

**Faculty of Engineering and Technology**

**Department of Electrical and Computer Engineering**

**ENCS 211**

**Digital Electronics and Computer Organization Lab**

**Experiment 2**

**Comparators, Adders and Subtractors**

**Student’s name : Sondos Sarraj**

**Student’s id’s : 1140275**

**Abstrac :**

**In this experiment we learned how to construct a half / full \_ adders , half / full \_ subtractors and comparators .**

**Theories :**

***\* Comparator* : a comparator has 3 outputs (a>b ,a=b , a<b). It works**

**by comparing the most significant bits with each other (binary numbers),depends on the value of each number (a,b) . If it’s not the same, it gives (1) to (a<b) or (a>b) but if it’s the same it moves on to the next bit and compare them as it did with the first bit and so on .**

**When it reaches the last bit, if (a) and (b) have the same value , it gives (1) to (a=b).**

**\* *half-adder* : half-adder is used to add just two bits (binary number), therefor it has 2 input (a,b) and 2 outputs (s = sum ,c= carry ), where s = a(XOR) b and c = a (AND) b .**

**\**full-adder* : full-adder is made of two half-adders , it has three inputs**

**(a,b, C\_in) and two output (S,C\_out). We use full-adder to add three bits, the sum (S) is taken from XOR gate and the carry ( c ) by OR gate.**

**\* *half-subtractor* : the half-subtractor has two inputs (a,b) and two outputs  (DF,BW) .To get the DF we use XOR gate and the BW we use AND gate.**

**\* *full-subtractor* : the full-subtractor is a combination of XOR,OR,NOT,**

**AND Gates. The full subtractor should have three inputs and two outputs.Two half- subtractor together gives a full subtractor .**

**Procedure :**

1. ***Comparator :***

|  |  |  |
| --- | --- | --- |
| **Inputs** |  | **Outputs** |
| **A** | **B** | **F1(a>b)** | **F2(a=b)** | **F3(a<b)** |
| **0** | **0** | **A=B** | **0** | **1** | **0** |
| **0** | **1** | **A<B** | **0** | **0** | **1** |
| **1** | **0** | **A>B** | **1** | **0** | **0** |
| **1** | **1** | **A=B** | **0** | **1** | **0** |

1-bit Comparator Truth Table

1. ***half-adder* :**

|  |  |  |
| --- | --- | --- |
| **Inputs** |  | **Outputs** |
| **A** | **B** | **F1(sum)** | **F2(carry)** |
| **0** | **0** | **0** | **0** |
| **0** | **1** | **1** | **0** |
| **1** | **0** | **1** | **0** |
| **1** | **1** | **0** | **1** |

Half-Adder Truth Table

***3- full-adder :***

|  |  |  |
| --- | --- | --- |
| **Inputs** |  | **Outputs** |
| **C** | **A** | **B** | **F1(sum)** | **F2(carry)** |
| **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** |

**Full-adder Truth Table**

1. ***Half/full-subtractor:***

|  |  |  |
| --- | --- | --- |
| **Inputs** |  |  **Outputs** |
| **C** | **A** | **B** |  | **F1(BW1)** | **F2(DF1)** | **F3(BW2)** | **F4(DF2)** |
| **0** | **0** | **0** |  | **0** | **0** | **0** | **0** |
| **0** | **0** | **1** |  | **1** | **1** | **1** | **1** |
| **0** | **1** | **0** |  | **0** | **1** | **0** | **1** |
| **0** | **1** | **1** |  | **0** | **0** | **0** | **0** |
| **1** | **0** | **0** |  | **0** | **0** | **1** | **1** |
| **1** | **0** | **1** |  | **1** | **1** | **1** | **0** |
| **1** | **1** | **0** |  | **0** | **1** | **0** | **0** |
| **1** | **1** | **1** |  | **0** | **0** | **1** | **1** |

**Half/full-subtractor Truth Table**

**Conclusion:**

**After implementing half/full adders, and half/full subtractor,**

**it is shown that these constructors are very useful in calculators.**

**During implementing**

**Some reselts weren’t like the truth table values ; because the reselt is based on the values we entered to the circuit .**