Trustworthy and Secure Semiconductors Workshop

Specifying the HW/SW interface

Peter Sewell

University of Cambridge

January 2013

~~Semiconductors~~

Trustworthy and Secure *Multiprocessors*

Specifying the HW/SW interface

Peter Sewell

University of Cambridge

January 2013

Trustworthy ~~and Secure~~ Multiprocessors

Specifying the HW/SW interface

Peter Sewell

University of Cambridge

January 2013

~~Trustworthy~~ Multiprocessors *for which we have a clue what they are supposed to do*

Specifying the HW/SW interface

Peter Sewell

University of Cambridge

January 2013

~~Trustworthy~~ Multiprocessors *for which we have a clue what they are supposed to do*

Processor Architectures Don't Really Exist (But They Should)

Peter Sewell

University of Cambridge

January 2013

What is a processor architecture anyway?

- interface between h/w and s/w development
- criterion for verification of processor design
- assumption for software verification

What is a processor architecture? Current Practice

Prose Books from Vendors:

- Intel 64 and IA-32 Architectures Software Developer's Manual
- AMD64 Architecture Programmer's Manual
- Power ISA specification
- ARM Architecture Reference Manual

Vendor-Internal Golden Simulation Models

Academic Formal Specifications of ISA Fragments

They have to be *loose* specifications

Sometimes easy to deal with — e.g., the IA-32 AAA instruction (ASCII Adjust After Addition):

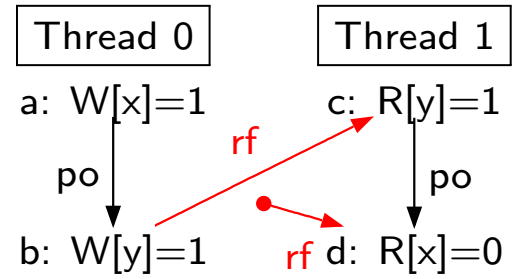
> *The AF and CF flags are set to 1 if the adjustment results in a decimal carry; otherwise they are set to 0. The OF, SF, ZF, and PF flags are undefined.*

Sometimes not so easy: multiprocessor concurrency

Multiprocessor Concurrency

Simple Message-Passing Example:

Thread 0	Thread 1
x=1	r1=y if this reads 1...
y=1	r2=x ...will this definitely read 1?
Initial state: x=0 y=0	

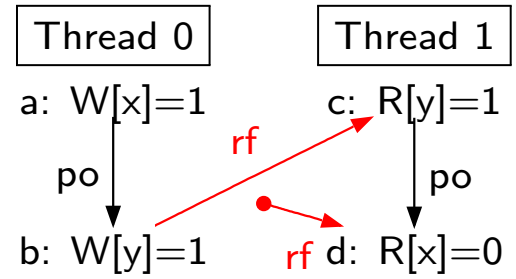


Test MP

Multiprocessor Concurrency

Simple Message-Passing Example:

Thread 0	Thread 1
x=1	r1=y if this reads 1...
y=1	r2=x ...will this definitely read 1?
Initial state: x=0 y=0	



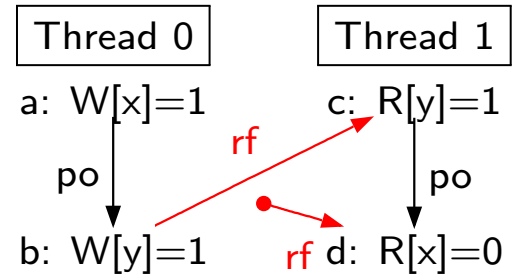
Test MP

		POWER			ARM			
	Kind	PowerG5	Power6	Power7	Tegra2	Tegra3	APQ8060	A5X
MP	Allow	10M/4.9G	6.5M/29G	1.7G/167G	40M/3.8G	138k/16M	61k/552M	437k/185M

Multiprocessor Concurrency

Simple Message-Passing Example:

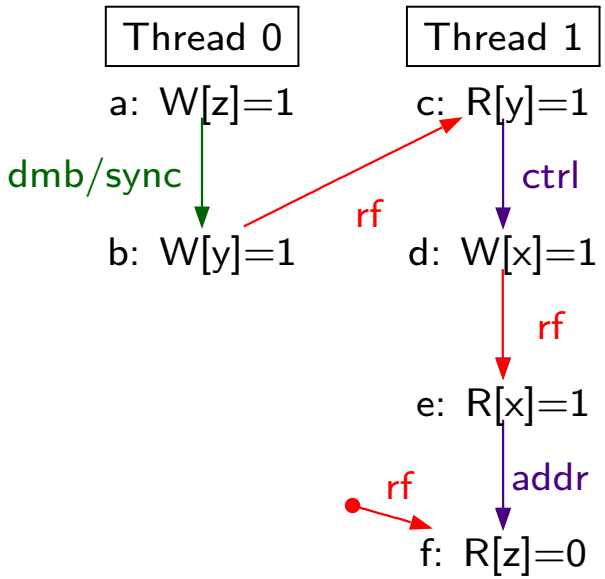
Thread 0	Thread 1
x=1	r1=y if this reads 1...
y=1	r2=x ...will this definitely read 1?
Initial state: x=0 y=0	



Test MP

Microarchitecturally: writes committed, writes propagated, and/or reads satisfied out-of-order

Less Simple Example



Test PPOCA: Allowed

		POWER			ARM			
	Kind	PowerG5	Power6	Power7	Tegra2	Tegra3	APQ8060	A5X
PPOCA	Allow	1.1k/3.4G	0/49G	175k/157G	0/24G	0/39G	233/743M	0/2.2G
PPOAA	Forbid	0/3.4G	0/46G	0/209G	0/24G	0/39G	0/26G	0/2.2G

What do the vendor architecture specs say?

“all that horrible horribly incomprehensible and confusing [...] text that no-one can parse or reason with — not even the people who wrote it”

Anonymous Processor Architect, 2011

“we’re terrible at specifying what we want”

S. Trimberger (Xilinx), 2013

Are the architectures serving their purpose?

- interface between h/w and s/w development?
  no: neither side understands them
- criterion for verification of processor design?
  no: they can’t be used to test or verify h/w against
- assumption for software verification?
  no: they can’t be used to test or verify s/w against

Are the architectures serving their purpose?

- interface between h/w and s/w development?
  no: neither side understands them
- criterion for verification of processor design?
  no: they can't be used to test or verify h/w against
- assumption for software verification?
  no: they can't be used to test or verify s/w against

In this sense, multiprocessor architectures **don't really exist**

Our Work (since 2007)

- clarify concurrency model for x86, IBM POWER, ARM (Sarkar et al.)
- clarify concurrency model for C11/C++11 (Batty et al.)

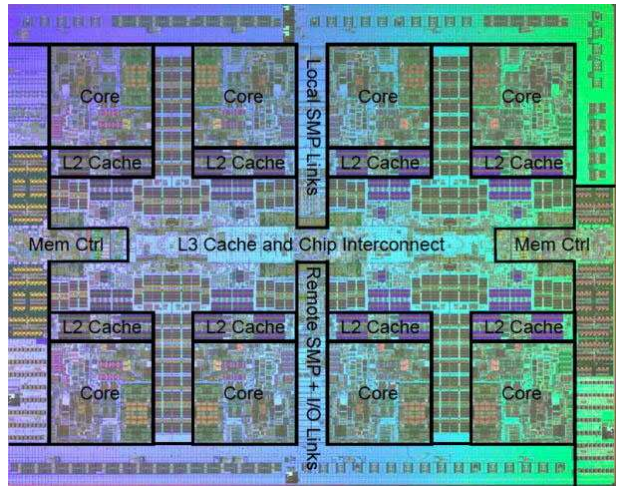
Industry Impact:

- x86 consensus spec
- in-depth discussion with IBM and ARM architects
- processor bugs found
- C/C++ standards committee
- compilation of C/C++11 concurrency to x86, Power, ARM
- compiler concurrency testing (Zappa Nardelli)

Using those models for s/w verification: CompCertTSO, C/C++11, take-up by others

Implementation vs Model

Programmer-observable relaxed-memory behaviour emerges from the whole system design: core speculative and OoO execution, cache protocol, SoC interconnect,...



1.2B, 45nm

but architectural model must be *abstract*: a usable programmers model, *sound w.r.t. implementations*, *loose enough* (not specific to any one implementation) but *strong enough*, *not confidential*, *precise*, *testable*, and *comprehensible to architects*

How?

1. test generation (manual and systematic)
2. test harness (pre-silicon and production - found many surprising phenomena plus some serious bugs)
3. write model in math (4000 lines)
4. generation of exhaustive simulator from model
5. auto-comparison between tests and model
6. English version of model, in sync with maths (few pages)
7. discussion with architects
8. `goto 1`

[Sarkar, Maranget, Alglave, Williams, Sewell]

Executable Specification

Must be able to:

- explore the model interactively
- compute the set of *all* model-allowed behaviours of small test programs
- reason about the model

QUICK PPCMEM DEMO

Big Success

clarified some very subtle things:

- for us
- for the hw vendors
- for the PL/compiler/sw-verification users

What's Missing?

Towards Real Architecture

that was just the basic concurrency model

To have an actual architecture spec, it has to be integrated with a full-scale ISA model, virtual memory, exceptions and interrupts, etc. In a well-tested and reusable form.

working on some of that now...

Tools

Lem: a lightweight *language for executable mathematics* (Owens et al.) for writing these models

tool compiles to executable code (OCaml), proof assistant definitions (Coq, HOL4, Isabelle/HOL), and readable LaTeX

used for x86, Power and ARM memory models, ISA fragments, C/C++ concurrency model

extensions to give an ISA description language that compiles via Lem to all those targets (c.f. Fox's L3)

testing tools

Discussion with Vendors

ARM

IBM

Intel, AMD, Centaur

Discussion with Academic Verification Groups

Academic fragments by many groups (us, Fox/Gordon, Leroy, Morrisett/Tan, Hunt, Benton/Kennedy, ...) for many purposes

Achievable Vision

Within the next few years, for each major processor family:

- a precise architecture model, shared between vendor and clients, covering (at least!) user-mode code
- used for verification/testing within vendor (giving confidence in soundness)
- used for verification and testing of critical software

Impact for Trustworthy and Secure Semiconductors?

Enabling

- apply same techniques to testing/correctness-proof of major subcomponents, e.g. interconnects or FPGA soft components – w.r.t. that architectural specification
- pre-silicon testing against real architectural model *(the same one as is exposed for s/w verification!)*, of whole system and subcomponents
- building architectural-envelope emulators from the specification, so can test s/w against that not just against particular impls
- verification of real systems s/w (isolation kernels)