

```
timescale 1ns / 1ps
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 11/07/2018 03:52:05 PM
// Design Name:
// Module Name: StateMachine
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
module StateMachine(
    input clk,
    input btnC,
    input btnU,
    input btnD,
    input TimeUp,
    input lastled,
    output ShowTime,
    output LoadTime,
    output RunTime,
    output IncU,
    output IncD,
    output ShowScore,
    output FlashU,
    output FlashD,
    output shift
);
    wire[11:0] D, Q;
    wire enable = 1'b1;
    wire btnCsyn,btnDsyn,btnUsyn;
    assign btnCsyn = btnC;
    assign btnDsyn = btnD;
    assign btnUsyn = btnU;
```

```
//      FDRE #(.INIT(1'b0)) syc0 (.C(clk), .CE(enable), .D(btnC), .Q(btnCsyn));
//      FDRE #(.INIT(1'b0)) syc1 (.C(clk), .CE(enable), .D(btnD), .Q(btnDsyn));
//      FDRE #(.INIT(1'b0)) syc2 (.C(clk), .CE(enable), .D(btnU), .Q(btnUsyn));

//Idle
assign D[0] = Q[0]&~btnCsyn | Q[2]&TimeUp&~btnDsyn&~btnUsyn;
FDRE #(.INIT(1'b1)) t0 (.C(clk), .CE(enable), .D(D[0]), .Q(Q[0]));

//wait
assign D[1] = Q[11] | Q[1]&~lastled;
FDRE #(.INIT(1'b0)) t1 (.C(clk), .CE(enable), .D(D[1]), .Q(Q[1]));

//Accp
assign D[2] = Q[2]&~btnUsyn&~btnDsyn&~TimeUp | Q[1]&lastled;
FDRE #(.INIT(1'b0)) t2 (.C(clk), .CE(enable), .D(D[2]), .Q(Q[2]));

//TIE
assign D[3] = Q[2]&~TimeUp&btnDsyn&btnUsyn;
FDRE #(.INIT(1'b0)) t3 (.C(clk), .CE(enable), .D(D[3]), .Q(Q[3]));

//UFirst
assign D[4] = Q[2]&~TimeUp&~btnDsyn&btnUsyn | Q[4]&~TimeUp&~btnDsyn;
FDRE #(.INIT(1'b0)) t4 (.C(clk), .CE(enable), .D(D[4]), .Q(Q[4]));

//Dfirst
assign D[5] = Q[2]&~TimeUp&btnDsyn&~btnUsyn | Q[5]&~TimeUp&~btnUsyn;
FDRE #(.INIT(1'b0)) t5 (.C(clk), .CE(enable), .D(D[5]), .Q(Q[5]));

//Uwin
assign D[6] = Q[5]&~TimeUp&btnUsyn | Q[4]&TimeUp&~btnDsyn;
FDRE #(.INIT(1'b0)) t6 (.C(clk), .CE(enable), .D(D[6]), .Q(Q[6]));

//Dwin
assign D[7] = Q[4]&~TimeUp&btnDsyn | Q[5]&TimeUp&~btnUsyn;
FDRE #(.INIT(1'b0)) t7 (.C(clk), .CE(enable), .D(D[7]), .Q(Q[7]));

//UFlash
assign D[8] = Q[6] | Q[8]&~btnCsyn;
FDRE #(.INIT(1'b0)) t8 (.C(clk), .CE(enable), .D(D[8]), .Q(Q[8]));

//Dflash
assign D[9] = Q[7] | Q[9]&~btnCsyn;
FDRE #(.INIT(1'b0)) t9 (.C(clk), .CE(enable), .D(D[9]), .Q(Q[9]));

//Tflash
```

```

assign D[10] = Q[3] | Q[10]&~btnCsyn;
FDRE #(.INIT(1'b0)) t10 (.C(clk), .CE(enable), .D(D[10]), .Q(Q[10]));

//Load
assign D[11] = (Q[0]|Q[8]|Q[9]|Q[10])&btnCsyn ;
FDRE #(.INIT(1'b0)) t11 (.C(clk), .CE(enable), .D(D[11]), .Q(Q[11]));

assign ShowTime = Q[1];
assign LoadTime = Q[11];
assign RunTime = Q[2]| Q[4]| Q[5];
assign IncU = Q[3]| Q[6];
assign IncD = Q[3]| Q[7];
assign ShowScore = Q[0]| Q[8]| Q[9]| Q[10];
assign FlashU = Q[8]| Q[10];
assign FlashD = Q[9]| Q[10];
assign shift = Q[1];

```

```

endmodule

```