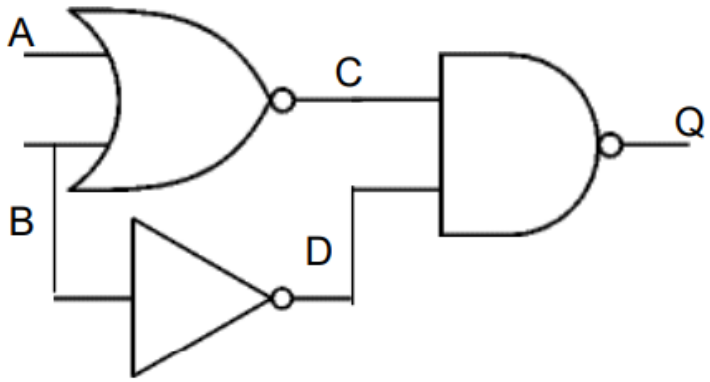
Week 5 (half term) worksheet

Logic

1. Observe the logic circuit below and create a truth table to find the possible outputs for Q. What should input A and B be for Q to turn on (output 1)?

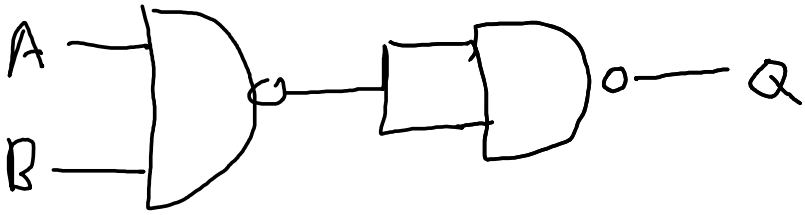


A	B	C	D	Q
0	0	1	1	0
0	1	0	0	1
1	0	0	1	1
1	1	0	0	1

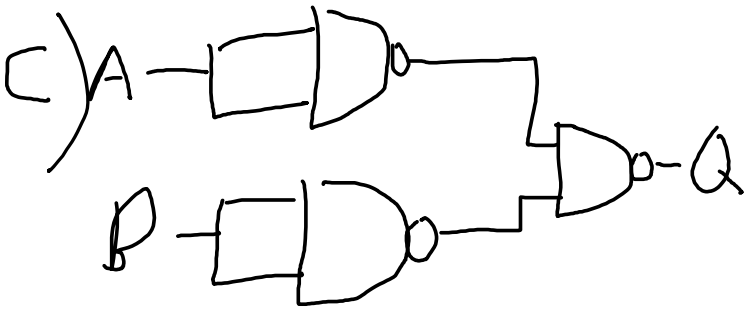
2. Draw only using NAND gate the equivalent to the following gates:
   a. NOT
   b. AND
   c. OR

a)

b)

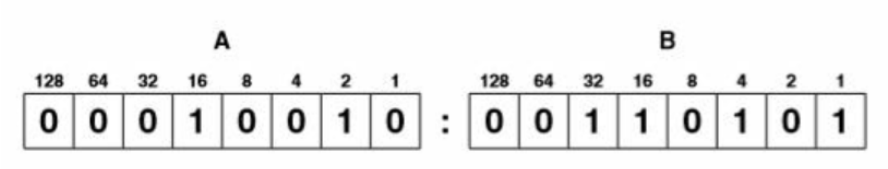


c)

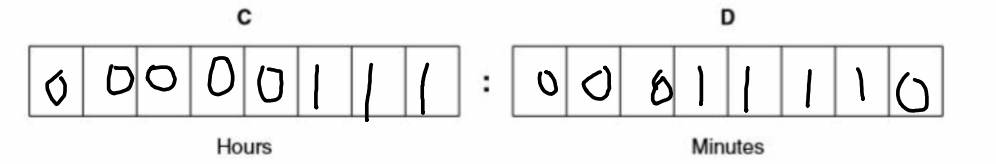


Binary

1. An alarm clock is controlled by a microprocessor. It uses the 24-hour clock. The hour is represented by 8-bit register **A**. The minutes is also represented by 8-bit register **B.**
   a. Identify what time is represented by the following.



   b. An alarm has been set for 7:30. two 8-bit registers **C** and **D** have been setup to show the hour and minute. Show how the hour and minute would be represented. Your answer should follow the below format:



2. Answer the following questions:

a. Add the following two 8-bit binary values.

```
  1 1 0 0 0 1 1 0 1
+ 0 0 1 1 0 1 1 0
  1 1 0 0 0 0 1 1
```

b. Add the following two 8-bit binary values.

```
  1 0 1 0 0 1 1 0
+ 1 1 0 0 0 1 0 1
  1 0 1 1 0 1 1 1
  0 1 1 0 1 0 1 1
```

c. An overflow error can occur when adding two 8-bit binary values. What is meant by an overflow error? Use the previous calculation as an example if you wish.

There is more bits then there can be stored. The result exceeds 8 bits and the result is more than 1111 1111

LMC (reading)

1. Observe the 2 codes below and explain what they do.
    a. What is the output when the inputs are 12 and 20 respectively.

```
          inp
          sta A
          inp
          sta B
          lda A
          sub B
          brp ispositive
          lda B
          out
          hlt
ispositive LDA A
          out
          hlt
A         dat
B         dat
```

12
20
12
12−20=−8
20
20

The output is 20 and it prints out the biggest number of the two numbers.

    b. What is the output when the input is 13 and 5 respectively?

```
          INP
          STA  arg1
          INP
          STA  arg2
          LDA  arg1
loop      SUB  arg2
          BRP  loop
          ADD  arg2
          OUT
arg1      DAT
arg2      DAT
```

13
13−5=8
8−5=3
3−5=−2
3

It outputs 3 and returns the remainder when the two numbers are divided.

LMC (writing)

Write down (you can screenshot your LMC) an assembly language that does the following:

a. Countdown: input a number x and output the countdown x until 0. (e.g. x =3 ; output 3 2 1 0)

```
    INP
    STA A

LOOP LDA A
    OUT
    SUB ONE
    STA A
    BRZ END
    BRA LOOP
END HLT

A DAT
ONE DAT 1
```

b. Input 2 numbers x and y and perform a multiplication operation.
Here is the trick:
2 x 3 = 6
So, 2 + 2 + 2 = 6
3 x 4 = 12
So, 3 + 3 + 3 + 3 = 12

So, with the aid of a loop we keep adding the **first** number as many as **second** number of times!

```
 INP
STA FIRST
INP
STA SECOND

LDA ZERO
STA PRODUCT
STA COUNTER

LDA SECOND
STA COUNTER

LOOP  BRZ END
LDA PRODUCT
ADD FIRST
STA PRODUCT

LDA COUNTER
SUB ONE
STA COUNTER

BRA LOOP
```

```
END LDA PRODUCT
      OUT
       HLT

ONE DAT 1
ZERO DAT 0
FIRST DAT
SECOND DAT
PRODUCT DAT
COUNTER DAT
```