2-bit Adder using Decoders

Moises Jahir González Márquez, Marco Baez González, Manuel Fernández Mercado, Eduardo Estrada González
Meritorious Autonomous University of Puebla, Faculty of Computer Science
February 2, 2025

In this practice, a full 2 bit adder was implemented by using the IC 3-8 decoder 74138, the IC 7408 AND gate, the IC 7404 NOT gate and a 7-segment display along with the IC 74147 BCD decoder as a better way to watch the results . The purpose of this practice was to understand the operation of a decoder as well as his hability of recreate/emulate a boolean function along with other logic gates.

***Index Terms*— adder, decoders, logic gates.**

I. INTRODUCTION

A decoder is a combinational circuit that converts a coded input into a specific output. Essentially its main function is to activate a single output according to the value of the combination of inputs received. The most common decoder is the binary decoder, it takes a binary input and outputs his decimal representation. Let's look at the next table to better understand how it works, in this example we are working with a 3 to 8 decoder.

Input (AB)	Output
00	Y_o
01	Y_1
10	Y_2
11	Y_3

Since the input is BCD, we could say that given n inputs we get 2^n possible outputs, these outputs can have positive (1-HIGH) or negative (0-LOW) logic depending on the chip model, IC 74138 for example, works with negative logic.

II. PRINCIPLES OF OPERATION

We can take advantage of the decoder's ability to map combinational inputs into specific outputs to build a full adder. As we know, a full adder takes 3 inputs: A, B as addends and C_{in} as the carry in. This operation gives us two variables as outputs: S (representing the sum) and C_{out} (representing the carry out), S is represented by the $(A \otimes B) \otimes C$ (XOR), while carry out is represented by $(A \cdot B) + [(A \otimes B) \cdot C_{in}]$. Let's build the truth table of a full adder with the information provided.

A	B	C	S	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Looking at the table we could represent S and C_{in} with the minterms by obtaining the following expressions $S(Y_1, Y_2, Y_4, Y_7)$, $C_{out}(Y_3, Y_5, Y_6, Y_7)$.

Since we are working with the IC 74138 which follows negative logic (active with LOW) we shall connect each function minterms into a NAND gate to achieve the correct behavior. This results in a 1-bit adder which has the following diagram:

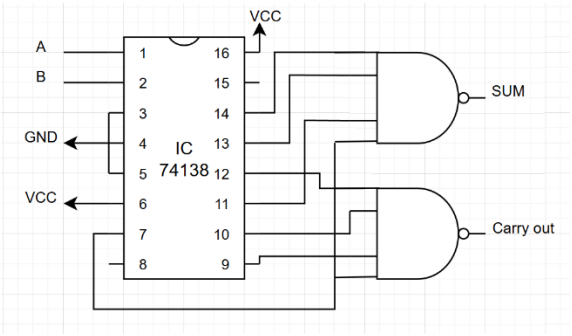

Fig. 1 74138 iC 1-bit full adder diagram

However, we want to make a 2-bit adder, so we need to connect two circuit blocks to do so. The next diagram explains the circuit. The carry out 1 is connected to the C_{in2} (CIN2). The result is composed of SUM1 (less significant bit), SUM2 (most significant bit) and COUT2 (overflow bit)

Laboratory V: 2-bit Adder using Decoders

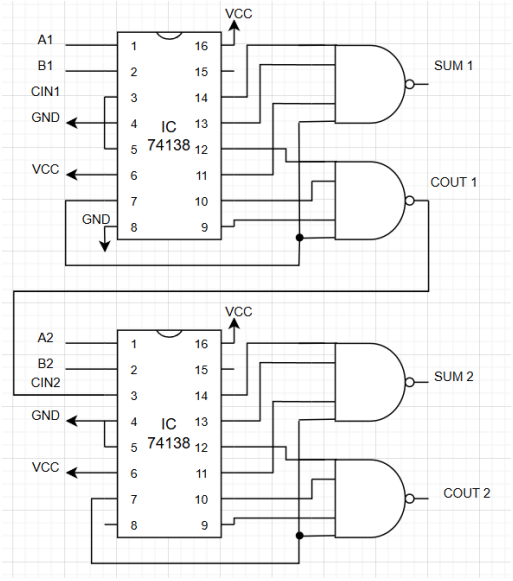

Fig. 2 74138 iC 2-bit full adder diagram

In the physical implementation we use use AND gates (7408) as well as NOT gates to emulate a NAND gate. The simulation will then be contrasted with the practice in the following two images, in both cases we will test the sum 2+3.

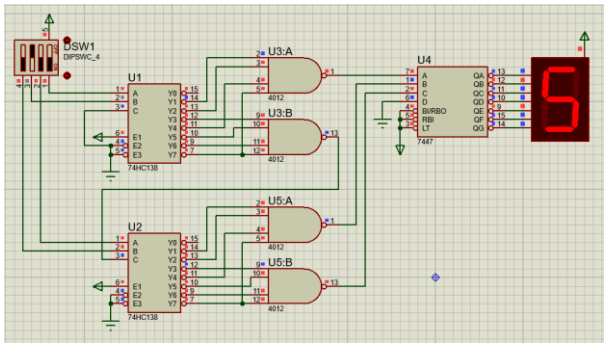

Fig. 4 Proteus simulation test

III. RESULTS AND CONCLUSIONS

The Proteus diagram of the circuit is shown in the next figure. In this case a dip switch is used to control the addends $(A_1, A_2; B_1, B_2,)$ and a 7447 decoder as well as a 7-segment common anode display to the sum.

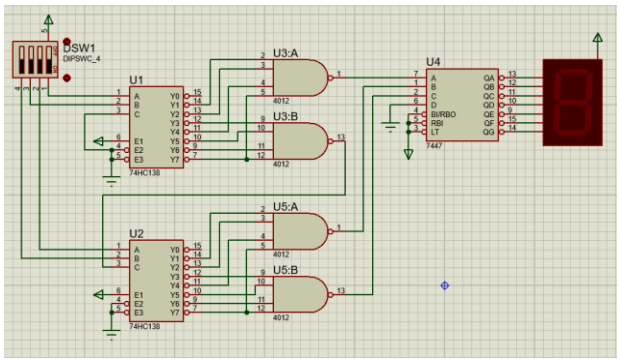

Fig. 3 Proteus simulation

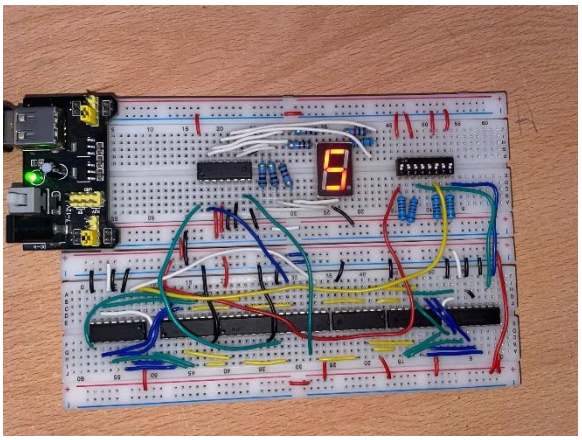

Fig. 5 Physical implementation test

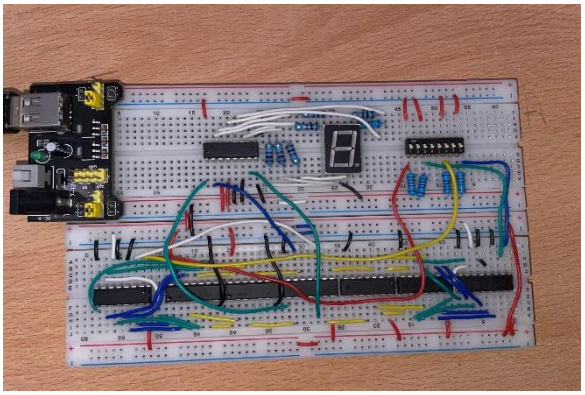

Fig. 6 Physical implementation