

```
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 btnC, btnL, btnD, Prime, NPrime,
    input [4:0] PS,
    output [4:0] NS,
    output LED_Input, LED_nprime, LED_prime, Show_In, Show_Div, Flash, Start
);

    wire INPUT, WORKING, PRIME, NPRIME, DIVISOR;
    wire Next_INPUT, Next_WORKING, Next_PRIME, Next_NPRIME, Next_DIVISOR;

    // Present State
    assign INPUT      = PS[0];
    assign WORKING    = PS[1];
    assign PRIME      = PS[2];
    assign NPRIME     = PS[3];
    assign DIVISOR    = PS[4];

    // Next State
    assign NS[0] = Next_INPUT;
    assign NS[1] = Next_WORKING;
    assign NS[2] = Next_PRIME;
    assign NS[3] = Next_NPRIME;
    assign NS[4] = Next_DIVISOR;
```

```
//Enter Logic
assign Next_INPUT    = (btnC&(PRIME|NPRIME)) | (~btnL&INPUT);
assign Next_WORKING  = (btnL&INPUT) | (WORKING&~(NPrime|Prime));
assign Next_PRIME    = (Prime&WORKING) | (~btnC&PRIME);
assign Next_NPRIME   = (NPrime&WORKING) | (~btnD&DIVISOR) | (~btnC&NPRIME);
assign Next_DIVISOR  = (btnD&(NPRIME|DIVISOR));
```

```
// Outputs
assign LED_Input     = INPUT;
assign LED_prime     = PRIME;
assign LED_nprime    = NPRIME;
assign Show_In       = INPUT;
assign Show_Div      = DIVISOR;
assign Flash         = NPRIME|DIVISOR;
assign Start         = WORKING;
```

```
endmodule
```