



Digital Design Verification

Lab Manual # 38 – Creating YAPP Interface UVM

Name: Ayesha Binte Safiullah
Submitted To: NCDC

Release: 1.0

Date: 6-Aug-2024

NUST Chip Design Centre (NCDC), Islamabad, Pakistan



Copyrights ©, NUST Chip Design Centre (NCDC). All Rights Reserved. This document is prepared by NCDC and is for intended recipients only. It is not allowed to copy, modify, distribute or share, in part or full, without the consent of NCDC officials.

Revision History

Revision Number	Revision Date	Revision By	Nature of Revision	Approved By
1.0	6/08/2024	Amber Khan	Complete Manual	-



Contents

Contents	2
Objective.....	3
Tools	3
Instructions for Lab Tasks	3
Task 1: Creating a Simple UVC	4
Task 2: Using Factories	6



Objective

The objectives of this lab are

- To the front end of a UVM Verification Component (UVC) and to explore the built-in phases of `uvm_component`.
- To create verification components and data using factory methods, and to implement test classes using configurations.

Tools

- SystemVerilog
- Cadence Xcelium

Instructions for Lab Tasks

The submission must follow the hierarchy below, with the folder named after the student (no spaces), and the file names exactly as listed below.

```
./student_name_lab10/
├── task1_uvc/
│   ├── tb/
│   │   ├── file.f
│   │   ├── top.sv
│   │   ├── router_tb.sv
│   │   └── router_test_lib.sv
│   └── sv/
│       ├── yapp_pkj.sv
│       ├── yapp_packet.sv
│       ├── yapp_tx_env.sv
│       ├── yapp_tx_agent.sv
│       ├── yapp_tx_driver.sv
│       ├── yapp_tx_monitor.sv
│       ├── yapp_tx_sequencer.sv
│       └── yapp_tx_seqs.sv
└── task2_factory/
    ├── tb/
    │   ├── file.f
    │   ├── top.sv
    │   ├── router_tb.sv
    │   └── router_test_lib.sv
    └── sv/
        ├── yapp_pkj.sv
        ├── yapp_packet.sv
        ├── yapp_tx_env.sv
        ├── yapp_tx_agent.sv
        ├── yapp_tx_driver.sv
        ├── yapp_tx_monitor.sv
        ├── yapp_tx_sequencer.sv
        └── yapp_tx_seqs.sv
```



Task 1: Creating a Simple UVC

You will be creating the driver, sequencer, monitor, agent and env for the UVC to drive the YAPP input port of the router. You will focus on the transmit (TX) agent for this task.

Checking the UVC Hierarchy

1. In the `task1_uvc/tb` directory, run a simulation using the `base_test` test class:

a. Find the topology print.

Does the hierarchy match your expectations?

Answer: Yes.

b. Use the topology print to find the full hierarchical pathname from your test class to your UVC sequencer (e.g., `tb.yapp.agent.sequencer`) and write it below.

Sequencer pathname: `env.uvc.agent.sequencer`

c. Use your topology to find the value of the `is_active` property of the YAPP agent. *What is the value of the `is_active` variable when you printed the hierarchy?*

Answer: `UVM_ACTIVE`

```

UVM INFO router_tb.sv(15) @ 0: uvm test top.env [router_tb] [Router TB] BUILD PHASE EXECUTING!
UVM INFO @ 0: reporter [UVMTOP] UVM testbench topology:
-----
Name                               Type                               Size Value
-----
uvm_test_top                       base_test                         - @0000
env                               router_tb                         - @0000
  uvc                             yapp_env                         - @0279
    agent                         yapp_agent                       - @0277
      driver                     yapp_driver                      - @0280
        rsp_port                 uvm_analysis_port                - @0292
          seq_item_port          uvm_seq_item_pull_port           - @0281
            monitor              yapp_monitor                     - @0292
              sequencer          yapp_sequencer                   - @0294
                rsp_export        uvm_analysis_export              - @0302
                  seq_item_export uvm_seq_item_pull_imp            - @0300
                    arbitration_queue array                             0 -
                      num_last_reqs integral 32 'd1
                      num_last_rsp integral 32 'd1
                      is_active uvm_active_passive_enum 1 UVM_ACTIVE
-----
UVM INFO ../sv/yapp_tx_driver.sv(23) @ 0: uvm test top.env.uvc.agent.driver [yapp_driver] Running Simulation Driver!
UVM INFO ../sv/yapp_tx_monitor.sv(14) @ 0: uvm test top.env.uvc.agent.monitor [yapp_monitor] Running Simulation Monitor!
UVM INFO ../sv/yapp_tx_sequencer.sv(10) @ 0: uvm test top.env.uvc.agent.sequencer [yapp_sequencer] Running Simulation Sequencer!
UVM INFO ../sv/yapp_tx_agent.sv(31) @ 0: uvm test top.env.uvc.agent [yapp_agent] Running Simulation Agent!
UVM INFO ../sv/yapp_env.sv(12) @ 0: uvm test top.env.uvc [yapp_env] Running Simulation Environment!
UVM INFO router_tb.sv(21) @ 0: uvm test top.env [router_tb] Running Simulation Router TB!
UVM INFO router_test_lib.sv(25) @ 0: uvm test top [base_test] Running Simulation Base Test!
UVM INFO ../sv/yapp_tx_monitor.sv(9) @ 0: uvm test top.env.uvc.agent.monitor [yapp_monitor] Executing Monitor Run Phase!
UVM INFO ../sv/yapp_tx_sequencer.sv(21) @ 0: uvm test top.env.uvc.agent.sequencer@yapp_5_packets [yapp_5_packets] raise objection
UVM INFO ../sv/yapp_tx_sequencer.sv(66) @ 0: uvm test top.env.uvc.agent.sequencer@yapp_5_packets [yapp_5_packets] sequence
UVM INFO ../sv/yapp_tx_driver.sv(17) @ 0: uvm test top.env.uvc.agent.driver [yapp_driver] Packet is
-----
Name                               Type                               Size Value
-----
req                               yapp_packet                       - @0300
  addr                           integral 2 'h0
  length                         integral 6 'h9
  payload                        da(integral) 9 -
    [0]                          integral 8 'h2
    [1]                          integral 8 'h7
    [2]                          integral 8 'hc0
    [3]                          integral 8 'h0
    [4]                          integral 8 'h0
    [5]                          integral 8 'h0
    [6]                          integral 8 'h46
    [7]                          integral 8 'h0
    [8]                          integral 8 'h0
  parity                         integral 8 'b1
  parity_type                    parity_type_e 1 GOOD_PARITY
  packet_delay                   integral 32 'h12
  begin_time                     time 64 0
  depth                          int 32 'd2
  parent_sequence (name)         string 24 yapp_5_packets
  
```

Running a Simple Sequence

2. Run a simulation using the `base_test` test class:

Your UVC should now generate and print YAPP packets. Check the correct number of packets are printed and every packet field is printed.



```
Activities Visual Studio Code Jun 25 4:30 PM lab38
```

```
File Edit Selection View Go Run ...
```

```
EXPLORER
```

- LAB38
 - task1_uvc/sv
 - yapp_tx_seqs.sv
 - task1_uvc
 - sv
 - yapp_uvm.sv
 - yapp_packet.sv
 - yapp_pkg.sv
 - yapp_tx_agent.sv
 - yapp_tx_driver.sv
 - yapp_tx_monitor.sv
 - yapp_tx_seqs.sv
 - yapp_tx_sequence.sv
 - tb
 - simulation
 - xcellum.d
 - file.f
 - qlog.txt
 - router_tb.sv
 - router_test_lib.sv
 - top.sv
 - xrun.history
 - xrun.key
 - xrun.log
 - UVM_LAB_38.docx

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS TERSHOLD LOG REPORT TERSHOLD TIMING
```

```
UVM INFO ../sv/yapp_tx_driver.sv(17) @ 40: uvm_test_top.env.uvc.agent.driver [yapp_driver] Packet is
```

Name	Type	Size	Value
req	yapp_packet	-	@3717
addr	integral	2	'h0
length	integral	6	'h16
payload	da(integral)	22	-
[0]	integral	8	'h5c
[1]	integral	8	'h1c
[2]	integral	8	'h6c
[3]	integral	8	'h09
[4]	integral	8	'h72
...
[17]	integral	8	'h08
[18]	integral	8	'h06
[19]	integral	8	'h04
[20]	integral	8	'h0d
[21]	integral	8	'h05
parity	integral	8	'b100101
parity type	parity_type_e	1	GOOD_PARITY
packet delay	integral	32	'h13
begin time	time	64	40
depth	int	32	'd2
parent sequence (name)	string	14	yapp_5_packets
parent sequence (full name)	string	51	uvm_test_top.env.uvc.agent.sequencer.yapp_5_packets
sequencer	string	36	uvm_test_top.env.uvc.agent.sequencer

```
UVM INFO /home/cc/mnt/KCELUM2309/tools/methodology/UVM/CDNS-1.1d/sv/src/base/uvm_objection.svh(1268) @ 50: reporter [TEST_DONE] 'run' phase is ready to proceed to the 'extract' phase
```

```
--- UVM Report catcher Summary ---
```

```
Number of demoted UVM FATAL reports : 0
```

```
Number of demoted UVM ERROR reports : 0
```

```
Ln 16, Col 15 Spaces: 4 UTF-8 LF System Verilog
```

3. Add the following compilation option to the end of you command line:

+SVSEED=random

This sets a random value for the initial randomization seed of the simulation. Re-run the Simulation(**do not recompile**) and you should see different packet data. The simulator reports the actual seed used for each simulation in the simulation log file.

```
Activities Visual Studio Code Jun 25 4:35 PM
```

```
File Edit Selection View Go Run Terminal Help
```

```
EXPLORER
```

- LAB38
 - task1_uvc/sv
 - yapp_tx_seqs.sv
 - task1_uvc
 - sv
 - yapp_uvm.sv
 - yapp_packet.sv
 - yapp_pkg.sv
 - yapp_tx_agent.sv
 - yapp_tx_driver.sv
 - yapp_tx_monitor.sv
 - yapp_tx_seqs.sv
 - yapp_tx_sequence.sv
 - tb
 - simulation
 - xcellum.d
 - file.f
 - qlog.txt
 - router_tb.sv
 - router_test_lib.sv
 - top.sv
 - xrun.history
 - xrun.key
 - xrun.log
 - UVM_LAB_38.docx

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS TERSHOLD LOG REPORT TERSHOLD TIMING
```

```
UVM INFO ../sv/yapp_tx_driver.sv(17) @ 38: uvm_test_top.env.uvc.agent.driver [yapp_driver] Packet is
```

Name	Type	Size	Value
req	yapp_packet	-	@3751
addr	integral	2	'h1
length	integral	6	'h7
payload	da(integral)	15	-
[0]	integral	8	'h42
[1]	integral	8	'h1e
[2]	integral	8	'h0f
[3]	integral	8	'h2
[4]	integral	8	'hdb
...
[10]	integral	8	'h7c
[11]	integral	8	'h21
[12]	integral	8	'h01
[13]	integral	8	'h10
[14]	integral	8	'h22
parity	integral	8	'b1001
parity type	parity_type_e	1	GOOD_PARITY
packet delay	integral	32	'h6
begin time	time	64	38
depth	int	32	'd2
parent sequence (name)	string	14	yapp_5_packets
parent sequence (full name)	string	51	uvm_test_top.env.uvc.agent.sequencer.yapp_5_packets
sequencer	string	36	uvm_test_top.env.uvc.agent.sequencer

```
UVM INFO ../sv/yapp_tx_driver.sv(17) @ 40: uvm_test_top.env.uvc.agent.driver [yapp_driver] Packet is
```

Name	Type	Size	Value
req	yapp_packet	-	@3718
addr	integral	2	'h1
length	integral	6	'h2d
payload	da(integral)	45	-
[0]	integral	8	'h24
[1]	integral	8	'h04
[2]	integral	8	'h26
[3]	integral	8	'h03
[4]	integral	8	'h3c
...
[40]	integral	8	'h23
[41]	integral	8	'h07
[42]	integral	8	'h2b
[43]	integral	8	'h2
[44]	integral	8	'h09
parity	integral	8	'b110100
parity type	parity_type_e	1	GOOD_PARITY
packet delay	integral	32	'h0
begin time	time	64	40
depth	int	32	'd2
parent sequence (name)	string	14	yapp_5_packets

```
Ln 21, Col 15 Spaces: 4 UTF-8 LF System Verilog
```

4. Run a simulation with `base_test` and check which `start_of_simulation_phase()` method was called first. Which is called last?



Driver is called first; base test is called last.

Why? You will need to set the right `+UVM_VERBOSITY` option to see the phase method messages.

```
UVM INFO ../sv/yapp_tx_driver.sv(23) @ 0: uvm_test_top.env.uvc.agent.driver [yapp_driver] Running Simulation Driver!
UVM INFO ../sv/yapp_tx_monitor.sv(14) @ 0: uvm_test_top.env.uvc.agent.monitor [yapp_monitor] Running Simulation Monitor!
UVM INFO ../sv/yapp_tx_sequencer.sv(10) @ 0: uvm_test_top.env.uvc.agent.sequencer [yapp_sequencer] Running Simulation Sequencer!
UVM INFO ../sv/yapp_tx_agent.sv(31) @ 0: uvm_test_top.env.uvc.agent [yapp_agent] Running Simulation Agent!
UVM INFO ../sv/yapp_env.sv(17) @ 0: uvm_test_top.env.uvc [yapp_env] Running Simulation Environment!
UVM INFO router_tb.sv(21) @ 0: uvm_test_top.env [router_tb] Running Simulation Router Tb!
UVM INFO router_test_lib.sv(25) @ 0: uvm_test_top [base_test] Running Simulation Base Test!
UVM INFO ../sv/yapp_tx_monitor.sv(9) @ 0: uvm_test_top.env.uvc.agent.monitor [yapp_monitor] Executing Monitor Run Phase!
UVM INFO ../sv/yapp_tx_seqs.sv(21) @ 0: uvm_test_top.env.uvc.agent.sequencer@yapp_5_packets [yapp_5_packets] raise objection
UVM INFO ../sv/yapp_tx_seqs.sv(66) @ 0: uvm_test_top.env.uvc.agent.sequencer@yapp_5_packets [yapp_5_packets] Executing yapp_5_packets sequence
UVM INFO ../sv/yapp_tx_driver.sv(17) @ 0: uvm_test_top.env.uvc.agent.driver [yapp_driver] Packet is
```

Task 2: Using Factories

Using the Factory

1. Create a new short packet test as follows:
 - a. Define a new packet type, `short_yapp_packet`, which extends from `yapp_packet`. Add this subclass definition to the end of your `sv/yapp_packet.sv` file.
 - b. Add an object constructor and utility macro.
 - c. Add a constraint in `short_yapp_packet` to limit packet length to less than 15.
 - d. Add a constraint in `short_yapp_packet` to exclude an address value of 2.
 - e. Define a new test, `short_packet_test`, in the file `router_test_lib.sv`. Extend this from `base_test`.
 - f. In the `build_phase()` method of `short_packet_test`, use a `set_type_override` method to change the packet type to `short_yapp_packet`.
 - g. Run the simulation using the new test, (`+UVM_TESTNAME=short_packet_test`), and check the correct packet type is created.



```
UVM INFO ../sv/yapp_tx_driver.sv(17) @ 30: uvm_test_top.env.uvc.agent.driver [yapp_driver] Packet is
-----
Name      Type      Size  Value
-----
req       short packet  -    @3787
addr      integral    2     'h0
length    integral    6     'hd
payload   da(integral) 13    -
[0]       integral    8     'h72
[1]       integral    8     'h76
[2]       integral    8     'hd4
[3]       integral    8     'h76
[4]       integral    8     'hea
...
[8]       integral    8     'h6a
[9]       integral    8     'heb
[10]      integral    8     'hbl
[11]      integral    8     'h59
[12]      integral    8     'he2
parity    integral    8     'b11011101
parity_type parity_type_e 1     0000_PARITY
packet_delay integral    32    'hd
begin time time        64    30
depth     int         32     02
parent sequence (name) string      14    yapp 5 packets
parent sequence (full name) string      51    uvm_test_top.env.uvc.agent.sequencer.yapp 5 packets
sequencer string      36    uvm_test_top.env.uvc.agent.sequencer
-----

UVM INFO ../sv/yapp_tx_driver.sv(17) @ 40: uvm_test_top.env.uvc.agent.driver [yapp_driver] Packet is
-----
Name      Type      Size  Value
-----
req       short packet  -    @3860
addr      integral    2     'h1
length    integral    6     'h7
payload   da(integral) 7     -
[0]       integral    8     'h1e
[1]       integral    8     'h1e
[2]       integral    8     'h98
[3]       integral    8     'h3b
[4]       integral    8     'hfa
[5]       integral    8     'hd6
[6]       integral    8     'hd2
parity    integral    8     'b11111111
parity_type parity_type_e 1     BAD_PARITY
packet_delay integral    32    'h9
begin time time        64    40
depth     int         32     02
parent sequence (name) string      14    yapp 5 packets
parent sequence (full name) string      51    uvm_test_top.env.uvc.agent.sequencer.yapp 5 packets
sequencer string      36    uvm_test_top.env.uvc.agent.sequencer
-----

UVM INFO ../sv/yapp_tx_seqs.sv(35) @ 50: uvm_test_top.env.uvc.agent.sequencer@yapp 5 packets [yapp 5 packets] drop objection
UVM INFO /home/cc/mtt/XCELLIUM2389/tools/methodology/UVM/CDMS-1.1d/sv/src/base/uvm_objection.svh(1268) @ 50: reporter [TEST DONE] 'run' phase is ready to proceed to the 'extract' phase
-----
La 77, Col 1, Spaces: 4, UTF-8, LF, System Verilog
```

2. Create a new configuration test in the file `router_test_lib.sv`.
 - a. Define a new test, `set_config_test`, which extends from `base_test`.
 - b. In the `build_phase()` method, use a configuration method to set the `is_active` property of the YAPP TX agent to `UVM_PASSIVE`. Remember to call the configuration method before building the `yapp_env` instance.
 - c. Run a simulation using the `set_config_test` test class (`UVM_TESTNAME=set_config_test`) and check the topology print to ensure your design is correctly configured.

d. You should get a configuration usage report from `check_config_usage()`. Why do you get this?

Answer: Because we have set the `is_active` property of the YAPP TX agent to `UVM_PASSIVE`.

Although the configuration report maybe expected, it is good practice to minimize the number of reports where possible.

Edit your test classes so that no configuration mismatch messages are reported, but all tests still work as required. Check your changes in simulation.



```
uvm_test_top      set config_test      - @2638
env               router_tb          - @2719
uvc               yapp_env            - @2694
agent             yapp_agent          - @2708
monitor           yapp_monitor        - @2823
recording_detail  integral            32 'd1
is_active         uvm_active_passive_enum 1 UVM PASSIVE
recording_detail  integral            32 'd1
recording_detail  integral            32 'd1
recording_detail  integral            32 'd1
recording_detail  integral            32 'd1
-----
UVM INFO ../sv/yapp_tx_monitor.sv(14) @ 0: uvm_test_top.env.uvc.agent.monitor [yapp_monitor] Running Simulation Monitor!
UVM INFO ../sv/yapp_tx_agent.sv(11) @ 0: uvm_test_top.env.uvc.agent [yapp_agent] Running Simulation Agent!
UVM INFO ../sv/yapp_env.sv(17) @ 0: uvm_test_top.env.uvc [yapp_env] Running Simulation Environment!
UVM INFO router_tb.sv(21) @ 0: uvm_test_top.env [router_tb] Running Simulation Router Tb!
UVM INFO router_test_lib.sv(27) @ 0: uvm_test_top [set config_test] Running Simulation Base Test!
UVM INFO ../sv/yapp_tx_monitor.sv(9) @ 0: uvm_test_top.env.uvc.agent.monitor [yapp_monitor] Executing Monitor Run Phase!
UVM INFO @ 0: uvm_test_top [CFGARD] ::: The following resources have at least one write and no reads :::
default sequence [/uvm_test_top/env/uvc/agent/sequence/run_phases/] : (uvm_pkg::uvm_object_wrapper) {}
-
-----
uvm_test_top reads: 0 @ 0 writes: 1 @ 0

--- UVM Report catcher Summary ---

Number of demoted UVM FATAL reports : 0
Number of demoted UVM ERROR reports : 0
Number of demoted UVM WARNING reports : 0
Number of caught UVM FATAL reports : 0
Number of caught UVM ERROR reports : 0
Number of caught UVM WARNING reports : 0

--- UVM Report Summary ---

** Report counts by severity
UVM INFO : 11
UVM WARNING : 0
UVM ERROR : 0
UVM FATAL : 0
** Report counts by id
[CFGARD] 1
[RNTST] 1
[UVRTP] 1
[router_tb] 2
[set config_test] 2
[yapp_agent] 1
[yapp_env] 1
[yapp_monitor] 2
Simulation complete via $finish(1) at time 0 FS + 179
/home/cc/mnt/ACELIUM2309/tools/methodology/UVM/CDMS-1.1d/sv/src/base/uvm_root.svh:457 $finish;
xcellium: exit
TOOL: xrun(64) 23.09-s006: Exiting on Jun 25, 2025 at 18:12:06 PKT (total: 00:00:00)
[cc@ncdc-0053 tb]$
```

THE END





