

Instructor: Dr. G.E. Antoniou

October 28, 2018

Day

Homework # 3

Dos Santos Ferreira, Joao Paulo

CSIT 230\_02 – Computer Systems

Department of CSIT

# Problem 1: ROM DESIGN

1. Arithmetic component

Design a Read Only Memory (ROM) [Decoder plus OR gates]. The logic  
circuit accepts a three–bit number and generates an output binary number  
equal to four times the input number.

i. What is the size of the initial (unsimplified) ROM ?  
ii. What is the size of the final (simplified) ROM ?  
iii. Show in detail the final memory layout.

**Solution (a):**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **N2** | **N1** | **N0** | **M4** | **M3** | **M2** | **M1** | **M0** |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |

**M0 = 0**

**M1 = 0**

**M2 = N0**

**M3 = N1**

**M4 = N2**

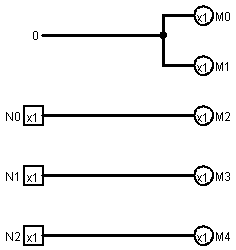
**i. What is the size of the initial (unsimplified) ROM ?**

The size of the ROM is given by the equation , where n is the number of inputs and m is the number of outputs. From the unsimplifed version of this problem we have three inputs (the combination of 3-bit numbers) and 5 outputs for a unsimplified ROM of size **.**

**ii. What is the size of the final (simplified) ROM ?**

We can simplify the ROM by checking for patterns between the output and input. In this case we see that **M0** and **M1**are always 0, **M2** is the same as **N0**, **M3** is the same as **N1**, and **M4** is the same as **N2**. Therefore for the simplified ROM we would still have 3 inputs but we would not need an output from the ROM since the output can either be expressed with much simpler expressions. The size of the simplified ROM would be **,** this means we do not need a ROM at all for this problem.

**iii. Show in detail the final memory layout.**



1. RGB LED’s

Three light-emitting diodes [LEDs] (one red, one green, one blue) turn on  
when a number 0-7 is passed through. Red turns on with even numbers,  
green turns on with odd numbers, blue turns on with multiples of 3. Zero  
means they are all off, seven means they are all on.

i. What is the size of the initial (unsimplified) ROM ?  
ii. What is the size of the final (simplified) ROM ?  
iii. Show in detail the final memory layout.

**Solution (b):**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **N2** | **N1** | **N0** | **R** | **G** | **B** |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 |

**i. What is the size of the initial (unsimplified) ROM ?**

From the unsimplified version of this problem we have three inputs (the combination of 3-bit numbers) and 3 outputs for a unsimplified ROM of size **.**

**ii. What is the size of the final (simplified) ROM ?**

We can simplify the ROM by checking for patterns between the output and input. In this case only **G** can be easily represented since **G** is the same as **N0**. Therefore for the simplified ROM we would still have 3 inputs but we would not need 2 outputs from the ROM. The size of the simplified ROM would be **.**

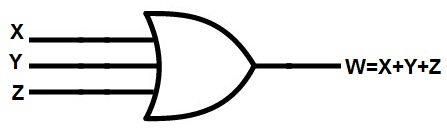
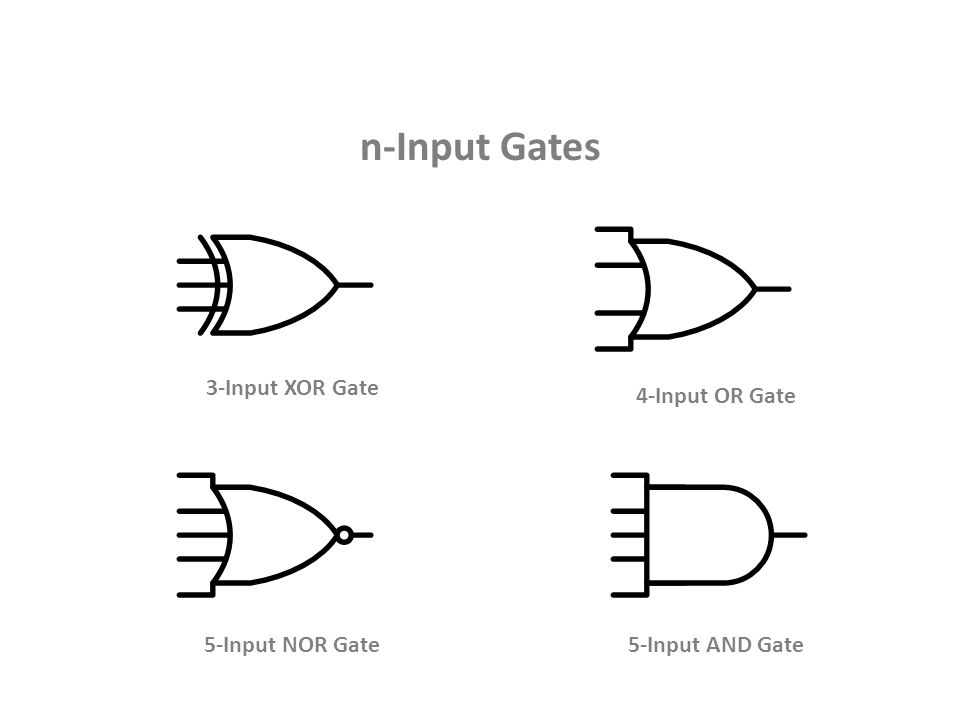
**iii. Show in detail the final memory layout.**

3-8 Decoder

**N2**

**N1**

**N0**



**G R B**

# Problem 2: ALU Design

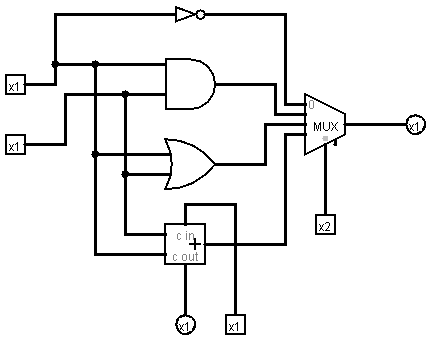
Design (step–by–step) and implement an 1–bit Arithmetic Logic Unit (ALU)  
that will perform the following arithmetic and logical operations:

*•* NOT b  
*•* a AND b  
*•* a OR b  
*•* a **+** b  
*•* a **-** b

Only one–Adder should be used for both Add (+) and Sub (–) operations.  
(a) Test the final design, using *LogiSim*, with just one set of data

**Solution:**

There are a total of 5 operations that needed to be performed by the ALU, however we can simplify it to 4 by using one adder to perform both a+b and a-b. In this case we will need a 4-1 Multiplexer with 2 bits for the select line. 00 will perform the NOT b operation, 01 will perform a AND b, 10 will perform a OR b and 11 will perform a+b and a-b. The full design by using Logisim is as follow:



# Problem 3: RAM Design

Design an 8*K ×* 8 RAM memory system, **using** 1*K ×* 8 memory chips.

(a) Draw and Explain the Memory Architecture

(b) Number of Data Bus lines?  
(c) Number of Address Bus lines?

**Solution:**

Since we are using 1K x 8 memory chips we will need 8 of those chips to design a 8K x 8 RAM. Also the address bus should contain at least 10 bits since for a 1k = . Since we have 8 chips we will need a 3-8 decoder to select the chip, increasing the address bus by 3 bits for a total of 13 bits for the Address Bus. The Data Bus will contain only 8 bits since the chips are 1K x 8.

**Result:**

(a)

A10, A11, A12

Address Bus: A0 – A9

W/R

CS

1K

1K

CS

CS

1K

1K

CS

CS

1K

1K

CS

CS

1K

1K

CS

3-8

**Decoder**

Data Bus: D0 – D7

(b) The Data Bus contains 8 bits

(c) The Address Bus contains 13 bits