KING MONGKUT'S INSTITUTE OF TECHNOLOGY

LATKRABANG SCHOOL OF ENGINEERING

COMPUTER ENGINEERING DEPARTMENT

DIGITAL FUNDAMENTALS SYSTEM PROJECT REPORT
Title : Character Jumping Game

INSTRUCTED & ADVISED BY:
Prof. Sorayut Glomglome

TEAM MEMBERS:
Pattamawat Pleethong (66011136)
Win Thawder Aung (66011610)
May Thu Kyaw (66011642)
Mihini Methasa Welatantrige Boteju
(66011666)

DATE OF SUB: 10/04/2024

Table of Contents

List of Figures

Motivation

This idea was motivated by the enthusiasm of platform games where players had to jump over obstacles to proceed. Making use of the Basys3 board's four seven-segment displays and push buttons, and knowledge of VHDL. We imagined developing an engaging game that combines enthusiasm for gaming with the knowledge we learned in the digital fundamental systems class.

We implement our VHDL code to detect our character's position and the locations of obstacles. VHDL senses obstacles when our character leaps, and updates the score. Our game challenges the player in real time with VHDL as perfect timing is more important than simply pressing buttons which makes the gaming experience a lot more thrilling to play. This game is an interactive and responsive one as when the player loses it displays the score for the player. We use four seven-segment displays and a push button to provide a user-friendly interface and a clear view of the game. The game is straightforward but fascinating because every display has an objective. Our goal is to create a game that stimulates interest in FPGA programming using VHDL code and seven-segment displays.

List of features

Obstacle Generation:

Generating obstacles randomly that the player character must jump over to survive in this game. The obstacle generation will happen for every 200ms randomly. The obstacles are represented using ‘e’ and ‘c’ segments of all four 7 segment displays.

Jumping over obstacles:

The player must jump over obstacles by holding the ‘up’ button for 2 seconds. After 2 seconds the player is dropped to the ground level. If the time that player drops to the ground and incoming obstacle’s time coincides there will be a game over.

Player and Obstacle Collision:

When the player fails to jump before the collision with the obstacle, on the left seven segment display there will be a ‘F’ displayed to represent that the player failed the attempt at jumping over the obstacle.

Scoring System:

The minimum and maximum score that the player can score in a one round is 0 to 15. When the player jumps over the obstacle, one point will be added and it is displayed using LEDs in the Basys3 board. Accumulated score will be output by lighting the LEDs which correspond to the score number.

Design Concept and Circuit Diagram

Top-down Design

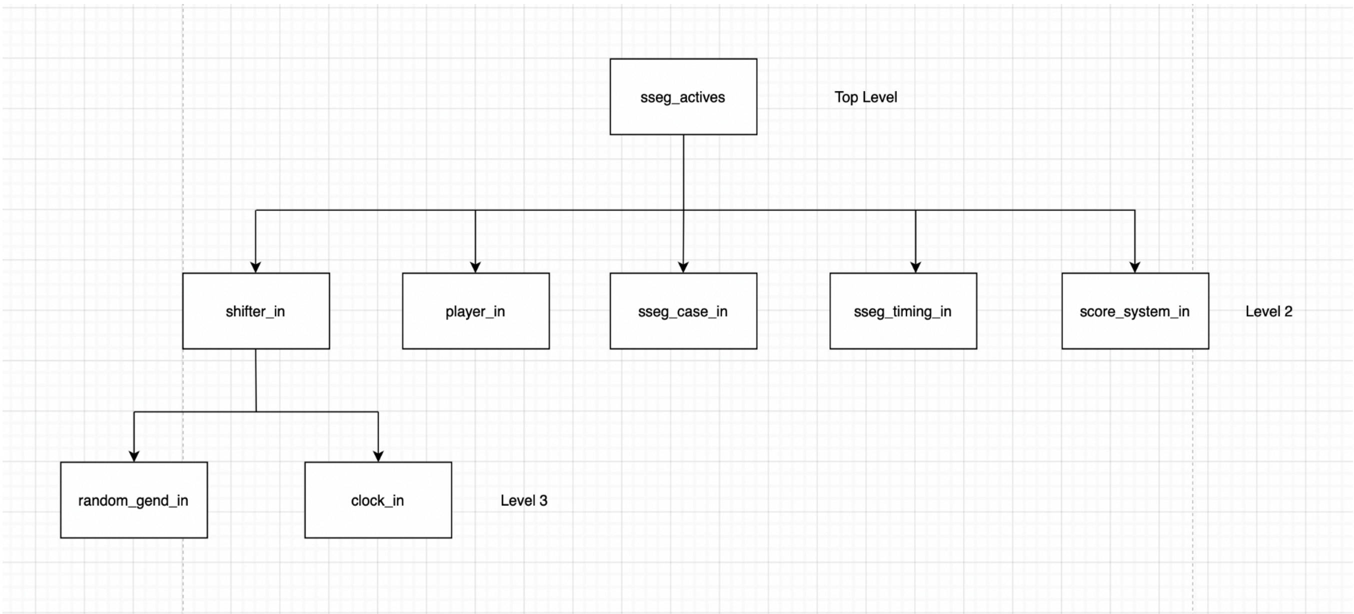


Fig 1: Top-down design of Character Jumping Game

Modular Design

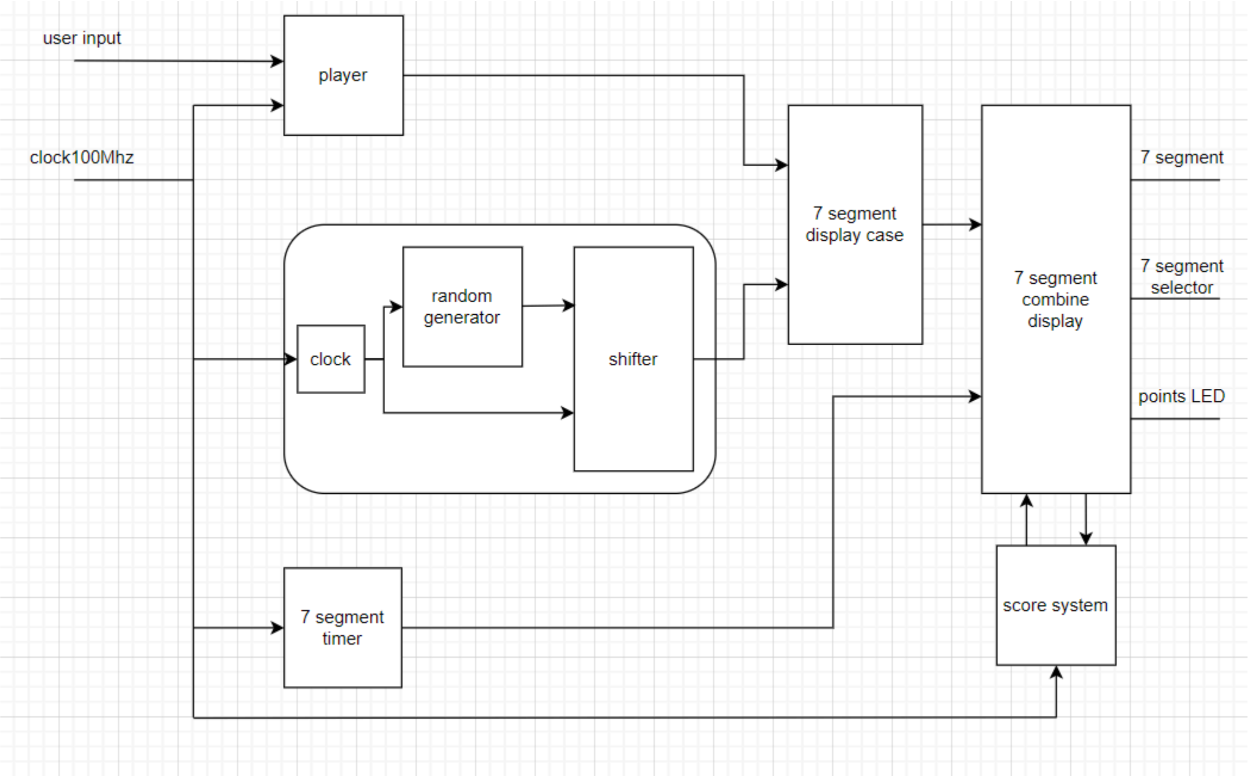


Fig 2: Modular Design of Character Jumping Game

Boolean Expressions

Seven segment case (4 to 7 decoder)

Truth Table

A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	1
0	0	0	1	1	1	0	1	1	1	1
0	0	1	0	1	1	1	1	0	1	1
0	0	1	1	1	1	0	1	0	1	1
0	1	0	0	1	1	1	1	0	1	1
0	1	0	1	1	1	0	1	0	1	1
0	1	1	0	0	1	1	1	0	0	0
0	1	1	1	0	1	1	1	0	0	0
1	0	0	0	1	1	1	1	1	0	1
1	0	0	1	1	1	0	1	1	0	1
1	0	1	0	1	1	1	1	0	0	1
1	0	1	1	1	1	0	1	0	0	1

For a:
F = NOT (B) + NOT(C)

For b:
F = 1

For c:
F = BC + NOT(D)

For d:
F = 1

For e:
F = AB + NOT(B)NOT(C)

For e:
F = AB + NOT(A)NOT(C)

For g:
F = NOT(B) + NOT(C)

Symbol Level

Seven segment case (4 to 7 decoder)

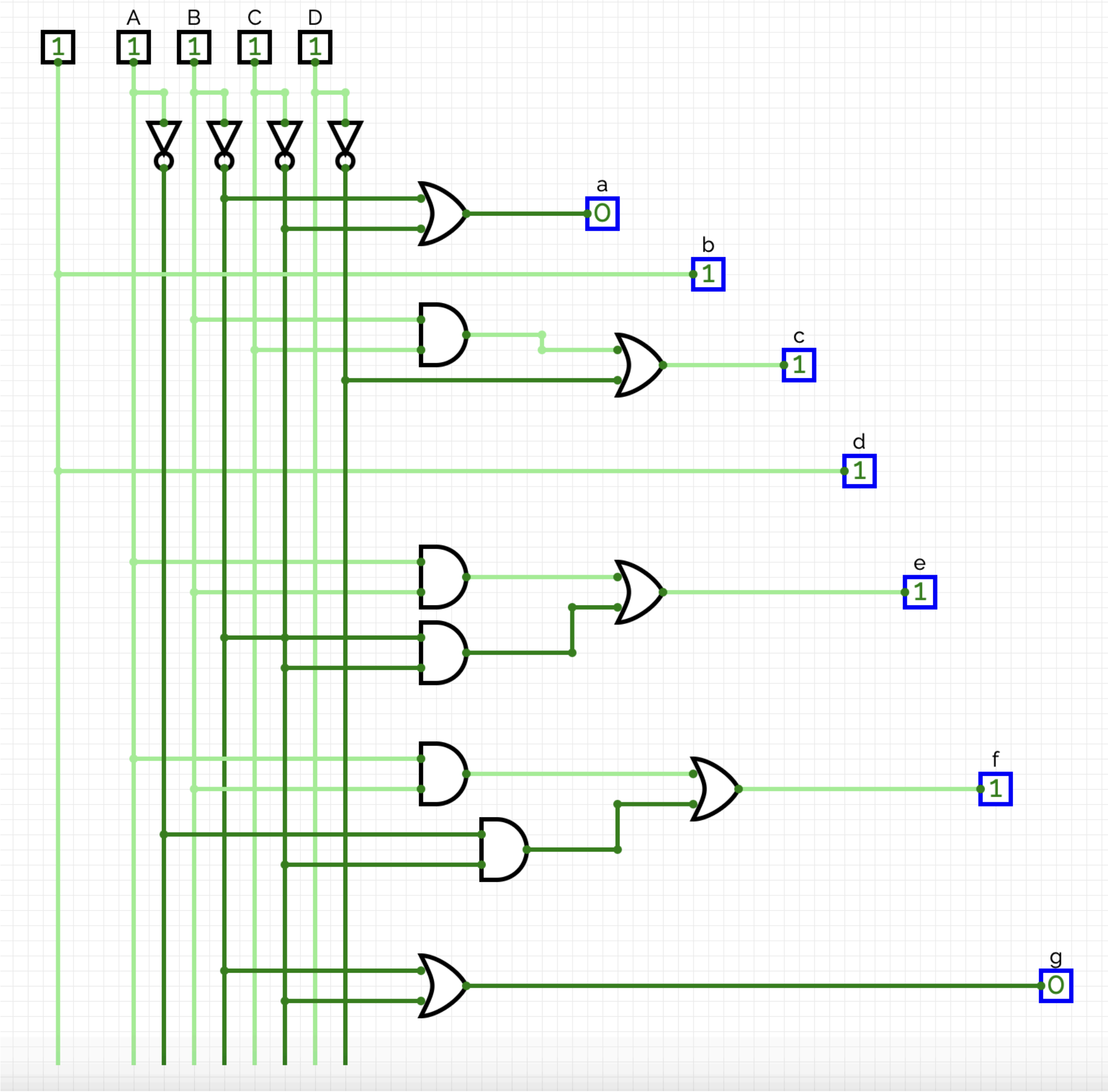


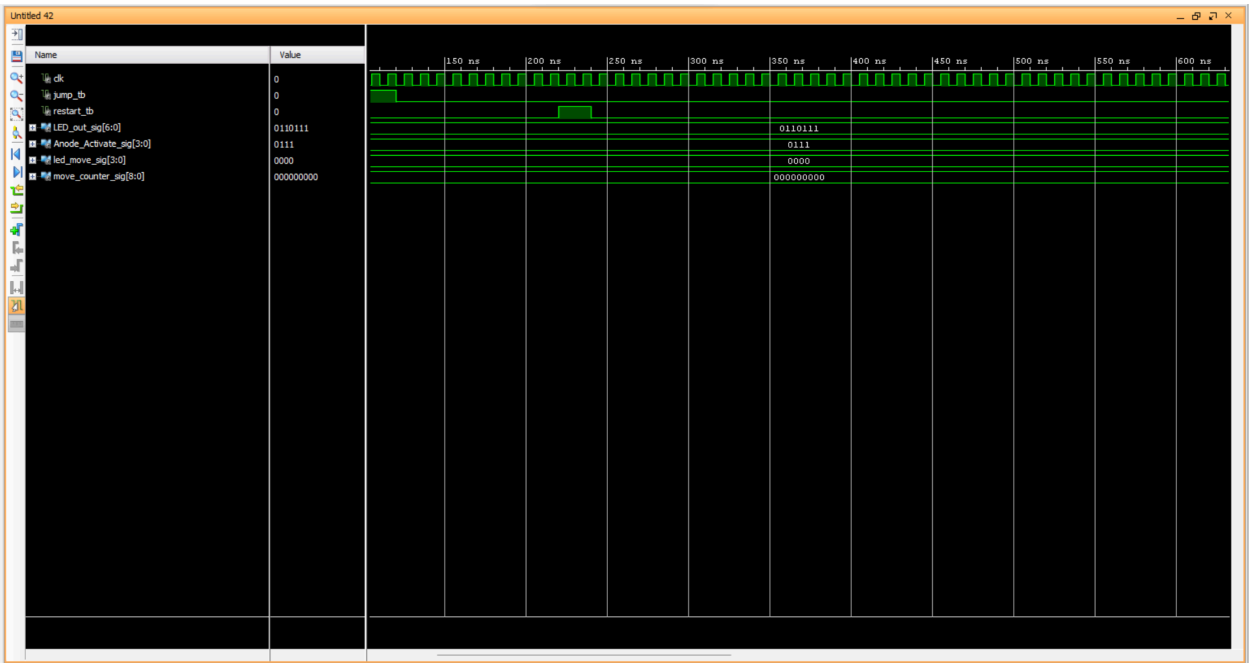
Fig 3: Symbol Level of seven segment case

Bill of Materials

- Digilent Basys Artix-7 Development Board

Test Bench Results

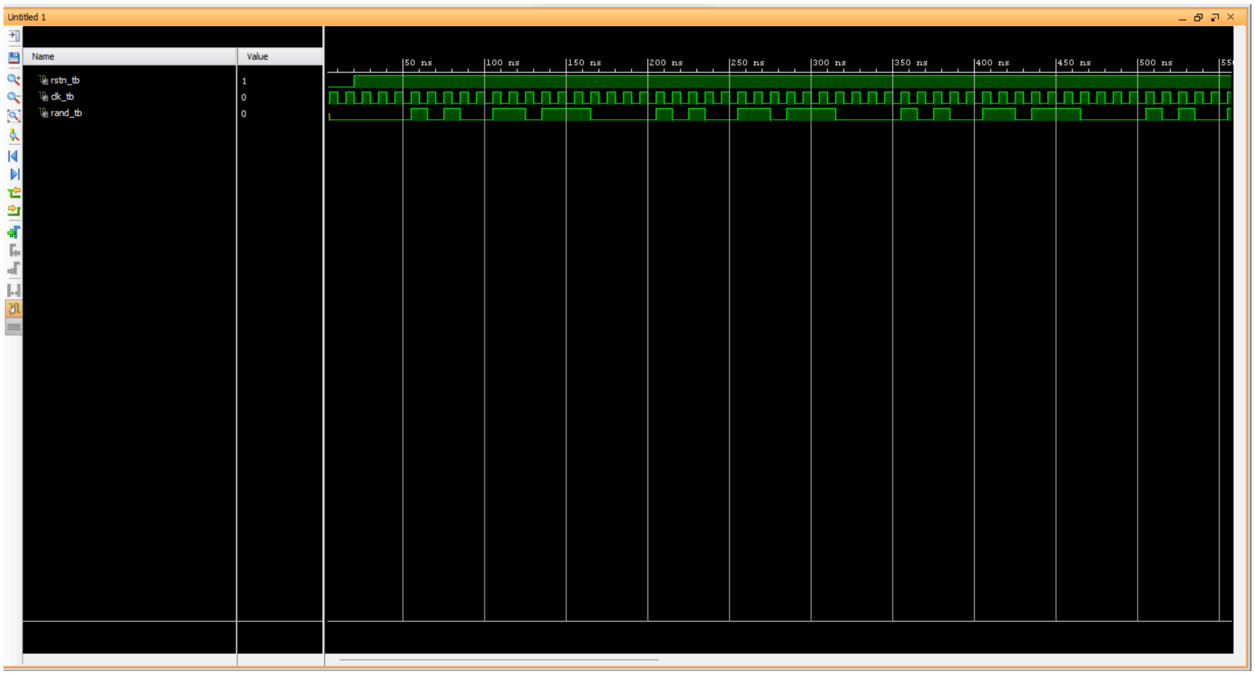
sseg_actives_tb.vhd(overall program)



Name	Value
clk	0
jump_tb	0
restart_tb	0
LED_out_sig[6:0]	0110111
Anode_Activate_sig[3:0]	0111
led_move_sig[3:0]	0000
move_counter_sig[8:0]	000000000

Fig 4: Test Bench Results of sseg_actives_tb.vhd

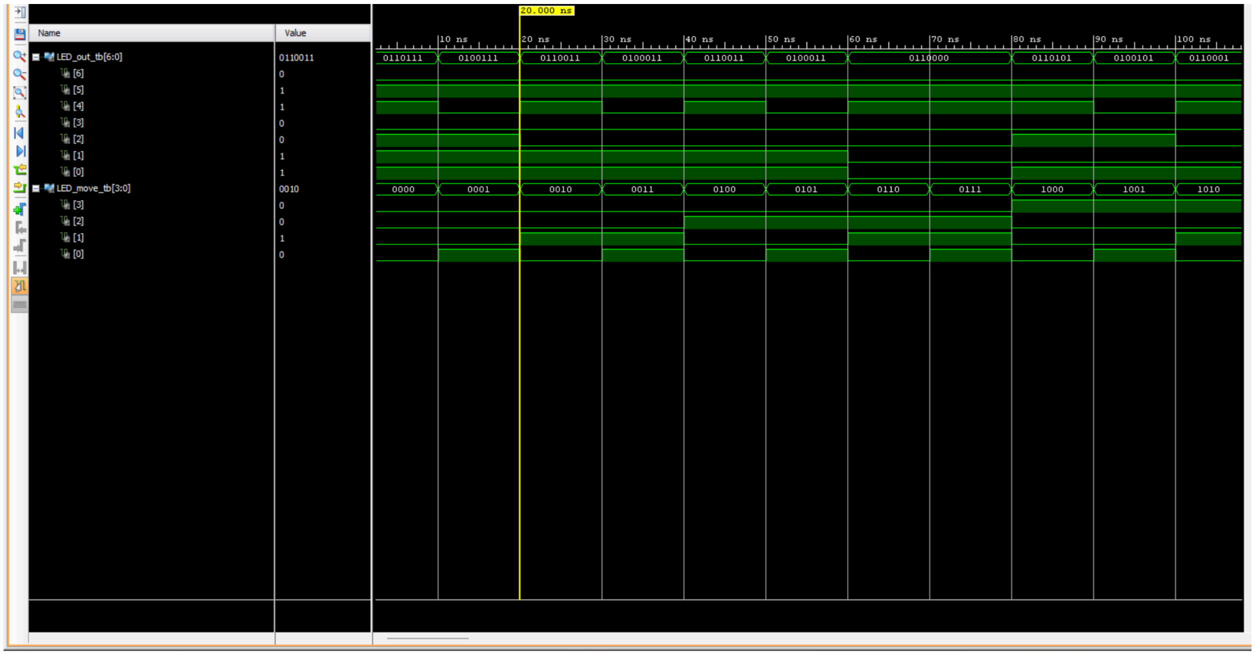
rand_gend.vhd



Name	Value
rstn_tb	1
clk_tb	0
rand_tb	0

Fig 5: Test Bench Results of rand_gend.vhd

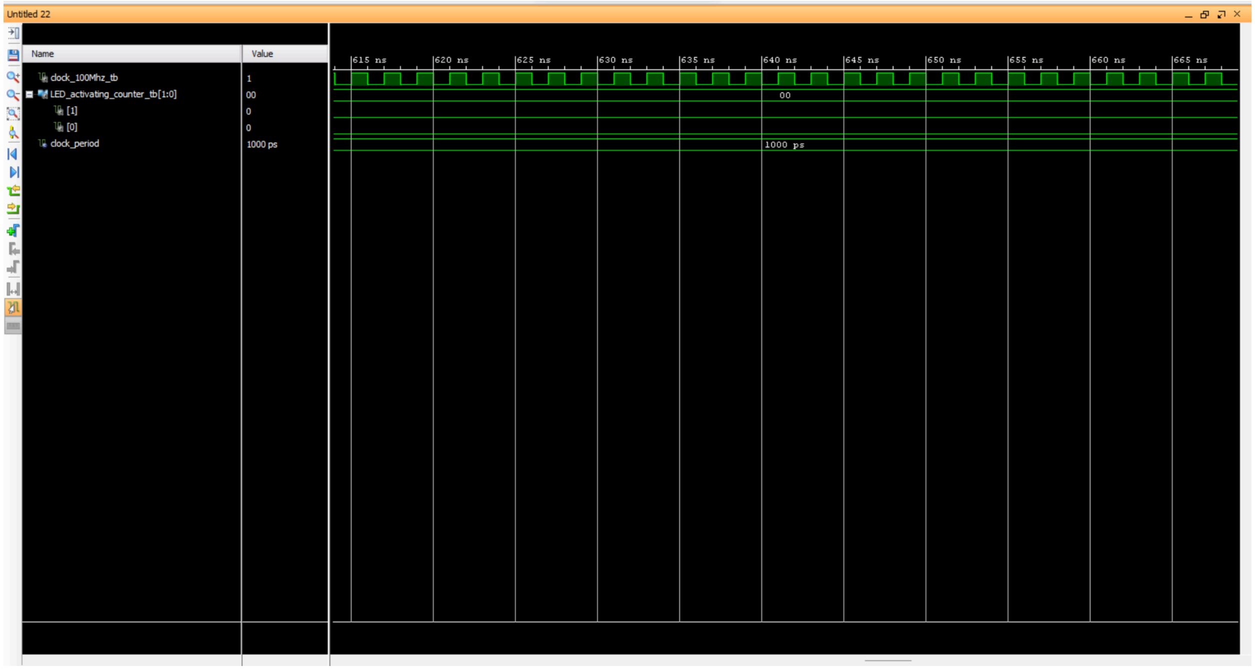
sseg_case_tb.vhd



Name	Value
LED_out_tb[6:0]	0110011
[6]	0
[5]	1
[4]	1
[3]	0
[2]	0
[1]	1
[0]	1
LED_move_tb[3:0]	0010
[3]	0
[2]	0
[1]	1
[0]	0

Fig 6: Test Bench Results of sseg_case_tb.vhd

sseg_timing_tb.vhd

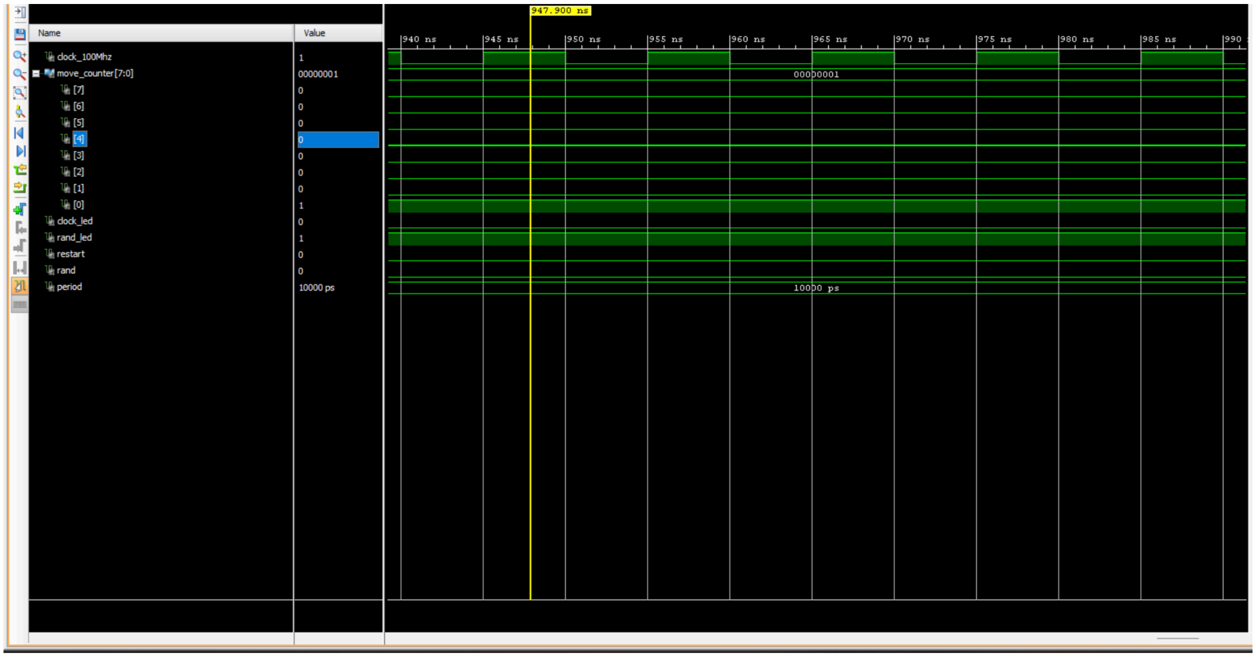


Untitled 22

Name	Value
clock_100Mhz_tb	1
LED_activating_counter_tb[1:0]	00
[1]	0
[0]	0
clock_period	1000 ps

Fig 7: Test Bench Results of sseg_timing_tb.vhd

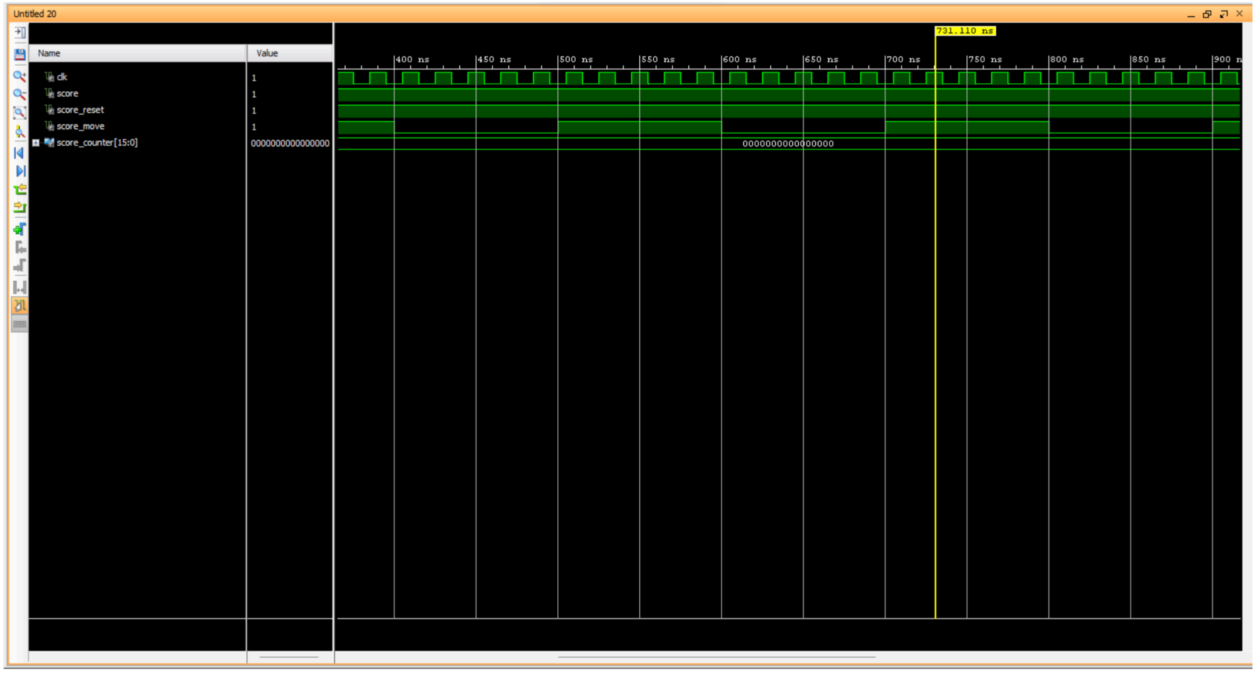
shifter_tb.vhd



Name	Value
clock_100Mhz	1
move_counter[7:0]	00000001
[7]	0
[6]	0
[5]	0
[4]	0
[3]	0
[2]	0
[1]	0
[0]	1
clock_led	0
rand_led	1
restart	0
rand	0
period	10000 ps

Fig 8: Test Bench Results of shifter_tb.vhd

score_system_tb.vhd

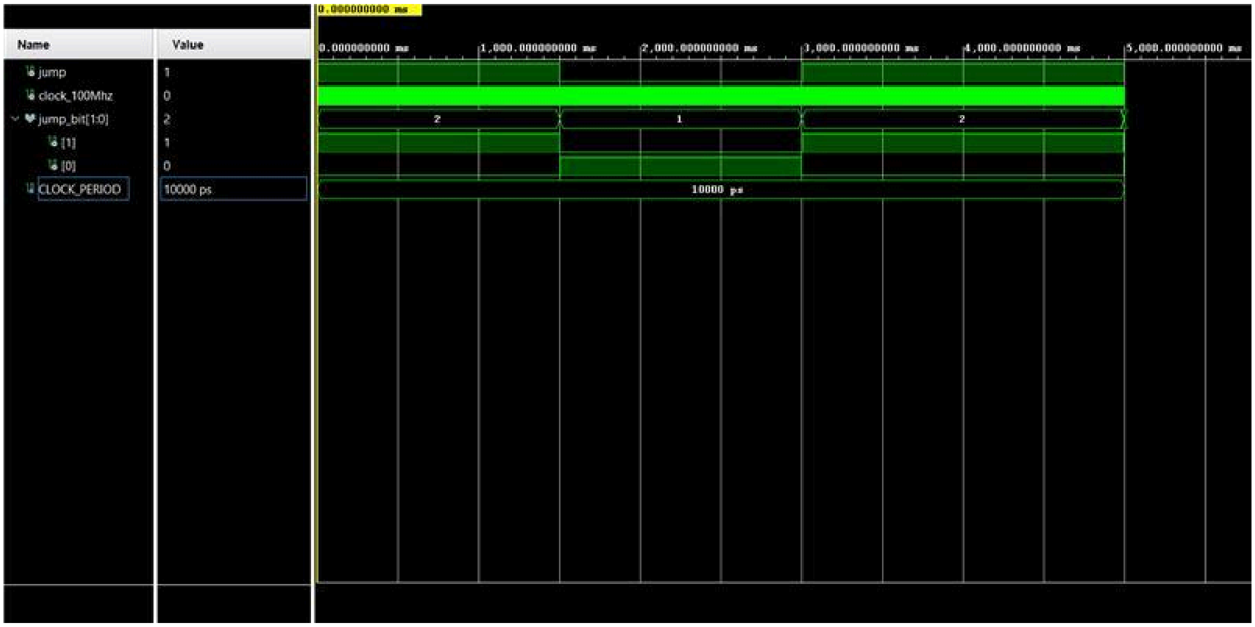


Untitled 20

Name	Value
clk	1
score	1
score_reset	1
score_move	1
score_counter[15:0]	0000000000000000

Fig 9: Test Bench Results of score_system_tb.vhd

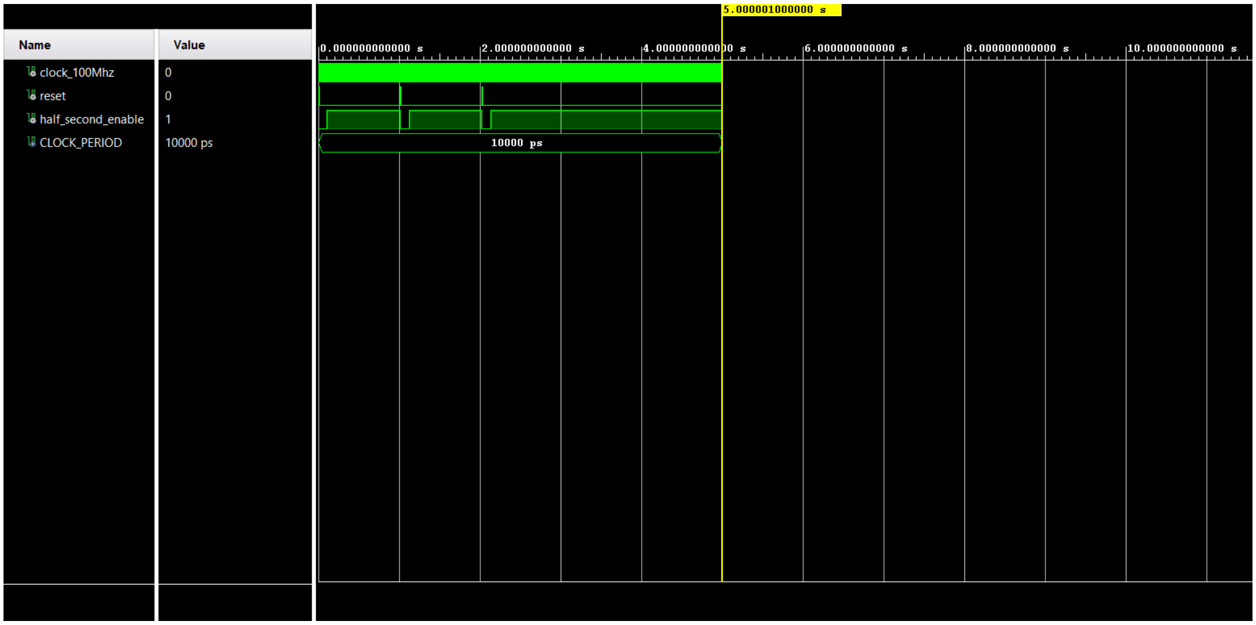
player_in_tb.vhd



Name	Value
jump	1
clock_100Mhz	0
jump_bit[1:0]	2
[1]	1
[0]	0
CLOCK_PERIOD	10000 ps

Fig 10: Test Bench Results of player_in_tb.vhd

Clock_divider_tb.vhd



Name	Value
clock_100Mhz	0
reset	0
half_second_enable	1
CLOCK_PERIOD	10000 ps

Fig 11: Test Bench Results of clock_divider_tb.vhd

Experiment results on Basys3 board

(1) Generate the obstacle randomly and moving to the player:

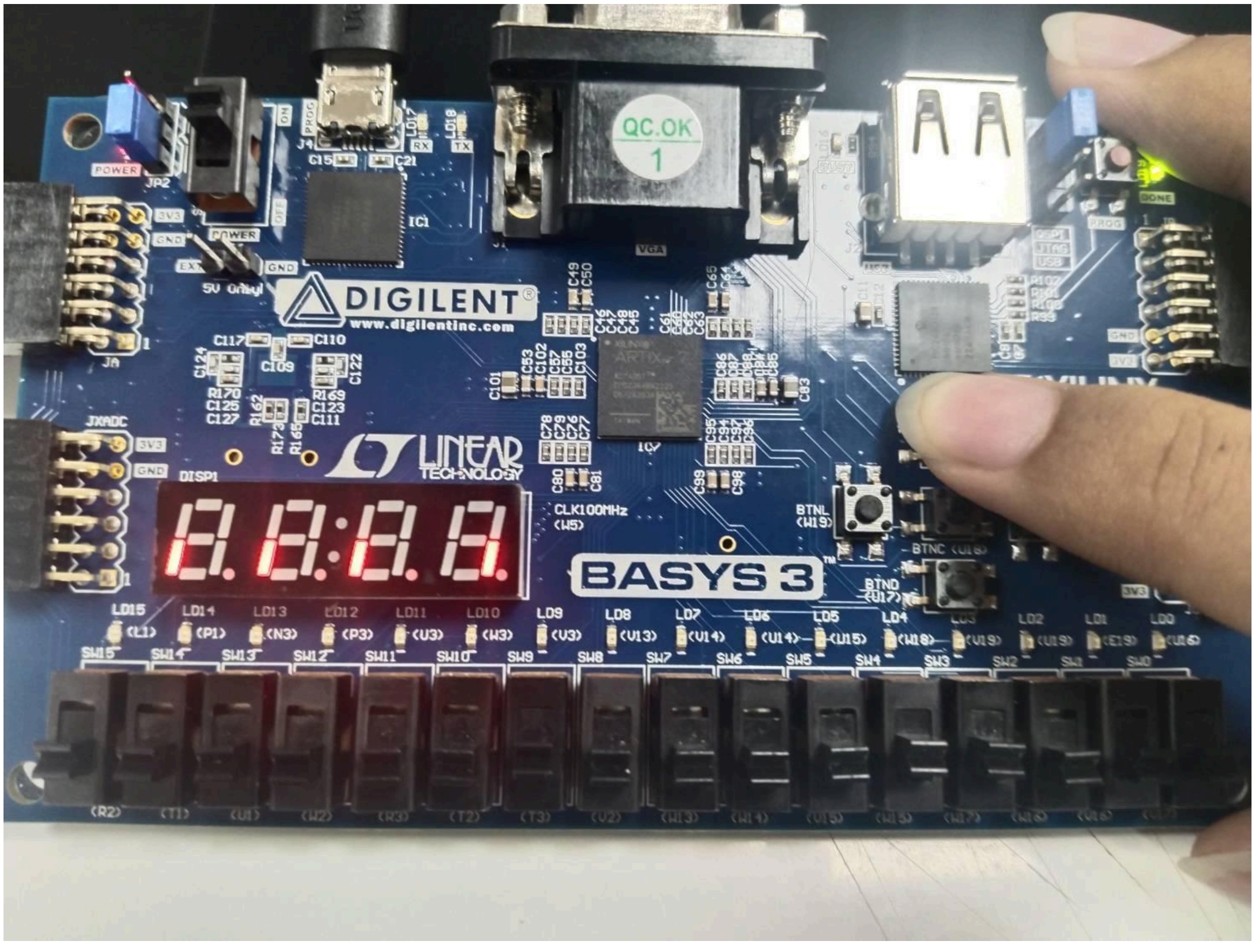


Fig 12: Show about the obstacle is working

(2) The player need to jump not to collide the obstacle to get the scores:

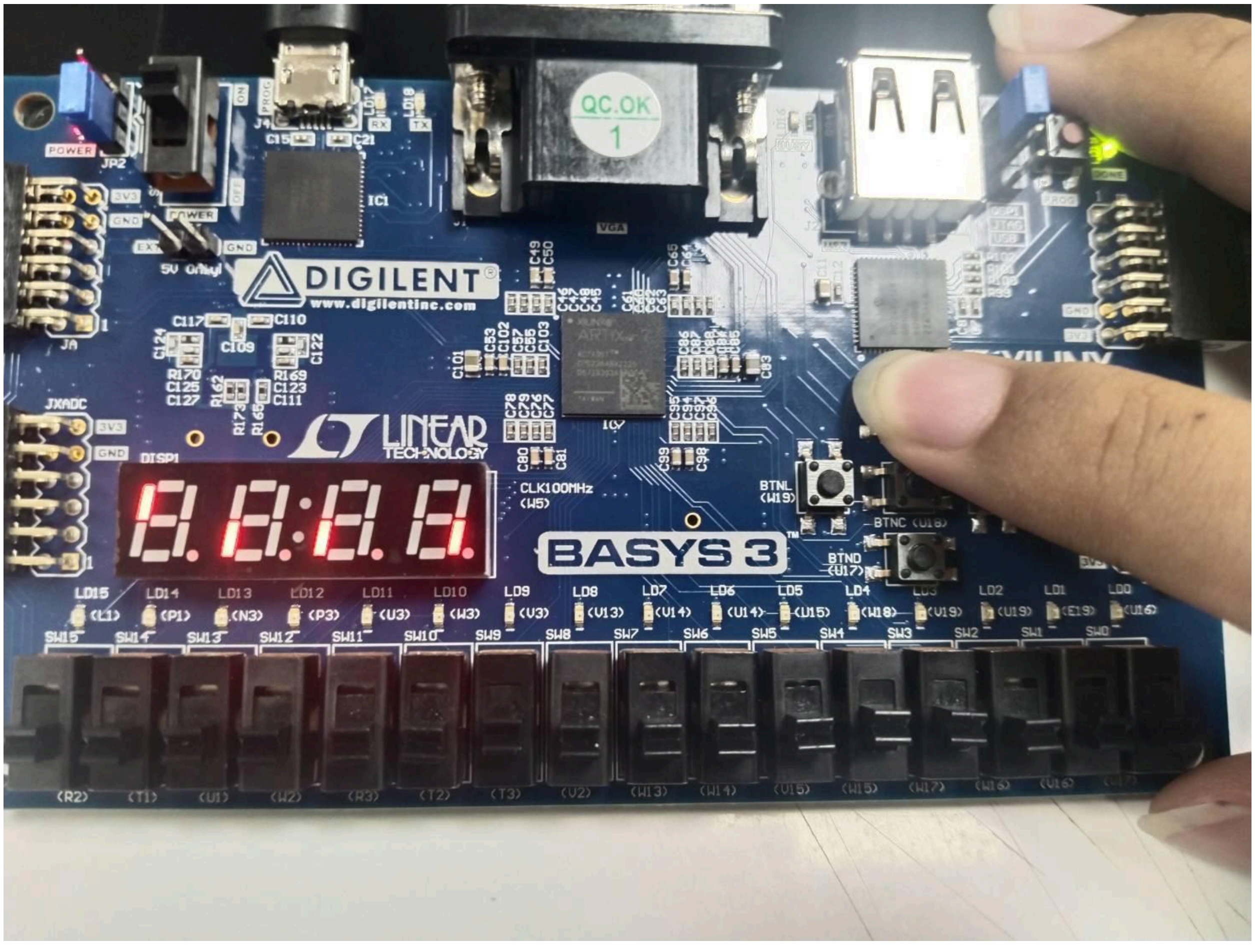


Fig 13. Show how to play our game

(3) Showing the scores how many points the player gets by using leds:

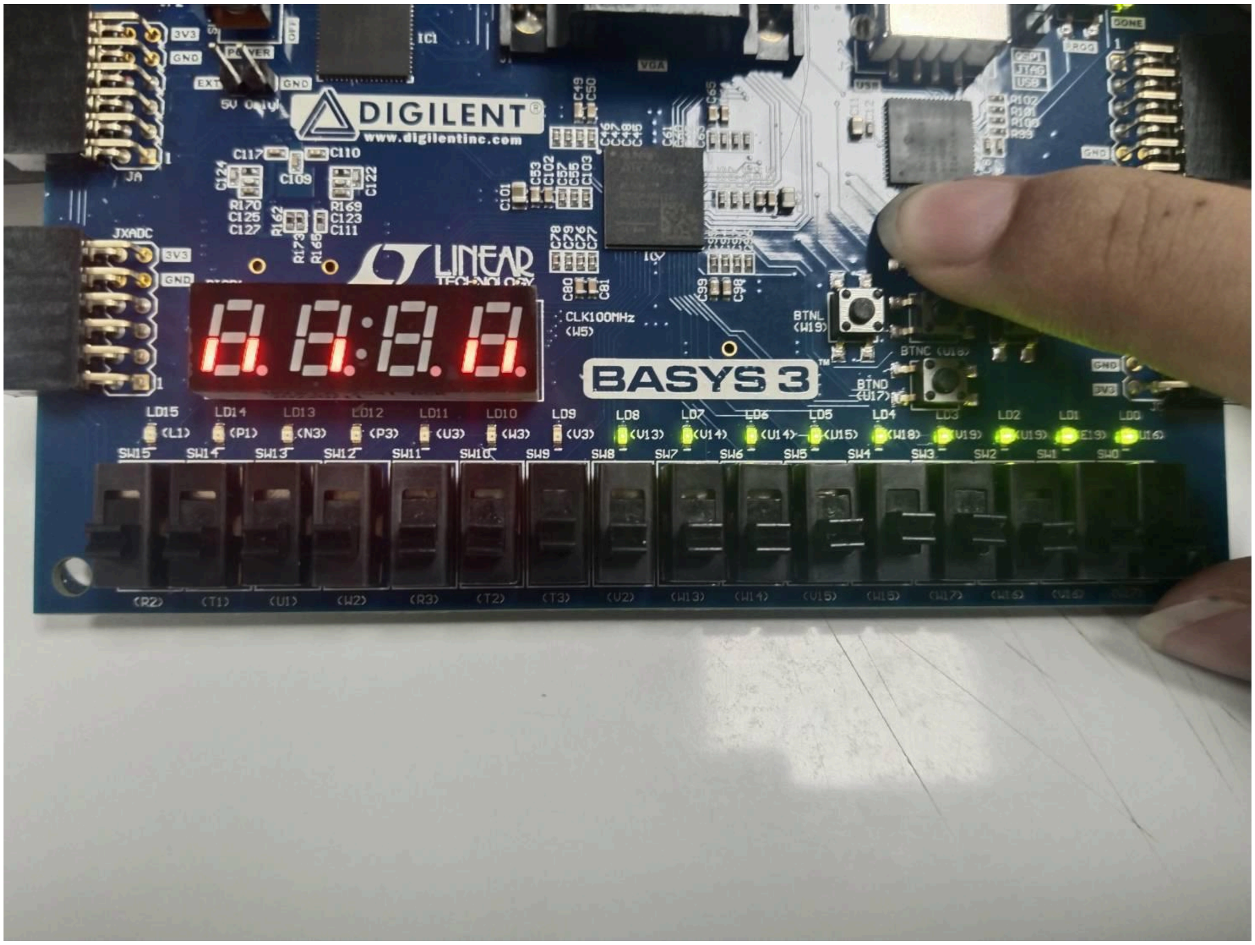


Fig 14: Show the scores

(4) When the player collides with the obstacle, the first segment shows ‘F’ which represents ‘Game Over’:

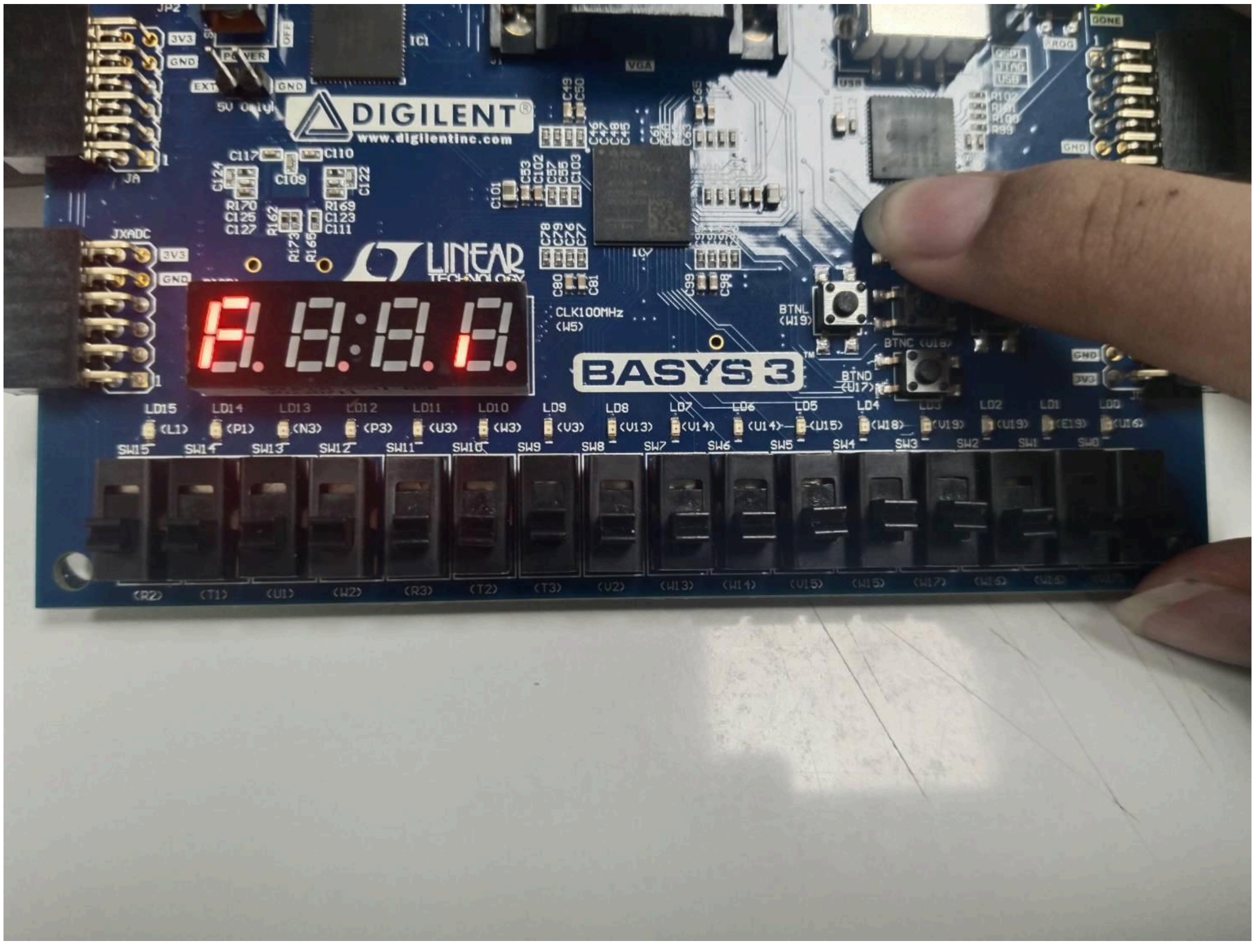


Fig 15: ‘Game Over’ condition

Project Pictures

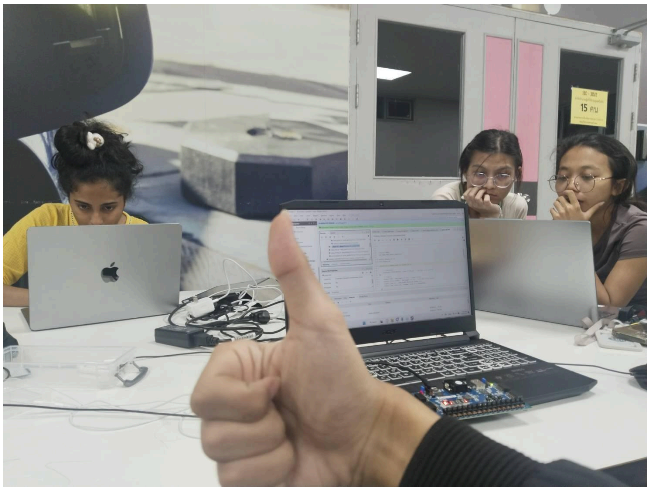


Fig 16: Project Pictures_1

Fig 17: Project Pictures_2

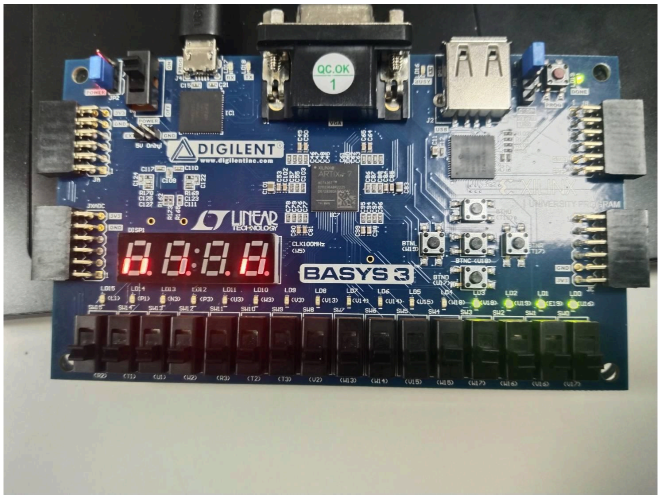


Fig 18: Project Pictures_3