A Generic ISS API for timed and untimed Simulation and Debug of MP2-SoCs

Nicolas Pouillon, Alexandre Becoulet, Aline Vieira de Mello, François Pêcheux and Alain Greiner

Laboratoire d'informatique de Paris 6

2009-06-25

Outline

Context

SoCLib foundation

SoCLib is a project

- funded by the French *Agence Nationale pour la Recherche*
- involving 6 industrial companies and 10 laboratories

- Magillem Design Services
- Orange Business Services
- ST Microelectronics
- Thales Communications
- Thomson R&D France
- TurboConcept

- CEA-LIST
- CEA-LETI
- CITI
- Telecom Paris'Tech
- INRIA Futurs
- IRISA
- Lester
- LIP6
- LIS
- TIMA

Context

SoCLib goals

SoCLib is a virtual prototyping platform aiming

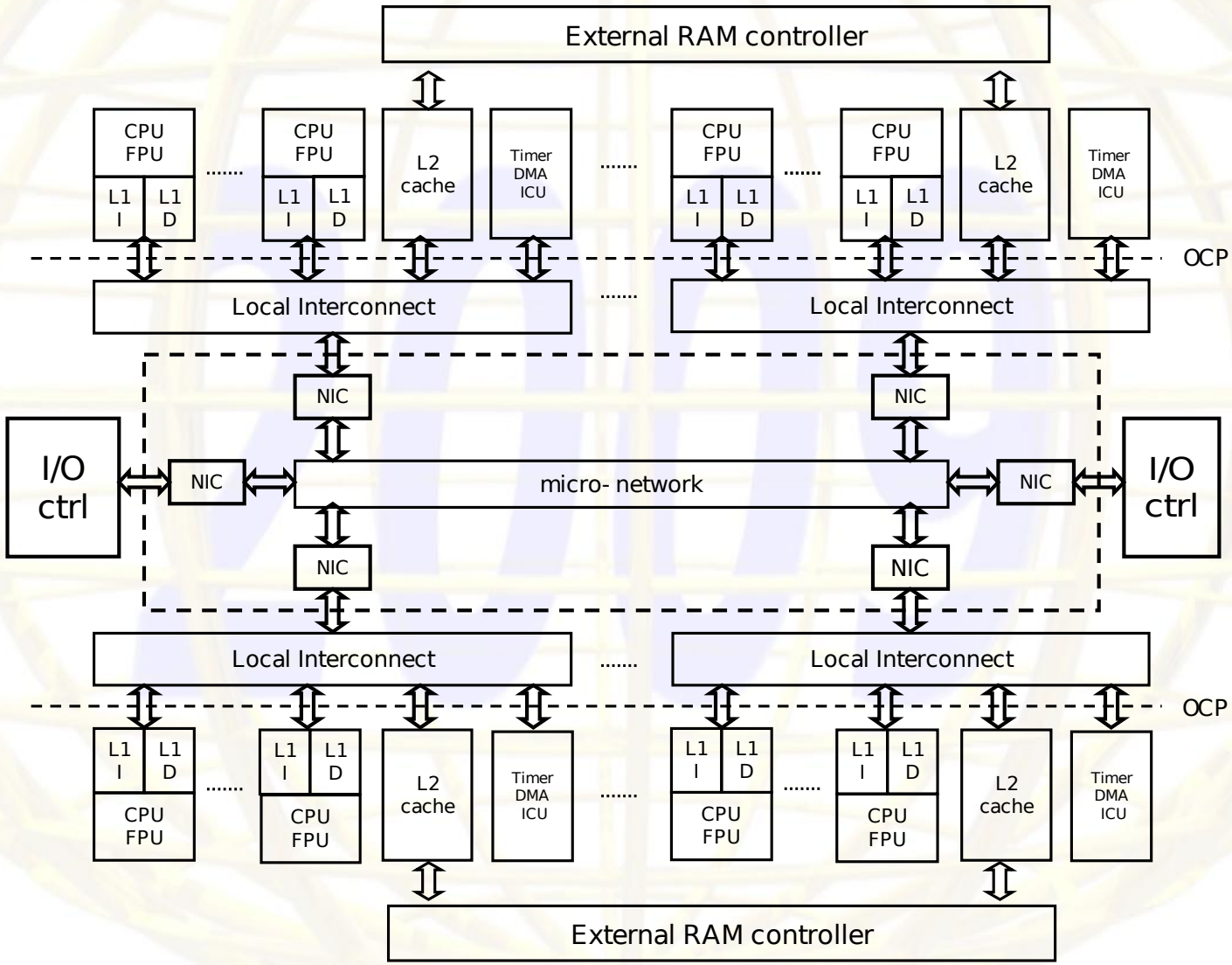
- fast and complete MP2SoC design (interconnects, processors, coprocessors, caches, peripheral controllers, ...)
- early performance evaluation
- instrumentation
- precise debugging

At different levels of simulation:

- Cycle-accurate (CABA)
- Transaction level with time (TLM-DT)
- Untimed transaction level (TLM)

Context

A sample platform



Outline

Problem: processor models

We need processor simulation models in the library

Problem: processor models

We need processor simulation models in the library

Problem: processor models

We need processor simulation models in the library

CABA Mips32	TLM Mips32

Problem: processor models

We need processor simulation models in the library

CABA Mips32	**TLM Mips32**
CABA PPC405	

Problem: processor models

We need processor simulation models in the library

CABA Mips32	TLM Mips32
CABA PPC405	TLM PPC405

Problem: processor models

We need processor simulation models in the library

CABA Mips32	**TLM Mips32**	**TLM−DT Mips32**
CABA PPC405	**TLM PPC405**	

Problem: processor models

We need processor simulation models in the library

CABA Mips32	TLM Mips32	TLM−DT Mips32
CABA PPC405	TLM PPC405	TLM−DT PPC405

Problem: processor models

We need processor simulation models in the library

	CABA	TLM	TLM−DT
Mips32	CABA Mips32	TLM Mips32	TLM−DT Mips32
PPC405	CABA PPC405	TLM PPC405	TLM−DT PPC405

Problem: processor models

We need processor simulation models in the library

	CABA	TLM	TLM−DT
Mips32	CABA Mips32	TLM Mips32	TLM−DT Mips32
PPC405	CABA PPC405	TLM PPC405	TLM−DT PPC405
SparcV8	CABA SparcV8		

Problem: processor models

We need processor simulation models in the library

	CABA	TLM	TLM−DT
Mips32	CABA Mips32	TLM Mips32	TLM−DT Mips32
PPC405	CABA PPC405	TLM PPC405	TLM−DT PPC405
SparcV8	CABA SparcV8	TLM SparcV8	TLM−DT SparcV8

Problem: processor models

We need processor simulation models in the library

	CABA	TLM	TLM−DT
Mips32	CABA Mips32	TLM Mips32	TLM−DT Mips32
PPC405	CABA PPC405	TLM PPC405	TLM−DT PPC405
SparcV8	CABA SparcV8	TLM SparcV8	TLM−DT SparcV8
ARM−v6k	CABA ARM−v6k		
ARM7tdmi	CABA ARM7tdmi		
ARM966	CABA ARM966		

Problem: processor models

We need processor simulation models in the library

	CABA	TLM	TLM−DT
Mips32	CABA Mips32	TLM Mips32	TLM−DT Mips32
PPC405	CABA PPC405	TLM PPC405	TLM−DT PPC405
SparcV8	CABA SparcV8	TLM SparcV8	TLM−DT SparcV8
ARM−v6k	CABA ARM−v6k	TLM ARM−v6k	TLM−DT ARM−v6k
ARM7tdmi	CABA ARM7tdmi	TLM ARM7tdmi	TLM−DT ARM7tdmi
ARM966	CABA ARM966	TLM ARM966	TLM−DT ARM966

Problem: processor models

We need processor simulation models in the library

	CABA	TLM	TLM−DT
Mips32	CABA Mips32	TLM Mips32	TLM−DT Mips32
PPC405	CABA PPC405	TLM PPC405	TLM−DT PPC405
SparcV8	CABA SparcV8	TLM SparcV8	TLM−DT SparcV8
ARM−v6k	CABA ARM−v6k	TLM ARM−v6k	TLM−DT ARM−v6k
ARM7tdmi	CABA ARM7tdmi	TLM ARM7tdmi	TLM−DT ARM7tdmi
ARM966	CABA ARM966	TLM ARM966	TLM−DT ARM966
TMS320	CABA TMS320	TLM TMS320	TLM−DT TMS320
MPC7447A	CABA MPC7447A	TLM MPC7447A	TLM−DT MPC7447A
MicroBlaze	CABA MicroBlaze	TLM MicroBlaze	TLM−DT MicroBlaze
Nios2	CABA Nios2	TLM Nios2	TLM−DT Nios2
LM32	CABA LM32	TLM LM32	TLM−DT LM32

Problem: processor models

We need processor simulation models in the library

	CABA	TLM	TLM−DT
Mips32	CABA Mips32	TLM Mips32	TLM−DT Mips32
PPC405	CABA PPC405	TLM PPC405	TLM−DT PPC405
SparcV8	CABA SparcV8	TLM SparcV8	TLM−DT SparcV8
ARM−v6k	CABA ARM−v6k	TLM ARM−v6k	TLM−DT ARM−v6k
ARM7tdmi	CABA ARM7tdmi	TLM ARM7tdmi	TLM−DT ARM7tdmi
ARM966	CABA ARM966	TLM ARM966	TLM−DT ARM966
TMS320	CABA TMS320	TLM TMS320	TLM−DT TMS320
MPC7447A	CABA MPC7447A	TLM MPC7447A	TLM−DT MPC7447A
MicroBlaze	CABA MicroBlaze	TLM MicroBlaze	TLM−DT MicroBlaze
Nios2	CABA Nios2	TLM Nios2	TLM−DT Nios2
LM32	CABA LM32	TLM LM32	TLM−DT LM32

→ 33 processor models !

Problem: processor models

We need to model processors, with different caches:

	CABA	TLM	TLM−DT
Mips32	CABA Mips32	TLM Mips32	TLM−DT Mips32
PPC405	CABA PPC405	TLM PPC405	TLM−DT PPC405
SparcV8	CABA SparcV8	TLM SparcV8	TLM−DT SparcV8
ARM−v6k	CABA ARM−v6k	TLM ARM−v6k	TLM−DT ARM−v6k
ARM7tdmi	CABA ARM7tdmi	TLM ARM7tdmi	TLM−DT ARM7tdmi
ARM966	CABA ARM966	TLM ARM966	TLM−DT ARM966
TMS320	CABA TMS320	TLM TMS320	TLM−DT TMS320
MPC7447A	CABA MPC7447A	TLM MPC7447A	TLM−DT MPC7447A
MicroBlaze	CABA MicroBlaze	TLM MicroBlaze	TLM−DT MicroBlaze
Nios2	CABA Nios2	TLM Nios2	TLM−DT Nios2
LM32	CABA LM32	TLM LM32	TLM−DT LM32

→ 33 processor models !

Problem: processor models

We need to model processors, with different caches:

	Simple cache controller		
	CABA	TLM	TLM−DT
Mips32	CABA Mips32	TLM Mips32	TLM−DT Mips32
PPC405	CABA PPC405	TLM PPC405	TLM−DT PPC405
SparcV8	CABA SparcV8	TLM SparcV8	TLM−DT SparcV8
ARM−v6k	CABA ARM−v6k	TLM ARM−v6k	TLM−DT ARM−v6k
ARM7tdmi	CABA ARM7tdmi	TLM ARM7tdmi	TLM−DT ARM7tdmi
ARM966	CABA ARM966	TLM ARM966	TLM−DT ARM966
TMS320	CABA TMS320	TLM TMS320	TLM−DT TMS320
MPC7447A	CABA MPC7447A	TLM MPC7447A	TLM−DT MPC7447A
MicroBlaze	CABA MicroBlaze	TLM MicroBlaze	TLM−DT MicroBlaze
Nios2	CABA Nios2	TLM Nios2	TLM−DT Nios2
LM32	CABA LM32	TLM LM32	TLM−DT LM32

→ 33 processor models !

Problem: processor models

We need to model processors, with different caches:

	Simple cache controller		
	CABA	TLM	TLM−DT
Mips32	CABA Mips32	TLM Mips32	TLM−DT Mips32
PPC405	CABA PPC405	TLM PPC405	TLM−DT PPC405
SparcV8	CABA SparcV8	TLM SparcV8	TLM−DT SparcV8
ARM−v6k	CABA ARM−v6k	TLM ARM−v6k	TLM−DT ARM−v6k
ARM7tdmi	CABA ARM7tdmi	TLM ARM7tdmi	TLM−DT ARM7tdmi
ARM966	CABA ARM966	TLM ARM966	TLM−DT ARM966
TMS320	CABA TMS320	TLM TMS320	TLM−DT TMS320
MPC7447A	CABA MPC7447A	TLM MPC7447A	TLM−DT MPC7447A
MicroBlaze	CABA MicroBlaze	TLM MicroBlaze	TLM−DT MicroBlaze
Nios2	CABA Nios2	TLM Nios2	TLM−DT Nios2
LM32	CABA LM32	TLM LM32	TLM−DT LM32

	MMU cache controller		
	CABA	TLM	TLM−DT
Mips32	CABA Mips32	TLM Mips32	TLM−DT Mips32
PPC405	CABA PPC405	TLM PPC405	TLM−DT PPC405
SparcV8	CABA SparcV8	TLM SparcV8	TLM−DT SparcV8
ARM−v6k	CABA ARM−v6k	TLM ARM−v6k	TLM−DT ARM−v6k
ARM7tdmi	CABA ARM7tdmi	TLM ARM7tdmi	TLM−DT ARM7tdmi
ARM966	CABA ARM966	TLM ARM966	TLM−DT ARM966
TMS320	CABA TMS320	TLM TMS320	TLM−DT TMS320
MPC7447A	CABA MPC7447A	TLM MPC7447A	TLM−DT MPC7447A
MicroBlaze	CABA MicroBlaze	TLM MicroBlaze	TLM−DT MicroBlaze
Nios2	CABA Nios2	TLM Nios2	TLM−DT Nios2
LM32	CABA LM32	TLM LM32	TLM−DT LM32

→ **66** processor models !

Problem: processor models

We need to model processors, with different caches:

	Simple cache controller		
	CABA	TLM	TLM-DT
Mips32	CABA Mips32	TLM Mips32	TLM-DT Mips32
PPC405	CABA PPC405	TLM PPC405	TLM-DT PPC405
SparcV8	CABA SparcV8	TLM SparcV8	TLM-DT SparcV8
ARM-v6k	CABA ARM-v6k	TLM ARM-v6k	TLM-DT ARM-v6k
ARM7tdmi	CABA ARM7tdmi	TLM ARM7tdmi	TLM-DT ARM7tdmi
ARM966	CABA ARM966	TLM ARM966	TLM-DT ARM966
TMS320	CABA TMS320	TLM TMS320	TLM-DT TMS320
MPC7447A	CABA MPC7447A	TLM MPC7447A	TLM-DT MPC7447A
MicroBlaze	CABA MicroBlaze	TLM MicroBlaze	TLM-DT MicroBlaze
Nios2	CABA Nios2	TLM Nios2	TLM-DT Nios2
LM32	CABA LM32	TLM LM32	TLM-DT LM32

	MMU cache controller		
	CABA	TLM	TLM-DT
Mips32	CABA Mips32	TLM Mips32	TLM-DT Mips32
PPC405	CABA PPC405	TLM PPC405	TLM-DT PPC405
SparcV8	CABA SparcV8	TLM SparcV8	TLM-DT SparcV8
ARM-v6k	CABA ARM-v6k	TLM ARM-v6k	TLM-DT ARM-v6k
ARM7tdmi	CABA ARM7tdmi	TLM ARM7tdmi	TLM-DT ARM7tdmi
ARM966	CABA ARM966	TLM ARM966	TLM-DT ARM966
TMS320	CABA TMS320	TLM TMS320	TLM-DT TMS320
MPC7447A	CABA MPC7447A	TLM MPC7447A	TLM-DT MPC7447A
MicroBlaze	CABA MicroBlaze	TLM MicroBlaze	TLM-DT MicroBlaze
Nios2	CABA Nios2	TLM Nios2	TLM-DT Nios2
LM32	CABA LM32	TLM LM32	TLM-DT LM32

	Coherent MMU cache ctrl.		
	CABA	TLM	TLM-DT
Mips32	CABA Mips32	TLM Mips32	TLM-DT Mips32
PPC405	CABA PPC405	TLM PPC405	TLM-DT PPC405
SparcV8	CABA SparcV8	TLM SparcV8	TLM-DT SparcV8
ARM-v6k	CABA ARM-v6k	TLM ARM-v6k	TLM-DT ARM-v6k
ARM7tdmi	CABA ARM7tdmi	TLM ARM7tdmi	TLM-DT ARM7tdmi
ARM966	CABA ARM966	TLM ARM966	TLM-DT ARM966
TMS320	CABA TMS320	TLM TMS320	TLM-DT TMS320
MPC7447A	CABA MPC7447A	TLM MPC7447A	TLM-DT MPC7447A
MicroBlaze	CABA MicroBlaze	TLM MicroBlaze	TLM-DT MicroBlaze
Nios2	CABA Nios2	TLM Nios2	TLM-DT Nios2
LM32	CABA LM32	TLM LM32	TLM-DT LM32

→ **99** processor models !

Problem: processor models

We need to model processors, factoring things:

	Simple cache controller		
	CABA	TLM	TLM-DT
Mips32	CABA Mips32	TLM Mips32	TLM-DT Mips32
PPC405	CABA PPC405	TLM PPC405	TLM-DT PPC405
SparcV8	CABA SparcV8	TLM SparcV8	TLM-DT SparcV8
ARM-v6k	CABA ARM-v6k	TLM ARM-v6k	TLM-DT ARM-v6k
ARM7tdmi	CABA ARM7tdmi	TLM ARM7tdmi	TLM-DT ARM7tdmi
ARM966	CABA ARM966	TLM ARM966	TLM-DT ARM966
TMS320	CABA TMS320	TLM TMS320	TLM-DT TMS320
MPC7447A	CABA MPC7447A	TLM MPC7447A	TLM-DT MPC7447A
MicroBlaze	CABA MicroBlaze	TLM MicroBlaze	TLM-DT MicroBlaze
Nios2	CABA Nios2	TLM Nios2	TLM-DT Nios2
LM32	CABA LM32	TLM LM32	TLM-DT LM32

	MMU cache controller		
	CABA	TLM	TLM-DT
Mips32	CABA Mips32	TLM Mips32	TLM-DT Mips32
PPC405	CABA PPC405	TLM PPC405	TLM-DT PPC405
SparcV8	CABA SparcV8	TLM SparcV8	TLM-DT SparcV8
ARM-v6k	CABA ARM-v6k	TLM ARM-v6k	TLM-DT ARM-v6k
ARM7tdmi	CABA ARM7tdmi	TLM ARM7tdmi	TLM-DT ARM7tdmi
ARM966	CABA ARM966	TLM ARM966	TLM-DT ARM966
TMS320	CABA TMS320	TLM TMS320	TLM-DT TMS320
MPC7447A	CABA MPC7447A	TLM MPC7447A	TLM-DT MPC7447A
MicroBlaze	CABA MicroBlaze	TLM MicroBlaze	TLM-DT MicroBlaze
Nios2	CABA Nios2	TLM Nios2	TLM-DT Nios2
LM32	CABA LM32	TLM LM32	TLM-DT LM32

	Coherent MMU cache ctrl.		
	CABA	TLM	TLM-DT
Mips32	CABA Mips32	TLM Mips32	TLM-DT Mips32
PPC405	CABA PPC405	TLM PPC405	TLM-DT PPC405
SparcV8	CABA SparcV8	TLM SparcV8	TLM-DT SparcV8
ARM-v6k	CABA ARM-v6k	TLM ARM-v6k	TLM-DT ARM-v6k
ARM7tdmi	CABA ARM7tdmi	TLM ARM7tdmi	TLM-DT ARM7tdmi
ARM966	CABA ARM966	TLM ARM966	TLM-DT ARM966
TMS320	CABA TMS320	TLM TMS320	TLM-DT TMS320
MPC7447A	CABA MPC7447A	TLM MPC7447A	TLM-DT MPC7447A
MicroBlaze	CABA MicroBlaze	TLM MicroBlaze	TLM-DT MicroBlaze
Nios2	CABA Nios2	TLM Nios2	TLM-DT Nios2
LM32	CABA LM32	TLM LM32	TLM-DT LM32

$\longrightarrow$ 99 models !

Problem: processor models

We need to model processors, factoring things:

	Simple cache controller			MMU cache controller			Coherent MMU cache ctrl.		
	CABA	TLM	TLM-DT	CABA	TLM	TLM-DT	CABA	TLM	TLM-DT
Mips32	CABA Mips32	TLM Mips32	TLM-DT Mips32	CABA Mips32	TLM Mips32	TLM-DT Mips32	CABA Mips32	TLM Mips32	TLM-DT Mips32
PPC405	CABA PPC405	TLM PPC405	TLM-DT PPC405	CABA PPC405	TLM PPC405	TLM-DT PPC405	CABA PPC405	TLM PPC405	TLM-DT PPC405
SparcV8	CABA SparcV8	TLM SparcV8	TLM-DT SparcV8	CABA SparcV8	TLM SparcV8	TLM-DT SparcV8	CABA SparcV8	TLM SparcV8	TLM-DT SparcV8
ARM-v6k	CABA ARM-v6k	TLM ARM-v6k	TLM-DT ARM-v6k	CABA ARM-v6k	TLM ARM-v6k	TLM-DT ARM-v6k	CABA ARM-v6k	TLM ARM-v6k	TLM-DT ARM-v6k
ARM7tdmi	CABA ARM7tdmi	TLM ARM7tdmi	TLM-DT ARM7tdmi	CABA ARM7tdmi	TLM ARM7tdmi	TLM-DT ARM7tdmi	CABA ARM7tdmi	TLM ARM7tdmi	TLM-DT ARM7tdmi
ARM966	CABA ARM966	TLM ARM966	TLM-DT ARM966	CABA ARM966	TLM ARM966	TLM-DT ARM966	CABA ARM966	TLM ARM966	TLM-DT ARM966
TMS320	CABA TMS320	TLM TMS320	TLM-DT TMS320	CABA TMS320	TLM TMS320	TLM-DT TMS320	CABA TMS320	TLM TMS320	TLM-DT TMS320
MPC7447A	CABA MPC7447A	TLM MPC7447A	TLM-DT MPC7447A	CABA MPC7447A	TLM MPC7447A	TLM-DT MPC7447A	CABA MPC7447A	TLM MPC7447A	TLM-DT MPC7447A
MicroBlaze	CABA MicroBlaze	TLM MicroBlaze	TLM-DT MicroBlaze	CABA MicroBlaze	TLM MicroBlaze	TLM-DT MicroBlaze	CABA MicroBlaze	TLM MicroBlaze	TLM-DT MicroBlaze
Nios2	CABA Nios2	TLM Nios2	TLM-DT Nios2	CABA Nios2	TLM Nios2	TLM-DT Nios2	CABA Nios2	TLM Nios2	TLM-DT Nios2
LM32	CABA LM32	TLM LM32	TLM-DT LM32	CABA LM32	TLM LM32	TLM-DT LM32	CABA LM32	TLM LM32	TLM-DT LM32

$\longrightarrow$ 18 models !

Outline

The ISS API

Overview

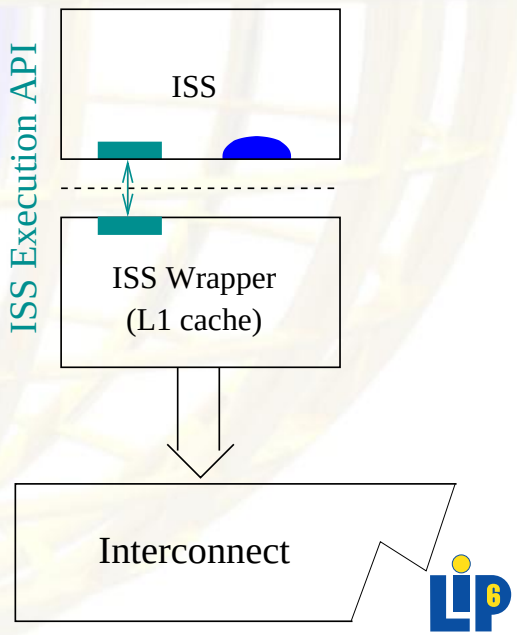
API is wrapper-driven.

The ISS API

Overview

API is wrapper-driven. The wrapper:

- is the L1-cache
- is dependent on the abstraction level
- connects to the external world (NoC, clock, ...)
- retrieves requests (instruction+data) from the ISS
- calls the ISS when needed so it can update its internal state



The ISS API

Description

- `getRequests(InstructionRequest &ireq, DataRequest &dreq)`
- `int executeNCycles(int cycles, const InstructionResponse &irsp, const DataResponse& drsp, uint32_t irq_bitfield)`

The ISS API

Description

- `getRequests(InstructionRequest &ireq, DataRequest &dreq)`
- `int executeNCycles(int cycles, const InstructionResponse &irsp, const DataResponse& drsp, uint32_t irq_bitfield)`

valid whether there is a request

mode processor mode at time of fetch (user/kernel)

address address of the instruction to fetch

The ISS API

Description

- `getRequests(InstructionRequest &ireq, DataRequest &dreq)`
- `int executeNCycles(int cycles, const InstructionResponse &irsp, const DataResponse& drsp, uint32_t irq_bitfield)`

valid whether there is a request

mode processor mode at time of fetch (user/kernel)

type type of the data access (read, write, ...)

address address of the data access

wdata written data (if needed)

be r/w data mask

The ISS API

Description

- `getRequests(InstructionRequest &ireq, DataRequest &dreq)`
- `int executeNCycles(int cycles, const InstructionResponse &irsp, const DataResponse& drsp, uint32_t irq_bitfield)`

cycles **maximum** number of cycles the iss may run

return value cycles the iss **actually** executed

irq_bitfield irq values during all the **cycles**

The ISS API

Description

- `getRequests(InstructionRequest &ireq, DataRequest &dreq)`
- `int executeNCycles(int cycles, const InstructionResponse &irsp, const DataResponse& drsp, uint32_t irq_bitfield)`

valid whether there is a response

error whether fetched raised an error

instruction memory word at fetched address

The ISS API

Description

- `getRequests(InstructionRequest &ireq, DataRequest &dreq)`
- `int executeNCycles(int cycles, const InstructionResponse &irsp, const DataResponse& drsp, uint32_t irq_bitfield)`

valid whether there is a response

error whether action raised an error

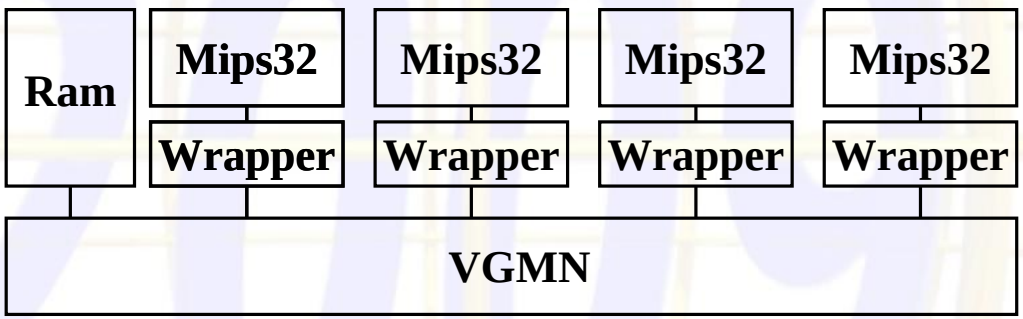
rdata memory word at fetched address (if needed)

Outline

Experimental results

The application

- The hardware platform:



- The software application is a multithreaded Smith-Waterman genome sequencing application (1 thread per processor).

Experimental results

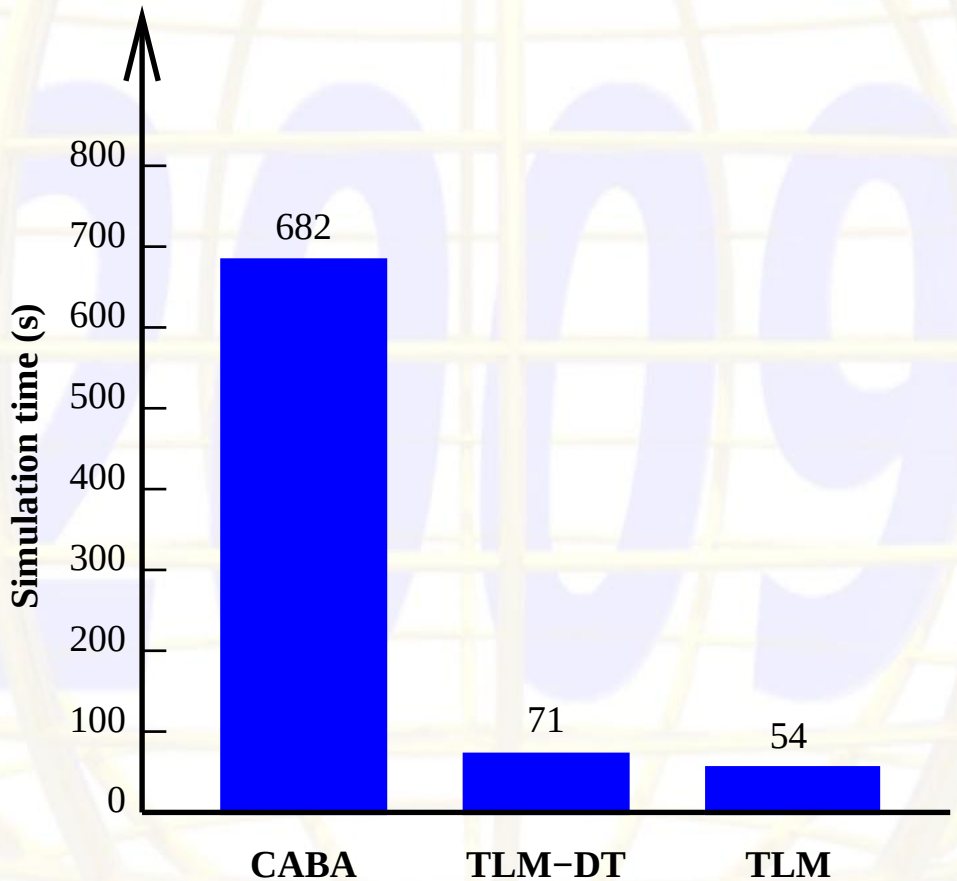
The application

Let's

- run this application for a fixed dataset on 1, 2, 3 and 4 processors.
- run this application on the same hardware architecture at various abstraction levels CABA, TLM-DT and TLM.
- compare the average simulation time for various abstraction level.

Experimental results

Chart

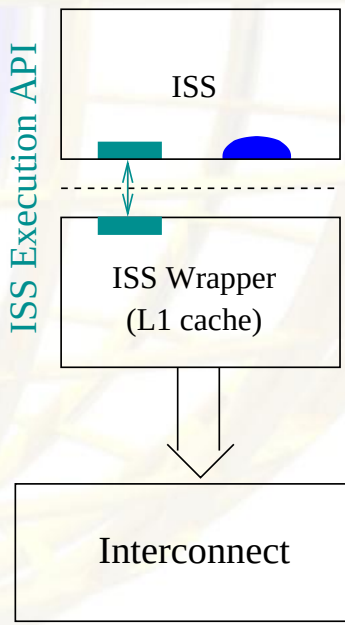


Outline

Instrumentation

Principles

Instrumentation modules are inserted between the wrapper and the ISS.

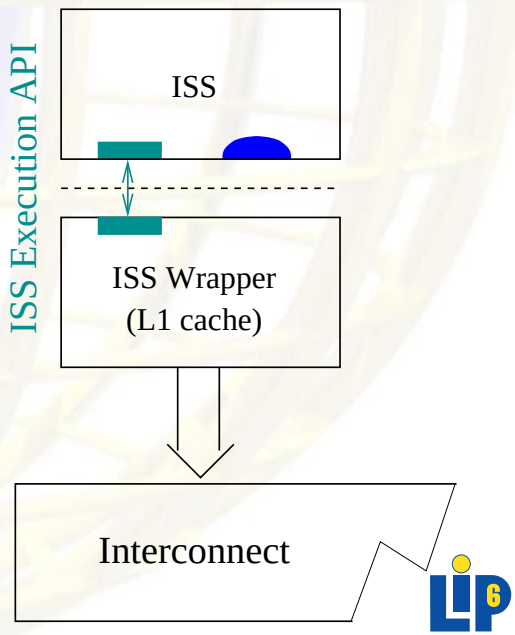


Instrumentation

Principles

Instrumentation modules are inserted between the wrapper and the ISS.

- they act as an ISS from the wrapper

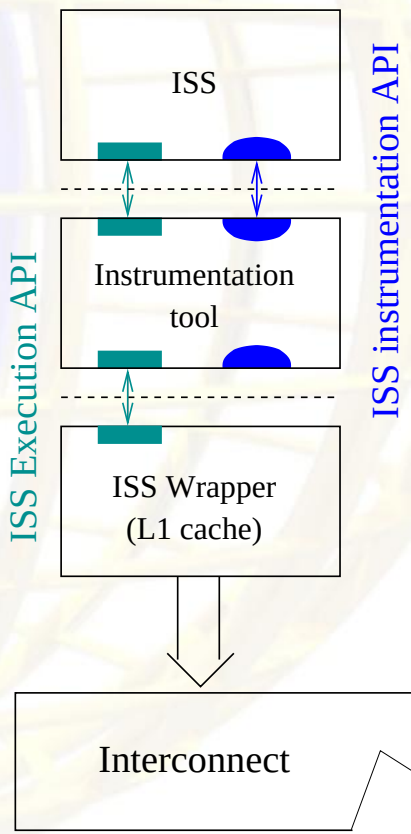


Instrumentation

Principles

Instrumentation modules are inserted between the wrapper and the ISS.

- they act as an ISS from the wrapper
- they act as a wrapper to the ISS

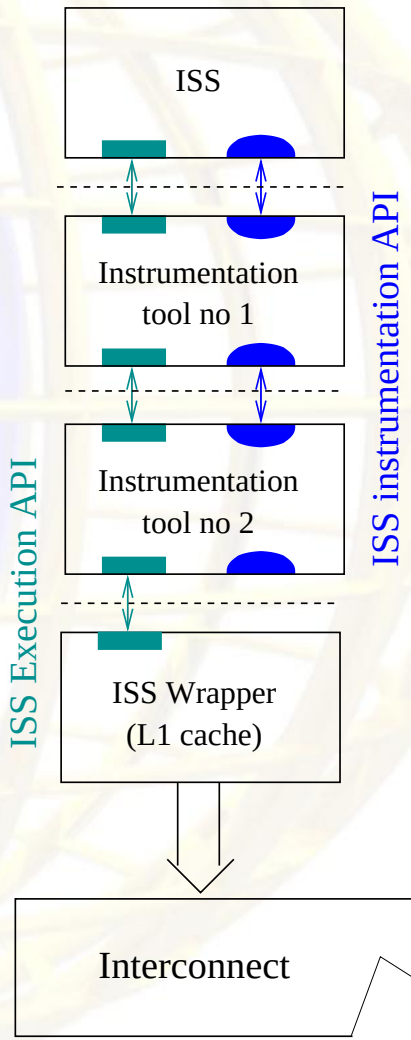


Instrumentation

Principles

Instrumentation modules are inserted between the wrapper and the ISS.

- they act as an ISS from the wrapper
- they act as a wrapper to the ISS
- they can be stacked!

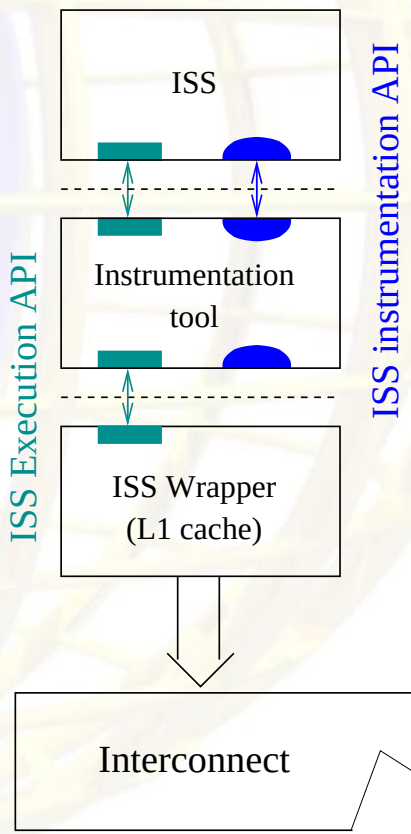


Instrumentation

Principles

Instrumentation modules are inserted between the wrapper and the ISS.

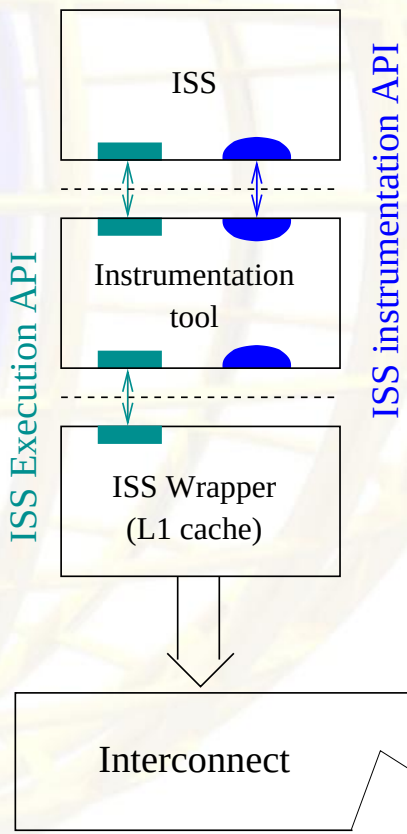
- they act as an ISS from the wrapper
- they act as a wrapper to the ISS
- they can be stacked!



Instrumentation

Features

Instrumentation modules

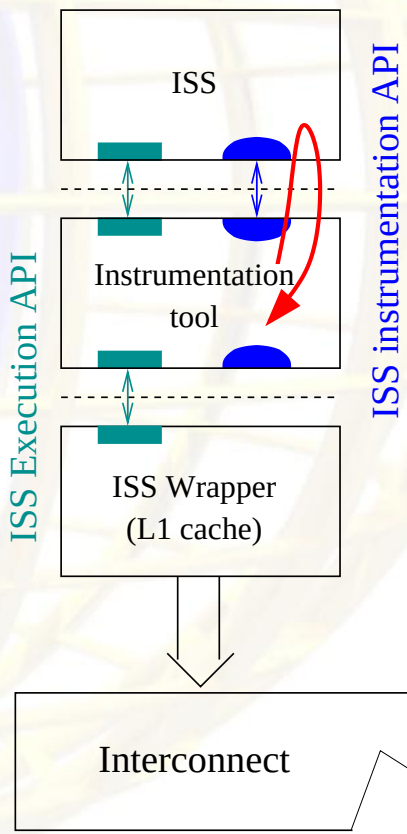


Instrumentation

Features

Instrumentation modules

- can read the ISS internal state

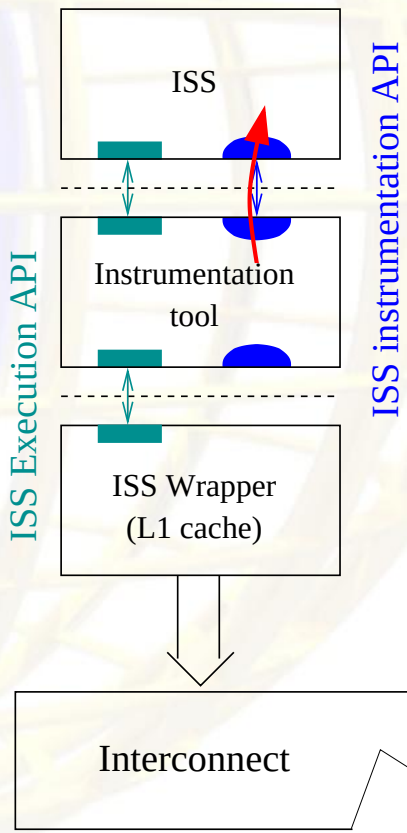


Instrumentation

Features

Instrumentation modules

- can read the ISS internal state
- can modify the ISS internal state

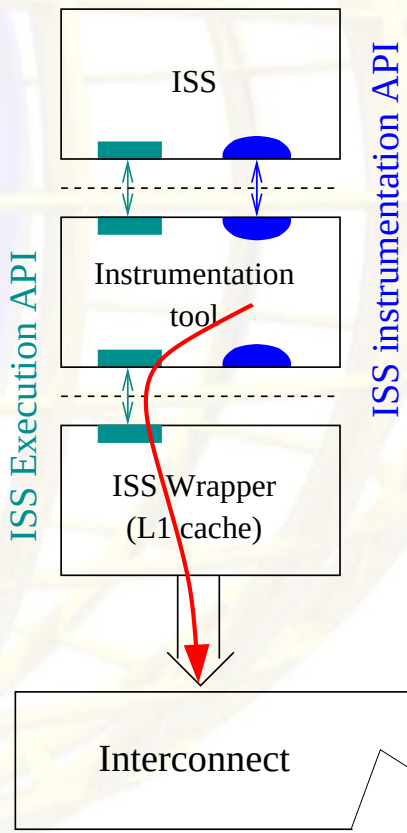


Instrumentation

Features

Instrumentation modules

- can read the ISS internal state
- can modify the ISS internal state
- can directly access the system memory from the ISS's point of view

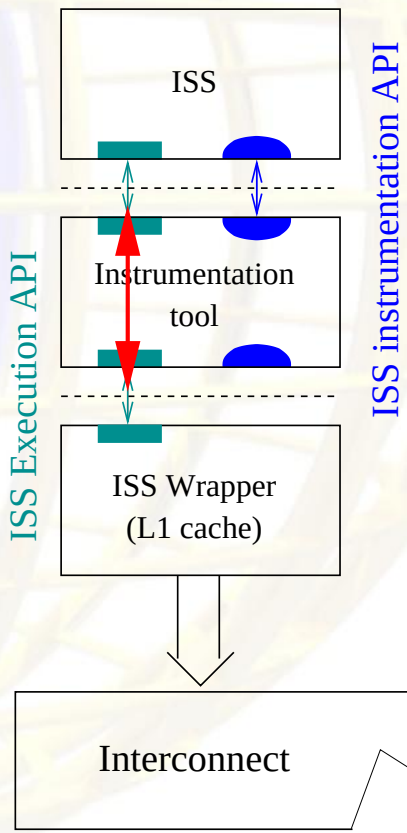


Instrumentation

Features

Instrumentation modules

- can read the ISS internal state
- can modify the ISS internal state
- can directly access the system memory from the ISS's point of view
- should forward functionnal communication between ISS and the wrapper

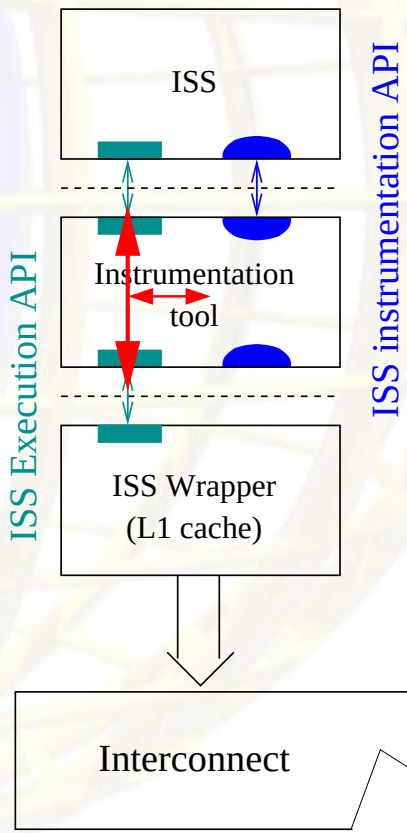


Instrumentation

Features

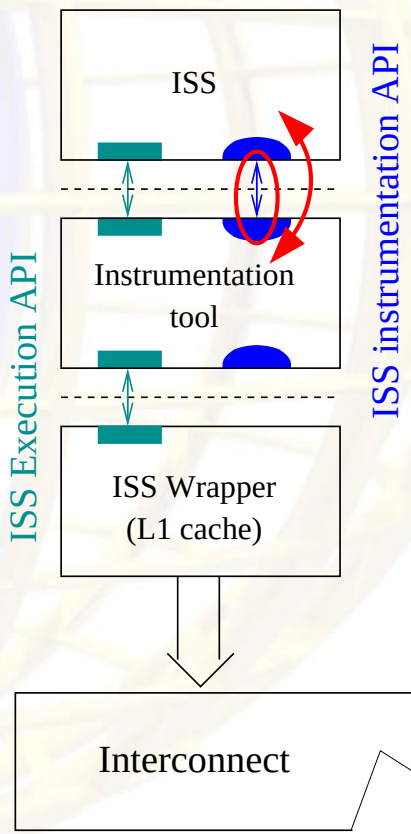
Instrumentation modules

- can read the ISS internal state
- can modify the ISS internal state
- can directly access the system memory from the ISS's point of view
- should forward functionnal communication between ISS and the wrapper
- can be intrusive or not depending on their functionality.



Instrumentation

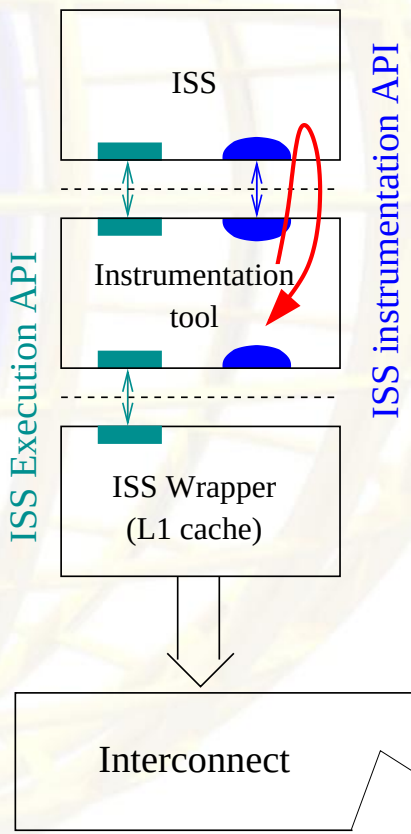
API



Instrumentation

API

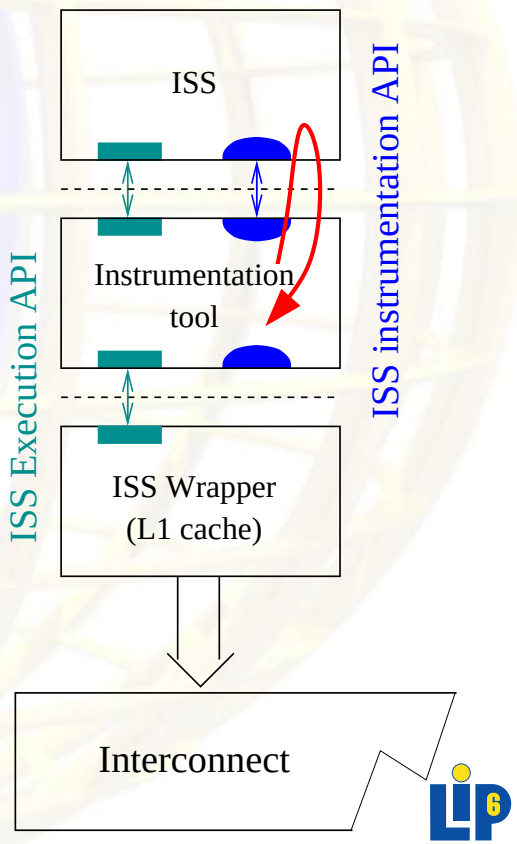
- `int debugGetRegisterCount()`



Instrumentation

API

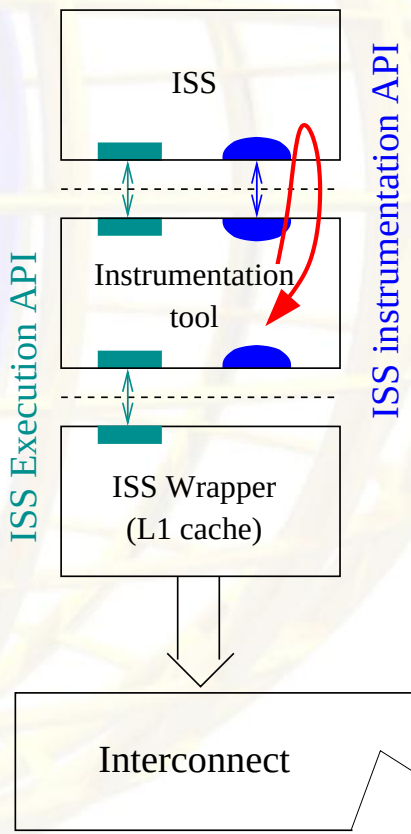
- `int debugGetRegisterCount()`
- `reg_t debugGetRegisterValue(int no)`



Instrumentation

API

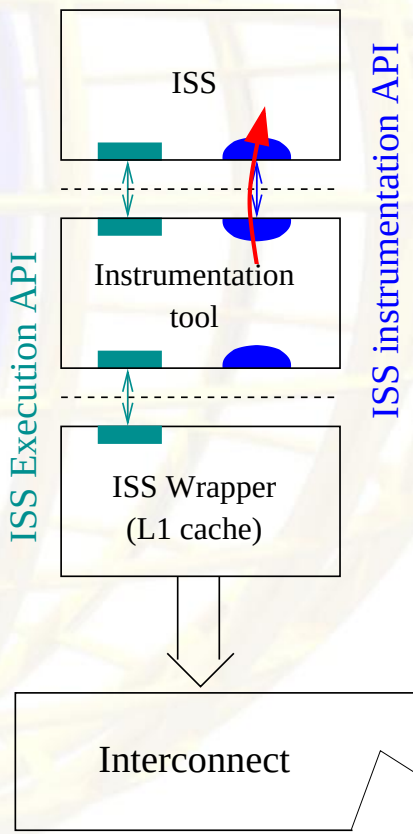
- `int debugGetRegisterCount()`
- `reg_t debugGetRegisterValue(int no)`
- `reg_t debugGetRegisterSize(int no)`



Instrumentation

API

- `int debugGetRegisterCount()`
- `reg_t debugGetRegisterValue(int no)`
- `reg_t debugGetRegisterSize(int no)`
- `void debugSetRegisterValue(int no, reg_t value)`



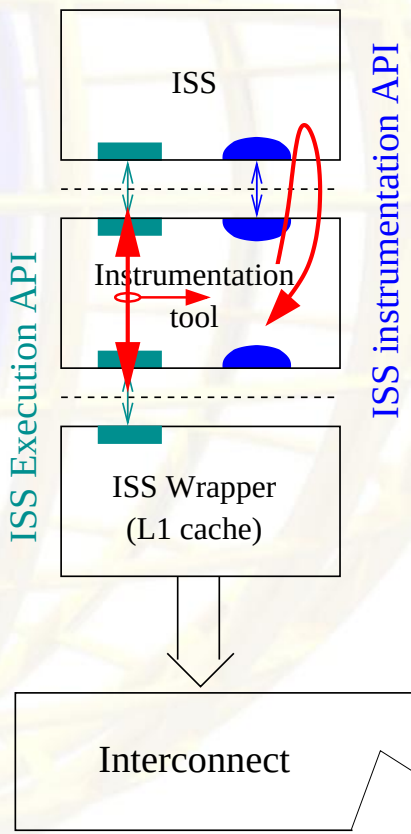
Instrumentation

API

- `int debugGetRegisterCount()`
- `reg_t debugGetRegisterValue(int no)`
- `reg_t debugGetRegisterSize(int no)`
- `void debugSetRegisterValue(int no, reg_t value)`

Profiler

Memory checker

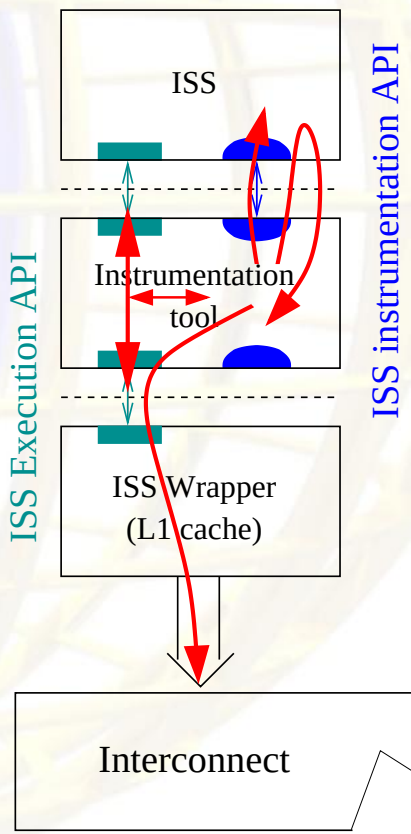


Instrumentation

API

- `int debugGetRegisterCount()`
- `reg_t debugGetRegisterValue(int no)`
- `reg_t debugGetRegisterSize(int no)`
- `void debugSetRegisterValue(int no, reg_t value)`

GDB



Outline

Conclusion

- The SoCLib platform supports virtual prototyping of MP2-SoCs.
- The generic ISS API
  - has been used by 5 different laboratories to develop 12 different processor cores
  - supports reliable performance evaluation using the actual embedded software application
  - can be used at different levels of abstraction (CABA, TLM-DT, TLM)
  - supports various (intrusive or non-intrusive) instrumentation tools: GDB, profiler, memory checker

SoCLib is available as free software[1]:

http://www.soclib.fr/

Questions ?

[1] LGPL-2