

The METASAT FPGA Prototype: A High-Performance Multicore, AI SIMD and GPU RISC-V Platform for On-board Data Processing

MODULAR MODEL-BASED DESIGN AND
TESTING FOR APPLICATIONS IN SATELLITES

Leonidas Kosmidis

ADCSS 2023



**Barcelona
Supercomputing
Center**
Centro Nacional de Supercomputación

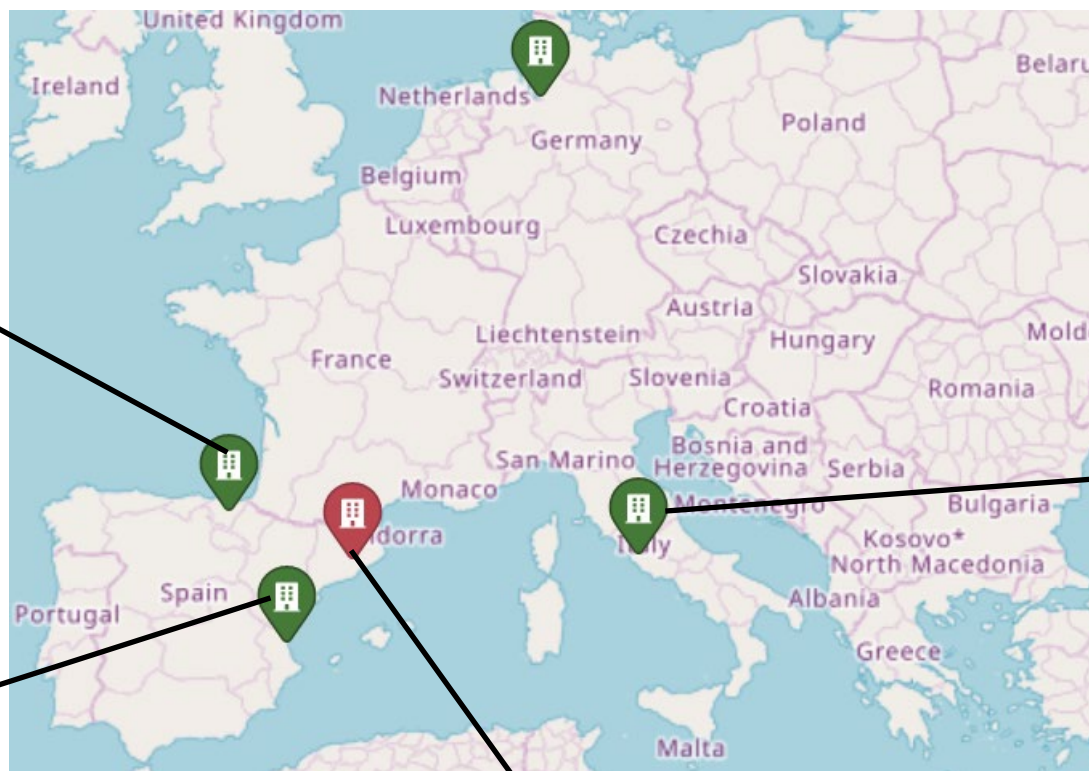


**UNIVERSITAT POLITÈCNICA
DE CATALUNYA**
BARCELONATECH

14th October 2023

METASAT Consortium

- 2-year Horizon Europe project: January 2023-December 2024
- TRL 3-4



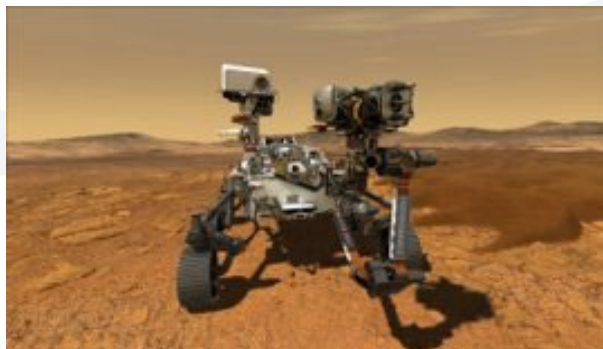
Collins Aerospace



**Barcelona
Supercomputing
Center**
Centro Nacional de Supercomputación

Introduction

- Modern and upcoming space systems require increasing levels of computing power
- Traditional space processors cannot provide this performance level
- Need for higher performance hardware in space systems



METASAT Overview

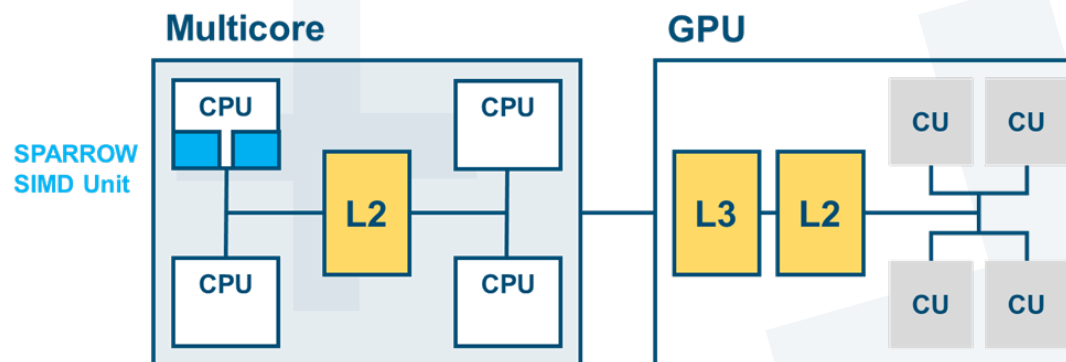
- Modern aerospace systems require new, advanced functionalities
 - Artificial Intelligence (AI)
 - High Resolution Sensors
 - Optical communications
 - Advanced Robotics...
- Advanced functionalities require complex hardware and software compared to the existing space technologies
- High Performance Hardware technologies: Advanced Multi-cores, GPUs, AI accelerators
- Programming high performance hardware requires complex software: parallel and GPU programming

Model-Based Design

- Model-Based Design can reduce the development and verification time for these complex platforms
- Development can be assisted by high level design methods (models) from which code can be automatically generated
 - Correct-by-construction
 - Various levels of verification: model-in-the-loop, software-in-the-loop, processor-in-the-loop etc
 - Virtual platforms allow starting software development before the hardware is ready
 - Break the dependency between hardware and software development

Hardware Selection

- No hardware with such architectural complexity exists for the space domain
- COTS Embedded Multicore and GPU devices provide these features but depend on non-qualifiable software stacks
 - GPU drivers available only for Linux
 - Blocking point for use in institutional missions
- Design a prototype hardware platform based on the RISC-V ISA



Virtualisation

- Time and Space isolation provide benefits for faster and easier integration
- Components can be developed and tested in isolation
- Fault Detection, Isolation and Recovery (FDIR)

METASAT



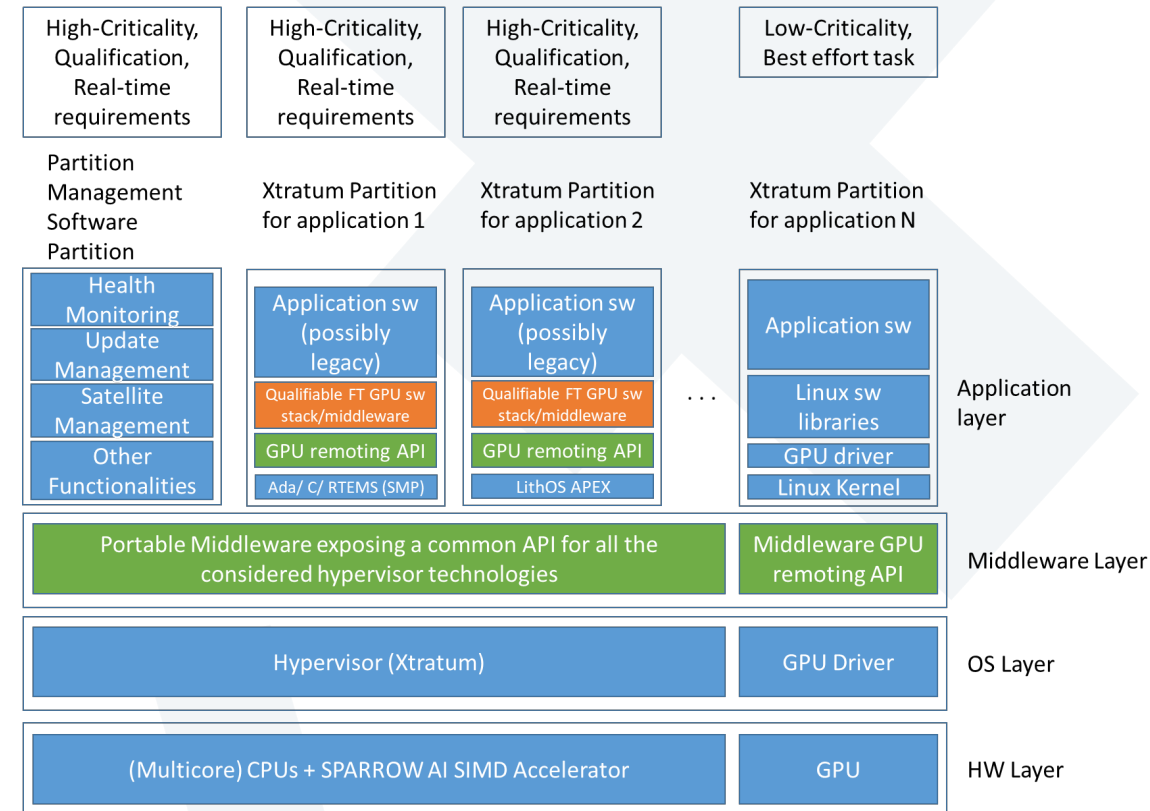
- METASAT will rely on open source and standardized technologies
 - Maximise interoperability and avoid vendor lock-in
 - Facilitate the development of a space ecosystem
- ESA's TASTE Open Source Model Based Design framework, enhanced with support for high performance platforms such as multicores and GPUs
- Open Source Processor technologies such as Gaisler's NOEL-V RISC-V processors
 - Enhancement with AI processing acceleration capabilities

taste

The METASAT RISC-V Platform

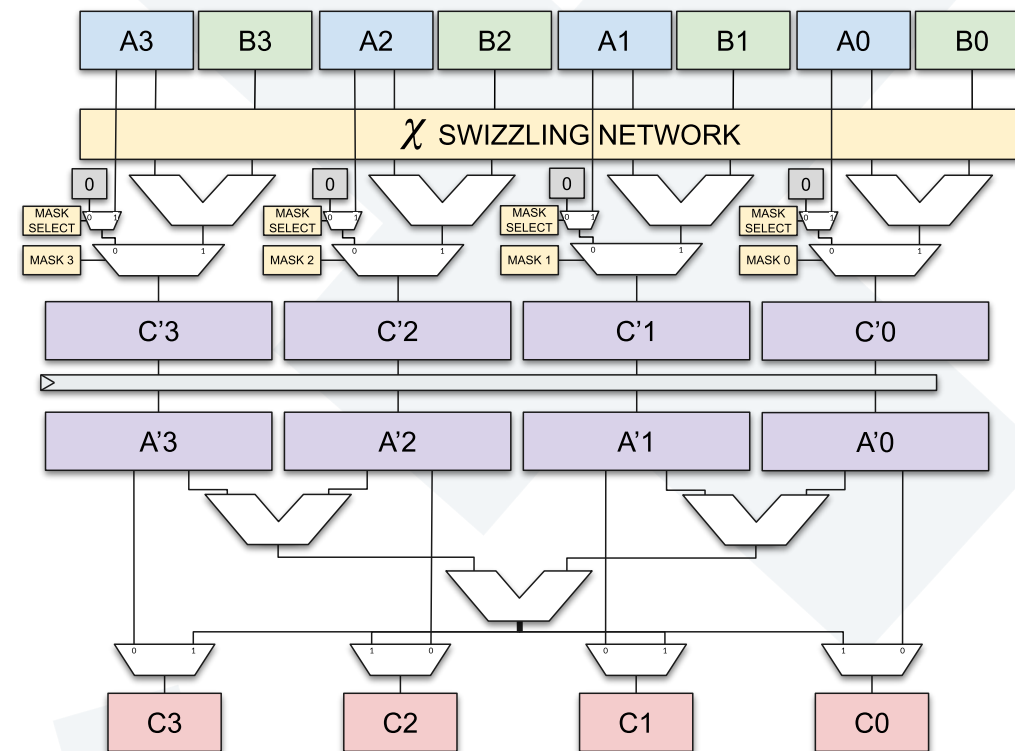


- Mixed Criticality Platform
- FPGA Prototype on a Xilinx VCU118
- Multicore CPU Based on NOEL-V + SPARROW AI SIMD Accelerator
 - Qualifiable software stack for high criticality software with moderate AI acceleration needs



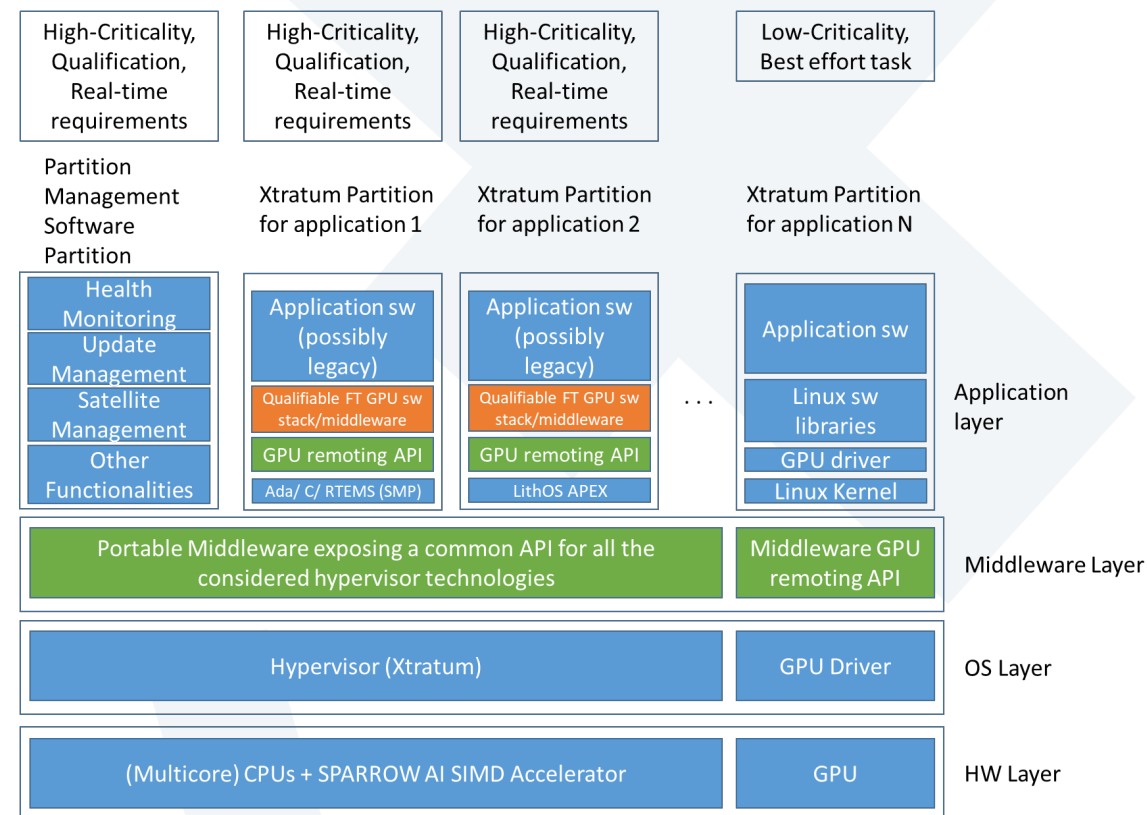
The METASAT RISC-V Platform

- SPARROW AI SIMD Accelerator
- High-performance, Low-cost at least 30% smaller than conventional vector processors with similar performance
- Minimal core modifications
 - incremental qualification
- Key features: reuse of integer register file, short SIMD unit (8-bit), swizzling, reductions
- Intrinsic-like software support similar to ARM's NEON



The METASAT RISC-V Platform

- Mixed Criticality Platform
- FPGA Prototype on a Xilinx VCU 118
- Configurable Vortex RISC-V GPU
 - Enhancements for real-time execution and reliability
 - Qualifiable software stack for tasks requiring very high performance
 - Enable the use of GPUs from bare metal, or RTOS
 - Share the GPU among partitions
- The hardware platform will be open sourced as well as much of its software



The METASAT RISC-V Platform: Current Status

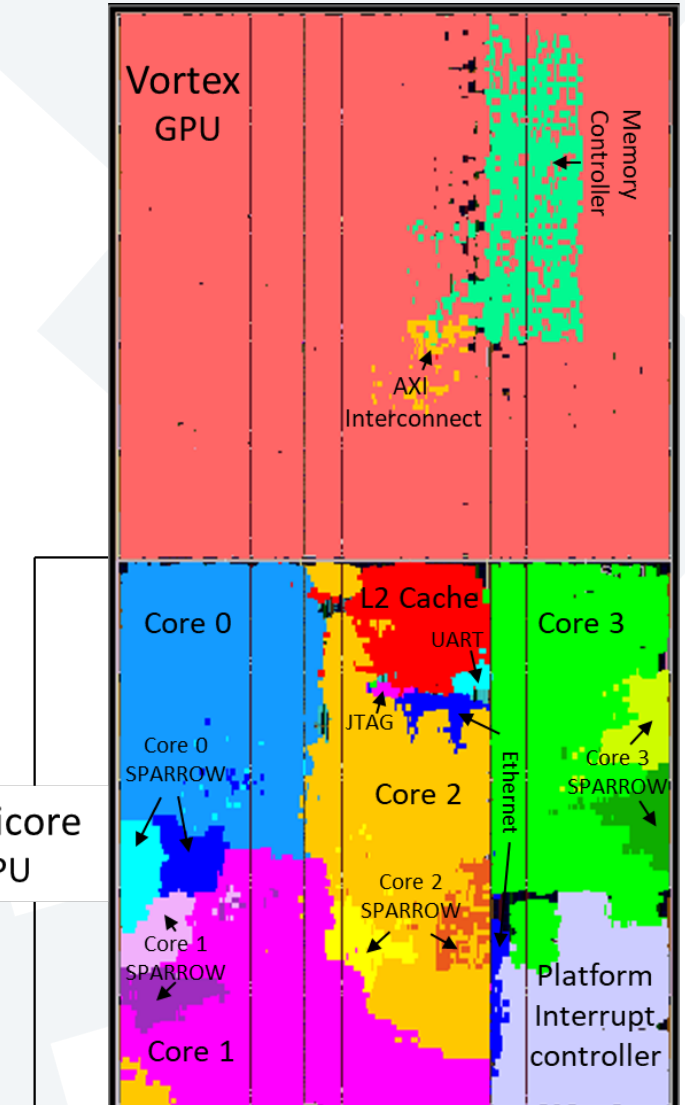


- NOEL-V Integration with Vortex GPU
 - AXI interface added to Vortex
 - Started a simple experiment offloading a GPU kernel to the GPU
 - Established a programming methodology for using OpenCL in the METASAT platform Precompile a GPU kernel
 - Common practice in safety critical systems (OpenGL SC, Vulkan SC)
 - Include the kernel binary in the program executable in a dedicated GPU memory area
 - No filesystem
 - Linker script modifications

The METASAT RISC-V Platform: Current Status



- FPGA Resource utilisation
- Current configuration:
 - 4 NOEL-V high performance + 2 SPARROW accelerators: 48% utilisation
 - To include also L2 cache L2Lite
 - GPU: 4 CUs, 4 threads each, 64bit L2 GPU cache 50% Utilisation
 - Once the design is fully functional a design space exploration will be performed to find the best configuration for the project use cases



The METASAT RISC-V Platform: Current Status



- Able to run OpenMP programs on both FPGA and QEMU under RTEMS
- See [1] for performance results and comparison with other multicore architectures
- On going work to support SPARROW in RTEMS
 - RTEMS Compiler modifications completed
 - Support for SPARROW control register to be added to RTEMS

```
unet32alpha1_0_upsample -- root@a3de48b...
...ogin1:~ -- -zsh ...
...olo_v8 -- -zsh ...
...ogin1:~ -- -zsh ...
...jwolf@caos17 ...
...nloads -- -zsh

[build-riscv-rtems6-rv64imafdc] finished successfully (0.802s)
[root@a3de48bca934:/workspace/app/matrix_multiplication_bench# ./sim-qemu

OPENMP DISPLAY ENVIRONMENT BEGIN
_OPENMP = '201511'
OMP_DYNAMIC = 'FALSE'
OMP_NUM_THREADS = '4'
OMP_SCHEDULE = 'DYNAMIC'
OMP_PROC_BIND = 'FALSE'
OMP_PLACES = ''
OMP_STACKSIZE = '0'
OMP_WAIT_POLICY = 'PASSIVE'
OMP_THREAD_LIMIT = '4294967295'
OMP_MAX_ACTIVE_LEVELS = '1'
OMP_NUM_TEAMS = '0'
OMP_TEAMS_THREAD_LIMIT = '0'
OMP_CANCELLATION = 'FALSE'
OMP_DEFAULT_DEVICE = '0'
OMP_MAX_TASK_PRIORITY = '0'
OMP_DISPLAY_AFFINITY = 'FALSE'
OMP_AFFINITY_FORMAT = 'level %L thread %i affinity %A'
OMP_ALLOCATOR = 'omp_default_mem_alloc'
OMP_TARGET_OFFLOAD = 'DEFAULT'
GOMP_CPU_AFFINITY = ''
GOMP_STACKSIZE = '0'
GOMP_SPINCOUNT = '30000'
OPENMP DISPLAY ENVIRONMENT END
Creating and starting an application task
bla
bla
Application task was invoked with argument and has id of 0x00000
4
4
s
1024
Using device: Generic device
Elapsed time Host->Device: 0.000000000 milliseconds
Elapsed time kernel: 84672.781250000 milliseconds
Elapsed time Device->Host: 0.000000000 milliseconds

[ RTEMS shutdown ]
CPU: 2
RTEMS version: 6.0.0.3612dc7d61d91e0bc121b2d226a1b3082ff9e333
RTEMS tools: 12.2.1 20230224 (RTEMS 6, RSB bfed51462eafcb6a5102a2d6d80b233f3c6ef635, Newlib 17ac400)
executing thread ID: 0xa010002
executing thread name: TA1
*** FATAL ***
```

[1] M. Solé, J. Wolf, I. Rodriguez, A. Jover, M. M. Trompouki, L Kosmidis, D. Steenari. Evaluation of the Multicore Performance Capabilities of the Next Generation Flight Computers. Digital Avionics Systems Conference (DASC) 2023

The METASAT RISC-V Platform: Current Status



- Preliminary Results

```
max number of threads in openmp using omp_get_max_threads() is 4
Block size: 1024
base 1core: 392.225896 s
transposed 1core: 86.701770 s - valid 4.4x
omp 4cores: 280.271344 s - valid
omp transposed 4core: 19.934385 s - valid 4.5x
sparrow 1core: 20.878695 s - expected error (non transposed sparrow)
sparrow transposed 1core: 19.400293 s - valid 15.3x
sparrow omp 4core: 6.340939 s - expected error (non transposed sparrow)
sparrow omp transposed 4core: 5.664588 s - valid
```

- SPARROW on a single core NOEL-V provides similar performance with a 4-core OpenMP implementation
- 15x overall speedup by using both multicore and SPARROW

Virtual The METASAT RISC-V Platform



- METASAT multicore CPU to be modeled in QEMU and SIS
 - Add also support for SPARROW
- Vortex GPU to be simulated in Verilator
 - Cycle-accurate behavioural simulation
 - SystemVerilog to SystemC/C++
- Work started with GR740 and GR712RC models
 - Accurate modeling of the GR740 GRFPU
 - Cannot handle subnormal numbers as input, raise *unfinished_Fpop* exception
 - Handling denormalized numbers with the GRFPU
 - <https://www.gaisler.com/doc/antn/GRLIB-AN-0007.pdf>
 - Can boot unmodified Linux and RTEMS binaries

```
lkosmidi@Caos17:~/scratch/lkosmidi/rtems$ qemu_inst/bin/qemu-syst
em-sparc -machine gr740 -serial stdio -kernel ~/rcc-1.3.0-gcc/rc
c-1.3.0-gcc/src/samples/rtems-shell_gr740
VNC server running on 127.0.0.1:5900
--- BUS TOPOLOGY ---
-> DEV 0xd52b8 GAISLER_LEON4
-> DEV 0xd5320 GAISLER_APBMS1
-> DEV 0xd5388 GAISLER_IRQMP
-> DEV 0xd53f0 GAISLER_GPTIMER
-> DEV 0xd5458 GAISLER_APBUART

You can use the shell commands drvmgr and pci to find out
more about the system

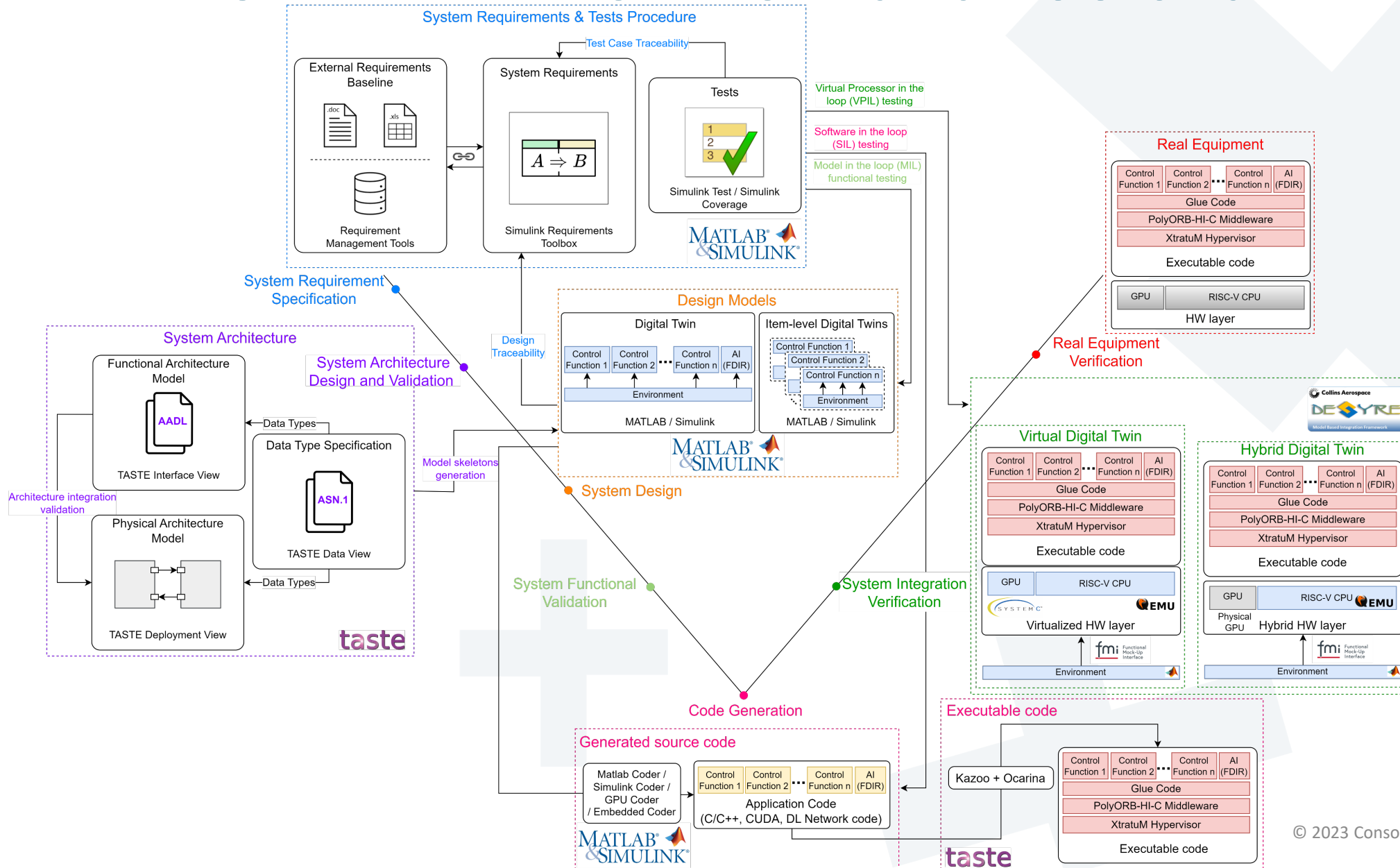
Creating /etc/passwd and group with three useable accounts
root/pwd , test/pwd, rtems/NO PASSWORD

RTEMS Shell on dev/console. Use 'help' to list commands.
SHLL [ / ] #
```

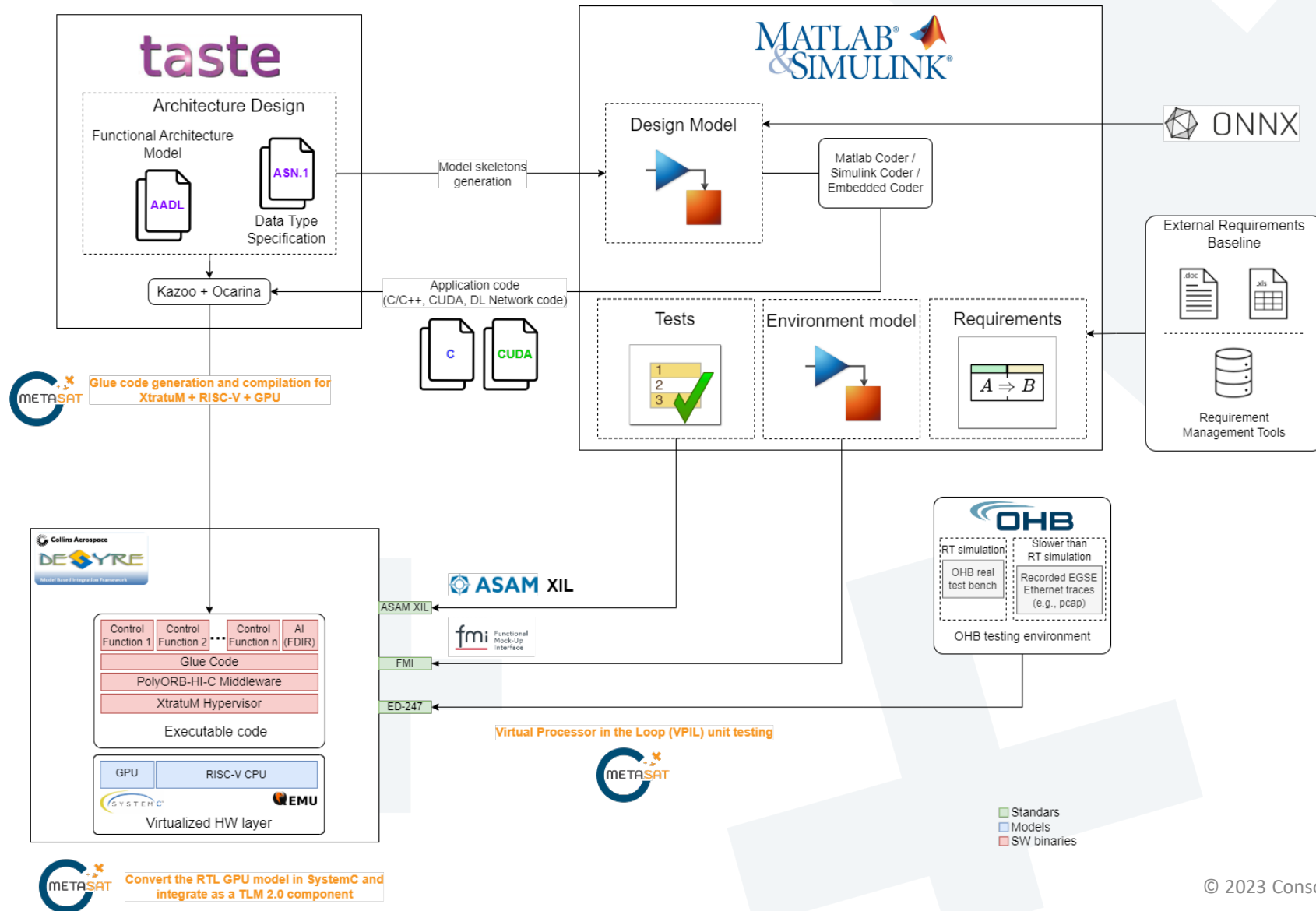
```
Segment Routing with IPv6
sit: IPv6, IPv4 and MPLS over IPv4 tunneling driver
NET: Registered protocol family 17
Key type dns_resolver registered
Freeing unused kernel memory: 5760K
This architecture does not have kernel memory protection.
Run /init as init process
Starting syslogd: OK
Starting klogd: OK
Running sysctl: OK
Saving random seed: random: dd: uninitialized urandom read (32 bytes read)
OK
Starting network: OK
random: crng init done
ssh-keygen: generating new host keys: RSA DSA ECDSA ED25519
Starting sshd: OK

Welcome to Buildroot
buildroot login:
0$ bash 1$ bash 2$ bash 3-$ bash
```

METASAT MBE Workflow and Toolchain



METASAT Toolchain Data Flow

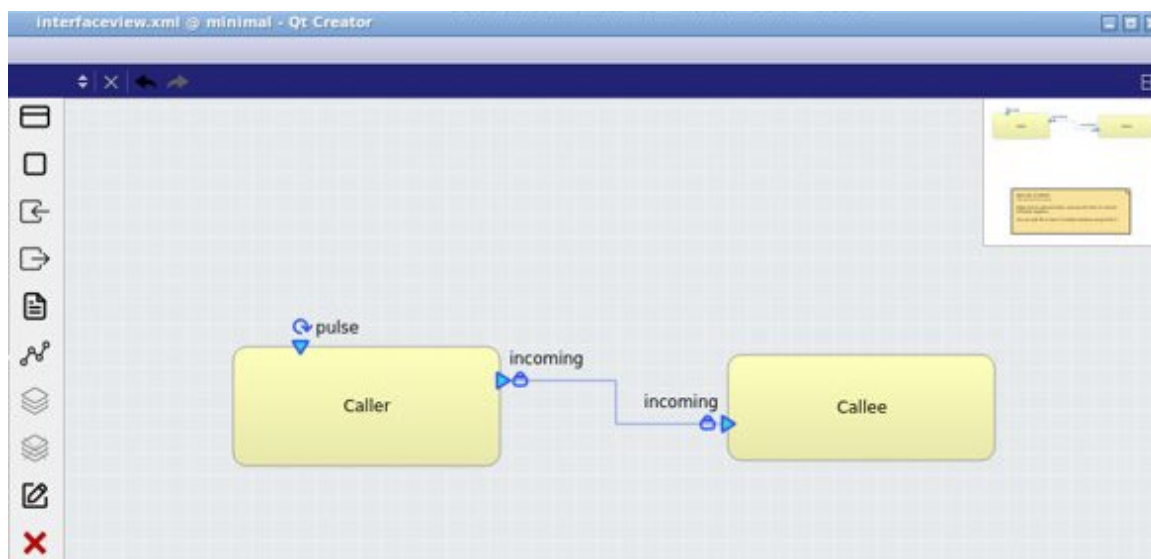


RISC-V Support in TASTE

RISC-V support added in TASTE

- RTEMS
- RISC-V compiler
- QEMU RISC-V Simulator

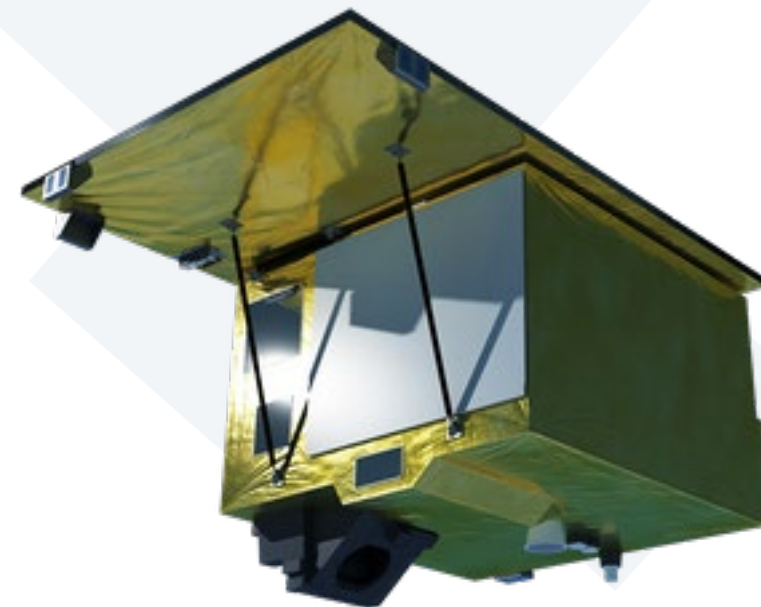
Minimal communication example:



```
taste@taste10 ~/communication_example/work/binaries
$ taste-simulate-leon3 partition_1_leon3_rtems.exe
[Caller] Startup
[Callee] Startup
Send a=1
Received b=2
Send a=2
Received b=3
Send a=3
Received b=4
Send a=4
Received b=5
Send a=5
Received b=6
Send a=6
Received b=7
```

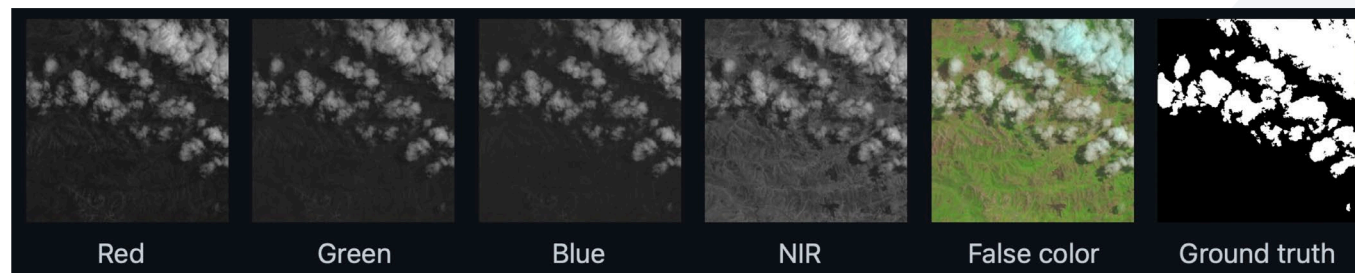
Project Use Cases

- 3 Project Use cases will be implemented
- OHB/DLR Use Case
 - Hardware interlocking
 - Protect against wrong software behaviour
 - Implement interlocks at software level instead of hardware
 - Reduce cost
 - Implement AI Based FDIR
 - To be accelerated on the CPU using the SPARROW AI accelerator
 - Housekeeping data from ENMAP



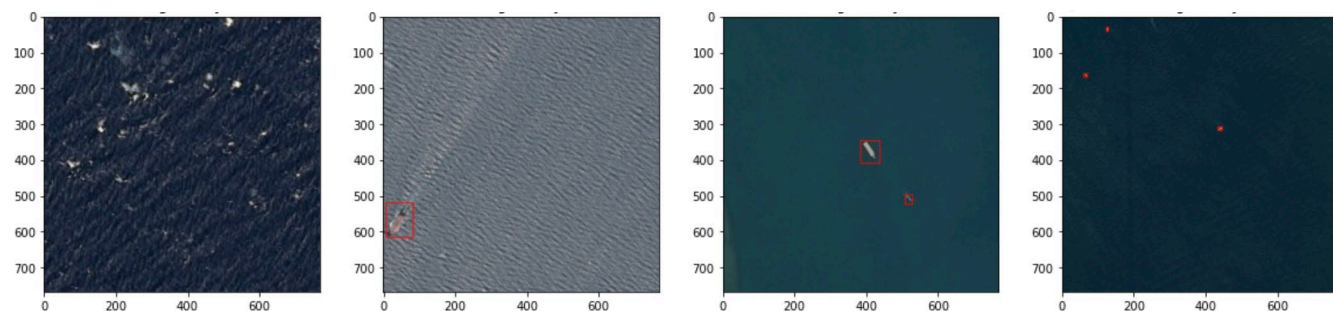
Project Use Cases

- 2 BSC provided use cases based on ESA's OBPMark-ML Open Source Benchmarking suite
- Cloud screening



4 Channels RGB/NIR mapped to binary mask (cloud/no cloud)

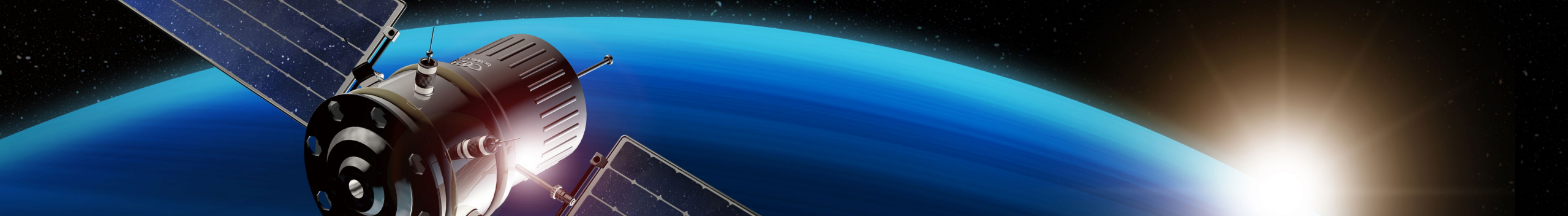
- Ship Detection



- To be executed on the GPU

Conclusion

- METASAT will achieve a major milestone towards the use of GPUs and high performance platforms in space
- Will provide an open source reference hardware platform
 - FPGA and virtual
 - Possible starting point for a future GPU tape out on a radiation tolerant/hardened process
- Solve key limitations preventing GPUs to be adopted today in institutional missions
 - Qualifiable software stack
- Model-Based design methodology supporting TASTE
- Evaluation to be performed with relevant space use cases
- Promising early results



<https://metasat-project.eu/>
info@metasat-project.eu



<https://twitter.com/MetasatProject>

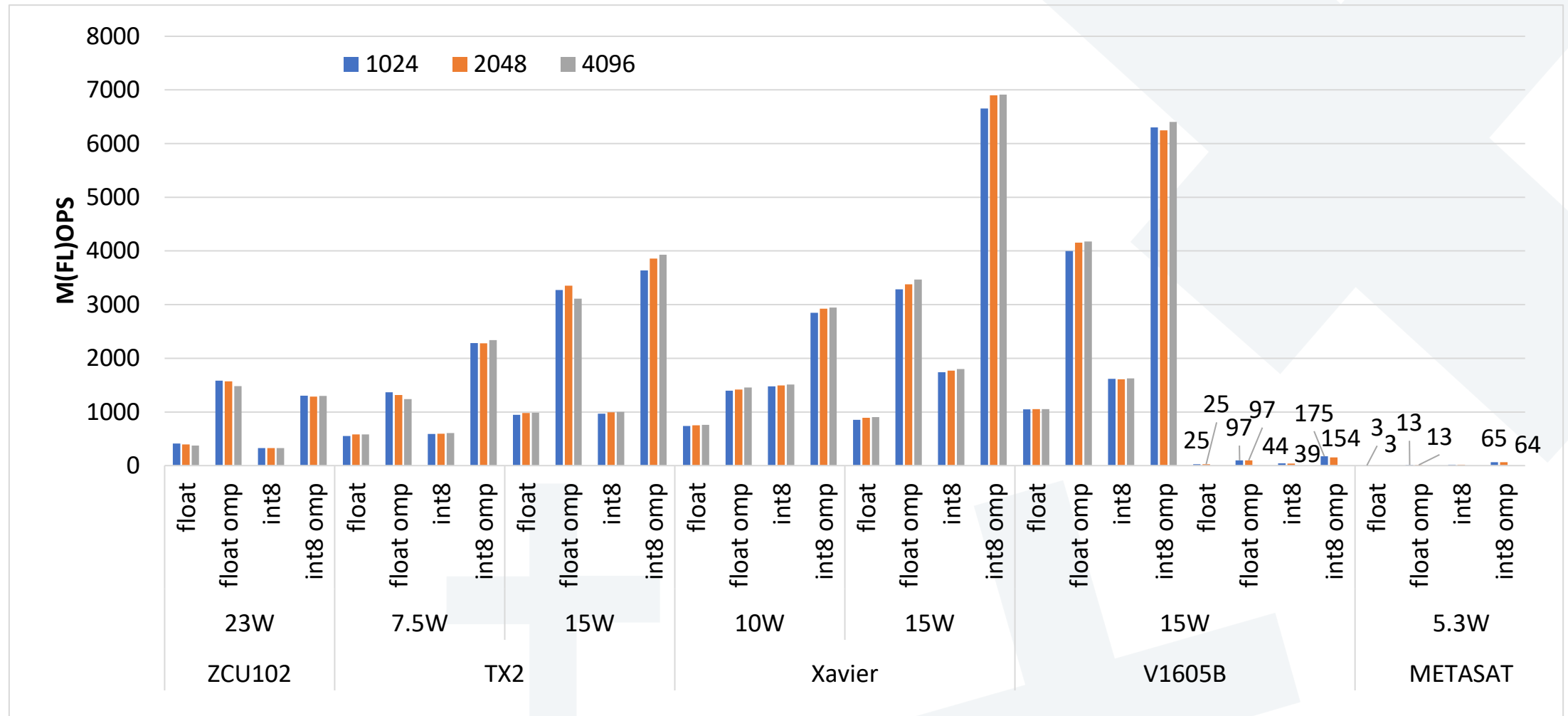


<https://www.linkedin.com/company/metasat-project>



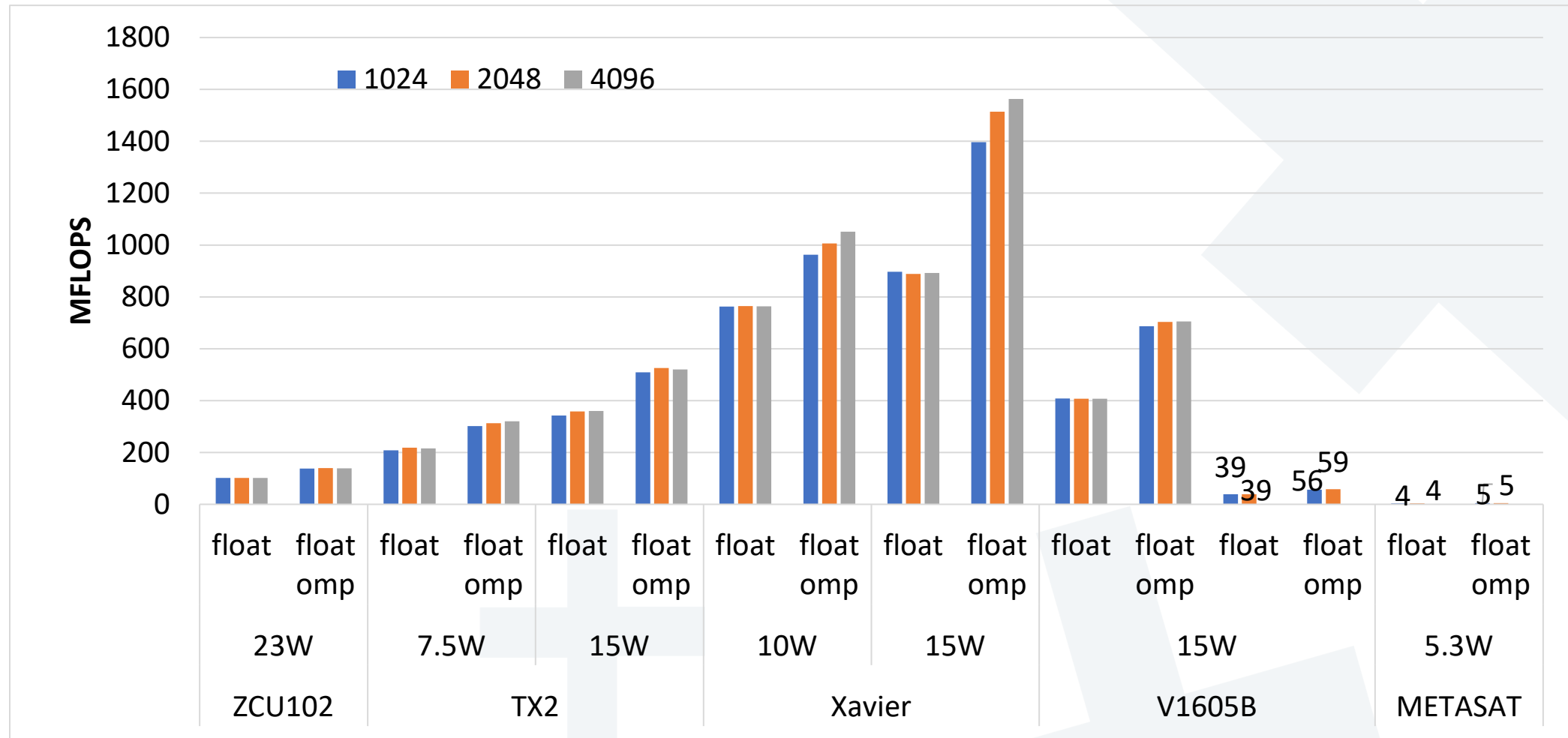
METASAT has received funding from the European Union's Horizon Europe programme under grant agreement number 101082622.

Evaluation: Matrix Multiplication from [1]



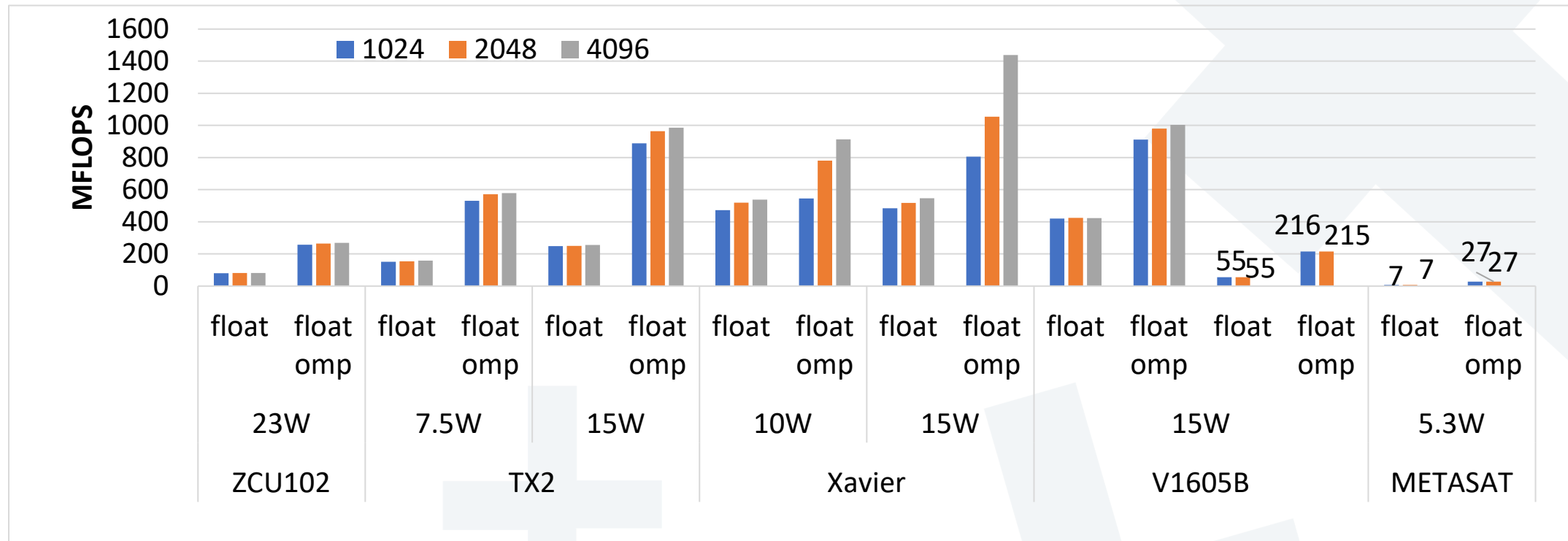
[1] M. Solé et al. Evaluation of the Multicore Performance Capabilities of the Next Generation Flight Computers.

Evaluation: 2D Correlation from [1]



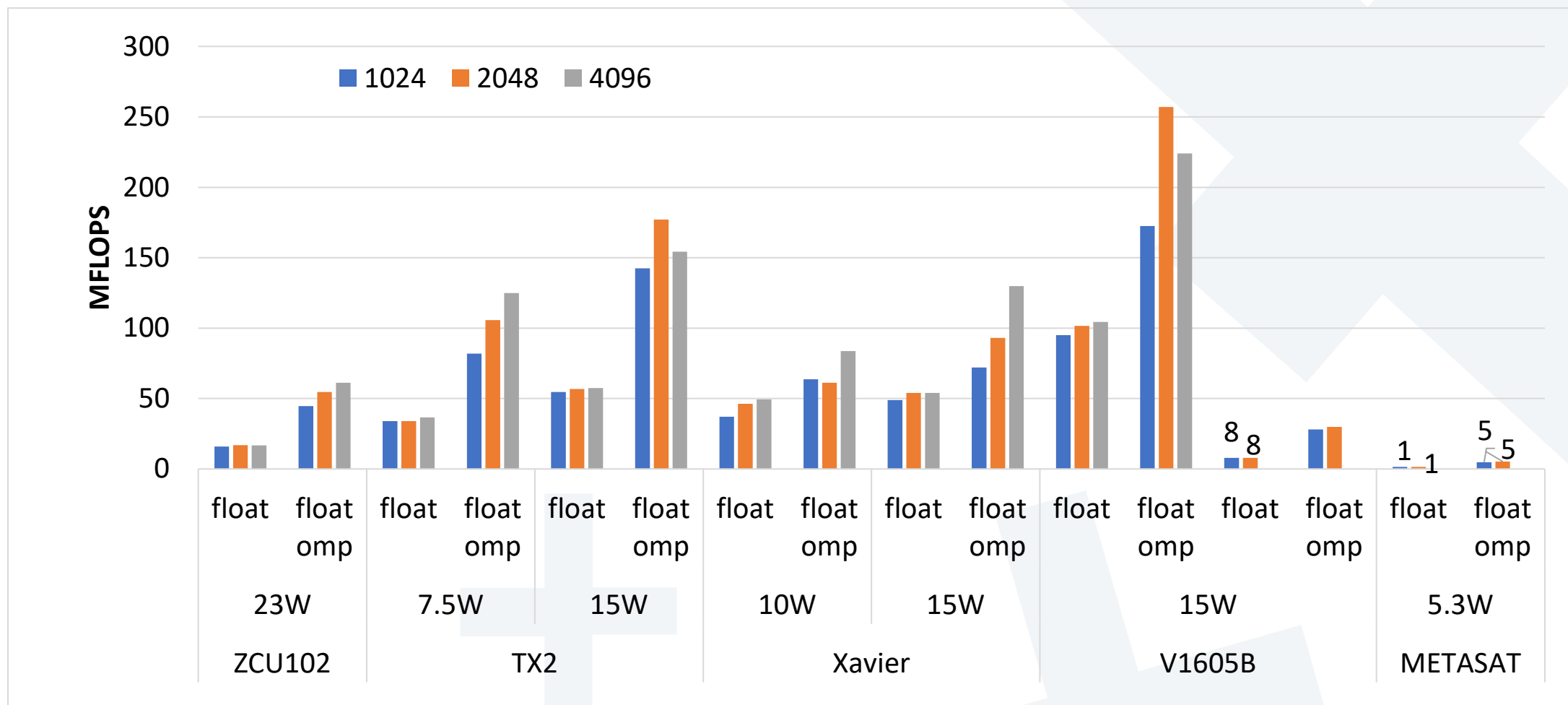
[1] M. Solé et al. Evaluation of the Multicore Performance Capabilities of the Next Generation Flight Computers.

Evaluation: Sliding FFT from [1]



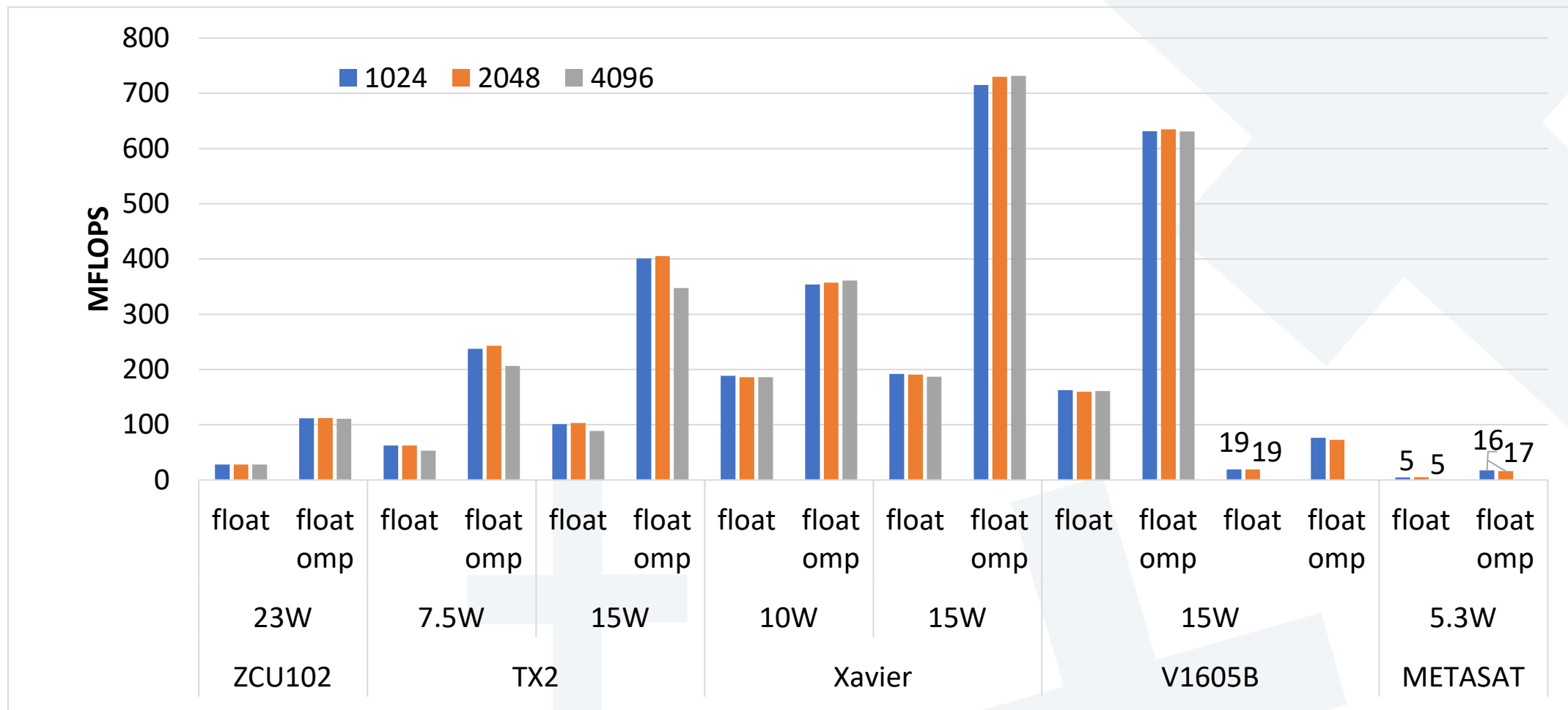
[1] M. Solé et al. Evaluation of the Multicore Performance Capabilities of the Next Generation Flight Computers.

Evaluation: FIR from [1]



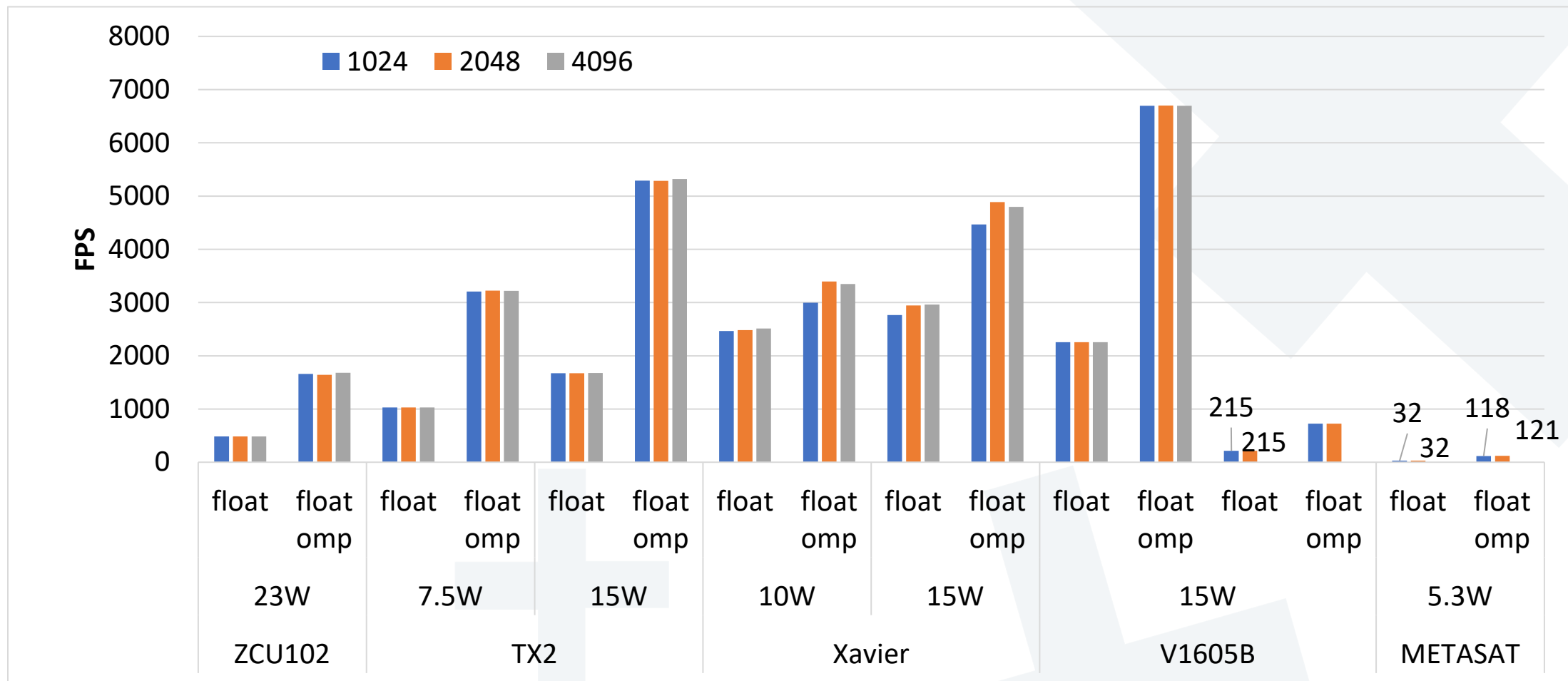
[1] M. Solé et al. Evaluation of the Multicore Performance Capabilities of the Next Generation Flight Computers.

Evaluation: 2D Convolution from [1]



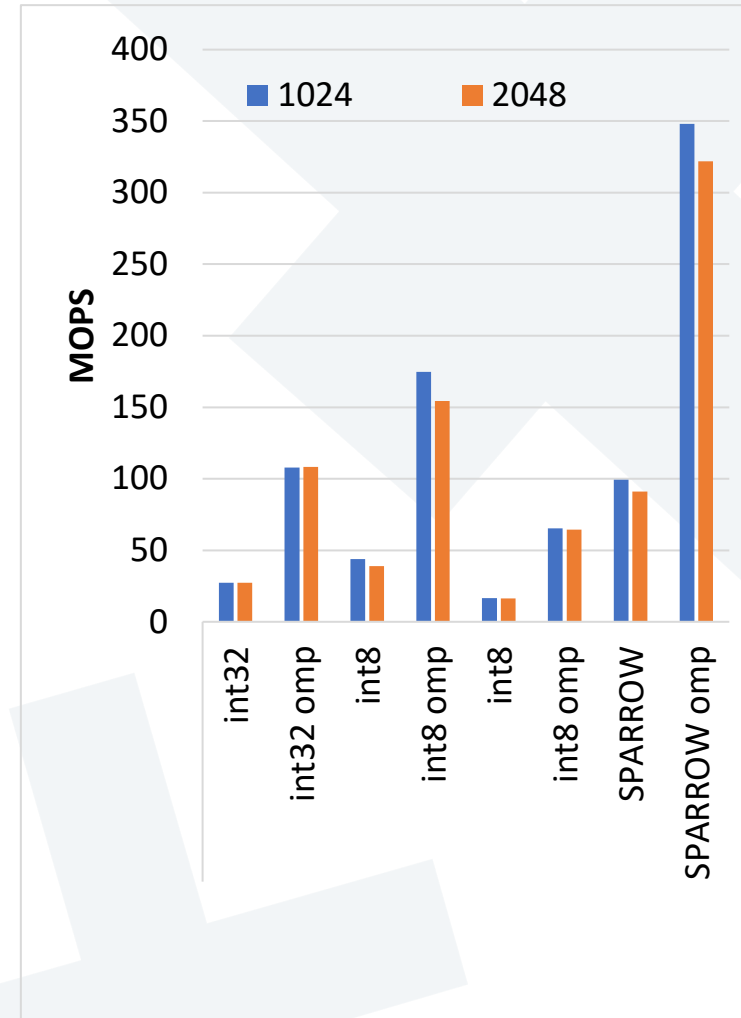
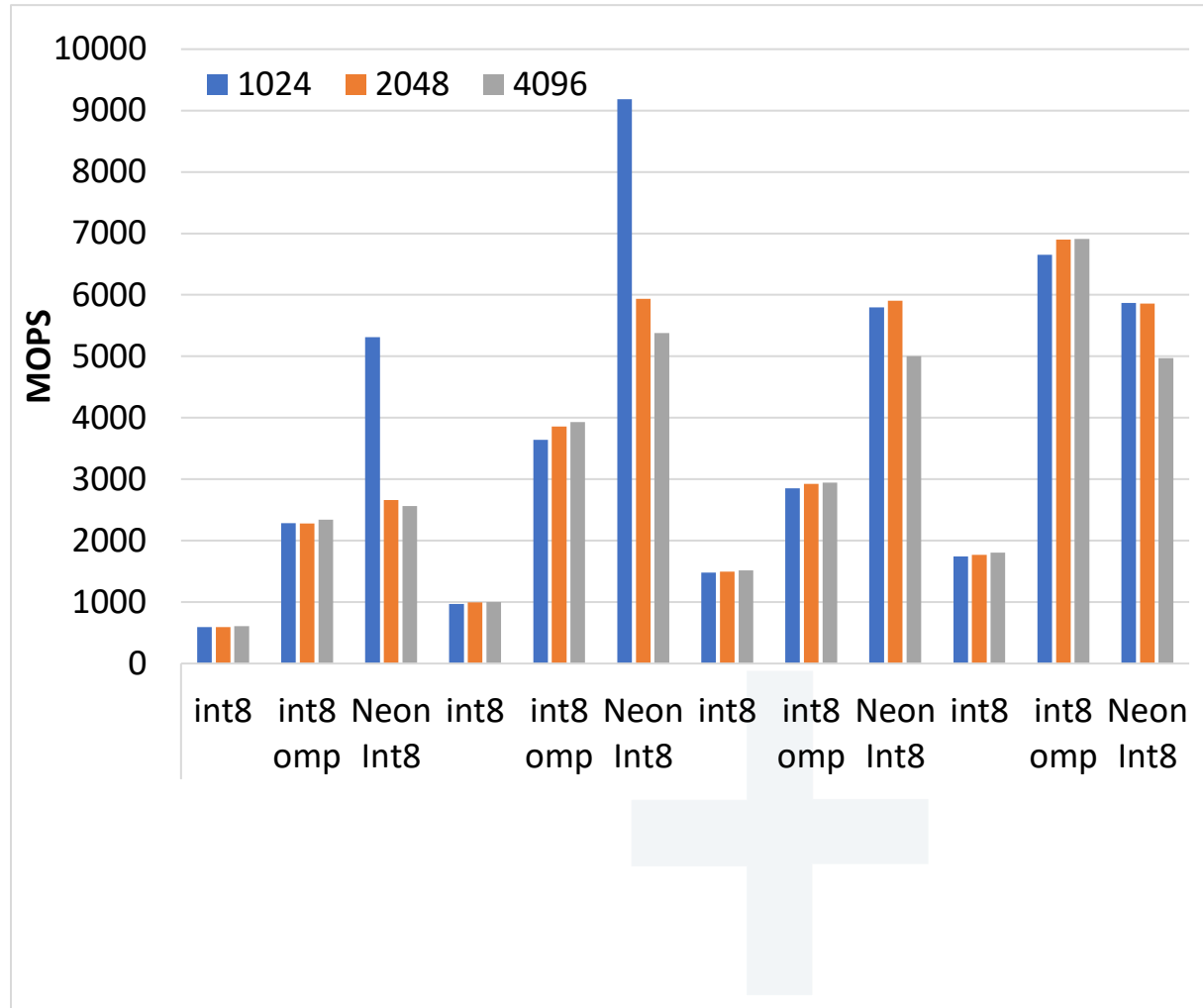
[1] M. Solé et al. Evaluation of the Multicore Performance Capabilities of the Next Generation Flight Computers.

Evaluation: CIFAR-10 Inference from [1]



[1] M. Solé et al. Evaluation of the Multicore Performance Capabilities of the Next Generation Flight Computers.

Evaluation: SIMD Matrix Multiplication [1]



[1] M. Solé et al. Evaluation of the Multicore Performance Capabilities of the Next Generation Flight Computers.