EXPERIMENT NO.1: ALL GATES

VERILOG CODE

1.DATAFLOW MODELLING

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//
// Company:
// Engineer:
//
// Create Date: 11.08.2017 10:46:49
// Design Name:
// Module Name: AllGates
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
//


module AllGates(
    input a,
    input b,
    output [6:0] y
    );
    assign y[0]=~a;
    assign y[1]=a&b;
    assign y[2]=a|b;
    assign y[3]=~(a&b);
    assign y[4]=~(a|b);
    assign y[5]=a^b;
    assign y[6]=~(a^b);
endmodule
```

SIMULATION RESULT

EXPERIMENT NO.2: 2:4 DECODER

VERILOG CODE

1.DATAFLOW MODELING

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//
// Company:
// Engineer:
//
// Create Date: 11.08.2017 11:06:34
// Design Name:
// Module Name: Decoder
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
//


module Decoder(
    input a,
    input b,
    input e,
    output [3:0] z
    );
    wire abar,bbar;
    assign abar=~a;
    assign bbar=~b;
    assign z[0]=~(abar&bbar&e);
    assign z[1]=~(abar&b&e);
    assign z[2]=~(a&bbar&e);
    assign z[3]=~(a&b&e);
endmodule
```

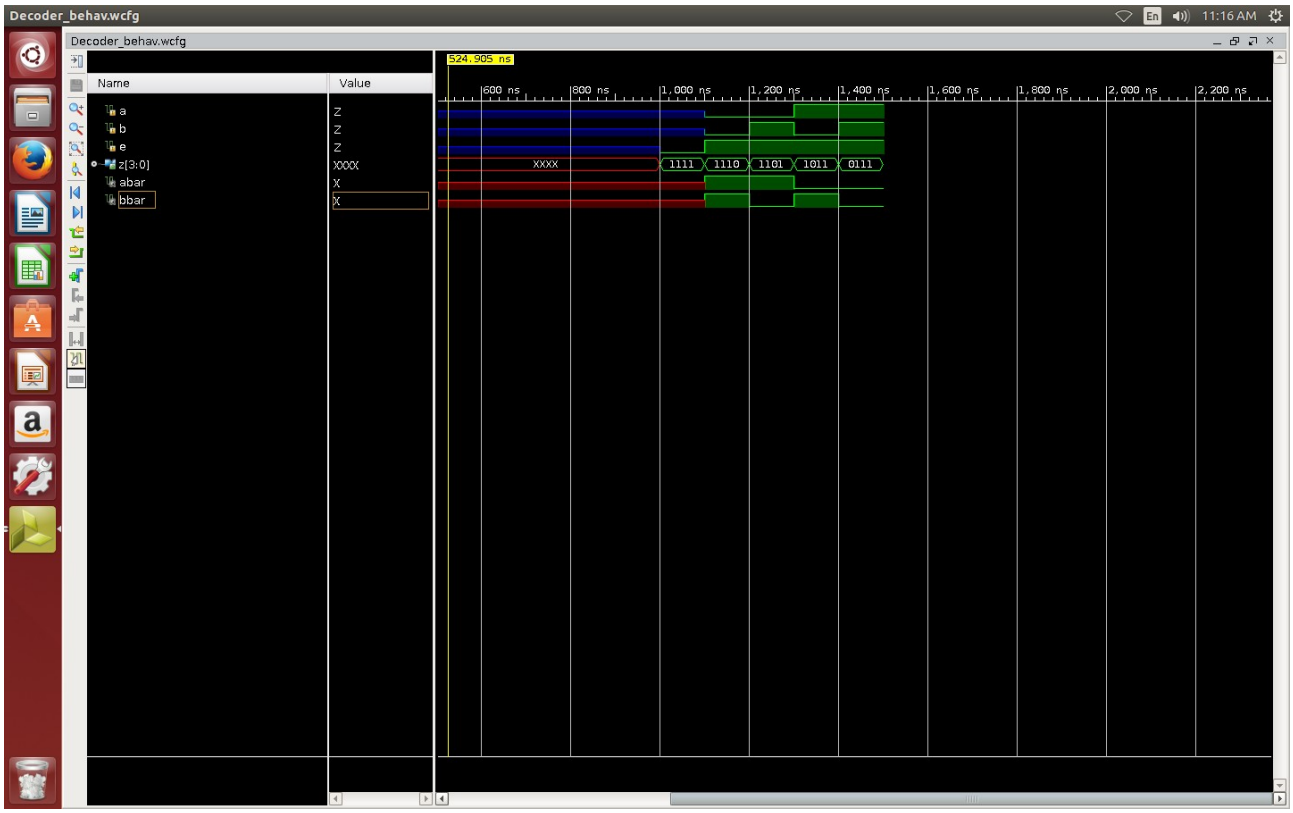
2.BEHAVIOURAL MODELLING

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//
// Company:
// Engineer:
//
// Create Date: 11.08.2017 11:20:22
// Design Name:
// Module Name: Decoder2
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
```

```
//
//////////////////////////////////////////////////////////////////////////////////
//


module Decoder2(
    input a,
    input b,
    input e,
    output [3:0] z
    );
    reg [3:0]z;
    always@(e,a,b)
    begin
    if(!e)
    z=4'b1111;
    else
    case({a,b})
    2'b00:z=4'b1110;
    2'b01:z=4'b1101;
    2'b10:z=4'b1011;
    2'b11:z=4'b0111;
    default:z=4'b1111;
    endcase
    end
endmodule

3.STRUCTURAL MODELLING
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//
// Company:
// Engineer:
//
// Create Date: 11.08.2017 11:28:22
// Design Name:
// Module Name: Decoder3
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
//


module Decoder3(
    input x,
    input y,
    input e,
    output [3:0] w
    );
    wire xbar,ybar;
    not x1(xbar,x);
    not x2(ybar,y);
    nand x3(w[0],xbar,ybar,e);
    nand x4(w[1],xbar,y,e);
    nand x5(w[2],x,ybar,e);
```

```
    nand x6(w[3],x,y,e);
endmodule
```

SIMULATION RESULT



EXPERIMENT NO.3: BINARY TO GREY CODE CONVERTER

VERILOG CODE

1.DATAFLOW MODELLING

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//
// Company:
// Engineer:
//
// Create Date: 18.08.2017 11:06:24
// Design Name:
// Module Name: b2g
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
//


module b2g(
    input [3:0] b,
    output [3:0] g
    );
    assign g[3]=b[3];
    assign g[2]=b[2]^b[3];
    assign g[1]=b[1]^b[2];
    assign g[0]=b[0]^b[1];
endmodule
```

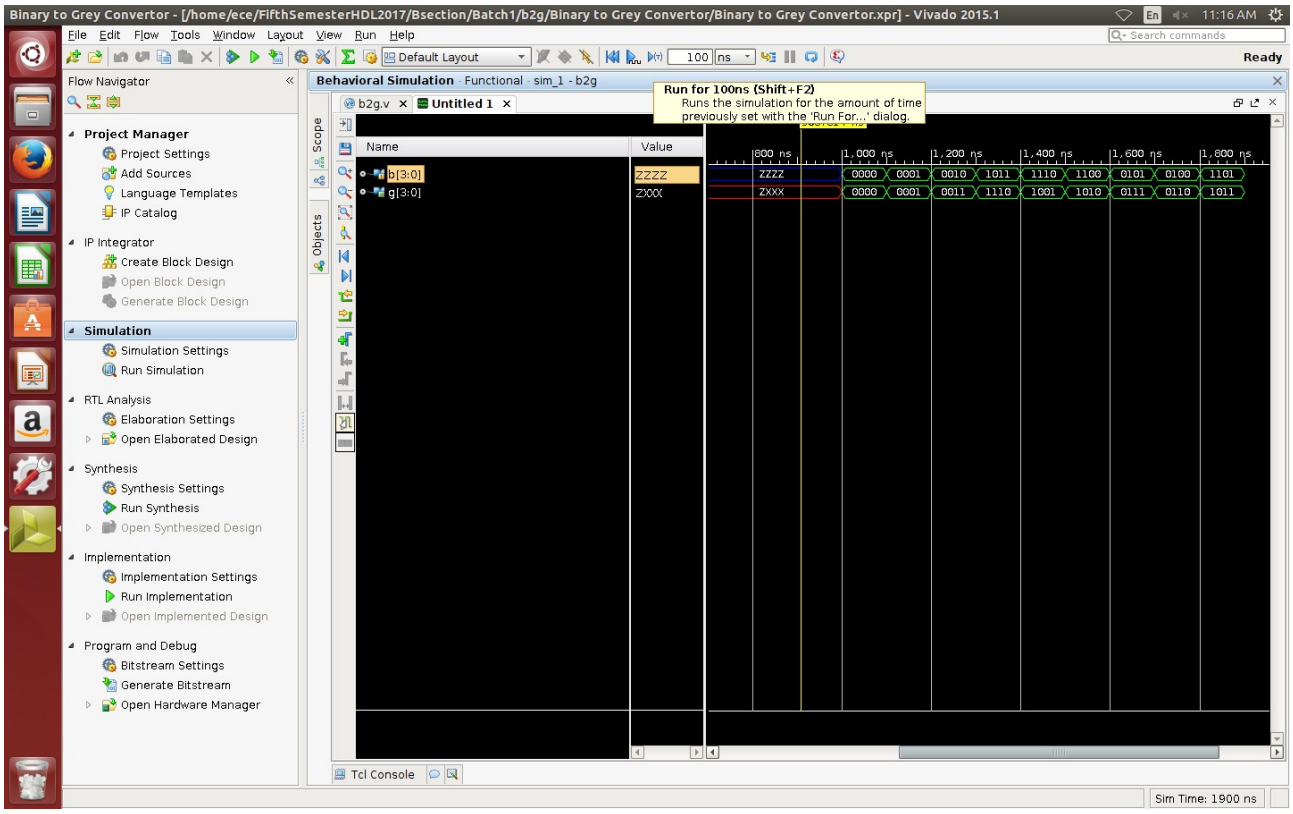
2.BEHAVIOURAL MODELLING

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//
// Company:
// Engineer:
//
// Create Date: 18.08.2017 11:18:40
// Design Name:
// Module Name: b2g1
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
//
```

```
module b2g1(
    input [3:0] b,
    output [3:0] g,
    input e
    );

    reg [3:0]g;
    always@(b)
    begin
    if (!e)
    g=4'b0000;
    else
    case(b)
    4'b0000:g=4'b0000;
    4'b0001:g=4'b0001;
    4'b0010:g=4'b0011;
    4'b0011:g=4'b0010;
    4'b0100:g=4'b0110;
    4'b0101:g=4'b0111;
    4'b0110:g=4'b0101;
    4'b0111:g=4'b0100;
    4'b1000:g=4'b1100;
    4'b1001:g=4'b1101;
    4'b1010:g=4'b1111;
    4'b1011:g=4'b1110;
    4'b1100:g=4'b1010;
    4'b1101:g=4'b1011;
    4'b1110:g=4'b1001;
    4'b1111:g=4'b1000;
    default:g=4'b1111;
    endcase
    end
endmodule

3.STRUCTURAL MODELLING
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//
// Company:
// Engineer:
//
// Create Date: 18.08.2017 11:30:02
// Design Name:
// Module Name: b2g2
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
//


module b2g2(
    input [3:0] b,
    output [3:0] g,
     input e
    );
```

```
    wire [3:0] o;
    xor x1(o[0],b[3]);
    and x2(g[0],o[0],e);
    xor x3(o[1],b[2],b[3]);
    and x4(g[1],o[1],e);
    xor x5(o[2],b[1],b[2]);
    and x6(g[2],o[2],e);
    xor x7(o[3],b[0],b[1]);
    and x8(g[3],o[3],e);
endmodule
```

SIMULATION RESULT



EXPERIMENT NO.4: 8:1 MULTIPLEXER

VERILOG CODE

1. DATAFLOW MODELLING

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//
// Company:
// Engineer:
//
// Create Date: 19.08.2017 11:23:24
// Design Name:
// Module Name: mux
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
//


module mux(
    input [7:0] i,
    input [2:0] a,
    input e,
    output y
    );

    assign y = (e==0)?0:(a==3'b000)?i[0]:(a==3'b001)?i[1]:(a==3'b010)?i[2]:
(a==3'b011)?i[3]:(a==3'b100)?i[4]:(a==3'b101)?i[5]:(a==3'b110)?i[6]:i[7];


endmodule
```

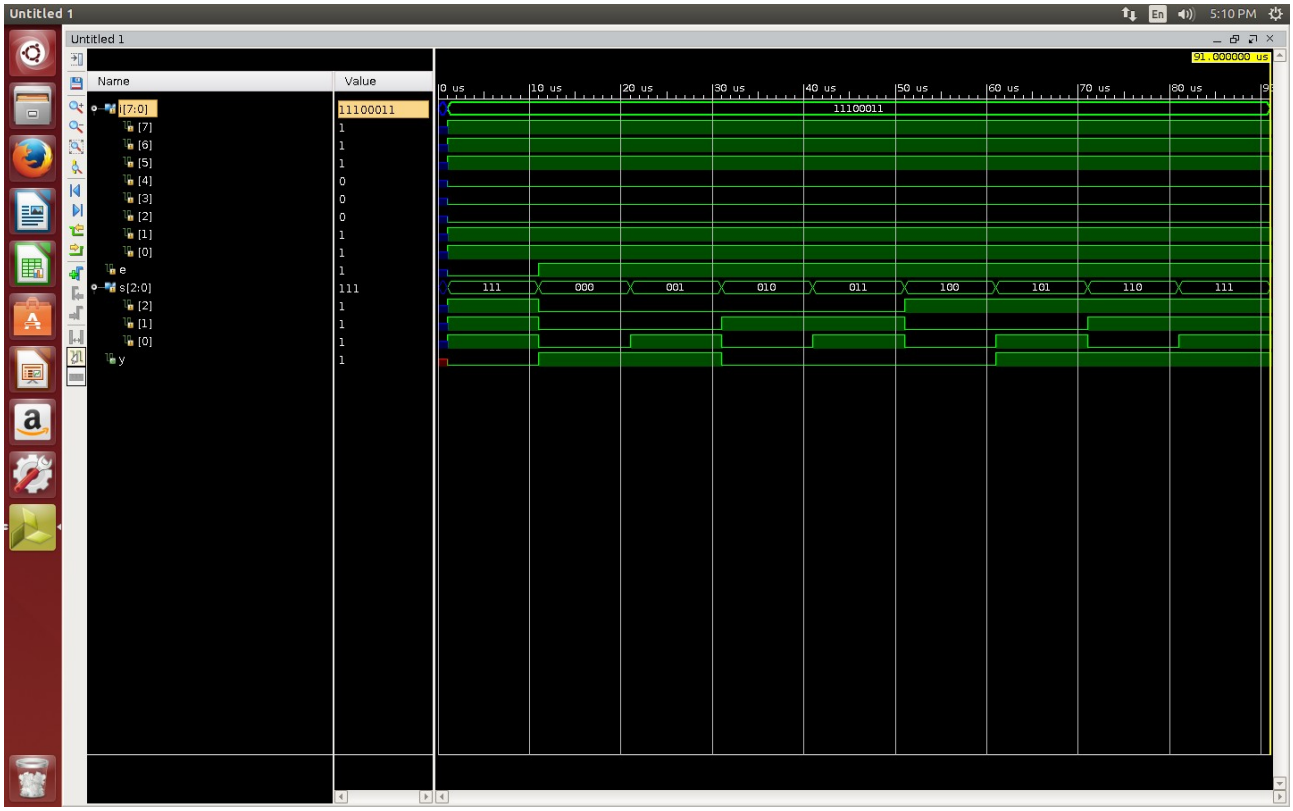
2.BEHAVIOURAL MODELLING

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//
// Company:
// Engineer:
//
// Create Date: 19.08.2017 11:18:52
// Design Name:
// Module Name: Multiplexer
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
```

```
//


module Multiplexer(
    input e,
    input [2:0] s,
    input [7:0] i,
    output o
    );
    reg o;
    always@(e,i,s)
    begin
    if(e==0)
    o=0;
    else
    case(s)
    3'b000:o=i[0];
    3'b001:o=i[1];
    3'b010:o=i[2];
    3'b011:o=i[3];
    3'b100:o=i[4];
    3'b101:o=i[5];
    3'b110:o=i[6];
    3'b111:o=i[7];
    default:o=0;
    endcase
    end
endmodule

3.STRUCTURAL MODELLING
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//
// Company:
// Engineer:
//
// Create Date: 19.08.2017 11:39:27
// Design Name:
// Module Name: muxsm
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
//


module muxsm(
    input [7:0] i,
    input [2:0] a,
    input e,
    output y
    );

    wire a,b,c;
    wire [7:0]o;
```

```
    not n1(a,a[2]);
    not n2(b,a[1]);
    not n3(c,a[0]);

    and n4(o[0],a,b,c,i[0],e);
    and n5(o[1],a,b,a[0],i[1],e);
    and n6(o[2],a,a[1],c,i[2],e);
    and n7(o[3],a,a[1],a[0],i[3],e);
    and n8(o[4],a[2],b,c,i[4],e);
    and n9(o[5],a[2],b,a[0],i[5],e);
    and n10(o[6],a[2],a[1],c,i[6],e);
    and n11(o[7],a[2],a[1],a[0],i[7],e);

    or n12(y,o[0],o[1],o[2],o[3],o[4],o[5],o[6],o[7]);
endmodule
```

SIMULATION RESULT



EXPERIMENT NO.5: 8:3 ENCODER

VERILOG CODE

A.WITHOUT PRIORITY

1.DATAFLOW MODELLING

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//
// Company:
// Engineer:
//
// Create Date: 01.09.2017 11:40:36
// Design Name:
// Module Name: encoderwithout
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
//


module encoderwithout(
    input [7:0] x,
    input e,
    output [2:0] y
    );
    assign y[0]=e&(x[1]|x[3]|x[5]|x[7]);
    assign y[1]=e&(x[2]|x[3]|x[6]|x[7]);
    assign y[2]=e&(x[4]|x[5]|x[6]|x[7]);
endmodule
```

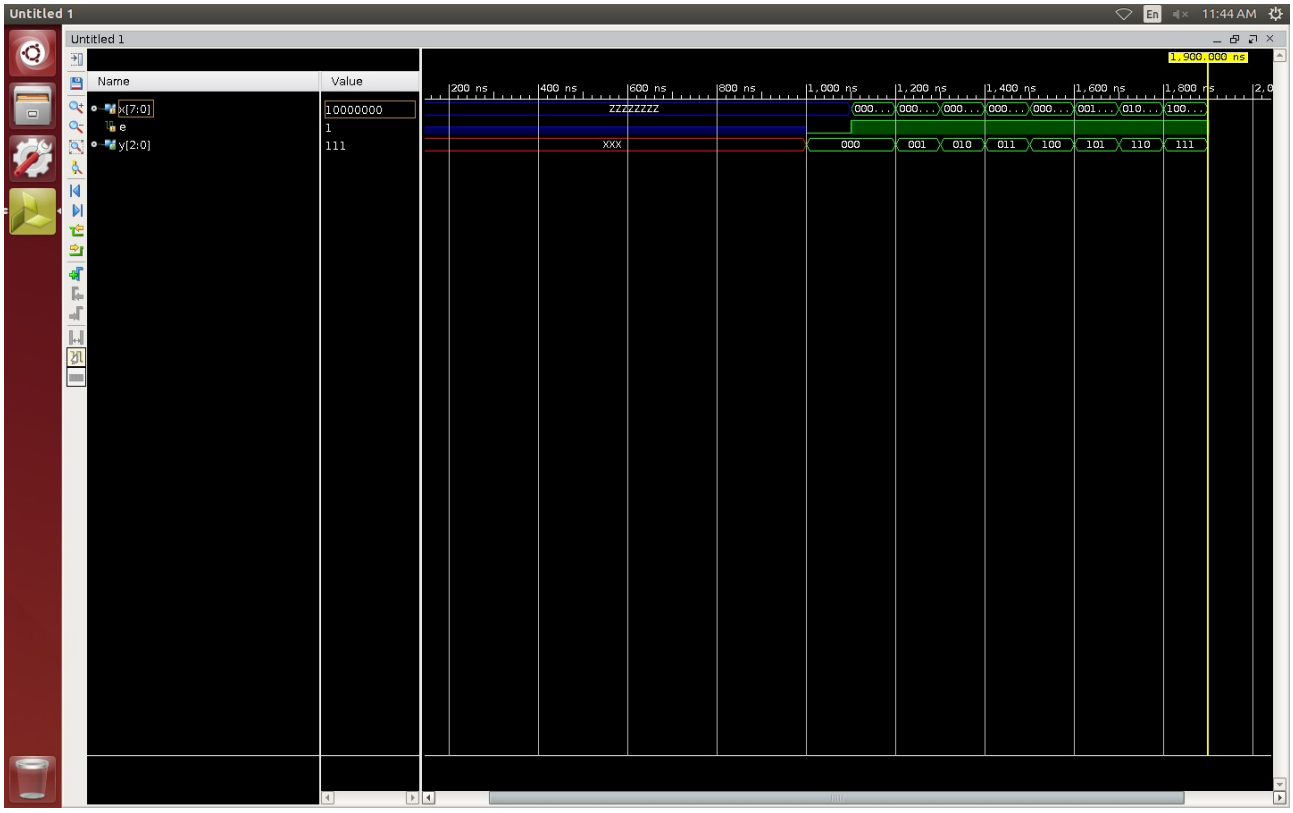
2.BEHAVIOURAL MODELLING

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//
// Company:
// Engineer:
//
// Create Date: 01.09.2017 11:46:04
// Design Name:
// Module Name: encoderwithout1
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
//
```

```
module encoderwithout1(
    input [7:0] x,
    input e,
    output [2:0] y
    );
    reg [2:0] y;
    always@(e,x)
    begin
    if (!e)
    y=3'b000;
    else
    case(x)
    8'b00000001:y=3'b000;
    8'b00000010:y=3'b001;
    8'b00000100:y=3'b010;
    8'b00001000:y=3'b011;
    8'b00010000:y=3'b100;
    8'b00100000:y=3'b101;
    8'b01000000:y=3'b110;
    8'b10000000:y=3'b111;
    default:y=3'b000;
    endcase
    end
endmodule

3.STRUCTURAL MODELLING
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//
// Company:
// Engineer:
//
// Create Date: 01.09.2017 11:53:27
// Design Name:
// Module Name: encoderwithout2
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
//


module encoderwithout2(
    input [7:0] x,
    input e,
    output [2:0] y
    );
    wire a,b,c;
    or x1(a,x[1],x[3],x[5],x[7]);
    and x2(y[0],a,e);
    or x3(b,x[2],x[3],x[6],x[7]);
    and x4(y[1],b,e);
    or x5(c,x[4],x[5],x[6],x[7]);
    and x6(y[2],c,e);
```

```
endmodule
```

SIMULATION RESULT



B.WITH PRIORITY

1.DATAFLOW MODELLING

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//
// Company:
// Engineer:
//
// Create Date: 01.09.2017 10:38:24
// Design Name:
// Module Name: EncoderPriority
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
//


module EncoderPriority(
    input [7:0] x,
    input e,
    output e1,
    output s,
```

```
    output [2:0] y
    );
    assign s=e&(x[0]|x[1]|x[2]|x[3]|x[4]|x[5]|x[6]|x[7]);
    assign e1=e&(~(x[0]|x[1]|x[2]|x[3]|x[4]|x[5]|x[6]|x[7]));
    assign y=(e==0)?0:(x[7]==1)?(3'b111):(x[6]==1)?(3'b110):(x[5]==1)?(3'b101):
(x[4]==1)?(3'b100):(x[3]==1)?(3'b011):(x[2]==1)?(3'b010):(x[1]==1)?(3'b001):
(3'b000);
endmodule
```

2.BEHAVIOURAL MODELLING

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//
// Company:
// Engineer:
//
// Create Date: 01.09.2017 10:52:12
// Design Name:
// Module Name: EncoderPriority1
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
//


module EncoderPriority1(
    input [7:0] x,
    input e,
    output e1,
    output s,
    output [2:0] y
    );
   reg [2:0]y;
   reg s,e1;
   always@(e,x)
   begin
   if(!e)
   begin
   y=3'b000;
   s=0;
   e1=0;
   end
   else if(x==8'b0)
   begin
   y=3'b0;
   s=0;
   e1=1;
   end
   else
   begin
   s=1;
   e1=0;
   begin
   if(x[7]==1)
   y=3'b111;
```

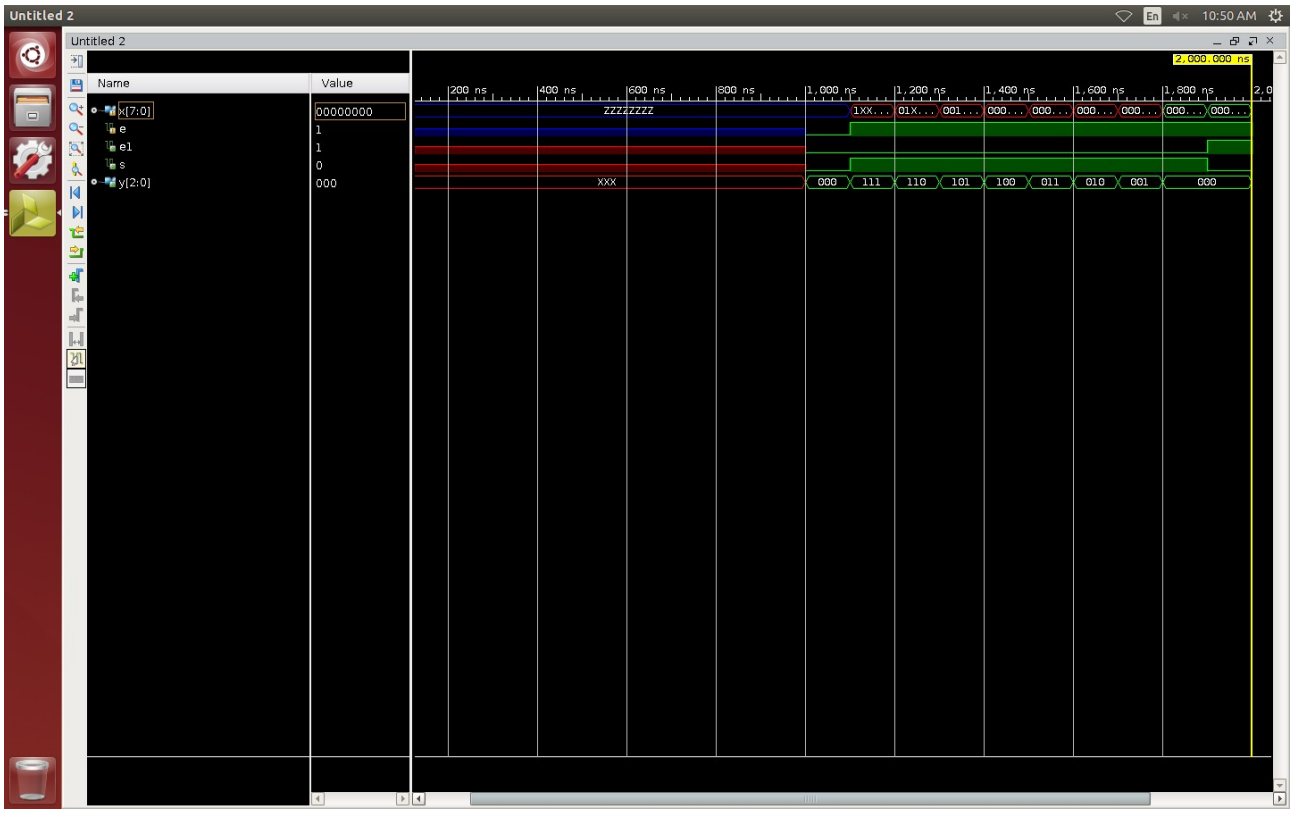
```
   else if(x[6]==1)
   y=3'b110;
   else if(x[5]==1)
      y=3'b101;
      else if(x[4]==1)
         y=3'b100;
         else if(x[3]==1)
            y=3'b011;
            else if(x[2]==1)
               y=3'b010;
               else if(x[1]==1)
                  y=3'b001;
                  else
                  y=3'b0;
                  end
                  end
                  end
endmodule
```

3.STRUCTURAL MODELLING

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//
// Company:
// Engineer:
//
// Create Date: 01.09.2017 10:59:34
// Design Name:
// Module Name: EncoderPriority2
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
//


module EncoderPriority2(
    input [7:0] x,
    input e,
    output e1,
    output s,
    output [2:0] y
    );
    wire [13:0]o;
    or x1(o[0],x[4],x[5],x[6],x[7]);
    and x2(y[2],o[0],e);
    not x3(o[1],x[5]);
    not x4(o[2],x[4]);
    or x5(o[3],x[2],x[3]);
    and x6(o[4],o[3],o[2],o[1]);
    and x7(o[5],o[4],o[1],o[2]);
    or x8(o[6],o[5],x[6],x[7]);
    and x9(y[1],o[6],e);
    not x10(o[7],x[2]);
    and x11(o[8],o[7],o[2],x[1]);
    and x12(o[9],x[2],x[3]);
```

```
    or x13(o[10],o[8],o[9],o[5]);
    not x14(o[11],x[6]);
    and x15(o[12],o[11],o[10]);
    or x16(o[13],o[12].x[7]);
    and x17(y[0],e,o[13]);
endmodule
```

SIMULATION RESULT



EXPERIMENT NO.6: 1:8 DEMUX

VERILOG CODE

1.DATAFLOW MODELLING

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//
// Company:
// Engineer:
//
// Create Date: 01.09.2017 12:08:49
// Design Name:
// Module Name: Demux
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
//


module Demux(
    input [2:0] a,
    input e,
    input d,
    output [7:0] i
    );
    assign i=(e==0)?(8'b0):(a==3'b000)?({7'b0,d}):(a==3'b001)?({6'b0,d,1'b0}):
(a==3'b010)?({5'b0,d,2'b0}):(a==3'b011)?({4'b0,d,3'b0}):(a==3'b100)?
({3'b0,d,4'b0}):(a==3'b101)?({2'b0,d,5'b0}):(a==3'b110)?({1'b0,d,6'b0}):
({d,7'b0});
endmodule
```

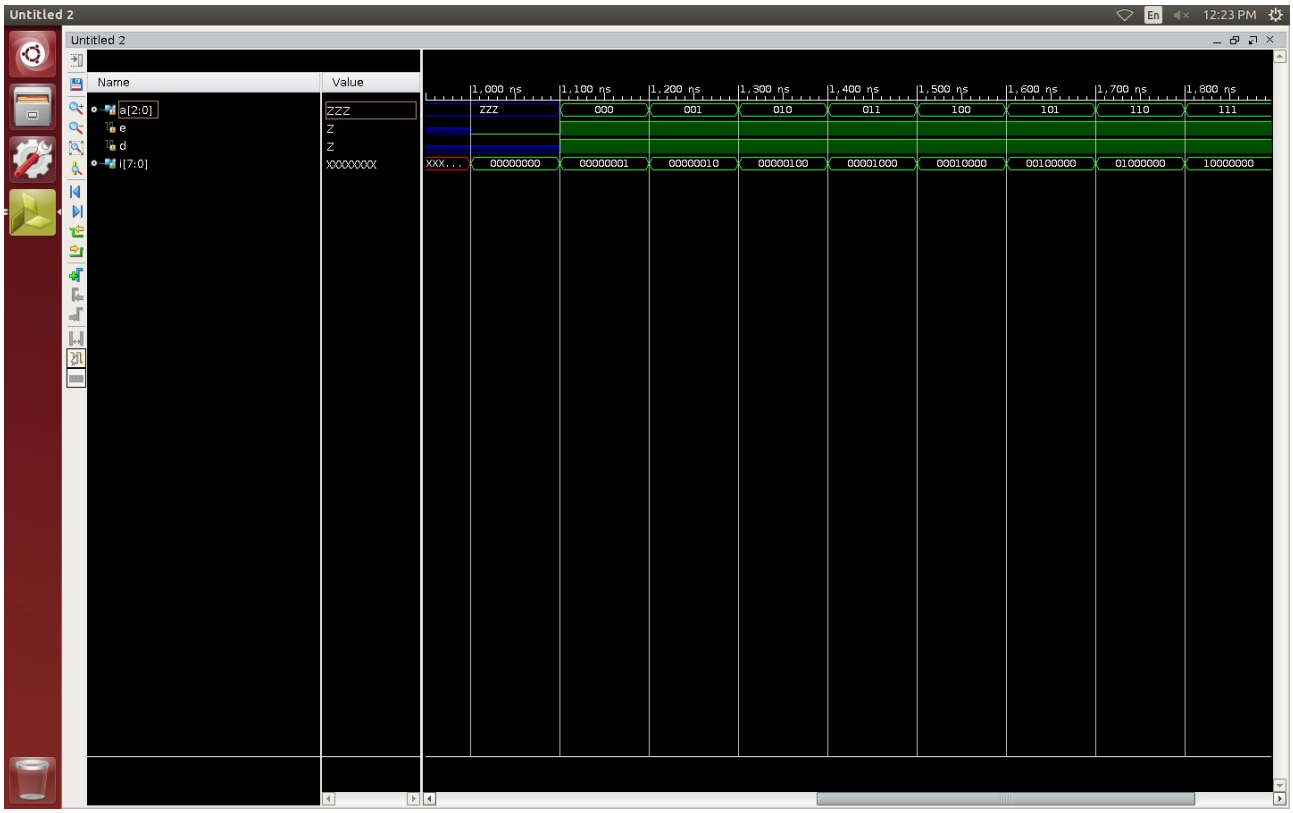
2.BEHAVIOURAL MODELLING

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//
// Company:
// Engineer:
//
// Create Date: 01.09.2017 12:25:29
// Design Name:
// Module Name: demux1
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
//
```

```
module demux1(
    output [7:0] i,
    input d,
    input [2:0] a,
    input e
    );
    reg [7:0]i;
    always@(e,a,d)
    begin
    if(!e)
    i=8'b0;
    else
    case(a)
    3'b000:i[0]=d;
    3'b001:i[1]=d;
    3'b010:i[2]=d;
    3'b011:i[3]=d;
    3'b100:i[4]=d;
    3'b101:i[5]=d;
    3'b110:i[6]=d;
    3'b111:i[7]=d;
    default:i=8'b0;
    endcase
    end
endmodule

3.STRUCTURAL MODELLING
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//
// Company:
// Engineer:
//
// Create Date: 01.09.2017 12:29:05
// Design Name:
// Module Name: demux2
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
//


module demux2(
    output [7:0] i,
    input [2:0] a,
    input e,
    input d
    );
    wire f,g,h;
    not n1(f,a[2]);
    not n2(g,a[1]);
    not n3(h,a[0]);
    and a1(i[0],f,g,h,e,d);
```

```
    and a2(i[1],f,g,a[o],e,d);
    and a3(i[2],f,a[1],h,e,d);
    and a4(i[3],f,a[1],a[0],e,d);
    and a5(i[4],a[2],g,h,e,d);
    and a6(i[5],a[2],g,a[0],e,d);
    and a7(i[6],a[2],a[1],h,e,d);
    and a8(i[7],a[2],a[1],a[0],e,d);
endmodule
```

SIMULATION RESULT



EXPERIMENT 7: COMPARATOR

VERILOG CODE

A.2 BIT COMPARATOR

1.DATAFLOW MODELLING

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//
// Company:
// Engineer:
//
// Create Date: 08.09.2017 10:37:17
// Design Name:
// Module Name: 2bit_Comp
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
//




module twobit_Comp(a,b,en,g,e,l);
    input [1:0]a;
    input [1:0]b;
    input en;
    output g,l,e;

    assign g=((a[1]&(~b[1]))|(a[0]&(~b[1])&(~b[0]))|(a[1]&a[0]&(~b[0])))&en;
    assign e=((~(a[1]^b[1]))&(~(b[0]^a[0])))&en;
    assign l=((~a[1]&b[1])|(~a[1]&~a[0]&b[0])|(~a[0]&b[1]&b[0]))&en;
    endmodule
```

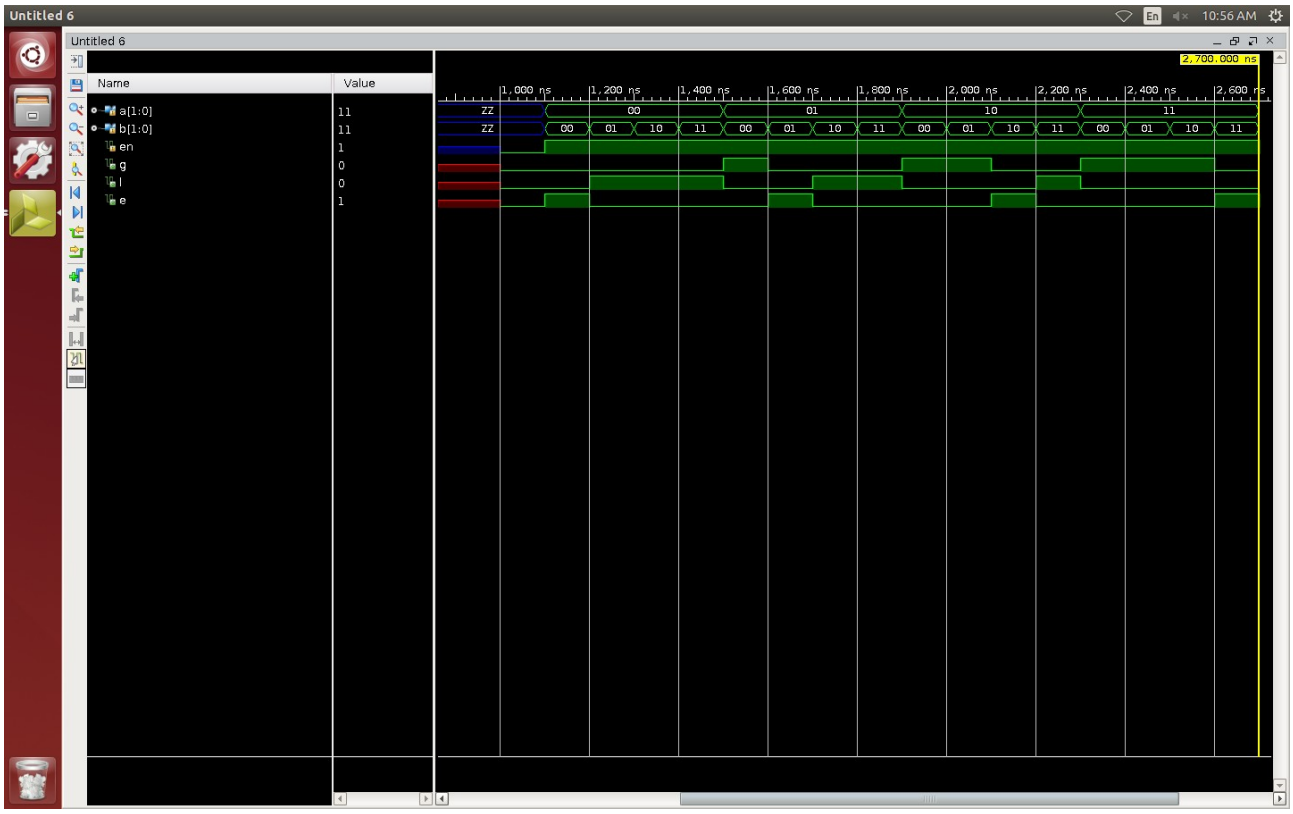
2.BEHAVIOURAL MODELLING

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//
// Company:
// Engineer:
//
// Create Date: 08.09.2017 11:04:05
// Design Name:
// Module Name: twobit_Comp1
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
```

```
//
//////////////////////////////////////////////////////////////////////////////////
//


module twobit_Comp1(
    input [1:0] a,
    input [1:0] b,
    input en,
    output [2:0] y
    );
    reg [2:0]y;
    always@(en,a,b)
    begin
    if (en==0)
        y=3'b0;
    else
    case({a,b})
    4'b0000:y=3'b001;
    4'b0001:y=3'b010;
    4'b0010:y=3'b010;
    4'b0011:y=3'b010;
    4'b0100:y=3'b100;
    4'b0101:y=3'b001;
    4'b0110:y=3'b010;
    4'b0111:y=3'b010;
    4'b1000:y=3'b100;
    4'b1001:y=3'b100;
    4'b1010:y=3'b001;
    4'b1011:y=3'b010;
    4'b1100:y=3'b100;
    4'b1101:y=3'b100;
    4'b1110:y=3'b100;
    4'b1111:y=3'b001;
    default:y=3'b000;
    endcase
    end
endmodule

3.STRUCTURAL MODELLING
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//
// Company:
// Engineer:
//
// Create Date: 08.09.2017 11:50:16
// Design Name:
// Module Name: twobit_Comp2
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
//
```

```
module twobit_Comp2(
    input [1:0] a,
    input [1:0] b,
    input en,
    output [2:0] y
    );
    wire [13:0]x;
    not z1(x[0],a[0]);
    not z2(x[1],a[1]);
    not z3(x[2],b[0]);
    not z4(x[3],b[1]);
    and z5(x[4],b[1],x[1]);
    and z6(x[5],b[1],b[0],x[0]);
    and z7(x[6],b[0],x[1],x[0]);
    or z8(x[7],x[4],x[5],x[6]);
    and z9(y[1],x[7],en);
    xnor q1(x[8],a[0],b[0]);
    xnor q2(x[9],a[1],b[1]);
    and q3(y[0],x[8],x[9],en);
    and w1(x[10],x[3],a[1]);
    and w2(x[11],a[0],x[3],x[2]);
    and w3(x[12],a[1],a[0],x[2]);
    or w4(x[13],x[10],x[11],x[12]);
    and w5(y[2],en,x[13]);
endmodule
```

SIMULATION RESULT



B.N BIT COMPARATOR

1.DATAFLOW MODELLING

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//
// Company:
// Engineer:
//
```

```
// Create Date: 08.09.2017 12:19:00
// Design Name:
// Module Name: nbit_Comp1
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
//


module nbit_Comp1(a,b,en,y
    );
    parameter n=4;
        input [n-1:0]a,b;
        input en;
        output  [2:0]y;
        assign y=(en==0)?3'b000:(a>b)?3'b100:(a<b)?3'b010:3'b001;

endmodule

2.BEHAVIOURAL MODELLING
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//
// Company:
// Engineer:
//
// Create Date: 08.09.2017 11:19:05
// Design Name:
// Module Name: nbit_Comp
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
//


module nbit_Comp(a,b,en,g,e,l);
parameter n=3;
input [n-1:0]a,b;
input en;
output reg g,e,l;

always@(en,a,b)
begin
    if(!en)
    {g,e,l}=3'bzzz;
    else
```

```
    begin
        if(a>b)
        {g,e,l}=3'b100;
        else if(a<b)
        {g,e,l}=3'b001;
        else
        {g,e,l}=3'b010;

        end
        end

endmodule
```

SIMULATION RESULT

EXPERIMENT NO.8: FULL ADDER

VERILOG CODE

1.DATAFLOW MODELLING

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//
// Company:
// Engineer:
//
// Create Date: 01.09.2017 14:31:07
// Design Name:
// Module Name: fulladder
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
//


module FullAdder(a,b,cin,en,sum,carry);
input a;
input b;
input cin;
input en;
output sum,carry;
assign sum= en&(a^b^cin);
assign carry= en&((a&b)|(b&cin)|(cin&a));
endmodule
```

2.BEHAVIOURAL MODELLING

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//
// Company:
// Engineer:
//
// Create Date: 15.09.2017 10:43:58
// Design Name:
// Module Name: FullAdder1
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
//
```

```
module FullAdder1(a,b,cin,en,sum,carry);
input a;
input b;
input cin;
input en;
output reg sum,carry;
always@(a,b,cin,en)
begin
if(!en)
begin
sum=0;
carry=0;
end
else
case({a,b,cin})
3'b000:{sum,carry}=2'b00;
3'b001:{sum,carry}=2'b10;
3'b010:{sum,carry}=2'b10;
3'b011:{sum,carry}=2'b01;
3'b100:{sum,carry}=2'b10;
3'b101:{sum,carry}=2'b01;
3'b110:{sum,carry}=2'b01;
3'b111:{sum,carry}=2'b11;
default:{sum,carry}=2'b00;
endcase
end
endmodule
```

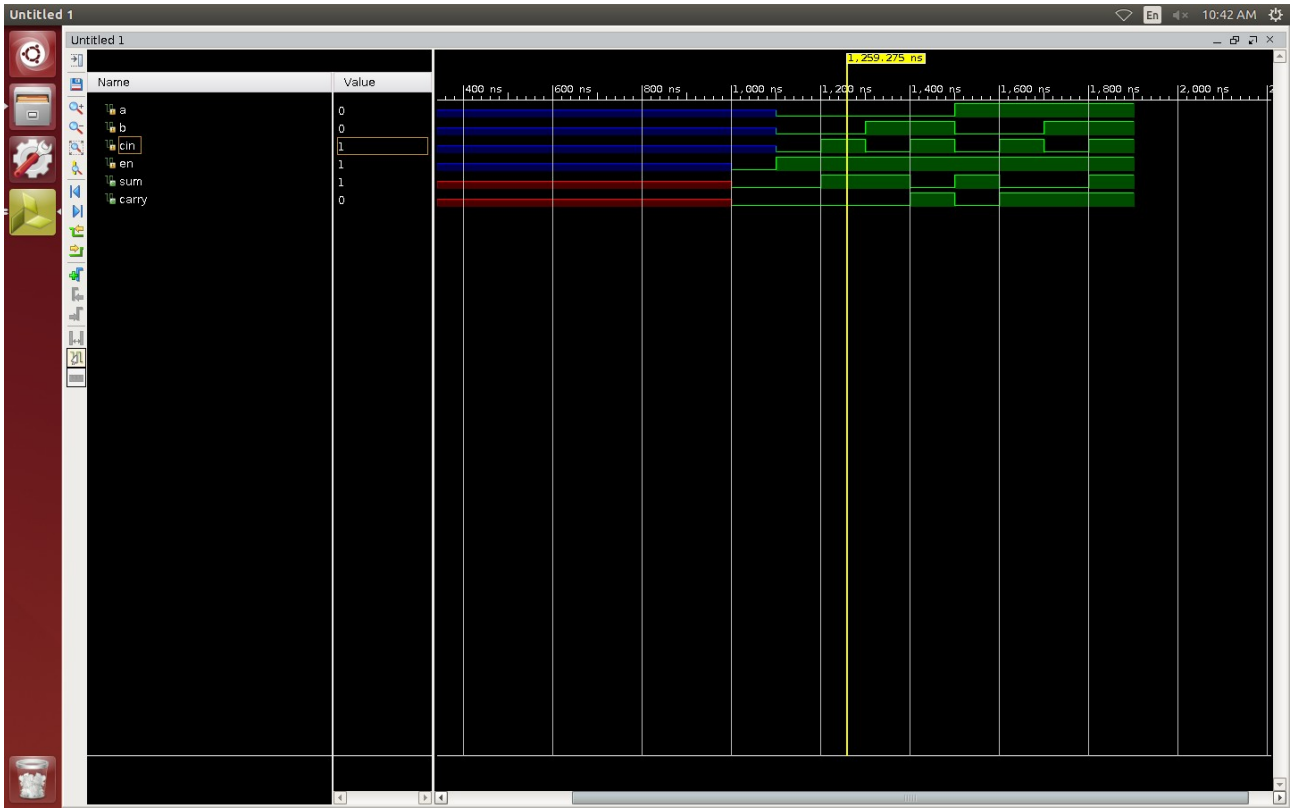
3.STRUCTURAL MODELLING

A.USING GATE PRIMITIVES

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//
// Company:
// Engineer:
//
// Create Date: 15.09.2017 10:56:18
// Design Name:
// Module Name: FullAdder2
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
//


module FullAdder2(en,a,b,cin,sum,cout);
input en,a,b,cin;
output sum,cout;
wire [2:0]c;
wire temp;
xor x1(temp,a,b,cin);
and x3(sum,temp,en);
and a1(c[0],a,b);
and a2(c[2],cin,a);
```

```
or o1(temp,c[0],c[1],c[2]);
and x2(cout,temp,en);
endmodule

B.USING HALF ADDER
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//
// Company:
// Engineer:
//
// Create Date: 15.09.2017 11:01:14
// Design Name:
// Module Name: FullAdder3
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
//


module FullAdder3(en,a,b,cin,sum,cout);
input en,a,b,cin;
output sum,cout;
wire s1,c1,c2;
HalfAdder h1(s1,c1,a,b);
HalfAdder h2(sum,c2,s1,cin);
or o1(cout,c1,c2);
endmodule



module HalfAdder(sum,cout,a,b);
input a,b;
output sum,cout;
xor x1(sum,a,b);
and x2(cout,a,b);
endmodule
```

SIMULATION RESULT



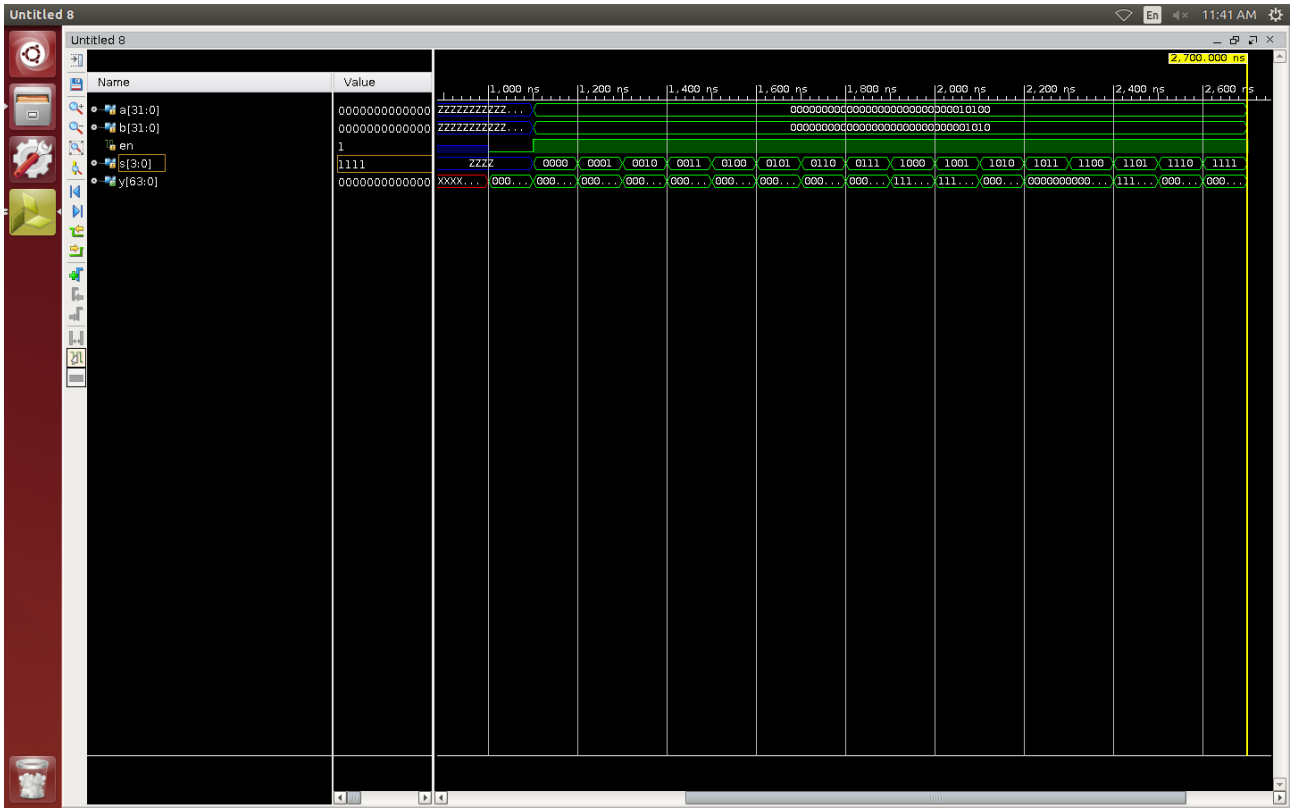
EXPERIMENT NO.9: 32 BIT ARITHMETIC LOGIC UNIT

VERILOG CODE

1.BEHAVIOURAL MODELLING

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//
// Company:
// Engineer:
//
// Create Date: 15.09.2017 11:19:45
// Design Name:
// Module Name: ALU
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
//


module ALU(
    input [31:0] a,
    input [31:0] b,
    input en,
    input [3:0] s,
    output [63:0] y
    );
    reg [63:0]y;
    always@(en,a,b,s)
    begin
    if(!en)
    y=0;
    else
    case(s)
    4'b0000:y=a+b;
    4'b0001:y=a-b;
    4'b0010:y=a*b;
    4'b0011:y=a/b;
    4'b0100:y=a+1;
    4'b0101:y=b+1;
    4'b0110:y=a-1;
    4'b0111:y=b-1;
    4'b1000:y=~a;
    4'b1001:y=~b;
    4'b1010:y=a&b;
    4'b1011:y=a|b;
    4'b1100:y=a^b;
    4'b1101:y=~(a&b);
    4'b1110:y=a>>1;
    4'b1111:y=b<<1;
    endcase
    end
endmodule
```

SIMULATION RESULT



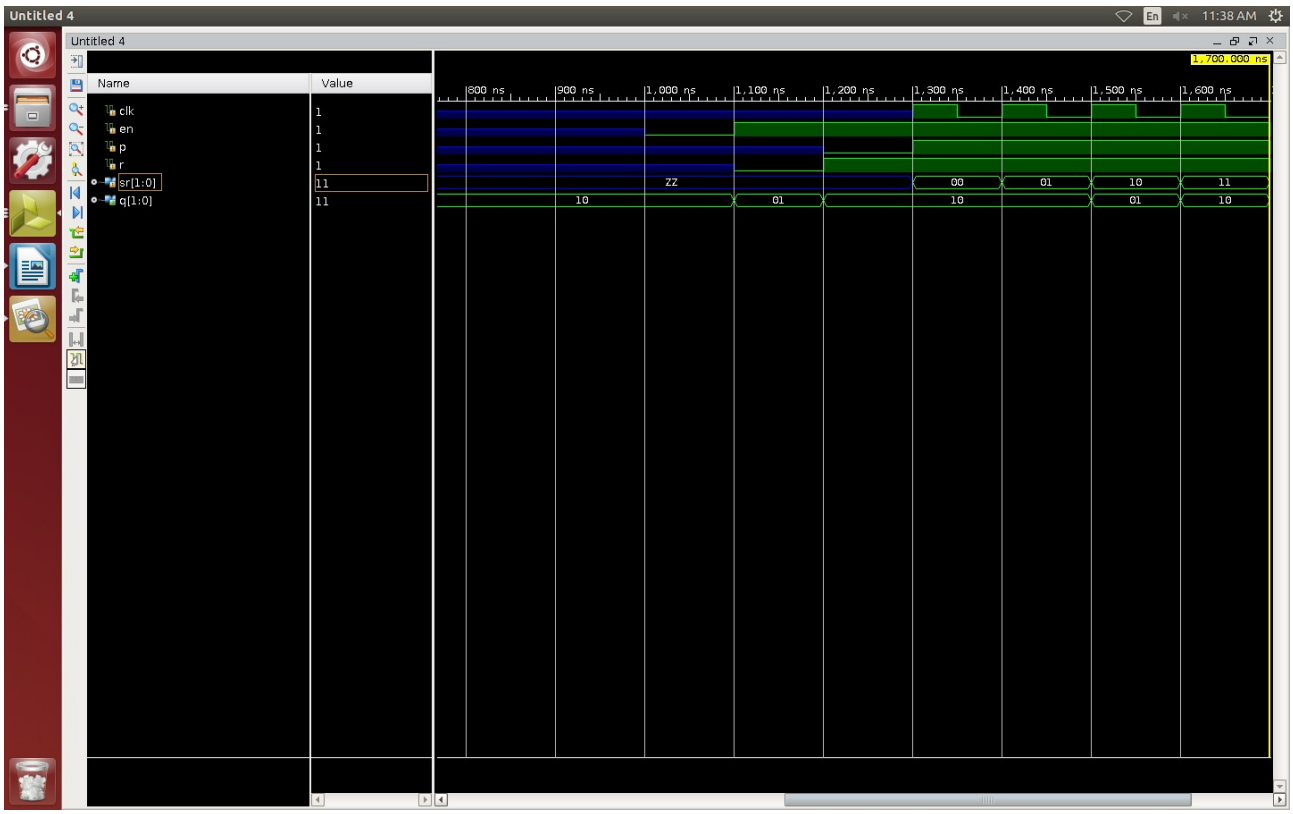
EXPERIMENT NO.10: SR FLIP FLOP

VERILOG CODE

1.BEHAVIOURAL MODELLING

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//
// Company:
// Engineer:
//
// Create Date: 22.09.2017 16:15:39
// Design Name:
// Module Name: sr_bh
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
//


module SRFF(q,sr,clk,p,r,en

    );
    input clk,en,p,r;
    input [1:0]sr;
    output reg [1:0]q;

    initial
      begin
        q=2'b10;
        end
    always@(posedge clk, negedge p,negedge r,posedge en)
    begin
    if (!en)
        q=2'b0;
    else if(!r)
        q=2'b01;
    else if(!p)
        q=2'b10;
    else
    case(sr)
    2'b00:q=q;
    2'b01:q=2'b01;
    2'b10:q=2'b10;
    2'b11:q=2'b11;//invalid state
    default:q=2'b10;
    endcase
    end

endmodule
```

SIMULATION RESULT


Untitled 4
11:38 AM
Untitled 4
Name
Value
clk
en
p
r
sr[1:0]
q[1:0]
1
1
1
1
11
11
1,700.000 ns
800 ns
900 ns
1,000 ns
1,100 ns
1,200 ns
1,300 ns
1,400 ns
1,500 ns
1,600 ns
ZZ
00
01
10
11
10
01
10
01
10

EXPERIMENT NO.11: D FLIP FLOP

VERILOG CODE

1.BEHAVIOURAL MODELLING

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//
// Company:
// Engineer:
//
// Create Date: 22.09.2017 10:36:29
// Design Name:
// Module Name: DFF
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
//


module DFF(cen,reset,preset,clk,d,q,qbar);
input d,cen,reset,preset,clk;
output reg q,qbar;
always@(cen,posedge clk,negedge preset,negedge reset)
begin
if(!e)
begin
q=0;
qbar=0;
end
else if(!reset)
begin
q=0;
qbar=1;
end
else if(!preset)
begin
q=1;
qbar=0;
end
else
case({d})
1'b0:{q,qbar}=2'b10;
1'b1:{q,qbar}=2'b11;
default:{q,qbar}=2'b00;
endcase
end
endmodule
```

SIMULATION RESULT

EXPERIMENT NO.12: T FLIP FLOP

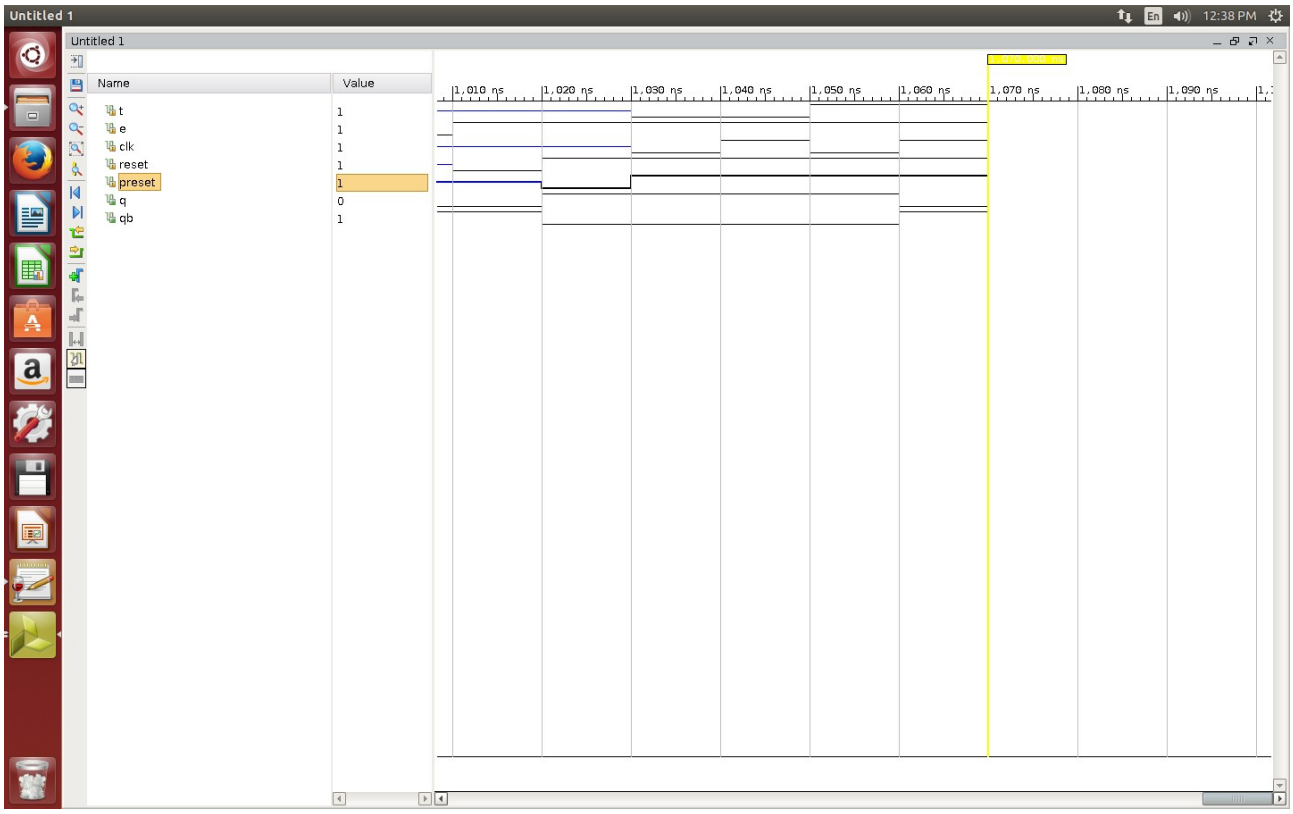
VERILOG CODE

1.BEHAVIOURAL MODELLING

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//
// Company:
// Engineer:
//
// Create Date: 22.09.2017 12:36:38
// Design Name:
// Module Name: tff
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
//


module tff(
    input t,
    input e,
    input clk,
    input reset,
    input preset,
    output q,
    output qb
    );

    reg q,qb;

        initial
    begin
    q=0;
    qb=1;
    end
    always@(posedge e,negedge preset,negedge reset,posedge clk)
    begin
    if(!e)
      begin
      q=0;
      qb=0;
      end
    else if(!reset)
      begin
      q=0;
      qb=1;
      end
    else if(!preset)
      begin
      q=1;
      qb=0;
      end
    else
```

```
    case(t)
    1'b0:begin q=q; qb=qb; end
    1'b1:begin q=~q;qb=~qb; end
    endcase
    end
endmodule
```

SIMULATION RESULT



EXPERIMENT NO.13: JK FLIP FLOP

VERILOG CODE

1.BEHAVIOURAL MODELLING

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//
// Company:
// Engineer:
//
// Create Date: 13.10.2017 11:08:38
// Design Name:
// Module Name: jkflipflop
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
//

module jkflipflop(
    input cen,
    input r,
    input p,
    input clk,
    input [1:0]jk,
    output q,
    output qbar
    );
    reg q,qbar;
    initial
    begin
    q=1'b0;
    qbar=1'b1;
    end
    always@(posedge cen,negedge r,negedge p,posedge clk)
    begin
    if(!cen)
    begin
    q=0;
    qbar=1;
    end
     else if(!r)
       begin
       q=0;
       qbar=1;
       end
        else if(!p)
         begin
          q=1;
          qbar=0;
          end
          else
          begin
          case(jk)
          2'b00:begin q=q;qbar=qbar;end
```

```
          2'b01:begin q=0;qbar=1;end
          2'b10:begin q=1;qbar=0;end
          2'b11:begin q=q;qbar=qbar;end
          default:begin q=0;qbar=1;end
          endcase
          end
          end

endmodule
```

SIMULATION RESULT

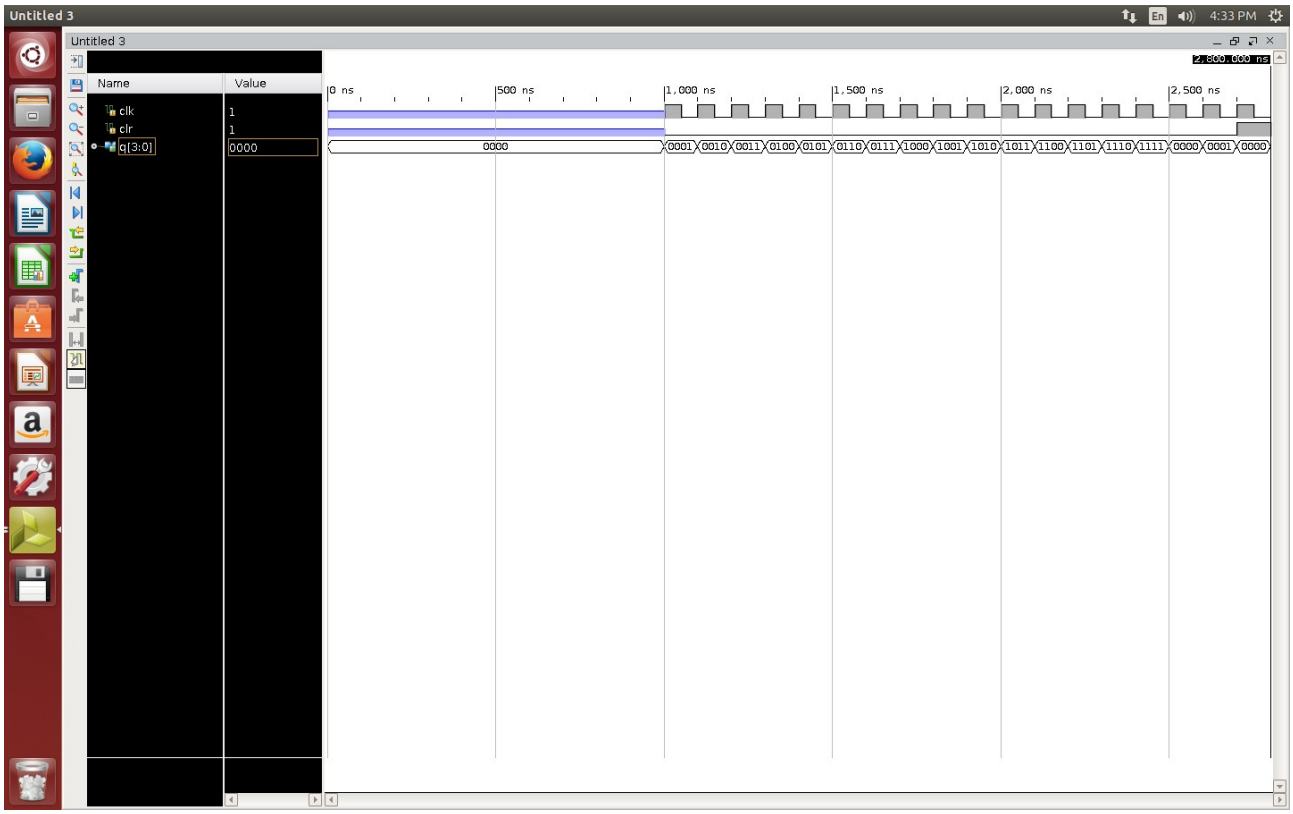
EXPERIMENT NO.14: 4 BIT BINARY SYNCHRONOUS COUNTER WITH ASYNCHRONOUS CONTROL SIGNAL

VERILOG CODE

1.BEHAVIOURAL MODELLING

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//
// Company:
// Engineer:
//
// Create Date: 13.10.2017 11:02:00
// Design Name:
// Module Name: fourbin_async
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
//


module fourbin_async(q,clk,clr

    );
    input clk,clr;
    output reg [3:0]q;
    initial
    q=4'b0;
    always@(posedge clk,posedge clr)
    if(clr==1)
    q=0;
    else
    q=q+1;
endmodule
```

SIMULATION RESULT



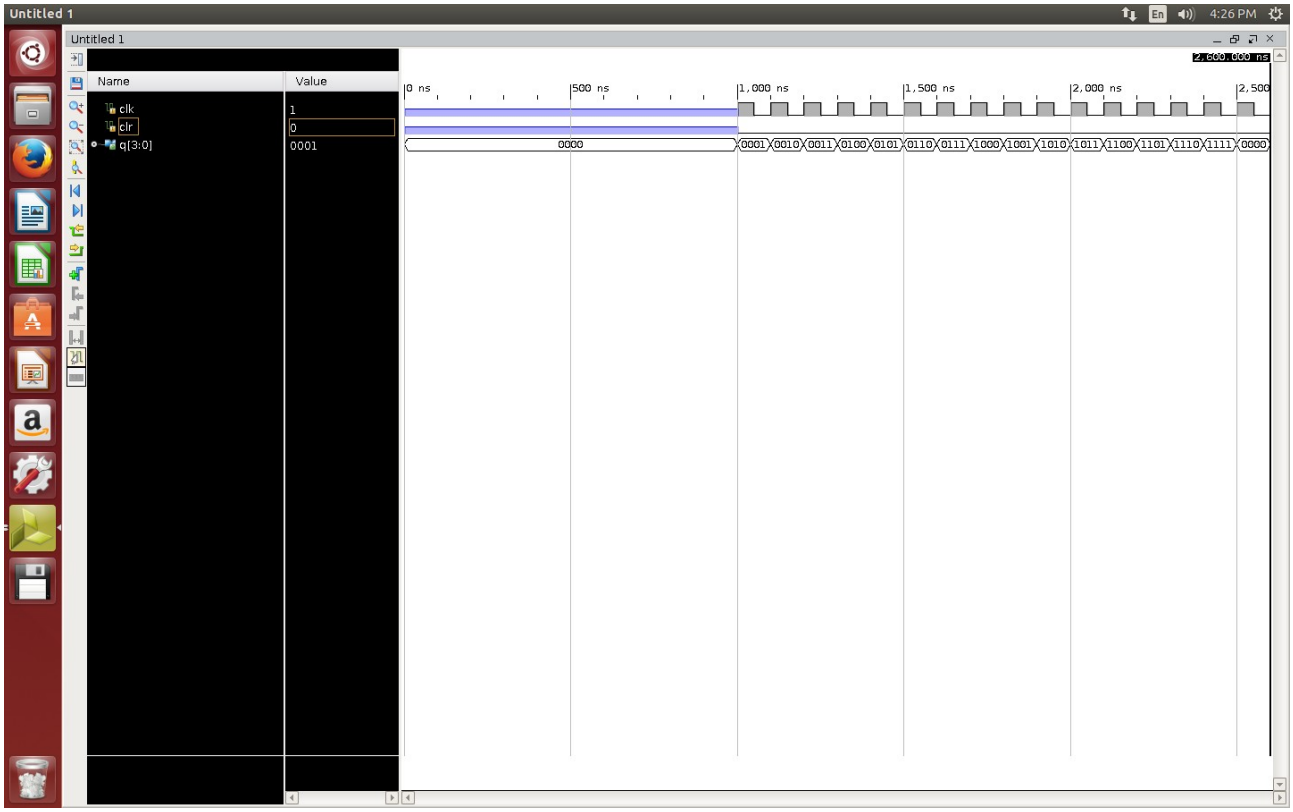
EXPERIMENT NO.15: 4 BIT BINARY SYNCHRONOUS COUNTER WITH SYNCHRONOUS CONTROL SIGNAL

VERILOG CODE

1.BEHAVIOURAL MODELLING

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//
// Company:
// Engineer:
//
// Create Date: 13.10.2017 12:00:04
// Design Name:
// Module Name: fourbin_sync
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
//


module fourbin_sync(clk,clr,q

    );
    input clk,clr;
    output [3:0]q;
    reg [3:0]q;
    reg [30:0]sclk;
    always@(posedge clk)
    sclk=sclk+1;

    initial
            q=4'b0;
    always@(posedge sclk[25])
    begin

    if(clr==1)
    q=4'b0;
    else
    q=q+1;
    end

endmodule
```

SIMULATION RESULT



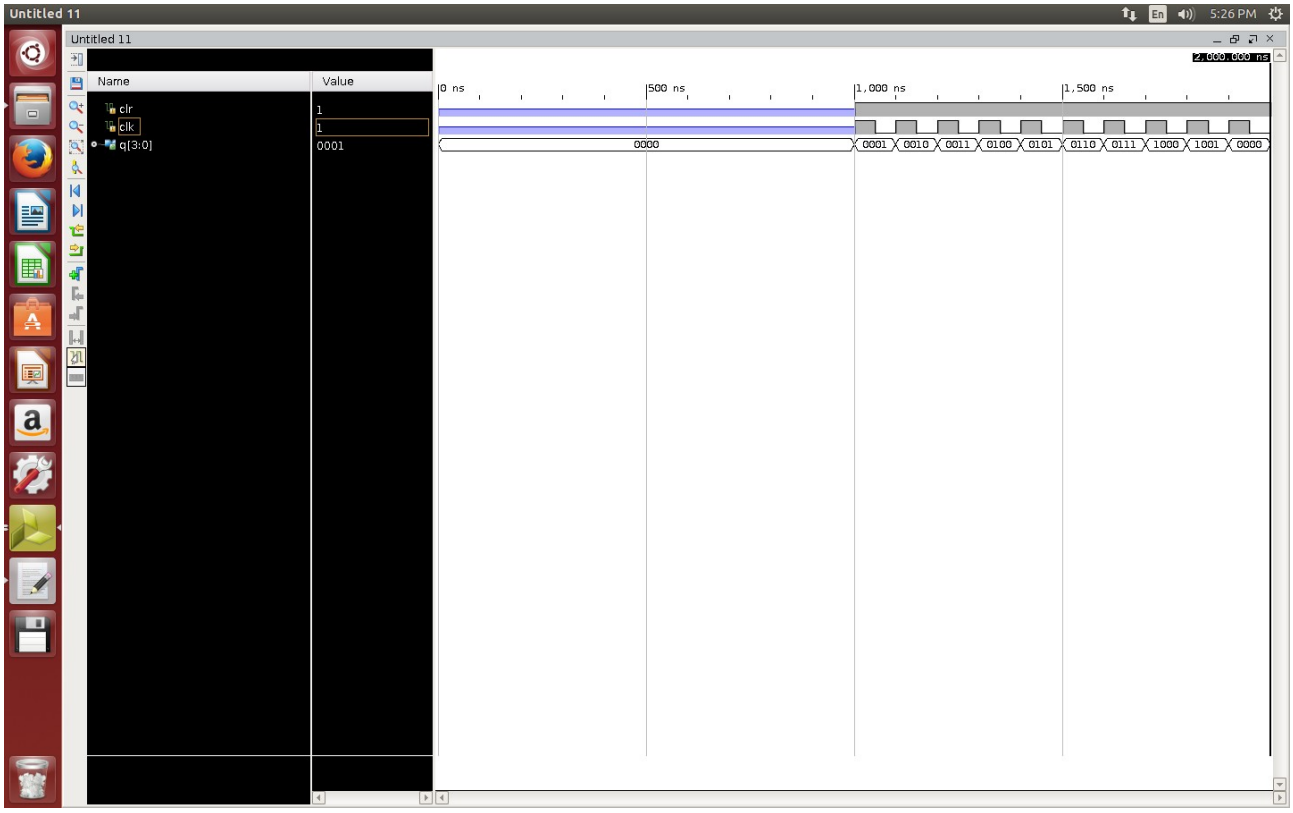
EXPERIMENT NO.16: BCD SYNCHRONOUS COUNTER WITH SYNCHRONOUS CONTROL SIGNAL

VERILOG CODE

1.BEHAVIOURAL MODELLING

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//
// Company:
// Engineer:
//
// Create Date: 13.10.2017 16:34:44
// Design Name:
// Module Name: bcd_syn
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
//


module bcd_syn(q,clr,clk

    );
    input clr,clk;
    output [3:0]q;
    reg [3:0]q;
    //reg [30:0]sclk;
    //always@(posedge clk)
    //sclk=sclk+1;

    initial q=0;
    always@(posedge clk)
    begin
    if(clr==0)
    q=0;
    else if(q>4'd8)
    q=0;
    else
    q=q+1;
    end

endmodule
```

SIMULATION RESULT



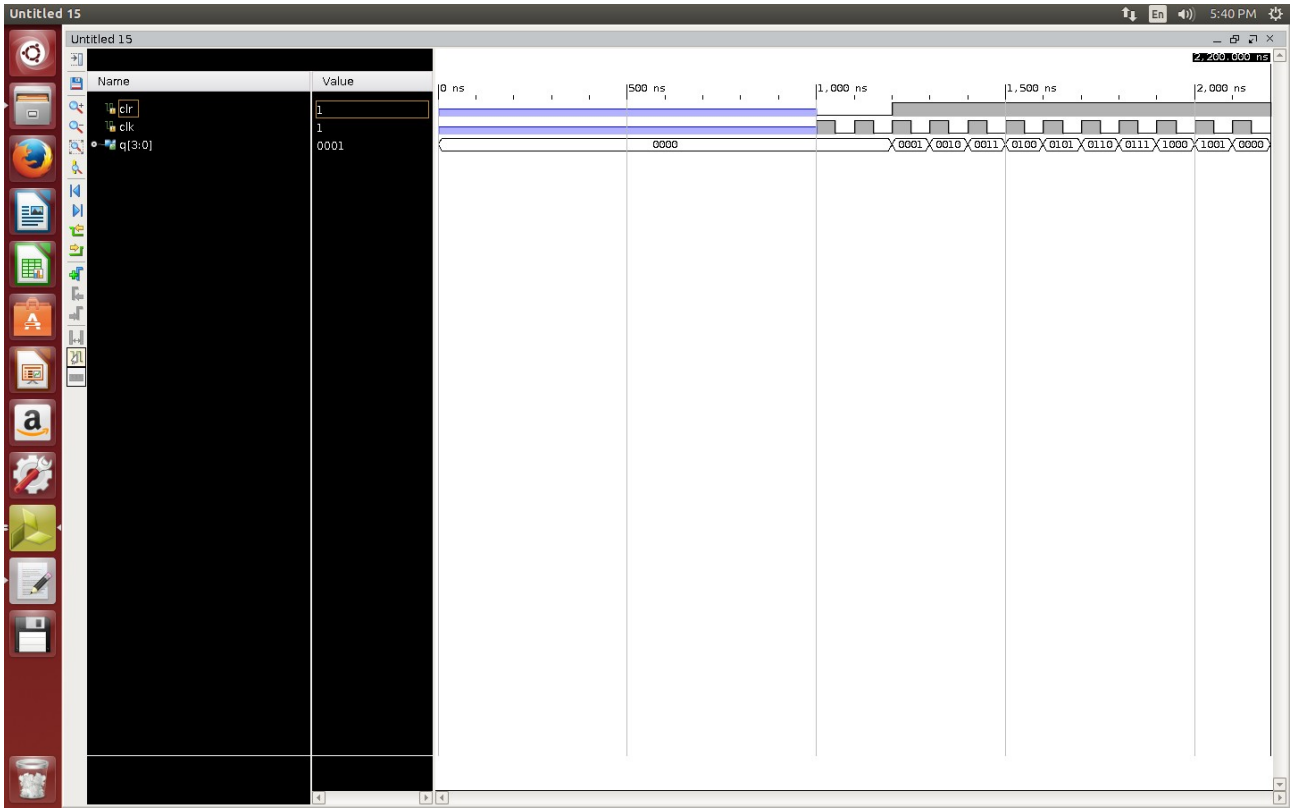
EXPERIMENT NO.17: BCD SYNCHRONOUS COUNTER WITH ASYNCHRONOUS CONTROL SIGNAL

VERILOG CODE

1.BEHAVIOURAL MODELLING

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//
// Company:
// Engineer:
//
// Create Date: 13.10.2017 17:38:03
// Design Name:
// Module Name: bcd_as
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
//


module bcd_as(q,clk,clr

    );
     input clr,clk;
       output [3:0]q;
       reg [3:0]q;
       //reg [30:0]sclk;
       //always@(posedge clk)
       //sclk=sclk+1;

       initial q=0;
       always@(posedge clk,posedge clr)
       begin
       if(clr==0)
       q=0;
       else if(q>4'd8)
       q=0;
       else
       q=q+1;
       end



endmodule
```

SIMULATION RESULT



EXPERIMENT NO.18:

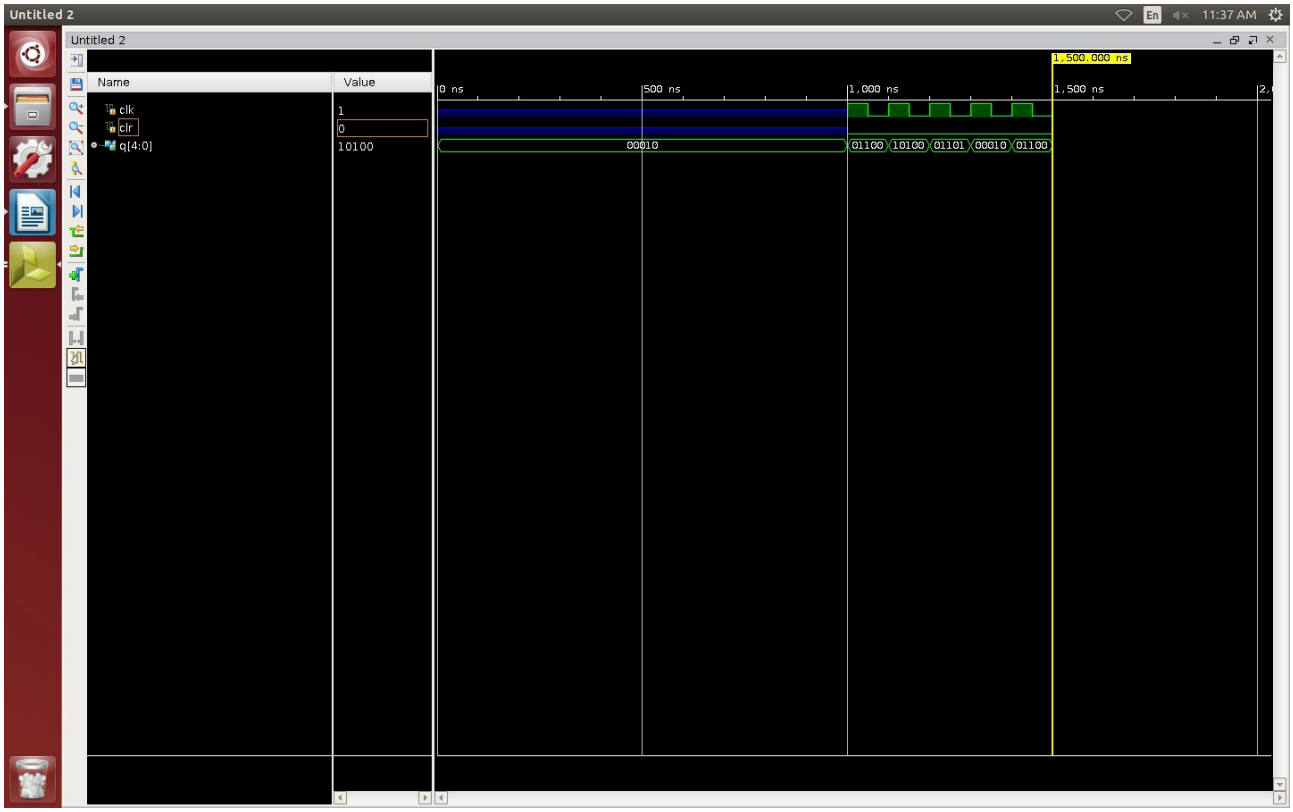
A) ARBITRARY COUNTER 1

VERILOG CODE

1.BEHAVIOURAL MODELLING

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//
// Company:
// Engineer:
//
// Create Date: 13.10.2017 17:27:54
// Design Name:
// Module Name: arb_count
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
//


module arbcounter_one(q, clk,clr

    );
    input clk,clr;
    output [4:0]q;

        reg [4:0]q;
         initial
                        q=5'd2;
        //reg [30:0]sclk;
        always@(posedge clk)
        //sclk=sclk+1;


        //always@(posedge sclk[25])
        begin

        if(clr==1)
        q=5'd2;
        else
        case(q)
        5'd2:q=5'd12;
        5'd12:q=5'd20;
        5'd20:q=5'd13;
        5'd13:q=5'd2;
        default:q=5'd2;
        endcase
        end
endmodule
```

SIMULATION RESULT



EXPERIMENT NO.19: SEVEN SEGMENT DISPLAY

1.VERILOG CODE

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//
// Company:
// Engineer:
//
// Create Date: 27.10.2017 16:17:06
// Design Name:
// Module Name: sevenseg
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
//


module sevenseg(col,an,dp,message,row,clk);
output reg [3:0] col,an;
output reg dp;
output reg [6:0] message;
input [3:0] row;
input clk;
reg [31:0] clk_div;
reg [1:0] cnt_2bit=2'b00;
always@ (posedge clk)
clk_div=clk_div+1;
always@ (posedge clk_div[26])
cnt_2bit=cnt_2bit+1;
always@ (cnt_2bit)
begin
case(cnt_2bit)
2'b00:col=4'b1110;
2'b01:col=4'b1101;
2'b10:col=4'b1011;
2'b11:col=4'b0111;
endcase
end
always@(row,col)
begin
if(col==4'b1110)
case(row)
4'b1110:message=7'b1111001;
4'b1101:message=7'b0011001;
4'b1011:message=7'b1111000;
4'b0111:message=7'b1000000;
default:message=7'b1111111;
endcase
else if(col==4'b1101)
case(row)
4'b1110:message=7'b0100100;
4'b1101:message=7'b0010010;
4'b1011:message=7'b0000000;
```

```
4'b0111:message=7'b0001110;
default:message=7'b1111111;
endcase
else if(col==4'b1011)
case(row)
4'b1110:message=7'b0110000;
4'b1101:message=7'b0000010;
4'b1011:message=7'b0011000;
4'b0111:message=7'b0000110;
default:message=7'b1111111;
endcase
else
case(row)
4'b1110:message=7'b0001000;
4'b1101:message=7'b0000011;
4'b1011:message=7'b1000110;
4'b0111:message=7'b0100001;
default:message=7'b1111111;
endcase
//else
//case(row)
//default:message=7'b1111111;
//endcase
dp=1'b0;
an=4'b0000;
end
endmodule
```

2.VHDL CODE

```
----------------------------------------------------------------------------------
--
-- Company:
-- Engineer:
--
-- Create Date: 27.10.2017 17:04:50
-- Design Name:
-- Module Name: sevenseg_display_vhdl - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------------
--


library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;
```

```
entity sevenseg_display_vhdl is
 Port (col: inout std_logic_vector(3 downto 0);
       an: out std_logic_vector(3 downto 0);
       dp: out std_logic;
       msg: out std_logic_vector(6 downto 0);
       row: in std_logic_vector(3 downto 0);
       clk: in std_logic );
end sevenseg_display_vhdl;

architecture Behavioral of sevenseg_display_vhdl is
signal clk_div: std_logic_vector(31 downto 0);
signal cnt_2bit: std_logic_vector(1 downto 0);
begin
process(clk)
begin
if clk='1' and clk'event then
clk_div <= clk_div+1;
end if;
end process;
process(clk_div(25))
begin
if clk_div(25)='1' and clk_div(25)'event then
cnt_2bit <= cnt_2bit+1;
end if;
end process;
process(cnt_2bit)
begin
case(cnt_2bit) is
when "00" => col <= "1110";
when "01" => col <= "1101";
when "10" => col <= "1011";
when "11" => col <= "0111";
when others => null;
end case;
end process;
process(row,col)
begin
case(col) is
when "1110" =>
case(row) is
when "1110" => msg <= "1111001";
when "1101" => msg <= "0011001";
when "1011" => msg <= "1111000";
when "0111" => msg <= "1000000";
when others => msg <= "1111111";
end case;
when "1101" =>
case(row) is
when "1110" => msg <= "0100100";
when "1101" => msg <= "0010010";
when "1011" => msg <= "0000000";
when "0111" => msg <= "0001110";
when others => msg <= "1111111";
end case;
when "1011" =>
case(row) is
when "1110" => msg <= "0110000";
when "1101" => msg <= "0000010";
when "1011" => msg <= "0011000";
when "0111" => msg <= "0000110";
when others => msg <= "1111111";
end case;
when "0111" =>
```

```
case(row) is
when "1110" => msg <= "0001000";
when "1101" => msg <= "0000011";
when "1011" => msg <= "1000110";
when "0111" => msg <= "0100001";
when others => msg <= "1111111";
end case;
when others => null;
end case;
end process;
dp <= '0';
an <= "0000";
end Behavioral;
```

XDC FILE

```
## This file is a general .xdc for the Basys3 rev B board
## To use it in a project:
## - uncomment the lines corresponding to used pins
## - rename the used ports (in each line, after get_ports) according to the top
level signal names in the project

## Clock signal
set_property PACKAGE_PIN W5 [get_ports clk]

      set_property IOSTANDARD LVCMOS33 [get_ports clk]
      #create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5}
[get_ports clk]

## Switches
#set_property PACKAGE_PIN V17 [get_ports {sw[0]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[0]}]
#set_property PACKAGE_PIN V16 [get_ports {sw[1]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[1]}]
#set_property PACKAGE_PIN W16 [get_ports {sw[2]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[2]}]
#set_property PACKAGE_PIN W17 [get_ports {sw[3]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[3]}]
#set_property PACKAGE_PIN W15 [get_ports {sw[4]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[4]}]
#set_property PACKAGE_PIN V15 [get_ports {sw[5]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[5]}]
#set_property PACKAGE_PIN W14 [get_ports {sw[6]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[6]}]
#set_property PACKAGE_PIN W13 [get_ports {sw[7]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[7]}]
#set_property PACKAGE_PIN V2 [get_ports {sw[8]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[8]}]
#set_property PACKAGE_PIN T3 [get_ports {sw[9]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[9]}]
#set_property PACKAGE_PIN T2 [get_ports {sw[10]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[10]}]
#set_property PACKAGE_PIN R3 [get_ports {sw[11]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[11]}]
#set_property PACKAGE_PIN W2 [get_ports {sw[12]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[12]}]
#set_property PACKAGE_PIN U1 [get_ports {sw[13]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[13]}]
#set_property PACKAGE_PIN T1 [get_ports {sw[14]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[14]}]
#set_property PACKAGE_PIN R2 [get_ports {sw[15]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[15]}]
```

```
## LEDs
#set_property PACKAGE_PIN U16 [get_ports {led[0]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[0]}]
#set_property PACKAGE_PIN E19 [get_ports {led[1]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[1]}]
#set_property PACKAGE_PIN U19 [get_ports {led[2]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[2]}]
#set_property PACKAGE_PIN V19 [get_ports {led[3]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[3]}]
#set_property PACKAGE_PIN W18 [get_ports {led[4]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[4]}]
#set_property PACKAGE_PIN U15 [get_ports {led[5]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[5]}]
#set_property PACKAGE_PIN U14 [get_ports {led[6]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[6]}]
#set_property PACKAGE_PIN V14 [get_ports {led[7]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[7]}]
#set_property PACKAGE_PIN V13 [get_ports {led[8]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[8]}]
#set_property PACKAGE_PIN V3 [get_ports {led[9]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[9]}]
#set_property PACKAGE_PIN W3 [get_ports {led[10]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[10]}]
#set_property PACKAGE_PIN U3 [get_ports {led[11]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[11]}]
#set_property PACKAGE_PIN P3 [get_ports {led[12]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[12]}]
#set_property PACKAGE_PIN N3 [get_ports {led[13]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[13]}]
#set_property PACKAGE_PIN P1 [get_ports {led[14]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[14]}]
#set_property PACKAGE_PIN L1 [get_ports {led[15]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[15]}]


##7 segment display
set_property PACKAGE_PIN W7 [get_ports {msg[0]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {msg[0]}]
set_property PACKAGE_PIN W6 [get_ports {msg[1]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {msg[1]}]
set_property PACKAGE_PIN U8 [get_ports {msg[2]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {msg[2]}]
set_property PACKAGE_PIN V8 [get_ports {msg[3]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {msg[3]}]
set_property PACKAGE_PIN U5 [get_ports {msg[4]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {msg[4]}]
set_property PACKAGE_PIN V5 [get_ports {msg[5]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {msg[5]}]
set_property PACKAGE_PIN U7 [get_ports {msg[6]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {msg[6]}]

set_property PACKAGE_PIN V7 [get_ports dp]

      set_property IOSTANDARD LVCMOS33 [get_ports dp]

set_property PACKAGE_PIN U2 [get_ports {an[0]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {an[0]}]
set_property PACKAGE_PIN U4 [get_ports {an[1]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {an[1]}]
set_property PACKAGE_PIN V4 [get_ports {an[2]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {an[2]}]
set_property PACKAGE_PIN W4 [get_ports {an[3]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {an[3]}]
```

```
##Buttons
#set_property PACKAGE_PIN U18 [get_ports btnC]
      #set_property IOSTANDARD LVCMOS33 [get_ports btnC]
#set_property PACKAGE_PIN T18 [get_ports btnU]
      #set_property IOSTANDARD LVCMOS33 [get_ports btnU]
#set_property PACKAGE_PIN W19 [get_ports btnL]
      #set_property IOSTANDARD LVCMOS33 [get_ports btnL]
#set_property PACKAGE_PIN T17 [get_ports btnR]
      #set_property IOSTANDARD LVCMOS33 [get_ports btnR]
#set_property PACKAGE_PIN U17 [get_ports btnD]
      #set_property IOSTANDARD LVCMOS33 [get_ports btnD]



##Pmod Header JA
##Sch name = JA1
#set_property PACKAGE_PIN J1 [get_ports {JA[0]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JA[0]}]
##Sch name = JA2
#set_property PACKAGE_PIN L2 [get_ports {JA[1]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JA[1]}]
##Sch name = JA3
#set_property PACKAGE_PIN J2 [get_ports {JA[2]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JA[2]}]
##Sch name = JA4
#set_property PACKAGE_PIN G2 [get_ports {JA[3]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JA[3]}]
##Sch name = JA7
#set_property PACKAGE_PIN H1 [get_ports {JA[4]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JA[4]}]
##Sch name = JA8
#set_property PACKAGE_PIN K2 [get_ports {JA[5]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JA[5]}]
##Sch name = JA9
#set_property PACKAGE_PIN H2 [get_ports {JA[6]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JA[6]}]
##Sch name = JA10
#set_property PACKAGE_PIN G3 [get_ports {JA[7]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JA[7]}]



##Pmod Header JB
##Sch name = JB1
#set_property PACKAGE_PIN A14 [get_ports {JB[0]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JB[0]}]
##Sch name = JB2
#set_property PACKAGE_PIN A16 [get_ports {JB[1]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JB[1]}]
##Sch name = JB3
#set_property PACKAGE_PIN B15 [get_ports {JB[2]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JB[2]}]
##Sch name = JB4
#set_property PACKAGE_PIN B16 [get_ports {JB[3]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JB[3]}]
##Sch name = JB7
#set_property PACKAGE_PIN A15 [get_ports {JB[4]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JB[4]}]
##Sch name = JB8
#set_property PACKAGE_PIN A17 [get_ports {JB[5]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JB[5]}]
##Sch name = JB9
#set_property PACKAGE_PIN C15 [get_ports {JB[6]}]
```

```
      #set_property IOSTANDARD LVCMOS33 [get_ports {JB[6]}]
##Sch name = JB10
#set_property PACKAGE_PIN C16 [get_ports {JB[7]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JB[7]}]



##Pmod Header JC
##Sch name = JC1
set_property PACKAGE_PIN K17 [get_ports {col[3]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {col[3]}]
##Sch name = JC2
set_property PACKAGE_PIN M18 [get_ports {col[2]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {col[2]}]
##Sch name = JC3
set_property PACKAGE_PIN N17 [get_ports {col[1]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {col[1]}]
##Sch name = JC4
set_property PACKAGE_PIN P18 [get_ports {col[0]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {col[0]}]
##Sch name = JC7
set_property PACKAGE_PIN L17 [get_ports {row[3]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {row[3]}]
##Sch name = JC8
set_property PACKAGE_PIN M19 [get_ports {row[2]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {row[2]}]
##Sch name = JC9
set_property PACKAGE_PIN P17 [get_ports {row[1]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {row[1]}]
##Sch name = JC10
set_property PACKAGE_PIN R18 [get_ports {row[0]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {row[0]}]


##Pmod Header JXADC
##Sch name = XA1_P
#set_property PACKAGE_PIN J3 [get_ports {JXADC[0]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[0]}]
##Sch name = XA2_P
#set_property PACKAGE_PIN L3 [get_ports {JXADC[1]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[1]}]
##Sch name = XA3_P
#set_property PACKAGE_PIN M2 [get_ports {JXADC[2]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[2]}]
##Sch name = XA4_P
#set_property PACKAGE_PIN N2 [get_ports {JXADC[3]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[3]}]
##Sch name = XA1_N
#set_property PACKAGE_PIN K3 [get_ports {JXADC[4]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[4]}]
##Sch name = XA2_N
#set_property PACKAGE_PIN M3 [get_ports {JXADC[5]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[5]}]
##Sch name = XA3_N
#set_property PACKAGE_PIN M1 [get_ports {JXADC[6]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[6]}]
##Sch name = XA4_N
#set_property PACKAGE_PIN N1 [get_ports {JXADC[7]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[7]}]



##VGA Connector
#set_property PACKAGE_PIN G19 [get_ports {vgaRed[0]}]
```

```
      #set_property IOSTANDARD LVCMOS33 [get_ports {vgaRed[0]}]
#set_property PACKAGE_PIN H19 [get_ports {vgaRed[1]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {vgaRed[1]}]
#set_property PACKAGE_PIN J19 [get_ports {vgaRed[2]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {vgaRed[2]}]
#set_property PACKAGE_PIN N19 [get_ports {vgaRed[3]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {vgaRed[3]}]
#set_property PACKAGE_PIN N18 [get_ports {vgaBlue[0]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {vgaBlue[0]}]
#set_property PACKAGE_PIN L18 [get_ports {vgaBlue[1]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {vgaBlue[1]}]
#set_property PACKAGE_PIN K18 [get_ports {vgaBlue[2]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {vgaBlue[2]}]
#set_property PACKAGE_PIN J18 [get_ports {vgaBlue[3]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {vgaBlue[3]}]
#set_property PACKAGE_PIN J17 [get_ports {vgaGreen[0]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {vgaGreen[0]}]
#set_property PACKAGE_PIN H17 [get_ports {vgaGreen[1]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {vgaGreen[1]}]
#set_property PACKAGE_PIN G17 [get_ports {vgaGreen[2]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {vgaGreen[2]}]
#set_property PACKAGE_PIN D17 [get_ports {vgaGreen[3]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {vgaGreen[3]}]
#set_property PACKAGE_PIN P19 [get_ports Hsync]
      #set_property IOSTANDARD LVCMOS33 [get_ports Hsync]
#set_property PACKAGE_PIN R19 [get_ports Vsync]
      #set_property IOSTANDARD LVCMOS33 [get_ports Vsync]


##USB-RS232 Interface
#set_property PACKAGE_PIN B18 [get_ports RsRx]
      #set_property IOSTANDARD LVCMOS33 [get_ports RsRx]
#set_property PACKAGE_PIN A18 [get_ports RsTx]
      #set_property IOSTANDARD LVCMOS33 [get_ports RsTx]


##USB HID (PS/2)
#set_property PACKAGE_PIN C17 [get_ports PS2Clk]

      #set_property IOSTANDARD LVCMOS33 [get_ports PS2Clk]
      #set_property PULLUP true [get_ports PS2Clk]
#set_property PACKAGE_PIN B17 [get_ports PS2Data]
      #set_property IOSTANDARD LVCMOS33 [get_ports PS2Data]
      #set_property PULLUP true [get_ports PS2Data]


##Quad SPI Flash
##Note that CCLK_0 cannot be placed in 7 series devices. You can access it using
the
##STARTUPE2 primitive.
#set_property PACKAGE_PIN D18 [get_ports {QspiDB[0]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {QspiDB[0]}]
#set_property PACKAGE_PIN D19 [get_ports {QspiDB[1]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {QspiDB[1]}]
#set_property PACKAGE_PIN G18 [get_ports {QspiDB[2]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {QspiDB[2]}]
#set_property PACKAGE_PIN F18 [get_ports {QspiDB[3]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {QspiDB[3]}]
#set_property PACKAGE_PIN K19 [get_ports QspiCSn]
      #set_property IOSTANDARD LVCMOS33 [get_ports QspiCSn]
```

EXPERIMENT NO.20: FULL ADDER USING VHDL

VHDL CODE

A)

1.DATAFLOW MODELLING

```
----------------------------------------------------------------------------------
--
-- Company:
-- Engineer:
--
-- Create Date: 10/05/2017 10:05:49 PM
-- Design Name:
-- Module Name: FullAdder - DataFlow
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------------
--
-- Full Adder using Dataflow Style.

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity FullAdder is
  port (en, a, b, cin : in std_logic; sum, cout : out std_logic);
end FullAdder;

architecture DataFlow of FullAdder is

begin
    sum <= ((a xor b xor cin) and en);
    cout <= (en and ((a and b) or (a and cin) or (b and cin)));

end DataFlow;
```

2.BEHAVIOURAL MODELLING

```
----------------------------------------------------------------------------------
--
-- Company:
-- Engineer:
--
-- Create Date: 10/05/2017 10:49:30 PM
-- Design Name:
-- Module Name: FullAdder - Behavioral
```

```
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------------
--
--Full Adder using Behavioral Style.

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity FullAdder is
  port (en, a, b, cin : in std_logic; sum, cout : out std_logic);
end FullAdder;

architecture Behavioral of FullAdder is

begin
    Adder: process (en, a, b, cin)
        variable abc : std_logic_vector (2 downto 0);
    begin
        abc := (a & b & cin);

        if (en = '0')then
            sum <= '0';
            cout <= '0';
        else
            case (abc) is
                when "000" => sum <= '0'; cout <= '0';
                when "001" => sum <= '1'; cout <= '0';
                when "010" => sum <= '1'; cout <= '0';
                when "011" => sum <= '0'; cout <= '1';
                when "100" => sum <= '1'; cout <= '0';
                when "101" => sum <= '0'; cout <= '1';
                when "110" => sum <= '0'; cout <= '1';
                when "111" => sum <= '1'; cout <= '1';
                when others => sum <= '0'; cout <= '0';
            end case;
        end if;
    end process;

end Behavioral;

3.STRUCTURAL MODELLING
----------------------------------------------------------------------------------
--
-- Company:
-- Engineer:
```

```
--
-- Create Date: 10/05/2017 07:09:54 PM
-- Design Name:
-- Module Name: FullAdder - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------------
--
--Full Adder using Gates, Gate-level style.

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity FullAdder is
    port (en, a, b, cin : in std_logic;
    sum, cout : out std_logic);
end FullAdder;

architecture Structural of FullAdder is
    component andThree
        port (A, B, C : in std_logic; D : out std_logic);
    end component;
    component xorThree
        port (A, B, C : in std_logic; D : out std_logic);
    end component;
    component orThree
            port (A, B, C : in std_logic; D : out std_logic);
    end component;
    signal a0, a1, a2, a3 : std_logic;
    begin
        X1 : xorThree port map(A => a, B => b, C => cin, D => a0);
        X2 : andThree port map(A => a0, B => en, C => '1', D => sum);
        X3 : andThree port map(A => en, B => a, C => b, D => a1);
        X4 : andThree port map(A => en, B => b, C => cin, D => a2);
        X5 : andThree port map(A => en, B => a, C => cin, D => a3);
        X6 : orThree port map(A => a1, B => a2, C => a3, D => cout);

end Structural;

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;
```

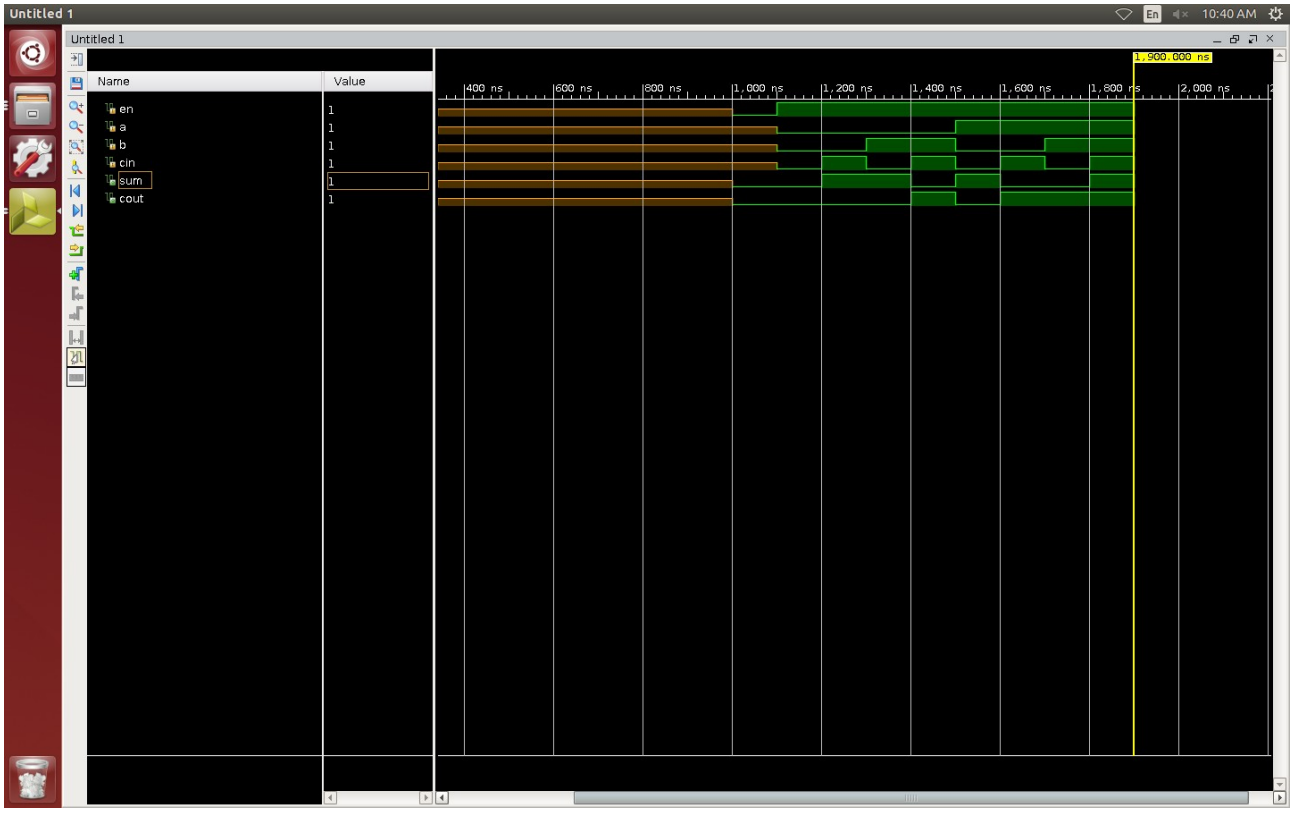
```
-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity andThree is
  port (A, B, C : in std_logic; D : out std_logic);
end andThree;

architecture DataFlow of andThree is

begin
    D <= A and B and C;

end DataFlow;

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity orThree is
   port (A, B, C : in std_logic; D : out std_logic);
end orThree;

architecture DataFlow of orThree is

begin
    D <= A or B or C;

end DataFlow;

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity xorThree is
  port (A, B, C : in std_logic; D : out std_logic);
end xorThree;

architecture DataFlow of xorThree is

begin
    D <= A xor B xor C;

end DataFlow;
```

B)FULL ADDER USING HALF ADDER

```
----------------------------------------------------------------------------------
--
-- Company:
-- Engineer:
--
-- Create Date: 10/05/2017 08:58:34 PM
-- Design Name:
-- Module Name: FullAdder1 - Structural
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------------
--
--Full Adder using Half Adders and OR gate, gate-level style.

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
library UNISIM;
use UNISIM.VComponents.all;

entity FullAdder1 is
  port (en, a, b, cin : in std_logic;
  sum, cout : out std_logic);
end FullAdder1;

architecture Structural of FullAdder1 is
    component HalfAdder
        port (EN, A, B : in std_logic; SUM, COUT : out std_logic);
    end component;
    component orTwo
        port (A, B : in std_logic; Y : out std_logic);
    end component;
    signal a1, a2, a3 : std_logic;
begin
    X1 : HalfAdder port map (EN => en, A => a, B => b, SUM => a1, COUT => a2);
    X2 : HalfAdder port map (EN => en, A => a1, B => cin, SUM => sum, COUT =>
a3);
    X3 : orTwo port map (A => a2, B => a3, Y => cout);

end Structural;

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
```

```
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity HalfAdder is
  port (EN, A, B : in std_logic; SUM, COUT : out std_logic);
end HalfAdder;

architecture DataFlow of HalfAdder is

begin
    SUM <= EN and (A xor B);
    COUT <= EN and (A and B);

end DataFlow;

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity orTwo is
    port (A, B : in std_logic; Y : out std_logic);

end orTwo;

architecture DataFlow of orTwo is

begin
    Y <= A or B;

end DataFlow;
```

SIMULATION RESULT



EXPERIMENT NO.21: LCD

1.VERILOG CODE

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//
// Company:
// Engineer:
//
// Create Date: 31.10.2017 18:59:06
// Design Name:
// Module Name: lcddisplay_interface_behavioral
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
//


module lcddisplay_interface_behavioral(E,RS,RW,DB,btnr,clk);
output E,RS,RW;
output [7:0] DB;
input btnr;
input clk;
wire [7:0] DB;
wire E,RS,RW;
wire writedone;
wire delayok;
reg [20:0] count=21'b0;
reg [10:0] sclk=11'b0;
reg [3:0] stcur=stpoweron_delay;
reg [3:0] stnext;
parameter [3:0] stfunctionset=0,
                stdisplayctrlset=1,
                stdisplayclear=2,
                stpoweron_delay=3,
                stfunctionset_delay=4,
                stdisplayctrlset_delay=5,
                stdisplayclear_delay=6,
                stinitdne=7,
                stactwr=8,
                stchardelay=9;
parameter [9:0] lcd_cmds[0:23]={ {2'b00,8'h3C},
                                 {2'b00,8'h0C},
                                 {2'b00,8'h01},
                                 {2'b00,8'h02},
                                 {2'b10,8'h48},
                                 {2'b10,8'h65},
                                 {2'b10,8'h6C},
                                 {2'b10,8'h6C},
                                 {2'b10,8'h6F},
                                 {2'b00,8'h07},
                                 {2'b10,8'h46},
                                 {2'b10,8'h72},
                                 {2'b10,8'h6F},
                                 {2'b10,8'h6D},
```

```
                                    {2'b10,8'h20},
                                    {2'b10,8'h44},
                                    {2'b10,8'h69},
                                    {2'b10,8'h67},
                                    {2'b10,8'h69},
                                    {2'b10,8'h6C},
                                    {2'b10,8'h65},
                                    {2'b10,8'h6E},
                                    {2'b10,8'h74},
                                    {2'b00,8'h18} };
reg [4:0] lcd_cmd_ptr;
assign writedone = (lcd_cmd_ptr==5'd23)?1'b1:1'b0;
assign RS = lcd_cmds[lcd_cmd_ptr][9];
assign RW = lcd_cmds[lcd_cmd_ptr][8];
assign DB = lcd_cmds[lcd_cmd_ptr][7:0];
assign E = (stcur==stfunctionset||
            stcur==stdisplayctrlset||
            stcur==stdisplayclear||
            stcur==stactwr)?1'b1:1'b0;
always@ (posedge clk)
sclk=sclk+1;
always@ (posedge sclk[7])
begin
if((stnext==stinitdne||stnext==stdisplayctrlset||stnext==stdisplayclear)
    &&(writedone==1'b0))
    lcd_cmd_ptr <= lcd_cmd_ptr+1;
else if(stcur==stpoweron_delay||stnext==stpoweron_delay)
    lcd_cmd_ptr <= 5'd0;
end
always@ (posedge sclk[7])
begin
if(btnr==1'b1)
stcur<=stpoweron_delay;
else
stcur<=stnext;
end
always@ (stcur or delayok or writedone or lcd_cmd_ptr)
begin
case(stcur)
stpoweron_delay : if(delayok==1'b1)stnext<=stfunctionset;else
stnext<=stpoweron_delay;
stfunctionset : stnext<=stfunctionset_delay;
stfunctionset_delay : if(delayok==1'b1)stnext<=stdisplayctrlset;else
stnext<=stfunctionset_delay;
stdisplayctrlset : stnext<=stdisplayctrlset_delay;
stdisplayctrlset_delay : if(delayok==1'b1)stnext<=stdisplayclear;else
stnext<=stdisplayctrlset_delay;
stdisplayclear : stnext<=stdisplayclear_delay;
stdisplayclear_delay : if(delayok==1'b1)stnext<=stinitdne;else
stnext<=stdisplayclear_delay;
stinitdne : stnext<=stactwr;
stactwr : stnext<=stchardelay;
stchardelay : if(delayok==1'b1)stnext<=stinitdne;else stnext<=stchardelay;
default : stnext<=stpoweron_delay;
endcase
end
always@ (posedge sclk[7])
begin
if(delayok==1'b1)
count <= 21'b0;
else
count <= count+1;
end
assign delayok=( ((stcur==stpoweron_delay)&&(count==21'd2000000))||
```

```
                ((stcur==stfunctionset_delay)&&(count==21'd4000))||
                ((stcur==stdisplayctrlset_delay)&&(count==21'd4000))||
                ((stcur==stdisplayclear_delay)&&(count==21'd16000))||
                ((stcur==stchardelay)&&(count==21'd26000))   )?1'b1:1'b0;

endmodule
```

2.VHDL CODE

```
----------------------------------------------------------------------------------
--
-- Company:
-- Engineer:
--
-- Create Date: 11/03/2017 08:55:45 PM
-- Design Name:
-- Module Name: LCD - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------------
--
-- LCD display using VHDL.

library IEEE;
use IEEE.std_logic_1164.ALL;
use IEEE.std_logic_arith.ALL;
use IEEE.std_logic_unsigned.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
library UNISIM;
use UNISIM.VComponents.all;

entity LCD is
    port
    (
        btnr : in std_logic;
        clk : in std_logic;
        db : out std_logic_vector (7 downto 0);
        en : out std_logic;
        rs : out std_logic;
        rw : out std_logic
    );
    constant FunctionSet : natural := 0;
    constant DisplayCtrlSet : natural := 1;
    constant DisplayClear : natural := 2;
    constant PowerOnDelay : natural := 3;
    constant FunctionSetDelay : natural := 4;
    constant DisplayCtrlSetDelay : natural := 5;
    constant DisplayClearDelay : natural := 6;
    constant InitDne : natural := 7;
    constant ActWr : natural := 8;
```

```
    constant CharDelay : natural := 9;

    type concat is
    record
        sw : std_logic_vector (1 downto 0);
        data: std_logic_vector (7 downto 0);
    end record;
    type lcdCmds is array (0 to 23) of concat;

    constant Cmds : lcdCmds :=
    (
        ("00", x"3C"),
        ("00", x"0C"),
        ("00", x"01"),
        ("00", x"02"),
        ("10", x"48"),  --H
        ("10", x"65"),  --e
        ("10", x"6C"),  --l
        ("10", x"6C"),  --l
        ("10", x"6F"),  --o
        ("10", x"20"),  --
        ("10", x"46"),  --F
        ("10", x"72"),  --r
        ("10", x"6F"),  --o
        ("10", x"6D"),  --m
        ("10", x"20"),  --
        ("10", x"44"),  --D
        ("10", x"69"),  --i
        ("10", x"67"),  --g
        ("10", x"69"),  --i
        ("10", x"6C"),  --l
        ("10", x"65"),  --e
        ("10", x"6E"),  --n
        ("10", x"74"),  --t
        ("00", x"18")
    );

end LCD;

architecture Mixed of LCD is

    signal writeDone : std_logic;
    signal count : natural := 0;
    signal delayOK : std_logic;

    signal sclk : std_logic_vector (10 downto 0);
    signal lcdCur : natural := PowerOnDelay;
    signal lcdNext : natural;
    signal cmdPtr : natural;

begin
    writeDone <= '1' when cmdPtr = 23 else '0';

    rs <= Cmds (cmdPtr).sw(1);
    rw <= Cmds (cmdPtr).sw(0);
    db <= Cmds (cmdPtr).data;

    en <= '1' when ((lcdCur = FunctionSet) or (lcdCur = DisplayCtrlSet) or
(lcdCur = DisplayClear) or (lcdCur = ActWr)) else '0';

    process (clk)
    begin
        if (rising_edge (clk)) then
            sclk <= sclk + '1';
```

```
        end if;
    end process;

    process (sclk(8))
    begin
        if (rising_edge (sclk (8))) then
            if ((lcdNext = InitDne or lcdNext = DisplayCtrlSet or lcdNext =
DisplayClear) and (writeDone = '1')) then
                cmdPtr <= cmdPtr + 1;
            elsif ((lcdCur = PowerOnDelay) or (lcdNext = PowerOnDelay)) then
                cmdPtr <= 0;
            else
                cmdPtr <= cmdPtr;
            end if;
        end if;
    end process;

    process (sclk(8))
    begin
        if (rising_edge (sclk (8))) then
            if (btnr = '1') then
                lcdCur <= PowerOnDelay;
            else
                lcdCur <= lcdNext;
            end if;
        end if;
    end process;

    process (lcdCur, delayOK, writeDone, cmdPtr)
    begin
        case (lcdCur) is
            when PowerOnDelay =>
                if (delayOk = '1') then
                    lcdNext <= DisplayCtrlSet;
                else
                    lcdNext <= FunctionSetDelay;
                end if;
            when FunctionSet =>
                lcdNext <= FunctionSetDelay;
            when FunctionSetDelay =>
                if (delayOK = '1') then
                    lcdNext <= DisplayCtrlSet;
                else
                    lcdNext <= FunctionSetDelay;
                end if;
            when DisplayCtrlSet =>
                lcdNext <= DisplayCtrlSetDelay;
            when DisplayCtrlSetDelay =>
                if (delayOK = '1') then
                    lcdNext <= DisplayClear;
                else
                    lcdNext <= DisplayCtrlSetDelay;
                end if;
            when DisplayClear =>
                lcdNext <= DisplayClearDelay;
            when DisplayClearDelay =>
                 if (delayOK = '1') then
                    lcdNext <= InitDne;
                 else
                    lcdNext <= DisplayClearDelay;
                 end if;
            when InitDne =>
                lcdNext <= ActWr;
            when ActWr =>
```

```
                lcdNext <= CharDelay;
            when CharDelay =>
                if (delayOK = '1') then
                    lcdNext <= InitDne;
                else
                    lcdNext <= CharDelay;
                end if;
            when others =>
                lcdNext <= PowerOnDelay;

        end case;

    end process;

    process (sclk (8))
    begin
        if (rising_edge (sclk (8))) then
            if (delayOK = '1') then
                count <= 0;
            else
                count <= count + 1;
            end if;
        end if;
    end process;

    delayOK <= '1' when ((lcdCur = PowerOnDelay and count = 2000000) or
                         (lcdCur = FunctionSetDelay and count = 4000) or
                         (lcdCur = DisplayCtrlSetDelay and count = 4000) or
                         (lcdCur = DisplayClearDelay and count = 6000) or
                         (lcdCur = CharDelay and count = 26000))
                   else '0';

end Mixed;

XDC FILE
## This file is a general .xdc for the Basys3 rev B board
## To use it in a project:
## - uncomment the lines corresponding to used pins
## - rename the used ports (in each line, after get_ports) according to the top
level signal names in the project

## Clock signal
set_property PACKAGE_PIN W5 [get_ports clk]

      set_property IOSTANDARD LVCMOS33 [get_ports clk]
      #create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5}
[get_ports clk]

## Switches
set_property PACKAGE_PIN V17 [get_ports {btnr}]
      set_property IOSTANDARD LVCMOS33 [get_ports {btnr}]
#set_property PACKAGE_PIN V16 [get_ports {sw[1]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[1]}]
#set_property PACKAGE_PIN W16 [get_ports {sw[2]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[2]}]
#set_property PACKAGE_PIN W17 [get_ports {sw[3]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[3]}]
#set_property PACKAGE_PIN W15 [get_ports {sw[4]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[4]}]
#set_property PACKAGE_PIN V15 [get_ports {sw[5]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[5]}]
#set_property PACKAGE_PIN W14 [get_ports {sw[6]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[6]}]
#set_property PACKAGE_PIN W13 [get_ports {sw[7]}]
```

```
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[7]}]
#set_property PACKAGE_PIN V2 [get_ports {sw[8]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[8]}]
#set_property PACKAGE_PIN T3 [get_ports {sw[9]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[9]}]
#set_property PACKAGE_PIN T2 [get_ports {sw[10]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[10]}]
#set_property PACKAGE_PIN R3 [get_ports {sw[11]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[11]}]
#set_property PACKAGE_PIN W2 [get_ports {sw[12]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[12]}]
#set_property PACKAGE_PIN U1 [get_ports {sw[13]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[13]}]
#set_property PACKAGE_PIN T1 [get_ports {sw[14]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[14]}]
#set_property PACKAGE_PIN R2 [get_ports {sw[15]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[15]}]


## LEDs
#set_property PACKAGE_PIN U16 [get_ports {led[0]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[0]}]
#set_property PACKAGE_PIN E19 [get_ports {led[1]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[1]}]
#set_property PACKAGE_PIN U19 [get_ports {led[2]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[2]}]
#set_property PACKAGE_PIN V19 [get_ports {led[3]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[3]}]
#set_property PACKAGE_PIN W18 [get_ports {led[4]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[4]}]
#set_property PACKAGE_PIN U15 [get_ports {led[5]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[5]}]
#set_property PACKAGE_PIN U14 [get_ports {led[6]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[6]}]
#set_property PACKAGE_PIN V14 [get_ports {led[7]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[7]}]
#set_property PACKAGE_PIN V13 [get_ports {led[8]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[8]}]
#set_property PACKAGE_PIN V3 [get_ports {led[9]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[9]}]
#set_property PACKAGE_PIN W3 [get_ports {led[10]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[10]}]
#set_property PACKAGE_PIN U3 [get_ports {led[11]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[11]}]
#set_property PACKAGE_PIN P3 [get_ports {led[12]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[12]}]
#set_property PACKAGE_PIN N3 [get_ports {led[13]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[13]}]
#set_property PACKAGE_PIN P1 [get_ports {led[14]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[14]}]
#set_property PACKAGE_PIN L1 [get_ports {led[15]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[15]}]


##7 segment display
#set_property PACKAGE_PIN W7 [get_ports {seg[0]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {seg[0]}]
#set_property PACKAGE_PIN W6 [get_ports {seg[1]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {seg[1]}]
#set_property PACKAGE_PIN U8 [get_ports {seg[2]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {seg[2]}]
#set_property PACKAGE_PIN V8 [get_ports {seg[3]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {seg[3]}]
#set_property PACKAGE_PIN U5 [get_ports {seg[4]}]
```

```
      #set_property IOSTANDARD LVCMOS33 [get_ports {seg[4]}]
#set_property PACKAGE_PIN V5 [get_ports {seg[5]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {seg[5]}]
#set_property PACKAGE_PIN U7 [get_ports {seg[6]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {seg[6]}]

#set_property PACKAGE_PIN V7 [get_ports dp]

      #set_property IOSTANDARD LVCMOS33 [get_ports dp]

#set_property PACKAGE_PIN U2 [get_ports {an[0]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {an[0]}]
#set_property PACKAGE_PIN U4 [get_ports {an[1]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {an[1]}]
#set_property PACKAGE_PIN V4 [get_ports {an[2]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {an[2]}]
#set_property PACKAGE_PIN W4 [get_ports {an[3]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {an[3]}]


##Buttons
#set_property PACKAGE_PIN U18 [get_ports btnC]
      #set_property IOSTANDARD LVCMOS33 [get_ports btnC]
#set_property PACKAGE_PIN T18 [get_ports btnU]
      #set_property IOSTANDARD LVCMOS33 [get_ports btnU]
#set_property PACKAGE_PIN W19 [get_ports btnL]
      #set_property IOSTANDARD LVCMOS33 [get_ports btnL]
#set_property PACKAGE_PIN T17 [get_ports btnR]
      #set_property IOSTANDARD LVCMOS33 [get_ports btnR]
#set_property PACKAGE_PIN U17 [get_ports btnD]
      #set_property IOSTANDARD LVCMOS33 [get_ports btnD]



##Pmod Header JA
##Sch name = JA1
#set_property PACKAGE_PIN J1 [get_ports {JA[0]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JA[0]}]
##Sch name = JA2
#set_property PACKAGE_PIN L2 [get_ports {JA[1]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JA[1]}]
##Sch name = JA3
#set_property PACKAGE_PIN J2 [get_ports {JA[2]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JA[2]}]
##Sch name = JA4
#set_property PACKAGE_PIN G2 [get_ports {JA[3]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JA[3]}]
##Sch name = JA7
#set_property PACKAGE_PIN H1 [get_ports {JA[4]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JA[4]}]
##Sch name = JA8
#set_property PACKAGE_PIN K2 [get_ports {JA[5]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JA[5]}]
##Sch name = JA9
#set_property PACKAGE_PIN H2 [get_ports {JA[6]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JA[6]}]
##Sch name = JA10
#set_property PACKAGE_PIN G3 [get_ports {JA[7]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JA[7]}]



##Pmod Header JB
##Sch name = JB1
```

```
set_property PACKAGE_PIN A14 [get_ports {DB[0]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {DB[0]}]
##Sch name = JB2
set_property PACKAGE_PIN A16 [get_ports {DB[1]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {DB[1]}]
##Sch name = JB3
set_property PACKAGE_PIN B15 [get_ports {DB[2]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {DB[2]}]
##Sch name = JB4
set_property PACKAGE_PIN B16 [get_ports {DB[3]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {DB[3]}]
##Sch name = JB7
set_property PACKAGE_PIN A15 [get_ports {DB[4]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {DB[4]}]
##Sch name = JB8
set_property PACKAGE_PIN A17 [get_ports {DB[5]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {DB[5]}]
##Sch name = JB9
set_property PACKAGE_PIN C15 [get_ports {DB[6]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {DB[6]}]
##Sch name = JB10
set_property PACKAGE_PIN C16 [get_ports {DB[7]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {DB[7]}]



##Pmod Header JC
##Sch name = JC1
#set_property PACKAGE_PIN K17 [get_ports {JC[0]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JC[0]}]
##Sch name = JC2
#set_property PACKAGE_PIN M18 [get_ports {JC[1]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JC[1]}]
##Sch name = JC3
#set_property PACKAGE_PIN N17 [get_ports {JC[2]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JC[2]}]
##Sch name = JC4
#set_property PACKAGE_PIN P18 [get_ports {JC[3]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JC[3]}]
##Sch name = JC7
set_property PACKAGE_PIN L17 [get_ports {RS}]
      set_property IOSTANDARD LVCMOS33 [get_ports {RS}]
##Sch name = JC8
set_property PACKAGE_PIN M19 [get_ports {RW}]
      set_property IOSTANDARD LVCMOS33 [get_ports {RW}]
##Sch name = JC9
set_property PACKAGE_PIN P17 [get_ports {E}]
      set_property IOSTANDARD LVCMOS33 [get_ports {E}]
##Sch name = JC10
#set_property PACKAGE_PIN R18 [get_ports {JC[7]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JC[7]}]


##Pmod Header JXADC
##Sch name = XA1_P
#set_property PACKAGE_PIN J3 [get_ports {JXADC[0]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[0]}]
##Sch name = XA2_P
#set_property PACKAGE_PIN L3 [get_ports {JXADC[1]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[1]}]
##Sch name = XA3_P
#set_property PACKAGE_PIN M2 [get_ports {JXADC[2]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[2]}]
##Sch name = XA4_P
```

```
#set_property PACKAGE_PIN N2 [get_ports {JXADC[3]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[3]}]
##Sch name = XA1_N
#set_property PACKAGE_PIN K3 [get_ports {JXADC[4]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[4]}]
##Sch name = XA2_N
#set_property PACKAGE_PIN M3 [get_ports {JXADC[5]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[5]}]
##Sch name = XA3_N
#set_property PACKAGE_PIN M1 [get_ports {JXADC[6]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[6]}]
##Sch name = XA4_N
#set_property PACKAGE_PIN N1 [get_ports {JXADC[7]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[7]}]



##VGA Connector
#set_property PACKAGE_PIN G19 [get_ports {vgaRed[0]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {vgaRed[0]}]
#set_property PACKAGE_PIN H19 [get_ports {vgaRed[1]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {vgaRed[1]}]
#set_property PACKAGE_PIN J19 [get_ports {vgaRed[2]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {vgaRed[2]}]
#set_property PACKAGE_PIN N19 [get_ports {vgaRed[3]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {vgaRed[3]}]
#set_property PACKAGE_PIN N18 [get_ports {vgaBlue[0]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {vgaBlue[0]}]
#set_property PACKAGE_PIN L18 [get_ports {vgaBlue[1]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {vgaBlue[1]}]
#set_property PACKAGE_PIN K18 [get_ports {vgaBlue[2]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {vgaBlue[2]}]
#set_property PACKAGE_PIN J18 [get_ports {vgaBlue[3]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {vgaBlue[3]}]
#set_property PACKAGE_PIN J17 [get_ports {vgaGreen[0]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {vgaGreen[0]}]
#set_property PACKAGE_PIN H17 [get_ports {vgaGreen[1]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {vgaGreen[1]}]
#set_property PACKAGE_PIN G17 [get_ports {vgaGreen[2]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {vgaGreen[2]}]
#set_property PACKAGE_PIN D17 [get_ports {vgaGreen[3]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {vgaGreen[3]}]
#set_property PACKAGE_PIN P19 [get_ports Hsync]
      #set_property IOSTANDARD LVCMOS33 [get_ports Hsync]
#set_property PACKAGE_PIN R19 [get_ports Vsync]
      #set_property IOSTANDARD LVCMOS33 [get_ports Vsync]


##USB-RS232 Interface
#set_property PACKAGE_PIN B18 [get_ports RsRx]
      #set_property IOSTANDARD LVCMOS33 [get_ports RsRx]
#set_property PACKAGE_PIN A18 [get_ports RsTx]
      #set_property IOSTANDARD LVCMOS33 [get_ports RsTx]


##USB HID (PS/2)
#set_property PACKAGE_PIN C17 [get_ports PS2Clk]

      #set_property IOSTANDARD LVCMOS33 [get_ports PS2Clk]
      #set_property PULLUP true [get_ports PS2Clk]
#set_property PACKAGE_PIN B17 [get_ports PS2Data]
      #set_property IOSTANDARD LVCMOS33 [get_ports PS2Data]
      #set_property PULLUP true [get_ports PS2Data]
```

```
##Quad SPI Flash
##Note that CCLK_0 cannot be placed in 7 series devices. You can access it using
the
##STARTUPE2 primitive.
#set_property PACKAGE_PIN D18 [get_ports {QspiDB[0]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {QspiDB[0]}]
#set_property PACKAGE_PIN D19 [get_ports {QspiDB[1]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {QspiDB[1]}]
#set_property PACKAGE_PIN G18 [get_ports {QspiDB[2]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {QspiDB[2]}]
#set_property PACKAGE_PIN F18 [get_ports {QspiDB[3]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {QspiDB[3]}]
#set_property PACKAGE_PIN K19 [get_ports QspiCSn]
      #set_property IOSTANDARD LVCMOS33 [get_ports QspiCSn]
```

EXPERIMENT NO.22: MOTORS

1.DC MOTOR

VERILOG CODE

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//
// Company:
// Engineer:
//
// Create Date: 09.11.2017 19:08:13
// Design Name:
// Module Name: dc_motor_verilog
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
//

module dc_motor_verilog(EN,DIR,reset,clk,speed);

output EN,DIR;
input reset;
input clk;
input [3:0]speed;
reg EN,DIR;
reg [25:0]clk_div;
wire clk_int;
integer onperiod;
integer T_off;
always@(posedge clk)
clk_div = clk_div + 1;

assign clk_int=clk_div[1];
always@(posedge clk_int)
onperiod = onperiod +1;
always@(speed)
begin
case (speed)
4'b1110 : T_off = 90; // DUTY CYCLE=10%
4'b1101 : T_off = 75; // DUTY CYCLE=25%
4'b1011 : T_off = 50; // DUTY CYCLE=50%
4'b0111 : T_off = 25; // DUTY CYCLE=75%
default: T_off <= 10;// -- DUTY CYCLE=90%
endcase
end
always@(onperiod)
begin
if (onperiod >=T_off) begin
if (reset==0) begin
EN =1;DIR=0;
end
else begin
EN=1;DIR=1;
end
end
```

```
end
endmodule

VHDL CODE
----------------------------------------------------------------------------------
--
-- Company:
-- Engineer:
--
-- Create Date: 09.11.2017 18:50:56
-- Design Name:
-- Module Name: dc_motor - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------------
--


library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity dc_motor is
Port(EN,DIR : out std_logic; reset: in std_logic;
clk :in std_logic;
speed : in std_logic_vector(3 downto 0));
end dc_motor;
architecture dc_motor1 of dc_motor is
signal clk_div : std_logic_vector(25 downto 0);
signal clk_int: std_logic;
signal onperiod: integer range 0 to 100 :=0;
signal T_off: integer range 0 to 100;
begin
process(clk)
begin
if rising_edge (clk) then
clk_div <= clk_div + '1';
end if;
end process;
clk_int<=clk_div(1);
process(clk_int)
begin
if rising_edge(clk_int) then
onperiod <= onperiod +1;
end if;
```

```
end process;
process(speed)
begin
case speed is
when "1110" => T_off <= 90; -- DUTY CYCLE=10%
when "1101" => T_off <= 75; -- DUTY CYCLE=25%
when "1011" => T_off <= 50; -- DUTY CYCLE=50%
when "0111" => T_off <= 25; -- DUTY CYCLE=75%
when others => T_off <= 10; -- DUTY CYCLE=90%
end case;
end process;
Process(onperiod)
begin
if onperiod >=T_off then
if reset='0' then
EN <='1';DIR<='0';
else
EN<='1';DIR<='1';
end if;
end if;
end process;
end dc_motor1;
```

XDC FILE

```
## This file is a general .xdc for the Basys3 rev B board
## To use it in a project:
## - uncomment the lines corresponding to used pins
## - rename the used ports (in each line, after get_ports) according to the top
level signal names in the project

## Clock signal
set_property PACKAGE_PIN W5 [get_ports clk]

      set_property IOSTANDARD LVCMOS33 [get_ports clk]
      #create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5}
[get_ports clk]

## Switches
set_property PACKAGE_PIN V17 [get_ports {reset}]
set_property IOSTANDARD LVCMOS33 [get_ports {reset}]
set_property PACKAGE_PIN V16 [get_ports {speed[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {speed[0]}]
set_property PACKAGE_PIN W16 [get_ports {speed[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {speed[1]}]
set_property PACKAGE_PIN W17 [get_ports {speed[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {speed[2]}]
set_property PACKAGE_PIN W15 [get_ports {speed[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {speed[3]}]
#set_property PACKAGE_PIN V15 [get_ports {sw[5]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[5]}]
#set_property PACKAGE_PIN W14 [get_ports {sw[6]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[6]}]
#set_property PACKAGE_PIN W13 [get_ports {sw[7]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[7]}]
#set_property PACKAGE_PIN V2 [get_ports {sw[8]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[8]}]
#set_property PACKAGE_PIN T3 [get_ports {sw[9]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[9]}]
#set_property PACKAGE_PIN T2 [get_ports {sw[10]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[10]}]
#set_property PACKAGE_PIN R3 [get_ports {sw[11]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[11]}]
#set_property PACKAGE_PIN W2 [get_ports {sw[12]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[12]}]
```

```
#set_property PACKAGE_PIN U1 [get_ports {sw[13]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[13]}]
#set_property PACKAGE_PIN T1 [get_ports {sw[14]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[14]}]
#set_property PACKAGE_PIN R2 [get_ports {sw[15]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[15]}]


## LEDs
#set_property PACKAGE_PIN U16 [get_ports {led[0]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[0]}]
#set_property PACKAGE_PIN E19 [get_ports {led[1]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[1]}]
#set_property PACKAGE_PIN U19 [get_ports {led[2]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[2]}]
#set_property PACKAGE_PIN V19 [get_ports {led[3]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[3]}]
#set_property PACKAGE_PIN W18 [get_ports {led[4]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[4]}]
#set_property PACKAGE_PIN U15 [get_ports {led[5]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[5]}]
#set_property PACKAGE_PIN U14 [get_ports {led[6]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[6]}]
#set_property PACKAGE_PIN V14 [get_ports {led[7]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[7]}]
#set_property PACKAGE_PIN V13 [get_ports {led[8]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[8]}]
#set_property PACKAGE_PIN V3 [get_ports {led[9]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[9]}]
#set_property PACKAGE_PIN W3 [get_ports {led[10]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[10]}]
#set_property PACKAGE_PIN U3 [get_ports {led[11]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[11]}]
#set_property PACKAGE_PIN P3 [get_ports {led[12]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[12]}]
#set_property PACKAGE_PIN N3 [get_ports {led[13]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[13]}]
#set_property PACKAGE_PIN P1 [get_ports {led[14]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[14]}]
#set_property PACKAGE_PIN L1 [get_ports {led[15]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[15]}]


##7 segment display
#set_property PACKAGE_PIN W7 [get_ports {seg[0]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {seg[0]}]
#set_property PACKAGE_PIN W6 [get_ports {seg[1]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {seg[1]}]
#set_property PACKAGE_PIN U8 [get_ports {seg[2]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {seg[2]}]
#set_property PACKAGE_PIN V8 [get_ports {seg[3]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {seg[3]}]
#set_property PACKAGE_PIN U5 [get_ports {seg[4]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {seg[4]}]
#set_property PACKAGE_PIN V5 [get_ports {seg[5]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {seg[5]}]
#set_property PACKAGE_PIN U7 [get_ports {seg[6]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {seg[6]}]

#set_property PACKAGE_PIN V7 [get_ports dp]

      #set_property IOSTANDARD LVCMOS33 [get_ports dp]

#set_property PACKAGE_PIN U2 [get_ports {an[0]}]
```

```
      #set_property IOSTANDARD LVCMOS33 [get_ports {an[0]}]
#set_property PACKAGE_PIN U4 [get_ports {an[1]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {an[1]}]
#set_property PACKAGE_PIN V4 [get_ports {an[2]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {an[2]}]
#set_property PACKAGE_PIN W4 [get_ports {an[3]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {an[3]}]


##Buttons
#set_property PACKAGE_PIN U18 [get_ports btnC]
      #set_property IOSTANDARD LVCMOS33 [get_ports btnC]
#set_property PACKAGE_PIN T18 [get_ports btnU]
      #set_property IOSTANDARD LVCMOS33 [get_ports btnU]
#set_property PACKAGE_PIN W19 [get_ports btnL]
      #set_property IOSTANDARD LVCMOS33 [get_ports btnL]
#set_property PACKAGE_PIN T17 [get_ports btnR]
      #set_property IOSTANDARD LVCMOS33 [get_ports btnR]
#set_property PACKAGE_PIN U17 [get_ports btnD]
      #set_property IOSTANDARD LVCMOS33 [get_ports btnD]



##Pmod Header JA
##Sch name = JA1
#set_property PACKAGE_PIN J1 [get_ports {JA[0]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JA[0]}]
##Sch name = JA2
#set_property PACKAGE_PIN L2 [get_ports {JA[1]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JA[1]}]
##Sch name = JA3
#set_property PACKAGE_PIN J2 [get_ports {JA[2]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JA[2]}]
##Sch name = JA4
#set_property PACKAGE_PIN G2 [get_ports {JA[3]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JA[3]}]
##Sch name = JA7
#set_property PACKAGE_PIN H1 [get_ports {JA[4]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JA[4]}]
##Sch name = JA8
#set_property PACKAGE_PIN K2 [get_ports {JA[5]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JA[5]}]
##Sch name = JA9
#set_property PACKAGE_PIN H2 [get_ports {JA[6]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JA[6]}]
##Sch name = JA10
#set_property PACKAGE_PIN G3 [get_ports {JA[7]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JA[7]}]



##Pmod Header JB
##Sch name = JB1
#set_property PACKAGE_PIN A14 [get_ports {JB[0]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JB[0]}]
##Sch name = JB2
#set_property PACKAGE_PIN A16 [get_ports {JB[1]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JB[1]}]
##Sch name = JB3
#set_property PACKAGE_PIN B15 [get_ports {JB[2]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JB[2]}]
##Sch name = JB4
#set_property PACKAGE_PIN B16 [get_ports {JB[3]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JB[3]}]
```

```
##Sch name = JB7
#set_property PACKAGE_PIN A15 [get_ports {JB[4]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JB[4]}]
##Sch name = JB8
#set_property PACKAGE_PIN A17 [get_ports {JB[5]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JB[5]}]
##Sch name = JB9
#set_property PACKAGE_PIN C15 [get_ports {JB[6]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JB[6]}]
##Sch name = JB10
#set_property PACKAGE_PIN C16 [get_ports {JB[7]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JB[7]}]



##Pmod Header JC
##Sch name = JC1
#set_property PACKAGE_PIN K17 [get_ports {JC[0]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JC[0]}]
##Sch name = JC2
#set_property PACKAGE_PIN M18 [get_ports {JC[1]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JC[1]}]
##Sch name = JC3
#set_property PACKAGE_PIN N17 [get_ports {JC[2]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JC[2]}]
##Sch name = JC4
#set_property PACKAGE_PIN P18 [get_ports {JC[3]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JC[3]}]
##Sch name = JC7
set_property PACKAGE_PIN L17 [get_ports {DIR}]
set_property IOSTANDARD LVCMOS33 [get_ports {DIR}]
##Sch name = JC8
set_property PACKAGE_PIN M19 [get_ports {EN}]
set_property IOSTANDARD LVCMOS33 [get_ports {EN}]
##Sch name = JC9
#set_property PACKAGE_PIN P17 [get_ports {JC[6]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JC[6]}]
##Sch name = JC10
#set_property PACKAGE_PIN R18 [get_ports {JC[7]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JC[7]}]


##Pmod Header JXADC
##Sch name = XA1_P
#set_property PACKAGE_PIN J3 [get_ports {JXADC[0]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[0]}]
##Sch name = XA2_P
#set_property PACKAGE_PIN L3 [get_ports {JXADC[1]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[1]}]
##Sch name = XA3_P
#set_property PACKAGE_PIN M2 [get_ports {JXADC[2]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[2]}]
##Sch name = XA4_P
#set_property PACKAGE_PIN N2 [get_ports {JXADC[3]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[3]}]
##Sch name = XA1_N
#set_property PACKAGE_PIN K3 [get_ports {JXADC[4]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[4]}]
##Sch name = XA2_N
#set_property PACKAGE_PIN M3 [get_ports {JXADC[5]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[5]}]
##Sch name = XA3_N
#set_property PACKAGE_PIN M1 [get_ports {JXADC[6]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[6]}]
```

```
##Sch name = XA4_N
#set_property PACKAGE_PIN N1 [get_ports {JXADC[7]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[7]}]



##VGA Connector
#set_property PACKAGE_PIN G19 [get_ports {vgaRed[0]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {vgaRed[0]}]
#set_property PACKAGE_PIN H19 [get_ports {vgaRed[1]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {vgaRed[1]}]
#set_property PACKAGE_PIN J19 [get_ports {vgaRed[2]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {vgaRed[2]}]
#set_property PACKAGE_PIN N19 [get_ports {vgaRed[3]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {vgaRed[3]}]
#set_property PACKAGE_PIN N18 [get_ports {vgaBlue[0]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {vgaBlue[0]}]
#set_property PACKAGE_PIN L18 [get_ports {vgaBlue[1]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {vgaBlue[1]}]
#set_property PACKAGE_PIN K18 [get_ports {vgaBlue[2]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {vgaBlue[2]}]
#set_property PACKAGE_PIN J18 [get_ports {vgaBlue[3]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {vgaBlue[3]}]
#set_property PACKAGE_PIN J17 [get_ports {vgaGreen[0]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {vgaGreen[0]}]
#set_property PACKAGE_PIN H17 [get_ports {vgaGreen[1]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {vgaGreen[1]}]
#set_property PACKAGE_PIN G17 [get_ports {vgaGreen[2]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {vgaGreen[2]}]
#set_property PACKAGE_PIN D17 [get_ports {vgaGreen[3]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {vgaGreen[3]}]
#set_property PACKAGE_PIN P19 [get_ports Hsync]
      #set_property IOSTANDARD LVCMOS33 [get_ports Hsync]
#set_property PACKAGE_PIN R19 [get_ports Vsync]
      #set_property IOSTANDARD LVCMOS33 [get_ports Vsync]


##USB-RS232 Interface
#set_property PACKAGE_PIN B18 [get_ports RsRx]
      #set_property IOSTANDARD LVCMOS33 [get_ports RsRx]
#set_property PACKAGE_PIN A18 [get_ports RsTx]
      #set_property IOSTANDARD LVCMOS33 [get_ports RsTx]


##USB HID (PS/2)
#set_property PACKAGE_PIN C17 [get_ports PS2Clk]

      #set_property IOSTANDARD LVCMOS33 [get_ports PS2Clk]
      #set_property PULLUP true [get_ports PS2Clk]
#set_property PACKAGE_PIN B17 [get_ports PS2Data]
      #set_property IOSTANDARD LVCMOS33 [get_ports PS2Data]
      #set_property PULLUP true [get_ports PS2Data]


##Quad SPI Flash
##Note that CCLK_0 cannot be placed in 7 series devices. You can access it using
the
##STARTUPE2 primitive.
#set_property PACKAGE_PIN D18 [get_ports {QspiDB[0]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {QspiDB[0]}]
#set_property PACKAGE_PIN D19 [get_ports {QspiDB[1]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {QspiDB[1]}]
#set_property PACKAGE_PIN G18 [get_ports {QspiDB[2]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {QspiDB[2]}]
```

```
#set_property PACKAGE_PIN F18 [get_ports {QspiDB[3]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {QspiDB[3]}]
#set_property PACKAGE_PIN K19 [get_ports QspiCSn]
      #set_property IOSTANDARD LVCMOS33 [get_ports QspiCSn]
```

2.STEPPER MOTOR

VERILOG CODE

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//
// Company:
// Engineer:
//
// Create Date: 09.11.2017 19:45:38
// Design Name:
// Module Name: stepper_verilog
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
//

    module stepper_verilog(motor,clk,reset,speed,direction);
    output [3:0]motor;
    input clk,reset;
    input [1:0]speed;
    input direction;

    reg [25:0]clk_div;
    wire clk_int;
    reg [3:0]angle =4'b1001;

    always@(posedge clk)
    clk_div=clk_div +1;

    assign clk_int =(speed == 2'b0)?clk_div[25]:(speed ==2'b01)?clk_div[23]:
(speed==2'b10)?clk_div[21]:clk_div[19];

    always@(negedge reset,posedge clk_int)
    begin
    if (reset==0)
                angle = 4'b1001;
    else if (clk_int==1) begin
        if (direction==0)
        angle ={ angle[0] , angle[3:1] };
        else
        angle ={ angle[2 : 0] , angle[3] };
    end
    end
    assign motor = angle;

endmodule
```

VHDL CODE

```
----------------------------------------------------------------------------------
--
-- Company:
-- Engineer:
--
-- Create Date: 09.11.2017 19:18:34
-- Design Name:
-- Module Name: stepper_vhdl - stepper
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------------
--


library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity stepper_vhdl is
Port (motor: out std_logic_vector(3 downto 0 );
        clk,reset: in std_logic;
        speed: in std_logic_vector(1 downto 0);
        direction: in std_logic);
end stepper_vhdl;

architecture stepper of stepper_vhdl is
signal clk_div : std_logic_vector(25 downto 0);
signal clk_int : std_logic;
signal angle : std_logic_vector(3 downto 0):="1001";
begin
process(clk)
begin
if rising_edge (clk) then
clk_div<=clk_div +'1';
end if;
end process;
clk_int<= clk_div(25) when speed ="00" else
            clk_div(23) when speed ="01" else
            clk_div(21) when speed ="10" else
            clk_div(19);
process(reset,clk_int,direction)
begin
if reset='0' then
```

```
            angle <= "1001";
elsif rising_edge(clk_int) then
    if direction='0' then
    angle <= angle(0) & angle(3 downto 1);
    else
    angle <= angle(2 downto 0) & angle(3);
    end if;
end if;
end process;
motor <= angle;

end stepper;
```

XDC FILE

```
### This file is a general .xdc for the Basys3 rev B board
### To use it in a project:
### - uncomment the lines corresponding to used pins
### - rename the used ports (in each line, after get_ports) according to the top
level signal names in the project

### Clock signal
set_property PACKAGE_PIN W5 [get_ports clk]

      set_property IOSTANDARD LVCMOS33 [get_ports clk]
#     create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5}
[get_ports clk]

### Switches
set_property PACKAGE_PIN V17 [get_ports {reset}]
      set_property IOSTANDARD LVCMOS33 [get_ports {reset}]
set_property PACKAGE_PIN V16 [get_ports {speed[1]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {speed[1]}]
set_property PACKAGE_PIN W16 [get_ports {speed[0]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {speed[0]}]
set_property PACKAGE_PIN W17 [get_ports {direction}]
      set_property IOSTANDARD LVCMOS33 [get_ports {direction}]
#set_property PACKAGE_PIN W15 [get_ports {speed[3]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {speed[3]}]
##set_property PACKAGE_PIN V15 [get_ports {sw[5]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {sw[5]}]
##set_property PACKAGE_PIN W14 [get_ports {sw[6]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {sw[6]}]
##set_property PACKAGE_PIN W13 [get_ports {sw[7]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {sw[7]}]
##set_property PACKAGE_PIN V2 [get_ports {sw[8]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {sw[8]}]
##set_property PACKAGE_PIN T3 [get_ports {sw[9]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {sw[9]}]
##set_property PACKAGE_PIN T2 [get_ports {sw[10]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {sw[10]}]
##set_property PACKAGE_PIN R3 [get_ports {sw[11]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {sw[11]}]
##set_property PACKAGE_PIN W2 [get_ports {sw[12]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {sw[12]}]
##set_property PACKAGE_PIN U1 [get_ports {sw[13]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {sw[13]}]
##set_property PACKAGE_PIN T1 [get_ports {sw[14]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {sw[14]}]
##set_property PACKAGE_PIN R2 [get_ports {sw[15]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {sw[15]}]


### LEDs
##set_property PACKAGE_PIN U16 [get_ports {y}]
```

```
#     #set_property IOSTANDARD LVCMOS33 [get_ports {y}]
##set_property PACKAGE_PIN E19 [get_ports {led[1]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {led[1]}]
##set_property PACKAGE_PIN U19 [get_ports {led[2]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {led[2]}]
##set_property PACKAGE_PIN V19 [get_ports {led[3]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {led[3]}]
##set_property PACKAGE_PIN W18 [get_ports {led[4]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {led[4]}]
##set_property PACKAGE_PIN U15 [get_ports {led[5]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {led[5]}]
##set_property PACKAGE_PIN U14 [get_ports {led[6]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {led[6]}]
##set_property PACKAGE_PIN V14 [get_ports {led[7]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {led[7]}]
##set_property PACKAGE_PIN V13 [get_ports {led[8]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {led[8]}]
##set_property PACKAGE_PIN V3 [get_ports {led[9]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {led[9]}]
##set_property PACKAGE_PIN W3 [get_ports {led[10]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {led[10]}]
##set_property PACKAGE_PIN U3 [get_ports {led[11]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {led[11]}]
##set_property PACKAGE_PIN P3 [get_ports {led[12]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {led[12]}]
##set_property PACKAGE_PIN N3 [get_ports {led[13]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {led[13]}]
##set_property PACKAGE_PIN P1 [get_ports {led[14]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {led[14]}]
##set_property PACKAGE_PIN L1 [get_ports {led[15]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {led[15]}]


###7 segment display
#set_property PACKAGE_PIN W7 [get_ports {message[0]}]
#     set_property IOSTANDARD LVCMOS33 [get_ports {message[0]}]
#set_property PACKAGE_PIN W6 [get_ports {message[1]}]
#     set_property IOSTANDARD LVCMOS33 [get_ports {message[1]}]
#set_property PACKAGE_PIN U8 [get_ports {message[2]}]
#     set_property IOSTANDARD LVCMOS33 [get_ports {message[2]}]
#set_property PACKAGE_PIN V8 [get_ports {message[3]}]
#     set_property IOSTANDARD LVCMOS33 [get_ports {message[3]}]
#set_property PACKAGE_PIN U5 [get_ports {message[4]}]
#     set_property IOSTANDARD LVCMOS33 [get_ports {message[4]}]
#set_property PACKAGE_PIN V5 [get_ports {message[5]}]
#     set_property IOSTANDARD LVCMOS33 [get_ports {message[5]}]
#set_property PACKAGE_PIN U7 [get_ports {message[6]}]
#     set_property IOSTANDARD LVCMOS33 [get_ports {message[6]}]

#set_property PACKAGE_PIN V7 [get_ports dp]

#     set_property IOSTANDARD LVCMOS33 [get_ports dp]

#set_property PACKAGE_PIN U2 [get_ports {an[0]}]
#     set_property IOSTANDARD LVCMOS33 [get_ports {an[0]}]
#set_property PACKAGE_PIN U4 [get_ports {an[1]}]
#     set_property IOSTANDARD LVCMOS33 [get_ports {an[1]}]
#set_property PACKAGE_PIN V4 [get_ports {an[2]}]
#     set_property IOSTANDARD LVCMOS33 [get_ports {an[2]}]
#set_property PACKAGE_PIN W4 [get_ports {an[3]}]
#     set_property IOSTANDARD LVCMOS33 [get_ports {an[3]}]


###Buttons
```

```
##set_property PACKAGE_PIN U18 [get_ports btnC]
#     #set_property IOSTANDARD LVCMOS33 [get_ports btnC]
##set_property PACKAGE_PIN T18 [get_ports btnU]
#     #set_property IOSTANDARD LVCMOS33 [get_ports btnU]
##set_property PACKAGE_PIN W19 [get_ports btnL]
#     #set_property IOSTANDARD LVCMOS33 [get_ports btnL]
##set_property PACKAGE_PIN T17 [get_ports btnR]
#     #set_property IOSTANDARD LVCMOS33 [get_ports btnR]
##set_property PACKAGE_PIN U17 [get_ports btnD]
#     #set_property IOSTANDARD LVCMOS33 [get_ports btnD]



###Pmod Header JA
###Sch name = JA1
#set_property PACKAGE_PIN J1 [get_ports {col[3]}]
#     set_property IOSTANDARD LVCMOS33 [get_ports {col[3]}]
###Sch name = JA2
#set_property PACKAGE_PIN L2 [get_ports {col[2]}]
#     set_property IOSTANDARD LVCMOS33 [get_ports {col[2]}]
###Sch name = JA3
#set_property PACKAGE_PIN J2 [get_ports {col[1]}]
#     set_property IOSTANDARD LVCMOS33 [get_ports {col[1]}]
###Sch name = JA4
#set_property PACKAGE_PIN G2 [get_ports {col[0]}]
#     set_property IOSTANDARD LVCMOS33 [get_ports {col[0]}]
###Sch name = JA7
#set_property PACKAGE_PIN H1 [get_ports {row[3]}]
#     set_property IOSTANDARD LVCMOS33 [get_ports {row[3]}]
###Sch name = JA8
#set_property PACKAGE_PIN K2 [get_ports {row[2]}]
#     set_property IOSTANDARD LVCMOS33 [get_ports {row[2]}]
###Sch name = JA9
#set_property PACKAGE_PIN H2 [get_ports {row[1]}]
#     set_property IOSTANDARD LVCMOS33 [get_ports {row[1]}]
###Sch name = JA10
#set_property PACKAGE_PIN G3 [get_ports {row[0]}]
#     set_property IOSTANDARD LVCMOS33 [get_ports {row[0]}]



###Pmod Header JB
###Sch name = JB1
#set_property PACKAGE_PIN A14 [get_ports {DB[0]}]
#     set_property IOSTANDARD LVCMOS33 [get_ports {DB[0]}]
####Sch name = JB2
#set_property PACKAGE_PIN A16 [get_ports {DB[1]}]
#     set_property IOSTANDARD LVCMOS33 [get_ports {DB[1]}]
####Sch name = JB3
#set_property PACKAGE_PIN B15 [get_ports {DB[2]}]
#     set_property IOSTANDARD LVCMOS33 [get_ports {DB[2]}]
####Sch name = JB4
#set_property PACKAGE_PIN B16 [get_ports {DB[3]}]
#     set_property IOSTANDARD LVCMOS33 [get_ports {DB[3]}]
####Sch name = JB7
#set_property PACKAGE_PIN A15 [get_ports {DB[4]}]
#     set_property IOSTANDARD LVCMOS33 [get_ports {DB[4]}]
####Sch name = JB8
#set_property PACKAGE_PIN A17 [get_ports {DB[5]}]
#     set_property IOSTANDARD LVCMOS33 [get_ports {DB[5]}]
####Sch name = JB9
#set_property PACKAGE_PIN C15 [get_ports {DB[6]}]
#     set_property IOSTANDARD LVCMOS33 [get_ports {DB[6]}]
####Sch name = JB10
```

```
#set_property PACKAGE_PIN C16 [get_ports {DB[7]}]
#     set_property IOSTANDARD LVCMOS33 [get_ports {DB[7]}]



###Pmod Header JC
###Sch name = JC1
#set_property PACKAGE_PIN K17 [get_ports {JC[0]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {JC[0]}]
###Sch name = JC2
##set_property PACKAGE_PIN M18 [get_ports {JC[1]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {JC[1]}]
###Sch name = JC3
##set_property PACKAGE_PIN N17 [get_ports {JC[2]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {JC[2]}]
###Sch name = JC4
##set_property PACKAGE_PIN P18 [get_ports {JC[3]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {JC[3]}]
###Sch name = JC7
set_property PACKAGE_PIN L17 [get_ports {motor[0]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {motor[0]}]
###Sch name = JC8
set_property PACKAGE_PIN M19 [get_ports {motor[1]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {motor[1]}]
###Sch name = JC9
set_property PACKAGE_PIN P17 [get_ports {motor[2]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {motor[2]}]
###Sch name = JC10
set_property PACKAGE_PIN R18 [get_ports {motor[3]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {motor[3]}]


###Pmod Header JXADC
###Sch name = XA1_P
##set_property PACKAGE_PIN J3 [get_ports {JXADC[0]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[0]}]
###Sch name = XA2_P
##set_property PACKAGE_PIN L3 [get_ports {JXADC[1]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[1]}]
###Sch name = XA3_P
##set_property PACKAGE_PIN M2 [get_ports {JXADC[2]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[2]}]
###Sch name = XA4_P
##set_property PACKAGE_PIN N2 [get_ports {JXADC[3]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[3]}]
###Sch name = XA1_N
##set_property PACKAGE_PIN K3 [get_ports {JXADC[4]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[4]}]
###Sch name = XA2_N
##set_property PACKAGE_PIN M3 [get_ports {JXADC[5]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[5]}]
###Sch name = XA3_N
##set_property PACKAGE_PIN M1 [get_ports {JXADC[6]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[6]}]
###Sch name = XA4_N
##set_property PACKAGE_PIN N1 [get_ports {JXADC[7]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[7]}]



###VGA Connector
##set_property PACKAGE_PIN G19 [get_ports {vgaRed[0]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {vgaRed[0]}]
##set_property PACKAGE_PIN H19 [get_ports {vgaRed[1]}]
```

```
#     #set_property IOSTANDARD LVCMOS33 [get_ports {vgaRed[1]}]
##set_property PACKAGE_PIN J19 [get_ports {vgaRed[2]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {vgaRed[2]}]
##set_property PACKAGE_PIN N19 [get_ports {vgaRed[3]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {vgaRed[3]}]
##set_property PACKAGE_PIN N18 [get_ports {vgaBlue[0]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {vgaBlue[0]}]
##set_property PACKAGE_PIN L18 [get_ports {vgaBlue[1]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {vgaBlue[1]}]
##set_property PACKAGE_PIN K18 [get_ports {vgaBlue[2]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {vgaBlue[2]}]
##set_property PACKAGE_PIN J18 [get_ports {vgaBlue[3]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {vgaBlue[3]}]
##set_property PACKAGE_PIN J17 [get_ports {vgaGreen[0]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {vgaGreen[0]}]
##set_property PACKAGE_PIN H17 [get_ports {vgaGreen[1]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {vgaGreen[1]}]
##set_property PACKAGE_PIN G17 [get_ports {vgaGreen[2]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {vgaGreen[2]}]
##set_property PACKAGE_PIN D17 [get_ports {vgaGreen[3]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {vgaGreen[3]}]
##set_property PACKAGE_PIN P19 [get_ports Hsync]

#     #set_property IOSTANDARD LVCMOS33 [get_ports Hsync]
##set_property PACKAGE_PIN R19 [get_ports Vsync]

#     #set_property IOSTANDARD LVCMOS33 [get_ports Vsync]


###USB-RS232 Interface
##set_property PACKAGE_PIN B18 [get_ports RsRx]
#     #set_property IOSTANDARD LVCMOS33 [get_ports RsRx]
##set_property PACKAGE_PIN A18 [get_ports RsTx]
#     #set_property IOSTANDARD LVCMOS33 [get_ports RsTx]


###USB HID (PS/2)
##set_property PACKAGE_PIN C17 [get_ports PS2Clk]

#     #set_property IOSTANDARD LVCMOS33 [get_ports PS2Clk]
#     #set_property PULLUP true [get_ports PS2Clk]
##set_property PACKAGE_PIN B17 [get_ports PS2Data]
#     #set_property IOSTANDARD LVCMOS33 [get_ports PS2Data]
#     #set_property PULLUP true [get_ports PS2Data]


###Quad SPI Flash
###Note that CCLK_0 cannot be placed in 7 series devices. You can access it
using the
###STARTUPE2 primitive.
##set_property PACKAGE_PIN D18 [get_ports {QspiDB[0]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {QspiDB[0]}]
##set_property PACKAGE_PIN D19 [get_ports {QspiDB[1]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {QspiDB[1]}]
##set_property PACKAGE_PIN G18 [get_ports {QspiDB[2]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {QspiDB[2]}]
##set_property PACKAGE_PIN F18 [get_ports {QspiDB[3]}]
#     #set_property IOSTANDARD LVCMOS33 [get_ports {QspiDB[3]}]
##set_property PACKAGE_PIN K19 [get_ports QspiCSn]
#     #set_property IOSTANDARD LVCMOS33 [get_ports QspiCSn]
```

EXPERIMENT NO.23: DAC

1.SAWTOOTH

VERILOG CODE

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//
// Company:
// Engineer:
//
// Create Date: 11/08/2017 10:20:08 PM
// Design Name:
// Module Name: SawTooth
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
//


module saw_verilog (dac_out, clk, reset);
    input clk, reset;
   output [7:0] dac_out;

    reg [7:0] dac_out = 8'b0;
    reg [20:0] sclk;

    always@ (posedge clk)
        sclk <= sclk + 21'b1;

    always@ (posedge sclk[4], negedge reset)
    begin
        if (reset == 1'b0)
            dac_out <= 8'b0;
        else
            dac_out <= dac_out + 8'b1;
    end

endmodule
```

VHDL CODE

```
----------------------------------------------------------------------------------
--
-- Company:
-- Engineer:
--
-- Create Date: 11.11.2017 14:19:43
-- Design Name:
-- Module Name: sawtooth - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
```

```
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------------
--


library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;


entity sawtooth is
Port ( clk : in std_logic;
       reset : in std_logic;
       dac_out : out std_logic_vector(7 downto 0));
end sawtooth;
architecture Behavioral of sawtooth is
signal clk_div : std_logic_vector(3 downto 0);
signal counter : std_logic_vector(7 downto 0);
signal en :std_logic;
begin
process(clk)
begin
if rising_edge(clk) then
clk_div <= clk_div + '1' ;
end if;
end process;
process(clk_div (3))
begin
if reset='1' then
counter <= "00000000";
elsif rising_edge(clk_div (3)) then
counter <= counter + 1 ;
end if;
end process;
dac_out <=counter;
end Behavioral;
```

2.SQUARE

VERILOG CODE

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//
// Company:
// Engineer:
//
// Create Date: 11/07/2017 12:08:41 AM
// Design Name:
// Module Name: Square
// Project Name:
```

```
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
//
//Square wave generator.

module sqr_verilog(dac_out, clk, reset);

    input clk;
    input reset;
    output [7:0] dac_out;

    reg [7:0] dac_out = 8'b0;
    reg [6:0] counter = 7'b0;
    reg dir = 1'b0;
    reg [25:0] sclk;

    always@ (posedge clk)
        sclk <= sclk + 26'b1;

    always@ (posedge sclk[2], negedge reset)
    begin
        if (reset == 1'b0)
        begin
            dac_out <= 8'b0;
            counter <= 7'b0;
            dir <= 1'b0;
        end
        else
        begin
            counter = counter + 1'b1;
            if (counter == 7'd127)
                dir = ~(dir);
            if (dir == 1'b1)
                dac_out = 8'hFF;
            else
                dac_out = 8'h00;
        end
    end

endmodule

VHDL CODE
--------------------------------------------------------------------------------
--
-- Company:
-- Engineer:
--
-- Create Date: 11.11.2017 14:07:03
-- Design Name:
-- Module Name: square - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
```

```
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------------
--


library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
 use IEEE.STD_LOGIC_UNSIGNED.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;


entity square is
Port ( clk : in std_logic;
       reset : in std_logic;
       dac_out : out std_logic_vector(7 downto 0));
end square;
architecture Behavioral of square is
signal clk_div : std_logic_vector(3 downto 0);
signal counter : std_logic_vector(7 downto 0);
signal en :std_logic;
begin
process(clk)
begin
if rising_edge(clk) then
clk_div <= clk_div + '1' ;
end if;
end process;
process(clk_div(3))
begin
if reset='1' then
counter <= "00000000";
elsif rising_edge(clk_div(3)) then
if counter<255 and en='0' then
counter <= counter + 1 ;
en<='0';
dac_out <="00000000";
elsif counter=0 then
en<='0';
else en<='1';
counter <= counter-1;
dac_out <="11111111";
end if;
end if;
end process;
end Behavioral;
```

3.TRIANGULAR

VERILOG CODE

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//
// Company:
// Engineer:
//
// Create Date: 11/06/2017 11:25:30 PM
// Design Name:
// Module Name: Triangle
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
//
//Triangular wave generator.

module triangle (dac_out, clk, reset);

    input clk;
    input reset;
    output [7:0] dac_out;
    reg [7:0] dac_out;
    reg [25:0] sclk = 26'b0;
    //reg [7:0] counter;
    reg dir = 1'b0;

    always@ (posedge clk)
        sclk <= sclk + 26'b1;

    always@ (posedge sclk[3], negedge reset)
    begin
        if (reset == 1'b0)
            dac_out <= 8'b0;
        else
        begin
            if (dir == 1'b0)
            begin
                dac_out = dac_out + 8'b1;
                if (dac_out == 8'hFF)
                    dir = 1'b1;
            end
            else
            begin
                dac_out = dac_out - 8'b1;
                if (dac_out == 8'h00)
                    dir = 1'b0;
            end
        end
    end

endmodule
```

VHDL CODE

```
----------------------------------------------------------------------------------
--
-- Company:
-- Engineer:
--
-- Create Date: 11.11.2017 13:53:11
-- Design Name:
-- Module Name: triangle - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------------
--


library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
 use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

 entity triangle is
 Port ( clk : in std_logic;
        reset : in std_logic;
        dac_out: out std_logic_vector(7 downto 0));
 end triangle ;
architecture Behavioral of triangle is
 signal counter : std_logic_vector(8 downto 0);
 signal clk_div : std_logic_vector(25 downto 0);
 begin
 process(clk)
 begin
 if rising_edge(clk)
 then clk_div <= clk_div + '1' ;
 end if;
 end process;
  process(clk_div (3))
   begin
   if reset='1' then counter <= "111111110";
   elsif rising_edge(clk_div (3)) then counter <= counter + 1 ;
   if counter(8)='0' then dac_out <=counter(7 downto 0);
   else dac_out <=not(counter(7 downto 0));
   end if;
   end if;
   end process;
   end Behavioral;
```

4.SINE

VERILOG CODE

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//
// Company:
// Engineer:
//
// Create Date: 11/07/2017 12:26:23 AM
// Design Name:
// Module Name: Sine
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
//


module sine_verilog (dac_out, clk, reset);
    input clk;
    input reset;
    output [7:0] dac_out;

    reg [7:0] dac_out;
    reg [7:0] sclk;
    reg [7:0] index;

    parameter [7:0] sine [0:179] =
    {
        8'd128, 8'd132, 8'd136, 8'd141, 8'd145, 8'd150, 8'd154, 8'd158, 8'd163,
8'd167, 8'd171, 8'd175, 8'd179, 8'd183, 8'd187, 8'd191, 8'd195, 8'd199, 8'd202,
8'd206, 8'd209, 8'd213, 8'd216, 8'd219, 8'd222, 8'd225, 8'd228, 8'd231, 8'd233,
8'd236, 8'd238, 8'd240, 8'd242, 8'd244, 8'd246, 8'd247, 8'd249, 8'd250, 8'd251,
8'd252, 8'd253, 8'd254, 8'd254, 8'd255, 8'd255, 8'd255,
        8'd255, 8'd255, 8'd254, 8'd254, 8'd253, 8'd252, 8'd251, 8'd250, 8'd249,
8'd247, 8'd246, 8'd244, 8'd242, 8'd240, 8'd238, 8'd236, 8'd233, 8'd231, 8'd228,
8'd225, 8'd222, 8'd219, 8'd216, 8'd213, 8'd209, 8'd206, 8'd202, 8'd199, 8'd195,
8'd191, 8'd187, 8'd183, 8'd179, 8'd175, 8'd171, 8'd167, 8'd163, 8'd158, 8'd154,
8'd150, 8'd145, 8'd141, 8'd136, 8'd132, 8'd128, 8'd123,
        8'd119, 8'd114, 8'd110, 8'd105, 8'd101, 8'd97, 8'd92, 8'd88, 8'd84,
8'd80, 8'd76, 8'd72, 8'd68, 8'd64, 8'd60, 8'd56, 8'd53, 8'd49, 8'd46, 8'd42,
8'd39, 8'd36, 8'd33, 8'd30, 8'd27, 8'd24, 8'd22, 8'd19, 8'd17, 8'd15, 8'd13,
8'd11, 8'd9, 8'd8, 8'd6, 8'd5, 8'd4, 8'd3, 8'd2, 8'd1, 8'd1, 8'd0, 8'd0, 8'd0,
8'd0,
        8'd0, 8'd1, 8'd1, 8'd2, 8'd3, 8'd4, 8'd5, 8'd6, 8'd8, 8'd9, 8'd11,
8'd13, 8'd15, 8'd17, 8'd19, 8'd22, 8'd24, 8'd27, 8'd30, 8'd33, 8'd36, 8'd39,
8'd42, 8'd46, 8'd49, 8'd53, 8'd56, 8'd60, 8'd64, 8'd68, 8'd72, 8'd76, 8'd80,
8'd84, 8'd88, 8'd92, 8'd97, 8'd101, 8'd105, 8'd110, 8'd114, 8'd119, 8'd123
    };

    always@ (posedge clk, negedge reset)
    begin
        if (reset == 1'b0)
            sclk <= 8'b0;
        else
            sclk <= sclk + 8'b1;
```

```
    end

    always@ (posedge sclk[3])
    begin
        dac_out <= sine [index];
        index <= (index + 7'b1) % 180;
    end

endmodule

VHDL CODE
----------------------------------------------------------------------------------
--
-- Company:
-- Engineer:
--
-- Create Date: 11.11.2017 14:14:06
-- Design Name:
-- Module Name: sinewave - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------------
--


library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;


Entity sinewave is
port ( clk : in std_logic;
       reset : in std_logic;
       dac_out : out std_logic_vector(7 downto 0));
End sinewave;
architecture Behavioral of sinewave is
Signal clk_div:std_logic_vector(7 downto 0);
Signal I :integer range 0 to 179;
Type sine is array (0 to 179) of integer range 0 to 255;
Constant value:
sine:=(128,132,136,141,154,150,154,158,163,167,171,175,180,184,188,192,195,199,2
03,206,210,213,216,220,223,226,228,231,234,236,238,241,243,244,246,247,248,249,2
50,251,252,253,254,255,255,255,255,255,254,254,253,252,251,249,246,244,243,241,2
38,236,234,231,228,226,223,220,216,213,210,206,203,199,195,192,188,184,180,175,1
71,167,163,158,154,150,145,141,136,132,128,123,119,114,110,105,101,97,92,88,84,8
```

```
0,75,71,67,64,60,56,52,49,45,42,39,35,32,29,27,24,21,19,17,14,12,11,9,7,6,4,3,2,
1,1,0,0,0,0,0,0,0,0,1,1,2,3,4,6,7,9,11,12,14,17,19,21,24,27,29,32,35,39,42,45,49
,52,56,60,64,67,71,75,80,84,88,92,97,101,105,110,114,119,123,128);
begin
Process(clk,reset)
begin
if(reset='1') then
clk_div <=(others=>'0');
elsif(clk'event and clk='1') then
clk_div <= clk_div +1;
end if;
end process;
process(clk_div (3))
begin
if(clk_div (3)'event and clk_div (3)='1')then
dac_out<=conv_std_logic_vector(value(i),8);
I<=I+1;
if(i=179) then i<=0;
end if;
end if;
end process;
end Behavioral;
```

XDC FILE

```
## This file is a general .xdc for the Basys3 rev B board
## To use it in a project:
## - uncomment the lines corresponding to used pins
## - rename the used ports (in each line, after get_ports) according to the top
level signal names in the project

## Clock signal
set_property PACKAGE_PIN W5 [get_ports clk]

      set_property IOSTANDARD LVCMOS33 [get_ports clk]
      #create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5}
[get_ports clk]

## Switches
set_property PACKAGE_PIN V17 [get_ports {reset}]
      set_property IOSTANDARD LVCMOS33 [get_ports {reset}]
#set_property PACKAGE_PIN V16 [get_ports {sw[1]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[1]}]
#set_property PACKAGE_PIN W16 [get_ports {sw[2]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[2]}]
#set_property PACKAGE_PIN W17 [get_ports {sw[3]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[3]}]
#set_property PACKAGE_PIN W15 [get_ports {sw[4]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[4]}]
#set_property PACKAGE_PIN V15 [get_ports {sw[5]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[5]}]
#set_property PACKAGE_PIN W14 [get_ports {sw[6]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[6]}]
#set_property PACKAGE_PIN W13 [get_ports {sw[7]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[7]}]
#set_property PACKAGE_PIN V2 [get_ports {sw[8]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[8]}]
#set_property PACKAGE_PIN T3 [get_ports {sw[9]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[9]}]
#set_property PACKAGE_PIN T2 [get_ports {sw[10]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[10]}]
#set_property PACKAGE_PIN R3 [get_ports {sw[11]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[11]}]
#set_property PACKAGE_PIN W2 [get_ports {sw[12]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[12]}]
```

```
#set_property PACKAGE_PIN U1 [get_ports {sw[13]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[13]}]
#set_property PACKAGE_PIN T1 [get_ports {sw[14]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[14]}]
#set_property PACKAGE_PIN R2 [get_ports {sw[15]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {sw[15]}]


## LEDs
#set_property PACKAGE_PIN U16 [get_ports {led[0]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[0]}]
#set_property PACKAGE_PIN E19 [get_ports {led[1]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[1]}]
#set_property PACKAGE_PIN U19 [get_ports {led[2]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[2]}]
#set_property PACKAGE_PIN V19 [get_ports {led[3]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[3]}]
#set_property PACKAGE_PIN W18 [get_ports {led[4]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[4]}]
#set_property PACKAGE_PIN U15 [get_ports {led[5]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[5]}]
#set_property PACKAGE_PIN U14 [get_ports {led[6]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[6]}]
#set_property PACKAGE_PIN V14 [get_ports {led[7]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[7]}]
#set_property PACKAGE_PIN V13 [get_ports {led[8]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[8]}]
#set_property PACKAGE_PIN V3 [get_ports {led[9]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[9]}]
#set_property PACKAGE_PIN W3 [get_ports {led[10]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[10]}]
#set_property PACKAGE_PIN U3 [get_ports {led[11]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[11]}]
#set_property PACKAGE_PIN P3 [get_ports {led[12]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[12]}]
#set_property PACKAGE_PIN N3 [get_ports {led[13]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[13]}]
#set_property PACKAGE_PIN P1 [get_ports {led[14]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[14]}]
#set_property PACKAGE_PIN L1 [get_ports {led[15]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {led[15]}]


##7 segment display
#set_property PACKAGE_PIN W7 [get_ports {seg[0]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {seg[0]}]
#set_property PACKAGE_PIN W6 [get_ports {seg[1]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {seg[1]}]
#set_property PACKAGE_PIN U8 [get_ports {seg[2]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {seg[2]}]
#set_property PACKAGE_PIN V8 [get_ports {seg[3]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {seg[3]}]
#set_property PACKAGE_PIN U5 [get_ports {seg[4]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {seg[4]}]
#set_property PACKAGE_PIN V5 [get_ports {seg[5]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {seg[5]}]
#set_property PACKAGE_PIN U7 [get_ports {seg[6]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {seg[6]}]

#set_property PACKAGE_PIN V7 [get_ports dp]

      #set_property IOSTANDARD LVCMOS33 [get_ports dp]

#set_property PACKAGE_PIN U2 [get_ports {an[0]}]
```

```
      #set_property IOSTANDARD LVCMOS33 [get_ports {an[0]}]
#set_property PACKAGE_PIN U4 [get_ports {an[1]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {an[1]}]
#set_property PACKAGE_PIN V4 [get_ports {an[2]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {an[2]}]
#set_property PACKAGE_PIN W4 [get_ports {an[3]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {an[3]}]


##Buttons
#set_property PACKAGE_PIN U18 [get_ports btnC]
      #set_property IOSTANDARD LVCMOS33 [get_ports btnC]
#set_property PACKAGE_PIN T18 [get_ports btnU]
      #set_property IOSTANDARD LVCMOS33 [get_ports btnU]
#set_property PACKAGE_PIN W19 [get_ports btnL]
      #set_property IOSTANDARD LVCMOS33 [get_ports btnL]
#set_property PACKAGE_PIN T17 [get_ports btnR]
      #set_property IOSTANDARD LVCMOS33 [get_ports btnR]
#set_property PACKAGE_PIN U17 [get_ports btnD]
      #set_property IOSTANDARD LVCMOS33 [get_ports btnD]



##Pmod Header JA
##Sch name = JA1
#set_property PACKAGE_PIN J1 [get_ports {JA[0]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JA[0]}]
##Sch name = JA2
#set_property PACKAGE_PIN L2 [get_ports {JA[1]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JA[1]}]
##Sch name = JA3
#set_property PACKAGE_PIN J2 [get_ports {JA[2]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JA[2]}]
##Sch name = JA4
#set_property PACKAGE_PIN G2 [get_ports {JA[3]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JA[3]}]
##Sch name = JA7
#set_property PACKAGE_PIN H1 [get_ports {JA[4]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JA[4]}]
##Sch name = JA8
#set_property PACKAGE_PIN K2 [get_ports {JA[5]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JA[5]}]
##Sch name = JA9
#set_property PACKAGE_PIN H2 [get_ports {JA[6]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JA[6]}]
##Sch name = JA10
#set_property PACKAGE_PIN G3 [get_ports {JA[7]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JA[7]}]



##Pmod Header JB
##Sch name = JB1
#set_property PACKAGE_PIN A14 [get_ports {JB[0]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JB[0]}]
##Sch name = JB2
#set_property PACKAGE_PIN A16 [get_ports {JB[1]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JB[1]}]
##Sch name = JB3
#set_property PACKAGE_PIN B15 [get_ports {JB[2]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JB[2]}]
##Sch name = JB4
#set_property PACKAGE_PIN B16 [get_ports {JB[3]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JB[3]}]
```

```
##Sch name = JB7
#set_property PACKAGE_PIN A15 [get_ports {JB[4]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JB[4]}]
##Sch name = JB8
#set_property PACKAGE_PIN A17 [get_ports {JB[5]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JB[5]}]
##Sch name = JB9
#set_property PACKAGE_PIN C15 [get_ports {JB[6]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JB[6]}]
##Sch name = JB10
#set_property PACKAGE_PIN C16 [get_ports {JB[7]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JB[7]}]



##Pmod Header JC
##Sch name = JC1
set_property PACKAGE_PIN K17 [get_ports {dac_out[0]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {dac_out[0]}]
##Sch name = JC2
set_property PACKAGE_PIN M18 [get_ports {dac_out[1]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {dac_out[1]}]
##Sch name = JC3
set_property PACKAGE_PIN N17 [get_ports {dac_out[2]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {dac_out[2]}]
##Sch name = JC4
set_property PACKAGE_PIN P18 [get_ports {dac_out[3]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {dac_out[3]}]
##Sch name = JC7
set_property PACKAGE_PIN L17 [get_ports {dac_out[4]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {dac_out[4]}]
##Sch name = JC8
set_property PACKAGE_PIN M19 [get_ports {dac_out[5]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {dac_out[5]}]
##Sch name = JC9
set_property PACKAGE_PIN P17 [get_ports {dac_out[6]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {dac_out[6]}]
##Sch name = JC10
set_property PACKAGE_PIN R18 [get_ports {dac_out[7]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {dac_out[7]}]


##Pmod Header JXADC
##Sch name = XA1_P
#set_property PACKAGE_PIN J3 [get_ports {JXADC[0]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[0]}]
##Sch name = XA2_P
#set_property PACKAGE_PIN L3 [get_ports {JXADC[1]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[1]}]
##Sch name = XA3_P
#set_property PACKAGE_PIN M2 [get_ports {JXADC[2]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[2]}]
##Sch name = XA4_P
#set_property PACKAGE_PIN N2 [get_ports {JXADC[3]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[3]}]
##Sch name = XA1_N
#set_property PACKAGE_PIN K3 [get_ports {JXADC[4]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[4]}]
##Sch name = XA2_N
#set_property PACKAGE_PIN M3 [get_ports {JXADC[5]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[5]}]
##Sch name = XA3_N
#set_property PACKAGE_PIN M1 [get_ports {JXADC[6]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[6]}]
```

```
##Sch name = XA4_N
#set_property PACKAGE_PIN N1 [get_ports {JXADC[7]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {JXADC[7]}]



##VGA Connector
#set_property PACKAGE_PIN G19 [get_ports {vgaRed[0]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {vgaRed[0]}]
#set_property PACKAGE_PIN H19 [get_ports {vgaRed[1]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {vgaRed[1]}]
#set_property PACKAGE_PIN J19 [get_ports {vgaRed[2]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {vgaRed[2]}]
#set_property PACKAGE_PIN N19 [get_ports {vgaRed[3]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {vgaRed[3]}]
#set_property PACKAGE_PIN N18 [get_ports {vgaBlue[0]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {vgaBlue[0]}]
#set_property PACKAGE_PIN L18 [get_ports {vgaBlue[1]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {vgaBlue[1]}]
#set_property PACKAGE_PIN K18 [get_ports {vgaBlue[2]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {vgaBlue[2]}]
#set_property PACKAGE_PIN J18 [get_ports {vgaBlue[3]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {vgaBlue[3]}]
#set_property PACKAGE_PIN J17 [get_ports {vgaGreen[0]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {vgaGreen[0]}]
#set_property PACKAGE_PIN H17 [get_ports {vgaGreen[1]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {vgaGreen[1]}]
#set_property PACKAGE_PIN G17 [get_ports {vgaGreen[2]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {vgaGreen[2]}]
#set_property PACKAGE_PIN D17 [get_ports {vgaGreen[3]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {vgaGreen[3]}]
#set_property PACKAGE_PIN P19 [get_ports Hsync]
      #set_property IOSTANDARD LVCMOS33 [get_ports Hsync]
#set_property PACKAGE_PIN R19 [get_ports Vsync]
      #set_property IOSTANDARD LVCMOS33 [get_ports Vsync]


##USB-RS232 Interface
#set_property PACKAGE_PIN B18 [get_ports RsRx]
      #set_property IOSTANDARD LVCMOS33 [get_ports RsRx]
#set_property PACKAGE_PIN A18 [get_ports RsTx]
      #set_property IOSTANDARD LVCMOS33 [get_ports RsTx]


##USB HID (PS/2)
#set_property PACKAGE_PIN C17 [get_ports PS2Clk]

      #set_property IOSTANDARD LVCMOS33 [get_ports PS2Clk]
      #set_property PULLUP true [get_ports PS2Clk]
#set_property PACKAGE_PIN B17 [get_ports PS2Data]
      #set_property IOSTANDARD LVCMOS33 [get_ports PS2Data]
      #set_property PULLUP true [get_ports PS2Data]


##Quad SPI Flash
##Note that CCLK_0 cannot be placed in 7 series devices. You can access it using
the
##STARTUPE2 primitive.
#set_property PACKAGE_PIN D18 [get_ports {QspiDB[0]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {QspiDB[0]}]
#set_property PACKAGE_PIN D19 [get_ports {QspiDB[1]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {QspiDB[1]}]
#set_property PACKAGE_PIN G18 [get_ports {QspiDB[2]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {QspiDB[2]}]
```

```
#set_property PACKAGE_PIN F18 [get_ports {QspiDB[3]}]
      #set_property IOSTANDARD LVCMOS33 [get_ports {QspiDB[3]}]
#set_property PACKAGE_PIN K19 [get_ports QspiCSn]
      #set_property IOSTANDARD LVCMOS33 [get_ports QspiCSn]
```