Express Design Flow Lab

Express Design Flow Lab

Introduction

This lab uses the WATCH project to demonstrate how a project is optimized for implementation in the M1 software using the Foundation Express software. Foundation Express uses FPGA architecture information to produce an optimized FPGA netlist. This optimized netlist can then be implemented in the M1 software with timing specifications entered in the Express Timing Specifications GUI. Results of the optimized project will be analyzed in the M1 Timing Analyzer.

Objective

After completing this lab, students will be able to:

- Implement and optimize a design with the Foundation Express software
- Complete an implementation within the M1 software
- Utilize the Express Project Report, M1 Timing Analyzer, and the M1 Report Browser to determine the design's utilization and performance

About the WATCH project

The WATCH project is designed to be a track coach's stopwatch (refer to Appendix for equivalent schematic and HDL design files). There are two inputs to the system (RSET and STRSTP). The configuration clock on the FPGA is used as a ten-hertz clock signal. Three seven-bit outputs are generated by this system for output to three seven-segment displays.

This design was chosen to demonstrate many of the possible design entry techniques Foundation Express allows in a single project. Since this design uses hierarchical design entry techniques, the WATCH project also provides an opportunity for students to utilize the Timing Specifications GUI provided by Express to control optimization.

Procedure (VHDL)

Creating a new project in Foundation Express

1) Start Foundation Express by clicking on **Start → Programs → Xilinx Foundation Series → Accessories Foundation → Xilinx Foundation Express**. 

2) Once inside Foundation Express, click on **File → New…**

3) Go to the **`c:\F15_labs\xpreslab`** directory. Enter the name of the project as **`WATCH`** and click on the **Create** button. When asked to identify sources, click on the **Cancel** button.

Identifying Source files

4) Once the **Foundation Express Project Window** opens, click on the **WATCH** project name in the **Design Sources** window. Then click on the command: **Synthesis → Identify Sources…** Go to the **`c:\F15_labs\xpreslab`** directory, select the **`CNT60.VHD`** file, and click on the **Open** button. Express will check the file for errors, and should not find any.

5) Click on the **`WATCH`** project name in the **Design Sources** window, and then click on the command: **Synthesis → Identify Sources…** Go to the **`c:\F15_labs\xpreslab`** directory, select the **`STOPWTCH.VHD`** file, and click on the **Open** button. Express will check this file for errors, and should not find any.

6) Click on the **`WATCH`** project name in the **Design Sources** window, and then click on the command: **Synthesis→Identify Sources…** Go to the **`c:\F15_labs\xpreslab`** directory, select the **`WATCH.VHD`** file, and click on the **Open** button. Express will check this file for errors, and report that an error was found in the **`WATCH.VHD`** file. This error will be fixed later in this lab.

7) To **instantiate** LogiBLOX modules, EDIF (.**EDN**) netlists, or Xilinx (**.XNF**) netlists with any project, make certain these files are in the project directory. Use the Windows File Manager or the Windows NT Explorer to verify that the **`TENTHS.NGO`** and **`HEX2LED.EDN`** files already exist in the **`c:\F15_labs\xpreslab`** directory.

8) The remaining components (**OSC4** and **INV**) are from the Xilinx Unified Libraries and will be added to the project within the M1 software. Note that the Express software cannot optimize the black box instantiations **TENTHS**, **HEX2LED**, **OSC4**, and **INV**. The M1 software will optimize these modules, if possible.

Creating a WATCH Implementation

9) After the identifying the necessary source files, each design file will be checked for syntax errors automatically. There is one syntax error in the **WATCH** file and this will be flagged with a red **X** next to the file name in the **Design Sources** window.

10) To correct the error, right-click on the **WATCH** design file and select the **Edit File** option. This will activate the error checking mechanism within Foundation Express. Make the necessary syntax change to the component, save the design file, and click on the **Update** button on the toolbar to recheck the syntax of the **WATCH** design file. (**Hint** – Express tells you what line to look on, and the error has to do with a missing  ";" semicolon.) After no errors are found, save the file and exit the editor. Then right-click this file to **Update** it. Note that all three HDL files now have green checks next to their names in the **Design Sources** window.

11) Within the Foundation Express software, an implementation is considered to be an assembly of all the necessary HDL files into a single project. To create the **WATCH** implementation from the HDL files, click on any of the design files in the **Design Sources** window, and select **WATCH** from the drop down box at the top of the screen (see **Figure 1**).

12) After the top level HDL file is selected, the **Create Implementation-WATCH** window will open. Select the **4005XLPC84** device with a **–3** speed grade, a **1 Mhz** default clock frequency, and leave the remaining options on their default settings (see Figure 2). Click on **OK**.

13) Review the messages placed on the implementation in the message window at the bottom of the **Project Window**. To view these messages, select the **WATCH** implementation created in the **Chips** window, and then click on any of the **Errors, Warnings,** or **Messages** tabs. See the Help menu for code explanations. Simply note the code at the end of the message (such as **FE-LINK-2**), and open the help menu with the **Help → Help Topics** command. Click on the **Index** tab and type in the code in window #1.

    What's going on in most of these cases is that the Express software is not able to find a link from the VHDL modules to the non-VHDL modules, such as TENTHS (*.NGO format) and HEX2LED (*.EDN, EDIF netlist format). This is because these modules were created using some other design entry tool, and thus there is no VHDL code to merge at this stage in the design synthesis.

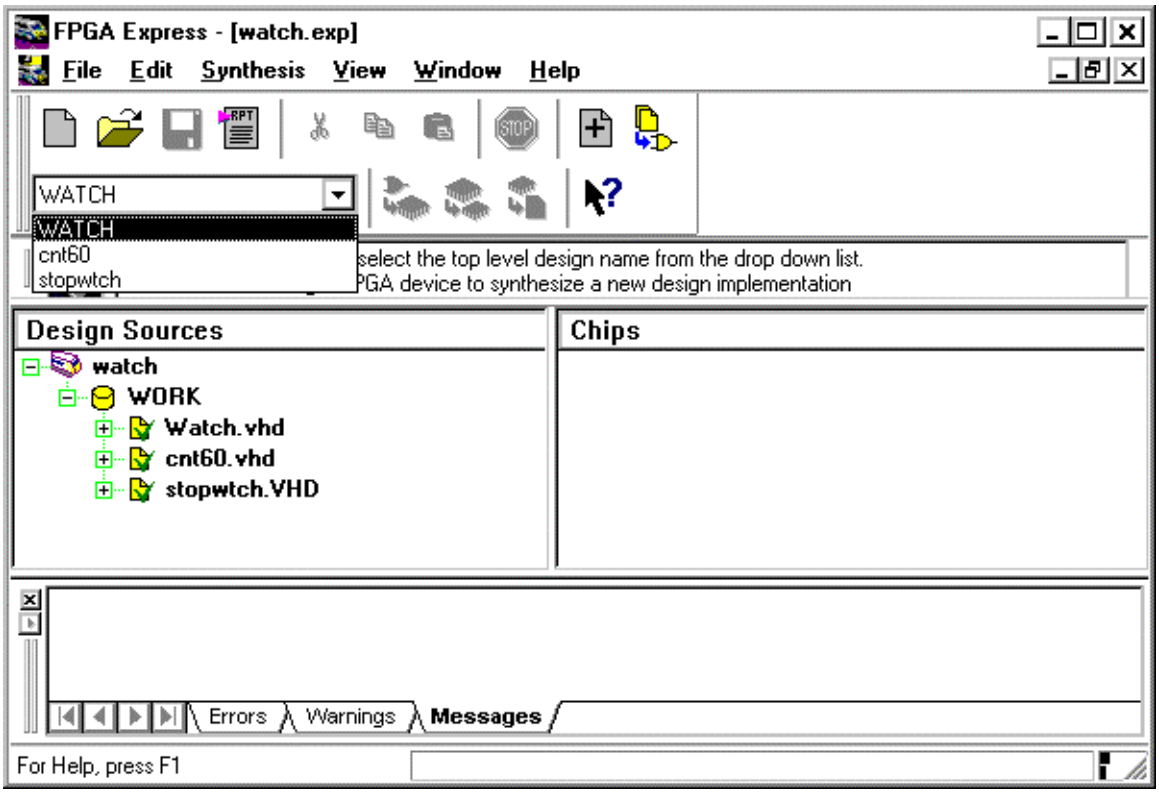

Figure 1. Choosing the WATCH file as the top file in the design hierarchy.

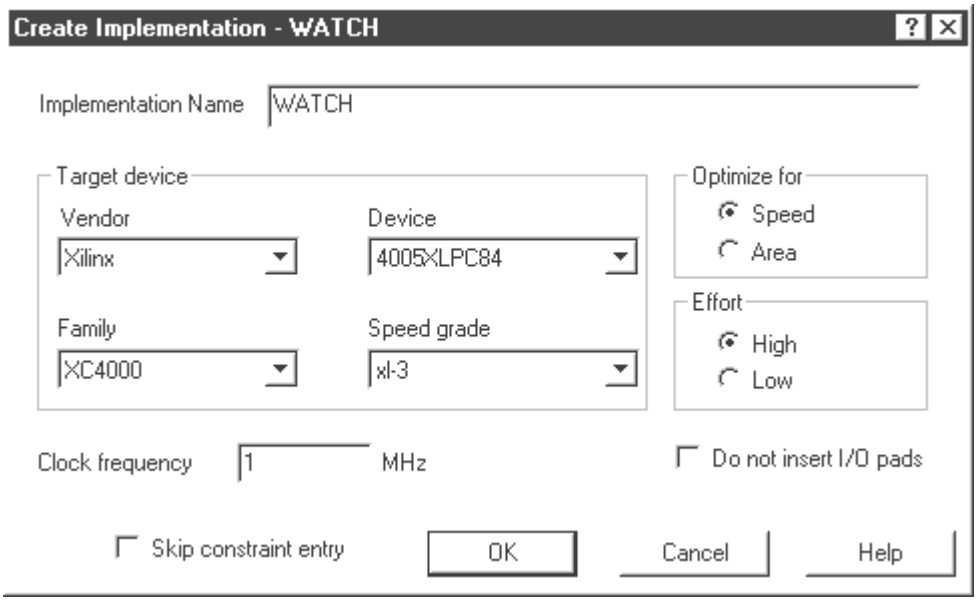

Figure 2. Assigning a device and a default clock frequency.

Optimizing the WATCH project and Viewing the Project Report

14) After the implementation has been completed successfully, optimize the associated HDL files for the chosen FPGA by selecting the **WATCH** implementation in the **Chips** window, and clicking on the **Optimize** button on the toolbar.

15) View any warnings associated with the optimized **WATCH** project by clicking on **WATCH-optimized** in the **Chips** window, and then clicking on the **Warnings** tab.

16) Generate a **Project Report** by selecting **WATCH-optimized** in the **Chips** window, and clicking on the **Project Report** icon on the toolbar. Save the report as **WCHRPT.TXT** in the **c:\F15_labs\xpreslab** directory. Use **Notepad** or the **Windows Explorer** to open the report. Use the information provided in the Project Report to answer questions 1 through 4 in the **Questions** section of this lab.

17) Finally, export the necessary **XNF** file to be read by the M1 software by selecting **WATCH-optimized** in the **Chips** window, and clicking on the **Export Netlist** icon on the toolbar. Save the **XNF** file in the **c:\F15_labs\xpreslab** directory and assert the **Export Timing Specifications** option.

Implementing the WATCH project in the M1 software

18) Start the M1 software by using the command **Start → Programs → Xilinx Foundation Series → Accessories →Design Manager**.

19) Click on the command **File → New Project** and browse for the **WATCH.XNF** file in the **c:\F15_labs\xpreslab** directory. It may be necessary to change the filter to **XNF** to locate the file.

20) After the project has been created, M1 will show the **WATCH** design as the current project in the **M1 Design Manager** window.

21) In the **M1 Design Manager** window, click on the left most button on the horizontal scroll bar, or click on **Design → Implement**. The Design Manager will prompt for the correct device by bringing up the **Implement** window. Make sure that the **XC4005XL-3-PC84** is selected and click on the **Options** button and then **Implemtation: Edit Templates → Timing Reports**.

22) You may deselect the **Produce Configuration Data** and **Produce Timing Simulation Data** options, since the finished design will not be downloaded or simulated in this lab. Assert the **Produce Logic Level Timing Report** and the **Produce Post Layout Timing Report** to have this information created by the Flow Engine (see Figure 3).

23) Click on the **OK** button in the **Options** window, and click on the **Run** button within the **Implement** window. This will start the implementation process and bring up the **Flow Engine** window.

24) While the design is being processed, monitor the implementation progress through the Flow Engine's Message window at the bottom of the **Flow Engine**. After the Flow Engine has stopped running, click on the **OK** button.

25) To review any of the reports, click on **Utilities → Report Browser**. The information in these reports will be necessary to answer questions 5 through 7 in the **Questions** portion of the lab. After answering the questions, exit the M1 Design Manager.

Questions

1) What are the default Optimization settings in the Foundation Express software as listed in the Chip Parameter section of the Project Report?

2) How many **D Flip-Flops** are required by the optimized WATCH netlist as listed in the Primitive Reference Count section of the Project Report? Is this the same number that should be expected after implementation in the M1 software?

3) How many four-input function generators (FMAPs) are required by the optimized WATCH netlist as reported in the **Primitive Reference Count** section of the **Project Report**?

4) What is the estimated clock frequency for the OSC_CK clock signal from the Project Report generated by Foundation Express?

5) From the **Place and Route Report** generated by the **M1 Flow Engine**, how many registers are required by the WATCH project? Why is this different from the number reported by the Foundation Express Project Report?

6) According to the **Logic Level Timing Report**, did the Flow Engine read Timing Specifications?

7) From the **Post Layout Timing Report**, what is the minimum period? Why is this different from the number reported by the Foundation Express Project Report?

Conclusion

In this lab, the Foundation Express software was used to optimize the HDL files associated with the WATCH project. The Express Project Report was used to analyze the design before placement and routing of the design. The reports generated by the M1 Flow Engine were used to determine the accuracy of the Express Project Report.

Express Design Flow Lab Answers

1) The 1 Mhz Default Clock Frequency placed in the Create Implementation template generated a variety of timing specifications that can be seen throughout the Project Report. It is important to note the effects of this single constraint on the clocks, timing path groups, input ports, and the output ports sections of the Project Report. The remaining default optimization settings can be seen in the Chip Parameters section and they include optimizing for speed and high optimization effort.

2) The WATCH project optimizes to approximately 14 DFFs. This is not accurate because Foundation Express cannot include the HEX2LED or TENTHS netlists in the project. Remember that Foundation only optimizes VHDL or Verilog netlists associated with the project. The only macros that are in HDL are STOPWTCH, CNT60, and WATCH.

3) From the Primitive Reference Count section of the Project Report, between 25 and 31 FMAPs, and between 2 and 5 HMAPs are required by the STOPWTCH and CNT60 components. Since the TENTHS and HEX2LED components are not included in the Express optimization, many more function generators will be necessary to implement the entire WATCH project.

4) The OSC_CK clock signal is estimated to run at approximately 35 Mhz. This is found in the Clocks section of the Chip Watch Optimized portion of the Project Report generated by Foundation Express.

5) The M1 Place and Route Report completely optimizes the WATCH project and includes the registers not included by the Express Project Report. From the Place and Route Report, about 24 registers are used (23 CLB and 1 I/O).

6) According to the Logic Level Timing Report, the M1 Flow Engine inferred 8 timing specifications. Since each one is 1000 ns. long, it is logical that these groups were inferred by the 1 Mhz default Clock Frequency placed in the Foundation Express software. The M1 implementation is different from the Express implementations since the M1 software combines all the design files needed for the project.

7) From the Post Layout Timing Report, the longest delay path limited by any timing constraint is between 29 and 35 ns. These and any other timing results generated by the M1 software can differ from the values generated by the Express Optimizer, because the Express delays include estimated net and block delays. The M1 software can frequently improve on the designer's expectations, because it has the ability to move logic closer together and choose shorter net delays. This is very apparent in this design because the design is very small and uses about 1/3 of the chosen device's resources.