CMPE 12

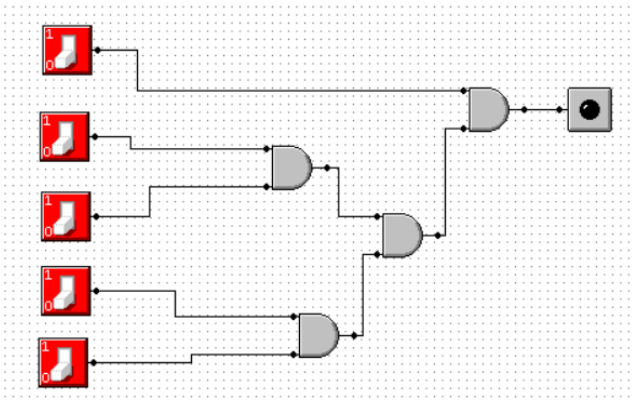
SECTION 01-C

Katerina Chinnappan

kachinna@ucsc.edu

1) (5 pts) Build a 5-input and gate out of 2-input and gates.

Answer:



2) (5 pts) How many output lines will a five-input decoder have?

Answer: 2^5 = 32→**So 32 output lines**

3) (5 pts) How many output lines will a 16-input multiplexer have? How many select lines will this multiplexer have?

Answer: 16 lines since 16 input.

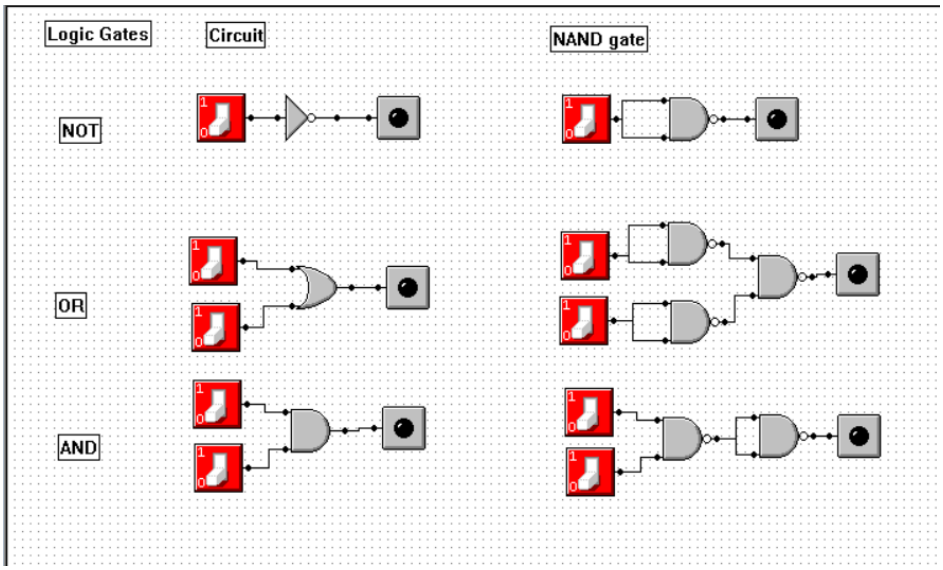
Log base (2) (16) = **4 output lines.**

4) (5 pts) You know a byte is 8 bits. We call a 4-bit quantity a nibble. if a byte-addressable memory has a 14-bit address, how many nibbles of storage are in this memory?

Answer: 14-bit address, so→ 2^14 = 16384. →16382 * 2 = **32768 nibbles**

5) (15 pts) All Logic circuits can be created by NAND gates. Prove this by building logic circuits for NOT, OR and AND using only NAND gates.

Answer:

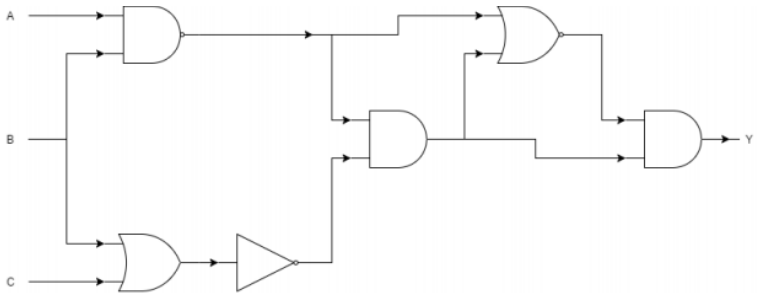


6) (10 pts) Distinguish between a memory address and the memory's addressability.

Answer: Memory is made up of many locations, where each location is unique and has the ability to store a value.

Addressability is the number of bits stored in each memory location.

7) (15 pts) Give the logic circuit below, fill in the truth table for the output value Y.



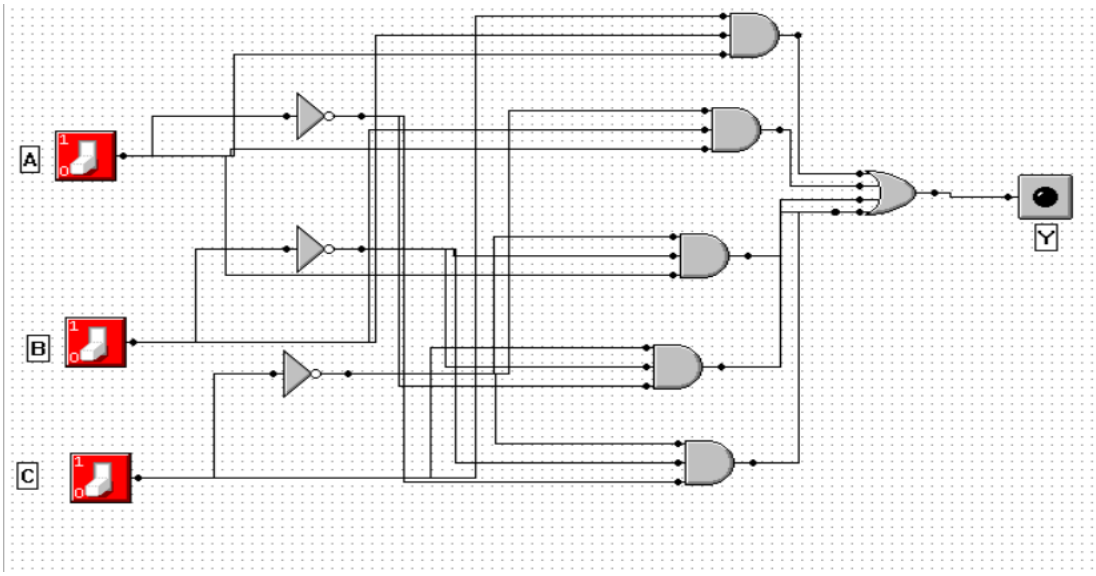
Answer:

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

8) (15 pts) Create the Logic gates for the truth Table below

A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Answer:



9) (25 pts) Convert the following numbers to binary and perform binary subtraction on them. Do not use additive inverse.

a. 39 - 22 b. 25 - 14 c. 39 - 12 d. 18 - 11 e. 30 – 26

Answer: (let's use 8 bit) ----------**REMEMBER: 1-0 = 0, 1-1 = 0, 0-0 = 0, 0-1 = 1 carry 1**

a) 1) convert 39 to binary→0010 0111
2) convert 22 to binary→0001 0110
3) subtract:

```
  0010 0111
- 0001 0110
-----------
  0001 0001
```

→**0001 0001 which is 17 in base 10.**

b) 1) convert 25 to binary→0001 1001
2) convert 14 to binary→0000 1110
3) subtract:

```
  0001 1001
- 0000 1110
-----------
  0000 1011
```

→**0000 1011 which is 11 in base 10.**

c) 1) convert 39 to binary→0010 0111
2) convert 12 to binary→0000 1100
3)subtract:

```
  0010 0111
- 0000 1100
-----------
  0001 1011
```

→**0001 1011 which is 27 in base 10.**

d) 1) convert 18 to binary→0001 0010
2) convert 11 to binary→0000 1011
3) subtract:

```
  0001 0010
- 0000 1011
-----------
  0000 0111
```

→**0000 0111which is 7 in base 10.**

e) 1) convert 30 to binary→0001 1110
2) convert 26 to binary→0001 1010
3) subtract:

```
  0001 1110
- 0001 1010
-----------
  0000 0100
```

→**0000 0100 which is 4 in base 10.**