EEL 3705L, Digital Logic Design Lab, Spring '11

Laboratory Assignment #0: Switches & Seven-segment Display

Version history:
v1.0, 1/7/07: Copied & updated last semester's version (not yet used). –M. Frank
v2.0, 8/28/08: Updated version with DE2 board – H. Hoang
Spr. '11, v0.1, 1/8/11: Carried over v2.0 as used in Spr. '10. -MF
v1.0, 1/9/11: Updated Quartus instructions, expanded assignment to make the output dependent on the state of an input switch. (Still easy, but teaches more.) -MF
v1.1, 1/10/11: Corrected year in document title. -MF

0. Document Description

This document describes a trivial lab assignment (a simple design tutorial) to be conducted the first week of classes, before we have even had time to teach about logic gates, except for the NOT gate, which was briefly introduced in the first lecture. However, it engages students in working with the real board hardware and seeing real outputs right away. At a circuit level, it is very simple, and this may help students to more easily understand what they are doing when they are just starting to learn Quartus. It does not include a pre-lab report requirement, because students in the Monday lab section will not have had a chance to prepare a pre-lab report between the first lecture and lab. However, it does include a simple design component and a lab report requirement.

1. Lab Synopsis

In the pre-lab, students design & compile a simple circuit for the Cyclone II EP2C35 chip that will display two team members' birth dates on the DE2 board's 8-digit 7-segment display, under control of a two-position slider switch. In lab they will test their circuit on the actual prototyping boards, modify their circuit if needed, and report their results.

2. Educational Objectives

The educational objectives of this lab assignment are:

1. Begin to familiarize students with the hardware available on the DE2 prototyping boards, including simple input and output devices.
2. Begin to familiarize students with the use, capabilities, and design flow of the Quartus II logic EDA (Electronic Design Automation) tool.
3. Introduce students to the Altera Cyclone II EP2C35 FPGA (Field-Programmable Gate Array) chip.
4. Teach students how to set up pin assignments in Quartus.
5. Teach students how LEDs are controlled using active-low digital signals.
6. Teach students how 7-segment displays are controlled.
7. Show students how NOT gates can be used usefully to carry out simple functions.
8. Introduce students to Quartus signal bus naming conventions.
9. Familiarize students with the process of keeping a lab notebook and writing up their results in a properly structured lab report.

3. Design Objective

Your goal in this lab assignment is to program a very simple circuit (requiring only wires and a single NOT gate!) into the Cyclone II EP2C35 FPGA which causes it to display either of two team members' birth dates (in MM DD YYYY format) on the board's eight-digit seven-segment display (which is laid out as two 2-digit displays and one 4-digit display). The choice of which team member's birth date to display at any given moment should be controlled by the slider switch SW0. The circuit design is extremely simple; the required function can be implemented using just a single NOT gate and appropriate wiring. (This ensures that students understand how to go through a complete design cycle before they start creating more complex designs.)

4. Pre-Lab Preparation

Before coming to the lab, you should have gone through the following steps:

4.1. Installing & learning to use Quartus

1. If you have your own Windows (or Linux) computer (or a Mac running Windows emulation software such as Parallels, VMware Fusion, or VirtualBox), download and install the latest version of Quartus II Web Edition (which was 10.1 (Build 153) as of Jan 8, 2011). The file is very large (3 GB), so please allow several hours for the download (depending on the speed of your broadband service). After you download and execute the self-extracting archive file (called 10.1_quartus_free_windows.exe), you may find that you need to manually launch the setup file that it generates, if it doesn't start automatically – so make note of what the destination folder is when you first run the 3 GB download file.

2. Alternatively, if you need to save download time, an older version of Quartus II (7.1) is included with the 3rd edition of the textbook; that version should suffice for most purposes in this class. However, these older versions of Web Edition (before 8.1) will require you to set up a license before you will be able to use all of their features. https://www.altera.com/download/licensing/free_software/lic-q2web.jsp should take you to an online form where you can request a license file based on the MAC address of your computer's network interface card. The file will arrive in an email attachment. Save it to the folder where you installed Quartus, e.g. in **`C:\altera\71\`** . Then restart Quartus, select "Specify valid license file", and fill in the path of the file.

3. Otherwise, you can just use the version of Quartus (9.1 or later) that is installed on the public PCs and computer labs in the Engineering building.

4. Once Quartus is started, pull down the menu item *Help* → *Tutorial* or *Help* → *Getting Started Tutorial* (name depends on Quartus version), and go through this interactive tutorial to gain some initial familiarity with the Quartus software. You will need a computer with speakers to hear the instructions in the tutorial. You may also need to modify the Global Security Settings in your web browser's Flash

media player plug-in to prevent errors. Visit http://www.macromedia.com/support/documentation/en/flashplayer/help/settings_manager04.html, and in the embedded "Adobe Flash Player Settings Manager" applet on that page, in the third tab (Global Security Settings), click "Edit locations..." → "Add location..." and enter C:\altera, or whatever is the name of the folder where the copy of Quartus you are running is installed. Also, there are a number of video demonstrations and online courses about Quartus on Altera's website which you should explore as well. (See http://www.altera.com/education/demonstrations/online/design-software/onl-design-software-demos.html.)

5. On Blackboard, under *COURSE LIBRARY* > *ALTERA RESOURCES (QUARTUS, DE2 BOARD)* > *QUARTUS II SOFTWARE* → *Quartus II Introduction Using Schematic Designs*, there is a 37-page tutorial document that takes you step-by-step through the design process using CAD (computer-aided design) style graphical entry of logic circuit schematics in Quartus. Read through this document so that you will have an idea what to expect.

4.2. Quartus Project Setup

1. Start Quartus. Please make sure that you are using version 7.0 or later. (The DE2 board comes with Version 7.0, the CD-ROM in the back of the textbook has version 7.1, version 9 should be installed in the lab, and you can download version 10 from Altera.) If running one of the older versions, it will ask you on startup whether you want the (new) Quartus II or (old) MAX+PLUS II look and feel – select Quartus II. The main window should look something like this:

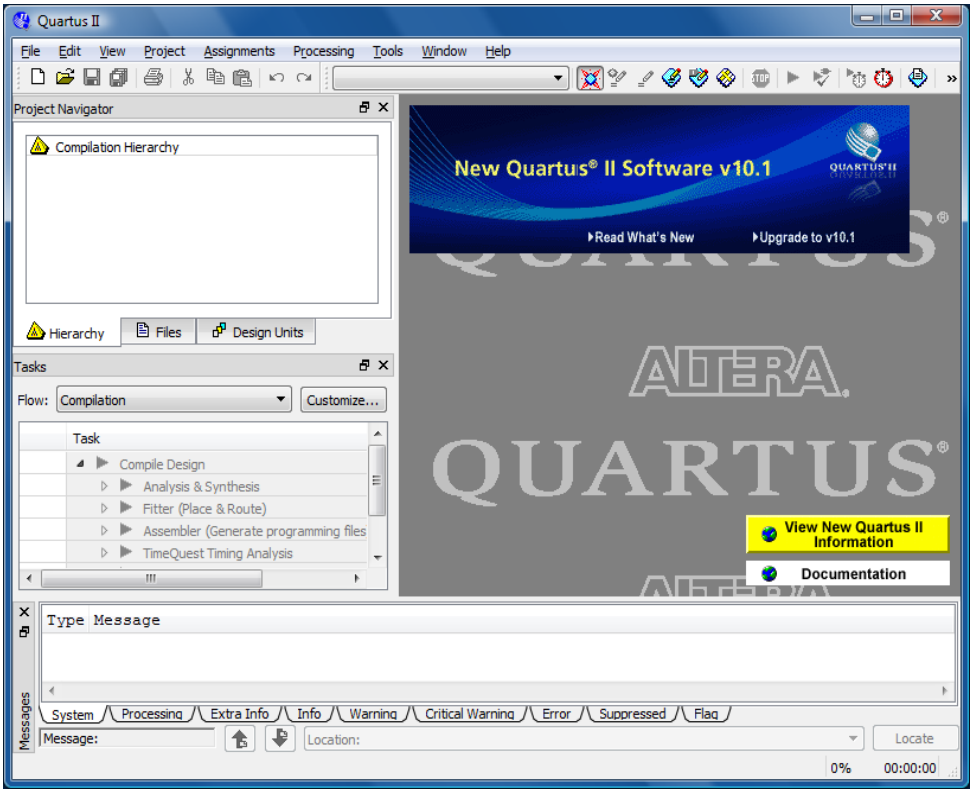

Figure 1. Appearance of the Quartus II (v10.1) main window. If you're running an older version, there may be a few minor differences.

2. The first thing we have to do is create a new project. Select *File → New Project Wizard…* Click "Next >". In the next page of the wizard, you need to select a working directory for your project. Hit the "…" button beside the text box, and browse to where you would like to create your directory. It must be a location writable by you. If you're using a college machine, this must be on your own flash drive, or on the network drive that accesses your CMS home directory. If you're using your own computer, it can be wherever you like; I recommend somewhere under your user Documents folder. In the file browse dialog, there is a button to create a new folder; click it. Name the new folder "Lab0". In the next text box (project file name), type "Lab0" also. This will be automatically used as the name of the top-level entity (design module) in the project as well. At this point, your wizard should look something like the following:

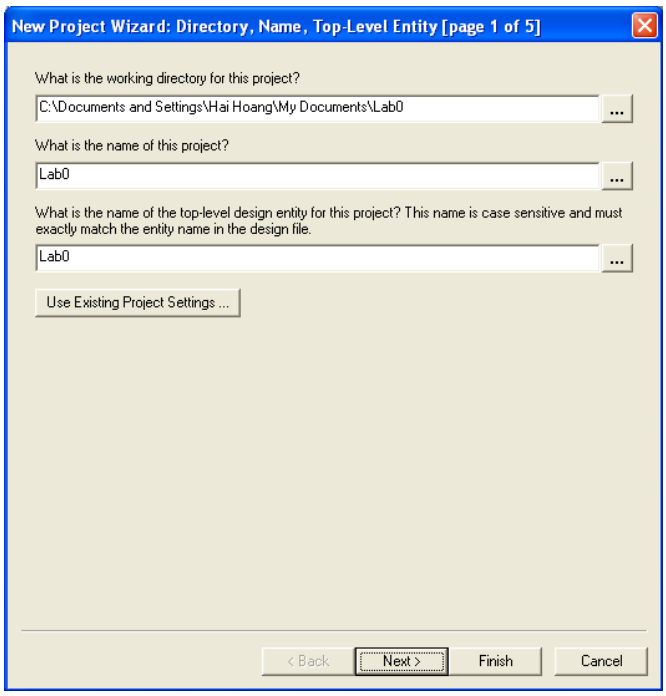

Figure 2. New Project Wizard after entering the names of the folder, project, and top-level design entity.

3. Click "Next >" in the new project wizard. The next page asks you to add design files to the project, but you haven't created any design files yet, so just click "Next >" again. The next page asks you to select the device to be programmed in this project. In the "Family" pulldown, select "**Cyclone II**". Scroll through the list of Available devices until you find the part number of our specific chip, **EP2C35F672C6**. Click "Finish". Now the Quartus window should look like below:

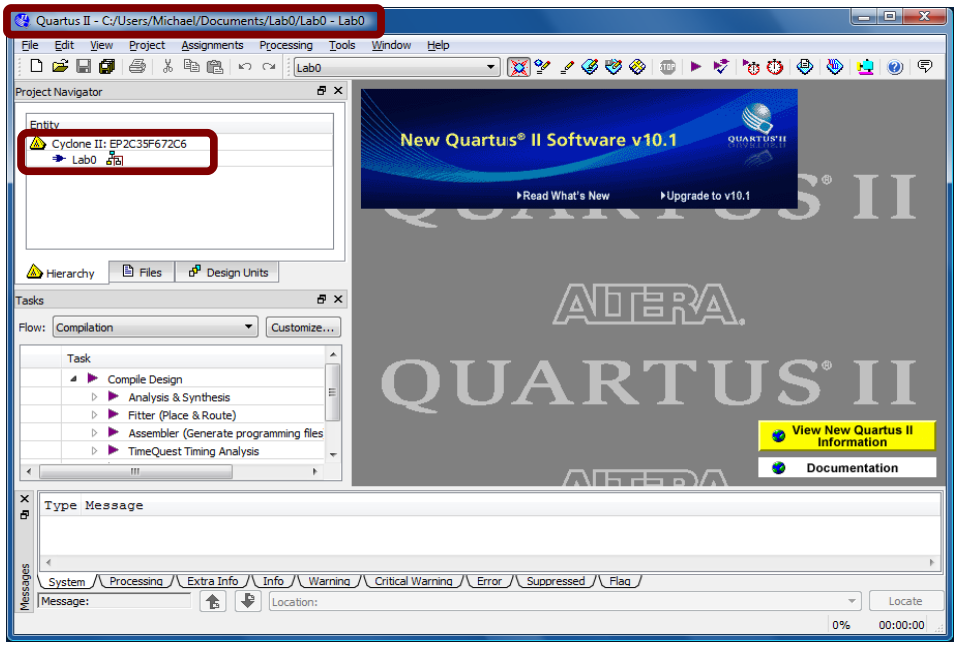

Figure 3. Appearance of Quartus after project creation.
Notice the changes in the title bar and the Project Navigator pane.

4. Now that you've created your project, the next step is to create a schematic entry file within the project. You will draw your circuit into this file. Select File → New… or just click the New icon (looks like a blank sheet of paper). Double-click on "Block Diagram/Schematic File". You should get a new window that looks something like this.

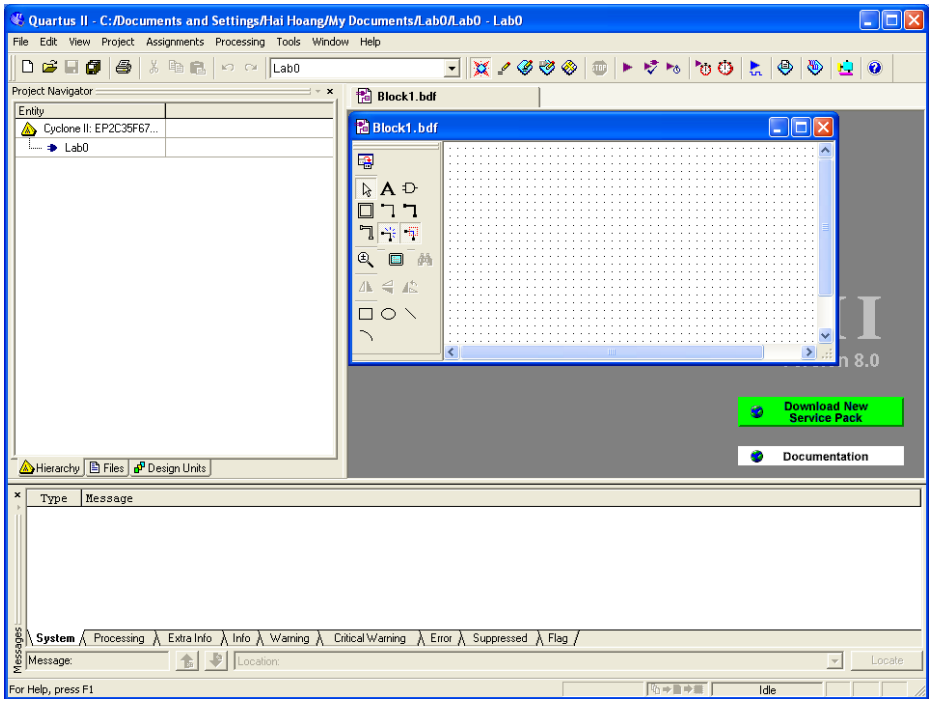

Figure 4. Child window for new schematic (.bdf) file.

5. Next you need to specify the name of this file. We'll call it Lab0 also, since it is the main design file for our project of that name. Select *File → Save As…,* and make sure the File name is Lab0, that you are navigated to the correct project folder, and that the "Add file to current project" checkbox is checked, and hit save. Now, the new empty file is saved with the correct name, and is added to the project. The child window should now be titled "Lab0.bdf". Maximize the child window, and now we're ready to start drawing circuits.

4.3. Design Planning

The next major phase in the pre-lab is to design your circuit. You'll start by writing up notes in your lab notebook, then enter your design into Quartus. Here is a suggested sequence of steps.

1. Write down in your lab notebook the birth dates of the two team members, team member #1 (whose birthday will be displayed when SW0 is in the "up" position) and team member #2 (whose birthday will be displayed when SW0 is in the "down" position). Write them in MM DD YYYY format, suitable for display:

   Team member #1 birth date: __ __ __ __ __ __ __ __
   Team member #2 birth date: __ __ __ __ __ __ __ __

2. Next, figure out exactly how each of these dates will be represented on the 8-digit, seven-segment LED numeric display. What you need to know about this how this display works is that each little segment of each digit is controlled independently. On Blackboard under COURSE LIBRARY > ALTERA RESOURCES (QUARTUS, DE2 BOARD) you can find the User Manual for the DE2 board. There is a section in this manual about "Using the 7-segment Displays," with a figure showing the numeric index (0-6) that is associated with each of the 7 segments. The rightmost digit is named "HEX0" and the leftmost digit is named "HEX7". Using this information, you should be able to figure out, for each segment of each digit, whether it should be ON (lit up) or OFF (unlit) for each of the two birth dates you want to display. Make a table in your lab notebook that contains all this information. You can format it as you wish, but it should be complete and detailed. For example, the table could include 56 rows (for all 7 segments of all 6 digits) and at least two columns (for the two birth dates). All rows and columns should be labeled clearly.

3. Next, think about how you will cause each of these dates to be displayed when the switch is in the corresponding position. Consider that the switch will provide you with one input bit of information (0 or 1, depending on whether it is up or down), and that each of the seven LED segments making up each digit in the 7-segment digit display panel will be controlled by one bit of information - 0 or 1, depending on whether the corresponding LED is to be ON (lit) or OFF (unlit). By looking at what the state of each segment should be for each of the two birth dates (as in the table you made for the previous step), you can see that the segment must have one of the following four possible behaviors:

a. It should always be turned on, no matter where the switch is.
b. It should always be turned off, no matter where the switch is.
c. It should be on whenever SW0 is up, and off whenever SW0 is down.
d. It should be on whenever SW0 is down, and off whenever SW0 is up.

Make up suitable short code names for each of these four cases which will remind you what they are, and add this information as a new column in your table.

4. To implement these four behaviors, you will need to create wires running to each of the 56 segments from each of 4 signal sources. One of these sources will produce a signal voltage that is always relatively low (ground). One will be a voltage that is always high (the logic voltage, usually called Vcc or Vdd for historical reasons). One will be the logic signal coming directly from SW0. And one will be the logical complement of this signal – which can be generated from using the NOT gate (logical inverter) mentioned in the first lecture. To figure out exactly which case is which, you will need to examine the DE2 manual more carefully. Carefully read sections 4.2 and 4.3 of the manual, and you will find that you can figure out from the information provided exactly which case is which. Make a note of this information in your lab notebook.

At this point, you are ready to start actually wiring up your design in Quartus.

4.4. Circuit Design

1. We will need our schematic to include icons representing the input and output pins connecting to the switch and to the LEDs in the 7-segment display. Before we can add these pins to our design, we need to know the formal names of their signals in the context of the DE2 board. This information is contained in the DE2.qsf file, which is available in the Course Library area on Blackboard, under Altera resources. Download this file. This specifies the pin assignments, in other words, which pins on the FPGA package are connected to which board-level signals. You'll need to import these pin assignments into your project – in Quartus, under assignments, select "Import Assignments...", select the DE2.qsf file you just downloaded, and hit OK. Now, if you go to Assignments → Assignment Editor, you can see all the pin assignments you just loaded. You'll see SW[0] is right at the top, and the signals for the 7-segment display start around line 91. These are then the names you should use for your input and output pins, to make sure they connect to the correct devices on the board.

2. OK, now let's start by creating an icon representing the input pin from SW0. Click on the tab for your Lab0.bdf file to bring it back to the front. Double-click anywhere on the grid to bring up the Symbol dialog. Type "input" in the Name field and click OK. Now click on the grid again to place the icon. Double-click on its name (which starts out as something generic like "pin_name") and change it to "SW[0]" (the official name of the pin as shown in the Assignment Editor).

3. Next, we'll create icons representing the output pins to each of the eight 7-segment digit displays. We'll create 8 output pin icons, where each icon represents the entire group of 7 pins for one digit. Double-click on the grid and enter the name "output," but this time before hitting OK, select the "Repeat-insert mode" checkbox. Click 8 times in 8 different grid locations to place the 8 output pins. Rename them "HEX0[0..6]", "HEX1[0..6]", …, "HEX7[0..6]." The two dots ".." inside the braces are important; they tell Quartus that this name refers to a parallel bus (array) of 7 signals, numbered 0 through 6.

5. Next, we need a constant high logic voltage source. This is called "vcc". Double-click on the grid and add it.

6. Next, we need a constant low voltage source. This is called "gnd". Double-click on the grid and add it.

7. Finally, we need a logic inverter to generate the signal that's logically complementary to SW0, which will be needed to control some of the segments. The inverter symbol is called "not". Double-click on the grid and add it.

8. Now, we need to create and name some circuit nodes (wires). Click and drag on the end of the line coming out from the SW[0] symbol to extend it into a wire. While this new wire is still selected (highlighted blue), before doing anything else, type a short name for it (just start typing, you don't have to click anything first). Do the same for the lines coming out of the inverter, the vcc icon, and the gnd icon.

9. Next, click the end of the line coming out of SW[0] and drag it onto the inverter input line. This connects the inverter's input to the signal from the switch, so that the output of the inverter will be the logical complement of SW[0].

10. Finally, we need to define the signal buses (groups of wires) feeding each of the output display digits. Click the Orthogonal Bus Tool button – it has a picture of a thick L-shaped line. Then click and drag away from the line on the first output pin HEX0[0..6]. Now we'll need to type a name for it. This name can just be a comma-separated sequence of input signal names, to specify in what sequence input signals should be fed through to form that output bus. For example, suppose the vcc signal is named "H", then if you give the bus that goes into HEX0[0..6] the name "H,H,H,H,H,H,H", all 7 segments of digit 0 (the rightmost digit) will be wired to vcc. If gnd is named "L", then if you name it "L,L,L,L,L,L,H", the 1st six segments (segments 0-5) of the digit will be wired to gnd, and the last one (segment 6) will be wired to vcc. (What decimal digit will this display?) Finally, by also using the switch-dependent signals SW[0] and its complement (whatever your names for those signal nodes are), you can give the HEX0 bus a name that causes the rightmost 7-segment display to show one digit (say "5") when SW1 is up, and another one (say "9") when SW1 is down. (Figure it out!) In this way, you can implement the entire application by just giving appropriate names to each

of the eight different 7-bit-wide output buses. If you already did your planning (previous section), entering these names should only take a couple of minutes!

11. At this point, your design is finished, and you can try to compile it. Save your Lab0.bdf (schematic) file and select Processing → Start Compilation (or click the purple rightwards-pointing triangle icon in the toolbar, which is also shown in the menu item, or just hold down the Ctrl key and hit L). Usually, warning messages that appear can be safely ignored. If there are any errors, you can modify your circuit and try again.

12. Next, we'll program your design into the actual board in the lab. Be sure to bring your lab notebook as well your electronic folder of project design files (copied onto a USB stick, say) with you to lab so you can continue the assignment there.

You do not have to turn in a formal pre-lab report this time, but if you have time before lab, you may want to write one anyway, as practice for later assignments, and because you will anyway need to write up a lab report after the lab, and most of the material in it can just come from the pre-lab report.

5. In-Lab Assignment

In this part of the assignment, you'll program the circuit you designed in the pre-lab into the actual boards in the lab, and test it to see if it works as expected.

1. Be sure that the board at your station is plugged in to the PC via the USB Blaster connector. Plug in the 9V power supply. (Make sure it is really 9V first!)

2. Start up Quartus on the lab PC, and open your project using the *File* → *Open Project*... menu item. (Or you could do both steps at once by just double-clicking your Lab0.qpf file.)

3. Select *Tools* → *Programmer* to bring up the Programmer window. This utility "burns" configurations onto the FPGA.

4. If the "Hardware Setup" icon is not already followed by the words "USB-Blaster [USB-0]," Click "Hardware Setup…" and in the resulting window pull down "Currently selected hardware", select "USB-Blaster [USB-0]" and hit Close.

5. Make sure the Mode setting in your Lab0.cdf window says "JTAG."

6. Now, in the row of text that begins "Lab0.pof", click the checkboxes under the columns "Program/Configure" and hit the Start button. You should see successful output something like the following.

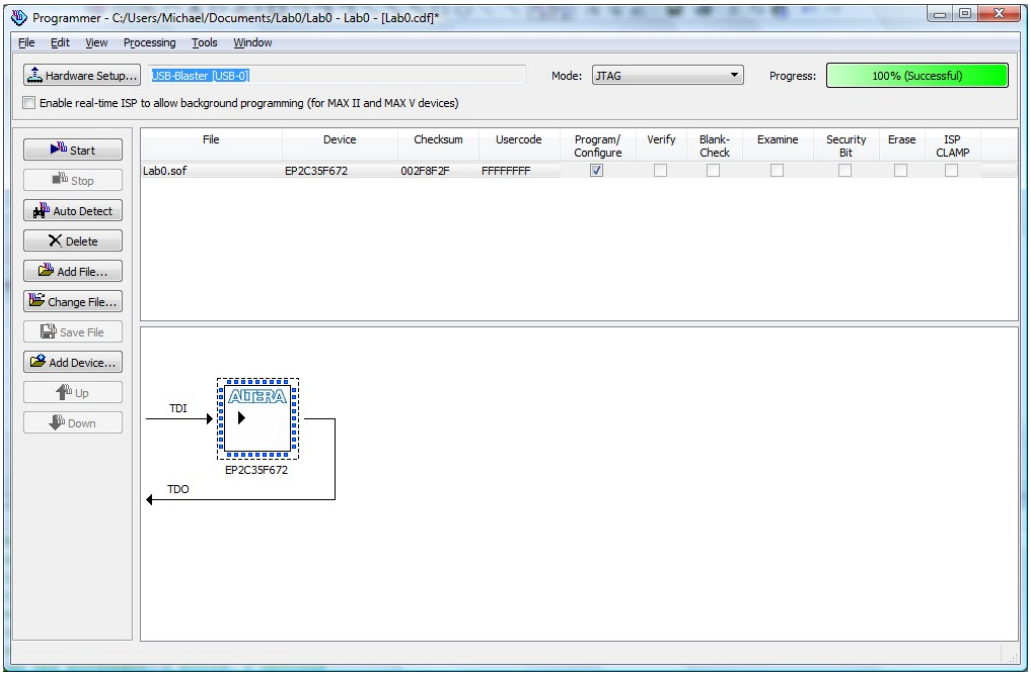

Figure 5. Example of successful completion of the process of programming our design into the board. (That's not to say that our design is necessarily correct yet!)

7. Now observe the seven-segment display on the actual board, and see what happens when you move the SW0 slider switch. If you have done everything correctly, you should see the two birthdays as expected. If you don't see the expected result, then take notes on what you did see, modify your design as appropriate, recompile it, and reprogram it into the board. (You don't have to redo the hardware setup or close the .cdf Programmer window.)

At the end of the assignment, you should show your TA your results, and make sure that you have sufficiently detailed notes in your lab notebook in order that you will be able to write a high-quality report on what you did. When writing up your report, please follow the detailed requirements given in the *General Document Requirements* and *General Lab Requirements* documents, which are available in the Assignments area on Blackboard. Make sure your report fully & clearly explains how you came up with your solution.

END OF DOCUMENT