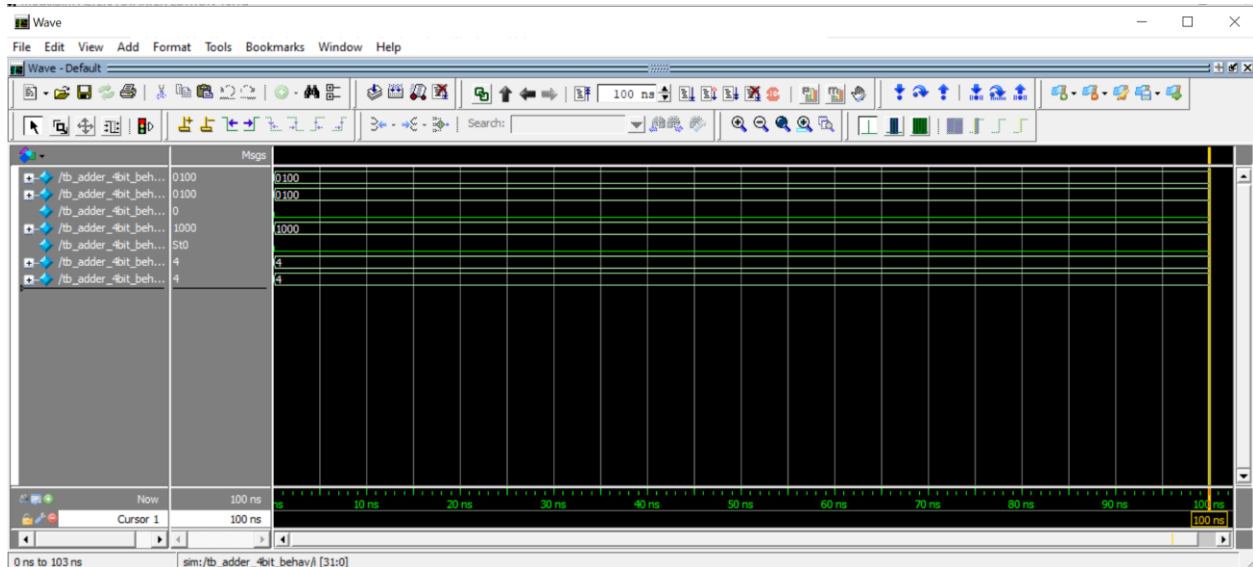


NAME: REIMARC G. CORPUZ
COURSE, YR & SECTION: BSCPE 3GF

DATE: OCTOBER 27, 2022
SCORE:

adder_4bit_behavioral

Wave:



Reflection:

In the Verilog coding style of a 4-bit adder using behavioral, the statement is assigned together with the output variable `carry_out` and `sum` enclosed in a curly bracket. In the testbench, it uses a parameter for the condition to satisfy the 4-bit adder. As always it will initiate the variable within the two connecting modules.

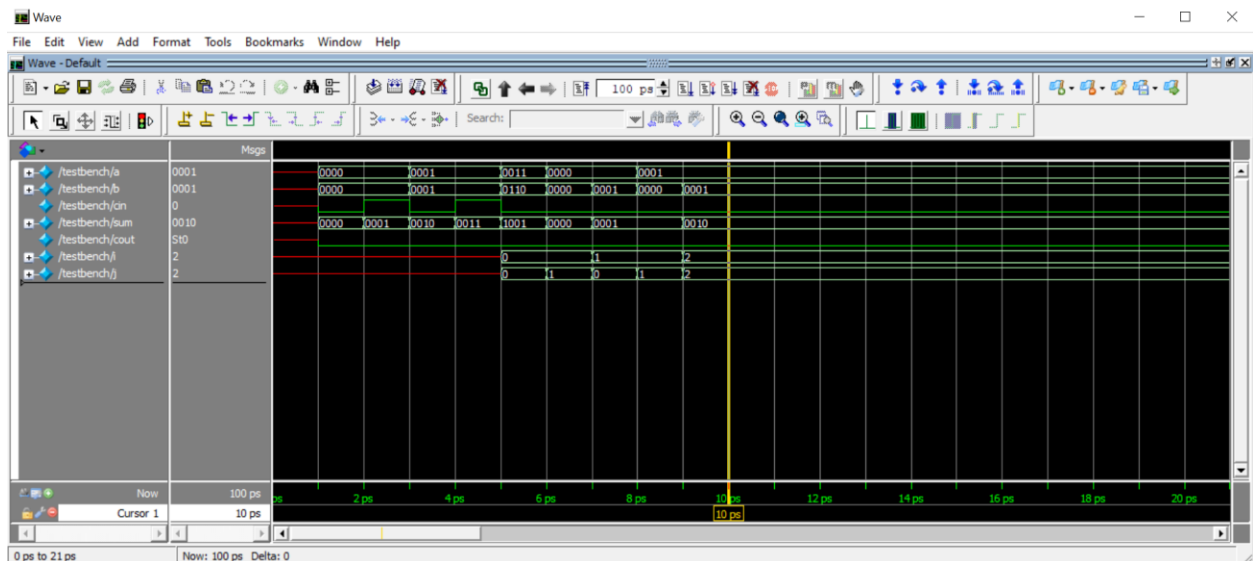
ripple_adder_4bit_structural

Transcript:

```
Transcript
File Edit View Bookmarks Window Help

# Reading C:/altera/13.0spl/modelsim_ase/tcl/vsim/pref.tcl
# Loading project ripple_adder_4bit_structural
# Compile of ripple_adder_4bit_structural.v was successful.
# Compile of ripple_adder_4bit_structural.v was successful.
ModelSim> vsim -gui work.testbench
# vsim -gui work.testbench
# Loading work.testbench
# Loading work.ripple_adder_4bit_structural
# Loading work.full_adder_structural
# Loading work.half_adder_structural
add wave -position insertpoint sim:/testbench/*
VSIM 3> run
# a=xxxx, b=xxxx, carry_in=x, sum=x, carry_out=x
# a=0000, b=0000, carry_in=0, sum=0, carry_out=0
# a=0000, b=0000, carry_in=1, sum=1, carry_out=0
# a=0001, b=0001, carry_in=0, sum=2, carry_out=0
# a=0001, b=0001, carry_in=1, sum=3, carry_out=0
# a=0011, b=0110, carry_in=0, sum=9, carry_out=0
# a=0000, b=0000, carry_in=0, sum=0, carry_out=0
# a=0000, b=0001, carry_in=0, sum=1, carry_out=0
# a=0001, b=0000, carry_in=0, sum=1, carry_out=0
# a=0001, b=0001, carry_in=0, sum=2, carry_out=0
VSIM 4>
```

Wave:



Reflection:

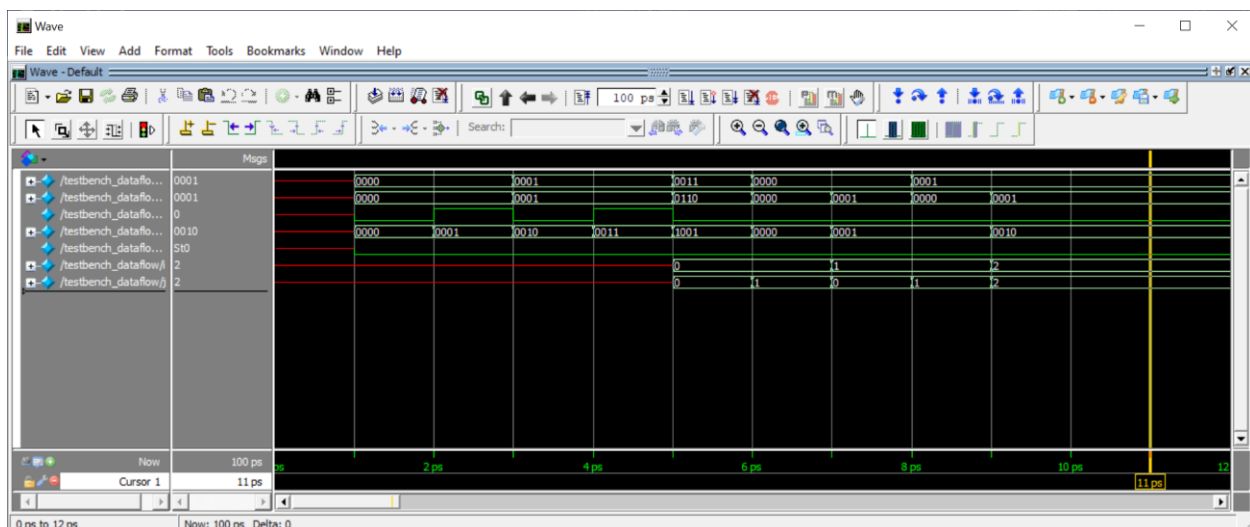
In the Verilog coding style of a 4-bit ripple adder using structural, it has 4 modules and it uses internal nets to connect those four. It also used a declaration and primitives to simulate the test bench. Within the test bench, the condition will satisfy to observe the effect in the module ports.

Ripple_adder_4bit_dataflow

Transcript:

```
Transcript
File Edit View Bookmarks Window Help
Transcript
# Reading C:/altera/13.0sp1/modelsim_ase/tcl/vsim/pref.tcl
# Loading project ripple_adder_4bit_dataflow
# Compile of ripple_adder_4bit_dataflow.v was successful.
ModelSim> vsim -gui work.testbench_dataflow
# vsim -gui work.testbench_dataflow
# Loading work.testbench_dataflow
# Loading work.ripple_adder_4bit_dataflow
# Loading work.full_adder_dataflow
add wave -position insertpoint sim:/testbench_dataflow/*
VSIIM 3> run
# a=xxxx, b=xxxx, carry_in=x, sum=x, carry_out=x
# a=0000, b=0000, carry_in=0, sum=0, carry_out=0
# a=0000, b=0000, carry_in=1, sum=1, carry_out=0
# a=0001, b=0001, carry_in=0, sum=2, carry_out=0
# a=0001, b=0001, carry_in=1, sum=3, carry_out=0
# a=0011, b=0110, carry_in=0, sum=9, carry_out=0
# a=0000, b=0000, carry_in=0, sum=0, carry_out=0
# a=0000, b=0001, carry_in=0, sum=1, carry_out=0
# a=0001, b=0000, carry_in=0, sum=1, carry_out=0
# a=0001, b=0001, carry_in=0, sum=2, carry_out=0
VSIIM 4> ]
```

Wave:



Reflection:

In a Verilog coding style using data flow, the wire output is assigned by the data flow style of the implement circuit in the modules. It also uses internal nets to connect four full adders using data flow. The data is transferred by the assigned value of the input to the wire/output variable.