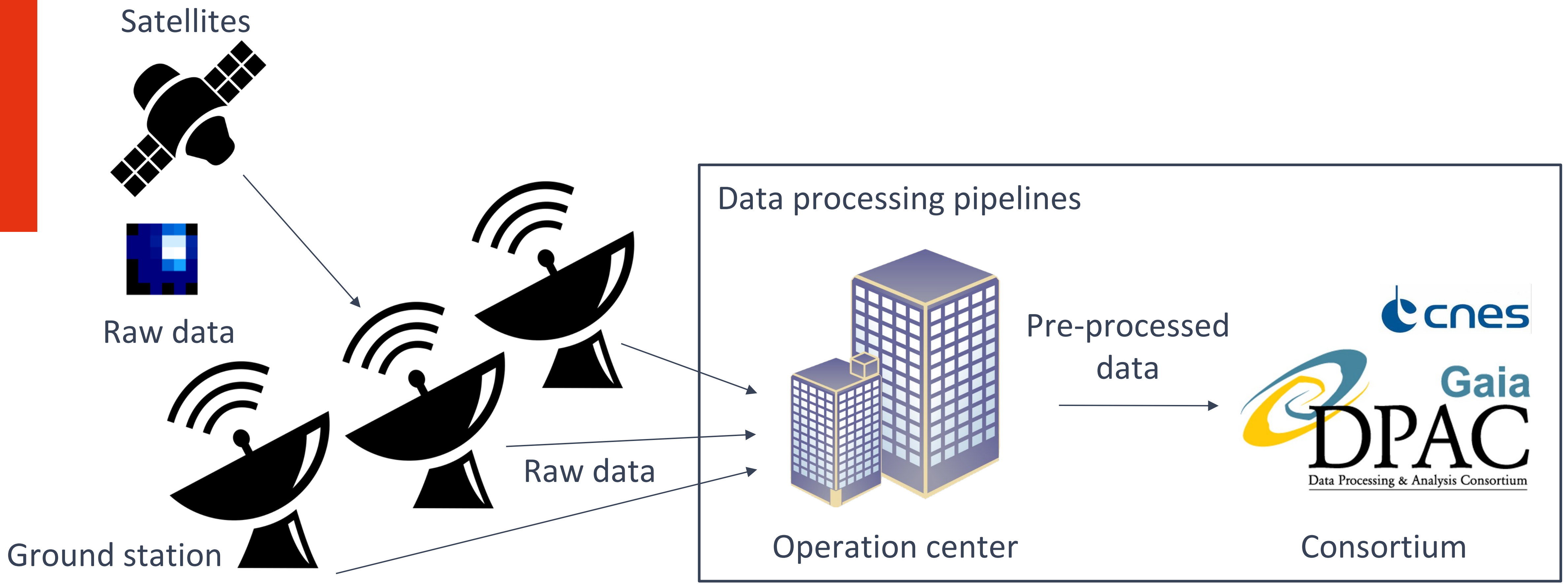
EDHPC 2023
European Data Handling &
Data Processing Conference for Space
2 - 6 October 2023 | Juan-Les-Pins | France

High-Level Synthesis (HLS)-Based On-board Payload Data Processing considering the Roofline Model

Seungah Lee, Ruben Salvador, Angeliki Kritikakou,
Olivier Sentieys, Julien Galizzi, and Emmanuel Casseau

Payload data processing

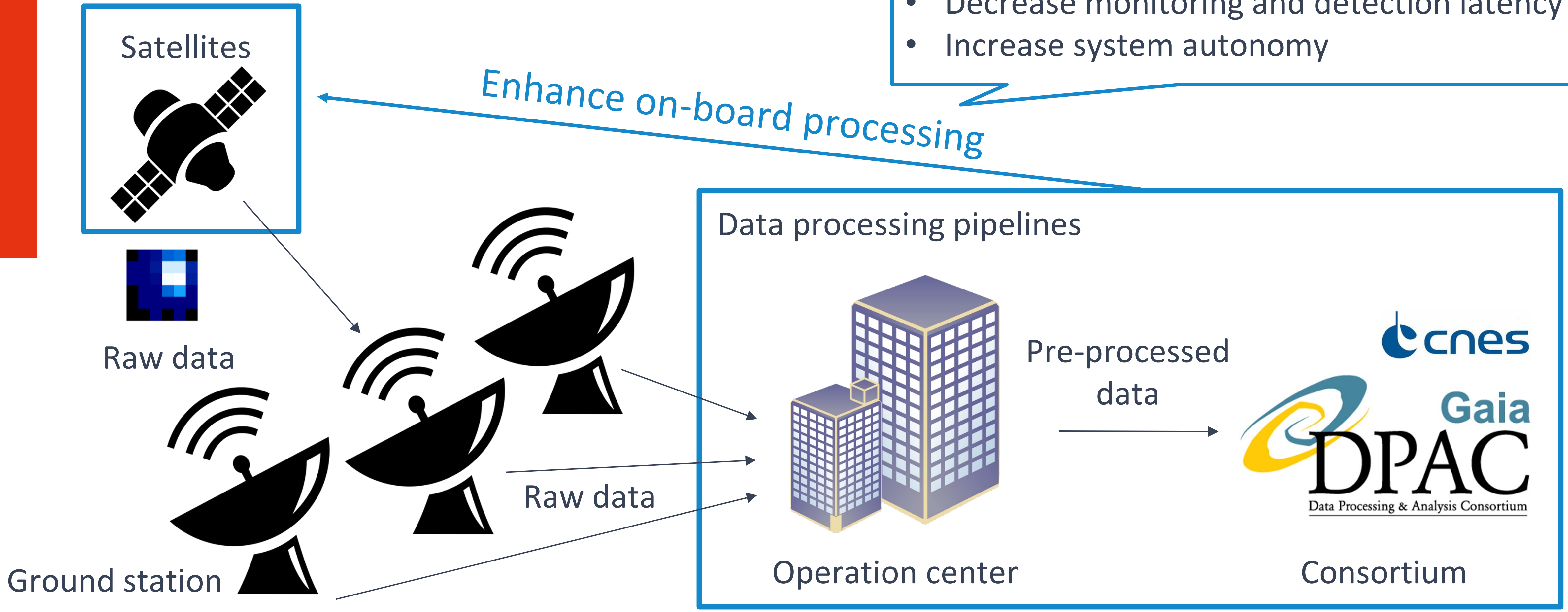


[1] Gaia Collaboration, "The Gaia mission," *A&A*, vol. 595, p. A1, Nov. 2016.

Payload data processing

Advantages

- Decrease monitoring and detection latency
- Increase system autonomy



[1] Gaia Collaboration, "The Gaia mission," *A&A*, vol. 595, p. A1, Nov. 2016.

Current data processing hardware

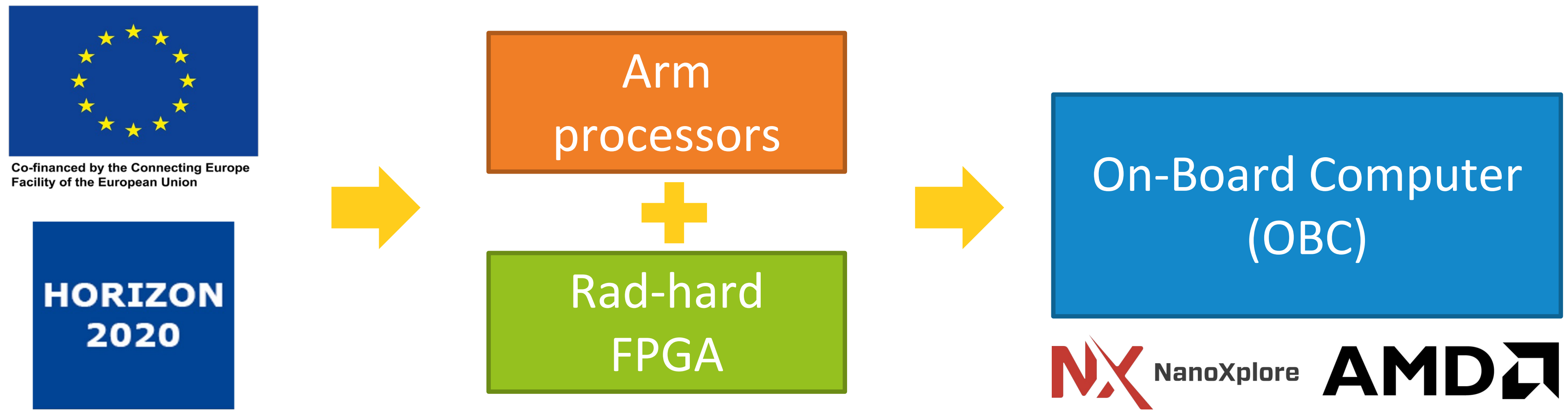
- Processing hardware
  - Frontgrade LEON processors and Microsemi FPGA

Mission	Launch year	Payload data processing hardware	Reference
Plato	2026*	GR712RC LEON3FT ASIC, MDPA LEON2FT ASIC, LEON3FT with RTAX2000 FPGA	[2]
JUICE	2023	GR712RC LEON3FT ASIC	[3]
Solar Orbiter	2020	LEON3FT with RTAX4000 FPGA	[4]
Cheops	2019	GR712RC LEON3FT ASIC	[5]
BepiColombo	2018	HIREC-MIPS-HR5000 CPU, RTAX2000 FPGA	[6]
Gaia	2013	Custom pre-processing board, SCS750 PowerPC board	[7]

*Estimated launch schedule

Future data processing hardware

European space technology roadmap



Research focus: FPGA design based on High-Level Synthesis (HLS)

Space science algorithms

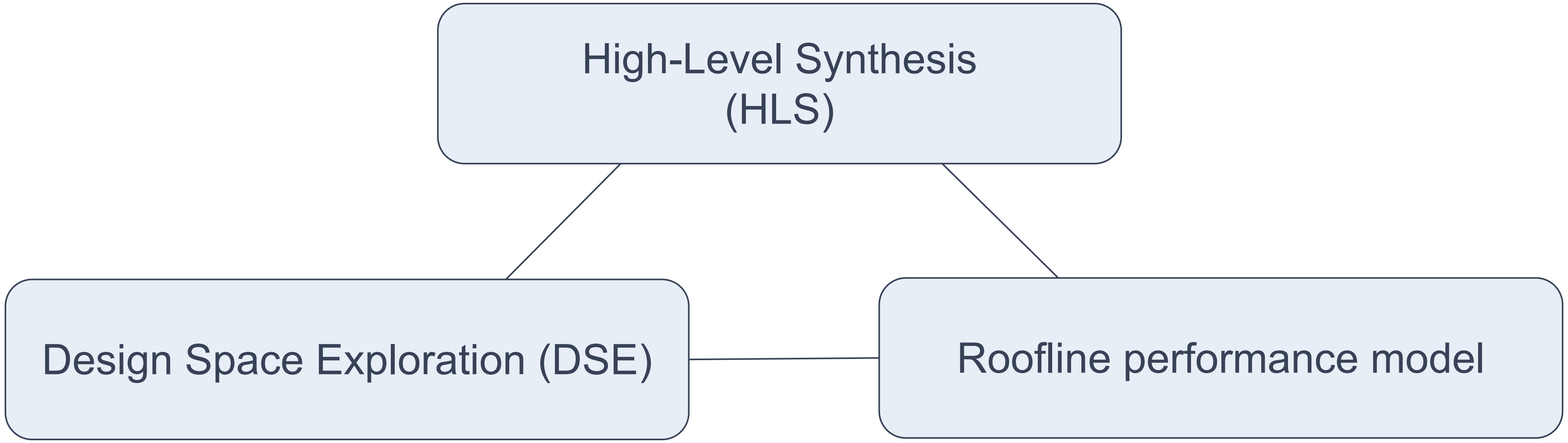
Survey results from payload teams

Classification	Sub-classification	Nb. of users
Fourier transform	FFT, IFFT, DFT	5
Filter	IIR	4
	CIC	4
	Kalman	1
Compression	CCSDS 121	3
	CCSDS 122	3
	CCSDS 123	3
	CCSDS 124	3
Optimization	Interpolation	3
	Fitting and correlation	2
	Gradient descent	2
Histogram		1
Digital Elevation Model		1

Selected key algorithm: 2-Dimensional Fast Fourier Transform (2-D FFT)

Optimization methodology

- High-Level Synthesis (HLS)-based hardware acceleration architectures
  - Combination of Design Space Exploration (DSE) and the roofline model [8]

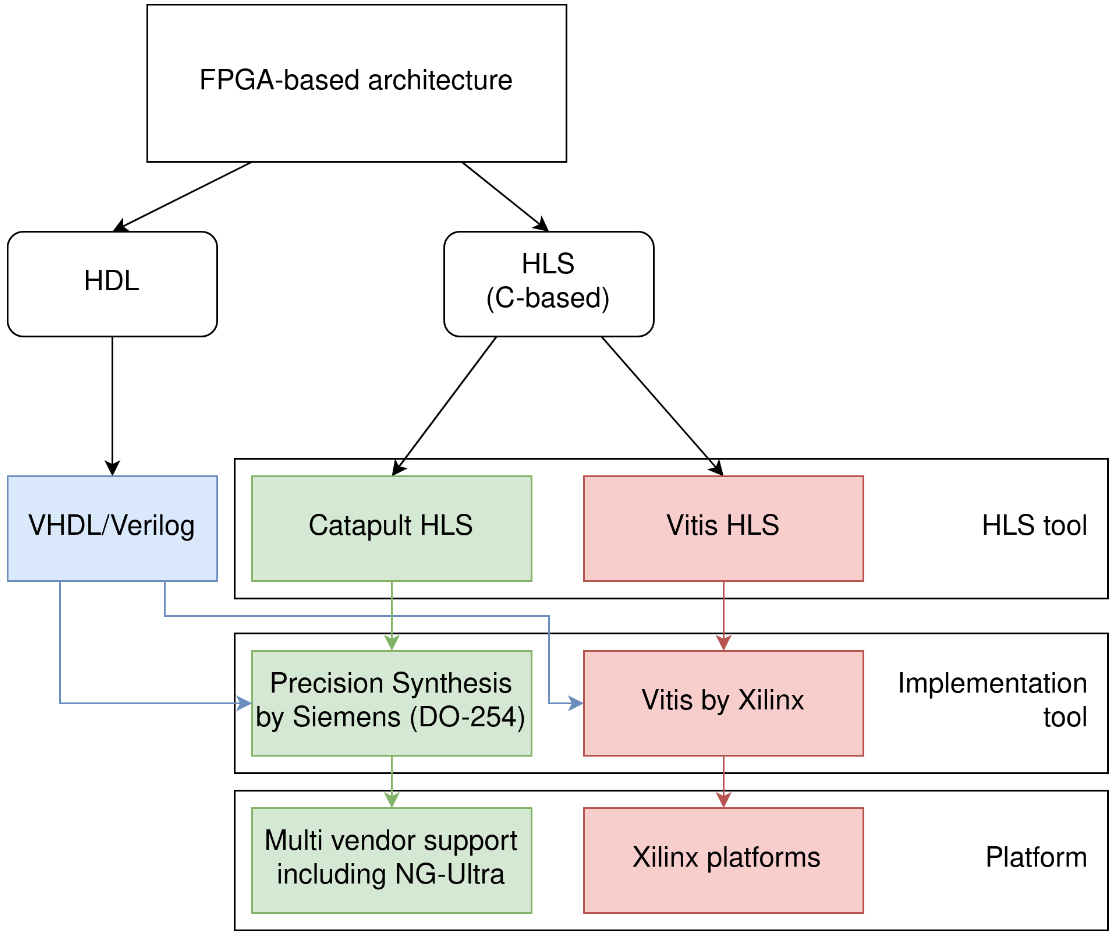

High-Level Synthesis (HLS)
Design Space Exploration (DSE)
Roofline performance model

[8] M. Siracusa et al., “A Comprehensive Methodology to Optimize FPGA Designs via the Roofline Model,” IEEE Trans Comput, vol. 71, no. 8, pp. 1903–1915, Aug. 2022

High-Level Synthesis (HLS)

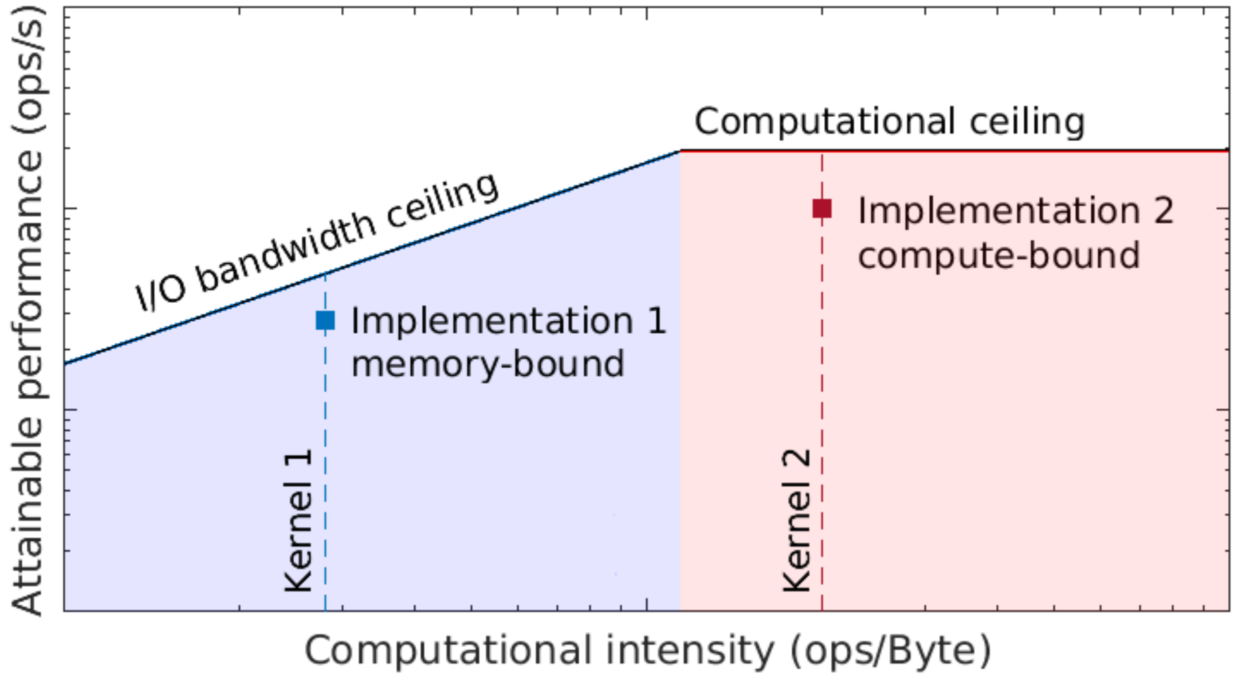
- HLS: refactor C/C++ using pragmas and directives
- Tools: Vitis HLS, Catapult HLS, Bambu HLS…
- Advantages
  - Generate architectures faster and efficiently
  - Adaptive to missions and design reviews
  - For payload teams with limited HDL experts

Optimization type	Vitis HLS pragmas
Loop unrolling	#pragma HLS unroll
Loop pipelining	#pragma HLS pipeline
Task pipelining	#pragma HLS dataflow
Array partitioning	#pragma HLS array_partition



Roofline performance model

- Characterize designs using the limits of bandwidth and performance on a given architecture [9].
- Originally appropriate to multicore CPUs and GPUs, but extended to FPGAs.
- The FPGA roofline model shall consider the reconfigurable characteristics of FPGA [10].



Use case: 2-D FFT

- Algorithm requirements

Item	Description	Note
Algorithm	2-D FFT	SVOM ECLAIRs payload [11]
FFT size	200×200 (256×256)	
Data type	Single-precision floating point (FP32)	
Execution time	< 100 ms	

- FPGA platform specifications

Item	Description	Note
Digital Signal Processing (DSP) blocks	2520 blocks (DSP48E2) Multiplier input: 27×18 bit	Xilinx Zynq UltraScale+ (XCZU9EG) embedded FPGA
Block RAM (BRAM)	32.1 Mb	
Interface	Advanced eXtensible Interface (AXI)	

Theoretical FPGA roofline model

- **Computational ceilings (C)**

$$C = \frac{available\ DSP\ blocks \times clock\ frequency}{required\ DSP\ blocks\ per\ operation}$$

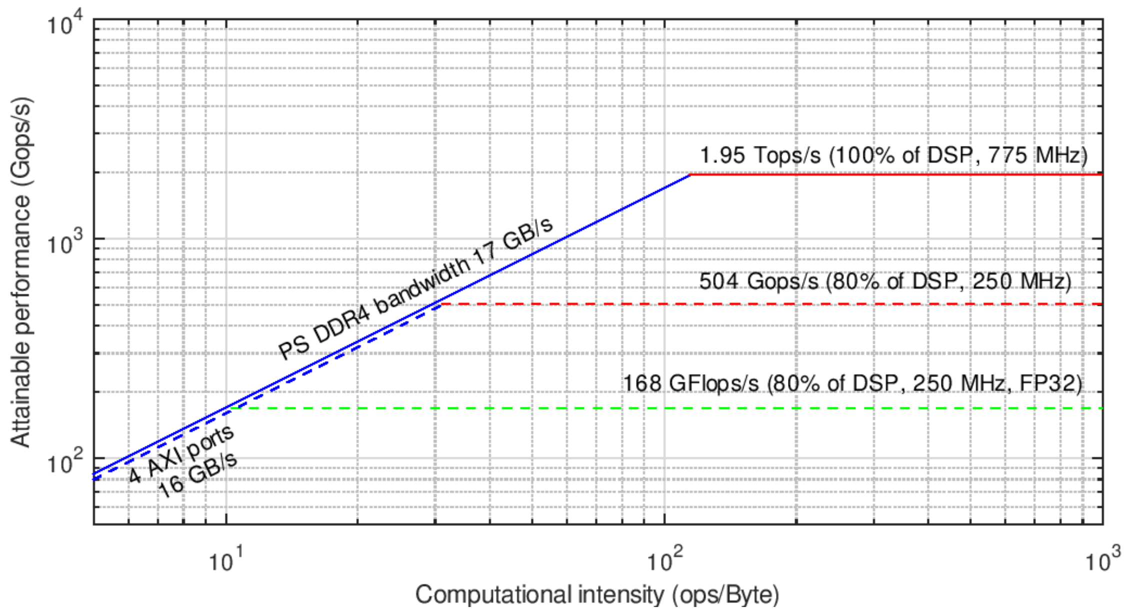
- DSP blocks per operation (DSP48E2)
  - 27-bit data: 1 DSP block per multiplication
  - FP32 data: 3 DSP blocks per FP multiplication

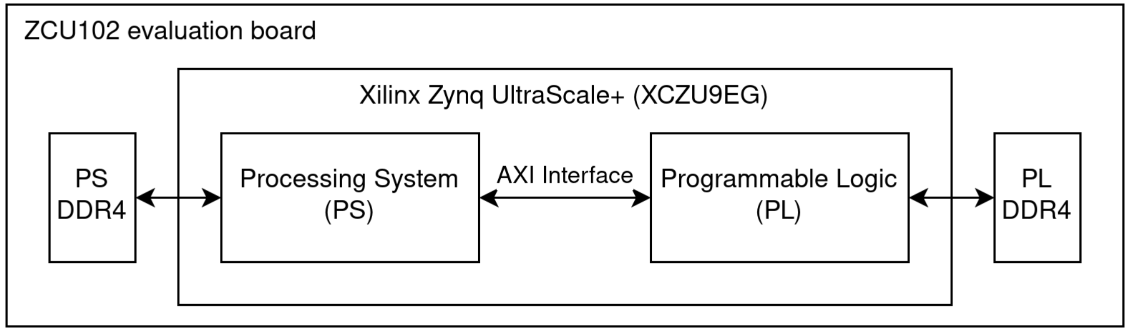
- **I/O Bandwidth ceiling (BW)**

$$BW = min(BW_{PS\_DDR4}, BW_{AXI})$$

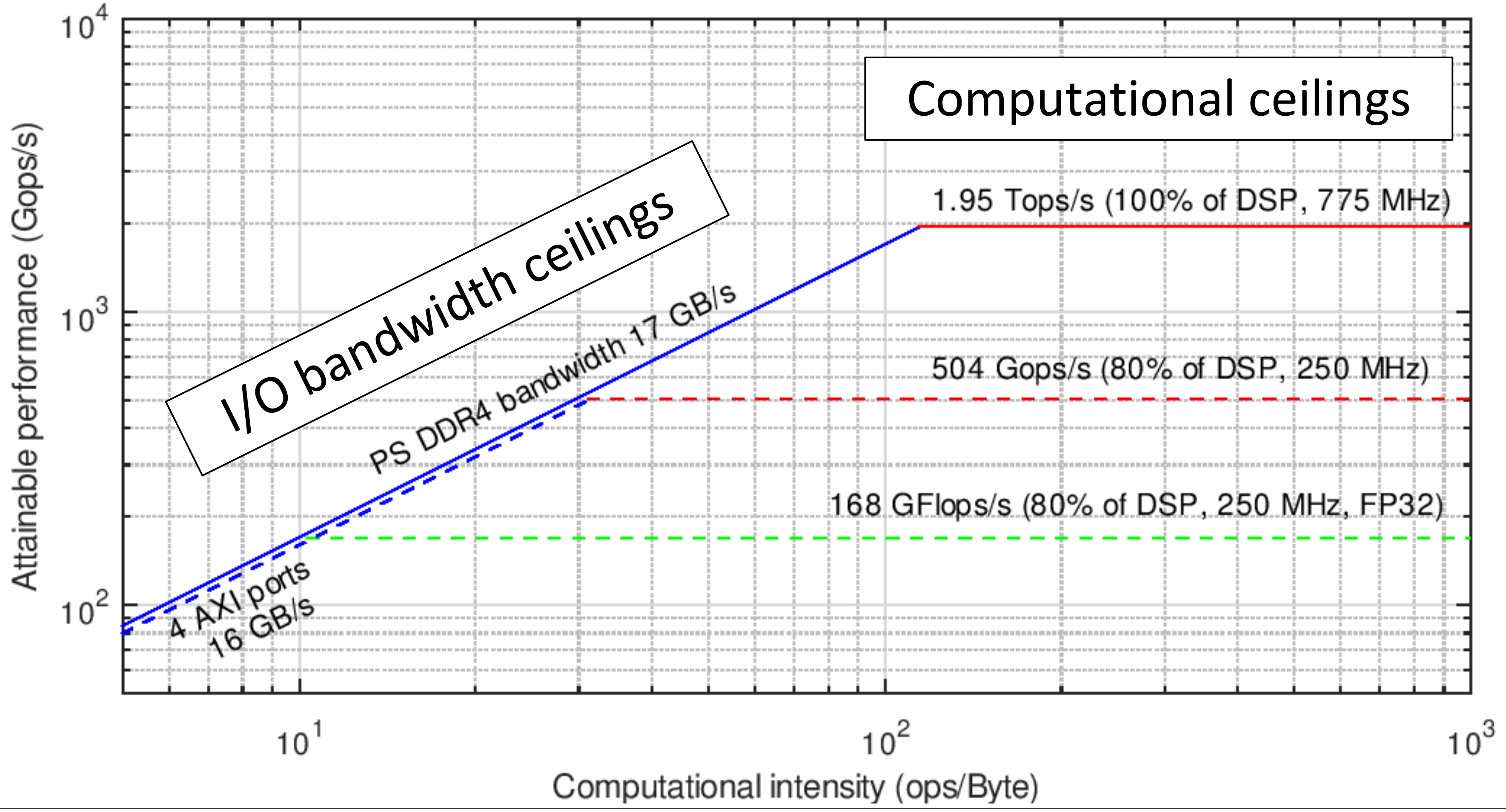
$$BW_{PS\_DDR4} = DDR4\ transfer\ rate \times DDR4\ width$$

$$BW_{AXI} = clk\ freq. \times transfer\ data\ bitwidth \times ports$$





Theoretical FPGA roofline model



Theoretical FPGA roofline model

- **Computational ceilings (C)**

$$C = \frac{available\ DSP\ blocks \times clock\ frequency}{required\ DSP\ blocks\ per\ operation}$$

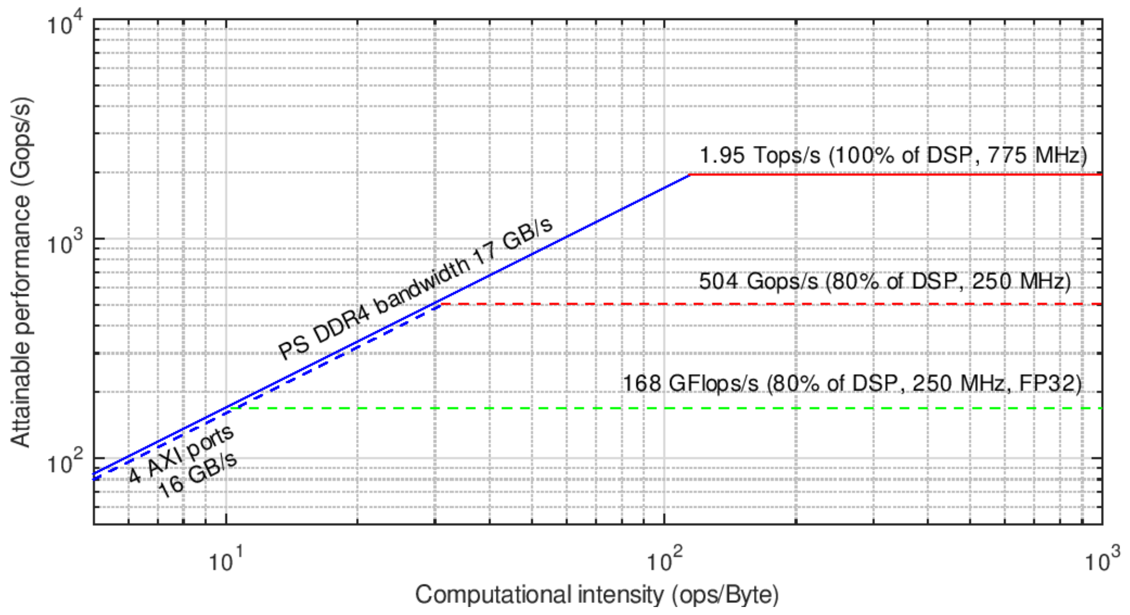
- DSP blocks per operation (DSP48E2)
  - 27-bit data: 1 DSP block per multiplication
  - FP32 data: 3 DSP blocks per FP multiplication

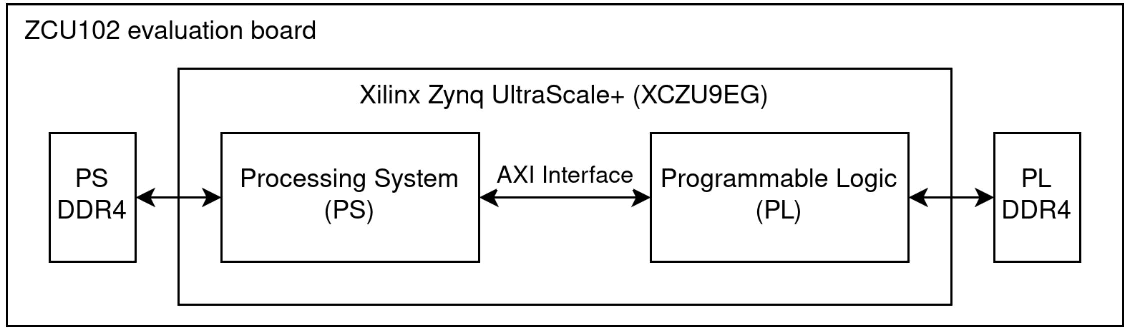
- **Bandwidth ceiling (BW)**

$$BW = min(BW_{PS\_DDR4}, BW_{AXI})$$

$$BW_{PS\_DDR4} = DDR4\ transfer\ rate \times DDR4\ width$$

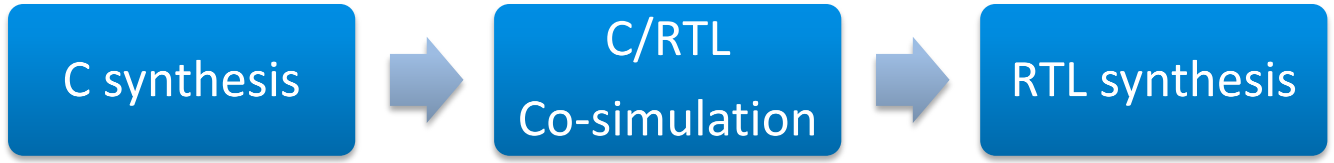
$$BW_{AXI} = clk\ freq. \times transfer\ data\ bitwidth \times ports$$





Parallelism and pipelining

- AMD-Xilinx open-source DSP library
  - HLS-based 2-D FFT
- Parallelism and pipelining
  - Loop unrolling combined with array partitioning
  - Loop pipelining and task pipelining
- Optimization (FP32)
  - Reference code: DSP blocks utilization > 100 %
  - Modification: loop pipelining (#pragmas HLS pipeline)
  - Reduction of resource utilization



FFT size	64 x 64		256x256	
Data type	27-bit FxP	FP32	27-bit FxP	FP32
DSP blocks	196 (7.8%)	928 (37%)	308 (12%)	1276 (51%)
BRAM	48 (2.6%)	32 (1.8%)	618 (34%)	528 (29%)
LUT	58843 (21%)	109870 (40%)	89508 (33%)	157710 (58%)
FF	82129 (15%)	199467 (36%)	103394 (19%)	262969 (51%)
Frequency	225 MHz	227 MHz	224 MHz	217 MHz

FPGA roofline model for the 2-D FFT

Ceilings

$$Computational\ ceiling = \frac{\#\ implemented\ DSP\ blocks \times clk\ freq.}{\#\ required\ DSP\ blocks\ per\ operation}$$

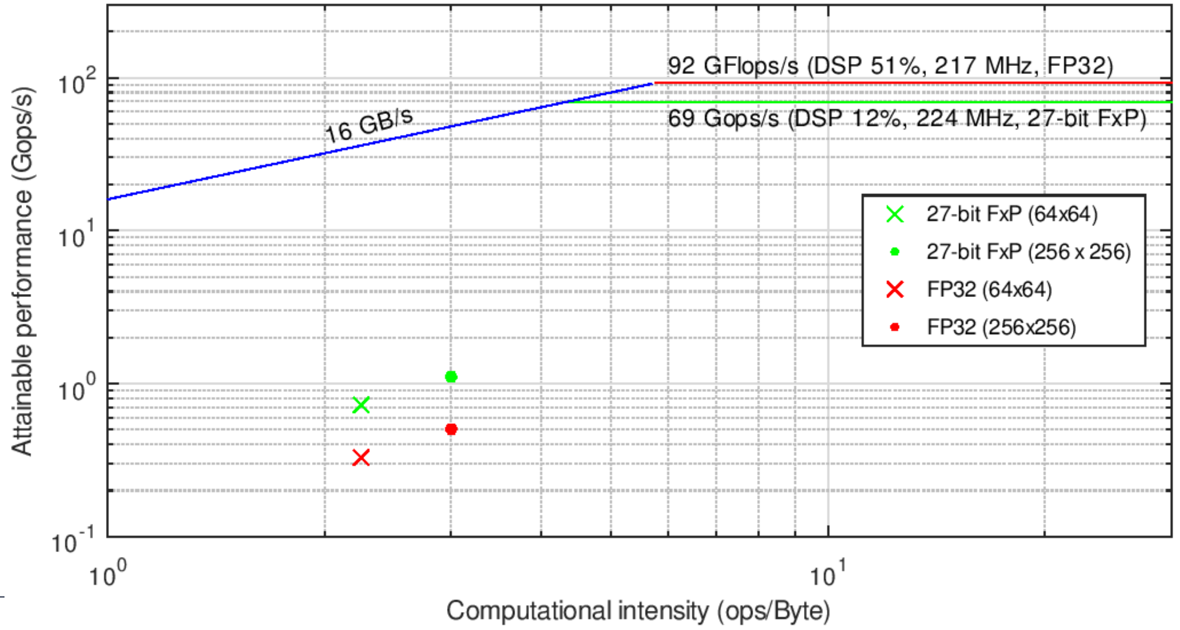
$$Bandwidth\ ceiling = min(BW_{PS_{DDR4}}, BW_{AXI}) = BW_{AXI}$$

Architecture designs

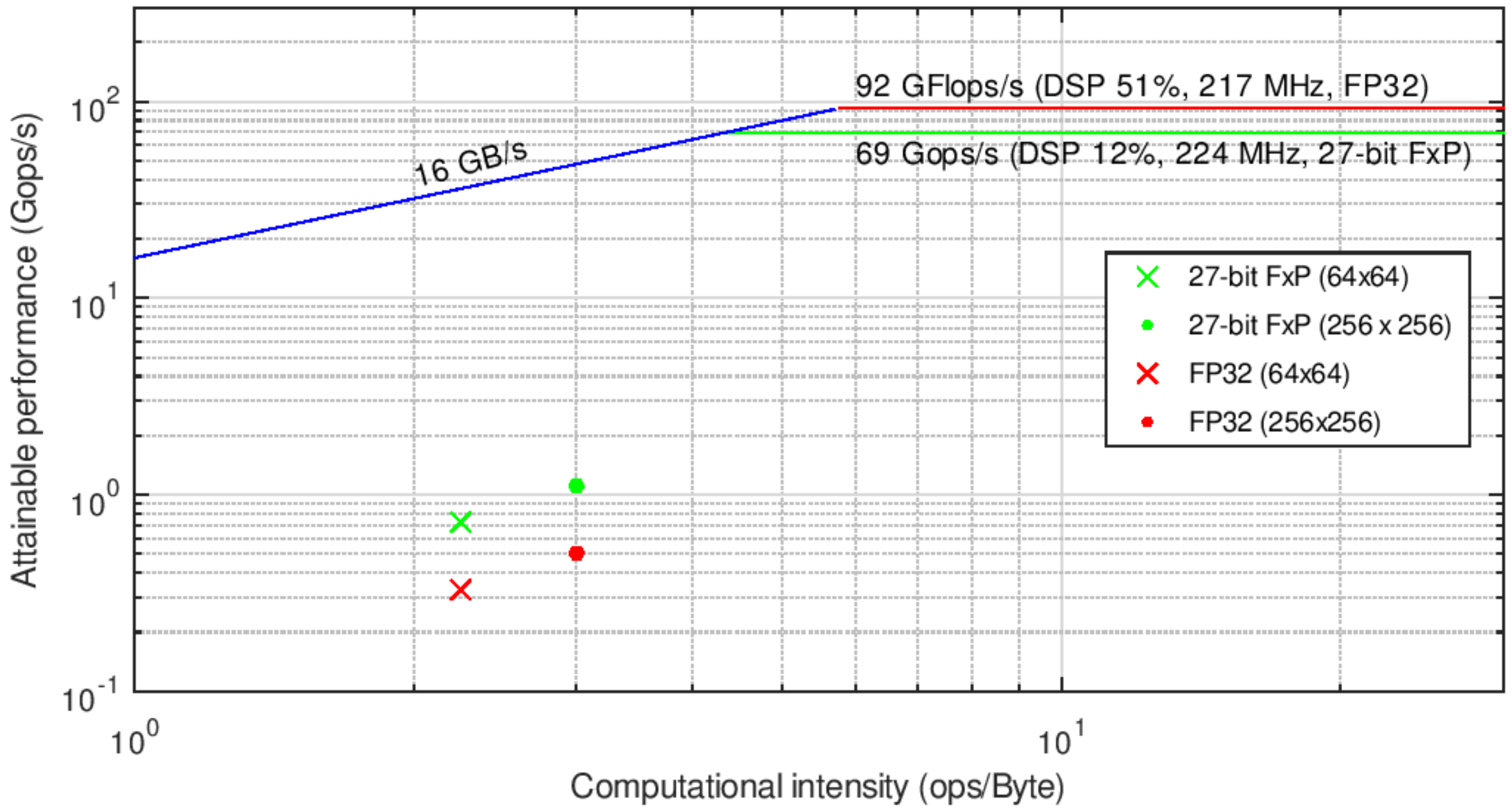
$$Computational\ intensity = \frac{\#\ real\ operations}{data\ accesssed\ in\ bytes}$$

$$Performance = \frac{\#\ real\ operations}{execution\ time}$$

FFT size	256x256	
Data type	27-bit FxP	FP32
DSP blocks	308 (12%)	1276 (51%)
BRAM	618 (34%)	528 (29%)
Frequency	224 MHz	217 MHz
Total latency	1.5 ms	3 ms



FPGA roofline model for the 2-D FFT



FPGA roofline model for the 2-D FFT

Time requirement: < 100 ms
30-65 times faster!

Ceilings

$$\text{Computational ceiling} = \frac{\#\,\text{implemented DSP blocks} \times \text{clk freq.}}{\#\,\text{required DSP blocks per operation}}$$

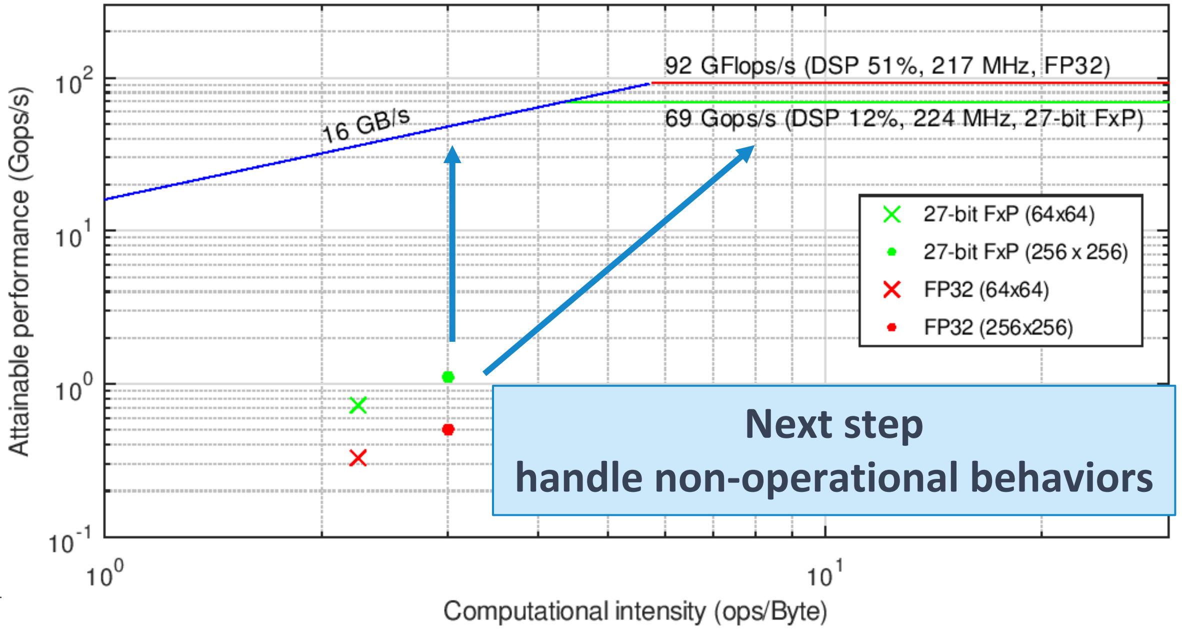
$$\text{Bandwidth ceiling} = \min\left(BW_{PS_{DDR4}}, BW_{AXI}\right) = BW_{AXI}$$

Architecture designs

$$\text{Computational intensity} = \frac{\#\,\text{real operations}}{\text{data accesssed in bytes}}$$

$$\text{Performance} = \frac{\#\,\text{real operations}}{\text{execution time}}$$

FFT size	**256x256**	
Data type	27-bit FxP	FP32
DSP blocks	308 (12%)	1276 (51%)
BRAM	618 (34%)	528 (29%)
Frequency	224 MHz	217 MHz
Total latency	**1.5 ms**	**3 ms**



Next step
handle non-operational behaviors

Conclusion

- Analysis of current and future on-board data processing hardware
- Survey on payload data processing algorithms
- Design methodology considering the roofline model with HLS-based DSE
- The 2-D FFT case study: 30-65 faster processing than the algorithm requirement
- On-going work: HW/SW benchmark and roofline model for Zynq UltraScale+ (Arm CPU&FPGA)

Thank you for your attention!

Main speaker: **Seungah Lee** (seungah.lee@irisa.fr)

Acknowledgement: this research was supported by CNES and University of Rennes thanks to PhD funding granted to Seungah Lee.

References

[1] Gaia Collaboration, “The Gaia mission,” A&A, vol. 595, p. A1, Nov. 2016.

[2] P. Plasson, G. Brusq, F. Singhoff, H. N. Tran, S. Rubini, and P. Dissaux, “PLATO N-DPU on-board software: an ideal candidate for multicore scheduling analysis,” in 11th European Congress ERTSS Embedded Real Time Software and System, Toulouse, France, 2022.

[3] F. Torelli, “Common DPU and Basic Software for JUICE Instruments,” in European Workshop on On-Board Data Processing (OBDP2019), European Space Research and Technology Centre (ESTEC), Feb. 2019.

[4] M. Maksimovic et al., “The Solar Orbiter Radio and Plasma Waves (RPW) instrument,” Astronomy & Astrophysics, vol. 642, p. A12, Oct. 2020.

[5] W. Benz et al., “The CHEOPS mission,” Experimental Astronomy, vol. 51, no. 1, pp. 109–151, Feb. 2021.

[6] Y. Kasaba et al., “Mission Data Processor Aboard the BepiColombo Mio Spacecraft: Design and Scientific Operation Concept,” Space Science Reviews, vol. 216, no. 3, p. 34, Apr. 2020.

[7] J. H. J. d. Bruijne, M. Allen, S. Azaz, A. Krone-Martins, T. Prod’homme, and D. Hestroffer, “Detecting stars, galaxies, and asteroids with Gaia,” Astronomy & Astrophysics, vol. 576, p. A74, Apr. 2015.

[8] M. Siracusa et al., “A Comprehensive Methodology to Optimize FPGA Designs via the Roofline Model,” IEEE Trans Comput, vol. 71, no. 8, pp. 1903–1915, Aug. 2022.

[9] S. Williams, A. Waterman, and D. Patterson, “Roofline: an insightful visual performance model for multicore architectures,” Communications of the ACM, vol. 52, no. 4, pp. 65–76, Apr. 2009.

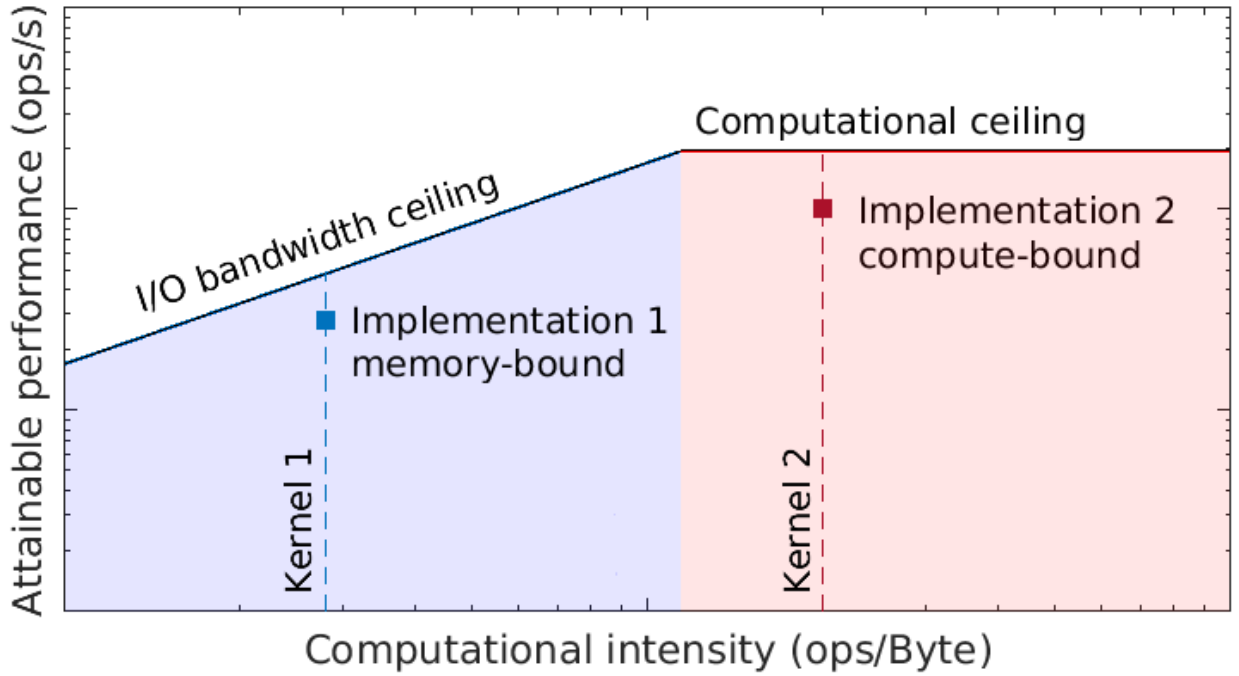
[10] B. da Silva Gomes, A. Braeken, E. D’Hollander, and A. Touhafi, “Performance modeling for FPGAs: extending the roofline model with high-level synthesis tools,” International Journal of Reconfigurable Computing, vol. 2013, pp. 1–10, 2013.

[11] S. Schanne et al., “A Scientific Trigger Unit for space-based real-time gamma ray burst detection I - Scientific software model and simulations,” IEEE Nuclear Science Symposium and Medical Imaging Conference (2013 NSS/MIC), Oct. 2013, pp. 1–5

Appendix

Roofline performance model

- Characterize designs using the limits of bandwidth and performance on a given architecture [9].
- Originally appropriate to multicore CPUs and GPUs, but extended to FPGAs.
- The FPGA roofline model shall consider the reconfigurable characteristics of FPGA [10].



Ceilings

C = func(resource, operation, clock frequency)

BW = func(external memory, interface)

Kernel designs

$$\textit{Computational intensity (CI)} = \frac{\textit{arithmetic operations}}{\textit{data byte accessed}}$$

$$\textit{Performance} = \frac{\textit{arithmetic operations}}{\textit{execution time}}$$