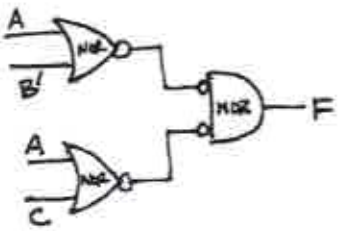
EECS 373 - Fall 14 - Homework #1 - Key
Prabal Dutta

Q1a. $F = (A + B') \cdot (A + C)$

A	B	C	(A+B')	(A+C)	F
0	0	0	1	0	0
0	0	1	1	1	1
0	1	0	0	0	0
0	1	1	0	1	0
1	0	0	1	1	1
1	0	1	1	1	1
1	1	0	1	1	1
1	1	1	1	1	1

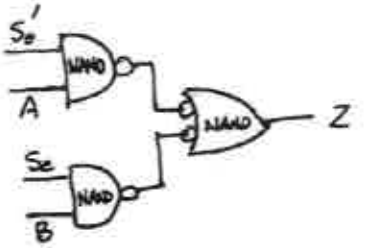
Q1b.



Q2a. $Z = S_e' \cdot A + S_e \cdot B$

A	B	Se	Z
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

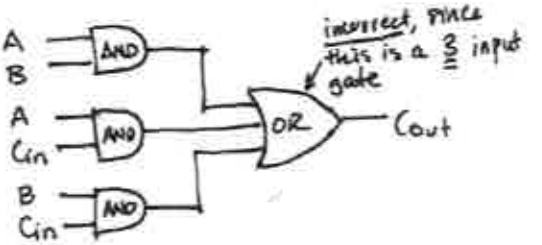
Q2b.



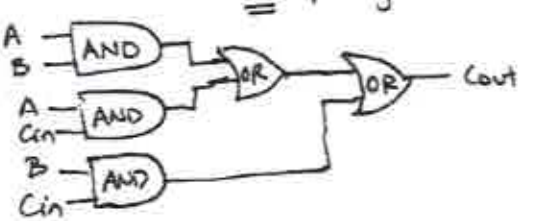
Q3a.

A	B	Cin	Cout	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Q3b.



⇓ redrawn using 2 input gates



Q4a. $Y = (A \cdot B') + (B \cdot C') + (A \cdot C)'$

⇓ verilog

```
assign Y = (A&~B) | (B&~C) | (~(A&C));
```

Q4b. Assume the following truth table:

S_1	S_0	Z
0	0	A
0	1	B
1	0	C
1	1	D

⇓ verilog

```
assign Z = (~S1 & ~S0 & A) | (~S1 & S0 & B) ... 2
         | (S1 & ~S0 & C) | (S1 & S0 & D);
```

Q5a. $Z = f(A, B, C, D, S_0, S_1)$ ⟶

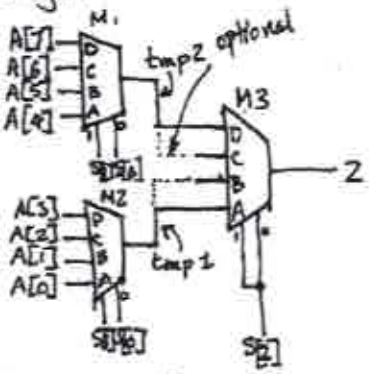
S_1	S_0	Z
0	0	A
0	1	B
1	0	C
1	1	D

Mux truth table

⇓ Verilog

```
module MUX41 (A, B, C, D, S0, S1, Z);
    input A, B, C, D, S0, S1;
    output Z;
    wire Z;
    assign Z = (~S1 & ~S0 & A) | (~S1 & S0 & B) | (S1 & ~S0 & C) | (S1 & S0 & D);
endmodule
```

Q5b. The key here is realizing there will be some unused inputs in the last stage mux. A possible design:



Also, note that there are other possible designs. For example, if mux M3's S_1 and S_0 inputs are connected to ϕ and S[2], respectively, then M3.A = tmp1 and M3.B = tmp2, rather than the way it is drawn here.

$Z = f(A[7:0], S[2:0])$

⇓ Verilog

```
module MUX81 (A, S, Z);
    input [7:0] A;
    input [2:0] S;
    output Z;
    wire Z, tmp1, tmp2;
    MUX41 M1 (A[4], A[5], A[6], A[7], S[0], S[1], tm
    MUX41 M2 (A[0], A[1], A[2], A[3], S[0], S[1], tm
    MUX41 M3 (tmp1, tmp1, tmp2, tmp2, S[2], S[2], Z,
endmodule
```

In this example, we didn't write out the entire parameter list in the module declaration, e.g.

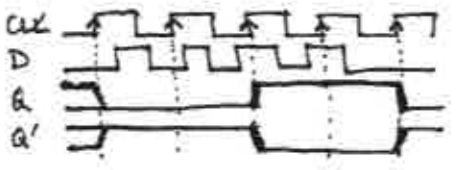
module MUX81 (A0, A1, A2, ... A7, S0, S1, S2, Z)

but instead chose to pass the arrays

Q6. In this case, Y must a reg and <u>not</u> a wire:

```
always @*
    Y = (A & ~B) | (B & ~C) | ~(A & C);
```

Q7.

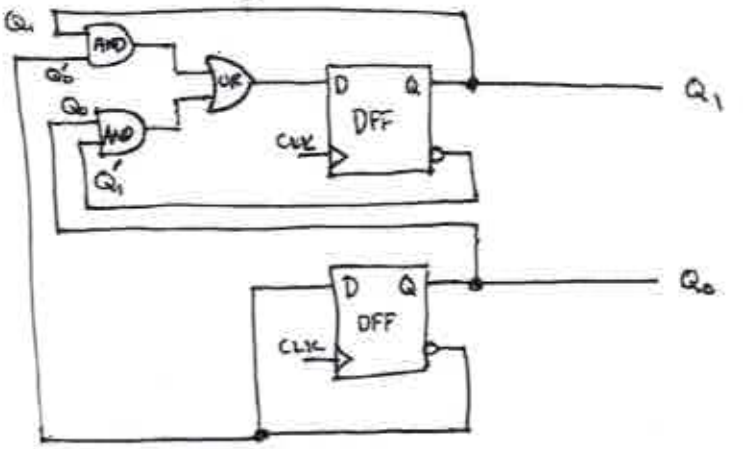


Q8. Build a modulo-4 counter: $0 \to 1 \to 2 \to 3$

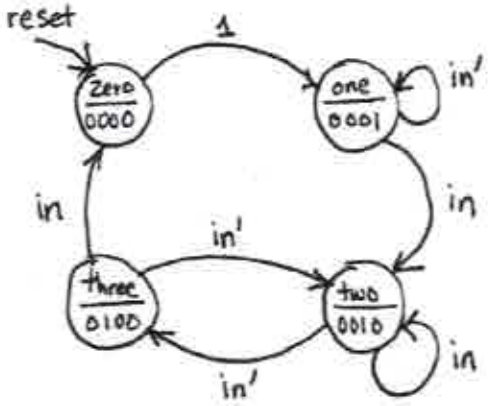
PS Q_1	PS Q_0	NS Q_1	NS Q_0	D_1	D_0
0	0	0	1	0	1
0	1	1	0	1	0
1	0	1	1	1	1
1	1	0	0	0	0

$D_1 = Q_1 \oplus Q_0 = Q_1' \cdot Q_0 + Q_1 \cdot Q_0'$

$D_0 = Q_0'$



Q9.



Q10. Consider a D flip flop (DFF): with inputs D and CLK and output Q, as shown to the right:

- Setup time – the time before a rising edge of the CLK during which D must be stable (must not change)
- Hold time - the time after a rising edge of the clock during which D must be stable / must not change