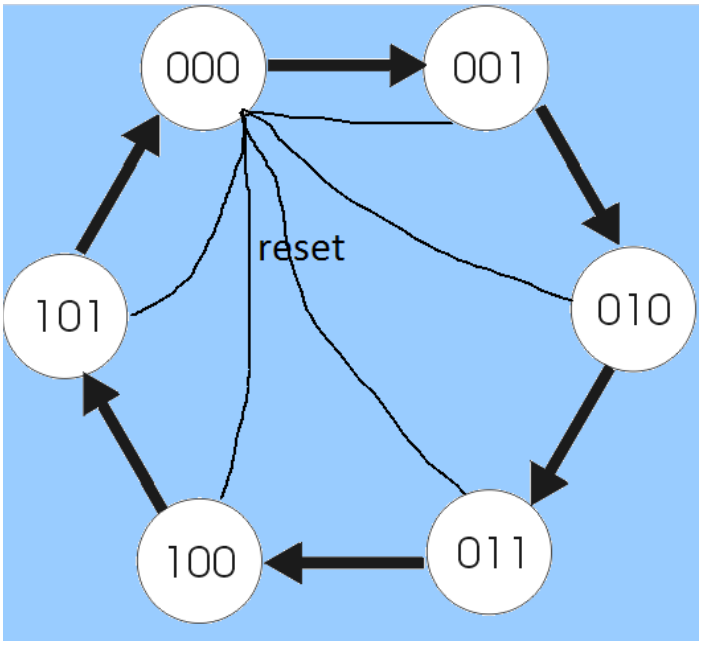
David DeGraw
Lab #9: Stopwatch
3/17/2015

Mod 6:

Reset	Inc	Q2	Q1	Q0	Q2+	Q1+	Q0+
0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	1
0	0	0	1	0	0	1	0
0	0	0	1	1	0	1	1
0	0	1	0	0	1	0	0
0	0	1	0	1	1	0	1
0	1	0	0	0	0	0	1
0	1	0	0	1	0	1	0
0	1	0	1	0	0	1	1
0	1	0	1	1	1	0	0
0	1	1	0	0	1	0	1
0	1	1	0	1	0	0	0
1	X	X	X	X	0	0	0



David DeGraw
Lab #9: Stopwatch
3/17/2015
Q2+:

		Q2 Q1			
		00	01	11	10
Q0	0	0	0	X	1
	1	0	1	X	0

Q2+ = Q1 Q0 + Q2 Q0'

Q1+:

		Q2 Q1			
		00	01	11	10
Q0	0	0	1	X	X
	1	1	0	X	0

Q1+ = Q1 Q0' + Q2' Q1' Q0

Q0+:

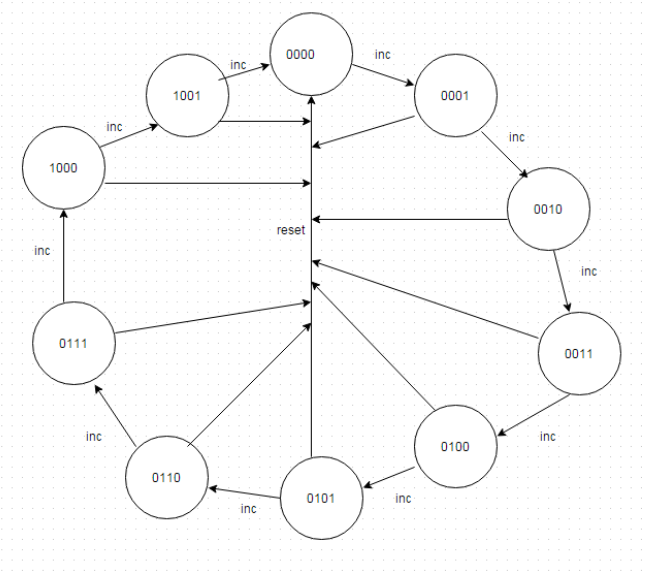
		Q2 Q1			
		00	01	11	10
Q0	0	1	1	0	1
	1	0	0	0	0

Q0+ = Q2' Q0' + Q1' Q0'

David DeGraw
Lab #9: Stopwatch
3/17/2015

Mod 10:

Q3	Q2	Q1	Q0	Q3+	Q2+	Q1+	Q0+
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	0	0	0	0



David DeGraw
Lab #9: Stopwatch
3/17/2015
Q3+:

		AB			
		00	01	11	10
CD	00	0	0	X	1
	01	0	0	X	0
	11	0	1	X	0
	10	0	0	X	0

Q3+= Q2Q1Q0 + Q3Q1'Q0'

Q2+:

		AB			
		00	01	11	10
CD	00	0	1	X	0
	01	0	1	X	0
	11	1	0	X	0
	10	0	1	X	0

Q2+ = Q2Q1' + Q2Q0' + Q3'Q2'Q1Q0

Q1+:

		AB			
		00	01	11	10
CD	00	0	0	X	0
	01	1	1	X	0
	11	0	0	X	0
	10	1	1	X	0

Q1+ = Q3'Q1'Q0 + Q3'Q1Q0'

David DeGraw
Lab #9: Stopwatch
3/17/2015

Q0+:

		AB			
		00	01	11	10
CD	00	1	1	X	1
	01	0	0	X	0
	11	0	0	X	0
	10	1	1	X	1

Q0+ = Q0’

SR Latch:

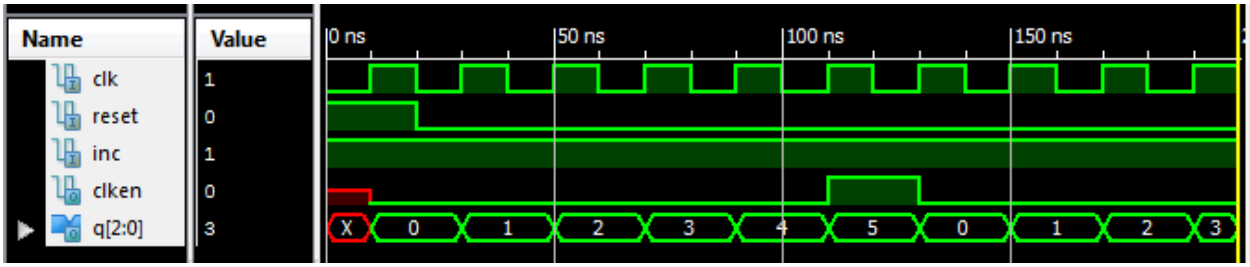
```
module srLatch (set, reset, q);
   input set, reset;
   output q;

   wire w1, w2;

   nor(w1, set, q);
   nor(q, reset, w1);

endmodule
```

```
isim force add set 0 -time 0 -value 1 -time 20ns
isim force add reset 1 -time 0 -value 0 -time 10ns -value 1 -
time 30ns
```

David DeGraw
Lab #9: Stopwatch
3/17/2015
Mod 6:

```
module mod6counter(q, clken, inc, clk, reset);
   input clk, reset, inc;
   output reg [2:0] q;
   output clken;

   always @(posedge clk)
   begin
   if (reset)
      q <= 3'b000;
   else if (inc)
      q <= q + 3'b001;

   if (q == 3'b101 && inc)
      q <= 3'b000;
   end

   assign clken = (q == 3'b101 && inc) ? 1'b1 : 1'b0;

endmodule
```

```
isim force add clk 0 -time 0 -value 1 -time 10ns -repeat 20ns
isim force add inc 1
isim force add reset 1 -time 0 -value 0 -time 20ns
```

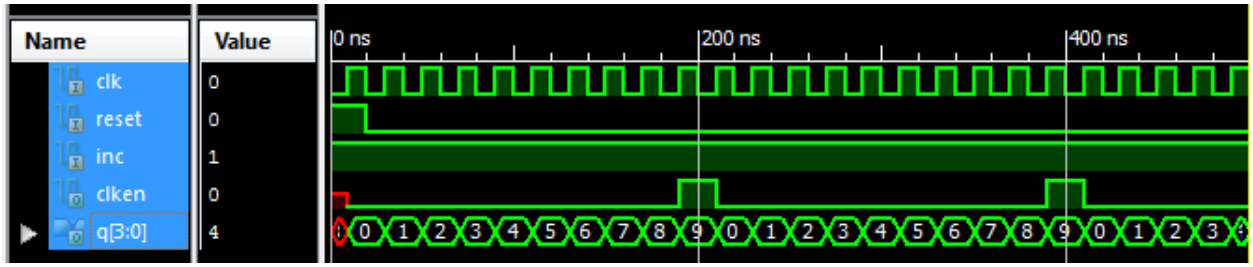


David DeGraw
Lab #9: Stopwatch
3/17/2015

Mod 10:

```
module mod10counter(q, clken, inc, clk, reset);
   input clk, reset, inc;
   output reg [3:0] q;
   output clken;

   always @(posedge clk)
   begin
   if (reset)
      q <= 4'b0000;
   else if (inc)
      q <= q + 4'b0001;

   if (q == 4'b1001 && inc)
      q <= 4'b0000;
   end

   assign clken = (q == 4'b1001 && inc) ? 1'b1 : 1'b0;

endmodule
```

```
isim force add clk 0 -time 0 -value 1 -time 10ns -repeat 20ns
isim force add inc 1
isim force add reset 1 -time 0 -value 0 -time 20ns
```

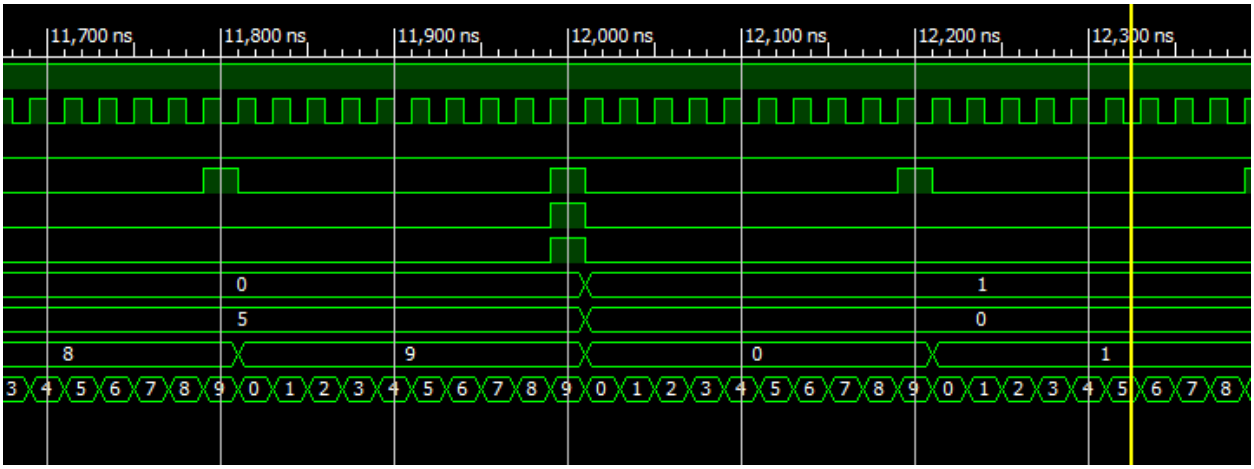


David DeGraw
Lab #9: Stopwatch
3/17/2015

Counter Block:

```
module counterBlock (d1, d2, d3, d4, inc, clk, reset);
   input inc, clk, reset;
   wire r1, r2, r3;
   wire [2:0] d2tmp;
   output [3:0] d1, d2, d3, d4;

   mod10counter thousandth (d4, r1, inc, clk, reset);
   mod10counter tenth (d3, r2, r1, clk, reset);
   mod6counter second (d2tmp, r3, r2, clk, reset);
   mod10counter minute (d1, clken, r3, clk, reset);

   assign d2 = {1'b0, d2tmp};

endmodule
```

```
isim force add clk 0 -time 0 -value 1 -time 10ns -repeat 20ns
isim force add inc 1
isim force add reset 1 -time 0 -value 0 -time 20ns
```



David DeGraw
Lab #9: Stopwatch
3/17/2015

Test Bench:

```
module testBench (clk, reset, stop, sett, w3, seg, odp, w1);
   input clk, reset, sett, stop;
   wire [3:0] d1, d2, d3, d4;
   output [3:0] w3;
   wire srTemp, zero, tp;
   wire r1, r2, r3;
   wire [2:0] d2tmp;
   output odp;
   output [6:0] seg;
   output [1:0] w1;

   counterBlock c1 (d1, d2, d3, d4, zero, clk, reset);

   prog_timer t1 (clk, reset, srTemp, 24'd5000000, counter, zero, tp);

   srLatch sr (sett, stop, srTemp);

   controller47 seg7 (4'b1010, w3, d1, d2, d3, d4, seg, odp, clk, 0, w1);
endmodule
```

```
## clock pin for Nexys 2 Board
NET clk   LOC = "B8"; # Bank = 0, Pi
```

```
## Buttons
NET reset LOC = "B18"; # Bank = 1, Pin name = IP, Type = INPUT, Sch
#NET increment LOC = "D18"; # Bank = 1, Pin name = IP/VREF_1, Type =
NET stop LOC = "E18"; # Bank = 1, Pin name = IP, Type = INPUT, Sch =
NET sett LOC = "H13"; # Bank = 1, Pin name = IP, Type = INPUT, Sch =

## 7 segment display
NET seg[6] LOC = "L18"; # Bank = 1, Pin name = IO_L10P_1, Type = I/
NET seg[5] LOC = "F18"; # Bank = 1, Pin name = IO_L19P_1, Type = I/
NET seg[4] LOC = "D17"; # Bank = 1, Pin name = IO_L23P_1/HDC, Type =
NET seg[3] LOC = "D16"; # Bank = 1, Pin name = IO_L23N_1/LDC0, Type
NET seg[2] LOC = "G14"; # Bank = 1, Pin name = IO_L20P_1, Type = I/
NET seg[1] LOC = "J17"; # Bank = 1, Pin name = IO_L13P_1/A6/RHCLK4/
NET seg[0] LOC = "H14"; # Bank = 1, Pin name = IO_L17P_1, Type = I/
NET odp     LOC = "C17"; # Bank = 1, Pin name = IO_L24N_1/LDC2, Typ

NET w3[0] LOC = "F17"; # Bank = 1, Pin name = IO_L19N_1, Type = I/O
NET w3[1] LOC = "H17"; # Bank = 1, Pin name = IO_L16N_1/A0, Type =
NET w3[2] LOC = "C18"; # Bank = 1, Pin name = IO_L24P_1/LDC1, Type =
NET w3[3] LOC = "F15"; # Bank = 1, Pin name = IO_L21P_1, Type = I/O
```

David DeGraw
Lab #9: Stopwatch
3/17/2015

Anomalies:

I was sending set (start) and reset into my srLatch (probably because of the name: set and reset latch) but I really needed to send my start and stop signals. Just a little confusing.

The instructions for this lab probably should be re written as they were very hard to follow.

The flip flop in the preparation section was totally unrelated to the lab?