SIT111 - Task 3.1P Implement the Full Adder in HDL

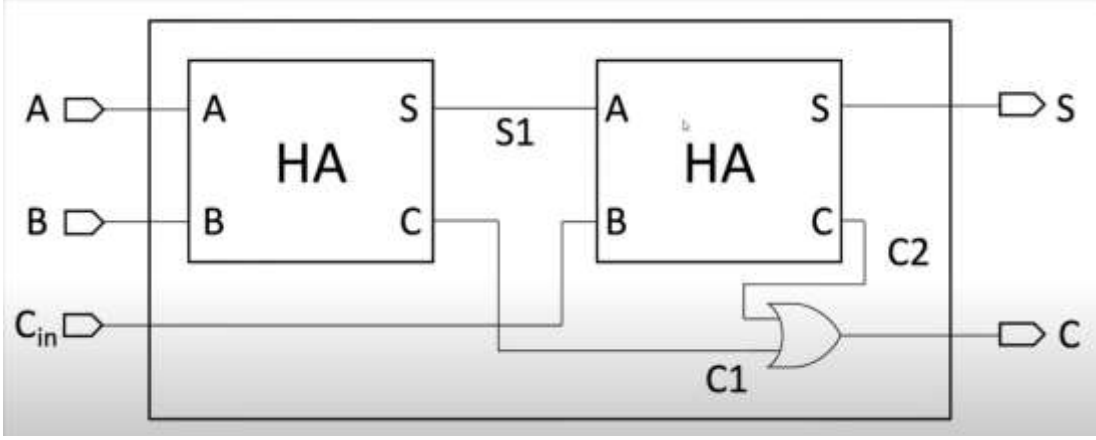
- Using one OR gate and two half adders, I implemented the full adder. However, I had to use an AND gate and Xor to implement the half adder first.

  HALF ADDER
  Xor (a=a, b=b, out=sum);
  And (a=a, b=b, out=carry);

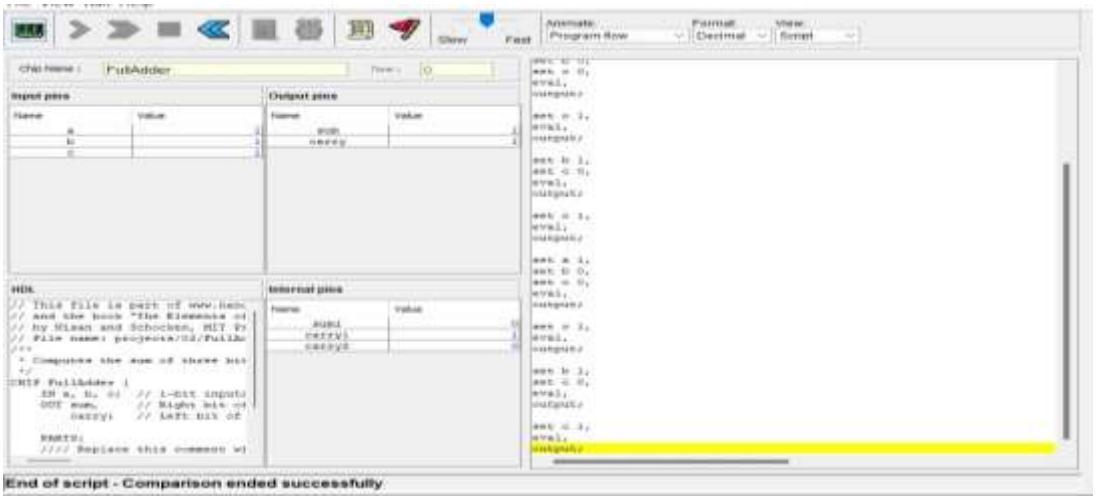
- After that, I utilised one OR gate, two half adders, and the script file for a full adder.

  FULL ADDER
  HalfAdder (a=a, b=b, sum=sum1, carry=carry1);
  HalfAdder (a=sum1, b=c, sum=sum, carry=carry2);
  Or (a=carry1, b=carry2, out=carry);

- The whole implemented adder's diagram is available here.



- Three inputs are needed for the complete adder: a, b, and c. The inputs of the first half adder are a=a and b=b, while the outputs are carry1 and sum1. The second half adder outputs the sum and carry2 after receiving inputs as a=sum1 and b=c. Next, two inputs are fed into the OR gate: b=carry2 from the second half adder and a=carry1 from the first half adder. outputs out=carry in the end. Thus, in total, it receives three inputs and produces two, which are the sum and carry. The file loaded and executed flawlessly in Nand Tetris. The page is provided below.


Slow
Fast
Animate:
Program flow
Format:
Decimal
View:
Script
Chip name :
FullAdder
Input pins
Output pins
HDL
Internal pins
End of script - Comparison ended successfully