```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 11/12/2020 03:57:20 PM
// Design Name:
// Module Name: Control_Logic
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module Control_Logic(
    input Left, Right,
    input [6:0] PS,
    output [6:0] NS,
    output CountUp, CountDw, L_LED, R_LED, Reset, On
    );

    wire IDLE, L, L_B, LR, R, R_B, RL;
    wire Next_IDLE, Next_L, Next_L_B, Next_LR, Next_R, Next_R_B, Next_RL;

    // Present State
    assign IDLE     = PS[0];
    assign L        = PS[1];
    assign L_B      = PS[2];
    assign LR       = PS[3];
    assign R        = PS[4];
    assign R_B      = PS[5];
    assign RL       = PS[6];

    // Next State
    assign NS[0] = Next_IDLE;
    assign NS[1] = Next_L;
    assign NS[2] = Next_L_B;
    assign NS[3] = Next_LR;
```

```
    assign NS[4] = Next_R;
    assign NS[5] = Next_R_B;
    assign NS[6] = Next_RL;

    //Enter Logic
    assign Next_IDLE    = (~Left&~Right&(IDLE|L|LR|R|RL));
    assign Next_L       = (Left&~Right&(IDLE|L|L_B));
    assign Next_L_B     = (Left&Right&(L|L_B|LR));
    assign Next_LR      = (~Left&Right&(L_B|LR));
    assign Next_R       = (~Left&Right&(IDLE|R|R_B));
    assign Next_R_B     = (Left&Right&(R|R_B|RL));
    assign Next_RL      = (Left&~Right&(R_B|RL));

    // Outputs
    assign Reset    = IDLE;
    assign CountUp  = LR&~Right&~Left;
    assign CountDw  = RL&~Left&~Right;
    assign L_LED    = ~Left;
    assign R_LED    = ~Right;
    assign On       = Left|Right;

endmodule
```