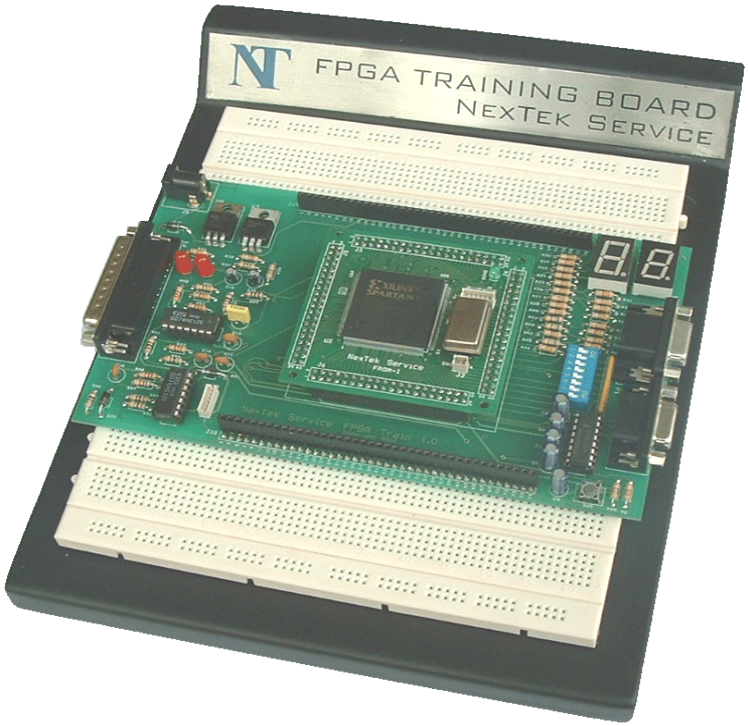
# Development Board For FPGA
# USER GUIDE
# MODEL NT-FTRAIN-02




**NexTek Service**

**Where Technology meets Reality**

**www.nextekservice.com**

# PRODUCT DETAILS

| | |
|---|---|
| Product Name | FPGA Training Kit |
| Model | NT-FTRAIN-02 |
| Configuration Code | FTRAIN0405 |
| Firmware Reference | NTFTRAIN-004-003 |
| Issue Date | 06/04/2005 |

# LIST OF TABLES

# LIST OF FIGURES

# Table of Contents

# CHAPTER 1

# INTRODUCTION

1. NT-FTRAIN-01 is a Field Programmable Gate Array Trainer. It is a development station for beginners as well as for professionals.
2. The learner of an FPGA (Field Programmable Gate Array) can use this development board to exercise the Digital Logic Designs (DLDs) keeping in view the hardware limitations of an FPGA used.
3. The professional can use this tool to develop his/her hardware and also to program the FROM modules in the production cycle.
4. The NT-FTRAIN is designed keeping in view the portability of FPGAs of different resources. Using this tool, FPGA can be programmed directly through the parallel port of a Personal Computer.

## General

5. The NexTek FPGA Trainer, NT-FTRAIN, has been developed with an aim of providing a combination of training as well as a development environment.
6. The NT-FTRAIN is a low cost FPGA development platform based on SPARTAN–II FPGA. Trainers for other families of FPGAs are also being developed at NexTek Service.
7. The kit has the following features:
   (a) The NT-FTRAIN has basic on–board resources to allow maximum target use.
   (b) Dual programming modes are provided i.e. Parallel port interface and a JTAG programmer.
   (c) Serial interface is given on NT-FTRAIN to communicate with a PC using RS232 interface.
   (d) Two Seven-segment displays are provided for a host of application that range from debugging to state monitoring.
   (e) VGA port is provided on NT-FTRAIN for interface with a monitor/LCD.
   (f) The FPGA is given on NT-FTRAIN in the form of a small PCB called FROM-II.
   (g) Configuration PROM (Programmable Read Only Memory), 16 MHz Crystal, Voltage Regulator, JTAG interfaces, and power connectors are built-in on FROM-II.
8. The FROM-II is an ideal module for the developers who do not want to get into the hassle of defining and designing components around the FPGA.
9. It takes a 5VDC power input and relieves the designer from designing surface mount boards and afterwards finding surface mount soldering sources.
10. A power adapter is given with NT-FTRAIN. On-board regulators are used to feed the desired power to the FPGA.
11. A small cable is given with the package, which is used to connect the NT-FTRAIN with FROM-II for the programming of an FPGA.

12. SPARTAN –II FPGA contains maximum of 150K user gates, and can operate at clock rates beyond 100MHz. The multi-volt I/O allows 2.5V, 3.3V or 5.0V to be used as VCCIO.

## BLOCK DIAGRAM

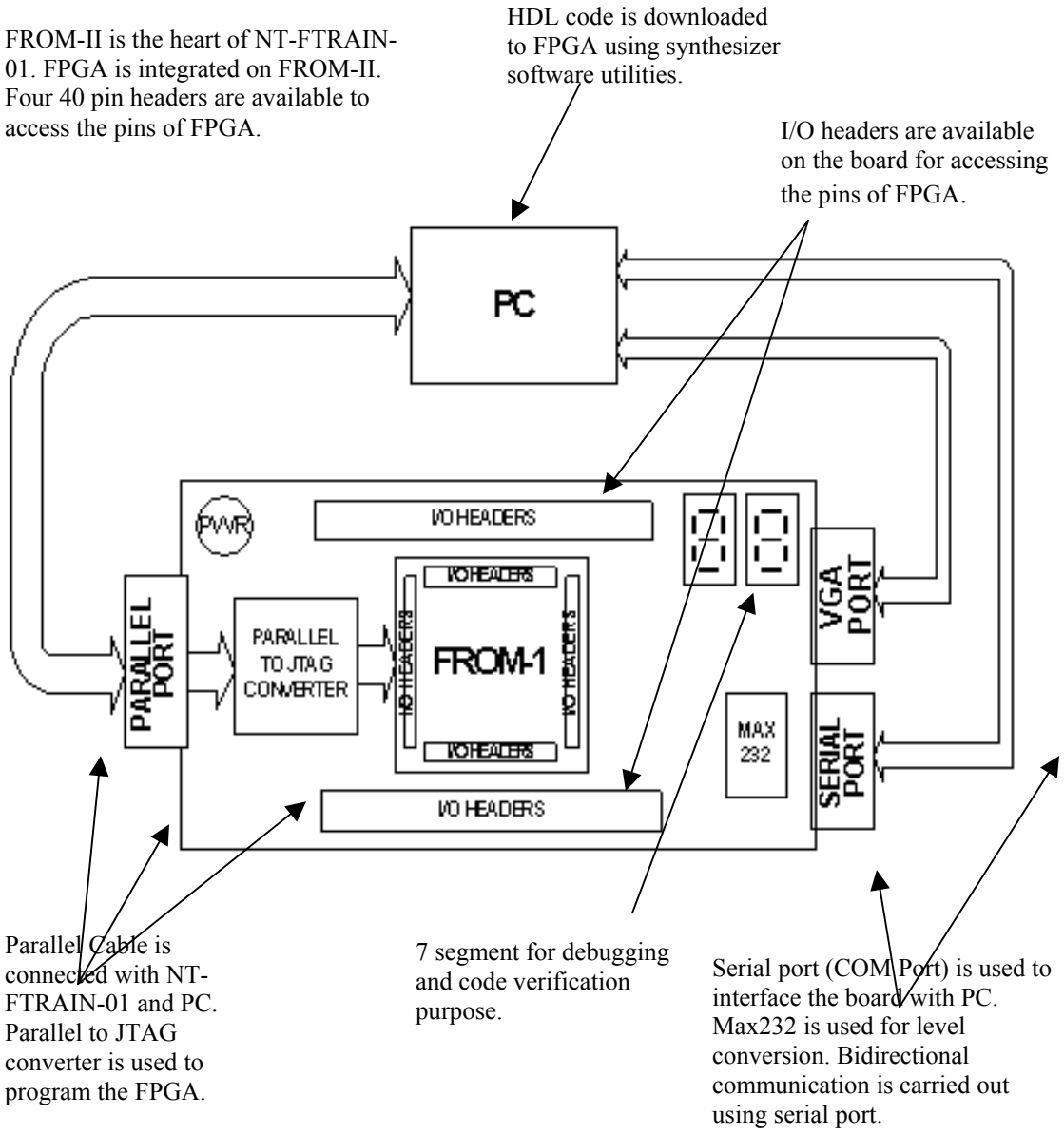



*Figure 1: NT-FTRAIN Block diagram*

# CHAPTER 2

# FEATURES

## Benefits of Board-Based Training

1. Learn language skills using a realistic project-based design rather than isolated code fragments.
2. Complete the design flow by implementing your project in hardware using our unique development board.
3. Take the board away with you, together with a design software package, at the end project.
4. Use the board to customize the lab project; create your own designs and experiment with the language & methodology.

## New NT-FTRAIN-02 Features

5. Straight-forward, clear-cut board design – easy to understand.
6. Ready to use “out of the box” – no additional logic required.
7. Expandable – extension port for interfacing with other hardware.
8. Flexible – additional logic on board for further development beyond the training course lab.
9. Generous device capacity – implement designs in excess of 150K gates
10. No CPLD/FPGA experience required – tool scripts provided for easy implementation and configuration.


*Figure 1: FPGA Training Board*

## NT-FTRAIN Package

11. The NT-FTRAIN-01 Development Board
12. An elegant skin finish, desktop with Brett Boards.
13. 9V 1A Adapter.
14. Parallel Cable (for downloading using parallel port).
15. Download cable (for downloading using JTAG programmer)
16. Serial Cable for interface with a personal computer.
17. Implementation software package
18. Operating Manual, datasheets, reference material.


*Figure 2: NT-FTRAIN-01Package*

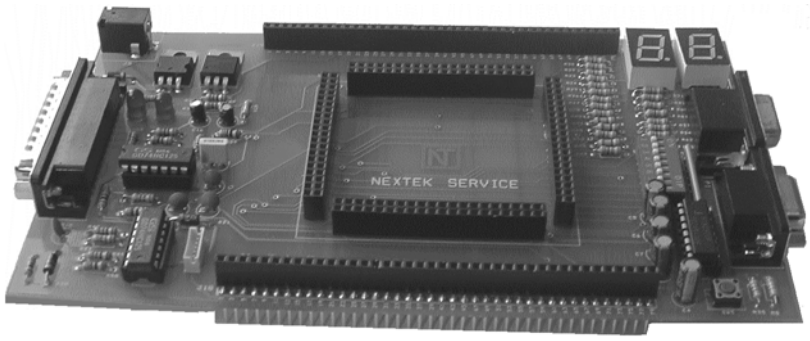



*Figure 3: Training Card without FROM*

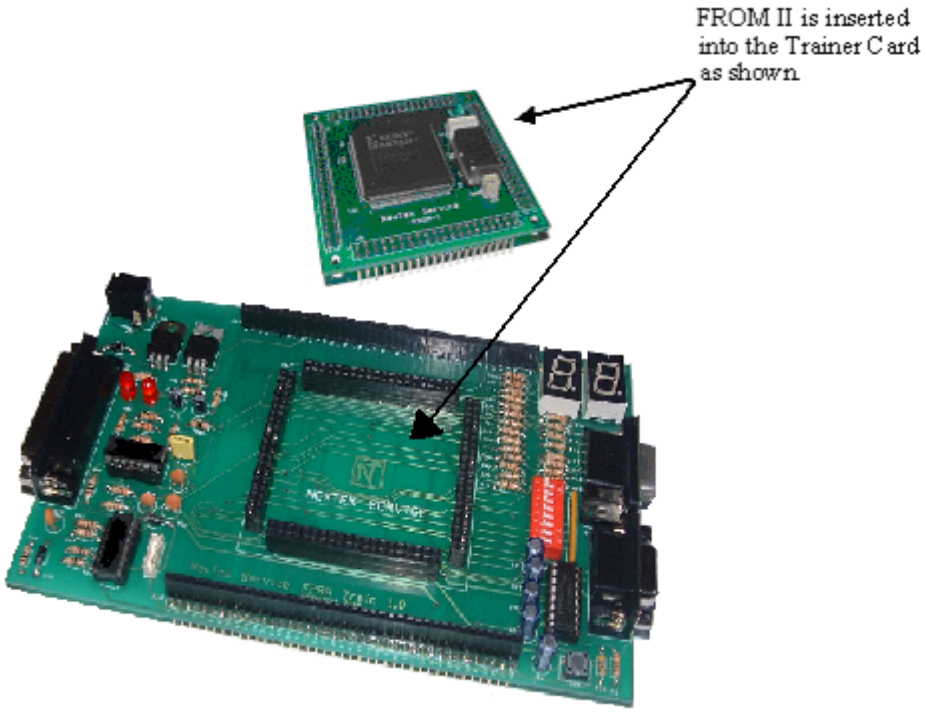



*Figure 4: Training Card with FROM*

## CHAPTER 3

# TECHNICAL DESCRIPTION

### POWER FEED

1. 9 Volt adapter is supplied with the package which is used for the power feed to the board. The pin of the adapter is connected with the J5 power jack.
2. The voltage is then regulated to 5 volts and 3.3 volts for the supply to FROM-II and the other components on the board.
3. Two LEDs D17 and D19 are the indication of 5V and 3.3V respectively.


*Figure 1: Power Adapter*

### PARALLEL PORT

4. Parallel port is used for the programming of an FPGA. Parallel cable is connected to the NT-FTRAIN and a PC.

*Figure 2: Parallel Cable and Connections*

**SERIAL PORT**

5. Standard DB9 connector is used to provide an interface between NT-FTRAIN and a PC.
6. The serial communication can be established between NT-FTRAIN and a PC using HyperTerminal.
7. Max 232 is used for providing RS232 and TTL levels. It is connected directly to the I/O pins of an FPGA.
8. The pinout of Serial Port with FPGA and its schematic is given in Appendix B.

*Figure 3: Serial Port Cable and Max 232 Connections*

**SEVEN SEGMENT DISPLAY**

**9.** The results of different labs can be shown on a 7-segment displays available on NT-FTRAIN.
**10.** It is a Common Cathode display. It can be used for debugging purposes also.
**11.** The user can single step through the code and keep on checking the code while displaying the results of each statements on 7 segment displays.
12. The Pinout of 7 Segment Display with FPGA and its schematic is given in Appendix C. The datasheet is given in the “datasheets” folder of Product CD.

**HEADERS**

13. General-purpose input output pins (I/Os) can be accessed through headers available on the NT-FTRAIN.
14. Total of 142 pins are available for general-purpose input output. These pins can be declared as inputs or outputs in the code.

The description of the pins and Schematic is given in Appendix A and D respectively.

Headers for accessing the I/O pins of FPGA.

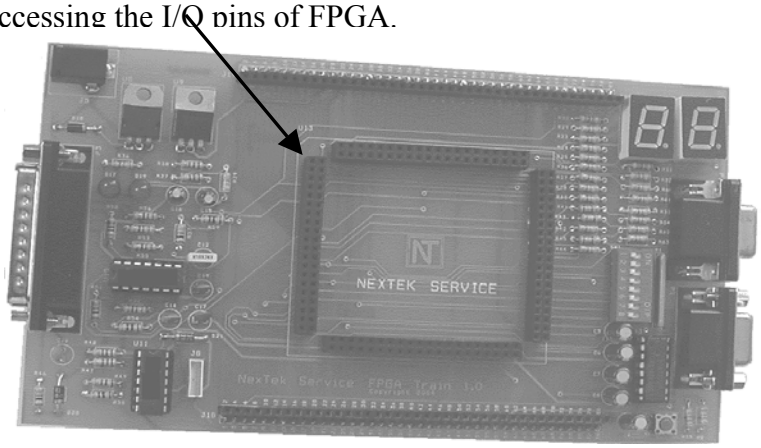



*Figure 4: I/O Headers*

**PROGRAMMING CABLE**

15. 6 pins cable is given with package for the programming of FROM-II. The programming signals from the parallel port are routed to FROM-II by this cable.

**PROGRAMMING CABLE CONNECTION**

16. The strips pasted on both ends of the cable can verify the orientation of programming cable.


*Figure 5: Programming cable orientation.*

## SYSTEM RESET

**17.** A switch given on NT-FTRAIN resets FPGA. This switch is connected to the GCK0 (pin 80) of FPGA.

## ORIENTATION OF FROM II IN TRAINER CARD

18. The orientation of FROM II in NT-FTRAIN-01 is shown below. After placing the FROM II at its proper orientation, remove the crystal either from FROM II or NT-FTRAIN-01 Trainer Card.

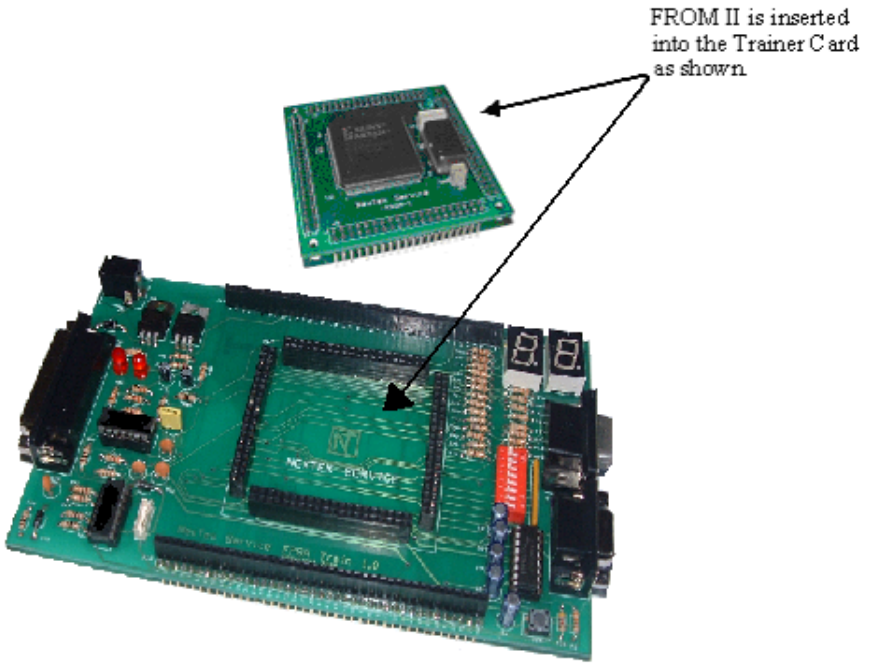
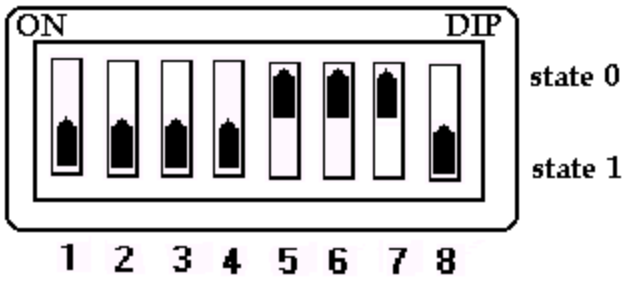



*Figure 6: Orientation of FROM-II In NT-FTRAIN-01*

## DIP SWITCH:

The FPGA TRAINING BOARD comes by default at low state. However, you may re-configure your unit multiple times upon your requirements. A person who is familiar with data communications products and terminology best undertakes such a task. You will observe that there is an 8-pin red Dipswitch assembly, with individual sliders numbered 1 to 8. The position of all these is shown in the diagram on the summary pages at the end of this section.




**CAUTION!**

**This unit contains device, which are static sensitive. Great care should be taken when adjusting switches and monitoring to avoid touching any connections directly. Whenever possible, anti-static precautions should be taken, such as the use of an earthed wrist-strap and anti-static mat.**

# CHAPTER 4

# Project Navigator

This chapter contains the following sections:

1. About Project Navigator
2. Starting Project Navigator
3. Project Navigator Interface
4. Snapshots and Archives

**About Project Navigator**

1. Project Navigator is the primary user interface for ISE 4.x. You create your FPGA or CPLD design using a suite of tools accessible from Project Navigator. Each step of the design process, from design entry to downloading the design to the chip, is managed from Project Navigator as part of a project.

**Project Navigator Processes**

2. Project Navigator integrates the following processes.
   - Design Entry
   - Constraint Entry
   - Synthesis
   - Simulation
   - Implementation
   - Device Programming

**Starting Project Navigator**

3. To start Project Navigator on personal computers:
Double-click the Project Navigator icon on your desktop.
OR
Click **Start → Programs → Xilinx ISE Series 4.x →Project Navigator**.

**Note** Your startup menu path is created during installation, and may differ from the path shown above.

4. The first time you open Project Navigator all of its windows are empty.
After that, Project Navigator opens with the last project you worked on if you selected **Edit → Preferences → General → Window Settings → Always Open Last Project**.

**Project Navigator Interface**

5. The Project Navigator interface contains the following:

- Project Navigator Main Window
- Sources In Project Window
- Sources in Project Windows tab
- Project Workspace
- HDL Editor Workspace

**Project Navigator Main Window**

6. The following figure shows the windows, toolbars, workspaces, and other objects in the Project Navigator main window.

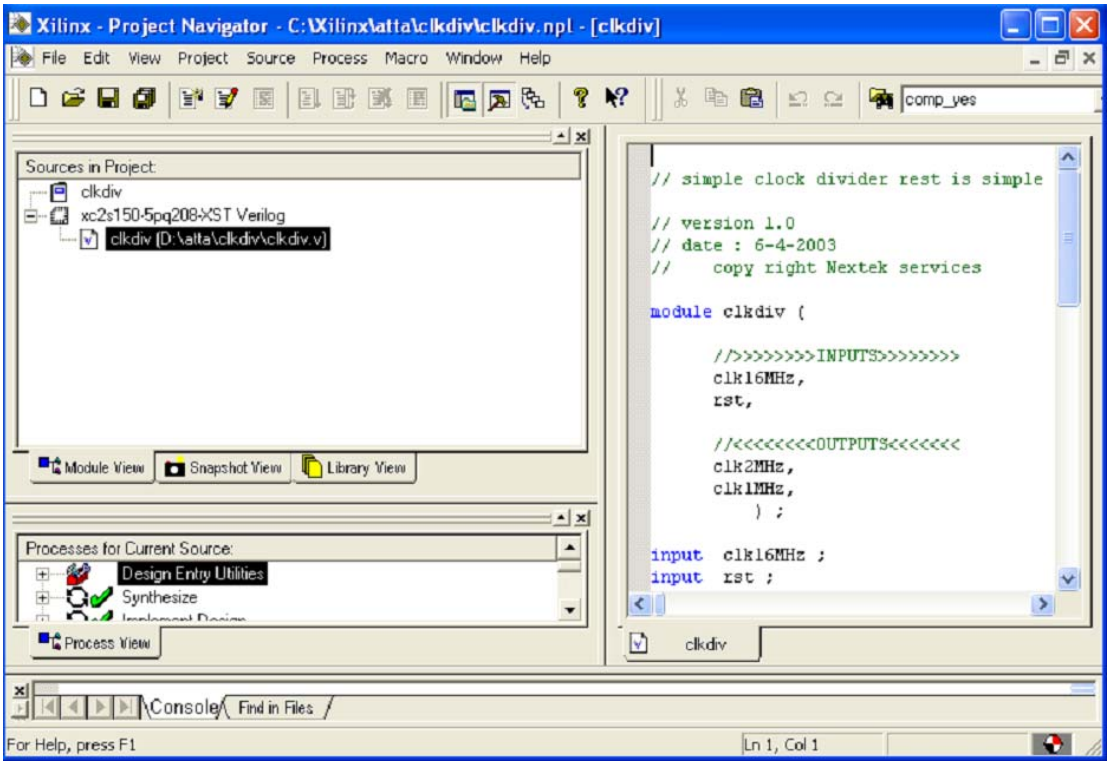



*Figure 1:Project Navigator Main Window*

7. The Project Navigator main window contains the following:

- ***Title Bar***

  The Title bar displays the name of the application and the path of the current project.

- ***Menu Bar***

  The Menu bar allows you to access the Project Navigator commands.

- ***Toolbars***

  The toolbars provide convenient access to frequently used commands.

**Sources In Project Window**

**8.** The Sources in Project window lists all the design files associated with a project. A source is any element that contains information about a design definition.

**Sources in Project Window Tabs**

9. Tabs at the bottom of the Sources in Project window allow you to select three different views of the source data:

- Module View
- Snapshot View
- Library View

**Project Workspace**

10. The Project Workspace consists of the Sources in Project window and the Processes for Current Source window. These windows are grouped together for viewing purposes. Click **View** → **Project Workspace** to toggle the display of these two windows as one item.

**HDL Editor Workspace**

**11.** The HDL Editor workspace is the main text editing area for HDL code. It is the only Project Navigator window that cannot be hidden or undocked.

**Snapshots and Archives**

12. To save a specific revision of your project, you can archive it or take a snapshot of it. When you archive a project, you create a ZIP file for that version and place it in a specified directory.

13. To archive a project, click **Project** →**Archive** in Project Navigator. To open an archive or see the files in it, unzip it.

14. When you take a snapshot of a project, the snapshot becomes part of the project. It is accessible from the Snapshot View of the Sources in Project window.

**Taking a Snapshot**

15. To take a snapshot of the current version of your project:

- Click **Project** → **Take Snapshot**.
- Enter a name and comments in the Take a Snapshot of the Project dialog box.

**Deleting a Snapshot**

16. To delete a snapshot:

- Go to the Sources in Project window.
- Click the Snapshot View tab.
- Click a snapshot name.
- Click **Source** → **Remove** from the Project Navigator.

# CHAPTER 5

# PROJECTS

This chapter contains the following sections:
1. About Projects
2. Creating a Project
3. Creating a New Source
4. Source Types

**About Projects**

1. ISE 4.x organizes and tracks your FPGA or CPLD design as a *project*. This section discusses some of the key concepts and features of projects.

- **Project**
  A project is a collection of all files necessary to create and download your design to the selected device.

- **Project Properties**
  Each project has a directory, device family, device, and design flow associated with it as *project properties*.
- **Sources**
  A *source* is any element that contains information about a design, such as HDL files, state diagrams, schematics, documentation files, simulation models, and test files. You create and add the sources to your project.
- **Process**
  A *process* is an action you can perform on a source to test and implement your design.

**Creating a Project**

2. Creating a project consists of the following:

- Setting Project Properties
- Specifying a Project Name and Directory
- Selecting a Device and Design Flow

**Setting Project Properties**

3. Specify the following project properties when you create a new project:

- ***Project name***
  Give your project a unique name.
- ***Project directory***

Specify a unique directory to store the project's source files, intermediate data files, and resulting files.

- ***Device family***
  Target your design for a specific Xilinx device family (architecture). Vertex, Spartan, XC9500 are examples of device families.
- ***Device***
  Target your design for a specific device within the selected device family. An example of a device is *S05 PC844-4*, where *S05* designates a specific Spartan device, *PC844* specifies the package, and *-4* indicates the speed grade.
- ***Design flow***
  Specify a design flow, either EDIF or a synthesis tool. FPGA Express and XST are the synthesis tools. VHDL, Verilog, and ABEL versions are available for these tools.

**Specifying a Project Name and Directory**

4. Specify a project name and directory when you create a new project. For each project, designate a separate, unique directory containing only one project file (*project_name*.npl).

**Selecting a Device and Design Flow**

5. The first time you create a new project, a default device and design flow appear in the New Project dialog box. For subsequent projects, the device and design flow you used for your last project are used as the default for the new project.

**Selecting a Device**

6. To select or change a device:
   - Open the project.
   - Click the Device and Design Flow line in the Sources in Project window.
   - Click **Source → Properties**.
   - Click the scroll button on the right side of the Device Family Value field to display the device family list.
   - Scroll through the device family list and select the device by clicking its name.

**Creating a New Source**

7. To create a new source file:
   - Open the project.
   - Click **Project → New Source**.
   - Select the source type you want to create from the list.
   - Enter a name for the new source file in the File Name field.
   - Check the Add to Project box to add this source automatically after it is created.
   - Click **Next**.

- Click **Next** in the Define Sources in Project window.
- Click **Finish** in the New Source Information window.

**Adding an Existing Source**

- Open a project.
- Click a source in the Sources in Project window.
- Click **Project → Add Source**.
- Use the Add Existing Sources in Project dialog box to browse through your directories and select the source you want to add.
- Click open.
- After selecting the file, the Choose Source Type dialog box may open.
- When you click **Open** to select a file in the Add Existing Sources in Project dialog box or **OK** in the Choose Source type dialog box, the file appears in the Sources in Project window for the current project.

**Source Types**

8. Projects can include the source types listed in the following table.

| Source Type | File Extension |
|---|---|
| Verilog module | .v |
| ABEL-HDL logic description | .abl |
| ABEL-HDL test vectors | .abv (or .abl) |
| COREgen IP | .xco |
| LogiBLOX module | .mod |
| Testbench Waveform | .tbw |
| Block Memory Map file | .bmm |
| Executable CPU code | .elf |
| EDIF Source file | .edn, .edf, .edif, .sedif |
| Project file | .npl |
| User document (such as a | .txt, .wri, .doc, .xls, .hlp |
| Schematic | .sch |
| State diagram | .dia |
| VHDL module | .vhd |
| VHDL package | .vhd |
| VHDL testbench | .vhd |
| Waveform stimulus | .wdl |

*Table 1: Source Types*

# CHAPTER 6

# DESIGN FLOW

This chapter contains the following sections:
1) About Design Flow
2) Design Entry
3) Constraint Types and Entry Tools
4) Synthesis
5) Simulation
6) Implementation
7) Device Programming

**About Design Flow**

1. Design flow is a multi-step, iterative process that includes:
    - Design Entry
    - Constraint Entry
    - Synthesis
    - Simulation
    - Implementation
    - Device Programming

**Design Entry**

2. You can create your design using HDL code, Intellectual Property such as CORE Generator, schematics, and state diagrams. You can create new sources or add existing sources to your project. ISE 4.x includes the following design entry tools, all of which are accessible from Project Navigator:

    - HDL Editor
    - StateCAD State Machine Editor
    - Engineering Capture System (ECS)
    - CORE Generator
    - LogiBLOX

**Constraint Types and Entry Tools**

3. The following constraints are available for use with your FPGA and CPLD designs:
    - Timing Constraints
    - Placement Constraints
    - Grouping Constraints
    - Mapping Directives
    - Routing Directives
    - Modular Design Constraints

- Synthesis Constraints
- Fitter Directives
- Initialization Directives
- DLL and DCM Constraints
- Logical and Physical Constraints

**Caution** Not all constraints are available for all devices, nor can they be entered with every tool or method.

**Constraint Entry Tools**

4. Constraints can be entered at various stages throughout the entire design process, using a variety of tools. Constraint entry methods and tools include the Xilinx Constraints Editor, UCF files, FPGA Express, and XST Constraint files, as well as several others.

**Synthesis**

5. After your design has been successfully analyzed, the next step is to translate the design into gates and optimize it for the target architecture. This is the synthesis phase. Two synthesis tools are generally used:

- XST from Xilinx
- FPGA Express from Synopsis

6. You choose the synthesis tool in the Design Flow option when you create a project.

**XST from Xilinx**

7. XST (Xilinx Synthesis Technology) is a Xilinx tool that synthesizes HDL designs to create an NGC file. The Project Navigator invokes
XST processing when you select a source and then select a synthesis process for a project that has the XST synthesis tool associated with it.

8. An XST flow project can contain either VHDL (XST VHDL) or Verilog (XST Verilog) modules, but not a mix of both. A functional VHDL model (XST VHDL) or Verilog model (XST Verilog) is created for schematics prior to synthesis. Process properties can be set to control XST synthesis.

**FPGA Express from Synopsis**

9. FPGA Express from Synopsys, Inc., can synthesize VHDL, Verilog, or mixed HDL designs to create EDIF netlists.

10. A functional VHDL model (FPGA Express VHDL) or Verilog module (FPGA Express Verilog) is created for schematics prior to synthesis.

**Simulation**

11. Simulation verifies the operation of your design before you implement it as hardware.

**Simulation Points**

12. Several simulation points are available to test your design:
    - Behavioral simulation to check the logic prior to synthesis
    - Functional Simulation to check the logic post-synthesis
    - Post-MAP simulation to verify behavior post-map
    - Post-route Simulation to verify that the design meets the timing requirements you set for your design in the targeted device.

**Implementation**

13. The implementation stage consists of taking the synthesized netlists through translation, mapping, and place and route.
14. To check your design as it is implemented, reports are available for each stage in the implementation process.

**Device Programming**

15. When your design meets all your requirements, you can create a programming file that can be downloaded to the target device.
16. Use the following tools to program a device:
    - iMPACT
    - PROM File Formatter

# CHAPTER 7

# Device Programming

This chapter contains the following sections:
1. Creating FPGA Programming Files
2. Generating CPLD Programming Files
3. Device Programming Tools

**Creating FPGA Programming Files**

1. After the design has been completely routed, you must configure the device so that it can execute the desired function. Xilinx's bitstream generation program, BitGen, takes a fully routed NCD (Native Circuit Description) file as its input and produces a configuration bitstream—a binary file with a .bit extension.
2. The BIT file contains all of the configuration information from the NCD file defining the internal logic and interconnections of the FPGA, plus device-specific information from other files associated with the target device. The binary data in the BIT file can then be downloaded into the FPGA's memory cells, or it can be used to create a PROM file.
3. To create a configuration bitstream file:
   - Click the top-level source for the project in the Sources in Project window.
   - Click **Generate Programming File** in the Processes for Current Source window.
   - Click **Process** → **Run** in the Project Navigator menu.
   - To view the Programming File Report in the report window, double-click **View Programming File Generation Report** in the Processes for Current Source window.

**Launching Programming Tools**

4. To launch a programming tool:
   - Click the top-level source file in the Sources in Project window.
   - Double-click the programming tool name in the Processes for Current Source window.

**Device Programming Tools**

5. After creating the programming file, use one of the following programming tools to configure your device:
   - PROM File Formatter
   - iMPACT

**PROM File Formatter**

6. An FPGA or daisy chain of FPGAs can be configured from serial or parallel PROMs.

7. The PROM File Formatter can create MCS, EXO, or TEK style files. The files are read by a PROM programmer that turns the image into a PROM.
8. A HEX file can also be used to configure an FPGA or a daisy chain of FPGAs through a microprocessor.
9. The PROM File Formatter is available for FPGA designs only. The PROM File Formatter provides a graphical user interface that allows you to:

- Format BIT files into a PROM file compatible with Xilinx and third-party PROM programmers
- Concatenate multiple bitstreams into a single PROM file for daisy chain applications
- Store several applications in the same PROM file

**iMPACT**

10. The iMPACT configuration tool, a command line and GUI based tool, allows you to configure your PLD designs using Boundary-Scan, Slave Serial, and Select MAP configuration modes. iMPACT supports both the Parallel (JTAG) and MultiLINX cables.

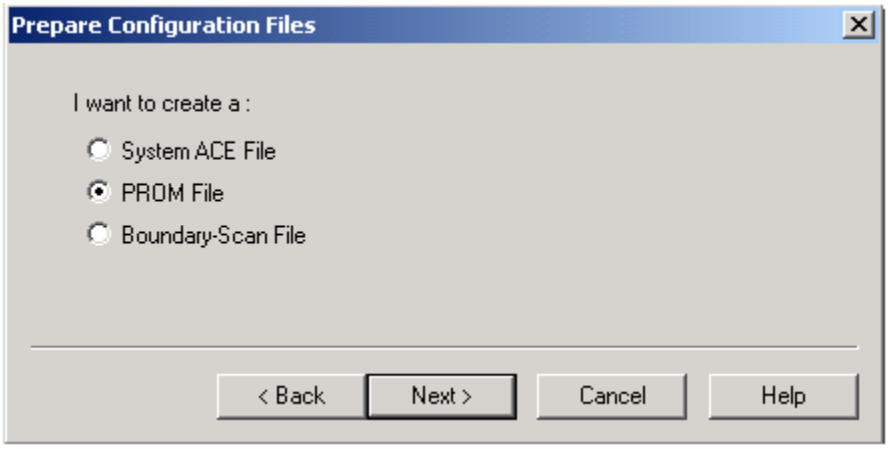
iMPACT also allows you to:

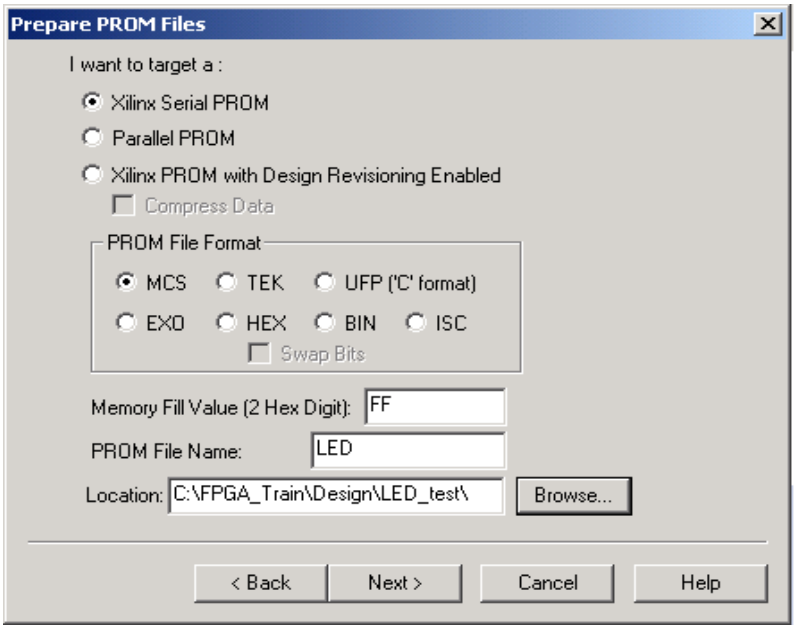
- ReadBack and Verify design configuration data
- Run Boundary-Scan TAP debug operations
- Create SVF and STAPL Files

# CHAPTER 8

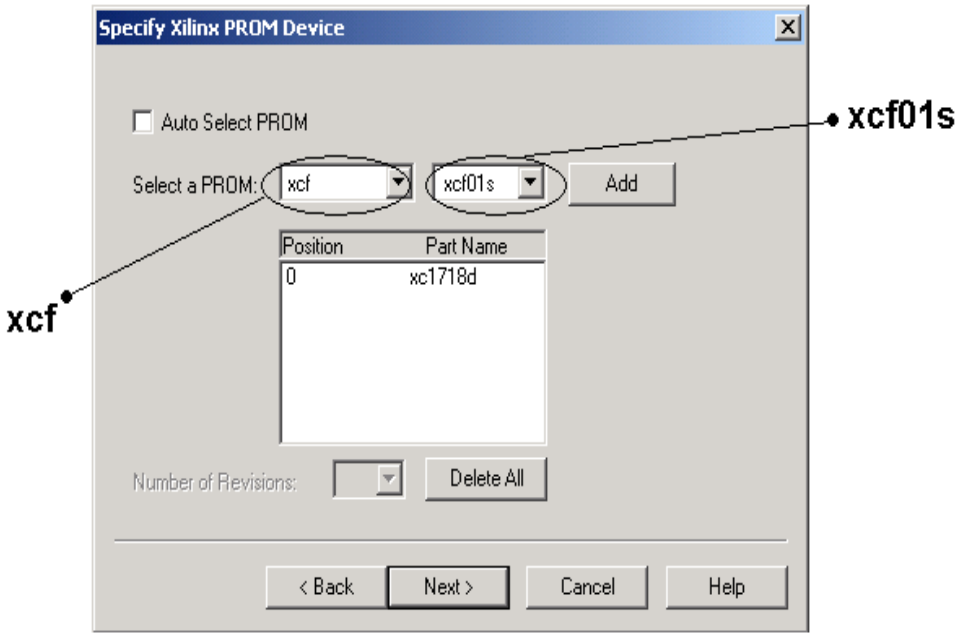
# Selection and Configuration

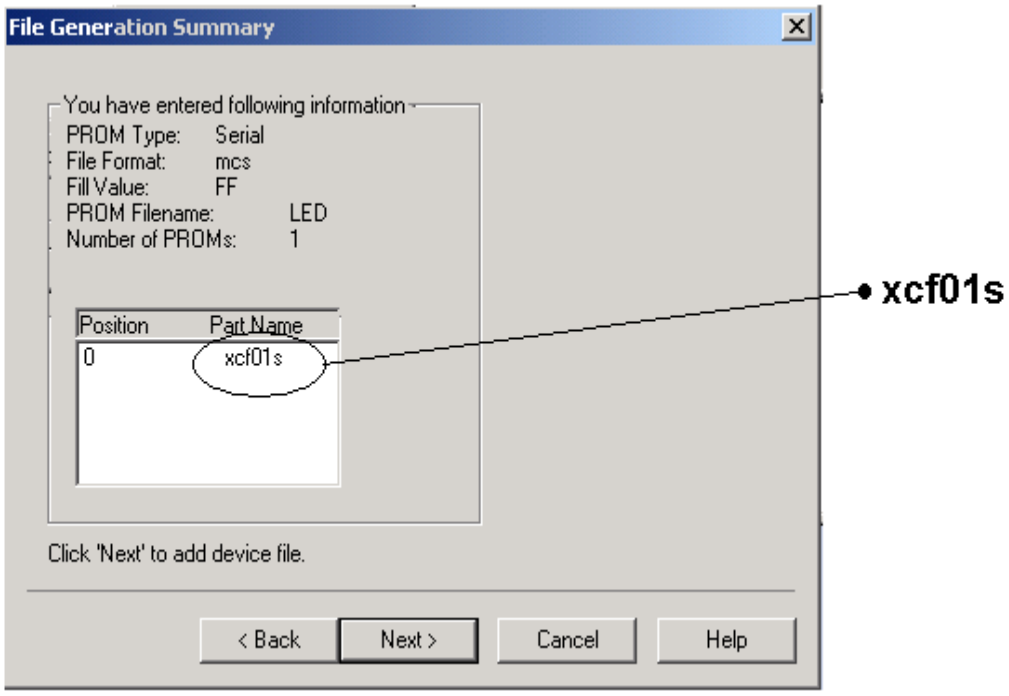
1. PROM file is generated with *.mcs extension.
2. Following dialog boxes show the complete procedure of generating a PROM file.



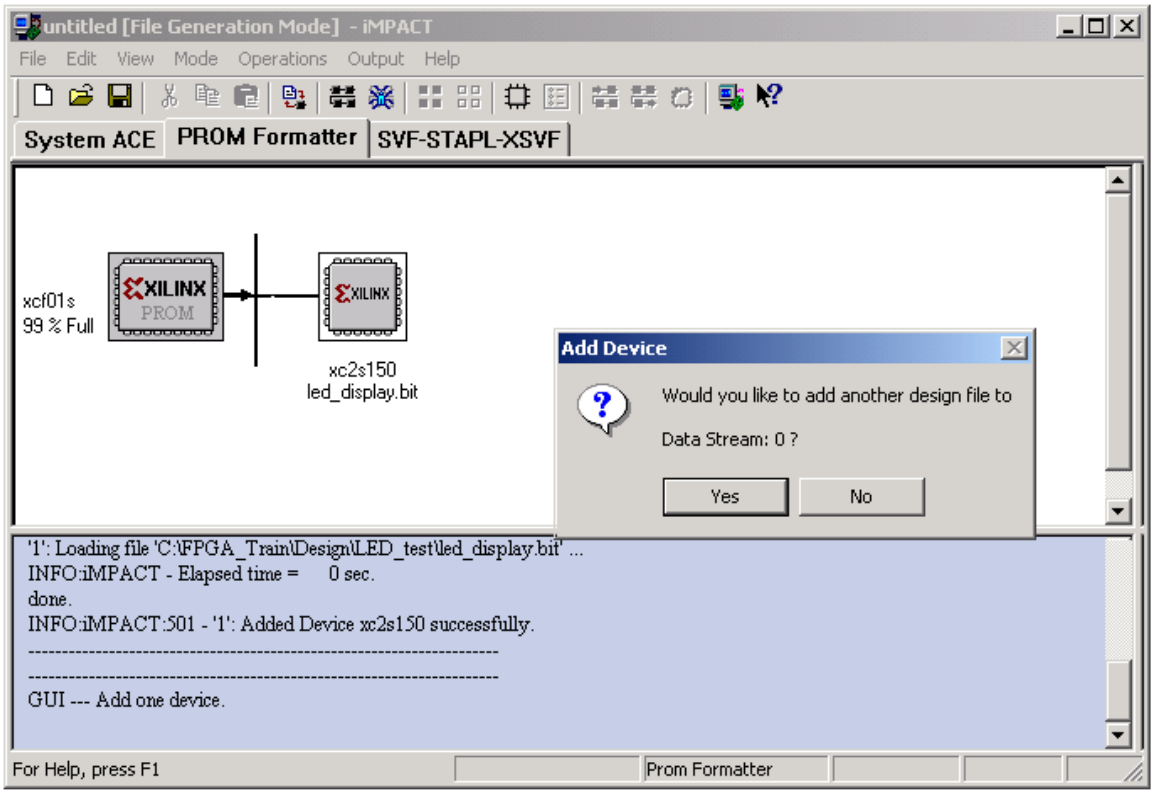

3. In PROM File Name give the name of the PROM file.
4. In Location give the path where the PROM file should be created.
5. Important point in generating a PROM file is to know the part number and family of PROM used with FPGA.
6. The PROM used with FROM is xcf01s.
7. Select xcf as a PROM.
8. Select the part number xcf01s as shown in the following dialog box and click Add.

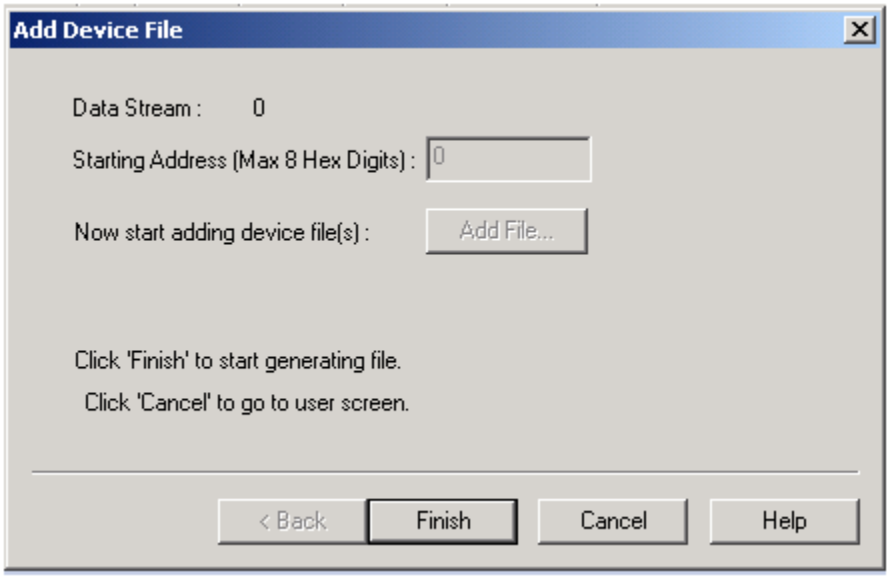


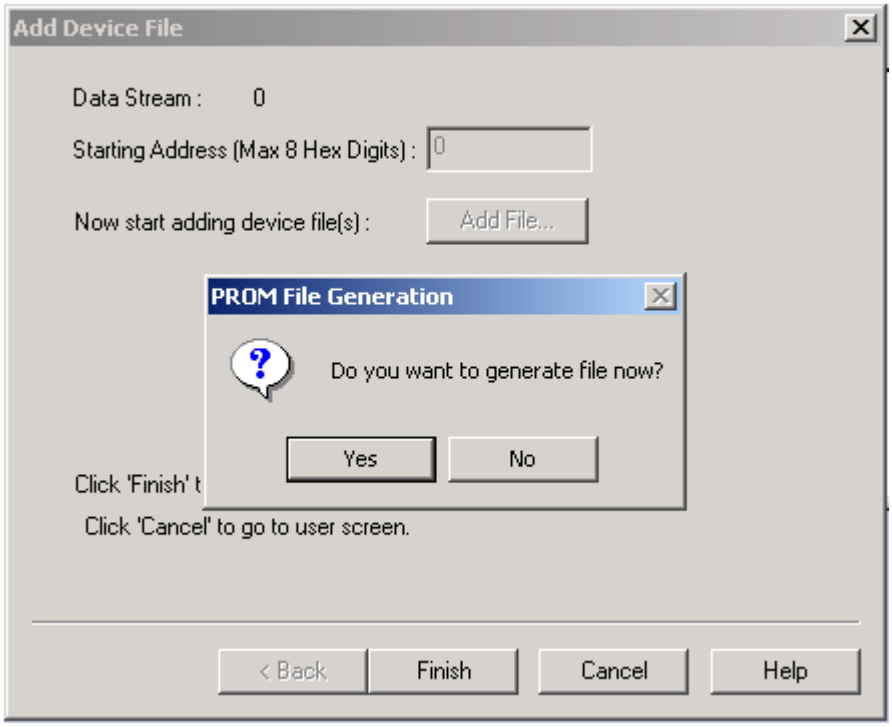




9. Now click Add File…

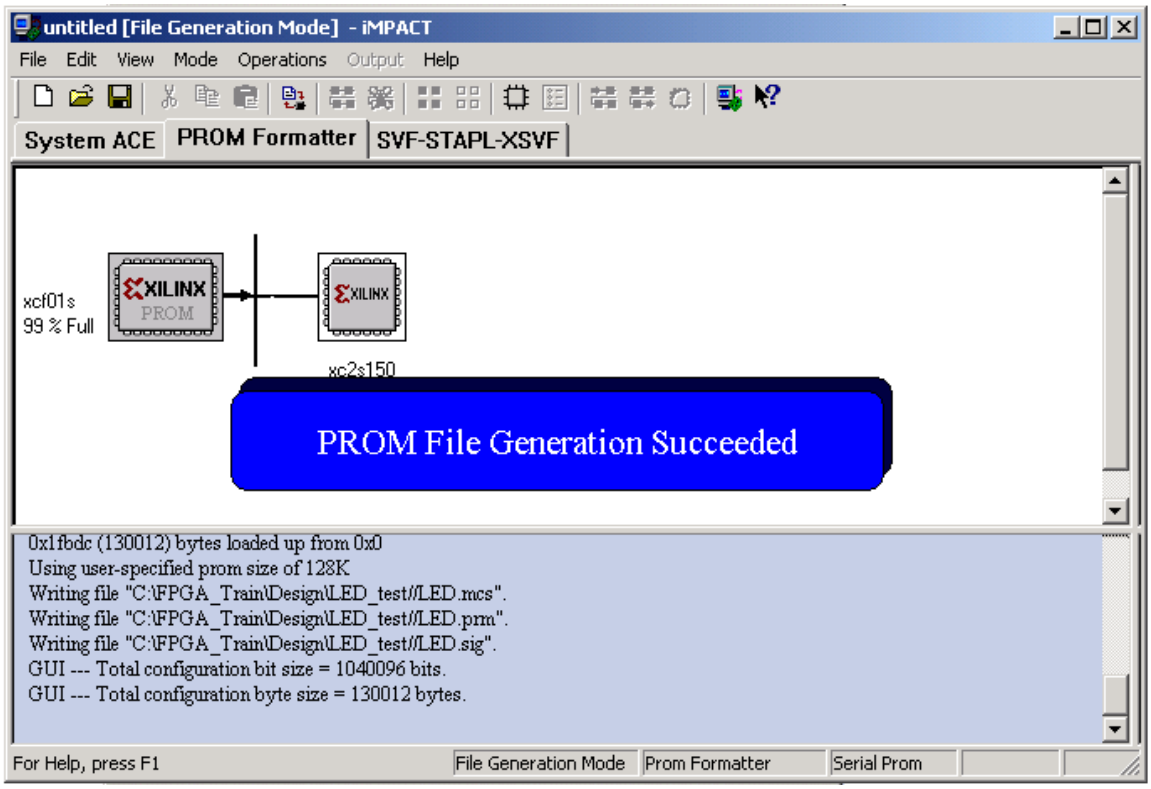

FPGA is detected and is displayed along with its part number.
10. Select No as we have only one device (FPGA) to configure.






11. Click yes to generate the PROM file and then click finish.




12. The above dialog box shows the successful generation of PROM file.
13. Close the window.
14. Now double click the last step of “Generate Programming File” i.e. “configure device (iMPACT)” in project navigator window.
15. Choose the options shown below.

# iMPACT Configuration:

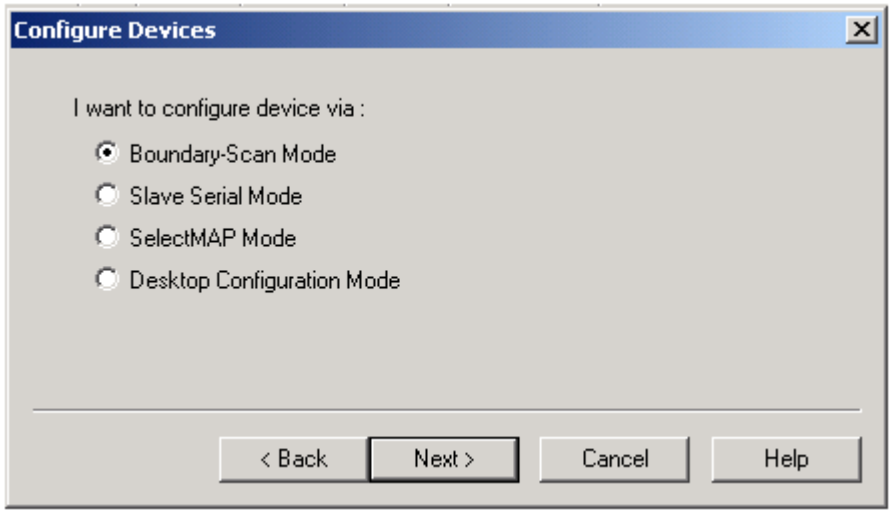
The iMPACT configuration tool, a command line and GUI based tool, allows you to configure your PLD designs using Boundary-Scan, Slave Serial, and Select MAP configuration modes. iMPACT supports both the Parallel (JTAG) and MultiLINX cables.

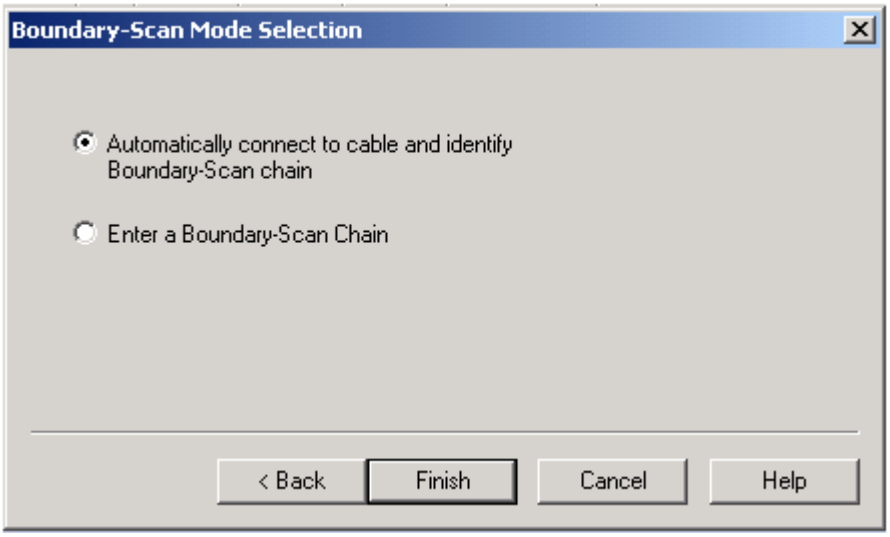
iMPACT also allows you to:

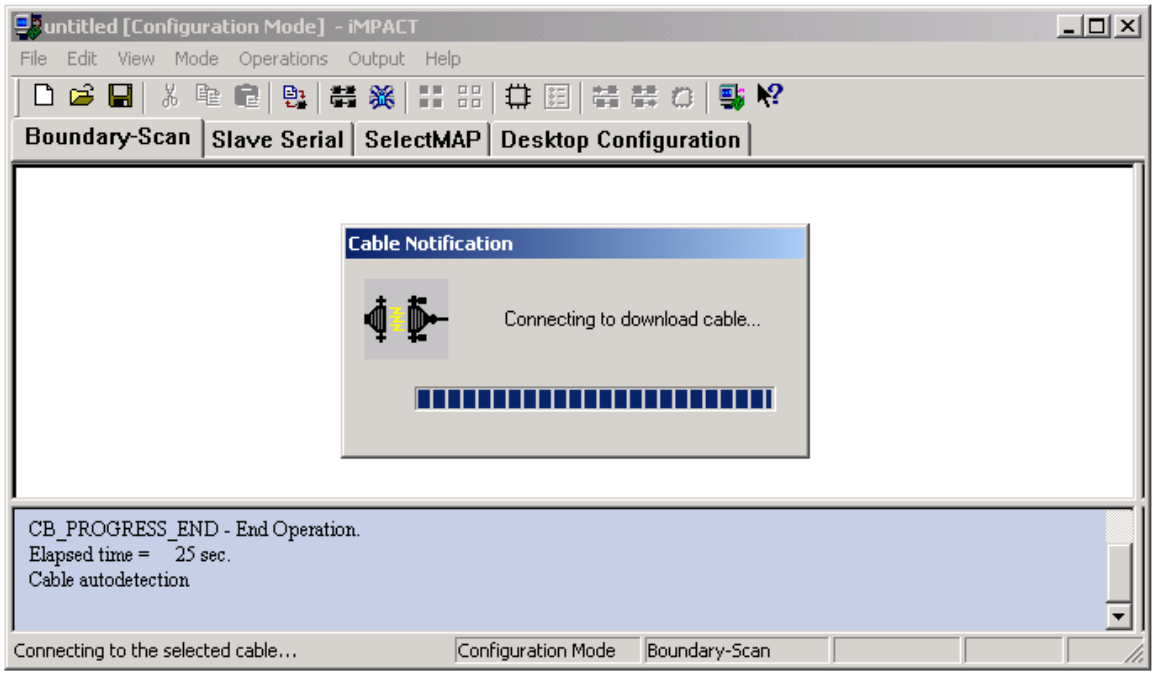
- ReadBack and Verify design configuration data
- Run Boundary-Scan TAP debug operations
- Create SVF and STAPL Files

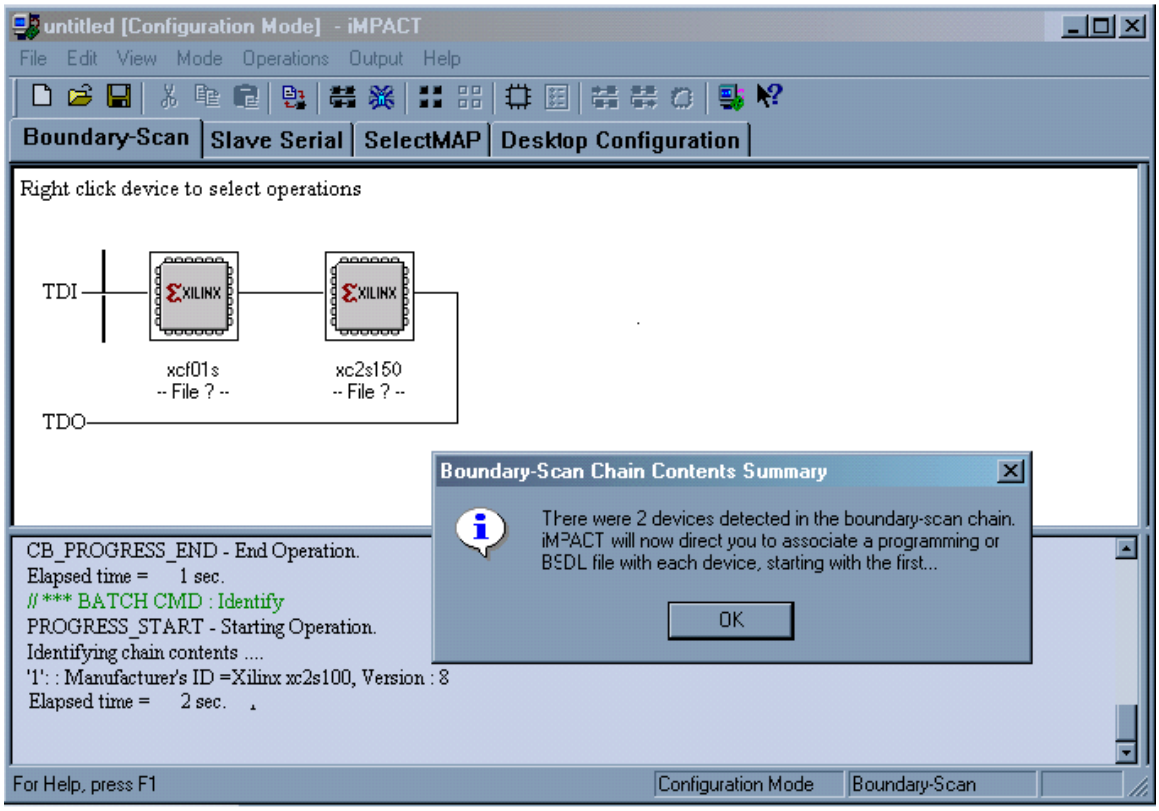
iMPACT configuration is described by hierarchical GUI as shown below.

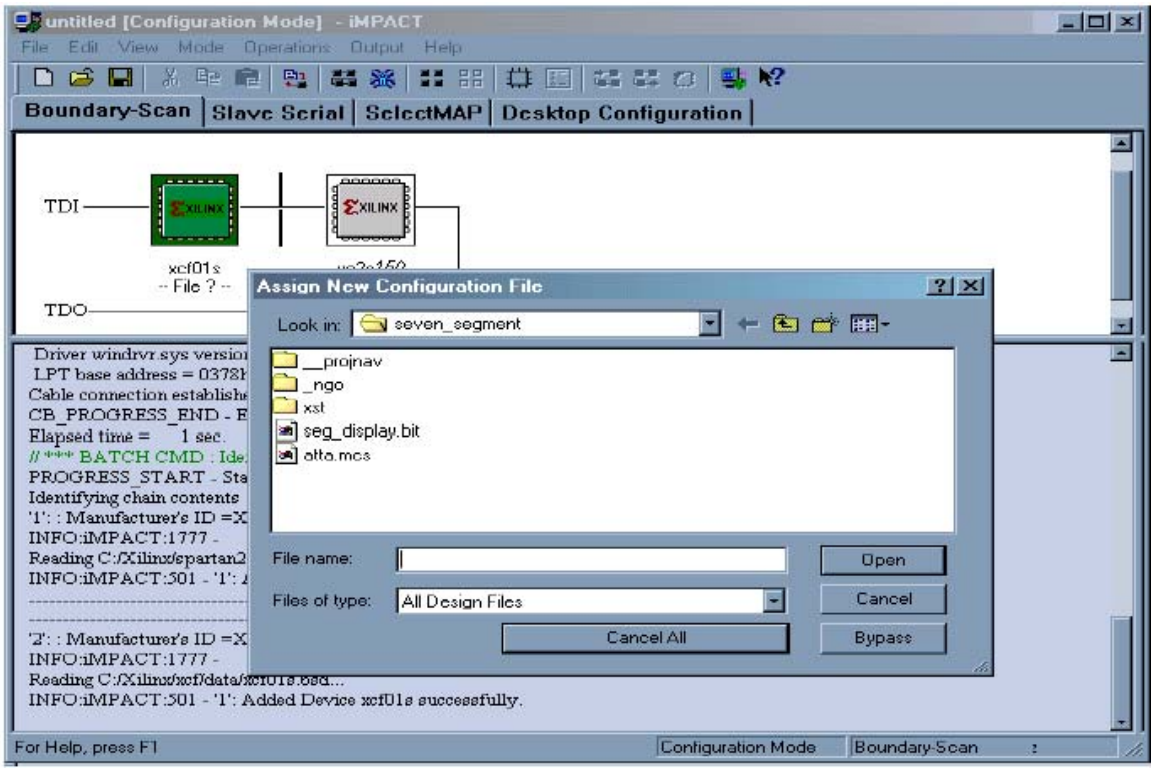
Now double click the last step of “Generate Programming File” i.e. “configure
device (iMPACT)” in project navigator window.
Choose the options shown below.



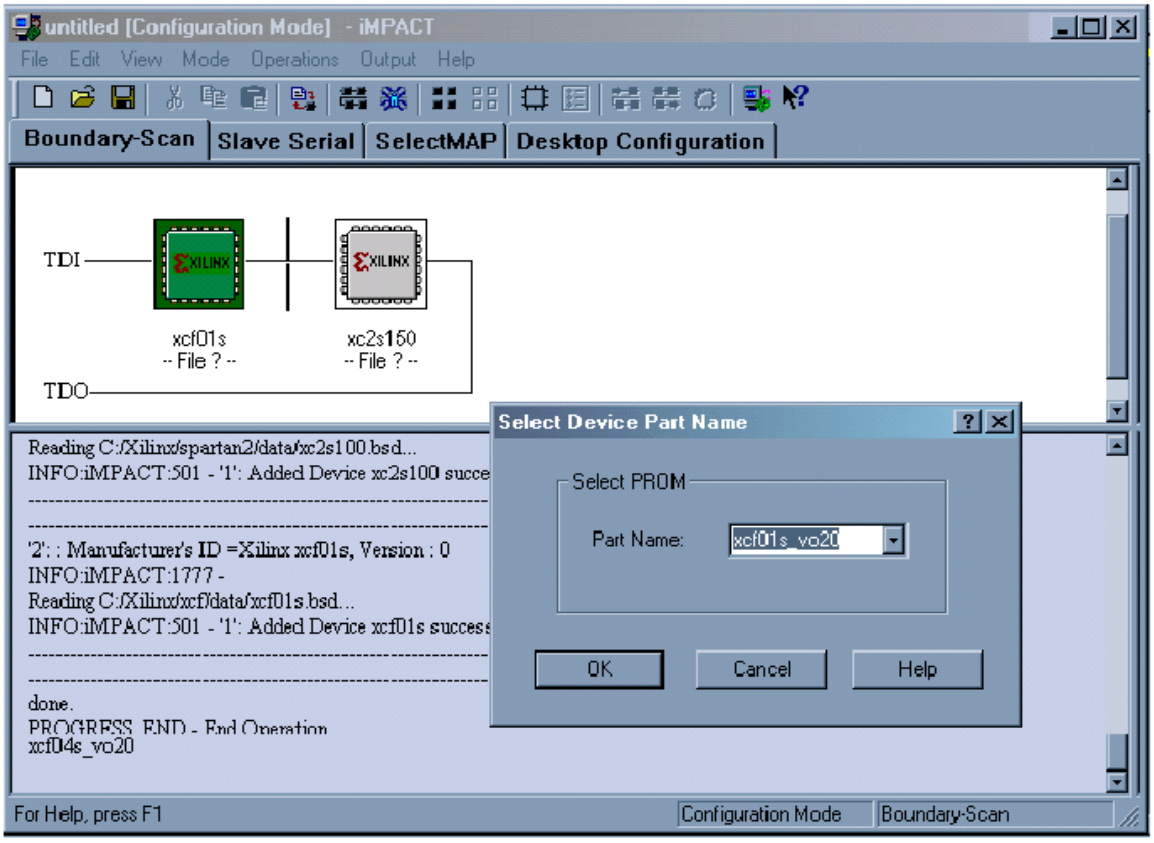

Boundary-Scan Mode Selection
Automatically connect to cable and identify Boundary-Scan chain
Enter a Boundary-Scan Chain
< Back
Finish
Cancel
Help

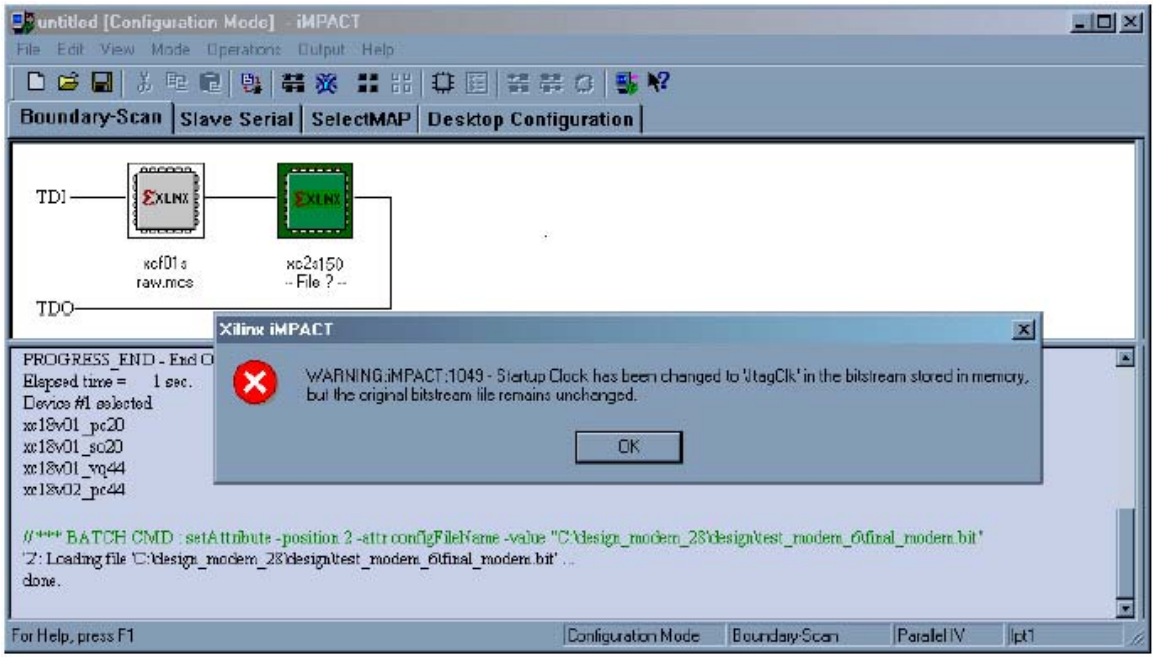

untitled [Configuration Mode] - iMPACT
File Edit View Mode Operations Output Help
Boundary-Scan
Slave Serial
SelectMAP
Desktop Configuration
Cable Notification
Connecting to download cable...
CB_PROGRESS_END - End Operation.
Elapsed time = 25 sec.
Cable autodetection
Connecting to the selected cable...
Configuration Mode
Boundary-Scan



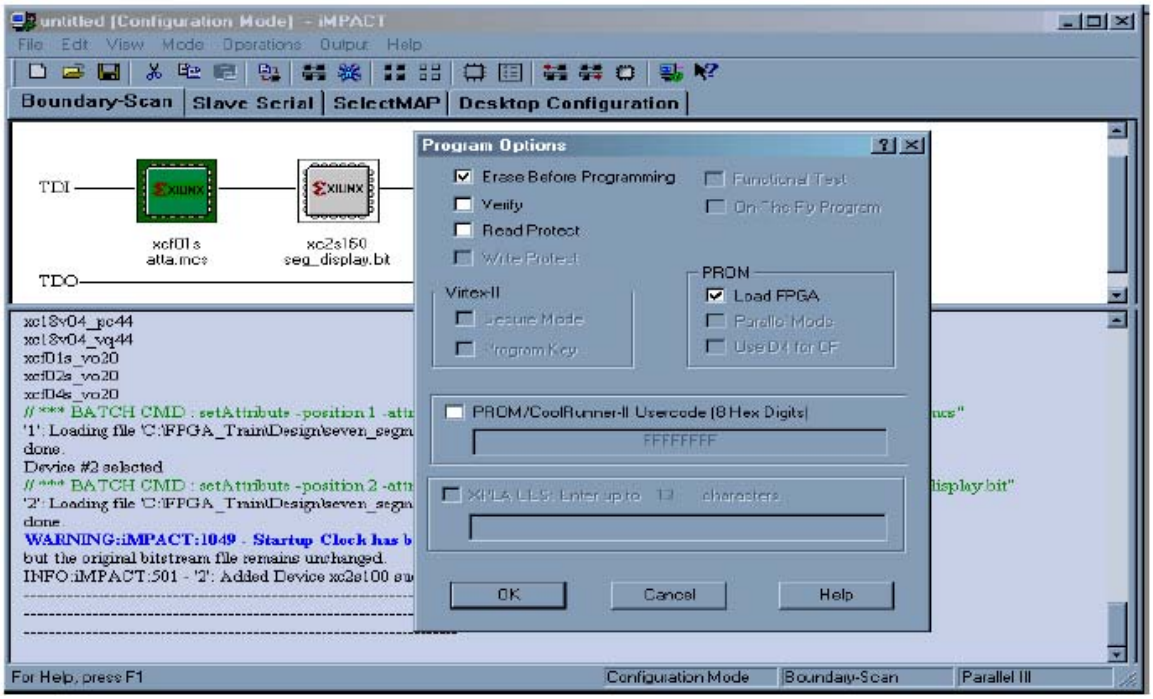



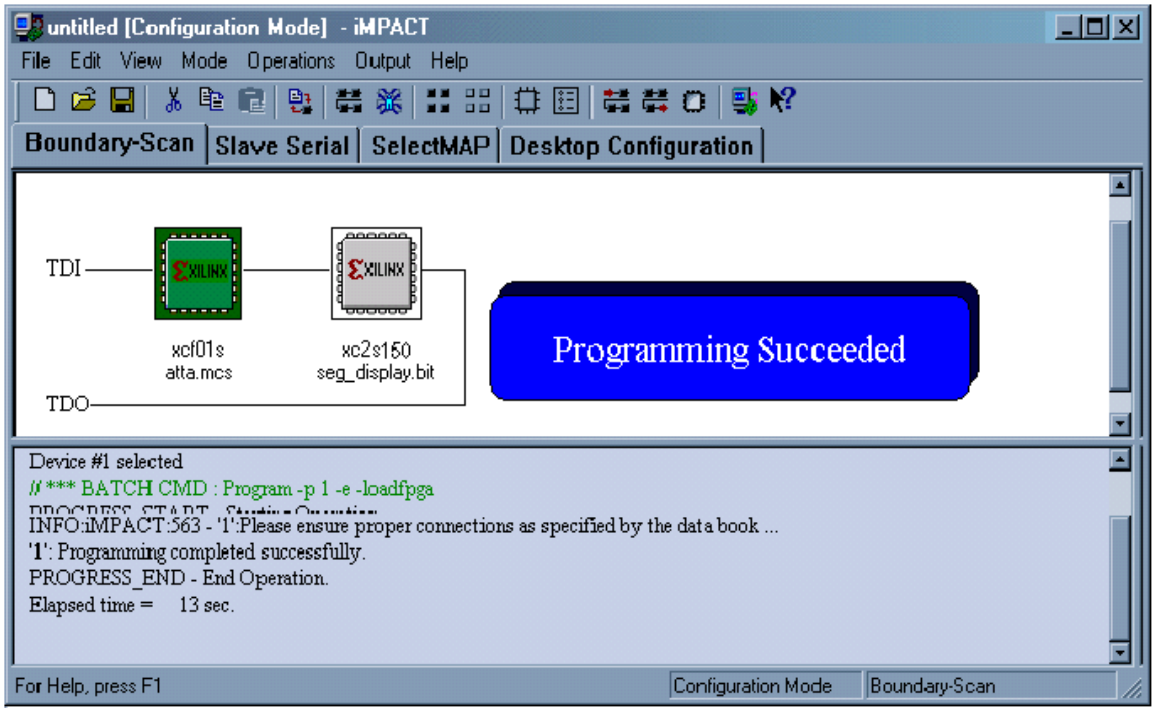

- Add .mcs file to PROM and .bit file to FPGA.
- Select the part number of PROM as xcf01s_vo20.




- Ignore the warning, Right click the PROM and click program.
- Click the following options in an appearing dialog box.
- The PROM is programmed successfully. The PROM will configure the FPGA automatically.


untitled [Configuration Mode] - iMPACT
File Edit View Mode Operations Output Help
Boundary-Scan Slave Serial SelectMAP Desktop Configuration
TDI
xcf01s
atta.mcs
xc2s150
seg_display.bit
TDO
Program Options
Erase Before Programming
Verify
Read Protect
Write Protect
Functional Test
On-The-Fly Program
PROM
Load FPGA
Parallel Mode
Use D4 for CF
Virtex-II
Secure Mode
Program Key
PROM/CoolRunner-II Usercode (8 Hex Digits)
FFFFFFFF
OK
Cancel
Help
xc18v04_pc44
xc18v04_vq44
xcf01s_vo20
xcf02s_vo20
xcf04s_vo20
// *** BATCH CMD : setAttribute -position 1 -attr
'1': Loading file 'C:\FPGA_Train\Design\seven_segm
done.
Device #2 selected
// *** BATCH CMD : setAttribute -position 2 -attr
'2': Loading file 'C:\FPGA_Train\Design\seven_segm
done.
WARNING:iMPACT:1049 - Startup Clock has b
but the original bitstream file remains unchanged.
INFO:iMPACT:501 - '2': Added Device xc2s100
For Help, press F1
Configuration Mode
Boundary-Scan
Parallel III


untitled [Configuration Mode] - iMPACT
File Edit View Mode Operations Output Help
Boundary-Scan Slave Serial SelectMAP Desktop Configuration
TDI
xcf01s
atta.mcs
xc2s150
seg_display.bit
TDO
Programming Succeeded
Device #1 selected
// *** BATCH CMD : Program -p 1 -e -loadfpga
INFO:iMPACT:563 - '1':Please ensure proper connections as specified by the data book ...
'1': Programming completed successfully.
PROGRESS_END - End Operation.
Elapsed time = 13 sec.
For Help, press F1
Configuration Mode
Boundary-Scan

# APPENDIX A
## CONNECTIONS B/W NT-FTRAIN, FROM II HEADERS AND FPGA

| Conn | Pin | Signal | Header | FPGA | | Conn | Pin | Signal | Header | FPGA |
|---|---|---|---|---|---|---|---|---|---|---|
| J18 | 1 | GCK3 | J5_21 | P185 | | J19 | 1 | I/O180 | J5_20 | P180 |
| J18 | 2 | GCK2 | J5_22 | P182 | | J19 | 2 | I/O181 | J5_19 | P181 |
| J18 | 3 | I/0188 | J5_23 | P188 | | J19 | 3 | I/O178 | J5_18 | P178 |
| J18 | 4 | I/O187 | J5_24 | P187 | | J19 | 4 | I/O179 | J5_17 | P179 |
| J18 | 5 | I/O189 | J5_25 | P189 | | J19 | 5 | I/O175 | J5_16 | P175 |
| J18 | 6 | I/O191 | J5_26 | P191 | | J19 | 6 | I/O176 | J5_15 | P176 |
| J18 | 7 | I/O192 | J5_27 | P192 | | J19 | 7 | I/O173 | J5_14 | P173 |
| J18 | 8 | I/O193 | J5_28 | P193 | | J19 | 8 | I/O174 | J5_13 | P174 |
| J18 | 9 | I/O194 | J5_29 | P194 | | J19 | 9 | I/O168 | J5_12 | P168 |
| J18 | 10 | I/O195 | J5_30 | P195 | | J19 | 10 | I/O172 | J5_11 | P172 |
| J18 | 11 | I/O199 | J5_31 | P199 | | J19 | 11 | I/O166 | J5_10 | P166 |
| J18 | 12 | I/O200 | J5_32 | P200 | | J19 | 12 | I/O167 | J5_9 | P167 |
| J18 | 13 | I/O201 | J5_33 | P201 | | J19 | 13 | I/O164 | J5_8 | P164 |
| J18 | 14 | I/O202 | J5_34 | P202 | | J19 | 14 | I/O165 | J5_7 | P165 |
| J18 | 15 | I/O203 | J5_35 | P203 | | J19 | 15 | I/O162 | J5_6 | P162 |
| J18 | 16 | I/O204 | J5_36 | P204 | | J19 | 16 | I/O163 | J5_5 | P163 |
| J18 | 17 | I/O205 | J5_37 | P205 | | J19 | 17 | I/O160 | J5_4 | P160 |
| J18 | 18 | I/O206 | J5_38 | P206 | | J19 | 18 | I/O161 | J5_3 | P161 |
| J18 | 19 | 3.3V | | | | J19 | 19 | 3.3V | | |
| J18 | 20 | 3.3V | | | | J19 | 20 | 3.3V | | |
| J18 | 21 | GND | J6_1 | P1 | | J19 | 21 | GND | J5_1 | P19 |
| J18 | 22 | I/O3 | J6_2 | P3 | | J19 | 22 | NC | | |
| J18 | 23 | I/O5 | J6_3 | P5 | | J19 | 23 | NC | | |
| J18 | 24 | I/O4 | J6_4 | P4 | | J19 | 24 | NC | | |
| J18 | 25 | I/O7 | J6_5 | P7 | | J19 | 25 | I/O154 | J3_5 | P154* |
| J18 | 26 | I/O6 | J6_6 | P6 | | J19 | 26 | I/O153 | J3_6 | P153* |
| J18 | 27 | I/O9 | J6_7 | P9 | | J19 | 27 | I/O152 | J3_7 | P152* |
| J18 | 28 | I/O8 | J6_8 | P8 | | J19 | 28 | I/O151 | J3_8 | P151* |
| J18 | 29 | I/O14 | J6_9 | P14 | | J19 | 29 | I/O150 | J3_9 | P150* |
| J18 | 30 | I/O10 | J6_10 | P10 | | J19 | 30 | I/O149 | J3_10 | P149* |
| J18 | 31 | I/O16 | J6_11 | P16 | | J19 | 31 | I/O148 | J3_11 | P148* |
| J18 | 32 | I/O15 | J6_12 | P15 | | J19 | 32 | I/O147 | J3_12 | P147* |
| J18 | 33 | I/O18 | J6_13 | P18 | | J19 | 33 | I/O142 | J3_13 | P142 |
| J18 | 34 | I/O17 | J6_14 | P17 | | J19 | 34 | I/O146 | J3_14 | P146 |
| J18 | 35 | I/O20 | J6_15 | P20 | | J19 | 35 | I/O141 | J3_15 | P141 |
| J18 | 36 | I/O21 | J6_16 | P21 | | J19 | 36 | I/O140 | J3_16 | P140 |
| J18 | 37 | I/O22 | J6_17 | P22 | | J19 | 37 | I/O138 | J3_17 | P138 |
| J18 | 38 | I/O23 | J6_18 | P23 | | J19 | 38 | I/O139 | J3_18 | P139 |
| J18 | 39 | I/O24 | J6_19 | P24 | | J19 | 39 | I/O135 | J3_19 | P135 |
| J18 | 40 | I/O27 | J6_20 | P27 | | J19 | 40 | I/O136 | J3_20 | P136 |

| Conn | Pin | Signal | Header | FPGA | | Conn | Pin | Signal | Header | FPGA |
|---|---|---|---|---|---|---|---|---|---|---|
| J18 | 41 | I/O29 | J6_21 | P29 | | J19 | 41 | I/O133 | J3_21 | p133 |
| J18 | 42 | INHIBIT | J6_22 | | | J19 | 42 | I/O134 | J3_22 | P134 |
| J18 | 43 | I/O30 | J6_23 | P30 | | J19 | 43 | I/O129 | J3_23 | P129 |
| J18 | 44 | I/O31 | J6_24 | P31 | | J19 | 44 | I/O132 | J3_24 | P132 |
| J18 | 45 | I/O33 | J6_25 | P33 | | J19 | 45 | I/O126 | J3_25 | P126 |
| J18 | 46 | I/O34 | J6_26 | P34 | | J19 | 46 | I/O127 | J3_26 | P127 |
| J18 | 47 | I/O35 | J6_27 | P35 | | J19 | 47 | I/O123 | J3_27 | P123 |
| J18 | 48 | I/O36 | J6_28 | P36 | | J19 | 48 | I/O125 | J3_28 | P125 |
| J18 | 49 | I/O37 | J6_29 | P37 | | J19 | 49 | I/O121 | J3_29 | p121 |
| J18 | 50 | I/O41 | J6_30 | P41 | | J19 | 50 | I/O122 | J3_30 | p122 |
| J18 | 51 | I/O42 | J6_31 | P42 | | J19 | 51 | I/O119 | J3_31 | p119 |
| J18 | 52 | I/O43 | J6_32 | P43 | | J19 | 52 | I/O120 | J3_32 | p120 |
| J18 | 53 | I/O44 | J6_33 | P44 | | J19 | 53 | I/O114 | J3_33 | p114 |
| J18 | 54 | I/O45 | J6_34 | P45 | | J19 | 54 | I/O115 | J3_34 | p115 |
| J18 | 55 | I/O46 | J6_35 | P46 | | J19 | 55 | I/O112 | J3_35 | P112 |
| J18 | 56 | I/O47 | J6_36 | P47 | | J19 | 56 | I/O113 | J3_36 | P113 |
| J18 | 57 | I/O48 | J6_37 | P48 | | J19 | 57 | I/O110 | J3_37 | P110 |
| J18 | 58 | I/O49 | J6_38 | P49 | | J19 | 58 | I/O111 | J3_38 | P111 |
| J18 | 59 | GND | J6_39 | P11 | | J19 | 59 | I/O108 | J3_39 | P108 |
| J18 | 60 | VIN | J6_40 | | | J19 | 60 | I/O109 | J3_40 | P109 |
| J18 | 61 | I/O58 | J4_40 | P58 | | J19 | 61 | GND | J4_1 | P25 |
| J18 | 62 | I/057 | J4_39 | P57 | | J19 | 62 | TDO | J4_2 | p157 |
| J18 | 63 | I/O60 | J4_38 | P60 | | J19 | 63 | CLK | J4_3 | P77 |
| J18 | 64 | I/O59 | J4_37 | P59 | | J19 | 64 | RESET | J4_4 | P36 |
| J18 | 65 | I/O62 | J4_36 | P62 | | J19 | 65 | TCK | J4_7 | P207 |
| J18 | 66 | I/O61 | J4_35 | P61 | | J19 | 66 | TDI | J4_6 | P159 |
| J18 | 67 | I/O67 | J4_34 | P67 | | J19 | 67 | TMS | J4_5 | P2 |
| J18 | 68 | I/O63 | J4_33 | P63 | | J19 | 68 | I/O102 | J4_8 | P102 |
| J18 | 69 | I/O69 | J4_32 | P69 | | J19 | 69 | I/O101 | J4_9 | P101 |
| J18 | 70 | I/O68 | J4_31 | P68 | | J19 | 70 | I/O100 | J4_10 | P100 |
| J18 | 71 | I/O71 | J4_30 | P71 | | J19 | 71 | I/O99 | J4_11 | P99 |
| J18 | 72 | I/O70 | J4_29 | P70 | | J19 | 72 | I/O98 | J4_12 | P98 |
| J18 | 73 | I/O74 | J4_28 | P74 | | J19 | 73 | I/O97 | J4_13 | P97 |
| J18 | 74 | I/O73 | J4_27 | P73 | | J19 | 74 | I/O96 | J4_14 | P96 |
| J18 | 75 | I/O75 | J4_26 | P75 | | J19 | 75 | I/O95 | J4_15 | P95 |
| J18 | 76 | I/O81 | J4_25 | P81 | | J19 | 76 | I/O94 | J4_16 | P94 |
| J18 | 77 | I/O83 | J4_24 | P83 | | J19 | 77 | I/O90 | J4_17 | P90 |
| J18 | 78 | I/O82 | J4_23 | P82 | | J19 | 78 | I/O89 | J4_18 | P89 |
| J18 | 79 | I/O86 | J4_22 | P86 | | J19 | 79 | I/O87 | J4_19 | p87 |
| J18 | 80 | I/O84 | J4_21 | P84 | | J19 | 80 | I/O88 | J4_20 | P88 |

*Table A : Header J18, J19, FROM II and FPGA connections*

# APPENDIX B

## FPGA TO SERIAL PORT CONNECTIONS

| S/NO | SERIAL PORT | FPGA PINS |
|---|---|---|
| 1 | 3 | I/O71 |
| 2 | 2 | I/O69 |

*TABLE B: SERIAL PORT TO FPGA CORRESPONDING CONNECTIONS*

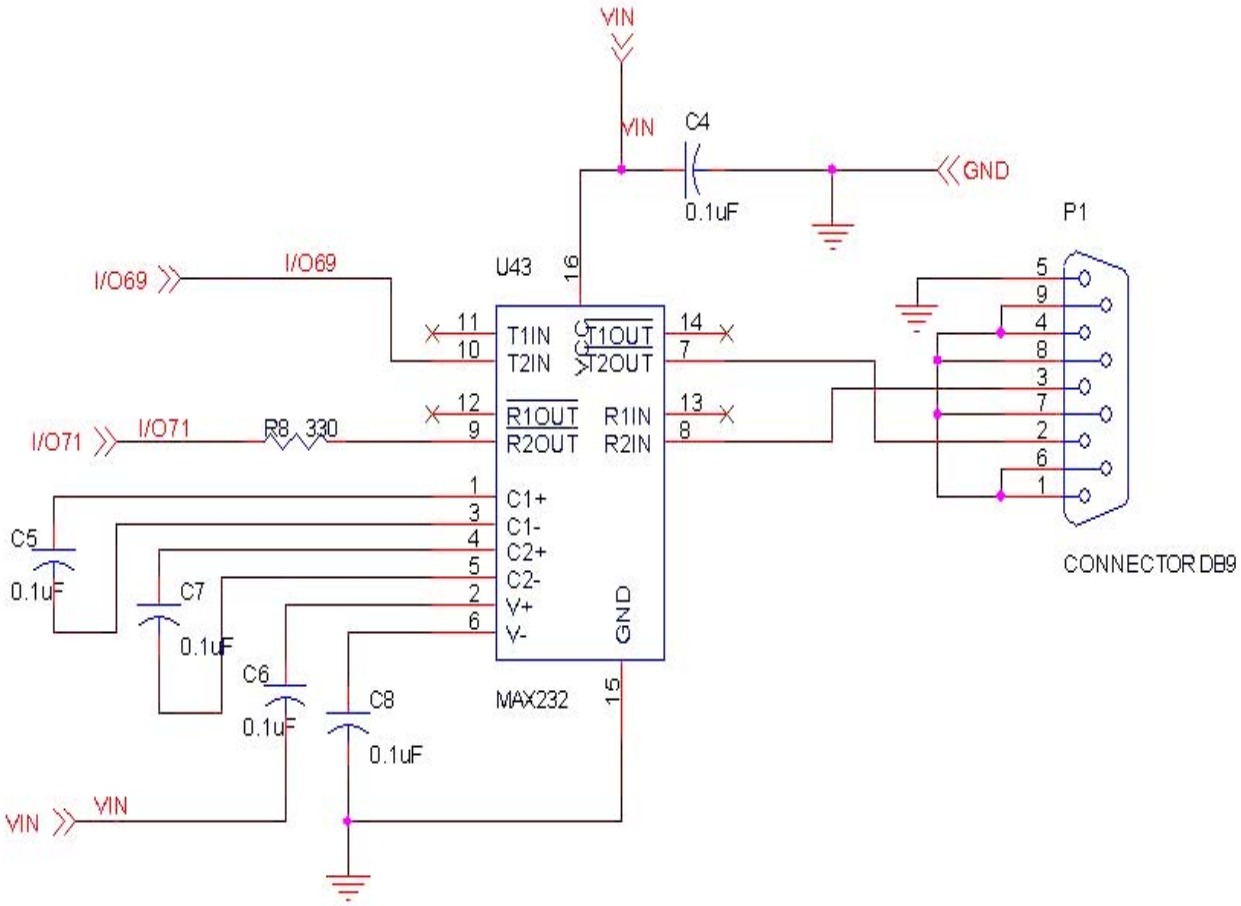
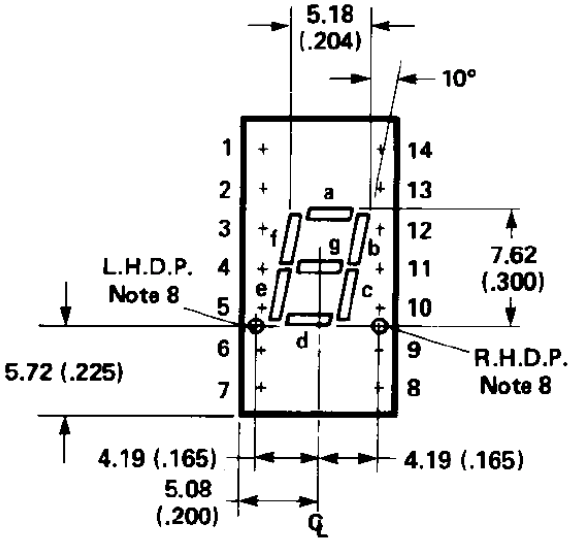



*Figure B: SERIAL PORT TO FPGA CONNECTIONS*

# APPENDIX C

## SEVEN SEGMENT CONNECTIONS




A,B,C

| S/NO | 7 SEGMENT PINS | HEADER | FPGA PINS |
|---|---|---|---|
| 1 | A | J4_12 | I/O98 |
| 2 | B | J4_13 | I/O97 |
| 3 | C | J4_15 | I/O95 |
| 4 | D | J4_16 | I/O94 |
| 5 | E | J4_17 | I/O90 |
| 6 | F | J4_20 | I/O88 |
| 7 | G | J4_18 | I/O89 |
| 8 | Dot (DP) | J4_14 | I/O96 |

*TABLE C: SEVEN SEGMENT (D24) TO FPGA CONNECTIONS*

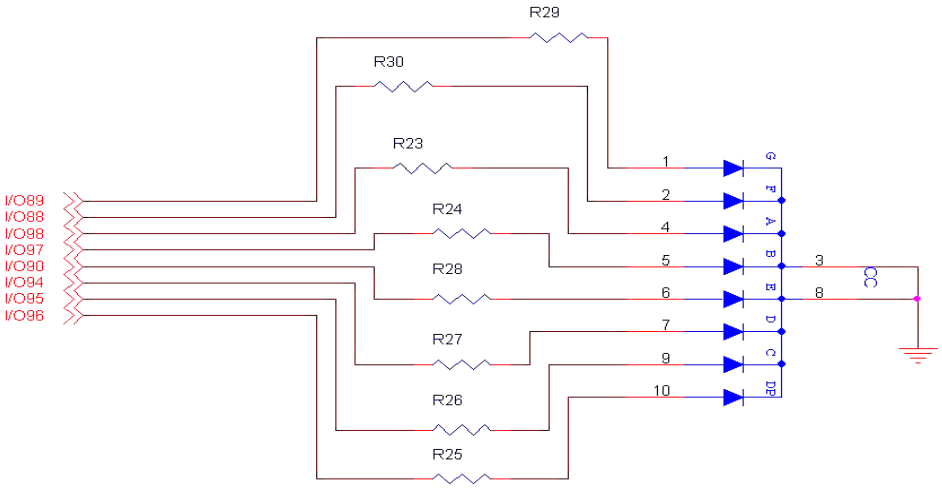



*Figure C: SEVEN SEGMENT TO FPGA CONNECTIONS*

## SEVEN SEGMENT CONNECTIONS

| S/NO | 7 SEGMENT PINS | HEADER | FPGA PINS |
|---|---|---|---|
| 1 | A | J3-10 | I/O149 |
| 2 | B | J3-09 | I/O150 |
| 3 | C | J3-06 | I/O153 |
| 4 | D | J3-07 | I/O152 |
| 5 | E | J3-08 | I/O151 |
| 6 | F | J3-11 | I/O148 |
| 7 | G | J3-12 | I/O147 |
| 8 | Dot | J3-05 | I/O154 |

*TABLE D: SEVEN SEGMENT (D25) TO FPGA CONNECTIONS*

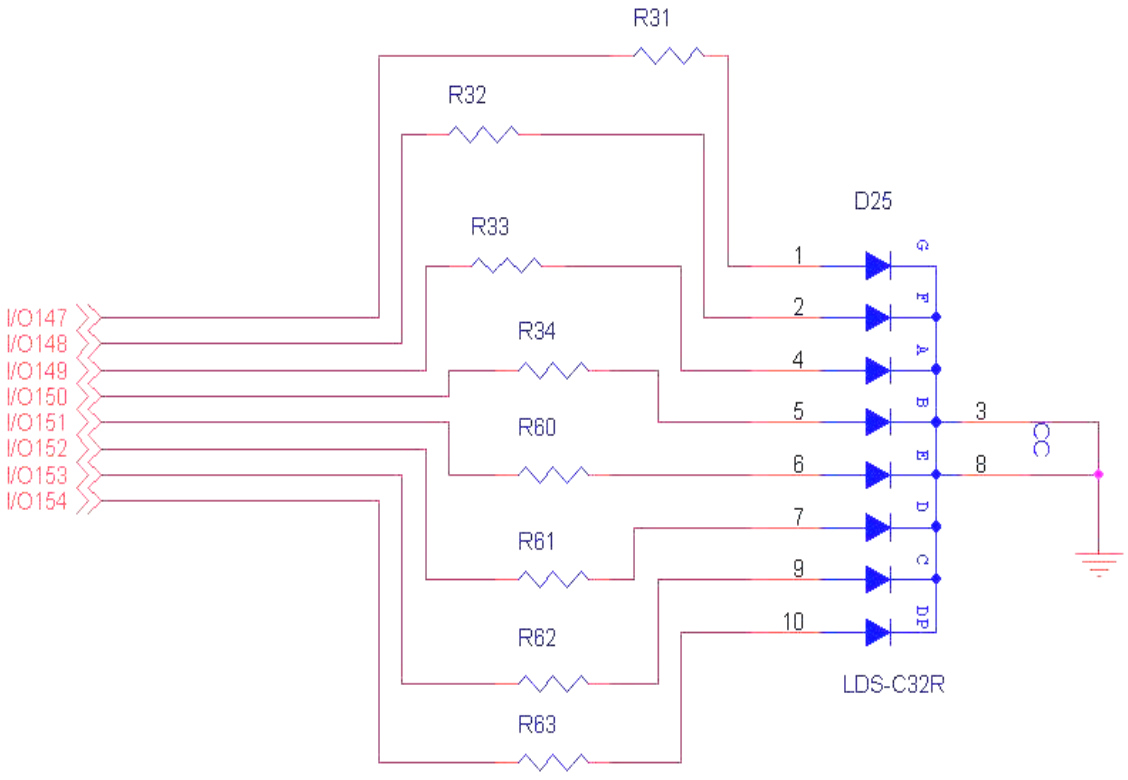
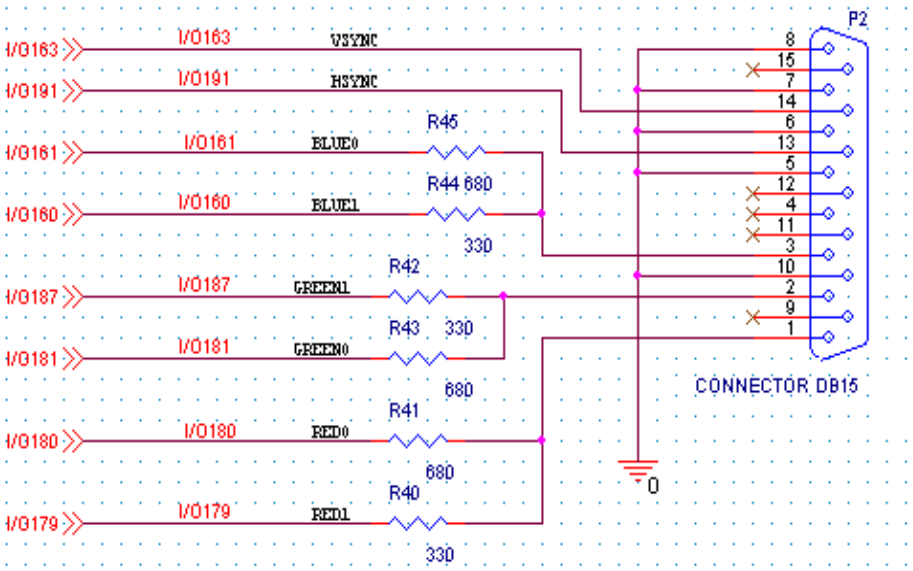



*Figure D: SEVEN SEGMENT TO FPGA CONNECTIONS*

## APPENDIX D

# VGA CONNECTIONS




| S/NO | VGA Connector | HEADER | FPGA PINS |
|---|---|---|---|
| 1 | VSYNC | J5-5 | I/O163 |
| 2 | HSYNC | J5-26 | I/O191 |
| 3 | Blue0 | J5-3 | I/O161 |
| 4 | Blue1 | J5-4 | I/O160 |
| 5 | Green0 | J5-19 | I/O181 |
| 6 | Green1 | J5-24 | I/O187 |
| 7 | Red0 | J5-20 | I/O180 |
| 8 | Red1 | J5-17 | I/O179 |

## APPENDIX E

### DIP SWITCH CONNECTIONS

| S.No. | Switch No | HEADER | FPGA PINS |
|---|---|---|---|
| 1 | Sw-1 | J5_23 | I/O 188 |
| 2 | Sw-2 | J5_25 | I/O 189 |
| 3 | Sw-3 | J5_27 | I/O 192 |
| 4 | Sw-4 | J5_29 | I/O 194 |
| 5 | Sw-5 | J5_31 | I/O199 |
| 6 | Sw-6 | J5_33 | I/O 201 |
| 7 | Sw-7 | J5_35 | I/O 203 |
| 8 | Sw-8 | J5_37 | I/O 205 |

# APPENDIX F

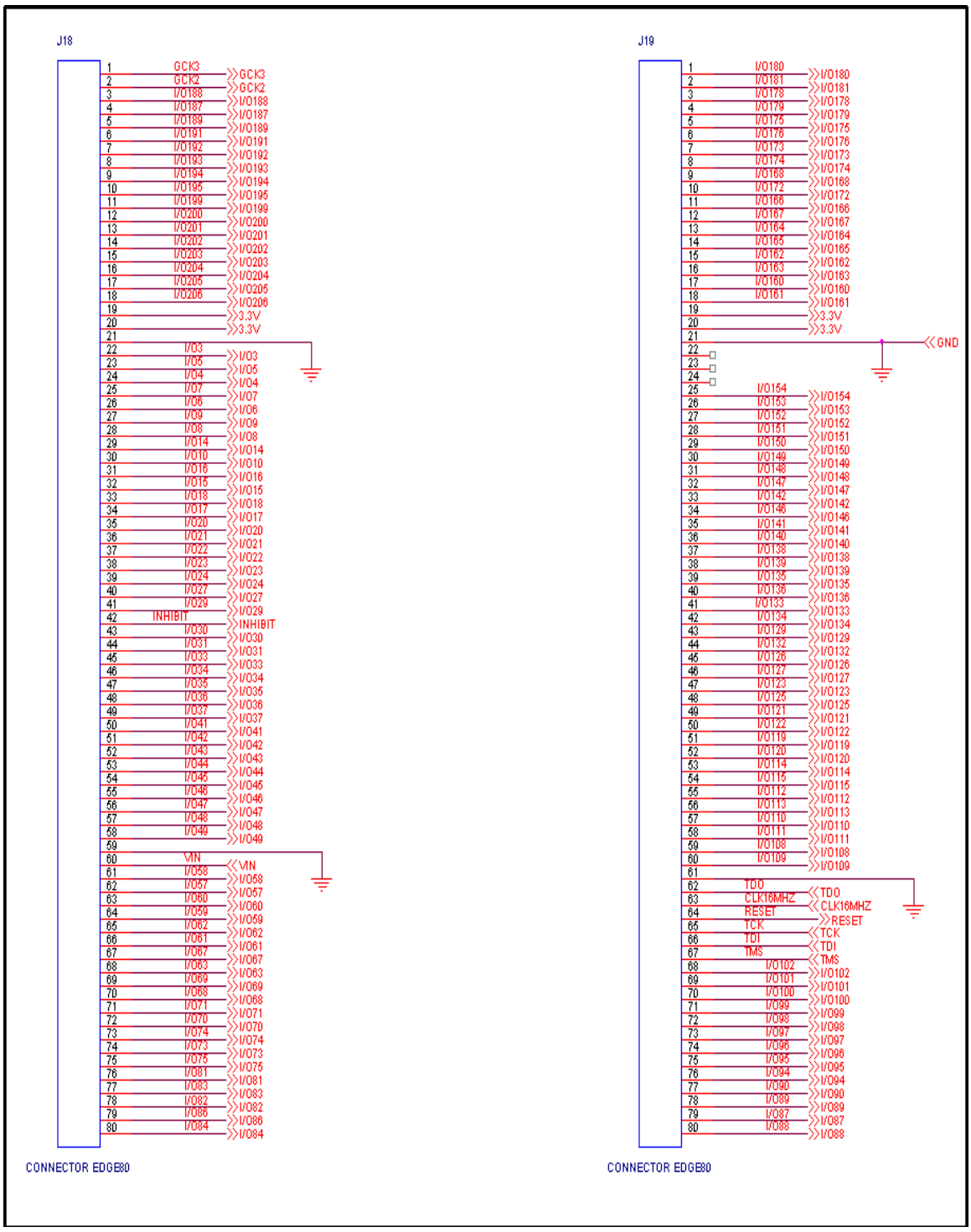



*Figure D: NT-FTRAIN-01 80 pins Connectors.*

# APPENDIX G

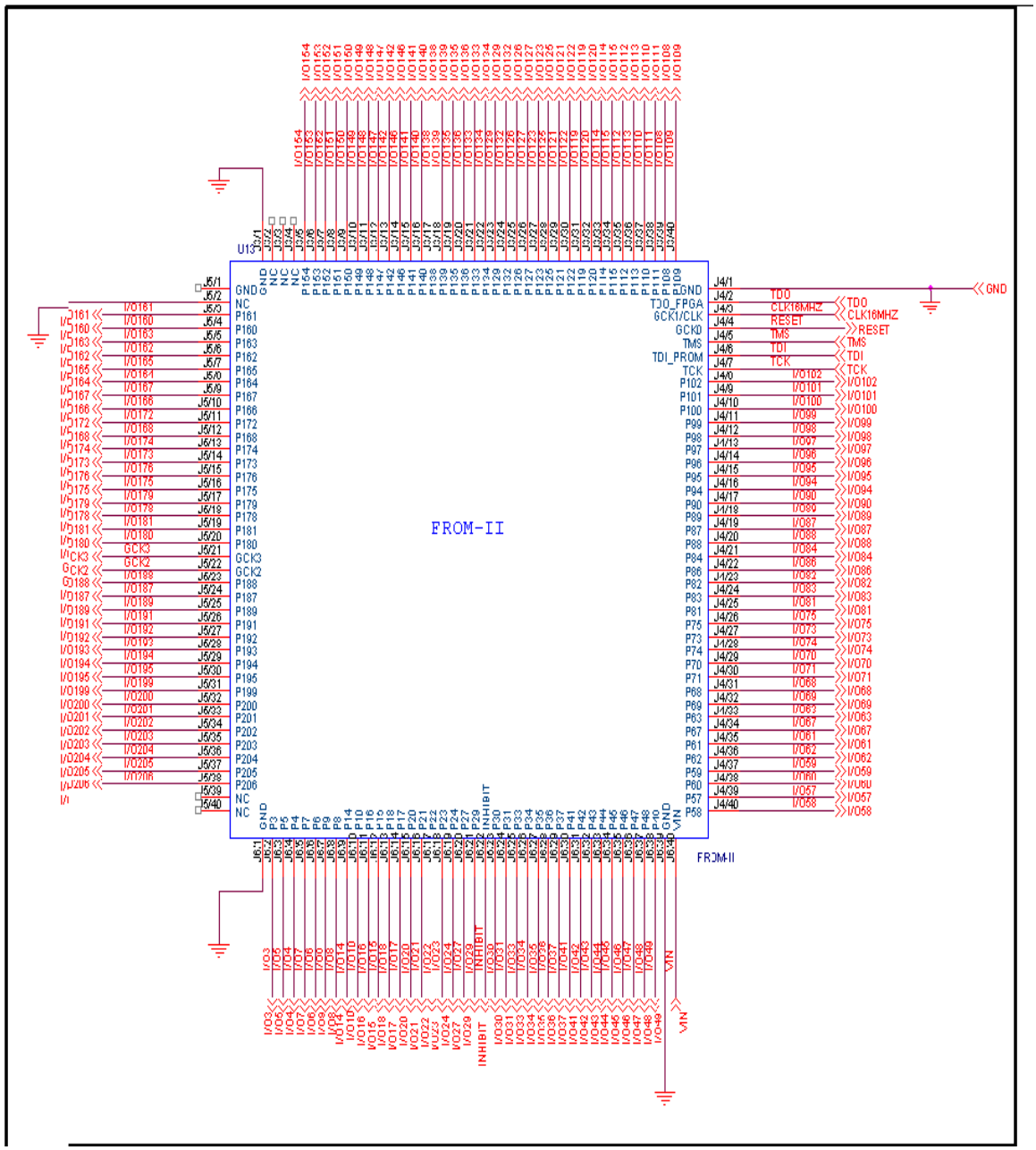
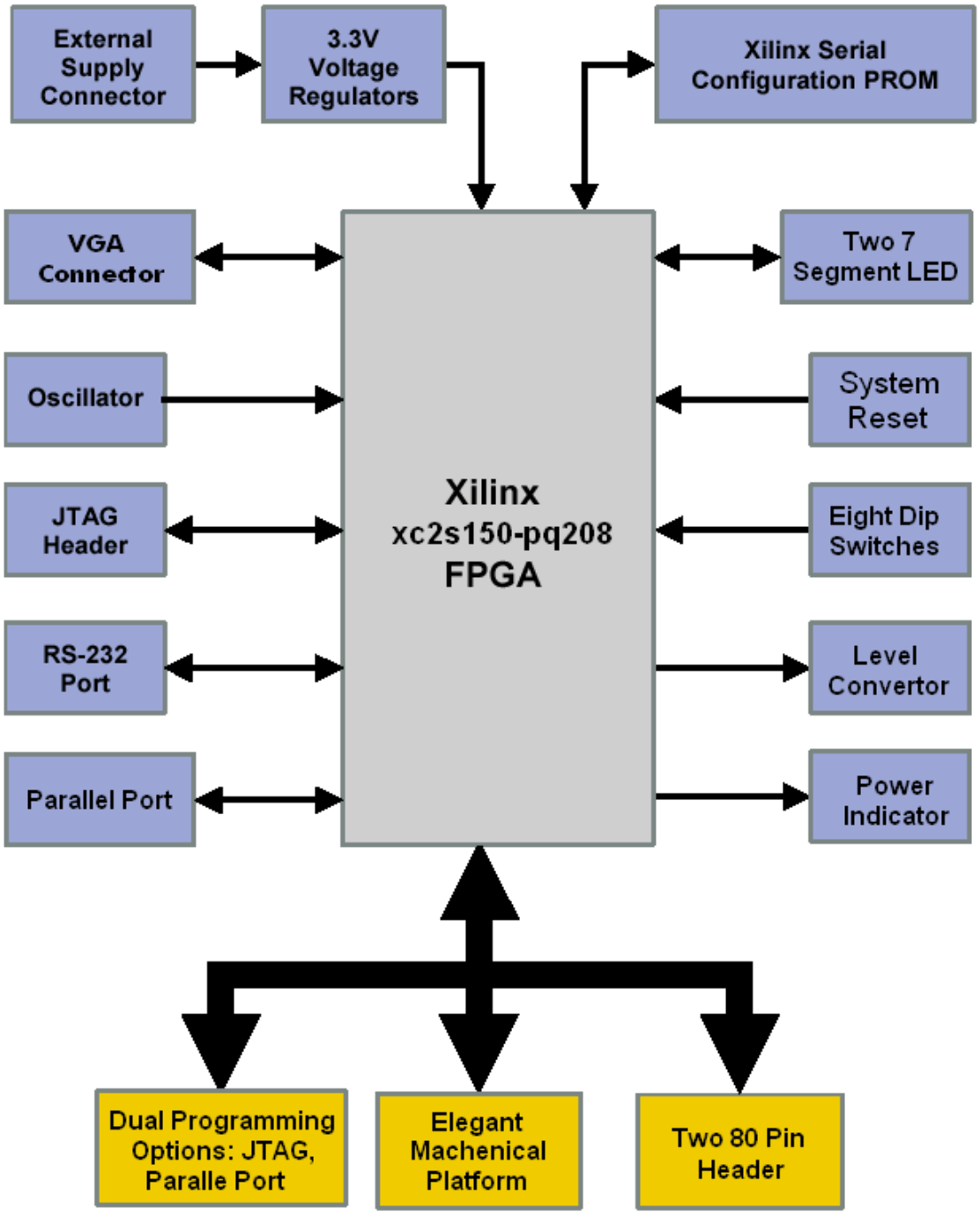



*Figure E: Pinout of FPGA*

## APPENDIX H

### Block Diagram of NexTek FPGA Training Board



# REFERENCES

The following references are used for the compilation of this technical manual.


- http://www.support.xilinx.com/
- http://www.xilinx.com/apps/
- http://www.synopsys.com/
- http://www.exemplar.com/
- http://www.model.com/
- http://www.verilog.com/
- http://www.xilinx.com/ipcenter/

# Ordering Information:

NT-FTRAIN-01-XXX -T

Where,

**XXX** is the number of logic gates FPGA to have. The values are:
'050' for 50,000 gates
'100' for 100,000 gates
and, '150' for 150,000 gates

**T** is the Temperature type of the module. The values are:
'C' for commercial range (0 Celsius to 75 Celsius) and, 'I' fro industrial version (-35 Celsius to 80 Celsius)

For pricing and further information:

**NexTek Service**
133, St-65. F-11/4. Islamabad-44000
Fax: +92-51-2103399
Email: sales@NexTekService.com