

Agile Hardware Design

For designing better chips

Priyanka Raina

AHA Retreat

June 23, 2018

Introduction

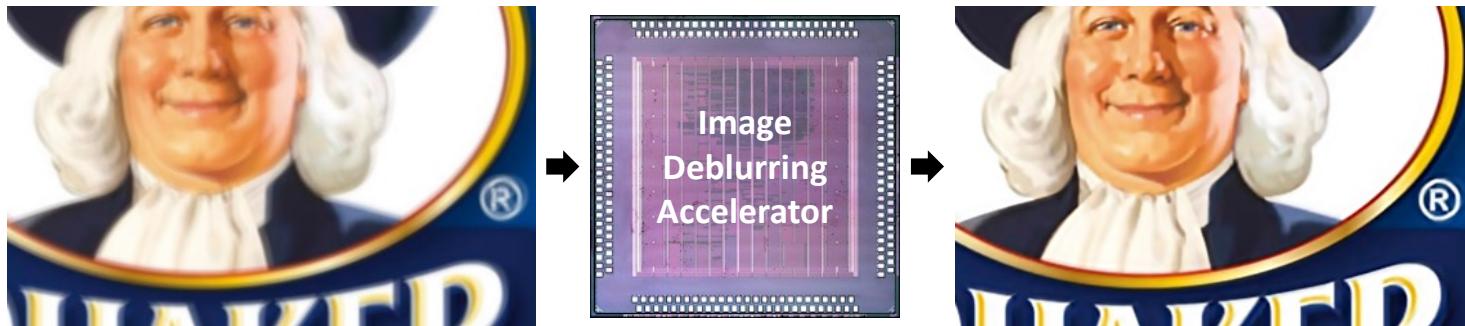
- Completed my PhD from MIT in January
 - Advisor: Anantha Chandrakasan
 - Thesis: Energy-efficient circuits and systems for computational imaging and vision on mobile devices
- Now a visiting research scientist at Nvidia Research
- Will start at Stanford as an assistant professor in September

My PhD in one slide

Energy-Efficient Imaging Accelerators

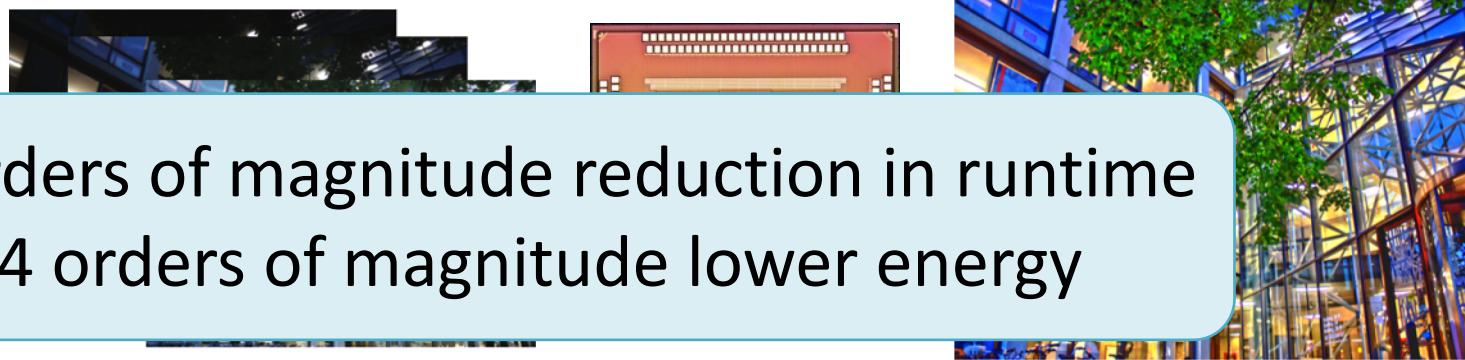
Image Deblurring

[P. Raina, M. Tikekar,
A. P. Chandrakasan
ESSCIR 2016,
JSSC 2017]



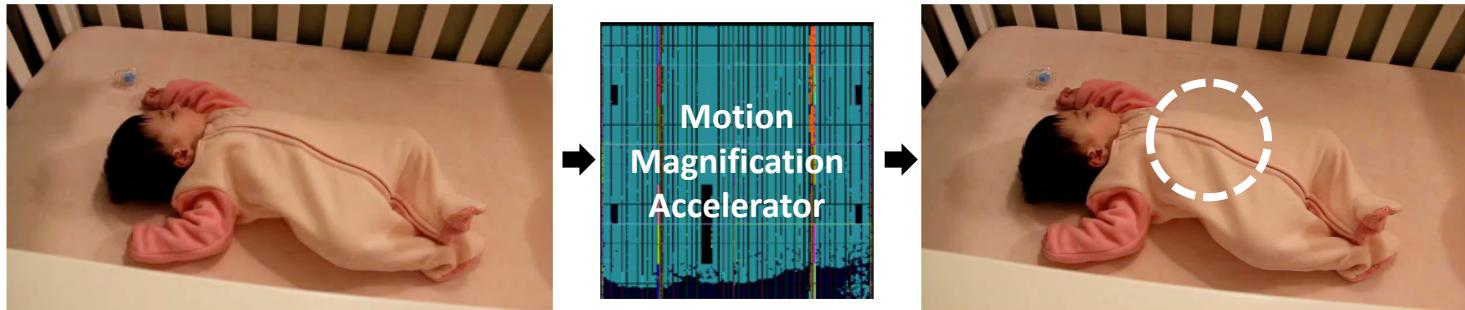
HDR & Low Light Imaging

[R. Rithe, P. Raina,
S. Tenneti, A. P.
Chandrakasan
ISSCC 2013,
JSSC 2013]



Motion Magnification

[P. Raina, D. Jeon, W. T.
Freeman, F. Durand, A. P.
Chandrakasan, In progress]

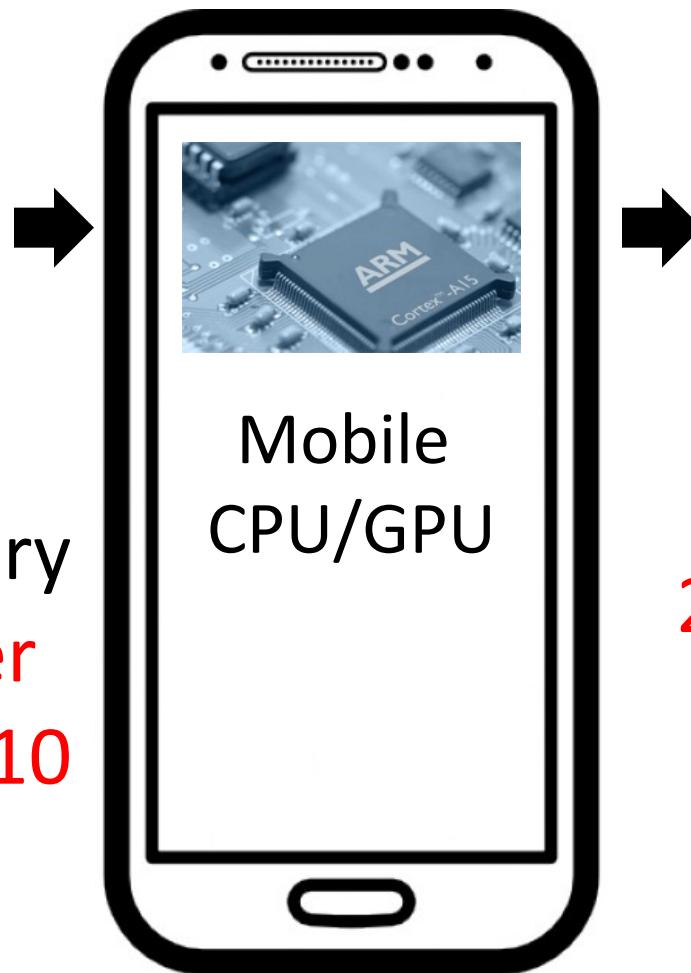


Mobile image processing is expensive

Image Capture



Computation



1810 mAH battery
Phone dead after
deblurring only 10
photos!



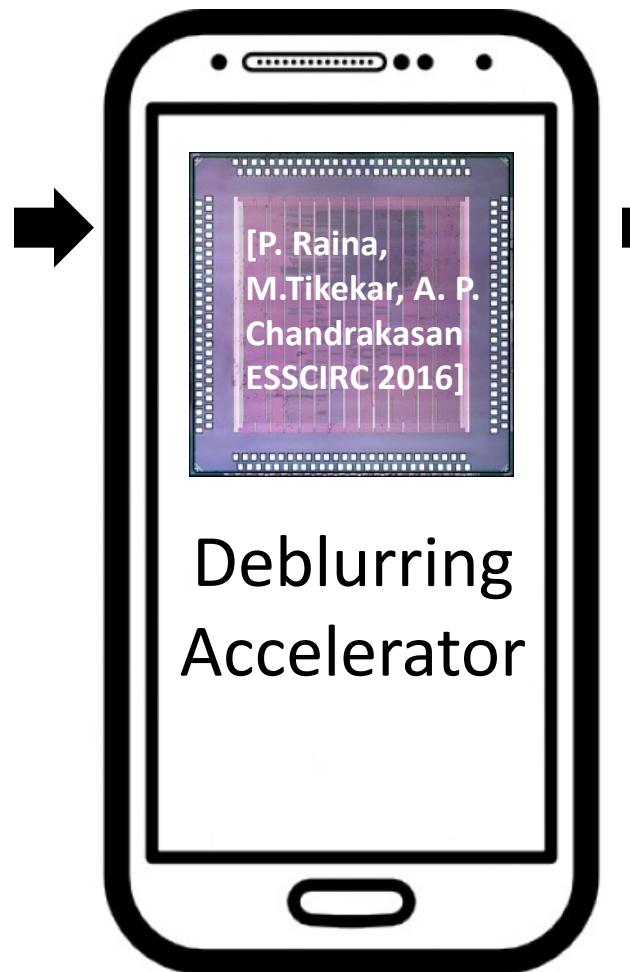
13.6 minutes
2284 J per frame

Accelerators to the rescue

Image Capture



Computation



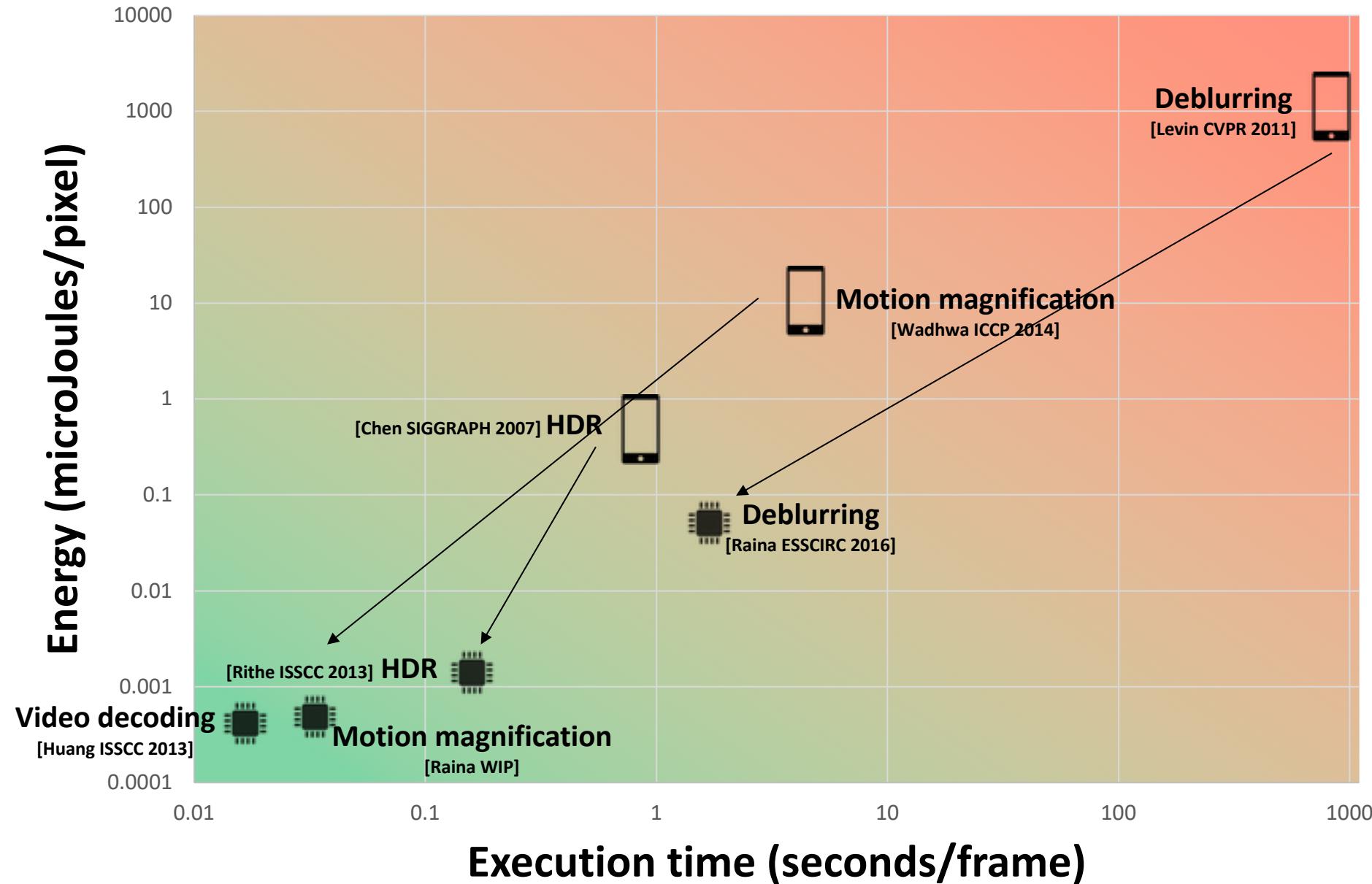
~~13.6 minutes~~

~~2284 J per frame~~

1.7 seconds

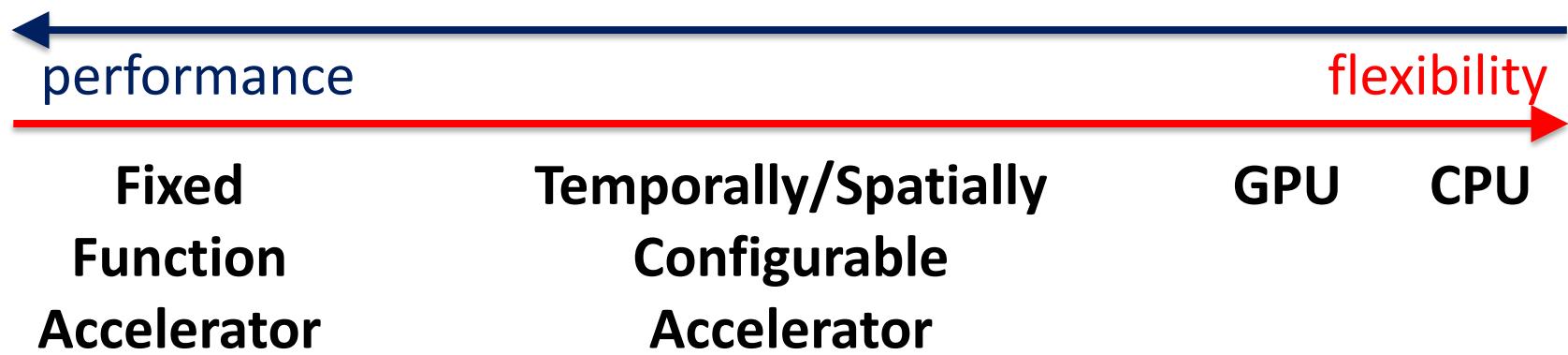
0.1 J per frame

Why are we designing accelerators?

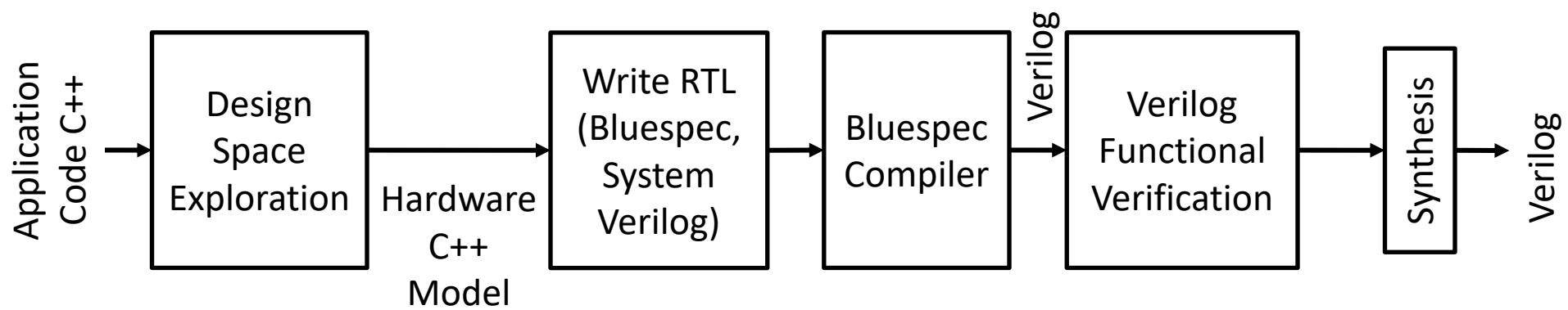


Designing Specialized Hardware

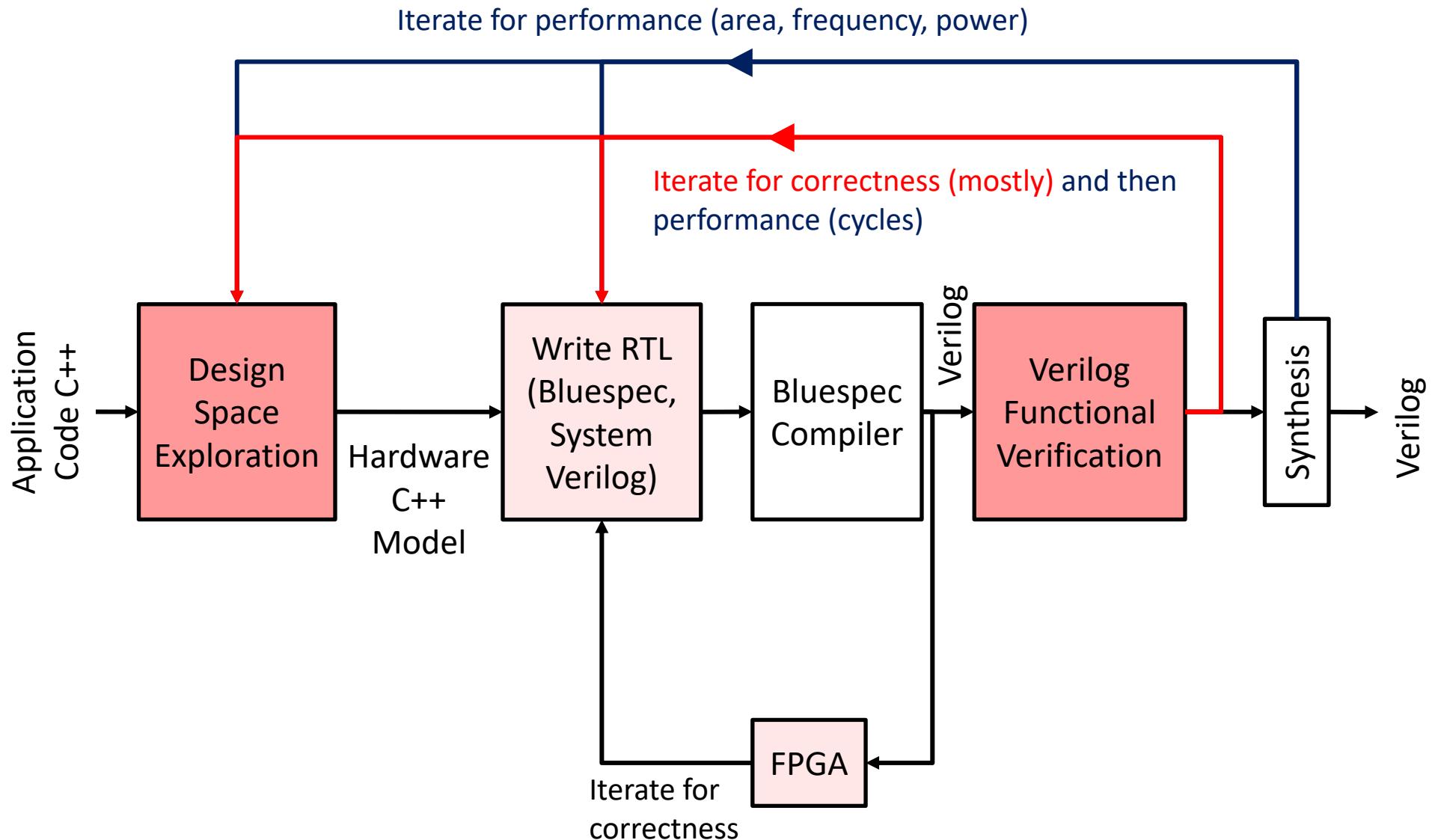
- Future is about applications running on accelerators
 - What applications to specialize?
 - To what degree to specialize?
- Determine it by actually running the application using our end-to-end flow



Design Process



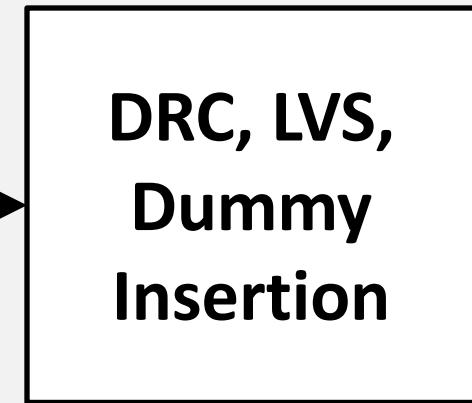
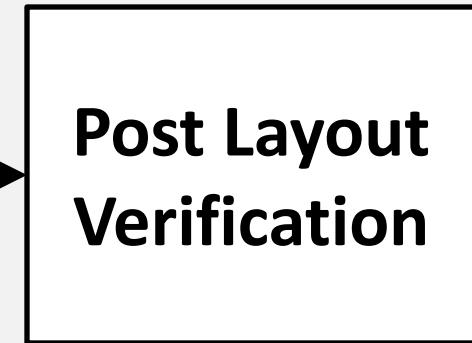
Design Process



Design Process

Backend

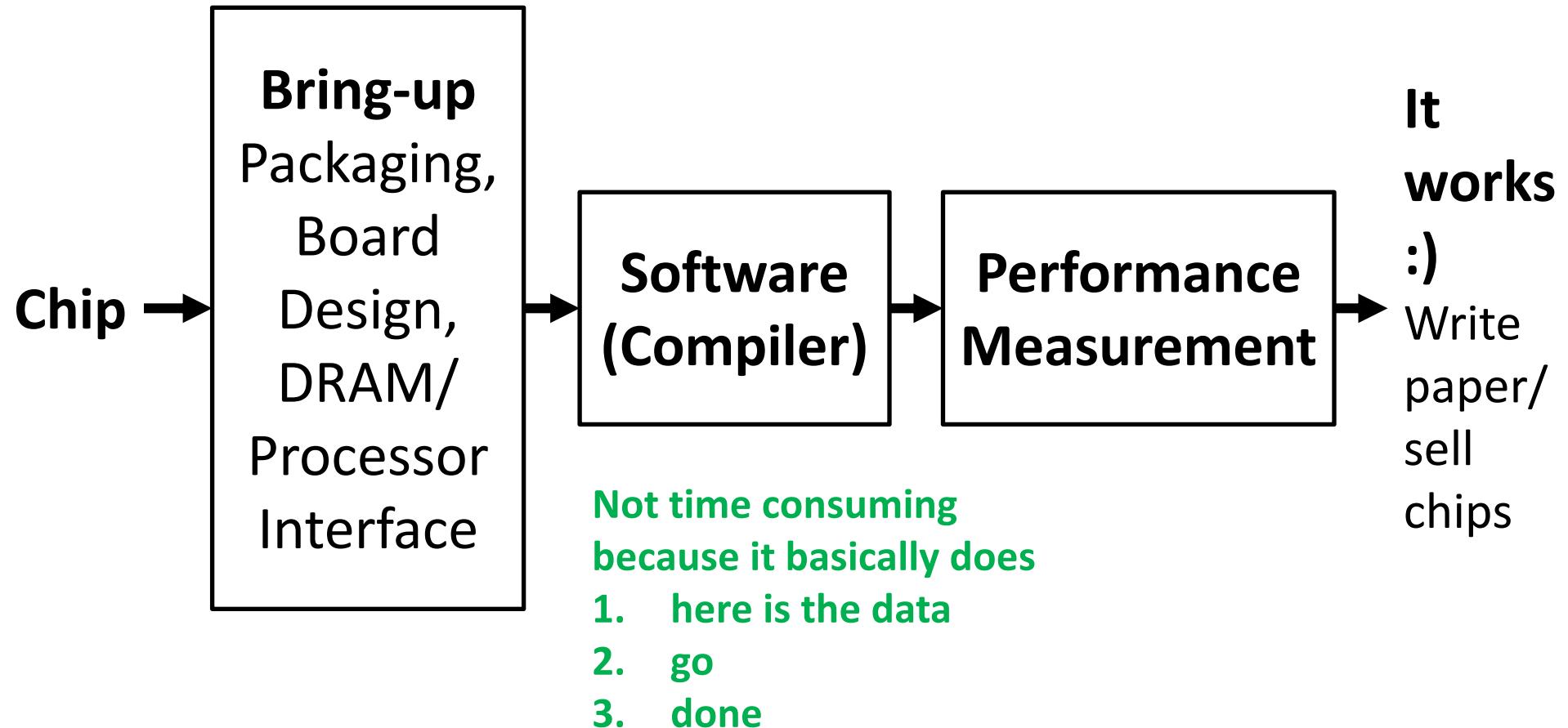
Verilog



Layout

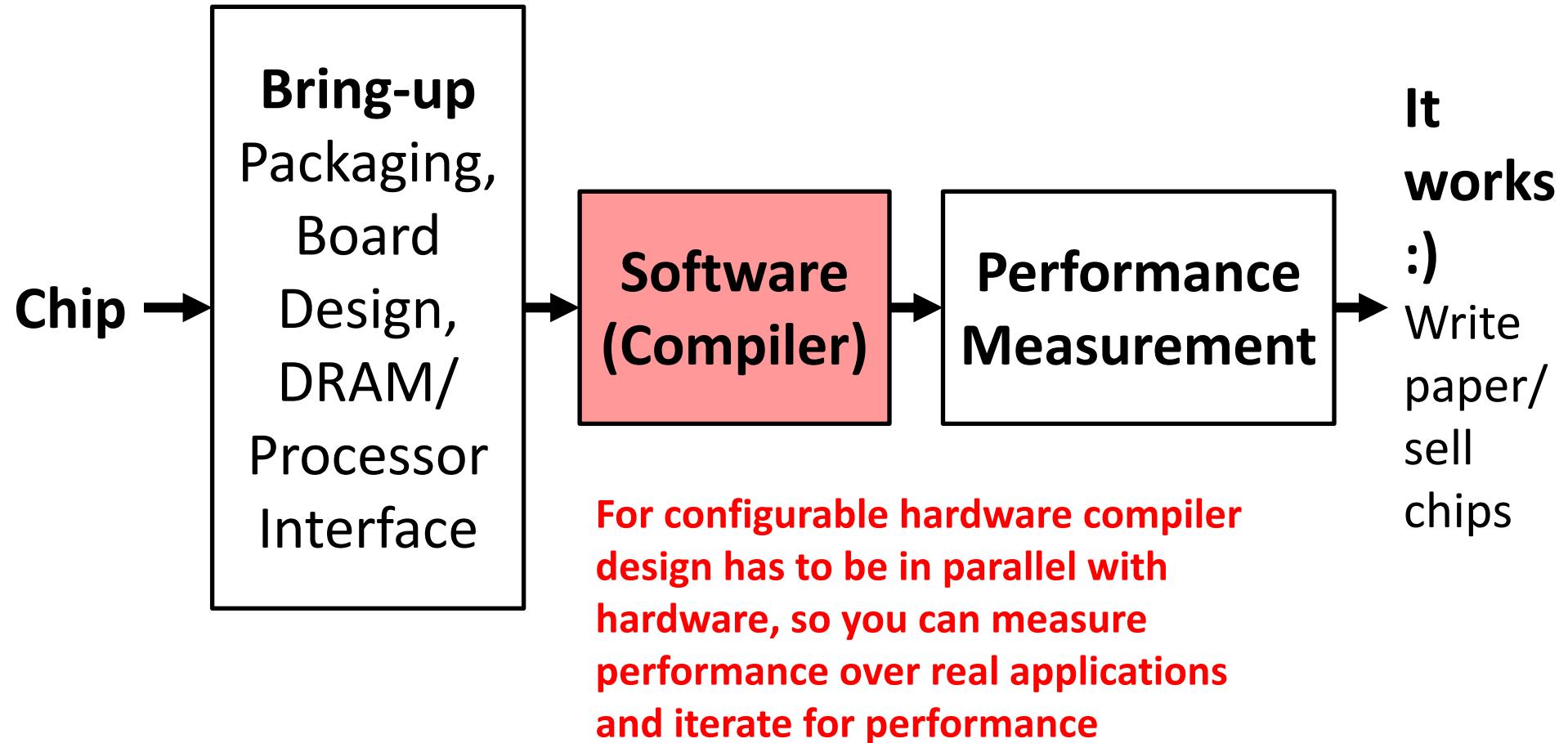
Design Process

Fixed Function Hardware



Design Process

Configurable Hardware

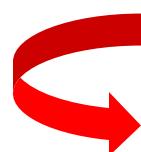


Where does the design time go?

For moderately flexible accelerators

- Most time consuming
 - Design space exploration
 - Software
 - Verification
- Moderately time consuming
 - Writing RTL
 - Iterating for performance

Depends on



What do I want to work on?

- Designing efficient accelerators by
 - (Incremental) design space exploration (with increasing levels of fidelity) using real applications
 - (Incremental) specialization -- only where required for performance
 - For these joint design of software (both scheduler and driver) is critical
- Architectural models that give quick feedback on performance

Hardware generators that expose parameters that are “important” for design space exploration

How to efficiently search through the parameter space

How to search for the best schedule