

**BITS-Pilani Dubai Campus**  
**I Sem 2021-22**  
**Digital Design Laboratory / ECE/INSTR/CS F215**  
**Submission Report**  
**Experiment No.- 5 (Comparators and Decoders)**

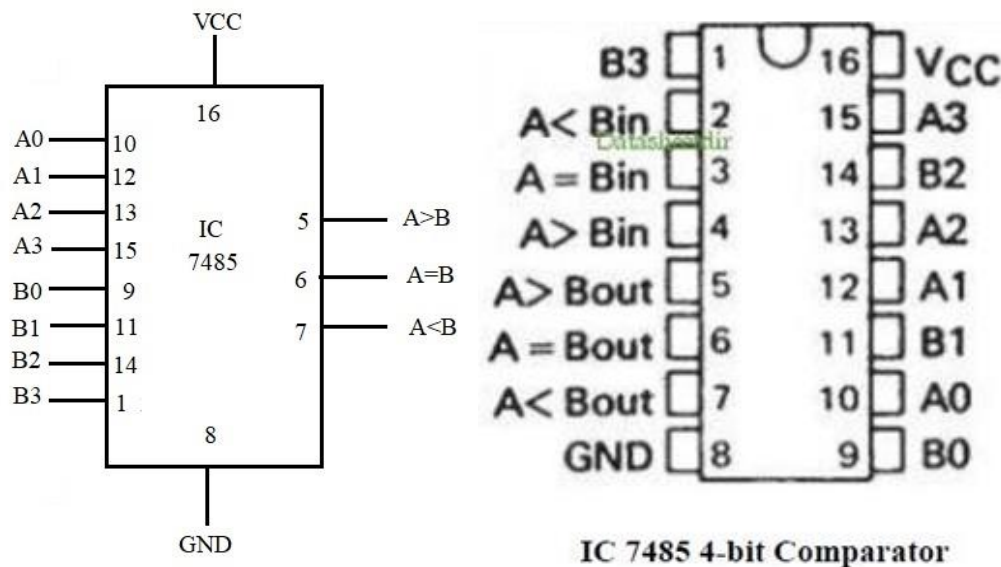
Name Anurag Kumar Jha

ID Number 2020A7PS0128U

**Hardware runs**

**Run 1: Comparator**

**Diagram**



**Truth Table**

| <b>A</b><br><b>(A3A2A1A0)</b> | <b>B</b><br><b>(B3B2B1B0)</b> | <b>A&gt;B</b> | <b>A=B</b> | <b>A&lt;B</b> |
|-------------------------------|-------------------------------|---------------|------------|---------------|
| 0000                          | 1111                          | 0             | 0          | 1             |
| 1111                          | 0000                          | 1             | 0          | 0             |
| 0000                          | 0000                          | 0             | 1          | 0             |
| 1111                          | 1111                          | 0             | 1          | 0             |
| 0101                          | 1000                          | 0             | 0          | 1             |
| 1010                          | 0011                          | 1             | 0          | 0             |
| 1001                          | 1001                          | 0             | 1          | 0             |

## Experiment No.- 5 (Comparators and Decoders)

Name Anurag Kumar Jha

ID Number 2020A7PS0128U

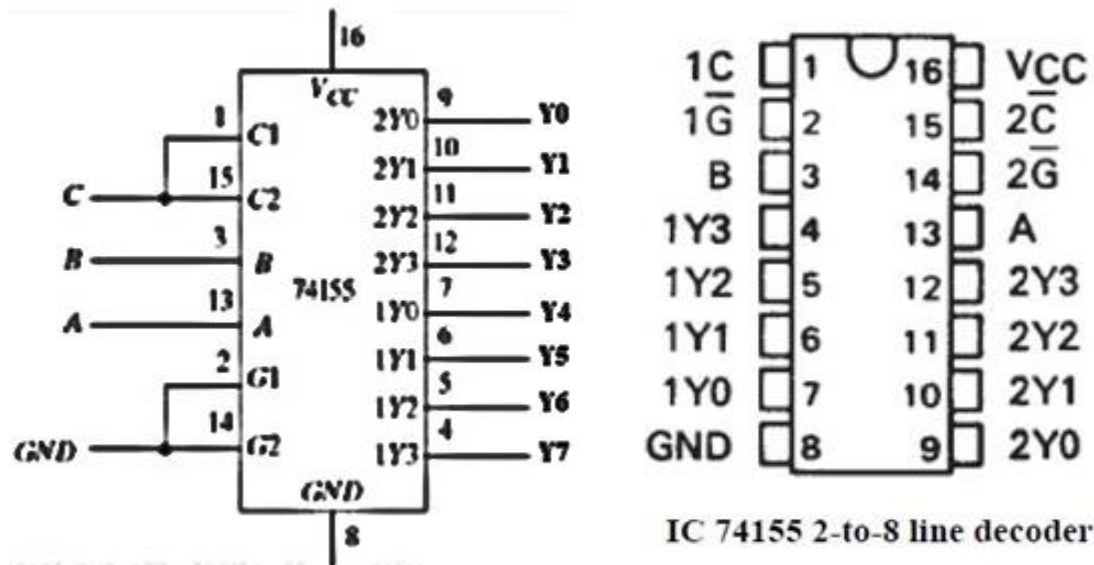
Q: Name the gates that can be used as one-bit comparators.

A: A combination of NOT gates, OR gates, and EXOR gate is used in a 1 bit comparator.

\*\*However, we can use NAND and NOR gates also as they are universal gates.

### Run 2: Decoder

#### Diagram



#### Truth Table

| C | B | A | Y0 | Y1 | Y2 | Y3 | Y4 | Y5 | Y6 | Y7 |
|---|---|---|----|----|----|----|----|----|----|----|
| 0 | 0 | 0 | 0  | 1  | 1  | 1  | 1  | 1  | 1  | 1  |
| 0 | 0 | 1 | 1  | 0  | 1  | 1  | 1  | 1  | 1  | 1  |
| 0 | 1 | 0 | 1  | 1  | 0  | 1  | 1  | 1  | 1  | 1  |
| 0 | 1 | 1 | 1  | 1  | 1  | 0  | 1  | 1  | 1  | 1  |
| 1 | 0 | 0 | 1  | 1  | 1  | 1  | 0  | 1  | 1  | 1  |
| 1 | 0 | 1 | 1  | 1  | 1  | 1  | 1  | 0  | 1  | 1  |
| 1 | 1 | 0 | 1  | 1  | 1  | 1  | 1  | 1  | 0  | 1  |
| 1 | 1 | 1 | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 0  |

Q: Are the outputs of the decoder active low or active high?

A: Active low

## Experiment No.- 5 (Comparators and Decoders)

Name Anurag Kumar Jha

ID Number 2020A7PS0128U

**Q: What external gate would have been used if the IC were to be active high?**

**A:** We would use NOT gates to get the complements of the outputs (outputs are currently in active low state, if we use an external NOT gate, we will get the outputs in active low states)

### Software runs

#### Run 3: Comparator

1. Write the Verilog code and testbench for 4-bit comparator using data flow modeling. (Hint: Use >, < and = to compare the two numbers). Complete the truth table.

**A: Verilog Code and testbench-**

```
module run3 (l,g,e,a,b);
input[3:0] a,b;
output l,g,e;
assign l = a<b;
assign g = a>b;
assign e = (a==b);
endmodule

module testbench();
reg[3:0] a,b;
wire l,g,e;
initial begin
#000 a = 4'b0000; b = 4'b0000;
#100 a = 4'b0000; b = 4'b1111;
#100 a = 4'b1111; b = 4'b0000;
#100 a = 4'b0000; b = 4'b0000;
#100 a = 4'b1111; b = 4'b1111;
#100 a = 4'b0010; b = 4'b0010;
#100 $stop;
end
run3 g1 (l,g,e,a,b);
endmodule
```

### Truth Table

| <b>A</b><br><b>(A3A2A1A0)</b> | <b>B</b><br><b>(B3B2B1Bo)</b> | <b>A&gt;B</b><br><b>G</b> | <b>A=B</b><br><b>E</b> | <b>A&lt;B</b><br><b>L</b> |
|-------------------------------|-------------------------------|---------------------------|------------------------|---------------------------|
| 0000                          | 1111                          | 0                         | 0                      | 1                         |
| 1111                          | 0000                          | 1                         | 0                      | 0                         |
| 0101                          | 1000                          | 0                         | 0                      | 1                         |
| 1010                          | 0011                          | 1                         | 0                      | 0                         |
| 1001                          | 1001                          | 0                         | 1                      | 0                         |

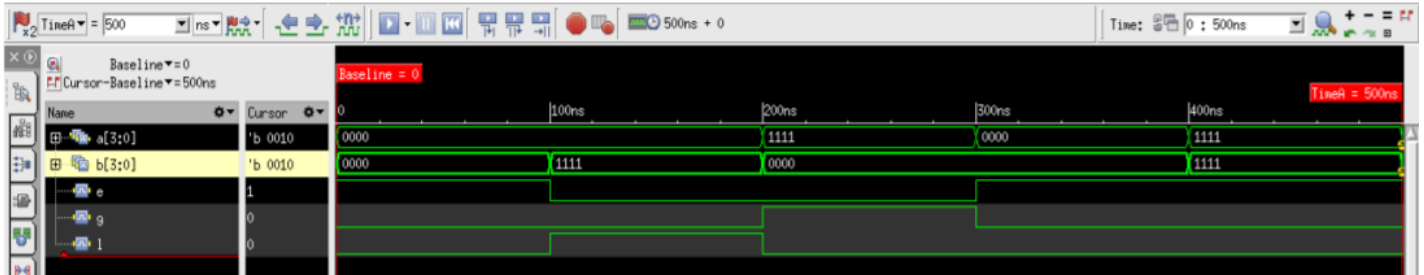
## Experiment No.- 5 (Comparators and Decoders)

Name Anurag Kumar Jha

ID Number 2020A7PS0128U

**Q:** Paste the screenshot of waveform window where you get the waveforms for the above code

**A:**



### Run 4: Encoders and Decoders

1. Write the Verilog code and testbench for 2:4 decoder using data flow modeling, with active high output. (Hint: part of code is written below; A is input and D is output.). Complete the truth table.

**A: Verilog Code and testbench-**

```
module run41 (d,a );
input[1:0] a;
output[3:0] d;
assign d[0] = (~a[1]&~a[0]);
assign d[1] = (~a[1]&a[0]);
assign d[2] = (a[1]&~a[0]);
assign d[3] = (a[1]&a[0]);
endmodule

module testbench();
reg[1:0] a;
wire[3:0] d;
initial begin
#000 a = 2'b00;
#100 a = 2'b01;
#100 a = 2'b10;
#100 a = 2'b11;
#100;
#100 $stop;
end
run41 g1 (d,a );
endmodule
```

## Experiment No.- 5 (Comparators and Decoders)

Name Anurag Kumar Jha

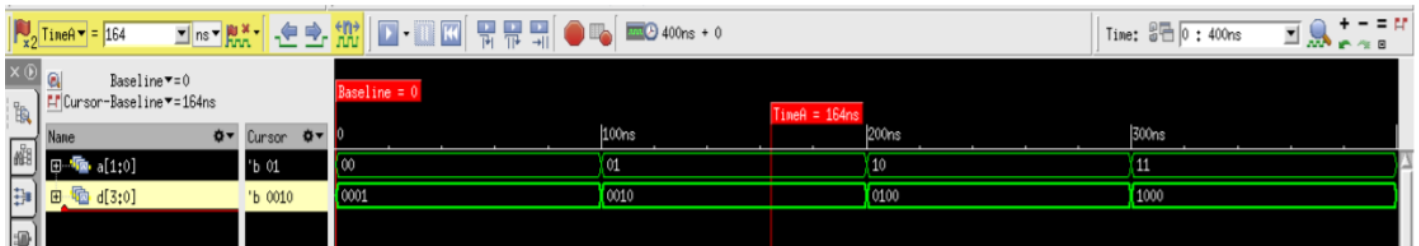
ID Number 2020A7PS0128U

### Truth Table

| A1 | A0 | D0 | D1 | D2 | D3 |
|----|----|----|----|----|----|
| 0  | 0  | 1  | 0  | 0  | 0  |
| 0  | 1  | 0  | 1  | 0  | 0  |
| 1  | 0  | 0  | 0  | 1  | 0  |
| 1  | 1  | 0  | 0  | 0  | 1  |

**Q:** Paste the screenshot of waveform window where you get the waveforms for the above code.

**A:**



2. Write the verilog code and testbench for 8-bit priority encoder using behavioral modeling. (Hint: Use **if**, **else** and **else if** statements). Complete the truth table.

**A: Verilog Code and testbench-**

```
module run44 (input [7:0] sel, output [2:0] code);
reg [2:0] code;
always @(sel)
begin
if (sel[0]) code = 3'b000;
else if (sel[1]) code = 3'b001;
else if (sel[2]) code = 3'b010;
else if (sel[3]) code = 3'b011;
else if (sel[4]) code = 3'b100;
else if (sel[5]) code = 3'b101;
else if (sel[6]) code = 3'b110;
else if (sel[7]) code = 3'b111;
else code = 3'bxxx;
end
endmodule
module testbench();
reg [7:0] e;
wire [2:0] f;
initial begin
#000 e = 8'b00000000;

```

## Experiment No.- 5 (Comparators and Decoders)

Name Anurag Kumar Jha

ID Number 2020A7PS0128U

```
#100 e = 8'b00000001;  
#100 e = 8'b00000010;  
#100 e = 8'b00000100;  
#100 e = 8'b00001000;  
#100 e = 8'b00010000;  
#100 e = 8'b00100000;  
#100 e = 8'b01000000;  
#100 e = 8'b10000000;  
#100;  
#100 $stop;  
end  
run44 g1 (e, f);  
endmodule
```

### Truth Table

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | Y2 | Y1 | Y0 |
|----|----|----|----|----|----|----|----|----|----|----|
| 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 0  |
| 0  | 0  | 0  | 0  | 0  | 0  | 1  | X  | 0  | 0  | 1  |
| 0  | 0  | 0  | 0  | 0  | 1  | X  | X  | 0  | 1  | 0  |
| 0  | 0  | 0  | 0  | 1  | X  | X  | X  | 0  | 1  | 1  |
| 0  | 0  | 0  | 1  | X  | X  | X  | X  | 1  | 0  | 0  |
| 0  | 0  | 1  | x  | X  | X  | X  | X  | 1  | 0  | 1  |
| 0  | 1  | X  | X  | X  | X  | X  | X  | 1  | 1  | 0  |
| 1  | X  | X  | X  | X  | X  | X  | X  | 1  | 1  | 1  |

**Q:** Paste the screenshot of waveform window where you get the waveforms for the above code

**A:**



## Experiment No.- 5 (Comparators and Decoders)

Name Anurag Kumar Jha

ID Number 2020A7PS0128U

**Assignment** All assignments are to be submitted strictly before start of next lab session through online only. Late assignments will not be entertained and will be awarded '0' marks.

1. Verilog code and testbench for 3-to-8 decoder using structural modeling (use 2-to-4 decoders as blocks).

**Ans: Link1:** <https://www.edaplayground.com/x/uUBy>

2. Write the Verilog code and testbench for 8:3 encoder using behavioral modeling. (Hint: use **case** statements)

**Ans: Link2:** <https://www.edaplayground.com/x/9Gu6>

3. Write the verilog code and testbench for 2:4 decoder using data flow modeling, with active high output and active low enable pin. (Hint: use 1-bit enable input pin in all the assign statements like  $D[3] = (A \& B \& \sim En)$ ).

**Ans: Link3:** <https://www.edaplayground.com/x/dMPA>

4. Write the verilog code and testbench for 4-bit (4-to-2) encoder using behavioral modeling. (Hint: Part of code is given below).

```
module encoder4_to_2 (input [3:0] d_in, output [1:0] d_out);
    reg [1:0] d_out;
    always @(d_in)
        case (d_in)
            4'b0001 : d_out = 2'b00;
            4'b0010 : d_out = 2'b01;
            4'b0100 : d_out = 2'b10;
            4'b1000 : d_out = 2'b11;
        endcase
endmodule
```

**Ans: Link4:** <https://www.edaplayground.com/x/cpbm>

### Self-Practice and self-evaluation

1. Name the gates that can be used as one-bit comparators.
2. What external gate would have been used if the IC were to be active high?
3. Are the outputs of the decoder active low or active high?
4. Find out about the functionality of 74138 and 74139 decoder ICs?
5. If the decoder had been internally constructed using AND gates, what external gate would be required to use the decoder as a Boolean function implementer.
6. Define synthesis. What is the need for synthesis?
7. Why comparators are called iterative circuits? Name some other iterative circuits.
8. How can magnitude comparators be used in memory decoding?
9. Write the behavioral description of a 4-bit magnitude comparator in Verilog.