



DSD MINI PROJECT REPORT

IMPLEMENTATION OF 2048 GAME

ON NEXYS 4 ARTIX 7 FPGA

SUBMITTED TO:

Mr. KIRAN KUMAR LAD

SUBMITTED BY:

NAVEEN JOHN 12EC67

ROUNAK SENGUPTA 12EC80

VIVEK VENKATRAMAN 12EC106

ABSTRACT

2048 is a single-player puzzle game created in March 2014, which is massively popular on both the android and the iOS platforms.

HOW TO PLAY

2048 is played on a simple gray 4×4 grid, with numbered tiles that slide smoothly when a player moves them using the four arrow keys. Every turn, a new tile will randomly appear in an empty spot on the board with a value of either 2 or 4. Tiles slide as far as possible in the chosen direction until they are stopped by either another tile or the edge of the grid. If two tiles of the same number collide while moving, they will merge into a tile with the total value of the two tiles that collided. The game is won when a tile with a value of 2048 appears on the board, hence the name of the game. When the player has no legal moves (there are no empty spaces and no adjacent tiles with the same value), the game ends.

IMAGES FROM THE GAME



DESCRIPTION

To implement the game on the Nexys 4 Artix 7 FPGA, we have utilised the Xilinx ISE software and the Xilinx PlanAhead tool.

The game consists of 3 stages,

- Obtaining the input from the user
- Processing the input on the FPGA
- Sending the corresponding output to the VGA display

The VGA with a resolution of 1280 x 1066 requires a refresh rate of 108Mhz. Since the onboard clock on the FPGA is of 100MHz, we utilised the IP core generator to generate a component which gives the 108MHz clock output for a 100Mhz input.

The user is given an option to move the tiles on the screen in 4 directions (left, right, down and up). He controls the direction of movement based on 2 control switches on the board.

To move in a particular direction, he has to set the switches to the desired value (00 – down / 01 – up / 10 – right / 11 – left) and then press a button when he is ready to supply the input . Since we are using a push button clock, we need to debounce this clock. This task is performed by another component which we have port mapped in the main .vhd file.

In case the user wants to restart the whole game, he is provided with a reset button for the same.

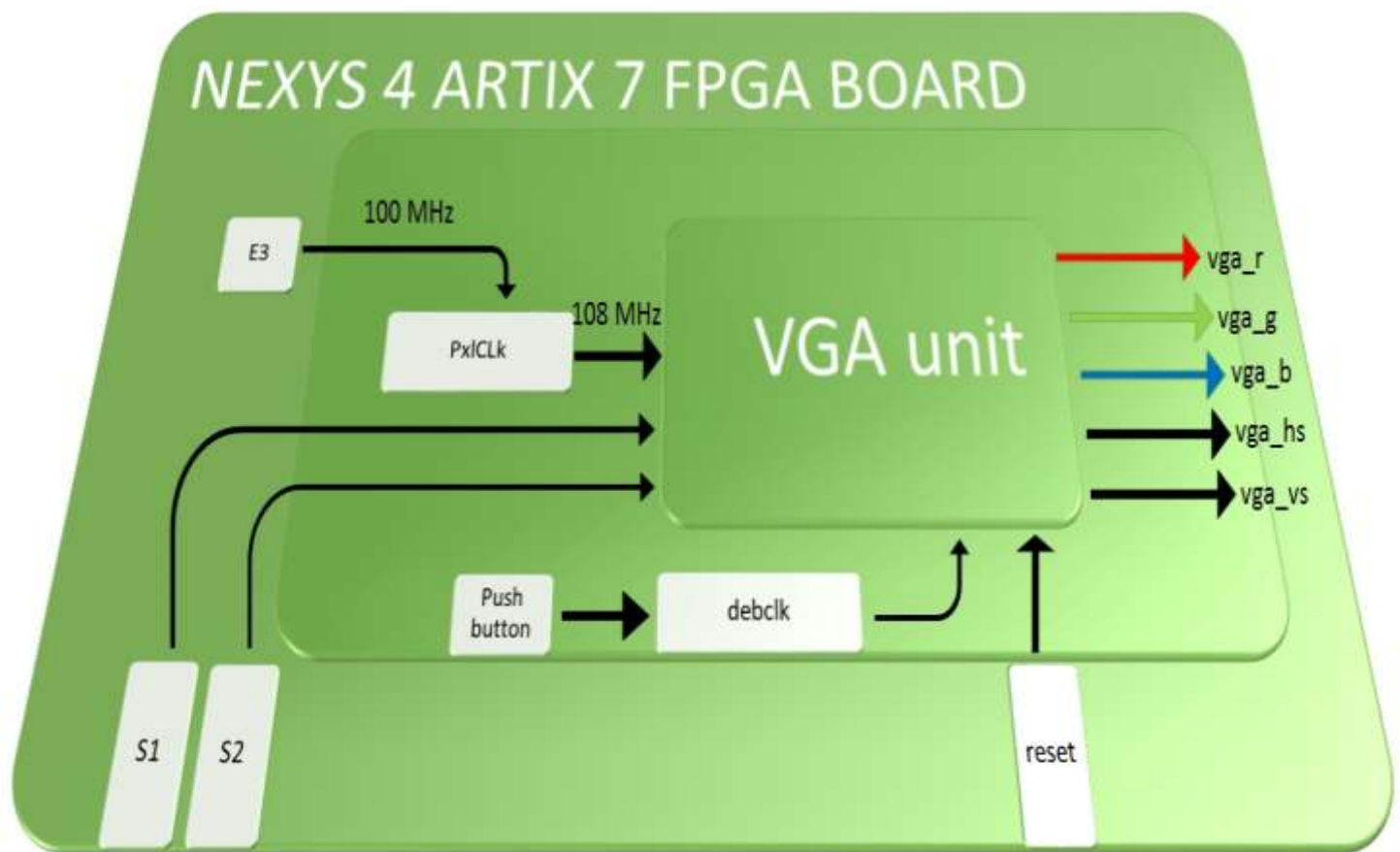
The game has been designed in the following manner:

There is a 4*4 array which is created as a shared variable since it is being manipulated in multiple process statements .This array stores the value of the number of tile. When the user chooses to move in a particular direction, the array is manipulated and updated.

As soon as the array value changes, the respective conditions for the pixels changes and they display a new number on the VGA. When the user manages to win the game, i.e get the 2048 tile, an LED lights up on the board. In case he is unable to proceed further and does not have any movements left, the end of the game is indicated by another LED.

After every movement, a new element should be added at a random location in the array and this is achieved by adding the element at the first empty location.

BLOCK DIAGRAM OF DIGITAL SYSTEM



HOW TO PLAY THE GAME

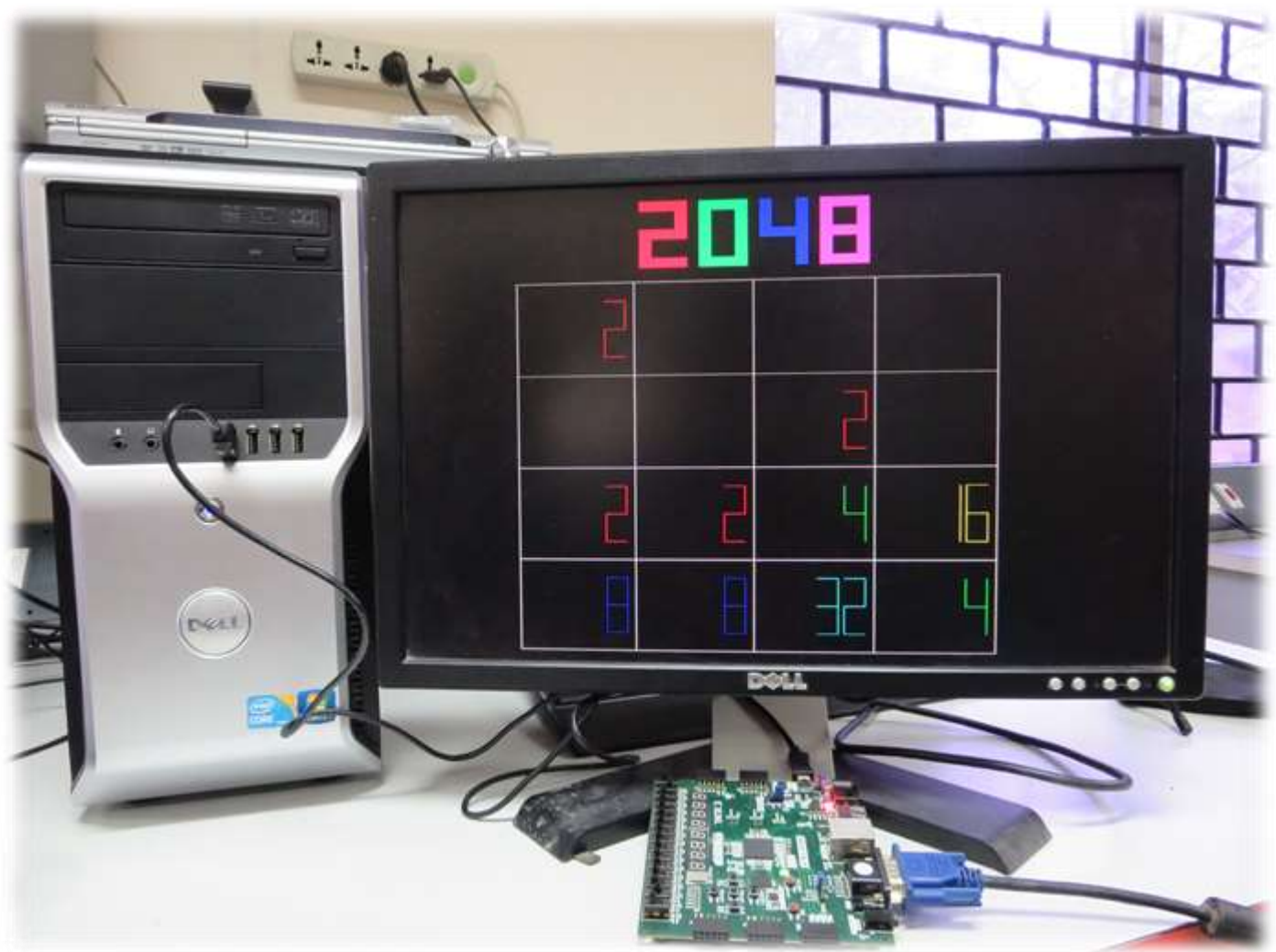
Set switches at one of the following positions and then use the push button to confirm choice.

S1	S2	DIRECTION OF MOTION
0	0	DOWN
0	1	RIGHT
1	0	UP
1	1	LEFT

RESULTS



2048 running on the VGA display



CONCLUSION:

In the project we have successfully implemented a digital system which can be used as a gaming device. The programmed digital system consists of different units like a pixel clock generator, which is essentially a PLL, a VGA unit, which channels the output of the system to the VGA screen, a clock buffer unit and a reset unit.

Through this project we have acquired programming skills in VHDL and gained knowledge on how to develop a digital system. The future scope of this project involves developing multiple games on this board and using the same board for multi gaming purposes.

Also the minimization of components should be considered and we should try and reduce the logic of the game to simpler units if possible.

REFERENCES:

- 1) Digital System Design – Charles J Roth
- 2) Altera DE1 board coding the vga www.youtube.com/alterade1