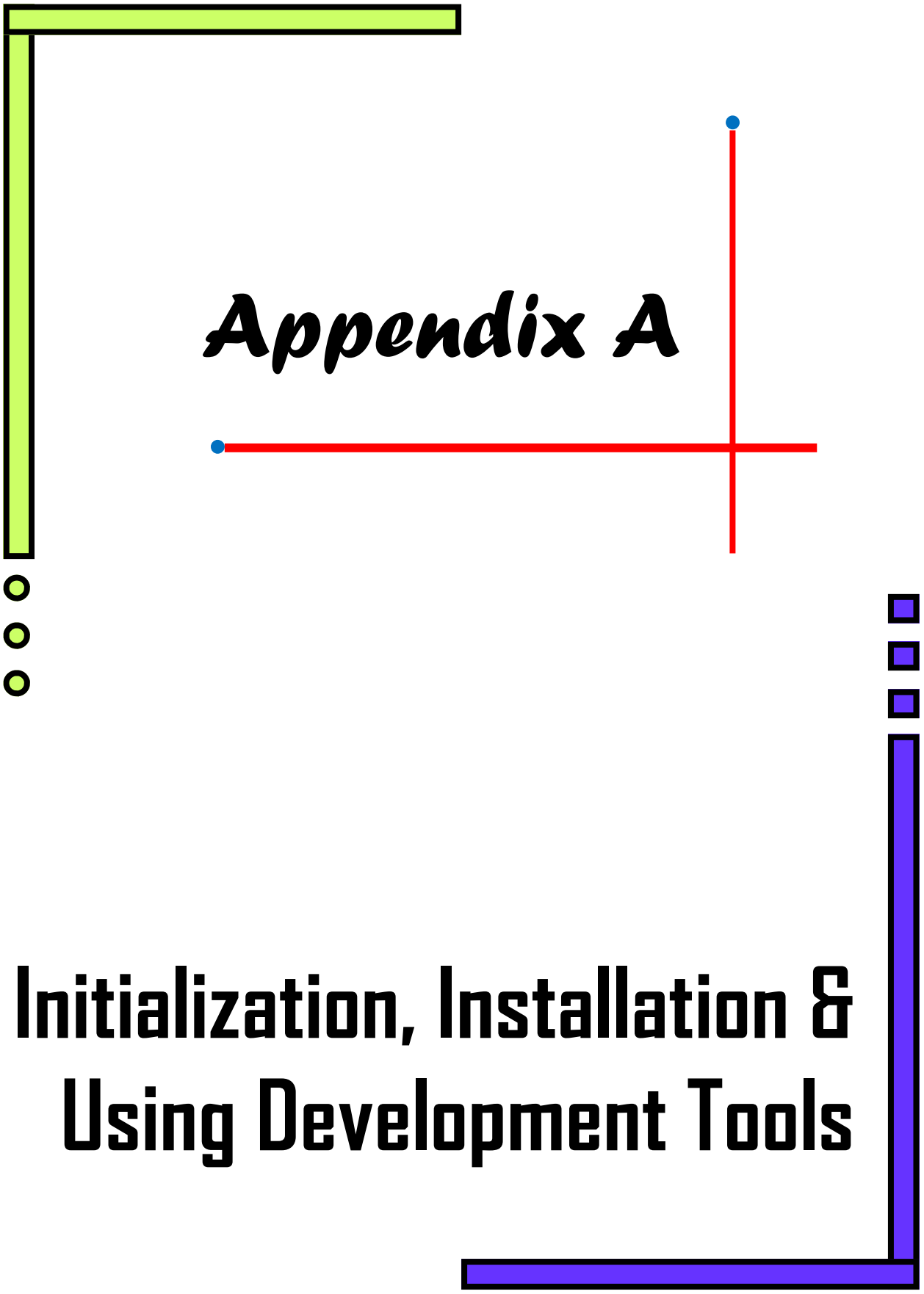


Appendix A

Initialization, Installation & Using Development Tools

APPENDIX A

A.1 Process of installation of NS2 on Fedora Linux 8

Following steps should be executed in sequence to install NS2 successfully:

1) Go to link: SourceForge.net:files
2) Download File: ns-allinone-2.33.tar.gz
3) Start Fedora and open Root Account
4) Place the downloaded file on Desktop and right click on it and select extract
5) Open console and write following commands:
   a) cd desktop
   b) cd ns-allinone-2.32
   c) ./install or ./INSTALL

- Now go to /etc folder and type

gedit ~/.bashrc

Add the following lines to the end of it. Remember replace "/your/path" by something like "/home/ddv". And accordingly also change the version numbers. This is for ns 2.29.3

```
# LD_LIBRARY_PATH
OTCL_LIB=/your/path/ns-allinone-2.31/otcl-1.13
NS2_LIB=/your/path/ns-allinone-2.31/lib
X11_LIB=/usr/X11R6/lib
USR_LOCAL_LIB=/usr/local/lib
export
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$OTCL_LIB:$NS2_LIB:$X11_LIB:$USR
_LOCAL_LIB

# TCL_LIBRARY
TCL_LIB=/your/path/ns-allinone-2.31/tcl8.4.14/library
USR_LIB=/usr/lib
export TCL_LIBRARY=$TCL_LIB:$USR_LIB

# PATH
```

```
XGRAPH=/your/path/ns-allinone-2.31/bin:/your/path/ns-allinone-
2.31/tcl8.4.14/unix:/your/path/ns-allinone-2.31/tk8.4.14/unix
NS=/your/path/ns-allinone-2.31/ns-2.31/
NAM=/your/path/ns-allinone-2.31/nam-1.13/
PATH=$PATH:$XGRAPH:$NS:$NAM

Let it take effect immediately
source ~/.bashrc
```

6) Now close all files opened and restart console and write ns and press enter.

NOTE:

If there is no error and an environment starts and shows % sign, it means the installation is successful and now start with this link: Marc Greis' Tutorial for the CB/LBNL/VINT Network Simulator "ns" Read the manual and begins learning of NS-2.

A.2 ISE Design Flow

The ISE® Design Suite is the Xilinx® design environment, which allows taking design from design entry to Xilinx device programming. With specific editions for logic, embedded processor, or Digital Signal Processing (DSP) system designers, the ISE Design Suite provides an environment tailored to meet specific design needs.

The ISE Design Suite allows going from design entry, through implementation and verification, to device programming from within the unified environment of the ISE Project Navigator or from the command line. This includes exclusive tools and technologies to help achieve optimal design results, including the following:

- **Xilinx Synthesis Technology (XST) -** synthesizes VHDL, Verilog, or mixed language designs.
- **ISim -** enables to perform functional and timing simulations for VHDL, Verilog and mixed VHDL/Verilog designs.
- **PlanAhead™ software -** enables to do advanced FPGA floorplanning. The PlanAhead software includes I/O Planner, an environment designed to help you to import or create the initial I/O Port list, group the related ports into separate folders called “Interfaces” and assign them to package pins. I/O Planner supports fully automatic pin placement or semi-automated interactive modes to allow controlled I/O Port assignment. With early, intelligent decisions in FPGA I/O assignments, you can more easily optimize the connectivity between the PCB and FPGA.

- **CORE Generator™ software -** provides an extensive library of Xilinx LogiCORE™ IP from basic elements to complex, system-level IP cores.
- **SmartGuide™ technology -** enables to use results from a previous implementation to guide the next implementation for faster incremental implementation.
- **Design Preservation -** enables to use placement and routing for unchanged blocks from a previous implementation to reduce iterations in the timing closure phase.
- **Team Design -** enables multiple engineers to synthesize and implement portions of a design independently.
- **Partial Reconfiguration -** enables dynamic design modification of a configured FPGA. The ISE software uses Partition technology to define and implement static and reconfigurable regions of the device. This feature requires an additional license code.
- **XPower Analyzer -** enables to analyze power consumption for Xilinx FPGA and CPLD devices.
- **iMPACT -** enables to directly configure Xilinx FPGAs or program Xilinx CPLDs and PROMs with the Xilinx cables. It also enables to create programming files, readback and verify design configuration data, debug configuration problems, and execute SVF and XSVF files.

The ISE® design flow is shown in the **Figure A.1.**

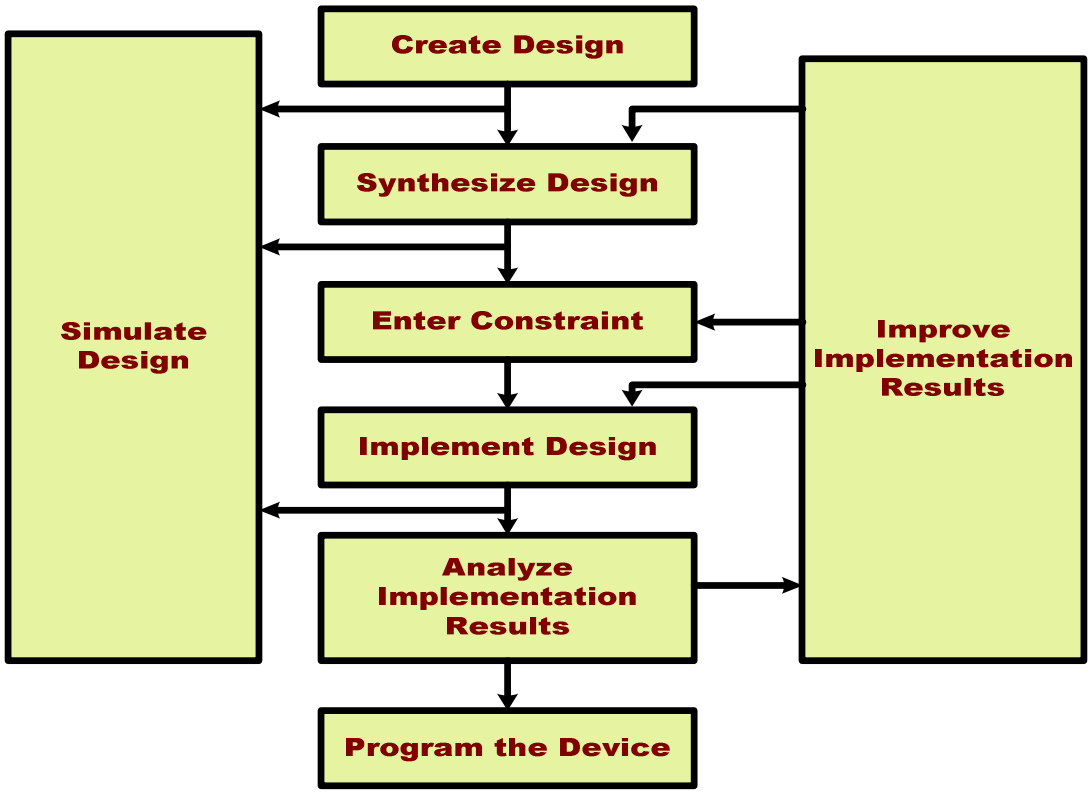


Figure A.1: ISE Design flow

∝ Design Creation

During design creation, we create an ISE project and then, create or add source files to that project. ISE projects can contain many types of source files and design modules, including HDL, EDIF/NGC netlist, schematic, intellectual property (IP), embedded processor, and Digital Signal Processing (DSP) modules.

∝ Synthesis

During synthesis, the synthesis engine compiles the design to transform HDL sources into an architecture-specific design netlist. The ISE software supports the use of Xilinx Synthesis Technology (XST), which is delivered with the ISE software, as well as third party synthesis tools, including, Simplify Pro, and Precision software.

∝ Simulation

At various points during the design flow, we can verify the functionality of the design using a simulation tool. From within the ISE viewing environment, we can use ISim, which is delivered with the ISE software or ModelSim simulators. Alternatively, we can simulate our design outside of ISE Project Navigator using any supported simulator.

∝ Constraints Entry

Using design constraints, we can specify timing, placement, and other design requirements. The ISE software provides editors to facilitate constraints entry for timing constraints as well as I/O pin and layout constraints.

∝ Implementation

After synthesis, we run design implementation, which converts the logical design into a physical file format that can be downloaded to the selected target device. Using the Project Navigator Design Goals and Strategies, we can modify process properties to control the implementation and optimization of the design. To attempt to meet our design goals faster, you can use SmartXplorer to automate multiple implementation runs with different process properties.

∝ Implementation Analysis

After implementation, we can analyze your design for performance against constraints, device resource utilization, timing performance, and power utilization. We can view results in static report files and by looking at actual device implementation in graphical layout tools, such as the PlanAhead™ software and FPGA Editor. We can interactively analyze timing and power results using the Timing

Analyzer and XPower Analyzer tools. And, We can perform in-system debugging using the ChipScope™ Pro tool.

∝ Implementation Improvement

Based on the analysis of our design results, we can make changes to design sources, process properties, or design constraints and then, rerun synthesis, implementation, or both to achieve design closure.

∝ Device Configuration and Programming

After generating a programming file, we can configure our device. During configuration, we generate configuration files and download the programming files from a host computer to a Xilinx® device.

After generating a programming file using the Generate Programming File process, we can configure our device; create PROM, System ACE™ solution, SVF, XSVF, or STAPL files. We can configure FPGAs or program Xilinx® CPLDs or PROMs in-system, directly from a host-computer using iMPACT with a Xilinx download cable.

A.2.1 iMPACT Files

∝ Input Files

For device configuration, we need to provide JEDEC or ISC files for each Xilinx® CPLD device, BMM, ELF, BIT, or ISC files for each Xilinx FPGA device, MCS, or ISC files for each Xilinx PROM device, and MCS, or HEX files for the supported SPI or BPI PROMs (SPI PROMs: Numonyx: M25P, M25PE, M45PE, N25Q, Atmel: AT45DB, Winbond: W25Q, or Spansion S25FL-P; BPI PROMs: Numonyx P30 (Strataflash™, Axcell™), J3, Spansion: S29GL-P and Xilinx: XCF128X).

We need to provide BIT files for PROM file generation, BMM, ELF, BIT, or BSDL files for System ACE™ CF file generation, and BMM, ELF, BIT, JEDEC, ISC, MCS, or BSDL files for SVF, STAPL, and XSVF file generation.

For project management, we need to provide the iMPACT Project file (IPF). For compatibility with previous versions of iMPACT, we can also use a Chain Description File (CDF) for retrieving saved project information.

∝ Boundary-Scan Description Language File (BSDL)

BSDL files are for Boundary-Scan programming only.The Boundary-Scan Description Language files use a subset of VHDL to describe the Boundary-Scan features of a device. iMPACT automatically recognizes and references Xilinx BSDL files which are loaded with the iMPACT install.

Use the Assign New Configuration File dialog box to specify the location of BSDL files for non-Xilinx devices. iMPACT automatically extracts the length of the instruction register from the BSDL file to place non-Xilinx devices in bypass mode. iMPACT can generate a minimal BSDL file automatically for devices for which the actual BSDL file is unavailable. The required extension for a BSDL file is .bsd.

∝ Configuration Format Information File (CFI)

A Configuration Format Information file is created by PROMGen and is used to collect additional information that describes the options and information content of the PROM data files. The required extension for a Configuration Format Information file is .cfi.

∝ Output Files

Output files are MCS, HEX, BIN, or ISC files for PROM file generation, ACE files for System ACE CF file generation, and SVF, STAPL, or XSVF files for SVF- STAPL-XSVF file generation and IPF files for project management.

∝ PROM Files (MCS/HEX/UFP/BIN/ISC)

PROM files are PROM programming files generated in iMPACT using the PROM File Formatter. They are ASCII text files used to specify configuration data. One PROM file is required for each Xilinx PROM in the Boundary-Scan chain. Use the Assign New Configuration File dialog box to specify the location of the MCS or ISC files for each Xilinx PROM. The required extensions for MCS and ISC are .mcs and .isc, respectively. Indirect PROM programming, in which an SPI or BPI PROM attached to a Xilinx FPGA is programmed using a proprietary IP core, also requires that each SPI or BPI PROM have one PROM file associated with it. One PROM file is also required for the SPI PROM in the Direct SPI mode. Use the Assign New Configuration File dialog box to specify the location of the MCS or HEX file for the selected SPI PROM (Numonyx: M25P, M25PE, M45PE, N25Q, Atmel: AT45DB, Winbond: W25Q, or Spansion S25FL-P).

UFP (User Formatted PROM) generates a PROM data file in a format appropriate for inclusion in a C programming language application. This is appropriate for users who are interested in developing their own software applications for configuring Xilinx FPGAs. The extension for a User Formatted PROM file is .ufp.

BIN (Binary) generates a PROM data file in binary format. This is appropriate for users who are interested in developing their own software applications for configuring Xilinx FPGAs. The extension for a Binary file is .bin.

⁂ Platform Cable USB/USB II

Platform Cable USB and Platform Cable USB II are high-performance download cables that attach to the USB port of computer for the purpose of programming or configuring any of the following Xilinx devices:

- Xilinx ISP configuration PROMs
- Xilinx CPLDs
- Xilinx FPGAs
- Select third-party SPI PROMs
- Select third-party BPI PROMs

A.3 Code Generation Using Xilinx ISE 13.1

The Xilinx® ISE Simulator (ISim) is a Hardware Description Language (HDL) simulator that enables to perform functional (behavioral) and timing simulations for VHDL, Verilog and mixed-language designs.This ISE Simulator environment is comprised of the following key elements:

- Vhpcomp (VHDL compiler
- Vlogcomp (Verilog compiler)
- fuse (HDL elaborator and linker)
- Simulation Executable
- isimgui (ISim Graphical User Interface)

We will use the New Project Wizard in ISE Project Navigator to quickly create an ISE project for the design.

1. Launching Project Navigator and Using New Project Wizard

Follow these steps to launch Project Navigator software and create an ISE project.

1) Double-click on the Xilinx ISE 13.1 desktop icon to launch the ISE Project Navigator shown in **Figure A.2**.

Figure A.2: Snap Shot Of Xilinx ISE Design Suite Icon

2) Click the New Project button to launch the New Project Wizard.

3) Provide a name and an appropriate location for the project (Refer to **Figure A.3**).

4) Click **Next** to continue.

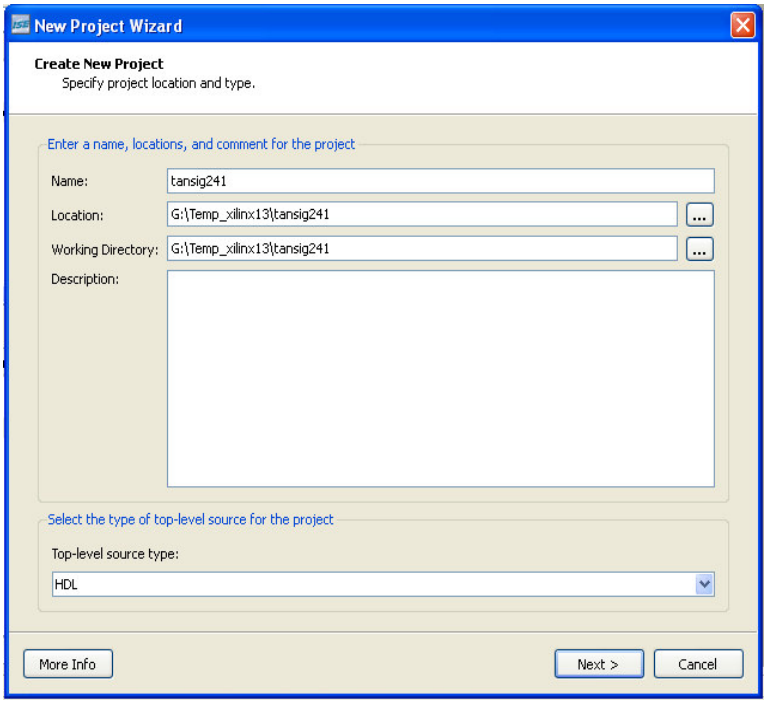


Figure A.3: New Project Wizard: Create New Project Page

5) In the window, select the device and project properties.

6) Change the settings according to FPGA board shown in **Figure A.4**.

7) Click Next to continue.

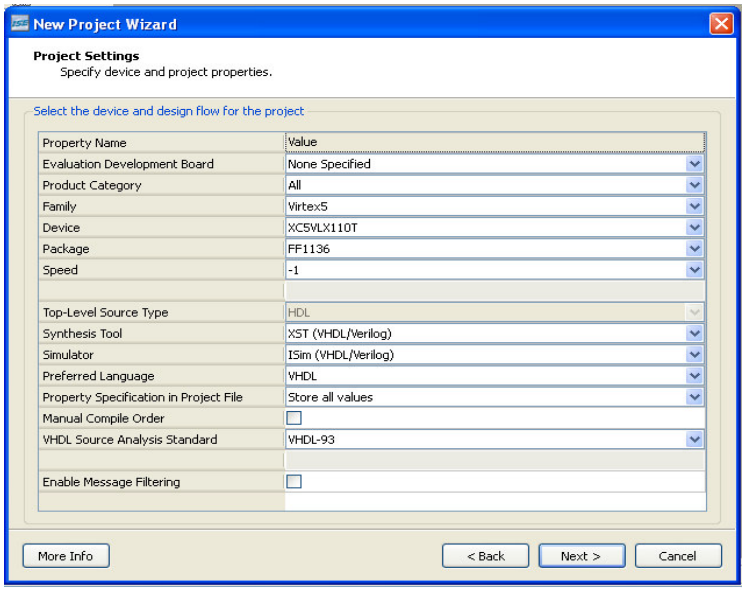


Figure A.4: New Project Wizard: Project Settings

8) Review the Project Summary page and make sure that the settings match those (**as shown in Figure A.5**).

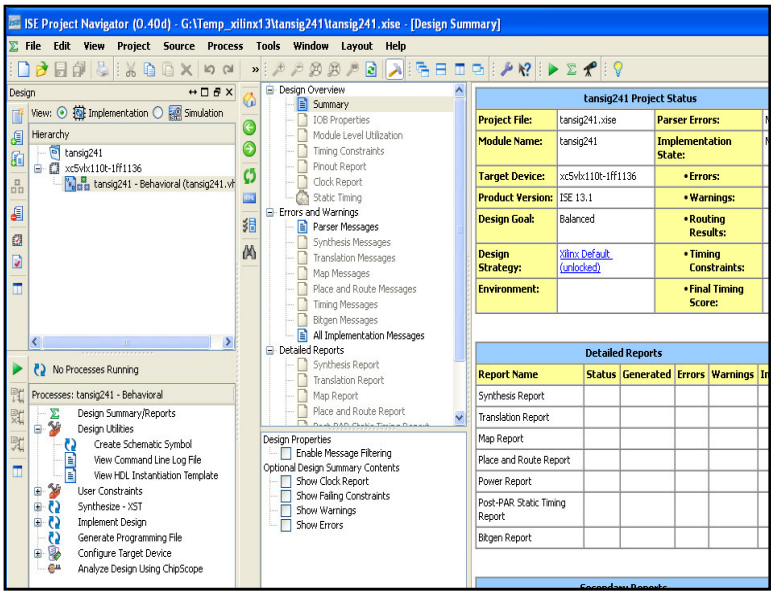


Figure A.5: Project Summary

9) Click Finish to continue.

2. Adding Source Files to the Project

1. Click the Add Source button in the Design Panel toolbar to select the sources provided for this tutorial.

2. In the next window, make sure that the association and libraries have been properly specified for the tutorial sources. Compare settings with those in **Figure A.6**.

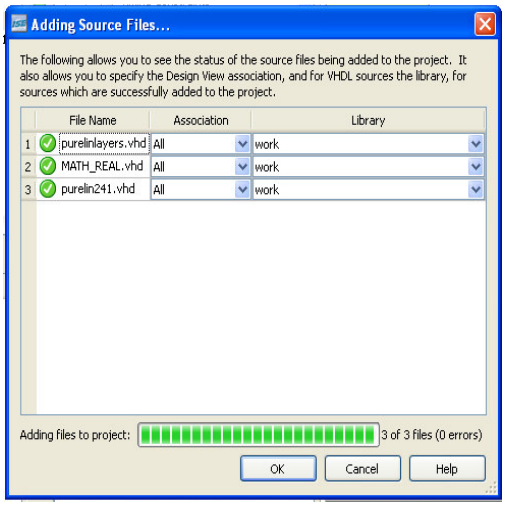


Figure A.6: Status of Source Files and Associations

3. Click **OK.**

The source files are added to the project.

3. Launching a Behavioral Simulation

1) Now that the ISE project has been created for the tutorial design, we can proceed to set up and launch a behavioral simulation using ISim.

Setting Behavioral Simulation Properties

- To set behavioral simulation properties in ISE:In the Design Panel, select **Behavioral Simulation** from the dropdown list.
- We should now see the simulation processes available for the design in the Processes pane. (Refer to **Figure A.7**)

2) Right-click Simulate Behavioral Model under the ISim Simulator process and select Properties. The ISim Properties dialog box displays (Refer to **Figure A.8**).

- In this window we can set different simulation properties, such as simulation runtime, waveform database file location, and even a user-defined simulation command file to launch the simulation.
- For the purposes of this tutorial, we will disable the feature that runs the simulation for a specified amount of time.

3) In the ISim Properties dialog box, uncheck the property Run for Specified Time, and click OK. (Refer to **Figure A.8**)

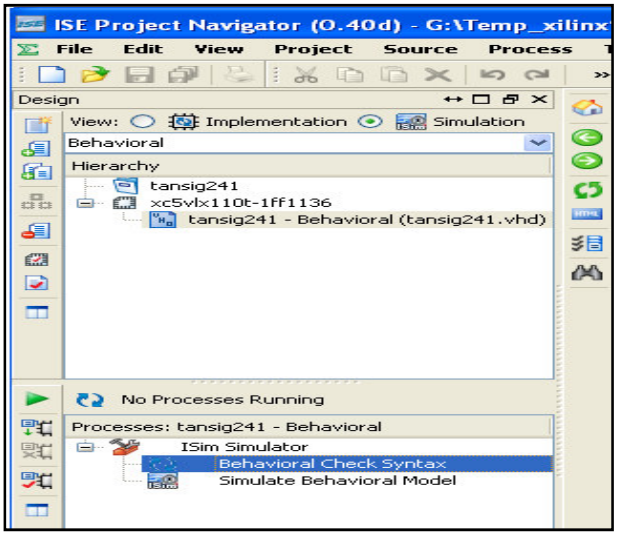


Figure A.7: Process Pane

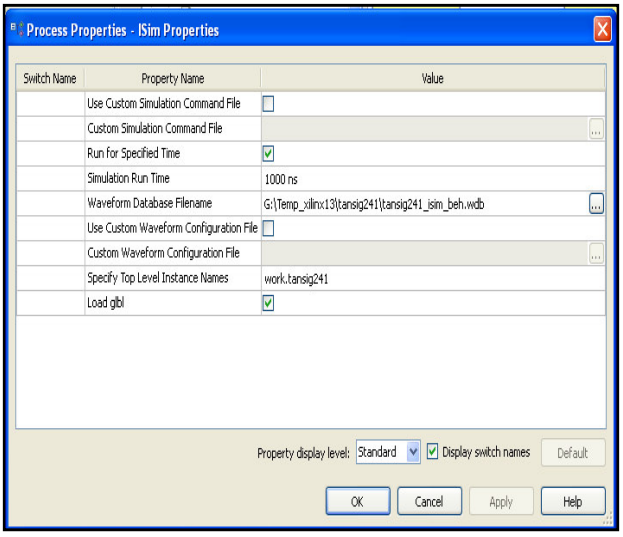


Figure A.8: ISim Properties Dialog Box

Now it is ready to launch the ISE Simulator to perform a behavioral simulation of the tutorial design. To launch the simulator:

In the Processes panel, double-click **Simulate Behavioral Model**.

- The ISim Graphical User Interface (GUI) (**Figure A.9**) will appear shortly after the design is successfully parsed and compiled.

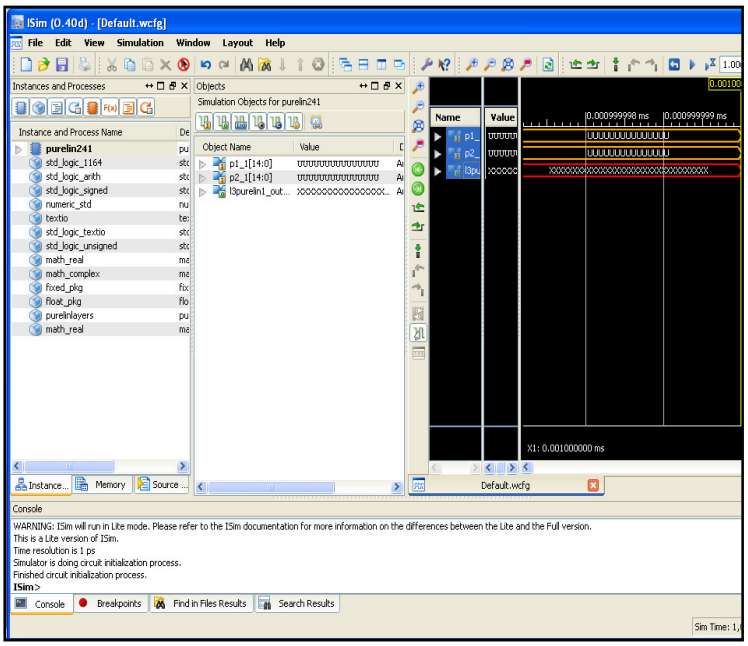


Figure A.9: ISim Graphical User Interface

The Console panel enables to view a log of messages generated by ISim, and to enter standard Tcl and ISim-specific commands at the command prompt.

The wave window now shows traces of the signals up to 5 microseconds in simulation time (Refer to **Figure A.10**).

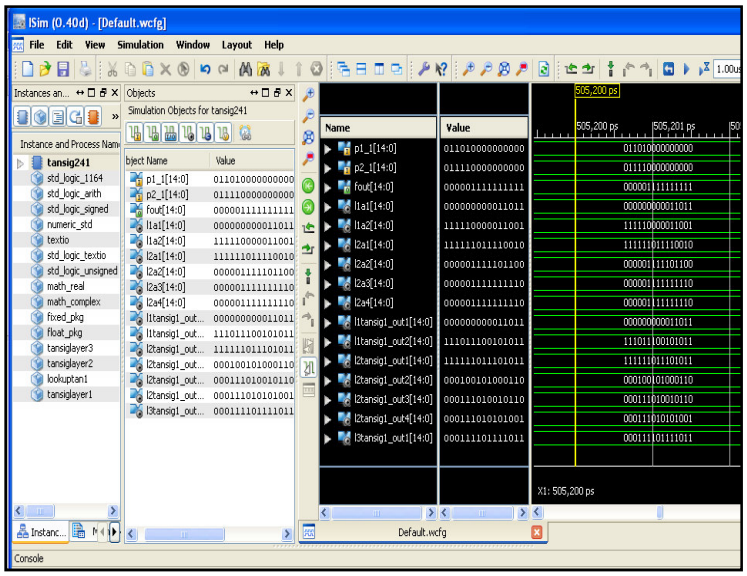


Figure A.10: Wave Window

 Display the full time spectrum in the Wave window, select **Edit > Zoom > Zoom Full View** or click the Zoom Full View button.

A.3.1 To Configure or Program a Device

The following steps are requiring to configure or to program a device:

1. Connect the cable to an appropriate port on the host computer and to the correct pins on the target board.

   When setting up the configuration chain and target device to configure or program a device, select the appropriate configuration mode, which is used to connect to, the boards from Xilinx download cable. The most commonly used mode is **Boundary Scan.** When using this mode,we must connect the cable leads to the following pins: TDO, TDI, TCK, TMS, VCC, and GND
2. In the View pane of the **Design panel,** select **Implementation**.
3. In the Hierarchy pane, select the top module.

In the Processes pane, double-click **Configure Target Device**.

A.3.1.1 Generating PROM Files

This operation generates PROM file(s) in PROM File Formatter mode (.mcs, .hex) or System ACE™ file(s) (.ace) using the bitstream and device information that we have already entered.

To generate PROM files follow the steps.

- Double-click **Generate File** in the iMPACT Processes panel or right-click in the PROM File Formatter window and select **Generate File**.

PROM file is generated and a large message displaying the result appears in the graphical view. A PROM file and a Configuration Format Information (.CFI) file are generated. The .CFI file is used to save advanced setup information for the PROM that automates the PROM programming flow. For optimal application performance, keep PROM and CFI files in the same directory.

After generating the file, use Boundary-Scan Mode to program a PROM using the PROM file.

A.3.1.2 Introduction to Boundary-Scan (JTAG)

Boundary-Scan (IEEE Std 1149.1, JTAG) configuration mode enables to perform Boundary-Scan-based configuration operations on any chain of IEEE Std 1149.1 compliant devices. The chain can consist of both Xilinx® and non-Xilinx devices, but only the BYPASS and HIGHZ operations are available for non-Xilinx devices.

Boundary-Scan is the most popular configuration mode due to its standardization and ability to program FPGAs, CPLDs, and PROMs through the same four JTAG pins. The data in this mode is loaded one bit per TCK pulse.

Before a chain or device operation is attempted, a Xilinx download cable must be connected from the computer to the target system JTAG pins: TDI, TCK, TMS, and TDO. The Xilinx download cable power pins need to be connected from the cable to the board. The Boundary-Scan chain that is created in the graphical view in iMPACT must match the chain on the board, exactly. This means that if the chain consists of eight devices, but only one of them is going to be configured, all eight devices must be added to the chain in the exact same order as they occur on the board in the graphical view.

- Select **Operations > Initialize Chain** or right-click in the Workspace and select **Initialize Chain (as shown in Figure A.11(a)),** after doing initialize chain Xilinx devices are available in boundry scan panel**(as shown in Figure A.11(b))**.

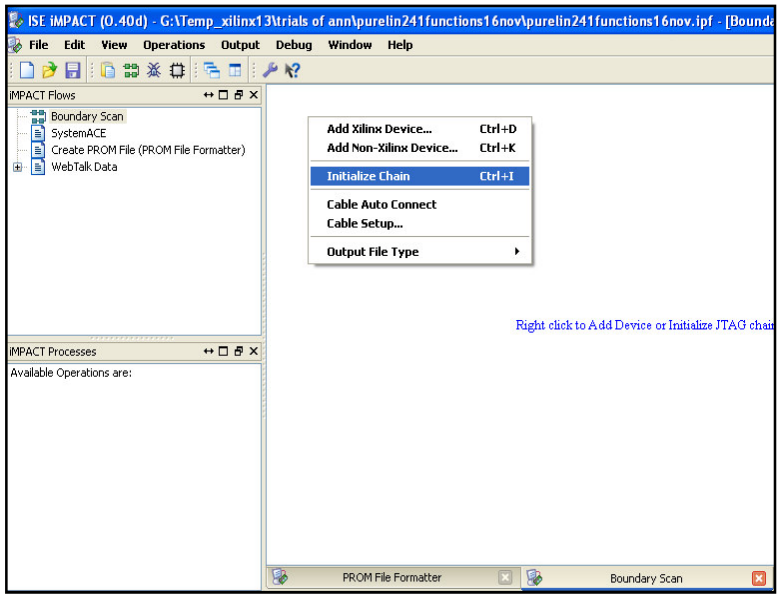


Figure A.11 (a): snap shot of boundary scan and adding Xilinx devices

Toolbar Icon: 

Data is passed through the devices and automatically identifies the size and composition of the Boundary-Scan chain. Any supported Xilinx® device is recognized and labeled and any other device is labeled as unknown. After the chain composition is completed, optionally have each device highlighted in the chain so you can be prompted to add an associated configuration file. Otherwise, we can add an associated configuration file to any device in the chain manually.

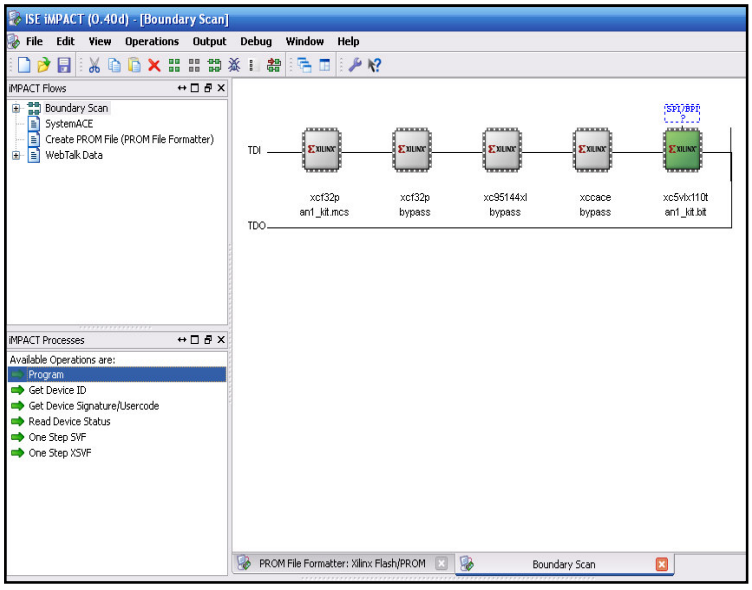


Figure A.11 (b): snap shot of boundary scan after adding Xilinx devices

After adding all the Xilinx devices assign the .mcs and .bin file to the PROM and FPGA device respectively using file launch assignment wizard. After assigning file select the target device of XCV5LX110T FPGA board and select load to FPGA option and press apply and then OK. Now double click on program option by right click on target device or double click on program in "**Impact Processes"** Pane, Which will now program the FPGA Chip and display the message as shown in **Figure A.12**.

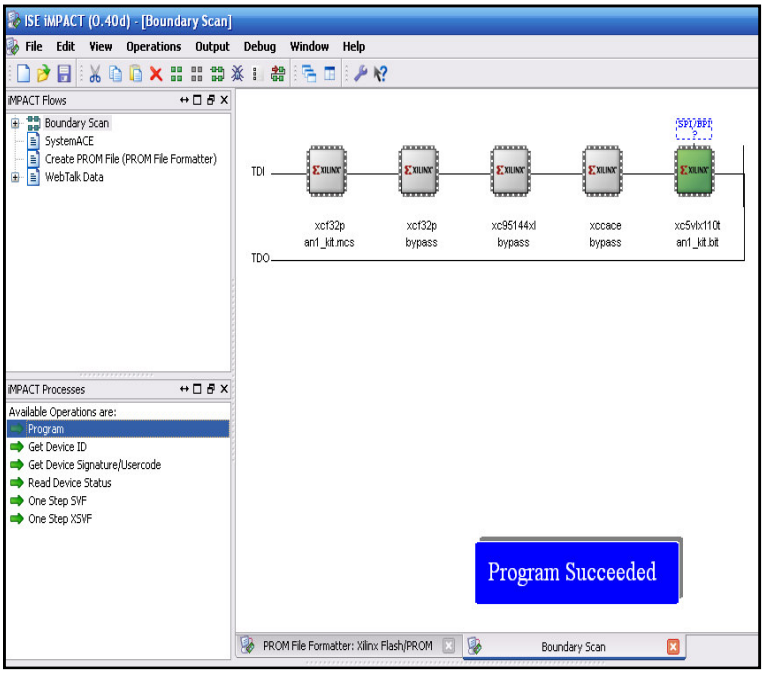


Figure A.12: PROGRAM Launch window

A.4 XCV5LX110T FPGA Board

The Field Programmable Gate Arrays(FPGA) are the family of programmable logic devices based in an array of configurable logic blocks which gives a great flexibility in the development of the digital system. The XCV5LX110T FPGA evaluation board provides system design engineers with an easy- to-use, cost effective way to take their high performance FPGA designs from the concept to production. It provides facility for fast development of networking, communication, imaging and other applications. **Figure A.13 (a) and Figure A.13 (b)** depicts the FPGA board. It operates at the 100 MHz. The board supports configuration in all modes: JTAG, Master Serial, Slave Serial, Master SelectMAP, Slave SelectMAP, Byte-wide Peripheral Interface (BPI) Up, BPI Down, and SPI modes. The board includes Two Xilinx XCF32P Platform Flash PROMs (32 Mb each) for storing large device configurations, Xilinx System ACE™ Compact Flash configuration controller with Type I Compact Flash connector, Xilinx XC95144XL CPLD for glue logic, 64-bit wide, 256-MB DDR2 small outline DIMM (SODIMM), compatible with EDK supported IP and software drivers, RS-232 serial port, DB9 and header for second serial port, 16-character x 2-line LCD display, One 8-Kb IIC EEPROM and other IIC capable devices, PS/2 mouse and keyboard connectors, Video input/output, Video input (VGA), Video output DVI connector (VGA supported with included adapter) JTAG configuration port for use with Parallel Cable III, Parallel Cable IV, or Platform USB download cable, Onboard power supplies for all necessary voltages.

A Xilinx XC95144XL CPLD provides general-purpose glue logic for the board. The CPLD is programmed from the main JTAG chain of the board. The CPLD is mainly used to implement level translators, simple gates, and buffers. The XC95144XL is a 3.3 V CPLD targeted for high-performance, low-voltage applications in leading-edge communications and computing systems. It is comprised of eight 54V18 Function Blocks, providing 3,200 usable gates with propagation delays of 5 ns.

In addition to the high-speed I/O paths, additional I/O signals and power connections are available to support expansion cards plugged into the ML50x board. Fourteen I/O pins from the general purpose pushbutton switches and LEDs on the board are connected to expansion connector J5. This permit additional I/Os to connect to the expansion connector if the pushbutton switches and LEDs are not used. The connection also allows the expansion card to utilize the pushbutton switches and LEDs on the board. The expansion connector also allows the board's JTAG chain to be extended onto the expansion card by setting jumper J21 accordingly.

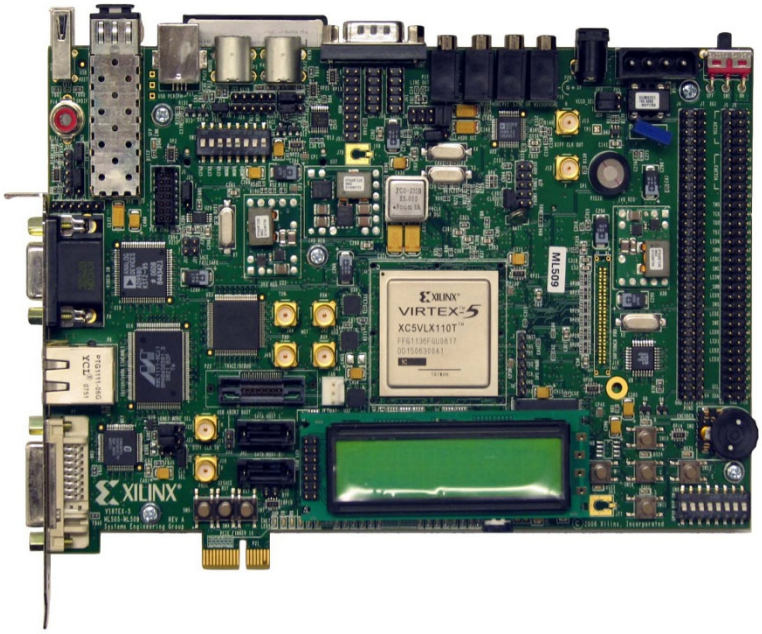


Figure A.13 (a): Snap Shot of XCV5LX110T FPGA Board

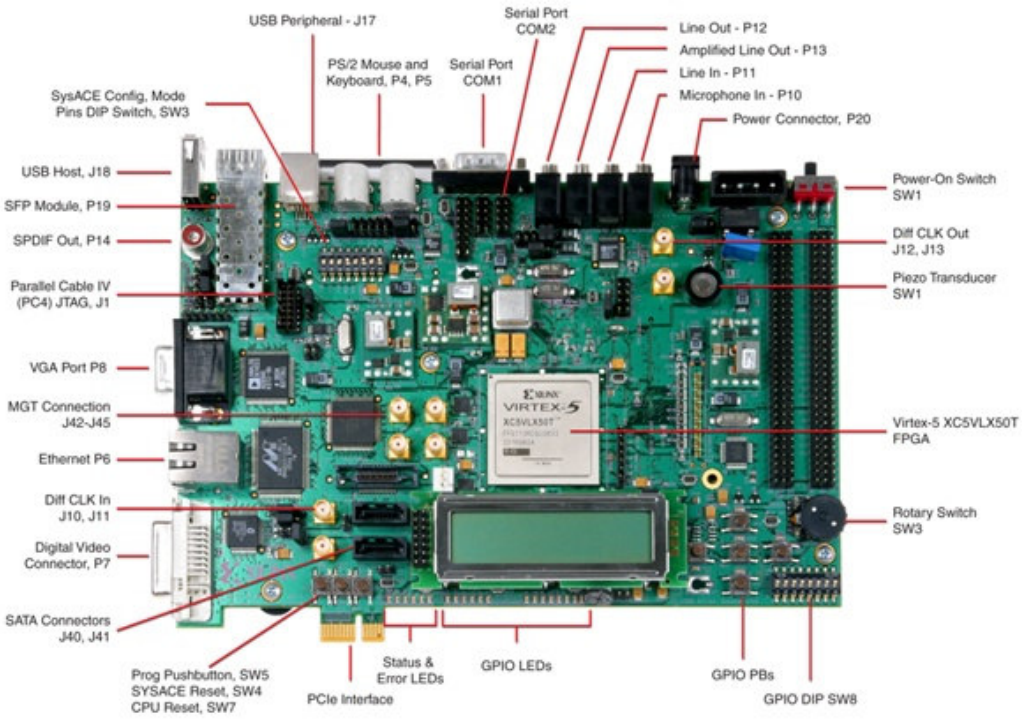


Figure A.13 (b): Snap Shot of XCV5LX110T FPGA Board with peripherals

The IIC bus on the board is also extended onto the expansion connector to allow additional Power supply connections to the expansion connectors provide ground, 2.5V, 3.3V, and 5V power pins. If the expansion card draws significant power from the ML50*x* board, ensure that the total power draw can be

supplied by the board. The ML50*x* expansion connector is backward compatible with the expansion connectors on the ML40*x*, ML32*x*, and ML42*x* boards, thereby allowing their daughter cards to be used with the ML50*x* Evaluation Platform.

The two onboard Xilinx XCF32P Platform Flash PROM configuration storage devices offer a convenient and easy-to-use configuration solution for the FPGA. The Platform Flash PROM holds up to two separate configuration images (up to four with compression) that can be accessed through the configuration address switches. To use the Platform Flash PROM to configure the FPGA, the configuration DIP switch must be set to the correct position. The Platform Flash PROM can program the FPGA by using the master or slave configuration in serial or parallel (SelectMap) modes. The Platform Flash PROM is programmed using Xilinx iMPACT software through the board's JTAG chain.

The ML50*x* board has a 16-character x 2-line LCD (Tianma TM162VBA6) on the board to display text information. Potentiometer R87 adjusts the contrast of the LCD. The data interface to the LCD is connected to the FPGA to support 4-bit mode only. The CPLD is used to shift the voltage level between the FPGA and the LCD

The Xilinx System ACE CompactFlash (CF) configuration controller allows a Type I CompactFlash card to program the FPGA through the JTAG port. Both hardware and software data can be downloaded through the JTAG port. The System ACE controller supports up to eight configuration images on a single CompactFlash card. The configuration address switches allow the user to choose which of the eight configuration images to use. The CompactFlash card shipped with the board is correctly formatted to enable the System ACE CF controller to access the data stored in the card. The System ACE CF controller requires a FAT16 file system, with only one reserved sector permitted, and a sector-per-cluster size of more than one (Unit Size greater than 512). The FAT16 file system supports partitions of up to 2 GB. If multiple partitions are used, the System ACE directory structure must reside in the first partition on the CompactFlash, with the xilinx.sys file located in the root directory. The xilinx.sys file is used by the System ACE CF controller to define the project directory structure, which consists of one main folder containing eight sub-folders used to store the eight ACE files containing the configuration images.

- # Platform Flash PROM Configuration

The Platform Flash PROMs can also be used to program the FPGA. A Platform Flash PROM can hold up to two configuration images (up to four with compression), which are selectable by the two least significant bits of the configuration address DIP switches. The board is wired so the Platform Flash PROM can download bit streams in Master Serial, Slave Serial, Master SelectMAP (parallel), or Slave SelectMAP (parallel) modes. Using the iMPACT tool to program the Platform Flash PROM, the user has the option to select which of the four modes to use for programming the FPGA. The configuration mode

DIP switches on the board must be set to match the programming method being used by the Platform Flash PROM. When set correctly, the Platform Flash PROM programs the FPGA upon power-up or whenever the **Prog** button is pressed.

- ## Linear Flash Memory Configuration

Data stored in the linear flash can be used to program the FPGA (BPI mode). Up to four configuration images can theoretically be supported. The configuration mode DIP switches on the board must be set to 010 for BPI_up or 011 for BPI_down. When set correctly, the FPGA is programmed upon power-up or whenever the Prog button is pressed.

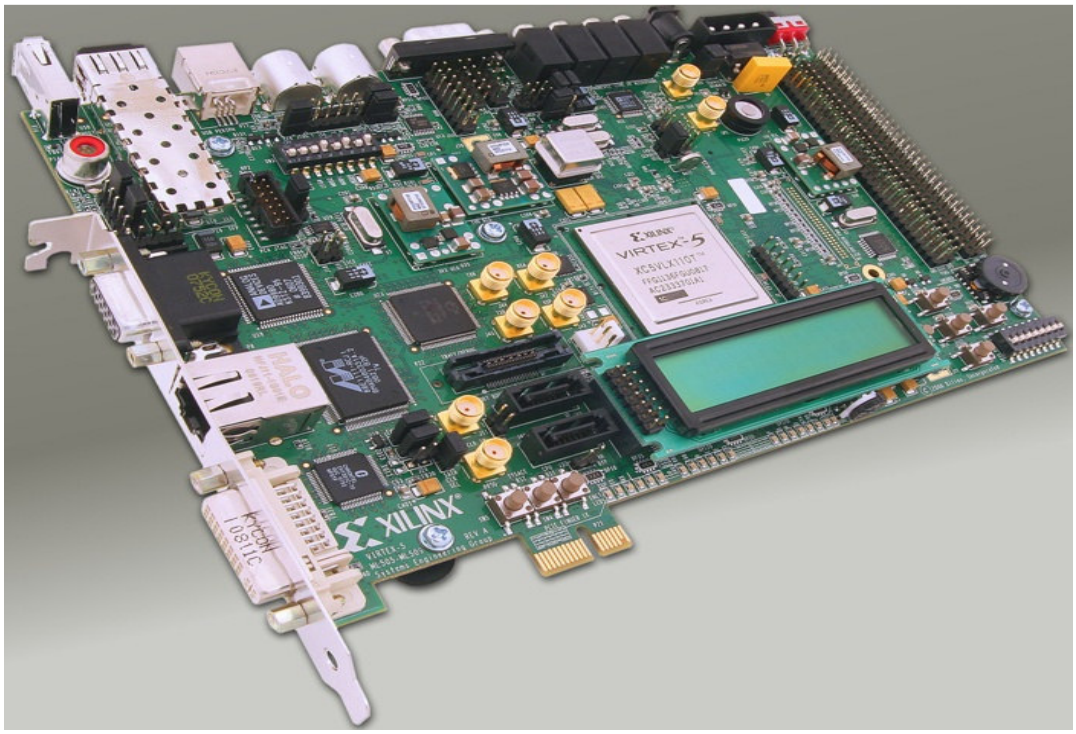
- ## SPI Flash Memory Configuration

Data stored in SPI can be used to program the FPGA. The configuration mode DIP switches must be set to 001 for SPI configuration. When set correctly, the FPGA is programmed upon power-up or whenever the **Prog button** is pressed.

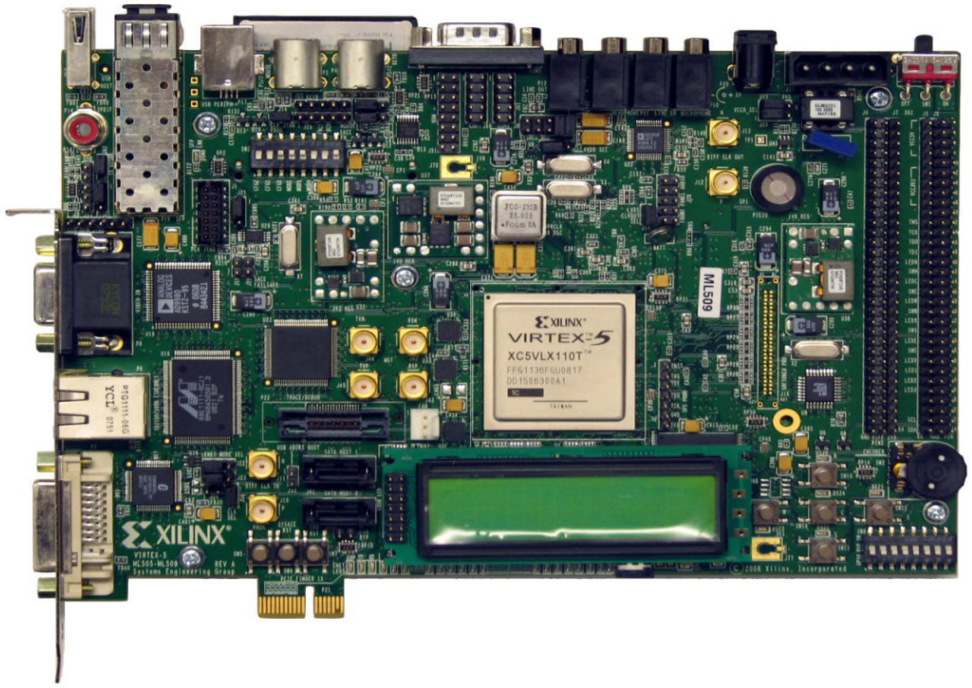
Appendix B

Photogallary

APPENDIX B: PHOTOGALLARY



B-I: XCV5LX110T FPGA Evaluation Board

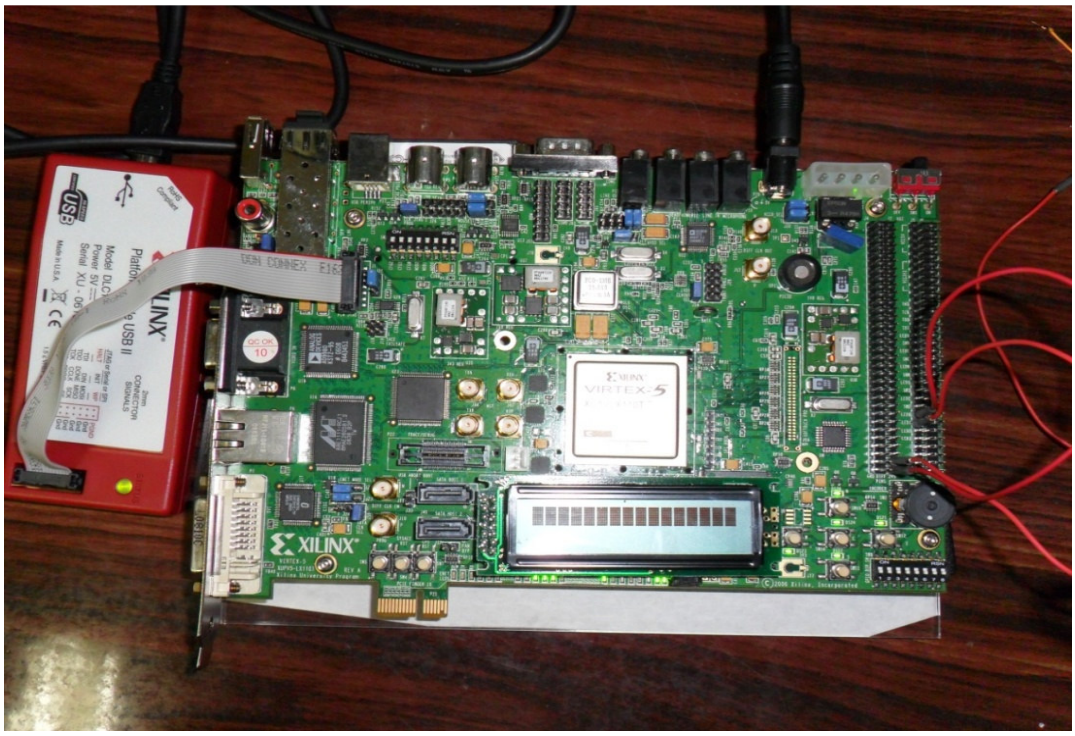


B-II: XCV5LX110T FPGA Evaluation Board: Front View

B-III: System Setup for FPGA Implementation of ANN

B-IV: Output observed on LEDs

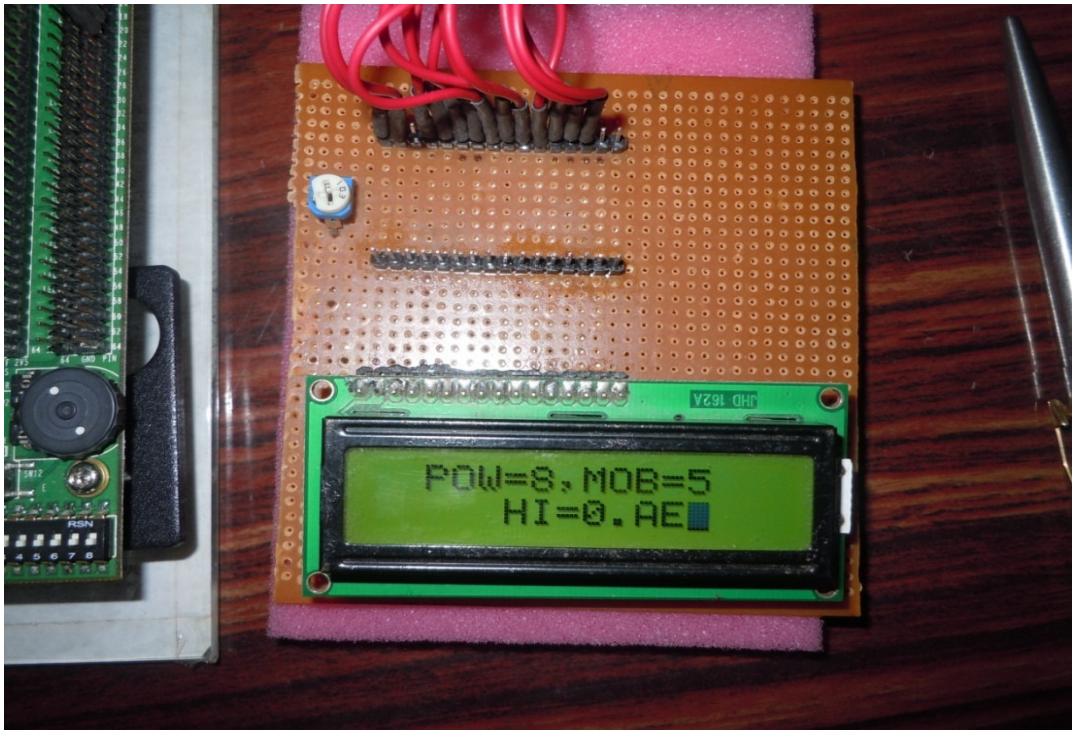
B-IV: Output Observed On LEDs for Different Inputs



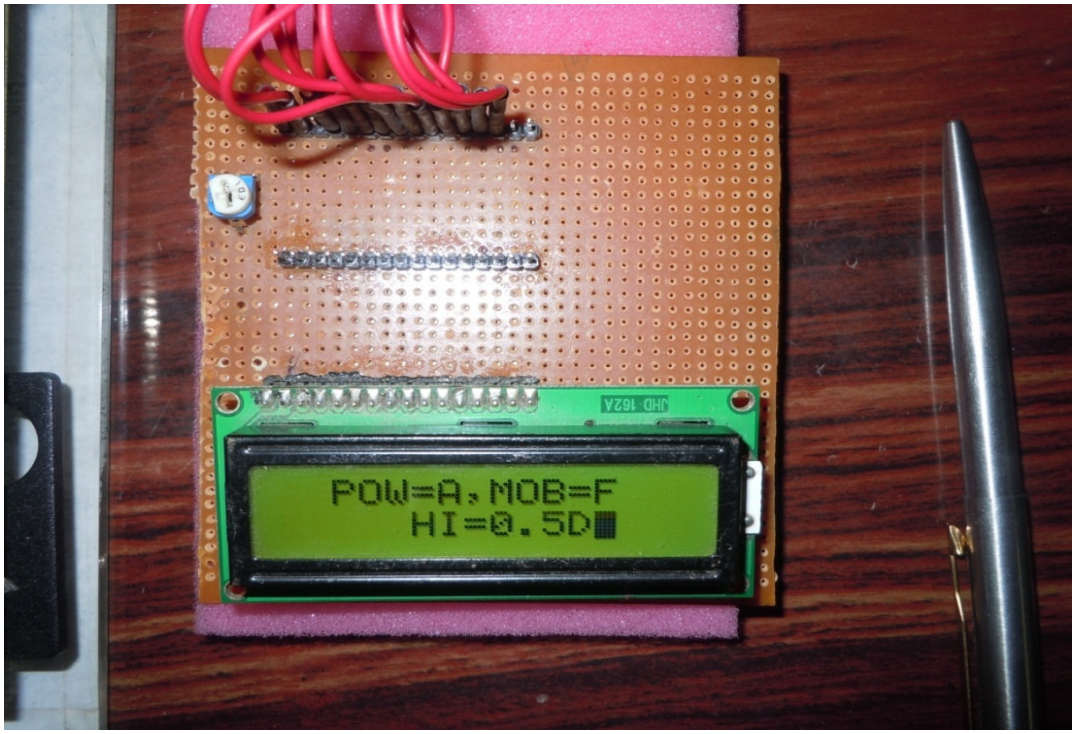
B-V: Output Observed On GPIO for Different Inputs

B-VI: Output Observed on GPIO for Different Inputs using Logic Analyzer

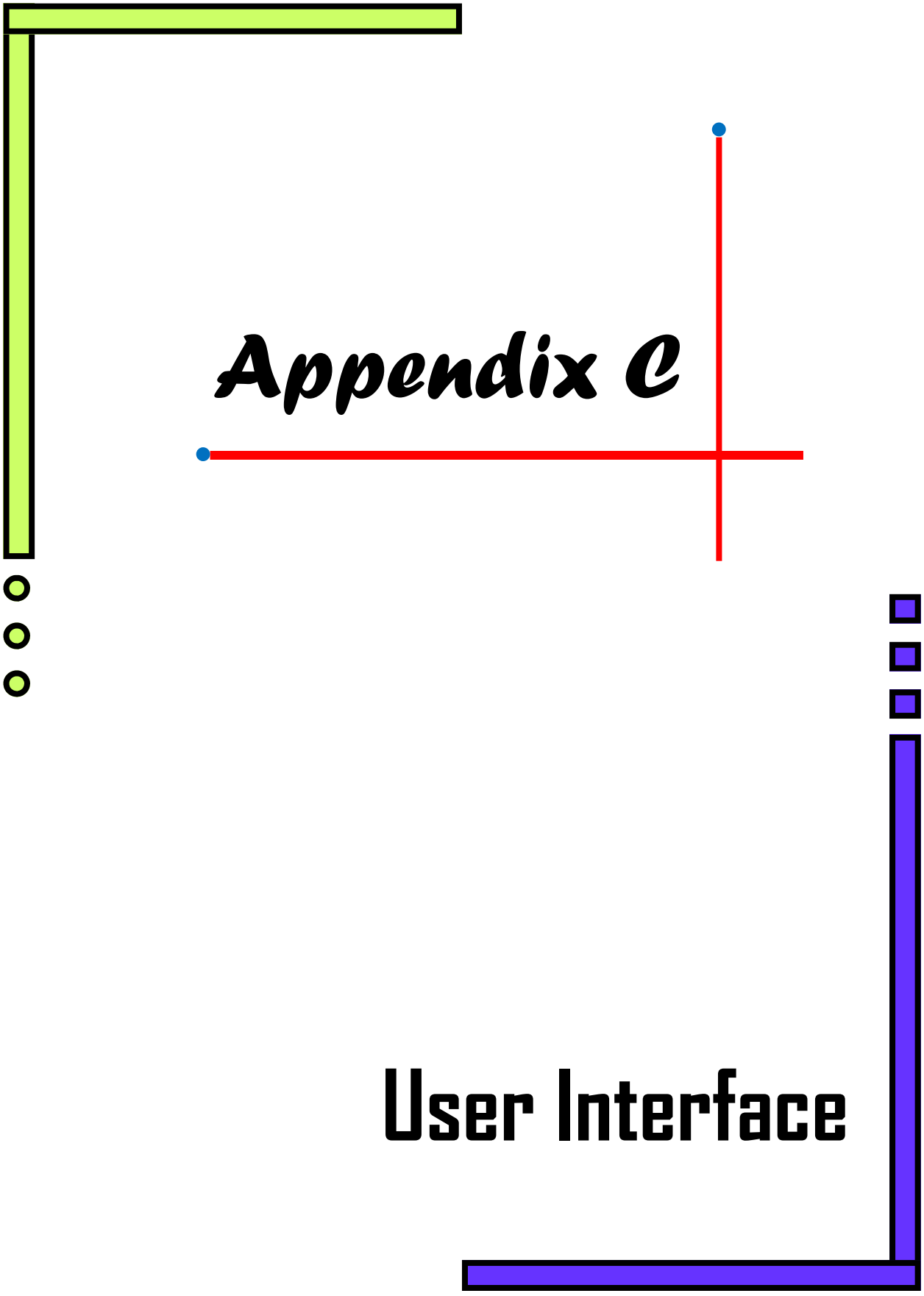
B-VII: Output displayed on LCD for Different Inputs for Tansig ANN



B-VII: Output displayed on LCD for Different Inputs of Purelin ANN



B-VIII: Output displayed on LCD for Different Inputs of Tansig ANN



Appendix C

User Interface

Appendix C: User Interface

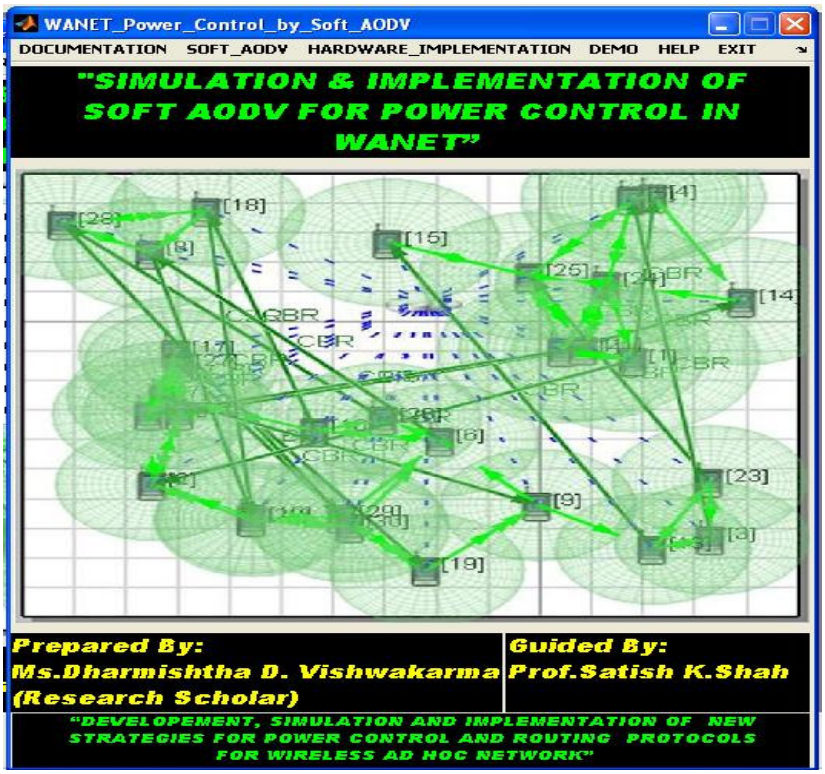


Figure C-I: WANET: Power Control by SOFTAODV User Interface

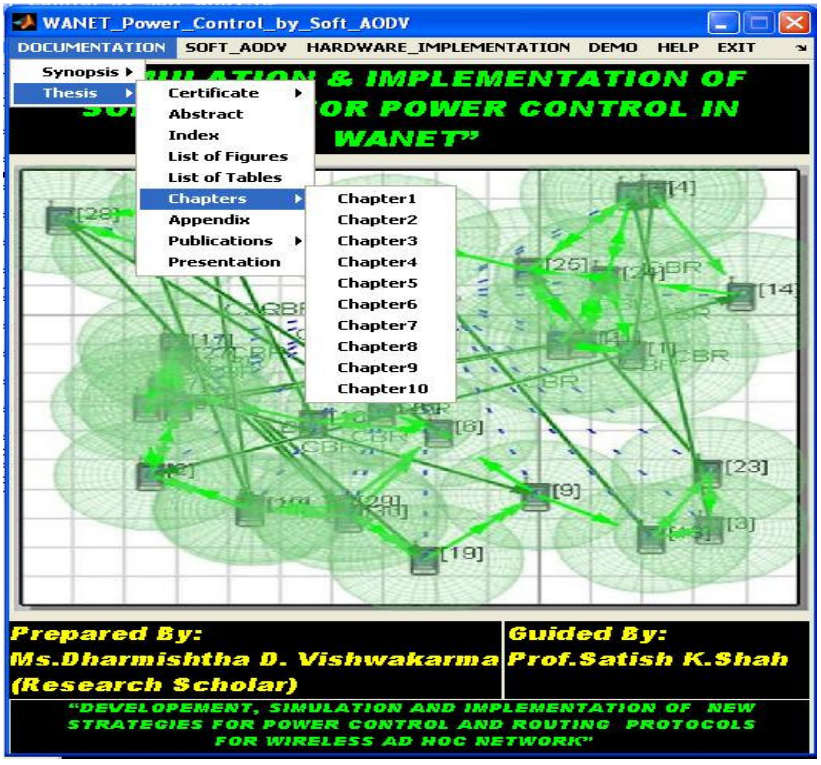


Figure C-II: WANET: Power Control by SOFTAODV User Interface, Documentation

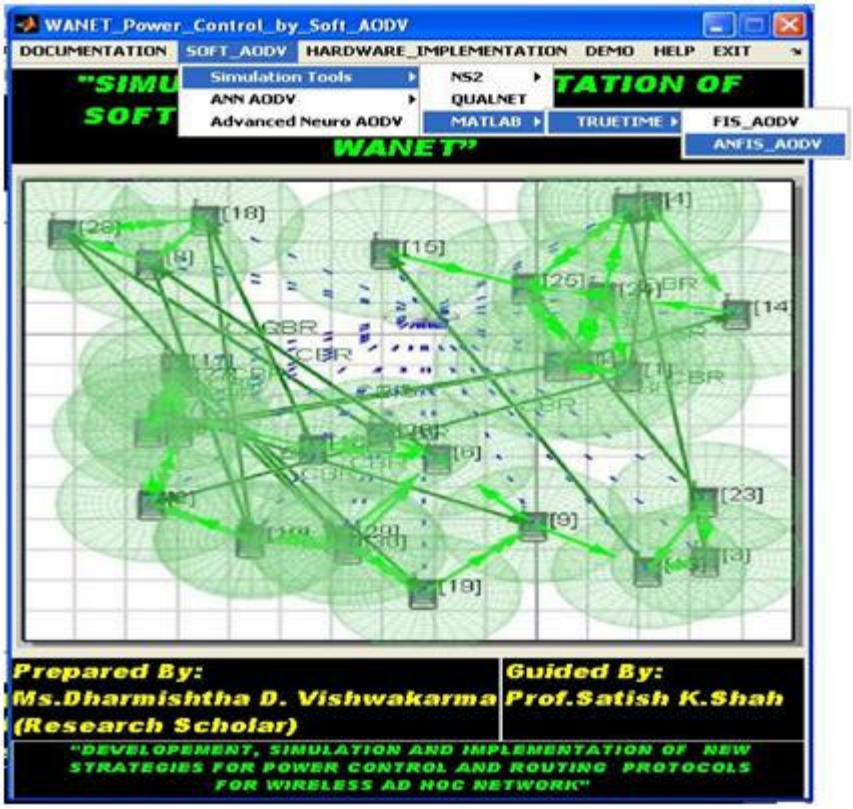


Figure C-III: WANET: Power Control by SOFTAODV User Interface, SOFT_AODV Menu

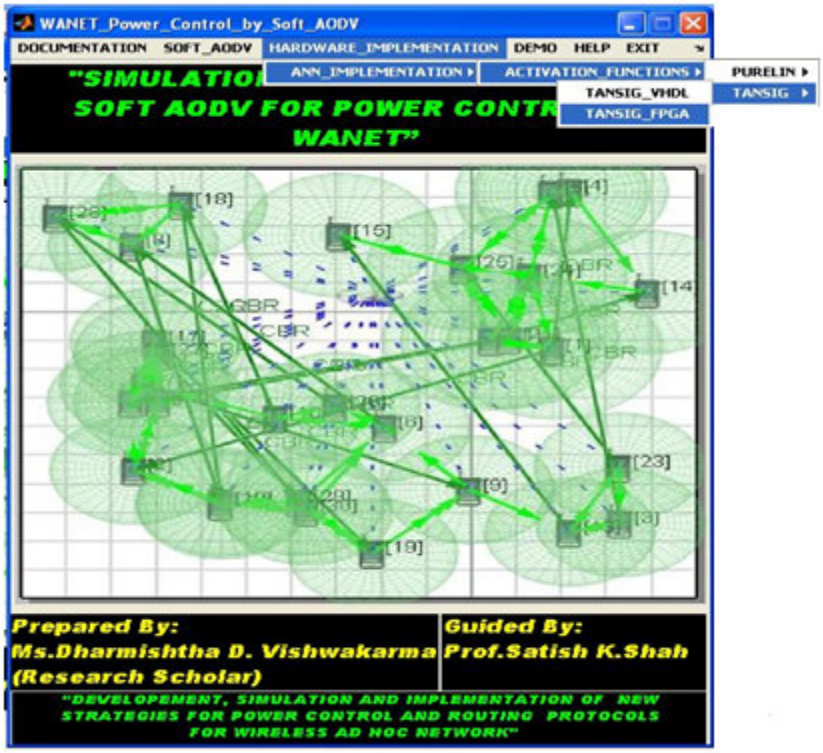


Figure C-IV: WANET: Power Control by SOFTAODV User Interface, Hardware implementation

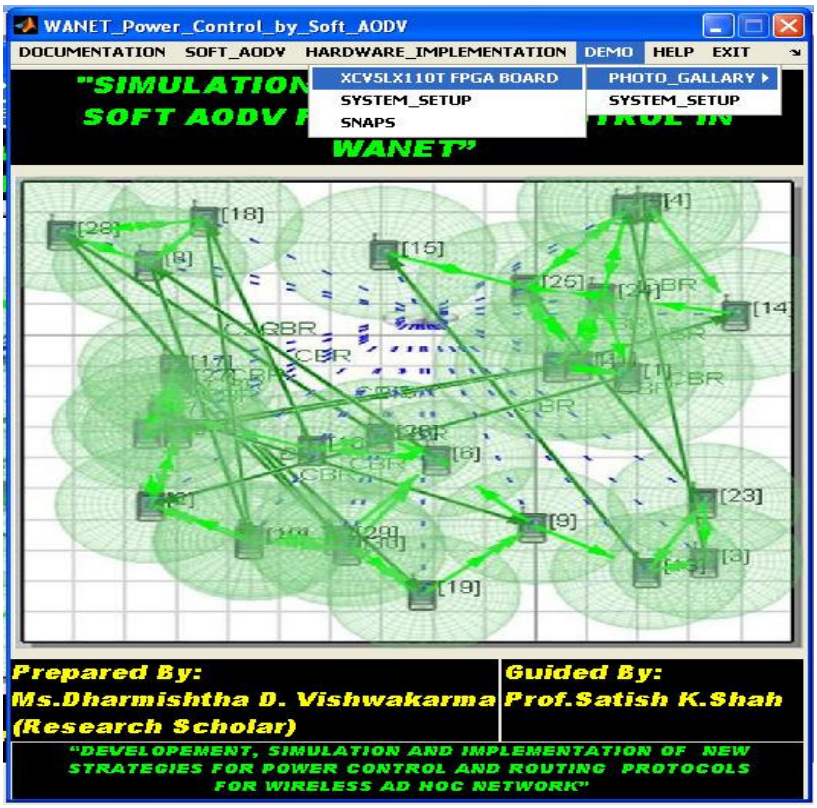


Figure C-V: WANET: Power Control by SOFTAODV User Interface, DEMO

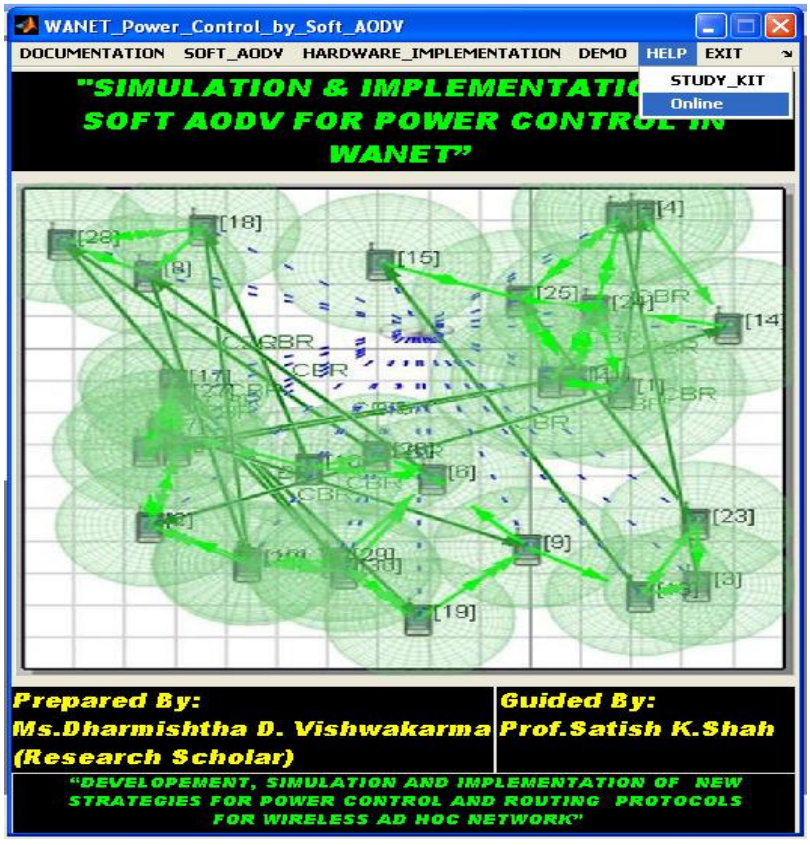


Figure C-VI: WANET: Power Control by SOFTAODV User Interface, Help

Appendix D

List of Papers Based on Research work

Appendix D: List of Papers based on Research work

Appendix D1:

Publications – Proceedings/Referred (National/International) Journals

1) Published in the Paper entitled journal of Institution of Engineers (India), Electronics and telecommunication Engineering Division paper entitled "Comparative Performance Analysis of Routing Protocols for WANET employing Qualnet 5" in volume 92, July 2011, page 12-17.

2) Published in Proceeding of International conference Paper entitled "Performance Optimization of Reactive Routing protocol Using Fuzzy Logic for MANET in Qualnet" in the WORLDCOMP'10, The 2010 World Congress in Computer Science, Computer Engineering, and Applied Computing held on July 12-15, 2010, Las Vegas, USA in ICWN'10: Wireless Networks Division page:345-348.

3) Published in Proceeding of International conference Paper entitled "Signal Reach Computation for AODV Power Threshold in WANET employing Fuzzy Logic" in the WORLDCOMP'10, The 2010 World Congress in Computer Science, Computer Engineering, and Applied Computing held on July 12-15, 2010, Las Vegas, USA in ICWN: Wireless Networks Division page:183-187.

4) To be Publish (accepted) Paper entitled "Performance Optimization of Reactive Routing Protocol for Mobile Ad-Hoc Network using Artificial Neural Network" in the International Journal of Sensors and Actuators, IGI Publication.

5) Published in Proceeding of International conference as well as on **IEEE explore (website)** Paper entitled "Development and Simulation of Artificial Neural Network based decision on parametric values for Performance Optimization of Reactive Routing Protocol for MANET using Qualnet" called "COMPUTATIONAL INTELLIGENCE AND COMMUNICATION NETWORKS CICN 2010" page 167-171 held at State University the R.G.P.V., Bhopal, India on 26-28 Nov 2010 organized by MIR Labs (USA) and RGPV Bhopal and co-sponsored by IEEE SMC Society, Mumbai Section.

6) Published in Proceeding of International conference as well as on **IEEE explore (website)** Paper entitled "Study of the effect of change in Power on parameters of reactive protocol implemented using MATLAB Based True time Network Simulator for WANET" called "COMPUTATIONAL INTELLIGENCE AND COMMUNICATION NETWORKS CICN 2010"page 183-187 held at State University the R.G.P.V., Bhopal, India on 26-28 Nov 2010 organized by MIR Labs (USA) and RGPV Bhopal and co-sponsored by IEEE SMC Society, Mumbai Section.

7) Published Paper in 4th IEEE INTERNATIONAL CONFERENCE proceeding "Performance Evaluation of Reactive Routing Protocol using Parametric Decision based on ANFIS for MANET using Qualnet" in International Journal of Computing Science & Communication Technologies (IJCSCT, ISSN-0974-3375) at a ASIA PACIFIC INSTITUTE OF INFORMATION TECHNOLOGY SD INDIA,A Faridpur Road, G.T. Road Karnal Side, Panipat ,Haryana (India) called International Conference On Advance Computing And Communication Technologies ICACCT- 2010"sponsored by IEEE Delhi Section, IEEE Computer Society Chapter, Delhi Section & IETE Delhi Centre on 30th October 2010.

8) Submitted Paper entitled "Analysis and Comparison of Proactive and Reactive Routing Protocols for WANET in Qualnet" in JOURNAL OF ADVANCE RESEARCH N COMPUTER ENGINEERING: An International Journal , ISSN: 0974-4320 Vol. 5 no 2 (July-Dec 2011)

APPENDIX D2:

Presentations -Regional/National/International Conferences

1) Presented a Paper on "Wireless Ad-Hoc Network" at a Regional seminar called "Wireless Communication and Networking Technologies: WVnNT-2008" held at Institution of Engineering (India), Vasvik Bhavan, Vadodara on 21st December, 2008 organized by IETE Vadodara.

2) Presented a Paper on "Modeling and Network Simulation Tools for Wireless Network" at a Regional seminar called " Research Trends in Secured Communication Technologies: RTSCT-2009" held at Institution of Engineering(India),Vasvik Bhavan,Vadodara on 27th December, 2009 organized by IETE,ISA and IE Vadodara.

3) Presented a Paper "Analysis and Comparison of Proactive and Reactive Routing Protocols for WANET in Qualnet" at a State Level Paper Contest called "Wireless Technologies in Automation and Communication : WTAC-2010" held at Institution of Engineering(India),Vasvik Bhavan,Vadodara on 10th January, 2010 organized by IETE Vadodara.

4) Presented a Paper "Simulation of Reactive Protocol Using Matlab/Truetime" at a State Level Paper Contest called "Wireless Technologies in Automation and Communication : WTAC-2010" held at Institution of Engineering(India),Vasvik Bhavan,Vadodara on 10th January, 2010 organized by IETE Vadodara.

5) Presented a Paper "Analysis and Comparison of Proactive and Reactive Routing Protocols for WANET in Qualnet" at a National Level Paper Contest called "National Technical Paper Contest : NTPC-2010" held at Institution of Engineering(India),Vasvik Bhavan,Vadodara on 7th March, 2010 organized by IETE Vadodara.

6) Presented a Paper in INTERNATIONAL CONFERENCE "Performance Optimization of Reactive Routing Protocol Using Fuzzy Logic for MANET in Qualnet" at a WORLDCOMP 2010 World Congress In Computer Science, Computer Engineering And Applied Computing called " ICWN'10" held at LAS WEGAS,USA on 12-14th July, 2010.

7) Presented a Paper in INTERNATIONAL CONFERENCE "Signal Reach Computation for AODV Power Threshold in WANET employing Fuzzy Logic" at a WORLDCOMP 2010 World Congress In Computer Science, Computer Engineering And Applied Computing called "ICWN'10" held at LAS WEGAS, USA on 12-14th July, 2010.

8) Presented a Paper "FPGA Implementation of ANN based Reactive Routing Protocol for MANET" at a State Level Paper Contest called "Control, Microcomputer, Electronics and Communication (CMEC-2011)" held at Seminar hall, Electrical Engineering Department, Faculty of Technology & Engineering, Kalabhavan, Vadodara on 20th February, 2011 organized by IETE & EED,FTE,MSU Vadodara.

APPENDIX D3: Awards/Prizes

1) Second prize in the State Level Paper Contest called ""Wireless Technologies in Automation and Communication : WTAC-2010" held at Institution of Engineering(India),Vasvik Bhavan,Vadodara on 10th January, 2010 organized by IETE Vadodara. Paper entitled "Analysis and Comparison of Proactive and Reactive Routing Protocols for WANET in Qualnet".

2) Consolation prize in the State Level Paper Contest called ""Wireless Technologies in Automation and Communication : WTAC-2010" held at Institution of Engineering(India),Vasvik Bhavan,Vadodara on 10th January, 2010 organized by IETE Vadodara. Paper entitled "Simulation of Reactive Protocol Using Matlab/Truetime".

3) Second prize in the National Level Paper Contest called "National Technical Paper Contest: NTPC-2010" held at Institution of Engineering (India), Vasvik Bhavan, Vadodara on 7th March, 2010 organized by IETE Vadodara. Paper entitled
"Analysis and Comparison of Proactive and Reactive Routing Protocols for WANET in Qualnet".

4) First prize in the State Level Paper Contest called "Control, Microcomputer, Electronics and Communication (CMEC-2011)" held at Faculty of Technology & Engineering, MSU BARODA on 20th February, 2011 organized by IETE Vadodara and EED,FTE,M.S.University of Baroda, Vadodara. Paper entitled "FPGA Implementation of ANN based Reactive Routing Protocol for MANET".