University of Stuttgart
Germany

Institute of Computer Architecture
and Computer Engineering
Dept. of Embedded Systems

Modeling, Simulation, and Specification

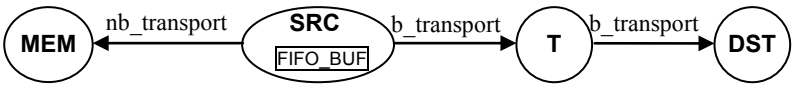
Prof. Dr. Martin Radetzki

Exercise 7: *Transaction Level Modelling with OSCI TLM2 (AT)*

This is a follow up of exercise 6. The source files for this exercise are available in the package exercise7.zip on the course webpage. The exercise was designed using OSCI TLM-2.0.1 or newer, which can be downloaded from www.systemc.org.

1. TLM2 programming style – approximately timed

In directory ex7_1, you are given a 2-stage pipelined MEM model. The pipeline consists of two stages **request** and **response**. You are also given an incomplete SRC model that is required to communicate with the pipelined MEM in TLM2 approximately timed style using non-blocking transport interfaces. In SRC, the data read from MEM is written to a big enough `sc_fifo`. A second thread reads data from the FiFo and sends it to T in the same way as in last exercises loosely timed model (i.e. `b_transport()`). Complete SRC using approximately timed communication to MEM.



a) Complete SRC, enabling approximately timed communication to MEM. Hint: you need to implement one thread in SRC to initiate read transactions to MEM and also the `nb_transport_bw()` function. Compile and run the simulation. What is the simulated time at the end of simulation?

b) Please explain the differences, compared to the results of last exercises loosely timed implementation.

c) Draw the message sequence chart of this scenario for one iteration with indicated simulation time.

2. Using Payload Event Queues in TLM2 AT Models

Copy all C++-Files (*.{h,cpp}) from your solution of problem 1 to the folder ex7_2.

a) Change the `MEM_REQ_ACCEPT_DELAY` defined in mem.cpp to another value, e.g. 2 or 7 ns. What is the simulation time at the end of simulation? Did it meet your expectations? Please explain the difference.

b) Replace the `sc_fifo` in SRC with a payload event queue (**peq**) for data buffering, using a fixed time offset of 1 ns. You may use either a peq_with_get or a peq_with_cb_and_phase. What changes are necessary in the system's other parts to use the callback based event queue? Note: although peqs are usually used in combination with non-blocking interfaces, you can leave the communication from SRC to T and from T to DST as they are (i.e. using blocking interfaces).