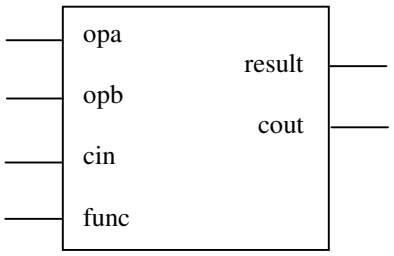
EECS 470 Homework 1

This assignment is worth a bit less than 2% of your grade in the class and is graded out of 30 points. Remember you may drop one homework assignment or quiz score.

1. Using a Karnaugh map, find the minimal sum-of-products solution for the following 1-bit ALU.



This module takes three inputs (opa, opb, and cin) and generates two outputs (result and cout) depending upon the value of func as follows:

func's value	result	cout
0	LSB of opa+opb+cin	MSB of opa+opb+cin
1	cin	opa

   a. Draw the K-map (complete with *clearly* shown coverings). [5]
   b. Draw the minimal sum-of-products solution using AND and OR gates. You may freely use inversion bubbles. [3]

2. Consider a typical 5-stage pipelined processor (Fetch, Decode, Execute, Memory and Write back) where:
   - Branches are predicted not-taken and resolved in Execute.
   - All possible forwarding cases exist. All forwarding results are sent to execute other than those needed by a load or a store that are not generated in Execute. Those are forwarded to the Memory stage. When forwarding cannot resolve the problem, the instruction stalls in Execute.
   - All instructions take 1 cycle to execute other than the multiply instruction which takes 2 cycles (and causes a stall.)
   - Address calculations are done in the EX stage.

a. How many cycles would it take before the following program would have halt enter the Memory stage? Clearly show your work by showing what instruction is in which stage at each cycle. [4]

```
      R1=MEM[R2+4]
      R4=MEM[R1+0]
      R2=R1+R4
      If (R1==R1) goto next
      R1=R1+R1
next: R2=R2*R2
      halt
```

b. What would be the CPI for a given program if:
   - The program consists of a LARGE number of instructions.
   - 20% of instructions are loads, 35% adds, 5% multiply, 20% branches and 20% stores
   - 60% of branches are not taken
   - Each instruction type is independent of what comes before it. (i.e. instruction distribution is random.)
   - Each *operand* has a 30% chance of being dependent on the result of the instruction in front of it.

   Clearly show your work. Your final answer should have 4 significant figures. [5]

3. Consider a 2-way associative, 32-KB cache with 16-byte cache lines. The address space is 32-bits with address line zero being the least significant. Which bits of the address are used for the offset? The index? The tag? [3]
4. Define the term "branch delay slot." Why are they used? [2]
5. Design an XOR gate using *only* NAND gates (no inversion bubbles!) [2]
6. In nanoseconds, what is the period of a 2GHz clock? In meters, how far can light travel (in a vacuum) in the time of that clock period? [2]

(These last two questions are not from prerequisite material. However, I will be referring to related ideas during class so I thought I'd add these in. A quick web search should turn up answers if you don't already know this stuff. Don't forget to provide cites as needed!)

7. In units of Ohm-meters, what is the resistivity of copper? Of aluminum? If you had a 1cm long wire that was 200nm in width and 300nm in height, what would be the resistance (in Ohms) of the wire if it were made of copper? If it were aluminum? [2]
8. Define the term "fringe effect" as it applies to capacitors. [2]