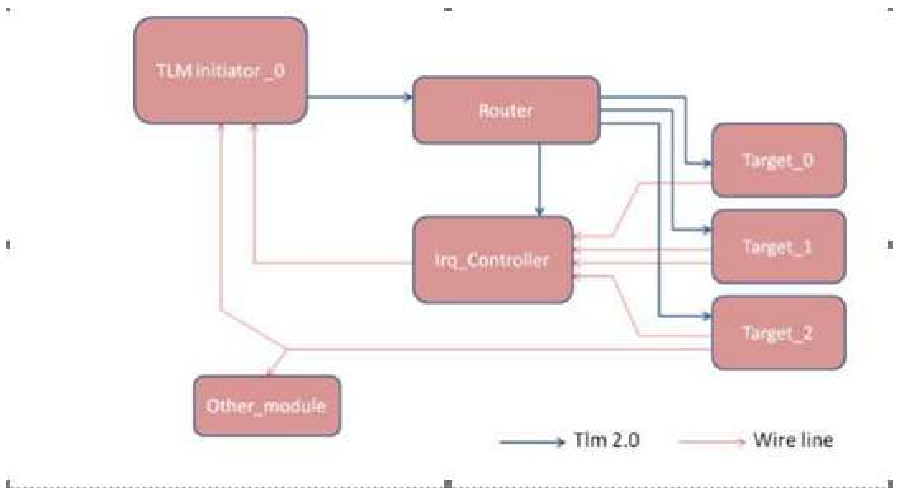
Wrapping Matchlib AXI4 Models for use in TLM2 Virtual Platforms

Stuart Swan | Platform Architect
Siemens EDA, a part of Siemens Digital Industries Software

Key Characteristics of TLM2 Virtual Platforms

- SOC model has enough detail to boot OS and run SW apps, but no more detail.
- View of the SOC is what the CPU sees:
  - "memory map accurate" modeling of bus transactions
  - interrupt lines



Some TLM2 Technical Aspects

- TLM2 bus transactions:
  - Abstracted read / write burst transactions
  - Burst requests just send a pointer to the data and a data length
  - All data pointers and lengths are in terms of bytes

- TLM2 optimizations for simulation performance:
  - Usually, there is no clock ticking in the system (as there would be in an RTL simulation)
  - Usually, entire burst is modeled with a single function call that does not block in the slave/target
  - Only send pointers
  - In some cases (DMI), don't even tell peripherals CPU is accessing their memory, just access it thru pointer silently

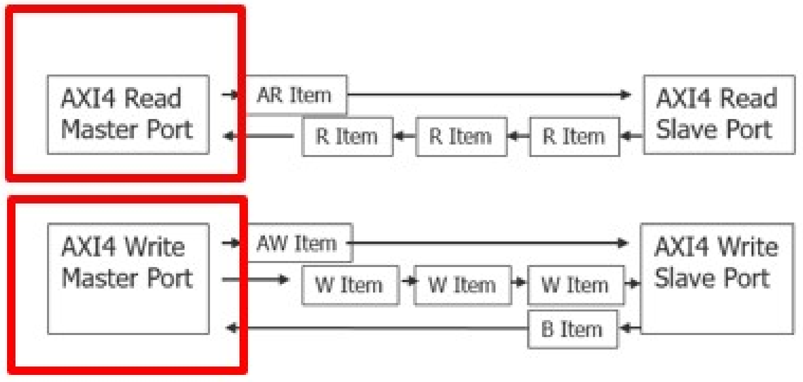
Areas Where TLM2 is Used

- TLM2 is the industry standard for processor ISS models, abstract bus fabrics.
  - Used for pure SW "virtual platforms"
  - Used for host code execution (HCE)
  - Used for "hybrid" emulation for portion not running on the emulator
  - HW architects may use TLM2 for architectural studies, but it is painful
  - TLM2 is NOT used for HW design or HLS.

- Almost all TLM2 models are "LT" or "Loosely Timed" models, which is what we are discussing in this presentation
- TLM2 also supports "AT" or "Accurately Timed" models, but these are rarely used and we are not discussing these in this presentation.
  - The Accellera SystemC TLM2 implementation provides automatic converters from AT to LT models if you encounter any TLM2 AT models

Key Characteristics of Matchlib AXI4 Models

- Precisely models AXI4 transactions on all 5 AXI4 channels (ar, r, aw, w, b)
  - All channels are modeled in a “bit accurate” and “beat accurate” way)
- Synthesizable thru HLS with high QOR.
- "thruput accurate" (ACCURATE_SIM), also FAST_SIM mode
- For more information on Matchlib AXI4 transactors, see:
  - https://forums.accellera.org/files/category/2-systemc/
  - Within the download, look at:
    - matchlib_examples/doc/matchlib_customer_training.pdf



Terminology Differences between TLM2 and AXI4

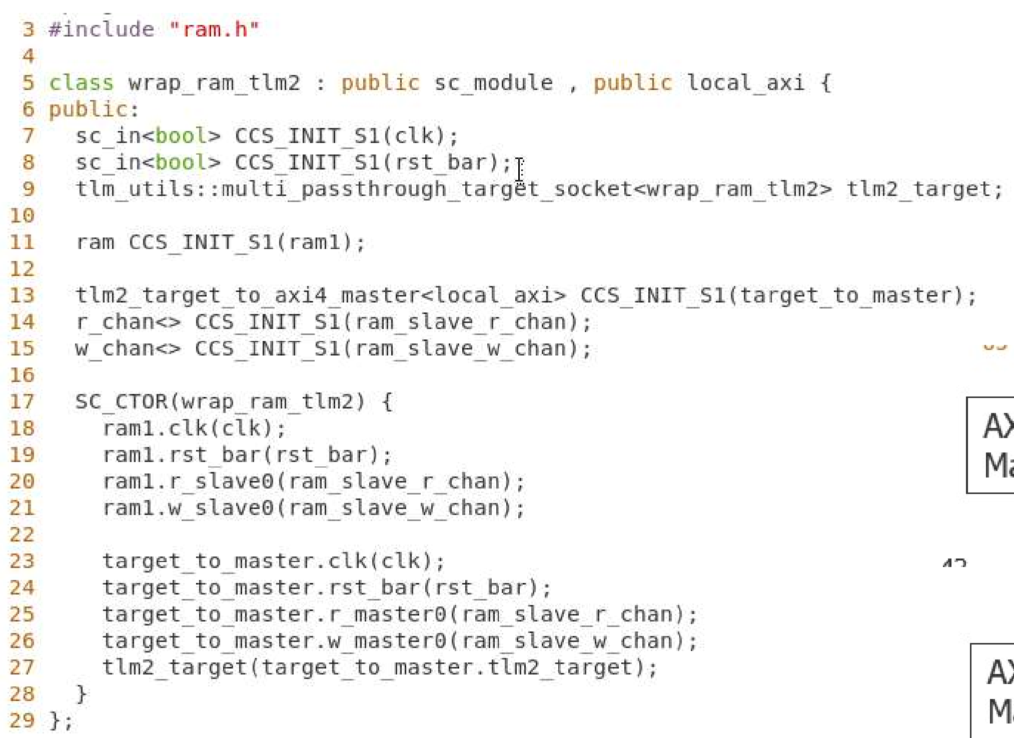
- TLM2: initiator / target
- AXI4: master / slave
- AXI4: "channel or chan" == "wires connecting a master and a slave"
- TLM2: does not use channels, instead initiators and targets are directly connected

So, how to mix TLM2 and Matchlib AXI4 models?

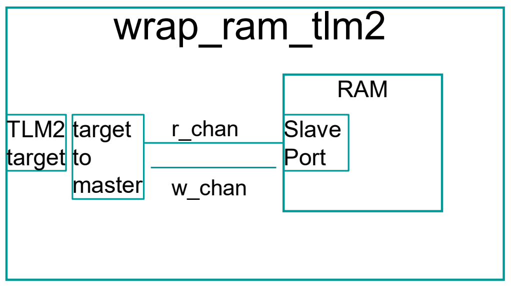
- Easy: wrap the Matchlib models so they look like TLM2 models.
  - We provide modular wrapper code that makes the wrapping process very simple/mechanical.
  - All the example code is in 30_tlm2_dma in Matchlib examples (not yet in 2022.1 release however).
- Keep the TLM2 parts and the Matchlib parts of the code cleanly separated.
- Then, the wrapped Matchlib models can be dropped into TLM2 VP and used as if they were native TLM2 models.

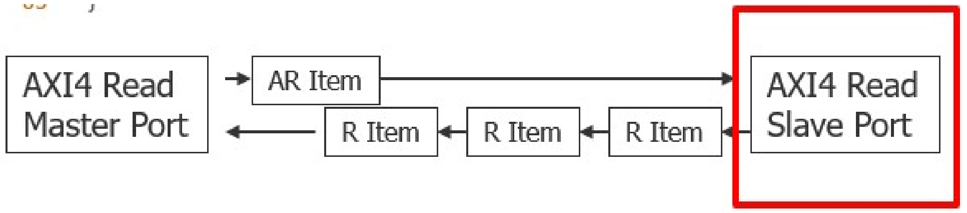
SIEMENS

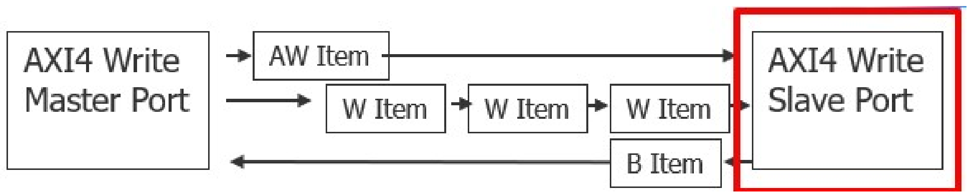
Walk-thru of TLM2 Wrapper for Matchlib RAM Model



```
#include "ram.h"

class wrap_ram_tlm2 : public sc_module , public local_axi {
public:
  sc_in<bool> CCS_INIT_S1(clk);
  sc_in<bool> CCS_INIT_S1(rst_bar);
  tlm_utils::multi_passthrough_target_socket<wrap_ram_tlm2> tlm2_target;

  ram CCS_INIT_S1(ram1);

  tlm2_target_to_axi4_master<local_axi> CCS_INIT_S1(target_to_master);
  r_chan<> CCS_INIT_S1(ram_slave_r_chan);
  w_chan<> CCS_INIT_S1(ram_slave_w_chan);

  SC_CTOR(wrap_ram_tlm2) {
    ram1.clk(clk);
    ram1.rst_bar(rst_bar);
    ram1.r_slave0(ram_slave_r_chan);
    ram1.w_slave0(ram_slave_w_chan);

    target_to_master.clk(clk);
    target_to_master.rst_bar(rst_bar);
    target_to_master.r_master0(ram_slave_r_chan);
    target_to_master.w_master0(ram_slave_w_chan);
    tlm2_target(target_to_master.tlm2_target);
  }
};
```







SIEMENS

Optimizing the Simulation Performance of Matchlib models in TLM2 Virtual Platforms

- Use FAST_SIM mode
  - This is a Matchlib simulation performance optimization mode that eliminates most cycles from the simulation so that models execute almost completely in “zero time”.
- Can set clock frequency to be 100x if desired
  - Some TLM2 virtual platforms may run for 10s of seconds of simulated time
  - For these long duration simulations, the SystemC clock ticking may slow down sim even though it is not really doing anything.
    - 1 ns clock X 60 seconds of simulated time = 60 billion ticks
    - Rather than removing the clock, just make it tick 100x or 1000x more slowly

Thank you!

SIEMENS