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
Introduction

Embedded World

...and many others
Heterogeneous devices

- App µP
- App µP
- App µP
- App µP

Real-time µP

Real-time µP

FPGA

Logic gates, RAM, DSP

DDR

GPU

ReCoSoC July 12-14, 2017
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
GOAL

- Heterogeneous devices (HW + SW)
- Easy to program
- Task scheduling
- Hardware accelerators generation
Main Contribution

Rapid prototyping of software applications enabling:

• deadlock-free code generation using PiSDF MoC

• custom hardware acceleration and generation
Video Processing Application

Sobel
Actors Specification

Parameters

Actors
Exposed parallelism

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

Solution Gantt

- Core0: display (x1)(30-130-10)
- Core1: Read_YUV (x1)(0-100-)
- Core2: Sobel_0 (x1)(213-313-1)
- Core3: Split (x1)(101-201-100)

- Core0: Sobel_3 (x1)(212-312-1)
- Core1: Sobel_1 (x1)(213-313-1)
- Core2: Sobel_5 (x1)(313-413-1)
- Core3: Sobel_2 (x1)(213-313-1)

- Core0: Sobel_4 (x1)(312-412-1)
- Core1: Sobel_7 (x1)(313-413-1)
- Core2: Merge (x1)(425-525-1)
- Core3: Sobel_6 (x1)(313-413-1)
Heterogeneous System

Zynq-7000 family

Processing System

ARM® CoreSight™

Cortex-A9 MPCore

Cortex-A9 MPCore

AMBA Switches

ACP

HP

System gates, DSP, RAM

Programmable Logic
High Level Synthesis and SDSoc

C, C++, OpenCL

High-Level Syntax

SDSoC Environment

APPLICATION

Bitstream Generation

Developer

Hardware Drivers

ReCoSoC
July 12-14, 2017
4 hardware accelerators

\[ \approx 5.25 \text{ ms} \]
Asynchronous calls

1 SW thread
[using only one ARM μP]

2 SW threads
[using two ARM μP]

≈1,9 ms

4 hardware accelerators
## Results

### 1 THREAD – 8 SLICES – 4 ACCELERATORS

<table>
<thead>
<tr>
<th>HW freq. [MHz]</th>
<th>142.86</th>
<th>166.67</th>
<th>200</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDSoC HW handling</td>
<td>115 fps</td>
<td>125 fps</td>
<td>135 fps</td>
</tr>
<tr>
<td>Manual handling</td>
<td>200 fps</td>
<td>205 fps</td>
<td>207 fps</td>
</tr>
</tbody>
</table>

### 2 THREAD – 8 SLICES – 4 ACCELERATORS

<table>
<thead>
<tr>
<th>HW freq. [MHz]</th>
<th>142.86</th>
<th>166.67</th>
<th>200</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDSoC HW handling</td>
<td>150 fps</td>
<td>160 fps</td>
<td>171 fps</td>
</tr>
<tr>
<td>Manual handling</td>
<td>211 fps</td>
<td>218 fps</td>
<td>222 fps</td>
</tr>
</tbody>
</table>
Comparison

HW clock frequency = 200 MHz

FRAME PER SECONDS

NUMB OF THREADS

SW version
Automatic schedule
Manual schedule
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