Stanford University

Amber Analysis

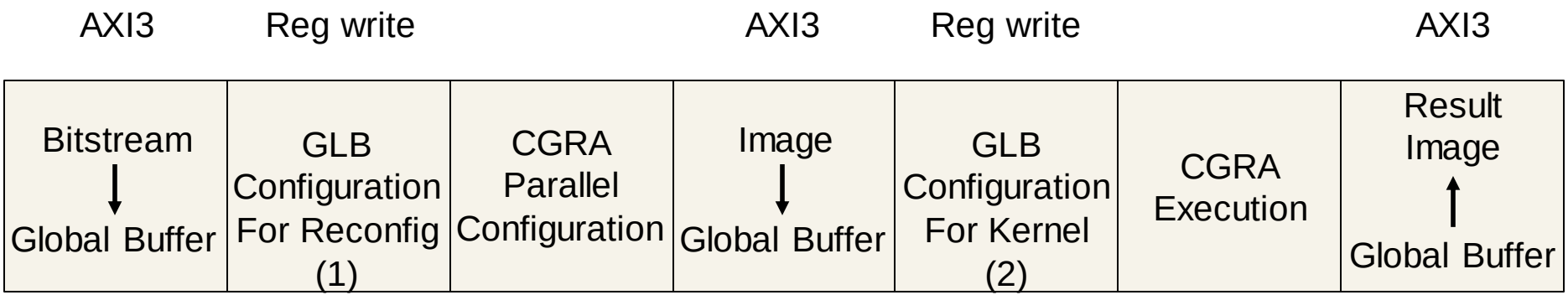
Kalhan Koul, Po-han Chen

Outline

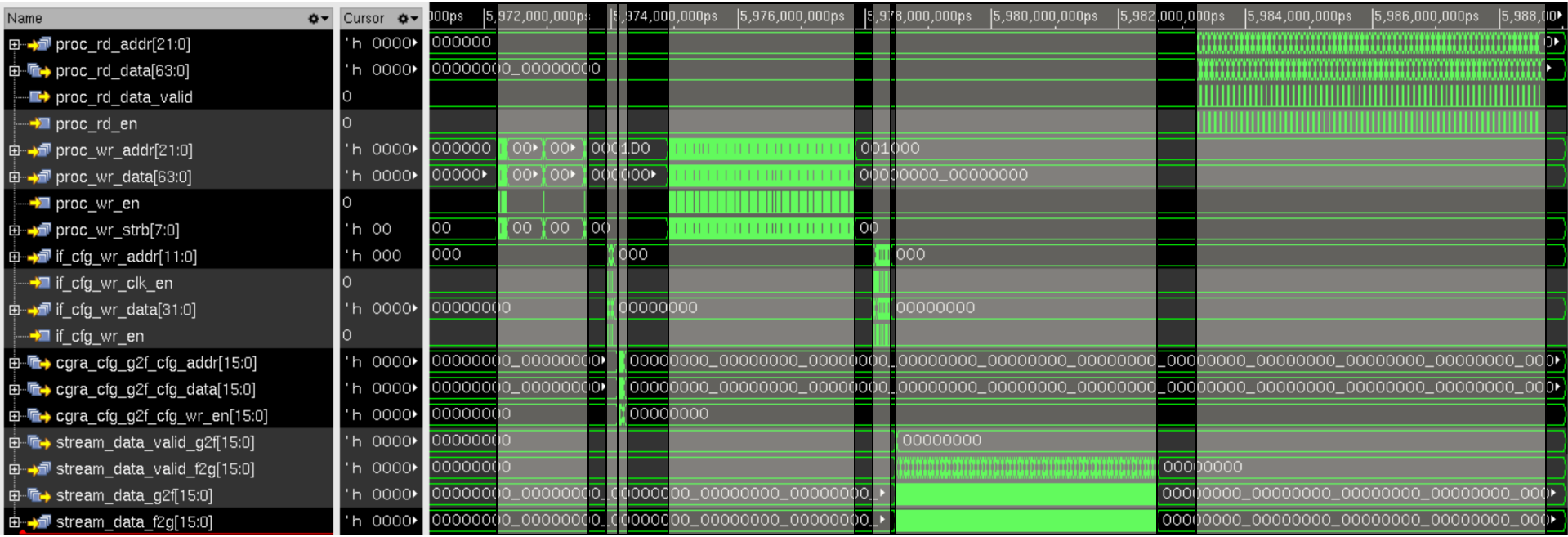
- Runtime
- Area
- Power
- Showcasing Amber
  - How far are we from industry competitors

Runtime

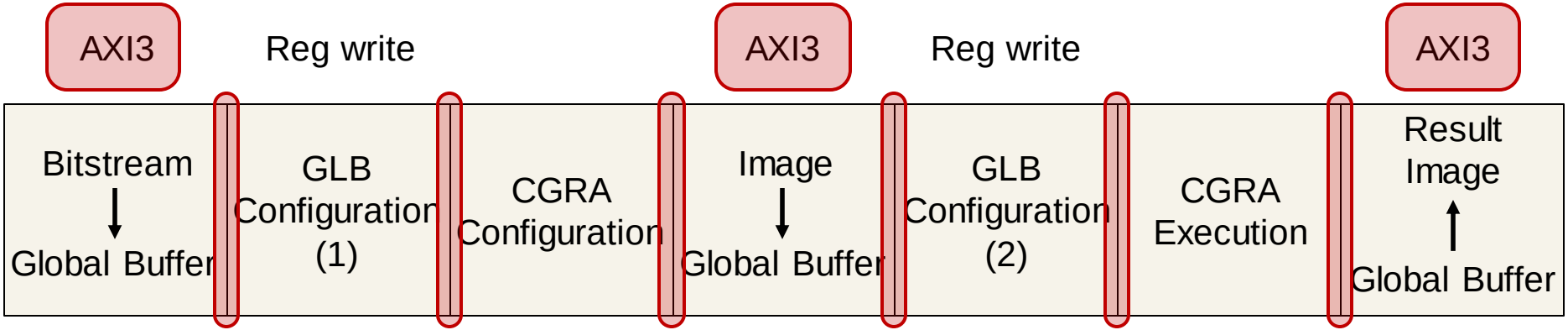
SoC Runtime Components



SoC Runtime Components



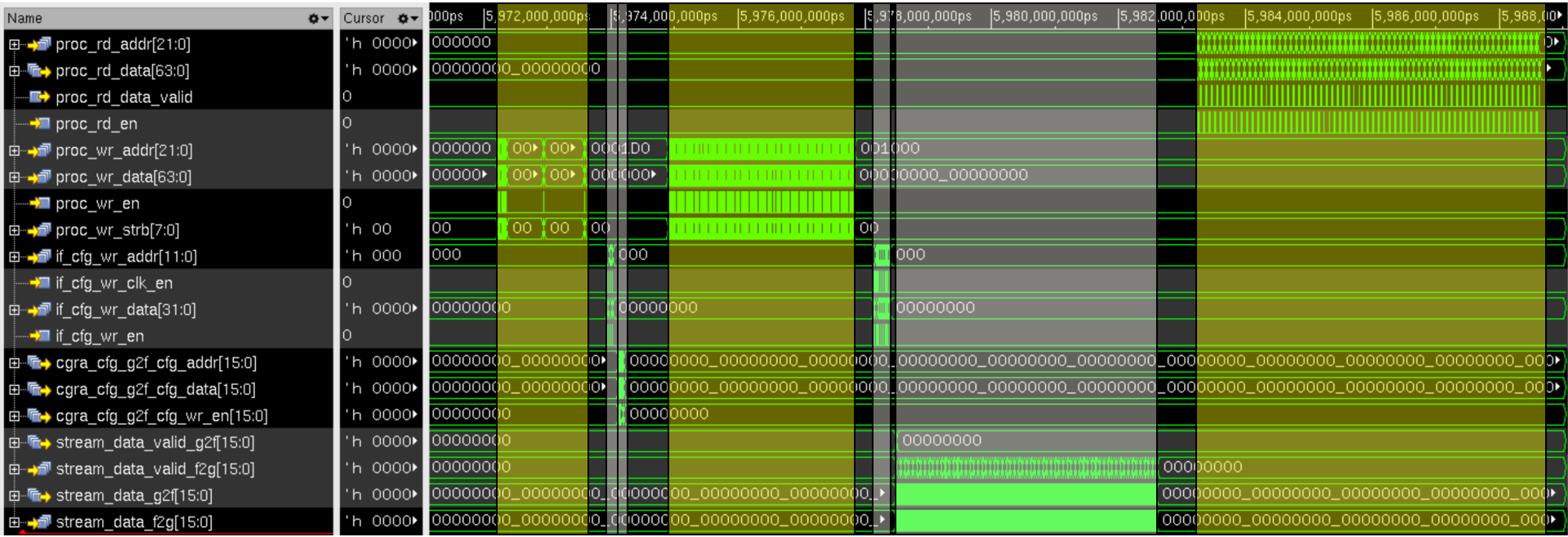
SoC Runtime Components



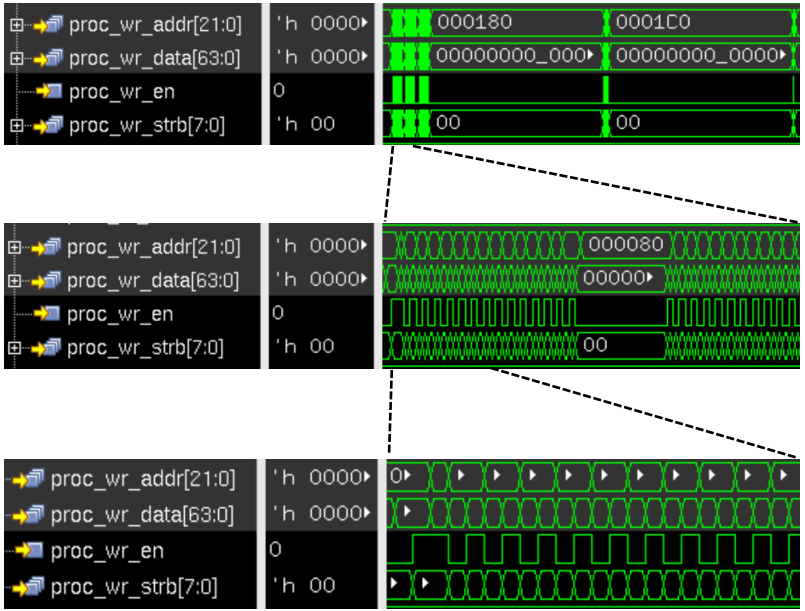
Idle cycles (bubbles) found in:

1. AXI-3
2. inter-stage transition

Runtime Problem 1: Bubbles in AXI-3



Runtime Problem 1: Bubbles in AXI-3

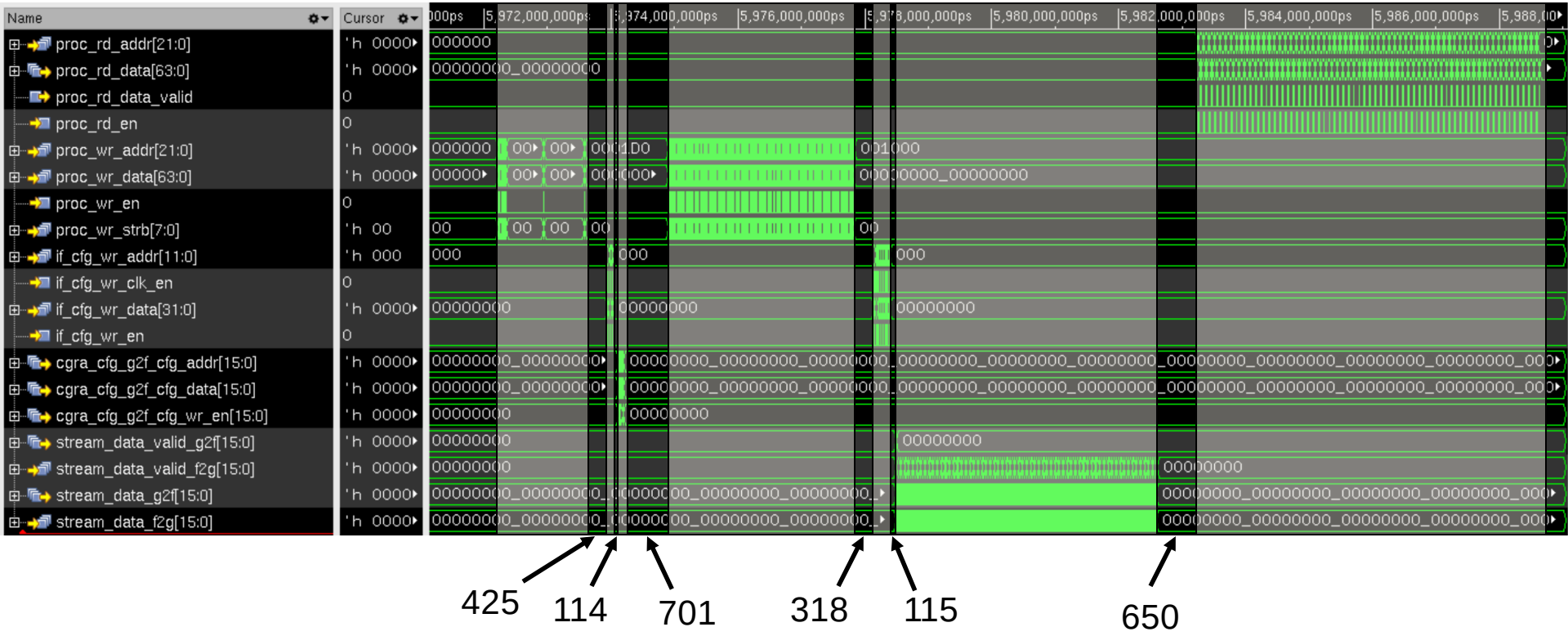


- Burst length changing overhead = 588 cycles
- Solution: Padding the data

- Inter-burst overhead = 14 cycles
- Solution: Change AXI-3 to AXI-4

- Frequency discrepancy between host and CGRA
- Solution: No need to deal with this

Runtime Problem 2: Inter-Stage Idle Cycles

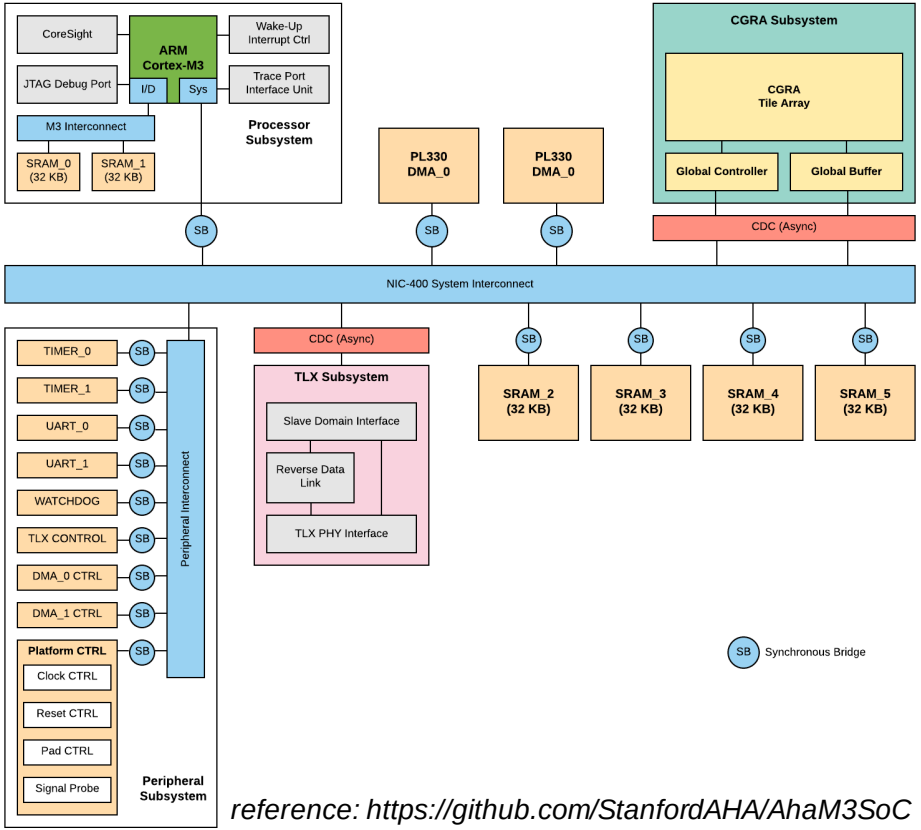


Runtime Problem 2: Inter-Stage Idle Cycles

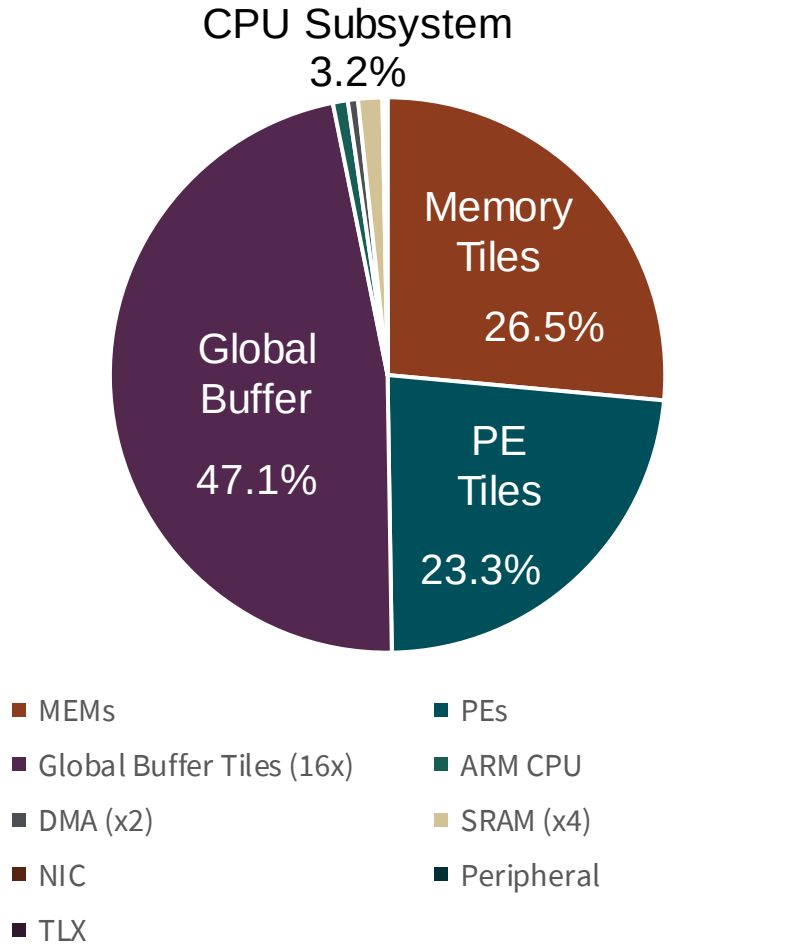
- Account for 15%~20% of total runtime
- Occurs when there is an interrupt handler in firmware
  - But interrupt handler shouldn’t take hundreds of cycles…
  - Still under investigation

Area

Amber SoC **Cell Area** Breakdown



reference: https://github.com/StanfordAHA/AhaM3SoC

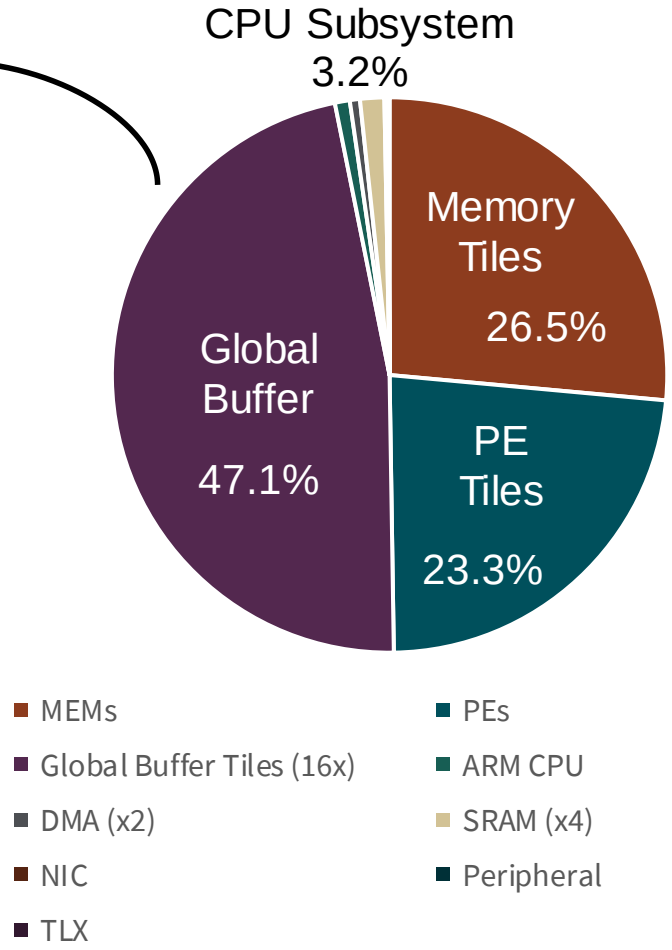


Cell Density is Low

$$Density = \frac{\sum(cell\ area\ of\ main\ components)}{PostP\&R\ Total\ Area}$$

$$= \frac{7.9\ mm^2}{14.2 mm^2} = 55.6\%$$

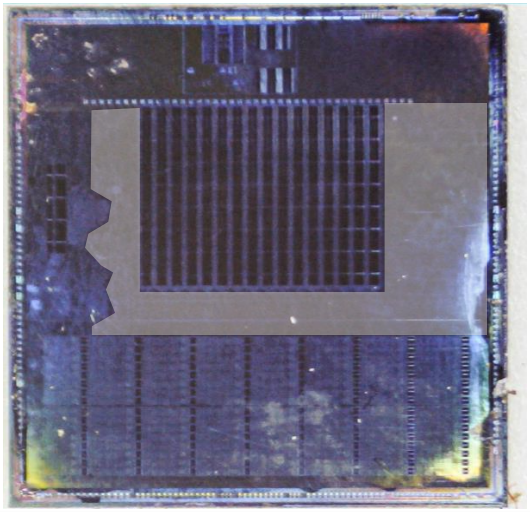
- Density
  - Memory Tile = 85.8%
  - PE Tile = 86.2%
  - Tile Array = 63.8%
  - SoC = 55.6%

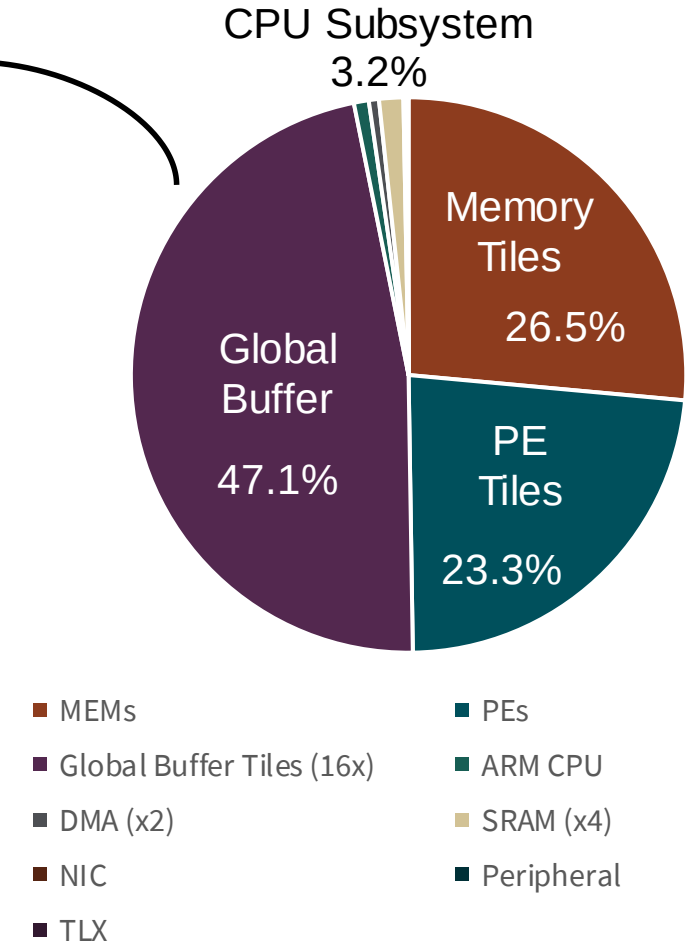


Cell Density is Low

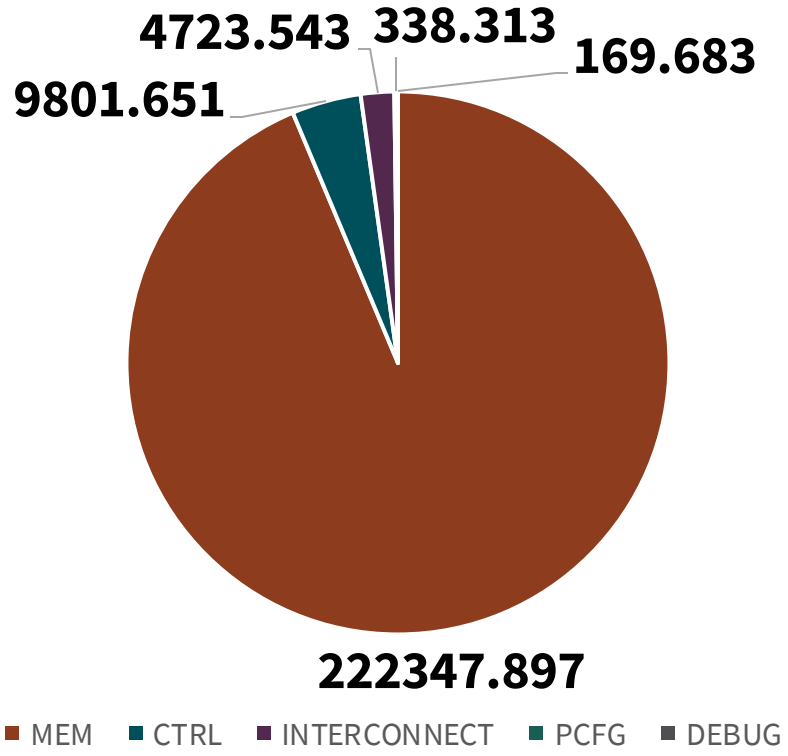
$$Density = \frac{\sum(cell\ area\ of\ main\ components)}{PostP\&R\ Total\ Area}$$

$$= \frac{7.9\ mm^2}{14.2 mm^2} = 55.6\%$$

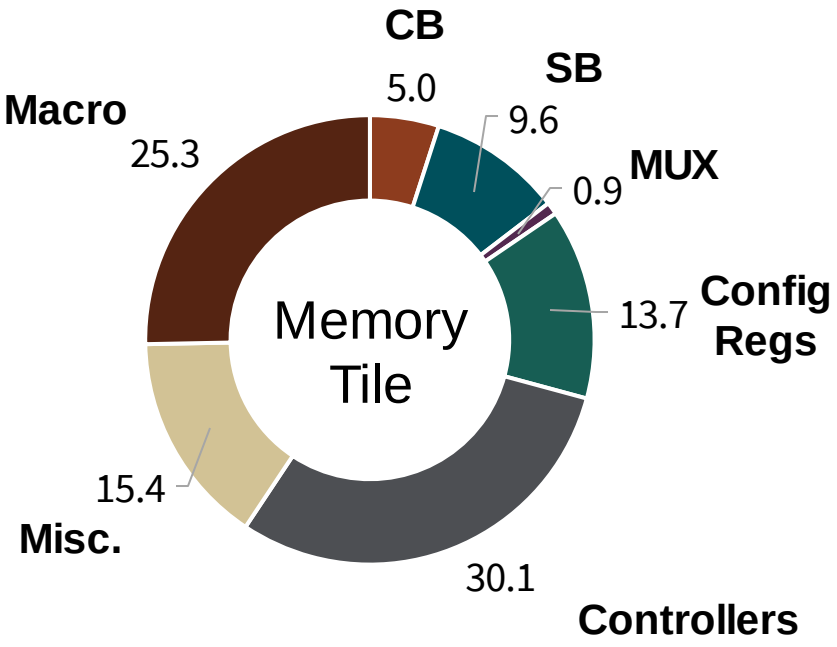


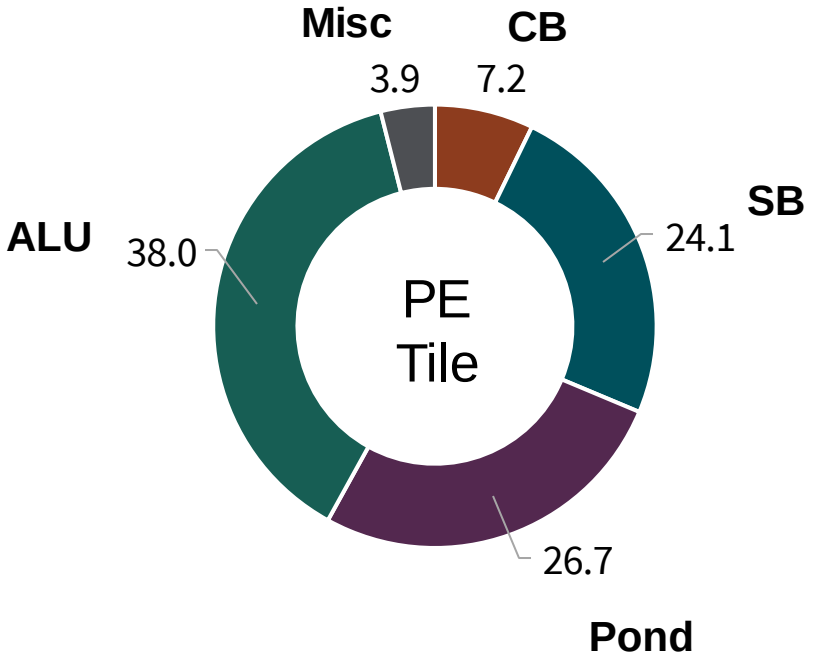


Global Buffer Area Breakdown



Memory Tile / PE Tile Area Breakdown





Power

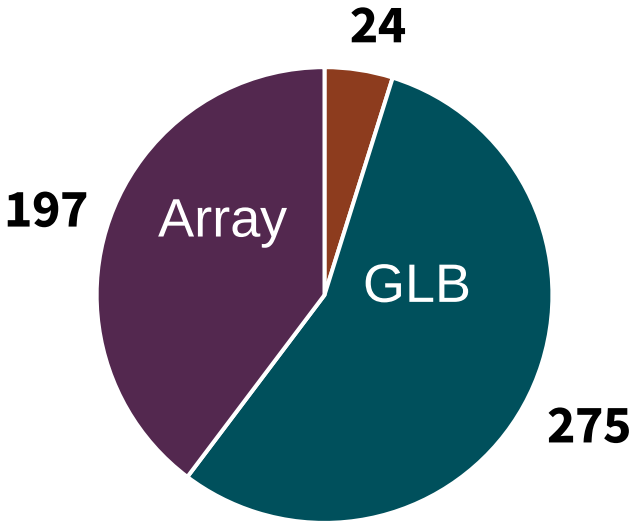
Power Breakdown Disclaimer

- Goal of this section is **not** to prove correlation with real chip, but to suggest improvements for Onyx
- We **have** correlated a single add operation to within 20%
- Why is full chip correlation difficult?
  - Hold Violation in GLB means we cannot use sdf annotation on current design
  - Need to match freq with simulation
  - Gate level simulation runtimes 24+ hours
- Power Analysis
  - Power of chip blocks (SoC, GLB, Tile Array)
  - Power Group (clock network, combinational, memory, etc.)
  - Functional Group (ex. PE tile: alu, pond, SB, CB etc.)

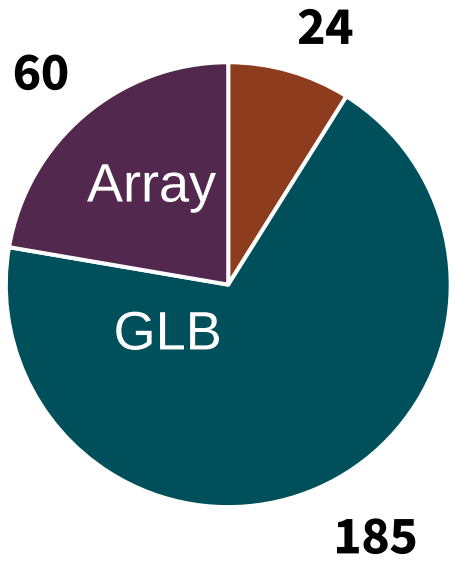
Stanford University

SoC Block Power Breakdown (Gaussian Unroll=1)

Baseline - 496 mW @ 1ns



Power Gate Array, Stall Unused GLB - 270 mW @ 1ns

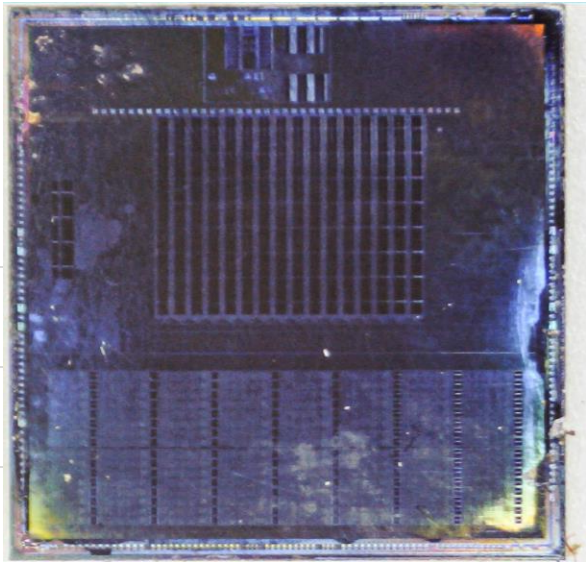


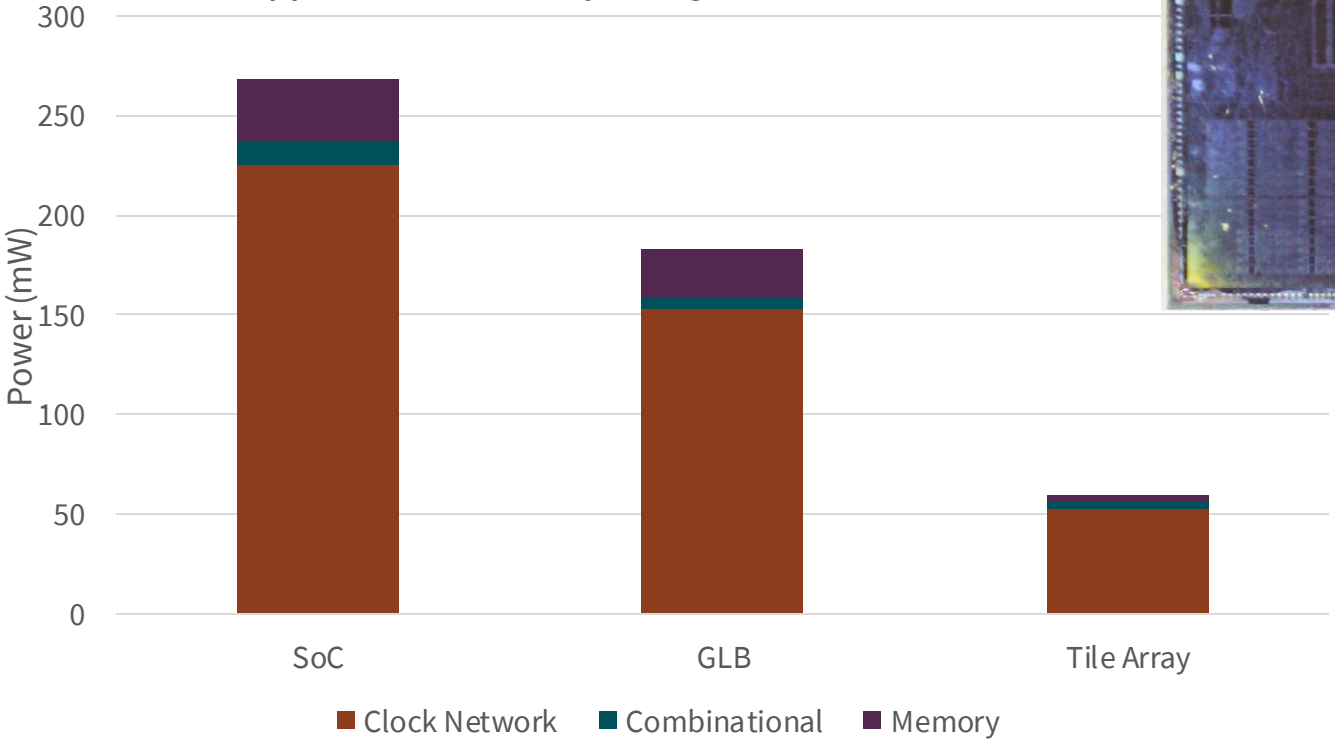
- 1 GLB tile for input output
- 9 PE tiles and 2 MEM tiles
- GLB consumption high even when unused tiles are clock gated

Stanford University

SoC Power Breakdown Unroll = 1

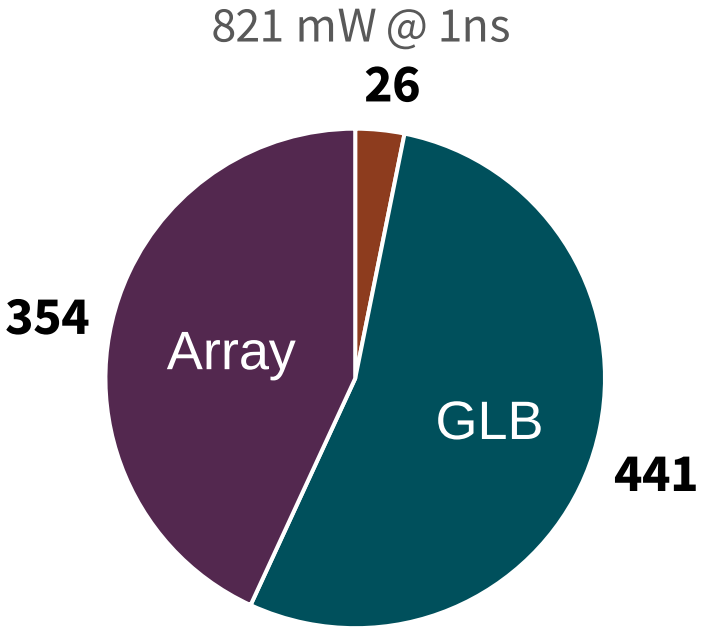
- All blocks have > 80% clock network power consumption
- Note: Small application, unfairly weighs clock network





Stanford University

SoC Block Power Breakdown (Gaussian Unroll=16)

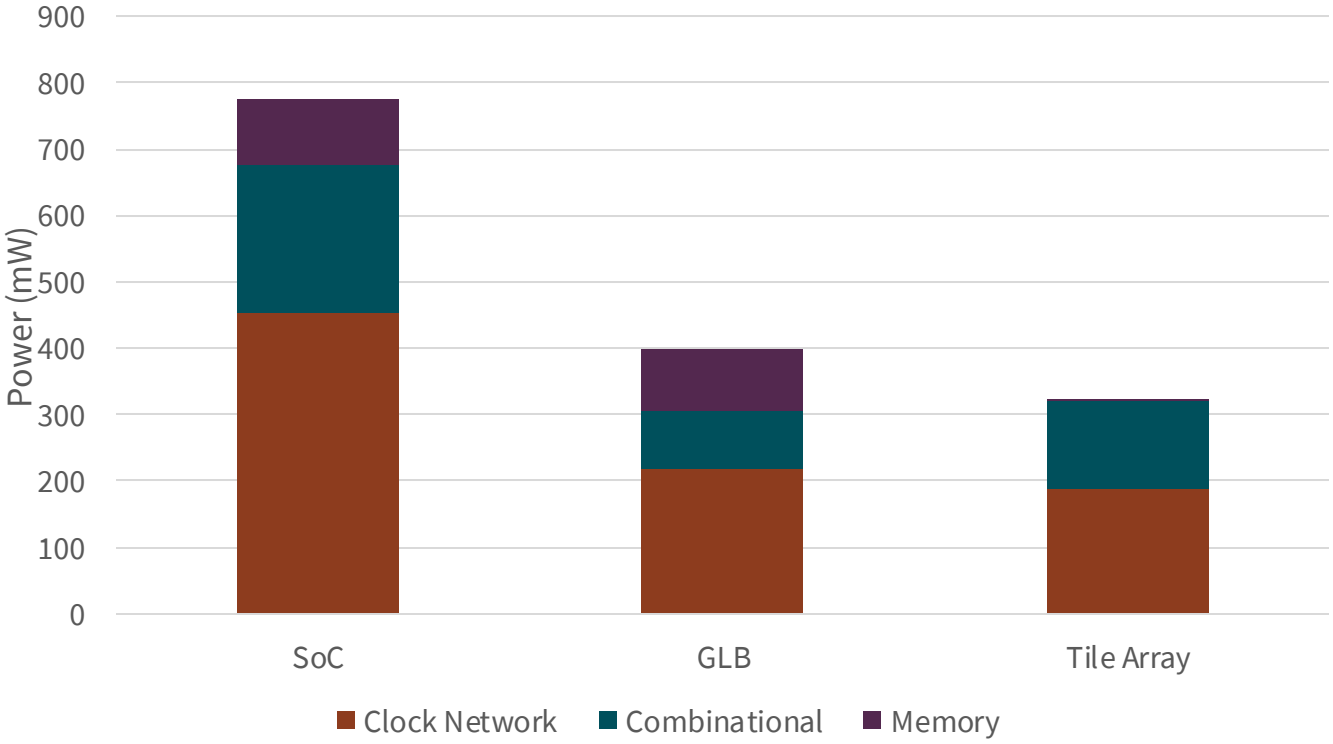


- 16 GLB tiles for input output
- 144/384 PE tiles and 16/128 MEM tile

Stanford University

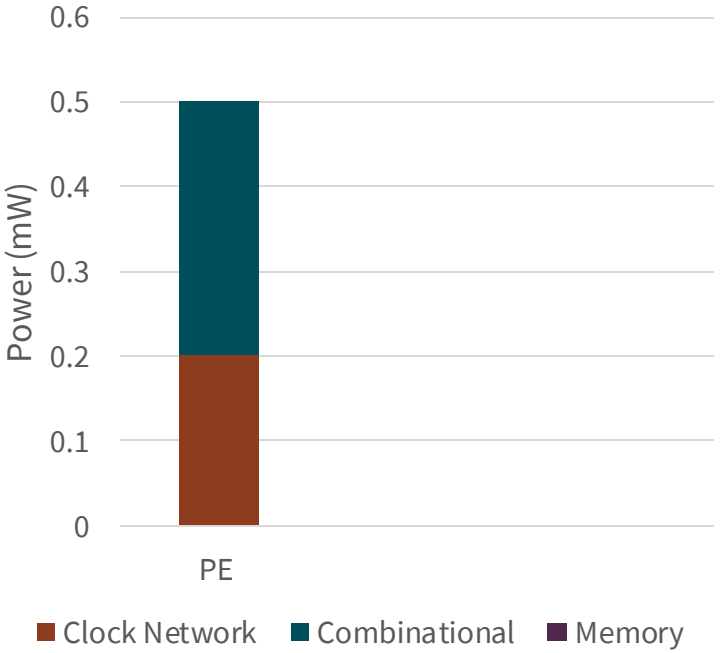
SoC Power Breakdown Unroll = 16

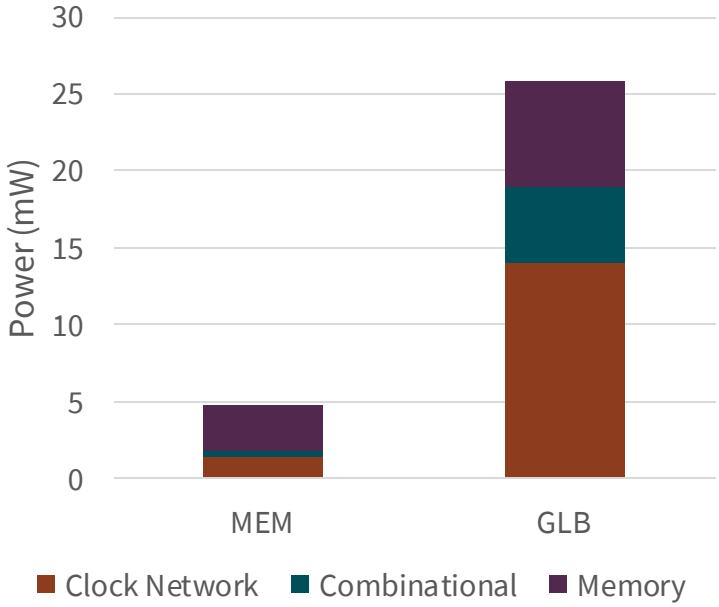
- All blocks have ~ 50% clock network power consumption
- Note: Looks correct for a larger application



Stanford University

In Use PE, MEM, and GLB Tile Breakdown

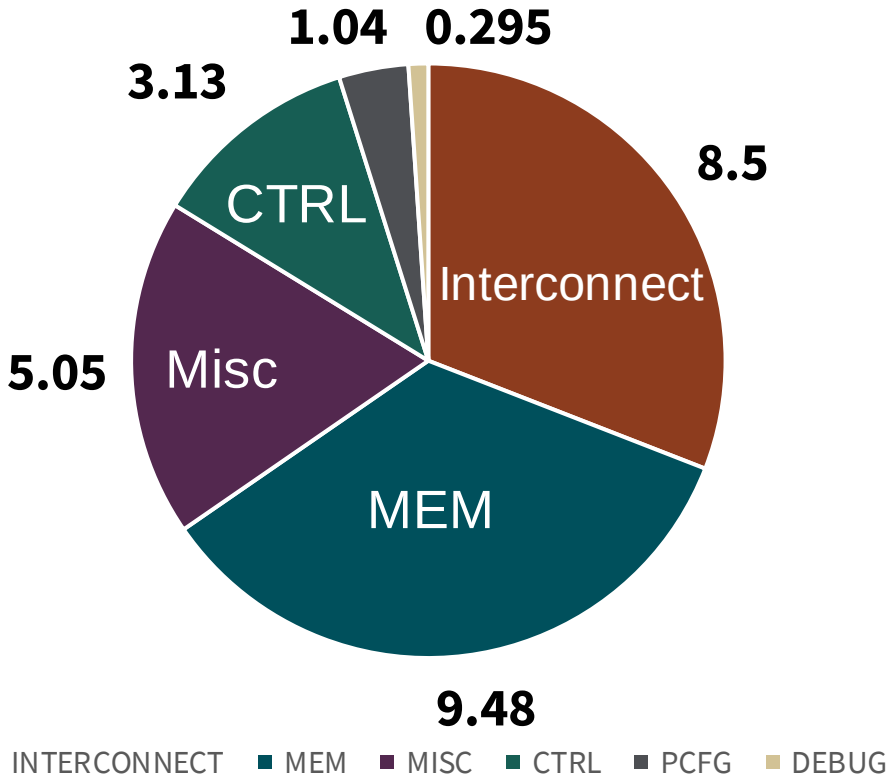




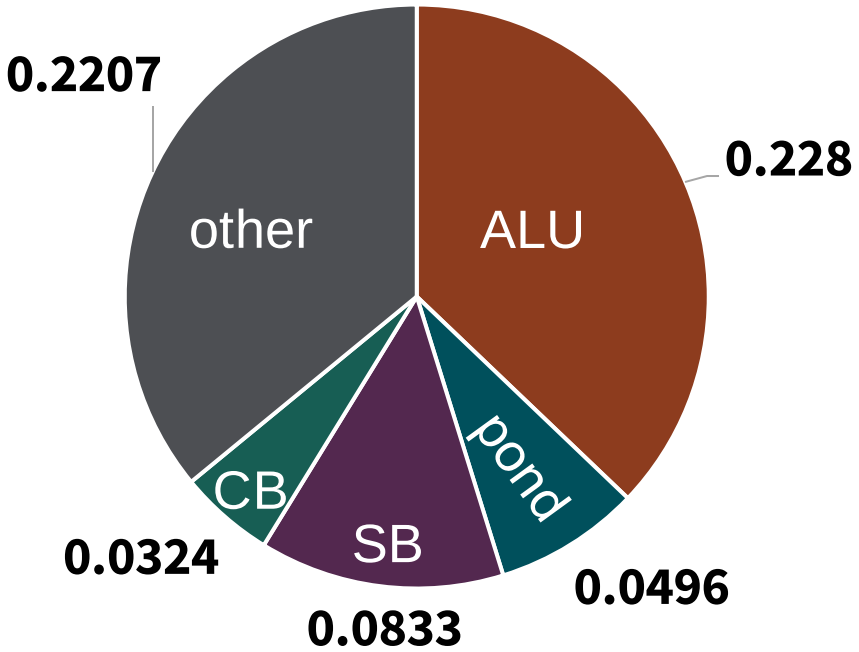
- These blocks have lower clock network power consumption, but MEM and GLB still > 50%

Global Buffer Tile Power Breakdown

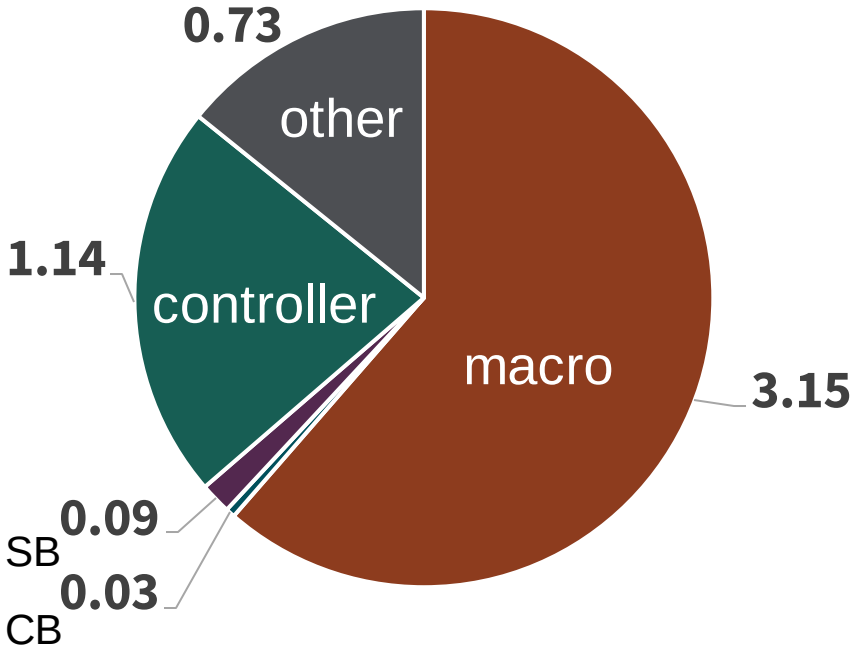
- SRAM – composed of 2 banks of 8 macros each
  - Used macros (2) – 17%
  - Unused macros (14) – 8%
- Interconnect - 30.92%
  - Due to tall/skinny shape, several buffers are needed to meet timing
- Control - 11.39%



PE and MEM Power Breakdown (mW)

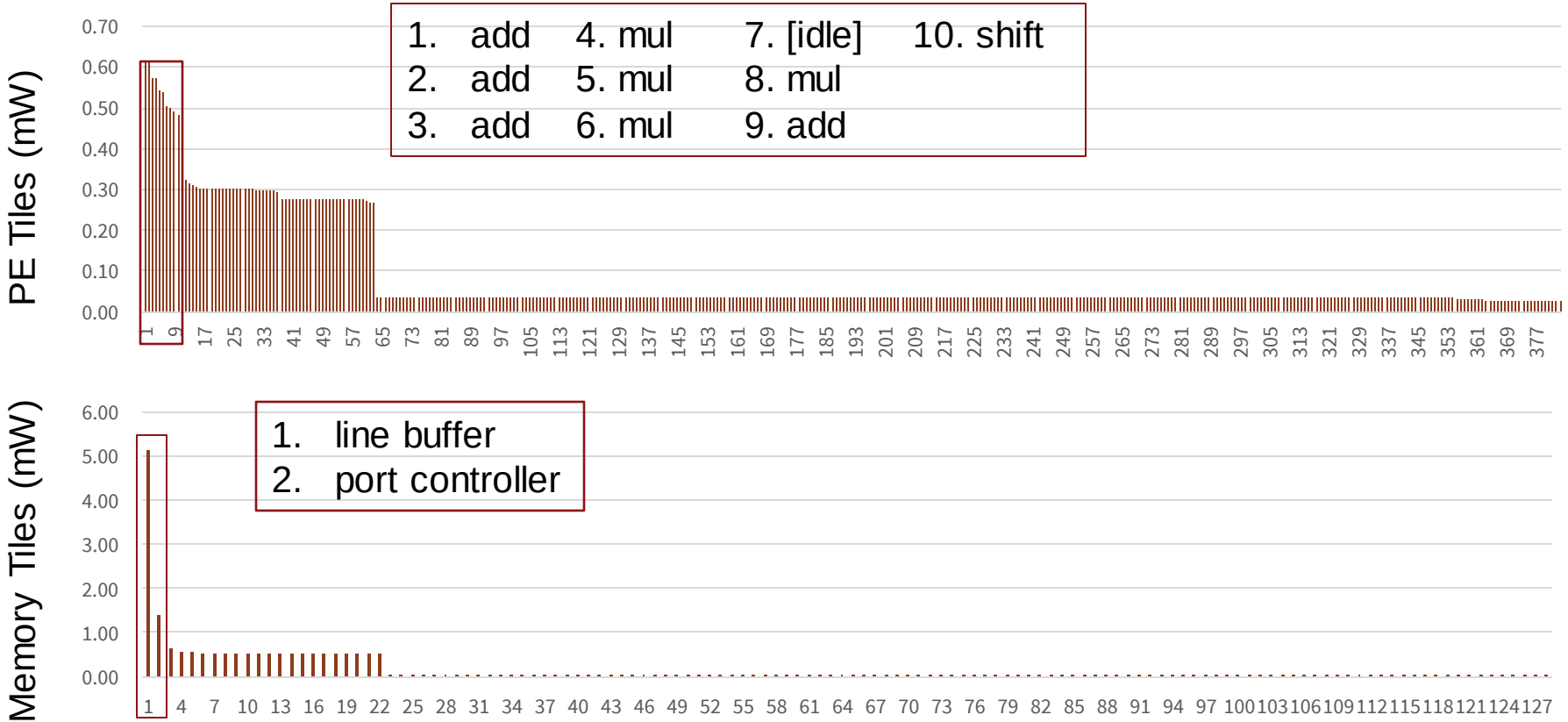


- Other is mostly clock buffers
- pond not used in this app

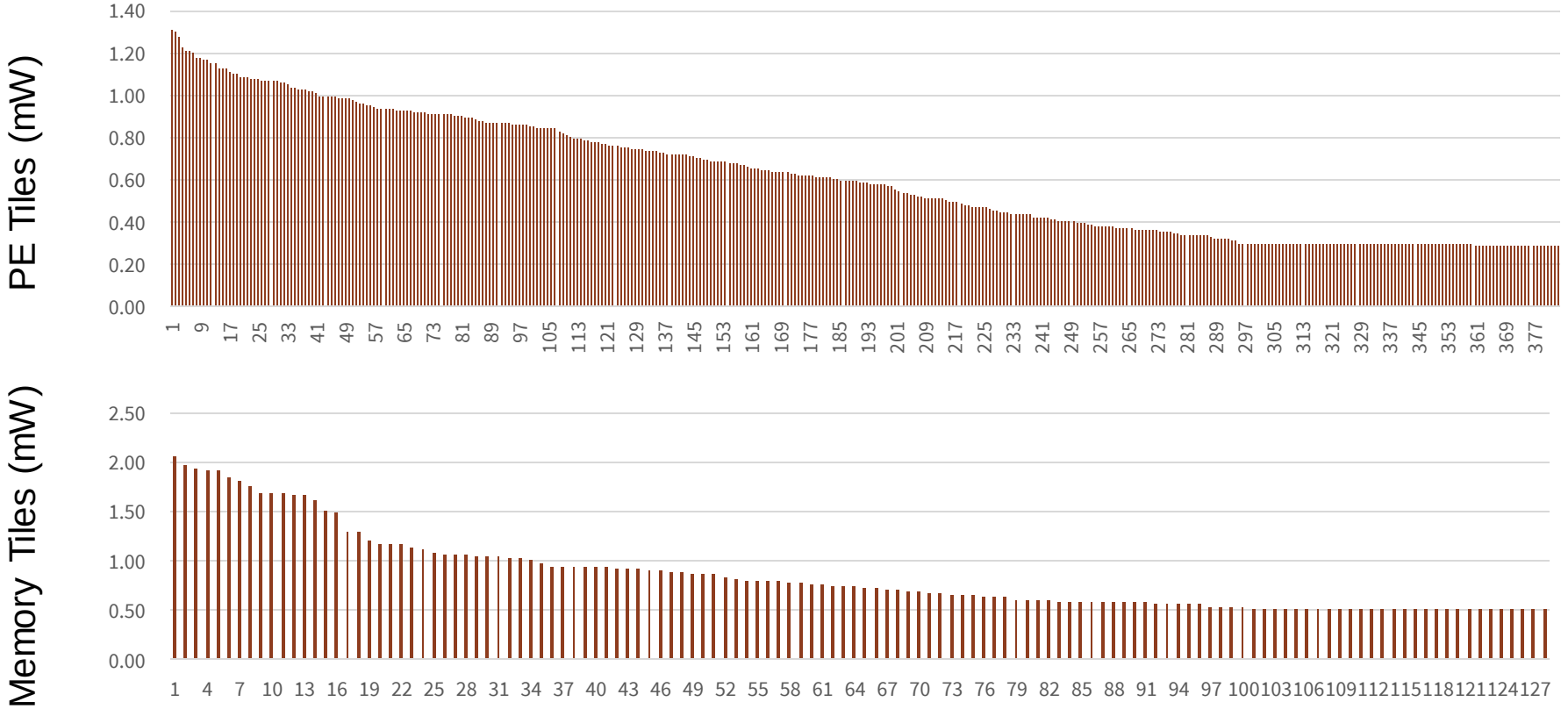


- Configurations regs and muxes are flattened out and counted in others

PE/MEM Tile Power Histogram, Unroll = 1



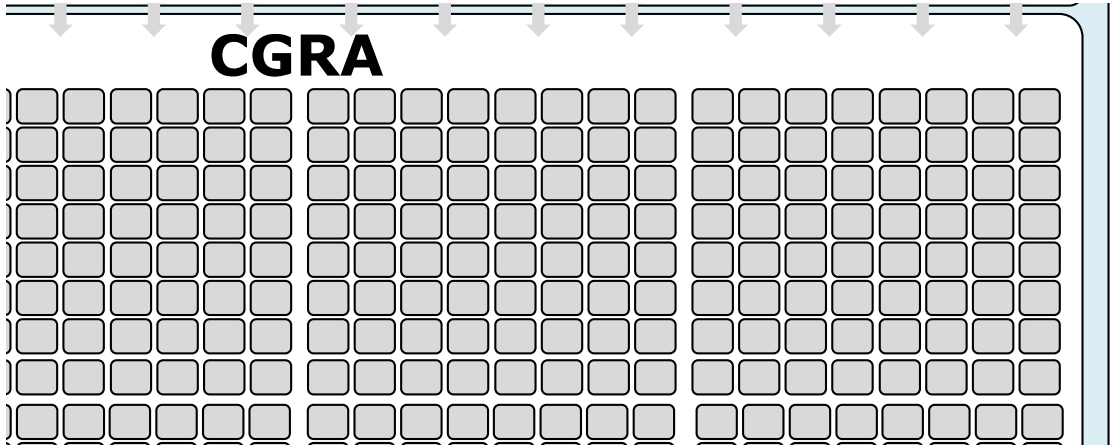
PE/MEM Tile Power Histogram, Unroll = 16

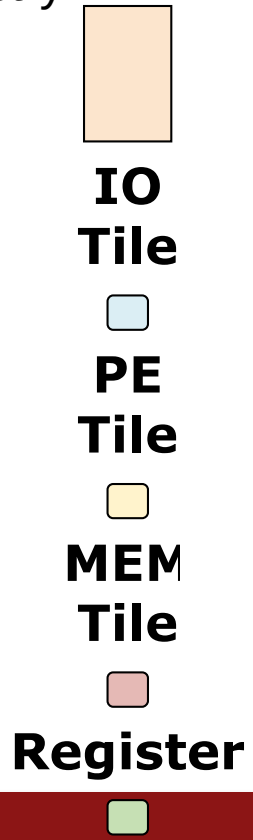


Showcasing Amber

Comparing our Applications against Industry Competitors

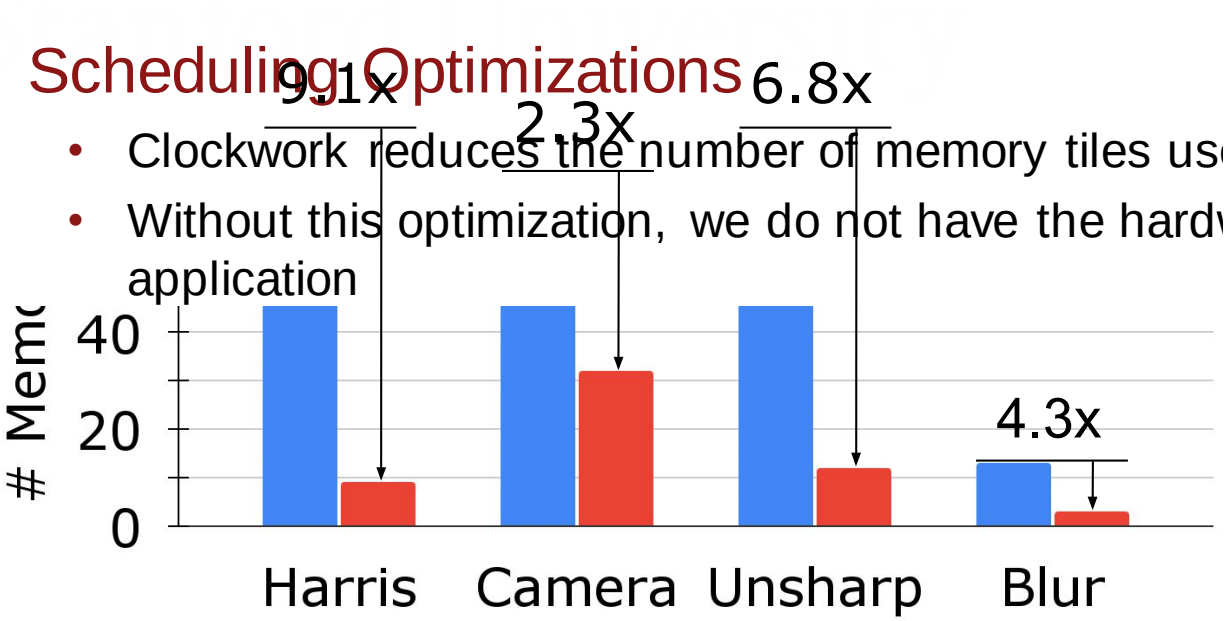
- Gaussian/Blur, Harris, Unsharp, and Camera compared against industry hardware
- Apply a Series of Optimizations
  - Scheduling Optimizations (Clockwork)
  - Unrolling
  - Pipelining

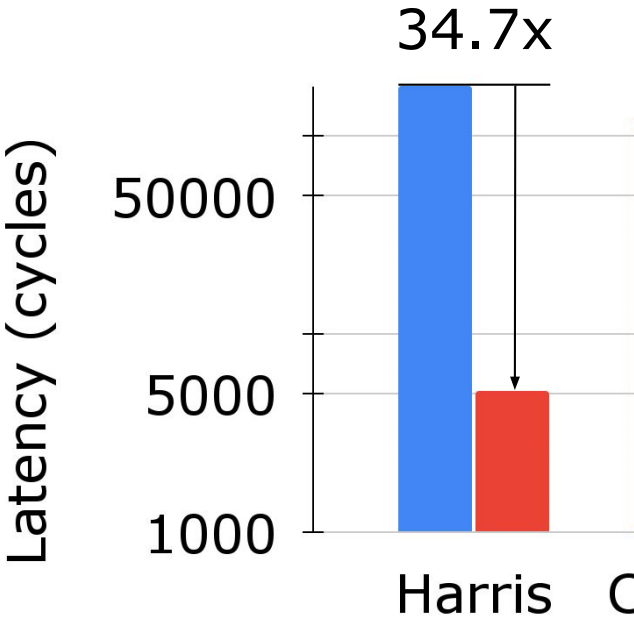




Scheduling Optimizations

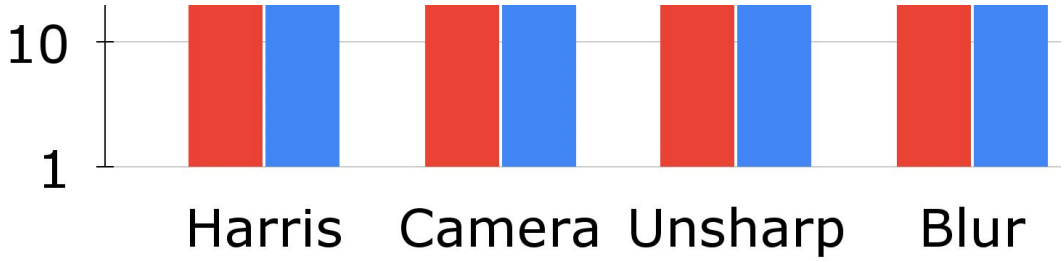
- Clockwork reduces the number of memory tiles us
- Without this optimization, we do not have the hardv application

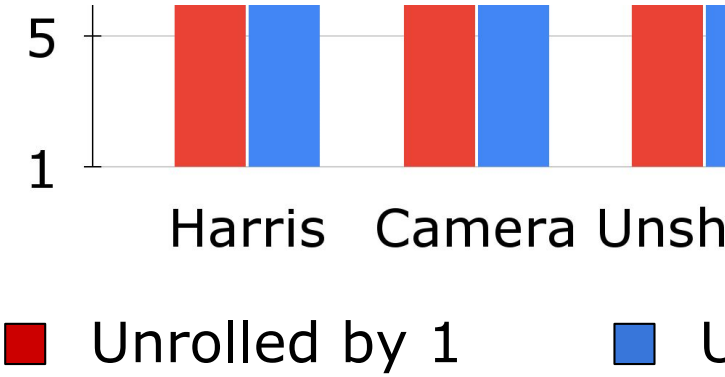




Unrolling

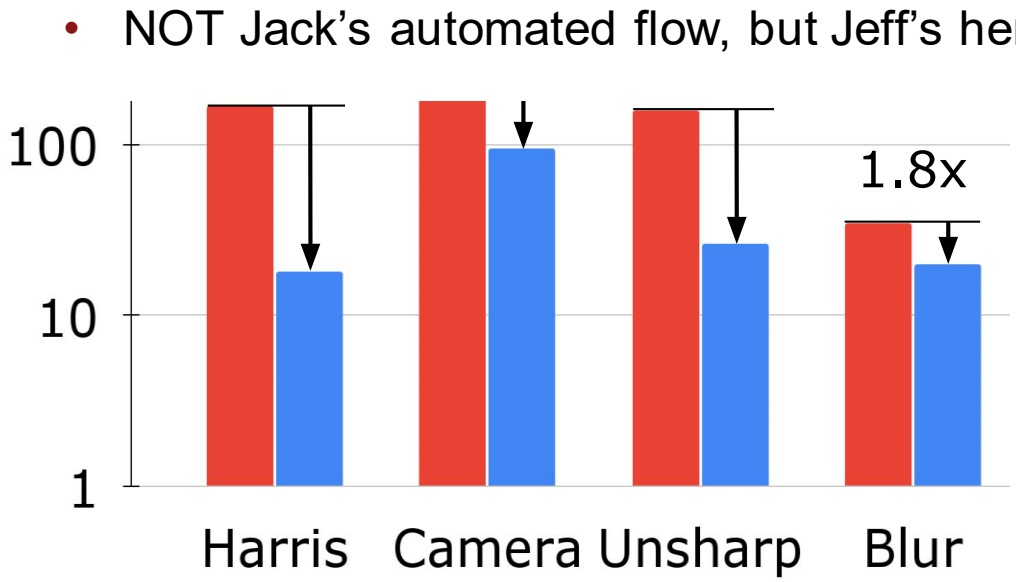
- Maximum unrolling bounded by compute or available GLB tiles
- Harris-2, Camera-1, Unsharp-3, Blur-14

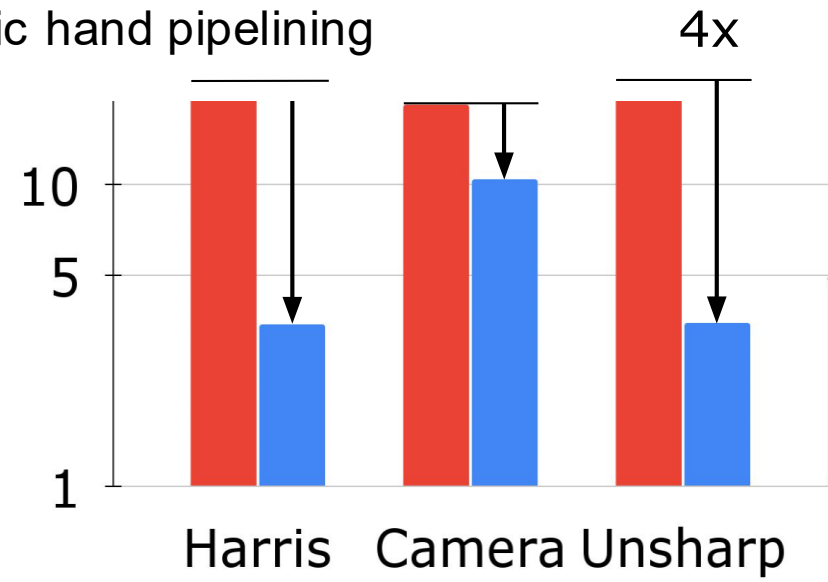




Pipelining

- NOT Jack's automated flow, but Jeff's heroic hand pipelining

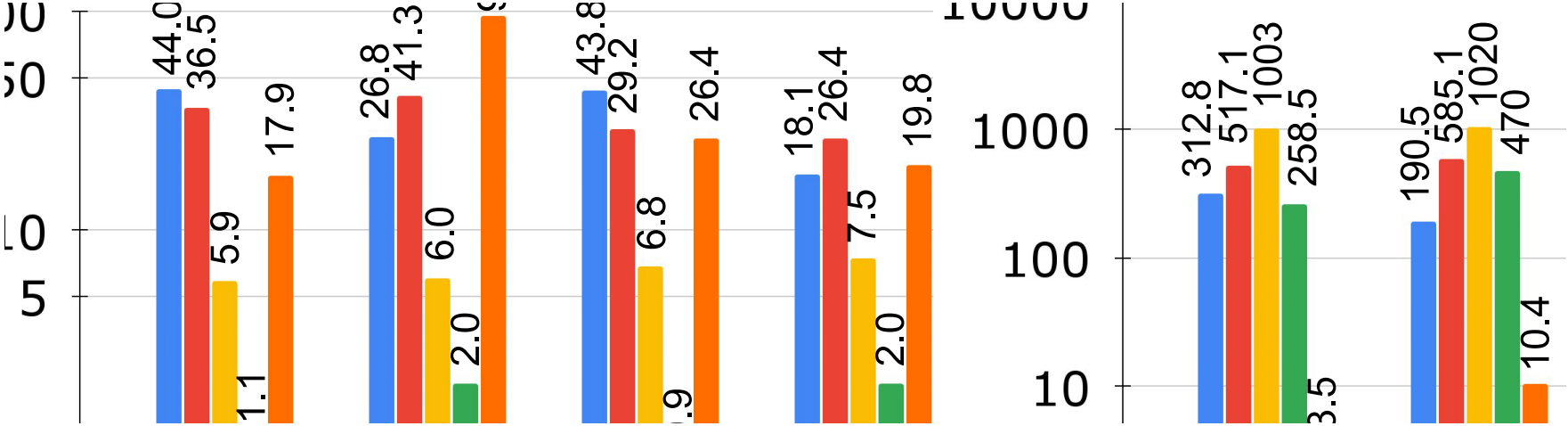




Fully unrolled but unpipelined

Full

Comparison with Industry



- Even with a 7x improvement in pipelining, our runtime will not beat industry hardware for applications like camera pipeline
- Need to look at the application graphs we are generating
- If apps like unsharp and camera pipeline cannot be unrolled very much, why do we have such a large global buffer?

Improve Amber Demonstration

Improve Current Results

- Continue fixing pipelining
- Verify measurements with new board
- Utilize TLX to run an end to end demo on large tiles, several images

Add to our Suite of Applications

- Add Machine Learning applications
- Showcase virtualization (design a nice IP/CV+ML story)

Improvements for Onyx

- Fix Bugs and Necessary Optimizations
  - Hold violation in GLB
  - Pond Enhancements
  - GLB input pattern support
- Debug enhancements
  - Read out of Memory tiles over AXI
- Rethink Design Decisions
  - Do we need 128 MEM tiles?
  - However, next PEs will have both Multipy-Add

Application	Blur	Unsharp	Camera	Harris
Target Output Rate (pixels/cycle)	14	9	3	2
Temporal Occupancy	72%	83%	73%	83%
Frequency	140 MHz	60 MHz	70 MHz	130 MHz
# PE / 384	266	303	294	206
# MEM / 128	14	36	34	17
# GLB / 16	14	9	3	6
# 1-bit Routing Tracks / 10240	69	296	417	226
# 16-bit Routing Tracks / 10240	1743	1892	1410	791