MNE3002 Assignment 10/1/2021
FPGA Interfacing: PS2 Keyboard Luca Ruggier

Aim: To interface a PS-2 Keyboard to the 7-segment Display on the FPGA. When a key is pressed on the PS-2 keyboard, the associated symbol should be displayed on the 7-segment display found on the FPGA.

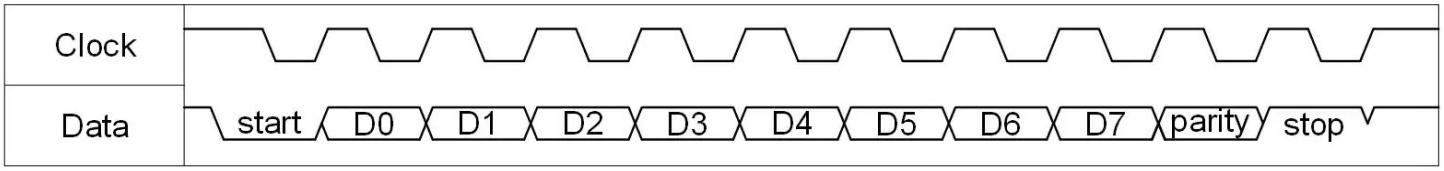
Equipment: Basys 2 FPGA
PS2 Keyboard

Assumptions Made: The environment and communication media on which the FPGA and PS2 Keyboard communicate would have little to none interference or errors, this is referenced later.

Resources:
1. The supplied .UCF File (Modified)
2. Basys-2 Reference Manual
3. Per-key Hex Codes from Digilent
4. Basic PS2 Protocol Diagram from DigiKey
5. Seven Segment Truth Table from BlogSpot

Files:
1. PS2.vhd
2. PS2.ucf
3. Keycode.xlsx

Description:



(Resource 4: Basic PS2 Protocol Diagram)

The PS2 interface does not use the system / host clock, thus it was preferable to rather read the PS2 clock as an input, instead of reading the data periodically governed by the system clock.

Using the above diagram, it was noted to read data bits on the falling edge of the PS2 clock as the falling edge occurred roughly in the center of each data bit. The FPGA waits for the first falling edge of the PS2 clock and a low data bit, which denotes the start of transmission. The next 8 falling edges occur and the FPGA reads and stores each bit. The 10th data bit is the parity bit, however it was assumed that there was little to no interference on the communication media, and thus this parity bit was ignored. In testing, this did not hinder results. The next bit signified a stop bit, denoting that the communication has completed.

The FPGA ran a basic state machine, consisting of a waiting state, 8 data states, and a stop state.

The FPGA would remain in the waiting state until the start of transmission was detected. It would then traverse all 8 data states as it read the 8 data bits, then head to the stop state. Once in the stop state, when the end of transmission was detected, it prints the data to the LEDs and Seven Segment, and goes back into the waiting state.

The per key hex codes from resource 3 were entered into an excel spreadsheet, and converted to 8 bit binary using the built in function =HEX2BIN([hexcode],8)
The data read from reading the 8 data bits was entered into a switch statement, in order to display the correct information on the seven segment display.

As the data was already in 8 bit form, and the FPGA had 8 LEDs on the development board, the data was directly written to the 8 LEDs.

The UCF file was modified in such a way that all pins for the PS2 interface, LEDs and Seven Segment Display were correctly assigned.