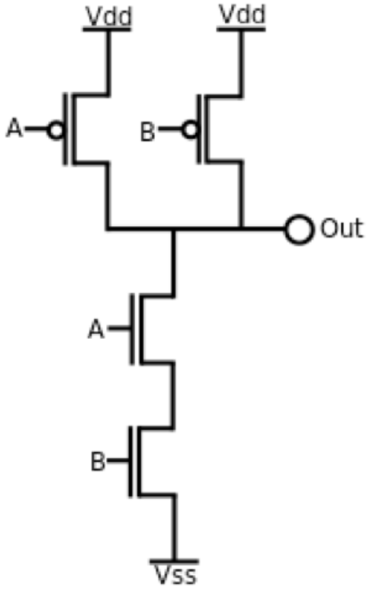
ECS404 Computer Systems and Networks

Lab Sheet
Week 2
Tuesday 4th October 2016

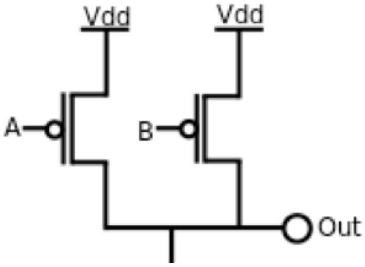
Transistors and Logic Gates

1. Consider the **nand gate** given in lectures.



Divide it into a top and a bottom half.

Top half:



The transistors here are in *parallel.*

a. Which of the two transistors have to be on for there to be a connection from Vdd to Out?

If either one of the transistors is open then there is a connection.

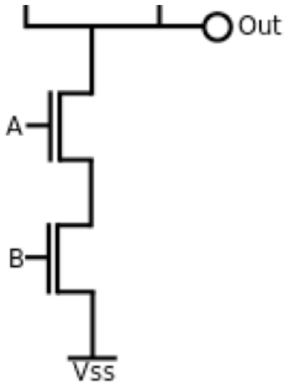
b. Are these pnp or npn transistors?

These are pnp transistors (circle = not = n). The gate type is in the middle of the name.

c. For which values of A and B is there a connection from Vdd to Out?

A=0 and B=0, A=0 and B=1, A=1 and B=0

Bottom half:



The transistors here are in *series.*

d. Which of the two transistors have to be on for there to be a connection from Vss to Out?
   Both need to be open.
e. Are these pnp or npn transistors?
   They are npn (no circle).
f. For which values of A and B is there a connection from Vss to Out?
   A=1 and B=1 is the only possibility.
g. Verify by listing all possible combinations of input values that there is a connection at the top iff there is no connection at the bottom.
   Possibilities are:
   A=0 and B=0: connection top not bottom
   A=0 and B=1; connection top not bottom
   A=1 and B=0: connection top not bottom
   A=1 and B=1: connection bottom not top
   So one connection is always open and never both.
h. Verify that the top half has a connection if and only if *not (A and B)* is *true.*
   Not(A and B) is *true* if and only if *(A and B)* is *false,* which holds if and only if at least one of *A* and *B* is *false.*
i. Verify that the bottom half has a connection if and only if *not (A and B)* is *false.*
   Not(A and B) is *false* if and only if *(A and B)* is *true,* which holds if and nonly if both *A* and *B* are *true.*

The gates we design will have top and bottom halves. The top half connects the output to Vdd=1 and the bottom to Vss=0. They are complementary. The complement of a gate half can be obtained by:

- Changing pnp transistors to npn and vice versa (reversing the switching)
- Changing transistors in parallel to transistors in series and vice versa.

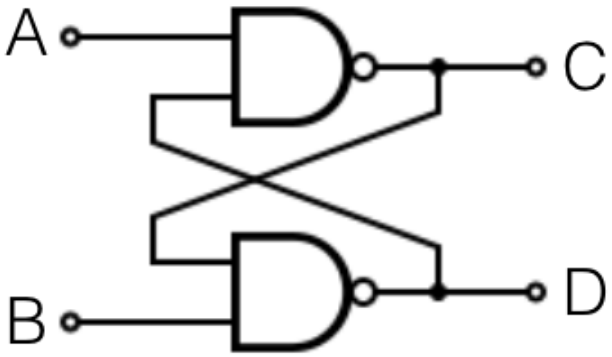
This is linked to the De Morgan laws in logic.

2. Produce a similar design that functions as an and gate.
    a. Design a top half that provides a connection between Vdd and Out if and only if *(A and B)* is *true.*
    Take the bottom half of the nand gate and connect to Vdd instead of Vss.
    b. Design a bottom half that provides a connection between Vss and Out if and only if *(A and B)* is *false.*
    Take the top half of the nand gate and connect to Vss instead of Vdd.
    c. Put these together to construct an and gate of similar design to the nand gate.

3. Produce a design that functions as an or gate.
    a. Design a top half that provides a connection between Vdd and Out if and only if *(A or B)* is *true.*
    Take the top half of the nand gate and use npn transistors instead of pnp.
    b. Design a bottom half that provides a connection between Vss and Out if and only if *(A or B)* is *false.*
    Take the bottom half of the nand gate and use pnp transistors instead of npn.
    c. Put these together to construct an or gate of similar design to the nand gate.

Flip flops and memory

4. Check the possible stable configurations of the flip flop:



    a. Show that if both A and B are 1, and D is 0 then C is 1.
    If D is 0, then the inputs to the top nand gate are 1 (from A) and 0 (from D). If any input to a nand gate is 0 then the nand gate produces 1. Hence C is 1. We do not use the fact that B is 1, but note that the inputs to the lower nand gate are 1 (from C) and 1 (from B). In this state the nand gate produces 0, which is the value at D. So the flip flop is in a stable configuration.
    b. Show that if both A and B are 1 and D is 1, then C is 0.

If D is 1, then the inputs to the top nand gate 1 (from A) and 1 (from D). If both inputs to a nand gate are 1, then the nand gate produces 0. Hence C=0. Again we do not use the fact that B is 1. Note that the inputs to the lower nand gate are 0 (from C) and 1 (from B), In this state the nand gate produces 1, which is the value at D. So once again the flip flop is in a stable configuration.

c. Deduce that it is not possible for A and B to both be 1 and C and D to both be 0.
We have just shown that if A and B are 1 and D is 0, then C is 1. Hence it is not possible for it to be 0.

d. Check that if A and B are both 1 and C and D also both 1 then both nand gates are in states that are not allowed. Why are they not allowed?
If all of A,B,C and D are 1, then both nand gates have 1 and 1 as input and are producing 1 as output. This is not allowed (not a stable configuration) for a nand gate.

e. Check that if A and B are both 1 and C and D also both 0 then both nand gates are in states that are not allowed. Why are they not allowed?
If A=B=1 and C=D=0, then both nand gates have a 1 and a 0 for input, so both should produce 1. But their outputs are given as 0. This is not a stable configuration.

5. Design a circuit with two inputs AA and BB, and two outputs A and B that has the following behaviour: when AA = 0 and BB = 1 then A = 1 and B = 0; when AA = 1 and BB = 1 then A = 0 and B = 1; when BB = 0 then A = 1 and B = 1.
This question is the hardest part of the lab because you have to design circuits that are a bit different from those in the nand gate, and because you are asked for circuits that have two outputs. The simplest thing is to design separate circuits to produce each output.
Let's start with computing B: B=1 when BB=0 or when AA=1 and BB=1. And B=0 when BB=1 and AA=0. So B can be computed as *(not BB or A)*. This can be computed using the or gate from question 3, changing the polarities of the transistors for the B input.
Now for A: A=1 when AA=0 and BB=1 or BB=0. There are three possibilities here which we should regroup: AA=0 and BB=0, AA=0 and BB=1, AA=1 and BB=0. This is *(AA nand BB).*

6. Using a flip flop and the circuit you have just designed, produce a memory cell with inputs AA and BB and outputs C and D. When BB=1 the value on AA is stored in the memory cell. When BB=0, the value in the cell is left unchanged. The current value in the cell is always readable on D.
You can just plug in your new circuit onto the front of the flip flop.

ITL Systems:

We will be using the Linux system as a default this term, and for some of the exercises this term (notably on networking), Windows is not a real option. If you are NOT familiar with Linux (if you have not been using it in Procedural Programming):

7. Reboot your lab machine into Linux and familiarise yourself with the system.
    a. Locate and launch a web browser (Firefox),
    b. log into QMPlus and access the ECS404 web page
    c. find the discussion forum and read the welcome message.
    d. Locate and launch the email client (Thunderbird)… or access College email via browser.
    e. Locate and launch a terminal window.

Powers of 2 and exponentiation:
The powers of 2 are very important in Computer Science (and not just because computers use binary). You should know all the powers of 2 up to 1024.

8. Complete the following table:

n	2^n	n	2^n
0	1	8	256
1	2	9	512
2	4	10	1024
3	8	11	2048
4	16	12	4096
5	32	20	1,048,576
6	64		
7	128		

Notice that 2^{10} is very close to 1000 (to 2 significant figures), and 2^{20} is 1,000,000 to 2 significant figures.

Petabytes and other phenomena:

9. You will be familiar with kilo{bytes,grams,metres}, and also mega{bytes} as well as milli{grams,metres,litres}.
    f. What do the prefixes: kilo, mega, giga, tera and peta mean? Similarly what do the prefixes milli, micro, nano mean?

Kilo = * 10^3
Mega = * 10^6
Giga = * 10^9
Tera = * 10^{12}
Peta = * 10^{15}
So when people say that experiments of CERN are producing petabytes of data, they mean that it would take thousands of laptop disks to store it on (since they are about a terabyte).

Similarly
Milli = * 10^{-3}
Micro = * 10^{-6}
Nano = * 10^{-9}
And then there are pico and femto corresponding to Tera and Peta respectively.

   g. Using Google, find out the size of a silicon atom: Google again and it's about 0.2nm.
   h. What is the current feature size in chip fabrication? (see eg www.intel.com). How many silicon atoms is this? We're now down to 14 nanometre feature sizes, which means that the "wires" in a modern chip are around 70 atoms wide.

Latency and clock cycles:

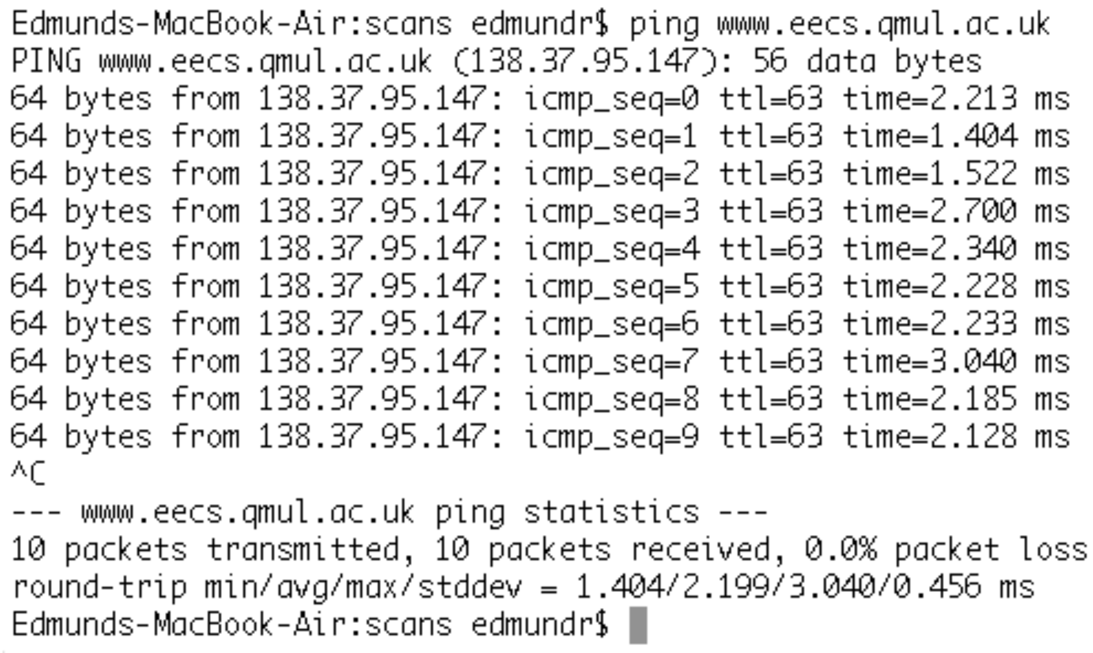
10. A typical computer clock speed is 2 GHz. How long (in seconds) is a single clock cycle? (Yes, this is easy).
The frequency Hertz (Hz) means per second. 2 GHz means 2 * 10^9 per second = 2,000,000,000 per second. If the clock speed is 2 GHz, that means it is "ticking" at 2 GHz, which in the computer context is producing a pulse 2 billion times a second. The clock cycle is the time between "ticks" or pulses, which is (1/2,000,000,000) = ½ * 10^{-9}, or half a nanosecond.
11. If light travels at 3 * 10^8 ms^{-1}, how far does light travel in a single clock cycle? You just need to multiply speed by time: 3 * 10^8 * ½ * 10^{-9} = 1.5 * 10^{-1} m, which is 15 cm. So what this means is that in the time it takes light (or anything else) to get from one side of a computer to another, the cpu has gone through several clock cycles. And if you are trying to get information off a device on the network some distance away, then we are talking about it taking thousands or millions of clock cycles to arrive (or more… see later).

12. If the arm on a computer disk takes one hundredth of a second to move between tracks, how many clock cycles of the cpu is this? If there are 2,000,000,000 clock cycles in one second, then there are 20,000,000 in one hundredth of a second.
13. If a computer disk spins at 7200rpm, how long (in seconds) does it take to make a single rotation? And how many clock cycles is this? Rpm is revolutions per minute, so 7200 rpm is 7200/60 = 120 revolutions per second. Each revolution takes 1/120 seconds, which is 2,000,000,000 * 1/120 clock cycles, which comes out to about 16,000,000.

14. What is the expected (average) latency of the computer disk described above, just concentrating on the physical causes. How many clock cycles is this? Quite a lot of you found this difficult, which is not surprising because you are not really familiar with the way a disk operates. The information on a disk is stored in a pattern of magnetic pulses, organised in circular tracks. When the head is in a particular position it can only read one track. There are a lot of tracks. So the odds are that if you ask for a random piece of information, the head is going to have to move. That takes 1/100 seconds, or 20,000,000 clock cycles. But even once it has moved it is very unlikely that the information will be just in the right position for the head to read it. It will probably have to come round under the head. On average, the disk will have to do half a rotation. That takes 1/240 seconds, or about 8,000,000 clock cycles.
15. Using the web, find out the average human reaction time (you can think of this as the latency of getting a response from a human). How many computer clock cycles is this?
    Very roughly reaction time is about 0.2 seconds, or 400 million clock cycles. People are slow.
16. Estimate how rapidly you type, and hence estimate the length of time between successive key strokes. (Hint: a very fast professional typist might reach 100 words a minute). How many computer clock cycles are there between successive key strokes? (NB When you get the answer you will realise one reason why computer windowing systems tend to split a key stroke into separate key press and key release events). Taking the professional typist as an example, and reckoning that words have about 5 letters each followed by a space, then we have roughly (5+1)*100 = 600 keystrokes a minute, or 10 a second. This leaves 200,000,000 clock cycles on average between keystrokes. People are still slow even when doing automatic actions.
17. Using the terminal window you opened earlier run the ping command to find out how long it takes for packets to make a round trip to different university servers. Type:

    ```
    ping www.eecs.qmul.ac.uk
    ```

    Wait for 8-10 cycles and then terminate the test by typing ctrl-c. You should get diagnostics of the form:



```
Edmunds-MacBook-Air:scans edmundr$ ping www.eecs.qmul.ac.uk
PING www.eecs.qmul.ac.uk (138.37.95.147): 56 data bytes
64 bytes from 138.37.95.147: icmp_seq=0 ttl=63 time=2.213 ms
64 bytes from 138.37.95.147: icmp_seq=1 ttl=63 time=1.404 ms
64 bytes from 138.37.95.147: icmp_seq=2 ttl=63 time=1.522 ms
64 bytes from 138.37.95.147: icmp_seq=3 ttl=63 time=2.700 ms
64 bytes from 138.37.95.147: icmp_seq=4 ttl=63 time=2.340 ms
64 bytes from 138.37.95.147: icmp_seq=5 ttl=63 time=2.228 ms
64 bytes from 138.37.95.147: icmp_seq=6 ttl=63 time=2.233 ms
64 bytes from 138.37.95.147: icmp_seq=7 ttl=63 time=3.040 ms
64 bytes from 138.37.95.147: icmp_seq=8 ttl=63 time=2.185 ms
64 bytes from 138.37.95.147: icmp_seq=9 ttl=63 time=2.128 ms
^C
--- www.eecs.qmul.ac.uk ping statistics ---
10 packets transmitted, 10 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 1.404/2.199/3.040/0.456 ms
Edmunds-MacBook-Air:scans edmundr$ 
```

What is the average time a packet took to travel to and from our webserver? How many clock cycles is this?
The average given her is 2.199 ms, which is just over 4 million clockcycles. This, however was taken when at home using commercial internet, so not close to the webserver. From QM I get 1.946 ms, or just under 4 million clock cycles.

Similarly test the web servers:
`www.cl.cam.ac.uk` (in Cambridge) 10.628 ms, 21 million clock cycles.
`www.aau.dk` (Aarhus University in Denmark) 32.072 ms, 64 million clock cycles
`www.cs.yale.edu` (Yale on the east coast of the US) 83.862 ms, 167 million clock cycles
`www.cs.stanford.edu` (Stanford on the west cost of the US) 178.86 ms, 356 million clock cycles.

How do these times compare to each other, and what does it say about the latency of accessing information held on the cloud.
You should notice that these are significant lengths of time in comparison with the clock speed, and that they get longer as the sites get further away.
How many clock cycles are there in a packet round trip to the US?

Estimate the length of time it would take for a packet travelling at the speed of light to reach the US (say Yale) and return. A rough estimate is fine. How does this compare to the times you have above, (i.e. by what factor are the packet times longer than this theoretical minimum)?
Say the east coast of the US is 6000km away, then it takes light 2 * 6000 * 1000 / (3 * 10^8) seconds to make the round trip (ping timings are for a packet to go and

an acknowledgement to come back). This is $4 * 10^6 * 10^{-8} = 4 * 10^{-2}$ seconds, or 40 ms. So the packet is travelling on average at about a quarter the speed of light.

Edmund Robinson