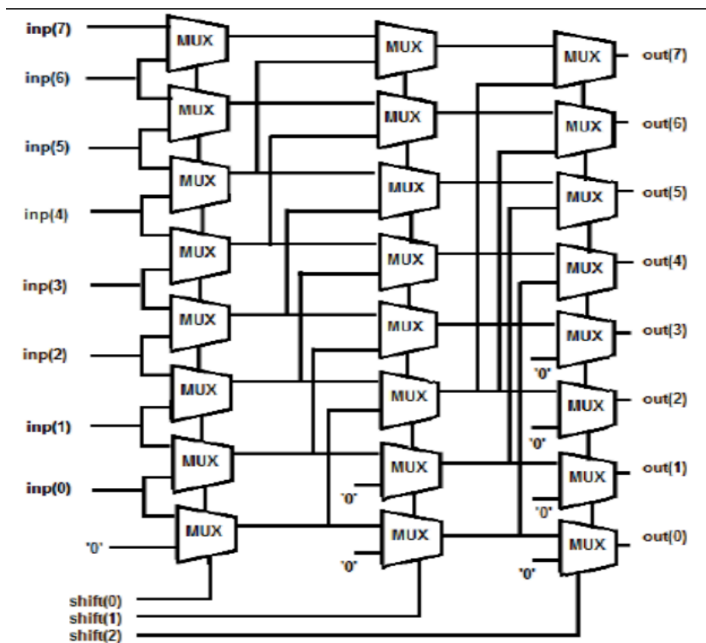


**KARTHICK RAJA B**  
**126004121**

**Given:**

Design a 2:1 multiplexer using Conditional select operator and use it as a component to build the following architecture.



**Verilog code:**

```
module RTL_Day_12(inp,shift,out);
input[7:0] inp;
input[2:0] shift;
output [7:0] out ;
wire [7:0] stage1, stage2;

// Stage 1: Shift by 1 bit if shift(0) is set
assign stage1 = shift[0] ? {inp[6:0], 1'b0} : inp;

// Stage 2: Shift by 2 bits if shift(1) is set
assign stage2 = shift[1] ? {stage1[5:0], 2'b00} : stage1;

// Stage 3: Shift by 4 bits if shift(2) is set
assign out = shift[2] ? {stage2[3:0], 4'b0000} : stage2;
```

endmodule

### **Testbench code:**

```
module test_bench_tb12();
```

```
reg[7:0] inp;
```

```
reg[2:0] shift;
```

```
wire[7:0] out;
```

```
wire[7:0] stage1,stage2;
```

```
RTL_Day_12 m12(inp,shift,out);
```

```
initial
```

```
begin
```

```
inp = 8'b10110011; shift = 3'b000; #100; // No shift
```

```
inp = 8'b10110011; shift = 3'b001; #100; // Shift left by 1
```

```
inp = 8'b10110011; shift = 3'b010; #100; // Shift left by 2
```

```
inp = 8'b10110011; shift = 3'b011; #100; // Shift left by 3
```

```
inp = 8'b10110011; shift = 3'b100; #100; // Shift left by 4
```

```
inp = 8'b10110011; shift = 3'b101; #100; // Shift left by 5
```

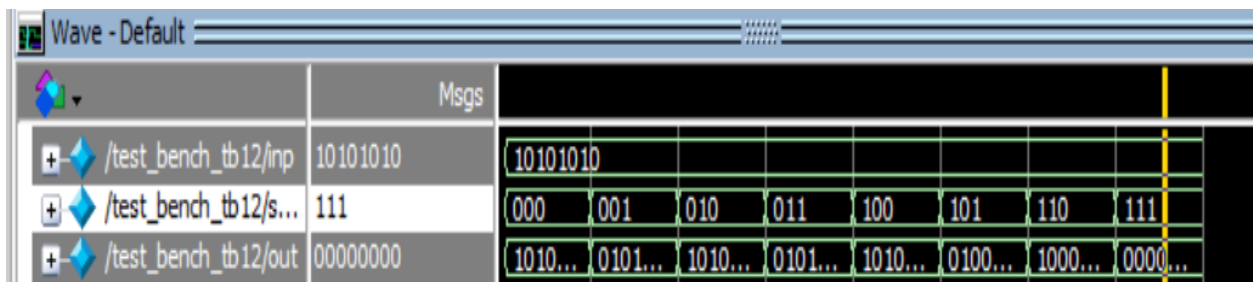
```
inp = 8'b10110011; shift = 3'b110; #100; // Shift left by 6
```

```
inp = 8'b10110011; shift = 3'b111; #100; // Shift left by 7
```

```
end
```

```
endmodule
```

### **Functional Simulation:**



The screenshot displays a functional simulation interface with a 'Wave - Default' window. It shows three signals: `/test_bench_tb12/inp`, `/test_bench_tb12/s...`, and `/test_bench_tb12/out`. The `inp` signal is a constant 8-bit value 10101010. The `s...` signal is a 3-bit shift register that cycles through values 111, 000, 001, 010, 011, 100, 101, 110, and 111. The `out` signal is an 8-bit output that shows the result of shifting the input by the specified amount. The output sequence is 1010..., 0101..., 1010..., 0101..., 1010..., 0100..., 1000..., and 0000....

Signal	Value
<code>/test_bench_tb12/inp</code>	10101010
<code>/test_bench_tb12/s...</code>	111
<code>/test_bench_tb12/out</code>	00000000