Stanford's AHA (Agile Hardware) Center

(A Personal Story)

Mark Horowitz

7/12/2018

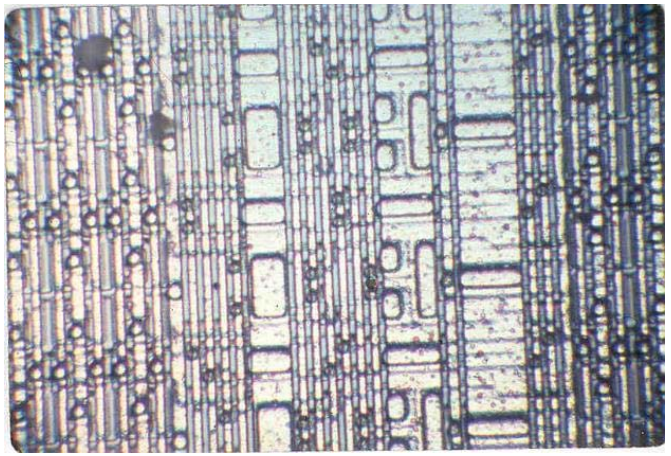
Stanford | ENGINEERING
Electrical Engineering

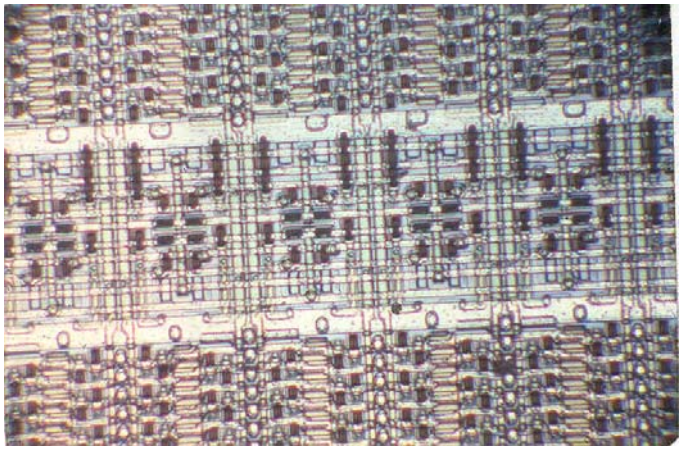
I’VE WORKED IN VLSI FOR A LONG TIME

1978 – Hello Silicon Valley

Hot new technologies

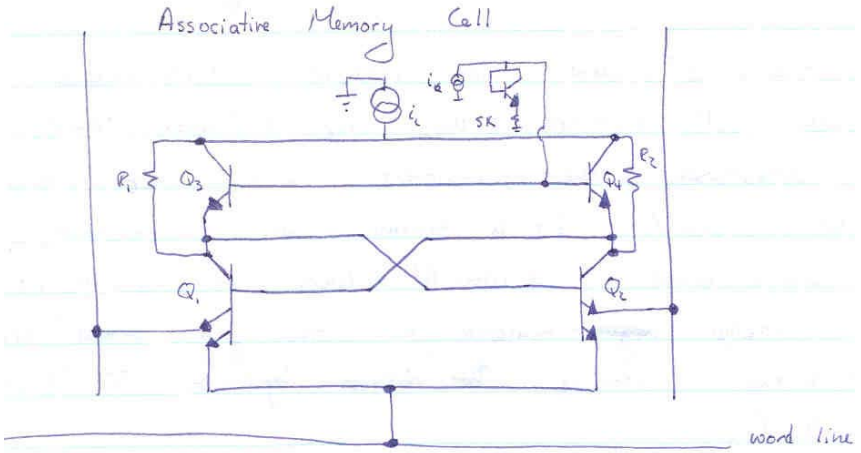
- 3μ nMOS
- Depletion loads and 5V operation, TTL I/O





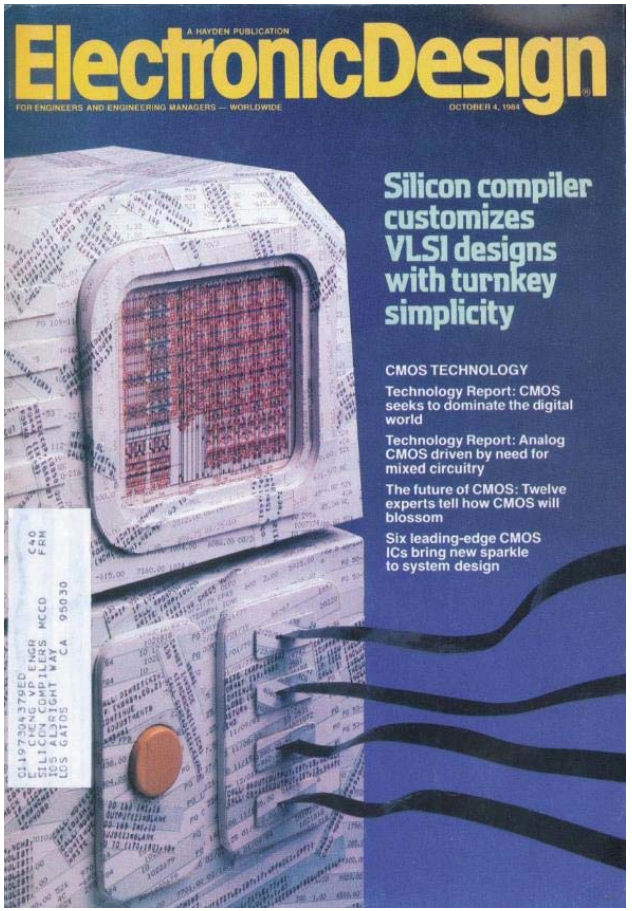
Worked at Signetics

- Worked on bipolar designs
  - 1 kbit ECL CAM, ISL gate array
- Design Flow
  - We did have circuit simulation
  - And I never used rubylith



Experienced the Innovative 1980s

- Schematic capture
  - Mentor, Daisy, Valid
- Academic design revolution
  - Mead and Conway
- Layout
  - Gate arrays
  - Stick diagrams
  - Silicon compilers



Optimal Solutions (a.k.a.) Synopsys

- Started as a logic optimization company
  - Netlist to better netlist
- When Verilog was a simulation language
- Place and route was what you did on boards

Chip Designers Were Skeptical of ASICs

- I should know, I was one of them
- They were ignoring tons of critical issues
  - Wires were important, set speed, routability
  - There were a lot of circuit “tricks” that they couldn’t do
- We were right and completely wrong

Why ASICs Won

No One Does Custom Design Today

- It allowed system designs to create the chips they wanted
  - Faster application performance using slower circuits
- For the same reason few people write assembly code today
  - And yes, libraries are still customized (for both)

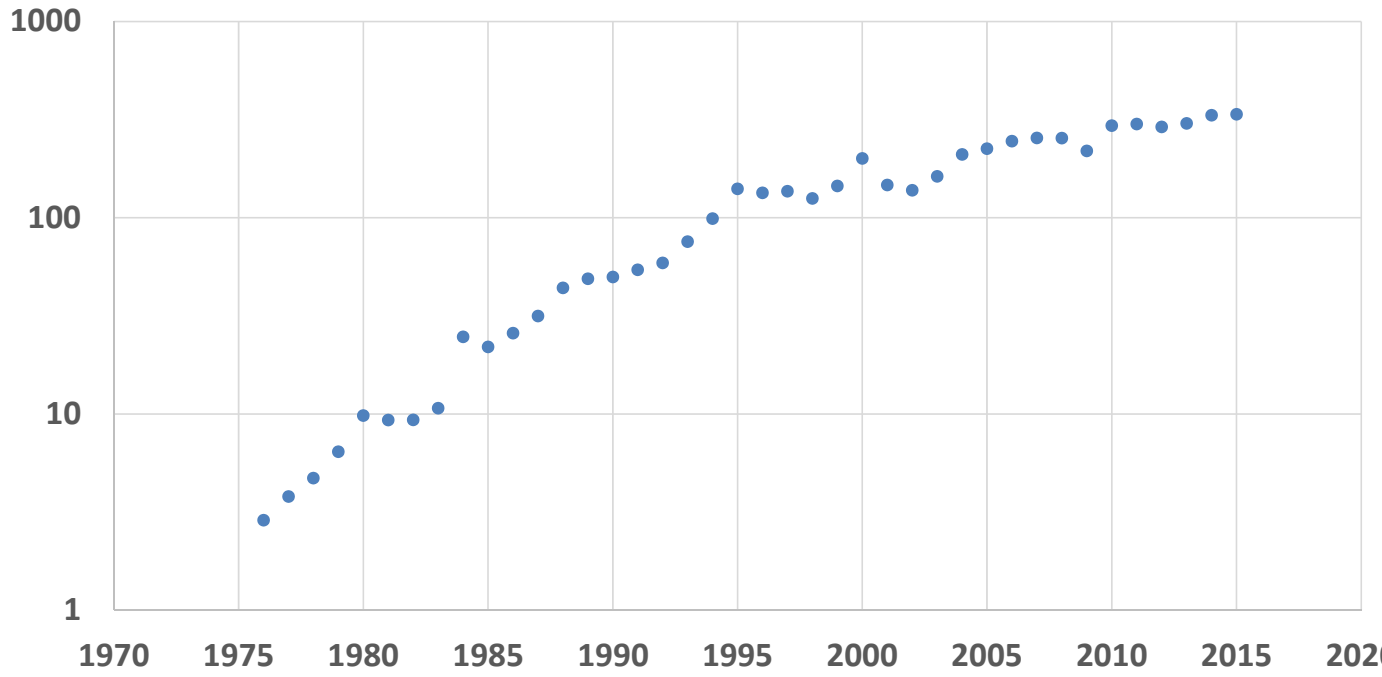
Why Synthesis, Place and Route Won

- It was a very clean abstraction layer
  - You abstracted away all layout issues
  - Allowed many people to start designing chips
- Leveraged the entire industry
  - New routers, and placers were constantly being created
  - No one company had all the good ideas
    - Or resources to do all the steps
- Silicon compilers bit off too much in one go
  - Had to create the libraries, and all the tools

30 Years Later

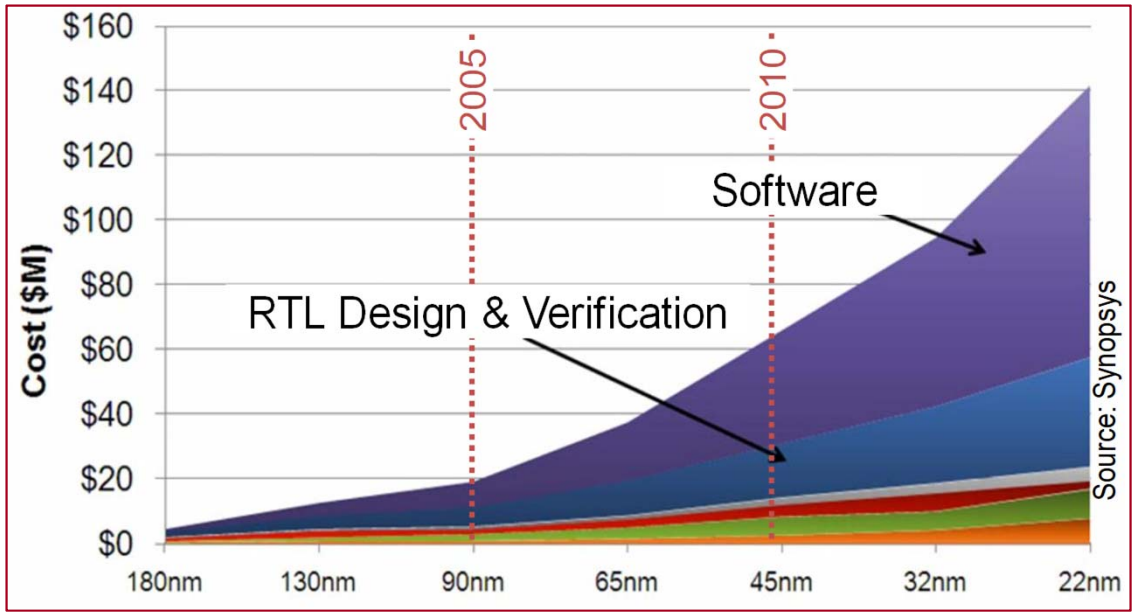
VLSI is Stalling

- Scaling is not what it used to be
  - Dennard scaling is dead, so power is everything
  - And scaling no longer has the economic gain it use to



VLSI is Stalling, cont'd

- Building complex chips costs lots of money
  - So fewer and fewer people can afford it
  - And when you build something expensive, the mantra is don't f*ck up.



It Was So Depressing …

- Started looking for new research areas …

But, It Was Clear To Me…

- To continue to create innovation silicon solutions
- We needed another design revolution
  - To decrease the cost to play
- Had an idea …

Cup Holders

- Small additions to a complex product
  - With large perceived value

The New Challenge:

- How to efficiently create these systems w/ cup holders
  - Which is **much** harder than producing the cup holder
- Porting an application to hardware
  - Is about reframing the algorithm as much as hardware design
  - Need expertise in the application area and hardware design

But …

Doing this would require a huge effort in:

- Creating the base system to add cup holders
- Tool flows to bring design to the next level

And VLSI tools research is even less popular than VLSI design …

My Saviors

Software Techniques

- Open source software
  - Large systems that people leverage for them specialized stuff
  - Use both open source tools to create stuff, and open source systems
- High level languages / Meta-programmed execution
  - Templates / Constructor
  - Don’t build a library
    - Build a tool that can optimize library for your application
- Agile Design
  - If you don’t know what is going to win, cycle quickly

Validation

- It is the hardest part of design
  - What you spend most of your time doing
- Built some of the best SMT solvers
  - Interested in using this technology in new ways
  - Symbolic QED
- Interested in creating new and better CAD tools, and tool flows

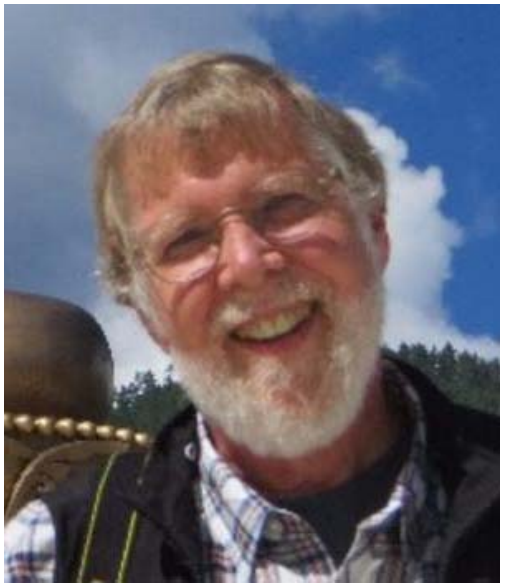
Intel ISTC on Agile Hardware

- Enable application designers to create cupholder systems
  - Explore custom accelerators for their application
- Requires creating agile (system &) hardware development tools
  - No one wants raw hardware
    - Need hardware and software to run on it
    - And software to seamlessly stitch it into the existing system
- Use image/video processing as driving app

Real Saviors

Aina Niemetz, Ankita Nayak, Artem Vasilyev, Caleb Donovick, Cristian Mattarei, Dillon Huff, Jeff Setter, Leonard Truong, Makai Michael Mann, Nate Chizgi, Nikhil Bhagdikar, Ross G Daly, Steven Bell, Steve Richardson, Xuan Yang, Alex Carsello,Taeyoung Kong, Keyi Zhang, David Durst, Qiaoyi Liu, James Thomas

Reinforcements



Vision

Change the world by:

- Showing that hardware/software system design
  - Doesn't need to cost huge amounts of money
  - Doesn't need to be done by silicon specialists
  - Doesn't need to take years to produce
- Our secret sauce:
  - Leveraging the lessons learned from software (again)
  - Realizing that we are going to reuse most of the system from last time

Make hardware/software system design fun again!

Specialized Hardware Mistake

- Study the application
- Build the hardware
- Then write the software for the machine
  - And that is the only application this hardware will ever support…

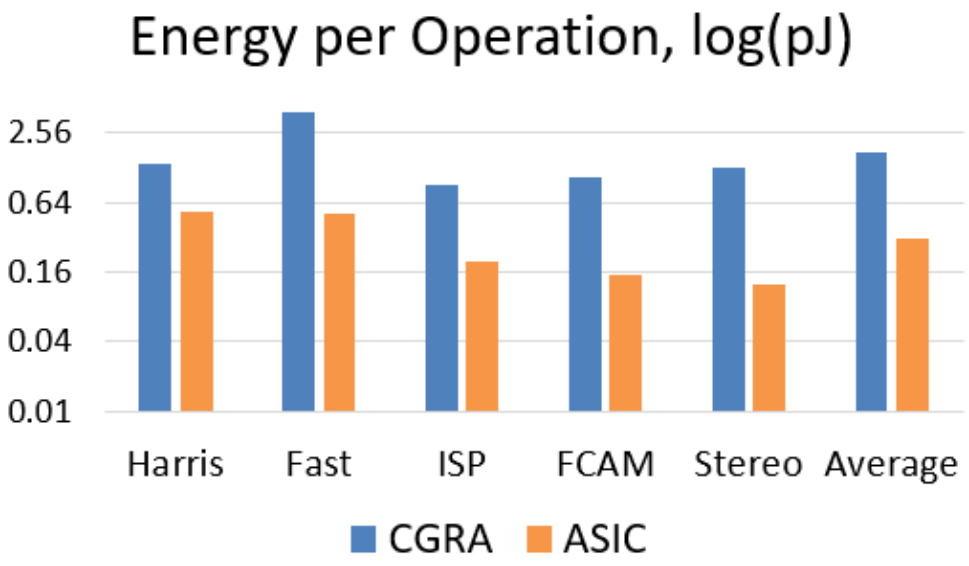
Need an “ISA”

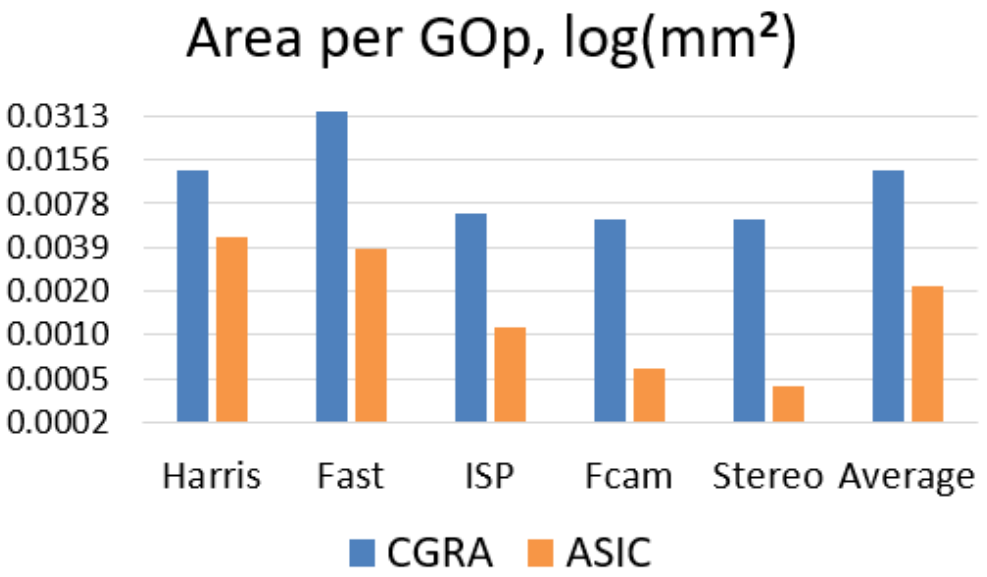
- Need a stable target for compiler and hardware

- This allows one to continue to optimize each side
  - And to run new applications on deployed hardware

But

- Cost of programmability is high



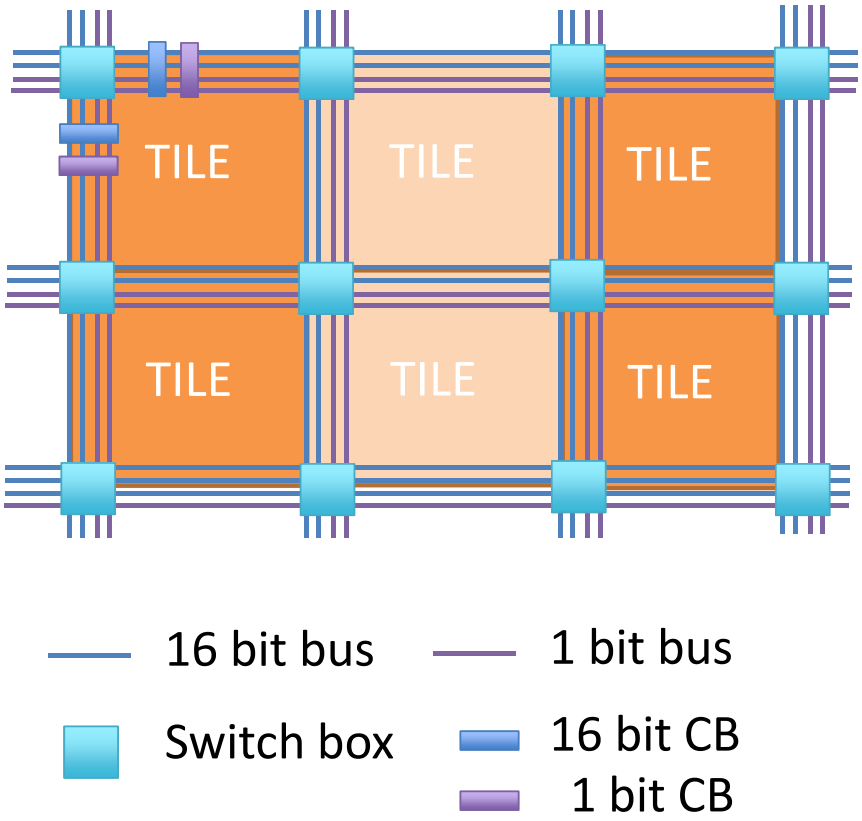


Optimizing w/ ISA

An Agile Approach to Accelerators

- Key idea:
  - Maintain end-to-end flow no matter what
  - Always be able to compile new applications
- Incrementally change tools/fabric to improve performance
- Accelerator becomes optimization of underlying fabric
  - Heterogeneous fabric, which may or may not need “new” instructions

CGRA is the Best Base Substrate

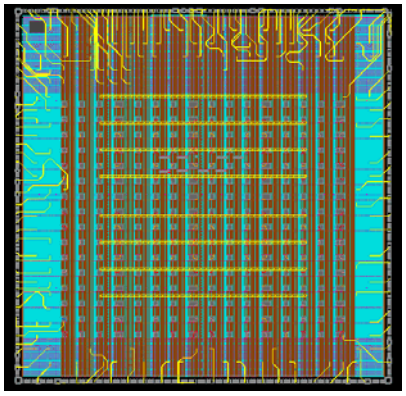


What We Are Going To Do

- Improved video analysis algorithms
- Improved Halide compiler
  - Includes an auto-scheduler
  - Includes a hardware back-end and driver/glue software generation
- Open source tool chain for compiling hardware
- Parameterized SoC generator
  - Including host cores (RISC-V), our CGRA, and other accelerators
- Formal methods for hardware validation and better SMT solvers

Our Design Philosophy

- Practice what we preach
  - Use agile practices in the design of our tools and hardware
    - Continuous integration
    - Rapid design cycles
  - Plan multiple tapeouts / year
- Eat our own dogfood
  - Use our tools to create our prototypes
  - Use our prototypes to create new systems
- Run a focused project
  - Already have taped out our first test chip



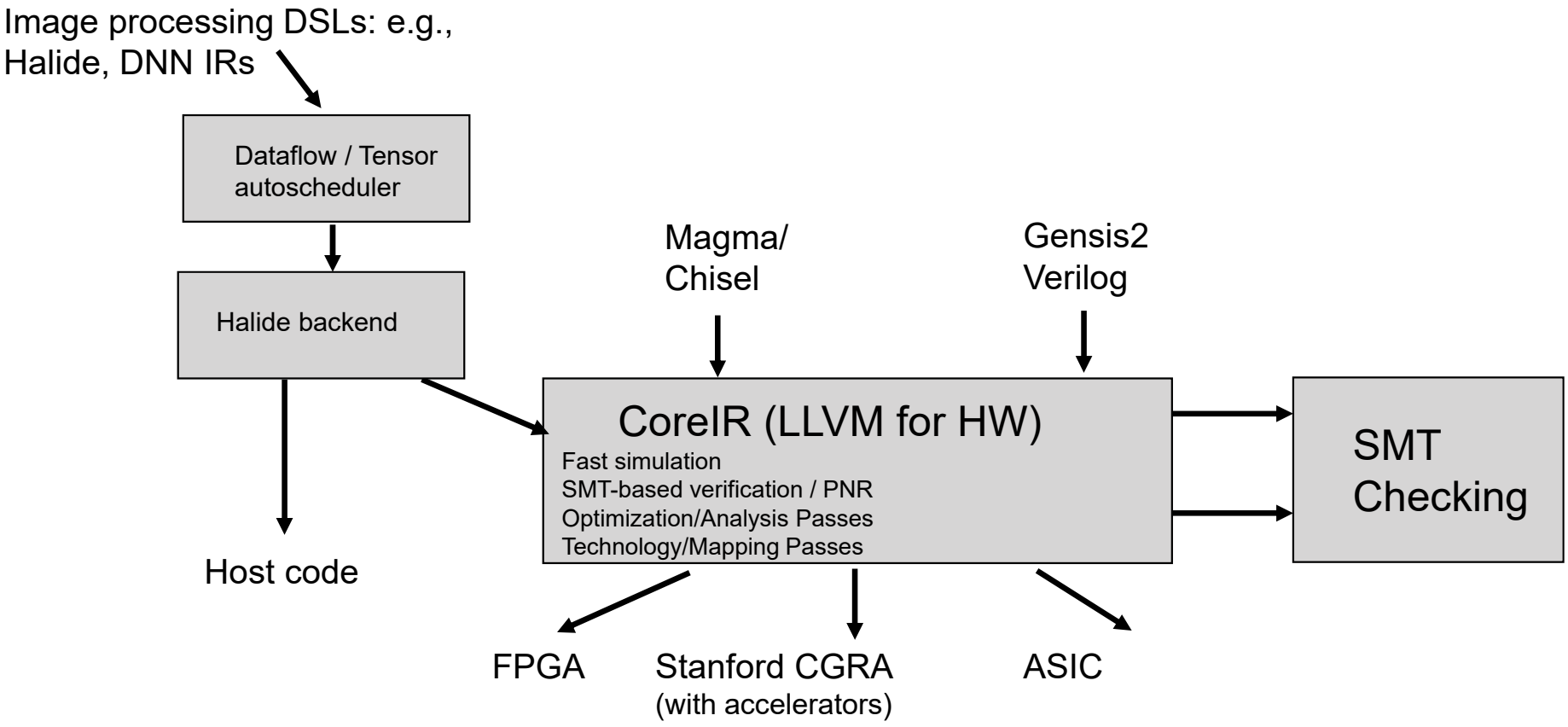
Video Analysis

- Cloud-scale video data mining
  - Image processing and DNN inference on millions of hours of video
  - Offline training on video collections
- Real-time video stream (& multi-stream) processing
  - Computational photography, autonomous vehicles, VR
- Low latency, ultra-low power deployments
  - Always on sensing/wakeboarding, sense/process/display

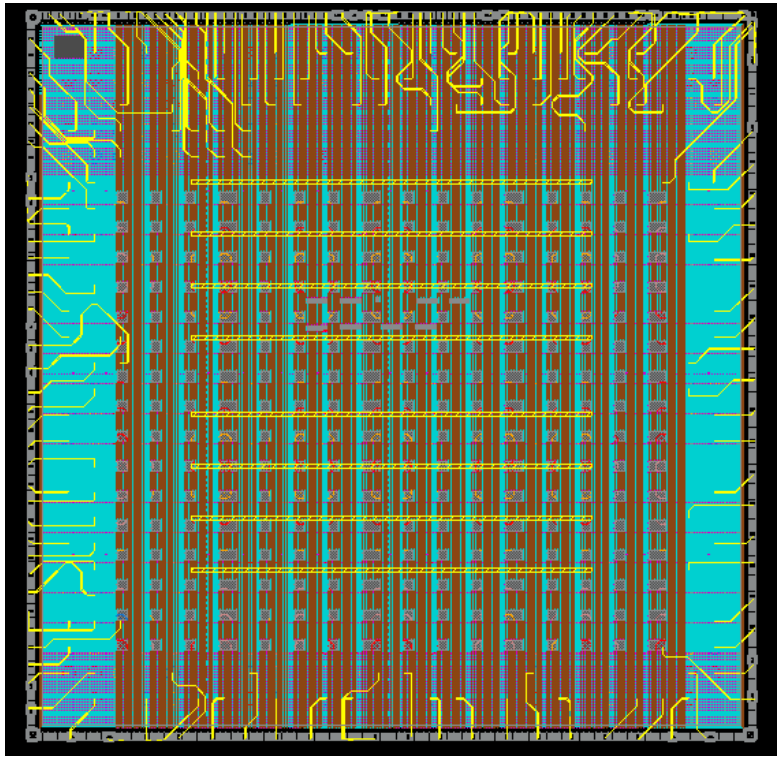
Halide Compiler Work

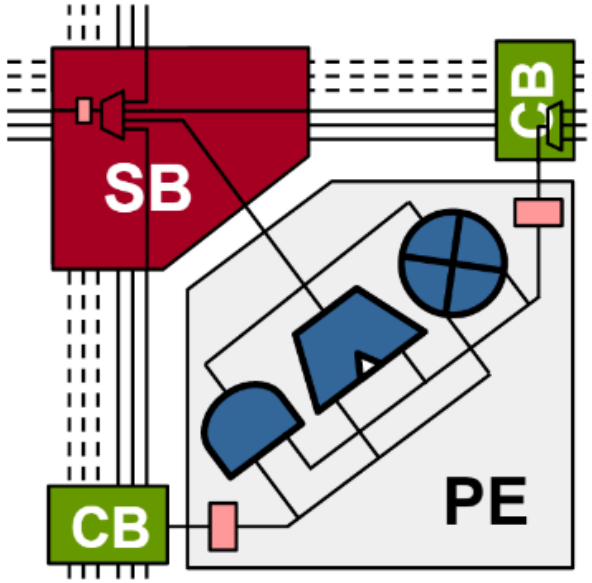
- Auto-schedulers
  - Both for Halide and for Halide like IR (joint work with Google)
- Generating hardware from Halide
  - Jing's work for line buffered pipelines
  - Xuan's work for dense linear algebra (xNN)
  - Steven's system glue to connect them to user applications
- Extending the framework to more dynamic applications

AHA toolchain



CGRA





The Journey is Just Beginning

- Thanks for helping with this project

- It is so nice to be excited about VLSI design again