



**SAVONIA**

## **Software Radio Application Using Warp FPGA Board**

**Sherif Ogunola**

Bachelor' Thesis

Accepted \_\_\_\_ . \_\_\_\_ . \_\_\_\_\_

Field of Study Bachelor of Engineering			
Degree Programme Information Technology			
Author(s) Sherif Ogunola			
Title of Thesis Software radio Application using Warp FPGA board			
Date	28 December, 2011	Pages/Appendices	30-40
Supervisor(s) Arto Toppinen			
Client Organisation/Partners SAVONIA UAS			
<p>Abstract</p> <p>The purpose of this final project was to implement the application of software radio using warp FPGA board.</p> <p>This was achieved by getting resources before starting the project at all. The first thing was to get information about the project and this was possible by reading articles concerning the Topic and getting previous projects of the same topic online through the Theseus data base. After that, the project proceeded in the laboratory by getting the necessary tools and software installed on a PC as read from the documentation of the WARP FPGA board.</p> <p>Xilinx platform studio 13.1 including the EDK was used to develop the application. The application was able to support MIMO technology on the board and the result was tested using Matlab.</p> <p>It was seen in the result that there was a data being transferred from the trasciever to the receiver part of the antenna, This was as a result of the blink in the led part of the board.</p> <p>Software radios have the ability to reconfigure itself regardless of the environment, It can be easily upgraded with enhanced features and they can talk and listen to multiple channels at the same time.</p> <p>In the global world today, the implementation of software radio application is being used in some organizations like in the Army and as time goes on in the future, the application will take over the communication society because of its vast advantages.</p> <p>The goal was achieved using a pc, antennas and a Warp FPGA vertex iv board from Mango Communications. The application was able to run on the Warp FPGA board which makes the transmission and receiving of data possible.</p>			
Keywords			
FPGA, WARP, EDK,			



## CONTENTS

<b>1 INTRODUCTION</b> .....	6
<b>1.1 Background Information</b> .....	6
<b>1.2 Working Environment of the project</b> .....	6
<b>2 The Purpose of the project</b> .....	7
<b>2.1 Project Description</b> .....	7
<b>2.1.1 FPGA</b> .....	7
<b>3 WARP FPGA_v2.2</b> .....	9
<b>The overview of the board</b> .....	9
<b>4 WARP FPGA Details</b> .....	11
<b>5 WARP FPGA power supply</b> .....	12
<b>5.1 External power supply</b> .....	12
<b>5.1.1 FPGA Power supply</b> .....	12
<b>5.1.2 Daughter card power supply</b> .....	12
<b>5.1.3 Monitoring Voltage Levels</b> .....	13
<b>5.1.4 Bypassing Power Supplies</b> .....	14
<b>5.1.5 Primary supplies</b> <b>TABLE 4. Identifying the screw terminals</b> .	15
<b>5.1.6 MGT Supplies</b> <b>TABLE 5. MGT Supplies</b> .....	15
<b>6 FPGA CONFIGURATION</b> .....	18
<b>6.1 USB Configuration</b> .....	18
<b>6.2 External JTAG cable</b> .....	18
<b>6.3 System ACE compact Flash</b> .....	19
<b>7 WARP FPGA BOARD CLOCKING</b> .....	21
<b>7.1 On board oscillators</b> .....	21
<b>7.2 Off-board clock sources</b> .....	21
<b>7.3 System ACE CF Clocking</b> .....	22
<b>8 MGT Clocking</b> .....	23
<b>8.1 Warp FPGA Board MGTs</b> .....	23
<b>8.2 MGT Interfaces</b> .....	23
<b>8.1 MGT Clocking</b> .....	26
<b>8.2 FPGA MGT Clock Input</b> .....	27
<b>9 Warp FPGA Board Memory</b> .....	28
<b>9.1 On-chip Memory</b> .....	28
<b>9.2 DDR2 SO- DIMM</b> .....	28
<b>9.3 Push Buttons</b> .....	30

<b>9.4 DIP-switch</b> .....	30
<b>9.5 LEDs</b> .....	30
<b>9.6 Hex-Display</b> .....	30
<b>10Procedure</b> .....	32
<b>10.1 Software</b> .....	38
<b>10.2 Requirement and Set up</b> .....	38
<b>11Result</b> .....	41
<b>12Conclusion</b> .....	42
<b>13REFERENCES</b> .....	43

## **1 INTRODUCTION**

### **1.1 Background Information**

Software radio is an art and science of building radio using software. Looking at today's technology, there is still some RF involved, but the idea is to get the software as close to the antenna as feasible. The priority is turning hardware to software. The problem of communicating with people using different types of equipments can only be solved with software re-programmable radio. For ages, radio systems were normally designed and built to communicate using one or two wave forms. One software radio can communicate with many different radios with only a change in software parameters. Software radio development is different from one another, the development of software radio in one country might be different from the one developed in another country but the aim is for the software to communicate with one another (IEEE, software radio 2009).

Warp FPGA board is a scalable and extensible programmable wireless platform, built from the ground up to prototyped advanced wireless system. The open access warp repository allows exchange and sharing of new physical and network layer network Architecture, building a true community platform. Xilinx FPGAs are used to enable programmability of both physical and network layer protocol on a single platform which is both deployable and observable at all layers.

### **1.2 Working Environment of the project**

The project was carried out at the wireless sensor laboratory of Savonia University of Applied Sciences, Kuopio in Finland. Lots of projects have been carried out in the laboratory both in research form and teaching. The laboratory is made up of different equipment for Telecommunication and sensor technology. It gives opportunity for students to develop their skills in preparation to real life projects.

## **2 The Purpose of the project**

The purpose of the project is to implement the application of software radio on the Warp FPGA board. The Warp FPGA must be able to work like a real time software radio after debugging the application on it. The implementation should also support MIMO technology (Multiple Input Multiple Output).

### **2.1 Project Description**

#### **2.1.1 FPGA**

Field Programmable Gate Array (FPGA) is flexible alternative to custom integrated circuits. They can integrate both combinatorial and sequential logic of ten thousand of gates. During the past, software was considered flexible with hardware.

All these changed in field was in the approach to computing: a reprogrammable gate array. This gate array is a step above programmable logic devices (PLDs) in complexity. There are some programmable technologies that can be reprogrammed any number of times. FPGAs resemble Mask programmed Gate Arrays (MPGAs) in their modularity, however they also capable of the field configuration by the user. FPGAs are more expensive in unit cost compare to Application specific Integrated circuits (ASICs) (IEEE 2009).

Some of the features of FPGAs are:

- The masking layers used in fabrication process are not customized.
- There is a method to program the basic logic blocks and interconnects.
- The core is regular array of programmable logic cells.
- A routing matrix surrounds the basic logic blocks.
- Programmable I/Os run along the perimeter of the array (IEEE Software radio, 2009)

The generic chip is tailored for an application by downloading a proprietary software configuration. The programmable software takes the form of a bit stream. A bit stream is a string of binary 1s and 0s. The bit stream is used on board the FPGA to control the on-off state of various pass resistors and transmission gates (IEEE software radio, 2009).



FIGURE 1: Sample Warp FPGA board (warp.rice.edu, Warp FPGA board overview)

Figure 1 shows an example of FPGA board designed by Rice University WARP project. It contains all features listed in the FPGA description above.

### **3 WARP FPGA\_v2.2**

#### **The overview of the board**

This board is a 8"x8" PCB built around a Xilinx XC4VFX100FFG1517-11C Virtex-4 FPGA. It has different components and this will be explained step by step including the configuration.

The features of WARP FPGA board can be seen below

- Xilinx Virtex-4 FX100 FPGA (XC4VFX100-11FFG1517C)
- 10/100/1000 Ethernet (Marvell 88e1111 PHY)
- 4 WARP daughtercard slots
- 8 Multi-gigabit transceivers
  - 2 SATA interfaces (1 target, 1 host)
  - 2 SFP interfaces
  - 4 HSSDC2 interfaces
- DDR2 SO-DIMM slot (2GB SO-DIMM included with board)
- UART interfaces (1 on-board USB-UART, 1 DB9 RS-232)
- User I/O (16 LEDs, five push buttons, three seven segment displays, 16-bit 3.3v I/O)
- USB, JTAG & Compact Flash FPGA configuration (mangocomm.com WARP fpga board).

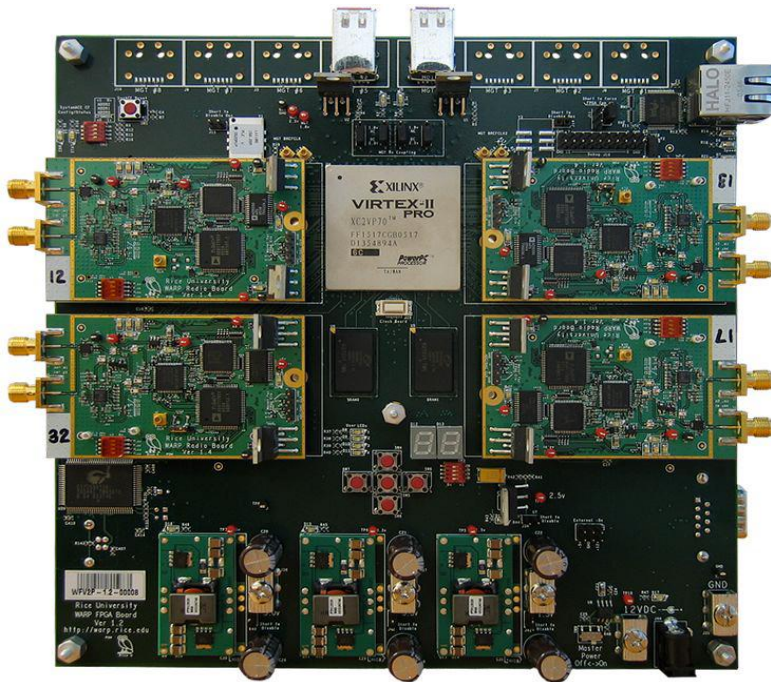


FIGURE 2: WARP FPGA board with its components (warp.rice.edu).

Figure 2 shows WARP FPGA boards with its component. Users are always taken precautions while using the board because nearly all the component on the board is sensitive to electrostatic discharge.

#### 4 WARP FPGA Details

The WARP FPGA is designed around the Xilinx Virtex-4 XC4VFX100FFG1517-11C FPGA.

TABLE 1. WARP FPGA details

Part	XC4VFX100
Package	FFG1517(1517 pin 1mm pitch BGA)
Speed grade	11 (middle grade)
Temperature range	C (commercial)

Table 1 shows the WARP FPGA details (Rice university wireless open Access research platform).

TABLE 2. V4FX100 FPGA resources

<b>Logic Slices</b>	42k
<b>DSP48 Slices</b>	160
<b>Block RAMs</b>	376
<b>PowerPC Cores</b>	2
<b>Tri-mode Ethernet MACs</b>	4

Table 2 refers to the V4FX100 resources (Rice university wireless open Access Research platform).

## 5 WARP FPGA power supply

### 5.1 External power supply

The WARP FPGA can work from a single external 12v supply. The supply is connected to the board's coaxial power connector. This connector requires an opposite connector with an inner diameter of 2.1mm, outer diameter of 5.5mm. A 50W AC-DC regulator from CUI (ETS120416UTC-P5P-SZ) was used in this project (Rice university wireless open Access research platform).

#### 5.1.1 FPGA Power supply

The virtex-4 FPGA has a number of different power inputs. Table 3 summarizes the power supply on the WARP FPGA board.

TABLE 3. Power supply on WARP FPGA board

Supply	Voltage	Description
VCC_EXT	12V	External supply
VCC_5	5.0v	Daughter card slot supply
VCC_INT	1.2v	FPGA core logic
VCC_AUX	2.5v	FPGA clock resources
VCC_O	3.3v, 2.5v & 1.8v	FPGA I/O banks
MGT_x	2.5v, 1.5v & 1.2v	MGT logic and I/O
VCC_0.9	0.9v	DDR2 SO-DIMM termination

The switching voltage regulator for WARP FPGA board is 5v, 3.3v, 1.8v and 1.2v. The 2.5v VCC\_AUX supply is a LT1764 linear regulator. The MGT supplies are all driven by linear supplies in order to minimize noise in the MGT circuitry. Three linear regulators (2 LT1963A and 1 LT1764) are used.

#### 5.1.2 Daughter card power supply

WARP FPGA board has four daughter card slots and they are supplied with 5v by a dedicated 18A switching regulator. A second power plane is also connected to the daughter card slots and can be driven by off-board supply via a dedicated 6-pin header on the FPGA board (J31). The header is not mounted by default (no current WARP daughter cards utilize this second pane). Rice university wireless open Access research platform.

### 5.1.3 Monitoring Voltage Levels

Every power and ground plane on the FPGA board includes a test point or screw terminal which can be used to monitor the plane's voltage. There are also test point and a screw terminal for ground. The location of the test points and screw terminals are illustrated in the picture below.

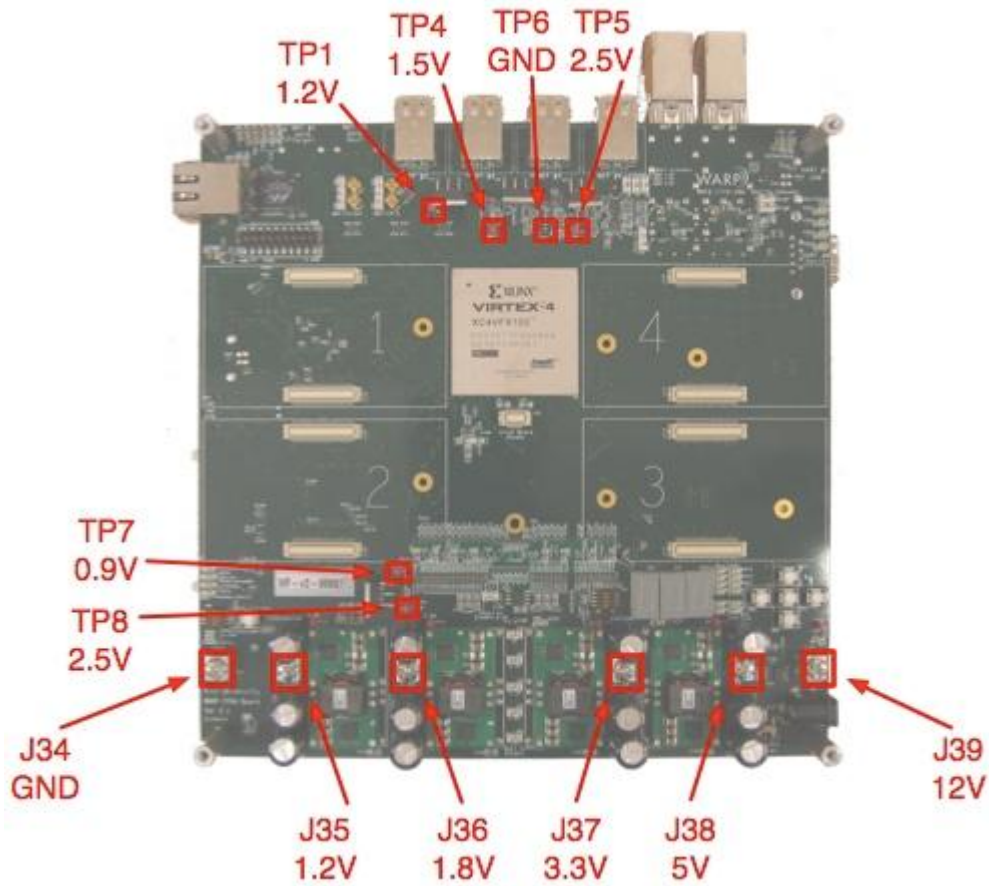


Figure 3 Test point screw terminals

Figure 3 shows the location of test point screw terminals. Most supplies (12v, 5v, 3.3v, digital 2.5v and MGT 2.5v) include a red LED which glows when power is applied. These five LEDs should always be illuminated when the FPGA board is powered on. If any of these LEDs is not glowing, immediately power must be switched off on the board in order to avoid damaging components (Rice university wireless open Access research platform).

#### **5.1.4 Bypassing Power Supplies**

The WARP FPGA does not contain any built in current measurement capabilities. To measure current consumption, the power plane of interest can be driven by an external, off board supply whose current can be measured. If an external supply is used, the on-board regulator for the plane must be disabled to avoid a drive fight between the on-board and off board supplies. Eight of the FPGA board's supply (5v, 3.3v, 2.5v, 1.8v, 1.2v, MGT 2.5, MGT 1.5 and MGT 1.2v) can be individually disabled. In order to disable this regulator, a shunt must be mount on the jumper adjacent to the regulator. The table below shows the specific reference designators for each jumper.

High current screw terminals are connected to the primary power planes: GND, 12v, 5v, 3.3v, 1.8v and 1.2v which can be use to connect external power supplies. This can be seen according to Table 4 and 5 (Rice university wireless open Access research platform).

**5.1.5 Primary supplies TABLE 4. Identifying the screw terminals**

Plane	Regulator	Disable jumper	Screw terminal	Test point
12v	Ext	-	J35	TP14
5V	U18	J32	J38	TP12
3.3V	U16	J33	J37	TP11
2.5V	U8	J40	-	TP8
1.8V	U13	J42	J36	TP10
1.2V	U11	J43	J35	TP13
GND	-	-	J34	TP9

**5.1.6 MGT Supplies TABLE 5. MGT Supplies**

Plane	Regulator	Disable Jumper	Test Point
2.5	J3	J8	TP5
1.5	u2	J7	TP4
1.2	u1	J9	TP1

Table 4 and 5 show the primary power supply and the MGT supply (Rice university wireless open Access research platform).

## Targeting WARP Hardware

The targeted WARP hardware development efforts are:

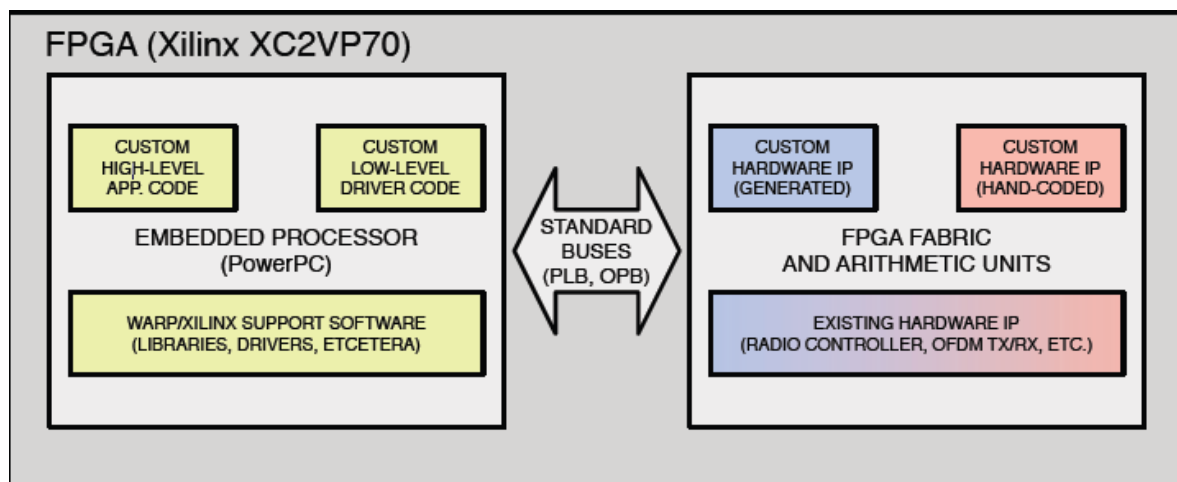
- Hand Coding Hardware Designs (Verilog, VHDL)
- Generating Hardware Designs (System Generator)
- Interfacing with existing Hardware IP

The targeted WARP Software development efforts are:

- Low level Driver development
- High Level Application Development
- Interfacing with existing Software

## Understanding the Development Environment

Figure 4 Development environment



## The Development tools

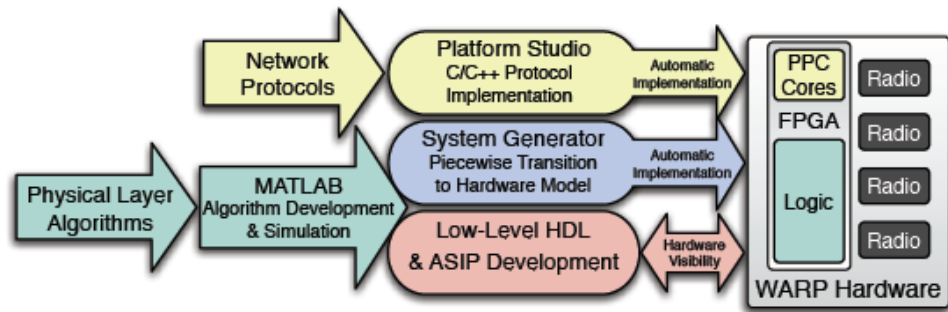


Figure 5: WARP development tools

## 6 FPGA CONFIGURATION

### 6.1 USB Configuration

Warp FPGA was configured by using the on-board USB configuration circuit method. The circuit act like a Xilinx platform USB cable; the drivers are included with the Xilinx ISE web pack. In order to use this interface, the Warp FPGA was connected directly to the computer using a standard USB cable. Windows will notify the user to search and install drivers, as long as windows bring up the pop up message. In this case windows should be allowed to complete the process before attempting to configure the FPGA. The Warp FPGA USB connector can be found at the back of the board as shown in the FIGURE 4 (Rice university wireless open Access research platform).

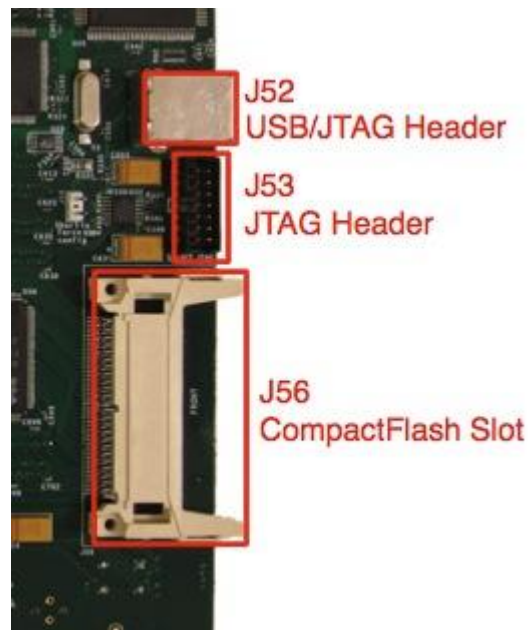


FIGURE 4: Warp FPGA USB connector

### 6.2 External JTAG cable

Not all PC's support USB configuration circuit on the FPGA board. This kind of situation occurs in windows server 2003 which is incompatible with Xilinx USB drivers. In this case, the FPGA can be configured using an external Xilinx JTAG cable.

The WARP FPGA board contains a standard JTAG connector which is compatible with both the Xilinx parallel IV and platform USB cables with a J53 component labeled 'SysACE JTAG' (Rice university wireless open Access research platform).

### 6.3 System ACE compact Flash

The WARP FPGA board also contains Xilinx's System ACE CompactFlash chip for managing the configuration process of FPGA. The SystemACE chip act as an interface between the FPGA and standard Compact Flash slot (Rice university wireless open Access research platform

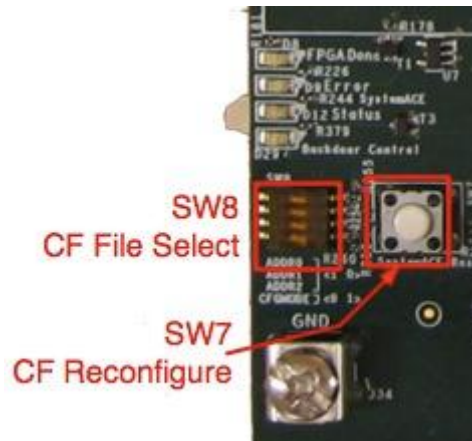


FIGURE 5: SystemACE compact Flash

The System ACE automatically configures the FPGA on when the power is up or when the reset button is pressed. Many configuration files can be stored on the CF card. The system ACE chip reads 3 bit address from a DIP switch to choose one of eight configuration files. The fourth position on the DIP switch, labeled CFGMODE was set to 1 (slide to the right). The image below shows the DIP switch position to select the desire configuration file (Rice university wireless open Access research platform).

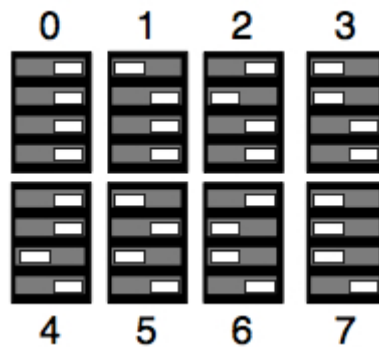


FIGURE 6: DIP switch position

From Figure 6, two LEDs indicate the status of the System ACE controller. The green LED (D12) glowed when the controller has successfully configured the FPGA using a bitstream on a CompactFlash card.

The red LED (D9) glowed when an error occurred during configuration from the CF card. Normally if error does not it will not glow otherwise it will. This error was due to format problem on the card or errors in the configuration files. When no CF file was inserted, neither light glowed. If the FPGA has been configured already, either through CF or JTAG, it will retain this configuration even after you remove the CF card.

Xilinx Impact was used to convert .bit files to System ACE .ace files and to construct the CF file card system.

## 7 WARP FPGA BOARD CLOCKING

### 7.1 On board oscillators

The FPGA board is made of two oscillator footprints for general clocks as shown in Figure 6. One 100MHz oscillator is mounted (component Y7) and one footprint is left empty (component Y9) by default for future customization. Both oscillator footprints are connected to global clock (GCLK) pins on the FPGA (Rice university wireless open Access research platform).

TABLE 6. On board Oscillations

<b>Clock</b>	<b>component</b>	<b>FPGA pin</b>
1000MHz	Y7	AM21
NM	Y9	AL20

### 7.2 Off-board clock sources

In FPGA board there is a header dedicated to off-board clock according to Figure 7. This header (component J25) is used by the WARP clock board. The header, which is connected to two global clock (GCLK) pairs on the FPGA (allowing for differential clocks), the 3.3v power plane and 8 general FPGA I/O (Rice university wireless open Access research platform).

TABLE 7. Off-board clock sources

<b>Header pin</b>	<b>FPGA pin</b>
3	AP 22
4	AP 21
7	AP 20
8	AP 20

### 7.3 System ACE CF Clocking

The System ACE controller needs a 33MHz clock which runs every time. The FPGA needs a copy of this clock in order to use the System Ace controller's microprocessor interface. A 33MHz oscillator (component Y6) is used on the FPGA board to supply this clock. The oscillator output is depicted and driven to both the FPGA and the System ACE CF controller (Rice university wireless open Access research platform).

TABLE 8. System ACE CF clocking

<b>clock</b>	<b>component</b>	<b>FPGA pin</b>
33MHz	Y4	AJ21

## 8 MGT Clocking

### 8.1 Warp FPGA Board MGTS

The virtex 4 FPGA has 10 pairs of differential multi gigabit transceivers. Each MGT is a full duplex transceiver supporting serial data rates up to 6.5 Gbps. The WARP FPGA board includes 8 MGT interfaces: 4 HSSDC2, 2 SATA and 2 SFP. The WARP FPGA virtex 4 FPGA are internally organized in two columns. Each column has two clock inputs and all the MGTs in the column can use either of those clocks

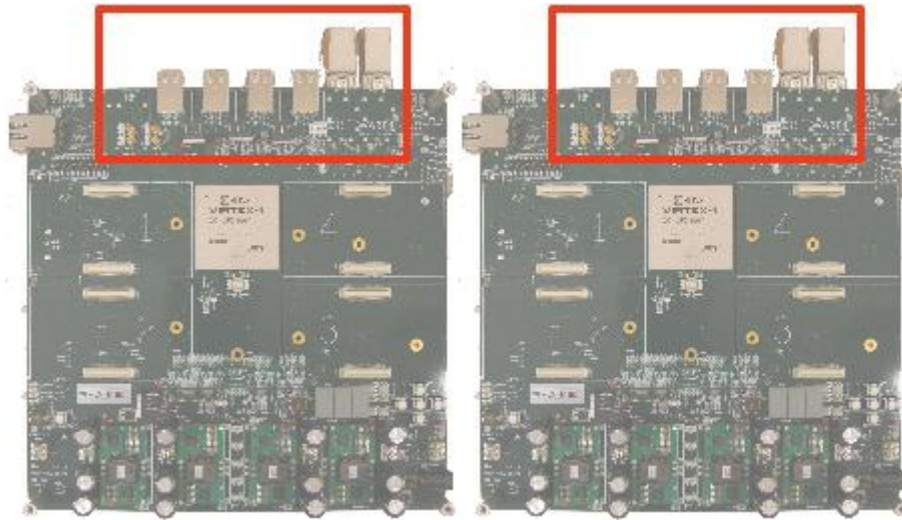


FIGURE 7. WARP FPGA board MGT

All the MGT connectors are located on north side of the FPGA board, both on the top and bottom (Rice university wireless open Access research platform).

### 8.2 MGT Interfaces

On the board, there three types of MGT connectors. Two small form factor pluggable (SFP) are connected to one column in the FPGA. The HSSDC2 and SATA interfaces are connected to the other column.

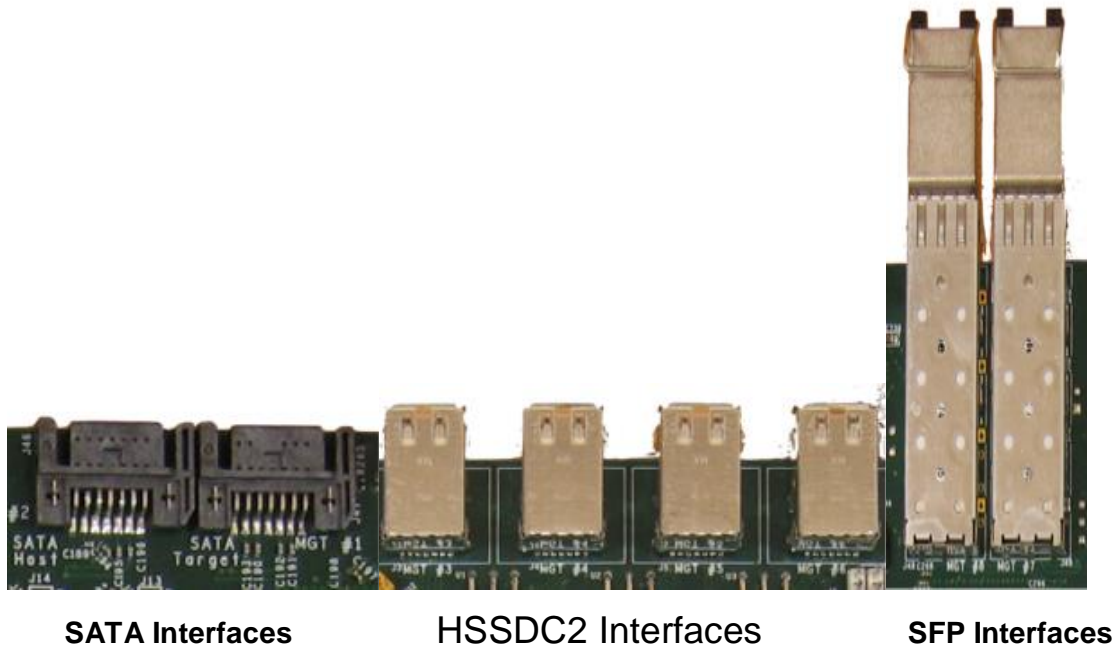


FIGURE 8. MGT Interface

Normally the MGT interfaces are labeled from “MGT 1” to “MGT 8” on the WARP FPGA board. The mapping of each interface to the corresponding MGT in the FPGA can be shown in the table below (Rice university wireless open Access research platform).

TABLE 9. Management Interface

MGT	Type	Connector	MGT Tile	Column	LOC Constraint	TXP	TXN	RXP	RXN	Ideal Clocks
1	SATA Target	J47	112B	1	GT11_X1Y4	P1	R1	U1	V1	MGTCLK_110 OR MGTCLK_113
2	SATA Host	J46	112A	1	GT11_X1Y5	M1	N1	J1	K1	MGTCLK_110 OR MGTCLK_113
3	HSSDC2	J3	113B	1	GT11_X1Y6	A4	A3	C1	D1	MGTCLK_110 OR MGTCLK_113
4	HSSDC2	J4	113A	1	GT11_X1Y7	A6	A5	A9	A8	MGTCLK_110 OR MGTCLK_113
5	HSSDC2	J5	114B	1	GT11_X1Y8	A14	A13	A11	A10	MGTCLK_110 OR MGTCLK_113
6	HSSDC2	J6	114A	1	GT11_X1Y9	A16	A15	A19	A18	MGTCLK_110 OR MGTCLK_113
7	SFP#1	J49	102A	0	GT11_X0Y7	A34	A35	A31	A32	MGTCLK_102 OR MGTCLK_105
8	SFP#2	J48	102B	0	GT11_X0Y6	A36	A37	C39	D39	MGTCLK_102 OR MGTCLK_105

## 8.1 MGT Clocking

The WARP FPGA board provides very flexible MGT clocking. The virtex 4 FPGA organizes the MGTs into two columns; each of the columns provides two clocks inputs. An MGT can use either clock driven into its column.

The FPGA board provides five MGT clocks sources- four oscillators and one off board interface. Two oscillators are installed by default, the remaining oscillator footprints can be populated as needed to support custom applications.

One oscillator is connected directly to an FPGA MGT clock input. The remaining clock sources (four oscillators and the off-board interface ) are connected to the FPGA through a flexible multiplexer network. This network allows the user to assign any of the four clock sources to any of the three FPGA MGT CLOCK inputs. It also provides an off board clock output which can be connected to another FPGA board, allowing multiple FPGA boards to share an MGT reference clock.

Details about WARP FPGA board's MGT clocking system can be found in Figure 9.

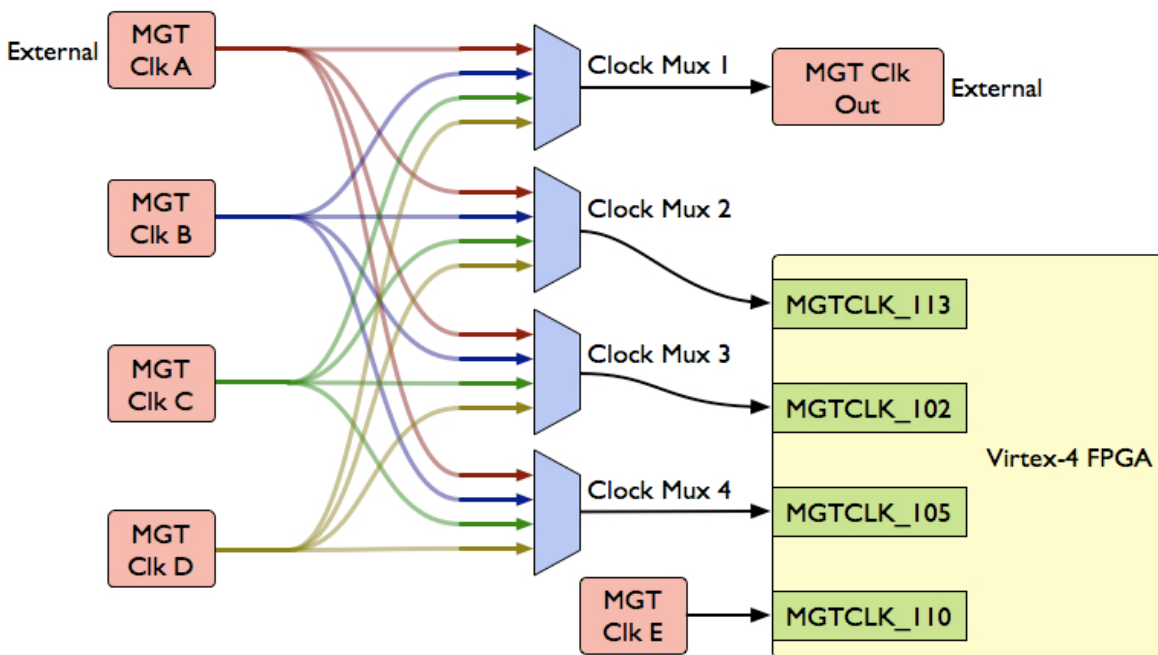


FIGURE 9. WARP FPGA board MGT clocking

## 8.2 FPGA MGT Clock Input

TABLE 10 FPGA MGT clock input

clock input	Tile	Column	GT11CLK LOC	P pin	N pin	Source
MGTCLK_102	102	0	GT11CLK_X0Y3	F39	G39	Mux 3
MGTCLK_105	105	0	GT11CLK_X0Y1	AW34	AW33	Mux 4
MGTCLK_110	110	1	GT11CLK_X1Y1	AW6	AW7	MGT Clk E (Y8)
MGTCLK_113	113	1	GT11CLK_X1Y3	F1	G1	Mux 2

The constraint of the MGT clocking can be found in Figure 10

```
#FPGA Board v2.2 Clock Constraints
#
# The constraints using the onboard 100MHz oscillator
Net sys_clk_pin LOC=AM21;
Net sys_clk_pin IOSTANDARD = LVTTTL;
Net sys_clk_pin TNM_NET = sys_clk_pin;
TIMESPEC TS_sys_clk_pin = PERIOD sys_clk_pin 10000 ps;
#
# The constraints using the Clock Board generated 40MHz
# clock for the design. NOTE: The clock_board_configurator
# must be instantiated to configure the clock board
Net sys_clk_pin LOC=AN20;
Net sys_clk_pin IOSTANDARD = LVTTTL;
Net sys_clk_pin TNM_NET = sys_clk_pin;
TIMESPEC TS_sys_clk_pin = PERIOD sys_clk_pin 25000 ps;
```

FIGURE 10 MGT CLK

## 9 Warp FPGA Board Memory

### 9.1 On-chip Memory

The virtex 4 FX100 FPGA has 376 18kb RAM blocks (6.7Mb total) on the chip. Xilinx has an alternative method to make logic slices as RAM, this is called distributed memory. the FX100 has up to 659kb of distributed memory (warp.rice.edu).

### 9.2 DDR2 SO- DIMM

There is DDR2 SO-DIMM slot in the Warp FPGA board v2.2. This connector is routed to dedicated FPGA I/O and clocking resources and support up to 2GB modules (warp.rice.edu).

FIGURE 11 FPGA SO-DIMM slots



FPGA Board SO-DIMM slots has 2GB SO-DIMM slot installed on it according to Figure 11.

Using the SO-DIMM, the DDR2 memory controller must be included by the user of FPGA Design. Xilinx has a high performance controller as part of their Multi port Memory Controller (MPMC).

In order to generate the MPMC designs, Base System Builder and XBD file in the documentation is recommended because there is large number of pins and parameters involved in instantiating the MPMC in a design.

The BSB (Base System builder) has 2 GB memory. This occupies half of the address space of the PowerPC core. If address space is needed for other peripherals, the address space of the MPMC must reduce without changing any parameters of the IP core.

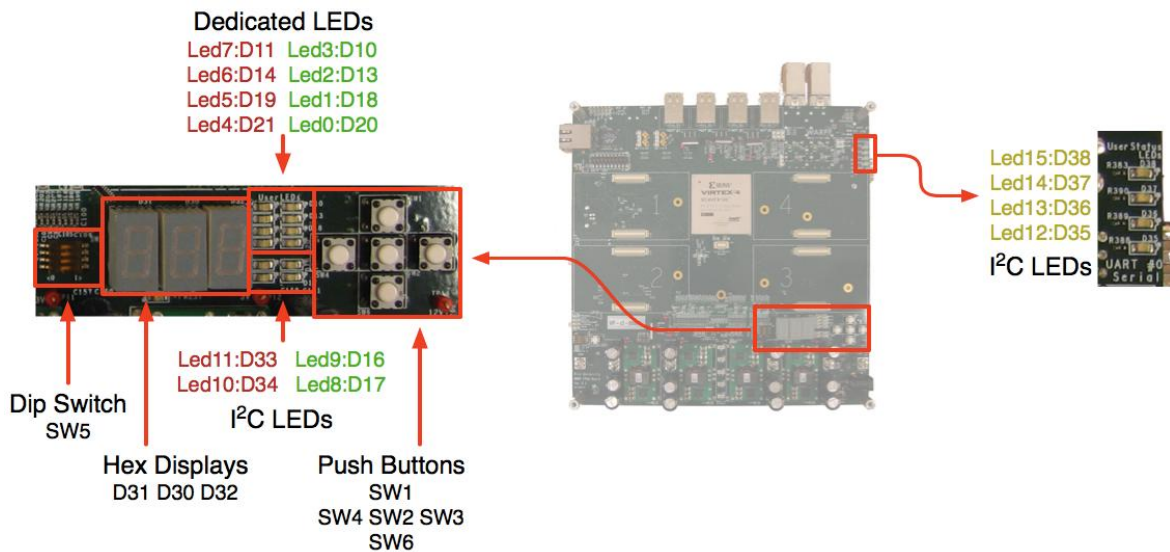
FIGURE 12 Instantiating the MPMC in a Design using BSB

```

BEGIN mpmc
  PARAMETER INSTANCE = DDR2_SDRAM_2GB
  PARAMETER HW_VER = 4.03.a
  PARAMETER C_NUM_PORTS = 2
  PARAMETER C_MEM_PARTNO = MT16HTF25664H-667
  PARAMETER C_MEM_TYPE = DDR2
  PARAMETER C_NUM_IDELAYCTRL = 4
  PARAMETER C_IDELAYCTRL_LOC = IDELAYCTRL_X0Y0-IDELAYCTRL_X0Y1-
IDELAYCTRL_X2Y1-IDELAYCTRL_X2Y0
  PARAMETER C_MEM_DQS_WIDTH = 8
  PARAMETER C_MEM_DM_WIDTH = 8
  PARAMETER C_MEM_ADDR_WIDTH = 14
  PARAMETER C_MEM_BANKADDR_WIDTH = 3
  PARAMETER C_PIM0_BASETYPE = 2
  PARAMETER C_PIM1_BASETYPE = 2
  PARAMETER C_MPMC_CLK0_PERIOD_PS = 6250
  PARAMETER C_MPMC_BASEADDR = 0x00000000
  PARAMETER C_MPMC_HIGHADDR = 0x7fffffff
  BUS_INTERFACE SPLB0 = ppc405_0_ip1b1
  BUS_INTERFACE SPLB1 = ppc405_0_dp1b1
  PORT DDR2_Addr = fpga_0_DDR2_SDRAM_2GB_DDR2_Addr
  PORT DDR2_BankAddr = fpga_0_DDR2_SDRAM_2GB_DDR2_BankAddr
  PORT DDR2_CAS_n = fpga_0_DDR2_SDRAM_2GB_DDR2_CAS_n
  PORT DDR2_CE = fpga_0_DDR2_SDRAM_2GB_DDR2_CE
  PORT DDR2_CS_n = fpga_0_DDR2_SDRAM_2GB_DDR2_CS_n
  PORT DDR2_RAS_n = fpga_0_DDR2_SDRAM_2GB_DDR2_RAS_n
  PORT DDR2_WE_n = fpga_0_DDR2_SDRAM_2GB_DDR2_WE_n
  PORT DDR2_DM = fpga_0_DDR2_SDRAM_2GB_DDR2_DM
  PORT DDR2_DQS = fpga_0_DDR2_SDRAM_2GB_DDR2_DQS
  PORT DDR2_DQS_n = fpga_0_DDR2_SDRAM_2GB_DDR2_DQS_n
  PORT DDR2_DQ = fpga_0_DDR2_SDRAM_2GB_DDR2_DQ
  PORT DDR2_Clk = fpga_0_DDR2_SDRAM_2GB_DDR2_Clk
  PORT DDR2_Clk_n = fpga_0_DDR2_SDRAM_2GB_DDR2_Clk_n
  PORT DDR2_ODT = fpga_0_DDR2_SDRAM_2GB_DDR2_ODT
  PORT MPMC_Clk0 = proc_clk_s
  PORT MPMC_Clk90 = DDR2_SDRAM_2GB_mpmc_clk_90_s
  PORT MPMC_Clk_200MHz = clk_200mhz_s
  PORT MPMC_Rst = sys_periph_reset
END

```

FIGURE 12 Warp FPGA Board Users I/O



In the FPGA Board there is a variety of interactive I/O devices, known as user I/O. These interfaces aid with observing and debugging custom designs in hardware.

### 9.3 Push Buttons

There are five buttons found on the FPGA Board and they are arranged in cross. They are open connections with external pull down resistors. The FPGA will experience logic high when a button is press and logic low when release.

### 9.4 DIP-switch

The function of the 4 positions DIP-switch is to drive four dedicated inputs on the FPGA. Sliding a switch to the left drives the input to logic low and to the right, drive the input to the logic high.

### 9.5 LEDs

There are 8 LEDs in total and they can control by the user designs. Out of these LEDs, eight are connected to dedicated FPGA I/O pins- four green, four red, according to Figure 12 above.

### 9.6 Hex-Display

In the FPGA Board, there are 7 segments displays. All the three are connected to the FPGA through 1C I/O input expanders (U36 and U39). While designing, 12C master as use to control the displays. Each display includes eight LED elements- seven forming segments of numerical digit and acting as small decimal point.

FIGURE 13 Constraints for User I/O device

```

#FPGA Board v2.2 I/O constraints for User I/O Devices
#
# 8 LEDs directly controlled using the FPGA I/O pins (D10, D11, D13,
# D14, D18, D19, D20, D21)
NET LED<0> LOC = N24 | IOSTANDARD = LVCMOS25; #RED D11
NET LED<1> LOC = N20 | IOSTANDARD = LVCMOS25; #RED D14
NET LED<2> LOC = L18 | IOSTANDARD = LVCMOS25; #RED D19
NET LED<3> LOC = N18 | IOSTANDARD = LVCMOS25; #RED D21
NET LED<4> LOC = M18 | IOSTANDARD = LVCMOS25; #GREEN D10
NET LED<5> LOC = M25 | IOSTANDARD = LVCMOS25; #GREEN D13
NET LED<6> LOC = N19 | IOSTANDARD = LVCMOS25; #GREEN D18
NET LED<7> LOC = P19 | IOSTANDARD = LVCMOS25; #GREEN D20
#
# 5 pushbuttons arranged a cross orientation.
NET PUSHB_CENTER LOC = L23 | IOSTANDARD = LVCMOS25;
NET PUSHB_DOWN LOC = M21 | IOSTANDARD = LVCMOS25;
NET PUSHB_LEFT LOC = N22 | IOSTANDARD = LVCMOS25;
NET PUSHB_RIGHT LOC = M23 | IOSTANDARD = LVCMOS25;
NET PUSHB_UP LOC = N23 | IOSTANDARD = LVCMOS25;
#
# 4-bit DIP Switch (SW5)
NET DIPSW<0> LOC = M17 | IOSTANDARD = LVCMOS25;
NET DIPSW<1> LOC = R18 | IOSTANDARD = LVCMOS25;
NET DIPSW<2> LOC = P17 | IOSTANDARD = LVCMOS25;
NET DIPSW<3> LOC = M16 | IOSTANDARD = LVCMOS25;
#
# 2 IO Expanders that control the three hex displays (D30, D31, D32) and
# and 8 additional LEDs (D16, D17, D33, D34, D35, D36, D37, D38)
NET HEX_SDA LOC = AL18 | IOSTANDARD = LVCMOS33;
NET HEX_SCL LOC = AK17 | IOSTANDARD = LVCMOS33;

```

## 10 Procedure

When carrying out the practical part of the project in the wireless Laboratory, the PC was running Windows OS and the Xilinx Design Suit was installed on the PC. Necessary cable connections were made on the Warp FPGA Board by putting the power cable in to a socket which power on the board by showing red light when switched on. Follow by the USB cable connected to the WARP FPGA Board. Thereafter, the serial cable was connected which goes from the serial port of the Board to the serial port of the PC.

FIGURE 14 Power cable on the Board

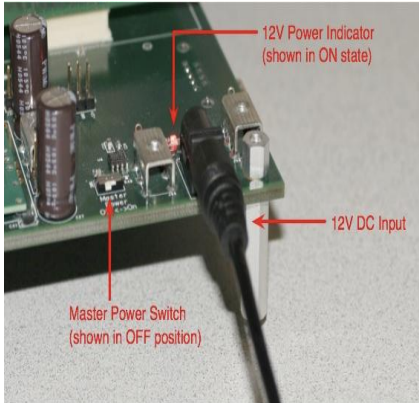


FIGURE 15 USB connection

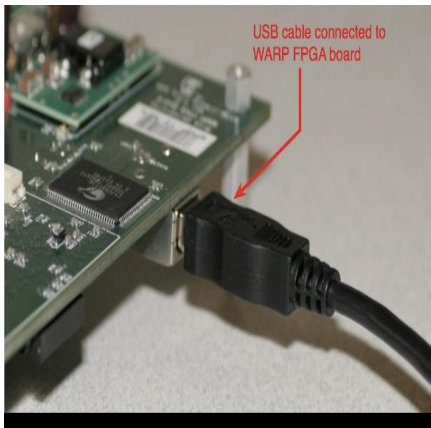


FIGURE 16 Serial cable connection

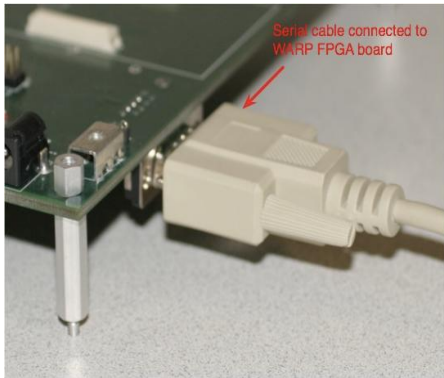


Figure 14, 15 and 16 show the connection of power cable, USB cable and Serial cable on WARP FPGA Board. The essence of the USB connection is allow communication between the board and the Board.

The next step was creating a connection between the WARP FPGA Board and the PC on which Xilinx tool is being run. Being the first time that FPGA Board was connected to the PC, there was a new hardware found pop message which appeared on the screen, update was allowed and the connection was completed.

FIGURE 17



Figure 17 shows the completion of USB connection. Sometimes the process might need to be repeated several times before the installation is completed.

In order to verify the ability communicate with the WARP FPGA Board over the USB link, an IMPACT software application which supports communication between a computer running IMPACT and the devices on WARP FPGA Board. Establishing communication through this link is the most important step in utilizing the WARP hardware. In Image 1, the Xilinx IMPACT application was opened.

IMAGE 1

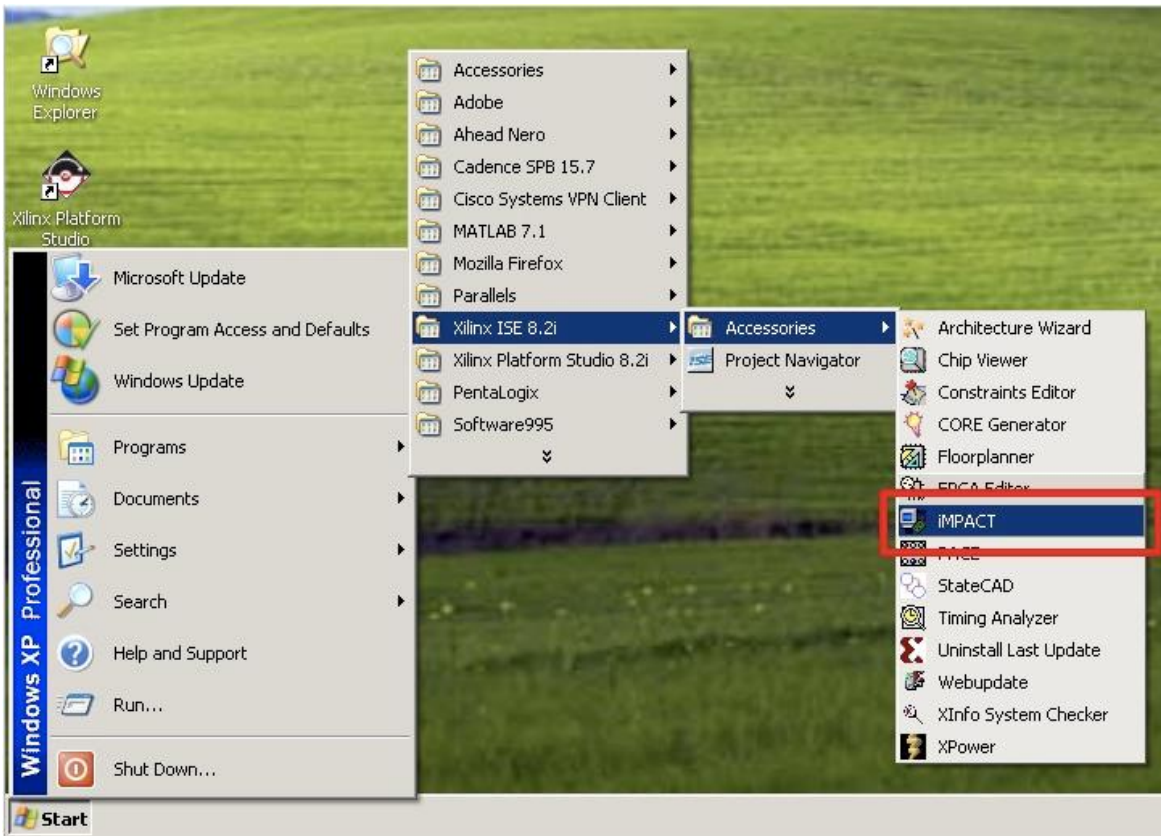


IMAGE 2 new project was created and pressed OK button

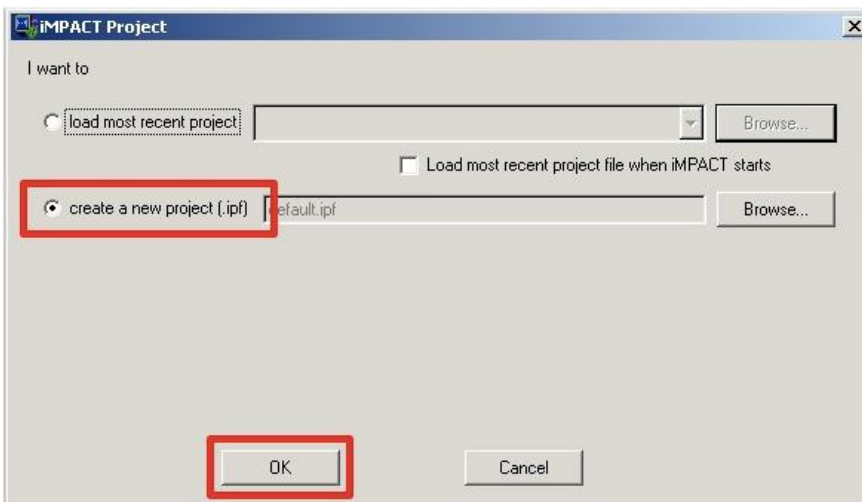


IMAGE 3 configure devices option was chosen and pressed Finish

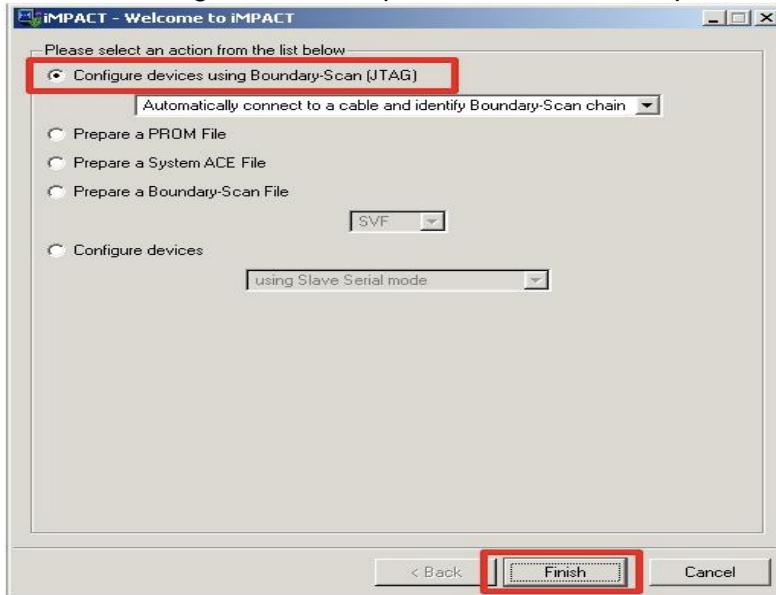


IMAGE 4 USB firmware on WARP FPGA was updated

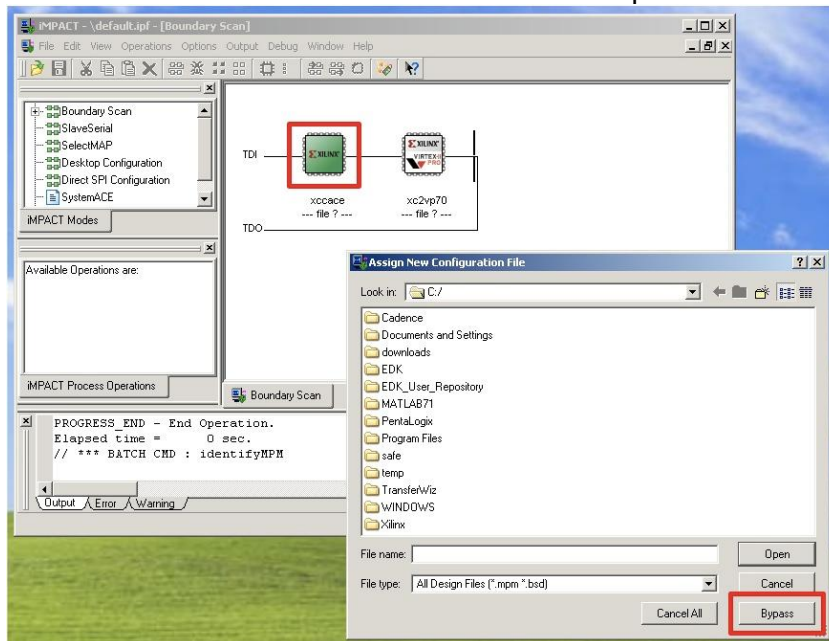


IMAGE 5 the bypass was clicked to configure the FPGA device

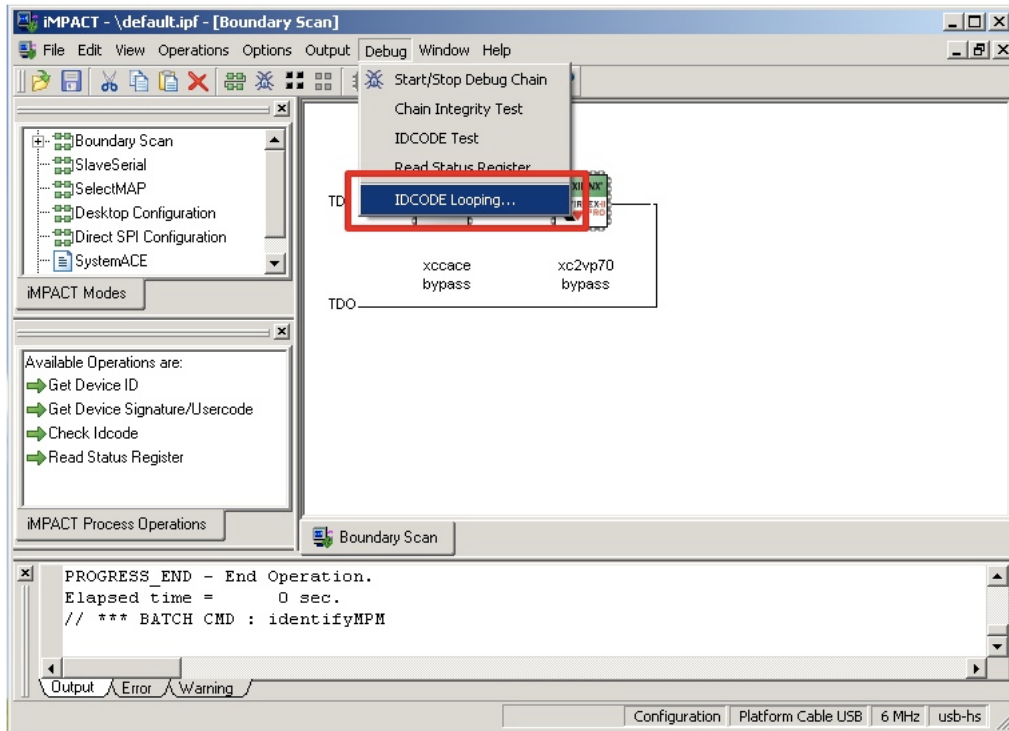


IMAGE 6 Dialog box showing the number of iterations

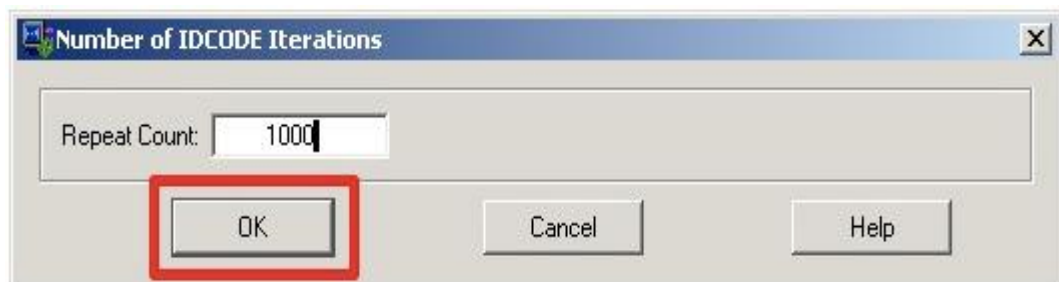
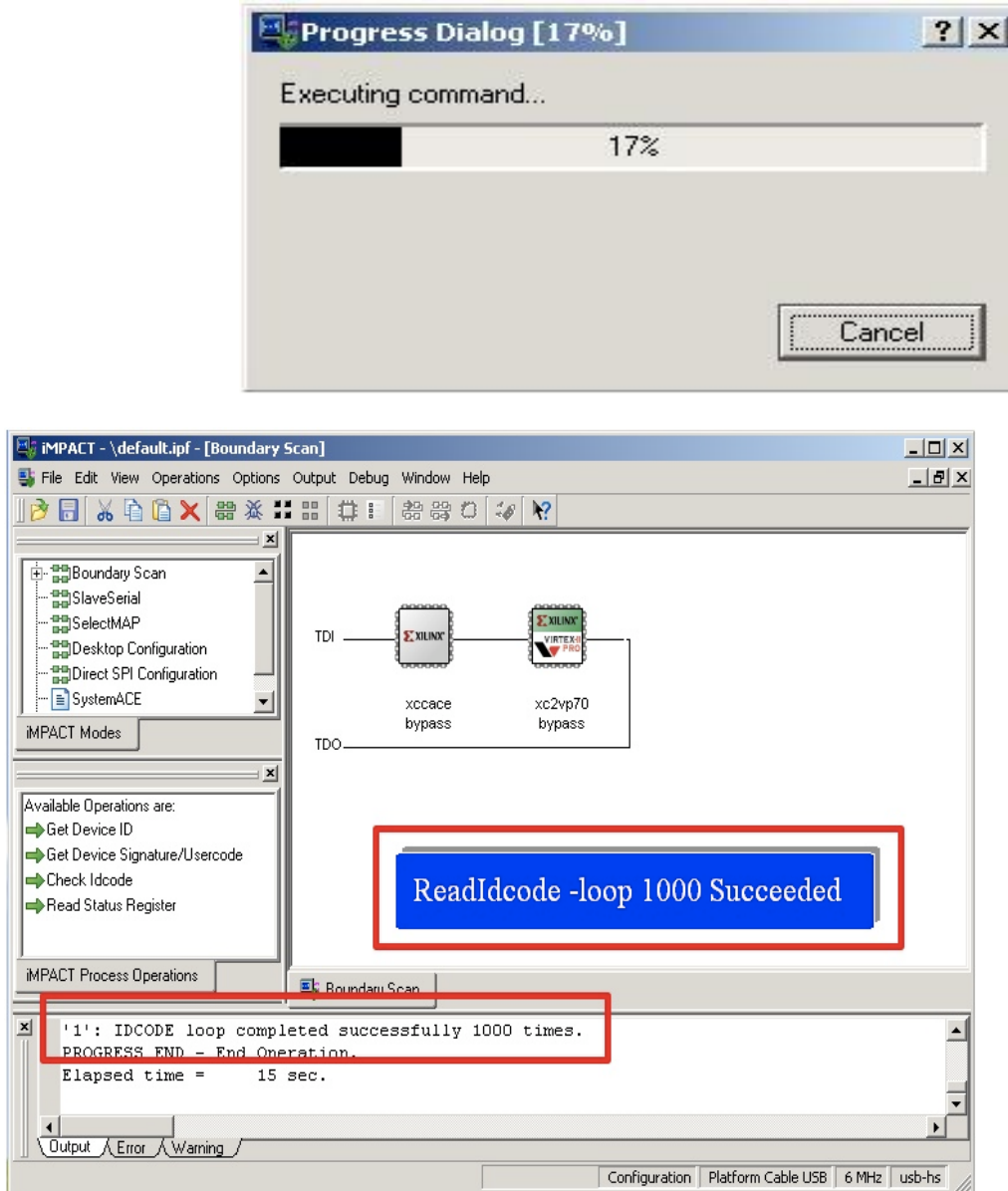


IMAGE 7 progress dialog box and the completion of the communication link



## 10.1 Software

### Xilinx Platform Studio

Xilinx platform studio is a software application that is use to develop FPGA based hardware systems from a variety of intellectual property (IP) cores. It allows the integration of hardware (compiled into FPGA fabric) with software (running on embedded processors within the FPGA) to deliver true system on a chip functionality (warp.rice.edu).

### 10.2 Requirement and Set up

The requirements for setting up a Base system Builder in order to make the application development on XPS are PC, DB9 serial port, USB port, a Warp FPGA board with a power supply.

For the software requirements, Xilinx tools (ISE 10.1 is required but 13.1 was used with EDK 10.1).

WARP FPGA EDK was downloaded and the WARP FPGA board serial and USB configuration interfaces were connected to the PC.

### Base System Builder FPGA board 2.2

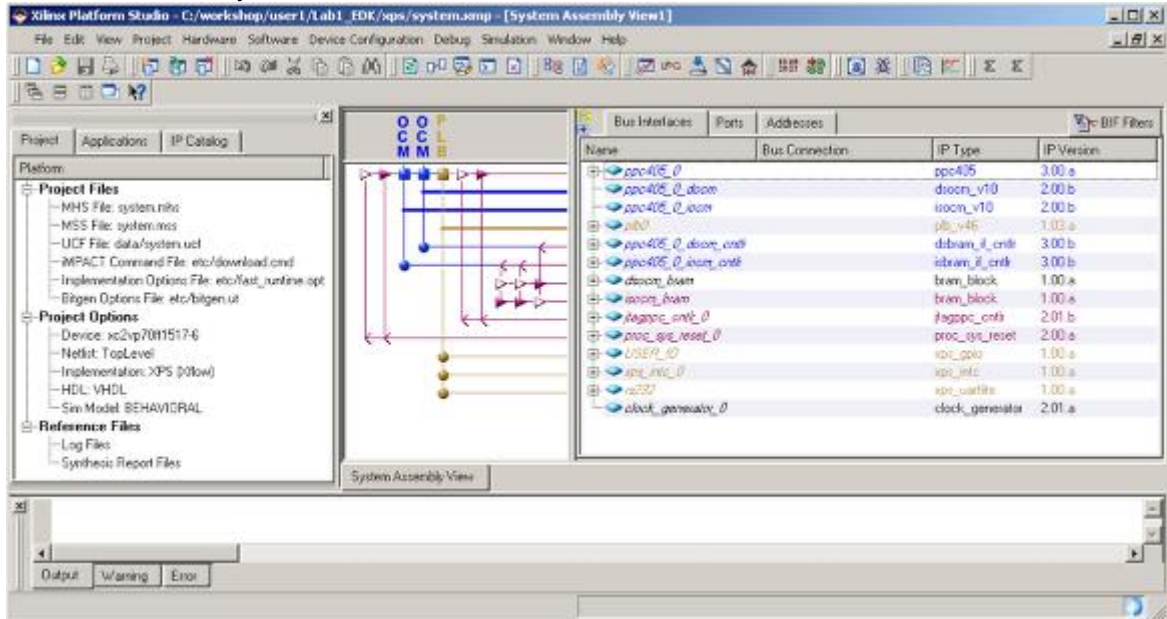
IMAGE 8 Launching the XPS



In order to create a simple hardware/software platform using Base System Builder, launching of XPS studio was done and lots of processes was undergone according to IMAGE 8.

The essence of the Base Builder is to be able to configure the WARP FPGA board components and a running application on the board. According to IMAGE 8, after the whole process of setting up Base System Builder, XPS opens the resulting project.

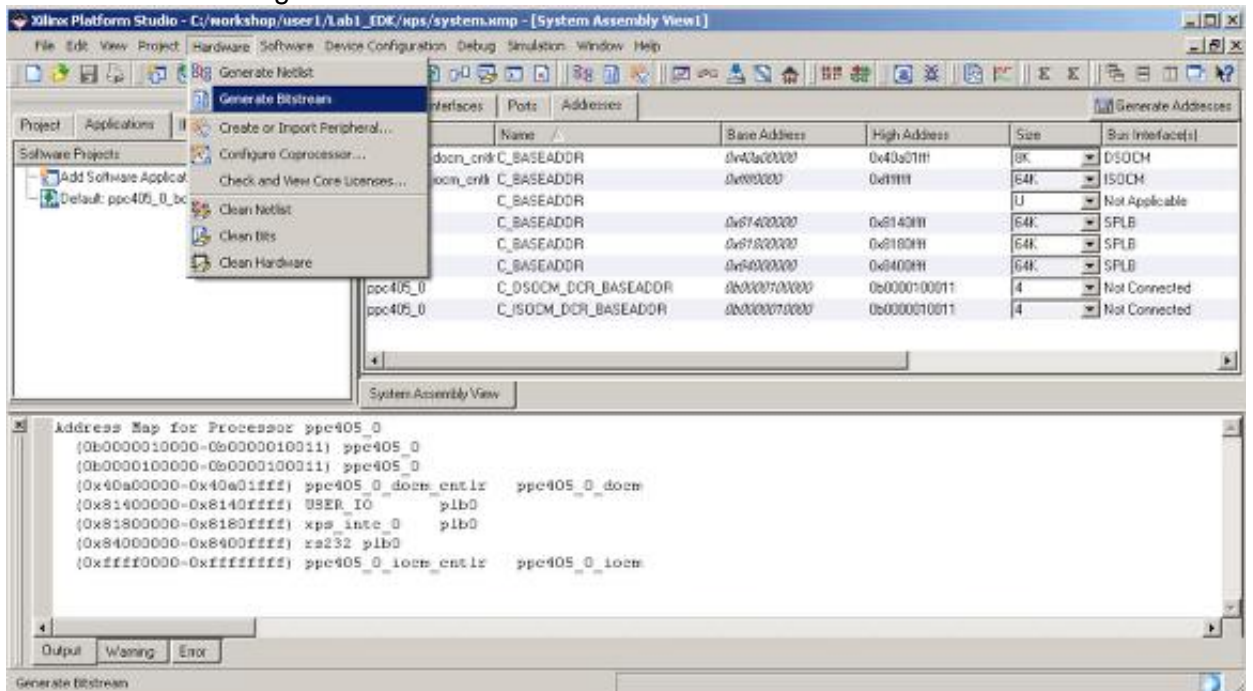
## IMAGE 8 Base System Builder



## Implementing the Hardware Platform

In order to implement the Hardware platform, from the hardware menu, generate bit stream was selected, this will initiate hardware synthesis process. The EDK used tools from Xilinx ISE to synthesize the VHDL description of each peripheral, apply timing and I/O constraints, map the design into hardware and generate an FPGA configuration file (the bit stream) according to IMAGE 9 (warp.rice.edu).

## IMAGE 9 Bit stream generation



Finally the log viewer showed that bit stream generation is complete. It means that the hardware is synthesized and ready for use. At the end of the process done! appeared then generation is complete, meaning that the project and application created successfully according to IMAGE 10 (warp.rice.edu).

### IMAGE 10 Bit stream generation

The screenshot displays the Xilinx Platform Studio interface. The top window, titled "System Assembly View", shows a table of bus interfaces. The bottom window shows the output log with the following text:

```

ccmp
jttagppo_cntlr_0/jtagppo_cntlr_0/auto_ppc_connectivity.auto_PPC405.PPC_auto_10
is not connected to the output of a DCM driven clock buffer.
DRC detected 0 errors and 4 warnings. Please see the previously displayed
individual error or warning messages for more details.
Creating bit map...
Saving bit stream in "system.bit".
Bitstream generation is complete.
Done!

```

The output log also includes buttons for "Output", "Warning", and "Error".

Instance	Name	Base Address	High Address	Size	Bus Interface(s)
ppc405_0_dcm_cntlr	C_BASEADDR	0x400000	0x4001ff	8K	D50CM
ppc405_0_ioem_cntlr	C_BASEADDR	0x600000	0x600fff	54K	IS0CM
pb0	C_BASEADDR				Not Applicable
USER_ID	C_BASEADDR	0x0140000	0x0140fff	54K	SPLB
ppc_intc_0	C_BASEADDR	0x0180000	0x0180fff	54K	SPLB
rs232	C_BASEADDR	0x0400000	0x0400fff	54K	SPLB
ppc405_0	C_D50CM_DCR_BASEADDR	0x000070000	0x000070001	4	Not Connected
ppc405_0	C_IS0CM_DCR_BASEADDR	0x000070000	0x000070001	4	Not Connected

## **11 Result**

After the whole experiment in Savonia wireless lab, it was noticed that there was a light beep from the led part of the WARP FPGA board which means that signal was being transferred through the USB part of the connection of the board. There were two antennas there connected to the board and they gave a clear signal to the transmission by avoiding noise.

Further research can be done using a virtual simulation using WARP MATLAB in combination with MATLAB; this is to verify the result virtually. With MATLAB, interacting with the WARP nodes directly from the MATLAB workspace and signals generated in MATLAB can be transmitted in real time over the air using WARP nodes ([warp.rice.edu](http://warp.rice.edu)).

## **12 Conclusion**

In the global world today, the implementation of software radio application is being used in some organizations like in the Army and as time goes on in the future, the application will take over the communication society because of its vast advantages. The problem of commutation can be solve with the introduction of Software radio, since it has the ability of reconfiguring itself no matter the environment.

### 13 REFERENCES

<http://warp.rice.edu/trac/wiki/Exercises/FPGABoardIntro/BoardConnections>

[http://warp.rice.edu/svn/WARP/Documentation/Tutorials/XPS\\_Intro/html/XPS\\_Intro.html](http://warp.rice.edu/svn/WARP/Documentation/Tutorials/XPS_Intro/html/XPS_Intro.html)

[http://warp.rice.edu/svn/WARP/Documentation/Tutorials/XPS\\_Intro/html/sections/Base\\_System\\_Builder\\_-\\_FPGA\\_Board\\_v1.html](http://warp.rice.edu/svn/WARP/Documentation/Tutorials/XPS_Intro/html/sections/Base_System_Builder_-_FPGA_Board_v1.html)

[http://warp.rice.edu/svn/WARP/Documentation/Tutorials/XPS\\_Intro/html/sections/XPS\\_Intro\\_-\\_Implementing\\_the\\_hardware\\_platform.html](http://warp.rice.edu/svn/WARP/Documentation/Tutorials/XPS_Intro/html/sections/XPS_Intro_-_Implementing_the_hardware_platform.html)

[http://warp.rice.edu/svn/WARP/Documentation/Tutorials/XPS\\_Intro/html/sections/XPS\\_Intro\\_-\\_Implementing\\_the\\_software\\_design.html](http://warp.rice.edu/svn/WARP/Documentation/Tutorials/XPS_Intro/html/sections/XPS_Intro_-_Implementing_the_software_design.html)

[http://warp.rice.edu/trac/wiki/HardwareUsersGuides/FPGABoard\\_v2.2/FPGA](http://warp.rice.edu/trac/wiki/HardwareUsersGuides/FPGABoard_v2.2/FPGA)

[http://warp.rice.edu/trac/wiki/HardwareUsersGuides/FPGABoard\\_v2.2/Power](http://warp.rice.edu/trac/wiki/HardwareUsersGuides/FPGABoard_v2.2/Power)

[http://warp.rice.edu/trac/wiki/HardwareUsersGuides/FPGABoard\\_v2.2/FPGAConfig](http://warp.rice.edu/trac/wiki/HardwareUsersGuides/FPGABoard_v2.2/FPGAConfig)

[http://warp.rice.edu/trac/wiki/HardwareUsersGuides/FPGABoard\\_v2.2/Clocking](http://warp.rice.edu/trac/wiki/HardwareUsersGuides/FPGABoard_v2.2/Clocking)

[http://warp.rice.edu/trac/wiki/HardwareUsersGuides/FPGABoard\\_v2.2/Memory](http://warp.rice.edu/trac/wiki/HardwareUsersGuides/FPGABoard_v2.2/Memory)

[http://warp.rice.edu/trac/wiki/HardwareUsersGuides/FPGABoard\\_v2.2/DaughtercardSlots](http://warp.rice.edu/trac/wiki/HardwareUsersGuides/FPGABoard_v2.2/DaughtercardSlots)

[http://warp.rice.edu/trac/wiki/HardwareUsersGuides/FPGABoard\\_v2.2/UserIO](http://warp.rice.edu/trac/wiki/HardwareUsersGuides/FPGABoard_v2.2/UserIO)