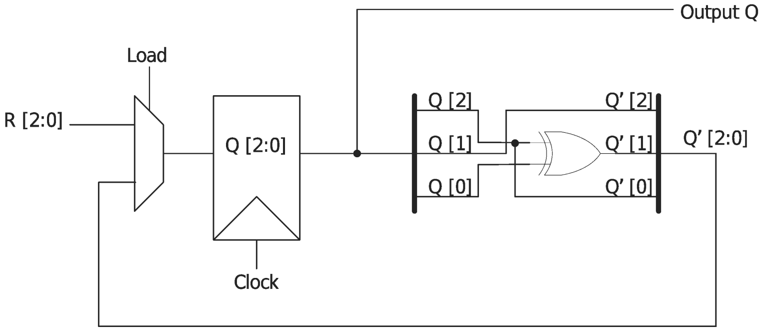
University of California at Berkeley
Department of Electrical Engineering and Computer Science

EECS150/EE141/EE241A, Fall 2014

Homework 2 Solutions

Total for CS150/EE141: 90 **Total for EE241A: 100**

1. (a) **[5]**

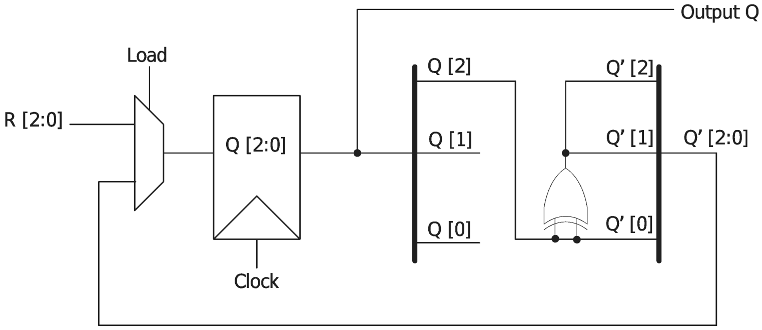


(b) **[4]**

Cycle	Q[2], Q[1], Q[0]
0	001
1	010
2	100
3	011
4	110
5	111
6	101
7	001

(c) If first cycle is cyle 0, it repeats on cycle 7. Therefore, repeats every 7 cycles. **[2]**

2. (a) **[6]**



(b) **[2]**

The generator always outputs 000 after the first cycle.

Cycle	Q[2], Q[1], Q[0]
0	001
1	000
2	000
3	...

3. **[6]**

```
module counter(
    input clk,
    input reset,
    input up_down,
    output [7:0] count
);
    reg [7:0] cnt;

    assign count = cnt;

    always@(posedge clk) begin

        if(reset)
            cnt <= 8'd0;
        else if(up_down == 0)
            cnt <= cnt + 1;
        else
            cnt <= cnt - 1;
    end

endmodule
```

4. (a) **[2]**

$$Y = a\overline{b}\overline{c} + a\overline{b}c + \overline{a}\overline{b}\overline{c} + \overline{a}\overline{b}c + b\overline{c}$$

(b) **[4]**

$$Y = a\overline{b}\overline{c} + a\overline{b}c + \overline{a}\overline{b}\overline{c} + \overline{a}\overline{b}c + b\overline{c}$$
$$Y = a\overline{b}(\overline{c} + c) + \overline{a}\overline{b}(\overline{c} + c) + b\overline{c}$$
$$Y = a\overline{b} + \overline{a}\overline{b} + b\overline{c}$$
$$Y = \overline{b}(a + \overline{a}) + b\overline{c}$$
$$Y = \overline{b} + b\overline{c}$$

.

5. (a) 3-bit input is a,b,c. Output is y. **[4]**

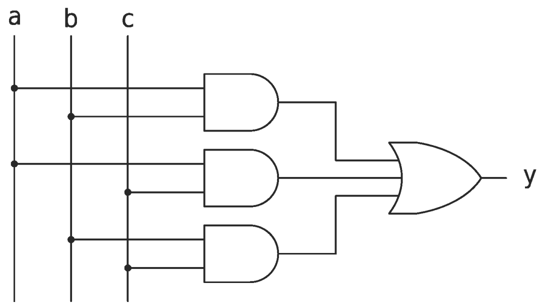
a	b	c	y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

(b) **[8]**

$$Y = \overline{a}bc + a\overline{b}c + ab\overline{c} + abc$$
$$Y = bc(\overline{a} + a) + a\overline{b}c + ab\overline{c}$$
$$Y = bc + a\overline{b}c + ab\overline{c}$$
$$Y = b(c + a\overline{c}) + a\overline{b}c$$
$$Y = b(c + a) + a\overline{b}c \text{ (by elimination theorem)}$$
$$Y = bc + ab + a\overline{b}c$$
$$Y = bc + a(b + \overline{b}c)$$
$$Y = bc + a(b + c) \text{ (by elimination theorem)}$$
$$Y = bc + ab + ac$$

(c) **[2]**

$$y = (a + b + c)(a + b + \overline{c})(a + \overline{b} + c)(\overline{a} + b + c)$$



6. **[12]**

w

cd \ ab	00	01	11	10
00	0	0	-	1
01	0	0	-	0
11	0	1	-	-
10	0	0	-	-

W = $bcd + a\bar{d}$

x

cd \ ab	00	01	11	10
00	0	1	-	0
01	0	1	-	0
11	1	0	-	-
10	0	1	-	-

X = $b\bar{c} + b\bar{d} + \bar{b}cd$

y

cd \ ab	00	01	11	10
00	0	0	-	0
01	1	1	-	0
11	0	0	-	-
10	1	1	-	-

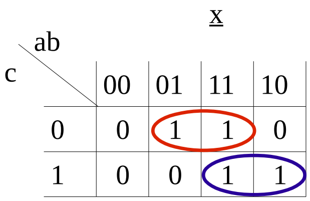
Y = $\overline{ac}d + c\bar{d}$

z

cd \ ab	00	01	11	10
00	1	1	-	1
01	0	0	-	0
11	0	0	-	-
10	1	1	-	-

Z = $\bar{d}$

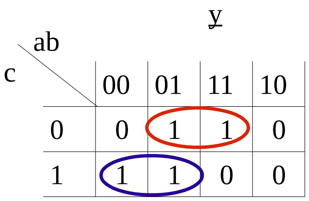
7. (a) **[12]**



x

c \ ab	00	01	11	10
0	0	1	1	0
1	0	0	1	1

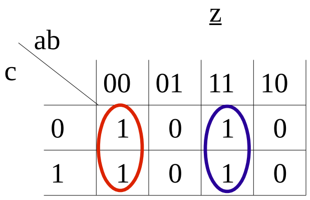
$X = b\bar{c} + ac$



y

c \ ab	00	01	11	10
0	0	1	1	0
1	1	1	0	0

$Y = b\bar{c} + \bar{a}c$



z

c \ ab	00	01	11	10
0	1	0	1	0
1	1	0	1	0

$Z = \bar{a}\,\bar{b} + ab$

(b) Note: x,y,z is equivalent to cnt[2], cnt[1], cnt[0]. **[5]**

```
module counter(
    input clk,
    output reg [2:0] cnt
);
    always@(posedge clk) begin
        cnt[2] <= (cnt[1] & !cnt[0]) | (cnt[2] & cnt[0]);
        cnt[1] <= (cnt[1] & !cnt[0]) | (!cnt[2] & cnt[0]);
        cnt[0] <= (!cnt[2] & !cnt[1]) | (cnt[2] & cnt[1]);
    end

endmodule
```

8. **[7]**

```
module serial_to_parallel(
    input clk,
    input sin,
    input shift_in,
    input load_pout,
    output reg [3:0] pout
);
    reg [3:0] temp_pout;

    always@(posedge clk) begin
```

```
        if(shift_in)
            temp_pout <= {temp_pout[2:0], sin};

        if(load_pout)
            pout <= temp_pout;
    end

endmodule
```

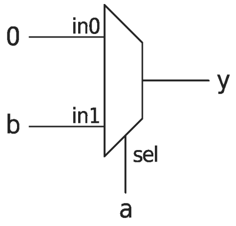
9. (a) There are three critical paths. **[3]**
   Path 1: from output of R1, through G2, G4, G1, to input of R1.
   Path 2: from output of R1, through G3, G4, G1, to input of R2.
   Path 3: from output of R2, through G3, G4, G1, to input of R1.

   (b) **[4]**

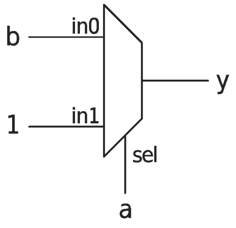
$$\begin{aligned}\text{Max delay} &= T_{\text{clk-q, max}} + 3 \times T_{\text{gate, max}} + T_{setup} \\ &= 60ps + 3 \times 100ps + 50 \\ &= 410ps \\ F_{max} &= \frac{1}{410ps} = 2.44\text{GHz}\end{aligned}$$

   (c) No. The shortest path in the circuit is R2 through G1 to R1. Minimum clk-to-q + minimum gate delay = 120ps. This is larger than the hold time of 45ps, hence hold time will not be violated. **[2]**

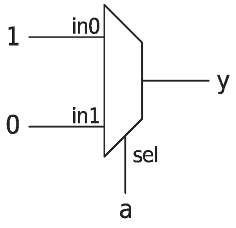
10. **For EE241A only:** We can build AND, OR and NOT gates as follows: **[10]**



Y = a AND b



Y = a OR b



Y = NOT a

To build a NAND gate: connect NOT to output of AND.
To build a NOR or XOR gate: connect NOT gates to the inputs/outputs of an OR gate.