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

```
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
   }
};
```

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

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.

LT b_transport energy annotation

```
tac_response tac_multiport_router::b_transport(tlm_generic_payload &trans, sc_time &delay)
{
  unsigned int len = trans.get_data_length();

   ... // Main body of the behavioural model

  sc_time activity_time = ...;

  delay += lt_activity_time; // Or use qk_inc to perform this addition

#ifdef TLM_POWER3
  // bit_width has been set in the constructor... etc
  sc_energy energy_cost = pw_energy((double) (5 * len), pw_energy_unit::PW_pJ);
  pw_module_base::update_energy(energy_cost, lt_activity_time);
#endif
}
```

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.

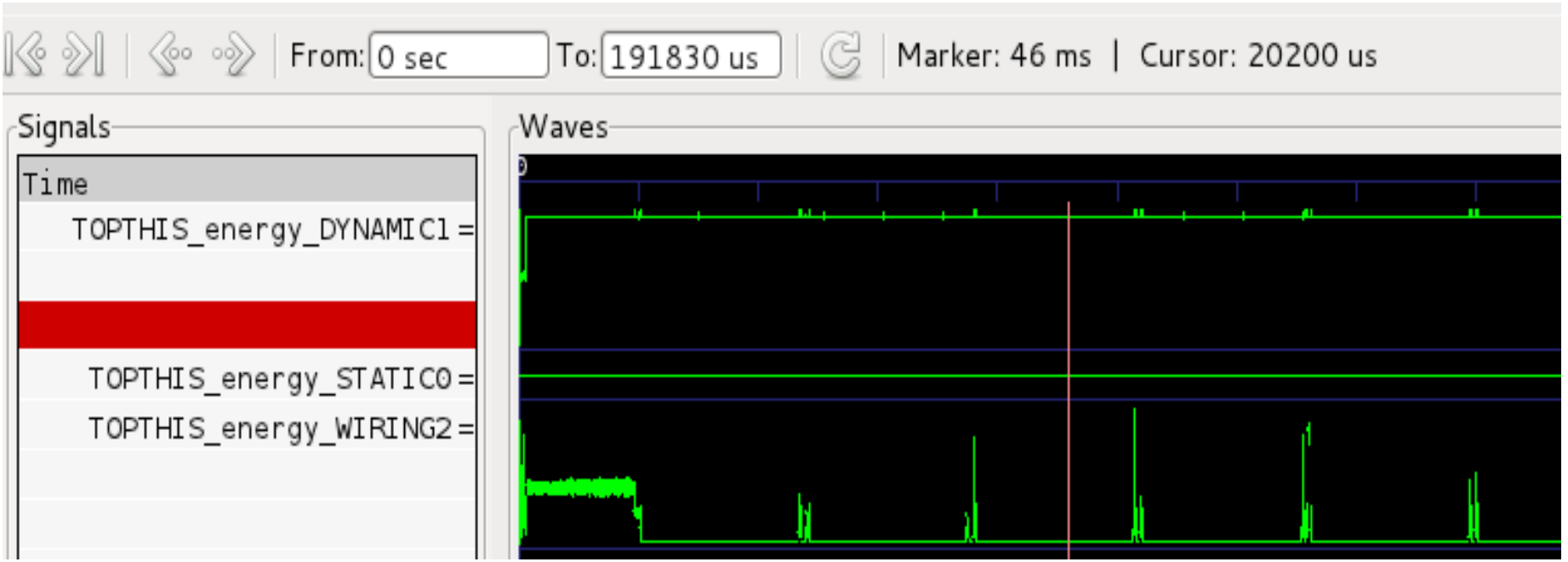
Hop Tracking: Origin/Hop/Terminus.

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)

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


Title: privmem-c1n6000-dramsim-withcache-nile-gash-harvard
# Simulation duration: 24826590001096 ps
# Simulation duration: 24826590001096 ps

+--------------------------+------------------------------+------------------------------+------------------------------+
| MODULE  NAME             |             STATIC0 ENERGY   |            DYNAMIC1 ENERGY   |             WIRING2 ENERGY   |
+--------------------------+------------------------------+------------------------------+------------------------------+
Standalone modules:
| Memory 0 (DRAM)          |       0.173879501J   3.49%   |       0.0875462788J   1.76%  |     4.48687512e-07J   0.00%  |
| the_top.uart0            |                0J   0.00%    |         1.644e-06J    0.00%  |         6.7041e-11J   0.00%  |
| the_top.busmux0          |                0J   0.00%    |     1.1905216e-05J    0.00%  |                0J   0.00%    |
| the_top.dram=0           |       0.173879501J   3.49%   |       0.0875462788J   1.76%  |     4.48687512e-07J   0.00%  |
| ...top.coreunit_0.core_0 |         0.2482659J   4.99%   |       0.0044012626J   0.09%  |     1.34648772e-05J   0.00%  |
| ...reunit_0.l1_d_cache_0 |                0J   0.00%    |    0.000594064671J    0.01%  |     6.14810556e-06J   0.00%  |
| ...0.l1_d_cache_0.Data_0 |      0.0333542257J   0.67%   |    0.000107935695J    0.00%  |                0J   0.00%    |
| ...0.l1_d_cache_0.Tags_0 |      0.0317907464J   0.64%   |     4.18042825e-05J   0.00%  |                0J   0.00%    |
| ...0.l1_d_cache_0.Data_1 |      0.0333542257J   0.67%   |    0.000105833853J    0.00%  |                0J   0.00%    |
| ...0.l1_d_cache_0.Tags_1 |      0.0317907464J   0.64%   |     3.37903219e-05J   0.00%  |                0J   0.00%    |
| ...0.l1_d_cache_0.Data_2 |      0.0333542257J   0.67%   |    0.000105435493J    0.00%  |                0J   0.00%    |
| ...0.l1_d_cache_0.Tags_2 |      0.0317907464J   0.64%   |     2.60627187e-05J   0.00%  |                0J   0.00%    |
| ...0.l1_d_cache_0.Data_3 |      0.0333542257J   0.67%   |    0.000108887529J    0.00%  |                0J   0.00%    |
| ...0.l1_d_cache_0.Tags_3 |      0.0317907464J   0.64%   |     1.83743234e-05J   0.00%  |                0J   0.00%    |
```

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).

```
Sim Start: cores=2
HHelleol lWoo rWlodr
ld
HHeellllloo  WWoorrlldd

HHeellllloo  Wwoorrlldd

HHeellllloo  WWoorrlldd
H
eHlellol oW oWrolrd
ld
CPU 0 exit: insns #717
CPU 1 exit: insns #717
```

Global Q = 5ns
Lock-step execution

```
Sim Start: cores=2
Hello World
HeHello World
Hello World

Hello Woolo World
Hello rld
Hello World
World
Hello Wor
Hello World
CPU 0 exit : insns #717
CPU 1 exit:  insns #717
```

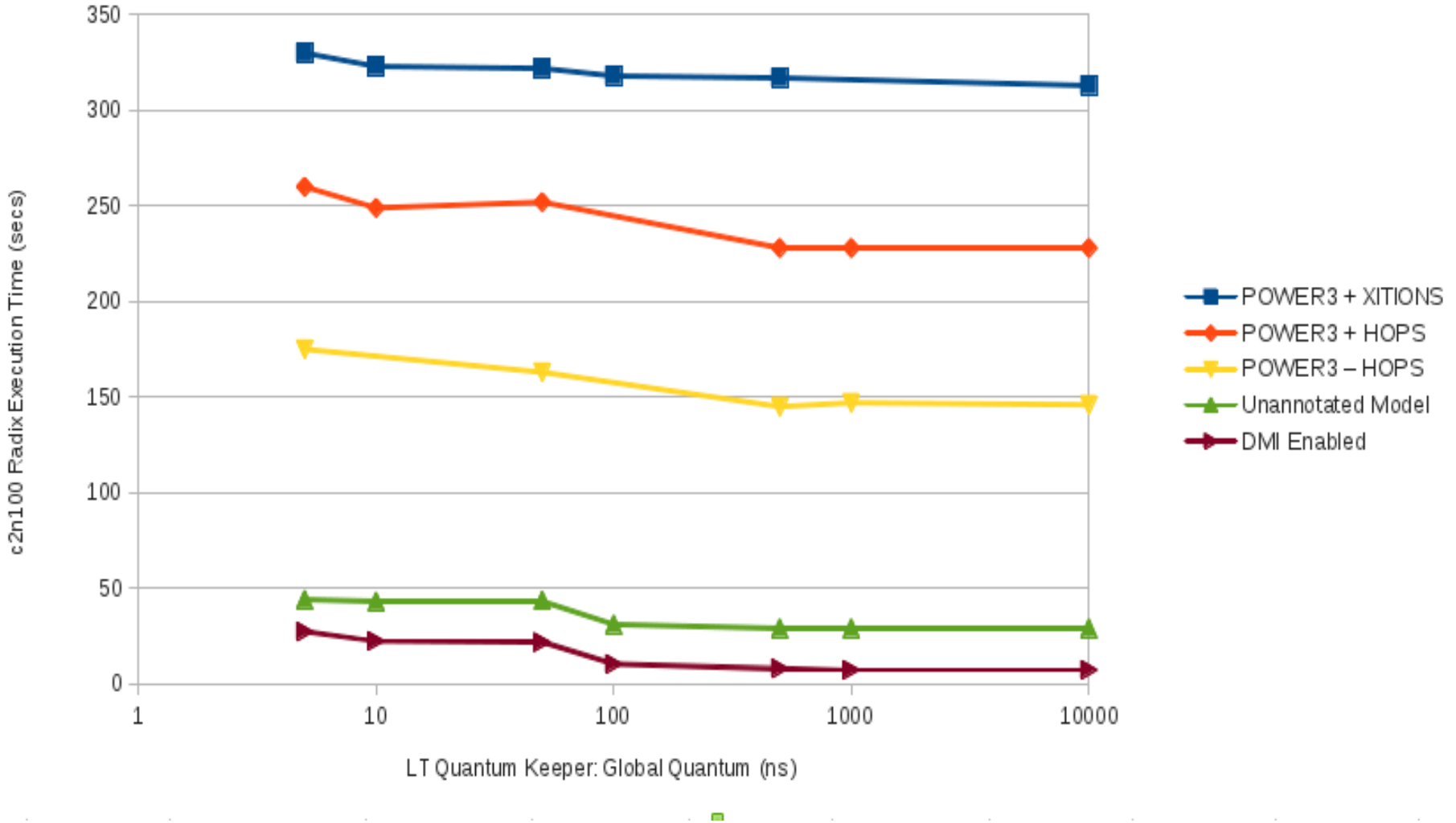
Global Q = 1us
Finely interleaved

```
Sim Start: cores=2
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
CPU 0 exit: insns #717
CPU 1 exit: insns #717
```

Global Q = 100us
Coarsely interleaved

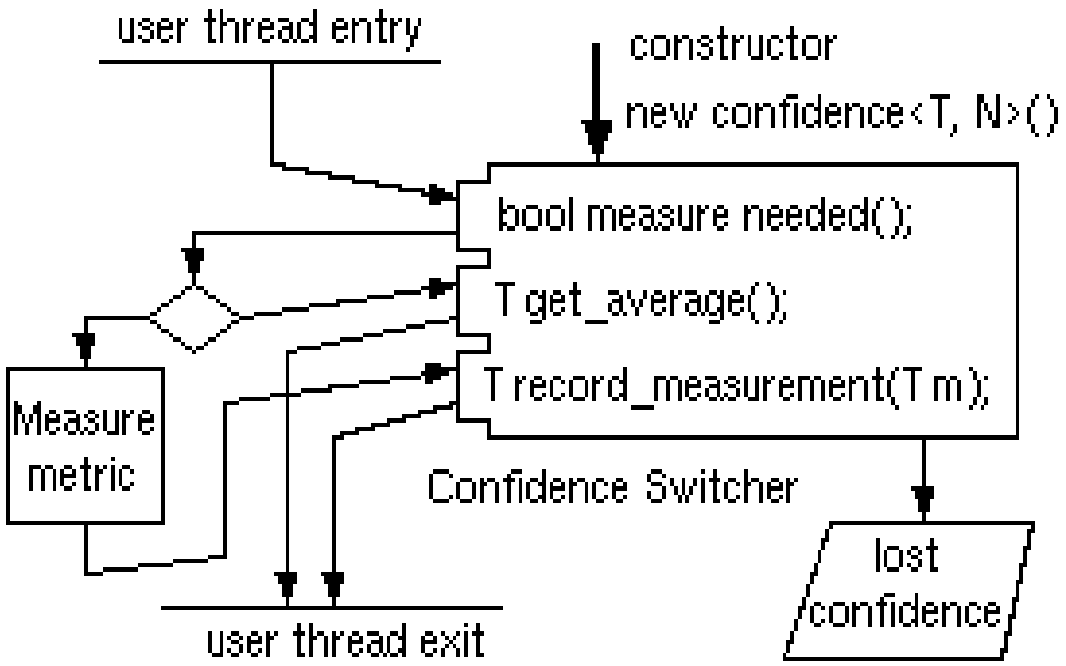
Three different settings of the global quantum.

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 ...

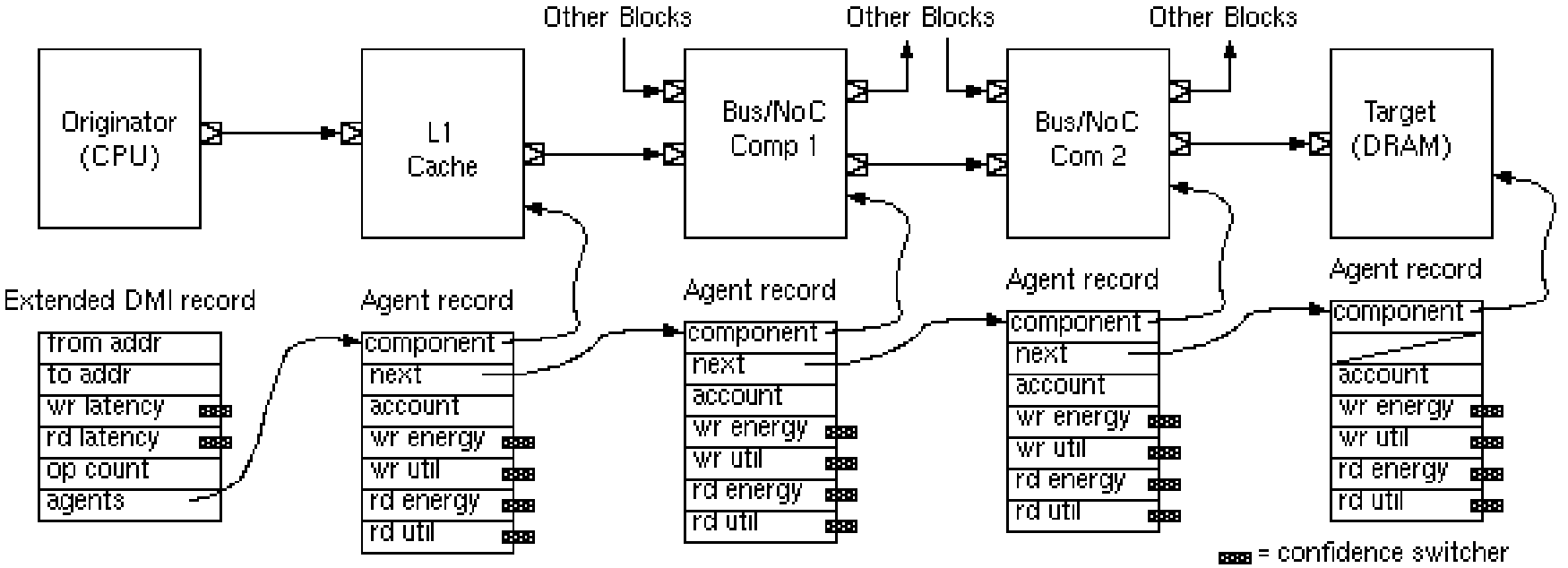
Very simple implementation if we just want an estimate of the average metric:

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:

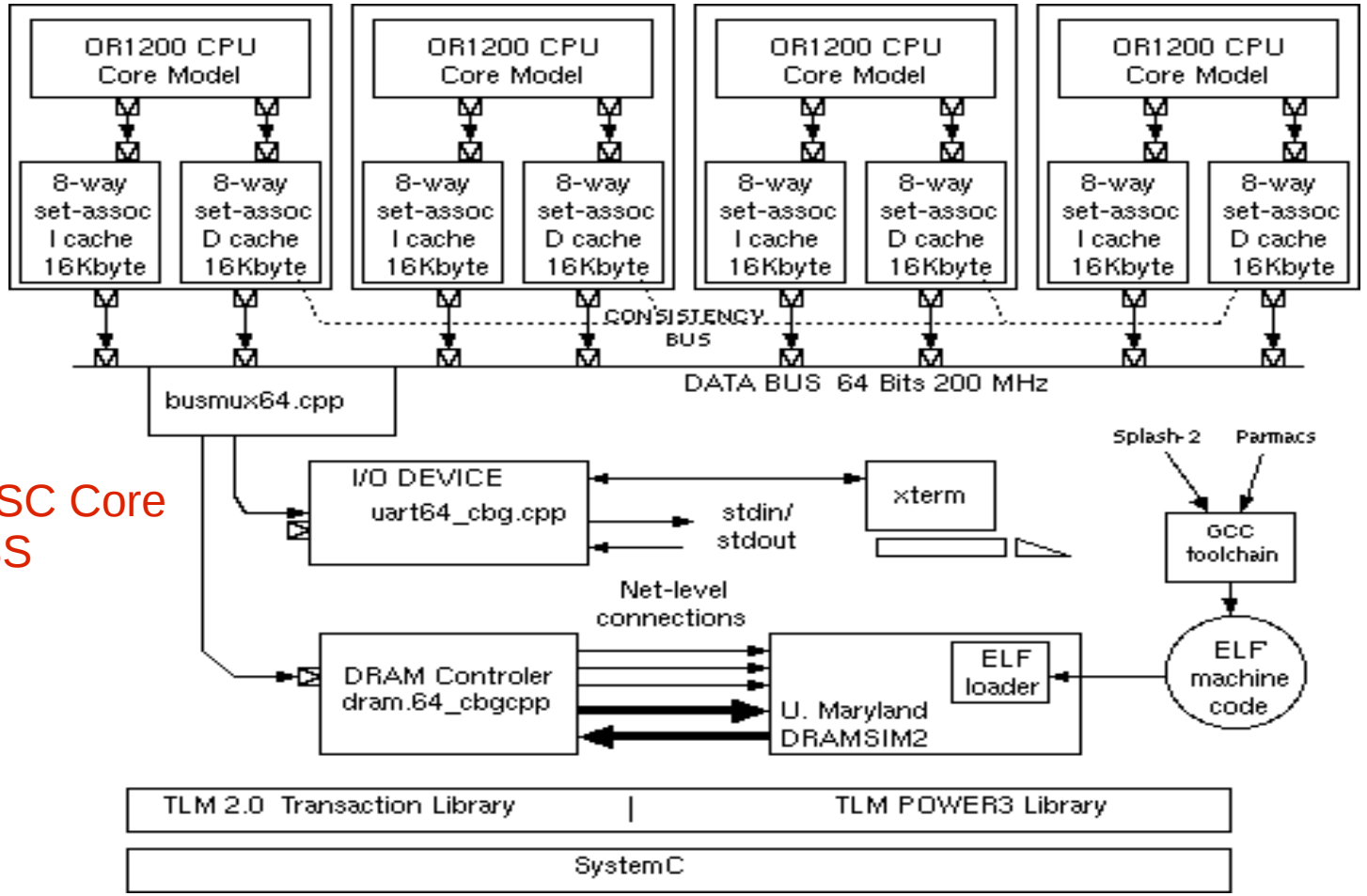
so DMI record at initiator is extended with either

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

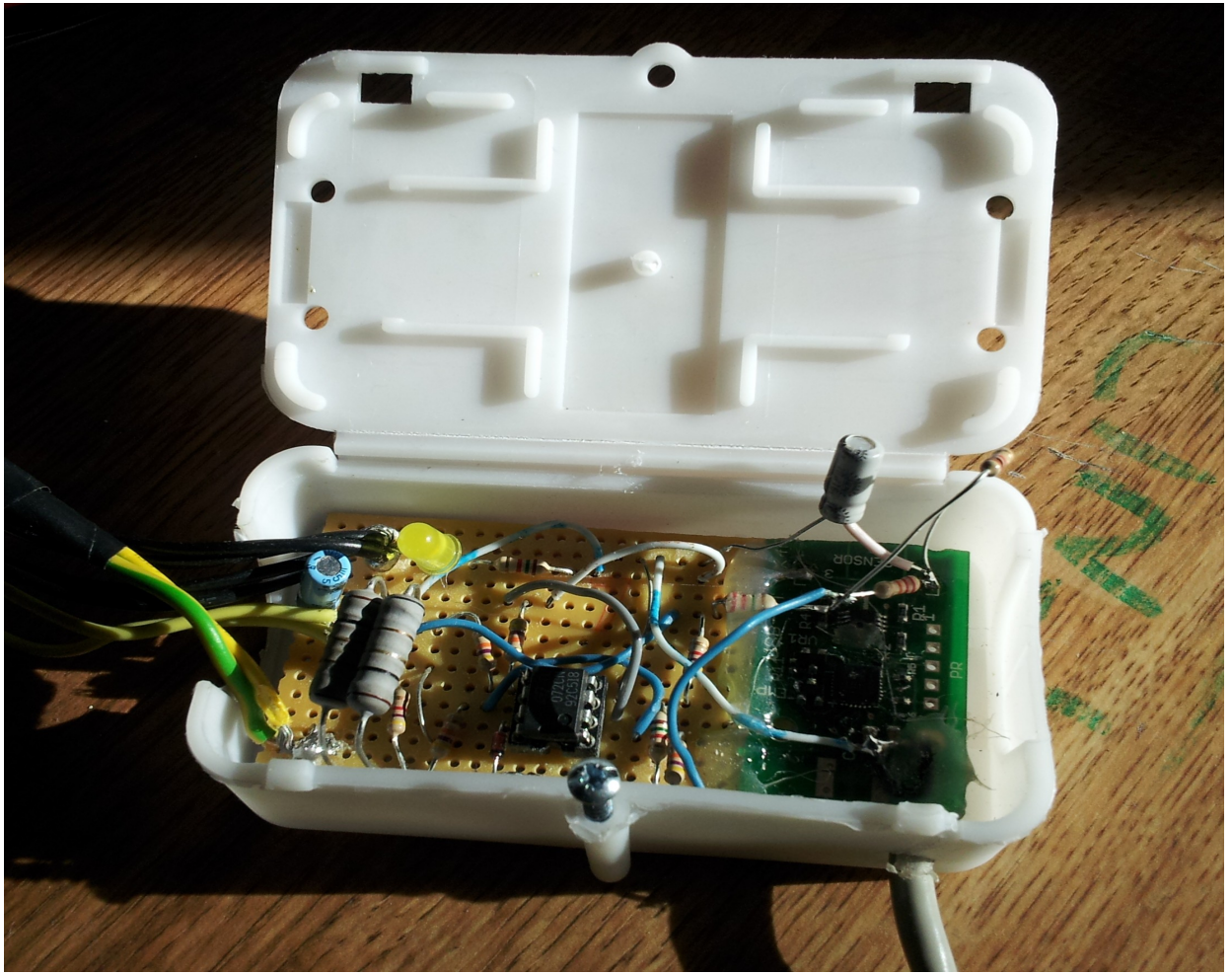


Verilated OpenRISC Core
Or fast ORSIM ISS
Or MIPS64 ...

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



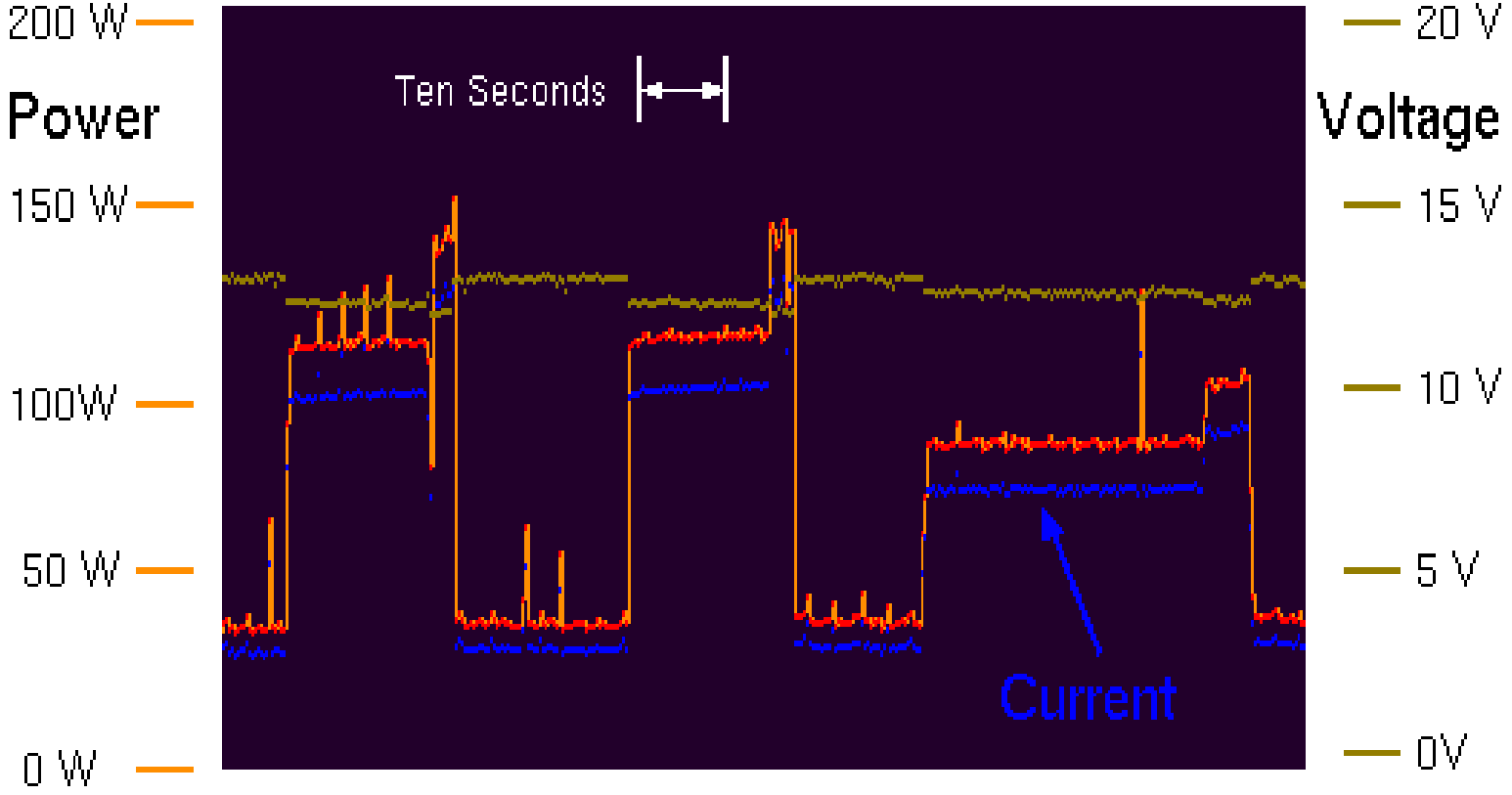
USB probe

Measures 12 volt rail to CPU socket.

Measures volts and amps at 10 Hz rate.

Accuracy:
consistency of about 1 percent between runs.

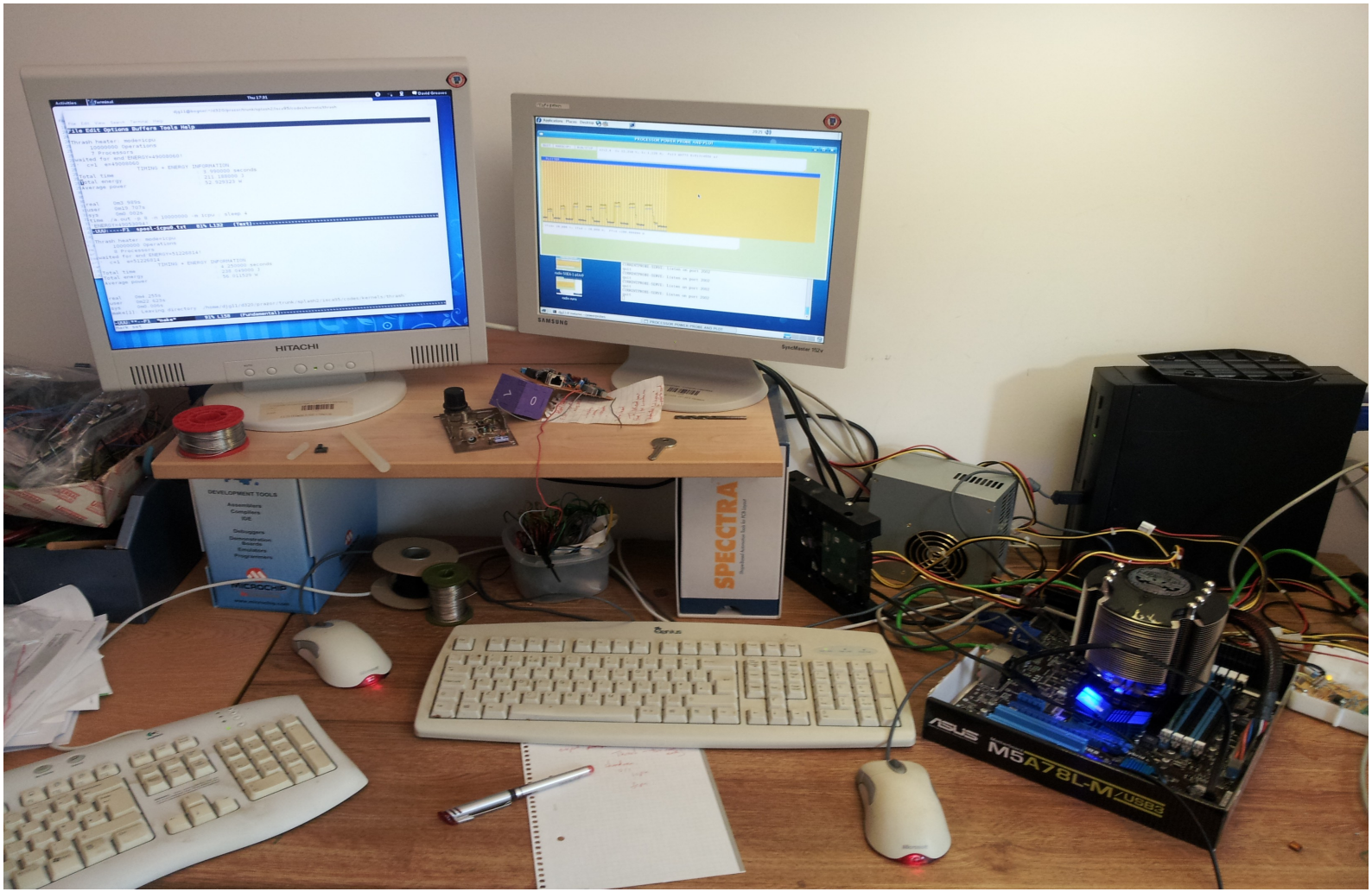
Splash-2 'RADIX' : First Test Setup



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...

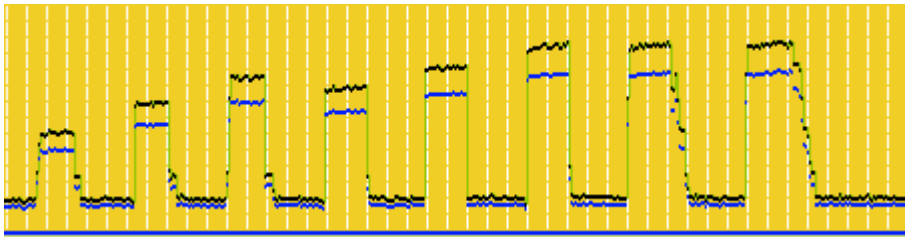
Probed and Probing Machines



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

Phenom Corner Cases: 1 to 8 threads

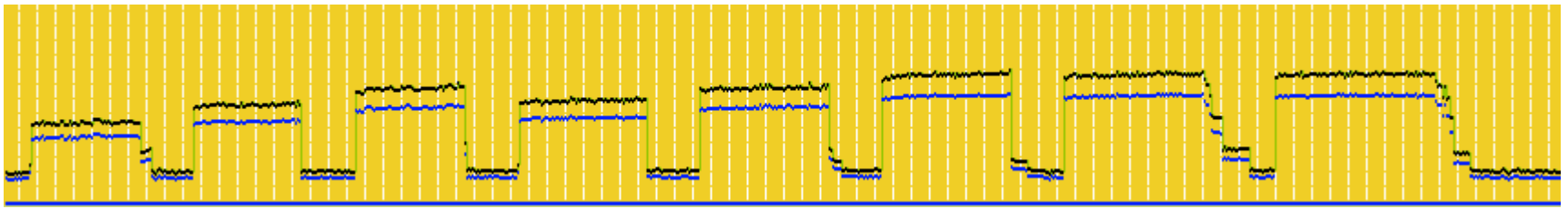
Integer ALU



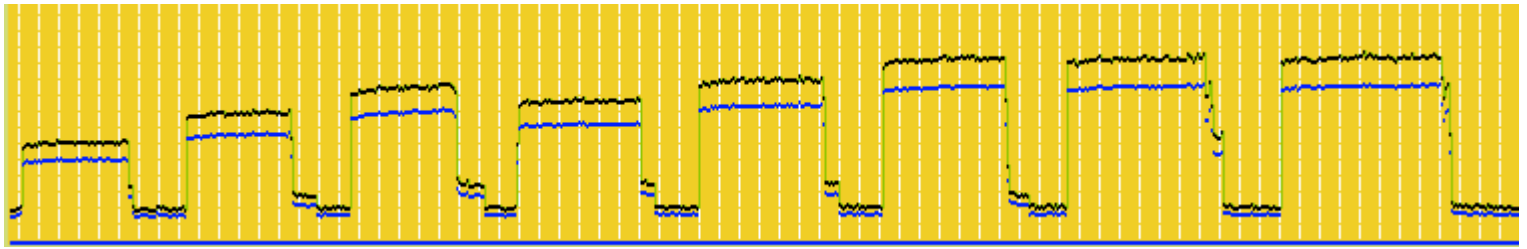
Vertical bar -> 1 second.
Horizontal scale -> 100 Watts.

System has 6 cores sharing one DRAM bank.

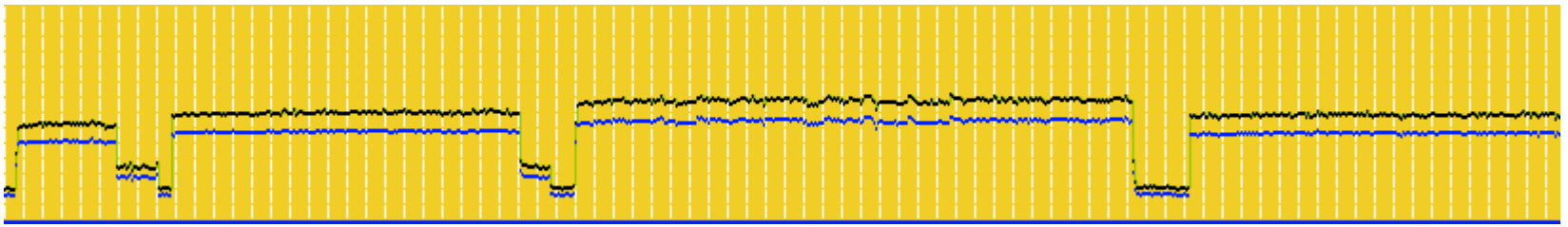
Floating Point ALU



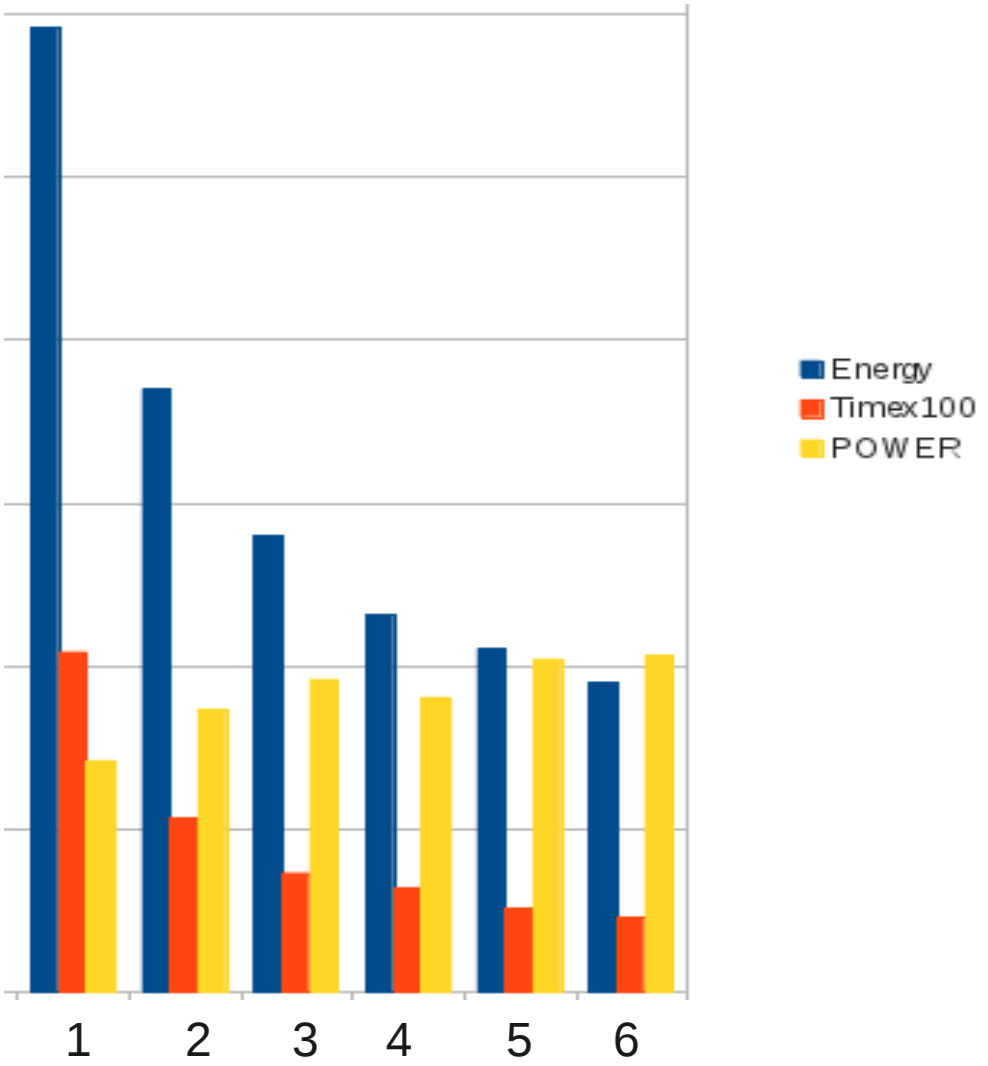
Memory Access: Disjoint Footprints



Memory Access: Overlapping Footprint



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 ---> increased performance.

Increasing n ---> better efficiency.

Strange power humps !

One DRAM DIMM shared.

Phenom Energy Coefficients

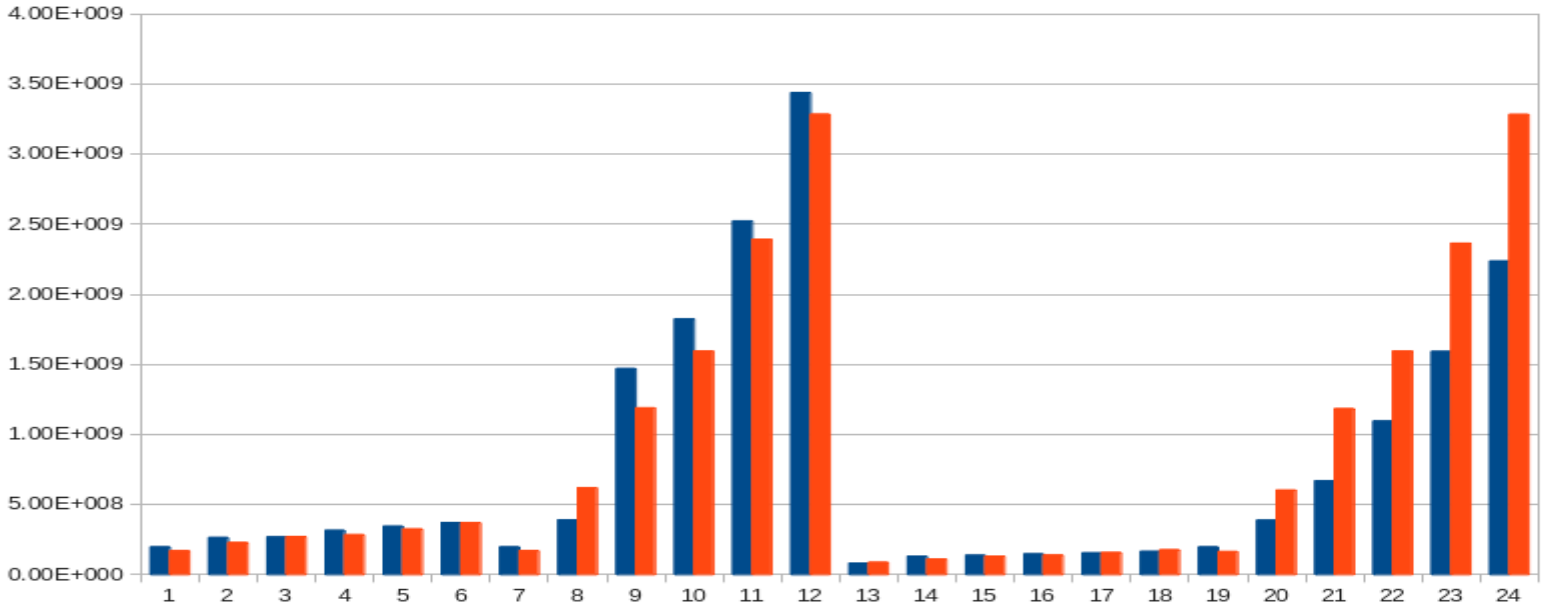
Instruction	1 nJ
I Cache Miss	50 nJ
D Cache Miss	15 uJ
D Cache Snoop Read	4 mJ
D Cache Evict	7 mJ

Values obtained from curve fitting

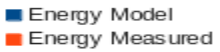
CPU + Caches only.

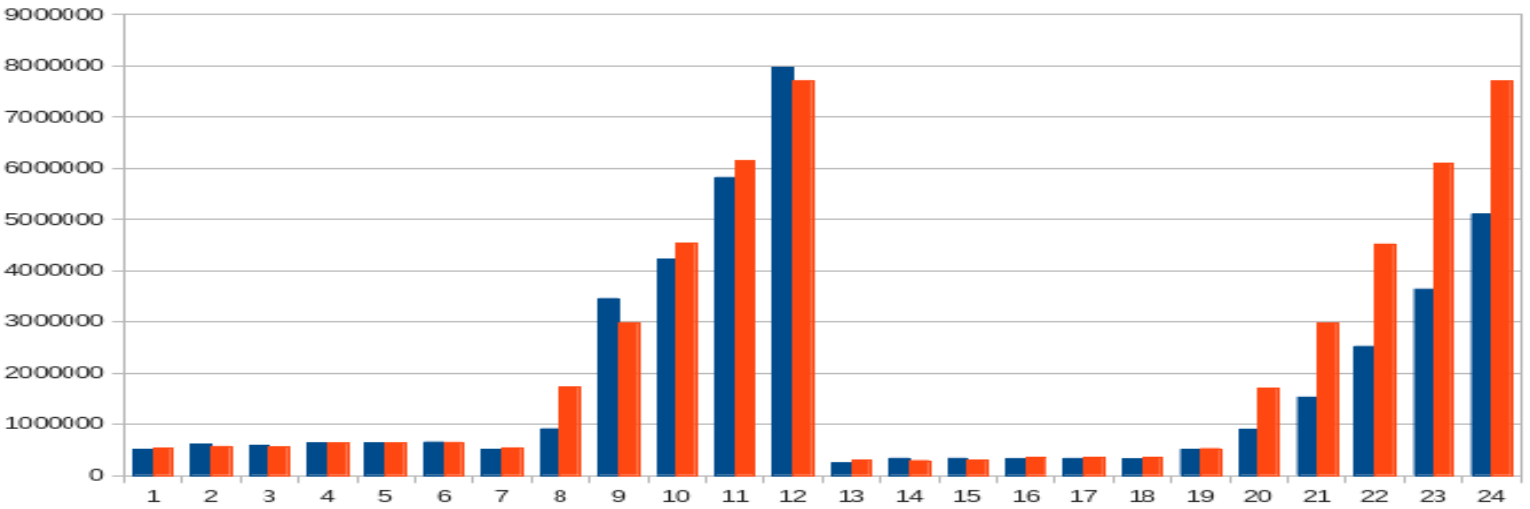
DRAM excluded.

Measured v Predicted: Runs 19-24 extrapolated from data fitting on 1-18.

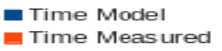


Energy





Time



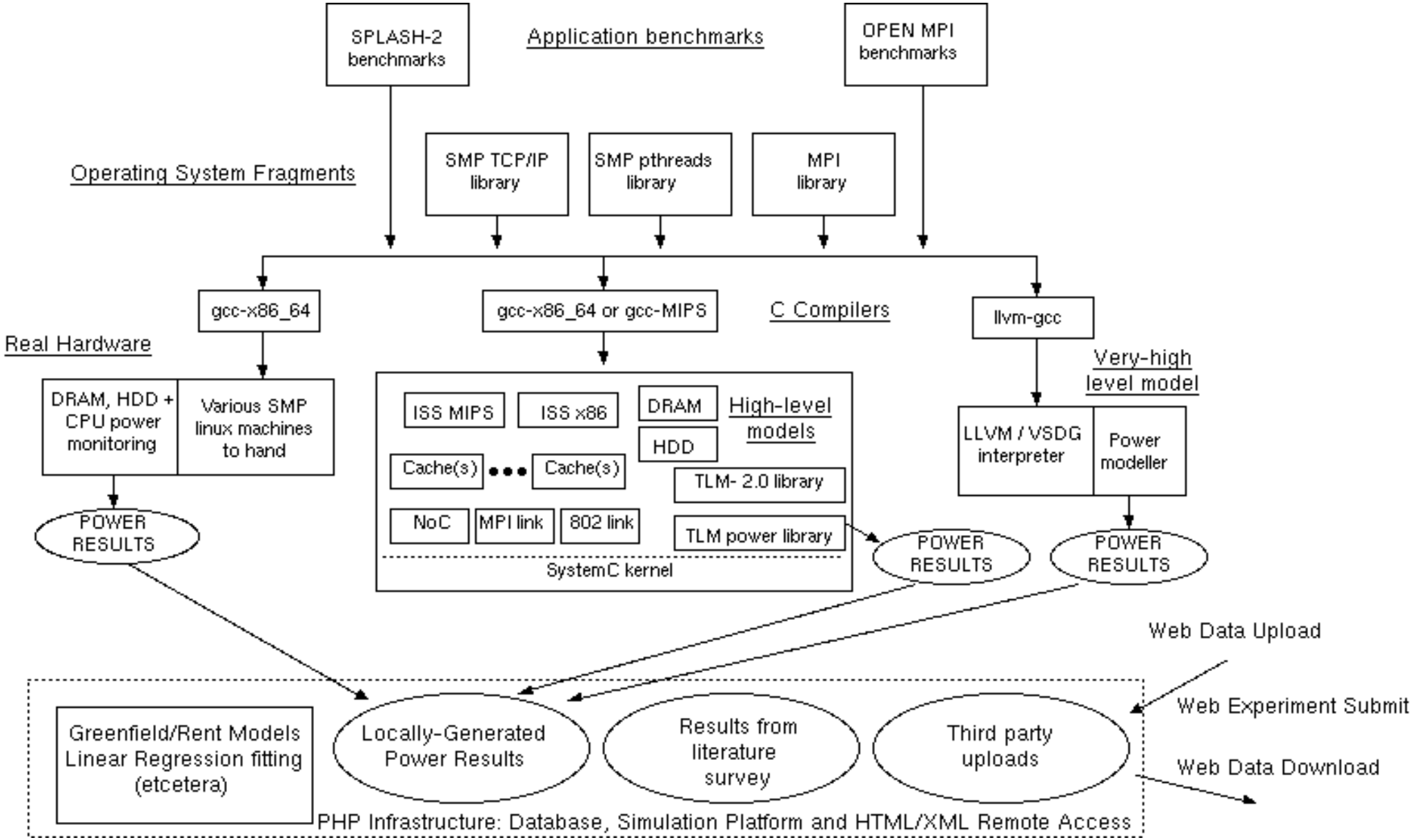
Thankyou for listening

David Greaves
M Yasin
(Milos Pusovic)

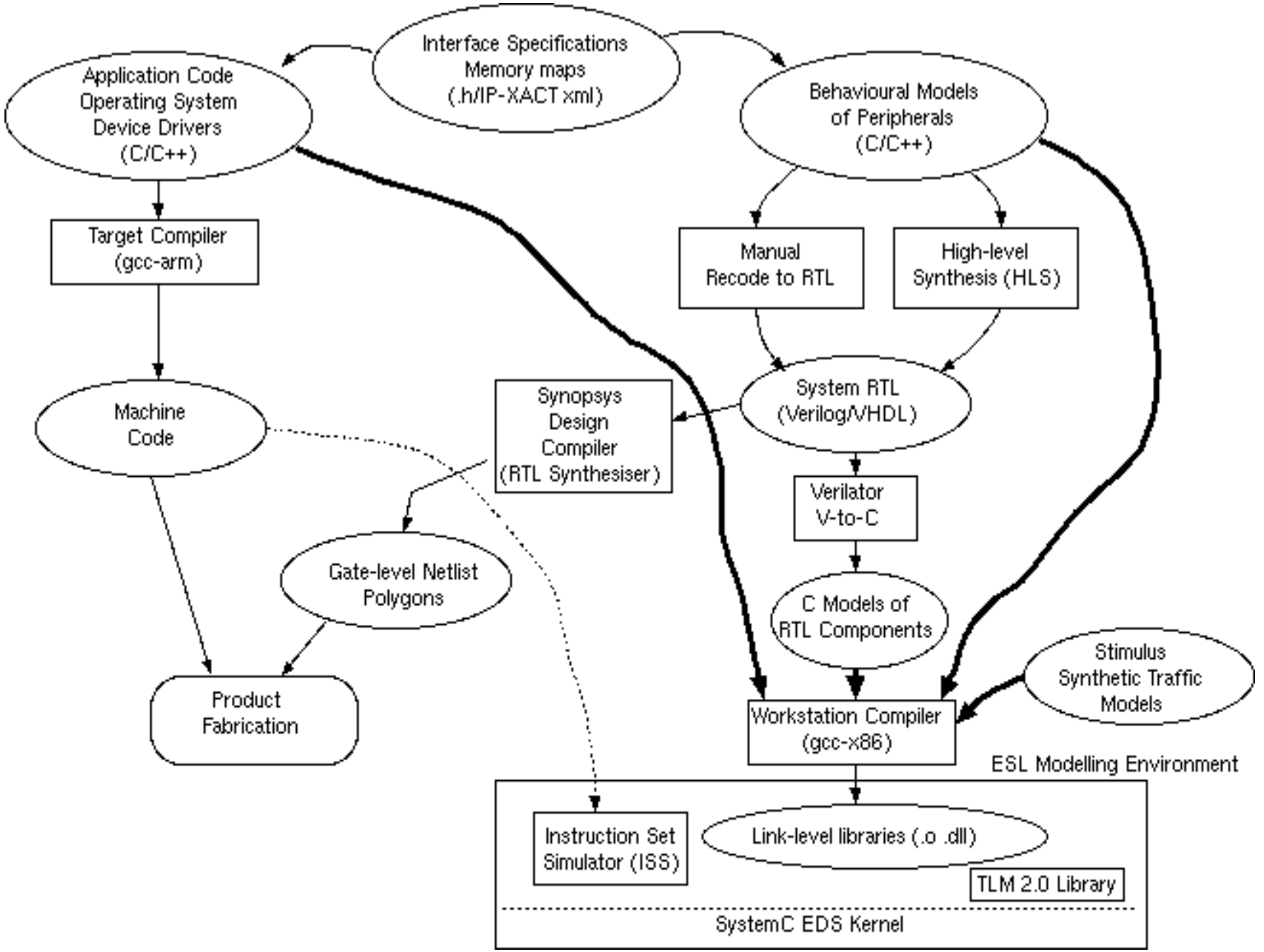
University of Cambridge
Computer Laboratory

FDL 2012 Vienna.

Power Estimation: Project Flow



Backup Slide: ESL Modelling Flow



Backup Slide: Loosely-timed TLM Modelling