Goals for a modern ISA specification

Alastair Reid

Saturday 17 June, 2023
PLARCH
Orlando, Florida, USA

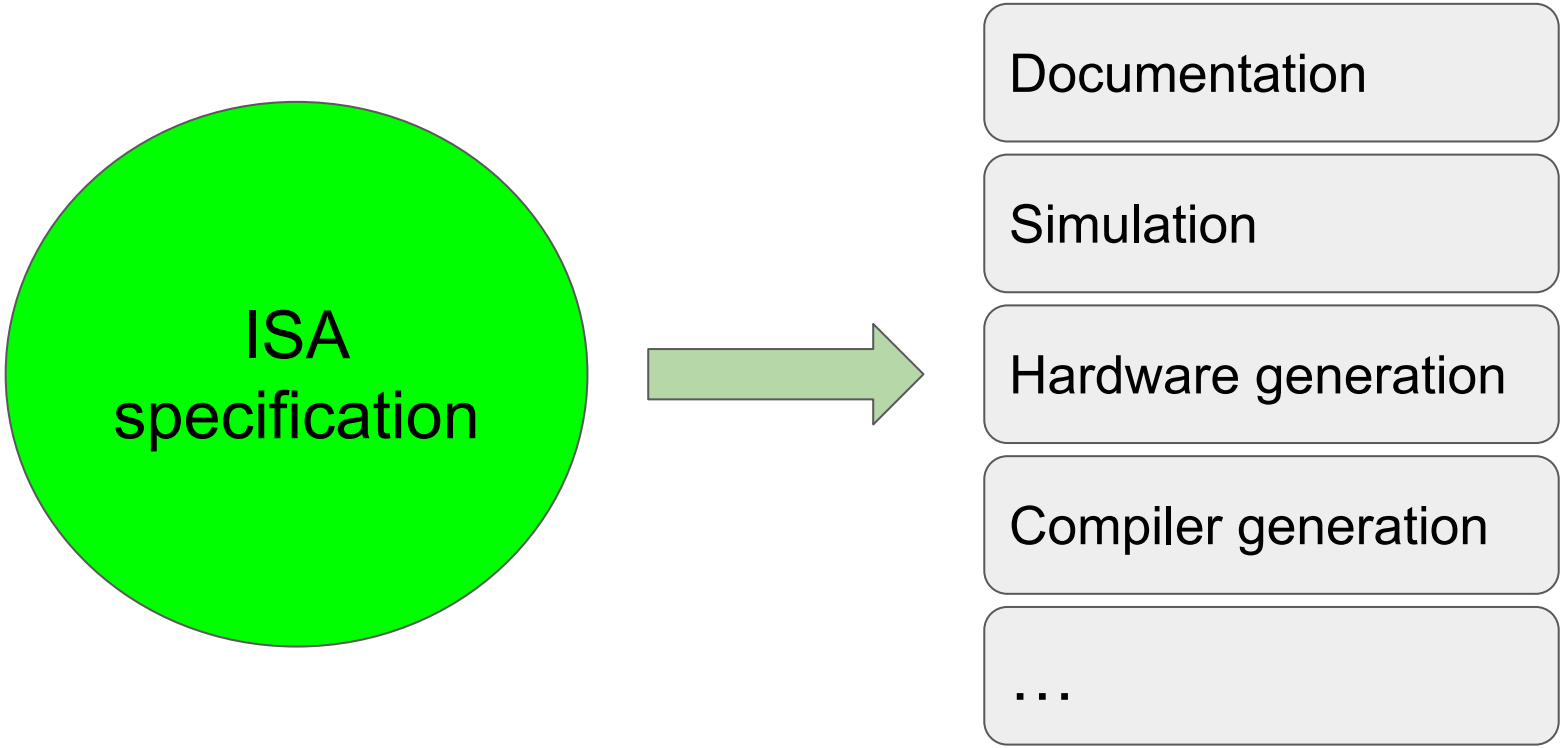
This is a personal talk - I am not representing my employer today.

The ISA spec I am currently working on may not satisfy all the goals.

The goals conflict with each other and we are still working on how to resolve the conflicts.

Please talk to me about which uses / goals matter most to you.

The key property of a “modern ISA specification”



Why using specifications multiple ways matters

Guaranteed consistency between uses

Virtuous cycle

Amortize cost across multiple uses

- Either: lower cost or enable higher investment of effort

Specs that enable many uses are more flexible

ISP: The first “modern ISA specification” (that I know of)

Bell & Newell, The PMS and ISP descriptive systems for computer structures, AFIPS, **1970**

Bell & Newell, Computer structures: Readings and examples, McGraw-Hill, **1971**

Barbacci, Bell, Siewiorek, ISP: A notation to describe a computer's instruction sets, IEEE Computer 6(3), **1973**

DEC Inc, PDP-11/45 processor handbook, **1973**

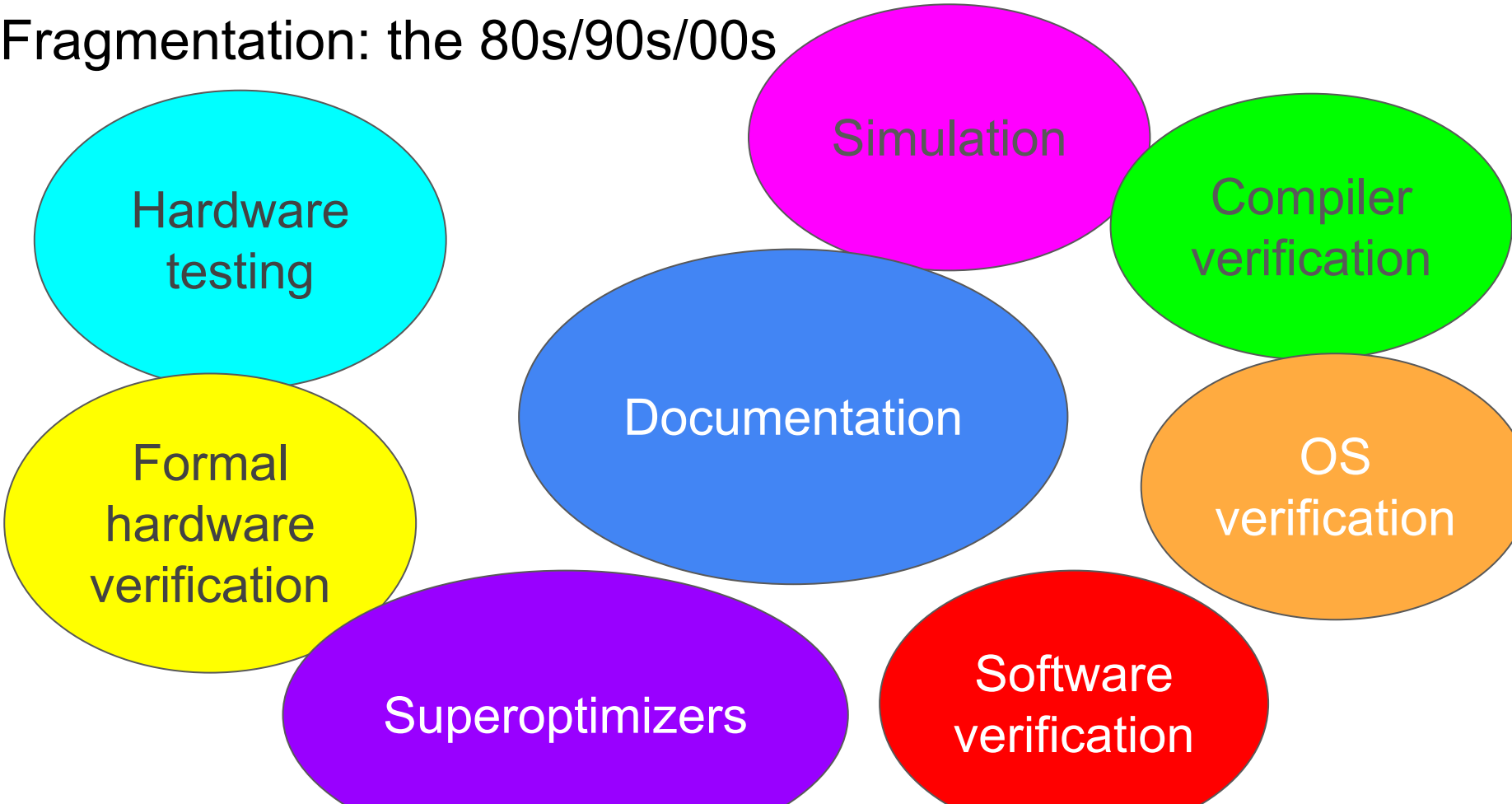
Fraser, A knowledge-based code generator generator, **1977**

…

Barbacci, Instruction set processor specifications (ISPS): The notation and its applications, IEEE TCS, **1981**

See: https://alastairreid.github.io/RelatedWork/notes/isps/

Fragmentation: the 80s/90s/00s



A revival in the last decade

Arm v8-A and Arm v8-M

- Reid, CAV 2016, “Trustworthy Specifications of ARM v8-A and v8-M System Level Architecture”

RISC-V

- Armstrong et al., POPL 2019, “ISA Semantics for ARMv8-A, RISC-V, and CHERI-MIPS”

Planning a multi-use ISA specification

What are all the uses that we care about?

What are the needs of each use case?

What are the conflicts between these different uses?

How close can we get to meeting all the needs?

	Read-able	Fast	Mechan-ized	Correct	Over approx	100% insn	Virt Mem	100% CSRs	100% Except-ions	Weak memory	Side channels
Docs	Y				Y	Y	Y	Y	Y		
HW test			Y			Y	Y	Y	Y		
HW verify			Y	Y		Y	Y	Y	Y		
SW v'fy			Y	Y	Y						
Compiler			Y	Y	Y						
Superopt			Y	Y	Y						
OS			Y	Y	Y		Y	Y	Y		
Simulate		Y	Y	Y	Y				Y		
MP			Y		Y					Y	
Security					Y		Y		Y		Y

	Read-able	Fast	Mechan-ized	Correct	Over approx	100% insn	Virt Mem	100% CSRs	100% Except-ions	Weak memory	Side channels
Docs	Y				Y	Y	Y	Y	Y		
HW test			Y			Y	Y	Y	Y		
HW verify			Y	Y		Y	Y	Y	Y		
SW v'fy			Y	Y	Y						
Compiler			Y	Y	Y						
Superopt			Y	Y	Y						
OS			Y	Y	Y		Y	Y	Y		
Simulate		Y	Y	Y	Y				Y		
MP			Y		Y					Y	
Security					Y		Y		Y		Y

	Read -able	Fast	Mechan -ized	Correct	Over approx	100% insn	Virt Mem	100% CSRs	100% Except -ions	Weak memory	Side channels
Docs	Y				Y	Y	Y	Y	Y		
HW test			Y			Y	Y	Y	Y		
HW verify			Y	Y		Y	Y	Y	Y		
SW v'fy			Y	Y	Y						
Compiler			Y	Y	Y						
Superopt			Y	Y	Y						
OS			Y	Y	Y		Y	Y	Y		
Simulate		Y	Y	Y	Y				Y		
MP			Y		Y					Y	
Security					Y		Y		Y		Y

	Read -able	Fast	Mechan -ized	Correct	Over approx	100% insn	Virt Mem	100% CSRs	100% Except -ions	Weak memory	Side channels
Docs	Y				Y	Y	Y	Y	Y		
HW test			Y			Y	Y	Y	Y		
HW verify			Y	Y		Y	Y	Y	Y		
SW v'fy			Y	Y	Y						
Compiler			Y	Y	Y						
Superopt			Y	Y	Y						
OS			Y	Y	Y		Y	Y	Y		
Simulate		Y	Y	Y	Y				Y		
MP			Y		Y					Y	
Security					Y		Y		Y		Y

	Read-able	Fast	Mechan-ized	Correct	Over approx	100% insn	Virt Mem	100% CSRs	100% Except-ions	Weak memory	Side channels
Docs	Y				Y	Y	Y	Y	Y		
HW test			Y			Y	Y	Y	Y		
HW verify			Y	Y		Y	Y	Y	Y		
SW v'fy			Y	Y	Y						
Compiler			Y	Y	Y						
Superopt			Y	Y	Y						
OS			Y	Y	Y		Y	Y	Y		
Simulate		Y	Y	Y	Y				Y		
MP			Y		Y					Y	
Security					Y		Y		Y		Y

	Read -able	Fast	Mechan -ized	Correct	Over approx	100% insn	Virt Mem	100% CSRs	100% Except -ions	Weak memory	Side channels
Docs	Y				Y	Y	Y	Y	Y		
HW test			Y			Y	Y	Y	Y		
HW verify			Y	Y		Y	Y	Y	Y		
SW v'fy			Y	Y	Y						
Compiler			Y	Y	Y						
Superopt			Y	Y	Y						
OS			Y	Y	Y		Y	Y	Y		
Simulate		Y	Y	Y	Y				Y		
MP			Y		Y					Y	
Security					Y		Y		Y		Y

	Read -able	Fast	Mechan -ized	Correct	Over approx	100% insn	Virt Mem	100% CSRs	100% Except -ions	Weak memory	Side channels
Docs	Y				Y	Y	Y	Y	Y		
HW test			Y			Y	Y	Y	Y		
HW verify			Y	Y		Y	Y	Y	Y		
SW v'fy			Y	Y	Y						
Compiler			Y	Y	Y						
Superopt			Y	Y	Y						
OS			Y	Y	Y		Y	Y	Y		
Simulate		Y	Y	Y	Y				Y		
MP			Y		Y					Y	
Security					Y		Y		Y		Y

	Read -able	Fast	Mechan -ized	Correct	Over approx	100% insn	Virt Mem	100% CSRs	100% Except -ions	Weak memory	Side channels
Docs	Y				Y	Y	Y	Y	Y		
HW test			Y			Y	Y	Y	Y		
HW verify			Y	Y		Y	Y	Y	Y		
SW v'fy			Y	Y	Y						
Compiler			Y	Y	Y						
Superopt			Y	Y	Y						
OS			Y	Y	Y		Y	Y	Y		
Simulate		Y	Y	Y	Y				Y		
MP			Y		Y					Y	
Security					Y		Y		Y		Y

A compromise between all use cases

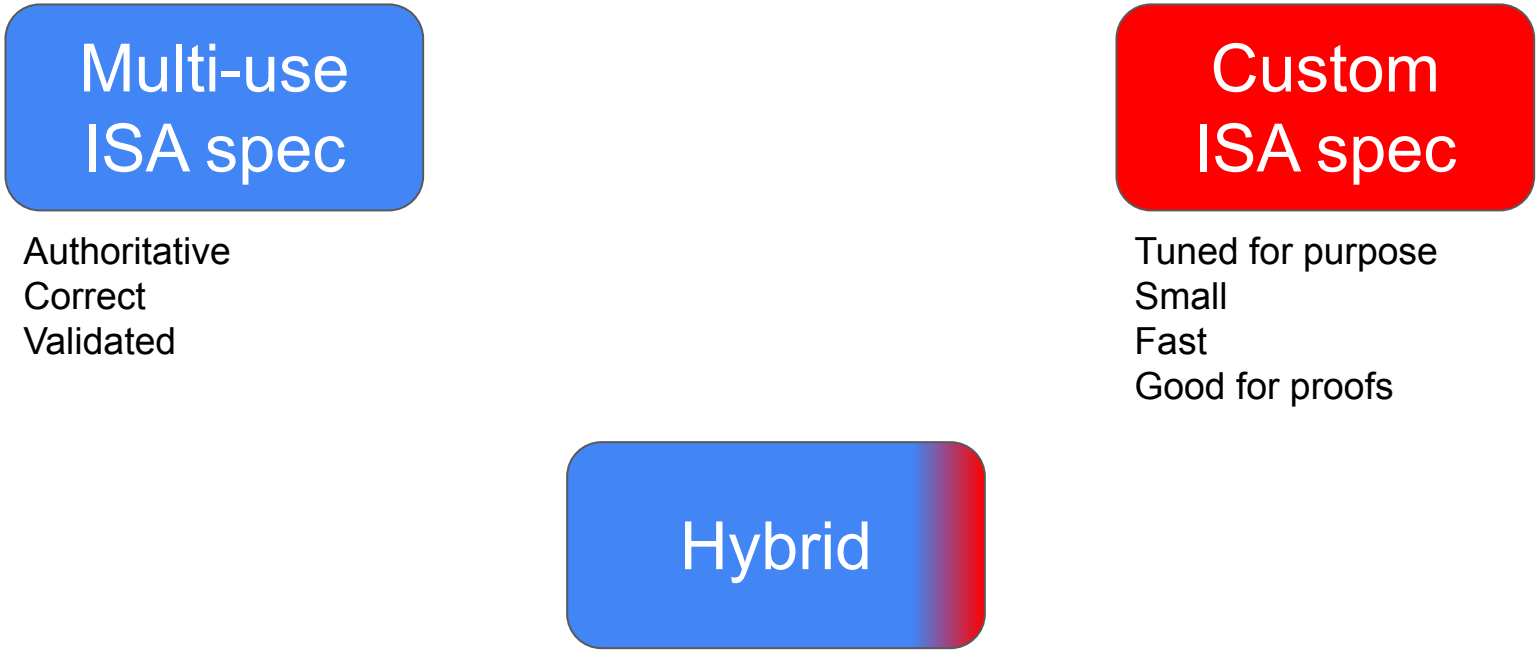
Multi-use ISA spec

Authoritative
Correct
Validated

Custom ISA spec

Tuned for purpose
Small
Fast
Good for proofs

A compromise between all use cases



Research challenges in creating a multi-use ISA spec

Engineering

100% insn, MSRs, exceptions, …? - “just” engineering / cost

Readable vs correct? - an ongoing discussion

Overapproximate or precisely matching a particular core? - the biggest daily issue

Weak memory - almost understand how to do it, hard to verify

uArch side channels - not yet clear how to specify in uArch-independent way

Research

Conclusions

We should strive to make specifications multi-use

Conflicts between different uses

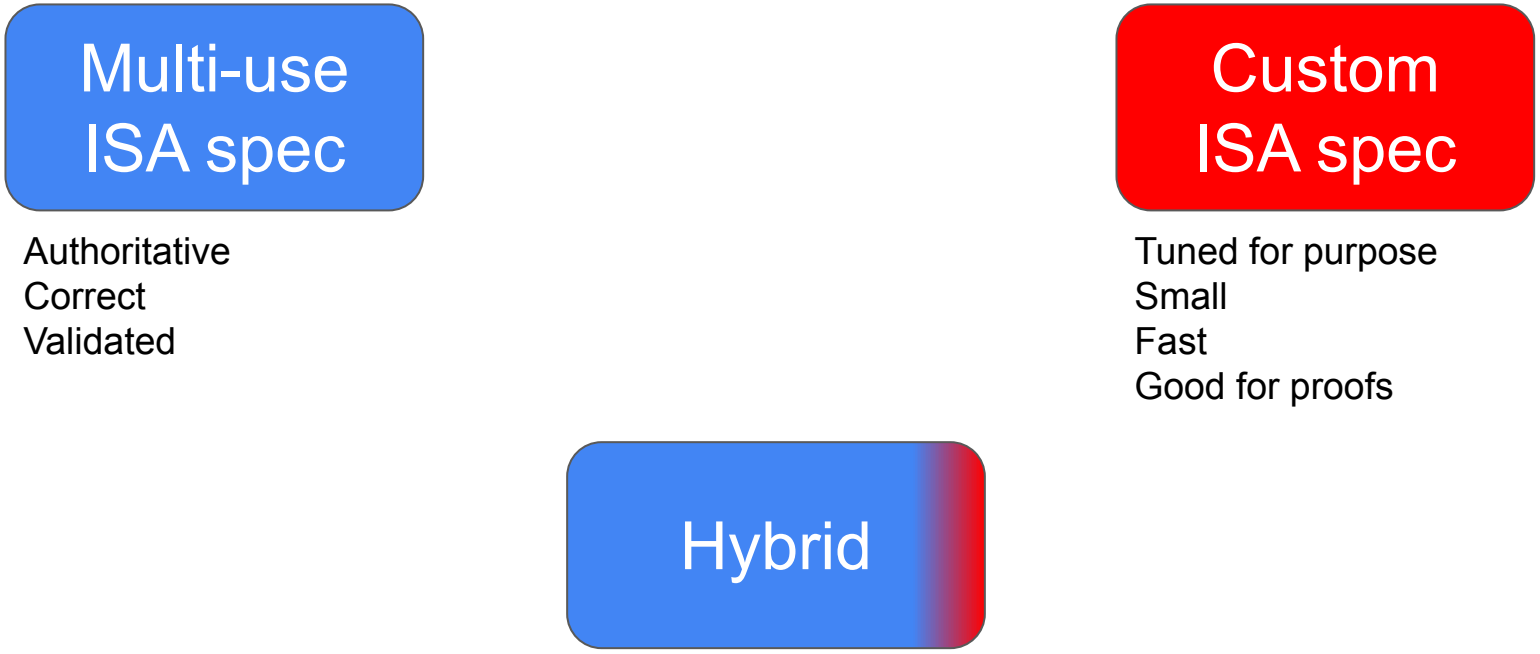
Still not clear if / how we can meet all needs in a single ISA spec

- Which is why it is fun to try

Questions?

The remaining slides are the originals from before when I had to remove all animation in order to produce a PDF

A compromise between all use cases



	Read-able	Fast	Mechan-ized	Correct	Over approx	100% insn	Virt Mem	100% CSRs	100% Except-ions	Weak memory	Side channels
Docs	Y				Y	Y	Y	Y	Y		
HW test			Y			Y	Y	Y	Y		
HW verify			Y	Y		Y	Y	Y	Y		
SW v'fy			Y	Y	Y						
Compiler			Y	Y	Y						
Superopt			Y	Y	Y						
OS			Y	Y	Y		Y	Y	Y		
Simulate		Y	Y	Y	Y				Y		
MP			Y		Y					Y	
Security					Y		Y		Y		Y