Welcome!

The 1st 128-bit RISC-V European Workshop

HiPEAC Workshop
Wednesday January 22nd, 2025, Barcelona

The CfP is open!
Deadline Friday February 7th, 2025, AOE.
https://riscv-europe.org

Acknowledgments to
« **Maplurinum — Machinæ pluribus unum** »
(Faire) une seule machine avec plusieurs
(Make) one machine out of many
French gov. grant n° ANR-21-CE25-0016

The Benagil team of INRIA and Institut Polytechnique de Paris hires a tenured assistant professor (young researcher). This is a system and distributed systems group at the frontier between hardware and software.
Contact: gael.thomas@inria.fr.

The CfP is open! Check https://riscv-europe.org

Deadline Friday February 7th, 2025, AOE.

Maplurinum — One Machine out of Many, or We had 64 bit, yes. What about second 64 bit?

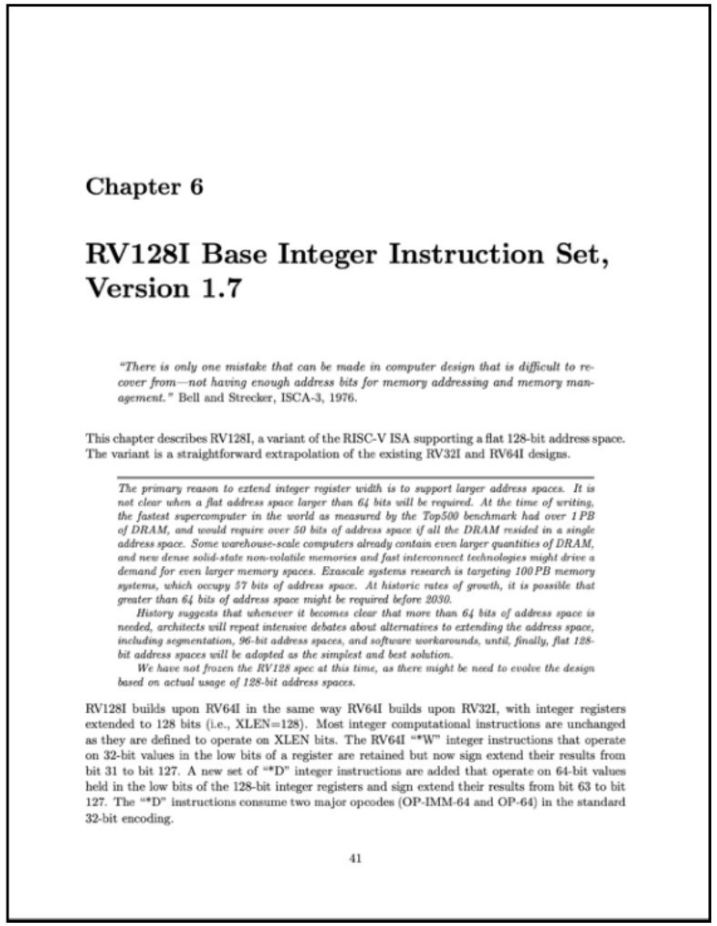
Mathieu Bacou[1], Adam Chader[1], Chandana Deshpande[2], Christian Fabre[3], César Fuguet[6], Pierre Michaud[4], Arthur Perais[2], Frédéric Pétrot[2], Gaël Thomas[5], Jana Toljaga[1], Eduardo Tomasi[2,3]

[1] Samovar, Télécom SudParis, IMT, IP Paris
[2] Université Grenoble Alpes, CNRS, Grenoble INP, TIMA
[3] Université Grenoble Alpes, CEA, List
[4] Inria, Université de Rennes, IRISA
[5] Inria Saclay
[6] Inria, Université Grenoble Alpes

French government grant ANR-21-CE25-0016
ANR project « Maplurinum — Machinæ pluribus unum » (Make) one machine out of many

"The 1st 128-bit RISC-V European Workshop", HiPEAC, Wednesday January 22nd, 2025, Barcelona.

What is RISC-V 128 bit anyway?



Chapter 6

RV128I Base Integer Instruction Set, Version 1.7

"There is only one mistake that can be made in computer design that is difficult to recover from—not having enough address bits for memory addressing and memory management." Bell and Strecker, ISCA-3, 1976.

This chapter describes RV128I, a variant of the RISC-V ISA supporting a flat 128-bit address space. The variant is a straightforward extrapolation of the existing RV32I and RV64I designs.

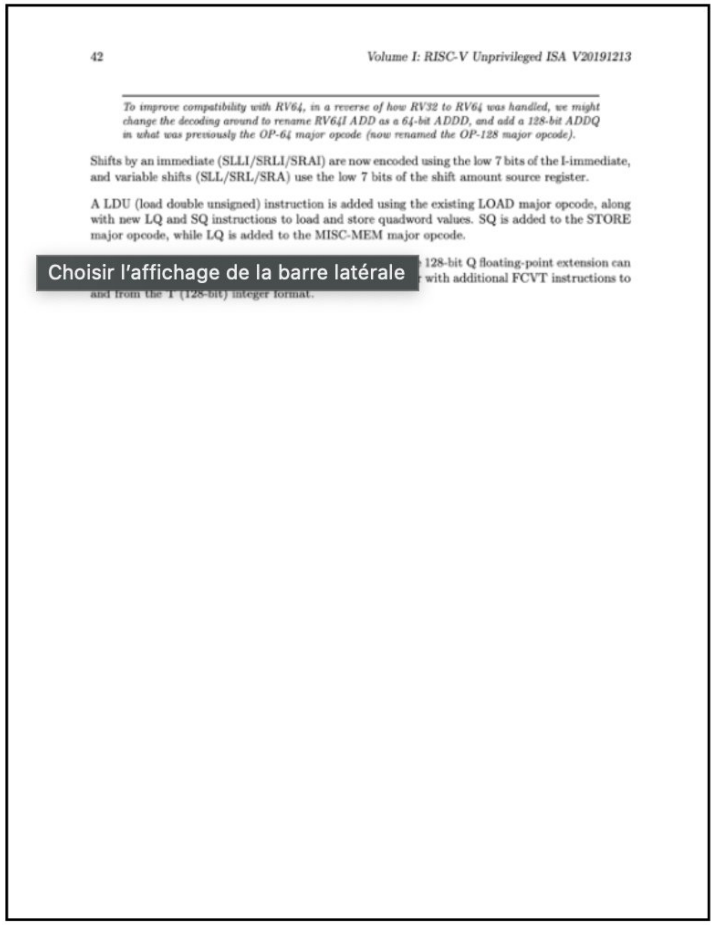
The primary reason to extend integer register width is to support larger address spaces. It is not clear when a flat address space larger than 64 bits will be required. At the time of writing, the fastest supercomputer in the world as measured by the Top500 benchmark had over 1 PB of DRAM, and would require over 50 bits of address space if all the DRAM resided in a single address space. Some warehouse-scale computers already contain even larger quantities of DRAM, and new denser solid-state non-volatile memories and fast interconnect technologies might drive a demand for even larger memory spaces. Exascale systems research is targeting 100 PB memory systems, which occupy 57 bits of address space. At historic rates of growth, it is possible that greater than 64 bits of address space might be required before 2030.

History suggests that whenever it becomes clear that more than 64 bits of address space is needed, architects will repeat intensive debates about alternatives to extending the address space, including segmentation, 96-bit address spaces, and software workarounds, until, finally, flat 128-bit address spaces will be adopted as the simplest and best solution.

We have not frozen the RV128 spec at this time, as there might be need to evolve the design based on actual usage of 128-bit address spaces.

RV128I builds upon RV64I in the same way RV64I builds upon RV32I, with integer registers extended to 128 bits (i.e., XLEN=128). Most integer computational instructions are unchanged as they are defined to operate on XLEN bits. The RV64I "*W" integer instructions that operate on 32-bit values in the low bits of a register are retained but now sign extend their results from bit 31 to bit 127. A new set of "*D" integer instructions are added that operate on 64-bit values held in the low bits of the 128-bit integer registers and sign extend their results from bit 63 to bit 127. The "*D" instructions consume two major opcodes (OP-IMM-64 and OP-64) in the standard 32-bit encoding.

41



To improve compatibility with RV64, in a reverse of how RV32 to RV64 was handled, we might change the decoding around to rename RV64I ADD as a 64-bit ADDD, and add a 128-bit ADDQ in what was previously the OP-64 major opcode (now renamed the OP-128 major opcode).

Shifts by an immediate (SLLI/SRLI/SRAI) are now encoded using the low 7 bits of the I-immediate, and variable shifts (SLL/SRL/SRA) use the low 7 bits of the shift amount source register.

A LDU (load double unsigned) instruction is added using the existing LOAD major opcode, along with new LQ and SQ instructions to load and store quadword values. SQ is added to the STORE major opcode, while LQ is added to the MISC-MEM major opcode.

Choisir l'affichage de la barre latérale

128-bit Q floating-point extension can with additional FCVT instructions to and from the T (128-bit) integer format.

Starts with a nice quote :
"There is only one mistake that can be made in computer design that is difficult to recover from—not having enough address bits for memory addressing and memory management."
Bell and Strecker, ISCA-3, 1976."

[1] A. Waterman and K. Asanović, "*Chapter 6, RV128I Base Integer Instruction Set, Version 1.7*," in The RISC-V Instruction Set Manual - Volume I: Unpriviliged ISA, 20191213, The RISC-V Foundation, 2019. Available online at **`https://riscv.org/technical/specifications/`**

This is not needed anytime soon! Why start so early?

DOI:10.1145/3282307

Innovations like domain-specific hardware, enhanced security, open instruction sets, and agile chip development will lead the way.

BY JOHN L. HENNESSY AND DAVID A. PATTERSON

A New Golden Age for Computer Architecture

WE BEGAN OUR Turing Lecture June 4, 2018[11] with a review of computer architecture since the 1960s. In addition to that review, here, we highlight current challenges and identify future opportunities, projecting another golden age for the field of computer architecture in the next decade, much like the 1980s when we did the research that led to our award, delivering gains in cost, energy, and security, as well as performance.

"Those who cannot remember the past are condemned to repeat it." —George Santayana, 1905

Software talks to hardware through a vocabulary called an instruction set architecture (ISA). By the early 1960s, IBM had four incompatible lines of computers, each with its own ISA, software stack, I/O system, and market niche—targeting small business, large business, scientific, and real time, respectively. IBM engineers, including ACM A.M. Turing Award laureate Fred Brooks, Jr., thought they could create a single ISA that would efficiently unify all four of these ISA bases.

They needed a technical solution for how computers as inexpensive as

» key insights

- Software advances can inspire architecture innovation.
- Elevating the hardware/software interface creates opportunities for architecture innovation.
- The marketplace ultimately settles architecture debates.

J. L Hennessy
& D. A. Patterson
Turing Award2018

https://iscaconf.org/isca2018/turing_lecture.html

"People who are serious about software should make their own hardware"

Alan Kay
Turing Award 2003
Apple Fellow

This presentation as a three courses light meal

Some ideas about the 128 bit software stack

Back to basics!

Single/Unified view of a large 128 bit machine

1. Operating system
2. Compilation chain

Architecture & Micro-architecture

Just double everything?

(partly) address complexity with:

1. Clustering
2. Compression

HW/SW Interface & Memory Management

How could the system architecture provide a 128 bit flat address space ?

1. Better tailor the virtual memory system to HPC systems
2. Unified 128 bit address space

Some ideas about the base software stack for future RISC-V 128 bit HPC machines with > 100 M cores

Christian FABRE (CEA LIST, Grenoble)

Let's explain "Some ideas about the base software stack for future RISC-V 128 bit HPC machine with > 100 M cores"

- Some ideas about ...
  - This is barely the beginning... We are not there yet — far from it!
- ... the base software stack ...
  - The base software stack:
    - Operating system: kernel, command and libraries. Forget about virtualization and other bleeding edge topics for a moment.
    - Compilers: code parallelization, code generation, runtime support.
    - File system: who need directories, files and data serialization, when you can have permanent pointers to data in byte addressable NV-RAM?
  - Long story short: unroll/remove the multiple software layers that have been coalescing over decades, like OpenMP or MPI.
- ... future ... HPC machines ...
  - Not there before a decade or more.
  - HPC are *closed* systems with *simple* heavy workloads. So it is easier analyze and understand, to kick start building something.
- ... with > 100 M cores.
  - We already have machines with 2-10 M cores in the Top 500.
- ... RISC-V ...
  - Machine will be made of RISC-V only cores. 100 k to 1 M interconnected clusters made of 100-1000 RISC-V cores each.
  - RISC-V as a unifying force: different clusters will support different sets of RISC-V extensions.
- ... 128 bit ...
  - Such a large address space is a chance to get a *Single System Image (SSI)* view of a program itself (process view) and the operating system (the OS as the first abstraction of the machine hardware)

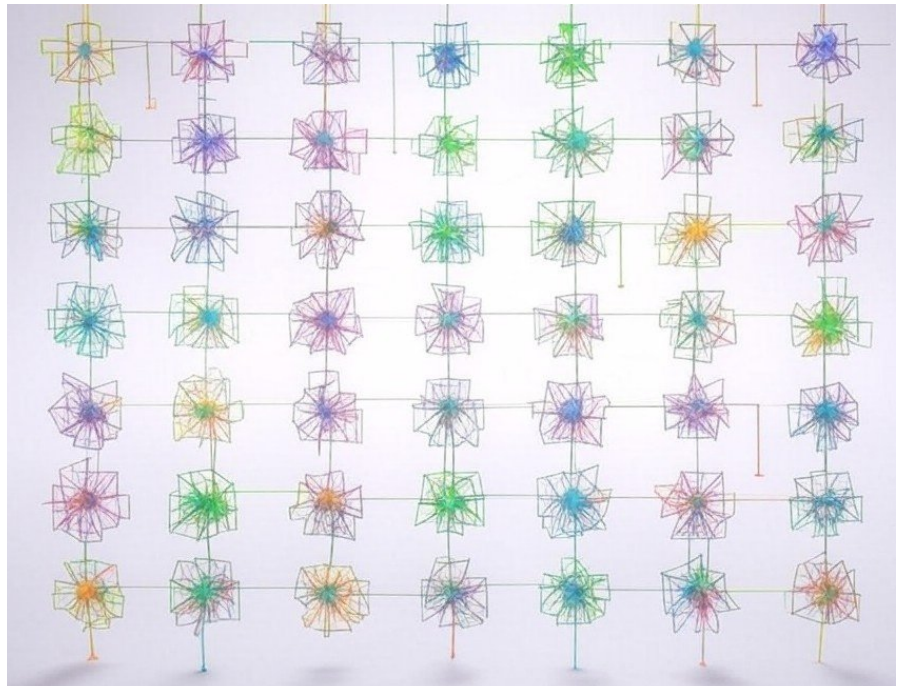
- The 128 tons elephant *not* in the room
  - Compute-intensive ISA extensions: crypto., vector, variable precision, matrices, etc.
  - The ecosystem is there already, alive and kicking — both for basic software and hardware

128 bit makes it for a large address space!

“Such a large address space is a chance to get a *Single System Image (SSI)* view of a program itself (process view) and the operating system (the OS as the first abstraction of the machine hardware)”

Opportunities to revisit the software stack and its basic concepts:

- Get rid of MPI? Replaced by cluster to cluster *virtual memory remapping* and memory transfers
- What is exactly a 128 bit *process* that would span 1 M shared memory clusters?
- What would be a *kernel* for such a machine?
- Ensure *virtual-to-physical memory mapping consistency* over that many cores
- Will we *still* be programming 100 cores with C code + pragmas?
- Why bother with a file system when you can store your data in non volatile memory, and get a permanent pointer to it ?



Generated by Grok2 on a request for a diagram of clusters connected by a high speed network

Microarchitectural tricks to support 128-bit RISC-V without “128-bit everywhere”

Arthur PERAIS (CNRS)

Opportunities and challenges: Microarchitecture

- A naive approach to 128-bit hardware risks introducing area/power/latency penalties

- Larger tags/payloads
  - TLBs, branch target predictors (BTB & Co) => Compress tags[1]

- Wider datapath
  - Physical registers
  - Operand bypass
  - Functional units

- “HW Tax” to 128-bit support

[1]Seznec, “Don't use the page number, but a pointer to it”, ISCA’96

Opportunities and challenges: Microarchitecture

- A simple example to illustrate opportunity to reduce the “HW tax”

```
int main()
{
  for(int i = 0; i < 64; i++)
  {
    c[i] = a[i] + b[i] * 10;
  }
}
```

gcc 13 –O1

```
loop:
  lw      a1,0(a3)     // *b
  slliw   a5,a1,2      // *b * 4
  addw    a5,a5,a1     // *b * 5
  slliw   a5,a5,1      // *b * 10
  lw      a1,0(a4)     // *a
  addw    a5,a5,a1     // *a + *b * 10
  sw      a5,0(a2)     // *c = ...
  addi    a4,a4,4      // a++
  addi    a3,a3,4      // b++
  addi    a2,a2,4      // c++
  addi    a6,a6,1      // i++
  bne     a6,a0, loop
```

What actually needs to use 128-bit?

Opportunities and challenges: Microarchitecture

- A simple example to illustrate opportunity to reduce the “HW tax”

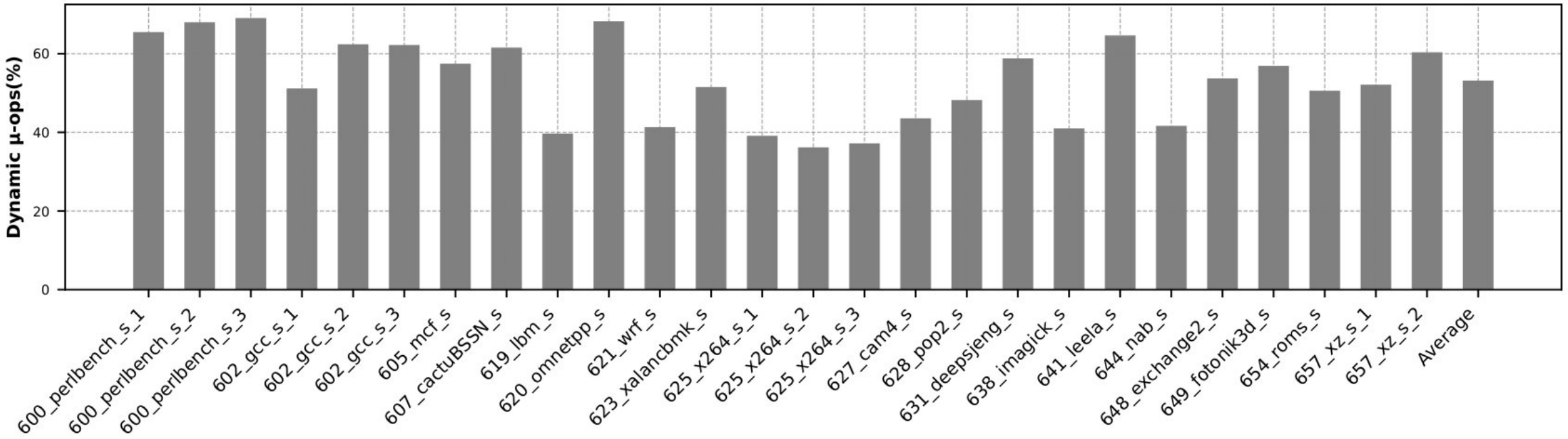
```
int main()
{
  for(int i = 0; i < 64; i++)
  {
    c[i] = a[i] + b[i] * 10;
  }
}
```

gcc 13 –O1

```
loop:
  lw     a1,0(a3)     // *b
  slliw  a5,a1,2      // *b * 4
  addw   a5,a5,a1     // *b * 5
  slliw  a5,a5,1      // *b * 10
  lw     a1,0(a4)     // *a
  addw   a5,a5,a1     // *a + *b * 10
  sw     a5,0(a2)     // *c = ...
  addi   a4,a4,4      // a++
  addi   a3,a3,4      // b++
  addi   a2,a2,4      // c++
  addi   a6,a6,1      // i++
  bne    a6,a0, loop
```

Address generation (in blue)!

Opportunities and challenges: Microarchitecture



- Fraction of retired uOps that participate in AGEN (SPEC 2k17) is around 55% on average.
- => Opportunity to save “something” for 45% of the retired uOps

Opportunities and challenges: Microarchitecture

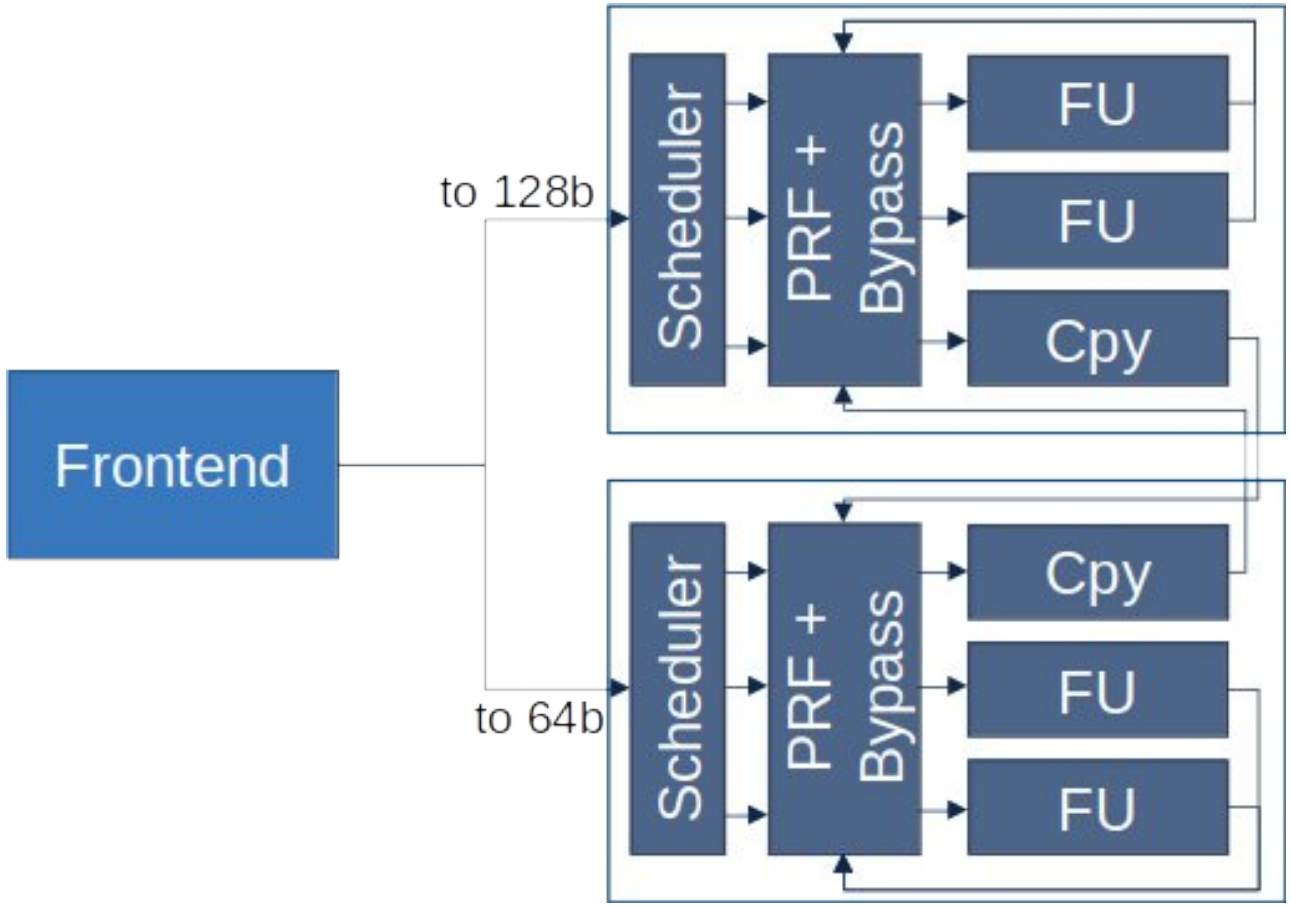
- Assume only address generation actually requires 128-bit
- Implement a 128-bit block to deal with AGEN and occasional 128-bit compute
  - Keep a 64-bit block to deal with the rest
- Works only if 64/128 is reasonably balanced
  - It is :)
    - in SPEC :|

Opportunities and challenges: Microarchitecture

- ADdress/Value (ADA) clustered microarchitecture[1]
  - Each cluster has their own resources

- Address cluster: 128b datapath
- Data cluster: 64b datapath

- “Dense” local bypass
- “Shallow” global bypass using explicit copy uOps

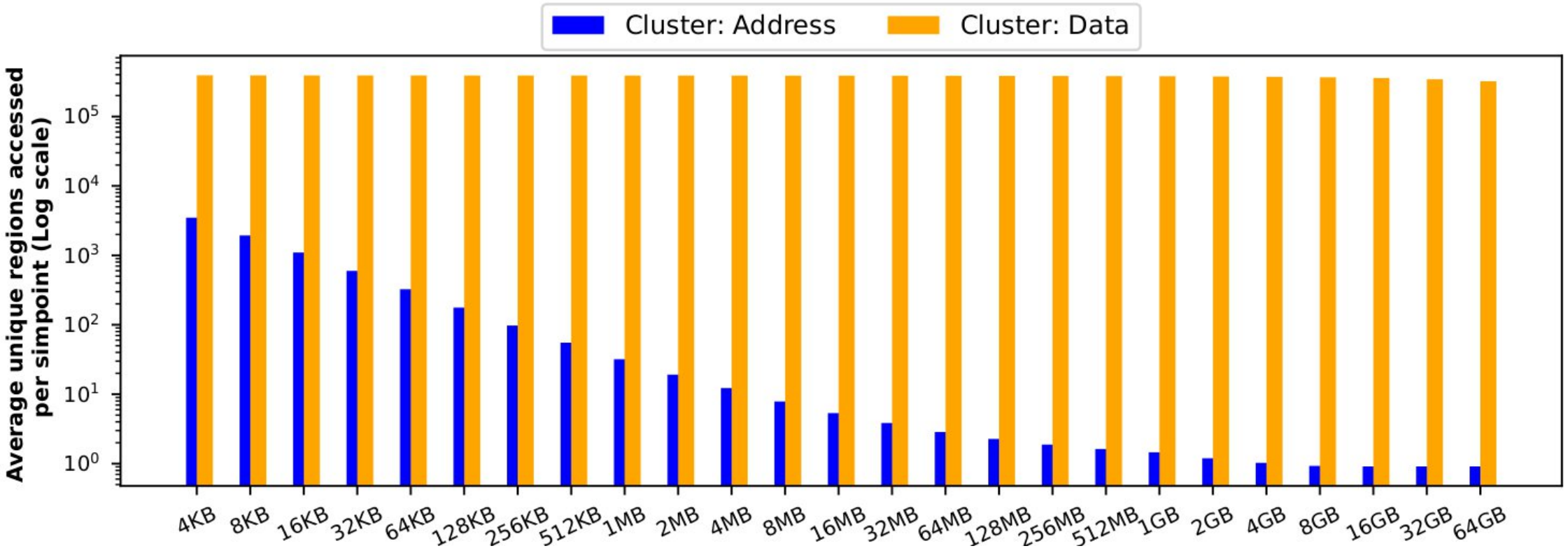


You mave have caught this on Chandana Desphande’s poster at a coffee break

[1]Canal et al., “Dynamic cluster assignment mechanisms.”, HPCA’00

Opportunities and challenges: Microarchitecture

- One optimization enabled by separating Addresses and Data: PRF Compression

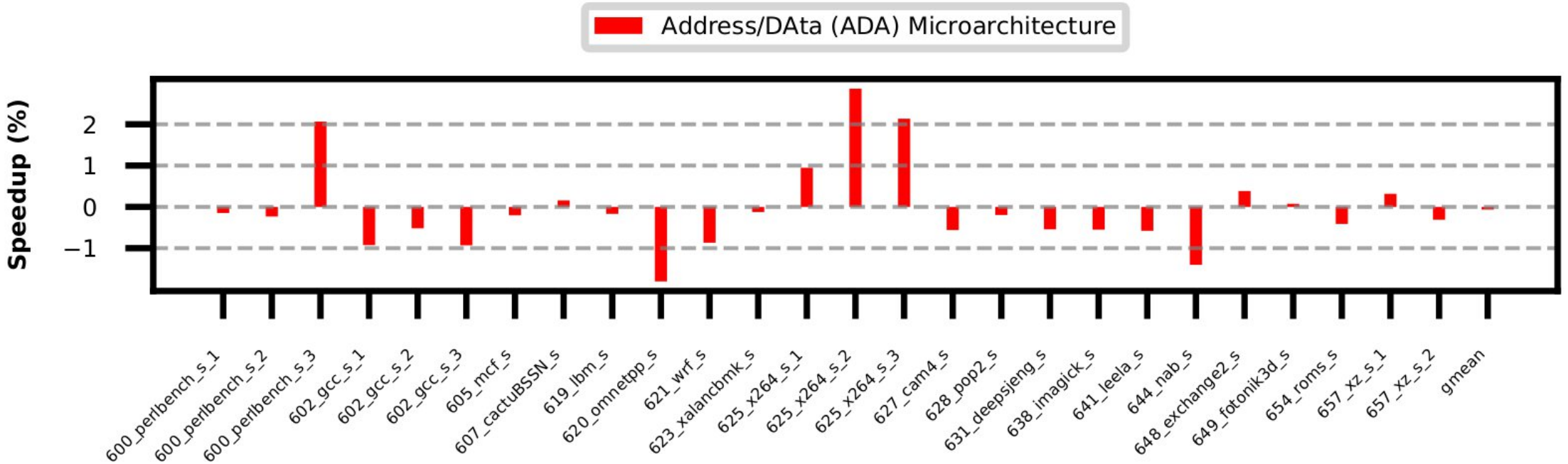


- Address cluster PRF has few different upper bits, achieve 40% storage reduction with region-based compression[1]

[1]Seznec, “Don't use the page number, but a pointer to it”, ISCA’96

Opportunities and challenges: Microarchitecture

- Performance on-par with monolithic 128-bit uarch, but significant savings (fewer operand ready broadcasts, less bypass, smaller PRFs)



Hardware-Software Interface and SoC Integration to efficiently support 128 bit address spaces

César FUGUET (Inria)

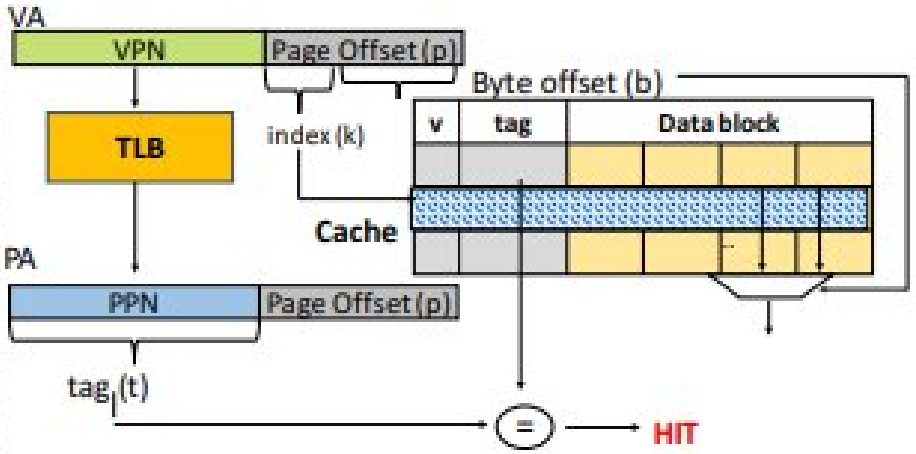
Scope

- This work focuses on High-Performance Computing (HPC) systems
- Current trend indicates that memory requirements in such systems may exceed 2^{64} bytes in 20 years.
- The transition to 128 bits addresses is an opportunity to review some old well-established architecture mechanisms
  - Virtual addressing
  - Data addressing and orchestration on distributed machines

Virtual Addressing: Page Size

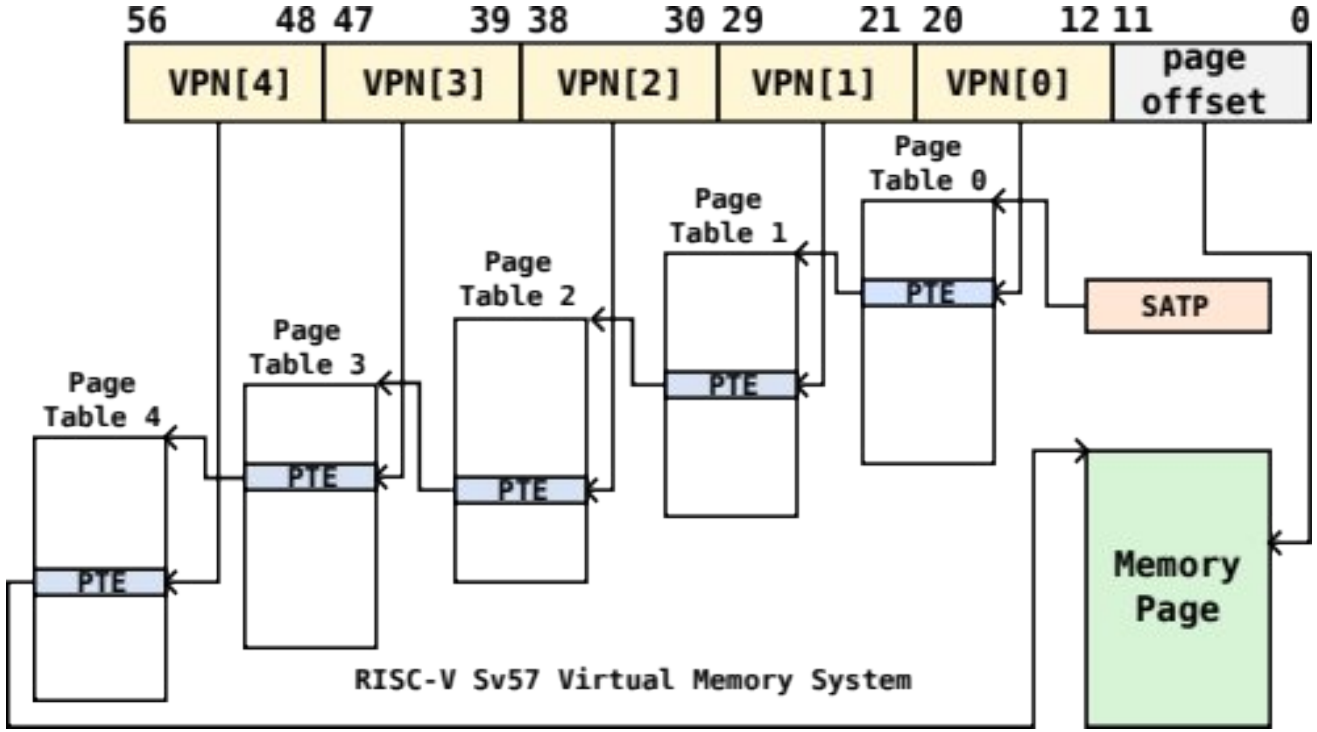
It is time to review the long-lasting 4K page size in processors

Address translation can be a performance bottleneck because of TLB misses and long page table walks



VIPT Constraint: k+b ≤ p

(b) Virtually-Indexed Physically Tagged (VIPT):
Parallel TLB and Cache Access



Page-Size Exploration: Methodology

- We conducted a study to see the impact of page size on:
  - performance (measured as TLB miss rate)
  - memory bloat (ratio of memory effectively used to memory allocated)
- The study considered different benchmarks: NPB, PARSEC, SPLASH3, SPECInt
- We used the QEMU simulator to redirect execution traces to a TLB simulator

E. Tomasi, C. Fuguet, C. Fabre, F. Pétrot, "Page size exploration for RISC-V systems: the case for HPC", 35th International Workshop on Rapid System Prototyping

Page-Size Exploration: Cost

- We defined a simple cost function: A weighted mean of the performance and memory bloat as a function of the page size.

$$J_{n,b}(p) = w \cdot mr_{n,b}(p) + (1 - w) \cdot mb_{n,b}(p)$$

$mr_{n,b}$ TLB miss rate for benchmark b and n cores

$mb_{n,b}$ Memory bloat for benchmark b and n cores

- The weight 0 < w < 1 throttle the importance of the one or the other criterion

```
          W <  0.4
0.4 <= W <= 0.6
          W >  0.6
```

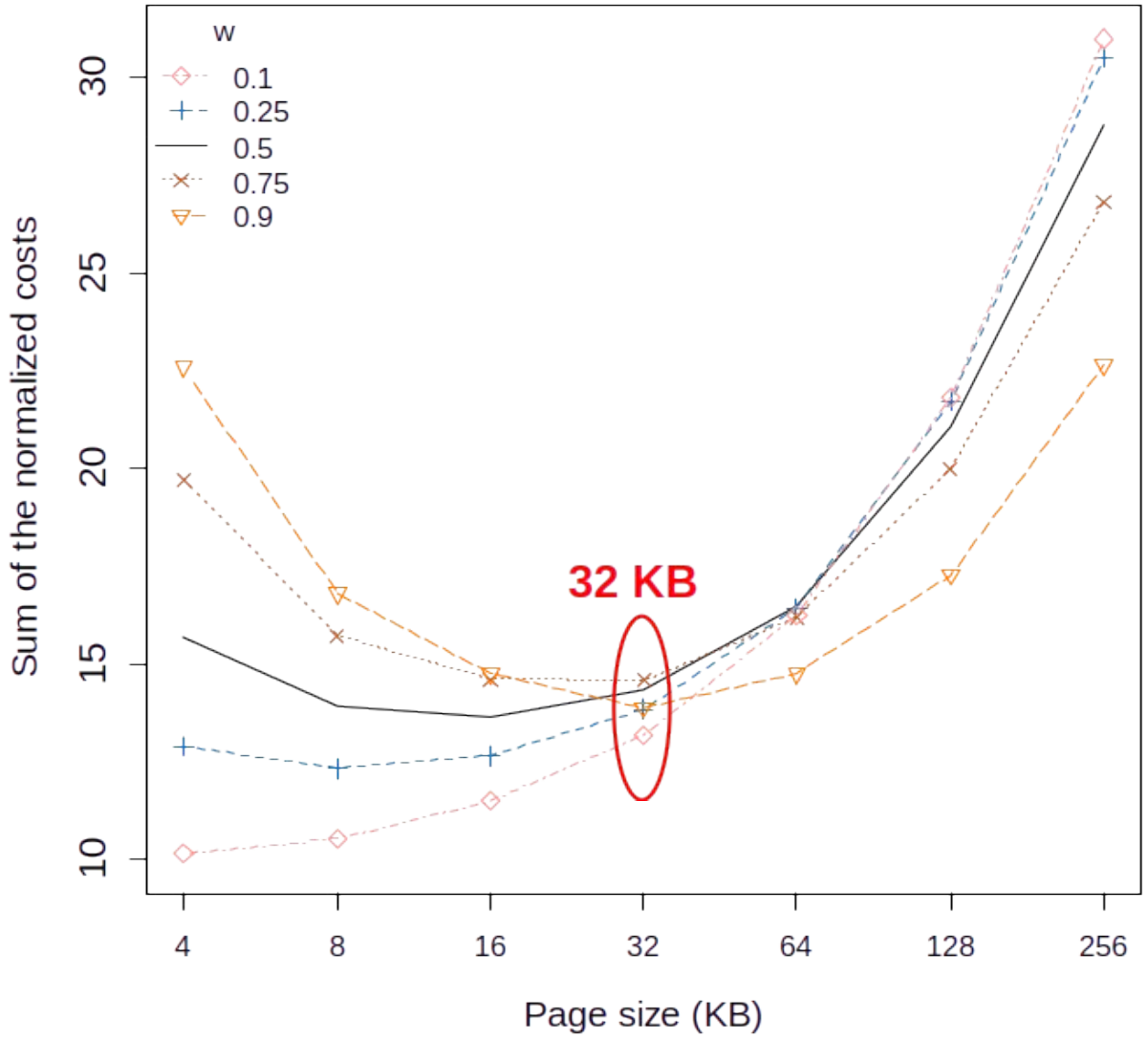
Embedded Systems

General Purpose Systems

High-Performance Computing systems

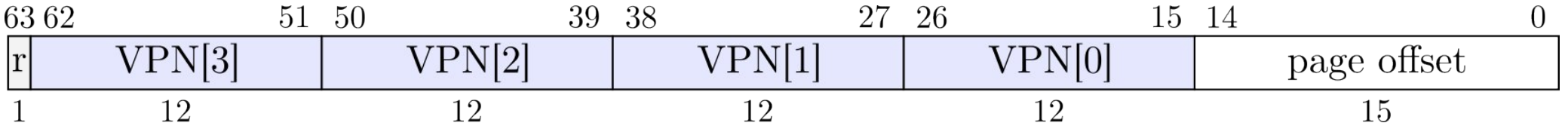
Page-Size Exploration: Results

- For both general purpose systems and HPC systems, a page size of 32 KB fits the best.
- For embedded systems, a smaller page size (e.g. 16 KB) gives the best result.

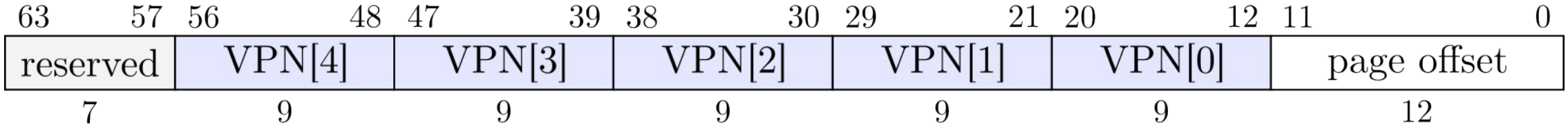


Page-Size Exploration: Conclusions

- Preliminary results indicate that a transition to 32 KB pages would be beneficial for future HPC systems
- Increasing the page size reduces TLB misses and allows shallower page table “walks” (hence reduces TLB miss penalty)



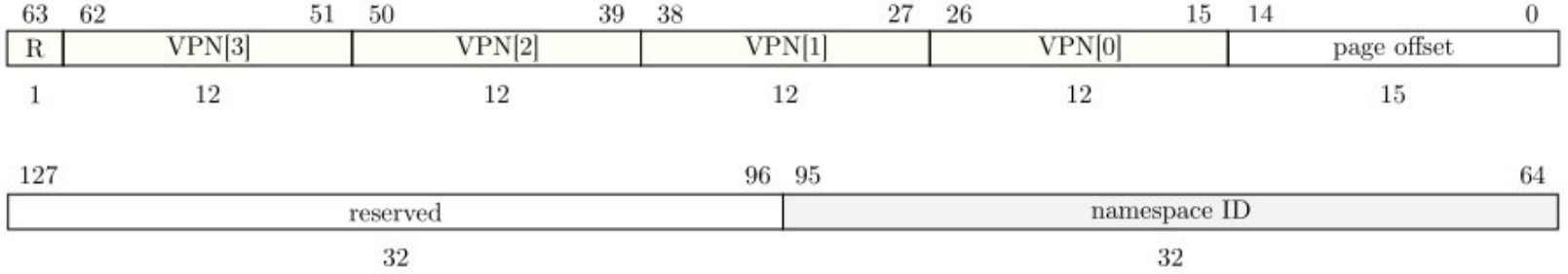
Proposed Sv63 addressing scheme



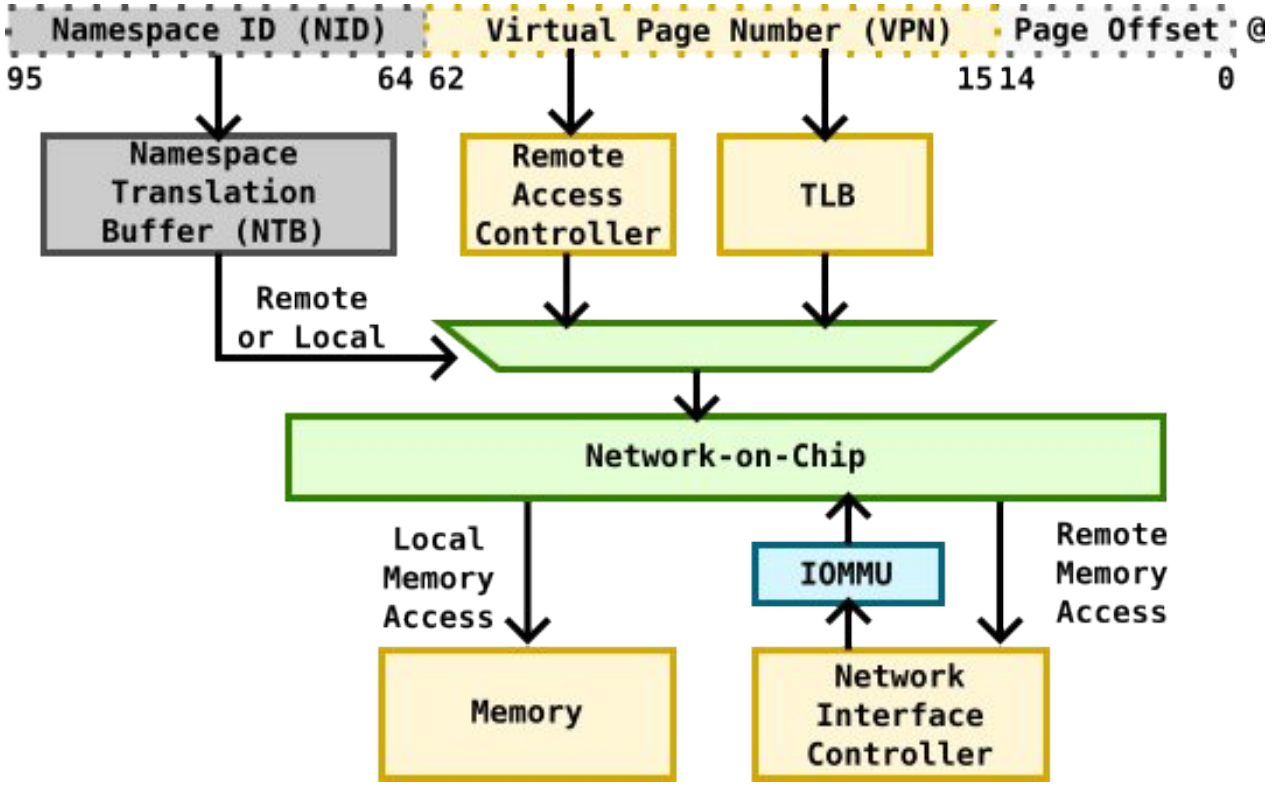
Current standard Sv57 addressing scheme

But what about 128-bits ???

Work in progress



- Our target: Flat address space for distributed machines (nodes)
- Namespace ID : virtual identifier of a node
- Local accesses are translated as usual by the local TLB
- Remote accesses can be forwarded to:
  - the remote node through the NIC
  - a local copy of the remote page.
- Remote NIC performs virtual to physical translation using an IOMMU



Conclusion & Questions

1. 128-bit pointers is more than “just more memory”
   - A thought experiment to rethink part of the stack
2. RISC-V 128 bit provides a tangible opportunity for hardware – software co-design
   - Some thoughts of the basic software stack
   - Some thoughts on the system
   - Some thoughts on micro-architecture
   - Some thoughts on virtual memory
   - Also, some thoughts on using 128-bit addresses differently: 2D addressing machines (https://inria.hal.science/hal-04816363v1, P. Michaud)
- Though 128 bit machines are probably far away, such work will take time, so we are starting now!

The CfP is open! Check https://riscv-europe.org

Deadline Friday February 7th, 2025, AOE.

The Benagil team of INRIA and Institut Polytechnique de Paris hires a tenured assistant professor (young researcher). This is a system and distributed systems group at the frontier between hardware and software. Contact: gael.thomas@inria.fr.

Work funded by the project « **Maplurinum — Machinæ pluribus unum** »
(Faire) une seule machine avec plusieurs (Make) one machine out of many

French gov. grant n° ANR-21-CE25-0016

Extra slides

Elephant in the Room : 128-bit ?

- The 128 tons elephant *not* in the room
  - Compute-intensive ISA extensions: crypto., vector, variable precision, matrices, etc.
  - The ecosystem is there already, alive and kicking — both for basic software and hardware

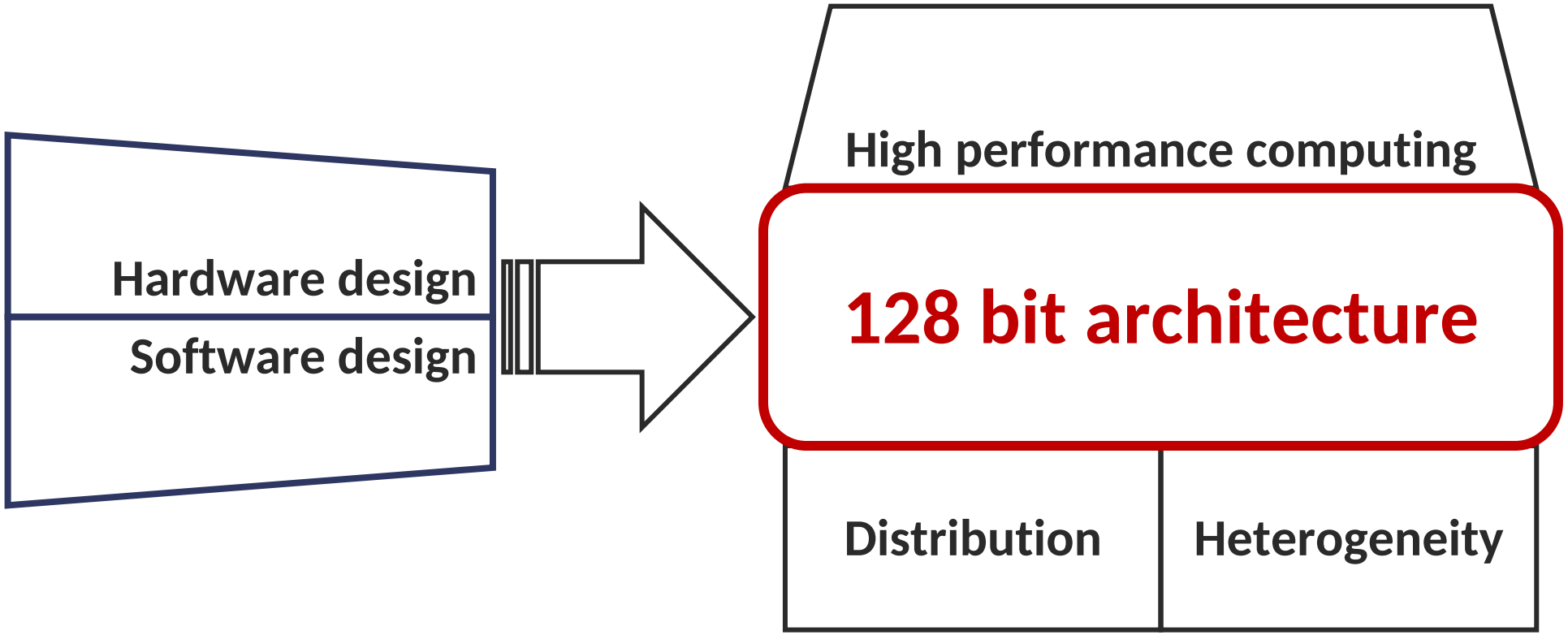
- For which machines?
  - 128-bit within a single socket or server is unlikely :
    - Rack-scale computing
    - Supercomputer-scale
  - Niche markets: filtering/fire-walling 128 bit addresses of IPv6

- But this is also an opportunity to revisit the software stack
  - File system: who need directories, files and data serialization, when you can have permanent pointers to data in byte addressable NV-RAM across the whole machine (machine to be defined :))?
    - Similarly: Share pointers to in-memory objects in the VA space with other sockets, blades, etc.?
  - Such a large address space is a chance to get a *Single System Image (SSI)* view of a program itself (process view) and the operating system (the OS as the first abstraction of the machine hardware)

Some ideas about the base software stack for future RISC-V 128 bit HPC machines with > 100 M cores

Christian FABRE (CEA LIST, Grenoble)

Overview


Hardware design
Software design
High performance computing
128 bit architecture
Distribution
Heterogeneity

HPC TOP 500 — Status & Trends

European machines in the TOP 500 as of November 2023:

- #5 HPE Cray — 2,752,704 cores — Fi
- #6 Bull — 1,824,768 cores — It
- #8 Bull — 680,960 cores — Es

→ **Increasing parallelism and distribution**

Meanwhile:

→ **Trend towards heterogeneity: GPUs, FPGAs, TPUs, variable precision FPUs...**

Hard to use efficiently, hard to program.

Systems	2012 BG/Q Computer	2022	Difference Today & 2022
System peak	20 Pflop/s	1 Eflop/s	O(100)
Power	8.6 MW (2 Gflops/W)	~20 MW (50 Gflops/W)	
System memory	1.6 PB (16*96*1024)	32 - 64 PB	O(10)
Node performance	205 GF/s (16*1.6GHz*8)	1.2 or 15TF/s	O(10) - O(100)
Node memory BW	42.6 GB/s	2 - 4TB/s	O(1000)
Node concurrency	64 Threads	O(1k) or 10k	O(100) - O(1000)
Total Node Interconnect BW	20 GB/s	200-400GB/s	O(10)
System size (nodes)	98,304 (96*1024)	O(100,000) or O(1M)	O(100) - O(1000)
Total concurrency	5.97 M	O(billion)	O(1,000)
MTTI	4 days	O(<1 day)	- O(10)

Source: J. Dongara, Grenoble Sep. 2019.
Big thanks to Henri-Pierre Charles (CEA).

A RISC-V HPC machine by 2030: vision and rationale

At historic rates of growth, it is possible that greater than 64 bits of address space might be required before 2030.
Let's assume that a full RISC-V 128 bit HPC machine could have (wild guess) 100 x 10^6 cores, as 1,000,000 heterogeneous clusters of 100 cores each with o(10 TB) RAM/cluster.

The challenge is how to take advantage of RISC-V and 128 bit to

- Manage the heterogeneity of the machine
- Optimize and simplify the operating system stack
- Increase the performance in distributed computing

Do not beat around the bush:
flat 128-bit address spaces will be adopted
as the simplest and best solution.

> "*There is only one mistake that can be made in computer design that is difficult to recover from — not having enough address bits for memory addressing and memory management.*"
>
> Bell and Strecker, ISCA-3, 1976.

RV128 spec is not frozen at this time, as there might be need to evolve the design based on actual usage of 128-bit address spaces.