EX. NO: 01	**ADDERS AND SUBTRACTORS**
DATE :	

AIM: To simulate Adders and Subtractors in behavior, structural, and dataflow model of Verilog HDL.

SOFTWARE USED: Xilinx Vivado

HARDWARE USED: Basys3 FPGA Board

ADDERS:

VERILOG HDL CODE:

Half Adder in Behavioral model:

```
module half_adder_behav (S,C,A,B);
input A,B;
output S,C;
reg S,C;
always@(A or B)
        begin
        if(A==0 && B==0)
                begin S=0;C=0; end
        else if(A==0 && B==1)
                begin S=1;C=0; end
      else if(A==1 && B==0)
                begin S=1;C=0; end
      else
                begin S=0;C=1; end
        end
endmodule
```

Half Adder in Structural model:

```
module half_adder_struct(S,C,A,B);
input A, B;
output S, C;
xor (S,A,B);
and (C,A,B);
endmodule
```

Half Adder in Data flow model:

```
module half_adder_data(S,C,A,B);
input A,B;
output S,C;
assign S=A ^ B;
assign C= A & B;
endmodule
```

Half Adder Test bench:

```
module half_adder_test;
reg A,B;
wire S,C;
//half_adder_behav HA1(S,C,A,B);
//half_adder_data HA2(S,C,A,B);
half_adder_struct HA3(S,C,A,B);
initial
begin
       A =0;      B = 0;
#10    A = 0;      B = 1;
#10    A = 1;      B = 0;
#10    A = 1;      B = 1;
#10 $stop;
$monitor ("A=%b, B=%b, C=%b, S=%b", A, B, C, S);
 end
endmodule
```

Full Adder in Behavioral model:

```
module full_adder_behav(a,b,c,sum,carry);
input a,b,c;
output sum ,carry;
reg sum,carry;
always @(a or b or c)
begin
if (a==0 && b==0 && c==0)
begin sum=0;carry=0; end
else if(a==0 && b==0 && c==1)
begin sum=1;carry=0;end
else if(a==0 && b==1 && c==0)
begin sum=1;carry=0;end
else if(a==0 && b==1 && c==1)
begin sum=0;carry=1;end
else if(a==1 && b==0 && c==0)
begin sum=1;carry=0;end
else if(a==1 && b==0 && c==1)
begin sum=0;carry=1;end
else if(a==1 && b==1 && c==0)
begin sum=0;carry=1;end
else
 begin sum=1;carry=1;end
end
endmodule
```

Full Adder in Structural model:

```
module full_adder_structural(a,b,c,sum,carry);
input a,b,c;
output sum,carry;
wire s1,c1,c2;
halfadder HA1(a,b,s1,c1);
halfadder HA2(s1,c,sum,c2);
or(carry,c1,c2);
endmodule

module halfadder (A, B,Sum,Carry);
input A, B;
output Sum, Carry;
xor (Sum,A,B);
and (Carry,A,B);
endmodule
```

Full Adder in Data flow model:

```
module full_adder_data(a,b,c,sum,carry);
input a,b,c;
output sum,carry;
assign sum=a^b^c;
assign carry =(a&b)|(b&c)|(c&a);
endmodule
```

Test bench for Full Adder:

```
module full_adder_test;
reg a,b,c;
wire sum, carry;
full_adder_structural FA1(a,b,c,sum,carry);
//full_adder_behav FA2(a,b,c,sum,carry);
//full_adder_data FA3(a,b,c,sum,carry);
initial
begin
a=0;b=0;c=0;
#10 a=0;b=0;c=1;
#10 a=0;b=1;c=0;
#10 a=0;b=1;c=1;
#10 a=1;b=0;c=0;
#10 a=1;b=0;c=1;
#10 a=1;b=1;c=0;
#10 a=1;b=1;c=1;
#10 $stop;
$monitor("a=%b,b=%b,c=%b,sum=%b,carry=%b",a,b,c,sum,carry);
end
endmodule
```

4-Bit Parallel Adder(Ripple Carry Adder):

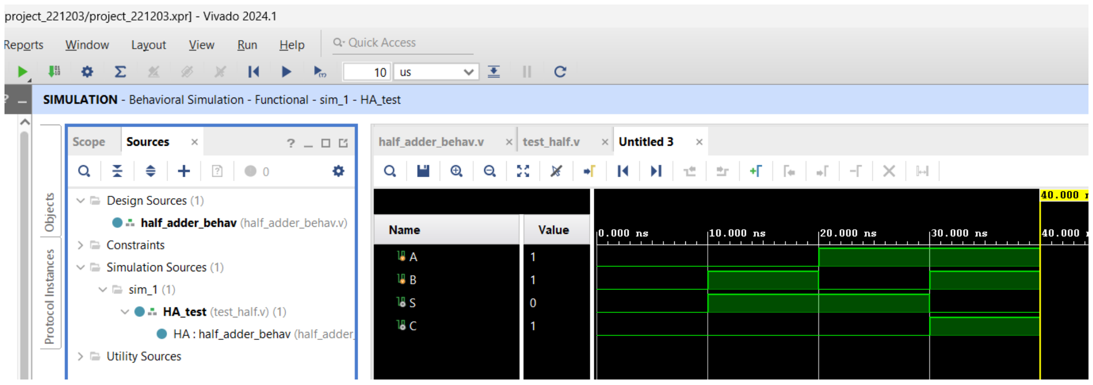
```
module full_adder (a,b,c,sum,carry);
input a,b,c;
output sum,carry;
assign sum=a^b^c;
assign carry =(a&b)|(b&c)|(c&a);
endmodule

module RCA(a,b,c,sum, carry);
input [3:0]a,b;
input c;
output [3:0]sum;
output carry;
wire c1,c2,c3;
full_adder fa1(a[0],b[0],c,sum[0], c1);
full_adder fa2(a[1],b[1],c1,sum[1], c2);
full_adder fa3(a[2],b[2],c2,sum[2], c3);
full_adder fa4(a[3],b[3],c3,sum[3], carry);
endmodule
```

Test bench for 4-Bit Parallel Adder:

```
module RCA_test;
reg [3:0]a,b;
reg c;
wire [3:0]sum;
wire carry;
RCA full_adder(a,b,c,sum, carry);
initial
begin
$monitor("%d a=%b, b=%b, c=%b, sum=%b, carry=%b",$time, a, b, c, sum, carry);
        a = 4'b1001; b = 4'b1100; c = 1; //3
#100  a = 4'b0001; b = 4'b1111; c = 1; // 31
#100  a = 4'b0011; b = 4'b1011; c = 0; // 59
#100  a = 4'b0101; b = 4'b0111; c = 1; // 87
#100  a = 4'b0111; b = 4'b0011; c = 0; // 115
#100  a = 4'b1000; b = 4'b1111; c = 1; // 143
#100  a = 4'b1010; b = 4'b1011; c = 0; // 171
#100  a = 4'b1100; b = 4'b0111; c = 1; // 199
end
endmodule
```

SIMULATION WAVEFORM:

Half Adder in Behavioral model:

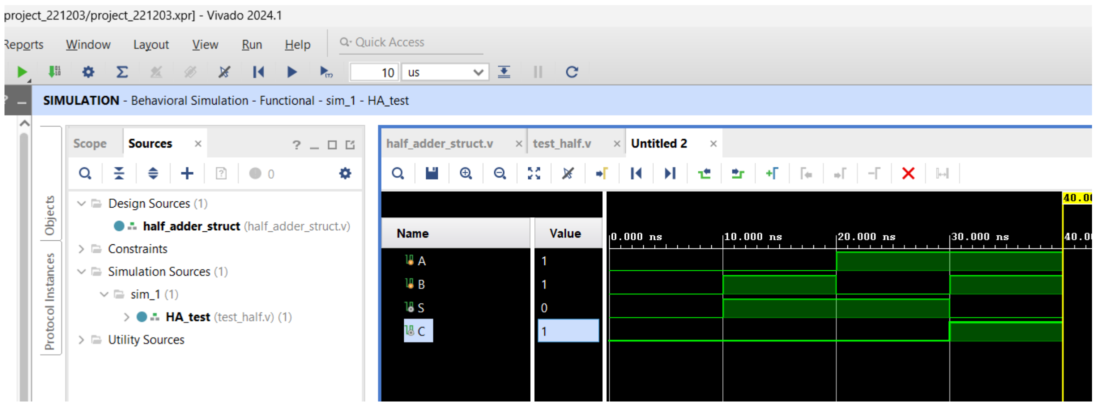

project_221203/project_221203.xpr] - Vivado 2024.1

SIMULATION - Behavioral Simulation - Functional - sim_1 - HA_test

- Design Sources (1)
  - half_adder_behav (half_adder_behav.v)
- Constraints
- Simulation Sources (1)
  - sim_1 (1)
    - HA_test (test_half.v) (1)
      - HA : half_adder_behav (half_adder
- Utility Sources

Name	Value
A	1
B	1
S	0
C	1

Half Adder in Structural model:

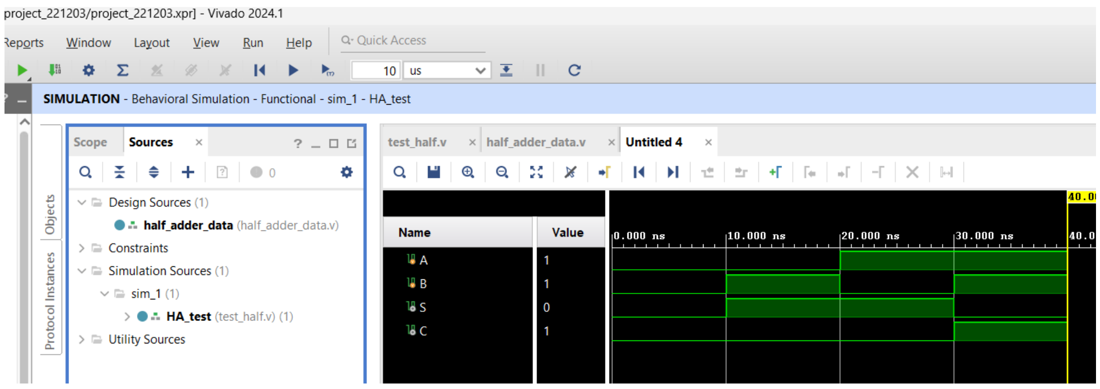

project_221203/project_221203.xpr] - Vivado 2024.1

SIMULATION - Behavioral Simulation - Functional - sim_1 - HA_test

- Design Sources (1)
  - half_adder_struct (half_adder_struct.v)
- Constraints
- Simulation Sources (1)
  - sim_1 (1)
    - HA_test (test_half.v) (1)
- Utility Sources

Name	Value
A	1
B	1
S	0
C	1

Half Adder in Data flow model:

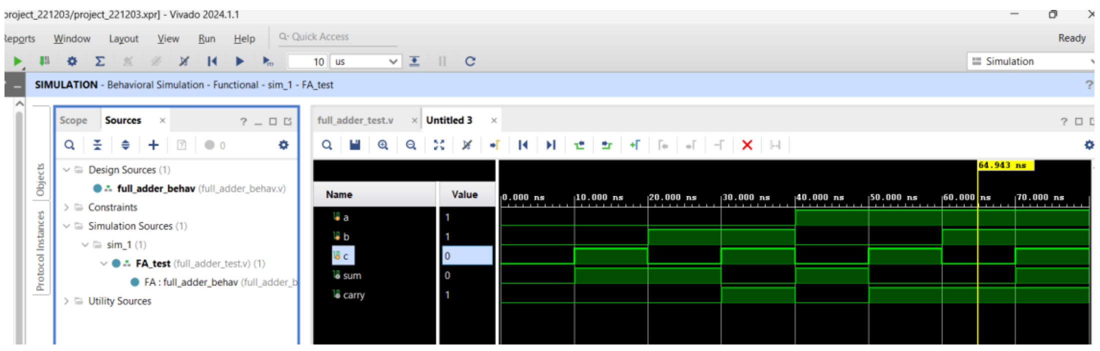

project_221203/project_221203.xpr] - Vivado 2024.1

SIMULATION - Behavioral Simulation - Functional - sim_1 - HA_test

- Design Sources (1)
  - half_adder_data (half_adder_data.v)
- Constraints
- Simulation Sources (1)
  - sim_1 (1)
    - HA_test (test_half.v) (1)
- Utility Sources

Name	Value
A	1
B	1
S	0
C	1

Full Adder in Behavioral model:

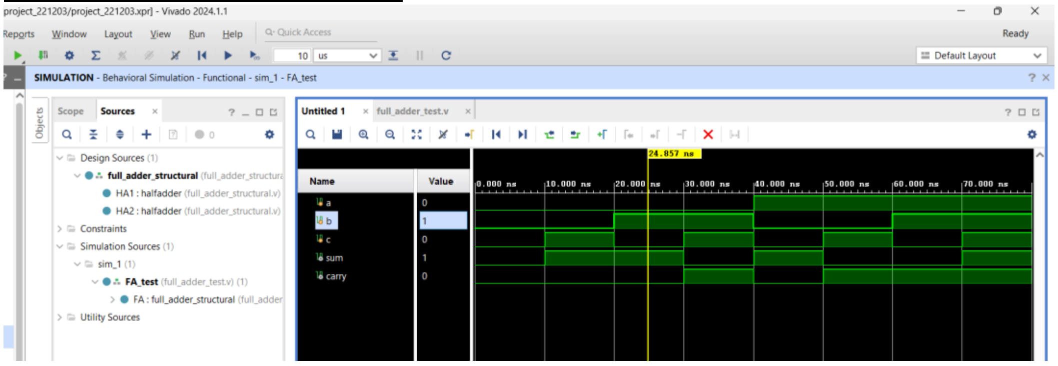

roject_221203/project_221203.xpr] - Vivado 2024.1.1

SIMULATION - Behavioral Simulation - Functional - sim_1 - FA_test

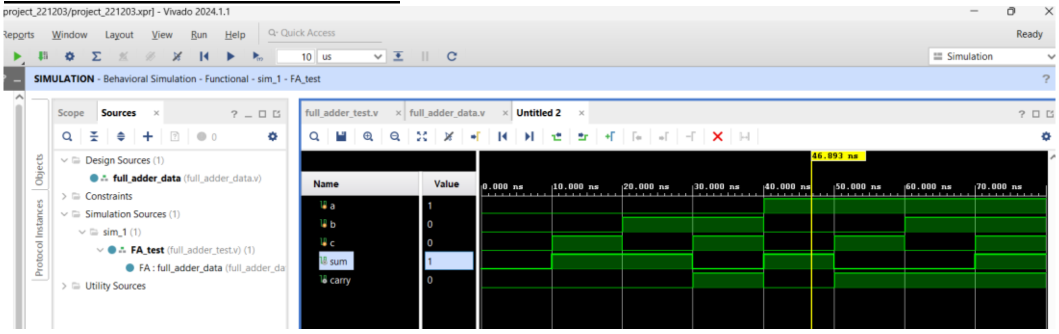
- Design Sources (1)
  - full_adder_behav (full_adder_behav.v)
- Constraints
- Simulation Sources (1)
  - sim_1 (1)
    - FA_test (full_adder_test.v) (1)
      - FA : full_adder_behav (full_adder_b
- Utility Sources

Name	Value
a	1
b	1
c	0
sum	0
carry	1

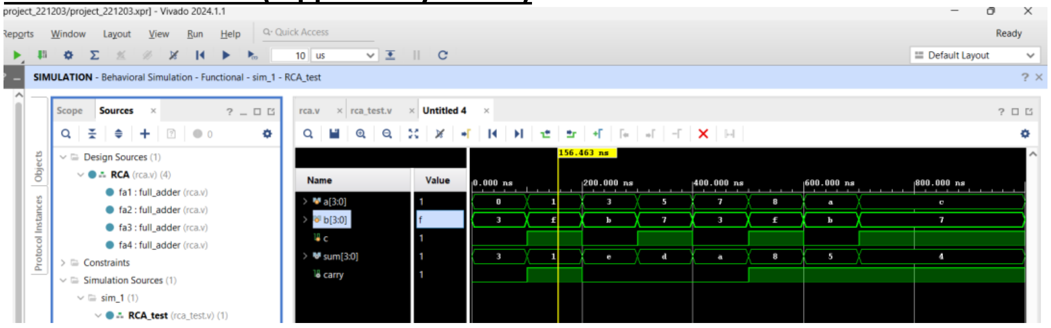
Full Adder in Structural model:


project_221203/project_221203.xpr] - Vivado 2024.1.1
SIMULATION - Behavioral Simulation - Functional - sim_1 - FA_test
Design Sources (1)
full_adder_structural (full_adder_structural.v)
HA1 : halfadder (full_adder_structural.v)
HA2 : halfadder (full_adder_structural.v)
Constraints
Simulation Sources (1)
sim_1 (1)
FA_test (full_adder_test.v) (1)
FA : full_adder_structural (full_adder
Utility Sources
Untitled 1
full_adder_test.v
24.857 ns
Name
Value
a 0
b 1
c 0
sum 1
carry 0

Full Adder in Data flow model:


project_221203/project_221203.xpr] - Vivado 2024.1.1
SIMULATION - Behavioral Simulation - Functional - sim_1 - FA_test
Design Sources (1)
full_adder_data (full_adder_data.v)
Constraints
Simulation Sources (1)
sim_1 (1)
FA_test (full_adder_test.v) (1)
FA : full_adder_data (full_adder_da
Utility Sources
full_adder_test.v
full_adder_data.v
Untitled 2
46.893 ns
Name
Value
a 1
b 0
c 0
sum 1
carry 0

4-Bit Parallel Adder(Ripple Carry Adder):



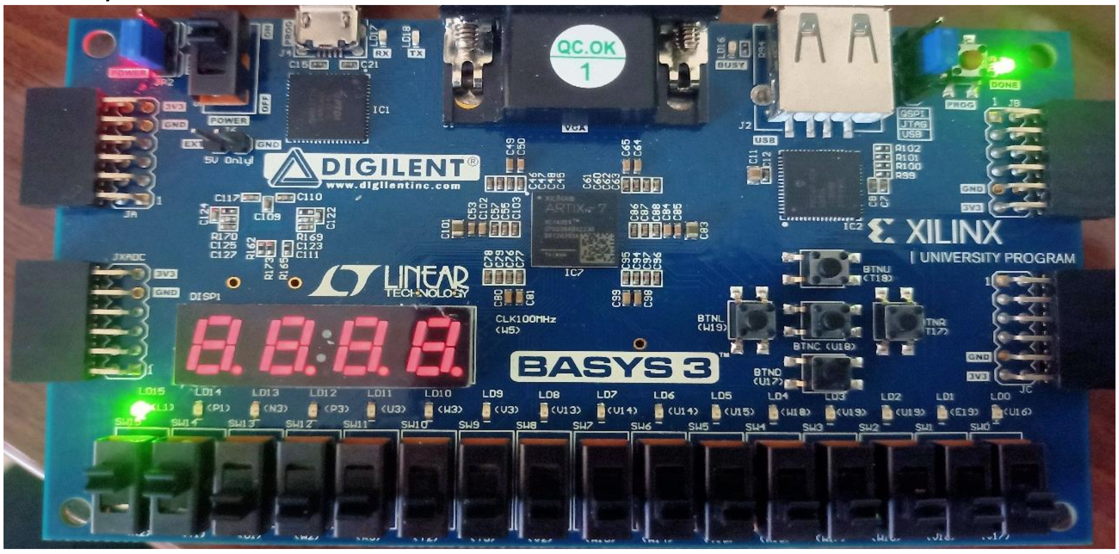
HADWARE OUTPUT:

Half Adder:

I/O ports:

A: R2	CARRY: L1
B: T1	SUM : P1

For A=1, B=1

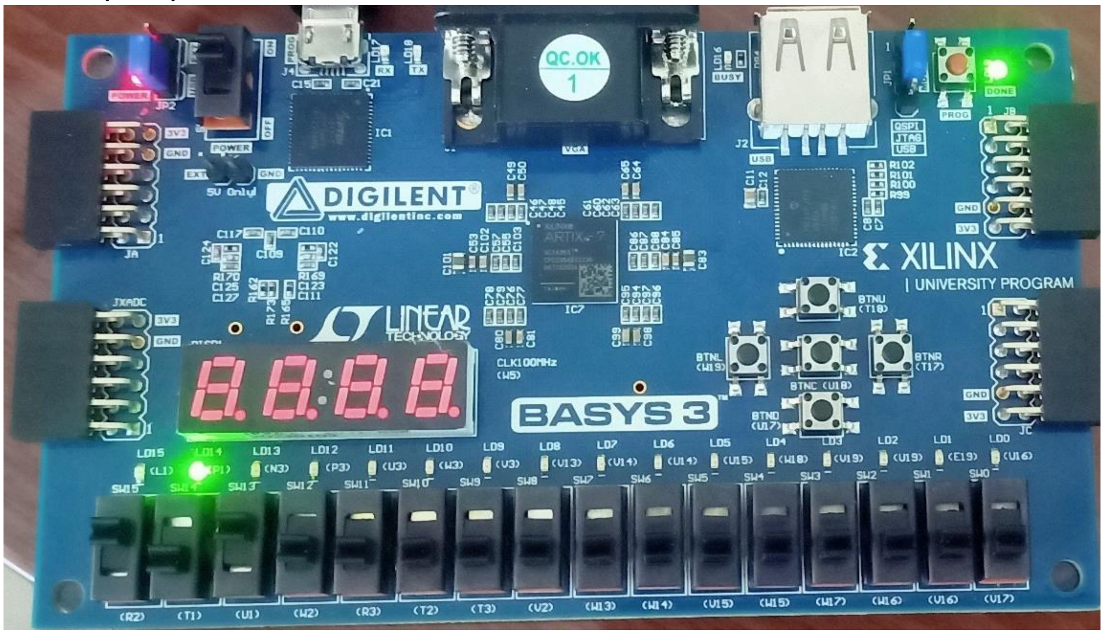


<u>Full Adder</u>:

<u>I/O ports</u>:

a: R2	carry: P1
b: T1	sum : L1
c: U1	

For A=1, B=0, C=1

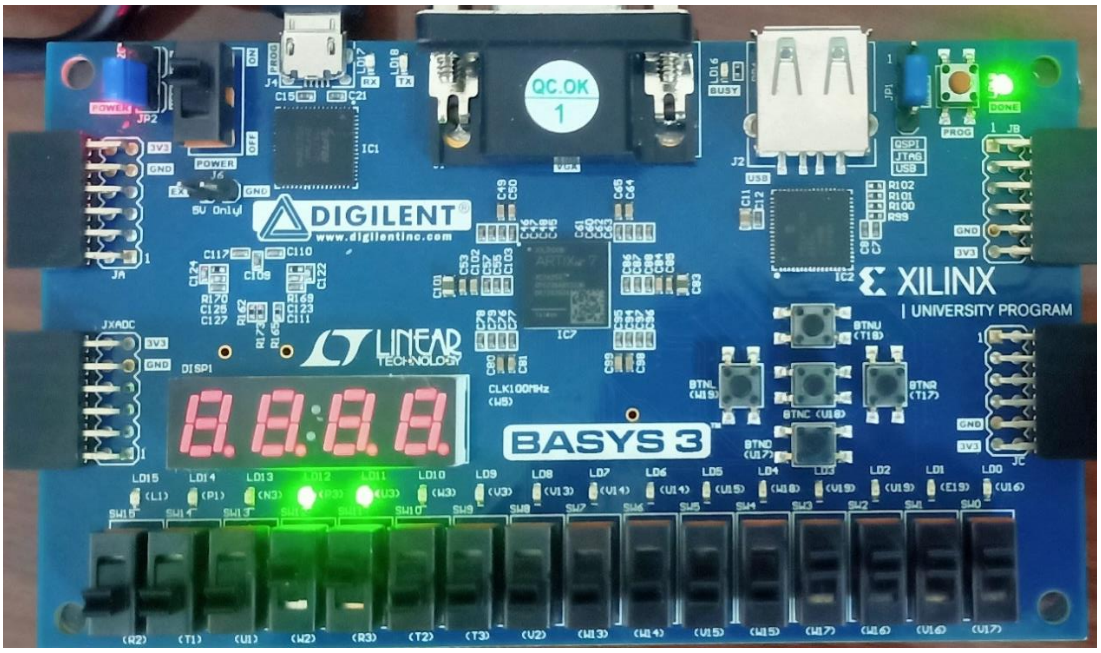


<u>4-Bit Parallel Adder(Ripple Carry Adder)</u>:

<u>I/O ports</u>:

a[3]: R2	b[3]: W17	sum[3]: L1
a[2]: T1	b[2]: W16	sum[2]: P1
a[1]: U1	b[1]: V16	sum[1]: N3
a[0]: W2	b[0]: V17	sum[0]: P3
	Cin : R3	Cout : U3

For a[3]=0, a[2]=0, a[1]=0, a[0]=1
b[3]=1, b[2]=1, b[1]=1, b[0]=1
Cin=1



TRUTH TABLE:

Half Adder:

Input		Output	
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Sum = A'B + AB'=A ⊕ B
Carry = AB

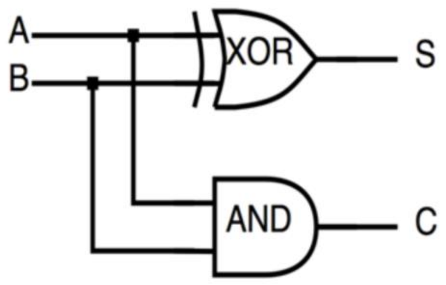
Full Adder:

Input			Output	
A	B	Cin	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

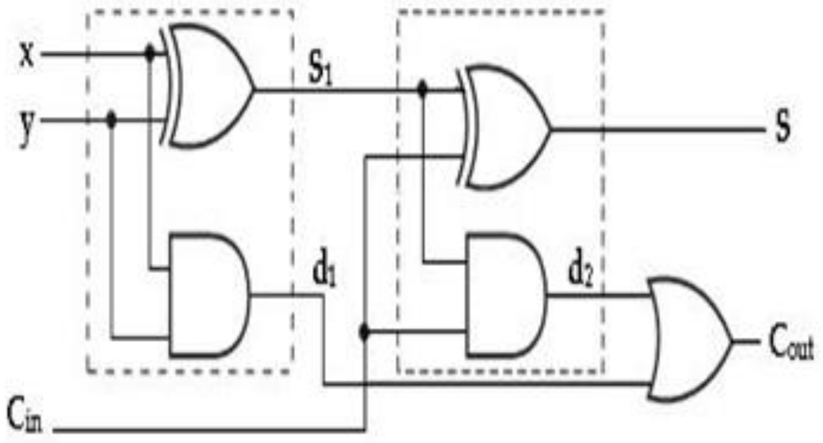
Sum = A ⊕ B ⊕ Cin
Carry = (AB) + CinA ⊕ CinB

CIRCUIT DIAGRAMS:

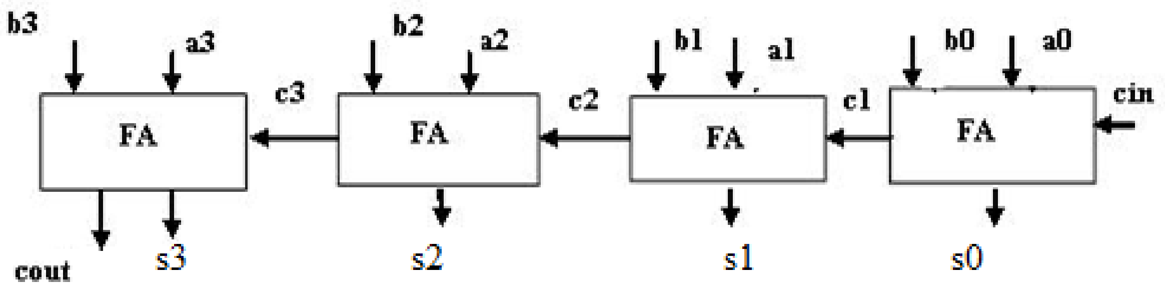
Half Adder:



Full Adder:



4-Bit Parallel Adder:



SUBTRACTORS:

VERILOG HDL CODE:

Half Subtractor in Behavioral model:

```
module
half_subtractor_behav(a,b,diff,borrow);
  input a, b;
  output diff, borrow;
  reg diff, borrow;
always @(a or b)
 begin
    if (a == 0 && b == 0)
     begin diff = 0; borrow = 0; end
     else if (a == 0 && b == 1)
     begin diff = 1; borrow = 1; end
     else if (a == 1 && b == 0)
     begin diff = 1; borrow = 0; end
     else if (a == 1 && b == 1)
     begin diff = 0; borrow = 0; end
  end
endmodule
```

Half Subtractor in Structural model:

```
module
half_subtractor_struct(a,b,differ,borrow);
    input a,b;
    output differ,borrow;
    wire x;
    xor (differ,a,b);
    not (x,a);
    and (borrow,x,b);
endmodule
```

Half Subtractor in Data flow model:

```
module
half_subtractor_data(a,b,differ,borrow);
    input a,b;
    output differ,borrow;
    assign differ=a^b;
    assign borrow=(~a)&b;
endmodule
```

Test bench for Half Subtractor:

```
module half_subtractor_test;
reg A,B;
wire DIFFER,BORROW;
half_subtractor_behav HS1(A,B,DIFFER,BORROW);
//half_subtractor_struct HS2(A,B,DIFFER,BORROW);
//half_subtractor_data HS3(A,B,DIFFER,BORROW);
initial
begin
       A =0;      B = 0;
#10    A = 0;     B = 1;
#10    A = 1;     B = 0;
#10    A = 1;     B = 1;
#10 $stop;
$monitor ("A=%b, B=%b, DIFFER=%b, BORROW=%b", A, B, DIFFER, BORROW);
 end
endmodule
```

Full Subtractor in Behavioral model:

```
module full_subtractor_behav(a,b,c,differ,borrow);
 input a,b,c;
 output reg differ,borrow;
 always @(a or b or c)
 begin
 if (a==0 && b==0 && c==0)
   begin differ=0; borrow=0; end
 else if (a==0 && b==0 && c==1)
   begin differ=1; borrow=1; end
 else if (a==0 && b==1 && c==0)
   begin differ=1; borrow=1; end
 else if (a==0 && b==1 && c==1)
   begin differ=0; borrow=1; end
 else if (a==1 && b==0 && c==0)
   begin differ=1; borrow=0; end
 else if (a==1 && b==0 && c==1)
   begin differ=0; borrow=0; end
 else if (a==1 && b==1 && c==0)
   begin differ=0; borrow=0; end
   else
   begin differ=1; borrow=1; end
 end
endmodule
```

Full Subtractor in Structural model:

```
module half_subtractor_struct(a,b,differ,borrow);
   input a,b;
   output differ,borrow;
   wire x;
   xor (differ,a,b);
   not (x,a);
   and (borrow,x,b);
endmodule

module full_subtractor_struct(a,b,c,differ,borrow);
input a,b,c;
output differ,borrow;
wire X,Y,Z;
half_subtractor_struct HS1(a,b,X,Y);
half_subtractor_struct HS2(X,c,differ,Z);
or (borrow,Z,Y);
endmodule
```

Full Subtractor in Data flow model:

```
module full_subtractor_dataflow(a,b,c,differ,borrow );
input a,b,c;
output differ,borrow;
assign  differ=(a^b^c);
assign  borrow=(c&(~(a^b)))|((~a)&b);
endmodule
```

Test Bench for Full Subtractor:

```
module full_subtractor_test;
reg a,b,c;
wire differ,borrow;
full_subtractor_struct FS1(a,b,c,differ,borrow);
//full_subtractor_dataflow FS2(a,b,c,differ,borrow );
//full_subtractor_behav FS3(a,b,c,differ,borrow);
initial
begin
a=0;b=0;c=0;
#10 a=0;b=0;c=1;
#10 a=0;b=1;c=0;
#10 a=0;b=1;c=1;
#10 a=1;b=0;c=0;
#10 a=1;b=0;c=1;
#10 a=1;b=1;c=0;
#10 a=1;b=1;c=1;
#10 $stop;
$monitor("a=%b,b=%b,c=%b,differ=%b,borrow=%b",a,b,c,differ,borrow);
end
endmodule
```

4-Bit Parallel Subtractor:

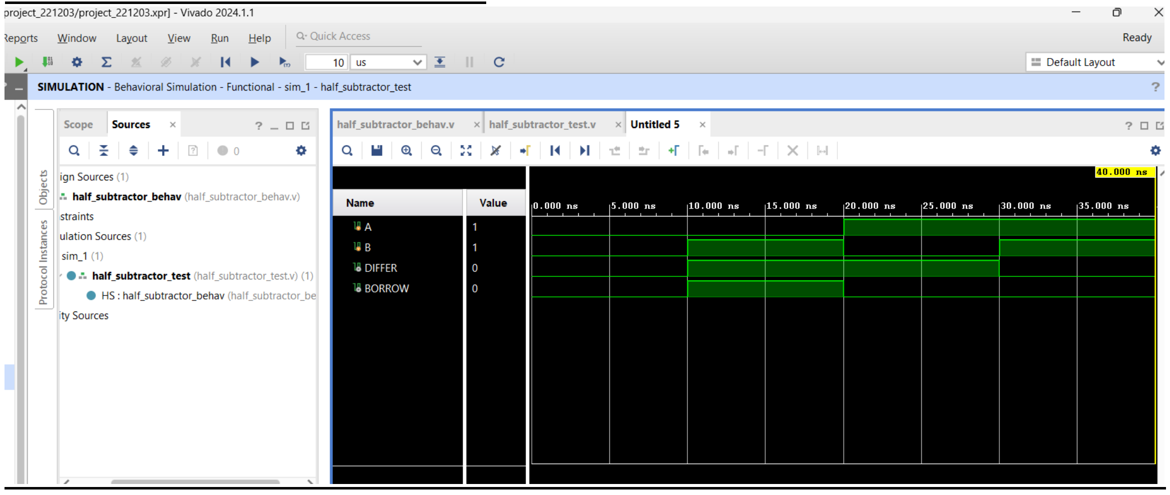
```
module four_bit_parallel_subtractor(a,b,c,differ,borrow);
 input [3:0] a,b;
 input c;
 output [3:0] differ;
 output borrow;
 wire b1,b2,b3;
 full_subtractor_data HS1(a[0],b[0],c,differ[0], b1);
 full_subtractor_data HS2(a[1],b[1],b1,differ[1], b2);
 full_subtractor_data HS3(a[2],b[2],b2,differ[2], b3);
 full_subtractor_data HS4(a[3],b[3],b3,differ[3], borrow);
endmodule


module full_subtractor_data(a,b,c,differ,borrow);
input a,b,c;
output differ,borrow;
assign  differ = a ^ b ^ c;
assign  borrow = (c & (~(a ^ b))) | ((~a) & b);
endmodule
```

<u>Test Bench for 4-Bit Parallel Subtractor</u>:

```
module FOUR_BIT_SUB_TEST;
reg [3:0] a,b;
reg c;
wire [3:0] differ;
wire borrow;
four_bit_parallel_subtractor HS(a,b,c,differ, borrow);
initial begin
        a = 4'b0000; b = 4'b0011; c = 0;
#100 a = 4'b0001; b = 4'b1111; c = 1;
#100 a = 4'b0011; b = 4'b1011; c = 0;
#100 a = 4'b0101; b = 4'b0111; c = 1;
#100 a = 4'b0111; b = 4'b0011; c = 0;
#100 a = 4'b1000; b = 4'b1111; c = 1;
#100 a = 4'b1010; b = 4'b1011; c = 0;
#100 a = 4'b1100; b = 4'b0111; c = 1;
$monitor("%d a=%b, b=%b, c=%b, differ=%b, borrow=%b", $time, a, b, c, differ, borrow);
end
endmodule
```

<u>SIMULATION WAVEFORMS</u>:

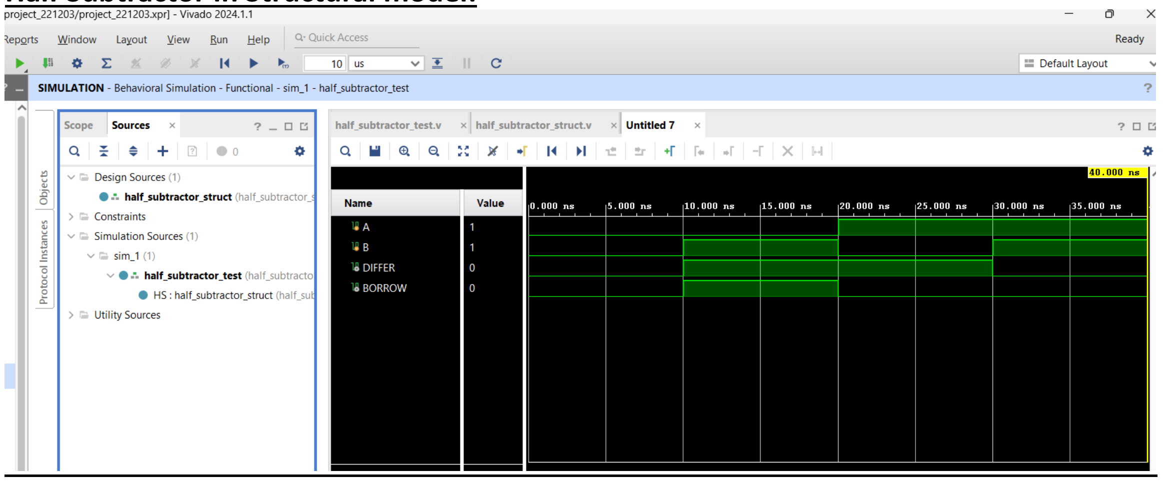
<u>Half Subtractor in Behavioral model</u>:


project_221203/project_221203.xpr] - Vivado 2024.1.1

SIMULATION - Behavioral Simulation - Functional - sim_1 - half_subtractor_test

Name	Value
A	1
B	1
DIFFER	0
BORROW	0

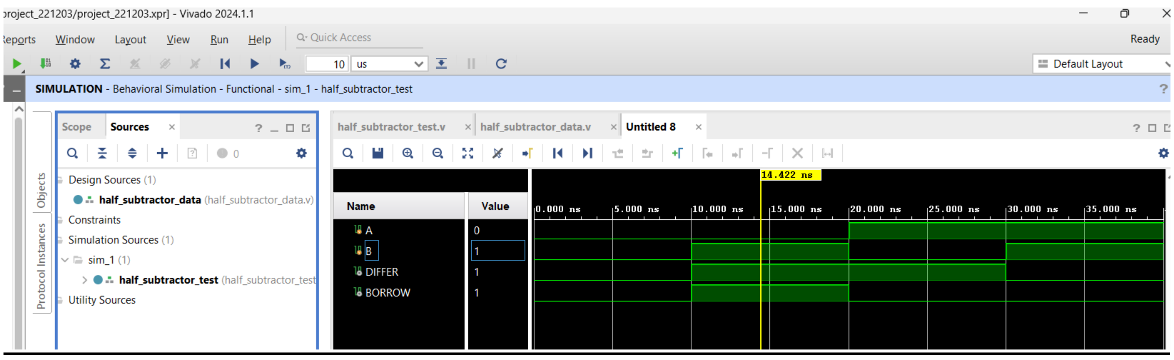
<u>Half Subtractor in Structural model:</u>


project_221203/project_221203.xpr] - Vivado 2024.1.1

SIMULATION - Behavioral Simulation - Functional - sim_1 - half_subtractor_test

Name	Value
A	1
B	1
DIFFER	0
BORROW	0

Half Subtractor in Data flow model:

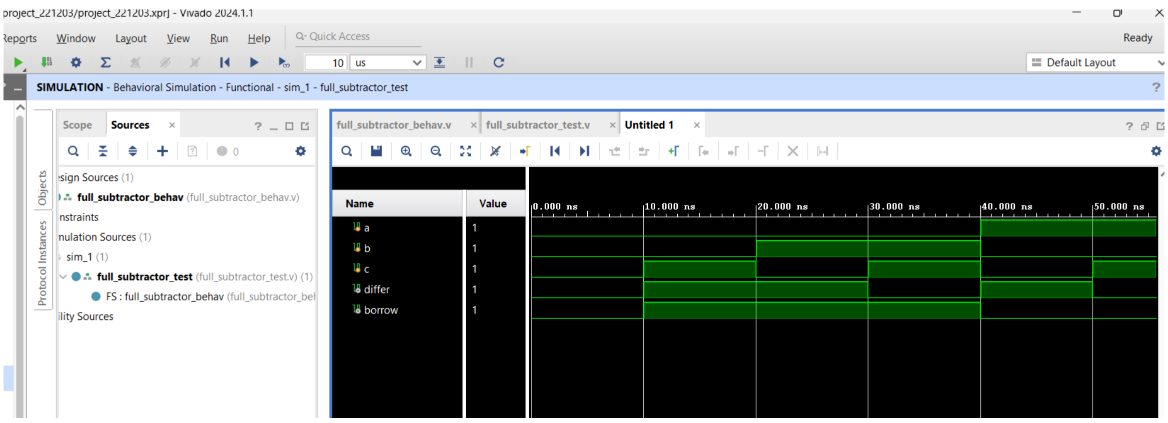


roject_221203/project_221203.xpr] - Vivado 2024.1.1

SIMULATION - Behavioral Simulation - Functional - sim_1 - half_subtractor_test

Name	Value
A	0
B	1
DIFFER	1
BORROW	1

Full Subtractor in Behavioral model:

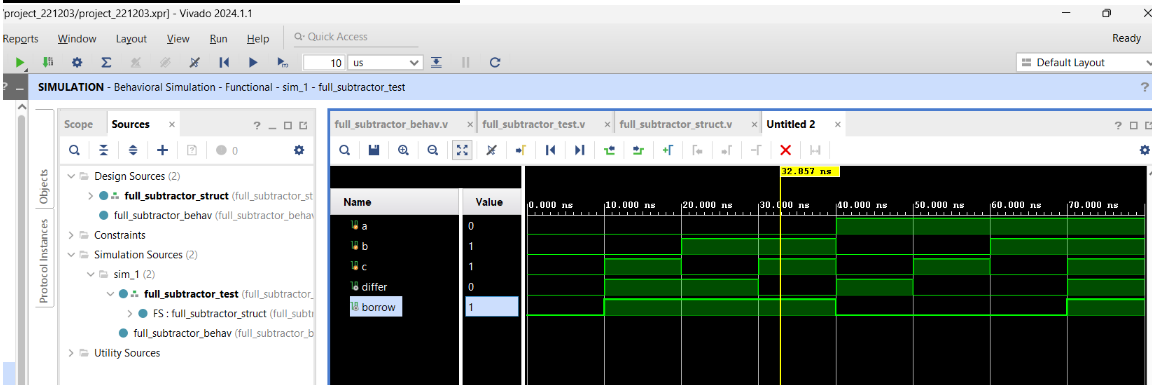


project_221203/project_221203.xpr] - Vivado 2024.1.1

SIMULATION - Behavioral Simulation - Functional - sim_1 - full_subtractor_test

Name	Value
a	1
b	1
c	1
differ	1
borrow	1

Full Subtractor in Structural model:

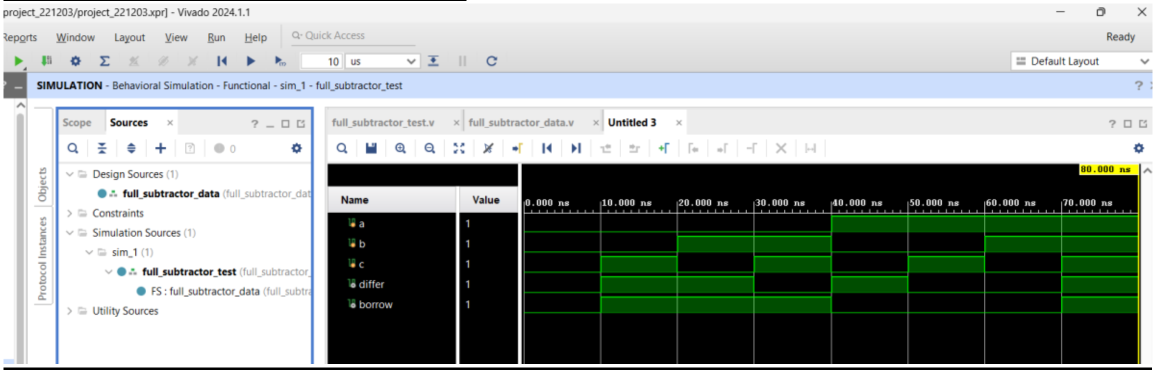


project_221203/project_221203.xpr] - Vivado 2024.1.1

SIMULATION - Behavioral Simulation - Functional - sim_1 - full_subtractor_test

Name	Value
a	0
b	1
c	1
differ	0
borrow	1

Full Subtractor in Data flow model:

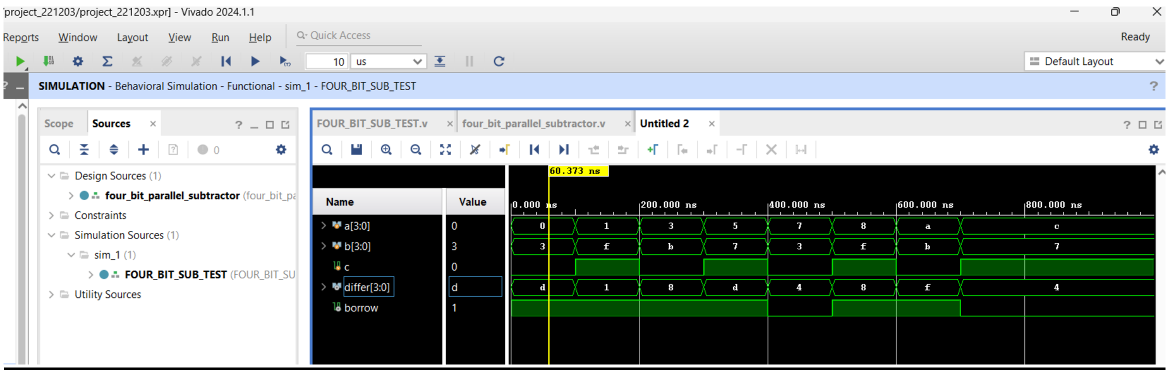


project_221203/project_221203.xpr] - Vivado 2024.1.1

SIMULATION - Behavioral Simulation - Functional - sim_1 - full_subtractor_test

Name	Value
a	1
b	1
c	1
differ	1
borrow	1

4-Bit Parallel Subtractor:



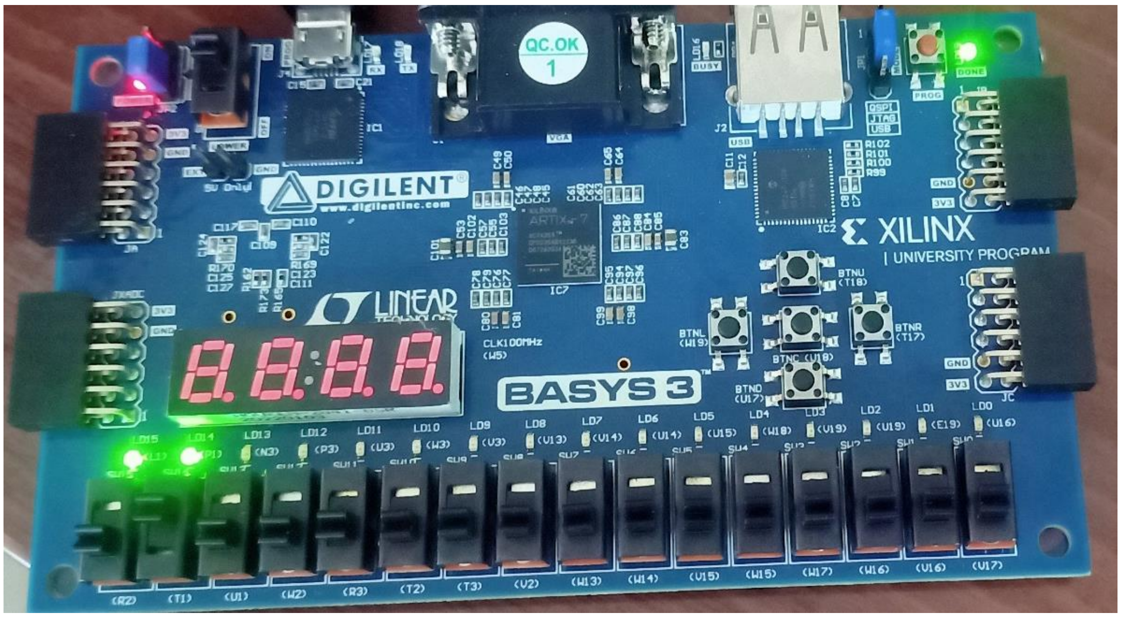
HADWARE OUTPUT:

Half Subtractor:

I/O ports:

a: R2	differ : L1
b: T1	borrow: P1

For A=0, B=1

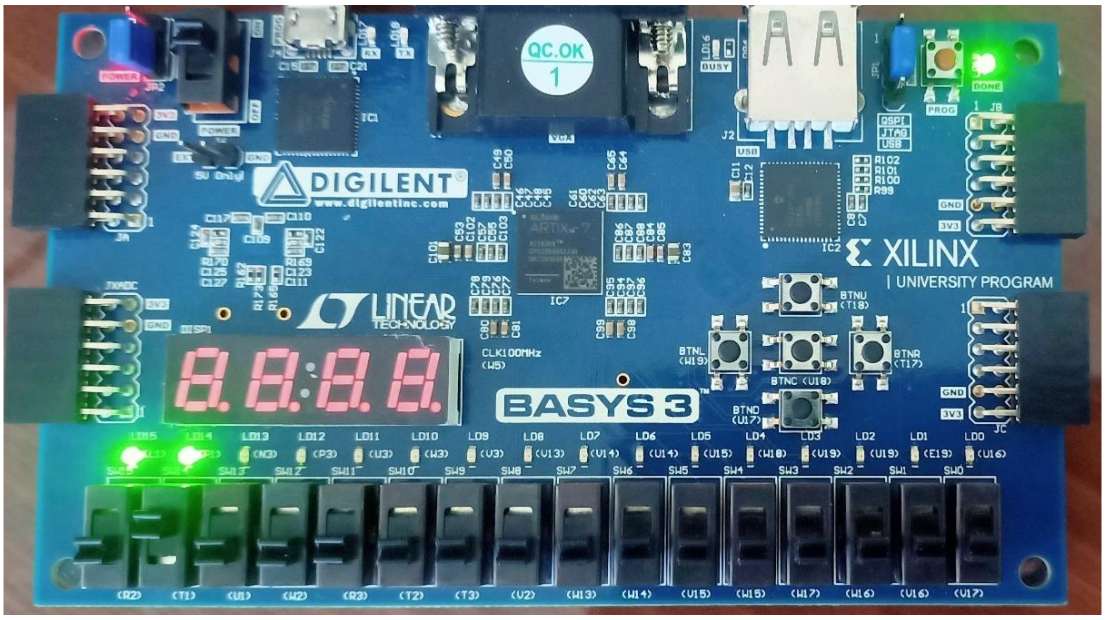


Full Subtractor:

I/O ports:

a: R2	differ : L1
b: T1	borrow: P1
c: U1	

For A=0, B=1, C=0

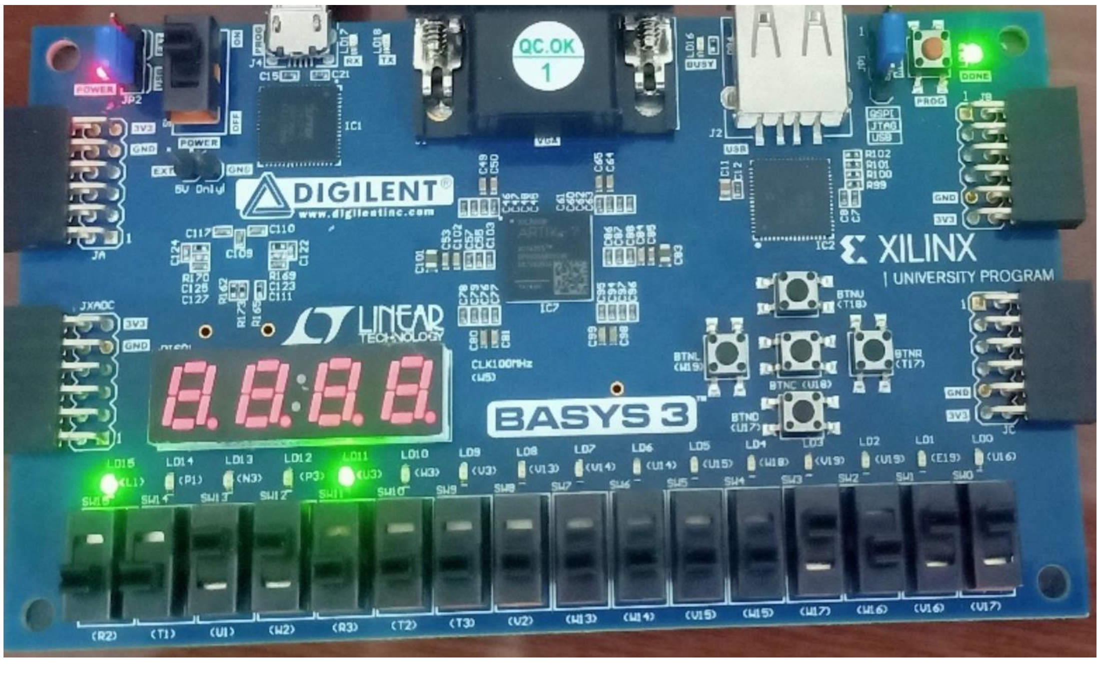


4-Bit Parallel Subtractor:

I/O ports:

a[3]: R2	b[3]: W17	sum[3]: L1
a[2]: T1	b[2]: W16	sum[2]: P1
a[1]: U1	b[1]: V16	sum[1]: N3
a[0]: W2	b[0]: V17	sum[0]: P3
	c(borrow in): R3	borrow(out): U3

For a[3]=0, a[2]=0, a[1]=1, a[0]=1
b[3]=1, b[2]=0, b[1]=1, b[0]=1
borrow in=0



TRUTH TABLE:

Half Subtractor:

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

DIFFERENCE=A⊕B
BORROW = A' B

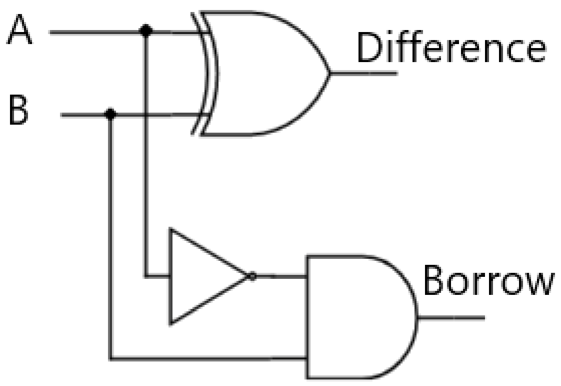
Full Subtractor:

A	B	B_{in}	D	B_{out}
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

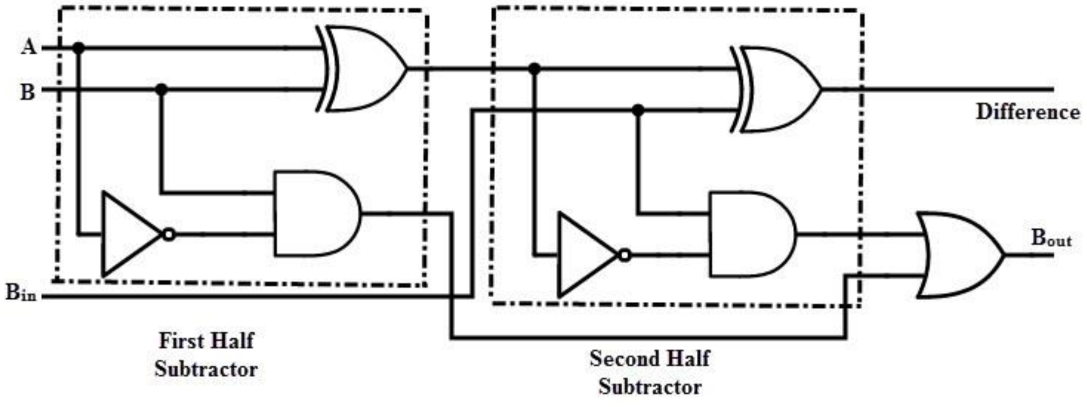
D=A⊕B⊕Bin
Bout = A' Bin + A' B + B Bin

CIRCUIT DIAGRAM:

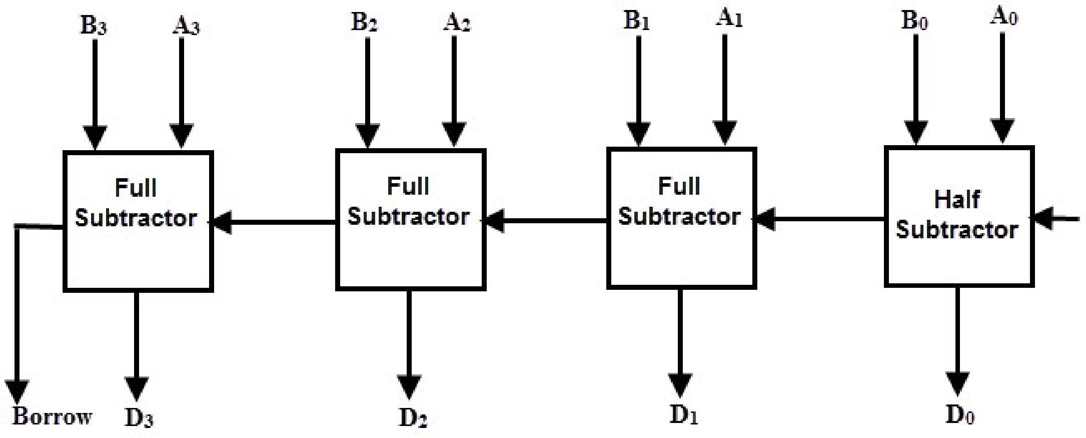
Half Subtractor:



Full Subtractor:



4-Bit Parallel Subtractor:



RESULT:

Thus the logic circuits for the adders and subtractors are designed in Verilog HDL and the output combinations are verified using Xilinx Vivado software on the Basys3 FPGA board.