CoDEx: CoDesign for Exascale

David Donofrio
Cy Chan, Farzad Fatollahi-Fard, George Michelogiannakis, John Shalf
ModSim
August 13, 2014
Objective
Create a comprehensive architectural simulation platform to accelerate hardware/software co-design for exascale computing
CoDEx Objective
Create a comprehensive architectural simulation platform to accelerate hardware/software co-design for exascale computing

Code Analysis
Static analysis, basic speeds and feeds, unbounded hardware models

Rapid Exploration
Use software based simulators and software kernels to explore hardware parameter space

Synthesis of Point Design
Use hardware emulation tools for full application optimization and extraction of accurate power and area estimates
CoDEx Tool Summary

Tools span from high-level, concept exploring tools to physical circuit synthesis

- **Concept**
  - ExaSAT and spreadsheet models

- **Software Simulation**
  - OpenSoC, DRAMSim, NVRAMSim, HMC Models, CHISEL generated components, Tensillica Xtensa models, SystemC based models from industry

- **Hardware Emulation**
  - Gateware, CHISEL generated components, NoCSim, Tensillica Xtensa processors

- **Circuit Synthesis**
  - FPGAs, ASIC, Power model parameter extraction
Conceptual Analysis
## The Old Way of Analytic Modeling

<table>
<thead>
<tr>
<th>Module</th>
<th>Read-only</th>
<th>Write-only</th>
<th>first read, then written</th>
<th>first written then read</th>
<th>Vars that can be read from registers after written</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ctoprim</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ctoprim(t)</td>
<td>loop1</td>
<td>U1-U5</td>
<td>Q1,Q5,Q6</td>
<td>Q2,Q3,Q4</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>loop2</td>
<td>Q1-Q5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ctoprim(f)</td>
<td>loop1</td>
<td>U1-U5</td>
<td>Q1,Q5,Q6</td>
<td>Q2,Q3,Q4</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>differm</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ux,vx,wx</td>
<td>loop1</td>
<td>Q2, Q3, Q4</td>
<td>ux, vx, wx</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>uy,vy,wy</td>
<td>loop2</td>
<td>Q2, Q3, Q4</td>
<td>uy, vy, wy</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>uz,vz,wz</td>
<td>loop3</td>
<td>Q2, Q3, Q4</td>
<td>uz, vz, wz</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>imx</td>
<td>loop4</td>
<td>Q2, vy, wz</td>
<td>D2</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>imy</td>
<td>loop5</td>
<td>Q3, ux, wz</td>
<td>D3</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>imz</td>
<td>loop6</td>
<td>Q4, ux, vy</td>
<td>D4</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>iene</td>
<td>loop7</td>
<td>Q2-4,Q6, D2-4</td>
<td>D5</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ux-z, vx-z, wx-z</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>hypterm</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>loop1</td>
<td>U2-U5, Q2, Q5</td>
<td>F1-F5</td>
<td>Yes</td>
<td>F1-F5 across loops</td>
</tr>
<tr>
<td></td>
<td>loop2</td>
<td>U2-U5, Q3, Q5</td>
<td>F1-F5</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>loop3</td>
<td>U2-U5, Q4, Q5</td>
<td>F1-F5</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td><strong>CalcU</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N+1/3</td>
<td>loop1</td>
<td>U, D, F</td>
<td>Unew</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>N+2/3</td>
<td>loop2</td>
<td>U, D, F</td>
<td>Unew</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>N+1</td>
<td>loop3</td>
<td>Unew, D, F</td>
<td>U</td>
<td>No</td>
<td></td>
</tr>
</tbody>
</table>
# The Old Way of Analytic Modeling

<table>
<thead>
<tr>
<th>Module</th>
<th># of Reads pt</th>
<th># of Reads w halo/pt</th>
<th># of Writes /pt</th>
<th>in bytes</th>
<th>in bytes</th>
<th>in bytes</th>
<th>in bytes</th>
<th>in bytes</th>
<th>Total Memory Access</th>
<th>in MB</th>
</tr>
</thead>
<tbody>
<tr>
<td>ctoprim</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>courno=true</td>
<td>loop1</td>
<td>5</td>
<td>0</td>
<td>6</td>
<td>1310720</td>
<td>0</td>
<td>1572864</td>
<td>2883584</td>
<td>2.75</td>
<td></td>
</tr>
<tr>
<td></td>
<td>loop2</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>1310720</td>
<td>0</td>
<td>0</td>
<td>1310720</td>
<td>1.25</td>
<td></td>
</tr>
<tr>
<td>courno=false</td>
<td>loop1</td>
<td>5</td>
<td>0</td>
<td>6</td>
<td>1310720</td>
<td>0</td>
<td>1572864</td>
<td>2883584</td>
<td>2.75</td>
<td></td>
</tr>
<tr>
<td>diffterm</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ux,vx,wx</td>
<td>loop1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>262144</td>
<td>262144</td>
<td>0.25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>uy,vy,wy</td>
<td>loop2</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>1376256</td>
<td>786432</td>
<td>2162688</td>
<td>2.0625</td>
<td></td>
</tr>
<tr>
<td>uz,vz,wz</td>
<td>loop3</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>1376256</td>
<td>786432</td>
<td>2162688</td>
<td>2.0625</td>
<td></td>
</tr>
<tr>
<td>imx</td>
<td>loop4</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>1376256</td>
<td>262144</td>
<td>1638400</td>
<td>1.5625</td>
<td></td>
</tr>
<tr>
<td>imy</td>
<td>loop5</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>1376256</td>
<td>262144</td>
<td>1638400</td>
<td>1.5625</td>
<td></td>
</tr>
<tr>
<td>imz</td>
<td>loop6</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>1376256</td>
<td>262144</td>
<td>1638400</td>
<td>1.5625</td>
<td></td>
</tr>
<tr>
<td>iene</td>
<td>loop7</td>
<td>15</td>
<td>1</td>
<td>1</td>
<td>3932160</td>
<td>458752</td>
<td>262144</td>
<td>4653056</td>
<td>4.4375</td>
<td></td>
</tr>
<tr>
<td>hypterm</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>loop1</td>
<td>0</td>
<td>6</td>
<td>5</td>
<td>0</td>
<td>2752512</td>
<td>1310720</td>
<td>4063232</td>
<td>3.875</td>
<td></td>
</tr>
<tr>
<td></td>
<td>loop2</td>
<td>5</td>
<td>6</td>
<td>5</td>
<td>1310720</td>
<td>2752512</td>
<td>1310720</td>
<td>5373952</td>
<td>5.125</td>
<td></td>
</tr>
<tr>
<td></td>
<td>loop3</td>
<td>5</td>
<td>6</td>
<td>5</td>
<td>1310720</td>
<td>2752512</td>
<td>1310720</td>
<td>5373952</td>
<td>5.125</td>
<td></td>
</tr>
<tr>
<td>CalcU</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N+1/3</td>
<td>loop1</td>
<td>15</td>
<td>0</td>
<td>5</td>
<td>3932160</td>
<td>0</td>
<td>1310720</td>
<td>5242880</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>N+2/3</td>
<td>loop2</td>
<td>20</td>
<td>0</td>
<td>5</td>
<td>5242880</td>
<td>0</td>
<td>1310720</td>
<td>6553600</td>
<td>6.25</td>
<td></td>
</tr>
<tr>
<td>N+1</td>
<td>loop3</td>
<td>20</td>
<td>0</td>
<td>25</td>
<td>5242880</td>
<td>0</td>
<td>6553600</td>
<td>11796480</td>
<td>11.25</td>
<td></td>
</tr>
</tbody>
</table>
CoDEx Tools: ExaSAT
Compiler driven performance analysis framework

- Extracts key application statistics in HW independent fashion
  - HW configs parameterized for code performance estimation
- Estimate performance benefits of code transforms \textit{without} changing the code
ExaSAT Analysis
Automate a lot of tedious analysis

Cache Blocking on Dynamics Kernel (53 species)

ExaSAT State Variable Analysis

Unique Variable IDs (sorted by number of accesses)
Energy Efficiency Analysis
Illustrate the benefit of large L1 scratchpads

![Graph showing Byte to Flop Ratios vs Cache Size for Loop Fusion Scenarios](image1)

![Graph showing Power Reduction Opportunity](image2)

![Graph showing Power consumed by large caches](image3)

![Graph showing Power w/DRAM](image4)
Hardware and Software Models
Processor, network, and memory models in software and hardware emulation.
CoDEx Tools: Putting Chisel to work for DOE

A path to hardware and software models

- New hardware DSL
- Scala based
  - Powerful generators
  - Obj Oriented constructs
- Generate C++ and Verilog models from single description
- Glue to existing infrastructure with SystemC
CoDEx Interoperability
SystemC as a platform for re-use

- **SystemC embraced by**
  - Industrial partners
    - Intel, Micron, ARM, Cadence, Synopsis
  - Other research efforts
    - CAL, FastForward
- **All CoDEx software simulation tools based on SystemC**
CoDEx Tools: FPGA Emulation

Enabling full application optimization

- 1000x Faster than SW models
  - Scales independent of processor count
- CoDEx Software models have parallel HW emulation path
- Performance and Energy counters provide SW model level of detail
CoDEx Tools: OpenSoC Fabric
Parameterized NoC generation tool

- Leverage Chisel to create highly parameterized, flexible model for NoC generation
  - Dimensions, topology, VCs, etc. all configurable
  - Fast, functional SW model with SystemC integration
  - Verilog model for FPGA and ASIC flows
- AXI Based Interface
  - Integrate with Tensilica as well as ARM based cores
- Builds on previous PhoenixSim network model
OpenSoC Fabric

Flits moving through a mesh network
CoDEx Tools: NVRAMSim
Non-Volatile Memory Modeling

- Collaboration with Myoungsoo Jung UT Austin
- Build on prior work of page/block addressable NVRAM simulator
  - Extend to byte / word addressability
  - Alternative memory cell architecture
- Dynamic energy model
  - Aware of internal NVRAM components with variable clock frequencies
- Validated against hardware evaluation platform
- Integrates with CoDEx processor and NoC models
  - Supplements existing CoDEx DRAM model (DRAMSim2)
NRAMsim: Alternative Burst Buffer RRAM Organization
CoDEx Tools: Processor Models
Highly configurable XTensa embedded core

› Tensilica XTensa processor generator
  • Fast configuration of cache, local store and easily extensible ISA
  • Highly flexible FIFO based interfaces (TIEQueues)
  • Rich performance counter and debug interface

› All custom cores generated with
  • Fast functional model
  • SystemC based performance model
  • Verilog implementation ready for FPGA or ASIC flow

› Verified power model

› Integrates with all CoDEx hardware and software models
CoDEx Summary

- CoDEx tools span range of abstraction levels
- Embrace SystemC as common glue
  - Enables interoperation with DOE and industrial models
- Validated processor model
- DRAM and NVRAM models
- Parameterized NoC generator
- Parallel modeling path includes
  - Flexible software models
  - High-speed FPGA based emulation
CoDEx: Summary

- **Key Contribution:**
  - Providing an integrated environment from static analysis to hardware synthesis

- **Interoperability Biggest Challenge**
  - CoDEx designed to be a collection of tools that stand alone or can be combined to create something more powerful

- **Opportunities exist to leverage capabilities of other simulation environments**