**CS223**

**DIGITAL DESIGN**

**PROJECT**

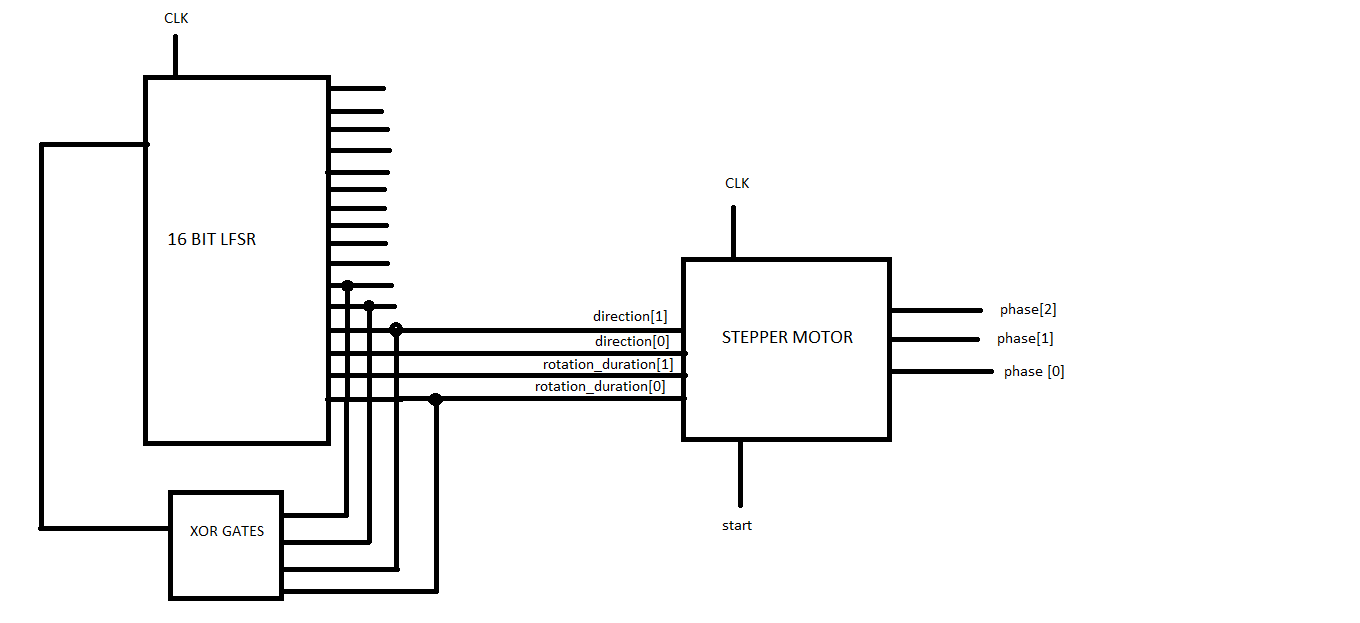
**PROGRESS REPORT**

**BERK YILDIZ**

**21502040**

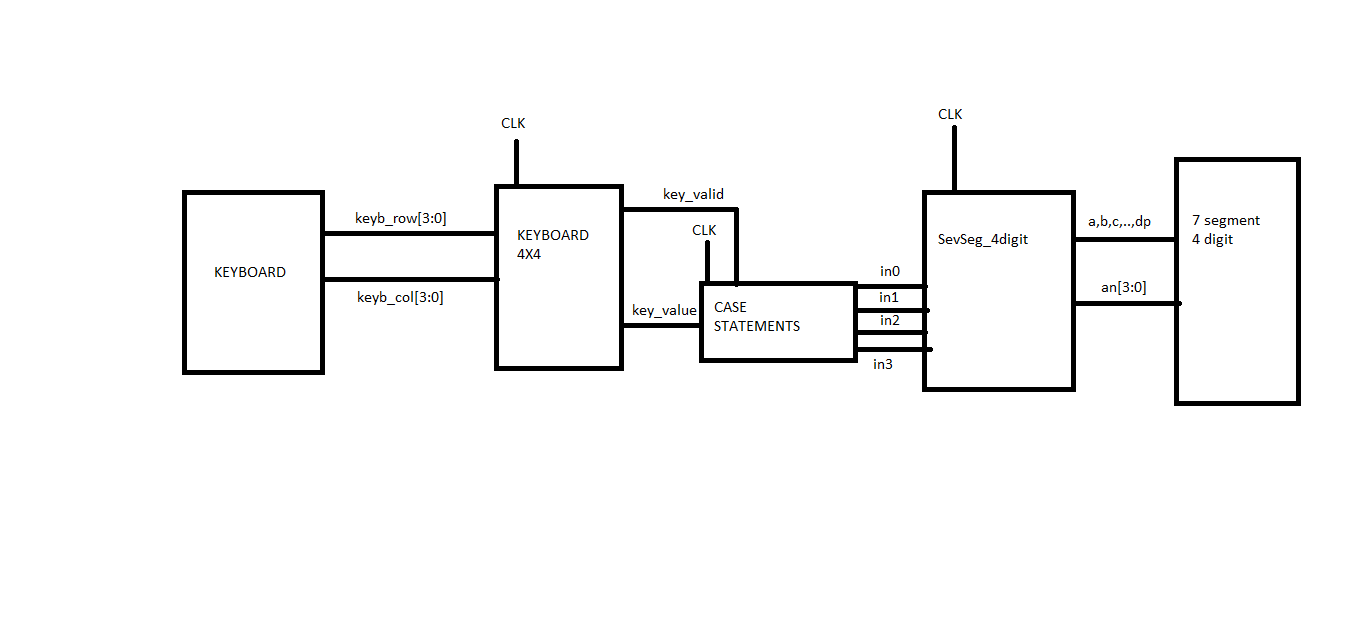
**PACK NUMBER 31**

**Block Diagram for Movement of the Stepper Motor**

****

* After programming the project on FPGA, 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. ‘0’ value is left and ‘1’ value means right.
* **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. ‘0’ value is short and ‘1’ value is long.
* **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.

**Block Diagram for Keyboard**

****

* After programming the project on FPGA, pressing the true button on 4x4 pinpad, increments the 7-segments digits by ‘1’. Pressing the false button on 4x4 pinpad, decrements the 7-segments digits by ‘1’.

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]} ).

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.

**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 |