TEIS **Technical report**

Author Jasim Abbasi		Task VHDL Engineering Job		
Email engr.jasim@gmail.com	Approved		Version	File TIES-Game

TIES AB

Tic Tac Toe Game

VHDL Engineering Job B

Jasim Abbasi

CONTENTS

1 Introduction

The tic tac toe is a fun game that is played between two players. In this project Altera DE2 and VGA monitor at 640X480 pixel resolution is used. The standard game of tic tac toe consists of nine empty squares as illustrate in the figure 1 that players may mark with zero **(0)** and cross**(X)** to designate it to be their box. I use **red** and **green** boxes instead of using **0** and **X** to differentiate two players**. Player 1** use red box and **player 2** use green box to mark the specific location of the tic tac board.

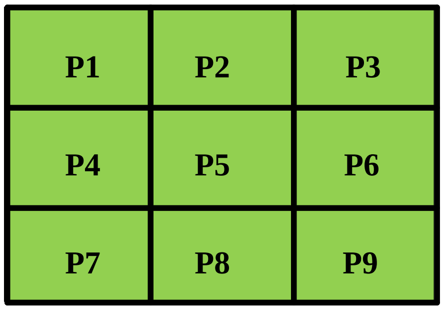


Figure 1: Tic tac toe board

1.1 Specification

In addition to the functionality of the construction the following construction requirements are going to be fulfilled.

Construction requirements:-

1) File header in each file
2) Commenting the code
3) Asynchronous "reset_n" for processes
4) Ending Active low signals name with "_N"
5) Select the same name for the input / output ports of the construction as in the DE2-115 manual
6) Name of the signal should end with _us or _'s to see if it is "unsigned" or "signed" s
7) All the outputs are to be defined for "x", "1" or "0" after "reset_n" goes high,
8) All asynchronous inputs (except the clock) will clocked through two flip-flops to avoid metastability problems,
9) Design Assistant should be run to the design,
10) Following VGA protocol and sharing 50MHz clock with a PLL
11) Test protocol should be written
12) Verifying the test bench and validating the construction on DE2 board with buttons and switches

1.2 Construction code requirements

The client has requirements that code should be written in VHDL and result should be simulate in modelsim and verify it on DE2-115 board and VGA monitor.

1.3 Delivery requirements

The delivery of the project should be in three weeks.

2 Time Table

Week 1

- VGA component from uppgift 8a is adapted in this project
- Tic toe board is drawn on the VGA screen

Week 2

- To distinguish between two player red and green boxes are constructed for both players
- Game interface using switches are constructed

Week 3

- Tic tac logic is implemented
- Validate result on Modelsim
- Validate result on DE2-board

Week 4

- Report writing

3 Test Protocol

Simple test bench have written to verify the working of the game

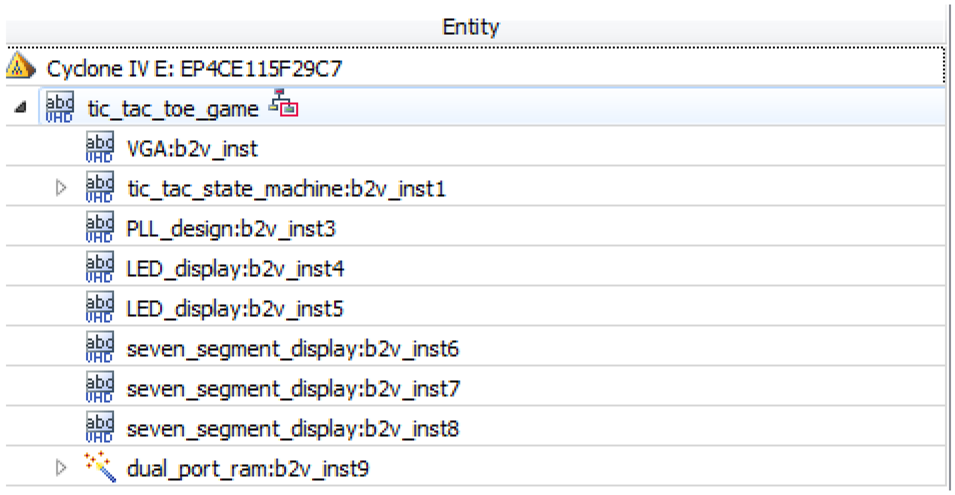
Table 1 : Test Protocol

Case	Description	OK
Test 1	SW=000000001010	LEDR_1=111111, LEDR_2=000000, HEX0=79
Test 2	SW=010010000000	LEDR_1=000000, LEDR_2=111111, HEX0=24
Test 4	Reset_n=0	No operation

4 Design

4.1 System architecture

This chapter contains a brief description of the components and modules to be used in this system. The overall architecture of system is shown in the figure



Entity
Cyclone IV E: EP4CE115F29C7
tic_tac_toe_game
VGA:b2v_inst
tic_tac_state_machine:b2v_inst1
PLL_design:b2v_inst3
LED_display:b2v_inst4
LED_display:b2v_inst5
seven_segment_display:b2v_inst6
seven_segment_display:b2v_inst7
seven_segment_display:b2v_inst8
dual_port_ram:b2v_inst9

Figure 1: Project window with the hierarchy structure

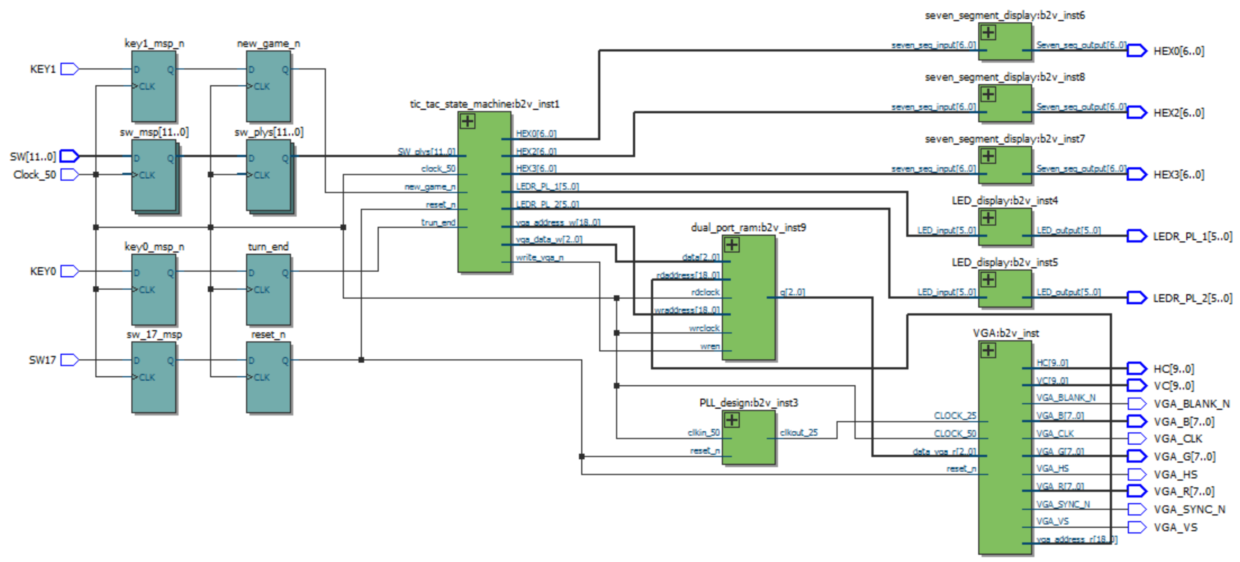


Figure 2: RTL view from the Quartus

The construction consists of following components

1. PLL Component
2. VGA Component

3. Tic tac toe state machine
4. Dual port ram
5. Seven segment display component (three 7_ segment component are used in this project)
6. LED display (two LED_display component are used in this project one for each player)

4.1.1 PLL Component

On the DE2-board a clock source of 50MHZ is available whereas the VGA controller requires 25 MHZ clock frequency.

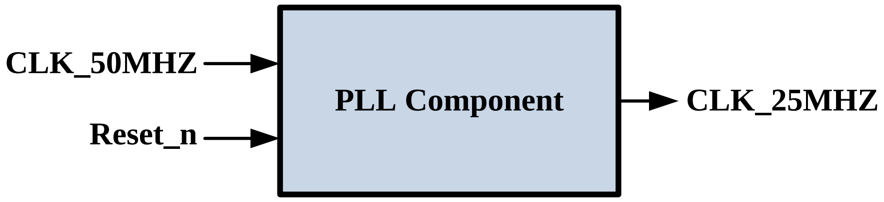
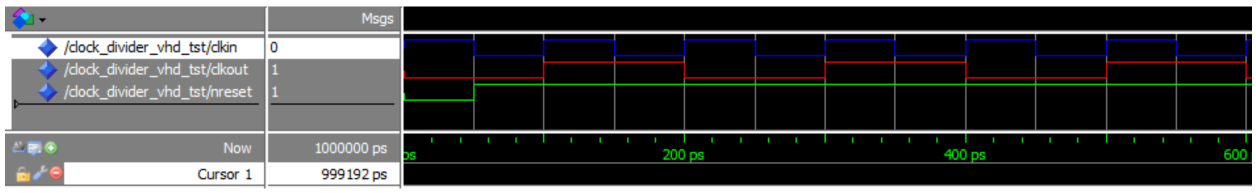


Figure 3: Clock Divider Component



```
                               Msgs
/clock_divider_vhd_tst/clkin     0
/clock_divider_vhd_tst/clkout    1
/clock_divider_vhd_tst/nreset    1

Now              1000000 ps
Cursor 1          999192 ps
```

Figure 4: Clock Divider Result (Clkin=50MHZ(Blue signal) and Clkout=25MHZ(red signal))

4.1.2 VGA Component

The standard VGA format monitor contains 640X480 resolutions of picture element as shown in the figure 3

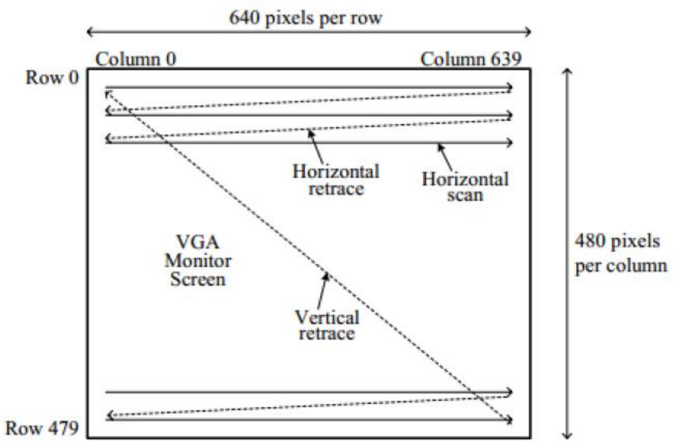


Figure 5: VGA monitor with 640X480 resolutions

The image can be display on the screen by turning **ON** and **OFF** of individual pixels. The monitor continuously scans through the entire screen at a very fast speed. The scan starts from row 0, column 0 and moves to the right until it reaches the last column in the row. When the scan reaches the end of a row, it continues move to the the beginning of the next row. When the scan reaches the last pixel it goes back to the top left corner of the screen, and repeats the scanning process again. All the pixels are truned off during the horizontal and the vertical retraces[1].

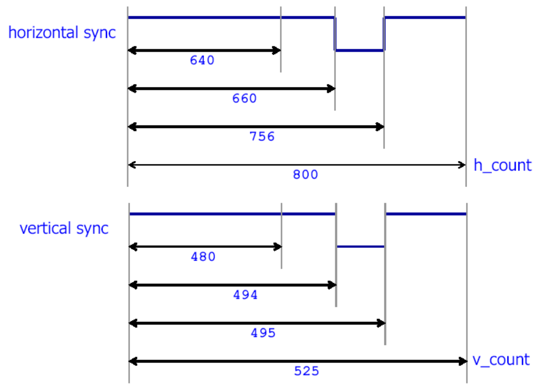


Figure 6: Horizantal Sync and Vertical sync

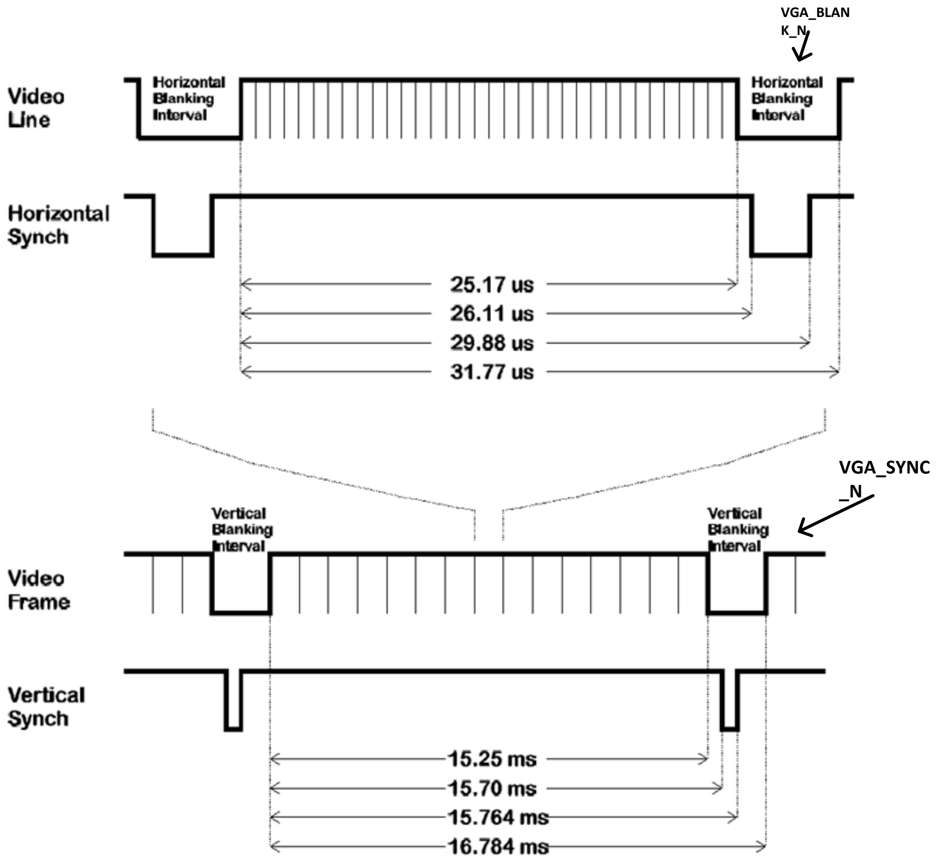


Figure 7: VGA Control Signal Timing (Protocol)

This module will generate the horizontal and vertical synchronization signals by counting the pixel clocks.

- The HSYNC should get low in the beginning of each horizontal row (96 pixel clocks) and high for the rest of time.
- The VSYNC should get low in the beginning of each frame and high for the rest of time.
- HCount is 10 bit horizontal counter that counts from 0 to 799(for one horizontal row) and then reset itself to count for next row.
- VCount is 10 bit counter that counts horizontal lines. It goes from 0 to 599 and reset itself to count the next frame.
- Video_on signal is high when video is written on the VGA display.
- The Graphic User Interface of the game can be control by counting the pixel clocks

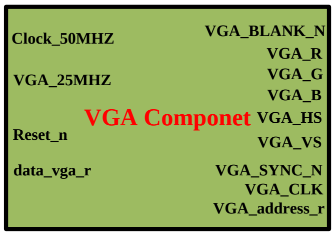
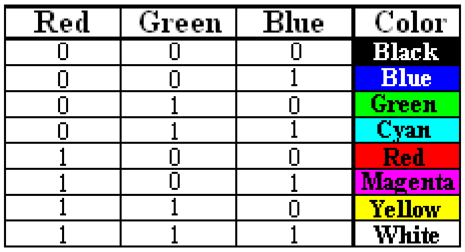


Figure 8: VGA Component

4.1.3 Dual port ram

VGA component include a memory interface connected to a port on the dual port memory as video memory. Dual port memory consists of the pixel memory 921600 (640 x 480 x 3) bits, as well as two separate processes for writing and reading. The dual port ram has two separate clocks for reading and writing. It allows it to print twice as fast as it needs to read. As output there is a three bit RGB color, which results in 8 colors.

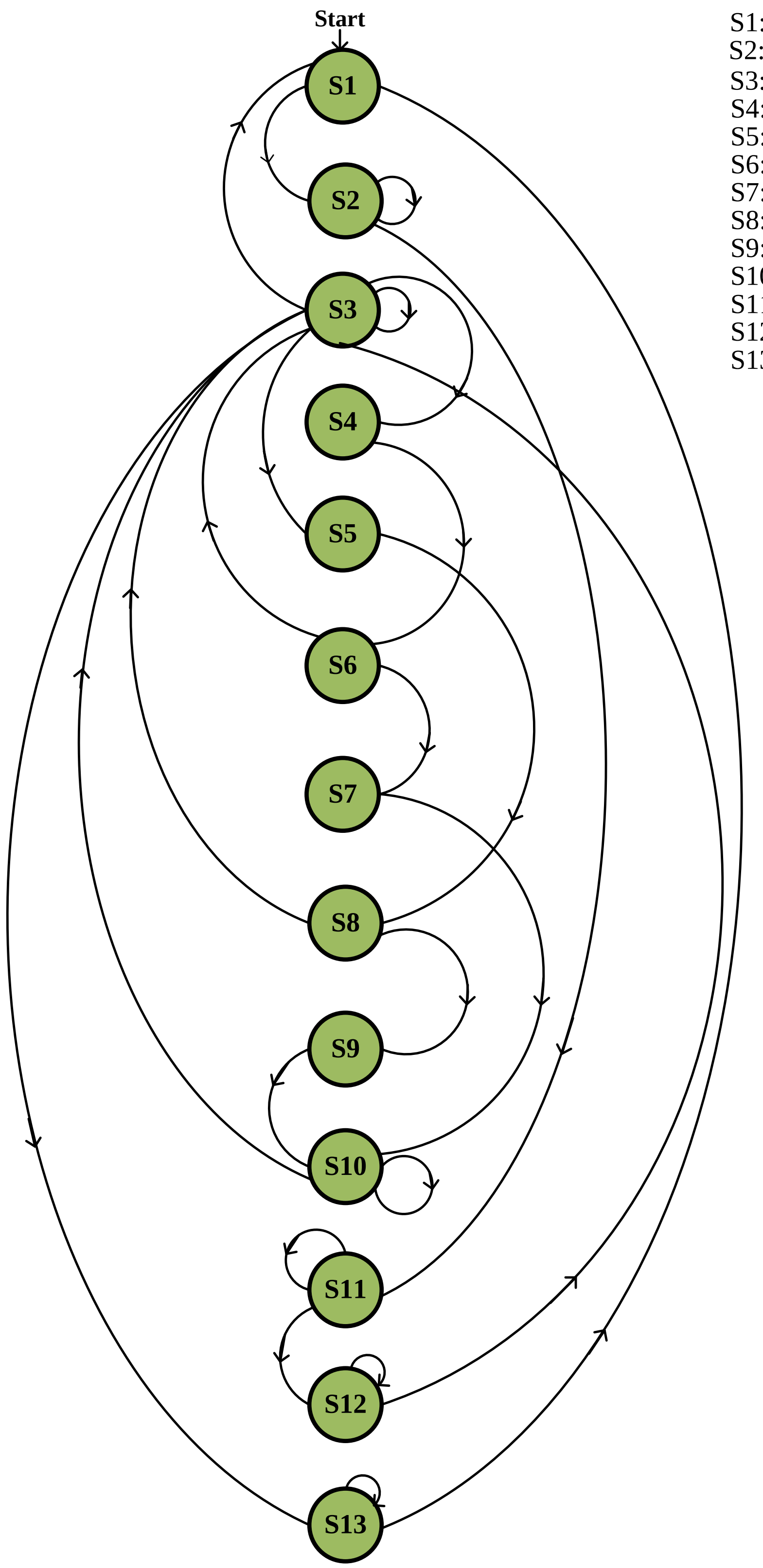


Red	Green	Blue	Color
0	0	0	Black
0	0	1	Blue
0	1	0	Green
0	1	1	Cyan
1	0	0	Red
1	0	1	Magenta
1	1	0	Yellow
1	1	1	White

Figure 9: Three bit RGB colors

4.1.4 Tic Tac Toe state machine

The state machine is design to make the game as simple as possible. The transition of each state is illustrated in the figure 10. The game start with the new game and then clear the screen, after that it will draw the horizantal and vertical lines to make the tic tac board. Then wait the first player to mark the specific location of the board. When the first player finish there turn it will press the turn end key so that second player will mark the specified location. After each turn, the status of the game will check whether any player win the game or not.



S1: New_game_state
S2: Screen_clear_state
S3: Ply_wait_state
S4: Ply_1_coordinate_state
S5: Ply_2_coordinate_state
S6: Ply_1_state
S7: Ply_1_mark_state
S8: Ply_2_state
S9: Ply_2_mark_state
S10: Box_draw_state
S11: Horizantal_draw_state
S12: Horizantal_draw_state
S13: Win_state

Figure 10: State diagram of tic tac toe

4.1.5 Seven segement display

In this project three 7-segment components (b2v_inst6, b2v_inst7, and b2v_inst8) are used one for each HEX0, HEX1 and HEX3 to display the status of the game on DE2-115 board. This component takes input from the tic tac toe state machine component and displays output on the DE2-115 board. The details of each seven segments are shown in the table 3.

7 segment	Display	Description
HEX3	P	It display P
HEX2	L	It display L
HEX0	1	When player 1 turn display 1
HEX0	2	When player 2 turn display 2
HEX0	A	When player 1 win display A
HEX0	b	When player 2 turn display b
HEX0	E	When match tie display E

Table 3: 7 segment displays

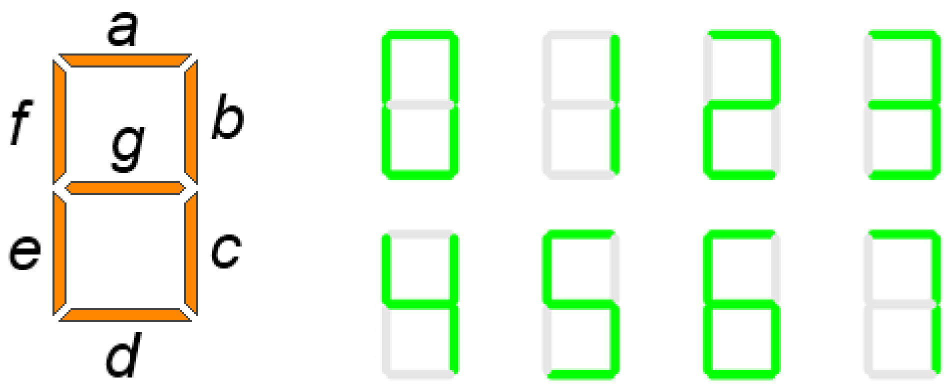


Figure: Seven segment display

4.1.6 LEDS display

In this project two LED display componet (b2v_inst4, b2v_inst5) are used one for each player. This component display the status of the game on DE2_115 boards LEDS .

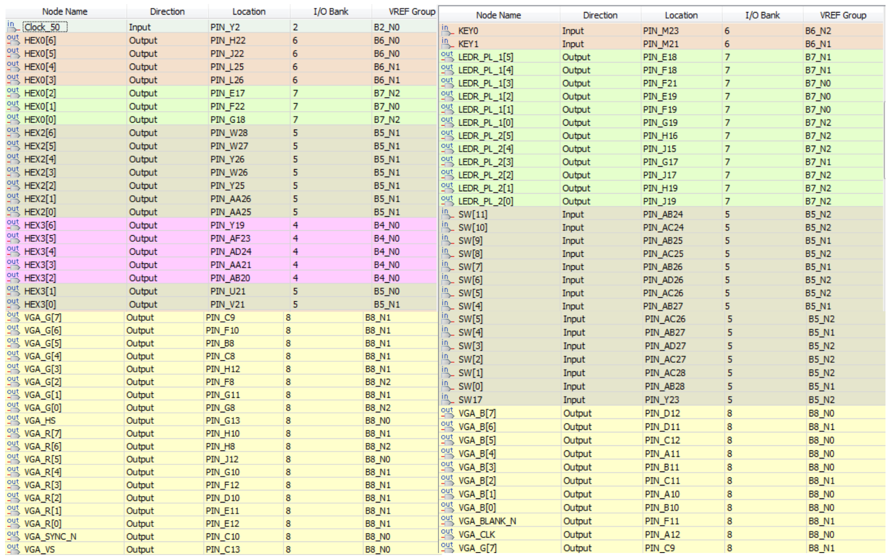
The b2v_inst4 component used first six red leds and b2v_inst5 used next six red leds. If player one have turn the first six red leds are turn ON and when player two have turn then next six red leds are turn ON. If any of player win or match is tie between them then all 12 leds are turn ON.

4.2 Meta Stability Problems resolving

To avoid the problem of metastability all asynchronous inputs clocked by two registers. This logic is applies to all signals coming from switches and push button of the development board.

4.3 I/O pins

All inputs and output pins used in the construction are same named as listed in the DE2-115 card manual. The following figure shows all the I/O pins that are used in the construction



Node Name	Direction	Location	I/O Bank	VREF Group
Clock_50	Input	PIN_Y2	2	B2_N0
HEX0[6]	Output	PIN_H22	6	B6_N0
HEX0[5]	Output	PIN_J22	6	B6_N0
HEX0[4]	Output	PIN_L25	6	B6_N1
HEX0[3]	Output	PIN_L26	6	B6_N1
HEX0[2]	Output	PIN_E17	7	B7_N2
HEX0[1]	Output	PIN_F22	7	B7_N0
HEX0[0]	Output	PIN_G18	7	B7_N2
HEX2[6]	Output	PIN_W28	5	B5_N1
HEX2[5]	Output	PIN_W27	5	B5_N1
HEX2[4]	Output	PIN_Y26	5	B5_N1
HEX2[3]	Output	PIN_W26	5	B5_N1
HEX2[2]	Output	PIN_Y25	5	B5_N1
HEX2[1]	Output	PIN_AA26	5	B5_N1
HEX2[0]	Output	PIN_AA25	5	B5_N1
HEX3[6]	Output	PIN_Y19	4	B4_N0
HEX3[5]	Output	PIN_AF23	4	B4_N0
HEX3[4]	Output	PIN_AD24	4	B4_N0
HEX3[3]	Output	PIN_AA21	4	B4_N0
HEX3[2]	Output	PIN_AB20	4	B4_N0
HEX3[1]	Output	PIN_U21	5	B5_N0
HEX3[0]	Output	PIN_V21	5	B5_N1
VGA_G[7]	Output	PIN_C9	8	B8_N1
VGA_G[6]	Output	PIN_F10	8	B8_N1
VGA_G[5]	Output	PIN_B8	8	B8_N1
VGA_G[4]	Output	PIN_C8	8	B8_N1
VGA_G[3]	Output	PIN_H12	8	B8_N1
VGA_G[2]	Output	PIN_F8	8	B8_N2
VGA_G[1]	Output	PIN_G11	8	B8_N1
VGA_G[0]	Output	PIN_G8	8	B8_N2
VGA_HS	Output	PIN_G13	8	B8_N0
VGA_R[7]	Output	PIN_H10	8	B8_N1
VGA_R[6]	Output	PIN_H8	8	B8_N2
VGA_R[5]	Output	PIN_J12	8	B8_N0
VGA_R[4]	Output	PIN_G10	8	B8_N1
VGA_R[3]	Output	PIN_F12	8	B8_N1
VGA_R[2]	Output	PIN_D10	8	B8_N1
VGA_R[1]	Output	PIN_E11	8	B8_N1
VGA_R[0]	Output	PIN_E12	8	B8_N1
VGA_SYNC_N	Output	PIN_C10	8	B8_N0
VGA_VS	Output	PIN_C13	8	B8_N0

Node Name	Direction	Location	I/O Bank	VREF Group
KEY0	Input	PIN_M23	6	B6_N2
KEY1	Input	PIN_M21	6	B6_N1
LEDR_PL_1[5]	Output	PIN_E18	7	B7_N1
LEDR_PL_1[4]	Output	PIN_F18	7	B7_N1
LEDR_PL_1[3]	Output	PIN_F21	7	B7_N0
LEDR_PL_1[2]	Output	PIN_E19	7	B7_N0
LEDR_PL_1[1]	Output	PIN_F19	7	B7_N0
LEDR_PL_1[0]	Output	PIN_G19	7	B7_N2
LEDR_PL_2[5]	Output	PIN_H16	7	B7_N2
LEDR_PL_2[4]	Output	PIN_J15	7	B7_N2
LEDR_PL_2[3]	Output	PIN_G17	7	B7_N1
LEDR_PL_2[2]	Output	PIN_J17	7	B7_N2
LEDR_PL_2[1]	Output	PIN_H19	7	B7_N2
LEDR_PL_2[0]	Output	PIN_J19	7	B7_N2
SW[11]	Input	PIN_AB24	5	B5_N2
SW[10]	Input	PIN_AC24	5	B5_N2
SW[9]	Input	PIN_AB25	5	B5_N1
SW[8]	Input	PIN_AC25	5	B5_N2
SW[7]	Input	PIN_AB26	5	B5_N1
SW[6]	Input	PIN_AD26	5	B5_N2
SW[5]	Input	PIN_AC26	5	B5_N2
SW[4]	Input	PIN_AB27	5	B5_N1
SW[5]	Input	PIN_AC26	5	B5_N2
SW[4]	Input	PIN_AB27	5	B5_N1
SW[3]	Input	PIN_AD27	5	B5_N2
SW[2]	Input	PIN_AC27	5	B5_N2
SW[1]	Input	PIN_AC28	5	B5_N2
SW[0]	Input	PIN_AB28	5	B5_N1
SW17	Input	PIN_Y23	5	B5_N2
VGA_B[7]	Output	PIN_D12	8	B8_N0
VGA_B[6]	Output	PIN_D11	8	B8_N1
VGA_B[5]	Output	PIN_C12	8	B8_N0
VGA_B[4]	Output	PIN_A11	8	B8_N0
VGA_B[3]	Output	PIN_B11	8	B8_N0
VGA_B[2]	Output	PIN_C11	8	B8_N1
VGA_B[1]	Output	PIN_A10	8	B8_N0
VGA_B[0]	Output	PIN_B10	8	B8_N0
VGA_BLANK_N	Output	PIN_F11	8	B8_N1
VGA_CLK	Output	PIN_A12	8	B8_N0
VGA_G[7]	Output	PIN_C9	8	B8_N1

4.4 Synopsys Design Constraints (SDC)

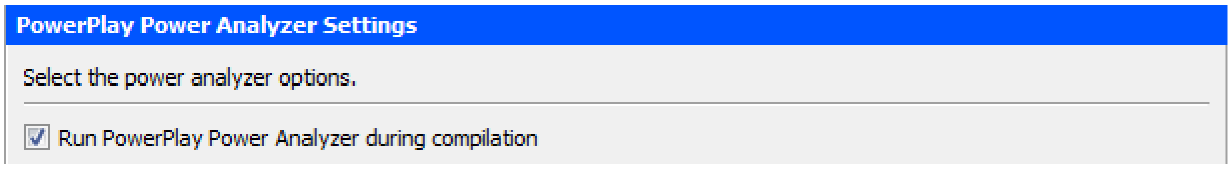
SDC file is generated by using TimeQuest Timing Analyzer Wizard toolbox in Quartus. The purpose of this file is to constraints clock, inputs and outputs and resolve the timing issues.

4.5 Tool settings and assignments

The following tools are set in Quartus II before compiling the code

4.5.1 Power Analysis

To run power analysis for the design selected Settings -> PowerPlay Power Analyzer Settings and the box for "Run PowerPlay Power Analyzer During compilation," is ticked as shown in the figure below



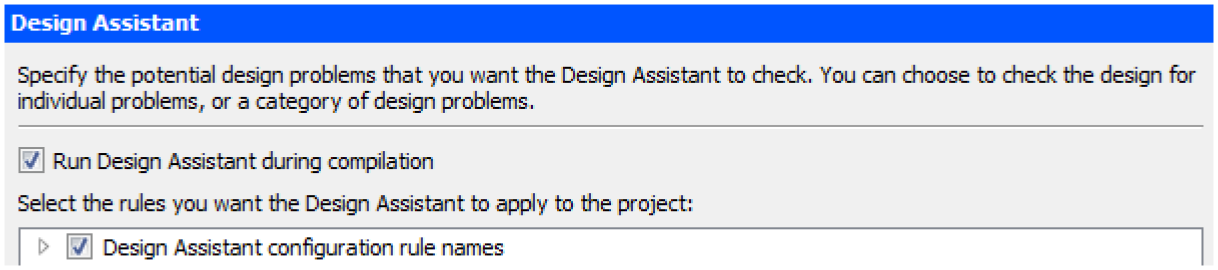
PowerPlay Power Analyzer Settings

Select the power analyzer options.

☑ Run PowerPlay Power Analyzer during compilation

4.5.2 Design Assistance

To run the Design Assistance for the design chosen Setting -> Design Assistance, and the box for "Run Design Assistance During compilation," is ticked as shown in the figure below


Design Assistant

Specify the potential design problems that you want the Design Assistant to check. You can choose to check the design for individual problems, or a category of design problems.

☑ Run Design Assistant during compilation

Select the rules you want the Design Assistant to apply to the project:

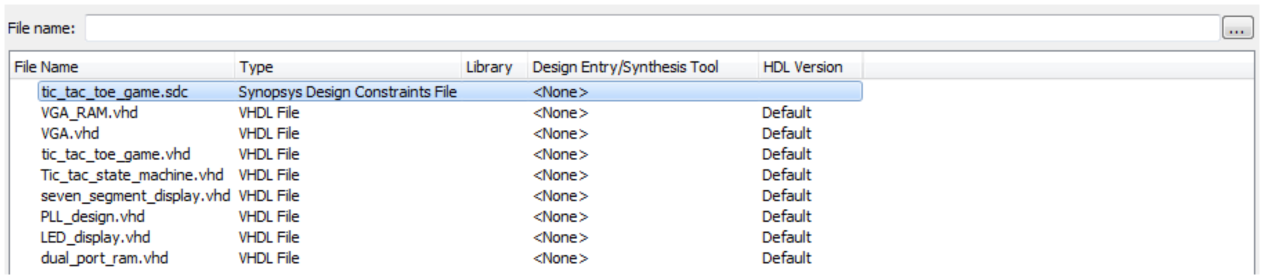
☑ Design Assistant configuration rule names

4.5.3 Pin Assignment

The pins are set according to the DE2-115 manual.

4.5.4 SDC

SDC file is added to the project by choosing Setting -> Time Quest Timing Analyzer and select the file in the "File name" and add. See the SDC file in the attachment part of this report.


File name:

File Name	Type	Library	Design Entry/Synthesis Tool	HDL Version
tic_tac_toe_game.sdc	Synopsys Design Constraints File		<None>	
VGA_RAM.vhd	VHDL File		<None>	Default
VGA.vhd	VHDL File		<None>	Default
tic_tac_toe_game.vhd	VHDL File		<None>	Default
Tic_tac_state_machine.vhd	VHDL File		<None>	Default
seven_segment_display.vhd	VHDL File		<None>	Default
PLL_design.vhd	VHDL File		<None>	Default
LED_display.vhd	VHDL File		<None>	Default
dual_port_ram.vhd	VHDL File		<None>	Default

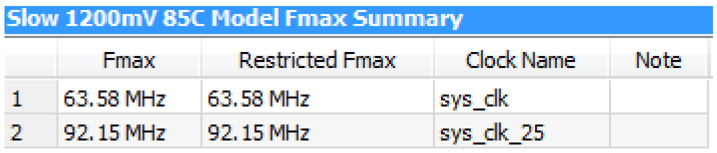
5 Analysis

In the beginning of this project there were many critical compilation warnings, but most of the warnings are removed except warning (169085)

- The warning (169085) as a result of no exact pin locations assignments for some pins. I leave these pins because of future use that why I didn’t assign pin no to these ports.

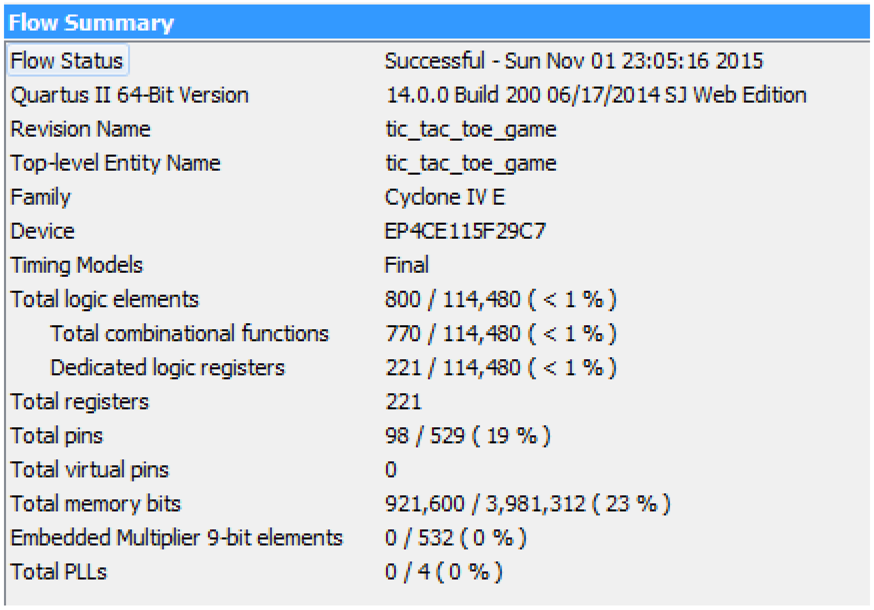
5.1 Maximum speed system clock

The figure shows FMAX for every clock in the design, regardless of the user-specified clock periods. FMAX is only computed for paths where the source and destination registers or ports are driven by the same clock. Paths of different clocks, including generated clocks are ignored.


Slow 1200mV 85C Model Fmax Summary

	Fmax	Restricted Fmax	Clock Name	Note
1	63.58 MHz	63.58 MHz	sys_clk	
2	92.15 MHz	92.15 MHz	sys_clk_25	

5.2 Size of the construction

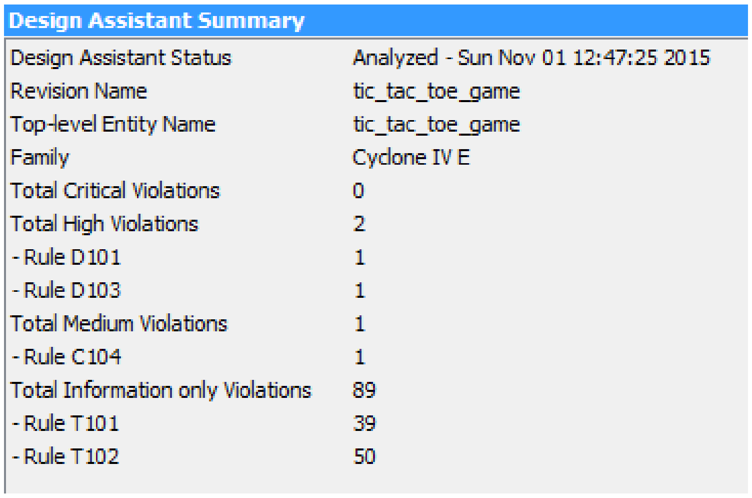


Flow Summary	
Flow Status	Successful - Sun Nov 01 23:05:16 2015
Quartus II 64-Bit Version	14.0.0 Build 200 06/17/2014 SJ Web Edition
Revision Name	tic_tac_toe_game
Top-level Entity Name	tic_tac_toe_game
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	800 / 114,480 (< 1 %)
Total combinational functions	770 / 114,480 (< 1 %)
Dedicated logic registers	221 / 114,480 (< 1 %)
Total registers	221
Total pins	98 / 529 (19 %)
Total virtual pins	0
Total memory bits	921,600 / 3,981,312 (23 %)
Embedded Multiplier 9-bit elements	0 / 532 (0 %)
Total PLLs	0 / 4 (0 %)

Figure 10: Logic elements used in FPGA

5.3 Design Assistant proposals

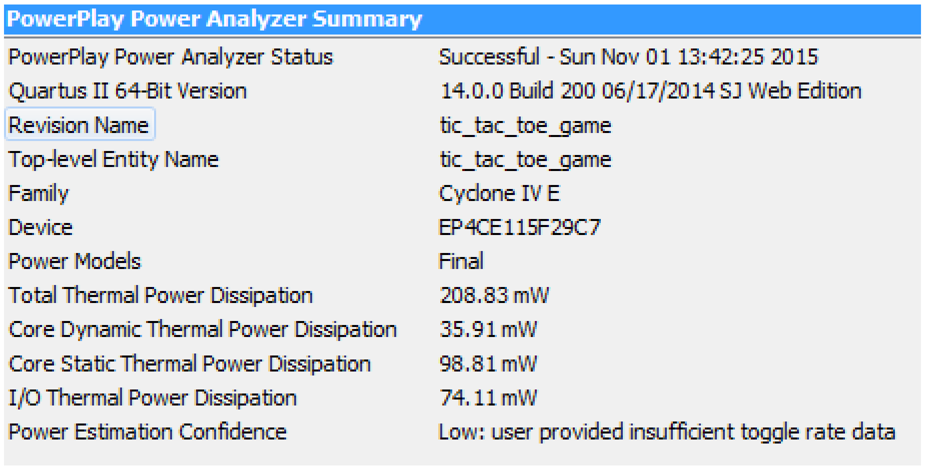
As shown in the design assistance summary it is stated that the construction has no critical violation.



Design Assistant Summary	
Design Assistant Status	Analyzed - Sun Nov 01 12:47:25 2015
Revision Name	tic_tac_toe_game
Top-level Entity Name	tic_tac_toe_game
Family	Cyclone IV E
Total Critical Violations	0
Total High Violations	2
- Rule D101	1
- Rule D103	1
Total Medium Violations	1
- Rule C104	1
Total Information only Violations	89
- Rule T101	39
- Rule T102	50

Figure 11: Design assistance summary

5.4 Power Estimation

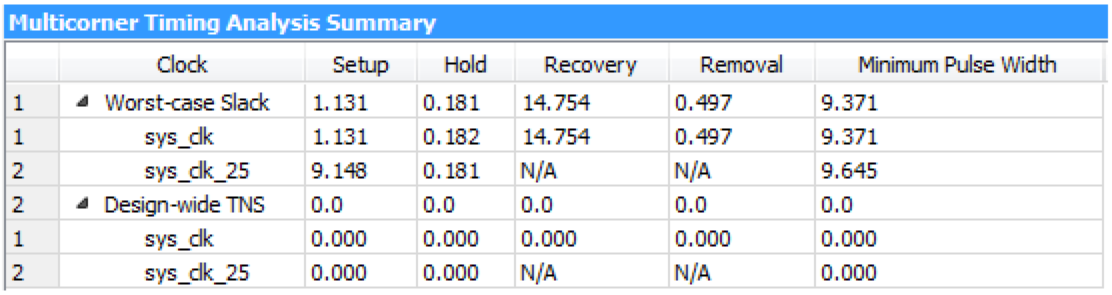


PowerPlay Power Analyzer Summary	
PowerPlay Power Analyzer Status	Successful - Sun Nov 01 13:42:25 2015
Quartus II 64-Bit Version	14.0.0 Build 200 06/17/2014 SJ Web Edition
Revision Name	tic_tac_toe_game
Top-level Entity Name	tic_tac_toe_game
Family	Cyclone IV E
Device	EP4CE115F29C7
Power Models	Final
Total Thermal Power Dissipation	208.83 mW
Core Dynamic Thermal Power Dissipation	35.91 mW
Core Static Thermal Power Dissipation	98.81 mW
I/O Thermal Power Dissipation	74.11 mW
Power Estimation Confidence	Low: user provided insufficient toggle rate data

Figure 12 PowerPlay power analyzer summary

5.5 Time Validation SDC results with slack

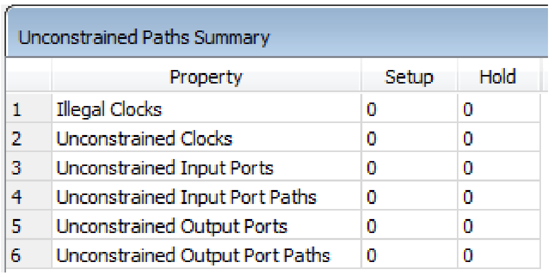
All the text has normal color in multicorner timing analyzer report. It means that all the inputs and outputs are constrained and have no timing problems.



Multicorner Timing Analysis Summary

	Clock	Setup	Hold	Recovery	Removal	Minimum Pulse Width
1	◢ Worst-case Slack	1.131	0.181	14.754	0.497	9.371
1	sys_clk	1.131	0.182	14.754	0.497	9.371
2	sys_clk_25	9.148	0.181	N/A	N/A	9.645
2	◢ Design-wide TNS	0.0	0.0	0.0	0.0	0.0
1	sys_clk	0.000	0.000	0.000	0.000	0.000
2	sys_clk_25	0.000	0.000	N/A	N/A	0.000

Figure 13: Time Quest Timing Analyzer



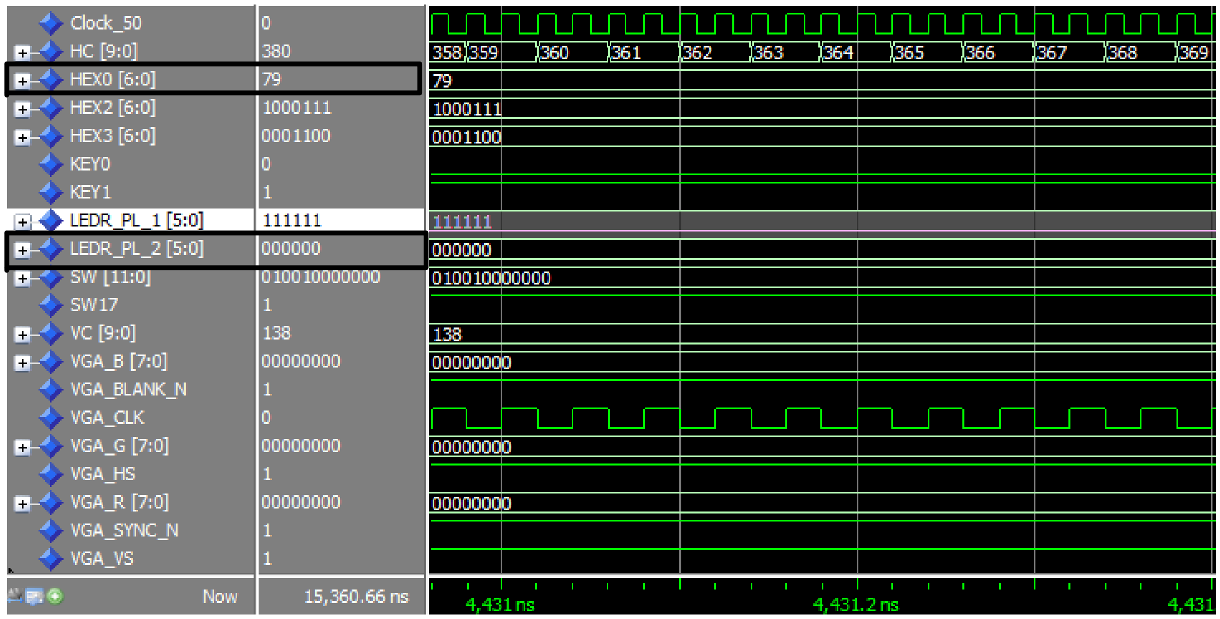
Unconstrained Paths Summary

	Property	Setup	Hold
1	Illegal Clocks	0	0
2	Unconstrained Clocks	0	0
3	Unconstrained Input Ports	0	0
4	Unconstrained Input Port Paths	0	0
5	Unconstrained Output Ports	0	0
6	Unconstrained Output Port Paths	0	0

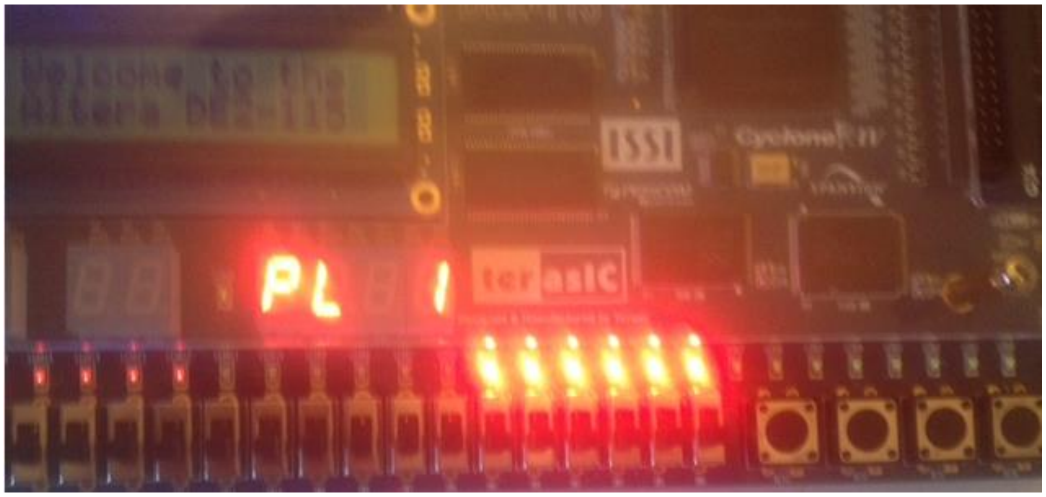
6 Verifications and Validation

6.1 Result from simulation

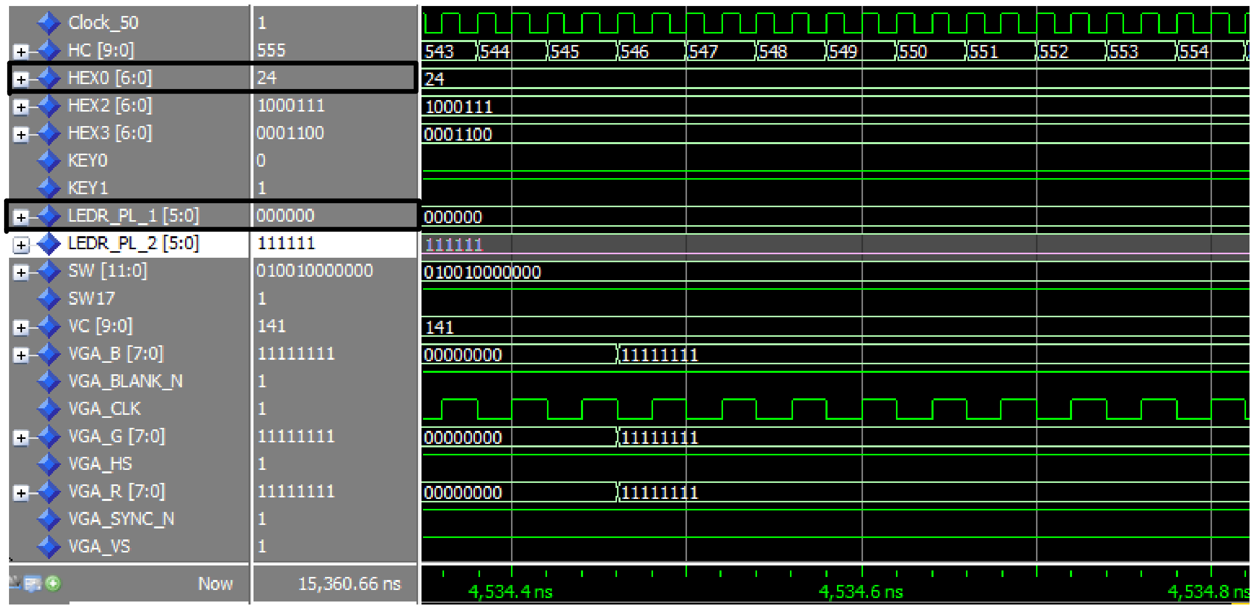
Case 1: The values are shown in hexadecimal format on HEX0 in simulation. The HEX0 shows 79 in simulation and display 1 on DE2-115 board.



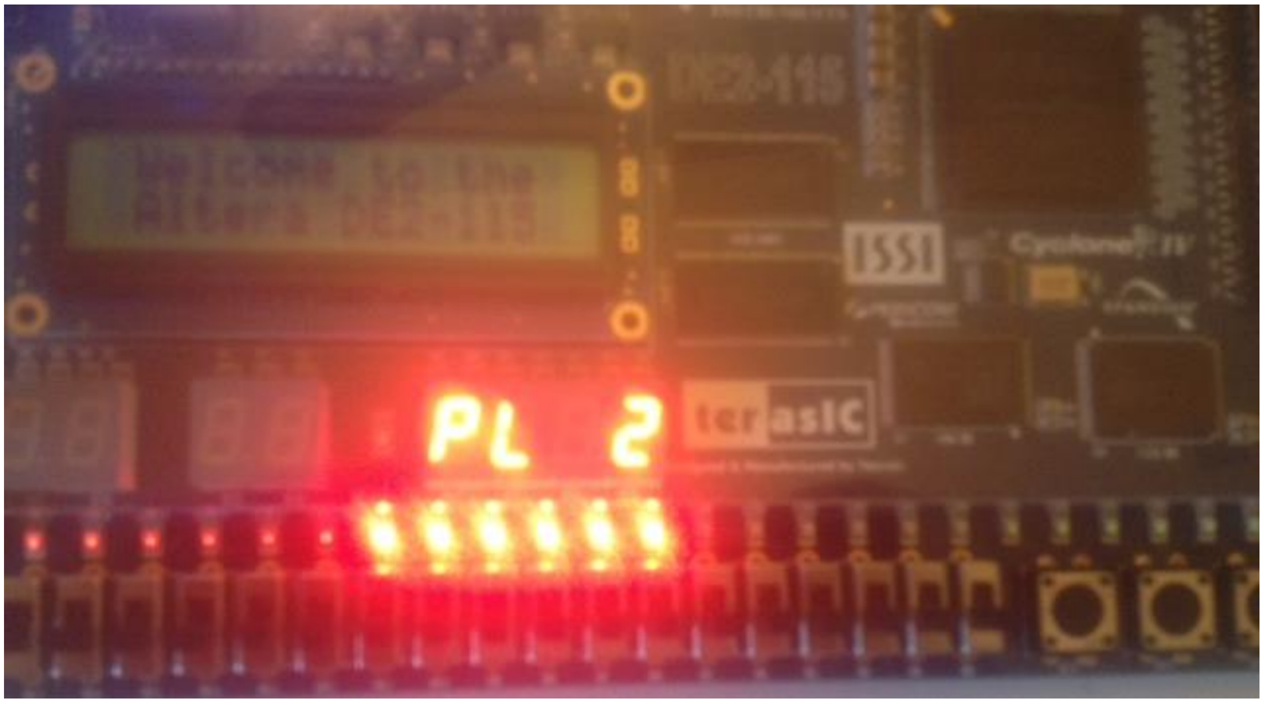
Clock_50	0	
HC [9:0]	380	358 359 360 361 362 363 364 365 366 367 368 369
HEX0 [6:0]	79	79
HEX2 [6:0]	1000111	1000111
HEX3 [6:0]	0001100	0001100
KEY0	0	
KEY1	1	
LEDR_PL_1 [5:0]	111111	111111
LEDR_PL_2 [5:0]	000000	000000
SW [11:0]	010010000000	010010000000
SW17	1	
VC [9:0]	138	138
VGA_B [7:0]	00000000	00000000
VGA_BLANK_N	1	
VGA_CLK	0	
VGA_G [7:0]	00000000	00000000
VGA_HS	1	
VGA_R [7:0]	00000000	00000000
VGA_SYNC_N	1	
VGA_VS	1	
Now	15,360.66 ns	4,431 ns 4,431.2 ns 4,431



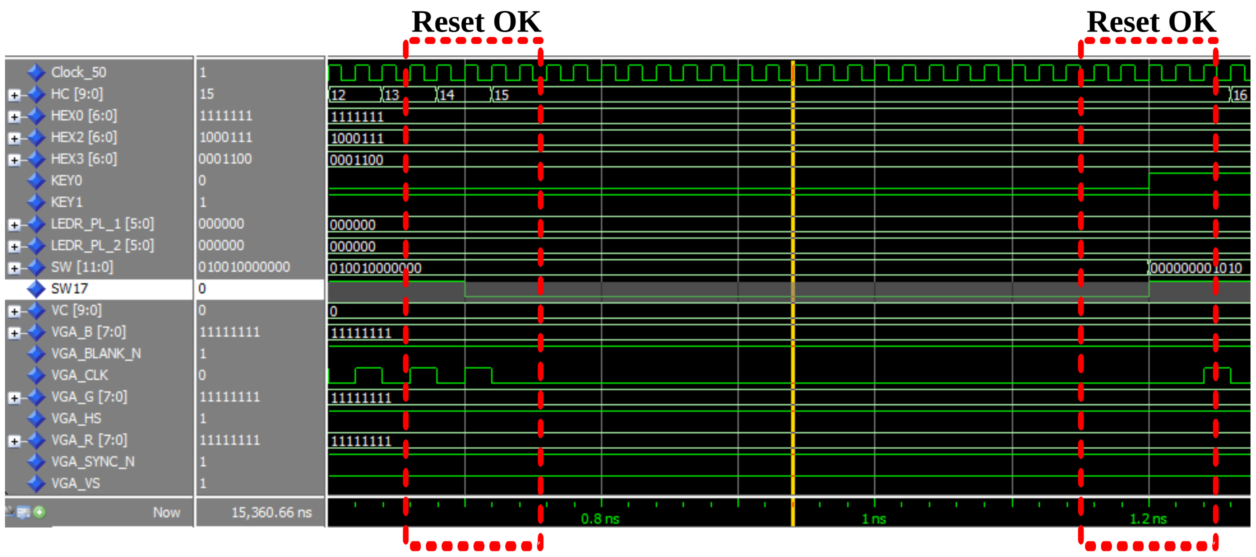
Case 2: The values are shown in hexadecimal format on HEX0 in simulation. The HEX0 shows 24 in simulation and display 2 on DE2-115 board.



```
Clock_50          1
HC [9:0]          555            543 544 545 546 547 548 549 550 551 552 553 554
HEX0 [6:0]        24             24
HEX2 [6:0]        1000111        1000111
HEX3 [6:0]        0001100        0001100
KEY0              0
KEY1              1
LEDR_PL_1 [5:0]   000000         000000
LEDR_PL_2 [5:0]   111111         111111
SW [11:0]         010010000000   010010000000
SW17              1
VC [9:0]          141            141
VGA_B [7:0]       11111111       00000000  11111111
VGA_BLANK_N       1
VGA_CLK           1
VGA_G [7:0]       11111111       00000000  11111111
VGA_HS            1
VGA_R [7:0]       11111111       00000000  11111111
VGA_SYNC_N        1
VGA_VS            1
Now               15,360.66 ns   4,534.4 ns   4,534.6 ns   4,534.8 ns
```

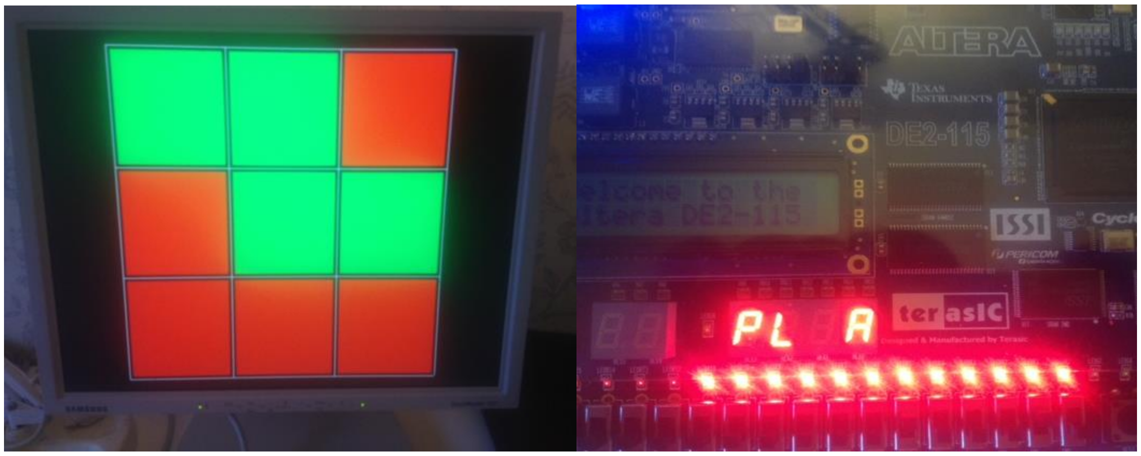


Case 3: Reset

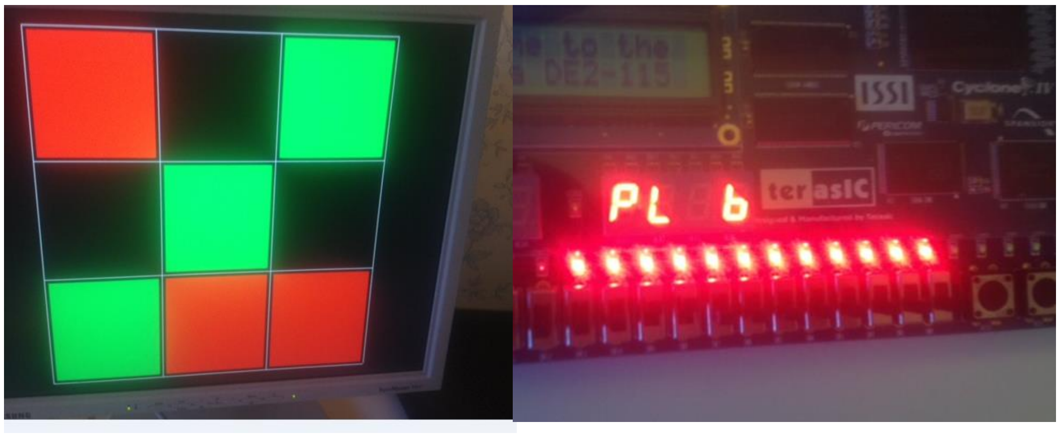


```
Clock_50          1
HC [9:0]          15
HEX0 [6:0]        1111111
HEX2 [6:0]        1000111
HEX3 [6:0]        0001100
KEY0              0
KEY1              1
LEDR_PL_1 [5:0]   000000
LEDR_PL_2 [5:0]   000000
SW [11:0]         010010000000
SW17              0
VC [9:0]          0
VGA_B [7:0]       11111111
VGA_BLANK_N       1
VGA_CLK           0
VGA_G [7:0]       11111111
VGA_HS            1
VGA_R [7:0]       11111111
VGA_SYNC_N        1
VGA_VS            1
Now               15,360.66 ns
```

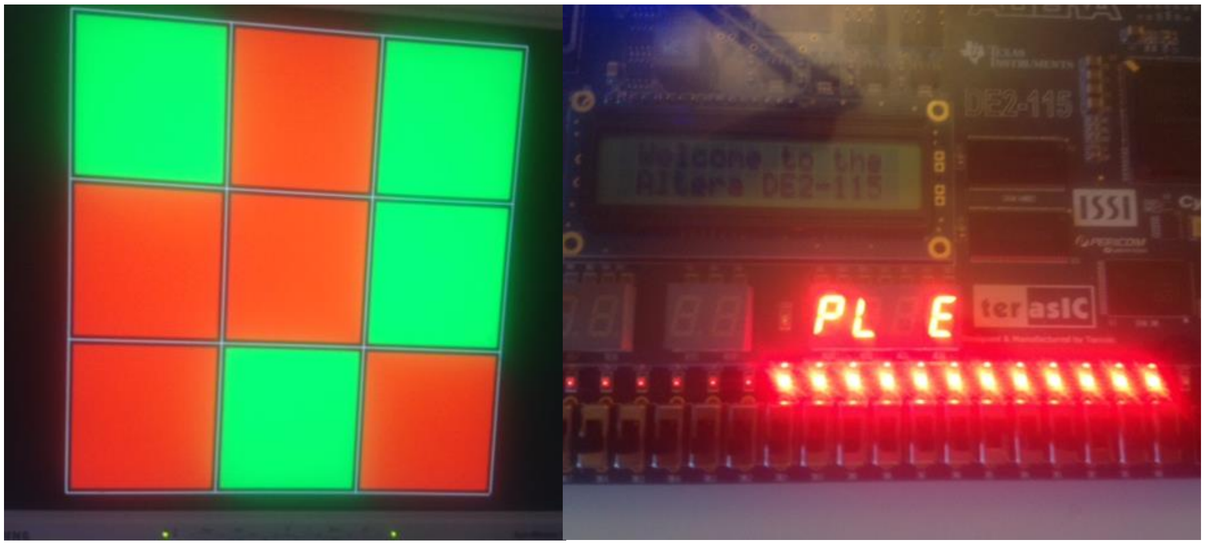
Case 4: When player 1 with mark red win the game.



Case 5 : When player 2 with mark green win the game.



Case 6: When match is tie between two players.



6.2 Validation

Table 1: Verification and validation of Game

Case	Description	OK	Verification Modelsim	Validation DE2-115
Test 1	SW=000000001010	LEDR_1=111111, LEDR_2=000000, HEX0=79	OK	OK
Test 2	SW=010010000000	LEDR_1=000000, LEDR_2=111111, HEX0=24	OK	OK
Test 3	Reset_n=0	No operation	OK	OK

7 Video link

https://www.youtube.com/watch?v=LC2IyUfyYc4

REFERENCES

[1] http://whatis.techtarget.com/definition/program-counter
[2] http://www.sqa.org.uk/e-learning/CompArch03CD/page_02.htm

APPENDIX

Tic Tac code

```
------------------------------------------------------------------------

LIBRARY ieee;

USE ieee.std_logic_1164.all;


LIBRARY work;


ENTITY tic_tac_toe_game IS

	PORT

	(

		SW17  :  IN  STD_LOGIC;

		Clock_50 :  IN  STD_LOGIC;

		KEY0 :  IN  STD_LOGIC;

		KEY1 :  IN  STD_LOGIC;

		SW :  IN  STD_LOGIC_VECTOR(11 DOWNTO 0);

		VGA_SYNC_N :  OUT  STD_LOGIC;

		VGA_BLANK_N :  OUT  STD_LOGIC;

		VGA_CLK :  OUT  STD_LOGIC;

		VGA_VS :  OUT  STD_LOGIC;

		VGA_HS :  OUT  STD_LOGIC;

		HC :  OUT  STD_LOGIC_VECTOR(9 DOWNTO 0);

		HEX0 :  OUT  STD_LOGIC_VECTOR(6 DOWNTO 0);

		HEX2 :  OUT  STD_LOGIC_VECTOR(6 DOWNTO 0);

		HEX3 :  OUT  STD_LOGIC_VECTOR(6 DOWNTO 0);
```

```
                LEDR_PL_1 :  OUT  STD_LOGIC_VECTOR(5 DOWNTO 0);

                LEDR_PL_2 :  OUT  STD_LOGIC_VECTOR(5 DOWNTO 0);

                VC :  OUT  STD_LOGIC_VECTOR(9 DOWNTO 0);

                VGA_B :  OUT  STD_LOGIC_VECTOR(7 DOWNTO 0);

                VGA_G :  OUT  STD_LOGIC_VECTOR(7 DOWNTO 0);

                VGA_R :  OUT  STD_LOGIC_VECTOR(7 DOWNTO 0)

        );

END tic_tac_toe_game;


ARCHITECTURE bdf_type OF tic_tac_toe_game IS


COMPONENT vga

        PORT(reset_n : IN STD_LOGIC;

                CLOCK_50 : IN STD_LOGIC;

                CLOCK_25 : IN STD_LOGIC;

                data_vga_r : IN STD_LOGIC_VECTOR(2 DOWNTO 0);

                VGA_HS : OUT STD_LOGIC;

                VGA_VS : OUT STD_LOGIC;

                VGA_CLK : OUT STD_LOGIC;

                VGA_BLANK_N : OUT STD_LOGIC;

                VGA_SYNC_N : OUT STD_LOGIC;

                HC : OUT STD_LOGIC_VECTOR(9 DOWNTO 0);

                VC : OUT STD_LOGIC_VECTOR(9 DOWNTO 0);

                vga_address_r : OUT STD_LOGIC_VECTOR(18 DOWNTO 0);

                VGA_B : OUT STD_LOGIC_VECTOR(7 DOWNTO 0);
                VGA_G : OUT STD_LOGIC_VECTOR(7 DOWNTO 0);

                VGA_R : OUT STD_LOGIC_VECTOR(7 DOWNTO 0)

        );

END COMPONENT;


COMPONENT tic_tac_state_machine

        PORT(reset_n : IN STD_LOGIC;

                clock_50 : IN STD_LOGIC;

                trun_end : IN STD_LOGIC;

                new_game_n : IN STD_LOGIC;

                SW_plys : IN STD_LOGIC_VECTOR(11 DOWNTO 0);

                write_vga_n : OUT STD_LOGIC;

                HEX0 : OUT STD_LOGIC_VECTOR(6 DOWNTO 0);

                HEX2 : OUT STD_LOGIC_VECTOR(6 DOWNTO 0);

                HEX3 : OUT STD_LOGIC_VECTOR(6 DOWNTO 0);

                LEDR_PL_1 : OUT STD_LOGIC_VECTOR(5 DOWNTO 0);

                LEDR_PL_2 : OUT STD_LOGIC_VECTOR(5 DOWNTO 0);

                vga_address_w : OUT STD_LOGIC_VECTOR(18 DOWNTO 0);

                vga_data_w : OUT STD_LOGIC_VECTOR(2 DOWNTO 0)

        );

END COMPONENT;


COMPONENT pll_design

        PORT(reset_n : IN STD_LOGIC;

                clkin_50 : IN STD_LOGIC;
                clkout_25 : OUT STD_LOGIC
        );
END COMPONENT;


COMPONENT led_display
        PORT(LED_input : IN STD_LOGIC_VECTOR(5 DOWNTO 0);
                LED_output : OUT STD_LOGIC_VECTOR(5 DOWNTO 0)
        );
END COMPONENT;


COMPONENT seven_segment_display
        PORT(seven_seg_input : IN STD_LOGIC_VECTOR(6 DOWNTO 0);
                Seven_seg_output : OUT STD_LOGIC_VECTOR(6 DOWNTO 0)
        );
END COMPONENT;


COMPONENT dual_port_ram
        PORT(wren : IN STD_LOGIC;
                wrclock : IN STD_LOGIC;
                rdclock : IN STD_LOGIC;
                data : IN STD_LOGIC_VECTOR(2 DOWNTO 0);
                rdaddress : IN STD_LOGIC_VECTOR(18 DOWNTO 0);
                wraddress : IN STD_LOGIC_VECTOR(18 DOWNTO 0);
                q : OUT STD_LOGIC_VECTOR(2 DOWNTO 0)
        );
```

```
END COMPONENT;


SIGNAL SYNTHESIZED_WIRE_0 :  STD_LOGIC;

SIGNAL SYNTHESIZED_WIRE_1 :  STD_LOGIC_VECTOR(2 DOWNTO 0);

SIGNAL SYNTHESIZED_WIRE_2 :  STD_LOGIC_VECTOR(5 DOWNTO 0);

SIGNAL SYNTHESIZED_WIRE_3 :  STD_LOGIC_VECTOR(5 DOWNTO 0);

SIGNAL SYNTHESIZED_WIRE_4 :  STD_LOGIC_VECTOR(6 DOWNTO 0);

SIGNAL SYNTHESIZED_WIRE_5 :  STD_LOGIC_VECTOR(6 DOWNTO 0);

SIGNAL SYNTHESIZED_WIRE_6 :  STD_LOGIC_VECTOR(6 DOWNTO 0);

SIGNAL SYNTHESIZED_WIRE_7 :  STD_LOGIC;

SIGNAL SYNTHESIZED_WIRE_8 :  STD_LOGIC_VECTOR(2 DOWNTO 0);

SIGNAL SYNTHESIZED_WIRE_9 :  STD_LOGIC_VECTOR(18 DOWNTO 0);

SIGNAL SYNTHESIZED_WIRE_10 :  STD_LOGIC_VECTOR(18 DOWNTO 0);

---------------------Metastability_signal_decleration----------------

signal sw_msp : std_logic_vector(11 downto 0);

signal sw_plys : std_logic_vector(11 downto 0);

signal reset_n : std_logic;

signal sw_17_msp: std_logic;

signal key0_msp_n : std_logic;

signal key1_msp_n : std_logic;

signal turn_end : std_logic;

signal new_game_n : std_logic;

-------------------------------------------------------------------

BEGIN

metastability_protection: process(clock_50)
```

```
  begin
              if rising_edge(clock_50) then
                          sw_17_msp <=sw17;
                               reset_n<=sw_17_msp;
                               sw_msp<=sw;
                               SW_plys<=SW_msp;
                     key0_msp_n <= KEY0;
                     turn_end <= key0_msp_n;
                     key1_msp_n <= KEY1;
                     new_game_n <= key1_msp_n;
            end if;
end process;
-----------------------



b2v_inst : vga
PORT MAP(reset_n => reset_n,
                  CLOCK_50 => Clock_50,
                  CLOCK_25 => SYNTHESIZED_WIRE_0,
                  data_vga_r => SYNTHESIZED_WIRE_1,
                  VGA_HS => VGA_HS,
                  VGA_VS => VGA_VS,
                  VGA_CLK => VGA_CLK,
                  VGA_BLANK_N => VGA_BLANK_N,
                  VGA_SYNC_N => VGA_SYNC_N,
                  HC => HC,

                  VC => VC,

                  vga_address_r => SYNTHESIZED_WIRE_9,

                  VGA_B => VGA_B,

                  VGA_G => VGA_G,

                  VGA_R => VGA_R);



b2v_inst1 : tic_tac_state_machine

PORT MAP(reset_n => reset_n,

                  clock_50 => Clock_50,

                  trun_end =>turn_end ,

                  new_game_n => new_game_n,

                  SW_plys => SW_plys,

                  write_vga_n => SYNTHESIZED_WIRE_7,

                  HEX0 => SYNTHESIZED_WIRE_4,

                  HEX2 => SYNTHESIZED_WIRE_6,

                  HEX3 => SYNTHESIZED_WIRE_5,

                  LEDR_PL_1 => SYNTHESIZED_WIRE_2,

                  LEDR_PL_2 => SYNTHESIZED_WIRE_3,

                  vga_address_w => SYNTHESIZED_WIRE_10,

                  vga_data_w => SYNTHESIZED_WIRE_8);



b2v_inst3 : pll_design
```

```
PORT MAP(reset_n => reset_n,
                    clkin_50 => Clock_50,
                    clkout_25 => SYNTHESIZED_WIRE_0);



b2v_inst4 : led_display
PORT MAP(LED_input => SYNTHESIZED_WIRE_2,
                    LED_output => LEDR_PL_1);



b2v_inst5 : led_display
PORT MAP(LED_input => SYNTHESIZED_WIRE_3,
                    LED_output => LEDR_PL_2);



b2v_inst6 : seven_segment_display
PORT MAP(seven_seg_input => SYNTHESIZED_WIRE_4,
                    Seven_seg_output => HEX0);



b2v_inst7 : seven_segment_display
PORT MAP(seven_seg_input => SYNTHESIZED_WIRE_5,
                    Seven_seg_output => HEX3);
```

```
b2v_inst8 : seven_segment_display

PORT MAP(seven_seg_input => SYNTHESIZED_WIRE_6,

         Seven_seg_output => HEX2);



b2v_inst9 : dual_port_ram

PORT MAP(wren => SYNTHESIZED_WIRE_7,

         wrclock => Clock_50,

         rdclock => Clock_50,

         data => SYNTHESIZED_WIRE_8,

         rdaddress => SYNTHESIZED_WIRE_9,

         wraddress => SYNTHESIZED_WIRE_10,

         q => SYNTHESIZED_WIRE_1);



END bdf_type;
```

Test Bench

```
LIBRARY ieee;

USE ieee.std_logic_1164.all;


ENTITY tic_tac_toe_game_vhd_tst IS

END tic_tac_toe_game_vhd_tst;

ARCHITECTURE tic_tac_toe_game_arch OF tic_tac_toe_game_vhd_tst IS
```

```
-- constants
-- signals
SIGNAL Clock_50 : STD_LOGIC;
SIGNAL HC : STD_LOGIC_VECTOR(9 DOWNTO 0);
SIGNAL HEX0 : STD_LOGIC_VECTOR(6 DOWNTO 0);
SIGNAL HEX2 : STD_LOGIC_VECTOR(6 DOWNTO 0);
SIGNAL HEX3 : STD_LOGIC_VECTOR(6 DOWNTO 0);
SIGNAL KEY0 : STD_LOGIC;
SIGNAL KEY1 : STD_LOGIC;
SIGNAL LEDR_PL_1 : STD_LOGIC_VECTOR(5 DOWNTO 0);
SIGNAL LEDR_PL_2 : STD_LOGIC_VECTOR(5 DOWNTO 0);
SIGNAL SW : STD_LOGIC_VECTOR(11 DOWNTO 0);
SIGNAL SW17 : STD_LOGIC;
SIGNAL VC : STD_LOGIC_VECTOR(9 DOWNTO 0);
SIGNAL VGA_B : STD_LOGIC_VECTOR(7 DOWNTO 0);
SIGNAL VGA_BLANK_N : STD_LOGIC;
SIGNAL VGA_CLK : STD_LOGIC;
SIGNAL VGA_G : STD_LOGIC_VECTOR(7 DOWNTO 0);
SIGNAL VGA_HS : STD_LOGIC;
SIGNAL VGA_R : STD_LOGIC_VECTOR(7 DOWNTO 0);
SIGNAL VGA_SYNC_N : STD_LOGIC;
SIGNAL VGA_VS : STD_LOGIC;
COMPONENT tic_tac_toe_game
        PORT (
        Clock_50 : IN STD_LOGIC;

        HC : BUFFER STD_LOGIC_VECTOR(9 DOWNTO 0);

        HEX0 : BUFFER STD_LOGIC_VECTOR(6 DOWNTO 0);

        HEX2 : BUFFER STD_LOGIC_VECTOR(6 DOWNTO 0);

        HEX3 : BUFFER STD_LOGIC_VECTOR(6 DOWNTO 0);

        KEY0 : IN STD_LOGIC;

        KEY1 : IN STD_LOGIC;

        LEDR_PL_1 : BUFFER STD_LOGIC_VECTOR(5 DOWNTO 0);

        LEDR_PL_2 : BUFFER STD_LOGIC_VECTOR(5 DOWNTO 0);

        SW : IN STD_LOGIC_VECTOR(11 DOWNTO 0);

        SW17 : IN STD_LOGIC;

        VC : BUFFER STD_LOGIC_VECTOR(9 DOWNTO 0);

        VGA_B : BUFFER STD_LOGIC_VECTOR(7 DOWNTO 0);

        VGA_BLANK_N : BUFFER STD_LOGIC;

        VGA_CLK : BUFFER STD_LOGIC;

        VGA_G : BUFFER STD_LOGIC_VECTOR(7 DOWNTO 0);

        VGA_HS : BUFFER STD_LOGIC;

        VGA_R : BUFFER STD_LOGIC_VECTOR(7 DOWNTO 0);

        VGA_SYNC_N : BUFFER STD_LOGIC;

        VGA_VS : BUFFER STD_LOGIC

        );

END COMPONENT;

BEGIN

        i1 : tic_tac_toe_game
```

```
	PORT MAP (
-- list connections between master ports and signals
	Clock_50 => Clock_50,
	HC => HC,
	HEX0 => HEX0,
	HEX2 => HEX2,
	HEX3 => HEX3,
	KEY0 => KEY0,
	KEY1 => KEY1,
	LEDR_PL_1 => LEDR_PL_1,
	LEDR_PL_2 => LEDR_PL_2,
	SW => SW,
	SW17 => SW17,
	VC => VC,
	VGA_B => VGA_B,
	VGA_BLANK_N => VGA_BLANK_N,
	VGA_CLK => VGA_CLK,
	VGA_G => VGA_G,
	VGA_HS => VGA_HS,
	VGA_R => VGA_R,
	VGA_SYNC_N => VGA_SYNC_N,
	VGA_VS => VGA_VS
	);
--------------------------------------
```

```
clk_signal:process

  begin

          clock_50<='1';

          wait for 10 ps;

          clock_50<='0';

          wait for 10 ps;

end process clk_signal;

---------------reset signal-------------

init : PROCESS

-- variable declarations

BEGIN

----------------------------------

SW17 <= '0';

wait for 100 ps;

SW17 <= '1';

----------------------------------

SW <= B"000000001001";

KEY1 <= '1';

KEY0 <= '1';

WAIT FOR 150 ps;

KEY0 <= '0';

WAIT FOR 150 ps;

-----------------------------------

SW <= B"010010000000";
```

```
KEY0 <= '1';

WAIT FOR 150 ps;

KEY0 <= '0';

WAIT FOR 150 ps;

--------reset Case--------

SW17 <= '0';

wait for 500 ps;

SW17 <= '1';

----------Case 1---------------

SW <= B"000000001010";

KEY0 <= '1';

WAIT FOR 100 ps;

KEY0 <= '0';

WAIT FOR 300 ps;

---------------Case 2 ----

SW <= B"010010000000";

KEY0 <= '1';

WAIT FOR 100 ps;

KEY0 <= '0';

WAIT FOR 300 ps;

----------------------------------

WAIT;

END PROCESS init;

always : PROCESS
```

```
-- optiona l sensitivity list
-- (        )
-- variable declarations
BEGIN
        -- code executes for every event on sensitivity list
WAIT;
END PROCESS always;
END tic_tac_toe_game_arch;
```

Do file

```
onerror {resume}
quietly WaveActivateNextPane {} 0
add wave -noupdate /tic_tac_toe_game_vhd_tst/Clock_50
add wave -noupdate -radix unsigned /tic_tac_toe_game_vhd_tst/HC
add wave -noupdate -radix hexadecimal /tic_tac_toe_game_vhd_tst/HEX0
add wave -noupdate /tic_tac_toe_game_vhd_tst/HEX2
add wave -noupdate /tic_tac_toe_game_vhd_tst/HEX3
add wave -noupdate /tic_tac_toe_game_vhd_tst/KEY0
add wave -noupdate /tic_tac_toe_game_vhd_tst/KEY1
add wave -noupdate /tic_tac_toe_game_vhd_tst/LEDR_PL_1
add wave -noupdate /tic_tac_toe_game_vhd_tst/LEDR_PL_2
add wave -noupdate /tic_tac_toe_game_vhd_tst/SW
add wave -noupdate /tic_tac_toe_game_vhd_tst/SW17
add wave -noupdate -radix unsigned /tic_tac_toe_game_vhd_tst/VC
add wave -noupdate /tic_tac_toe_game_vhd_tst/VGA_B
add wave -noupdate /tic_tac_toe_game_vhd_tst/VGA_BLANK_N

add wave -noupdate /tic_tac_toe_game_vhd_tst/VGA_CLK

add wave -noupdate /tic_tac_toe_game_vhd_tst/VGA_G

add wave -noupdate /tic_tac_toe_game_vhd_tst/VGA_HS

add wave -noupdate /tic_tac_toe_game_vhd_tst/VGA_R

add wave -noupdate /tic_tac_toe_game_vhd_tst/VGA_SYNC_N

add wave -noupdate /tic_tac_toe_game_vhd_tst/VGA_VS

TreeUpdate [SetDefaultTree]

WaveRestoreCursors {{Cursor 1} {4534844 ps} 0}

quietly wave cursor active 1

configure wave -namecolwidth 150

configure wave -valuecolwidth 100

configure wave -justifyvalue left

configure wave -signalnamewidth 1

configure wave -snapdistance 10

configure wave -datasetprefix 0

configure wave -rowmargin 4

configure wave -childrowmargin 2

configure wave -gridoffset 0

configure wave -gridperiod 1000

configure wave -griddelta 40

configure wave -timeline 0

configure wave -timelineunits ns

update
```

```
WaveRestoreZoom {4534348 ps} {4535348 ps}
```