CO200 - Computer Organization and Architecture

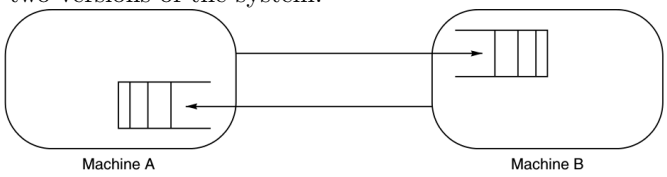
A1 - Hardware Modules in SystemC

Points to Note

- Design and test each of the listed hardware modules in the SystemC language.
- In the comments section of each SystemC code, include a short description of the program, your name, roll number, date of writing the program and other information you deem relevant.
- (a) Report: Every question can have a corresponding answer containing block diagrams/ microarchitecture, brief explanation, other relevant info. (b) Auxilary files to submit: Per question, include the following files along with the report: SystemC code, the testbench including the monitor and stimulus, execution screenshots, VCD dump, gtkwave screenshots. (c) Archive: Organize your submission into directories per question.
- This is a team assignment. One submission per team.
- (a) Submission is due October, 11, midnight. Pack your report, code, screenshots and other files in an archive and mail to `co200.nitk@gmail.com`.
- **Download and install** the latest version of SystemC from Accellera's SystemC page. SystemC is a library in C++ that enables a designer to simulate concurrent processes (hardware blocks can be described as concurrent processes).

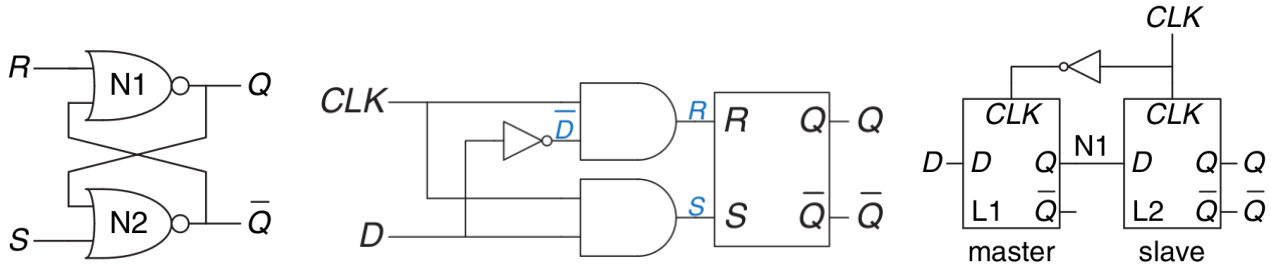
Modules

1. **Hello World Program**. A hello world message should be printed out by a module periodically. The module should also print the current timestamp (in clock cycles). Module terminates after 100 clock cycles.
2. **Encoders and Decoders**. Implement a combinational 4-to-16 decoder and a 16-to-4 encoder.
3. **Half Adder**. Implement a combinational half adder (HA).
4. **Full Adder**. Using the HA module from the previous question, Implement a combinational full adder (FA).
5. **n-bit Adder/Subtractor**. Using the FA module from the previous question, build an n-bit Adder/Subtractor module. Value of 'n' should be configurable. Possible n values are 4,8,...,256.
6. **Basic interconnection network**. Implement a 2 node point to point interconnection network as shown in the following figure. Implement two versions of the system.



   - Version 1. Periodically, both A and B send messages to each other. Each prints out the received message along with the timestamp at the output.
   - Version 2. After random intervals, A sends one message to B. B responds with 4 replies. A prints sent and received messages at the output.

7. **Single bit Sequential Memory Element**. Implement a 1 bit sequential memory cell (a D-Flip Flop). The block diagrams useful for this question follow.



(a) SR latch schematic (b) D latch schematic (c) D flip-flop

Hint: For the block diagrams and design of the D-flip flop, refer Section 3.2 of *Harris and Harris, Digital Design and Computer Architecture, 2e, MK, 2013* or Section B.8 of *Patterson and Hennesy, COD5e.*

8. **Register**. Use the memory cell designed above to implement a 32 bit register. The data input and data output are now 32 bit wide (instead of 1-bit in the case of the memory cell). The reset signal is still 1-bit signal as in the case of the memory cell.