.5: Prozor u kome se podešavaju parametri simulacije

1. Iz *Capture PSpice* menija, izabrati *Run Optimizer*, čime se poziva alat *PSpice Optimizer*. Pojavljuje se prozor u kome se, kao parametar optimizacije, pojavljuje R1val, koji je ranije izaban u OPTPARAM instanci.
2. U *Edit* meniju, izabrati opciju *Specifications*.
3. U *Specifications* dialog box-u kliknuti na *Add*.
4. Uneti karakteristike Id1 kao što je to prikazano u prozoru sa slike 6.6:



Name = Id1
 Target = 1ma
 Range = 5ua
 Analysis = DC
 Circuit File = mydiode
 Evaluate = I(d1)

Slika 6.6: Prozor za unos karakteristika Id1

Ovde treba napomenuti, da se u okviru *Simulation Profile or Circuit File* opcije, ime MyDiode ne upisuje, već se bira lokacija projekta koji smo otvorili na početku.

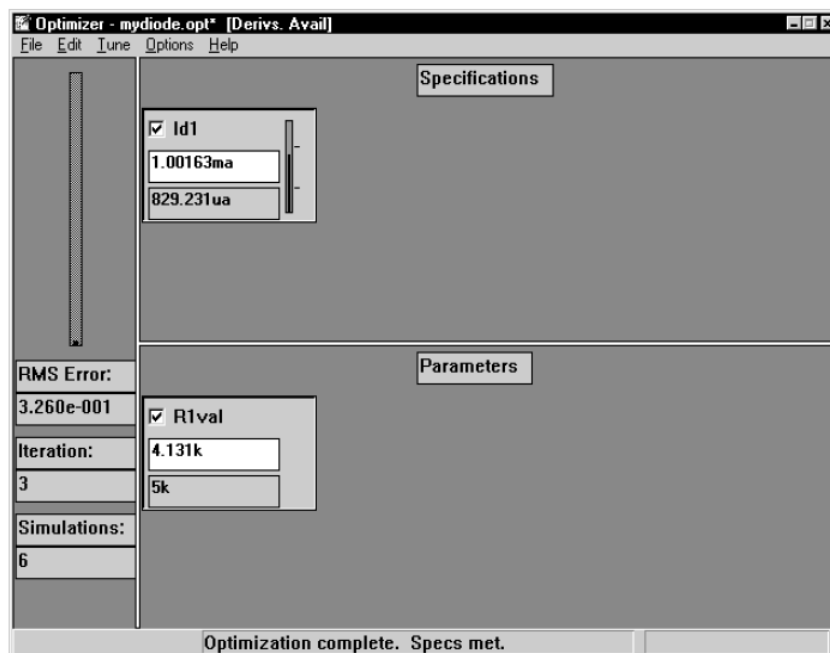
6.2.9 Pokretanje optimizacije

Optimizacija koja se pokreće *PSpice Optimizer* alatom može da bude automatska. Kada se izabere ovakva optimizacija, program automatski računa koeficijente osetljivosti (parcijalne izvode) svih željenih odziva (svake izlazne veličine koja se specificira kao cilj optimizacije), po svim parametrima (veličinama koje su zadate kao parametri optimizacije). Na osnovu znaka i vrednosti koeficijenata osetljivosti, program određuje smer i veličinu promene vrednosti parametara koja dovodi do smanjenja greške između željenog i dobijenog odziva. Nakon što se dobiju nove vrednosti parametara, program ponovo računa izvode i ponavlja proces sve dok se ne postigne jedan od sledećih rezultata:

- Postignuti su ciljevi optimizacije (uspeh)
- Nemoguće je postići dalja poboljšanja performansi (neuspeh)
- Nasilno se prekine rad programa od strane projektanta.

Da bi se pokrenula optimizacija treba izvršiti sledeće korake:

Iz *Tune* menija izabrati opciju *Auto*, a zatim *Start*. Program obavi nekoliko simulacija. Za vrednosti parametara iz svake iteracije, računaju se željene performanse i program ih grafički prikazuje. Računaju se, takođe, i vrednosti posmatrane izlazne veličine (odziva) i prikazuju se u istom prozoru. Nakon 3. iteracije konačno rešenje, u ovom primeru, trebalo bi da konvergira do vrednosti od 4.131k, kao što je to prikazano na slici 6.7:



Slika 6.7: Rezultati optimizacije kola za polarizaciju diode

6.2.10 Dodavanje ograničenja i standardizacija dobijenih parametara

Do sada je prikazan postupak optimizacije jednog parametra da bi se postigao jedan cilj, odnosno dobila željena vrednost jednog odziva. Moguće je, po potrebi, uvoditi ograničenja u postupak optimizacije koja se tiču npr. maksimalne potrošnje, vrednosti elemenata parametara i sl. Nakon zadavanja novih ograničenja, potrebno je ponoviti postupak optimizacije, pri čemu su početna rešenja ona koja smo dobili prethodno opisanim postupkom optimizacije.

Posle uspešno okončane optimizacije, dobija se prozor koji je prikazan na slici 6.7. Međutim, moguće je da izračunata vrednost otpornosti parametra ne odgovara nekoj standardizovanoj vrednosti otpornosti koja se može kupiti. Otpornici koji se kupuju u prodavnicama ne mogu da imaju proizvoljne vrednosti. Da bi se korigovala vrednost otpornosti dobijene nakon optimizacije tako da ona ima standardizovanu vrednost, potrebno je uraditi sledeće:

- Iz *Edit* menija, izabrati opciju *Round Nearest*. Tada se, saglasno sa zadatim specifikacijama tolerancije, automatski koriguje vrednost otpornosti R1, na standardnih 3.9k, kao što je to prikazano na slici 6.8.



Slika 6.8: Standardizovane vrednosti parametra optimizacije

6.2.11 Generisanje izveštaja

Program može da generiše izveštaj o svakom projektu. U izveštaju se, između ostalog, nalaze:

- trenutne vrednosti parametara, ciljevi optimizacije i njena podešavanja
- izračunate vrednosti izvoda izlaznih veličina zajedno sa vrednostima koeficijenata osetljivosti parametara (*Lagrange*-ovih koeficijenata)

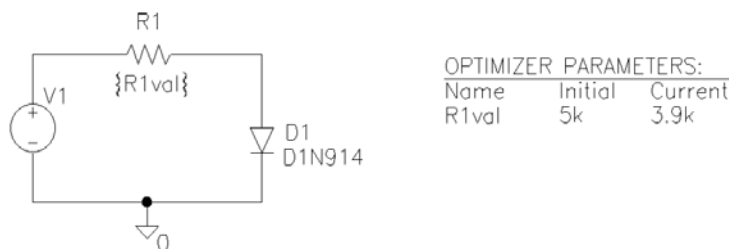
Da bi se generisao izveštaj potrebno je iz *File* menija izabrati opciju *Report*. **PSpice** onda snima konačne rezultate u vidu tekstualnog fajla pod imenom `MyDiode.opt`.

6.2.12 Snimanje rezultata i ažuriranje električne šeme

Projekat se može snimiti zajedno sa dobijenim vrednostima tako što se iz *File* menija izabere opcija *Save*.

Kada se to uradi, potrebno je ažurirati električnu šemu novodobijenim vrednostima parametara i to na sledeći način:

Iz *Edit* menija, izabrati opciju *Update Schematic*. Tada *PSpice Optimizer* šalje poruku *Capture* alatu da ažurira projekat. Time se upisuje nova vrednost parametra od 3.9k u okviru instance OPTPARAM, kao što je prikazano na sl. 6.9.

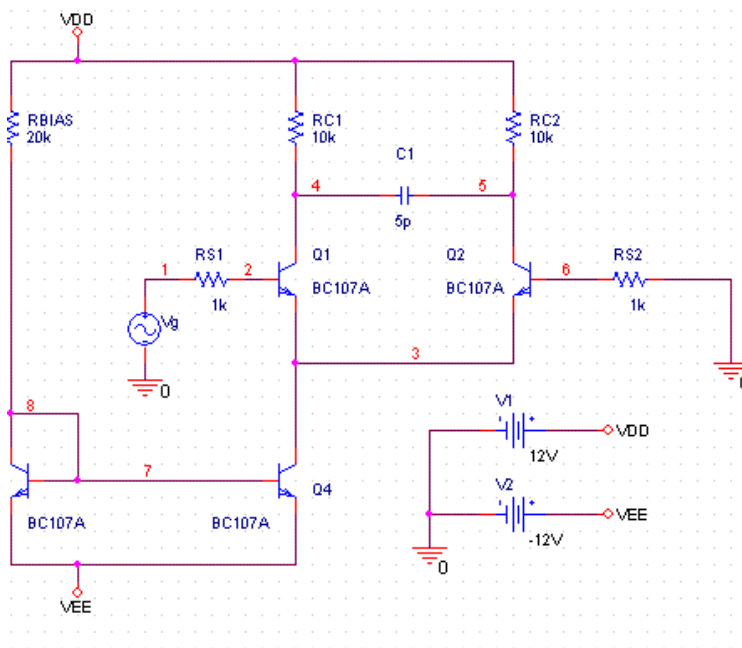


Slika 6.9: Ažurirana električna šema novom vrednošću otpornosti R1

Z A D A T A K

1. Primenom programa **PSpice Optimizer** odrediti vrednost otpornika RBIAS i kapacitivnosti kondenzatora C1 tako da naponsko pojačanje pojačavača sa slike 6.10 iznosi $a=20\log(|V(4,5)/V(1)|)=45\text{dB}$ na frekvenciji $f_{\text{nom}}=10\text{kHz}$, a da gornja granična frekvencija bude $f_{\text{gornje}}=300\text{kHz}$.
2. Prikazati amplitudsku karakteristiku pojačavača dobijenu pre optimizacije. Koristiti početne vrednosti parametara naznačene na slici 6.10.
3. Prikazati rezultate simulacije dobijene posle optimizacije:
 - a. Vrednosti pojačanja na nominalnoj frekvenciji, gornju graničnu frekvenciju
 - b. Vrednosti parametara RBIAS i C1
 - c. Vrednost srednjekvadratne greške i broj iteracija
 - d. Amplitudsku karakteristiku pojačavača

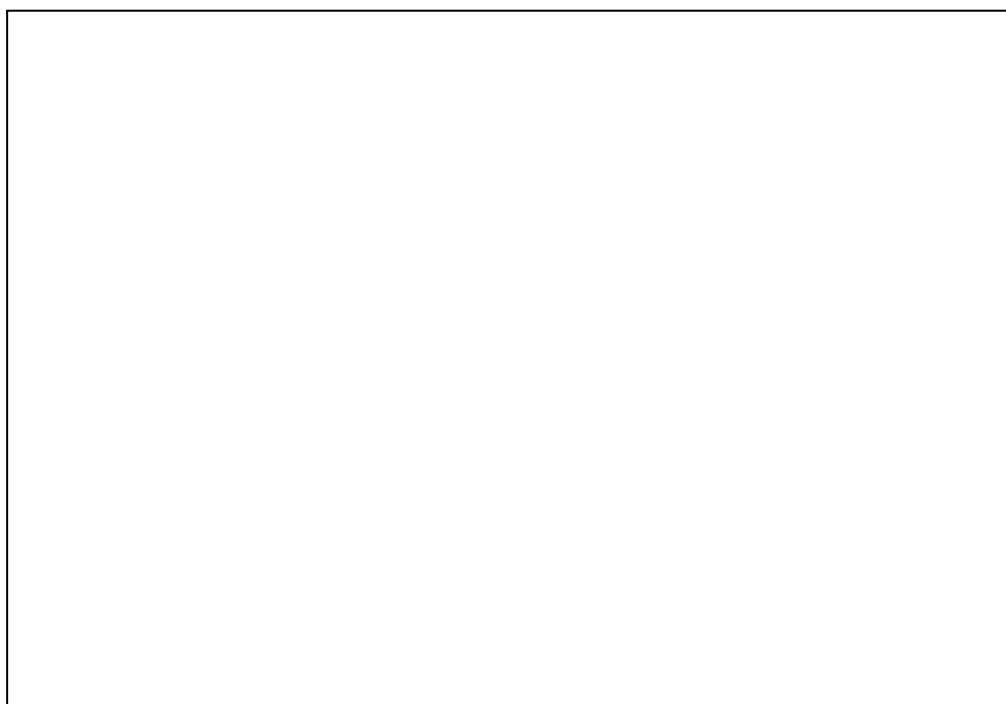
Model tranzistora BC107A nalazi se u biblioteci EBIPOLAR.LIB.



Slika 6.10: Električna šema pojačavača

Rezultati pre optimizacije:

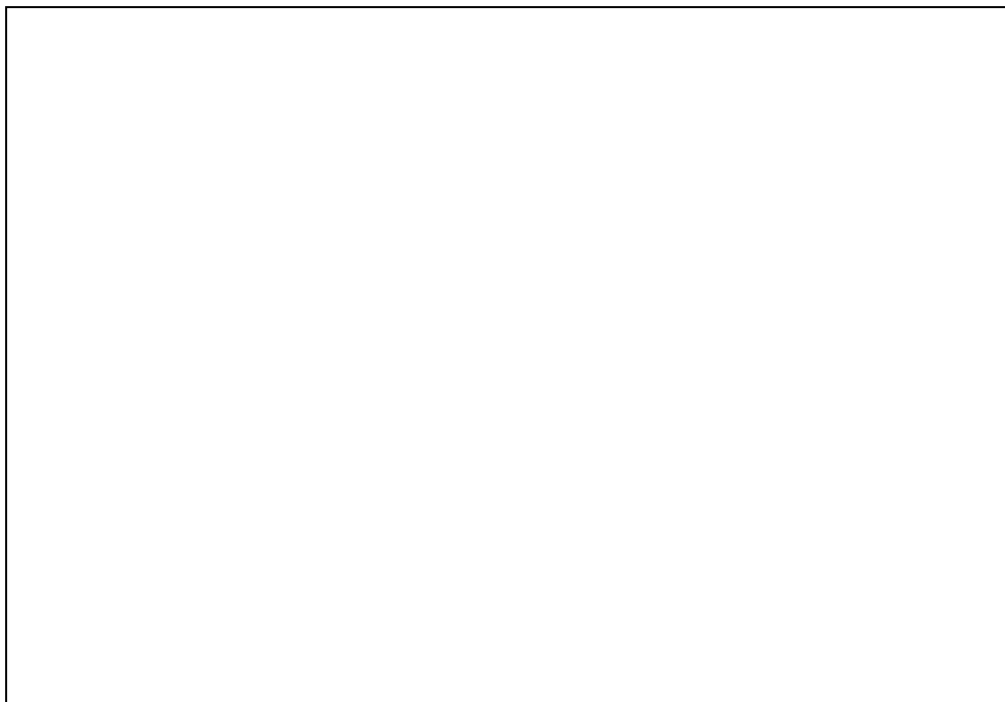
$$\begin{aligned} a(f_{\text{nom}}=10\text{Hz})= \\ f_{\text{gornje}} = \end{aligned}$$



Slika 6.11: Amplitudska karakteristika pojačavača sa slike 6.10 pre optimizacije

Rezultati posle optimizacije:

$$\begin{aligned} a(f_{\text{nom}}=10\text{Hz})= \\ f_{\text{gornje}} = \\ R_{\text{BIAS}} = \\ C_1 = \\ \text{Greška} = \\ \text{Broj iteracija} = \end{aligned}$$



Slika 6.12: Amplitudska karakteristika pojačavača sa slike 6.10 posle optimizacije

Datum : _____

Student : _____

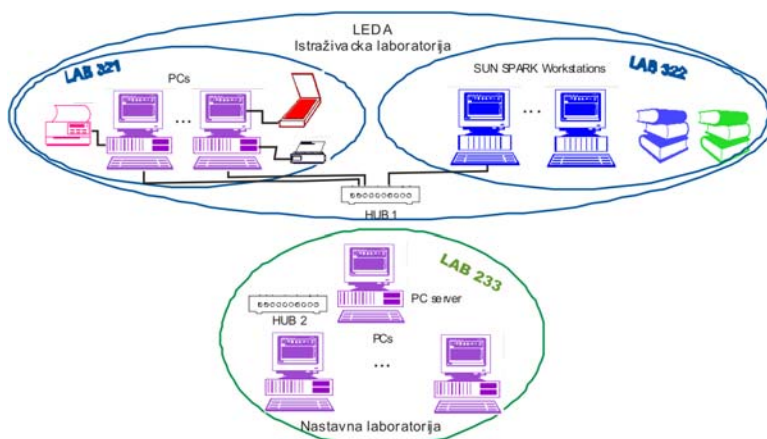
Overa : _____

II Praktikum laboratorijskih vežbi iz predmeta

Projektovanje digitalnih integrisanih kola

II.1 Uvod

U drugom delu praktikuma biće opisane vežbe koje studenti rade iz predmeta Projektovanje digitalnih integrisanih kola (PDIK) i Alati za projektovanje (AZP). U okviru PDIK kursa očekuje se da studenti samostalno kreiraju jednu logičku ćeliju, od električne šeme do crtanja lejaute. Kurs AZP ima za cilj da obučiti studente da projektuju ASIC integrisana kola, odnosno da koriste preprojektovane strukture u obliku standardnih ćelija. Studentima su na raspolaganju alati za projektovanje integrisanih kola koje distribuira *Mentor Graphics (MGC)*. Ovi alati rade pod *Unix* operativnim sistemom, tako da će upoznavanje sa tim okruženjem početi opisom minimalnog seta *Unix* komandi. Osnovni program instaliran je na Unix serveru SUN Spark Workstation koji se fizički nalazi u prostoriji 322. On je preko HUB 1 vezan za PC server u nastavnoj laboratoriji 233 kao što ilustruje slika II.1.



Slika II.1: Izgled lokalne računarske mreže u Laboratoriji 233

U okviru Laboratorije 233 postoji lokalna mreža između PC servera i PC računara na kojima rade studenti. Da bi ovi računari mogli da služe kao terminali Unix serveru, na njima su instalirani programi za emulaciju Unix operativnog sistema iz Windows okruženja. Zato je, pre početka rada neophodno da se pokrene emulator *xwin*.

Nakon toga dobija se na PC-u Unix okruženje u kome se nastavlja rad. Posle uspešne prijave, ulazi se u radni direktorijum *PDIKgg-gg* u kome *gg-gg* označava poslednje dve cifre školske godine u kojoj se vežba obavlja. Ovaj direktorijum zvaće se nadalje *osnovni* ili *root* direktorijum.

Potrebno je da svaki student u njemu kreira svoj lični radni prostor (direktorijum) koji će imenovati *ime_prezime_index*, što se čini ukucavanjem Unix naredbe:

```
mkdir ime_prezime_index
```

Zatim treba preći na taj direktorijum naredbom

```
cd ime_prezime_index
```

Kada se jednom kreira korisnički radni direktorijum, nadalje će ceo projekat da se odvija u njemu. Zato je neophodno da se u njemu kreira odgovarajuće *radno okruženje za projektovanje*. Treba imati u vidu da je podešavanje radnog okruženja:

- Različito za različite stilove projektovanja,
- Različito za pojedine alate,
- Različito za različite tehnologije.

Projektovanje integrisanih kola obavlja se u nekoj od raspoloživih tehnologija. Zato je neophodno da se za korisničko radno okruženje vežu sve komponente (fajlovi) koje se odnose na izabranu tehnologiju. Podaci o raspoloživim tehnologijama nalaze se u okviru *opšte biblioteke* negde na serveru. Ona sadrži skupove podataka o broju i tipovima maski neophodnih za realizaciju projekta u određenoj tehnologiji kao i o pravilima projektovanja. Pored toga sadrže podatke o pretprojektovanim strukturama (standardnim i makro-ćelijama) u različitim formatima koje prepoznaju pojedini alati za projektovanje. Jedan projekat zahteva opis u više formata. U prvom delu ovog praktikuma studenti su imali prilike da se sretu sa formatima vezanim za električnu šemu (.sch) i za net listu (.cir) jednog istog projekta. Kada se radi o projektovanju integrisanih kola postoji potreba za memorisanjem podataka u mnogo širem setu formata. Bez želje da zbunjujemo čitaoca ovde ćemo navesti samo neke od njih.

LEF (*Library Exchange Format*) predstavlja ASCII fajl koji sadrži opis svih ćelija za određenu tehnologiju sa nazivima i rasporedom pinova (uključujući VDD i VSS). Ove podatke daje vlasnik biblioteke standardnih ćelija, odnosno tehnologije, ali i korisnik može da ih generiše posebnim alatima kada kreira sopstvenu ćeliju.

TLF (*Timing Library Format*) ASCII fajl koji sadrži opis vremenskih parametara (kašnjenja) svih ćelija za određenu tehnologiju; uključuje sve pinove kao i LEF sem VDD, VSS i prolaz (pinovi kroz koje ne postoji kašnjenje).

CTLF (*Compiled Timing Library Format*) predstavlja kompilovani TLF fajl tako da je kompaktniji – podaci su binarnog a ne ASCII tipa.

ALF fajlovi (*AMBIT Library Format*) alternativni oblik zadavanja podataka o kašnjenjima. Sadrži dinamičke osobine standardnih ćelija.

LUT (*Layer Usages Table*) tabela predstavlja ASCII fajl sa podacima o pojedinim slojevima (maskama): procentualna zastupljenost pojedinih slojeva metala, otpornost via, prosečan broj via na vezi,...

GCF (*General Constraints Format*) sadrži informacije o različitim tipovima ograničenja vezanim za rad određene ćelije (naponi, temperature, pritisak, ograničenja za ulazne, izlazne i bidirekzione signale u odnosu na zadato vreme uspostavljanja ulaznih, izlaznih i signala takta, pobudna moć (fan-out), podaci o parazitnim opterećenjima na izlazu). Sadrži ukazivač na CTLF fajlove.

Pre početka svakog projekta korisnik u radnom okruženju definiše

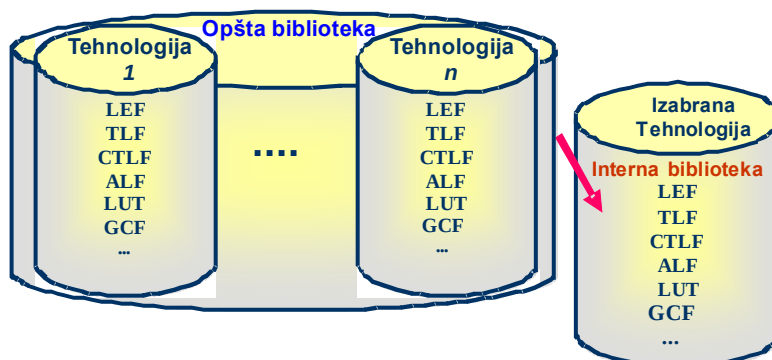
- internu biblioteku i
- korisničku datoteku

• Interna biblioteka sadrži fajlove sa podacima vezanim za tehnologiju kao što su pravila projektovanja, izgled i osobine pretprojektovanih struktura (standardnih ćelija i/ili makroćelija) koje se učitavaju iz opšte biblioteke. Ova aktivnost obavlja se automatski posredstvom alata za projektovanje a svodi se na selektovanje podataka o željenoj tehnologiji iz padajućih menija. Sadržaj menija zavisi od sadržaja opšte biblioteke a izbor se svodi na markiranje neophodnih fajlova. Program automatski formira kopiju ili link u okviru interne tehnološke biblioteke. Na ovaj način isti sadržaj iz opštih biblioteka mogu da ‘pozajmljuju’ različiti projekti.

Izbor fajlova/formata koji se učitavaju zavisi od izbora alata i stila projektovanja. Treba napomenuti da bibliotečke datoteke često u nazivu sadrže oznaku **lib** tako da se mogu pretraživati i po tom kriterijumu. Pored toga, najčešće

se korisniku nude samo biblioteke koje su relevantne za izabrani stil projektovanja i tehnologiju, tako da je smanjena mogućnost pogrešnog izbora.

Slika II.2 ilustruje pridruživanje fajlova iz opšte u internu biblioteku.



Slika II.2: Veza između opšte i interne tehnološke biblioteke

- Korisnička datoteka (*Design Database*) predstavlja bazu u koju će se smeštati (među)rezultati tokom realizacije konkretnog projekta. Ona najčešće sadrži fajlove sledećih tipova:

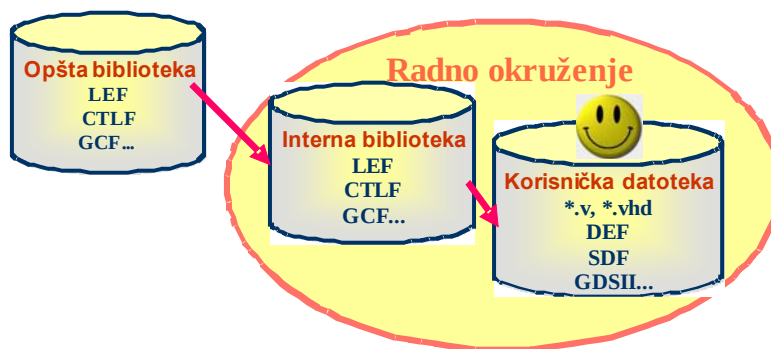
HDL fajlovi (fajlovi na nekom od jezika za opis hardvera *Hardware Description Language*): U ovom obliku opisuje se projekt na jeziku koji razumeju čovek i mašina. Radi se o opisu u formatima **.vhd* (VHDL jezik) i/ili **.v* (Verilog jezik) svih ćelija, kako novoformiranih (kreira ih projektant) tako i za standardne ćelije u određenoj tehnologiji (daje ih vlasnik tehnologije) ili za opis makroćelija (daje ih vlasnik IP).

DEF (*Design Exchange Format*) netlista za tekući projekat predstavlja isto što i LEF za bibliotečke ćelije. Sadrži netlistu svih ćelija sa spiskom međusobnih veza, podatke o ugaonim ćelijama, razvođenju napajanja kao i podatke o fizičkim ograničenjima (via, net, ...).

GDSII (*Graphic Data System II*) je binarni fajl o izgledu lejauta projektovanog integrisanog kola. Praktično je ovaj format postao industrijski standard na osnovu koga se prave maske neophodne za proizvodnju integrisanih kola. Zato on, najčešće predstavlja završni oblik projekta. Više podataka o ovom formatu čitalac može naći u <http://www.rulabinsky.com/cavd/text/chapc.html>.

Sve fajlove iz korisničke datoteke projektant kreira tokom izrade projekta.

Slika II.3 ilustruje vezu između opšte, interne i korisničke baze.



Slika II.3: Korisnik ima prava da modifikuje samo fajlove iz korisničke datoteke

Osim kreiranja ličnog radnog prostora direktno iz operativnog sistema, što je bilo navedeno na početku ovog uvodnog odeljka, ostale aktivnosti vezane za pridruživanje tehnoloških fajlova i podataka o standardnim ćelijama internoj biblioteci i korisničkoj datoteci obavljaju se u alatima. Konkretno u setu Mentor Graphics alata taj posao obavlja se u programu *IC Studio*. Naravno, zavisno od konkretnog zadatka pozivaju se i druge komponente (alati) sa MGC platforme. Način pozivanja pojedinih alata i njihova namena opisani su u vežbama koje slede i to:

Vežba 7. opisuje primenu alata *Design Architect-IC* za crtanje električne šeme.

Vežba 8. opisuje poziv programa za simulaciju elektronskih kola iz alata *IC Studio*.

Vežba 9. opisuje postupak crtanja lejauta korišćenjem alata *Design Architect-IC*.

Vežba 10. opisuje tok projektovanja ASIC kola na bazi standardnih ćelija upotrebom *ADK* alata iz koga se pozivaju sledeće komponente:

- *Leonardo Spectrum* za sintezu
- *Design Architect-IC* za verifikaciju
- *IC Station* za fizičko projektovanje.

Vežba broj 7

Crtanje električne šeme invertora korišćenjem alata *Design Architect-IC*

7.1 Cilj vežbe

Proučiti upotrebu programa *Design Architect-IC* za crtanje električne šeme integrisanih kola.

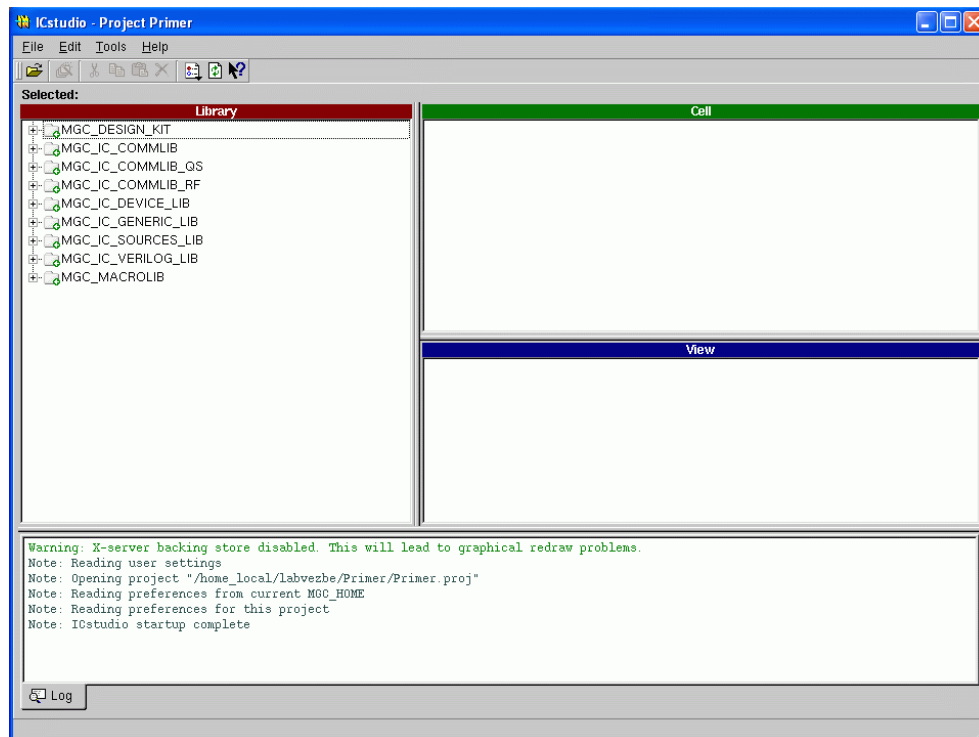
7.2 Teorijska postavka vežbe

Električna šema predstavlja uobičajeni način opisa strukture elektronskih kola. Zato, najčešće, projektovanje elektronskog kola počinje crtanjem šeme. Naravno, inicijalni opis, naročito na višem nivou apstrakcije može da se zada u nekom od jezika za opis hardvera. Ipak, grafička interpretacija predstavlja najpregledniji opis strukture kola, tako da se iz VHDL ili Verilog kôda uvek prelazi i na šematski prikaz. Zato je prva vežba u ovom praktikumu i bila posvećena crtanju električne šeme.

U okviru sledećeg seta laboratorijskih vežbi studenti će se upoznati sa alatima za projektovanje integrisanih kola koje distribuira *Mentor Graphics*. U tom kontekstu ovo poglavlje otvaramo upoznavanjem sa opcijom crtanja električne šeme primenom alata *ICstudio*.

7.3 Pokretanje alata *ICstudio*

Da bi se pokrenuo *ICstudio*, u komandnoj liniji treba ukucati: `icstudio&`. Ovom komandom otvara se prozor koji je prikazan na slici 7.1.



Slika 7.1: Izgled prozora kada je pokrenut *ICstudio*

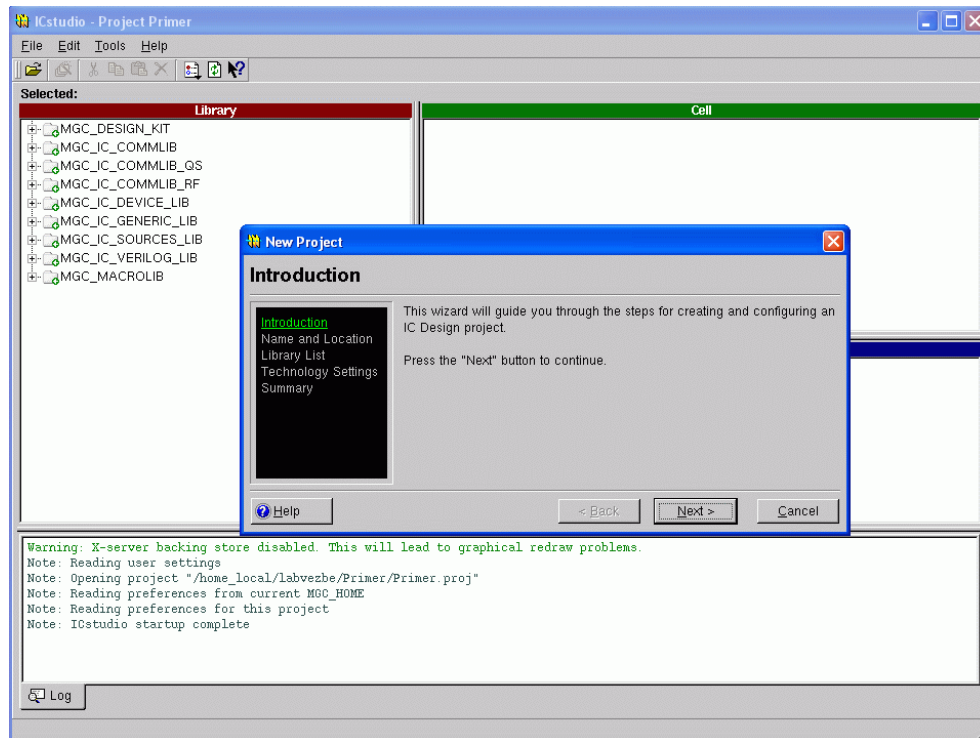
7.4 Kreiranje projekta

Da bi se započeo projekat, treba uraditi tri sledeća koraka:

1. U *ICstudio* prozoru:

- kliknuti **File** -> **New** -> **Project** da bi se započeo novi projekat.
- kliknuti **Next** u **New project** pop-up prozoru.

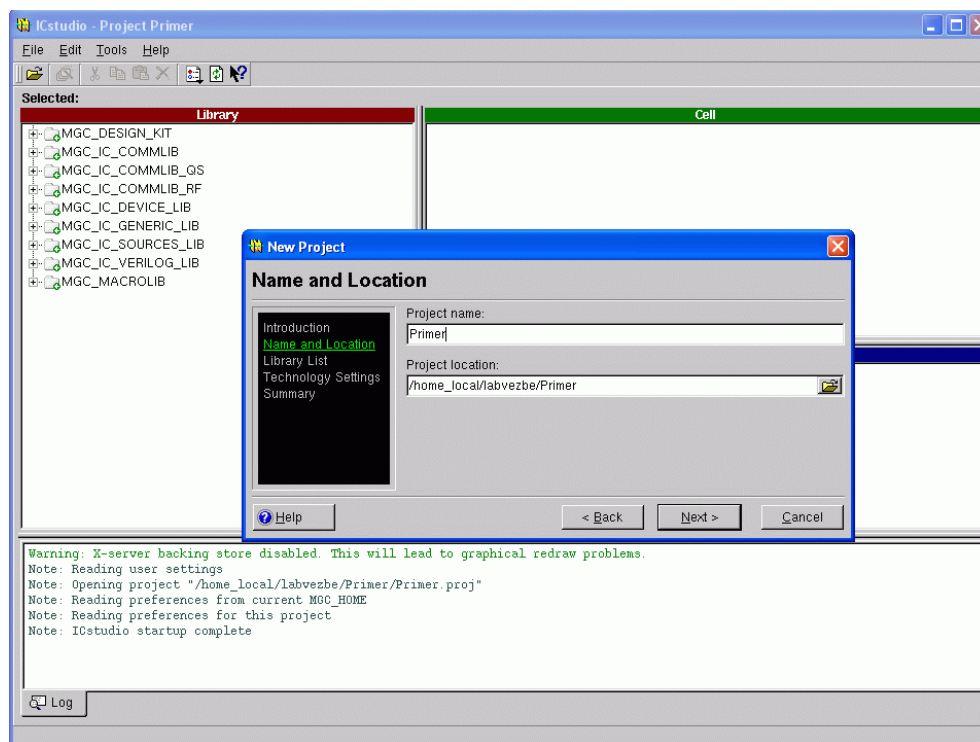
Prozor izgleda kao na slici 7.2.



Slika 7.2: Izgled prozora kada se počinje novi projekat

uneti ime projekta (**Project name**), npr. **Primer**, i lokaciju projekta (**Project location**), tj. ime direktorijuma koji je kreiran da bi se tu sačuvao projekat. Otvoriće se prozor kao na slici 7.3.

Zatim kliknuti **Next** u **New project** pop-up prozoru.

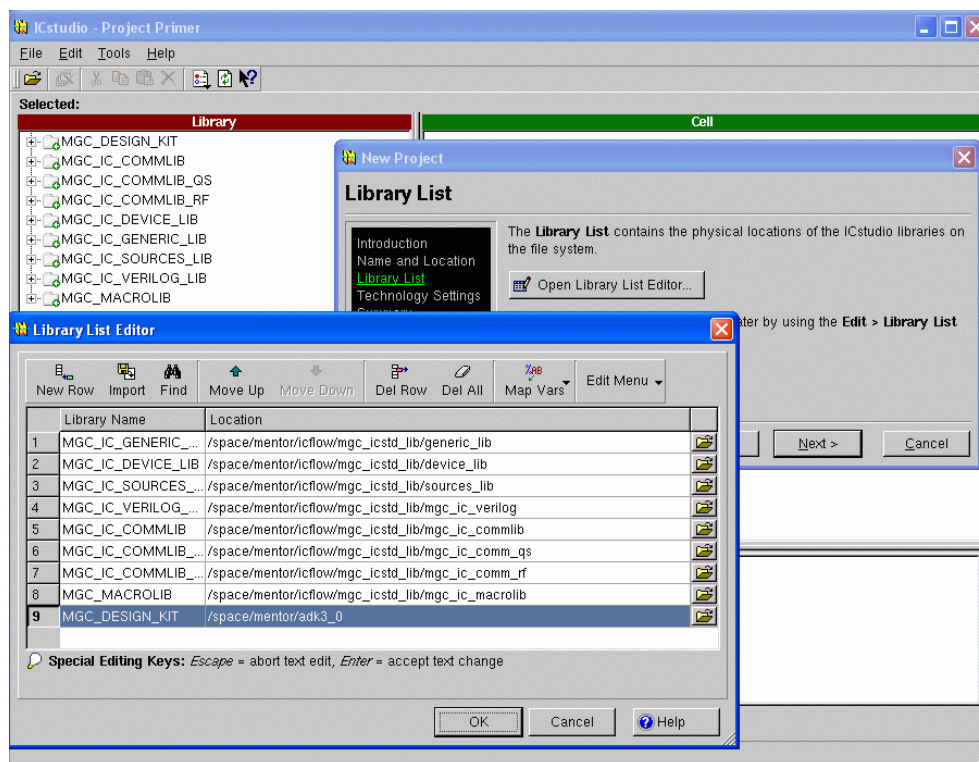


Slika 7.3: Izgled prozora New Project kada se unosi ime i lokacija projekta

Kada se pojavi sledeći prozor, treba kliknuti na dugme **Open Library List Editor**. Da bi se dodale biblioteke standardnih ćelija potrebne za projekat, kliknuti **Edit Menu>Add Standard MGC Libraries**, čime se sledeće biblioteke automatski pridružuju projektu:

```
/space/mentor/icflow/mgc_icstd_lib/generic_lib
/space/mentor/icflow/mgc_icstd_lib/device_lib
/space/mentor/icflow/mgc_icstd_lib/sources_lib
/space/mentor/icflow/mgc_icstd_lib/mgc_ic_verilog
/space/mentor/icflow/mgc_icstd_lib/mgc_ic_commlib
/space/mentor/icflow/mgc_icstd_lib/mgc_ic_comm_qs
/space/mentor/icflow/mgc_icstd_lib/mgc_ic_comm_rf
/space/mentor/icflow/mgc_icstd_lib/mgc_ic_macrolib
```

U istom prozoru kliknuti **Edit Menu>Add MGC Design Kit**, pa specificirati putanju za *MGC Design Kit* kao: `/space/mentor/adk3_0/`. Prozor **Library List Editor** izgleda kao na slici 7.4. Kliknuti **OK**, a zatim **Next** u prozoru **New project**.



Slika 7.4: Izgled prozora **Library List Editor** iz koga se biblioteke standardnih ćelija priključuju projektu

U sledećem koraku treba specificirati tehnološke fajlove.

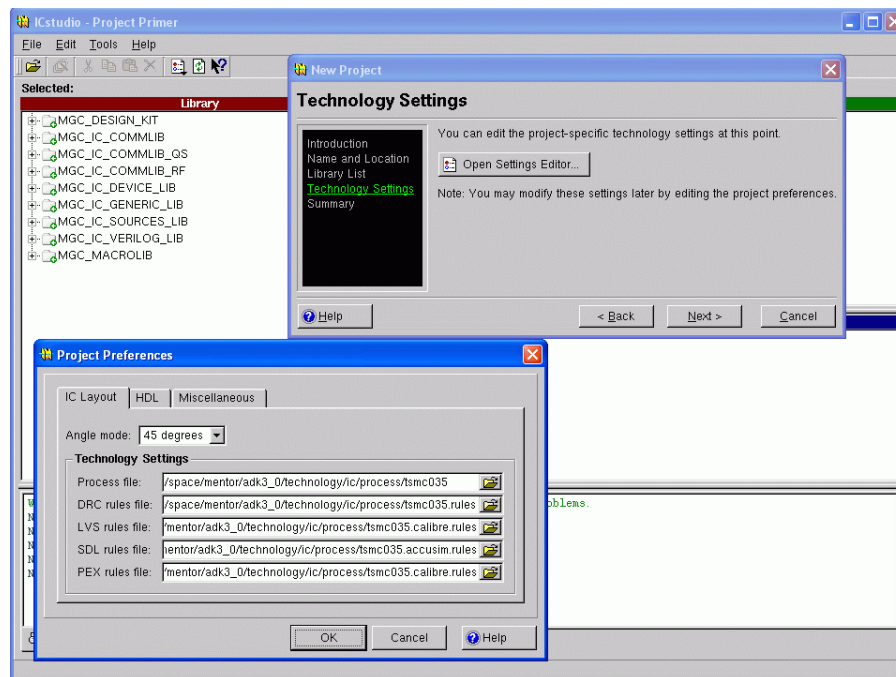
- Kliknuti dugme **Open Settings Editor**. Otvara se prozor **Project Preferences**.

- Da bi se učitao tehnološki **Process file** (slika 7.5) za tehnologiju TSMC035 u kojoj će se realizovati projekat, treba izabrati putanju `/space/mentor/adk3/technology/ic/process/tsmc035`.

- Zatim treba učitati fajlove sa pravilima projektovanja za datu tehnologiju. Pravila se učitavaju prateći putanju: `/space/mentor/adk3/technology/ic/process`. Potrebno je izabrati sledeće fajlove:

- o fajl sa DRC (*design rule check*) pravilima: `tsmc035.rules`
- o fajl sa LVS (*layout versus schematic*) pravilima: `tsmc035.calibre.rules`
- o fajl sa SDL (*schematic-driven layout*) pravilima: `tsmc035.accusim.rules`
- o fajl sa PEX (*parasitic extraction*) pravilima: `tsmc035.calibre.rules`

Izgled prozora prikazan je na slici 7.5. Zatim kliknuti **OK** u dijalogu prozoru. Kliknuti **Next** u **New project** prozoru, pregledati **Summary** da biste proverili da li su svi podaci ispravni. Na kraju kliknuti **Finish**.



Slika 7.5: Izgled prozora **Project Preferences** u kome se specificiraju tehnološki fajlovi

7.5 Kreiranje biblioteke i ćelija

Da bi kreirali biblioteku sopstvenih ćelija

- kliknuti **New>Library**. Otvara se **Create Library** dijalog.
- uneti ime biblioteke koju želimo da kreiramo, na primer

Nova_biblioteka.

- kliknuti **OK**.

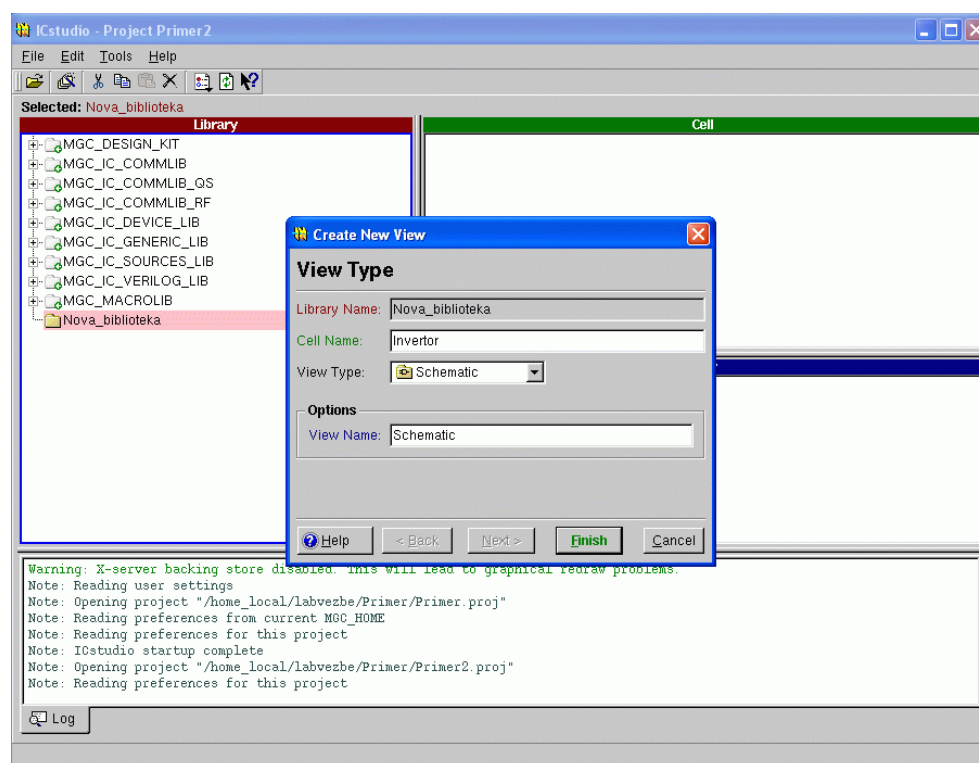
Da bi kreirali novu ćeliju u šematskom editoru:

- izaberemo biblioteku u kojoj želimo da kreiramo ćeliju (npr.

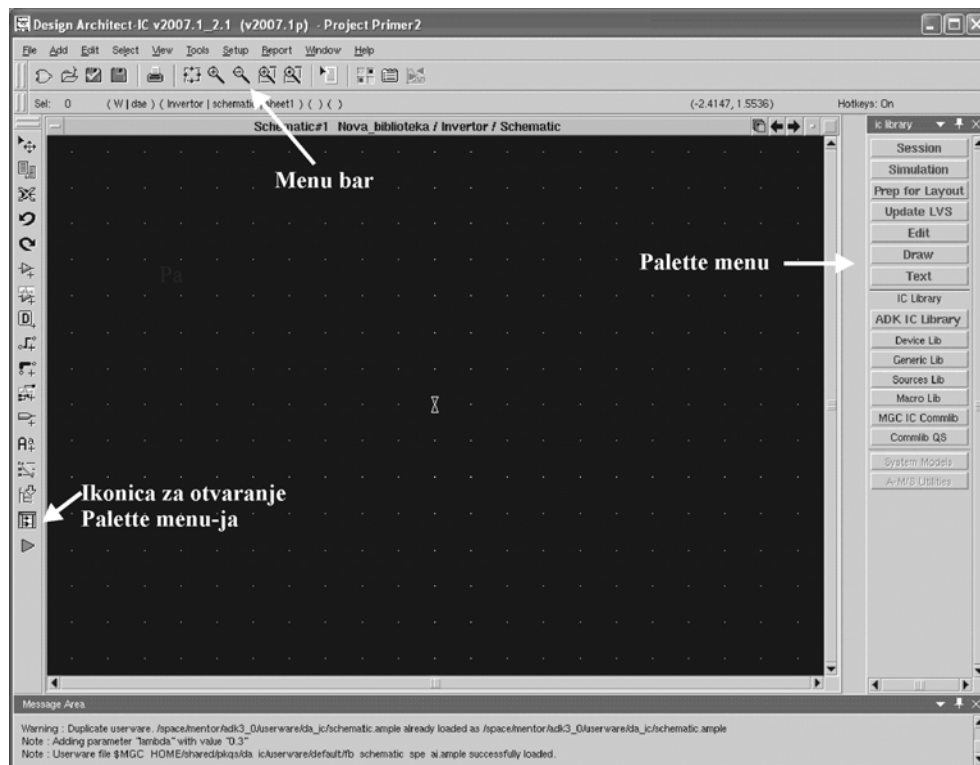
Nova_biblioteka)

- kliknuti **File>New>Cell View**. Pojavljuje se prozor **Create New View** (kao na slici 7.6)
- uneti ime ćelije (npr. **Invertor**)
- da bi se počelo sa crtanjem električne šeme, treba specificirati **View Type** kao **Schematic** i kliknuti **Finish**.

Posle ovog koraka, automatski se pokreće program *Design Architect-IC*. Pojavljuje se prozor u kome se crta šema našeg projekta, kao na slici 7.7. Kliknuti na ikonicu sa leve strane da bi se otvorio **Palette menu** sa desne strane (strelicom označeno na slici 7.7).



Slika 7.6: Izgled prozora kada se kreira nova ćelija



Slika 7.7: Izgled prostora u koji se unosi šema

7.6 Crtanje električne šeme

Opisaćemo postupak unošenja električne šeme CMOS invertora.

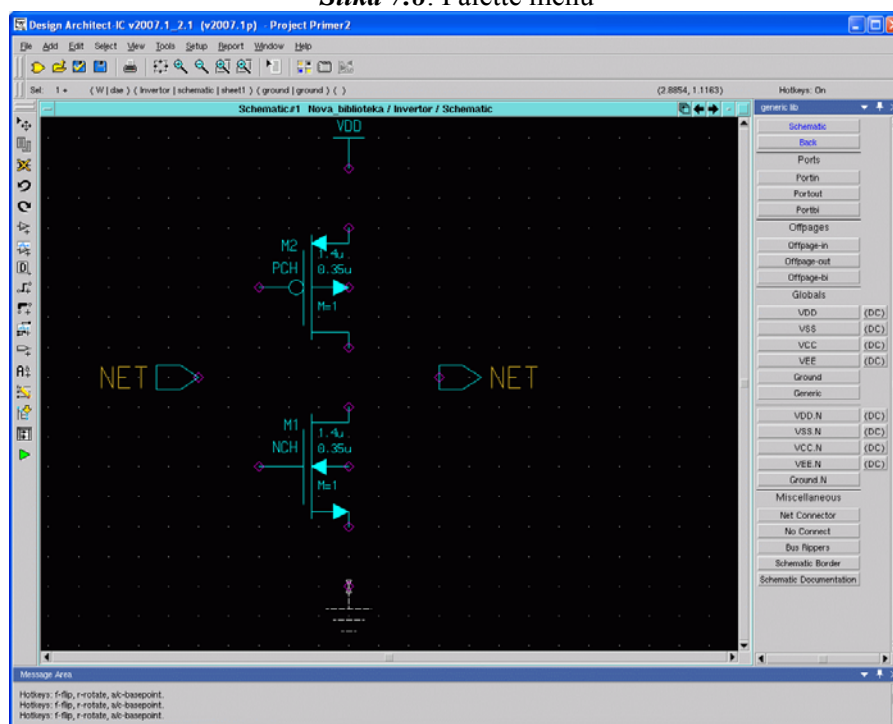
Iz biblioteke komponenata biramo delove koji su nam potrebni za crtanje šeme invertora.

Da bismo postavili tranzistore na radnu površinu

- na **palette menu**-ju (slika 7.8), kliknuti na **device lib**
- izabrati **NMOS (4-pin)** i postaviti na radnu površinu klikom na željenu lokaciju. Na sličan način, izabrati **PMOS (4-pin)** i postaviti ga na radnu površinu.
- kliknuti **BACK** da bismo se vratili na **IC library** paletu i izabrati biblioteku u kojoj se nalaze opšte komponente a koja se zove **Generic lib**.
- Iz **Generic lib**-a izabrati **Portin** za ulazni port, **Portout**, za izlazni port, **VDD** za napon napajanja i **Ground** za masu i postaviti ih na radnu površinu, kao na slici 7.9.



Slika 7.8: Palette menu



Slika 7.9: Izgled prozora nakon unošenja komponenti CMOS invertora

Za povezivanje komponenti treba koristiti taster **F3** na tastaturi:

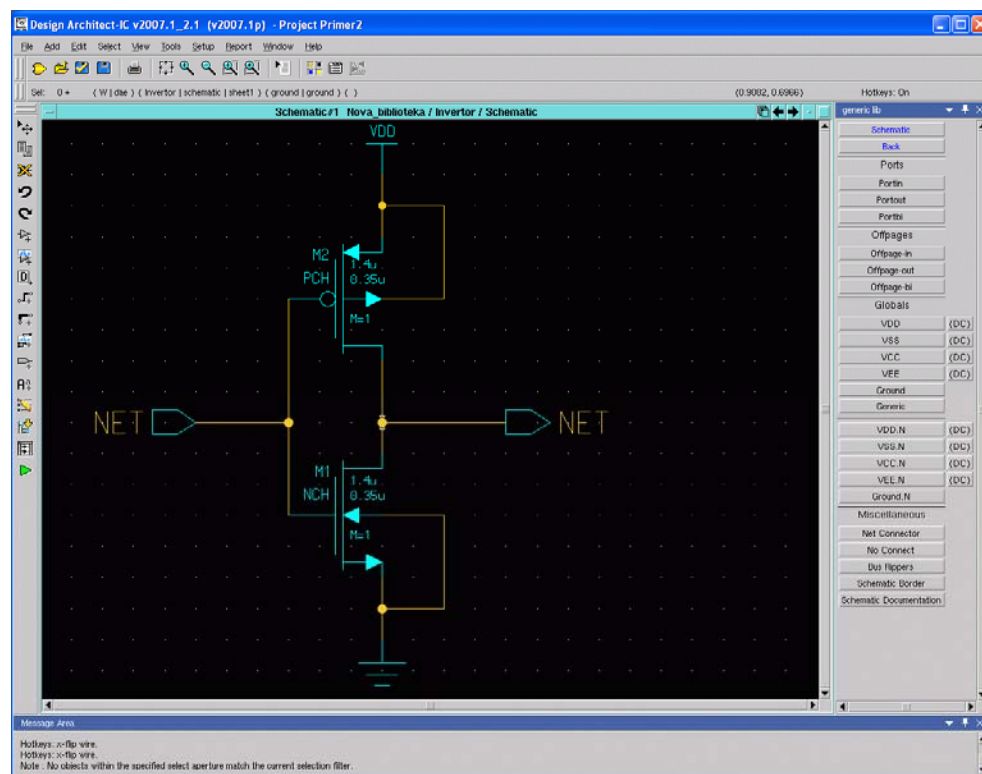
- pritiskom na **F3** aktivira se komanda za povezivanje
- pritiskom na **Esc** deaktivira se komanda za povezivanje

Alternativno, povezivanje se može obaviti korišćenjem **schematic edit palette** ili sa menu bar-a klikom na **ADD -> WIRE**.

Povezati pojedine komponente prema šemi kola sa slike 7.10

Da bi se promenio tekst ili vrednost koji definišu neku komponentu, treba postaviti kursor na taj tekst/vrednost i pritisnuti **SHIFT+F7**. (Nije potrebno selektovati tekst/vrednost već samo na to mesto postaviti kursor.)

U dnu ekrana sa leve strane se pojavljuje konzola sa trenutnim tekstom/vrednošću u pravougaoniku **New Value box**. Uneti novo ime (new name) i kliknuti **OK**.

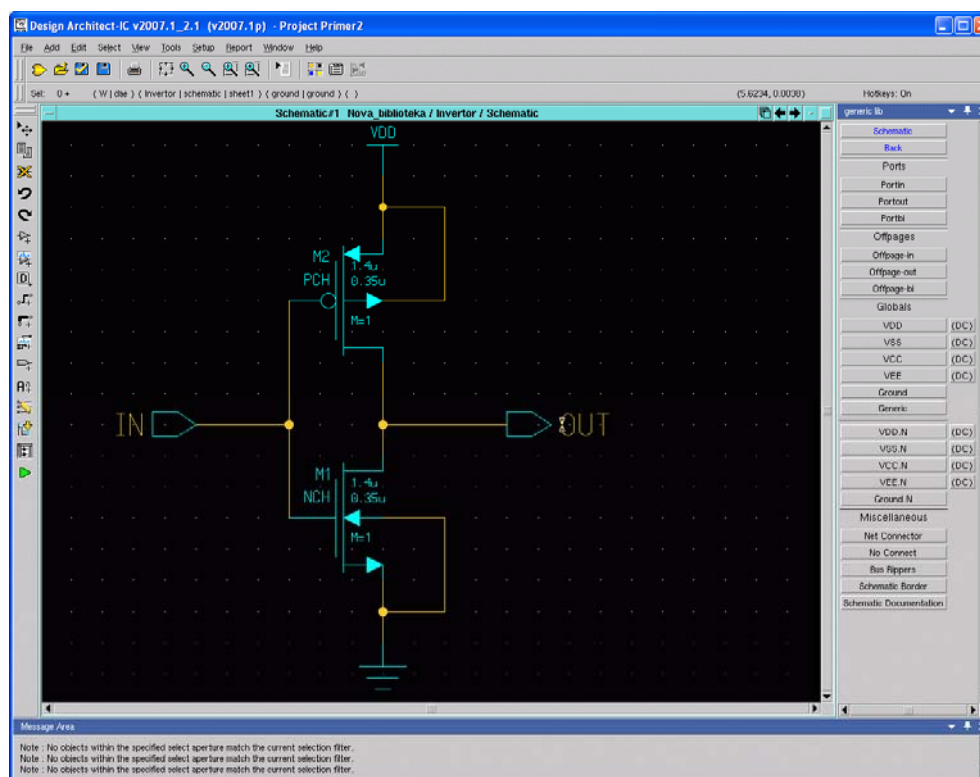


Slika 7.10: Izgled prozora nakon povezivanja komponenti CMOS invertora

Na primer, da bi se promenila labela **NET** ulaznog porta sa slike 7.10. u labelu **IN** potrebno je:

- postaviti kursor na tekst **NET** blizu simbola ulaznog porta i pritisnuti **SHIFT+F7**.
- u dnu ekrana sa leve strane se pojavljuje konzola. ukucati **IN** za **New Value** i kliknuti **OK**.

Na sličan način promeniti labelu izlaznog porta iz **NET** u **OUT**, tako da šema izgleda kao na slici 7.11.



Slika 7.11: Izgled prozora nakon promene imena portova

Na ovaj način se mogu menjati vrednosti veličina tranzistora, ili neke tekstualne vrednosti, a ako želimo da promenimo više parametara neke komponente istovremeno (npr. NMOS-a), treba postupiti na sledeći način:

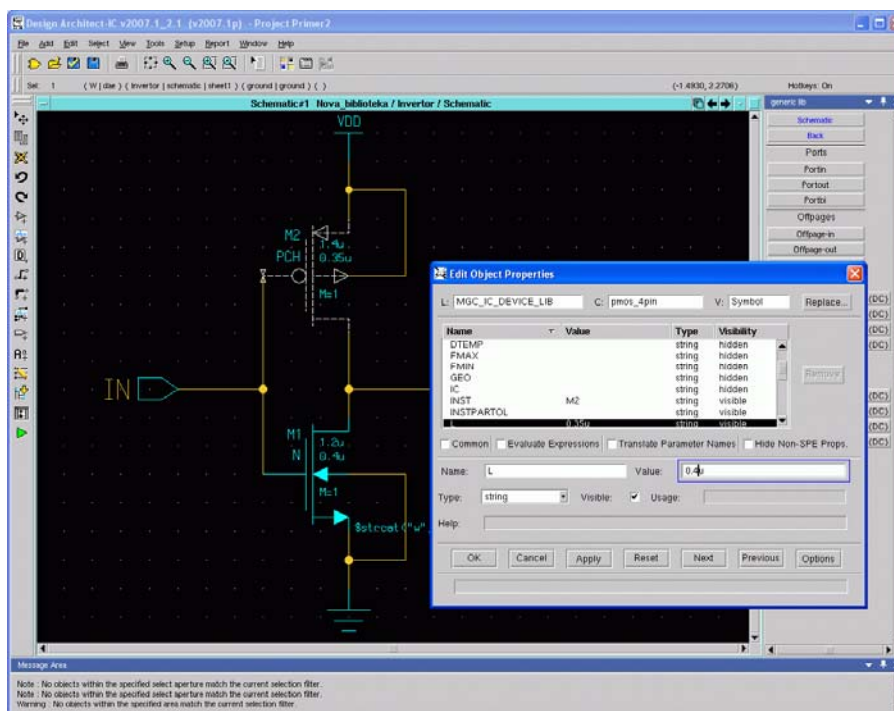
• Selektovati komponentu NMOS tranzistora, kliknuti desnim klikom miša, i izabrati **Properties**→**Edit**. Pojavljuje se novi prozor (slika 7.12), i u tom dijalogu treba promeniti sledeće osobine (posle svake promene obavezno kliknuti *Apply*):

```

INST:           M1
L:              0.4u
W:              1.2u
ASIM_MODEL:     N
PHY_COMP:       mn
INSTPARTOL:
$strcat("w", <space>w, <space>"l", <space>L)

```

Promeniti tip za INSTPARTOL u "expression". Takođe, voditi računa o tačnom unošenju izraza za INSTPARTOL, naročito o blanko znakovima (<space>). Ukoliko ne postoji osobina INSTPARTOL, ili neka druga, uneti je u poslednji prazan red u tabeli.

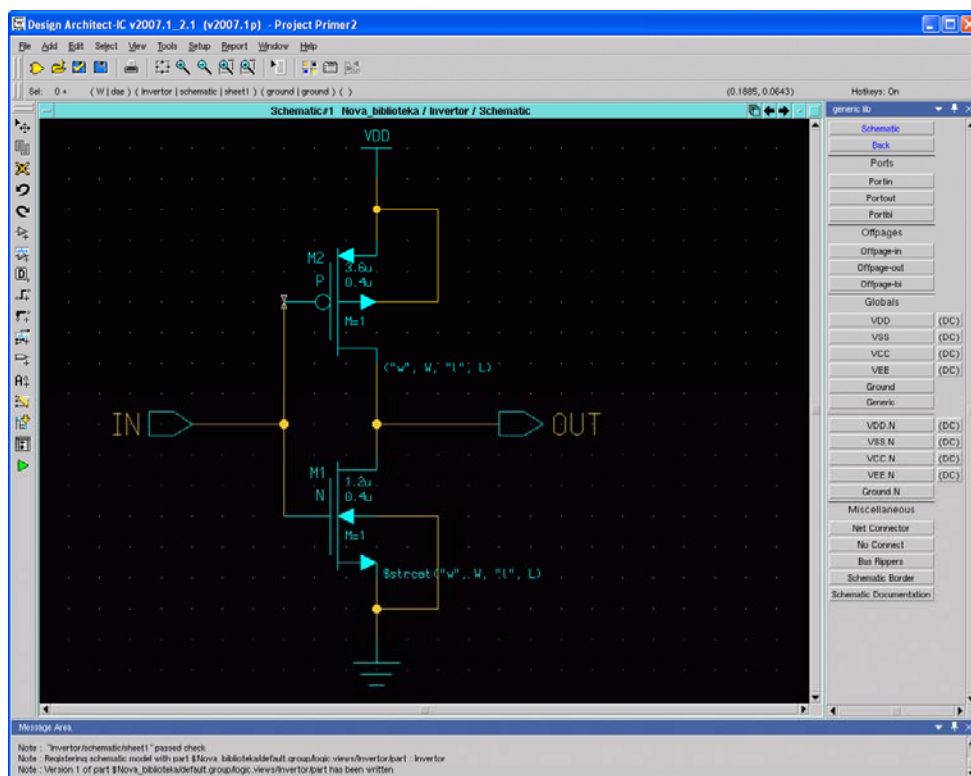


Slika 7.12: Izgled prozora **Edit Object Properties** pri promeni nekih osobina tranzistora

Na sličan način treba izabrati PMOS tranzistor i promeniti sledeće osobine:

INST: **M2**
 L: **0.4u**
 W: **3.6u**
 ASIM_MODEL: **P**
 PHY_COMP: **mp**
 INSTPARTOL:
\$strcat("w", <space>w, <space>"l", <space>L)
 Promeniti tip za INSTPARTOL u "expression".

Konačna šema izgledaće kao na slici 7.13.



Slika 7.13: Konačna šema CMOS invertora

Kada je šema gotova, kliknuti **Check&Save** sa **schematic edit** palete ili iz menija na vrhu ekrana da bi projekat bio proveren i snimljen.

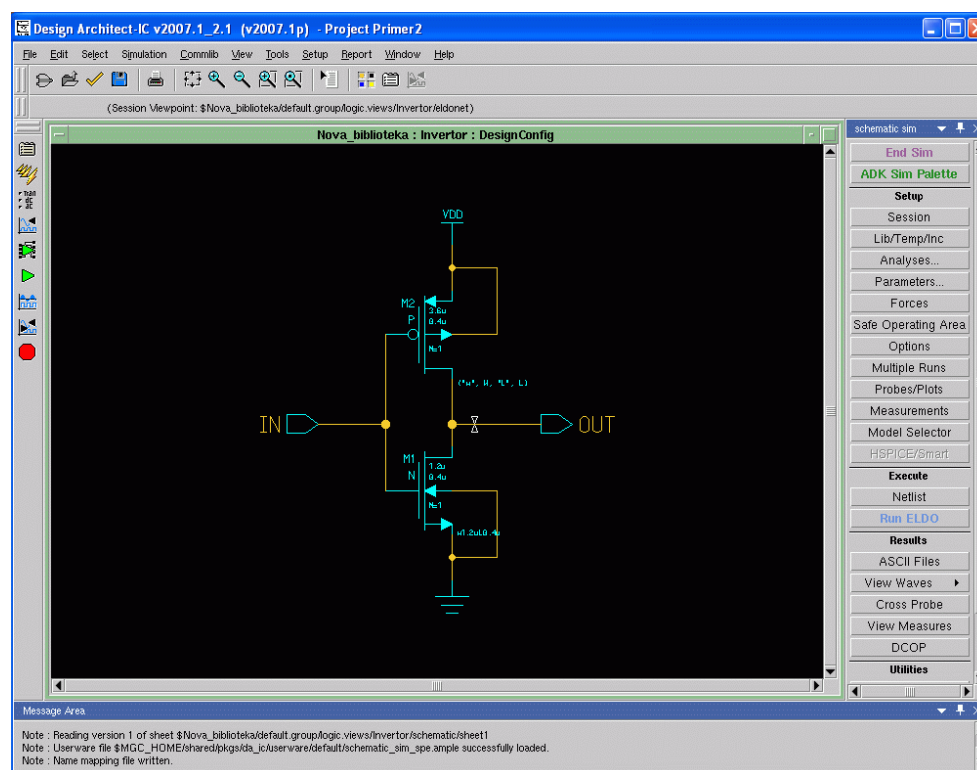


Ako projekat prođe sve provjere, dobija se sledeća poruka u dnu ekrana (slika 7.13):

Note: "Invertor/schematic/sheet1" passed check.

Da bismo kolo pripremili za simulaciju, treba kreirati *Design Viewpoint*, koji se automatski generiše kada se uđe u mod simulacije. U ovaj mod ulazi se na sledeći način.

- Kliknuti na **simulation** iz **schematic edit** palete sa desne strane ili na sa levog vertikalnog menija sa ikonicama da bismo ušli u mod simulacije.



Slika 7.14: Viewpoint

Time se na ekranu u okviru istog prozora programa *Design Architect-IC* otvara nova radna površina koja se zove *Design Viewpoint* koja je prikazana na slici 7.14. Primećuje se da iznad radne površine ne piše *Schematic#1* i *Invertor/Schematics*, već *Invertor: Design Config*, a u donjem

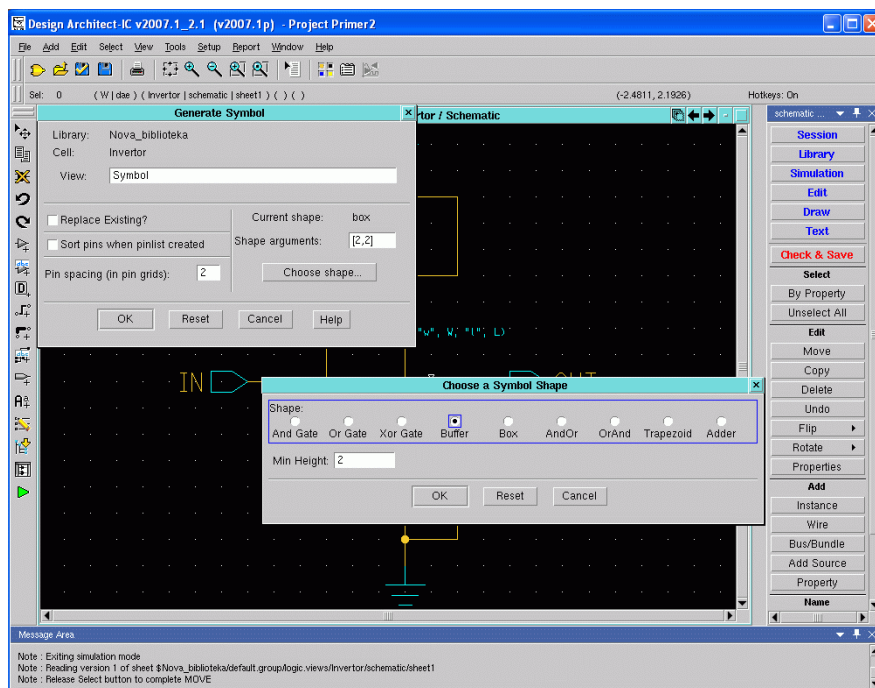
redu horizontalnog gornjeg menija piše (Session Viewpoint: lokacija i naziv projekta).

- Kliknuti **OK** da bi se prihvatile predefinisane opcije i kreirao viewpoint, koji se koristi u toku kreiranja leajauta kola.

Za povratak u šematski editor potrebno je:

- kliknuti na **end simulation** u **Schematic_sim** paleti ili na  sa leve strane vertikalnog menija sa ikonicama.

Projektovanje složenih kola olakšano je hijerarhijskim pristupom projektu. Naime, program *Design Architect –IC* dopušta da se kola koja su već opisana na tranzistorskom nivou zamene simbolima koji će se koristiti u opisu kola na višem nivou hijerarhije.



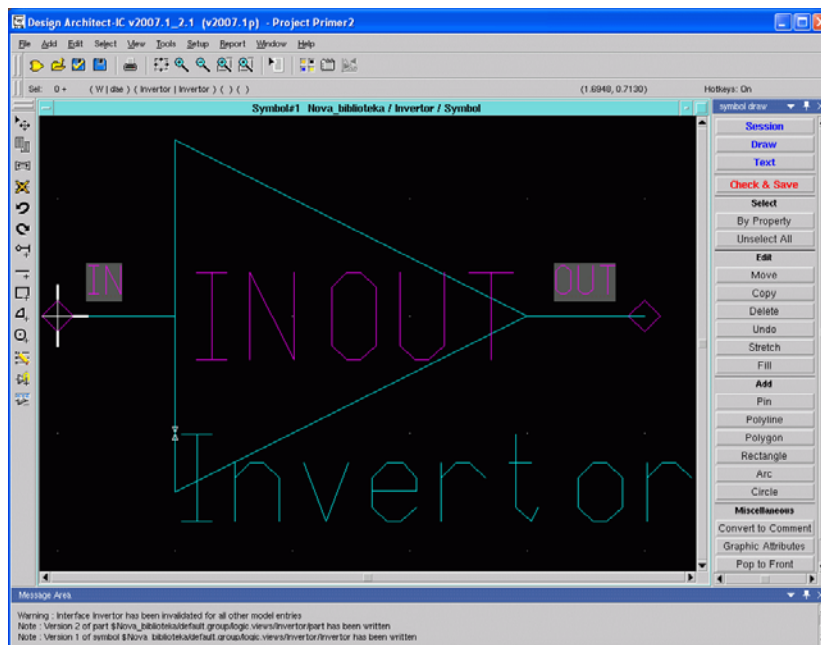
Slika 7.15: Izgled prozora pri generisanju simbola kola

Pre nego što se za neku električnu šemu kreira simbol, neophodno je da uspešno prođe verifikaciju i da bude snimljena.

Da bismo automatski generisali simbol potrebno je:

- iz prvog reda gornjeg menija izabrati **Tools -> Generate Symbol**

- u **Generate Symbol** dijalogu kliknuti: **Choose shape**, čime se otvara prozor **Choose a Symbol Shape** prikazan na slici 7.15. U njemu postoji veliki broj dostupnih simbola koji se biraju markiranjem željenog ponuđenog polja (kružić).
- izabrati **Buffer** i kliknuti **OK** (slika 7.15).
- simbol je kreiran automatski i prikazan u novom prozoru na čijem vrhu piše **Symbol#1** lokacija/naziv projekta/Symbol, kao što pokazuje slika 7.16. Ovaj simbol treba snimiti da bi mogao da se poziva za crtanje u drugim električnim šemama.



Slika 7.16: Simbol kola

Vežba broj 8

Simulacija elektronskih kola korišćenjem alata *ICstudio*

8.1 Cilj vežbe

Proučiti upotrebu *Mentor Graphics ICstudio* alata kroz primer simulacije jednostavnog elektronskog kola.

8.2 Teorijska postavka vežbe

Simulacija predstavlja nezaobilazni korak u procesu projektovanja elektronskih kola. Od rezultata simulacije projektant očekuje da dobije odgovore na sva pitanja vezana za ponašanje kola pre nego što se uđe u skupi proces fabrikacije. Imajući u vidu kompleksnost svake faze projektovanja, neracionalno je prelaziti na novu fazu pre nego što se verifikuje ispravnost prethodne. Zato se svaka faza projektovanja obavezno završava verifikacijom. Naravno, zavisno od tipa projektovanog kola (digitalno/analogno), nivoa (sistemski, algoritamski, RTL,..., električni) za verifikaciju se koriste simulatori na višem nivou ili programi za analizu kola na tranzistorskom nivou). Naravno, svaki domen opisa projekta (funkcionalni, strukturni, fizički) zahteva specifični tip simulacije.

Ne sumnjamo da se čitalac do sada susreo sa mnogim tipovima simulatora i da su mu osnovni pojmovi dovoljno jasni da bi mogao da uradi vežbu koja sledi. U svakom slučaju podsećamo da su osnovni principi vezani za simulaciju kola opisani u vežbama 2, 3, 4 i 5 ovog praktikuma. Zato prelazimo odmah na opis primene programa *ICstudio*.

8.3 Pokretanje alata *ICstudio*

Da bi se pokrenuo *ICstudio*, u komandnoj liniji treba ukucati: `icstudio&`. Korisnik treba da pokrene *ICstudio* iz direktorijuma u kome se nalazi električna šema kola koje se analizira. U konkretnom slučaju to je CMOS invertor kreiran u sastavu vežbe broj 7.

8.4 Otvaranje projekta

Da bi se otvorio projekat, treba uraditi tri sledeća koraka:

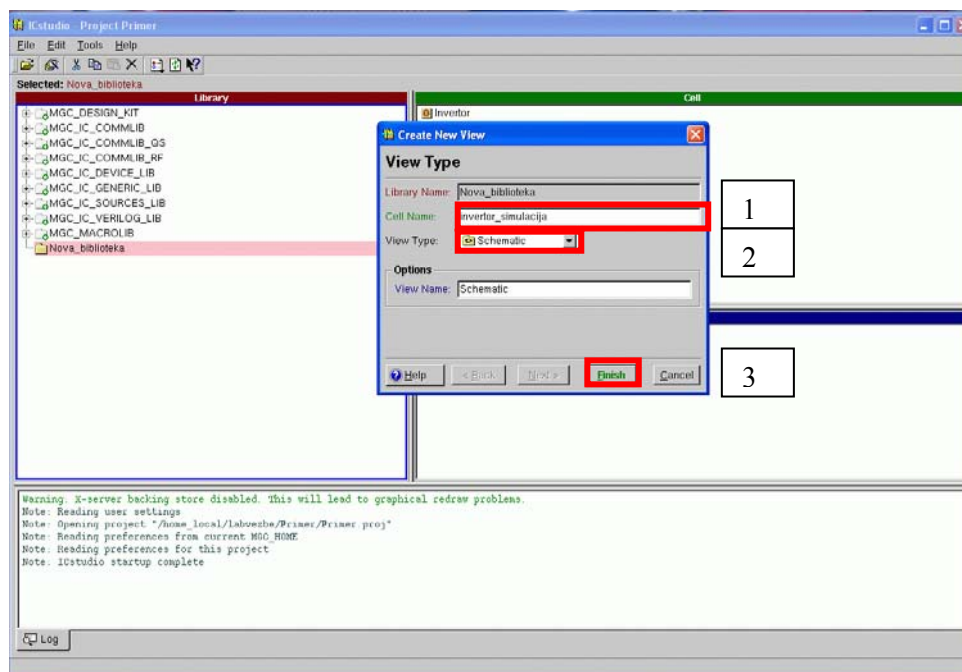
1. U *ICstudio* prozoru:

- kliknuti **File -> Open ->Project**
- uneti ime i lokaciju projekta
- kliknuti **Open** u **Open Project** prozoru

Treba kreirati novu ćeliju u postojećoj biblioteci *Nova_biblioteka*. Da bismo kreirali novu ćeliju u šematskom editoru:

- izabrati biblioteku u kojoj želimo da kreiramo ćeliju (u našem slučaju *Nova_biblioteka*)
- kliknuti **File>New>Cell View**. Pojavljuje se prozor **Create New View** (kao na slici 8.1)
- uneti ime ćelije (npr. *invertor_simulacija*) – blok 1 na slici 8.1
- izabrati **Schematic** za **View Type** – blok 2 na slici 8.1
- kliknuti **Finish** – blok 3 na slici 8.1.

Prozor izgleda kao na slici 8.1.

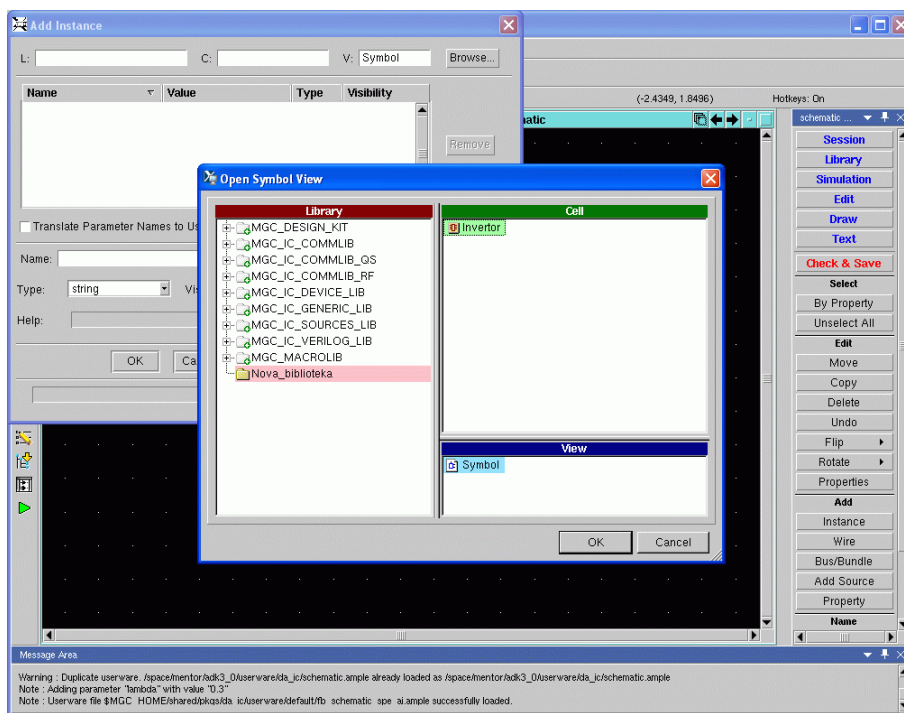


Slika 8.1: Izgled prozora kada se kreira nova ćelija u programu ICstudio

Automatski se otvara šematski editor iz programa *Design Architect-IC*.

Treba dodati simbol invertora koji je kreiran u vežbi broj 7. Sa glavnog menija izabrati **Add/Instance** tako da se otvara prozor **Add instance** (slika 8.2). Izabrati putanju ćelije Invertor iz biblioteke *Nova_biblioteka* i postaviti simbol na radnu površinu kao na slici 8.3. Sada treba uneti i druge elemente da bi se formiralo električno kolo za simulaciju:

- dodati izvor jednosmernog napajanja tako što se sa menija sa desne strane izabere komponenta **Library/Sources Lib/Independent/DC(V)**. Pozicionirati miša na simbol DC i pritisnuti **shift + F7**. Otvara se novi prozor u kome se menjaju podrazumevane vrednosti atributa. U konkretnom slučaju treba promeniti jednosmernu vrednost DC generatora sa 1V na 3.3V. Nakon toga obavezno kliknuti **Apply/OK**. Na sličan način menjaju se atributi i ostalih komponentata.



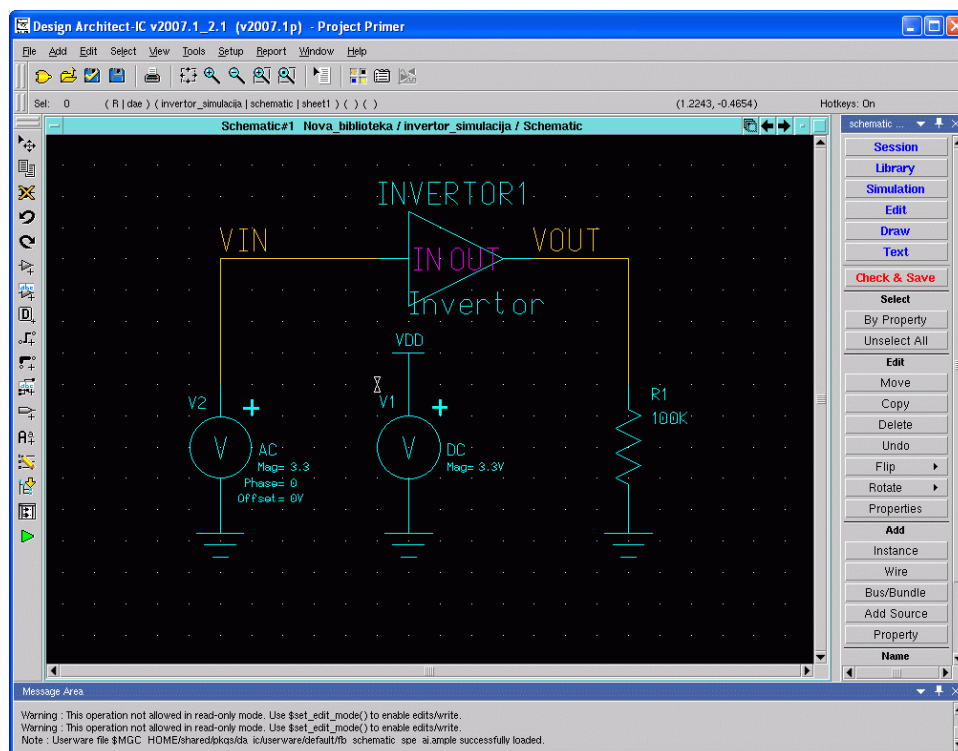
Slika 8.2: Dodavanje simbola invertora kreiranog u vežbi broj 7

- dodati naizmenični generator za AC analizu tako što se sa menija sa desne strane izabere komponenta **Library/Sources Lib/Independent/AC(V)**. Postaviti atribut *Mag* na 3.3V na isti način kao za jednosmerni generator.
- dodati potrošač tako što se sa menija sa desne strane izabere komponenta **Library/Device Lib/Resistor/ideal**. Postaviti atribut *Mag* na 100KΩ.
- dodati globalne simbole mase i napajanja tako što se sa menija sa desne strane izabere **Library/Sources Lib/Globals/VDD** i **Library/Sources Lib/Globals/Ground**.

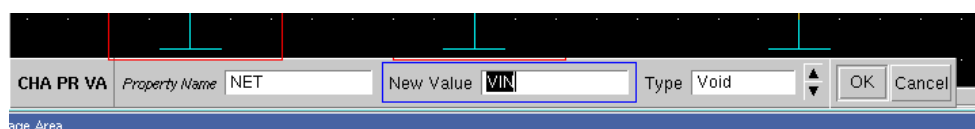
Povezati simbole unete u šematski editor tako da šema izgleda kao na slici 8.3. Komanda povezivanja aktivira se pritiskom na funkcijski taster **F3**.

Dodeliti nazive čvorovima (VIN i VOUT, slika 8.3). (Napominjemo da se u originalnom uputstvu za ICstudio deo kola koji se nalazi na istom potencijalu naziva *net* - u prevodu znači mreža; jasno je da ovaj pojam odgovara pojmu čvora. Da bi se čitalac lakše prilagodio nazivima koji se sreću u menijima ICstudio alata, u daljem tekstu koristiće se reč *net*.) Selektovati *net* levim klikom, pritiskom na desni klik otvara se padajući meni iz koga treba izabrati *Name nets*:. U donjem

delu ekrana se pojavljuje prozor kao sa slike 8.4; dodeliti ime atributu *Name Value*, kliknuti OK.



Slika 8.3: Izgled električne šeme



Slika 8.4: Dodeljivanje imena čvorovima

Kada je šema nacrtana, kliknuti *Check&Save* sa *schematic edit* palete ili  iz menija na vrhu ekrana da bi projekat bio proveren i snimljen. Ukoliko ima grešaka otvoriće se novi prozor u kome se mogu videti postojeće greške u vidu tekstualnog fajla.

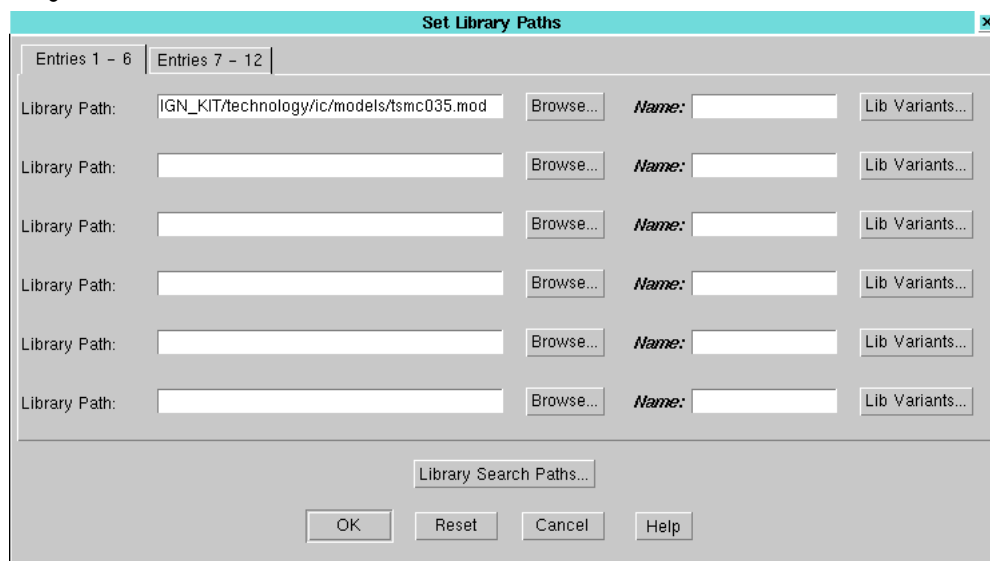
Sljedeći korak je ulazak u *Design Context mode*. U ovom modu se pravi virtuelno okruženje ćelije koja se testira tj. definiše se tip simulacije koja će se

izvršavati i zadaju se odgovarajući parametri. U ovaj mod se ulazi klikom na **simulation** iz **schematic edit** palete sa desne strane ili na  sa levog vertikalnog menija sa ikonicama.

Ulaskom u ovaj mod automatski se kreira novi izgled ćelije (*View*). Dodeljuje mu se ime *DesignConfig*. Korisnik može promeniti ovaj naziv kako bi ga asocirao na tip simulacije koju će obaviti (npr. *DesignConfig_AC*, ako se radi o AC analizi). Na ovaj način se svi predefinisani parametri kao i tip simulacije pamte u novokreiranom konfiguracionom (*DesignConfig*) izgledu. Ova opcija je jako korisna ukoliko se kasnije želi iznova simulirati ista ćelija. Korisnik samo treba da pokrene *DesignConfig* izgled čime se svi parametri i tip simulacije učitavaju u radnu memoriju. Nakon toga može se odmah pokrenuti simulacija pozivanjem ELDO simulatora bez ponovnog zadavanja parametara i tipa simulacije.

Podešavanje parametara simulacije obavlja se kroz sledeće korake:

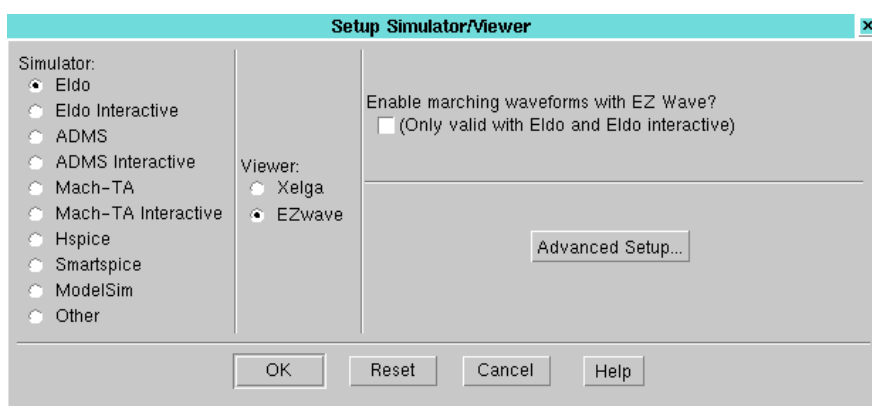
- sa menija sa desne strane izabrati **Setup/(Lib/Tech/Inc)/Libraries**. Pojavljuje se dijalog **Setup Library Paths** kao na slici 8.5 gde treba učitati sledeći fajl: *space/mentor/adk3_0/technology/ic/models/tsmc035.mod* i kliknuti **OK**.



Slika 8.5: Učitavanje fajla potrebnog za simulaciju

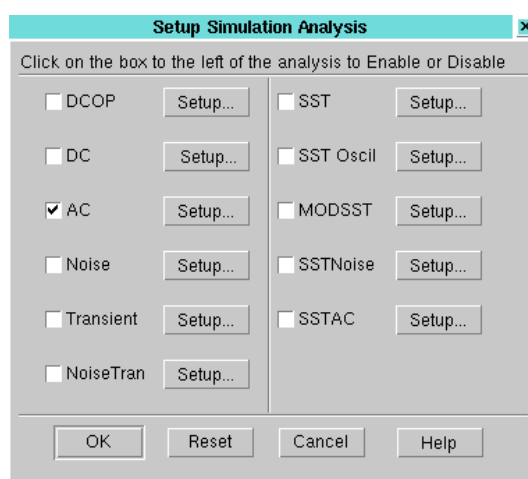
Da bismo izabrali simulator i grafički postprocesor, potrebno je iz glavnog menija izabrati **Setup/(Simulator/Viewer)**. Dobija se dijalog **Setup**

Simulator/Viewer kao na slici 8.6. U polju za izbor simulatora treba izabrati Eldo a u polju za izbor Viewer-a treba izabrati EZwave.



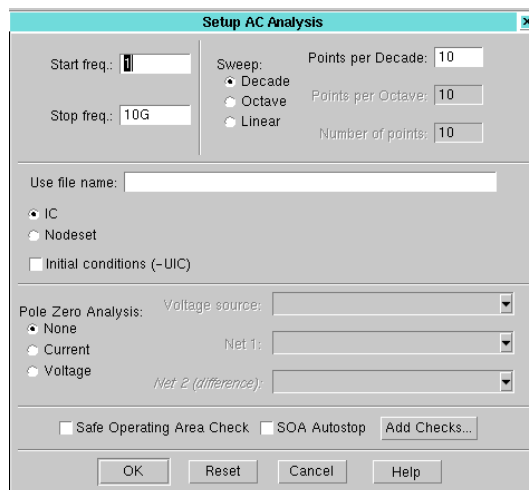
Slika 8.6: Dijalog za izbor simulatora i viewer-a

Tip simulacije se bira klikom na **ADK Sim palette/Set up sim/Analysis** sa menija sa desne strane. Otvara se prozor **Setup Simulation Analysis** kao na slici 8.7 gde treba izabrati AC, a zatim kliknuti **Setup** radi podešavanja parametara simulacije.



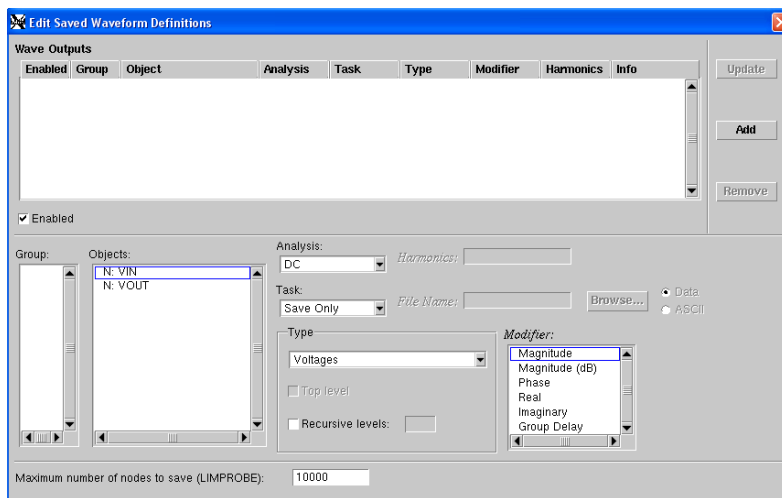
Slika 8.7: Dijalog za izbor tipa simulacije

Treba postaviti parametre kao na slici 8.8. i kliknuti OK.



Slika 8.8: Dijalog za podešavanje parametara simulacije

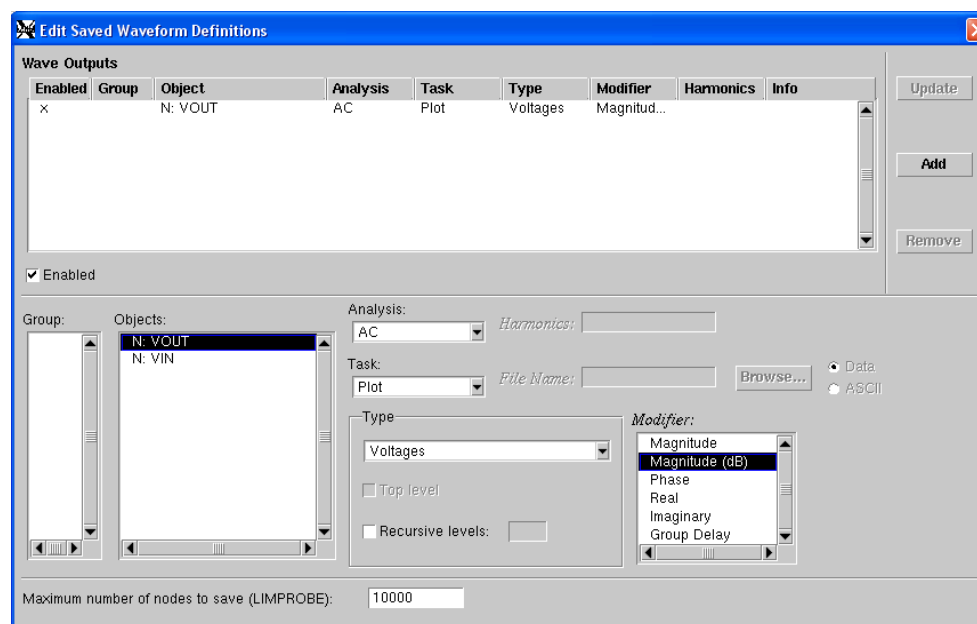
Sledeći korak je selektovanje netova koje želimo da iscrtamo: na šemi selektovati netove VIN i VOUT (*shift* + levi klik za višestruko selektovanje). Odabrati opciju sa glavnog menija **Setup/Outputs**. Pojavljuje se dijalog **Edit Saved Waveform Definitions** kao sa slike 8.9 koji treba popuniti na sledeći način:



Slika 8.9: Dijalog za izbor netova

- U polju **Objects** selektovati net koji se iscrtava na grafiku (u našem slučaju VOUT),

- U polju **Analysis** odabrati tip simulacije (u našem slučaju AC)
- U polju **Task** odabrati **Plot**
- U polju **Type** odabrati **Voltages**
- U polju **Modifier** odabrati **Magnitude(dB)**
- Kliknuti na dugme **Add** i selektovani net sa zadatim parametrima će se pojaviti u prozoru **Wave Outputs** kao na slici 8.10. Zatvoriti formu.

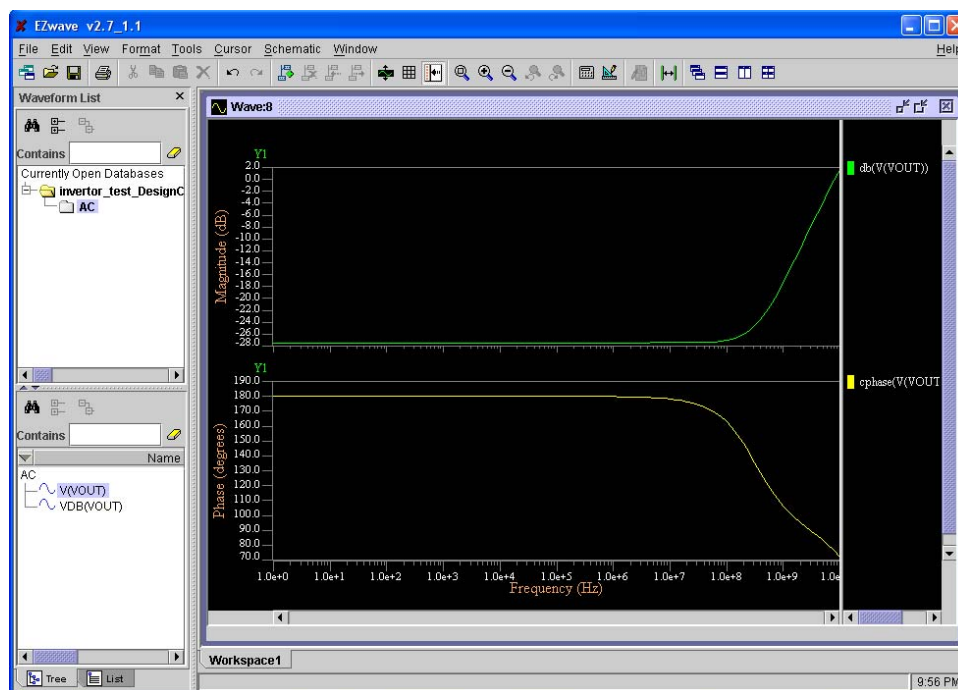


Slika 8.10: Dijalog za izbor netova sa podešenim opcijama

Netlista se kreira klikom na **Execute/Netlist** sa menija sa desne strane. Nakon ove akcije u **Message Area** prozoru treba da stoji "Note: Netlist completed successfully".

Simulacija se pokreće klikom na **Execute/Run Eldo** sa menija sa desne strane. Nakon ove akcije u **Message Area** prozoru treba da stoji "Note: Simulation completed successfully". Ukoliko ima nekih grešaka one se mogu videti u **Log** fajlu u donjem radnom prozoru pored **Message Area** prozora.

Pregled rezultata simulacije moguće je obaviti klikom na **Results/View Waves** (treba zadržati levi klik) čime se aktivira grafički postprocesor EZwawe. Kao rezultat AC analize dobijaju se amplitudska i fazna karakteristika kao na slici 8.11.



Slika 8.11: Rezultati AC analize

Da bi se ovaj izgled sačuvao, kliknuti na ikonicu *Save*. Klikom na **End sim** sa menija sa desne strane izlazi se iz *Design Context* moda, a prikaz editora *Design Architect*-a se vraća u standardan šematski izgled.

Na sličan način mogu se izvršavati simulacije u drugim domenima (DC, vremenski).

Vežba broj 9

Crtanje lejauta invertora korišćenjem alata *ICstudio*

9.1 Cilj vežbe

Proučiti upotrebu *Mentor Graphics ICstudio* alata kroz primer crtanja lejauta CMOS invertora. Uraditi verifikaciju lejauta (DRC-*Design Rule Check*) i poređenje lejauta i šeme invertora iz vežbe broj 7 (*LVS-Layout Versus Schematic*).

9.2 Teorijska postavka vežbe

Posle uspešne verifikacije projekta invertora (čija je električna šema nacrtana u vežbi 7) simulacijom u prethodnoj vežbi, prelazi se na fizičko projektovanje. Postupak fizičkog projektovanja koji će biti predmet ove vežbe odgovara stilu potpunog projektovanja po narudžbini (*Full custom design*). Da bi se upustio u fizičko projektovanje neophodno je da projektant detaljno poznaje tehnologiju izrade integrisanih kola. Ovo podrazumeva da zna koliko i koji tehnološki slojevi mu stoje na raspolaganju. Osim toga, potrebno je da zna koje fizičke maske mora da projektuje da bi integrisano kolo moglo da se realizuje u raspoloživoj tehnologiji. Zato se čitalac upućuje da prouči poglavlje „CMOS proces“ u [Pet09].

U okviru ove vežbe studenti će naučiti kako da:

- fizički nacrtaju sve maske neophodne za realizaciju invertora;
- provere da li su ispoštovana sva pravila projektovanja (*Design Rule Check* - DRC);
- provere da li se električna šema ekstrahovana iz nacrtanog lejauta poklapa sa željenom električnom šemom kola (*Layout Versus Schematics* - LVS).

Kao i u prethodne dve vežbe, sve faze projektovanja realizuju se u okruženju alata *ICstudio* sa platforme *Mentor Graphics*.

9.3 Pokretanje alata *ICstudio*

Da bi se pokrenuo *ICstudio*, u komandnoj liniji treba ukucati: `icstudio&`. Korisnik treba da pokrene *ICstudio* iz direktorijuma u kome se nalazi električna šema CMOS invertora (kreirana u vežbi broj 7).

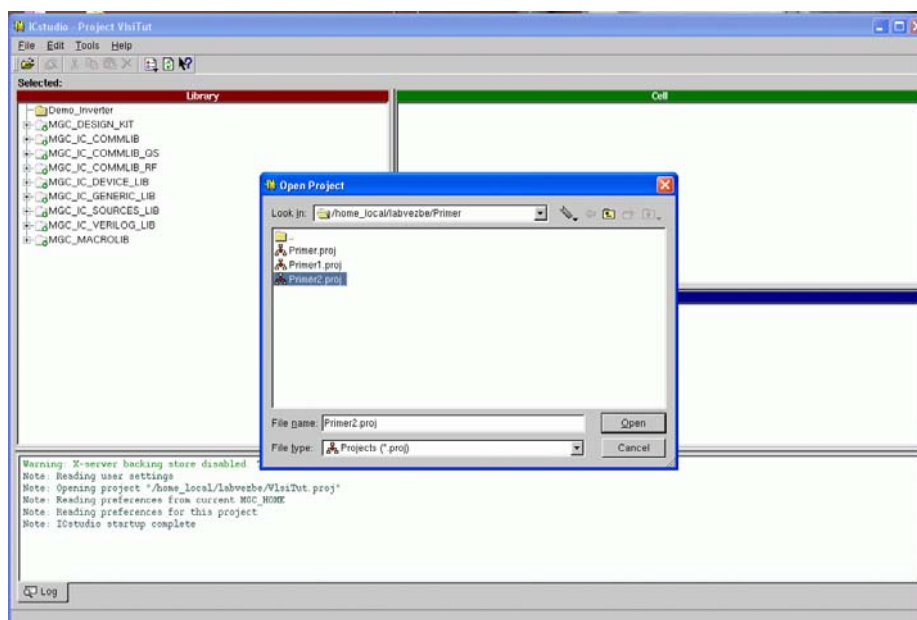
9.4 Otvaranje projekta

Da bi se otvorio projekat, treba uraditi tri sledeća koraka:

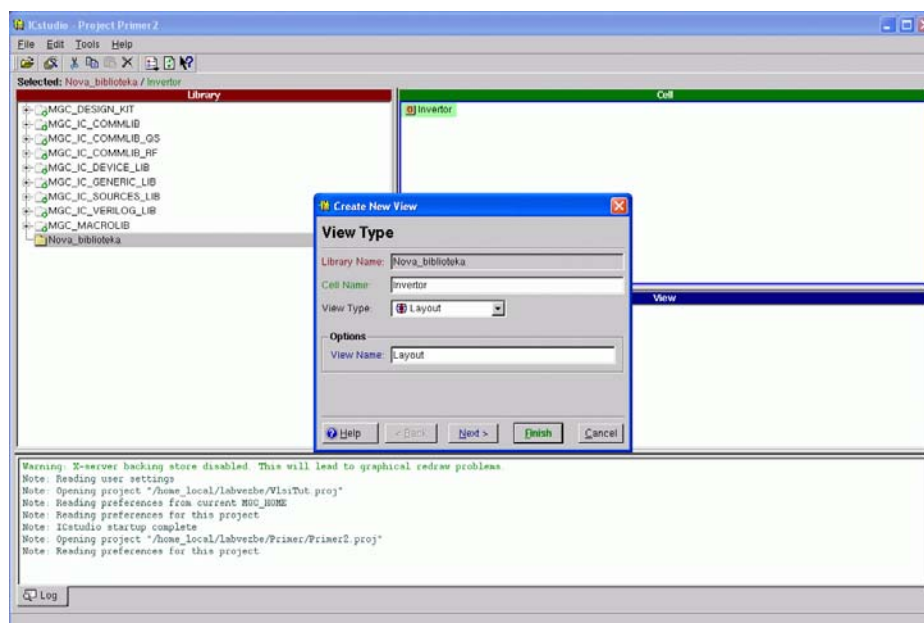
–U *ICstudio* prozoru:

- kliknuti **File -> Open ->Project**
- uneti ime i lokaciju projekta
- kliknuti **Open** u **Open Project** prozoru

Prozor izgleda kao na slici 9.1.



Slika 9.1: Izgled prozora kada se otvara projekat



Slika 9.2: Izgled prozora kada se otvara leajut ćelije

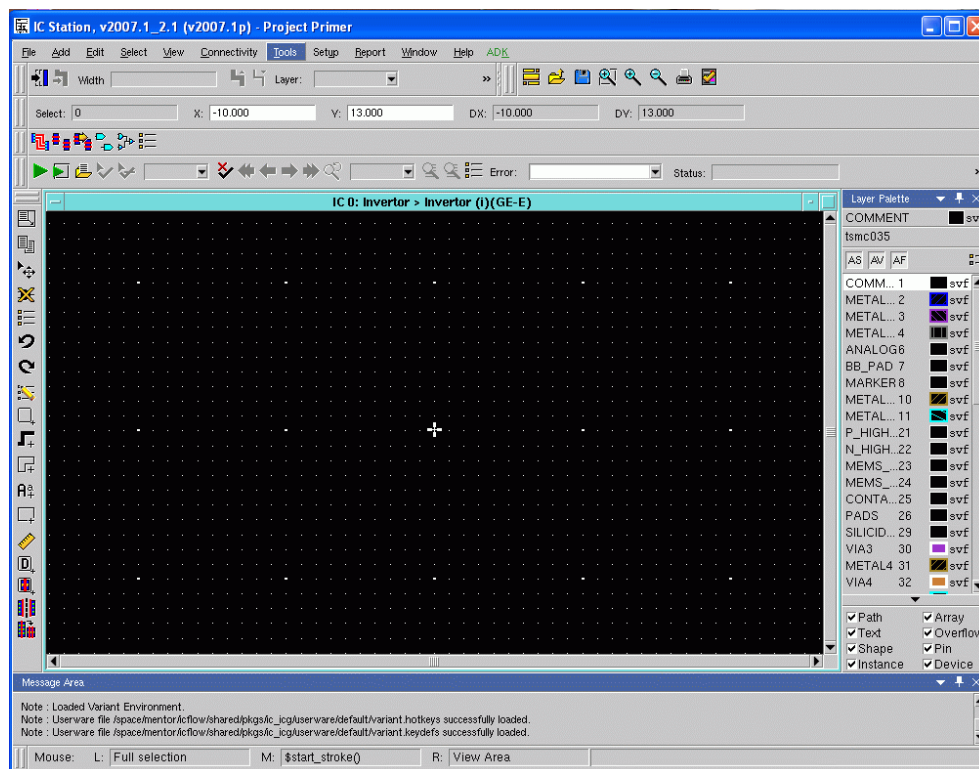
Da bi se otvorio lejaut ćelije, treba:

–U *ICstudio* prozoru:

- kliknuti desnim klikom ime ćelije koju smo kreirali u vežbi 7 (u ovom slučaju **Invertor**) i zatim kliknuti **New View**.
- izabrati **Layout** za **View Type**
- kliknuti **Finish**.

Prozor izgleda kao na slici 9.2.

Automatski se otvara prozor sa slike 9.3. U prozoru se osim menija i radne površine u kojoj se crta lejaut kola, sa desne strane nalazi i paleta slojeva (*layer pallette*) raspoloživih za datu tehnologiju.



Slika 9.3: Radna površina u kojoj se crta lejaut kola

9.5 Crtanje leajauta invertora

Pre početka rada, treba proveriti da li se nalazimo u modu za editovanje. Da bismo bili u tom modu, treba:

- kliknuti **File>Enable editing>Current Context**.

Na radnoj površini, jedna jedinica je jednaka 1λ . Ako poštujemo pravila projektovanja za *TSMC0.35u* tehnologiju, najmanja veličina će biti $2\lambda=0.4u$ (za širinu poli-gejta i veličinu kontakta). Iz toga sledi $\lambda=0.2u$.

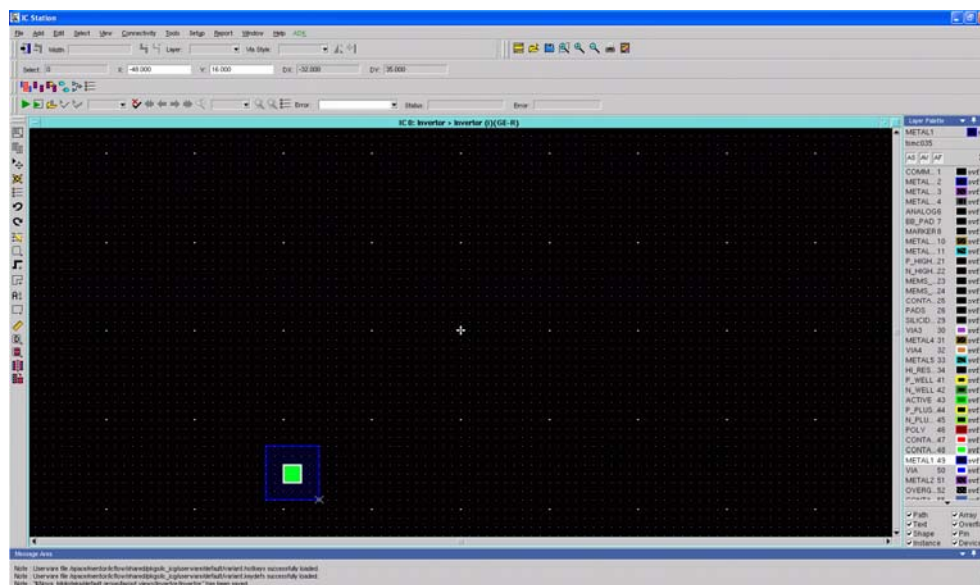
Da bismo nacrtali leajaut NMOS tranzistora dužine $0.4u$ (2λ) i širine $1.2u$ (6λ) (dimenzije tranzistora su definisane u vežbi 7), treba:

- kliknuti **CONTACT_TO_ACTIVE** sa palete slojeva. To je sloj označen brojem 48.

- kliknuti **Draw Rectangle** sa leve strane vertikalnog menija

- nacrtati kvadrat površine $2\lambda \times 2\lambda$ u sloju 48

- oko kvadrata $2\lambda \times 2\lambda$ istim postupkom nacrtati kvadrat $6\lambda \times 6\lambda$ u sloju **METAL1** (bira se sa palete slojeva, kao sloj označen brojem 49) koji je zapravo kontakt. Dobija se prozor sa slike 9.4.



Slika 9.4: Izgled prozora kada je nacrtan jedan kontakt za NMOS

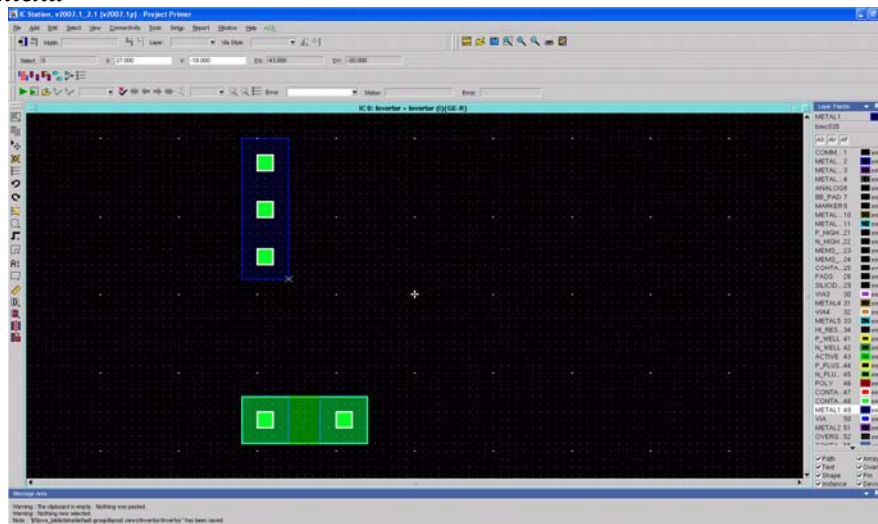
- selektovati oba kvadrata i kliknuti **Edit>Copy>Selected**
- kliknuti **Edit>Paste**
- postaviti kvadrate pored originala tako da rastojanje bude 4λ , kao na slici 9.5. Ovo su kontakti za sors i drejn.
- kontakte za drejn i sors pokriti površinom $6\lambda \times 16\lambda$ slojem **ACTIVE (43)**.

Da bismo nacrtali lejaut PMOS tranzistora dužine $0.4u$ (2λ) i širine $3.6u$ (18λ) (dimenzije tranzistora su definisane u vežbi 7), treba:

- kliknuti **CONTACT_TO_ACTIVE (48)** sa palete slojeva.
- kliknuti **Draw Rectangle** sa leve strane vertikalnog menija
- nacrtati kvadrat površine $2\lambda \times 2\lambda$
- kvadrat površine $2\lambda \times 2\lambda$ iskopirati još 2 puta tako da rastojanje između 2 kvadrata bude 4λ , kao na slici 9.5.
- napraviti kontakt površine $18\lambda \times 6\lambda$ u sloju **METAL1**. Kontakt kod PMOS tranzistora je veći 3 puta nego kod NMOS-a zbog ranije definisanih dimenzija tranzistora.

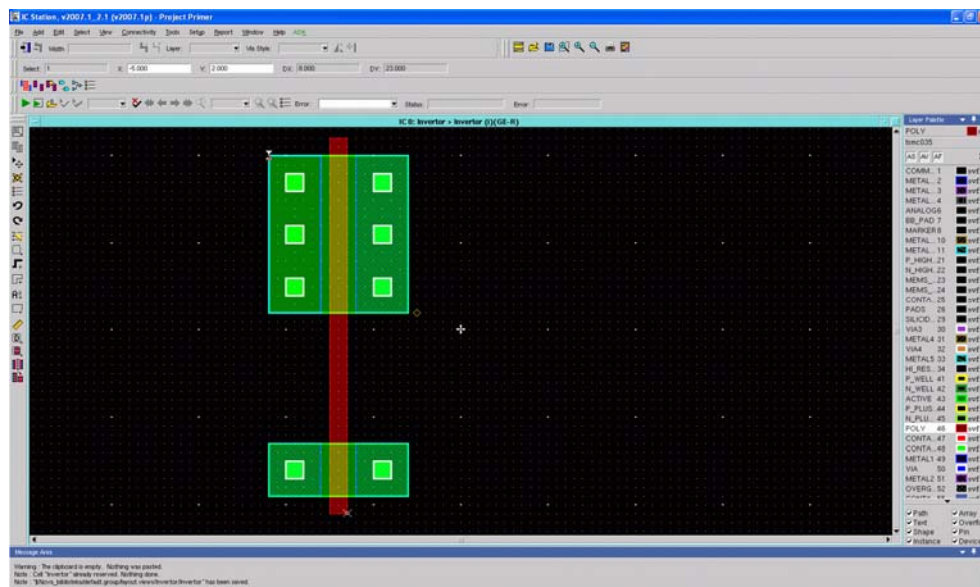
Rastojanje između NMOS i PMOS tranzistora treba da bude minimalno 13λ , a u našem primeru sa slike 9.5 je 15λ .

Napomena: Projekat treba snimati što češće u toku rada, zato što ne postoji opcija automatskog snimanja na određeno vreme. Posle svakog snimanja projekta, dalje editovanje je moguće samo nakon komande **File>Enable editing>Current Context**.



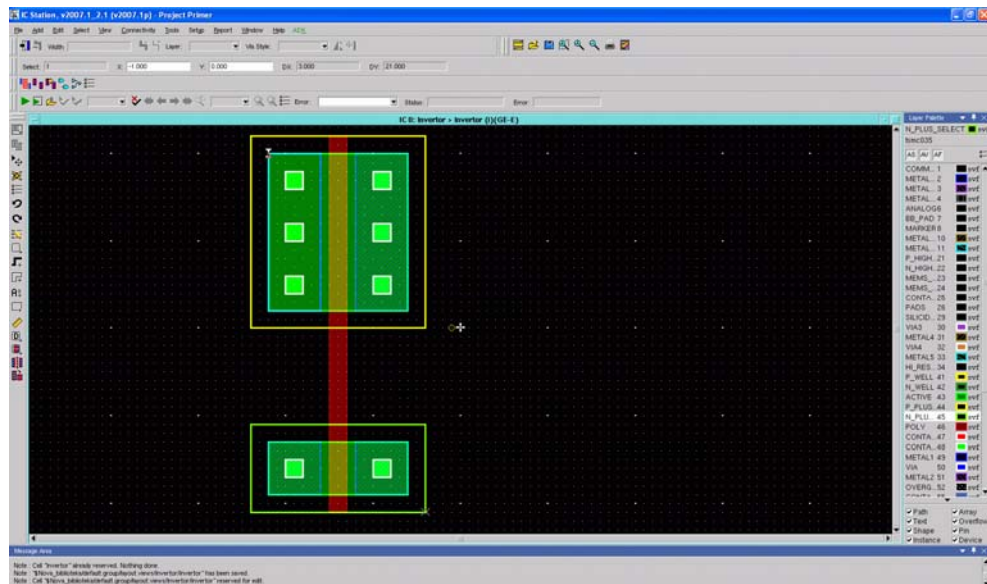
Slika 9.5: Difuzija za NMOS (dole) i početak crtanja PMOS tranzistora (gore)

- selektovati kontakt PMOS-a i kliknuti **Edit>Copy>Selected**
- kliknuti **Edit>Paste**
- postaviti kontakt pored originala, tako da rastojanje bude 4λ , kao na slici 9.6. Ovo su kontakti za sors i drejn.
- kontakte za drejn i sors pokriti površinom $18\lambda \times 16\lambda$ slojem **ACTIVE** (43).
- napraviti poli gejt površine $43\lambda \times 2\lambda$ kao pravougaonik koristeći sloj **POLY** (46), kao na slici 9.6. Poli gejt treba da bude između kontakata drejna i sorsa NMOS i PMOS tranzistora.

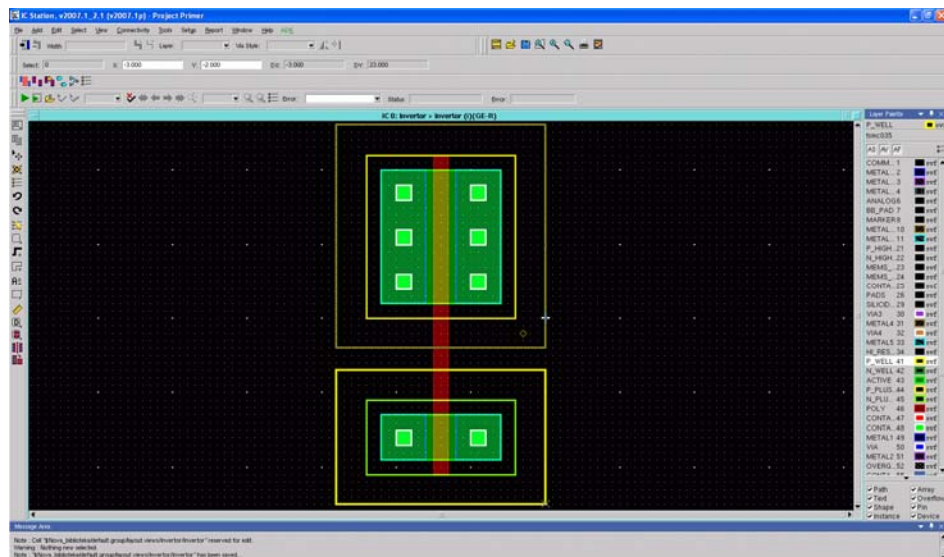


Slika 9.6: Povezivanje CMOS para zajedničkim gejtom

- uokviriti PMOS pravougaonikom veličine $22\lambda \times 20\lambda$ u sloju **P_PLUS_SELECT** (44), a NMOS pravougaonikom veličine $10\lambda \times 20\lambda$ u sloju **N_PLUS_SELECT** (45). Ovo nisu fizički slojevi, već samo služe za lakše razlikovanje P i N oblasti. Rastojanje između slojeva **N_PLUS_SELECT** i **P_PLUS_SELECT** treba da bude minimalno 9λ , u našem primeru je to rastojanje 11λ (slika 9.7).
- dodati NWELL za PMOS u sloju **N_WELL** (42) veličine $30\lambda \times 28\lambda$, i PWELL za NMOS u sloju **P_WELL** (41) veličine $18\lambda \times 28\lambda$. Rastojanje između NWELL-a i PWELL-a treba da bude minimalno 1λ , u našem primeru je to rastojanje 3λ (slika 9.8).

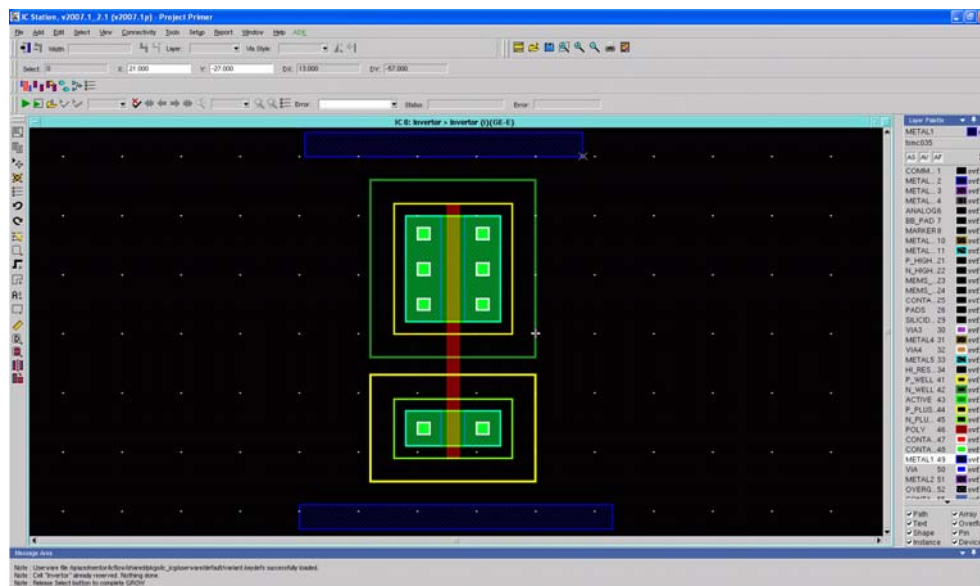


Slika 9.7: Ucertavanje *n_plus_select* i *p_plus_select* slojeva



Slika 9.8: Ucertavanje *n-well* i *p-well* slojeva

Sledeći korak je dodavanje linija za napajanje i masu. Linije se dodaju u sloju **METAL1**, imajući u vidu da širina linije u sloju **METAL1** mora da bude minimalno 3λ , kao i da rastojanje dve linije u sloju **METAL1** mora da bude minimalno 3λ . Posle dodavanja ovih linija, prozor izgleda kao na slici 9.9.

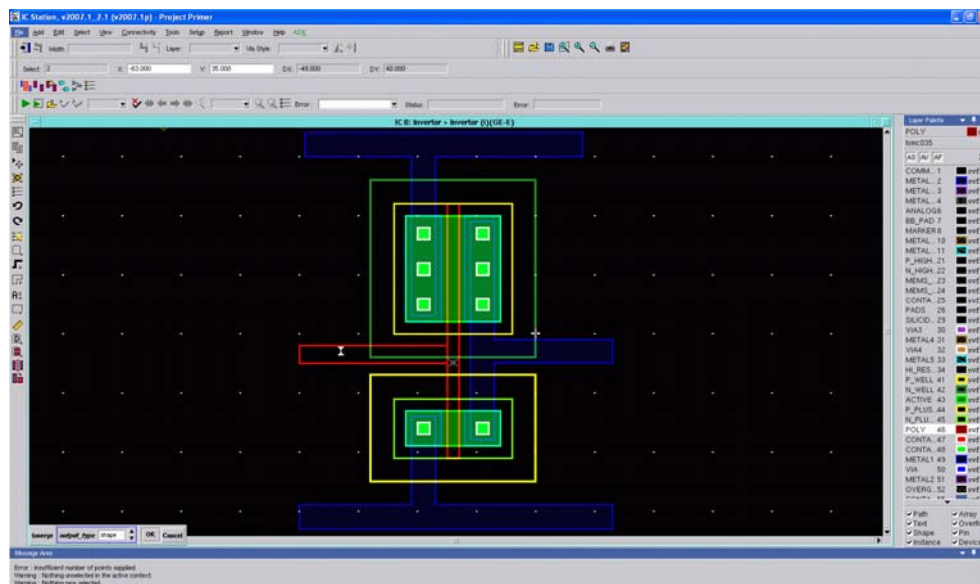


Slika 9.9: Dodavanje metala za razvođenje VDD (gore) i VSS (dole)

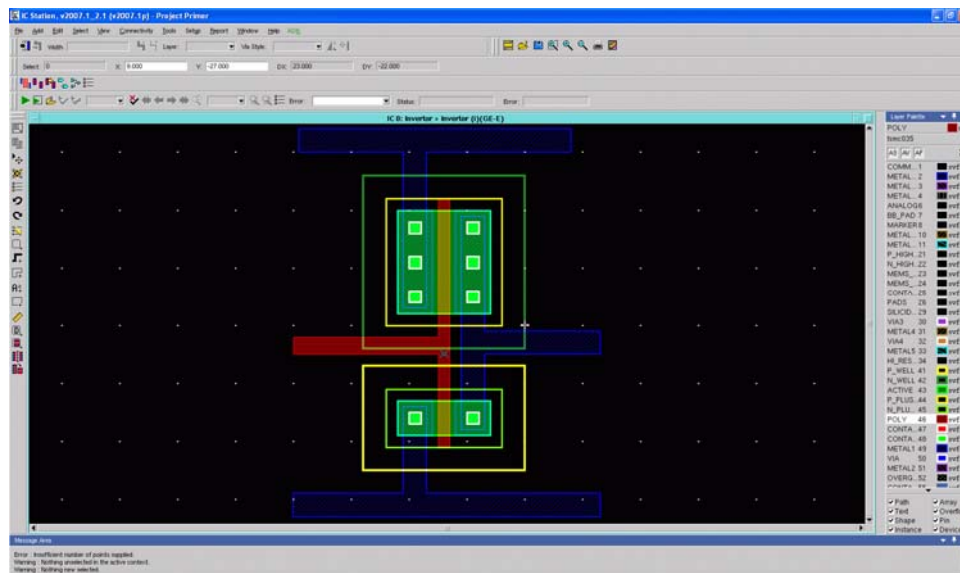
Linije za napajanje i masu treba povezati sa tranzistorima. Najpre povezujemo sors NMOS tranzistora sa masom i to na sledeći način:

- nacrtati pravougaonik u sloju **METAL1** koji treba da poveže sors NMOS tranzistora i masu. Prvo selektovati taj pravougaonik, a onda, držeći taster SHIFT selektovati i pravougaonik koji predstavlja liniju za masu, pa kad su selektovana oba pravougaonika, kliknuti **Edit>Merge**. U donjem levom uglu pojavljuje se mali dijalog-prozor (slika 9.10), i kada se u tom prozoru klikne **OK**, dva pravougaonika su povezana, tj. vezan je sors NMOS-a za masu. Na sličan način se vezuje i sors PMOS-a za napajanje.

Da bismo kreirali vezu do izlaza invertora, treba u sloju **METAL1** povezati drejnovne dva tranzistora i napraviti izlaznu liniju. Veza do ulaza invertora razvodi se u sloju **POLY**, slika 9.11.



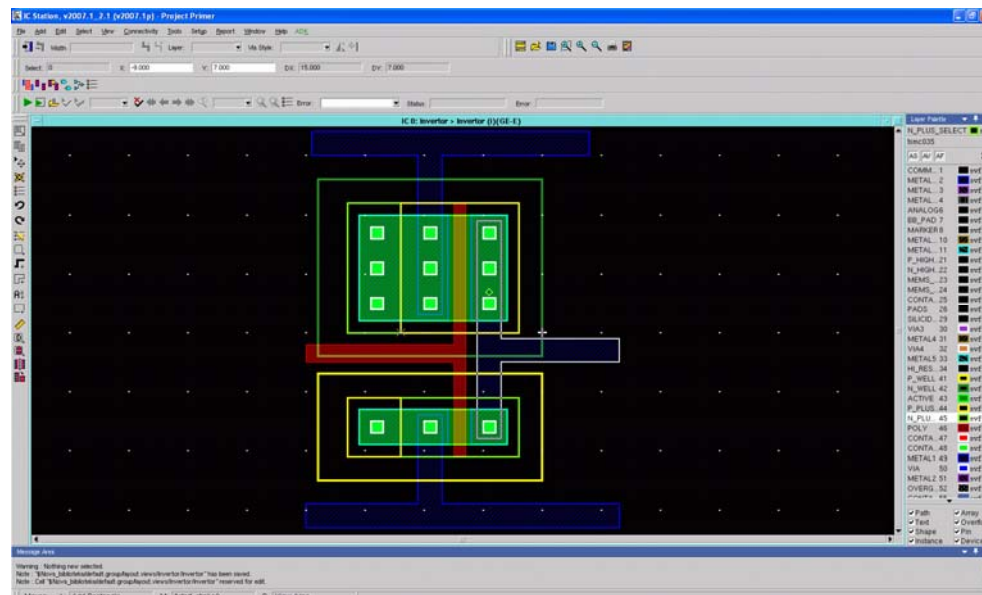
Slika 9.10: Vezivanje sorsa nMOS za VSS, pMOS za VDD, priprema za ulazni i izlazni priključak



Slika 9.11: Vezivanje ulaznog i izlaznog priključka

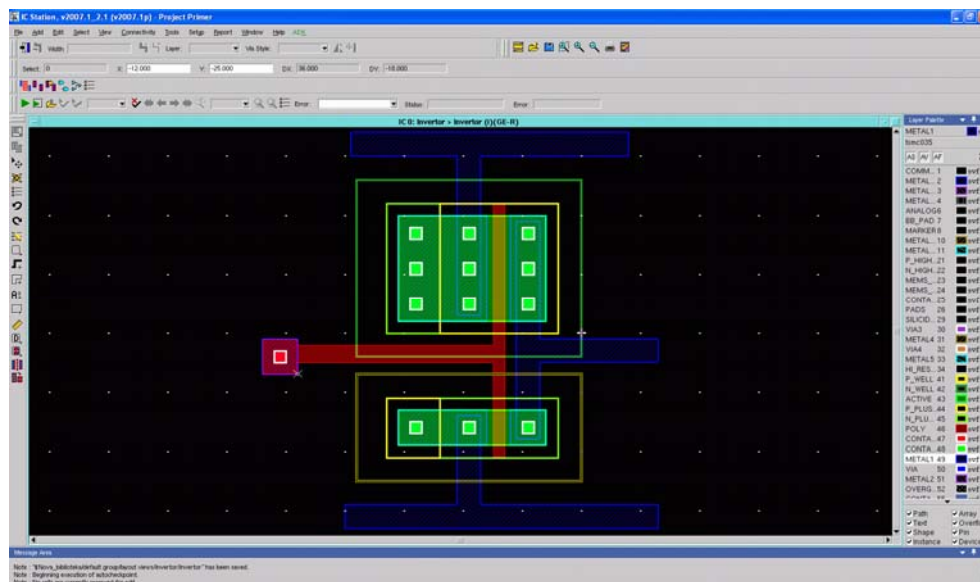
Sledeći korak je “širenje” područja slojeva **METAL1** i **ACTIVE** sa strane sorsa da bi se kreirao kontakt do osnove kod oba tranzistora. Objasnićemo postupak na primeru NMOS tranzistora.

- kliknuti **Edit>Stretch**.
- stati kursorom na levu ivicu sorsa NMOS-a, kliknuti levim tasterom, i vući ulevo za 9λ . Kada se pređe rastojanje od 9λ , kliknuti ponovo levim tasterom i tada se komanda deaktivira. Kao rezultat dobija se proširena oblast. Ovo treba uraditi za područje slojeva **METAL1** i **ACTIVE**. Takođe treba dodati i kvadrat površine $2\lambda \times 2\lambda$ u sloju **CONTACT_TO_ACTIVE**. Istim postupkom treba proširiti i **P_WELL** za 10λ , a **N_PLUS_SELECT** ne treba menjati. Oko kreiranog kontakta treba postaviti sloj **P_PLUS_SELECT** površine $9\lambda \times 10\lambda$ kao što je prikazano na slici 9.12.



Slika 9.12: Vezivanje osnove nMOS za VSS a pMOS za VDD

Sličan postupak se primenjuje i kod PMOS-a, osim što se oko kontakta postavlja sloj **N_PLUS_SELECT** površine $9\lambda \times 22\lambda$. Posle ovog postupka, leajaut izgleda kao na slici 9.12.



Slika 9.13: Crtanje ulaznog pina

Pinovi se crtaju na sledeći način:

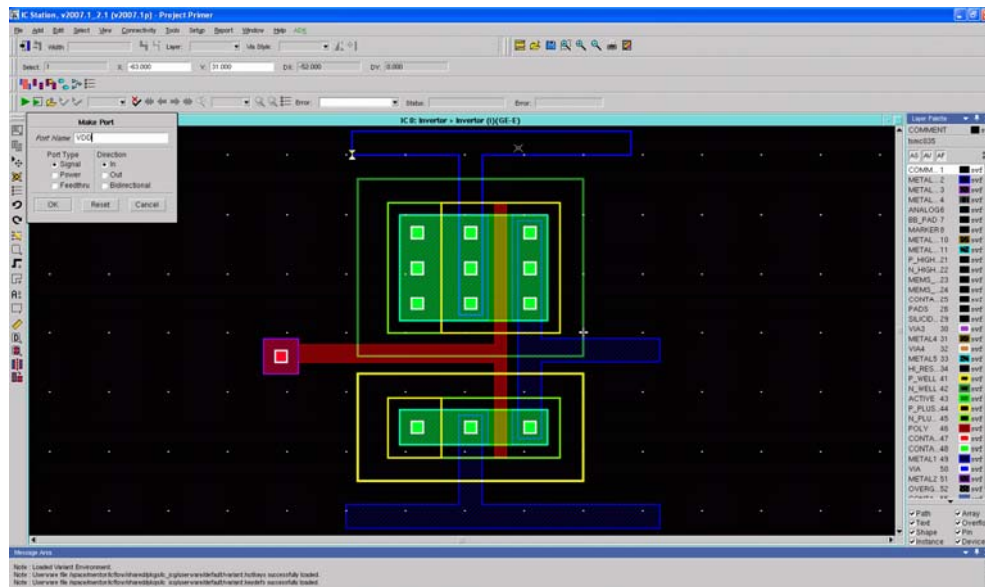
- izabrati sloj **CONTACT_TO_POLY (47)** sa palete slojeva;
- nacrtati pravougaonik površine $2\lambda \times 2\lambda$ sa leve strane gejta – ulazni kontakt;
- oko tog pravougaonika nacrtati najpre pravougaonik površine $6\lambda \times 6\lambda$ u sloju **POLY**, a zatim preko njega još jedan iste površine u sloju **METAL1**;
- povezati zajednički gejt sa stopicom koristeći opciju **Edit>Merge** kao što je ranije opisano.

Nakon ovog postupka, ležaut izgleda kao na slici 9.13.

Označavanje portova objasnićemo na primeru označavanja linije za napajanje VDD:

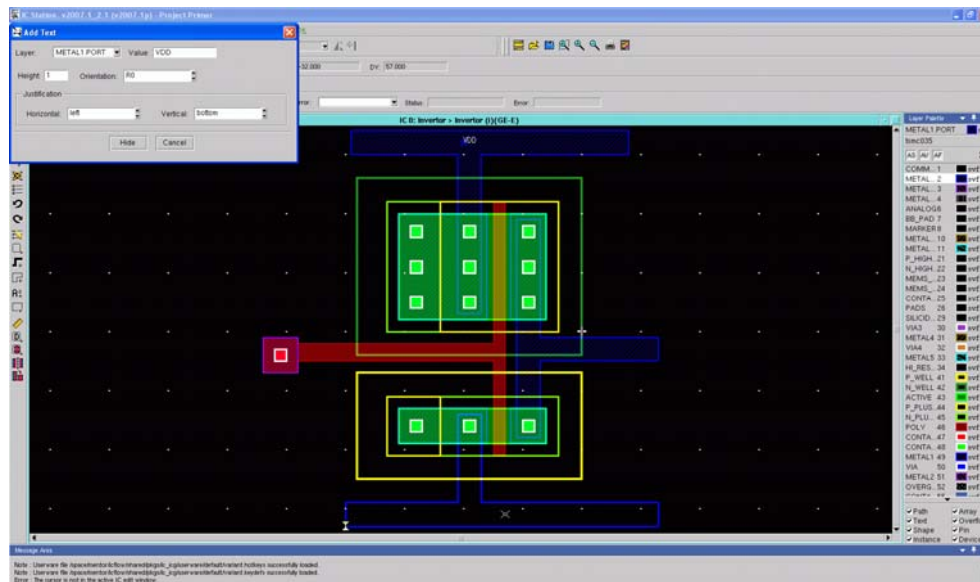
- kliknuti na površinu u sloju **METAL1** koja predstavlja liniju za napajanje da bi bila selektovana
- iz glavnog menija izabrati **Connectivity -> Port -> Make Port:**
- u polje **Port Name** uneti VDD. Proveriti da li je **Port Type** izabran da bude **Signal** i da li je **Direction** izabran da bude **in** (slika 9.14)
- kliknuti **OK**

Ponoviti ove korake za označavanje portova GND, IN i OUT, za masu, ulazni i izlazni port, respektivno. U polje **Direction** kod GND i IN treba upisati mod **in**, a kod OUT mod **out**.

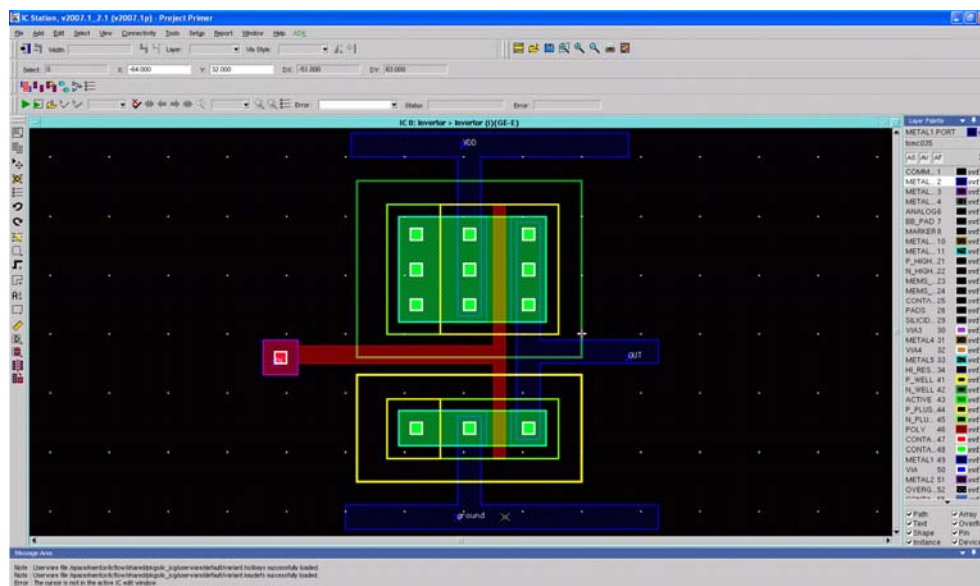


Slika 9.14: Označavanje porta VDD (prozor gore levo)

- izabrati sloj **METAL1.PORT (2)** sa palete slojeva
 - sa glavnog menija izabrati **Add ->Text**
 - uneti labelu VDD u polje *Value* prozora **Add Text**, slika 9.15
 - stati kursorom u levi donji ugao radne površine jer se na tom mestu nalazi nova labela (njen početni položaj je definisan u prozoru **Add Text** kao Justification, slika 9.15, i može se promeniti)
 - kursorom vući labelu do mesta gde želimo da je postavimo
 - kliknuti desnim klikom da bi labela bila postavljena
- Ponoviti ovaj postupak za OUT, IN i GROUND linije. Voditi računa da se linija za masu označi kao GROUND zbog LVS provere. Posle dodavanja labela, leajaut izgleda kao na slici 9.16.



Slika 9.15: Upis labela za VDD (prozor gore levo)

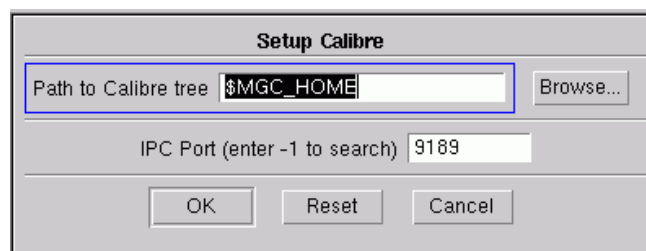


Slika 9.16: Izgled leajauta posle upisa svih labela

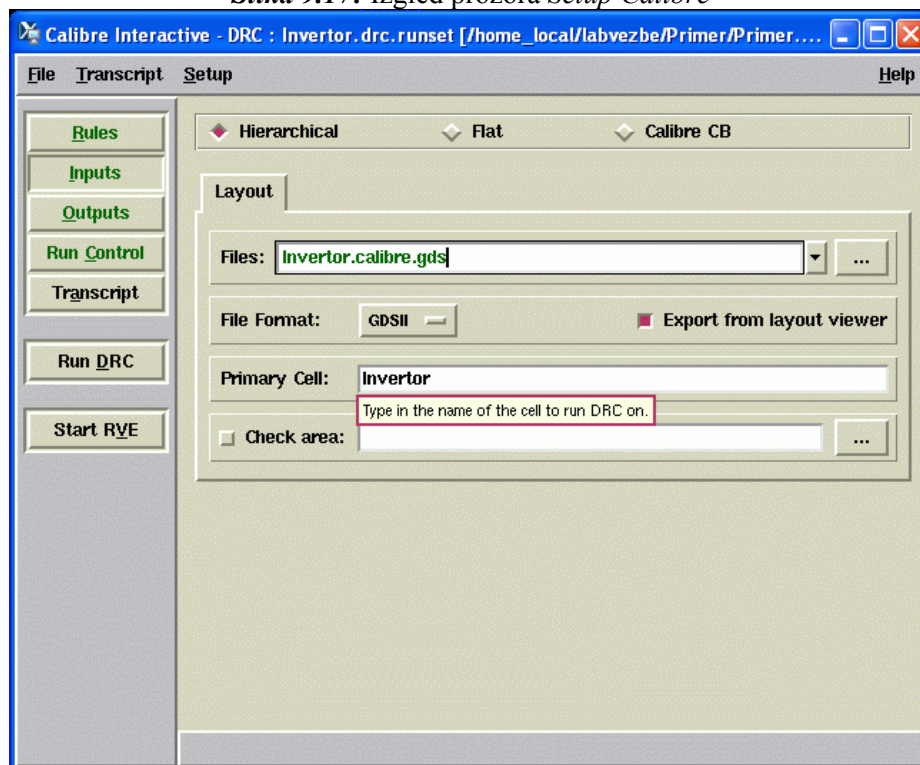
9.6 Alati za proveru pravila projektovanja (DRC i LVS)

Da bismo proverili da li su pri crtanju leajuta ispoštovana sva pravila projektovanja, najpre:

- kliknuti **Tools>Calibre>Run DRC**
- otvara se prozor **Setup Calibre** kao na slici 9.17.



Slika 9.17: Izgled prozora *Setup Calibre*

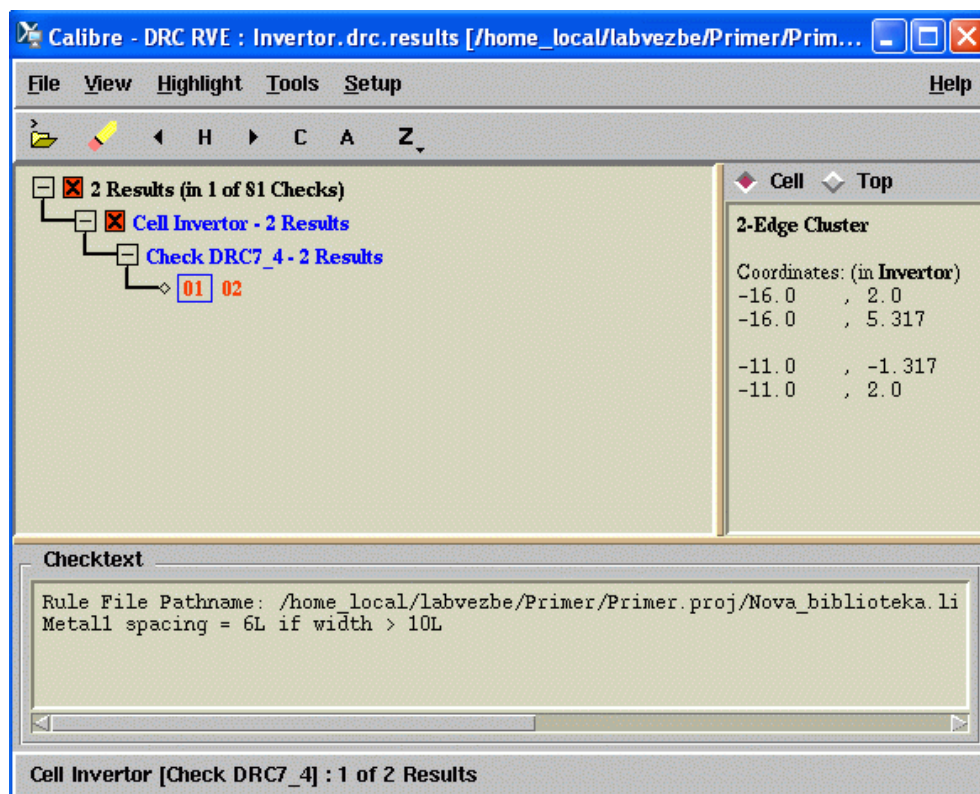


Slika 9.18: Izgled prozora *Calibre Interactive* pre početka DRC provere

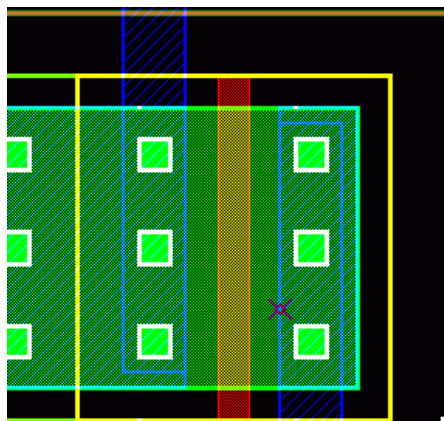
Treba učitati putanju fajlova koji su potrebni za proveru: /space/mentor/calibre. Zatim aktivirati proveru pritiskom na *Run DRC* sa leve strane prozora sa slike 9.18.

Kada se uradi provera našeg lejauta, dobija se prozor kao na slici 9.19. Alat za proveru javlja da su pronađene dve greške istog tipa (**2 Results (in 1 of 81 Checks)**). Alat proverava 81 pravilo.

U donjem delu prozora je opis greške koja je pronađena dva puta: **Metall spacing = 6L if width > 10L**. Treba ispoštovati pravilo da je rastojanje između dva metala 6λ , ukoliko je širina linije u metalu veća od 10λ . Date su i koordinate tačaka u kojim ovo pravilo nije ispoštovano, a odnosi se na metale sorsa i drejna PMOS-a, i kada se greška ispravi, tj. metal se suzi za 1λ sa svake strane, taj deo lejauta izgleda kao na slici 9.20.

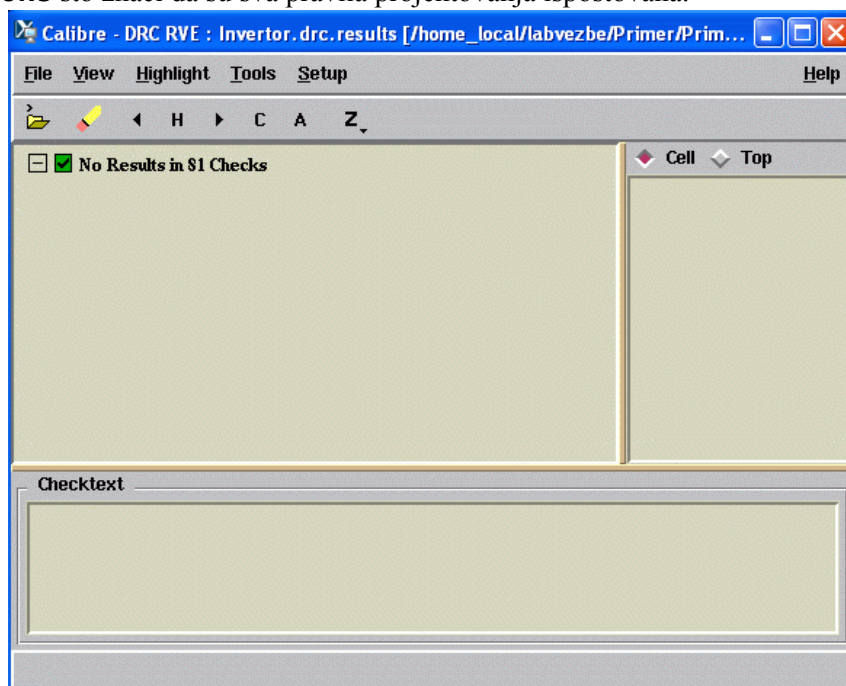


Slika 9.19: Izgled prozora Calibre po završetku DRC provere



Slika 9.20: Oznaka mesta na kome je prekršeno pravilo projektovanja simbolom 

Posle ispravljanja greške, treba ponovo aktivirati alat DRC, tako da se posle provere dobija prozor kao na slici 9.21. gde piše **No Results in 81 Checks** što znači da su sva pravila projektovanja ispoštovana.



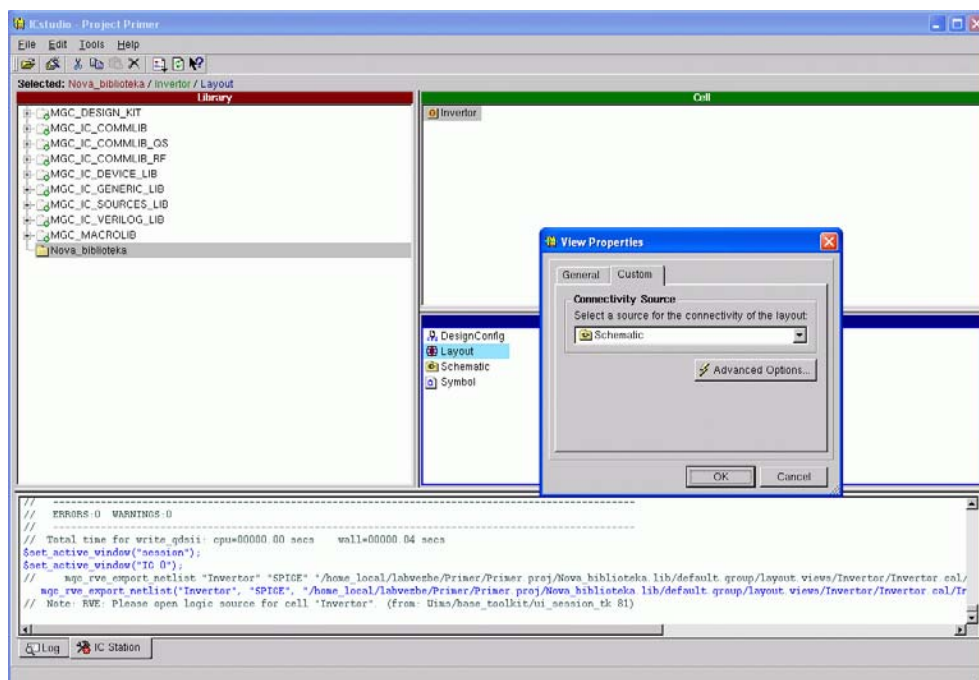
Slika 9.21: Izgled prozora Calibre posle uspešne DRC provere

Sledeći korak je poređenje šeme i lejauta invertora, tj. LVS.

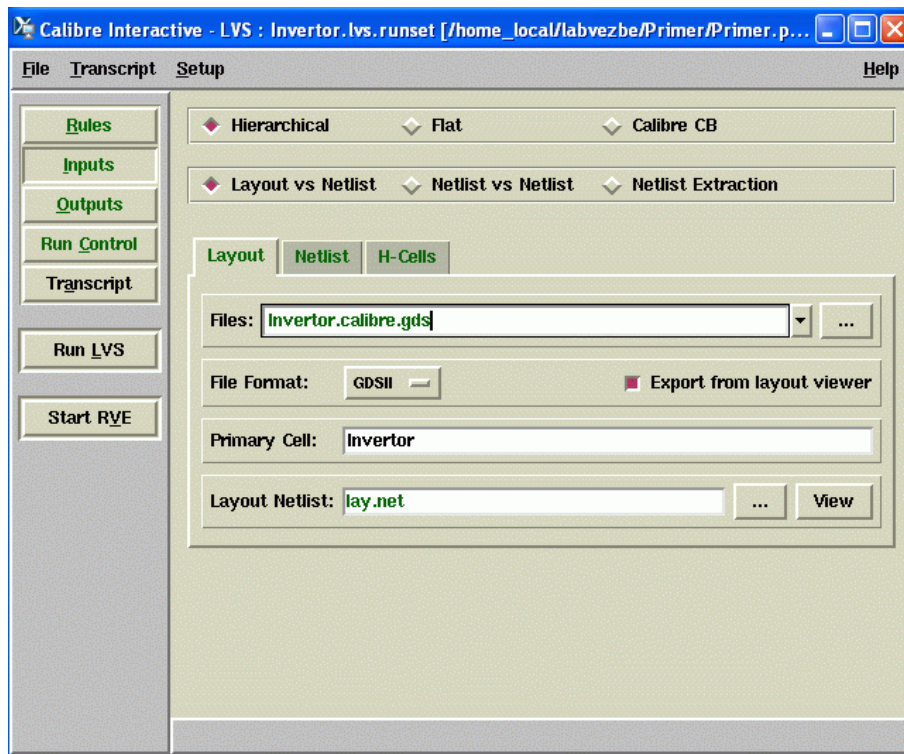
Najpre treba u *ICstudio* prozoru (slika 9.22) desnim klikom kliknuti na *Layout*, i za **Connectivity source** izabrati *Schematic*. Na taj način alat za poređenje ima za zadatak da poredi lejaut i šemu kola. LVS alat se pokreće:

- kliknuti **Tools>Calibre>Run LVS**
- otvara se prozor kao na slici 9.23. Treba imati u vidu da se fajlovi potrebni za LVS i DRC nalaze na istom mestu, pa ako su učitani za DRC nije ih potrebno ponovo učitavati, a ako nisu, ili ako je program pokrenut ponovo, treba ih ponovo učitati. Putanja je: `/space/mentor/calibre`.
- kliknuti **Run LVS**

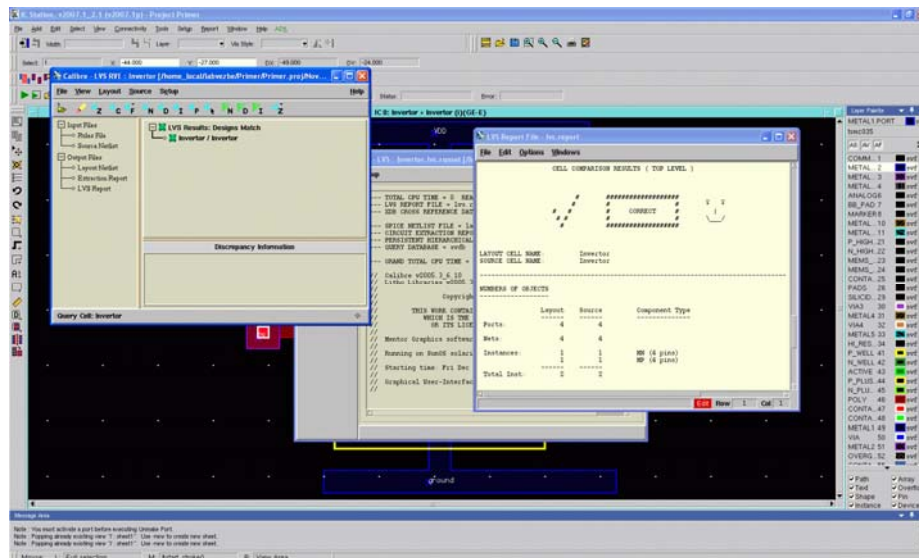
Posle izvršenja ovog poređenja, dobija se prozor kao na slici 9.24. u kome su rezultati poređenja. Ako se šema i lejaut slažu, dobija se odgovor sistema **CORRECT**, a ako to nije slučaj, specificirana je svaka greška ponaosob.



Slika 9.22: ICstudio prozor za pripremu LVS provere



Slika 9.23: Calibre prozor za početak LVS provere



Slika 9.24: Izgled prozora na kraju uspešne LVS provere

Vježba broj 10

Projektovanje ASIC kola na bazi standardnih ćelija upotrebom *ADK* alata

10.1 Cilj vežbe

Proučiti postupak projektovanja integrisanog kola pomoću ASIC Design Kit (ADK) alata koji distribuira *Mentor Graphics*.

10.2 Teorijska postavka vežbe

U ovoj vežbi je opisan postupak projektovanja integrisanog kola pomoću ASIC Design Kit (ADK) alata koji distribuira *Mentor Graphics*. ADK obuhvata alate za logičku sintezu, planiranje površine, razmeštaj ćelija i povezivanje.

Opis primene alata biće ilustrovan na projektovanju integrisanog kola koje obavlja funkciju brojača. Pri tome, koristiće se metod projektovanja zasnovan na primeni standardnih ćelija.

Projektovanje digitalnog integrisanog kola počinje opisom funkcije ili strukture kola na nekom od jezika za opis hardvera (HDL). S obzirom na prethodna iskustva, očekuje se da studenti u tu svrhu koriste VHDL. Iako se za ovu namenu može koristiti bilo koji editor, u ovoj vežbi koristiće se *ModelSim SE* alat. U

njemu kolo može da se opiše, da se proverí sintaksa, ali i da se verifikuje ponašanje kola simulacijom.

Nakon uspešnog kompajliranja i simulacije VHDL kodova, potrebno je izvršiti sintezu. Pod sintezom se podrazumeva prevođenje VHDL opisa u strukturu (logičku šemu) u kojoj se kao gradivni elementi kola pojavljuju logičke ćelije iz raspoložive biblioteke standardnih ćelija. Za sintezu se koristi alat *Leonardo Spectrum*. Ovaj program rezultat sinteze daje u obliku dva fajla. Jedan omogućava verifikaciju rada sintetizovanog kola, a drugi se koristi tokom faze fizičkog projektovanja u alatima za razmeštanje i povezivanje ćelija, odnosno projektovanja leajuta.

Leonardo Spectrum može da generiše fajl namenjen verifikaciji u više izlaznih formata, kao što su Verilog, VHDL, EDIF itd. Za razliku od nega, fajl namenjen fizičkom projektovanju generiše se samo u Verilog formatu. Ovaj fajl koriste alati za fizičko projektovanje, tako da nije neophodno da korisnik poznaje pravila Verilog jezika za opis hardvera.

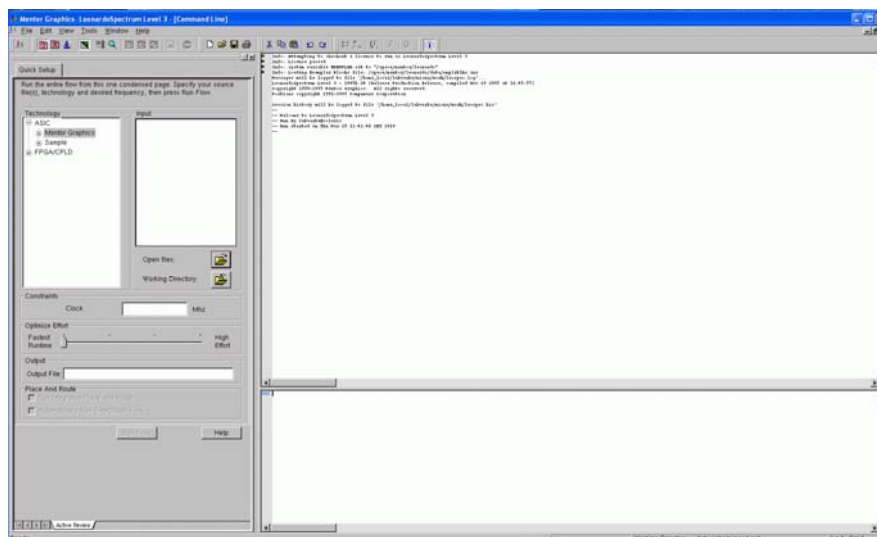
Nakon sinteze u alatu *Leonardo Spectrum* treba verifikovati dobijenu netlistu simulacijom. Tek posle uspešne verifikacije nastavlja se sa fizičkim projektovanjem. Priprema za generisanje leajuta sprovodi se kroz dva softverska alata. Alat *Design Architect* importuje Verilog netlistu i priprema je za dalju obradu. Zatim treba pokrenuti prelazi se u *IC Station*, pomoću koga se generiše leajut čipa.

10.3 *Leonardo Spectrum*

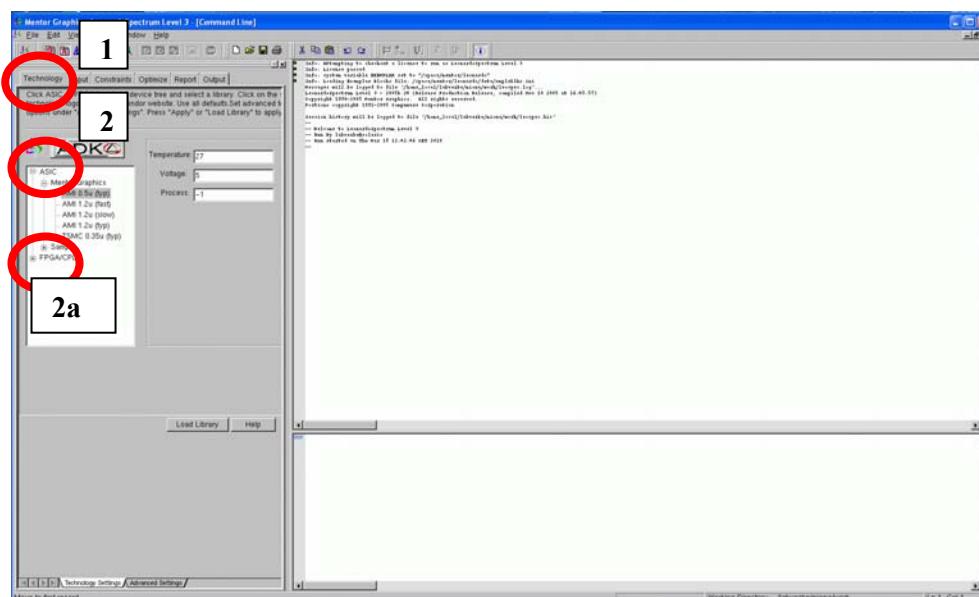
Alat za sintezu kola, uprošćeno govoreći, omogućuje prevođenje HDL opisa u hardver. Svojim algoritmima, alat za sintezu prepoznaje karakteristične delove kôda i na osnovu biblioteke tj. tehnologije za koju se kolo projektuje, povezuje module koji su u datoj tehnologiji dostupni. Zato je i potrebno, prilikom opisa kola, voditi računa o pravilima projektovanja za sintezu. *Leonardo Spectrum* je alat veoma jednostavan za korišćenje. U narednom delu teksta biće prikazano kako se koristi program za sintezu HDL koda.

Da bi se pokrenuo *Leonardo Spectrum*, u komandnoj liniji treba ukucati: `leonardo&`.

Pojaviće se dijalog za potvrdu licence, gde je potrebno selektovati treću opciju (*Leonardo Spectrum Level 3*), ako već nije automatski selektovana, i kliknuti na **OK**. Pojaviće se prozor prikazan na slici 10.1.



Slika 10.1: Izgled prozora koji se otvara kada se pokrene Leonardo Spectrum



Slika 10.2: Izgled prozora za učitavanje odgovarajuće tehnološke biblioteke

Projektovanje počinje izborom tehnologije u kojoj će se realizovati digitalno kolo. Zato je potrebno da se radnoj biblioteci priključi željena tehnološka biblioteka. U tom cilju treba kliknuti na opciju **Tools>Flow Tabs** iz glavnog menija, čime se otvara prozor kao na slici 10.2.

U okviru taba *Technology* koji omogućava izbor tehnologije (označen sa 1 na Slici 10.2) može se izabrati stil projektovanja, ASIC (2 na Slici 10.2) ili FPGA, (2a na Slici 10.2). Za izabrani stil bira se lista raspoloživih tehnologija, u okviru *Technology* tab-a, slika 10.2.

Kako je cilj ove vežbe projektovanje ASIC čipa, potrebno je selektovati neku od ASIC biblioteka. U radu sa studentskom verzijom paketa za projektovanje koristi se *ami05(typ)* tehnologija.

Osim tehnologije potrebno je definisati nominalnu temperaturu (27 °C) i odgovarajući napon napajanja (za izabranu tehnologiju preporučeni napon napajanja iznosi 5V).

Nakon definisanja stila projektovanja i tehnoloških parametara, potrebno je učitati sve potrebne biblioteke klikom na polje **Load Library** (dole desno). U radnom delu prozora (gore desno) program će prijaviti izveštaj o uspešno ili neuspešno učitanim bibliotekama. Plavim slovima naznačeni su parametri koje je korisnik zadao, kao što ilustruje slika 10.3.

```

Info: Attempting to checkout a license to run as LeonardoSpectrum Level 3
Info: License passed
Info: system variable EKEEMPLAR set to "/space/mentor/leonardo"
Info: Loading Exemplar Blocks file: /space/mentor/leonardo/data/xmplrblks.ini
Messages will be logged to file '/home_local/labvezbe/miona/work/leospec.log'...
LeonardoSpectrum Level 3 - 2005b.24 (Release Production Release, compiled Nov 10 2005 at 16:45:57)
Copyright 1990-2005 Mentor Graphics. All rights reserved.
Portions copyright 1991-2005 Synware Corporation

Session history will be logged to file '/home_local/labvezbe/miona/work/leospec.his'
--
-- Welcome to LeonardoSpectrum Level 3
-- Run By labvezbe@solaris
-- Run started on Thu Mar 18 13:38:13 CET 2010
--
->set temp 27
Info: setting temp to 27
27
->set voltage 5
Info: setting voltage to 5
5
->load_library ami05_typ
Reading library file '/space/mentor/leonardo/lib/ami05_typ.syn'...
Library version = 3.0 Release : Patch (a) : 03/13/04
Delays assume: Process=typical Temp= 27.0 C Voltage=5.00 V

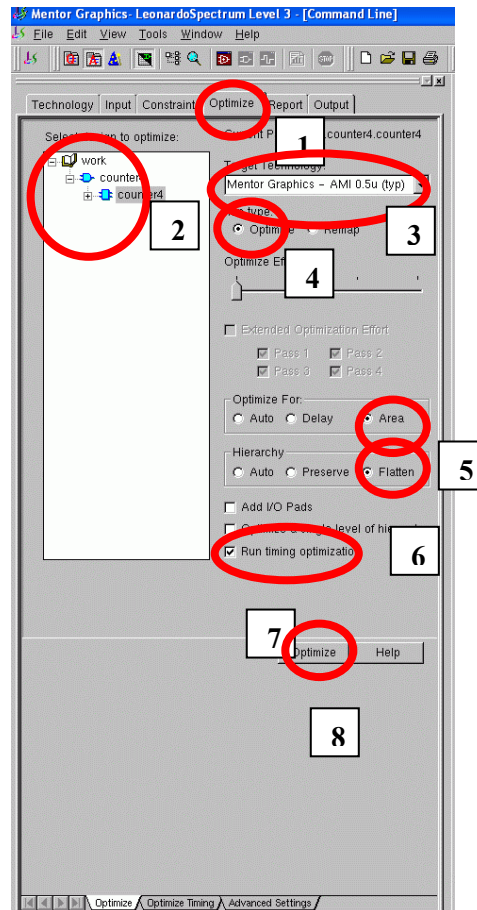
Info: setting encoding to auto
Info: Command 'load_library' finished successfully

```

Slika 10.3: Informacija o uspešno učitanoj biblioteci



Prelaskom na tab *Input* (korak 1 na Slici 10.4.), omogućava se učitavanje HDL opisa, što predstavlja sledeći korak u inicijalizaciji sinteze. Učitavanje fajlova za sintezu obavlja se izborom radnog direktorijuma (u kome se nalazi ulazni fajl) *Working directory* (korak 2 na Slici 10.4.) i pritiskom na dugme *Open files* (korak 3 na Slici 10.4.). Izabrani fajl priključuje se listi fajlova za sintezu, odnosno radnoj biblioteci. Opcija kodiranja stanja služi da se definiše način kodiranja stanja ako projekat koji se sintetizuje sadrži konačni automat. Podrazumevano (default) se postavlja na automatski način kodiranja, a opciono se može postaviti i drugi način kodiranja (korak 4 na Slici 10.4.). Treba zadržati selektovana polja *Resource Sharing*, *Run Elaborate* i *Run Pre-Optimization*, a ove opcije se mogu dodatno podešavati biranjem tabova u dnu. Nakon svih potrebnih podešavanja potrebno je učitati željene fajlove, pritiskom na *Read* (korak 5 na Slici 10.4.).



Slika 10.6: Postavljanje parametara optimizacije

Selekcijom tab-a *Optimize* označenog sa 1 u komandnom delu prozora na Slici 10.6, prelazi se na naredni korak. Kao rezultat prethodnih koraka, pojavljuje se naziv ulaznog fajla *counter4* u radnoj biblioteci (2 na Slici 10.6) i *AMI 05 typ* kao *Target Technology* (3 na Slici 10.6). Potrebno je postaviti ostale parametre prema slici 10.6 i to: *Run Type* kao *Optimize* (4 na Slici 10.6)

Optimize For kao *Area* (5 na Slici 10.6)

Hierarchy kao *Flatten* (6 na Slici 10.6)

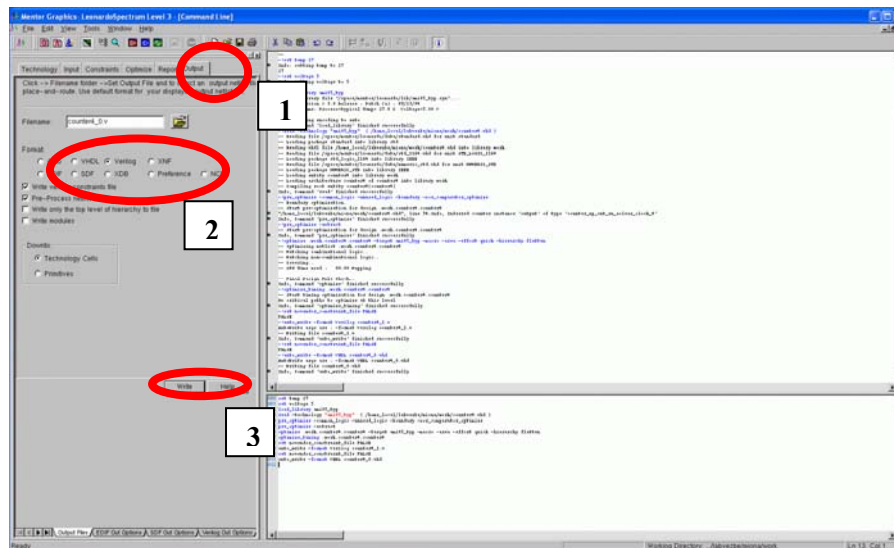
Selektovati *Run Timing Optimization* (7 na Slici 10.6) i pokrenuti optimizaciju izborom *Optimize* (8 na Slici 10.6).

U okviru tab-a *Report* u komandnom prozoru definiše se ime izlaznog tekstualnog fajla vezanog za rezultate sinteze. Potrebno je navesti ime fajla i pritisnuti **report area**. Ovaj fajl služi samo za prikazivanje izveštaja o sintezi a ne koristi se u narednim fazama projektovanja.

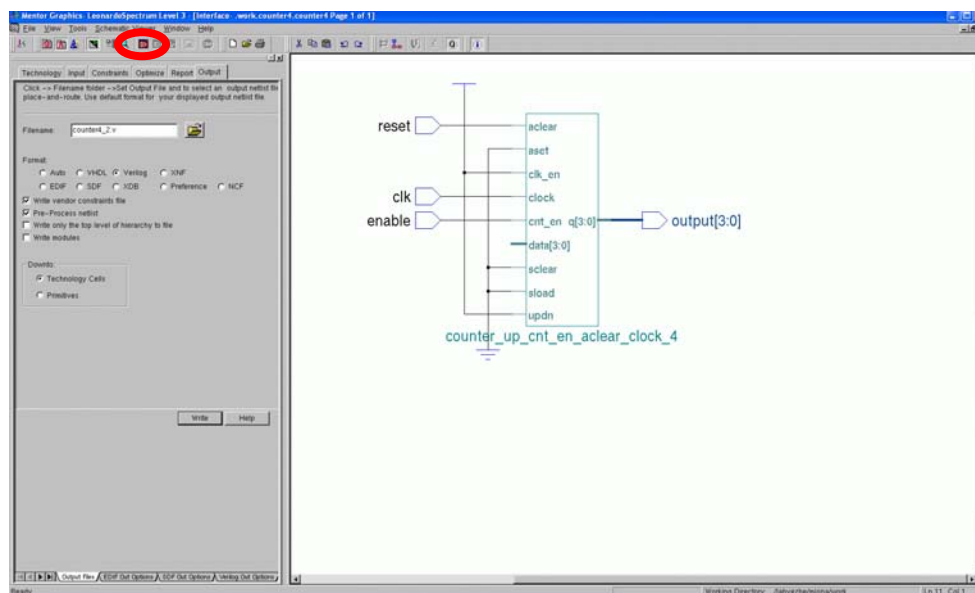
Fajlovi koji su neophodni u daljem postupku projektovanja jesu fajl za verifikaciju posle sinteze (u VHDL formatu) i fajl na osnovu koga se generiše leajut (u Verilog formatu). Oni se specificiraju u okviru tab-a *Output*, u komandnom prozoru na slici 10.7. označenog sa 1. Generisanje ovih fajlova predstavlja poslednji korak sinteze.

Potrebno je definisati i imenovati izlazni fajl koji treba sačuvati kao Verilog i/ili kao VHDL fajl. Format izlaznih fajlova definiše se u polju menija označenog sa 2 na Slici 10.7.

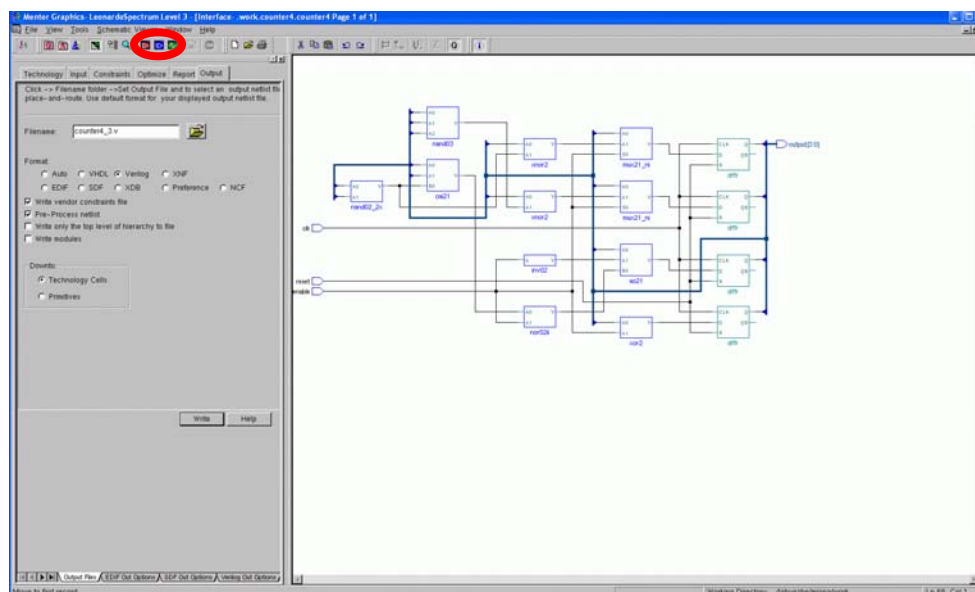
Verilog fajl će se kasnije koristiti za razmeštaj i povezivanje standardnih ćelija iz biblioteke korišćene tehnologije, dok će se VHDL fajl koristiti za simuliranje projekta nakon sinteze. Selekcijom polja **Write** (označeno sa 3 na Slici 10.7) ovi fajlovi biće upisani ispod radnog direktorijuma.



Slika 10.7: Postavljanje parametara za generisanje izlaznih fajlova nakon sinteze



Slika 10.8: Sintetizovana šema brojača, prikaz na RTL nivou



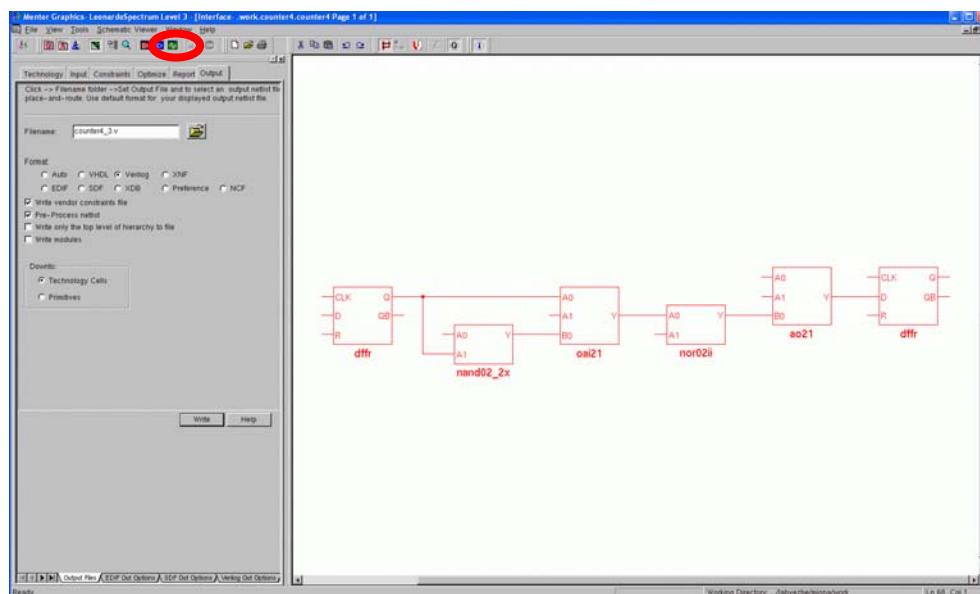
Slika 10.9: Sintetizovani brojač - šema brojača na nivou tehnoloških ćelija

Klikom na ikonicu  sa glavnog menija (*View RTL schematic*) moguće je videti generalnu logičku šemu brojača, slika 10.8.

Klikom na ikonicu  sa glavnog menija (*View technology schematic*) moguće je sagledati šeme date na nivou tehnoloških ćelija, slika 10.9. Može se primetiti da je struktura brojača definisana preko blokova koji se nalaze u biblioteci ćelija za tehnologiju AMI05. To su blokovi sa nazivima funkcija (nand03, oa21, xor2,...) ali i sa definisanim fanoutom (nand02_x2, inv02).

Klikom na ikonicu  sa glavnog menija (*View critical path schematic*) moguće je sagledati deo kola gde je najveće kašnjenje signala, slika 10.10.

Prilikom sinteze složenih projekata, opisanih u više hijerarhijskih nivoa, potrebno je izvršiti sintezu svih hijerarhijski nižih modula, a zatim sintetizovati kolo na najvišem nivou, *top-level* opis u koji će se učitati svi ostali makro moduli sa nižeg nivoa.



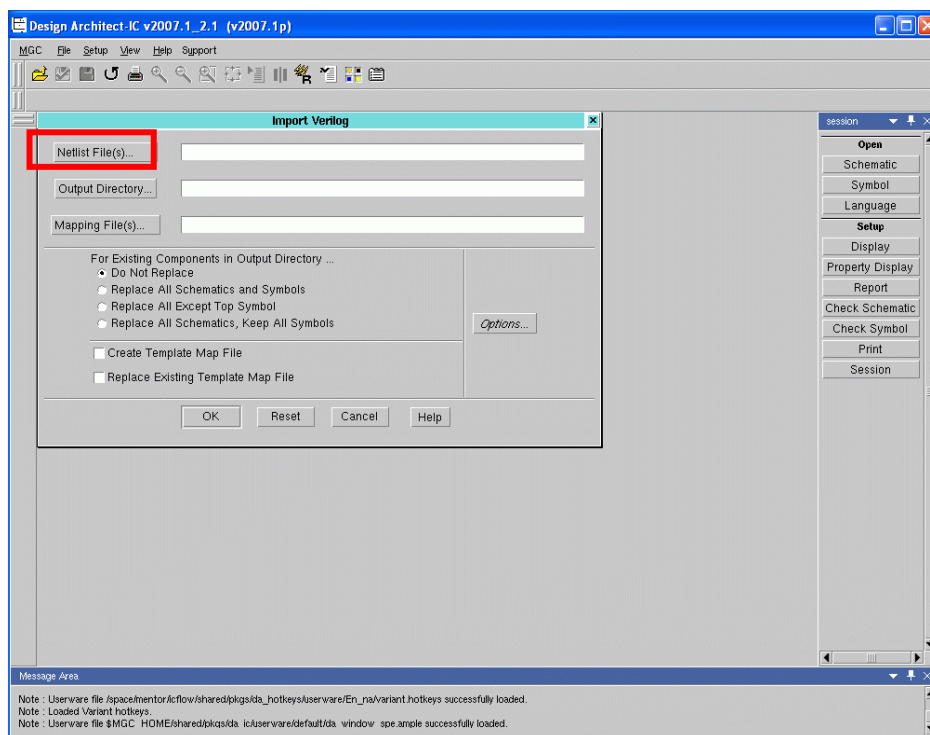
Slika 10.10: Sagledavanje dela kola sa najvećim kašnjenjem

10.4 Verifikacija posle sinteze - *Design Architect*

Pre fizičkog projektovanja potrebno je verifikovati rezultat sinteze simulacijom i pripremiti fajlove za generisanje lejauta. U tu svrhu koristi se *Design Architect* alat.

Radi se o alatu koji služi za kreiranje, modifikaciju i verifikaciju električne i/ili logičke šeme. Ovaj program podržava i rad sa analognim kolima.

Od svih mogućnosti koje ovaj program nudi, kao što su simulacija analognih i digitalnih kola, crtanje šema, itd. biće izdvojena ona koja je nama od interesa u ovoj vežbi, a to je sinteza na tehnološkom nivou na bazi standardnih ćelija. Kao ulazni fajlovi koriste se opisi dobijeni iz alata *Leonardo Spectrum*, odnosno fajlovi u VHDL i Verilog formatu. S obzirom da se postupak verifikacije ne razlikuje od onoga pre sinteze, ovde će biti opisan deo koji se odnosi na pripremu fajlova za generisanje lejauta.



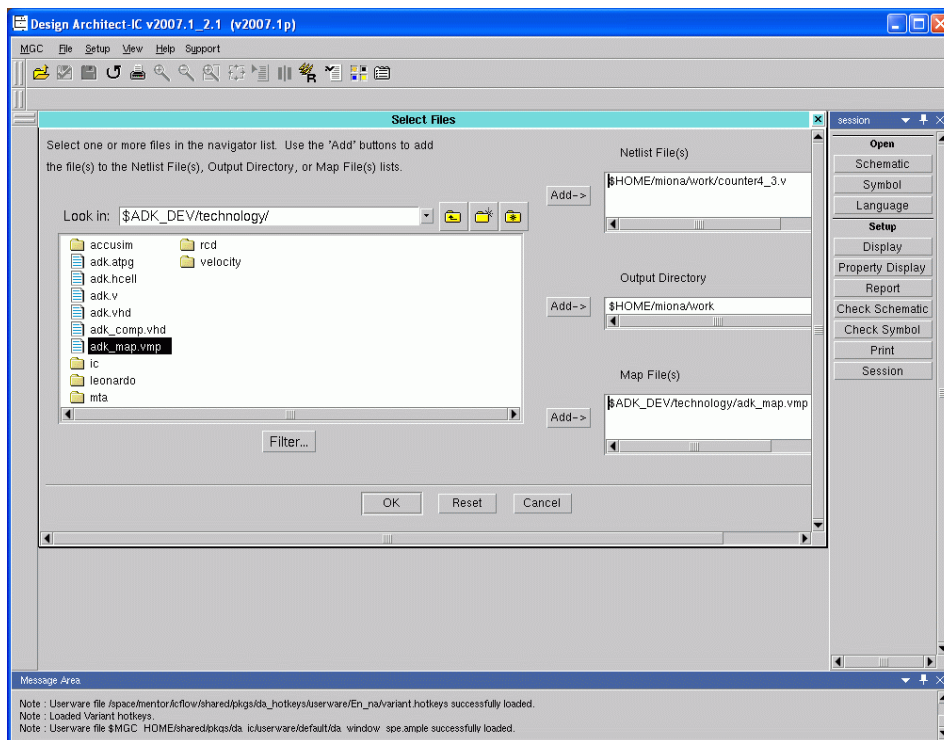
Slika 10.11: Učitavanje Verilog fajla u *Design Architect*

Da bi se pokrenuo program *Design Architect*, u komandnoj liniji treba ukucati: `da_ic&`.

Prvi korak je unošenje Verilog fajla dobijenog sintezom. Kada se klikne **File>ImportVerilog** dobija se dijalog prikazan na slici 10.11.

Selekcijom *Netlist File(s)* sa slike 10.11. pojaviće se novi dijalog, dat na slici 10.12. Treba učitati Verilog fajl dobijen sintezom i odrediti izlazni direktorijum u koji će se smeštati podaci za generisanje leajuta. Najvažnije je ispravno ukazati na *map* fajl, koji sadrži podatke o načinu mapiranja ćelija, čije je mesto na hard disku u okviru direktorijuma gde je instaliran *Mentor Graphics* (*ADK_DEV/technology/adk_map.vmp*).

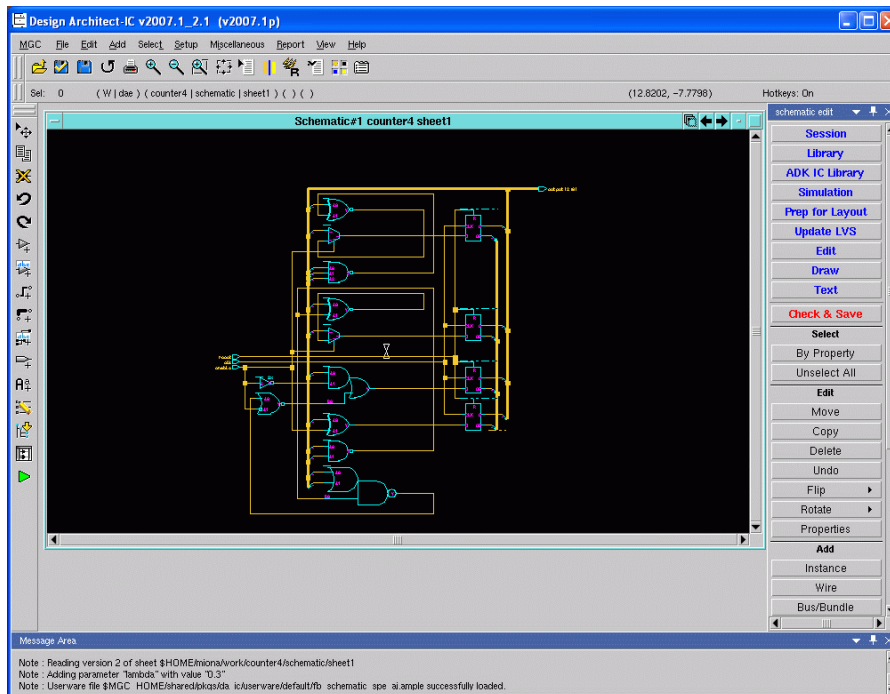
Nakon uspešnog učitavanja, šema projekta neće još uvek biti prezentovana. Potrebno je šemu otvoriti selektovanjem **File > Open> Schematic** i onda izabrati putanju fajla koji se nalazi u radnom direktorijumu, definisanom u prethodnom koraku.



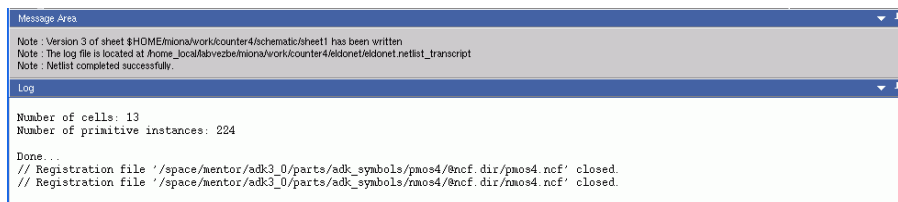
Slika 10.12: Učitavanje potrebnih fajlova u *Design Architect*

Ako je sve prilikom učitavanja Verilog fajla bilo dobro podešeno, pojaviće se šema kao na slici 10.13.

Najpre je potrebno proveriti da li šema zadovoljava sve električne i logičke norme. Provera se startuje klikom na polje sa menija sa desne strane **Check&Save**. Ako je testiranje uspešno izvršeno, pritiskom na polje sa menija sa desne strane **Prep for Layout** počinje generisanje fajlova za *layout*. Nakon toga aktivirati polje **Update LVS**. Posle uspešnog generisanja net liste za razmeštaj i povezivanje dobija se obaveštenje kao na slici 10.14.



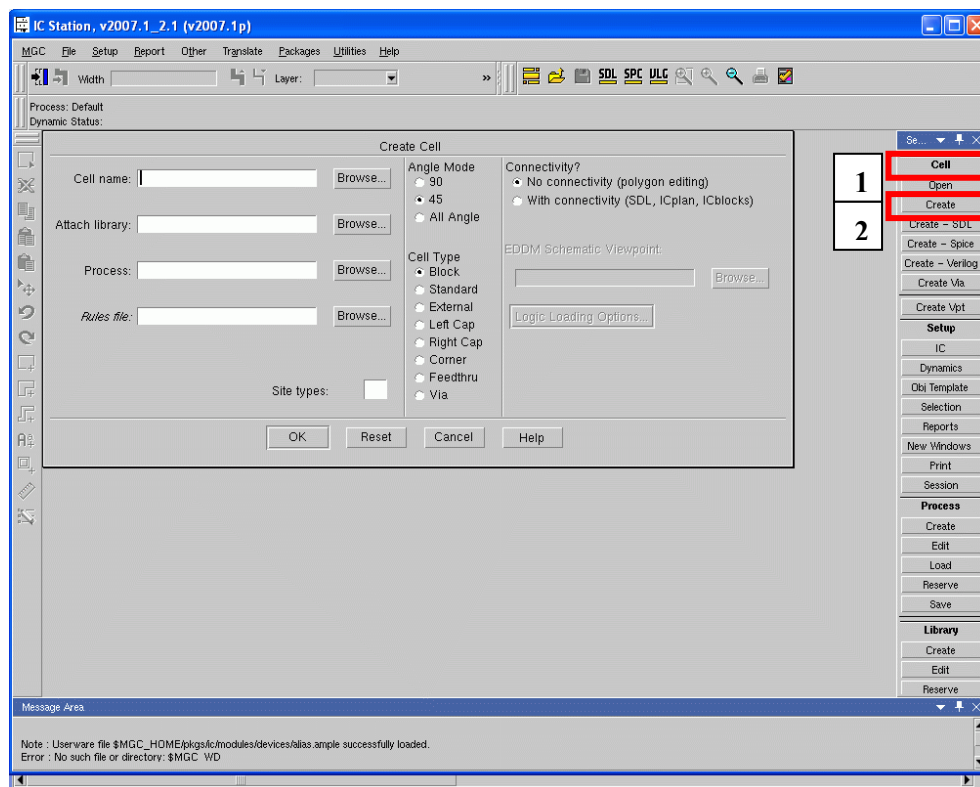
Slika 10.13: Šema brojača counter4 u Design Architect-u



Slika 10.14: Obaveštenje o uspešnom generisanju fajlova za layout

10.5 IC Station

IC Station predstavlja poslednji alat vezan za projektovanje integrisanih kola u okviru ASIC Design Kit (ADK). Njegova namena jeste da na osnovu netliste generiše leajut projektovane ćelije (ili čipa). Generisanje leajuta obavlja se u više koraka, koji će u narednom tekstu biti objašnjeni.



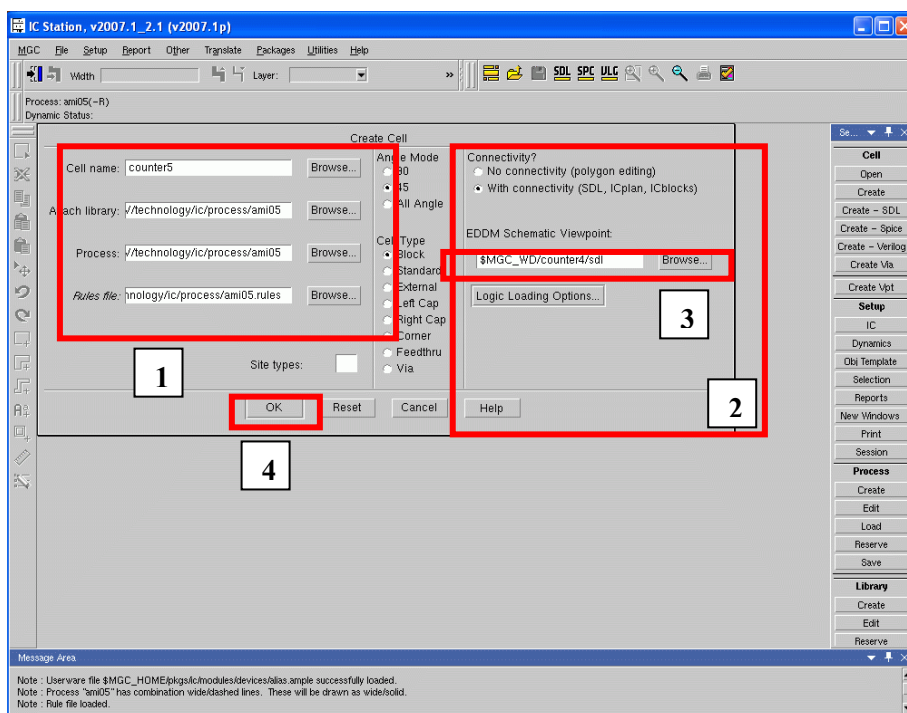
Slika 10.15: Dijalog za kreiranje nove ćelije

Da bi se pokrenuo *IC Station*, u komandnoj liniji treba ukucati: `ic&`. Otvara se novi prozor sa radnom površinom, i da bismo od projektovanog brojača kreirali novu ćeliju, treba kliknuti na **Create** (označeno sa 2 na Slici 10.15) u okviru **Cell** palete (označene sa 1 na Slici 10.15) sa desne strane menija. Pojavljuje se dijalog za kreiranje ćelije, dat na slici 10.15.

Dijalog je potrebno popuniti na način prikazan na slici 10.16. Ime ćelije se bira proizvoljno – ovde je izabrano counter5. Biblioteke i procesi smešteni su u

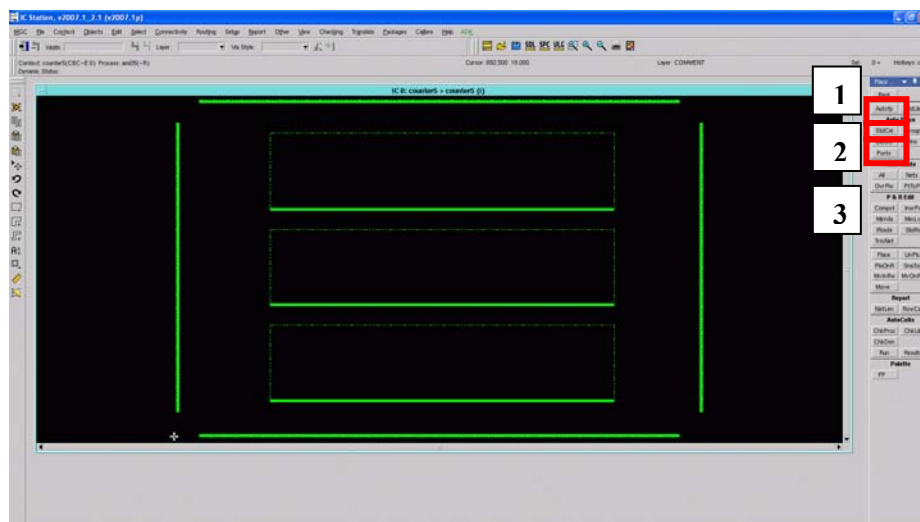
direktorijumu *ADK_DEV/technology/ic/process*, pri čemu se bira fajl *ami05* za tehnologiju koju smo izabrali na početku (polja označena sa 1 na Slici 10.16).

U delu vezanom za *viewpoint* (desna strana dijaloga, označena sa 2 na Slici 10.16) jako je važno navesti *sdl* (polje označeno sa 3 na Slici 10.16). Ovaj naziv ukazuje na *Schematic Driven Layout* što znači da će lejaut biti generisan na osnovu učitane šeme. Put se odnosi na direktorijum na koji će se *IC* pozivati prilikom generisanja lejauta, a koji je kreiran u prethodnim koracima. Nakon kreiranja ćelije pritiskom na **OK** (polje 4 na Slici 10.16) otvoriće se prazan prozor. Sada je potrebno aktivirati *ADK_edit* paletu koja se nalazi na dnu palete alata sa desne strane.

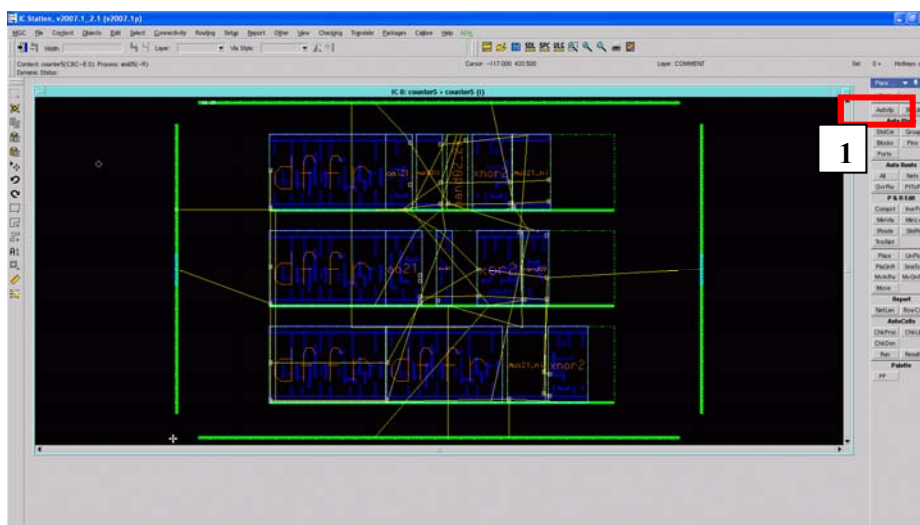


Slika 10.16: Popunjavanje dijaloga za kreiranje nove ćelije

Nakon aktiviranja *ADK_edit* palete, koraci koji slede su potpuno automatizovani. Najpre je potrebno napraviti okvirni plan površine na koju će se smeštati standardne ili makro-ćelije. Selektovanjem *Autofp* sa Place & Route palete (**P&R**), polje 1 na Slici 10.17, počinje automatsko generisanje plana površine. Nakon toga se na ekranu prikazuje polje sa slike 10.17.



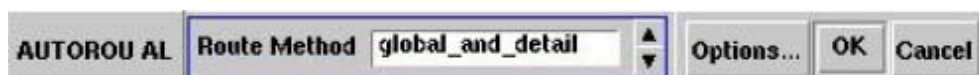
Slika 10.17: Generisan floorplan makro-ćelije



Slika 10.18: Izgled lejauta sa razmeštenim ćelijama i portovima (pre povezivanja)

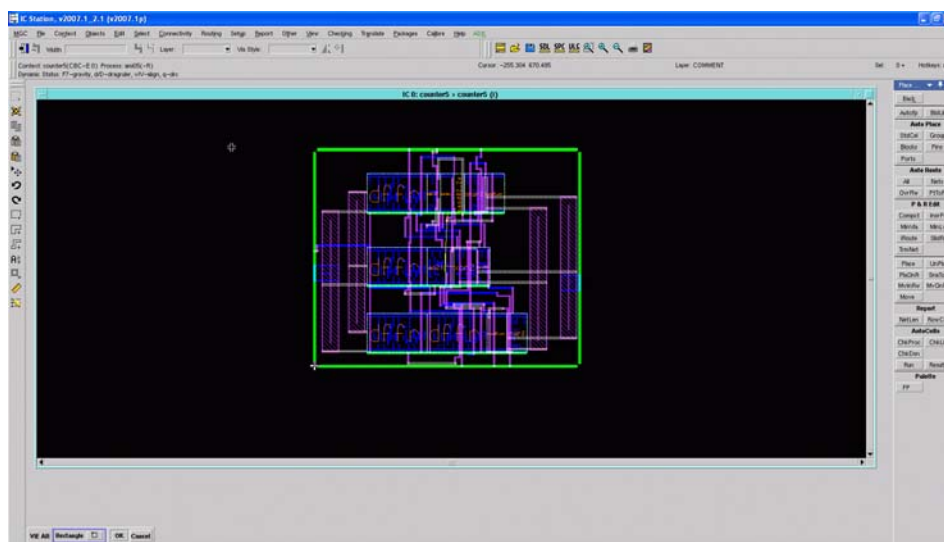
Sada je potrebno učitati standardne ćelije. Ovaj korak se izvršava klikom na **StdCell**, sa iste palete, polje 2 na Slici 10.17. Pored toga potrebno je definisati i portove klikom na **Ports**, polje 3 na Slici 10.17. Slika 10.18 prikazuje popunjen prozor standardnim ćelijama sa portovima.

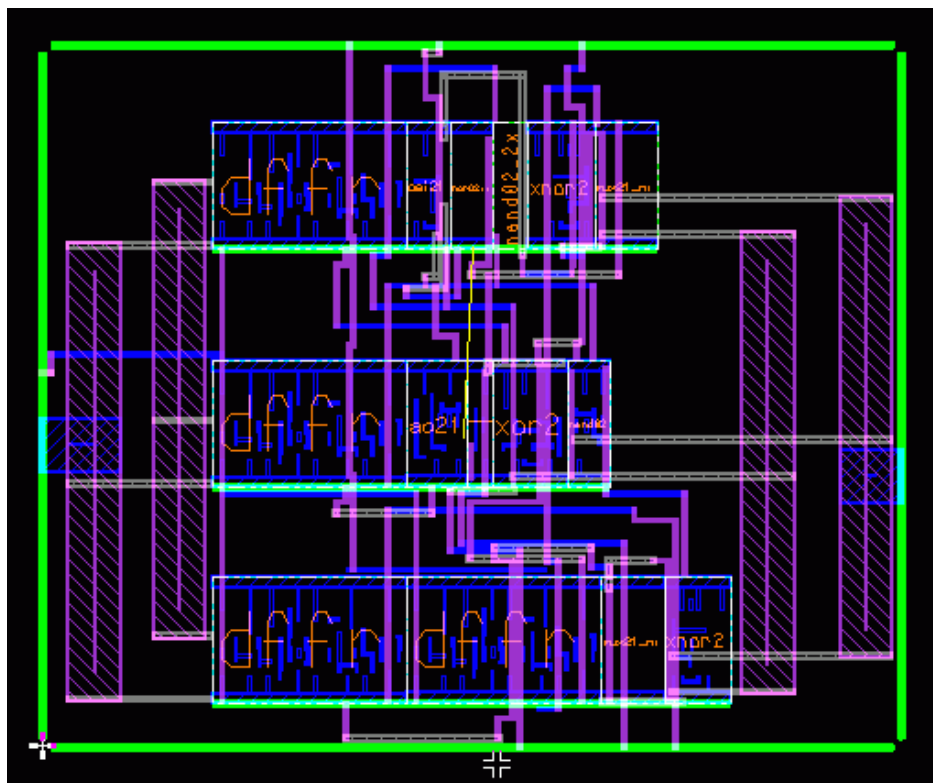
Sledeći korak jeste fizičko povezivanje komponenti. Potrebno je kliknuti na polje **All** u **AutoRoute** paleti, označeni sa 1 na Slici 10.18. Nakon toga, pojaviće se dijalog sa slike 10.19, i treba kliknuti **OK**.



Slika 10.19: Dijalog za povezivanje komponenti

Nakon povezivanja dobija se *layout* ćelije kao na slici 10.20.





Slika 10.20: Layout projektovane ćelije

Prilog 1: PSpice - numerički sufixi

Sufiks	Značenje	Ekvivalentni metrički prefiks
F	10^{-15}	femto
P	10^{-12}	piko
N	10^{-9}	nano
U	10^{-6}	mikro
M	10^{-3}	mili
K	10^{+3}	kilo
MEG	10^{+6}	mega
G	10^{+9}	giga
T	10^{+12}	tera

Prilog 2: PSpice - elementi i komande

Korišćene oznake:

Oznaka	Primer	Značenje
(...)	(model)	komentar
[...]	[vrednost]	opciona stavka
[...]*	[<vrednost> ,]*	stavka koja se pojavljuje nula ili više puta
< ... >	<ime>	obavezna stavka
< ... >*	<čvor>*	stavka koja se pojavljuje jednom ili više puta
" ... "	"ime datoteke"	ime datoteke (uključujući i ekstenziju)

1. B - GaAsFET

B<ime> <(drejn) čvor> <(gejt) čvor> <(sors) čvor>
 + <ime modelske kartice> [vrednost parametra area]
 .MODEL <ime modelske kartice> GASFET [parametri modela]

2. C - kondenzator

C<ime> <(+) čvor> <(-) čvor> [ime modelske kartice]
 + <vrednost> [IC=<početna vrednost napona>]
 .MODEL <ime modelske kartice> CAP [parametri modela]

3. D - dioda

D<ime> <(+) čvor> <(-) čvor> <ime modelske kartice>
 + [vrednost parametra area]
 .MODEL <ime modelske kartice> D [parametri modela]

4. E - naponski generator kontrolisan naponom (NGKN)

E<ime> <(+) čvor> <(-) čvor> <(+) kontrolišući čvor>
 + <(-) kontrolišući čvor> <pojačanje>

E<ime> <(+) čvor> <(-) čvor> POLY(<vrednost>)
 + <<(+) kontrolišući čvor> <(-) kontrolišući čvor>>*<vrednost koeficijenta polinoma>*

E<ime> <(+) čvor> <(-) čvor> VALUE = { <izraz> }

E<ime> <(+) čvor> <(-) čvor> TABLE { <izraz> } =
 + <<vrednost ulaza> , <vrednost izlaza> >*

E<ime> <(+) čvor> <(-) čvor> LAPLACE { <izraz> } =
 + { <transformacija> }

E<ime> <(+) čvor> <(-) čvor> FREQ { <izraz> } =
 + <<frekvencija> , <amplituda> , <faza> >*

5. F - strujni generator kontrolisan strujom (SGKS)

F<ime> <(+) čvor> <(-) čvor>
 + <ime kontrolišućeg V elementa> <pojačanje>

F<ime> <(+) čvor> <(-) čvor> POLY(<vrednost>)
 + <ime kontrolišućeg V elementa>*<vrednost koeficijenta polinoma>*

6. G - strujni generator kontrolisan naponom (SGKN)

G<ime> <(+) čvor> <(-) čvor> <(+) kontrolišući čvor>
 + <(-) kontrolišući čvor> <prenosna odvodnost>

G<ime> <(+) čvor> <(-) čvor> POLY(<vrednost>)
 + <<(+) kontrolišući čvor> <(-) kontrolišući čvor>>*<vrednost koeficijenta polinoma>*

G<ime> <(+) čvor> <(-) čvor> VALUE = { <izraz> }

G<ime> <(+) čvor> <(-) čvor> TABLE { <izraz> } =
 + <<vrednost ulaza> , <vrednost izlaza> >*

G<ime> <(+) čvor> <(-) čvor> LAPLACE { <izraz> } =
 + { <transformacija> }

G<ime> <(+) čvor> <(-) čvor> FREQ { <izraz> } =
 + <<frekvencija> , <amplituda> , <faza> >*

7. H - naponski generator kontrolisan strujom (NGKS)

H<ime> <(+) čvor> <(-) čvor>
 + <ime kontrolišućeg V elementa>
 + <prenosna otpornost>
 H<ime> <(+) čvor> <(-) čvor> POLY(<vrednost>)
 + <ime kontrolišućeg V elementa>*<vrednost koeficijenta polinoma>*

8. I - nezavisni strujni izvor i pobudni generator

I<ime> <(+) čvor> <(-) čvor> [DC <vrednost>]
 + [AC <amplituda> [faza]]
 + [talasni oblik u vremenskom domenu]

9. J - JFET

J<ime> <(drejn)čvor> <(gejt)čvor> <(sors)čvor>
 + <ime modelske kartice> [vrednost parametra area]
 .MODEL <ime modelske kartice> NJF [parametri modela]
 .MODEL <ime modelske kartice> PJF [parametri modela]

10. K - spregnute induktivnosti

K<ime> L<ime kalema> < L<ime kalema> >*<vrednost medjusobne induktivnosti>
 + <vrednost medjusobne induktivnosti>
 K<ime> < L<ime kalema> >*<vrednost medjusobne induktivnosti>
 + <ime modelske kartice> [vrednost parametra size]
 .MODEL <ime modelske kartice> CORE [parametri modela]

11. L - kalem

L<ime> <(+) čvor> <(-) čvor> [ime modelske kartice]
 + <vrednost> [IC=<početna vrednost struje>]
 .MODEL <ime modelske kartice> IND [parametri modela]

12. M - MOSFET

M<ime> <(drejn)čvor> <(gejt)čvor> <(sors)čvor>
 + <(balk)čvor> <ime modelske kartice> [L=<vrednost>]
 + [W=<vrednost>] [AD=<vrednost>] [AS=<vrednost>]
 + [PD=<vrednost>] [PS=<vrednost>] [NRD=<vrednost>]

```

+      [NRS=<vrednost>] [NRG=<vrednost>]
+      [NRB=<vrednost>] [M=<vrednost>]
.MODEL <ime modelske kartice> NMOS [parametri modela]
.MODEL <ime modelske kartice> PMOS [parametri modela]

```

13. N - D/A konvertor (interfejs)

```

N<ime>    <izlazni analogni čvor> <(Vss)čvor>
+         <(Vdd)čvor> <ime modelske kartice>
+         DGTLNET = <ulazni digitalni čvor>
+         <ime U/I modela> [IS = <inicijalno stanje>]

N<ime>    <izlazni analogni čvor> <(Vss)čvor>
+         <(Vdd)čvor> <ime modelske kartice>
+         [ SIGNAME = <ime ulaznog digitalnog čvora> ]
+         [ IS = <inicijalno stanje> ]

.MODEL    <ime modelske kartice> DINPUT [parametri modela]

```

14. O - A/D konvertor (interfejs)

```

O<ime>    <ulazni analogni čvor> <referentni čvor>
+         <ime modelske kartice>
+         DGTLNET = <izlazni digitalni čvor>
+         <ime U/I modela>

O<ime>    <ulazni analogni čvor> <referentni čvor>
+         <ime modelske kartice>
+         [ SIGNAME = <izlazni digitalni čvor> ]

.MODEL    <ime modelske kartice> DOUTPUT [parametri modela]

```

15. Q - bipolarni tranzistor

```

Q<ime>    <(kolektor) čvor> <(baza) čvor> <(emitor) čvor>
+         [(supstrat) čvor] <ime modelske kartice>
+         [vrednost parametra area]
.MODEL    <ime modelske kartice> NPN [parametri modela]
.MODEL    <ime modelske kartice> PNP [parametri modela]
.MODEL    <ime modelske kartice> LPNP [parametri modela]

```

16. R - otpornik

R<ime> <(+) čvor> <(-) čvor> [ime modelske kartice]
 + <vrednost> [TC = <TC1> [, <TC2>]]
 .MODEL <ime modelske kartice> RES [parametri modela]

17. S - naponom kontrolisan prekidač

S<ime> <(+) čvor prekidača> <(-) čvor prekidača>
 + <(+) kontrolišući čvor> <(-) kontrolišući čvor>
 + <ime modelske kartice>
 .MODEL <ime modelske kartice> VSWITCH [parametri modela]

18. T - vod

T<ime> <(+) čvor A pristupa> <(-) čvor A pristupa>
 + <(+) čvor B pristupa> <(-) čvor B pristupa>
 + Z0 = <vrednost> [TD = <vrednost>]
 + [F = <vrednost> [NL = <vrednost>]]

19. U - digitalni element

U<ime> <tip elementa> ([vrednost parametra]*)
 + <čvor>* <ime vremenskog modela> <ime U/I modela>
 U<ime> STIM (<broj izlaza>, <format>) <čvor>*
 + <ime U/I modela> [TIMESTEP=<vrednost>]
 + <(opis talasnog oblika)>

U/I model :

.MODEL <ime modelske kartice> UIO [parametri modela]

Vremenski modeli:

.MODEL <ime modelske kartice> UGATE [parametri modela]
 .MODEL <ime modelske kartice> UTGATE [parametri modela]
 .MODEL <ime modelske kartice> UEFF [parametri modela]
 .MODEL <ime modelske kartice> UGFF [parametri modela]
 .MODEL <ime modelske kartice> UDLY [parametri modela]
 .MODEL <ime modelske kartice> UWDTH [parametri modela]
 .MODEL <ime modelske kartice> USUHDT [parametri modela]

20. V - nezavisni naponski izvor i pobudni generator

V<ime> <(+) čvor> <(-) čvor> [[DC] <vrednost>]
 + [AC <amplituda> [faza]]
 + [talasni oblik u vremenskom domenu]

21. W - strujom kontrolisan prekidač

W<ime> <(+) čvor prekidača> <(-) čvor prekidača>
 + <ime kontrolišućeg V elementa>
 + <ime modelske kartice>
 .MODEL <ime modelske kartice> ISWITCH [parametri modela]

22. X - poziv potkola

X<ime> [čvor]* <ime potkola>
 + [PARAMS: < <ime> = <vrednost> >*]
 + [TEXT: < <ime> = <tekstualna vrednost> >*]

23. .AC - AC analiza

.AC [LIN] [OCT] [DEC] <vrednost>
 + <početna vrednost frekvencije>
 + <završna vrednost frekvencije>

24. .DC - DC analiza

.DC [LIN] <ime varirane promenljive>
 + <početna vrednost> <završna vrednost>
 + <vrednost koraka>
 + [višestruka specifikacija]
 .DC [OCT] [DEC] <ime varirane promenljive>
 + <početna vrednost> <završna vrednost>
 + <vrednost broja tačaka>
 + [višestruka specifikacija]
 .DC <ime varirane promenljive> LIST <vrednost>*

 + [višestruka specifikacija]

25. .DISTRIBUTION - definicija funkcije raspodele

.DISTRIBUTION <ime> (< <devijacija> <verovatnoća> >)*

26. .END - kraj opisa kola

.END

27. .ENDS - kraj definicije potkola

.ENDS [ime potkola]

28. .FOUR - Furijeova analiza

.FOUR <vrednost frekvencije osnovnog harmonika>
+ <promenljiva>*

29. .FUNC - definicija funkcije

.FUNC <ime> ([<argument>*]) <telo funkcije>

30. .IC - zadavanje početnih uslova

.IC < V(<čvor> [, <čvor>]) = <vrednost> >*

31. .INC - priključivanje datoteke

.INC <"(ime priključene datoteke)">

32. .LIB - zadavanje bibliotečkih datoteka

.LIB ["(ime bibliotečke datoteke)"]

33. .LOADBIAS - datoteka za učitavanje početnih uslova

.LOADBIAS <"(ime datoteke sa početnim uslovima)">

34. .MC - Monte Karlo analiza

.MC <vrednost broja prolaza> <analiza>
+ <izlazna promenljiva> <funkcija> [opcija]*
+ [SEED = (vrednost)]

35. .MODEL - definisanje modelske kartice

.MODEL <ime modelske kartice>
+ [AKO: <ime referentnog modela>] <tip modela>
+ ([<ime parametra> = <vrednost>
+ [specifikacija tolerancija]]*)

36. .NODESET - predlog početnih uslova

.NODESET < V(<čvor> [, <čvor>]) = <vrednost> >*

37. .NOISE - analiza šuma

.NOISE V(<čvor> [, <čvor>]) <ime>
+ [<interval štampanja>]

38. .OP - štampanje izračunatih graničnih uslova

.OP

39. .OPTIONS - opcije simulacije

.OPTIONS [parametar]* [<parametar> = <vrednost>]*

40. .PARAM - definicija parametara

.PARAM < <ime> = <vrednost> >*
.PARAM < <ime> = { <izraz> } >*

41. .PLOT - "crtanje" u tekstualnoj izlaznoj datoteci

.PLOT [DC] [AC] [NOISE] [TRAN] [izlazna promenljiva]*
+ ([<vrednost donje granice> ,
+ <vrednost gornje granice>])*

42. .PRINT - štampanje u tekstualnu izlaznu datoteku

.PRINT[/DGTLC] [DC] [AC] [NOISE] [TRAN]
+ [izlazna promenljiva]*

43. .PROBE - priprema grafičke izlazne datoteke probe.dat

.PROBE[/CSDF] [izlazna promenljiva]*

44. .SAVEBIAS - zapisivanje graničnih uslova u datoteku

.SAVEBIAS <"ime datoteke"> <[OP] [TRAN] [DC]>
+ [NOSUBCKT]
+ [TIME=<vrednost> [REPEAT]] [TEMP=<vrednost>]
+ [STEP=<vrednost>] [MCRUN=<vrednost>]
+ [DC=<vrednost>]
+ [DC1=<vrednost>] [DC2=<vrednost>]

45. .SENS - analiza osetljivosti

.SENS <ime promenljive>*

46. .STEP - parametarska analiza

.STEP [LIN] <ime varirane promenljive>
 + <početna vrednost> <završna vrednost>
 + <vrednost koraka>
 .STEP [OCT] [DEC] <ime varirane promenljive>
 + <početna vrednost> <završna vrednost> <vrednost>
 .STEP <ime varirane promenljive> LIST <vrednost>*

47. .SUBCKT - definicija potkola

.SUBCKT <ime potkola> [čvor]*
 + [OPTIONAL: <<čvor>=<defoltna vrednost>>*]
 + [PARAMS: <<ime> = <vrednost> >*]
 + [TEXT: <<ime> = <tekstualna vrednost> >*]

48. .TEMP - zadavanje temperature

.TEMP <vrednost temperature>*

49. .TEXT - definicija tekstualnog parametra

.TEXT <<ime> = "<tekstualna vrednost>" >*
 .TEXT <<ime> = | <tekstualni izraz> | >*

50. .TF - transfer funkcije

.TF <ime izlazne promenljive>
 + <ime pobudnog generatora>

51. .TRAN - analiza u vremenskom domenu

.TRAN[/OP] <vrednost koraka štampanja>
 + <vreme završetka> [(vreme početka štampanja)
 + [(maksimalna vrednost koraka)] [UIC]

52. .WATCH - posmatranje rezultata analize u toku analize

.WATCH [DC] [AC] [TRAN] [<promenljiva>
+ [(<vrednost donje granice>,
+ <vrednost gornje granice>)]]*

53. .WCASE - analiza najgoreg slučaja

.WCASE <tip analize> <izlazna promenljiva> <funkcija>
+ [opcija]*

54. .WIDTH - dužina linije u tekstualnoj izlaznoj datoteci

.WIDTH OUT = <vrednost>

55. (tekst) - naslovna linija

(proizvoljni tekst)

56. * - komentar

* (tekst komentara)

57. ; - linijski komentar

(proizvoljna komanda ili opis elementa) ; (tekst komentara)

Prilog 3: PSpice - standardne funkcije

Funkcija	Matematički zapis	Komentar
ABS(x)	$ x $	
SQRT(x)	$x^{1/2}$	
EXP(x)	e^x	
LOG(x)	$\ln(x)$	prirodni logaritam
LOG10(x)	$\log_{10}(x)$	logaritam za osnovu 10
PWR(x, y)	$ x ^y$	
PWRS(x, y)	$\begin{matrix} + x ^y \text{ za } y>0, \\ - x ^y \text{ za } y<0 \end{matrix}$	
SIN(x)	$\sin(x)$	x u radijanima
COS(x)	$\cos(x)$	x u radijanima
TAN(x)	$\text{tg}(x)$	x u radijanima
ATAN(x)	$\text{arctg}(x)$	rezultat u radijanima
ARCTAN(x)	$\text{arctg}(x)$	rezultat u radijanima

Literatura

- [91] -, "PSpice Circuit Analysis", Version 4.05, *MicroSim Corporation*, January 1991.
- [98] -, "OrCAD PSpice – User's guide", <http://www.electronics-lab.com/downloads/schematic/013/tutorial/PSPICE.pdf>, 1998.
- [An88] Antognetti, P., Massobrio, G., editors, "Semiconductor Device Modeling with SPICE", *McGraw-Hill Book Company*, New York, 1988.
- [Lit00] Litovski V., "Projektovanje elektronskih kola", *Nova Jugoslavija*, Vranje, 2000, ISBN: 86-7369-015-3.
- [Lit06] Litovski V., "Osnovi elektronike, teorija, rešeni zadaci i ispitna pitanja", *Akadska misao*, Beograd, 2006, ISBN: 86-7466-227-7.
- [Lit09] Litovski V., "Modelovanje komponenta elektronskih kola i sistema", *Univerzitet u Nišu, Elektronski fakultet*, 2009, ISBN: 978-86-85195-71-6.
- [Mi88] Milovanović, G., "Numerička analiza", *Naučna knjiga*, Beograd, 1988.
- [Pet09] Petković P. M., "Projektovanje CMOS integrisanih kola sa mešovitim signalima", *Univerzitet u Nišu, Elektronski fakultet*, 2009, ISBN: 978-86-85195-86-0.