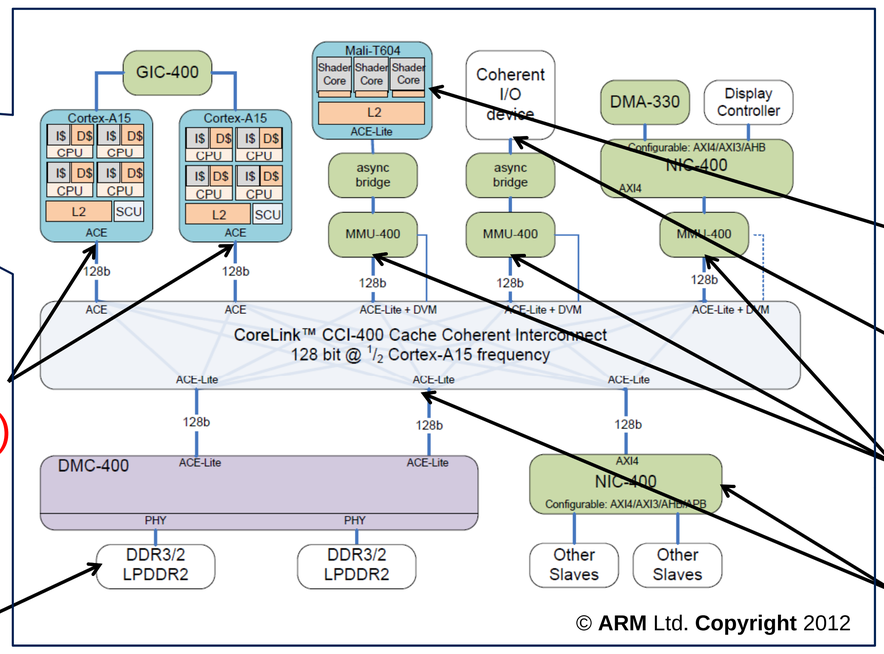
Formal Analysis of the ACE specification for Cache Coherent Systems-on-Chip (SoC)

Abderahman KRIOUILE (joint work with Wendelin SERWE)

System-Level Cache Coherency for Heterogeneous SoCs

Processors (with caches)



Memory

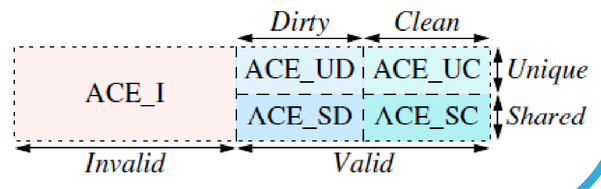
Components without caches:

- Accelerators
- I/O Blocks
- Dedicated IPs
- Interconnects

de facto standard for System Level Cache Coherency:

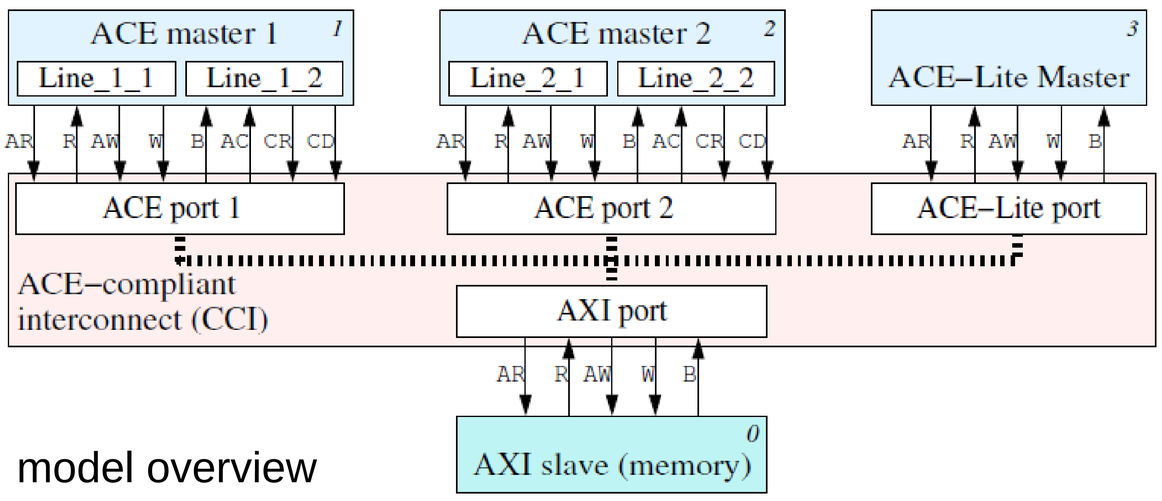
ACE protocol ARM

ACE states of a cache line



Formally Modeling an ACE-compliant SoC using CADP (http://cadp.inria.fr)

- Modular toolbox for asynchronous systems: formal modeling & verification, simulation
- Based on concurrency theory
- User-friendly input language



model overview

Experimental results: state space generation and verification

allowed transactions m1	allowed transactions m2	allowed transactions lite	global constraints	LTS size states	LTS size transitions	properties φ_1	φ_2	φ_3	φ_4	φ_5
S_0	$\{\mathcal{A}\}$	S_0	yes	93,481,270	308,087,560	√	√	√	√	√
S_0	$\{\mathcal{A}\}$	S_0	no	105,376,971	351,344,207	√	√	√	√	×
S_0	$\emptyset$	S_0	yes	7,518,552	21,227,610	√	√	√	√	√
S_1	$\emptyset$	S_1	yes	3,685,311	10,649,422	√	√	√	√	√
S_1	$\emptyset$	S_1	no	3,127,707	9,121,134	√	√	×	×	×
S_2	S_2	$\emptyset$	yes	3,545,801	11,122,536	√	√	√	√	√
S_2	S_2	$\emptyset$	no	2,819,505	9,095,620	√	√	×	×	√
S_3	$\emptyset$	S_3'	yes	1,834,195	5,170,829	√	√	√	√	√
S_3	$\emptyset$	S_3'	no	1,437,412	4,547,398	√	√	√	√	×
S_4	S_4	$\emptyset$	yes	560,299	1,669,886	√	√	√	√	√
S_4	S_4	$\emptyset$	no	599,971	1,780,634	√	√	×	×	×
S_5	S_5	$\emptyset$	yes	40,983	63,922	√	√	√	√	√
S_5	S_5	$\emptyset$	no	55,439	98,688	√	√	√	√	√

In the table above, we use those sets of allowed transactions:
S_0 = set of all ACE (respectively ACE-Lite) transactions
$S_1 = \{MakeUnique, ReadOnce, ReadUnique, WriteBack\}$
$S_2 = \{MakeInvalid, MakeUnique, ReadShared, ReadUnique, WriteBack\}$
$S_3 = \{MakeUnique, WriteBack\}$, $S_3' = \{ReadOnce\}$
$S_4 = \{CleanInvalid, CleanShared, ReadUnique, WriteBack\}$
$S_5 = \{MakeInvalid, MakeUnique, WriteBack\}$

Verified Properties in MCL (Model Checking Language)

Complete Execution of Transactions

```
[ true * . { AR ?op:String ?n:Nat ?l:Nat ... } ] inev ( { R !op !n !l } )
```

Cache Coherency

```
[ true * .
   {G ?m1:Nat ?indM:Nat "ACE_UD"} .
   ( not ({G !m1 !indM ?s1:String where s1<>"ACE_UD"}) ) * .
   {G ?m2:Nat !indM ?s2:String where (m2<>m1) and (s2<>"ACE_I")}
] false
```

Data Integrity

```
[ true * .
   { W !"WRITEBACK" ?c:Nat ?l:Nat ?d:Nat } .
   ( not { W !"WRITEBACK" !"0" !l !d !c } ) * .
   { W !"WRITEBACK" !"0" !l !d !c } .
   (
      ( not { AC ?any of String ?any of Nat !c ?any of Nat !l } ) and
      ( not { W ?any of String !"0" !l ?any of Nat ?any of Nat } )
   ) * .
   { W ?any of String !"0" !l ?h:Nat ?any of Nat where h<>d }
] false
```

Perspectives

- Generic interconnect model for analyzing impact of a coherent interconnect in a model of a concrete SoC
- Model-based test and validation: automatic test-scenario extraction and guided (co-)simulation

References

- ARM. *AMBA AXI and ACE Protocol Specification*, version ARM IHI0022E, http://infocenter.arm.com/help/topic/com.arm.doc.ihi0022e, Feb. 2013.
- H. Garavel, F. Lang, R. Mateescu, and W. Serwe. *CADP 2011: A toolbox for the construction and analysis of distributed processes*. Software Tools for Technology Transfer, 15(2):89–107, Apr. 2013.
- A. Kriouile and W. Serwe. *Formal Analysis of the ACE Specification for Cache Coherent Systems-on-Chip*. Proc. of Formal Methods for Industrial Critical Systems, LNCS 8187, pp. 108-122, Sept, 2013.