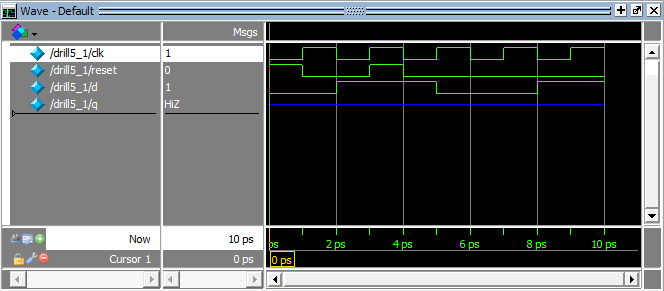
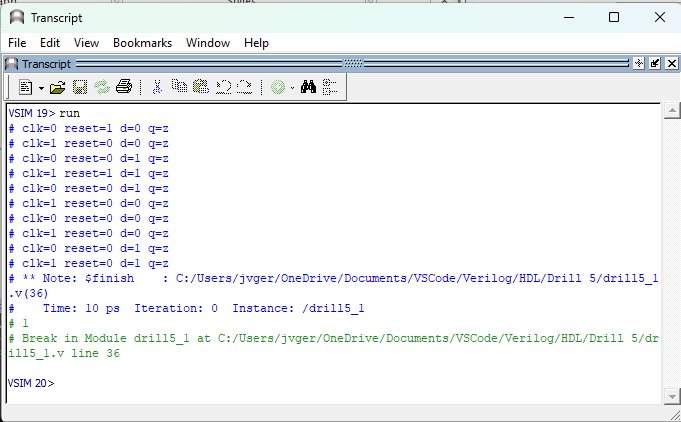
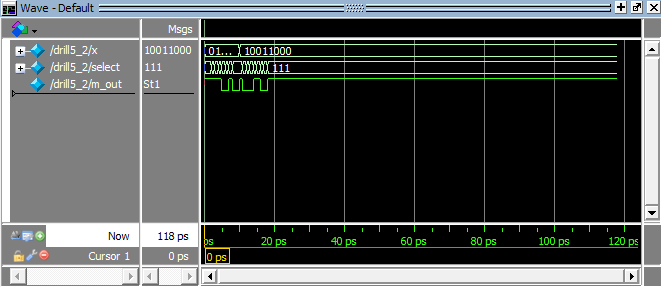
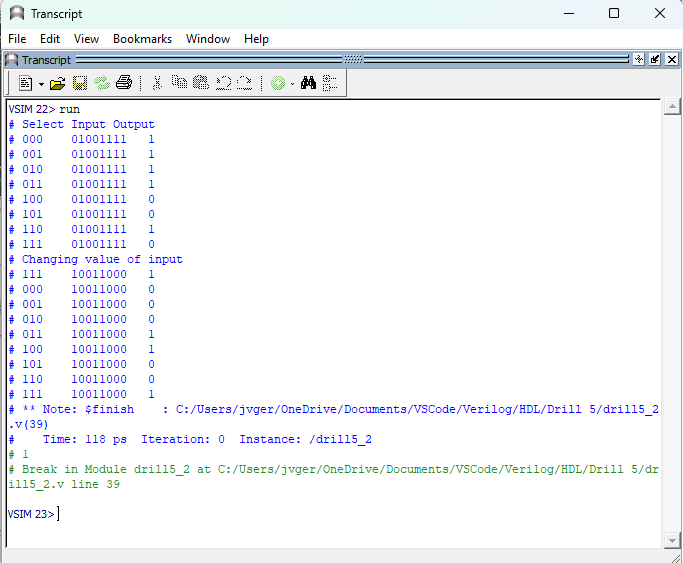
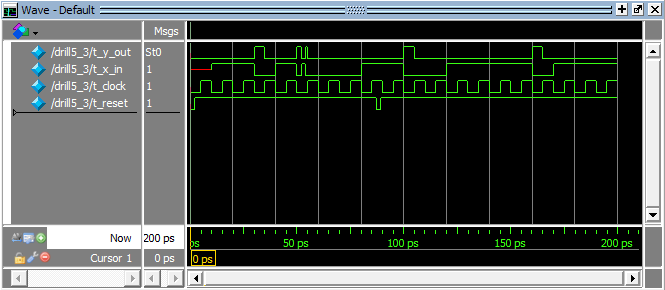
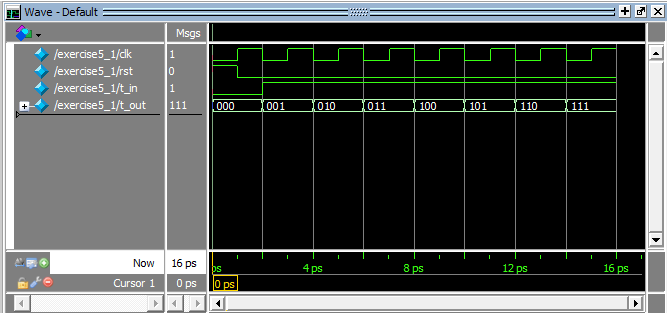
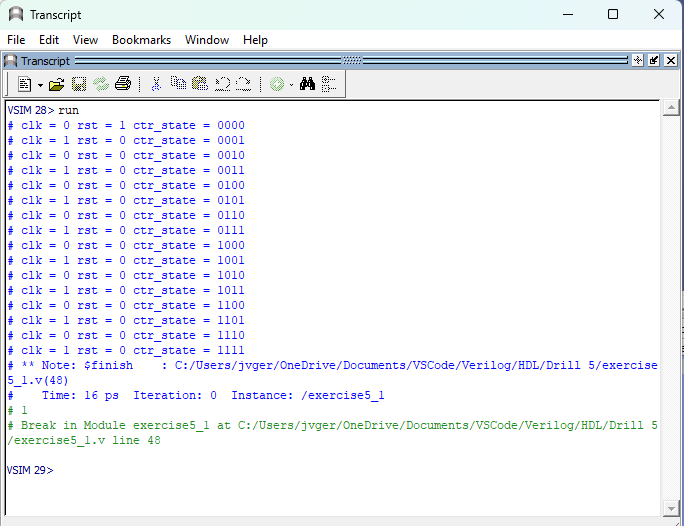
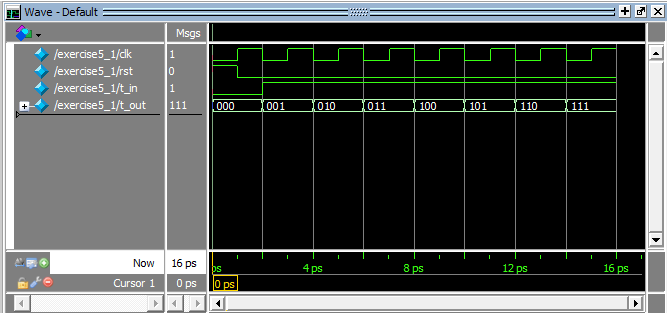
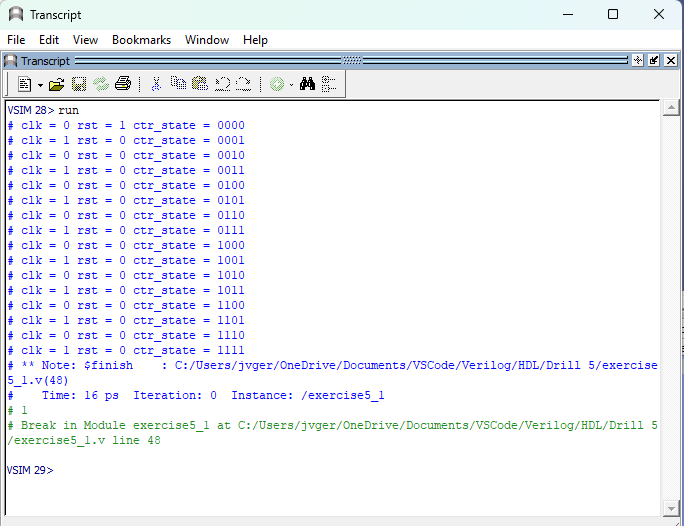
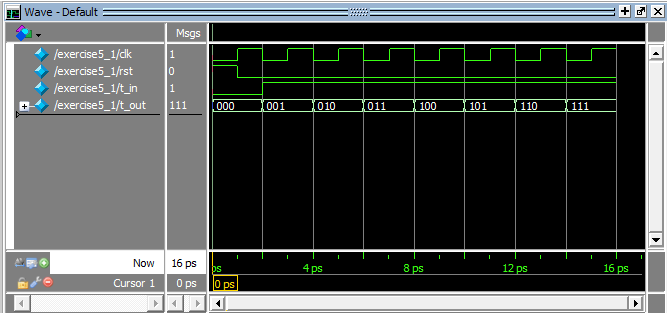
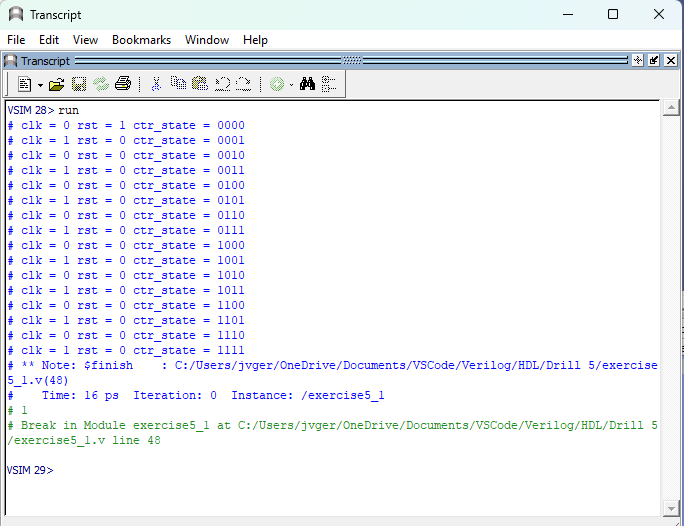
DRILL 5

* **Drill5\_1.v**
  + Testbench
  + Transcript
* **Drill5\_2.v**
  + Testbench
  + Transcript
* **Drill5\_3.v**
  + Testbench
  + Transcript



* **Exercise5\_1.v**
  + Testbench
  + Transcript
* **Exercise5\_2.v**
  + Testbench
  + Transcript
* **Exercise5\_1.v**
  + Testbench
  + Transcript

**Review Questions**

1. **Can you implement UDP for circuits with multiple outputs? If yes, how?**

*Yes you can, to enable UDP with multiple outputs, you can achieve this by generating individual UDPs for each output with its unique identification, and then integrating all the UDPs into the module that will utilize the UDP***.**

1. **Why do you need to declare the output of sequential UDPs as reg?**

*Sequential UDP outputs are typically denoted as "reg" to indicate the presence of a state. The output values of a UDP always correspond to its internal state and remain consistent.*

1. **What are the advantages and disadvantages of using gate-level models?**

*The tables for combinational and sequential UDP are different from each other. The initial statement is used for initialization of sequential UDPs. Here's an example:*

*primitive udp\_initial(a, b, c):*

*output x;*

*input a, b;*

*reg x;*

*initial x = 1'b1 // optional value of x at the start of the simulation*

*table*

*//truth table or UDP behavior*

*endtable*

*endprimitive*

1. ***Explain the entry below:***

***// Clk Clr Data : Q(state) : Q(next)***

***(?0) 1 b : 0 : - ;***

*On this entry, after the data is inserted, any entry from the current state will be change to 0 then changes again with dash.*