NAME : SHAH RAZA

REG NO: 18PWCSE1658

SECTION: B

ASSIGNMENT # 4

SUBJECT : DSD

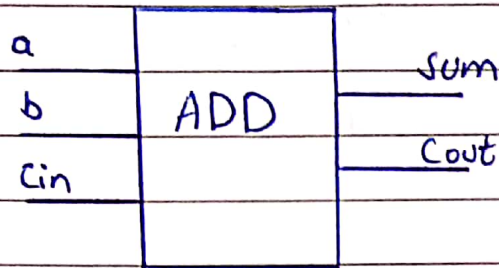## PROBLEM #1:

a)

Solution:

```
        a  ┌───────┐  sum
        b  │  ADD  │
       Cin │       │  Cout
           └───────┘
```

b)

Solution:

```
      I ──────┌──────┐
              │  FF  │────O
      E ────▷ │      │
              └──────┘
```

c)

Solution:

```
      I ──────┌─────────────┐──── O1
              │ FF_blocking │
      E ────▷ │             │──── O2
              └─────────────┘
```

d)

Solution:

I ──── | FF_non_blocking | ──── O1

E ────▷ | | ──── O2

e)

Solution:

I ──── | FF_reset |

C ────▷ | | ──── O
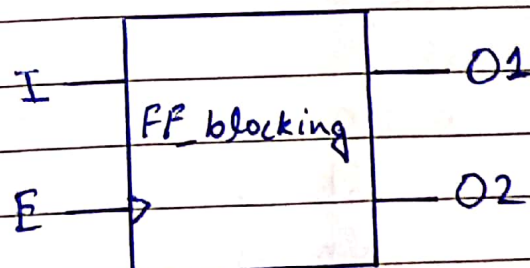
Rn ───▷ | |
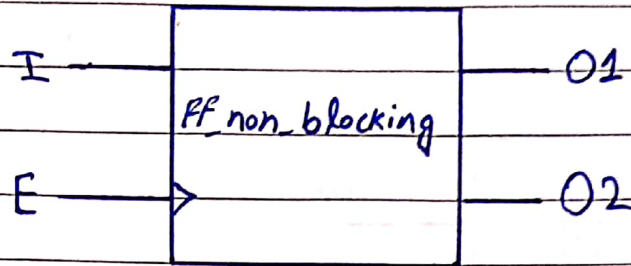
PROBLEM #2:

Solution:

```
module    clk_gen;
    reg  clk;
    initial
        begin
            clk = 0;
        always
            begin
                #10  clk = 1;
                #3   clk = 0;
            end
        end
endmodule
```

Name, Shah Raza

PROBLEM #3:

Solution:

```verilog
module   shift_left ( CLK, SI, SO, CLR, SHIFT);
         input  CLK, SI, CLR, SHIFT;
         output    SO;

         reg [7:0]  SHIFTER;
         reg  SO;

         always  @( posedge CLK)
                if (SHIFT)
                  begin
                        SO = SHIFTER[7];
                        SHIFTER = { SHIFTER[6:0], 1'b0};
                  end
                else
                        SHIFTER[0] = SI;

         always  @ (CLR)
                if (CLR)
                        SHIFTER = 8'h0;

endmodule
```

PROBLEM #4:

Solution:

```verilog
module    UD_Counter(CLK, RESET, UP_DN,
                            LD, DIN, Q);
    input  CLK, RESET, UP_DN, LD;
    input [2:0] DIN;
    output [2:0] Q;
    reg  [2:0] Q;

    always @(posedge CLK)
            if(UP_DN)
                    Q = Q-1;
            else
                    Q = Q+1;

    always @(LD, RESET)
            if(!RESET)
                    Q = 3'b011;
            else if(LD)
                    Q = DIN;
endmodule
```

PROBLEM # 5:

a)

b)
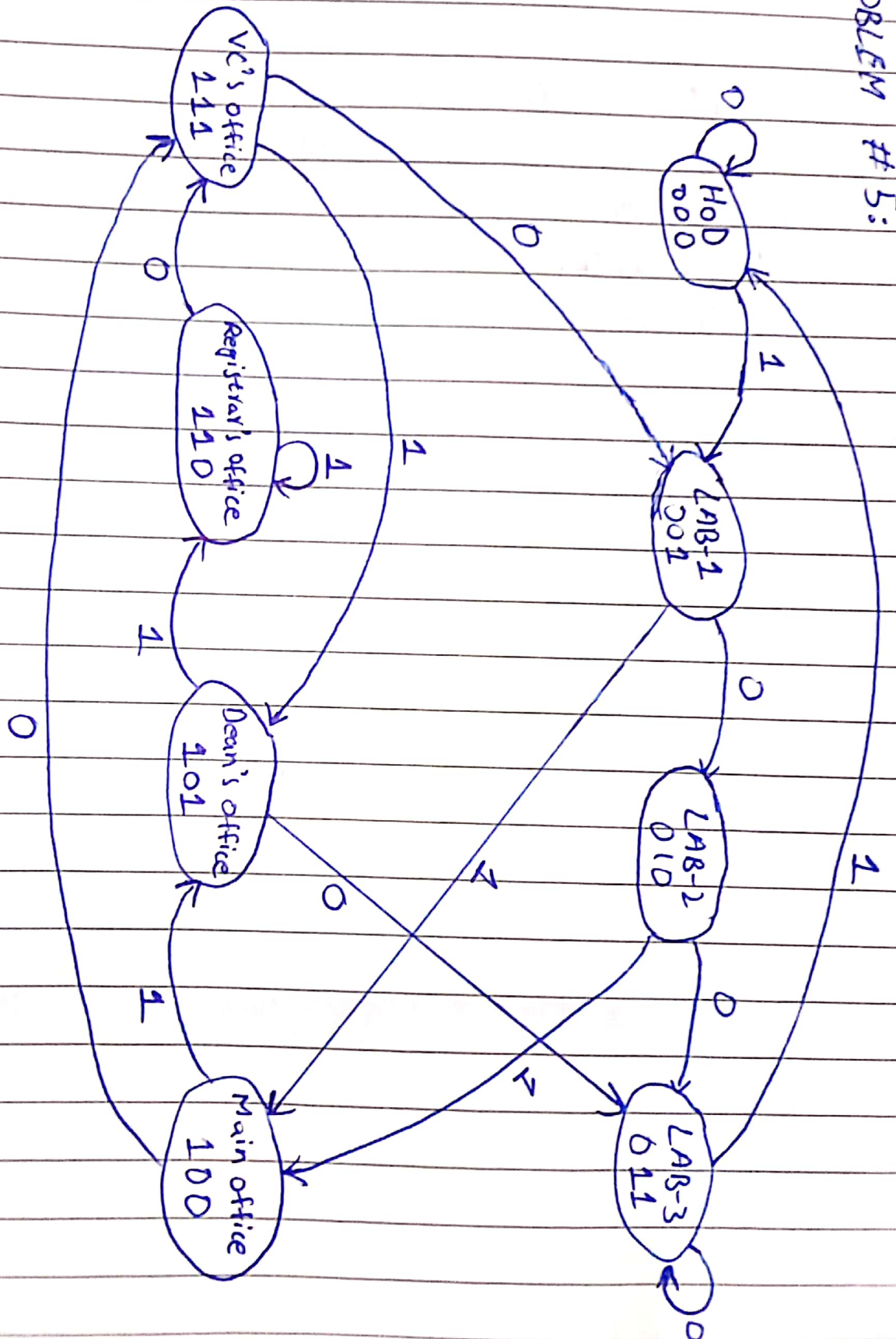
Solution:

```verilog
module   Peerzada_movement(CLK, RESET, MOVE, STATE);
   input   CLK, RESET, MOVE;
   output  [2:0]  STATE;

   reg [2:0] STATE;
   reg [2:0] NEXT_STATE;

   parameter [2:0]  A = 3'h0 , B = 3'h1, C = 3'h2
                    D = 3'h3, E = 3'h4, F = 3'h5
                    G = 3'h6, H = 3'h7 ;

   always @(posedge CLK)
         STATE = NEXT_STATE;

   always @(STATE, RESET)
         if (RESET)
            STATE = A;
         if (MOVE)
            case (STATE)
               A:  NEXT_STATE = B;
               B:  NEXT_STATE = E;
               C:  NEXT_STATE = E;
               D:  NEXT_STATE = A;
               E:  NEXT_STATE = F;
               F:  NEXT_STATE = G;
               G:  NEXT_STATE = G;
               H:  NEXT_STATE = F;


            endcase
```

```verilog
                                else
                                    case (STATE)
                                        A: NEXT_STATE = A;
                                        B: NEXT_STATE = C;
                                        C: NEXT_STATE = D;
                                        D: NEXT_STATE = D;
                                        E: NEXT_STATE = H;
                                        F: NEXT_STATE = D;
                                        G: NEXT_STATE = H;
                                        H: NEXT_STATE = B;
                                    endcase

endmodule
```