

ECEN 220

Jae Lee

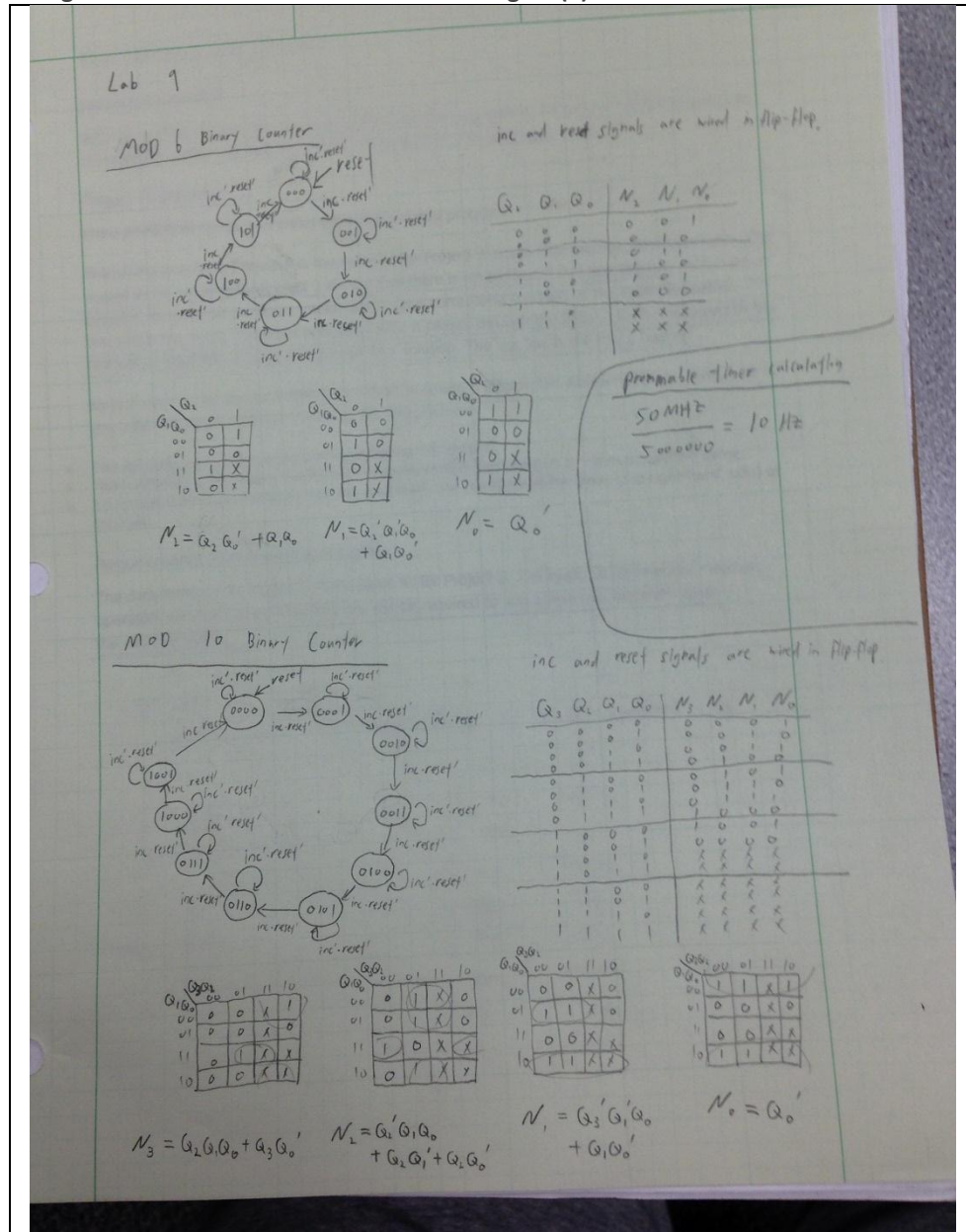
Lab #9 – Counter, Stopwatch

11/13/2013

MOD6 state graph, truth table, k-maps, and equations (4)

MOD10 state graph, truth table, k-maps, and equations (4)

Programmable timer calculations for 10Hz signal (2)



SR latch Verilog module, simulation, and TCL file (3)

module SRLatch(q, qbar, r, s);

output q, qbar;

```

        input r, s;

        nor(q, r, qbar);
        nor(qbar, s, q);

    endmodule

```



```

wave add / -radix hex

isim force add r 0 -time 0 -value 1 -time 10ns -value 0 -time 30ns -value 1 -time 80ns
isim force add s 0 -time 0 -value 1 -time 40ns -value 0 -time 60ns

run 120ns

```

MOD6 Verilog module (3)

```

module MOD6(q, rollover, inc, clk, clr);

    output[2:0] q;
    output rollover;
    input inc, clk, clr;
    wire[2:0] n;
    assign n[2] = (q[2] & ~q[0]) | (q[1] & q[0]);
    assign n[1] = (~q[2] & ~q[1] & q[0]) | (q[1] & ~q[0]);
    assign n[0] = (~q[0]);
    assign rollover = (inc & ~clr & q[2] & ~q[1] & q[0]);

    FF_DCE ff2(q[2], clk, n[2], clr, inc);
    FF_DCE ff1(q[1], clk, n[1], clr, inc);
    FF_DCE ff0(q[0], clk, n[0], clr, inc);

endmodule

```

MOD6 TCL file (1)

```

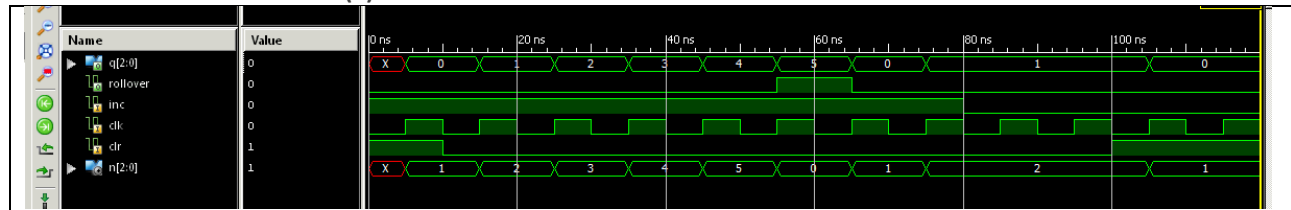
wave add / -radix hex

isim force add clk 0 -time 0 -value 1 -time 5ns -repeat 10ns
isim force add inc 1 -time 0 -value 0 -time 80ns
isim force add clr 1 -time 0 -value 0 -time 10ns -value 1 -time 100ns

run 120ns

```

MOD6 simulation waveform (2)



MOD10 Verilog module (3)

```
module MOD10(q, rollover, inc, clk, clr);
```

```
    output[3:0] q;
    output rollover;
    input inc, clk, clr;
    wire[3:0] n;
    assign n[3] = (q[2] & q[1] & q[0]) | (q[3] & ~q[0]);
    assign n[2] = (~q[2] & q[1] & q[0]) | (q[2] & ~q[1]) | (q[2] & ~q[0]);
    assign n[1] = (~q[3] & ~q[1] & q[0]) | (q[1] & ~q[0]);
    assign n[0] = (~q[0]);
    assign rollover = (inc & ~clr & q[3] & ~q[2] & ~q[1] & q[0]);
```

```
    FF_DCE ff3(q[3], clk, n[3], clr, inc);
    FF_DCE ff2(q[2], clk, n[2], clr, inc);
    FF_DCE ff1(q[1], clk, n[1], clr, inc);
    FF_DCE ff0(q[0], clk, n[0], clr, inc);
```

```
endmodule
```

MOD10 TCL file (1)

```
wave add / -radix hex
```

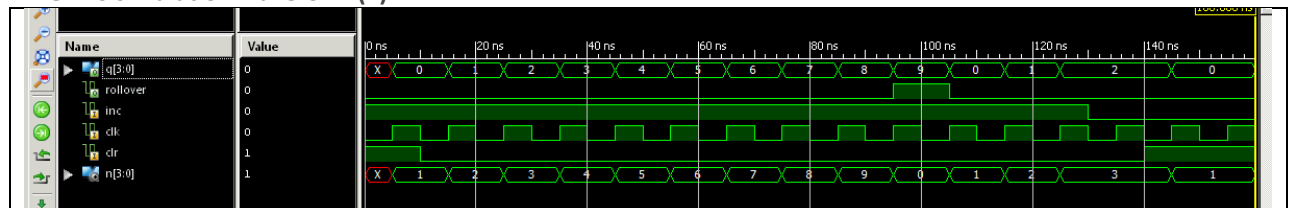
```
isim force add clk 0 -time 0 -value 1 -time 5ns -repeat 10ns
```

```
isim force add inc 1 -time 0 -value 0 -time 130ns
```

```
isim force add clr 1 -time 0 -value 0 -time 10ns -value 1 -time 140ns
```

```
run 160ns
```

MOD10 simulation waveform (2)



Counter block Verilog module (3)

```
module counterblock(q, rollover, clk, reset, cei);
```

```
    output[15:0] q;
```

```

output rollover;
input clk, reset, cei;
assign q[11] = 1'b0;
wire rollover1, rollover2, rollover3;

```

```

MOD10 mymod101(q[15:12], rollover, rollover3, clk, reset);
MOD6 mymod61(q[10:8], rollover3, rollover2, clk, reset);
MOD10 mymod102(q[7:4], rollover2, rollover1, clk, reset);
MOD10 mymod103(q[3:0], rollover1, cei, clk, reset);

```

endmodule

Counter block TCL file (1)

wave add / -radix hex

isim force add clk 0 -time 0 -value 1 -time 5ns -repeat 10ns

isim force add cei 1 -time 0 -value 0 -time 80ns -value 1 -time 100ns

isim force add reset 1 -time 0 -value 0 -time 10ns -value 1 -time 100ns -value 0 -time 110ns

run 30us

Counter block simulation waveform (2)



Programmable timer Verilog code (for testing timer) (3)

```
module timer10hz(zero, tp, clk, cei, reset);
    output zero, tp;
    input clk, reset, cei;

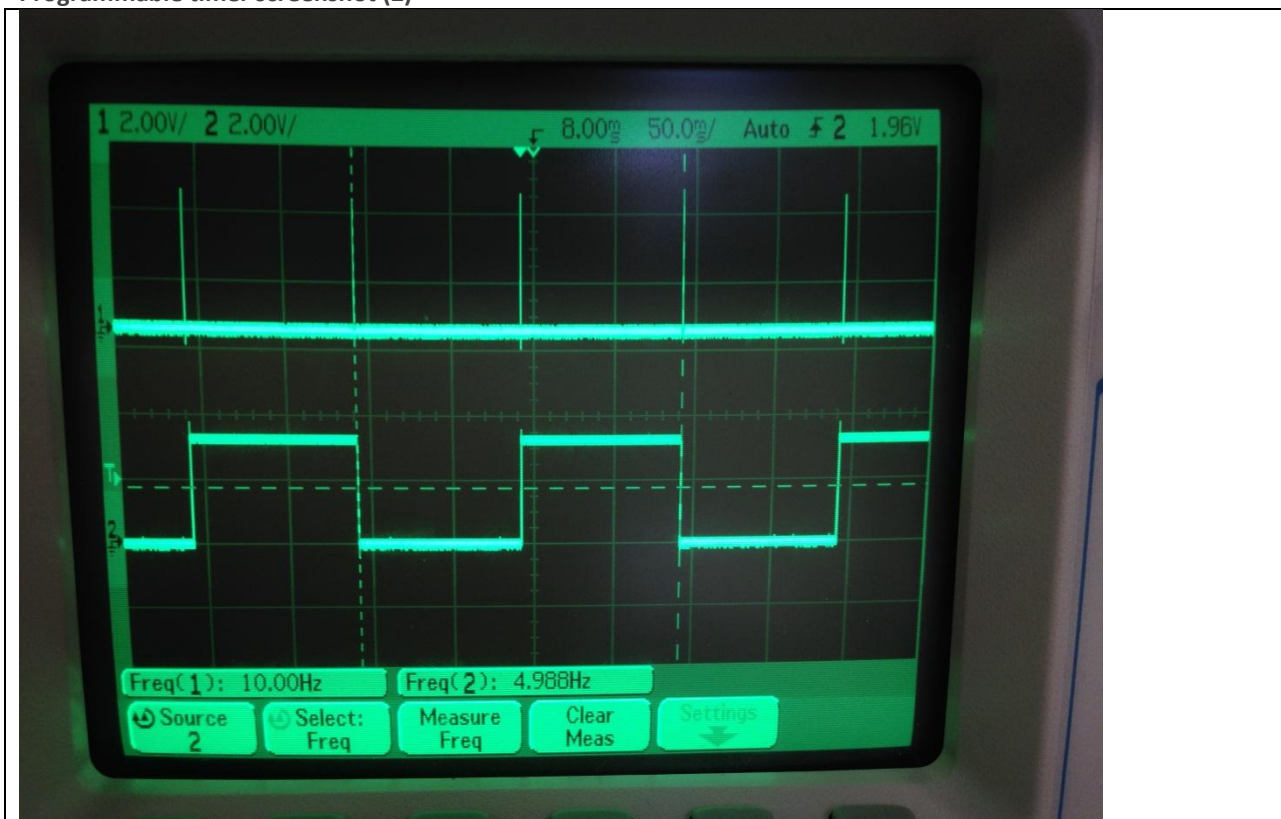
    prog_timer timer1(clk, reset, cei, 24'd5000000, counter, zero, tp);

endmodule
```

Programmable timer UCF file (1)

```
NET clk LOC = "B8"; # Bank = 0, Pin name = IP_L13P_0/GCLK8, Type = GCLK, Sch name = GCLK0
NET reset LOC = "H13"; # Bank = 1, Pin name = IP, Type = INPUT, Sch name = BTN3
NET zero LOC = "B4"; # Bank = 0, Pin name = IO_L24N_0, Type = I/O, Sch name = R-IO1
NET tp LOC = "A4"; # Bank = 0, Pin name = IO_L24P_0, Type = I/O, Sch name = R-IO2
```

Programmable timer screenshot (2)



TestBench Verilog code (2)

```
module stopwatch(Ca, Cb, Cc, Cd, Ce, Cf, Cg,
    AN0, AN1, AN2, AN3, DP, start, stop, reset, System_Clock, Dp0, Dp1, Dp2, Dp3);

    output Ca, Cb, Cc, Cd, Ce, Cf, Cg, AN0, AN1, AN2, AN3, DP;
    input start, stop, reset, System_Clock, Dp0, Dp1, Dp2, Dp3;
```

```

        wire[15:0] connector;
        wire latchsignal;

//        output Ca, Cb, Cc, Cd, Ce, Cf, Cg, AN0, AN1, AN2, AN3, DP, tp, zero;
//        input[3:0] Digit1, Digit2, Digit3, Digit4;
//        input System_Clock, Reset, Dp0, Dp1, Dp2, Dp3;

        segcontroller47 seg1(Ca, Cb, Cc, Cd, Ce, Cf, Cg, AN0, AN1, AN2, AN3, DP,
            connector[15:12], connector[11:8], connector[7:4], connector[3:0], System_Clock,
            1'b0, Dp0, 1'b1, Dp2, 1'b1);

//        output[15:0] q;
//        output rollover;
//        input clk, reset, cei;
        counterblock count1(connector, rollover, System_Clock, reset, zero);

//        output zero, tp;
//        input clk, reset, cei;
        timer10hz timer(zero, tp, System_Clock, latchsignal, reset);

//        output q, qbar;
//        input r, s;
        SRLatch latch1(latchsignal, qbar, stop, start);
endmodule

```

TestBench UCF file (1)

```

## clock pin for Nexys 2 Board
NET System_Clock LOC = "B8"; # Bank = 0, Pin name = IP_L13P_0/GCLK8, Type = GCLK, Sch name = GCLK0

## Switches
NET Dp0 LOC = "G18"; # Bank = 1, Pin name = IP, Type = INPUT, Sch name = SW0
NET Dp1 LOC = "H18"; # Bank = 1, Pin name = IP/VREF_1, Type = VREF, Sch name = SW1
NET Dp2 LOC = "K18"; # Bank = 1, Pin name = IP, Type = INPUT, Sch name = SW2
NET Dp3 LOC = "K17"; # Bank = 1, Pin name = IP, Type = INPUT, Sch name = SW3

## Buttons
NET stop LOC = "D18"; # Bank = 1, Pin name = IP/VREF_1, Type = VREF, Sch name = BTN1
NET start LOC = "E18"; # Bank = 1, Pin name = IP, Type = INPUT, Sch name = BTN2
NET reset LOC = "H13"; # Bank = 1, Pin name = IP, Type = INPUT, Sch name = BTN3

## 7 segment display
NET Ca LOC = "L18"; # Bank = 1, Pin name = IO_L10P_1, Type = I/O, Sch name = CA
NET Cb LOC = "F18"; # Bank = 1, Pin name = IO_L19P_1, Type = I/O, Sch name = CB
NET Cc LOC = "D17"; # Bank = 1, Pin name = IO_L23P_1/HDC, Type = DUAL, Sch name = CC
NET Cd LOC = "D16"; # Bank = 1, Pin name = IO_L23N_1/LDC0, Type = DUAL, Sch name = CD
NET Ce LOC = "G14"; # Bank = 1, Pin name = IO_L20P_1, Type = I/O, Sch name = CE
NET Cf LOC = "J17"; # Bank = 1, Pin name = IO_L13P_1/A6/RHCLK4/IRDY1, Type = RHCLK/DUAL, Sch name = CF

```

```
NET Cg LOC = "H14"; # Bank = 1, Pin name = IO_L17P_1, Type = I/O, Sch name = CG
NET DP LOC = "C17"; # Bank = 1, Pin name = IO_L24N_1/LDC2, Type = DUAL, Sch name = DP

NET AN0 LOC = "F17"; # Bank = 1, Pin name = IO_L19N_1, Type = I/O, Sch name = AN0
NET AN1 LOC = "H17"; # Bank = 1, Pin name = IO_L16N_1/A0, Type = DUAL, Sch name = AN1
NET AN2 LOC = "C18"; # Bank = 1, Pin name = IO_L24P_1/LDC1, Type = DUAL, Sch name = AN2
NET AN3 LOC = "F15"; # Bank = 1, Pin name = IO_L21P_1, Type = I/O, Sch name = AN3
```

Anomalies

This lab was fun to accomplish. Now, I understand much more about hierarchy and how to keep track of all the inputs and outputs. Timer concept is pretty cool as well. What I had a trouble with this lab was to connect the counter block to 4x7 segment controller. It took so much time, but eventually it was good exercise to improve in Verilog. Thank you all the TAs !