A Flexible Hybrid Simulation Platform Targeting Multiple Configurable Processors SoC

TIMA Laboratory

*Hao Shen Frédéric Pétrot
System Level Synthesis Group, TIMA Laboratory,
CNRS/Grenoble INP/UJF
43, Avenue Félix Viallet, 38031, Grenoble, France

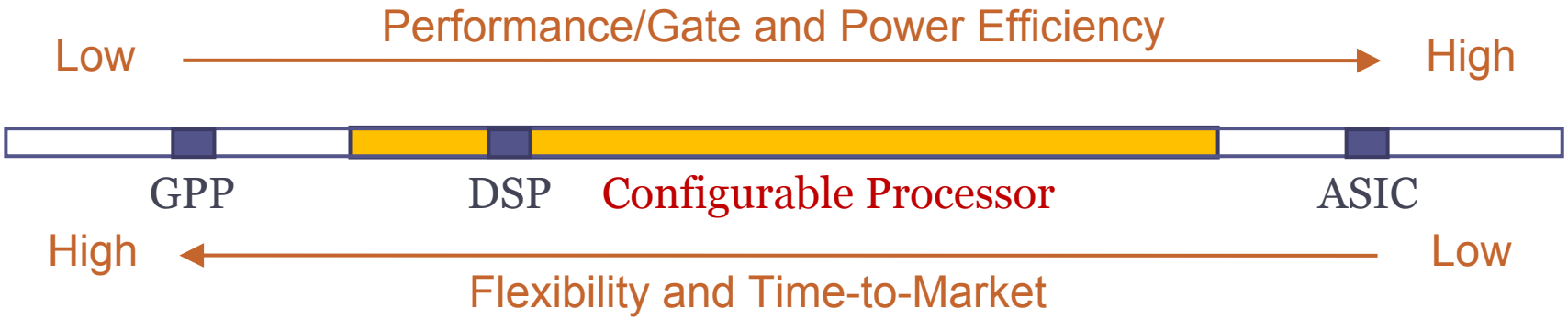
Outline

- Introduction of Configurable Processor
- Hybrid Platform Overview
- Hybrid Platform Realization Details
- Experimental Results
- Conclusion and Future Works

Outline

- Introduction of Configurable Processor
- Hybrid Platform Overview
- Hybrid Platform Realization Details
- Experimental Results
- Conclusion and Future Works

Configurable Processors

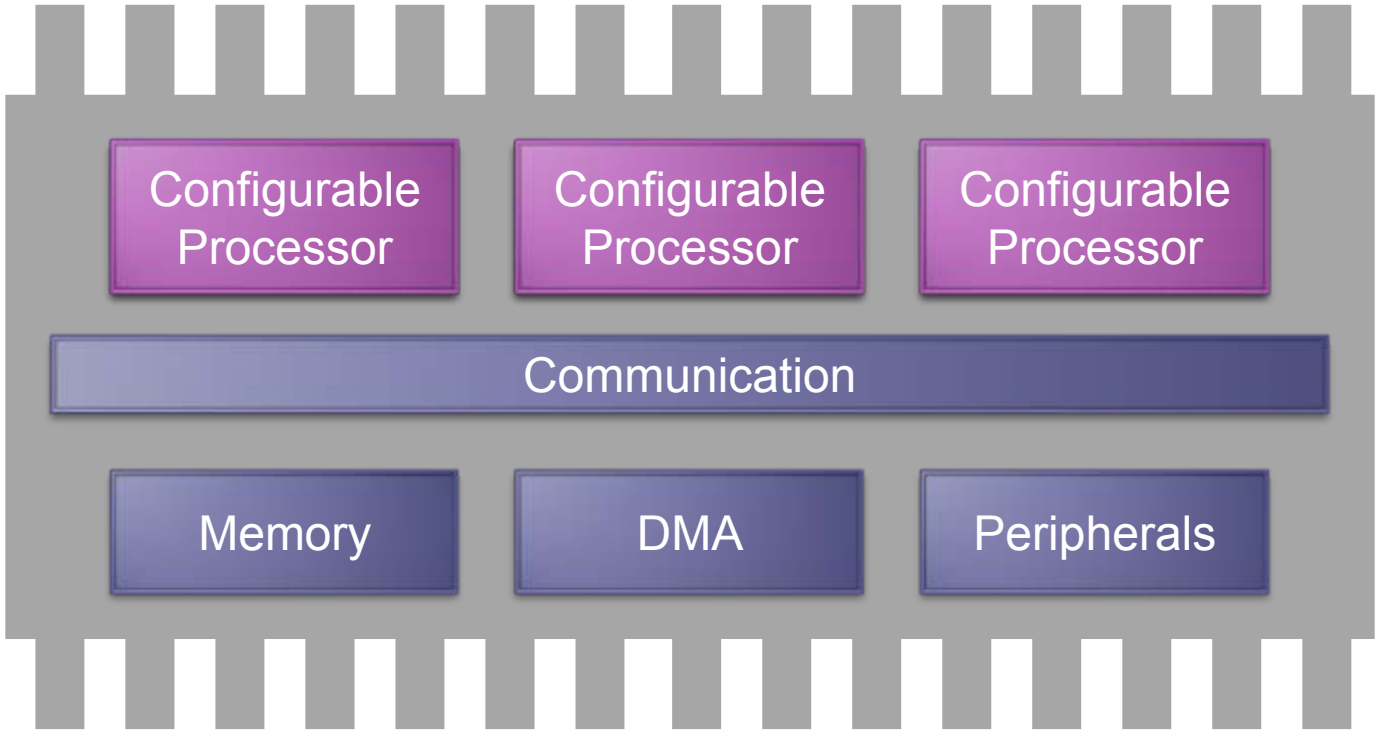


Tool chain supports

- Automatic generation of dedicated compilers for extended instruction sets
- Automatic generation of simulation and profiling environments
- RTL code available for synthesis

ARC CoWare The ESL Design Leader tensilica

MPSoC Architecture



How to program an MPSoC with configurable processors?

- **Different processors with different configurations**
- **The operating system and device driver porting difficulty**

Motivation of the Hybrid Simulation

Modifications during the design space exploration

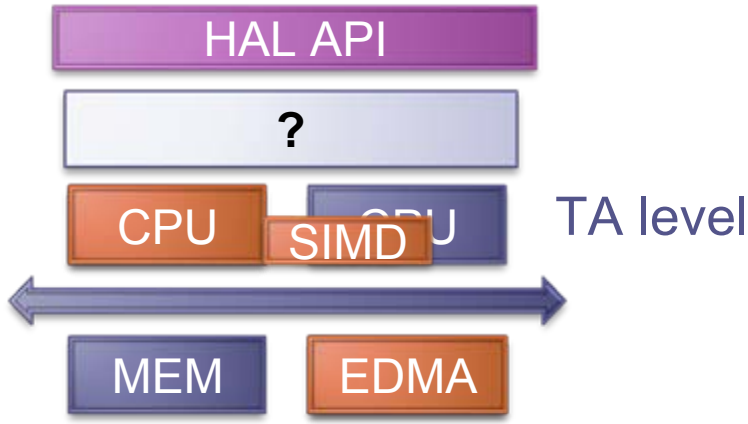
- MPSoC architecture modification
- Processor configuration modification
- Hardware module (I/O device) realization and interfaces modification

Hardware dependent Software (HdS) requirements

- Flexibility (as less assembly code as possible & no hard coded I/O address)
- Very easy to be integrated with the SystemC TLM modules

Solution: Hybrid Simulation

- Realize the HAL and Drivers with SystemC modules



Outline

- Introduction of Configurable Processor
- Hybrid Platform Overview
- Hybrid Platform Realization Details
- Experimental Results
- Conclusion and Future Works

What Is Hybrid Simulation Platform?

Semi-hosting functional call

- A specific instruction that is only valid within the ISS
- The ISS calls a host hooking function (handler)
- Used by lots of simulators such as the SPIM MIPS simulator

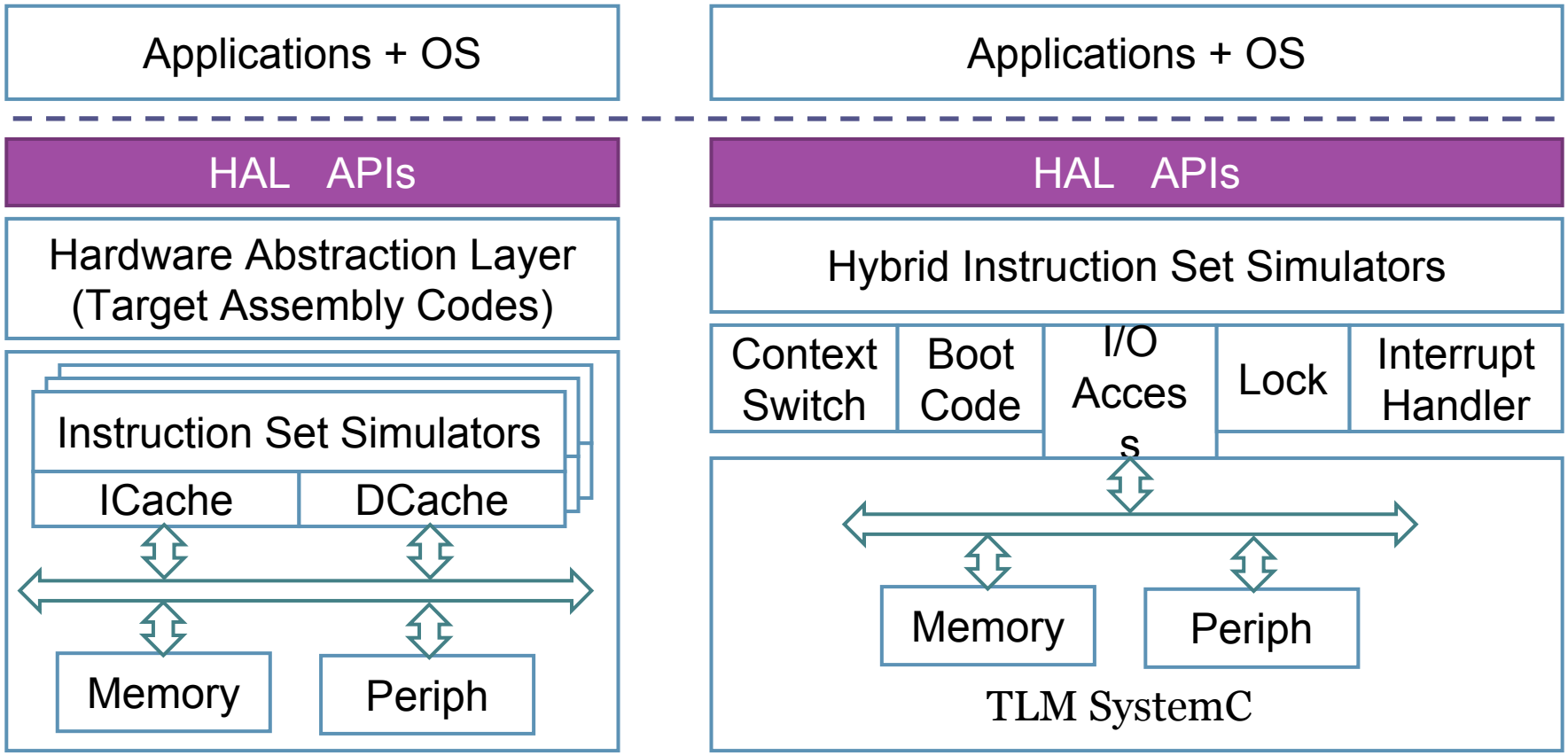
Realize most of HAL APIs with SystemC modules

- Avoid processor related assembly codes
- Avoid HdS porting and modification works

Integrate I/O devices with their drivers

- Avoid inconsistent register access
- Avoid the use of I/O addresses
- Compatible with SystemC TLM

Hybrid Simulation MPSoC Platform



Outline

- Introduction of Configurable Processor
- Hybrid Platform Overview
- Hybrid Platform Realization Details
- Experimental Results
- Conclusion and Future Works

Hybrid Simulation Platform Details

Checking and modifying processor statuses

- The ISS should provide the register get and set functions
  - `get_registers` and `set_registers`
- The simulation platform should provide the memory access functions
  - `peek_physical` and `poke_physical`

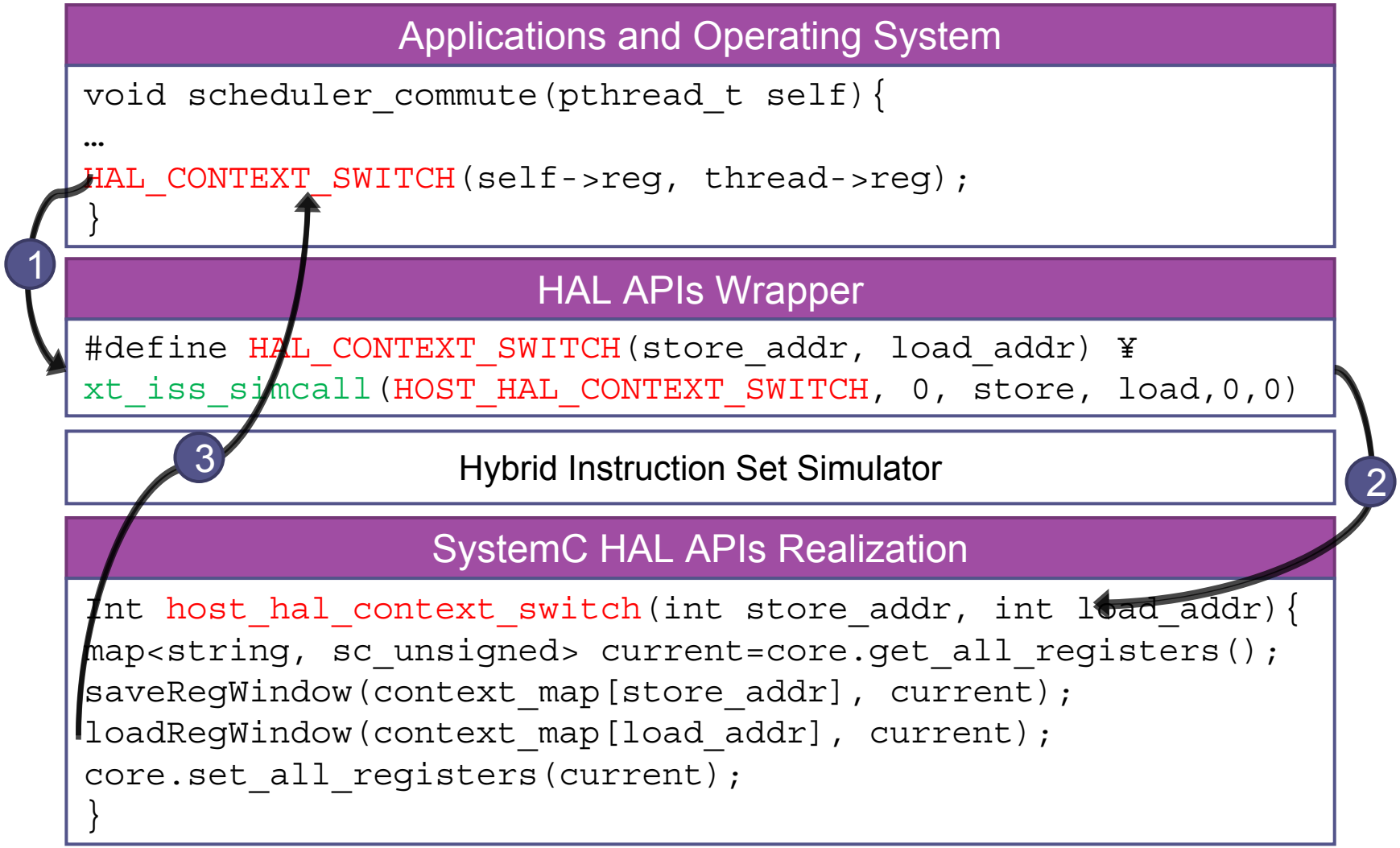
Context switch

- APIs: `HAL_SAVE_REGISTERS`, `HAL_CONTEXT_INIT`, `HAL_CONTEXT_SWITCH` and `HAL_CONTEXT_LOAD`
- Assembly code => SystemC realization

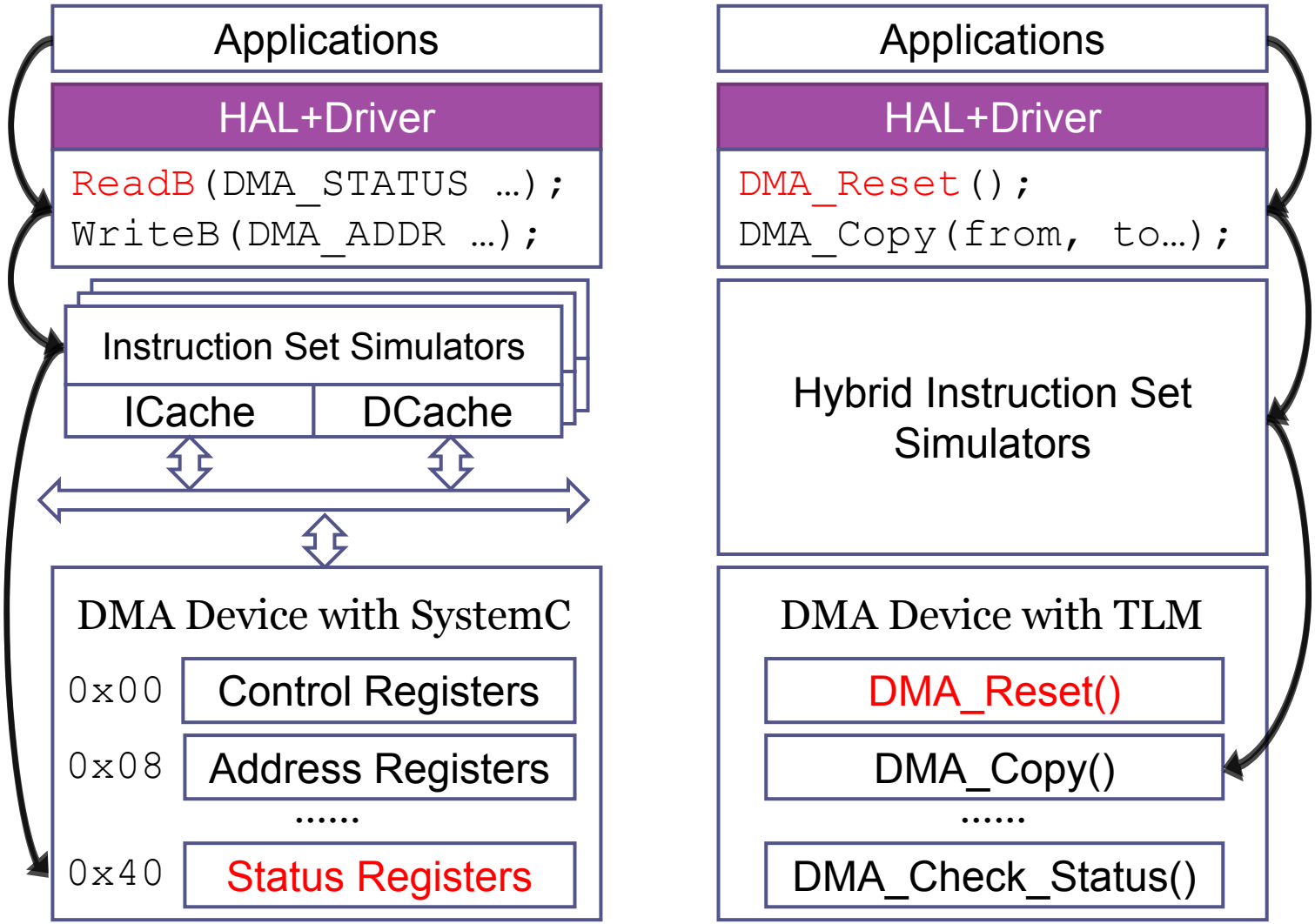
I/O devices

- Register access => TLM functional call

HAL APIs Realization



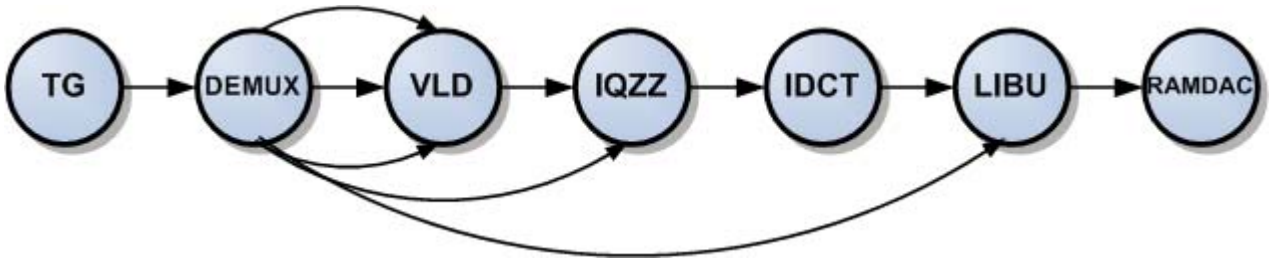
Driver Hybrid Simulation



Outline

- Introduction of Configurable Processor
- Hybrid Platform Overview
- Hybrid Platform Realization Details
- Experimental Results
- Conclusion and Future Works

Motion-JPEG Decoder Example



Motion-JPEG Application

- A multimedia format with separately compressed JPEG images

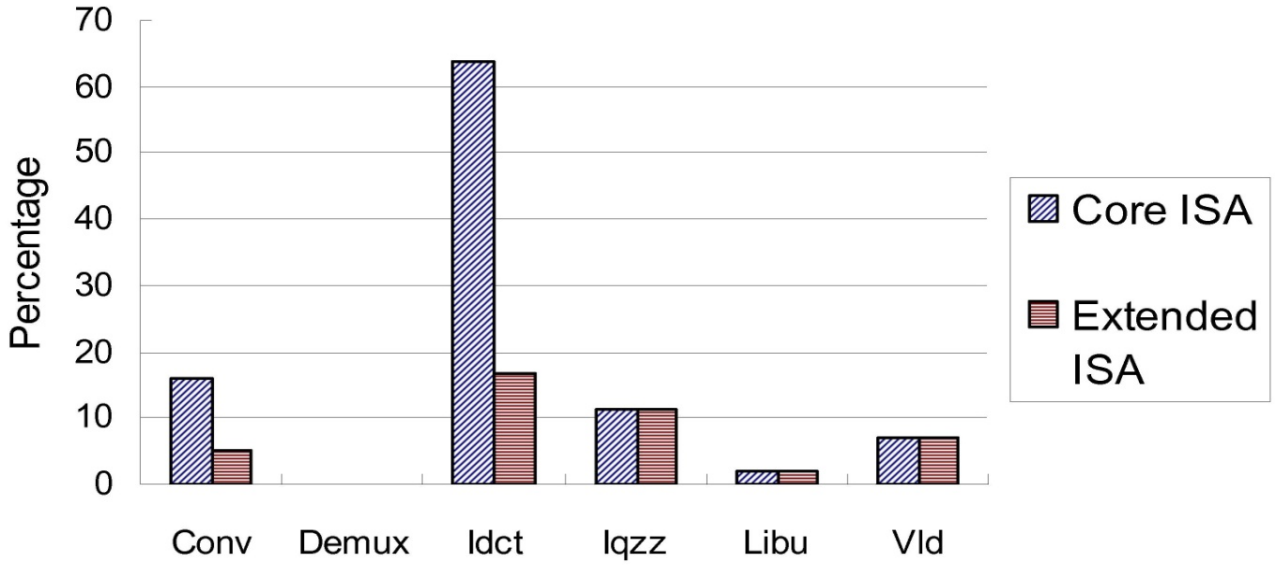
Mutek operating system with SMP support

- Open source and lightweight
- POSIX Thread APIs

MPSoC architecture modeling

- Xtensa LX2 configurable processor
- Xtensa SystemC (XTSC) platform

Accuracy of Hybrid Platform



Estimation accuracy of the hybrid simulation platform

	Hybrid Platform	Traditional ISS Platform	Difference
Extended Instructions Acceleration Ratio	1.71	1.53	10%

DMA Transfer Example

Memory copy using the DMA device

- Use peek_physical and poke_physical for memory access
- Timing annotation based on transfer size

```
int ta_dma::dma_ss_copy ( xtsc_core &core,
                          int from, int to, int size)
{
    // Copy original data to DMA buffer.
    core.peek_physical(from, size, buff);
    // Consume time for this copy operation
    consume(size/4*time_per_word);
    // Copy DMA buffer to Target Memory
    core.poke_physical(to, size, buff);
    return size;
}
```

Development Effort

Hybrid simulation can save development effort

- Use *Source Lines Of Code* **(SLOC)** to evaluate design effort
- Useful for both HdS and driver development
- Simplify the I/O device modeling

SLOC for both hybrid and traditional simulation platform

Module Name	SLOC of the hybrid platform	SLOC of the traditional platform
DMA	79	267
context	202	257 (206 lines of asm code)
stack	48	14 (6 lines of asm code)

Outline

- Introduction of Configurable Processor
- Hybrid Platform Overview
- Hybrid Platform Realization Details
- Experimental Results
- Conclusion and Future Works

Conclusion and Future Works

A hybrid MPSoC simulation platform

- Simplify the HdS adaptation for processor and architecture modifications
- Simplify the I/O device modeling and the driver adaptation
- Specially useful for early design stages

Future works

- HdS adaptation for Linux OS
- Open source hybrid ISS: such as QEMU

Questions & Answers

Hao.Shen@imag.fr