## CS 312 Computer Architecture II

## 

## George Kim

**Garrett Marshall**

**Jason Nickell**

## 

**Lab 2. Decisions.**

Lethargic Larry does not like to make decisions, but every Saturday he has to figure out what to do. His choices are: go to the **Beach**, go to the **Movies** in town, or stay home and watch **TV**. If it is **not raining**, he will always walk to the **beach**. If his **car** is working and he has enough **money** to fill the tank he can use the **car** to go to the **movies**, however he must also have enough **money** to get into the **movies**. Otherwise, he will stay home and watch **TV**. He would like you to design a circuit that he can use to make the decision for him. He is willing to set the **switches** and wants one and only **one led** to light up telling him what to do.

Design and implement the circuit. Test the circuit to verify it works. Save the circuit to your lab account as Lab2. Create and fill in a Truth Table that will show the results.

The inputs for the table are **R (rain), C (car), G (gas money), M (movie money)**. The outputs are **B (beach), S (show time-movie), T (TV)**. For outputs, 1 means the activity to do and **0** means the activity will **not** be done. Note, **one** and only one activity is acceptable for each combination. This is why one led can be sufficient if you use enough switches.

You may use gates of your choosing to implement your design. Use **switches** that are **on** if the condition is true and **off** if the condition is not true. **Label** the switches on your working circuit.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **R** | **C** | **G** | **M** | **B** | **S** | **T** |
| **0** | **0** | **0** | **0** | **1** | **0** | **0** |
| **0** | **0** | **0** | **1** | **1** | **0** | **0** |
| **0** | **0** | **1** | **0** | **1** | **0** | **0** |
| **0** | **0** | **1** | **1** | **1** | **0** | **0** |
| **0** | **1** | **0** | **0** | **1** | **0** | **0** |
| **0** | **1** | **0** | **1** | **1** | **0** | **0** |
| **0** | **1** | **1** | **0** | **1** | **0** | **0** |
| **0** | **1** | **1** | **1** | **1** | **0** | **0** |
| **1** | **0** | **0** | **0** | **0** | **0** | **1** |
| **1** | **0** | **0** | **1** | **0** | **0** | **1** |
| **1** | **0** | **1** | **0** | **0** | **0** | **1** |
| **1** | **0** | **1** | **1** | **0** | **0** | **1** |
| **1** | **1** | **0** | **0** | **0** | **0** | **1** |
| **1** | **1** | **0** | **1** | **0** | **0** | **1** |
| **1** | **1** | **1** | **0** | **0** | **0** | **1** |
| **1** | **1** | **1** | **1** | **0** | **1** | **0** |

Use the **sum of products** (SOP) method to write the equations for each of the outputs as they are defined by the truth table. It means that you record in SOP combinations of RCGM where a respective output B,S or T is 1.

Then you use **Karnaugh** maps to **simplify** the equations. If you came up with simpler equations than the sum of products then compare your results with what you get from the reduced equations from the Karnaugh maps.

RC

B=C+

~~=(C+)~~

**B=**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 00 | 01 | 11 | 10 |
| 00 | 1 | 1 |  |  |
| 01 | 1 | 1 |  |  |
| 11 | 1 | 1 |  |  |
| 10 | 1 | 1 |  |  |

GM

RC

**S=RCGM**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 00 | 01 | 11 | 10 |
| 00 |  |  |  |  |
| 01 |  |  |  |  |
| 11 |  |  | 1 |  |
| 10 |  |  |  |  |

GM

RC

SEE PICTURE BELOW FOR WORK ON HOW WE GOT THE FOLLOWING EQUATION….

**T = R\***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 00 | 01 | 11 | 10 |
| 00 |  |  | 1 | 1 |
| 01 |  |  | 1 | 1 |
| 11 |  |  |  | 1 |
| 10 |  |  | 1 | 1 |

GM

Picture of work for T equation….

Text, letter

Description automatically generated

**Use a separate page to put a screenshot of the circuit that you designed and implemented. Note, the circuit you design may not be the same as the equations since you are doing one circuit to cover all conditions. Write the equations for the outputs as defined by your circuit design.**

Here is a picture of our original circuit made from the equations mentioned on page 2.

Diagram

Description automatically generated

But after some playing around, we found another circuit that worked with one less gate. The only difference is that the T logic is now T =

