Worcester Polytechnic Institute
Electrical and Computer Engineering Department
Methodologies for System Level Design and Modeling

Online Offering – Fall 2010

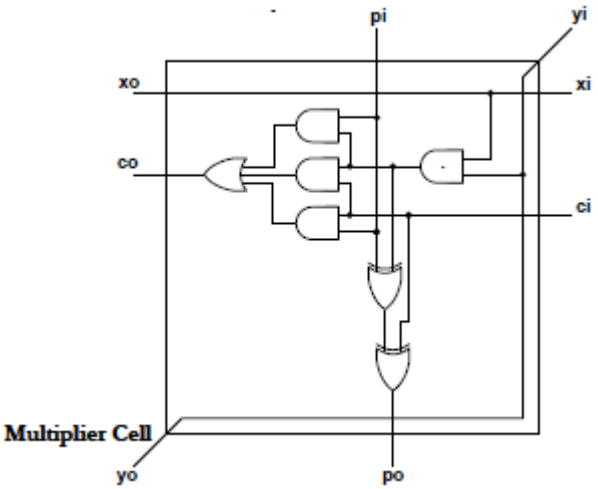
Final Exam – System Level Design, C/C++ to TLM-2.0

Embedded Design

1. Design a wait state insertion logic that performs a specific number of wait states depending on the slave it corresponds to. The wait state insertion logic has a *request* input from the master and an *init* input from the slave. For the initialization the *init* input is issued which leads the count input for the particular slave in the wait state insertion logic counter. During the operation of the wait state insertion logic the master makes its request by asserting its *request* output (wait state insertion logic input). The wait handler responds by asserting its wait request output. This output remains asserted for the duration of the wait state count, when de-asserted the request signal from the master will be de-asserted as well. Show the hardware implementation of wait state insertion logic.

C Coding

2. Using C++ gate models with timing, model a 4*4 array multiplier. The following figure shows the structure of each cell of the multiplier. Generate a testbench to simulate this circuit. Assuming inputs change at zero time. Apply inputs to produce the worst case delay of the circuit.



SystemC Channels

3. The *sc_semaphore* channel can be used for arbitration of a shared device between several master modules. In this problem you will have to model five master modules wanting to access a fictitious device that allows only four simultaneous accesses. For this purpose, you are to use a semaphore of size 4. SystemC descriptions are required. Write five SystemC SC_MODULE master modules that connect to an *sc_semaphore*. Each module requests the use of the fictitious device at random times and uses the device for a random amount of time before it is done with it. Each master module prints the time it requests the use of the device, reports (prints) if it was granted the permission to use the device, and reports the time it releases the device. Write codes for the modules such that there will be times that the semaphore cannot respond to a request because all its tokens are taken. Simulate and write activities to an activity file.

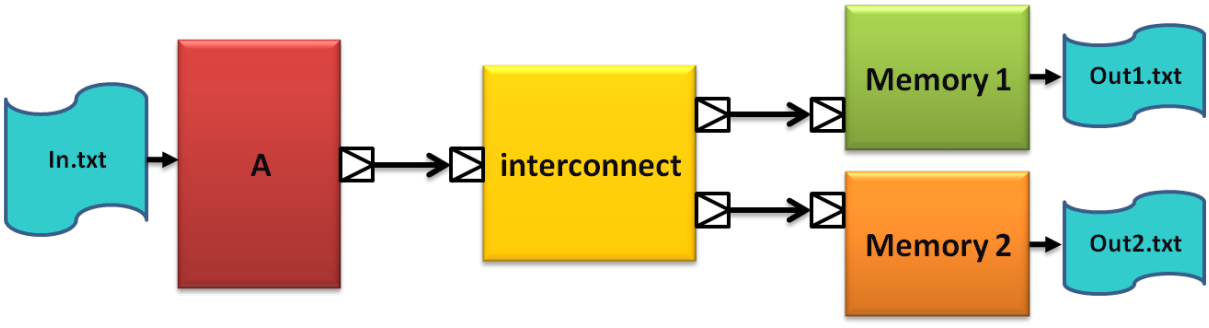
TLM-2.0

4. Design the following system using TLM-2.0 standard. ***A*** reads "in.txt". The format of each line of "in.txt" is as follow:

R/W	Address	Data (in case that the command is W)

According to the address, command and data, ***A*** sets attributes of generic payload and sends it to targets ***Memory 1*** (for odd addresses) and ***Memory 2*** (for even addresses) using *b_transport* interface. If read command is issued, the target modules will write the data into "out1.txt" or "out2.txt" (according to the address) in the following format:

Address	Data



The "in.txt" file is provided.