Experiment 3 : Full Adder and Subtractor

Digital Systems and Microcontrollers | Monsoon 24

Objective : To design , assemble and test binary adders and subtractors. In the previous experiment, you learned the working of the basic logic gates. In this experiment we are going to use those logic gates to design a half adder, half subtractor , full adder and full subtractor.

Experiment:

Part A : Binary Half Adder

The first step to achieve a binary full adder is to make a binary Half Adder, which adds two binary inputs A and B to give a sum S1 and a carry C1 according to the following Boolean expressions for the outputs S1 and C1:

$$S1 = A' \cdot B + A \cdot B' = A \oplus B \text{ and } C1 = A \cdot B.$$

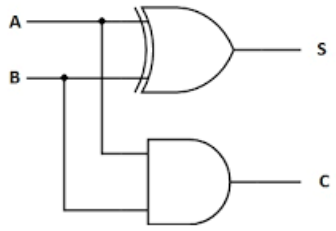

Fig. Half Adder

Set up the circuit of a Half Adder using an XOR gate and an AND gate. Apply the inputs A and B from two input switches and observe the outputs S1 and C1 on two LED displays for all combinations of the inputs. Tabulate these values and verify the operation of the Half Adder.

Part B : Binary Full Adder

A binary Full Adder adds two bits A and B along with a carry in C to generate SUM and CARRY bits as output. So the binary half adder made in part **A** is used with another Half Adder to generate the final SUM by adding the third binary input C to the S1 bit generated by the first Half Adder:

$$SUM = S1 \oplus C.$$

The carry bit generated by this Half Adder is given by

$$C2 = S1 \bullet C.$$

Write down the complete truth table of a Full Adder, including columns for the intermediate outputs S1, C1 and C2. Find out the logic for generating the final CARRY output from C1 and C2. As XOR and AND gates are going to be used for the Half Adders, try to obtain a logic for CARRY using the same type of gates, so that the complete realization of the Full Adder is possible without necessitating a third IC.

1. Set up another Half Adder using another XOR and another AND gate out of the same ICs used in part **A**, and connect the C input and the S1 output generated by the first Half Adder in part **A** as its inputs to generate the final SUM output and the C2 output.

2. Generate the final CARRY output from the intermediate carry outputs C1 and C2, using the unused gates in the XOR and AND ICs deployed so far (instead of the OR gate as in figure).

3. Verify the truth table experimentally by applying the inputs A, B and C through three input switches and displaying the S1, C1, C2, SUM and CARRY outputs.

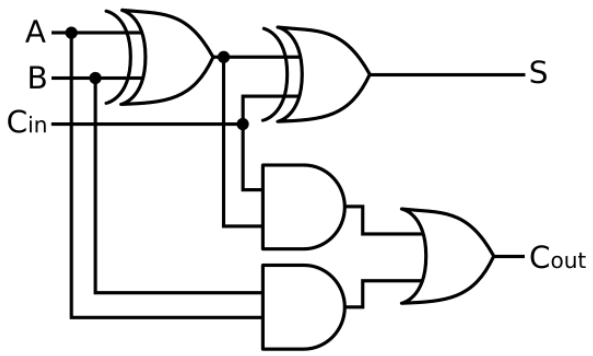

Fig. Full Adder

Part C : Binary Half Subtractor

The first step to achieve a binary full subtractor is to make a binary Half Subtractor, which subtracts two binary inputs A and B to give a DIFFERENCE D1 and a BORROW B1 according to the following Boolean expressions for the outputs D1 and B1:

$$D1 = A' \cdot B + A \cdot B' = A \oplus B \text{ and } B1 = A' \cdot B.$$

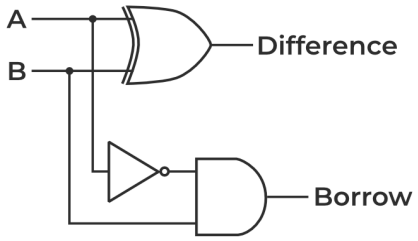

Fig. Half Subtractor

Set up the circuit of a Half Subttractor using an XOR , NOT gate and an AND gate. As a NOT gate is used in a Half - Subtractor , try to implement it without the NOT IC using the other logic gate ICs used in the Half - Subtractor. Apply the inputs A and B from two input switches and observe the outputs D1 and B1 on two LED displays for all combinations of the inputs. Tabulate these values and verify the operation of the Half Subtractor.

Part D : Binary Full Subtractor

A binary Full Subtractor Subtracts two bits A and B along with a BORROW to generate DIFFERENCE and BORROW bits as output (two's complement form of output) . So the binary half subtractor made in part **C** is used with another Half Subtractor is then used to generate the final DIFFERENCE by Subtracting the third binary input C to the D1 bit generated by the first Half Subtractor:

$$\text{DIFFERENCE} = D1 \oplus C.$$

The BORROW bit generated by this Half Subtractor is given by

$$D2 = D1' \cdot C.$$

Write down the complete truth table of a Full Subtractor, including columns for the intermediate outputs D1, B1 and B2. Find out the logic for generating the final BORROW output from B1 and B2.

1. Set up another Half Subtractor using another XOR , NOT and another AND gate out of the same ICs used in part **C**, and connect the C input and the D1 output generated by the first Half Subtractor in part **C** as its inputs to generate the final DIFFERENCE output and the B2 output.

2. Generate the final BORROW output from the intermediate BORROW outputs B1 and B2, using the unused gates in the XOR and AND ICs deployed so far (instead of the OR gate as in figure).

3. Verify the truth table experimentally by applying the inputs A, B and C through three input switches and displaying the D1, B1, B2, DIFFERENCE and BORROW outputs.

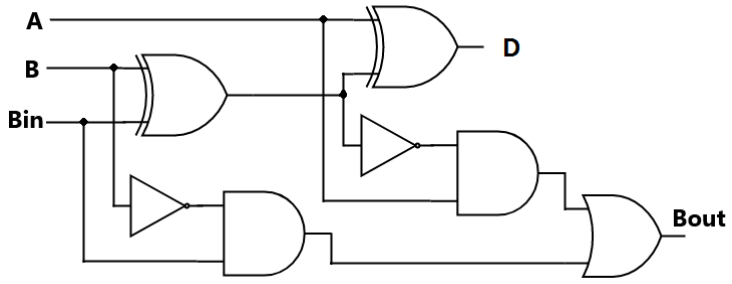

Fig. Full Subtractor