Design and Implementation (in Verilog) of Pong Game

Vladimir Vakhter

ECE Department, WPI

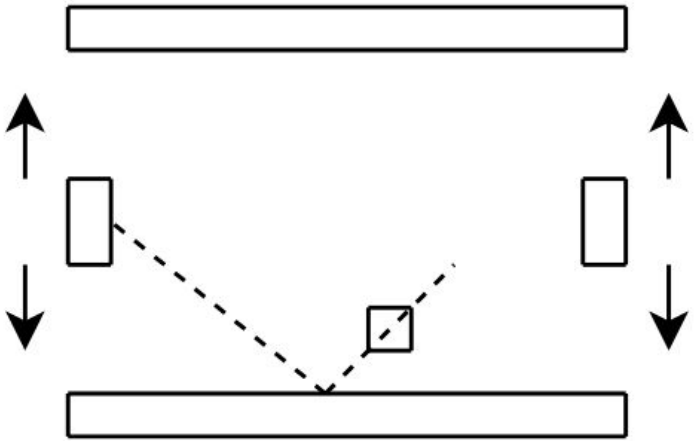
December 5, 2019

Outline

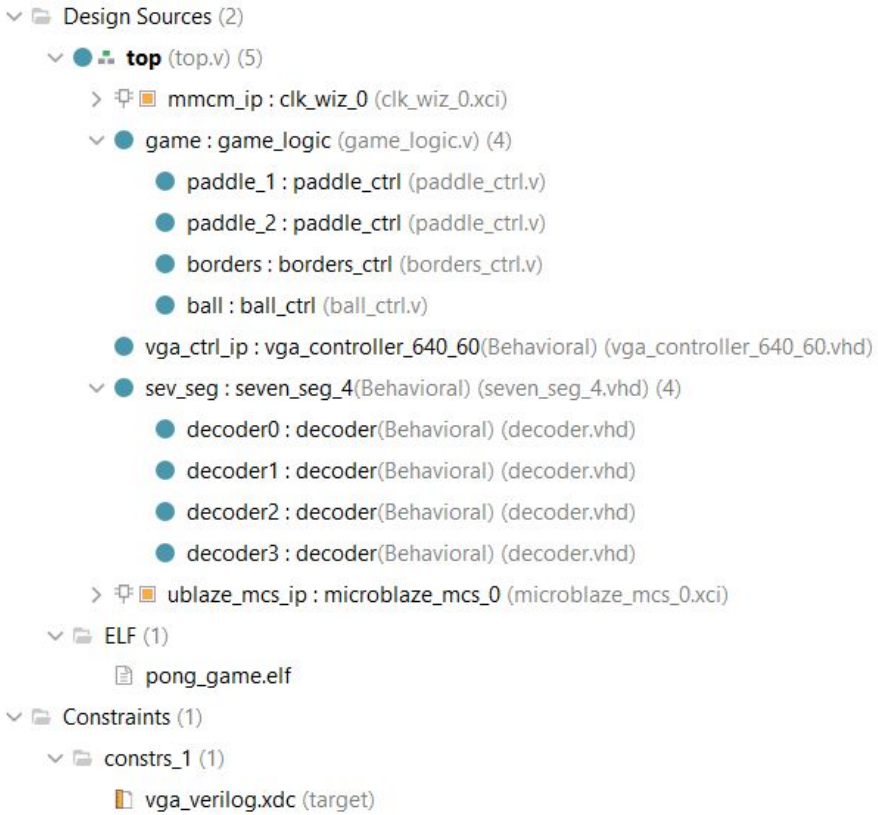
- ❏ Purpose.
- ❏ Hierarchy of Modules.
- ❏ Block Diagram.
- ❏ IP Cores.
- ❏ Custom Hardware Modules.
- ❏ FPGA resource usage.
- ❏ Video of the Implementation of the Design.

Purpose

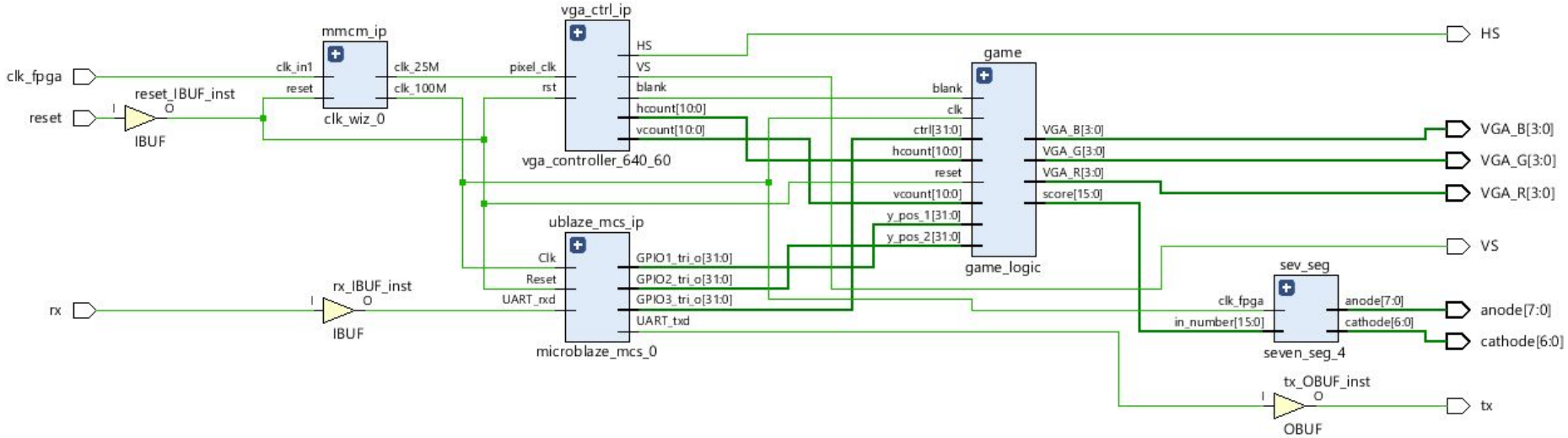
- To design Pong Game in Verilog:
  - the 2-player mode.



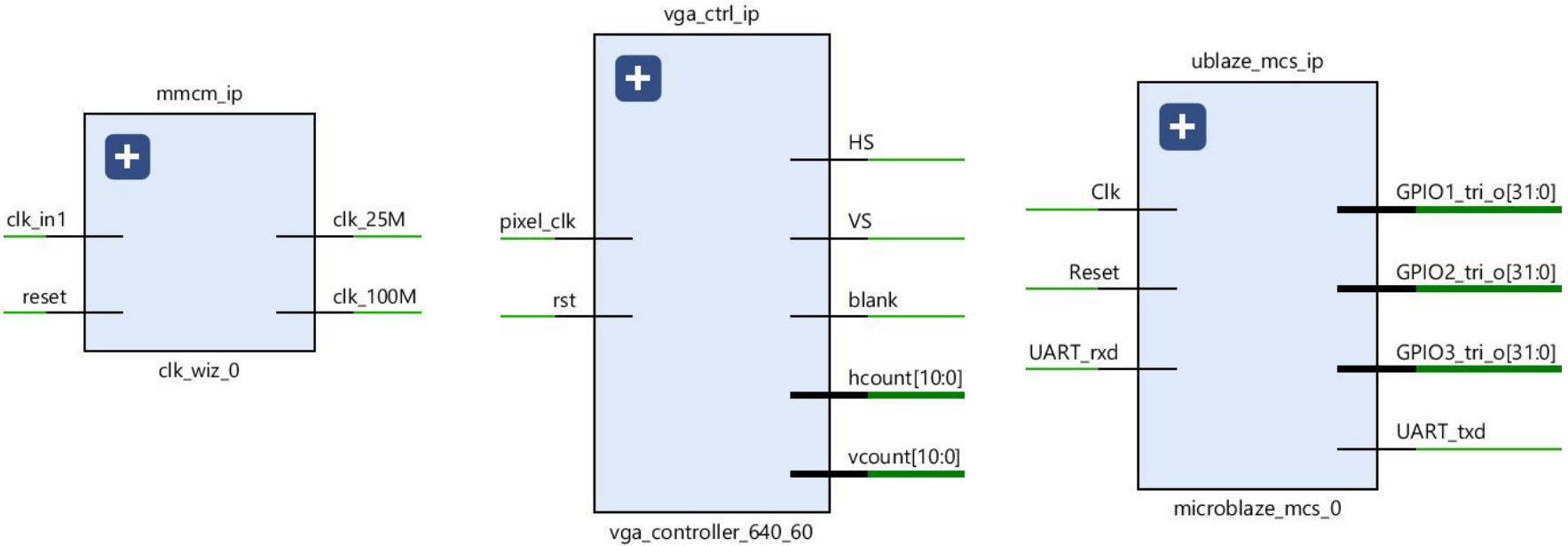
Hierarchy of Modules



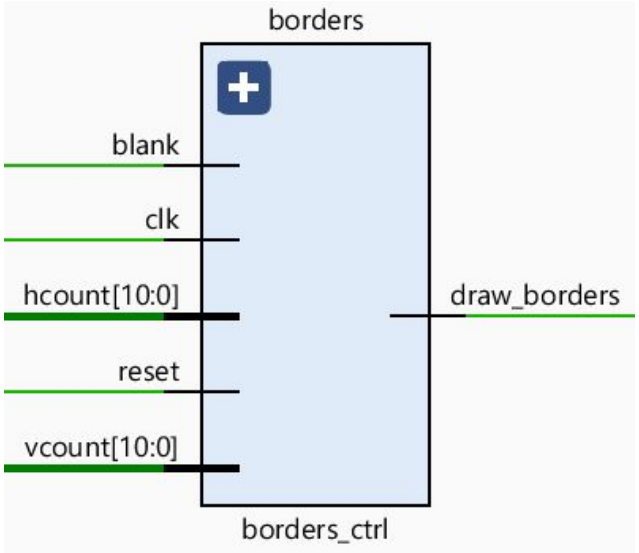
Block Diagram: top.v



IP Cores: Clock Wizard, VGA Controller, MicroBlaze

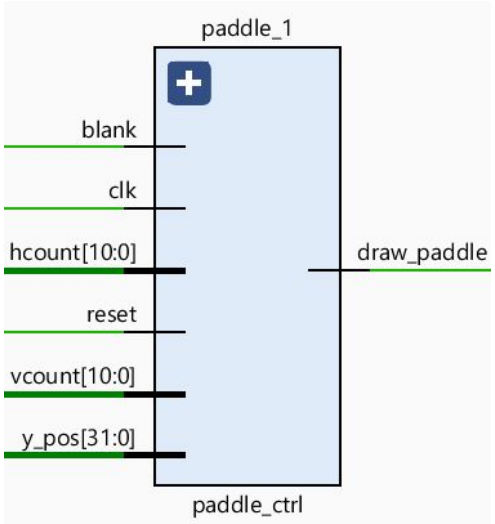


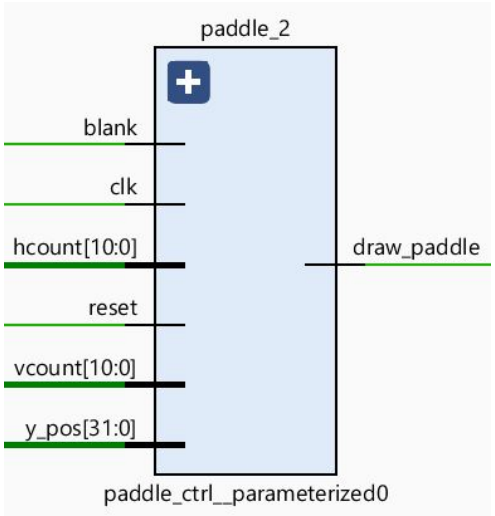
Custom Modules: borders_ctrl.v




```
module borders_ctrl
    #(//dimensions of the game field
        parameter BORDER_WIDTH   = 4'd10,
        parameter X_LEFT_BORDER  = 10'd19,
        parameter X_RIGHT_BORDER = 10'd620,
        parameter Y_UP_BORDER    = 10'd19,
        parameter Y_DOWN_BORDER  = 10'd460)
    (
        input clk,
        input reset,
        input[10:0] hcount,
        input[10:0] vcount,
        input blank,
        output reg draw_borders
    );
```

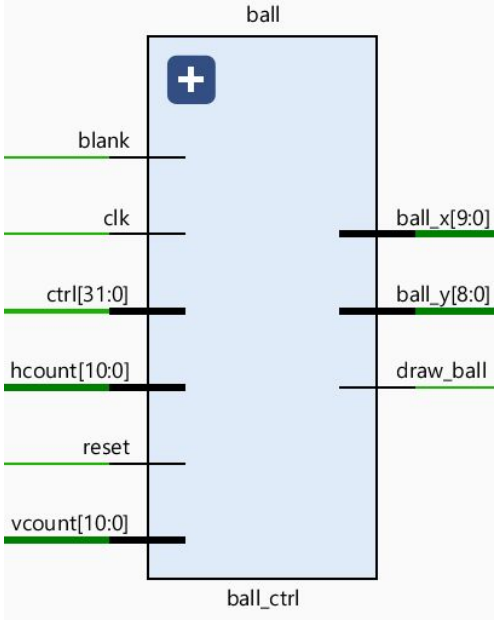
Custom Modules: paddle_ctrl.v





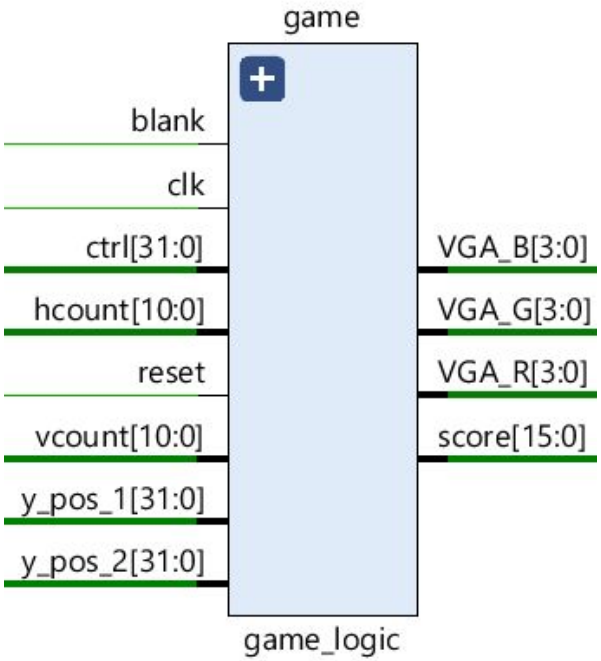

```
module paddle_ctrl
    #(//the upper left corner of the paddle in pixels
      parameter PADDLE_X = 10'd616,
      //paddle's width and height in pixels
      parameter PADDLE_WIDTH  = 10'd5,
      parameter PADDLE_HEIGHT = 10'd48)
    (
        input clk,
        input reset,
        input[10:0] hcount,
        input[10:0] vcount,
        input blank,
        input[31:0] y_pos,
        output reg draw_paddle
    );
```

Custom Modules: ball_ctrl.v



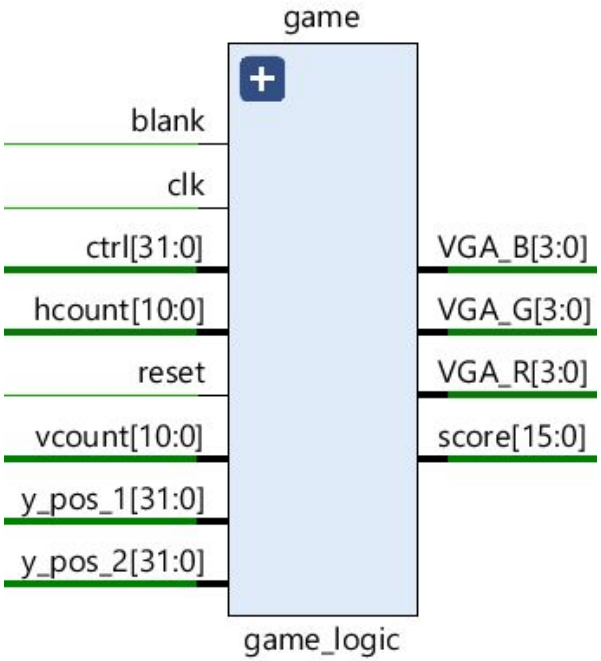

```
module ball_ctrl
    #(//screen resolution in pixels
      parameter SCREEN_WIDTH   = 10'd640,
      parameter SCREEN_HEIGHT  = 9'd480,
      //dimensions of the game field in pixels
      parameter BORDER_WIDTH   = 4'd10,
      parameter Y_UP_BORDER    = 10'd19,
      parameter Y_DOWN_BORDER  = 10'd460,
      //the positions of the paddle_1 and paddle_2
      parameter PADDLE_X_1 = 5'd19,
      parameter PADDLE_X_2 = 10'd616,
      //paddle's width in pixels
      parameter PADDLE_WIDTH  = 10'd5,
      //ball's speed
      parameter BALL_SPEED = 20'd1_000_000,
      //ball's width and height in pixels
      parameter BALL_SIZE  = 4'd10)
    (
        input clk,
        input reset,
        input[10:0] hcount,
        input[10:0] vcount,
        input blank,
        input[31:0] ctrl,
        output reg draw_ball,
        output reg[9:0] ball_x,
        output reg[8:0] ball_y
    );
```

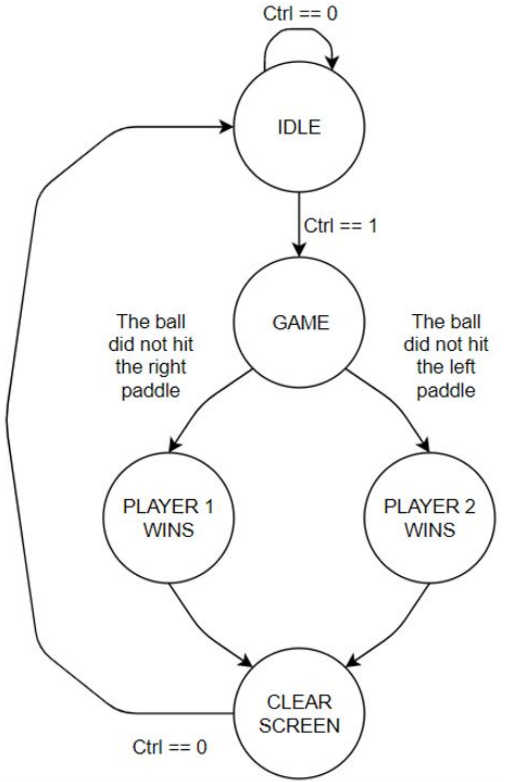
Custom Modules: game_logic.v



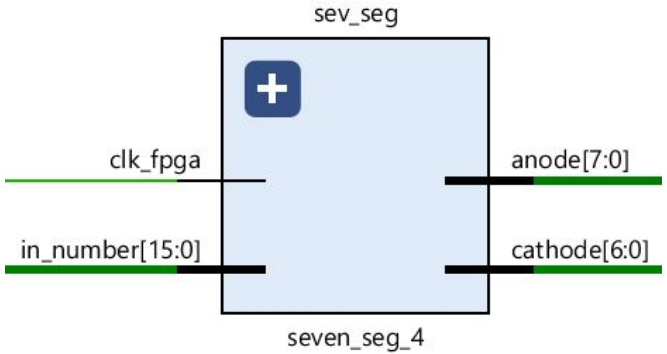

```
module game_logic(
    input clk,
    input reset,
    input[10:0] hcount,
    input[10:0] vcount,
    input blank,
    input[31:0] y_pos_1,
    input[31:0] y_pos_2,
    input[31:0] ctrl,
    output [3:0] VGA_R,
    output [3:0] VGA_G,
    output [3:0] VGA_B,
    output wire [15:0] score
    );
```

Custom Modules: game_logic.v

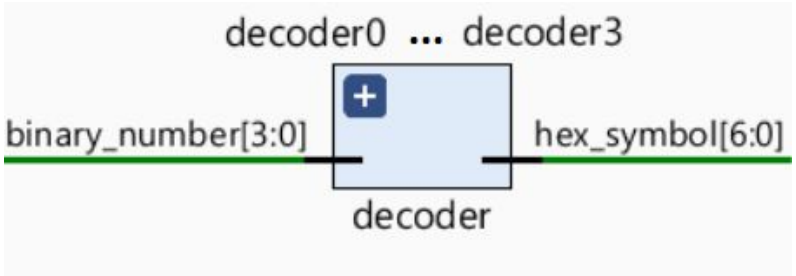




Custom Modules: seven_segment_4.vhd



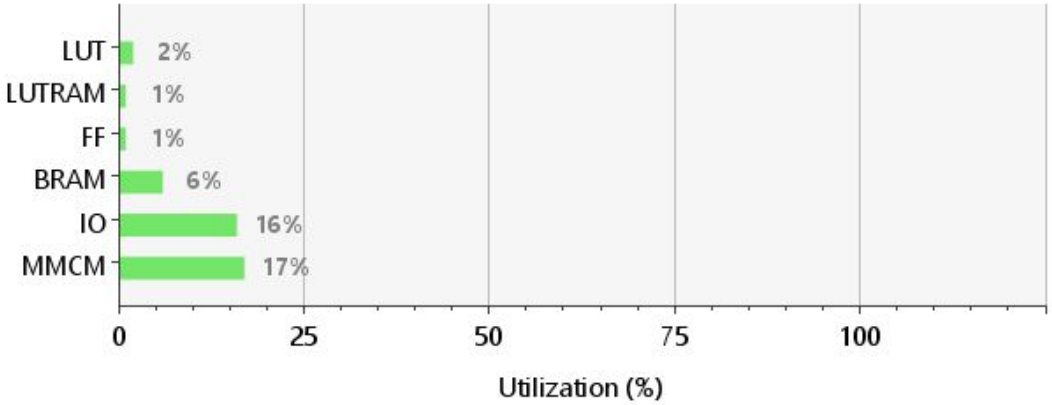

```
entity seven_seg_4 is
    Port(
         in_number : in std_logic_vector(15 downto 0);
         clk_fpga  : in std_logic;
         cathode   : out std_logic_vector(6 downto 0);
         anode     : out std_logic_vector(7 downto 0)
        );
end seven_seg_4;
```




```
entity decoder is
    port (
         binary_number : in std_logic_vector(3 downto 0);
         hex_symbol : out std_logic_vector(6 downto 0)
         );
end decoder;
```

FPGA Resource Usage

Resource	Utilization	Available	Utilization %
LUT	1122	63400	1.77
LUTRAM	182	19000	0.96
FF	1056	126800	0.83
BRAM	8	135	5.93
IO	33	210	15.71
MMCM	1	6	16.67



FPGA Resource Usage

Name	Slice LUTs (63400)	Slice Registers (126800)	F7 Muxes (31700)	Block RAM Tile (135)	Bonded IOB (210)	BUFGCTRL (32)	MMCME2_ADV (6)
N top	1122	1056	32	8	33	3	1
game (game_logic)	257	81	0	0	0	0	0
ball (ball_ctrl)	107	59	0	0	0	0	0
borders (borders_ctrl)	0	1	0	0	0	0	0
paddle_1 (paddle_ctrl)	20	1	0	0	0	0	0
paddle_2 (paddle_ctrl__parameter	20	1	0	0	0	0	0
mmcm_ip (clk_wiz_0)	0	0	0	0	0	3	1
sev_seg (seven_seg_4)	9	20	0	0	0	0	0
ublaze_mcs_ip (microblaze_mcs_0)	776	930	32	8	0	0	0
vga_ctrl_ip (vga_controller_640_60)	92	25	0	0	0	0	0

Thank you!