

DIGITAL ELECTRONICS AND COMPUTER ORGANIZATION LABORATORY

Dr. Anjad Badran

Experiment No. 2 - Comparators, Adders and Subtractors

Islam Jihad Joma 1191375

T. Enas

1-3-2021

**Abstract**

In this experiment we have to build half subtracter and adder and full adder and subtracter and comparators with different number of bits for all circuits, we have to use both basic gates and some designed blocks, we have to simulate them via Proteus and find the truth table for each circuit.

**Table of content:**

We need some pieces but as we are in online learning so all we need is proteus and his software of different pieces.

**Introduction (Theory)**

In Comparator Circuit At least two numbers are required to perform any comparison. The simplest form of the comparator has two inputs. If the two inputs are called A and B, there are three possible outputs: A>B, A=B, an d A<B, we use it to create 4 bit comparator which is most used.

Digital computers perform a variety of information processing tasks. Among the functions encountered are the various arithmetic operations. The most basic arithmetic operation is the addition of two binary digits. Combinational circuit that performs the addition of two bits is called a half adder. One that performs the addition of three bits (two significant bits and previous carry) is a full adder. The names if the circuits stem from the fact that two half adders can be employed to implement a full adder.

In Half- and Full-Subtractor Circuits, binary subtraction is usually performed by using 2’s complement. Two steps are required to obtain 2’s complement. First, the subtrahend is inverted to 1’s complement, i.e. a “1” to a “0” and a “0” to a “1”. Secondly, a “1” is added to the least significant bit of the subtrahend in 1’s complement. A half-subtractor performs the task if subtraction 1-bit at a time regardless of whether the minuend is greater or less than the subtrahend. “Borrow” from previous subtraction is not taken into consideration. Full subtractor it’s made as same as half subtractor but with some difference to get the carry from the previous bit.

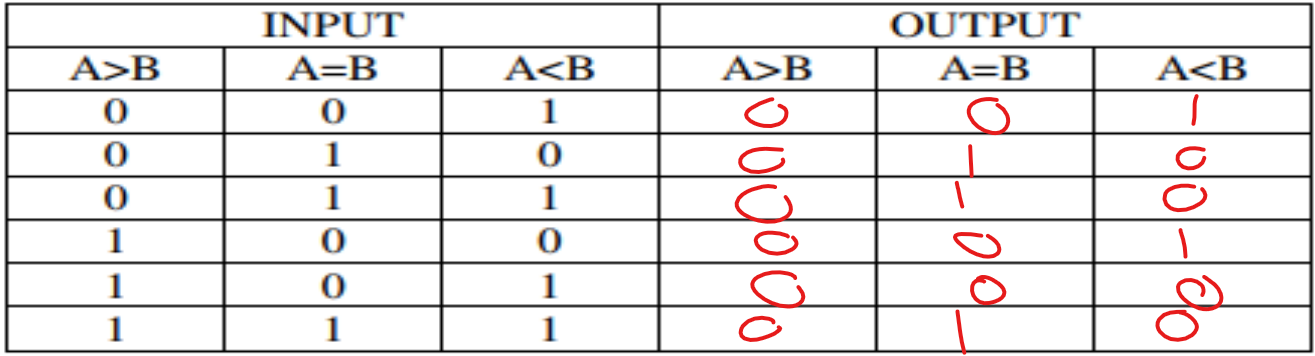
**Procedure (Discussion & Results)**

First, I made 1 bit comparator using basic gates but in the inverse form and the truth table was as following:

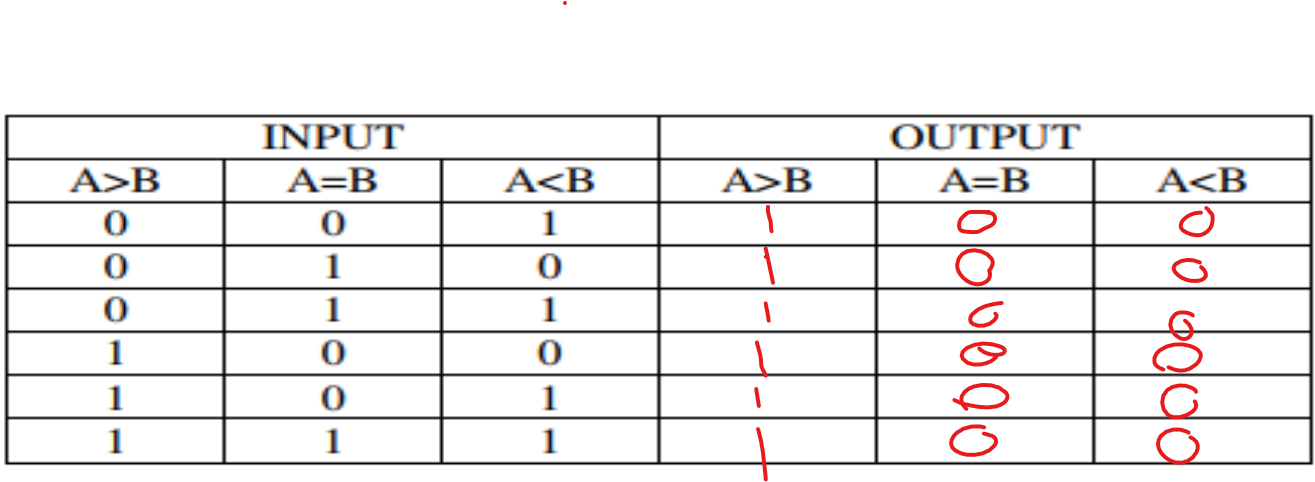
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | inputs |  | outputs |  |
| A | B | A>B | A=B | A<B |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 |

And by simulating it by protues we got those results which are true for one bit comparoter, I used basic gates in the all experimnts file diagram and give it input bit and output bit and connected them with wires .

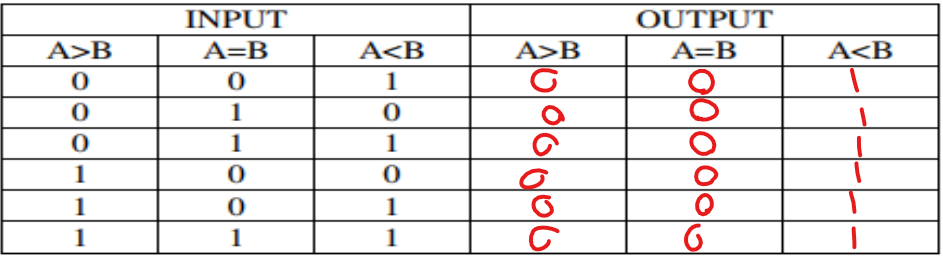
Then we made it bigger by make it a 4 bit comparator using 74LS85 block, and we got this table from simulating:



Those results when A=B



And rhose when A>B (they might be a problem but I checked it well and I didn’t find and wrong thing)



And those when B>A (they might be a problem but I checked it well and I didn’t find and wrong thing).