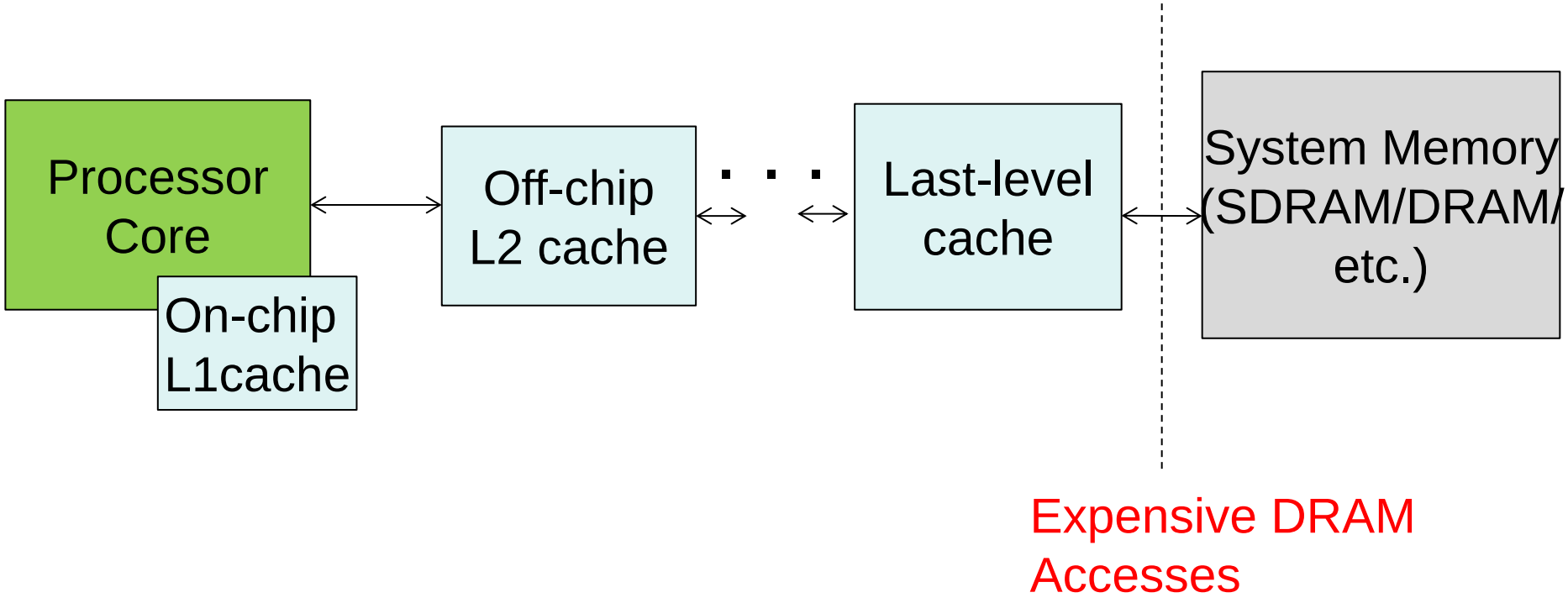
RExCache: Rapid Exploration of Unified Last-level Cache

Su Myat Min, Haris Javaid, Sri Parameswaran
University of New South Wales
25th Jan 2013
ASP-DAC 2013

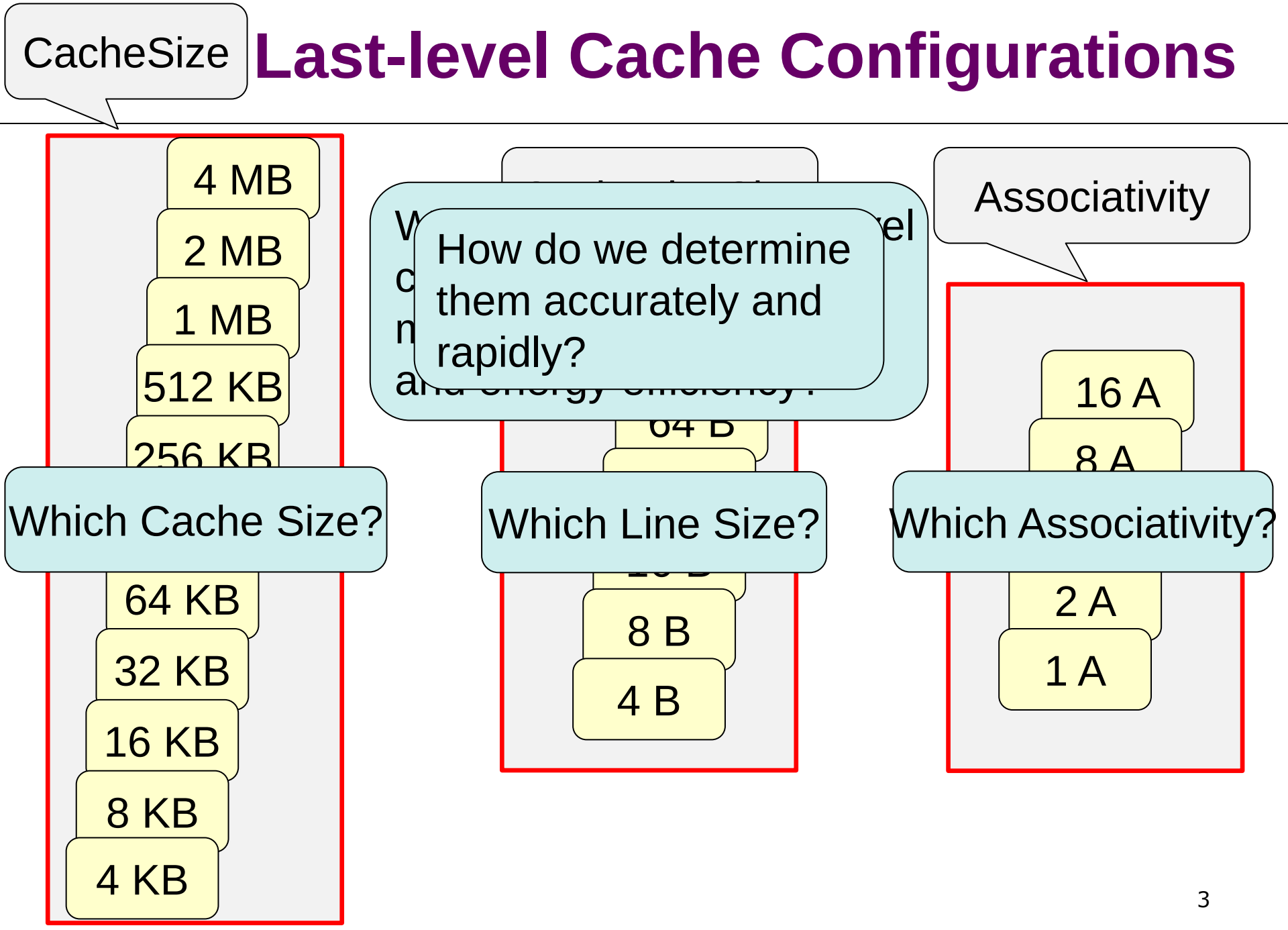
Introduction

- Last-level cache plays an important role in avoiding expensive memory accesses to the off-chip memory system

  [Arkaprava2007 at Micro,Wang2009 at IEEE NAS]

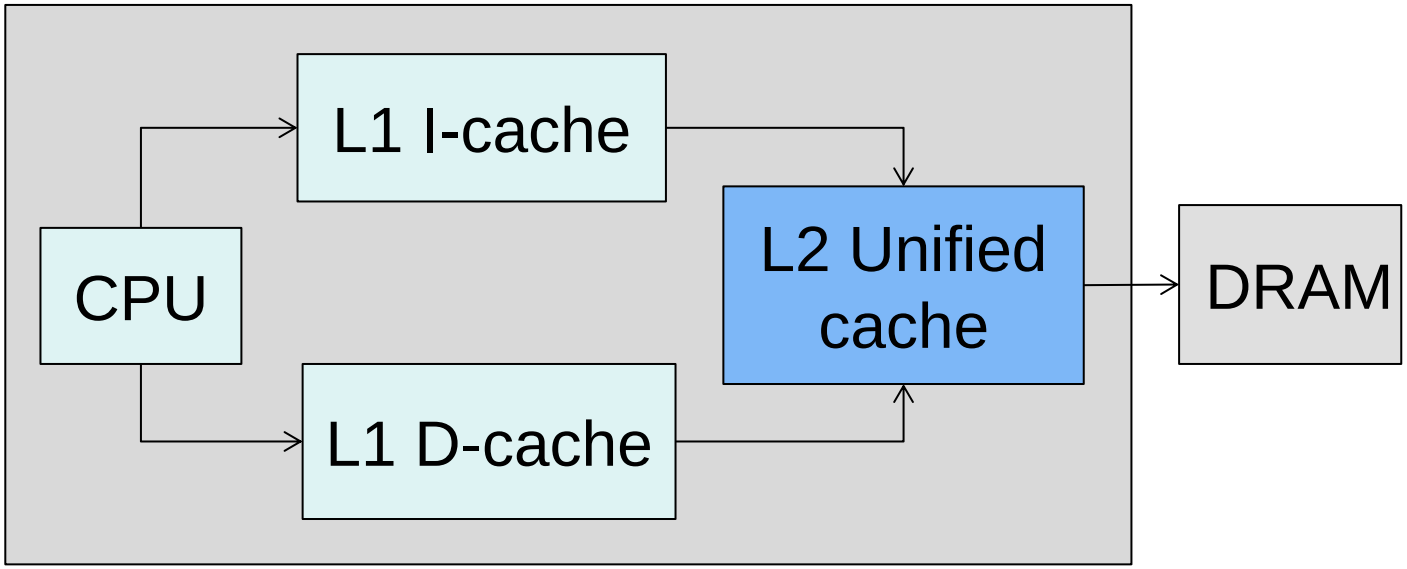


Last-level Cache Configurations



Target System

- Uniprocessor system with multi-level cache hierarchy
- Unified off-chip last-level cache
- Assumption: On-chip caches are already preconfigured

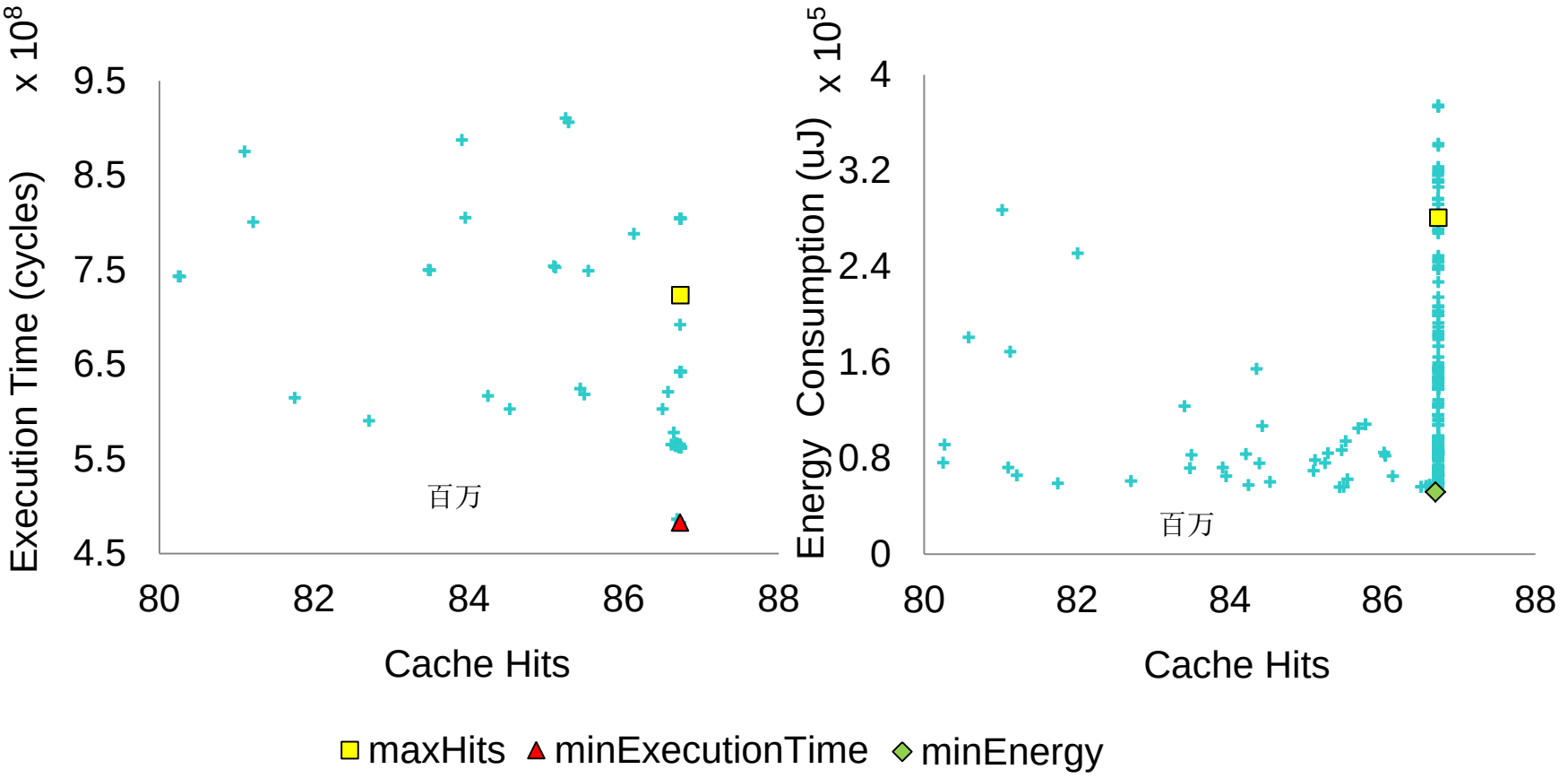


Outline

- Motivation
- Related Work
- Proposed Approach
  - RExCache Framework
  - Execution time and Energy estimators
- Experiment Results
- Conclusion

Motivation

- g721enc application running on uniprocessor system for different last-level cache (L2 cache) configurations



Related Work

- Cycle accurate simulators [PTLSim, Xtensa ISS, etc.]
  - Very slow to explore hundreds of cache configurations
- Trace-driven cache simulation [Sim-Cache, CacheSim, Dinero, SuseSim, etc.,]
  - Mostly provide hit/miss information
  - cannot capture whole application behavior
- Analytical models by simultaneous execution of on-chip L1 & L2 caches [Gordon-Ross et. al. ,2011 and Silva-Filho et. al. , 2011]
  - Quickly predicts cache performance but obtain inaccurate output
- Single stack-based technique for two-level caches [Zhang et. al.,2011]
  - The technique focus only on two-level cache hierarchy
- **Our approach:**
  Rapid exploration of unified last-level cache

Proposed Approach

- **RExCache Framework**

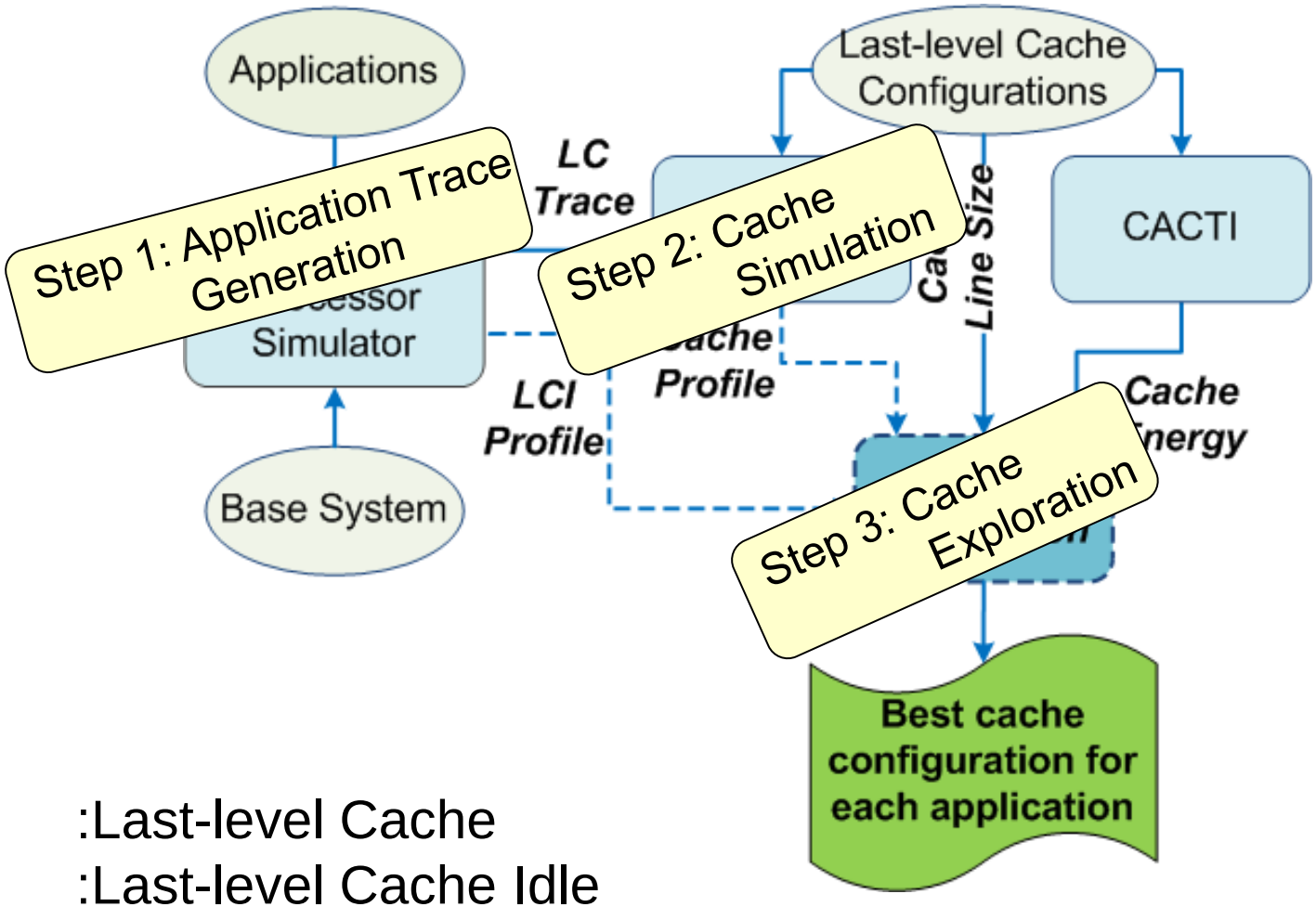
  (Framework for Rapid Exploration of Unified Last-level Cache)

  - integrates cycle accurate simulator and trace-driven cache simulator

- **Execution Time Estimator**

- **Energy Estimator**

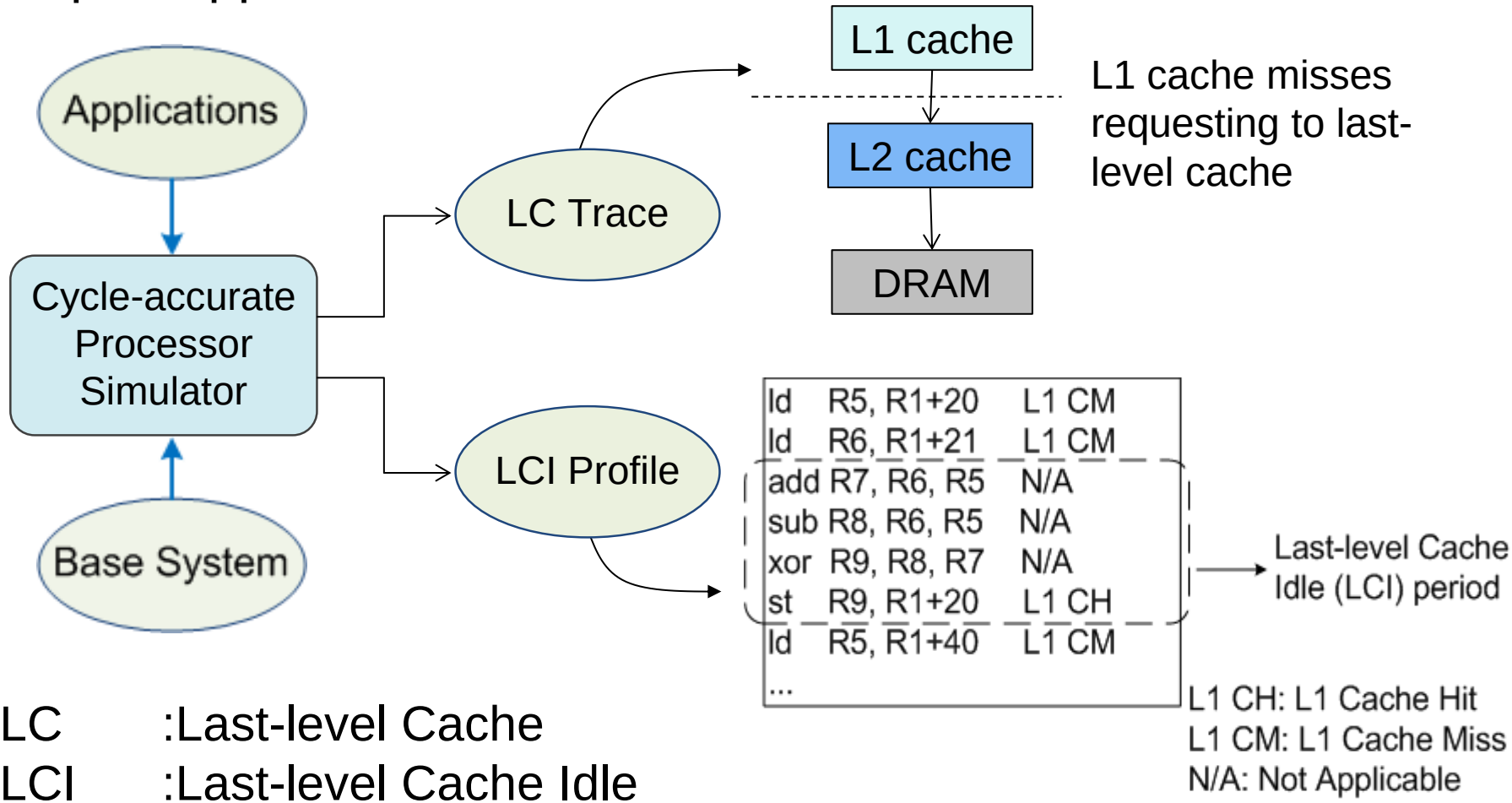
RExCache Framework



LC :Last-level Cache
LCI :Last-level Cache Idle
Base System has on-chip caches and largest last-level cache

RExCache Framework, Cont'd

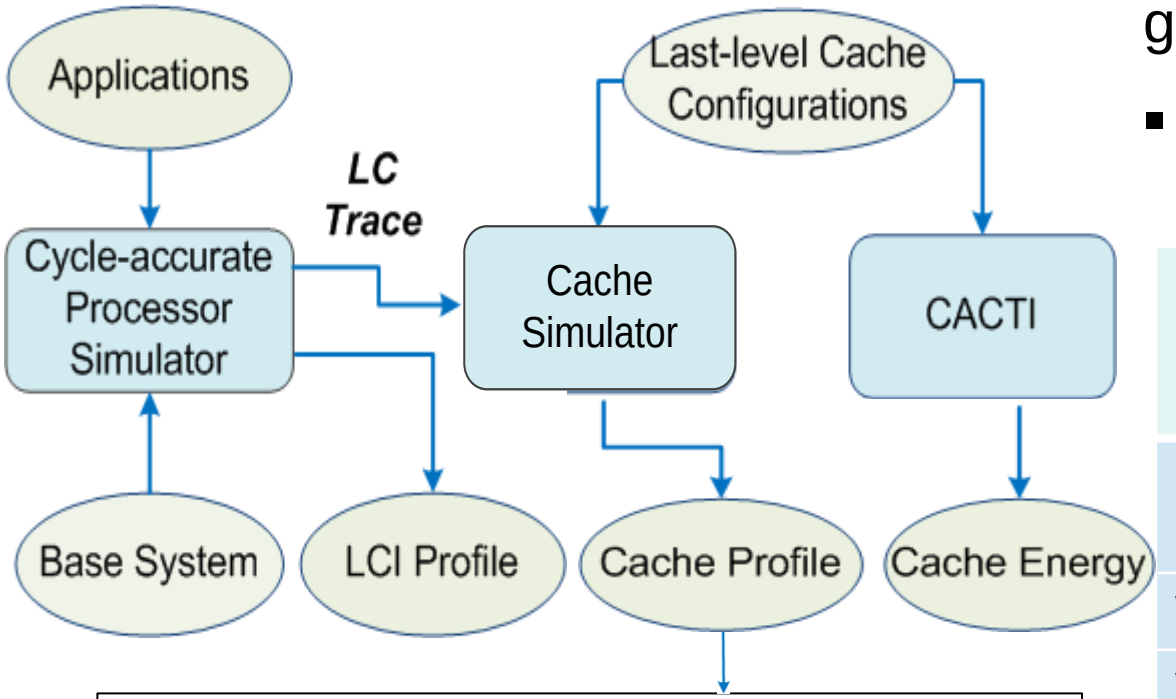
Step 1. Application Trace Generation



LC :Last-level Cache
LCI :Last-level Cache Idle
Base System has on-chip caches and largest last-level cache

RExCache Framework, Cont'd

Step 2. Cache Simulation



0x400000000	R	Mem Access
0x400000004	W	Mem Access
0x400000008	W	Mem Access
0x40000000c	R	Mem Access
0x400000004	W	Mem Non-Access
0x400000008	W	Mem Non-Access

Factors considered in generation of cache profile

- Cache Policy & Cache Hit/Miss Status

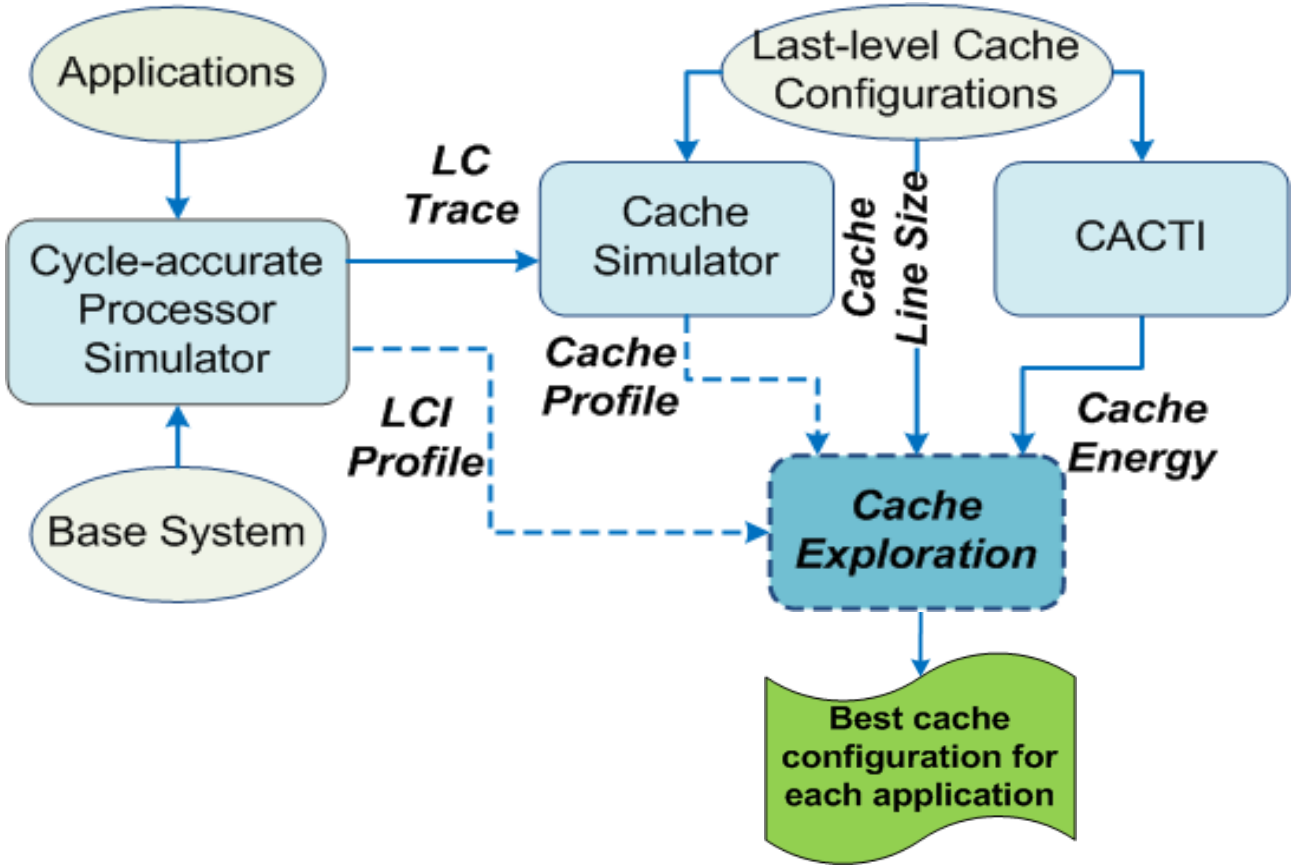
	No. of Memory Access for write-back policy
Read miss	If Dirty, 2 Else, 1
Write hit	0
Write miss-allocate	If Dirty, 1 Else, 0
Write miss no- allocate	1

RExCache Framework, cont'd

Step 3. Cache Exploration ⇨

Step 1: Generate Application Execution (AE) profile

Step 2: Execution Time & Energy Estimator



RExCache Framework, Cont'd

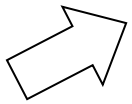
- Application Execution (AE) Profile of Cache Exploration

Cache Profile

MR1	Mem Non-Access
MR2	Mem Non-Access
MR3	Mem Non-Access
MR4	Mem Non-Access
MR5	Mem Access
MR6	Mem Access
MR7	Mem Non-Access
MR8	Mem Non-Access
MR9	Mem Access
MR10	Mem Access
...	

MR1
LCI period (20 cycles)
MR3
LCI period (110 cycles)
MR5
LCI period (70 cycles)
MR8
LCI period (20 cycles)
...

LCI Profile



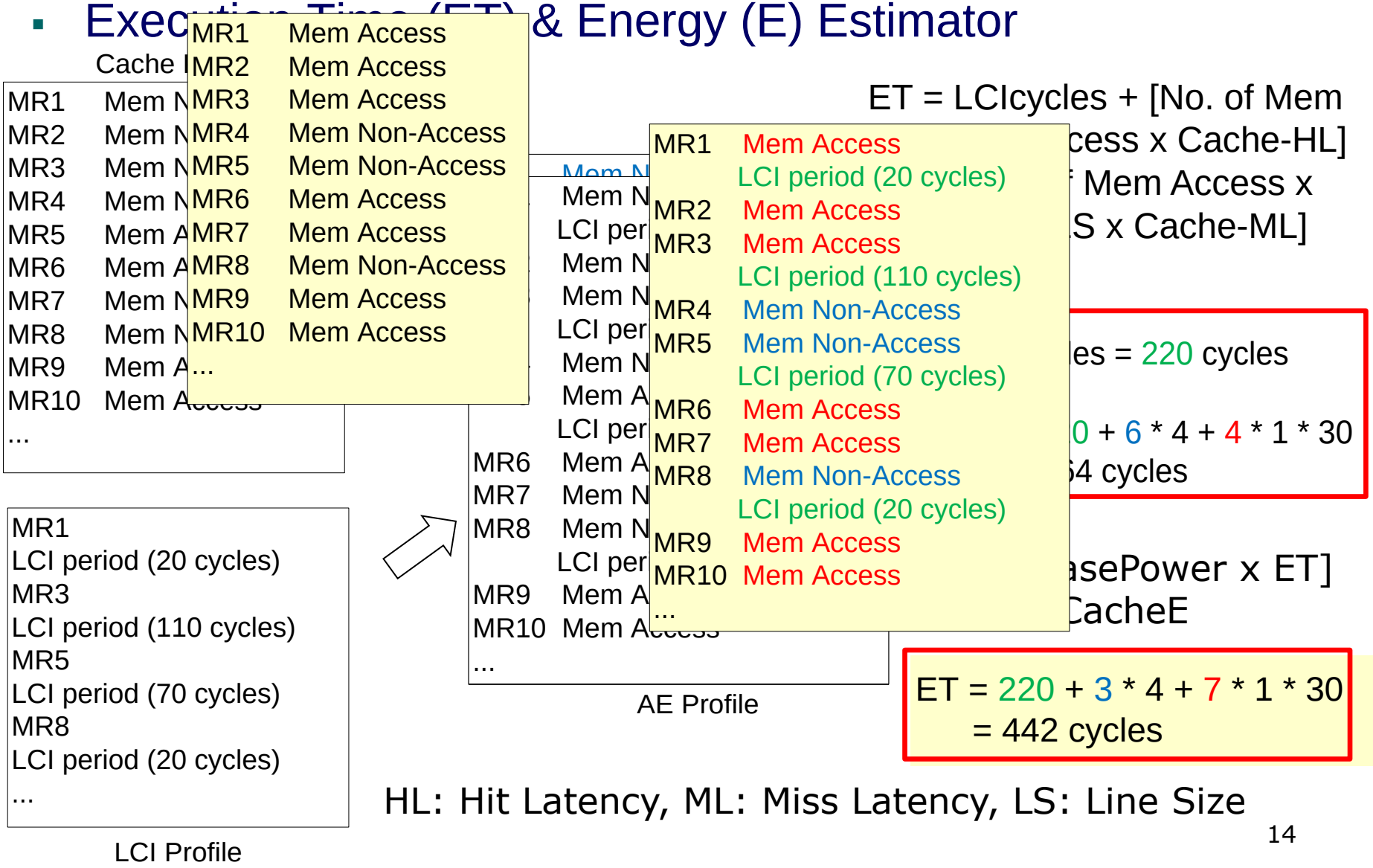
MR1 Mem Non-Access
LCI period (20 cycles)
MR2 Mem Non-Access
MR3 Mem Non-Access
LCI period (110 cycles)
MR4 Mem Non-Access
MR5 Mem Access
LCI period (70 cycles)
MR6 Mem Access
MR7 Mem Non-Access
MR8 Mem Non-Access
LCI period (20 cycles)
MR9 Mem Access
MR10 Mem Access
...

AE Profile

MR stands for Memory Request.

RExCache Framework, Cont'd

- Execution Time (ET) & Energy (E) Estimator



Results & Analysis

- **Execution Time Estimator**
  - RExCache

    ET = LCIcycles + [No. of Mem Non-Access x CacheHitLatency]
    + [No. of Mem Access x CacheLineSize x CacheMissLatency]

  - Traditional Method (from “Rapid Runtime Estimation methods for pipeline MPSoCs”) [extended for L2 Last-level cache]

    ET = [L1-Hits x L1-HitLatency] + [Total Instructions x **Avg.NCPI**]
    + [L2-Hits x L2-HitLatency] + [L2-Misses x L2-LineSize x L2-MissLatency]

    NCPI : Net Clock Cycles Per Instruction

    Avg.NCPI : Obtained from cycle-accurate simulations from a subset of cache configurations

Exploration Time Comparison

App.	Cycle-Accurate Simulator	Traditional			RExCache		
		TGP	CE	Total	TGP	CE	Total
adpcmD	**2h**	14.3m	8s	**14.4m**	20s	8s	**28s**
jpegE	**10h**	2h	13s	**2h**	2m	13s	**2.3m**
jpegD	**4h**	30.7m	8s	**30.8m**	42s	8s	**50s**
g721E	**7d**	1d	6m	**1d**	35m	6m	**41m**
g721D	**8d**	1d	6m	**1d**	37m	6m	**43m**
mpeg2E	**116d**	16.4d	16m	**16.7d**	9h	16m	**9.3h**
mpeg2D	**32d**	4.1d	8m	**4.1d**	3h	8m	**3.1h**
H.264E	**257d**	35.8d	2h	**35.9d**	19h	2h	**21h**

TGP : Trace Generation and Processing

- includes the time to compute average NCPI in Traditional Method
- includes the time to compute LCI cycles in RExCache

CE : Exploration of Cache Configurations

Detailed Analysis of Execution Time and Energy Estimators

App.	RExCache				Traditional estimators Extended to L2 cache			
	Exe. Time		Energy		Exe. Time		Energy	
	Abs. Err.(%)		Abs. Err(%)		Abs. Err(%)		Abs. Err(%)	
	Avg.	Max.	Avg.	Max.	Avg.	Max.	Avg.	Max.
adpcmE	0.00	0.00	18.63	30.42	0.67	4.44	19.34	31.03
adpcmD	0.02	1.73	19.69	31.54	1.08	9.24	21.28	33.38
jpegE	0.26	1.17	13.64	23.38	2.46	4.19	14.31	23.99
jpegD	0.03	0.29	12.38	22.90	2.00	2.91	13.15	23.63
g721E	0.00	0.57	13.76	27.04	3.90	11.98	13.68	25.50
g721D	0.00	0.01	13.69	27.75	3.16	13.73	13.92	26.69
mpeg2E	0.00	0.01	17.25	26.65	0.23	1.05	18.18	27.58
mpeg2D	0.01	0.43	16.66	26.55	0.22	1.75	17.64	27.71
H.264E	0.11	0.40	13.63	22.67	1.24	3.26	14.10	23.40

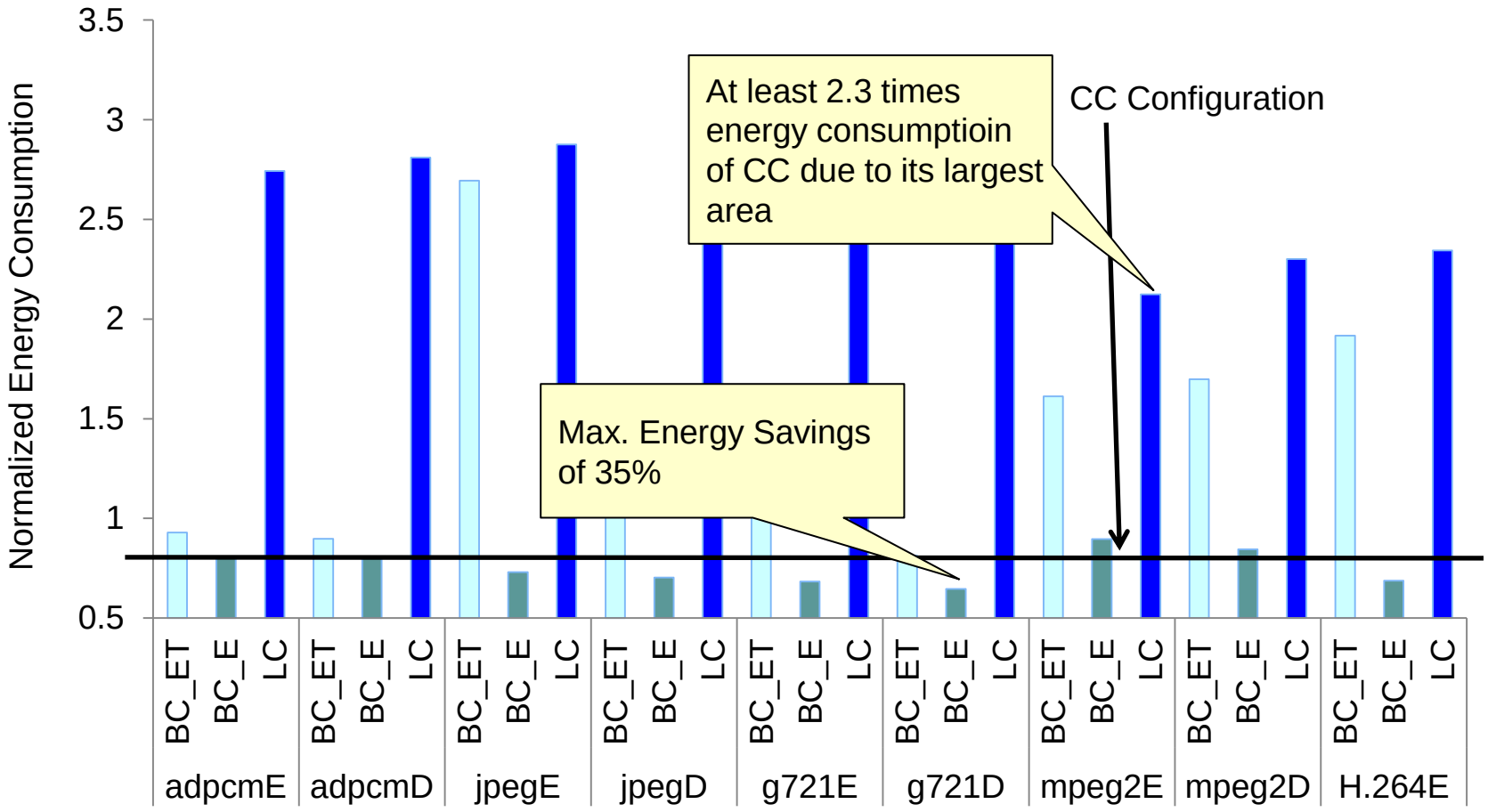
Execution Time Analysis

CC: System with Common Cache Configuration [64KB CacheSize, 64B line size, direct-mapped L2 cache]

BC_ET: System with Best Cache configuration w.r.t Execution Time

BC_E: System with Best Cache configuration w.r.t Energy consumption

Energy Consumption Analysis

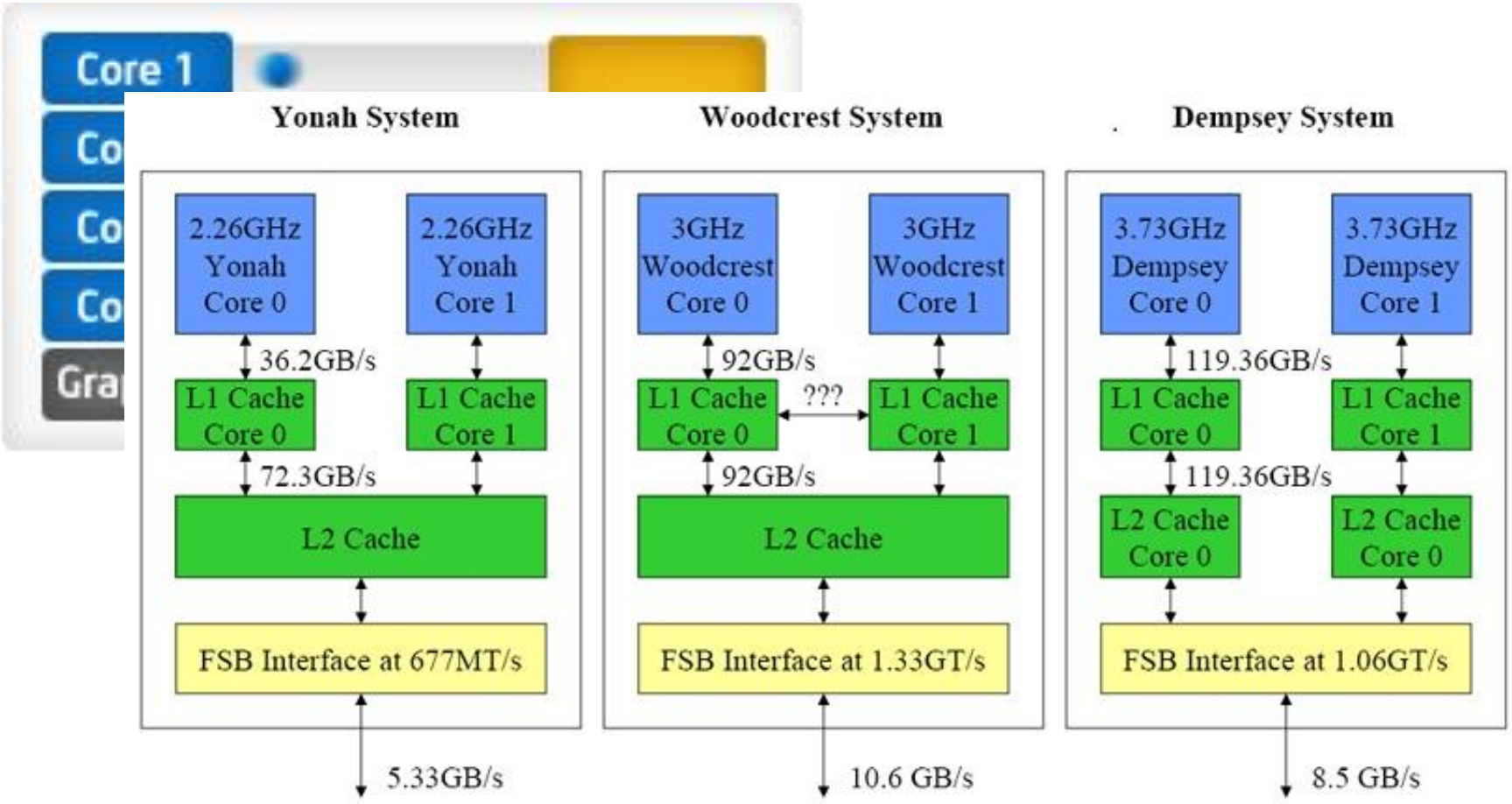


Conclusions

- **Proposed rapid exploration of unified last-level cache**
- **To avoid slow cycle-accurate simulations, proposed RExCache Framework**:
  - integrates cycle-accurate and cache simulator
  - includes fast and highly accurate performance estimator
  - includes fast and reasonably accurate energy estimator
- **Suitable last-level cache configuration selected by RExCache:**
  - up to 50% execution time reduction w.r.t common cache
  - up to 35% energy reduction w.r.t common cache
- **Exploration time for possible 330 configurations**
  - RExCache took only a few hours
  - Cycle-accurate took several days
  - Traditional method took 98% more than RExCache

Thank you

Q & A



Intel's Merom