Embedded System Design and Modeling

EE382V, Fall 2008

Lecture Notes 5

System-Level Design Tools

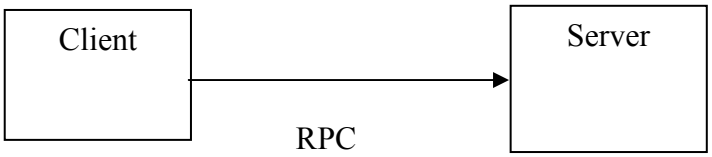
Dates: Sep 23&25, 2008
Scribe: Mahesh Prabhu

Computation Design:

Architectural Exploration:

Consider a complex channel which uses a semaphore. Initially we will map such a channel to a client server behavior where the client and server communicate with each other through a message passing system (eg. RPC).

Later we map the client and server to different processing elements.



Communication Design:

Network Exploration:

Bus Bridges: These are low level network devices that are used to connect two dissimilar busses.

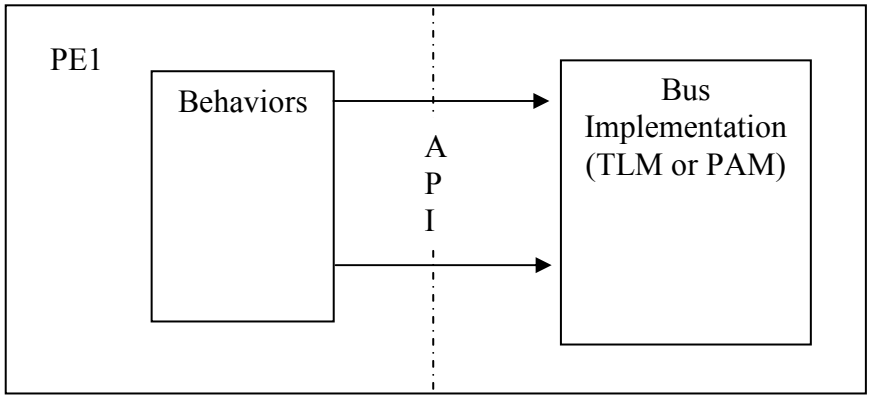
Transducers: These are more sophisticated as compared to Bus Bridges. They would have the facility for storing and forwarding information. For example a UART which buffers and transmits over a RS-232 interface.

Communication Synthesis:

(1) Addressing: how will the memory and devices share the address space?
(2) Synchronization: How will the communication happen from the Master to slave devices?
    There are several options:
    a. Interrupts
    b. Polling
    c. Interrupt Sharing
(3) Transfer Mode: how will the transfer of data take place once the initial synchronization information is exchanged? eg. DMA

Pin Accurate Model: This is an accurate structural model down to the details of all the wires used for the protocol.

Initially we would have a TLM for the bus and later when we proceed to have a more detailed simulation of the system we would replace the TLM with a PAM. The actual bus implementation from TLM to PAM is got from the component library/database. The tools then synthesize the communication over the given bus TLM/PAM. This synthesis is achieved by having canonical bus interfaces for the PEs to communicate over (which makes it possible to have tools generated code for any bus, as long as it provides the canonical API). The component library buses are given in the form of sir files with SpecC annotations for meta-information.



Also the SpecC model interface must reflect the choice of hardware. For example if a processor has 2 bus interfaces, then the SpecC processor model should communicate on two bus channels.

Q: If we have 3% timing accuracy after communication refinement then why not have higher levels of timing accuracy & iterate fewer times even though the simulation takes longer?

This is avoided because we first need to narrow down the exploration space. This can be done effectively only by simulating faster (as we have time to market constraints). Once the choices are down to 10 or so then we can go for more accurate models.