Lab2:
GCD Computation

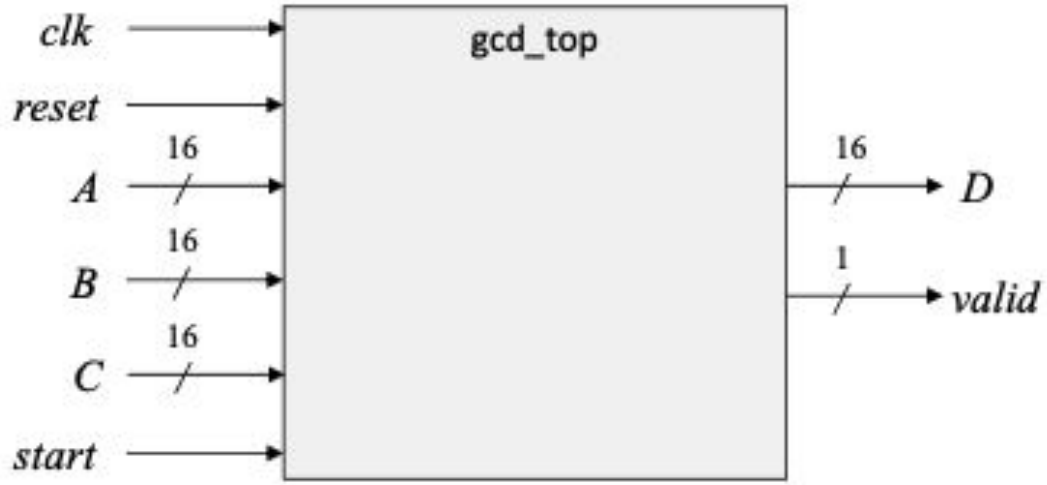
Goal

- Design a sequential circuit that computes the greatest common divisor (GCD) of three 16-bit unsigned numbers.
- Submit your Verilog module to E3 with Student ID (ie. Lab2_112550001.v)
- Deadline: 5/22(WED), 10:00 AM.
- No plagiarism !!
- Please remove all the delays before upload files to E3!

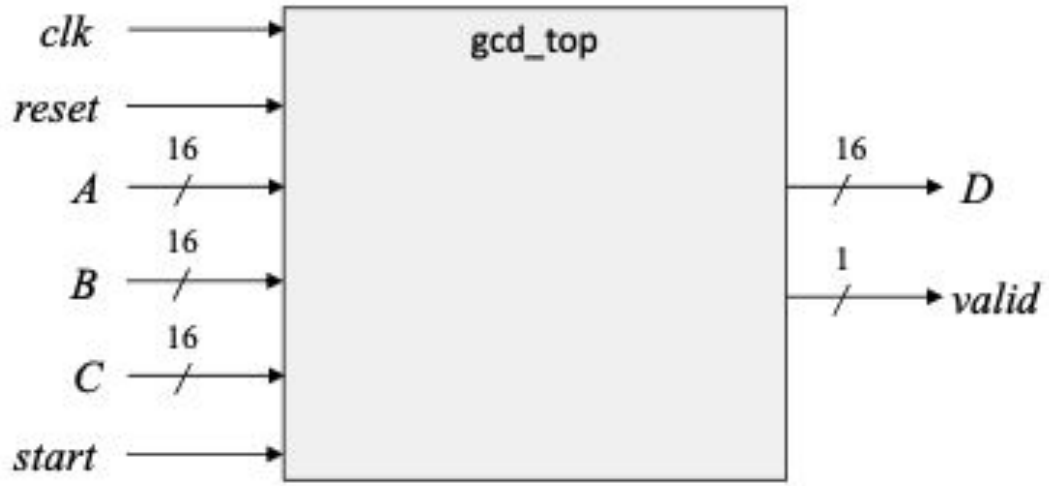
Requirements

Lab 2 Descriptions

- Goal: Design a sequential circuit that computes the greatest common divisor (GCD) of three 16-bit unsigned numbers.
  Your module ‘gcd_top’ should have the following ports:



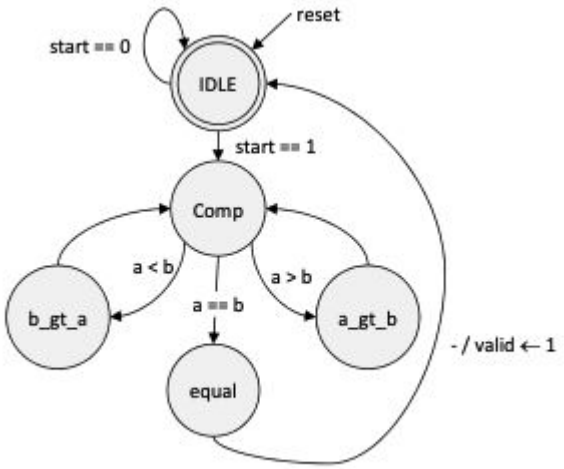
Lab 2 module



- A, B, and C are the input 16-bit nonzero numbers.
- D is the 16-bit GCD of A, B, and C.
- Start is the 1-bit signal that trigger the GCD computation.
- valid is a 1-bit signal that indicates the data D is ready for fetch.

Lab 2 About Your GCD Module

- Your design should have a gcd module that compute the GCD of a and b that contains an FSM
- For the Euclidean algorithm, a two-state FSM is good enough. But you should implement the following FSM, just for practice:



Lab 2 Requirements

- In your design, you must implement the Euclidean algorithm to compute the GCD
- You cannot use any division, modulus, or multiplication operations in your design
- The reset signal should set the circuit state to IDLE and clear all internal registers to zero
- The circuit can be invoked repeatedly
  - If the start signal is activated again before the previous computation is finished (i.e., before the signaling of valid is finished), it shall be ignored

Submission format

Lab2_{student_id}.v

Don’t upload the testbench module to E3!