

Bilkent University
Computer Engineering
CS 223 Digital Design
PROJECT FINAL REPORT

Game of Codes

By: Naisila Puka

21600336

Section 05

Trainer pack #1

21/12/2017

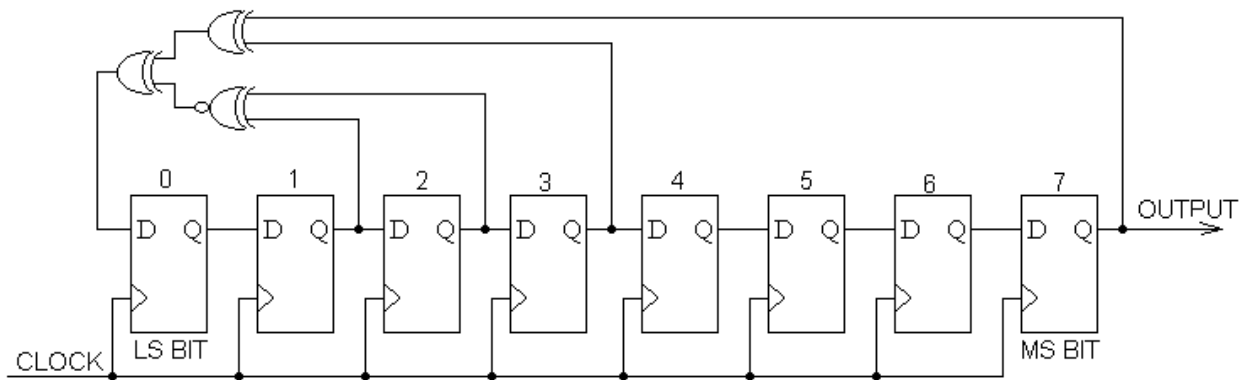
1) Mapping table

This is the randomly-defined mapping table for “Game Of Codes”. Each character from the character set of the 4x4 keypad has a unique movement code composed of two parts, where each one represents one of four movements short right (SR), long right (LR), short left (SL), long left (LL).

<i>Character</i>	<i>Code</i>
0	SR - LL
1	LL - SL
2	SL - SL
3	LR - SR
4	LR - LL
5	SR - SR
6	SL - LR
7	LL - LL
8	SR - LR
9	LR - SL
A	LR - LR
B	SR - SL
C	SL - LL
D	LL - SR
#	SL - SR
*	LL - LR

Pseudo-random generator

This is the pseudo-random generator logic diagram of the “Game Of Codes”. It is implemented using an 8-bit linear-feedback shift register (LFSR). As it is seen from the diagram, the output of the second and the third flip-flop are inputs to an xnor gate with output 1 and the output of the fourth and the eighth flip-flop are inputs to an xor gate with output 2, and then 1 and 2 are inputs to another xor gate, whose output becomes the input of the first flip-flop. This is how the pseudo-random sequence is generated. For the game, I will use the 4 least significant bits of this 8-bit output.

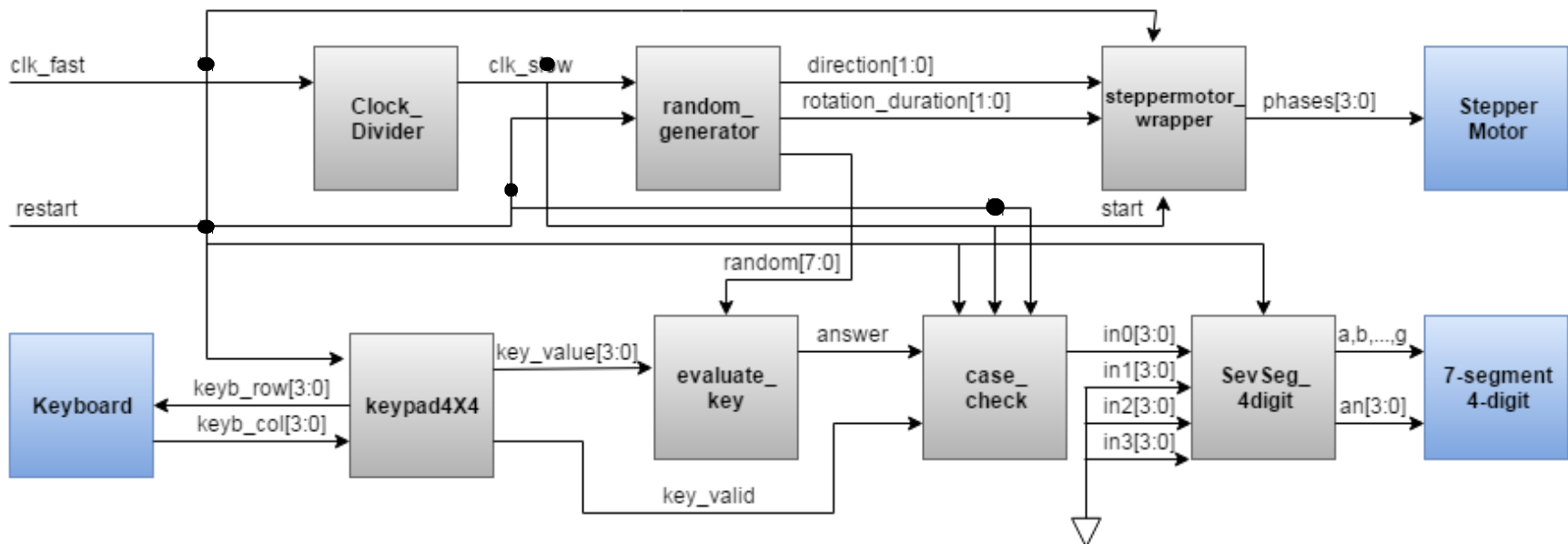


High Level Block Diagram

clk_fast is BASYS-3 system clock (100Mhz)

clk_slow is ~1.25dHz (approximately 8 seconds clock period)

I have used the same modules as in the progress report. I have also added the clock_divider module from Lab4. The user has approximately 8 seconds to answer until the next motor movement.



My modules' functionality and interfaces

This is the module for the Pseudo random generator. It generates sequence of 256 values, each of them 8-bit. For the game, the 4 least significant bits of each value will be considered. The random generator was chosen this way in order to make the codes as “random” as possible. Also, the values of direction and rotation of the motor will be assigned in this module as {random[0], random[2]} and {random[1], random[3]} respectively.

```
module random_generator(  
    input logic clk_slow,  
    output logic [1:0] rotation_duration,  
    output logic [1:0] direction,  
    output logic [7:0] random  
);
```

- clk_slow: slowed down clock from BASYS-3 system clock (100Mhz) to (~1.25dHz)
- rotation_duration: values for rotation duration of the motor {random[1], random[3]}
- direction: values for direction of the motor {random[0], random[2]}
- random: 8-bit random output

This is the module for evaluating the value that comes from the random sequence generator. It uses the 4 least significant bits of the input, checks whether the key_value coming from the keypad corresponds to the random value with regard to my mapping table.

```
module evaluate_key(  
    input logic [7:0] random,  
    input logic [3:0] key_value,  
    output logic answer  
);
```

random: input that comes from the random generator. Its 4 least significant bits will be considered as the code.

key_value: This comes from the keypad module. The value of key_value holds the code of the pressed key ({row[1:0], col[1:0]}). This module will check whether key_value corresponds to the value of random[3:0] according to my mapping table.

answer: indicator of whether the answer given by the user was correct or not. It becomes 1 if the answer is correct and 0 if the answer is wrong.

This is the module for updating the points of the user in the seven segment display. It takes the answer from the previous module, and if key_valid is 1, which means the user pressed one button in the time slot (approximately 8 seconds), it will increase or decrease one point according to his answer. If the user does not press any key during the time slot, it is considered as a wrong answer and decreases one point. Also, if the user attempts to press

again another key from the keypad after having pressed one, it will be ignored. The points will increase up to 9, then anything will be ignored unless restart switch is turned on. Lastly, if the user turns the restart switch on, the points will go to zero again.

```
module case_check(
    input logic clk_fast,
    input logic clk_slow,
    input logic key_valid,
    input logic restart,
    input logic answer,
    output logic [3:0]in0
);
```

clk_fast: BASYS-3 system clock (100Mhz).

clk_slow: slowed down clock from BASYS-3 system clock (100Mhz) to (~1.25dHz)

key_valid: this comes from the keypad module. When any key of pinpad is pressed for long enough time, key_valid enters as '1' for just one fast clock cycle.

restart: this is a switch in the Basys. If the user turns the switch on, points will automatically go to zero.

answer: variable for validity of user's answer (pressed key)

- in0: This hex number is the output of the module and is passed as input on 7-segment. It is set by the user's answer validity. In this way his points will be updated.

This is the top module of the game, where I connect all the inputs and outputs of the different modules I have for the game, as shown in the high level block diagram at part 3.

```
module top(input logic clk_fast,
    input logic restart,
    output [3:0] phases,
    output [3:0] keyb_row,
    input [3:0] keyb_col,
    output a, b, c, d, e, f, g, dp,
    output [3:0] an
);
```

```

logic clk_slow, [7:0] random, answer, [1:0]direction,
[1:0]rotation_duration, [3:0]key_value, key_valid, [3:0]in0;

initial begin

in0 = 4'h0; //initial value

end

Clock_Divider divideClock(clk_fast, clk_slow);

random_generator pseudoRandom(clk_slow, restart, q, direction,
rotation_duration);

steppermotor_wrapper steppermotor_wrapper_inst0(clk_fast, direction,
rotation_duration, phases, clk_slow);

keypad4X4 keypad4X4_inst0(clk_fast, keyb_row, keyb_col, key_value,
key_valid);

evaluate_key check(q, key_value, answer);

SevSeg_4digit SevSeg_4digit_inst0(clk_fast, in0, 4'h0, 4'h0, 4'h0, a, b, c,
d, e, f, g, dp, an);

case_check points(clk_fast, clk_slow, key_valid, restart, answer, in0);

endmodule

```

References

Clock_Divider module from **Lab4 description**.

Seven segment, stepper motor and keypad modules from **Project Description**