**CS223**

**DIGITAL DESIGN**

**PROJECT FINAL REPORT**

**BERK YILDIZ**

**21502040**

Table Of Contents

1.0 Project Description3

2.0 Mapping Table4

3.0 Changes from Progress Report to Final Report...................................................................4

4.0 Block Diagram.....................................................................................................................5

5.0 Implementation....................................................................................................................5

5.1 Implementation of LFSR.........................................................................................5

5.2 Implementation of keypad4x4.................................................................................6

5.3 Implementation of Comparator................................................................................6

5.4 Implementation of SevSeg\_4digit...........................................................................7

5.5 Implementation of stepmotor...................................................................................7

5.6 Keyboard..................................................................................................................8

5.7 Case Statements........................................................................................................8

5.8 7-Segment 4 Digit.....................................................................................................8

**1.0 Project Description**

The project is the implementation of a game called ‘Game of Codes’. According to the game, the player watchs the movements of stepper motor and enters the corresponding ASCII character sequence using 4x4 keypad on the Beti Board. For each correct answer, player earns 1 point and for each wrong answer player loses 1 point. The score of the player is displayed on the 7-segment display module on the BASYS-3.

For building up the game, first of all a mapping table is needed to assign a unique code to each button on 4x4 keypad. According to assigned codes, the stepper motor will carry out its movement. A code is composed of two parts and each part represents one of four movements: short right, long right, short left and long left.

For the random sequence generation, a pseudo random generator is needed. A linear-feedback shift register is goint to be used for generating pseudo-random sequence.

Stepper motor module which is available on Beti Board plays the each sequence according to code that generated by LFSR.

Player enters the recognized code in to 4x4 keypad which is available on Beti Board.

Player is able to see his/her score at the 7-segment display module on the BASYS-3. The minimum score is ‘0’ and the maximum score limit is ‘9’.

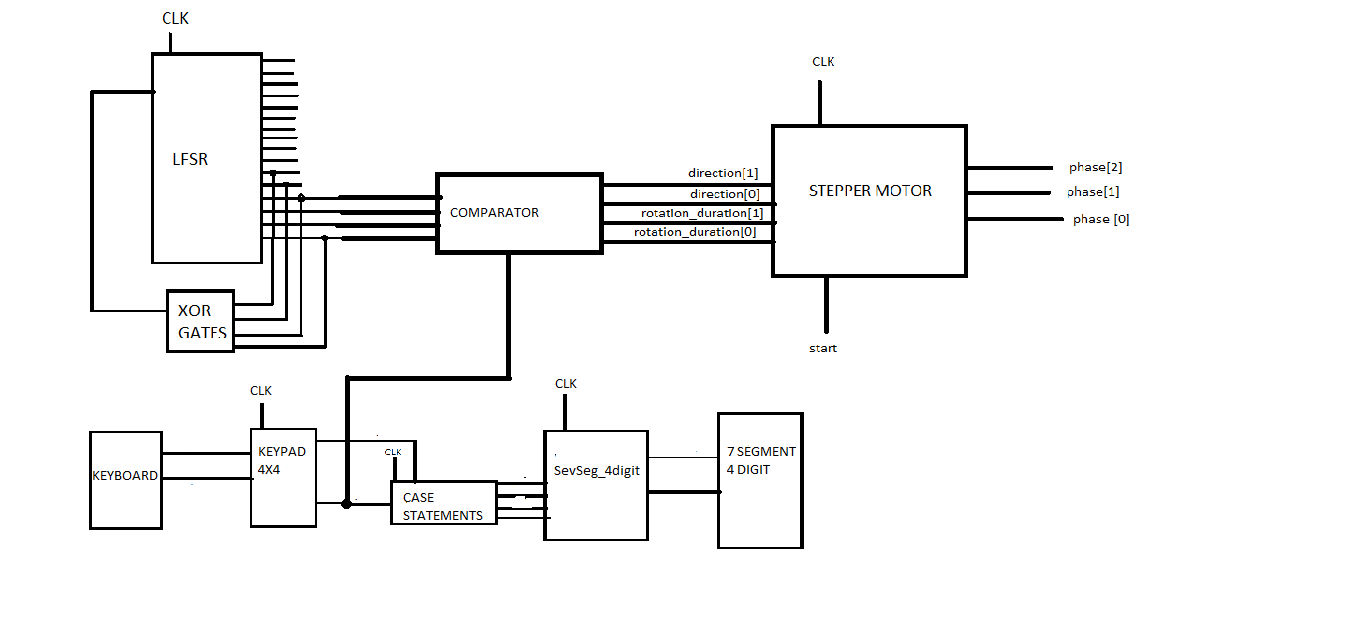
**2.0 Mapping Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **0** | LL-LS | **8** | SL-SL |
| **1** | SR-SR | **9** | SL-LL |
| **2** | SR-LR | **A** | LL-LL |
| **3** | SR-SL | **B** | LR-SR |
| **4** | SR-LL | **C** | SL-SR |
| **5** | LR-LR | **D** | LL-SR |
| **6** | LR-SL | **#** | SL-LR |
| **7** | LR-LL | **\*** | LL-LR |

**3.0 Changes from Progress Report to Final Report**

While I was creating the block diagram of the project for the progress report, I could not provide the connections between LFSR, stepper motor and keyboard. However, while the lectures goes on, I learnt about the comparator and it is able to compare the output code of LFSR with the keyboard for the increasing and decreasing of the score on 7-segment display module. Thus, differently from the progress report I tried to include and implement the comparator in to my high-level design.

**4.0 Block Diagram**

****

**5.0 Implementation**

**5.1 Implementation of LFSR**

LFSR is the module that generates the pseudo-random code sequence. For the project an 16 bit LFSR is used to have sufficient number of random codes. Only last 4 outputs of the LFSR is used. 3 XOR gates are used to provide variety. For the implementation of XOR gates, the inputs of the first XOR gate are 16th and 13th outputs of the LFSR. The inputs of the second XOR gate is output of the first XOR gate and 12th output of the LFSR. The inputs of the third XOR gate are output of second XOR gate and 11th output of the LFSR. The output of the third XOR gate is connected as feedback to the LFSR again for starting the generation process again.

**5.2 Implementation of keypad4x4**

module keypad4X4(

input clk,

output [3:0] keyb\_row,

input [3:0] keyb\_col,

output [3:0] key\_value,

output key\_valid

)

• **clk :** BASYS-3 system clock (100Mhz).

• **keyb\_row, keyb\_col :** Using these ports, SystemVerilog module is connected to physical keypad. Then you need to connect them to FPGA pins.

• **key\_value, key\_valid :** When any key of pinpad is pressed for long enough time, key\_valid becomes '1' for just one clock cycle. At the same time, the value of key\_value holds the code of the pressed key ( {row[1:0], col[1:0]} ).

By the inputs, the module becomes connected with keyboard. The ‘key\_value’ is connected to comparator and case statement. ‘key\_value’ is going to be compared with the output LFSR to determine the accuracy of the player’s choice.

**5.3 Implementation of Comparator**

module comp(a,b,aeqb);

input [3:0] a,b;

output [3:0] c;

output aeqb;

reg aeqb;

This would be 4-bit comparator that determines the accuracy of player’s choice. The inputs of the comparators are coming from the LFSR and keypad4x4. The output of the comparator will reach to stepper motor for the movement of it.

**5.4 Implementation of SevSeg\_4digit**

module SevSeg\_4digit(

input clk,

input [3:0] in0, in1, in2, in3,

output a, b, c, d, e, f, g, dp,

output [3:0] an

)

• **clk :** BASYS-3 system clock (100Mhz).

• **in0, in1, in2, in3:** These four hex numbers are set by user and then are displayed on 7-segment.

• **a, b, c, d, e, f, g, dp, an :** Using these pins, SystemVerilog module is connected to physical 7-segment. Then you need to connect them to FPGA pins.

Inputs in0, in1, in2, in3 are set to 0 at begining. The all outputs of the module are connected to 7-segment display of BASYS-3.

**5.5 Implementation of stepmotor**

module stepmotor(

input clk,

input direction,

input [1:0] speed,

output [3:0] phases,

input start

);

* The psuedo random codes which come from 16-bit LFSR rotate the stepper motor in 4 ways (short-right, long-right, short-left, long left).
* **clk :** BASYS-3 system clock (100Mhz).
* **Direction :** user input for motor rotation directions. Direction[0] is direction of first movement and direction[1] is direction of second movement.
* **rotation\_duration:** user input for motor rotation duration. Rotation\_duration[0] is duration of first movement and rotation\_duration[1] is duration of second movement.
* **phases:** Using these ports SystemVerilog module is connected to motor. Then you need to connect them to FPGA pins. SW14 and SW15 (two left-hand side switches on BASYS-3) set the rotation durations. SW12 and SW13 set the rotation directions.
* **Start:** user input to initiate motor movement. A pulse (at least one clock cycle) starts 2 movements of motor to represent a code. Direction and duration of both movements are captured together at the time of applying start command. If you re-apply start or change the value of direction/durations before end of both movements, they are ignored. To play each code, you need to assign correct values of direction/duration from mapping table to inputs. Then apply start command.

The stepper motor module which is available on the Beti board play each code by moving the motor according to the character mapping table defined above. For example, for character ‘1’ with code LR-SL, motor first longly rotates right, then takes a short rotate to left. Motor should wait for a few moments between each code, so the player person can differentiate successive codes.

**5.6 Keyboard**

The physical module which is available on Beti board with including 4x4 keypad. User is going to enter his/her choice to the keyboard.

**5.7 Case Statements**

Determines the behavior of the outputs of keypad4x4. According to the project each button on keypad is able to increment the 7-segment display by 1 or decrement the 7-segment display by 1 according the accuracy of player’s choice.

**5.8 7 Segment-4 Digit Display**

A pyhsical module which is available on BASYS-3. The score of the player is going to be displayed on this module. The score of the player can not be lower than 0 and higher than 9.