

**Information Technology**

Instructor: Dr. G.E. Antoniou

Day, Month, Year

Day

CSIT230\_SP21, 03

Department of CSIT

FINAL

PROJECT

Rana, Karan

**Problem 1**

1.

Design a digital logic circuit (all steps are required), using minimal amount of gates, that will realize a digital logic circuit corresponding to the following *A097803 polynomial* :

*A* = 3(2*x*2 + 1)*, with input* (*x*) : 0 *≤* (*x*) *≤* 7

(a) Set–up the truth–table

[Binary inputs: (*x*); Binary outputs: *A*1*, A*2*, A*3*, A*4*, ...*]

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **X1** | **X2** | **X3** | **A** | **A1** | **A2** | **A3** | **A4** | **A5** | **A6** | **A7** | **A8** | **A9** |
| **0** | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| **1** | 0 | 0 | 1 | 9 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| **2** | 0 | 1 | 0 | 27 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| **3** | 0 | 1 | 1 | 57 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| **4** | 0 | 0 | 0 | 99 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| **5** | 0 | 0 | 1 | 153 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| **6** | 0 | 1 | 0 | 219 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| **7** | 0 | 1 | 1 | 297 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |

(b) Derive the output expressions

A1 = x1x2x3

A2 = x1x2'x3 + x1x2x3'

A3 = x1x2'x3' + x1x2x3'

A4 = x1'x2x3 + x1x2'x3' + x1x2x3

A5 = x1'x2x3' + x1'x2x3 + x1x2'x3 + x1x2x3'

A6 = x1'x2'x3 + x1'x2x3' + x1'x2x3 + x1x2'x3 + x1x2x3' + x1x2x3

A7 = 0

A8 = x1'x2'x3' + x1'x2x3' + x1x2'x3' + x1x2x3'

(c) Simplify (optimally) the derived expressions, USING ONLY K–Maps

A1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **00** | **01** | **11** | **10** |
| **0** | 0 | 0 | 0 | 0 |
| **1** | 0 | 0 | 1 | 0 |

A2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 00 | 01 | 11 | 10 |
| **0** | 0 | 0 | 0 | 0 |
| **1** | 0 | 1 | 0 | 1 |

A3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **00** | **01** | **11** | **10** |
| **0** | 0 | 0 | 0 | 0 |
| **1** | 1 | 0 | 0 | 1 |

A4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **00** | **01** | **11** | **10** |
| **0** | 0 | 0 | 1 | 0 |
| **1** | 1 | 0 | 1 | 0 |

A5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **00** | **01** | **11** | **10** |
| **0** | 0 | 0 | 1 | 1 |
| **1** | 0 | 1 | 0 | 1 |

A6

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **00** | **01** | **11** | **10** |
| **0** | 0 | 1 | 1 | 1 |
| **1** | 0 | 1 | 1 | 1 |

A7

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **00** | **01** | **11** | **10** |
| **0** | 0 | 0 | 0 | 0 |
| **1** | 0 | 0 | 0 | 0 |

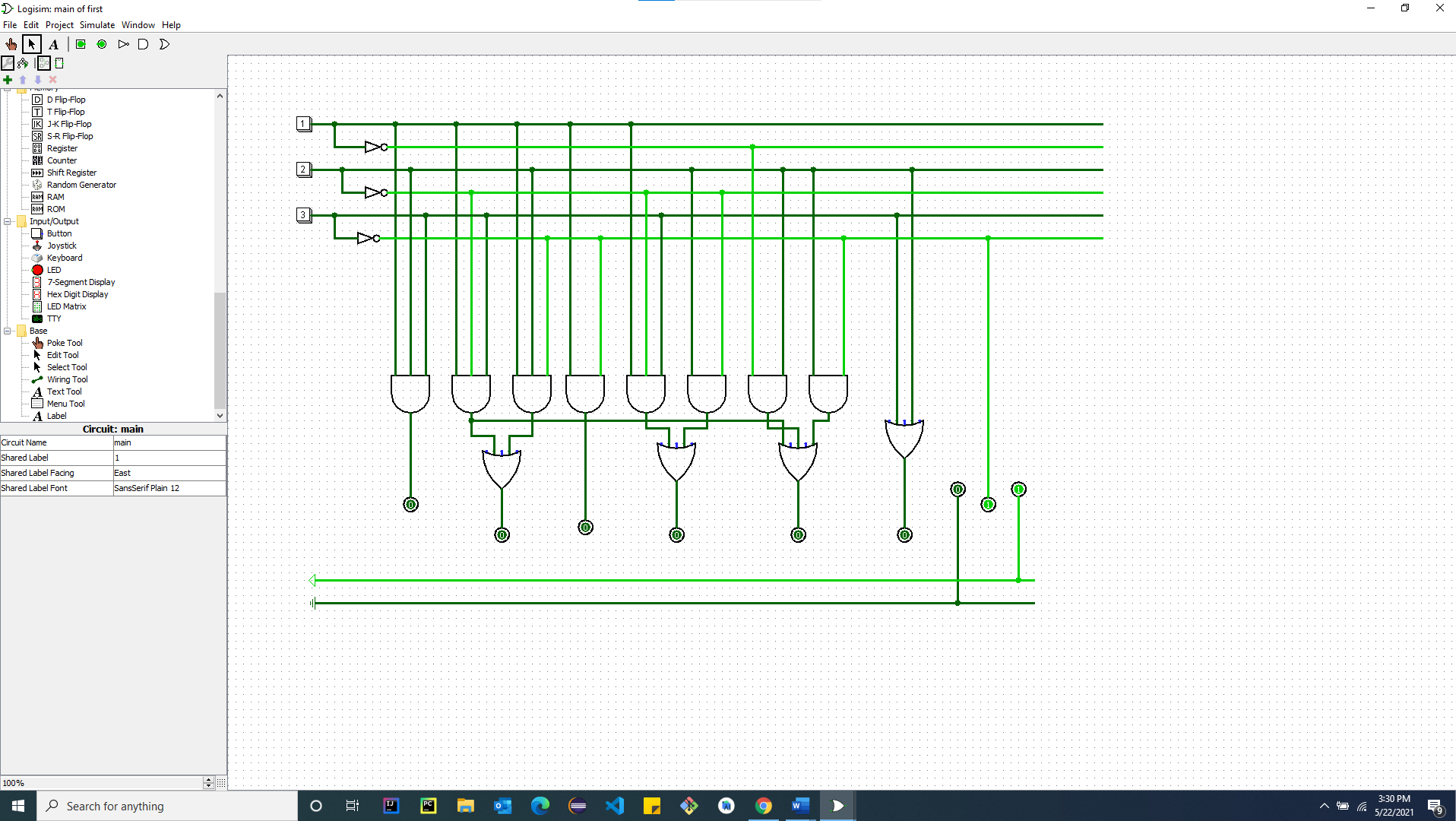
A8

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **00** | **01** | **11** | **10** |
| **0** | 1 | 0 | 0 | 1 |
| **1** | 1 | 0 | 0 | 1 |

A9

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **00** | **01** | **11** | **10** |
| **0** | 1 | 1 | 1 | 1 |
| **1** | 1 | 1 | 1 | 1 |

(d) Implement the simplified digital logic circuit, using *LogiSim*.



**Problem 2**

2. Design a Read Only Memory (ROM) to implement the following, *A097803*, polynomial

*A* = 3(2*x*2 + 1)*, with input* (*x*) : 0 *≤* (*x*) *≤* 7

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

Knowing that it is a ROM we need 4-8 decoders and 16 or gates are needed.

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

When the input X = 0

-> 3(2\*(0)^2 + 1)

= 0

The same thing goes on when X = 1,2,3,4,5,6,7,8

X= 1

= 3(2\*(1)^2 + 1)

= 9

X=2

= 3(2\*(2)^2 + 1)

= 27

X = 3

= 3(2\*(3)^2 + 1)

= 57

X = 4

= 3(2\*(4)^2 + 1)

= 99

X =5

=3(2\*(5)^2 + 1)

= 153

X = 6

= 3(2\*(6)^2 + 1)

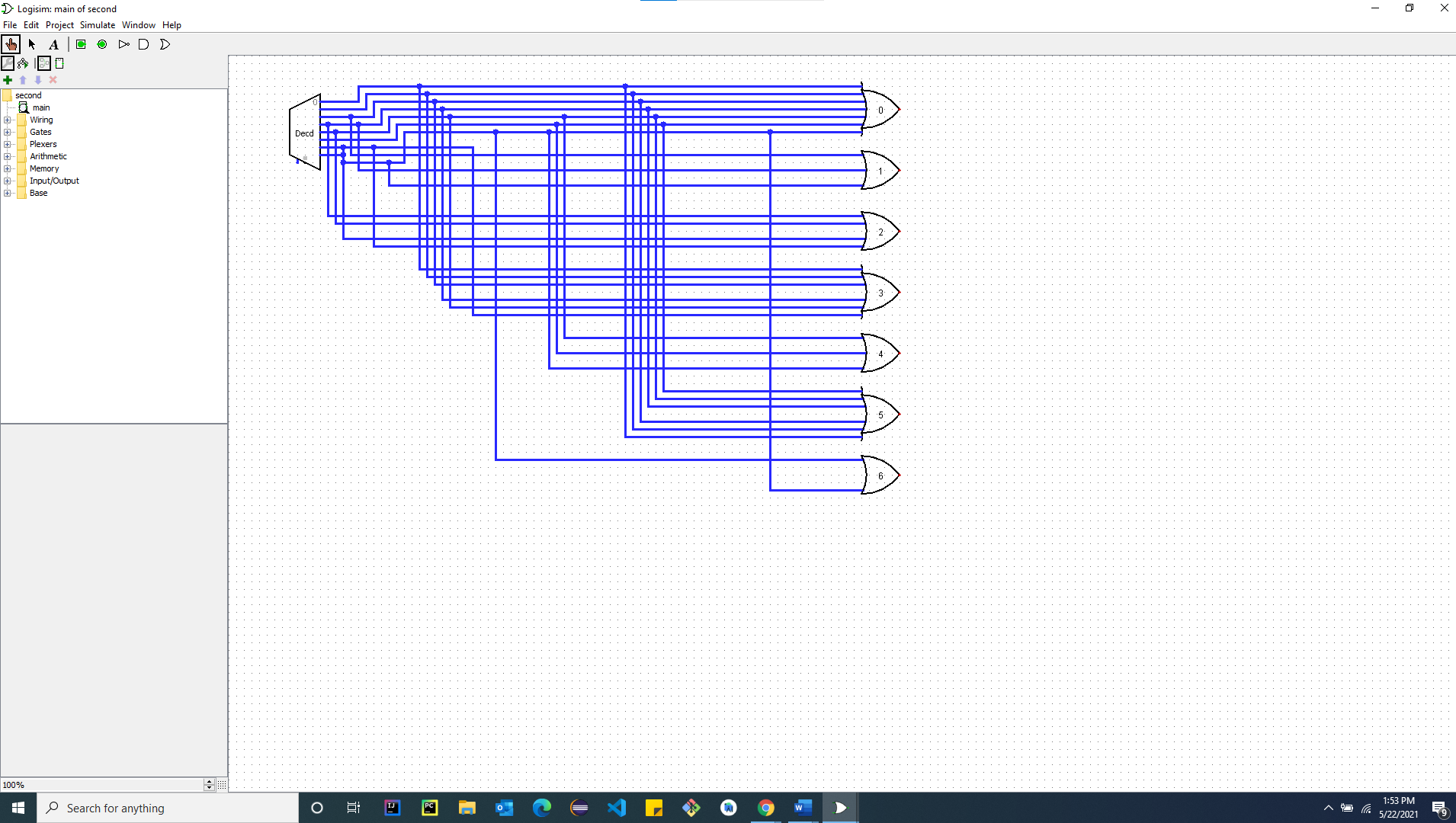
=219

X = 7

= 3(2\*(7)^2 + 1)

= 297

(c) Show in detail the final ROM layout, using *LogiSim*.



**Problem 3**

3.

Using MIPS assembly–language evaluate the following, *A097803*, polynomial:

*A* = 3(2*x*2 + 1)*, for x* = 0*,* 1*,* 2*,* 3*,* 4*,* 5*,* 6*,* 7

(a) The program should be simple modularized and well–documented

(b) Appropriate comments in the program are necessary

(c) Place the results, in the command line window (console). In the report include a clear screenshot of the DECIMAL results (command line window–console area) and the REGISTERS with the DECIMAL results (registers area)

(d) Indicate if the program runs successfully according to specifications and clearly state the results (DECIMAL numbers).

.data

pr1: .asciiz # formula A= 3(2\*x^2+1)

newline: .asciiz#print new line

pr2: .asciiz #The Program run successfully

re1: .asciiz

.text

.globl main

main:

li $v0, 4 #call for string for 1

la $a0, pr1 #String being stored

syscall

li $v0, 4 # call for new line instructions

la $a0, newline # string being stored

syscall

li $t0, 0 #storing 0 in $t0

li $t1, 7 #storing 7 in $t1

startingloop:

mul $t2, $t0, $t0 #square root

mul $t2, $t2, 2 #multiplying 2 into t2

add $t2, $t2 1 #adding 1 into t2

mul $t2, $t2, 3 #multiplying 3 into t2

li $v0, 4 #call for print string 1

la $a0, re1 #address of string

syscall

li $v0, 1 #print new line integer

move $a0, $t2 #move t2 to a0

syscall

add $t0, $t0, 1 # 1++ incrementation

ble $t0,$t1,startingloop #loop

#finaloop

li $v0, 4 #call for new line

la $a0, newline #String loaded

syscall

li $v0, 4 #printing node 2

la $a0, pr2 #string location in register

syscall

li $v0, 10 #exit

syscall

