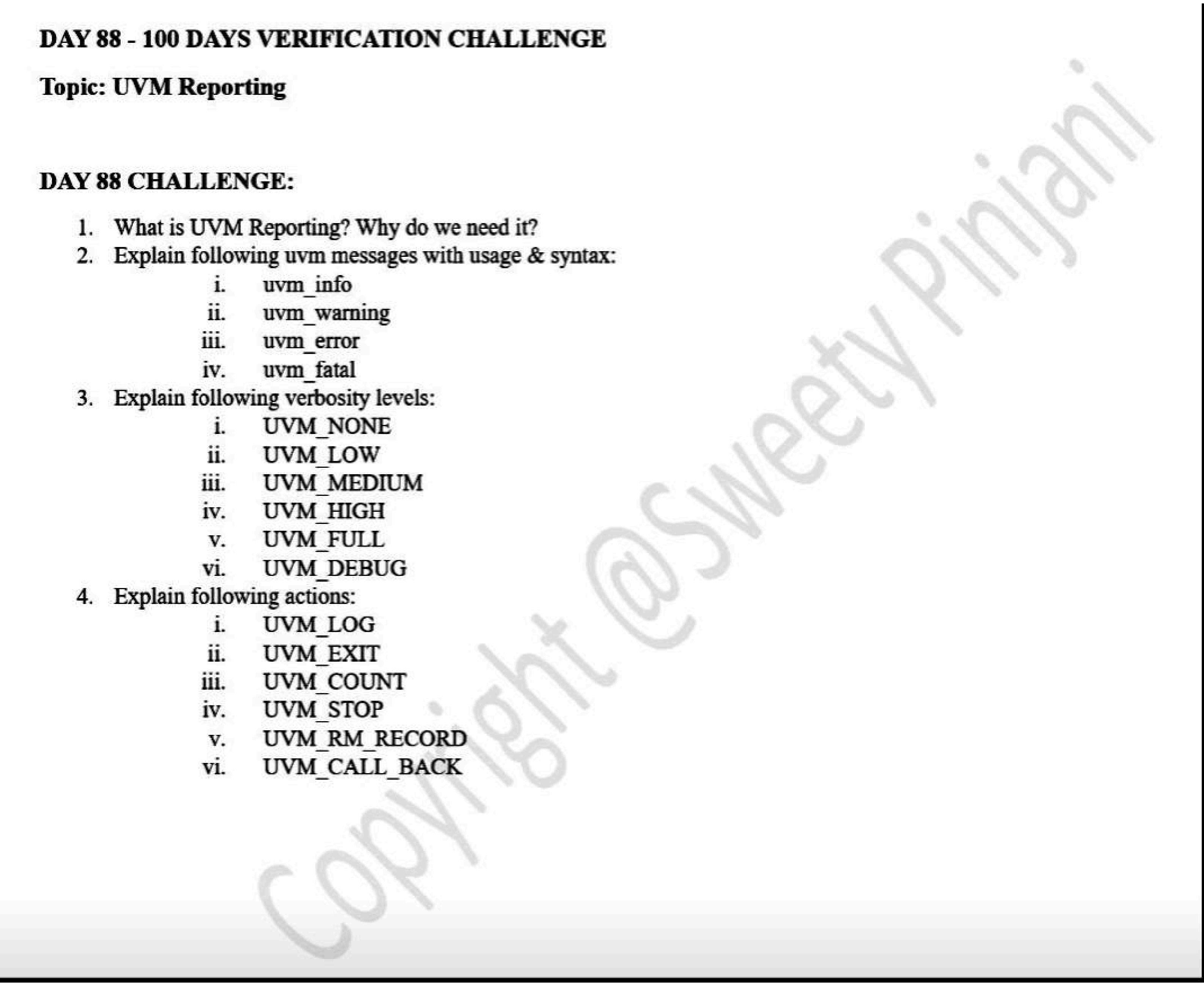

DAY 88 - 100 DAYS VERIFICATION CHALLENGE

Topic: UVM Reporting

DAY 88 CHALLENGE:

1. What is UVM Reporting? Why do we need it?
2. Explain following uvm messages with usage & syntax:
   i. uvm_info
   ii. uvm_warning
   iii. uvm_error
   iv. uvm_fatal
3. Explain following verbosity levels:
   i. UVM_NONE
   ii. UVM_LOW
   iii. UVM_MEDIUM
   iv. UVM_HIGH
   v. UVM_FULL
   vi. UVM_DEBUG
4. Explain following actions:
   i. UVM_LOG
   ii. UVM_EXIT
   iii. UVM_COUNT
   iv. UVM_STOP
   v. UVM_RM_RECORD
   vi. UVM_CALL_BACK

antityagi_uvm2024

DAY : 28

Topic : uvm Reporting

sol^n ① what is uvm Reporting ? why do we need it ?

- uvm Reporting or messaging has a rich set of message- display commands and methods to alter the numbers & types of messages that are displayed without recompilation of the design.
- it helps to tracks the progress of verification, identify issue or bugs and ensure the design meets all requirements before its put into production.

sol^n ② explain following uvm messages with usages & syntax :

i> uvm_info : This is used to provide informational massages during simulation, its typically used for non-fatal information that.

'uvm_info (ID, messages, verbosity)

// blue collor // used to represent informative message //uvm_Display

ii.> uvm_warning : it is used to report warning messages. warning indicate potential issues or unexpected behaviour that do not stop the simulation but should be invested.

'uvm_warning (ID , message)

// yellow colour ; used to represent potential problem //uvm_Display

iii> uvm_error :

it is used to represent /report errors during simulation. when an error occurs it ~~simulation~~ indicates problem that may require immediate attention.

'uvm_error (ID , message) // used to represent real problem // uvm_count .

iv.> uvm_fatal : 'uvm_ fatal (ID , message)

// used to represent problem from which simulation cannot be recovered . simulation stop and $finish executed in # delay | uvm_exit.

Soln ③ **Explain following verbosity levels:**

- **uvm_NONE :** No message are displayed at this verbosity level. It Suppresses all uvm message.

  `uvm_info ("STATUS", "TEST PASSES", UVM_NONE)

- **uvm_LOW :** low level information is displayes. This may include basic progress updates or noncritical information messages.
- **uvm_MEDIUM :** medium level verbosity provides more detailed information than low verbosity.
- **uvm_HIGH :** High verbosity include even more detailed information than medium verbosity.
- **uvm_full :** full verbosity provides the most detailed information during simulation.
- **uvm_DEBUG :** Debug verbosity is used for very detailed debugging information. It includes messages specifically intended for debugging purposes.

∴ **uvm Verbosity levels:**

- uvm_verbosity leves, applicable to uvm_info messages are crucial tool for printing message output.
- This Control is essential coz an excessive number of printed line slow down simulations and make debugging more challenging.

UVM_NONE	= 0;
UVM_LOW	= 100;
UVM_MEDIUM	= 200;
UVM_HIGH	= 300;
UVM_FULL	= 400;
UVM_DEBUG	= 500;

increasing verbosity ↓

- By Controlling message printing with verbosity level you can reduces the number of lines written to the log file during simulation.
- Setting verbosity UVM_NONE minimize the log output, while DEBUG maximizes it.

sol^n 4) Explain following actions:

i.> UVM_LOG : The report is written to a file for the specified severity and ID pair.

Syntax : +uvm_set_action = Component name, id, severity, action

ii.> UVM_EXIT : • it immeeleately terminates the simulation.
• it is typically used in respons to a critical error or when a simulation condition requres an early termination.

iii.> UVM_COUNT : it Count the number of reports with the COUNT attribute. when this count reaches a certain threshold (max quit count), the simulation terminate

iv.> UVM_STOP : Causes $stop to be executed, putting the simulation into intaactive mode.

v.> UVM_RM_RECORD:
• Abstract class which defines the recorder API.

vi.> UVM_CALL_BACK : it is used to invoke a callback function or method at a specific point during the verification process.

- The uvm_action is an enum type that encompasses all possible values for report actions in UVM.
- Each report is configured to execute one or more actions which are determined through a Bitwise OR operation on the following enumeration constants.
- it can be set using the Command-line processor in the format:

+uvm_set_action = Component name, id, Severity, action

eg:-

+uvm_set_action => uvm_test_top.env.agent.drv._All_.UVM_INFO.
UVM_DISPLAY|UVM_COUNT.