**FPGA Demo Board Tutorial**

**written by**

**Jason P. Irwin**

This tutorial takes the student through the process of implementing a digital design into a Xilinx FPGA chip. In this tutorial two completely separate designs are considered. The first is a design that simply passes inputs from a switch to outputs that are displayed by LEDs. The second is an eight bit counter design.

The steps in this process include synthesis, creation of a bit file(Xilinx format), and verification of design. The first section of this tutorial is devoted to properly setting up your UNIX account to use the Synopsys design tools. The second section takes you through the process of analyzing and compiling your design on the Sun platform to produce a Xilinx Netlist Format (XNF) file. In the third section of this tutorial the XACT design tools on the PC platform are used to produce a downloadable file. The fourth section of this tutorial describes how to download and verify the design using the Xilinx FPGA Demo Board. The fifth and final section describes how to correctly connect and interface the demo board to the PC.

The appendix to this document contains several resources that will be useful when you try to download your own design to the demo board. Appendix A gives a schematic and detailed description of the XC3020A pin layout on the demo board. Appendix B describes how to construct and use a constraint (CST) file. Appendix C describes the compile script (SCRIPT) file and what steps are done in the compilation process, and finally Appendix D is a list of resources that are available to help in the completion of this tutorial and any other design you might try.

## Section 1 - Unix account setup

In order to be able to complete this tutorial your UNIX account needs to be setup according to the following steps.

1. In order for the Synopsys tools to function properly, the following lines need to be added to your **.cshrc** file.

   *Set path = ($path /usr/synopsys/sparc/syn/bin)*
   *source ~/.synop*

2. Obtain a copy of the **.synop**.

   *cp ~jasoni/demoboard/.synop .*

3. Source your .cshrc file for the changes to take effect.

   *source .cshrc*

## Section 2 - Creation of Xilinx Netlist Format (XNF) files on the SUN Platform

In this section you will be creating directories, copying the design files to be analyzed and compiled, and creating a XNF file.

As stated in the introduction to this tutorial there are two different designs to use. The first is a simple combinational design that routes the switches contained on the demo board to the LED bank on the demo board. When a switch is active the corresponding LED will be on and when the switch is off the corresponding LED will be off. The second design is a simple synchronous eight bit counter that displays the current count on the LEDs.

### LED-Switch Design Example

To complete this design example you need to set up your account according to the following steps:

1. Create a design directory.

   *mkdir ledswitch*

2. Change to the design directory and copy the following files.

   *cd ledswitch*
   *cp ~jasoni/demoboard/xc3000/.synopsys_dc.setup .*
   *cp ~jasoni/demoboard/xc3000/ledswitch/ledswitch.vhd .*
   *cp ~jasoni/demoboard/xc3000/ledswitch/ledswitch.cst .*
   *cp ~jasoni/demoboard/xc3000/ledswitch/ledswitch.script .*

3. Create a work directory for temporary storage in the design directory.

   *mkdir WORK*

The next step is to familiarize yourself with the VHDL description of the digital design. The code for the design is given below for easy reference, However since it is such an easy example no explanation is included here.

```
-- Led Switch - Behavioral Model
-- XSI v3.2
-- filename -> ledswitch.vhd

Library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.STD_LOGIC_UNSIGNED.all;
```

```
entity ledswitch is
    port (
        CIN : in STD_LOGIC_VECTOR (7 downto 0);
        COUT: out STD_LOGIC_VECTOR (7 downto 0));
end ledswitch;

architecture BEHAVIORAL of ledswitch is

begin

COUT <= NOT(CIN);

end BEHAVIORAL;
```

It is now time to analyze and compile the design using the Synopsys Design Analyzer by completing the following steps:

1. Make sure you are in the correct design directory (ledswitch) and invoke the Synopsys Design Analyzer.

   *design_analyzer*

2. This will cause the Design Analyzer window to pop up in your windowing environment. From the Setup menu activate the command window so that you can watch the compile script as it executes.

   **Setup -> Command Window…**

3. After the command window is active, the next step is to execute the compile script to analyze the design. From the Setup menu choose the execute script option and select the **ledswitch.script** file with the left mouse button and choose **OK**.

   **Setup -> Execute Script…**
   select **ledswitch.script** in the file name window with left mouse button
   select **OK** in the file name window

4. Observe the compilation steps as the compile scripts executes in the command window. After the compilation is complete a window will pop up and ask if you want to quit the design analyzer. Choose OK.

   Select **OK** in the quit design analyzer window

5. Convert the **ledswitch.sxnf** file to a **xnf** type file. In order to do this enter the following command at the UNIX shell prompt.

   *syn2xnf ledswitch*

6. Copy the following files to a floppy disk from your UNIX account using the **mtools**. If you are not familiar with the **mtools** look at the manual by typing *man mtools*.

   **ledswitch.xnf**
   **ledswitch.cst**

You will use these files in the next section to create the **ledswitch.bit** file on the PC platform in the Windows environment.

## Eight Bit Counter Design Example

In this design example you will go through the same steps as in the LED switch design to produce a Xilinx netlist format file for the counter design. Just as the name implies, the eight bit counter design is a counter that counts from 0 to 255.

The next step is to acquaint yourself with the VHDL design of the counter. The code is given below for easy reference.

```
-- Count8 - Behavioral Model
-- 8-bit Counter with Enable and Clear
-- XSI v3.2
-- filename -> count8.vhd

Library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.STD_LOGIC_UNSIGNED.all;

entity count8 is
    port (
        CLOCK, CLEAR, ENABLE: in STD_LOGIC;
        COUT: out STD_LOGIC_VECTOR (7 downto 0));
end count8;

architecture BEHAVIORAL of count8 is
signal QOUT: STD_LOGIC_VECTOR (7 downto 0);
signal CLK_INT : STD_LOGIC;

component GCLK_F
port( I : in STD_LOGIC;
      O : out STD_LOGIC);
```

```
end component;

begin

COUT <= NOT(QOUT);

  clk_buf : GCLK_F
    port map(
      I => CLOCK,
     O => CLK_INT);

   process (CLEAR, CLK_INT, ENABLE)
      begin
           if (CLEAR = '1') then
              QOUT <= "00000000";
           elsif (CLK_INT'event and CLK_INT='1') then
               if (ENABLE = '1') then
                   QOUT <= QOUT + "00000001";
               end if;
            end if;
    end process;
end BEHAVIORAL;
```

As you can see from the VHDL code the count8 design has 3 single signal input ports (CLOCK, ENABLE, CLEAR) and 1 output port (COUT) which is an eight bit vector. This design acts as any counter design you might encounter. The CLEAR input signal resets the counter while the ENABLE input signals allows the counter to count up. After the CLEAR signal has been raised and then dropped and when the ENABLE signal is high, the output increments by 1 on each rising clock edge. The ultimate output signal COUT is inverted due to the fact that the LEDs on the demo board used to view the output are low enabled.

The other point of interest in this design is the insertion of a global clock buffer GCLK_F. The FPGA compiler assigns a clock buffer to any input signal that drives a clock signal, so the buffer has to be instantiated in the VHDL code.

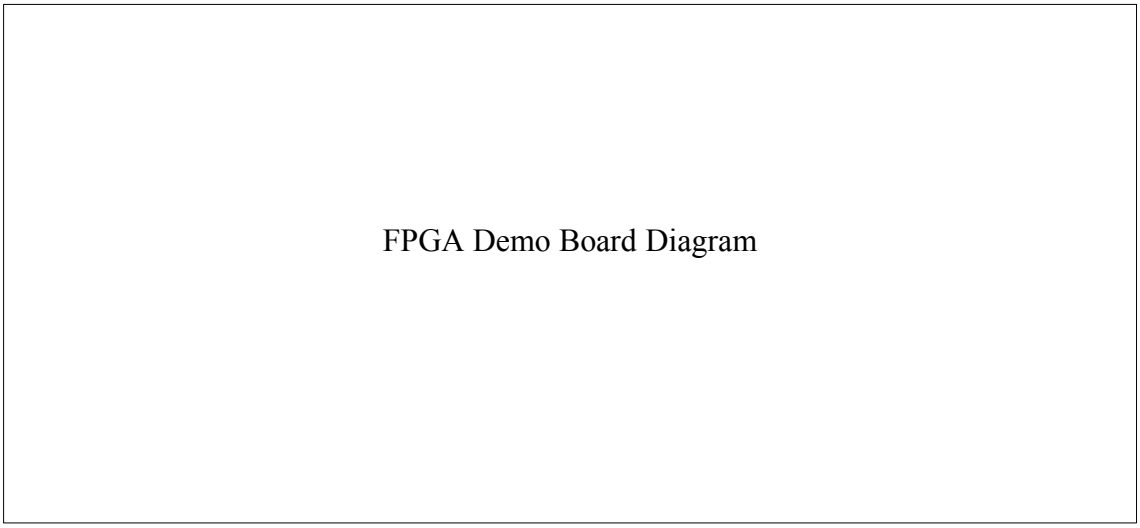
After familiarizing yourself with the count8 design, create a **xnf** file for the design by following the same steps as in the LED Switch example. To do this substitute **count8** for every reference of **ledswitch** in the directions. As in the LED Switch example make sure the following files are contained in your design directory before continuing.

**count8.xnf**
**count8.cst**

## Section 3 - Creation of a Xilinx FPGA Downloadable (BIT) file on the PC Platform

In order to create the file that can be downloaded to the FPGA device the XACT xmake utility will be used. The XACT development tools are contained on the PC Platform, so in order to complete this section of the tutorial you will need to transfer your files to a PC which has these tools installed.

The xmake utility uses a constraint file (CST) to produce the bit file for a specific FPGA device. In this tutorial you will be downloading the design to the Xilinx X3020A FPGA chip. Now is a good time to look at the features of the demo board and the X3020A chip. A diagram of the demo board that was taken from the Xilinx Online Hardware & Peripherals User Guide is given below.




**Xilinx FPGA Demo Board**

The areas on the demo board that will be used in testing and verifying these designs are the set of 8 dip switches (SW3) between the two FPGA chips and the row of 8 LEDs (D1-D8) below the XC4003A FPGA chip. The design constraint file (CST) is used during the xmake utility to constrain the pins on the Xilinx chip to the signals in the VHDL design. The table on the next page shows the pin connections for the XC3020A chip on the demo board.

| SWITCH | XC3020A PIN | LED | XC3020A PIN |
|---|---|---|---|

| | | | |
|---|---|---|---|
| SW3-1 | 11 | D1 | 37 |
| SW3-2 | 13 | D2 | 36 |
| SW3-3 | 15 | D3 | 41 |
| SW3-4 | 17 | D4 | 33 |
| SW3-5 | 19 | D5 | 32 |
| SW3-6 | 21 | D6 | 31 |
| SW3-7 | 23 | D7 | 28 |
| SW3-8 | 24 | D8 | 29 |

**LED-Switch Design Example**

The **ledswitch.cst** file constrains the input vector CIN to be controlled by the SW3 switch set and the output vector COUT to be displayed on the D1-D8 LED set.

1. Boot up the PC and start the Windows operating system.

2. Open a MSDOS prompt from the Main program group.

3. From the MSDOS prompt change to the root directory and create a design directory.

   c*:*
   *md ledswitch*

   Make sure that the XACT directory is include in the path in the **config.sys** and **autoexec.bat** files in the root directory of the PC so the xmake executable file is accessible from your design directory.

   If this directory is not included in the path, please edit these files accordingly and reboot the PC for the changes to take effect.

4. Copy the design file from your floppy disk to the design directory.

   *a:*
   *copy ledswitch.xnf c:\ledswitch*
   *copy ledswitch.cst c:\ledswitch*

5. Change to the design directory and run the xmake utility on the design.

   *cd ledswitch*
   *xmake -x ledswitch*

When the xmake utility completes there will be a several new files in your directory. The ledswitch.bit file is the file that will be downloaded to the FPGA Demo board in the next section.

**Eight Bit Counter Design Example**

Complete these same steps for the eight bit counter design. In this design the CLOCK signal is constrained to switch SW3-1, the ENABLE signal is constrained to switch SW3-2, and the RESET signal is constrained to switch SW3-3. The output signal for the counter is constrained to the LED set (D1-D8).

## Section 4 - Downloading the Design to the FPGA Demo Board and Verifying the design works

In this section you will be downloading the bit file to the demo board. After creating the bit file the following steps need to be completed to download the design.

1. Make sure the demo board is turned on and correctly interfaced to the PC workstation. Section 5 of this tutorial describes how to do this.

2. From the Windows environment Program Manager double click on the XACT step 6.0 program group to open it.

3. Double click on the Xilinx PROflow icon to open the Xilinx PROflow tool.

4. From the Xilinx PROflow window choose the download icon to invoke the Hardware Debugger.

5. In the Hardware Debugger select Open Bitstream… under the file menu, change to the ledswitch directory, and choose the ledswitch.bit file.

   **File -> Open Bitstream…**
   Change to the ledswitch directory
   Choose the **ledswitch.bit** file

6. Select Yes to proceed.

7. Select OK in the Window stating Debugging options will be disabled.

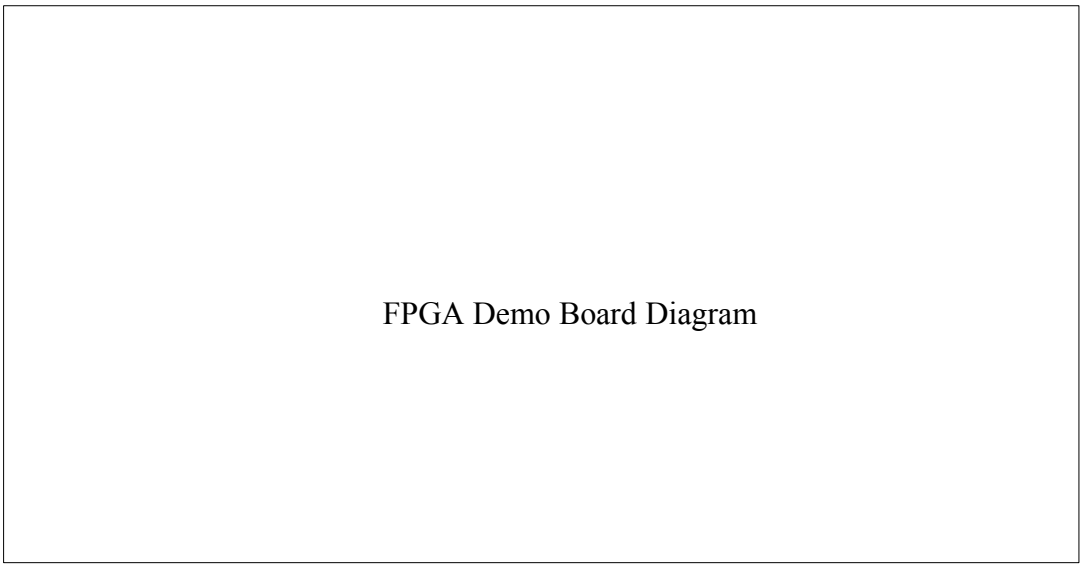
8. Click on the download icon (lightning bolt) to initiate the download process.

The design is now contained in the X3020A chip. Test the LED switch design by toggling the SW3 switches and making sure they turn on the corresponding LEDs.

Repeat this process for the counter design. Set the ENABLE signal (SW3-2) to high and toggle the CLOCK signal0 (SW3-1) to increment the count which is displayed on the LEDs (D1-D8). Due to problems with switch bounce, the counter may act sporadically, however you should still be able to test the validity. To reset the counter raise and then lower the RESET signal (SW3-3).

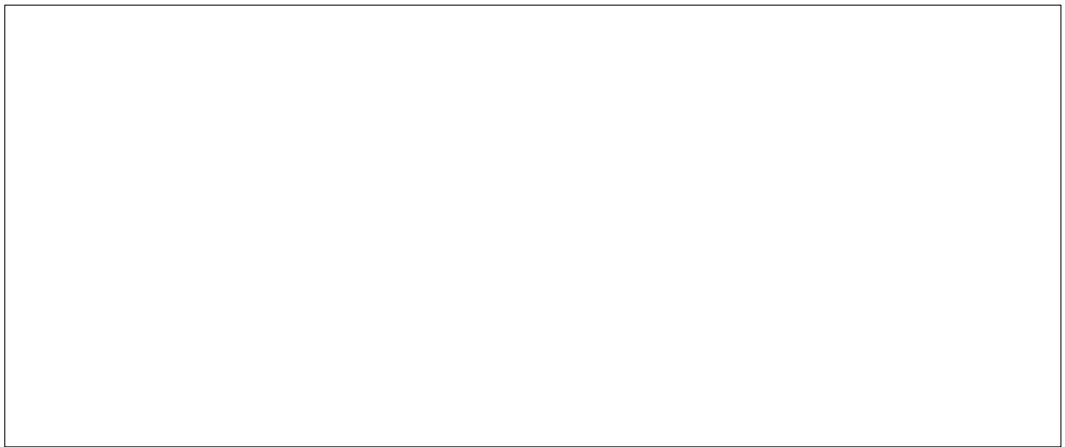
## Section 5 - Connecting the FPGA Demo Board

The diagram below which was taken from the Xilinx Online Hardware & Peripherials User Guide shows the layout of the Xilinx FPGA Demo Board.




**FPGA Demo Board Diagram**

In order to download the design from the PC to the Demo Board, the Xilinx XChecker cable has to be correctly connected. A picture of the XChecker cable and accessories taken from the Xilinx Online Hardware and Peripherials User Guide is shown below.



XChecker Diagram

## XChecker Hardware and Accessories

To connect the XChecker cable plug the DB-9 connector into the serial port on the back of the PC, and connect the Flying Lead connector 1 to the cable in the appropriate slot. (Note: The XChecker cable draws power from the Demo Board not the host computer.)

**WARNING - PRIOR TO CONNECTING THE XCHECKER CABLE TO THE DEMO BOARD, MAKE SURE THAT POWER IS SUPPLIED TO THE DEMO BOARD AND IS TURNED ON.**

The Demo Board is powered by a regulated 5 volt power supply. The power supply is to be connected to pins **J9-1** and **J9-2** which are at the top of the center of the board. The +5V lead of the power supply connects to pin **J9-1** while the ground connection is to pin **J9-2**. These connections can be made using the wire wrap tool. When the power supply is correctly connected and turned on the period segment of the 2 leftmost 7 segment LEDs will be on.

After the power supply is connected and turned on, the XChecker cable can be connected to the demo board using the Flying Lead Connector 1. For the designs to be downloaded properly, the 5 leads in the table on the next page have to be connected to the corresponding pins on the demo board. (Note: The **J1** pins are in the top left corner of the board and are numbered left to right and top to bottom.)

| Flying Lead Connector | Demo Board Pin Number |
|---|---|
| | |

| | |
|---|---|
| VCC (Red) | J1-1 |
| GND (Black) | J1-3 |
| No Connection | J1-5 |
| CCLK (Yellow) | J1-7 |
| D/P (Blue) | J1-9 |
| DIN (Green) | J1-11 |

The final step in interfacing the boards is correctly setting the switch banks **SW1**, **SW2**, and **SW3** to download the designs to the X3020A chip. The correct switch setup is given in the table below. (Note: X stands for don't care.)

| **Switch 1** | **Setting** | **Switch 2** | **Setting** | **Switch 3** | **Setting** |
|---|---|---|---|---|---|
| SW1-1(Left) | X | SW2-1 | OFF | SW3-1 | OFF |
| SW1-2 | OFF | SW2-2 | X | SW3-2 | OFF |
| SW1-3 | OFF | SW 2-3 | X | SW3-3 | OFF |
| SW1-4 | ON | SW2-4 | X | SW3-4 | OFF |
| SW1-5 | ON | SW2-5 | X | SW3-5 | OFF |
| SW1-6 | ON | SW2-6 | X | SW3-6 | OFF |
| SW1-7 | OFF | SW2-7 | X | SW3-7 | OFF |
| SW1-8 | OFF | SW2-8 | X | SW3-8 | OFF |

The demo board is now correctly interfaced and ready to receive a design for the X3020A chip.

# APPENDIX A - Demo Board Layout

On the following two pages there is a schematic of the demo board and how the XC3020A and XC4003A components are interfaced to the it. Also, shown below for easy reference is a table listing hardwired connections of the pins of the XC3020A to switches, LEDs, etc. of the demo board. (Note: The XC3020A is connected to the following devices on the demo board: Switch 1, Switch 3, LEDs D1-D8, 7 segment LED U6.)

| XC3020A Pin Number | Demo Board Device |
|---|---|
| 11 | Switch 3-1 |
| 13 | Switch 3-2 |
| 15 | Switch 3-3 |
| 17 | Switch 3-4 |
| 19 | Switch 3-5 |
| 21 | Switch 3-6 |
| 23 | Switch 3-7 |
| 24 | Switch 3-8 |
| 28 | LED D7 |
| 29 | LED D8 |
| 30 | 7 Seg LED decimal point |
| 31 | LED D6 |
| 32 | LED D5 |
| 33 | LED D4 |
| 36 | LED D2 |
| 37 | LED D1 |
| 38 | 7 seg LED segment a |
| 39 | 7 seg LED segment b |
| 40 | 7 seg LED segment c |
| 41 | LED D3 |
| 49 | 7 seg LED segment e |
| 53 | 7 seg LED segment f |
| 55 | 7 seg LED segment g |
| 56 | 7seg LED segment d |

**7 Segment Display (U6)**

# APPENDIX B - The Design Constraint (.CST) file

The design constraint file (designname.cst) is used by the Xilinx **xmake** tool to constrain signals in the design to specific pins on the FPGA device. The following is the constraint file for the eight bit counter design in this tutorial.

```
place block ENABLE : P13;
place block CLEAR : P15;
place block COUT<0> : P29;
place block COUT<1> : P28;
place block COUT<2> : P31;
place block COUT<3> : P32;
place block COUT<4> : P33;
place block COUT<5> : P41;
place block COUT<6> : P36;
place block COUT<7> : P37;
```

If this file is correctly setup and in the design directory, the **xmake** tool will automatically use it. Notice that **CLOCK** signal is not constrained here. Since the Global clock buffer using a dedicated pad location was included in the VHDL design, this signal must be controlled by **Switch 3-1** and should not be included in the constraint file.

To produce a constraint file for a different design the same format should be used. Using a text editor the file should be created as follows:

place block **SIGNALNAME** : **PINLOCATION**;

# APPENDIX C - The Compile Script (.SCRIPT) file

The following is the compile script file used in the eight bit counter design in this tutorial. All designs should use a compile script very similar to this.

```
/*   Script for Synopys to Xilinx design compiler

/* User defined constants used later in the design script
TOP = count8
PART = "3020apc68-7"
designer = "Your Name"
Company = "Clemson University"

/* Analyzes and elaborates the VHDL design
analyze -format vhdl TOP + ".vhd"
elaborate TOP

/* Inserts and correctly sets up the input and output signals
set_port_is_pad "*"
set_pad_type -slewrate HIGH all_outputs()
insert_pads
remove_constraint -all

/* Specifies the CLOCK signal to be a clock with a 50ns period
create_clock "CLOCK" -period 50

/* compiles the design
compile

/*  Generates reports for FPGA resources used and design timing characteristics
report_area > TOP + ".area"
report_timing > TOP + ".timing"

/* Constrains the design to the XC3020A part
set_attribute TOP "part" -type string PART

/* Saves the compiled design the Xilinx database and Synopsys/Xilinx netlist
/*  formats
write -format db -hierarchy -output TOP + ".db"
write -format xnf -hierarchy -output TOP + ".sxnf"

exit
```

# APPENDIX D - Available Resources

The following is a list of available resources that might be useful in the completion of this tutorial or the implementation of another design.

**Xilinx Online Docs** - Online documentation is provided in the XACT Step 6.0 software. These documents are in Adobe Acrobat Reader Format (.pdf) and can be accessed via a PC that has the XACT software installed on it. The XACT Step 6.0 program group contains an icon that is labeled Xilinx Online Docs. By double clicking on this icon the Adobe Acroreader is invoked and a menu for the online documents is displayed. The following topics are of particular interest:

**Hardware & Peripherals User Guide**
**Hardware Debugger Reference/User Guide**

**Xilinx World Wide Web Page** - Xilinx maintains a very useful and up to date web page. The web page contains links to Product Specifications as well as a very useful search tool that will search many non Xilinx FPGA web sites. The URL for their web site is listed below.

**http://www.xilinx.com**

**Xilinx XACT Step Documentation** - Xilinx has produced a series of books containing information and tutorials for the FPGA Demo Board and Xilinx and Synopsys software. The following is a list of these books that may prove useful:

**XACT Step Synopsys (XSI) For FPGAs Interface/Tutorial Guide**
**XACT Step Viewlogic Tutorials**