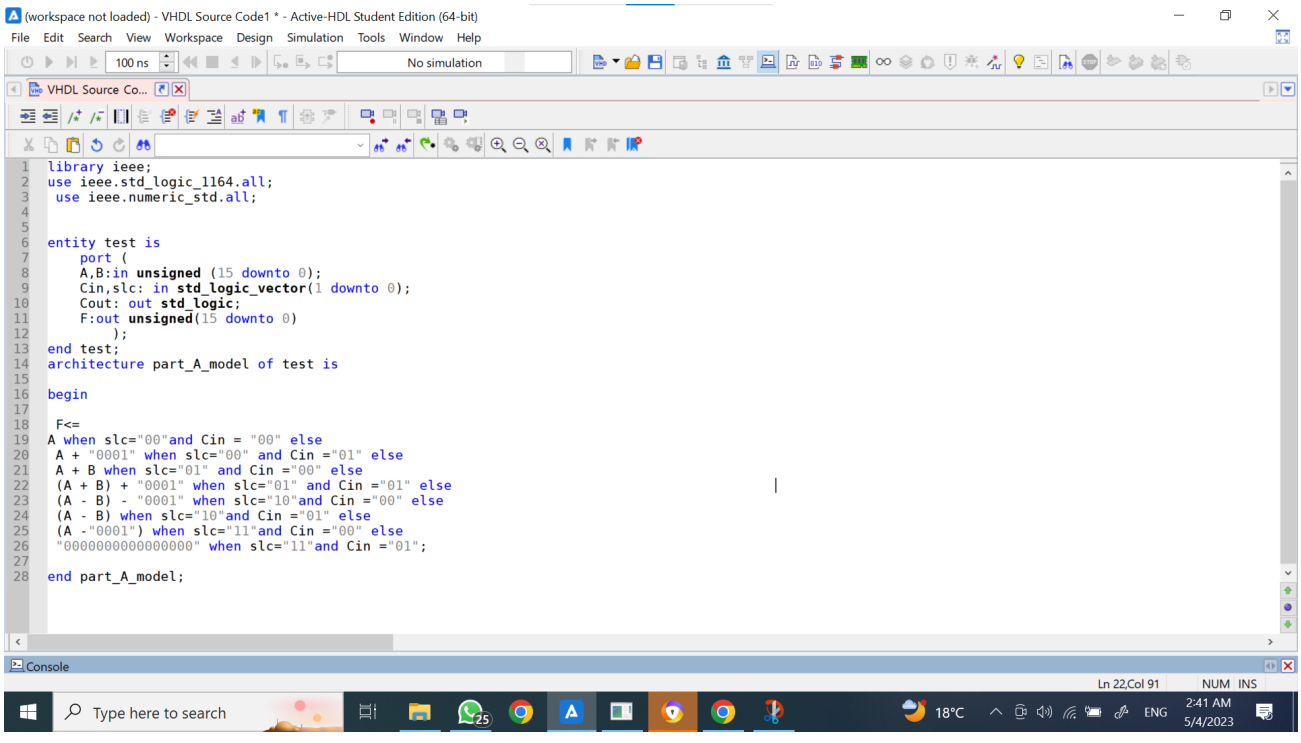
Code part A



The provided code represents a digital circuit described in VHDL (VHSIC Hardware Description Language). Let's break it down step by step:

1. Library Declarations:

- The code begins with three library declarations:
- **ieee.std_logic_1164.all:** This library provides the definitions and operations for working with signals of the type **std_logic**, which represents digital signals with multiple logic levels (e.g., '0', '1', 'Z', 'X', 'U', etc.).
- **ieee.numeric_std.all:** This library includes the definitions and operations for working with numeric types such as **unsigned** and arithmetic operations on them.

2.Entity Declaration:

- The **entity** section defines the interface of the module, which is named "test".
- The module has the following ports:
- **A** and **B**: Unsigned input signals of size 16 bits (ranging from 15 to 0).
- **Cin** and **slc**: 2-bit input signals represented by **std_logic_vector**.
- **Cout**: An output signal represented by **std_logic**.
- **F**: An output signal of type **unsigned** with a size of 16 bits.

3.Architecture Definition:

- The **architecture** section defines the behavior of the module and is named "part_A_model".
- The **begin** keyword marks the start of the architecture body.

4.Signal Assignment:

- The **F** signal is assigned a value based on conditional statements using the **when** keyword.
- The conditional statements are evaluated sequentially, and the first matching condition determines the value assigned to **F**.

Here's a breakdown of the conditional statements:

- If **slc** is "00" and **Cin** is "00", **F** is assigned the value of signal **A**.
- If **slc** is "00" and **Cin** is "01", **F** is assigned the value of signal **A** incremented by 1.
- If **slc** is "01" and **Cin** is "00", **F** is assigned the value of the sum of signals **A** and **B**.
- If **slc** is "01" and **Cin** is "01", **F** is assigned the value of the sum of signals **A, B,** and **1**.
- If **slc** is "10" and **Cin** is "00", **F** is assigned the value of the difference between signals **A** and **B**, decreased by 1.
- If **slc** is "10" and **Cin** is "01", **F** is assigned the value of the difference between signals **A** and **B**.
- If **slc** is "11" and **Cin** is "00", **F** is assigned the value of the difference between signal **A** and 1.
- If **slc** is "11" and **Cin** is "01", **F** is assigned the value of 0 (16-bit zero value).

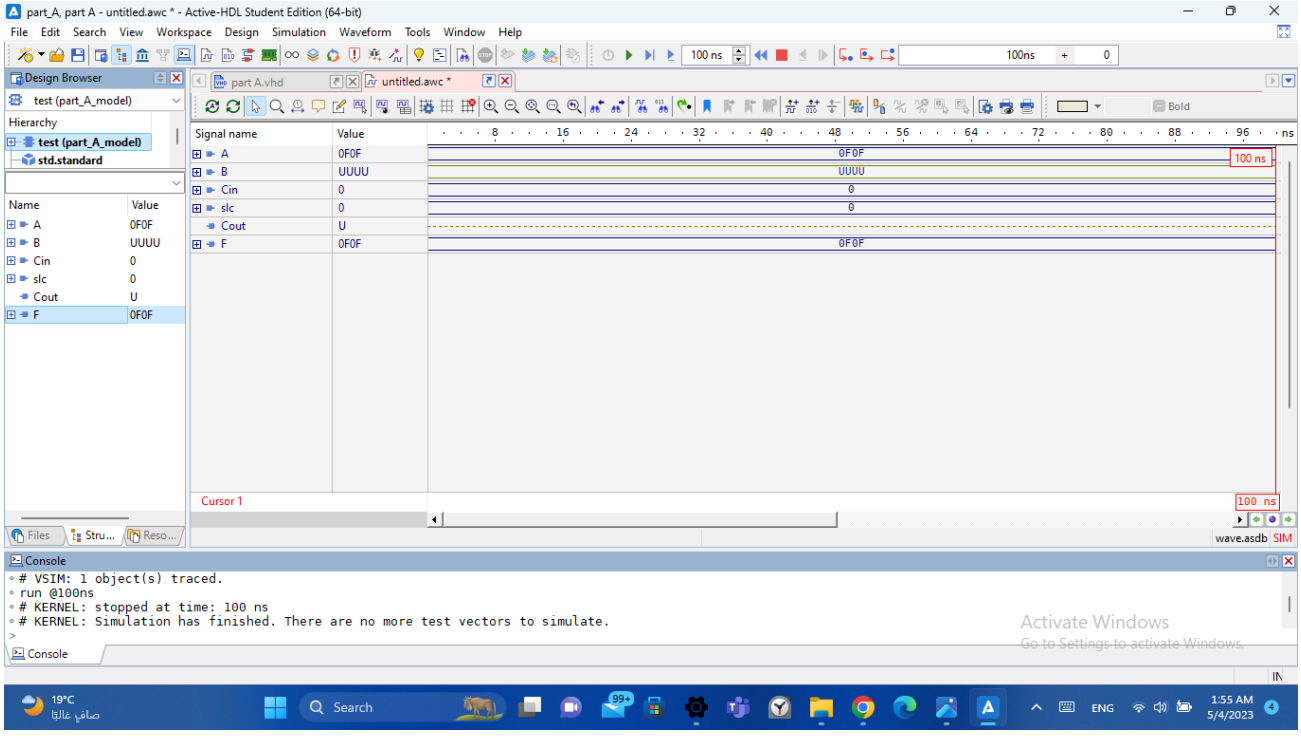
5.End of Architecture:

- The **end part_A_model**; statement marks the end of the architecture.

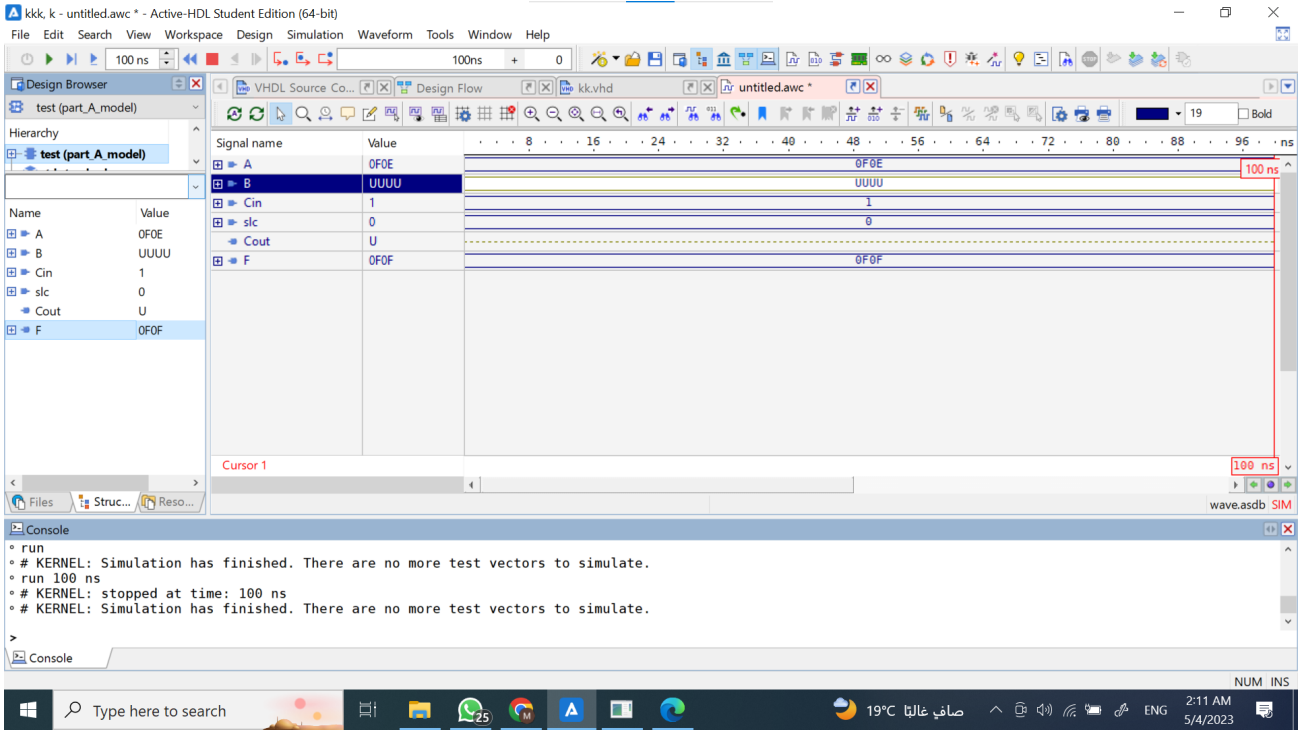
In summary, the code describes a VHDL module named "test" that performs different operations based on the values of input signals **A, B, Cin,** and **slc.** The output signal **F** is assigned different values according to specific conditions.

Waveform

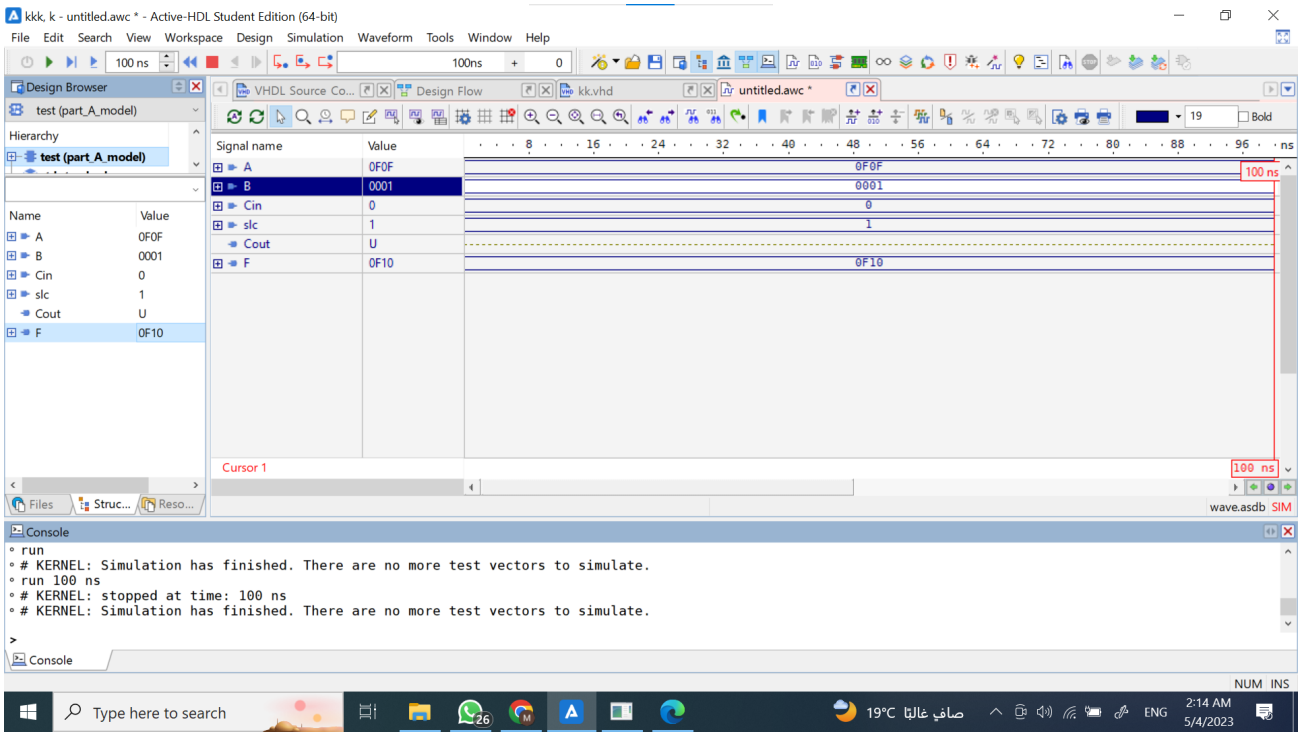
1.F = A



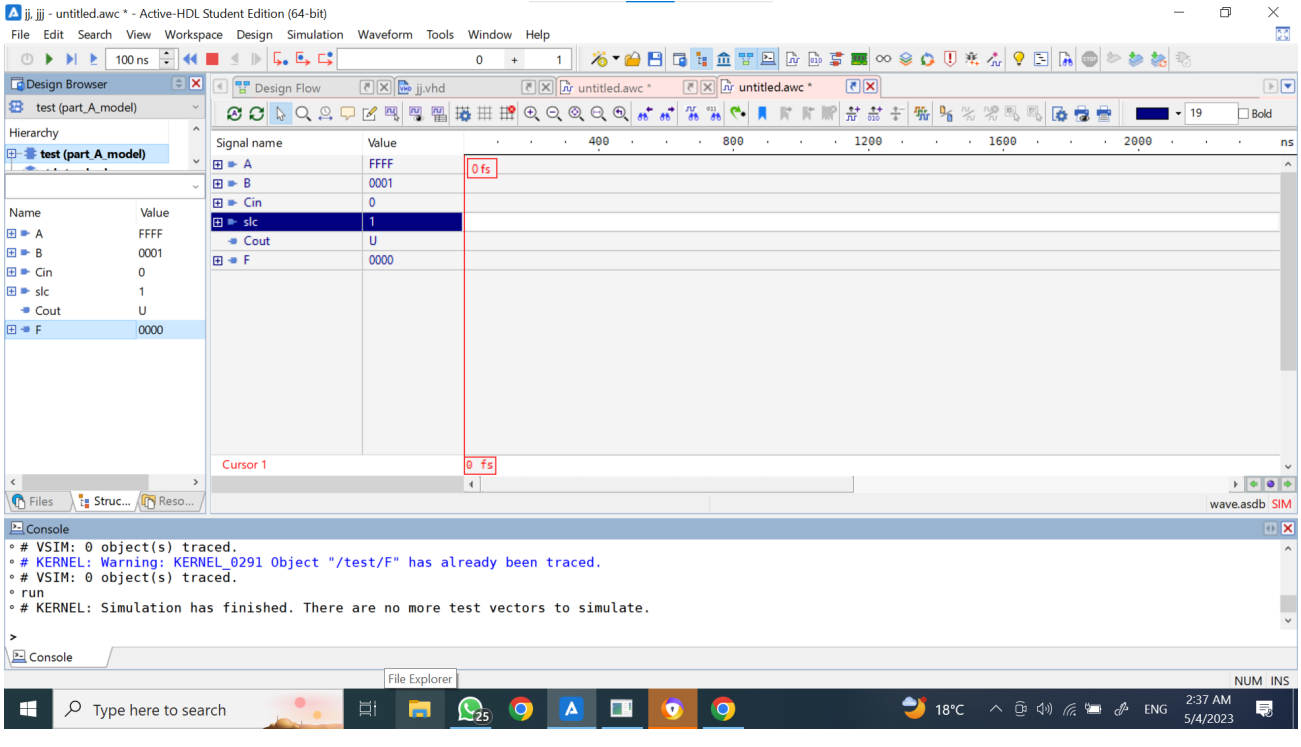
2.F = A + 1



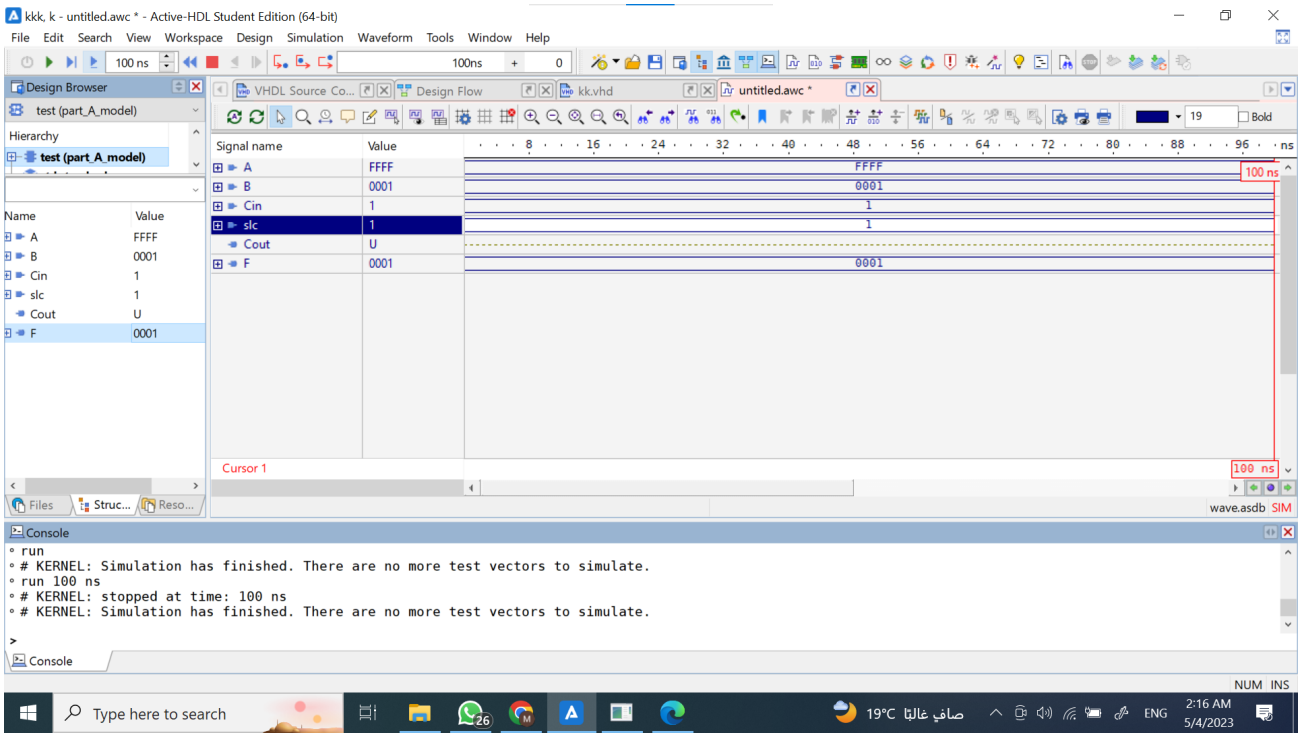
3.F = A + B test 1



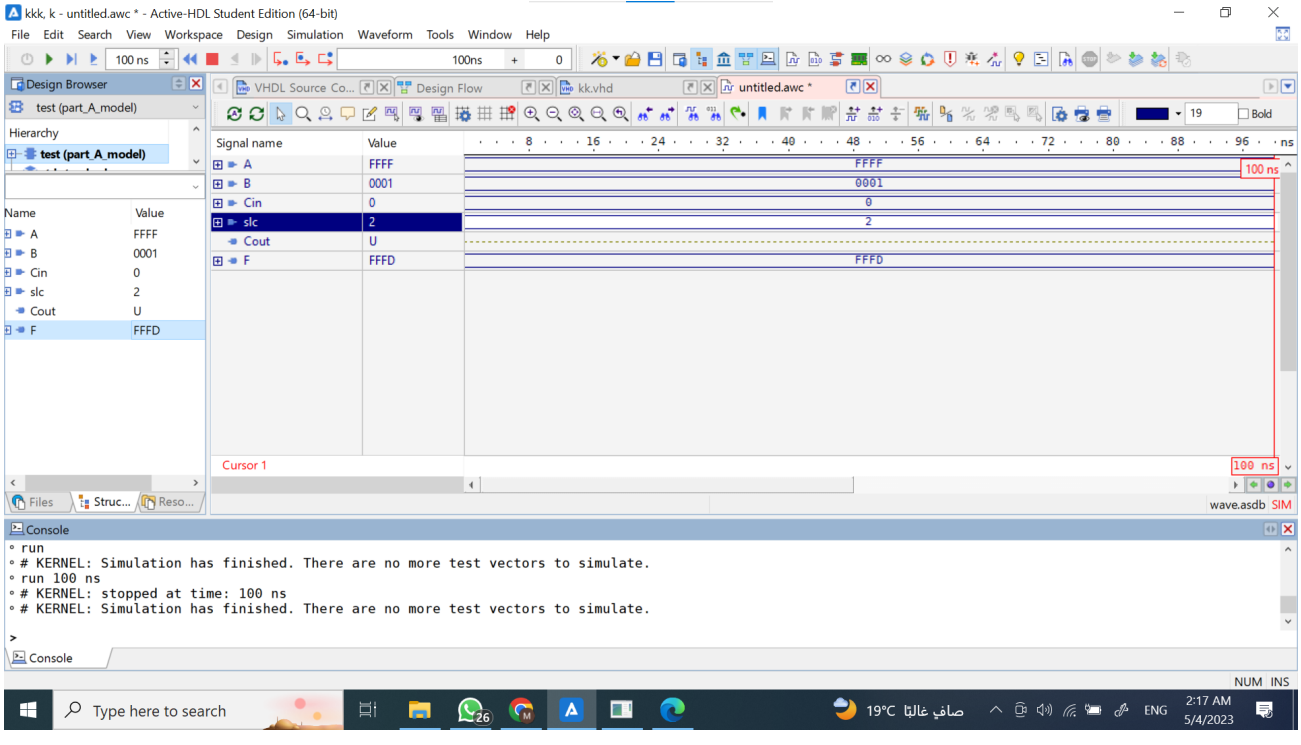
3.F = A + B test 2



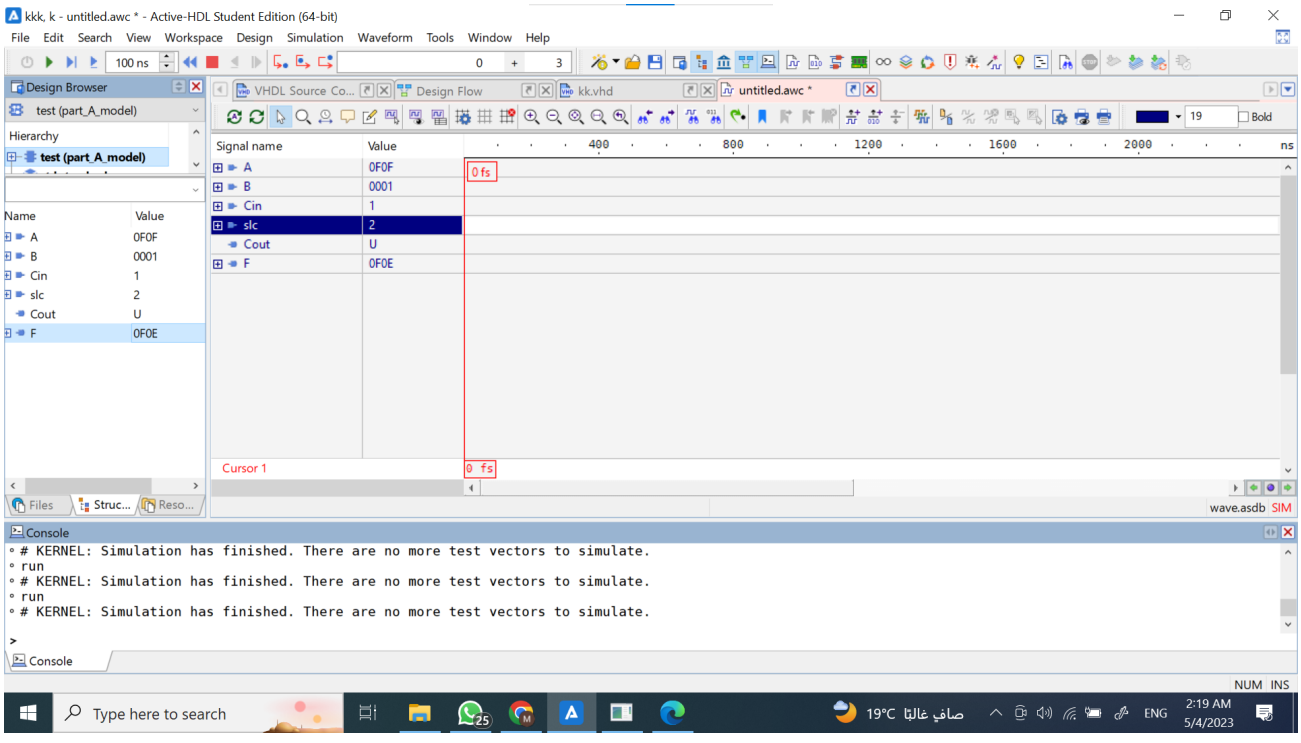
4.F = A + B + 1


kkk, k - untitled.awc * - Active-HDL Student Edition (64-bit)
File Edit Search View Workspace Design Simulation Waveform Tools Window Help
Design Browser
test (part_A_model)
Hierarchy
test (part_A_model)
Name Value
A FFFF
B 0001
Cin 1
slc 1
Cout U
F 0001
Signal name Value
A FFFF
B 0001
Cin 1
slc 1
Cout U
F 0001
Cursor 1
100 ns
wave.asdb SIM
Console
run
KERNEL: Simulation has finished. There are no more test vectors to simulate.
run 100 ns
KERNEL: stopped at time: 100 ns
KERNEL: Simulation has finished. There are no more test vectors to simulate.
Type here to search
19°C صافي غالبًا
ENG
2:16 AM
5/4/2023

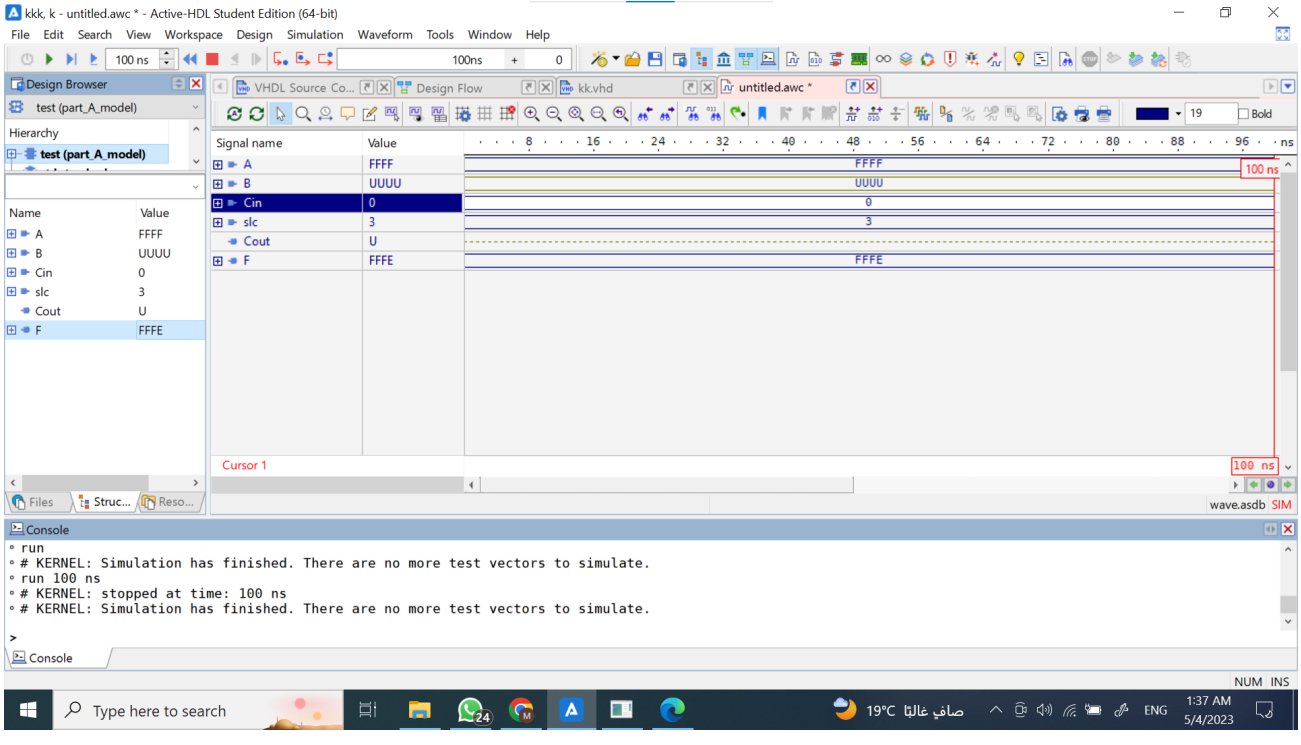
5.F = A - B - 1


kkk, k - untitled.awc * - Active-HDL Student Edition (64-bit)
File Edit Search View Workspace Design Simulation Waveform Tools Window Help
Design Browser
test (part_A_model)
Hierarchy
test (part_A_model)
Name Value
A FFFF
B 0001
Cin 0
slc 2
Cout U
F FFFD
Signal name Value
A FFFF
B 0001
Cin 0
slc 2
Cout U
F FFFD
Cursor 1
100 ns
wave.asdb SIM
Console
run
KERNEL: Simulation has finished. There are no more test vectors to simulate.
run 100 ns
KERNEL: stopped at time: 100 ns
KERNEL: Simulation has finished. There are no more test vectors to simulate.
Type here to search
19°C صافي غالبًا
ENG
2:17 AM
5/4/2023

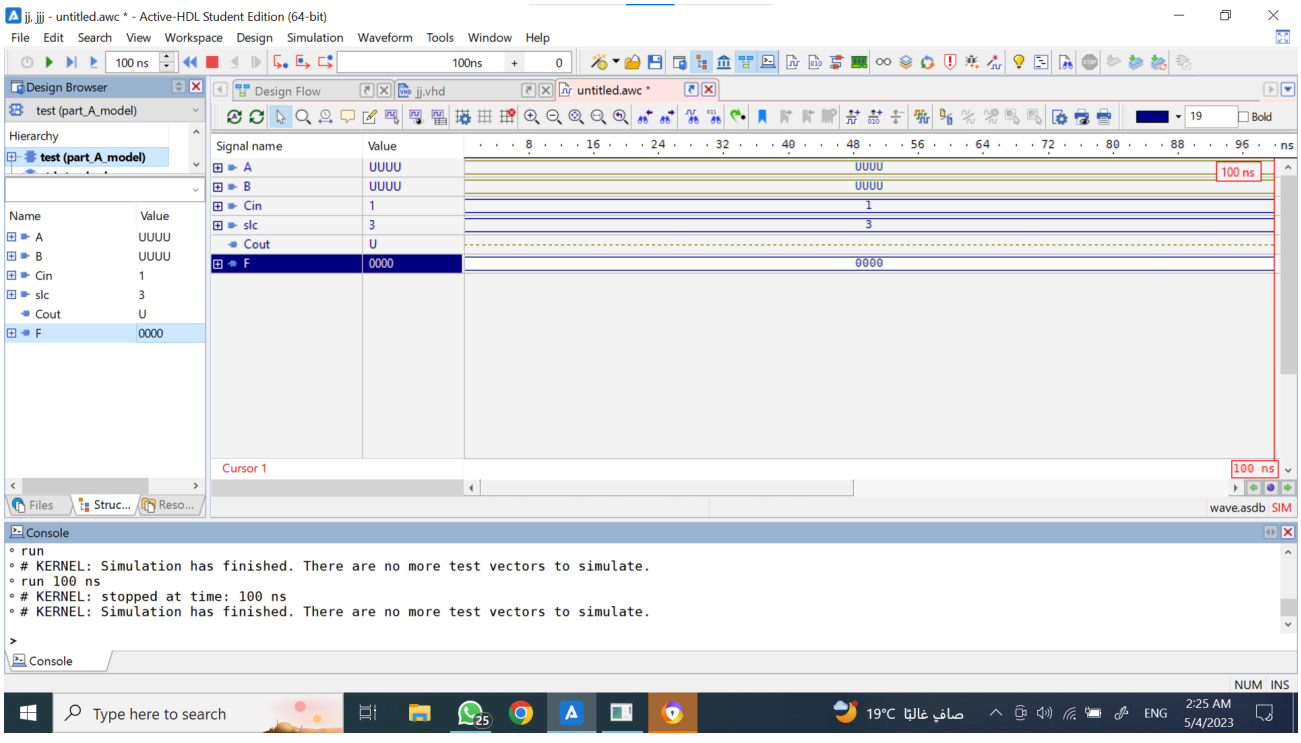
6.F = A - B


kkk, k - untitled.awc * - Active-HDL Student Edition (64-bit)
File Edit Search View Workspace Design Simulation Waveform Tools Window Help
Design Browser
test (part_A_model)
Hierarchy
test (part_A_model)
Name Value
A 0F0F
B 0001
Cin 1
slc 2
Cout U
F 0F0E
Signal name Value
A 0F0F
B 0001
Cin 1
slc 2
Cout U
F 0F0E
0 fs
Cursor 1
0 fs
Console
KERNEL: Simulation has finished. There are no more test vectors to simulate.
run
KERNEL: Simulation has finished. There are no more test vectors to simulate.
run
KERNEL: Simulation has finished. There are no more test vectors to simulate.
Type here to search
19°C
ENG
2:19 AM
5/4/2023

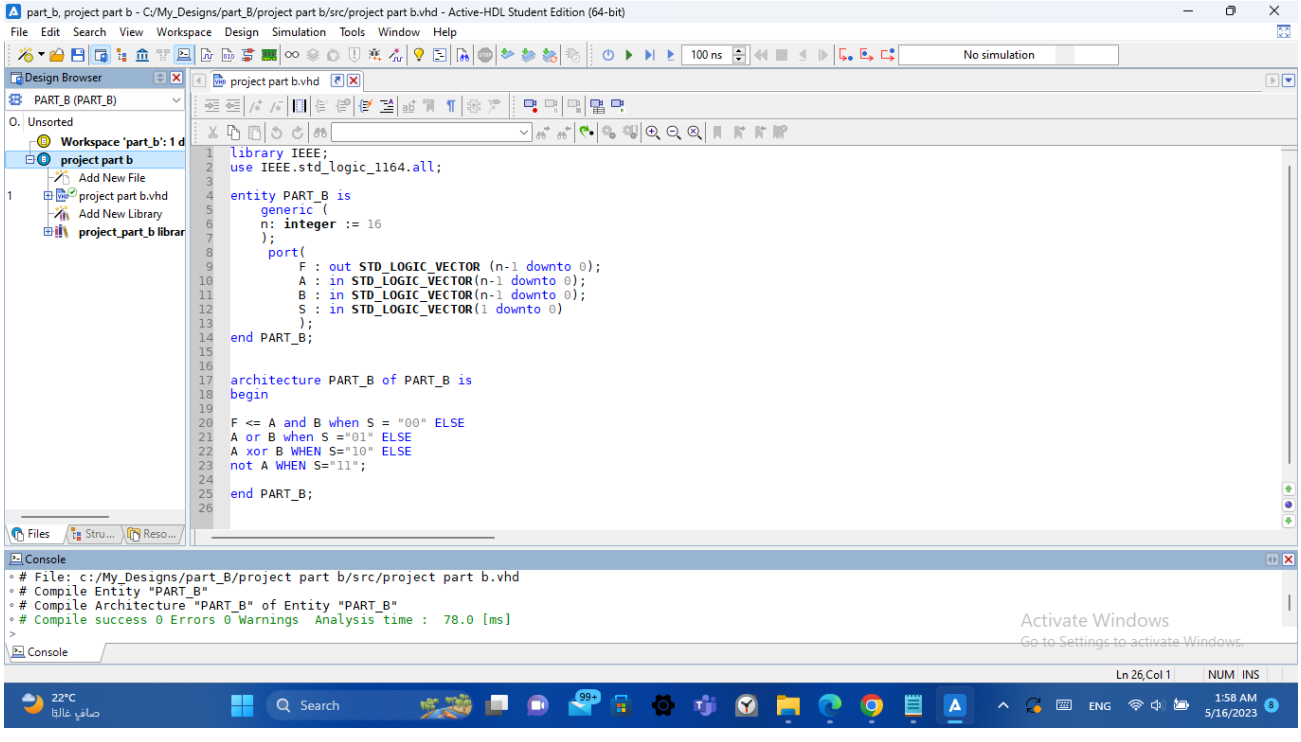
7.F = A - 1


kkk, k - untitled.awc * - Active-HDL Student Edition (64-bit)
File Edit Search View Workspace Design Simulation Waveform Tools Window Help
Design Browser
test (part_A_model)
Hierarchy
test (part_A_model)
Name Value
A FFFF
B UUUU
Cin 0
slc 3
Cout U
F FFFE
Signal name Value
A FFFF
B UUUU
Cin 0
slc 3
Cout U
F FFFE
100 ns
Cursor 1
100 ns
Console
run
KERNEL: Simulation has finished. There are no more test vectors to simulate.
run 100 ns
KERNEL: stopped at time: 100 ns
KERNEL: Simulation has finished. There are no more test vectors to simulate.
Type here to search
19°C
ENG
1:37 AM
5/4/2023

8.F = 0



Code part B



1. Library Declarations:
   - The code begins with the library declaration: **library IEEE**; This allows you to use IEEE standard libraries in your code.
   - The **use IEEE.std_logic_1164.all;** statement imports the **std_logic_1164** package from the IEEE library. This package provides definitions and operations for working with digital signals represented by **std_logic**.
2. Entity Declaration:
   - The **entity** section defines the interface of the module, which is named "PART_B".
   - The module has the following elements:
     - **n**: A generic parameter of type **integer** with a default value of 16.
     - Ports:
       - **F**: An output signal of type **std_logic_vector** with a size of n bits, ranging from n-1 downto 0.
       - **A** and **B:** Input signals of type **std_logic_vector** with a size of n bits.
       - **S**: An input signal of type **std_logic_vector** with a size of 2 bits.
3. Architecture Definition:
   - The **architecture** section defines the behavior of the module, and it is also named "PART_B".
   - The **begin** keyword marks the start of the architecture body.
4. Signal Assignment:
   - The **F** signal is assigned a value based on conditional statements using the **when** keyword.
   - The conditional statements are evaluated sequentially, and the first matching condition determines the value assigned to **F**.

Here's a breakdown of the conditional statements:

- If **S** is "00", **F** is assigned the bitwise logical AND of signals **A** and **B**.
- If **S** is "01", **F** is assigned the bitwise logical OR of signals **A** and **B**.
- If **S** is "10", **F** is assigned the bitwise logical XOR (exclusive OR) of signals **A and B**.
- If **S** is "11", **F** is assigned the bitwise logical negation (NOT) of signal **A**.

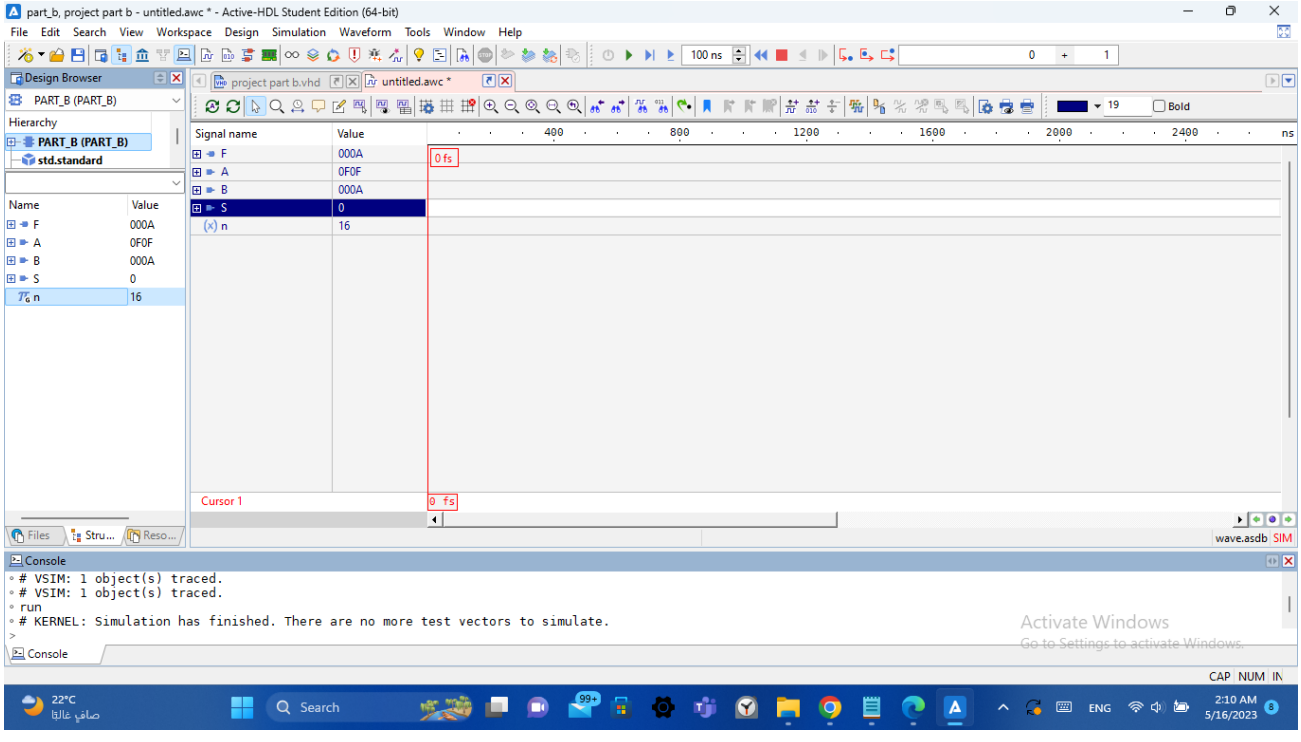
5. End of Architecture:

- The **end PART_B**; statement marks the end of the architecture.

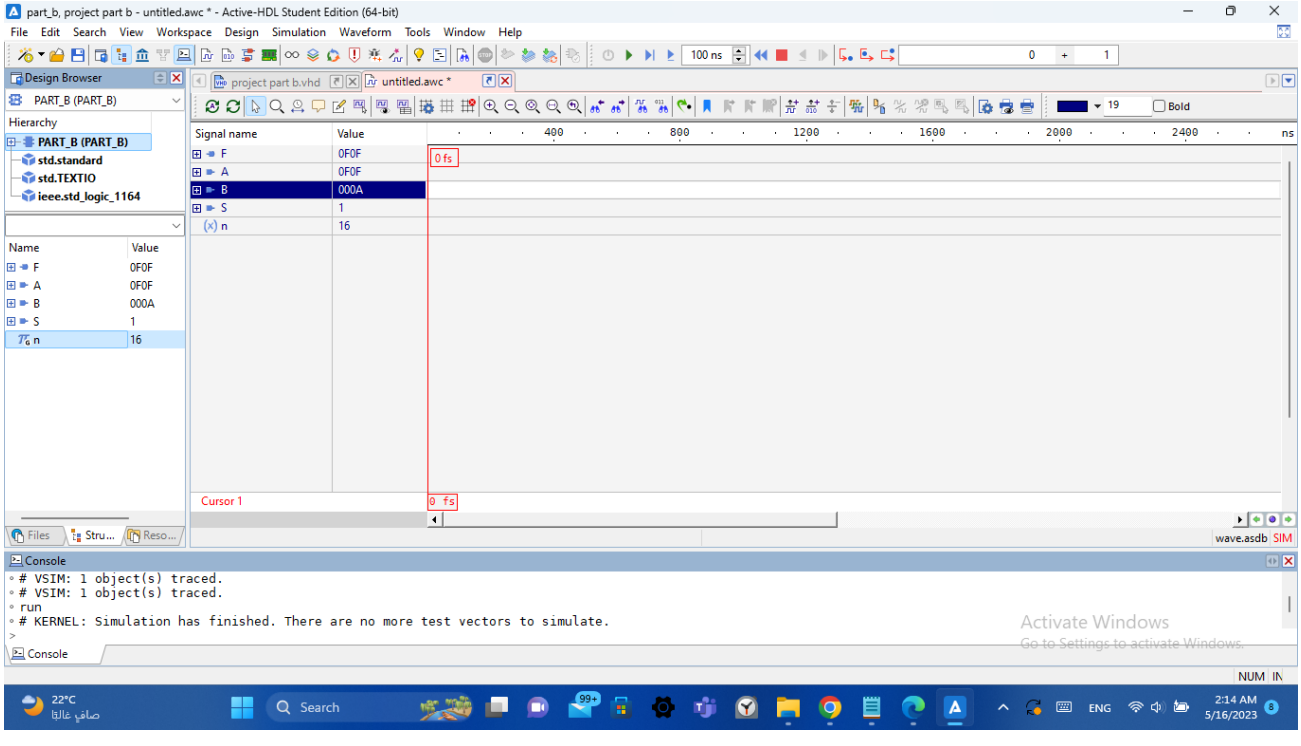
In summary, the code describes a VHDL module named "PART_B" that performs different bitwise logical operations based on the values of input signals A, B, and S. The output signal F is assigned different values according to specific conditions. The specific logical operation applied to A and B depends on the value of S.

Waveform

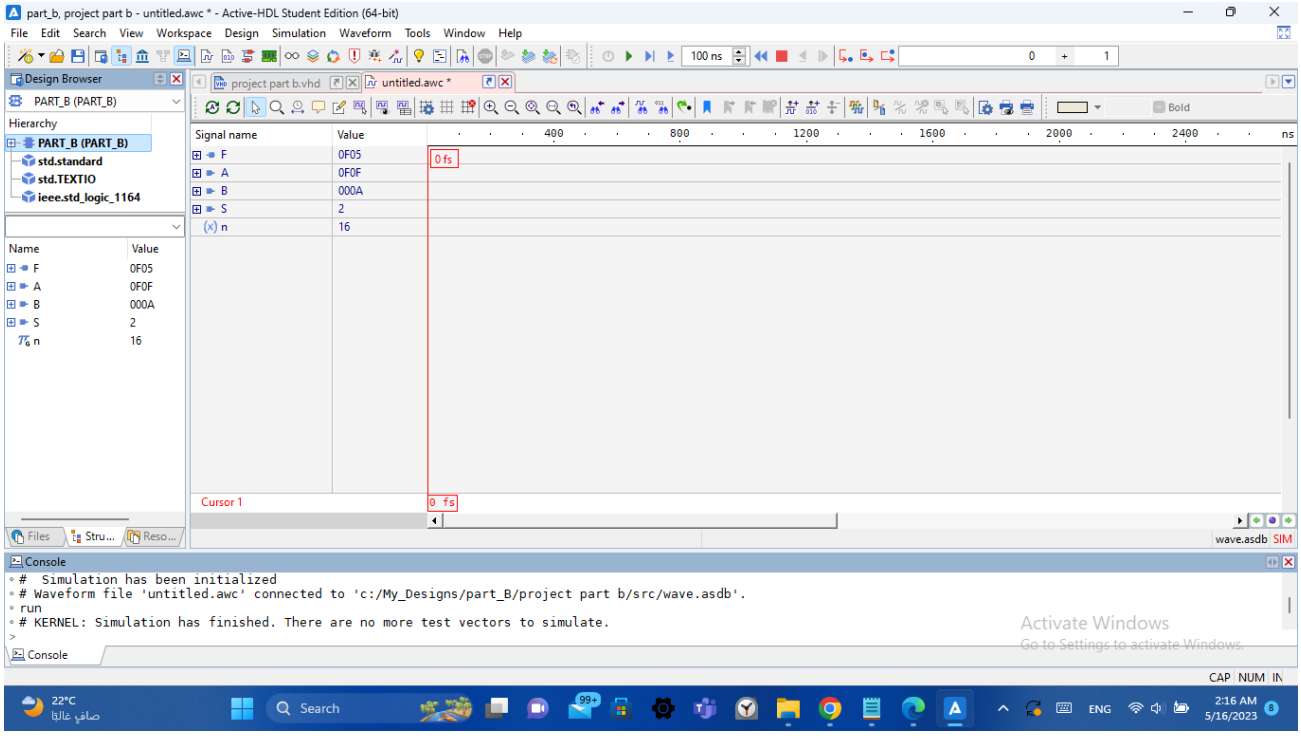
1.F = A and B



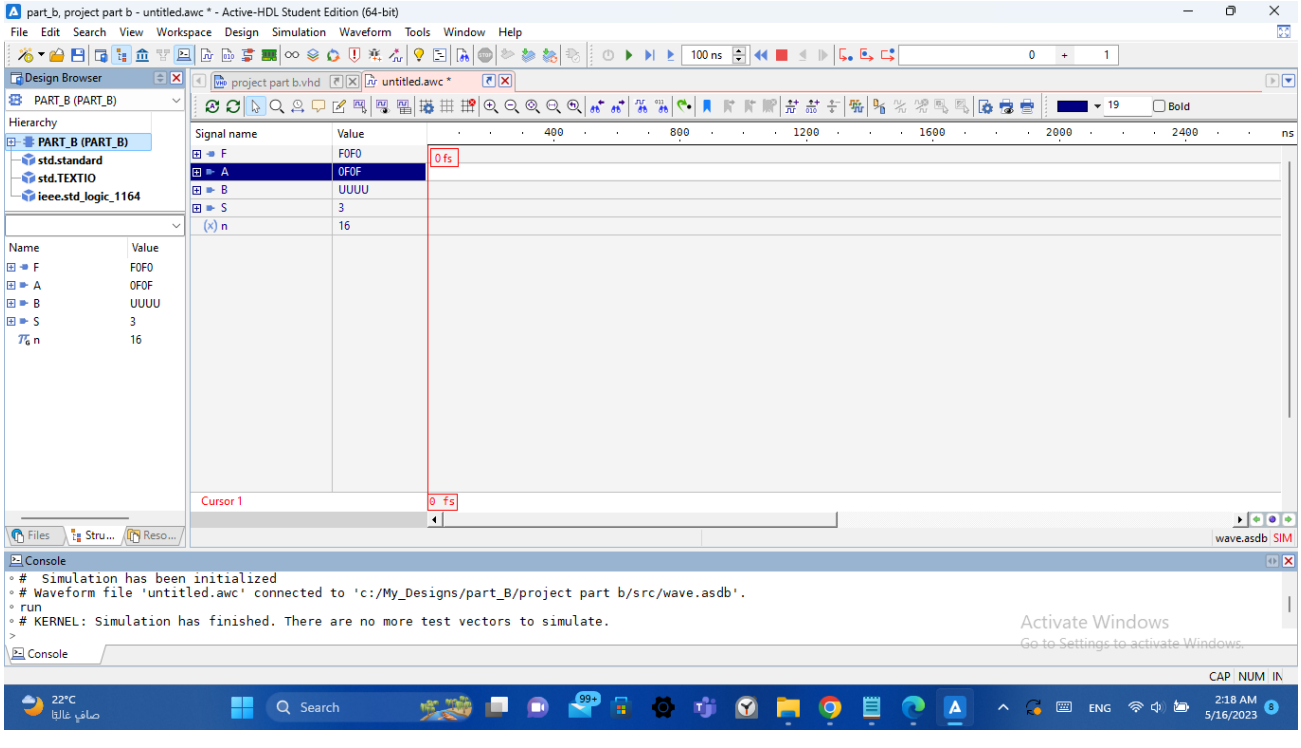
2.F = A or B



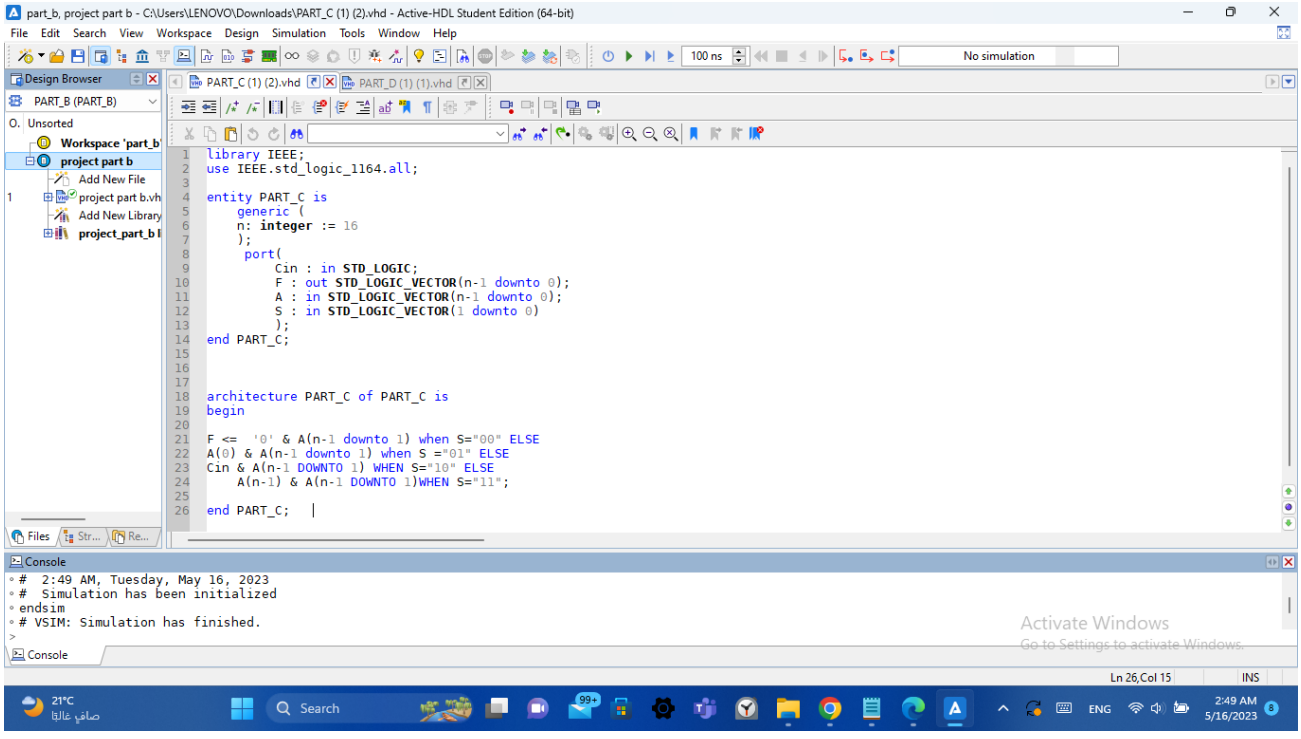
3.F = A xor B



4.F = Not A



Code part C



1. Library Declarations:
   - The code begins with the library declaration: **library IEEE;** This allows you to use IEEE standard libraries in your code.
   - The **use IEEE.std_logic_1164.all**; statement imports the **std_logic_1164** package from the IEEE library. This package provides definitions and operations for working with digital signals represented by **std_logic.**
2. Entity Declaration:
   - The **entity** section defines the interface of the module, which is named "PART_C".
   - The module has the following elements:
     - **n**: A generic parameter of type **integer** with a default value of 16.
     - Ports:
       - **Cin**: An input signal of type **std_logic**.
       - **F**: An output signal of type **std_logic_vector** with a size of n bits, ranging from n-1 downto 0.
       - **A**: An input signal of type **std_logic_vector** with a size of n bits.
       - **S**: An input signal of type **std_logic_vector** with a size of 2 bits.
3. Architecture Definition:
   - The **architecture** section defines the behavior of the module, and it is also named "PART_C".
   - The **begin** keyword marks the start of the architecture body.
4. Signal Assignment:

- The **F** signal is assigned a value based on conditional statements using the **when** keyword.
- The conditional statements are evaluated sequentially, and the first matching condition determines the value assigned to **F**.

Here's a breakdown of the conditional statements:

- If **S** is "00", **F** is assigned the value of concatenating a '0' (zero) with bits **A(n-1)** to **A(1)** (excluding the least significant bit **A(0)**).
- If **S** is "01", **F** is assigned the value of concatenating **A(0)** (least significant bit) with bits **A(n-1)** to **A(1)**.
- If **S** is "10", **F** is assigned the value of concatenating **Cin** with bits **A(n-1)** to **A(1)**.
- If **S** is "11", **F** is assigned the value of concatenating **A(n-1)** (most significant bit) with bits **A(n-1) to A(1)**.

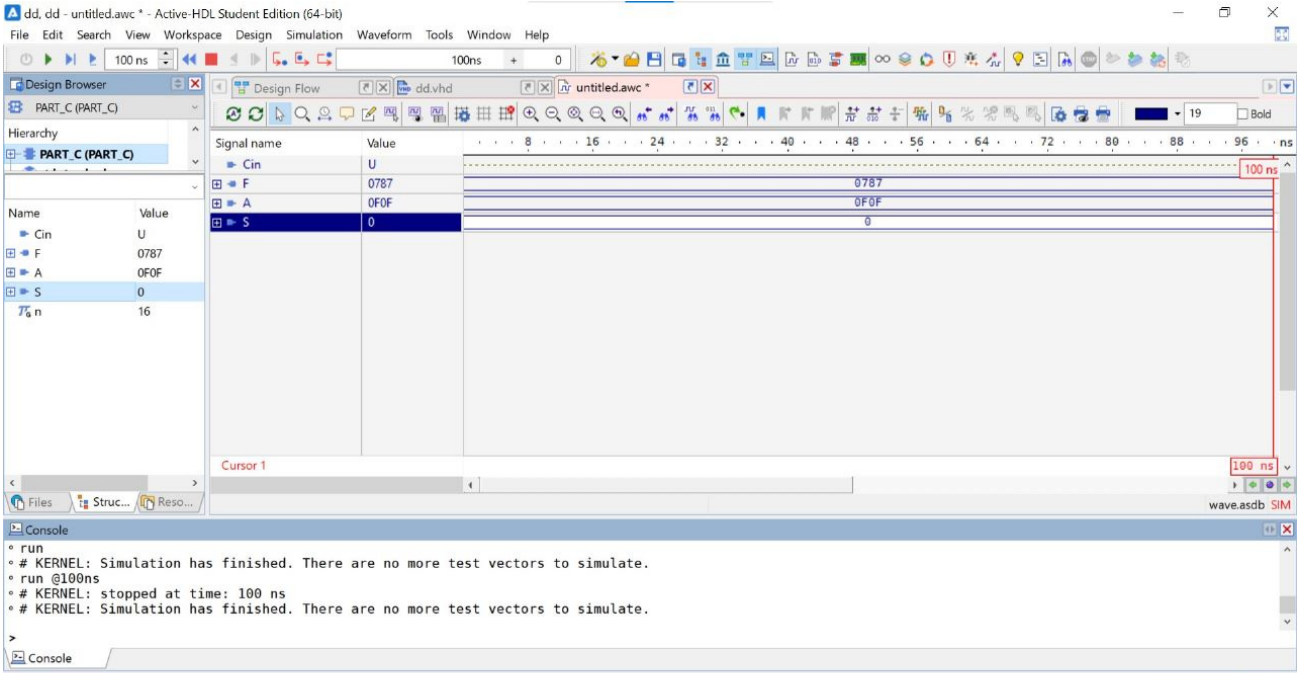
5.End of Architecture:

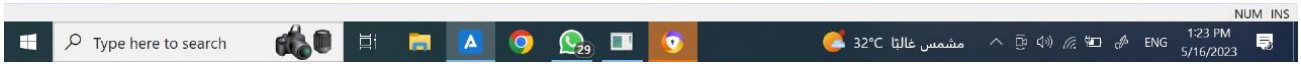
- The **end PART_C**; statement marks the end of the architecture.

In summary, the code describes a VHDL module named "PART_C" that performs different concatenation operations based on the values of input signals **A, Cin,** and **S**. The output signal **F** is assigned different values according to specific conditions. The specific concatenation operation performed depends on the value of **S** and determines which bits from **A** and additional bits (**Cin** or fixed '0'/'1') are included in **F**.

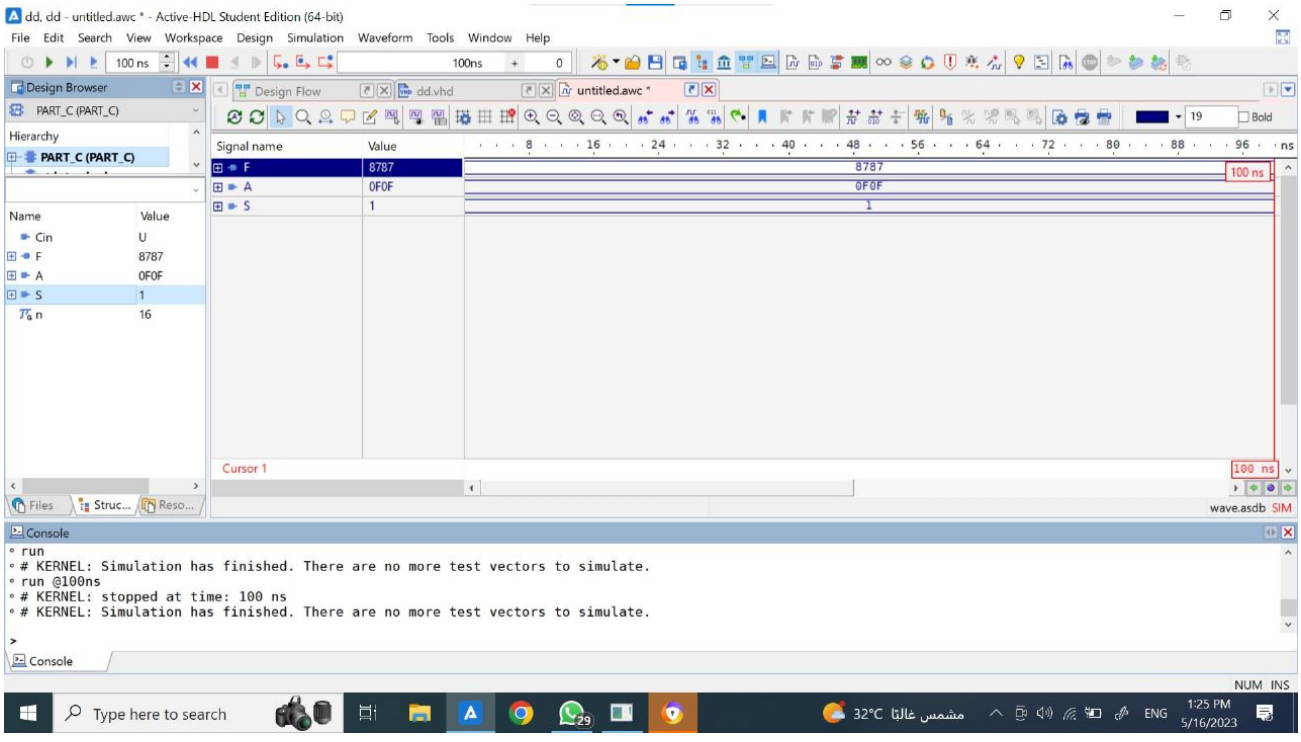
Waveform

1.F = Logic shift right A

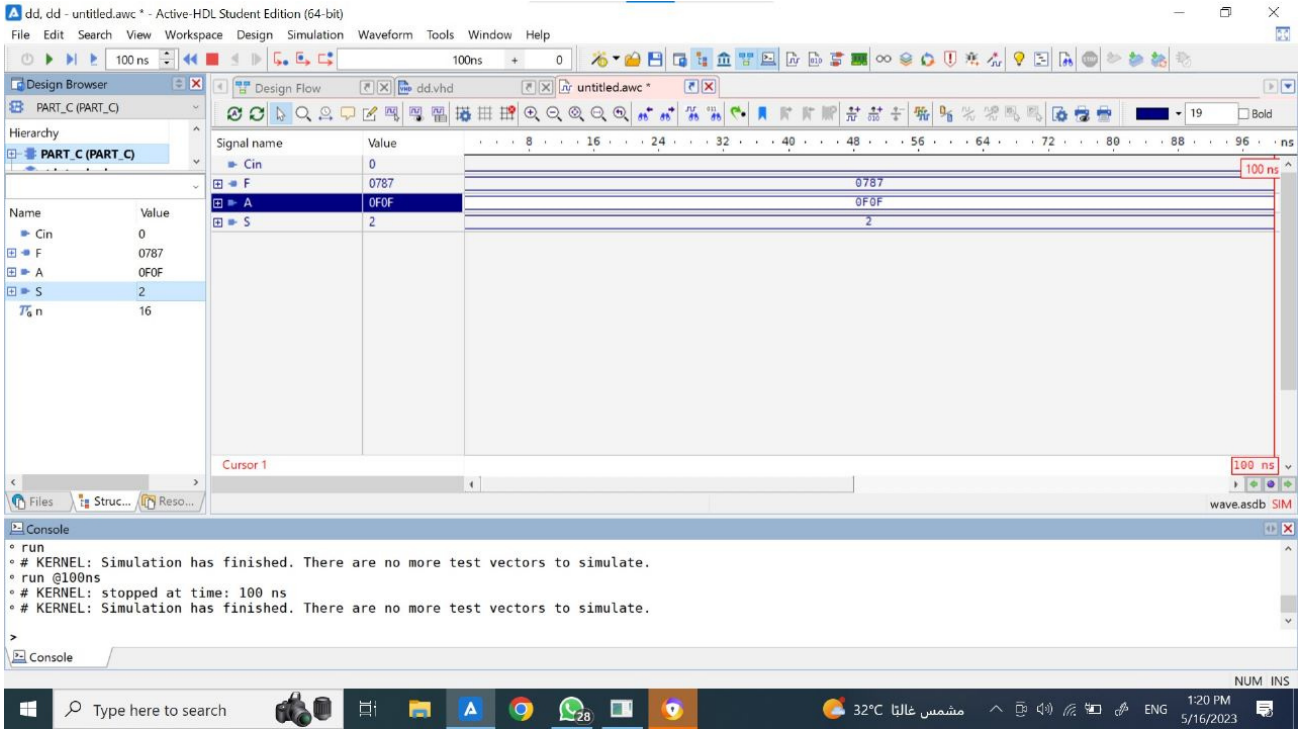




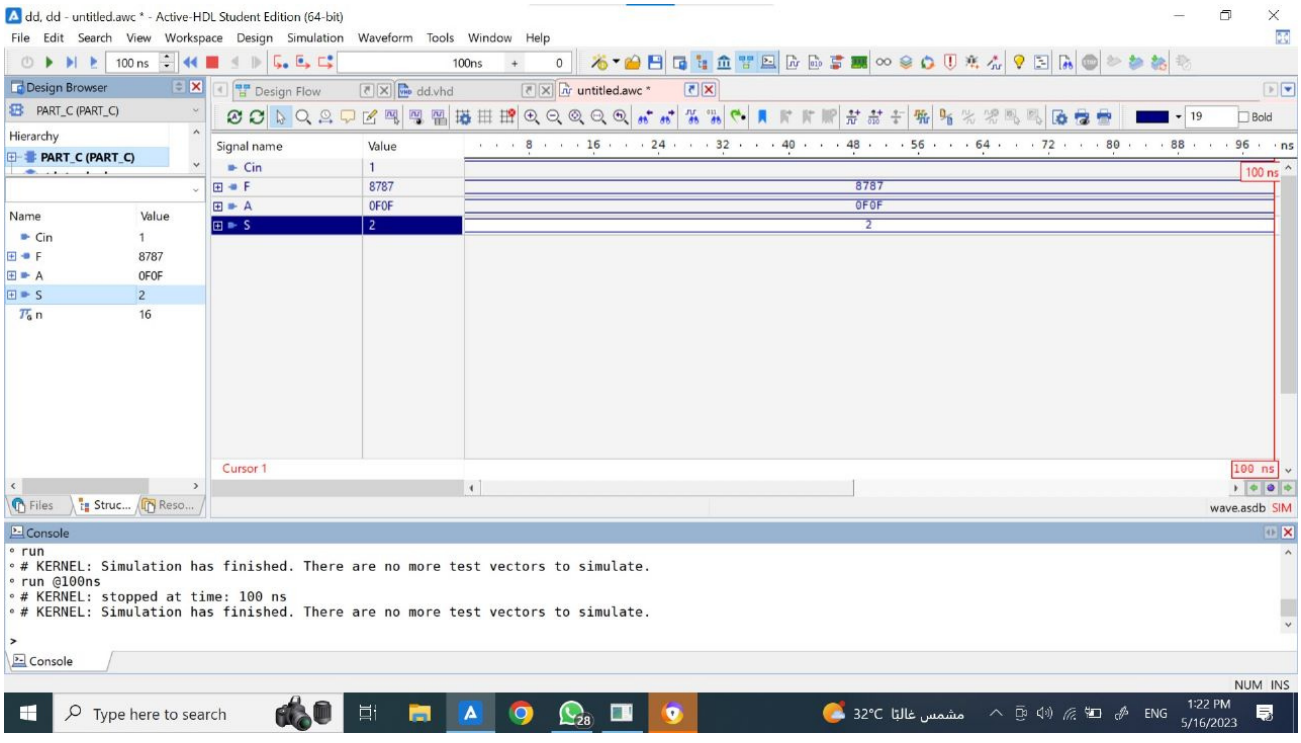
2.F = Logic right A



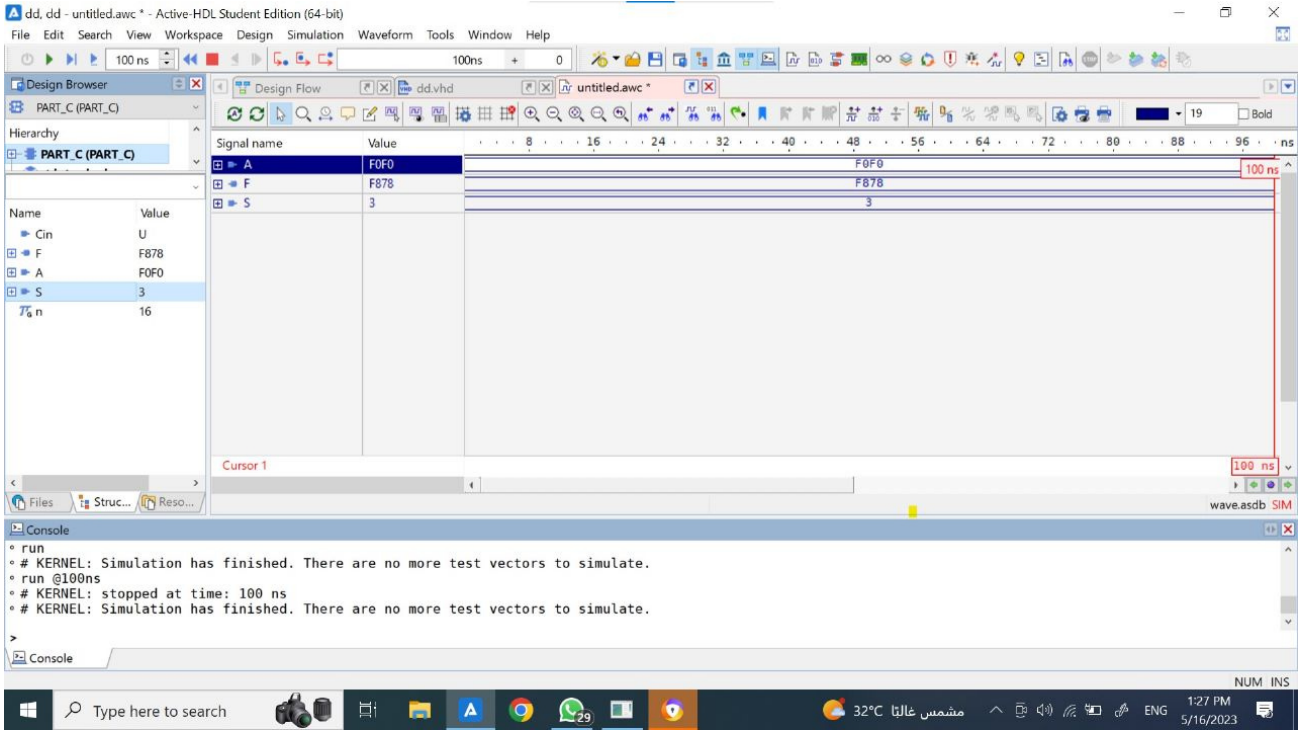
3.F = Rotate right A with carry part 1



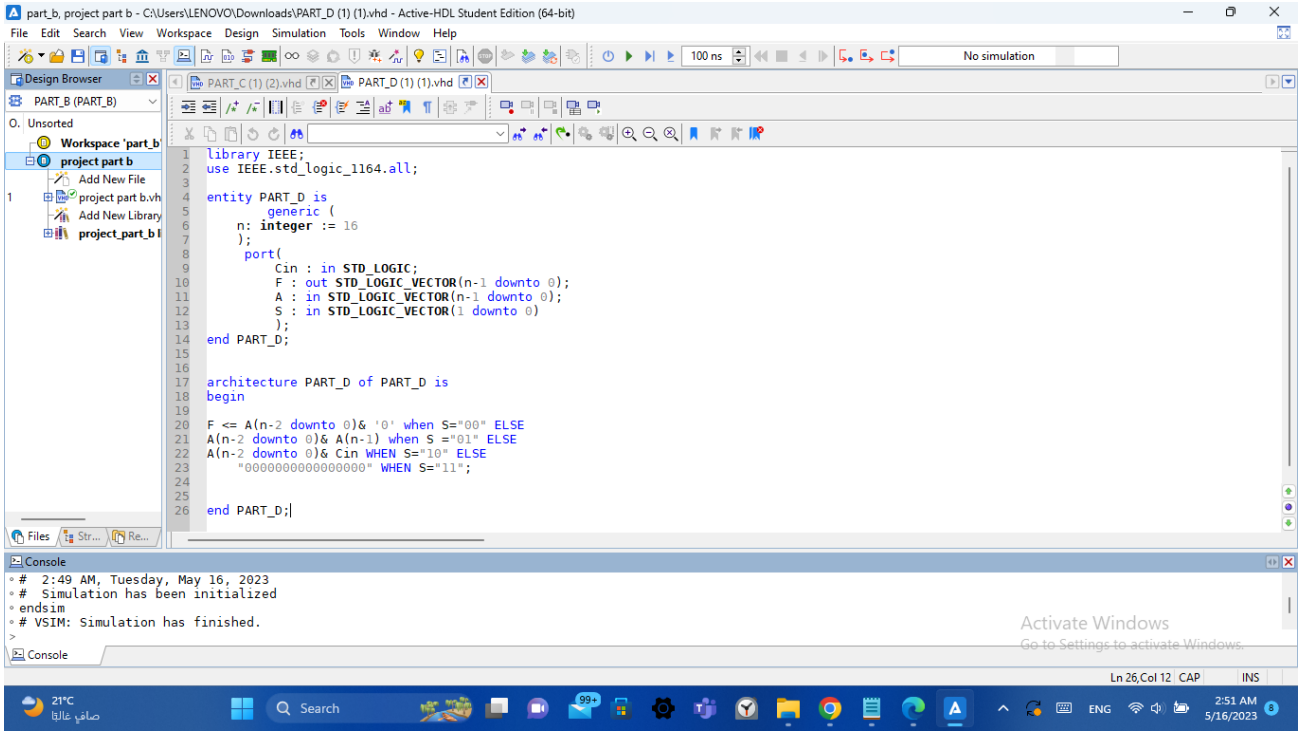
3.F = Rotate right A with carry part 2



4.F = Arithmetic shift right A



Code part D



1. Library Declarations:
   - The code begins with the library declaration: **library IEEE**; This allows you to use IEEE standard libraries in your code.
   - The **use IEEE.std_logic_1164.all;** statement imports the **std_logic_1164** package from the IEEE library. This package provides definitions and operations for working with digital signals represented by **std_logic**.
2. Entity Declaration:
   - The **entity** section defines the interface of the module, which is named "PART_D".
   - The module has the following elements:
     - **n**: A generic parameter of type **integer** with a default value of 16.
     - Ports:
       - **Cin**: An input signal of type **std_logic**.
       - **F**: An output signal of type **std_logic_vector** with a size of **n** bits, ranging from **n-1** downto 0.
       - **A**: An input signal of type **std_logic_vector** with a size of **n** bits.
       - **S**: An input signal of type **std_logic_vector** with a size of 2 bits.
3. Architecture Definition:
   - The **architecture** section defines the behavior of the module, and it is also named "PART_D".
   - The **begin** keyword marks the start of the architecture body.
4. Signal Assignment:

- The **F** signal is assigned a value based on conditional statements using the **when** keyword.
- The conditional statements are evaluated sequentially, and the first matching condition determines the value assigned to **F**.

Here's a breakdown of the conditional statements:

- If **S** is "00", **F** is assigned the value of concatenating bits **A(n-2)** to **A(0)** (excluding the most significant bit **A(n-1)**) with a '0' (zero) appended at the least significant position.
- If **S** is "01", **F** is assigned the value of concatenating bits **A(n-2)** to **A(0)** (excluding the most significant bit **A(n-1)**) with **A(n-1)** (most significant bit) appended at the least significant position.
- If **S** is "10", **F** is assigned the value of concatenating bits **A(n-2)** to **A(0)** (excluding the most significant bit A(n-1)) with **Cin** appended at the least significant position.
- If **S** is "11", **F** is assigned the value of "0000000000000000" (16-bit zero value).

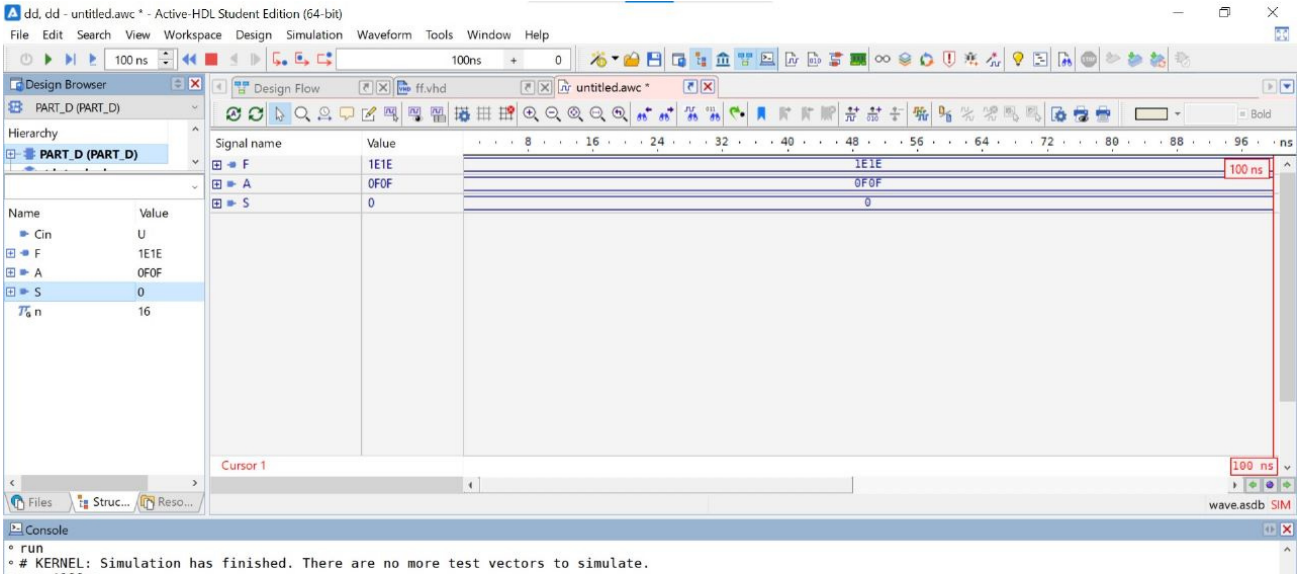
5.End of Architecture:

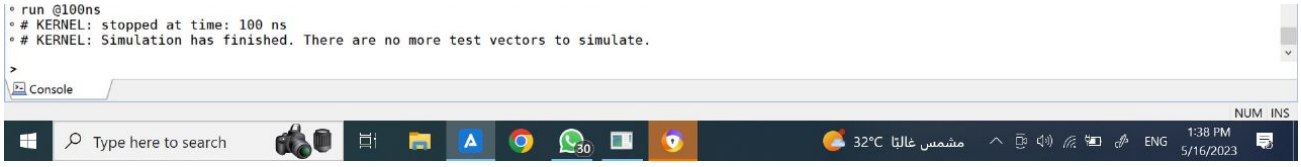
- The **end PART_D**; statement marks the end of the architecture.

In summary, the code describes a VHDL module named "PART_D" that performs different concatenation operations based on the values of input signals **A, Cin, and S.** The output signal **F** is assigned different values according to specific conditions. The specific concatenation operation performed depends on the value of **S** and determines which bits from **A** and additional bits (**Cin**, '0', or fixed value) are included in **F**.

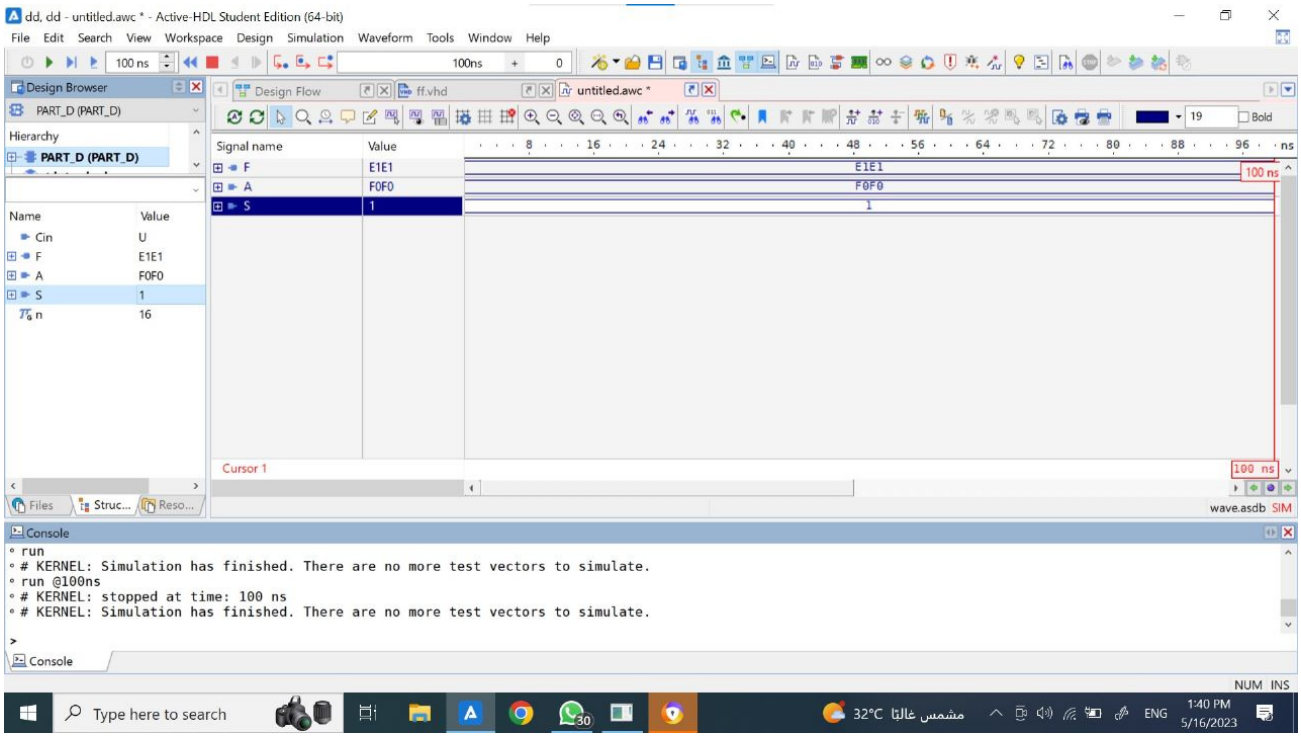
Waveform

1.F = Logic shift A

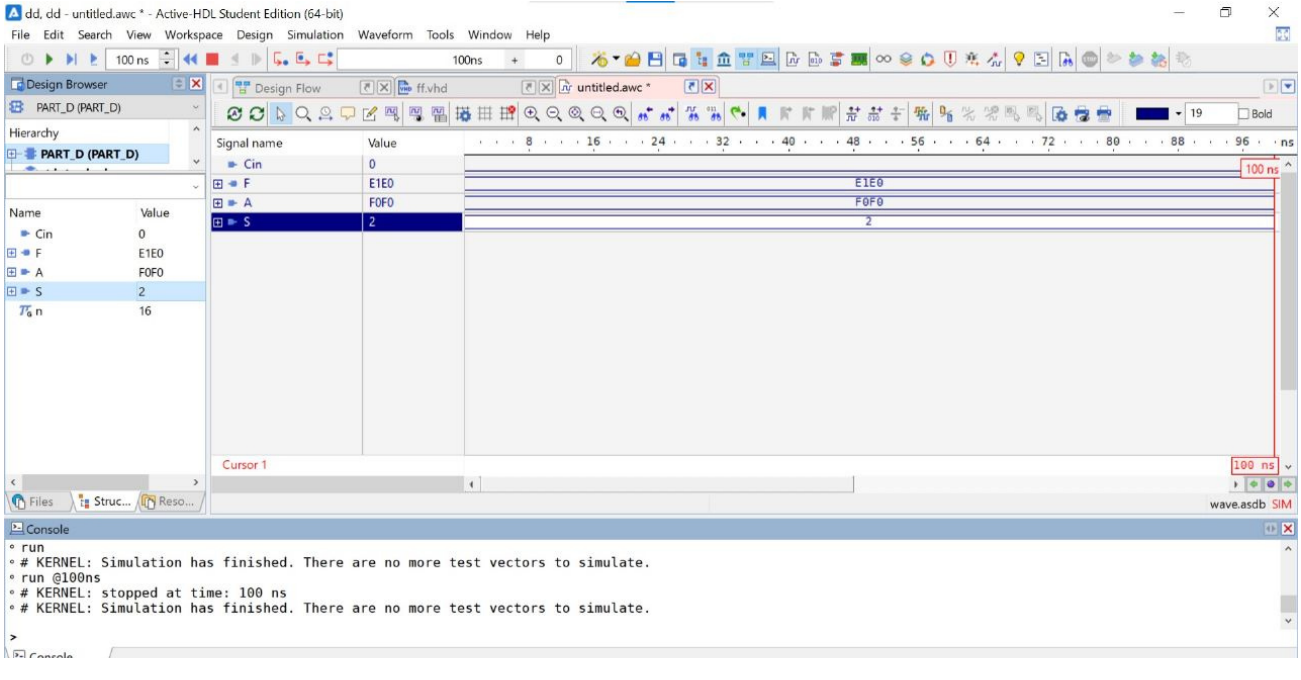


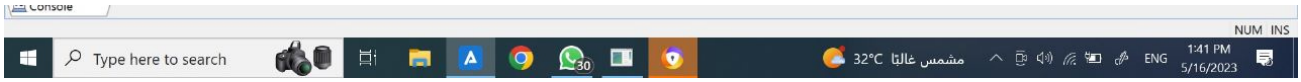


2.F = Rotate left A

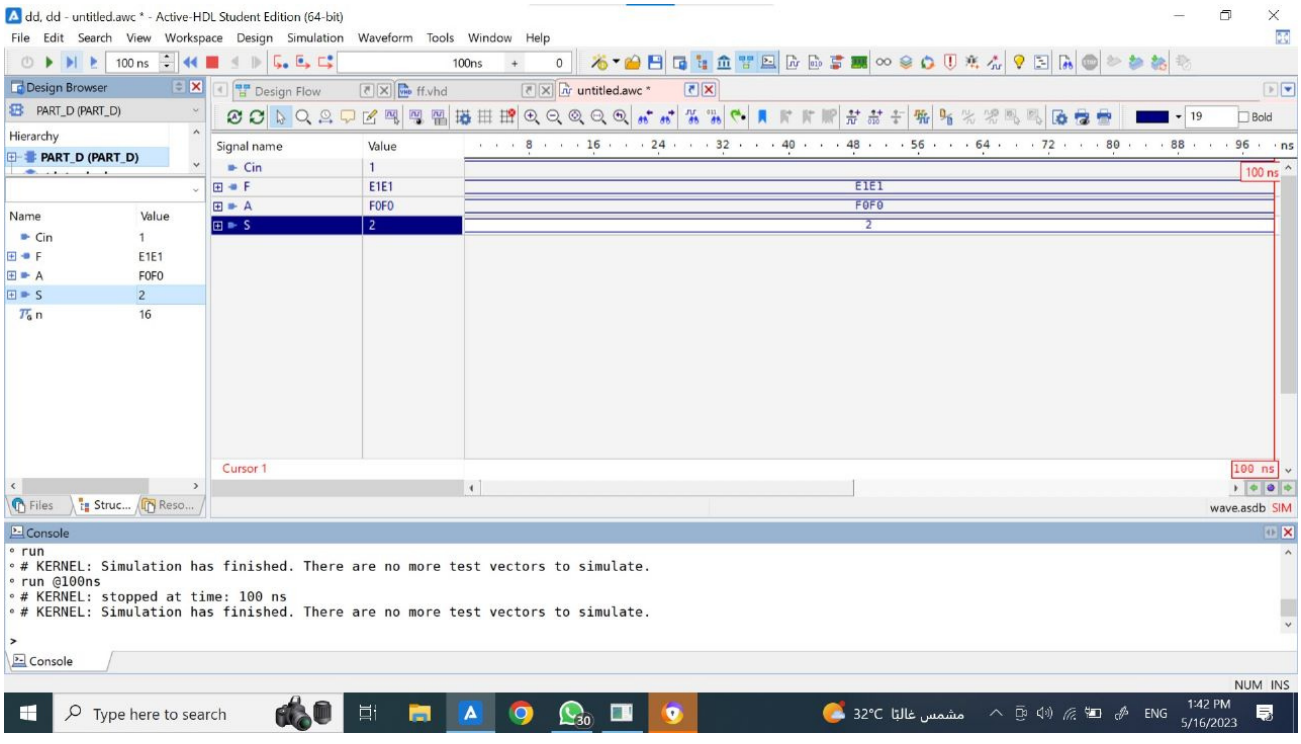


3.F = Rotate left A with carry part 1

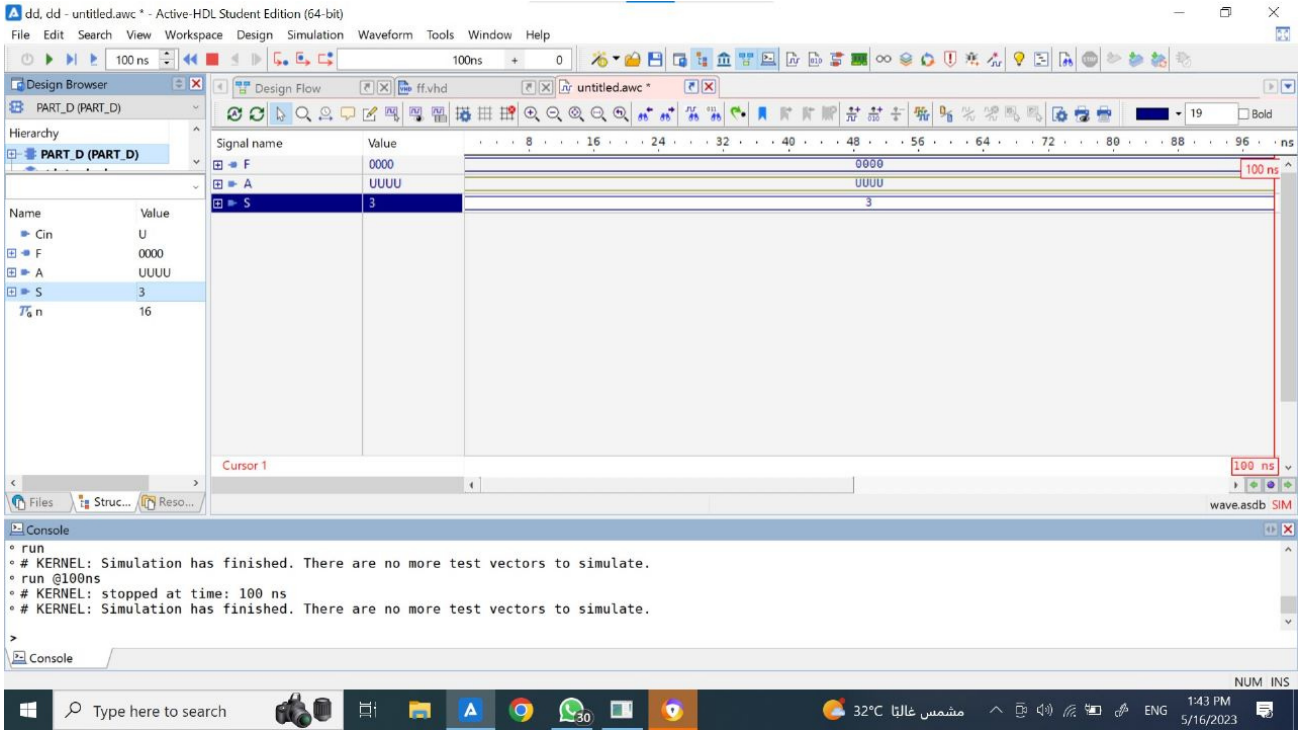




3.F = Rotate left A with carry part 2



4.F = X"0000"



Team members