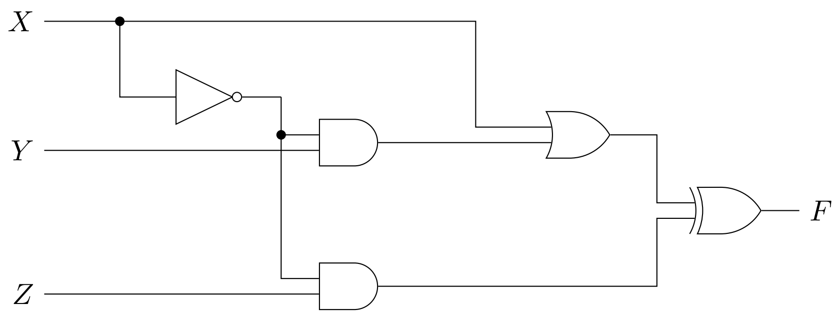
CS2002 Tutorial Week 5

Ian Gent

February 2023

1. Use DPLL to prove by *reductio ad absurdum* that the statement $\neg A \wedge \neg B$ follows from the three premises:
$$\neg A \vee \neg B$$
$$A \vee \neg B$$
$$\neg A \vee B$$

2. Consider the following circuit:



   (a) Give a logical expression for F in terms of X, Y and Z, and construct the truth table.

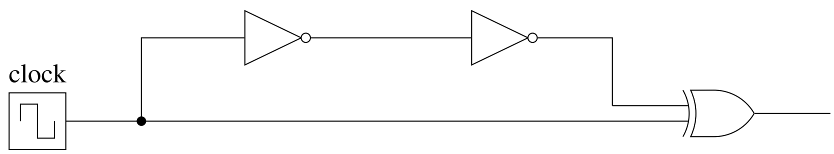
   (b) Draw an equivalent circuit with as few gates as possible

3. Consider the following truth table:

x	y	z	f_1	f_2
0	0	0	1	0
0	0	1	1	0
0	1	0	1	1
0	1	1	0	1
1	0	0	0	0
1	0	1	0	0
1	1	0	0	1
1	1	1	0	1

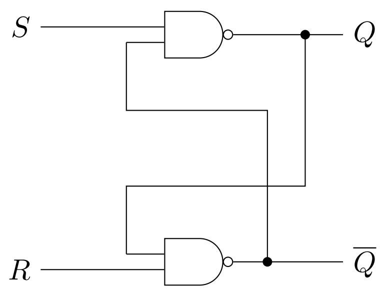
Devise a single circuit computing both f_1 and f_2 from x, y, z, with preferably few gates. Do this by first writing down a formula in CNF, given the truth table, and then simplifying it with the laws of Boolean algebra.

4. What does the following circuit do? Take into account that a signal takes a measurable time to propagate through a gate.



Verify your answer using LogicSim, by running the simulation on `delay.lsim` from the `Examples` directory.

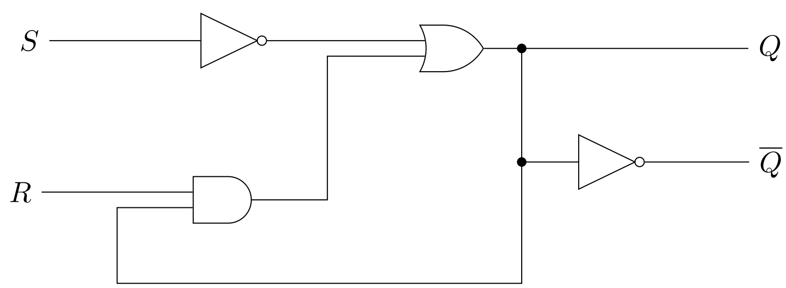
5. This is the SR NAND latch from lectures:



Suppose both R and S inputs equal 0. In normal behaviour, one of R and S changes to 1 first, and then the other one. Explain what happens in this case.

In the 'unstable' case of R and S both changing from 0 to 1 at the same time, what happens and why?

6. Consider the following as an alternative SR latch to the one above:



Analyse this circuit using a truth table, distinguishing between Q_{old} and Q_{new}, and using a finite automaton. Does it behave like the version from lectures? Do any differences matter?

7. The binary coded decimal (BCD) representation of integers uses four bits for each decimal digit. So, for instance, 13_{10} would be represented in 8 bits by $0001\ 0011$.

Give a digital circuit *with as few gates as possible* that multiplies a 4-bit BCD input number by 5, giving an 8-bit (2-digit) BCD output.

*H*int: This is somewhat of a trick question, as we can do it without any gates at all! See if you can work out how.