Power Model
ESESC Tutorial

Speaker: Meeta Sinha

ESESC

Department of Computer Engineering,
University of California, Santa Cruz
http://masc.soe.ucsc.edu

UC SANTA CRUZ

Power Model

- You will learn:
  - High level view of ESESC power model
  - Run a power simulation
  - Use report.pl to view power numbers
  - LibPeq

References

CACTI

1. Naveen Muralimanohar, Rajeev Balasubramonian, and Norman P. Jouppi. CACTI 6.0: A tool to model large caches. *Technical Report, HP Laboratories*, 2009.
2. Sheng Li et al. CACTI-P: Architecture-level modeling for SRAM-based structures with advanced leakage reduction techniques. *IEEE/ACM International Conference on Computer-Aided Design*, pages 694–701, 2011.

McPAT

3. Sheng Li et al. McPAT: An integrated power, area, and timing modeling framework for multicore and manycore architectures. *IEEE/ACM International Symposium on Microarchitecture*, pages 469–480, 2009.

LibPeq

4. Elnaz Ebrahimi. Pareto-optimal methodology for cache and SRAM modeling. *MS thesis, University of California, Santa Cruz*, 2011.
5. Meeta Sinha. Equation-based power model integration in ESESC. MS thesis, University of California, Santa Cruz, 2013.

Outline

- Overview of the ESESC power model
  - High level description of the power model
  - Structure & components
  - Configuration
- Power Model Demo & Reported values
- LibPeq
- Code structure

Dynamic Power

- *Dynamic Energy =*

Event Counters ✖ **Energy associated with each event.**

- Read accesses
- Write accesses
- Miss buffer accesses
- Branch instructions
- ALU accesses

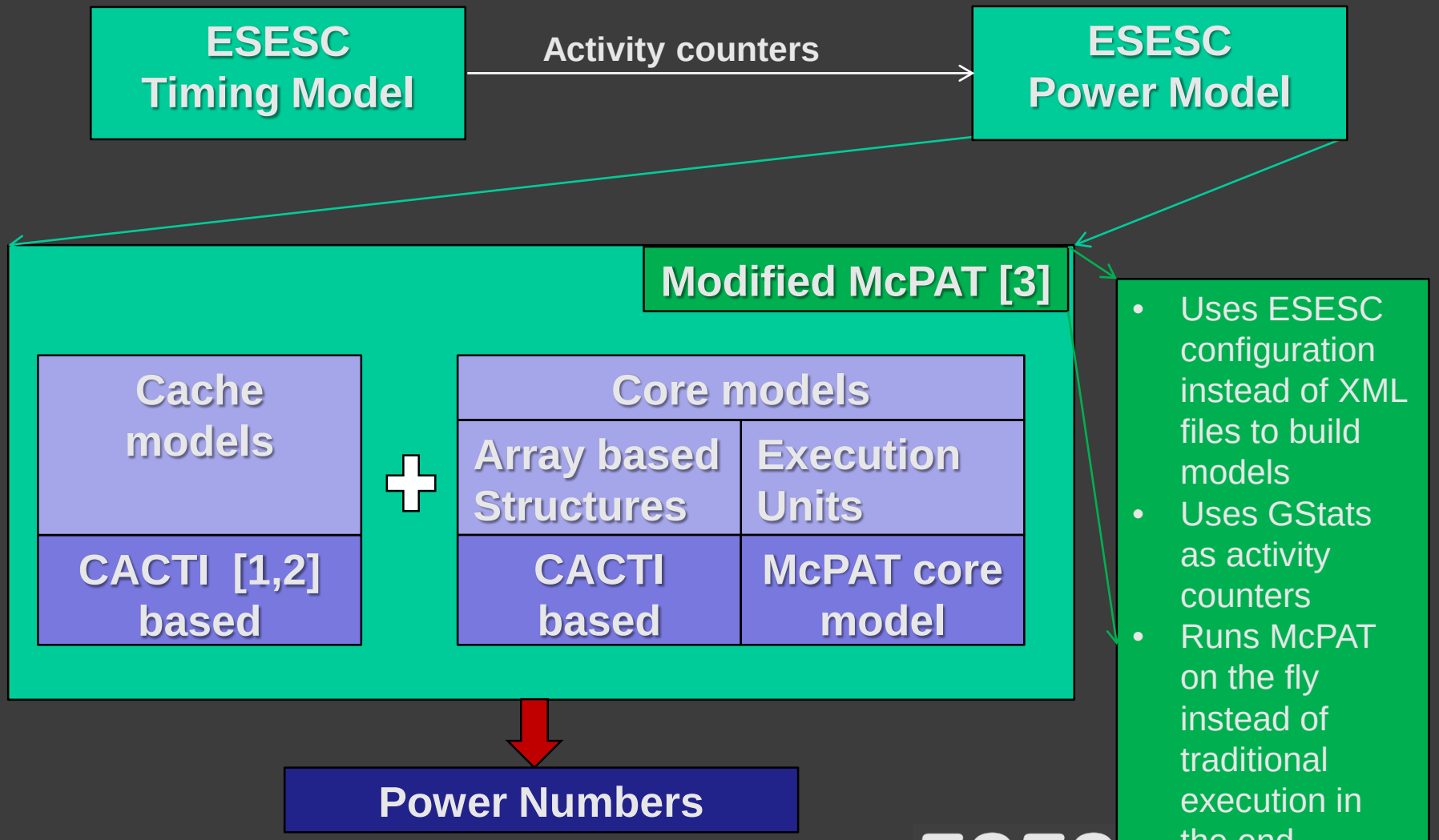
Etc.

- # memory ports
- Linesize
- Core tech node
- Cache Size
- Issue width

Etc.

MODEL

Power Model - Structure



Power Model - Components

Renaming Unit (RNU)

iFRAT
iRRAT
ifreeL
fFRAT
fRRAT
ffreeL

Fetch Unit

global BPT
L1_local BPT
chooser
RAS
BTB
IB

Icache

mmu-itlb
ifu-icache

dcache

mmu-dtlb
lsu-dcache
lsu-dcachecc

Load Store Unit (LSU)

lsu-LSQ
lsu-LoadQ
lsu-ssit
lsu-lfst

Execution Unit (EXE)

EXEU
fp_u
int_inst_window
fp_inst_window

Register File

IRF
FRF

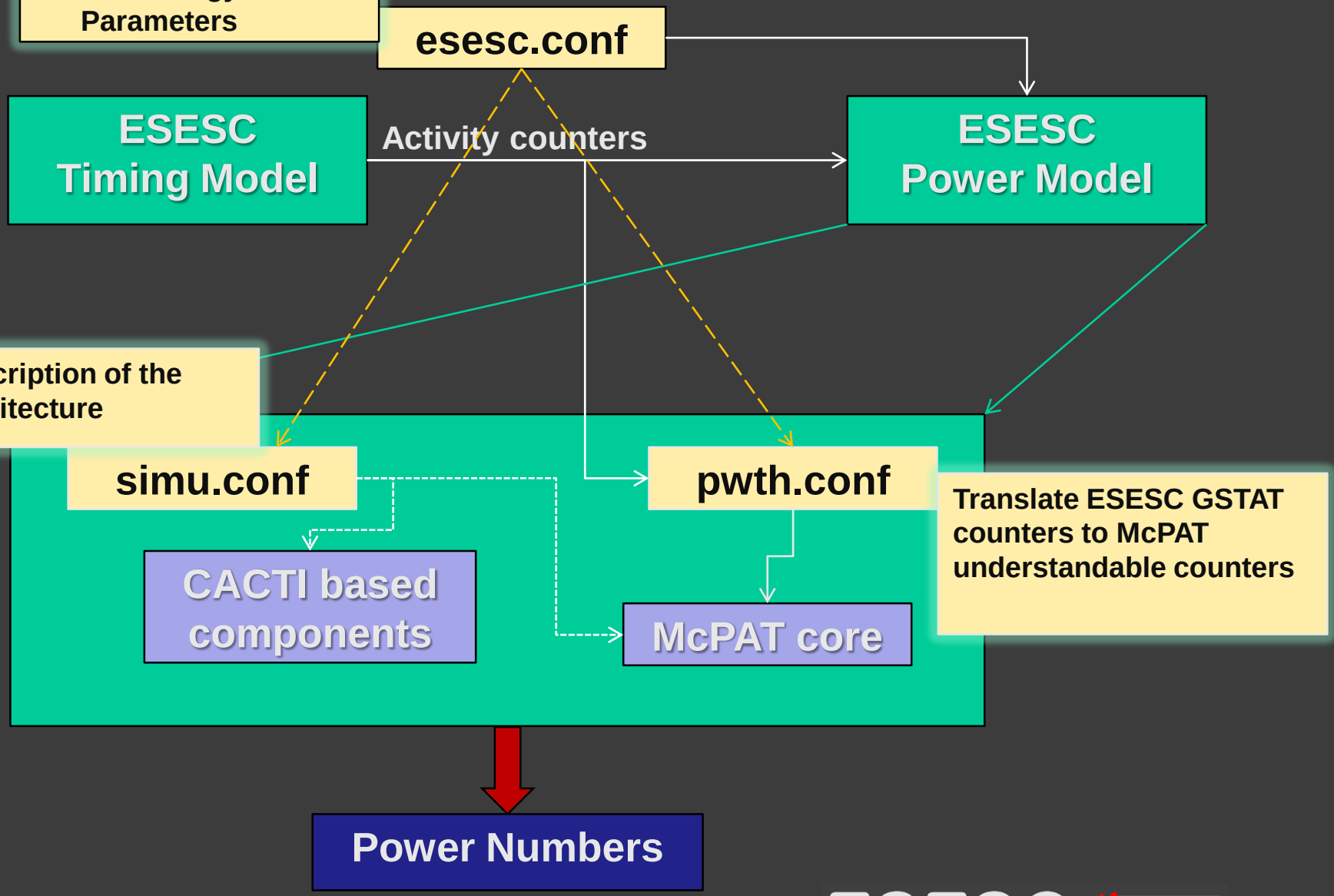
Reorder Buffer(ROB)

Array-based structures modeled by CACTI for McPAT

McPAT Core Model

Power Model - Config

- Enable or disable the power model
- Technology Parameters



Outline

- Overview of the ESESC power model
  - High level description of the power model.
  - Structure & components
  - Configuration
- Power Model Demo & Reported values
- LibPeq
- Code structure

Power Model - Demo

- Enable the power model and run a benchmark
- Use report.pl to view the power numbers

Reported numbers: report.pl

```
**********************************************************************************************************
# File  : esesc_iscademo.1WQqoZ : Fri Jun 13 09:39:48 2014
**********************************************************************************************************
Sampler 0 (Procs  0)
        Rabbit Warmup  Detail  Timing  Total  KIPS
  KIPS     N/A  11653     525     527    6054     0.0%   49.7%   12.7%   37.6% : Sim Time (s) 16.529 Exe   1.921 ms Sim (1700MHz)
  Inst    0.0%   95.6%    1.1%    3.3%          : Approx Total Time 64.854 ms Sim (1700MHz)
**********************************************************************************************************
Proc : Avg.Time :  BPType          :  Total  :         RAS         :  BPred  :         BTB         :  BTAC
   0 :   31.074 :  hybrid          :  89.69% : (100.00% of 8.18%) :  94.10% : ( 89.23% of 35.13%) :  0.00%
----------------------------------------------------------------------------------------------------------
Proc :  rawInst  :   nCommit   :   nInst   :  AALU   :  BALU   :  CALU   :  LALU   :  SALU   : LD Fwd :    Replay    : Worst Unit  (clk)
   0 :   3270620 :    6410740  :   6410881 :  62.54% :   5.86% :   0.11% :  16.79% :  14.70% :  0.00% :      N/A     : SUNIT_AALU  1.32
----------------------------------------------------------------------------------------------------------
Proc  IPC     uIPC    Active        Cycles    Busy   LDQ   STQ  IWin   ROB  Regs    IO  maxBr  MisBr Br4Clk brDelay
   0  01.00   1.96    0.90         3264934    49.1   2.1   2.1  15.1   1.5   0.0   1.8   0.0    8.9    0.0    1.4
**********************************************************************************************************
Cache          Occ  AvgMemLat MemAccesses  MissRate  (  RD ,    WR,    BUS)  Dyn_Pow (mW) Lkg_Pow (mW)
IL1(0)         0.0  3.2       2254695       0.93%    ( 99.0%,  0.0%,  0.0%)      250          1
ITLB(0)        0.0  3.3       2254376       0.02%    (100.0%,  0.0%,  0.0%)        0          0
----------------------------------------------------------------------------------------------------------
DL1(0)         0.0  9.8       2510395       0.58%    ( 98.9%, 98.3%,  0.0%)     1322         51
----------------------------------------------------------------------------------------------------------
L2(0)          0.0  51.6      36128        16.58%    ( 85.7%, 26.9%,  0.0%)        0         77
L3(0)          0.0  166.0     5998         97.22%    (  3.2%,  0.6%,  0.0%)        0        355
MemBus(0)      0.0  141.9     5833          0.00%    (100.0%,  0.0%,  0.0%)        0         36
PTLB(0)        0.0  9.9       2509957       2.39%    ( 97.6%,  0.0%,  0.0%)        0          0
**********************************************************************************************************
CPU Power Metrics: (Dynamic Power,Leakage Power)
Proc |  RF (mW)    |  ROB (mW)   |  fetch (mW) |  EXE (mW)   |  RNU (mW)   |  LSU (mW)   |  Total (W)
----------------------------------------------------------------------------------------------------------
 0   |  115  , 0   |  115  , 0   |  78   , 1   |  185  , 0   |  173  , 78  |  210  , 0   |  2.45 , 0.60
**********************************************************************************************************
```

Power number for the caches

Power number per architectural block

Outline

- Overview of the ESESC power model
  - High level description of the power model.
  - Structure & components
  - Configuration
- Power Model Demo & Reported values
- Power Model options
- LibPeq
- Code structure

Power Model – LibPeq (alpha)

- What is LibPeq[4,5]?
  - Simplified analytical power model derived using statistical techniques on CACTI results.
- Why LibPeq?
  - CACTI initialization time very high.
  - CACTI covers a subset of design search space. Full exploration extremely time consuming.
  - Increasing complexity of microprocessors lead to exponentially increasing design search space.

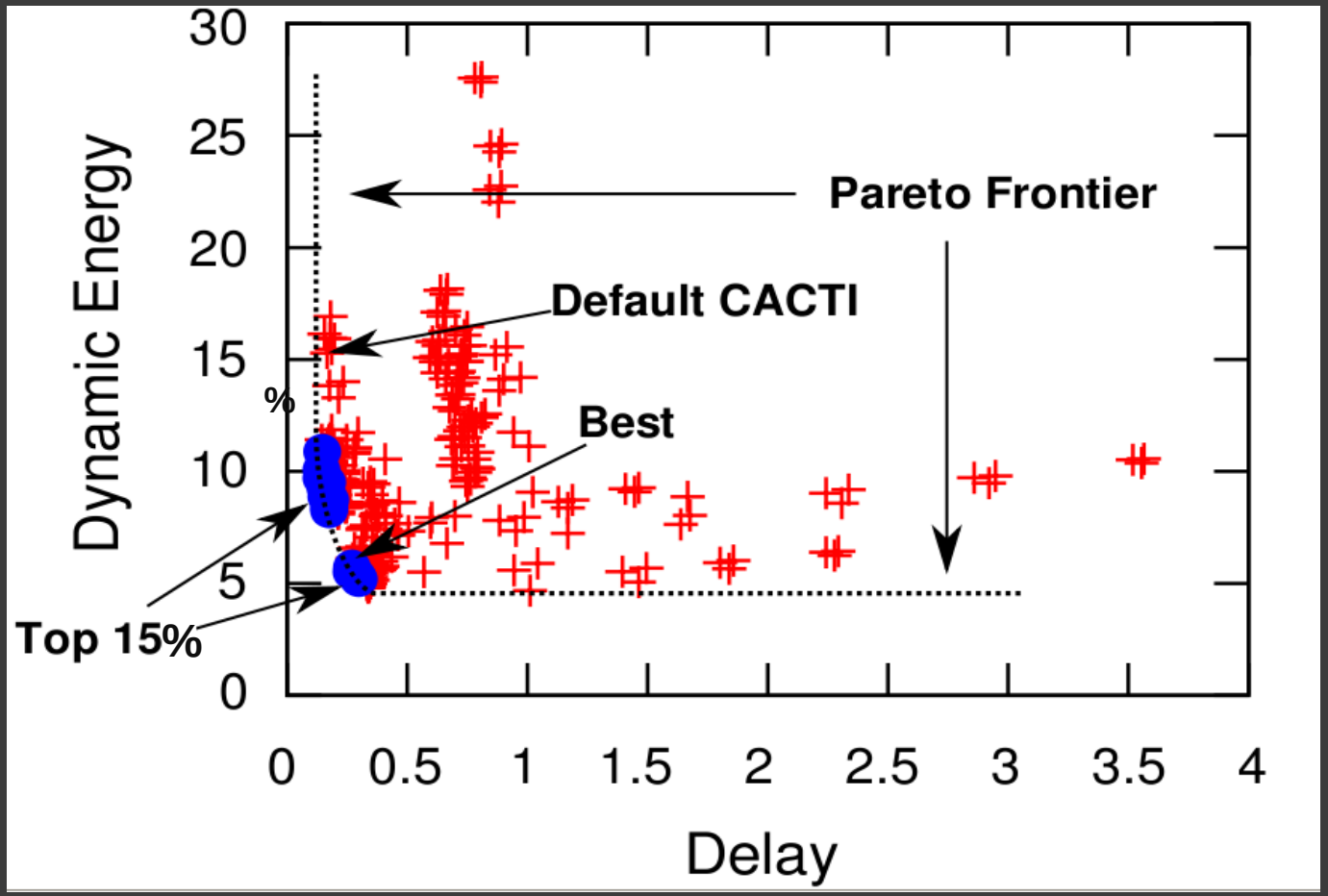
LibPeq: Modeling Technique

- LibPeq: Pareto efficient power model.

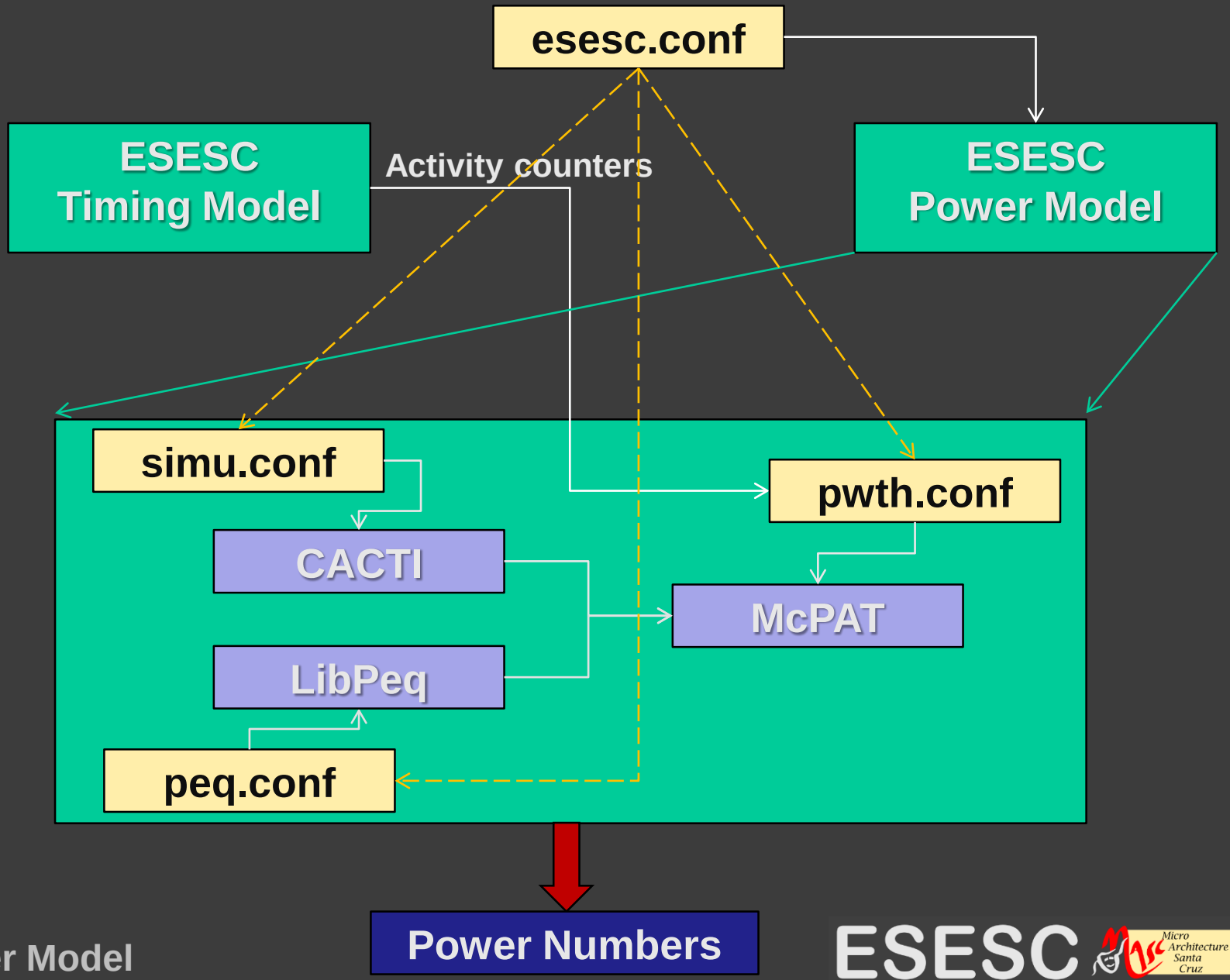
- Keywords:
  - **Pareto Efficiency :** Optimal approach for analyzing tradeoff between parameters.
  - **Pareto Frontier:** Set of choices that are pareto efficient.

LibPeq: Modeling Technique

- Plot: Energy and Delay value from CACTI runs for a particular array based structure.



LibPeq - Structure



LibPeq Speedup

- Running crafty with the default parameters **without** LibPeq

```
time esesc

real    1m0.626s
user    1m25.870s
sys     0m26.930s
```

- Running crafty with the default parameters **with** LibPeq

```
time esesc

real    0m37.680s
user    0m43.160s
sys     0m25.720s
```

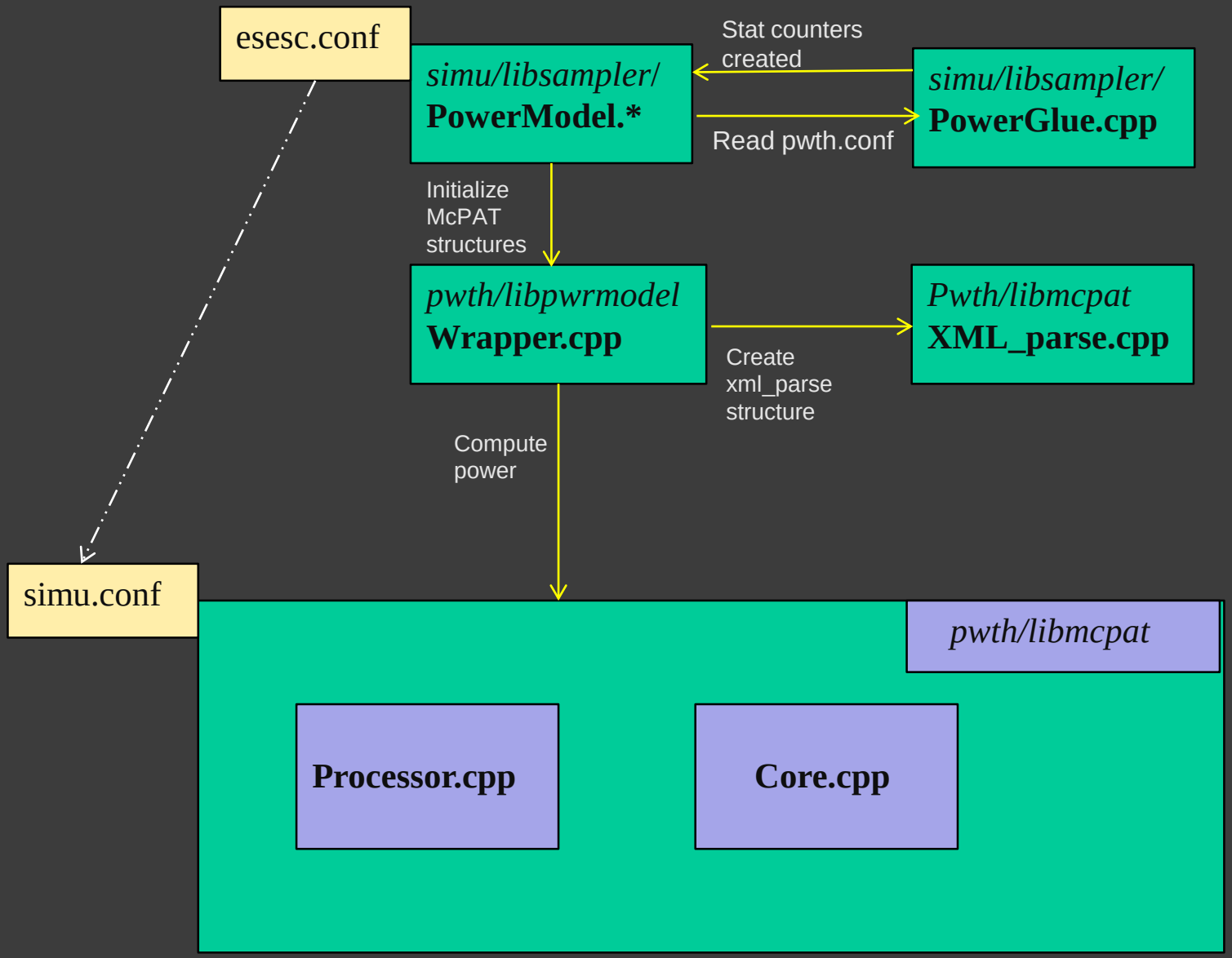
Outline

- Overview of the ESESC power model
  - High level description of the power model.
  - Structure & components
  - Configuration
- Power Model Demo & Reported value
- Power Model options
- LibPeq
- Code structure

Code Structure: Main files

Files / Directories	Significance
simu/libsampler/Powermodel.*	Main hook to the ESESC power model
pwth/libmcpat	McPAT source code (modified to support ESESC)
pwth/libmcpat/XML_parse.cpp	Reading pwth.conf, translating ESESC GSTATS to McPAT counters
pwth/libmcpat/processor.cpp	McPAT models for various blocks in a processor
pwth/libmcpat/core.cpp	McPAT models for core components.
pwth/libpwrmodel	Wrapper for the power model
pwth/libpeq	Library to parse LibPeq equations.

Code Structure: Flow Chart



Code Structure: Pwth.conf

not showing all templates

```
[OoOPwrCounterTemplate]
template[0] = "stats[%d] = %s_core_total_instructions +%s:nCommitted +testCounter"

[inorderPwrCounterTemplate]
template[0] = "stats[%d] = %s_core_total_instructions +%s:nCommitted +testCounter"

[MemPwrCounterTemplate]
template[0] = "stats[%d] = %s_read_accesses +%s:readHit +%s:readMiss +%s:readHalfMiss +testCounter"

[MCPwrCounterTemplate]
template[0] = "stats[%d] = mc_read_accesses +L3:readMiss +testCounter"
template[1] = "stats[%d] = mc_write_accesses +L3:writeBack +L3:writeMiss +testCounter"
```

Component Category e.g IL1,DL1,L2,L3

McPAT counter

Gstat counters

Delimiter

- McPAT to Gstat translation for membus write access [MCPwrCounterTemplate]:
  - `mc_write_accesses = L3:writeBack +L3:writeMiss`
  - Component Category mapping in *simu/libsampler/PowerGlue.cpp*

Summary

Important Input Files

- esesc.conf : Enable/Disable Power Model
- simu.conf : Description of architecture
- pwth.conf : Translate GStat counters to McPAT understandable counters. (Don’t modify unless you know what you are doing)
- peq.conf: Contains equation for SRAM and caches. CAM equation and leakage equation can be added in future.

Questions?

(Backup slide)Power Model – LibPeq

- LibPeq
  - **Alpha** version
  - Analytical model
  - Developed from statistical analysis of thousands of CACTI simulations covering exhaustive design search space
  - Significantly faster
- Only models SRAM and cache structures
- Does not model leakage

(Backup slide)peq.conf

- Equations defined for array based structures

```
[SRAM_Small1]
dynamic ="exp(-4.982+2.196* ln(tech)+0.4961* ln(ports)-0.00986*
sqrt(size)+0.5464* ln(size)-0.016961* width+0.4027* sqrt(width))* (10^(-9))"


[SRAM_Large1]
dynamic = "exp(-5.446+2.094* ln(tech)+0.886* ln(ports)+0.000458*
sqrt(size)+0.5296* ln(size)-0.011965* width+0.31001* sqrt(width))* (10^(-9))"
```

- Included by esesc.conf
- Should not be modified (unless you know what you are doing!)

(backup)Code Structure: Core.cpp

- Important Functions:
  # not all components shown

```
Core::Core(ParseXML* XML_interface, int ithCore_, InputParameter* interface_ip_)
 :XML(XML_interface),
  ithCore(ithCore_),
  interface_ip(*interface_ip_),
  ifu  (0),
  lsu  (0),
  mmu  (0),
  exu  (0),
```

Sub-components

```
void Core::computeEnergy(bool is_tdp)
```

```
void Core::displayEnergy(uint32_t indent,int plevel,bool is_tdp)
```

(backup)Code Structure: Processor.cpp

```
void Processor::dumpStatics(ChipEnergyBundle *eBundle) {

  sprintf(name, "P(%d)_RNU",ii);

 tdp = (cores[i]->rnu->iFRAT->power.readOp.dynamic  +
     cores[i]->rnu->iRRAT->power.readOp.dynamic  +
     cores[i]->rnu->ifreeL->power.readOp.dynamic +
     cores[i]->rnu->fFRAT->power.readOp.dynamic  +
     cores[i]->rnu->fRRAT->power.readOp.dynamic  +
     cores[i]->rnu->ffreeL->power.readOp.dynamic);
```

Power for RNU sub -components