University of Engineering and Management
Institute of Engineering & Management, New Town Campus
Department of Computer Science & Engineering

Computer Organization & Architecture Laboratory
PCCCS492

EXPERIMENT NO.:

TITLE: Implementation of Half Adder, Half Subtractor, Full Adder, Full Subtractor using VHDL (Dataflow model).

OBJECTIVE: To implement of Half Adder, Half Subtractor, Full Adder, Full Subtractor using VHDL (Dataflow model).

Subtractor, Full Adder, Full Subtractor using VHDL (Dataflow model).

OBJECTIVE: To implement of Half Adder, Half Subtractor, Full Adder, Full Subtractor using VHDL (Dataflow model).

THEORY:

Half Adder

Half adder is a combinational logic circuit with two inputs and two outputs. The half adder circuit is designed to add two single bit binary number A and B. It is the basic building block for addition of two single bit numbers. This circuit has two outputs carry and sum.

$S = A'B + AB'$

$C = AB$

Department of Computer Science & Engineering

Computer Organization & Architecture Laboratory

PCCCS492

Block diagram

Truth Table

INPUT		OUTPUT	
A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Full Adder

Full adder is developed to overcome the drawback of Half Adder circuit. It can add two one-bit numbers A and B, and carry c. The full adder is a three input and two output combinational circuit.

$$Sum = \bar{A}\bar{B}C_{in} + \bar{A}B\bar{C}_{in} + A\bar{B}\bar{C}_{in} + ABC_{in}$$

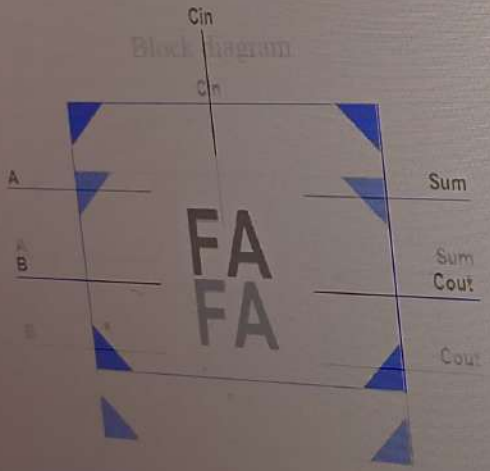
$$C_{out} = AB + A\,C_{in} + B\,C_{in}$$

Full adder is developed to overcome the drawback of Half Adder circuit. It can add two one-bit numbers A and B, and carry c. The full adder is a three input and two output combinational circuit.

$Sum = \overline{A}\,\overline{B}C_{in} + \overline{A}B\overline{C}_{in} + A\overline{B}\,\overline{C}_{in} + ABC_{in}$

$C_{out} = AB + AC_{in} + BC_{in}$

Block diagram



Truth Table

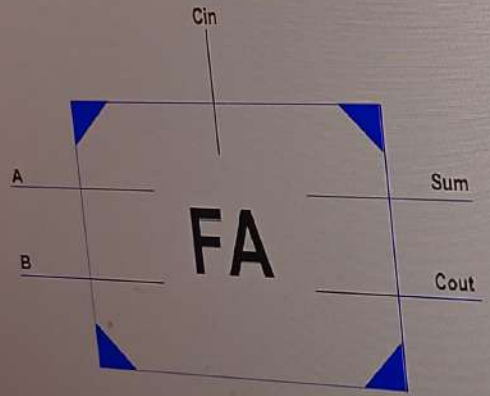
INPUT			OUTPUT	
A	B	Cin	Cout	Sum
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Full adder is developed to overcome the drawback of Half Adder circuit. It can add two one-bit numbers A and B, and carry c. The full adder is a three input and two output combinational circuit.

$$Sum = \overline{A}\,\overline{B}C_{in} + \overline{A}B\overline{C}_{in} + A\overline{B}\,\overline{C}_{in} + ABC_{in}$$

$$C_{out} = AB + A\,C_{in} + B\,C_{in}$$

Block diagram



Truth Table

INPUT			OUTPUT	
A	B	Cin	Cout	Sum
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	1
1	1	0	1	0
1	1	1	1	1

Computer Organization & Architecture Laboratory

PCCCS492

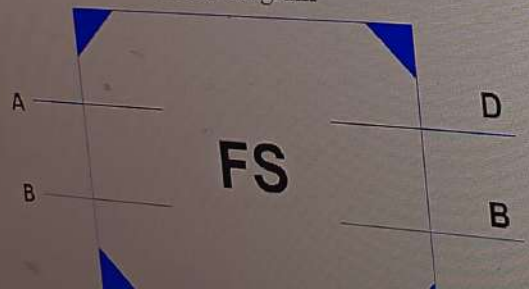
Half Subtractor

Half subtractor is a combination circuit with two inputs and two outputs (difference and borrow). It produces the difference between the two binary bits at the input and also produces an output (Borrow) to indicate if a 1 has been borrowed. In the subtraction (A-B), A is called as Minuend bit and B is called as Subtrahend bit.

Difference $= A \oplus B$

Borrow $= \overline{A}B$

Block diagram



Truth Table

INPUT		OUTPUT	
A	B	B	D
0	0	0	0
0	1	1	1
1	0	0	1

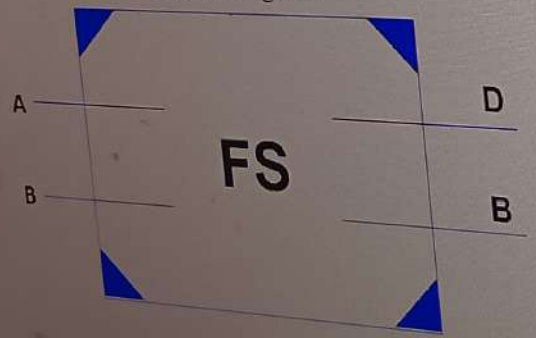
Half Subtractor

Half subtractor is a combination circuit with two inputs and two outputs (difference and borrow). It produces the difference between the two binary bits at the input and also produces an output (Borrow) to indicate if a 1 has been borrowed. In the subtraction (A-B), A is called as Minuend bit and B is called as Subtrahend bit.

Difference $= A \oplus B$

Borrow $= \bar{A}B$

Block diagram



Truth Table

INPUT		OUTPUT	
A	B	B	D
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

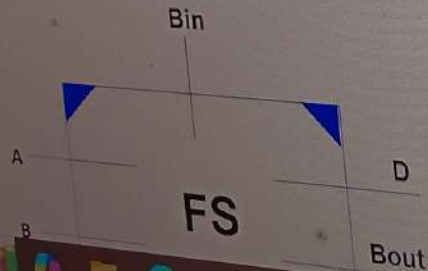
Full Subtractor

The disadvantage of a half subtractor is overcome by full subtractor. The full subtractor is a combinational circuit with three inputs A,B,Bin and two output D and Bout. A is the 'minuend', B is 'subtrahend', Bin is the 'borrow' produced by the previous stage, D is the difference output and Bout is the borrow output.

D=A′B′Bin + AB′Bin′ + A′BBin′ + ABBin

Bout=A′Bin + A′B + BBin

Block diagram



Truth Table

INPUT			OUTPUT	
A	B	Bin	D	Bout
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

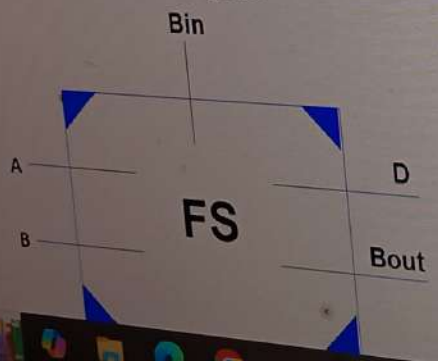
Full Subtractor

The disadvantage of a half subtractor is overcome by full subtractor. The full subtractor is a combinational circuit with three inputs A,B,Bin and two output D and Bout. A is the 'minuend', B is 'subtrahend', Bin is the 'borrow' produced by the previous stage, D is the difference output and Bout is the borrow output.

D=A'B'Bin + AB'Bin' + A'BBin' + ABBin

Bout=A'Bin + A'B + BBin

Block diagram



Truth Table

INPUT			OUTPUT	
A	B	Bin	D	Bout
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

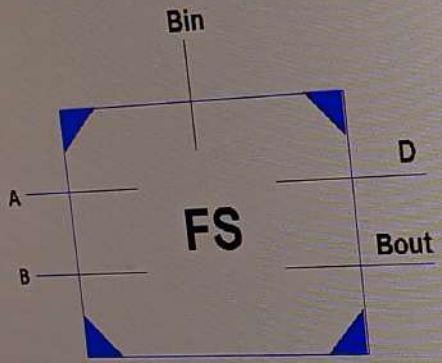
Bout=A'Bin + A'B + BBin

Block diagram

Truth Table

INPUT			OUTPUT	
A	B	Bin	D	Bout
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

Block diagram



Truth Table

INPUT			OUTPUT	
A	B	Bin	D	Bout
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

PROBLEM STATEMENT:

1. Write a VHDL program for a Half adder using the dataflow modeling style.
2. Write a VHDL program for a Half Subtractor using the dataflow modeling style.
3. Write a VHDL program for a Full Adder using the dataflow modeling style.
4. Write a VHDL program for a Full Subtrator using the dataflow modeling style.

SOFTWARE USED:

SOFTWARE USED:

PROGRAM:

1. Half adder:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity half_adder is
  port(a,b:in bit; sum,carry:out bit);
end half_adder;

architecture data of half_adder is
begin
  sum<= a xor b;
  carry <= a and b;
end data;
```

```
port(a,b:in bit; sum,carry:out bit);
end half_adder;

architecture data of half_adder is
begin
  sum<= a xor b;
  carry <= a and b;
end data;
```

2. Half-Subtractor:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
```

```
carry <= a and b;
end data;
```

2. Half-Subtractor:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
```

```
entity half_sub is
  port ( a,b : in std_logic;
       dif,bo: out std_logic );
end half_sub;
architecture sub_arch of half_sub is
  begin
    dif <= a xor b;
    bo <= (not a) and b;
  end sub_arch;
```

3. Full adder:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
```

3. Full adder:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity full_adder is port
(a,b,c:in bit; sum,carry:out bit);
end full_adder;

architecture data of full_adder is
```

```
begin
  sum<= a xor b xor c;
  carry <= ((a and b) or (b and c) or (a and c));
end data;
```

4. Full-Subtractor :

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity full_sub is
  port (A,B,Bin: in std_logic;
  D, Bout: out std_logic);
  end full_sub;
```

```
begin
  sum<= a xor b xor c;
  carry <= ((a and b) or (b and c) or (a and c));
end data;
```

4. Full-Subtractor :

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity full_sub is
  port (A,B,Bin: in std_logic;
  D, Bout: out std_logic);
  end full_sub;
```

4. Full-Subtractor :

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;


entity full_sub is
  port (A,B,Bin: in std_logic;
D, Bout: out std_logic);
end full_sub;


  architecture data of full_sub is
  begin
    D <= A xor B xor Bin;
    Bout <= ((NOT(A) and Bin) or (NOT(A) and B) or (B and Bin));
  end data;
```

```
carry <= ((a and b) or (b and c) or (a and c));
end data;
```

4. Full-Subtractor :

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity full_sub is
  port (A,B,Bin: in std_logic;
D, Bout: out std_logic);
end full_sub;

architecture data of full_sub is
begin
  D <= A xor B xor Bin;
  Bout <= ((NOT(A) and Bin) or (NOT(A) and B) or (B and Bin));
end data;
```

University of Engineering and Management
Institute of Engineering & Management, New Town Campus
Department of Computer Science & Engineering

Computer Organization & Architecture Laboratory
PCCCS492

TIMING DIAGRAM:

CONCLUSION: