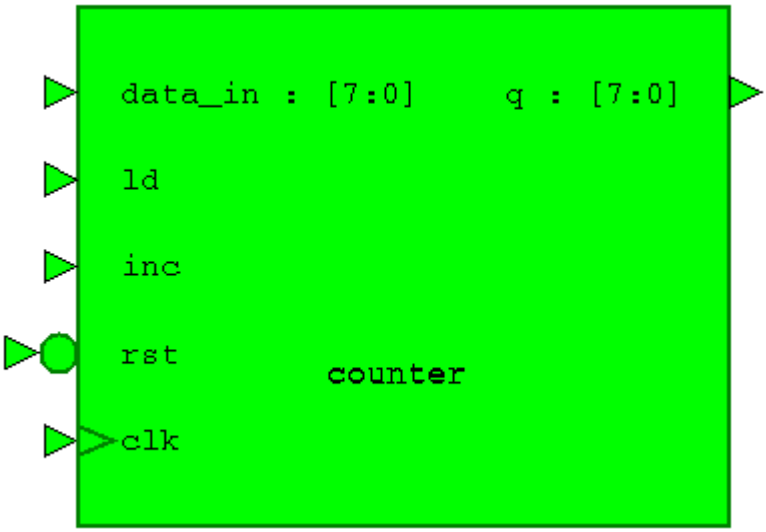
Project 2: Writing a SystemVerilog Test Bench (100 Points)

In this Project, you have been handed a device and a specification.

The Device under Test

The device under test (DUT) is a simple 8-bit counter. It has the following symbols and signal behavior:



- `clk` – The counter clock.
- `rst` – A synchronous reset
- `data_in[7:0]` – An 8-bit bus input
- `q [7:0]` – An 8-bit bus output
- `inc` – `q` moves to 1 `q`+1 on the next clock cycle if `inc` is 1. If q is `8'hFF` it wraps around to `8'h00`.
- `ld` – `q` is set equal to `data_in` if `ld` is 1. `ld` takes precedence over `inc`

Your task

1. Make a verification plan for the device
2. Make sure that the device matches the specification using a SystemVerilog test bench. The test bench has been assembled for you. You just need to provide the testing functionality.

Test Bench Assignment

- Create a test bench that proves that this design works as expected. You should be able to demonstrate that the design works to a third party.
- The test bench has already been wired up and there is a compile script in place. There are three files in the lab directory:
- run.do – This script compiles the DUT and the test bench.
- DUT.sv – This SystemVerilog file contains the device under test. We are treating this file as a black box, so it is not human readable.
- top.sv – This is the test bench. It instantiates the DUT.

You will create your test bench in top.sv. The top.sv file looks like this:

```
module top;
   logic [7:0] data_in;
   bit         clk, inc, ld, rst;
   wire [7:0]  q;


   counter DUT(.data_in(data_in),.q(q),.clk(clk),.inc(inc),.ld(ld),.rst(rst));


endmodule // top
```

The DUT is instantiated, and you can assign values to the `clk`, `inc`, `ld`, `rst`, and `data_in` signals using any combination of `always` and `initial` blocks that will test the design.

This design compiles and simulates in ModelSim or Questa using the `run.do` script. Use this test bench to demonstrate that the design works as described.