Energy theft

Logic design project

Section: 2C6	Group name: Logisim
Student names:	ID:
Mjd Alamri	443007585
[redacted]	

- Table of Contents

1.Introduction, and Project Aim

Have you ever got your bill so high when you don't used it that much? You might got robbed. In this project we will create system that detects electricity thieves.

Our main goal is help you to find out if someone is using your energy.

Before we explain our project let's start with these definitions:

iоT: The Internet of Things describes the network of physical objects—"things"—that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet.

Arduino: is an open-source electronics platform based on easy-to-use hardware and software which can read inputs and turn them into an output that either sends a message or lights an LCD. It is used to make it easy to deal with electronics.

2.The Problem Solution

We will design a circuit will cheek for any change in your Electric system and will decide if it is a theft or a normal change then tells you about it.

2.1. Inputs and Outputs

We have 8 inputs (m4m3m2m1 AND a4a3a2a1) and 4 outputs (F_{energy_change}, F_{change_detect}, F_{Hbc}, F_{Lbc}).

2.2 The Truth Tables

2.2.1: F_{energy_change} truth table

m4	m3	m2	m1	a4	a3	a2	a1	$f_{energychange}$
0	0	0	0	0	0	0	0	1
-	-	-	-	-	-	-	-	0
0	0	0	1	0	0	0	1	1
-	-	-	-	-	-	-	-	0
0	0	1	0	0	0	1	0	1
-	-	-	-	-	-	-	-	0
0	0	1	1	0	0	1	1	1
-	-	-	-	-	-	-	-	0
0	1	0	0	0	1	0	0	1
-	-	-	-	-	-	-	-	0
0	1	0	1	0	1	0	1	1
-	-	-	-	-	-	-	-	0
0	1	1	0	0	1	1	0	1
-	-	-	-	-	-	-	-	0
0	1	1	1	0	1	1	1	1
-	-	-	-	-	-	-	-	0
1	0	0	0	1	0	0	0	1
-	-	-	-	-	-	-	-	0
1	0	0	1	1	0	0	1	1
-	-	-	-	-	-	-	-	0
1	0	1	0	1	0	1	0	1
-	-	-	-	-	-	-	-	0
1	0	1	1	1	0	1	1	1
-	-	-	-	-	-	-	-	0
1	1	0	0	1	1	0	0	1
-	-	-	-	-	-	-	-	0
1	1	0	1	1	1	0	1	1
-	-	-	-	-	-	-	-	0
1	1	1	0	1	1	1	0	1
-	-	-	-	-	-	-	-	0
1	1	1	1	1	1	1	1	1

The truth table has 8input because it reads 4input from the main energy port and 4 from actual energy port, the output (F_{energy_change}) give the result of 1 if all are equal, otherwise, 0.

but we can't insert all the possibilities so we had to make it shorter and put all the different output in one line.

2.2.2: F_{hbc} truth table

This truth table is for $F_{hbc,}$ it checks the two leftmost bits of our reading if they are equal, the results will be 1; otherwise, 0.

m4	m3	m4	m3	f_{hbc}
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

If the result = 1 we need to check Low bits (next page)

2.2.3: F_{lbc} truth table

The F_{Lbc} is 0 if the change was only in one bit and 1 if the two-bits change.

m2	m1	a2	a1	f_{lbc}
0	0	0	0	-
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	-
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	-
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	-

we used this table f_{lbc} to check low bits when high bits from f_{hbc} are equal, if the low bits from f_{lbc} are equal too, that means $f_{lbc}=f_{hbc}$ in this case we use the symbol “- “(the circuit is closed).

2.2.4: F_{energy_detect} truth table

this is Energy theft detection truth table indicates 1 as output only if (F_{Hbc} = 1and (F_{Lbc} =0) which means it's safe.

F_{hbc}	f_{lbc}	F_{energy_detect}
0	0	0
0	1	0
1	0	1
1	1	0

If there is no theft the LCD screen will display green light

2.3 k-map:

2.3.1: K-map for f_{lbc}:

We use the previous truth table to drew this k-map

0	1	3 1	2
4	5	7	6 1
12 1	13	15	14
8	9 1	11	10

2.4 Boolean Functions:

We found these functions using the previse truth table:

F_{energy_change}:

(m4 XOR a4)’ (m3 XOR a3)’ (m2 XOR a2) (m1 XOR a1)’

F_{hbc}:

m4’m3’a4’a3’+m4’m3a4’a3+m4m3’a4a3’+m4m3a4a3

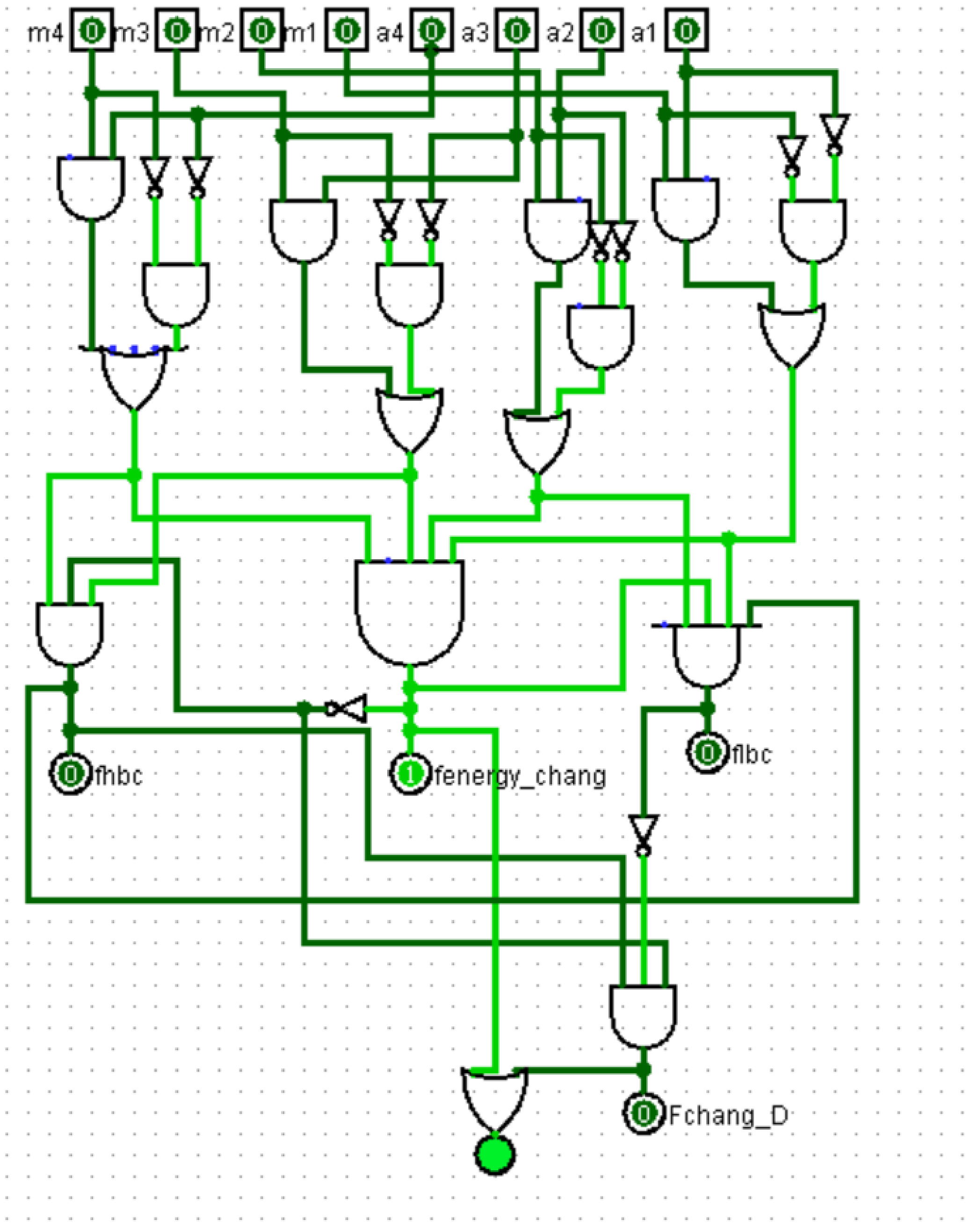
F_{lbc}:

m2’m1’a2a1+m2’m1a2’a1+m2m1’a2’a1+m2m1a2’a1’

F_{energy_detect}:

$(F_{hbc})(F_{lbc}')\ (F_{energy\_change'})$

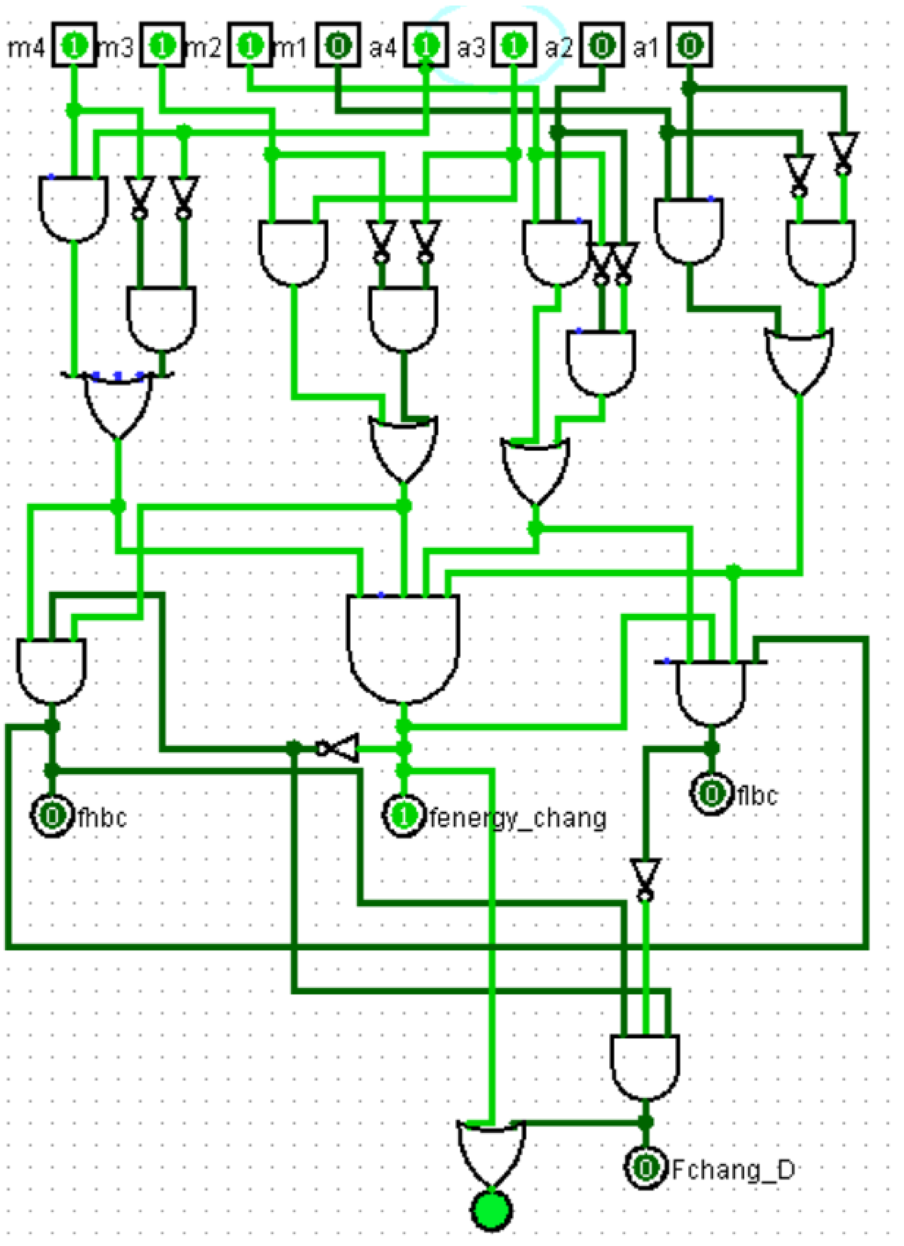
2.5 The Logic Circuits Design:



We created this circuit based on all the privies functions

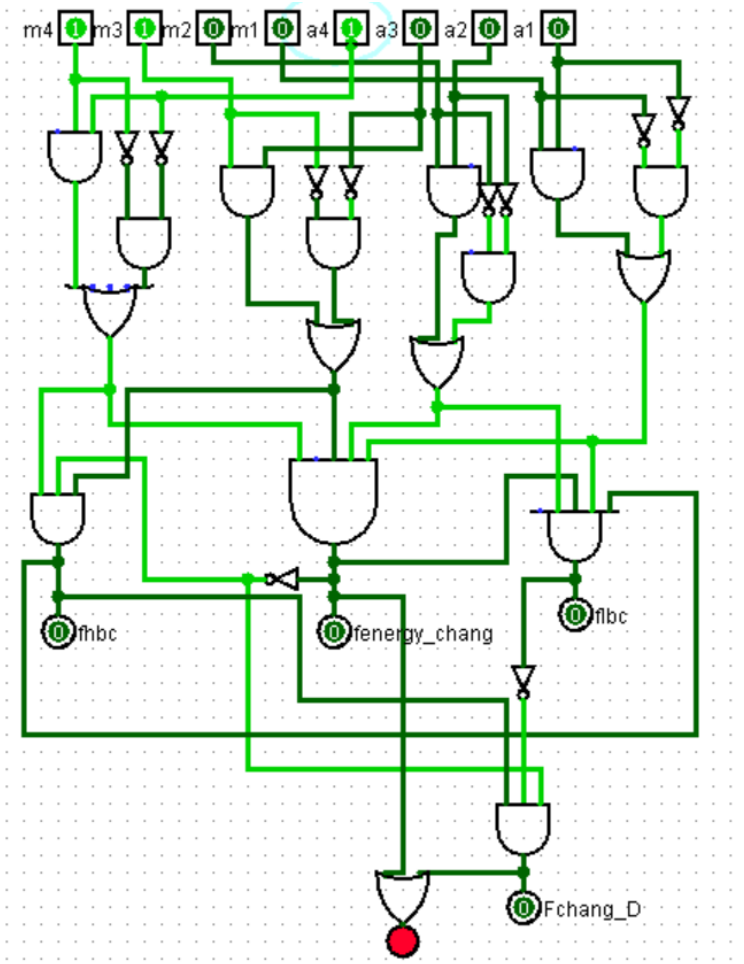
2.6 Experiments using Logisim with three cases: no theft, theft, and a special case:

2.6.1: No theft



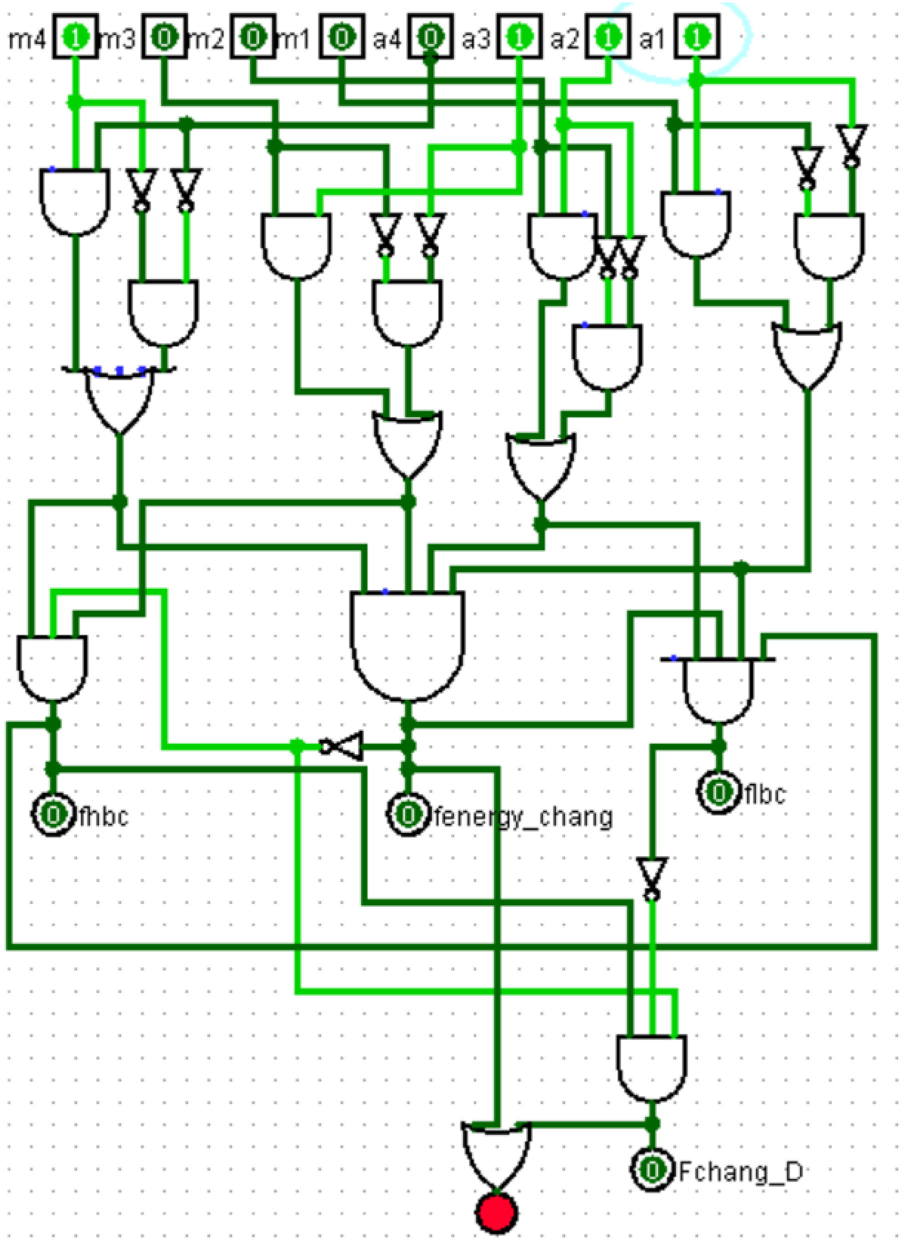
This case considers as “No theft” because of the change was in the low bit

2.6.2: Theft



This case considers as "Theft" because of the change in the high bit

2.6.3: Special case



This case is a special one because even though the bill shows one value difference this change in all bits will flag theft detection.