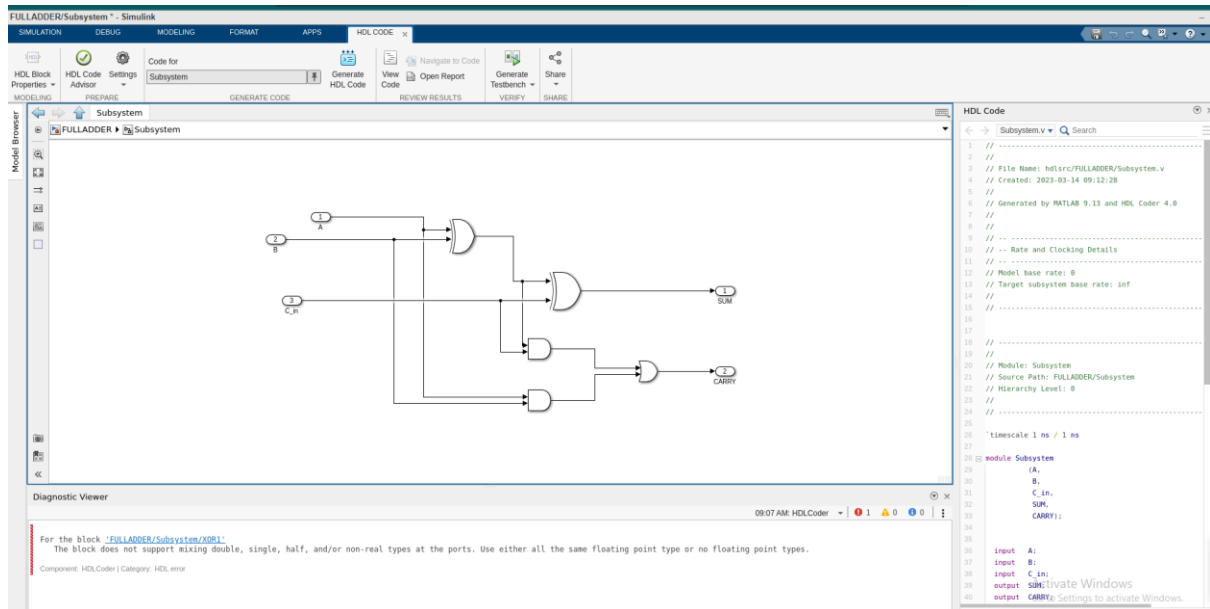


GENERATE VERILOG CODE FOR FULL ADDER.

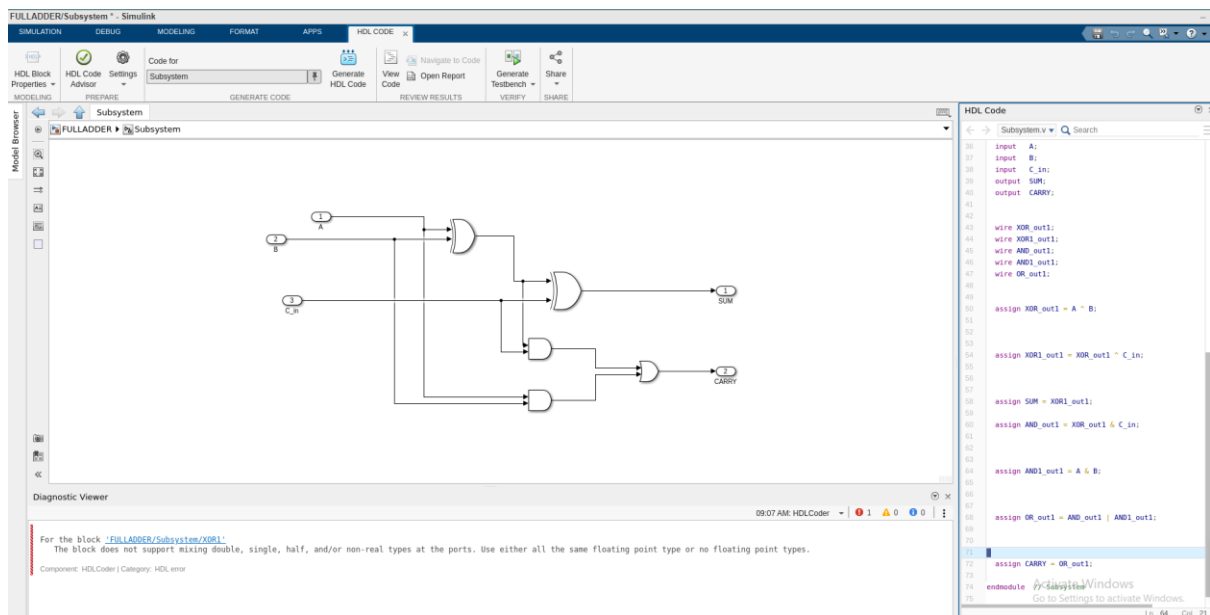
SOLUTION:-



The screenshot shows the Simulink HDL Code window for a Full Adder subsystem. The HDL Code pane displays the following Verilog code:

```
// Subsystem.v
//
// File Name: hdlsrc/FULLADDER/Subsystem.v
// Created: 2023-03-14 09:12:29
//
// Generated by MATLAB 9.13 and HDL Coder 4.8
//
//
// --- Rate and Clocking Details
// Model base rate: 0
// Target subsystem base rate: inf
//
//
// Module: Subsystem
// Source Path: FULLADDER/Subsystem
// Hierarchy Level: 0
//
//
// Timescale 1 ns / 1 ns
//
module Subsystem
    (A,
     B,
     C_in,
     SUM,
     CARRY);
    input A;
    input B;
    input C_in;
    output SUM;
    output CARRY;
endmodule
```

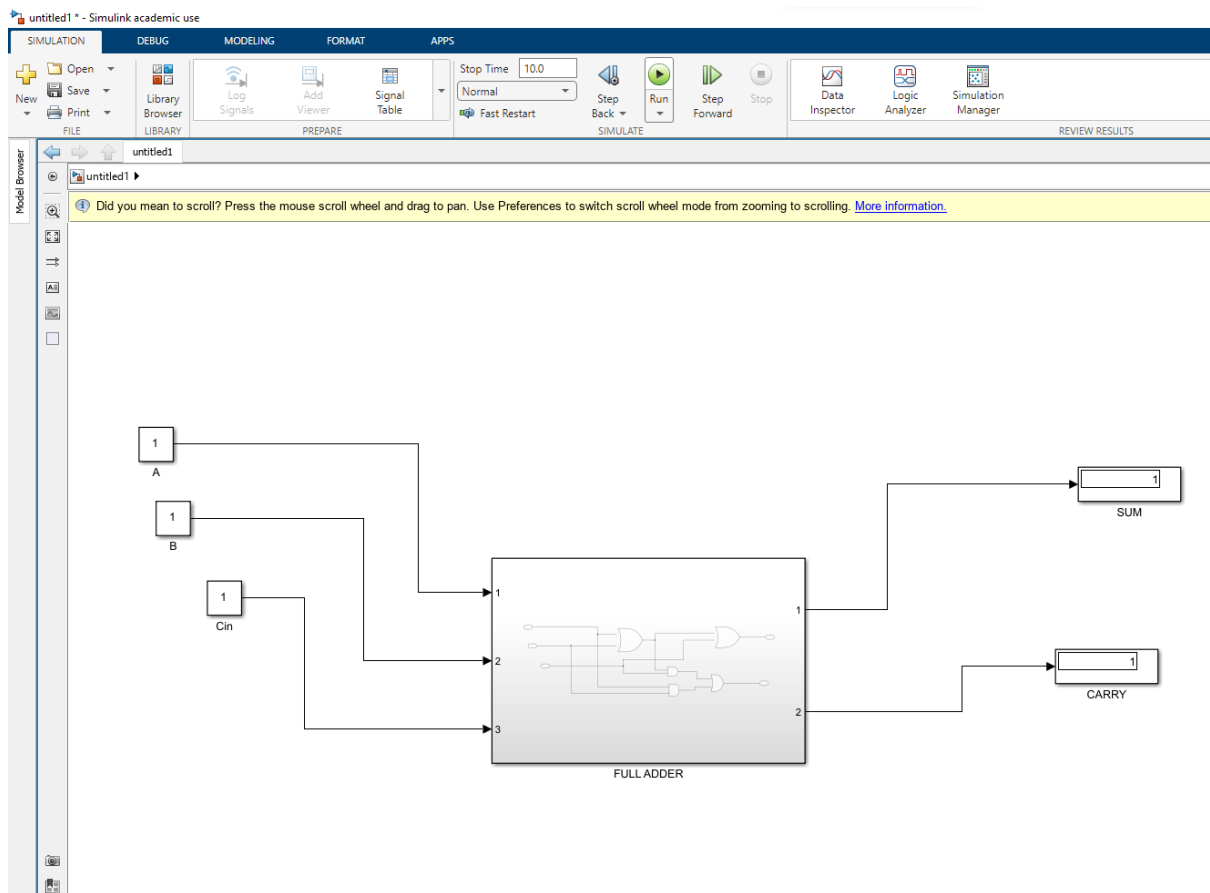
The Model Browser shows the subsystem hierarchy: FULLADDER > Subsystem. The Diagnostic Viewer shows a warning for the block 'FULLADDER/Subsystem/XOR1'.



The screenshot shows the Simulink HDL Code window for a Full Adder subsystem. The HDL Code pane displays the following Verilog code:

```
// Subsystem.v
//
// File Name: hdlsrc/FULLADDER/Subsystem.v
// Created: 2023-03-14 09:12:29
//
// Generated by MATLAB 9.13 and HDL Coder 4.8
//
// --- Rate and Clocking Details
// Model base rate: 0
// Target subsystem base rate: inf
//
//
// Module: Subsystem
// Source Path: FULLADDER/Subsystem
// Hierarchy Level: 0
//
//
// Timescale 1 ns / 1 ns
//
module Subsystem
    (A,
     B,
     C_in,
     SUM,
     CARRY);
    input A;
    input B;
    input C_in;
    output SUM;
    output CARRY;
    wire XOR_out1;
    wire XOR1_out1;
    wire AND_out1;
    wire OR_out1;
    assign XOR_out1 = A ^ B;
    assign XOR1_out1 = XOR_out1 ^ C_in;
    assign SUM = XOR1_out1;
    assign AND_out1 = XOR_out1 & C_in;
    assign AND1_out1 = A & B;
    assign OR_out1 = AND_out1 | AND1_out1;
    assign CARRY = OR_out1;
endmodule
```

The Model Browser shows the subsystem hierarchy: FULLADDER > Subsystem. The Diagnostic Viewer shows a warning for the block 'FULLADDER/Subsystem/XOR1'.



→ VERILOG CODE GENERATED BY HDL CODER:-

```
// File Name: hdlsrc/FULLADDER/Subsystem.v
```

```
// Created: 2023-03-14 09:12:28
```

```
//
```

```
// Generated by MATLAB 9.13 and HDL Coder 4.0
```

```
//
```

```
//
```

```
// -- -----
```

```
// -- Rate and Clocking Details
```

```
// -- -----
```

```
// Model base rate: 0
```

```
// Target subsystem base rate: inf
```

```
//
```

```
// -----
```

```
// -----
```

```
//  
// Module: Subsystem  
// Source Path: FULLADDER/Subsystem  
// Hierarchy Level: 0  
//  
// -----
```

```
`timescale 1 ns / 1 ns
```

```
module Subsystem
```

```
    (A,  
     B,  
     C_in,  
     SUM,  
     CARRY);
```

```
input A;
```

```
input B;
```

```
input C_in;
```

```
output SUM;
```

```
output CARRY;
```

```
wire XOR_out1;
```

```
wire XOR1_out1;
```

```
wire AND_out1;
```

```
wire AND1_out1;
```

```
wire OR_out1;
```

```
assign XOR_out1 = A ^ B;
```

```
assign XOR1_out1 = XOR_out1 ^ C_in;
```

```
assign SUM = XOR1_out1;
```

```
assign AND_out1 = XOR_out1 & C_in;
```

```
assign AND1_out1 = A & B;
```

```
assign OR_out1 = AND_out1 | AND1_out1;
```

```
assign CARRY = OR_out1;
```

```
endmodule // Subsystem
```

HDL Code

Subsystem.v

Search

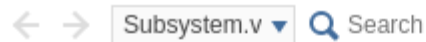
```
1 // -----
2 //
3 // File Name:hdlsrc/FULLADDER/Subsystem.v
4 // Created: 2023-03-14 09:12:28
5 //
6 // Generated by MATLAB 9.13 and HDL Coder 4.0
7 //
8 //
9 // -- -----
10 // -- Rate and Clocking Details
11 // -- -----
12 // Model base rate: 0
13 // Target subsystem base rate: inf
14 //
15 // -----
16
17
18 // -----
19 //
20 // Module: Subsystem
21 // Source Path: FULLADDER/Subsystem
22 // Hierarchy Level: 0
23 //
24 // -----
25
26 `timescale 1 ns / 1 ns
27
28 module Subsystem
29     (A,
30     B,
31     C_in,
32     SUM,
33     CARRY);
34
35
36     input A;
37     input B;
38     input C_in;
39     output SUM;
40     output CARRY;
```

HDL Code

Subsystem.v Search

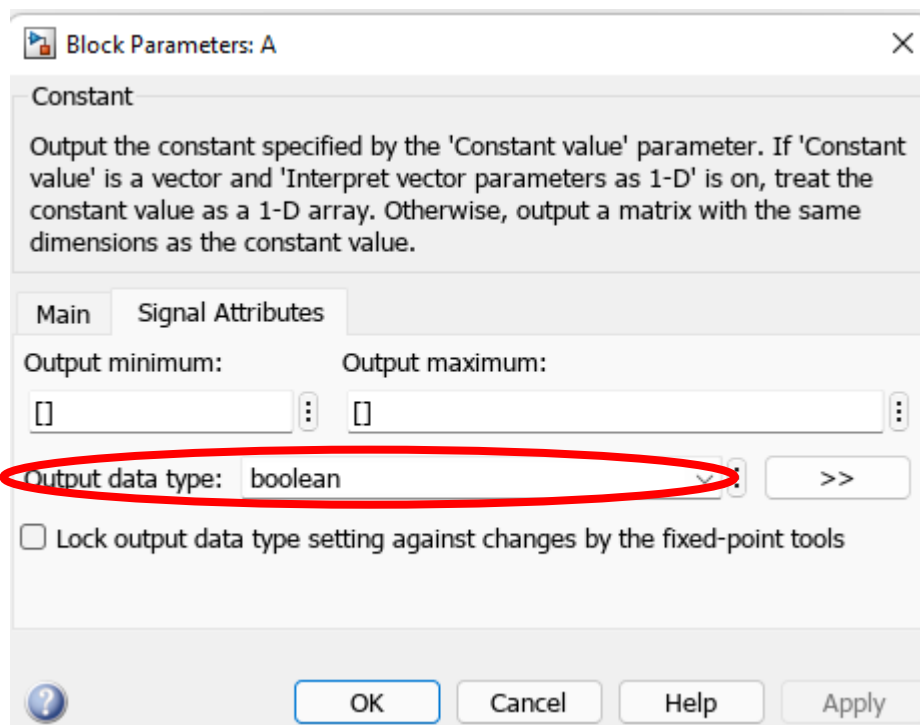
```
21 // Source Path: FULLADDER/Subsystem
22 // Hierarchy Level: 0
23 //
24 // -----
25
26 `timescale 1 ns / 1 ns
27
28 module Subsystem
29     (A,
30      B,
31      C_in,
32      SUM,
33      CARRY);
34
35
36     input  A;
37     input  B;
38     input  C_in;
39     output SUM;
40     output CARRY;
41
42
43     wire XOR_out1;
44     wire XOR1_out1;
45     wire AND_out1;
46     wire AND1_out1;
47     wire OR_out1;
48
49
50     assign XOR_out1 = A ^ B;
51
52
53
54     assign XOR1_out1 = XOR_out1 ^ C_in;
55
56
57
58     assign SUM = XOR1_out1;
59     assign AND_out1 = XOR_out1 & C_in;
```

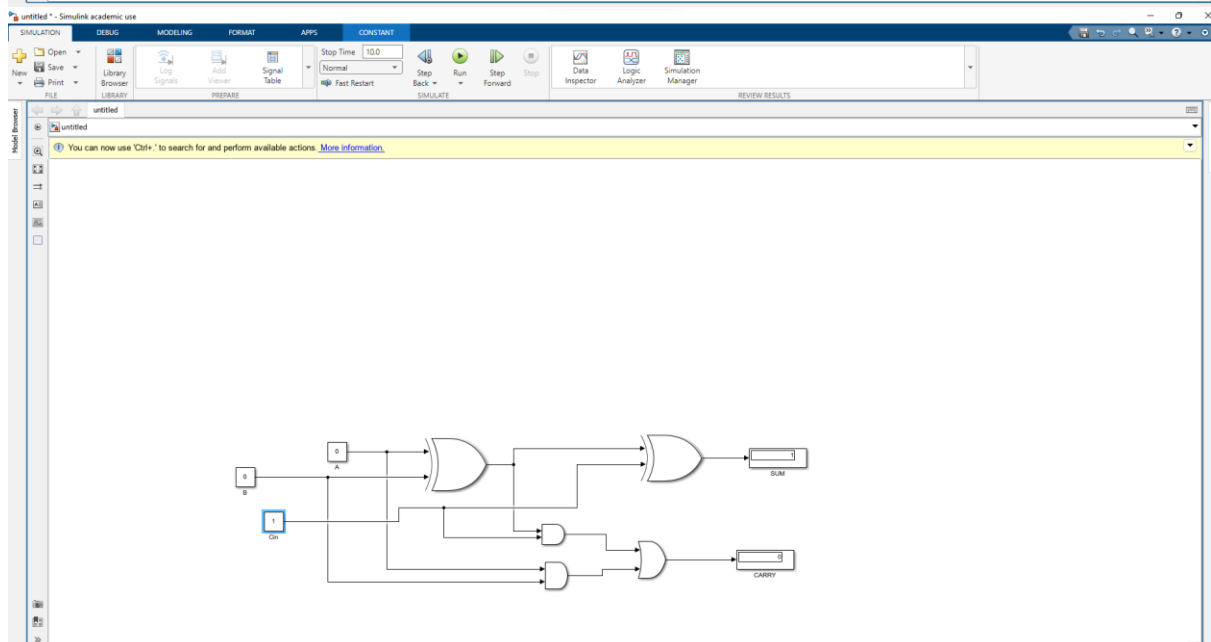
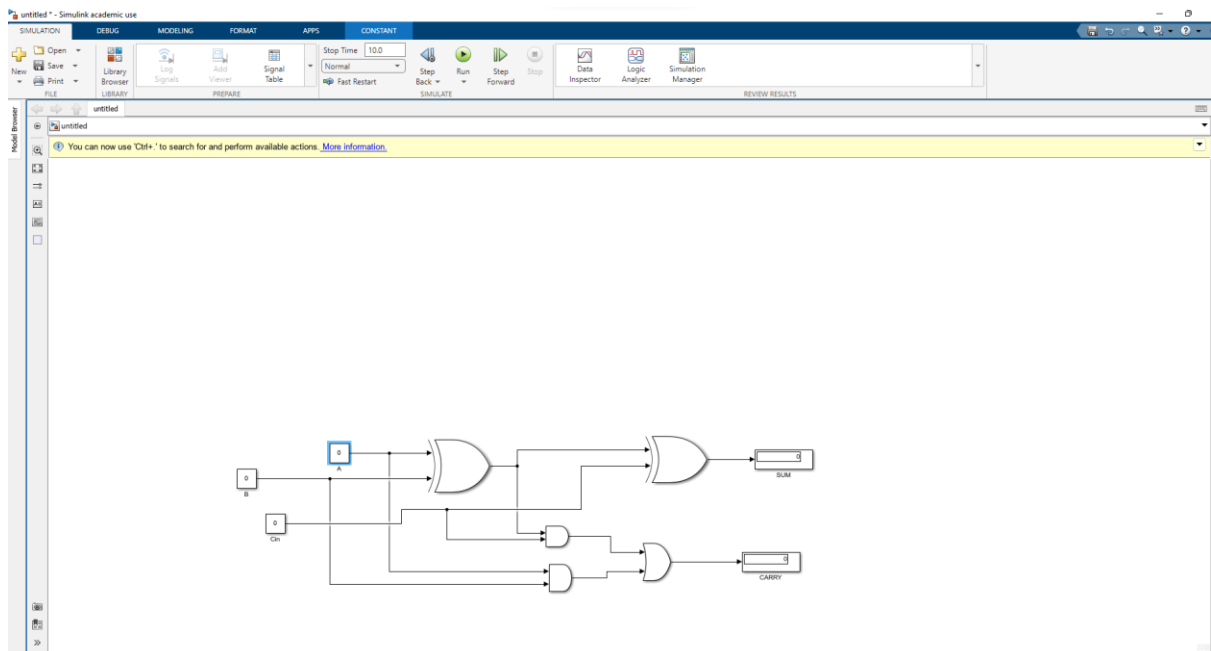
HDL Code

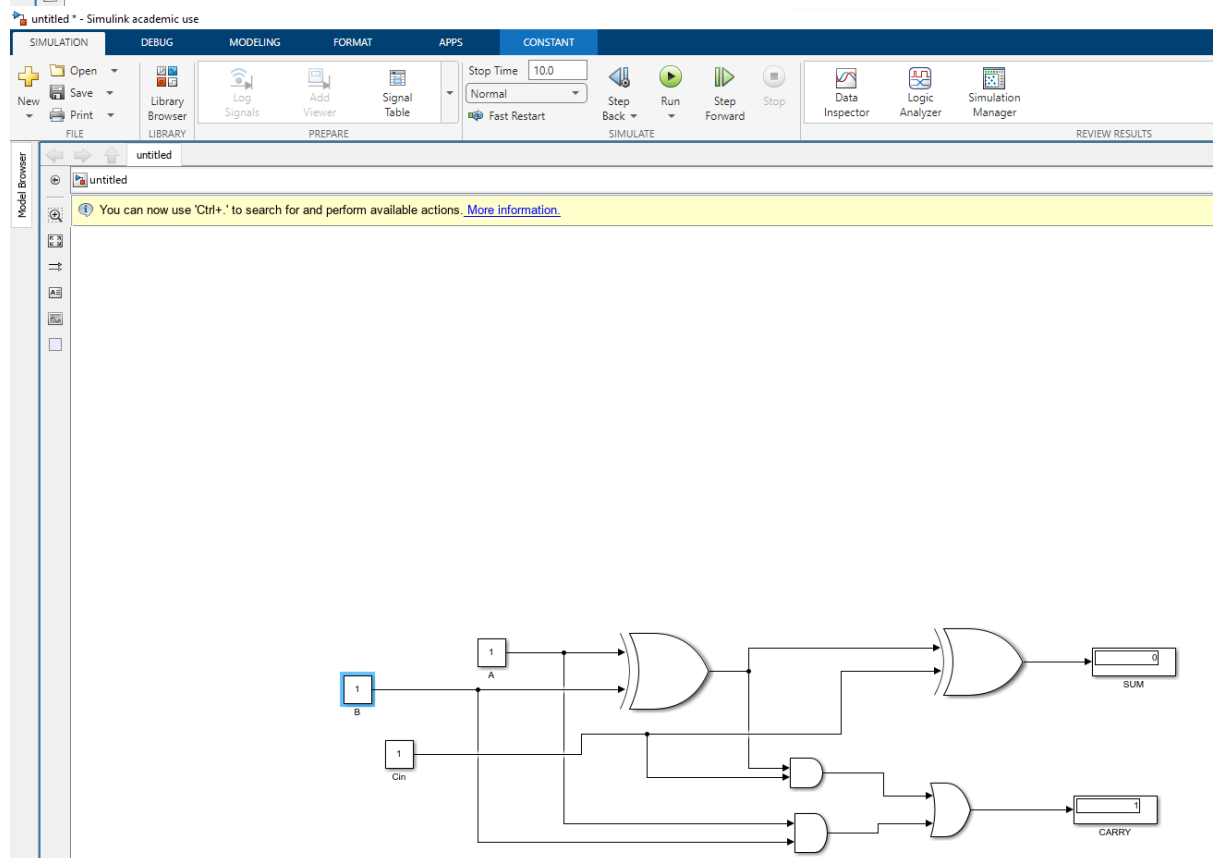
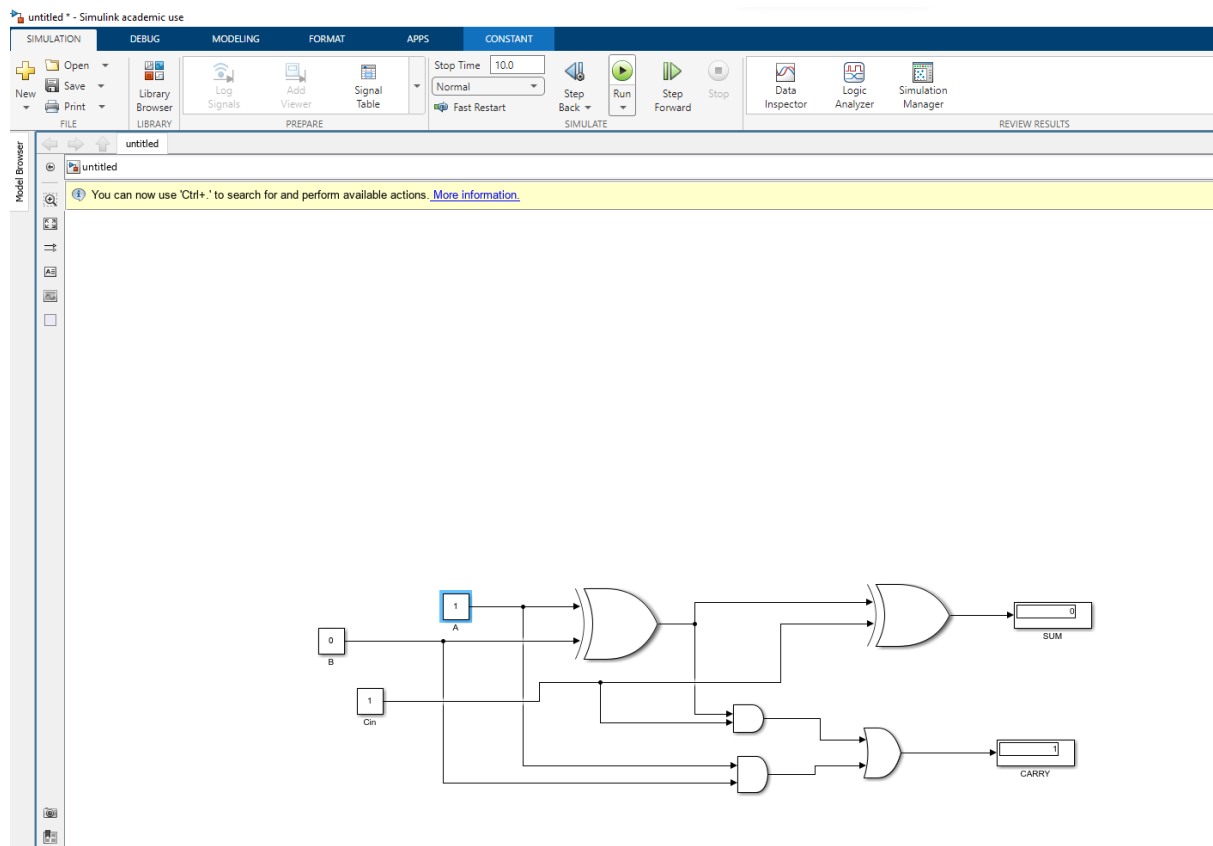
 Subsystem.v Search

```
37  input  B;
38  input  C_in;
39  output SUM;
40  output CARRY;
41
42
43  wire XOR_out1;
44  wire XOR1_out1;
45  wire AND_out1;
46  wire AND1_out1;
47  wire OR_out1;
48
49
50  assign XOR_out1 = A ^ B;
51
52
53
54  assign XOR1_out1 = XOR_out1 ^ C_in;
55
56
57
58  assign SUM = XOR1_out1;
59
60  assign AND_out1 = XOR_out1 & C_in;
61
62
63
64  assign AND1_out1 = A & B;
65
66
67
68  assign OR_out1 = AND_out1 | AND1_out1;
69
70
71
72  assign CARRY = OR_out1;
73
74  endmodule // Subsystem
75
76
```

Activate Windows
Go to Settings to activate Windows.







. Generate VHDL code for Fulladder.

Solution:-

The image displays two screenshots of the MATLAB/Simulink HDL Code Generator interface, showing the process of generating VHDL code for a Full Adder.

Top Screenshot: The interface shows the "HDL CODE" tab. The "Subsystem" block is selected in the Model Browser. The HDL Code window displays the generated VHDL code, which includes the entity definition for the Full Adder, the port declarations (A, B, C, SUM, CARRY), and the architecture body. The code is highlighted in yellow, indicating it is the current selection.

Bottom Screenshot: The interface shows the "HDL CODE" tab. The "Subsystem" block is selected in the Model Browser. The HDL Code window displays the generated VHDL code, which includes the entity definition for the Full Adder, the port declarations (A, B, C, SUM, CARRY), and the architecture body. The code is highlighted in yellow, indicating it is the current selection.

FULLADDER/Subsystem - Simulink

Model Browser

Subsystem

Diagram: A logic diagram of a 1-bit full adder. It has three inputs: A (1-bit), B (1-bit), and C_in (1-bit). It has two outputs: SUM (1-bit) and CARRY (1-bit). The logic is implemented using XOR and AND gates.

Diagnostic Viewer

For the block 'FULLADDER/Subsystem/XOR1':
The block does not support mixing double, single, half, and/or non-real types at the ports. Use either all the same floating point type or no floating point types.
Component: HDLCoder | Category: HDL error

HDL Code

```

17 USE IEEE.std_logic_1164.ALL;
18 USE IEEE.numeric_std.ALL;
19
20 ENTITY Subsystem IS
21   PORT(
22     A : IN
23     B : IN
24     C_in : IN
25     SUM : OUT
26     CARRY : OUT
27   );
28 END Subsystem;
29
30 ARCHITECTURE rtl OF Subsystem IS
31   -- Signals
32   SIGNAL XOR_out1 : std_logic;
33   SIGNAL XOR1_out1 : std_logic;
34   SIGNAL AND_out1 : std_logic;
35   SIGNAL AND1_out1 : std_logic;
36   SIGNAL OR_out1 : std_logic;
37
38 BEGIN
39   XOR_out1 <= A XOR B;
40   XOR1_out1 <= XOR_out1 XOR C_in;
41   AND_out1 <= XOR_out1 AND C_in;
42   AND1_out1 <= A AND B;
43   OR_out1 <= AND_out1 OR AND1_out1;
44   SUM <= XOR1_out1;
45   CARRY <= OR_out1;
46 END rtl;

```

FULLADDER/Subsystem - Simulink

Model Browser

Subsystem

Diagram: A logic diagram of a 1-bit full adder. It has three inputs: A (1-bit), B (1-bit), and C_in (1-bit). It has two outputs: SUM (1-bit) and CARRY (1-bit). The logic is implemented using XOR and AND gates.

Diagnostic Viewer

For the block 'FULLADDER/Subsystem/XOR1':
The block does not support mixing double, single, half, and/or non-real types at the ports. Use either all the same floating point type or no floating point types.
Component: HDLCoder | Category: HDL error

HDL Code

```

17 USE IEEE.numeric_std.ALL;
18
19 ENTITY Subsystem IS
20   PORT(
21     A : IN
22     B : IN
23     C_in : IN
24     SUM : OUT
25     CARRY : OUT
26   );
27 END Subsystem;
28
29 ARCHITECTURE rtl OF Subsystem IS
30   -- Signals
31   SIGNAL XOR_out1 : std_logic;
32   SIGNAL XOR1_out1 : std_logic;
33   SIGNAL AND_out1 : std_logic;
34   SIGNAL AND1_out1 : std_logic;
35   SIGNAL OR_out1 : std_logic;
36
37 BEGIN
38   XOR_out1 <= A XOR B;
39   XOR1_out1 <= XOR_out1 XOR C_in;
40   AND_out1 <= XOR_out1 AND C_in;
41   AND1_out1 <= A AND B;
42   OR_out1 <= AND_out1 OR AND1_out1;
43   SUM <= XOR1_out1;
44   CARRY <= OR_out1;
45 END rtl;

```

FULLADDER/Subsystem - Simulink

Model Browser

Subsystem

Diagram: A logic diagram of a 1-bit full adder. It has three inputs: A (1-bit), B (1-bit), and C_in (1-bit). It has two outputs: SUM (1-bit) and CARRY (1-bit). The logic is implemented using XOR and AND gates.

Diagnostic Viewer

For the block 'FULLADDER/Subsystem/XOR1':
The block does not support mixing double, single, half, and/or non-real types at the ports. Use either all the same floating point type or no floating point types.
Component: HDLCoder | Category: HDL error

HDL Code

```

17 USE IEEE.numeric_std.ALL;
18
19 ENTITY Subsystem IS
20   PORT(
21     A : IN std_logic;
22     B : IN std_logic;
23     C_in : IN std_logic;
24     SUM : OUT std_logic;
25     CARRY : OUT std_logic;
26   );
27 END Subsystem;
28
29 ARCHITECTURE rtl OF Subsystem IS
30   -- Signals
31   SIGNAL XOR_out1 : std_logic;
32   SIGNAL XOR1_out1 : std_logic;
33   SIGNAL AND_out1 : std_logic;
34   SIGNAL AND1_out1 : std_logic;
35   SIGNAL OR_out1 : std_logic;
36
37 BEGIN
38   XOR_out1 <= A XOR B;
39   XOR1_out1 <= XOR_out1 XOR C_in;
40   AND_out1 <= XOR_out1 AND C_in;
41   AND1_out1 <= A AND B;
42   OR_out1 <= AND_out1 OR AND1_out1;
43   SUM <= XOR1_out1;
44   CARRY <= OR_out1;
45 END rtl;

```

→VHDL CODE GENERATED BY HDL CODER OF MATLAB SIMULINK:-

```
-- -----  
--  
-- File Name:hdlsrc/FULLADDER/Subsystem.vhd  
-- Created: 2023-03-14 09:32:36  
--  
-- Generated by MATLAB 9.13 and HDL Coder 4.0  
--  
--  
-- -----  
-- Rate and Clocking Details  
-- -----  
-- Model base rate: 0  
-- Target subsystem base rate: inf  
--  
-- -----  
-- -----  
--  
-- Module: Subsystem  
-- Source Path: FULLADDER/Subsystem  
-- Hierarchy Level: 0  
--  
-- -----  
  
LIBRARY IEEE;  
  
USE IEEE.std_logic_1164.ALL;
```

```
USE IEEE.numeric_std.ALL;
```

```
ENTITY Subsystem IS
```

```
    PORT( A           : IN  std_logic;
          B           : IN  std_logic;
          C_in        : IN  std_logic;
          SUM          : OUT std_logic;
          CARRY        : OUT std_logic
    );
```

```
END Subsystem;
```

```
ARCHITECTURE rtl OF Subsystem IS
```

```
-- Signals
```

```
SIGNAL XOR_out1      : std_logic;
SIGNAL XOR1_out1     : std_logic;
SIGNAL AND_out1      : std_logic;
SIGNAL AND1_out1     : std_logic;
SIGNAL OR_out1       : std_logic;
```

```
BEGIN
```

```
XOR_out1 <= A XOR B;
XOR1_out1 <= XOR_out1 XOR C_in;
AND_out1 <= XOR_out1 AND C_in;
AND1_out1 <= A AND B;
OR_out1 <= AND_out1 OR AND1_out1;
SUM <= XOR1_out1;
CARRY <= OR_out1;
```

```
END rtl;
```

1 / 1 < > X

39 

Activate Windows

Go to Settings to activate Windows.

HDL Code

Subsystem.vhd (0) Search

Highlighting: lines modified in last build 1 / 1

```
25 LIBRARY IEEE;
26 USE IEEE.std_logic_1164.ALL;
27 USE IEEE.numeric_std.ALL;
28
29 ENTITY Subsystem IS
30     PORT( A           : IN    std_logic;
31           B           : IN    std_logic;
32           C_in        : IN    std_logic;
33           SUM          : OUT   std_logic;
34           CARRY        : OUT   std_logic
35     );
36 END Subsystem;
37
38
39 ARCHITECTURE rtl OF Subsystem IS
40
41     -- Signals
42     SIGNAL XOR_out1    : std_logic;
43     SIGNAL XOR1_out1   : std_logic;
44     SIGNAL AND_out1    : std_logic;
45     SIGNAL AND1_out1   : std_logic;
46     SIGNAL OR_out1     : std_logic;
47
48 BEGIN
49     XOR_out1 <= A XOR B;
50
51     XOR1_out1 <= XOR_out1 XOR C_in;
52
53     AND_out1 <= XOR_out1 AND C_in;
54
55     AND1_out1 <= A AND B;
56
57     OR_out1 <= AND_out1 OR AND1_out1;
58
59     SUM <= XOR1_out1;
60
61     CARRY <= OR_out1;
62
63 END rtl;
```

Ln 33 Col 37

Activate Windows
Go to Settings to activate Windows.