

## 8<sup>th</sup> Day

### Universal Shift Register

Shift registers play a crucial role in digital circuits by **moving data in a specific direction**—either left or right—based on control signals. They are widely used in data storage, transmission, and arithmetic operations.

#### Types of Shifts Implemented:

- Linear Left Shift (Logical Shift Left - LSL)**: Shifts bits to the left and inserts '0' at the LSB.
- Linear Right Shift (Logical Shift Right - LSR)**: Shifts bits to the right and inserts '0' at the MSB.
- Circular Left Shift (Rotate Left - ROL)**: Rotates bits to the left, with MSB moving to the LSB position.
- Circular Right Shift (Rotate Right - ROR)**: Rotates bits to the right, with LSB moving to the MSB position.

#### Working Mechanism:

- **Enable (en)**: Shifting occurs only when enabled.
- **Control Signals (lrbar, lcbar)**: Decide whether the shift is linear or circular.
- **Reset (rst)**: Resets the register to a known state.

### Design

```
module univ_shift_register(input clk,rst,en,lrbar,lcbar,input [3:0]in,output reg[3:0]y );  
always @(posedge clk)  
begin  
    if(rst)  
        y<=in;  
    else  
        begin  
            if(en)  
                begin
```

```
        case({lcbar,lrbar})
            2'b00:y<={y[0],y[3:1]};
            2'b01:y<={y[2:0],y[3]};
            2'b10:y<={1'b0,y[3:1]};
            2'b11:y<={y[2:0],1'b0};
        endcase
    end
    else
        y<=4'd0;
    end
end
endmodule
```

## Testbench

```
module univ_shift_tb();
reg clk,rst,en,lcbar,lrbar;
reg [3:0]in;
wire [3:0]y;
univ_shift_register dut(clk,rst,en,lrbar,lcbar,in,y);
always #5 clk=~clk;
initial
begin
clk=0;
rst=1;
in=4'b0110;
#20;
rst=0;
en=1;
repeat(10)
begin
```

```
{lcbar,lrbar}=$random;
```

```
#5;
```

```
end
```

```
rst=1;
```

```
in=4'b1100;
```

```
#20;
```

```
rst=0;
```

```
end
```

```
initial
```

```
begin
```

```
#200; $finish;
```

```
end
```

```
endmodule
```

## Simulation

