# Homework 4

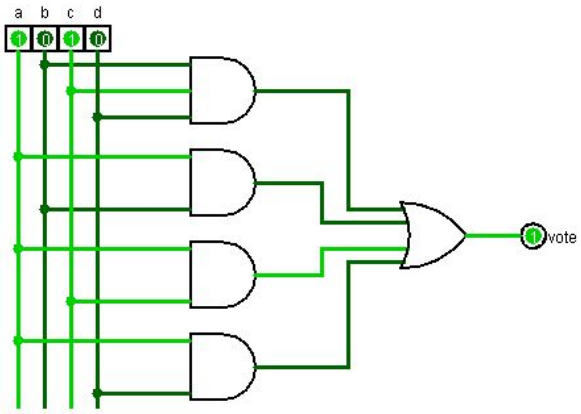
**Q. 1** A company's board consists of 4 people: Abigail (founder & CEO), Benoit, Charlie, and Debra
When they take a vote, if there is a majority (either up or down), that vote prevails.
If, however, there is a tie, then Abigail's vote is the tie-breaker.

| A | B | C | D | VOTE |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

**For a': the only contribution is from a'.b.c.d**
**For a: the only non-contribution is from a.b'.c'.d'**

**So the full expression is vote = b.c.d + a.(b + c + d)**
**- meaning the motion passes if all three of b, c & d vote for it, with a against;**
**or if a and any one of {b, c, d} vote for it.**
***(this can be deduced from our standard simplification rules, or simply by inspection)***

vote = b.c.d + a.(b + c + d)




**3.31**

3-bit addressing => address space = 2^3 = 8
8 byte adresability
(i.e. each location stores 8 bytes = 64 bits, so this is probably the register bank of a 64-bit processor)
Total memory = 8 locations * 8 bytes = 64 bytes total

**3.32**

A memory address refers to a location in memory (like a single "register")
Addressability refers to the number of bits that can be stored at that location

**3.33**

a) To read the 4th memory location: A[1:0] = 11, WE = 0
b) To address 60 locations requires 6-bit adressing
addresability is unchanged.
c) 2^6 = 64, so 4 more locations could be added

**3.34**

a) 16 bits
b) 4 bits
c) 0001

**3.35**

22-bit addressing => addres space = 2^22 = 4M
3 bit addresability (each location holds 3 bits - very strange memory system!!)
Total memory = 12M bits