TLM 2.0 Loosely Timed (LT) System Example - Temporal Decoupled

Jack Donovan, Anna Keist, Charles Wilson

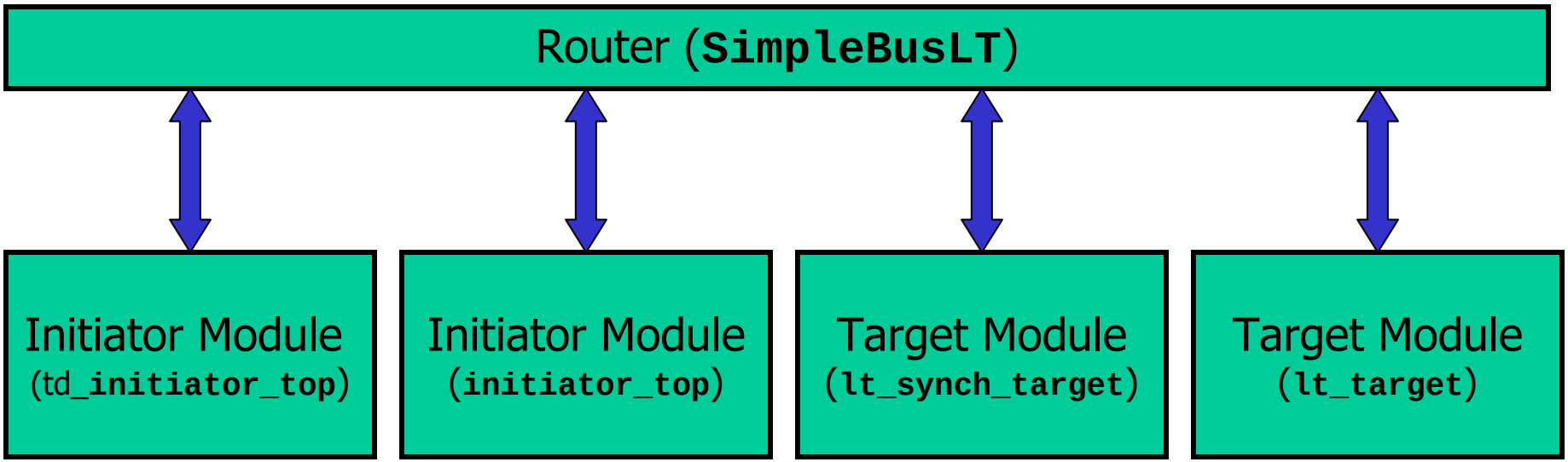
ESLX, Inc.

June 2008

AT System Example - Annotated Timing

- **The Goal is to Illustrate:**
  - **Application of TLM 2.0 in a real system**
  - **Annotated non-blocking (NB) option of the non-blocking style**
    - **NB annotated timing has been referred to as "1 phase"**
    - **Simplest version of non-blocking/AT**
- **Possible Applications:**
  - **Architectural exploration**
  - **Early software development**

Example Block Diagram



TLM 2 GP

How to run this example (Linux)

- Set **`SYSTEMC_HOME`**
- **`cd examples/tlm/lt_temporal_decouple/build-unix`**
- **`make clean`**
- **`make`**
- **`make run`**

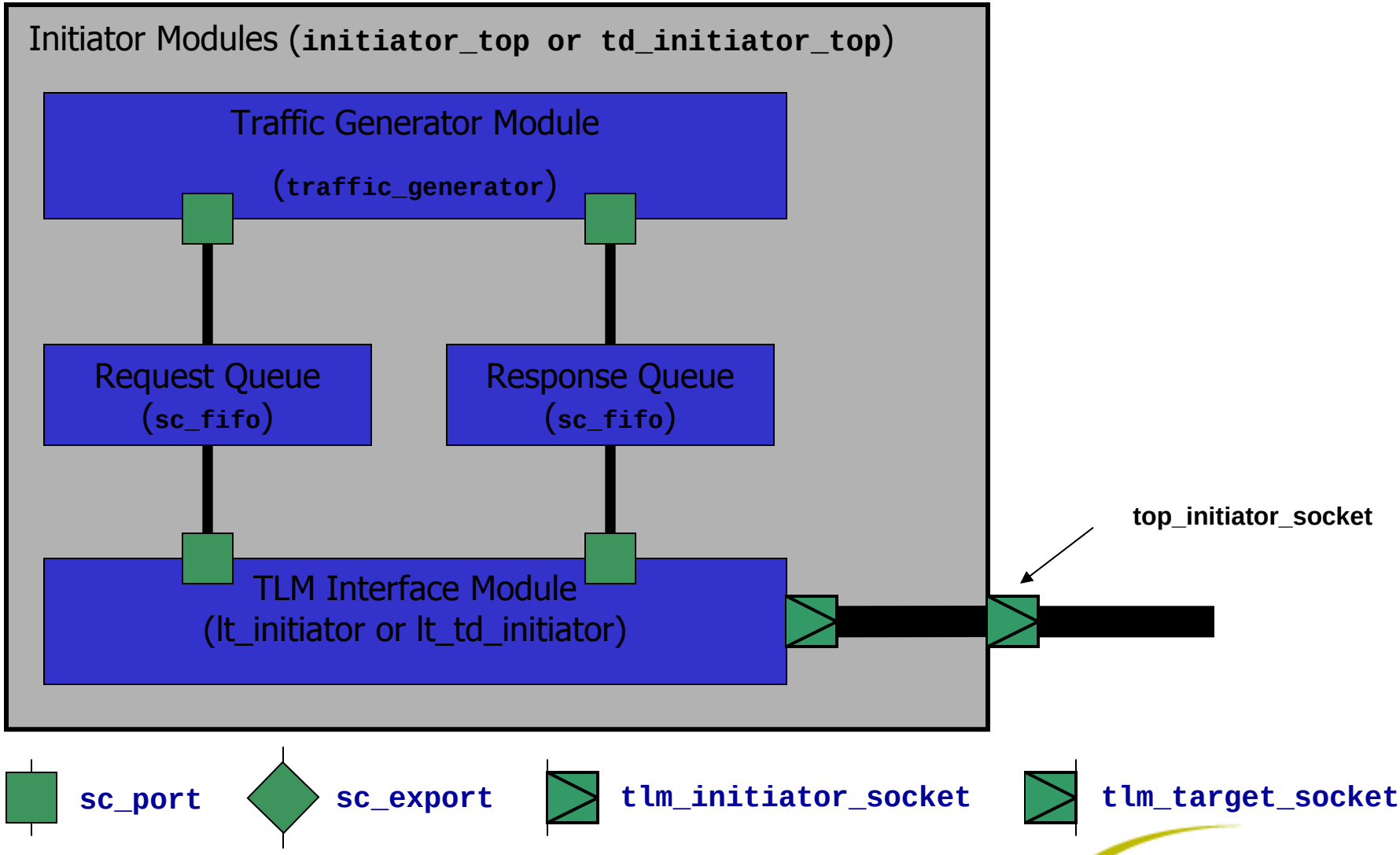
How to run this example (MSVC)

- Open a explorer window on **`examples/tlm/lt_temporal_decouple/build-windows`**
- Launch **`lt.sln`**
- Select **‘Property Manager’** from the **‘View’** menu
- Under **‘lt_temporal_decouple > Debug | Win32’** select **‘systemc’**
- Select **‘Properties’** from the **‘View’** menu
- Select **‘User Macros’** under **‘Common Properties’**
- Update the **‘SYSTEMC’** entry and apply
- Build and run

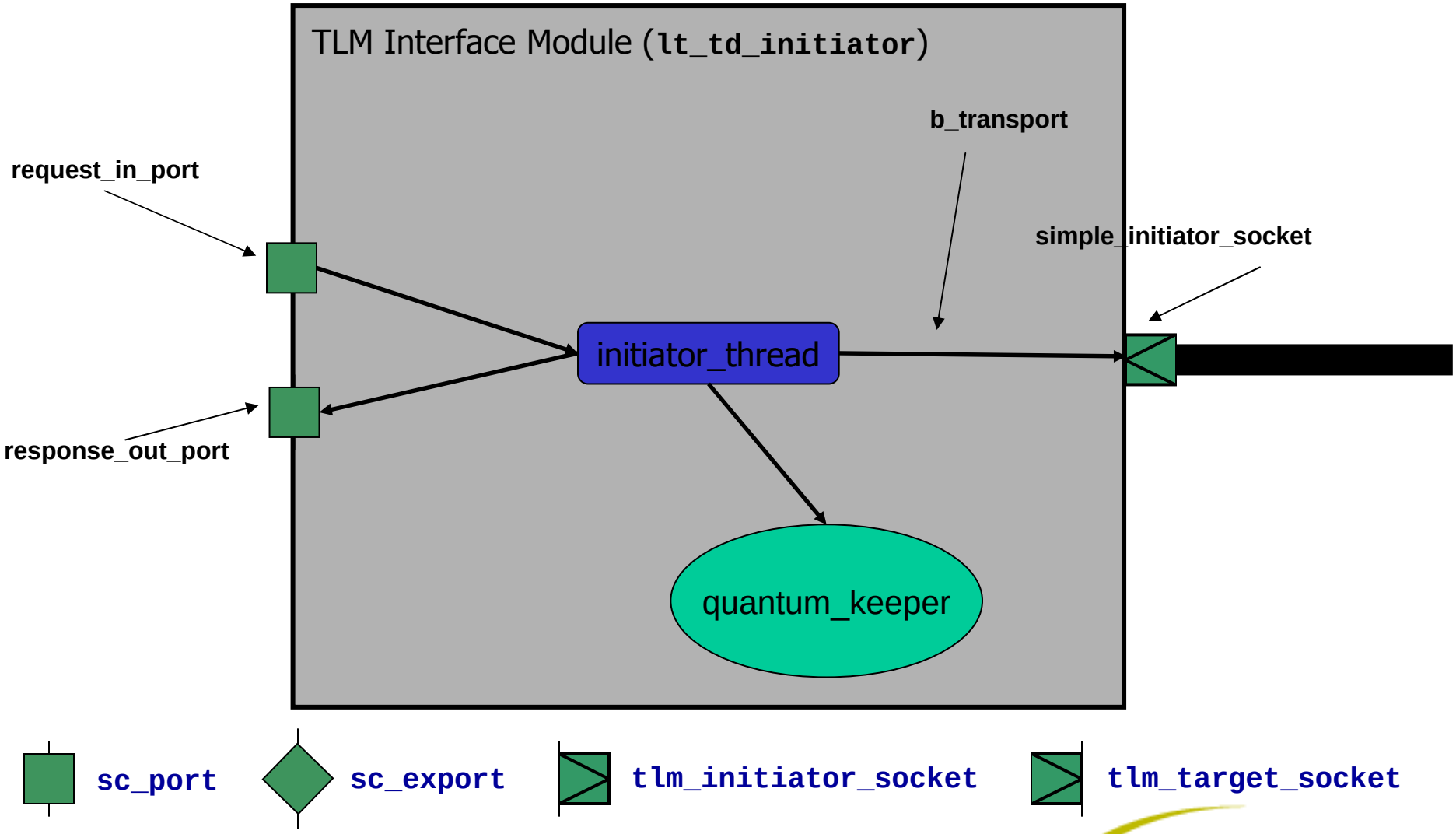
Expected Output (`expected.log`)

```
....
Info: lt_initiator.cpp: 0 s - initiator_thread
      Initiator: 102 b_transport(GP, 0 s)

Info: memory.cpp: 0 s - print
      ID: 201 COMMAND: WRITE Length: 04
      Addr: 0x0000000000000000 Data: 0x00000000

Info: lt_synch_target.cpp: 0 s - custom_b_transport
      Target: 201 Forcing a synch in a temporal decoupled initiator with wait( 80 ns),

Info: lt_synch_target.cpp: 80 ns - custom_b_transport
      Target: 201 return from wait will return a delay of 0 s
....
Info: lt_initiator.cpp: 80 ns - initiator_thread
      Initiator: 102 b_transport returned delay = 0 s
```

Initiator Module

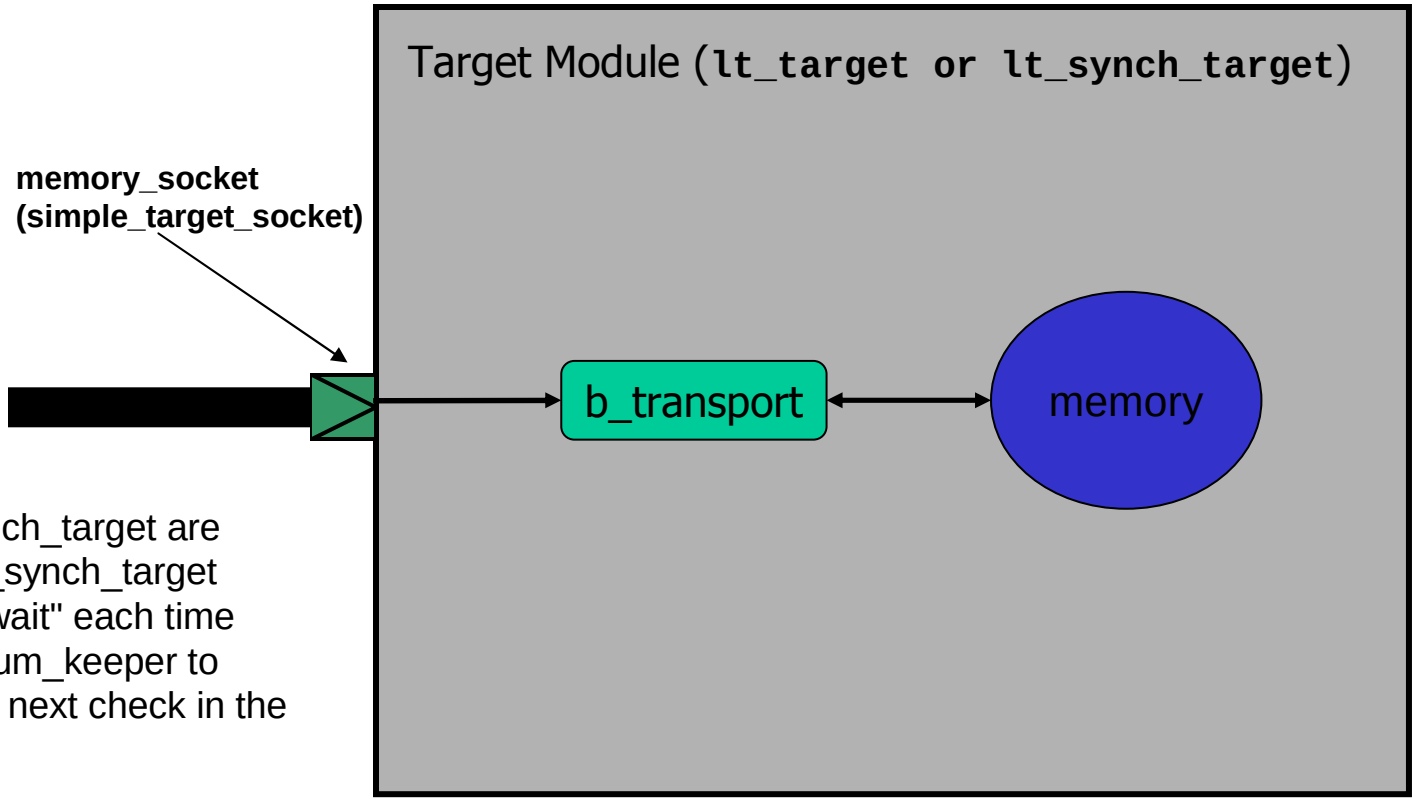


TLM Interface Module

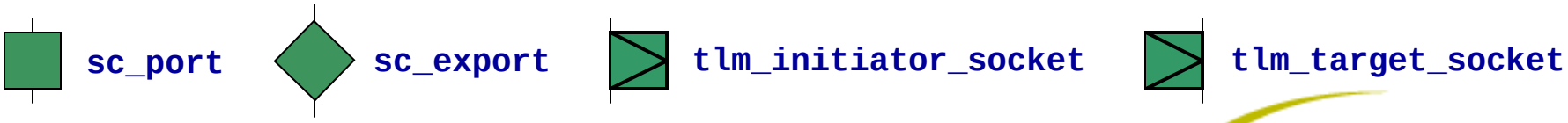


SYSTEMC™

Target Module (lt_target or lt_synch_target)

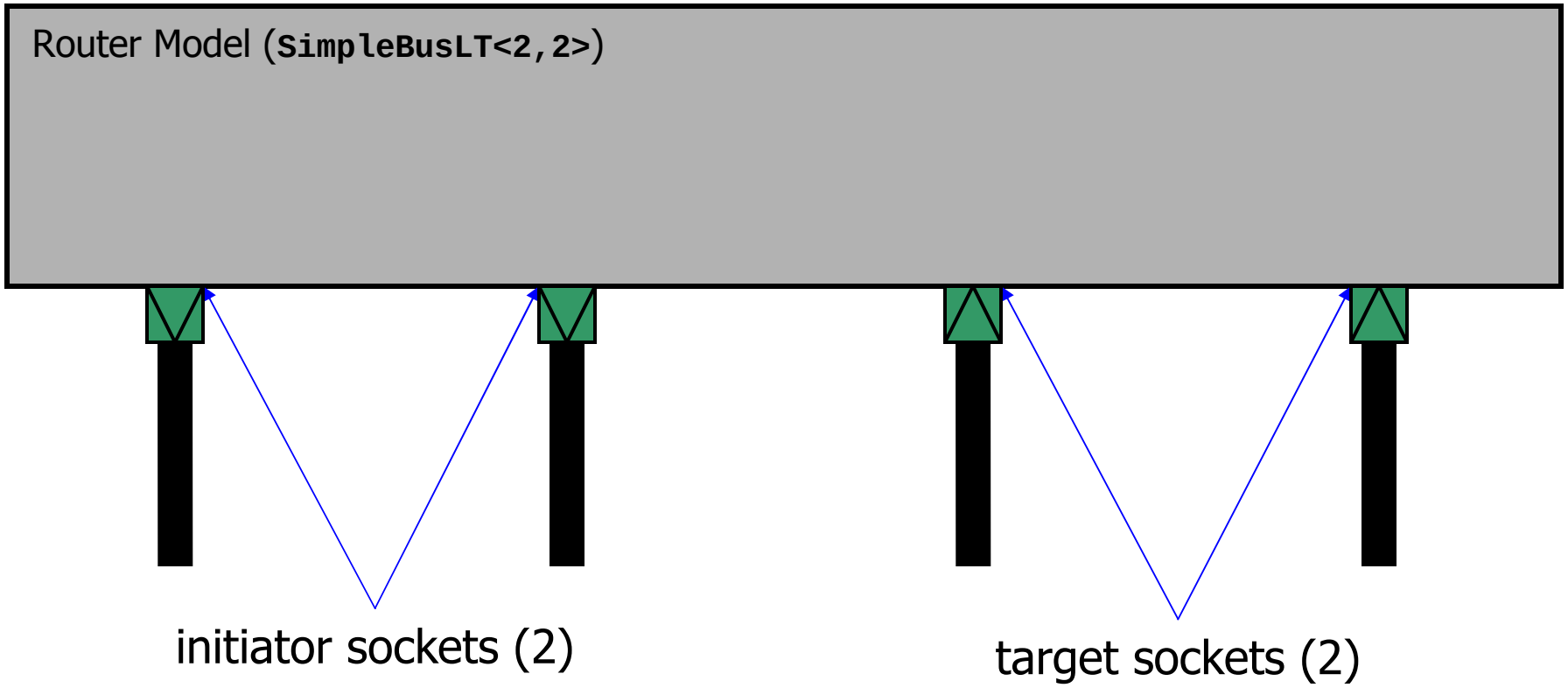


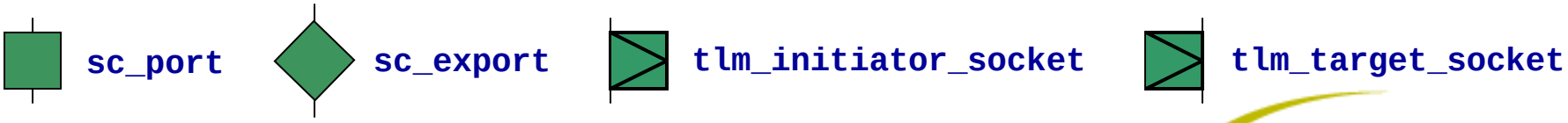
lt_target and lt synch_target are identical except lt_synch_target calls SystemC's "wait" each time causing the quantum_keeper to "synch" during the next check in the initiator



SYSTEMC™

Router Component





SYSTEMC™