SystemC, OCCN and VPs

M. Grammatikakis, E. Politis and C. Papadas
Integrated Systems Development
NoC Round Table, ESTEC
17 September 2009
On SoC/NoC Modeling Developments

- ISD has contributed to the design, documentation, and validation of
  - a custom system-level C++-based IP modeling and simulation framework (IPSIM),
  - standardizing modeling objects and methodology in SystemC
  - a custom SystemC-based IP modeling library (IPSIM subset) has been ported to Synopsys Concentric studio, and
  - an open-source SystemC-based On-Chip Communication Network framework (OCCN)
  - ST Spidergon …
  - Real life applications (power, fault tolerance, etc)
About IPSIM

- ISD designed, developed and validated objects for
  - memory: FIFOs, memory and cache (data, instruction and unified),
  - hierarchical finite state machines, and
  - high-level performance modeling and visualization.
- ISD contributed to
  - developing intra/inter-module communication (mailbox, message)
  - standardizing intra-module synchronization objects in SystemC (e.g. Mutex & Semaphore)
  - debugging a custom simulation kernel (context switching, fairness)
  - interfacing models with VxWorks RTOS for hw/sw codesign, and
  - designing, validating & integrating models for a SoC soho gateway (customer CISCO).
  - importing IPSIM objects into Synopsys Cocentric Studio (SystemC)
- Documentation and Technical Support
IPSIM in Synopsys Concentric Studio

- ISD customized and ported SystemC2.0-based library objects (data types, memory, intra- and inter-module communication) and test suites into Synopsys CoCentric System Studio, both at user & system level.
- This feasibility study enabled Synopsys CoCentric System Studio (with interactive model design, validation and additional performance modeling support) to become the design environment for
  - transferring SystemC-based SoC models (e.g. BBNT) within STM,
  - sharing IPSIM design flow within STM and with external partners.
OCCN Methodology for SoC Modeling

- OCCN is an open-source communication modeling environment, with a SystemC-based library, a run/test environment, and communication refinement extending general SoC design methodology.

- OCCN abstraction levels range from functional to transaction-level clock-cycle, bit-accurate. RTL is supported by SystemC-based behavioral synthesis tools, e.g. Synopsys Co-Centric System Studio.
About OCCN

- ISD developed
  - fundamental OCCN objects, such as Pdu and MsgBox,
  - debug almost the entire OCCN library, and
  - port the OCCN library and tests to Linux platform (from Solaris).
- Moreover ISD has
  - administered the open-source sourceforge project,
  - documented OCCN with white paper, articles and user/channel design manuals,
  - designed, developed and validated all high-level performance modeling objects,
  - provided technical support on demand to users & collaborators.
- A new release will be shortly available.
OCCN Recognition

- OCCN is the only open-source NoC modeling framework available.
- OCCN heavily cited and utilized by 300+ organizations worldwide (with 2000+ recorded downloads) from prestigious schools, research institutions, and semiconductor, electronics, and EDA industry.
- OCCN used for efficient modeling and design space exploration of STMicroelectronics Spidergon STNoC. A generic Spidergon STNoC router model has been built using OCCN facilities in a few months.
- OCCN won the STMicro Corporate Community of Practice Gold Award in 2005
The ST Spidergon NoC

- ST Octagon and STBus are starting points
- Regular, constant degree topology
- Network expandability of 2 nodes
- Deterministic routing algorithm:
  - source-based routing
  - simple (no lookup table)
  - diameter = \( \text{ceil}(N/4) \)
- Wormhole switching
- Directly output buffers
- High thoughput
- Virtual channels to avoid deadlock
About ST Spidergon NoC

- ISD contributed towards the design of an efficient NoC topology by
  - developing static and dynamic theoretical performance models,
  - developing/validating SystemC and Omnet simulation models, and
  - adjusting existing network simulators (e.g. Chaos router).

- ISD contributed to NoC router architecture by developing efficient
  - flow control, switching strategy, and buffer management strategies,
  - p-p and intensive communication algorithms, and
  - VC allocation/scheduling.

- ISD improved the NoC network interface by examining
  - packet reordering within the network or network interface,
  - communication layering and interfacing to standard buses (AMBA Axi, STBus, etc), and
  - end-to-end flow control, e.g. VC scheduling & buffer management.
OCCN Modeling and Design Exploration

- An OCCN-based Spidergon STNoC router model took ~1mm to develop. It allows configuring
  - NoC topology,
  - packet and flit size,
  - number of virtual channels and allocation/scheduling policy,
  - routing function and arbitration policy, and
  - output buffer configuration and size at router and network interface.
Application Specific NoC Selection Tools

SW Application Models (Task Graphs)  
HW Blocks Model (Calc, Mem, Comm Cap)

Partitioning Tool // systems (i.e. Scotch)  
Topology Metrics Tool (Metis, Neato)

NoC Topology Exploration & Analysis

SystemC Executable Specifications (configurable routers & network interfaces)  
This contribution by ISD has led to a customized Linux-based NoC design toolsuite: Iput, Imap, Irun & Isee.

Performance Analysis & Synthesis

design optimization

automatic code generation
Real life concerns early in the development cycle...

- How fast the design of a new product can go?
- How much the SW development can be anticipated?
- What is the best architecture to be adopted?
- Which is the best target technology (cost vs perf)?
- What would be the power consumption? Can it be reduced?
- What is the impact of variability on .... power?
- What actions should be taken to minimize cost?

⇒ in order to provide answers in time a holistic approach is required …
Moore’s Law: Capacity of integrated chips doubles every 18-24mo

Potential solutions in the horizon:
- IP reuse – standardization
- High-level synthesis
- Power-aware modeling at higher abstraction level, e.g. using SystemC or even UML
System-Level Design

- C/C++ based design methodology

1. conceptualize
2. simulate in C/C++
3. write specification document

4. hand over
   - executable specification
   - testbenches
   - written specification

5. understand specification
6. refine in C/C++
7. validate re_using testbenches
8. synthesize from C/C++
Benefits from High Level Modeling

- System-level modeling in a certain level of abstraction
  - reduces significantly the number of errors,
  - enables block and system reuse through rapid communication refinement, composition and reconfiguration,
  - supports efficient design exploration and verification of complex multicore SoC architectures for which synchronization hazards and QoS cannot be examined unless the RTL is available,
  - accelerates early development of firmware/software through a virtual prototype which provides much higher simulation efficiency, and
  - meets smaller time-to-market.

- What about the link between Power and Technology?
System-level power estimation provides sufficient accuracy at improved cost, performance and productivity compared to RTL flow.

Low-level power useful only for late design optimization.
System-Level Power Estimation

- ISD developed an efficient, high-level SoC dynamic power estimation methodology based on bit- and cycle-accurate transaction-level SystemC macro-models.

- The dynamic power is estimated by multiplying the switching activity (number of transactions and/or bit transitions for all input and output gate signals) with appropriate bit energy coefficient.

- The accuracy depends on the level of abstraction

- Bit energy coefficients for a given technology node (e.g. 32 or 22nm) and derivatives (e.g. high performance or low power) can be extracted from statistical measurements on fabricated test structures, the semiconductor technology roadmap (e.g. ITRS) and simulation platforms (e.g. MASTAR).

- For current technologies, calibration for absolute power estimation can be based on statistical measurements on fabricated test structures.
ISD Contribution: SystemC Power Modeling

- System-Level Power Estimation
  Transaction and Transition-based Models

ISD PULLNANO Contribution

- TLM
  Transaction Level
- HLS
  Synthesis Subset
- SCV
  Verification Library
- AMS
  Analog Mixed Signal

SystemC Extensions

- Core Language
  - Module
  - Ports
  - Processes
  - Events
  - Interfaces
  - Time
- Channels
  - Signal
  - Buffer
  - Clock
  - Mutex
  - Semaphore
  - FIFO
- Data Types
  - Logic
  - Bit vectors
  - Logic vectors
  - Fixed point
  - C++ types
  - …
- Utility Classes
  - simulate control
  - tracing
  - …

Event-driven simulation kernel

C++ Language Standard
ISD applied its system-level power estimation methodology to two complex, bit- and cycle-accurate SystemC virtual platforms for future 32/22nm technologies.

- A network bridge providing sequential access to internal memory tables for inserting, searching or retrieving connection contexts.
- A NoC in a shared-memory multicore SoC supporting SIMD array processing.
Ethernet Screening

- Cycle- and bit-accurate SystemC VP of a core network bridge component providing VLAN support (IEEE 802.1q). It features:
  - MAC address type identification of incoming Ethernet packets.
  - Complex FSM performs fast search of several lookup tables.
  - Significant computation load: 2-level CRC-based hashing.

> computation-intensive SoC benchmark for system-level dynamic power estimation
Although during initial insertions memory power dissipation is large, as time progresses and conflicts occur, registers and other combinational circuitry start to consume much more power.

Architecture refinement
Cycle- and bit-accurate SystemC of a NoC-based homogeneous multicore exploiting data & instruction parallelism common in DSP and multimedia applications. Features include:

- NoC topology configured as a quad-tree of generic 1x4 switches.
- Power-aware 1x4 expander multicast switch supports SIMD-based parallel computation, communication & synchronization patterns.
System-Level NoC: Multicast vs Broadcast

- Using relative metrics, NoC simulation shows that multicast is more power-efficient only for a limited number of multicast addresses.
- Power normalization is based on a 32nm note (linear regression).
- Communication protocol refinement
System-Level NoC: SIMD Processing

- For the more complex vector parallel processing routines, such as bitonic sorting, power consumption increases.
- Algorithm selection
- Target Technology selection
System-Level NoC: Power Gating

- Early selection of the architecture and the technology
Extend SystemC Models with Roadmap Data

Bit- and cycle-accurate SystemC models can be automatically annotated from technology roadmap, e.g. ATRS, datasheets or device models (e.g. MASTAR or bsim3) for reliable predictions of system power and speed in future 32/22nm technology nodes and variations.

SystemC-based power instrumentation can enable design exploration of architectural, algorithmic and technology features for energy-aware product differentiation and variation-aware power analysis.
A highly parallel Virtual Platform

- ISD examines architectural & algorithmic support for correct multicore execution of simultaneous shared memory and message passing directives, i.e. remote read/write & send/receive primitives.

- ISD also implements a transaction-level bit- and cycle-accurate SystemC-based virtual platform of a multicore SoC composed of a NoC connecting together tens or hundreds of processing elements with local memory to several interleaved storage elements.

- The VP will support both message passing and shared memory multiprocessing primitives, while a model extension will cover streaming and transactional memory.

- Using this VP, ISD will address cost, performance, scalability and power tradeoffs of different programming models (and hybrid combinations) for computationally intensive multimedia applications.
Fault Tolerant (MODERN)

- With unreliable processes we should make reliable products
- ISD implements fault diagnosis, packet encoding/retransmission, fault tolerant routing and reconfiguration mechanisms in a VP of a multicore SoC composed of a NoC and many nodes.
- On top of this fault tolerant VP, ISD will implement dynamic power estimation based on measuring system-level switching activity.
- Using this VP, ISD plans to consider the design of low cost, reliable and power-efficient multicore systems by exploring algorithmic, topological, architectural, and technological characteristics.
- THL will port their (very interesting) tile over this VP.
End of Presentation.

Thank you.