Module 5: Laboratory 2: Arithmetic Circuits and Comparators

Objectives

The objectives of this experiment are to:

- Examine how addition and subtraction are performed using logic gates.
- Examine how binary numbers are compared using logic gates.

Introduction

Binary arithmetic is implemented in hardware by direct application of the basic logic gates. In fact, the AND gate performs a simple form of binary multiplication and the OR gate supplies the basic addition property. In this experiment we examine some of the common arithmetic operations performed on groups of bits.

Procedure

1. Load the circuit E10-1.MS7, shown in Figure 10.1.

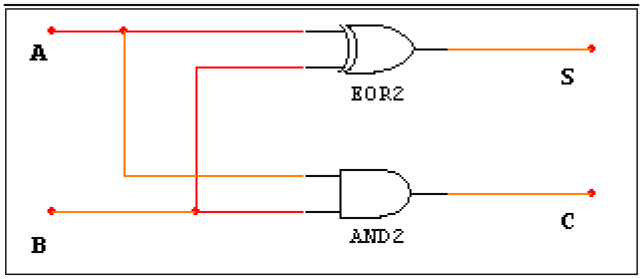

Figure 10.1: Half adder

This circuit is the *half adder*, and is capable of adding two bits together, generating a sum (S) and carry (C). Add the components needed to verify the operation of the half adder, whose truth table is shown in Table 10.1.

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Table 10.1: Truth table for the half adder

2. Circuit E10-2.MS7, shown in Figure 10.2, performs the same logic function as the half adder. This part is in the Miscellaneous Digital parts bin.

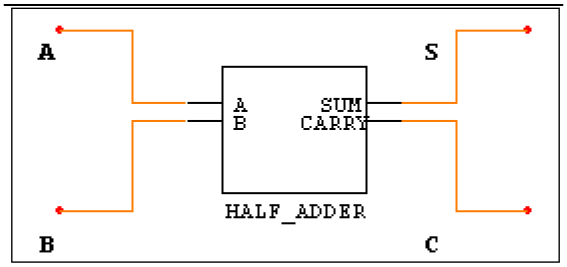

Figure 10.2: Simplified half adder circuit

Test the circuit to verify its operation.

3. A *full adder* adds three bits together. The A and B inputs, as well as a Carry input, are added. Figure 10.3 shows the diagram of the full adder. Load circuit E10-3.MS7 to examine it.

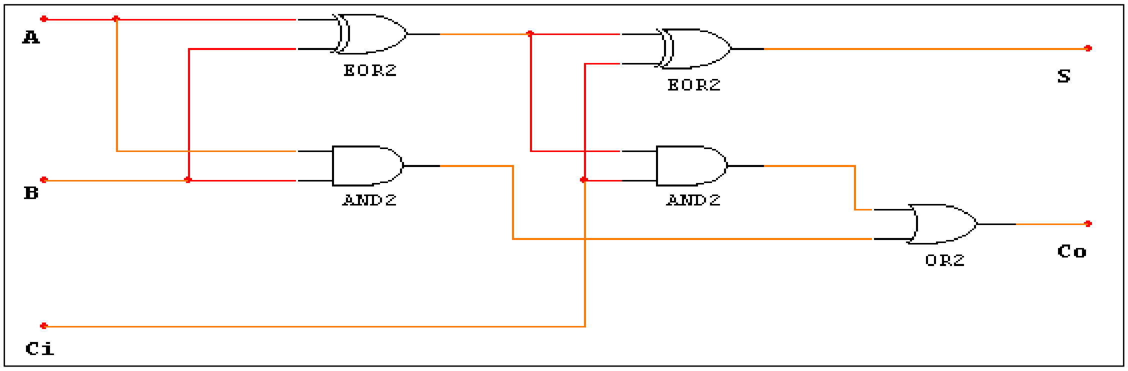

Figure 10.3: Full adder

Use the Logic Converter to test the full adder, whose truth table is shown in Table 10.2.

A	B	Ci	S	Co
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

Table 10.2: Truth table for the full adder

4. The Miscellaneous Digital parts bin also contains a simplified full adder circuit, which is shown in Figure 10.4. Open E10-4.MS7 to examine it.

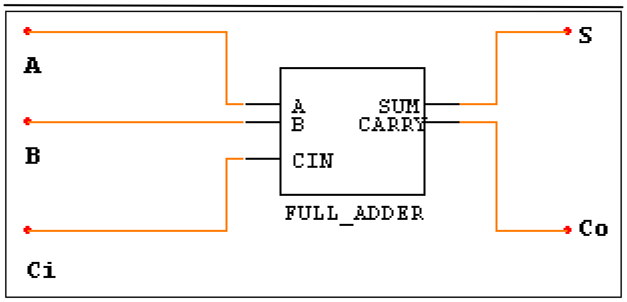

Figure 10.4: Simplified full adder circuit

5. A *parallel* adder adds two or more bits from two input numbers at the same time. Open circuit E10-5.MS7 (shown in Figure 10.5) to examine a 2-bit parallel adder.

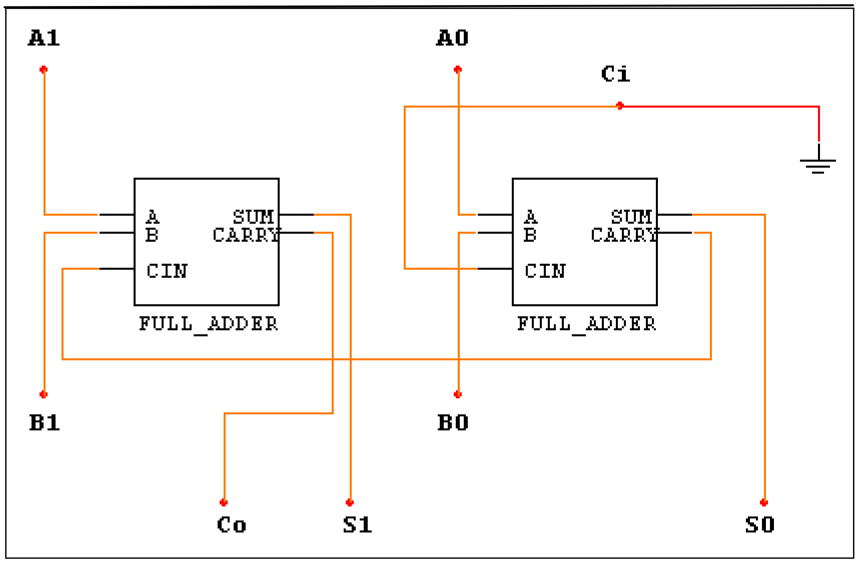

Figure 10.5: 2-bit parallel adder

The Ci input is grounded to apply a logic zero level, guaranteeing that the initial sum bit S0 is correct. Add the components required to test the 2-bit adder.

6. Make a 4-bit parallel adder by cascading two copies of the 2-bit adder. Connect the Co output of the first 2-bit stage to the Ci input of the second 2-bit stage. Note that the sum may now require five bits.

7. Now modify the 4-bit adder into a 4-bit *subtractor*. Subtraction is performed using 2's-complement arithmetic. For example, when subtracting 0011, first complement all bits (1100)

and then add one (1100 plus 0001 equals 1101). So, 1101 is the 2's complement of 0011. Our interpretation of both numbers is that 0011 equals positive 3 and 1101 equals negative 3. To subtract, we actually *add* 1101 to the other number. So, in general, A – B becomes A + (–B) and we can perform subtraction using our adder hardware, with a few additional logic gates. Modify the 4-bit adder by inserting inverters on all the B inputs and by tying the Ci input high. This will simulate the 2's complement of B. Place 0111 on the A inputs and 0011 on the B inputs. The S outputs should equal 0100.

8. It is often necessary to compare two binary numbers to determine if they are equal. Circuit E10-6.MS7 (shown in Figure 10.6) uses exclusive NOR gates to perform the individual comparisons (LSB to LSB, etc.).

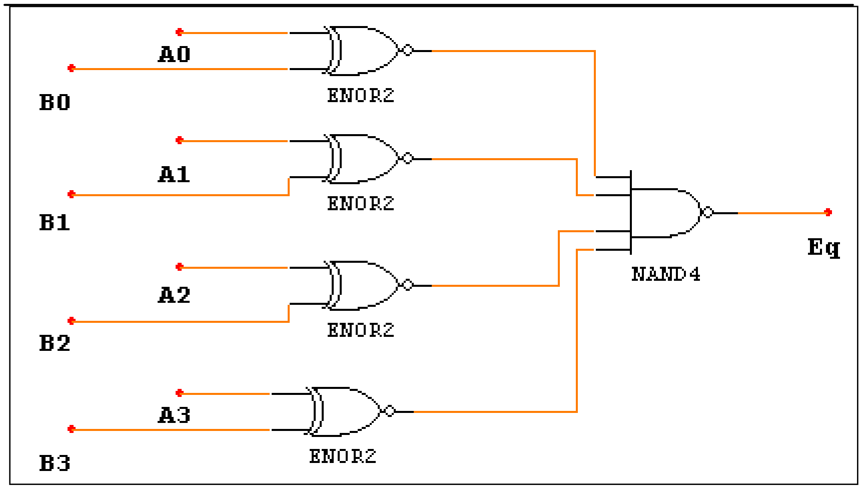

Figure 10.6: 4-bit equality comparator

The Eq output goes low when the 4-bit inputs numbers are equal. Verify the operation of the equality comparator.

9. A *magnitude* comparator is able to determine if two numbers are smaller than, equal to, or larger than one another. A simple 2-bit less-than comparator is shown in Figure 10.7. Open E10-7.MS7 to examine the circuit.

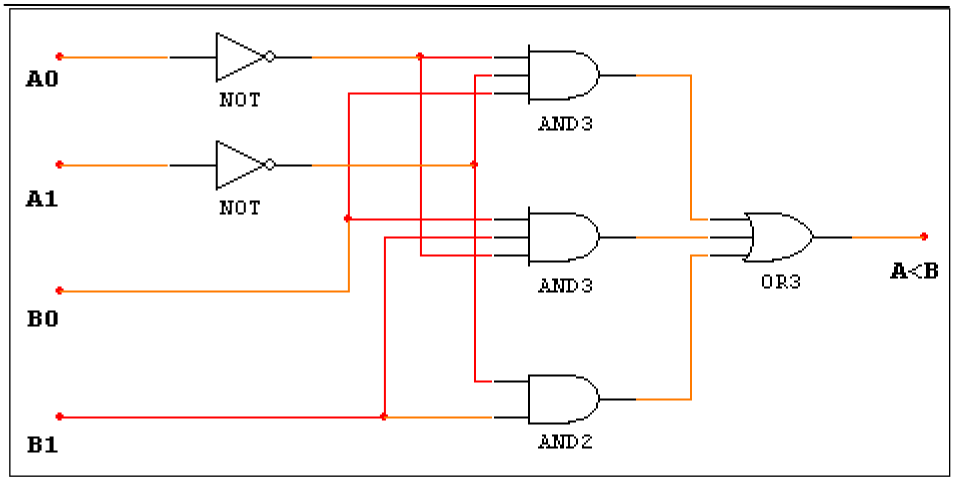

Figure 10.7: 2-bit less-than comparator

A 4-input Karnaugh map was used to design the less-than comparator, with ones placed into each position where A is less than B. Thus, the AB output goes high whenever A1A0 is less than B1B0.

Add the required components needed to test the less-than comparator. Fill in the Karnaugh map as you perform the test.

	B1B0	B1B0	B1B0	B1B0
A1A0	00	01	11	10
00				
01				
11				
10				

Figure 10.8: Karnaugh map for less-than comparator

10. Use two copies of circuit E10-7.MS7, plus some additional logic, to make a 4-bit less-than comparator. The additional logic is required because the upper two bits of each number may be equal, forcing the lower 2-bit stage to make the final decision. Thus, the 2-bit less-than comparator must be modified slightly to allow for cascading.

Discussion:

While reviewing your data and results, provide detailed answers to each of the following:

1. What is the difference between a half adder and a full adder?
2. Is it necessary to use a full adder as the first stage in a parallel adder?
3. Why is the Ci input tied high in the 4-bit subtractor?

4. How are exclusive NOR gates used to perform comparisons?