Novel Task Migration Framework on Configurable Heterogeneous MPSoC Platforms

Hao Shen Frédéric Pétrot*
System Level Synthesis Group, TIMA Laboratory
CNRS/Grenoble INP/UJF
43, Avenue Félix Viallet, 38031, Grenoble, France
hao.shen@imag.fr frederic.petrot@imag.fr

Outline

Outline

Trends (Power and Cost Constraints for Embedded Systems)

Multiple Processor System-on-Chip (MPSoC)

- Provide high Thread Level Parallelism (TLP)
- Provide high performance
- Power consumption and cost advantages

Configurable processor

- Different extended instruction set for different processors
- Dedicated compiler for each extended instruction set
- Provide high Instruction Level Parallelism (ILP)
- Power consumption and cost advantages

Heterogeneity

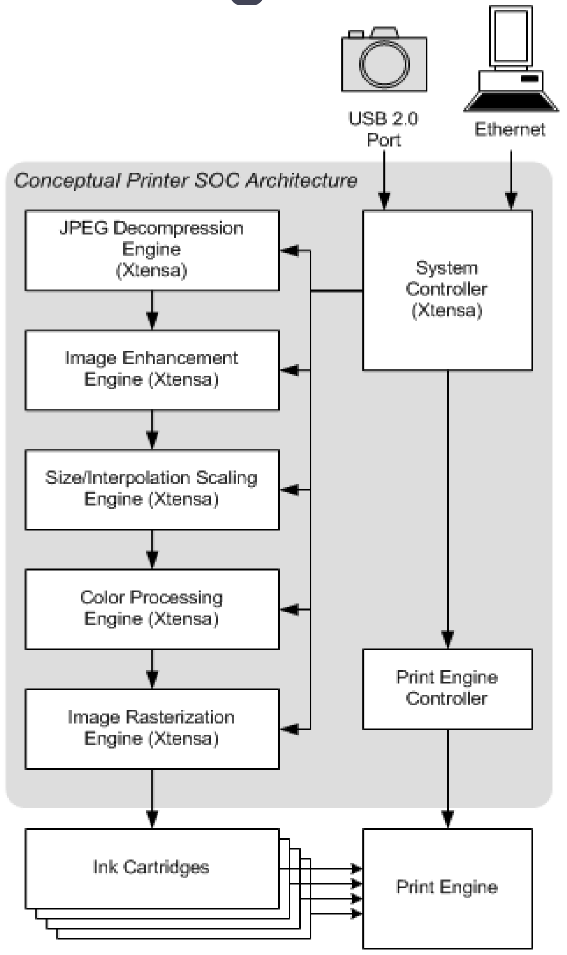
- Different configurable processor in a MPSoC for different kinds of applications
- Power consumption and cost advantages

Configurable Processors

Application Specific Instruction-set Processors (ASIP)	• Configuration for one application or a group of applications
Provide high Instruction Level Parallelism (ILP)	• SIMD instructions, MIMD instructions, specific instructions (such as Multiply-accumulate)
Configurable processors VS. RISC processors	• Provide higher performance
Configurable processors VS. ASIC	• More flexibility and shorter time-to-market

tensilica CoWare The ESL Design Leader ARC

One Existing Heterogeneous Multiple Configurable SoC Example


USB 2.0 Port
Ethernet
Conceptual Printer SOC Architecture
JPEG Decompression Engine (Xtensa)
System Controller (Xtensa)
Image Enhancement Engine (Xtensa)
Size/Interpolation Scaling Engine (Xtensa)
Color Processing Engine (Xtensa)
Image Rasterization Engine (Xtensa)
Print Engine Controller
Ink Cartridges
Print Engine

EPSON®

- General printer solution
- 6 heterogeneous Xtensa processors
- Fixed task mapping
- Communication with FIFOs

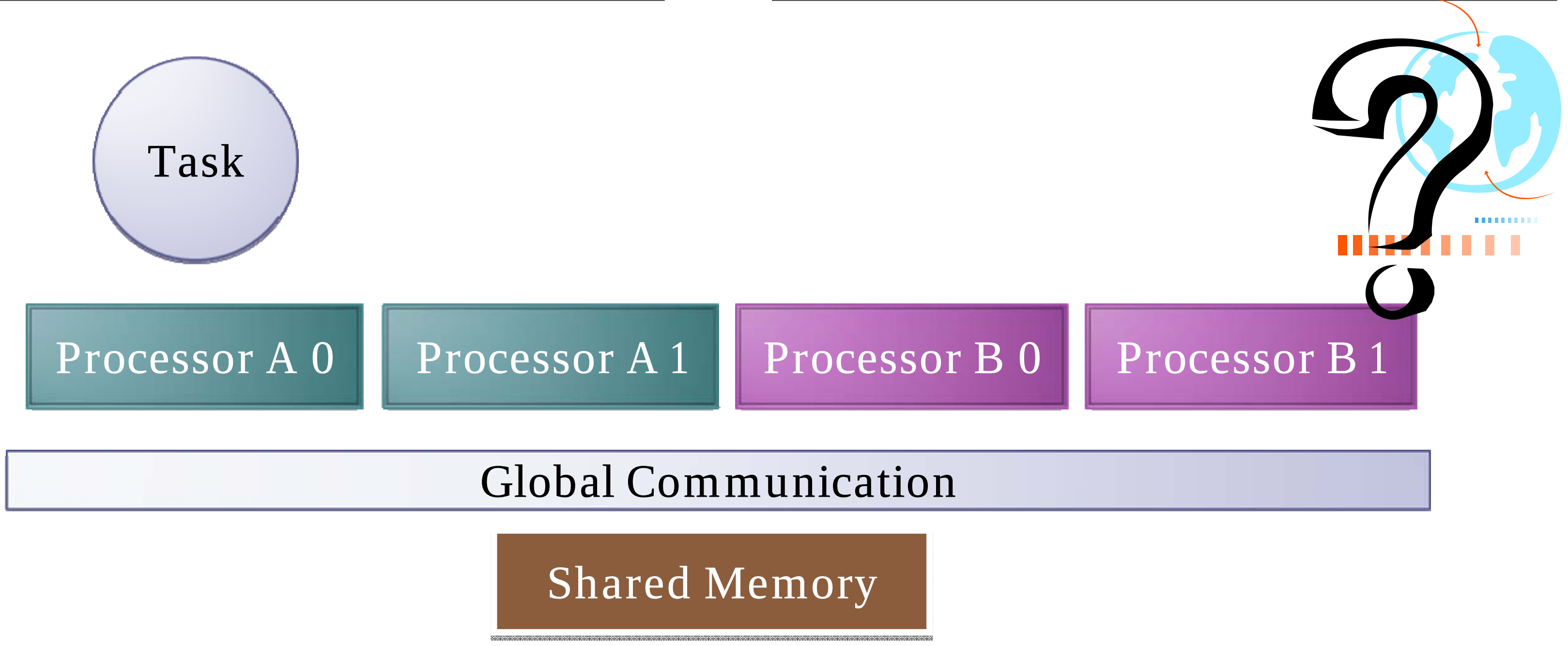
What is the Problem of Configurable Processors?

Instruction Set

- Core instruction set
- Extended instruction set

Register File

- Core registers
- Extended registers



Contribution of This Work

Task migration framework

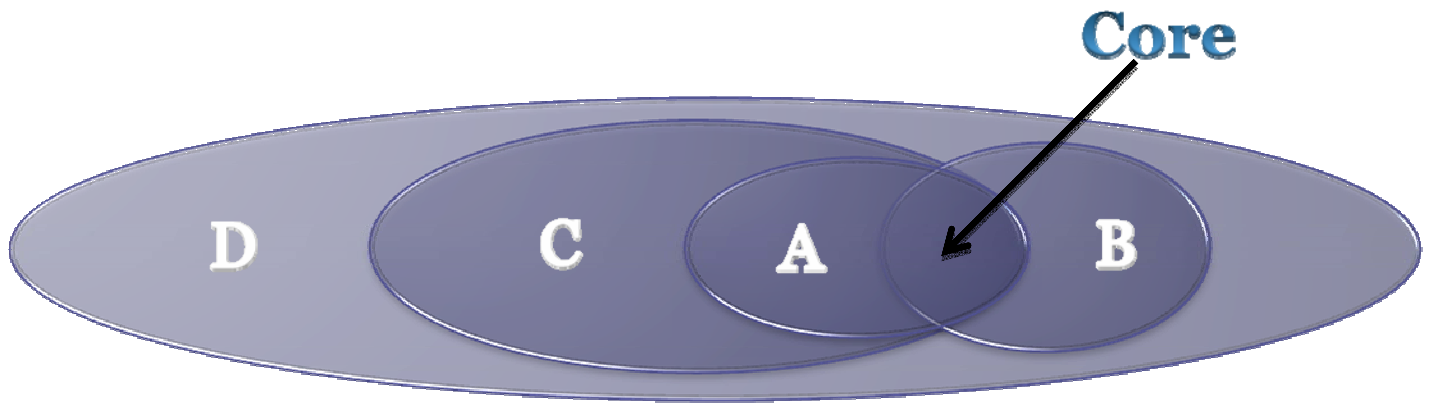
- Support heterogeneous multiple configurable processor SoC
- Present realization details of this framework
- Add some formal description

Task migration algorithms

- Compare the efficiency of each algorithm
- Compare the migration cost of each algorithm

Outline

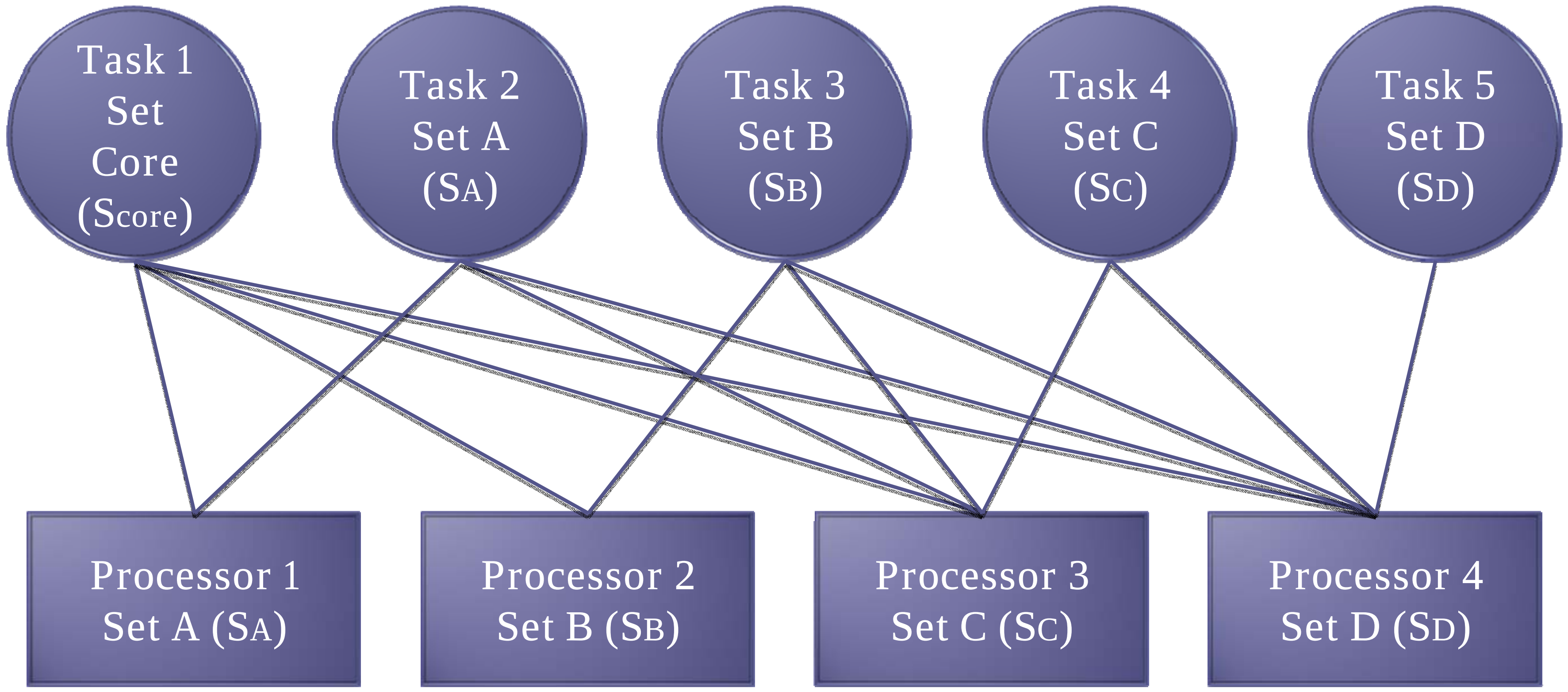
Instruction Set Relationship



$\mathbb{S} = \{S_{core}, S_A, S_B, S_C, S_D\}$

$S_{core} \subset S_A, \ldots, S_{core} \subset S_D, S_A \subset S_C, S_A \subset S_D, S_C \subset S_D, S_B \subset S_D$

Relationship Between Tasks and Processors



Instruction Set Identification

Assign ID for each CPU type (CPU_ISA_ID)

- Indicate the instruction set which it realizes

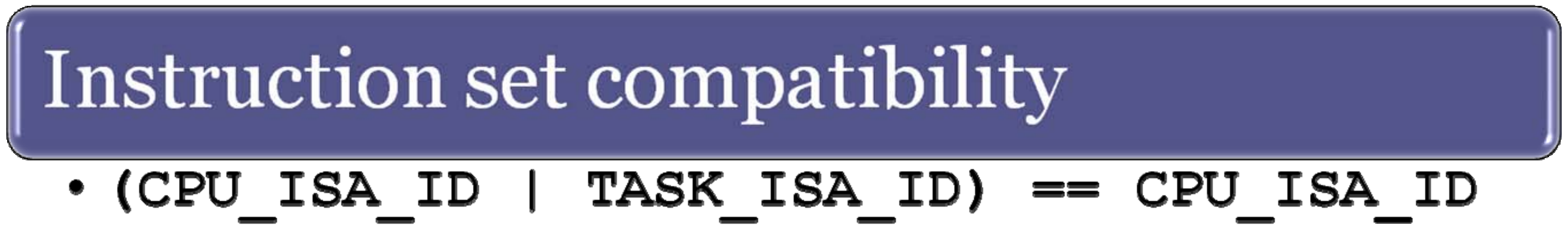
Assign ID for each task type (TASK_ISA_ID)

- Indicate the instruction set which it uses

Use bit operation to accelerate scheduling

- Use one bit to represent a standalone instruction set
- Test compatibility by using bit operations
- Save storage space for the OS realization

Scheduler Realization



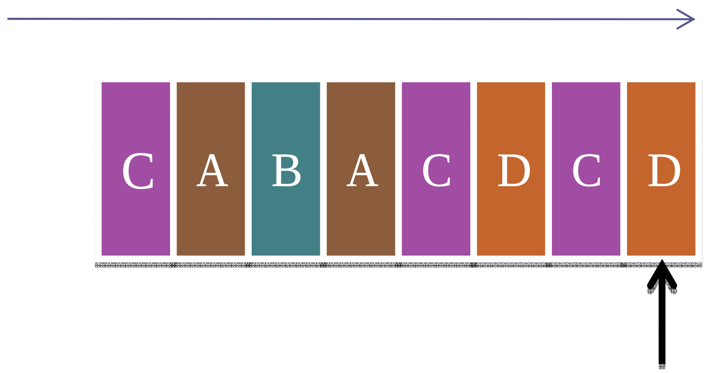
Instruction set compatibility

- (CPU_ISA_ID | TASK_ISA_ID) == CPU_ISA_ID

Set	CPU_ISA_ID	Compatible Tasks	TASK_ISA_ID
Core	0x0000	Core	0x0000
A	0x0001	Core and A	0x0000, 0x0001
B	0x0010	Core and B	0x0000, 0x0010
C	0x0101	Core, A and C	0x0000, 0x0001, 0x0101
D	0x1111	Core, A, B, C and D	0x0000, 0x0001, 0x0010, 0x0101, 0x1111

Outline

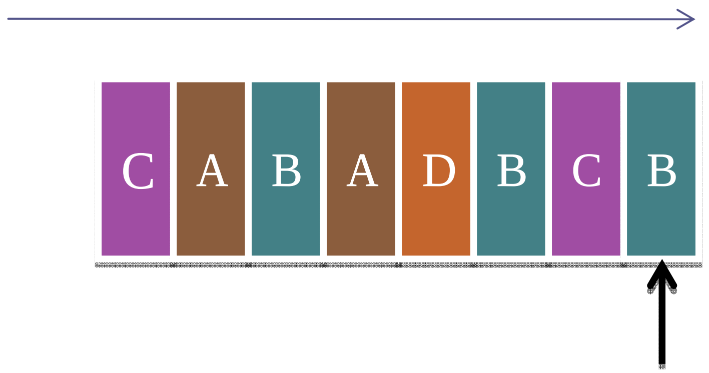
First Match First Serve Algorithm



Available Processor (SA)

```
For j=1 to $|\mathbb{T}_{queue}|$ in the FIFO order
    If $c(T_j, P_i) \in \mathbb{C}$ // $T_j$ is compatible with $P_i$
        Choose the task $T_j$
```

Most Compatible Algorithm

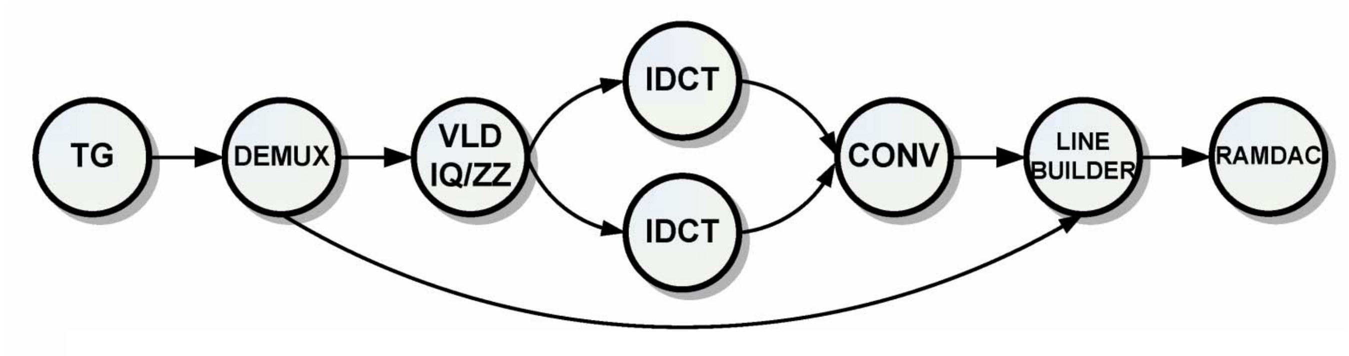


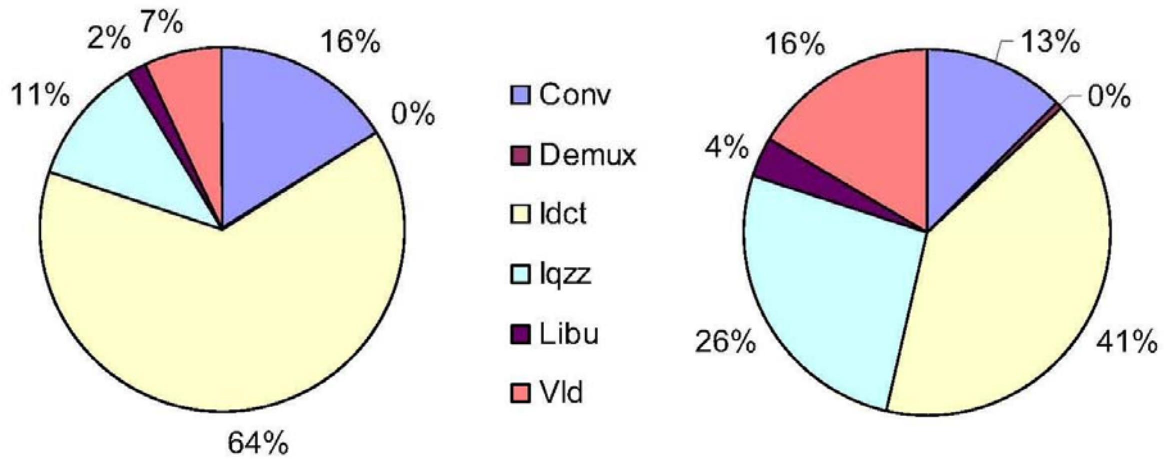
Available Processor (SD)

```
Forall $T_j \in \mathbb{T}_{queue}$
    If $c(T_j, P_i) \in \mathbb{C}$ // $T_j$ is compatible with $P_i$
        $\mathbb{T}_{candidate} = \mathbb{T}_{candidate} \cup T_j$ //Add $T_j$ to candidate set
If $\mathbb{T}_{candidate} \neq \emptyset$
    choose the task $T = \min(D(T_j \in \mathbb{T}_{candidate}, P_i))$
```

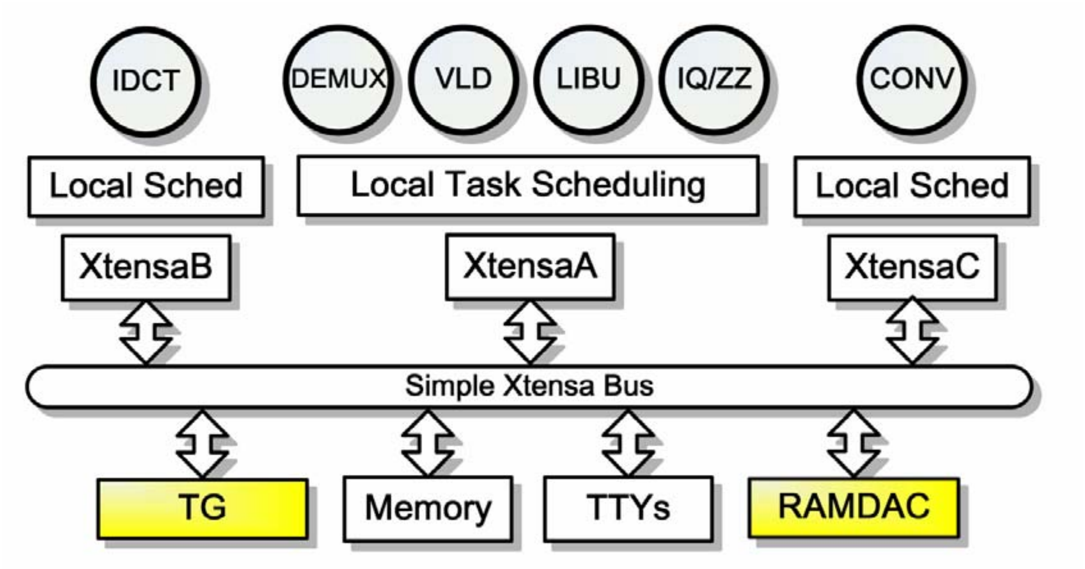
Outline

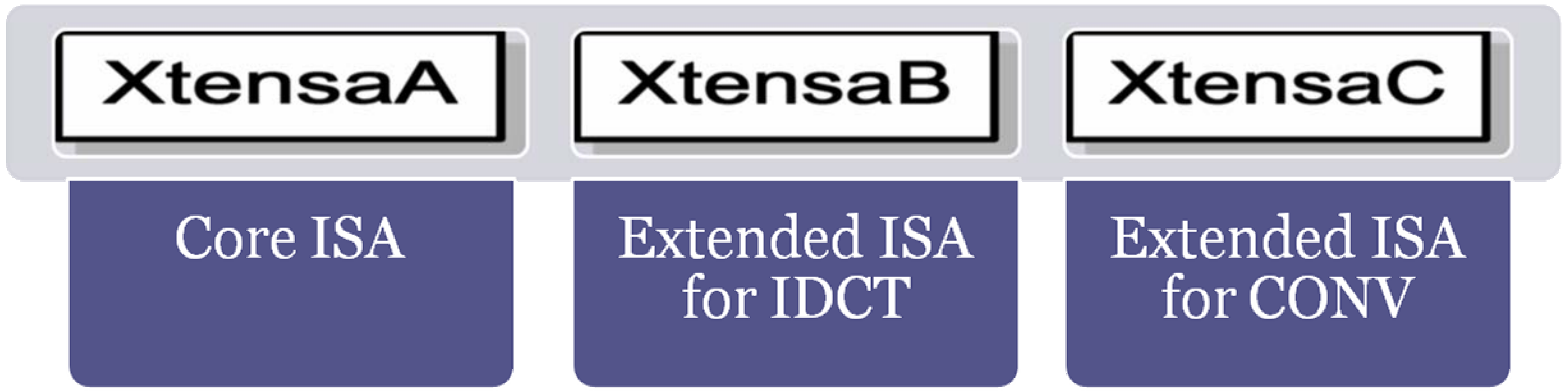
Motion-JPEG Decoder Example and the Optimization





Heterogeneous MPSoC Architecture





Performance and Cost Advantages

	Fixed Task Assignment	FMFS Algorithm	Most Comp Algorithm	SMP Task Scheduling
Frame/s	1.44	2.88	2.70	2.88
CPUs Gate number	341,773	341,773	341,773	611,295
Perf/Cost	0.50	1.00	0.94	0.56

Heterogeneous architecture VS. homogeneous architecture

- May achieve the same performance
- May need smaller chip size (higher performance/cost ratio)

Task migration VS. fixed task assignment

- Need the same chip size
- Provide higher system performance (shorten the processor waiting time)

Different task migration algorithms have different performance

- FMFS requires less computation resource during migration

Outline

Conclusion and Future Works

A task migration framework

- Support configurable processors
- Support heterogeneous MPSoC architectures
- Support several migration algorithms

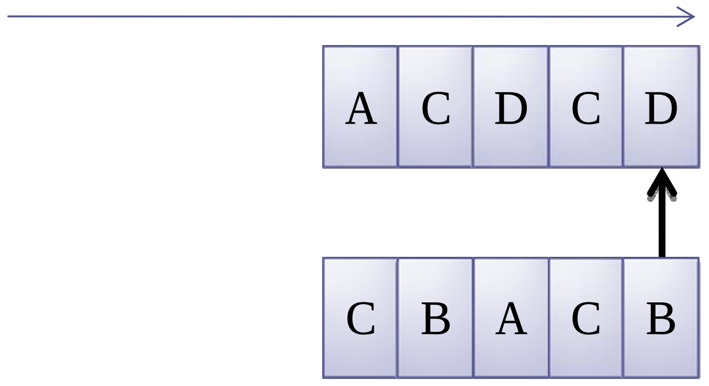
Future works

- Formalization this framework
- More complex case studies (benchmarks)

Questions & Answers

Hao.Shen@imag.fr Frederic.Petrot@imag.fr

Priority Based Most Compatible Algorithm



Available Processor (S_B)

```
For k = 1 to n // n queues with different priorities
    Forall $T_j \in \mathbb{T}_{queue}$
        If $c(T_j, P_i) \in \mathbb{C}$ // $T_j$ is compatible with $P_i$
            $\mathbb{T}_{candidate} = \mathbb{T}_{candidate} \cup T_j$
    If $\mathbb{T}_{candidate} \neq \emptyset$
        choose the task $T = \min(D(T_j \in \mathbb{T}_{candidate}, P_i))$
```

Time

- 20’ presentation
- 5’ question