JAYPEE UNIVERSITY OF ENGINEERING & TECHNOLOGY, GUNA
DEPARMENT OF COMPUTER SCIENCE & ENGINEERING
Course: Computer Organization & Architecture Lab
Course Code: 14B17CI673
B. Tech. (CSE VI Sem.)

Experiment # 2

Aim: Design of basic binary adders and subtractors.

Basic binary adders/subtractors are the type of combinational digital circuits (output depends only on present inputs not the previous output and do not require clock signal) that combine binary values to obtain sum/difference. These are the most basic operators of the digital circuits. These are classified according to their ability to accept and combine the number of input bits as followings:-

- **Half Adder/Subtractor**
  - Performs the addition/subtraction of two bits and gives the two outputs sum/difference and carry/borrow. There may be a quarter adder/subtractor that combines two bits and produces only the single (sum/difference) output without carry/borrow.
  - The inputs of half adder/subtractor are known as augend/minuend and addend/subtrahend.
  - Major disadvantage of the half adder/subtractor circuit is that there is no provision for a "carry/borrow-in" from the previous circuit.
- **Full Adder/Subtractor**
  - Performs the addition/subtraction of three bits (two significant bits to be added and one carry/borrow bit from previous circuit) and gives the two outputs sum/difference and carry/borrow.
  - Two half adders/subtractors and one OR gate are needed to implement one full adder/subtractor.

Half Adder

Logic Diagram

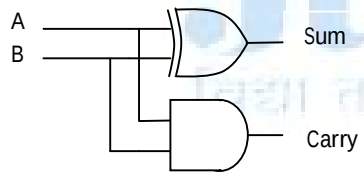

Figure 1: Half Adder

Boolean Expressions

$$Sum = A \oplus B = \bar{A}B + A\bar{B}$$
$$Carry = AB$$

Truth Table

Inputs		Outputs	
A	**B**	**Sum**	**Carry**
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Half Subtractor

Logic Diagram

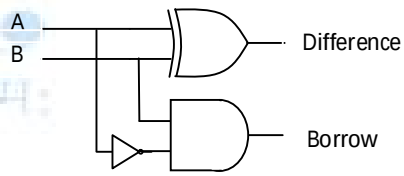

Figure 2: Half Subtractor

Boolean Expressions

$$Difference = A \oplus B = \bar{A}B + A\bar{B}$$
$$Borrow = \bar{A}B$$

Truth Table

Inputs		Outputs	
A	**B**	**Difference**	**Borrow**
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Full Adder

Logic Diagram

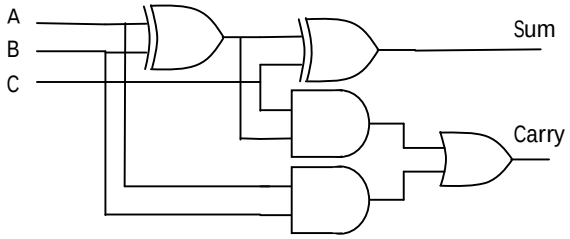

Figure 3: Gate level diagram of full adder

Boolean Expressions

$$Sum = A \oplus B \oplus C$$
$$= \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$
$$Carry = AB + AC + BC$$

Truth Table

Inputs			Outputs	
A	B	C	Sum	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 Subtractor

Logic Diagram

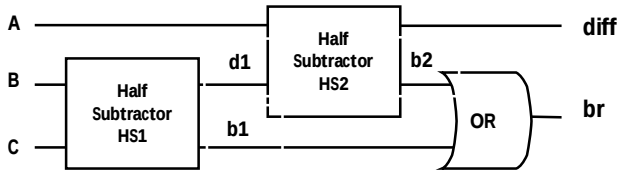

Figure 4: Full subtractor using two half subtractors

Boolean Expressions

$$Differenc = A \oplus B \oplus C$$
$$= \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$
$$Borrow = \bar{A}B + BC + \bar{A}C$$

Truth Table

Inputs			Outputs	
A	B	C	Difference	Borrow
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Exercise#1: Design half adder in **data flow style** of modeling.

Exercise#2: Design half subtractor in **behavioral style** (using either **if-then** or **case when**) of modeling. (Take reference of example given at page no. 7 in Manual Xilix).

Exercise#3: Design **gate level logic diagram** of full adder (shown in Figure 3) in **structural style** of modeling. (Take reference of example given at page no. 7 in Manual Xilix).

Note: Use following steps (for Exercise 3) in structural style of modeling

- Write VHDL code for all types of components (or Add them if you already have) required in the design.
- Copy the entity of each component to be called in the main program without making any change in the entity name and its port variables.
- Replace the keyword **"entity"** with the keyword **"component"** and delete keyword **"is"**
- Replace the entity name after the keyword **"end"** with the keyword **"component"**.
- Write **port map** statements for each component used in the design.