TLM POWER3

SystemC TLM Power Library

PEHAM Project: Power estimation from high-level models

David Greaves
MM Yasin
(+ M Pusovic)

University of Cambridge
Computer Laboratory

FDL, September 2012, Vienna.
TLM POWER 3: Motivation

- Power estimation from high-level models.
- Rapid architectural exploration using SystemC.
- **Absolute accuracy goal:** correct order of magnitude at least!
- **Relative accuracy goal:** 20 percent or so.

- Want correct polarity of the parameter derivatives: *A change is better or worse!*
Talk Overview

- TLM POWER 2
- TLM POWER 3
  - Loose timing
  - Energy based
  - Layout aware
  - Bit transition counting
- Splash-2 benchmarks, power probed.
- Data fit x86_64 to OpenRISC!
TLM Power 2 Library

- TLM POWER 2 developed at France CEA (Lebreton/Vivet)
  - Used phase/mode modelling
  - No LT
  - No TLM socket integration

```java
class FOO:
    public sc_module,
    public pw_module
{
    public:
        SC_HAS_PROCESS(FOO);
        FOO(const sc_module_name& p_name):
            sc_module(p_name),
            pw_module("config.txt")
        {
            SC_THREAD(process);
        }
    
    void process(void)
    {
        update_power (PW_MODE_ON, PW_PHASE_IDLE);
        wait(10, SC_NS);
        // Perform some computation
        update_power(PW_MODE_ON, PW_PHASE_COMPUTE);
        wait(20, SC_NS);

        update_power(PW_MODE_OFF);       // Turn off module
    }

};
```
Physical Units

- SystemC provides overloaded sc_time units
- TLM POWER 2 added pw_energy and pw_power units with all appropriate overloads.
- TLM POWER 3 adds pw_voltage for F/V scaling.
- TLM POWER 3 also adds pw_length and pw_area.

Basic physics: energy divided by time ---> power

Basic physics: length times length ---> area
Records, Accounts and Observers

• Every monitored module is tied to a *power record*
  • by inheritance or
  • by SystemC attribute.

• Every power record contains a set of accounts.

• Accounts have common (user-defined) names and purposes across the system. *Typically:*
  • A1 Static power
  • A2 Dynamic energy
  • A3 Wiring energy

• Each account can track both energy and power.

• An *observer* sums activity in a collection of records keeping accounts separate

• A report file has one observer per line.
bad:
This shows computation of energy per transaction in the body of the transaction.

better:
Energy and floating point computations done in RECOMPUTE_PVT callback.
Static or Initial Parameters (1)

class FOO:
    public sc_module,
    public pw_module
{
    public:
        SC_HAS_PROCESS(FOO);
        FOO(const sc_module_name& p_name, int width):
            sc_module(p_name),
            pw_module("config.txt")
        {
            set_excess_area(pw_length(50.0 * width, PW_um), pw_length(5.0 * width, PW_um));
        }
};
Static or Initial Parameters (2)

- Set up static parameters in constructor:
  - Excess or actual area or dimensions
  - Static power consumption
  - Chip/region name
  - VCC supply voltage

- Optional per-instance or per-kind technology file (XML) can be accessed (defines phases and modes and default VCC ...).

- Some are less static:
  - Set these in PVT change callback (virtual function).
  - Call that yourself from constructor.

- PVT called-back when VCC changes.
Spatial Layout Support

- Every SC_MODULE has a chip/region designation.
- The area of a module is sum of
  - its children with the same chip/region name
  - its locally defined 'excess area'.
- Inter-module wiring lengths can be estimated using Rent's Rule on area of lowest-common-parent.
- Actual X-Y co-ordinates could be allocated by a placer.
Hop Tracking: Origin/Hop/Terminus.

Option 1: Track transaction trajectory to get distance travelled.

trans.pw_set_origin(this, PW_TGP_ADDRESS | PW_TGP_ACCT_SRC, &frontside_bus);
initiator_socket->b_transport(trans, delay);
trans.pw_terminus(this);

- Initiator makes the origin and terminus calls.
- Intermediate nodes (cache and bus models) call log_hop.
- Flags enable energy to be logged at src or dest.
- Options 1+2:
  - For additional transition counting, need to know which bus transaction is on and which fields in TLM payload are active.
Report Formats (3: VCD)

- Each account and their summations can be plotted in various forms
  - 1: Ascii-art table format
  - 2: SYLK or CSV spreadsheet format
  - 3: VCD temporal display (using dirac impulse response or average over interval)
- A physical layout file is also written.
### Report Formats (2: Ascii-art text file)

```plaintext
# TLM POWER3 (Univ Cambridge, UK)
# Statistics file: energy/power consumption.
# For more information see the TLM POWER3 manual pdf.
# Creation Date: 17:27:22 -- 15/09/2012

Title: privmem-cln6000-dramsim-withcache-nile-gash-harvard
# Simulation duration: 24826590001096 ps
# Simulation duration: 24826590001096 ps

<table>
<thead>
<tr>
<th>MODULE NAME</th>
<th>STATIC0 ENERGY</th>
<th>DYNAMIC1 ENERGY</th>
<th>WIRING2 ENERGY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standalone modules:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Memory 0 (DRAM)</td>
<td>0.173879501J</td>
<td>0.0875462788J</td>
<td>4.48687512e-07J</td>
</tr>
<tr>
<td>the_top.uart0</td>
<td>0J</td>
<td>1.644e-06J</td>
<td>6.7041e-11J</td>
</tr>
<tr>
<td>the_top.bsmux0</td>
<td>0J</td>
<td>1.1905216e-05J</td>
<td>0J</td>
</tr>
<tr>
<td>the_top.dram=0</td>
<td>0.173879501J</td>
<td>0.0875462788J</td>
<td>4.48687512e-07J</td>
</tr>
<tr>
<td>...top.coreunit_0.core_0</td>
<td>0.2482659J</td>
<td>0.0044012626J</td>
<td>1.34648772e-05J</td>
</tr>
<tr>
<td>...reunit_0.l1_d_cache_0</td>
<td>0J</td>
<td>0.00594064671J</td>
<td>6.14810556e-06J</td>
</tr>
<tr>
<td>...l1_d_cache_0.Data_0</td>
<td>0.0333542257J</td>
<td>0.000107935695J</td>
<td>0J</td>
</tr>
<tr>
<td>...l1_d_cache_0.Tags_0</td>
<td>0.0317907464J</td>
<td>4.18042825e-05J</td>
<td>0J</td>
</tr>
<tr>
<td>...l1_d_cache_0.Data_1</td>
<td>0.0333542257J</td>
<td>0.000105833853J</td>
<td>0J</td>
</tr>
<tr>
<td>...l1_d_cache_0.Tags_1</td>
<td>0.0317907464J</td>
<td>3.37903219e-05J</td>
<td>0J</td>
</tr>
<tr>
<td>...l1_d_cache_0.Data_2</td>
<td>0.0333542257J</td>
<td>0.000105435493J</td>
<td>0J</td>
</tr>
<tr>
<td>...l1_d_cache_0.Tags_2</td>
<td>0.0317907464J</td>
<td>2.60627187e-05J</td>
<td>0J</td>
</tr>
<tr>
<td>...l1_d_cache_0.Data_3</td>
<td>0.0333542257J</td>
<td>0.000100887529J</td>
<td>0J</td>
</tr>
<tr>
<td>...l1_d_cache_0.Tags_3</td>
<td>0.0317907464J</td>
<td>1.83743234e-05J</td>
<td>0J</td>
</tr>
</tbody>
</table>
```

David GREAVESM/M Yasin, Computer Laboratory, Cambridge. FDL Sept 2012, Vienna
Loosely-Timed: Effect of Quantum

Two cores running: main() { for(i=0;i<5;i++) puts(“Hello World”); }

Core clock is 200 MHz (5ns period).

Global Q = 5ns
Lock-step execution

Global Q = 1us
Finely interleaved

Global Q = 100us
Coarsely interleaved

Three different settings of the global quantum.
Loosely-Timed Performance Lost

Relative performance of LT TLM Model (2 cores, running SPLASH-2 Radix Sort n=100)
Confidence Switcher

Generic API for a measuring and estimating component.

Use for time, energy, transition count and so on ...

Very simple implementation if we just want an estimate of the average metric:

Discard first $N$ measurements, average next $N$, return this value while making an actual measurement one in every $N$ to check for LOSS OF CONFIDENCE.
LT Performance Restored

FIGURE MISSING THIS DRAFT
Augmented DMI Flow

Latency can be credited to the initiating thread's 'delay' as always.

Energy *should* be credited to the intermediate components:

so DMI record at initiator is extended with either
   a) a list of intermediate agents that have their own records
      or
   b) bulk read and write energy records (simpler, not shown).
SMP OpenRISC Reference Design

1 to 64 cores (four shown)
Shared or split or no L1 Cache
Flexible cache architectures
L2 and L3 caches easily added

Detailed model of AMD Opteron L2/Hypertransport
DRAM modelled by Univ Maryland DRAMSIM2
PC CPU Power Probe

USB probe

Measures 12 volt rail to CPU socket.

Measures volts and amps at 10 Hz rate.

Accuracy:
consistency of about 1 percent between runs.
Splash-2 'RADIX' : First Test Setup

Plot shows two runs with two cores and then one run with one core.

Problem: Power probe was running on same machine (spikes).
Problem: Some spikes missed owing to aliasing (missing ADC LPF).
Fixed thereafter (use separate probe machine and add an RC filter).
Also this CPU uses 3x power compared with phenom...
Probed and Probing Machines

AMD 6-Core Phenom 64 Processor with TCP connection to power probe machine.

David GREAVESM/M Yasin, Computer Laboratory, Cambridge. FDL Sept 2012, Vienna
Phenom Corner Cases: 1 to 8 threads

System has 6 cores sharing one DRAM bank.

Vertical bar -> 1 second.
Horizontal scale -> 100 Watts.
Splash-2 'RADIX': Power + Energy

Running the RADIX test on $n = 1$ to 6 cores.

Program modified to suit $n$ not a power of 2.

Increasing $n$ $\rightarrow$ increased performance.

Increasing $n$ $\rightarrow$ better efficiency.

*Strange power humps!* 

One DRAM DIMM shared.
## Phenom Energy Coefficients

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Energy (J)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instruction</td>
<td>1 nJ</td>
</tr>
<tr>
<td>I Cache Miss</td>
<td>50 nJ</td>
</tr>
<tr>
<td>D Cache Miss</td>
<td>15 uJ</td>
</tr>
<tr>
<td>D Cache Snoop Read</td>
<td>4 mJ</td>
</tr>
<tr>
<td>D Cache Evict</td>
<td>7 mJ</td>
</tr>
</tbody>
</table>

Values obtained from curve fitting

CPU + Caches only.

DRAM excluded.

---

David GREAVESM/M Yasin, Computer Laboratory, Cambridge.  FDL Sept 2012, Vienna
Measured v Predicted: Runs 19-24 extrapolated from data fitting on 1-18.
Thankyou for listening

David Greaves
M Yasin
(Milos Pusovic)

University of Cambridge
Computer Laboratory

FDL 2012 Vienna.
Power Estimation: Project Flow

Operating System Fragments

- SPLASH-2 benchmarks
- Application benchmarks
- OPEN MPI benchmarks

C Compilers

- gcc-x86_64
- gcc-x86_64 or gcc-MIPS
- llvm-gcc

Real Hardware

- gcc-x86_64
- Various SMP Linux machines to hand
- DRAM, HDD + CPU power monitoring

SystemC kernel

- ISS MIPS
- ISS x86
- DRAM
- HDD
- Cache(s)
- NoC
- MPI link
- 802 link
- TLM 2.0 library
- TLM power library

POWER RESULTS

PHP Infrastructure: Database, Simulation Platform and HTML/XML Remote Access

Locally-Generated Power Results

Results from literature survey

Third party uploads

Web Data Upload

Web Experiment Submit

Web Data Download
Backup Slide: Loosely-timed TLM Modelling