

Research and implementation challenges of RTOS support for heterogeneous computing platforms

TULIPP European project

Martin Cornil
Antonio Paolillo, Prof. Joël Goossens, Ben Rodriguez

May 2017

HIPPEROS S.A.
Faculty of Computer Science, ULB

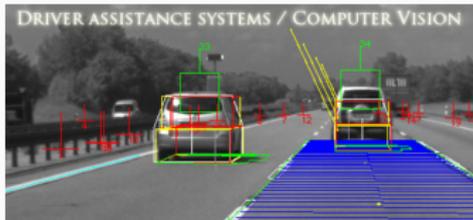
ULB



HIPPEROS®

What, why and how?

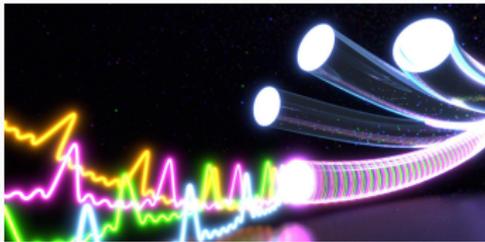
Tomorrow's embedded systems



Low-power embedded camera
e.g.: UAV's, self-driving cars,...



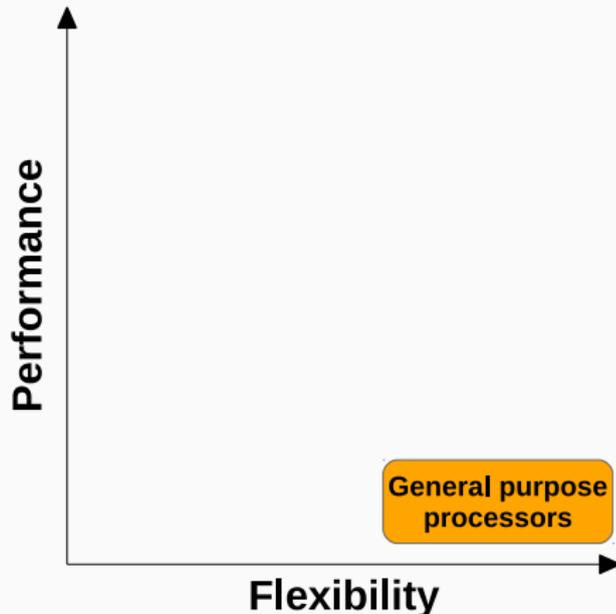
High resolutions multimedia
e.g.: 4K TV, real-time streaming,...



High bandwidth networks
e.g.: GPON, multi-path TCP,...

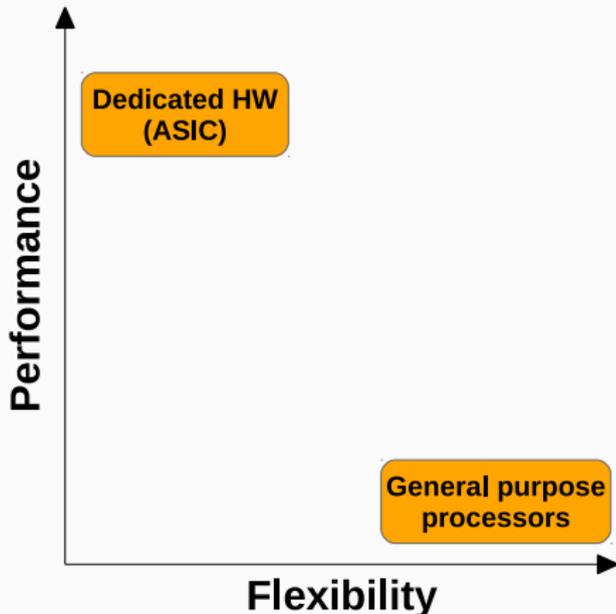
General purpose processors?

- ✓ Cost
- ✗ Computing power
- ✗ Energy consumption
- ✓ Flexibility
- ✓ Ease of use



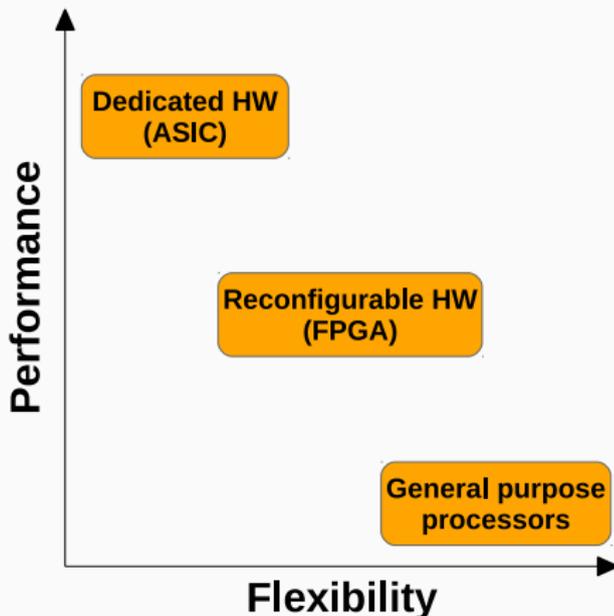
Dedicated HW (ASIC)?

- ✗ Cost
- ✓ Computing power
- ✓ Energy consumption
- ✗ Flexibility
- ✓ Ease of use



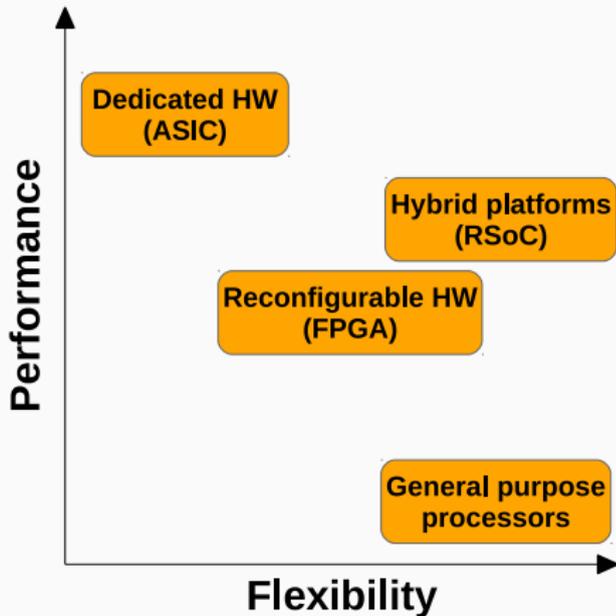
FPGA?

- ✓ Cost
- ✓ Computing power
- ✗ Energy consumption
- ✓ Flexibility
- ✗ Ease of use



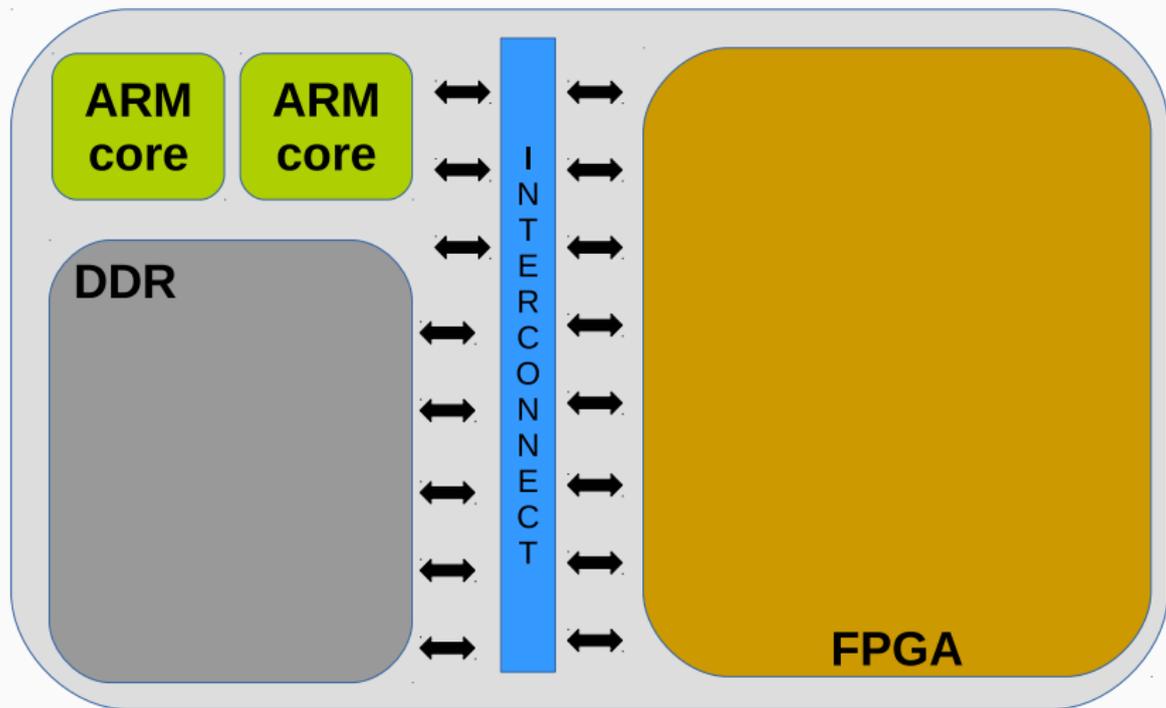
RSoC?

- ✓ Cost
- ✓ Computing power
- ✓ Energy consumption
- ✓ Flexibility
- ✗ Ease of use

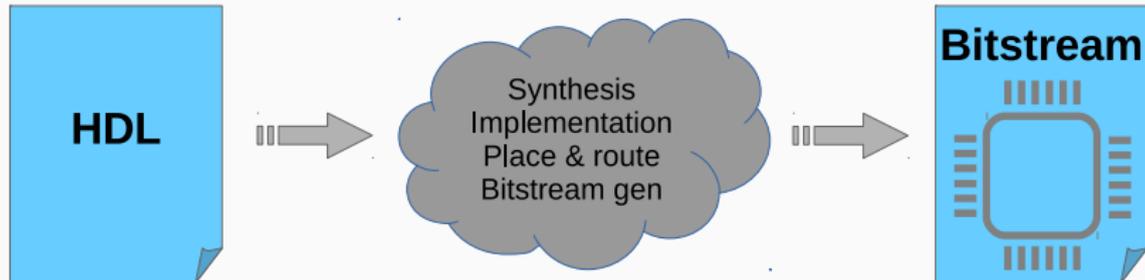


Reconfigurable System-On-Chip

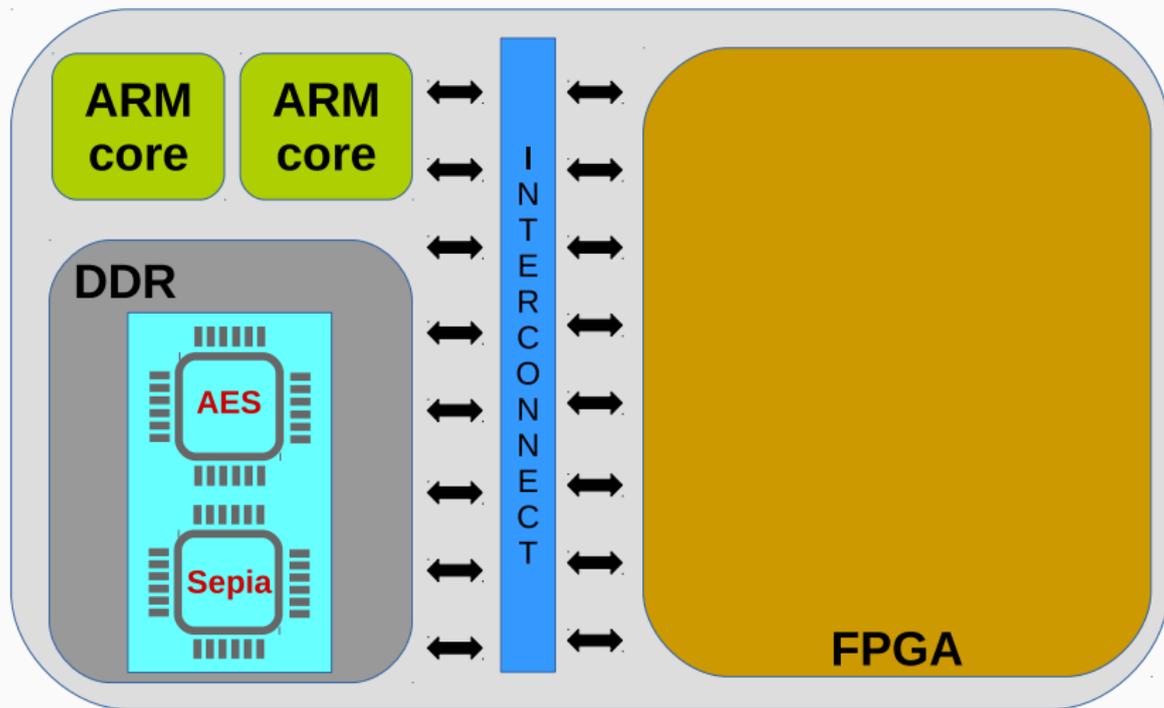
Reconfigurable System-On-Chip



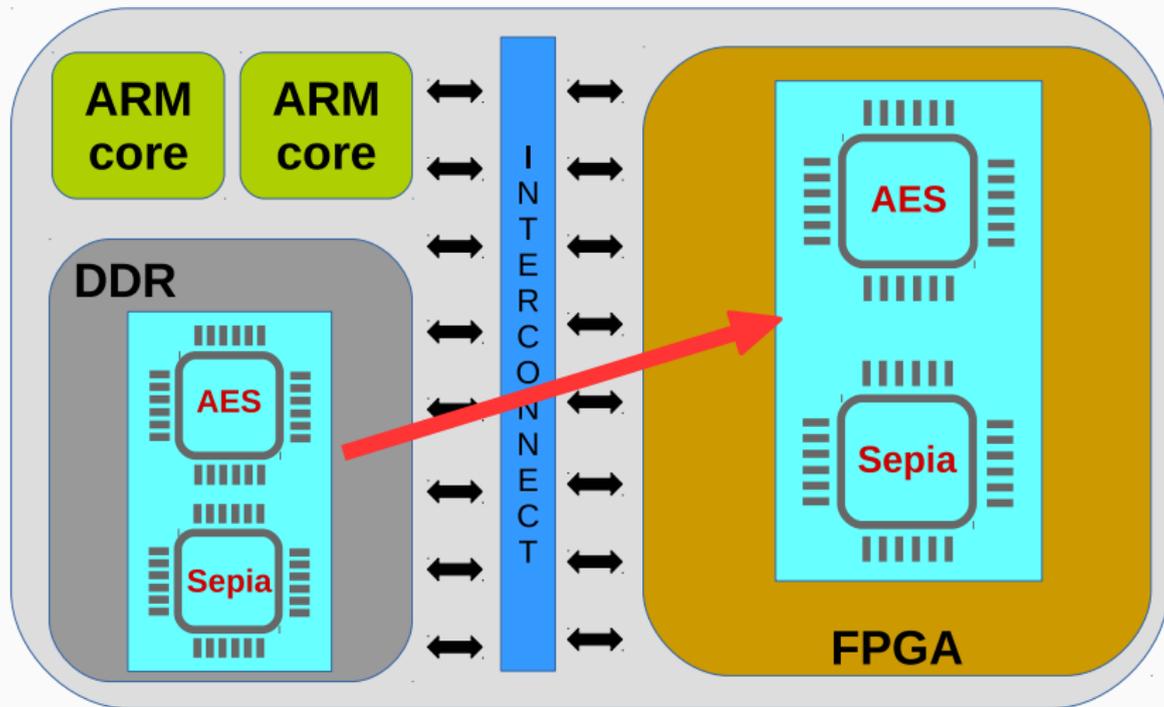
Bitstreams



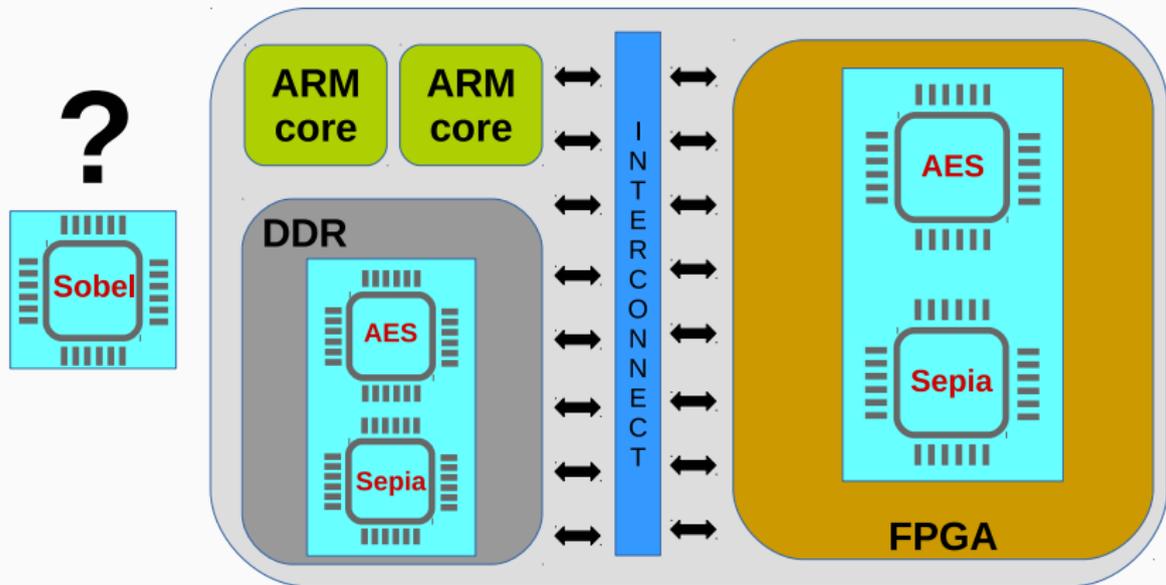
RSoC: fully static



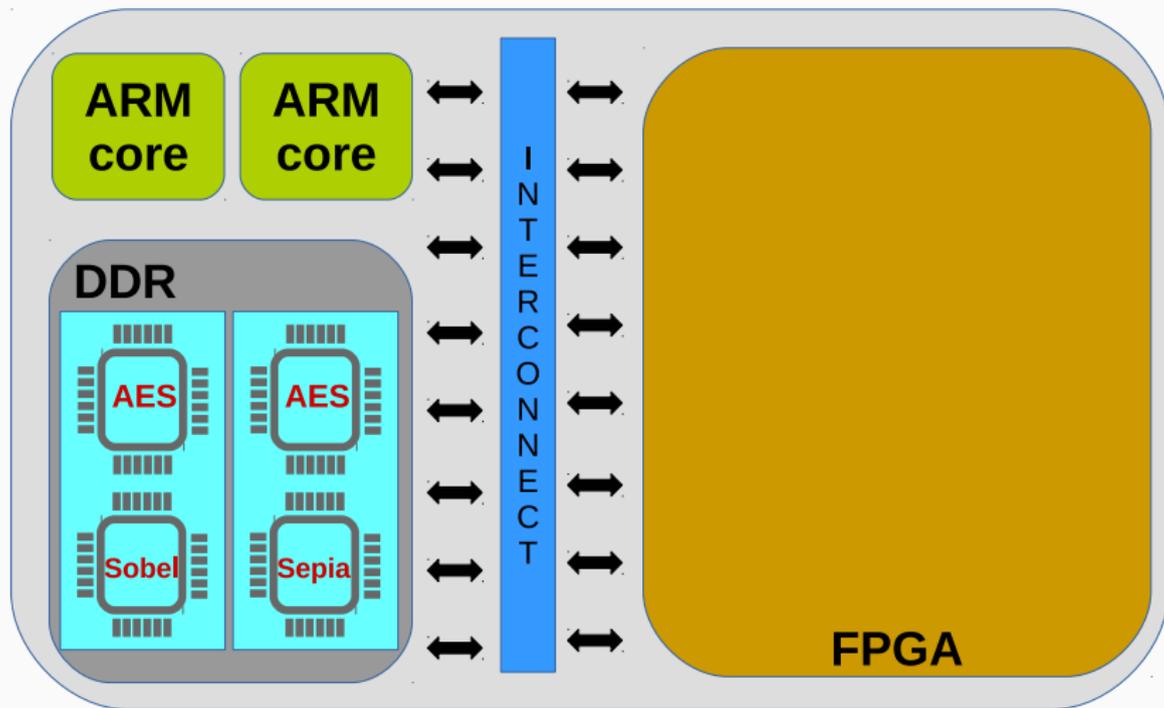
RSoC: fully static



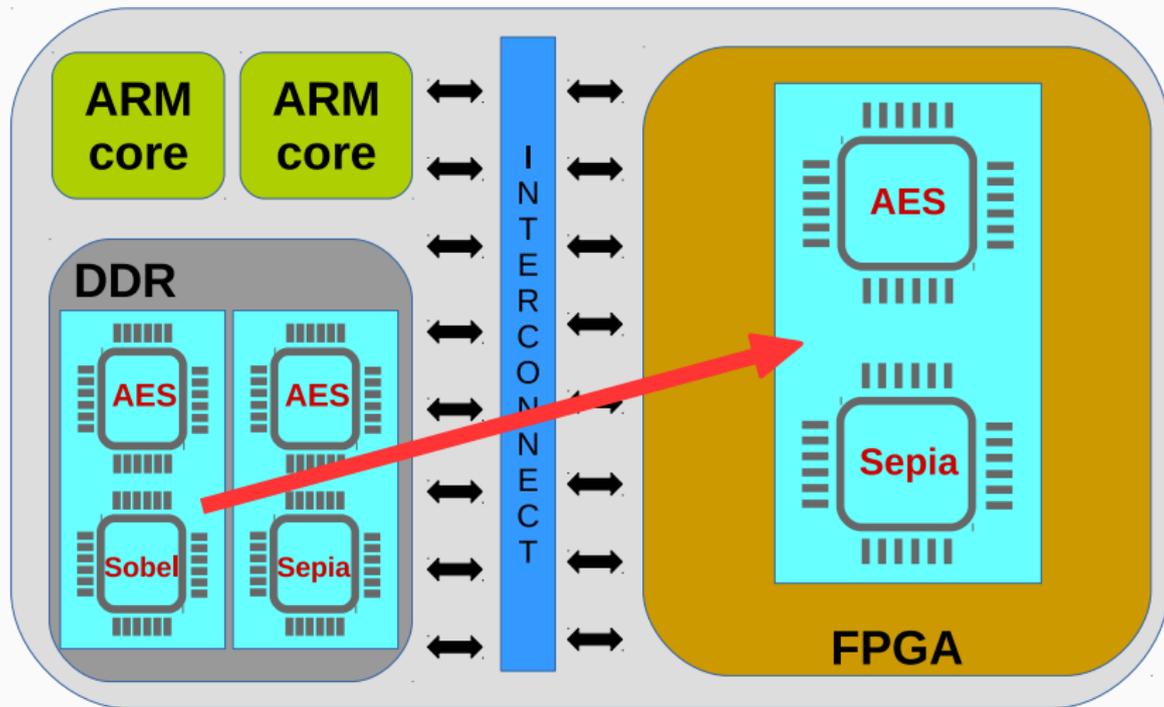
RSoC: normal flow problem



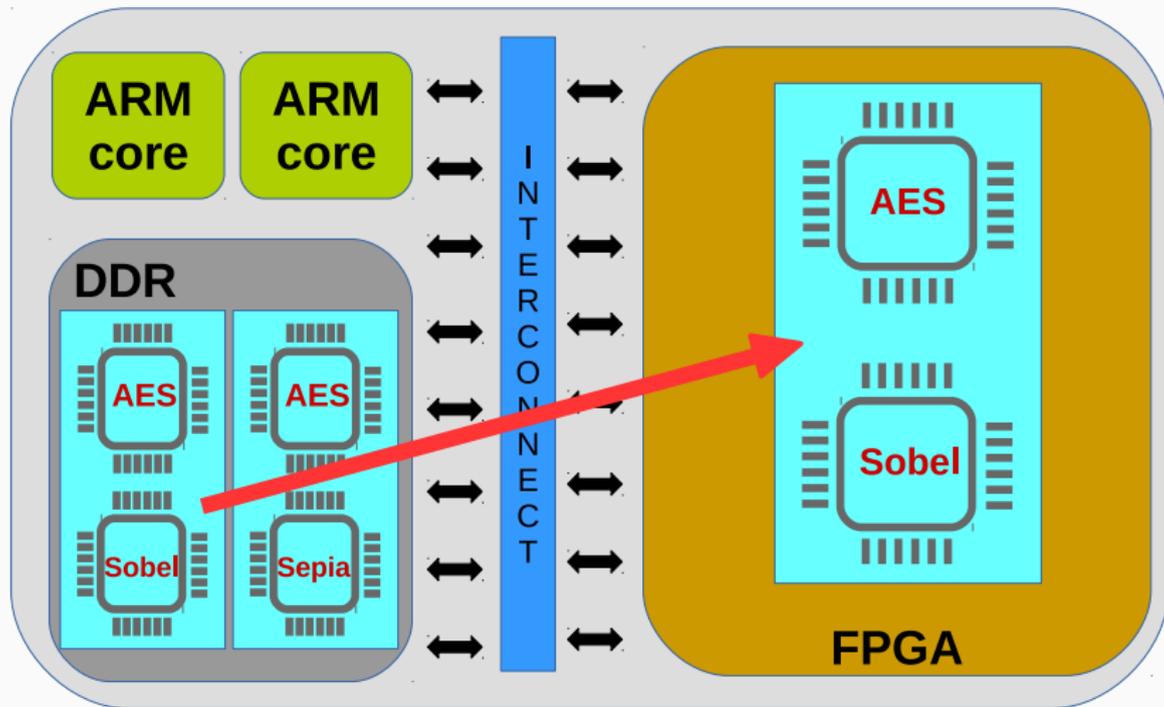
RSoC: 2 static configurations



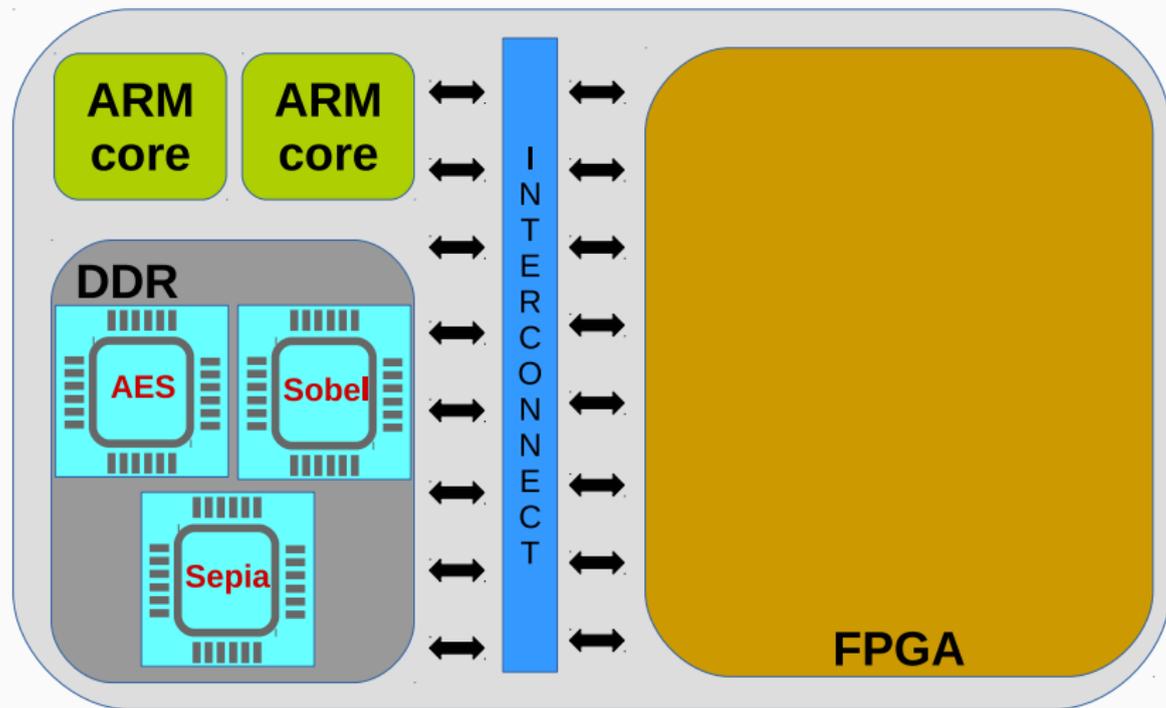
RSoC: configuration 1



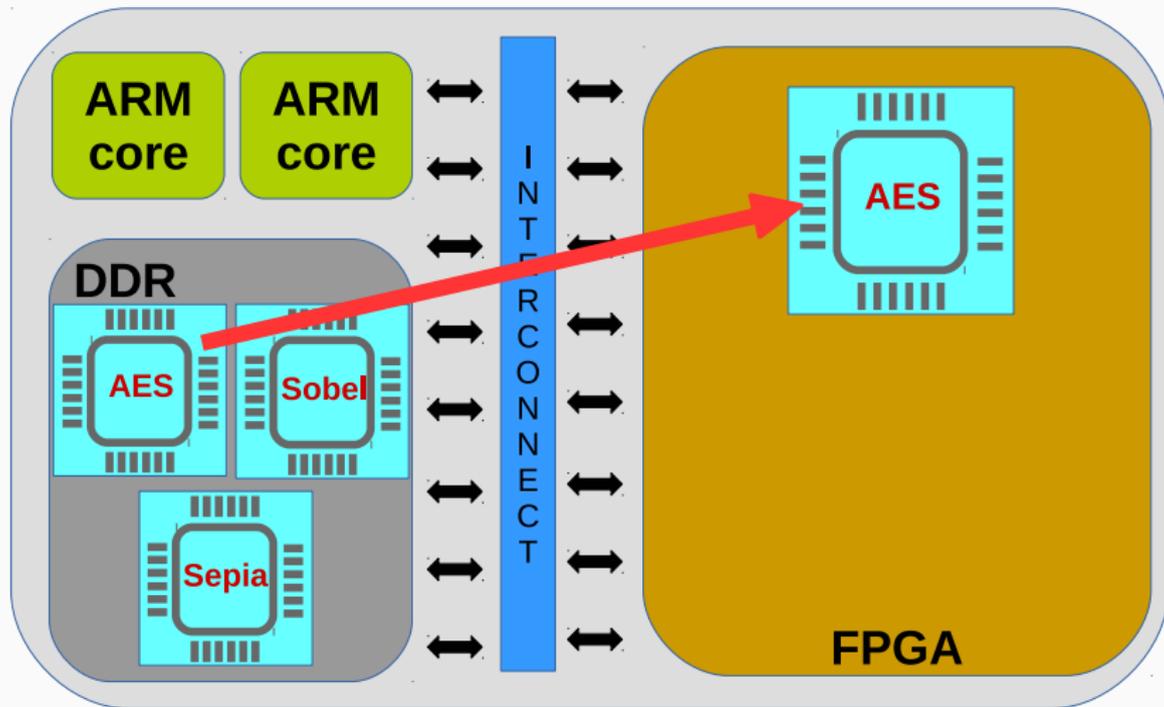
RSoC: configuration 2



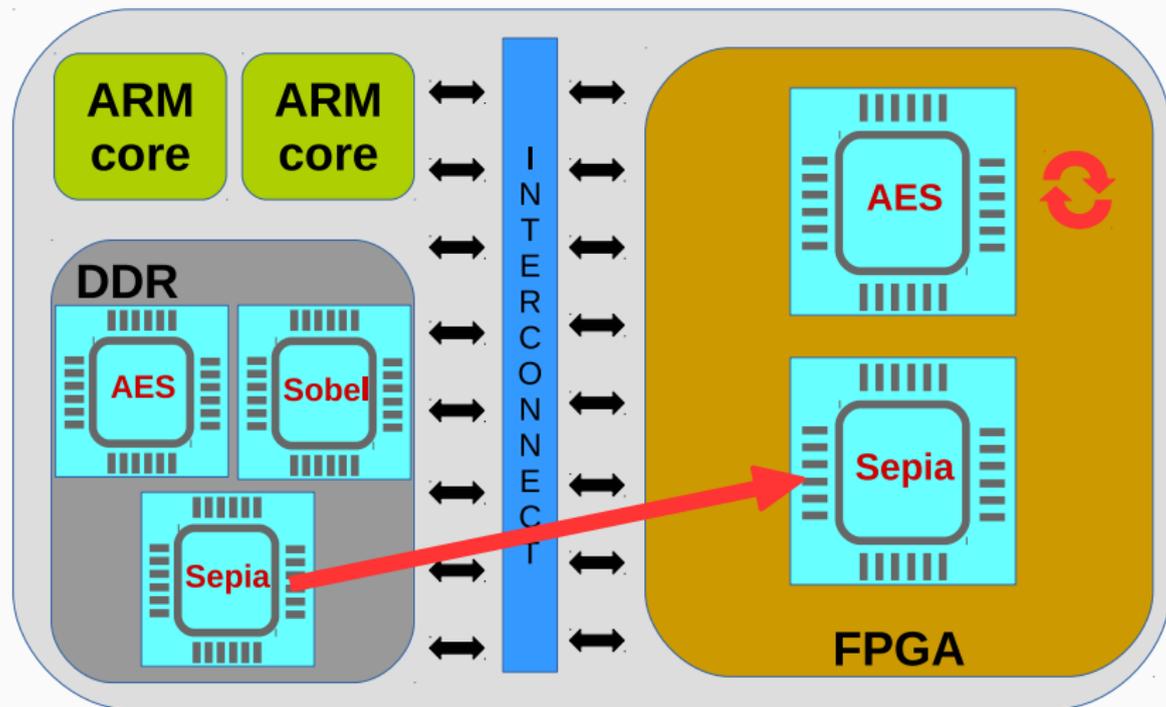
RSoC: Dynamic Partial Reconfiguration



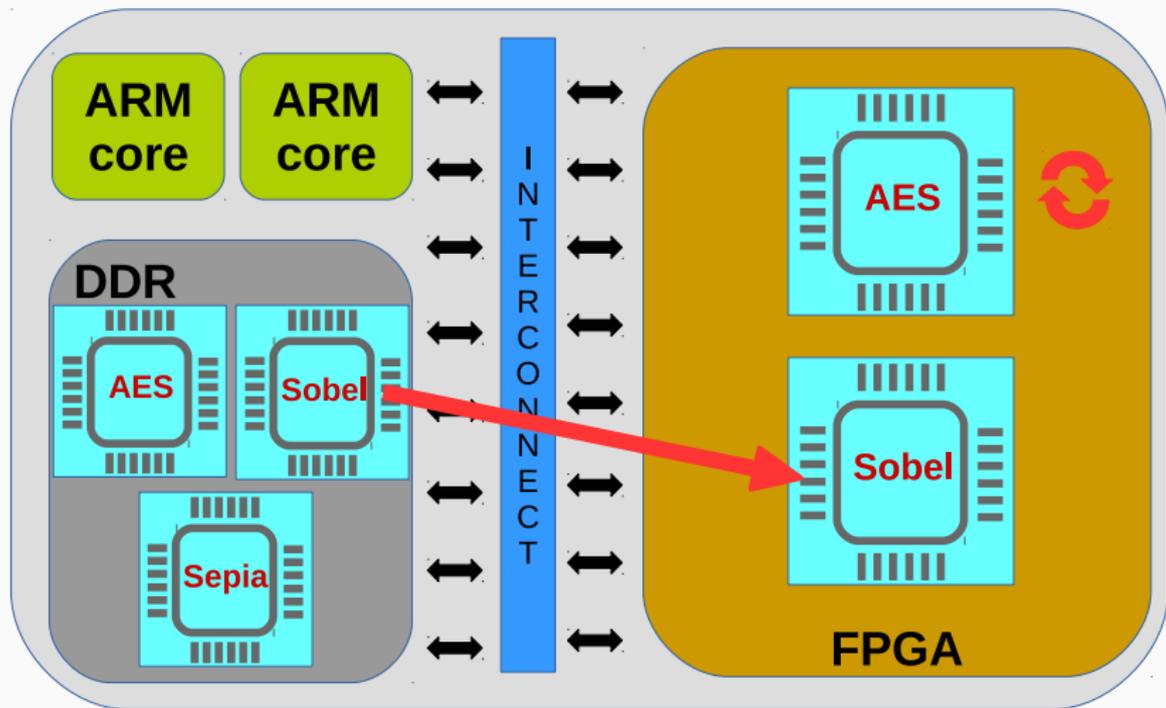
RSoC: Dynamic Partial Reconfiguration



RSoC: Dynamic Partial Reconfiguration



RSoC: Dynamic Partial Reconfiguration

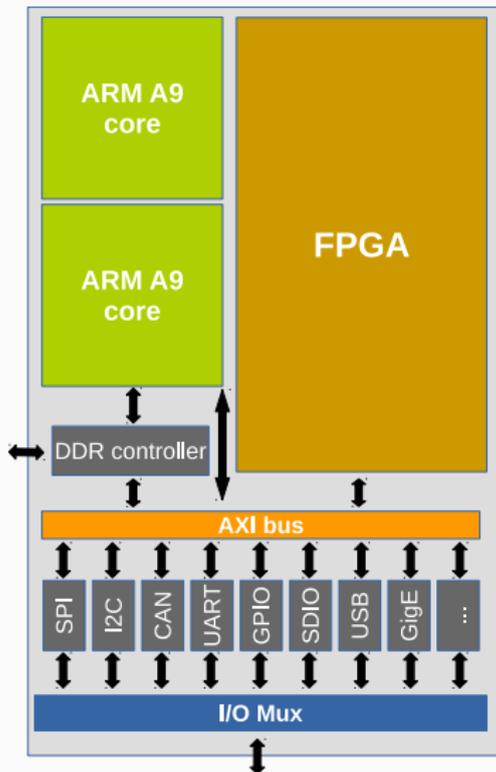


Reconfigurable System-on-Chip (RSoC) - Zynq-7000

Xilinx Zynq-7000 RSoC:

- dual-core ARM-A9 (32bits),
- Xilinx FPGA,
- a high speed interconnect bus,
- lot of peripherals

The FPGA is **partially reconfigurable at run-time !**



Applications requirements

Today's embedded applications:

- ✓ Must be safe/deterministic,
- ✓ Are multi-process,
- ✓ Need computing power,
- ✓ Must be energy efficient,
- ✓ Shared lot of resources (CPU, FPGA, memory,...),
- ✓ Use lot of peripherals,
- ✓ ...

We need a RTOS !

HIPPEROS

High **P**erformance **P**arallel **E**mbedded **R**eal-time **O**perating **S**ystem

- ✓ Microkernel architecture
- ✓ Hard real-time operating system
- ✓ Virtual memory isolation
- ✓ Multicore support
- ✓ Highly configurable



Contributions

Hardware

CPU

FPGA



Applications

User application

User application

Hardware

CPU

FPGA



Applications

User application

User application

Kernel

HIPPEROS

Hardware

CPU

FPGA



Applications

User application

User application

?

?

?

Kernel

HIPPEROS

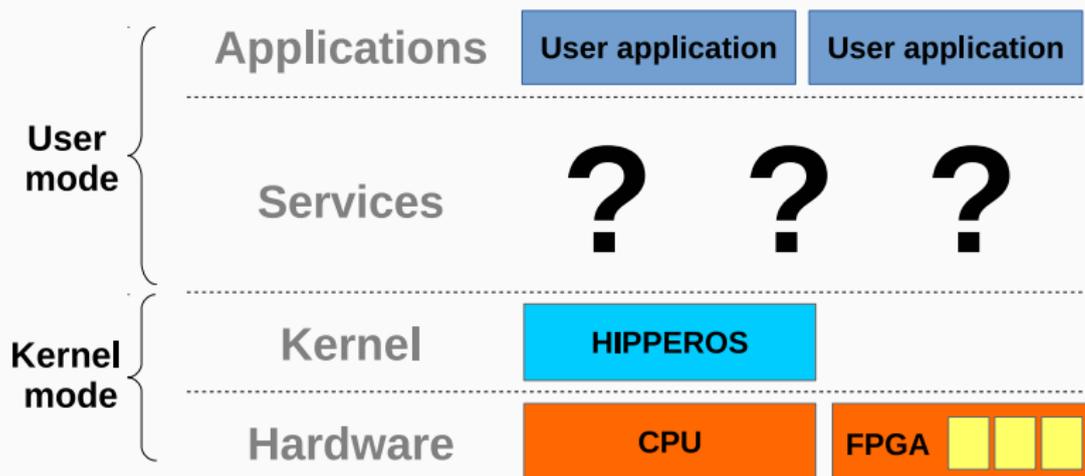
Hardware

CPU

FPGA

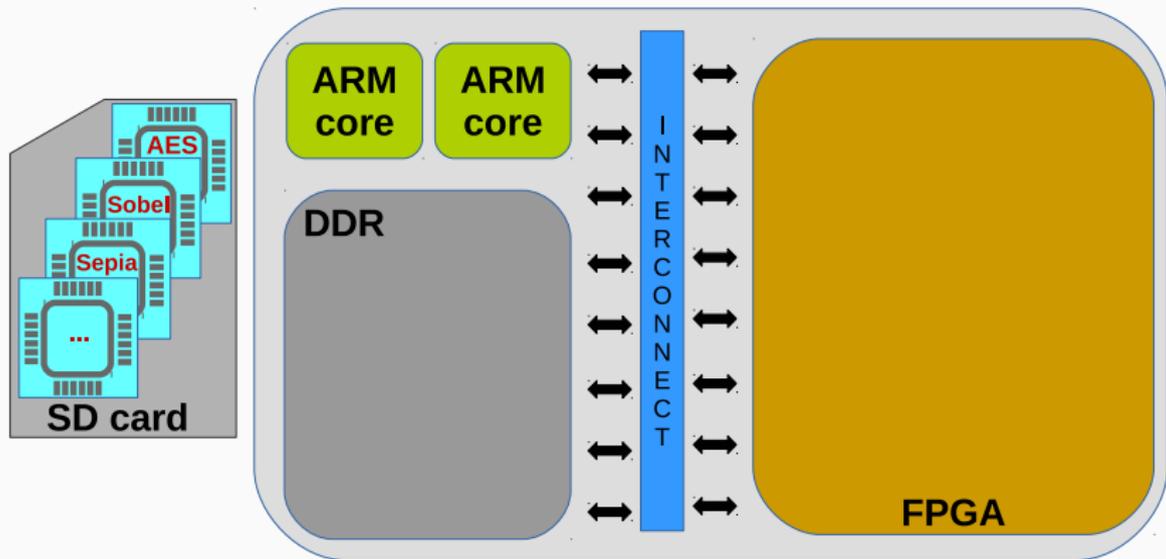


Contributions: DPR service

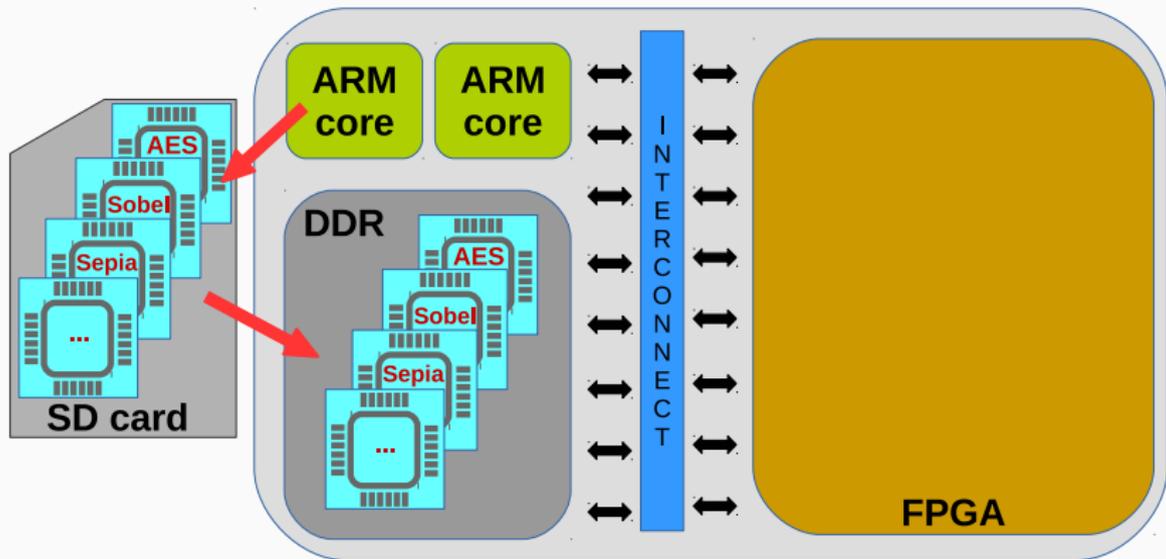


Dynamic Partial Reconfiguration in a RTOS

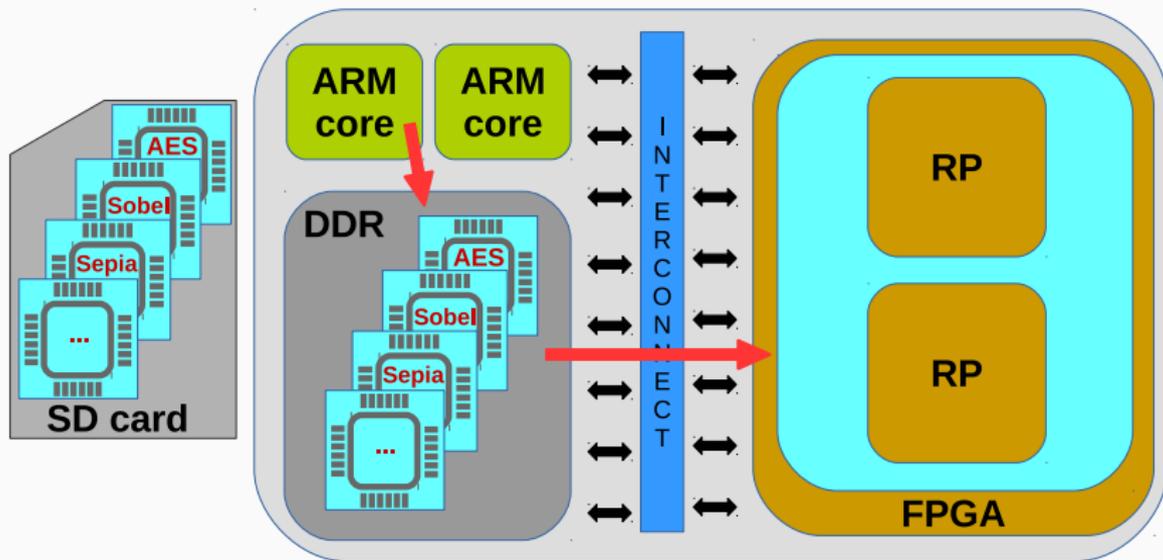
DPR service: bitstreams



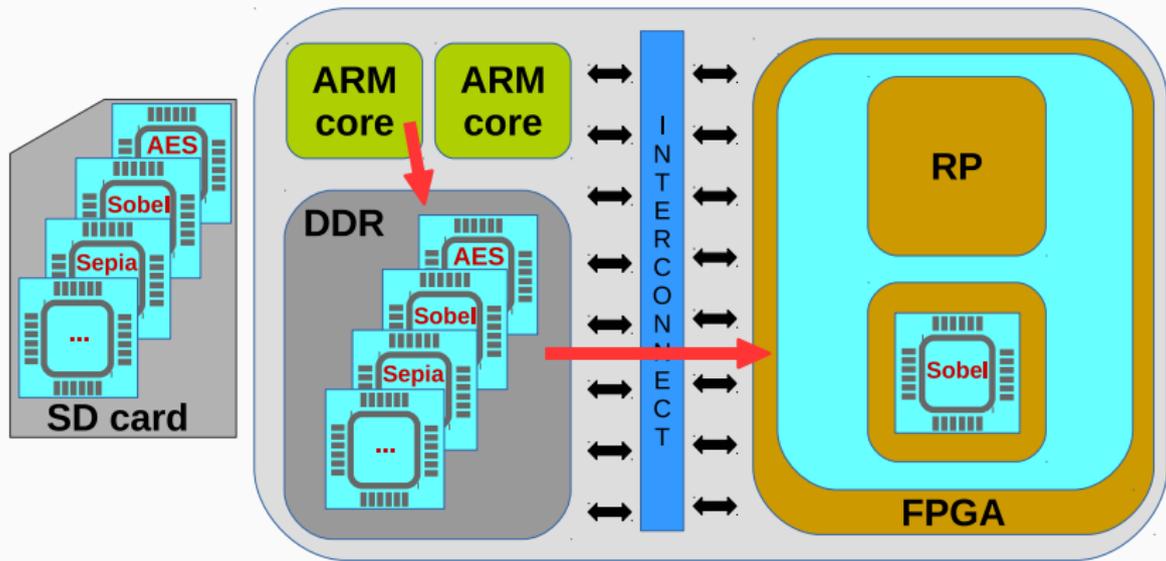
DPR service: file system



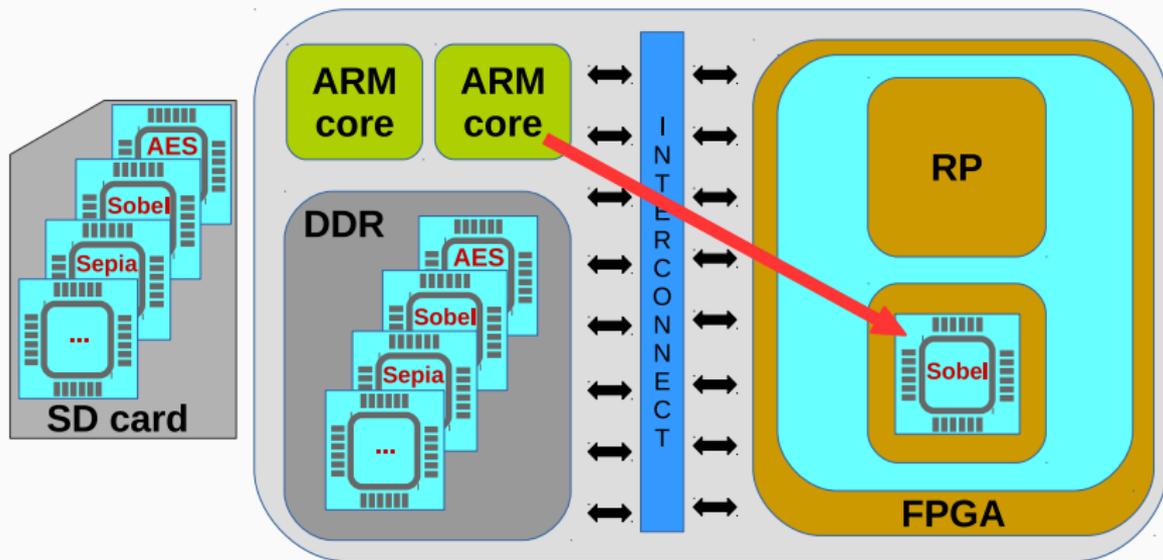
DPR service: static configuration



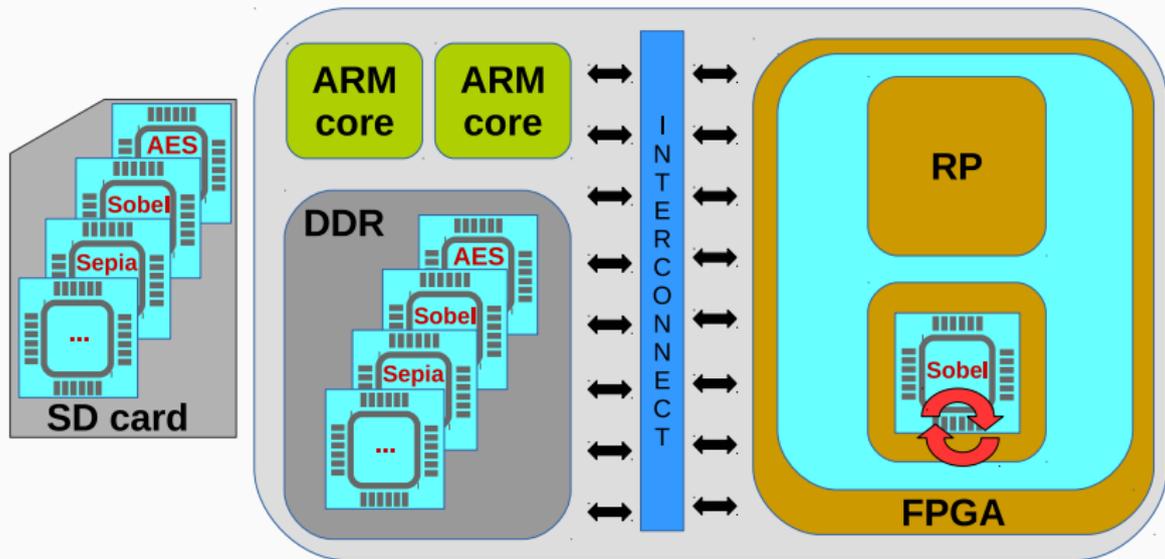
DPR service: dynamic modules upload



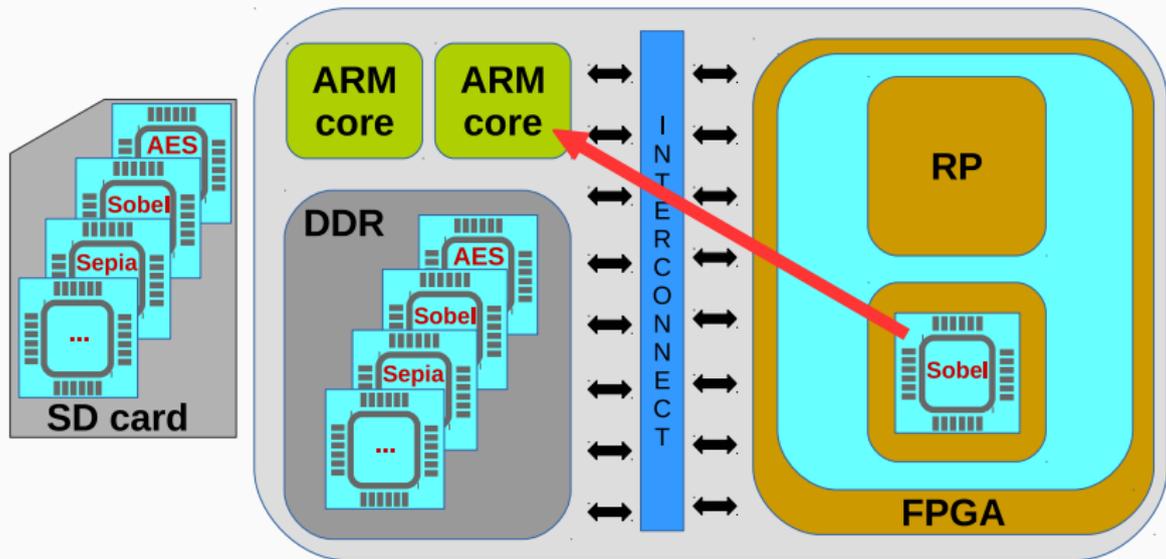
DPR service: dynamic modules configuration



DPR service: execution



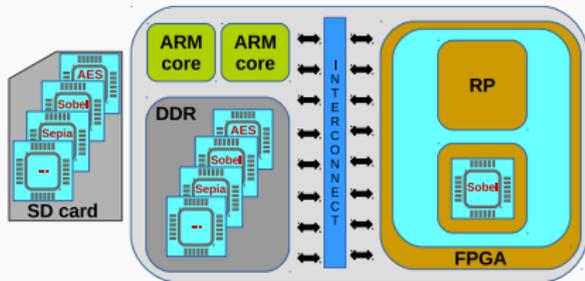
DPR service: interrupt



Requirements for DPR

New features implemented in HIPPEROS to support the DPR:

- ✓ a SD card driver(+file system),
- ✓ a FPGA driver,
- ✓ co-processors access,
- ✓ interrupts mechanism,
- ✓ a DPR service.



```
int hdpr_configure(const char* name)
```

```
int hdpr_open(const char* name, uint* id)
```

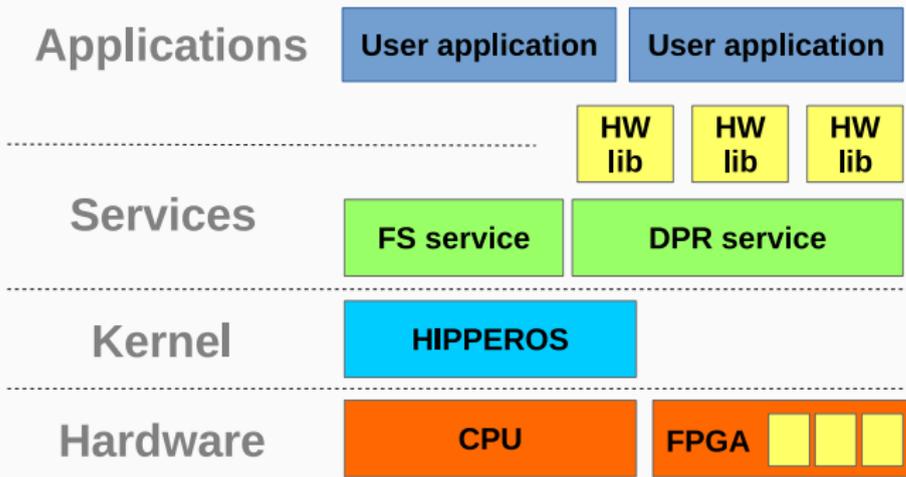
```
int hdpr_read(uint id, uint offset, uint value)
```

```
int hdpr_write(uint id, uint offset, uint value)
```

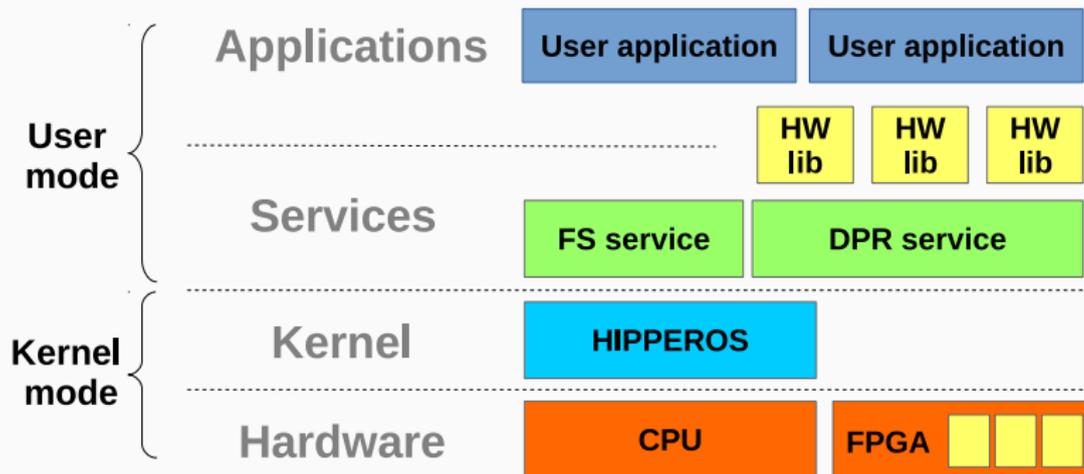
```
bool hdpr_interrupt(uint id, bool blocking)
```

```
int hdpr_close(uint id)
```

Abstraction layers



Abstraction layers

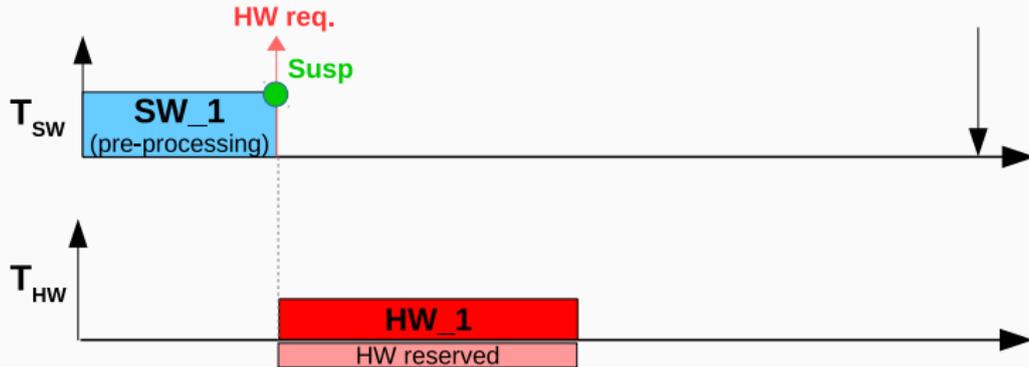


Scheduling Approach

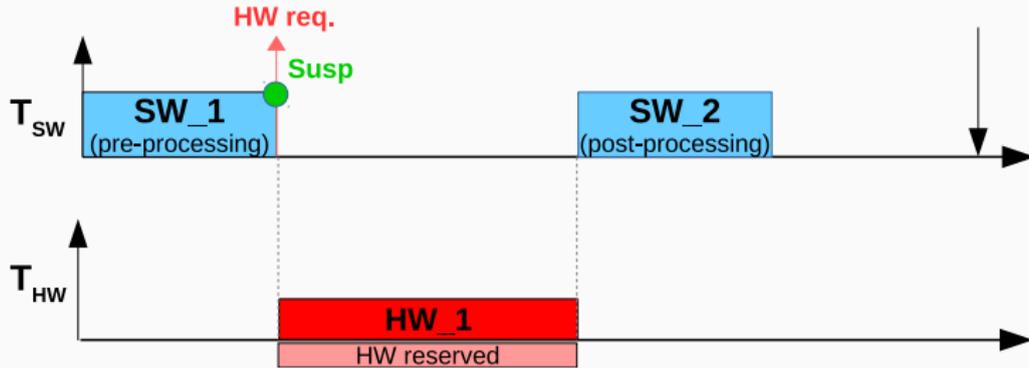
Coarse Heterogeneous Model



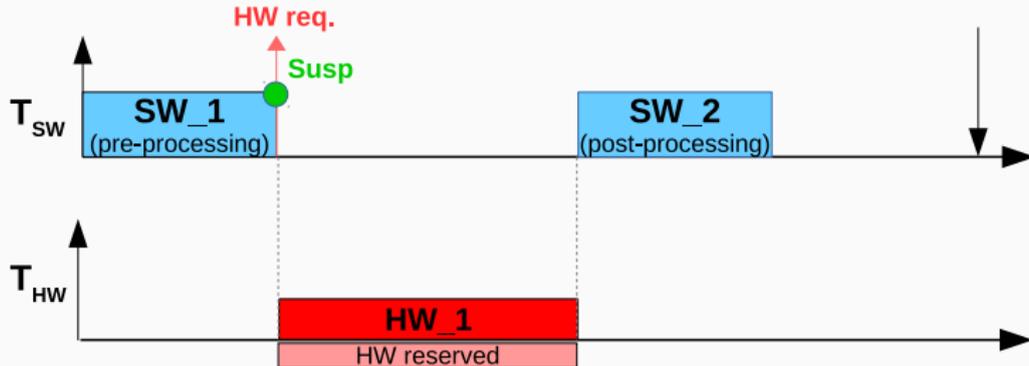
Heterogeneous Model: pre-processing



Heterogeneous Model: hardware



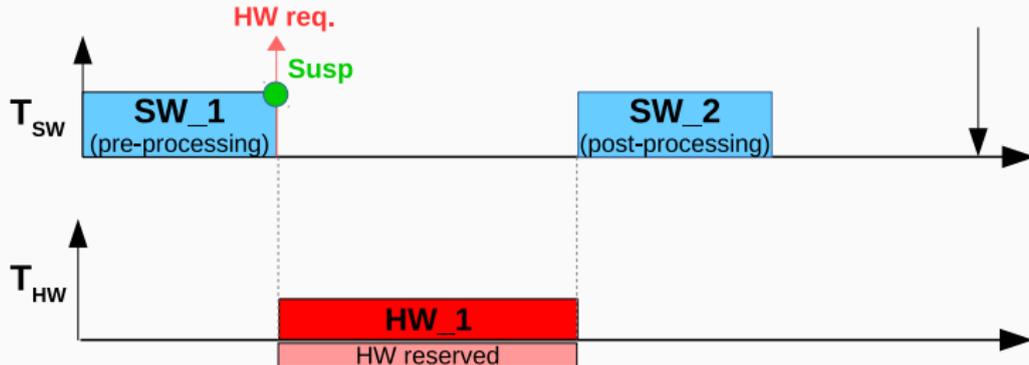
Heterogeneous Model



Constraints:

- **unrelated:** migrations are not allowed,
-

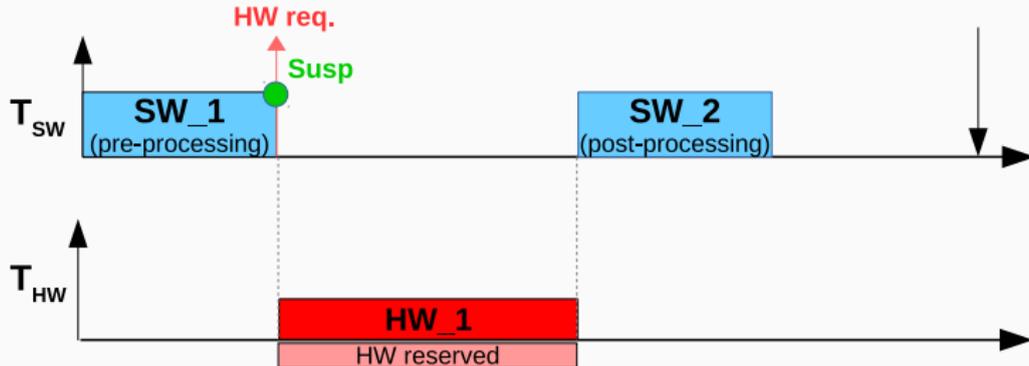
Heterogeneous Model



Constraints:

- **unrelated**: migrations are not allowed,
- **precedence**: sub-tasks needs to be sequentially executed,
-

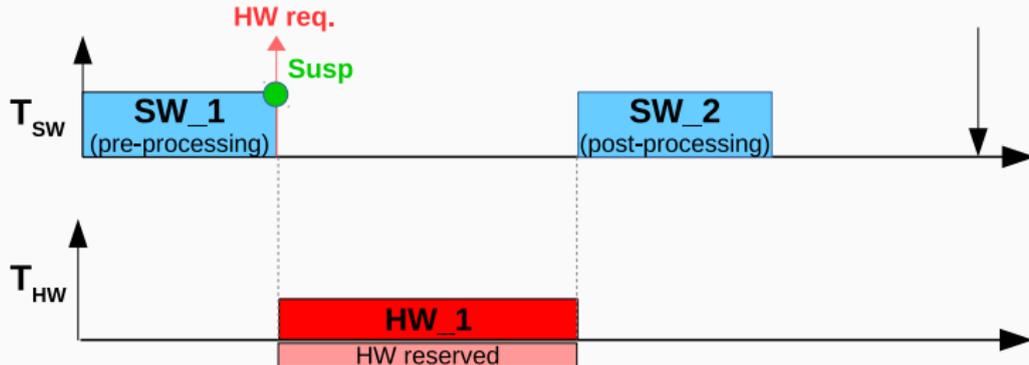
Heterogeneous Model



Constraints:

- **unrelated:** migrations are not allowed,
- **precedence:** sub-tasks needs to be sequentially executed,
- **non-preemptive:** interrupting the hardware is not allowed,
-

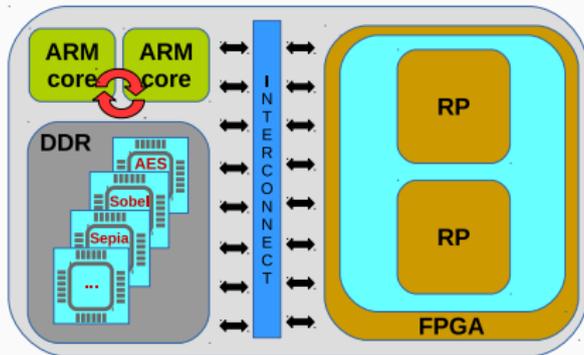
Heterogeneous Model



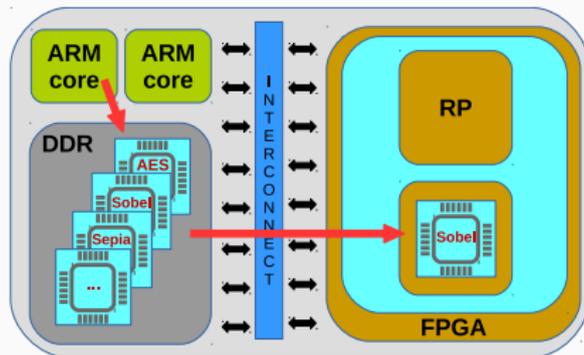
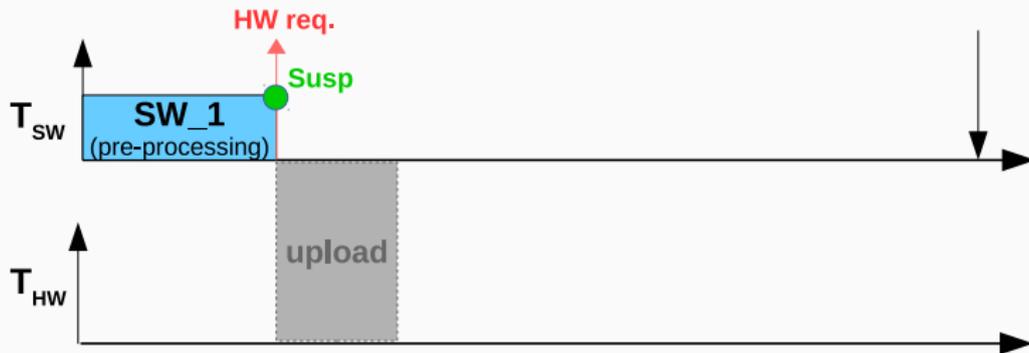
Constraints:

- **unrelated:** migrations are not allowed,
- **precedence:** sub-tasks needs to be sequentially executed,
- **non-preemptive:** interrupting the hardware is not allowed,
- **shared:** co-processors/RP's are shared by the tasks but must be exclusively used.

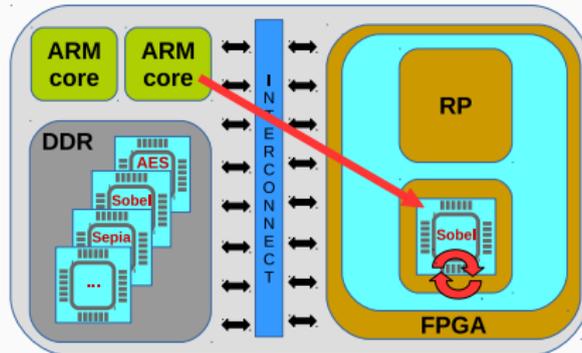
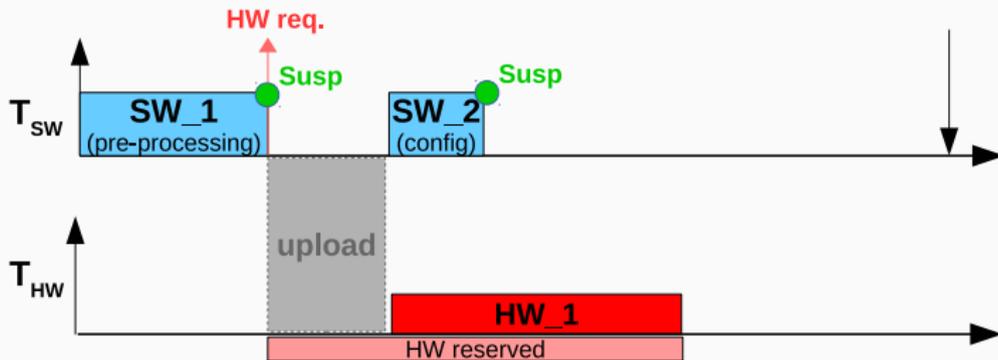
DPR Model: pre-processing



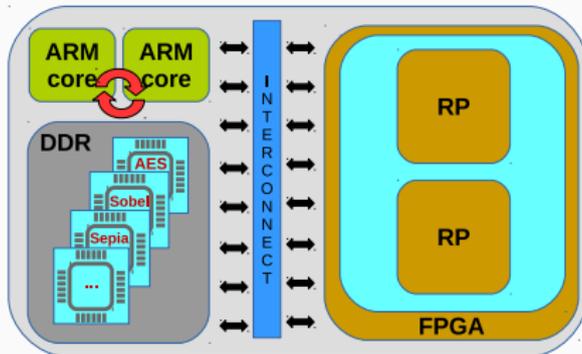
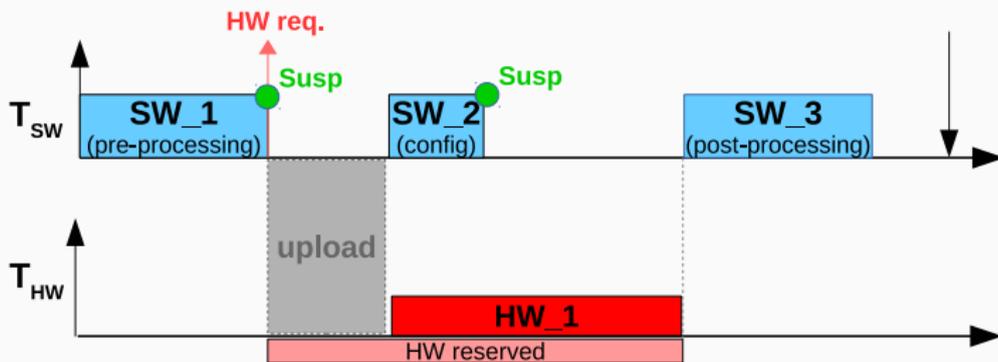
DPR Model: upload



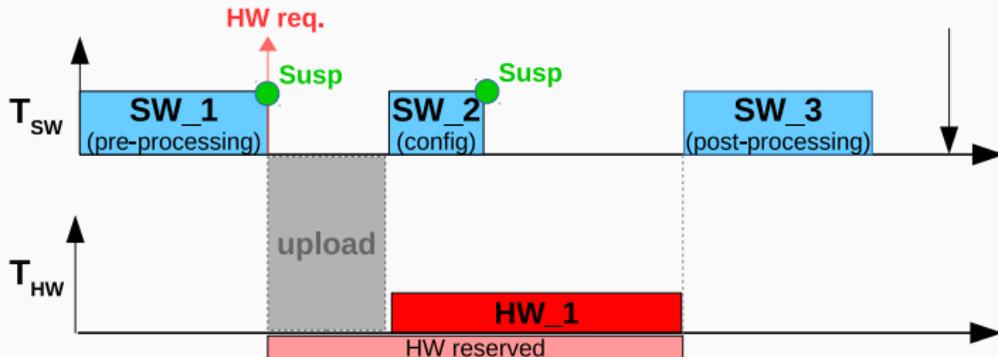
DPR Model: hardware



DPR Model: post-processing



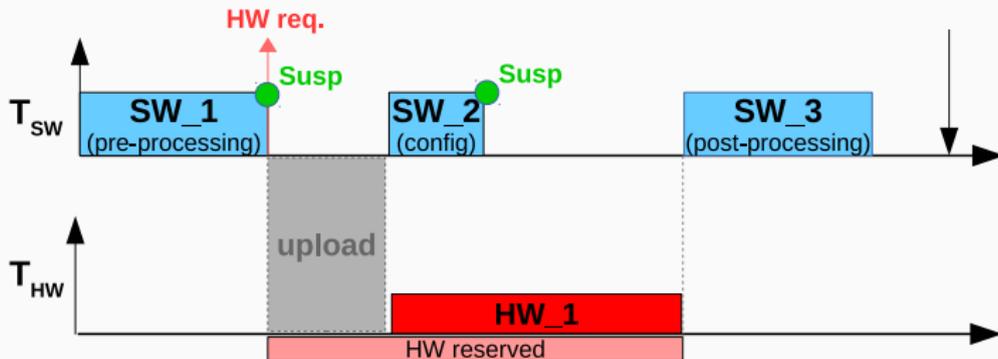
DPR Model



Additional/missing constraints:

- **Upload overhead:** the upload time is not negligible (GPU: executable, FPGA: bitstream)
-

DPR Model



Additional/missing constraints:

- **Upload overhead:** the upload time is not negligible (GPU: executable, FPGA: bitstream)
- **Configuration:** most of the time, co-processors must be configured (e.g.: provide the address of the data's to process)

Existing reusable models:

- **Shared resource protocols**

e.g.: PCP, SRP,...

[P. Gai, ECRTS 02]

→ Keep the CPU idle, too pessimist !!

- **Self-suspending task model**

e.g.: fixed-segment self-suspending tasks

[A. Biondi, RTSS 16]

→ NP-hard problem

[F. Ridouard, RTSS 05]

→ lot of mistakes in the literature !

[JJ Chen, TU-Dortmund 16]

... lot of work still need to be done !

Future Works

- integrate the self-suspending policy,
- Implement a image processing demo,
- Compare GPU-based system vs. FPGA-based system,
- Integrate HIPPEROS in the Xilinx tool-chain (TULIPP)



TULIPP: Towards Ubiquitous Low-power Image Processing Platforms

This European project wants to define a reference platform for low power image processing applications.

3 use-cases:

- Advanced driver assistance : Safer driving experience,
- Surveillance and rescue UAVs : Bring intelligence to the drones,
- Medical x-ray imaging : Reduce radiation dose by 75%.

Thank you!

Contact:

Martin Cornil: martin.cornil@hipperos.com

Antonio Paolillo: antonio.paolillo@hipperos.com

Joël Goossens: joel.goossens@ulb.ac.be

DPR Generic Interface

To allow HIPPEROS to handle these DM's without knowing their details, a minimal **generic interface** needs to be defined:

- a **bus connection** to memory map the device
- an **interrupt line** to notify HIPPEROS

