Edreese Basharyar
Taylor Finelli
CPE 200L-1001
Dr. Sarah Harris

Final Project: Decimal-To-Binary Conversion Game

The purpose of this project is to design a collection of modules and programs that would allow for a DE2-115 FPGA board to test the user's knowledge of unsigned decimal-to-binary number conversion. All switches should be in the downward position before powering on. Once powered on and the start button is pressed, the FPGA board will display the current amount of points (initialized at 0) earned using the two leftmost seven-segment displays (HEX7 and HEX6), a countdown timer (initialized at 99) using the two rightmost seven-segment displays (HEX1 and HEX0), and the decimal number (initialized at 5) using the two center seven-segment displays (HEX5 and HEX4).

The goal of the user is to convert the decimal number currently shown on the display to its equivalent binary value using the five rightmost switches on the bottom of the FPGA board. On the DE2-115 board, these five switches represent the five bits of the binary value; the five switches are SW4-SW0 from left to right, where SW4 will be the most significant bit. Switches in the downward position represent a zero, while the switches in the upward position represent a one. With this representation, the user will use their knowledge and experience of number conversion to translate the decimal number shown in the seven-segment display. For the first unsigned decimal number, the countdown timer will stay at 99 so that there is no pressure when they are running the program for the first time. Once the user is comfortable with their decision, they will press the push-button, which conveys that the user has locked in their conclusive answer. However, the timer will activate and start counting down by a second using the 1Hz frequency given by the oscillator. During the same time the user locks in their conclusive answer, the user will gain a point if their answer is correct, and a new decimal number will appear on the display. However, if the user's conclusive answer is incorrect, the LEDs above the switches will turn on if their respective switch is incorrect. Then, the user has to flip the incorrect switches in the correct position and lock in the answer again. Any answers locked in after the LEDs light up will not receive points.

The user will have to go through ten total conversions, and the numbers will become larger and larger as the user advances through the program. The countdown timer will also place more pressure depending on the amount of points that the user has accumulated. If the total score is from 0 to 3 points, then the countdown timer will decrease by one every second. However, if the total score is from 4 to 6 points, the countdown timer will decrease by two every second. Finally, if the user has a score from 7 to 10 points, the countdown timer will decrease by three every second. With these features, the program will be valuable for users of any range of experience within the field. It will be an educational experience for new and casual users, and a reasonable challenge for experienced users that will really test their skill and knowledge.

Edreese Basharyar
Taylor Finelli
CPE 200L-1001
Dr. Sarah Harris

Evolution of the Project

When comparing the initial ideas and thoughts of creating this project to the latest version, there are several ideas which have changed and strengthened over time while still focusing on the key themes and concepts of the project. One of these suggestions given in the feedback was to increase the complexity of the countdown timer so that it provides more pressure as the user advances in the program. Our initial countdown timer had a maximum of 15 seconds for each conversion, but it did not feel like it had put any pressure on the user after many test trials. So, we reformed the countdown timer by giving it a base initialization of 99 seconds. Although it may seem like it is more than enough time to complete these conversions successfully, this is where we increased the complexity of this timer. Instead of counting down by one every second, we decided the timer should decrease by a larger value as the user accumulates more points. So, we did many trials on the board and evaluated the proper ranges for when the timer will decrease. For example, if the user has anywhere from 4 to 6 points, the timer will decrease by two every second. For 7-10 points, the timer will decrease by three every second. Even with a lot of experience with the conversions, we felt a significant amount of pressure when facing the timer and evaluated that the complexity of the timer has increased properly.

Another main suggestion incorporated in the feedback was to track the performance of the user. To accomplish this, we have set the total number of conversions to a reasonable amount of 10. This number provides a nice medium for the user to be able to understand their abilities, while also not taking too long to finish the conversions. The user should, theoretically, be able to give themselves a grade on their decimal-to-binary conversion abilities in less than 100 seconds. With the implementation of the score counter, the user is able to get real-time feedback of how they are performing as the test goes on via the two leftmost seven-segment displays. If the user does exceptionally well and completes all 10 conversions on the first try, the total amount of points displayed will be 10. With these changes, the program is more appealing to the user as it provides a challenge to those looking to get a perfect score.

Edreese Basharyar
Taylor Finelli
CPE 200L-1001
Dr. Sarah Harris

Challenges and Solutions

Throughout the creation of this project, there were many lessons we learned that helped us advance towards the project's final version. One of the first obstacles that we encountered was understanding how to connect all the submodules together. After looking in-depth on how to properly connect these modules, we realized that these modules can be connected through the process of instantiation. This process led to the creation of a top-level module for our project instead of having separate submodules without any relations. With this new understanding, we improved on this concept by linking the ports with their name instead of instantiating them by order; this process makes sure all ports are connected properly without worrying about the order of the ports themselves. Then, we learned to create logic signals inside the top-level module that would help connect the output of one module and the input of another module together. These internal signals provided more progress for our project as we finally had the chance to connect internal signals from all of the submodules together.

Another obstacle that we encountered was figuring out how to combine both sequential and combinational logic based modules so that the values of the signals flow in the proper way. This led to the idea of creating a variable within the "InputChecker" module that would interact with the sequential logic in a manner that would allow combinational logic to flow in the same manner as sequential logic. This particular variable is known as "i" inside the module, and it allows the module to flow in a way where the next decimal number can be shown without disturbing the values of the scores or the LED values. Without this variable, the new decimal number would show up at the same time and the answer would automatically be incorrect as there is no time to convert the decimal value.

However, we continued to face more obstacles as we advanced further into the project. In an attempt to keep the project as simple and user-friendly as possible, the pushbutton is used to control most of the functions involved in the project. The dependence of the pushbutton makes it very complex to create modules that all flow together. Furthermore, all of the displays, the timer, the amount of points, and the change in decimal numbers all depend on the single pushbutton. Although it was difficult to revolve multiple modules around the pushbutton, we eventually figured out how to create the submodules in a way where they work with each other to provide an undisrupted flow.

Edreese Basharyar
Taylor Finelli
CPE 200L-1001
Dr. Sarah Harris

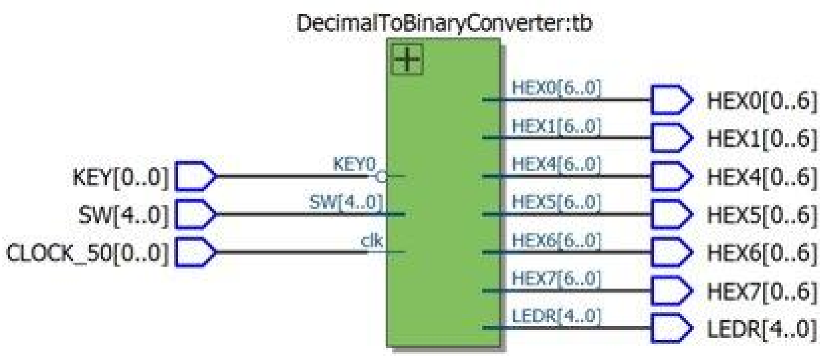
Future Improvements

For the time being, we are satisfied with how the current version of our project turned out. However, we feel that with more time and knowledge, this project could be more advanced and efficient than ever. The first feature we would add to this project would be the ability to change between unsigned binary and two's complement numbers. This feature would really open up the capabilities of this project because it unlocks the ability to bring in negative numbers. This is important because computers generally use two's complement to represent negative numbers, instead of using signed binary to represent negatives. Another feature we would like to add to this project would be the ability for the user to choose between different bases. So, instead of only converting to base 2, the user could convert to base 8 or 16. This would deeply test the user's knowledge of binary numbers. Finally, we would have also liked to add a feature that could change the mode of this project. The first mode would be what we have now, which is a timed mode so the user can see how many numbers they can get in 99 seconds. The new mode would allow the user to keep converting numbers until they get one wrong. There would be a timer of between 5 and 10 seconds for the user to convert the number to binary, and the user can convert an unlimited amount of numbers until they make a mistake. This would also require us to implement a random number generator. Currently, we use a fixed set of numbers for the user to convert. A random number generator would make the test more accurate, as the user cannot just remember the specific numbers to convert. Overall, we are pleased with the final version of our project, but we feel that with some more time, we could have the capabilities to reach a higher standard.

Edreese Basharyar
Taylor Finelli
CPE 200L-1001
Dr. Sarah Harris

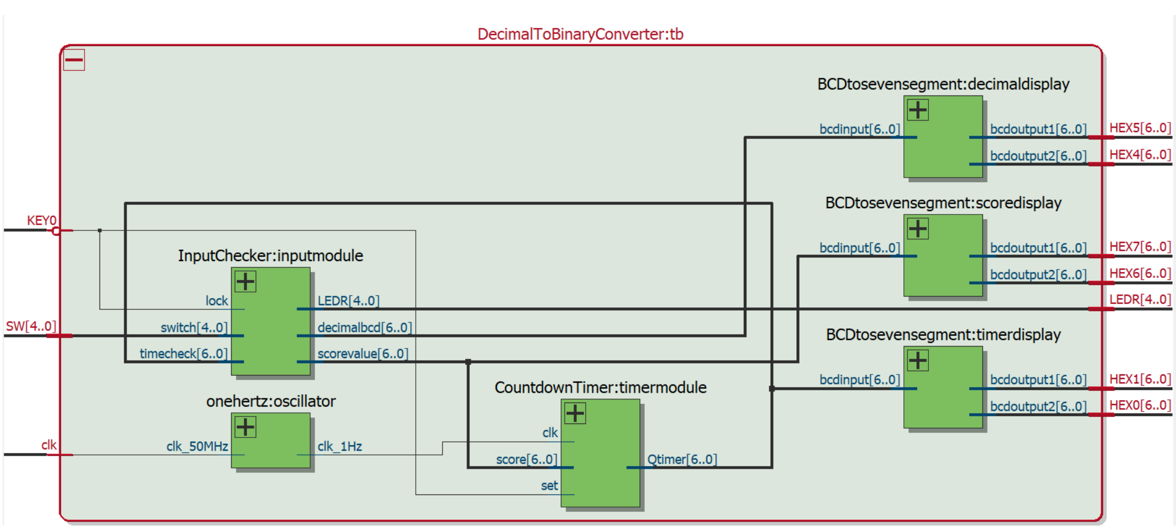
Top-Level Module Diagram

Below you will see the top-level diagram of the entire project. KEY[0] is the pushbutton that the user presses to start the board and lock in their answers. Note that the pushbutton is inverted because it is initially assigned to have a high value when it is depressed, so we need to invert the pushbutton so that the user controls the rising edge of the clock when they press it. SW[4:0] are the switches that the user flips to show the 5-bit binary representation of the decimal number that the user must convert. CLOCK_50 is the oscillator input from the DE2-115 board itself. This oscillator is converted to a 1Hz output which is used for the countdown timer. The HEX output on the right side are the output displays for the numbers. HEX0 and HEX1 are the outputs for the timer; HEX4 and HEX5 are the outputs for the number to convert; HEX6 and HEX7 are the outputs for the score. LEDR[4..0] is the output for the LED lights above the switches.



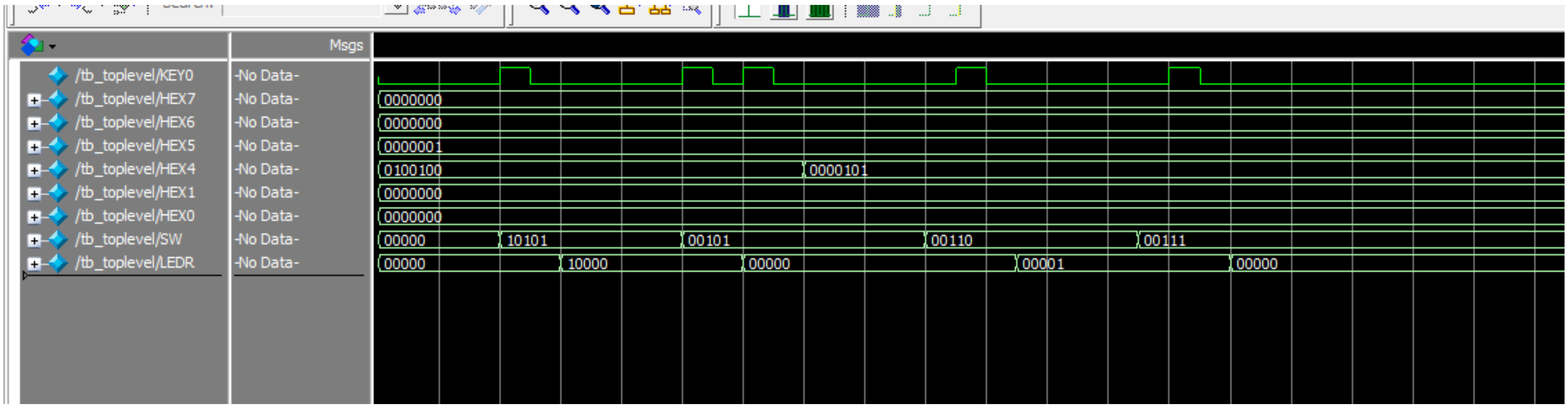
Lower-Level Module Diagram

The diagram below shows each individual submodule within the top-level module. The "InputChecker" module is responsible for taking the input from the user and verifies if the answer is correct. The "onehertz" module uses the 50MHz oscillator within the DE2-115 board and creates an output of 1Hz. This 1Hz output is then input into the "CountdownTimer" module which generates an automatic clock for the down counter. This clock will activate the 99-second timer once KEY[0] is pressed for the final answer of the first conversion. The module "BCDtosevensegment" outputs the numbers (score, decimal number to convert, and timer) to the seven-segment displays. Notice how this module is called three times as each call outputs different numbers to different displays.



Edreese Basharyar
Taylor Finelli
CPE 200L-1001
Dr. Sarah Harris

Simulation Waveform of Top-Level Module + Description



The waveform above shows two iterations of a user converting the given decimal number to binary numbers via the switches. In the first iteration (*before* KEY0 reaches 1 the second time), you will see that HEX5 = 0100100 and HEX4 = 00000. This is the 7-segment display representation of the number 5. You will also see that SW goes from 00000 to 10101. This is the user changing the switches to represent the binary number 5. After the user locks in their answer, the LEDR value has a value of 10000, which shows the user that SW[4] is in the incorrect position. Following this, SW changes from 10101 to 00101, which represents the user changing the position of SW[4] to 0. KEY0 is pressed again which locks in the answer and presents a new decimal number on the HEX5 and HEX4 displays. On the second iteration (*after* KEY0 goes to 1 the third time) SW goes to 00110. This represents the user's attempt at converting the number 7 (now shown on displays HEX5 and HEX4). KEY0 is then pressed again which shows the user that the SW[0] switch is in the incorrect position, as LEDR[0] is lit up. SW goes to 00111 which represents the user fixing the switches. KEY0 is pressed again and the LEDR[0] turns off, shown in the waveform as 00000. Note that the displays HEX0, HEX1, HEX6, and HEX7 all stay at zero. HEX0 and HEX1 stay at zero because it would take too many cycles for the oscillator to function properly in the testbench; HEX6 and HEX7 stay at zero because the "user" did not get either of the answers correct on the first attempt and therefore was not awarded points.