

**Problem 1:**

**Computer Science or Information Technology**

Instructor: Dr. G.E. Antoniou

Day, Month, Year

Day

CSIT 502

Department of CSIT

Assessment

Module-4

Hidalgo, Rafael

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

(a) What is the size (number of bits) of the initial (unsimplified) ROM ?

(b) What is the size (number of bits) of the final (simplified) ROM ?

(c) Show in detail the final memory layout

**Solution**

(a)

First, to determine what decoder we will have to Implement, we must take the input for the decoder (n) to figure out what the output of the decoder will be, which is determined by (2^n).

Therefore, our decoder will be a 3 to 8 decoder.

To figure out how many or gates we will be using, we first need to make a truth table. Before we do that, we need to figure out what is the largest number of bits per word need for the RO. Since our largest output will be four times the input, the largest input we can have is 7 in decimal form, or 111 in binary. Four times 7 is 28 in decimal form (11100 in binary). Therefore, the maximum number of bits will be five. Thus, the truth table is constructed as shown below.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D0** | **D1** | **D2** | **D3** | **D4** |
| **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** |

For this unsimplified truth table, the maximum number of bits for the ROM is

(2^3) \* 5, which equals to 40 bits

(b) What is the size (number of bits) of the final (simplified) ROM ?

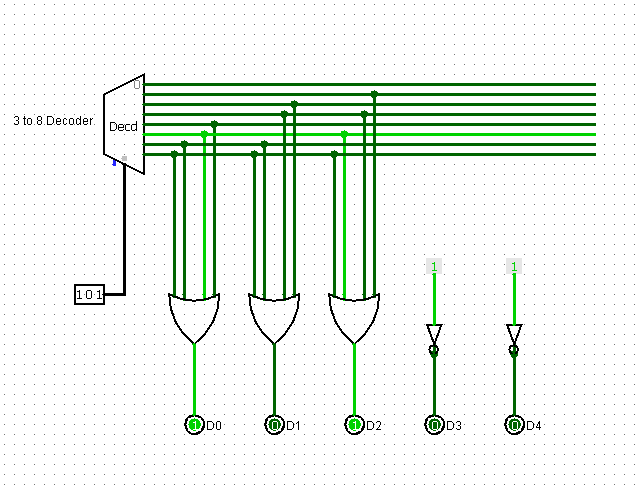
To simplify this truth table, we simply remove the final two columns since they are comprised of purely 0’s.

This makes the size of this ROM (2^3) \* 3, which equals to 24 bits.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D0** | **D1** | **D2** |
| **0** | **0** | **0** | **0** | **0** | **0** |
| **0** | **0** | **1** | **0** | **0** | **1** |
| **0** | **1** | **0** | **0** | **1** | **0** |
| **0** | **1** | **1** | **0** | **1** | **1** |
| **1** | **0** | **0** | **1** | **0** | **0** |
| **1** | **0** | **1** | **1** | **0** | **1** |
| **1** | **1** | **0** | **1** | **1** | **0** |
| **1** | **1** | **1** | **1** | **1** | **1** |

(c) Show in detail the final memory layout

Thus, the final memory layout is as follows.



**Problem 2:**

Design a digital logic circuit as a Read Only Memory (ROM) [Decoder plus OR gates]: 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.

(a) What is the size (number of bits) of the initial (unsimplified) ROM ?

(b) What is the size (number of bits) of the final (simplified/smallest size) ROM ?

(c) Show in detail the final memory layout.

**Solution**

(a)

The truth table for the problem is outlined below.

There will be three inputs and therefore the rom will have 2^3 words. There are three outputs since the word size is three.

Thus the size of the ROM for this unsimplified truth table is (2^3)\*(3) which equals to 24 bits.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **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** |

(b)

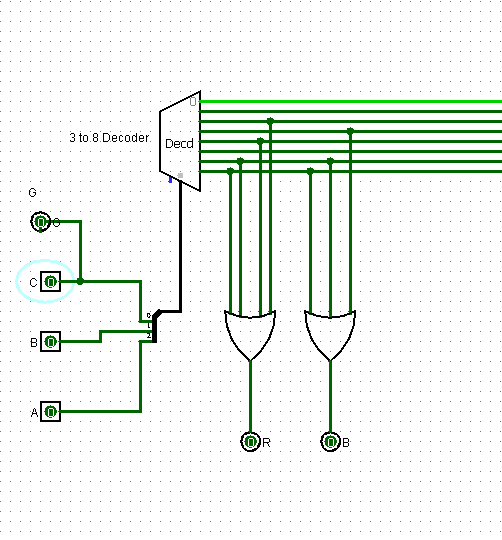
This ROM can be further simplified due G being equal to C.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **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** |

Therefore, we can omit the G OR gate, thereby making our new ROM (2^3) \* 2 or 16 bits large.

(c)

Thus the final memory layout is as follows.



**Problem 3:**

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

•NOT b

• a AND b

• a OR b

• a NAND b

• a NOR b

• a XOR b

• a XNOR b

(b) Arithmetic operations

• a + b

• a - b

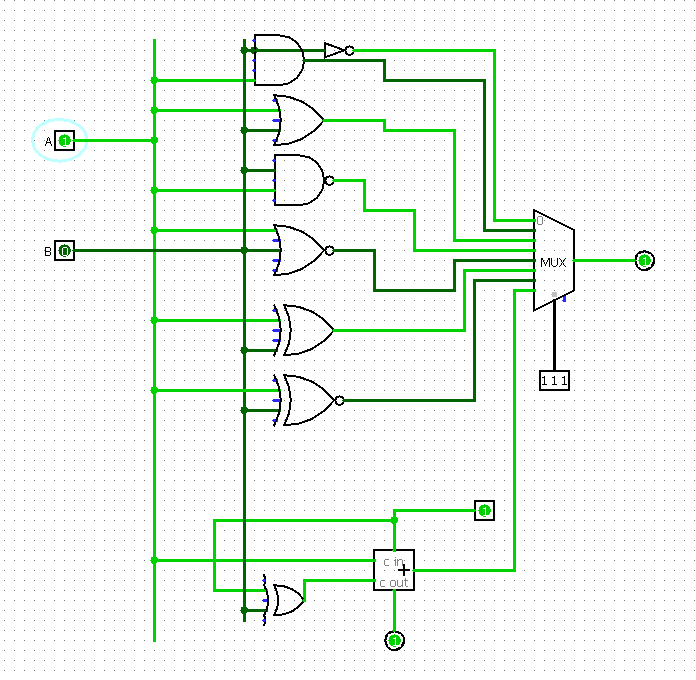
Only one adder should be used for both Add (+) and Sub (–) operations.

(a) Test the final design with one set of data. (The LogiSim circuit should be active)

**Solution**

There will need to be 3 selectors to account for all 8 operations. The truth table for the selectors will look like follows. Also the diagram is pasted below the table.

|  |  |  |  |
| --- | --- | --- | --- |
| **S0** | **S1** | **S2** | **Out** |
| **0** | **0** | **0** | NOT b |
| **0** | **0** | **1** | a AND b |
| **0** | **1** | **0** | a OR b |
| **0** | **1** | **1** | a NAND b |
| **1** | **0** | **0** | a NOR b |
| **1** | **0** | **1** | a XOR b |
| **1** | **1** | **0** | a XNOR b |
| **1** | **1** | **1** | **Add & Sub** |



**Problem 4:**

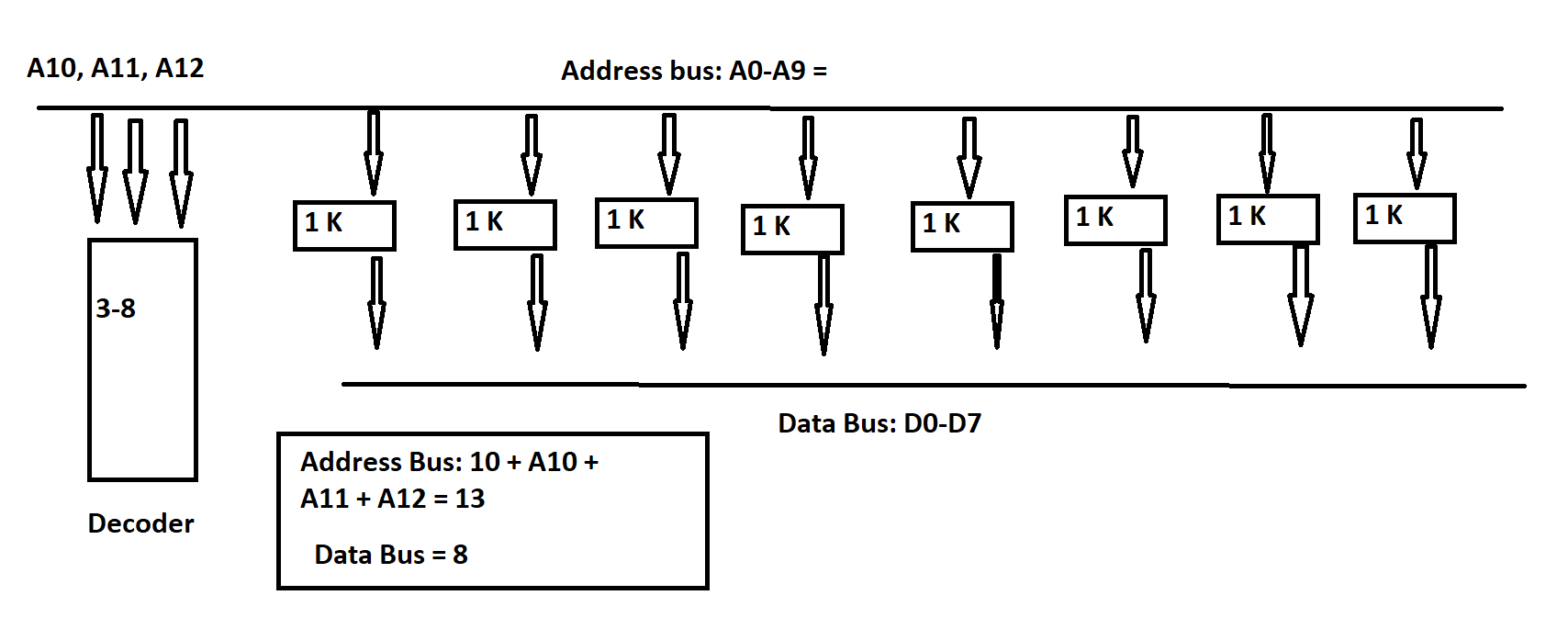
Design a 8K × 8 RAM (memory) system, using 1K × 8 RAM chips.

(a) Number of Data Bus lines?

(b) Number of Address Bus lines?

(c) Draw and briefly explain the overall memory architecture layout.

**Solution**



1. As you can see form the above diagram, the data bus lines will be 8 since all of the memory chips will feed through 8 lines.
2. The number of address lines is 10 + A10 + A11 + A12, which equals to 13.
3. See above for the drawing. The chips need 10 address lines since n in 2^n = 1024 equals to 10. Three is added since the decoder will need three address bus lines to assign addresses to the chips.