SystemC-based Co-Simulation for ESL HW/SW Co-Design of Heterogeneous Parallel Embedded Systems

Supervisor: Dott. Ric. Ing. Luigi Pomante

Co-Supervisor: Dott. Ric. Ing. Vittoriano Muttillo

Candidate: Marinella Negrini
265205

Main Purpose of this thesis

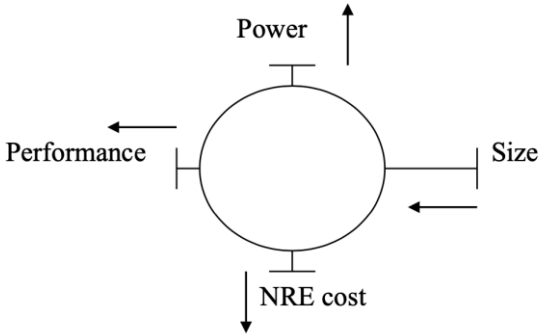
- Extend **HEPSIM** (part of *HEPSYCODE* methodology)
  - Multi-Link
  - Energy
  - Multiple CC4CS
  - CTF-compliant trace generation
- Validate the extension with a specific **Case Study**

www.hepsycode.com/

Contents

01 Introduction: ESL HW/SW Co-Design

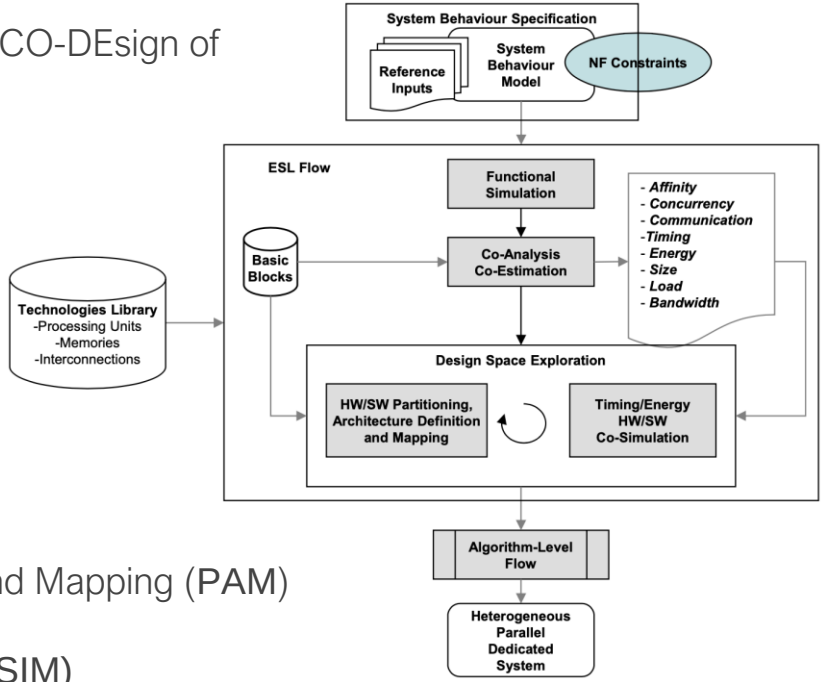
- ESL **HW/SW Co-Design** emphasizes a unified view of HW and SW at system level
- Embedded System Design is a complex task
- Heterogeneous technologies on a single chip (**MP-SoC**)
- Respect **Functional Requirements** and optimize competing Design Metrics



01 Introduction: HEPSYCODE

Hepsy Code
www.hepsycode.com/

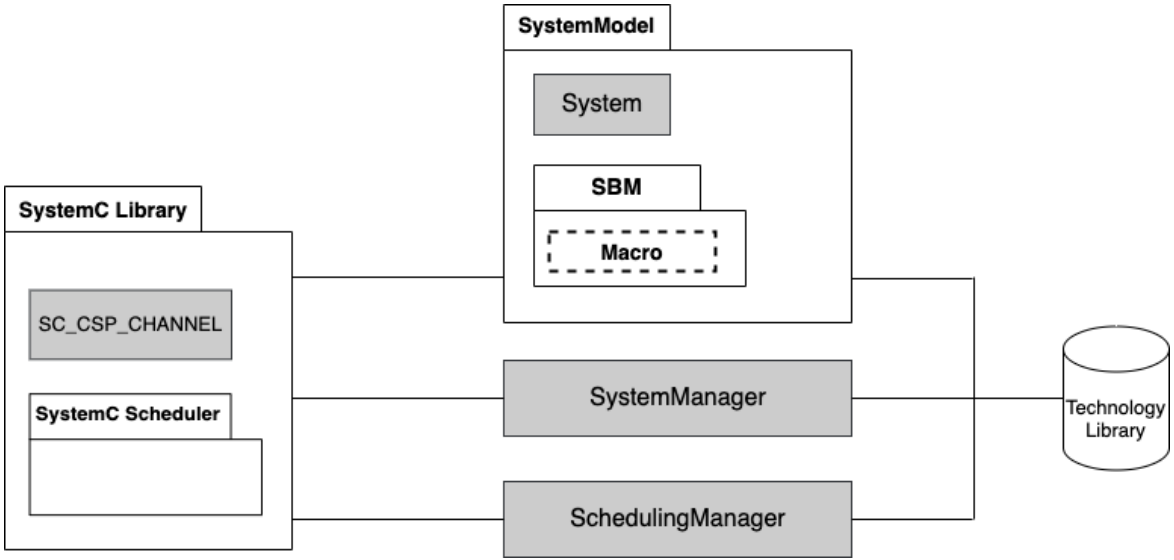
- HEPSYCODE = System-Level methodology for HW/SW CO-DEsign of HEterogeneous Parallel dedicated SYstems
- System Behaviour Model
- Functional Simulation
- Co-Analysis and Co-Estimation
- Design Space Exploration
  - HW/SW Partitioning, Architecture Definition and Mapping (PAM)
  - Timing/Energy HW/SW Co-Simulation (HEPSIM)



Contributions to HEPSYCODE

HEPSIM Architecture

- SystemManager
- SchedulingManager
- Macros
- SC_CSP_CHANNEL
- Technology Library



02 Contributions to HEPSYCODE: HEPSIM Multi-Link

- Possibility of adopting multiple **Physical Links** in communications, instead of a single shared bus
- The available Links are inserted in Technologies Library
- Each Channel composing the SBM can be mapped on a different Physical Link

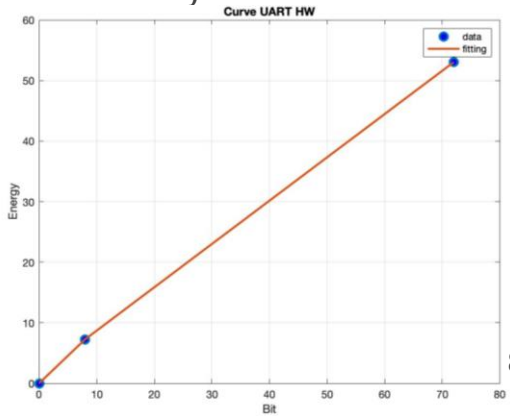
```
<mapping>
    <allocation CHid="0">
        <channelId CHid="0" Lname="RAM@20_min" value="1" />
    </allocation>
    <allocation CHid="1">
        <channelId CHid="1" Lname="RAM@20_min" value="1" />
    </allocation>
    <allocation CHid="2">
        <channelId CHid="2" Lname="RAM@20_min" value="1" />
    </allocation>
    <allocation CHid="3">
        <channelId CHid="3" Lname="RAM@20_min" value="1" />
    </allocation>
    <allocation CHid="4">
        <channelId CHid="4" Lname="GPIO_PORT@20_medium" value="7" />
    </allocation>
    <allocation CHid="5">
        <channelId CHid="5" Lname="GPIO_PIN_min" value="13" />
    </allocation>
</mapping>
```

02 Contributions to HEPSYCODE: Energy

- **Program's statements** (J4CS metric)

  - Energy Per Instruction (EPI) method
    - SW: $J4CS = EPI \times I4CS = \frac{P}{MIPS} \times \frac{I}{CS}$
    - HW: $J4CS = P \times \frac{1}{f} \times CC4CS = (V_{dd} \times I_{dd}) \times CC4CS \times \frac{1}{f}$
  - Energy Per Cycle (EPC) method → $J4CS = EPC \times CC4CS = \frac{P}{f} \times CC4CS$

- **Context Switch**: $CS_{EN} = CS_{OH} \times P$
- **Data transfers**: characterization of Links in the TL with the coefficients of an interpolating polynomial of degree 2 (energy as a function of #bits transmitted)



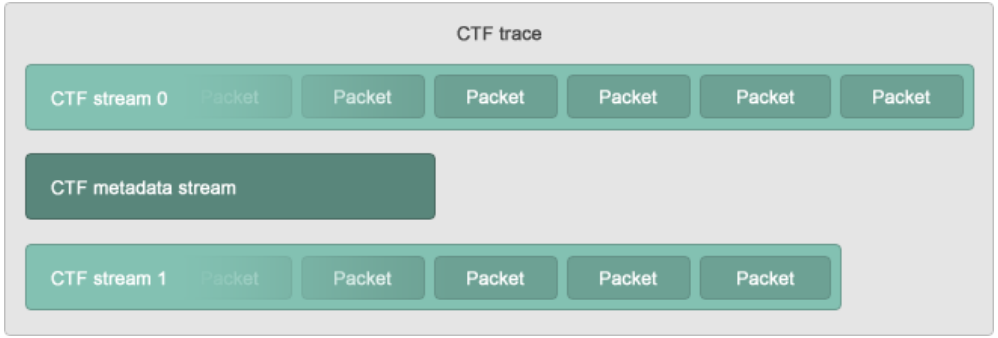
02 Contributions to HEPSYCODE: Multiple CC4CS for different data types

- CC4CS metric: number of clock cycles needed by a processor to execute a common C statement
- Possibility to distinguish between processes, according to their dominant data type
- Processors in the TL are characterized by a set of CC4CS metrics (one for each data type)
- Designers specify the dominant data type for each process of the SBM

```
<processingUnit>
    <name>MPU8051@12</name>
    <idprocessor>0</idprocessor>
    <processorType>GPP</processorType>
    <cost>1</cost>
    <ISA>8051</ISA>
    <frequency>12</frequency>
    <CC4SminInt8>84.9</CC4SminInt8>
    <CC4SmaxInt8>139.8</CC4SmaxInt8>
    <CC4SminInt16>118.4</CC4SminInt16>
    <CC4SmaxInt16>190.1</CC4SmaxInt16>
    <CC4SminInt32>166.3</CC4SminInt32>
    <CC4SmaxInt32>279.7</CC4SmaxInt32>
    <CC4SminFloat>293.5</CC4SminFloat>
    <CC4SmaxFloat>614.6</CC4SmaxFloat>
    <CC4Smin>125.5</CC4Smin>
    <CC4Smax>293.3</CC4Smax>
    <Power>0.6</Power>
    <MIPS>0.72</MIPS>
    <I4CSmin>7.5</I4CSmin>
    <I4CSmax>17.2</I4CSmax>
    <Vdd>6</Vdd>
    <Idd>NA</Idd>
    <overheadCS>405</overheadCS>
    <capacity />
</processingUnit>
```

Contributions to HEPSYCODE: HEPSIM CTF-compliant trace generation

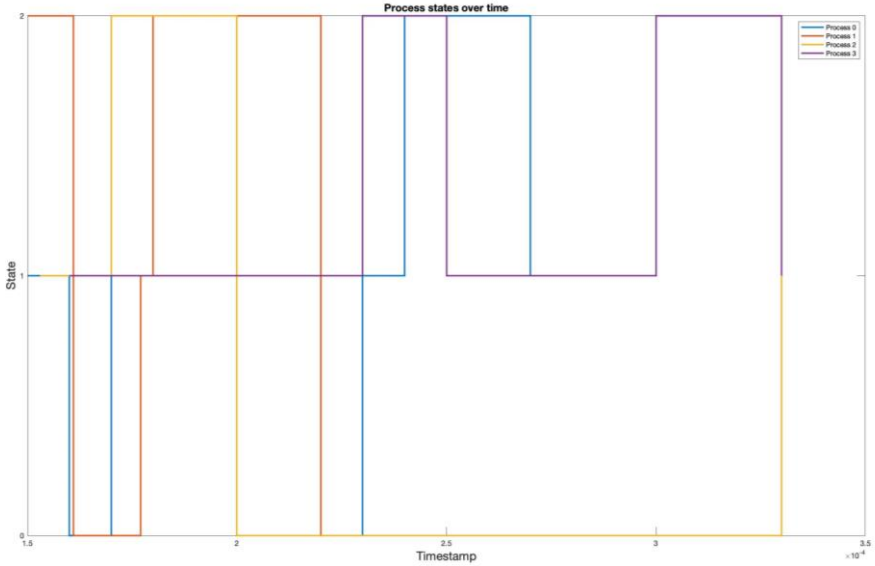
- The **Common Trace Format** (CTF) is a binary trace format
- The CTF is the reference format adopted in **FitOptiVis**
- CTF Trace = multiple **streams** of binary **events** (organized in **packets**) + **metadata** stream
- **Babeltrace** is the reference implementation of the CTF (Python3 bindings to read and write the CTF)



02 Contributions to HEPSYCODE: HEPSIM CTF-compliant trace generation

- HEPSIM generates a **LOG** during the execution
- A CTF trace is generated from the LOG file

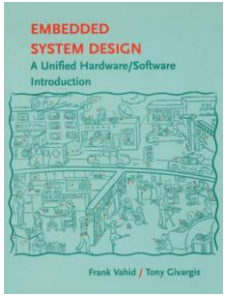
```
0.0002265 – PS:2 – 0
0.0002265 – CH:4 – 2
0.0002265 – PS:2 – 0
0.0002265 – SC:1 – 2
0.0002271 – CH:4 – 1
0.0002271 – CH:4 – 0
```



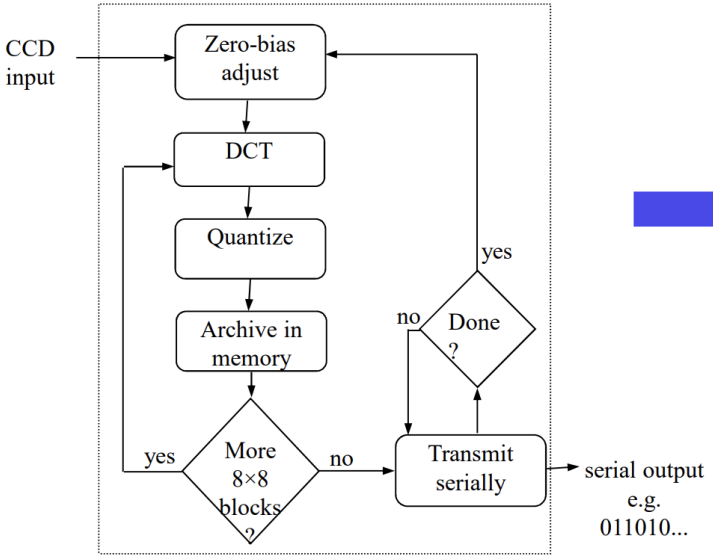

```
marinella@marinella-VirtualBox:/home/ctf$ ls -all
total 16024
drwxrwxrwx 2 root      root          4096 ott  1 08:39 
drwxr-xr-x 4 root      root          4096 ott  1 08:39 ..
-rw-rw---- 1 marinella marinella     2083 ott  1 08:39 metadata
-rw-rw---- 1 marinella marinella  1376256 ott  1 08:39 stream_profiler_1_0
-rw-rw---- 1 marinella marinella    32768 ott  1 08:39 stream_profiler_1_1
-rw-rw---- 1 marinella marinella   917504 ott  1 08:39 stream_profiler_1_2
-rw-rw---- 1 marinella marinella        0 ott  1 08:39 stream_profiler_1_3
-rw-rw---- 1 marinella marinella 13369344 ott  1 08:39 stream_profiler_1_4
-rw-rw---- 1 marinella marinella        0 ott  1 08:39 stream_profiler_1_5
-rw-rw---- 1 marinella marinella   688128 ott  1 08:39 stream_profiler_1_6
```

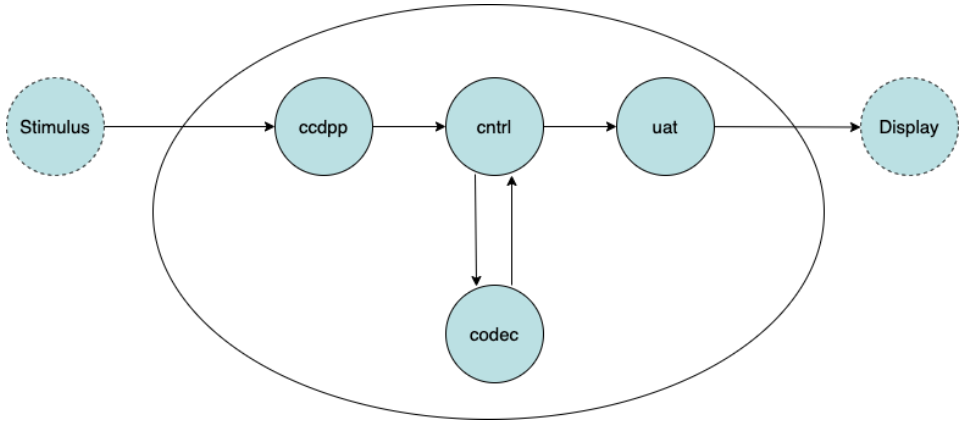
03 Validation – DC Case Study: objectives

- Case study presented in the book *«Embedded System Design: A Unified Hardware/Software Introduction»*, by T.Givargis and F.Vahid
- Digital Camera performing a simplified **JPEG compression** and successively improved **implementations**
- **Validation** of the extensions, **comparison** of the results and **DSE**



03 Validation – DC Case Study: Functional Specification





03 Validation – DC Case Study: Implementations

Four different implementations:

1. Microcontroller (8051) alone (AllSW)
2. Microcontroller (8051) and HW CCDPP/UART
3. Microcontroller (8051) and Fixed-Point CODEC and HW CCDPP/UART
4. Microcontroller (8051) and HW CCDPP/UART and HW Fixed-Point CODEC

VHDL (RTL) simulation

ASIC synthesis tool (gate-level)

	Imp1 (manually estimated ccdpp only)	Imp2	Imp3	Imp4
Performance (second)	near 0.5	9.1	1.5	0.099
Power (Watt)	-	0.033	0.033	0.040
Size (gate)	-	98000	90000	128000
Energy (Joule)	-	0.30	0.050	0.0040

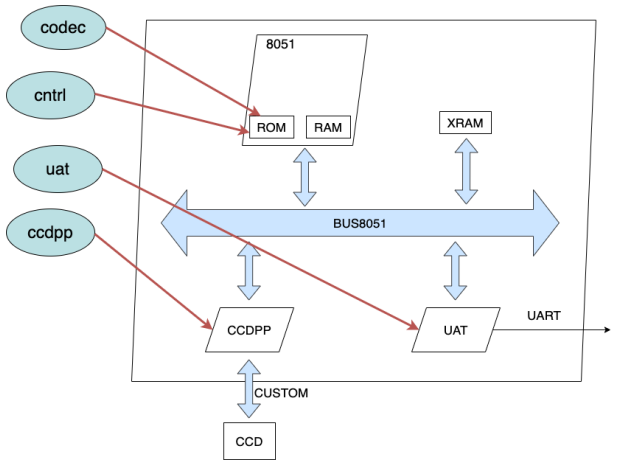
03 Validation – DC Case Study: Simulation with HEPSIM

Processors in the TL

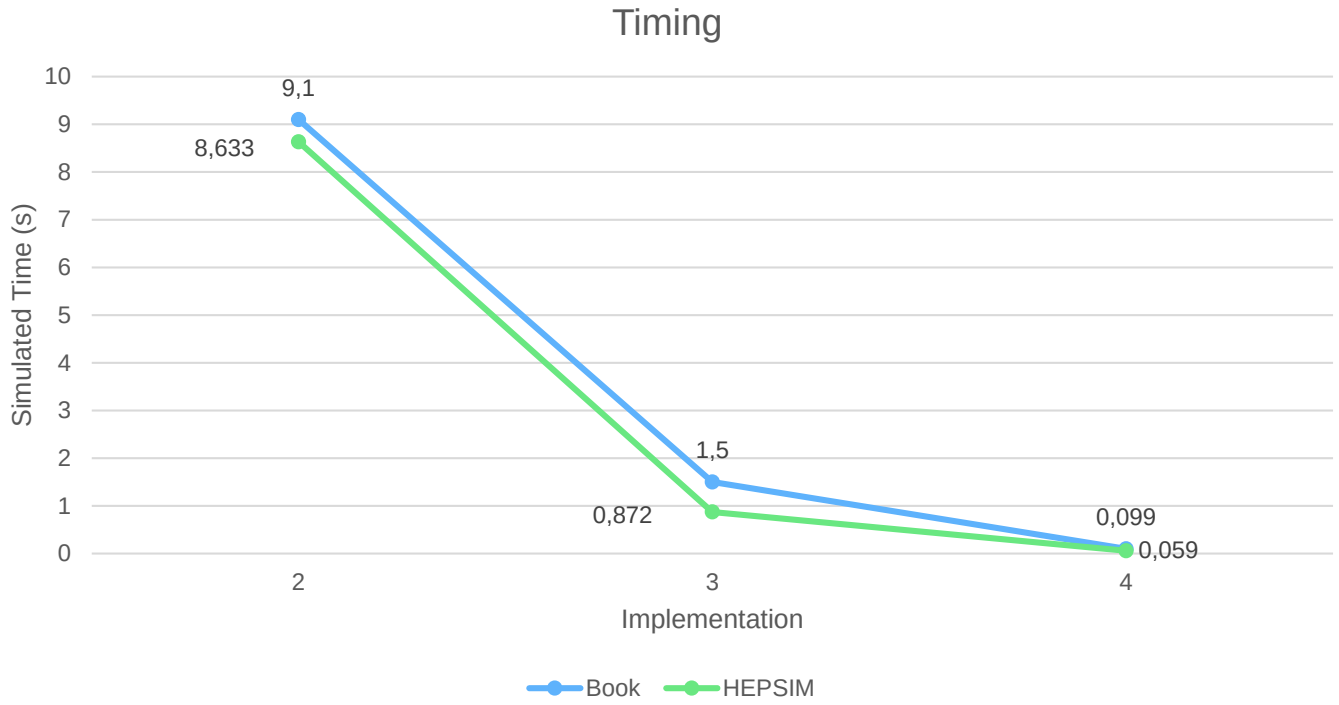
Processor	Frequency	CC4CSmin	CC4CSmax	Power	Vdd	Idd	I4CSmin	I4CSmax	MIPS	CS_OH
Intel 8051 (GPP)	12MHz	125.5	293.3	600mW	6V	-	7.5	17.2	0.72	405us
	20MHz								1.2	243us
Xilinx Artix7 (FPGA)	12MHz	0.9	1.9	-	2V	15uA	-	-	-	-
	20MHz									

Manual mapping to reproduce the Implementations

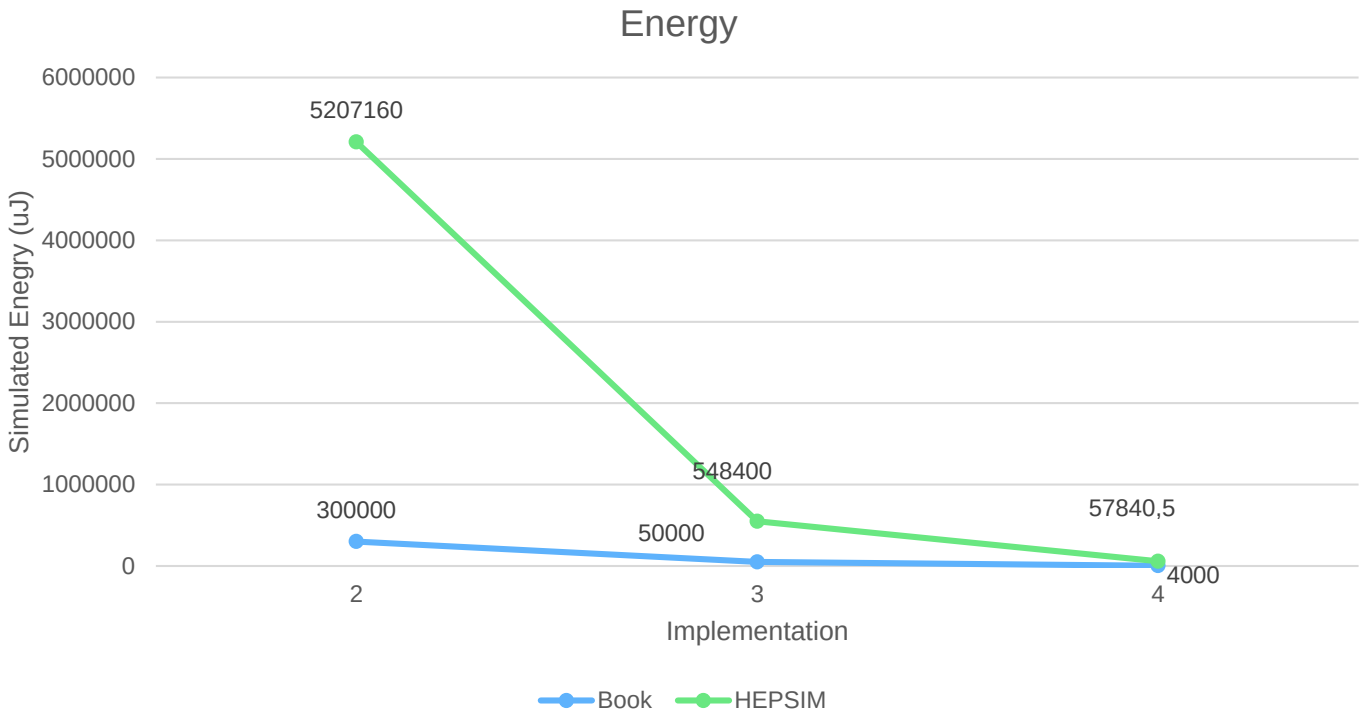
Example: Implementation 2 →



03 Validation – DC Case Study: Results (updated)



03 Validation – DC Case Study: Results (updated)



03 Validation – DC Case Study: Analysis and Considerations (updated)

Average Relative Errors (16x16)

Av. Rel. Error Time	Av. Rel. Error Energy
29,1%	N/A (SoC vs SoB)

HEPSIM vs RTL simulation

		HEPSIM	RTL
Simulation Time	Implementation 2	~10 seconds	~15 minutes
	Implementation 3	~10 seconds	~10 minutes
	Implementation 4	~5 seconds	~5 minutes
Modeling Time		Some minutes	Some hours

- HEPSIM estimation results are less **accurate** than results obtained via RTL simulation (System-level vs RTL)
- However, the HEPSIM estimation results are **reliable** (**comparison** and **estimation**)
- HEPSIM **Simulation** requires less time than RTL Simulation
- HEPSIM **Modeling** requires less time than RTL modeling

04 Conclusions

- HW/SW Co-Design techniques and **HEPSYCODE**
- Extend **HEPSIM**
- Validate using a **Case Study**

- Development of the SBM for **parallel DC**

- Deliverable D3.3 **FitOptiVis**

smart IntegraTion and OPtimisation Technologies for highly efficient Image and VIdeo processing Systems
https://cordis.europa.eu/project/id/783162

05 Future works

- Model SW partitions based on **hypervisor**
- Consider **Power** analysis and peak power
- Model with **Multi-MoC**

Thank you for your attention!