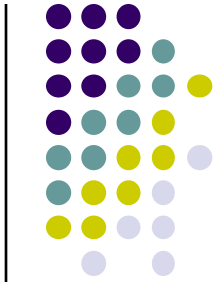
Lab 01
Ping-Pong Counter

Due on Sep 29th 2016 11:59pm

黃稚存

Department of Computer Science
National Tsing Hua University

Description



- This is a simple Verilog practice to make sure that you can manipulate the HDL well.
- Design a 4-bit Ping-Pong counter, which counts up from 4'b0000 to 4'b1111 by one clock cycle at a time.
- After reaching 4'b1111, the counter goes to 4'b1110, and down to 4'b0000.
- Then it counts up again periodically.

One Counting Cycle of Ping-Pong Counter

- The timing diagram is given as the spec

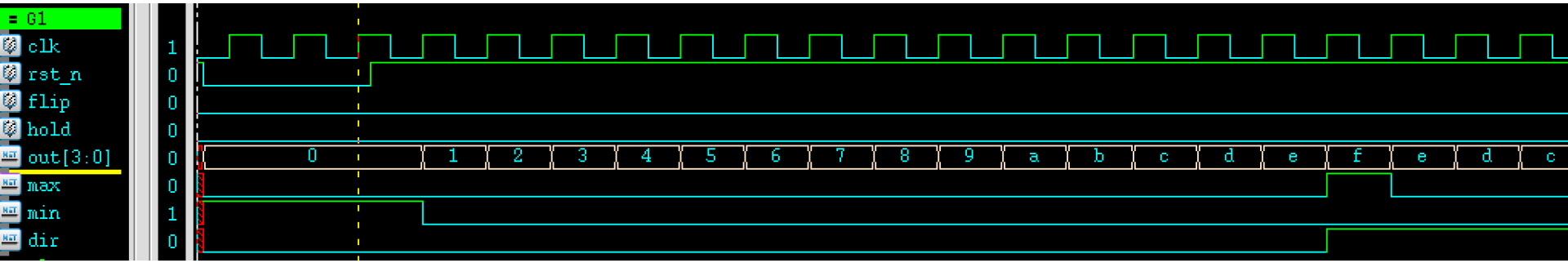
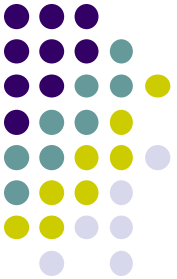


Fig 1-1: A counting cycle.

Primary Inputs and Outputs (1/5)



- clk (1-bit input): Clock.
  - Every primary output is synchronized to the positive clock edge. See Fig. 1-1.
- rst_n (1-bit input): Reset.
  - Negative-edge-triggered reset. See Fig. 1-2.

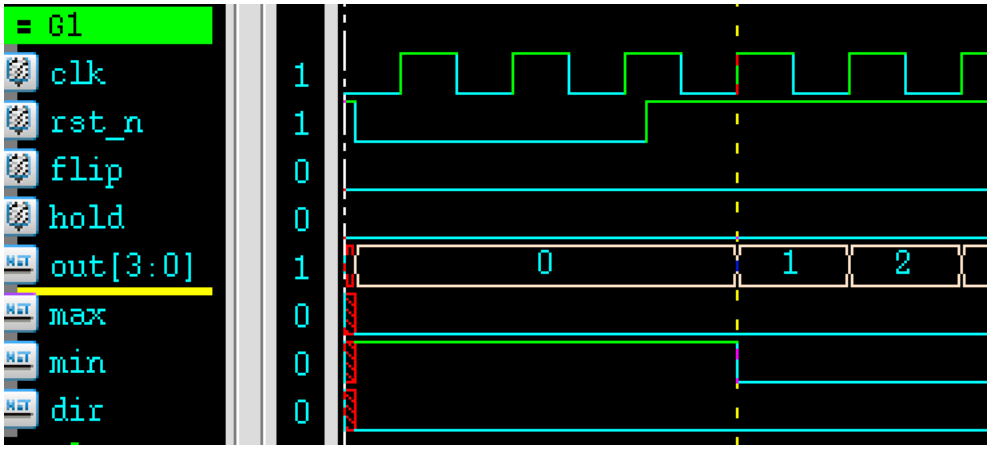


Fig 1-2: Example for the reset.

Primary Inputs and Outputs (2/5)

- out (4-bit output):
  - The output of the ping-pong counter. This signal is initially 4'b0. See Figs. 1-1 and 1-2.
- hold (1-bit input):
  - When asserted high, the output will hold still. Otherwise, the counter continues in its normal way. See Fig. 1-3.

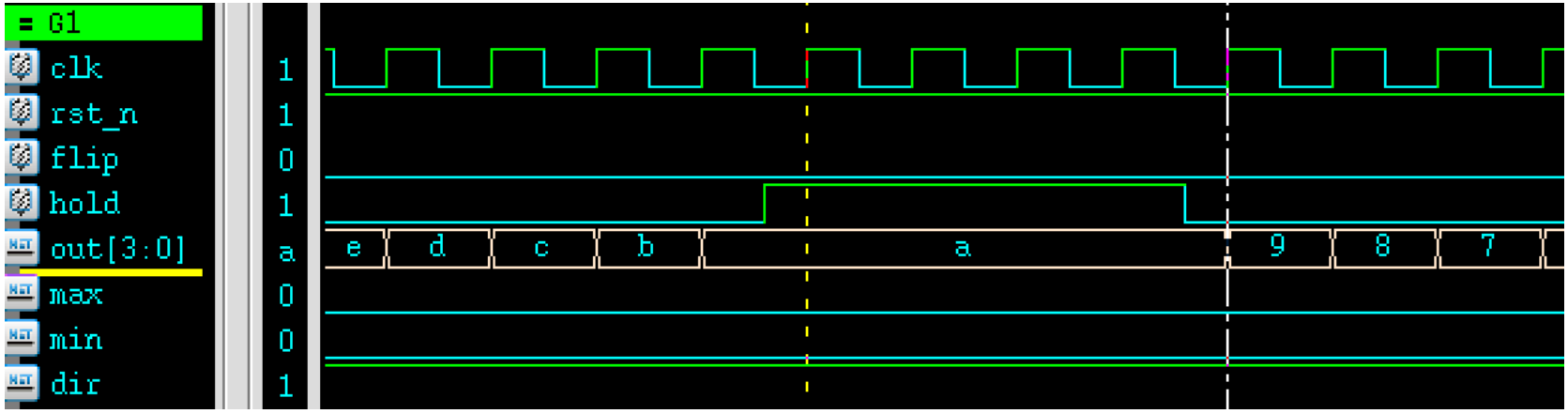
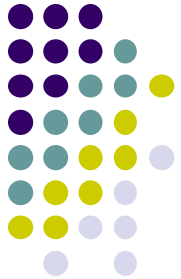


Fig 1-3: Example for hold signal.

Primary Inputs and Outputs (3/5)



- flip (1-bit input):
  - When asserted high, the counter will change its counting direction after the successive positive clock edge. See Fig. 1-4.

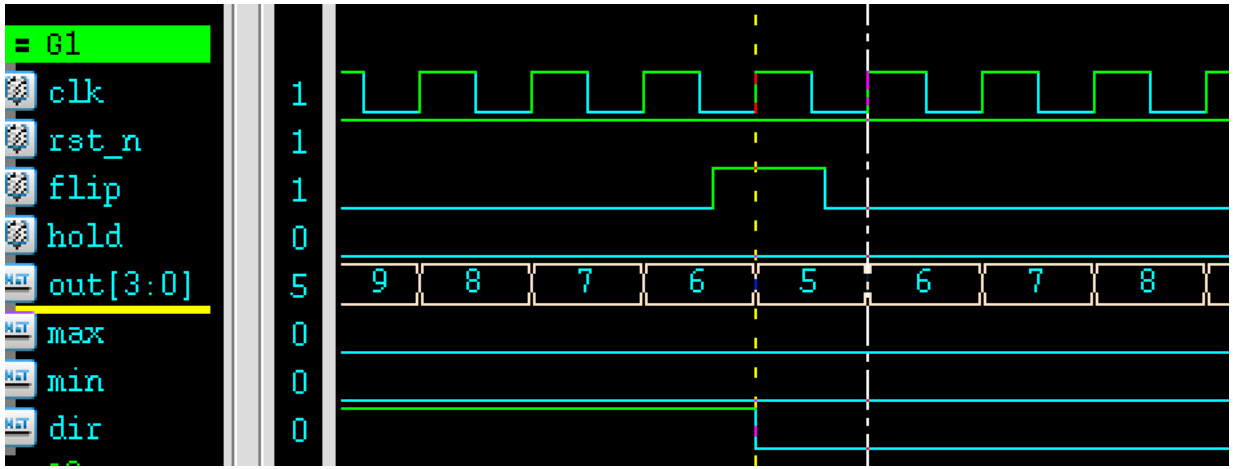


Fig 1-4: Example for flip signal.

Primary Inputs and Outputs (4/5)

- The signal flip should be asserted for one effective clock cycle, and then deasserted. Otherwise, the output will toggles back and forth. See Fig. 1-5.

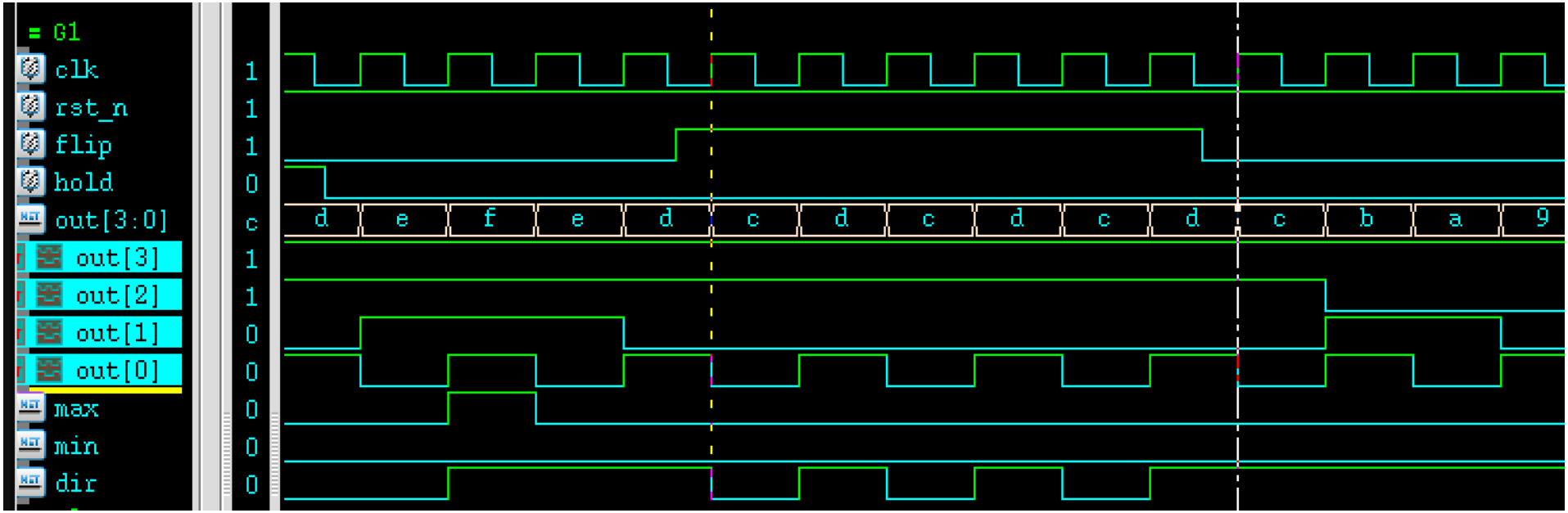
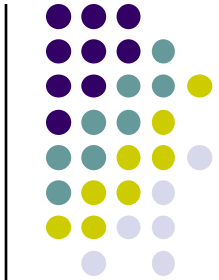


Fig. 1-5: Example for flip signal with multiple cycles asserted.

Primary Inputs and Outputs (5/5)



- Also implement three additional output signals to indicate the status of the counter. You can also take a look at the figures above for the reference. See Figs. 1-1 to 1-5.
  - dir (1-bit output): When it is low (0), the design is counting up; and when it becomes high (1), the design is counting down.
    - We may specifically define that dir=0 when the output reaches its minimal value; dir=1 when the output reaches its maximal value.
  - max (1-bit output): The max is high when the output counts to its maximum value, and low otherwise.
  - min (1-bit output): The min is high when the output counts to its minimum value, and low otherwise. This signal is initially high.

Design Procedure

- Start with spec, primary IOs, and block diagram
- Design the overall architecture
- Specify finite state machine(s), data path, and internal signals
- List the timing diagrams of major operations
- Verilog coding
- Prepare comprehensive test environment

Requirements

- Extend the design to a 5-bit ping-pong counter.
  - Complete the RTL design in pingpong.v; the test bench in pingpong_test.v.
  - Assume the clock frequency is 100MHz.
- Use ncverilog to simulate the design.
  - Write a pingpong.f to integrate the simulation by -f method.
  - Using a Makefile to integrate the simulation, without using -f method.
- Verify the design by applying every possible combination you can image about, including the conditions in Figs. 1-1 to 1-5.
- Use the text-based debugging mechanism of $display and $monitor tasks. Compare their difference.
- Use the waveform viewer nWave to validate the simulation waveform.
- Verilog directive `include is forbidden in this lab.
- A free-format report has to be delivered to brief
  - the design concept, and
  - simulation patterns and setup.

Guideline of Verilog Design

- Prepare the block diagram and FSM well
  - Before your Verilog coding
- Try to symbolize every condition for
  - State transitions
  - Mode selection for datapath
- FSM: the simpler the better
  - One single state can take multiple cycles
  - States can be nested
- There is no best style for every individual design
  - State arrangement
  - Naming convention