

architect(s/ure) at Work

Alum@Alma Talk '23@CSE-IITM

Biswa

biswa@cse.iitb.ac.in

Shhh... for the next 3600000000000ns

Btw, feel free to ask/interrupt

Two checkpoints: Slide #35 and 67

My journey at IITM (CS10S003 – CS10D019)

Madhu Mutyam <madhumutyam@gmail.com>

Mon, Nov 30, 2009, 10:17 PM

to me ▼

Hi Panda,

We made an offer to you. You will receive the offer letter in a week time.

Regards,
Madhu Mutyam

Biswabandan Panda

Last position held : PhD Scholar (Roll No: CS10D019)

Duration with CSE Dept : Sep 2012 to Jul 2015

Advisor(s) : Shankar Balachandran

[Link to Personal Homepage](#)

Brownian motion between RISE lab and PACE lab, through DCF

Un Dino Ki Baat Hai (Same T-shirt 😊)



BSB-349 😊



CWC 2011

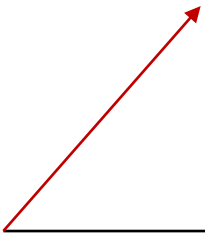


PACT 2012

Takeaway: Consistency is the key 😊

The ups and downs

Joined MS,
December
2009 😊 😊



The ups and downs

Joined MS,
December
2009 😊 😊



December 2011,
Fixing simulator 😞
Research 😞😞😞
Scooped too 😞😞😞

The ups and downs

Joined MS,
December
2009 😊 😊



December 2011,
Fixing simulator 😞
Research 😞😞😞
Scooped too 😞😞😞

January 2012
Progress meeting 😞 😞

The ups and downs

Joined MS,
December
2009 😊 😊

December 2011,
Fixing simulator 😞
Research 😞😞😞
Scooped too 😞😞😞

January 2012
Progress meeting 😞 😞

February 5, 2012
Applied for MS to
PhD?
Committee? Nah
Committee? Nah
Committee?
Ohhhkkkk

The ups and downs@Research

Joined MS,
December
2009 😊 😊

All is well 😊

March 2014
to July 2015

Papers at
DATE, PACT,
CAL, TACO

December 2011,
Fixing simulator 😞
Research 😞😞😞
Scooped too 😞😞😞

The real story 😊 😞 😊

Thanks @CSE-IITM

January 2012
Progress meeting 😞 😞

February 5, 2012
Applied for MS to
PhD?

Committee? Nah
Committee? Nah
Committee?

Ohhhkkkk , *will I ?*

After 2015: It is a daily affair

Papers at DATE, CAL, ISPASS, PACT, MICRO, ISCA ☺

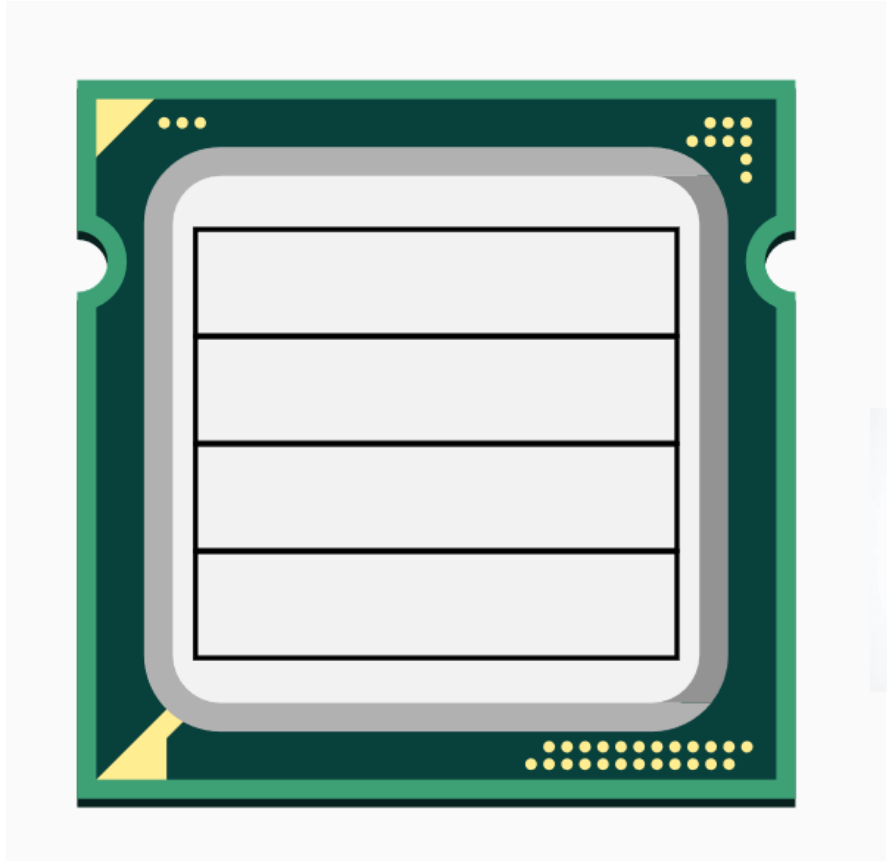


Tons of questions ?



Shhh... Time for the Talk now

Microarchitecture:101



Programmer
(user/compiler/OS)

Architecture

ISA

Registers

Caches, TLBs,

Branch predictors,

Prefetchers, Interconnect,

Out-of-order execution



Not exposed to programmer

Memory

But, Microarchitecture research is dead?



BENEFIT ▾

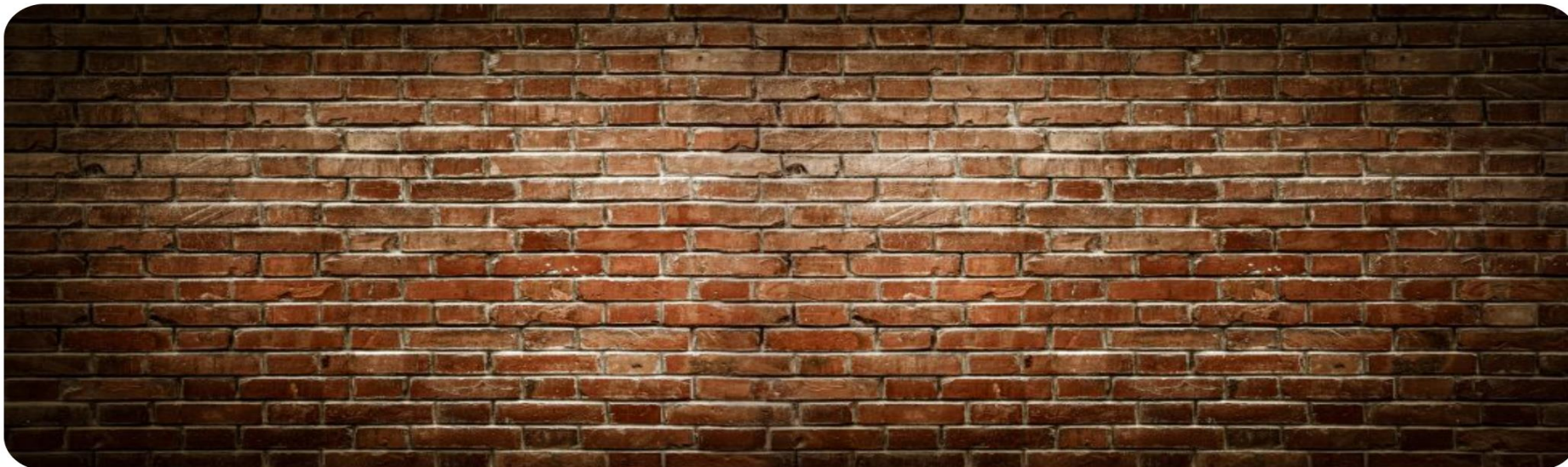
CONTRIBUTE ▾

DISCOVER ▾

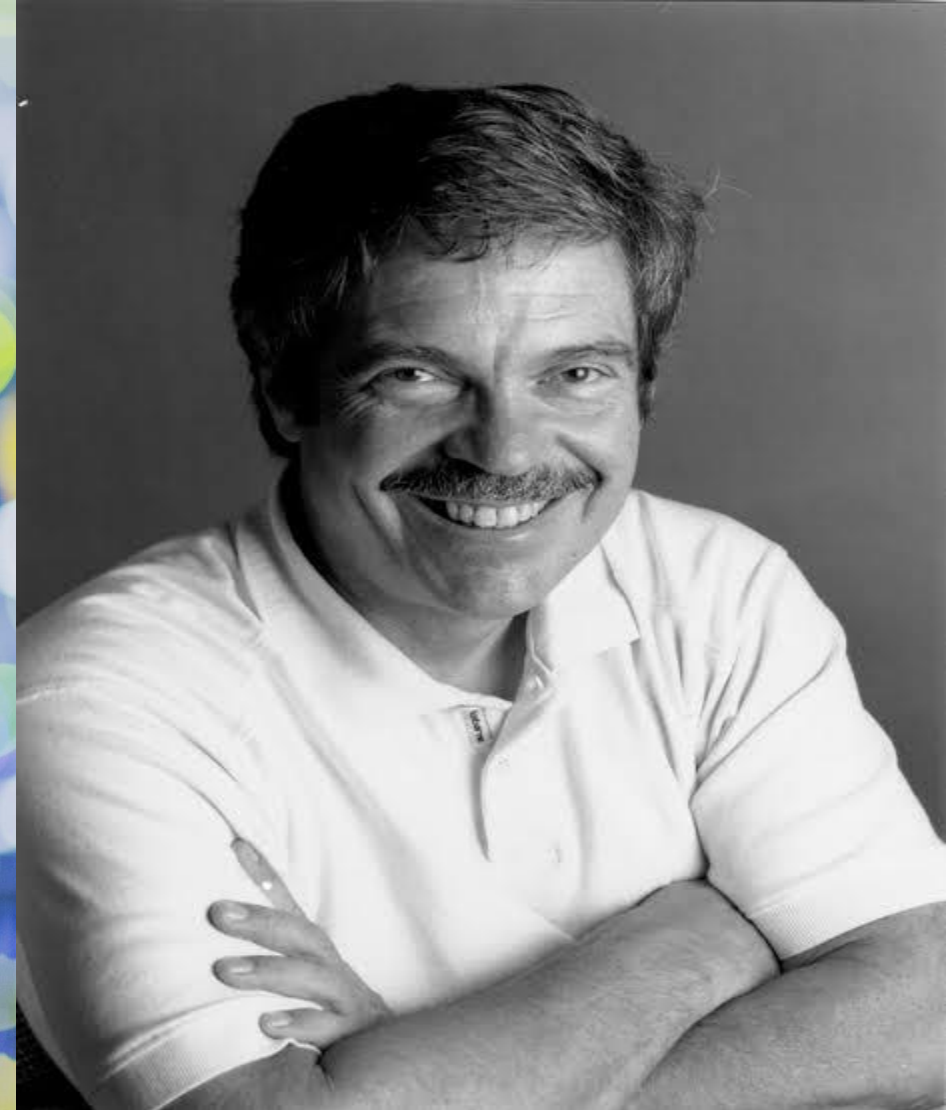
CAR

The Microarchitecture Research Wall and its Renaissance

by Biswabandan Panda on May 31, 2022 | Tags: Academia, Microarchitecture, Research



The day is April 14, 2023. The MICRO deadline is just a few hours away, and micro-architects are skeptical about



Hardware is new software

*People who are really serious about software should make their own hardware
- Alan Kay, father of PCs*

and... Domain specific processors

AWS Graviton Processor

Enabling the best price performance in Amazon EC2

Get Started with AWS Graviton-based EC2 Instances

OPINION

Microsoft's Innovative 4-Processor PC

By Rob Enderle | May 30, 2022 4:00 AM PT | [Email Article](#)

[Tweet](#) 6 [Share](#) 0 [in Share](#) 0 [Share](#) 6

Facebook is just crazy enough to make its own processors

Job listings for a chip design team have surfaced online.

NVIDIA Grace CPU

Purpose-built to solve the world's largest computing problems.



Google Replaces Millions of Intel's CPUs With Its Own Homegrown Chips

By Anton Shilov published June 04, 2021

YouTube now uses homegrown Argos VCUs

Dead? Stop listening ...

Context@Talk

Programs (including OS) running on CPUs

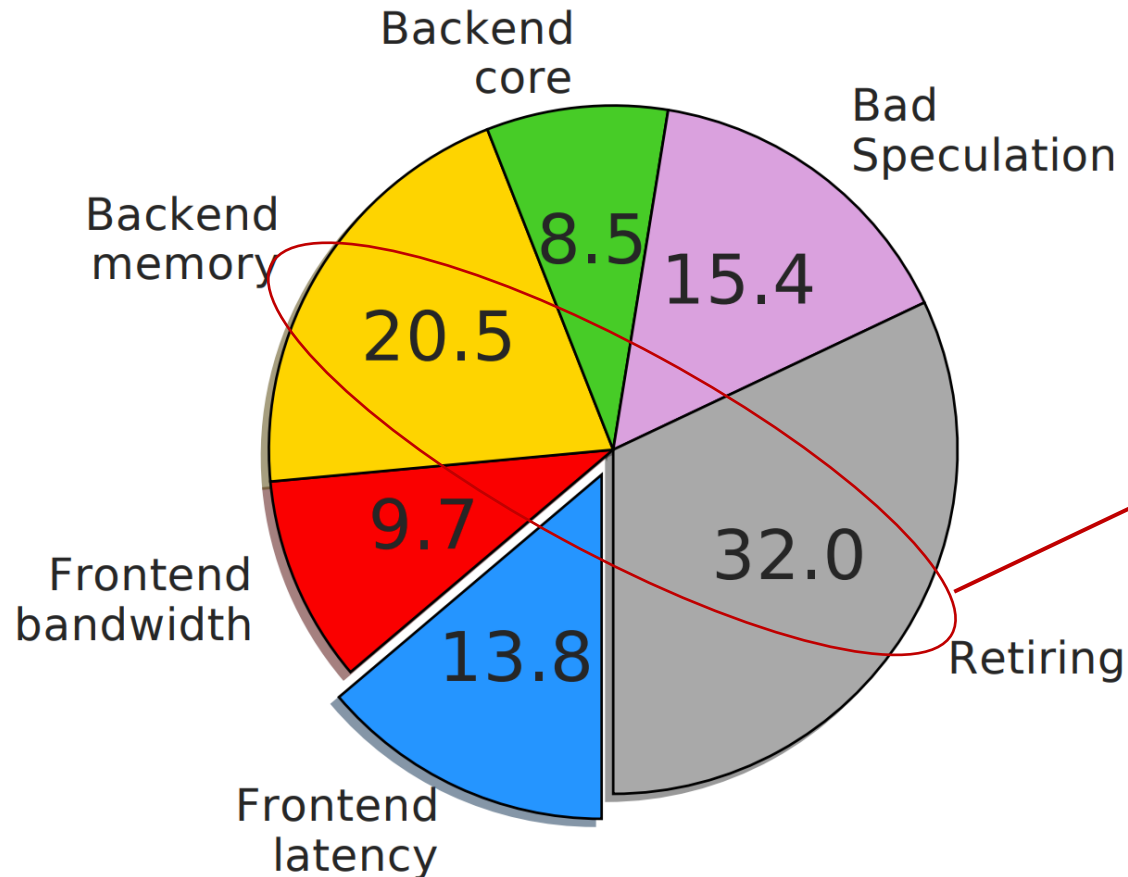


Let's run an application

Web search, next time chatGPT maybe 😊

CPU: Intel Haswell

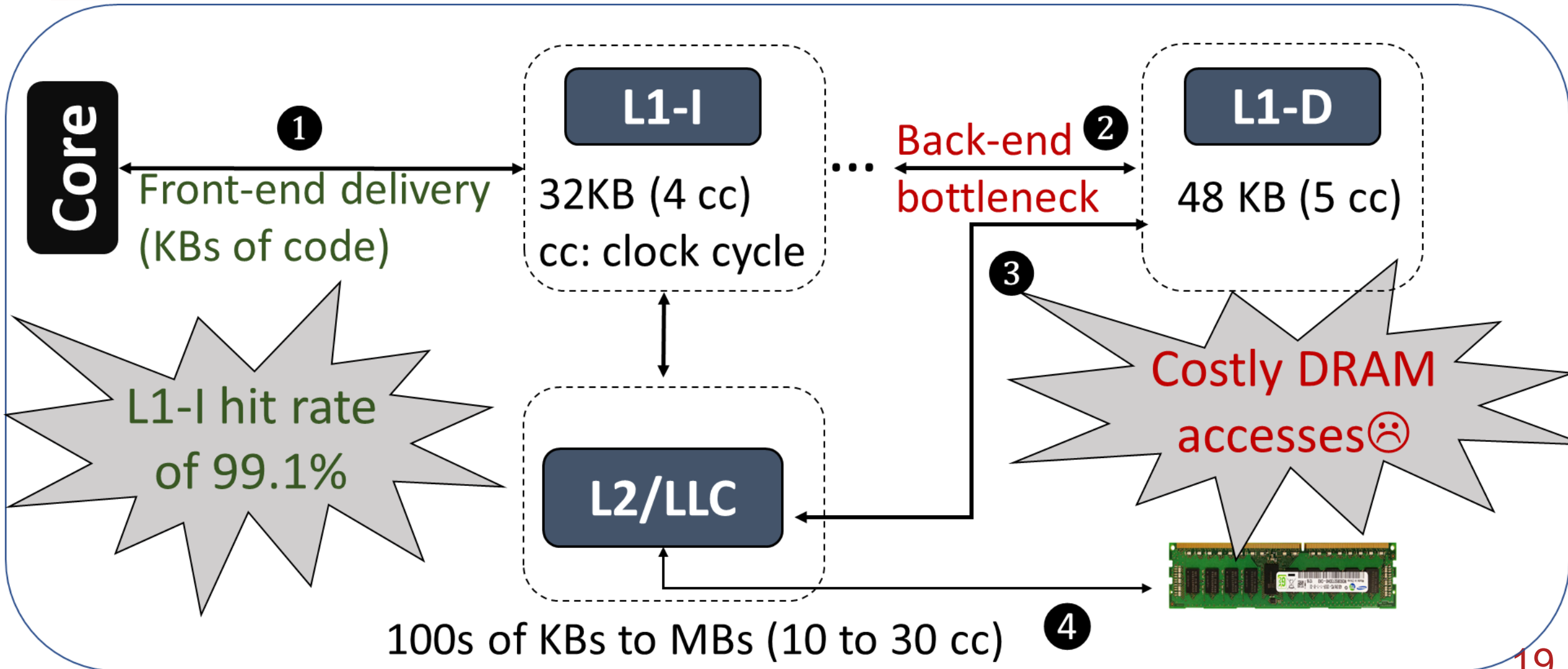
And the bottleneck [Google's websearch]



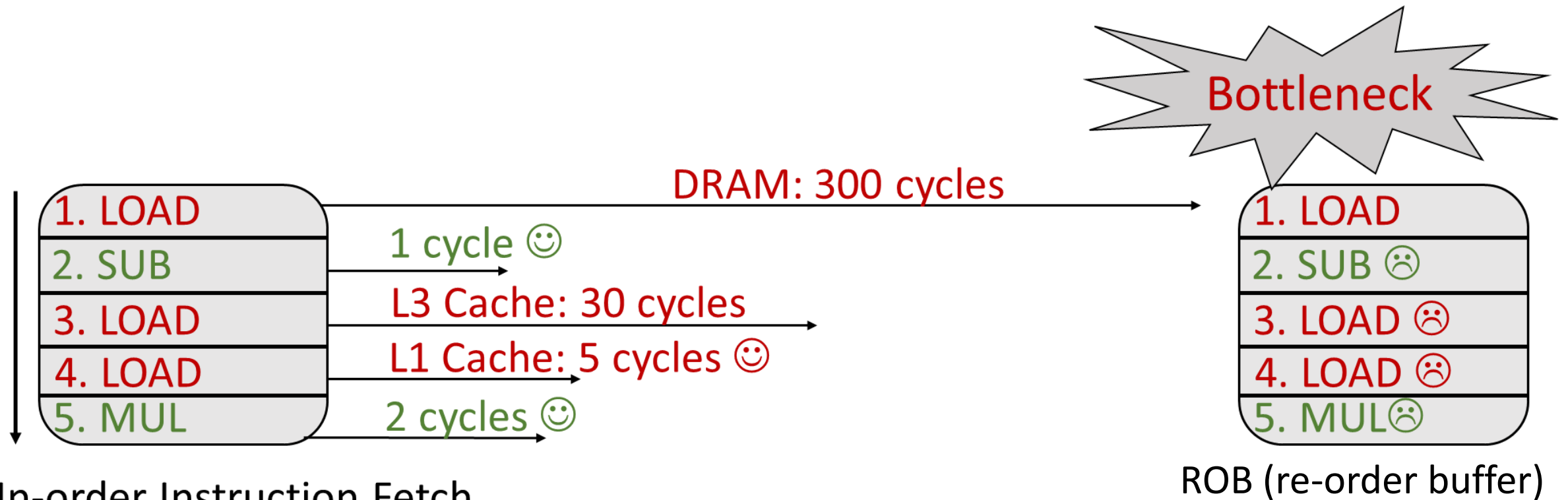
Focus of my talk

More than 50% ☹️ even with caches ☹️ ☹️

Back-end memory bottleneck (100s cc)



Retiring bottleneck in an out-of-order Core



In-order Instruction Fetch
(Multiple fetch in one cycle)

Even an L1 hit has to wait for a DRAM access ☹️

Microarchitect's dream (impossible though)

Core

L1-D hit rate of 100% (a dream 😊)



L1-D

L1-D Prefetcher

Memory access latency: five cycles

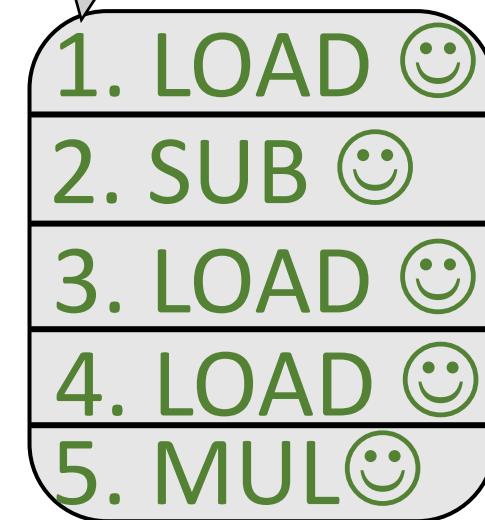
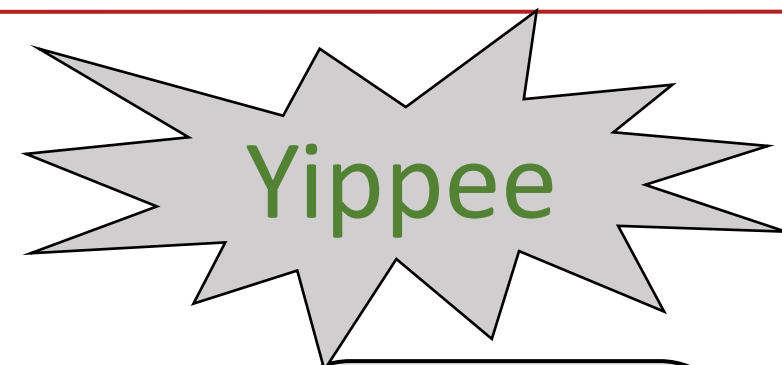
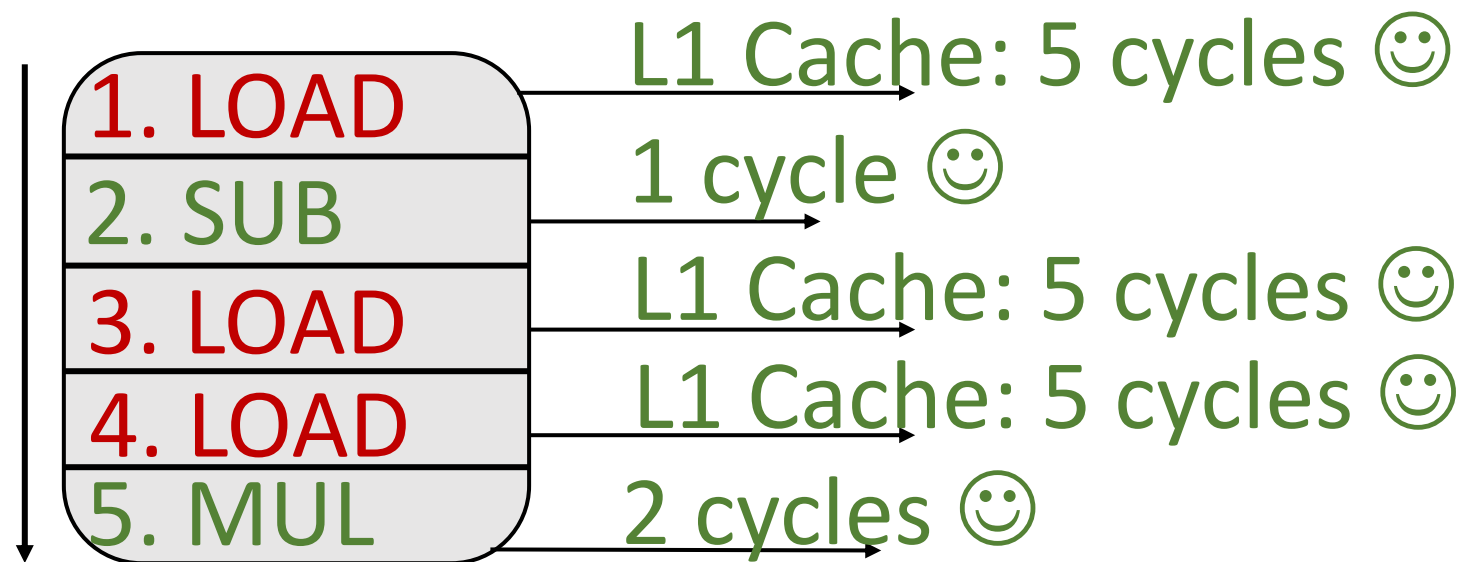
L2

L3

Memory access 😊



Retiring bottleneck in an out-of-order Core



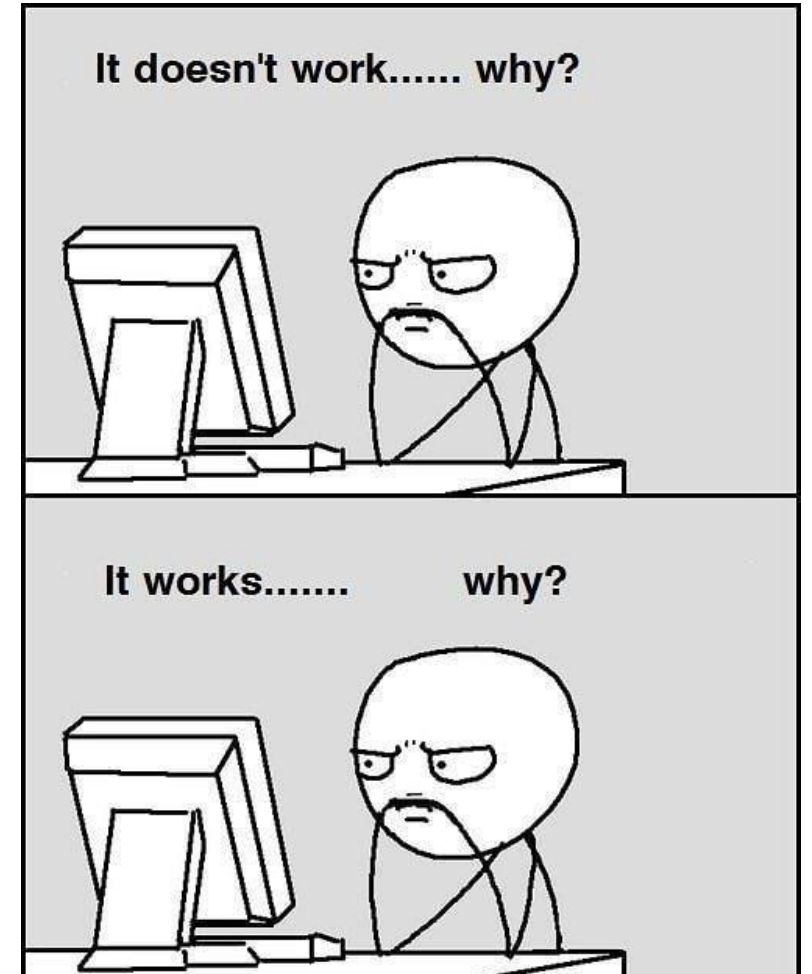
ROB (re-order buffer)

Even an L1 hit has to wait, Nah, no more 😊

Microarchitects ?

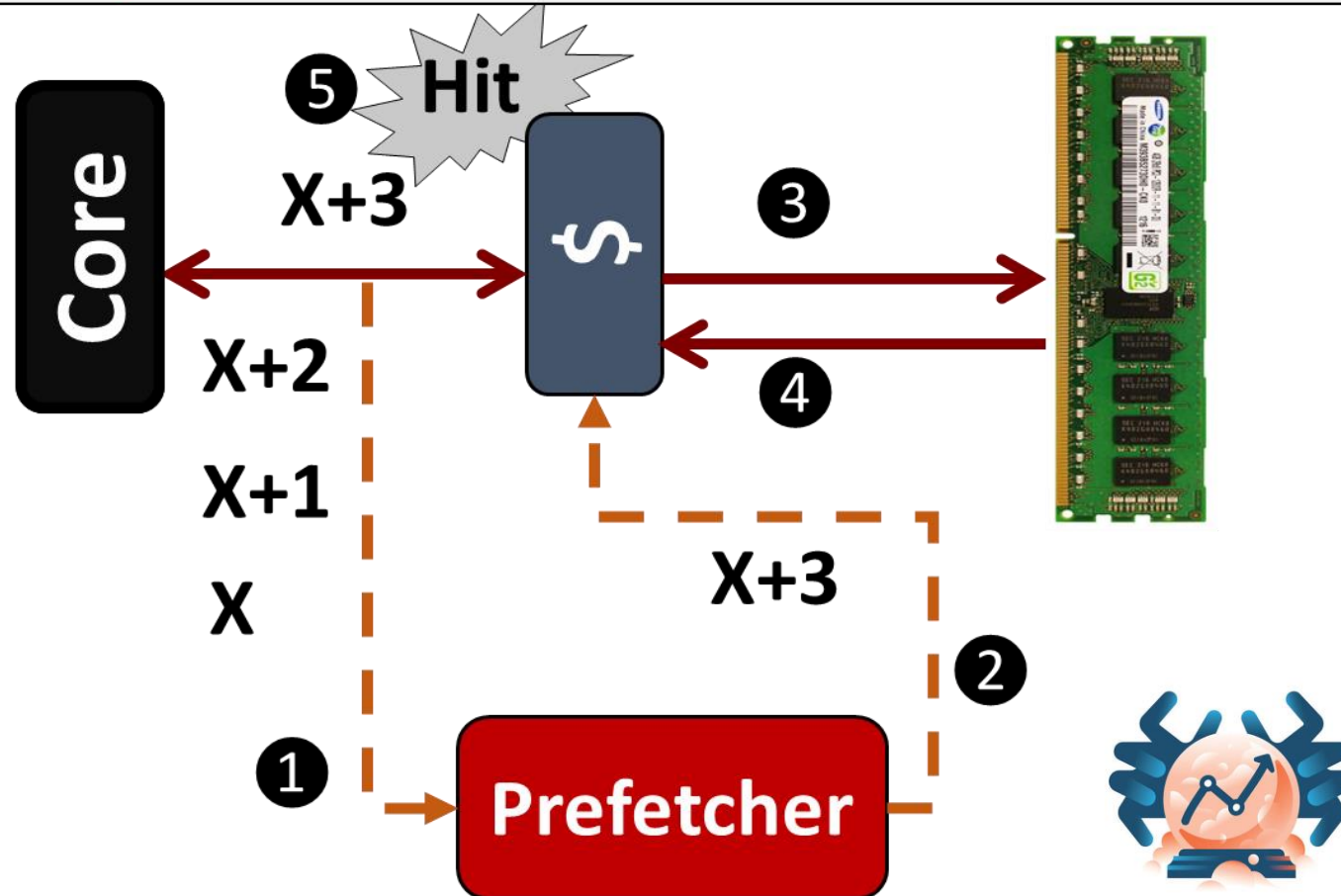
Microscopic view on
microarchitecture problems

Microarchitecture solutions



Microarchitecture: A Hardware Prefetcher

A hardware prefetcher can make impossible, possible 😊



The reality from last 30 years

Academia and industry: L2 prefetchers

Challenges: many for a practical L1 prefetcher ☹️

A lightweight/high-performing L1 prefetcher:
Impossible

Guru Gyan



Start solving a research problem
when the most@research
community: “we are done”

Andre Seznec,
My mentor, post-PhD



What about impact?

- Write the first flagship conference paper on a research topic (US centric, large groups)

or

- Write the final flagship conference paper on a research topic (*My approach, ekla chalo re* 😊)

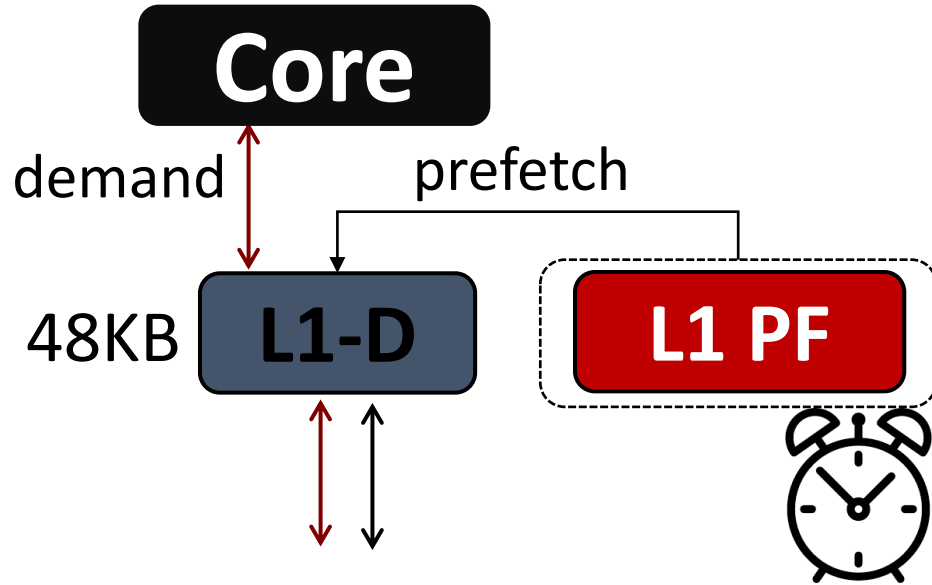
Is it Really Impossible? 2018 around

*Is it **possible** to design a competitive/practical L1 data prefetcher?*

Industry: No ☹️

Academia: No ☹️

Why? Challenges

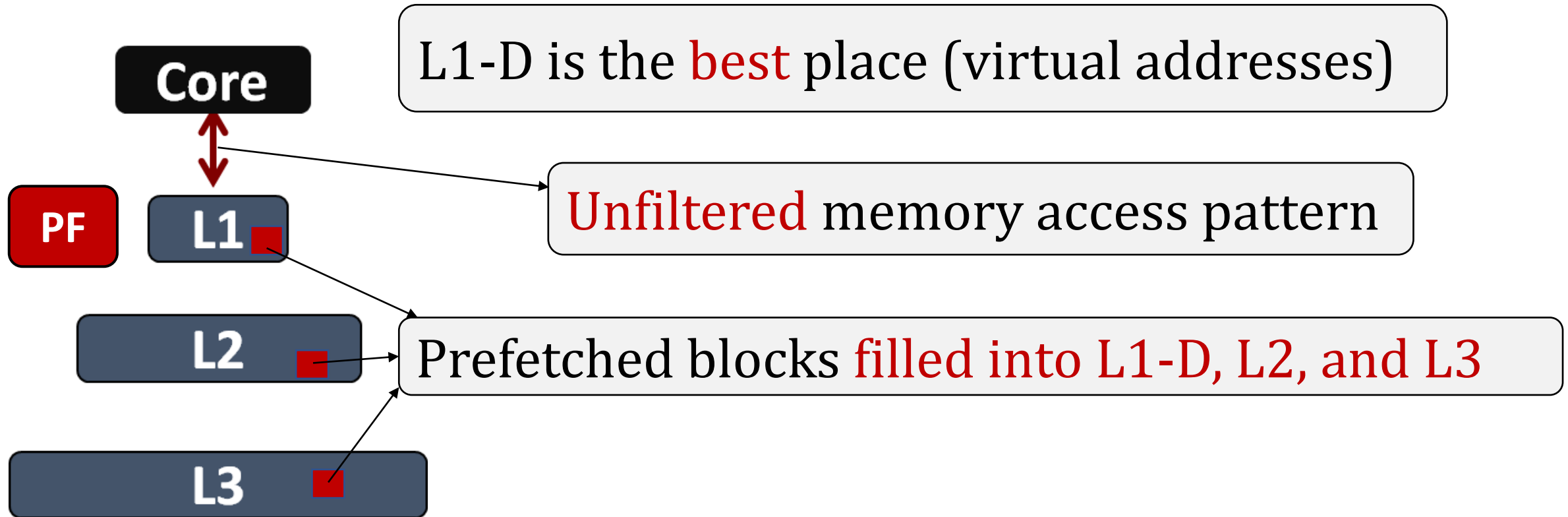


Should be **lightweight**, L1-D is 48KB

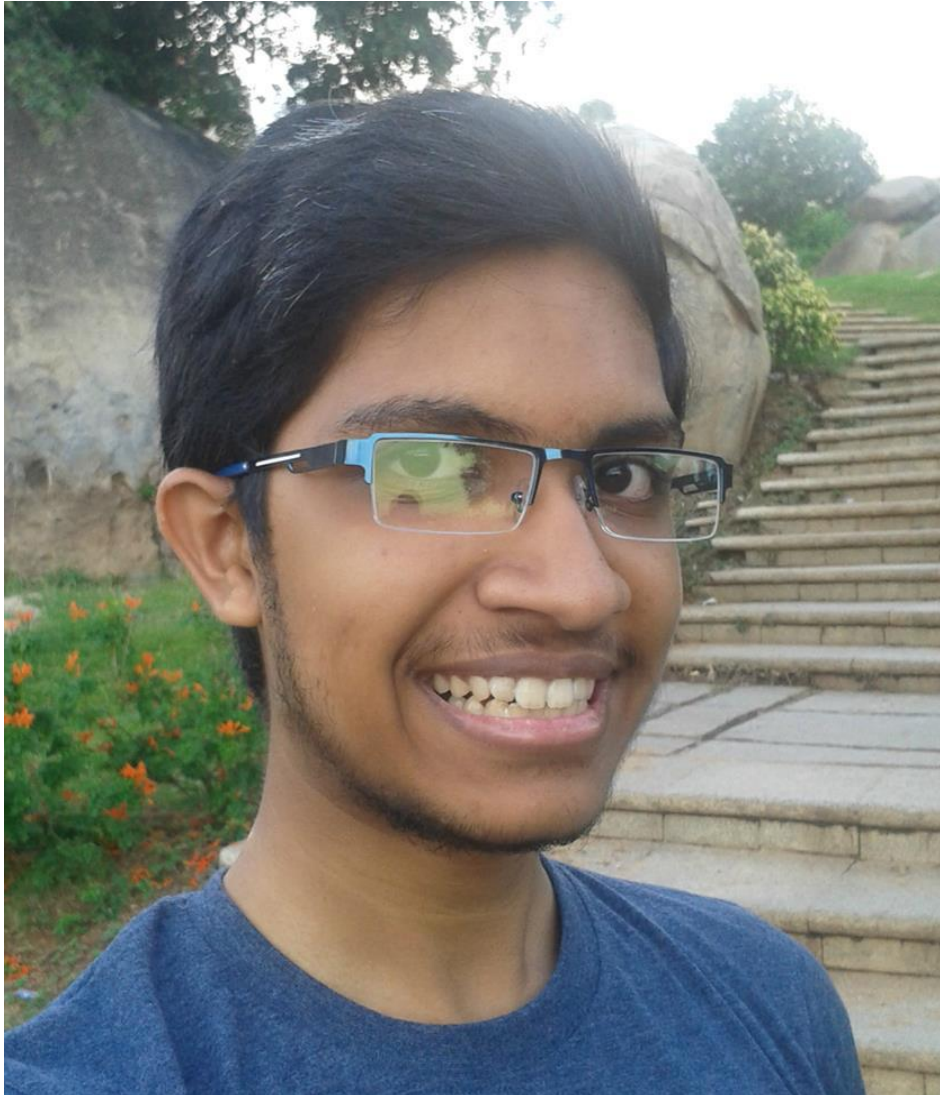
lookup latency: **five cycles**
prefetcher should be **agile**

Many more issues: bandwidth, port contention, and others

Opportunity: L1-D Prefetching



Oh yes, it is possible



*Bouquet of Instruction Pointer
Classifier based prefetching
[ISCA 2020]*

*Samuel, Remote Mentee, BITS Pilani
[2018-2020]*

M.S. @ Texas A&M

A large crowd of stylized human figures in various colors (dark blue, light blue, orange, white) is shown. The figures are arranged in a circular pattern, with one white figure standing out in the center. The background is a gradient of dark blue and purple.

Message I: Dare to be different

Well, It is possible



SPP [MICRO '16]:	6KB, 35%
PPF [ISCA '19]:	34KB, 39%
DSPATCH [MICRO '19]:	40KB, 42%
Bingo [HPCA '19]:	119KB, 43%
IPCP [ISCA '20]:	800B, 45%

Bouquet approach for prefetching

What is the FUSS? Only 2% improvement

1% improvement matters

- *“Microarchitects can kill their grandmothers to get 0.5% improvement”*
- *“1% improvement in industry is a cause for celebration”*
- *“1% improvements on multiple microarchitecture ideas make a big difference in the final revenue”*





PAUSE for a minute

What next?

Community started looking at L1 prefetchers 😊

Hang on.

What next for me and my mentees?

Why no FUSS about energy? 😞



Do not forget energy 😊

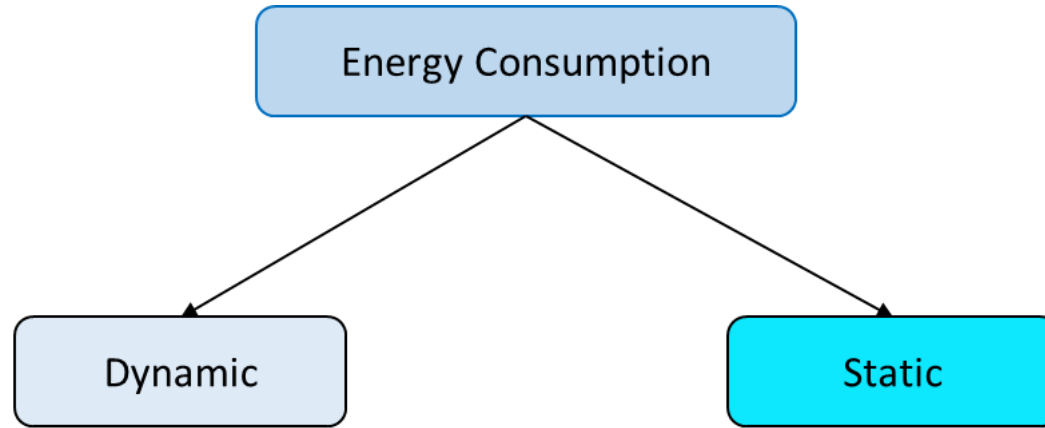


*Energy-Efficient Hardware Data
Prefetching [IEEE CAL 2021]*

*Neelu, M.S. by Research, IIT Kanpur
[2019-2021]*

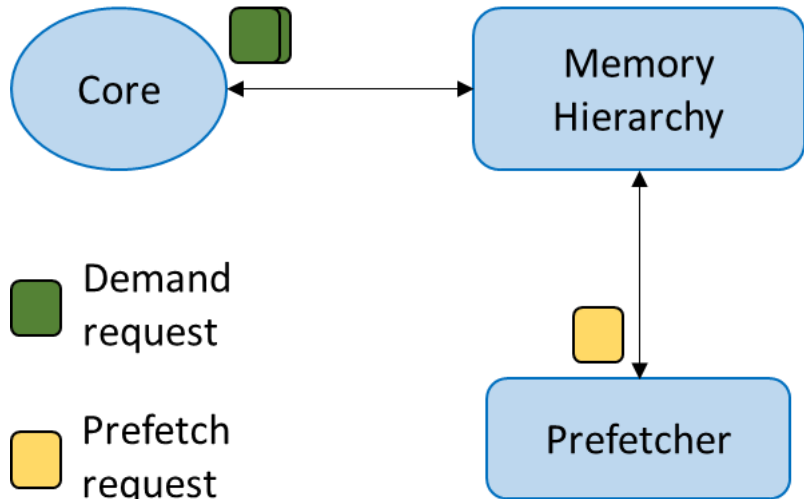
Ph.D.@EPFL

Energy Consumption: 101



Energy \propto Time

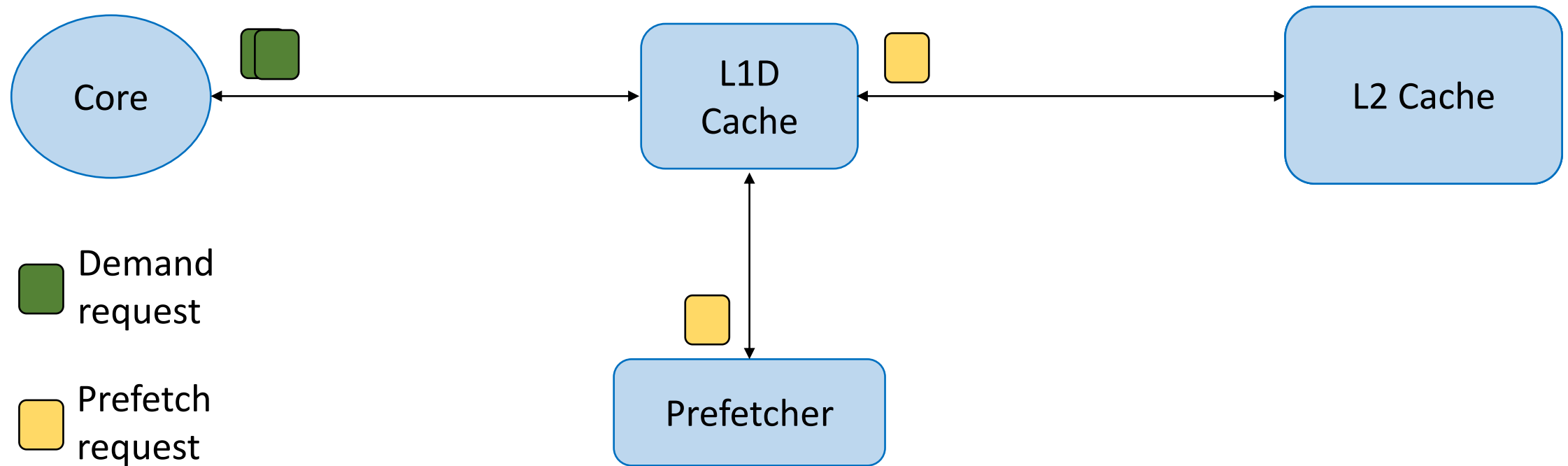
Less execution time
=
Lower static energy
consumption



More requests
=
Higher dynamic energy
consumption

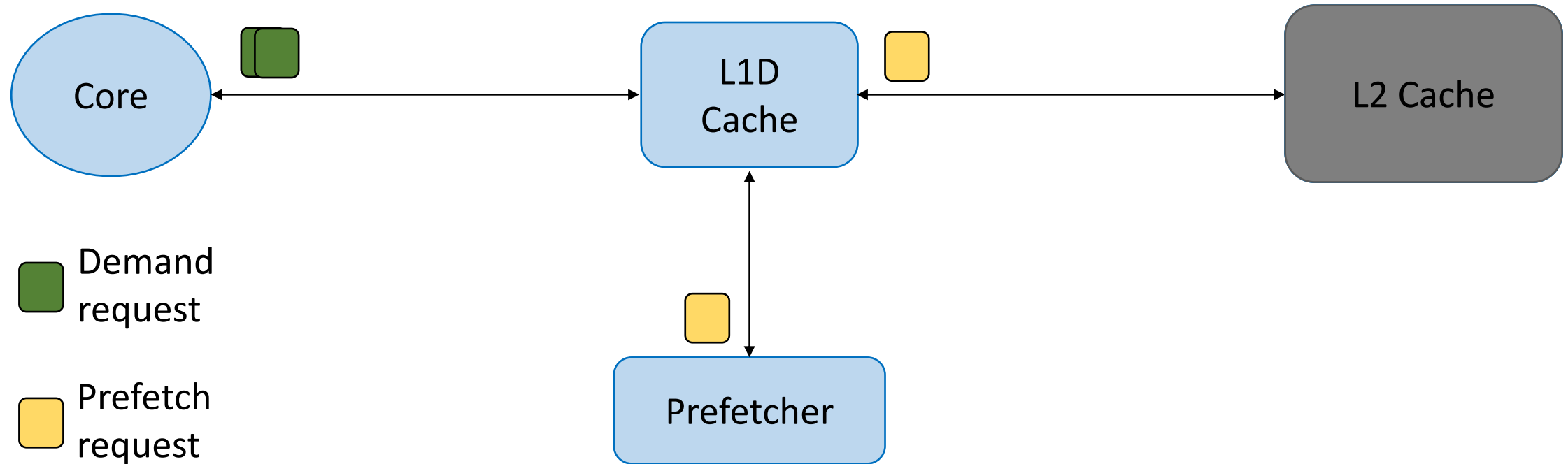
What if there is power-gating?

Power-gating for mitigating static energy



More requests in the memory hierarchy can lead to higher energy consumption

Power-gating for mitigating static energy



More requests in the memory hierarchy can lead to higher energy consumption

Let's Quantify it

No publicly available tool that can provide faithful numbers 😞 😞

We showed our results to major research labs 😊 😊

They said No, Yes, and No 😞 😊

Took 10 months and finally Intel said YES





Message II: Hang in there,
persist++

Where is the problem?



Coverage: Fraction of cache misses that become hits

Accuracy: Fraction of prefetch requests that provide hits

Of course, accuracy is not 100%, not even 90% 😞

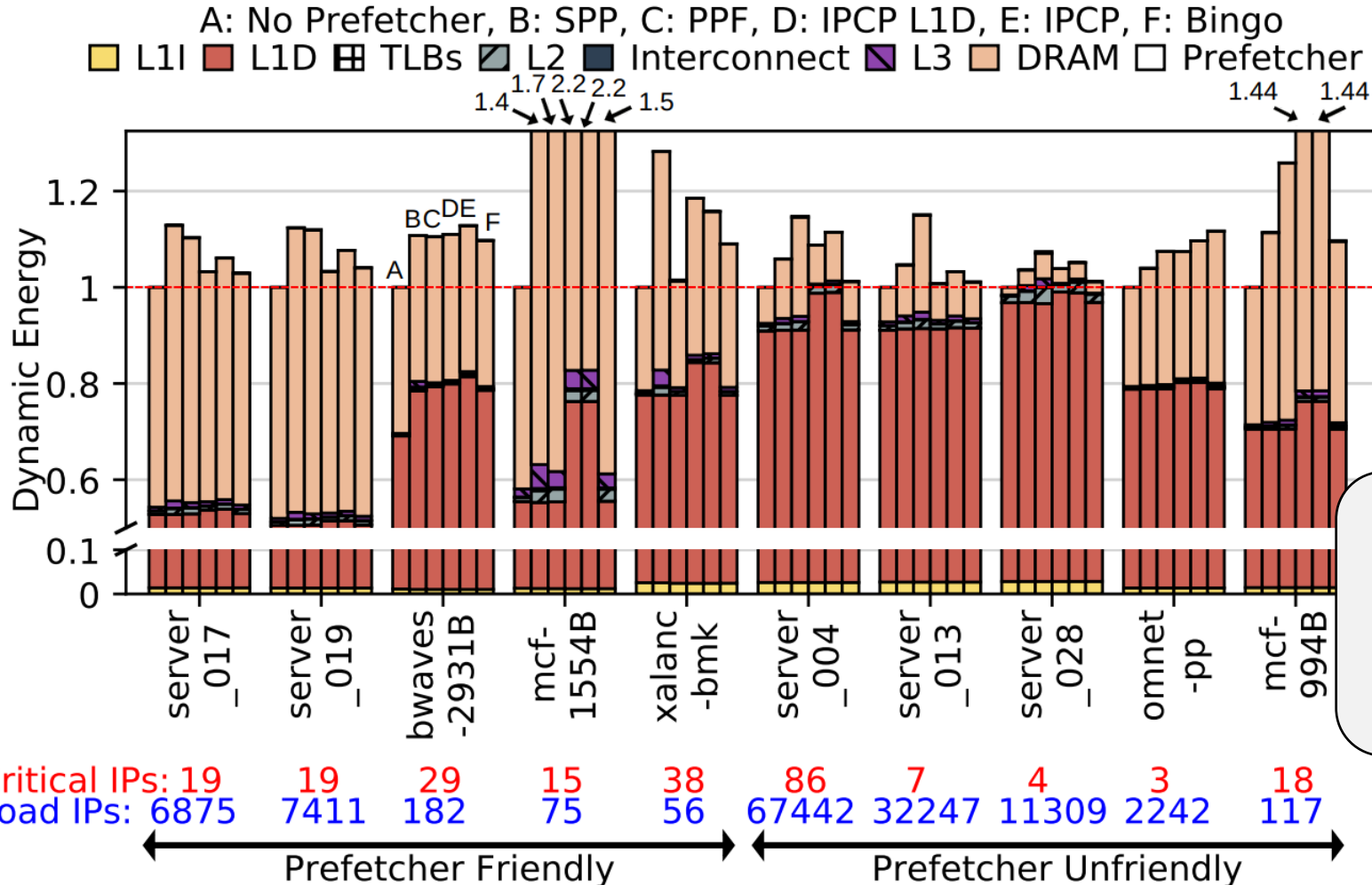
Trivial Solution: Instruction Criticality

In-order Instruction Sequence	Interaction With Memory	Execution Cycles
Load R1, [R2]	LLC Miss	150 cycles
Add R3, R1, R4		1 cycle
Load R5, [R6]	L2 Hit	15 cycles
Mult R7, R3, R8		3 cycles

← ROB Head

Prefetch data only for critical loads that delay retiring instructions

Why is this a big deal?



Only tens of IPs are critical among hundreds/thousands! 😊

Is this New?

Focusing Processor Policies via Critical-Path Prediction

Brian Fields

Shai Rubin

Rastislav Bodík

Computer Sciences Department
University of Wisconsin–Madison

{fields,shai,bodik}@cs.wisc.edu

Performance Oriented Prefetching Enhancements Using Commit Stalls

R Manikantan

R Govindarajan

Indian Institute of Science, Bangalore, India

RMANI@CSA.IISC.ERNET.IN

GOVIND@CSA.IISC.ERNET.IN

Criticality Aware Tiered Cache Hierarchy: A Fundamental Relook at Multi-level Cache Hierarchies

Anant Vithal Nori*, Jayesh Gaur*, Siddharth Rai†, Sreenivas Subramoney* and Hong Wang*

*Microarchitecture Research Lab, Intel

Criticality-Based Optimizations for Efficient Load Processing

Samantika Subramaniam

Anne Bracy†

Hong Wang†

Gabriel H. Loh

Georgia Institute of Technology

College of Computing

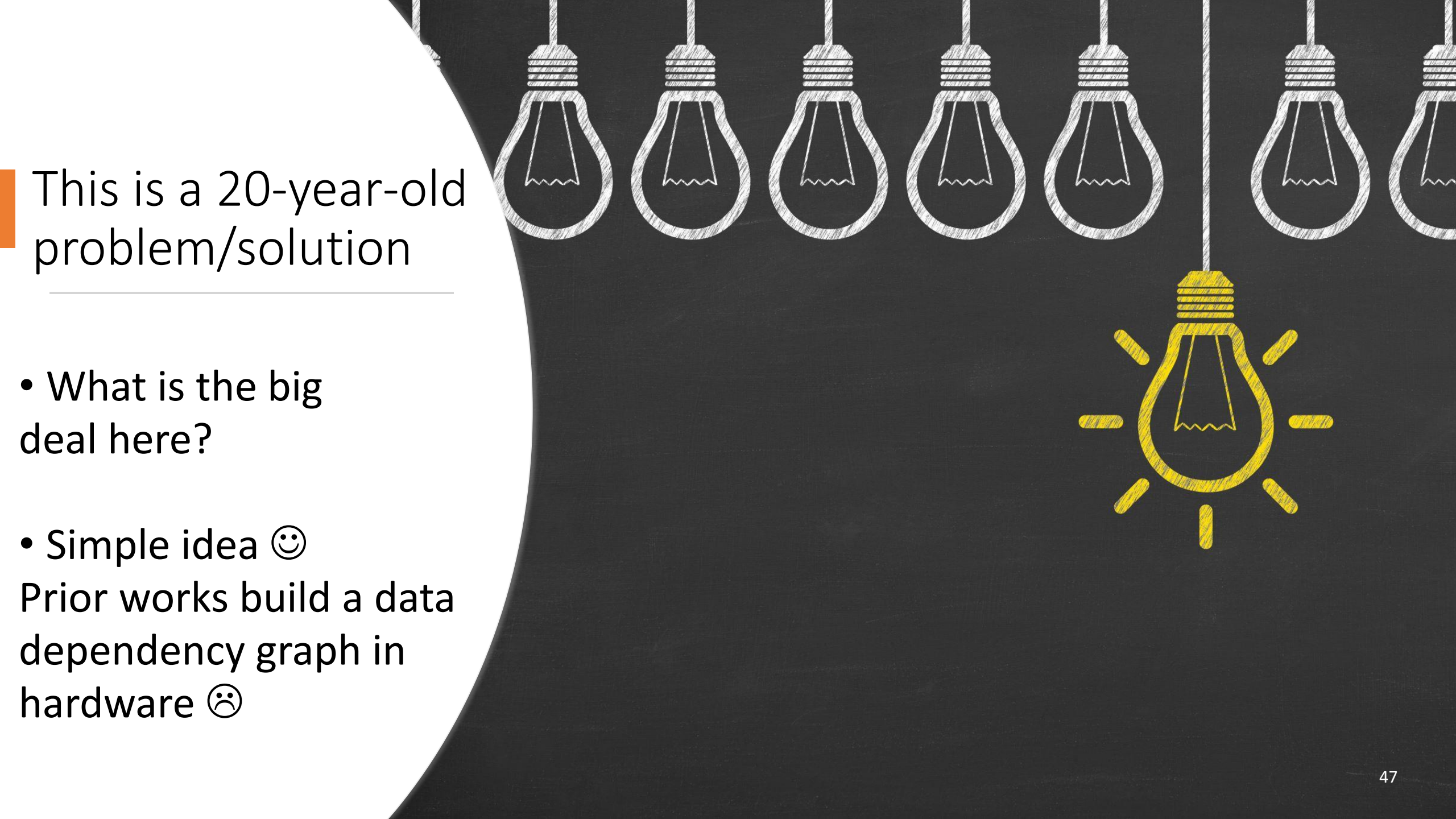
Atlanta, GA, USA

†Intel Corporation

Microarchitecture Research Laboratory

Santa Clara, CA, USA

Key insight: Only the instructions on the critical execution path matter to performance



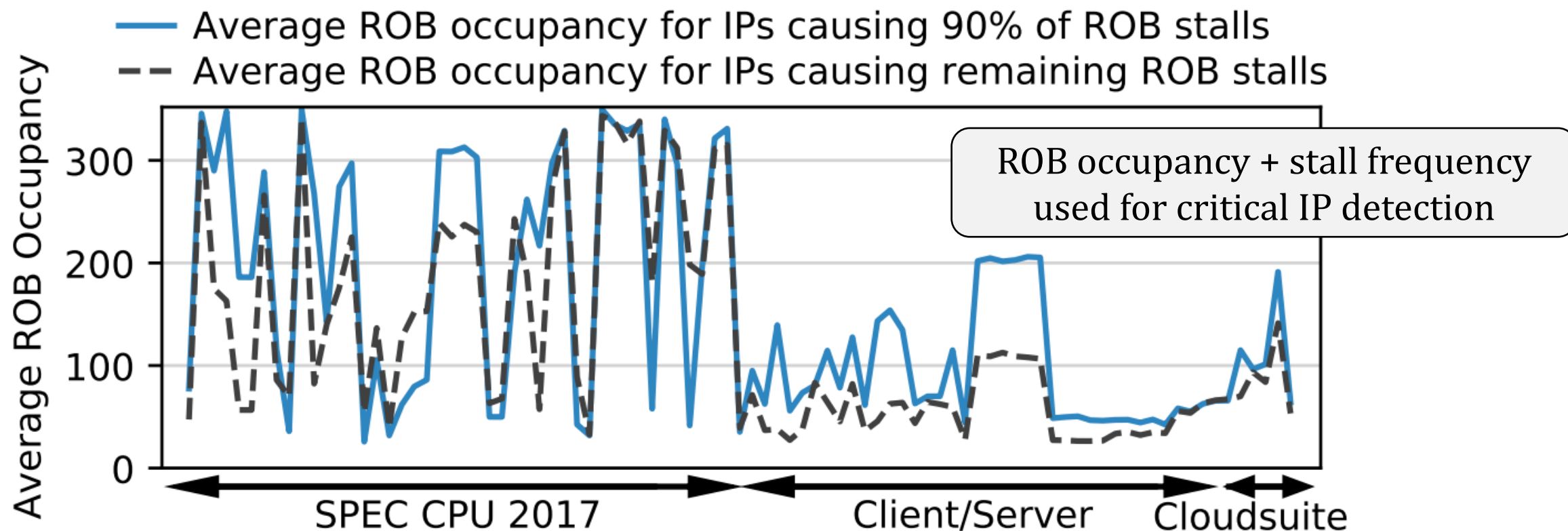
This is a 20-year-old problem/solution

- What is the big deal here?

- Simple idea 😊

Prior works build a data dependency graph in hardware 😞

Simplest metric: ROB Occupancy + Stall frequency



IPs causing 90% of ROB stalls have a higher average ROB occupancy

Stalls Per Kilo Cycles (SPKC):
IPs causing 90% of ROB stalls: 27.7
IPs causing remaining ROB stalls: 3

Connecting the dots

Idea	Storage <i>(Lower)</i>	Performance <i>(Higher)</i>	Energy <i>(Lower)</i>
ISCA '20	1KB	45%	50%
CAL'21	+2.5KB	43%	45%

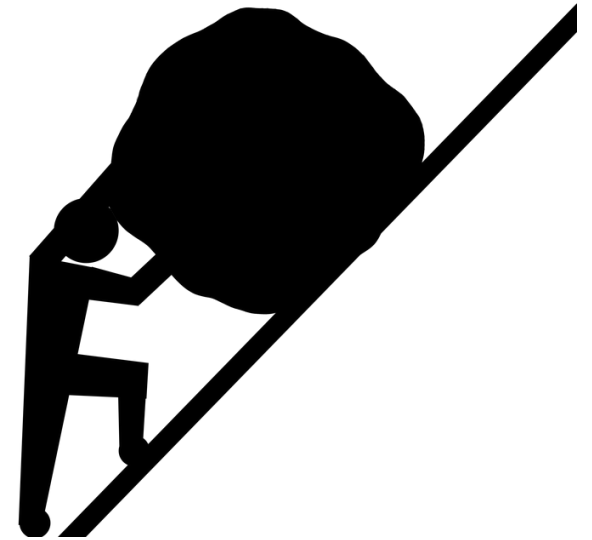
Is this a good deal?



2% loss in performance, no big deal 😊 ??

Hardware Prefetching research:

2 to 3% performance improvement in **2 years**



The Pertinent Question



Can we have a prefetcher that is energy-efficient and yet high performing?

Why not?

Pushing the limits of an L1 Prefetcher

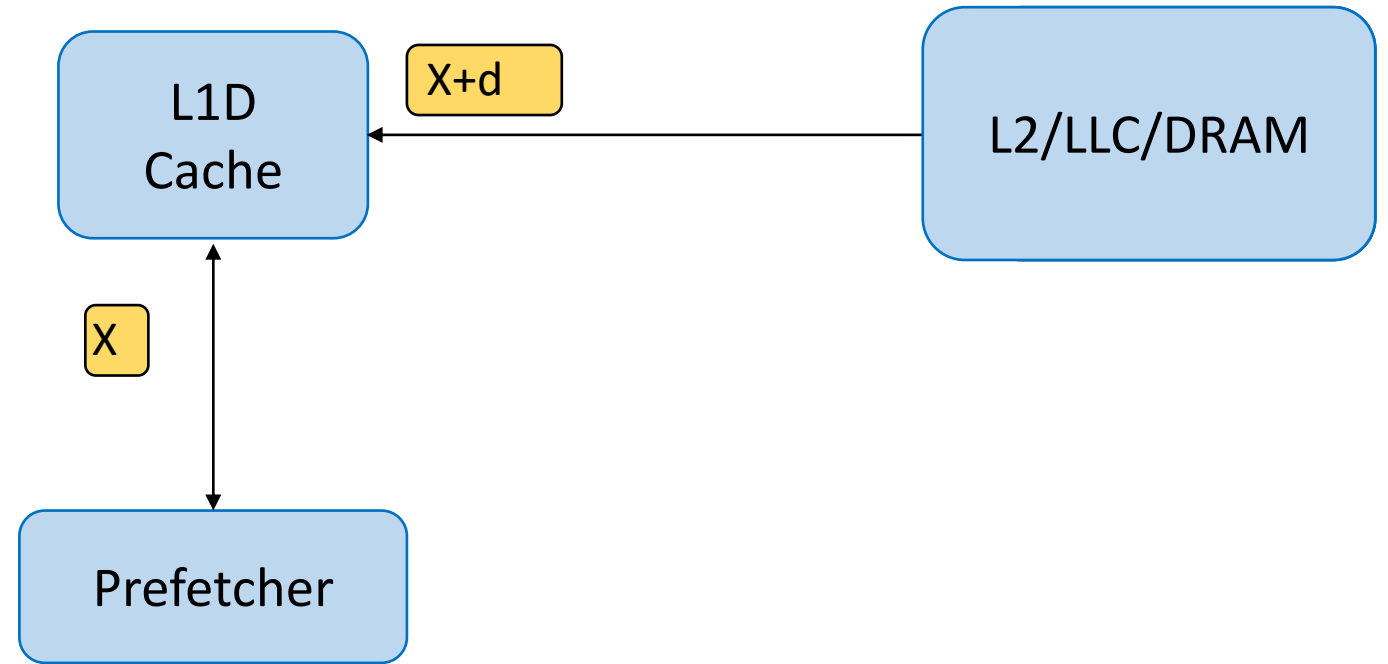


*Berti, State-of-the-art L1 Prefetcher
[MICRO 2022]*

Agustin, Ph.D. Universidad de Zaragoza

Defending next week 😊

The Problem and the approach



Access to X , prefetch $X+d$

Approach: given an access to address X , what should be the d ?

We call it the “delta”

Observation-1

```
for (i = 0; i < N; i=i+16) {  
    sum += c[i];  
    if(i%4==0)  
        reduce += d[i];  
}
```

Deltas for each load (IP) is different

Where existing local prefetchers fail?

Inorder LOADs to L1



Out of order LOADs to L1

```
for (i=1;i<=7;i++)  
{  
    x = a[i] // 1, 2, 3, 4, 5, 6, 7  
}
```

Strides: +1, +1, +1, +1, +1, +1 😊 😊

```
for (i=1;i<=7;i++)  
{  
    x = a[i] // 1, 3, 2, 4, 6, 7, 5  
}
```

Strides: +2, -1, +2, +2, +1, -2 😞 😞

Presenting Berti

Berti: per-IP **best request time**
aggregate of deltas

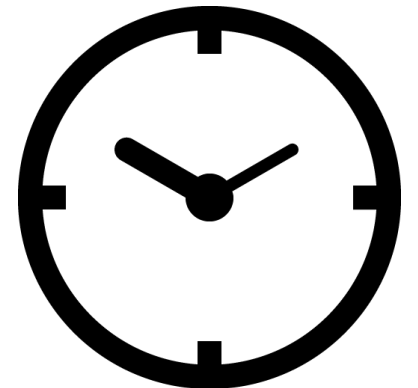
Why Time? Devil is in the details

Time to fetch the data into L1 is not constant

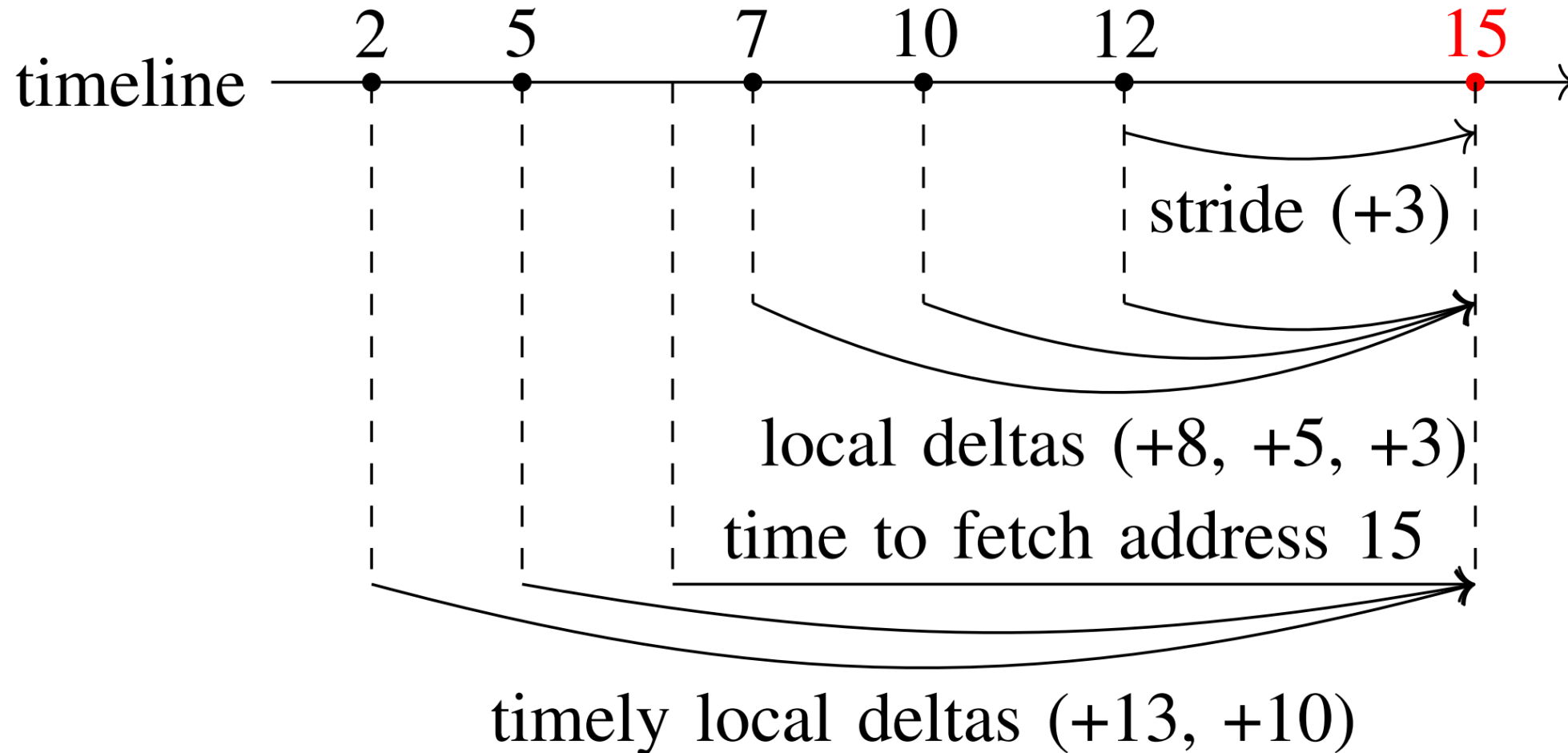
22 to 2098 cycles with an average of 278 cycles on a 4-core system

Each IP has a different time to fetch (locality, reuse, queueing delay etc)

In summary out of order memory hierarchy



The notion of timely local deltas



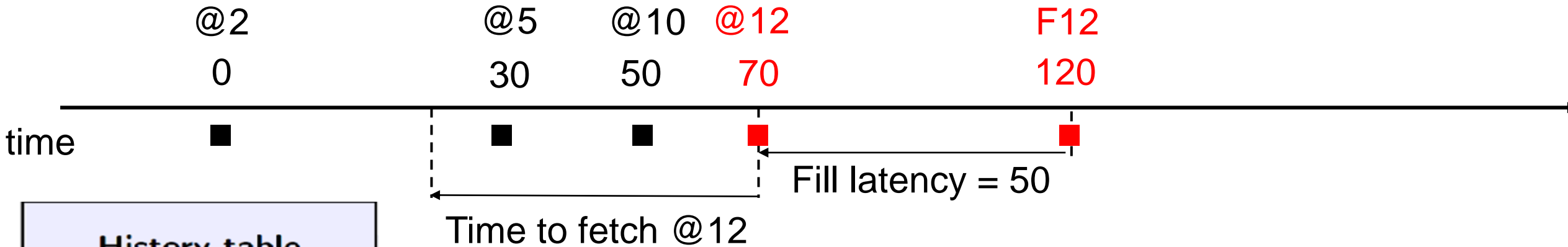
The updated question of interest

*“for an L1D access to address X , **what is the timely and accurate delta (d) that should be used for prefetching?**”*



Idea in one slide

Timely delta: +10



History table	
@	TimeStamp
2	0
5	30
10	50
12	70

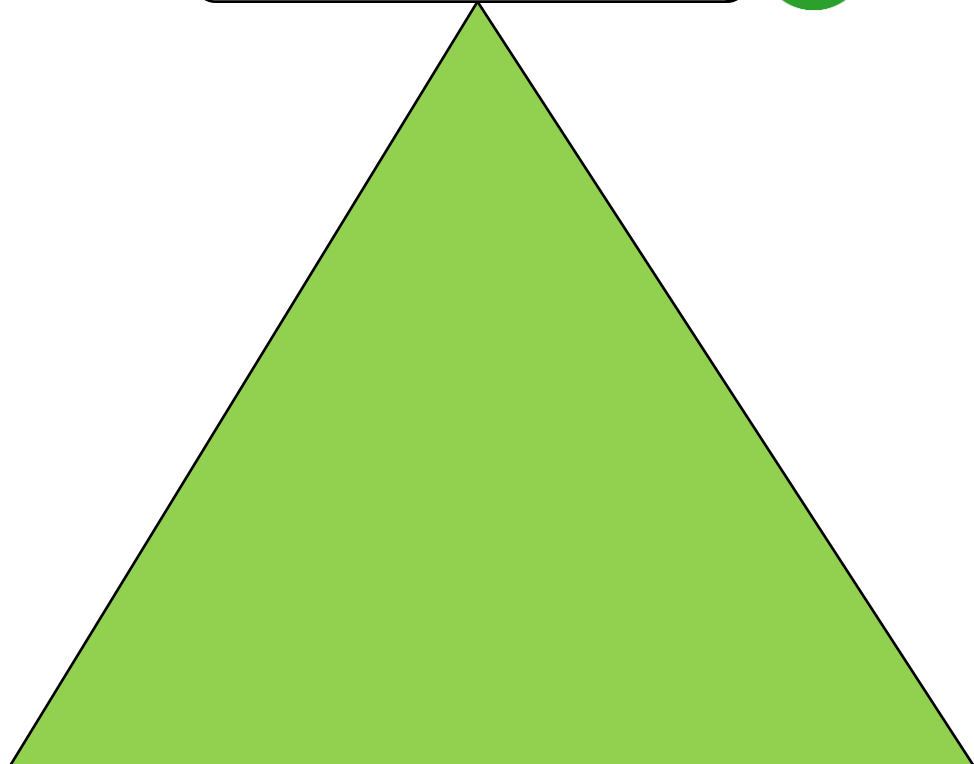
Table of deltas		
Delta	Coverage	Destination
+10	2/2 (100%)	L1D
+13	1/2 (50%)	L2

Our Insights

*“Timely deltas that provide the **best local coverage** also contribute to high global accuracy.”*

Berti as a package

High coverage 



High accuracy, low traffic and energy 

Low storage overhead 

What about performance?

Berti provides ~90% accuracy 😊 😊

Berti consumes additional ~11% energy


**3.5% performance improvement over IPCP
[ISCA 2020]**

Connecting the dots


Idea	Storage <i>(Lower)</i>	Performance <i>(Higher)</i>	Energy <i>(Lower)</i>
ISCA '20	1KB	45%	50%
CAL'21	+2.5KB	43%	45%
MICRO'22	+1.5KB	48.5%	11%



Message-III: Keep Pushing

A close-up, black and white photograph of a watch face. The watch has a dark dial with light-colored hour markers and hands. The hands are positioned to indicate a time around 10:10. The watch case is visible at the top and left edges. The background is dark and out of focus.

PAUSE for a minute

A solid orange horizontal bar located at the bottom of the image, spanning most of the width.

Finally, do not forget address translations 😊



*Address Translation Conscious Cache
Hierarchy [ISPASS 2022]*

*Vasudha, M.S. by Research, IIT Kanpur
[2019-2021]*

Qualcomm Microarchitecture Team

Virtual Memory

App. 1

Virtual address space

Printf (“%d”, &a);

App. 2

Virtual address space

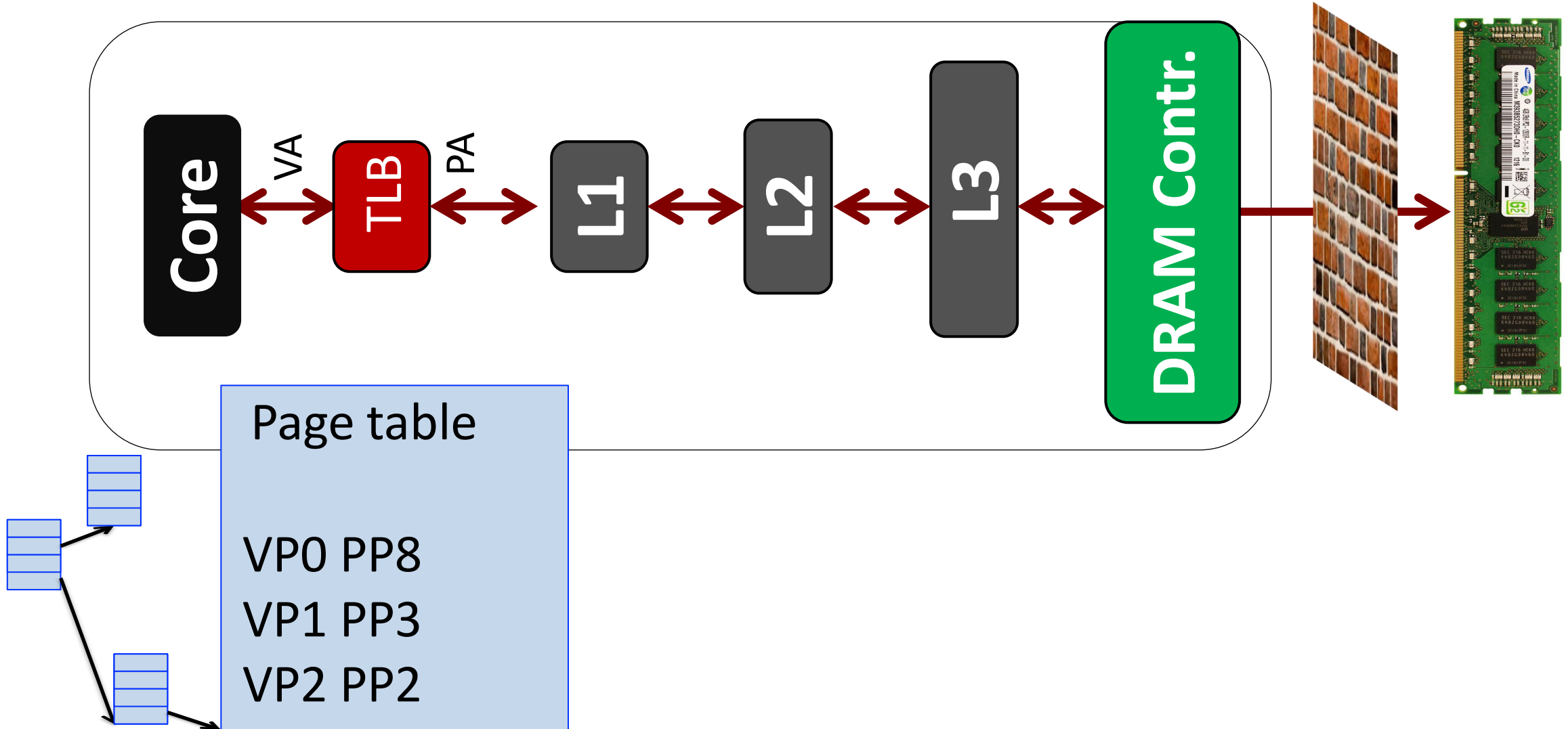
Printf (“%d”, &a);

100

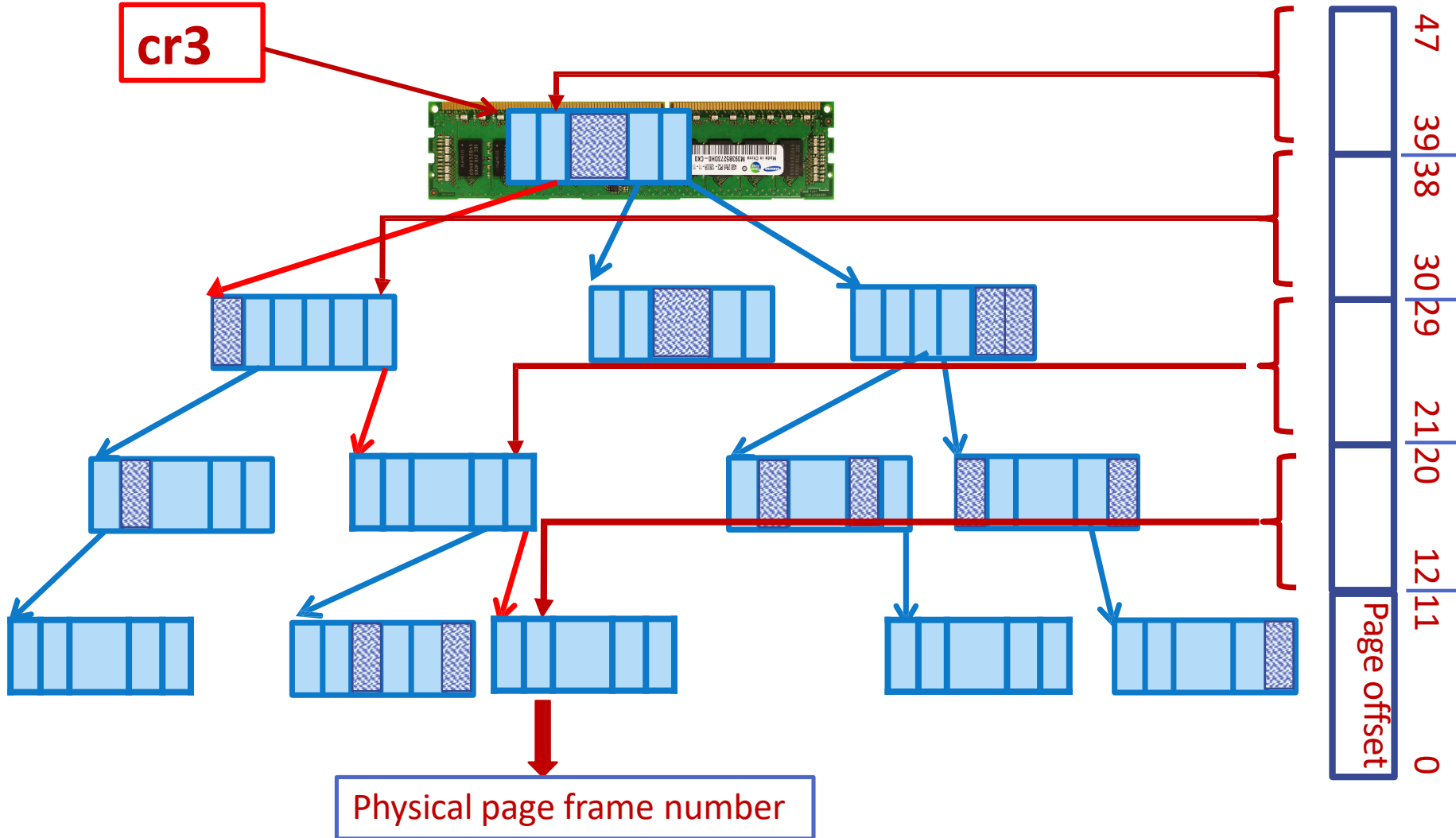
100



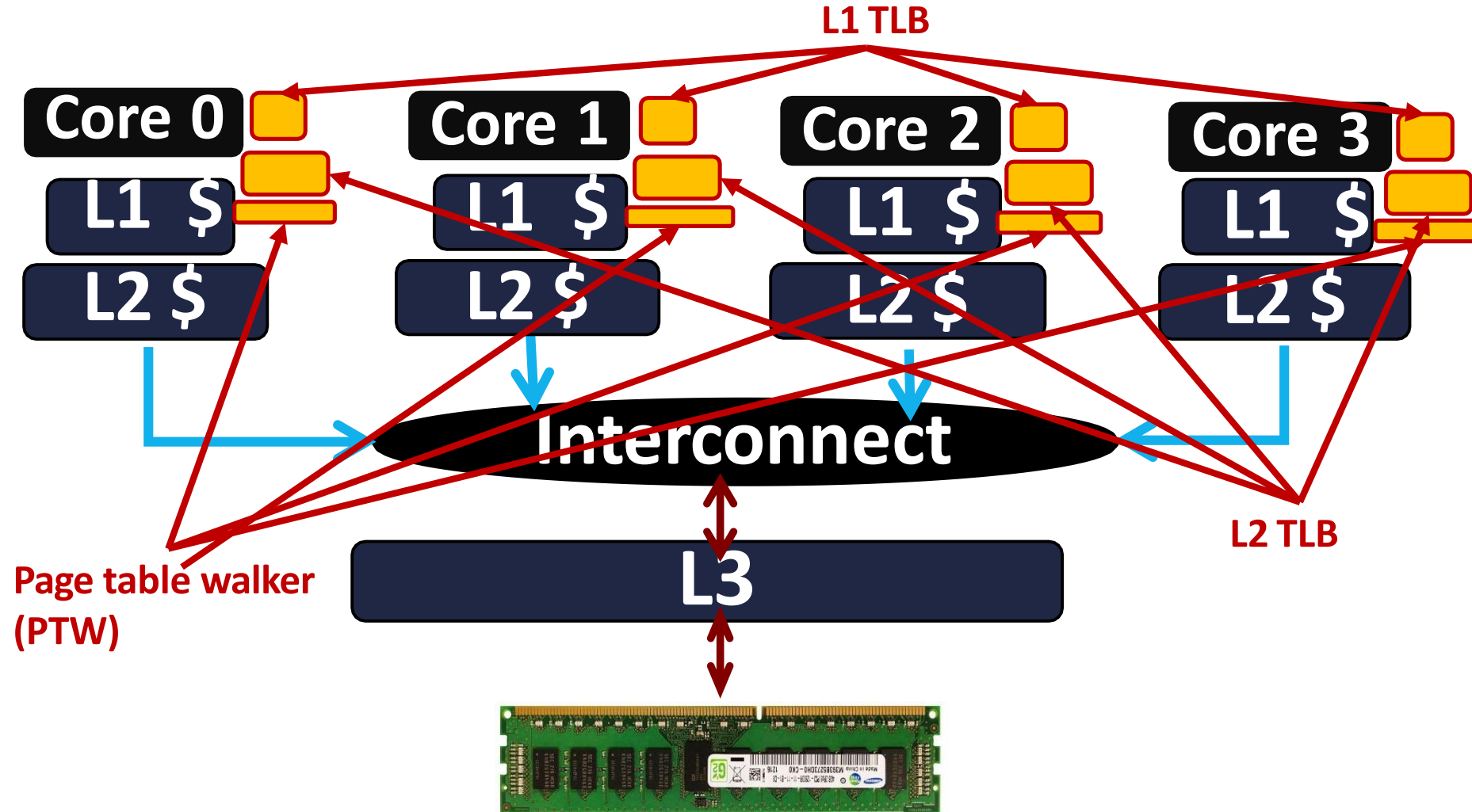
Page Table



Page Table Walk



The Memory Hierarchy



Misses and latencies

**TRANSLATION
MISS
(TLB)**

5 DRAM accesses in worst case

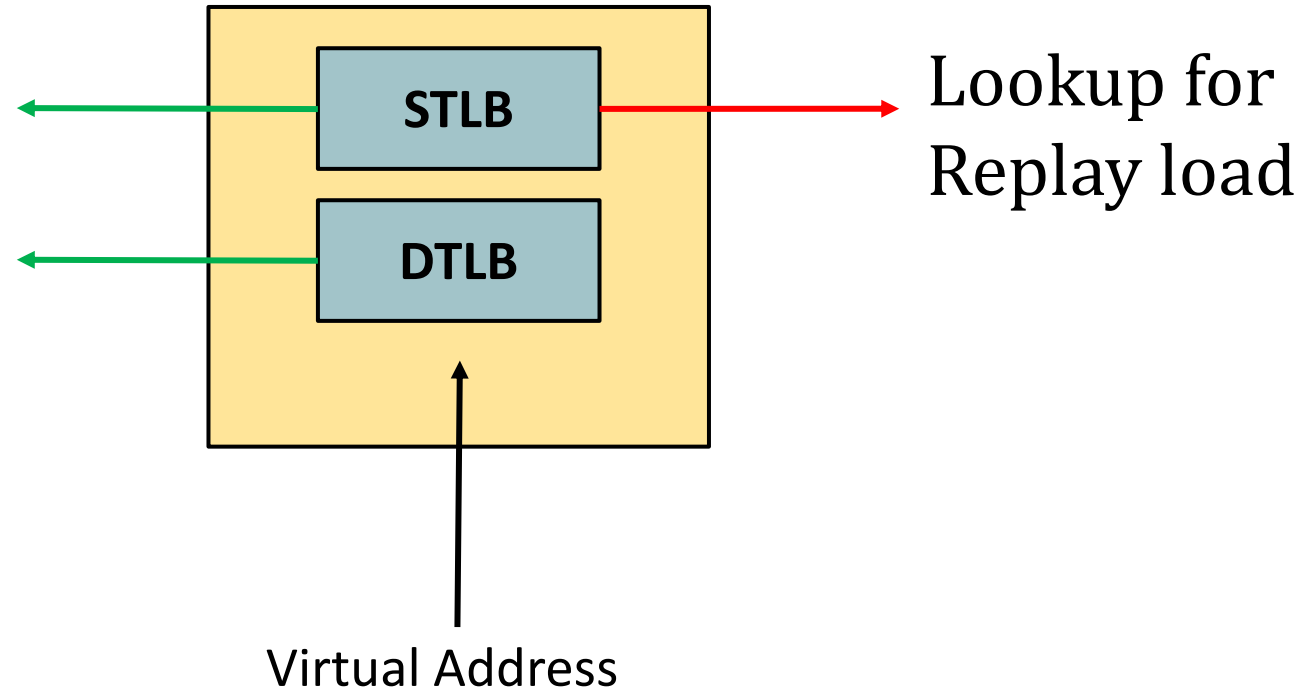
**DATA MISS
(CACHE)**

1 DRAM access in worst case

What happens if we have both?... 6 DRAM accesses

New terms

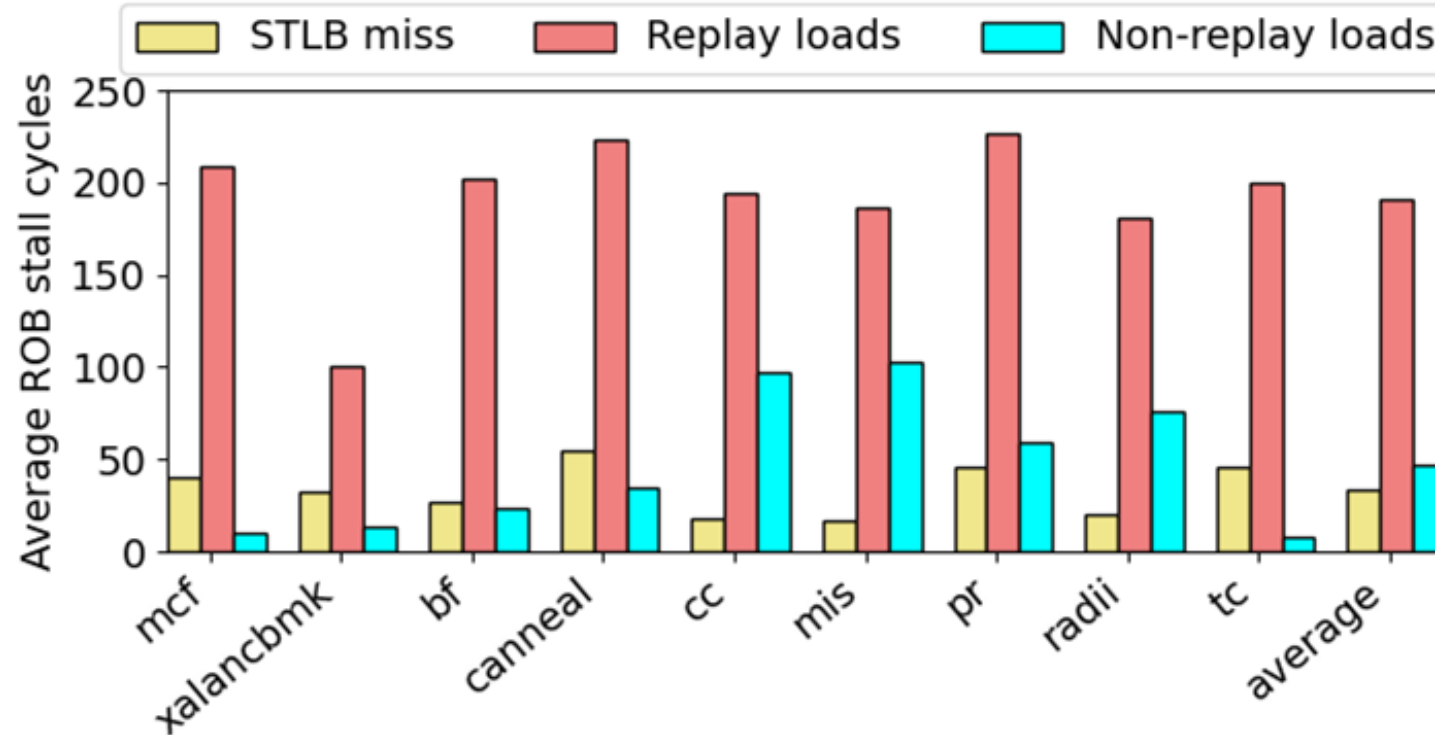
Lookup for Non-
Replay load



Hit

Miss

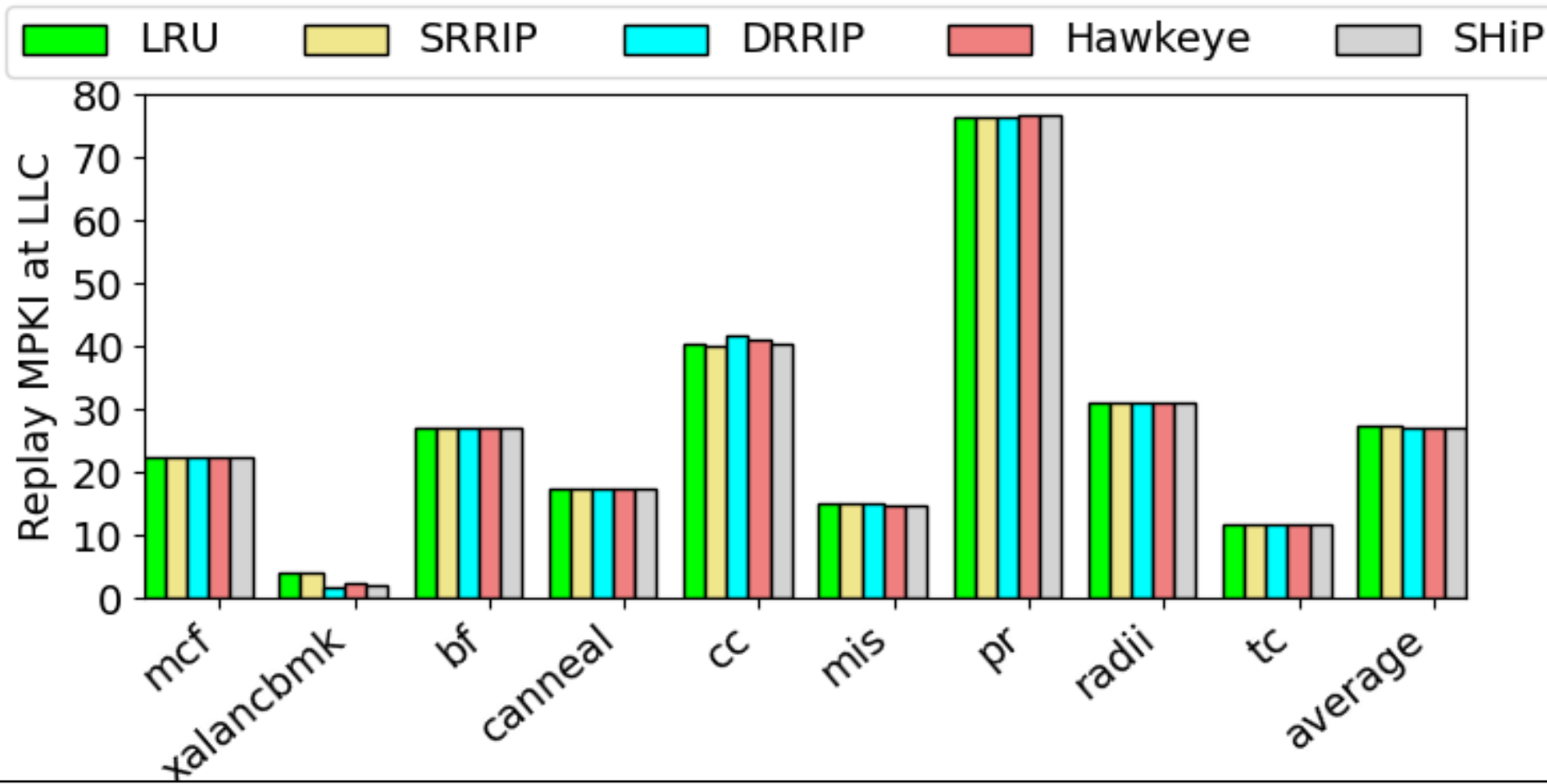
Processor Stalls because of translations



Average ROB stall cycles due to STLB miss is 33,
replay is 191 and remaining loads is 47.

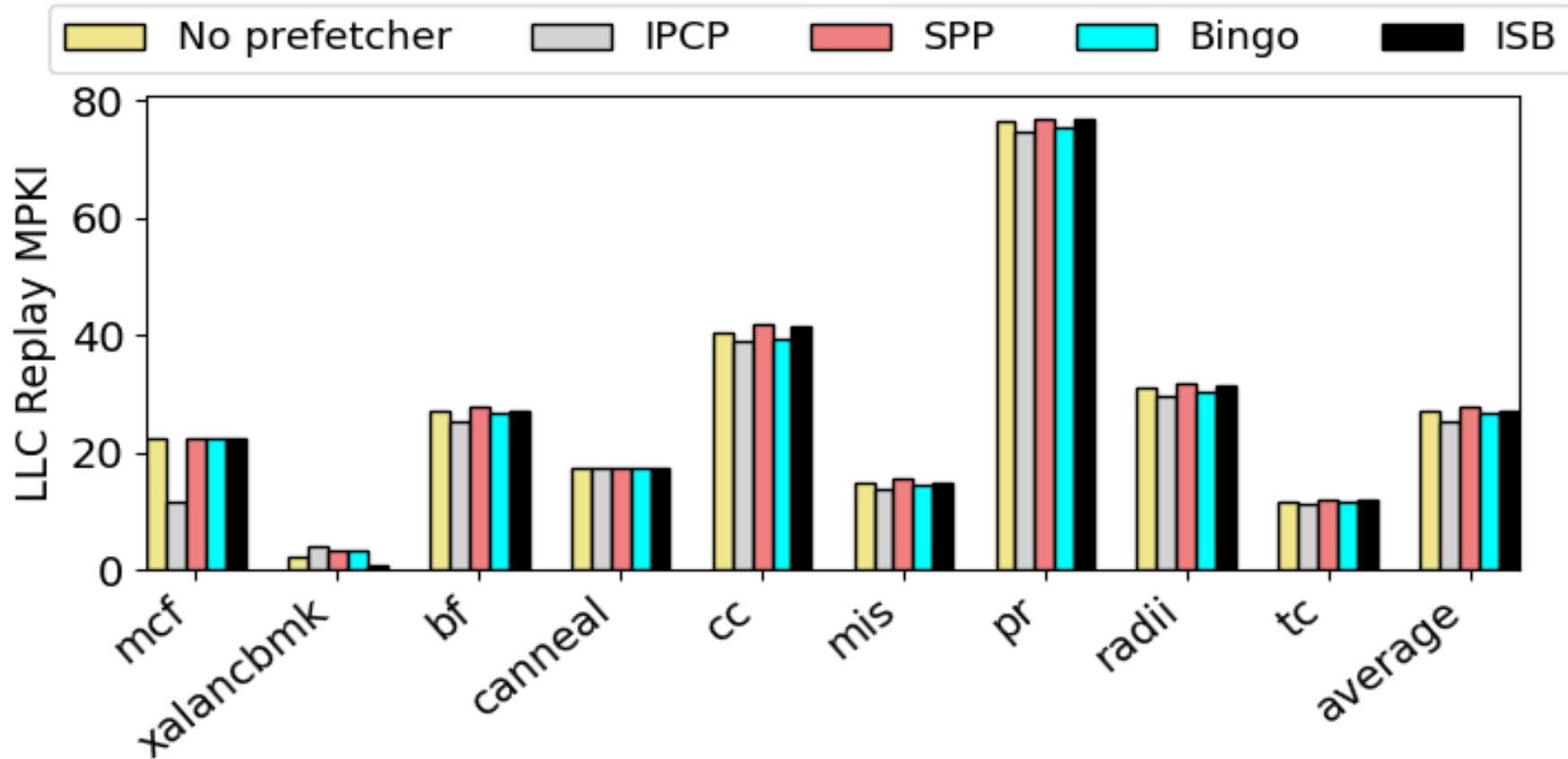
If an OS page is cold then data will be **even cooler**

How cache management policies react to Replays



Current cache management policies fail to reduce the replay misses at LLC

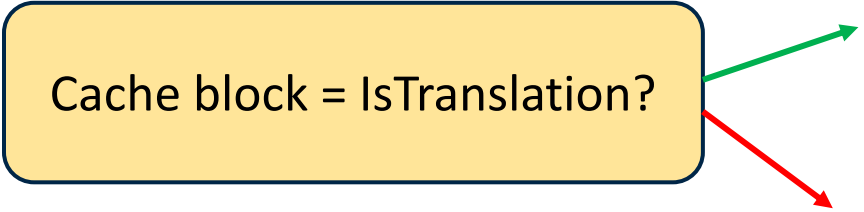
Will data prefetchers work for replay?



State-of-the-art data prefetchers also fail to
reduce the replay misses

Enhancement-I: Treat translations differently

Cache block = IsTranslation?



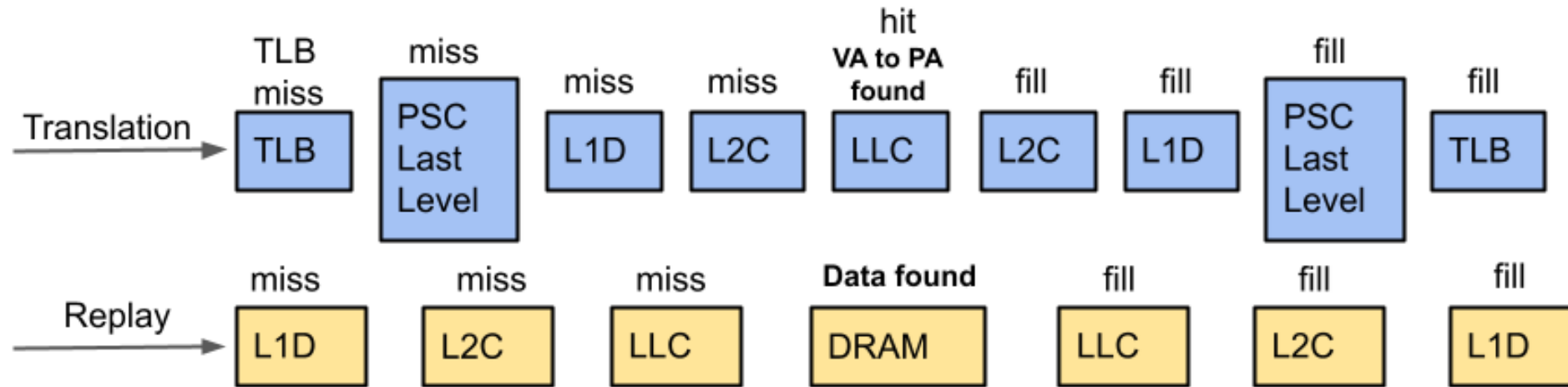
Keep them in cache hierarchy for a long time

Data blocks will come and go

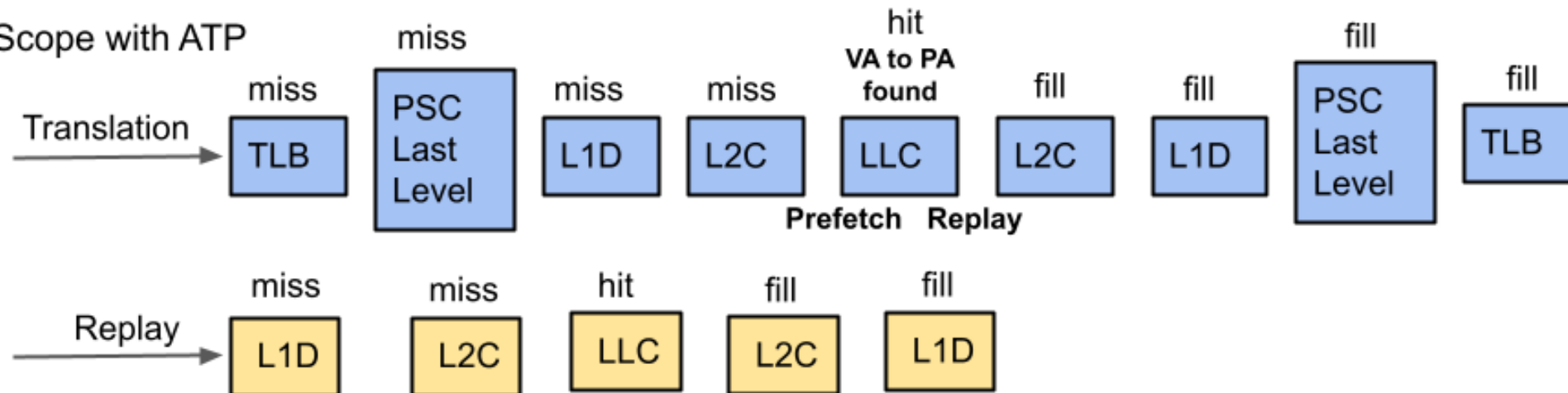
Yipee. 99% hit rate at the cache hierarchy for translations

Enhancement-II: Translation hit triggered Prefetcher

a. baseline



b. Scope with ATP



Takeaway message: Common Sense

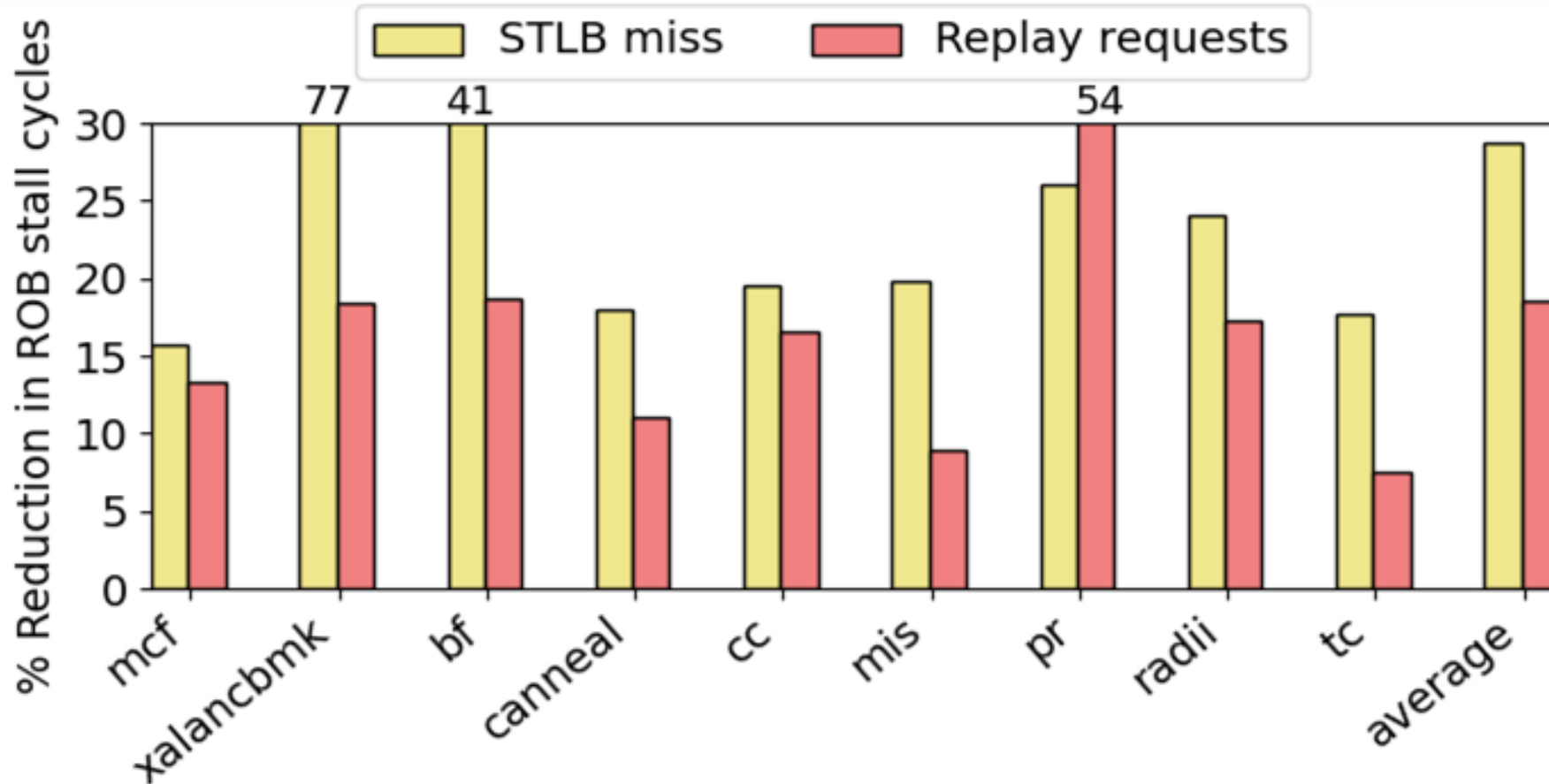


Common sense is not
so common.

Voltaire

“ quote fancy

Reduction in Processor Stalls



ROB stall cycles get reduced by 28.76% for translations and 18.5% for replay loads, leading to 4.8% performance improvement

Bouquet of microarchitecture ideas

Idea	Storage <i>(Lower)</i>	Performance <i>(Higher)</i>	Energy <i>(Lower)</i>
ISCA '20	1KB	45%	50%
CAL'21	+2.5KB	43%	45%
MICRO'22	2.5KB	48.5%	11%
ISPASS'22	+0.0KB 😊	+4.5%, data-intensive apps.	

Shh.. Microarchitects at work !

Bouquet of microarchitecture ideas



Microarchitecture research is fun and rewarding 😊

Stop listening to ...

What Next? Security, performance, both

Miles to go before I (we) sleep

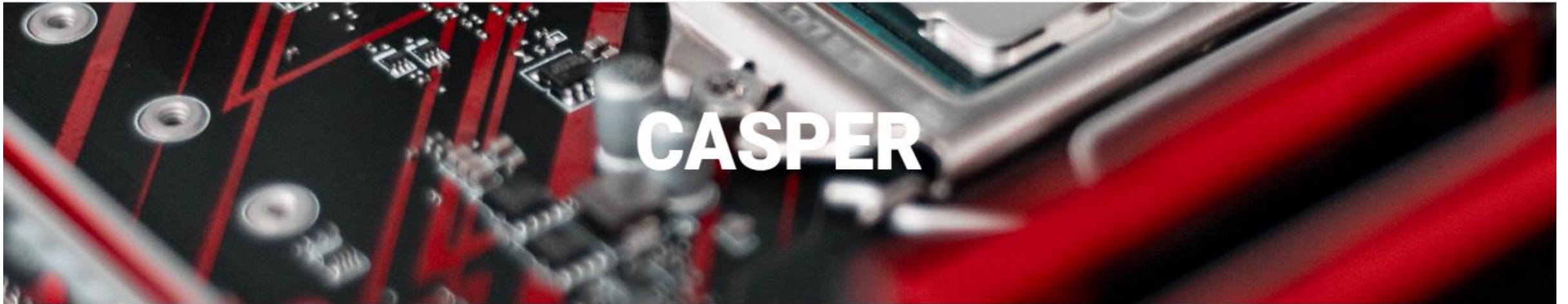


Microarchitects are on it 😊

CASPER@IITB



[About](#) [CASPERIANS](#) [Publications](#) [Blogs](#)



Welcome to CASPER: *Computer Architecture for Security and Performance Research* group!

If you like vada pav along with dosa, do consider joining 😊

In a nutshell

What do we do?

Dream the impossible and ask the right questions
to make it possible

My take from last five years:

*If you know the problem well then
you know the solution well too*

Microarch. Research – Test Match Cricket

"Play Life like a Cricket match.

Don't try to hit hard in every situation, Just keep rotating, moving, and then look for that one delivery and hit it as hard as you can."

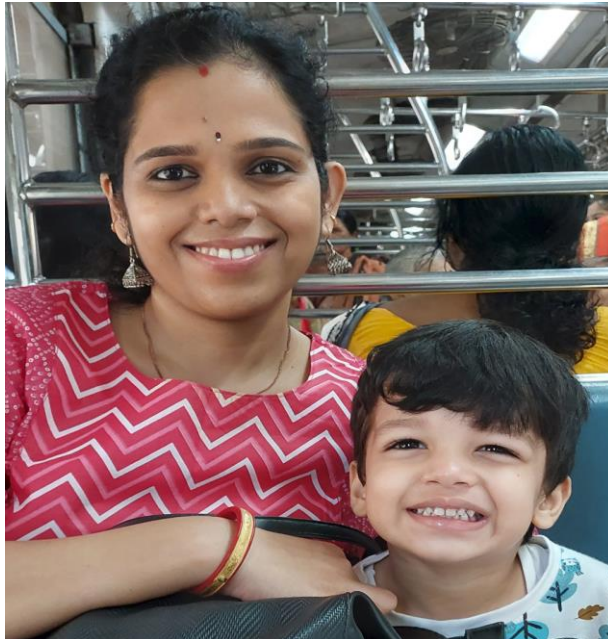
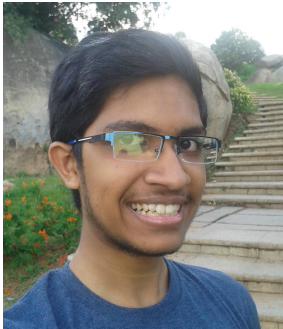
Dhanyavaad

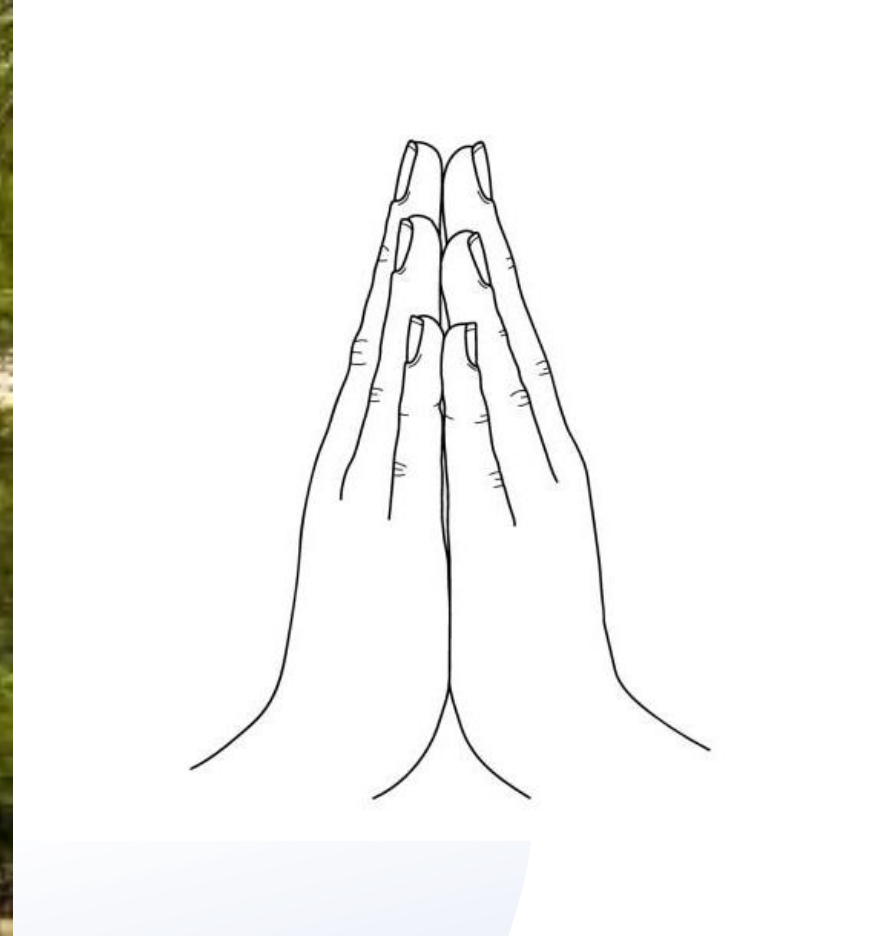


Qualcomm

Google Research

Work done remotely (COVID-19 😞)





Dhanyavaad CSE-IITM