Technology Advantages

- No basic change in design methodology required
  - ESE supported design follows present design process
- Productivity gain of more than 1000X demonstrated
  - Designers do not write models
- Simple design update: 1-day change
  - No rework for new design decisions
- High error-reduction: Automation + verification
  - Error-prone tasks are automated
- Simplified globally-distributed design
  - Fast exchange of design decisions and easy impact estimates
- Benefit through derivatives designs
  - No need for complete redesign
- Better market penetration through customization
- Shorter Time-to-Market through automation
ES Environment

ESE Front – End
System Capture + Platform Development

Timed TLM

ESE Back – End
SW Development + HW Development

C + RTL

Decision User Interface (DUI)
- Create
- Select
- Partition
- Map
- Compile
- Replace

Validation User Interface (VUI)
- Compiler
- Debugger
- Stimulate
- Verify
- TIMED
- CYCLE ACCURATE

Application Tools: Compilers/Debuggers
Commercial Tools: FPGA, ASIC
Model Accuracy vs. Execution Time

TLM: Transaction Level Model
ISM: Instruction Set Model
PCAM: Pin/Cycle Accurate Model

Time and accuracy trade off among different models
Platform Architecture

Components:
- Processors
- Memories
- IPs
- Custom HW
- Buses
- Bridges

Components:
- Processors
- Memories
- IPs
- Custom HW
- Buses
- Bridges

Components:
- Processors
- Memories
- IPs
- Custom HW
- Buses
- Bridges
Application Spec

Computation
• Processes (in C)

Communication
• Channels (in C)
• Variables (in C)

P1 → P2 → v1 → C2 → P3 → P4
System Definition = Platform + Application
Output: SystemC Timed TLM

TLM Generation Technique
- Application code → sc_thread
- Processing element → sc_module
- Bus → sc_channel
- Memory → Array inside sc_module
- Bridge → FIFO channel + sc_process
System Modifications

TLM is generated/upgraded automatically with changes in Spec or Platform, including:
  * Software changes
  * Hardware changes
  * Communication changes
Output: Modified TLM

CPU

Mem

Bridge

CPU Bus

IP Bus

HW

IP

P1 P2

P5

OS

P6 P3

P4
TLM Generation Features

• **Processing Elements (PEs)**
  - Any number of processes mapped to any PE
  - Any number of bus connections

• **Connectivity**
  - Point-to-point links
  - Shared bus architecture
  - Multi-hop transactions
  - NoC platforms

• **Bridges and routers**
  - Any size, number and partition of FIFOs
  - Any number of bus connections
  - Static and dynamic routing

• **Memories**
  - Any number of bus connections
  - Local (inside PE) and shared memories
Timing Estimation Technique

- DFG scheduling to compute basic block delay
- RTOS model added for PEs with multiple processes
Timing Estimation Features

• Retargetable Processor Models
  • Any type of control/datapath pipelining
  • Any number of pipelined datapaths
  • Multi-cycle units, forwarding, chaining
  • Branch prediction
  • VLIW and SuperScalar

• Statistical/Dynamic Cache Models

• RTOS models

• Integration with high level synthesis for custom HW

• Estimation reports
  • Basic block level, function level and transaction level
ESE: Platform and Application Capture
ESE: TLM Generation and Estimation
MP3 Decoder Application

- Functional block diagram (major blocks only)

  ![Diagram showing the block diagram of the MP3 decoder.]

- Application features
  - 12K lines of C code
  - IMDCT and DCT are compute intensive
    - Candidates for HW implementation
  - Left channel and right channel are data independent
    - Concurrent execution possible
MP3 Platforms

- MP3 Decoder on Xilinx Multimedia FPGA board
  - Microblaze soft-core with 0/1/2/4 HW components
Results: Functional TLM Generation and Simulation

- Functional TLM generation in seconds vs. weeks of manual coding
  - Huge productivity gain
- Functional TLM simulation in fraction of a second
  - Early application development and debugging

<table>
<thead>
<tr>
<th>Design</th>
<th>SystemC LoC</th>
<th>Manual Coding</th>
<th>Func. TLM Generation</th>
<th>Func. TLM Simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>2095</td>
<td>2 weeks</td>
<td>0.63 s</td>
<td>0.01 s</td>
</tr>
<tr>
<td>M2</td>
<td>2894</td>
<td>3 weeks</td>
<td>0.66 s</td>
<td>0.01 s</td>
</tr>
<tr>
<td>M3</td>
<td>3148</td>
<td>4 weeks</td>
<td>0.66 s</td>
<td>0.01 s</td>
</tr>
<tr>
<td>M4</td>
<td>3653</td>
<td>4 weeks</td>
<td>0.74 s</td>
<td>0.01 s</td>
</tr>
<tr>
<td>Average</td>
<td>2948</td>
<td>~3 weeks</td>
<td>~ 0.7 s</td>
<td>0.01 s</td>
</tr>
</tbody>
</table>
Results: Estimation Quality

Error % = (1 - Estimated cycles / Board Cycles) * 100

<table>
<thead>
<tr>
<th>Cache size</th>
<th>M1 Board</th>
<th>ISM Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0K/0K</td>
<td>27215K</td>
<td>39.48%</td>
</tr>
<tr>
<td>2K/2K</td>
<td>8914K</td>
<td>18.38%</td>
</tr>
<tr>
<td>8K/4K</td>
<td>5828K</td>
<td>3.55%</td>
</tr>
<tr>
<td>16K/16K</td>
<td>4413K</td>
<td>-16.32%</td>
</tr>
<tr>
<td>32K/16K</td>
<td>4384K</td>
<td>-16.60%</td>
</tr>
<tr>
<td>Average</td>
<td>N/A</td>
<td>18.86%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cache Size</th>
<th>M1</th>
<th>M2</th>
<th>M3</th>
<th>M4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0K/0K</td>
<td>6.27%</td>
<td>9.00%</td>
<td>18.18%</td>
<td>18.61%</td>
</tr>
<tr>
<td>2K/2K</td>
<td>6.68%</td>
<td>-7.16%</td>
<td>-15.79%</td>
<td>-9.35%</td>
</tr>
<tr>
<td>8K/4K</td>
<td>4.74%</td>
<td>9.13%</td>
<td>-1.66%</td>
<td>-0.18%</td>
</tr>
<tr>
<td>16K/16K</td>
<td>-13.83%</td>
<td>4.66%</td>
<td>2.63%</td>
<td>3.65%</td>
</tr>
<tr>
<td>32K/16K</td>
<td>-13.89%</td>
<td>-8.29%</td>
<td>1.57%</td>
<td>2.29%</td>
</tr>
<tr>
<td>Average</td>
<td>9.08%</td>
<td>7.65%</td>
<td>7.97%</td>
<td>6.82%</td>
</tr>
</tbody>
</table>

- TLM estimation applicable to all designs
  - ISM only available for SW
- TLM estimation error < ½ of ISM error
  - Reliable design exploration with timed TLMs
### Results: Timed TLM Generation and Simulation

<table>
<thead>
<tr>
<th>Timed TLM Generation</th>
<th>Timed TLM Simulation</th>
<th>ISM Sim.</th>
<th>CA Sim.</th>
</tr>
</thead>
<tbody>
<tr>
<td>31 s</td>
<td>0.01 s</td>
<td>3.6 h</td>
<td>16 h</td>
</tr>
<tr>
<td>50 s</td>
<td>0.22 s</td>
<td>N/A</td>
<td>18 h</td>
</tr>
<tr>
<td>47 s</td>
<td>0.25 s</td>
<td></td>
<td>18 h</td>
</tr>
<tr>
<td>71 s</td>
<td>0.36 s</td>
<td></td>
<td>18 h</td>
</tr>
<tr>
<td>~ 1 min</td>
<td>~ 0.2 s</td>
<td>3.6 h</td>
<td>~ 18 h</td>
</tr>
</tbody>
</table>

- **Timed TLM generated in minutes vs. hours of CA/ISM simulation**
  - Early SW/HW performance estimation
- **Timed TLM simulation in < 1 sec.**
  - Extensive design exploration
ESE Advantages

- Platform and Application can be easily captured using GUI
- Functional TLMs are automatically generated for development and testing of application code
- Timed TLMs are automatically generated for early design exploration
- Legacy SW and HW IPs can be easily added for design reuse and upgrade
- ESE allows concurrent development of platform SW, HW and application code
Acknowledgments

• We would like to acknowledge the previous R&D teams who contributed many concepts and methods used in ESE 2.0
  • SpecCharts/SpecSyn ('92): F. Vahid, S. Narayan, J. Gong, S. Bakshi
  • SpecC/SCE ('00) team: R. Doemer, J. Zhu, A. Gerstlauer, J. Peng, D. Shin, L. Cai, H. Yu

• We also want to thank P. Chandraiah for MP3 reference code