Analysis of a Heterogeneous Multi-Core, Multi-HW-Accelerator-Based System Designed Using PREESM and SDSoC

Leonardo Suriano, Alfonso Rodriguez, Karol Desnos,

Maxime Pelcat, Eduardo de la Torre

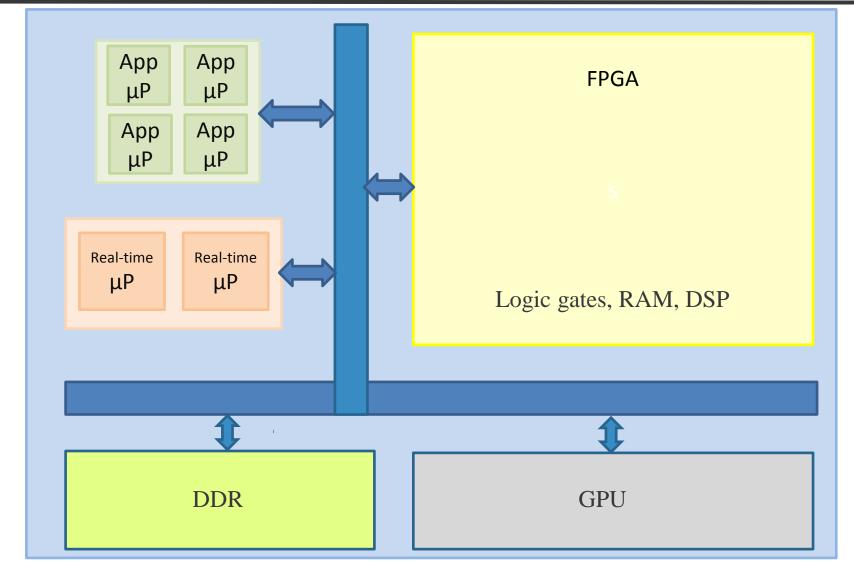UNIVERSIDAD POLITÉCNICA DE MADRID

cei@upm.es

Introduction

Embedded
World

…and many others

CEI UPM

POLITÉCNICA

Heterogeneous devices

SW + HW productivity GAP

Lines of code/chip
x2 every 10 months

Transistors/chip
x2 every 18 months

SW + HW Productivity Gaps

Lines of code/day
x2 every 5 years

Software complexity

Hardware complexity

Software productivity

log

1990 1995 2000 2005 2010 2015

CEI UPM

POLITÉCNICA

GOAL

Heterogeneous devices (HW + SW)

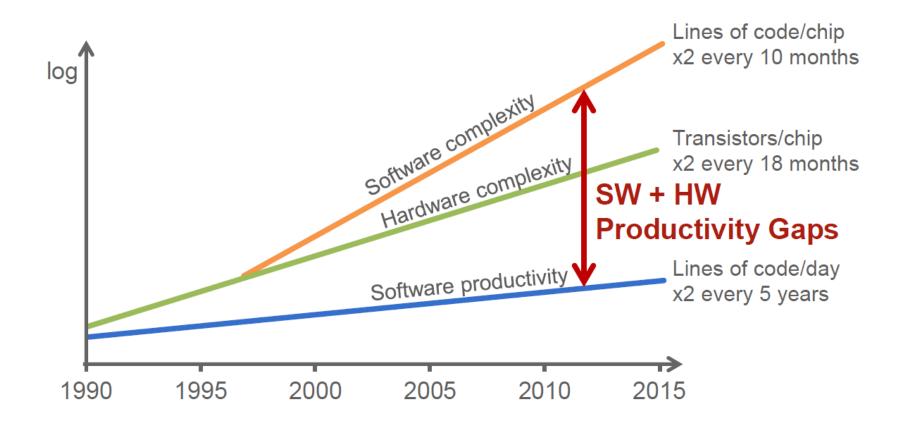Easy to program

Task scheduling
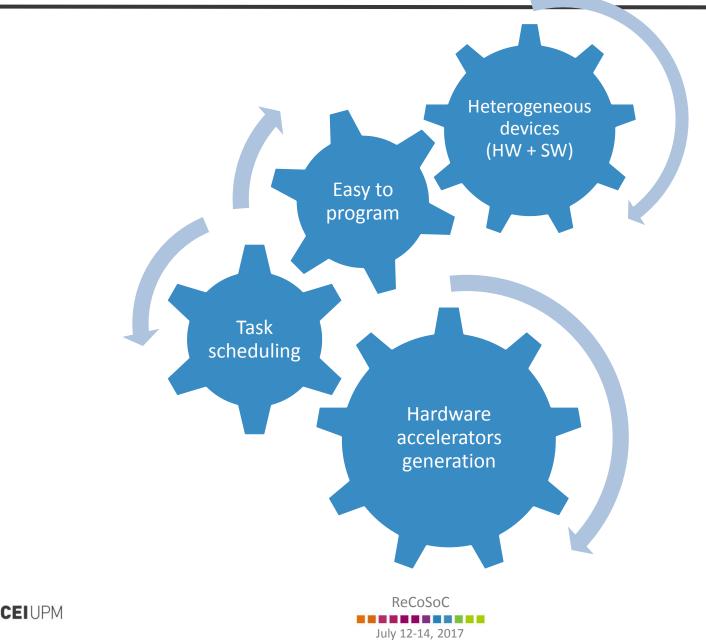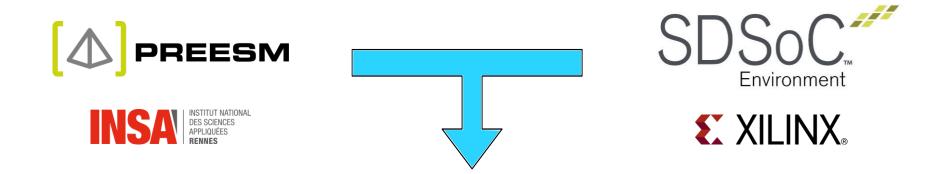
Hardware accelerators generation

CEI UPM

POLITÉCNICA

Main Contribution

Rapid prototyping of software applications enabling:

- deadlock-free code generation using **PiSDF MoC**

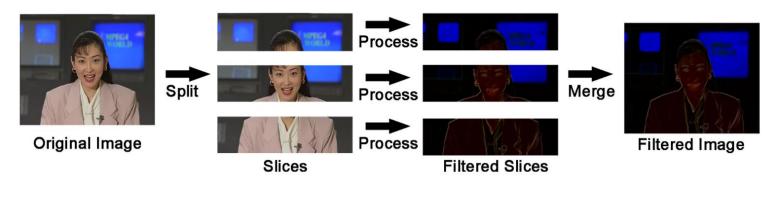- custom **hardware acceleration** and generation
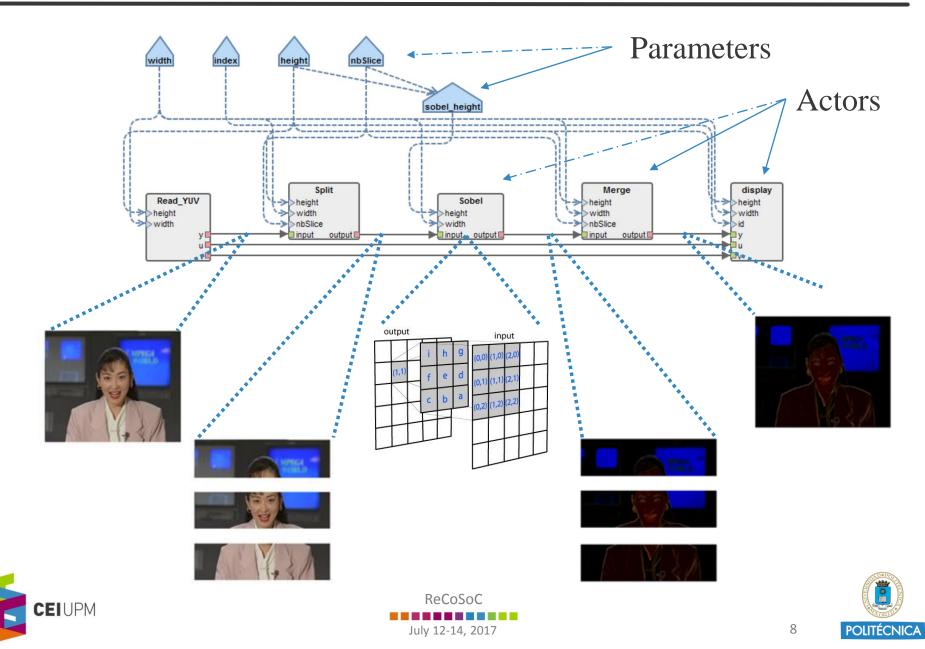
Video Processing Application

Sobel

Original Image → Split → Slices → Process → Filtered Slices → Merge → Filtered Image

CEIUPM

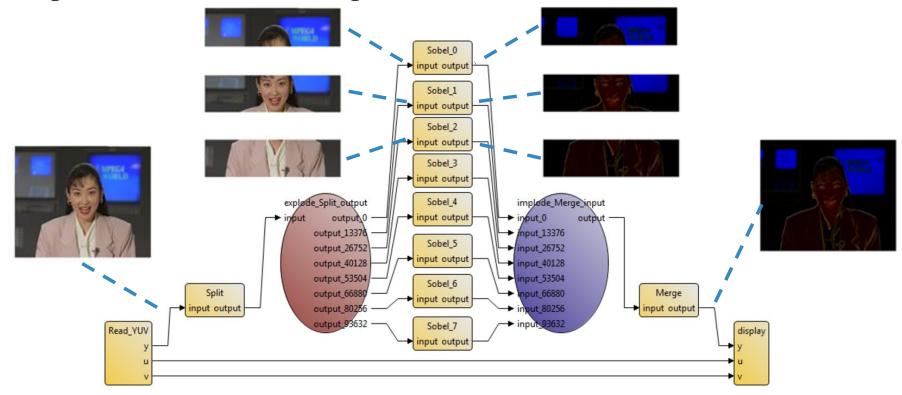POLITÉCNICA

Actors Specification

Parameters

Actors

CEI UPM

Exposed parallelism

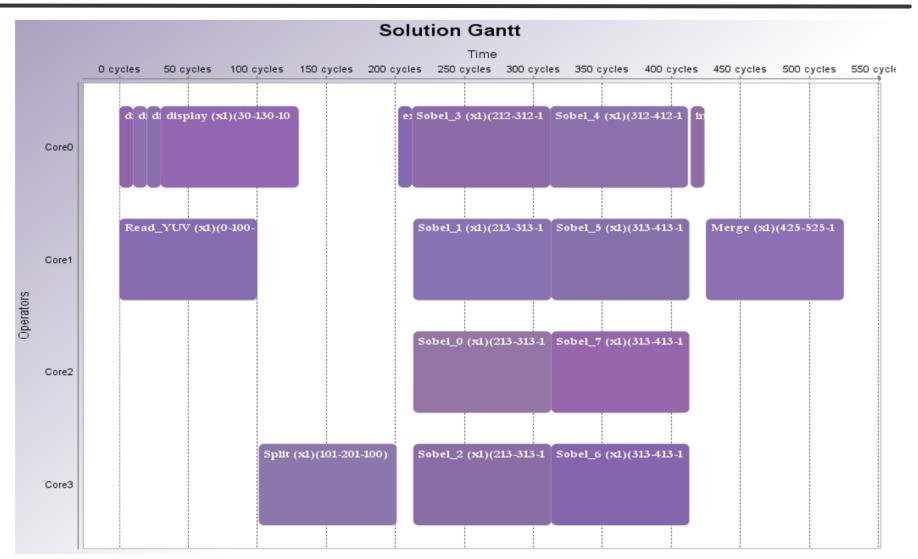**Equivalent single-rate graph** where each edge has equal production and consumption rates of token.
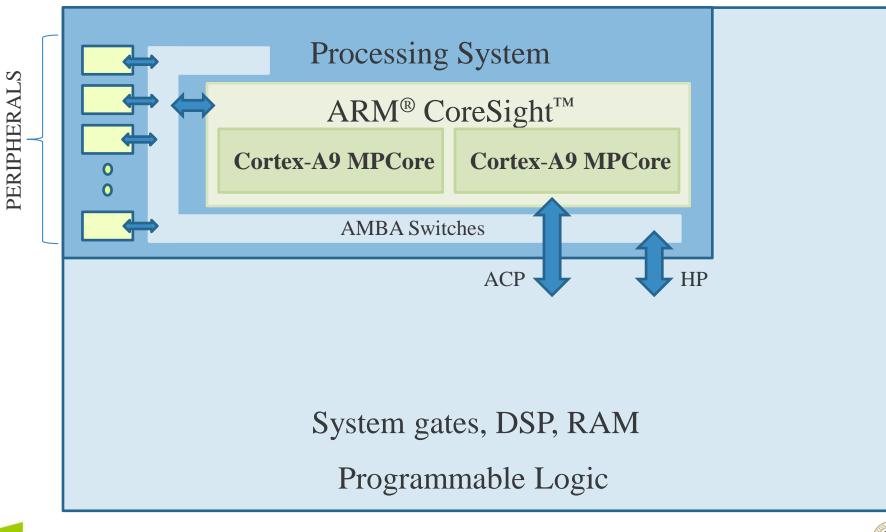
Actors scheduled

Solution Gantt

Time

0 cycles   50 cycles   100 cycles   150 cycles   200 cycles   250 cycles   300 cycles   350 cycles   400 cycles   450 cycles   500 cycles   550 cycle

Core0: d d d display (x1)(30-130-10)   e: Sobel_3 (x1)(212-312-1)   Sobel_4 (x1)(312-412-1)   in

Core1: Read_YUV (x1)(0-100-   Sobel_1 (x1)(213-313-1)   Sobel_5 (x1)(313-413-1)   Merge (x1)(425-525-1)

Core2: Sobel_0 (x1)(213-313-1)   Sobel_7 (x1)(313-413-1)

Core3: Split (x1)(101-201-100)   Sobel_2 (x1)(213-313-1)   Sobel_6 (x1)(313-413-1)

Operators

CEI UPM

POLITÉCNICA

Heterogeneous System

Zynq-7000 family

PERIPHERALS

Processing System

ARM® CoreSight™

Cortex-A9 MPCore   **Cortex-A9 MPCore**

AMBA Switches

ACP   HP

System gates, DSP, RAM

Programmable Logic

CEI UPM

POLITÉCNICA
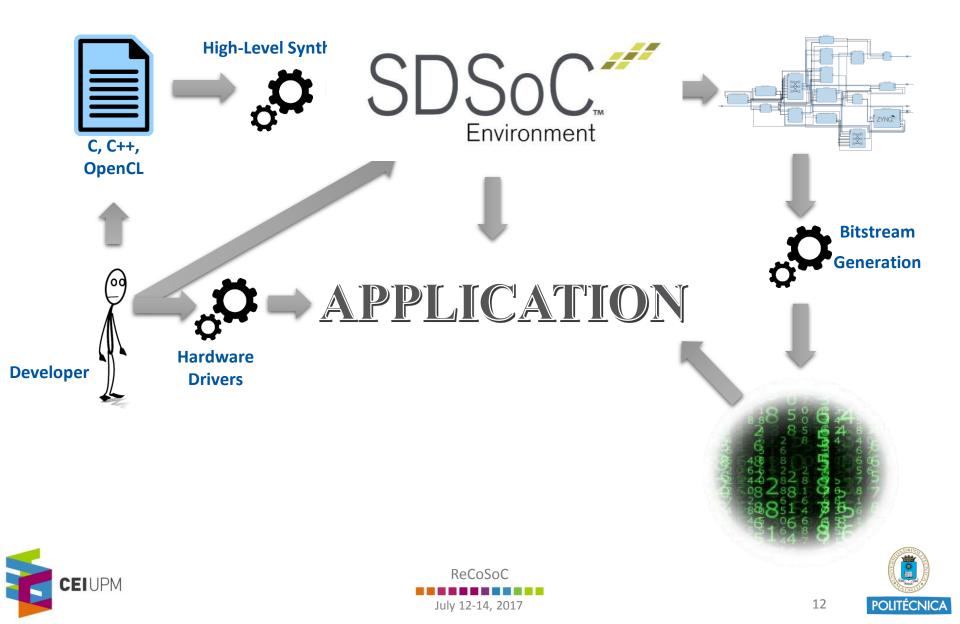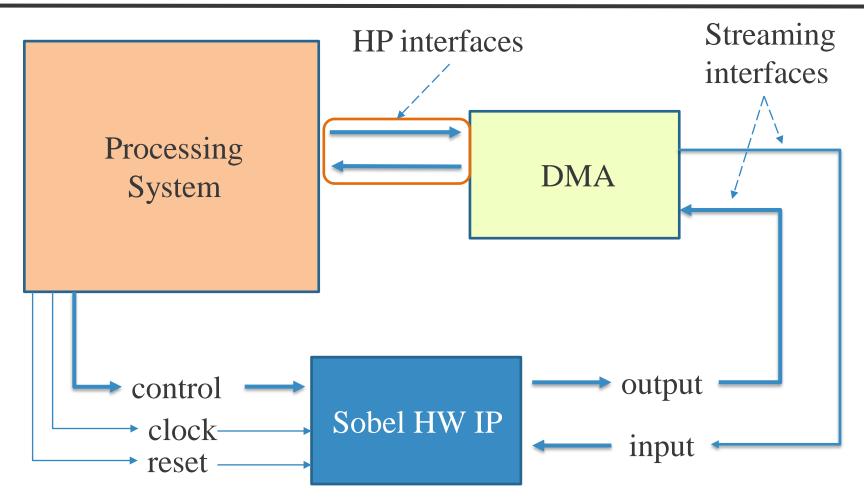
High Level Synthesis and SDSoC

C, C++, OpenCL

High-Level Synth

SDSoC™ Environment

Bitstream Generation

APPLICATION

Developer

Hardware Drivers

From PREESM to SDSoC

Processing System

HP interfaces

Streaming interfaces

DMA

control

clock

reset

Sobel HW IP

output

input

CEI UPM
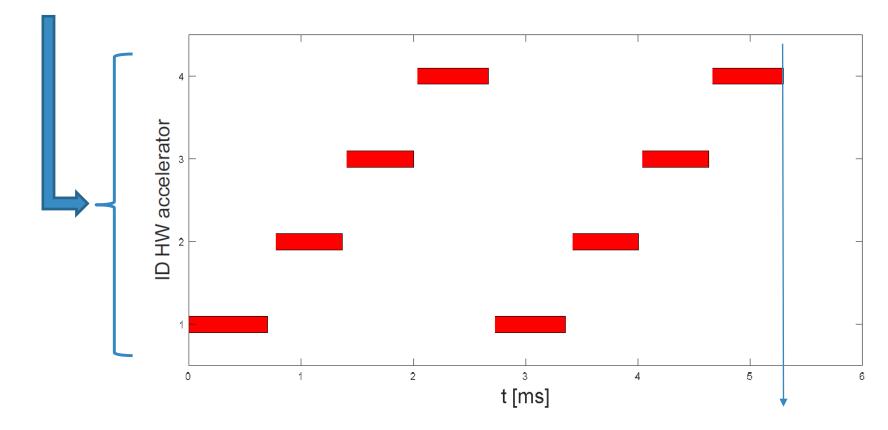
Results

4 hardware accelerators
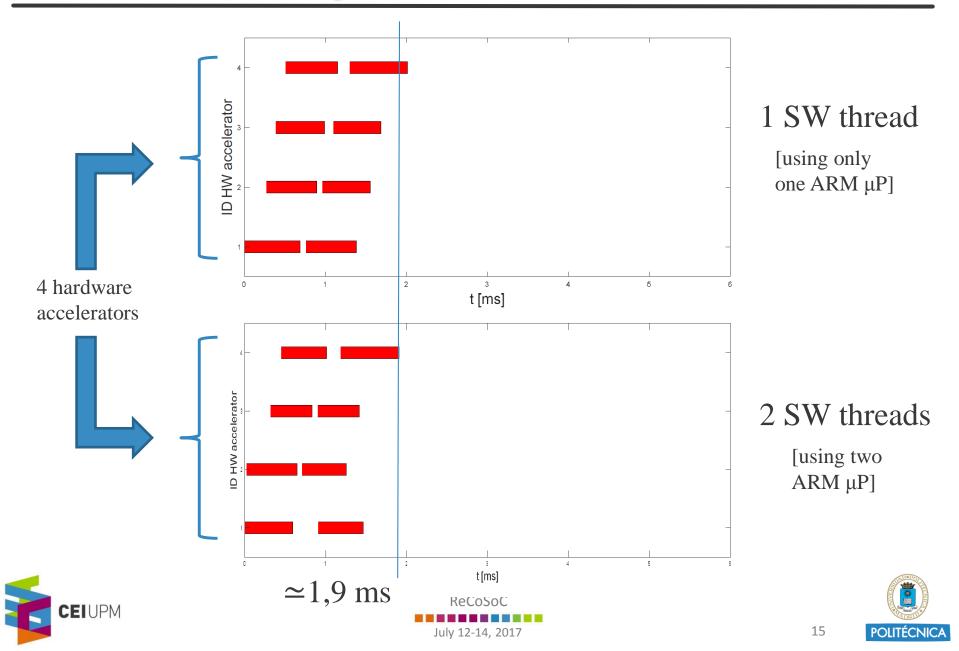
$\simeq 5{,}25$ ms

Asynchronous calls

4 hardware accelerators

1 SW thread

[using only one ARM μP]

2 SW threads

[using two ARM μP]

$\simeq 1{,}9$ ms

CEI UPM

POLITÉCNICA

Results

1 THREAD – 8 SLICES – 4 ACCELERATORS			
HW freq. [MHz]	142.86	166.67	200
SDSoC HW handling	115 fps	125 fps	135 fps
Manual handling	200 fps	205 fps	207 fps

2 THREAD – 8 SLICES – 4 ACCELERATORS			
HW freq. [MHz]	142.86	166.67	200
SDSoC HW handling	150 fps	160 fps	171 fps
Manual handling	211 fps	218 fps	222 fps

Comparison

HW clock frequency = 200 MHz

CEI UPM

POLITÉCNICA
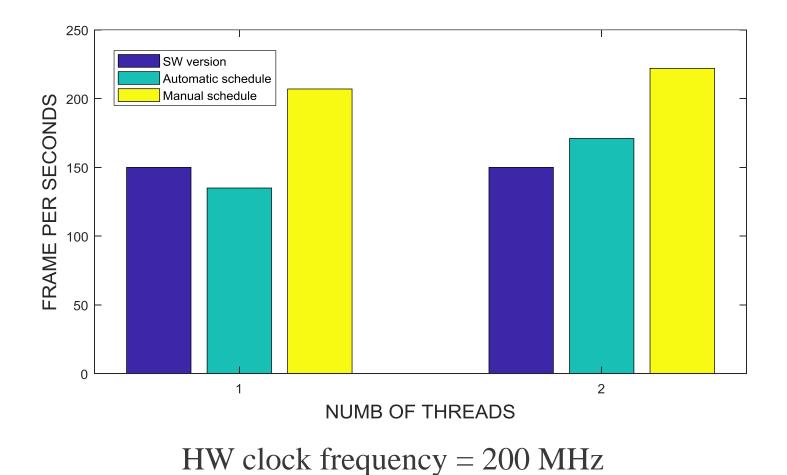
Conclusion and future work

Integration of existing tools for
deadlock-free code generation and hardware acceleration ●

Manual strategy of hardware calls can improve performance
by asynchronous hardware invocations ●

PREESM evolution:
automatic generation of code for hardware generation ●

Integration with ARTICo3 for parallel processing speed up, flexibility,
resource multiplexing in time, fault tolerance, energy efficient execution ●

● done

● on going

Thank you for your attention