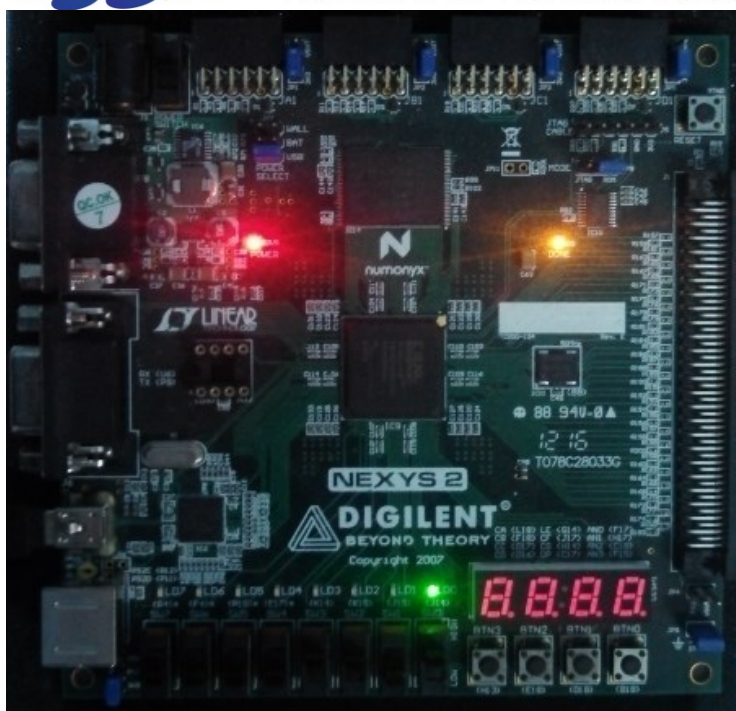

ПРИРУЧНИК ЗА ПРОЈЕКТОВАЊЕ ДИГИТАЛНИХ МРЕЖА КОРИШЋЕЊЕМ FPGA ИНТЕГРИСАНОГ КОЛА СА ПРИМЕРИМА

Вања Луковић
Александар Пеулић
Ђорђе Дамњановић
Радојка Крнета

Чачак, Март 2017





NeReLa
Network of Remote Labs

Пројектовање дигиталних мрежа коришћењем FPGA интегрисаних кола са примерима

This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Универзитет у Крагујевцу
Факултет техничких наука Чачку

ПРИРУЧНИК ЗА ПРОЈЕКТОВАЊЕ ДИГИТАЛНИХ МРЕЖА КОРИШЋЕЊЕМ FPGA ИНТЕГРИСАНОГ КОЛА СА ПРИМЕРИМА

Аутори:

Др Вања Луковић, доцент Факултета техничких наука у Чачку, Универзитет у Крагујевцу

Проф. Др Александар Пеулић, редовни професор Факултета инжењерских наука у Крагујевцу

Ђорђе Дамњановић, асистент Факултета техничких наука у Чачку, Универзитет у Крагујевцу

Проф. Др Радојка Крнета, редовни професор Факултета техничких наука у Чачку, Универзитет у Крагујевцу

Рецензенти

Проф. др Горан Ђорђевић, редовни професор Електронског факултета у Нишу
Др. Захарије Радивојевић, доцент Електротехничког факултета у Београду

Издавач

Факултет техничких наука у Чачку Универзитета у Крагујевцу, Светог Саве 65, 32000 Чачак

За издавача

Декан Факултета техничких наука: проф. др Небојша Митровић, дипл. инж. ел.

Тираж

100 примерака

Назив и седиште штампарije

Графичка радња штампарија Бајић, В. Игњатовића 12, Трбушани, Чачак

ISBN 978-86-7776-212-4

НАПОМЕНА: Забрањено је прештампавање и фотокопирање. Сва права задржава издавач и аутори.

Одлуком Наставно-научног већа Факултета техничких наука у Чачку број 28-606/11 од 3.4.2017. године одобрено је штампање књиге ПРИРУЧНИК ЗА ПРОЈЕКТОВАЊЕ ДИГИТАЛНИХ МРЕЖА КОРИШЋЕЊЕМ FPGA ИНТЕГРИСАНОГ КОЛА СА ПРИМЕРИМА.

Чачак, 2017.

САДРЖАЈ

САДРЖАЈ	ERROR! BOOKMARK NOT DEFINED.
УВОД.....	1
1. ТЕОРЕТСКЕ ОСНОВЕ	3
2. КРЕИРАЊЕ АПЛИКАЦИОНОГ ПРОЈЕКТА У <i>XILINX ISE DESIGN SUITE</i> СОФТВЕРСКОМ АЛАТУ	11
3. СИМУЛАЦИЈА РАДА ДИЗАЈНИРАНЕ ДИГИТАЛНЕ МРЕЖЕ	26
4. ИМПЛЕМЕНТАЦИЈА ДИЗАЈНИРАНЕ ДИГИТАЛНЕ МРЕЖЕ	36
5. ОПИС <i>LIREX</i> ЕКСПЕРИМЕНТА <i>NEXYS 2 FPGA PLATFORMA</i>	42
6. ПРИЛОГ – РАЗНИ ПРИМЕРИ ДИЗАЈНИРАЊА ДИГИТАЛНИХ МРЕЖА КОРИШЋЕЊЕМ <i>FPGA</i> ИНТЕГРИСАНОГ КОЛА.....	44
ПРИМЕР 1.	44
ПРИМЕР 2.	45
ПРИМЕР 3.	48
ПРИМЕР 4.	53
ПРИМЕР 5.	57
ПРИМЕР 6.	60
ПРИМЕР 7.	66
ЛИТЕРАТУРА	75
ЛИСТА СЛИКА.....	76

УВОД

Приручник за пројектовање дигиталних мрежа коришћењем *FPGA* интегрисаног кола са примерима је настао као резултат активности на Темпус пројекту *NeReLa* (2013-2016), чији је пуни назив „*Building Network of Remote Labs for strengthening university-secondary vocational schools collaboration*“ („Изградња мреже удаљених лабораторија за јачање сарадње универзитета и средњих стручних школа“) [1] и који има за циљ јачање сарадње техничких факултета и средњих стручних школа кроз увођење иновација у настави стручних предмета на техничким факултетима и средњим стручним школама, применом удаљених експеримената.

Овај Приручник је намењен студентима Електротехничког и рачунарског инжењерства који изучавају логичко пројектовање дигиталних мрежа из предмета као што су Основи рачунарске технике 1 и 2 [2], [3], Логичко пројектовање [4], Дигитална електроника [5] и њима слични. Иновативни наставни метод који се примењује у настави ових предмета од школске 2015/16 године на Факултету техничких наука у Чачку и Факултету инжењерских наука у Крагујевцу подразумева употребу удаљеног експеримента базираног на програмирању *FPGA* интегрисаног кола коришћењем *HDL* језика (*VHDL* или *Verilog* језика) или шематских дијаграма. Применом овог иновативног наставног метода је омогућено да велики број студената електротехничког и рачунарског инжењерства на раним годинама студија стекне практична знања и вештине за рад са *FPGA* и тиме поткрепе стечена теоријска знања из логичког пројектовања дигиталних мрежа, експериментишући при том на једној истој поставци експеримента у унапред резервисаним временским терминима.

Према детаљном упутству из Приручника потребно је да студент на свом рачунару претходно инсталира следеће софтверске алате: *Xilinx ISE Design Suite* [6], [7], *Digilent Adept2* [8], [9], *.Net Framework 4.5* [10] и *CEyeClon Viewer* [11], који ће му омогућити програмирање *FPGA* интегрисаног кола *Spartan3E-XC3S1200E-FG320* смештеног на *Digilent Nexys 2* платформи која се налази у удаљеној лабораторији Факултета инжењерских наука у Крагујевцу. Програмирање *FPGA* интегрисаног кола детаљно је објашњено на примеру пројектовања једне дигиталне мреже. Најпре је објашњено како се креира нови пројекат у *Xilinx ISE Design Suite* софтверском алату, да би потом била објашњена сва три начина дизајнирања дигиталне мреже коришћењем:

- шематског приступа;
- програмирања у *VHDL* језику;
- програмирања у *Verilog* језику.

Симулација рада дизајниране дигиталне мреже подразумева промену вредности излазних сигнала у зависности од унапред задате промене улазних сигнала, током одређеног временског периода. У приручнику је објашњено како се може вршити симулација у *ISim* симулатору [12] на следећа три начина:

- Коришћењем *VHDL* фајла;
- Коришћењем *Verilog* фајла;
- Директно у *ISim* симулатору;

Одабир улазно/излазних сигнала и уређаја врши се у оквиру имплементације реализованог шематског, *VHDL* или *Verilog* пројекта на *FPGA* интегрисано коло.

Удаљени експеримент „[Nexys 2 FPGA platforma](#)“ представља један од удаљених експеримената/ресурса Библиотеке удаљених експеримената *LiReX* (*Library of Remote Experiment*) [13], којима се може приступити преко интернета. Овај експеримент омогућава програмирање *Spartan3E-XC3S1200E-FG320 FPGA* интегрисаног кола на *Digilent Nexys 2* платформи, која се налази у удаљеној лабораторији. Уносом приступног кључа на *LiReX* платформи, преко *CEyeClon Viewer*-а врши се логовање/повезивање на рачунар у удаљеној лабораторији и програмирање *FPGA* интегрисаног кола. Експеримент је постављен тако да су четири прекидача (*SW0:3*) постављени у положај укључено, а преостала четири (*SW4:7*) у положају искључено. Резултат програмирања студент може видети у реалном времену преко веб камере која је постављена изнад *Nexys 2* плоче у лабораторији.

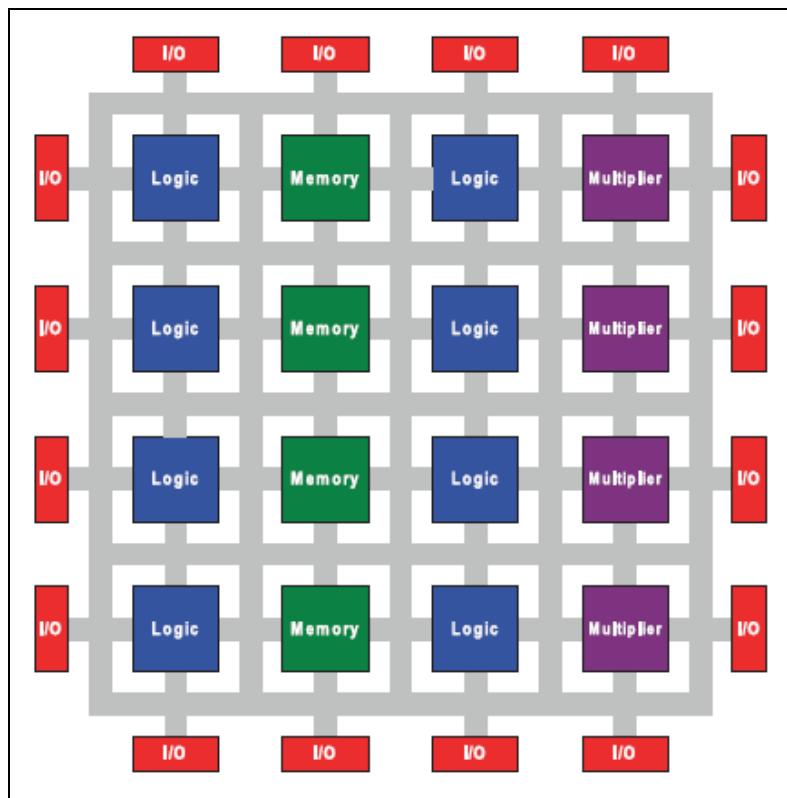
На крају приручника дати су разни примери пројектовања дигиталних мрежа програмирањем *FPGA* интегрисаног кола са упутством за решавање и решењима.

Приручник који је расположив и у електронском облику у оквиру *LiReX* библиотеке, могу користити и наставници и студенти на другим техничким факултетима који су заинтересовани да уз теоретску наставу/учење логичког пројектовања дигиталних мрежа уведу и коришћење удаљених експеримената из *LiReX* библиотеке.

1. ТЕОРЕТСКЕ ОСНОВЕ

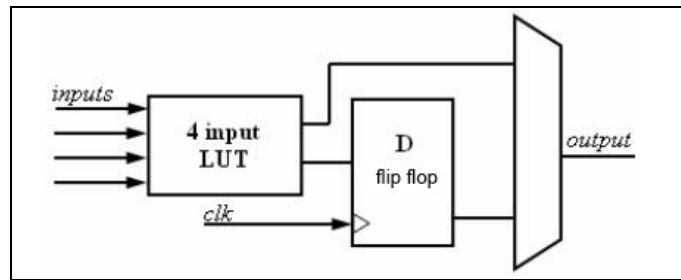
1.1. Основне компоненте *FPGA* интегрисаног кола

FPGA (*Field-Programmable Gate Array*) представља интегрисано коло чија се унутрашња структура може конфигурисати од стране крајњег корисника. Дефинисање унутрашње структуре *FPGA* интегрисаног кола се врши коришћењем *HDL* језика или шематских дијаграма. У зависности од начина на који су имплементирани, нека *FPGA* интегрисана кола могу да се програмирају само једанпут, а друга могу бити репрограмирана изнова и изнова.



Слика 1. Основна структура *FPGA* интегрисаної кола

Основна структура *FPGA* интегрисаног кола је приказана на Сл. 1, а састоји се од: *RAM* и флеш *EPROM* програмске меморије, матрице програмабилних логичких блокова, потенцијално различитог типа, укључујући основне логичке и меморијске елементе и мултипликаторе блокова, окружених програмабилном *сјрежном мрежом* (програмабилном *архијектуром за рутирање*) која омогућава програмирање међусобног повезивања блокова. Матрица је окружена програмабилним улазно/излазним блоковима, који су означени са *I/O* на Сл. 1, а који служе за повезивање *FPGA* интегрисаног кола са спољашњим окружењем. *FPGA* се карактеришу ситно-зрнастом (*finegrain*) архитектуром. То значи да се логичке операције обављају на нивоу бита или на нивоу речи малог капацитета (< 4 бита), што омогућава креирање веома флексибилних архитектура прилагођених специфичним захтевима апликације.

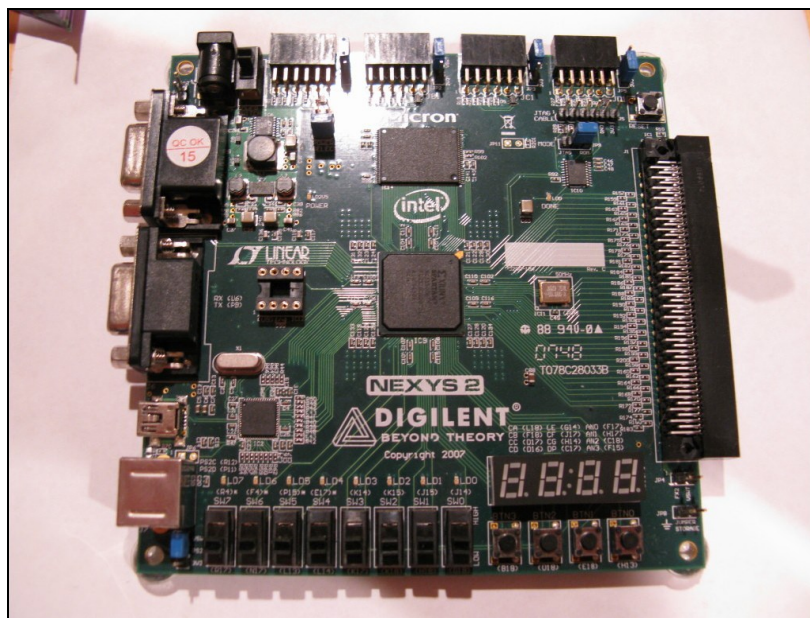


Слика 2.. Основна структура *FPGA* логичкој блока

Класичан *FPGA* логички блок има 4 улаза у логички део, по један *D* флип-флоп и мултиплексер (Сл. 2). Постоји само један излаз, који може бити синхронизован са тактом или се може директно водити са излаза из логичког блока. Пошто је сигнал такта (*CLK - clock*) од посебне важности и како је његова улога углавном стандардна за сваку мрежу, он се блоку придружује као посебан сигнал (одвојен од осталих улаза).

Главне произвођаче *FPGA* компоненти на тржишту чине компаније *Xilinx* и *Altera*. Заједно контролишу преко 80% тржишта, а сам *Xilinx* преко 50% читавог тржишта. Компаније које такође производе *FPGA* компоненте су: *Lattice Semiconductor*, *Actel*, *Atmel*, *SiliconBlue Technologies*, *QuickLogic* и *Achronix*.

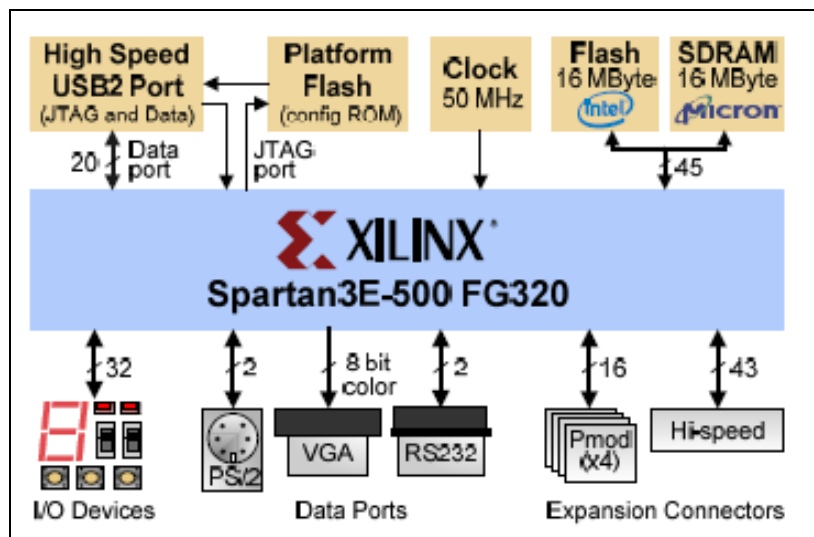
1.2. Digilent Nexys 2 платформа



Слика 3. *Digilent Nexys 2* платформа

Digilent Nexys 2 платформа је платформа за креирање дигиталних мрежа која се базира на *Xilinx Spartan 3E FPGA* интегрисаном колу (Сл. 3). *Digilent Nexys 2* платформа поседује следеће компоненте приказане на Сл. 4:

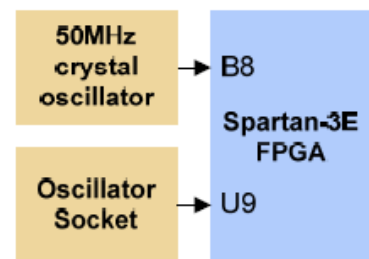
- *Xilinx Spartan 3E FPGA* интегрисано коло;
- *USB2* (high speed) портове за повезивање са рачунаром у циљу конфигурисање *FPGA* и преноса података;
- *USB* порт за екстерно напајање;
- 16 MB *SDRAM* и 16 MB *Flash ROM* меморије;
- *Platform Flash* за не-променљиво конфигурисање *FPGA* интегрисаног кола;
- Осцилатор (Clock) од 50MHz и подножје за секундарни осцилатор (Сл. 6);
- Низ улазно/излазних (I/O) уређаја и портова: 8 *LED* диода, 4 тастера, 8 прекидача, четвороцифарни седмо-сегментни дисплеј (Сл. 5a));
- *PS/2* порт за прикључивање миша или тастатуре, *VGA* порт, серијски порт *RS232*, један *Hi-speed Hirose FX2* конектор;
- 4x12-pin *Peripheral Module (Pmod)* конектор за управљање моторима, *A/D* и *D/A* конверзију, прикључивање аудио уређаја, сензора и актуатора (<http://store.digilentinc.com/>).



Слика 4. Блок дијаграм Digilent Nexys 2 платформе



a)



b)

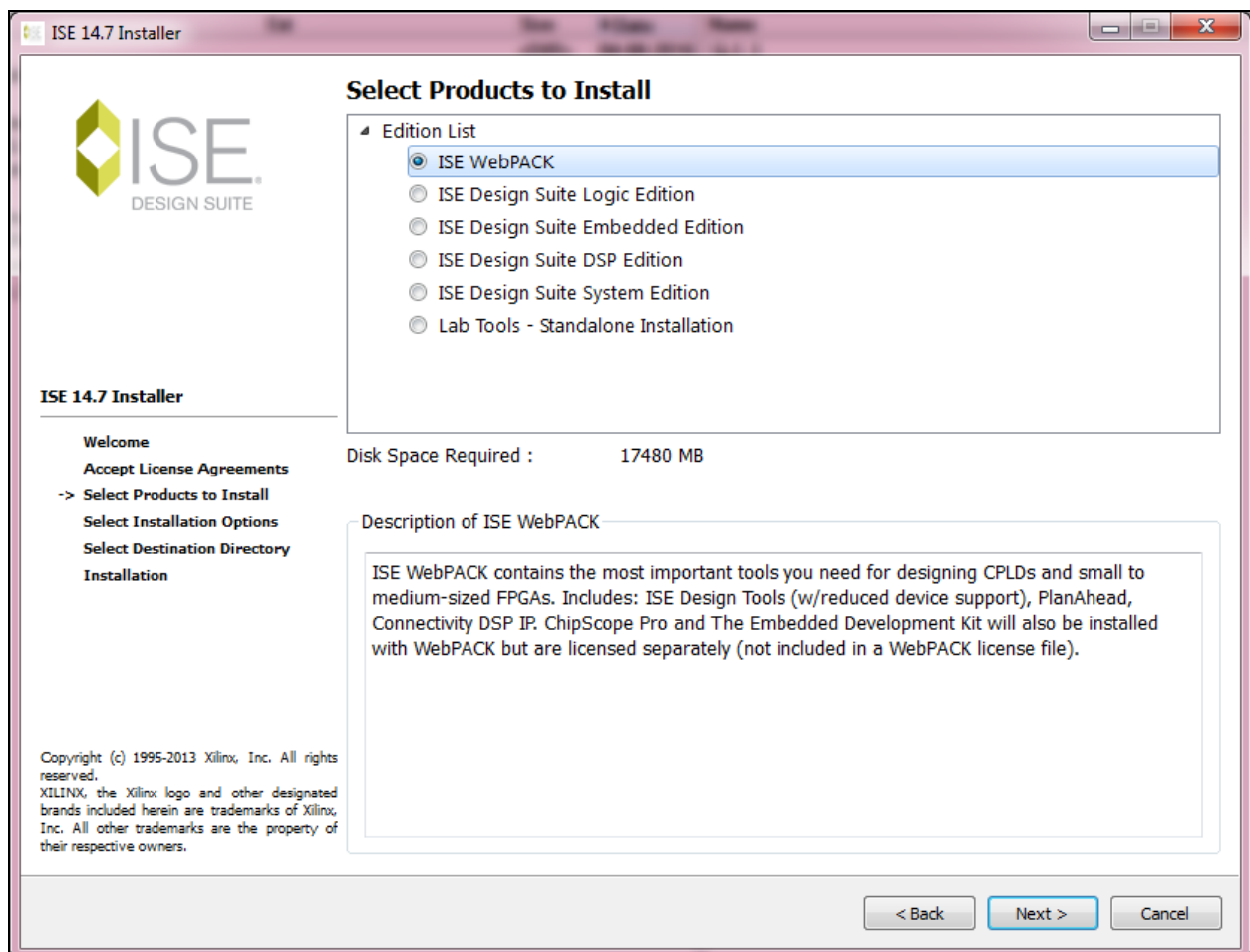
Слика 5. а) Улазно/излазни (I/O) уређаји Digilent Nexys 2 платформе б) Осцилатор (Clock) од 50MHz и подножје за секундарни осцилатор

1.3. Инсталирање Xilinx ISE Design Suite софтверког алата

Xilinx ISE Design Suite софтверски алат подржава програмирање следећих врста Xilinx FPGA интегрисаних кола: *Spartan-6*, *Virtex-6* и *CoolRunner*, као и свих њихових претходних фамилија. Са друге стране, Xilinx Vivado Design Suite софтверски алат подржава програмирање: *Virtex-7*, *Kintex-7*, *Artix-7* и *Zynq-7000* врста Xilinx FPGA интегрисаних кола и њихових претходних верзија. С обзором да је тип FPGA који нам је на располагању *Spartan3E-1200E FG320*, то значи да је за програмирање потребно користити *Xilinx ISE Design Suite* софтверски алат.

Постоје три основне варијанте Xilinx ISE Design Suite софтверског алата:

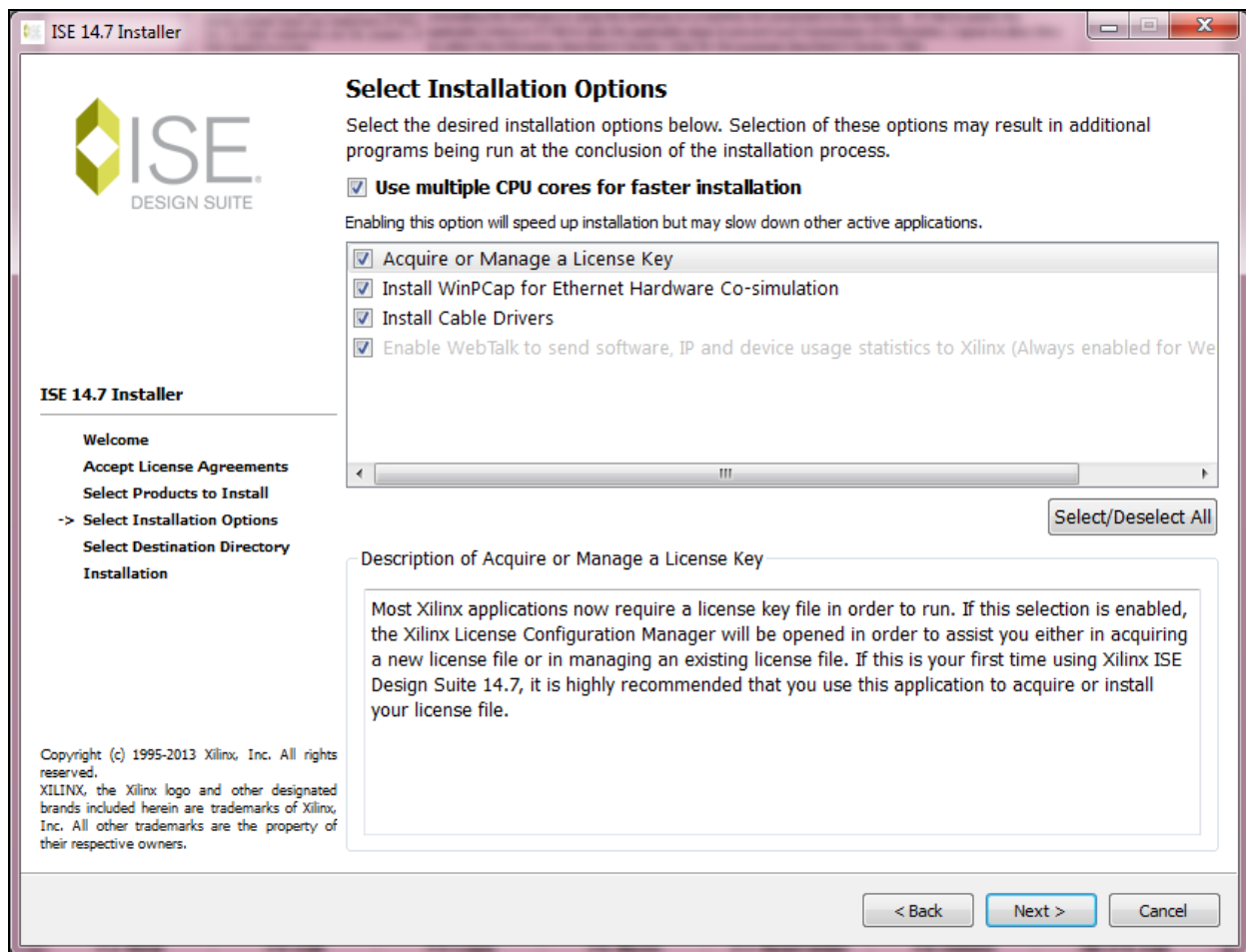
- *Xilinx ISE Design Suite - Embedded edition*
- *Xilinx ISE Design Suite - System edition*
- *Xilinx ISE Design Suite - Web pack edition*



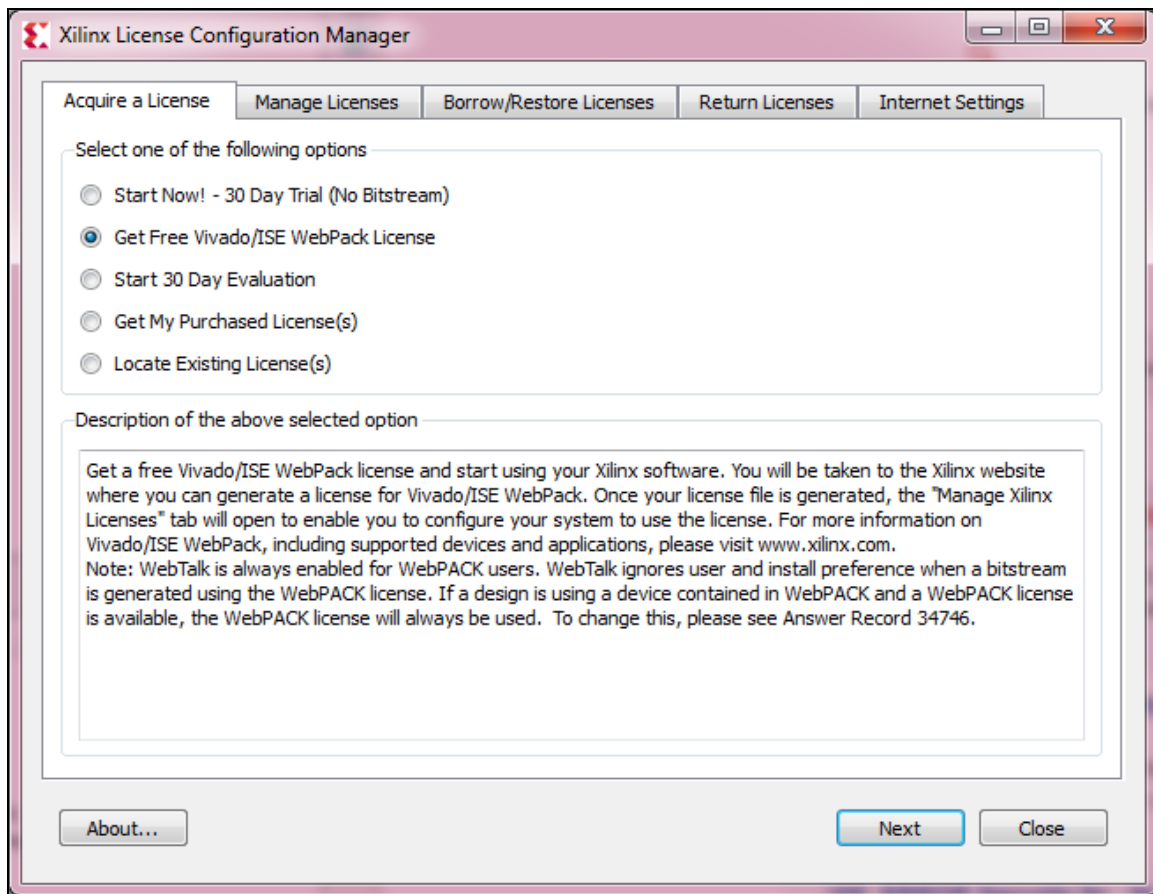
Слика 6. Прозор за селекцију варијанте Xilinx ISE Design софтвера

Детаљне карактеристике свих наведених варијанти овог софтверског алата се могу пронаћи на <http://www.xilinx.com/products/design-tools/ise-design-suite.html>. Међутим, бесплатно је доступна једино *Xilinx ISE Design Suite: Web pack edition* варијанта овог софтверског алата, а компатибилна је са следећим оперативним системима: *Linux*, *Windows XP* и *Windows 7*. Њена инсталација се може преузети са следеће веб адресе: <http://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/design-tools.html>, при чему величина инсталације варира у опсегу од 6 GB до 7GB, у зависности од врсте оперативног система.

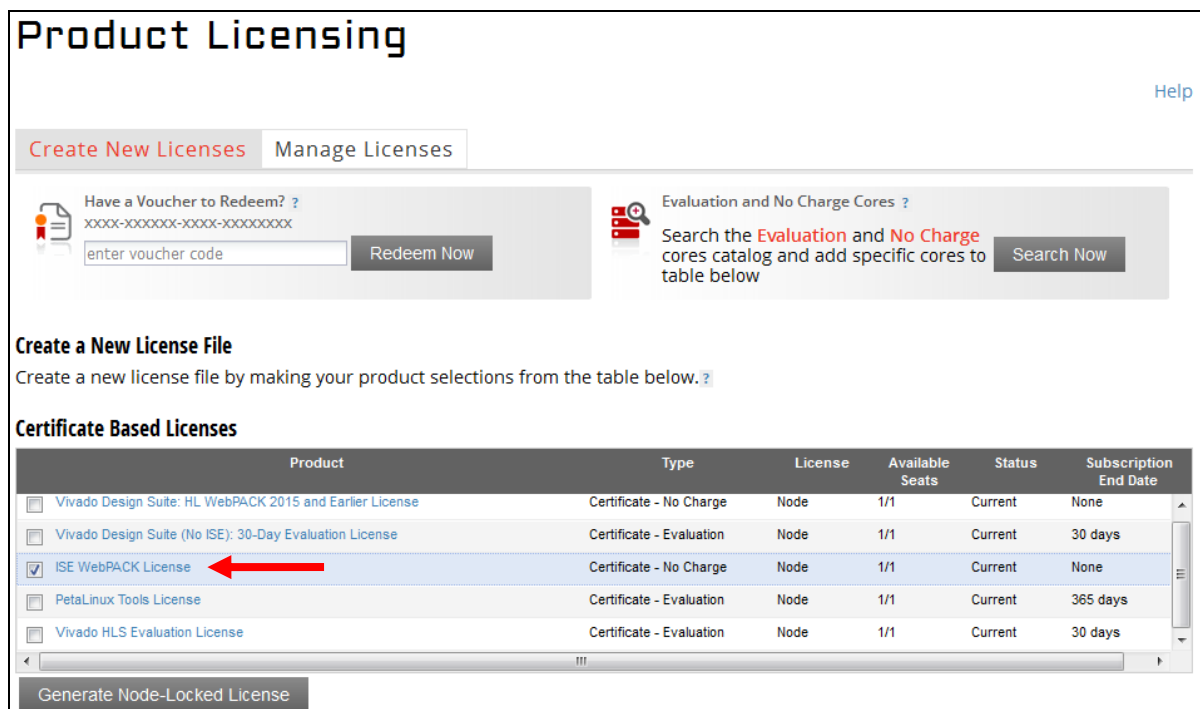
Након преузимања и стартовања процеса инсталације, најпре се појављује прозор добродошлице у коме се појављује упозорење да је за скраћивање времена инсталирања препоручљиво привремено искључити антивирусни софтвер на рачунару. Након тога појављују се прозори у којима је потребно прихватити услове лиценцирања, а потом прозор у коме је потребно извршити селекцију *ISE Web pack* варијанте *Xilinx ISE Design Suite* софтверског алата (Сл. 6). Надаље се врши избор програмских компоненти, које ће бити инсталиране, при чему се у овом кораку могу одабрати све понуђене компоненте (Сл. 7). У претпоследњем кораку се бира локација инсталационе фасцикле на диску, да би у последњем кораку започео процес инсталирања *Xilinx ISE Design Suite* софтвера.



Слика 7. Избор опционалних компоненти Xilinx ISE Design софтвера за инсталацију

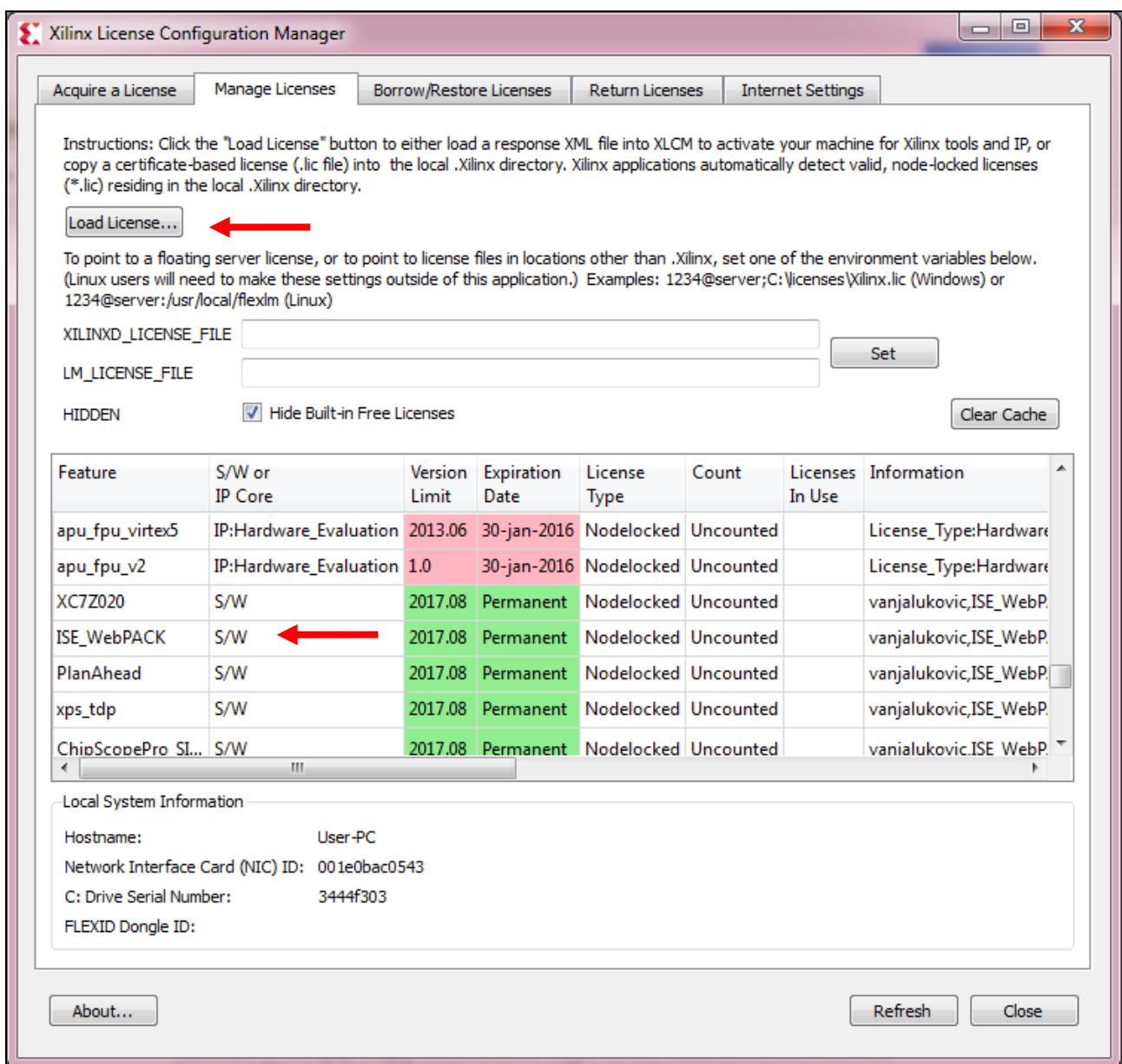


Слика 8. Селектовање опције преузимања бесплајне лиценце Xilinx ISE Design софтвера



Слика 9. Избор ISE WebPACK лиценце

Након завршеног процеса инсталирања, потребно је извршити лиценцирање преузетог софтвера. Након првог покретања инсталираног *Xilinx ISE Design Suite* софтверског алата, аутоматски се појављује *Xilinx License Configuration Manager* за преузимање лиценце у коме је потребно селектовати *Get Free Vivaldo/ISE Web Pack License* опцију преузимања бесплатне лиценце (Сл. 8). *Xilinx License Configuration Manager* могуће је и директно отворити коришћењем *Help/Manage License* опције главног менија *Xilinx ISE Design Suite* софтверског алата. Након одабирања врсте лиценце, успоставља се конекција са *Xilinx Licensing Site* сајтом где је најпре потребно регистровати се. Регистрација подразумева дефинисање корисничког имена и шифре и уноса основних података о кориснику и намени инсталираног софтвера.



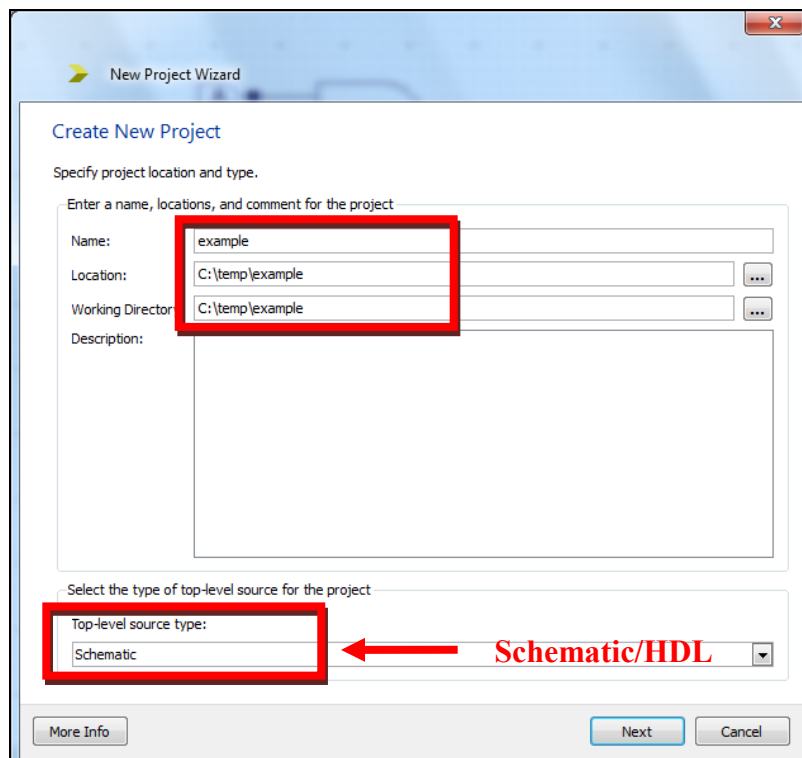
Слика 10. Приказ учинице ISE WebPACK лиценце Xilinx License Configuration Manager-a

Након завршеног процеса регистрације појављује се веб страница у којој је потребно одабрати врсту лиценце *ISE WebPACK*, као што је приказано на Сл. 9. Селектована лиценца се кликом на дугме *Generate Node Locked licence* на Сл. 9 прослеђује на *e-mail* адресу, која је наведена приликом регистрације. Преузету лиценцу потребно је учитати преко *Manage Licenses* прозора *Xilinx License Configuration Manager-a* кликом на дугме *Load License* (Сл. 10). На Сл. 10 приказан је изглед *Manage Licenses* прозора *Xilinx License Configuration Manager-a* након учитавања *ISE WebPACK* лиценце. Након завршеног процеса лиценцирања *Xilinx ISE Design Suite* софтверски алат је коначно спреман за коришћење.

2. КРЕИРАЊЕ АПЛИКАЦИОНОГ ПРОЈЕКТА У *XILINX ISE DESIGN SUITE* СОФТВЕРСКОМ АЛАТУ

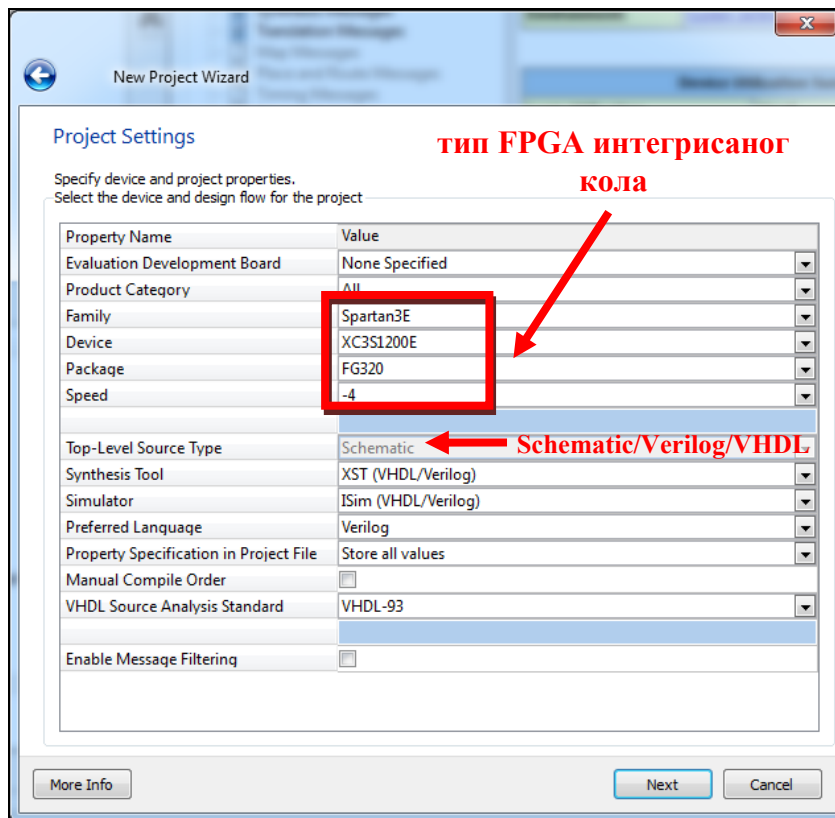
Digilent Nexys 2 платформа је потпуно компатибилна са свим верзијама *Xilinx ISE Design Suite* софтверског алата за програмирање *Xilinx Spartan 3E FPGA* интегрисаног кола. Нови пројекат у *Xilinx ISE Design Suite* софтверском алату се креира избором команде *New Project* из каскадног менија *File*. Избором ове команде стартује се чаробњак, при чему се у првом кораку специфицира назив и локација пројекта и начин програмирања *FPGA* интегрисаног кола (Сл. 11).

На Сл. 11 одабран је шематски тип (*Schematic*) програмирања *FPGA* интегрисаног кола. Осим шематске на располагању је и *HDL* (*Hardware Description Language*) опција програмирања, коришћењем *VHDL* или *Verilog* језика. Као резултат претходног корака, врши се креирање подфасцикле назива *example* у којој се ће се налазити сви пројектни фајлови, при чему је главни пројектни фајл *example.xise*.

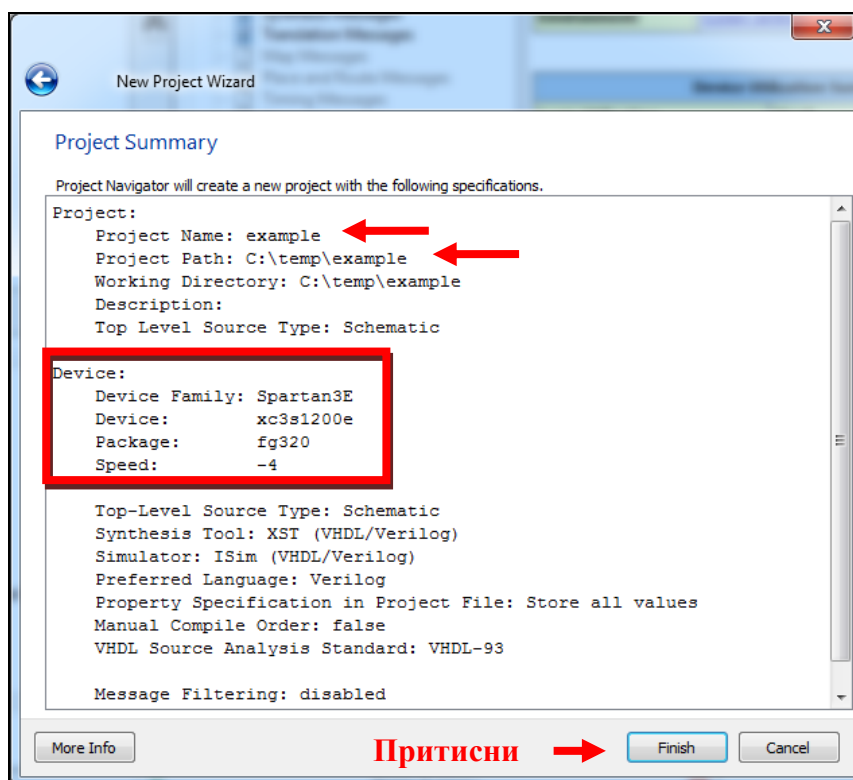


Слика 11. Креирање новој шематској пројекта у софтверском алату *Xilinx ISE Design*

У следећем и последњем кораку врши се спецификација типа *FPGA* интегрисаног кола, као и језик програмирања (Сл. 12). У нашем случају одабран је следећи тип *FPGA* интегрисаног кола: *Family-Spartan3E*, *Device-XC3S1200E* и *Package-FG320*. Као језик програмирања у случају шематског пројекта, као и пројекта у коме се *FPGA* интегрисано коло програмира у *VHDL* језику, може се одабрати *VHDL* језик програмирања, док се у случају пројекта у коме за *FPGA* интегрисано коло програмира у *Verilog* језику, потребно одабрати *Verilog* језик.

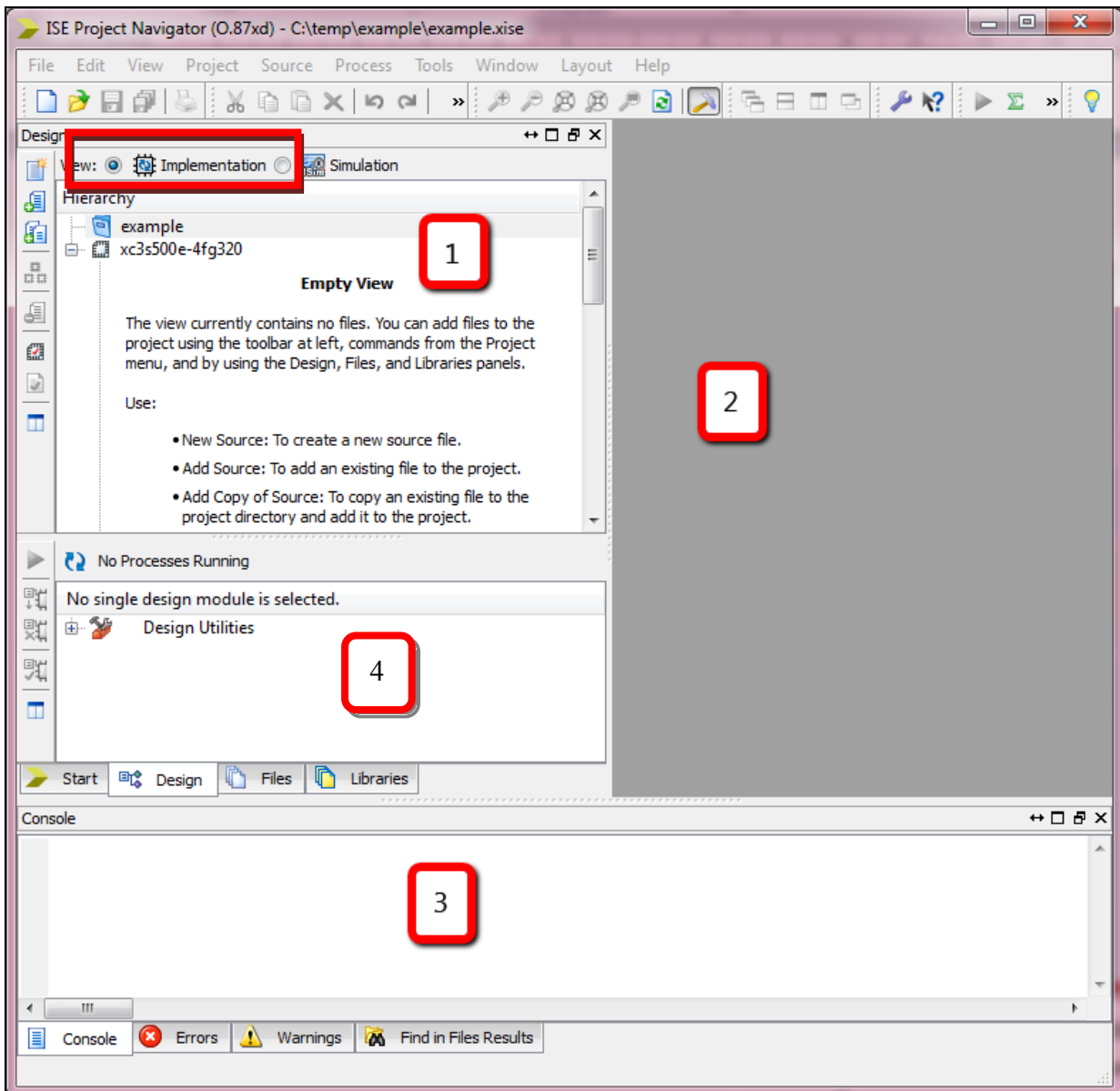


Слика 12. Дефинисање типича FPGA интегрисаної кола и језика програмирања



Слика 13. Дефинисање спецификације шемајскої пројекта креираної у Xilinx ISE Design Suite софтверском алату

Након овог корака креира се празан пројекат са дефинисаним спецификацијама, које су приказане на Сл. 13. На Сл. 14 приказан је изглед радног окружења након креирања пројекта у *Xilinx ISE Design Suite* софтверском алату.



Слика 14. *Xilinx ISE Design Suite* развојно окружење: 1- преглед садржаја пројекта; 2- радно окружење; 3-поруке о грешкама, ујозорења, конзола; 4 - алаји за покрећање процеса дизајнирања *FPGA* интегрисаної кола

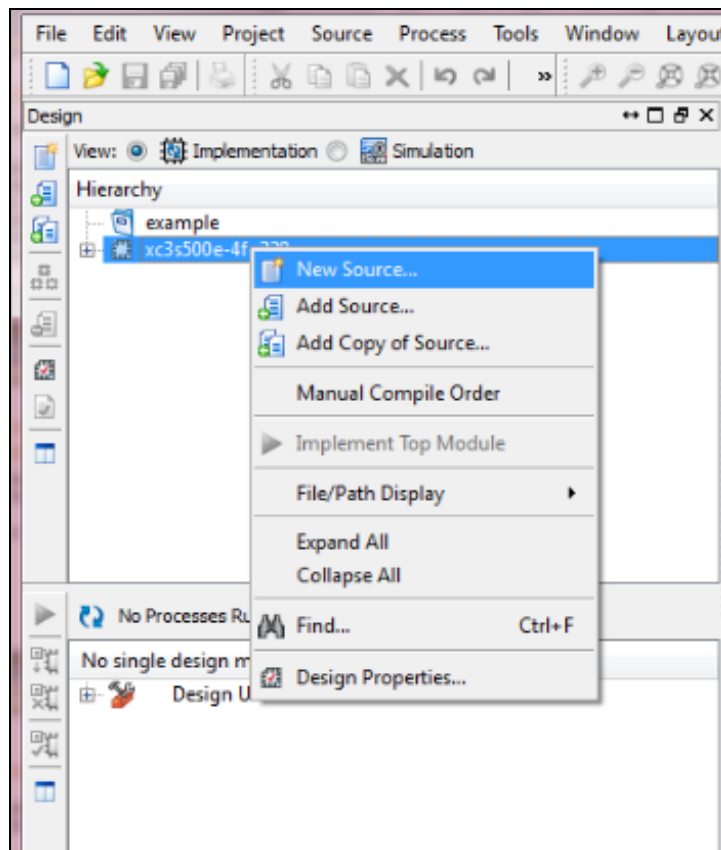
Након креирања пројекта може се приступити дизајнирању дигиталне мреже. То се може учинити на више начина:

- коришћењем шематског приступа (видети поглавље 2.1);
- програмирањем у *VHDL* језику (видети поглавље 2.2) или

- програмирањем у *Verilog* језику (видети поглавље 2.3).

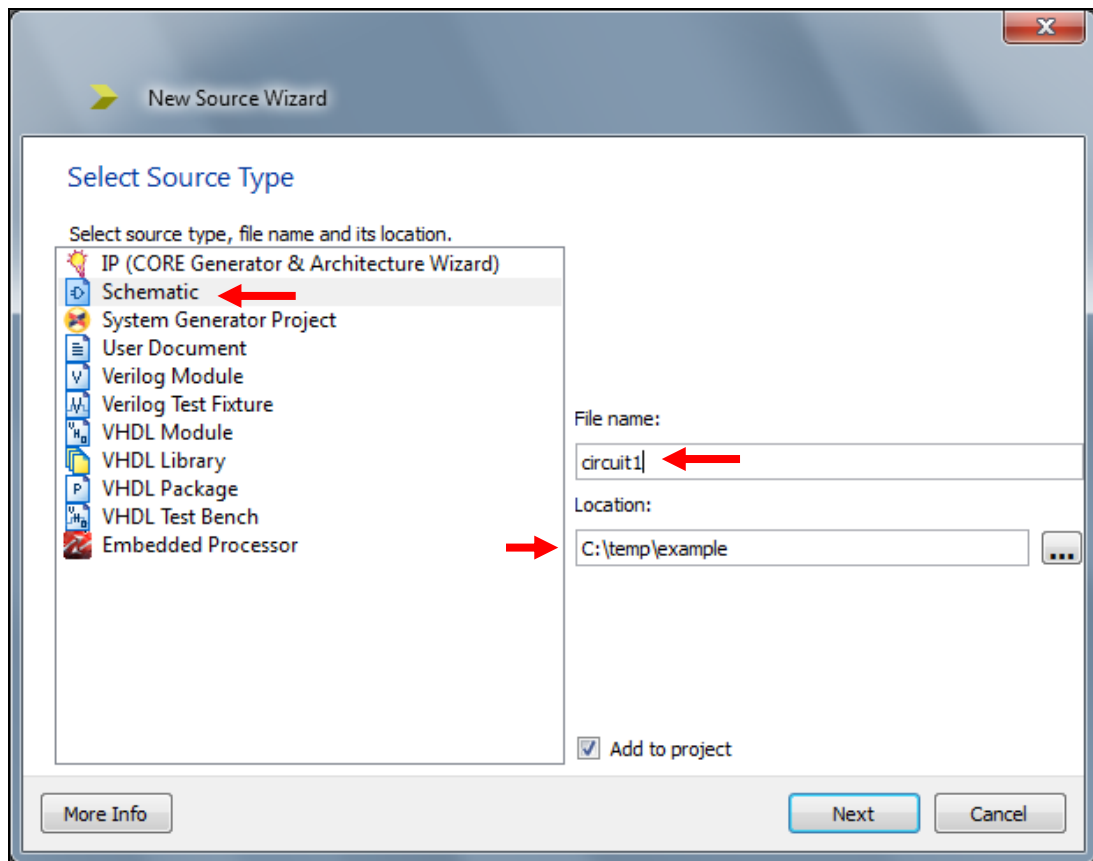
2.1. Креирање шематског фајла и додавање пројекту

У овом поглављу биће описан поступак креирања шематског типа фајла назива *circuit1* и његово додавање креираном пројекту у *Xilinx ISE Design Suite* софтверском алату. Након креирања пројекта према поступку описаном у поглављу 1, потребно му је додати фајл који ће да садржи изворни програмски код (*source*), десним кликом на пројекат у прозору 1 на Сл. 14 и избором опције *New Source* (Сл. 15). Овом приликом се стартује чаробњак у коме је потребно дефинисати тип и назив фајла који се креира, при чему је у овом случају потребно одабрати шематски тип фајла (Сл. 16).

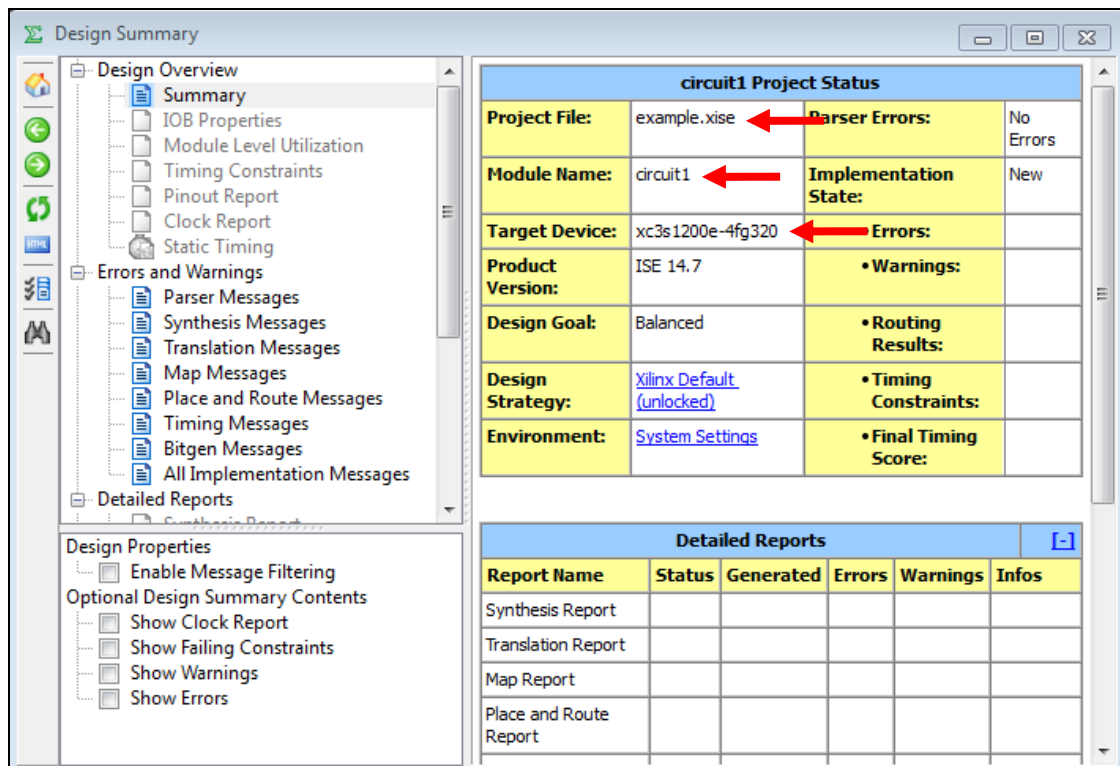


Слика 15. Додавање новој изворној фајла пројекту

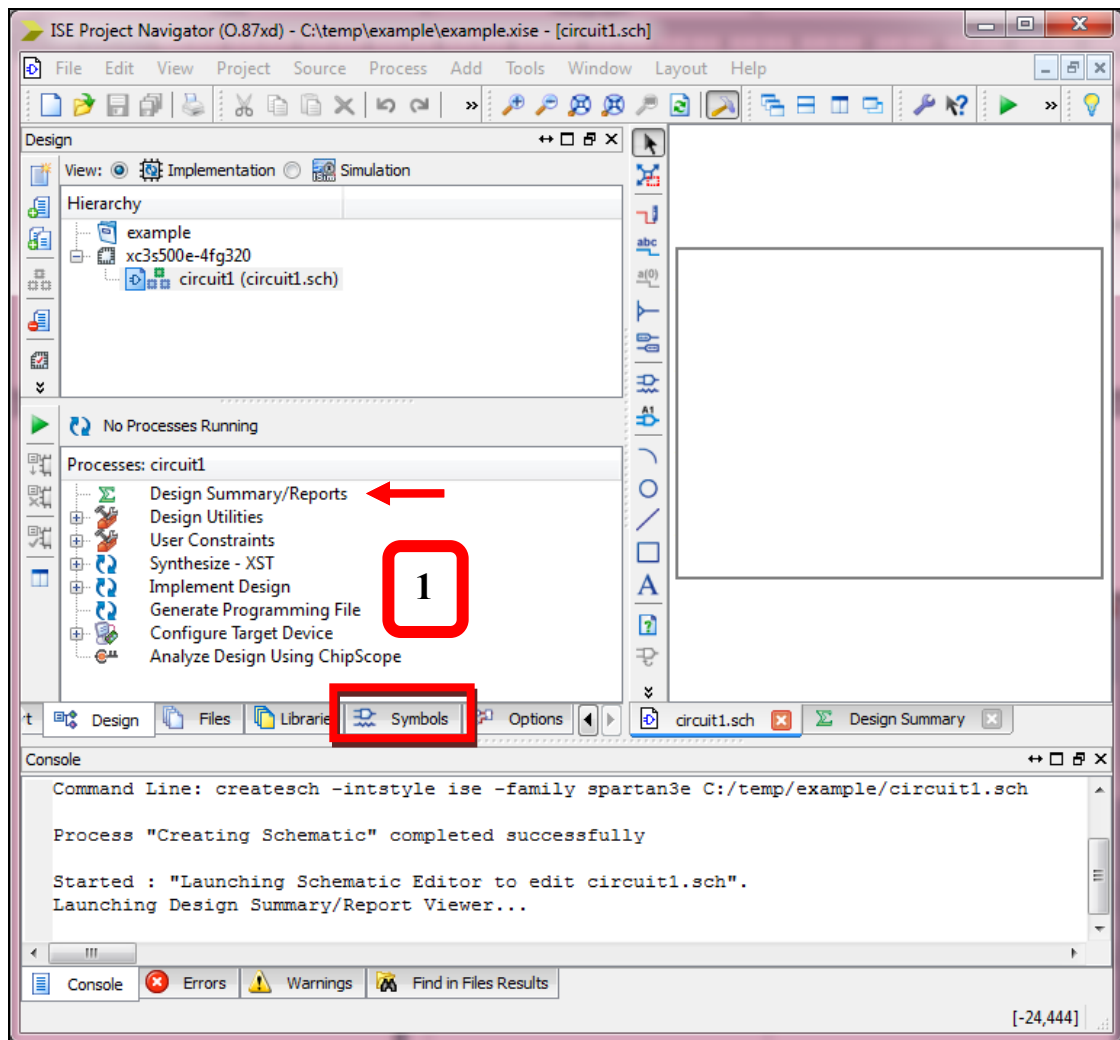
На Сл. 17 приказан је сумарни преглед карактеристика креираног пројекта, након додавања шематског фајла, коришћењем *Design Summary* прозора, који се може отворити у прозору за покретање процеса 1 на Сл. 18. На Сл. 18 приказан је изглед радног окружења након креирања шематског фајла, док је на Сл. 19 приказана палета алата шематског типа пројекта.



Слика 16. Дефінісање назива и шемајскої шийа фајла који се додаје креираном пројекту



Слика 17. Сумарни преглед карактеристика креираног пројекта

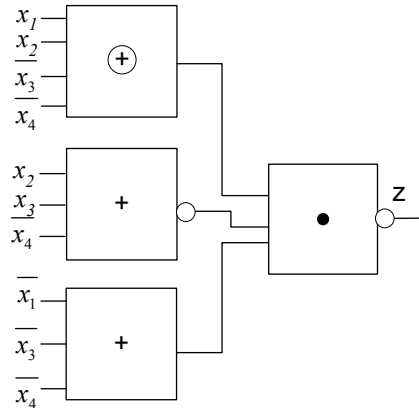


Слика 18. Изглед радної окружења након додавања изворної фајла; 1 – ґрозор за сґартување ґроцєса

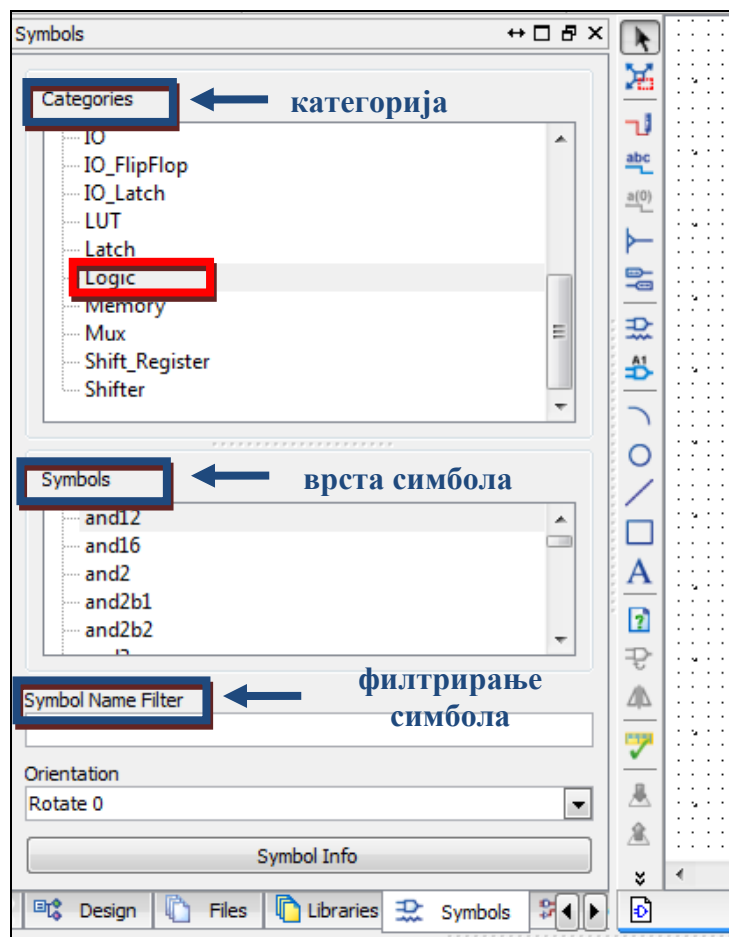
	Select tool (default)
	Add wire
	Add Net Name
	Rename Selected Bus
	Add bus tap
	Add Input/Output marker
	Add Symbol
	Add Instance Name
	Check Schematic 
	Add drawing & text 
	Push into a symbol
	Pop to the next higher level
	Zoom controls
	Zoom back and forth

Слика 19. Палєта алати шємайскої тииа ґројєкѝа

У нашем примеру потребно је реализовати дигиталну мрежу приказану на Сл. 20. У том циљу најпре је потребно одабрати *Symbols* опцију менија процесног прозора 1 на Сл. 18.



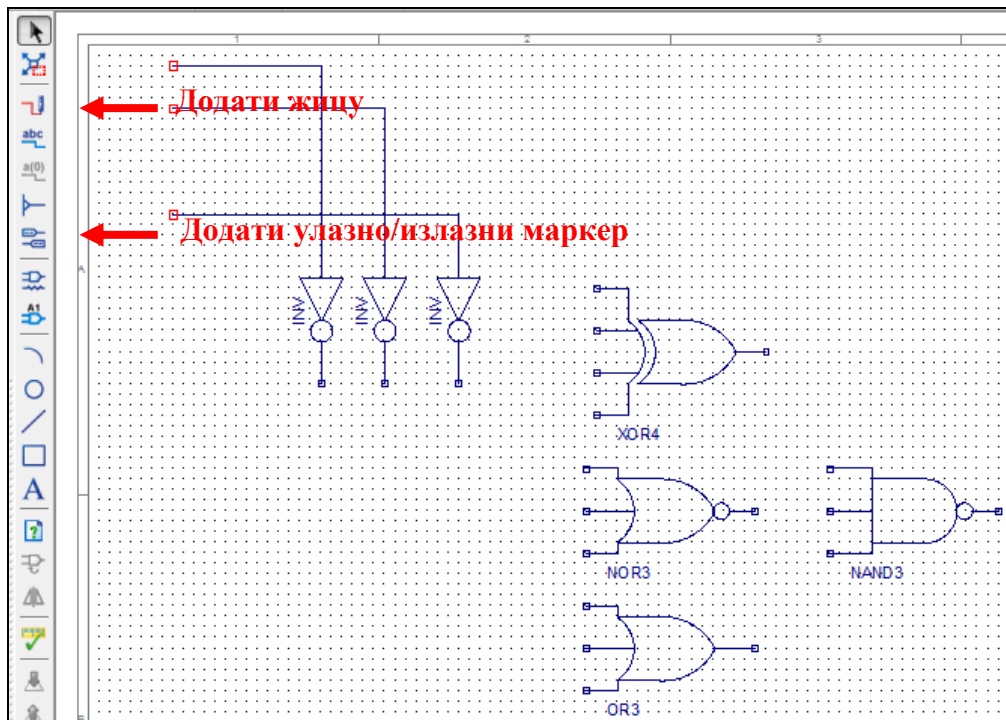
Слика 20. Дигитална мрежа за реализацију



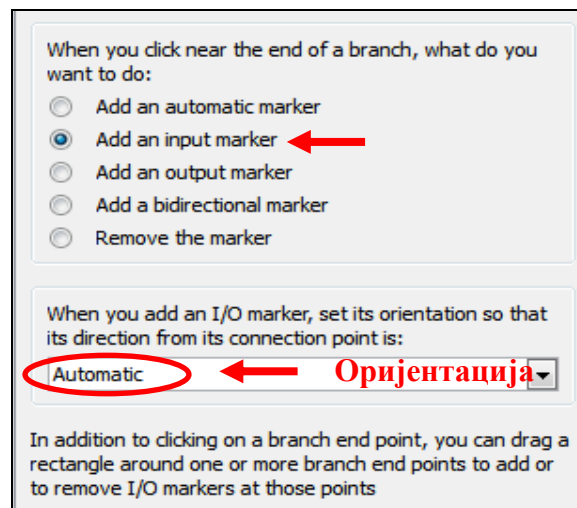
Слика 21. *Symbols* мени за дефинисање категорије и врсте симбола дигиталної мреже

Том приликом се појављује прозор на Сл. 21 у коме је потребно одабрати категорију симбола *Logic*. Након тога се врши додавање потребних логичких елемената на радну површину:

XOR4 (EXILI елемент са 4 улаза), *NOR3* (NILI елемент са 3 улаза), *OR3* (ILI елемент са 3 улаза) и *NAND3* (NI елемент са 3 улаза), као што је приказано на Сл. 22. Након тога додају се *INV* (*NE*) елементи за реализацију негација појединих улазних сигнала.



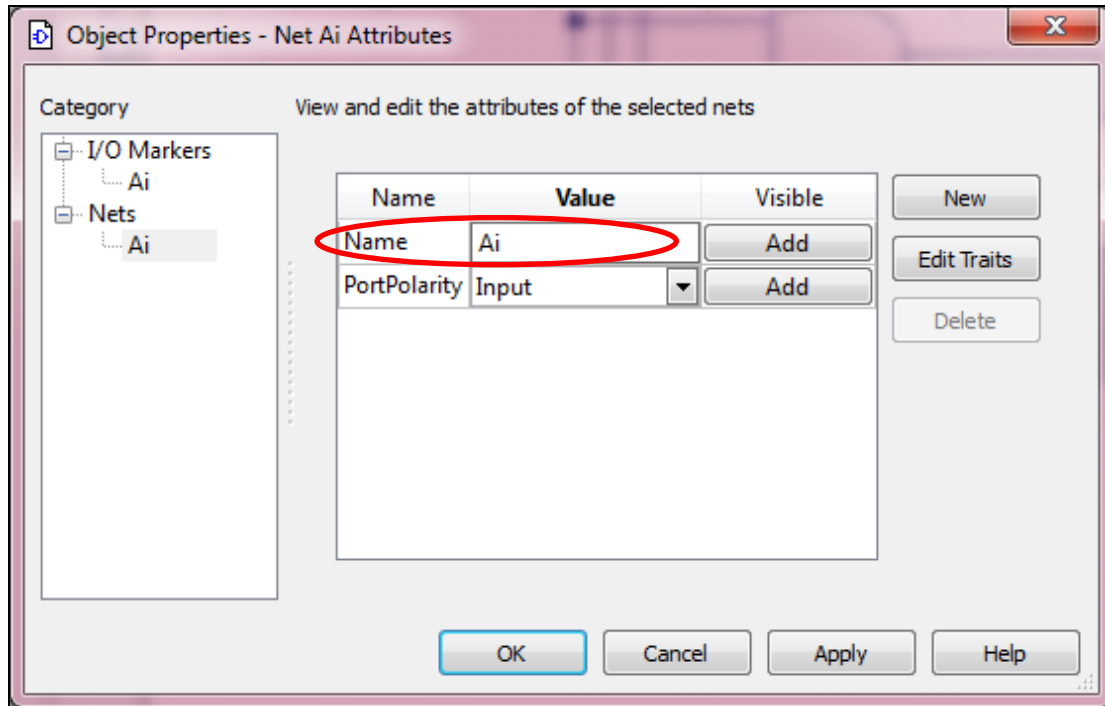
Слика 22. Дигитално коло – прва фаза додавања логичких елемената



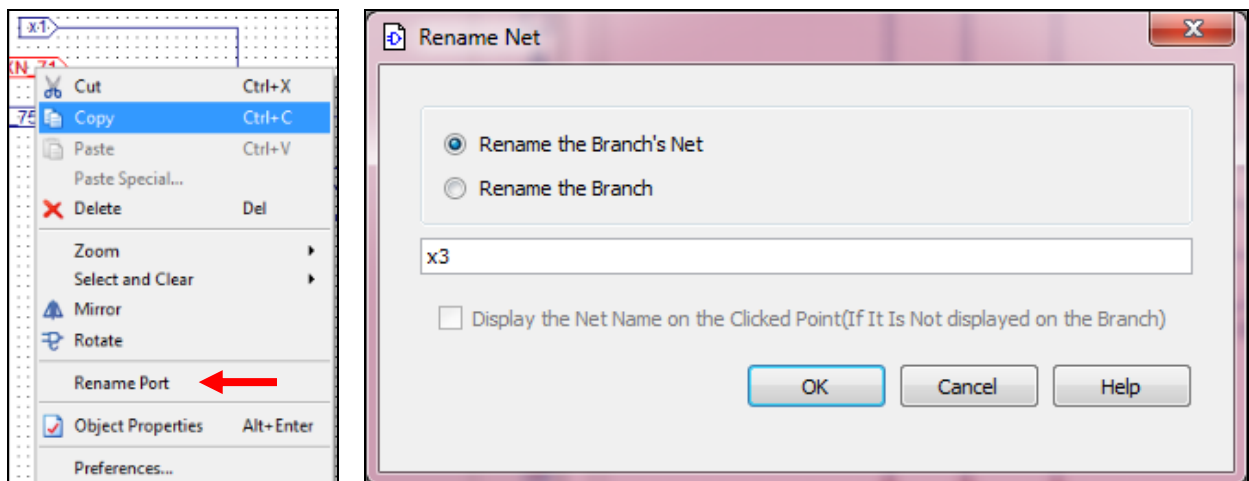
Слика 23. Спецификација типа и оријентације улазної/излазної сигнала

Након додавања логичких елемената потребно је извршити њихово међусобно повезивање коришћењем *Add wire* алата, као и додавање улазно/излазних сигнала коришћењем *Add Input/Output marker* опције на Сл. 22. Том приликом могуће је извршити спецификацију типа сигнала, као и дефинисати њихову оријентацију (Сл. 23). Двоструким кликом на додати

улазно/излазни сигнал врши се промена његовог назива, као што је приказано на Сл. 24. Промену назива улазно/излазних сигнала могуће је такође извршити и десним кликом на сигнал, након чега се појављује падајући мени на Сл. 25 у коме је потребно одабрати опцију *Rename port*.



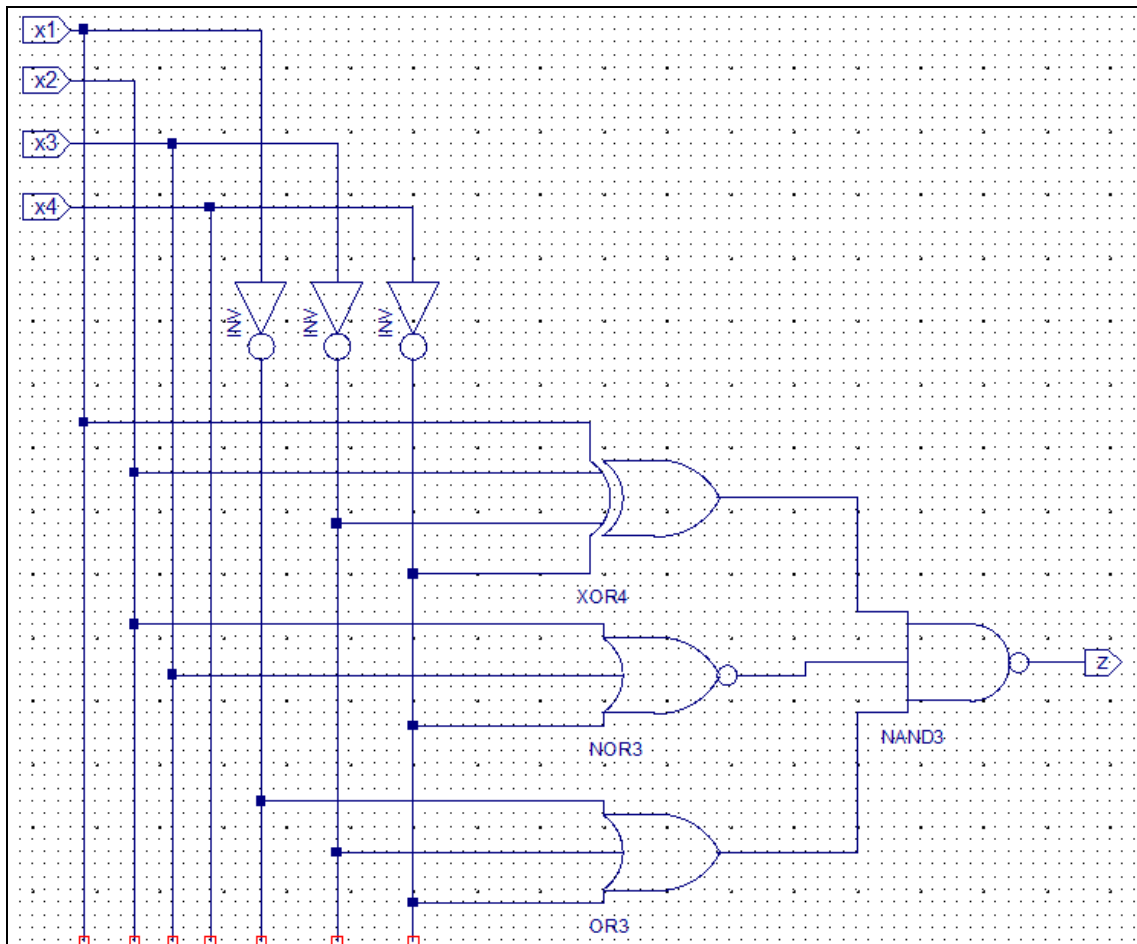
Слика 24. Измена назива улазної/излазної сиїнала – један начин



Слика 25. Измена назива улазної/излазної сиїнала – груїи начин

На Сл. 26 приказан је коначан изглед дигиталног мреже, чија коректност се може проверити коришћењем опције *Check schematic* (Сл. 19). Ради прегледности шематског приказа

дигиталне мреже, препоручљиво је креирати вертикалне линије за све улазне сигнале и њихове комплементе, као што је приказано на Сл. 26.

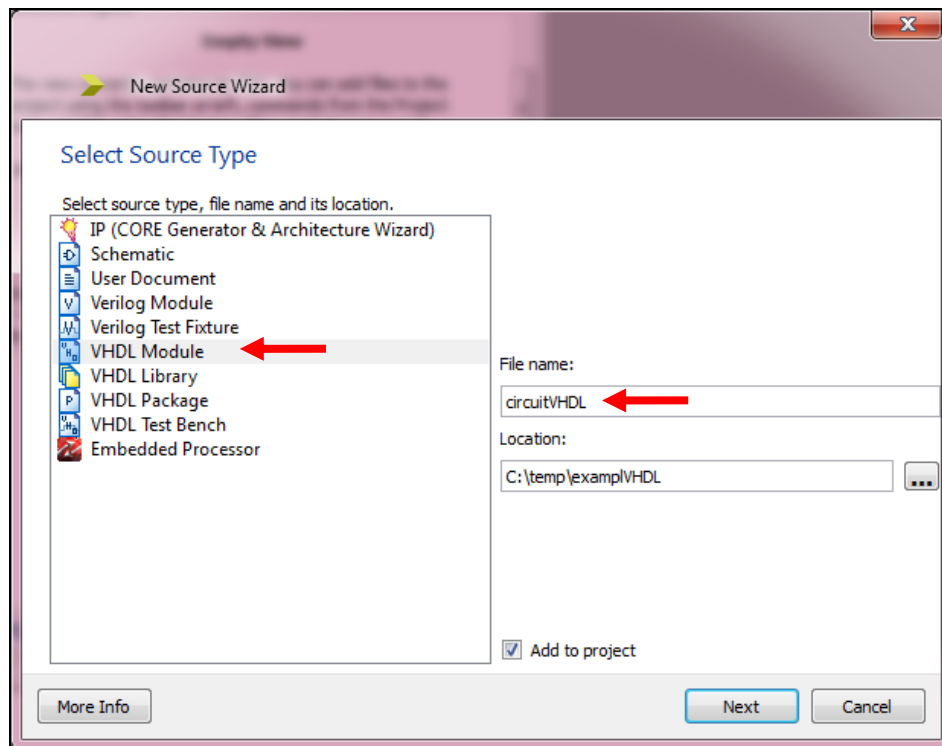


Слика 26. Коначан изглед дигиталне мреже

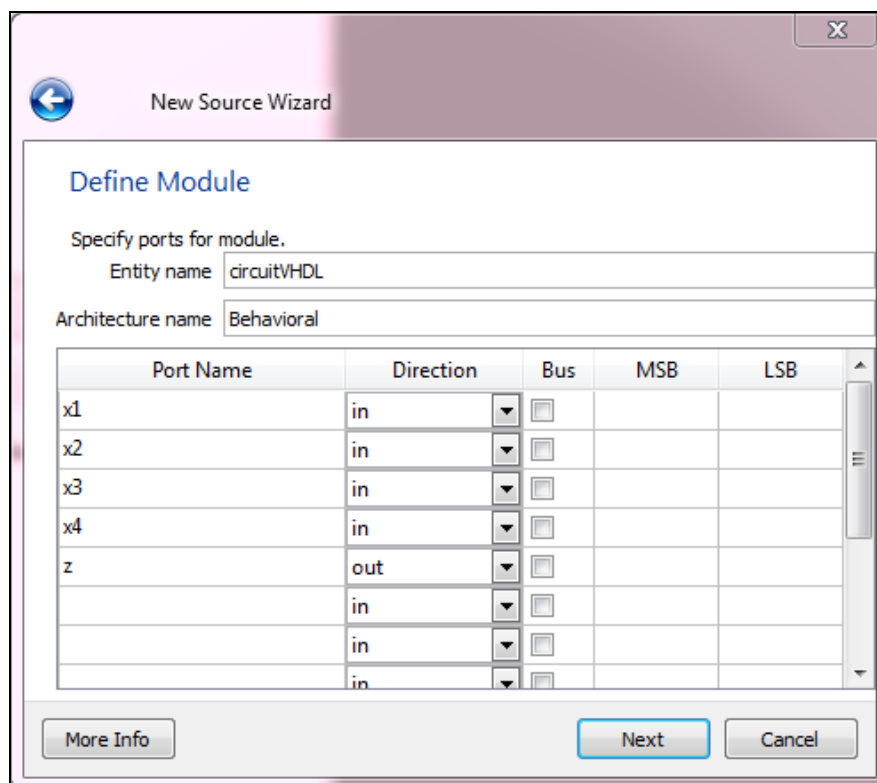
2.2. Креирање *VHDL* фајла и додавање пројекту

У дескриптивне језике за опис хардвера тзв. *HDL* (*Hardware Description Language*) језике спадају следећи језици: *VHDL* (*VHIC* (*Very High Speed IC*)), *Verilog*, *System Verilog* и *SystemC*. У овом поглављу биће описан поступак креирања *VHDL* тип фајла назива *circuitVHDL* и његово додавање креираном пројекту у *Xilinx ISE Design Suite* софтверском алату.

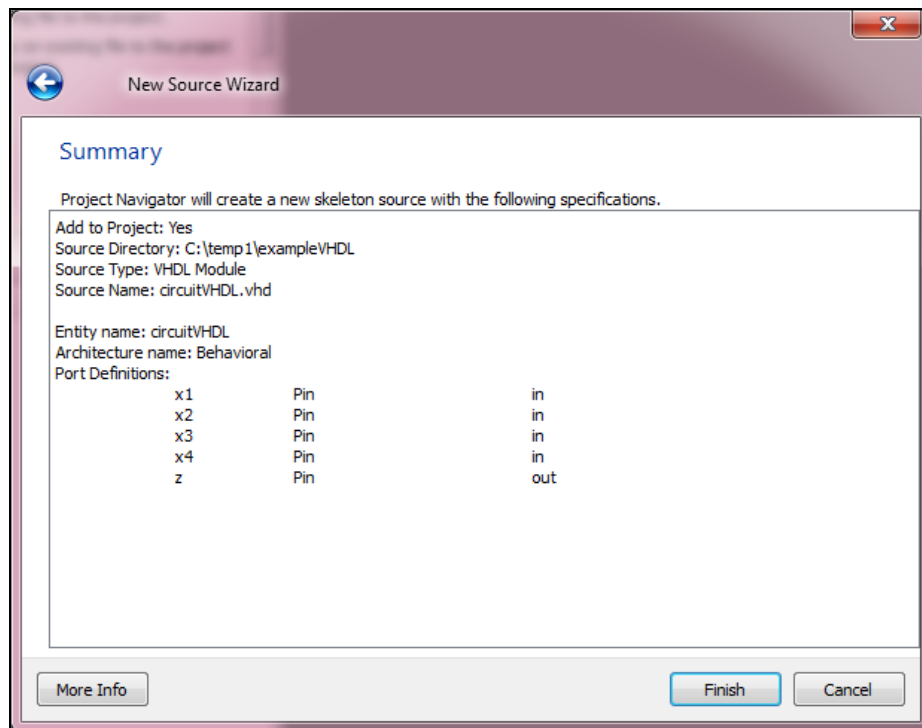
Након креирања пројекта у *Xilinx ISE Design Suite* софтверском алату, према поступку описаном у поглављу 1, потребно му је додати фајл који ће да садржи изворни програмски код (*source*), десним кликом на пројекат у прозору 1 на Сл. 14 и избором опције *New Source* (Сл. 15). Овом приликом се стартује чаробњак у коме је потребно дефинисати тип и назив фајла који се креира, при чему је у овом случају потребно одабрати *VHDL Module* тип фајла (Сл. 27).



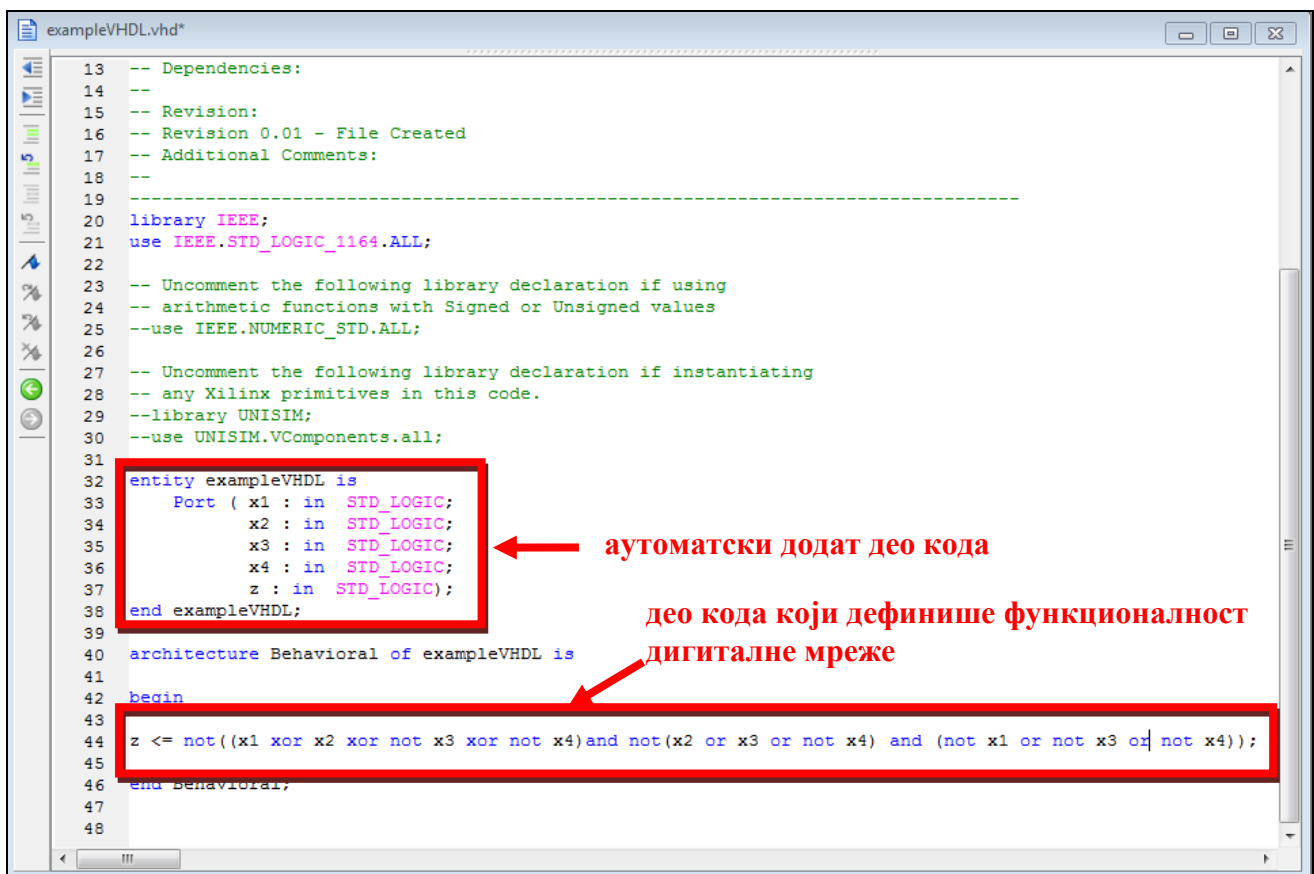
Слика 27. Дефинисање назива и VHDL Module типџа фајла који се додаје креираном пројекту



Слика 28. Дефинисање назива улазних/излазних портова



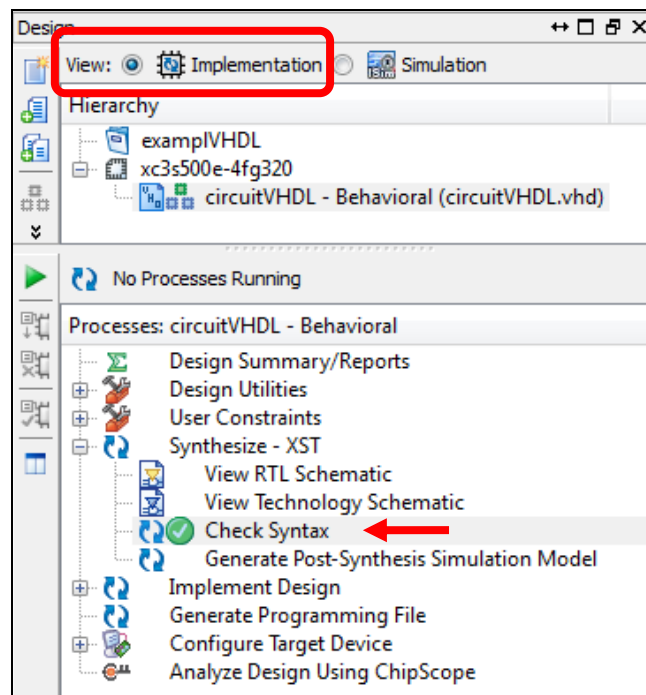
Слика 29. Дефинисане спецификације VHDL Module фајла



Слика 30. Садржај VHDL изворної фајла

Надаље се отвара прозор на Сл. 28 на коме се врши спецификација назива улазних и излазних портова. С обзиром да ће и у овом примеру бити показан начин реализације дигиталне мреже са Сл. 20, називи улазних портова, које треба специфицирати су: $x1$, $x2$, $x3$ и $x4$, док је назив излазног порта z . На Сл. 29 приказане су све спецификације креираног изворног *VHDL Module* фајла.

На Сл. 30 приказан је аутоматски додат део *VHDL* кода изворног фајла, који представља спецификацију улазних и излазних портова, као и део *VHDL* кода кога треба додати да би се дефинисала функционалност дигиталне мреже са Сл. 20. За проверу синтаксе *VHDL* кода изворног фајла потребно је док је селектован имплементациони приказ пројекта и посматрани *VHDL* фајл двоструким кликом стартовати процес *Check Syntax* у оквиру процеса *Synthesize - XST* (Сл. 31).

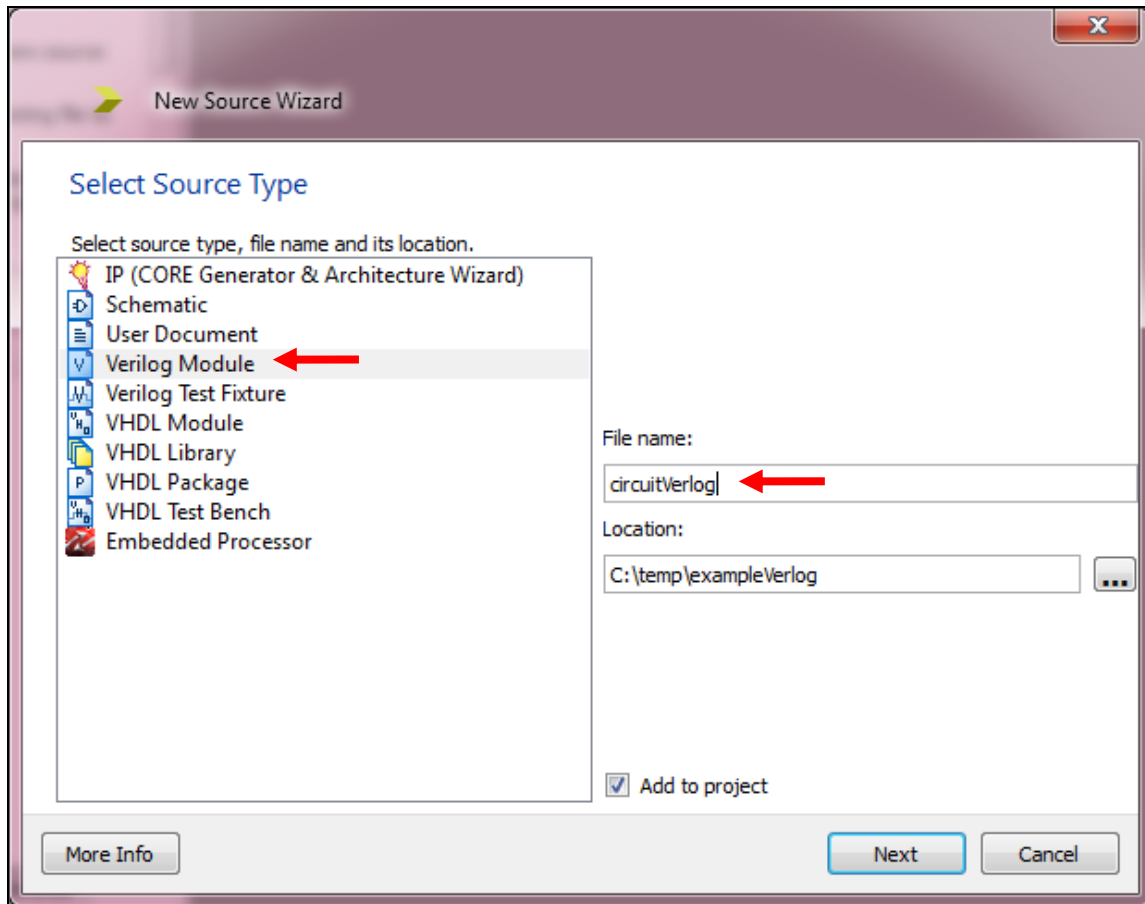


Слика 31. Провера синтаксе *VHDL* изворног фајла

2.3. Креирање *Verilog* фајла и додавање пројекту

У овом поглављу биће описан поступак креирања *Verilog* тип фајла назива *circuitVerilog* и његово додавање креираном пројекту у *Xilinx ISE Design Suite* софтверском алату. Након креирања апликационог пројекта у *Xilinx ISE Design Suite* софтверском алату, према поступку описаном у поглављу 1, потребно му је додати фајл који ће да садржи изворни програмски код (*source*), десним кликом на пројекат у прозору 1 на Сл. 14 и избором опције *New Source* (Сл. 15). Овом приликом се стартује чаробњак у коме је потребно дефинисати тип и назив

фајла који се креира, при чему је у овом случају потребно одабрати *Verilog Module* тип фајла (Сл. 32).

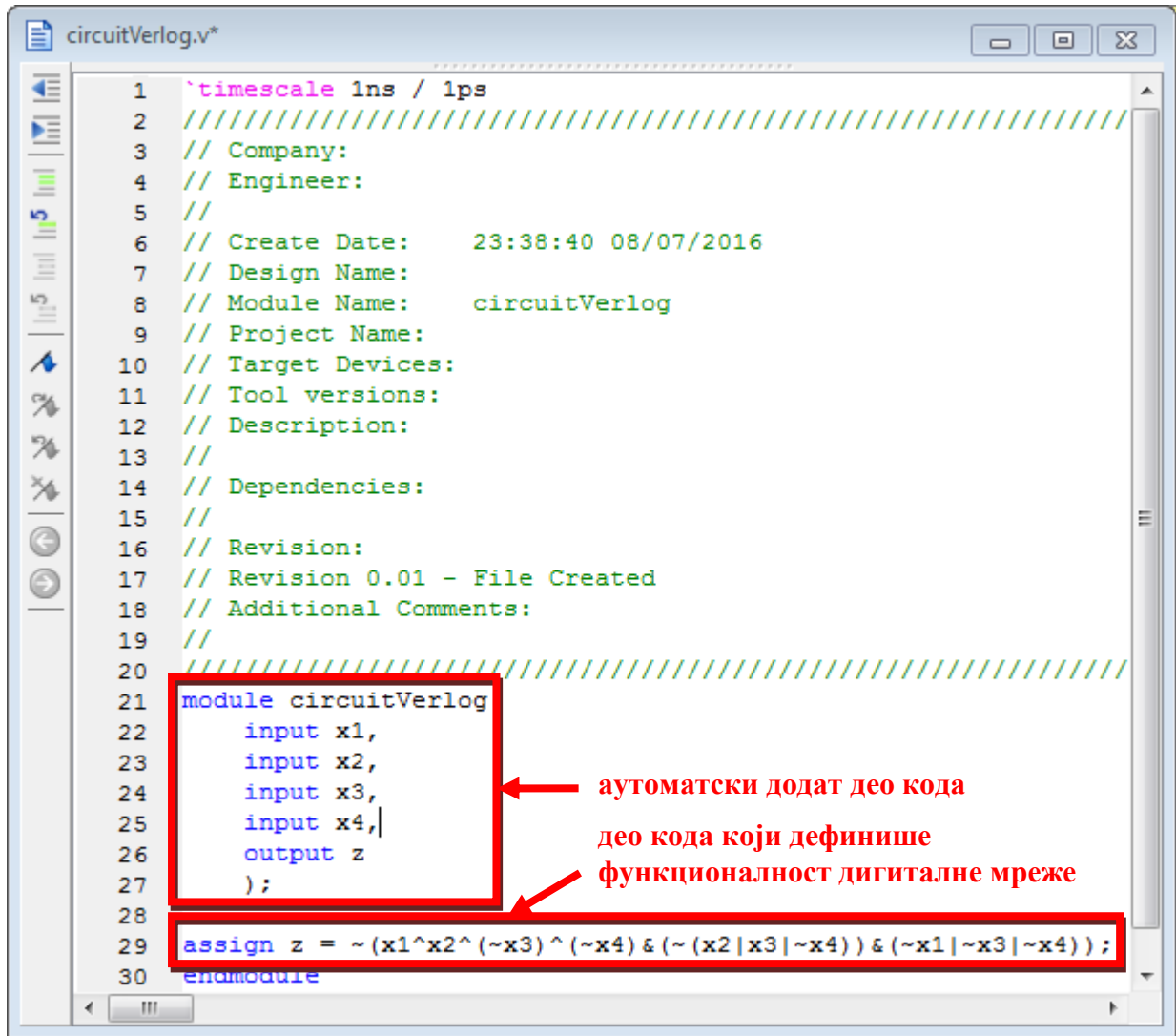


Слика 32. Дефинисање назива и *Verilog Module* типа фајла који се догађе креираном пројекту

Надаље се отвара прозор, попут прозора на Сл. 28 на коме се врши спецификација назива улазних и излазних портова. С обзиром да ће и у овом примеру бити показан начин реализације дигиталне мреже са Сл. 20, називи улазних портова, које треба специфицирати су: $x1$, $x2$, $x3$ и $x4$, док је назив излазног порта z , као што је приказано на Сл. 28. Овом приликом се креирају спецификације креираног *Verilog Module* фајла, попут оних које су приказане на Сл. 29.

На Сл. 33 приказан је аутоматски додат део *Verilog* кода изворног фајла, који представља спецификацију улазних и излазних портова, као и део *Verilog* кода кога треба додати да би се дефинисала функционалност дигиталне мреже са Сл. 20. За проверу синтаксе *Verilog* кода изворног фајла потребно је док је селектован имплементациони приказ пројекта и

посматрани *Verilog* фајл, двоструким кликом стартовати процес *Check Syntax* у оквиру процеса *Synthesize - XST*, као што је приказано на Сл. 31.



```

1  `timescale 1ns / 1ps
2  //////////////////////////////////////////////////
3  // Company:
4  // Engineer:
5  //
6  // Create Date:    23:38:40 08/07/2016
7  // Design Name:
8  // Module Name:    circuitVerlog
9  // Project Name:
10 // Target Devices:
11 // Tool versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 //////////////////////////////////////////////////
21 module circuitVerlog
22     input x1,
23     input x2,
24     input x3,
25     input x4,
26     output z
27 );
28
29 assign z = ~(x1^x2^(~x3)^(~x4) & ~(x2|x3|~x4)) & (~x1|~x3|~x4) ;
30 endmodule
  
```

аутоматски додат део кода

део кода који дефинише функционалност дигиталне мреже

Слика 33. Садржај *Verilog* изворної фајла

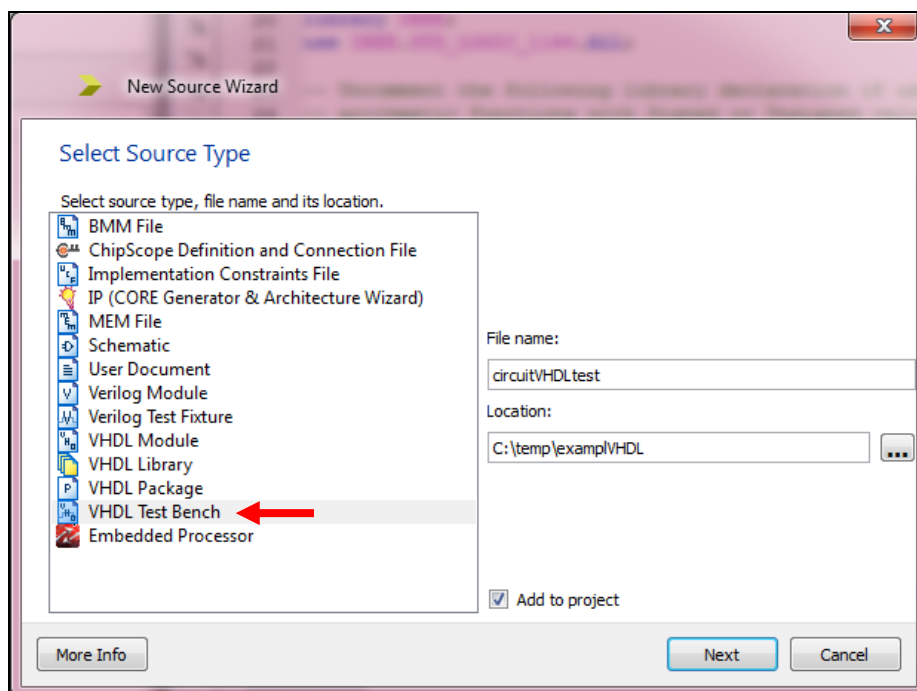
3. СИМУЛАЦИЈА РАДА ДИЗАЈНИРАНЕ ДИГИТАЛНЕ МРЕЖЕ

Симулација рада дизајниране дигиталне мреже подразумева промену вредности излазних сигнала у зависности од унапред задате промене улазних сигнала током одређеног временског периода. Симулација рада дигиталне мреже може се извршити на више начина, коришћењем *ISim* софтвера, при чему ће у овом поглављу бити разматрани следећи начини симулације:

- Симулација коришћењем *VHDL* фајла (видети поглавље 3.1);
- Симулација коришћењем *Verilog* фајла (видети поглавље 3.2);
- Директна симулација у *ISim* софтверу (видети поглавље 3.3);

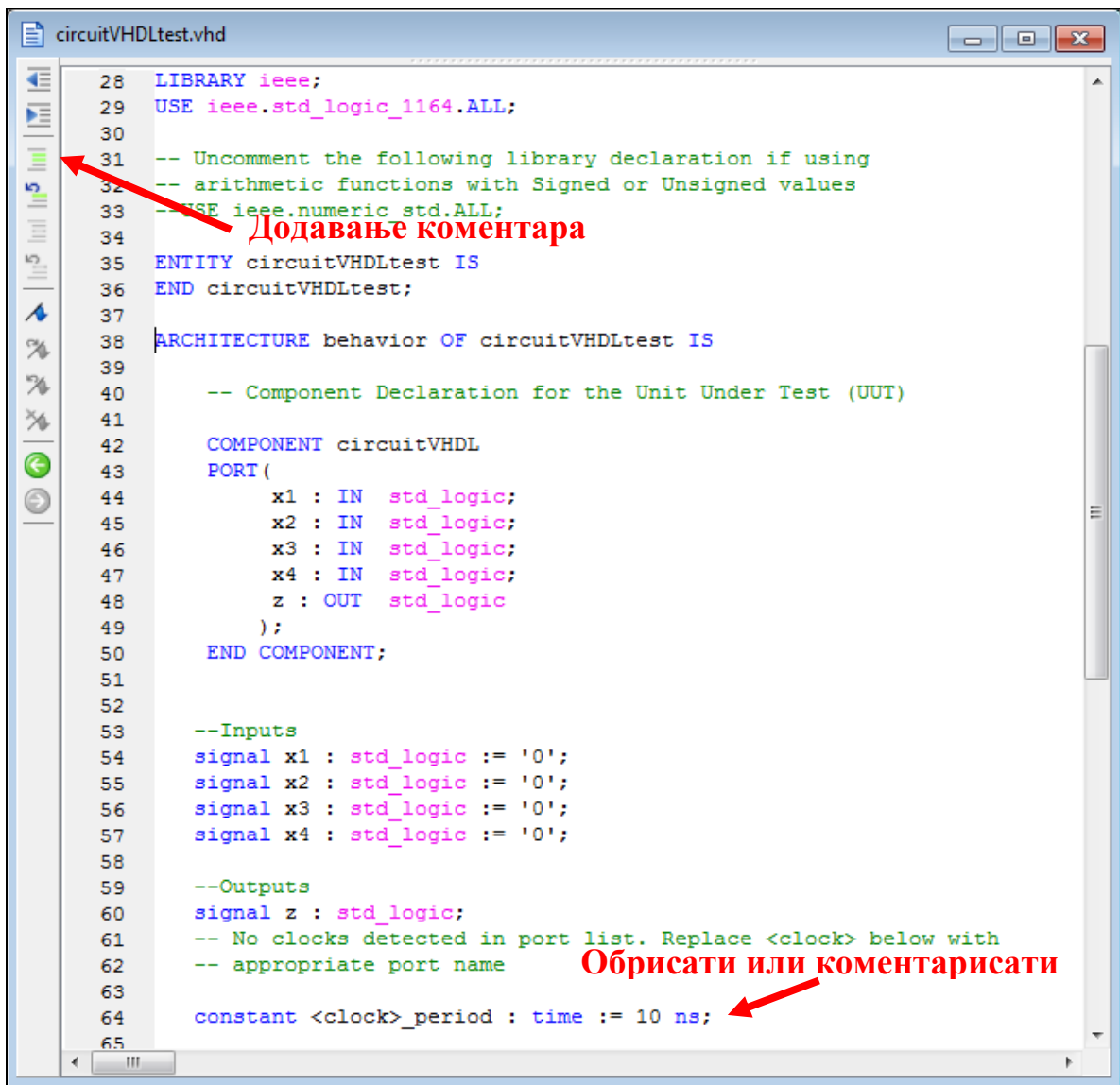
3.1. Симулација рада шематског типа пројекта дигиталне мреже и дигиталне мреже креиране коришћењем *VHDL* језика

Симулација рада дизајниране дигиталне коришћењем шематског типа пројекта, као и дигиталне мреже креиране коришћењем *VHDL* језика може да се врши коришћењем *VHDL Test Bench* фајла у *ISim* софтверу. Најпре је потребно десним кликом на изворни пројектни фајл у прозору 1 на Сл. 14 и избором опције *New Source* креирати нови *VHDL Test Bench* фајл и додати га шематском пројекту или пројекту који се базира на коришћењу *VHDL* језика, као што је приказано на Сл. 34. Том приликом се на основу садржаја шематског фајла или *VHDL* фајла пројекта генерише одговарајући садржај *VHDL Test Bench* фајла, названог *circuitVHDLtest*, чији је део приказан на Сл. 35.



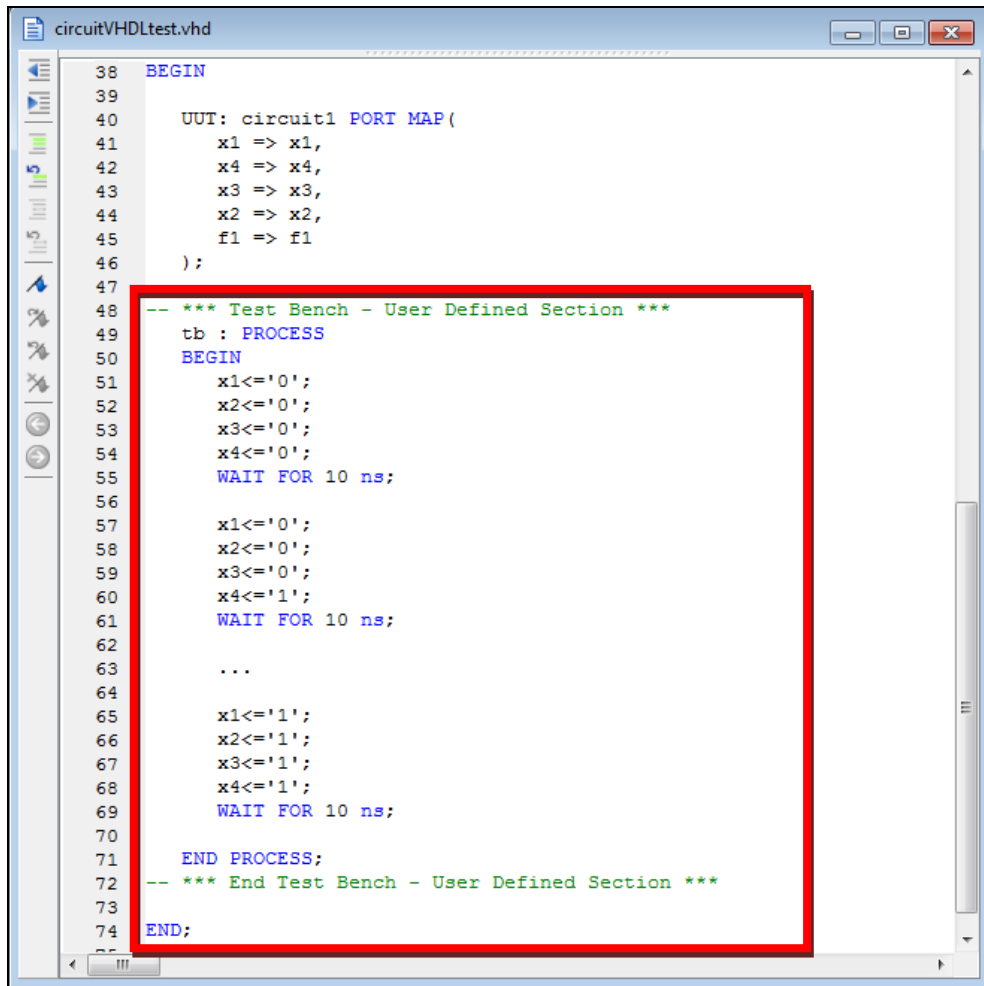
Слика 34. Додавање *VHDL Test Bench* фајла

Уколико се након креирања *VHDL Test Bench* фајла, појаве поруке о грешкама, потребно је обратити пажњу на делове кода у којима се појављује сигнал такта *<clock>*, с обзиром да креирана дигитална мрежа међу улазним сигнаlima не поседује сигнал такта. Стога је у *VHDL Test Bench* фајлу потребно те делове кода обрисати или претворити у коментар селекцијом и кликом на знак за коментар, као што је приказано на Сл. 35.

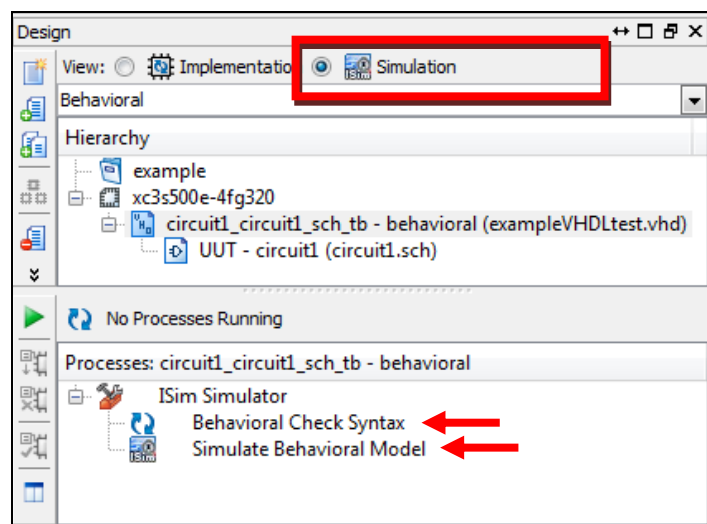


Слика 35. Аутоматски креиран гео садржаја *VHDL Test Bench* фајла

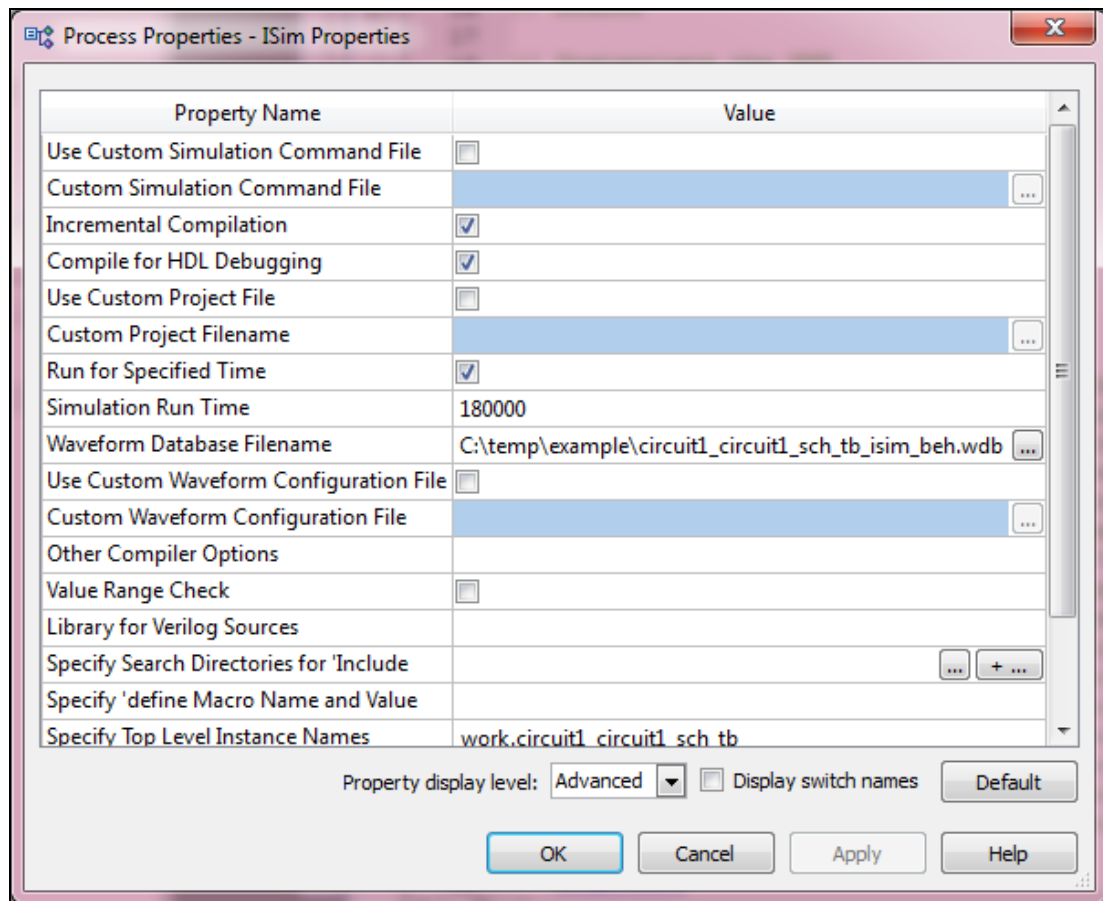
У делу кода *VHDL Test Bench* фајла за дефинисање процеса симулације потребно је улазним сигнаlima доделити све могуће бинарне вредности, са дефинисаним временским паузама од 10 ns, као што је приказано на Сл. 36, при чему на овој слици нису приказане све вредности улазних сигнала, због прегледности. За проверу синтаксе *VHDL Test Bench* фајла потребно двоструким кликом стартовати процес *Behavioral Check Syntax* (Сл. 37).



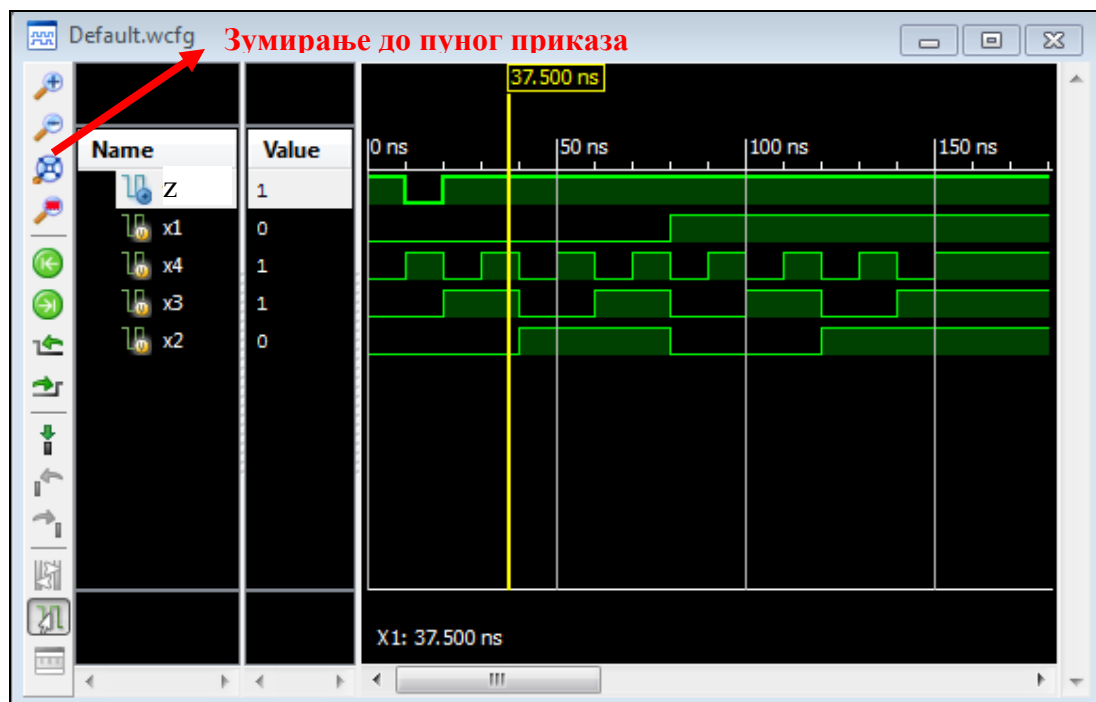
Слика 36. Кориснички додај део садржаја VHDL Test Bench фајл за симулацију начина функционисања дигиталне мреже



Слика 37. Стартовање симулације



Слика 38. Подешавање времена симулације

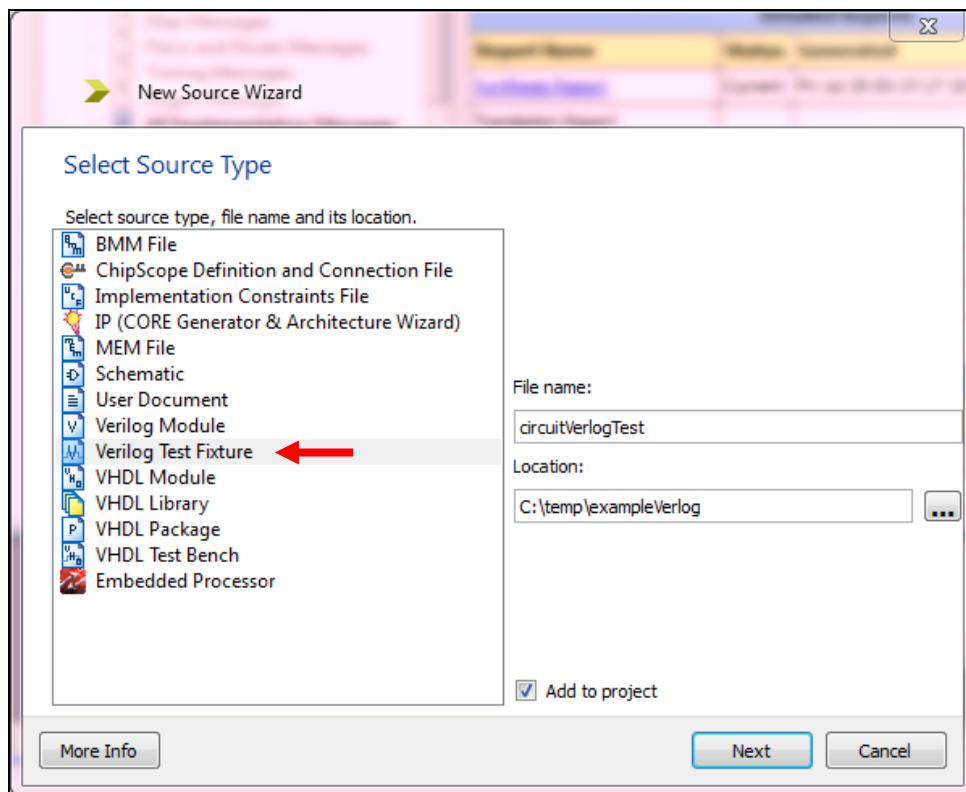


Слика 39. Симулација начина функционисања креиране дигиталне мреже у ISim симулатору

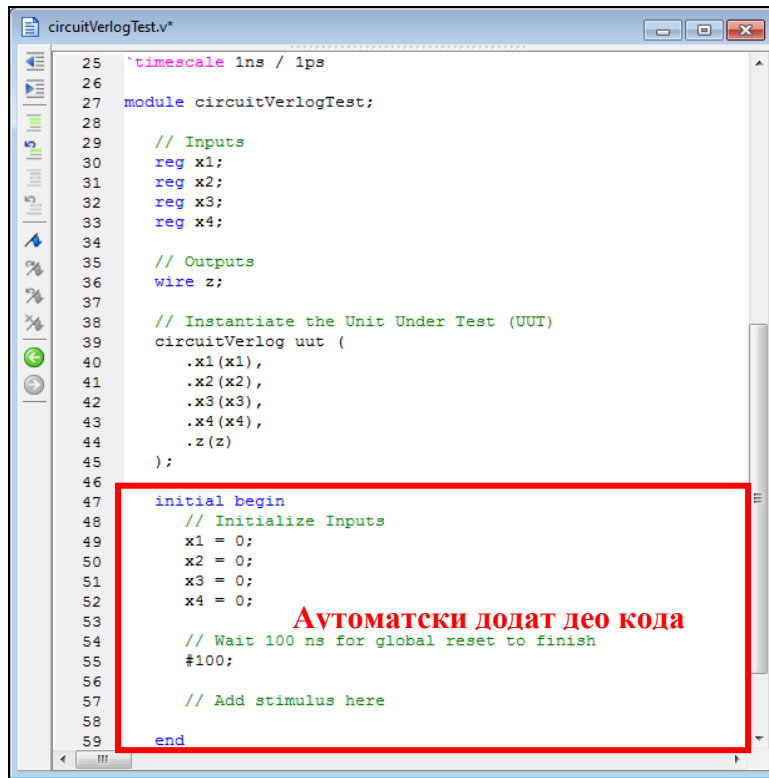
Пре стартовања симулације, потребно је десним кликом на *Simulate Behavioral Model* у падајућем менију одабирати *Process properties* и дефинисати време симулације од 180 000 ps, што одговара времену од 180 ns (Сл. 38). Двоструким кликом на *Simulate Behavioral Model* процес (Сл. 37) стартује се симулација начина функционисања креиране дигиталне мреже у *ISim* симулатору, која је приказана на Сл. 39.

3.2. Симулација рада дигиталне мреже креиране коришћењем Verilog језика

Симулација рада дигиталне мреже, дизајниране коришћењем *Verilog* језика, може да се врши коришћењем *Verilog Test Fixture* фајла и *ISim* софтвера. Најпре је потребно десним кликом на изворни *Verilog* фајл у прозору 1 на Сл. 14 креирати нови *Verilog Test Fixture* фајл и додати га пројекту, као што је приказано на Сл. 40. Том приликом се на основу садржаја *Verilog* фајла пројекта генерише одговарајући садржај *Verilog Test Fixture* фајла, који је назван *circuitVerlogTest* (Сл. 41). У наведеном коду *timescale 1ns/1ps* означава да је временска јединица 1 ns, а резолуција је 1 ps, док *#100* означава временску паузу од 100 временских јединица тј. 100 ns. Наведени код је потребно проширити тако да се у њему врши измена свих могућих вредности улазних сигнала са временском паузом од 10 ns, као што је приказано на сликама 42 и 43, при чему на Сл. 42 нису приказане све вредности улазних сигнала, због прегледности.



Слика 40. Креирање *Verilog Test Fixture* фајла и додавање шематском пројекту

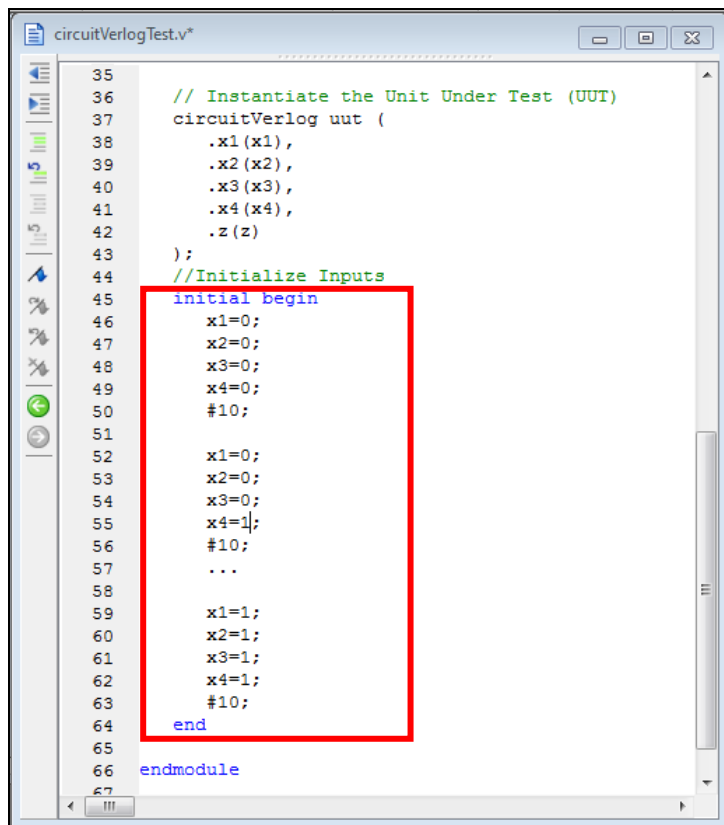


```

25 `timescale 1ns / 1ps
26
27 module circuitVerlogTest;
28
29     // Inputs
30     reg x1;
31     reg x2;
32     reg x3;
33     reg x4;
34
35     // Outputs
36     wire z;
37
38     // Instantiate the Unit Under Test (UUT)
39     circuitVerlog uut (
40         .x1(x1),
41         .x2(x2),
42         .x3(x3),
43         .x4(x4),
44         .z(z)
45     );
46
47     initial begin
48         // Initialize Inputs
49         x1 = 0;
50         x2 = 0;
51         x3 = 0;
52         x4 = 0;
53
54         // Wait 100 ns for global reset to finish
55         #100;
56
57         // Add stimulus here
58
59     end
  
```

Аутоматски додат део кода

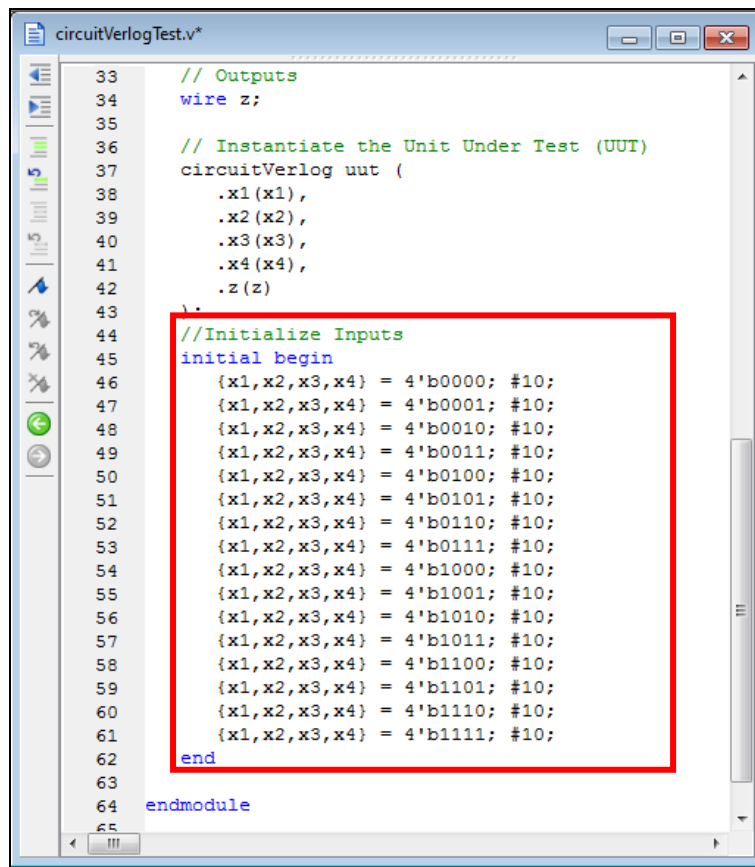
Слика 41. Аутоматски креиран део кода Verilog Test Fixture фајла



```

35
36     // Instantiate the Unit Under Test (UUT)
37     circuitVerlog uut (
38         .x1(x1),
39         .x2(x2),
40         .x3(x3),
41         .x4(x4),
42         .z(z)
43     );
44     //Initialize Inputs
45     initial begin
46         x1=0;
47         x2=0;
48         x3=0;
49         x4=0;
50         #10;
51
52         x1=0;
53         x2=0;
54         x3=0;
55         x4=1;
56         #10;
57         ...
58
59         x1=1;
60         x2=1;
61         x3=1;
62         x4=1;
63         #10;
64     end
65
66 endmodule
67
  
```

Слика 42. Кориснички додати део садржаја Verilog Test Fixture фајла за симулацију начина функционисања дигиталне мреже – први начин



```

33 // Outputs
34 wire z;
35
36 // Instantiate the Unit Under Test (UUT)
37 circuitVerilog uut (
38     .x1(x1),
39     .x2(x2),
40     .x3(x3),
41     .x4(x4),
42     .z(z)
43 );
44
45 //Initialize Inputs
46 initial begin
47     {x1,x2,x3,x4} = 4'b0000; #10;
48     {x1,x2,x3,x4} = 4'b0001; #10;
49     {x1,x2,x3,x4} = 4'b0010; #10;
50     {x1,x2,x3,x4} = 4'b0011; #10;
51     {x1,x2,x3,x4} = 4'b0100; #10;
52     {x1,x2,x3,x4} = 4'b0101; #10;
53     {x1,x2,x3,x4} = 4'b0110; #10;
54     {x1,x2,x3,x4} = 4'b0111; #10;
55     {x1,x2,x3,x4} = 4'b1000; #10;
56     {x1,x2,x3,x4} = 4'b1001; #10;
57     {x1,x2,x3,x4} = 4'b1010; #10;
58     {x1,x2,x3,x4} = 4'b1011; #10;
59     {x1,x2,x3,x4} = 4'b1100; #10;
60     {x1,x2,x3,x4} = 4'b1101; #10;
61     {x1,x2,x3,x4} = 4'b1110; #10;
62     {x1,x2,x3,x4} = 4'b1111; #10;
63 end
64 endmodule
  
```

Слика 43. Кориснички *gdaai* *geo* садржаја *Verilog Test Fixture* фајла за симулацију начина функционисања дигиталне мреже – *групи начин*

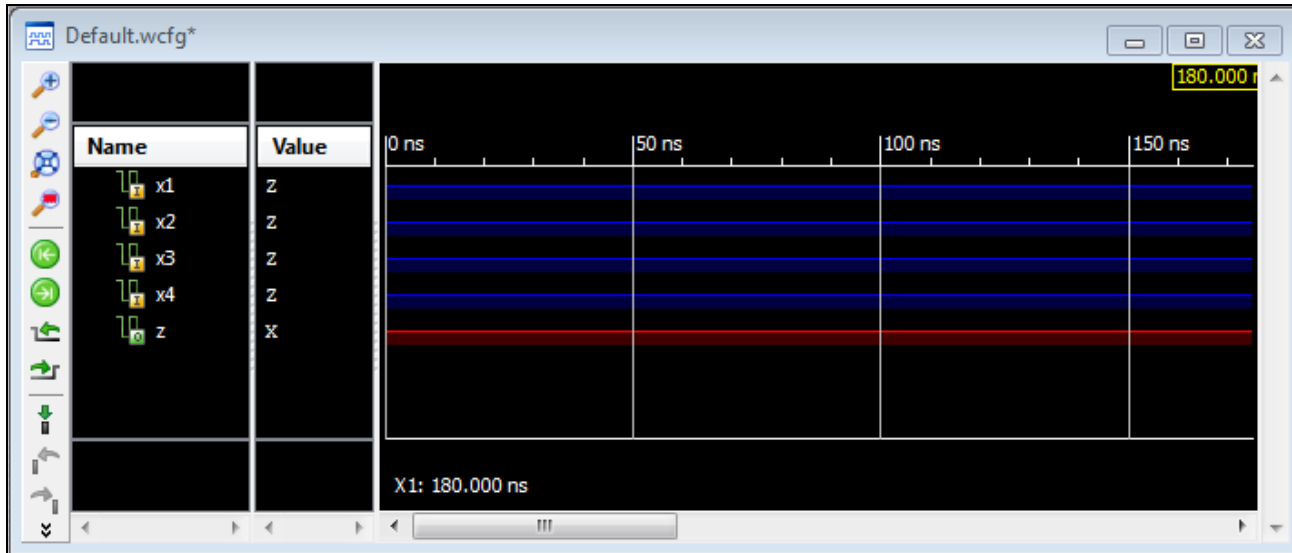
За проверу синтаксе *VHDL Test Bench* фајла потребно двоструким кликом стартовати процес *Behavioral Check Syntax* (Сл. 37). Пре стартовања симулације, потребно је подесити време симулације на 180 000 ps (што одговара времену од 180 ns), на начин који је описан у поглављу 3.2 (Сл. 38). Двоструким кликом на *Simulate Behavioral Model* процес (Сл. 37) стартује се симулација начина функционисања креиране дигиталне мреже у *ISim* симулатору, која је приказана на Сл. 39.

3.3. Директна симулација у *ISim* софтверу

Симулација рада дизајниране дигиталне мреже креиране коришћењем шематског типа пројекта, као и дигиталне мреже креиране коришћењем *VHDL* или *Verilog* језика може да се врши директно у *ISim* софтверу, који се покреће двоструким кликом на *Simulate Behavioral Model* процес (Сл. 37). Пре стартовања симулације, потребно је десним кликом на *Simulate Behavioral Model* падајућег менија одабирати *Process properties* и дефинисати време симулације од 0 ps (Сл. 38).

На Сл. 44 приказан је почетни недефинисан изглед улазно/излазних сигнала. Десним кликом на сваки од улазних сигнала појединачно са Сл. 44 и одабиром опције *Force Constant*

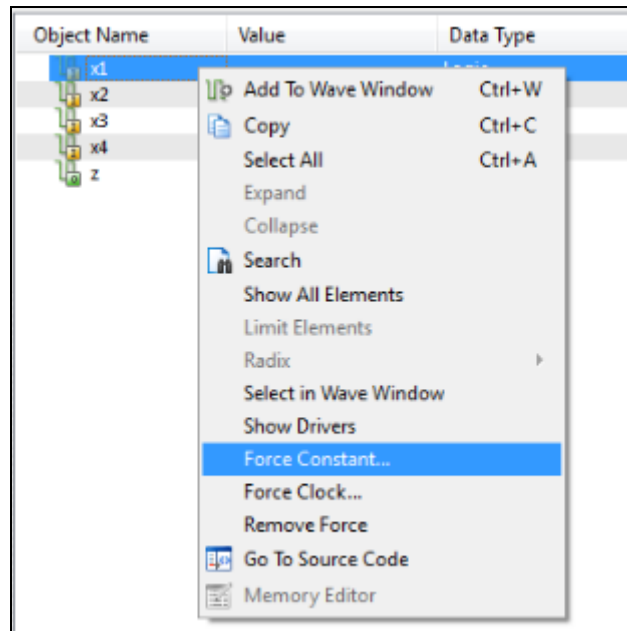
падајућег менија са Сл. 45 отвара се прозор у коме је могуће специфицирати константне вредности улазних сигнала у дефинисаним временским интервалима (Сл. 46a)). У случају сигнала такта могуће је дефинисати периодичне поворке правоугаоних импулса (Сл. 46b), одабиром опције *Force Clock* падајућег менија са Сл. 45.



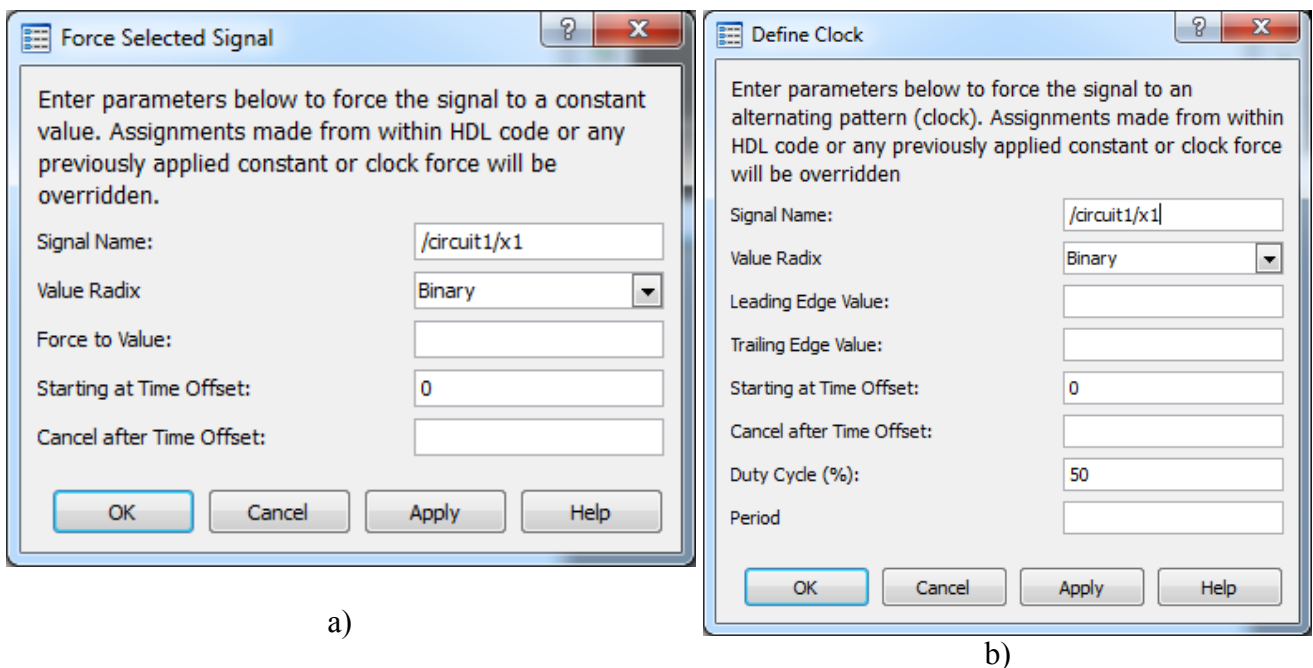
Слика 44. Прозор за графички приказ вредности улазно излазних сигнала у ISim софтверу

У прозору за форсирање константне вредности улазних сигнала *Signal Name* означава назив улазног сигнала, *Value Radix* падајућа листа омогућава избор бројног система (бинарни, октални, хексадецимални, децимални означени, децимални не-означени број, итд.) у коме ће бити дефинисана вредност улазних сигнала у текстуалном пољу *Force to Value* (Сл. 46a)). Текстуална поља *Starting at Time Offset* и *Cancel after Time Offset* служе за дефинисање почетка и краја временског интервала симулације.

У прозору за дефинисање поворци правоугаоних импулса, текстуално поље *Leading Edge Value* служи за дефинисање вредности предњег дела интервала такта, док *Trailing Edge Value* текстуално поље служи за дефинисање задње вредности интервала такта. *Duty cycle (%)* текстуално поље служи за процентуалну спецификацију временског периода активне вредности сигнала такта, док текстуално поље *Period* служи за дефинисање периоде сигнала такта.



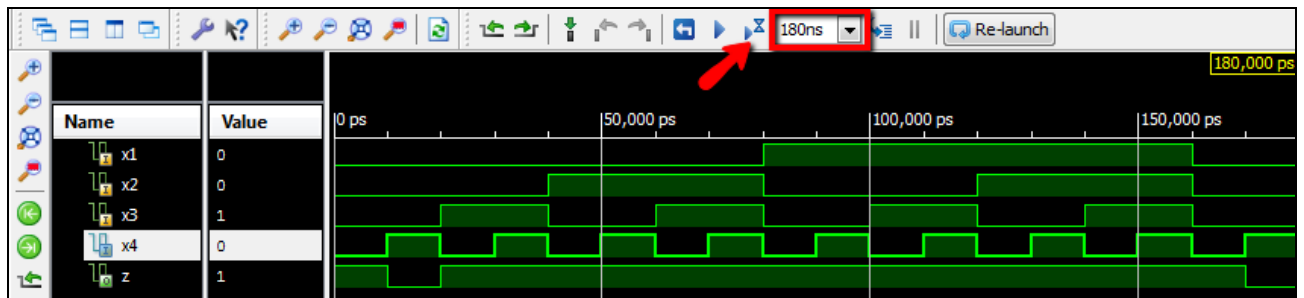
Слика 45. Падајући мени за дефинисање вредности улазних сигнала



Слика 46. Подешавање вредности и времена симулације за а) улазни сигнал б) сигнал тактиа

Да би се добили резултати симулације, који су приказани на Сл. 39, сигнали $x1$, $x2$, $x3$ и $x4$, треба да буду форсирани као поворке правоугаоних имплуса при чему, периода сигнала $x1$ треба да буде подешена на 160ns, сигнала $x2$ на 80ns, $x3$ на 40ns, а $x4$ на 20ns.

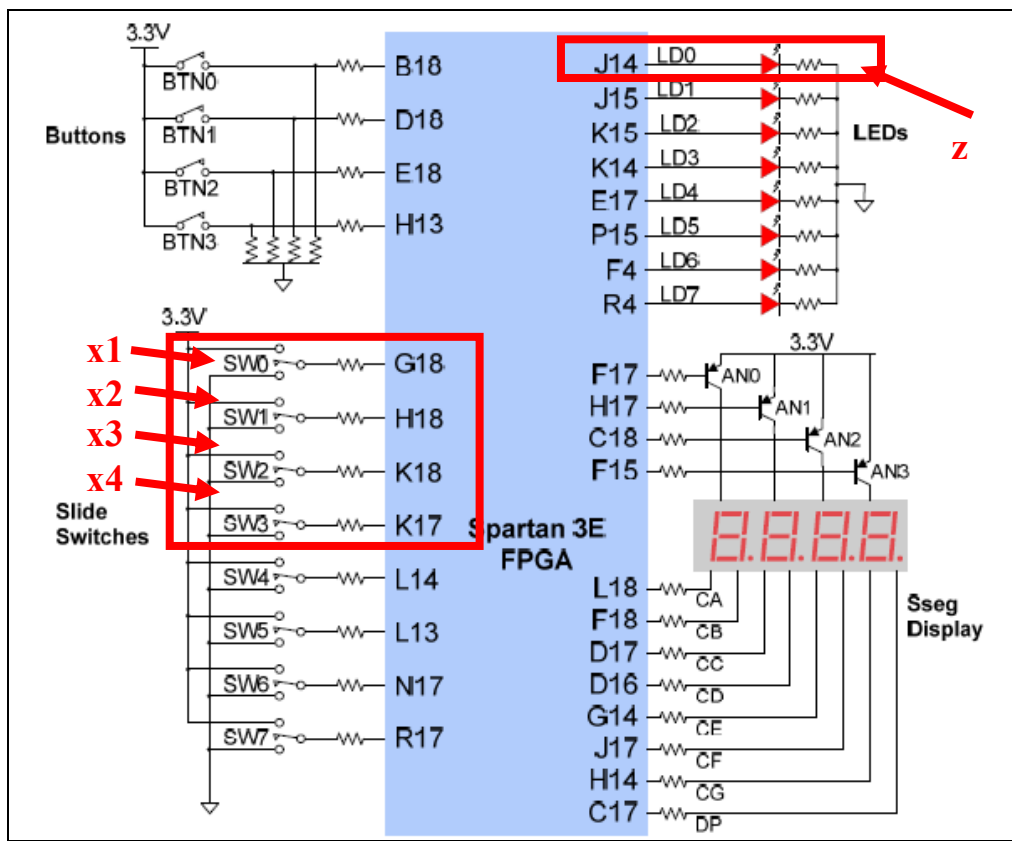
Након форсирања вредности улазних сигнала потребно је подесити време симулације на 180ns, као што је приказано на Сл. 47 и покренути симулацију у специфицираном времену.



Слика 47. Подешавање времена симулације и њено покретању специфицираном времену

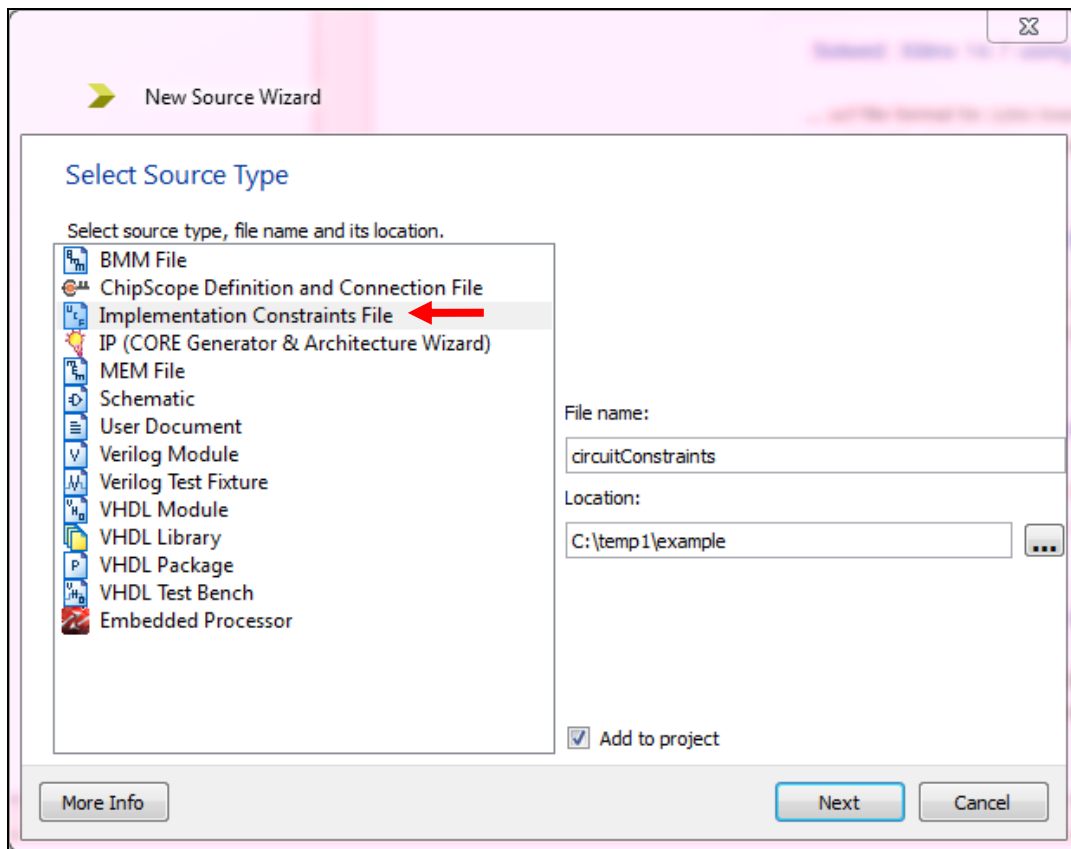
4. ИМПЛЕМЕНТАЦИЈА ДИЗАЈНИРАНЕ ДИГИТАЛНЕ МРЕЖЕ

За имплементацију реализованог шематског, *VHDL* или *Verilog* пројекта на *FPGA* интегрисаном колу најпре је потребно уместо симулационог селектовати имплементациони преглед пројекта (Сл. 31). Након тога, потребно дефинисати локације улазно/излазних сигнала. Улазно/излазни сигнали се могу одабрати из групација *Side Switchers*, *Push Buttons*, *LED* диода и 7-сегментног дисплеја (Сл. 48). За то је потребно десним кликом на изворни фајл у прозору 1 на Сл. 14 креирати фајл корисничких ограничења (*Implementation Constraints File*) са екстензијом *.ucf* и додати га пројекту (Сл. 49).

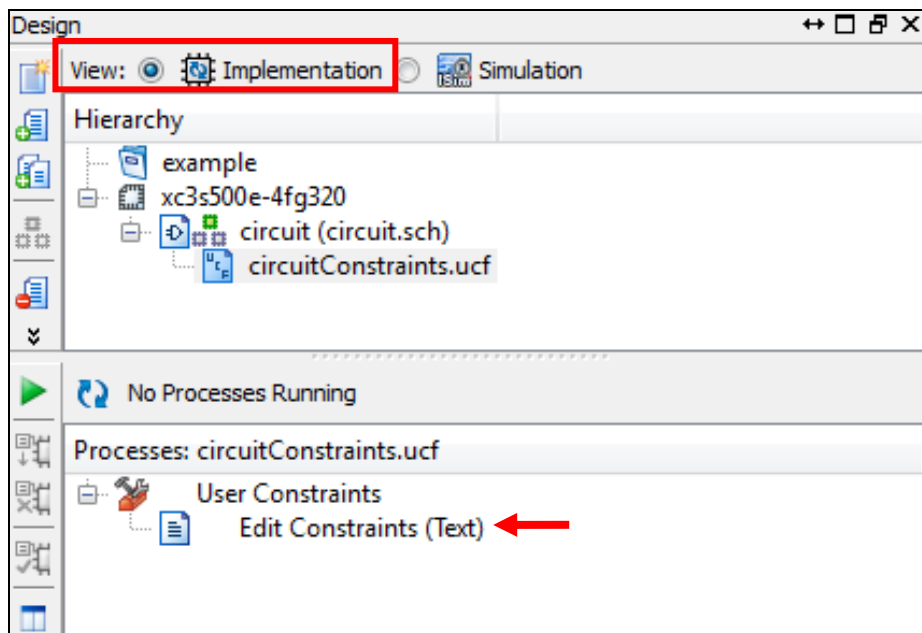


Слика 48. Шематски приказ улазно/излазних (I/O) уређаја Digilent Nexys 2 платформе

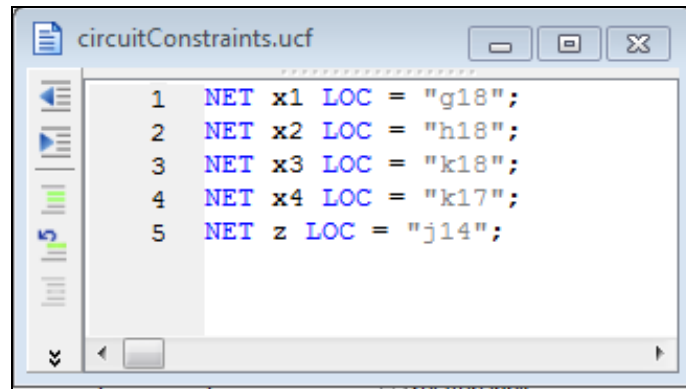
Креирани фајл корисничких ограничења се уређује двоструким кликом на *Edit Constraints (Text)* у процесном прозору (Сл. 50). У нашем случају улазни сигнали одабрани су из групације прекидача *Slide Switchers* (SW0:3), док је излазни сигнал одабран из групације *LED* диода (LD0) (Сл. 48). Имајући у виду одабране улазно/излазне сигнале и уређаје, на Сл. 51 приказан је садржај фајла ограничења у коме *NET* дефинише назив улазно/излазних портова, а *LOC* означава њихову локацију која је одређена према Сл. 48.



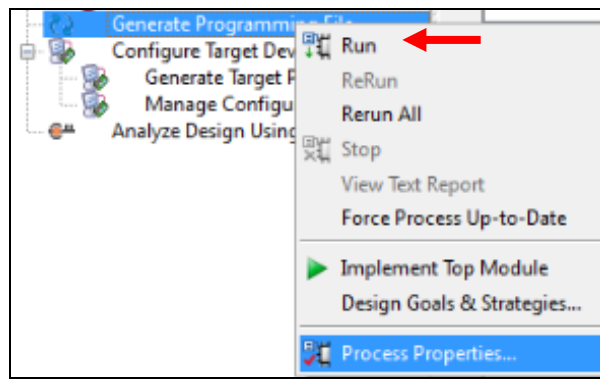
Слика 49. Креирање фајла за дефинисање корисничких оґраничења



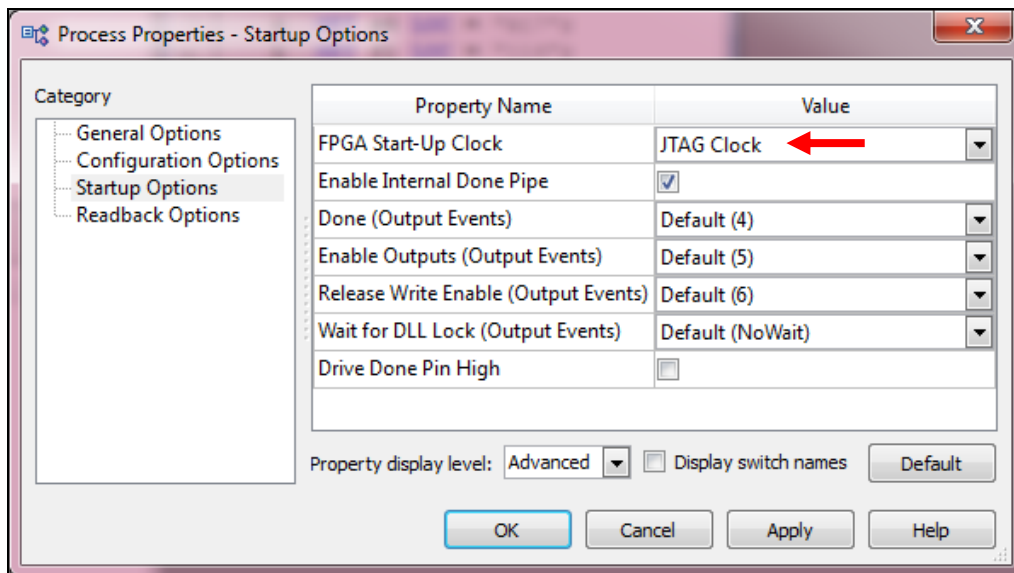
Слика 50. Измена фајла корисничких оґраничења



Слика 51. Садржај фајла оґраничења .ucf



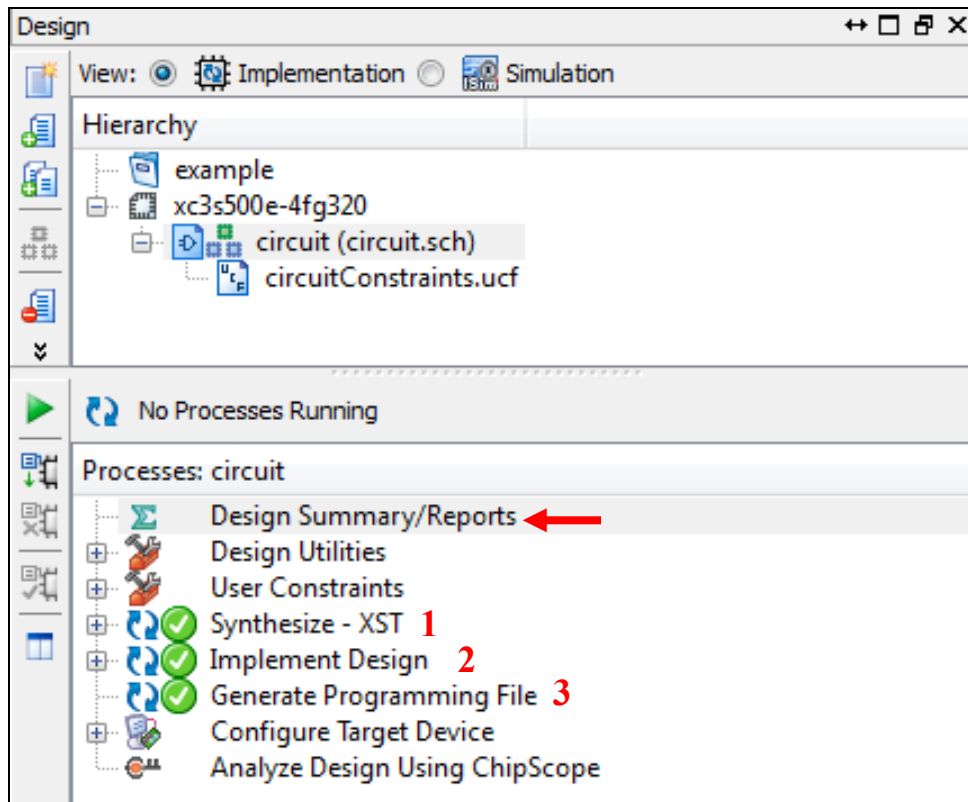
Слика 52. Падајући мени за дефинисање својстава процеса сшаршовања имілементације



Слика 53. Дефинисање својстава процеса сшаршовања имілементације

Уколико је један од улазних сигнала сигнал такта, као што је то чест случај у примерима у којима је потребно дизајнирати секвенцијалне дигиталне мреже, онда је у фајлу оґраничења потребно дефинисати његову подразумевану локацију „b8“, као што је приказано на Сл. 5b).

Пре процеса стартовања имплементације, потребно је десним кликом на *Generate Programming File* одабрати опцију *Process Properties* падајућег менија (Сл. 52). Тада се отвара прозор приказан на Сл. 53 у коме је потребно селектовати *Startup Options* и одабрати *JTAG Clock* за *FPGA Start-Up Clock*.



Слика 54. Кораци имплементације

Имплементација дизајниране дигиталне мреже се постиже двоструким кликом на *Generate Programming File* или десним кликом на *Generate Programming File*, при чему је потребно одабрати опцију *Run* (Сл. 52). Том приликом се реализују кораци: *Synthesize – XST*, *Implement Design* и *Generate Programming File*, као што је приказано на Сл. 54. Као резултат, креира се бинарни фајл са екстензијом *.bit* у пројектном фолдеру пројекта за програмирање *FPGA* интегрисаног кола.

Након креирања бинарног фајла, специфициране локације улазних сигнала *x1*, *x2*, *x3* и *x4*, као и излазног сигнала *z* се могу видети кликом на *Design Summary/Reports* (Сл. 54) у оквиру *Pinout Report* извештаја, као што је приказано на Сл. 55.

Design Summary (Programming File Generated)

- Design Overview
 - Summary
 - IOB Properties
 - Module Level Utilization
 - Timing Constraints
 - Pinout Report
 - Clock Report
 - Static Timing
- Errors and Warnings
 - Parser Messages
 - Synthesis Messages
 - Translation Messages
 - Map Messages
 - Place and Route Messages
 - Timing Messages
 - Bitgen Messages
 - All Implementation Messages
- Detailed Reports
 - Synthesis Report
 - Translation Report
 - Map Report
 - Place and Route Report
 - Post-PAR Static Timing Report
 - Power Report
 - Bitgen Report
- Secondary Reports
 - ISIM Simulator Log

Pin Number	Signal Name	Pin Usage	Pin Name	Direction	IO Standard	IO Bank Number	Drive (mA)
126	G18	IBUF	IP	INPUT	LVCNOS...	1	
127	H1	DIFFM	IO_L10N_3	UNUSED		3	
128	H2	DIFFM	IO_L10P_3	UNUSED		3	
129	H3	DIFFM	IO_L09N_3	UNUSED		3	
130	H4	DIFFM	IO_L09P_3	UNUSED		3	
131	H5	DIFFM	IO_L08N_3	UNUSED		3	
132	H6	DIFFM	IO_L08P_3	UNUSED		3	
133	H7		VCCO_3			3	
134	H8		GND				
135	H9		GND				
136	H10		GND				
137	H11		GND				
138	H12		VCCO_1			1	
139	H13	IBUF	IP	UNUSED		1	
140	H14	DIFFM	IO_L17P_1	UNUSED		1	
141	H15	DIFFM	IO_L17N_1	UNUSED		1	
142	H16	DIFFM	IO_L16P_1	UNUSED		1	
143	H17	DIFFM	IO_L16N_1/A0	UNUSED		1	
144	H18	IBUF	IP/VREF_1	INPUT	LVCNOS...	1	
145	J1	DIFFM	IO_L12P_3/LHCLK2	UNUSED		3	
146	J2	DIFFM	IO_L12N_3/LHCLK3/IRDY2	UNUSED		3	
147	J3		GND				
148	J4	DIFFM	IO_L11N_3/LHCLK1	UNUSED		3	
149	J5	DIFFM	IO_L11P_3/LHCLK0	UNUSED		3	
150	J6	IBUF	IP/VREF_3	UNUSED		3	
151	J7	IBUF	IP	UNUSED		3	
152	J8		GND				
153	J11		GND				
154	J12	DIFFM	IO_L15P_1/A2	UNUSED		1	
155	J13	DIFFM	IO_L15N_1/A1	UNUSED		1	
156	J14	IOB	IO_L14N_1/A3/RHCLK7	OUTPUT	LVCNOS...	1	12

Pinout Report

Show Columns

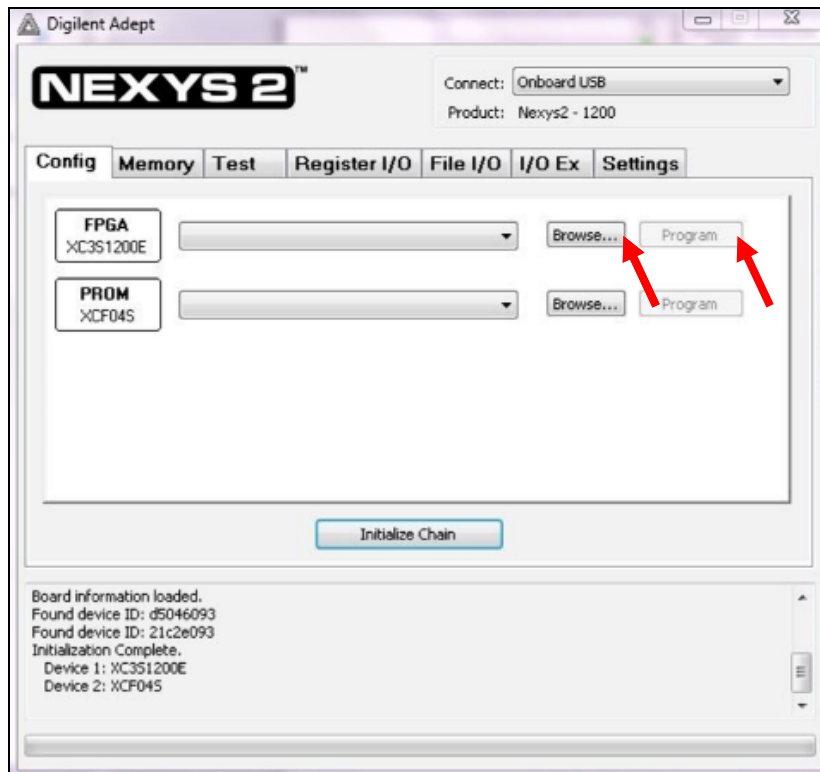
- ☒ Pin Number
- ☒ Signal Name
- ☒ Pin Usage
- ☒ Pin Name
- ☒ Direction
- ☒ IO Standard
- ☒ IO Bank Number
- ☒ Drive (mA)
- ☒ Slew Rate
- ☒ Termination
- ☒ IOB Delay

Слика 55. Део Pinout Report извештаја сумарног прегледа дизајнираној пројектиа дигиталне мреже (Design Summary/Reports)

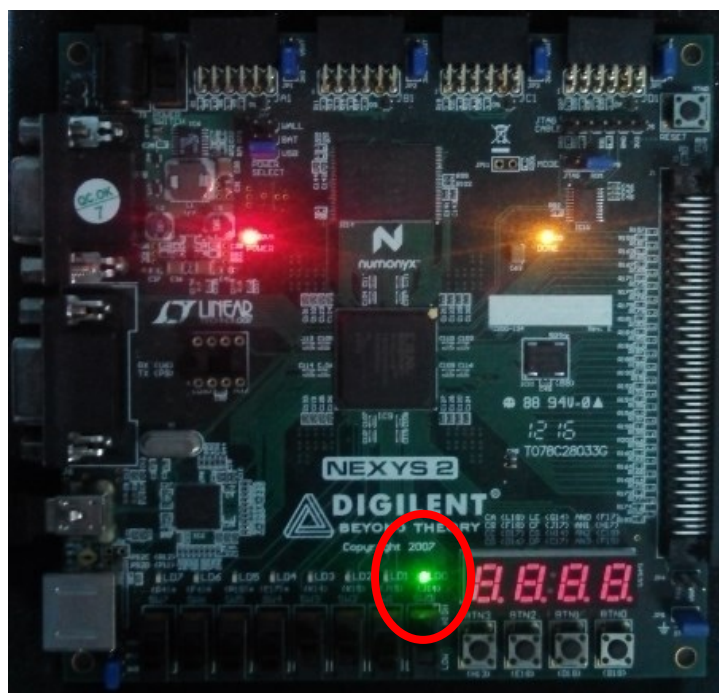
За стартовање програмирања *FPGA* интегрисаног кола, потребно је најпре инсталирати па покренути [Digilent Adept 2](#) софтвер и притиснути дугме *Initialize Chain* (Сл. 56), чиме се у главном прозору појављује могућност програмирања *FPGA* интегрисаног кола и *PROM* меморије *Digilent Nexys 2* платформе. Треба имати у виду и то да се *FPGA* интегрисано коло при нестанку напајања понаша као *RAM* меморија јер се њен садржај губи, за разлику од *PROM* меморије чији садржај се задржава нестанком напајања. Притиском на *Browse* дугме, које одговара *FPGA* интегрисаном колу потребно је учитати креирани *.bit* фајл и притиснути дугме *Program*, чиме се стартује програмирање *FPGA* интегрисаног кола (Сл. 56).

Експеримент је постављен тако да су четири прекидача (*SW0:3*) постављени у положај укључено, а преостала четири (*SW4:7*) у положају искључено (Сл. 48). С обзиром да су у фајлу корисничких ограничења, прекидачи из групе (*SW0:3*) дефинисани као улазни сигнали *x1*, *x2*, *x3* и *x4*, на основу слика 39 и 47, које приказују симулацију начина функционисања

посматране дигиталне мреже, може се закључити да ће вредност излазног сигнала z бити 1. С обзиром да је у фајлу корисничких ограничења као излазни сигнал дефинисана диода $LD0$, резултујући ефекат овог експеримента ће бити светлење ове диоде, као што је приказано на Сл. 56.



Слика 56. Digilent Adept софтверски алај



Слика 57. Резултујући ефекат експеримента – светлење диоде $LD0$

5. ОПИС LIREX ЕКСПЕРИМЕНТА *Nexys 2 FPGA platforma*

LiReX библиотека удаљених експеримената је Веб репозиторијум удаљених експеримената који су развијени у оквиру *Темпус* пројекта [NeReLa](#) [1]. Библиотека удаљених експеримената садржи удаљене експерименте и вежбе које су развијене и постављене у лабораторијама четири партнерска универзитета у Србији: Универзитет у Крагујевцу, Универзитет у Београду, Универзитет у Нишу и Универзитет у Новом Саду.

LiReX експеримент „[Nexys 2 FPGA platforma](#)“ [13] обезбеђује даљински приступ и програмирање *XC3S1200E* типа *FPGA* интегрисаног кола *Digilent Nexys 2* платформе, која се налази у удаљеној лабораторији Факултета инжењерских наука у Крагујевцу. Уносом приступног кључа врши се повезивање даљинским путем преко *CEyeClon Viewer*-а удаљеном рачунару и програмирање *FPGA* интегрисаног кола. Експеримент је постављен тако да су четири прекидача (*SW0:3*) постављени у положај укључено, а преостала четири (*SW4:7*) у положају искључено (Сл. 48). Резултат програмирања може се видети у реалном времену преко веб камере постављене изнад *Nexys 2* плоче.

Потребне припремне активности за извођење експеримента

- Преузети и инсталирати [Xilinx ISE Design Suite](#) софтвер [6];
- Преузети и инсталирати [Digilent Adept 2](#) софтвер [8];
- Преузети и инсталирати [Net Framework 4.5](#) софтвер [10];
- Преузети и инсталирати [CEyeClon Viewer](#) софтвер [11].

Потребна предзнања за извођење експеримента

- Познавање основних компоненти *FPGA* интегрисаног кола и *Digilent Nexys 2* платформе
- Познавање развојног окружења *Xilinx ISE Design Suite* софтвера
- Познавање анализе и синтезе комбинационих и секвенцијалних мрежа

Активности за време извођења експеримента

- Креирање апликационог пројекта у *Xilinx ISE Design Suite* софтверском алату
- Дизајнирање дигиталне мреже на један од следећих начина: коришћењем шематског приступа (видети поглавље 2.1), програмирањем у *VHDL* језику (видети поглавље 2.2) или програмирањем у *Verilog* језику (видети поглавље 2.3)

- Симулација рада дизајниране дигиталне мреже у зависности од начина дизајнирања коришћењем: *VHDL* фајла (видети поглавље 3.1), *Verilog* фајла (видети поглавље 3.2) или пак директна симулација у *ISim* софтверу (видети поглавље 3.3)
- Имплементација дизајниране дигиталне мреже, којом се као коначни резултат креира бинарни фајл са екстензијом *.bit* у пројектном фолдеру
- Логовање на [Lirex](http://lirex.ftn.kg.ac.rs) експеримент „*Nexys 2 FPGA platforma*“ преко *Ceyeclon viewera*. За приступни кључ контактирати администратора на: lirex@ftn.kg.ac.rs
- Креирање фасцикле са корисничким именом и презименом у оквиру постојеће фасцикле *Digilent_experiment* удаљеног сервера
- Копирање бинарног фајла са екстензијом *.bit*, у корисничку креирану фасциклу удаљеног сервера
- Дуплим кликом покренути икону „*pokretanje*“ на десктопу удаљеног сервера, чиме се на њему покреће *Digilent Adept* софтверски алат и стартује камера
- Учитавање корисничког бинарног фајла са екстензијом *.bit* и покретање програмирања *FPGA* интегрисаног кола, коришћењем *Digilent Adept* софтверског алата

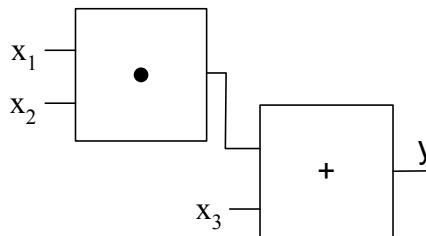
Активности после извођења експеримента

- Приступ камери преко апликације која се налази на десктопу под називом „*AMCAP*“ и визуелна провера тачности решења
- Снимити добијени ефекат у виду видео фајла у оквиру корисничке фасцикле удаљеног сервера

6. ПРИЛОГ – РАЗНИ ПРИМЕРИ ДИЗАЈНИРАЊА ДИГИТАЛНИХ МРЕЖА КОРИШЋЕЊЕМ *FPGA* ИНТЕГРИСАНОГ КОЛА

Пример 1.

Коришћењем *Xilinx ISE Design Suite* софтвера програмирати *FPGA* интегрисано коло за креирање структурне шеме комбинационе мреже на Сл. 58.

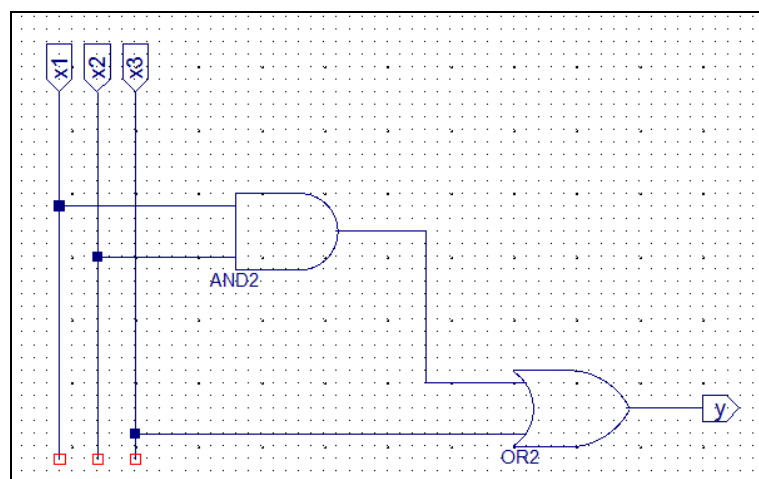


Слика 58. Структурна шема комбинационе мреже

- За дизајнирање задате комбинационе мреже користити шематски приступ (поглавље 2.1);
- Након дизајнирања извршити симулацију коришћењем *ISim* симулатора (поглавље 3);
- Након дизајнирања и симулације, извршити имплементацију прекидачке мреже (поглавље 4), и креирати *.bit* фајл за програмирање *FPGA* интегрисаног кола;
- Приступити експерименту „[Nexys 2 FPGA platform](#)“ и пратити упутства у опису експеримента која су дата у поглављу 5.

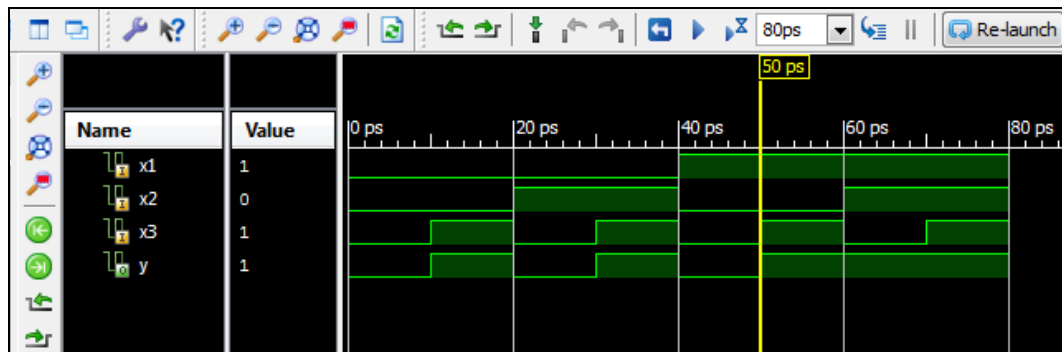
Решење:

- На Сл. 59 дат је шематски приказ задате комбинационе мреже. Шематски фајл је креиран коришћењем поступка који је описан у поглављу 2.1.



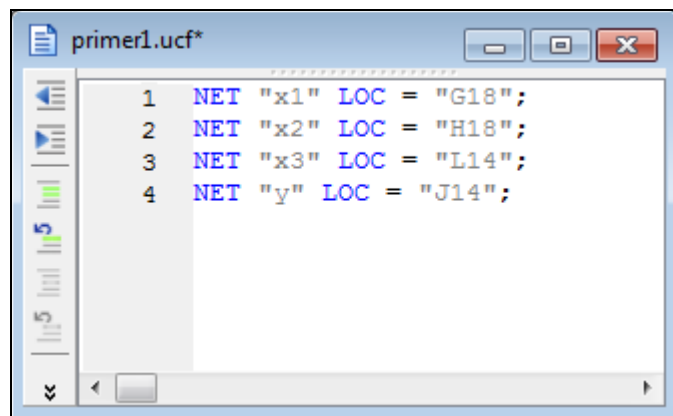
Слика 59. Шематски приказ задате комбинационе мреже

- b) На Сл. 60 је приказан је резултат симулације задате дигиталне мреже у *ISim* симулатору.



Слика 60. Симулација начина функционисања дигиталне мреже у *ISim* симулатору

- c) Будући да је експеримент постављен тако да су четири прекидача (*SW0:3*) укључена, а преостала четири (*SW4:7*) искључена (Сл. 48), за улазне сигнале *x1* и *x2* одабрани су укључени прекидачи *SW0* и *SW1*, док је за улазни сигнал *x3* одабран искључен прекидачи *SW4*. За излазни сигнал *y* одабрана је диода *LD0*. На Сл. 61 је приказан *.ucf* фајл корисничких ограничења, имајући у виду распоред улазно/излазних (I/O) уређаја *Digilent Nexys 2* платформе, који је дат на Сл. 48.



Слика 61. Фајл корисничких ограничења *.ucf*

- d) На основу Сл. 60, која приказује симулацију начина функционисања посматране дигиталне мреже, може се закључити да ће резултујући ефекат овог експеримента бити светлење диоде *LD0*, као што је приказано на Сл. 57.

Пример 2.

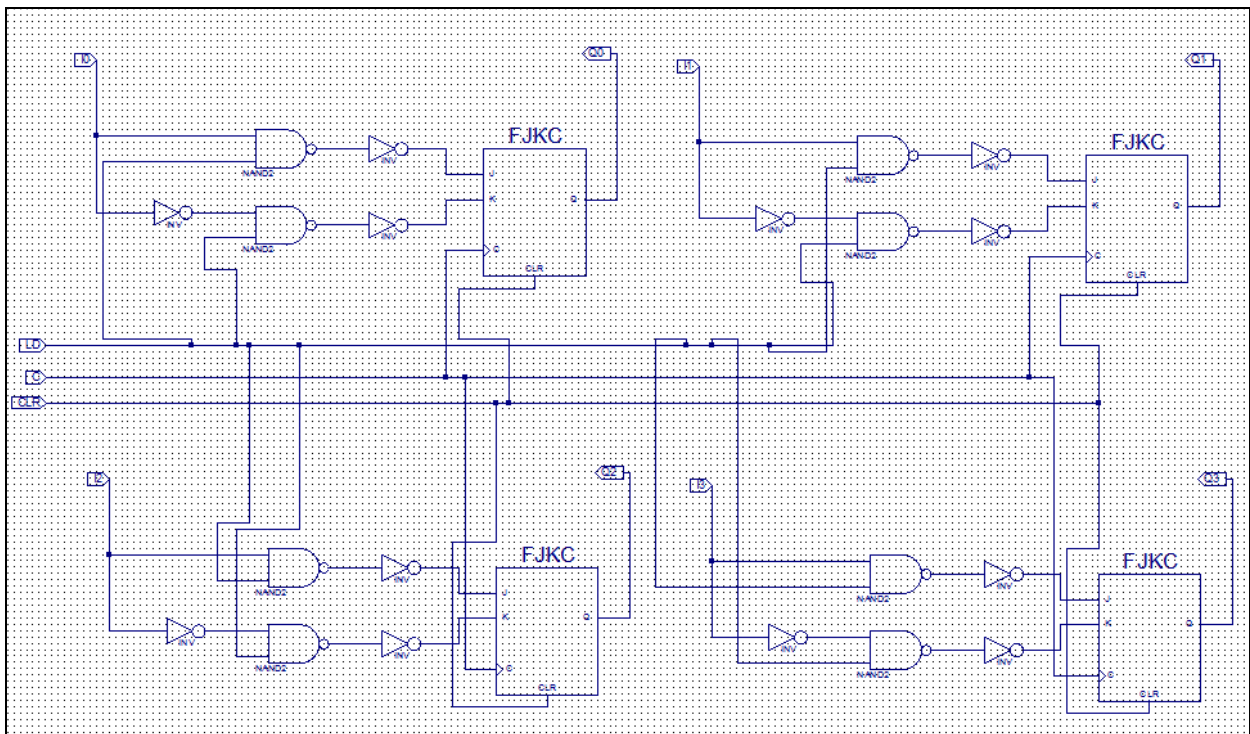
Коришћењем *Xilinx ISE Design Suite* софтвера програмирати *FPGA* интегрисано коло за креирање структурне шеме четворо-разредног регистра са паралелним уписом и читањем и

JK флип флоповима код којих је 0 активна вредност улазних сигнала, код кога се уписом управља помоћу LD сигнала, независно од сигнала такта.

- За дизајнирање задате секувенцијалне мреже користити шематски приступ (поглавље 2.1);
- Након дизајнирања извршити симулацију за следеће вредности улазних сигнала $LD = 1$, $RD = 1$, $I0 = 1$, $I1 = 1$, $I2 = 0$ и $I3 = 0$ коришћењем *ISim* симулатора (поглавље 3);
- Након дизајнирања и симулације, извршити имплементацију задате прекидачке мреже (поглавље 4), и креирати *.bit* фајл за програмирање *FPGA* интегрисаног кола;
- Приступити експерименту „[Nexys 2 FPGA platforma](#)“ и пратити упутства у опису експеримента која су дата у поглављу 5.

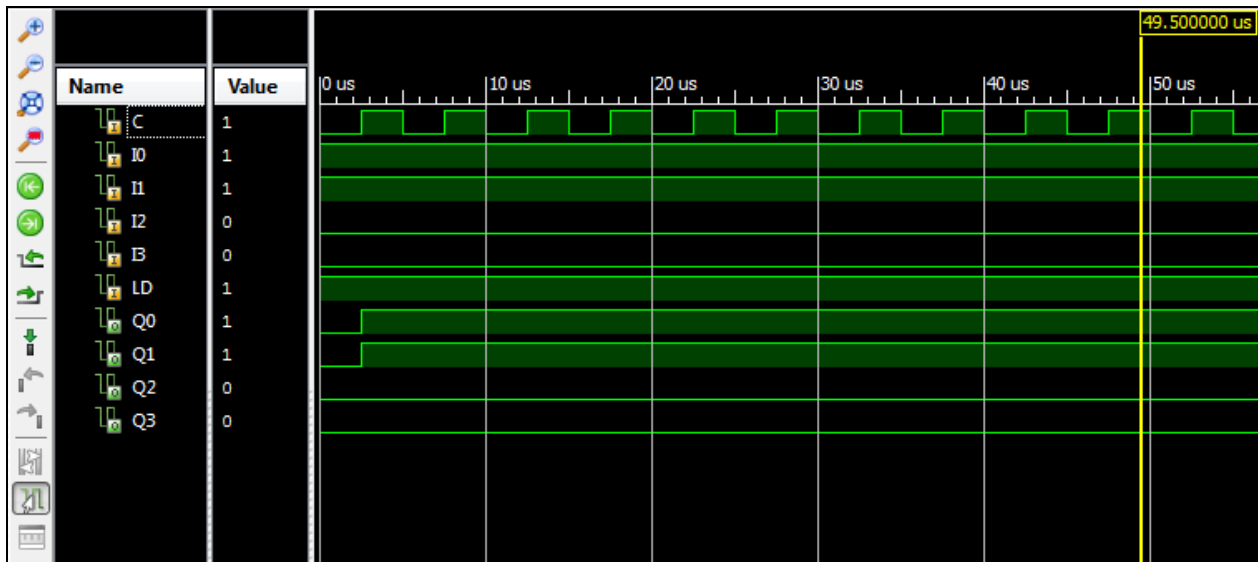
Решење:

- На Сл. 62 дат је шематски приказ четворо-разредног регистра са паралелним уписом и читањем и JK флип флоповима код којих је 0 активна вредност улазних сигнала, код кога се уписом управља помоћу LD сигнала, независно од сигнала такта. Шематски фајл је креиран коришћењем поступка који је описан у поглављу 2.1.



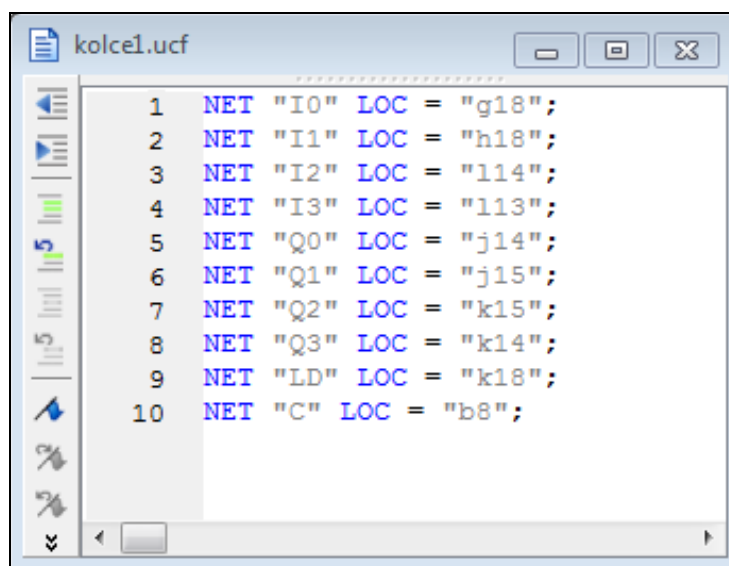
Слика 62. Шематски приказ четворо-разредног регистра

- На Сл. 63 је приказан је резултат симулације четворо-разредног регистра у *ISim* симулатору.



Слика 63. Симулација начина функционисања четворо-разредної ретисира у ISim симулятору

- с) Будући да је експеримент постављен тако да су четири прекидача ($SW0:3$) укључена, а преостала четири ($SW4:7$) искључена (Сл. 48), за улазне сигнале $I0$ и $I1$ одабрани су укључени прекидачи $SW0$ и $SW1$, док су за улазне сигнале $I2$ и $I3$ одабрани искључени прекидачи $SW4$ и $SW5$. За управљачке сигнале LD и RD одабрани су укључени прекидачи $SW2$ и $SW3$. Излазни сигнали $Q0$, $Q1$, $Q2$ и $Q3$ одабрани су из групе диода $LD0$, $LD1$, $LD2$ и $LD3$ респективно. На Сл. 64 је приказан *.ucf* фајл корисничких ограничења, имајући у виду шематски распоред улазно/излазних (I/O) уређаја Digilent Nexys 2 платформе (Сл. 48).

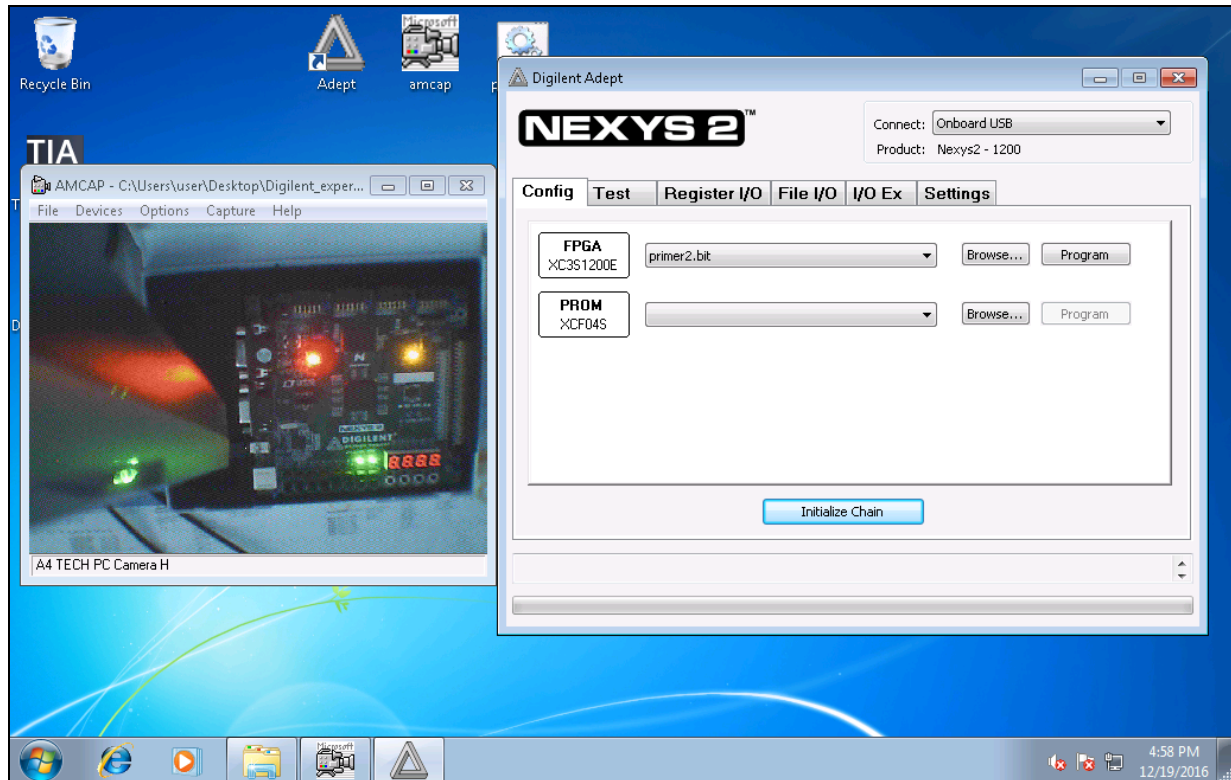


```

kolcel.ucf
1 NET "I0" LOC = "g18";
2 NET "I1" LOC = "h18";
3 NET "I2" LOC = "l14";
4 NET "I3" LOC = "l13";
5 NET "Q0" LOC = "j14";
6 NET "Q1" LOC = "j15";
7 NET "Q2" LOC = "k15";
8 NET "Q3" LOC = "k14";
9 NET "LD" LOC = "k18";
10 NET "C" LOC = "b8";
  
```

Слика 64. Фајл корисничких оїраничења *.ucf*

- d) На основу Сл. 63 која приказује симулацију начина функционисања посматране дигиталне мреже, може се закључити да ће резултујући ефекат овог експеримента бити светлење диода *LD0* и *LD1*, као што је приказано на Сл. 65.



Слика 65. Резултујући ефекат експеримента светлење диода *LD0* и *LD1*

Пример 3.

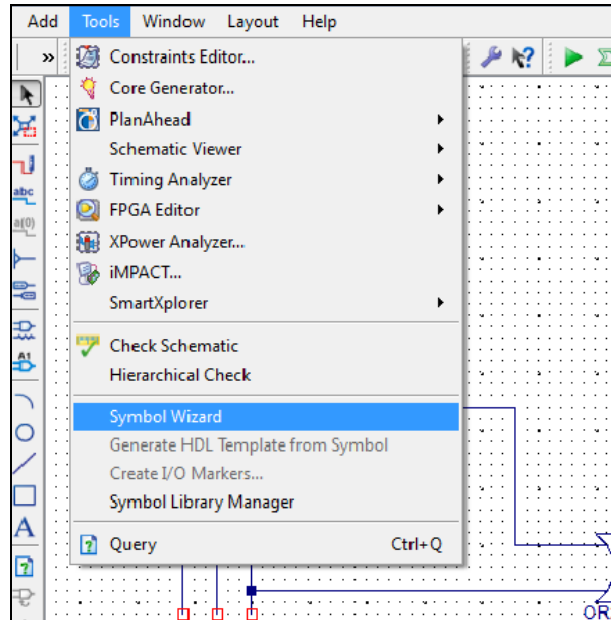
- a) Коришћењем креиране дигиталне мреже из примера 1 и хијерархијског начина дизајнирања дигиталних мрежа шематским путем, креирати комбинациону мрежу у *Xilinx ISE Design Suite* софтверу, чији је закон функционисања дат следећом функцијом преноса:

$$y = x_1 x_2 \overline{x_3} x_4 + \overline{x_2} \overline{x_4}$$

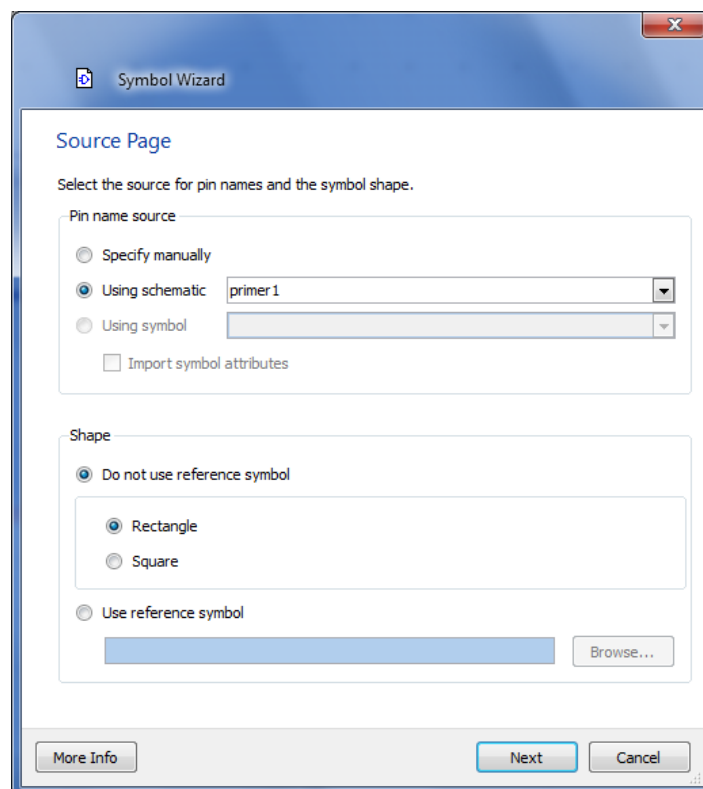
- b) Након дизајнирања извршити симулацију коришћењем *ISim* симулатора (поглавље 3);
 c) Након дизајнирања и симулације, извршити имплементацију прекидачке мреже (поглавље 4), и креирати *.bit* фајл за програмирање *FPGA* интегрисаног кола;
 d) Приступити експерименту „[Nexys 2 FPGA platforma](#)“ и пратити упутства у опису експеримента која су дата у поглављу 5.

Решење:

- а) За хијерархијско дизајнирање дигиталних мрежа, у главном менију *Xilinx ISE Design Suite* софтвера потребно је одабрати *Tools* опцију. Том приликом се појављује падајући мени у коме је потребно селектовати *Symbol Wizard* опцију, као што је приказано на Сл. 66, чиме се стартује чаробњак, за помоћ у процесу креирања новог симбола.

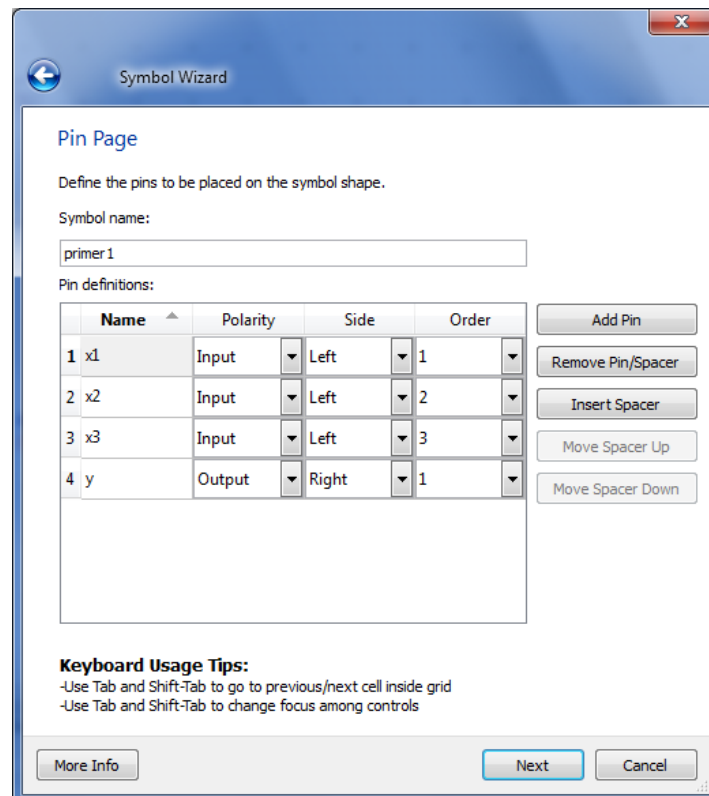


Слика 66. Избор *Symbol Wizard* опције падајућег менија *Tools*



Слика 67. Избор шеме која ће се користити за дефинисање функционалности и облика новог симбола

У *Pin name source* одељку ново отвореног прозора могуће је ручно специфицирати нови симбол који се креира или аутоматски на основу постојеће шеме, као што је приказано на Сл. 67. У овом случају потребно је одабрати другу *Using schematic* опцију и дигитално коло из примера 1. У одељку *Shape* прозора на Сл. 67, могуће је специфицирати и референтни симбол који ће се користити за дигитално коло из примера 1. У овом случају одабран је правоугаони облик представе новог симбола.



Pin Page

Define the pins to be placed on the symbol shape.

Symbol name:

Pin definitions:

	Name	Polarity	Side	Order
1	x1	Input	Left	1
2	x2	Input	Left	2
3	x3	Input	Left	3
4	y	Output	Right	1

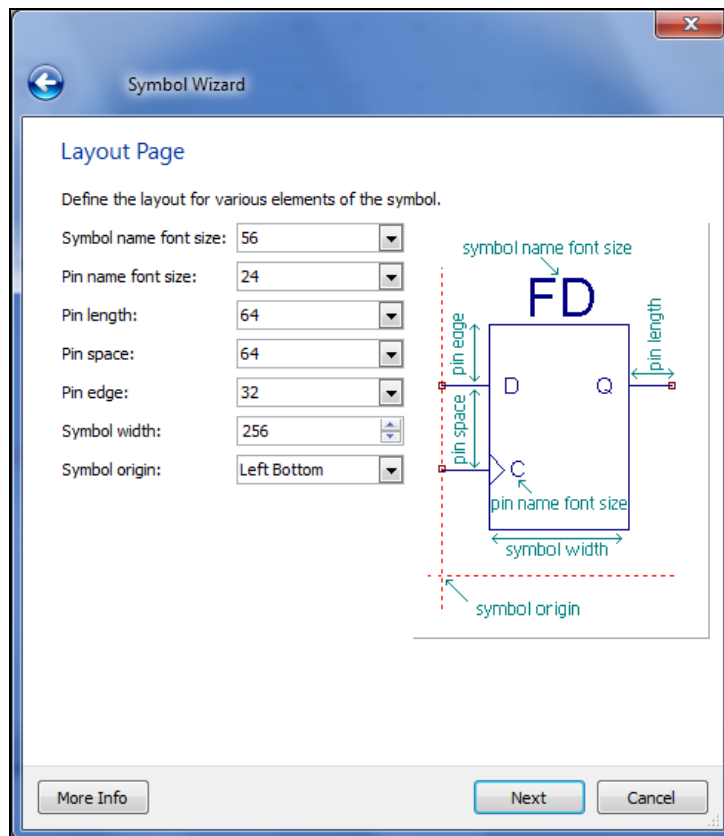
Buttons: Add Pin, Remove Pin/Spacer, Insert Spacer, Move Spacer Up, Move Spacer Down

Keyboard Usage Tips:
 -Use Tab and Shift-Tab to go to previous/next cell inside grid
 -Use Tab and Shift-Tab to change focus among controls

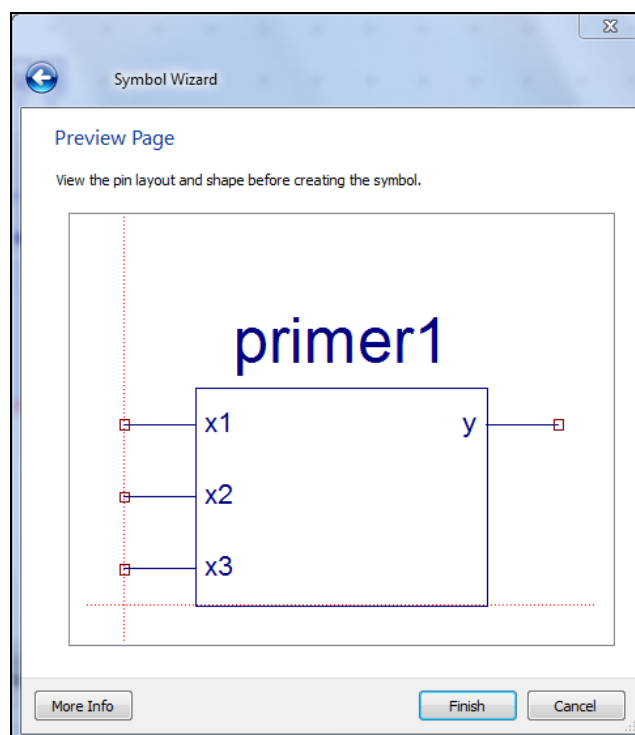
Buttons: More Info, Next, Cancel

Слика 68. Спецификација назива нової симбола, као и позиције и распореда улазно/излазних пинова

Када се притисне дугме *Next* тада се отвара прозор у коме је могуће специфицирати назив новог симбола, као и позицију и распоред улазно/излазних пинова. Подразумевано се улазни пинови налазе са леве стране, а излазни пинови са десне стране дигиталног кола (Сл. 68). Када се притисне дугме *Next* надаље се отвара прозор на Сл. 69 у коме је могуће извршити спецификацију изгледа новог симбола: његове ширине, удаљености између појединих пинова, дужине пинова, величине фонта ознака на симболу..., итд. На Сл. 70 је приказан добијени изглед новог симбола из примера 1.

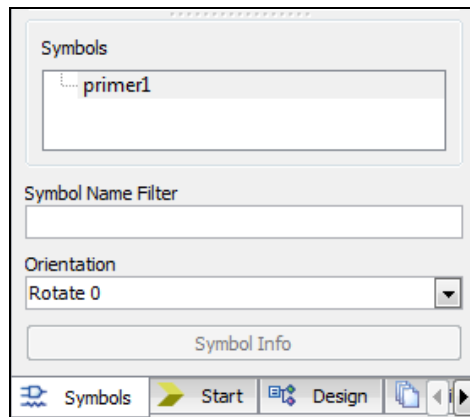


Слика 69. Спецификація вигляда нової симбола

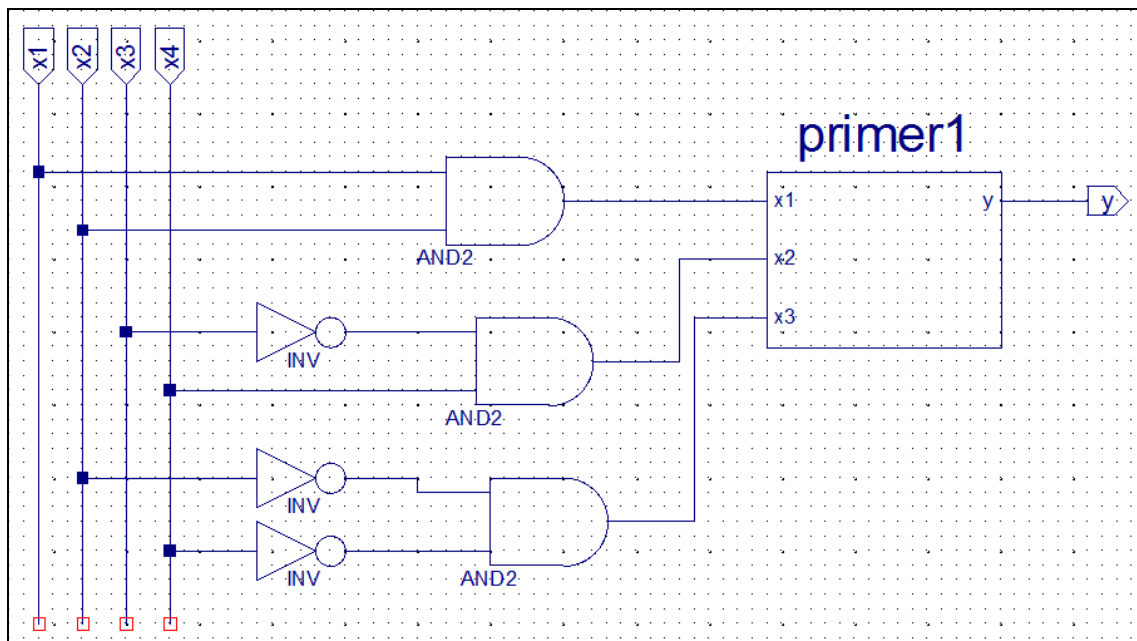


Слика 70. Добијени изглед нової симбола

Надаље је потребно сачувати измене и креирати ново дигитално коло назива *primer3*. У *Symbols* менију налази се на располагању управо креирани симбол назива *primer1*, који се може директно додати на радну површину примера 3.



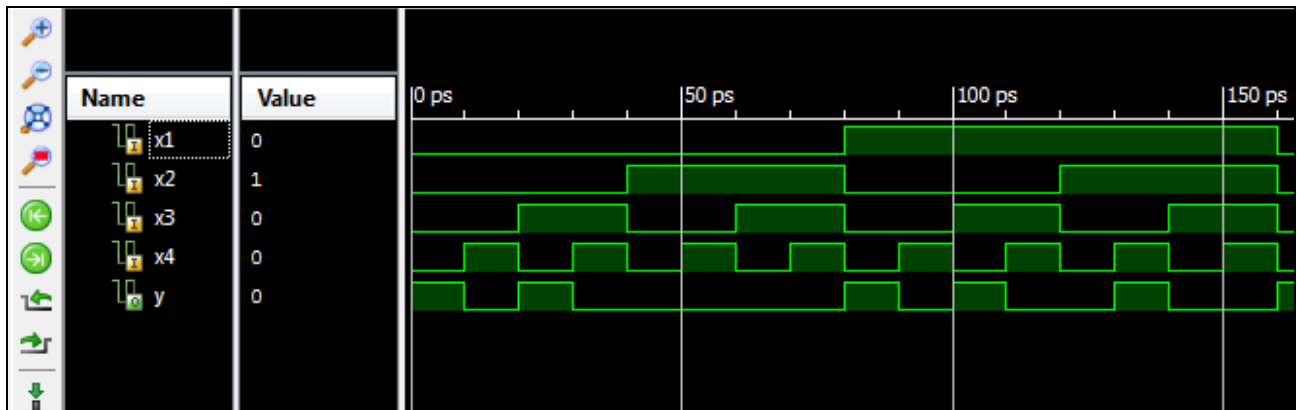
Слика 71. Ново креирани симбол се налази у *Symbols* менију



Слика 72. Коначан изглед дигиталне мреже

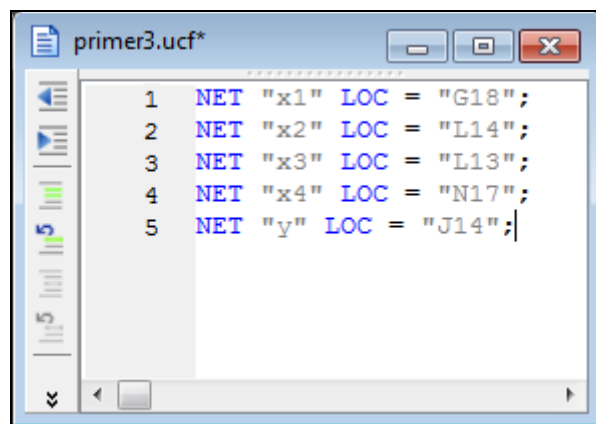
На Сл. 72 приказан је коначни изглед дигиталне мреже, која реализује задату функцију преноса.

- b) На Сл. 73 је приказан је резултат симулације задате дигиталне мреже у *ISim* симулатору (поглавље 3).



Слика 73. Симулација начина функционисања задатје дигиталне мреже у ISim симулатору

- с) Будући да је експеримент постављен тако да су четири прекидача (*SW0:3*) укључена, а преостала четири (*SW4:7*) искључена (Сл. 48), за улазни сигнал *x1* одабран је укључени прекидач *SW0*, док су за улазне сигнале *x2*, *x3* и *x4* одабрани искључени прекидачи *SW4*, *SW5* и *SW6*. За излазни сигнал *y* одабрана је диода *LD0*. На Сл. 74 је приказан .ucf фајл корисничких ограничења имајући у виду шематски приказ улазно/излазних (I/O) уређаја *Digilent Nexys 2* платформе са Сл. 48. На основу Сл. 73, која приказује симулацију начина функционисања посматране дигиталне мреже, може се закључити да ће резултујући ефекат овог експеримента бити светлење диоде *LD0*, као што је приказано на Сл. 57.



Слика 74. Фајл корисничких ограничења .ucf

Пример 4.

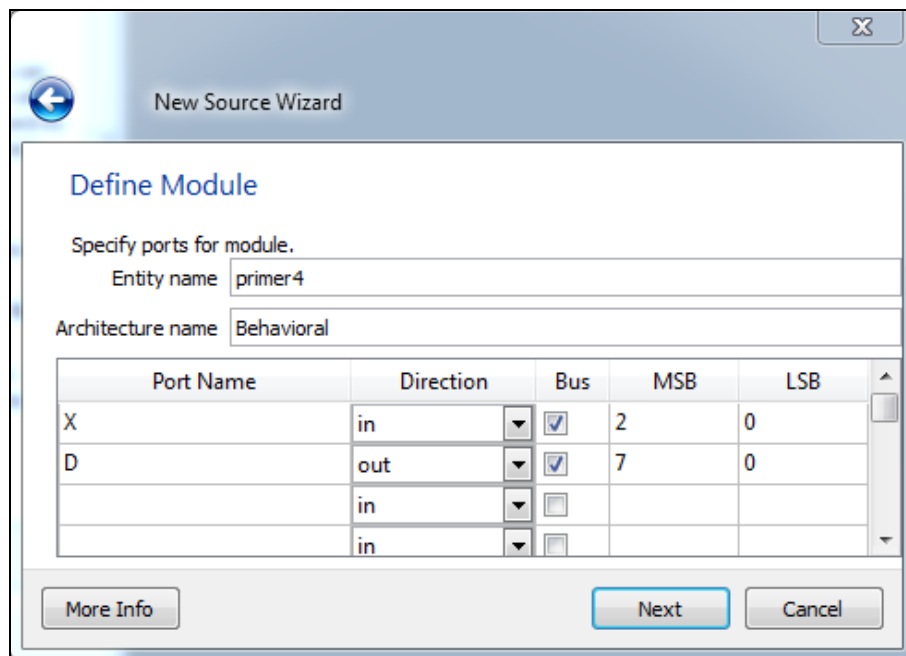
Коришћењем *Xilinx ISE Design Suite* софтвера програмирати *FPGA* интегрисано коло за креирање структурне шеме комбинационе мреже која ће имати функционалност декодера 3/8.

- а) Задату секвенцијалну мрежу дизајнирати програмирањем у *VHDL* – у (поглавље 2.2);

- b) Након дизајнирања извршити симулацију коришћењем *ISim* симулатора (поглавље 3);
- c) Након дизајнирања и симулације, извршити имплементацију прекидачке мреже (поглавље 4), и креирати *.bit* фајл за програмирање *FPGA* интегрисаног кола;
- d) Приступити експерименту „[Nexys 2 FPGA platforma](#)“ и пратити упутства у опису експеримента која су дата у поглављу 5.

Решење:

- a) С обзиром да је потребно дизајнирати прекидачку мрежу која ће се понашати као декодер са 3 улаза и 8 излаза, приликом креирања *VHDL module* фајла, према поступку описаном у поглављу 2.2, улазе декодера треба дефинисати као бинарни вектор *X*, који се доводи преко тро-битне магистрале (2:0), док излазе декодера треба дефинисати као бинарни вектор *D*, који се одводи преко осмо-битне магистрале (7:0), као што је приказано на Сл. 75. На Сл. 76 приказан је аутоматски додат део *VHDL* кода изворног фајла, који представља спецификацију улазних и излазних портова, као и део *VHDL* кода кога треба додати да би се дефинисала функционалност декодера.



Слика 75. Дефинисање назива и величине улазних/излазних портова

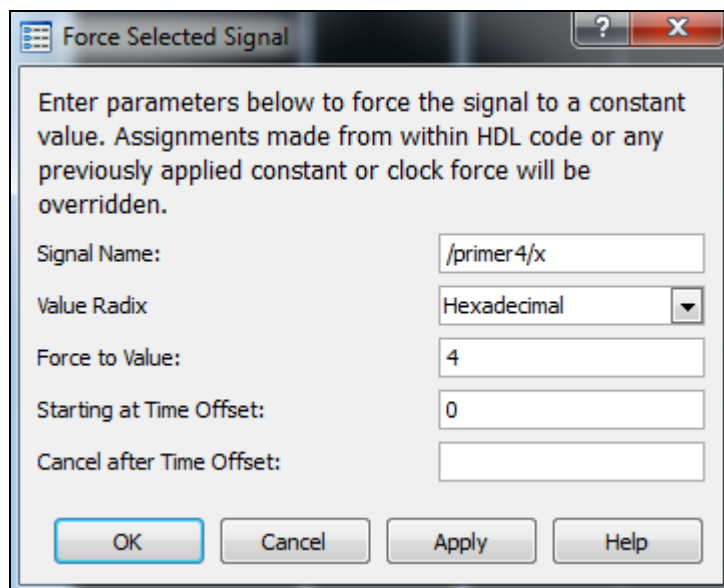
```

32 entity primer4 is
33     Port ( X : in  STD_LOGIC_VECTOR (2 downto 0);
34           D : out  STD_LOGIC_VECTOR (7 downto 0));
35 end primer4;
36
37 architecture Behavioral of primer4 is
38
39 begin
40
41 D <= "00000001" when X="000" else
42     "00000010" when X="001" else
43     "00000100" when X="010" else
44     "00001000" when X="011" else
45     "00010000" when X="100" else
46     "00100000" when X="101" else
47     "01000000" when X="110" else
48     "10000000";
49
50 end Behavioral;

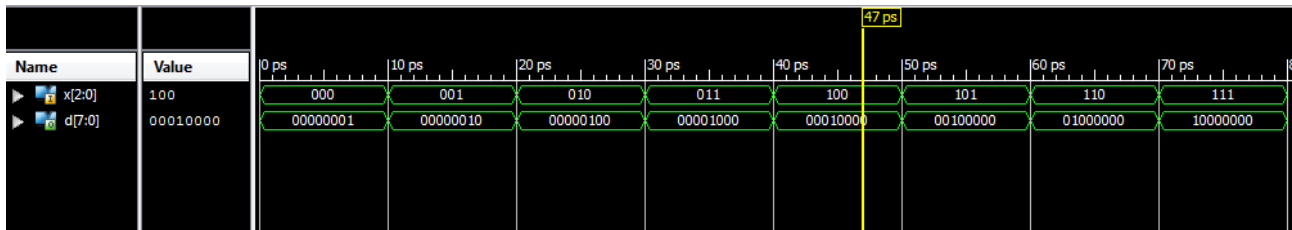
```

Слика 76. Садржај VHDL изворној фајла за дефинисање функционалности декодера

- b) За добијање резултата симулације, улазни вектор X треба форсирати као хексадецималне вредности од 0, 1, ... до 7 у једнаким временским интервалима (Сл. 77), при чему су у овом случају одабрани временски интервали од по 10 ps. На Сл. 78 приказан је резултат симулације ове дигиталне мреже, која се понаша као декодер у *ISim* симулатору.

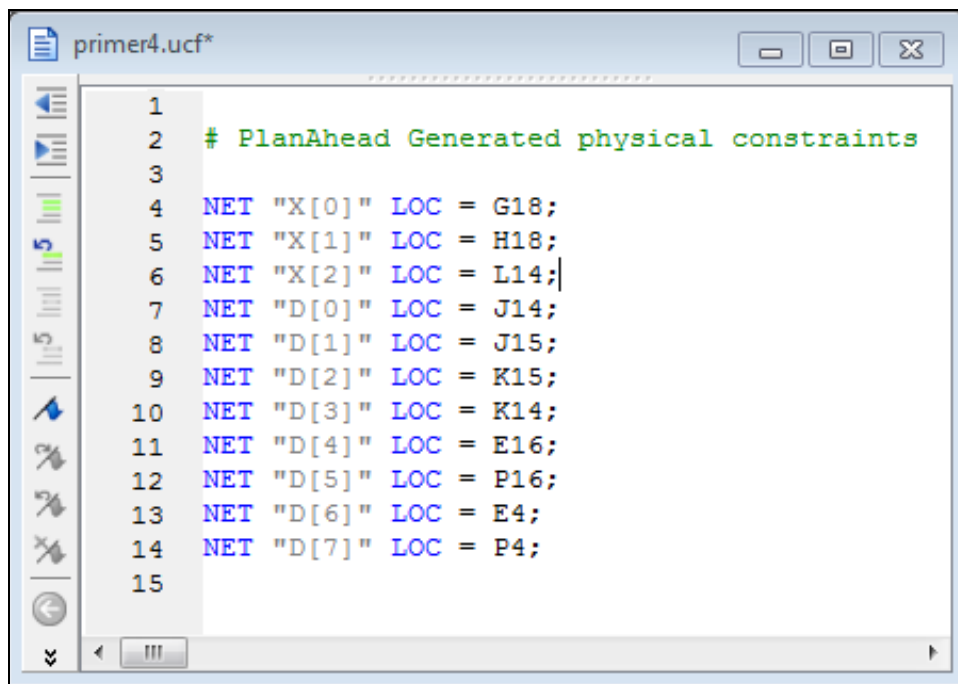


Слика 77. Форсирање хексадецималних вредности улазних сигнала



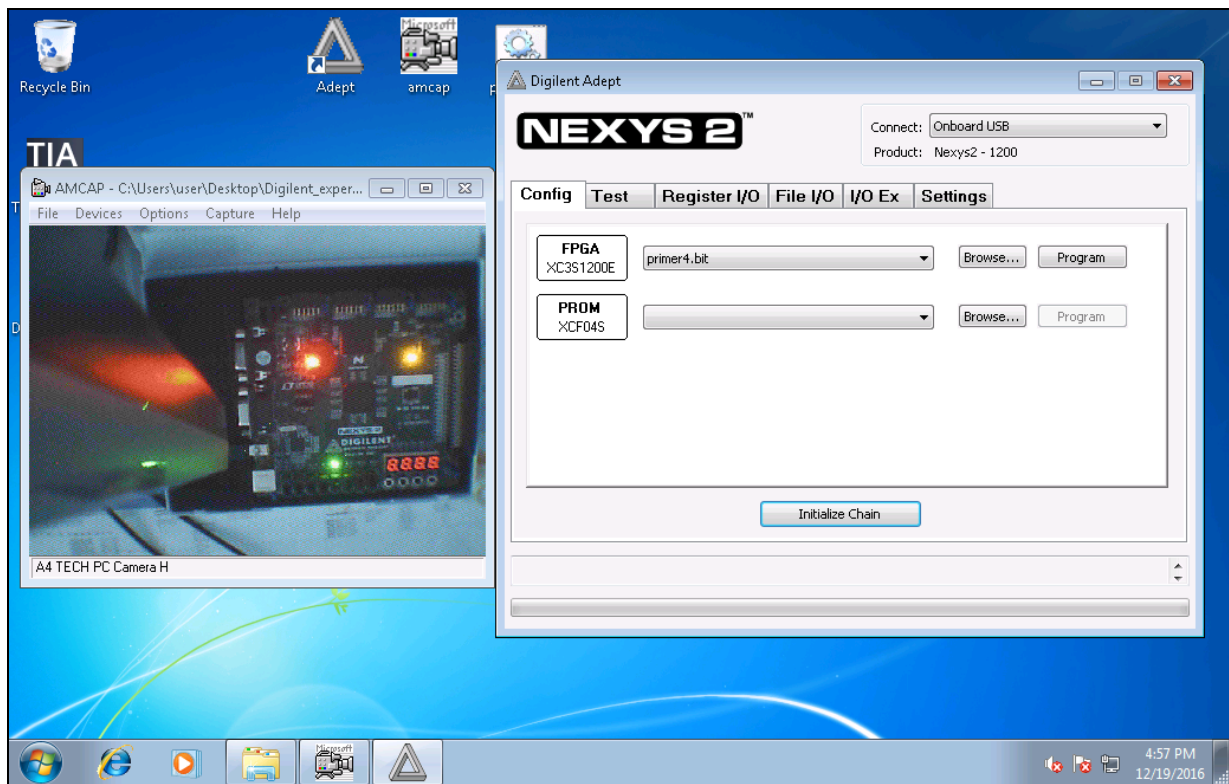
Слика 78. Резултат симулације дигиталне мреже која се понаша као декодер у ISim симулатору

- с) Приликом дефинисања локација улазно/излазних сигнала сходно Сл. 48, за улазни сигнал $X[0]$ одабран је укључен прекидач $SW0$, за улазни сигнал $X[1]$ одабран је укључен прекидач $SW1$, док је за улазни сигнал $X[2]$ одабран искључени прекидач $SW4$. За излазне сигнале $D[0]$, $D[1]$, ..., $D[7]$ су редом одабране диоде $LD0$, $LD1$, ..., $LD7$. На Сл. 79 је приказан .ucf фајл корисничких ограничења.



Слика 79. Фајл корисничких ограничења .ucf

- д) На основу Сл. 79, која приказује симулацију начина функционисања посматране дигиталне мреже, може се закључити да ће резултујући ефекат овог експеримента бити светлење диоде $LD3$, као што је приказано на Сл. 80.



Слика 80. Резултујући ефекат експеримента – свећење диоде LD3

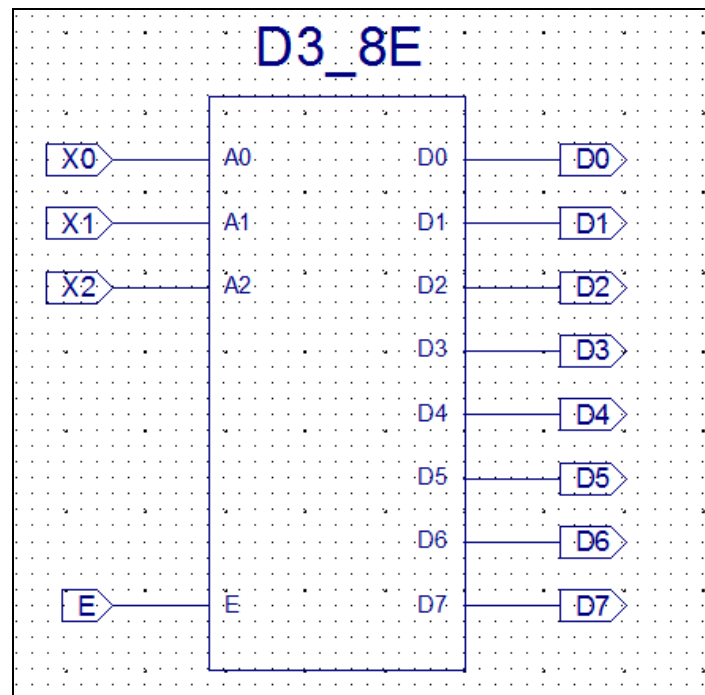
Пример 5.

Коришћењем *Xilinx ISE Design Suite* софтвера програмирати *FPGA* интегрисано коло за креирање декодера 3/8.

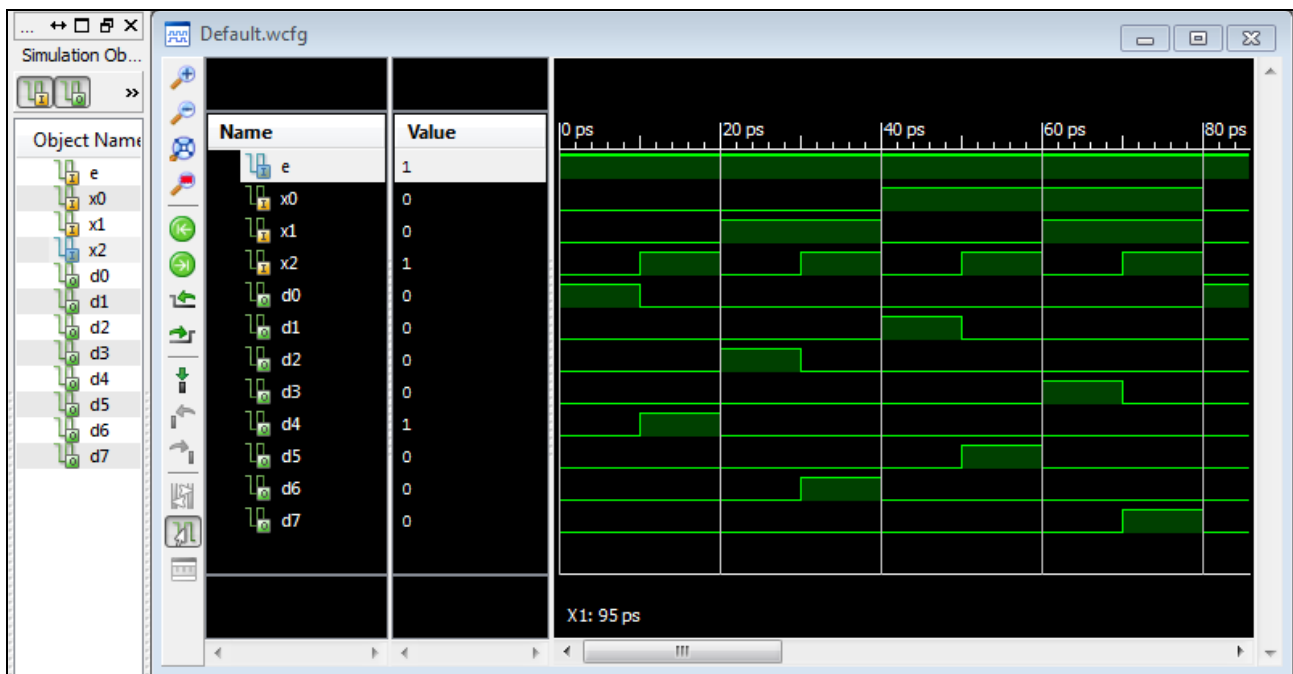
- За креирање дате комбинационе мреже користити шематски приступ (поглавље 2.1);
- Након дизајнирања извршити симулацију коришћењем *ISim* симулатора (поглавље 3);
- Након дизајнирања и симулације, извршити имплементацију прекидачке мреже (поглавље 4), и креирати *.bit* фајл за програмирање *FPGA* интегрисаног кола;
- Приступити експерименту „[Nexys 2 FPGA platforma](#)“ и пратити упутства у опису експеримента која су дата у поглављу 5.

Решење:

- На Сл. 81 приказана је шема декодера 3/8, која је директно доступна у *Symbol* менију *Xilinx ISE Design Suite* софтверу.
- На Сл. 82 приказан је резултат симулације декодера у *ISim* симулатору.

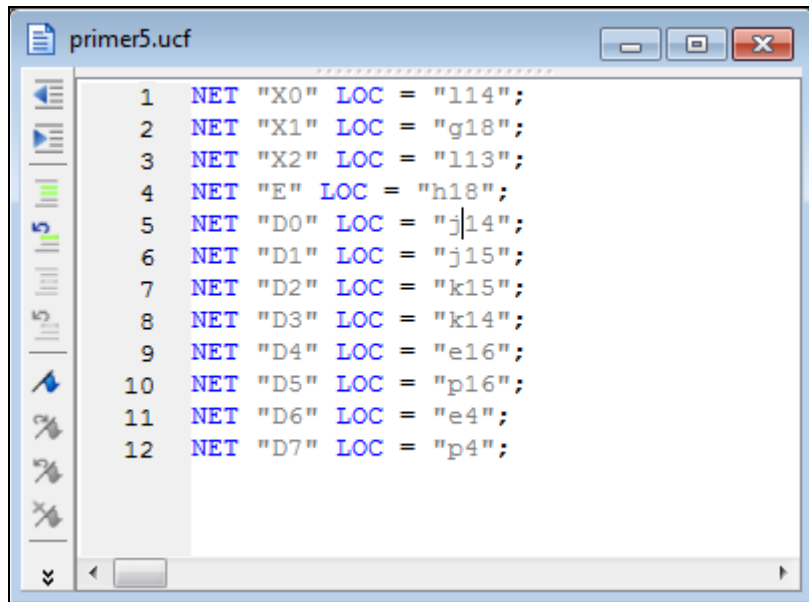


Слика 81. Шематски приказ декодера



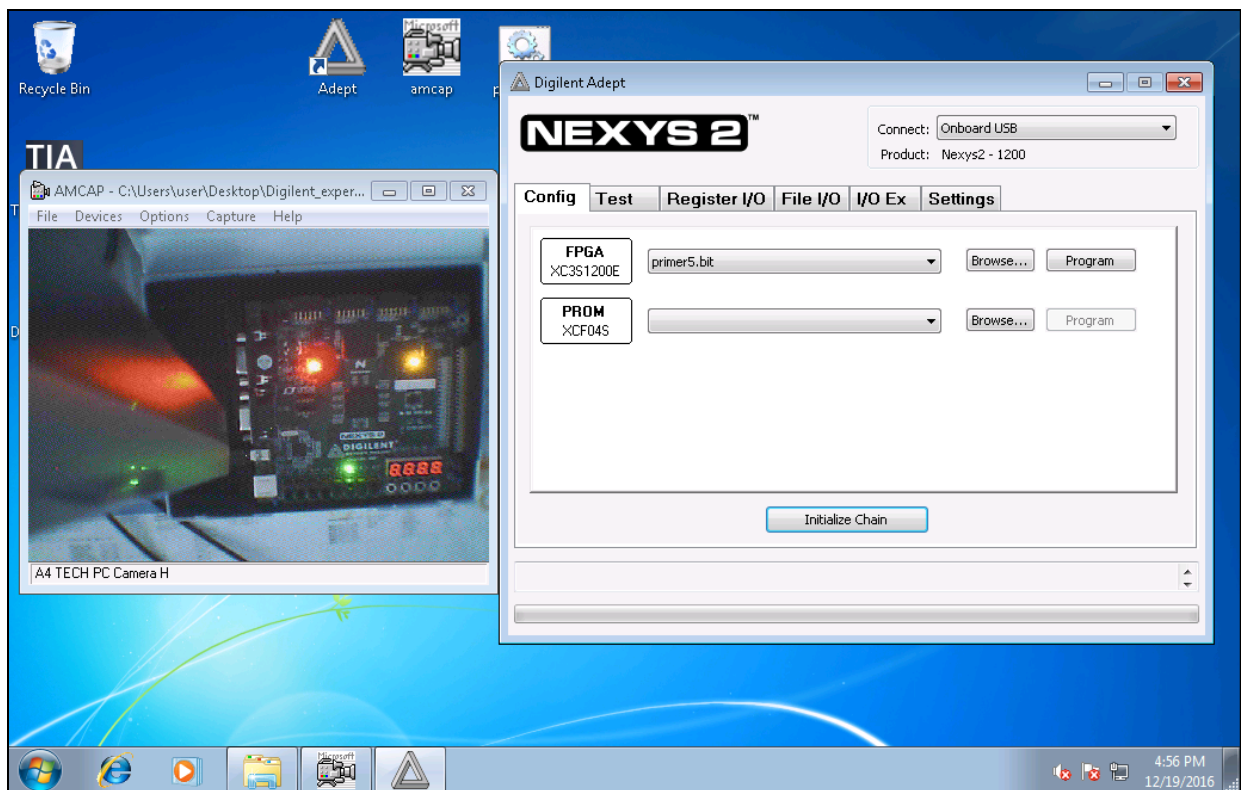
Слика 82. Резултати симулације декодера у ISim симулятору

- с) На Сл. 83 приказан садржај .lscf фајла корисничких ограничења у коме је дефинисан положај улазно/излазних пинова. Приликом дефинисања локација улазно/излазних сигнала сходно Сл. 48, за улазни сигнал $X0$ и $X2$ одабрани су искључени прекидач $SW4$ и $SW5$, за улазни сигнал $X1$ и E одабрани су укључени прекидачи $SW0$ и $SW1$. За излазне сигнале $D0, D1, \dots, D7$ су редом одабране диоде $LD0, LD1, \dots, LD7$.



Слика 83. Фајл корисничких ојраничења .ucf

- d) На основу Сл. 82, која приказује симулацију начина функционисања посматране дигиталне мреже, може се закључити да ће резултујући ефекат овог експеримента бити светлење диоде LD2, као што је приказано на Сл. 84.



Слика 84. Резултујући ефекат експеримента – светлење диоде LD2

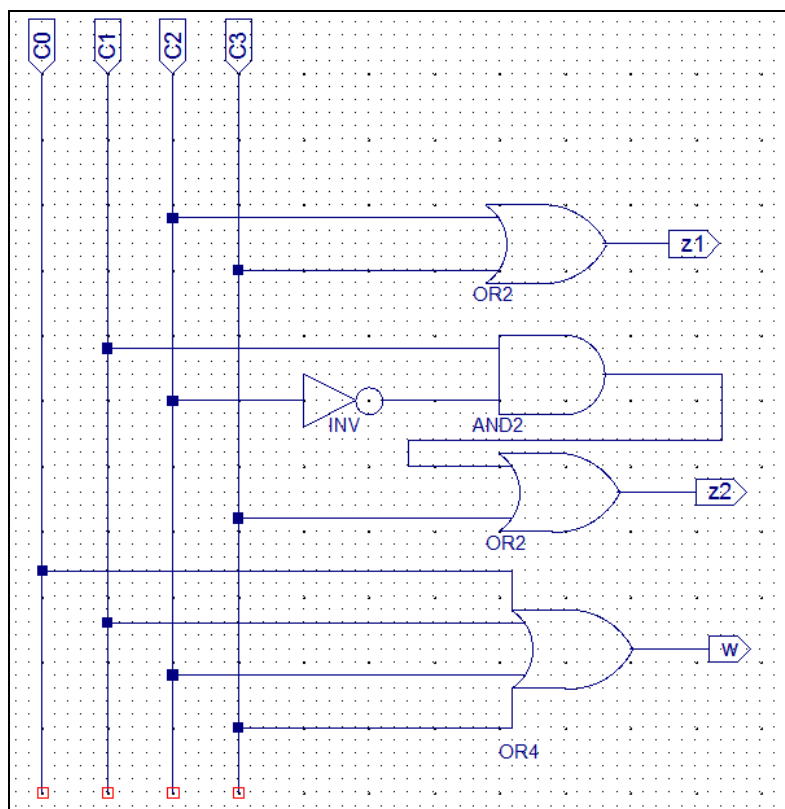
Пример 6.

Коришћењем *Xilinx ISE Design Suite* софтвера програмирати *FPGA* интегрисано коло за креирање структурне шеме комбинационе мреже кодера са приоритетом 4/2.

- За креирање дате комбинационе мреже користити шематски приступ (поглавље 2.1);
- Након дизајнирања извршити симулацију коришћењем *ISim* симулатора (поглавље 3);
- Након дизајнирања и симулације, коришћењем *Plan Ahead* софтвера извршити креирање *.ucf* фајла корисничких ограничења и креирати *.bit* фајл за програмирање *FPGA* интегрисаног кола (поглавље 4);
- Приступити експерименту „[Nexys 2 FPGA platforma](#)“ и пратити упутства у опису експеримента која су дата у поглављу 5;
- Креирати нови симбол кодера, коришћењем *Symbol Wizard-a* и креиране структурне шеме кодера са приоритетом 4/2, који ће се моћи користити при дизајнирању других прекидачких мрежа, које садрже овај тип кодера.

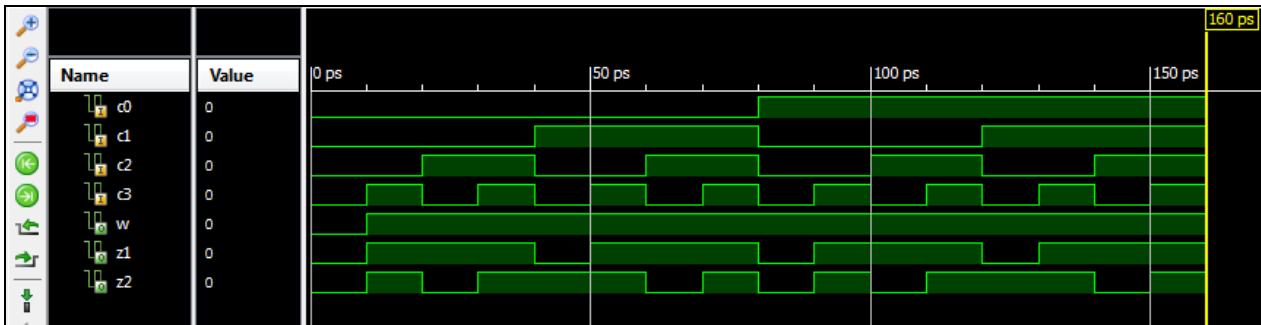
Решење:

- На Сл. 85 приказана је структурна шема кодера 4/2, која је израђена коришћењем шематског приступа креирања дигиталних мрежа у *Xilinx ISE Design Suite* софтверу.



Слика 85. Шематски приказ кодера 4/2 у *Xilinx ISE Design Suite* софтверу

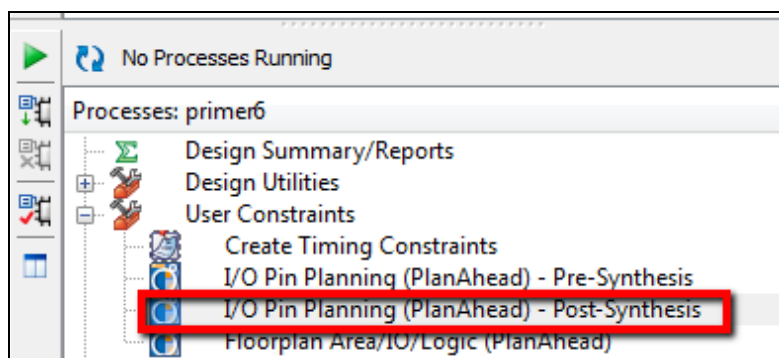
- б) На Сл. 86 приказан је резултат симулације комбинационе мреже кодера у *ISim* симулатору.



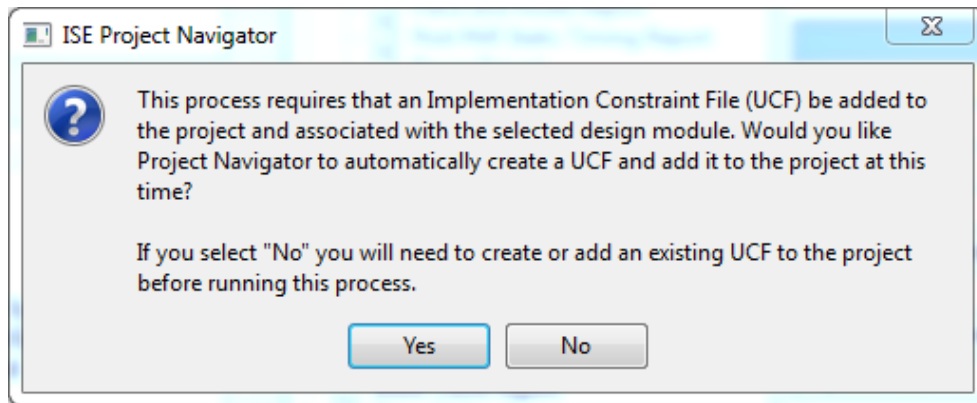
Слика 86. Резултат симулације комбинационе мреже кодера са приоритетом 4/2 у *ISim* симулатору

- с) За креирање *.ucf* фајла корисничких ограничења потребно је покренути [PlanAhead](#) софтвер [14] двоструким кликом на *I/O Pin Planning (PlanAhead) – Post-Synthesis* процес, који се налази у оквиру *User Constraints* процеса, као што је приказано на Сл. 87. Након тога се појављује упозорење да је за покретање овог процеса потребно у пројекат додати *.ucf* фајл корисничких ограничења или дозволити *Project Navigator*–у да га аутоматски креира (Сл. 88). Потребно је кликом на *Yes* одобрити аутоматско креирање *.ucf* фајла корисничких ограничења.

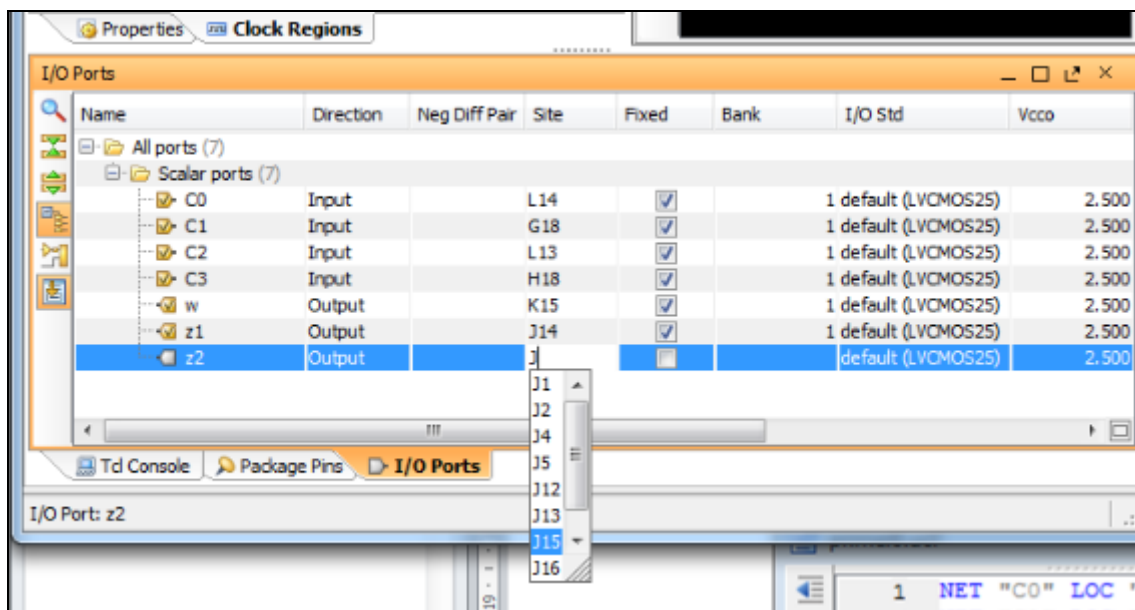
Након тога покреће се *PlanAhead* софтвер у коме је потребно у *I/O Ports* секцији дефинисати локације улазно излазних пинова у коришћењем *Site* својства, као што је приказано на Сл. 89. Локације улазно/излазних пинова се могу специфицирати и на тај начин што се улазно/излазни пинови из *I/O Ports* прозора (*C0*, *C1*, *C2*, *C3*, *W*, *z1* и *z2*) директно превуку на одговарајућу локацију у оквиру *Package* прозора, који је приказан на Сл. 90. Након тога, *PlanAhead* софтвер директно креира *.ucf* фајл корисничких ограничења, који је приказан на Сл. 91.



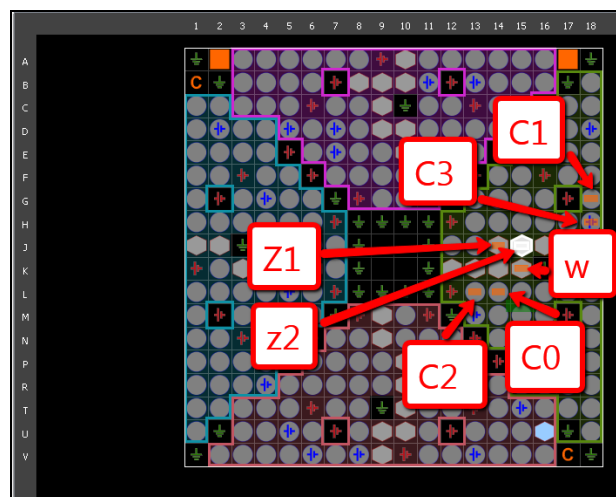
Слика 87. Покрећање *I/O Pin Planning (PlanAhead) – Post-Synthesis* процеса



Слика 88. Уџорење Project Navigator – а за креирање .ucf фајла корисничких ојраничења



Слика 89. Дефинисање локација улазно/излазних њинова

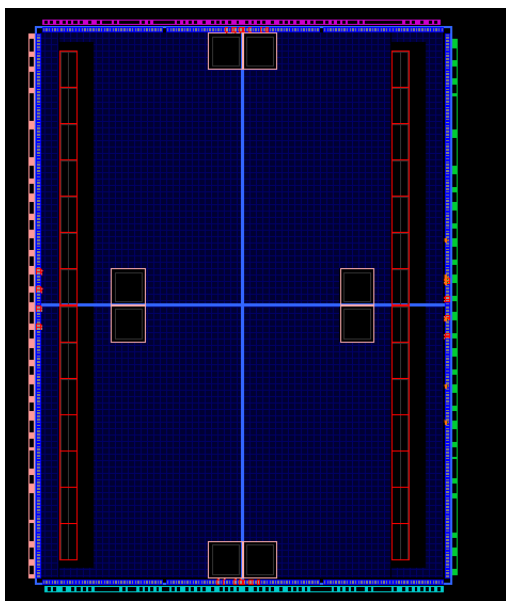


Слика 90. Package ѡрозор за директну сјецификацију локација улазно/излазних њинова

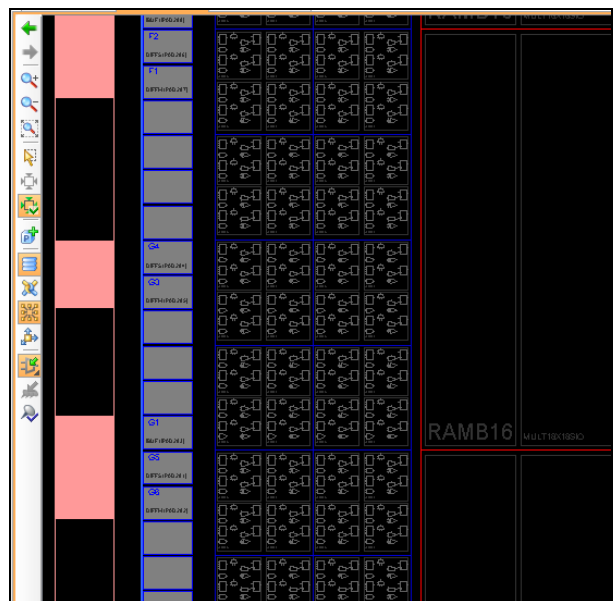
```

1
2 # PlanAhead Generated physical constraints
3
4 NET "C0" LOC = L14;
5 NET "C1" LOC = G18;
6 NET "C2" LOC = L13;
7 NET "C3" LOC = H18;
8 NET "w" LOC = K15;
9 NET "z1" LOC = J14;
10 NET "z2" LOC = J15;
11
  
```

Слика 91. Фајл корисничких ојраничења који је креиран у *PlanAhead* софтверу



a)

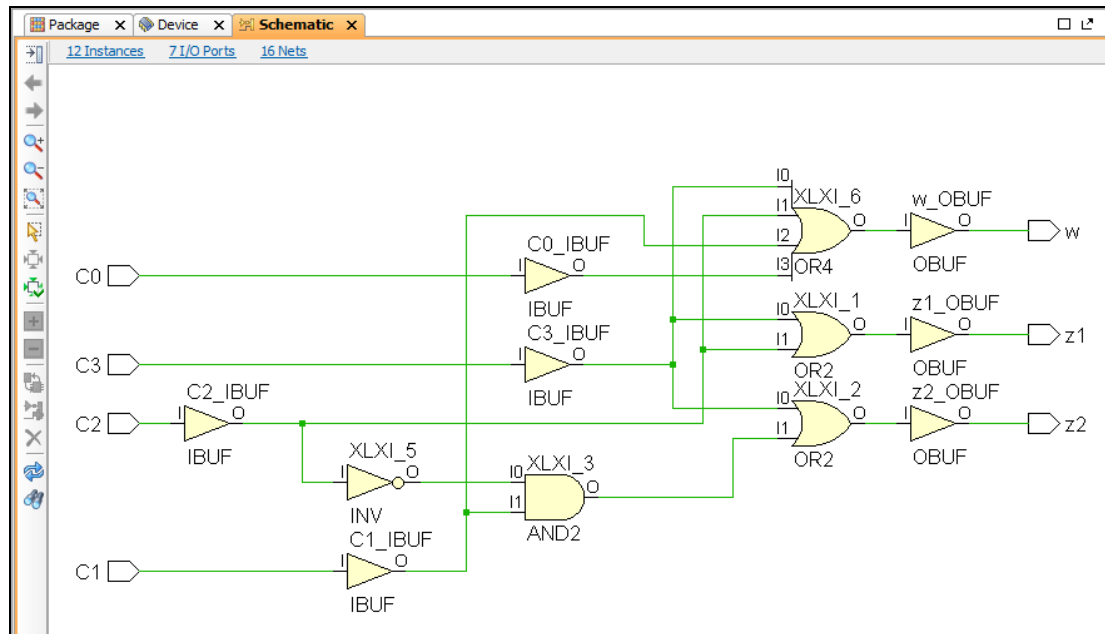


b)

Слика 92. а) *Device* прозор *PlanAhead* софтвера који приказује, основну структуру *FPGA* интегрисаної кола б) Увеличани део *FPGA* интегрисаної кола

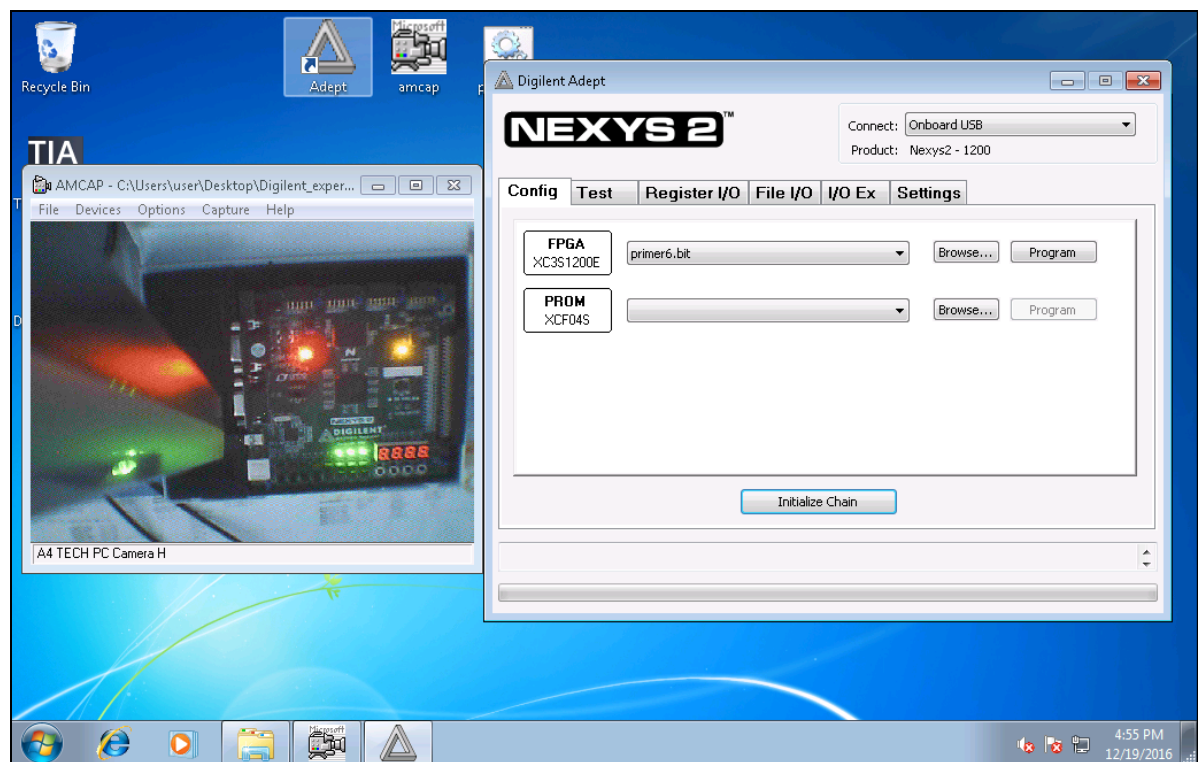
На Сл. 92а) је приказан *Device* прозор *PlanAhead* софтвера који приказује, основну структуру *FPGA* интегрисаног кола (види поглавље 1.1), чији је део увеличан и приказан на Сл. 92б). Слика 93 приказује шематски изглед креираног дигиталног кола *PlanAhead* софтвера у оквиру *Schematic* прозора *PlanAhead* софтвера.

Приликом дефинисања локација улазно/излазних сигнала сходно Сл. 48, за улазни сигнал *C0* одабран је искључен прекидач *SW4*, за улазни сигнал *C1* одабран је укључен прекидач *SW0*, за улазни сигнал *C2* одабран искључени прекидач *SW5*, док је за *C3* сигнал одабран укључен прекидач *SW1*. За излазне сигнале *z1*, *z2* и *w* су редом одабране диоде *LD0*, *LD1*, и *LD2*.



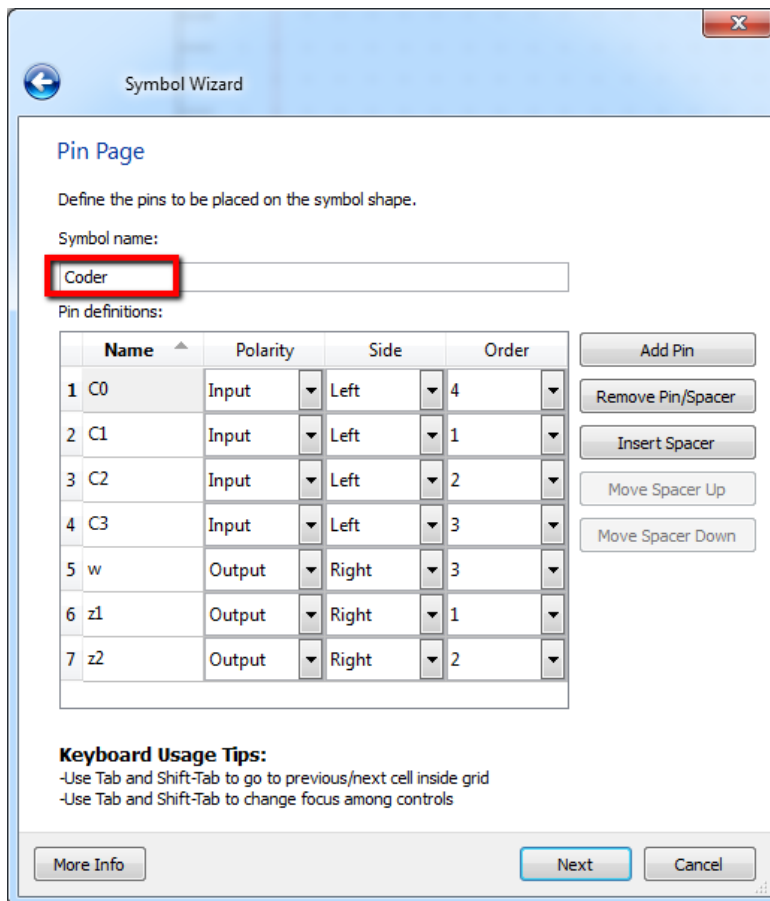
Слика 93. Шематски приказ креираної дигиталної кола у Schematic прозору PlanAhead софтвера

- d) На основу Сл. 86, која приказује симулацију начина функционисања кодера са приоритетом 4/2, може се закључити да ће резултујући ефекат овог експеримента бити светлење све три диоде LD0, LD1, и LD2, као што је приказано на Сл. 94.



Слика 94. Резултујући ефекат експеримента – светлење диода LD0, LD1 и LD2

- е) За креирање новог симбола кодера са приоритетом потребно је одабрати *Symbol Wizard* опцију падајућег менија *Tools, Xilinx ISE Design Suite* софтвера. У *Pin name source* одељку прозора који се том приликом отвара, потребно је одабрати другу *Using schematic* опцију креирања новог симбола и шему кодера креирану у овом примеру (види Сл. 67). У одељку *Shape* прозора на Сл. 67, могуће је специфицирати и референтни симбол који ће се користити за представу кодера са приоритетом 4/2. У овом случају одабран је правоугаони облик представе новог симбола. Надаље се отвара прозор у коме је специфициран назив *Coder* новог симбола, у коме су задржане подразумеване позиције улазно/излазних пинова (Сл. 95). На Сл. 96 је приказан добијени изглед новог симбола кодера са приоритетом 4/2, који ће се моћи користити при дизајнирању других прекидачких мрежа, које садрже овај тип кодера.



Symbol Wizard

Pin Page

Define the pins to be placed on the symbol shape.

Symbol name:

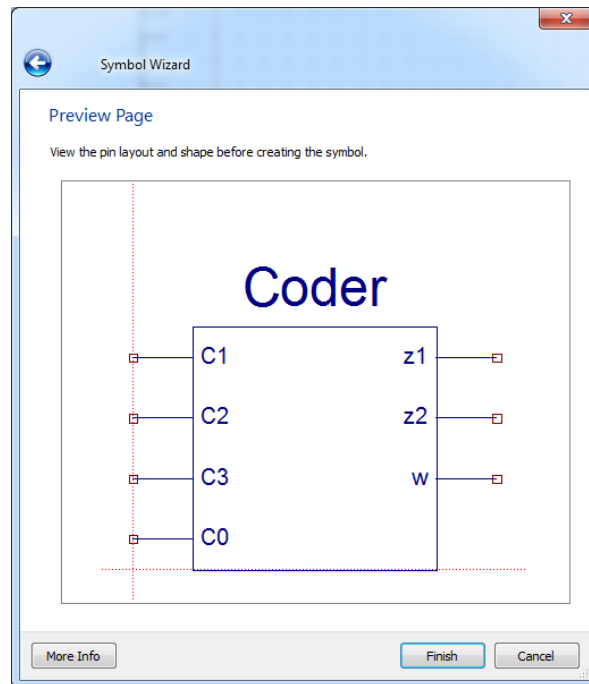
Pin definitions:

	Name	Polarity	Side	Order
1	C0	Input	Left	4
2	C1	Input	Left	1
3	C2	Input	Left	2
4	C3	Input	Left	3
5	w	Output	Right	3
6	z1	Output	Right	1
7	z2	Output	Right	2

Keyboard Usage Tips:
 -Use Tab and Shift-Tab to go to previous/next cell inside grid
 -Use Tab and Shift-Tab to change focus among controls

More Info Next Cancel

Слика 95. Дефинисање локација и распореда улазно/излазних сигнала у оквиру *Symbol Wizarda*



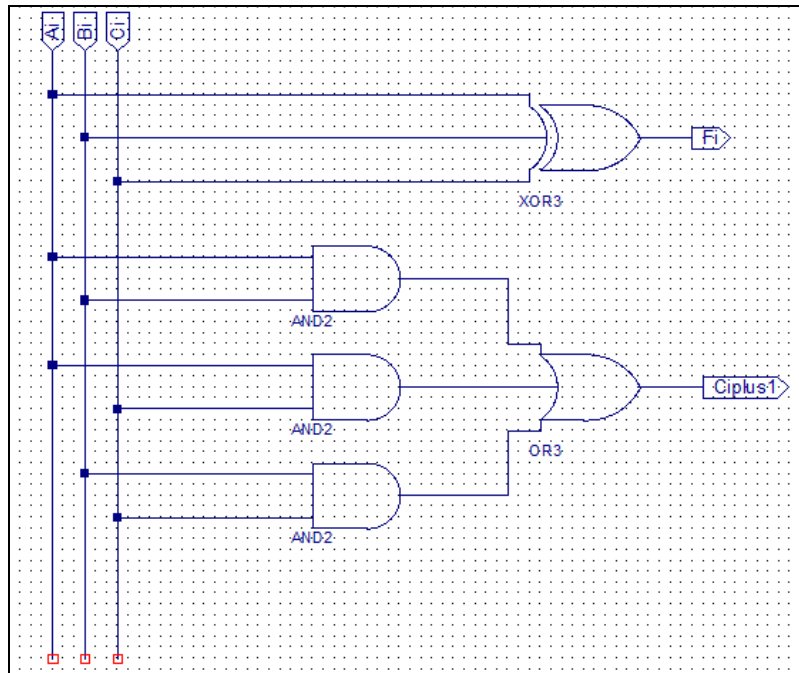
Слика 96. Изглед симбола кодера са приоритетом 4/2, креираної коришћењем *Symbol Wizarda*

Пример 7.

Коришћењем *Xilinx ISE Design Suite* софтвера програмирати *FPGA* интегрисано коло за креирање структурне шеме једног разреда потпуног сабирача.

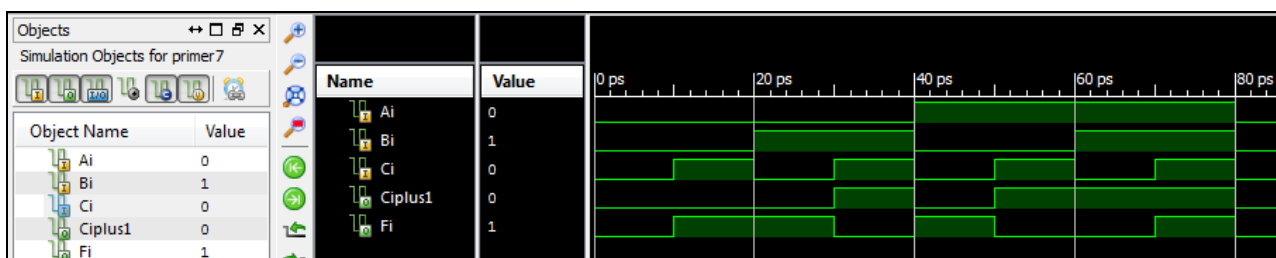
- За креирање дате комбинационе мреже користити шематски приступ (поглавље 2.1);
- Након дизајнирања извршити симулацију једног разреда потпуног сабирача коришћењем *ISim* симулатора (поглавље 3);
- Креирати нови симбол једног разреда потпуног сабирача, коришћењем *Symbol Wizarda* и креиране структурне шеме једног разредна потпуног сабирача;
- Коришћењем креираног симбола једног разреда потпуног сабирача, креирати шему сабирача са четири разреда;
- Након дизајнирања извршити симулацију начина функционисања сабирача са четири разреда коришћењем *ISim* симулатора за следеће вредности улазних података *A* и *B*: *A*=0111, *B*= 0101 за првих 10ps, а за наредних 10ps *A*= 0101, *B* = 1010 (поглавље 3);
- Након дизајнирања и симулације извршити креирање *.ucf* фајла корисничких ограничења и креирати *.bit* фајл за програмирање *FPGA* интегрисаног кола (поглавље 4);
- Приступити експерименту „[Nexys 2 FPGA platforma](#)“ и пратити упутства у опису експеримента која су дата у поглављу 5.

Решење:

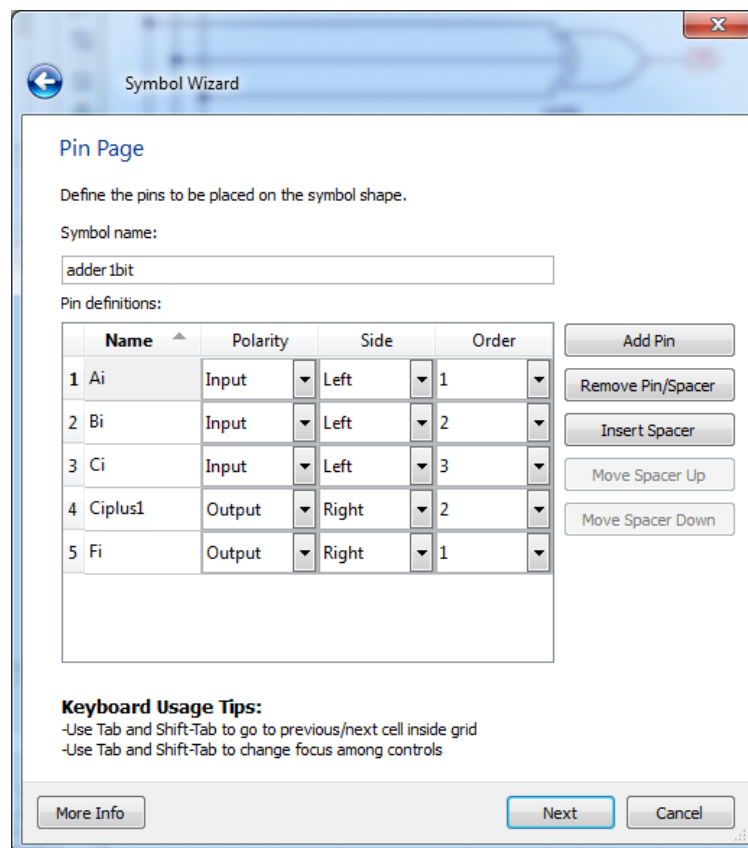


Слика 97. Шематски приказ једног разреда двојичног сабирача у *Xilinx ISE Design Suite* софтверу

- а) На Сл. 97 приказана је структурна шема једног разреда потпуног сабирача, која је израђена коришћењем шематског приступа креирања дигиталних мрежа у *Xilinx ISE Design Suite* софтверу.
- б) На Сл. 98 приказан је резултат симулације комбинационе мреже једног разреда потпуног сабирача у *ISim* симулятору



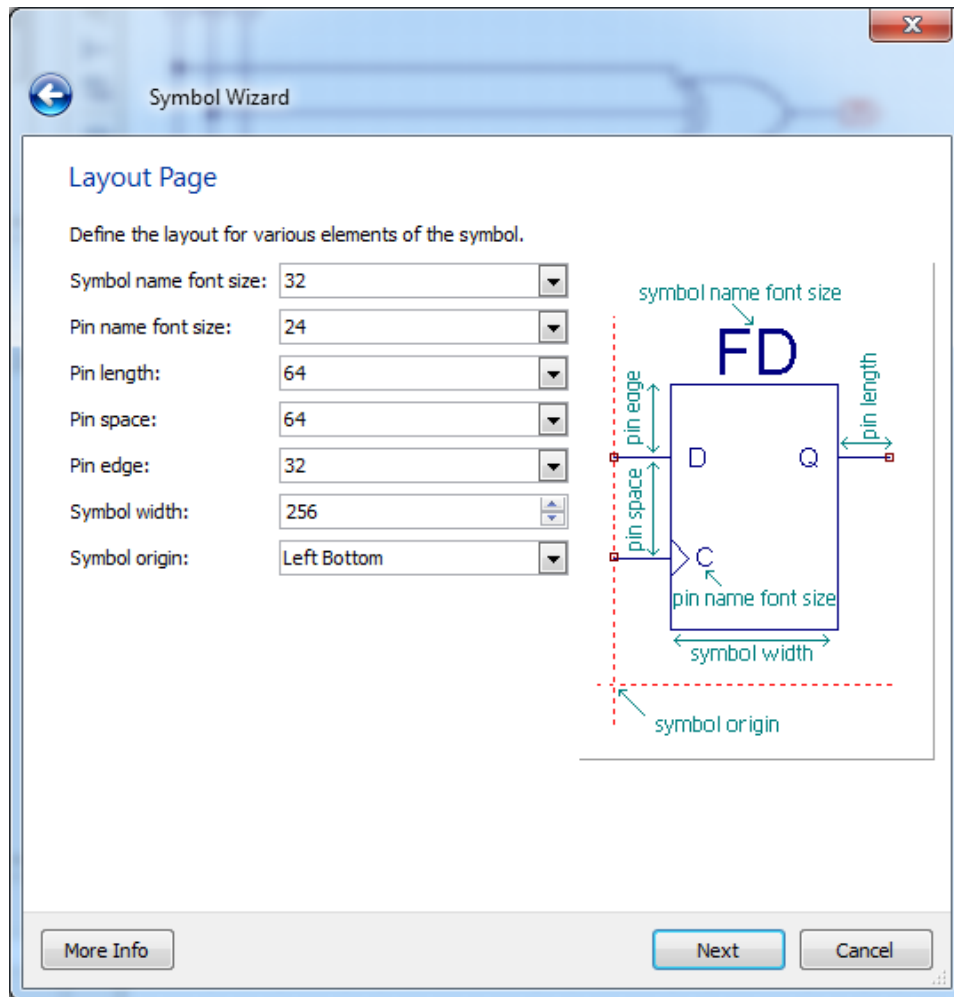
Слика 98. Резултати симулације комбинационе мреже једног разреда двојичног сабирача у *ISim* симулятору



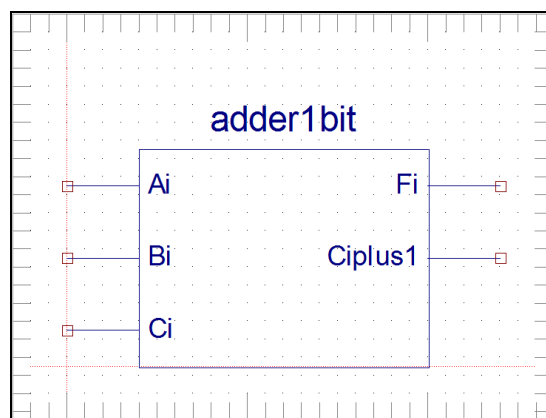
Слика 99. Спецификација изгледа нової симбола

- с) За креирање новог симбола једног разреда потпуног сабирача потребно је одабрати *Symbol Wizard* опцију падајућег менија *Tools, Xilinx ISE Design Suite* софтвера. У *Pin name source* одељку прозора који се том приликом отвара, потребно је одабрати другу *Using schematic* опцију креирања новог симбола и шему једног разреда потпуног сабирача, креирану у овом примеру (види Сл. 67). У одељку *Shape* прозора на Сл. 67, могуће је специфицирати и референтни симбол који ће се користити за представу једног разреда потпуног сабирача. У овом случају одабран је правоугаони облик представе новог симбола.

Надаље се отвара прозор у коме је специфициран назив *adder1bit* новог симбола, који треба да одговара називу шематског пројекта у коме је симбол креиран. У овом прозору су задржане подразумеване позиције улазно/излазних пинова (Сл. 99). Када се притисне дугме *Next* надаље се отвара прозор на Сл. 100 у коме је могуће извршити спецификацију изгледа новог симбола. У овом прозору задржана су сва подразумевана својства, само је величина фонта назива новог симбола подешена на 32, као што је приказано на Сл. 100. На Сл. 101 је приказан добијени изглед новог симбола потпуног сабирача за један разред.



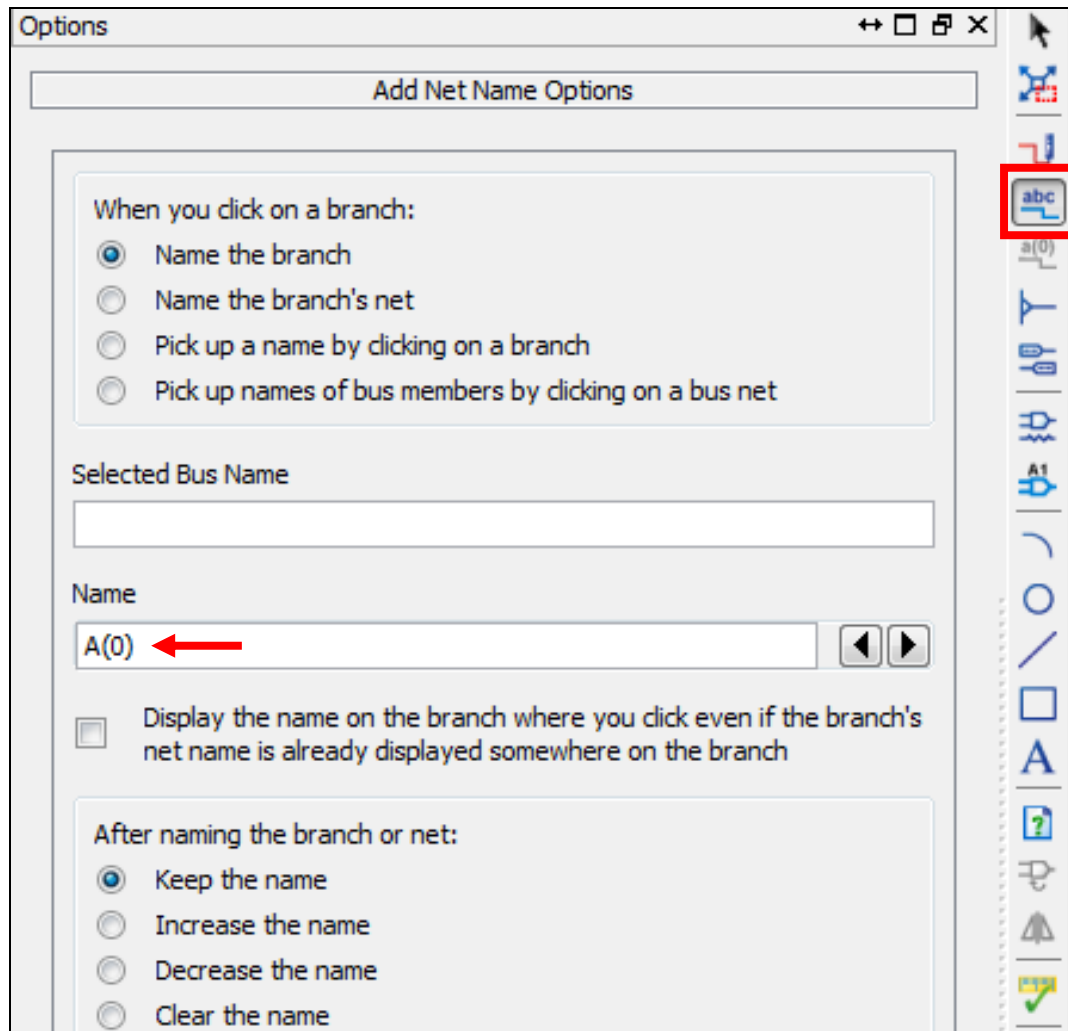
Слика 100. Спецификација изгледа нової симбола



Слика 101. Добијени изглед нової симбола пошћуної сабирача за један разред

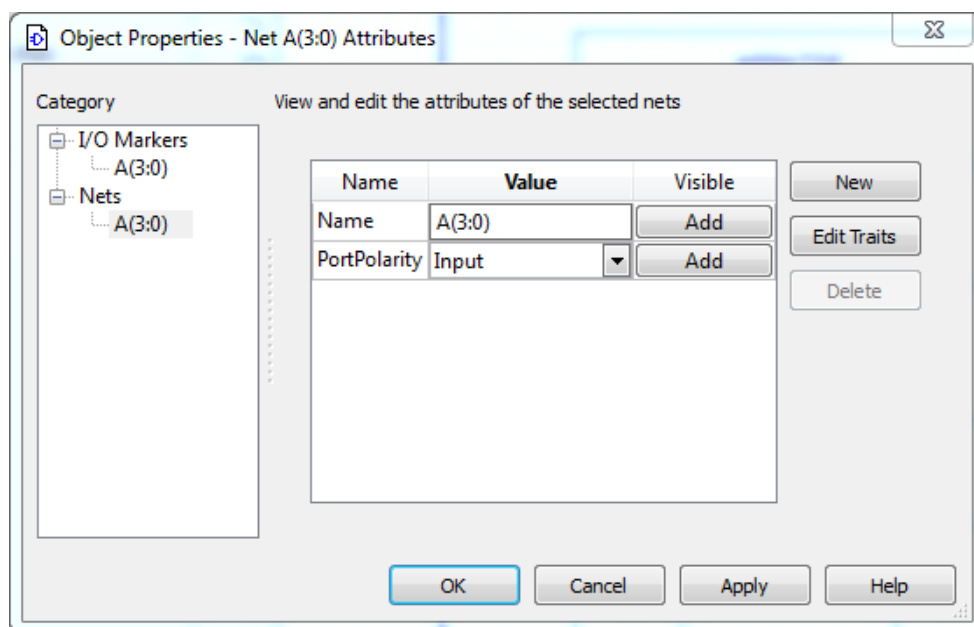
- d) За креирање шеме сабирача са четири разреда, најпре је потребно у *Symbol* менију пронаћи креирани симбол једног разреда потпуног сабирача. Претрагу је могуће извршити коришћењем *Symbol Name Filter* опције *Symbols* менија (види Сл. 21). Након тога на радну површину је потребно ставити четири таква симбола. Надаље је

коришћењем *Add Net Name Options* (Сл. 102) потребно именовати улазне пинове нултог, првог, другог и трећег разреда сабирача са: $A(0)$, $B(0)$, $C0$, $A(1)$, $B(1)$, $C1$, $A(2)$, $B(2)$, $C2$, $A(3)$, $B(3)$, $C3$, а излазне пинове са: $F(0)$, $C1$, $F(1)$, $C2$, $F(2)$, $C3$ и $F(3)$, $C4$, респективно.

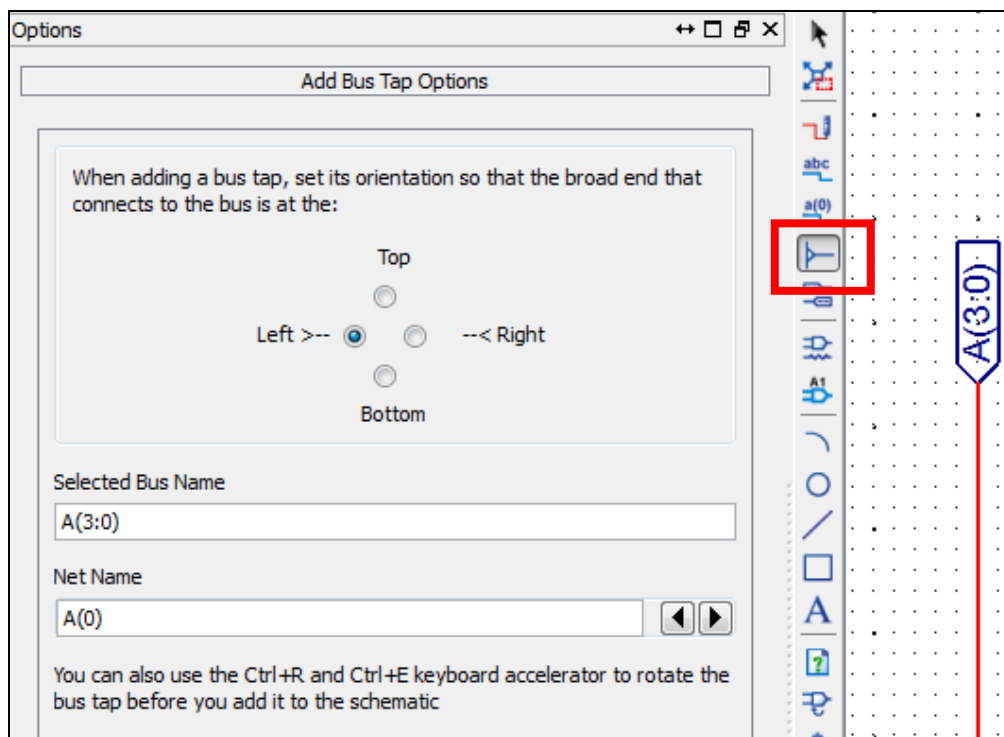


Слика 102. Именовање улазно/излазних пинова

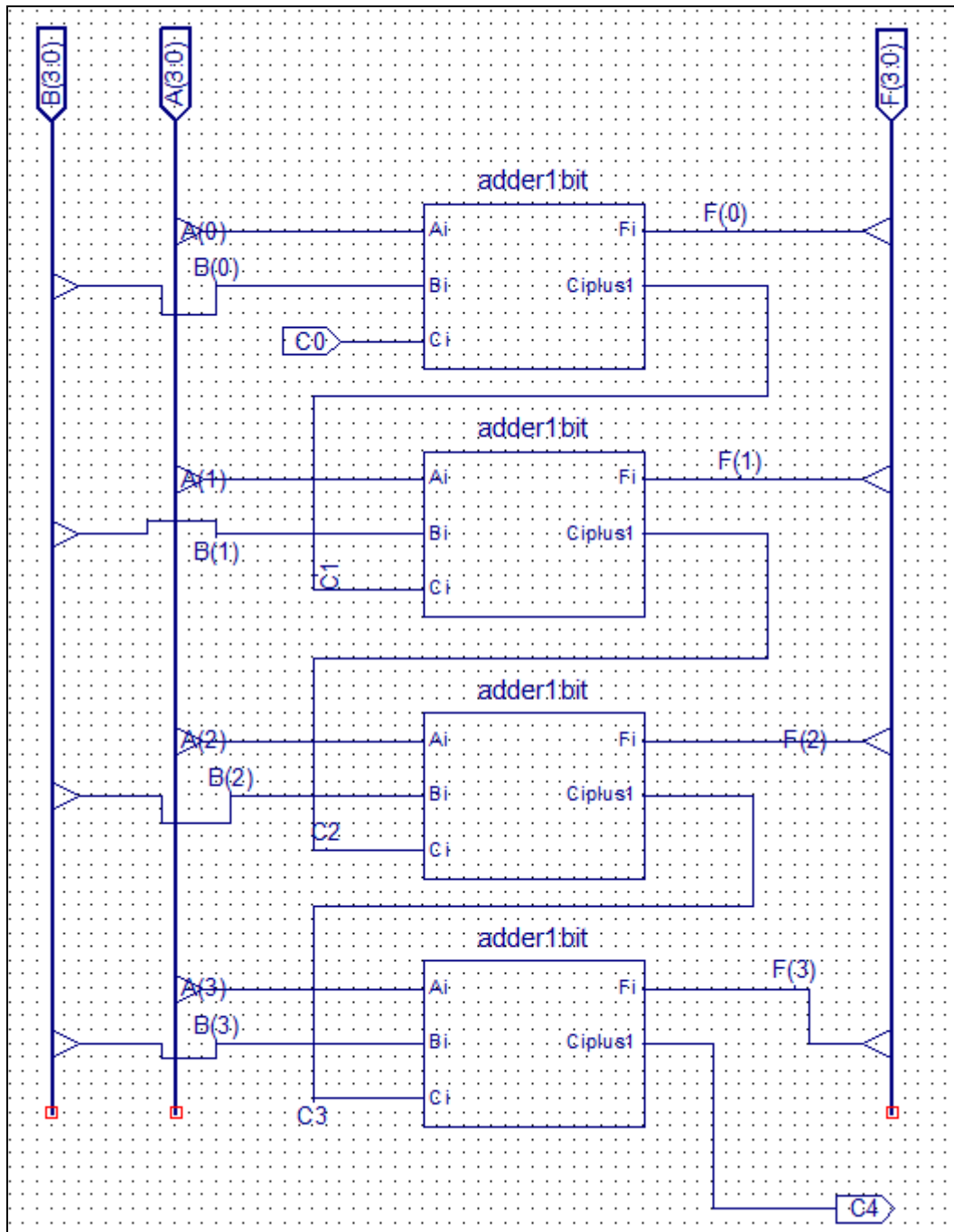
Надаље је потребно додати улазне магистрале $A(3:0)$ и $B(3:0)$ и излазну магистралу $F(3:0)$, коришћењем *Add Input/Output marker* опције (Сл. 103). Коришћењем *Add Bus Tap Options* која је приказана на Сл. 104 врши се повезивање појединих улазних/излазних пинова са магистралама. На Сл. 105 приказан је коначан изглед шеме сабирача са четири разреда.



Слика 103. Спецификација назива мјестисирале

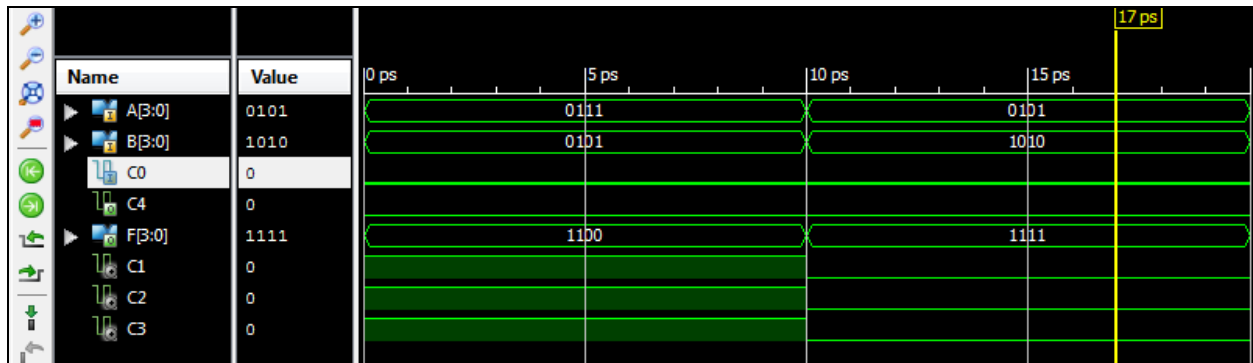


Слика 104. Add Bus Tap Options за повезивање улазно/излазних инова са мјестисиралом

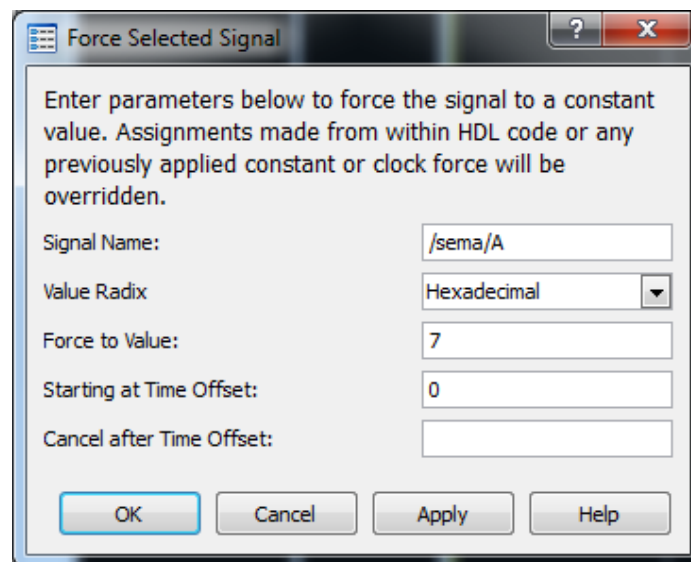


Слика 105. Коначан изглед шеме сабирача са четири разреда

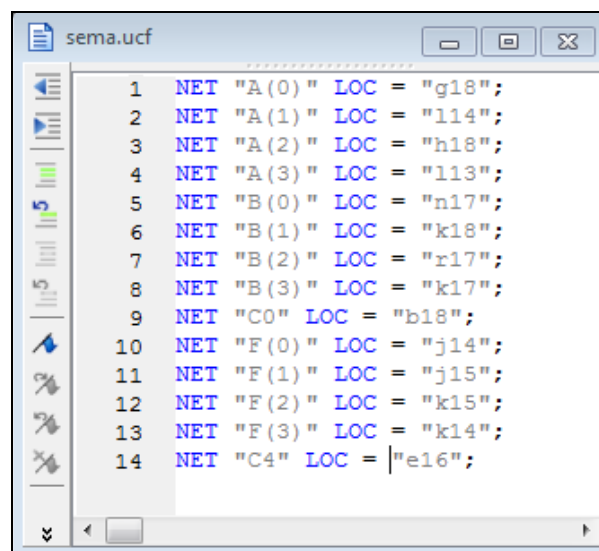
- е) На Сл. 106 приказан је резултат симулације начина функционисања сабирача са четири разреда у *ISim* симулатору за задате вредности улазних података *A* и *B*: $A=0111$, $B=0101$ за првих 10ps, а за наредних 10ps $A=0101$, $B=1010$. Вредности улазних података *A* и *B* форсиране су као константне хексадецималне вредности 7 и 5 за првих 10ps, а као константе хексадецималне вредности 5 и 10 у наредних 10ps (Сл. 107).



Слика 106. Резултат симулације комбинационе мреже сабирача са четири разреда у ISim симулатору

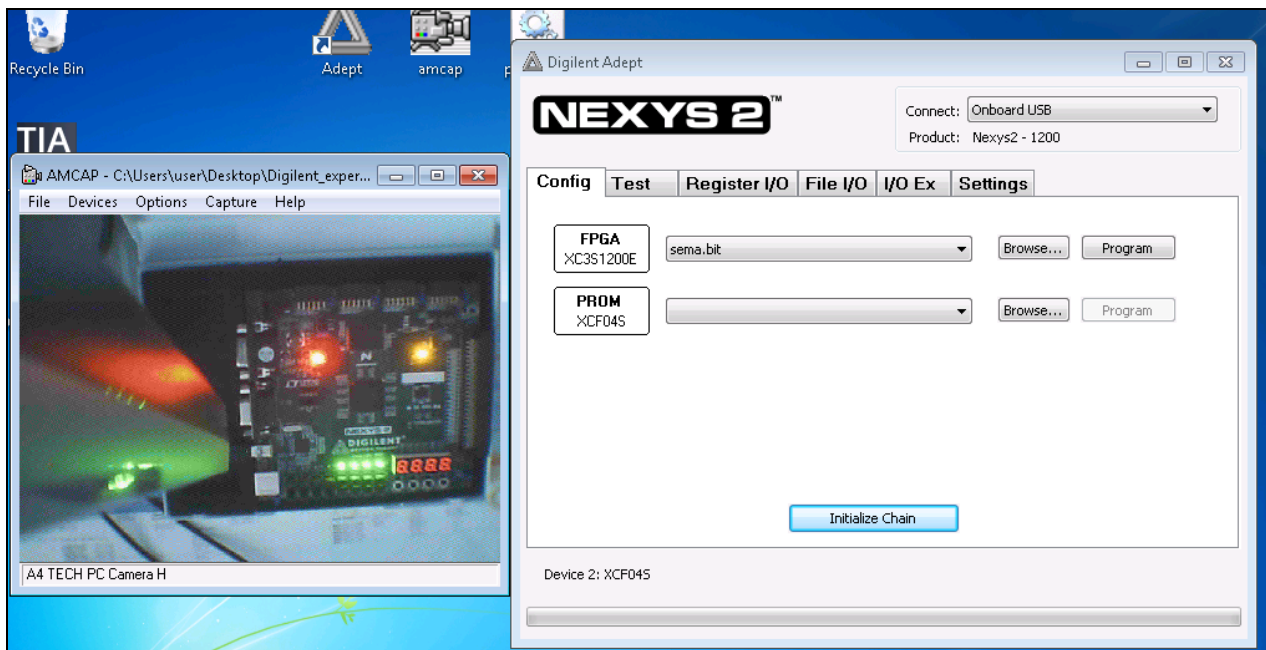


Слика 107. Форсирање константних хексадецималних вредности улазних података



Слика 108. Фајл корисничких ограничења

- f) Садржај *.ucf* фајла корисничких ограничења у коме је дефинисан положај улазно/излазних пинова приказан је на Сл. 108. Приликом дефинисања локација улазно/излазних сигнала сходно Сл. 48, за улазне сигнале $A(1)$, $A(3)$, $B(0)$ и $B(2)$ одабрани су искључени прекидачи: $SW4$, $SW5$, $SW6$ и $SW7$, док је за улазни сигнал $C0$ одабрано не притиснуто дугме $BTN0$. За улазне сигнале $A(0)$, $A(2)$, $B(1)$ и $B(3)$ одабрани су укључени прекидачи: $SW0$, $SW1$, $SW2$ и $SW3$. За излазне сигнале $F(0)$, $F(1)$, $F(2)$, $F(3)$ и $C4$ су редом одабране диоде $LD0$, $LD1$, $LD2$, $LD3$ и $LD4$.
- g) На основу Сл. 105, која приказује симулацију начина функционисања посматране дигиталне мреже, може се закључити да ће резултујући ефекат овог експеримента бити светлење диода: $LD0$, $LD1$, $LD2$ и $LD3$, као што је приказано на Сл. 109.



Слика 109. Резултујући ефекат експеримента – светлење диода $LD0$, $LD1$, $LD2$ и $LD3$

Литература

- [1] “The project TEMPUS-1-2013-1-RS-TEMPUS-JPHES ‘Building Network of Remote Labs for strengthening university-secondary vocational schools collaboration.’” [Online]. Available: <http://www.nerela.kg.ac.rs/>.
- [2] Б. Лазић, *Основи рачунарске технике - прекидачке мреже*. Академска мисао, Београд, Србија, 2006.
- [3] Ј. Протић and А. Миленковић, *Основи рачунарске технике - збирка решених испитних задатака*. Наука, Београд, Србија, 2000.
- [4] Б. Лазић, *Лоичко пројектовање рачунара*. Наука, Београд, Србија, 1994.
- [5] Д. Живковић and М. Поповић, *Имплусна и дигитална електроника*. Наука, Београд, Србија, 1992.
- [6] “Downloads.” [Online]. Available: <http://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/design-tools.html>. [Accessed: 23-Apr-2016].
- [7] “ISE In-Depth Tutorial.” [Online]. Available: https://www.xilinx.com/support/documentation/sw_manuals/xilinx14_1/ise_tutorial_ug695.pdf. [Accessed: 09-Jan-2017].
- [8] “Digilent Documentation [Reference.Digilentinc].” [Online]. Available: <https://reference.digilentinc.com/>. [Accessed: 09-Jan-2017].
- [9] “Digilent Nexys2 Board Reference Manual.” [Online]. Available: https://www.ece.umd.edu/class/enee245.F2016/nexys2_reference_manual.pdf. [Accessed: 10-Jan-2017].
- [10] “Download Microsoft .NET Framework 4.5 from Official Microsoft Download Center.” [Online]. Available: <https://www.microsoft.com/en-us/download/details.aspx?id=30653>. [Accessed: 15-Apr-2016].
- [11] “CEyeClon: Software.” [Online]. Available: <http://ceyclon.com/en/download/software/>. [Accessed: 15-Apr-2016].
- [12] “ISim User Guide.” [Online]. Available: https://www.xilinx.com/support/documentation/sw_manuals/xilinx12_2/plugin_ism.pdf. [Accessed: 10-Jan-2017].
- [13] “LiReX - Library of Remote Experiments.” [Online]. Available: <http://ad-biel-00001.ceyclon.com/WebAdmin/szenario/lirex/main.xhtml>. [Accessed: 10-Jan-2017].
- [14] “PlanAhead User Guide.” p. 433, 2013.

Листа слика

Слика 1. Основна структура FPGA интегрисаної кола.....	3
Слика 2.. Основна структура FPGA логичкої блока.....	4
Слика 3. Digilent Nexys 2 платформа.....	4
Слика 4. Блок дијаграм Digilent Nexys 2 платформе.....	5
Слика 5. а) Улазно/излазни (I/O) уређаји Digilent Nexys 2 платформе б) Осцилатор (Clock) од 50MHz и подножје за секундарни осцилатор.....	5
Слика 6. Прозор за селекцију варијанте Xilinx ISE Design софтвера.....	6
Слика 7. Избор опционих компоненти Xilinx ISE Design софтвера за инсталацију.....	7
Слика 8. Селектовање опције преузимања бесплатне лиценце Xilinx ISE Design софтвера.....	8
Слика 9. Избор ISE WebPACK лиценце.....	8
Слика 10. Приказ учешће ISE WebPACK лиценце Xilinx License Configuration Manager-а.....	9
Слика 11. Креирање нової шематскої пројекта у софтверском алату Xilinx ISE Design.....	11
Слика 12. Дефинисање типича FPGA интегрисаної кола и језика програмирања.....	12
Слика 13. Дефинисање спецификације шематскої пројекта креираної у Xilinx ISE Design Suite софтверском алату.....	12
Слика 14. Xilinx ISE Design Suite развојно окружење: 1- преглед садржаја пројекта; 2- радно окружење; 3-поруке о грешкама, упозорења, конзола; 4 - алати за покрећање процеса дизајнирања FPGA интегрисаної кола.....	13
Слика 15. Додавање нової изворної фајла пројекту.....	14
Слика 16. Дефинисање назива и шематскої типича фајла који се додаје креираном пројекту.....	15
Слика 17. Сумарни преглед карактеристика креираної пројекта.....	15
Слика 18. Изглед радног окружења након додавања изворної фајла; 1 – прозор за стварање процеса.....	16
Слика 19. Палета алата шематскої типича пројекта.....	16
Слика 20. Дигитална мрежа за реализацију.....	17
Слика 21. Symbols мени за дефинисање категорије и врсте симбола дигиталної мреже.....	17
Слика 22. Дигитално коло – прва фаза додавања логичких елемената.....	18
Слика 23. Спецификација типича и оријентације улазног/излазног сигнала.....	18
Слика 24. Измена назива улазног/излазног сигнала – један начин.....	19
Слика 25. Измена назива улазног/излазног сигнала – други начин.....	19
Слика 26. Коначан изглед дигиталне мреже.....	20
Слика 27. Дефинисање назива и VHDL Module типича фајла који се додаје креираном пројекту.....	21
Слика 28. Дефинисање назива улазних/излазних портова.....	21
Слика 29. Дефинисање спецификације VHDL Module фајла.....	22
Слика 30. Садржај VHDL изворної фајла.....	22
Слика 31. Провера синтаксе VHDL изворної фајла.....	23
Слика 32. Дефинисање назива и Verilog Module типича фајла који се додаје креираном пројекту.....	24
Слика 33. Садржај Verilog изворної фајла.....	25
Слика 34. Додавање VHDL Test Bench фајла.....	26
Слика 35. Аутоматски креиран гео садржаја VHDL Test Bench фајла.....	27
Слика 36. Кориснички подаци гео садржаја VHDL Test Bench фајла за симулацију начина функционисања дигиталне мреже.....	28



Слика 37. Симулација симулације.....	28
Слика 38. Подешавање времена симулације	29
Слика 39. Симулација начина функционисања креиране дигиталне мреже у ISim симулајору.....	29
Слика 40. Креирање Verilog Test Fixture фајла и додавање шематском пројекту	30
Слика 41. Аутоматски креиран гео кода Verilog Test Fixture фајла	31
Слика 42. Кориснички догађи гео садржаја Verilog Test Fixture фајла за симулацију начина функционисања дигиталне мреже – први начин.....	31
Слика 43. Кориснички догађи гео садржаја Verilog Test Fixture фајла за симулацију начина функционисања дигиталне мреже – други начин.....	32
Слика 44. Прозор за графички приказ вредности улазно излазних сигнала у ISim софтверу	33
Слика 45. Падајући мени за дефинисање вредности улазних сигнала.....	34
Слика 46. Подешавање вредности и времена симулације за а) улазни сигнал б) сигнал такта.....	34
Слика 47. Подешавање времена симулације и њено покретање специфицираном времену.....	35
Слика 48. Шематски приказ улазно/излазних (I/O) уређаја Digilent Nexys 2 платформе ...	36
Слика 49. Креирање фајла за дефинисање корисничких ограничења.....	37
Слика 50. Измена фајла корисничких ограничења	37
Слика 51. Садржај фајла ограничења .isf.....	38
Слика 52. Падајући мени за дефинисање својстава процеса симуловања имплементације	38
Слика 53. Дефинисање својстава процеса симуловања имплементације.....	38
Слика 54. Кораци имплементације.....	39
Слика 55. Део Pinout Report извештаја сумарног прегледа дизајниране пројекта дигиталне мреже (Design Summary/Reports)	40
Слика 56. Digilent Adept софтверски алат.....	41
Слика 57. Резултујући ефекти експеримента – светлење диода LD0	41
Слика 58. Структурна шема комбинационе мреже	44
Слика 59. Шематски приказ задатке комбинационе мреже.....	44
Слика 60. Симулација начина функционисања дигиталне мреже у ISim симулајору.....	45
Слика 61. Фајл корисничких ограничења .isf.....	45
Слика 62. Шематски приказ четворо-разредног регистра	46
Слика 63. Симулација начина функционисања четворо-разредног регистра у ISim симулајору.....	47
Слика 64. Фајл корисничких ограничења .isf.....	47
Слика 65. Резултујући ефекти експеримента светлење диода LD0 и LD1.....	48
Слика 66. Избор Symbol Wizard опције падајућег менија Tools.....	49
Слика 67. Избор шеме која ће се користити за дефинисање функционалности и облика новог симбола.....	49
Слика 68. Спецификација назива новог симбола, као и позиције и распореда улазно/излазних пинова	50
Слика 69. Спецификација изгледа новог симбола	51
Слика 70. Добијени излед новог симбола	51
Слика 71. Ново креирани симбол се налази у Symbols менију.....	52
Слика 72. Коначан излед дигиталне мреже.....	52
Слика 73. Симулација начина функционисања задатке дигиталне мреже у ISim симулајору	53
Слика 74. Фајл корисничких ограничења .isf.....	53



Слика 75. Дефинисање назива и величине улазних/излазних портова.....	54
Слика 76. Садржај VHDL изворној фајла за дефинисање функционалности декодера	55
Слика 77. Форсирање хексадецималних вредности улазних сигнала.....	55
Слика 78. Резултат симулације дигиталне мреже која се понаша као декодер у ISim симулятору	56
Слика 79. Фајл корисничких ограничења .icf.....	56
Слика 80. Резултујући ефекат експеримента – свећење диода LD3	57
Слика 81. Шемајски приказ декодера	58
Слика 82. Резултат симулације декодера у ISim симулятору.....	58
Слика 83. Фајл корисничких ограничења .icf.....	59
Слика 84. Резултујући ефекат експеримента – свећење диода LD2	59
Слика 85. Шемајски приказ кодера 4/2 у Xilinx ISE Design Suite софтверу	60
Слика 86. Резултат симулације комбинационе мреже кодера са приоритетом 4/2 у ISim симулятору	61
Слика 87. Покрећање I/O Pin Planning (PlanAhead) – Post-Synthesis процеса.....	61
Слика 88. Увозорење Project Navigator – а за креирање .icf фајла корисничких ограничења	62
Слика 89. Дефинисање локација улазно/излазних пинова	62
Слика 90. Package прозор за директну спецификацију локација улазно/излазних пинова ...	62
Слика 91. Фајл корисничких ограничења који је креиран у PlanAhead софтверу.....	63
Слика 92. а) Device прозор PlanAhead софтвера који приказује, основну структуру FPGA интегрисаної кола б) Увеличани део FPGA интегрисаної кола	63
Слика 93. Шемајски приказ креираної дигиталної кола у Schematic прозору PlanAhead софтвера.....	64
Слика 94. Резултујући ефекат експеримента – свећење диода LD0, LD1 и LD2.....	64
Слика 95. Дефинисање локација и распореда улазно/излазних сигнала у оквиру Symbol Wizarda	65
Слика 96. Изглед симбола кодера са приоритетом 4/2, креираної коришћењем Symbol Wizarda	66
Слика 97. . Шемајски приказ једног разреда двојичног сабирача у Xilinx ISE Design Suite софтверу.....	67
Слика 98. Резултат симулације комбинационе мреже једног разреда двојичног сабирача у ISim симулятору.....	67
Слика 99. Спецификација изгледа нової симбола	68
Слика 100. Спецификација изгледа нової симбола	69
Слика 101. Добијени изглед нової симбола двојичног сабирача за један разред	69
Слика 102. Именовање улазно/излазних пинова.....	70
Слика 103. Спецификација назива мајстисрале	71
Слика 104. Add Bus Tap Options за повезивање улазно/излазних пинова са мајстисралом ...	71
Слика 105. Коначан изглед шеме сабирача са четири разреда.....	72
Слика 106. Резултат симулације комбинационе мреже сабирача са четири разреда у ISim симулятору	73
Слика 107. Форсирање константних хексадецималних вредности улазних података.....	73
Слика 108. Фајл корисничких ограничења.....	73
Слика 109. Резултујући ефекат експеримента – свећење диода LD0, LD1, LD2 и LD3 ..	74

CIP - Каталогизација у публикацији - Народна библиотека Србије, Београд

004.383(075.8)(076)

004.42:004.312(075.8)(076)

ПРИРУЧНИК за пројектовање дигиталних мрежа коришћењем *FPGA*
интегрисаног кола са примерима / Вања Луковић ... [и др.]. - Чачак :
Универзитет у Крагујевцу, Факултет техничких наука, 2017 (Чачак : Бајић). -
78 стр. : илустр. ; 30 cm

Тираж 100. - Библиографија: стр. 5.

ISBN 978-86-7776-212-4

1. Луковић, Вања, 1976- [аутор]

а) Локално програмирање логичко поље - Вежбе б) Логичка кола -

Програмирање - Вежбе

COBISS.SR-ID 232547340