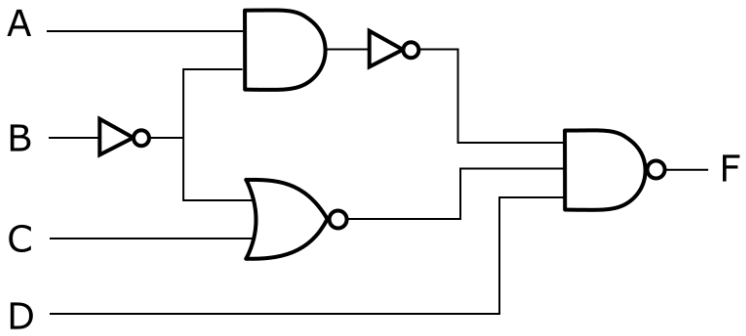
Problem 1



Write a Boolean expression for the circuit above. Do not simplify it (e.g. through algebraic manipulation).

$$F = \overline{\overline{A \cdot \overline{B}} \cdot \overline{\overline{B} + C} \cdot D}$$

Complete the truth table for the circuit above.

A	B	C	D	F
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

If D=0, F=1
If B=0 or C=1, F=1
If A=1 and B=0, F=1

$A \cdot \overline{B} + \overline{B} + C + \overline{D}$

(Just putting this in a box to use the empty space over here. It's a separate question, unrelated to anything else on this page.)

Write a Boolean expression that implements the following logic.
If input "cat" is 1, the output should be 1 no matter what, otherwise the output should be the opposite of input "dog".

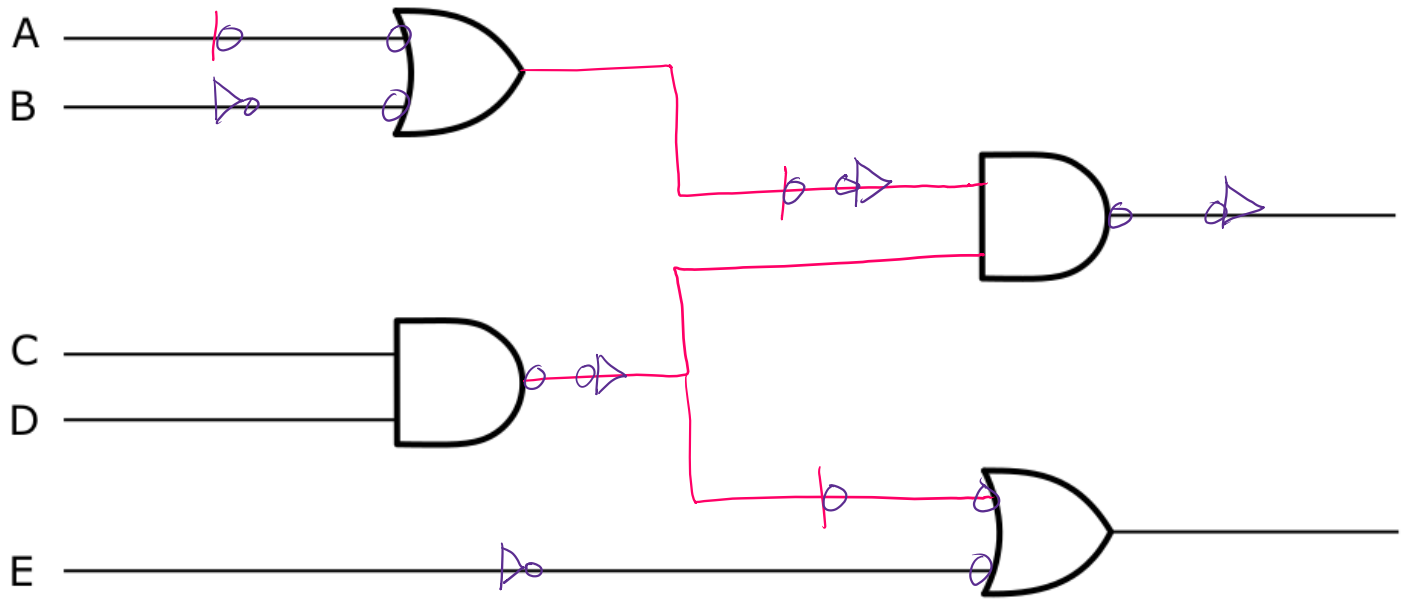
$cat + \overline{dog}$

Problem 2

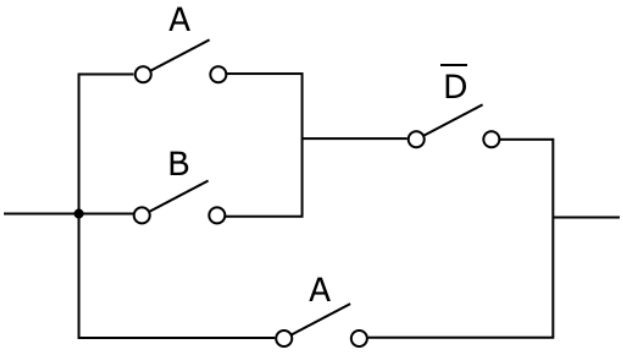
Draw a mixed-logic diagram that produces the two logic expressions below. Re-use the repeated term $C \cdot D$. A starting point has been provided to make grading easier, so you only need to fill in the middle connections. Once drawn, convert the diagram to be implemented using only NAND gates (and inverters). Minimize the number of inverters.

$$X = \overline{\overline{A} + B} \cdot C \cdot D$$

$$Y = \overline{C \cdot D} + E$$



Problem 3



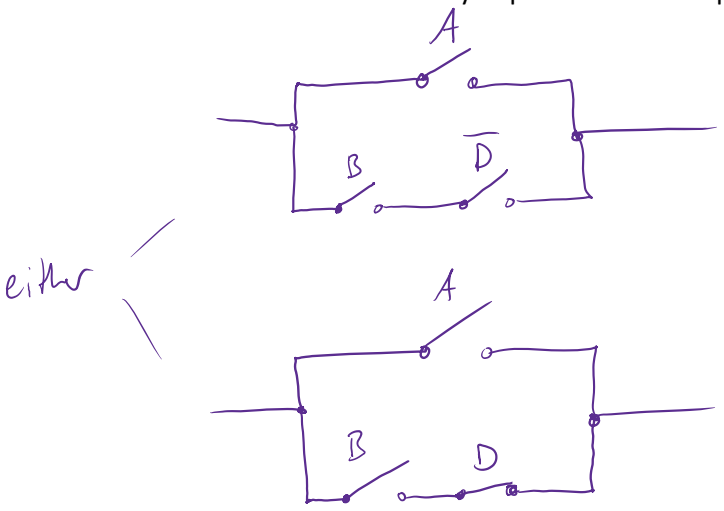
Write a Boolean expression that represents the connection between the sides of the switch network above. Do not simplify the expression (e.g. through algebraic manipulation).

$(A + B) \cdot \overline{D} + A$

Now simplify the above expression as much as possible using algebraic manipulation.

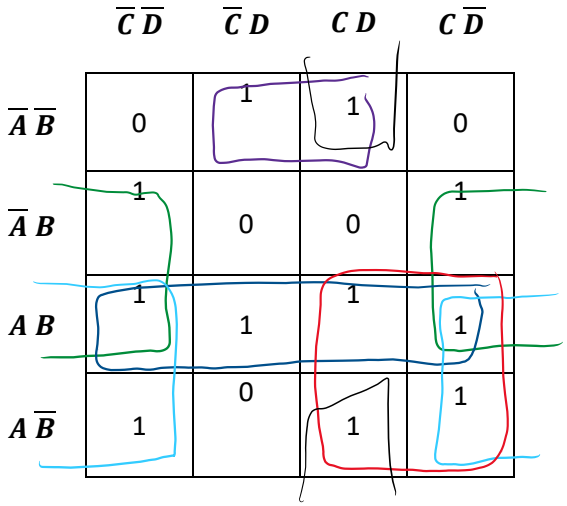
$A\overline{D} + B\overline{D} + A = A + B\overline{D}$

Draw a new switch network that directly implements the simplified expression derived above.



Problem 4

Circle the minimum number of prime implicants of 1s on the K-map below to cover the function.



Write the sum-of-products expression corresponding to the implicants you circled on the K-map above.

$\bar{A}\bar{B}D + B\bar{D} + A\bar{D} + AB + AC$

Are there any prime implicants in the K-map above (regardless of whether you circled them or not) that are not essential? If so, what are the Boolean expressions for them?

$\bar{B}CD,\ AC$

How many maxterms are in the K-map above?

Five

No longer specifically referring to the K-map above, but to some other K-map of four inputs A, B, C, and D, give an example of an implicant (in Boolean expression form) that covers a single minterm.

Anything with all inputs.

$ABCD,\ \bar{A}B\bar{C}D,\ \bar{A}\bar{B}C\bar{D}$, etc.

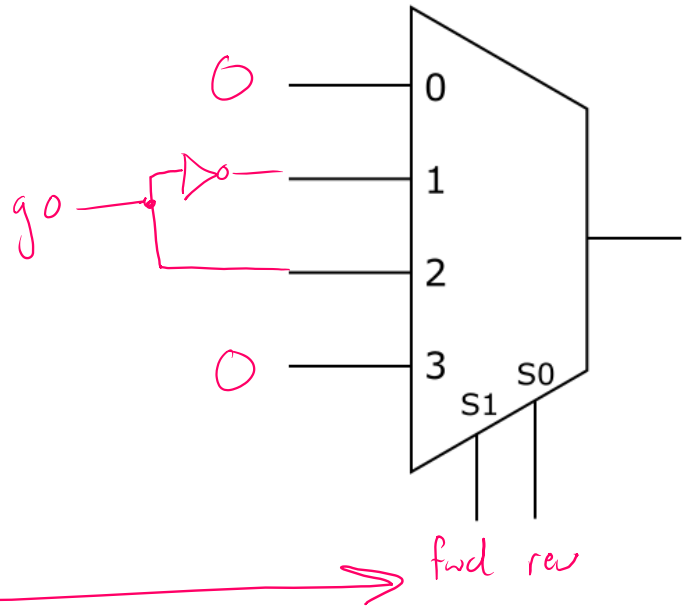
Problem 5

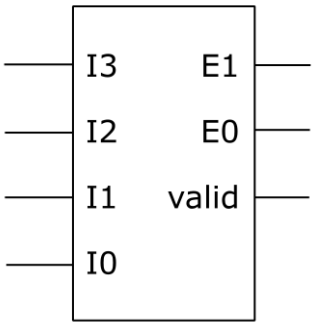
You've been asked to create a circuit that implements the following behavior, where input signals "fwd" and "rev" select an appropriate output for "dir", which is sometimes based on another input "go". To make the design easier, you've been asked to use a multiplexer. Design a circuit around the multiplexer by adding signal labels (which can include the named signals and '1' and '0') and primitive gates (AND, OR, NOT, etc.).

that's what a mux does!

fwd	rev	dir
0	0	0
0	1	$\overline{go}$
1	0	go
1	1	0

select





You know that the 4-to-2 encoder above uses priority of either $I_3>I_2>I_1>I_0$ or $I_0>I_1>I_2>I_3$. Describe the simplest way to test the device to determine which priority it uses.

Make I3 and I0 1, see what comes out.

Regardless of the priority, what set of inputs will ensure that all three outputs are active?

$I_{3-0} = 1000$

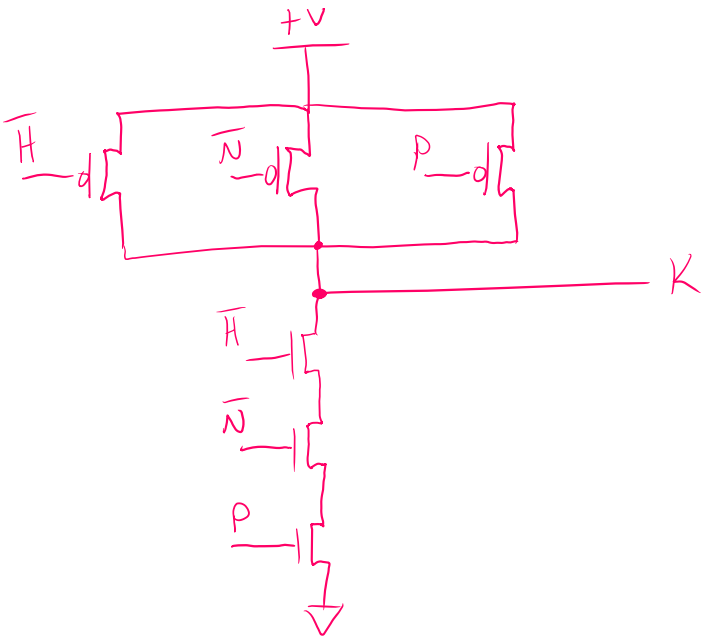
Problem 6

Create a proper CMOS circuit that implements the following Boolean expression.

$$K = H + \overline{\overline{N} \cdot P}$$

$pu = H + N + \bar{P}$

$pd = \bar{H} \cdot \bar{N} \cdot P$



Create a CMOS-like circuit (NFETs in a pull-down network and PFETs in a pull-up network) that tri-states output R when input S is low and connects Vdd and GND (a bad thing to do) when input S is high.

$pu =$

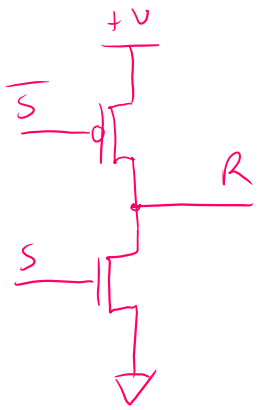
S	conn
0	0
1	1

$= S$

$pd =$

S	conn
0	0
1	1

$= S$



Problem 7

What is the following (base-10) value represented in unsigned binary? Use as many digits as necessary.

Hmm... $2^3 - 1 = 8 - 1 = 7 = 111_2$

$2^{12}-1 =$ 1111111111111

Convert the following base-10 numbers to 8-bit signed 2's complement representations.

13 = 8+4+1 = 00001101

negate 00001101 → 11110010 → +1 → 11110011

-13 = 11110011

Convert this 8-bit 2's complement number to 12 bits.

11011000 -> negative 111111011000

Convert the following base-10 number to an 8-bit sign-magnitude representation.

-13 = 10001101

Perform these unsigned binary additions.

```
    1
  0 0 0 1 1 1 1 1
+ 1 0 0 1 0 0 0 0
-----------------
  1 0 1 0 1 1 1 1
```

```
                  1 1 1 1 1 1 1 1 1
  0 1 1 1 1 1 1 1 1 1
+ 0 0 0 0 0 0 0 0 0 1
---------------------
  1 0 0 0 0 0 0 0 0 0
```

Interpret the following as 2's complement numbers. Keeping the same number of bits, will the additions result in correct or incorrect answers?

both medium-magnitude negative numbers

```
  11111101010110101101111010001010000101110111011010111101010100100100111011 1
+ 11111111111111110100110101010100100101011111010101010101010010010010010011 1
-------------------------------------------------------------------------------
```

correct / incorrect

both very large positive numbers

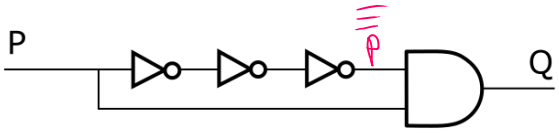
```
  01111101010110101101111010001010000101110111011010111101010100100100111011 1
+ 01111111111111110100110101010100100101011111010101010101010010010010010011 1
-------------------------------------------------------------------------------
```

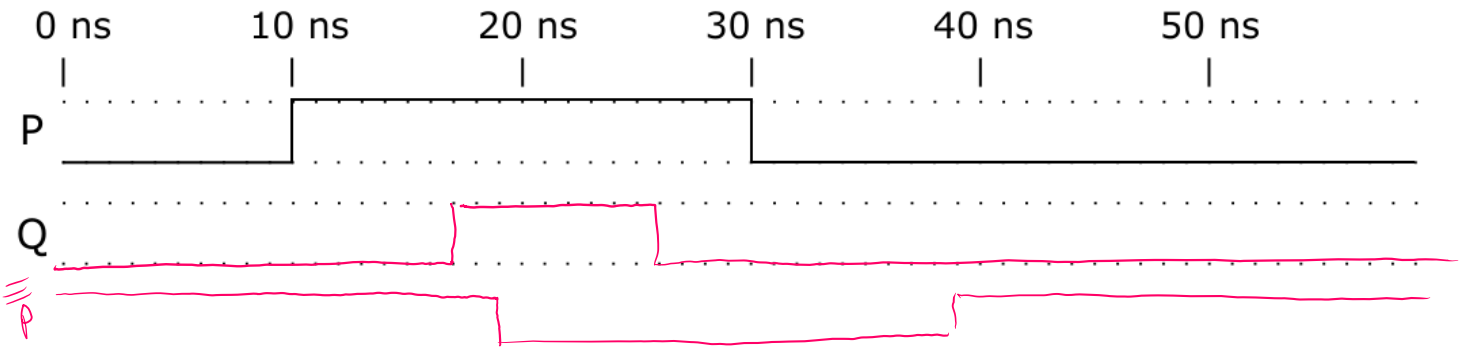
correct / incorrect

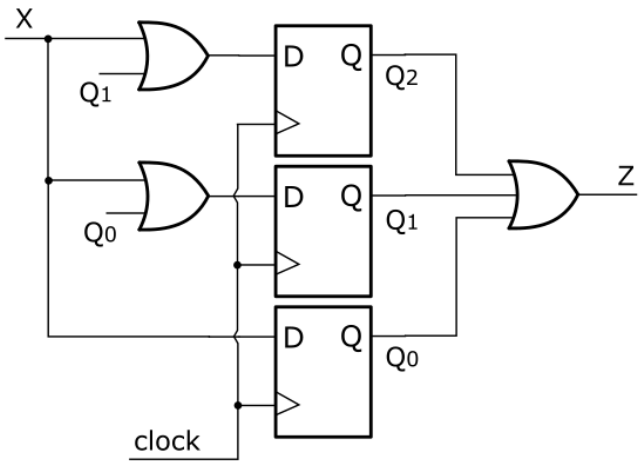
Problem 8

Sketch a timing diagram for this circuit's response to the provided input. Use the provided propagation delays.

OR gate	7 ns
NOT gate	3 ns







In the state machine above, after a rising edge on the clock, what is the soonest that the output Z might change? Express your answer in terms of parameters of the circuit elements; e.g. propagation delay of a 2-input OR gate T_{pOR2}.

$T_{pDFF} + T_{pOR3}$

In the circuit above, assume that nothing has happened for a long time, and then X changes at time t=0. To ensure correct state machine behavior (i.e. not only correct flip-flop behavior but also functional behavior of the state machine), when is the soonest that the first rising clock edge can occur?

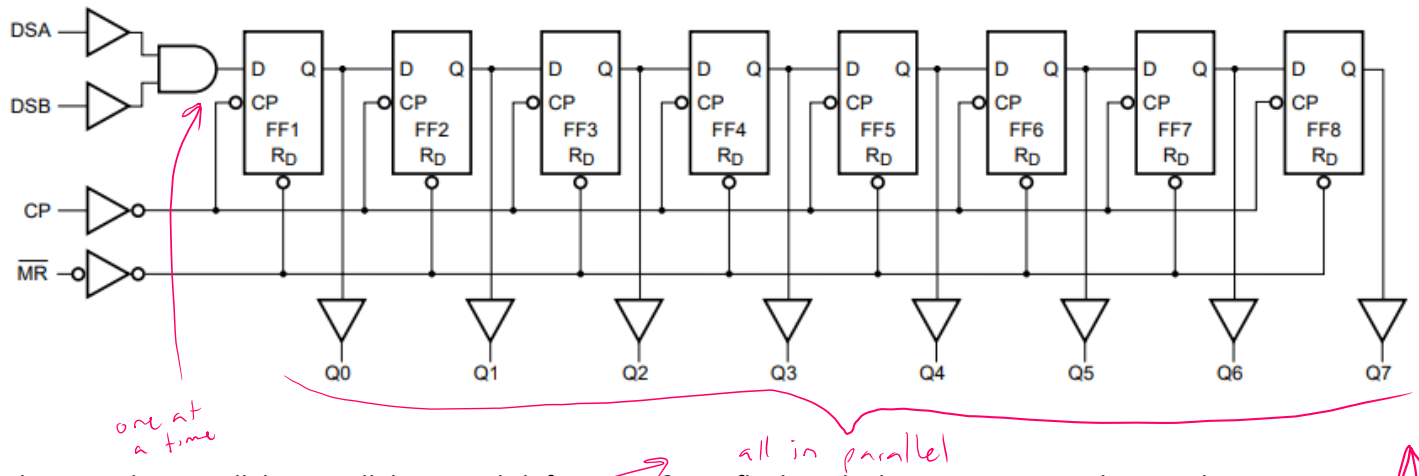
$T_{pOR2} + T_{su}$

Continuing from the above problem, what is the soonest that the second rising clock edge can occur?

$(T_{pOR2} + T_{su}) + \underbrace{T_{pDFF} + T_{pOR2} + T_{su}}_{\text{min clk period.}}$

Problem 9

This is the internal diagram of a 74HC164 chip, which is a shift register. Signal "MR" is "master reset", and the function of the rest of the signals you should be able to determine from the diagram.



one at a time

all in parallel

Is this a seral-to-parallel or parallel-to-serial shift register? Briefly describe how you came to that conclusion.

serial to parallel

The device has two signals DSA and DSB. If you only need one of those for a system you're designing, what should you do with the other?

Tie it high or to the other signal

If the outputs, as numbered, were used as a binary number $Q_{7\text{-}0}$, would data be shifted into this device most-significant or least-significant bit first?

Msb first

shifted in first

Problem 10

Create a state diagram for a state machine that implements the behavior described here.

The state machine monitors two buttons that produce signals, named A and B, that are 1 when the button is pressed and 0 when the button is not pressed. The state machine controls one output, named T.

When one button is pressed (not both), the output should toggle (i.e. switch from 0 to 1 or 1 to 0). Even if the button is held down, the output should only toggle once.

If both buttons are pressed, the output should toggle every clock cycle.

When no buttons are pressed, the output should turn off as soon as possible.