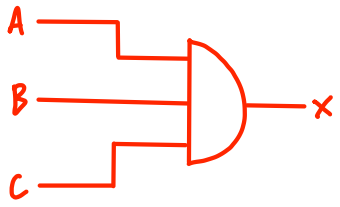
CSE369 Practice Problems (lec 2)

1. Circuit diagrams

Draw a truth table and circuit diagram for each of the following boolean equations

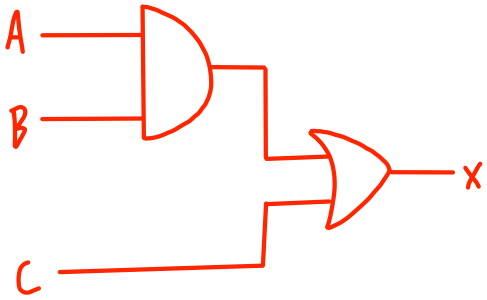
1.1] $X = ABC$

A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1



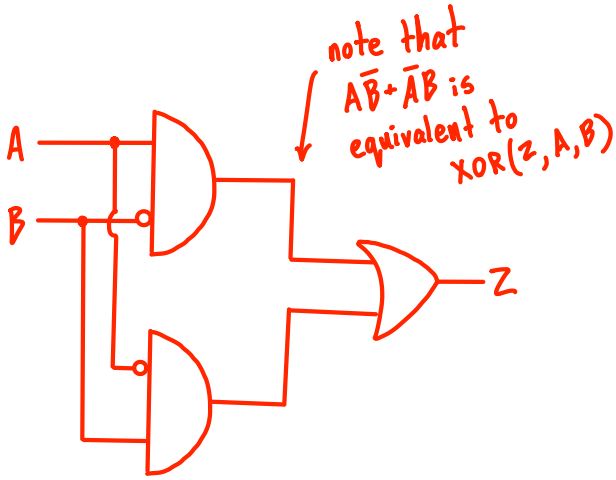
1.2] $X = AB + C$

A	B	C	X
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



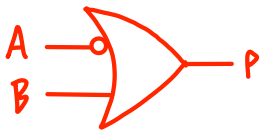
1.3] $Z = A\overline{B} + \overline{A}B$

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

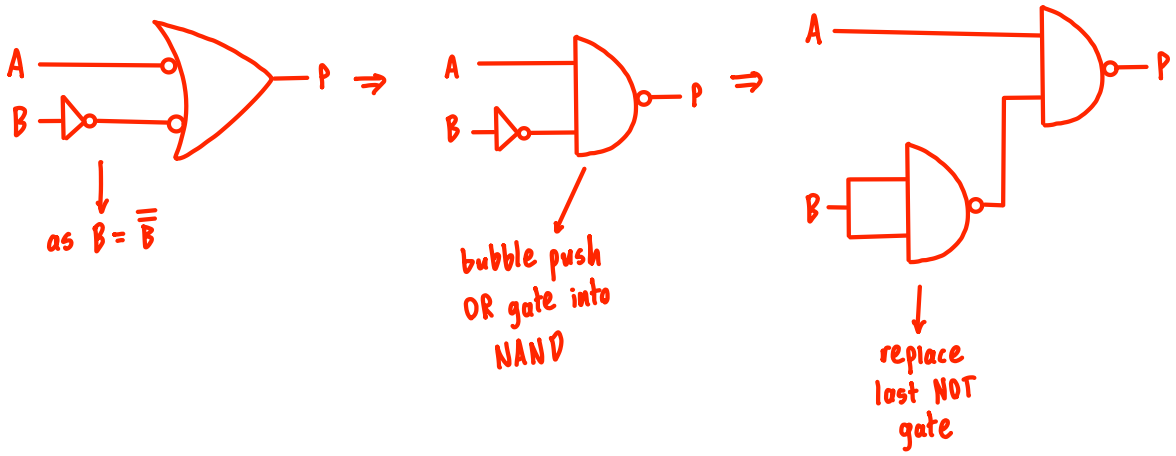


2. DeMorgan's Law

2.1.1] Draw this equation as a circuit: $P = \overline{A} + B$



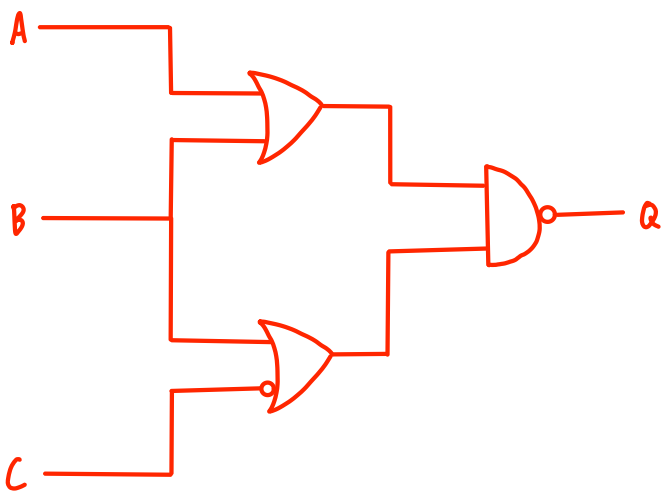
2.1.2] Use "bubble pushing" to convert the circuit diagram to *only* use **NAND** gates



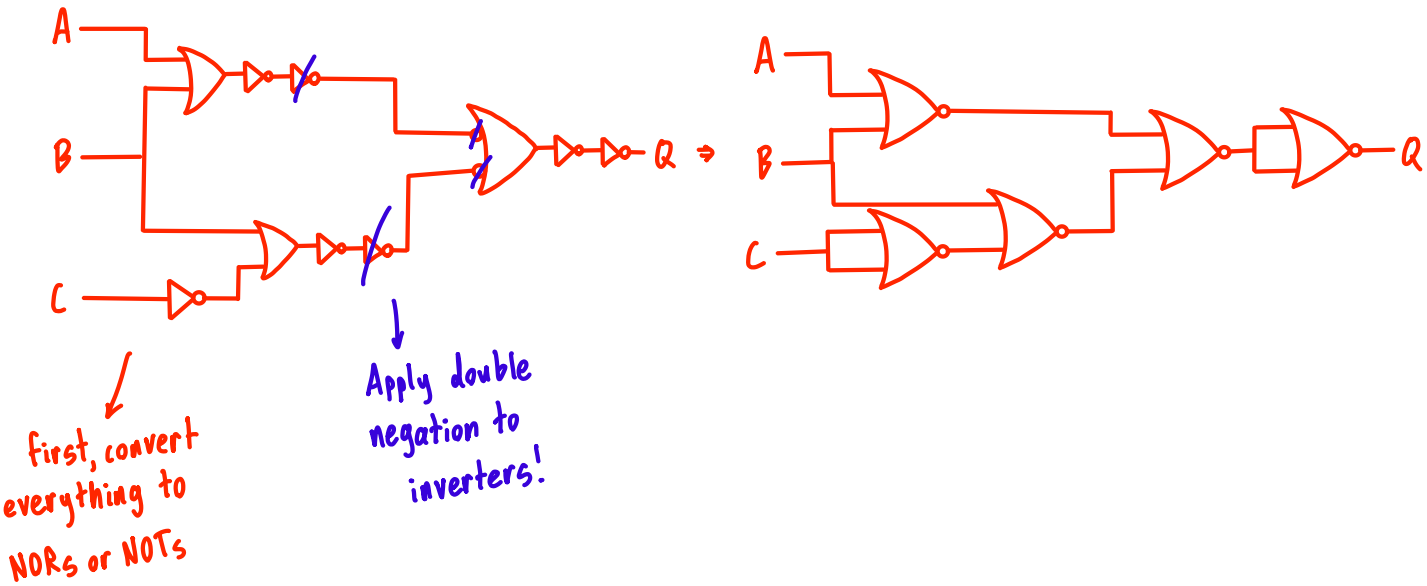
2.1.3] Write the circuit as a boolean equation

$P = \overline{\overline{B} \cdot B} \cdot A$

2.2.1] Draw this equation as a circuit: $Q = \overline{(A + B)(B + \sim C)}$



2.2.2] Use "bubble pushing" to convert the circuit diagram to *only* use **NOR** gates



2.2.3] Write the circuit as a boolean equation

$$Q = \overline{\left(\overline{A + B}\right) + \left(\left(\overline{C + C}\right) + B\right)}$$

3. Verilog Signals

3.1] Declare an six bit bus called “twix” and a 16 bit bus called “kitkat”

```
logic [5:0] twix;
logic [15:0] kitkat;
```

3.2] Declare a 16 bit bus called “katkit”. Use concatenation to assign its value to be the bytes of “kitkat” in the opposite order.

```
logic [15:0] katkit;
assign katkit = {kitkat[7:0], kitkat[15:8]};
```

3.3] Declare a 16 bit bus called “nokit”. Use concatenation to assign its value to be the upper byte of “kitkat” padded out to 16 bits with 0s

```
logic [15:0] nokit;
assign nokit = {8{0}, kitkat[15:8]};
```

3.4] Declare a six bit bus called “xiwt”. Use concatenation to assign its value to be the reversed bits of twix.

```
logic [5:0] xiwt;
assign xiwt = {twix[0], twix[1], twix[2], twix[3], twix[4], twix[5]};
```

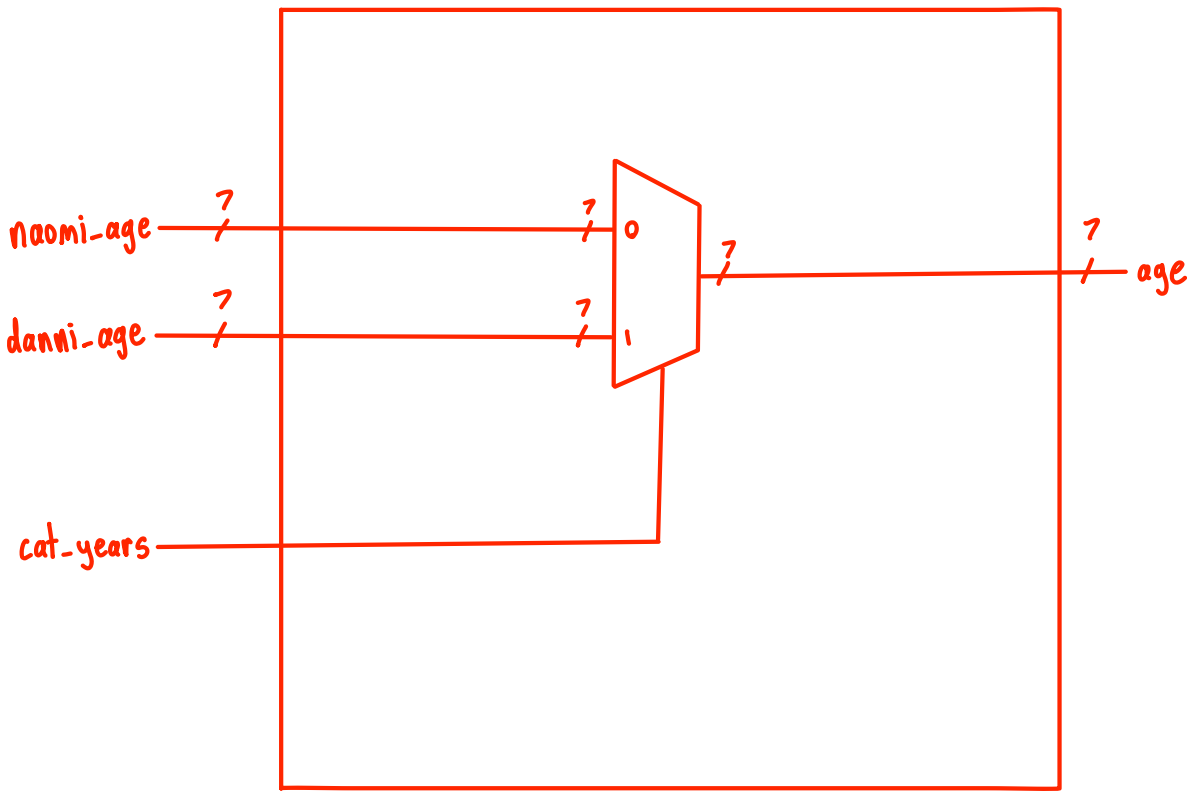
3.5] Declare a three bit bus called “tw” and a three bit bus calle “ix”. Use concatenation on the left side of an assignment to break the value of twix apart into tw and ix.

```
logic [2:0] tw;
logic [2:0] ix;
assign {tw[2:0], ix[2:0]} = twix[5:0];
```

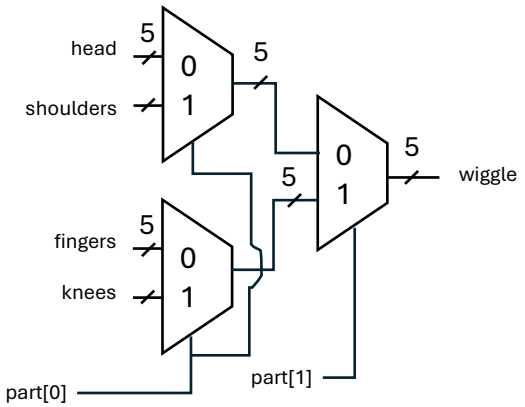
4. Block diagramming

4.1] Draw the following Verilog as a block diagram:

```
logic [6:0] danni_age;
logic [6:0] naomi_age;
logic [6:0] age;
logic cat_years ;
always_comb begin
        if (cat_years) begin
                age = danni_age;
        end else begin
                age = naomi_age;
        end
end
```



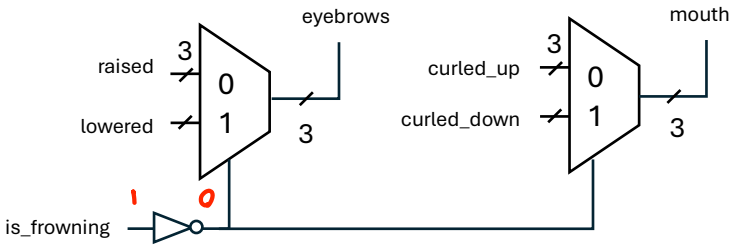
4.2] Write Verilog for the following block diagram:




```
logic [4:0] head, shoulders, fingers, knees, wiggle;
logic [1:0] part;

always_comb begin
    if (part == 0) begin
        wiggle = head;
    end else if (part == 1) begin
        wiggle = shoulders;
    end else if (part == 2) begin
        wiggle = fingers;
    end else begin
        wiggle = knees;
    end
end
```

4.3] Write Verilog for the following block diagram:




```
logic [2:0] raised, lowered, eyebrows;
logic [2:0] curled_up, curled_down, mouth;
logic is_frowning;

always_comb begin
    if (is_frowning) begin
        eyebrows = raised;
        mouth = curled_up;
    end else begin
        eyebrows = lowered;
        mouth = curled_down;
    end
end
```