Digital Electronics Verilog Lab 7

Submitted By:
Saurabh Kumar
SC ID: SC22B146
Course: DIGITAL ELECTRONICS AND VLSI DESIGN LAB (AV232)

LAB SHEET

Question No. 1. Instantiate a block random access memory (BRAM) of FPGA to store 25 contents of the last lab sheet. Provide input to BRAM using:

i) test bench.
ii) Assignment: Read the data using $readmemb or $readmemh from a text file.
iii) Assignment: Initialize the BRAM with the data.

Sol: Code:

```
module RAM (clk,add,din,en,we,dout);
 input wire clk;
 input wire[4:0] add;
 input wire[7:0] din;
 input wire en;
 input wire[0:0] we;
 output wire[7:0] dout;

// Block Memory Generator Instantiation
 blk_mem_gen_0 your_instance_name (
 .clk(clk),
 .en(en),
 .we(we),
 .addr(addr),
 .din(din),
 .dout(dout));
endmodule
```

Testbench Code:

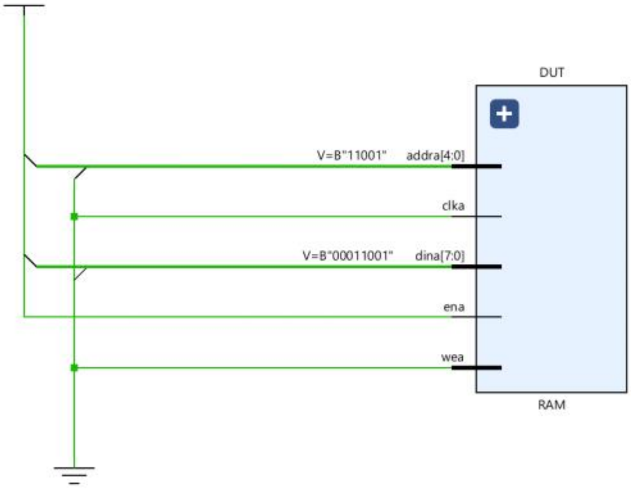
```
module tb_RAM();
 reg t_clk, t_cs, t_we;
 reg [4:0] t_address;
 reg [7:0] t_data;
 integer i;
 RAM DUT(t_clk, t_address, t_data, t_cs, t_we);
initial begin
 t_clk = 0;
 t_cs = 0;
 t_we = 0;
 end

initial begin
 forever #5 t_clk = ~t_clk;
 end
initial begin
 t_data = 8'b00000000;
 t_address = 5'b00000;
 #10 t_cs = 1;
 t_we = 1;
 for (i = 0; i < 25; i = i + 1) begin
 #10 t_data = t_data + 8'b00000001;
 t_address = t_address + 5'b00001;
 end
 #10 t_cs = 1;
 t_re = 1;
 t_we = 0;
 t_address = 5'b00000;
 for (i = 0; i < 25; i = i + 1) begin
 #10 t_address = t_address + 5'b00001;
 end
 end
endmodule
```

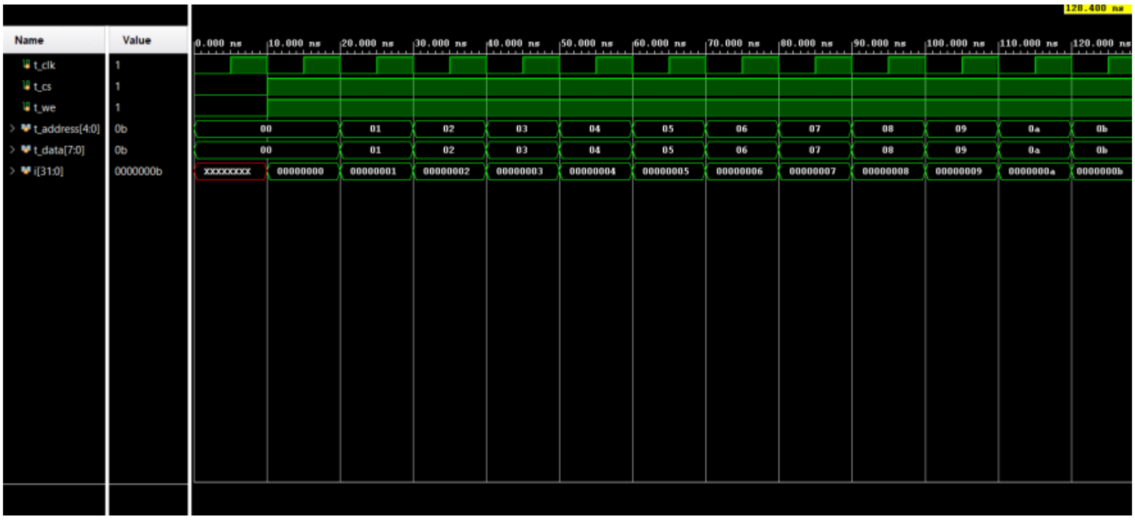
Constraint File:

```
set_property PACKAGE_PIN H16 [get_ports clka_0]
set_property IOSTANDARD LVCMOS33 [get_ports clka_0]
set_property C_CLK_INPUT_FREQ_HZ 300000000 [get_debug_cores dbg_hub]
set_property C_ENABLE_CLK_DIVIDER false [get_debug_cores dbg_hub]
set_property C_USER_SCAN_CHAIN 1 [get_debug_cores dbg_hub]
connect_debug_port dbg_hub/clk [get_nets clka_0_IBUF]
```

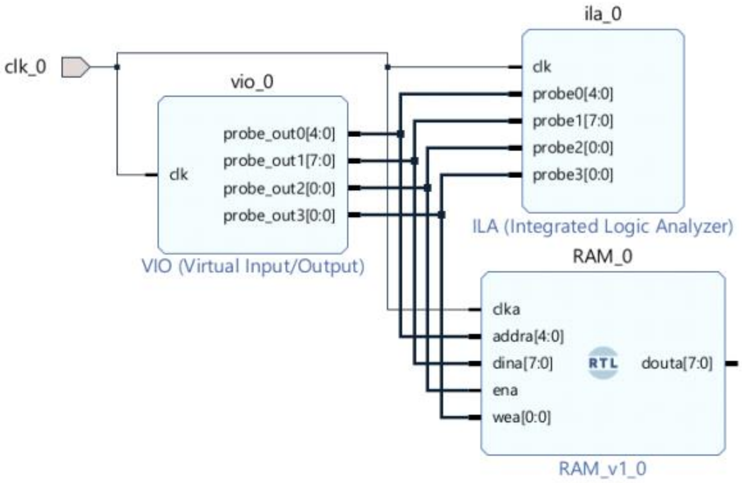
Schematic:



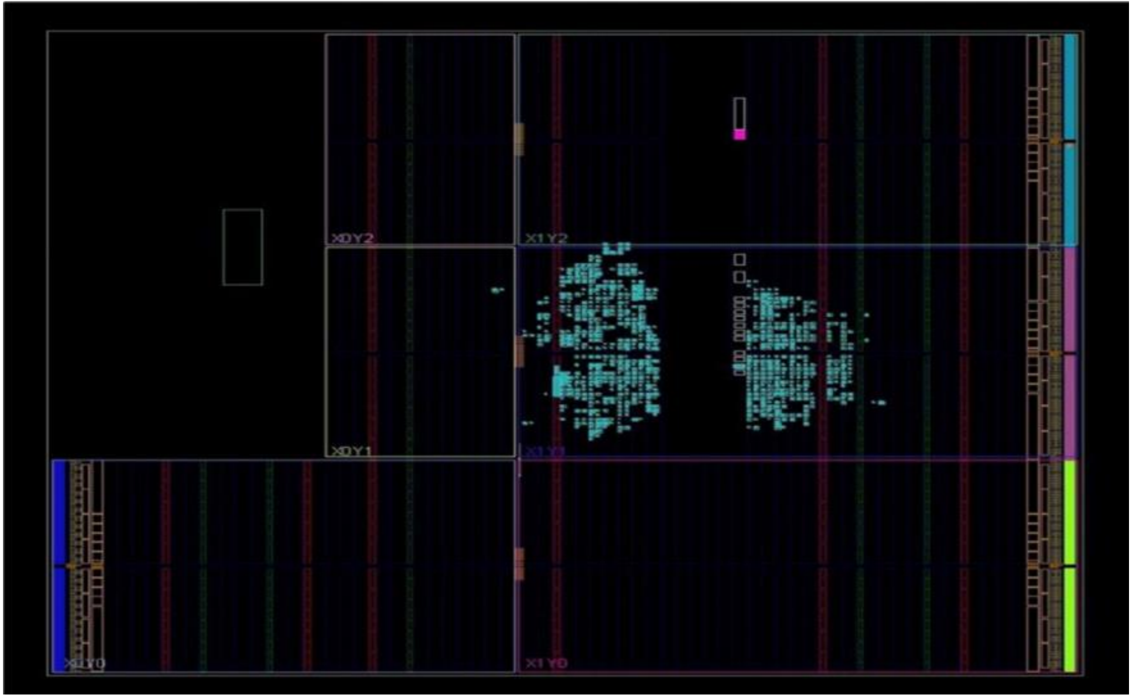
Simulation:



Block Design:



Synthesis:



Question No. 2. Design a 4-bit Ring counter with [7:0]q bits. Perform behavioural simulations and follow the steps:

i) Open block diagram.
ii) Right click -> Import module -> add the counter module to the diagram.
iii) Add Virtual Input-Output (VIO) and Integrated Logic Analyser (ILA) IP from IP catalogue.
iv) Follow the tutorial.
Provide inputs reset and clear through VIO and use system clock H16 for the design.

Sol: Code:

```
// Ring Counter
module counter(clk,reset,en,Q);
   input clk,reset,en;
   output reg [0:7]Q;

//assign Q[0] = 1;
```

```
initial begin
  Q[0] <= 1;
  Q[1] <= 0;
  Q[2] <= 0;
  Q[3] <= 0;
  Q[4] <= 0;
  Q[5] <= 0;
  Q[6] <= 0;
  Q[7] <= 0;
end

always@(posedge clk)
begin
if(en)
begin
  if(reset)
    begin
    Q = 0;
    end
  else
    begin
//      Q[0] = 1;
    Q[1] <= Q[0];
    Q[2] <= Q[1];
    Q[3] <= Q[2];
    Q[4] <= Q[3];
    Q[5] <= Q[4];
    Q[6] <= Q[5];
    Q[7] <= Q[6];
    Q[0] <= Q[7];
    end
end
end
endmodule
```

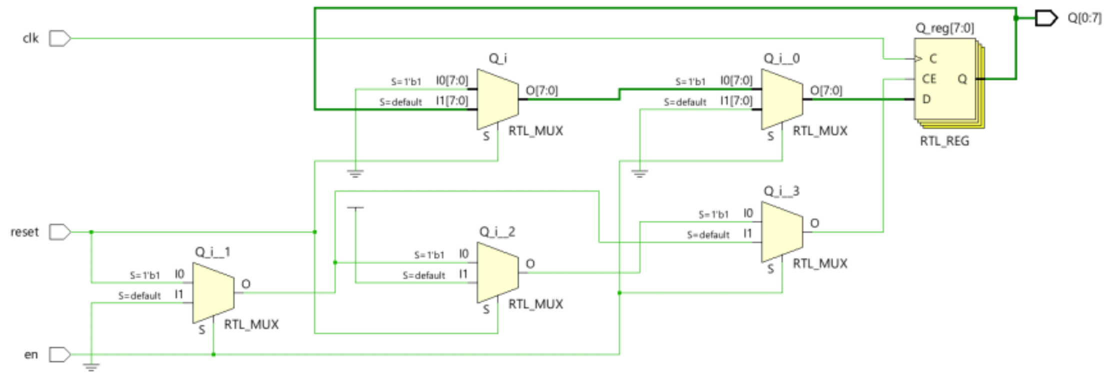
Testbench Code:

```
module tb_counter();
reg enable,reset,clk;
wire [7:0]q;
counter c1(clk,reset,enable,q);
initial
begin
reset = 0;
enable = 0; clk = 0;
end
always forever begin
#5 clk = ~clk;
end
initial begin
#5 reset = 1;
#20 reset = 0;
enable = 1;
end
endmodule
```

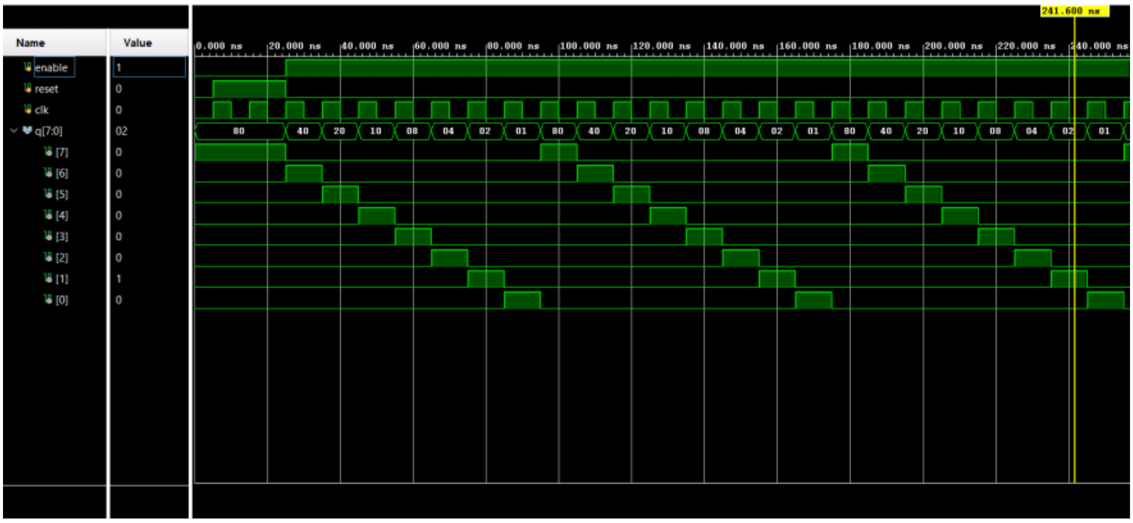
Constraint File:

```
set_property IOSTANDARD LVCMOS33 [get_ports clk_0]
set_property PACKAGE_PIN H16 [get_ports clk_0]
set_property C_CLK_INPUT_FREQ_HZ 300000000 [get_debug_cores
dbg_hub]
set_property C_ENABLE_CLK_DIVIDER false [get_debug_cores
dbg_hub]
set_property C_USER_SCAN_CHAIN 1 [get_debug_cores dbg_hub]
connect_debug_port dbg_hub/clk [get_nets clk_0_IBUF]
```

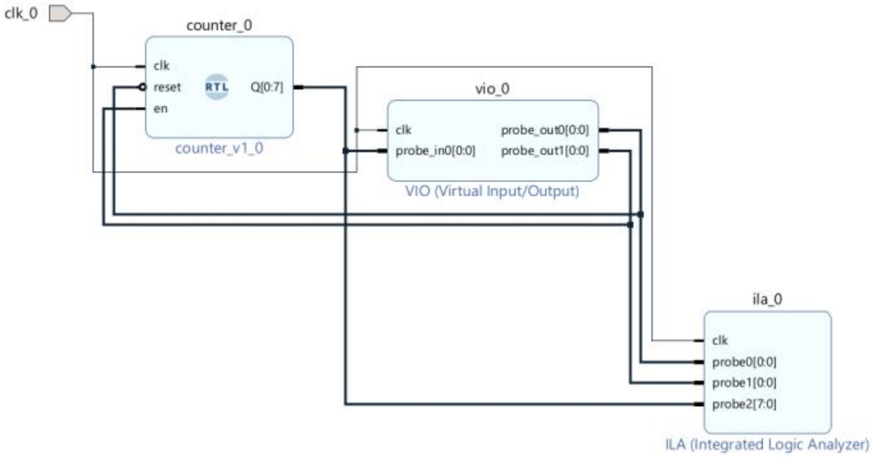
Schematic:



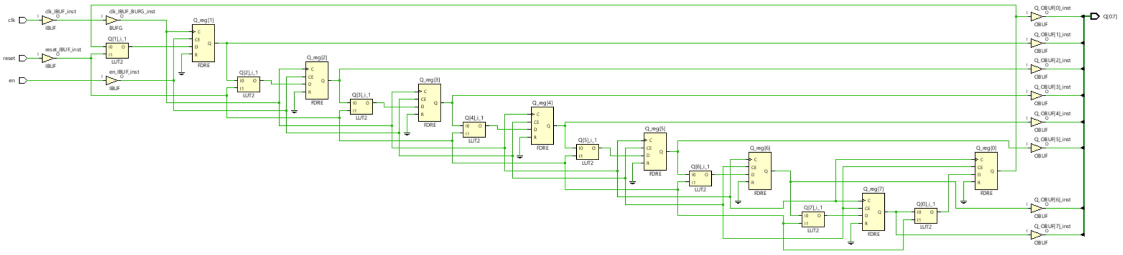
Simulation:



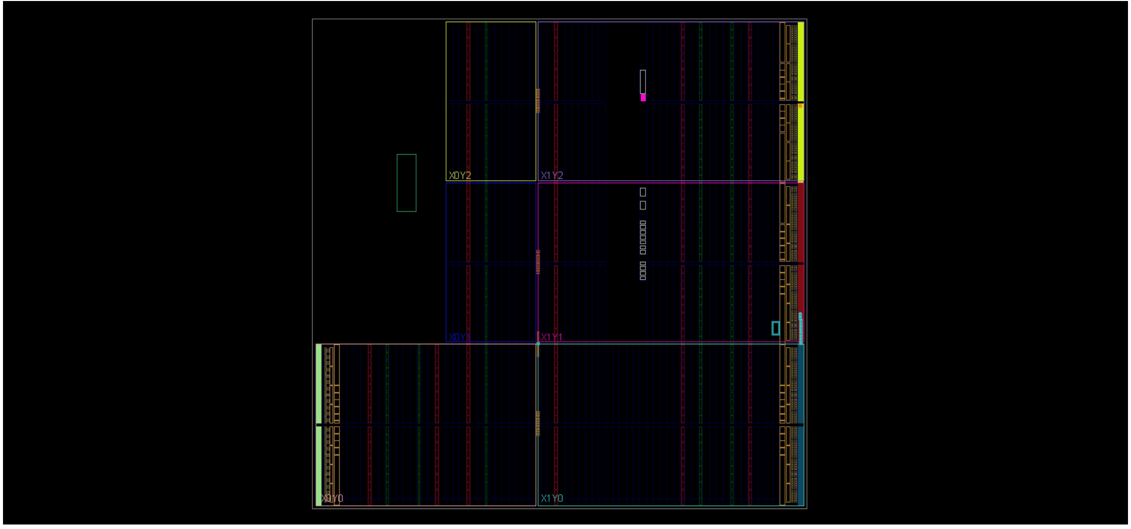
Block Diagram:



Synthesized Schematic:


clk
reset
en
clk_IBUF_inst
IBUF
clk_IBUF_BUFG_inst
BUFG
reset_IBUF_inst
IBUF
en_IBUF_inst
IBUF
Q[1]_i_1
LUT2
Q_reg[1]
FDRE
Q[2]_i_1
LUT2
Q_reg[2]
FDRE
Q[3]_i_1
LUT2
Q_reg[3]
FDRE
Q[4]_i_1
LUT2
Q_reg[4]
FDRE
Q[5]_i_1
LUT2
Q_reg[5]
FDRE
Q[6]_i_1
LUT2
Q_reg[6]
FDRE
Q[7]_i_1
LUT2
Q_reg[7]
FDRE
Q[0]_i_1
LUT2
Q_reg[0]
FDRE
Q_OBUF[0]_inst
Q_OBUF[1]_inst
Q_OBUF[2]_inst
Q_OBUF[3]_inst
Q_OBUF[4]_inst
Q_OBUF[5]_inst
Q_OBUF[6]_inst
Q_OBUF[7]_inst
OBUF
Q[0:7]

Synthesis:


X0Y2
X1Y2
X0Y1
X1Y1
X0Y0
X1Y0