# Digital Verification using SV and UVM

# Assignment-1

# Name: Fares Khalaf Sultan

# Q1)

# Before modification:

# Issue: reset signal didn’t change from 0 to 1 during simulation.

# Fix:

# 

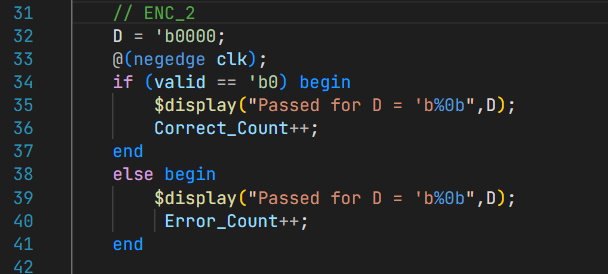
# Q2) Priority Encoder:

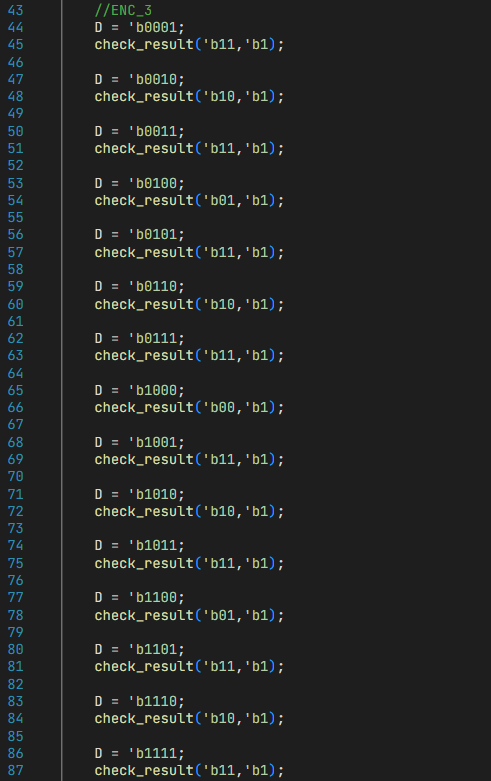
# Verification Plan:

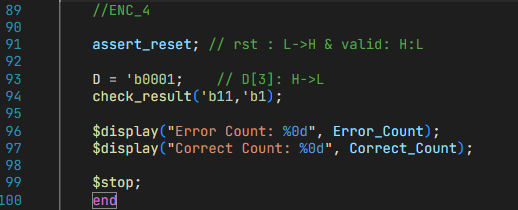
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Label** | **Design Requirement Description** | **Stimulus Generation** | **Functional Coverage** | **Functionality Check** |
| ENC\_1 | at posedge clk, if rst is asserted all outputs should be low | Directed at the start of the simulation | - | A checker in the testbench to make sure the output is correct |
| ENC\_2 | When D = ‘b0000, Valid should be low | Directed during the simulation | - | A Checker in the testbench to make sure the output is correct |
| ENC\_3 | exhaustively verify all possible inputs combination. | Directed during the simulation | - | A checker in the testbench to make sure the output is correct |
| ENC\_4 | Some values of the inputs to complete the toggle coverage | Directed at the end of the simulation |  | A checker in the testbench to make sure the output is correct |

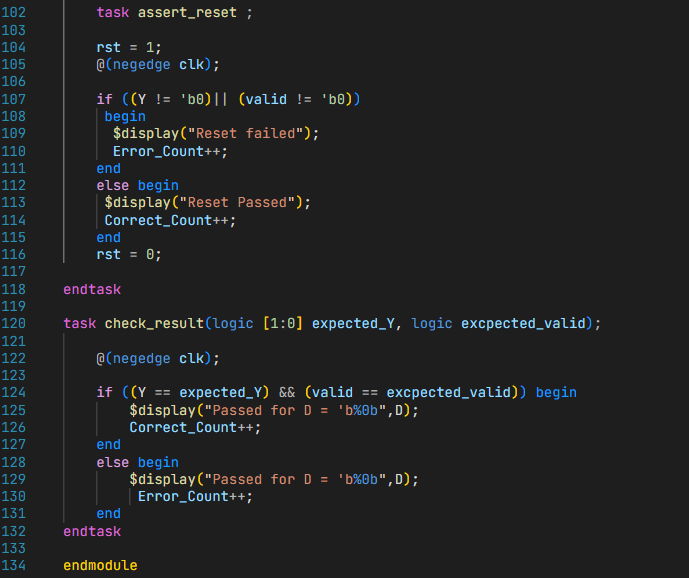
# TestBench:

# 





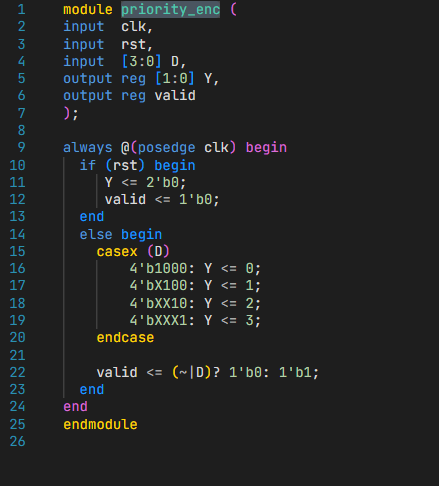




# Bugs found:

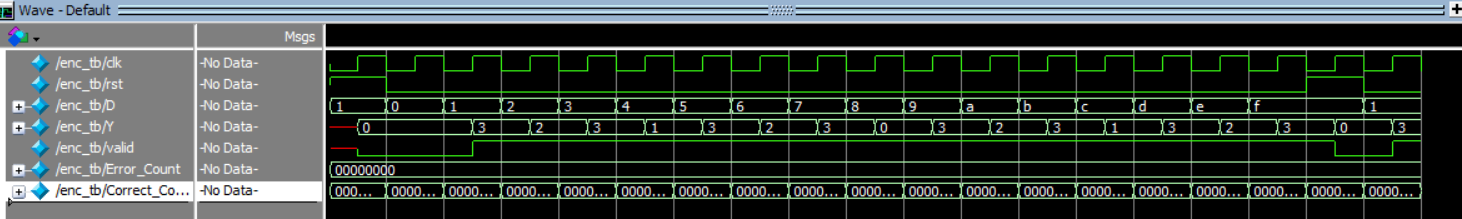
|  |  |  |  |
| --- | --- | --- | --- |
| **Label** | **Bug Description** | **Wrong code** | **Correction** |
|  | Output (Y) was defined as wire, not reg. |  | Defined Y as output reg |
| ENC1 | rst signal only resets the value of Y but does nothing to valid |  | (rst) signal must reset the value of valid too, and valid signal should take its value inside the else statement, to ensure the priority of rst signal |

# Fixed design:



# Result:

Waveform:



# Coverage Report:

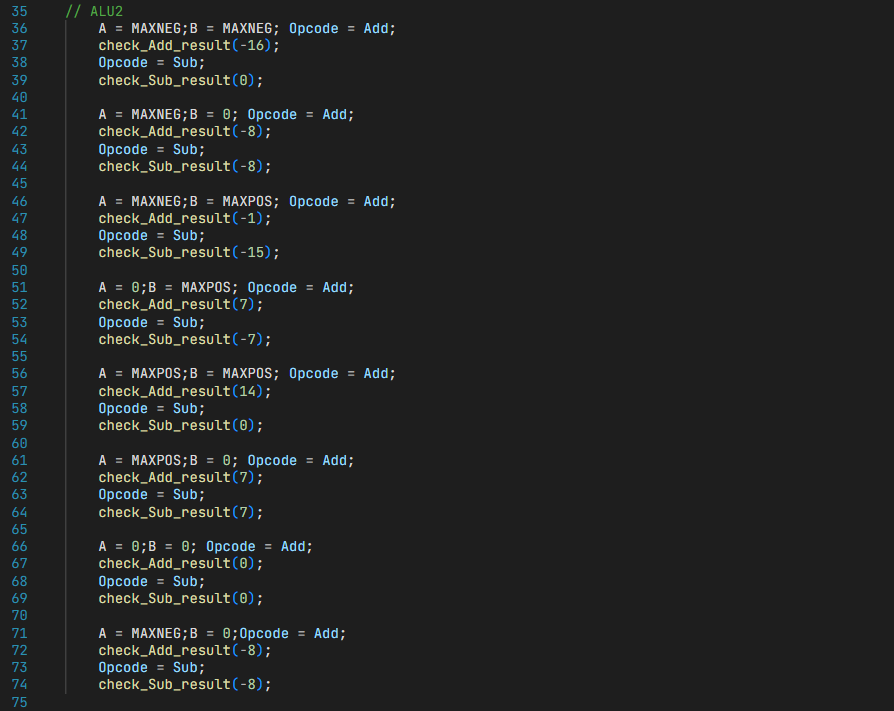
# Q3) ALU

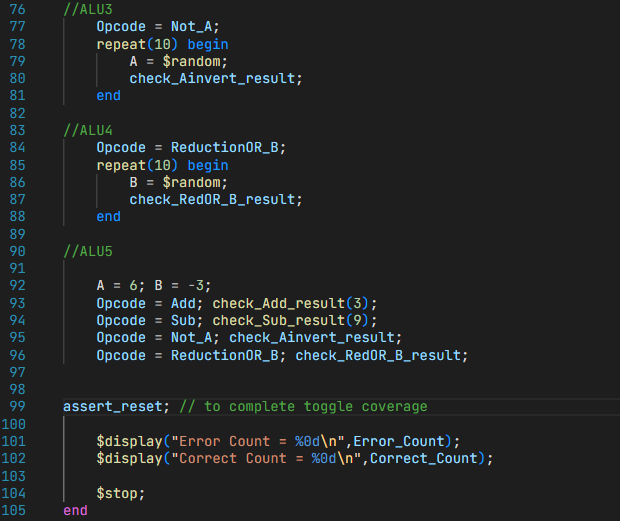
# Verification Plan:

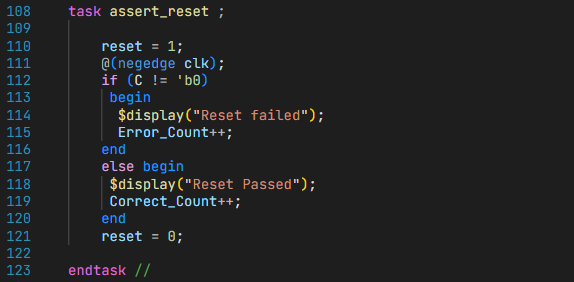
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Label** | **Design Requirement Description** | **Stimulus Generation** | **Functional Coverage** | **Functionality Check** |
| ALU1 | when reset is asserted, Output (C) should be Low | Directed at the start of the simulation | - | A checker in the testbench to make sure the output is correct |
| ALU2 | Hitting the bounary cases of A and B and try addition then subtraction | Directed during the simulation | - | A checker in the testbench to make sure the output is correct |
| ALU3 | verify bitwise invert input A | Random during the simulation | - | A checker in the testbench to make sure the output is correct |
| ALU4 | verify reduction OR input B | Random during the simulation | - | A checker in the testbench to make sure the output is correct |
| ALU5 | overall functionality of the ALU by select two values for A,B and the apply all opcodes on them | Random during the simulation | - | A checker in the testbench to make sure the output is correct |

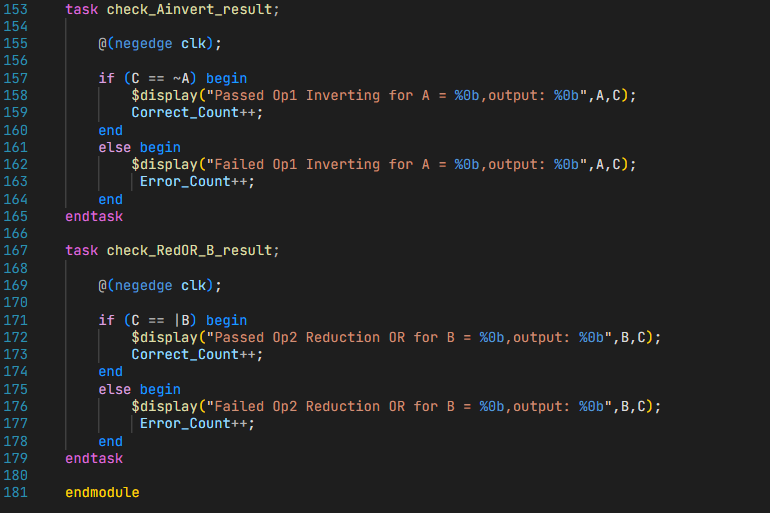
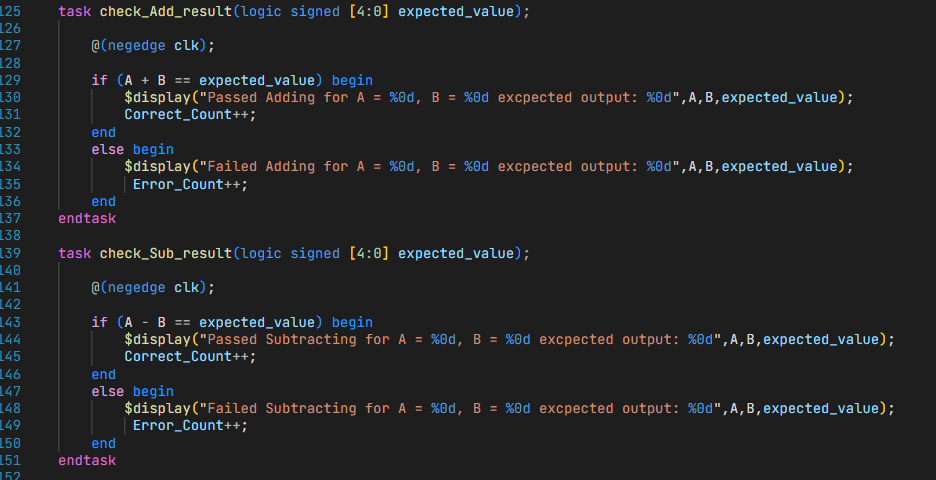
# TestBench:

# 







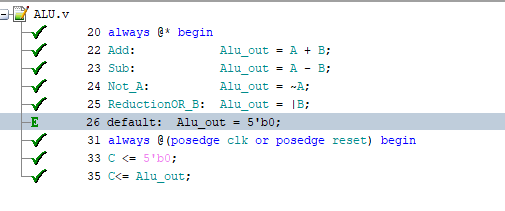


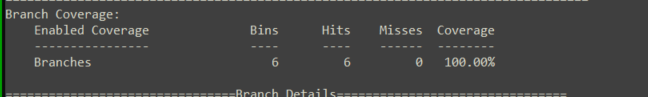
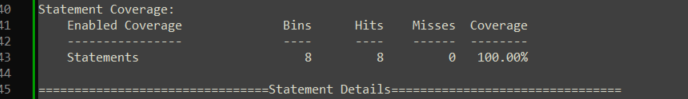
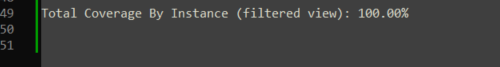
# Results:

# Waveform:

# Coverage Report:

* Missing Bin in Branch and statement coverages is the **default** of the Opcode Case Statement.
* This statement can’t be reached because the Case statement handles all the possible Opcode value, so this miss will be excluded from both reports.





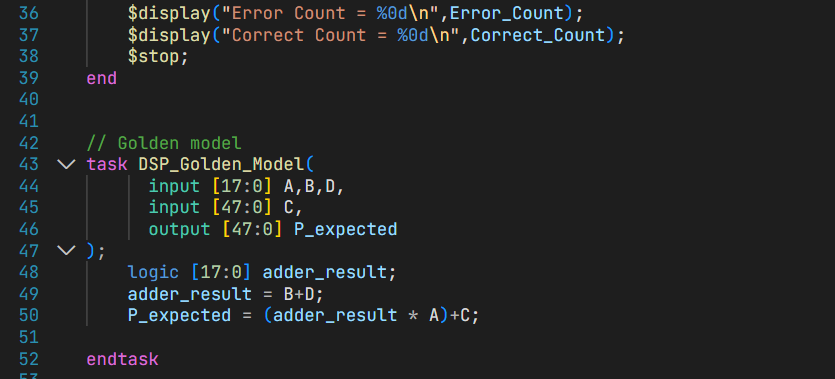
# Q3) DSP

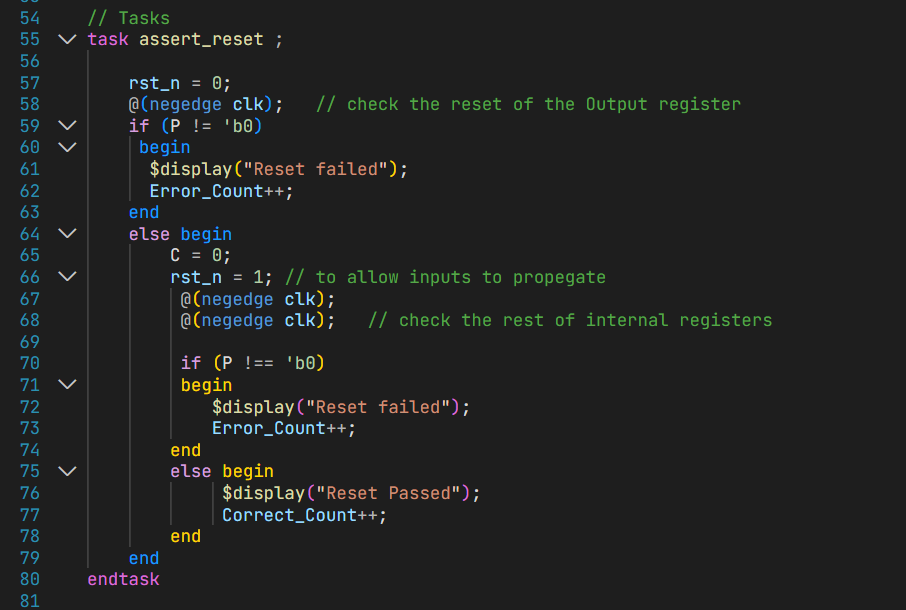
# Verification Plan:

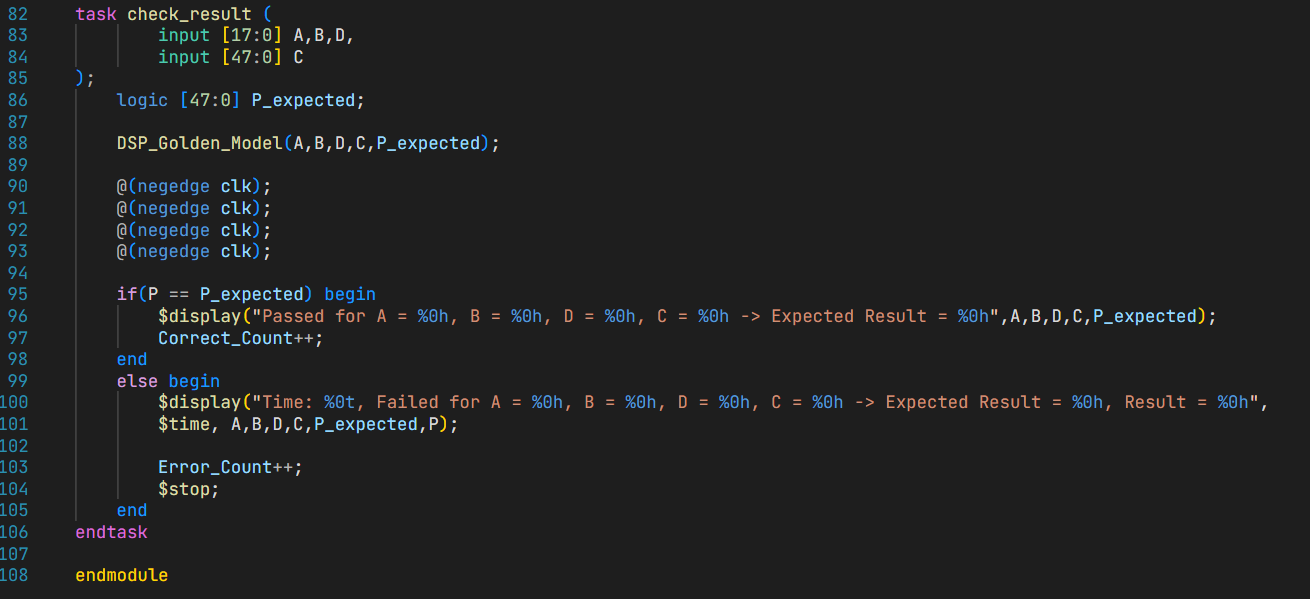
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Label** | **Design Requirement Description** | **Stimulus Generation** | **Functional Coverage** | **Functionality Check** |
| DSP1 | when reset is de-asserted, Output (P) should be **Low for three clock cycles** (Internal registers should also be reset). | Directed at the start of the simulation | - | A checker in the testbench to make sure the output is correct |
| DSP2 | checking the normal operation out of reset. | Randomized during the simulation | - | Comparing P to the output of a reference golden model’s P\_expected. |

# TestBench:

# 







# Results:

# Transcript:

# 

# DSP\_1:

# 

# DSP\_2:

# Bugs found:

|  |  |  |  |
| --- | --- | --- | --- |
| **Label** | **Bug Description** | **Wrong code** | **Correction** |
| DSP\_1 | Internal register(C\_reg) wasn’t affected by rst\_n. |  | Added C\_reg to the reset condition in the RTL. |
| DSP\_2 | the internal signal (adder\_out\_stg2) is redundant and causes an extra register after the first stage adder |  | Signal is totally removed, and multiplier\_out signal will directly take the value of adder\_out\_stg1. |

# Coverage Report:

# 

* Un-Toggled upper bits of mult\_out (mult\_out[47:36]) will never be toggled, since the multiplier inputs (A and B/D) are **18-bit signals**, so their multiplication will need at most **36 bits** (mult\_out[35:0]).

