Department of Electrical & Computer Engineering

ECCE5214: Advanced Logic & Computer Interfacing

PS/2 protocol decoder & BONUS task

Team Members:

Al Muala Talal Al Maawali 135591)

Omar Masoud AL Alawi 123901)

Contents

Table of Contents

introduction

In this project, we aspire to build. An efficient method to decode Basic keyboard inputs Into Binary numbers We believe that running such projects on FPGA would make it extremely efficient and would help to decode such Logic. One of the most important Constraints in this project is to understand the limitation and. The PS/2 Protocol. The PS/2 protocol is old-fashioned protocol used to Interface keyboards to many devices. it Uses 11 bits of DataStream to transmit all the keyboard inputs. Understanding this makes it noticeably clear that the end result and the final output of this project should be a device and a logic that the end user can use to insert certain characters and the logic should output its binary representation. Getting the Binary representation of the PS/2 Is extremely helpful and can be used in different applications. In fact, we used it in a remarkably interesting application that we will explain later. The most important requirement for this project is to find an efficient and reliable module to interface with PS/2 port keyboard. Using the FPGA DE-01 A certain logic would be Implemented to get our desired output.

Design Specifications

Despite the extreme complexity in this project, the inputs and Outputs are relatively simple. The PS/2 board only requires two pins to work. One of them is PS/2-clock. The other one is PS/2-data. Both Pins will supply all the required inputs to Implement this project logic. Another important input in this project is a simple switch that will be used to implement the loading animation on the bonus task. The PS/2 clock will be used to synchronize the keyboard with FPGA while the PS/2 data will be used to send the data from the keyboard to the FPGA. The switch will simply control the mode of operation. Either to work as encoder, or to show the loading animation on the bonus task. Last one would be the clock where we used the first clock in the FPGA that signals 50M Hz signal to control the logic.

Figure 1: INPUTS & OUTPUTS

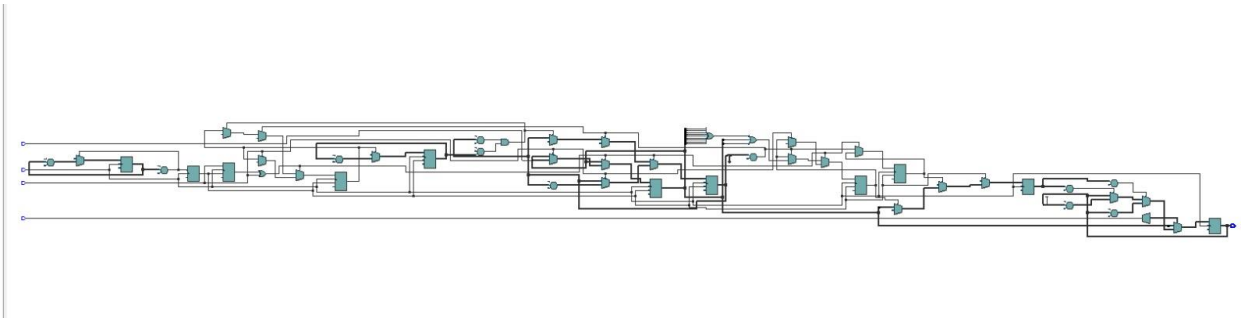

Figure 2: RTL view of the logic

As the input and output figure above shows there are Four inputs and one output. The four inputs consist of the two PS2 inputs and one switch and the input clock. The only output is the LED Outputs that represent the encoded PS/2 protocol representation.

Hardware Design

Hardware wise speaking, the code is extremely efficient. There is only a single module in this project. The single module is called “prj” It controls everything in this project. It firstly Interfaces with the keyboard to Get its output to the logic. after getting the Keyboard output the Output is Processed and used in different application. in this block of code, we used a simple function to output the pS/2 binary representation into the LEDS of the FPGA. In the second part of this project, we used a simple function that can be controlled by the arrow keys in the keyboard to show the Loading animation in the LEDs. The mode of operation is controlled by the switch.

With the presence of a PS/2 interface unit, it is synchronized with the FPGA through clock synchronization in the data transfer process to prevent data loss or corruption. There is also a decoder that converts inputs into binary numbers using lookup tables to perform the conversion efficiently.

Functional Simulation

To validate the design starts from validating the functionality of the module (prj) through accurate results from the PS/2 keyboard and the correct working of the loading animation in the bounce part. In the testing section, the PS/2 clock and data input signals were stimulated to simulate the data input to the keyboard, and the FPGA clock was also simulated to verify and confirm the timing of the model. After this step, we work to verify correct conversion to binary representation on the FPGA board, and the control switch is made to switch to the loading animation mode and it operates sequentially on the LEDs in the FPGA, and the switch works smoothly to convert between the two modes and is also synchronized with the FPGA clock. Fast typing on the PS/2 keyboard is simulated to ensure timer and timing-sensitive functions are correct and figure3 shows the simulation waves of (prj) data and clk also after doing the testbenches part using Modelism.

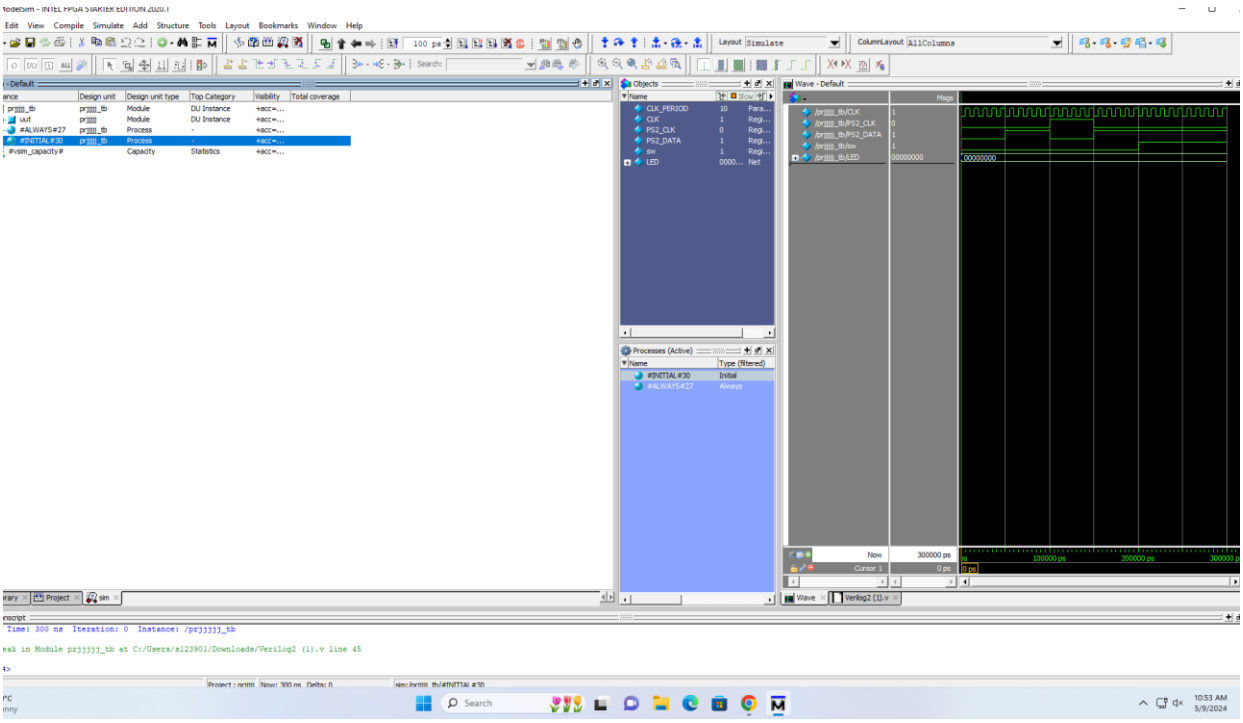

Figure 3: Modelsim Simulation

Design Implementation

The project was implemented to decode the PS/2 inputs into binary numbers and appear directly on the FPGA board through synthesis, which converts the RTL of the design into a netlist of logic gates and flip-flops in order for the ps/2 to be decoded into binary numbers. In the second part, the same equation was used to control the keyboard arrow keys to show the Loading animation in the LEDs. In the final stage of implementation, we performed the Place and Route (PAR) process, which sends both designs inside the FPGA device to their specific locations on the chip.

Upon completion of the implementation of the project, we performed non-behavioral simulations to verify the correctness of the design, its performance, and its timing, and that the timing for performing the process is appropriate for the timing requirements of the FPGA device, all of this to ensure the efficient and rapid operation of the process of decoding the inputs and translating it into binary numbers on the FPGA output .

Results and Discussion

The experiments on the design were crowned with success in decoding the PS/2 keyboard into binary numbers on the FPGA and displaying loading animation in the LEDs. Figure-4 below shows the PS/2 keyboard interface and each button with its representation in hexadecimal. After the outputs appear on the FPGA, we compare them. The results were identical to the entered characters and synchronized due to the use of the PS/2 clock to synchronize the keyboard keys with the FPGA.

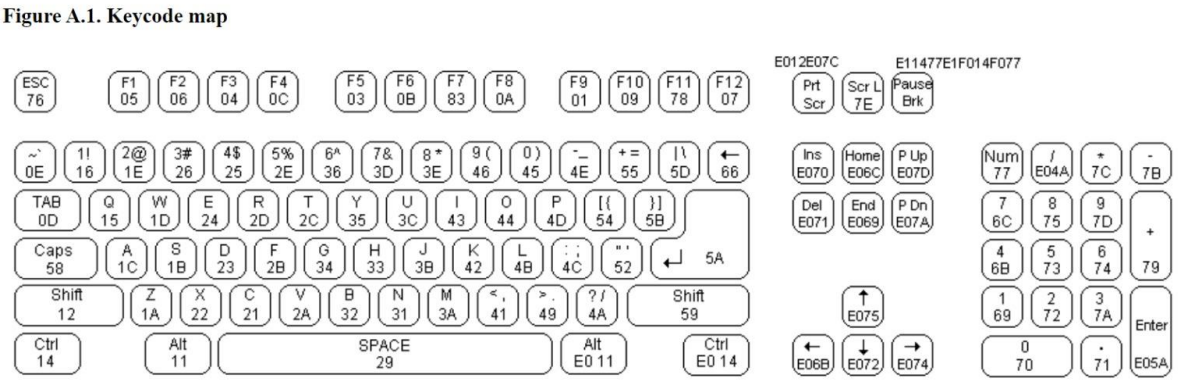

Figure 4: PS/2 Keycode map

Experiments were also conducted on the second design, loading animation. The t switch and arrow keys worked smoothly, so the user could switch between encryption mode and loading animation mode smoothly and easily. The results matching the project's requirements indicate that it was designed successfully and complies with the standards of FPGA device.

Conclusion

In conclusion, our project aims to build an effective method to decode keyboard inputs into binary numbers and display the results on the FPGA. We succeeded in this, and the limitations related to the old PS/2 keyboard protocol were understood and solved successfully. The design includes efficient logic for communicating between keyboard inputs and processing the data entered into the FPGA and converting it to binary representations. The process of synthesis, Place and route (PAR) was carried out precisely to translate the design into the FPGA DE-01, and to ensure the correctness of performance, timing, and power consumption, a non-behavioral simulation process was also conducted.

After the experiments conducted on the FPGA device, we demonstrated the ability to communicate with the PS/2 keyboard and decode it, in addition to the success of controlling the animation loading, which shows the diversity in design and the possibility of switching from one design to another using the switch.