Two-player Tic-Tac-Toe game using DE-1 SoC Board

Digital Design using FPGA Term Project report
Submitted in Partial Fulfillment of the Requirements for the Degree of
MASTER OF TECHNOLOGY
in
SIGNAL PROCESSING AND MACHINE LEARNING
by

Madhu A Ningareddy (222SP013)
Mansi(222SP014)
Vasuprada G(222SP032)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA
SURATHKAL, MANGALORE - 575025
December 2022

CERTIFICATE

This is to certify that the project report entitled "Two-player Tic-Tac-Toe game using DE-1 SoC Board " submitted by Madhu A Ningareddy(222SP013),Mansi (222SP013) and Vasuprada G(222SP032) as a record of the Digital Design using FPGA course work, presented by them is accepted as the Project Report Submission in partial fulfillment of the requirements for the award of Master of Technology (Signal Processing and Machine Learning) in the Department of Electronics and Communication Engineering , National Institute of Technology Karnataka, Surathkal.

Course Coordinator
(Dr. Sumam David)

ABSTRACT

This report aims to describe the implementation of Tic-Tac-Toe game on a DE-1 SoC board using external peripherals. A two player Tic-Tac-Toe game is realized using a finite state machine which controls the player turns and win logic. A Digilent Pmod Keypad is used as input peripheral that selects the location desired to be played by the players. The monitor is used as an output peripheral device, controlled using the VGA port, to display the board, stages in the game and the result.

Table of Contents

Table Of Figures

1. INTRODUCTION

In the game of tic tac toe, two players take turns placing Xs and Os in the compartments of a figure made by two horizontal lines crossing one vertical line. The goal is to be the first to get a row of three Xs or three Os before the other player. If a player selects the initial move from the following list each time it is their turn to play, they will be able to play a flawless game of tic-tac-toe (to win or at least draw),

1. Win: If the player has two in a row, they can place a third to get three in a row.
2. Block: If the opponent has two in a row, the player must play the third themselves to block the opponent.
3. Fork: Cause a scenario where the player has two ways to win (two non-blocked lines of 2).
4. Blocking an opponent's fork: If there is only one possible fork for the opponent, the player should block it. Otherwise, the player should block all forks in any way that simultaneously allows them to make two in a row. Otherwise, the player should make a two in a row to force the opponent into defending, as long as it does not result in them producing a fork. For example, if "X" has two opposite corners and "O" has the center, "O" must not play a corner move to win. (Playing a corner move in this scenario produces a fork for "X" to win.)
5. Center: A player marks the center. (If it is the first move of the game, playing a corner move gives the second player more opportunities to make a mistake and may therefore be the better choice
6. Opposite corner: If the opponent is in the corner, the player plays the opposite corner.
7. Empty corner: The player plays in a corner square.
8. Empty side: The player plays in a middle square on any of the four sides.

2. METHODOLOGY

The tic tac toe game is implemented on DE1 SOC Board with outputs being displayed on Monitor and inputs taken from Pmod Keypad

The Figure 1 presents the communication between the FPGA Board and the external peripherals.The keypad is connected to the GPIO pins on the board and VGA port present in the board establishes the communication between monitor and board.

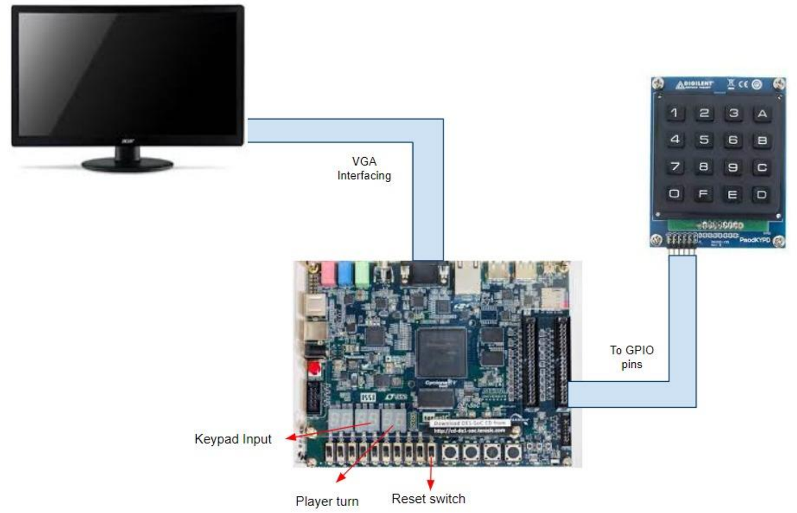


Figure 1 - Peripheral Interface

2.1. COMPONENTS DESCRIPTION

DE1 SOC board

The DE1-SoC Development Kit presents a robust hardware design platform built around the Altera System-on-Chip (SoC) FPGA, which combines the latest dual-core Cortex-A9 embedded cores with industry-leading programmable logic for ultimate design flexibility.

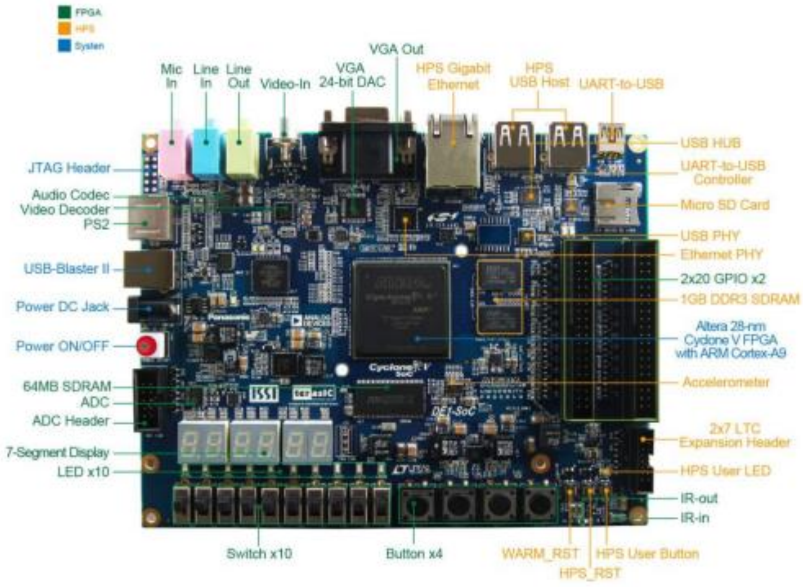


Figure 2 - DE1-SoC development board

CLOCK CIRCUITRY

The four 50MHz clock signals connected to the FPGA are used as clock sources for user logic. One 25MHz clock signal is connected to two HPS clock inputs, and the other one is connected to the clock input of Gigabit Ethernet Transceiver. We have used the on board available 50 MHz clock through pin number AF14.

2.2. PERIPHERALS USED

1. Switches

   There are ten slide switches connected to the FPGA. These switches are not debounced and to be used as level-sensitive data inputs to a circuit. Each switch is connected directly and individually to the FPGA. When the switch is set to the DOWN position (towards the edge of the board), it generates a low logic level to the FPGA. When the switch is set to the UP position, a high logic level is generated to the FPGA.

2. 7-segment Displays

   The DE1-SoC board has six 7-segment displays. These displays are paired to display numbers in various sizes. Figure 3 shows the connection of seven segments (common anode) to pins on Cyclone V SoC FPGA. The segment can be turned on or off by applying a low logic level or high logic level from the FPGA, respectively. Each segment in a display is indexed from 0 to 6, with corresponding positions given in the Figure.

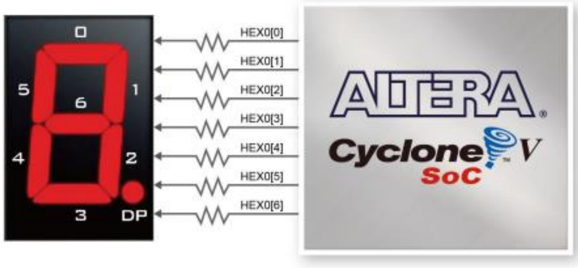


Figure 3 - 7-segment display

3. 2x20 GPIO Expansion Headers
   The board has two 40-pin expansion headers. Each header has 36 user pins connected directly to the Cyclone V SoC FPGA. It also comes with DC +5V (VCC5), DC +3.3V (VCC3P3), and two GND pins. Each pin on the expansion headers is connected to two diodes and a resistor for protection against high or low voltage level. Figure 4 shows the protection circuitry applied to all 2x36 data pins.

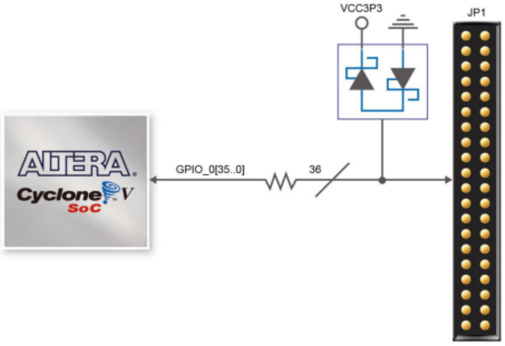


Figure 4 - GPIO Expansion Headers

4. VGA
   The DE1-SoC board has a 15-pin D-SUB connector populated for VGA output. The VGA synchronization signals are generated directly from the Cyclone V SoC FPGA, and the Analog Devices ADV7123 triple 10-bit high-speed video DAC (only the higher 8-bits are used) transforms signals from digital to analog to represent three fundamental colors (red, green, and blue). It can support up to SXGA standard (1280*1024) with signals transmitted at 100MHz. Figure 5 shows the signals connected between the FPGA and VGA.

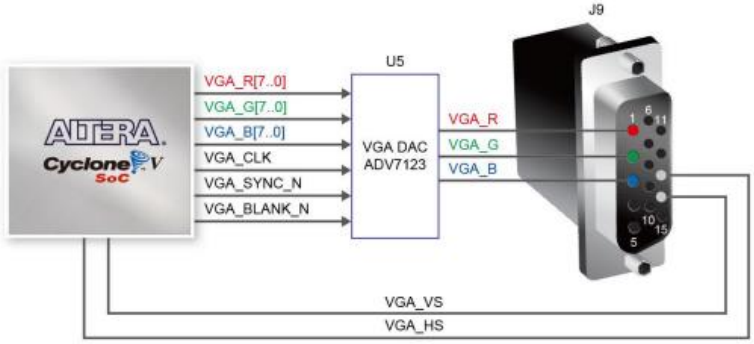


Figure 5 - Connection between FPGA and VGA

Figure 6 illustrates the basic timing requirements for each row (horizontal) displayed on a VGA monitor. An active-low pulse of specific duration is applied to the horizontal synchronization (hsync) input of the monitor, which signifies the end of one row of data and the start of the next. The data (RGB) output to the monitor must be off (driven to 0 V) for a time period called the back porch (b) after the hsync pulse occurs, which is followed by the display interval (c). During the data display interval the RGB data drives each pixel in turn across the row being displayed. Finally, there is a time period called the front porch (d) where the RGB signals must again be off before the next hsync pulse can occur. The timing of vertical synchronization (vsync) is similar to the one shown in Figure 6, except that a vsync pulse signifies the end of one frame and the start of the next, and the data refers to the set of rows in the frame (horizontal timing).

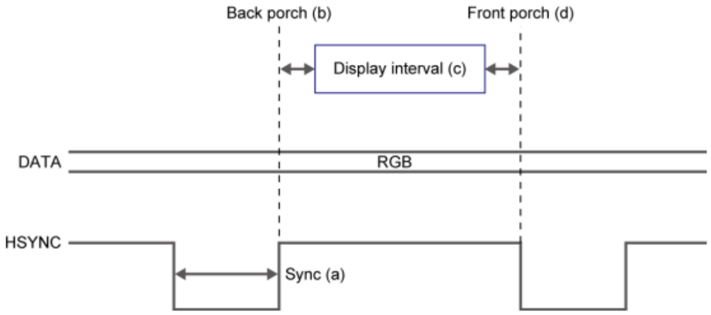


Figure 6 - VGA horizontal timing specification

5. Pmod Keypad

The keypad module consists of 16 momentary pushbutton switches wired in a matrix of 4 columns and 4 rows. Internally, the row lines are pulled up by a large (10K ohm) resistor to logic high. The column lines each have a 470 ohm current limiting resistor to protect from overload. The first column line is wired in parallel to each of the four switches in the first column – 1, 4, 7 and 0. Pressing a button connects the corresponding column and a row. That is, pressing the switch labeled 1 connects column 1 and row 1 together. The row lines are normally pulled high by the 10K resistor connected to the voltage supply. With a column line pulled low, a button press effectively connects the corresponding row line to ground. Since the column line has only a 470 ohm resistor the column can sink more current than the row can source through the 10K resistor. The result is that the row line for the button is pulled to logic low.

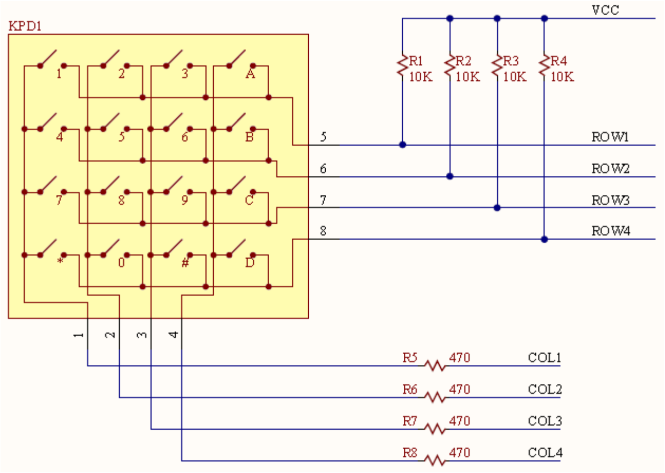
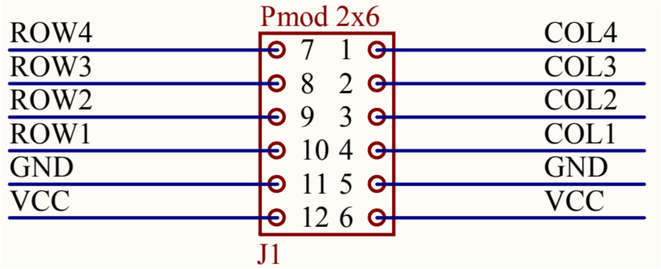


Figure 7 - PMOD keypad internal wiring



2.3. IMPLEMENTATION

The logic of Tic Tac Toe game is implemented using a state machine. The below figure depicts the finite state machine. Player 1 and Player 2 take turn one after the other, when either of them is won, the game is over. The game is reset when the reset switch is set high. Players take their turn to play once the reset is low by giving inputs through pmod keypad.

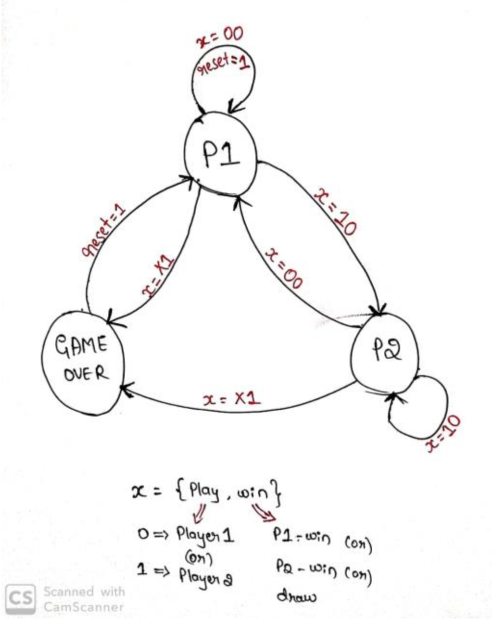
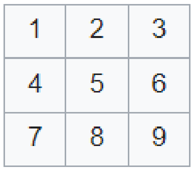


Figure 8 - Tic Tac Toe State Diagram

The Tic Tac Toe board is a 3x3 grid where tiles are numbered from 1 to 9 as shown in Figure 9. When Player 1 takes his turn, the input is displayed as “X” in the Tic Tac Toe grid on the monitor. Similarly for Player 2, “O” is displayed. When Player 1 wins, “X” is displayed at the center of the monitor and “O” is displayed at the center when Player 2 is won. When the game results in a draw the monitor turns white to represent the same.



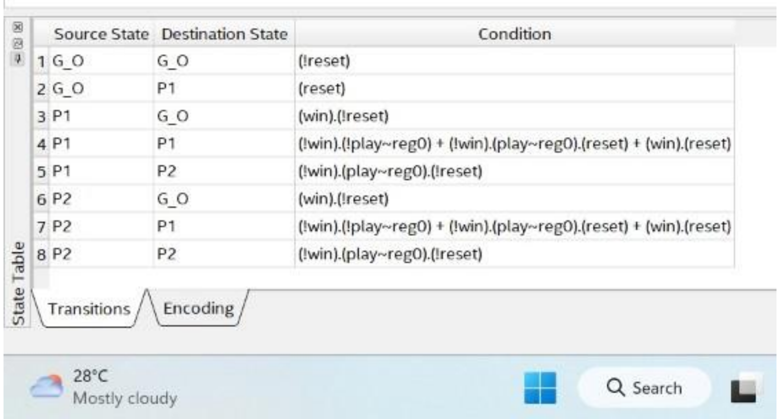
1	2	3
4	5	6
7	8	9

Figure 9 - Board numbering

2.4. SOFTWARE MODULES

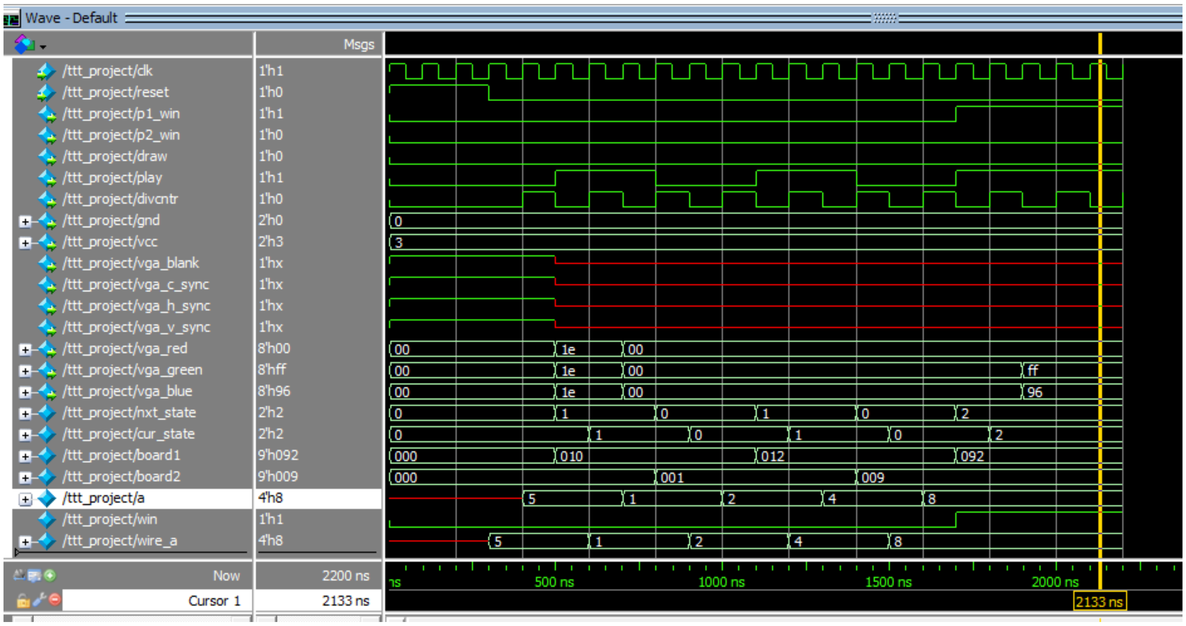
1. Top module – ttt_project:
   This module implements the logic upon which the game works and combines rest of the modules to get inputs. It also synchronizes the function of finite state machine, pmod keypad and VGA.
2. Keypad module – this module establishes effective communication between pmod keypad and FPGA board.
3. VGA interface module – this module consist the logic which interfaces the monitor with FPGA through VGA port.

3. RESULT



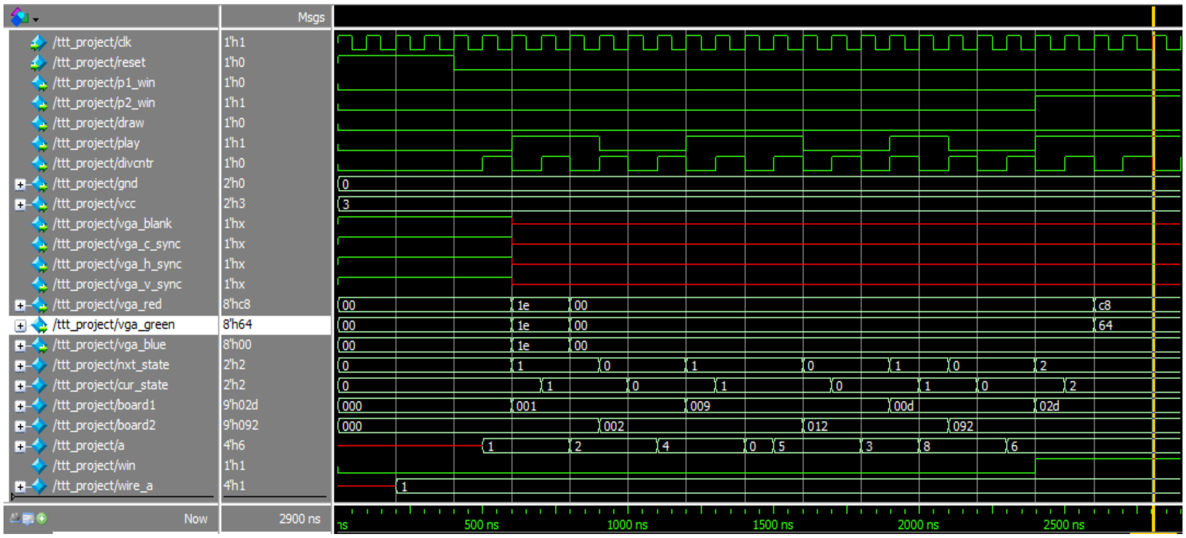
	Source State	Destination State	Condition
1	G_O	G_O	(!reset)
2	G_O	P1	(reset)
3	P1	G_O	(win).(!reset)
4	P1	P1	(!win).(!play~reg0) + (!win).(play~reg0).(reset) + (win).(reset)
5	P1	P2	(!win).(play~reg0).(!reset)
6	P2	G_O	(win).(!reset)
7	P2	P1	(!win).(!play~reg0) + (!win).(play~reg0).(reset) + (win).(reset)
8	P2	P2	(!win).(play~reg0).(!reset)

Figure 10 - Generated State Diagram



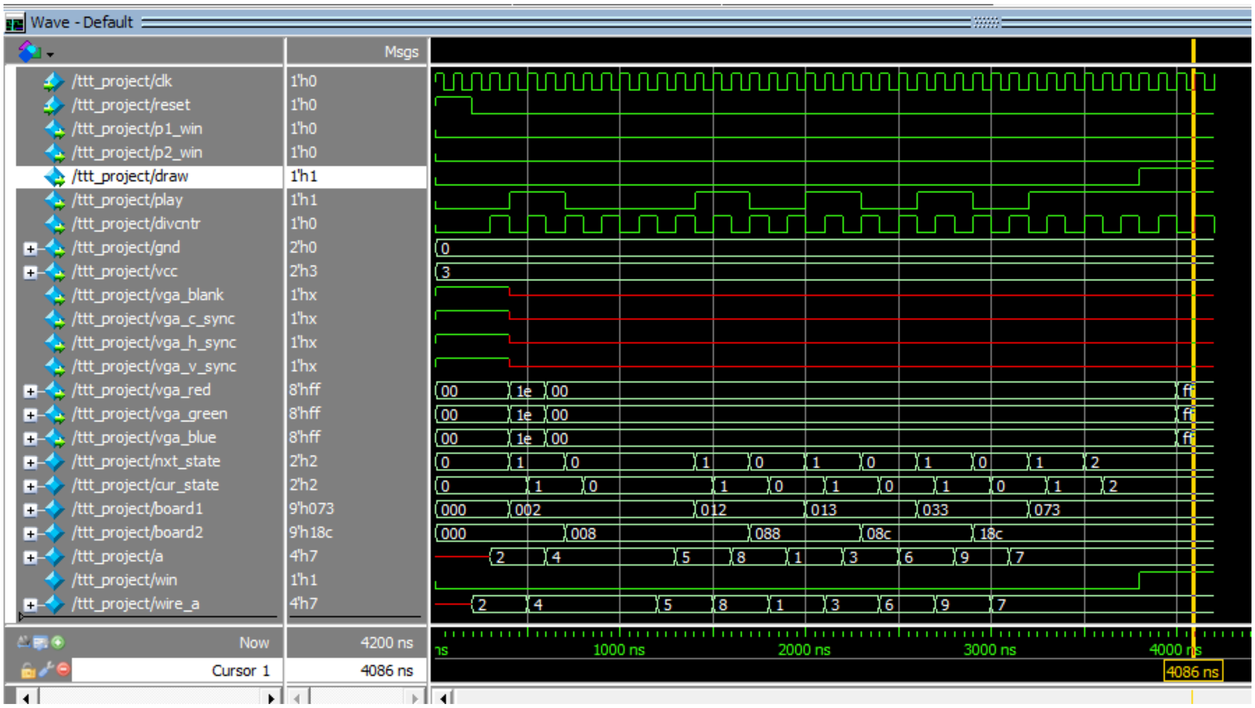
	Msgs
/ttt_project/clk	1'h1
/ttt_project/reset	1'h0
/ttt_project/p1_win	1'h1
/ttt_project/p2_win	1'h0
/ttt_project/draw	1'h0
/ttt_project/play	1'h1
/ttt_project/divcntr	1'h0
/ttt_project/gnd	2'h0
/ttt_project/vcc	2'h3
/ttt_project/vga_blank	1'hx
/ttt_project/vga_c_sync	1'hx
/ttt_project/vga_h_sync	1'hx
/ttt_project/vga_v_sync	1'hx
/ttt_project/vga_red	8'h00
/ttt_project/vga_green	8'hff
/ttt_project/vga_blue	8'h96
/ttt_project/nxt_state	2'h2
/ttt_project/cur_state	2'h2
/ttt_project/board1	9'h092
/ttt_project/board2	9'h009
/ttt_project/a	4'h8
/ttt_project/win	1'h1
/ttt_project/wire_a	4'h8
Now	2200 ns
Cursor 1	2133 ns

Figure 11 - Player1 win simulation



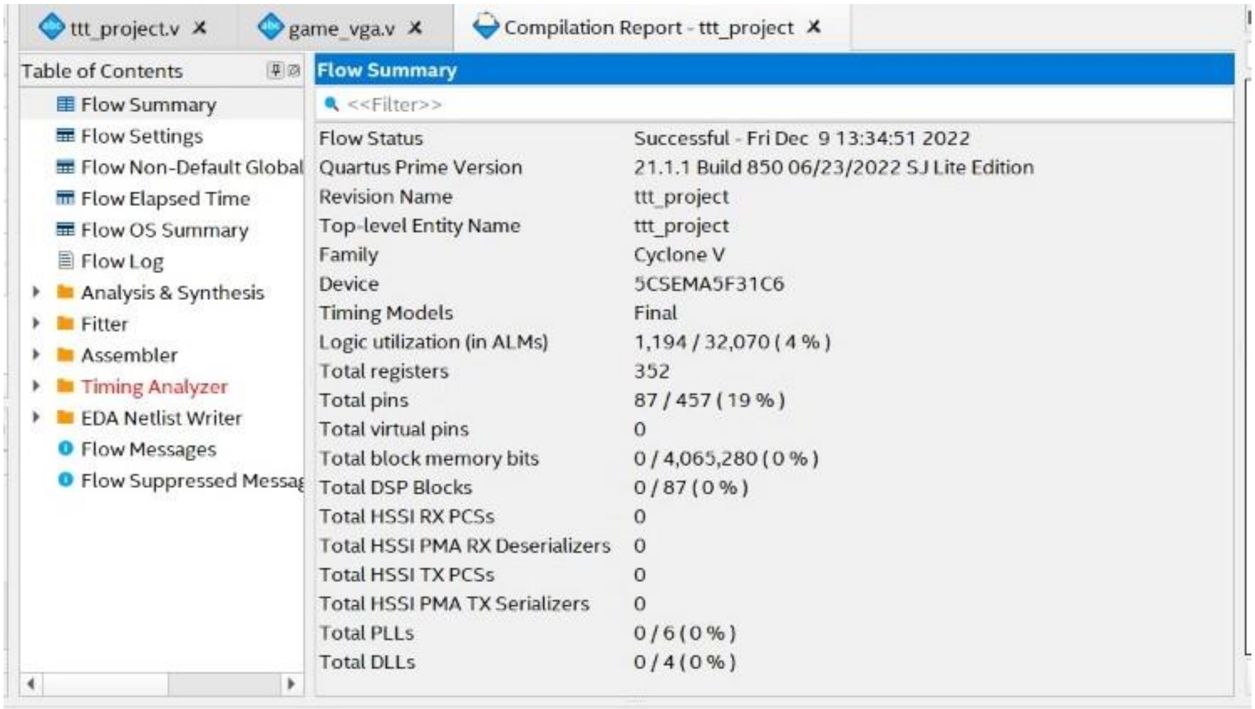
Signal	Msgs
/ttt_project/clk	1'h1
/ttt_project/reset	1'h0
/ttt_project/p1_win	1'h0
/ttt_project/p2_win	1'h1
/ttt_project/draw	1'h0
/ttt_project/play	1'h1
/ttt_project/divcntr	1'h0
/ttt_project/gnd	2'h0
/ttt_project/vcc	2'h3
/ttt_project/vga_blank	1'hx
/ttt_project/vga_c_sync	1'hx
/ttt_project/vga_h_sync	1'hx
/ttt_project/vga_v_sync	1'hx
/ttt_project/vga_red	8'hc8
/ttt_project/vga_green	8'h64
/ttt_project/vga_blue	8'h00
/ttt_project/nxt_state	2'h2
/ttt_project/cur_state	2'h2
/ttt_project/board1	9'h02d
/ttt_project/board2	9'h092
/ttt_project/a	4'h6
/ttt_project/win	1'h1
/ttt_project/wire_a	4'h1

Figure 12 - Player2 win simulation



Signal	Msgs
/ttt_project/clk	1'h0
/ttt_project/reset	1'h0
/ttt_project/p1_win	1'h0
/ttt_project/p2_win	1'h0
/ttt_project/draw	1'h1
/ttt_project/play	1'h1
/ttt_project/divcntr	1'h0
/ttt_project/gnd	2'h0
/ttt_project/vcc	2'h3
/ttt_project/vga_blank	1'hx
/ttt_project/vga_c_sync	1'hx
/ttt_project/vga_h_sync	1'hx
/ttt_project/vga_v_sync	1'hx
/ttt_project/vga_red	8'hff
/ttt_project/vga_green	8'hff
/ttt_project/vga_blue	8'hff
/ttt_project/nxt_state	2'h2
/ttt_project/cur_state	2'h2
/ttt_project/board1	9'h073
/ttt_project/board2	9'h18c
/ttt_project/a	4'h7
/ttt_project/win	1'h1
/ttt_project/wire_a	4'h7

Figure 13 - Draw condition



Flow Summary

Flow Status	Successful - Fri Dec 9 13:34:51 2022
Quartus Prime Version	21.1.1 Build 850 06/23/2022 SJ Lite Edition
Revision Name	ttt_project
Top-level Entity Name	ttt_project
Family	Cyclone V
Device	5CSEMA5F31C6
Timing Models	Final
Logic utilization (in ALMs)	1,194 / 32,070 (4 %)
Total registers	352
Total pins	87 / 457 (19 %)
Total virtual pins	0
Total block memory bits	0 / 4,065,280 (0 %)
Total DSP Blocks	0 / 87 (0 %)
Total HSSI RX PCSs	0
Total HSSI PMA RX Deserializers	0
Total HSSI TX PCSs	0
Total HSSI PMA TX Serializers	0
Total PLLs	0 / 6 (0 %)
Total DLLs	0 / 4 (0 %)

Figure 14 - Device Utilization

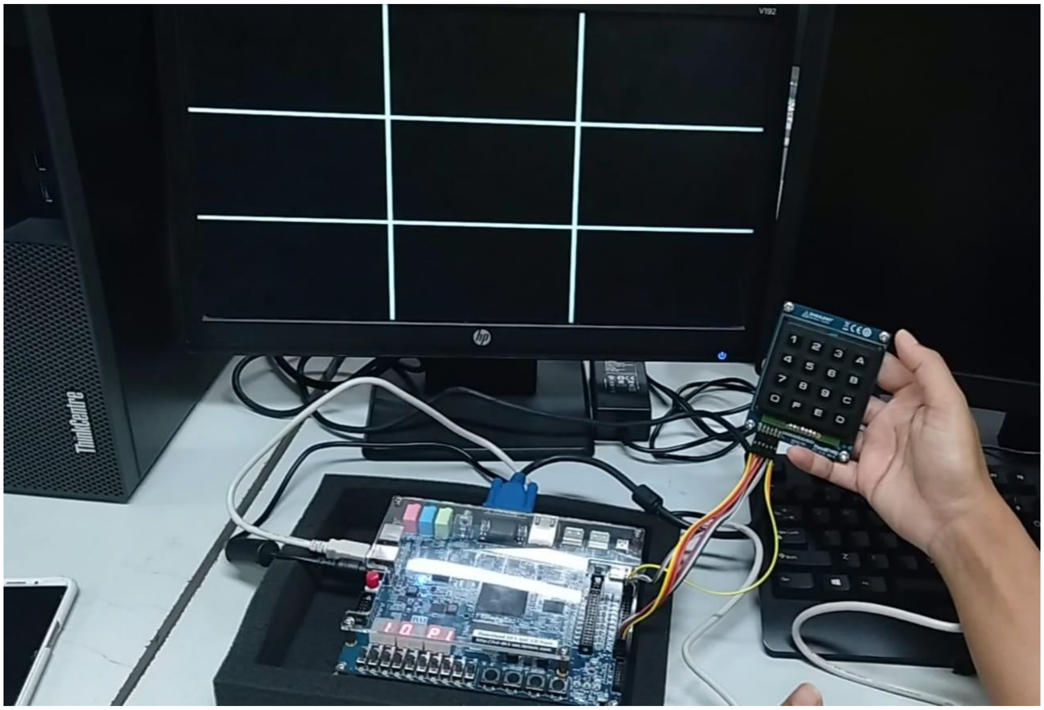


Figure 15-Tic Tac Toe Grid

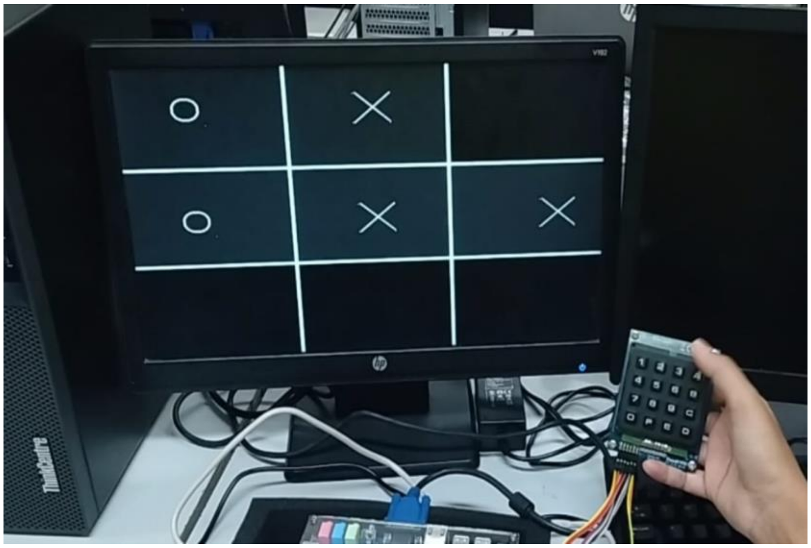


Figure 16-Grid with inputs

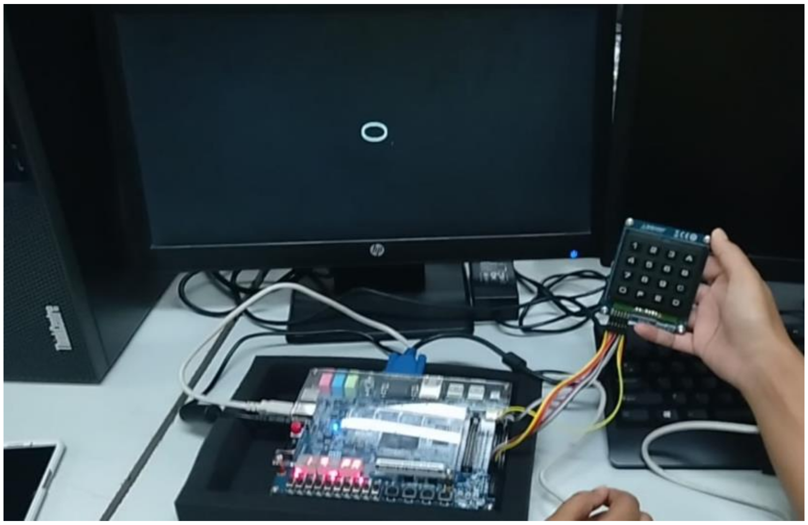


Figure 17 - Player2 "O" wins

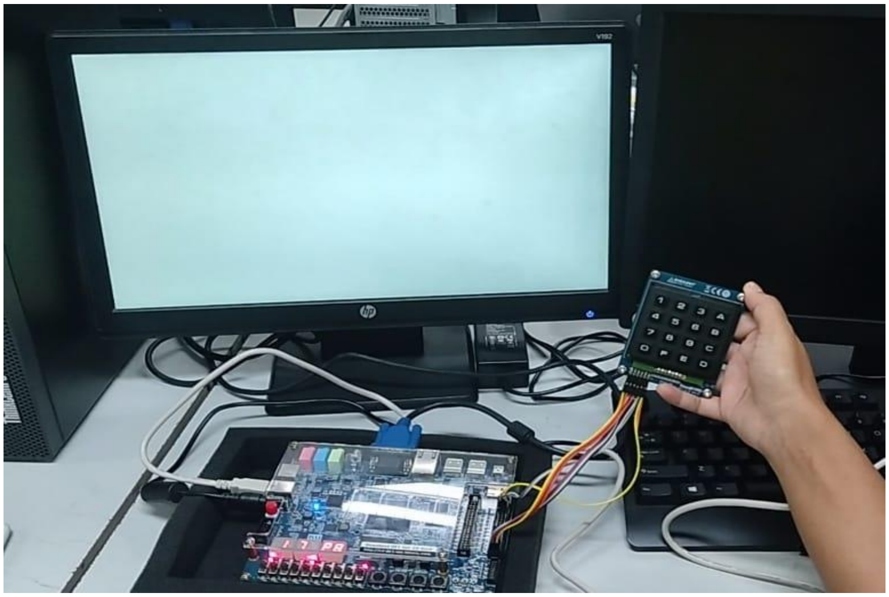


Figure 18-Draw Case

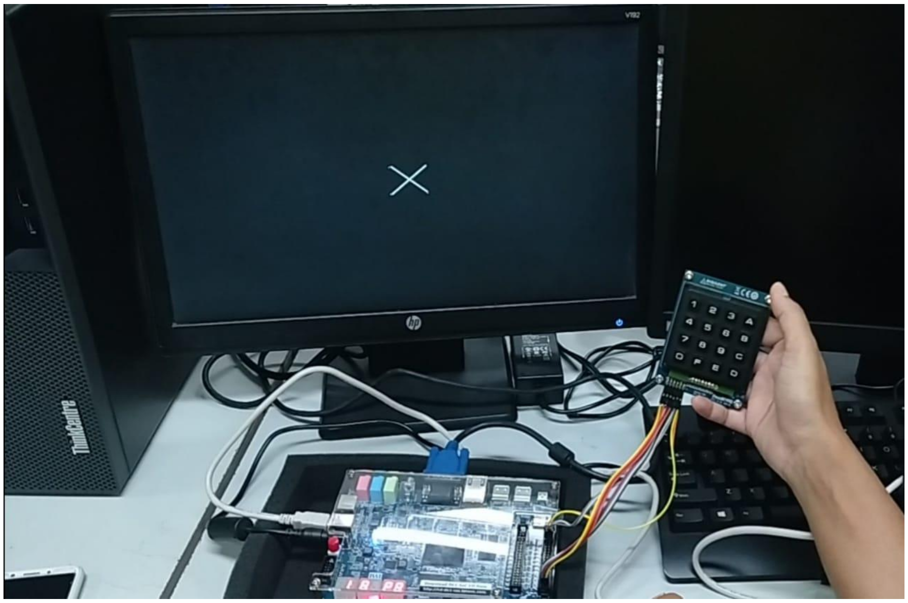


Figure 19-Player1 "X" wins

4. CONCLUSION

This design accomplishes two player Tic-Tac-Toe game on FPGA DE1 SoC board. It utilizes keypad as input peripheral device and LCD monitor as output display to represent Tic-Tac-Toe game. The design results in win of Player1, win of Player2 or draw. A “X” is displayed in the monitor, when Player1 wins and similarly “O” is displayed when Player2 wins. In case of draw, the monitor turns white. Further this design can be extended as a human vs computer game, with different levels of difficulty.

References

[1] T. Technologies, “De-1 soc user manual,” Available at http://www.ee.ic.ac.uk/pcheung/teaching/ee2_digital/de1-soc_user_manual.pdf.

[2] Digilent, “Digilent pmod keypad reference manual,” Available at https://digilent.com/reference/pmod/pmodkypd/reference-manual?redirect=1.

[3] U. of Toronto, “Interfaces on the de1-soc board,” Available at http://www-ug.eecg.toronto.edu/msl/manuals/tutorial_DE1-SoC-v5.4.pdf.