VLSI LAB REPORT

Complete Lab Report

Submitted By
Yutika Chandrashekhar Kulwe
CED15I017

Experiment no. 1:

Aim: To simulate and synthesize the verilog modules of the processor using Cadence Genus Tool.

Observation:

ADP and PDP analysis

Module Name	Area	Internal Power	Switching Power	Timing	ADP	PDP
ripple carry adder	597.92	5213.88	6540.15	1531.4	915654.688	10015585.71
Carry Look ahead adder	437.77	7151.83	12615.74	1011.2	442673.024	12757036.29
Prefix Adder	904.89	7522.71	26107.24	80	72391.2	2088579.2
floating Point Adder	4057.34	46186.17	94711.11	1506	6110354.04	142634931.7
floating Point Multiplier	8054.78	248604.65	315199.31	550	4430129	173359620.5
Wallace Tree Multiplier	7500.06	431019.33	589326.41	869.9	6524302.194	512655044.1
right shifter	1041.03	7869.29	16780.13	104	108267.12	1745133.52
right rotator	1041.03	8342.19	17620.91	104	108267.12	1832574.64
left shifter	1041.03	7866.58	16775.36	104	108267.12	1744637.44
left rotator	1041.03	8336.76	17611.69	104	108267.12	1831615.76
register	6072.66	18162.32	27915.43	250	1518165	6978857.5
ALU	4193.47	25931.15	37743.62	301	1262234.47	11360829.62
Processor	13592.08	23289.93	29355.05	289.5	3934907.16	8498286.975
Memory	1403.91	712.63	2184.48	381.8	536012.838	834034.464

Inference

As per area, CLA is better as compared to other adders since, the area required is less.

As per power, ripple carry adder is better as compared to other adders since, the area required is less.

On the basis of timing, prefix adder is better. Not because it is pipelined. Pipelining doesn't make it compute faster. It just makes the computation happen concurrently.

Wallace tree multiplier has the highest ADP and PDP.

Conclusion:

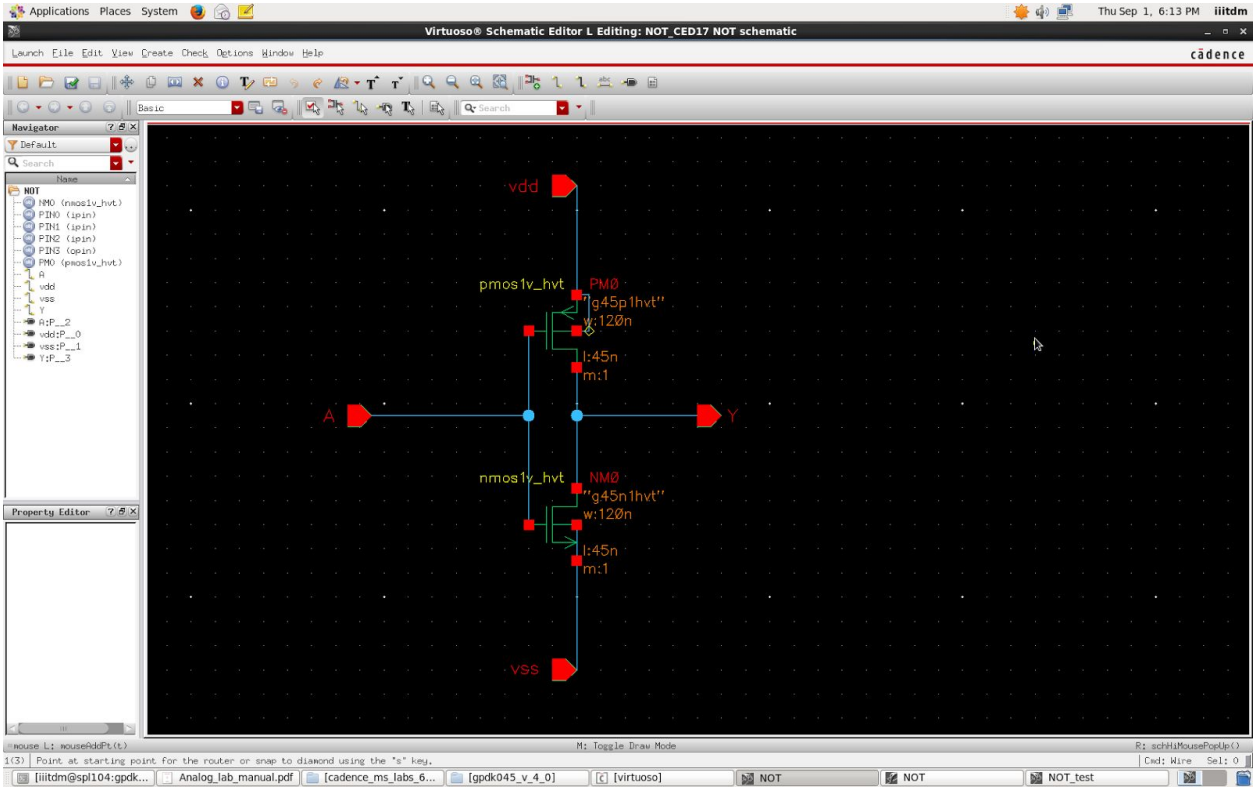
We learnt how the high level Verilog constructs gets transformed into low level logical constructs which can be modelled in the form of transistor logic.

Experiment no. 2:

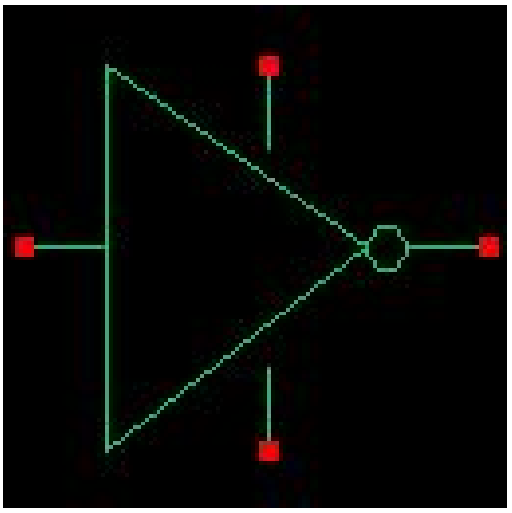
Aim: To learn the Virtuoso tool as well learn the flow of the Full
Custom IC design cycle. To do the DRC check for the designs.

Observation:

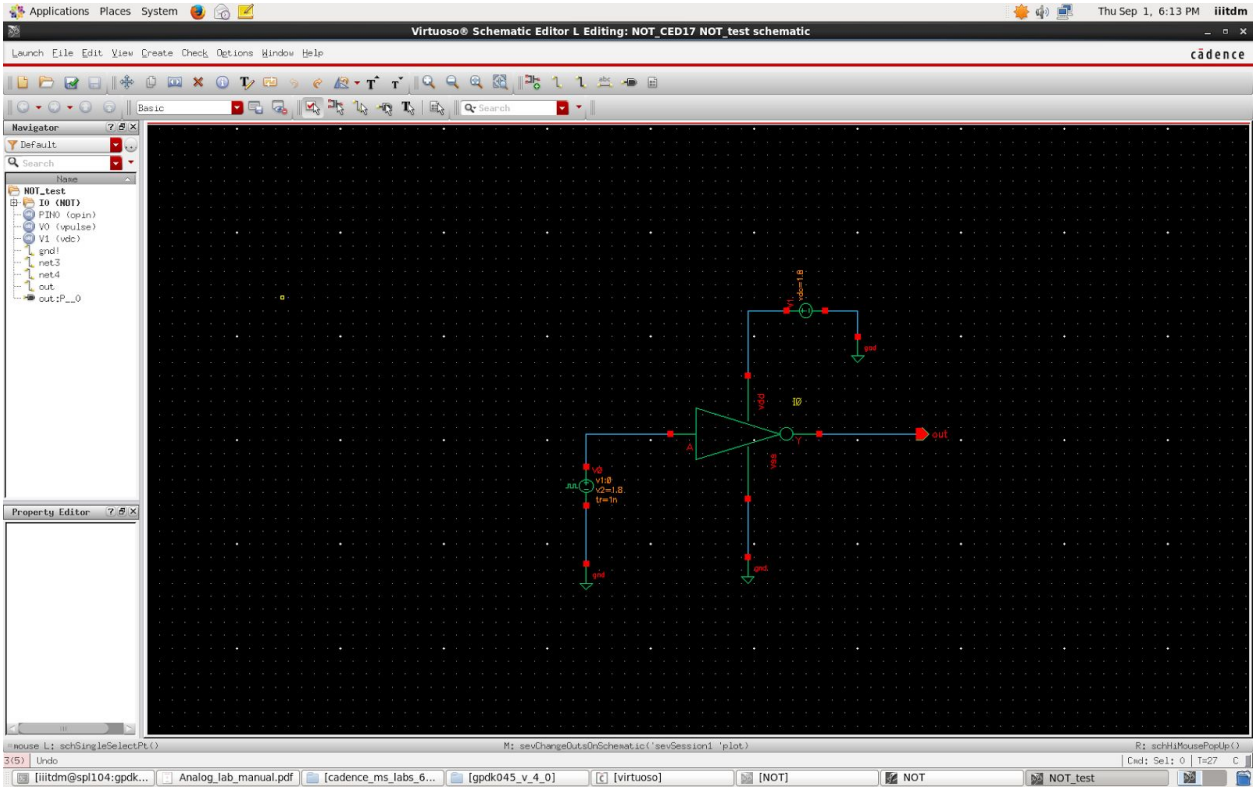
Inverter Gate Schematic



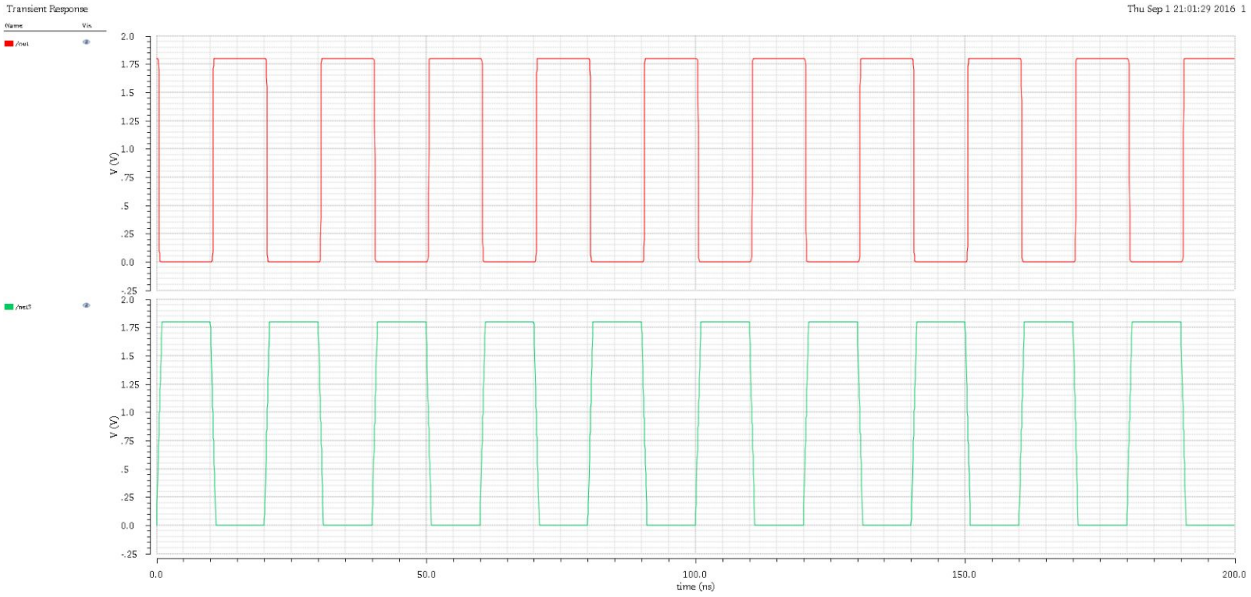
Inverter Gate Symbol



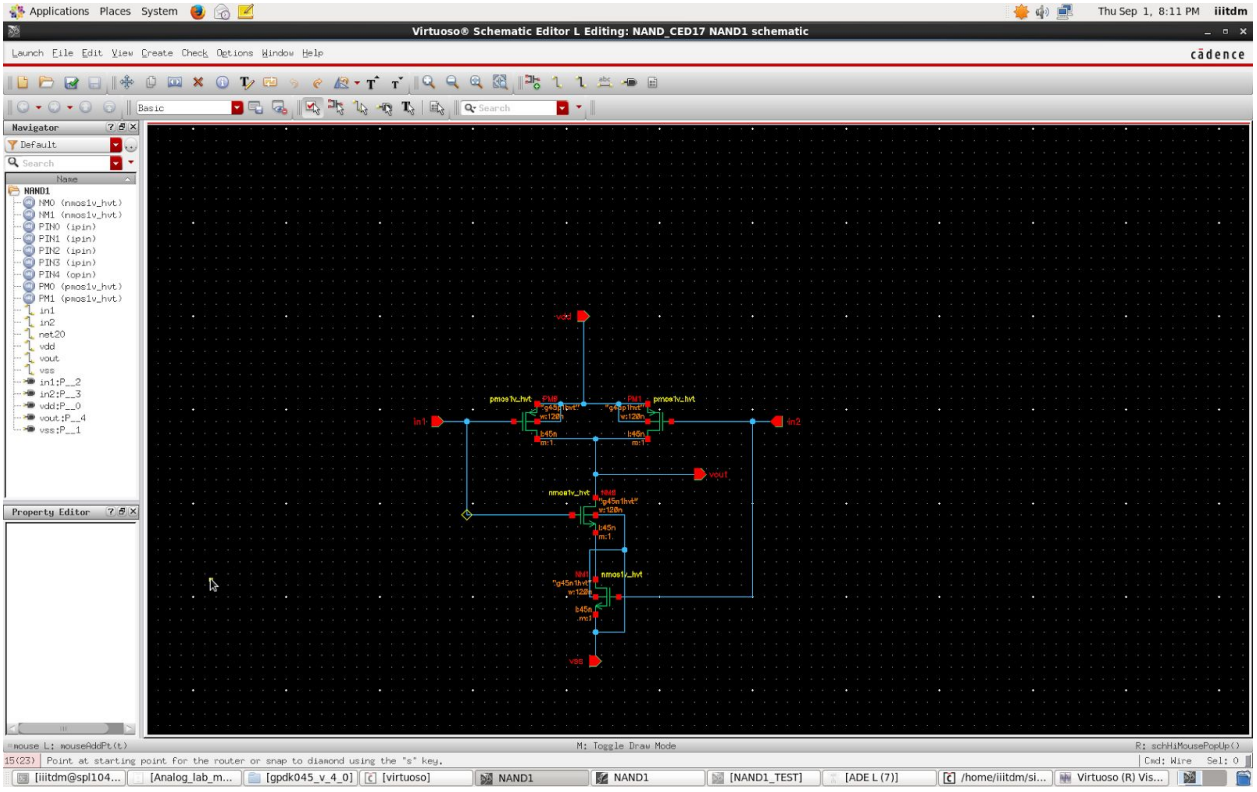
Inverter Gate Test



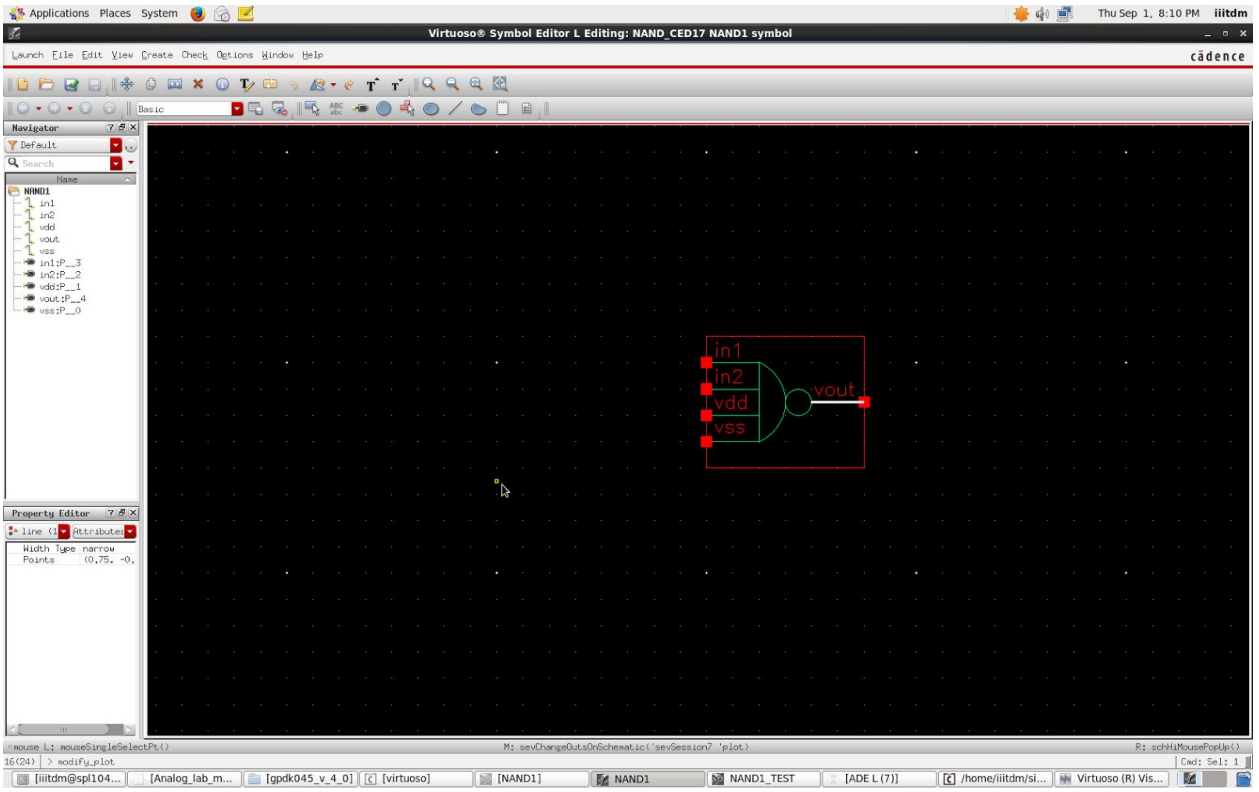
Inverter Gate Simulated Graph



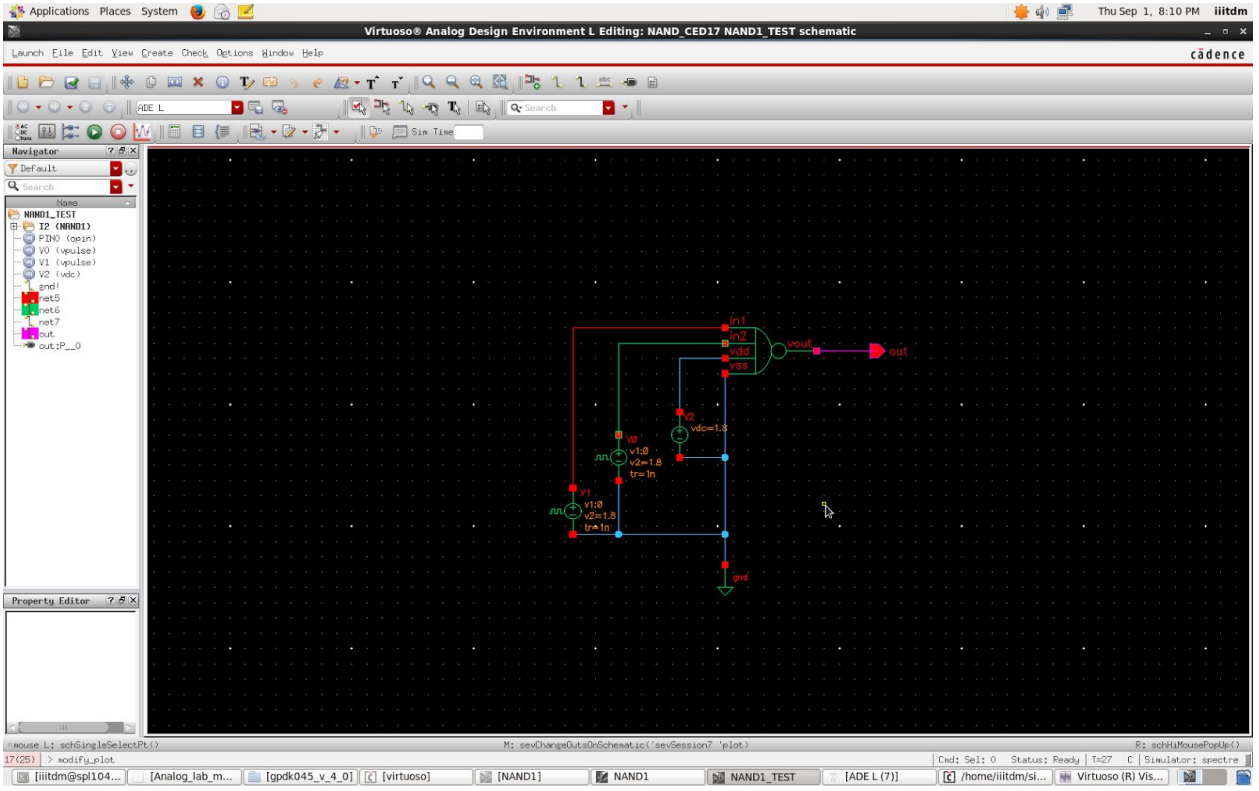
Nand Gate Schematic



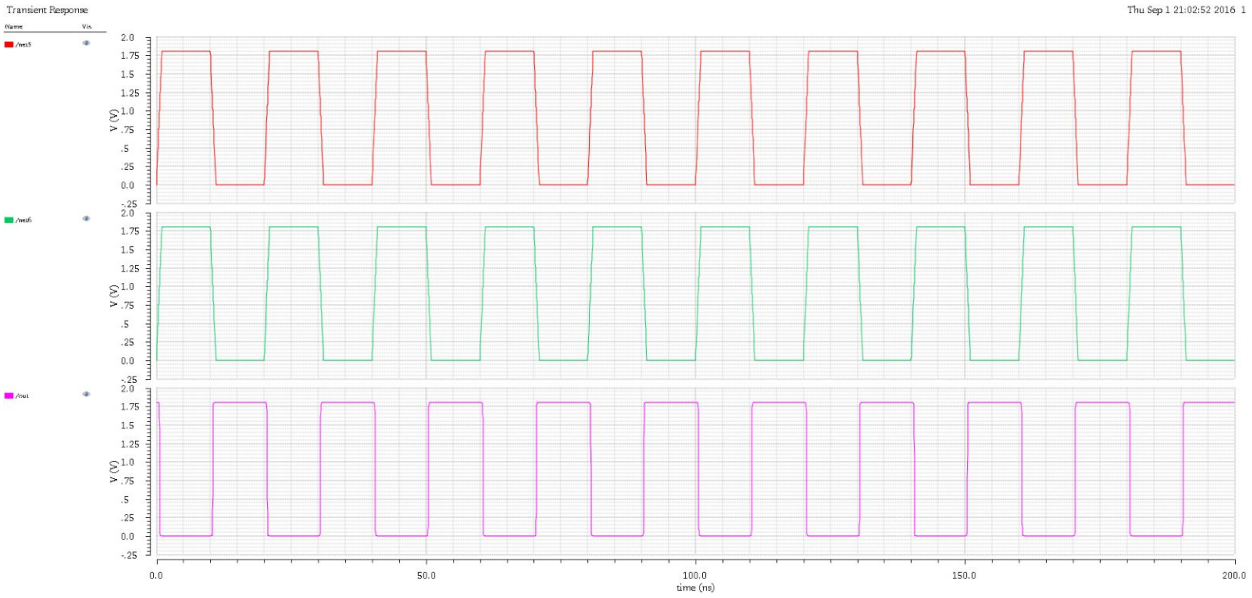
Nand Gate Symbol



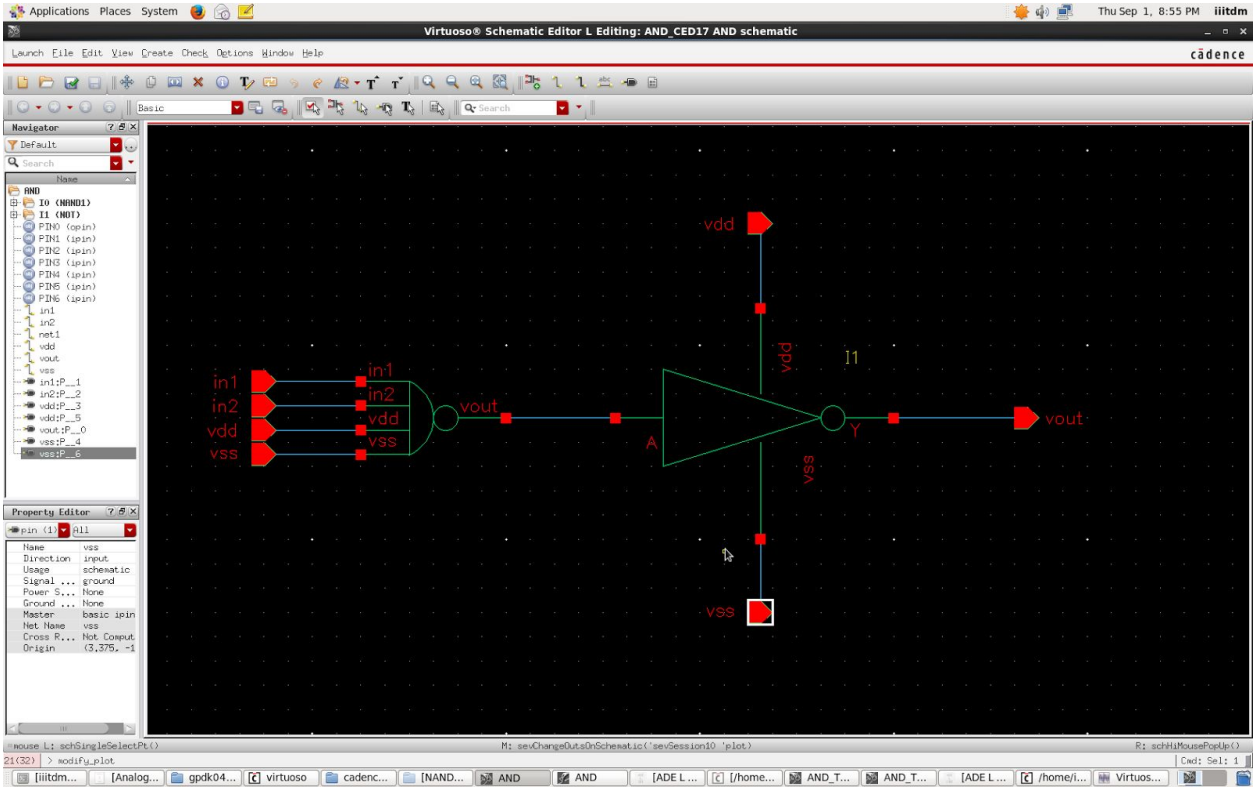
Nand Gate Test



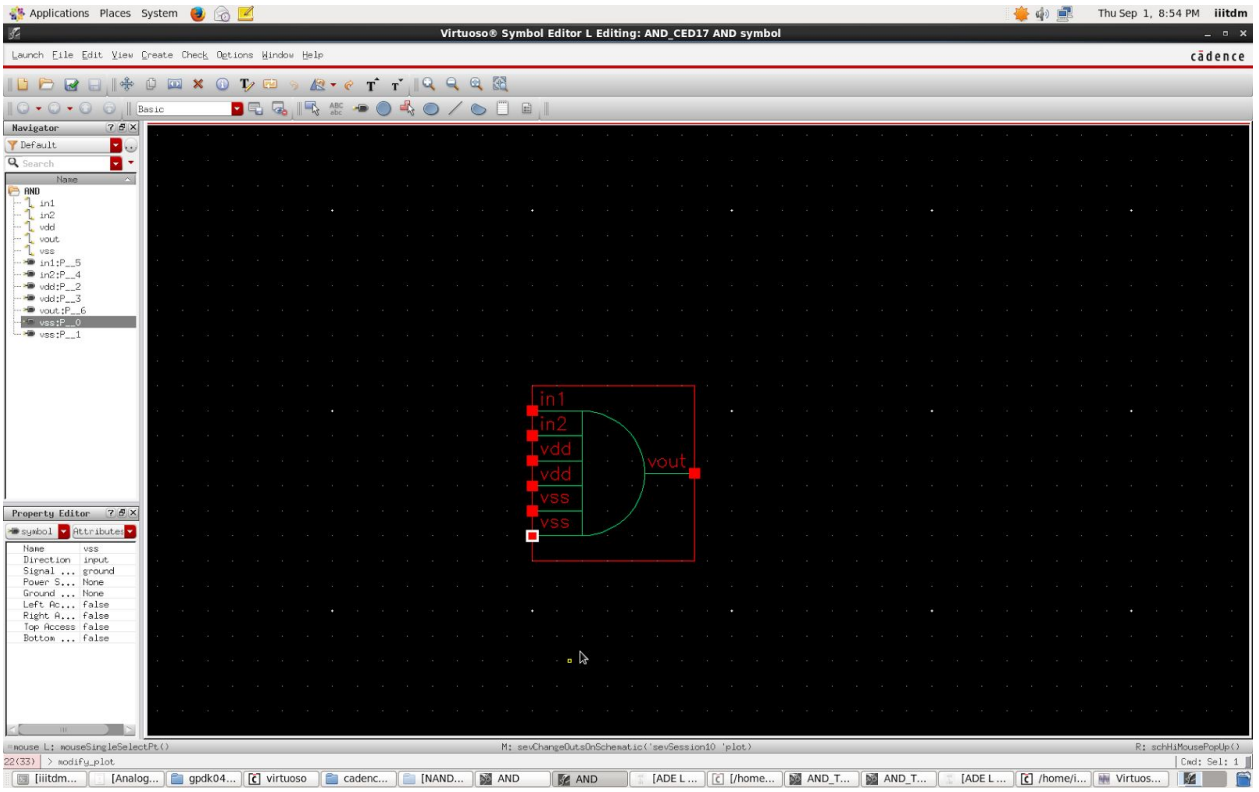
Nand Gate Simulated Graph



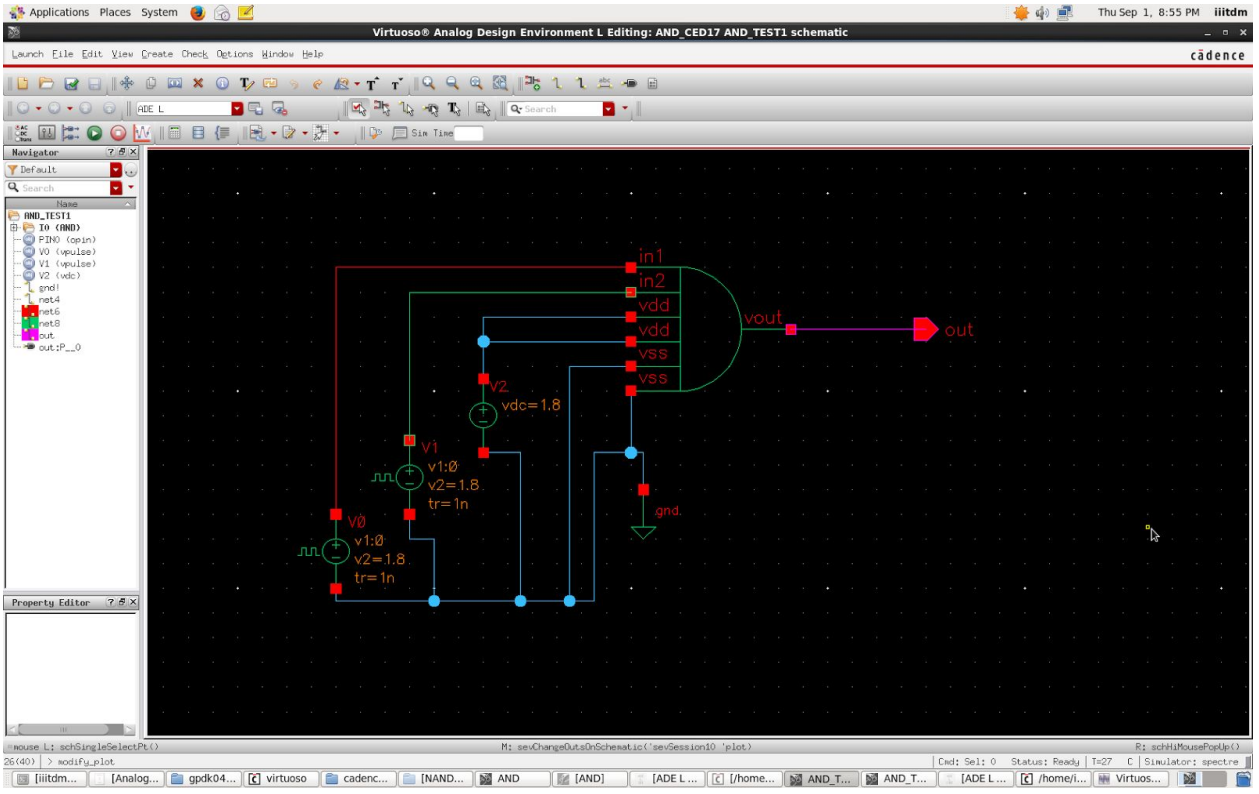
AND Gate Schematic



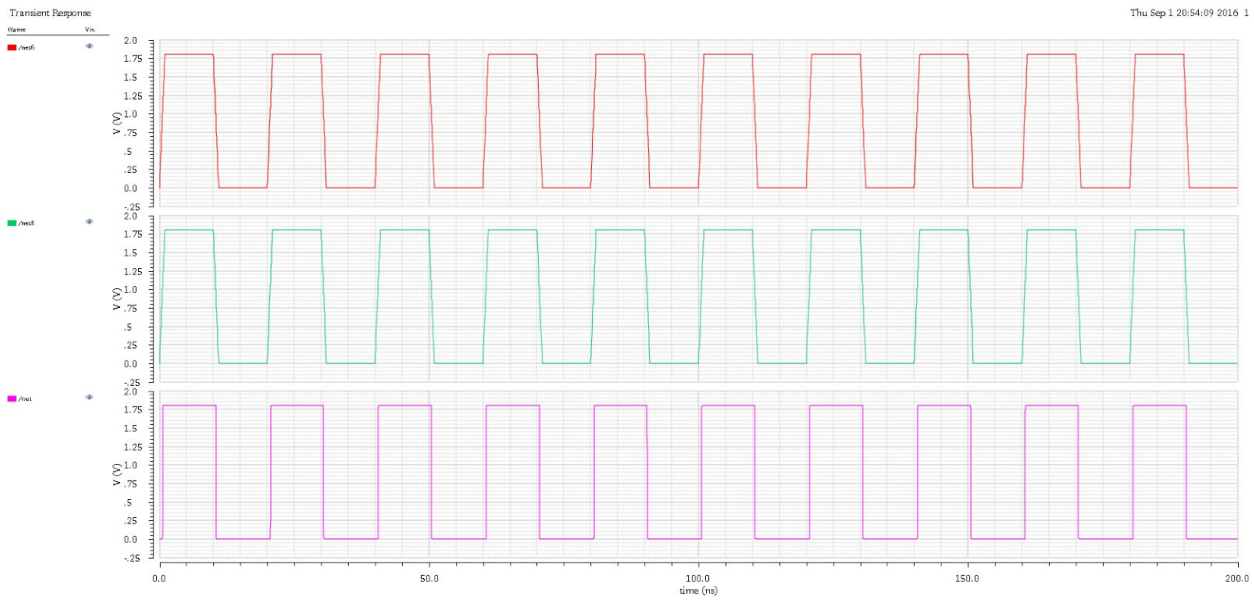
AND Gate Symbol



AND Gate Test



AND Gate Simulated Graph



<u>Conclusion</u>**:** We learnt to do the Schematic construction and simulation of NAND, NOR, AND gates.

Experiment no. 3:

<u>Aim</u>: To write a Python code that produces a Netlist for given algebraic expression of a CMOS Circuit.

<u>Code</u>:

```
class ssstack:
    def __init__(self):
      self.items = []

    def isEmpty(self):
      return self.items == []

    def push(self, item):
      self.items.append(item)

    def pop(self):
      return self.items.pop()
```

```
    def peek(self):
        return self.items[len(self.items)-1]

    def size(self):
        return len(self.items)

def infix_to_postfix():
        infx=str(input('Enter the infix expression : '))
        e=list(infx)
        print(e)
        pr = {}
        pr["!"] = 4
        pr["."] = 3
        pr["+"] = 2
        pr["("] = 1
        pr["$"] = 0
        postfix_list=[]
        opstack=ssstack()

        for ch in e:
                if ch in
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnop
qrstuvwxyz" :
                        postfix_list.append(ch)
                elif ch=='(':
                        opstack.push(ch)
                elif ch==')':
                        elem=opstack.pop()
                        while elem!='(':
                                postfix_list.append(elem)
                                elem=opstack.pop()
```

```
            else:
                while (not opstack.isEmpty()) and
(pr[opstack.peek()]>=pr[ch]):
                    postfix_list.append(opstack.pop())
                    #k=k+1
                opstack.push(ch)

        while not opstack.isEmpty():
            postfix_list.append(opstack.pop())
        Eval(postfix_list)
        return "".join(postfix_list)


def Eval(post_exp):
        nmosStack = ssstack()
        pmosStack = ssstack()
        e = list(post_exp)

        for ch in e:
            if ch in
"ABCDEFGHIJKLMNRSTUVWXYZabcdefghijklmnrstuvw
xyz":
                nmosStack.push(ch)
                pmosStack.push(ch)
            else:
                #nmos
                result = nmosOper(ch,nmosStack)
                nmosStack.push(result)
          #pmos
                result = pmosOper(ch,pmosStack)
                pmosStack.push(result)
```

```
        global out_counter
        print("OUTPUT O" + str(out_counter) )

out_counter=0
#input counters
x_counter=0
y_counter=0

#performing the NMOS operation
def nmosOper(op, nmosStack):
        global out_counter
        global x_counter
        global y_counter

        out_counter=out_counter+1
        output= "O" +str(out_counter)

        #nmos series connection
        if op==".":
                operand2=nmosStack.pop()
                operand1=nmosStack.pop()

                x_counter=x_counter+1
                y_counter=y_counter+1
                print("nmos ( GND, " + str(operand2) + ", X" +
str(x_counter) + " )")
                print("nmos ( X" +str(x_counter)+ " , " +
str(operand1) +", Y" + str(y_counter)+ " )")

                 # Inverted in(parallel)
                print("nmos ( GND, Y" + str(y_counter) + ", " +
str(output) + " )" )
```

```
    #nmos parallel connection
        elif op == "+":
                operand2=nmosStack.pop()
                operand1=nmosStack.pop()

                x_counter=x_counter+1
                y_counter=y_counter+1
                print("nmos ( GND, " + str(operand2) + ", Y" +
str(y_counter) + " )" )
                print("nmos ( GND, " + str(operand1) + ", Y" +
str(y_counter) + " )" )

        #inverted in (series)
                print("nmos ( GND, Y" + str(y_counter) + ", " +
str(output) + " )" )
        else: # not case
                operand1 = nmosStack.pop()

        # Invert output
                print("nmos ( GND, " + str(operand1) + ", " +
str(output) + " )" )
        return output

#performing the PMOS operaation
def pmosOper(op, pmosStack):
        global out_counter
        global x_counter
        global y_counter

        #out_counter=out_counter+1
        output= "O" +str(out_counter)
```

```
        if op==".":
                operand2=pmosStack.pop()
                operand1=pmosStack.pop()


                print("pmos ( VDD, " + str(operand2) + ", Y" +
str(y_counter) + " )")
                print("pmos ( VDD, " + str(operand1) + ", Y" +
str(y_counter) + " )")

                #Invert output
                print("pmos ( VDD, Y" + str(y_counter) + ", " +
str(output) + " )" )
        elif op == "+":
                operand2 = pmosStack.pop()
                operand1 = pmosStack.pop()

        # PMOS(S,G,D)
        # Series
                print("pmos ( VDD, " + str(operand2) + ", X" +
str(x_counter) + " )" )
                print("pmos ( X" + str(x_counter) + " , " +
str(operand1) + ", Y" + str(y_counter) + " )" )

        # Invert output
                print("pmos ( VDD, Y" + str(y_counter) + ", " +
str(output) + " )" )
        else: # not case
                operand1 = pmosStack.pop()

        # Invert output
```

```
            print("pmos ( VDD, " + str(operand1) + ", " +
str(output) + " )" )
        return output

#converting the given infix expression to postfix
k=infix_to_postfix()
#printing the postfix form of the given input
print("POSTFIX expression: " + k)
```

Output:

```
yutika@kulwe ~/VLSI LAB/lab3 $  Sat Nov-11 6:00:54pm
> python3 a5.py
Enter the infix expression : (a+b.c)!
['(', 'a', '+', 'b', '.', 'c', ')', '!']
nmos ( GND, c, X1 )
nmos ( X1 , b, Y1 )
nmos ( GND, Y1, O1 )
pmos ( VDD, c, Y1 )
pmos ( VDD, b, Y1 )
pmos ( VDD, Y1, O1 )
nmos ( GND, O1, Y2 )
nmos ( GND, a, Y2 )
nmos ( GND, Y2, O2 )
pmos ( VDD, O1, X2 )
pmos ( X2 , a, Y2 )
pmos ( VDD, Y2, O2 )
nmos ( GND, O2, O3 )
pmos ( VDD, O2, O3 )
OUTPUT O3
POSTFIX expression: abc.+!
yutika@kulwe ~/VLSI LAB/lab3 $  Sat Nov-11 6:00:54pm
```

Experiment no. 4:

Aim: To write a Python code that determines the state of every PMOS and NMOS Transistors present in a CMOS circuit for a given input.

Code:

```
class ssstack:
    def __init__(self):
        self.items = []

    def isEmpty(self):
        return self.items == []
```

```
    def push(self, item):
        self.items.append(item)

    def pop(self):
        return self.items.pop()

    def peek(self):
        return self.items[len(self.items)-1]

    def size(self):
        return len(self.items)

def infix_to_postfix():
        infx=str(input('Enter the infix expression : '))
        e=list(infx)
        print(e)
        pr = {}
        pr["!"] = 4
        pr["."] = 3
        pr["+"] = 2
        pr["("] = 1
        pr["$"] = 0
        postfix_list=[]
        opstack=ssstack()

        for ch in e:
                if ch in
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnop
qrstuvwxyz" :
                        postfix_list.append(ch)
                elif ch=='(':
```

```
                opstack.push(ch)
            elif ch==')':
                elem=opstack.pop()
                while elem!='(':
                    postfix_list.append(elem)
                    elem=opstack.pop()
            else:
                while (not opstack.isEmpty()) and
(pr[opstack.peek()]>=pr[ch]):
                    postfix_list.append(opstack.pop())
                    #k=k+1
                opstack.push(ch)

        while not opstack.isEmpty():
            postfix_list.append(opstack.pop())
        Eval(postfix_list)
        return "".join(postfix_list)


def Eval(post_exp):
        nmosStack = ssstack()
        pmosStack = ssstack()
        e = list(post_exp)

        for ch in e:
            if ch in
"ABCDEFGHIJKLMNRSTUVWXYZabcdefghijklmnrstuvw
xyz":
                nmosStack.push(ch)
                pmosStack.push(ch)
            else:
                #nmos
```

```
                result = nmosOper(ch,nmosStack)
                nmosStack.push(result)
        #pmos
                result = pmosOper(ch,pmosStack)
                pmosStack.push(result)

    global out_counter
    print("OUTPUT O" + str(out_counter) )

out_counter=0
#input counters
x_counter=0
y_counter=0

#performing the NMOS operation
def nmosOper(op, nmosStack):
    global out_counter
    global x_counter
    global y_counter

    out_counter=out_counter+1
    output= "O" +str(out_counter)

    #nmos series connection
    if op==".":
        operand2=nmosStack.pop()
        operand1=nmosStack.pop()

        x_counter=x_counter+1
        y_counter=y_counter+1
        print("nmos ( GND, " + str(operand2) + ", X" +
str(x_counter) + " )")
```

```
            print("nmos ( X" +str(x_counter)+ " , " +
str(operand1) +", Y" + str(y_counter)+ " )")

             # Inverted in(parallel)
            print("nmos ( GND, Y" + str(y_counter) + ", " +
str(output) + " )" )

   #nmos parallel connection
      elif op == "+":
            operand2=nmosStack.pop()
            operand1=nmosStack.pop()

            x_counter=x_counter+1
            y_counter=y_counter+1
            print("nmos ( GND, " + str(operand2) + ", Y" +
str(y_counter) + " )" )
            print("nmos ( GND, " + str(operand1) + ", Y" +
str(y_counter) + " )" )

      #inverted in (series)
            print("nmos ( GND, Y" + str(y_counter) + ", " +
str(output) + " )" )
      else: # not case
            operand1 = nmosStack.pop()

      # Invert output
            print("nmos ( GND, " + str(operand1) + ", " +
str(output) + " )" )
      return output

#performing the PMOS operaation
def pmosOper(op, pmosStack):
```

```
        global out_counter
        global x_counter
        global y_counter

        #out_counter=out_counter+1
        output= "O" +str(out_counter)

        if op==".":
                operand2=pmosStack.pop()
                operand1=pmosStack.pop()


                print("pmos ( VDD, " + str(operand2) + ", Y" +
str(y_counter) + " )")
                print("pmos ( VDD, " + str(operand1) + ", Y" +
str(y_counter) + " )")

                #Invert output
                print("pmos ( VDD, Y" + str(y_counter) + ", " +
str(output) + " )" )
        elif op == "+":
                operand2 = pmosStack.pop()
                operand1 = pmosStack.pop()

        # PMOS(S,G,D)
        # Series
                print("pmos ( VDD, " + str(operand2) + ", X" +
str(x_counter) + " )" )
                print("pmos ( X" + str(x_counter) + " , " +
str(operand1) + ", Y" + str(y_counter) + " )" )

        # Invert output
```

```
            print("pmos ( VDD, Y" + str(y_counter) + ", " +
str(output) + " )" )
        else: # not case
            operand1 = pmosStack.pop()

        # Invert output
            print("pmos ( VDD, " + str(operand1) + ", " +
str(output) + " )" )
        return output

#converting the given infix expression to postfix
k=infix_to_postfix()
#printing the postfix form of the given input
print("POSTFIX expression: " + k)
```

<u>Output</u>:

```
yutika@kulwe ~/VLSI LAB/lab3
File  Edit  View  Search  Terminal  Help
(A+b.c)!
['(', 'A', '+', 'b', '.', 'c', ')', '!']
Enter the values as 0|1 for the given operands

Enter the value for A
1
Enter the value for b
0
Enter the value for c
1
['(', '1', '+', '0', '.', '1', ')', '!']
nmos ( GND, 1, X1 )
ON

nmos ( X1 , 0, Y1 )
ON

nmos ( GND, Y1, O1 )
ON

pmos ( VDD, 1, Y1 )
OFF

pmos ( VDD, 0, Y1 )
OFF

pmos ( VDD, Y1, O1 )
OFF

nmos ( GND, 1, Y2 )
ON

nmos ( GND, 1, Y2 )
ON

nmos ( GND, Y2, O2 )
ON

pmos ( VDD, O1, X2 )
OFF

pmos ( X2 , 1, Y2 )
OFF

pmos ( VDD, Y2, O2 )
```

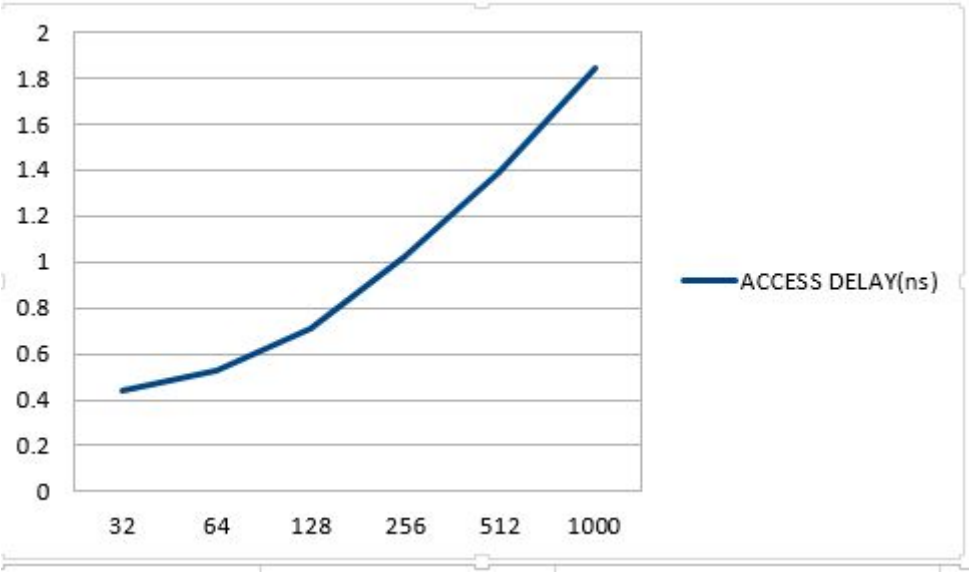
```
pmos ( X2 , 1, Y2 )
OFF

pmos ( VDD, Y2, O2 )
OFF

nmos ( GND, 1, O3 )
ON

pmos ( VDD, O2, O3 )
OFF

OUTPUT O3
POSTFIX expression: 101.+!
0
yutika@kulwe ~/VLSI LAB/lab3 $  Sat Nov-11 6:00:54pm
```

Experiment no. 5:

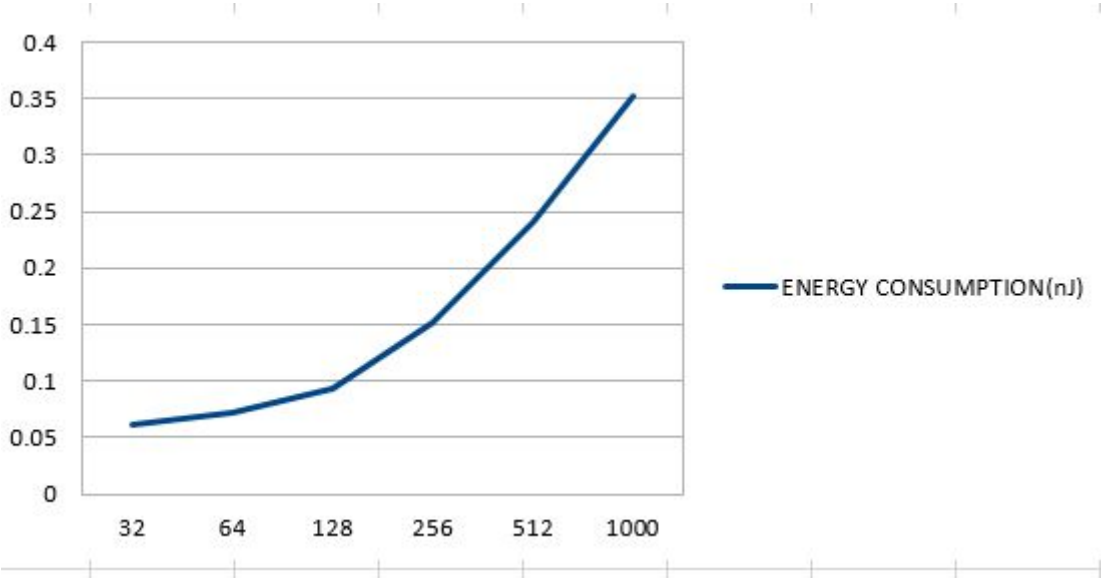
Aim: To plot the "Memory Size Vs Energy Consumption & Access Delay" for the following sizes -- 32 Kb, 64Kb, 128Kb, 256Kb, 512 Kb, and 1Mb SRAMs using the latest CACTI tool version.

Observation:

	SRAM		
MEMORY SIZE	ACCESS DELAY(ns)	ENERGY CONSUMPTION(nJ)	AVERAGE
32	0.44249	0.0616486	0.2520693
64	0.527409	0.0720062	0.2997076
128	0.712282	0.0931197	0.40270085
256	1.02563	0.152539	0.5890845
512	1.38959	0.240552	0.815071
1000	1.84607	0.352186	1.099128



Memory Size vs Access Delay



Memory Size vs Energy Consumption

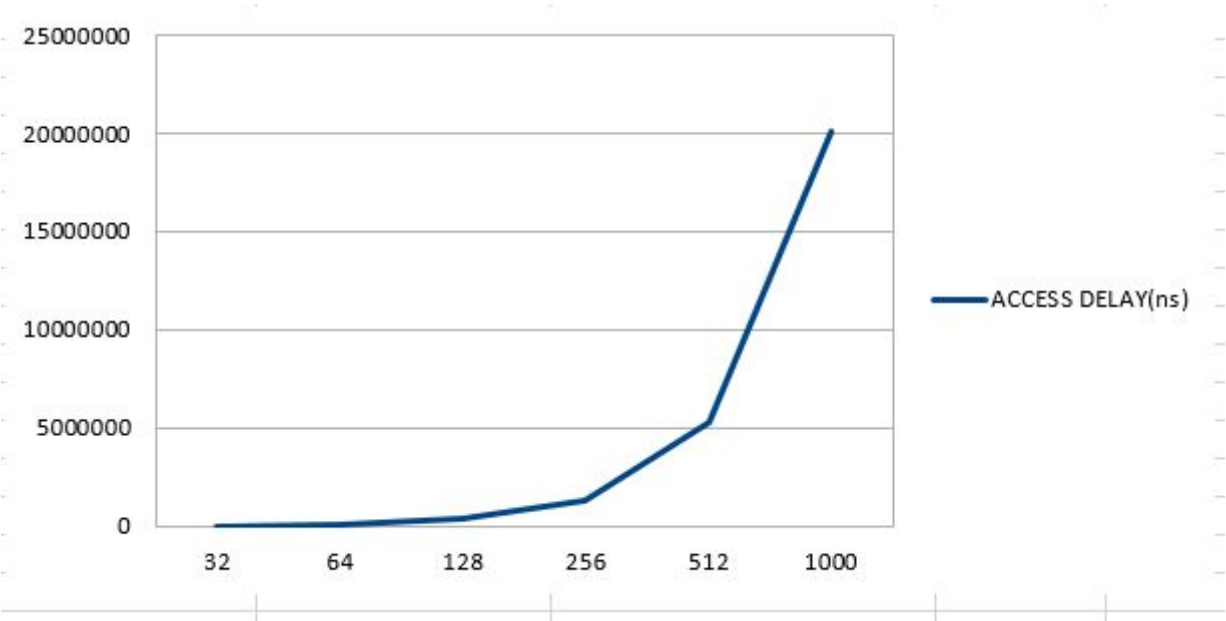
Conclusion: We studied the SRAM memory performance metrics in terms of power and delay primarily.

Experiment no. 6:

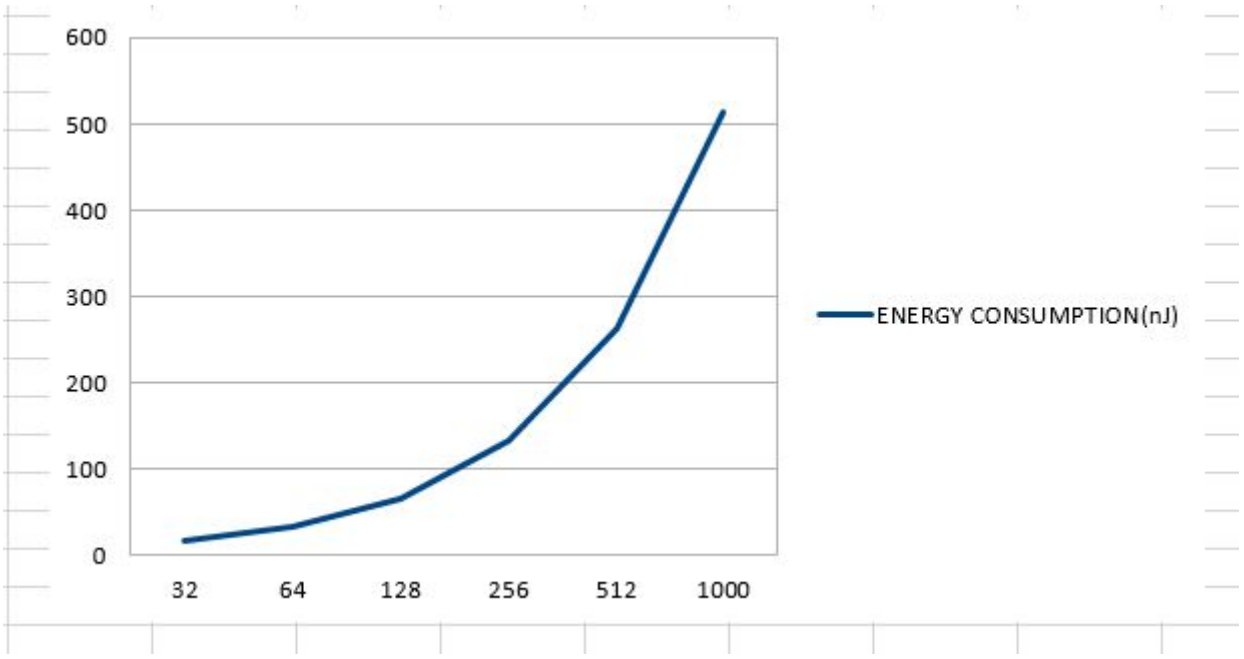
Aim: To plot the "Memory Size Vs Energy Consumption & Access Delay" for the following sizes -- 32 Kb, 64Kb, 128Kb, 256Kb, 512 Kb, and 1Mb SRAMs using the latest TCAM tool version.

Observation:

	TCAM		
MEMORY SIZE	ACCESS DELAY(ns)	ENERGY CONSUMPTION(nJ)	AVERAGE
32	20983.61802	16.443847	10500.03093
64	83126.53517	32.887318	41579.71125
128	330891.5184	65.774266	165478.6463
256	1320338.342	131.548148	660234.9452
512	5274900.235	263.095927	2637581.666
1000	20110135.9	513.858879	10055324.88



Memory Size vs Access Delay



Memory Size vs Energy Consumption

Conclusion: We assessed the TCAM performance using TCAM tool in terms of delay and power against their sizes.

Experiment no. 7:

Aim: To write a verilog code for Priority Encoder.

Code:

Priority4bit.v:

```
module PriorityEncoder_4Bit(d,y,v);
	input [3:0]d;
	output [3:0]y;
	output v;
	wire v;
	wire [3:0]y;
	assign y[0] = d[0] & 1'b1;
	assign y[1] = d[1] & ~d[0] & 1'b1;
	assign y[2] = d[2] & ~d[1] & ~d[0] & 1'b1;
	assign y[3] = d[3] & ~d[2] & ~d[1] & ~d[0] & 1'b1;
```

```
        assign v = d[0] | d[1] | d[2] | d[3];

endmodule
```

***Priority256bit.v**:*

```
// Usage: iverilog -o abc Priority256bit.v Priority4bit.v
Conversion_element.v
// vvp abc

module pe_256bit(d,y,v);
        input [255:0]d;
        output v;
        output [255:0]y;
        wire v;
        wire [255:0]y;
        wire [255:0]p,k,q,j,r,h,c,x,z;

        //first layer
        PriorityEncoder_4Bit    a1(d[ 3 : 0 ],y[ 3 : 0
  ],v1);
        PriorityEncoder_4Bit    a2(d[ 7 : 4 ],y[ 7
        : 4 ],v2);
        PriorityEncoder_4Bit    a3(d[ 11 : 8 ],y[ 11
        : 8 ],v3);
        PriorityEncoder_4Bit    a4(d[ 15 : 12 ],y[ 15
        : 12 ],v4);
        PriorityEncoder_4Bit    a5(d[ 19 : 16 ],y[ 19
        : 16 ],v5);
        PriorityEncoder_4Bit    a6(d[ 23 : 20 ],y[ 23
        : 20 ],v6);
        PriorityEncoder_4Bit    a7(d[ 27 : 24 ],y[ 27
        : 24 ],v7);
```

```
PriorityEncoder_4Bit a8(d[ 31 : 28 ],y[ 31 : 28 ],v8);
PriorityEncoder_4Bit a9(d[ 35 : 32 ],y[ 35 : 32 ],v9);
PriorityEncoder_4Bit a10(d[ 39 : 36 ],y[ 39 : 36 ],v10);
PriorityEncoder_4Bit a11(d[ 43 : 40 ],y[ 43 : 40 ],v11);
PriorityEncoder_4Bit a12(d[ 47 : 44 ],y[ 47 : 44 ],v12);
PriorityEncoder_4Bit a13(d[ 51 : 48 ],y[ 51 : 48 ],v13);
PriorityEncoder_4Bit a14(d[ 55 : 52 ],y[ 55 : 52 ],v14);
PriorityEncoder_4Bit a15(d[ 59 : 56 ],y[ 59 : 56 ],v15);
PriorityEncoder_4Bit a16(d[ 63 : 60 ],y[ 63 : 60 ],v16);
PriorityEncoder_4Bit a17(d[ 67 : 64 ],y[ 67 : 64 ],v17);
PriorityEncoder_4Bit a18(d[ 71 : 68 ],y[ 71 : 68 ],v18);
PriorityEncoder_4Bit a19(d[ 75 : 72 ],y[ 75 : 72 ],v19);
PriorityEncoder_4Bit a21(d[ 79 : 76 ],y[ 79 : 76 ],v20);
PriorityEncoder_4Bit a22(d[ 83 : 80 ],y[ 83 : 80 ],v21);
PriorityEncoder_4Bit a23(d[ 87 : 84 ],y[ 87 : 84 ],v22);
PriorityEncoder_4Bit a24(d[ 91 : 88 ],y[ 91 : 88 ],v23);
```

```
PriorityEncoder_4Bit a25(d[ 95 : 92 ],y[ 95 : 92 ],v24);
PriorityEncoder_4Bit a26(d[ 99 : 96 ],y[ 99 : 96 ],v25);
PriorityEncoder_4Bit a27(d[ 103 : 100 ],y[ 103 : 100 ],v26);
PriorityEncoder_4Bit a28(d[ 107 : 104 ],y[ 107 : 104 ],v27);
PriorityEncoder_4Bit a29(d[ 111 : 108 ],y[ 111 : 108 ],v28);
PriorityEncoder_4Bit a30(d[ 115 : 112 ],y[ 115 : 112 ],v29);
PriorityEncoder_4Bit a31(d[ 119 : 116 ],y[ 119 : 116 ],v30);
PriorityEncoder_4Bit a32(d[ 123 : 120 ],y[ 123 : 120 ],v31);
PriorityEncoder_4Bit a33(d[ 127 : 124 ],y[ 127 : 124 ],v32);
PriorityEncoder_4Bit a34(d[ 131 : 128 ],y[ 131 : 128 ],v33);
PriorityEncoder_4Bit a35(d[ 135 : 132 ],y[ 135 : 132 ],v34);
PriorityEncoder_4Bit a36(d[ 139 : 136 ],y[ 139 : 136 ],v35);
PriorityEncoder_4Bit a37(d[ 143 : 140 ],y[ 143 : 140 ],v36);
PriorityEncoder_4Bit a38(d[ 147 : 144 ],y[ 147 : 144 ],v37);
PriorityEncoder_4Bit a39(d[ 151 : 148 ],y[ 151 : 148 ],v38);
PriorityEncoder_4Bit a40(d[ 155 : 152 ],y[ 155 : 152 ],v39);
```

```
PriorityEncoder_4Bit a41(d[ 159 : 156 ],y[
159 : 156 ],v40);
PriorityEncoder_4Bit a42(d[ 163 : 160 ],y[
163 : 160 ],v41);
PriorityEncoder_4Bit a43(d[ 167 : 164 ],y[
167 : 164 ],v42);
PriorityEncoder_4Bit a44(d[ 171 : 168 ],y[
171 : 168 ],v43);
PriorityEncoder_4Bit a45(d[ 175 : 172 ],y[
175 : 172 ],v44);
PriorityEncoder_4Bit a46(d[ 179 : 176 ],y[
179 : 176 ],v45);
PriorityEncoder_4Bit a47(d[ 183 : 180 ],y[
183 : 180 ],v46);
PriorityEncoder_4Bit a48(d[ 187 : 184 ],y[
187 : 184 ],v47);
PriorityEncoder_4Bit a49(d[ 191 : 188 ],y[
191 : 188 ],v48);
PriorityEncoder_4Bit a50(d[ 195 : 192 ],y[
195 : 192 ],v49);
PriorityEncoder_4Bit a51(d[ 199 : 196 ],y[
199 : 196 ],v50);
PriorityEncoder_4Bit a52(d[ 203 : 200 ],y[
203 : 200 ],v51);
PriorityEncoder_4Bit a53(d[ 207 : 204 ],y[
207 : 204 ],v52);
PriorityEncoder_4Bit a54(d[ 211 : 208 ],y[
211 : 208 ],v53);
PriorityEncoder_4Bit a55(d[ 215 : 212 ],y[
215 : 212 ],v54);
PriorityEncoder_4Bit a56(d[ 219 : 216 ],y[
219 : 216 ],v55);
```

```
PriorityEncoder_4Bit a57(d[ 223 : 220 ],y[ 223 : 220 ],v56);
PriorityEncoder_4Bit a58(d[ 227 : 224 ],y[ 227 : 224 ],v57);
PriorityEncoder_4Bit a59(d[ 231 : 228 ],y[ 231 : 228 ],v58);
PriorityEncoder_4Bit a60(d[ 235 : 232 ],y[ 235 : 232 ],v59);
PriorityEncoder_4Bit a61(d[ 239 : 236 ],y[ 239 : 236 ],v60);
PriorityEncoder_4Bit a62(d[ 243 : 240 ],y[ 243 : 240 ],v61);
PriorityEncoder_4Bit a63(d[ 247 : 244 ],y[ 247 : 244 ],v62);
PriorityEncoder_4Bit a64(d[ 251 : 248 ],y[ 251 : 248 ],v63);
PriorityEncoder_4Bit a20(d[ 255 : 252 ],y[ 255 : 252 ],v64);

//second layer

assign p[ 0 ]=y[ 0 ] | y [ 1 ] | y [ 2 ] | y [ 3 ];
assign p[ 1 ]=y[ 4 ] | y [ 5 ] | y [ 6 ] | y [ 7 ];
assign p[ 2 ]=y[ 8 ] | y [ 9 ] | y [ 10 ] | y [ 11 ];
assign p[ 3 ]=y[ 12 ] | y [ 13 ] | y [ 14 ] | y [ 15 ];
assign p[ 4 ]=y[ 16 ] | y [ 17 ] | y [ 18 ] | y [ 19 ];
```

```
assign p[ 5 ]=y[ 20 ] | y [ 21 ] | y [ 22 ] |
y [ 23 ];
assign p[ 6 ]=y[ 24 ] | y [ 25 ] | y [ 26 ] |
y [ 27 ];
assign p[ 7 ]=y[ 28 ] | y [ 29 ] | y [ 30 ] |
y [ 31 ];
assign p[ 8 ]=y[ 32 ] | y [ 33 ] | y [ 34 ] |
y [ 35 ];
assign p[ 9 ]=y[ 36 ] | y [ 37 ] | y [ 38 ] |
y [ 39 ];
assign p[ 10 ]=y[ 40 ] | y [ 41 ] | y [ 42 ] |
y [ 43 ];
assign p[ 11 ]=y[ 44 ] | y [ 45 ] | y [ 46 ] |
y [ 47 ];
assign p[ 12 ]=y[ 48 ] | y [ 49 ] | y [ 50 ] |
y [ 51 ];
assign p[ 13 ]=y[ 52 ] | y [ 53 ] | y [ 54 ] |
y [ 55 ];
assign p[ 14 ]=y[ 56 ] | y [ 57 ] | y [ 58 ] |
y [ 59 ];
assign p[ 15 ]=y[ 60 ] | y [ 61 ] | y [ 62 ] |
y [ 63 ];
assign p[ 16 ]=y[ 64 ] | y [ 65 ] | y [ 66 ] |
y [ 67 ];
assign p[ 17 ]=y[ 68 ] | y [ 69 ] | y [ 70 ] |
y [ 71 ];
assign p[ 18 ]=y[ 72 ] | y [ 73 ] | y [ 74 ] |
y [ 75 ];
assign p[ 19 ]=y[ 76 ] | y [ 77 ] | y [ 78 ] |
y [ 79 ];
assign p[ 20 ]=y[ 80 ] | y [ 81 ] | y [ 82 ] |
y [ 83 ];
```

```
assign p[ 21 ]=y[ 84 ] | y [ 85 ] | y [ 86 ] |
y [ 87 ];
assign p[ 22 ]=y[ 88 ] | y [ 89 ] | y [ 90 ] |
y [ 91 ];
assign p[ 23 ]=y[ 92 ] | y [ 93 ] | y [ 94 ] |
y [ 95 ];
assign p[ 24 ]=y[ 96 ] | y [ 97 ] | y [ 98 ] |
y [ 99 ];
assign p[ 25 ]=y[ 100 ] | y [ 101 ] | y [ 102 ] |
y [ 103 ];
assign p[ 26 ]=y[ 104 ] | y [ 105 ] | y [ 106 ] |
y [ 107 ];
assign p[ 27 ]=y[ 108 ] | y [ 109 ] | y [ 110 ] |
y [ 111 ];
assign p[ 28 ]=y[ 112 ] | y [ 113 ] | y [ 114 ] |
y [ 115 ];
assign p[ 29 ]=y[ 116 ] | y [ 117 ] | y [ 118 ] |
y [ 119 ];
assign p[ 30 ]=y[ 120 ] | y [ 121 ] | y [ 122 ] |
y [ 123 ];
assign p[ 31 ]=y[ 124 ] | y [ 125 ] | y [ 126 ] |
y [ 127 ];
assign p[ 32 ]=y[ 128 ] | y [ 129 ] | y [ 130 ] |
y [ 131 ];
assign p[ 33 ]=y[ 132 ] | y [ 133 ] | y [ 134 ] |
y [ 135 ];
assign p[ 34 ]=y[ 136 ] | y [ 137 ] | y [ 138 ] |
y [ 139 ];
assign p[ 35 ]=y[ 140 ] | y [ 141 ] | y [ 142 ] |
y [ 143 ];
assign p[ 36 ]=y[ 144 ] | y [ 145 ] | y [ 146 ] |
y [ 147 ];
```

```
assign p[ 37 ]=y[ 148 ] | y [ 149 ] | y [ 150 ] |
y [ 151 ];
assign p[ 38 ]=y[ 152 ] | y [ 153 ] | y [ 154 ] |
y [ 155 ];
assign p[ 39 ]=y[ 156 ] | y [ 157 ] | y [ 158 ] |
y [ 159 ];
assign p[ 40 ]=y[ 160 ] | y [ 161 ] | y [ 162 ] |
y [ 163 ];
assign p[ 41 ]=y[ 164 ] | y [ 165 ] | y [ 166 ] |
y [ 167 ];
assign p[ 42 ]=y[ 168 ] | y [ 169 ] | y [ 170 ] |
y [ 171 ];
assign p[ 43 ]=y[ 172 ] | y [ 173 ] | y [ 174 ] |
y [ 175 ];
assign p[ 44 ]=y[ 176 ] | y [ 177 ] | y [ 178 ] |
y [ 179 ];
assign p[ 45 ]=y[ 180 ] | y [ 181 ] | y [ 182 ] |
y [ 183 ];
assign p[ 46 ]=y[ 184 ] | y [ 185 ] | y [ 186 ] |
y [ 187 ];
assign p[ 47 ]=y[ 188 ] | y [ 189 ] | y [ 190 ] |
y [ 191 ];
assign p[ 48 ]=y[ 192 ] | y [ 193 ] | y [ 194 ] |
y [ 195 ];
assign p[ 49 ]=y[ 196 ] | y [ 197 ] | y [ 198 ] |
y [ 199 ];
assign p[ 50 ]=y[ 200 ] | y [ 201 ] | y [ 202 ] |
y [ 203 ];
assign p[ 51 ]=y[ 204 ] | y [ 205 ] | y [ 206 ] |
y [ 207 ];
assign p[ 52 ]=y[ 208 ] | y [ 209 ] | y [ 210 ] |
y [ 211 ];
```

```
assign p[ 53 ]=y[ 212 ] | y [ 213 ] | y [ 214 ] | y [ 215 ];
assign p[ 54 ]=y[ 216 ] | y [ 217 ] | y [ 218 ] | y [ 219 ];
assign p[ 55 ]=y[ 220 ] | y [ 221 ] | y [ 222 ] | y [ 223 ];
assign p[ 56 ]=y[ 224 ] | y [ 225 ] | y [ 226 ] | y [ 227 ];
assign p[ 57 ]=y[ 228 ] | y [ 229 ] | y [ 230 ] | y [ 231 ];
assign p[ 58 ]=y[ 232 ] | y [ 233 ] | y [ 234 ] | y [ 235 ];
assign p[ 59 ]=y[ 236 ] | y [ 237 ] | y [ 238 ] | y [ 239 ];
assign p[ 60 ]=y[ 240 ] | y [ 241 ] | y [ 242 ] | y [ 243 ];
assign p[ 61 ]=y[ 244 ] | y [ 245 ] | y [ 246 ] | y [ 247 ];
assign p[ 62 ]=y[ 248 ] | y [ 249 ] | y [ 250 ] | y [ 251 ];
assign p[ 63 ]=y[ 252 ] | y [ 253 ] | y [ 254 ] | y [ 255 ];


//third layer
PriorityEncoder_4Bit b1(p[ 3 : 0 ],k[ 3 : 0 ],v65);
PriorityEncoder_4Bit b2(p[ 7 : 4 ],k[ 7 : 4 ],v66);
PriorityEncoder_4Bit b3(p[ 11 : 8 ],k[ 11 : 8 ],v67);
```

```
PriorityEncoder_4Bit b4(p[ 15 : 12 ],k[ 15 : 12 ],v68);
PriorityEncoder_4Bit b5(p[ 19 : 16 ],k[ 19 : 16 ],v69);
PriorityEncoder_4Bit b6(p[ 23 : 20 ],k[ 23 : 20 ],v70);
PriorityEncoder_4Bit b7(p[ 27 : 24 ],k[ 27 : 24 ],v71);
PriorityEncoder_4Bit b8(p[ 31 : 28 ],k[ 31 : 28 ],v72);
PriorityEncoder_4Bit b9(p[ 35 : 32 ],k[ 35 : 32 ],v78);
PriorityEncoder_4Bit b10(p[ 39 : 36 ],k[ 39 : 36 ],v73);
PriorityEncoder_4Bit b11(p[ 43 : 40 ],k[ 43 : 40 ],v74);
PriorityEncoder_4Bit b12(p[ 47 : 44 ],k[ 47 : 44 ],v75);
PriorityEncoder_4Bit b13(p[ 51 : 48 ],k[ 51 : 48 ],v76);
PriorityEncoder_4Bit b14(p[ 55 : 52 ],k[ 55 : 52 ],v77);
PriorityEncoder_4Bit b15(p[ 59 : 56 ],k[ 59 : 56 ],v79);
PriorityEncoder_4Bit b16(p[ 63 : 60 ],k[ 63 : 60 ],v80);

coversion_element u21(y[ 3 : 0 ],k[ 0 ],z[ 3 : 0 ]);
coversion_element u22(y[ 7 : 4 ],k[ 1 ],z[ 7 : 4 ]);
```

```
coversion_element u23(y[ 11 : 8 ],k[ 2 ],z[ 11 : 8 ]);
coversion_element u24(y[ 15 : 12 ],k[ 3 ],z[ 15 : 12 ]);
coversion_element u25(y[ 19 : 16 ],k[ 4 ],z[ 19 : 16 ]);
coversion_element u26(y[ 23 : 20 ],k[ 5 ],z[ 23 : 20 ]);
coversion_element u27(y[ 27 : 24 ],k[ 6 ],z[ 27 : 24 ]);
coversion_element u28(y[ 31 : 28 ],k[ 7 ],z[ 31 : 28 ]);
coversion_element u29(y[ 35 : 32 ],k[ 8 ],z[ 35 : 32 ]);
coversion_element u30(y[ 39 : 36 ],k[ 9 ],z[ 39 : 36 ]);
coversion_element u31(y[ 43 : 40 ],k[ 10 ],z[ 43 : 40 ]);
coversion_element u32(y[ 47 : 44 ],k[ 11 ],z[ 47 : 44 ]);
coversion_element u33(y[ 51 : 48 ],k[ 12 ],z[ 51 : 48 ]);
coversion_element u34(y[ 55 : 52 ],k[ 13 ],z[ 55 : 52 ]);
coversion_element u35(y[ 59 : 56 ],k[ 14 ],z[ 59 : 56 ]);
coversion_element u36(y[ 63 : 60 ],k[ 15 ],z[ 63 : 60 ]);
coversion_element u37(y[ 67 : 64 ],k[ 16 ],z[ 67 : 64 ]);
coversion_element u38(y[ 71 : 68 ],k[ 17 ],z[ 71 : 68 ]);
```

```
coversion_element u39(y[ 75 : 72 ],k[ 18 ],z[ 75 : 72 ]);
coversion_element u40(y[ 79 : 76 ],k[ 19 ],z[ 79 : 76 ]);
coversion_element u41(y[ 83 : 80 ],k[ 20 ],z[ 83 : 80 ]);
coversion_element u42(y[ 87 : 84 ],k[ 21 ],z[ 87 : 84 ]);
coversion_element u43(y[ 91 : 88 ],k[ 22 ],z[ 91 : 88 ]);
coversion_element u44(y[ 95 : 92 ],k[ 23 ],z[ 95 : 92 ]);
coversion_element u45(y[ 99 : 96 ],k[ 24 ],z[ 99 : 96 ]);
coversion_element u46(y[ 103 : 100 ],k[ 25 ],z[ 103 : 100 ]);
coversion_element u47(y[ 107 : 104 ],k[ 26 ],z[ 107 : 104 ]);
coversion_element u48(y[ 111 : 108 ],k[ 27 ],z[ 111 : 108 ]);
coversion_element u49(y[ 115 : 112 ],k[ 28 ],z[ 115 : 112 ]);
coversion_element u50(y[ 119 : 116 ],k[ 29 ],z[ 119 : 116 ]);
coversion_element u51(y[ 123 : 120 ],k[ 30 ],z[ 123 : 120 ]);
coversion_element u52(y[ 127 : 124 ],k[ 31 ],z[ 127 : 124 ]);
coversion_element u53(y[ 131 : 128 ],k[ 32 ],z[ 131 : 128 ]);
coversion_element u54(y[ 135 : 132 ],k[ 33 ],z[ 135 : 132 ]);
```

```
coversion_element u55(y[ 139 : 136 ],k[ 34 ],z[ 139 : 136 ]);
coversion_element u56(y[ 143 : 140 ],k[ 35 ],z[ 143 : 140 ]);
coversion_element u57(y[ 147 : 144 ],k[ 36 ],z[ 147 : 144 ]);
coversion_element u58(y[ 151 : 148 ],k[ 37 ],z[ 151 : 148 ]);
coversion_element u60(y[ 155 : 152 ],k[ 38 ],z[ 155 : 152 ]);
coversion_element u59(y[ 159 : 156 ],k[ 39 ],z[ 159 : 156 ]);
coversion_element u84(y[ 163 : 160 ],k[ 40 ],z[ 163 : 160 ]);
coversion_element u61(y[ 167 : 164 ],k[ 41 ],z[ 167 : 164 ]);
coversion_element u62(y[ 171 : 168 ],k[ 42 ],z[ 171 : 168 ]);
coversion_element u63(y[ 175 : 172 ],k[ 43 ],z[ 175 : 172 ]);
coversion_element u64(y[ 179 : 176 ],k[ 44 ],z[ 179 : 176 ]);
coversion_element u65(y[ 183 : 180 ],k[ 45 ],z[ 183 : 180 ]);
coversion_element u66(y[ 187 : 184 ],k[ 46 ],z[ 187 : 184 ]);
coversion_element u67(y[ 191 : 188 ],k[ 47 ],z[ 191 : 188 ]);
coversion_element u68(y[ 195 : 192 ],k[ 48 ],z[ 195 : 192 ]);
coversion_element u69(y[ 199 : 196 ],k[ 49 ],z[ 199 : 196 ]);
```

```
coversion_element u70(y[ 203 : 200 ],k[ 50 ],z[ 203 : 200 ]);
coversion_element u71(y[ 207 : 204 ],k[ 51 ],z[ 207 : 204 ]);
coversion_element u72(y[ 211 : 208 ],k[ 52 ],z[ 211 : 208 ]);
coversion_element u73(y[ 215 : 212 ],k[ 53 ],z[ 215 : 212 ]);
coversion_element u74(y[ 219 : 216 ],k[ 54 ],z[ 219 : 216 ]);
coversion_element u75(y[ 223 : 220 ],k[ 55 ],z[ 223 : 220 ]);
coversion_element u76(y[ 227 : 224 ],k[ 56 ],z[ 227 : 224 ]);
coversion_element u77(y[ 231 : 228 ],k[ 57 ],z[ 231 : 228 ]);
coversion_element u78(y[ 235 : 232 ],k[ 58 ],z[ 235 : 232 ]);
coversion_element u79(y[ 239 : 236 ],k[ 59 ],z[ 239 : 236 ]);
coversion_element u80(y[ 243 : 240 ],k[ 60 ],z[ 243 : 240 ]);
coversion_element u81(y[ 247 : 244 ],k[ 61 ],z[ 247 : 244 ]);
coversion_element u82(y[ 251 : 248 ],k[ 62 ],z[ 251 : 248 ]);
coversion_element u83(y[ 255 : 252 ],k[ 63 ],z[ 255 : 252 ]);

//fourth layer
```

```
assign q[ 0 ]=k[ 0 ] | k [ 1 ] | k [ 2 ] | k [ 3 ];
assign q[ 1 ]=k[ 4 ] | k [ 5 ] | k [ 6 ] | k [ 7 ];
assign q[ 2 ]=k[ 8 ] | k [ 9 ] | k [ 10 ] | k [ 11 ];
assign q[ 3 ]=k[ 12 ] | k [ 13 ] | k [ 14 ] | k [ 15 ];
assign q[ 4 ]=k[ 16 ] | k [ 17 ] | k [ 18 ] | k [ 19 ];
assign q[ 5 ]=k[ 20 ] | k [ 21 ] | k [ 22 ] | k [ 23 ];
assign q[ 6 ]=k[ 24 ] | k [ 25 ] | k [ 26 ] | k [ 27 ];
assign q[ 7 ]=k[ 28 ] | k [ 29 ] | k [ 30 ] | k [ 31 ];
assign q[ 8 ]=k[ 32 ] | k [ 33 ] | k [ 34 ] | k [ 35 ];
assign q[ 9 ]=k[ 36 ] | k [ 37 ] | k [ 38 ] | k [ 39 ];
assign q[ 10 ]=k[ 40 ] | k [ 41 ] | k [ 42 ] | k [ 43 ];
assign q[ 11 ]=k[ 44 ] | k [ 45 ] | k [ 46 ] | k [ 47 ];
assign q[ 12 ]=k[ 48 ] | k [ 49 ] | k [ 50 ] | k [ 51 ];
assign q[ 13 ]=k[ 52 ] | k [ 53 ] | k [ 54 ] | k [ 55 ];
assign q[ 14 ]=k[ 56 ] | k [ 57 ] | k [ 58 ] | k [ 59 ];
assign q[ 15 ]=k[ 60 ] | k [ 61 ] | k [ 62 ] | k [ 63 ];
```

```
//fifth layer
PriorityEncoder_4Bit c1(q[ 3 : 0 ],j[ 3 : 0 ],v81);
PriorityEncoder_4Bit c2(q[ 7 : 4 ],j[ 7 : 4 ],v82);
PriorityEncoder_4Bit c3(q[ 11 : 8 ],j[ 11 : 8 ],v83);
PriorityEncoder_4Bit c4(q[ 15 : 12 ],j[ 15 : 12 ],v84);

coversion_element u5 (z[3:0],j[0],x[3:0]);
coversion_element u6 (z[7:4],j[0],x[7:4]);
coversion_element u7 (z[11:8],j[0],x[11:8]);
coversion_element u8 (z[15:12],j[0],x[15:12]);
coversion_element u9 (z[19:16],j[1],x[19:16]);
coversion_element u10 (z[23:20],j[1],x[23:20]);
coversion_element u11 (z[27:24],j[1],x[27:24]);
coversion_element u12 (z[31:28],j[1],x[31:28]);
coversion_element u13 (z[35:32],j[2],x[35:32]);
coversion_element u14 (z[39:36],j[2],x[39:36]);
coversion_element u15 (z[43:40],j[2],x[43:40]);
coversion_element u16 (z[47:44],j[2],x[47:44]);
coversion_element u17 (z[51:48],j[3],x[51:48]);
coversion_element u18 (z[55:52],j[3],x[55:52]);
coversion_element u19 (z[59:56],j[3],x[59:56]);
coversion_element u20 (z[63:60],j[3],x[63:60]);
coversion_element u85 (z[67:64],j[4],x[67:64]);
coversion_element u103(z[ 71 : 68 ],j[ 4 ],x[ 71 : 68 ]);
coversion_element u104(z[ 75 : 72 ],j[ 4 ],x[ 75 : 72 ]);
```

```
coversion_element u105(z[ 79 : 76 ],j[ 4 ],x[ 79 : 76 ]);
coversion_element u106(z[ 83 : 80 ],j[ 5 ],x[ 83 : 80 ]);
coversion_element u107(z[ 87 : 84 ],j[ 5 ],x[ 87 : 84 ]);
coversion_element u108(z[ 91 : 88 ],j[ 5 ],x[ 91 : 88 ]);
coversion_element u109(z[ 95 : 92 ],j[ 5 ],x[ 95 : 92 ]);
coversion_element u110(z[ 99 : 96 ],j[ 6 ],x[ 99 : 96 ]);
coversion_element u111(z[ 103 : 100 ],j[ 6 ],x[ 103 : 100 ]);
coversion_element u112(z[ 107 : 104 ],j[ 6 ],x[ 107 : 104 ]);
coversion_element u113(z[ 111 : 108 ],j[ 6 ],x[ 111 : 108 ]);
coversion_element u114(z[ 115 : 112 ],j[ 7 ],x[ 115 : 112 ]);
coversion_element u115(z[ 119 : 116 ],j[ 7 ],x[ 119 : 116 ]);
coversion_element u116(z[ 123 : 120 ],j[ 7 ],x[ 123 : 120 ]);
coversion_element u117(z[ 127 : 124 ],j[ 7 ],x[ 127 : 124 ]);
coversion_element u118(z[ 131 : 128 ],j[ 8 ],x[ 131 : 128 ]);
coversion_element u119(z[ 135 : 132 ],j[ 8 ],x[ 135 : 132 ]);
coversion_element u120(z[ 139 : 136 ],j[ 8 ],x[ 139 : 136 ]);
```

```
coversion_element u121(z[ 143 : 140 ],j[ 8 ],x[ 143 : 140 ]);
coversion_element u122(z[ 147 : 144 ],j[ 9 ],x[ 147 : 144 ]);
coversion_element u123(z[ 151 : 148 ],j[ 9 ],x[ 151 : 148 ]);
coversion_element u124(z[ 155 : 152 ],j[ 9 ],x[ 155 : 152 ]);
coversion_element u125(z[ 159 : 156 ],j[ 9 ],x[ 159 : 156 ]);
coversion_element u126(z[ 163 : 160 ],j[ 10 ],x[ 163 : 160 ]);
coversion_element u127(z[ 167 : 164 ],j[ 10 ],x[ 167 : 164 ]);
coversion_element u128(z[ 171 : 168 ],j[ 10 ],x[ 171 : 168 ]);
coversion_element u129(z[ 175 : 172 ],j[ 10 ],x[ 175 : 172 ]);
coversion_element u130(z[ 179 : 176 ],j[ 11 ],x[ 179 : 176 ]);
coversion_element u131(z[ 183 : 180 ],j[ 11 ],x[ 183 : 180 ]);
coversion_element u132(z[ 187 : 184 ],j[ 11 ],x[ 187 : 184 ]);
coversion_element u133(z[ 191 : 188 ],j[ 11 ],x[ 191 : 188 ]);
coversion_element u134(z[ 195 : 192 ],j[ 12 ],x[ 195 : 192 ]);
coversion_element u135(z[ 199 : 196 ],j[ 12 ],x[ 199 : 196 ]);
coversion_element u136(z[ 203 : 200 ],j[ 12 ],x[ 203 : 200 ]);
```

```
coversion_element u137(z[ 207 : 204 ],j[ 12 ],x[ 207 : 204 ]);
coversion_element u138(z[ 211 : 208 ],j[ 13 ],x[ 211 : 208 ]);
coversion_element u139(z[ 215 : 212 ],j[ 13 ],x[ 215 : 212 ]);
coversion_element u140(z[ 219 : 216 ],j[ 13 ],x[ 219 : 216 ]);
coversion_element u141(z[ 223 : 220 ],j[ 13 ],x[ 223 : 220 ]);
coversion_element u142(z[ 227 : 224 ],j[ 14 ],x[ 227 : 224 ]);
coversion_element u143(z[ 231 : 228 ],j[ 14 ],x[ 231 : 228 ]);
coversion_element u144(z[ 235 : 232 ],j[ 14 ],x[ 235 : 232 ]);
coversion_element u145(z[ 239 : 236 ],j[ 14 ],x[ 239 : 236 ]);
coversion_element u146(z[ 243 : 240 ],j[ 15 ],x[ 243 : 240 ]);
coversion_element u147(z[ 247 : 244 ],j[ 15 ],x[ 247 : 244 ]);
coversion_element u148(z[ 251 : 248 ],j[ 15 ],x[ 251 : 248 ]);
coversion_element u149(z[ 255 : 252 ],j[ 15 ],x[ 255 : 252 ]);

//sixth layer

assign r[ 0 ]=j[ 0 ] | j [1 ] | j [2 ] | j [3 ];
```

```
assign r[ 1 ]=j[ 4 ] | j [5 ] | j [6 ] | j [7 ];
assign r[ 2 ]=j[ 8 ] | j [9 ] | j [10 ] | j [11 ];
assign r[ 3 ]=j[ 12 ] | j [13 ] | j [14 ] | j [15 ];

//seventh layer
PriorityEncoder_4Bit d1(r[3:0],h[3:0],v85);

//eighth layer
//assign t[0] = h[0]|h[1]|h[2]|h[3];

coversion_element j86 (x[ 3 : 0 ],h[ 0
],c[ 3 : 0 ]);
coversion_element j87 (x[ 7 : 4 ],h[ 0
],c[ 7 : 4 ]);
coversion_element j88 (x[ 11 : 8 ],h[ 0
],c[ 11 : 8 ]);
coversion_element j89 (x[ 15 : 12 ],h[ 0
],c[ 15 : 12 ]);
coversion_element j90 (x[ 19 : 16 ],h[ 0
],c[ 19 : 16 ]);
coversion_element j91 (x[ 23 : 20 ],h[ 0
],c[ 23 : 20 ]);
coversion_element j92 (x[ 27 : 24 ],h[ 0
],c[ 27 : 24 ]);
coversion_element j93 (x[ 31 : 28 ],h[ 0
],c[ 31 : 28 ]);
coversion_element j94 (x[ 35 : 32 ],h[ 0
],c[ 35 : 32 ]);
coversion_element j95 (x[ 39 : 36 ],h[ 0
],c[ 39 : 36 ]);
coversion_element j96 (x[ 43 : 40 ],h[ 0
],c[ 43 : 40 ]);
```

```
coversion_element j97 (x[ 47 : 44 ],h[ 0
],c[ 47 : 44 ]);
coversion_element j98 (x[ 51 : 48 ],h[ 0
],c[ 51 : 48 ]);
coversion_element j99 (x[ 55 : 52 ],h[ 0
],c[ 55 : 52 ]);
coversion_element j100 (x[ 59 : 56 ],h[ 0
],c[ 59 : 56 ]);
coversion_element j101 (x[ 63 : 60 ],h[ 0
],c[ 63 : 60 ]);
coversion_element j102 (x[ 67 : 64 ],h[ 1
],c[ 67 : 64 ]);
coversion_element j103 (x[ 71 : 68 ],h[ 1
],c[ 71 : 68 ]);
coversion_element j104 (x[ 75 : 72 ],h[ 1
],c[ 75 : 72 ]);
coversion_element j105 (x[ 79 : 76 ],h[ 1
],c[ 79 : 76 ]);
coversion_element j106 (x[ 83 : 80 ],h[ 1
],c[ 83 : 80 ]);
coversion_element j107 (x[ 87 : 84 ],h[ 1
],c[ 87 : 84 ]);
coversion_element j108 (x[ 91 : 88 ],h[ 1
],c[ 91 : 88 ]);
coversion_element j109 (x[ 95 : 92 ],h[ 1
],c[ 95 : 92 ]);
coversion_element j110 (x[ 99 : 96 ],h[ 1
],c[ 99 : 96 ]);
coversion_element j111 (x[ 103 : 100 ],h[ 1
],c[ 103 : 100 ]);
coversion_element j112 (x[ 107 : 104 ],h[ 1
],c[ 107 : 104 ]);
```

```
coversion_element j113 (x[ 111 : 108 ],h[ 1
],c[ 111 : 108 ]);
coversion_element j114 (x[ 115 : 112 ],h[ 1
],c[ 115 : 112 ]);
coversion_element j115 (x[ 119 : 116 ],h[ 1
],c[ 119 : 116 ]);
coversion_element j116 (x[ 123 : 120 ],h[ 1
],c[ 123 : 120 ]);
coversion_element j117 (x[ 127 : 124 ],h[ 1
],c[ 127 : 124 ]);
coversion_element j118 (x[ 131 : 128 ],h[ 2
],c[ 131 : 128 ]);
coversion_element j119 (x[ 135 : 132 ],h[ 2
],c[ 135 : 132 ]);
coversion_element j120 (x[ 139 : 136 ],h[ 2
],c[ 139 : 136 ]);
coversion_element j121 (x[ 143 : 140 ],h[ 2
],c[ 143 : 140 ]);
coversion_element j122 (x[ 147 : 144 ],h[ 2
],c[ 147 : 144 ]);
coversion_element j123 (x[ 151 : 148 ],h[ 2
],c[ 151 : 148 ]);
coversion_element j124 (x[ 155 : 152 ],h[ 2
],c[ 155 : 152 ]);
coversion_element j125 (x[ 159 : 156 ],h[ 2
],c[ 159 : 156 ]);
coversion_element j126 (x[ 163 : 160 ],h[ 2
],c[ 163 : 160 ]);
coversion_element j127 (x[ 167 : 164 ],h[ 2
],c[ 167 : 164 ]);
coversion_element j128 (x[ 171 : 168 ],h[ 2
],c[ 171 : 168 ]);
```

```
coversion_element j129 (x[ 175 : 172 ],h[ 2
],c[ 175 : 172 ]);
coversion_element j130 (x[ 179 : 176 ],h[ 2
],c[ 179 : 176 ]);
coversion_element j131 (x[ 183 : 180 ],h[ 2
],c[ 183 : 180 ]);
coversion_element j132 (x[ 187 : 184 ],h[ 2
],c[ 187 : 184 ]);
coversion_element j133 (x[ 191 : 188 ],h[ 2
],c[ 191 : 188 ]);
coversion_element j134 (x[ 195 : 192 ],h[ 3
],c[ 195 : 192 ]);
coversion_element j135 (x[ 199 : 196 ],h[ 3
],c[ 199 : 196 ]);
coversion_element j136 (x[ 203 : 200 ],h[ 3
],c[ 203 : 200 ]);
coversion_element j137 (x[ 207 : 204 ],h[ 3
],c[ 207 : 204 ]);
coversion_element j138 (x[ 211 : 208 ],h[ 3
],c[ 211 : 208 ]);
coversion_element j139 (x[ 215 : 212 ],h[ 3
],c[ 215 : 212 ]);
coversion_element j140 (x[ 219 : 216 ],h[ 3
],c[ 219 : 216 ]);
coversion_element j141 (x[ 223 : 220 ],h[ 3
],c[ 223 : 220 ]);
coversion_element j142 (x[ 227 : 224 ],h[ 3
],c[ 227 : 224 ]);
coversion_element j143 (x[ 231 : 228 ],h[ 3
],c[ 231 : 228 ]);
coversion_element j144 (x[ 235 : 232 ],h[ 3
],c[ 235 : 232 ]);
```

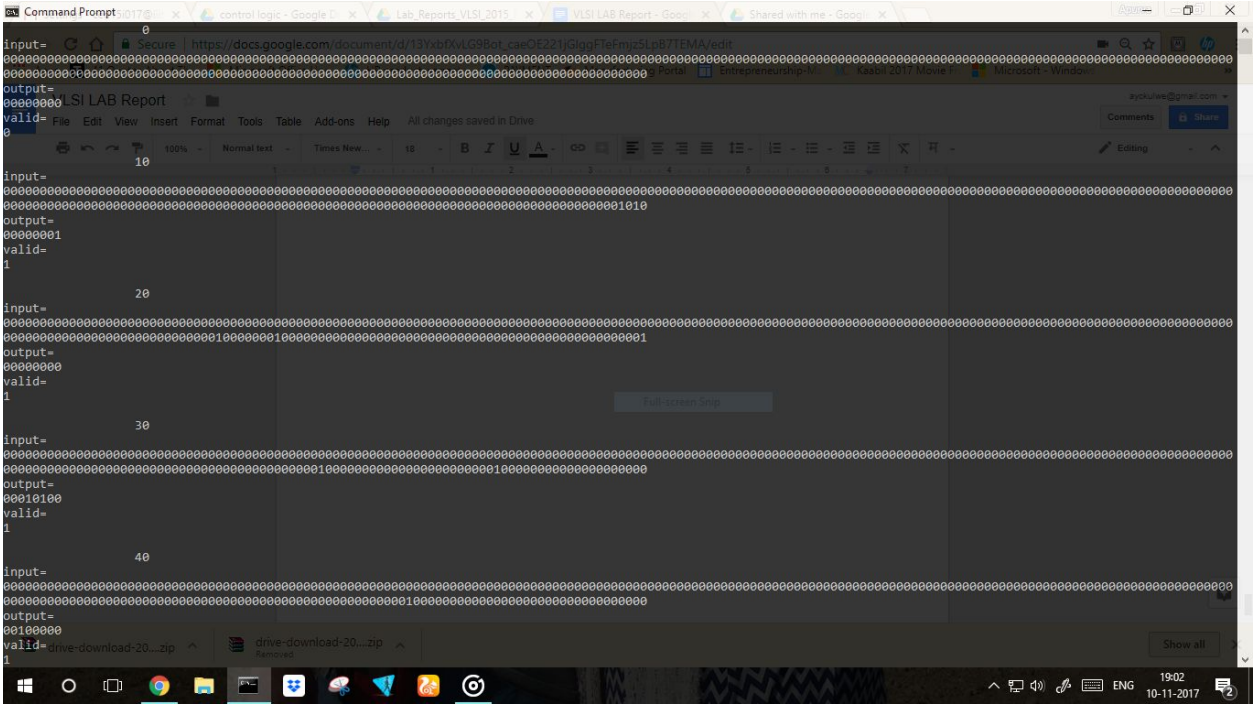
```
        coversion_element j145 (x[   239  :      236  ],h[  3
],c[  239  :      236  ]);
        coversion_element j146 (x[   243  :      240  ],h[  3
],c[  243  :      240  ]);
        coversion_element j147 (x[   247  :      244  ],h[  3
],c[  247  :      244  ]);
        coversion_element j148 (x[   251  :      248  ],h[  3
],c[  251  :      248  ]);
        coversion_element j149 (x[   255  :      252  ],h[  3
],c[  255  :      252  ]);

endmodule

module pe_256bit_Test();
reg [255:0] D;
wire [255:0] Y;
wire V;

pe_256bit i(D,Y,V);

initial
begin
// Initialize Inputs
D = 0;

#100;
#2 D = 256'b00001;
#2 D = 256'b00010;
#2 D = 256'b00011;
#2 D = 256'b00100;
#2 D = 256'b10101;
end
initial
```

```
begin
$monitor("time=",$time,, "D=%b : Y=%b V=%b",D,Y,V);
end
endmodule
```

Conversion_element.v:

```
module coversion_element(a, b, res);
input [3:0]a;
input b;
output [3:0]res;
	wire [3:0]res;

	assign res[0] = a[0] & b;
	assign res[1] = a[1] & b;
	assign res[2] = a[2] & b;
	assign res[3] = a[3] & b;

endmodule
```

<u>Output:</u>



```
                   0
input=
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
output=
00000000
valid=
0
                  10
input=
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000001010
output=
00000001
valid=
1

                  20
input=
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000010000000100000000000000000000000000000000000000000000000001
output=
00000000
valid=
1

                  30
input=
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000100000000000000000000000100000000000000000000
output=
00010100
valid=
1

                  40
input=
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000100000000000000000000000000000000
output=
00100000
valid=
1
```

Experiment no. 8:

Aim: To write a verilog code for TCAM using SRAM.

Code:

Priority256.v:

```
module pe_256(D, result,l4);
        input[255:0]D;
        wire[3:0]w[84:0];// stage outputs
        wire [63:0]o;//or output 1st stage
        wire [16:0]o1;
        wire [3:0]o2;
```

```
wire [63:0]l1;
wire [15:0]l2;
wire [3:0]l3;
output l4;
wire [3:0]cout[127:0];
wire [3:0]fin[63:0];
wire [255:0]outfi;
real result1;
integer n,p;
output [255:0]result;

//stage1
priority4bit a(D[3:0],w[0],l1[0]);
priority4bit a1(D[7:4],w[1],l1[1]);
priority4bit  a2(D[11:8] ,w[2],l1[2]);
priority4bit  a3(D[15:12],w[3],l1[3]);
priority4bit  a4(D[19:16],w[4],l1[4]);
priority4bit  a5(D[23:20],w[5],l1[5]);
priority4bit  a6(D[27:24],w[6],l1[6]);
priority4bit  a7(D[31:28],w[7],l1[7]);
priority4bit  a8(D[35:32],w[8],l1[8]);
priority4bit  a9(D[39:36],w[9],l1[9]);
priority4bit  a10(D[43:40],w[10],l1[10]);
priority4bit  a11(D[47:44],w[11],l1[11]);
priority4bit  a12(D[51:48],w[12],l1[12]);
priority4bit  a13(D[55:52],w[13],l1[13]);
priority4bit  a14(D[59:56],w[14],l1[14]);
priority4bit  a15(D[63:60],w[15],l1[15]);
priority4bit  a16(D[67:64],w[16],l1[16]);
priority4bit  a17(D[71:68],w[17],l1[17]);
priority4bit  a18(D[75:72],w[18],l1[18]);
```

```
priority4bit  a19(D[79:76],w[19],l1[19]);
priority4bit  a20(D[83:80],w[20],l1[20]);
priority4bit  a21(D[87:84],w[21],l1[21]);
priority4bit  a22(D[91:88],w[22],l1[22]);
priority4bit  a23(D[95:92],w[23],l1[23]);
priority4bit  a24(D[99:96],w[24],l1[24]);
priority4bit  a25(D[103:100],w[25],l1[25]);
priority4bit  a26(D[107:104],w[26],l1[26]);
priority4bit  a27(D[111:108],w[27],l1[27]);
priority4bit  a28(D[115:112],w[28],l1[28]);
priority4bit  a29(D[119:116],w[29],l1[29]);
priority4bit  a30(D[123:120],w[30],l1[30]);
priority4bit  a31(D[127:124],w[31],l1[31]);
priority4bit  a32(D[131:128],w[32],l1[32]);
priority4bit  a33(D[135:132],w[33],l1[33]);
priority4bit  a34(D[139:136],w[34],l1[34]);
priority4bit  a35(D[143:140],w[35],l1[35]);
priority4bit  a36(D[147:144],w[36],l1[36]);
priority4bit  a37(D[151:148],w[37],l1[37]);
priority4bit  a38(D[155:152],w[38],l1[38]);
priority4bit  a39(D[159:156],w[39],l1[39]);
priority4bit  a40(D[163:160],w[40],l1[40]);
priority4bit  a41(D[167:164],w[41],l1[41]);
priority4bit  a42(D[171:168],w[42],l1[42]);
priority4bit  a43(D[175:172],w[43],l1[43]);
priority4bit  a44(D[179:176],w[44],l1[44]);
priority4bit  a45(D[183:180],w[45],l1[45]);
priority4bit  a46(D[187:184],w[46],l1[46]);
priority4bit  a47(D[191:188],w[47],l1[47]);
priority4bit  a48(D[195:192],w[48],l1[48]);
priority4bit  a49(D[199:196],w[49],l1[49]);
priority4bit  a50(D[203:200],w[50],l1[50]);
```

```
priority4bit  a51(D[207:204],w[51],l1[51]);
priority4bit  a52(D[211:208],w[52],l1[52]);
priority4bit  a53(D[215:212],w[53],l1[53]);
priority4bit  a54(D[219:216],w[54],l1[54]);
priority4bit  a55(D[223:220],w[55],l1[55]);
priority4bit  a56(D[227:224],w[56],l1[56]);
priority4bit  a57(D[231:228],w[57],l1[57]);
priority4bit  a58(D[235:232],w[58],l1[58]);
priority4bit  a59(D[239:236],w[59],l1[59]);
priority4bit  a60(D[243:240],w[60],l1[60]);
priority4bit  a61(D[247:244],w[61],l1[61]);
priority4bit  a62(D[251:248],w[62],l1[62]);
priority4bit  a63(D[255:252],w[63],l1[63]);

//stage 1 or
assign o[ 0 ]=w[ 0 ][0]|w[ 0 ][1]|w[ 0 ][2]|w[
0 ][3];
assign o[ 1 ]=w[1 ][0]|w[ 1 ][1]|w[ 1
][2]|w[ 1 ][3];
assign o[ 2 ]=w[2 ][0]|w[ 2 ][1]|w[ 2
][2]|w[ 2 ][3];
assign o[ 3 ]=w[3 ][0]|w[ 3 ][1]|w[ 3
][2]|w[ 3 ][3];
assign o[ 4 ]=w[4 ][0]|w[ 4 ][1]|w[ 4
][2]|w[ 4 ][3];
assign o[ 5 ]=w[5 ][0]|w[ 5 ][1]|w[ 5
][2]|w[ 5 ][3];
assign o[ 6 ]=w[6 ][0]|w[ 6 ][1]|w[ 6
][2]|w[ 6 ][3];
assign o[ 7 ]=w[7 ][0]|w[ 7 ][1]|w[ 7
][2]|w[ 7 ][3];
```

```
assign o[ 8 ]=w[8 ][0]|w[ 8 ][1]|w[ 8 ][2]|w[ 8 ][3];
assign o[ 9 ]=w[9 ][0]|w[ 9 ][1]|w[ 9 ][2]|w[ 9 ][3];
assign o[ 10 ]=w[10 ][0]|w[ 10 ][1]|w[ 10 ][2]|w[ 10 ][3];
assign o[ 11 ]=w[11 ][0]|w[ 11 ][1]|w[ 11 ][2]|w[ 11 ][3];
assign o[ 12 ]=w[12 ][0]|w[ 12 ][1]|w[ 12 ][2]|w[ 12 ][3];
assign o[ 13 ]=w[13 ][0]|w[ 13 ][1]|w[ 13 ][2]|w[ 13 ][3];
assign o[ 14 ]=w[14 ][0]|w[ 14 ][1]|w[ 14 ][2]|w[ 14 ][3];
assign o[ 15 ]=w[15 ][0]|w[ 15 ][1]|w[ 15 ][2]|w[ 15 ][3];
assign o[ 16 ]=w[16 ][0]|w[ 16 ][1]|w[ 16 ][2]|w[ 16 ][3];
assign o[ 17 ]=w[17 ][0]|w[ 17 ][1]|w[ 17 ][2]|w[ 17 ][3];
assign o[ 18 ]=w[18 ][0]|w[ 18 ][1]|w[ 18 ][2]|w[ 18 ][3];
assign o[ 19 ]=w[19 ][0]|w[ 19 ][1]|w[ 19 ][2]|w[ 19 ][3];
assign o[ 20 ]=w[20 ][0]|w[ 20 ][1]|w[ 20 ][2]|w[ 20 ][3];
assign o[ 21 ]=w[21 ][0]|w[ 21 ][1]|w[ 21 ][2]|w[ 21 ][3];
assign o[ 22 ]=w[22 ][0]|w[ 22 ][1]|w[ 22 ][2]|w[ 22 ][3];
assign o[ 23 ]=w[23 ][0]|w[ 23 ][1]|w[ 23 ][2]|w[ 23 ][3];
```

```
assign o[24]=w[24][0]|w[24][1]|w[24][2]|w[24][3];
assign o[25]=w[25][0]|w[25][1]|w[25][2]|w[25][3];
assign o[26]=w[26][0]|w[26][1]|w[26][2]|w[26][3];
assign o[27]=w[27][0]|w[27][1]|w[27][2]|w[27][3];
assign o[28]=w[28][0]|w[28][1]|w[28][2]|w[28][3];
assign o[29]=w[29][0]|w[29][1]|w[29][2]|w[29][3];
assign o[30]=w[30][0]|w[30][1]|w[30][2]|w[30][3];
assign o[31]=w[31][0]|w[31][1]|w[31][2]|w[31][3];
assign o[32]=w[32][0]|w[32][1]|w[32][2]|w[32][3];
assign o[33]=w[33][0]|w[33][1]|w[33][2]|w[33][3];
assign o[34]=w[34][0]|w[34][1]|w[34][2]|w[34][3];
assign o[35]=w[35][0]|w[35][1]|w[35][2]|w[35][3];
assign o[36]=w[36][0]|w[36][1]|w[36][2]|w[36][3];
assign o[37]=w[37][0]|w[37][1]|w[37][2]|w[37][3];
assign o[38]=w[38][0]|w[38][1]|w[38][2]|w[38][3];
assign o[39]=w[39][0]|w[39][1]|w[39][2]|w[39][3];
```

```
assign o[40]=w[40][0]|w[40][1]|w[40][2]|w[40][3];
assign o[41]=w[41][0]|w[41][1]|w[41][2]|w[41][3];
assign o[42]=w[42][0]|w[42][1]|w[42][2]|w[42][3];
assign o[43]=w[43][0]|w[43][1]|w[43][2]|w[43][3];
assign o[44]=w[44][0]|w[44][1]|w[44][2]|w[44][3];
assign o[45]=w[45][0]|w[45][1]|w[45][2]|w[45][3];
assign o[46]=w[46][0]|w[46][1]|w[46][2]|w[46][3];
assign o[47]=w[47][0]|w[47][1]|w[47][2]|w[47][3];
assign o[48]=w[48][0]|w[48][1]|w[48][2]|w[48][3];
assign o[49]=w[49][0]|w[49][1]|w[49][2]|w[49][3];
assign o[50]=w[50][0]|w[50][1]|w[50][2]|w[50][3];
assign o[51]=w[51][0]|w[51][1]|w[51][2]|w[51][3];
assign o[52]=w[52][0]|w[52][1]|w[52][2]|w[52][3];
assign o[53]=w[53][0]|w[53][1]|w[53][2]|w[53][3];
assign o[54]=w[54][0]|w[54][1]|w[54][2]|w[54][3];
assign o[55]=w[55][0]|w[55][1]|w[55][2]|w[55][3];
```

```
assign o[ 56 ]=w[56 ][0]|w[ 56 ][1]|w[ 56 ][2]|w[ 56 ][3];
assign o[ 57 ]=w[57 ][0]|w[ 57 ][1]|w[ 57 ][2]|w[ 57 ][3];
assign o[ 58 ]=w[58 ][0]|w[ 58 ][1]|w[ 58 ][2]|w[ 58 ][3];
assign o[ 59 ]=w[59 ][0]|w[ 59 ][1]|w[ 59 ][2]|w[ 59 ][3];
assign o[ 60 ]=w[60 ][0]|w[ 60 ][1]|w[ 60 ][2]|w[ 60 ][3];
assign o[ 61 ]=w[61 ][0]|w[ 61 ][1]|w[ 61 ][2]|w[ 61 ][3];
assign o[ 62 ]=w[62 ][0]|w[ 62 ][1]|w[ 62 ][2]|w[ 62 ][3];
assign o[ 63 ]=w[63 ][0]|w[ 63 ][1]|w[ 63 ][2]|w[ 63 ][3];

//stage 2
priority4bit b(o[3:0],w[64],l2[0]);
priority4bit b1(o[7:4],w[65],l2[1]);
priority4bit b2(o[11:8],w[66],l2[2]);
priority4bit b3(o[15:12],w[67],l2[3]);
priority4bit b4(o[19:16],w[68],l2[4]);
priority4bit b5(o[23:20],w[69],l2[5]);
priority4bit b6(o[27:24],w[70],l2[6]);
priority4bit b7(o[31:28],w[71],l2[7]);
priority4bit b8(o[35:32],w[72],l2[8]);
priority4bit b9(o[39:36],w[73],l2[9]);
priority4bit b10(o[43:40],w[74],l2[10]);
priority4bit b11(o[47:44],w[75],l2[11]);
priority4bit b12(o[51:48],w[76],l2[12]);
priority4bit b13(o[55:52],w[77],l2[13]);
```

```
priority4bit     b14(o[59:56],w[78],l2[14]);
priority4bit     b15(o[63:60],w[79],l2[15]);

//ce
ce   e  (w[0    ],w[64  ][0 ],cout[0    ]);
ce   e1     (w[1 ],w[64     ][1   ],cout[1   ]);
ce   e2     (w[2 ],w[64     ][2   ],cout[2   ]);
ce   e3     (w[3 ],w[64     ][3   ],cout[3   ]);
ce   e4     (w[4 ],w[65     ][0   ],cout[4   ]);
ce   e5     (w[5 ],w[65     ][1   ],cout[5   ]);
ce   e6     (w[6 ],w[65     ][2   ],cout[6   ]);
ce   e7     (w[7 ],w[65     ][3   ],cout[7   ]);
ce   e8     (w[8 ],w[66     ][0   ],cout[8   ]);
ce   e9     (w[9 ],w[66     ][1   ],cout[9   ]);
ce   e10    (w[10      ],w[66     ][2   ],cout[10  ]);
ce   e11    (w[11      ],w[66     ][3   ],cout[11  ]);
ce   e12    (w[12      ],w[67     ][0   ],cout[12  ]);
ce   e13    (w[13      ],w[67     ][1   ],cout[13  ]);
ce   e14    (w[14      ],w[67     ][2   ],cout[14  ]);
ce   e15    (w[15      ],w[67     ][3   ],cout[15  ]);
ce   e16    (w[16      ],w[68     ][0   ],cout[16  ]);
ce   e17    (w[17      ],w[68     ][1   ],cout[17  ]);
ce   e18    (w[18      ],w[68     ][2   ],cout[18  ]);
ce   e19    (w[19      ],w[68     ][3   ],cout[19  ]);
ce   e20    (w[20      ],w[69     ][0   ],cout[20  ]);
ce   e21    (w[21      ],w[69     ][1   ],cout[21  ]);
ce   e22    (w[22      ],w[69     ][2   ],cout[22  ]);
ce   e23    (w[23      ],w[69     ][3   ],cout[23  ]);
ce   e24    (w[24      ],w[70     ][0   ],cout[24  ]);
ce   e25    (w[25      ],w[70     ][1   ],cout[25  ]);
ce   e26    (w[26      ],w[70     ][2   ],cout[26  ]);
ce   e27    (w[27      ],w[70     ][3   ],cout[27  ]);
```

```
ce e28 (w[28 ],w[71 ][0 ],cout[28 ]);
ce e29 (w[29 ],w[71 ][1 ],cout[29 ]);
ce e30 (w[30 ],w[71 ][2 ],cout[30 ]);
ce e31 (w[31 ],w[71 ][3 ],cout[31 ]);
ce e32 (w[32 ],w[72 ][0 ],cout[32 ]);
ce e33 (w[33 ],w[72 ][1 ],cout[33 ]);
ce e34 (w[34 ],w[72 ][2 ],cout[34 ]);
ce e35 (w[35 ],w[72 ][3 ],cout[35 ]);
ce e36 (w[36 ],w[73 ][0 ],cout[36 ]);
ce e37 (w[37 ],w[73 ][1 ],cout[37 ]);
ce e38 (w[38 ],w[73 ][2 ],cout[38 ]);
ce e39 (w[39 ],w[73 ][3 ],cout[39 ]);
ce e40 (w[40 ],w[74 ][0 ],cout[40 ]);
ce e41 (w[41 ],w[74 ][1 ],cout[41 ]);
ce e42 (w[42 ],w[74 ][2 ],cout[42 ]);
ce e43 (w[43 ],w[74 ][3 ],cout[43 ]);
ce e44 (w[44 ],w[75 ][0 ],cout[44 ]);
ce e45 (w[45 ],w[75 ][1 ],cout[45 ]);
ce e46 (w[46 ],w[75 ][2 ],cout[46 ]);
ce e47 (w[47 ],w[75 ][3 ],cout[47 ]);
ce e48 (w[48 ],w[76 ][0 ],cout[48 ]);
ce e49 (w[49 ],w[76 ][1 ],cout[49 ]);
ce e50 (w[50 ],w[76 ][2 ],cout[50 ]);
ce e51 (w[51 ],w[76 ][3 ],cout[51 ]);
ce e52 (w[52 ],w[77 ][0 ],cout[52 ]);
ce e53 (w[53 ],w[77 ][1 ],cout[53 ]);
ce e54 (w[54 ],w[77 ][2 ],cout[54 ]);
ce e55 (w[55 ],w[77 ][3 ],cout[55 ]);
ce e56 (w[56 ],w[78 ][0 ],cout[56 ]);
ce e57 (w[57 ],w[78 ][1 ],cout[57 ]);
ce e58 (w[58 ],w[78 ][2 ],cout[58 ]);
ce e59 (w[59 ],w[78 ][3 ],cout[59 ]);
```

```
ce e60 (w[60 ],w[79 ][0 ],cout[60 ]);
ce e61 (w[61 ],w[79 ][1 ],cout[61 ]);
ce e62 (w[62 ],w[79 ][2 ],cout[62 ]);
ce e63 (w[63 ],w[79 ][3 ],cout[63 ]);


//or stage2

assign o1[ 0 ]=w[64 ][0]|w[ 64 ][1]|w[
64 ][2]|w[ 64 ][3];
assign o1[ 1 ]=w[65 ][0]|w[ 65 ][1]|w[
65 ][2]|w[ 65 ][3];
assign o1[ 2 ]=w[66 ][0]|w[ 66 ][1]|w[
66 ][2]|w[ 66 ][3];
assign o1[ 3 ]=w[67 ][0]|w[ 67 ][1]|w[
67 ][2]|w[ 67 ][3];
assign o1[ 4 ]=w[68 ][0]|w[ 68 ][1]|w[
68 ][2]|w[ 68 ][3];
assign o1[ 5 ]=w[69 ][0]|w[ 69 ][1]|w[
69 ][2]|w[ 69 ][3];
assign o1[ 6 ]=w[70 ][0]|w[ 70 ][1]|w[
70 ][2]|w[ 70 ][3];
assign o1[ 7 ]=w[71 ][0]|w[ 71 ][1]|w[
71 ][2]|w[ 71 ][3];
assign o1[ 8 ]=w[72 ][0]|w[ 72 ][1]|w[
72 ][2]|w[ 72 ][3];
assign o1[ 9 ]=w[73 ][0]|w[ 73 ][1]|w[
73 ][2]|w[ 73 ][3];
assign o1[ 10 ]=w[74 ][0]|w[ 74 ][1]|w[
74 ][2]|w[ 74 ][3];
assign o1[ 11 ]=w[75 ][0]|w[ 75 ][1]|w[
75 ][2]|w[ 75 ][3];
```

```
assign o1[ 12 ]=w[76 ][0]|w[ 76 ][1]|w[
76 ][2]|w[ 76 ][3];
assign o1[ 13 ]=w[77 ][0]|w[ 77 ][1]|w[
77 ][2]|w[ 77 ][3];
assign o1[ 14 ]=w[78 ][0]|w[ 78 ][1]|w[
78 ][2]|w[ 78 ][3];
assign o1[ 15 ]=w[ 79 ][0]|w[ 79 ][1]|w[ 79
][2]|w[ 79 ][3];

//stage3
priority4bit c(o1[3:0],w[80],l3[0]);
priority4bit c1(o1[7:4],w[81],l3[1]);
priority4bit c2(o1[11:8],w[82],l3[2]);
priority4bit c3(o1[15:12],w[83],l3[3]);
//ce
ce f (cout[0 ],w[80 ][0 ],cout[64]);
ce f1 (cout[1 ],w[80 ][0 ],cout[65 ]);
ce f2 (cout[2 ],w[80 ][0 ],cout[66 ]);
ce f3 (cout[3 ],w[80 ][0 ],cout[67 ]);
ce f4 (cout[4 ],w[80 ][1 ],cout[68 ]);
ce f5 (cout[5 ],w[80 ][1 ],cout[69 ]);
ce f6 (cout[6 ],w[80 ][1 ],cout[70 ]);
ce f7 (cout[7 ],w[80 ][1 ],cout[71 ]);
ce f8 (cout[8 ],w[80 ][2 ],cout[72 ]);
ce f9 (cout[9 ],w[80 ][2 ],cout[73 ]);
ce f10(cout[10],w[80 ][2 ],cout[74 ]);
ce f11(cout[11],w[80 ][2 ],cout[75 ]);
ce f12(cout[12],w[80 ][3 ],cout[76 ]);
ce f13(cout[13],w[80 ][3 ],cout[77 ]);
ce f14(cout[14],w[80 ][3 ],cout[78 ]);
ce f15(cout[15],w[80 ][3 ],cout[79 ]);
ce f16(cout[16],w[81 ][0 ],cout[80 ]);
```

```
ce   f17(cout[17],w[81    ][0   ],cout[81  ]);
ce   f18(cout[18],w[81    ][0   ],cout[82  ]);
ce   f19(cout[19],w[81    ][0   ],cout[83  ]);
ce   f20(cout[20],w[81    ][1   ],cout[84  ]);
ce   f21(cout[21],w[81    ][1   ],cout[85  ]);
ce   f22(cout[22],w[81    ][1   ],cout[86  ]);
ce   f23(cout[23],w[81    ][1   ],cout[87  ]);
ce   f24(cout[24],w[81    ][2   ],cout[88  ]);
ce   f25(cout[25],w[81    ][2   ],cout[89  ]);
ce   f26(cout[26],w[81    ][2   ],cout[90  ]);
ce   f27(cout[27],w[81    ][2   ],cout[91  ]);
ce   f28(cout[28],w[81    ][3   ],cout[92  ]);
ce   f29(cout[29],w[81    ][3   ],cout[93  ]);
ce   f30(cout[30],w[81    ][3   ],cout[94  ]);
ce   f31(cout[31],w[81    ][3   ],cout[95  ]);
ce   f32(cout[32],w[82    ][0   ],cout[96  ]);
ce   f33(cout[33],w[82    ][0   ],cout[97  ]);
ce   f34(cout[34],w[82    ][0   ],cout[98  ]);
ce   f35(cout[35],w[82    ][0   ],cout[99  ]);
ce   f36(cout[36],w[82    ][1   ],cout[100     ]);
ce   f37(cout[37],w[82    ][1   ],cout[101     ]);
ce   f38(cout[38],w[82    ][1   ],cout[102     ]);
ce   f39(cout[39],w[82    ][1   ],cout[103     ]);
ce   f40(cout[40],w[82    ][2   ],cout[104     ]);
ce   f41(cout[41],w[82    ][2   ],cout[105     ]);
ce   f42(cout[42],w[82    ][2   ],cout[106     ]);
ce   f43(cout[43],w[82    ][2   ],cout[107     ]);
ce   f44(cout[44],w[82    ][3   ],cout[108     ]);
ce   f45(cout[45],w[82    ][3   ],cout[109     ]);
ce   f46(cout[46],w[82    ][3   ],cout[110     ]);
ce   f47(cout[47],w[82    ][3   ],cout[111     ]);
ce   f48(cout[48],w[83    ][0   ],cout[112     ]);
```

```
ce   f49(cout[49],w[83    ][0   ],cout[113       ]);
ce   f50(cout[50],w[83    ][0   ],cout[114       ]);
ce   f51(cout[51],w[83    ][0   ],cout[115       ]);
ce   f52(cout[52],w[83    ][1   ],cout[116       ]);
ce   f53(cout[53],w[83    ][1   ],cout[117       ]);
ce   f54(cout[54],w[83    ][1   ],cout[118       ]);
ce   f55(cout[55],w[83    ][1   ],cout[119       ]);
ce   f56(cout[56],w[83    ][2   ],cout[120       ]);
ce   f57(cout[57],w[83    ][2   ],cout[121       ]);
ce   f58(cout[58],w[83    ][2   ],cout[122       ]);
ce   f59(cout[59],w[83    ][2   ],cout[123       ]);
ce   f60(cout[60],w[83    ][3   ],cout[124       ]);
ce   f61(cout[61],w[83    ][3   ],cout[125       ]);
ce   f62(cout[62],w[83    ][3   ],cout[126       ]);
ce   f63(cout[63],w[83    ][3   ],cout[127       ]);


//or stage3
assign o2[      0      ]=w[80    ][0]|w[     80    ][1]|w[
80    ][2]|w[     80    ][3];
assign o2[      1      ]=w[81    ][0]|w[     81    ][1]|w[
81    ][2]|w[     81    ][3];
assign o2[      2      ]=w[82    ][0]|w[     82    ][1]|w[
82    ][2]|w[     82    ][3];
assign o2[      3      ]=w[83    ][0]|w[     83    ][1]|w[
83    ][2]|w[     83    ][3];
//stage 4
priority4bit    d(o2[3:0],w[84],l4);
//

ce   g       (cout[64  ],w[84     ][0   ],fin[0      ]);
```

```
ce g1  (cout[65 ],w[84 ][0 ],fin[1 ]);
ce g2  (cout[66 ],w[84 ][0 ],fin[2 ]);
ce g3  (cout[67 ],w[84 ][0 ],fin[3 ]);
ce g4  (cout[68 ],w[84 ][0 ],fin[4 ]);
ce g5  (cout[69 ],w[84 ][0 ],fin[5 ]);
ce g6  (cout[70 ],w[84 ][0 ],fin[6 ]);
ce g7  (cout[71 ],w[84 ][0 ],fin[7 ]);
ce g8  (cout[72 ],w[84 ][0 ],fin[8 ]);
ce g9  (cout[73 ],w[84 ][0 ],fin[9 ]);
ce g10(cout[74 ],w[84 ][0 ],fin[10 ]);
ce g11(cout[75 ],w[84 ][0 ],fin[11 ]);
ce g12(cout[76 ],w[84 ][0 ],fin[12 ]);
ce g13(cout[77 ],w[84 ][0 ],fin[13 ]);
ce g14(cout[78 ],w[84 ][0 ],fin[14 ]);
ce g15(cout[79 ],w[84 ][0 ],fin[15 ]);
ce g16(cout[80 ],w[84 ][1 ],fin[16 ]);
ce g17(cout[81 ],w[84 ][1 ],fin[17 ]);
ce g18(cout[82 ],w[84 ][1 ],fin[18 ]);
ce g19(cout[83 ],w[84 ][1 ],fin[19 ]);
ce g20(cout[84 ],w[84 ][1 ],fin[20 ]);
ce g21(cout[85 ],w[84 ][1 ],fin[21 ]);
ce g22(cout[86 ],w[84 ][1 ],fin[22 ]);
ce g23(cout[87 ],w[84 ][1 ],fin[23 ]);
ce g24(cout[88 ],w[84 ][1 ],fin[24 ]);
ce g25(cout[89 ],w[84 ][1 ],fin[25 ]);
ce g26(cout[90 ],w[84 ][1 ],fin[26 ]);
ce g27(cout[91 ],w[84 ][1 ],fin[27 ]);
ce g28(cout[92 ],w[84 ][1 ],fin[28 ]);
ce g29(cout[93 ],w[84 ][1 ],fin[29 ]);
ce g30(cout[94 ],w[84 ][1 ],fin[30 ]);
ce g31(cout[95 ],w[84 ][1 ],fin[31 ]);
ce g32(cout[96 ],w[84 ][2 ],fin[32 ]);
```

```
ce g33(cout[97 ],w[84 ][2 ],fin[33 ]);
ce g34(cout[98 ],w[84 ][2 ],fin[34 ]);
ce g35(cout[99 ],w[84 ][2 ],fin[35 ]);
ce g36(cout[100 ],w[84 ][2 ],fin[36 ]);
ce g37(cout[101 ],w[84 ][2 ],fin[37 ]);
ce g38(cout[102 ],w[84 ][2 ],fin[38 ]);
ce g39(cout[103 ],w[84 ][2 ],fin[39 ]);
ce g40(cout[104 ],w[84 ][2 ],fin[40 ]);
ce g41(cout[105 ],w[84 ][2 ],fin[41 ]);
ce g42(cout[106 ],w[84 ][2 ],fin[42 ]);
ce g43(cout[107 ],w[84 ][2 ],fin[43 ]);
ce g44(cout[108 ],w[84 ][2 ],fin[44 ]);
ce g45(cout[109 ],w[84 ][2 ],fin[45 ]);
ce g46(cout[110 ],w[84 ][2 ],fin[46 ]);
ce g47(cout[111 ],w[84 ][2 ],fin[47 ]);
ce g48(cout[112 ],w[84 ][3 ],fin[48 ]);
ce g49(cout[113 ],w[84 ][3 ],fin[49 ]);
ce g50(cout[114 ],w[84 ][3 ],fin[50 ]);
ce g51(cout[115 ],w[84 ][3 ],fin[51 ]);
ce g52(cout[116 ],w[84 ][3 ],fin[52 ]);
ce g53(cout[117 ],w[84 ][3 ],fin[53 ]);
ce g54(cout[118 ],w[84 ][3 ],fin[54 ]);
ce g55(cout[119 ],w[84 ][3 ],fin[55 ]);
ce g56(cout[120 ],w[84 ][3 ],fin[56 ]);
ce g57(cout[121 ],w[84 ][3 ],fin[57 ]);
ce g58(cout[122 ],w[84 ][3 ],fin[58 ]);
ce g59(cout[123 ],w[84 ][3 ],fin[59 ]);
ce g60(cout[124 ],w[84 ][3 ],fin[60 ]);
ce g61(cout[125 ],w[84 ][3 ],fin[61 ]);
ce g62(cout[126 ],w[84 ][3 ],fin[62 ]);
ce g63(cout[127 ],w[84 ][3 ],fin[63 ]);
```

```
assign outfi[0  ]=fin[0    ][0   ];
assign outfi[1  ]=fin[0    ][1   ];
assign outfi[2  ]=fin[0    ][2   ];
assign outfi[3  ]=fin[0    ][3   ];
assign outfi[4  ]=fin[1    ][0   ];
assign outfi[5  ]=fin[1    ][1   ];
assign outfi[6  ]=fin[1    ][2   ];
assign outfi[7  ]=fin[1    ][3   ];
assign outfi[8  ]=fin[2    ][0   ];
assign outfi[9  ]=fin[2    ][1   ];
assign outfi[10]=fin[2    ][2   ];
assign outfi[11]=fin[2    ][3   ];
assign outfi[12]=fin[3    ][0   ];
assign outfi[13]=fin[3    ][1   ];
assign outfi[14]=fin[3    ][2   ];
assign outfi[15]=fin[3    ][3   ];
assign outfi[16]=fin[4    ][0   ];
assign outfi[17]=fin[4    ][1   ];
assign outfi[18]=fin[4    ][2   ];
assign outfi[19]=fin[4    ][3   ];
assign outfi[20]=fin[5    ][0   ];
assign outfi[21]=fin[5    ][1   ];
assign outfi[22]=fin[5    ][2   ];
assign outfi[23]=fin[5    ][3   ];
assign outfi[24]=fin[6    ][0   ];
assign outfi[25]=fin[6    ][1   ];
assign outfi[26]=fin[6    ][2   ];
assign outfi[27]=fin[6    ][3   ];
assign outfi[28]=fin[7    ][0   ];
assign outfi[29]=fin[7    ][1   ];
assign outfi[30]=fin[7    ][2   ];
```

```
assign outfi[31]=fin[7    ][3   ];
assign outfi[32]=fin[8    ][0   ];
assign outfi[33]=fin[8    ][1   ];
assign outfi[34]=fin[8    ][2   ];
assign outfi[35]=fin[8    ][3   ];
assign outfi[36]=fin[9    ][0   ];
assign outfi[37]=fin[9    ][1   ];
assign outfi[38]=fin[9    ][2   ];
assign outfi[39]=fin[9    ][3   ];
assign outfi[40]=fin[10   ][0   ];
assign outfi[41]=fin[10   ][1   ];
assign outfi[42]=fin[10   ][2   ];
assign outfi[43]=fin[10   ][3   ];
assign outfi[44]=fin[11   ][0   ];
assign outfi[45]=fin[11   ][1   ];
assign outfi[46]=fin[11   ][2   ];
assign outfi[47]=fin[11   ][3   ];
assign outfi[48]=fin[12   ][0   ];
assign outfi[49]=fin[12   ][1   ];
assign outfi[50]=fin[12   ][2   ];
assign outfi[51]=fin[12   ][3   ];
assign outfi[52]=fin[13   ][0   ];
assign outfi[53]=fin[13   ][1   ];
assign outfi[54]=fin[13   ][2   ];
assign outfi[55]=fin[13   ][3   ];
assign outfi[56]=fin[14   ][0   ];
assign outfi[57]=fin[14   ][1   ];
assign outfi[58]=fin[14   ][2   ];
assign outfi[59]=fin[14   ][3   ];
assign outfi[60]=fin[15   ][0   ];
assign outfi[61]=fin[15   ][1   ];
assign outfi[62]=fin[15   ][2   ];
```

```
assign outfi[63]=fin[15  ][3  ];
assign outfi[64]=fin[16  ][0  ];
assign outfi[65]=fin[16  ][1  ];
assign outfi[66]=fin[16  ][2  ];
assign outfi[67]=fin[16  ][3  ];
assign outfi[68]=fin[17  ][0  ];
assign outfi[69]=fin[17  ][1  ];
assign outfi[70]=fin[17  ][2  ];
assign outfi[71]=fin[17  ][3  ];
assign outfi[72]=fin[18  ][0  ];
assign outfi[73]=fin[18  ][1  ];
assign outfi[74]=fin[18  ][2  ];
assign outfi[75]=fin[18  ][3  ];
assign outfi[76]=fin[19  ][0  ];
assign outfi[77]=fin[19  ][1  ];
assign outfi[78]=fin[19  ][2  ];
assign outfi[79]=fin[19  ][3  ];
assign outfi[80]=fin[20  ][0  ];
assign outfi[81]=fin[20  ][1  ];
assign outfi[82]=fin[20  ][2  ];
assign outfi[83]=fin[20  ][3  ];
assign outfi[84]=fin[21  ][0  ];
assign outfi[85]=fin[21  ][1  ];
assign outfi[86]=fin[21  ][2  ];
assign outfi[87]=fin[21  ][3  ];
assign outfi[88]=fin[22  ][0  ];
assign outfi[89]=fin[22  ][1  ];
assign outfi[90]=fin[22  ][2  ];
assign outfi[91]=fin[22  ][3  ];
assign outfi[92]=fin[23  ][0  ];
assign outfi[93]=fin[23  ][1  ];
assign outfi[94]=fin[23  ][2  ];
```

```
assign outfi[95]=fin[23 ][3 ];
assign outfi[96]=fin[24 ][0 ];
assign outfi[97]=fin[24 ][1 ];
assign outfi[98]=fin[24 ][2 ];
assign outfi[99]=fin[24 ][3 ];
assign outfi[100 ]=fin[25 ][0 ];
assign outfi[101 ]=fin[25 ][1 ];
assign outfi[102 ]=fin[25 ][2 ];
assign outfi[103 ]=fin[25 ][3 ];
assign outfi[104 ]=fin[26 ][0 ];
assign outfi[105 ]=fin[26 ][1 ];
assign outfi[106 ]=fin[26 ][2 ];
assign outfi[107 ]=fin[26 ][3 ];
assign outfi[108 ]=fin[27 ][0 ];
assign outfi[109 ]=fin[27 ][1 ];
assign outfi[110 ]=fin[27 ][2 ];
assign outfi[111 ]=fin[27 ][3 ];
assign outfi[112 ]=fin[28 ][0 ];
assign outfi[113 ]=fin[28 ][1 ];
assign outfi[114 ]=fin[28 ][2 ];
assign outfi[115 ]=fin[28 ][3 ];
assign outfi[116 ]=fin[29 ][0 ];
assign outfi[117 ]=fin[29 ][1 ];
assign outfi[118 ]=fin[29 ][2 ];
assign outfi[119 ]=fin[29 ][3 ];
assign outfi[120 ]=fin[30 ][0 ];
assign outfi[121 ]=fin[30 ][1 ];
assign outfi[122 ]=fin[30 ][2 ];
assign outfi[123 ]=fin[30 ][3 ];
assign outfi[124 ]=fin[31 ][0 ];
assign outfi[125 ]=fin[31 ][1 ];
assign outfi[126 ]=fin[31 ][2 ];
```

```
assign outfi[127    ]=fin[31  ][3  ];
assign outfi[128    ]=fin[32  ][0  ];
assign outfi[129    ]=fin[32  ][1  ];
assign outfi[130    ]=fin[32  ][2  ];
assign outfi[131    ]=fin[32  ][3  ];
assign outfi[132    ]=fin[33  ][0  ];
assign outfi[133    ]=fin[33  ][1  ];
assign outfi[134    ]=fin[33  ][2  ];
assign outfi[135    ]=fin[33  ][3  ];
assign outfi[136    ]=fin[34  ][0  ];
assign outfi[137    ]=fin[34  ][1  ];
assign outfi[138    ]=fin[34  ][2  ];
assign outfi[139    ]=fin[34  ][3  ];
assign outfi[140    ]=fin[35  ][0  ];
assign outfi[141    ]=fin[35  ][1  ];
assign outfi[142    ]=fin[35  ][2  ];
assign outfi[143    ]=fin[35  ][3  ];
assign outfi[144    ]=fin[36  ][0  ];
assign outfi[145    ]=fin[36  ][1  ];
assign outfi[146    ]=fin[36  ][2  ];
assign outfi[147    ]=fin[36  ][3  ];
assign outfi[148    ]=fin[37  ][0  ];
assign outfi[149    ]=fin[37  ][1  ];
assign outfi[150    ]=fin[37  ][2  ];
assign outfi[151    ]=fin[37  ][3  ];
assign outfi[152    ]=fin[38  ][0  ];
assign outfi[153    ]=fin[38  ][1  ];
assign outfi[154    ]=fin[38  ][2  ];
assign outfi[155    ]=fin[38  ][3  ];
assign outfi[156    ]=fin[39  ][0  ];
assign outfi[157    ]=fin[39  ][1  ];
assign outfi[158    ]=fin[39  ][2  ];
```

```
assign outfi[159    ]=fin[39  ][3   ];
assign outfi[160    ]=fin[40  ][0   ];
assign outfi[161    ]=fin[40  ][1   ];
assign outfi[162    ]=fin[40  ][2   ];
assign outfi[163    ]=fin[40  ][3   ];
assign outfi[164    ]=fin[41  ][0   ];
assign outfi[165    ]=fin[41  ][1   ];
assign outfi[166    ]=fin[41  ][2   ];
assign outfi[167    ]=fin[41  ][3   ];
assign outfi[168    ]=fin[42  ][0   ];
assign outfi[169    ]=fin[42  ][1   ];
assign outfi[170    ]=fin[42  ][2   ];
assign outfi[171    ]=fin[42  ][3   ];
assign outfi[172    ]=fin[43  ][0   ];
assign outfi[173    ]=fin[43  ][1   ];
assign outfi[174    ]=fin[43  ][2   ];
assign outfi[175    ]=fin[43  ][3   ];
assign outfi[176    ]=fin[44  ][0   ];
assign outfi[177    ]=fin[44  ][1   ];
assign outfi[178    ]=fin[44  ][2   ];
assign outfi[179    ]=fin[44  ][3   ];
assign outfi[180    ]=fin[45  ][0   ];
assign outfi[181    ]=fin[45  ][1   ];
assign outfi[182    ]=fin[45  ][2   ];
assign outfi[183    ]=fin[45  ][3   ];
assign outfi[184    ]=fin[46  ][0   ];
assign outfi[185    ]=fin[46  ][1   ];
assign outfi[186    ]=fin[46  ][2   ];
assign outfi[187    ]=fin[46  ][3   ];
assign outfi[188    ]=fin[47  ][0   ];
assign outfi[189    ]=fin[47  ][1   ];
assign outfi[190    ]=fin[47  ][2   ];
```

```
assign outfi[191    ]=fin[47  ][3  ];
assign outfi[192    ]=fin[48  ][0  ];
assign outfi[193    ]=fin[48  ][1  ];
assign outfi[194    ]=fin[48  ][2  ];
assign outfi[195    ]=fin[48  ][3  ];
assign outfi[196    ]=fin[49  ][0  ];
assign outfi[197    ]=fin[49  ][1  ];
assign outfi[198    ]=fin[49  ][2  ];
assign outfi[199    ]=fin[49  ][3  ];
assign outfi[200    ]=fin[50  ][0  ];
assign outfi[201    ]=fin[50  ][1  ];
assign outfi[202    ]=fin[50  ][2  ];
assign outfi[203    ]=fin[50  ][3  ];
assign outfi[204    ]=fin[51  ][0  ];
assign outfi[205    ]=fin[51  ][1  ];
assign outfi[206    ]=fin[51  ][2  ];
assign outfi[207    ]=fin[51  ][3  ];
assign outfi[208    ]=fin[52  ][0  ];
assign outfi[209    ]=fin[52  ][1  ];
assign outfi[210    ]=fin[52  ][2  ];
assign outfi[211    ]=fin[52  ][3  ];
assign outfi[212    ]=fin[53  ][0  ];
assign outfi[213    ]=fin[53  ][1  ];
assign outfi[214    ]=fin[53  ][2  ];
assign outfi[215    ]=fin[53  ][3  ];
assign outfi[216    ]=fin[54  ][0  ];
assign outfi[217    ]=fin[54  ][1  ];
assign outfi[218    ]=fin[54  ][2  ];
assign outfi[219    ]=fin[54  ][3  ];
assign outfi[220    ]=fin[55  ][0  ];
assign outfi[221    ]=fin[55  ][1  ];
assign outfi[222    ]=fin[55  ][2  ];
```

```
assign outfi[223    ]=fin[55  ][3  ];
assign outfi[224    ]=fin[56  ][0  ];
assign outfi[225    ]=fin[56  ][1  ];
assign outfi[226    ]=fin[56  ][2  ];
assign outfi[227    ]=fin[56  ][3  ];
assign outfi[228    ]=fin[57  ][0  ];
assign outfi[229    ]=fin[57  ][1  ];
assign outfi[230    ]=fin[57  ][2  ];
assign outfi[231    ]=fin[57  ][3  ];
assign outfi[232    ]=fin[58  ][0  ];
assign outfi[233    ]=fin[58  ][1  ];
assign outfi[234    ]=fin[58  ][2  ];
assign outfi[235    ]=fin[58  ][3  ];
assign outfi[236    ]=fin[59  ][0  ];
assign outfi[237    ]=fin[59  ][1  ];
assign outfi[238    ]=fin[59  ][2  ];
assign outfi[239    ]=fin[59  ][3  ];
assign outfi[240    ]=fin[60  ][0  ];
assign outfi[241    ]=fin[60  ][1  ];
assign outfi[242    ]=fin[60  ][2  ];
assign outfi[243    ]=fin[60  ][3  ];
assign outfi[244    ]=fin[61  ][0  ];
assign outfi[245    ]=fin[61  ][1  ];
assign outfi[246    ]=fin[61  ][2  ];
assign outfi[247    ]=fin[61  ][3  ];
assign outfi[248    ]=fin[62  ][0  ];
assign outfi[249    ]=fin[62  ][1  ];
assign outfi[250    ]=fin[62  ][2  ];
assign outfi[251    ]=fin[62  ][3  ];
assign outfi[252    ]=fin[63  ][0  ];
assign outfi[253    ]=fin[63  ][1  ];
assign outfi[254    ]=fin[63  ][2  ];
```

```
        assign outfi[255      ]=fin[63   ][3    ];


endmodule
```

Tcam.v:

```
module tcam(W,N);
input [143:0]W;
output [255:0]N;

        wire [255:0]d1[3:0];
        assign d1[0][255:0] = 256'd120;
        assign d1[1][255:0] = 256'd121;
        assign d1[2][255:0] = 256'd122;
        assign d1[3][255:0] = 256'd123;
        wire [255:0]d2[3:0];
        assign d2[0][255:0] = 256'd120;
        assign d2[1][255:0] = 256'd121;
        assign d2[2][255:0] = 256'd122;
        assign d2[3][255:0] = 256'd123;
        wire [255:0]d3[3:0];
        assign d3[0][255:0] = 256'd120;
        assign d3[1][255:0] = 256'd121;
        assign d3[2][255:0] = 256'd122;
        assign d3[3][255:0] = 256'd123;
        wire [255:0]d4[3:0];
        assign d4[0][255:0] = 256'd120;
        assign d4[1][255:0] = 256'd121;
        assign d4[2][255:0] = 256'd122;
        assign d4[3][255:0] = 256'd123;
        wire [255:0]d5[3:0];
        assign d5[0][255:0] = 256'd120;
```

```
assign d5[1][255:0] = 256'd121;
assign d5[2][255:0] = 256'd122;
assign d5[3][255:0] = 256'd123;
wire [255:0]d6[3:0];
assign d6[0][255:0] = 256'd120;
assign d6[1][255:0] = 256'd121;
assign d6[2][255:0] = 256'd122;
assign d6[3][255:0] = 256'd123;
wire [255:0]d7[3:0];
assign d7[0][255:0] = 256'd120;
assign d7[1][255:0] = 256'd121;
assign d7[2][255:0] = 256'd122;
assign d7[3][255:0] = 256'd123;
wire [255:0]d8[3:0];
assign d8[0][255:0] = 256'd120;
assign d8[1][255:0] = 256'd121;
assign d8[2][255:0] = 256'd122;
assign d8[3][255:0] = 256'd123;
wire [255:0]d9[3:0];
assign d9[0][255:0] = 256'd120;
assign d9[1][255:0] = 256'd121;
assign d9[2][255:0] = 256'd122;
assign d9[3][255:0] = 256'd123;
wire [255:0]d10[3:0];
assign d10[0][255:0] = 256'd120;
assign d10[1][255:0] = 256'd121;
assign d10[2][255:0] = 256'd122;
assign d10[3][255:0] = 256'd123;
wire [255:0]d11[3:0];
assign d11[0][255:0] = 256'd120;
assign d11[1][255:0] = 256'd121;
assign d11[2][255:0] = 256'd122;
```

```
assign d11[3][255:0] = 256'd123;
wire [255:0]d12[3:0];
assign d12[0][255:0] = 256'd120;
assign d12[1][255:0] = 256'd121;
assign d12[2][255:0] = 256'd122;
assign d12[3][255:0] = 256'd123;
wire [255:0]d13[3:0];
assign d13[0][255:0] = 256'd120;
assign d13[1][255:0] = 256'd121;
assign d13[2][255:0] = 256'd122;
assign d13[3][255:0] = 256'd123;
wire [255:0]d14[3:0];
assign d14[0][255:0] = 256'd120;
assign d14[1][255:0] = 256'd121;
assign d14[2][255:0] = 256'd122;
assign d14[3][255:0] = 256'd123;
wire [255:0]d15[3:0];
assign d15[0][255:0] = 256'd120;
assign d15[1][255:0] = 256'd121;
assign d15[2][255:0] = 256'd122;
assign d15[3][255:0] = 256'd123;
wire [255:0]d16[3:0];
assign d16[0][255:0] = 256'd120;
assign d16[1][255:0] = 256'd121;
assign d16[2][255:0] = 256'd122;
assign d16[3][255:0] = 256'd123;
wire [255:0]d17[3:0];
assign d17[0][255:0] = 256'd120;
assign d17[1][255:0] = 256'd121;
assign d17[2][255:0] = 256'd122;
assign d17[3][255:0] = 256'd123;
wire [255:0]d18[3:0];
```

```
assign d18[0][255:0] = 256'd120;
assign d18[1][255:0] = 256'd121;
assign d18[2][255:0] = 256'd122;
assign d18[3][255:0] = 256'd123;
wire [255:0]d19[3:0];
assign d19[0][255:0] = 256'd120;
assign d19[1][255:0] = 256'd121;
assign d19[2][255:0] = 256'd122;
assign d19[3][255:0] = 256'd123;
wire [255:0]d20[3:0];
assign d20[0][255:0] = 256'd120;
assign d20[1][255:0] = 256'd121;
assign d20[2][255:0] = 256'd122;
assign d20[3][255:0] = 256'd123;
wire [255:0]d21[3:0];
assign d21[0][255:0] = 256'd120;
assign d21[1][255:0] = 256'd121;
assign d21[2][255:0] = 256'd122;
assign d21[3][255:0] = 256'd123;
wire [255:0]d22[3:0];
assign d22[0][255:0] = 256'd120;
assign d22[1][255:0] = 256'd121;
assign d22[2][255:0] = 256'd122;
assign d22[3][255:0] = 256'd123;
wire [255:0]d23[3:0];
assign d23[0][255:0] = 256'd120;
assign d23[1][255:0] = 256'd121;
assign d23[2][255:0] = 256'd122;
assign d23[3][255:0] = 256'd123;
wire [255:0]d24[3:0];
assign d24[0][255:0] = 256'd120;
assign d24[1][255:0] = 256'd121;
```

```
assign d24[2][255:0] = 256'd122;
assign d24[3][255:0] = 256'd123;
wire [255:0]d25[3:0];
assign d25[0][255:0] = 256'd120;
assign d25[1][255:0] = 256'd121;
assign d25[2][255:0] = 256'd122;
assign d25[3][255:0] = 256'd123;
wire [255:0]d26[3:0];
assign d26[0][255:0] = 256'd120;
assign d26[1][255:0] = 256'd121;
assign d26[2][255:0] = 256'd122;
assign d26[3][255:0] = 256'd123;
wire [255:0]d27[3:0];
assign d27[0][255:0] = 256'd120;
assign d27[1][255:0] = 256'd121;
assign d27[2][255:0] = 256'd122;
assign d27[3][255:0] = 256'd123;
wire [255:0]d28[3:0];
assign d28[0][255:0] = 256'd120;
assign d28[1][255:0] = 256'd121;
assign d28[2][255:0] = 256'd122;
assign d28[3][255:0] = 256'd123;
wire [255:0]d29[3:0];
assign d29[0][255:0] = 256'd120;
assign d29[1][255:0] = 256'd121;
assign d29[2][255:0] = 256'd122;
assign d29[3][255:0] = 256'd123;
wire [255:0]d30[3:0];
assign d30[0][255:0] = 256'd120;
assign d30[1][255:0] = 256'd121;
assign d30[2][255:0] = 256'd122;
assign d30[3][255:0] = 256'd123;
```

```
wire [255:0]d31[3:0];
assign d31[0][255:0] = 256'd120;
assign d31[1][255:0] = 256'd121;
assign d31[2][255:0] = 256'd122;
assign d31[3][255:0] = 256'd123;
wire [255:0]d32[3:0];
assign d32[0][255:0] = 256'd120;
assign d32[1][255:0] = 256'd121;
assign d32[2][255:0] = 256'd122;
assign d32[3][255:0] = 256'd123;
wire [255:0]d33[3:0];
assign d33[0][255:0] = 256'd120;
assign d33[1][255:0] = 256'd121;
assign d33[2][255:0] = 256'd122;
assign d33[3][255:0] = 256'd123;
wire [255:0]d34[3:0];
assign d34[0][255:0] = 256'd120;
assign d34[1][255:0] = 256'd121;
assign d34[2][255:0] = 256'd122;
assign d34[3][255:0] = 256'd123;
wire [255:0]d35[3:0];
assign d35[0][255:0] = 256'd120;
assign d35[1][255:0] = 256'd121;
assign d35[2][255:0] = 256'd122;
assign d35[3][255:0] = 256'd123;
wire [255:0]d36[3:0];
assign d36[0][255:0] = 256'd120;
assign d36[1][255:0] = 256'd121;
assign d36[2][255:0] = 256'd122;
assign d36[3][255:0] = 256'd123;
wire [255:0]d37[3:0];
assign d37[0][255:0] = 256'd120;
```

```
assign d37[1][255:0] = 256'd121;
assign d37[2][255:0] = 256'd122;
assign d37[3][255:0] = 256'd123;
wire [255:0]d38[3:0];
assign d38[0][255:0] = 256'd120;
assign d38[1][255:0] = 256'd121;
assign d38[2][255:0] = 256'd122;
assign d38[3][255:0] = 256'd123;
wire [255:0]d39[3:0];
assign d39[0][255:0] = 256'd120;
assign d39[1][255:0] = 256'd121;
assign d39[2][255:0] = 256'd122;
assign d39[3][255:0] = 256'd123;
wire [255:0]d40[3:0];
assign d40[0][255:0] = 256'd120;
assign d40[1][255:0] = 256'd121;
assign d40[2][255:0] = 256'd122;
assign d40[3][255:0] = 256'd123;
wire [255:0]d41[3:0];
assign d41[0][255:0] = 256'd120;
assign d41[1][255:0] = 256'd121;
assign d41[2][255:0] = 256'd122;
assign d41[3][255:0] = 256'd123;
wire [255:0]d42[3:0];
assign d42[0][255:0] = 256'd120;
assign d42[1][255:0] = 256'd121;
assign d42[2][255:0] = 256'd122;
assign d42[3][255:0] = 256'd123;
wire [255:0]d43[3:0];
assign d43[0][255:0] = 256'd120;
assign d43[1][255:0] = 256'd121;
assign d43[2][255:0] = 256'd122;
```

```
assign d43[3][255:0] = 256'd123;
wire [255:0]d44[3:0];
assign d44[0][255:0] = 256'd120;
assign d44[1][255:0] = 256'd121;
assign d44[2][255:0] = 256'd122;
assign d44[3][255:0] = 256'd123;
wire [255:0]d45[3:0];
assign d45[0][255:0] = 256'd120;
assign d45[1][255:0] = 256'd121;
assign d45[2][255:0] = 256'd122;
assign d45[3][255:0] = 256'd123;
wire [255:0]d46[3:0];
assign d46[0][255:0] = 256'd120;
assign d46[1][255:0] = 256'd121;
assign d46[2][255:0] = 256'd122;
assign d46[3][255:0] = 256'd123;
wire [255:0]d47[3:0];
assign d47[0][255:0] = 256'd120;
assign d47[1][255:0] = 256'd121;
assign d47[2][255:0] = 256'd122;
assign d47[3][255:0] = 256'd123;
wire [255:0]d48[3:0];
assign d48[0][255:0] = 256'd120;
assign d48[1][255:0] = 256'd121;
assign d48[2][255:0] = 256'd122;
assign d48[3][255:0] = 256'd123;
wire [255:0]d49[3:0];
assign d49[0][255:0] = 256'd120;
assign d49[1][255:0] = 256'd121;
assign d49[2][255:0] = 256'd122;
assign d49[3][255:0] = 256'd123;
wire [255:0]d50[3:0];
```

```
assign d50[0][255:0] = 256'd120;
assign d50[1][255:0] = 256'd121;
assign d50[2][255:0] = 256'd122;
assign d50[3][255:0] = 256'd123;
wire [255:0]d51[3:0];
assign d51[0][255:0] = 256'd120;
assign d51[1][255:0] = 256'd121;
assign d51[2][255:0] = 256'd122;
assign d51[3][255:0] = 256'd123;
wire [255:0]d52[3:0];
assign d52[0][255:0] = 256'd120;
assign d52[1][255:0] = 256'd121;
assign d52[2][255:0] = 256'd122;
assign d52[3][255:0] = 256'd123;
wire [255:0]d53[3:0];
assign d53[0][255:0] = 256'd120;
assign d53[1][255:0] = 256'd121;
assign d53[2][255:0] = 256'd122;
assign d53[3][255:0] = 256'd123;
wire [255:0]d54[3:0];
assign d54[0][255:0] = 256'd120;
assign d54[1][255:0] = 256'd121;
assign d54[2][255:0] = 256'd122;
assign d54[3][255:0] = 256'd123;
wire [255:0]d55[3:0];
assign d55[0][255:0] = 256'd120;
assign d55[1][255:0] = 256'd121;
assign d55[2][255:0] = 256'd122;
assign d55[3][255:0] = 256'd123;
wire [255:0]d56[3:0];
assign d56[0][255:0] = 256'd120;
assign d56[1][255:0] = 256'd121;
```

```
assign d56[2][255:0] = 256'd122;
assign d56[3][255:0] = 256'd123;
wire [255:0]d57[3:0];
assign d57[0][255:0] = 256'd120;
assign d57[1][255:0] = 256'd121;
assign d57[2][255:0] = 256'd122;
assign d57[3][255:0] = 256'd123;
wire [255:0]d58[3:0];
assign d58[0][255:0] = 256'd120;
assign d58[1][255:0] = 256'd121;
assign d58[2][255:0] = 256'd122;
assign d58[3][255:0] = 256'd123;
wire [255:0]d59[3:0];
assign d59[0][255:0] = 256'd120;
assign d59[1][255:0] = 256'd121;
assign d59[2][255:0] = 256'd122;
assign d59[3][255:0] = 256'd123;
wire [255:0]d60[3:0];
assign d60[0][255:0] = 256'd120;
assign d60[1][255:0] = 256'd121;
assign d60[2][255:0] = 256'd122;
assign d60[3][255:0] = 256'd123;
wire [255:0]d61[3:0];
assign d61[0][255:0] = 256'd120;
assign d61[1][255:0] = 256'd121;
assign d61[2][255:0] = 256'd122;
assign d61[3][255:0] = 256'd123;
wire [255:0]d62[3:0];
assign d62[0][255:0] = 256'd120;
assign d62[1][255:0] = 256'd121;
assign d62[2][255:0] = 256'd122;
assign d62[3][255:0] = 256'd123;
```

```
wire [255:0]d63[3:0];
assign d63[0][255:0] = 256'd120;
assign d63[1][255:0] = 256'd121;
assign d63[2][255:0] = 256'd122;
assign d63[3][255:0] = 256'd123;
wire [255:0]d64[3:0];
assign d64[0][255:0] = 256'd120;
assign d64[1][255:0] = 256'd121;
assign d64[2][255:0] = 256'd122;
assign d64[3][255:0] = 256'd123;
wire [255:0]d65[3:0];
assign d65[0][255:0] = 256'd120;
assign d65[1][255:0] = 256'd121;
assign d65[2][255:0] = 256'd122;
assign d65[3][255:0] = 256'd123;
wire [255:0]d66[3:0];
assign d66[0][255:0] = 256'd120;
assign d66[1][255:0] = 256'd121;
assign d66[2][255:0] = 256'd122;
assign d66[3][255:0] = 256'd123;
wire [255:0]d67[3:0];
assign d67[0][255:0] = 256'd120;
assign d67[1][255:0] = 256'd121;
assign d67[2][255:0] = 256'd122;
assign d67[3][255:0] = 256'd123;
wire [255:0]d68[3:0];
assign d68[0][255:0] = 256'd120;
assign d68[1][255:0] = 256'd121;
assign d68[2][255:0] = 256'd122;
assign d68[3][255:0] = 256'd123;
wire [255:0]d69[3:0];
assign d69[0][255:0] = 256'd120;
```

```
assign d69[1][255:0] = 256'd121;
assign d69[2][255:0] = 256'd122;
assign d69[3][255:0] = 256'd123;
wire [255:0]d70[3:0];
assign d70[0][255:0] = 256'd120;
assign d70[1][255:0] = 256'd121;
assign d70[2][255:0] = 256'd122;
assign d70[3][255:0] = 256'd123;
wire [255:0]d71[3:0];
assign d71[0][255:0] = 256'd120;
assign d71[1][255:0] = 256'd121;
assign d71[2][255:0] = 256'd122;
assign d71[3][255:0] = 256'd123;
wire [255:0]d72[3:0];
assign d72[0][255:0] = 256'd120;
assign d72[1][255:0] = 256'd121;
assign d72[2][255:0] = 256'd122;
assign d72[3][255:0] = 256'd123;

wire [255:0]out1;
wire [255:0]out2;
wire [255:0]out3;
wire [255:0]out4;
wire [255:0]out5;
wire [255:0]out6;
wire [255:0]out7;
wire [255:0]out8;
wire [255:0]out9;
wire [255:0]out10;
wire [255:0]out11;
wire [255:0]out12;
wire [255:0]out13;
```

```
wire [255:0]out14;
wire [255:0]out15;
wire [255:0]out16;
wire [255:0]out17;
wire [255:0]out18;
wire [255:0]out19;
wire [255:0]out20;
wire [255:0]out21;
wire [255:0]out22;
wire [255:0]out23;
wire [255:0]out24;
wire [255:0]out25;
wire [255:0]out26;
wire [255:0]out27;
wire [255:0]out28;
wire [255:0]out29;
wire [255:0]out30;
wire [255:0]out31;
wire [255:0]out32;
wire [255:0]out33;
wire [255:0]out34;
wire [255:0]out35;
wire [255:0]out36;
wire [255:0]out37;
wire [255:0]out38;
wire [255:0]out39;
wire [255:0]out40;
wire [255:0]out41;
wire [255:0]out42;
wire [255:0]out43;
wire [255:0]out44;
wire [255:0]out45;
```

```
wire [255:0]out46;
wire [255:0]out47;
wire [255:0]out48;
wire [255:0]out49;
wire [255:0]out50;
wire [255:0]out51;
wire [255:0]out52;
wire [255:0]out53;
wire [255:0]out54;
wire [255:0]out55;
wire [255:0]out56;
wire [255:0]out57;
wire [255:0]out58;
wire [255:0]out59;
wire [255:0]out60;
wire [255:0]out61;
wire [255:0]out62;
wire [255:0]out63;
wire [255:0]out64;
wire [255:0]out65;
wire [255:0]out66;
wire [255:0]out67;
wire [255:0]out68;
wire [255:0]out69;
wire [255:0]out70;
wire [255:0]out71;
wire [255:0]out72;

integer i1;
mux m1(W[1:0],d1[0],d1[1],d1[2],d1[3],out1);
//assign N[255:0] = out1[255:0];
mux m2(W[3:2],d2[0],d2[1],d2[2],d2[3],out2);
```

```
mux m3(W[5:4],d3[0],d3[1],d3[2],d3[3],out3);
mux m4(W[7:6],d4[0],d4[1],d4[2],d4[3],out4);
mux m5(W[9:8],d5[0],d5[1],d5[2],d5[3],out5);
mux m6(W[11:10],d6[0],d6[1],d6[2],d6[3],out6);
mux m7(W[13:12],d7[0],d7[1],d7[2],d7[3],out7);
mux m8(W[15:14],d8[0],d8[1],d8[2],d8[3],out8);
mux m9(W[17:16],d9[0],d9[1],d9[2],d9[3],out9);
mux
m10(W[19:18],d10[0],d10[1],d10[2],d10[3],out10);
mux
m11(W[21:20],d11[0],d11[1],d11[2],d11[3],out11);
mux
m12(W[23:22],d12[0],d12[1],d12[2],d12[3],out12);
mux
m13(W[25:24],d13[0],d13[1],d13[2],d13[3],out13);
mux
m14(W[27:26],d14[0],d14[1],d14[2],d14[3],out14);
mux
m15(W[29:28],d15[0],d15[1],d15[2],d15[3],out15);
mux
m16(W[31:30],d16[0],d16[1],d16[2],d16[3],out16);
mux
m17(W[33:32],d17[0],d17[1],d17[2],d17[3],out17);
mux
m18(W[35:34],d18[0],d18[1],d18[2],d18[3],out18);
mux
m19(W[37:36],d19[0],d19[1],d19[2],d19[3],out19);
mux
m20(W[39:38],d20[0],d20[1],d20[2],d20[3],out20);
mux
m21(W[41:40],d21[0],d21[1],d21[2],d21[3],out21);
```

```
mux
m22(W[43:42],d22[0],d22[1],d22[2],d22[3],out22);
mux
m23(W[45:44],d23[0],d23[1],d23[2],d23[3],out23);
mux
m24(W[47:46],d24[0],d24[1],d24[2],d24[3],out24);
mux
m25(W[49:48],d25[0],d25[1],d25[2],d25[3],out25);
mux
m26(W[51:50],d26[0],d26[1],d26[2],d26[3],out26);
mux
m27(W[53:52],d27[0],d27[1],d27[2],d27[3],out27);
mux
m28(W[55:54],d28[0],d28[1],d28[2],d28[3],out28);
mux
m29(W[57:56],d29[0],d29[1],d29[2],d29[3],out29);
mux
m30(W[59:58],d30[0],d30[1],d30[2],d30[3],out30);
mux
m31(W[61:60],d31[0],d31[1],d31[2],d31[3],out31);
mux
m32(W[63:62],d32[0],d32[1],d32[2],d32[3],out32);
mux
m33(W[65:64],d33[0],d33[1],d33[2],d33[3],out33);
mux
m34(W[67:66],d34[0],d34[1],d34[2],d34[3],out34);
mux
m35(W[69:68],d35[0],d35[1],d35[2],d35[3],out35);
mux
m36(W[71:70],d36[0],d36[1],d36[2],d36[3],out36);
mux
m37(W[73:72],d37[0],d37[1],d37[2],d37[3],out37);
```

```
mux
m38(W[75:74],d38[0],d38[1],d38[2],d38[3],out38);
mux
m39(W[77:76],d39[0],d39[1],d39[2],d39[3],out39);
mux
m40(W[79:78],d40[0],d40[1],d40[2],d40[3],out40);
mux
m41(W[81:80],d41[0],d41[1],d41[2],d41[3],out41);
mux
m42(W[83:82],d42[0],d42[1],d42[2],d42[3],out42);
mux
m43(W[85:84],d43[0],d43[1],d43[2],d43[3],out43);
mux
m44(W[87:86],d44[0],d44[1],d44[2],d44[3],out44);
mux
m45(W[89:88],d45[0],d45[1],d45[2],d45[3],out45);
mux
m46(W[91:90],d46[0],d46[1],d46[2],d46[3],out46);
mux
m47(W[93:92],d47[0],d47[1],d47[2],d47[3],out47);
mux
m48(W[95:94],d48[0],d48[1],d48[2],d48[3],out48);
mux
m49(W[97:96],d49[0],d49[1],d49[2],d49[3],out49);
mux
m50(W[99:98],d50[0],d50[1],d50[2],d50[3],out50);
mux
m51(W[101:100],d51[0],d51[1],d51[2],d51[3],out51);
mux
m52(W[103:102],d52[0],d52[1],d52[2],d52[3],out52);
mux
m53(W[105:104],d53[0],d53[1],d53[2],d53[3],out53);
```

```
mux
m54(W[107:106],d54[0],d54[1],d54[2],d54[3],out54);
mux
m55(W[109:108],d55[0],d55[1],d55[2],d55[3],out55);
mux
m56(W[111:110],d56[0],d56[1],d56[2],d56[3],out56);
mux
m57(W[113:112],d57[0],d57[1],d57[2],d57[3],out57);
mux
m58(W[115:114],d58[0],d58[1],d58[2],d58[3],out58);
mux
m59(W[117:116],d59[0],d59[1],d59[2],d59[3],out59);
mux
m60(W[119:118],d60[0],d60[1],d60[2],d60[3],out60);
mux
m61(W[121:120],d61[0],d61[1],d61[2],d61[3],out61);
mux
m62(W[123:122],d62[0],d62[1],d62[2],d62[3],out62);
mux
m63(W[125:124],d63[0],d63[1],d63[2],d63[3],out63);
mux
m64(W[127:126],d64[0],d64[1],d64[2],d64[3],out64);
mux
m65(W[129:128],d65[0],d65[1],d65[2],d65[3],out65);
mux
m66(W[131:130],d66[0],d66[1],d66[2],d66[3],out66);
mux
m67(W[133:132],d67[0],d67[1],d67[2],d67[3],out67);
mux
m68(W[135:134],d68[0],d68[1],d68[2],d68[3],out68);
mux
m69(W[137:136],d69[0],d69[1],d69[2],d69[3],out69);
```

```
        mux
m70(W[139:138],d70[0],d70[1],d70[2],d70[3],out70);
        mux
m71(W[141:140],d71[0],d71[1],d71[2],d71[3],out71);
        mux
m72(W[143:142],d72[0],d72[1],d72[2],d72[3],out72);

        wire [255:0]and1;
        wire [255:0]and2;
        assign and1 = out1 & out2 & out3 & out4 & out5 &
out6 & out7 & out8 & out9 & out10 &out11 & out12 & out13
& out14 & out15 & out16 & out17 & out18 & out19 & out20
& out21 & out22 & out23 & out24 & out25 & out26 & out27
& out28 & out29 & out30 & out31 & out32 & out33 & out34
& out35 & out36 & out37 & out38 & out39 & out40 & out41
& out42 & out43 & out44 & out45 & out46 & out47 & out48
& out49 & out50 & out51 & out52 & out53 & out54 & out55
& out56 & out57 & out58 & out59 & out60 & out61 & out62
& out63 & out64 & out65 & out66 & out67 & out68 & out69
& out70 & out71 & out72;
        wire u;
        pe_256 hjk(and1,and2,u);

        assign N[255:0] = and1[255:0];
endmodule
```

Ce.v:

```
module ce(w,r,out);
input [3:0]w;
input r;
output[3:0]out;
```

```
assign out[0]=w[0]& r;
assign out[1]=w[1]& r;
assign out[2]=w[2]& r;
assign out[3]=w[3]& r;
endmodule
```

Mux.v:

```
module mux(a,b,d,e,f,c);
input [1:0]a;
input [255:0]b;
input [255:0]d;
input [255:0]e;
input [255:0]f;

output [255:0]c;
reg [255:0]c;

always@*
begin
	if(a[0] == 0 && a[1] == 0)
		c[255:0] = b[255:0];
	if(a[0] == 1 && a[1] == 0)
		c[255:0] = d[255:0];
	if(a[0] == 0 && a[1] == 1)
		c[255:0] = e[255:0];
	if(a[0] == 1 && a[1] == 1)
		c[255:0] = f[255:0];
end
endmodule
```

Priority4.v:

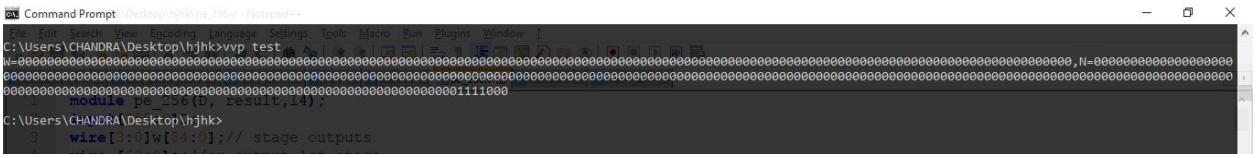
```
module priority4bit(D,Y,V);
```

```
input [3:0] D;
output [3:0] Y;
output V;

assign Y[0]=1&D[0];
assign Y[1]=1&D[1]&~D[0];
assign Y[2]=1&D[2]&~D[0]&~D[1];
assign Y[3]=1&D[3]&~D[0]&~D[1]&~D[2];
assign V = D[0]|D[1]|D[2]|D[3];
endmodule
```

Tcam_test.v:

```
module tcam_test;

        reg [143:0]W;
        wire [255:0]N;

        tcam t1(W,N);

        initial
        begin
                W = 144'd0;
        end

        initial
        begin
         $monitor("W=%b,N=%b",W,N);
        end
endmodule
```

Output:



```
C:\Users\CHANDRA\Desktop\hjhk>vvp test
W=000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000,N=0000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000001111000

C:\Users\CHANDRA\Desktop\hjhk>
```

Experiment no. 9:

Aim: To synthesize the Priority encoder and map it onto the FPGA.

Code:

Priority4bit.v:

```
module PriorityEncoder_4Bit(d,y,v);
        input [3:0]d;
        output [3:0]y;
        output v;
        wire v;
        wire [3:0]y;
        assign y[0] = d[0] & 1'b1;
        assign y[1] = d[1] & ~d[0] & 1'b1;
        assign y[2] = d[2] & ~d[1] & ~d[0] & 1'b1;
        assign y[3] = d[3] & ~d[2] & ~d[1] & ~d[0] & 1'b1;
        assign v = d[0] | d[1] | d[2] | d[3];

endmodule
```

Priority256bit.v:

```
// Usage: iverilog -o abc Priority256bit.v Priority4bit.v
Conversion_element.v
// vvp abc

module pe_256bit(d,y,v);
	input [255:0]d;
	output v;
	output [255:0]y;
	wire v;
	wire [255:0]y;
	wire [255:0]p,k,q,j,r,h,c,x,z;

	//first layer
	PriorityEncoder_4Bit	a1(d[	3	:	0	],y[	3	:	0
	],v1);
	PriorityEncoder_4Bit	a2(d[	7	:	4	],y[	7
	:	4	],v2);
	PriorityEncoder_4Bit	a3(d[	11	:	8	],y[	11
	:	8	],v3);
	PriorityEncoder_4Bit	a4(d[	15	:	12	],y[	15
	:	12	],v4);
	PriorityEncoder_4Bit	a5(d[	19	:	16	],y[	19
	:	16	],v5);
	PriorityEncoder_4Bit	a6(d[	23	:	20	],y[	23
	:	20	],v6);
	PriorityEncoder_4Bit	a7(d[	27	:	24	],y[	27
	:	24	],v7);
	PriorityEncoder_4Bit	a8(d[	31	:	28	],y[	31
	:	28	],v8);
	PriorityEncoder_4Bit	a9(d[	35	:	32	],y[	35
	:	32	],v9);
```

```
PriorityEncoder_4Bit a10(d[ 39 : 36 ],y[ 39 : 36 ],v10);
PriorityEncoder_4Bit a11(d[ 43 : 40 ],y[ 43 : 40 ],v11);
PriorityEncoder_4Bit a12(d[ 47 : 44 ],y[ 47 : 44 ],v12);
PriorityEncoder_4Bit a13(d[ 51 : 48 ],y[ 51 : 48 ],v13);
PriorityEncoder_4Bit a14(d[ 55 : 52 ],y[ 55 : 52 ],v14);
PriorityEncoder_4Bit a15(d[ 59 : 56 ],y[ 59 : 56 ],v15);
PriorityEncoder_4Bit a16(d[ 63 : 60 ],y[ 63 : 60 ],v16);
PriorityEncoder_4Bit a17(d[ 67 : 64 ],y[ 67 : 64 ],v17);
PriorityEncoder_4Bit a18(d[ 71 : 68 ],y[ 71 : 68 ],v18);
PriorityEncoder_4Bit a19(d[ 75 : 72 ],y[ 75 : 72 ],v19);
PriorityEncoder_4Bit a21(d[ 79 : 76 ],y[ 79 : 76 ],v20);
PriorityEncoder_4Bit a22(d[ 83 : 80 ],y[ 83 : 80 ],v21);
PriorityEncoder_4Bit a23(d[ 87 : 84 ],y[ 87 : 84 ],v22);
PriorityEncoder_4Bit a24(d[ 91 : 88 ],y[ 91 : 88 ],v23);
PriorityEncoder_4Bit a25(d[ 95 : 92 ],y[ 95 : 92 ],v24);
PriorityEncoder_4Bit a26(d[ 99 : 96 ],y[ 99 : 96 ],v25);
```

```
PriorityEncoder_4Bit a27(d[ 103 : 100 ],y[
103 : 100 ],v26);
PriorityEncoder_4Bit a28(d[ 107 : 104 ],y[
107 : 104 ],v27);
PriorityEncoder_4Bit a29(d[ 111 : 108 ],y[
111 : 108 ],v28);
PriorityEncoder_4Bit a30(d[ 115 : 112 ],y[
115 : 112 ],v29);
PriorityEncoder_4Bit a31(d[ 119 : 116 ],y[
119 : 116 ],v30);
PriorityEncoder_4Bit a32(d[ 123 : 120 ],y[
123 : 120 ],v31);
PriorityEncoder_4Bit a33(d[ 127 : 124 ],y[
127 : 124 ],v32);
PriorityEncoder_4Bit a34(d[ 131 : 128 ],y[
131 : 128 ],v33);
PriorityEncoder_4Bit a35(d[ 135 : 132 ],y[
135 : 132 ],v34);
PriorityEncoder_4Bit a36(d[ 139 : 136 ],y[
139 : 136 ],v35);
PriorityEncoder_4Bit a37(d[ 143 : 140 ],y[
143 : 140 ],v36);
PriorityEncoder_4Bit a38(d[ 147 : 144 ],y[
147 : 144 ],v37);
PriorityEncoder_4Bit a39(d[ 151 : 148 ],y[
151 : 148 ],v38);
PriorityEncoder_4Bit a40(d[ 155 : 152 ],y[
155 : 152 ],v39);
PriorityEncoder_4Bit a41(d[ 159 : 156 ],y[
159 : 156 ],v40);
PriorityEncoder_4Bit a42(d[ 163 : 160 ],y[
163 : 160 ],v41);
```

```
PriorityEncoder_4Bit a43(d[ 167 : 164 ],y[ 167 : 164 ],v42);
PriorityEncoder_4Bit a44(d[ 171 : 168 ],y[ 171 : 168 ],v43);
PriorityEncoder_4Bit a45(d[ 175 : 172 ],y[ 175 : 172 ],v44);
PriorityEncoder_4Bit a46(d[ 179 : 176 ],y[ 179 : 176 ],v45);
PriorityEncoder_4Bit a47(d[ 183 : 180 ],y[ 183 : 180 ],v46);
PriorityEncoder_4Bit a48(d[ 187 : 184 ],y[ 187 : 184 ],v47);
PriorityEncoder_4Bit a49(d[ 191 : 188 ],y[ 191 : 188 ],v48);
PriorityEncoder_4Bit a50(d[ 195 : 192 ],y[ 195 : 192 ],v49);
PriorityEncoder_4Bit a51(d[ 199 : 196 ],y[ 199 : 196 ],v50);
PriorityEncoder_4Bit a52(d[ 203 : 200 ],y[ 203 : 200 ],v51);
PriorityEncoder_4Bit a53(d[ 207 : 204 ],y[ 207 : 204 ],v52);
PriorityEncoder_4Bit a54(d[ 211 : 208 ],y[ 211 : 208 ],v53);
PriorityEncoder_4Bit a55(d[ 215 : 212 ],y[ 215 : 212 ],v54);
PriorityEncoder_4Bit a56(d[ 219 : 216 ],y[ 219 : 216 ],v55);
PriorityEncoder_4Bit a57(d[ 223 : 220 ],y[ 223 : 220 ],v56);
PriorityEncoder_4Bit a58(d[ 227 : 224 ],y[ 227 : 224 ],v57);
```

```
PriorityEncoder_4Bit a59(d[ 231 : 228 ],y[ 231 : 228 ],v58);
PriorityEncoder_4Bit a60(d[ 235 : 232 ],y[ 235 : 232 ],v59);
PriorityEncoder_4Bit a61(d[ 239 : 236 ],y[ 239 : 236 ],v60);
PriorityEncoder_4Bit a62(d[ 243 : 240 ],y[ 243 : 240 ],v61);
PriorityEncoder_4Bit a63(d[ 247 : 244 ],y[ 247 : 244 ],v62);
PriorityEncoder_4Bit a64(d[ 251 : 248 ],y[ 251 : 248 ],v63);
PriorityEncoder_4Bit a20(d[ 255 : 252 ],y[ 255 : 252 ],v64);

//second layer

assign p[ 0 ]=y[ 0 ] | y [ 1 ] | y [ 2 ] | y [ 3 ];
assign p[ 1 ]=y[ 4 ] | y [ 5 ] | y [ 6 ] | y [ 7 ];
assign p[ 2 ]=y[ 8 ] | y [ 9 ] | y [ 10 ] | y [ 11 ];
assign p[ 3 ]=y[ 12 ] | y [ 13 ] | y [ 14 ] | y [ 15 ];
assign p[ 4 ]=y[ 16 ] | y [ 17 ] | y [ 18 ] | y [ 19 ];
assign p[ 5 ]=y[ 20 ] | y [ 21 ] | y [ 22 ] | y [ 23 ];
assign p[ 6 ]=y[ 24 ] | y [ 25 ] | y [ 26 ] | y [ 27 ];
```

```
assign p[ 7 ]=y[ 28 ] | y [ 29 ] | y [ 30 ] |
y [ 31 ];
assign p[ 8 ]=y[ 32 ] | y [ 33 ] | y [ 34 ] |
y [ 35 ];
assign p[ 9 ]=y[ 36 ] | y [ 37 ] | y [ 38 ] |
y [ 39 ];
assign p[ 10 ]=y[ 40 ] | y [ 41 ] | y [ 42 ] |
y [ 43 ];
assign p[ 11 ]=y[ 44 ] | y [ 45 ] | y [ 46 ] |
y [ 47 ];
assign p[ 12 ]=y[ 48 ] | y [ 49 ] | y [ 50 ] |
y [ 51 ];
assign p[ 13 ]=y[ 52 ] | y [ 53 ] | y [ 54 ] |
y [ 55 ];
assign p[ 14 ]=y[ 56 ] | y [ 57 ] | y [ 58 ] |
y [ 59 ];
assign p[ 15 ]=y[ 60 ] | y [ 61 ] | y [ 62 ] |
y [ 63 ];
assign p[ 16 ]=y[ 64 ] | y [ 65 ] | y [ 66 ] |
y [ 67 ];
assign p[ 17 ]=y[ 68 ] | y [ 69 ] | y [ 70 ] |
y [ 71 ];
assign p[ 18 ]=y[ 72 ] | y [ 73 ] | y [ 74 ] |
y [ 75 ];
assign p[ 19 ]=y[ 76 ] | y [ 77 ] | y [ 78 ] |
y [ 79 ];
assign p[ 20 ]=y[ 80 ] | y [ 81 ] | y [ 82 ] |
y [ 83 ];
assign p[ 21 ]=y[ 84 ] | y [ 85 ] | y [ 86 ] |
y [ 87 ];
assign p[ 22 ]=y[ 88 ] | y [ 89 ] | y [ 90 ] |
y [ 91 ];
```

```
assign p[ 23 ]=y[ 92 ] | y [ 93 ] | y [ 94 ] |
y [ 95 ];
assign p[ 24 ]=y[ 96 ] | y [ 97 ] | y [ 98 ] |
y [ 99 ];
assign p[ 25 ]=y[ 100 ] | y [ 101 ] | y [ 102 ] |
y [ 103 ];
assign p[ 26 ]=y[ 104 ] | y [ 105 ] | y [ 106 ] |
y [ 107 ];
assign p[ 27 ]=y[ 108 ] | y [ 109 ] | y [ 110 ] |
y [ 111 ];
assign p[ 28 ]=y[ 112 ] | y [ 113 ] | y [ 114 ] |
y [ 115 ];
assign p[ 29 ]=y[ 116 ] | y [ 117 ] | y [ 118 ] |
y [ 119 ];
assign p[ 30 ]=y[ 120 ] | y [ 121 ] | y [ 122 ] |
y [ 123 ];
assign p[ 31 ]=y[ 124 ] | y [ 125 ] | y [ 126 ] |
y [ 127 ];
assign p[ 32 ]=y[ 128 ] | y [ 129 ] | y [ 130 ] |
y [ 131 ];
assign p[ 33 ]=y[ 132 ] | y [ 133 ] | y [ 134 ] |
y [ 135 ];
assign p[ 34 ]=y[ 136 ] | y [ 137 ] | y [ 138 ] |
y [ 139 ];
assign p[ 35 ]=y[ 140 ] | y [ 141 ] | y [ 142 ] |
y [ 143 ];
assign p[ 36 ]=y[ 144 ] | y [ 145 ] | y [ 146 ] |
y [ 147 ];
assign p[ 37 ]=y[ 148 ] | y [ 149 ] | y [ 150 ] |
y [ 151 ];
assign p[ 38 ]=y[ 152 ] | y [ 153 ] | y [ 154 ] |
y [ 155 ];
```

```
assign p[ 39 ]=y[ 156 ] | y [ 157 ] | y [ 158 ] |
y [ 159 ];
assign p[ 40 ]=y[ 160 ] | y [ 161 ] | y [ 162 ] |
y [ 163 ];
assign p[ 41 ]=y[ 164 ] | y [ 165 ] | y [ 166 ] |
y [ 167 ];
assign p[ 42 ]=y[ 168 ] | y [ 169 ] | y [ 170 ] |
y [ 171 ];
assign p[ 43 ]=y[ 172 ] | y [ 173 ] | y [ 174 ] |
y [ 175 ];
assign p[ 44 ]=y[ 176 ] | y [ 177 ] | y [ 178 ] |
y [ 179 ];
assign p[ 45 ]=y[ 180 ] | y [ 181 ] | y [ 182 ] |
y [ 183 ];
assign p[ 46 ]=y[ 184 ] | y [ 185 ] | y [ 186 ] |
y [ 187 ];
assign p[ 47 ]=y[ 188 ] | y [ 189 ] | y [ 190 ] |
y [ 191 ];
assign p[ 48 ]=y[ 192 ] | y [ 193 ] | y [ 194 ] |
y [ 195 ];
assign p[ 49 ]=y[ 196 ] | y [ 197 ] | y [ 198 ] |
y [ 199 ];
assign p[ 50 ]=y[ 200 ] | y [ 201 ] | y [ 202 ] |
y [ 203 ];
assign p[ 51 ]=y[ 204 ] | y [ 205 ] | y [ 206 ] |
y [ 207 ];
assign p[ 52 ]=y[ 208 ] | y [ 209 ] | y [ 210 ] |
y [ 211 ];
assign p[ 53 ]=y[ 212 ] | y [ 213 ] | y [ 214 ] |
y [ 215 ];
assign p[ 54 ]=y[ 216 ] | y [ 217 ] | y [ 218 ] |
y [ 219 ];
```

```
assign p[ 55 ]=y[ 220 ] | y [ 221 ] | y [ 222 ] | y [ 223 ];
assign p[ 56 ]=y[ 224 ] | y [ 225 ] | y [ 226 ] | y [ 227 ];
assign p[ 57 ]=y[ 228 ] | y [ 229 ] | y [ 230 ] | y [ 231 ];
assign p[ 58 ]=y[ 232 ] | y [ 233 ] | y [ 234 ] | y [ 235 ];
assign p[ 59 ]=y[ 236 ] | y [ 237 ] | y [ 238 ] | y [ 239 ];
assign p[ 60 ]=y[ 240 ] | y [ 241 ] | y [ 242 ] | y [ 243 ];
assign p[ 61 ]=y[ 244 ] | y [ 245 ] | y [ 246 ] | y [ 247 ];
assign p[ 62 ]=y[ 248 ] | y [ 249 ] | y [ 250 ] | y [ 251 ];
assign p[ 63 ]=y[ 252 ] | y [ 253 ] | y [ 254 ] | y [ 255 ];


//third layer
PriorityEncoder_4Bit b1(p[ 3 : 0 ],k[ 3 : 0 ],v65);
PriorityEncoder_4Bit b2(p[ 7 : 4 ],k[ 7 : 4 ],v66);
PriorityEncoder_4Bit b3(p[ 11 : 8 ],k[ 11 : 8 ],v67);
PriorityEncoder_4Bit b4(p[ 15 : 12 ],k[ 15 : 12 ],v68);
PriorityEncoder_4Bit b5(p[ 19 : 16 ],k[ 19 : 16 ],v69);
```

```
PriorityEncoder_4Bit b6(p[ 23 : 20 ],k[ 23 : 20 ],v70);
PriorityEncoder_4Bit b7(p[ 27 : 24 ],k[ 27 : 24 ],v71);
PriorityEncoder_4Bit b8(p[ 31 : 28 ],k[ 31 : 28 ],v72);
PriorityEncoder_4Bit b9(p[ 35 : 32 ],k[ 35 : 32 ],v78);
PriorityEncoder_4Bit b10(p[ 39 : 36 ],k[ 39 : 36 ],v73);
PriorityEncoder_4Bit b11(p[ 43 : 40 ],k[ 43 : 40 ],v74);
PriorityEncoder_4Bit b12(p[ 47 : 44 ],k[ 47 : 44 ],v75);
PriorityEncoder_4Bit b13(p[ 51 : 48 ],k[ 51 : 48 ],v76);
PriorityEncoder_4Bit b14(p[ 55 : 52 ],k[ 55 : 52 ],v77);
PriorityEncoder_4Bit b15(p[ 59 : 56 ],k[ 59 : 56 ],v79);
PriorityEncoder_4Bit b16(p[ 63 : 60 ],k[ 63 : 60 ],v80);

coversion_element u21(y[ 3 : 0 ],k[ 0 ],z[ 3 : 0 ]);
coversion_element u22(y[ 7 : 4 ],k[ 1 ],z[ 7 : 4 ]);
coversion_element u23(y[ 11 : 8 ],k[ 2 ],z[ 11 : 8 ]);
coversion_element u24(y[ 15 : 12 ],k[ 3 ],z[ 15 : 12 ]);
```

```
coversion_element u25(y[ 19 : 16 ],k[ 4 ],z[ 19 : 16 ]);
coversion_element u26(y[ 23 : 20 ],k[ 5 ],z[ 23 : 20 ]);
coversion_element u27(y[ 27 : 24 ],k[ 6 ],z[ 27 : 24 ]);
coversion_element u28(y[ 31 : 28 ],k[ 7 ],z[ 31 : 28 ]);
coversion_element u29(y[ 35 : 32 ],k[ 8 ],z[ 35 : 32 ]);
coversion_element u30(y[ 39 : 36 ],k[ 9 ],z[ 39 : 36 ]);
coversion_element u31(y[ 43 : 40 ],k[ 10 ],z[ 43 : 40 ]);
coversion_element u32(y[ 47 : 44 ],k[ 11 ],z[ 47 : 44 ]);
coversion_element u33(y[ 51 : 48 ],k[ 12 ],z[ 51 : 48 ]);
coversion_element u34(y[ 55 : 52 ],k[ 13 ],z[ 55 : 52 ]);
coversion_element u35(y[ 59 : 56 ],k[ 14 ],z[ 59 : 56 ]);
coversion_element u36(y[ 63 : 60 ],k[ 15 ],z[ 63 : 60 ]);
coversion_element u37(y[ 67 : 64 ],k[ 16 ],z[ 67 : 64 ]);
coversion_element u38(y[ 71 : 68 ],k[ 17 ],z[ 71 : 68 ]);
coversion_element u39(y[ 75 : 72 ],k[ 18 ],z[ 75 : 72 ]);
coversion_element u40(y[ 79 : 76 ],k[ 19 ],z[ 79 : 76 ]);
```

```
coversion_element u41(y[ 83 : 80 ],k[ 20 ],z[ 83 : 80 ]);
coversion_element u42(y[ 87 : 84 ],k[ 21 ],z[ 87 : 84 ]);
coversion_element u43(y[ 91 : 88 ],k[ 22 ],z[ 91 : 88 ]);
coversion_element u44(y[ 95 : 92 ],k[ 23 ],z[ 95 : 92 ]);
coversion_element u45(y[ 99 : 96 ],k[ 24 ],z[ 99 : 96 ]);
coversion_element u46(y[ 103 : 100 ],k[ 25 ],z[ 103 : 100 ]);
coversion_element u47(y[ 107 : 104 ],k[ 26 ],z[ 107 : 104 ]);
coversion_element u48(y[ 111 : 108 ],k[ 27 ],z[ 111 : 108 ]);
coversion_element u49(y[ 115 : 112 ],k[ 28 ],z[ 115 : 112 ]);
coversion_element u50(y[ 119 : 116 ],k[ 29 ],z[ 119 : 116 ]);
coversion_element u51(y[ 123 : 120 ],k[ 30 ],z[ 123 : 120 ]);
coversion_element u52(y[ 127 : 124 ],k[ 31 ],z[ 127 : 124 ]);
coversion_element u53(y[ 131 : 128 ],k[ 32 ],z[ 131 : 128 ]);
coversion_element u54(y[ 135 : 132 ],k[ 33 ],z[ 135 : 132 ]);
coversion_element u55(y[ 139 : 136 ],k[ 34 ],z[ 139 : 136 ]);
coversion_element u56(y[ 143 : 140 ],k[ 35 ],z[ 143 : 140 ]);
```

```
coversion_element u57(y[ 147 : 144 ],k[ 36 ],z[ 147 : 144 ]);
coversion_element u58(y[ 151 : 148 ],k[ 37 ],z[ 151 : 148 ]);
coversion_element u60(y[ 155 : 152 ],k[ 38 ],z[ 155 : 152 ]);
coversion_element u59(y[ 159 : 156 ],k[ 39 ],z[ 159 : 156 ]);
coversion_element u84(y[ 163 : 160 ],k[ 40 ],z[ 163 : 160 ]);
coversion_element u61(y[ 167 : 164 ],k[ 41 ],z[ 167 : 164 ]);
coversion_element u62(y[ 171 : 168 ],k[ 42 ],z[ 171 : 168 ]);
coversion_element u63(y[ 175 : 172 ],k[ 43 ],z[ 175 : 172 ]);
coversion_element u64(y[ 179 : 176 ],k[ 44 ],z[ 179 : 176 ]);
coversion_element u65(y[ 183 : 180 ],k[ 45 ],z[ 183 : 180 ]);
coversion_element u66(y[ 187 : 184 ],k[ 46 ],z[ 187 : 184 ]);
coversion_element u67(y[ 191 : 188 ],k[ 47 ],z[ 191 : 188 ]);
coversion_element u68(y[ 195 : 192 ],k[ 48 ],z[ 195 : 192 ]);
coversion_element u69(y[ 199 : 196 ],k[ 49 ],z[ 199 : 196 ]);
coversion_element u70(y[ 203 : 200 ],k[ 50 ],z[ 203 : 200 ]);
coversion_element u71(y[ 207 : 204 ],k[ 51 ],z[ 207 : 204 ]);
```

```
coversion_element u72(y[ 211 : 208 ],k[ 52 ],z[ 211 : 208 ]);
coversion_element u73(y[ 215 : 212 ],k[ 53 ],z[ 215 : 212 ]);
coversion_element u74(y[ 219 : 216 ],k[ 54 ],z[ 219 : 216 ]);
coversion_element u75(y[ 223 : 220 ],k[ 55 ],z[ 223 : 220 ]);
coversion_element u76(y[ 227 : 224 ],k[ 56 ],z[ 227 : 224 ]);
coversion_element u77(y[ 231 : 228 ],k[ 57 ],z[ 231 : 228 ]);
coversion_element u78(y[ 235 : 232 ],k[ 58 ],z[ 235 : 232 ]);
coversion_element u79(y[ 239 : 236 ],k[ 59 ],z[ 239 : 236 ]);
coversion_element u80(y[ 243 : 240 ],k[ 60 ],z[ 243 : 240 ]);
coversion_element u81(y[ 247 : 244 ],k[ 61 ],z[ 247 : 244 ]);
coversion_element u82(y[ 251 : 248 ],k[ 62 ],z[ 251 : 248 ]);
coversion_element u83(y[ 255 : 252 ],k[ 63 ],z[ 255 : 252 ]);

//fourth layer

assign q[ 0 ]=k[ 0 ] | k [ 1 ] | k [ 2 ] | k [ 3 ];
assign q[ 1 ]=k[ 4 ] | k [ 5 ] | k [ 6 ] | k [ 7 ];
```

```
assign q[ 2 ]=k[ 8 ] | k [ 9 ] | k [ 10 ] | k [ 11 ];
assign q[ 3 ]=k[ 12 ] | k [ 13 ] | k [ 14 ] | k [ 15 ];
assign q[ 4 ]=k[ 16 ] | k [ 17 ] | k [ 18 ] | k [ 19 ];
assign q[ 5 ]=k[ 20 ] | k [ 21 ] | k [ 22 ] | k [ 23 ];
assign q[ 6 ]=k[ 24 ] | k [ 25 ] | k [ 26 ] | k [ 27 ];
assign q[ 7 ]=k[ 28 ] | k [ 29 ] | k [ 30 ] | k [ 31 ];
assign q[ 8 ]=k[ 32 ] | k [ 33 ] | k [ 34 ] | k [ 35 ];
assign q[ 9 ]=k[ 36 ] | k [ 37 ] | k [ 38 ] | k [ 39 ];
assign q[ 10 ]=k[ 40 ] | k [ 41 ] | k [ 42 ] | k [ 43 ];
assign q[ 11 ]=k[ 44 ] | k [ 45 ] | k [ 46 ] | k [ 47 ];
assign q[ 12 ]=k[ 48 ] | k [ 49 ] | k [ 50 ] | k [ 51 ];
assign q[ 13 ]=k[ 52 ] | k [ 53 ] | k [ 54 ] | k [ 55 ];
assign q[ 14 ]=k[ 56 ] | k [ 57 ] | k [ 58 ] | k [ 59 ];
assign q[ 15 ]=k[ 60 ] | k [ 61 ] | k [ 62 ] | k [ 63 ];

//fifth layer
```

```
PriorityEncoder_4Bit c1(q[ 3 : 0 ],j[ 3 : 0 ],v81);
PriorityEncoder_4Bit c2(q[ 7 : 4 ],j[ 7 : 4 ],v82);
PriorityEncoder_4Bit c3(q[ 11 : 8 ],j[ 11 : 8 ],v83);
PriorityEncoder_4Bit c4(q[ 15 : 12 ],j[ 15 : 12 ],v84);

coversion_element u5 (z[3:0],j[0],x[3:0]);
coversion_element u6 (z[7:4],j[0],x[7:4]);
coversion_element u7 (z[11:8],j[0],x[11:8]);
coversion_element u8 (z[15:12],j[0],x[15:12]);
coversion_element u9 (z[19:16],j[1],x[19:16]);
coversion_element u10 (z[23:20],j[1],x[23:20]);
coversion_element u11 (z[27:24],j[1],x[27:24]);
coversion_element u12 (z[31:28],j[1],x[31:28]);
coversion_element u13 (z[35:32],j[2],x[35:32]);
coversion_element u14 (z[39:36],j[2],x[39:36]);
coversion_element u15 (z[43:40],j[2],x[43:40]);
coversion_element u16 (z[47:44],j[2],x[47:44]);
coversion_element u17 (z[51:48],j[3],x[51:48]);
coversion_element u18 (z[55:52],j[3],x[55:52]);
coversion_element u19 (z[59:56],j[3],x[59:56]);
coversion_element u20 (z[63:60],j[3],x[63:60]);
coversion_element u85 (z[67:64],j[4],x[67:64]);
coversion_element u103(z[ 71 : 68 ],j[ 4 ],x[ 71 : 68 ]);
coversion_element u104(z[ 75 : 72 ],j[ 4 ],x[ 75 : 72 ]);
coversion_element u105(z[ 79 : 76 ],j[ 4 ],x[ 79 : 76 ]);
```

```
coversion_element u106(z[ 83 : 80 ],j[ 5 ],x[ 83 : 80 ]);
coversion_element u107(z[ 87 : 84 ],j[ 5 ],x[ 87 : 84 ]);
coversion_element u108(z[ 91 : 88 ],j[ 5 ],x[ 91 : 88 ]);
coversion_element u109(z[ 95 : 92 ],j[ 5 ],x[ 95 : 92 ]);
coversion_element u110(z[ 99 : 96 ],j[ 6 ],x[ 99 : 96 ]);
coversion_element u111(z[ 103 : 100 ],j[ 6 ],x[ 103 : 100 ]);
coversion_element u112(z[ 107 : 104 ],j[ 6 ],x[ 107 : 104 ]);
coversion_element u113(z[ 111 : 108 ],j[ 6 ],x[ 111 : 108 ]);
coversion_element u114(z[ 115 : 112 ],j[ 7 ],x[ 115 : 112 ]);
coversion_element u115(z[ 119 : 116 ],j[ 7 ],x[ 119 : 116 ]);
coversion_element u116(z[ 123 : 120 ],j[ 7 ],x[ 123 : 120 ]);
coversion_element u117(z[ 127 : 124 ],j[ 7 ],x[ 127 : 124 ]);
coversion_element u118(z[ 131 : 128 ],j[ 8 ],x[ 131 : 128 ]);
coversion_element u119(z[ 135 : 132 ],j[ 8 ],x[ 135 : 132 ]);
coversion_element u120(z[ 139 : 136 ],j[ 8 ],x[ 139 : 136 ]);
coversion_element u121(z[ 143 : 140 ],j[ 8 ],x[ 143 : 140 ]);
```

```
coversion_element u122(z[ 147 : 144 ],j[ 9 ],x[ 147 : 144 ]);
coversion_element u123(z[ 151 : 148 ],j[ 9 ],x[ 151 : 148 ]);
coversion_element u124(z[ 155 : 152 ],j[ 9 ],x[ 155 : 152 ]);
coversion_element u125(z[ 159 : 156 ],j[ 9 ],x[ 159 : 156 ]);
coversion_element u126(z[ 163 : 160 ],j[ 10 ],x[ 163 : 160 ]);
coversion_element u127(z[ 167 : 164 ],j[ 10 ],x[ 167 : 164 ]);
coversion_element u128(z[ 171 : 168 ],j[ 10 ],x[ 171 : 168 ]);
coversion_element u129(z[ 175 : 172 ],j[ 10 ],x[ 175 : 172 ]);
coversion_element u130(z[ 179 : 176 ],j[ 11 ],x[ 179 : 176 ]);
coversion_element u131(z[ 183 : 180 ],j[ 11 ],x[ 183 : 180 ]);
coversion_element u132(z[ 187 : 184 ],j[ 11 ],x[ 187 : 184 ]);
coversion_element u133(z[ 191 : 188 ],j[ 11 ],x[ 191 : 188 ]);
coversion_element u134(z[ 195 : 192 ],j[ 12 ],x[ 195 : 192 ]);
coversion_element u135(z[ 199 : 196 ],j[ 12 ],x[ 199 : 196 ]);
coversion_element u136(z[ 203 : 200 ],j[ 12 ],x[ 203 : 200 ]);
coversion_element u137(z[ 207 : 204 ],j[ 12 ],x[ 207 : 204 ]);
```

```
coversion_element u138(z[ 211 : 208 ],j[ 13 ],x[ 211 : 208 ]);
coversion_element u139(z[ 215 : 212 ],j[ 13 ],x[ 215 : 212 ]);
coversion_element u140(z[ 219 : 216 ],j[ 13 ],x[ 219 : 216 ]);
coversion_element u141(z[ 223 : 220 ],j[ 13 ],x[ 223 : 220 ]);
coversion_element u142(z[ 227 : 224 ],j[ 14 ],x[ 227 : 224 ]);
coversion_element u143(z[ 231 : 228 ],j[ 14 ],x[ 231 : 228 ]);
coversion_element u144(z[ 235 : 232 ],j[ 14 ],x[ 235 : 232 ]);
coversion_element u145(z[ 239 : 236 ],j[ 14 ],x[ 239 : 236 ]);
coversion_element u146(z[ 243 : 240 ],j[ 15 ],x[ 243 : 240 ]);
coversion_element u147(z[ 247 : 244 ],j[ 15 ],x[ 247 : 244 ]);
coversion_element u148(z[ 251 : 248 ],j[ 15 ],x[ 251 : 248 ]);
coversion_element u149(z[ 255 : 252 ],j[ 15 ],x[ 255 : 252 ]);


//sixth layer


assign r[ 0 ]=j[ 0 ] | j [1 ] | j [2 ] | j [3 ];
assign r[ 1 ]=j[ 4 ] | j [5 ] | j [6 ] | j [7 ];
assign r[ 2 ]=j[ 8 ] | j [9 ] | j [10 ] | j [11 ];
```

```
assign r[ 3 ]=j[ 12 ] | j [13 ] | j [14 ] | j [15 ];

//seventh layer
PriorityEncoder_4Bit d1(r[3:0],h[3:0],v85);

//eighth layer
//assign t[0] = h[0]|h[1]|h[2]|h[3];

coversion_element j86 (x[ 3 : 0 ],h[ 0
],c[ 3 : 0 ]);
coversion_element j87 (x[ 7 : 4 ],h[ 0
],c[ 7 : 4 ]);
coversion_element j88 (x[ 11 : 8 ],h[ 0
],c[ 11 : 8 ]);
coversion_element j89 (x[ 15 : 12 ],h[ 0
],c[ 15 : 12 ]);
coversion_element j90 (x[ 19 : 16 ],h[ 0
],c[ 19 : 16 ]);
coversion_element j91 (x[ 23 : 20 ],h[ 0
],c[ 23 : 20 ]);
coversion_element j92 (x[ 27 : 24 ],h[ 0
],c[ 27 : 24 ]);
coversion_element j93 (x[ 31 : 28 ],h[ 0
],c[ 31 : 28 ]);
coversion_element j94 (x[ 35 : 32 ],h[ 0
],c[ 35 : 32 ]);
coversion_element j95 (x[ 39 : 36 ],h[ 0
],c[ 39 : 36 ]);
coversion_element j96 (x[ 43 : 40 ],h[ 0
],c[ 43 : 40 ]);
coversion_element j97 (x[ 47 : 44 ],h[ 0
],c[ 47 : 44 ]);
```

```
coversion_element j98 (x[ 51 : 48 ],h[ 0
],c[ 51 : 48 ]);
coversion_element j99 (x[ 55 : 52 ],h[ 0
],c[ 55 : 52 ]);
coversion_element j100 (x[ 59 : 56 ],h[ 0
],c[ 59 : 56 ]);
coversion_element j101 (x[ 63 : 60 ],h[ 0
],c[ 63 : 60 ]);
coversion_element j102 (x[ 67 : 64 ],h[ 1
],c[ 67 : 64 ]);
coversion_element j103 (x[ 71 : 68 ],h[ 1
],c[ 71 : 68 ]);
coversion_element j104 (x[ 75 : 72 ],h[ 1
],c[ 75 : 72 ]);
coversion_element j105 (x[ 79 : 76 ],h[ 1
],c[ 79 : 76 ]);
coversion_element j106 (x[ 83 : 80 ],h[ 1
],c[ 83 : 80 ]);
coversion_element j107 (x[ 87 : 84 ],h[ 1
],c[ 87 : 84 ]);
coversion_element j108 (x[ 91 : 88 ],h[ 1
],c[ 91 : 88 ]);
coversion_element j109 (x[ 95 : 92 ],h[ 1
],c[ 95 : 92 ]);
coversion_element j110 (x[ 99 : 96 ],h[ 1
],c[ 99 : 96 ]);
coversion_element j111 (x[ 103 : 100 ],h[ 1
],c[ 103 : 100 ]);
coversion_element j112 (x[ 107 : 104 ],h[ 1
],c[ 107 : 104 ]);
coversion_element j113 (x[ 111 : 108 ],h[ 1
],c[ 111 : 108 ]);
```

```
coversion_element j114 (x[ 115 : 112 ],h[ 1
],c[ 115 : 112 ]);
coversion_element j115 (x[ 119 : 116 ],h[ 1
],c[ 119 : 116 ]);
coversion_element j116 (x[ 123 : 120 ],h[ 1
],c[ 123 : 120 ]);
coversion_element j117 (x[ 127 : 124 ],h[ 1
],c[ 127 : 124 ]);
coversion_element j118 (x[ 131 : 128 ],h[ 2
],c[ 131 : 128 ]);
coversion_element j119 (x[ 135 : 132 ],h[ 2
],c[ 135 : 132 ]);
coversion_element j120 (x[ 139 : 136 ],h[ 2
],c[ 139 : 136 ]);
coversion_element j121 (x[ 143 : 140 ],h[ 2
],c[ 143 : 140 ]);
coversion_element j122 (x[ 147 : 144 ],h[ 2
],c[ 147 : 144 ]);
coversion_element j123 (x[ 151 : 148 ],h[ 2
],c[ 151 : 148 ]);
coversion_element j124 (x[ 155 : 152 ],h[ 2
],c[ 155 : 152 ]);
coversion_element j125 (x[ 159 : 156 ],h[ 2
],c[ 159 : 156 ]);
coversion_element j126 (x[ 163 : 160 ],h[ 2
],c[ 163 : 160 ]);
coversion_element j127 (x[ 167 : 164 ],h[ 2
],c[ 167 : 164 ]);
coversion_element j128 (x[ 171 : 168 ],h[ 2
],c[ 171 : 168 ]);
coversion_element j129 (x[ 175 : 172 ],h[ 2
],c[ 175 : 172 ]);
```

coversion_element j130 (x[179 : 176],h[2],c[179 : 176]);
coversion_element j131 (x[183 : 180],h[2],c[183 : 180]);
coversion_element j132 (x[187 : 184],h[2],c[187 : 184]);
coversion_element j133 (x[191 : 188],h[2],c[191 : 188]);
coversion_element j134 (x[195 : 192],h[3],c[195 : 192]);
coversion_element j135 (x[199 : 196],h[3],c[199 : 196]);
coversion_element j136 (x[203 : 200],h[3],c[203 : 200]);
coversion_element j137 (x[207 : 204],h[3],c[207 : 204]);
coversion_element j138 (x[211 : 208],h[3],c[211 : 208]);
coversion_element j139 (x[215 : 212],h[3],c[215 : 212]);
coversion_element j140 (x[219 : 216],h[3],c[219 : 216]);
coversion_element j141 (x[223 : 220],h[3],c[223 : 220]);
coversion_element j142 (x[227 : 224],h[3],c[227 : 224]);
coversion_element j143 (x[231 : 228],h[3],c[231 : 228]);
coversion_element j144 (x[235 : 232],h[3],c[235 : 232]);
coversion_element j145 (x[239 : 236],h[3],c[239 : 236]);

```
        coversion_element  j146 (x[   243  :      240  ],h[  3
],c[  243  :      240  ]);
        coversion_element  j147 (x[   247  :      244  ],h[  3
],c[  247  :      244  ]);
        coversion_element  j148 (x[   251  :      248  ],h[  3
],c[  251  :      248  ]);
        coversion_element  j149 (x[   255  :      252  ],h[  3
],c[  255  :      252  ]);

encoder_256 prrr(outfi,result,clk);


endmodule

module pe_256bit_Test();
reg [255:0] D;
wire [255:0] Y;
wire V;

pe_256bit i(D,Y,V);

initial
begin
// Initialize Inputs
D = 0;

#100;
#2 D = 256'b00001;
#2 D = 256'b00010;
#2 D = 256'b00011;
#2 D = 256'b00100;
#2 D = 256'b10101;
end
initial
```

```
begin
$monitor("time=",$time,, "D=%b : Y=%b V=%b",D,Y,V);
end
endmodule
```

Conversion_element.v:

```
module coversion_element(a, b, res);
input [3:0]a;
input b;
output [3:0]res;
	wire [3:0]res;

	assign res[0] = a[0] & b;
	assign res[1] = a[1] & b;
	assign res[2] = a[2] & b;
	assign res[3] = a[3] & b;

endmodule
```

Encoder256_8.v

```
module encoder_256 (in, out, clk);


   input [255:0]in;

   output [7:0]out;

   input clk;


   assign out[0] = in[1] | in[3] | in[5] | in[7] | in[9] | in[11] |
in[13] | in[15] | in[17] | in[19] | in[21] | in[23] | in[25] | in[27] |
in[29] | in[31] | in[33] | in[35] | in[37] | in[39] | in[41] | in[43] |
```

in[45] | in[47] | in[49] | in[51] | in[53] | in[55] | in[57] | in[59] | in[61] | in[63] | in[65] | in[67] | in[69] | in[71] | in[73] | in[75] | in[77] | in[79] | in[81] | in[83] | in[85] | in[87] | in[89] | in[91] | in[93] | in[95] | in[97] | in[99] | in[101] | in[103] | in[105] | in[107] | in[109] | in[111] | in[113] | in[115] | in[117] | in[119] | in[121] | in[123] | in[125] | in[127] | in[129] | in[131] | in[133] | in[135] | in[137] | in[139] | in[141] | in[143] | in[145] | in[147] | in[149] | in[151] | in[153] | in[155] | in[157] | in[159] | in[161] | in[163] | in[165] | in[167] | in[169] | in[171] | in[173] | in[175] | in[177] | in[179] | in[181] | in[183] | in[185] | in[187] | in[189] | in[191] | in[193] | in[195] | in[197] | in[199] | in[201] | in[203] | in[205] | in[207] | in[209] | in[211] | in[213] | in[215] | in[217] | in[219] | in[221] | in[223] | in[225] | in[227] | in[229] | in[231] | in[233] | in[235] | in[237] | in[239] | in[241] | in[243] | in[245] | in[247] | in[249] | in[251] | in[253] | in[255];

assign out[1] = in[2] | in[3] | in[6] | in[7] | in[10] | in[11] | in[14] | in[15] | in[18] | in[19] | in[22] | in[23] | in[26] | in[27] | in[30] | in[31] | in[34] | in[35] | in[38] | in[39] | in[42] | in[43] | in[46] | in[47] | in[50] | in[51] | in[54] | in[55] | in[58] | in[59] | in[62] | in[63] | in[66] | in[67] | in[70] | in[71] | in[74] | in[75] | in[78] | in[79] | in[82] | in[83] | in[86] | in[87] | in[90] | in[91] | in[94] | in[95] | in[98] | in[99] | in[102] | in[103] | in[106] | in[107] | in[110] | in[111] | in[114] | in[115] | in[118] | in[119] | in[122] | in[123] | in[126] | in[127] | in[130] | in[131] | in[134] | in[135] | in[138] | in[139] | in[142] | in[143] | in[146] | in[147] | in[150] | in[151] | in[154] | in[155] | in[158] | in[159] | in[162] | in[163] | in[166] | in[167] | in[170] | in[171] | in[174] | in[175] | in[178] | in[179] | in[182] | in[183] |

in[186] | in[187] | in[190] | in[191] | in[194] | in[195] | in[198] | in[199] | in[202] | in[203] | in[206] | in[207] | in[210] | in[211] | in[214] | in[215] | in[218] | in[219] | in[222] | in[223] | in[226] | in[227] | in[230] | in[231] | in[234] | in[235] | in[238] | in[239] | in[242] | in[243] | in[246] | in[247] | in[250] | in[251] | in[254] | in[255];

assign out[2] = in[4] | in[5] | in[6] | in[7] | in[12] | in[13] | in[14] | in[15] | in[20] | in[21] | in[22] | in[23] | in[28] | in[29] | in[30] | in[31] | in[36] | in[37] | in[38] | in[39] | in[44] | in[45] | in[46] | in[47] | in[52] | in[53] | in[54] | in[55] | in[60] | in[61] | in[62] | in[63] | in[68] | in[69] | in[70] | in[71] | in[76] | in[77] | in[78] | in[79] | in[84] | in[85] | in[86] | in[87] | in[92] | in[93] | in[94] | in[95] | in[100] | in[101] | in[102] | in[103] | in[108] | in[109] | in[110] | in[111] | in[116] | in[117] | in[118] | in[119] | in[124] | in[125] | in[126] | in[127] | in[132] | in[133] | in[134] | in[135] | in[140] | in[141] | in[142] | in[143] | in[148] | in[149] | in[150] | in[151] | in[156] | in[157] | in[158] | in[159] | in[164] | in[165] | in[166] | in[167] | in[172] | in[173] | in[174] | in[175] | in[180] | in[181] | in[182] | in[183] | in[188] | in[189] | in[190] | in[191] | in[196] | in[197] | in[198] | in[199] | in[204] | in[205] | in[206] | in[207] | in[212] | in[213] | in[214] | in[215] | in[220] | in[221] | in[222] | in[223] | in[228] | in[229] | in[230] | in[231] | in[236] | in[237] | in[238] | in[239] | in[244] | in[245] | in[246] | in[247] | in[252] | in[253] | in[254] | in[255];

assign out[3] = in[8] | in[9] | in[10] | in[11] | in[12] | in[13] | in[14] | in[15] | in[24] | in[25] | in[26] | in[27] | in[28] | in[29] |

in[30] | in[31] | in[40] | in[41] | in[42] | in[43] | in[44] | in[45] | in[46] | in[47] | in[56] | in[57] | in[58] | in[59] | in[60] | in[61] | in[62] | in[63] | in[72] | in[73] | in[74] | in[75] | in[76] | in[77] | in[78] | in[79] | in[88] | in[89] | in[90] | in[91] | in[92] | in[93] | in[94] | in[95] | in[104] | in[105] | in[106] | in[107] | in[108] | in[109] | in[110] | in[111] | in[120] | in[121] | in[122] | in[123] | in[124] | in[125] | in[126] | in[127] | in[136] | in[137] | in[138] | in[139] | in[140] | in[141] | in[142] | in[143] | in[152] | in[153] | in[154] | in[155] | in[156] | in[157] | in[158] | in[159] | in[168] | in[169] | in[170] | in[171] | in[172] | in[173] | in[174] | in[175] | in[184] | in[185] | in[186] | in[187] | in[188] | in[189] | in[190] | in[191] | in[200] | in[201] | in[202] | in[203] | in[204] | in[205] | in[206] | in[207] | in[216] | in[217] | in[218] | in[219] | in[220] | in[221] | in[222] | in[223] | in[232] | in[233] | in[234] | in[235] | in[236] | in[237] | in[238] | in[239] | in[248] | in[249] | in[250] | in[251] | in[252] | in[253] | in[254] | in[255];

assign out[4] = in[16] | in[17] | in[18] | in[19] | in[20] | in[21] | in[22] | in[23] | in[24] | in[25] | in[26] | in[27] | in[28] | in[29] | in[30] | in[31] | in[48] | in[49] | in[50] | in[51] | in[52] | in[53] | in[54] | in[55] | in[56] | in[57] | in[58] | in[59] | in[60] | in[61] | in[62] | in[63] | in[80] | in[81] | in[82] | in[83] | in[84] | in[85] | in[86] | in[87] | in[88] | in[89] | in[90] | in[91] | in[92] | in[93] | in[94] | in[95] | in[112] | in[113] | in[114] | in[115] | in[116] | in[117] | in[118] | in[119] | in[120] | in[121] | in[122] | in[123] | in[124] | in[125] | in[126] | in[127] | in[144] | in[145] | in[146] | in[147] | in[148] | in[149] | in[150] | in[151] | in[152] | in[153] | in[154] | in[155] | in[156] | in[157] | in[158] | in[159] | in[176] | in[177] | in[178] | in[179] | in[180]

| in[181] | in[182] | in[183] | in[184] | in[185] | in[186] | in[187] | in[188] | in[189] | in[190] | in[191] | in[208] | in[209] | in[210] | in[211] | in[212] | in[213] | in[214] | in[215] | in[216] | in[217] | in[218] | in[219] | in[220] | in[221] | in[222] | in[223] | in[240] | in[241] | in[242] | in[243] | in[244] | in[245] | in[246] | in[247] | in[248] | in[249] | in[250] | in[251] | in[252] | in[253] | in[254] | in[255];

assign out[5] = in[32] | in[33] | in[34] | in[35] | in[36] | in[37] | in[38] | in[39] | in[40] | in[41] | in[42] | in[43] | in[44] | in[45] | in[46] | in[47] | in[48] | in[49] | in[50] | in[51] | in[52] | in[53] | in[54] | in[55] | in[56] | in[57] | in[58] | in[59] | in[60] | in[61] | in[62] | in[63] | in[96] | in[97] | in[98] | in[99] | in[100] | in[101] | in[102] | in[103] | in[104] | in[105] | in[106] | in[107] | in[108] | in[109] | in[110] | in[111] | in[112] | in[113] | in[114] | in[115] | in[116] | in[117] | in[118] | in[119] | in[120] | in[121] | in[122] | in[123] | in[124] | in[125] | in[126] | in[127] | in[160] | in[161] | in[162] | in[163] | in[164] | in[165] | in[166] | in[167] | in[168] | in[169] | in[170] | in[171] | in[172] | in[173] | in[174] | in[175] | in[176] | in[177] | in[178] | in[179] | in[180] | in[181] | in[182] | in[183] | in[184] | in[185] | in[186] | in[187] | in[188] | in[189] | in[190] | in[191] | in[224] | in[225] | in[226] | in[227] | in[228] | in[229] | in[230] | in[231] | in[232] | in[233] | in[234] | in[235] | in[236] | in[237] | in[238] | in[239] | in[240] | in[241] | in[242] | in[243] | in[244] | in[245] | in[246] | in[247] | in[248] | in[249] | in[250] | in[251] | in[252] | in[253] | in[254] | in[255];

```
    assign out[6] = in[64] | in[65] | in[66] | in[67] | in[68] |
in[69] | in[70] | in[71] | in[72] | in[73] | in[74] | in[75] | in[76] |
in[77] | in[78] | in[79] | in[80] | in[81] | in[82] | in[83] | in[84] |
in[85] | in[86] | in[87] | in[88] | in[89] | in[90] | in[91] | in[92] |
in[93] | in[94] | in[95] | in[96] | in[97] | in[98] | in[99] | in[100]
| in[101] | in[102] | in[103] | in[104] | in[105] | in[106] |
in[107] | in[108] | in[109] | in[110] | in[111] | in[112] | in[113]
| in[114] | in[115] | in[116] | in[117] | in[118] | in[119] |
in[120] | in[121] | in[122] | in[123] | in[124] | in[125] | in[126]
| in[127] | in[192] | in[193] | in[194] | in[195] | in[196] |
in[197] | in[198] | in[199] | in[200] | in[201] | in[202] | in[203]
| in[204] | in[205] | in[206] | in[207] | in[208] | in[209] |
in[210] | in[211] | in[212] | in[213] | in[214] | in[215] | in[216]
| in[217] | in[218] | in[219] | in[220] | in[221] | in[222] |
in[223] | in[224] | in[225] | in[226] | in[227] | in[228] | in[229]
| in[230] | in[231] | in[232] | in[233] | in[234] | in[235] |
in[236] | in[237] | in[238] | in[239] | in[240] | in[241] | in[242]
| in[243] | in[244] | in[245] | in[246] | in[247] | in[248] |
in[249] | in[250] | in[251] | in[252] | in[253] | in[254] |
in[255];


    assign out[7] = in[128] | in[129] | in[130] | in[131] | in[132]
| in[133] | in[134] | in[135] | in[136] | in[137] | in[138] |
in[139] | in[140] | in[141] | in[142] | in[143] | in[144] | in[145]
| in[146] | in[147] | in[148] | in[149] | in[150] | in[151] |
in[152] | in[153] | in[154] | in[155] | in[156] | in[157] | in[158]
| in[159] | in[160] | in[161] | in[162] | in[163] | in[164] |
in[165] | in[166] | in[167] | in[168] | in[169] | in[170] | in[171]
| in[172] | in[173] | in[174] | in[175] | in[176] | in[177] |
in[178] | in[179] | in[180] | in[181] | in[182] | in[183] | in[184]
```

```
| in[185] | in[186] | in[187] | in[188] | in[189] | in[190] |
in[191] | in[192] | in[193] | in[194] | in[195] | in[196] | in[197]
| in[198] | in[199] | in[200] | in[201] | in[202] | in[203] |
in[204] | in[205] | in[206] | in[207] | in[208] | in[209] | in[210]
| in[211] | in[212] | in[213] | in[214] | in[215] | in[216] |
in[217] | in[218] | in[219] | in[220] | in[221] | in[222] | in[223]
| in[224] | in[225] | in[226] | in[227] | in[228] | in[229] |
in[230] | in[231] | in[232] | in[233] | in[234] | in[235] | in[236]
| in[237] | in[238] | in[239] | in[240] | in[241] | in[242] |
in[243] | in[244] | in[245] | in[246] | in[247] | in[248] | in[249]
| in[250] | in[251] | in[252] | in[253] | in[254] | in[255];


endmodule
```

rca.v

```
module rca3(s,cout,a,b,c);


input [2:0]a,b;

input c;

output [2:0]s;

output cout;

wire [1:0]w;


fulladder f1(s[0],w[0],a[0],b[0],c);

fulladder f2(s[1],w[1],a[1],b[1],w[0]);
```

```
fulladder f3(s[2],cout,a[2],b[2],w[1]);

endmodule
```

cu.v

```
module cu(out,in,clk);

input clk;
input [31:0]in;
output [31:0]out;

assign out[31:0]=in[31:0];
```

adder.v

```
module fulladder(s,cout,a,b,c);

input a,b,c;
output s,cout;
assign s=a^b^c;
assign cout=(a&b) | (b&c) | (c&a);
endmodule
```

controlunit.v (top module)

```
/*
Initializing BRAM in .coe file
*/
module pe_cu(a,d,clk,ena,wea);
output [7:0]d;
input clk;
input ena;
input wea;
input [31:0]a;
wire [255:0]r,out,m;
wire [31:0]out1;

//BRAM(clk,ena,wea,addra,dina,douta);
wire [31:0]k[7:0];

wire cout;
wire [2:0]s;
wire [2:0]s1;
wire [2:0]s2;
```

```
wire [2:0]s3;
wire [2:0]s4;
wire [2:0]s5;
wire [2:0]s6;


blk_mem_gen_0 y1(clk,ena,wea,var1,a,k[0]);
rca3 z1(s,cout,var1);

blk_mem_gen_0 y2(clk,ena,wea,s,a,k[1]);
rca3 z2(s1,cout,s);

blk_mem_gen_0 y3(clk,ena,wea,s1,a,k[2]);
rca3 z3(s2,cout,s1 );

blk_mem_gen_0 y4(clk,ena,wea,s2,a,k[3]);
rca3 z4(s3,cout,s2);

blk_mem_gen_0 y5(clk,ena,wea,s3,a,k[4]);
rca3 z5(s4,cout,s3);
```

```
blk_mem_gen_0 y6(clk,ena,wea,s4,a,k[5]);
rca3 z6(s5,cout,s4);

blk_mem_gen_0 y7(clk,ena,wea,s5,a,k[6]);
rca3 z7(s6,cout,s5);

blk_mem_gen_0 y8(clk,ena,wea,s6,a,k[7]);

//CU
cu fr0(m[31:0],k[0],clk);
cu fr1(m[63:32],k[1],clk);
cu fr2(m[95:64],k[2],clk);
cu fr3(m[127:96],k[3],clk);
cu fr4(m[159:128],k[4],clk);
cu fr5(m[191:160],k[5],clk);
cu fr6(m[223:192],k[6],clk);
cu fr7(m[255:224],k[7],clk);

wire v;
```

```
pe_256 p1(m[255:0],d[7:0],v,clk);
endmodule
```

Output:

Mapped the output on the FPGA Zedboard.

Experiment no. 10:

Aim: To synthesize the TCAM using SRAM and map it onto the FPGA.

Code:

controlunit.v (top module)

```
module pe_cu(a,var,d,clk,ena,wea);
output [7:0]d;

input clk;
input ena;
input wea;

wire [255:0]r;
wire [143:0]m;
wire cout;
```

```
wire [1:0]s;
wire [1:0]s1;
wire [1:0]s2;
wire [1:0]s3;

input [1:0]var;
input [35:0]a;
wire [35:0]k[3:0];
blk_mem_gen_0 y1(clk,ena,wea,var,a,k[0]);
rca2 z1(s,cout,var);

blk_mem_gen_0 y2(clk,ena,wea,s,a,k[1]);
rca2 z2(s1,cout,s);

blk_mem_gen_0 y3(clk,ena,wea,s1,a,k[2]);
rca2 z3(s2,cout,s1);

blk_mem_gen_0 y4(clk,ena,wea,s2,a,k[3]);
//CU
cu fr0(m[35:0],k[0],clk);
cu fr1(m[71:36],k[1],clk);
cu fr2(m[107:72],k[2],clk);
```

```
cu fr3(m[143:108],k[3],clk);
//TCAM
tcam t1(m,r);
wire v;
pe_256 p1(r[255:0],d[7:0],v,clk);

endmodule
```

tcam.v

```
module tcam(W,N);
input [143:0]W;
output [255:0]N;

wire [255:0]d1[3:0];
assign d1[0][255:0] = 256'd120;
assign d1[1][255:0] = 256'd121;
assign d1[2][255:0] = 256'd122;
assign d1[3][255:0] = 256'd123;
wire [255:0]d2[3:0];
assign d2[0][255:0] = 256'd120;
assign d2[1][255:0] = 256'd121;
assign d2[2][255:0] = 256'd122;
assign d2[3][255:0] = 256'd123;
```

```
wire [255:0]d3[3:0];
assign d3[0][255:0] = 256'd120;
assign d3[1][255:0] = 256'd121;
assign d3[2][255:0] = 256'd122;
assign d3[3][255:0] = 256'd123;
wire [255:0]d4[3:0];
assign d4[0][255:0] = 256'd120;
assign d4[1][255:0] = 256'd121;
assign d4[2][255:0] = 256'd122;
assign d4[3][255:0] = 256'd123;
wire [255:0]d5[3:0];
assign d5[0][255:0] = 256'd120;
assign d5[1][255:0] = 256'd121;
assign d5[2][255:0] = 256'd122;
assign d5[3][255:0] = 256'd123;
wire [255:0]d6[3:0];
assign d6[0][255:0] = 256'd120;
assign d6[1][255:0] = 256'd121;
assign d6[2][255:0] = 256'd122;
assign d6[3][255:0] = 256'd123;
wire [255:0]d7[3:0];
assign d7[0][255:0] = 256'd120;
```

```
assign d7[1][255:0] = 256'd121;
assign d7[2][255:0] = 256'd122;
assign d7[3][255:0] = 256'd123;
wire [255:0]d8[3:0];
assign d8[0][255:0] = 256'd120;
assign d8[1][255:0] = 256'd121;
assign d8[2][255:0] = 256'd122;
assign d8[3][255:0] = 256'd123;
wire [255:0]d9[3:0];
assign d9[0][255:0] = 256'd120;
assign d9[1][255:0] = 256'd121;
assign d9[2][255:0] = 256'd122;
assign d9[3][255:0] = 256'd123;
wire [255:0]d10[3:0];
assign d10[0][255:0] = 256'd120;
assign d10[1][255:0] = 256'd121;
assign d10[2][255:0] = 256'd122;
assign d10[3][255:0] = 256'd123;
wire [255:0]d11[3:0];
assign d11[0][255:0] = 256'd120;
assign d11[1][255:0] = 256'd121;
assign d11[2][255:0] = 256'd122;
```

```
assign d11[3][255:0] = 256'd123;
wire [255:0]d12[3:0];
assign d12[0][255:0] = 256'd120;
assign d12[1][255:0] = 256'd121;
assign d12[2][255:0] = 256'd122;
assign d12[3][255:0] = 256'd123;
wire [255:0]d13[3:0];
assign d13[0][255:0] = 256'd120;
assign d13[1][255:0] = 256'd121;
assign d13[2][255:0] = 256'd122;
assign d13[3][255:0] = 256'd123;
wire [255:0]d14[3:0];
assign d14[0][255:0] = 256'd120;
assign d14[1][255:0] = 256'd121;
assign d14[2][255:0] = 256'd122;
assign d14[3][255:0] = 256'd123;
wire [255:0]d15[3:0];
assign d15[0][255:0] = 256'd120;
assign d15[1][255:0] = 256'd121;
assign d15[2][255:0] = 256'd122;
assign d15[3][255:0] = 256'd123;
wire [255:0]d16[3:0];
```

```
assign d16[0][255:0] = 256'd120;
assign d16[1][255:0] = 256'd121;
assign d16[2][255:0] = 256'd122;
assign d16[3][255:0] = 256'd123;
wire [255:0]d17[3:0];
assign d17[0][255:0] = 256'd120;
assign d17[1][255:0] = 256'd121;
assign d17[2][255:0] = 256'd122;
assign d17[3][255:0] = 256'd123;
wire [255:0]d18[3:0];
assign d18[0][255:0] = 256'd120;
assign d18[1][255:0] = 256'd121;
assign d18[2][255:0] = 256'd122;
assign d18[3][255:0] = 256'd123;
wire [255:0]d19[3:0];
assign d19[0][255:0] = 256'd120;
assign d19[1][255:0] = 256'd121;
assign d19[2][255:0] = 256'd122;
assign d19[3][255:0] = 256'd123;
wire [255:0]d20[3:0];
assign d20[0][255:0] = 256'd120;
assign d20[1][255:0] = 256'd121;
```

```
assign d20[2][255:0] = 256'd122;
assign d20[3][255:0] = 256'd123;
wire [255:0]d21[3:0];
assign d21[0][255:0] = 256'd120;
assign d21[1][255:0] = 256'd121;
assign d21[2][255:0] = 256'd122;
assign d21[3][255:0] = 256'd123;
wire [255:0]d22[3:0];
assign d22[0][255:0] = 256'd120;
assign d22[1][255:0] = 256'd121;
assign d22[2][255:0] = 256'd122;
assign d22[3][255:0] = 256'd123;
wire [255:0]d23[3:0];
assign d23[0][255:0] = 256'd120;
assign d23[1][255:0] = 256'd121;
assign d23[2][255:0] = 256'd122;
assign d23[3][255:0] = 256'd123;
wire [255:0]d24[3:0];
assign d24[0][255:0] = 256'd120;
assign d24[1][255:0] = 256'd121;
assign d24[2][255:0] = 256'd122;
assign d24[3][255:0] = 256'd123;
```

```
wire [255:0]d25[3:0];
assign d25[0][255:0] = 256'd120;
assign d25[1][255:0] = 256'd121;
assign d25[2][255:0] = 256'd122;
assign d25[3][255:0] = 256'd123;
wire [255:0]d26[3:0];
assign d26[0][255:0] = 256'd120;
assign d26[1][255:0] = 256'd121;
assign d26[2][255:0] = 256'd122;
assign d26[3][255:0] = 256'd123;
wire [255:0]d27[3:0];
assign d27[0][255:0] = 256'd120;
assign d27[1][255:0] = 256'd121;
assign d27[2][255:0] = 256'd122;
assign d27[3][255:0] = 256'd123;
wire [255:0]d28[3:0];
assign d28[0][255:0] = 256'd120;
assign d28[1][255:0] = 256'd121;
assign d28[2][255:0] = 256'd122;
assign d28[3][255:0] = 256'd123;
wire [255:0]d29[3:0];
assign d29[0][255:0] = 256'd120;
```

```
assign d29[1][255:0] = 256'd121;
assign d29[2][255:0] = 256'd122;
assign d29[3][255:0] = 256'd123;
wire [255:0]d30[3:0];
assign d30[0][255:0] = 256'd120;
assign d30[1][255:0] = 256'd121;
assign d30[2][255:0] = 256'd122;
assign d30[3][255:0] = 256'd123;
wire [255:0]d31[3:0];
assign d31[0][255:0] = 256'd120;
assign d31[1][255:0] = 256'd121;
assign d31[2][255:0] = 256'd122;
assign d31[3][255:0] = 256'd123;
wire [255:0]d32[3:0];
assign d32[0][255:0] = 256'd120;
assign d32[1][255:0] = 256'd121;
assign d32[2][255:0] = 256'd122;
assign d32[3][255:0] = 256'd123;
wire [255:0]d33[3:0];
assign d33[0][255:0] = 256'd120;
assign d33[1][255:0] = 256'd121;
assign d33[2][255:0] = 256'd122;
```

```
assign d33[3][255:0] = 256'd123;
wire [255:0]d34[3:0];
assign d34[0][255:0] = 256'd120;
assign d34[1][255:0] = 256'd121;
assign d34[2][255:0] = 256'd122;
assign d34[3][255:0] = 256'd123;
wire [255:0]d35[3:0];
assign d35[0][255:0] = 256'd120;
assign d35[1][255:0] = 256'd121;
assign d35[2][255:0] = 256'd122;
assign d35[3][255:0] = 256'd123;
wire [255:0]d36[3:0];
assign d36[0][255:0] = 256'd120;
assign d36[1][255:0] = 256'd121;
assign d36[2][255:0] = 256'd122;
assign d36[3][255:0] = 256'd123;
wire [255:0]d37[3:0];
assign d37[0][255:0] = 256'd120;
assign d37[1][255:0] = 256'd121;
assign d37[2][255:0] = 256'd122;
assign d37[3][255:0] = 256'd123;
wire [255:0]d38[3:0];
```

```
assign d38[0][255:0] = 256'd120;
assign d38[1][255:0] = 256'd121;
assign d38[2][255:0] = 256'd122;
assign d38[3][255:0] = 256'd123;
wire [255:0]d39[3:0];
assign d39[0][255:0] = 256'd120;
assign d39[1][255:0] = 256'd121;
assign d39[2][255:0] = 256'd122;
assign d39[3][255:0] = 256'd123;
wire [255:0]d40[3:0];
assign d40[0][255:0] = 256'd120;
assign d40[1][255:0] = 256'd121;
assign d40[2][255:0] = 256'd122;
assign d40[3][255:0] = 256'd123;
wire [255:0]d41[3:0];
assign d41[0][255:0] = 256'd120;
assign d41[1][255:0] = 256'd121;
assign d41[2][255:0] = 256'd122;
assign d41[3][255:0] = 256'd123;
wire [255:0]d42[3:0];
assign d42[0][255:0] = 256'd120;
assign d42[1][255:0] = 256'd121;
```

```
assign d42[2][255:0] = 256'd122;
assign d42[3][255:0] = 256'd123;
wire [255:0]d43[3:0];
assign d43[0][255:0] = 256'd120;
assign d43[1][255:0] = 256'd121;
assign d43[2][255:0] = 256'd122;
assign d43[3][255:0] = 256'd123;
wire [255:0]d44[3:0];
assign d44[0][255:0] = 256'd120;
assign d44[1][255:0] = 256'd121;
assign d44[2][255:0] = 256'd122;
assign d44[3][255:0] = 256'd123;
wire [255:0]d45[3:0];
assign d45[0][255:0] = 256'd120;
assign d45[1][255:0] = 256'd121;
assign d45[2][255:0] = 256'd122;
assign d45[3][255:0] = 256'd123;
wire [255:0]d46[3:0];
assign d46[0][255:0] = 256'd120;
assign d46[1][255:0] = 256'd121;
assign d46[2][255:0] = 256'd122;
assign d46[3][255:0] = 256'd123;
```

```
wire [255:0]d47[3:0];
assign d47[0][255:0] = 256'd120;
assign d47[1][255:0] = 256'd121;
assign d47[2][255:0] = 256'd122;
assign d47[3][255:0] = 256'd123;
wire [255:0]d48[3:0];
assign d48[0][255:0] = 256'd120;
assign d48[1][255:0] = 256'd121;
assign d48[2][255:0] = 256'd122;
assign d48[3][255:0] = 256'd123;
wire [255:0]d49[3:0];
assign d49[0][255:0] = 256'd120;
assign d49[1][255:0] = 256'd121;
assign d49[2][255:0] = 256'd122;
assign d49[3][255:0] = 256'd123;
wire [255:0]d50[3:0];
assign d50[0][255:0] = 256'd120;
assign d50[1][255:0] = 256'd121;
assign d50[2][255:0] = 256'd122;
assign d50[3][255:0] = 256'd123;
wire [255:0]d51[3:0];
assign d51[0][255:0] = 256'd120;
```

```
assign d51[1][255:0] = 256'd121;
assign d51[2][255:0] = 256'd122;
assign d51[3][255:0] = 256'd123;
wire [255:0]d52[3:0];
assign d52[0][255:0] = 256'd120;
assign d52[1][255:0] = 256'd121;
assign d52[2][255:0] = 256'd122;
assign d52[3][255:0] = 256'd123;
wire [255:0]d53[3:0];
assign d53[0][255:0] = 256'd120;
assign d53[1][255:0] = 256'd121;
assign d53[2][255:0] = 256'd122;
assign d53[3][255:0] = 256'd123;
wire [255:0]d54[3:0];
assign d54[0][255:0] = 256'd120;
assign d54[1][255:0] = 256'd121;
assign d54[2][255:0] = 256'd122;
assign d54[3][255:0] = 256'd123;
wire [255:0]d55[3:0];
assign d55[0][255:0] = 256'd120;
assign d55[1][255:0] = 256'd121;
assign d55[2][255:0] = 256'd122;
```

```
assign d55[3][255:0] = 256'd123;
wire [255:0]d56[3:0];
assign d56[0][255:0] = 256'd120;
assign d56[1][255:0] = 256'd121;
assign d56[2][255:0] = 256'd122;
assign d56[3][255:0] = 256'd123;
wire [255:0]d57[3:0];
assign d57[0][255:0] = 256'd120;
assign d57[1][255:0] = 256'd121;
assign d57[2][255:0] = 256'd122;
assign d57[3][255:0] = 256'd123;
wire [255:0]d58[3:0];
assign d58[0][255:0] = 256'd120;
assign d58[1][255:0] = 256'd121;
assign d58[2][255:0] = 256'd122;
assign d58[3][255:0] = 256'd123;
wire [255:0]d59[3:0];
assign d59[0][255:0] = 256'd120;
assign d59[1][255:0] = 256'd121;
assign d59[2][255:0] = 256'd122;
assign d59[3][255:0] = 256'd123;
wire [255:0]d60[3:0];
```

```
assign d60[0][255:0] = 256'd120;
assign d60[1][255:0] = 256'd121;
assign d60[2][255:0] = 256'd122;
assign d60[3][255:0] = 256'd123;
wire [255:0]d61[3:0];
assign d61[0][255:0] = 256'd120;
assign d61[1][255:0] = 256'd121;
assign d61[2][255:0] = 256'd122;
assign d61[3][255:0] = 256'd123;
wire [255:0]d62[3:0];
assign d62[0][255:0] = 256'd120;
assign d62[1][255:0] = 256'd121;
assign d62[2][255:0] = 256'd122;
assign d62[3][255:0] = 256'd123;
wire [255:0]d63[3:0];
assign d63[0][255:0] = 256'd120;
assign d63[1][255:0] = 256'd121;
assign d63[2][255:0] = 256'd122;
assign d63[3][255:0] = 256'd123;
wire [255:0]d64[3:0];
assign d64[0][255:0] = 256'd120;
assign d64[1][255:0] = 256'd121;
```

```
assign d64[2][255:0] = 256'd122;
assign d64[3][255:0] = 256'd123;
wire [255:0]d65[3:0];
assign d65[0][255:0] = 256'd120;
assign d65[1][255:0] = 256'd121;
assign d65[2][255:0] = 256'd122;
assign d65[3][255:0] = 256'd123;
wire [255:0]d66[3:0];
assign d66[0][255:0] = 256'd120;
assign d66[1][255:0] = 256'd121;
assign d66[2][255:0] = 256'd122;
assign d66[3][255:0] = 256'd123;
wire [255:0]d67[3:0];
assign d67[0][255:0] = 256'd120;
assign d67[1][255:0] = 256'd121;
assign d67[2][255:0] = 256'd122;
assign d67[3][255:0] = 256'd123;
wire [255:0]d68[3:0];
assign d68[0][255:0] = 256'd120;
assign d68[1][255:0] = 256'd121;
assign d68[2][255:0] = 256'd122;
assign d68[3][255:0] = 256'd123;
```

```
wire [255:0]d69[3:0];
assign d69[0][255:0] = 256'd120;
assign d69[1][255:0] = 256'd121;
assign d69[2][255:0] = 256'd122;
assign d69[3][255:0] = 256'd123;
wire [255:0]d70[3:0];
assign d70[0][255:0] = 256'd120;
assign d70[1][255:0] = 256'd121;
assign d70[2][255:0] = 256'd122;
assign d70[3][255:0] = 256'd123;
wire [255:0]d71[3:0];
assign d71[0][255:0] = 256'd120;
assign d71[1][255:0] = 256'd121;
assign d71[2][255:0] = 256'd122;
assign d71[3][255:0] = 256'd123;
wire [255:0]d72[3:0];
assign d72[0][255:0] = 256'd120;
assign d72[1][255:0] = 256'd121;
assign d72[2][255:0] = 256'd122;
assign d72[3][255:0] = 256'd123;

wire [255:0]out1;
```

```
wire [255:0]out2;
wire [255:0]out3;
wire [255:0]out4;
wire [255:0]out5;
wire [255:0]out6;
wire [255:0]out7;
wire [255:0]out8;
wire [255:0]out9;
wire [255:0]out10;
wire [255:0]out11;
wire [255:0]out12;
wire [255:0]out13;
wire [255:0]out14;
wire [255:0]out15;
wire [255:0]out16;
wire [255:0]out17;
wire [255:0]out18;
wire [255:0]out19;
wire [255:0]out20;
wire [255:0]out21;
wire [255:0]out22;
wire [255:0]out23;
```

```
wire [255:0]out24;
wire [255:0]out25;
wire [255:0]out26;
wire [255:0]out27;
wire [255:0]out28;
wire [255:0]out29;
wire [255:0]out30;
wire [255:0]out31;
wire [255:0]out32;
wire [255:0]out33;
wire [255:0]out34;
wire [255:0]out35;
wire [255:0]out36;
wire [255:0]out37;
wire [255:0]out38;
wire [255:0]out39;
wire [255:0]out40;
wire [255:0]out41;
wire [255:0]out42;
wire [255:0]out43;
wire [255:0]out44;
wire [255:0]out45;
```

```
wire [255:0]out46;
wire [255:0]out47;
wire [255:0]out48;
wire [255:0]out49;
wire [255:0]out50;
wire [255:0]out51;
wire [255:0]out52;
wire [255:0]out53;
wire [255:0]out54;
wire [255:0]out55;
wire [255:0]out56;
wire [255:0]out57;
wire [255:0]out58;
wire [255:0]out59;
wire [255:0]out60;
wire [255:0]out61;
wire [255:0]out62;
wire [255:0]out63;
wire [255:0]out64;
wire [255:0]out65;
wire [255:0]out66;
wire [255:0]out67;
```

```
wire [255:0]out68;
wire [255:0]out69;
wire [255:0]out70;
wire [255:0]out71;
wire [255:0]out72;

integer i1;
mux m1(W[1:0],d1[0],d1[1],d1[2],d1[3],out1);
//assign N[255:0] = out1[255:0];
mux m2(W[3:2],d2[0],d2[1],d2[2],d2[3],out2);
mux m3(W[5:4],d3[0],d3[1],d3[2],d3[3],out3);
mux m4(W[7:6],d4[0],d4[1],d4[2],d4[3],out4);
mux m5(W[9:8],d5[0],d5[1],d5[2],d5[3],out5);
mux m6(W[11:10],d6[0],d6[1],d6[2],d6[3],out6);
mux m7(W[13:12],d7[0],d7[1],d7[2],d7[3],out7);
mux m8(W[15:14],d8[0],d8[1],d8[2],d8[3],out8);
mux m9(W[17:16],d9[0],d9[1],d9[2],d9[3],out9);
mux m10(W[19:18],d10[0],d10[1],d10[2],d10[3],out10);
mux m11(W[21:20],d11[0],d11[1],d11[2],d11[3],out11);
mux m12(W[23:22],d12[0],d12[1],d12[2],d12[3],out12);
mux m13(W[25:24],d13[0],d13[1],d13[2],d13[3],out13);
mux m14(W[27:26],d14[0],d14[1],d14[2],d14[3],out14);
```

```
mux m15(W[29:28],d15[0],d15[1],d15[2],d15[3],out15);
mux m16(W[31:30],d16[0],d16[1],d16[2],d16[3],out16);
mux m17(W[33:32],d17[0],d17[1],d17[2],d17[3],out17);
mux m18(W[35:34],d18[0],d18[1],d18[2],d18[3],out18);
mux m19(W[37:36],d19[0],d19[1],d19[2],d19[3],out19);
mux m20(W[39:38],d20[0],d20[1],d20[2],d20[3],out20);
mux m21(W[41:40],d21[0],d21[1],d21[2],d21[3],out21);
mux m22(W[43:42],d22[0],d22[1],d22[2],d22[3],out22);
mux m23(W[45:44],d23[0],d23[1],d23[2],d23[3],out23);
mux m24(W[47:46],d24[0],d24[1],d24[2],d24[3],out24);
mux m25(W[49:48],d25[0],d25[1],d25[2],d25[3],out25);
mux m26(W[51:50],d26[0],d26[1],d26[2],d26[3],out26);
mux m27(W[53:52],d27[0],d27[1],d27[2],d27[3],out27);
mux m28(W[55:54],d28[0],d28[1],d28[2],d28[3],out28);
mux m29(W[57:56],d29[0],d29[1],d29[2],d29[3],out29);
mux m30(W[59:58],d30[0],d30[1],d30[2],d30[3],out30);
mux m31(W[61:60],d31[0],d31[1],d31[2],d31[3],out31);
mux m32(W[63:62],d32[0],d32[1],d32[2],d32[3],out32);
mux m33(W[65:64],d33[0],d33[1],d33[2],d33[3],out33);
mux m34(W[67:66],d34[0],d34[1],d34[2],d34[3],out34);
mux m35(W[69:68],d35[0],d35[1],d35[2],d35[3],out35);
mux m36(W[71:70],d36[0],d36[1],d36[2],d36[3],out36);
```

```
mux m37(W[73:72],d37[0],d37[1],d37[2],d37[3],out37);
mux m38(W[75:74],d38[0],d38[1],d38[2],d38[3],out38);
mux m39(W[77:76],d39[0],d39[1],d39[2],d39[3],out39);
mux m40(W[79:78],d40[0],d40[1],d40[2],d40[3],out40);
mux m41(W[81:80],d41[0],d41[1],d41[2],d41[3],out41);
mux m42(W[83:82],d42[0],d42[1],d42[2],d42[3],out42);
mux m43(W[85:84],d43[0],d43[1],d43[2],d43[3],out43);
mux m44(W[87:86],d44[0],d44[1],d44[2],d44[3],out44);
mux m45(W[89:88],d45[0],d45[1],d45[2],d45[3],out45);
mux m46(W[91:90],d46[0],d46[1],d46[2],d46[3],out46);
mux m47(W[93:92],d47[0],d47[1],d47[2],d47[3],out47);
mux m48(W[95:94],d48[0],d48[1],d48[2],d48[3],out48);
mux m49(W[97:96],d49[0],d49[1],d49[2],d49[3],out49);
mux m50(W[99:98],d50[0],d50[1],d50[2],d50[3],out50);
mux m51(W[101:100],d51[0],d51[1],d51[2],d51[3],out51);
mux m52(W[103:102],d52[0],d52[1],d52[2],d52[3],out52);
mux m53(W[105:104],d53[0],d53[1],d53[2],d53[3],out53);
mux m54(W[107:106],d54[0],d54[1],d54[2],d54[3],out54);
mux m55(W[109:108],d55[0],d55[1],d55[2],d55[3],out55);
mux m56(W[111:110],d56[0],d56[1],d56[2],d56[3],out56);
mux m57(W[113:112],d57[0],d57[1],d57[2],d57[3],out57);
mux m58(W[115:114],d58[0],d58[1],d58[2],d58[3],out58);
```

```
mux m59(W[117:116],d59[0],d59[1],d59[2],d59[3],out59);
mux m60(W[119:118],d60[0],d60[1],d60[2],d60[3],out60);
mux m61(W[121:120],d61[0],d61[1],d61[2],d61[3],out61);
mux m62(W[123:122],d62[0],d62[1],d62[2],d62[3],out62);
mux m63(W[125:124],d63[0],d63[1],d63[2],d63[3],out63);
mux m64(W[127:126],d64[0],d64[1],d64[2],d64[3],out64);
mux m65(W[129:128],d65[0],d65[1],d65[2],d65[3],out65);
mux m66(W[131:130],d66[0],d66[1],d66[2],d66[3],out66);
mux m67(W[133:132],d67[0],d67[1],d67[2],d67[3],out67);
mux m68(W[135:134],d68[0],d68[1],d68[2],d68[3],out68);
mux m69(W[137:136],d69[0],d69[1],d69[2],d69[3],out69);
mux m70(W[139:138],d70[0],d70[1],d70[2],d70[3],out70);
mux m71(W[141:140],d71[0],d71[1],d71[2],d71[3],out71);
mux m72(W[143:142],d72[0],d72[1],d72[2],d72[3],out72);

wire [255:0]and1;
wire [255:0]and2;
assign and1 = out1 & out2 & out3 & out4 & out5 & out6 &
out7 & out8 & out9 & out10 &out11 & out12 & out13 &
out14 & out15 & out16 & out17 & out18 & out19 & out20 &
out21 & out22 & out23 & out24 & out25 & out26 & out27 &
out28 & out29 & out30 & out31 & out32 & out33 & out34 &
out35 & out36 & out37 & out38 & out39 & out40 & out41 &
```

```
out42 & out43 & out44 & out45 & out46 & out47 & out48 &
out49 & out50 & out51 & out52 & out53 & out54 & out55 &
out56 & out57 & out58 & out59 & out60 & out61 & out62 &
out63 & out64 & out65 & out66 & out67 & out68 & out69 &
out70 & out71 & out72;

wire u;

pe_256 mn(and1,and2,u);

endmodule
```

priority256.v

```
module pe_256(D, result,l4,clk);

input clk;

input[255:0]D;

wire[3:0]w[84:0];// stage outputs

wire [63:0]o;//or output 1st stage

wire [16:0]o1;

wire [3:0]o2;

wire [63:0]l1;

wire [15:0]l2;

wire [3:0]l3;

output l4;

wire [3:0]cout[127:0];

wire [3:0]fin[63:0];
```

```
wire [255:0]outfi;
real result1;
integer n,p;
//real result1;
output [7:0]result;
//output [7:0]result1;

//stage1
priority4bit a(D[3:0],w[0],l1[0]);
priority4bit a1(D[7:4],w[1],l1[1]);
priority4bit  a2(D[11:8] ,w[2],l1[2]);
priority4bit  a3(D[15:12],w[3],l1[3]);
priority4bit  a4(D[19:16],w[4],l1[4]);
priority4bit  a5(D[23:20],w[5],l1[5]);
priority4bit  a6(D[27:24],w[6],l1[6]);
priority4bit  a7(D[31:28],w[7],l1[7]);
priority4bit  a8(D[35:32],w[8],l1[8]);
priority4bit  a9(D[39:36],w[9],l1[9]);
priority4bit  a10(D[43:40],w[10],l1[10]);
priority4bit  a11(D[47:44],w[11],l1[11]);
priority4bit  a12(D[51:48],w[12],l1[12]);
priority4bit  a13(D[55:52],w[13],l1[13]);
```

```
priority4bit  a14(D[59:56],w[14],l1[14]);
priority4bit  a15(D[63:60],w[15],l1[15]);
priority4bit  a16(D[67:64],w[16],l1[16]);
priority4bit  a17(D[71:68],w[17],l1[17]);
priority4bit  a18(D[75:72],w[18],l1[18]);
priority4bit  a19(D[79:76],w[19],l1[19]);
priority4bit  a20(D[83:80],w[20],l1[20]);
priority4bit  a21(D[87:84],w[21],l1[21]);
priority4bit  a22(D[91:88],w[22],l1[22]);
priority4bit  a23(D[95:92],w[23],l1[23]);
priority4bit  a24(D[99:96],w[24],l1[24]);
priority4bit  a25(D[103:100],w[25],l1[25]);
priority4bit  a26(D[107:104],w[26],l1[26]);
priority4bit  a27(D[111:108],w[27],l1[27]);
priority4bit  a28(D[115:112],w[28],l1[28]);
priority4bit  a29(D[119:116],w[29],l1[29]);
priority4bit  a30(D[123:120],w[30],l1[30]);
priority4bit  a31(D[127:124],w[31],l1[31]);
priority4bit  a32(D[131:128],w[32],l1[32]);
priority4bit  a33(D[135:132],w[33],l1[33]);
priority4bit  a34(D[139:136],w[34],l1[34]);
priority4bit  a35(D[143:140],w[35],l1[35]);
```

```
priority4bit  a36(D[147:144],w[36],l1[36]);
priority4bit  a37(D[151:148],w[37],l1[37]);
priority4bit  a38(D[155:152],w[38],l1[38]);
priority4bit  a39(D[159:156],w[39],l1[39]);
priority4bit  a40(D[163:160],w[40],l1[40]);
priority4bit  a41(D[167:164],w[41],l1[41]);
priority4bit  a42(D[171:168],w[42],l1[42]);
priority4bit  a43(D[175:172],w[43],l1[43]);
priority4bit  a44(D[179:176],w[44],l1[44]);
priority4bit  a45(D[183:180],w[45],l1[45]);
priority4bit  a46(D[187:184],w[46],l1[46]);
priority4bit  a47(D[191:188],w[47],l1[47]);
priority4bit  a48(D[195:192],w[48],l1[48]);
priority4bit  a49(D[199:196],w[49],l1[49]);
priority4bit  a50(D[203:200],w[50],l1[50]);
priority4bit  a51(D[207:204],w[51],l1[51]);
priority4bit  a52(D[211:208],w[52],l1[52]);
priority4bit  a53(D[215:212],w[53],l1[53]);
priority4bit  a54(D[219:216],w[54],l1[54]);
priority4bit  a55(D[223:220],w[55],l1[55]);
priority4bit  a56(D[227:224],w[56],l1[56]);
priority4bit  a57(D[231:228],w[57],l1[57]);
```

```
priority4bit  a58(D[235:232],w[58],l1[58]);
priority4bit  a59(D[239:236],w[59],l1[59]);
priority4bit  a60(D[243:240],w[60],l1[60]);
priority4bit  a61(D[247:244],w[61],l1[61]);
priority4bit  a62(D[251:248],w[62],l1[62]);
priority4bit  a63(D[255:252],w[63],l1[63]);

//stage 1 or
assign o[   0   ]=w[    0   ][0]|w[ 0   ][1]|w[ 0   ][2]|w[ 0   ][3];
assign o[  1      ]=w[1      ][0]|w[     1      ][1]|w[     1      ][2]|w[
      1      ][3];
assign o[  2      ]=w[2      ][0]|w[     2      ][1]|w[     2      ][2]|w[
      2      ][3];
assign o[  3      ]=w[3      ][0]|w[     3      ][1]|w[     3      ][2]|w[
      3      ][3];
assign o[  4      ]=w[4      ][0]|w[     4      ][1]|w[     4      ][2]|w[
      4      ][3];
assign o[  5      ]=w[5      ][0]|w[     5      ][1]|w[     5      ][2]|w[
      5      ][3];
assign o[  6      ]=w[6      ][0]|w[     6      ][1]|w[     6      ][2]|w[
      6      ][3];
assign o[  7      ]=w[7      ][0]|w[     7      ][1]|w[     7      ][2]|w[
      7      ][3];
```

```
assign o[8]=w[8][0]|w[8][1]|w[8][2]|w[8][3];
assign o[9]=w[9][0]|w[9][1]|w[9][2]|w[9][3];
assign o[10]=w[10][0]|w[10][1]|w[10][2]|w[10][3];
assign o[11]=w[11][0]|w[11][1]|w[11][2]|w[11][3];
assign o[12]=w[12][0]|w[12][1]|w[12][2]|w[12][3];
assign o[13]=w[13][0]|w[13][1]|w[13][2]|w[13][3];
assign o[14]=w[14][0]|w[14][1]|w[14][2]|w[14][3];
assign o[15]=w[15][0]|w[15][1]|w[15][2]|w[15][3];
assign o[16]=w[16][0]|w[16][1]|w[16][2]|w[16][3];
assign o[17]=w[17][0]|w[17][1]|w[17][2]|w[17][3];
assign o[18]=w[18][0]|w[18][1]|w[18][2]|w[18][3];
assign o[19]=w[19][0]|w[19][1]|w[19][2]|w[19][3];
```

```
assign o[20]=w[20][0]|w[20][1]|w[20][2]|w[20][3];
assign o[21]=w[21][0]|w[21][1]|w[21][2]|w[21][3];
assign o[22]=w[22][0]|w[22][1]|w[22][2]|w[22][3];
assign o[23]=w[23][0]|w[23][1]|w[23][2]|w[23][3];
assign o[24]=w[24][0]|w[24][1]|w[24][2]|w[24][3];
assign o[25]=w[25][0]|w[25][1]|w[25][2]|w[25][3];
assign o[26]=w[26][0]|w[26][1]|w[26][2]|w[26][3];
assign o[27]=w[27][0]|w[27][1]|w[27][2]|w[27][3];
assign o[28]=w[28][0]|w[28][1]|w[28][2]|w[28][3];
assign o[29]=w[29][0]|w[29][1]|w[29][2]|w[29][3];
assign o[30]=w[30][0]|w[30][1]|w[30][2]|w[30][3];
assign o[31]=w[31][0]|w[31][1]|w[31][2]|w[31][3];
```

```
assign o[32]=w[32][0]|w[32][1]|w[32][2]|w[32][3];
assign o[33]=w[33][0]|w[33][1]|w[33][2]|w[33][3];
assign o[34]=w[34][0]|w[34][1]|w[34][2]|w[34][3];
assign o[35]=w[35][0]|w[35][1]|w[35][2]|w[35][3];
assign o[36]=w[36][0]|w[36][1]|w[36][2]|w[36][3];
assign o[37]=w[37][0]|w[37][1]|w[37][2]|w[37][3];
assign o[38]=w[38][0]|w[38][1]|w[38][2]|w[38][3];
assign o[39]=w[39][0]|w[39][1]|w[39][2]|w[39][3];
assign o[40]=w[40][0]|w[40][1]|w[40][2]|w[40][3];
assign o[41]=w[41][0]|w[41][1]|w[41][2]|w[41][3];
assign o[42]=w[42][0]|w[42][1]|w[42][2]|w[42][3];
assign o[43]=w[43][0]|w[43][1]|w[43][2]|w[43][3];
```

```
assign o[44]=w[44][0]|w[44][1]|w[44][2]|w[44][3];
assign o[45]=w[45][0]|w[45][1]|w[45][2]|w[45][3];
assign o[46]=w[46][0]|w[46][1]|w[46][2]|w[46][3];
assign o[47]=w[47][0]|w[47][1]|w[47][2]|w[47][3];
assign o[48]=w[48][0]|w[48][1]|w[48][2]|w[48][3];
assign o[49]=w[49][0]|w[49][1]|w[49][2]|w[49][3];
assign o[50]=w[50][0]|w[50][1]|w[50][2]|w[50][3];
assign o[51]=w[51][0]|w[51][1]|w[51][2]|w[51][3];
assign o[52]=w[52][0]|w[52][1]|w[52][2]|w[52][3];
assign o[53]=w[53][0]|w[53][1]|w[53][2]|w[53][3];
assign o[54]=w[54][0]|w[54][1]|w[54][2]|w[54][3];
assign o[55]=w[55][0]|w[55][1]|w[55][2]|w[55][3];
```

```
assign o[ 56 ]=w[56 ][0]|w[ 56 ][1]|w[ 56 ][2]|w[
    56 ][3];
assign o[ 57 ]=w[57 ][0]|w[ 57 ][1]|w[ 57 ][2]|w[
    57 ][3];
assign o[ 58 ]=w[58 ][0]|w[ 58 ][1]|w[ 58 ][2]|w[
    58 ][3];
assign o[ 59 ]=w[59 ][0]|w[ 59 ][1]|w[ 59 ][2]|w[
    59 ][3];
assign o[ 60 ]=w[60 ][0]|w[ 60 ][1]|w[ 60 ][2]|w[
    60 ][3];
assign o[ 61 ]=w[61 ][0]|w[ 61 ][1]|w[ 61 ][2]|w[
    61 ][3];
assign o[ 62 ]=w[62 ][0]|w[ 62 ][1]|w[ 62 ][2]|w[
    62 ][3];
assign o[ 63 ]=w[63 ][0]|w[ 63 ][1]|w[ 63 ][2]|w[
    63 ][3];

//stage 2
priority4bit  b(o[3:0],w[64],l2[0]);
priority4bit  b1(o[7:4],w[65],l2[1]);
priority4bit  b2(o[11:8],w[66],l2[2]);
priority4bit  b3(o[15:12],w[67],l2[3]);
priority4bit  b4(o[19:16],w[68],l2[4]);
priority4bit  b5(o[23:20],w[69],l2[5]);
```

```
priority4bit     b6(o[27:24],w[70],l2[6]);
priority4bit     b7(o[31:28],w[71],l2[7]);
priority4bit     b8(o[35:32],w[72],l2[8]);
priority4bit     b9(o[39:36],w[73],l2[9]);
priority4bit     b10(o[43:40],w[74],l2[10]);
priority4bit     b11(o[47:44],w[75],l2[11]);
priority4bit     b12(o[51:48],w[76],l2[12]);
priority4bit     b13(o[55:52],w[77],l2[13]);
priority4bit     b14(o[59:56],w[78],l2[14]);
priority4bit     b15(o[63:60],w[79],l2[15]);

//ce
ce   e  (w[0    ],w[64  ][0 ],cout[0    ]);
ce   e1     (w[1 ],w[64     ][1    ],cout[1    ]);
ce   e2     (w[2 ],w[64     ][2    ],cout[2    ]);
ce   e3     (w[3 ],w[64     ][3    ],cout[3    ]);
ce   e4     (w[4 ],w[65     ][0    ],cout[4    ]);
ce   e5     (w[5 ],w[65     ][1    ],cout[5    ]);
ce   e6     (w[6 ],w[65     ][2    ],cout[6    ]);
ce   e7     (w[7 ],w[65     ][3    ],cout[7    ]);
ce   e8     (w[8 ],w[66     ][0    ],cout[8    ]);
ce   e9     (w[9 ],w[66     ][1    ],cout[9    ]);
```

```
ce e10 (w[10 ],w[66 ][2 ],cout[10 ]);
ce e11 (w[11 ],w[66 ][3 ],cout[11 ]);
ce e12 (w[12 ],w[67 ][0 ],cout[12 ]);
ce e13 (w[13 ],w[67 ][1 ],cout[13 ]);
ce e14 (w[14 ],w[67 ][2 ],cout[14 ]);
ce e15 (w[15 ],w[67 ][3 ],cout[15 ]);
ce e16 (w[16 ],w[68 ][0 ],cout[16 ]);
ce e17 (w[17 ],w[68 ][1 ],cout[17 ]);
ce e18 (w[18 ],w[68 ][2 ],cout[18 ]);
ce e19 (w[19 ],w[68 ][3 ],cout[19 ]);
ce e20 (w[20 ],w[69 ][0 ],cout[20 ]);
ce e21 (w[21 ],w[69 ][1 ],cout[21 ]);
ce e22 (w[22 ],w[69 ][2 ],cout[22 ]);
ce e23 (w[23 ],w[69 ][3 ],cout[23 ]);
ce e24 (w[24 ],w[70 ][0 ],cout[24 ]);
ce e25 (w[25 ],w[70 ][1 ],cout[25 ]);
ce e26 (w[26 ],w[70 ][2 ],cout[26 ]);
ce e27 (w[27 ],w[70 ][3 ],cout[27 ]);
ce e28 (w[28 ],w[71 ][0 ],cout[28 ]);
ce e29 (w[29 ],w[71 ][1 ],cout[29 ]);
ce e30 (w[30 ],w[71 ][2 ],cout[30 ]);
ce e31 (w[31 ],w[71 ][3 ],cout[31 ]);
```

```
ce e32 (w[32 ],w[72 ][0 ],cout[32 ]);
ce e33 (w[33 ],w[72 ][1 ],cout[33 ]);
ce e34 (w[34 ],w[72 ][2 ],cout[34 ]);
ce e35 (w[35 ],w[72 ][3 ],cout[35 ]);
ce e36 (w[36 ],w[73 ][0 ],cout[36 ]);
ce e37 (w[37 ],w[73 ][1 ],cout[37 ]);
ce e38 (w[38 ],w[73 ][2 ],cout[38 ]);
ce e39 (w[39 ],w[73 ][3 ],cout[39 ]);
ce e40 (w[40 ],w[74 ][0 ],cout[40 ]);
ce e41 (w[41 ],w[74 ][1 ],cout[41 ]);
ce e42 (w[42 ],w[74 ][2 ],cout[42 ]);
ce e43 (w[43 ],w[74 ][3 ],cout[43 ]);
ce e44 (w[44 ],w[75 ][0 ],cout[44 ]);
ce e45 (w[45 ],w[75 ][1 ],cout[45 ]);
ce e46 (w[46 ],w[75 ][2 ],cout[46 ]);
ce e47 (w[47 ],w[75 ][3 ],cout[47 ]);
ce e48 (w[48 ],w[76 ][0 ],cout[48 ]);
ce e49 (w[49 ],w[76 ][1 ],cout[49 ]);
ce e50 (w[50 ],w[76 ][2 ],cout[50 ]);
ce e51 (w[51 ],w[76 ][3 ],cout[51 ]);
ce e52 (w[52 ],w[77 ][0 ],cout[52 ]);
ce e53 (w[53 ],w[77 ][1 ],cout[53 ]);
```

```
ce e54 (w[54 ],w[77 ][2 ],cout[54 ]);
ce e55 (w[55 ],w[77 ][3 ],cout[55 ]);
ce e56 (w[56 ],w[78 ][0 ],cout[56 ]);
ce e57 (w[57 ],w[78 ][1 ],cout[57 ]);
ce e58 (w[58 ],w[78 ][2 ],cout[58 ]);
ce e59 (w[59 ],w[78 ][3 ],cout[59 ]);
ce e60 (w[60 ],w[79 ][0 ],cout[60 ]);
ce e61 (w[61 ],w[79 ][1 ],cout[61 ]);
ce e62 (w[62 ],w[79 ][2 ],cout[62 ]);
ce e63 (w[63 ],w[79 ][3 ],cout[63 ]);
```

//or stage2

```
assign o1[ 0 ]=w[64 ][0]|w[ 64 ][1]|w[ 64 ][2]|w[ 64 ][3];
assign o1[ 1 ]=w[65 ][0]|w[ 65 ][1]|w[ 65 ][2]|w[ 65 ][3];
assign o1[ 2 ]=w[66 ][0]|w[ 66 ][1]|w[ 66 ][2]|w[ 66 ][3];
assign o1[ 3 ]=w[67 ][0]|w[ 67 ][1]|w[ 67 ][2]|w[ 67 ][3];
```

```
assign o1[ 4 ]=w[68 ][0]|w[ 68 ][1]|w[ 68 ][2]|w[ 68 ][3];
assign o1[ 5 ]=w[69 ][0]|w[ 69 ][1]|w[ 69 ][2]|w[ 69 ][3];
assign o1[ 6 ]=w[70 ][0]|w[ 70 ][1]|w[ 70 ][2]|w[ 70 ][3];
assign o1[ 7 ]=w[71 ][0]|w[ 71 ][1]|w[ 71 ][2]|w[ 71 ][3];
assign o1[ 8 ]=w[72 ][0]|w[ 72 ][1]|w[ 72 ][2]|w[ 72 ][3];
assign o1[ 9 ]=w[73 ][0]|w[ 73 ][1]|w[ 73 ][2]|w[ 73 ][3];
assign o1[ 10 ]=w[74 ][0]|w[ 74 ][1]|w[ 74 ][2]|w[ 74 ][3];
assign o1[ 11 ]=w[75 ][0]|w[ 75 ][1]|w[ 75 ][2]|w[ 75 ][3];
assign o1[ 12 ]=w[76 ][0]|w[ 76 ][1]|w[ 76 ][2]|w[ 76 ][3];
assign o1[ 13 ]=w[77 ][0]|w[ 77 ][1]|w[ 77 ][2]|w[ 77 ][3];
assign o1[ 14 ]=w[78 ][0]|w[ 78 ][1]|w[ 78 ][2]|w[ 78 ][3];
assign o1[ 15 ]=w[ 79 ][0]|w[ 79 ][1]|w[ 79 ][2]|w[ 79 ][3];
```

```
//stage3
priority4bit    c(o1[3:0],w[80],l3[0]);
priority4bit     c1(o1[7:4],w[81],l3[1]);
priority4bit     c2(o1[11:8],w[82],l3[2]);
priority4bit     c3(o1[15:12],w[83],l3[3]);
//ce
ce   f  (cout[0 ],w[80  ][0 ],cout[64]);
ce   f1     (cout[1    ],w[80     ][0   ],cout[65  ]);
ce   f2     (cout[2    ],w[80     ][0   ],cout[66  ]);
ce   f3     (cout[3    ],w[80     ][0   ],cout[67  ]);
ce   f4     (cout[4    ],w[80     ][1   ],cout[68  ]);
ce   f5     (cout[5    ],w[80     ][1   ],cout[69  ]);
ce   f6     (cout[6    ],w[80     ][1 ],cout[70    ]);
ce   f7     (cout[7    ],w[80     ][1   ],cout[71  ]);
ce   f8     (cout[8    ],w[80     ][2   ],cout[72  ]);
ce   f9     (cout[9    ],w[80     ][2   ],cout[73  ]);
ce   f10(cout[10],w[80    ][2   ],cout[74  ]);
ce   f11(cout[11],w[80    ][2   ],cout[75  ]);
ce   f12(cout[12],w[80    ][3   ],cout[76  ]);
ce   f13(cout[13],w[80    ][3   ],cout[77  ]);
ce   f14(cout[14],w[80    ][3   ],cout[78  ]);
ce   f15(cout[15],w[80    ][3   ],cout[79  ]);
```

```
ce   f16(cout[16],w[81    ][0   ],cout[80  ]);
ce   f17(cout[17],w[81    ][0   ],cout[81  ]);
ce   f18(cout[18],w[81    ][0   ],cout[82  ]);
ce   f19(cout[19],w[81    ][0   ],cout[83  ]);
ce   f20(cout[20],w[81    ][1   ],cout[84  ]);
ce   f21(cout[21],w[81    ][1   ],cout[85  ]);
ce   f22(cout[22],w[81    ][1   ],cout[86  ]);
ce   f23(cout[23],w[81    ][1   ],cout[87  ]);
ce   f24(cout[24],w[81    ][2   ],cout[88  ]);
ce   f25(cout[25],w[81    ][2   ],cout[89  ]);
ce   f26(cout[26],w[81    ][2   ],cout[90  ]);
ce   f27(cout[27],w[81    ][2   ],cout[91  ]);
ce   f28(cout[28],w[81    ][3   ],cout[92  ]);
ce   f29(cout[29],w[81    ][3   ],cout[93  ]);
ce   f30(cout[30],w[81    ][3   ],cout[94  ]);
ce   f31(cout[31],w[81    ][3   ],cout[95  ]);
ce   f32(cout[32],w[82    ][0   ],cout[96  ]);
ce   f33(cout[33],w[82    ][0   ],cout[97  ]);
ce   f34(cout[34],w[82    ][0   ],cout[98  ]);
ce   f35(cout[35],w[82    ][0   ],cout[99  ]);
ce   f36(cout[36],w[82    ][1   ],cout[100       ]);
ce   f37(cout[37],w[82    ][1   ],cout[101       ]);
```

```
ce   f38(cout[38],w[82    ][1    ],cout[102       ]);
ce   f39(cout[39],w[82    ][1    ],cout[103       ]);
ce   f40(cout[40],w[82    ][2    ],cout[104       ]);
ce   f41(cout[41],w[82    ][2    ],cout[105       ]);
ce   f42(cout[42],w[82    ][2    ],cout[106       ]);
ce   f43(cout[43],w[82    ][2    ],cout[107       ]);
ce   f44(cout[44],w[82    ][3    ],cout[108       ]);
ce   f45(cout[45],w[82    ][3    ],cout[109       ]);
ce   f46(cout[46],w[82    ][3    ],cout[110       ]);
ce   f47(cout[47],w[82    ][3    ],cout[111       ]);
ce   f48(cout[48],w[83    ][0    ],cout[112       ]);
ce   f49(cout[49],w[83    ][0    ],cout[113       ]);
ce   f50(cout[50],w[83    ][0    ],cout[114       ]);
ce   f51(cout[51],w[83    ][0    ],cout[115       ]);
ce   f52(cout[52],w[83    ][1    ],cout[116       ]);
ce   f53(cout[53],w[83    ][1    ],cout[117       ]);
ce   f54(cout[54],w[83    ][1    ],cout[118       ]);
ce   f55(cout[55],w[83    ][1    ],cout[119       ]);
ce   f56(cout[56],w[83    ][2    ],cout[120       ]);
ce   f57(cout[57],w[83    ][2    ],cout[121       ]);
ce   f58(cout[58],w[83    ][2    ],cout[122       ]);
ce   f59(cout[59],w[83    ][2    ],cout[123       ]);
```

```
ce   f60(cout[60],w[83    ][3   ],cout[124       ]);
ce   f61(cout[61],w[83    ][3   ],cout[125       ]);
ce   f62(cout[62],w[83    ][3   ],cout[126       ]);
ce   f63(cout[63],w[83    ][3   ],cout[127       ]);



//or stage3
assign o2[       0      ]=w[80    ][0]|w[     80    ][1]|w[     80
][2]|w[     80    ][3];
assign o2[       1      ]=w[81    ][0]|w[     81    ][1]|w[     81
][2]|w[     81    ][3];
assign o2[       2      ]=w[82    ][0]|w[     82    ][1]|w[     82
][2]|w[     82    ][3];
assign o2[       3      ]=w[83    ][0]|w[     83    ][1]|w[     83
][2]|w[     83    ][3];
//stage 4
priority4bit    d(o2[3:0],w[84],l4);
//


ce   g       (cout[64  ],w[84     ][0   ],fin[0      ]);
ce   g1      (cout[65  ],w[84     ][0   ],fin[1      ]);
ce   g2      (cout[66  ],w[84     ][0   ],fin[2      ]);
```

```
ce   g3     (cout[67  ],w[84     ][0   ],fin[3     ]);
ce   g4     (cout[68  ],w[84     ][0   ],fin[4     ]);
ce   g5     (cout[69  ],w[84     ][0   ],fin[5     ]);
ce   g6     (cout[70  ],w[84     ][0   ],fin[6     ]);
ce   g7     (cout[71  ],w[84     ][0   ],fin[7     ]);
ce   g8     (cout[72  ],w[84     ][0   ],fin[8     ]);
ce   g9     (cout[73  ],w[84     ][0   ],fin[9     ]);
ce   g10(cout[74     ],w[84     ][0   ],fin[10    ]);
ce   g11(cout[75     ],w[84     ][0   ],fin[11    ]);
ce   g12(cout[76     ],w[84     ][0   ],fin[12    ]);
ce   g13(cout[77     ],w[84     ][0   ],fin[13    ]);
ce   g14(cout[78     ],w[84     ][0   ],fin[14    ]);
ce   g15(cout[79     ],w[84     ][0   ],fin[15    ]);
ce   g16(cout[80     ],w[84     ][1   ],fin[16    ]);
ce   g17(cout[81     ],w[84     ][1   ],fin[17    ]);
ce   g18(cout[82     ],w[84     ][1   ],fin[18    ]);
ce   g19(cout[83     ],w[84     ][1   ],fin[19    ]);
ce   g20(cout[84     ],w[84     ][1   ],fin[20    ]);
ce   g21(cout[85     ],w[84     ][1   ],fin[21    ]);
ce   g22(cout[86     ],w[84     ][1   ],fin[22    ]);
ce   g23(cout[87     ],w[84     ][1   ],fin[23    ]);
ce   g24(cout[88     ],w[84     ][1   ],fin[24    ]);
```

```
ce   g25(cout[89     ],w[84     ][1    ],fin[25    ]);
ce   g26(cout[90     ],w[84     ][1    ],fin[26    ]);
ce   g27(cout[91     ],w[84     ][1    ],fin[27    ]);
ce   g28(cout[92     ],w[84     ][1    ],fin[28    ]);
ce   g29(cout[93     ],w[84     ][1    ],fin[29    ]);
ce   g30(cout[94     ],w[84     ][1    ],fin[30    ]);
ce   g31(cout[95     ],w[84     ][1    ],fin[31    ]);
ce   g32(cout[96     ],w[84     ][2    ],fin[32    ]);
ce   g33(cout[97     ],w[84     ][2    ],fin[33    ]);
ce   g34(cout[98     ],w[84     ][2    ],fin[34    ]);
ce   g35(cout[99     ],w[84     ][2    ],fin[35    ]);
ce   g36(cout[100    ],w[84     ][2    ],fin[36    ]);
ce   g37(cout[101    ],w[84     ][2    ],fin[37    ]);
ce   g38(cout[102    ],w[84     ][2    ],fin[38    ]);
ce   g39(cout[103    ],w[84     ][2    ],fin[39    ]);
ce   g40(cout[104    ],w[84     ][2    ],fin[40    ]);
ce   g41(cout[105    ],w[84     ][2    ],fin[41    ]);
ce   g42(cout[106    ],w[84     ][2    ],fin[42    ]);
ce   g43(cout[107    ],w[84     ][2    ],fin[43    ]);
ce   g44(cout[108    ],w[84     ][2    ],fin[44    ]);
ce   g45(cout[109    ],w[84     ][2    ],fin[45    ]);
ce   g46(cout[110    ],w[84     ][2    ],fin[46    ]);
```

```
ce   g47(cout[111    ],w[84     ][2   ],fin[47    ]);
ce   g48(cout[112    ],w[84     ][3   ],fin[48    ]);
ce   g49(cout[113    ],w[84     ][3   ],fin[49    ]);
ce   g50(cout[114    ],w[84     ][3   ],fin[50    ]);
ce   g51(cout[115    ],w[84     ][3   ],fin[51    ]);
ce   g52(cout[116    ],w[84     ][3   ],fin[52    ]);
ce   g53(cout[117    ],w[84     ][3   ],fin[53    ]);
ce   g54(cout[118    ],w[84     ][3   ],fin[54    ]);
ce   g55(cout[119    ],w[84     ][3   ],fin[55    ]);
ce   g56(cout[120    ],w[84     ][3   ],fin[56    ]);
ce   g57(cout[121    ],w[84     ][3   ],fin[57    ]);
ce   g58(cout[122    ],w[84     ][3   ],fin[58    ]);
ce   g59(cout[123    ],w[84     ][3   ],fin[59    ]);
ce   g60(cout[124    ],w[84     ][3   ],fin[60    ]);
ce   g61(cout[125    ],w[84     ][3   ],fin[61    ]);
ce   g62(cout[126    ],w[84     ][3   ],fin[62    ]);
ce   g63(cout[127    ],w[84     ][3   ],fin[63    ]);


assign outfi[0  ]=fin[0     ][0   ];
assign outfi[1  ]=fin[0     ][1   ];
assign outfi[2  ]=fin[0     ][2   ];
```

```
assign outfi[3  ]=fin[0    ][3   ];
assign outfi[4  ]=fin[1    ][0   ];
assign outfi[5  ]=fin[1    ][1   ];
assign outfi[6  ]=fin[1    ][2   ];
assign outfi[7  ]=fin[1    ][3   ];
assign outfi[8  ]=fin[2    ][0   ];
assign outfi[9  ]=fin[2    ][1   ];
assign outfi[10]=fin[2    ][2   ];
assign outfi[11]=fin[2    ][3   ];
assign outfi[12]=fin[3    ][0   ];
assign outfi[13]=fin[3    ][1   ];
assign outfi[14]=fin[3    ][2   ];
assign outfi[15]=fin[3    ][3   ];
assign outfi[16]=fin[4    ][0   ];
assign outfi[17]=fin[4    ][1   ];
assign outfi[18]=fin[4    ][2   ];
assign outfi[19]=fin[4    ][3   ];
assign outfi[20]=fin[5    ][0   ];
assign outfi[21]=fin[5    ][1   ];
assign outfi[22]=fin[5    ][2   ];
assign outfi[23]=fin[5    ][3   ];
assign outfi[24]=fin[6    ][0   ];
```

```
assign outfi[25]=fin[6     ][1    ];
assign outfi[26]=fin[6     ][2    ];
assign outfi[27]=fin[6     ][3    ];
assign outfi[28]=fin[7     ][0    ];
assign outfi[29]=fin[7     ][1    ];
assign outfi[30]=fin[7     ][2    ];
assign outfi[31]=fin[7     ][3    ];
assign outfi[32]=fin[8     ][0    ];
assign outfi[33]=fin[8     ][1    ];
assign outfi[34]=fin[8     ][2    ];
assign outfi[35]=fin[8     ][3    ];
assign outfi[36]=fin[9     ][0    ];
assign outfi[37]=fin[9     ][1    ];
assign outfi[38]=fin[9     ][2    ];
assign outfi[39]=fin[9     ][3    ];
assign outfi[40]=fin[10    ][0    ];
assign outfi[41]=fin[10    ][1    ];
assign outfi[42]=fin[10    ][2    ];
assign outfi[43]=fin[10    ][3    ];
assign outfi[44]=fin[11    ][0    ];
assign outfi[45]=fin[11    ][1    ];
assign outfi[46]=fin[11    ][2    ];
```

```
assign outfi[47]=fin[11  ][3  ];
assign outfi[48]=fin[12  ][0  ];
assign outfi[49]=fin[12  ][1  ];
assign outfi[50]=fin[12  ][2  ];
assign outfi[51]=fin[12  ][3  ];
assign outfi[52]=fin[13  ][0  ];
assign outfi[53]=fin[13  ][1  ];
assign outfi[54]=fin[13  ][2  ];
assign outfi[55]=fin[13  ][3  ];
assign outfi[56]=fin[14  ][0  ];
assign outfi[57]=fin[14  ][1  ];
assign outfi[58]=fin[14  ][2  ];
assign outfi[59]=fin[14  ][3  ];
assign outfi[60]=fin[15  ][0  ];
assign outfi[61]=fin[15  ][1  ];
assign outfi[62]=fin[15  ][2  ];
assign outfi[63]=fin[15  ][3  ];
assign outfi[64]=fin[16  ][0  ];
assign outfi[65]=fin[16  ][1  ];
assign outfi[66]=fin[16  ][2  ];
assign outfi[67]=fin[16  ][3  ];
assign outfi[68]=fin[17  ][0  ];
```

```
assign outfi[69]=fin[17   ][1    ];
assign outfi[70]=fin[17   ][2    ];
assign outfi[71]=fin[17   ][3    ];
assign outfi[72]=fin[18   ][0    ];
assign outfi[73]=fin[18   ][1    ];
assign outfi[74]=fin[18   ][2    ];
assign outfi[75]=fin[18   ][3    ];
assign outfi[76]=fin[19   ][0    ];
assign outfi[77]=fin[19   ][1    ];
assign outfi[78]=fin[19   ][2    ];
assign outfi[79]=fin[19   ][3    ];
assign outfi[80]=fin[20   ][0    ];
assign outfi[81]=fin[20   ][1    ];
assign outfi[82]=fin[20   ][2    ];
assign outfi[83]=fin[20   ][3    ];
assign outfi[84]=fin[21   ][0    ];
assign outfi[85]=fin[21   ][1    ];
assign outfi[86]=fin[21   ][2    ];
assign outfi[87]=fin[21   ][3    ];
assign outfi[88]=fin[22   ][0    ];
assign outfi[89]=fin[22   ][1    ];
assign outfi[90]=fin[22   ][2    ];
```

```
assign outfi[91]=fin[22   ][3   ];
assign outfi[92]=fin[23   ][0   ];
assign outfi[93]=fin[23   ][1   ];
assign outfi[94]=fin[23   ][2   ];
assign outfi[95]=fin[23   ][3   ];
assign outfi[96]=fin[24   ][0   ];
assign outfi[97]=fin[24   ][1   ];
assign outfi[98]=fin[24   ][2   ];
assign outfi[99]=fin[24   ][3   ];
assign outfi[100      ]=fin[25   ][0   ];
assign outfi[101      ]=fin[25   ][1   ];
assign outfi[102      ]=fin[25   ][2   ];
assign outfi[103      ]=fin[25   ][3   ];
assign outfi[104      ]=fin[26   ][0   ];
assign outfi[105      ]=fin[26   ][1   ];
assign outfi[106      ]=fin[26   ][2   ];
assign outfi[107      ]=fin[26   ][3   ];
assign outfi[108      ]=fin[27   ][0   ];
assign outfi[109      ]=fin[27   ][1   ];
assign outfi[110      ]=fin[27   ][2   ];
assign outfi[111      ]=fin[27   ][3   ];
assign outfi[112      ]=fin[28   ][0   ];
```

```
assign outfi[113      ]=fin[28   ][1    ];
assign outfi[114      ]=fin[28   ][2    ];
assign outfi[115      ]=fin[28   ][3    ];
assign outfi[116      ]=fin[29   ][0    ];
assign outfi[117      ]=fin[29   ][1    ];
assign outfi[118      ]=fin[29   ][2    ];
assign outfi[119      ]=fin[29   ][3    ];
assign outfi[120      ]=fin[30   ][0    ];
assign outfi[121      ]=fin[30   ][1    ];
assign outfi[122      ]=fin[30   ][2    ];
assign outfi[123      ]=fin[30   ][3    ];
assign outfi[124      ]=fin[31   ][0    ];
assign outfi[125      ]=fin[31   ][1    ];
assign outfi[126      ]=fin[31   ][2    ];
assign outfi[127      ]=fin[31   ][3    ];
assign outfi[128      ]=fin[32   ][0    ];
assign outfi[129      ]=fin[32   ][1    ];
assign outfi[130      ]=fin[32   ][2    ];
assign outfi[131      ]=fin[32   ][3    ];
assign outfi[132      ]=fin[33   ][0    ];
assign outfi[133      ]=fin[33   ][1    ];
assign outfi[134      ]=fin[33   ][2    ];
```

```
assign outfi[135      ]=fin[33   ][3    ];
assign outfi[136      ]=fin[34   ][0    ];
assign outfi[137      ]=fin[34   ][1    ];
assign outfi[138      ]=fin[34   ][2    ];
assign outfi[139      ]=fin[34   ][3    ];
assign outfi[140      ]=fin[35   ][0    ];
assign outfi[141      ]=fin[35   ][1    ];
assign outfi[142      ]=fin[35   ][2    ];
assign outfi[143      ]=fin[35   ][3    ];
assign outfi[144      ]=fin[36   ][0    ];
assign outfi[145      ]=fin[36   ][1    ];
assign outfi[146      ]=fin[36   ][2    ];
assign outfi[147      ]=fin[36   ][3    ];
assign outfi[148      ]=fin[37   ][0    ];
assign outfi[149      ]=fin[37   ][1    ];
assign outfi[150      ]=fin[37   ][2    ];
assign outfi[151      ]=fin[37   ][3    ];
assign outfi[152      ]=fin[38   ][0    ];
assign outfi[153      ]=fin[38   ][1    ];
assign outfi[154      ]=fin[38   ][2    ];
assign outfi[155      ]=fin[38   ][3    ];
assign outfi[156      ]=fin[39   ][0    ];
```

```
assign outfi[157      ]=fin[39   ][1    ];
assign outfi[158      ]=fin[39   ][2    ];
assign outfi[159      ]=fin[39   ][3    ];
assign outfi[160      ]=fin[40   ][0    ];
assign outfi[161      ]=fin[40   ][1    ];
assign outfi[162      ]=fin[40   ][2    ];
assign outfi[163      ]=fin[40   ][3    ];
assign outfi[164      ]=fin[41   ][0    ];
assign outfi[165      ]=fin[41   ][1    ];
assign outfi[166      ]=fin[41   ][2    ];
assign outfi[167      ]=fin[41   ][3    ];
assign outfi[168      ]=fin[42   ][0    ];
assign outfi[169      ]=fin[42   ][1    ];
assign outfi[170      ]=fin[42   ][2    ];
assign outfi[171      ]=fin[42   ][3    ];
assign outfi[172      ]=fin[43   ][0    ];
assign outfi[173      ]=fin[43   ][1    ];
assign outfi[174      ]=fin[43   ][2    ];
assign outfi[175      ]=fin[43   ][3    ];
assign outfi[176      ]=fin[44   ][0    ];
assign outfi[177      ]=fin[44   ][1    ];
assign outfi[178      ]=fin[44   ][2    ];
```

```
assign outfi[179      ]=fin[44   ][3    ];
assign outfi[180      ]=fin[45   ][0    ];
assign outfi[181      ]=fin[45   ][1    ];
assign outfi[182      ]=fin[45   ][2    ];
assign outfi[183      ]=fin[45   ][3    ];
assign outfi[184      ]=fin[46   ][0    ];
assign outfi[185      ]=fin[46   ][1    ];
assign outfi[186      ]=fin[46   ][2    ];
assign outfi[187      ]=fin[46   ][3    ];
assign outfi[188      ]=fin[47   ][0    ];
assign outfi[189      ]=fin[47   ][1    ];
assign outfi[190      ]=fin[47   ][2    ];
assign outfi[191      ]=fin[47   ][3    ];
assign outfi[192      ]=fin[48   ][0    ];
assign outfi[193      ]=fin[48   ][1    ];
assign outfi[194      ]=fin[48   ][2    ];
assign outfi[195      ]=fin[48   ][3    ];
assign outfi[196      ]=fin[49   ][0    ];
assign outfi[197      ]=fin[49   ][1    ];
assign outfi[198      ]=fin[49   ][2    ];
assign outfi[199      ]=fin[49   ][3    ];
assign outfi[200      ]=fin[50   ][0    ];
```

```
assign outfi[201      ]=fin[50  ][1   ];
assign outfi[202      ]=fin[50  ][2   ];
assign outfi[203      ]=fin[50  ][3   ];
assign outfi[204      ]=fin[51  ][0   ];
assign outfi[205      ]=fin[51  ][1   ];
assign outfi[206      ]=fin[51  ][2   ];
assign outfi[207      ]=fin[51  ][3   ];
assign outfi[208      ]=fin[52  ][0   ];
assign outfi[209      ]=fin[52  ][1   ];
assign outfi[210      ]=fin[52  ][2   ];
assign outfi[211      ]=fin[52  ][3   ];
assign outfi[212      ]=fin[53  ][0   ];
assign outfi[213      ]=fin[53  ][1   ];
assign outfi[214      ]=fin[53  ][2   ];
assign outfi[215      ]=fin[53  ][3   ];
assign outfi[216      ]=fin[54  ][0   ];
assign outfi[217      ]=fin[54  ][1   ];
assign outfi[218      ]=fin[54  ][2   ];
assign outfi[219      ]=fin[54  ][3   ];
assign outfi[220      ]=fin[55  ][0   ];
assign outfi[221      ]=fin[55  ][1   ];
assign outfi[222      ]=fin[55  ][2   ];
```

```
assign outfi[223      ]=fin[55   ][3    ];
assign outfi[224      ]=fin[56   ][0    ];
assign outfi[225      ]=fin[56   ][1    ];
assign outfi[226      ]=fin[56   ][2    ];
assign outfi[227      ]=fin[56   ][3    ];
assign outfi[228      ]=fin[57   ][0    ];
assign outfi[229      ]=fin[57   ][1    ];
assign outfi[230      ]=fin[57   ][2    ];
assign outfi[231      ]=fin[57   ][3    ];
assign outfi[232      ]=fin[58   ][0    ];
assign outfi[233      ]=fin[58   ][1    ];
assign outfi[234      ]=fin[58   ][2    ];
assign outfi[235      ]=fin[58   ][3    ];
assign outfi[236      ]=fin[59   ][0    ];
assign outfi[237      ]=fin[59   ][1    ];
assign outfi[238      ]=fin[59   ][2    ];
assign outfi[239      ]=fin[59   ][3    ];
assign outfi[240      ]=fin[60   ][0    ];
assign outfi[241      ]=fin[60   ][1    ];
assign outfi[242      ]=fin[60   ][2    ];
assign outfi[243      ]=fin[60   ][3    ];
assign outfi[244      ]=fin[61   ][0    ];
```

```
assign outfi[245      ]=fin[61   ][1    ];
assign outfi[246      ]=fin[61   ][2    ];
assign outfi[247      ]=fin[61   ][3    ];
assign outfi[248      ]=fin[62   ][0    ];
assign outfi[249      ]=fin[62   ][1    ];
assign outfi[250      ]=fin[62   ][2    ];
assign outfi[251      ]=fin[62   ][3    ];
assign outfi[252      ]=fin[63   ][0    ];
assign outfi[253      ]=fin[63   ][1    ];
assign outfi[254      ]=fin[63   ][2    ];
assign outfi[255      ]=fin[63   ][3    ];

encoder_256 prrr(outfi,result,clk);



endmodule
```

priority4.v

```
module pe_256(D, result,l4,clk);
input clk;
input[255:0]D;
wire[3:0]w[84:0];// stage outputs
wire [63:0]o;//or output 1st stage
```

```
wire [16:0]o1;
wire [3:0]o2;
wire [63:0]l1;
wire [15:0]l2;
wire [3:0]l3;
output l4;
wire [3:0]cout[127:0];
wire [3:0]fin[63:0];
wire [255:0]outfi;
real result1;
integer n,p;
//real result1;
output [7:0]result;
//output [7:0]result1;

//stage1
priority4bit a(D[3:0],w[0],l1[0]);
priority4bit a1(D[7:4],w[1],l1[1]);
priority4bit  a2(D[11:8] ,w[2],l1[2]);
priority4bit  a3(D[15:12],w[3],l1[3]);
priority4bit  a4(D[19:16],w[4],l1[4]);
priority4bit  a5(D[23:20],w[5],l1[5]);
```

```
priority4bit  a6(D[27:24],w[6],l1[6]);
priority4bit  a7(D[31:28],w[7],l1[7]);
priority4bit  a8(D[35:32],w[8],l1[8]);
priority4bit  a9(D[39:36],w[9],l1[9]);
priority4bit  a10(D[43:40],w[10],l1[10]);
priority4bit  a11(D[47:44],w[11],l1[11]);
priority4bit  a12(D[51:48],w[12],l1[12]);
priority4bit  a13(D[55:52],w[13],l1[13]);
priority4bit  a14(D[59:56],w[14],l1[14]);
priority4bit  a15(D[63:60],w[15],l1[15]);
priority4bit  a16(D[67:64],w[16],l1[16]);
priority4bit  a17(D[71:68],w[17],l1[17]);
priority4bit  a18(D[75:72],w[18],l1[18]);
priority4bit  a19(D[79:76],w[19],l1[19]);
priority4bit  a20(D[83:80],w[20],l1[20]);
priority4bit  a21(D[87:84],w[21],l1[21]);
priority4bit  a22(D[91:88],w[22],l1[22]);
priority4bit  a23(D[95:92],w[23],l1[23]);
priority4bit  a24(D[99:96],w[24],l1[24]);
priority4bit  a25(D[103:100],w[25],l1[25]);
priority4bit  a26(D[107:104],w[26],l1[26]);
priority4bit  a27(D[111:108],w[27],l1[27]);
```

```
priority4bit  a28(D[115:112],w[28],l1[28]);
priority4bit  a29(D[119:116],w[29],l1[29]);
priority4bit  a30(D[123:120],w[30],l1[30]);
priority4bit  a31(D[127:124],w[31],l1[31]);
priority4bit  a32(D[131:128],w[32],l1[32]);
priority4bit  a33(D[135:132],w[33],l1[33]);
priority4bit  a34(D[139:136],w[34],l1[34]);
priority4bit  a35(D[143:140],w[35],l1[35]);
priority4bit  a36(D[147:144],w[36],l1[36]);
priority4bit  a37(D[151:148],w[37],l1[37]);
priority4bit  a38(D[155:152],w[38],l1[38]);
priority4bit  a39(D[159:156],w[39],l1[39]);
priority4bit  a40(D[163:160],w[40],l1[40]);
priority4bit  a41(D[167:164],w[41],l1[41]);
priority4bit  a42(D[171:168],w[42],l1[42]);
priority4bit  a43(D[175:172],w[43],l1[43]);
priority4bit  a44(D[179:176],w[44],l1[44]);
priority4bit  a45(D[183:180],w[45],l1[45]);
priority4bit  a46(D[187:184],w[46],l1[46]);
priority4bit  a47(D[191:188],w[47],l1[47]);
priority4bit  a48(D[195:192],w[48],l1[48]);
priority4bit  a49(D[199:196],w[49],l1[49]);
```

```
priority4bit  a50(D[203:200],w[50],l1[50]);
priority4bit  a51(D[207:204],w[51],l1[51]);
priority4bit  a52(D[211:208],w[52],l1[52]);
priority4bit  a53(D[215:212],w[53],l1[53]);
priority4bit  a54(D[219:216],w[54],l1[54]);
priority4bit  a55(D[223:220],w[55],l1[55]);
priority4bit  a56(D[227:224],w[56],l1[56]);
priority4bit  a57(D[231:228],w[57],l1[57]);
priority4bit  a58(D[235:232],w[58],l1[58]);
priority4bit  a59(D[239:236],w[59],l1[59]);
priority4bit  a60(D[243:240],w[60],l1[60]);
priority4bit  a61(D[247:244],w[61],l1[61]);
priority4bit  a62(D[251:248],w[62],l1[62]);
priority4bit  a63(D[255:252],w[63],l1[63]);

//stage 1 or
assign o[   0   ]=w[    0   ][0]|w[ 0   ][1]|w[ 0   ][2]|w[ 0   ][3];
assign o[  1      ]=w[1      ][0]|w[     1      ][1]|w[     1      ][2]|w[
        1      ][3];
assign o[  2      ]=w[2      ][0]|w[     2      ][1]|w[     2      ][2]|w[
        2      ][3];
```

```
assign o[3]=w[3][0]|w[3][1]|w[3][2]|w[3][3];
assign o[4]=w[4][0]|w[4][1]|w[4][2]|w[4][3];
assign o[5]=w[5][0]|w[5][1]|w[5][2]|w[5][3];
assign o[6]=w[6][0]|w[6][1]|w[6][2]|w[6][3];
assign o[7]=w[7][0]|w[7][1]|w[7][2]|w[7][3];
assign o[8]=w[8][0]|w[8][1]|w[8][2]|w[8][3];
assign o[9]=w[9][0]|w[9][1]|w[9][2]|w[9][3];
assign o[10]=w[10][0]|w[10][1]|w[10][2]|w[10][3];
assign o[11]=w[11][0]|w[11][1]|w[11][2]|w[11][3];
assign o[12]=w[12][0]|w[12][1]|w[12][2]|w[12][3];
assign o[13]=w[13][0]|w[13][1]|w[13][2]|w[13][3];
assign o[14]=w[14][0]|w[14][1]|w[14][2]|w[14][3];
```

```
assign o[15]=w[15][0]|w[15][1]|w[15][2]|w[15][3];
assign o[16]=w[16][0]|w[16][1]|w[16][2]|w[16][3];
assign o[17]=w[17][0]|w[17][1]|w[17][2]|w[17][3];
assign o[18]=w[18][0]|w[18][1]|w[18][2]|w[18][3];
assign o[19]=w[19][0]|w[19][1]|w[19][2]|w[19][3];
assign o[20]=w[20][0]|w[20][1]|w[20][2]|w[20][3];
assign o[21]=w[21][0]|w[21][1]|w[21][2]|w[21][3];
assign o[22]=w[22][0]|w[22][1]|w[22][2]|w[22][3];
assign o[23]=w[23][0]|w[23][1]|w[23][2]|w[23][3];
assign o[24]=w[24][0]|w[24][1]|w[24][2]|w[24][3];
assign o[25]=w[25][0]|w[25][1]|w[25][2]|w[25][3];
assign o[26]=w[26][0]|w[26][1]|w[26][2]|w[26][3];
```

```
assign o[27]=w[27][0]|w[27][1]|w[27][2]|w[27][3];
assign o[28]=w[28][0]|w[28][1]|w[28][2]|w[28][3];
assign o[29]=w[29][0]|w[29][1]|w[29][2]|w[29][3];
assign o[30]=w[30][0]|w[30][1]|w[30][2]|w[30][3];
assign o[31]=w[31][0]|w[31][1]|w[31][2]|w[31][3];
assign o[32]=w[32][0]|w[32][1]|w[32][2]|w[32][3];
assign o[33]=w[33][0]|w[33][1]|w[33][2]|w[33][3];
assign o[34]=w[34][0]|w[34][1]|w[34][2]|w[34][3];
assign o[35]=w[35][0]|w[35][1]|w[35][2]|w[35][3];
assign o[36]=w[36][0]|w[36][1]|w[36][2]|w[36][3];
assign o[37]=w[37][0]|w[37][1]|w[37][2]|w[37][3];
assign o[38]=w[38][0]|w[38][1]|w[38][2]|w[38][3];
```

```
assign o[39]=w[39][0]|w[39][1]|w[39][2]|w[39][3];
assign o[40]=w[40][0]|w[40][1]|w[40][2]|w[40][3];
assign o[41]=w[41][0]|w[41][1]|w[41][2]|w[41][3];
assign o[42]=w[42][0]|w[42][1]|w[42][2]|w[42][3];
assign o[43]=w[43][0]|w[43][1]|w[43][2]|w[43][3];
assign o[44]=w[44][0]|w[44][1]|w[44][2]|w[44][3];
assign o[45]=w[45][0]|w[45][1]|w[45][2]|w[45][3];
assign o[46]=w[46][0]|w[46][1]|w[46][2]|w[46][3];
assign o[47]=w[47][0]|w[47][1]|w[47][2]|w[47][3];
assign o[48]=w[48][0]|w[48][1]|w[48][2]|w[48][3];
assign o[49]=w[49][0]|w[49][1]|w[49][2]|w[49][3];
assign o[50]=w[50][0]|w[50][1]|w[50][2]|w[50][3];
```

```
assign o[51]=w[51][0]|w[51][1]|w[51][2]|w[51][3];
assign o[52]=w[52][0]|w[52][1]|w[52][2]|w[52][3];
assign o[53]=w[53][0]|w[53][1]|w[53][2]|w[53][3];
assign o[54]=w[54][0]|w[54][1]|w[54][2]|w[54][3];
assign o[55]=w[55][0]|w[55][1]|w[55][2]|w[55][3];
assign o[56]=w[56][0]|w[56][1]|w[56][2]|w[56][3];
assign o[57]=w[57][0]|w[57][1]|w[57][2]|w[57][3];
assign o[58]=w[58][0]|w[58][1]|w[58][2]|w[58][3];
assign o[59]=w[59][0]|w[59][1]|w[59][2]|w[59][3];
assign o[60]=w[60][0]|w[60][1]|w[60][2]|w[60][3];
assign o[61]=w[61][0]|w[61][1]|w[61][2]|w[61][3];
assign o[62]=w[62][0]|w[62][1]|w[62][2]|w[62][3];
```

```
assign o[ 63 ]=w[63 ][0]|w[ 63 ][1]|w[ 63 ][2]|w[
    63 ][3];

//stage 2
priority4bit  b(o[3:0],w[64],l2[0]);
priority4bit   b1(o[7:4],w[65],l2[1]);
priority4bit   b2(o[11:8],w[66],l2[2]);
priority4bit   b3(o[15:12],w[67],l2[3]);
priority4bit   b4(o[19:16],w[68],l2[4]);
priority4bit   b5(o[23:20],w[69],l2[5]);
priority4bit   b6(o[27:24],w[70],l2[6]);
priority4bit   b7(o[31:28],w[71],l2[7]);
priority4bit   b8(o[35:32],w[72],l2[8]);
priority4bit   b9(o[39:36],w[73],l2[9]);
priority4bit   b10(o[43:40],w[74],l2[10]);
priority4bit   b11(o[47:44],w[75],l2[11]);
priority4bit   b12(o[51:48],w[76],l2[12]);
priority4bit   b13(o[55:52],w[77],l2[13]);
priority4bit   b14(o[59:56],w[78],l2[14]);
priority4bit   b15(o[63:60],w[79],l2[15]);

//ce
```

```
ce   e  (w[0    ],w[64  ][0 ],cout[0    ]);
ce   e1     (w[1 ],w[64     ][1   ],cout[1   ]);
ce   e2     (w[2 ],w[64     ][2   ],cout[2   ]);
ce   e3     (w[3 ],w[64     ][3   ],cout[3   ]);
ce   e4     (w[4 ],w[65     ][0   ],cout[4   ]);
ce   e5     (w[5 ],w[65     ][1   ],cout[5   ]);
ce   e6     (w[6 ],w[65     ][2   ],cout[6   ]);
ce   e7     (w[7 ],w[65     ][3   ],cout[7   ]);
ce   e8     (w[8 ],w[66     ][0   ],cout[8   ]);
ce   e9     (w[9 ],w[66     ][1   ],cout[9   ]);
ce   e10    (w[10      ],w[66     ][2   ],cout[10  ]);
ce   e11    (w[11      ],w[66     ][3   ],cout[11  ]);
ce   e12    (w[12      ],w[67     ][0   ],cout[12  ]);
ce   e13    (w[13      ],w[67     ][1   ],cout[13  ]);
ce   e14    (w[14      ],w[67     ][2   ],cout[14  ]);
ce   e15    (w[15      ],w[67     ][3   ],cout[15  ]);
ce   e16    (w[16      ],w[68     ][0   ],cout[16  ]);
ce   e17    (w[17      ],w[68     ][1   ],cout[17  ]);
ce   e18    (w[18      ],w[68     ][2   ],cout[18  ]);
ce   e19    (w[19      ],w[68     ][3   ],cout[19  ]);
ce   e20    (w[20      ],w[69     ][0   ],cout[20  ]);
ce   e21    (w[21      ],w[69     ][1   ],cout[21  ]);
```

```
ce e22 (w[22 ],w[69 ][2 ],cout[22 ]);
ce e23 (w[23 ],w[69 ][3 ],cout[23 ]);
ce e24 (w[24 ],w[70 ][0 ],cout[24 ]);
ce e25 (w[25 ],w[70 ][1 ],cout[25 ]);
ce e26 (w[26 ],w[70 ][2 ],cout[26 ]);
ce e27 (w[27 ],w[70 ][3 ],cout[27 ]);
ce e28 (w[28 ],w[71 ][0 ],cout[28 ]);
ce e29 (w[29 ],w[71 ][1 ],cout[29 ]);
ce e30 (w[30 ],w[71 ][2 ],cout[30 ]);
ce e31 (w[31 ],w[71 ][3 ],cout[31 ]);
ce e32 (w[32 ],w[72 ][0 ],cout[32 ]);
ce e33 (w[33 ],w[72 ][1 ],cout[33 ]);
ce e34 (w[34 ],w[72 ][2 ],cout[34 ]);
ce e35 (w[35 ],w[72 ][3 ],cout[35 ]);
ce e36 (w[36 ],w[73 ][0 ],cout[36 ]);
ce e37 (w[37 ],w[73 ][1 ],cout[37 ]);
ce e38 (w[38 ],w[73 ][2 ],cout[38 ]);
ce e39 (w[39 ],w[73 ][3 ],cout[39 ]);
ce e40 (w[40 ],w[74 ][0 ],cout[40 ]);
ce e41 (w[41 ],w[74 ][1 ],cout[41 ]);
ce e42 (w[42 ],w[74 ][2 ],cout[42 ]);
ce e43 (w[43 ],w[74 ][3 ],cout[43 ]);
```

```
ce e44 (w[44 ],w[75 ][0 ],cout[44 ]);
ce e45 (w[45 ],w[75 ][1 ],cout[45 ]);
ce e46 (w[46 ],w[75 ][2 ],cout[46 ]);
ce e47 (w[47 ],w[75 ][3 ],cout[47 ]);
ce e48 (w[48 ],w[76 ][0 ],cout[48 ]);
ce e49 (w[49 ],w[76 ][1 ],cout[49 ]);
ce e50 (w[50 ],w[76 ][2 ],cout[50 ]);
ce e51 (w[51 ],w[76 ][3 ],cout[51 ]);
ce e52 (w[52 ],w[77 ][0 ],cout[52 ]);
ce e53 (w[53 ],w[77 ][1 ],cout[53 ]);
ce e54 (w[54 ],w[77 ][2 ],cout[54 ]);
ce e55 (w[55 ],w[77 ][3 ],cout[55 ]);
ce e56 (w[56 ],w[78 ][0 ],cout[56 ]);
ce e57 (w[57 ],w[78 ][1 ],cout[57 ]);
ce e58 (w[58 ],w[78 ][2 ],cout[58 ]);
ce e59 (w[59 ],w[78 ][3 ],cout[59 ]);
ce e60 (w[60 ],w[79 ][0 ],cout[60 ]);
ce e61 (w[61 ],w[79 ][1 ],cout[61 ]);
ce e62 (w[62 ],w[79 ][2 ],cout[62 ]);
ce e63 (w[63 ],w[79 ][3 ],cout[63 ]);
```

//or stage2

```
assign o1[ 0 ]=w[64 ][0]|w[ 64 ][1]|w[ 64 ][2]|w[ 64 ][3];
assign o1[ 1 ]=w[65 ][0]|w[ 65 ][1]|w[ 65 ][2]|w[ 65 ][3];
assign o1[ 2 ]=w[66 ][0]|w[ 66 ][1]|w[ 66 ][2]|w[ 66 ][3];
assign o1[ 3 ]=w[67 ][0]|w[ 67 ][1]|w[ 67 ][2]|w[ 67 ][3];
assign o1[ 4 ]=w[68 ][0]|w[ 68 ][1]|w[ 68 ][2]|w[ 68 ][3];
assign o1[ 5 ]=w[69 ][0]|w[ 69 ][1]|w[ 69 ][2]|w[ 69 ][3];
assign o1[ 6 ]=w[70 ][0]|w[ 70 ][1]|w[ 70 ][2]|w[ 70 ][3];
assign o1[ 7 ]=w[71 ][0]|w[ 71 ][1]|w[ 71 ][2]|w[ 71 ][3];
assign o1[ 8 ]=w[72 ][0]|w[ 72 ][1]|w[ 72 ][2]|w[ 72 ][3];
assign o1[ 9 ]=w[73 ][0]|w[ 73 ][1]|w[ 73 ][2]|w[ 73 ][3];
assign o1[ 10 ]=w[74 ][0]|w[ 74 ][1]|w[ 74 ][2]|w[ 74 ][3];
```

```
assign o1[ 11 ]=w[75 ][0]|w[ 75 ][1]|w[ 75 ][2]|w[ 75 ][3];
assign o1[ 12 ]=w[76 ][0]|w[ 76 ][1]|w[ 76 ][2]|w[ 76 ][3];
assign o1[ 13 ]=w[77 ][0]|w[ 77 ][1]|w[ 77 ][2]|w[ 77 ][3];
assign o1[ 14 ]=w[78 ][0]|w[ 78 ][1]|w[ 78 ][2]|w[ 78 ][3];
assign o1[ 15 ]=w[ 79 ][0]|w[ 79 ][1]|w[ 79 ][2]|w[ 79 ][3];

//stage3
priority4bit c(o1[3:0],w[80],l3[0]);
priority4bit c1(o1[7:4],w[81],l3[1]);
priority4bit c2(o1[11:8],w[82],l3[2]);
priority4bit c3(o1[15:12],w[83],l3[3]);
//ce
ce f (cout[0 ],w[80 ][0 ],cout[64]);
ce f1 (cout[1 ],w[80 ][0 ],cout[65 ]);
ce f2 (cout[2 ],w[80 ][0 ],cout[66 ]);
ce f3 (cout[3 ],w[80 ][0 ],cout[67 ]);
ce f4 (cout[4 ],w[80 ][1 ],cout[68 ]);
ce f5 (cout[5 ],w[80 ][1 ],cout[69 ]);
```

```
ce f6  (cout[6 ],w[80 ][1 ],cout[70 ]);
ce f7  (cout[7 ],w[80 ][1 ],cout[71 ]);
ce f8  (cout[8 ],w[80 ][2 ],cout[72 ]);
ce f9  (cout[9 ],w[80 ][2 ],cout[73 ]);
ce f10(cout[10],w[80 ][2 ],cout[74 ]);
ce f11(cout[11],w[80 ][2 ],cout[75 ]);
ce f12(cout[12],w[80 ][3 ],cout[76 ]);
ce f13(cout[13],w[80 ][3 ],cout[77 ]);
ce f14(cout[14],w[80 ][3 ],cout[78 ]);
ce f15(cout[15],w[80 ][3 ],cout[79 ]);
ce f16(cout[16],w[81 ][0 ],cout[80 ]);
ce f17(cout[17],w[81 ][0 ],cout[81 ]);
ce f18(cout[18],w[81 ][0 ],cout[82 ]);
ce f19(cout[19],w[81 ][0 ],cout[83 ]);
ce f20(cout[20],w[81 ][1 ],cout[84 ]);
ce f21(cout[21],w[81 ][1 ],cout[85 ]);
ce f22(cout[22],w[81 ][1 ],cout[86 ]);
ce f23(cout[23],w[81 ][1 ],cout[87 ]);
ce f24(cout[24],w[81 ][2 ],cout[88 ]);
ce f25(cout[25],w[81 ][2 ],cout[89 ]);
ce f26(cout[26],w[81 ][2 ],cout[90 ]);
ce f27(cout[27],w[81 ][2 ],cout[91 ]);
```

```
ce   f28(cout[28],w[81    ][3   ],cout[92  ]);
ce   f29(cout[29],w[81    ][3   ],cout[93  ]);
ce   f30(cout[30],w[81    ][3   ],cout[94  ]);
ce   f31(cout[31],w[81    ][3   ],cout[95  ]);
ce   f32(cout[32],w[82    ][0   ],cout[96  ]);
ce   f33(cout[33],w[82    ][0   ],cout[97  ]);
ce   f34(cout[34],w[82    ][0   ],cout[98  ]);
ce   f35(cout[35],w[82    ][0   ],cout[99  ]);
ce   f36(cout[36],w[82    ][1   ],cout[100       ]);
ce   f37(cout[37],w[82    ][1   ],cout[101       ]);
ce   f38(cout[38],w[82    ][1   ],cout[102       ]);
ce   f39(cout[39],w[82    ][1   ],cout[103       ]);
ce   f40(cout[40],w[82    ][2   ],cout[104       ]);
ce   f41(cout[41],w[82    ][2   ],cout[105       ]);
ce   f42(cout[42],w[82    ][2   ],cout[106       ]);
ce   f43(cout[43],w[82    ][2   ],cout[107       ]);
ce   f44(cout[44],w[82    ][3   ],cout[108       ]);
ce   f45(cout[45],w[82    ][3   ],cout[109       ]);
ce   f46(cout[46],w[82    ][3   ],cout[110       ]);
ce   f47(cout[47],w[82    ][3   ],cout[111       ]);
ce   f48(cout[48],w[83    ][0   ],cout[112       ]);
ce   f49(cout[49],w[83    ][0   ],cout[113       ]);
```

```
ce   f50(cout[50],w[83    ][0   ],cout[114       ]);
ce   f51(cout[51],w[83    ][0   ],cout[115       ]);
ce   f52(cout[52],w[83    ][1   ],cout[116       ]);
ce   f53(cout[53],w[83    ][1   ],cout[117       ]);
ce   f54(cout[54],w[83    ][1   ],cout[118       ]);
ce   f55(cout[55],w[83    ][1   ],cout[119       ]);
ce   f56(cout[56],w[83    ][2   ],cout[120       ]);
ce   f57(cout[57],w[83    ][2   ],cout[121       ]);
ce   f58(cout[58],w[83    ][2   ],cout[122       ]);
ce   f59(cout[59],w[83    ][2   ],cout[123       ]);
ce   f60(cout[60],w[83    ][3   ],cout[124       ]);
ce   f61(cout[61],w[83    ][3   ],cout[125       ]);
ce   f62(cout[62],w[83    ][3   ],cout[126       ]);
ce   f63(cout[63],w[83    ][3   ],cout[127       ]);



//or stage3
assign o2[       0       ]=w[80    ][0]|w[     80    ][1]|w[     80
][2]|w[     80    ][3];
assign o2[       1       ]=w[81    ][0]|w[     81    ][1]|w[     81
][2]|w[     81    ][3];
```

```
assign o2[ 2 ]=w[82 ][0]|w[ 82 ][1]|w[ 82 ][2]|w[ 82 ][3];
assign o2[ 3 ]=w[83 ][0]|w[ 83 ][1]|w[ 83 ][2]|w[ 83 ][3];
//stage 4
priority4bit d(o2[3:0],w[84],l4);
//

ce g (cout[64 ],w[84 ][0 ],fin[0 ]);
ce g1 (cout[65 ],w[84 ][0 ],fin[1 ]);
ce g2 (cout[66 ],w[84 ][0 ],fin[2 ]);
ce g3 (cout[67 ],w[84 ][0 ],fin[3 ]);
ce g4 (cout[68 ],w[84 ][0 ],fin[4 ]);
ce g5 (cout[69 ],w[84 ][0 ],fin[5 ]);
ce g6 (cout[70 ],w[84 ][0 ],fin[6 ]);
ce g7 (cout[71 ],w[84 ][0 ],fin[7 ]);
ce g8 (cout[72 ],w[84 ][0 ],fin[8 ]);
ce g9 (cout[73 ],w[84 ][0 ],fin[9 ]);
ce g10(cout[74 ],w[84 ][0 ],fin[10 ]);
ce g11(cout[75 ],w[84 ][0 ],fin[11 ]);
ce g12(cout[76 ],w[84 ][0 ],fin[12 ]);
ce g13(cout[77 ],w[84 ][0 ],fin[13 ]);
```

```
ce   g14(cout[78     ],w[84     ][0   ],fin[14    ]);
ce   g15(cout[79     ],w[84     ][0   ],fin[15    ]);
ce   g16(cout[80     ],w[84     ][1   ],fin[16    ]);
ce   g17(cout[81     ],w[84     ][1   ],fin[17    ]);
ce   g18(cout[82     ],w[84     ][1   ],fin[18    ]);
ce   g19(cout[83     ],w[84     ][1   ],fin[19    ]);
ce   g20(cout[84     ],w[84     ][1   ],fin[20    ]);
ce   g21(cout[85     ],w[84     ][1   ],fin[21    ]);
ce   g22(cout[86     ],w[84     ][1   ],fin[22    ]);
ce   g23(cout[87     ],w[84     ][1   ],fin[23    ]);
ce   g24(cout[88     ],w[84     ][1   ],fin[24    ]);
ce   g25(cout[89     ],w[84     ][1   ],fin[25    ]);
ce   g26(cout[90     ],w[84     ][1   ],fin[26    ]);
ce   g27(cout[91     ],w[84     ][1   ],fin[27    ]);
ce   g28(cout[92     ],w[84     ][1   ],fin[28    ]);
ce   g29(cout[93     ],w[84     ][1   ],fin[29    ]);
ce   g30(cout[94     ],w[84     ][1   ],fin[30    ]);
ce   g31(cout[95     ],w[84     ][1   ],fin[31    ]);
ce   g32(cout[96     ],w[84     ][2   ],fin[32    ]);
ce   g33(cout[97     ],w[84     ][2   ],fin[33    ]);
ce   g34(cout[98     ],w[84     ][2   ],fin[34    ]);
ce   g35(cout[99     ],w[84     ][2   ],fin[35    ]);
```

```
ce   g36(cout[100   ],w[84    ][2   ],fin[36   ]);
ce   g37(cout[101   ],w[84    ][2   ],fin[37   ]);
ce   g38(cout[102   ],w[84    ][2   ],fin[38   ]);
ce   g39(cout[103   ],w[84    ][2   ],fin[39   ]);
ce   g40(cout[104   ],w[84    ][2   ],fin[40   ]);
ce   g41(cout[105   ],w[84    ][2   ],fin[41   ]);
ce   g42(cout[106   ],w[84    ][2   ],fin[42   ]);
ce   g43(cout[107   ],w[84    ][2   ],fin[43   ]);
ce   g44(cout[108   ],w[84    ][2   ],fin[44   ]);
ce   g45(cout[109   ],w[84    ][2   ],fin[45   ]);
ce   g46(cout[110   ],w[84    ][2   ],fin[46   ]);
ce   g47(cout[111   ],w[84    ][2   ],fin[47   ]);
ce   g48(cout[112   ],w[84    ][3   ],fin[48   ]);
ce   g49(cout[113   ],w[84    ][3   ],fin[49   ]);
ce   g50(cout[114   ],w[84    ][3   ],fin[50   ]);
ce   g51(cout[115   ],w[84    ][3   ],fin[51   ]);
ce   g52(cout[116   ],w[84    ][3   ],fin[52   ]);
ce   g53(cout[117   ],w[84    ][3   ],fin[53   ]);
ce   g54(cout[118   ],w[84    ][3   ],fin[54   ]);
ce   g55(cout[119   ],w[84    ][3   ],fin[55   ]);
ce   g56(cout[120   ],w[84    ][3   ],fin[56   ]);
ce   g57(cout[121   ],w[84    ][3   ],fin[57   ]);
```

```
ce   g58(cout[122   ],w[84    ][3   ],fin[58   ]);
ce   g59(cout[123   ],w[84    ][3   ],fin[59   ]);
ce   g60(cout[124   ],w[84    ][3   ],fin[60   ]);
ce   g61(cout[125   ],w[84    ][3   ],fin[61   ]);
ce   g62(cout[126   ],w[84    ][3   ],fin[62   ]);
ce   g63(cout[127   ],w[84    ][3   ],fin[63   ]);


assign outfi[0  ]=fin[0    ][0   ];
assign outfi[1  ]=fin[0    ][1   ];
assign outfi[2  ]=fin[0    ][2   ];
assign outfi[3  ]=fin[0    ][3   ];
assign outfi[4  ]=fin[1    ][0   ];
assign outfi[5  ]=fin[1    ][1   ];
assign outfi[6  ]=fin[1    ][2   ];
assign outfi[7  ]=fin[1    ][3   ];
assign outfi[8  ]=fin[2    ][0   ];
assign outfi[9  ]=fin[2    ][1   ];
assign outfi[10]=fin[2    ][2   ];
assign outfi[11]=fin[2    ][3   ];
assign outfi[12]=fin[3    ][0   ];
assign outfi[13]=fin[3    ][1   ];
```

```
assign outfi[14]=fin[3     ][2    ];
assign outfi[15]=fin[3     ][3    ];
assign outfi[16]=fin[4     ][0    ];
assign outfi[17]=fin[4     ][1    ];
assign outfi[18]=fin[4     ][2    ];
assign outfi[19]=fin[4     ][3    ];
assign outfi[20]=fin[5     ][0    ];
assign outfi[21]=fin[5     ][1    ];
assign outfi[22]=fin[5     ][2    ];
assign outfi[23]=fin[5     ][3    ];
assign outfi[24]=fin[6     ][0    ];
assign outfi[25]=fin[6     ][1    ];
assign outfi[26]=fin[6     ][2    ];
assign outfi[27]=fin[6     ][3    ];
assign outfi[28]=fin[7     ][0    ];
assign outfi[29]=fin[7     ][1    ];
assign outfi[30]=fin[7     ][2    ];
assign outfi[31]=fin[7     ][3    ];
assign outfi[32]=fin[8     ][0    ];
assign outfi[33]=fin[8     ][1    ];
assign outfi[34]=fin[8     ][2    ];
assign outfi[35]=fin[8     ][3    ];
```

```
assign outfi[36]=fin[9    ][0   ];
assign outfi[37]=fin[9    ][1   ];
assign outfi[38]=fin[9    ][2   ];
assign outfi[39]=fin[9    ][3   ];
assign outfi[40]=fin[10   ][0   ];
assign outfi[41]=fin[10   ][1   ];
assign outfi[42]=fin[10   ][2   ];
assign outfi[43]=fin[10   ][3   ];
assign outfi[44]=fin[11   ][0   ];
assign outfi[45]=fin[11   ][1   ];
assign outfi[46]=fin[11   ][2   ];
assign outfi[47]=fin[11   ][3   ];
assign outfi[48]=fin[12   ][0   ];
assign outfi[49]=fin[12   ][1   ];
assign outfi[50]=fin[12   ][2   ];
assign outfi[51]=fin[12   ][3   ];
assign outfi[52]=fin[13   ][0   ];
assign outfi[53]=fin[13   ][1   ];
assign outfi[54]=fin[13   ][2   ];
assign outfi[55]=fin[13   ][3   ];
assign outfi[56]=fin[14   ][0   ];
assign outfi[57]=fin[14   ][1   ];
```

```
assign outfi[58]=fin[14   ][2   ];
assign outfi[59]=fin[14   ][3   ];
assign outfi[60]=fin[15   ][0   ];
assign outfi[61]=fin[15   ][1   ];
assign outfi[62]=fin[15   ][2   ];
assign outfi[63]=fin[15   ][3   ];
assign outfi[64]=fin[16   ][0   ];
assign outfi[65]=fin[16   ][1   ];
assign outfi[66]=fin[16   ][2   ];
assign outfi[67]=fin[16   ][3   ];
assign outfi[68]=fin[17   ][0   ];
assign outfi[69]=fin[17   ][1   ];
assign outfi[70]=fin[17   ][2   ];
assign outfi[71]=fin[17   ][3   ];
assign outfi[72]=fin[18   ][0   ];
assign outfi[73]=fin[18   ][1   ];
assign outfi[74]=fin[18   ][2   ];
assign outfi[75]=fin[18   ][3   ];
assign outfi[76]=fin[19   ][0   ];
assign outfi[77]=fin[19   ][1   ];
assign outfi[78]=fin[19   ][2   ];
assign outfi[79]=fin[19   ][3   ];
```

```
assign outfi[80]=fin[20  ][0   ];
assign outfi[81]=fin[20  ][1   ];
assign outfi[82]=fin[20  ][2   ];
assign outfi[83]=fin[20  ][3   ];
assign outfi[84]=fin[21  ][0   ];
assign outfi[85]=fin[21  ][1   ];
assign outfi[86]=fin[21  ][2   ];
assign outfi[87]=fin[21  ][3   ];
assign outfi[88]=fin[22  ][0   ];
assign outfi[89]=fin[22  ][1   ];
assign outfi[90]=fin[22  ][2   ];
assign outfi[91]=fin[22  ][3   ];
assign outfi[92]=fin[23  ][0   ];
assign outfi[93]=fin[23  ][1   ];
assign outfi[94]=fin[23  ][2   ];
assign outfi[95]=fin[23  ][3   ];
assign outfi[96]=fin[24  ][0   ];
assign outfi[97]=fin[24  ][1   ];
assign outfi[98]=fin[24  ][2   ];
assign outfi[99]=fin[24  ][3   ];
assign outfi[100     ]=fin[25  ][0   ];
assign outfi[101     ]=fin[25  ][1   ];
```

```
assign outfi[102      ]=fin[25   ][2    ];
assign outfi[103      ]=fin[25   ][3    ];
assign outfi[104      ]=fin[26   ][0    ];
assign outfi[105      ]=fin[26   ][1    ];
assign outfi[106      ]=fin[26   ][2    ];
assign outfi[107      ]=fin[26   ][3    ];
assign outfi[108      ]=fin[27   ][0    ];
assign outfi[109      ]=fin[27   ][1    ];
assign outfi[110      ]=fin[27   ][2    ];
assign outfi[111      ]=fin[27   ][3    ];
assign outfi[112      ]=fin[28   ][0    ];
assign outfi[113      ]=fin[28   ][1    ];
assign outfi[114      ]=fin[28   ][2    ];
assign outfi[115      ]=fin[28   ][3    ];
assign outfi[116      ]=fin[29   ][0    ];
assign outfi[117      ]=fin[29   ][1    ];
assign outfi[118      ]=fin[29   ][2    ];
assign outfi[119      ]=fin[29   ][3    ];
assign outfi[120      ]=fin[30   ][0    ];
assign outfi[121      ]=fin[30   ][1    ];
assign outfi[122      ]=fin[30   ][2    ];
assign outfi[123      ]=fin[30   ][3    ];
```

```
assign outfi[124      ]=fin[31   ][0   ];
assign outfi[125      ]=fin[31   ][1   ];
assign outfi[126      ]=fin[31   ][2   ];
assign outfi[127      ]=fin[31   ][3   ];
assign outfi[128      ]=fin[32   ][0   ];
assign outfi[129      ]=fin[32   ][1   ];
assign outfi[130      ]=fin[32   ][2   ];
assign outfi[131      ]=fin[32   ][3   ];
assign outfi[132      ]=fin[33   ][0   ];
assign outfi[133      ]=fin[33   ][1   ];
assign outfi[134      ]=fin[33   ][2   ];
assign outfi[135      ]=fin[33   ][3   ];
assign outfi[136      ]=fin[34   ][0   ];
assign outfi[137      ]=fin[34   ][1   ];
assign outfi[138      ]=fin[34   ][2   ];
assign outfi[139      ]=fin[34   ][3   ];
assign outfi[140      ]=fin[35   ][0   ];
assign outfi[141      ]=fin[35   ][1   ];
assign outfi[142      ]=fin[35   ][2   ];
assign outfi[143      ]=fin[35   ][3   ];
assign outfi[144      ]=fin[36   ][0   ];
assign outfi[145      ]=fin[36   ][1   ];
```

```
assign outfi[146      ]=fin[36   ][2    ];
assign outfi[147      ]=fin[36   ][3    ];
assign outfi[148      ]=fin[37   ][0    ];
assign outfi[149      ]=fin[37   ][1    ];
assign outfi[150      ]=fin[37   ][2    ];
assign outfi[151      ]=fin[37   ][3    ];
assign outfi[152      ]=fin[38   ][0    ];
assign outfi[153      ]=fin[38   ][1    ];
assign outfi[154      ]=fin[38   ][2    ];
assign outfi[155      ]=fin[38   ][3    ];
assign outfi[156      ]=fin[39   ][0    ];
assign outfi[157      ]=fin[39   ][1    ];
assign outfi[158      ]=fin[39   ][2    ];
assign outfi[159      ]=fin[39   ][3    ];
assign outfi[160      ]=fin[40   ][0    ];
assign outfi[161      ]=fin[40   ][1    ];
assign outfi[162      ]=fin[40   ][2    ];
assign outfi[163      ]=fin[40   ][3    ];
assign outfi[164      ]=fin[41   ][0    ];
assign outfi[165      ]=fin[41   ][1    ];
assign outfi[166      ]=fin[41   ][2    ];
assign outfi[167      ]=fin[41   ][3    ];
```

```
assign outfi[168      ]=fin[42   ][0    ];
assign outfi[169      ]=fin[42   ][1    ];
assign outfi[170      ]=fin[42   ][2    ];
assign outfi[171      ]=fin[42   ][3    ];
assign outfi[172      ]=fin[43   ][0    ];
assign outfi[173      ]=fin[43   ][1    ];
assign outfi[174      ]=fin[43   ][2    ];
assign outfi[175      ]=fin[43   ][3    ];
assign outfi[176      ]=fin[44   ][0    ];
assign outfi[177      ]=fin[44   ][1    ];
assign outfi[178      ]=fin[44   ][2    ];
assign outfi[179      ]=fin[44   ][3    ];
assign outfi[180      ]=fin[45   ][0    ];
assign outfi[181      ]=fin[45   ][1    ];
assign outfi[182      ]=fin[45   ][2    ];
assign outfi[183      ]=fin[45   ][3    ];
assign outfi[184      ]=fin[46   ][0    ];
assign outfi[185      ]=fin[46   ][1    ];
assign outfi[186      ]=fin[46   ][2    ];
assign outfi[187      ]=fin[46   ][3    ];
assign outfi[188      ]=fin[47   ][0    ];
assign outfi[189      ]=fin[47   ][1    ];
```

```
assign outfi[190    ]=fin[47  ][2   ];
assign outfi[191    ]=fin[47  ][3   ];
assign outfi[192    ]=fin[48  ][0   ];
assign outfi[193    ]=fin[48  ][1   ];
assign outfi[194    ]=fin[48  ][2   ];
assign outfi[195    ]=fin[48  ][3   ];
assign outfi[196    ]=fin[49  ][0   ];
assign outfi[197    ]=fin[49  ][1   ];
assign outfi[198    ]=fin[49  ][2   ];
assign outfi[199    ]=fin[49  ][3   ];
assign outfi[200    ]=fin[50  ][0   ];
assign outfi[201    ]=fin[50  ][1   ];
assign outfi[202    ]=fin[50  ][2   ];
assign outfi[203    ]=fin[50  ][3   ];
assign outfi[204    ]=fin[51  ][0   ];
assign outfi[205    ]=fin[51  ][1   ];
assign outfi[206    ]=fin[51  ][2   ];
assign outfi[207    ]=fin[51  ][3   ];
assign outfi[208    ]=fin[52  ][0   ];
assign outfi[209    ]=fin[52  ][1   ];
assign outfi[210    ]=fin[52  ][2   ];
assign outfi[211    ]=fin[52  ][3   ];
```

```
assign outfi[212      ]=fin[53   ][0   ];
assign outfi[213      ]=fin[53   ][1   ];
assign outfi[214      ]=fin[53   ][2   ];
assign outfi[215      ]=fin[53   ][3   ];
assign outfi[216      ]=fin[54   ][0   ];
assign outfi[217      ]=fin[54   ][1   ];
assign outfi[218      ]=fin[54   ][2   ];
assign outfi[219      ]=fin[54   ][3   ];
assign outfi[220      ]=fin[55   ][0   ];
assign outfi[221      ]=fin[55   ][1   ];
assign outfi[222      ]=fin[55   ][2   ];
assign outfi[223      ]=fin[55   ][3   ];
assign outfi[224      ]=fin[56   ][0   ];
assign outfi[225      ]=fin[56   ][1   ];
assign outfi[226      ]=fin[56   ][2   ];
assign outfi[227      ]=fin[56   ][3   ];
assign outfi[228      ]=fin[57   ][0   ];
assign outfi[229      ]=fin[57   ][1   ];
assign outfi[230      ]=fin[57   ][2   ];
assign outfi[231      ]=fin[57   ][3   ];
assign outfi[232      ]=fin[58   ][0   ];
assign outfi[233      ]=fin[58   ][1   ];
```

```
assign outfi[234      ]=fin[58   ][2    ];
assign outfi[235      ]=fin[58   ][3    ];
assign outfi[236      ]=fin[59   ][0    ];
assign outfi[237      ]=fin[59   ][1    ];
assign outfi[238      ]=fin[59   ][2    ];
assign outfi[239      ]=fin[59   ][3    ];
assign outfi[240      ]=fin[60   ][0    ];
assign outfi[241      ]=fin[60   ][1    ];
assign outfi[242      ]=fin[60   ][2    ];
assign outfi[243      ]=fin[60   ][3    ];
assign outfi[244      ]=fin[61   ][0    ];
assign outfi[245      ]=fin[61   ][1    ];
assign outfi[246      ]=fin[61   ][2    ];
assign outfi[247      ]=fin[61   ][3    ];
assign outfi[248      ]=fin[62   ][0    ];
assign outfi[249      ]=fin[62   ][1    ];
assign outfi[250      ]=fin[62   ][2    ];
assign outfi[251      ]=fin[62   ][3    ];
assign outfi[252      ]=fin[63   ][0    ];
assign outfi[253      ]=fin[63   ][1    ];
assign outfi[254      ]=fin[63   ][2    ];
assign outfi[255      ]=fin[63   ][3    ];
```

```
encoder_256 prrr(outfi,result,clk);
endmodule
```

priority4.v

```
module priority4bit(D,Y,V);
 input [3:0] D;
 output [3:0] Y;
 output V;

 assign Y[0]=1&D[0];
 assign Y[1]=1&D[1]&~D[0];
 assign Y[2]=1&D[2]&~D[0]&~D[1];
 assign Y[3]=1&D[3]&~D[0]&~D[1]&~D[2];

 assign V = D[0]|D[1]|D[2]|D[3];

endmodule
```

mux.v

```
module mux(a,b,d,e,f,c);
input [1:0]a;
input [255:0]b;
input [255:0]d;
```

```
input [255:0]e;
input [255:0]f;

output [255:0]c;
reg [255:0]c;

always@*
begin
	if(a[0] == 0 && a[1] == 0)
		c[255:0] = b[255:0];
	if(a[0] == 1 && a[1] == 0)
		c[255:0] = d[255:0];
	if(a[0] == 0 && a[1] == 1)
		c[255:0] = e[255:0];
	if(a[0] == 1 && a[1] == 1)
		c[255:0] = f[255:0];
end
endmodule
```

rca3.v

adder.v

Output:

Mapped the output on the FPGA Zedboard.