IBM PowerPC Enablement Kit:

ChipBench-SLD: System Level Analysis and Design Tool Suite

Dr. Nagu Dhanwada, Chief System Level Design Solutions Architect, Electronic Design Automation, IBM Systems and Technology Group.
Agenda

- Context and Introduction
- ChipBench-System Level Design (SLD) Overview
- Examples
- POWER Architecture Evaluation Kit V1.0
- Conclusions
System Level Design Tasks

Algorithm/Specification Level

Design / System Architecture Exploration Level

System Specification and Modeling (UML, Matlab, etc)

Early design exploration,
Map functions in spec to hardware blocks,
Determine IP blocks, Bus architecture, etc
Simulation based tools that uses high level IP models
Graphical specification and visualization

Architecture Implementation Level

Generation of Configured IP (based on architecture exploration)
Generate assembled system automatically
Manage system configuration data
Synthesis front end for detailed design (RTLdown)
Generate Verification environment
Early floorplanning, die-size estimation, chip-power estimation and packaging analysis

RTL Handoff

Traditional RTL-GDSII Flow

SystemC
Performance
Power,
Timing,
Area Models

Configuration Information

Synthesis Scripts,
Timing constraints,
Floorplan
Constraints

System Level Design Tasks

Algorithm/Specification Level

Design / System Architecture Exploration Level

System Specification and Modeling (UML, Matlab, etc)

Early design exploration,
Map functions in spec to hardware blocks,
Determine IP blocks, Bus architecture, etc
Simulation based tools that uses high level IP models
Graphical specification and visualization

Architecture Implementation Level

Generation of Configured IP (based on architecture exploration)
Generate assembled system automatically
Manage system configuration data
Synthesis front end for detailed design (RTLdown)
Generate Verification environment
Early floorplanning, die-size estimation, chip-power estimation and packaging analysis

RTL Handoff

Traditional RTL-GDSII Flow

SystemC
Performance
Power,
Timing,
Area Models

Configuration Information

Synthesis Scripts,
Timing constraints,
Floorplan
Constraints

Algorithm/Specification Level

Design / System Architecture Exploration Level

System Specification and Modeling (UML, Matlab, etc)

Early design exploration,
Map functions in spec to hardware blocks,
Determine IP blocks, Bus architecture, etc
Simulation based tools that uses high level IP models
Graphical specification and visualization

Architecture Implementation Level

Generation of Configured IP (based on architecture exploration)
Generate assembled system automatically
Manage system configuration data
Synthesis front end for detailed design (RTLdown)
Generate Verification environment
Early floorplanning, die-size estimation, chip-power estimation and packaging analysis

RTL Handoff

Traditional RTL-GDSII Flow

SystemC
Performance
Power,
Timing,
Area Models

Configuration Information

Synthesis Scripts,
Timing constraints,
Floorplan
Constraints

Algorithm/Specification Level

Design / System Architecture Exploration Level

System Specification and Modeling (UML, Matlab, etc)

Early design exploration,
Map functions in spec to hardware blocks,
Determine IP blocks, Bus architecture, etc
Simulation based tools that uses high level IP models
Graphical specification and visualization

Architecture Implementation Level

Generation of Configured IP (based on architecture exploration)
Generate assembled system automatically
Manage system configuration data
Synthesis front end for detailed design (RTLdown)
Generate Verification environment
Early floorplanning, die-size estimation, chip-power estimation and packaging analysis

RTL Handoff

Traditional RTL-GDSII Flow

SystemC
Performance
Power,
Timing,
Area Models

Configuration Information

Synthesis Scripts,
Timing constraints,
Floorplan
Constraints

Algorithm/Specification Level

Design / System Architecture Exploration Level

System Specification and Modeling (UML, Matlab, etc)

Early design exploration,
Map functions in spec to hardware blocks,
Determine IP blocks, Bus architecture, etc
Simulation based tools that uses high level IP models
Graphical specification and visualization

Architecture Implementation Level

Generation of Configured IP (based on architecture exploration)
Generate assembled system automatically
Manage system configuration data
Synthesis front end for detailed design (RTLdown)
Generate Verification environment
Early floorplanning, die-size estimation, chip-power estimation and packaging analysis

RTL Handoff

Traditional RTL-GDSII Flow

SystemC
Performance
Power,
Timing,
Area Models

Configuration Information

Synthesis Scripts,
Timing constraints,
Floorplan
Constraints

Algorithm/Specification Level

Design / System Architecture Exploration Level

System Specification and Modeling (UML, Matlab, etc)

Early design exploration,
Map functions in spec to hardware blocks,
Determine IP blocks, Bus architecture, etc
Simulation based tools that uses high level IP models
Graphical specification and visualization

Architecture Implementation Level

Generation of Configured IP (based on architecture exploration)
Generate assembled system automatically
Manage system configuration data
Synthesis front end for detailed design (RTLdown)
Generate Verification environment
Early floorplanning, die-size estimation, chip-power estimation and packaging analysis

RTL Handoff

Traditional RTL-GDSII Flow

SystemC
Performance
Power,
Timing,
Area Models

Configuration Information

Synthesis Scripts,
Timing constraints,
Floorplan
Constraints

Algorithm/Specification Level

Design / System Architecture Exploration Level

System Specification and Modeling (UML, Matlab, etc)

Early design exploration,
Map functions in spec to hardware blocks,
Determine IP blocks, Bus architecture, etc
Simulation based tools that uses high level IP models
Graphical specification and visualization

Architecture Implementation Level

Generation of Configured IP (based on architecture exploration)
Generate assembled system automatically
Manage system configuration data
Synthesis front end for detailed design (RTLdown)
Generate Verification environment
Early floorplanning, die-size estimation, chip-power estimation and packaging analysis

RTL Handoff

Traditional RTL-GDSII Flow

SystemC
Performance
Power,
Timing,
Area Models

Configuration Information

Synthesis Scripts,
Timing constraints,
Floorplan
Constraints
Context in an SLD Flow

Initial Application Specification

- HW/SW tradeoff Analysis, Processor selection, etc

Abstract Software Specification | Initial HW Architecture

- Software Refinement
- HW Architecture Refinement

Verification

- Application Code
- Refined HW Architecture

ChipBench-SLD

- SystemC Functional Transaction Level Models

Embedded SW Analysis

- Hardware Design

Architecture Analysis

- SystemC Performance Models, Power, Timing, Area Models

RTL—GDSII Flow
ChipBench-SLD: Motivation and Goals

❖ An environment to allow early analysis of SoCs

❖ How early and how quickly?
  • Prior to the existence of any detailed hardware specification
  • Short turnaround time from system specification to analyzing results

❖ Accurate enough to allow decisions to be made on:
  • Architecture
  • Components
  • Floorplan (Area + timing)
  • Chip/Die Size and Cost

❖ Direct links to implementation (e.g., detailed synthesis, P&R)
ChipBench-System Level Design: Conceptual View

SoC Integration

SoC Virtual Design

V1  V3  V5
V2  V4

SoC System-level Diagram

SoC Verification, Performance & Power Analysis w/embedded SW

Power

Analysis

Map

Top-level Program
Core Models
Instrumentation
Scheduler

Performance

Map

Analysis

Early Timing Closure

Fast Physical Prototyping

Physical Planning

Floorplanning

V1  V2
V4  V3
V5

SoC Floorplan + Die Size

Implementation

SoC RTL design

SoC Verification, Performance & Power Analysis w/embedded SW

SLD solution incorporates SW development, architectural exploration, and floorplanning in a single integrated environment.

September 2005 | ChipBench-SLD | Power.org
ChipBench-SLD Environment: Design and Verification Views

**HW Design and Development**

- Block Diagram
  - Performance, Power Analysis
  - Floorplan, Timing
  - SoC Netlist Generation
  - Architectural Changes
  - FP, Core Changes
  - Detailed Netlist

**HW Verification**

- Early Functional Verification (Transaction Level Simulation)
- Interconnect Verification (TOS)
- IBM/External Simulators
- Functional Verification

**Software Design and Development**

- Software
- HW/SW Co-simulation (IC-SIM)
- Accelerated Simulation (AWAN)

**Design Implementation**

- Design Release

---

*September 2005 | ChipBench-SLD | Power.org*
ChipBench-SLD: CoreConnect SystemC TLM Simulation Platform

- Simulate real application software interacting with IP core models
  - Simulation performance to run a significant software application

- Verify that system supports enough bandwidth and concurrency for target applications

- Transactions modeled as occurring over communication channels

- Computation (inside a core) may not be modeled on a cycle-by-cycle basis, as long as input-output delays are cycle-approximate

- Software-based Instruction Set Simulator (ISS) is used for the CPU,
SystemC Modeling of IBM’s CoreConnect Architecture

```
#define Ext_Mem 0x6F00
main(..) {
    char *p = Ext_Mem;
    // ...;
    *p = A + B;
    // ...;
}
```

1) Issw.plb_mp2->non_blocking_write (0x6F00, sum(A+B))
2) plb_bus.plb_ap->arbitrate_request(…)
3) Plb queries its slaves and finds plb_opb_bridge mapped to address 0x6F00
4) plb_bus.plb_sp1->write(0x6F00, sum)
5) brg.opb_mp1->non_blocking_write (0x6F00, sum)
6) opb_bus.opb_sp1->write(0x6F00, sum)
7) mc.mem_p->write(0x6F00, sum)
8) Mem[0x6F00] = sum

Source: RAB, IBM Research
Running Software on the SystemC Models

- Load the cross-compiled executable from the host file system using the RISCWatch debugger

- Executables are cross-compiled to target the PowerPC 405

- A linker script insures that executables map into the model’s memory space

- Newlib library provides most operating system level (libc) functions executables need
  - http://sources.redhat.com/newlib/
  - Newlib I/O functions work with the SystemC I/O Device model

- Redirect all ISS I/O operations to the host file system
ChipBench SLD - Power Analysis

_voltage

SoC Virtual Design

SoC System-level Diagram

Top-level Program
Core Models
Instrumentation
Scheduler

Power Augmented Transaction Level Model (SystemC)

_DMA

SI_MEM_MEM_TX

RD_FROM_PLB
Power parameter

RD_FROM_OPB
Power parameter

WR_TO_OPB
Power parameter

WR_TO_PLB
Power parameter

 владельец

Power Models for cores

- Transaction based characterization (read, write...) determines power values
- Transaction level simulation determines activity of each core,
- Power augmented SystemC model, executes performance and power simulation
- Workload-driven power analysis
ChipBench SLD - Area Analysis

SoC Virtual Design

SoC System-level Diagram

- Hard-core: area is known
- Soft-core: run area-planning to estimate area
- Estimate number of real pins and assign locations of major cores as needed
- Run FP to get initial placement & pin-assignment to refine pin positions (initially in the center)
- Re-run FP
- Run global routing
- Compute die-size and congestion metrics
Mapping to Real Design (RTL)

- **From Virtual Design to RTL**
- **Chipbench SLD virtual design creation**
  - Map Virtual components to Real RTL components + glue logic
  - Expand all virtual nets and pins into real nets and real pins
  - Make RTL connections as specified by pin properties
  - Generate RTL VHDL/Verilog

- **Pass Virtual FP initial placements to Real FP initial placements**

- **Complete FP of RTL / Real Design**
- **Die Size**
- **Timing analysis**

- **Consistent timing-placement algorithms SLD-> Prototyping->Physical Synthesis Implementation if following with IBM RTL-GDS implementation flow**
SoC Early Analysis System Experiment

- PPC405 Platform Based Design
  - Ethernet Subsystem
    - 1 EMAC
    - 1 Madmal

- Real Embedded Application along with simple OS executing on TLM models to evaluate performance of Ethernet Sub-System
  - Common example to test performance of ethernet subsystems
  - Sets the Ethernet Macro in a loopback mode, and compares the transmitted and received packets

- Measure effects on performance, die-size, fp, power

- Change to improve performance
  - Added an extra EMAC + Fifos

- Two EMAC mode of the application works with packets being transmitted from one EMAC and received by the other

---

**System Throughput (KBytes/sec)**

- 2 Emac
- 1 Emac

**CPU Utilization (% of total time)**

- 1 Emac
- 2 Emac

**Power (mW)**

- 2 Emac
- 1 Emac
Power Architecture Evaluation Kit V1.0

- **Description**
  - The Power™ Evaluation Kit (PEK) is an SoC Analysis framework which includes a toolkit and a set of architecture models that are designed to enable embedded software development and performance analysis for consumer applications based on the PowerPC Architecture.

- **Use Scenario**
  - Enables IP and tool providers to package, embed and distribute IP and knowledge that is ready for use by SoC developers.

- **Relevance**
  - Enables designers to evaluate, build and verify SoC designs using Architecture analysis models and tools using an industry standard modeling language enabling integration of third party IP.

- **Release mechanism**
  - Will be announced at the Power.org event in Shanghai on September 26th and to be made available through Developer works.

- **Features**
  - IBM SystemC Models for PPC4xx and CoreConnect IP Cores
  - 60-days evaluation licensed for a limited capacity version of IBM ChipBench System Level Design tool
  - IBM RISCWatch debugger tools
  - Pre-integrated SoC platform optimized for architecture exploration and extensions
  - Sample application code with tutorial
  - Enabled for GCC tool chain
  - Reference SW API documentation
  - Technical support via IBM DeveloperWorks
  - Reference technical manual
Conclusions

- Integrated tool suite along with System Level IP supporting the PowerPC and CoreConnect Architectures.
  - Performance, power analysis of embedded software, co-simulation capability, along with core stitching & physical prototyping early in the design flow.

- Abstract Input Specification. Easy to write and change. Minimum complexity
  - Analysis algorithms that understand this abstract spec (through models)

- Complete system optimization possibilities provided through integrated early analysis with strong links to implementation.
# PowerPC Evaluation Kit Contacts

## Sales

<table>
<thead>
<tr>
<th>Name</th>
<th>Title</th>
<th>Contact Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Craig Wentzel</td>
<td>VP Sales</td>
<td>1-845 892 3380</td>
</tr>
<tr>
<td></td>
<td></td>
<td><a href="mailto:cwentzel@us.ibm.com">cwentzel@us.ibm.com</a></td>
</tr>
<tr>
<td>Tom Osterday</td>
<td>Sales Leader &amp; Managing Principal</td>
<td>1-602-217-2499</td>
</tr>
<tr>
<td></td>
<td></td>
<td><a href="mailto:osterday@us.ibm.com">osterday@us.ibm.com</a></td>
</tr>
<tr>
<td>Bill Pilkington</td>
<td>Sales Leader &amp; Managing Principal</td>
<td>1-845-892-4007</td>
</tr>
<tr>
<td></td>
<td></td>
<td><a href="mailto:wpilking@us.ibm.com">wpilking@us.ibm.com</a></td>
</tr>
<tr>
<td>William Kou</td>
<td>Sales Leader &amp; Managing Principal</td>
<td>1-408-718-3722</td>
</tr>
<tr>
<td></td>
<td></td>
<td><a href="mailto:kou@us.ibm.com">kou@us.ibm.com</a></td>
</tr>
</tbody>
</table>

## Product Development

<table>
<thead>
<tr>
<th>Name</th>
<th>Title</th>
<th>Contact Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dale Hoffman</td>
<td>Director Business Development &amp; CTO</td>
<td>1-845 892 4346</td>
</tr>
<tr>
<td></td>
<td></td>
<td><a href="mailto:daleh@us.ibm.com">daleh@us.ibm.com</a></td>
</tr>
<tr>
<td>Tom Wilson, Power</td>
<td>Technology Offerings Program Dir.</td>
<td>1-802 769 7019</td>
</tr>
<tr>
<td>Wilson, Power</td>
<td></td>
<td><a href="mailto:tjwilson@us.ibm.com">tjwilson@us.ibm.com</a></td>
</tr>
</tbody>
</table>