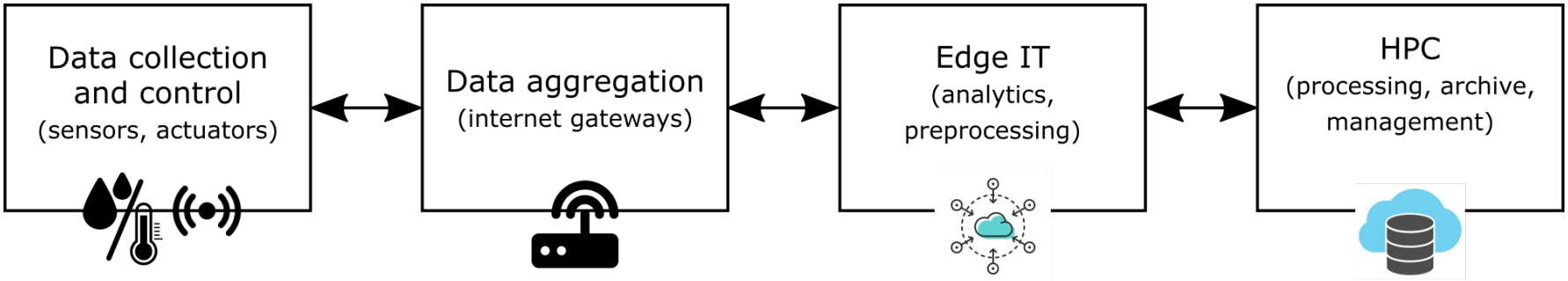
Design and implementation of an efficient Floating Point Unit (FPU) in the IoT era

OPRECOMP summer of code
NiPS Summer School
Sept 3-6, 2019 Perugia, Italy

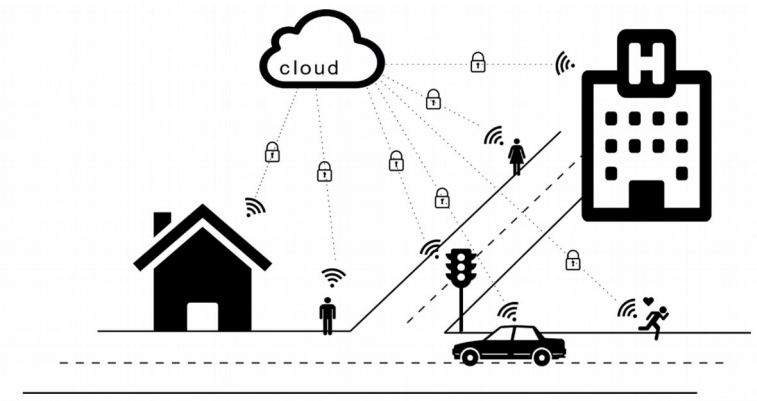
Andrea Galimberti
Andrea Romanoni
Davide Zoni

The computing platforms in the Internet-of-Things (IoT) era



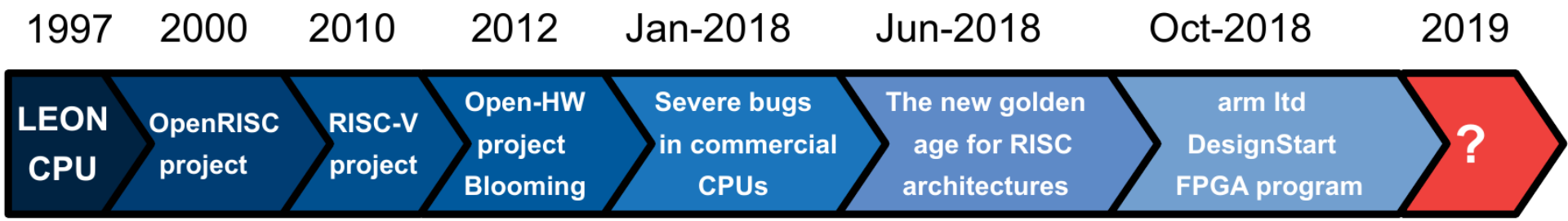
Efficient computing: challenges and opportunities in the Internet-of-Things (HiPEAC vision 2019*)

- Distributed data processing
- Specific requirements and constraints at each stage
- Revise the design of the System-on-Chip to maximize efficiency



* HiPEAC Vision 2019: https://www.hipeac.net/vision/2019/

Open-hardware platforms for research



Motivations (HiPEAC 2019):

- Design open-hardware computing platforms for efficiency, security and safety
- Trustfulness and security requirements
- Computational efficiency

Spectre and meltdown: ICT cannot rely on closed-hardware solutions

EPI European Processor Initiative

PULP Parallel Ultra Low Power

SiFive

Successful open-hardware projects

Limitations of current open-hardware solutions:

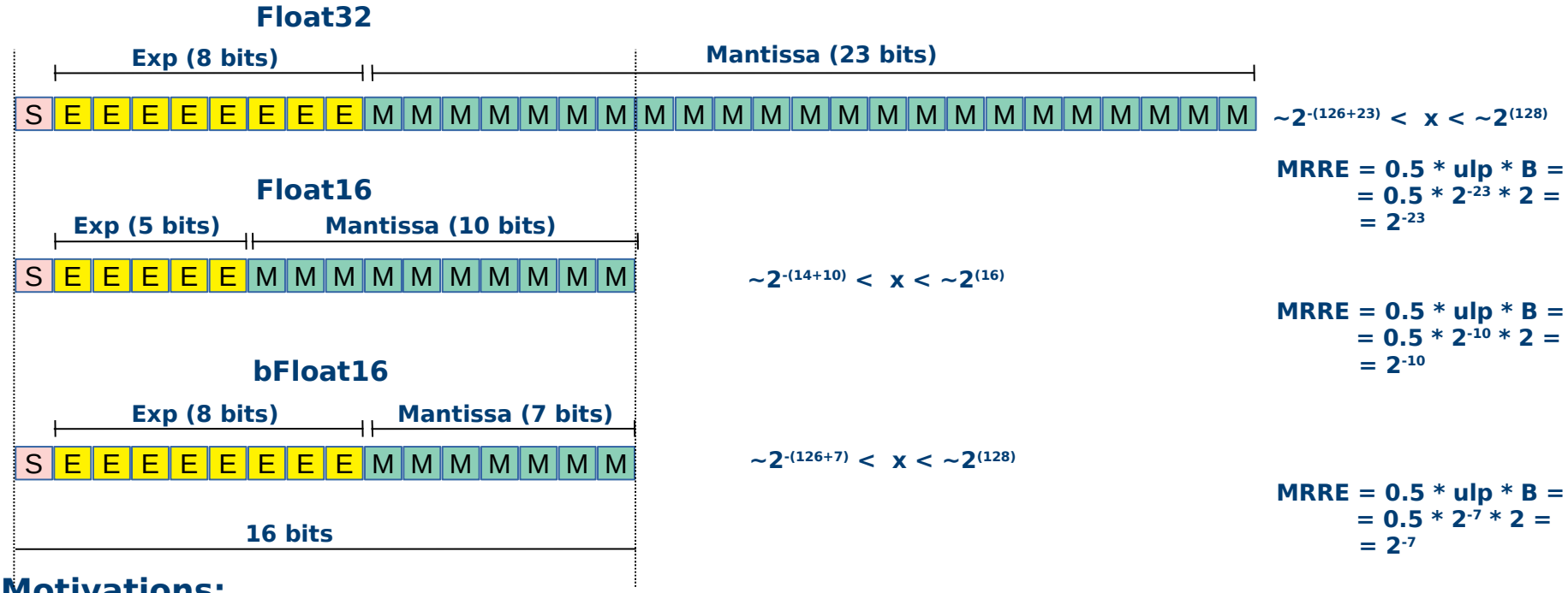
- Monolithic solutions, not modular
- Targeting efficient computation
- Mainly focused to ASIC design flow
- Not for research in microarchitecture

Let's focus on a single optimization direction ...

Floating point data transfer and computation dominate the energy consumption

The motivation behind the bfloat16 format

Brain-inspired Float (bfloat16): same format of the IEEE754 single-precision (SP) floating-point standard but truncates the mantissa field from 23 bits to 7 bits.



Motivations:

- Better timing and smaller area
- For machine learning tasks a 7 bit mantissa is enough (Google and Intel)
- Still possible to represent tiny values
- Easier debugging than float16 since bfloat16 shares the same format of the IEEE754-SP standard

Goals and status of the project

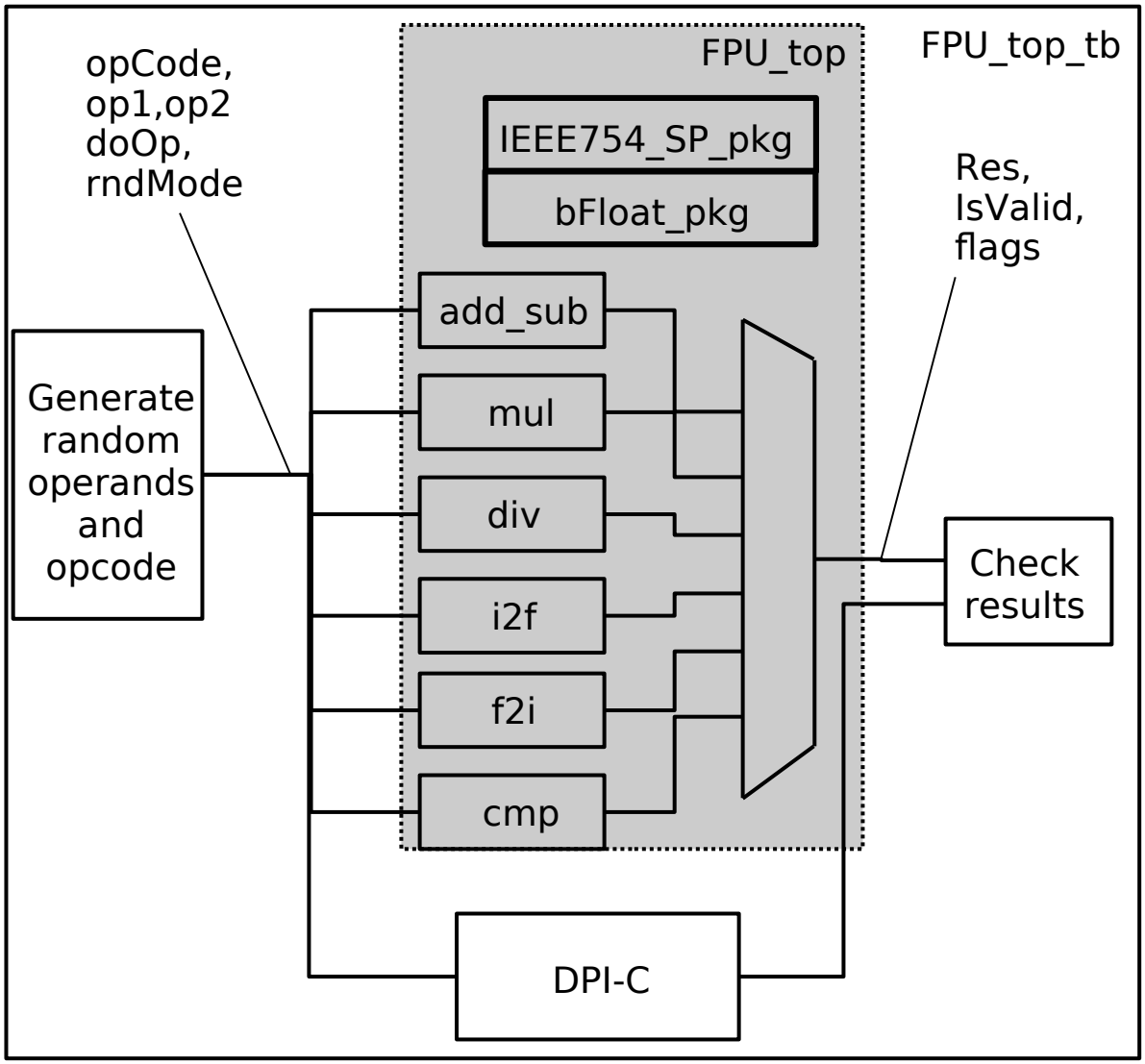
single SystemVerilog design that implements **bFloat** and **IEEE-754** single-precision

Efficient FPU for IoT SoCs, i.e., lamp-platform: both timing and area efficient

Reproducible results:

- new set of benchmarks targeting machine learning, vision and IA
- validation infrastructure using Xilinx Vivado, coupled with SystemVerilog DPI-C and random features. Free tools only

Design and verification methodology



Flexibility

- Two SV packages implementing the functions for each operation according to the selected size of the mantissa;
- Each operation can be either implemented or not

Efficiency:

- Rounding scheme can be configured to consider N LSB bits to optimize timing/area;

Constrained rnd verification using gcc and Xilinx Vivado

Area and timing: preliminary results

Operation	Area (LUT)		
	mor1kx (rnd=248LUT)	FPU-IEEE754-SP	FPU-bFoat
mul	625	1114	504
div			
add/sub	281	683	248
f2i	170	222	140
i2f	124	225	157
cmp	-	83	48
Total	1492	3245	1497

Operation	Timing		
	mor1kx	FPU-IEEE754-SP	FPU-bFoat
mul	50	28.03	22.09
div	170		
add/sub	40	22.12	16.71
f2i	20	9.28	8.48
i2f	20	11.17	10.25
cmp	10	4.26	4.27
worst	170	28.03	22.09

- We optimize each operation in the FPU to improve the timing
- The reduction in the mantissa improves both area and timing
- The proposed FPU allows to selectively implement a subset of the available operations
- NOTE: Flip-Flops results are not reported

Ongoing

- New set of benchmarks targeting machine learning, vision and IA
- FPU integration in the lamp-platform

FPU benchmarks for IoT System-on-Chip*

Motivations:

- Bare-metal C code for IoT devices (different from PARSEC and SPLASH2 bench suites)
- Consider emerging computing scenarios:

  (computer vision, computer graphics, machine learning)

Errors as function of the bits in the mantissa (tolerance 0.001)

	Number of bits in the mantissa											
	23	22	21	19	17	15	13	11	9	7	5	3
Laplacian Smoothing	0	0	0	0	0	67	161	199	220	228	229	231
Nuormals computation	0	0	0	0	0	0	0	0	127	378	426	444
Facet Subdivision	0	0	0	0	0	52	160	198	208	217	222	224
Self Intersection	0	0	0	0	0	6	5	5	5	6	22	30
ZNCC	0	0	0	0	0	0	1	1	1	1	1	1
Ray Triangle Intesection	0	0	0	0	0	0	1	4	15	18	21	21
Kalman Filter	0	0	0	0	0	3	49	83	92	98	98	100
Neares Neighbor	0	0	0	0	0	0	0	0	3	12	31	84
Linear Regression	0	0	8	46	62	67	89	124	128	134	133	134
K-Means	0	0	0	0	0	0	0	3	3	3	304	304
Q-learning	0	0	0	12	13	13	13	13	13	13	13	13
LSTM	0	0	4	16	18	19	20	20	20	20	19	20

* Project website: www.lamp-platform.org

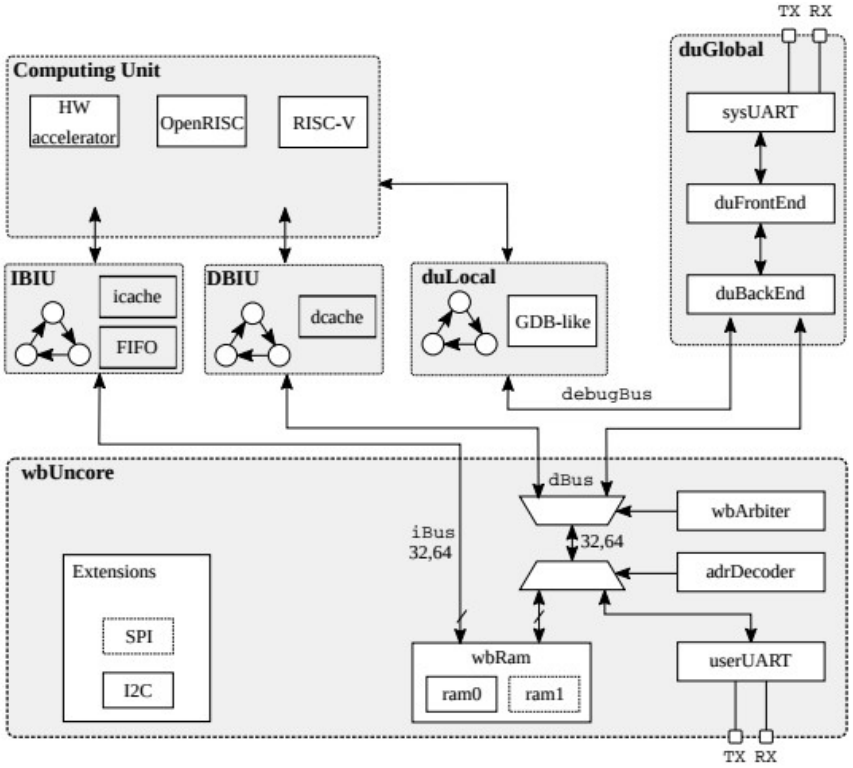
Lightweight Application-specific Modular Processor (LAMP)*

Key idea: The Raspberry successful story

A small and affordable computer that you can use to learn programming

Our vision:

LAMP-platform: a flexibile and affordable System-on-Chip for investigating IoT challenges



Contributions:

- FPGA targets with up to 4k LUTs with CPU

(a RISC-V 5 stage CPU is offered as well)

- Modular design to target different design metrics
  - For example: Side-channel evaluation using the same design in post-synthesis, post-implementation and board
- Designed for Artix 7 FPGA family
- Easily plug different computing devices
  (request/response data/instruction interface)

* Project website: www.lamp-platform.org

Thank you

Andrea Galimberti, PhD Student
Email: andrea.galimberti@polimi.it

Andrea Romanoni, PhD
Email: andrea.romanoni@polimi.it

Davide Zoni, PhD
Email: davide.zoni@polimi.it