CSE398: Network Systems Design

Instructor: Dr. Liang Cheng
Department of Computer Science and Engineering
P.C. Rossin College of Engineering & Applied Science
Lehigh University

April 13, 2005

Outline

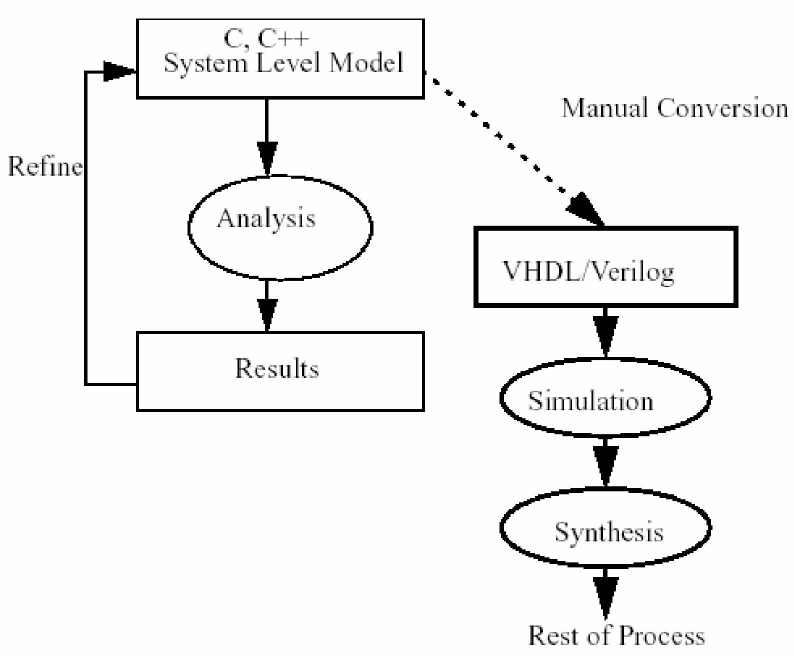
- Recap
  - APP550 processor architecture
    - Classification, policing, traffic & stream editor engines
  - FPL classification language
    - Pattern / action language at packet bit level
    - Patterns compile to a constant-time matching tree
  - State engine and C-NP
- System architecture and modeling
- Summary and homework

Outline

- Recap
- System architecture and modeling
- Summary and homework

SystemC in One Sentence

- SystemC provides hardware-oriented constructs within the context of C++ as a class library implemented in standard C++.



SystemC Simulation Modeling

- An open source C++ environment
  - Modeling electronic systems and circuits
- Cosimulation models
  - Execution of software on simulated hardware
- Stepwise refinement
  - Supports simulation at increasingly detailed levels of fidelity
- Exploring next generation network processing
  - Architecture modeling and/or ASIC design

SystemC C++ Constructs

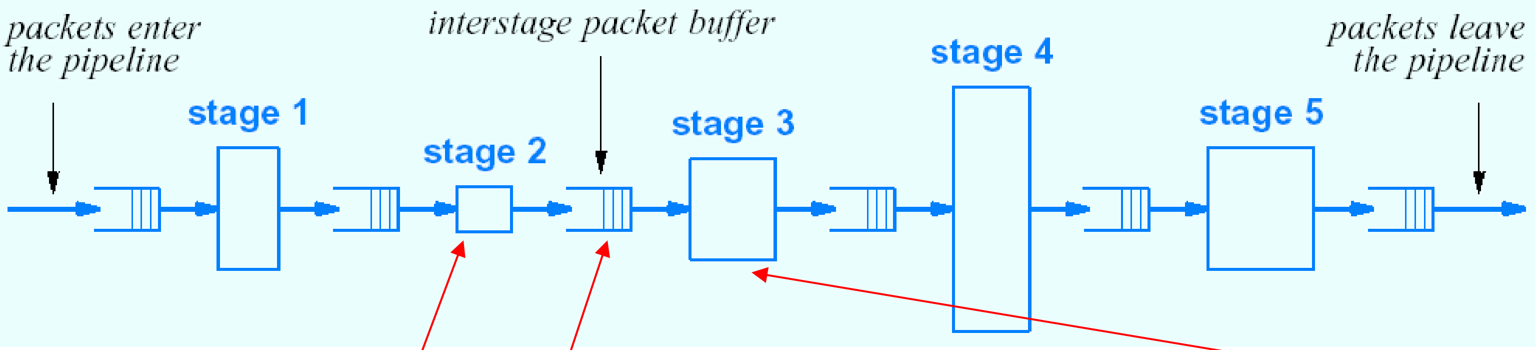
- SystemC uses class derivation, macros, and template classes to provide circuit structure and behavior modeling primitives.
- Programmers create models by defining classes and interconnecting their instances (objects) using C++ constructs and SystemC libraries.

SystemC Levels of Refinement

- Architecture model – model flow of data through system blocks.
- Transaction model – add timing.
- Transfer model – add clocks, interface circuitry.
- RTL (register transfer level) model – include all circuitry in enough detail for a circuit synthesizer.

An Example of Architecture Model

- Pipeline architecture model



- A producer block that sends bytes to a consumer block via a FIFO
  - FIFO suspends the producer or consumer as necessary
- Problem
  - Given processing speeds of stage 2 and stage 3, find out the FIFO size to achieve a certain average overall processing speed or throughput of the stage2 and stage 3

An Example of Architecture Model

- The consumer block (stage 3) will consume exactly one byte every 100 ns unless it is suspended waiting for input from the FIFO.
- The producer block (stage 2) produces between 1 and 19 bytes every 1000 ns unless it is suspended waiting to write to the FIFO.
- Determine the size of the FIFO needed to sustain a throughput of 1 byte per 110ns or 110 ns per byte

Terminology in SystemC Code

- Modules
  - A container class: a hierarchical entity that can have other modules or processes contained in it.
- Processes
  - Processes are used to describe functionality. Processes are contained inside modules.
- Ports
  - Modules have ports through which they connect to other modules.
- Threads
  - A process that runs in an infinite loop and waits on an event that activates its execution.

Code Study

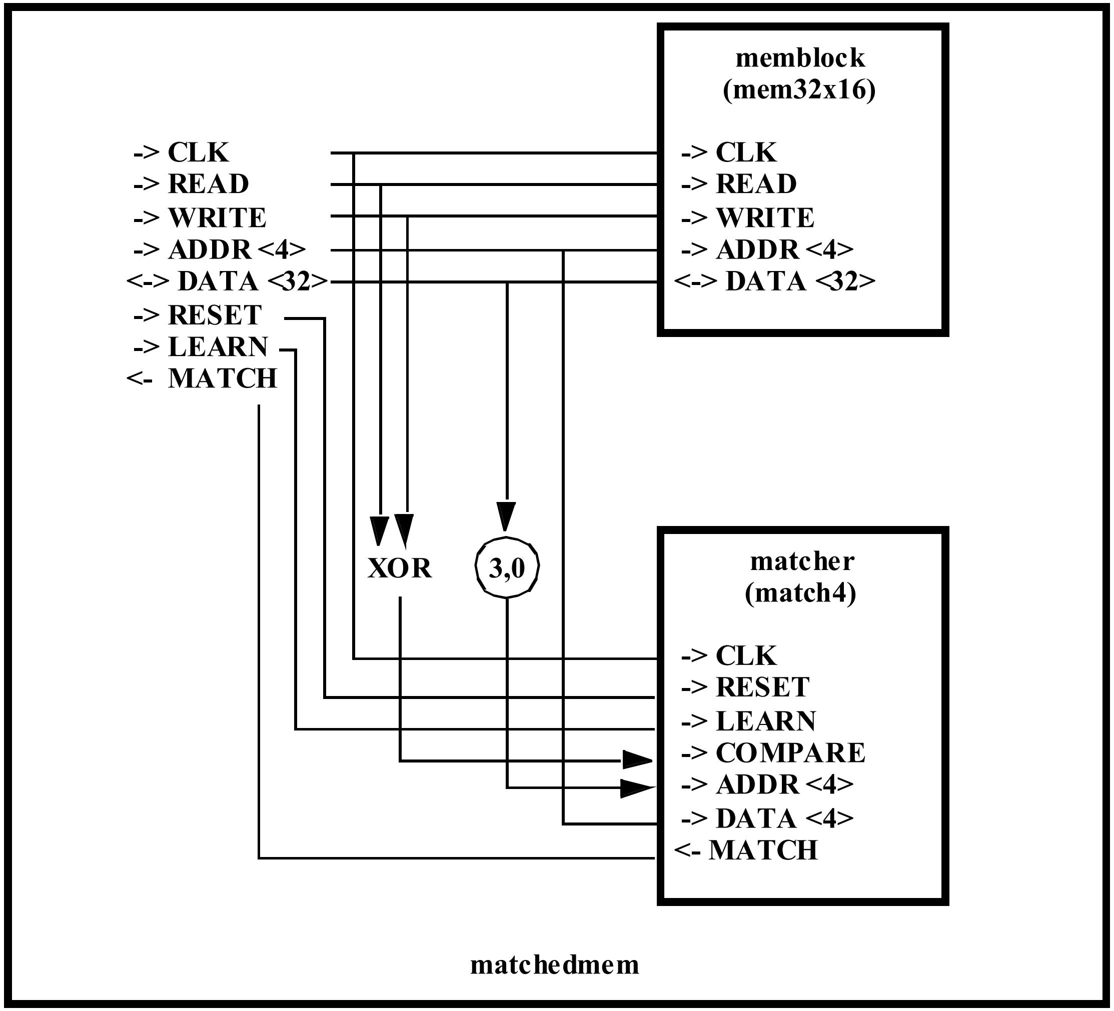
Another Example SystemC Model

- Matchedmem is memory with a breakpoint detector on the address bus.
  - Hierarchical module containing a memory block model, a match detector model consisting of registers and comparators, signals that interconnect these nested models, and ports that provide connection hooks to peer models.

Matchedmem’s Behavior

- Models read/write memory using class mem32x16 as nested object memblock.
- Models breakpoint comparator on its address bus using class match4 as nested object matcher.
- Module driver_matchedmem drives data on inputs, monitor_matchedmem watches outputs, testmem_main runs the simulation.

Matched Memory Schematic



What is a Module?

- A module is a class derived from a SC base class that models a circuit.
- You test a module by instantiating it in a testbench module, driving its inputs from a driver module, and sampling its outputs using a monitor module.
- See SC_MODULE (matchedmem) in file matchedmem.h, also testmem_main.cpp.

What is a Port?

- A port is a C++ container that houses a data object such as a bool or an integer, and that provides a connection into its module from external modules.
- There are input, output, and inout ports.
- See sc_in<bool> CLK, sc_out<bool> MATCH, sc_inout<sc_uint<32> > DATA, etc. in file matchedmem.h.

What is a Signal?

- A signal is a C++ container that houses a data object such as a bool or an integer, and that provides data storage or a connection between modules nested within a module.
- See sc_signal<bool> match4_compare and sc_signal<sc_uint<4> > DATA_low_4_to_match4_addr in file matchedmem.h.

Ports/Signals Contain:

- Built-in C++ data types such as a bool for a single bit or int for a bus.
- sc_uint<N> is a SC template class that uses N to model the width of a bus.
- sc_bv<N> bit vectors support N > 64.
- A clock is a single-bit port or signal whose transitions control storage within a module and output from a module.
- Value-bearing objects include type-specific read(), write() and “=“ functions.

What Is a Process?

- A process is a C++ function that defines behavior for its module.
  - A method is a process that SC invokes when an input/inout on its sensitivity list changes. It can change outputs/inouts and state variables before it returns.
  - A thread is a process that runs in an infinite loop and waits on an event that activates its execution. It never returns.

When to Use a Method?

- Use a method when a change on an input/inout maps via simple code to change in state variable(s) or output/inout.
  - Most of our models use methods.
- See SC_METHOD (prc_xor_read_write) in files matchedmem.h and matchedmem.cpp, SC_METHOD (prc_match4) in match4.h and match4.cpp.
  - Note “sensitive_neg << CLK << RESET” in match4 defines prc_match4’s sensitivity list – inputs that activate the method when they change value.

When to Use a Thread?

- Use a thread when a model only drives outputs (has no inputs) or when it models complex behavior in an architecture model.
    - Only our driver modules use threads.
- See SC_THREAD(prc_driver_match4) in files driver_match4.h and driver_match4.cpp.

Outline

- Recap
- System architecture and modeling
- Summary and homework

Summary

- SystemC is a collection of C++ base classes, template classes and macros that supports simulation of electronic systems.
- Multiple levels of fidelity allow for architecture exploration and refinement.
- Modules, ports, signals and processes (methods and threads) are the main constructs.
- SystemC supports ASIC design and processor/software cosimulation in network processing.

Reading Materials

- Examine SystemC code for tcam and memory in CD.