Rethink Hardware-Software Interface for Power-Aware Computing
Conventional Architectures only Expose Performance

Current RISC/VLIW ISAs only expose hardware features that affect critical path through computation
Energy Consumption is Hidden

Most energy is consumed in microarchitectural operations that are hidden from software!
Reward compile-time knowledge with run-time energy savings

- hardware provides mechanisms to disable microarchitectural activity, a software power grid
- compile-time analysis determines which pieces of microarchitecture can be disabled for given application

⇒ Co-develop energy-exposed architectures and energy-conscious compilers
Energy Management Layers

PACE Focus Areas

- Application
- Algorithm
- Source Code
- Compiler
- Run-Time/O.S.
- Instruction Set
- Microarchitecture
- Circuit Design
- Fabrication Technology
SCALE Strawman Processor

- 32 processing tiles
- Fast on-chip data network
- 128x32b FLOP/cycle total
- 4096x8b OP/cycle total
- 128MB on-chip DRAM/16MB SRAM
- External DRAM interface
- Chip-to-chip interconnect channels
- 20x20mm² in 0.1μm CMOS
Vector
- most streaming applications highly vectorizable
- vectors reduce instruction fetch/decode energy up to 20-60x (depends on vector length)
- mature programming and compilation model
⇒ **SCALE supports vectors in hardware**
- address and data units optimized for vectors
- hardware vector control logic

VLIW/Reconfigurable
- exploit instruction-level parallelism for non-vectorizable applications
- superscalar ILP expensive in hardware
⇒ **SCALE supports VLIW-style ILP**
- reuse address and data unit datapath resources
- expose datapath control lines
- single wide instruction = configuration
- provide control/configuration cache distributed along datapaths

Multithreading/CMP
- run separate threads on different tiles
- any mix of vector or VLIW across tiles
SCALE Exposes Locality at Multiple Levels

- 2D Tile and DRAM layout
  - software maps computation to minimize network hops

- Local SRAM within tile
  - software split between instruction/data/unified storage
  - software scratchpad RAMs or hardware-managed caches

- Distributed cached control state within tile
  - control unit: instruction buffer
  - data/address unit: vector instructions or VLIW/configuration cache

- Distributed register file and ALU clusters within tile
  - Control Unit: scalar (C) registers versus branch (B) registers
  - Address Unit: address (A) registers
  - Data Unit: Four clusters of data registers (D0-D4)
  - Accumulators and sneak paths to bypass register files
- Turn off unused register banks and ALUs
- Reduce datapath width
  - set width separately for each unit in tile (e.g., 32b in control unit, 16b in address unit, 64b in data unit)
- Turn off individual local memory banks
- Configure memory addressing model
  - From hardware cache-coherence to local scratchpad RAM
- Turn off idle tiles and idle inter-tile network segments
- Turn off refresh to unused DRAM banks
Existing Infrastructure

- RAW Compiler Technology
  - SUIF-based C/FORTRAN compiler for tiled arrays
  - SPAN pointer analysis
  - Bitwise bitwidth analysis
  - Superword Level Parallelism
  - Space/Time scheduling
  - MAPS compiler-managed memory system

- Pekoe Low-Power Microprocessor Library Cells
  - Full-custom processor blocks in 0.25µm CMOS process
  - Designed for voltage-scaled operation

- SyCHOSys Energy-Performance Simulator
  - Fast, multi-level compiled simulation
  - Energy models for Pekoe processor blocks
Bitwidth Analysis

- Compile-time detection of minimum bitwidth required for each variable at every static location in the program
- A collection of techniques
  - Arithmetic operations
  - Boolean operations
  - Bitmask operations
  - Loop induction variable bounding
  - Clamping optimization
  - Type promotion
  - Back propagation
  - Array index optimization
- Value-range propagation using data-flow analysis
- Loop analysis
- Incorporated pointer alias analysis
- Paper in PLDI’00
Methodology

- C → RTL
- RTL simulation gives switching
- Synthesis tool reports dynamic power
- IBM SA27E process, 0.15μm drawn, 200 MHz
SyCHOSys compiles a custom cycle simulator from a structural machine description

- Supports gate level to behavioral level, or any mixture
  - Behavior specified in C++, compiles to C++ object

- Can selectively compile in transition counting on nets
  - Automatically factors out common counts for faster simulation

- Arbitrary energy models for functional units/memories
  - Capacitances extracted from circuit layout or estimated
  - Use fast bit-parallel structural energy models (much faster than lookups)

- Paper in Complexity-Effective Workshop, ISCA’00
SyCHOSys Evaluation

- GCD circuit benchmark
  - full-custom datapath layout (0.25µm TSMC CMOS process)
  - mixture of static and precharged blocks

<table>
<thead>
<tr>
<th></th>
<th>Simulation Speed (Hz)</th>
<th>Error in power prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>C-Behavioral (gcc)</td>
<td>109,000,000.00</td>
<td>N/A</td>
</tr>
<tr>
<td>Verilog-Behavioral (VCS)</td>
<td>544,000.00</td>
<td>N/A</td>
</tr>
<tr>
<td>Verilog-Structural (VCS)</td>
<td>341,000.00</td>
<td>N/A</td>
</tr>
<tr>
<td>SyCHOSys-Structural</td>
<td>8,000,000.00</td>
<td>N/A</td>
</tr>
<tr>
<td>SyCHOSys-Power</td>
<td>195,000.00</td>
<td>0.5% - 8.2%</td>
</tr>
<tr>
<td>PowerMill (extracted layout)</td>
<td>0.73</td>
<td>7.2% - 13.7%</td>
</tr>
<tr>
<td>Star-Hspice (extracted layout)</td>
<td>0.01</td>
<td>0%(reference)</td>
</tr>
</tbody>
</table>
SyCHOSys Processor Model

- Five-stage pipelined MIPS RISC processor+caches
- User/kernel mode, precise interrupts, validated with architectural test suite+random test programs
- Runs SPECint95 benchmarks
- Simulation speeds (Sun Ultra-5, 333MHz workstation)
  - (ISA-level interpreter 3 MHz)
  - Behavioral RTL 400kHz
  - Structural model 40kHz
  - Energy model 16kHz
- A Gigacycle/CPU-day or Megacycle/CPU-minute with better accuracy than Powermill
PACE Milestones

- **Year 2000: Baseline design**
  - Baseline SCALE architecture definition
  - RAW compiler generating code for baseline SCALE design
  - Baseline SCALE architecture energy-performance simulator

- **Year 2001: Single tile**
  - Energy-exposed SCALE tile architecture definition
  - Energy-conscious compiler passes for SCALE tile
  - Energy-exposed SCALE tile energy-performance simulator
  - Evaluation of energy-exposed SCALE tile

- **Year 2002: Multi-tile**
  - Energy-exposed SCALE multi-tile architecture definition
  - Multi-tile energy-performance simulator
  - Multi-tile energy-conscious compiler passes
  - Evaluation of multi-tile SCALE processor

- (Options: Fabricate SCALE prototype)