Lab 4: UVM Introduction

Sri Sai Harshith Yellanki
Shamit Savant

Part 3 – DUT Extension with Checksum Functionality

Objective

In this final part of the lab, our goal was to add new functionality to the packet filter design. Specifically, we had to compute an 8-bit checksum across the entire packet and append it at the end of the outgoing data stream. We then had to verify this feature using the updated testbench and confirm that our changes were correct through both simulation and synthesis.

What We Did

This part had two main tasks: modifying the DUT to compute and send the checksum, and updating the testbench to validate it. Below is a breakdown of the key files we worked on and what we changed in each one.

Filter.sv – DUT:

- We updated the filter module to calculate a running 8-bit checksum using XOR for each byte of data, starting from the packet header and including the payload.
- We ensured that for packets with an odd number of valid bytes, the checksum was sent as part of the final beat alongside the last data byte.
- For even-length packets, we delayed the final tlast signal and inserted an additional beat that contained only the checksum.
- We added logic to properly handle the AXI stream signaling, ensuring that tvalid and tlast were asserted in the correct cycles.

Filter_env.svh – Environment:

- We ensured that is_packet_level was set correctly so that transactions were processed at the packet level, which is necessary for checksum validation.

Filter_tb.sv – Top-Level Testbench:

- This helped ensure the checksum computation worked correctly under various flow control conditions.

Makefile:

- We reverted the UVM test name to filter_packet_test to run the final test for checksum validation.
- This allowed us to execute the correct test directly via make sim or make gui.

Simulation and Results

After making the changes, we ran the simulation using GUI mode. The updated DUT behaved as expected:

- For every packet that matched a supported message type, the output included the original data plus a checksum byte.
- The test passed cleanly, and the scoreboard confirmed that all checksums were valid and that input and output packets matched in structure.
- We saw multiple “Test passed” messages in the simulation log, confirming that the design was working as intended.

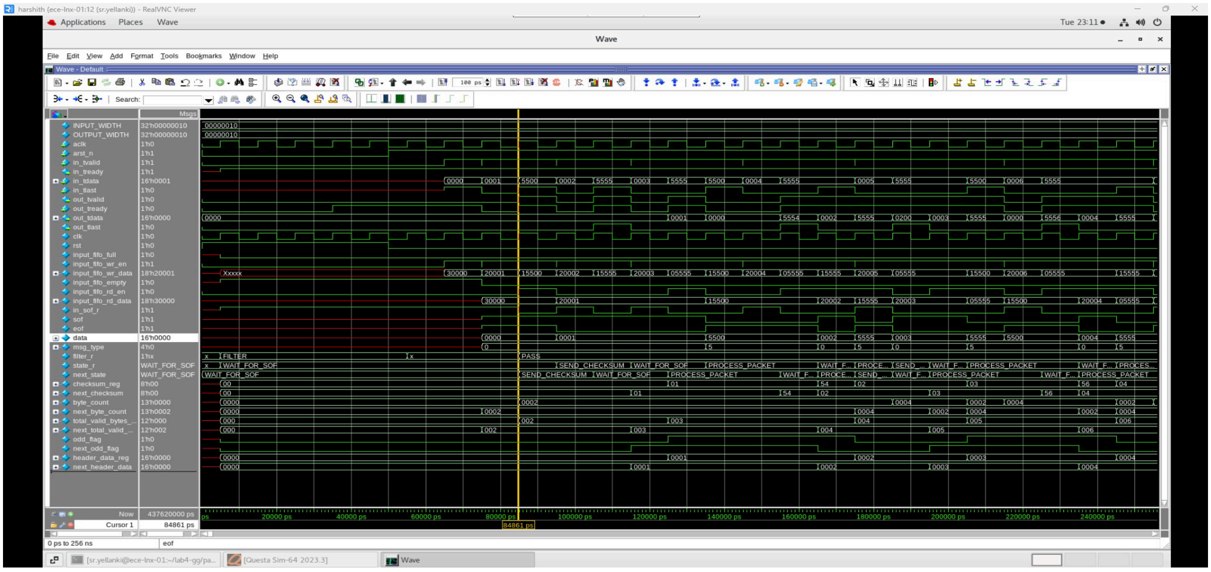
Additionally, we confirmed that packets with odd and even lengths were handled correctly. In the waveform, the checksum appeared either as a second byte in the last beat (for odd-length packets) or as a separate beat (for even-length packets), as per the specification.

Synthesis in Vivado

We also synthesized the design in Vivado to check for logic issues or warnings. The design synthesized without errors, and we verified that there were no latch-related warnings or critical path issues. The added checksum logic was implemented cleanly using simple XOR gates and conditional control for data output. A screenshot of the synthesis summary is attached as part of the final report.

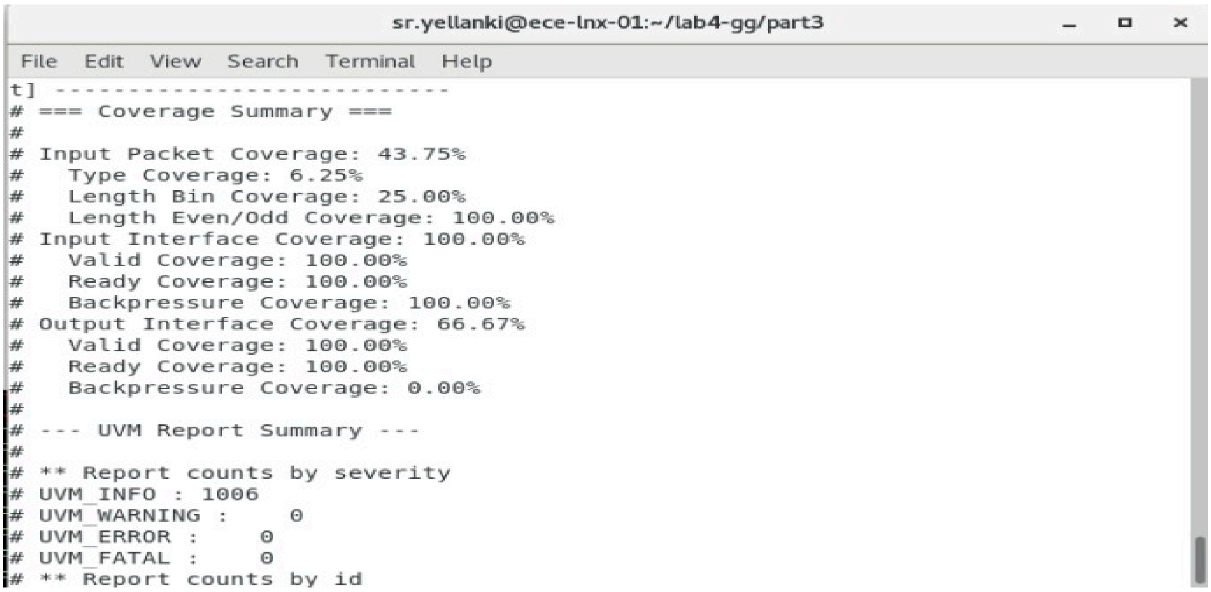
Screenshots

Our debug test simulation



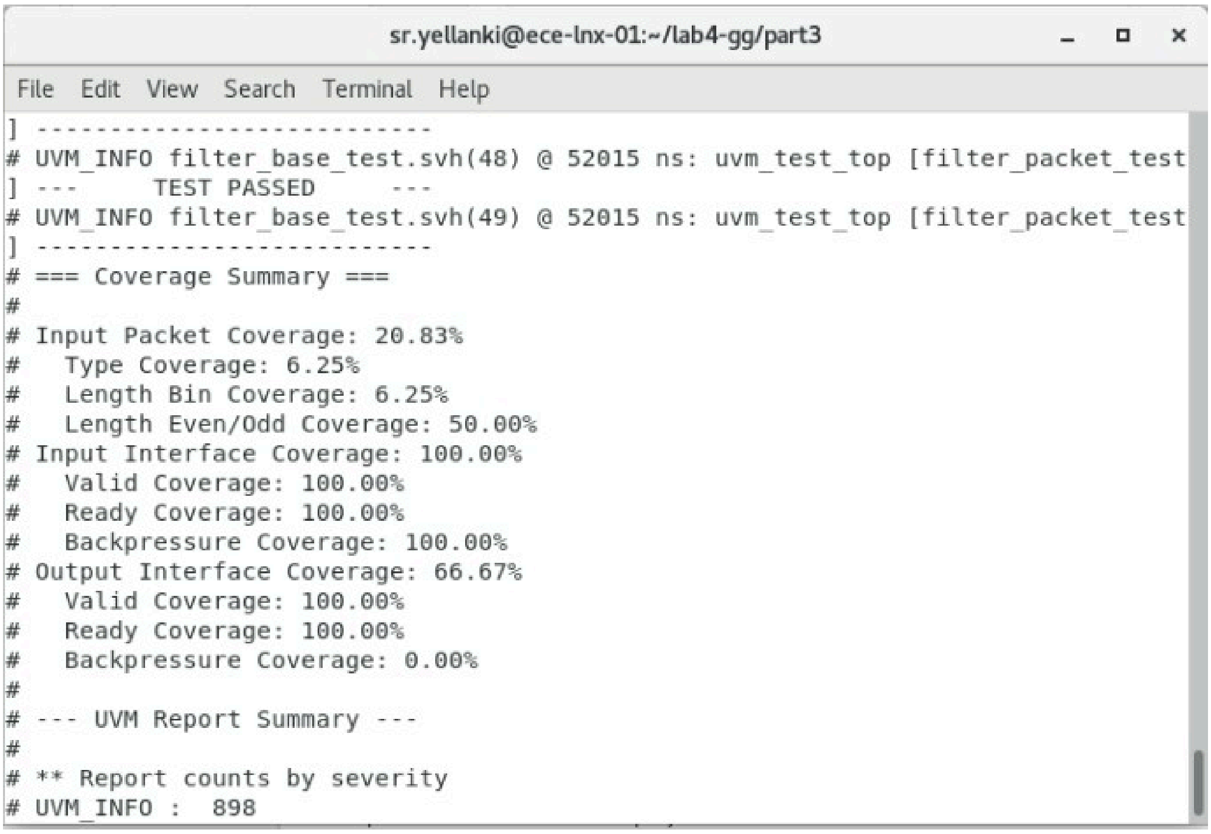
```
INPUT_WIDTH            32'h00000010
OUTPUT_WIDTH           32'h00000010
aclk                   1'h0
arst_n                 1'h1
in_tvalid              1'h1
in_tready              1'h1
in_tdata               16'h0001
in_tlast               1'h0
out_tvalid             1'h0
out_tready             1'h0
out_tdata              16'h0000
out_tlast              1'h0
clk                    1'h0
rst                    1'h0
input_fifo_full        1'h0
input_fifo_wr_en       1'h1
input_fifo_wr_data     18'h20001
input_fifo_empty       1'h0
input_fifo_rd_en       1'h0
input_fifo_rd_data     18'h30000
in_sof_r               1'h1
sof                    1'h1
eof                    1'h1
data                   16'h0000
msg_type               4'h0
filter_r               1'hx
state_r                WAIT_FOR_SOF
next_state             WAIT_FOR_SOF
checksum_reg           8'h00
next_checksum          8'h00
byte_count             13'h0000
next_byte_count        13'h0002
total_valid_bytes_...  12'h000
next_total_valid_...   12'h002
odd_flag               1'h0
next_odd_flag          1'h0
header_data_reg        16'h0000
next_header_data       16'h0000
Now                    437620000 ps
Cursor 1               84861 ps
0 ps to 256 ns         eof
```

Our debug test coverage



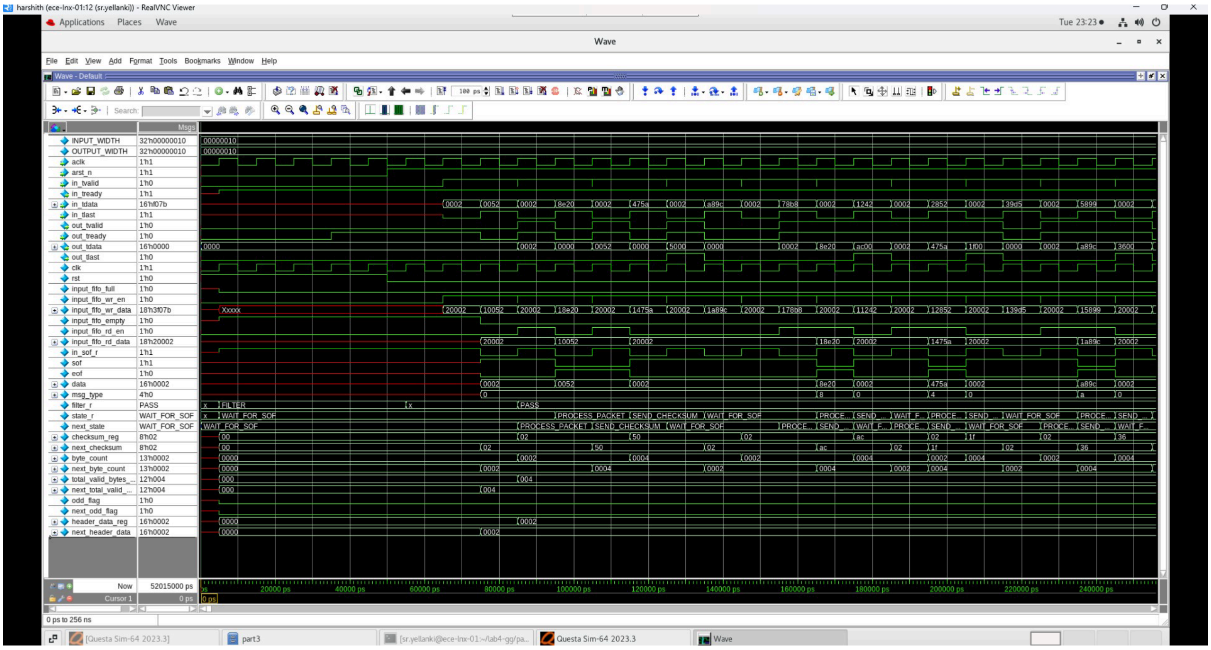
```
t] ----------------------------
# === Coverage Summary ===
#
# Input Packet Coverage: 43.75%
#   Type Coverage: 6.25%
#   Length Bin Coverage: 25.00%
#   Length Even/Odd Coverage: 100.00%
# Input Interface Coverage: 100.00%
#   Valid Coverage: 100.00%
#   Ready Coverage: 100.00%
#   Backpressure Coverage: 100.00%
# Output Interface Coverage: 66.67%
#   Valid Coverage: 100.00%
#   Ready Coverage: 100.00%
#   Backpressure Coverage: 0.00%
#
# --- UVM Report Summary ---
#
# ** Report counts by severity
# UVM_INFO : 1006
# UVM_WARNING :    0
# UVM_ERROR :    0
# UVM_FATAL :    0
# ** Report counts by id
```

Given packet test coverage



```
] ---------------------------
# UVM_INFO filter_base_test.svh(48) @ 52015 ns: uvm_test_top [filter_packet_test
] ---      TEST PASSED      ---
# UVM_INFO filter_base_test.svh(49) @ 52015 ns: uvm_test_top [filter_packet_test
] ---------------------------
# === Coverage Summary ===
#
# Input Packet Coverage: 20.83%
#   Type Coverage: 6.25%
#   Length Bin Coverage: 6.25%
#   Length Even/Odd Coverage: 50.00%
# Input Interface Coverage: 100.00%
#   Valid Coverage: 100.00%
#   Ready Coverage: 100.00%
#   Backpressure Coverage: 100.00%
# Output Interface Coverage: 66.67%
#   Valid Coverage: 100.00%
#   Ready Coverage: 100.00%
#   Backpressure Coverage: 0.00%
#
# --- UVM Report Summary ---
#
# ** Report counts by severity
# UVM_INFO :   898
```

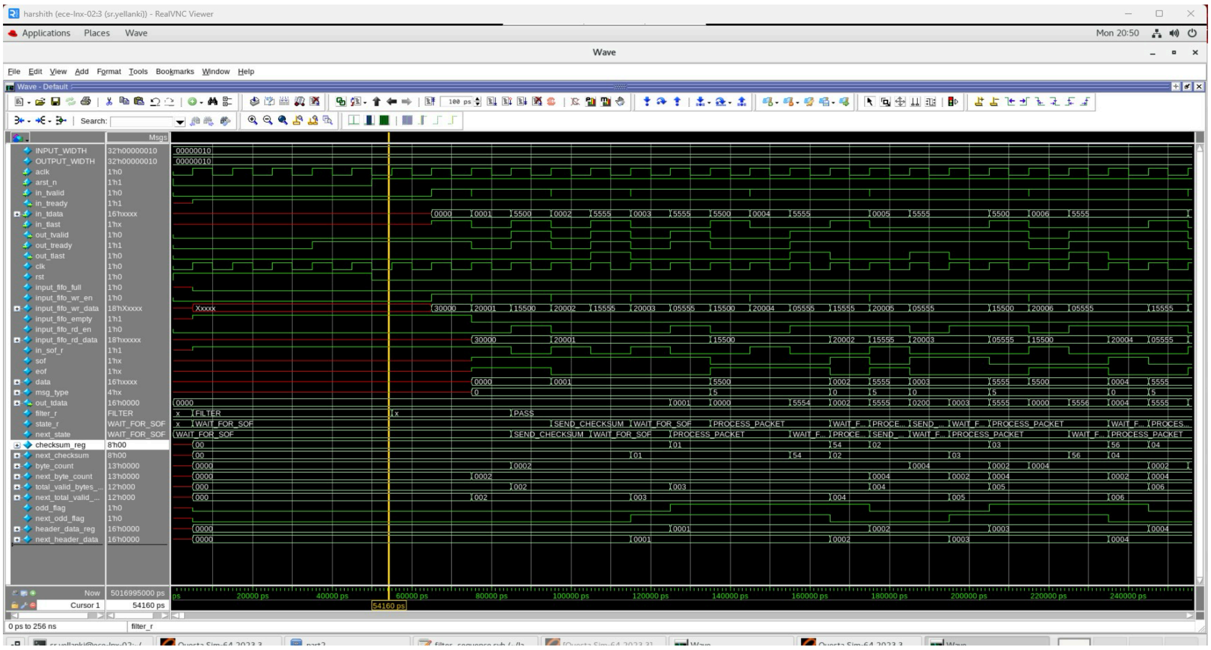
Given packet test simulation



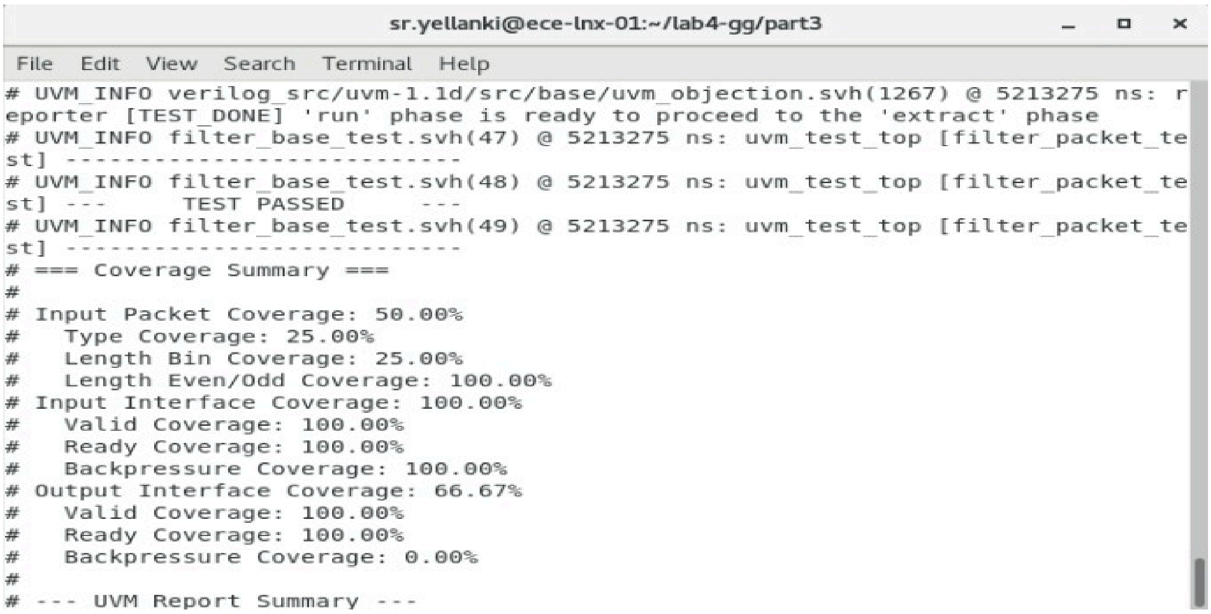
	Msgs
INPUT_WIDTH	32'h00000010
OUTPUT_WIDTH	32'h00000010
aclk	1'h1
arst_n	1'h1
in_tvalid	1'h0
in_tready	1'h1
in_tdata	16'hf07b
in_tlast	1'h1
out_tvalid	1'h0
out_tready	1'h0
out_tdata	16'h0000
out_tlast	1'h0
clk	1'h1
rst	1'h0
input_fifo_full	1'h0
input_fifo_wr_en	1'h0
input_fifo_wr_data	18'h3f07b
input_fifo_empty	1'h0
input_fifo_rd_en	1'h0
input_fifo_rd_data	18'h20002
in_sof_r	1'h1
sof	1'h1
eof	1'h0
data	16'h0002
msg_type	4'h0
filter_r	PASS
state_r	WAIT_FOR_SOF
next_state	WAIT_FOR_SOF
checksum_reg	8'h02
next_checksum	8'h02
byte_count	13'h0002
next_byte_count	13'h0002
total_valid_bytes_...	12'h004
next_total_valid_...	12'h004
odd_flag	1'h0
next_odd_flag	1'h0
header_data_reg	16'h0002
next_header_data	16'h0002
Now	52015000 ps
Cursor 1	0 ps

0 ps to 256 ns

Our packet test simulation

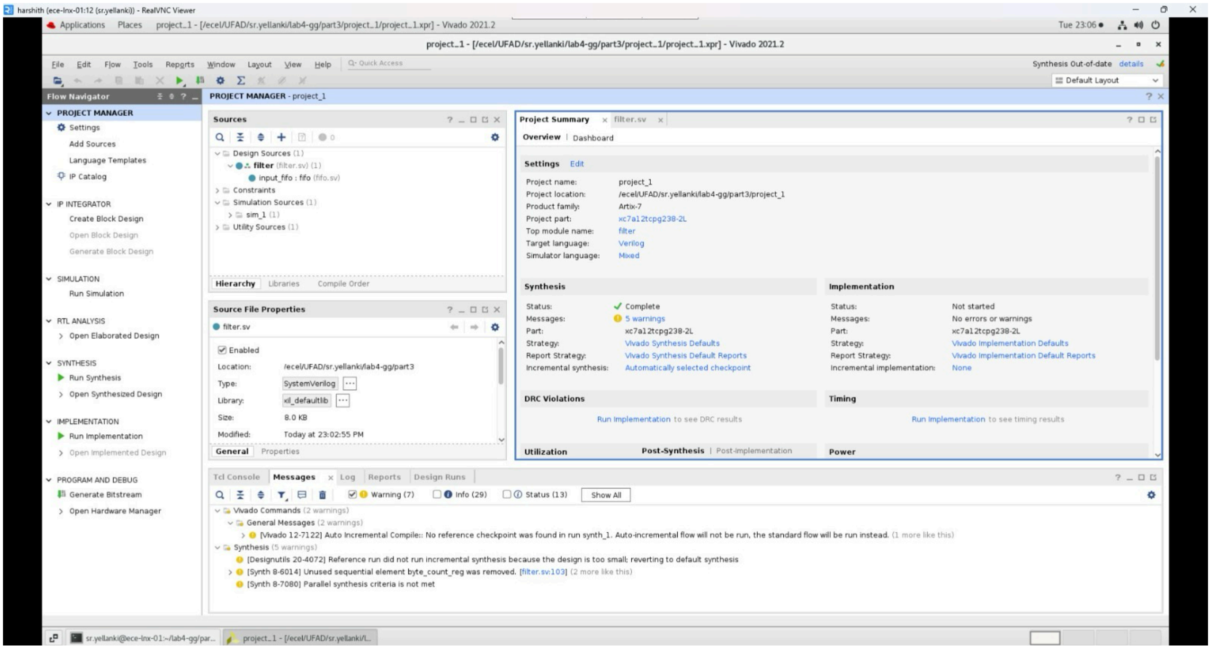


Our packet test coverage


sr.yellanki@ece-lnx-01:~/lab4-gg/part3
File Edit View Search Terminal Help

```
# UVM_INFO verilog_src/uvm-1.1d/src/base/uvm_objection.svh(1267) @ 5213275 ns: r
eporter [TEST_DONE] 'run' phase is ready to proceed to the 'extract' phase
# UVM_INFO filter_base_test.svh(47) @ 5213275 ns: uvm_test_top [filter_packet_te
st] ---------------------------
# UVM_INFO filter_base_test.svh(48) @ 5213275 ns: uvm_test_top [filter_packet_te
st] ---      TEST PASSED      ---
# UVM_INFO filter_base_test.svh(49) @ 5213275 ns: uvm_test_top [filter_packet_te
st] ---------------------------
# === Coverage Summary ===
#
# Input Packet Coverage: 50.00%
#   Type Coverage: 25.00%
#   Length Bin Coverage: 25.00%
#   Length Even/Odd Coverage: 100.00%
# Input Interface Coverage: 100.00%
#   Valid Coverage: 100.00%
#   Ready Coverage: 100.00%
#   Backpressure Coverage: 100.00%
# Output Interface Coverage: 66.67%
#   Valid Coverage: 100.00%
#   Ready Coverage: 100.00%
#   Backpressure Coverage: 0.00%
#
# --- UVM Report Summary ---
```

Synthesis



project_1 - [/ecel/UFAD/sr.yellanki/lab4-gg/part3/project_1/project_1.xpr] - Vivado 2021.2

Sources

- Design Sources (1)
  - filter (filter.sv) (1)
    - input_fifo : fifo (fifo.sv)
- Constraints
- Simulation Sources (1)
  - sim_1 (1)
- Utility Sources (1)

Source File Properties

filter.sv

☑ Enabled

Location:	/ecel/UFAD/sr.yellanki/lab4-gg/part3
Type:	SystemVerilog
Library:	xil_defaultlib
Size:	8.0 KB
Modified:	Today at 23:02:55 PM

Settings Edit

Project name:	project_1
Project location:	/ecel/UFAD/sr.yellanki/lab4-gg/part3/project_1
Product family:	Artix-7
Project part:	xc7a12tcpg238-2L
Top module name:	filter
Target language:	Verilog
Simulator language:	Mixed

Synthesis

Status:	✓ Complete
Messages:	5 warnings
Part:	xc7a12tcpg238-2L
Strategy:	Vivado Synthesis Defaults
Report Strategy:	Vivado Synthesis Default Reports
Incremental synthesis:	Automatically selected checkpoint

Implementation

Status:	Not started
Messages:	No errors or warnings
Part:	xc7a12tcpg238-2L
Strategy:	Vivado Implementation Defaults
Report Strategy:	Vivado Implementation Default Reports
Incremental implementation:	None

DRC Violations

Run Implementation to see DRC results

Timing

Run Implementation to see timing results

Utilization Post-Synthesis | Post-Implementation

Power

☑ Warning (7) ☐ Info (29) ☐ Status (13)

```
Vivado Commands (2 warnings)
  General Messages (2 warnings)
    [Vivado 12-7122] Auto Incremental Compile:: No reference checkpoint was found in run synth_1. Auto-incremental flow will not be run, the standard flow will be run instead. (1 more like this)
Synthesis (5 warnings)
  [Designutils 20-4072] Reference run did not run incremental synthesis because the design is too small; reverting to default synthesis
  [Synth 8-6014] Unused sequential element byte_count_reg was removed. [filter.sv:103] (2 more like this)
  [Synth 8-7080] Parallel synthesis criteria is not met
```