Breakout

This is a basic implementation of the famous game Breakout running on a FPGA, which developed for the final project of HW/SW course at the TEIS AGSTU. The development board used is DE1 with Cyclone V FPGA.

Version	Date	Responsible	Description
1.0.0	2017/03/21	Navid Kheradmand	Description of two previous job and Feasibility study report
1.1.0	2017/04/19	Navid Kheradmand	Final reports with new headlines
1.1.1	2017/05/08	Navid Kheradmand	Qsys is removed, hardware architecture is added and software is described

Navid Kheradmand 2017/04/19 Navidkh@gmail.com
Http://kheradmand.eu

Table of Contents

NAVID KHERADMAND
NAVIDKH@GMAIL.COM
HTTP://KHERADMAND.EU

Tables

NAVID KHERADMAND
NAVIDKH@GMAIL.COM
HTTP://KHERADMAND.EU

NAVID KHERADMAND
NAVIDKH@GMAIL.COM
HTTP://KHERADMAND.EU

1. Specification

Table 1 – Specification

Req	Description	Completed Yes / No
	Constructional Code	
1	Rules and Guidelines for VHDL and C should be followed.	
2	All code should fit on the internal RAM. Except for BeMicro Cart can also be code in external SDRAM.	
3	The card will boot up automatically after power-up. This requirement applies only BeMicrokortet otherwise the award criteria.	
	Documentation and structure of the folders	
4	Delivery to the following folders and files: 1. The folder documentation a. Report (Note filename: fornamn_efternamn_HWSW_eng_B) i. The first page of descriptive title (NOT Engineering Jobs), name, date, email address and brief summary ii. Table of Contents iii. Specification (from specifications A, cv. Modified) iv. Mature projects (from specifications A, recycle) v. Time schedule, the estimated number of hours and the actual number of hours vi. Construction Description HW (overall architecture, each block so forth describe how meta-stability problems are avoided, flow chart or similar to already developed IP components, the size, the estimated power consumption and settings in the tools) vii. Construction Description SW (overall architecture, if necessary. Flowchart, size, settings tools etc.) viii. Verification (test benches with explanation HW) ix. Validation, shown with a movie (link), see next requirement x. Conclusions, analysis, reflections and future enhancements xi. Cost of the project (600 SEK / hour) xii. references xiii. attachments xiv. Delivery format Word or pdf, Swedish or English, it is up to you b. Game Manual for the user, if it's a game 2. The folder design files a. Software b. Quartus projects. Archived projects c. Other IP components, e.g. VGA d. Spel_konsol.sof	

	e. The whole system in a POF file (HW + E) f. Other such second IP components 3. The folder miscellaneous	
5	Certificate Validation and presentation of results; Create a short film (please add it on Youtube), please bring an introduction by HW and SW architecture, show clearly how the system will boot after power on, and show how the game (or anything else) works. One recommendation is to add your name (optional), good advertising. Nice if tag "TEIS" is added, then it will be easy to get all the movies from TEIS training. Note, if you do not want it to be published, write it clearly movie link.	
Delivery Requirements		
6	Delivery must be made to the platform Itslearning, Engineering Job B. The name of the file should be "fornamn_efternamn_HWSW_eng_B.zip" (a file). The last delivery day see rate schedule (VG).	

2. description of two previous jobs

2.1. Space Invaders by Lars Bengtsson

Lars developed a simple version of Classic Space Invaders on DE2-115 card with an external graphic display connected to the GPIO connector. In the game, a monster moves on tops and player's task is to shoot them down. The weapon can be moved to the right and left with KEY0 and KEY2 button. Shots against Monsters fired with KEY1 button and the new game is started with KEY3 button. At the top of the screen displays the number of monster hit.

He uses Sierra to create Tasks for the Games and Display_hw_driver for display the game on the external graphic display [1].

2.2. Rabbit vs. Robots 2 by André Norberg

Andre Developed Rabbit Vs. Robots, The player should move the rabbit to avoid the robots, and try to get them to collide with each other and become statues instead. If a monster collides with a statue, it will also be a statue. The rabbit and the robots will be able to move to each pixel of the court. With the help of an accelerometer, the player should be able to move in 8 directions, horizontally, vertically and diagonally. Once the player has moved two steps, all robots moved a step closer to the player. The game is to clear all the monsters from the game board.

The architecture of this project is planned to include a Nios2 CPU, Sierra RTK and the associated components, such as memory, sysid and JTAG. A proprietary component for managing communications for an I2C bus will be added. Another proprietary component that will be used is the one that allows for a 16-bit binary number to 5 decimal digits displayed on the 7-segment displays. To show the game on a screen is VGA component from a previous task to be included in the project [2].

3. REQUIREMENTS SPECIFICATION

3.1. History

Breakout is an arcade game developed and published by Atari, Inc. It was conceptualized by Nolan Bushnell and Steve Bristow, influenced by the 1972 Atari arcade game Pong, and built by Steve Wozniak aided by Steve Jobs [3].

3.2. Overview

The player controls the paddle, which prevents a ball from falling from the playing field, attempting to bounce it against a number of bricks. The ball striking a brick causes the brick to disappear. When all the bricks are gone, the player will win.

3.3. System requirements

- Game work with DE1-SOC board
- The Game element can be change by User with Nios ii
- The gmae will boot by default value when the device is turn on.
- The game Paddle can be controlled by KEY.
- Start/Pause and restart game button.

3.4. Construction code requirements

- header file for all files,
- Variable names, should be clear and understandable,
- Asynchronous reset "reset_n" for processes
- Code should have clear comments
- Low active signals should end with "_n" in their names
- Name for numbers should end either with "_us" or "s" to show if it is signed or unsigned number
- Processes should be described with a short description in the code
- All out signals should have "X", "1" or "0" after " reset_n" goes high
- All asynchronous signals should should be clocked by two flip flops to be protected from metastability problem.
- Run Design Assistant for construction

4. Time Table

Table 2 - Time Table

Activity	predicted time (Hours)	Time(Hours)
Planning	2	2
Requirment Analysis	8	6
Analysis & Design & Impelementation	30	38
Testing & Evaluation	10	
Documantation	4	7
Total	54	53

Becuase the old project is used Requirment analysis, designing and testing neededd less time. But unfortunatly documantation take more time than it was evaluated in the first place.

5. Construction Description HW

5.1. Overall architecture

Figure 1 shows RTL view of overall architecture of the system.

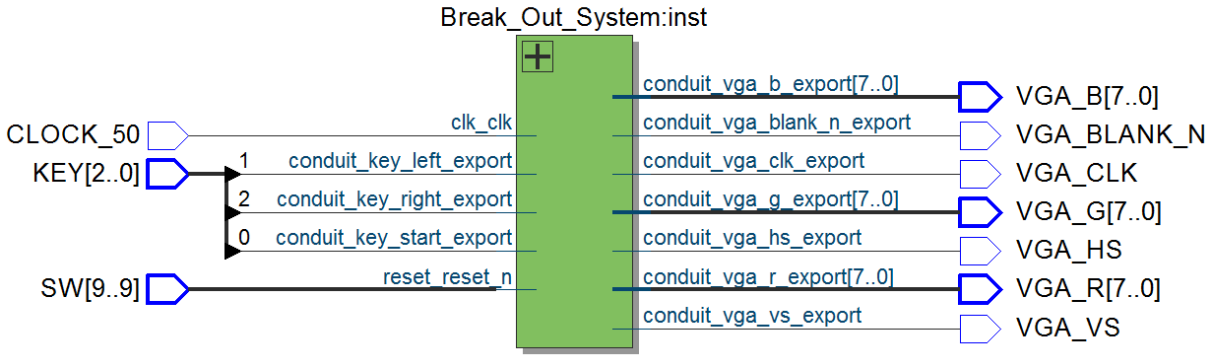


Figure 1 – RTL

Figure 2 shows architecture of hardware system.

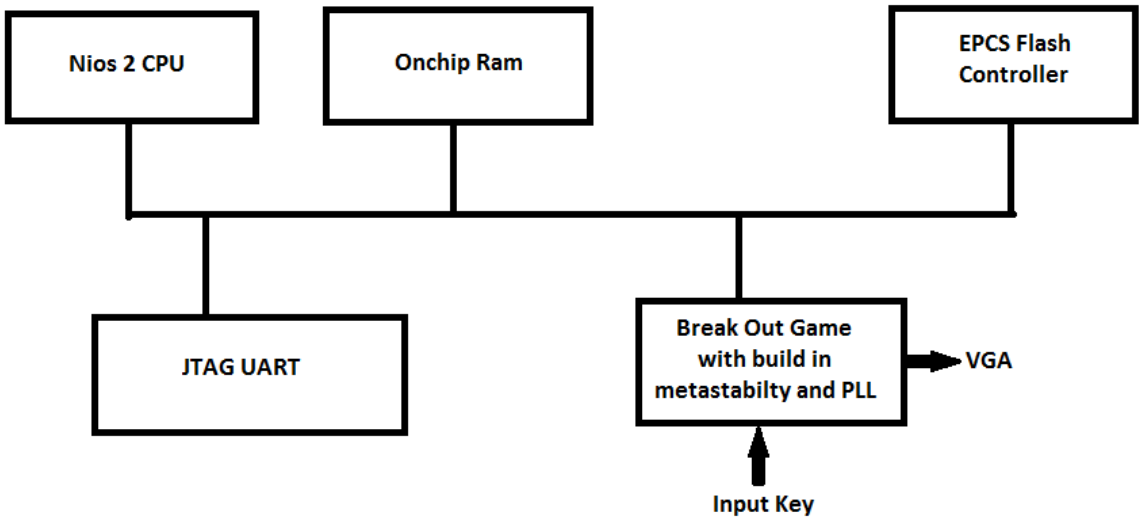


Figure 2 - Hardware Architecture

5.2. Developed IP Components

5.2.1. Break_out_Game

Its the old Engineer job which is developed for VHDL course with some modification. The project has been changed to be used as IP component.Figure 3 shows component RTL. For More Information see Break_out_Game Documentation in attachment.

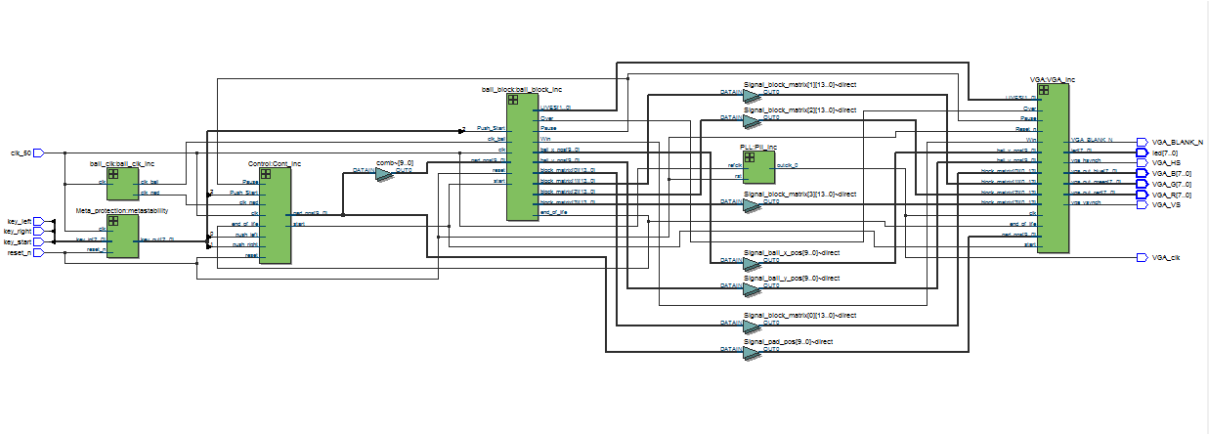


Figure 3 - Breakout RTL veiw

5.2.1.1. Package Logic

Logic is a package which contain data for desiging the game. Table 3 Below shows parameter, which can be change in game.

Table 3 - Package Logic

Parameter	Description
MAX_LIVES	change these constants to define the number of Lives
BLOCK_COLS	change these constants to define the number of columns of blocks
BLOCK_ROWS	change these constants to define the number of rows of blocks
BLOCK_WIDTH	change these constants to define the width size of a block
BLOCK_HEIGHT	change these constants to define the height size of a block
FIELD_BORDER	change these constants to define how many pixels from the edge of the screen the playing field starts
PAD_LENGTH	change these constants to define the length (width) of the pad
BALL_RADIUS	change these constants to define the radius of the ball

5.2.1.2. PLL

VGA system is powered by a 25MHz clock. This creates a so-called PLL (phase locked loop). A PLL can increase or decrease the frequency and phase shift kockan. In this case, Refrence clock is a 50MHz clock(Figure 4).

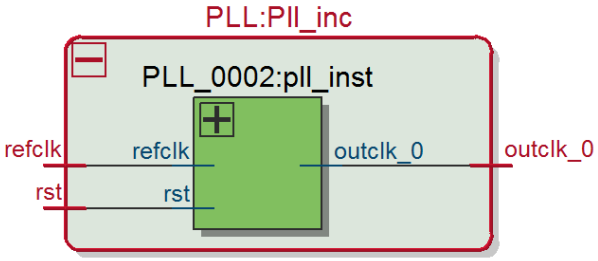


Figure 4 - PLL RTL

5.2.1.2.1. Inputs

Refclk,rst

5.2.1.2.2. Outputs

Outclk_0

5.2.1.3. Meta_Protection

In multiple clock system on chip design (SOC), where the signal transfers from one clock domain to another clock domain, a synchronizer is needed which acquires the asynchronous signal and offers synchronize signal. Most of the time, synchronizer suffers from the effect of metastability. There are many causes for the occurrence of metastability which includes fluctuations in data signal within setup and hold time window, low-temperature condition, low voltage, multiple changes in input, strength of input signal and clock skew and clock pulse width etc. Change in data signal within setup and hold time window are uncontrollable &unpredictable while other causes can be controlled.The signal from the button and switches metastability secured on this component by two D flip-flops (Figure 5 and Figure 6).

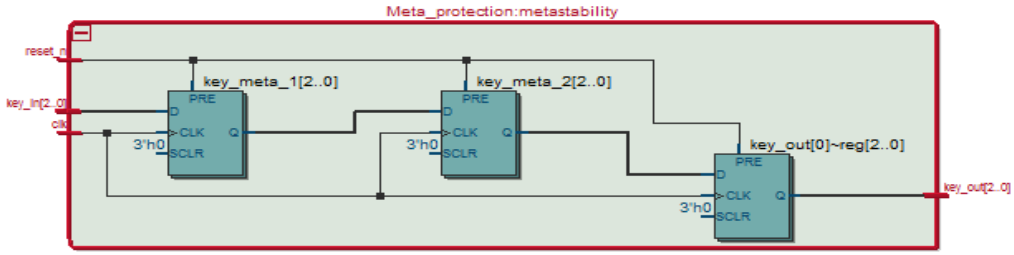


Figure 5 - Meta_Protection RTL

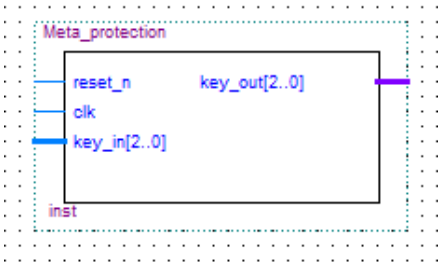


Figure 6 - Meta Protection

5.2.1.3.1. Inputs

Reset_n,Clk,Key_in [0..2]

5.2.1.3.2. Outputs

Key_out[2..0]

5.2.1.4. Font

This is a package which is used to defeind Char Matrix for writing on the screen. There is two type of font 12x12 and 24x24. 12x12 is used to write designer information on top of the screen and 24x24 font used to write instraction on the screen.

5.2.1.5. Ball_Block

Controlling the ball movment and pridicting its next ball behaviar is the main job of this component. It also manage the Block matrix. Hitting the border of the game will change the ball direction as same as when the ball hit the paddle or block. When the ball hit the block this componet will remove that block by putting 0 in block matrix.when the ball pass the paddle program will the remaining Lives if there is more lifes it will remove one of them and change the Pause data to 0 and put the ball in

default possition. If there isn't more life it will change over and puase data to 0 and if all the block disappear it will change Win and pause data to 0.Lives, Pause,Over,Win,Block_matrix and ball pos will send to other component to draw and control the game (Figure 7 and Figure 8).

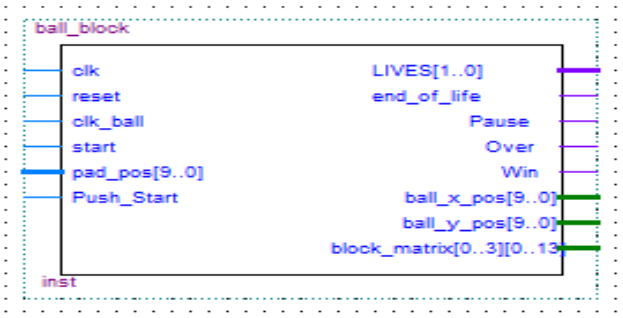


Figure 7 - Ball_Block

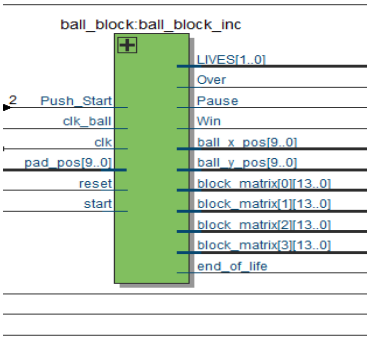


Figure 8 - Ball_Block RTL

5.2.1.5.1. Inputs

Clk, Reset, Pad_pos, Clk_ball,Push_start and start

5.2.1.5.2. Outputs

Lives, Over, Pause, Win, Block_Matrix, Ball_pos, End_of_life

5.2.1.6. Ball_Clock

Relevent timing bitween Paddle and Ball which helps to makes movement of ball and paddle more smooth and related to each other(Figure 9 and Figure 10).

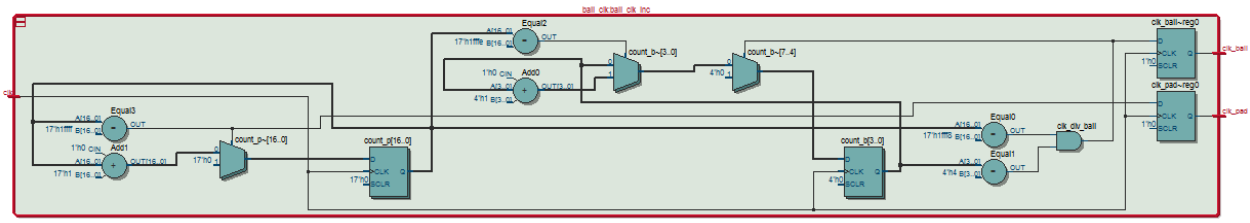


Figure 9 - Ball Clock RTL

Figure 10 - Ball_clock

5.2.1.6.1. Inputs

Clk

5.2.1.6.2. Outputs

Clk_ball, Clk_pad

5.2.1.7. Control

Three pushed key used in other to control the game, KEY0 is used to move paddle to right , KEY3 is used to move paddle to left and KEY1 is used as Control button which is pause the game when its not puase or start the game when its puase. This Component also base on the data from Pause and end_of_life will pause the game.movment of paddle and its possion is the main duty of Control component (see Figure 11and Figure 12).

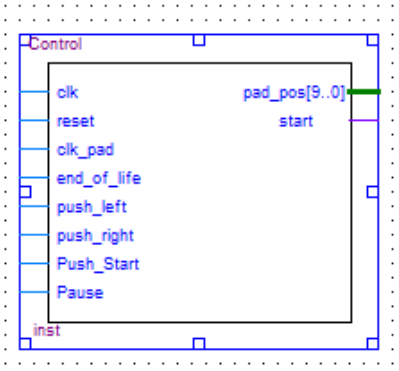


Figure 11 – Control

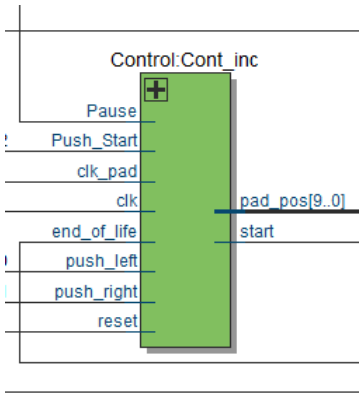


Figure 12 - Control RTL

5.2.1.7.1. Inputs

Pause, Push_start, Clk_pad,Clk,end_of_life, Push_left, Push_right, Reset

5.2.1.7.2. Outputs

Pad_pos, Start

5.2.1.8. VGA

VGA component contain information about drawing on Screen through the VGA Protocol. VGA component have resulation of 640X480 with 25MHz clock. The Component base on information recieve from other Component draw paddle, ball, blocks, Lives and information on the screen. It can write and read the data on screen and controls all sync of VGA signals (Figure 13 Figure 14).

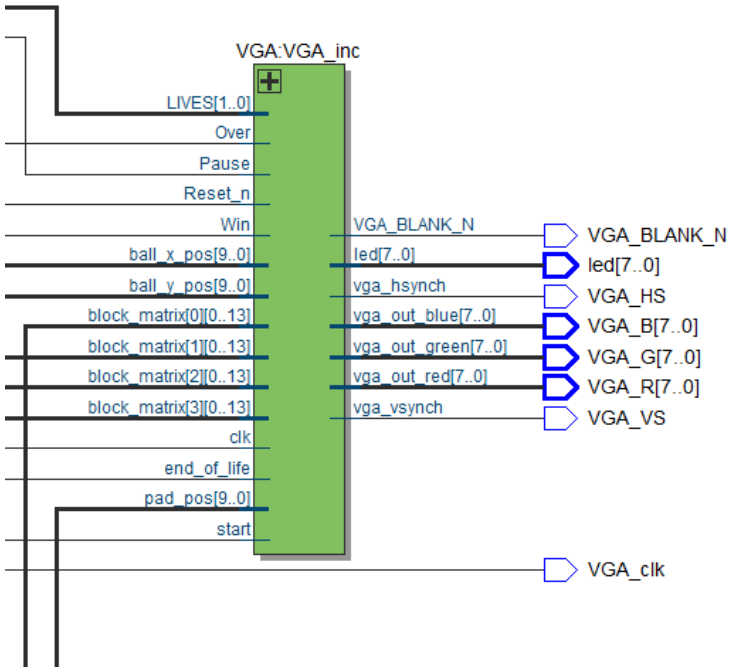


Figure 13 - VGA RTL

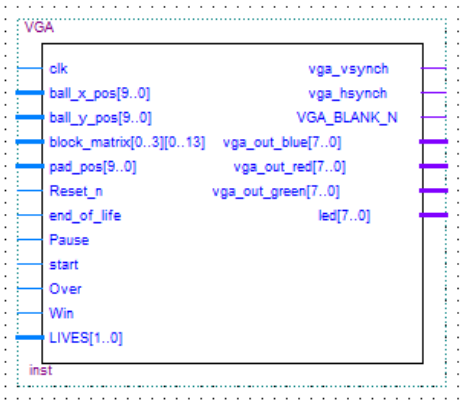


Figure 14 – VGA

5.2.1.8.1. Inputs

Clk,Ball_Pos, Block_Matrix, Pad_Pos, Reset_N,End_of_life, Puase, Start, Over, Win, Lives

5.2.1.8.2. Outputs

VGA_Vsynch, VGA_Hsynch,VGA_Blank_N,VGA_out_Blue,Vga_Out_Green,LED

5.3. Memory Map

Memory map can be seen in Figure 15.

System: Break_Out_System **Path:** epcs_flash_controller_0

	nios2_CPU.data_master	nios2_CPU.instruction_master
Break_out_Game_0.avalon_slave_0	0x0810_0000 - 0x081f_ffff	0x0810_0000 - 0x081f_ffff
epcs_flash_controller_0.epcs_control_...	0x0000_0000 - 0x0000_07ff	0x0000_0000 - 0x0000_07ff
jtag_uart.avalon_jtag_slave	0x0820_9008 - 0x0820_900f	0x0820_9008 - 0x0820_900f
nios2_CPU.debug_mem_slave	0x0820_8800 - 0x0820_8fff	0x0820_8800 - 0x0820_8fff
onchip_ram.s1	0x0820_4000 - 0x0820_7fff	0x0820_4000 - 0x0820_7fff

Figure 15 - Memory Map

5.4. Power Consumption

Power Analysor result can be seen in Figure 16.

PowerPlay Power Analyzer Summary

PowerPlay Power Analyzer Status	Successful - Wed Apr 19 12:25:22 2017
Quartus Prime Version	16.0.0 Build 211 04/27/2016 SJ Lite Edition
Revision Name	Engineer_Job
Top-level Entity Name	Engineer_Job
Family	Cyclone V
Device	5CSEMA5F31C6
Power Models	Final
Total Thermal Power Dissipation	425.44 mW
Core Dynamic Thermal Power Dissipation	0.81 mW
Core Static Thermal Power Dissipation	415.25 mW
I/O Thermal Power Dissipation	9.38 mW
Power Estimation Confidence	Low: user provided insufficient toggle rate data

Figure 16 - Power Consumption

5.5. Tools Setting

5.5.1. I/O Pins

Pins used for DEC-Soc board can be seen in Figure 17.

CLOCK_50	Input	PIN_AF14	3B	B3B_N0	PIN_AF14	2.5 V	12mA (default)		
KEY[2]	Input	PIN_W15	3B	B3B_N0	PIN_W15	2.5 V	12mA (default)		
KEY[1]	Input	PIN_AA15	3B	B3B_N0	PIN_AA15	2.5 V	12mA (default)		
KEY[0]	Input	PIN_AA14	3B	B3B_N0	PIN_AA14	2.5 V	12mA (default)		
SW[9]	Input	PIN_AE12	3A	B3A_N0	PIN_AE12	2.5 V	12mA (default)		
VGA_B[7]	Output	PIN_J14	8A	B8A_N0	PIN_J14	2.5 V	12mA (default)	1 (default)	
VGA_B[6]	Output	PIN_G15	8A	B8A_N0	PIN_G15	2.5 V	12mA (default)	1 (default)	
VGA_B[5]	Output	PIN_F15	8A	B8A_N0	PIN_F15	2.5 V	12mA (default)	1 (default)	
VGA_B[4]	Output	PIN_H14	8A	B8A_N0	PIN_H14	2.5 V	12mA (default)	1 (default)	
VGA_B[3]	Output	PIN_F14	8A	B8A_N0	PIN_F14	2.5 V	12mA (default)	1 (default)	
VGA_B[2]	Output	PIN_H13	8A	B8A_N0	PIN_H13	2.5 V	12mA (default)	1 (default)	
VGA_B[1]	Output	PIN_G13	8A	B8A_N0	PIN_G13	2.5 V	12mA (default)	1 (default)	
VGA_B[0]	Output	PIN_B13	8A	B8A_N0	PIN_B13	2.5 V	12mA (default)	1 (default)	
VGA_BLANK_N	Output	PIN_F10	8A	B8A_N0	PIN_F10	2.5 V	12mA (default)	1 (default)	
VGA_CLK	Output	PIN_A11	8A	B8A_N0	PIN_A11	2.5 V	12mA (default)	1 (default)	
VGA_G[7]	Output	PIN_E11	8A	B8A_N0	PIN_E11	2.5 V	12mA (default)	1 (default)	
VGA_G[6]	Output	PIN_F11	8A	B8A_N0	PIN_F11	2.5 V	12mA (default)	1 (default)	
VGA_G[5]	Output	PIN_G12	8A	B8A_N0	PIN_G12	2.5 V	12mA (default)	1 (default)	
VGA_G[4]	Output	PIN_G11	8A	B8A_N0	PIN_G11	2.5 V	12mA (default)	1 (default)	
VGA_G[3]	Output	PIN_G10	8A	B8A_N0	PIN_G10	2.5 V	12mA (default)	1 (default)	
VGA_G[2]	Output	PIN_H12	8A	B8A_N0	PIN_H12	2.5 V	12mA (default)	1 (default)	
VGA_G[1]	Output	PIN_J10	8A	B8A_N0	PIN_J10	2.5 V	12mA (default)	1 (default)	
VGA_G[0]	Output	PIN_J9	8A	B8A_N0	PIN_J9	2.5 V	12mA (default)	1 (default)	
VGA_HS	Output	PIN_B11	8A	B8A_N0	PIN_B11	2.5 V	12mA (default)	1 (default)	
VGA_R[7]	Output	PIN_F13	8A	B8A_N0	PIN_F13	2.5 V	12mA (default)	1 (default)	
VGA_R[6]	Output	PIN_E12	8A	B8A_N0	PIN_E12	2.5 V	12mA (default)	1 (default)	
VGA_R[5]	Output	PIN_D12	8A	B8A_N0	PIN_D12	2.5 V	12mA (default)	1 (default)	
VGA_R[4]	Output	PIN_C12	8A	B8A_N0	PIN_C12	2.5 V	12mA (default)	1 (default)	
VGA_R[3]	Output	PIN_B12	8A	B8A_N0	PIN_B12	2.5 V	12mA (default)	1 (default)	
VGA_R[2]	Output	PIN_E13	8A	B8A_N0	PIN_E13	2.5 V	12mA (default)	1 (default)	
VGA_R[1]	Output	PIN_C13	8A	B8A_N0	PIN_C13	2.5 V	12mA (default)	1 (default)	
VGA_R[0]	Output	PIN_A13	8A	B8A_N0	PIN_A13	2.5 V	12mA (default)	1 (default)	
VGA_VS	Output	PIN_D11	8A	B8A_N0	PIN_D11	2.5 V	12mA (default)	1 (default)	

Figure 17 I/O Pins

5.5.2. Device Option

Device use Active serial port. Device setting can be seen in Figure 18

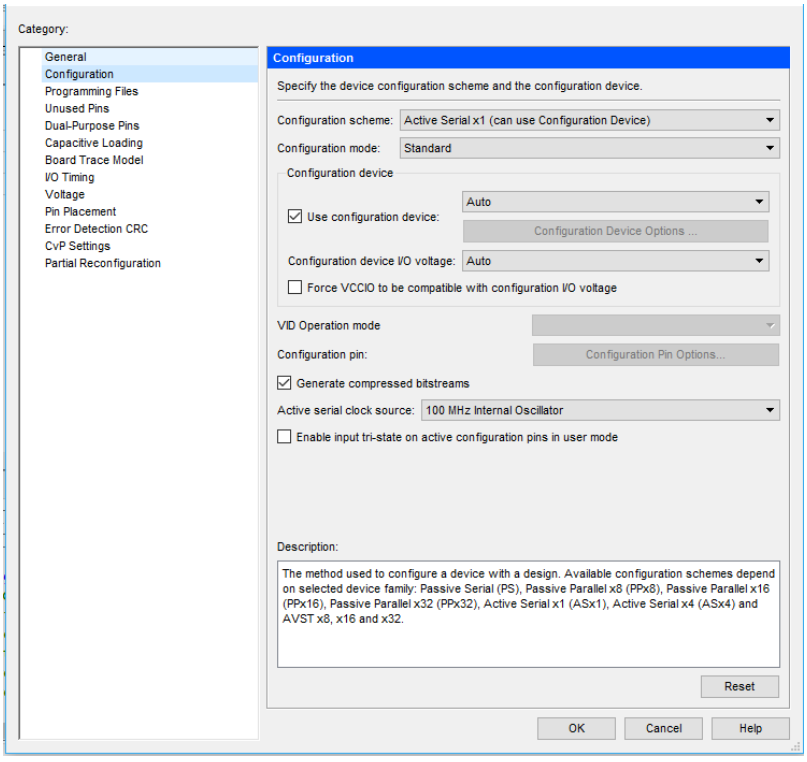


Figure 18 - HW device option

6. Construction Description SW

6.1. Overall

Architecture of SW is simple. It has one driver, which has to be add in the system, Break_Out_Game. Using driver is simple because for now it has only to function GAME_START and GAME_STOP. For more information, see Break out Game User Guide in attachment.

When the developer Start the game the software will send a '1' signal through the Jtag-Uart to the breakout game component and the game will draw on the screen. When the developer Stop the game '0' signal will be send to the breakout game component and it would remove the screen and restart the game.

6.2. Size

Code size can be seen in Figure 19.

```
Navid@Navid-PC /cygdrive/c/Users/Navid/Documents/VHDL/Home_work/HWSW/Engineer_Job/software/SW_Engineer_Job
$ nios2-elf-size SW_Engineer_Job.elf
   text    data     bss     dec     hex filename
   4348     724      16    5088    13e0 SW_Engineer_Job.elf
```

Figure 19 - SW Size

6.3. Tools Settings

Setting used is software can be seen in Figure 20

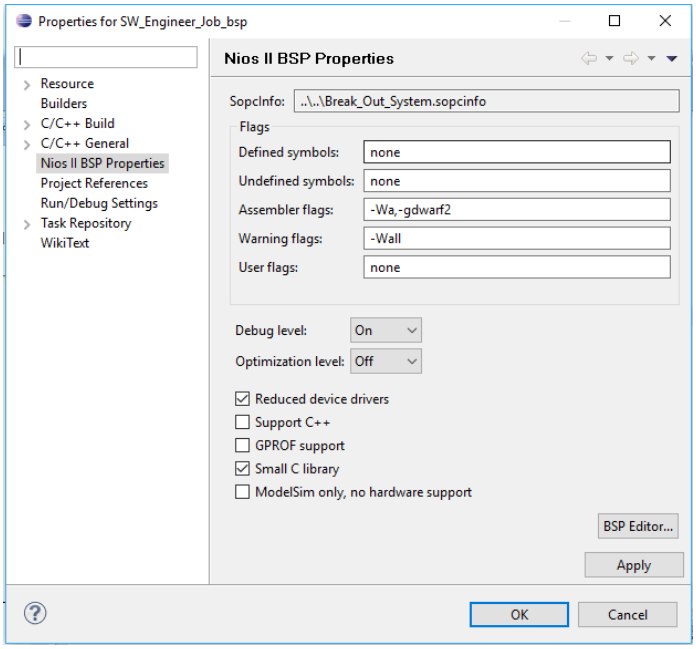


Figure 20 - SW Setting

7. Verification

7.1. Test Protocol

Test Protocol for Validation can be seen in Table 4:

Table 4 - Test Protocol For Validation

Test Case	Description	Result	OK / NOT OK
1	Turn on FPGA	The HW Load	OK see Video
2	GAME_START	The game load	OK see Video
3	Press Start key	Ball Move	OK see Video
4	Press Left Key	Paddle goes left	OK see Video
5	Press Right Key	Paddle goes right	OK see Video
6	Press Start Key	Pause the Game	OK see Video

7.2. Model Sim

The result from the last project is used here.

7.2.1. Type1

The game verified by Model sim Type 1 (see Figure 21- Figure 23)

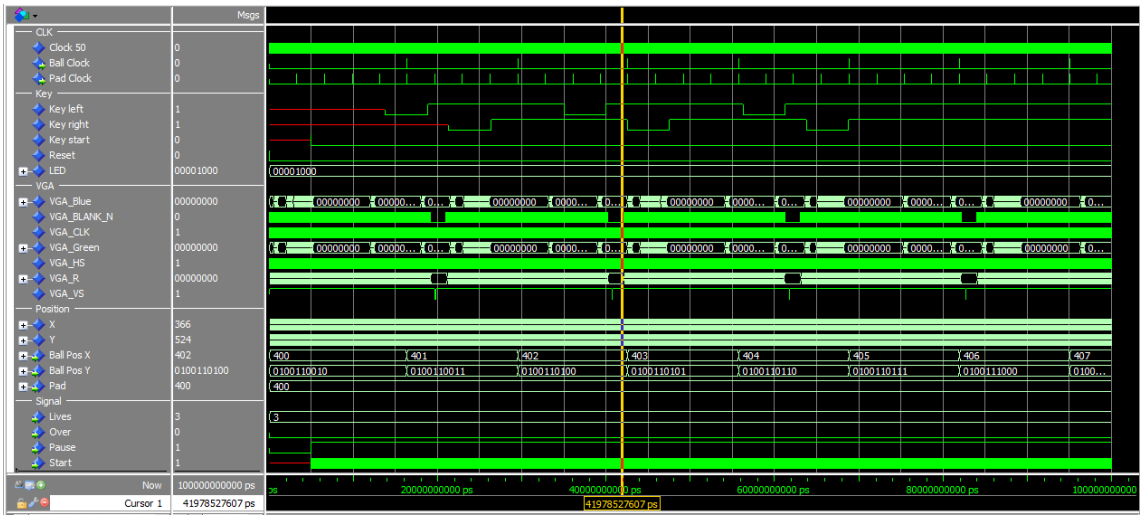


Figure 21 - Test Bench Type 1

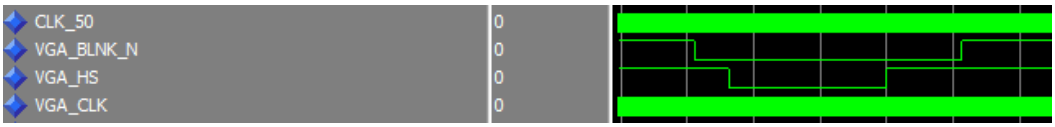


Figure 22 - Test bench VGA

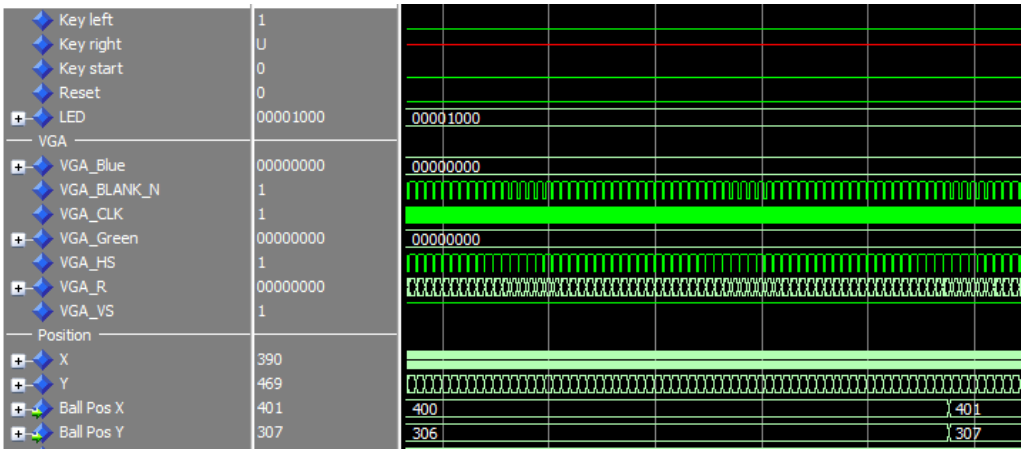


Figure 23 - Test Bench Moving ball

7.2.2. *Type 2*

The game verified by Model sim Type 1. (See Figure 24)

```
# Info: ================================================
# Info:           Generic PLL Summary
# Info: ================================================
# Time scale of (engineer_job_vhd_tst.i1.Pll_inc.pll_altera_pll_altera_pll_i_639.new_model.gpll.no_need_to_gen) is  1ps /  1ps
# Info: hierarchical_name = engineer_job_vhd_tst.i1.Pll_inc.pll_altera_pll_altera_pll_i_639.new_model.gpll.no_need_to_gen
# Info: reference_clock_frequency = 50.0 MHz
# Info: output_clock_frequency = 25 MHZ
# Info: phase_shift = 0 ps
# Info: duty_cycle = 50
# Info: sim_additional_refclk_cycles_to_lock = 0
# Info: output_clock_high_period = 20000.000000
# Info: output_clock_low_period = 20000.000000
# ** Note: TEST 1: VGA_HS falling Start
#    Time: 33262500 ps  Iteration: 7  Instance: /engineer_job_vhd_tst
# ** Note: TEST 1: OK VGA_HS falling
#    Time: 33262500 ps  Iteration: 7  Instance: /engineer_job_vhd_tst
# ** Note: TEST 2: VGA_HS rising Start
#    Time: 38012500 ps  Iteration: 7  Instance: /engineer_job_vhd_tst
# ** Note: TEST 2: OK VGA_HS rising
#    Time: 38012500 ps  Iteration: 7  Instance: /engineer_job_vhd_tst
# ** Note: TEST 3: VGA_BLANK_N rising X Start
#    Time: 40262500 ps  Iteration: 7  Instance: /engineer_job_vhd_tst
# ** Note: TEST 3: OK VGA_BLANK_N X rising
#    Time: 40262500 ps  Iteration: 7  Instance: /engineer_job_vhd_tst
# ** Note: TEST 4: VGA_BLANK_N Y rising Start
#    Time: 72212500 ps  Iteration: 7  Instance: /engineer_job_vhd_tst
# ** Note: TEST 4: OK VGA_BLANK_N Y rising
#    Time: 72212500 ps  Iteration: 7  Instance: /engineer_job_vhd_tst
# ** Note: TEST 5: VGA_VS falling Start
#    Time: 19715662500 ps  Iteration: 7  Instance: /engineer_job_vhd_tst
# ** Note: TEST 5: OK VGA_VS falling
#    Time: 19715662500 ps  Iteration: 7  Instance: /engineer_job_vhd_tst
# ** Note: TEST 6: VGA_VS rising Start
#    Time: 19755662500 ps  Iteration: 7  Instance: /engineer_job_vhd_tst
# ** Note: TEST 6: OK VGA_VS rising
#    Time: 19755662500 ps  Iteration: 7  Instance: /engineer_job_vhd_tst
```

Figure 24 - Test Bench Type 2

8. Validation

8.1. Video

Link Below shows the video of how the game work.

https://www.youtube.com/watch?v=HphfG98GtbE

9. Analysis

9.1. Warnings

Warning can be seen in Figure 25.

```
18236 Number of processors has not been specified which may cause overloading on shared machines.  Set the global assignment NUM_PARALLEL_PROCESSC
10037 Verilog HDL or VHDL warning at Break_Out_System_epcs_flash_controller_0.v(401): conditional expression evaluates to a constant
10541 VHDL Signal Declaration warning at Break_out.vhd(33): used implicit default value for signal "RD" because signal was never assigned a value
10036 Verilog HDL or VHDL warning at Break_out.vhd(72): object "addr_to_ram" assigned a value but never read
20013 Ignored 24 assignments for entity "DE1_SOC_golden_top" -- entity does not exist in design
10933 Combinational logic "Break_Out_System:inst|Break_out_top:break_out_game_0|VGA:VGA_inc|vga_out_red" depth is over 6000, which may cause stack
20013 Ignored 24 assignments for entity "DE1_SOC_golden_top" -- entity does not exist in design
12241 5 hierarchies have connectivity warnings - see the Connectivity Checks report folder
20013 Ignored 24 assignments for entity "DE1_SOC_golden_top" -- entity does not exist in design
      RST port on the PLL is not properly connected on instance Break_Out_System:inst|Break_out_top:break_out_game_0|PLL:Pll_inc|altpll:altpll_com
18236 Number of processors has not been specified which may cause overloading on shared machines.  Set the global assignment NUM_PARALLEL_PROCESSC
      RST port on the PLL is not properly connected on instance Break_Out_System:inst|Break_out_top:break_out_game_0|PLL:Pll_inc|altpll:altpll_com
21300 LOCKED port on the PLL is not properly connected on instance "Break_Out_System:inst|Break_out_top:break_out_game_0|PLL:Pll_inc|altpll:altpll
292013 Feature LogicLock is only available with a valid subscription license. You can purchase a software subscription to gain full access to this
15714 Some pins have incomplete I/O assignments. Refer to the I/O Assignment Warnings report for details
177007 PLL(s) placed in location FRACTIONALPLL_X0_Y15_N0 do not have a PLL clock to compensate specified - the Fitter will attempt to compensate al
335093 TimeQuest Timing Analyzer is analyzing 1 combinational loops as latches. For more details, run the Check Timing command in the TimeQuest Tim
332060 Node: CLOCK_50 was determined to be a clock but was found without an associated clock assignment.
332060 Node: Break_Out_System:inst|Break_Out_System_jtag_uart:jtag_uart|alt_jtag_atlantic:Break_Out_System_jtag_uart_alt_jtag_atlantic|rst1 was det
332056 PLL cross checking found inconsistent PLL clock settings:
15705 Ignored locations or region assignments to the following nodes
171167 Found invalid Fitter assignments. See the Ignored Assignments panel in the Fitter Compilation Report for more information.
18236 Number of processors has not been specified which may cause overloading on shared machines.  Set the global assignment NUM_PARALLEL_PROCESSC
20013 Ignored 24 assignments for entity "DE1_SOC_golden_top" -- entity does not exist in design
18236 Number of processors has not been specified which may cause overloading on shared machines.  Set the global assignment NUM_PARALLEL_PROCESSC
335093 TimeQuest Timing Analyzer is analyzing 1 combinational loops as latches. For more details, run the Check Timing command in the TimeQuest Tim
332060 Node: CLOCK_50 was determined to be a clock but was found without an associated clock assignment.
332060 Node: Break_Out_System:inst|Break_Out_System_jtag_uart:jtag_uart|alt_jtag_atlantic:Break_Out_System_jtag_uart_alt_jtag_atlantic|rst1 was det
332056 PLL cross checking found inconsistent PLL clock settings:
332060 Node: CLOCK_50 was determined to be a clock but was found without an associated clock assignment.
332060 Node: Break_Out_System:inst|Break_Out_System_jtag_uart:jtag_uart|alt_jtag_atlantic:Break_Out_System_jtag_uart_alt_jtag_atlantic|rst1 was det
332056 PLL cross checking found inconsistent PLL clock settings:
332060 Node: CLOCK_50 was determined to be a clock but was found without an associated clock assignment.
332060 Node: Break_Out_System:inst|Break_Out_System_jtag_uart:jtag_uart|alt_jtag_atlantic:Break_Out_System_jtag_uart_alt_jtag_atlantic|rst1 was det
332056 PLL cross checking found inconsistent PLL clock settings:
```

Figure 25 – Warning

9.2. Quartus reports

Quartus reports can be seen in **Error! Reference source not found.** to Figure 28.

9.2.1. *Flow summary*

Fitter Summary	
Fitter Status	Successful - Thu Apr 20 14:24:06 2017
Quartus Prime Version	16.0.0 Build 211 04/27/2016 SJ Lite Edition
Revision Name	Engineer_Job
Top-level Entity Name	Engineer_Job
Family	Cyclone V
Device	5CSEMA5F31C6
Timing Models	Final
Logic utilization (in ALMs)	3,058 / 32,070 (10 %)
Total registers	1380
Total pins	33 / 457 (7 %)
Total virtual pins	0
Total block memory bits	150,016 / 4,065,280 (4 %)
Total RAM Blocks	22 / 397 (6 %)
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	1 / 6 (17 %)
Total DLLs	0 / 4 (0 %)

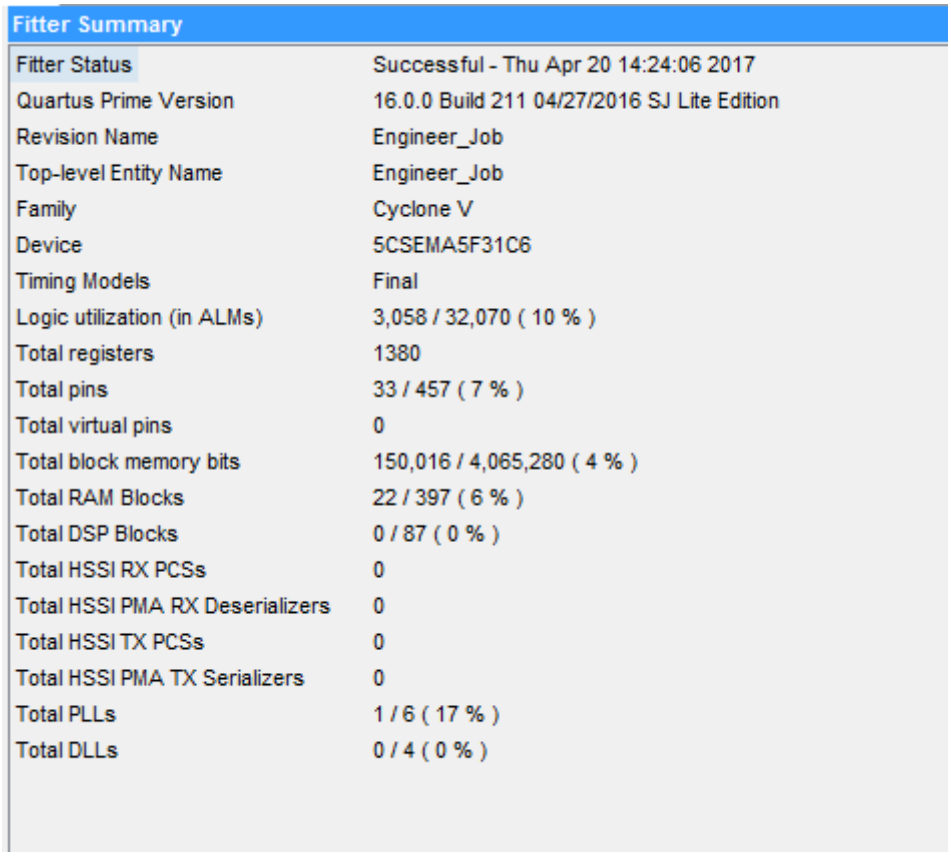
Figure 26 - Flow Summary

9.2.2. *PowerPlay Power Analyzer*

PowerPlay Power Analyzer Summary	
PowerPlay Power Analyzer Status	Successful - Wed Apr 19 12:25:22 2017
Quartus Prime Version	16.0.0 Build 211 04/27/2016 SJ Lite Edition
Revision Name	Engineer_Job
Top-level Entity Name	Engineer_Job
Family	Cyclone V
Device	5CSEMA5F31C6
Power Models	Final
Total Thermal Power Dissipation	425.44 mW
Core Dynamic Thermal Power Dissipation	0.81 mW
Core Static Thermal Power Dissipation	415.25 mW
I/O Thermal Power Dissipation	9.38 mW
Power Estimation Confidence	Low: user provided insufficient toggle rate data

Figure 27 - Power Play Analyzer

9.2.3. *Fitter Summary*



Fitter Summary	
Fitter Status	Successful - Thu Apr 20 14:24:06 2017
Quartus Prime Version	16.0.0 Build 211 04/27/2016 SJ Lite Edition
Revision Name	Engineer_Job
Top-level Entity Name	Engineer_Job
Family	Cyclone V
Device	5CSEMA5F31C6
Timing Models	Final
Logic utilization (in ALMs)	3,058 / 32,070 (10 %)
Total registers	1380
Total pins	33 / 457 (7 %)
Total virtual pins	0
Total block memory bits	150,016 / 4,065,280 (4 %)
Total RAM Blocks	22 / 397 (6 %)
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	1 / 6 (17 %)
Total DLLs	0 / 4 (0 %)

Figure 28 - Fitter Summary

10. Future Enhancements

In future by upgrading Break out game component, Low level C programmer have the opportunity to make the game himself. For example, programmer can define how many lives player can have or how many column and Row should block have or even can change the size of the ball, Paddle and color of the elements in the game.

This project was the first version and game can be improve in future.

11. Cost of the project

Project took totally 53 hour therefore Cost of the project would be 53 * 600 = 31800 SEK.

12. References

[1] L. Bengtsson, "Space Invaders," TEIS AB, 2015.

[2] A. Norberg, "Rabbit vs Robots 2," TEIS AB, 2015.

[3] K. Orland, "Obituary: Gaming pioneer Steve Bristow helped design Tank, Breakout," Ars Technical, 2015.

[4] Altera, "Nios II Gen2 Processor Reference Guide," 2016.

[5] Altera, "Embedded Peripherals IP User Guide," 2016.

[6] Altra, "SOPC Builder User Guide," 2010.