



Facebook Technology Transfer

July 21, 2022

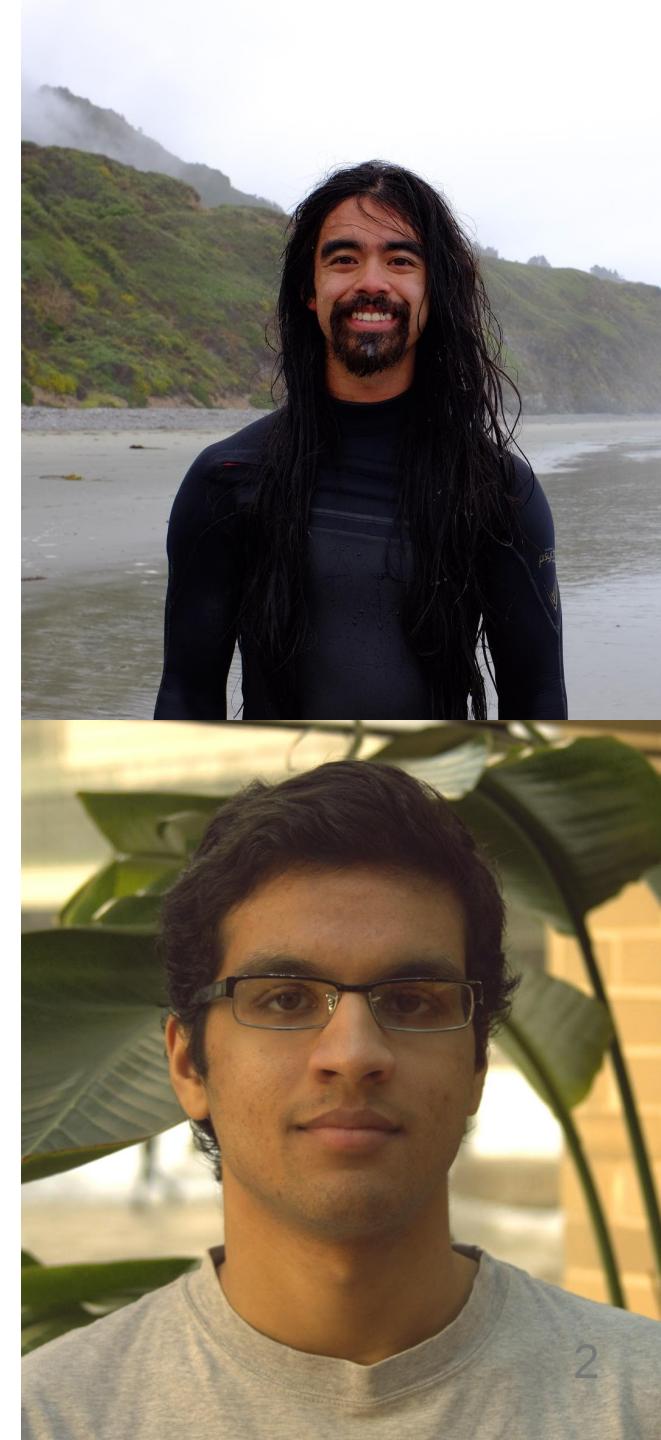
AHA DARPA QPR

lenny@stanford.edu



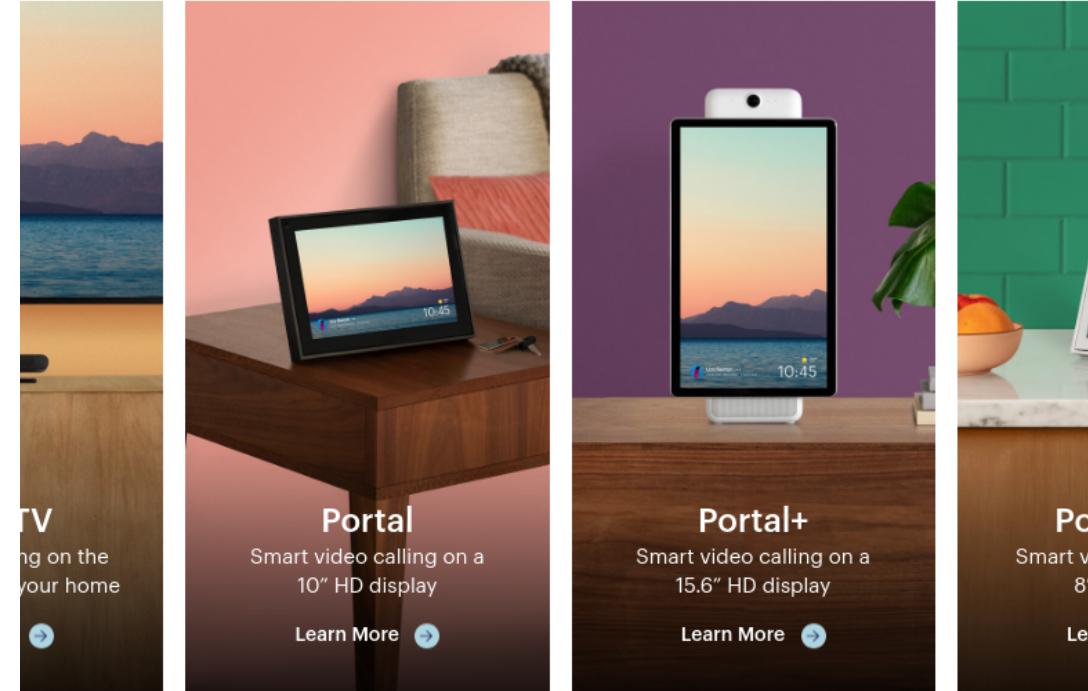
Progress Overview

- Three years and counting of two Stanford PHD students working with FB engineering team (Lenny and Raj)
 - Bug fixes, new features, infrastructure improvement, library development, documentation
 - Now under contract as a company (Silicate Systems)
- Active work on a custom accelerator for a production SoC
 - Sophisticated generators without loss of quality
- New effort to use magma at SoC architecture level
 - Smart components methodology



Hardware at Facebook

- Portal - video calling devices
(8"-15" displays + TV integration)
- AR/VR - Facebook Reality Labs (Oculus)
- Small form factors (mobile), video processing, machine learning
- Major infrastructure investment for modern software methodologies
 - Believes in open source tools



Tech Transfer Requires the Right People

Opportunity initiated by Ofer Shacham (Stanford PhD advised by Mark)

- Sr. Director of Silicon, Head of Custom SoC Compute Platforms
- Previously Head of Pixel Visual Core (Google)

Engineering team lead by Harshit Khaitan

- Technical Lead / Manager, AI Accelerators
- Previously lead Google's team building the Edge TPU with Chisel

Recipe for Success

- Senior manager willing to risk a team/project using a new tool,
- Technical manager willing to lead engineers through execution



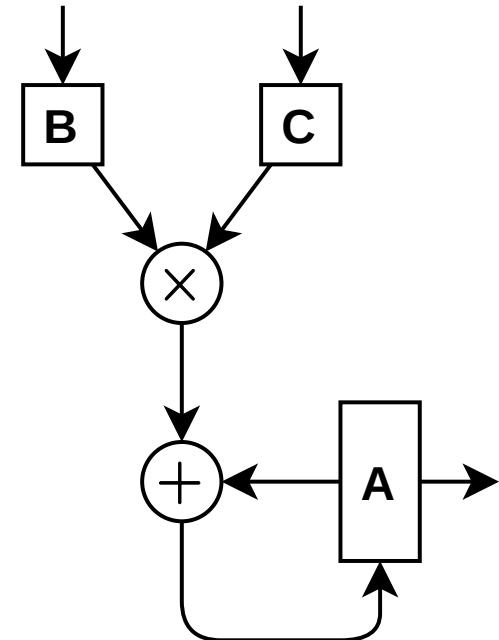
Custom Accelerator

- Block viewed as a *library component* to be reused
 - Continuous development benefits future projects
- Highly parameterized to support diverse use cases
 - E.g., desktop, standalone headset, smartphone
- Development has contributed to open source
 - Standard library, bug fixes, new features, ...



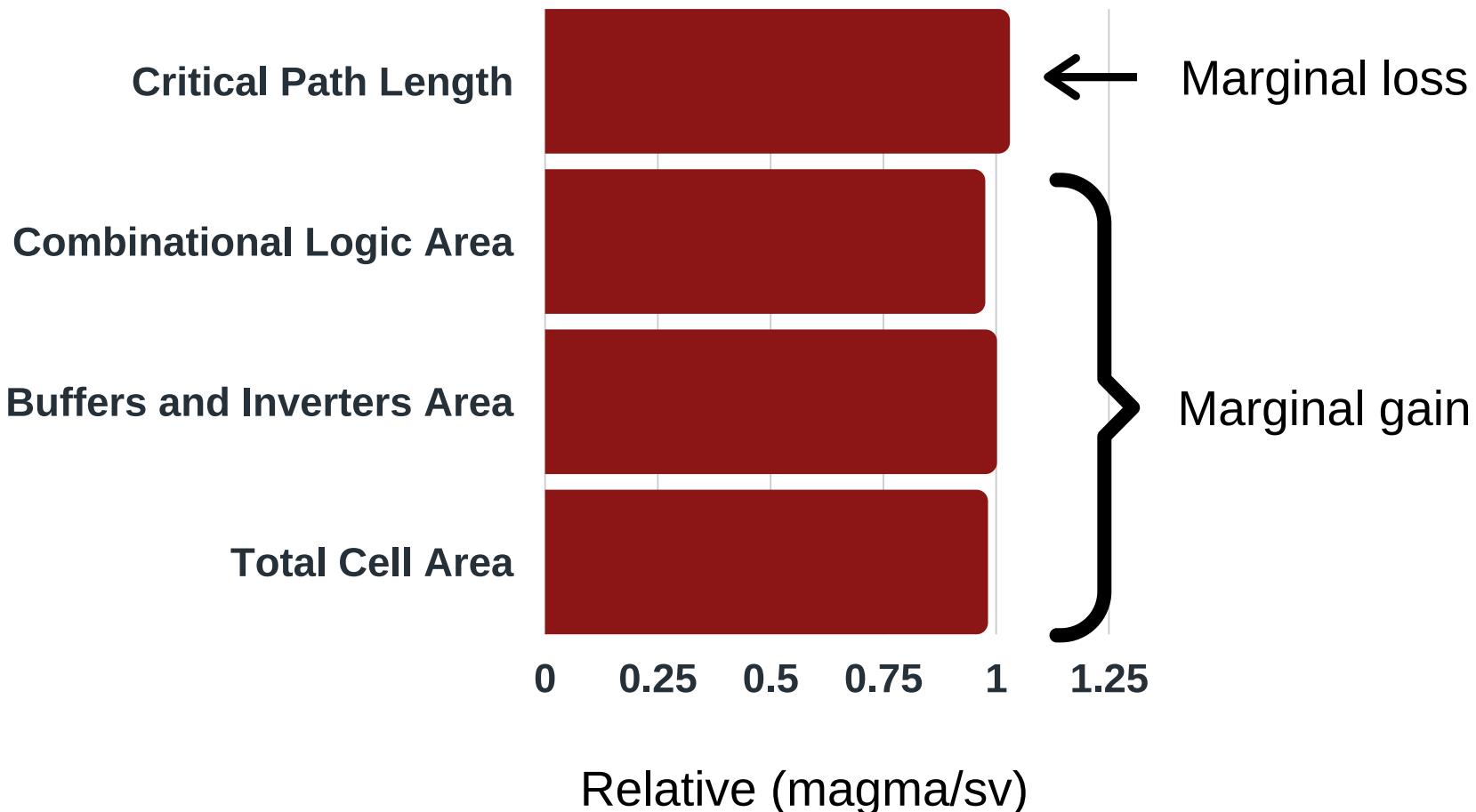
Evaluating Magma's Datapath Performance at FB

- Translated performance critical MAC unit from SV to magma
 - Direct comparison of standalone block
 - Broad comparison of whole design integration (~1k MACs)
- Magma is competitive and in some cases outperforms SV
 - No significant degradation in area and timing quality
 - Similar leakage power results (distribution of Vt cells)
 - Small increase in tool spin (5%, 24:41 to 26:01)
 - Likely caused by style of compiler generated verilog



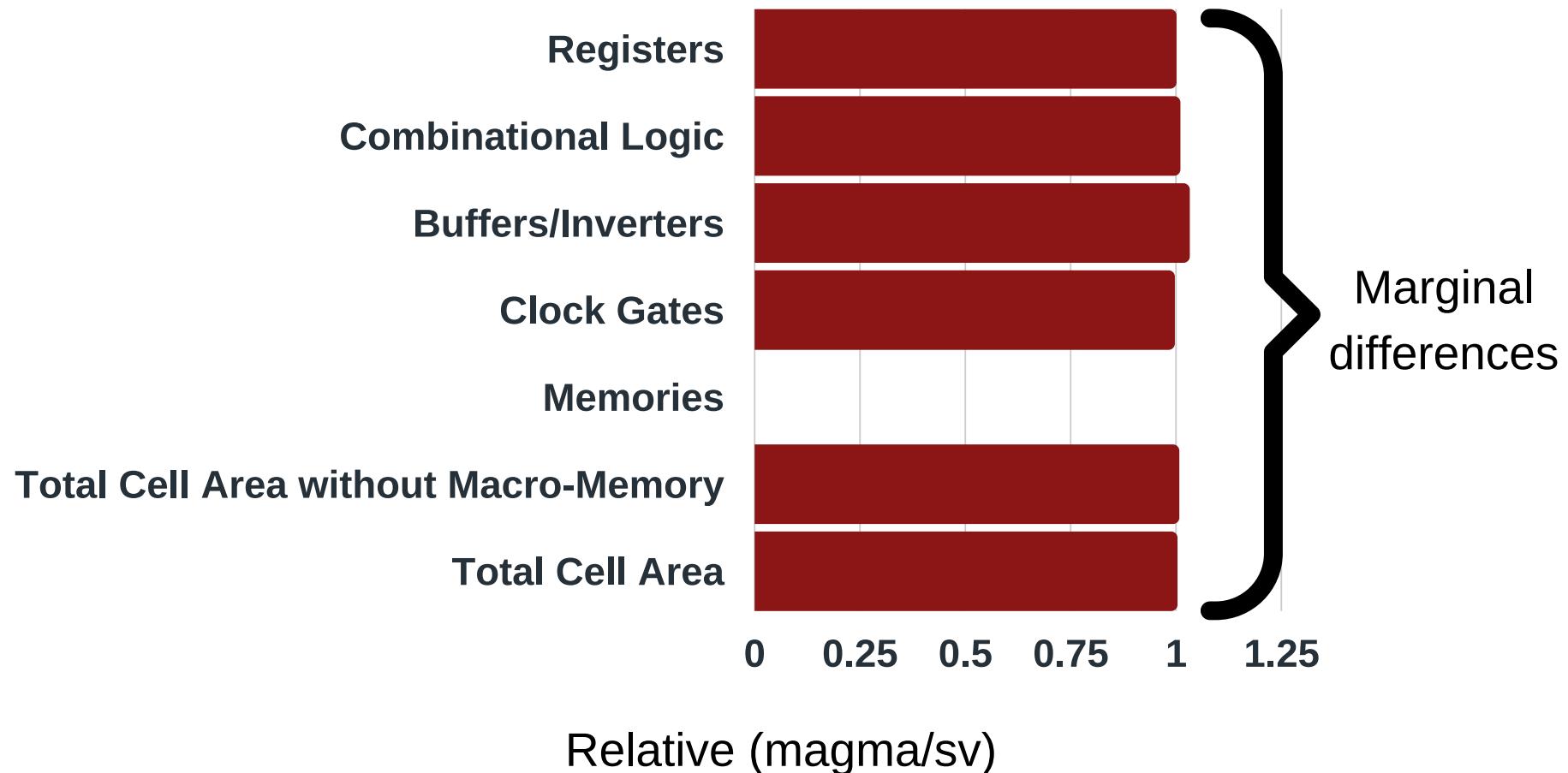
Area and Timing

Standalone MAC



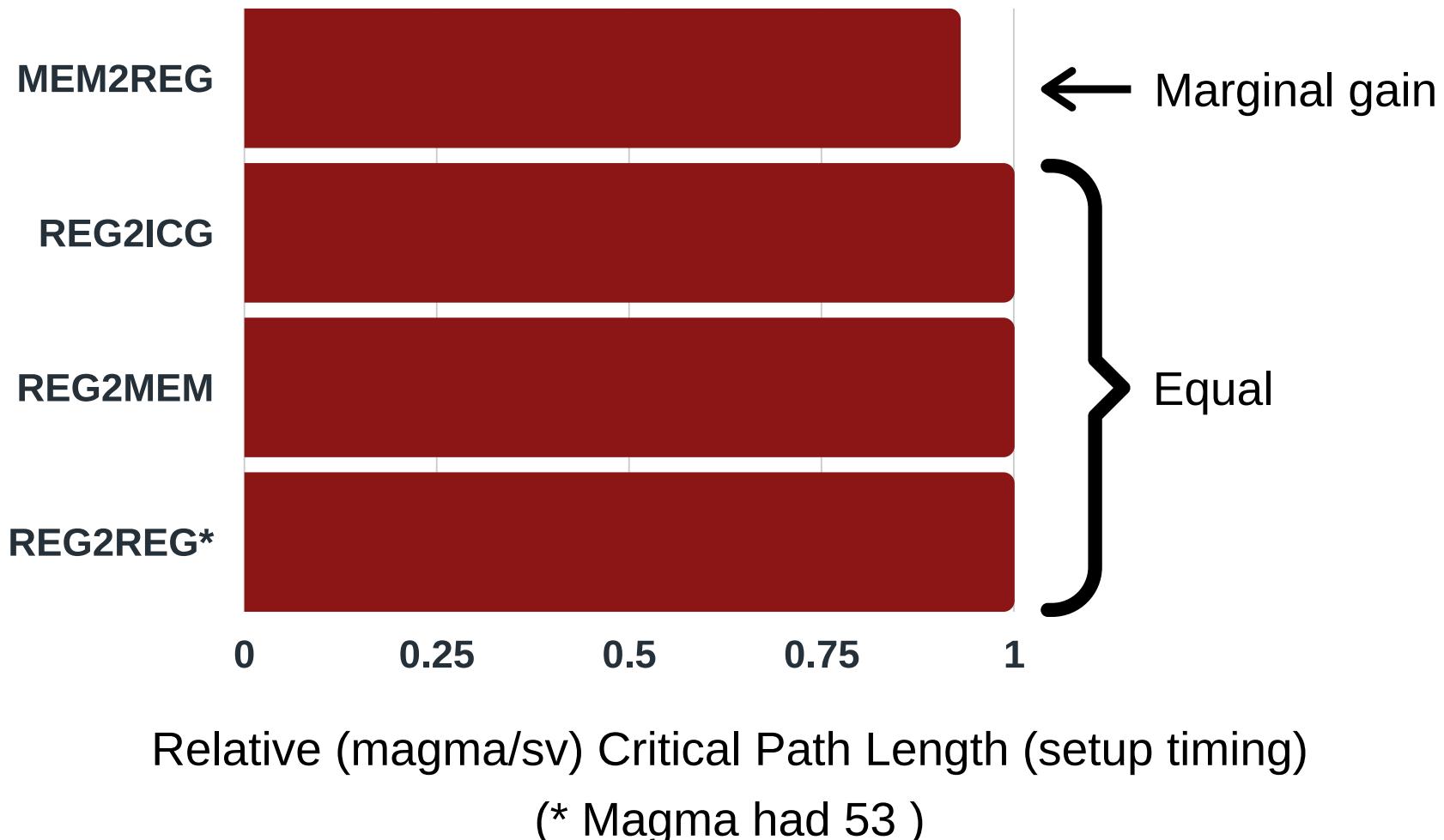
Area Breakdown

Whole Design



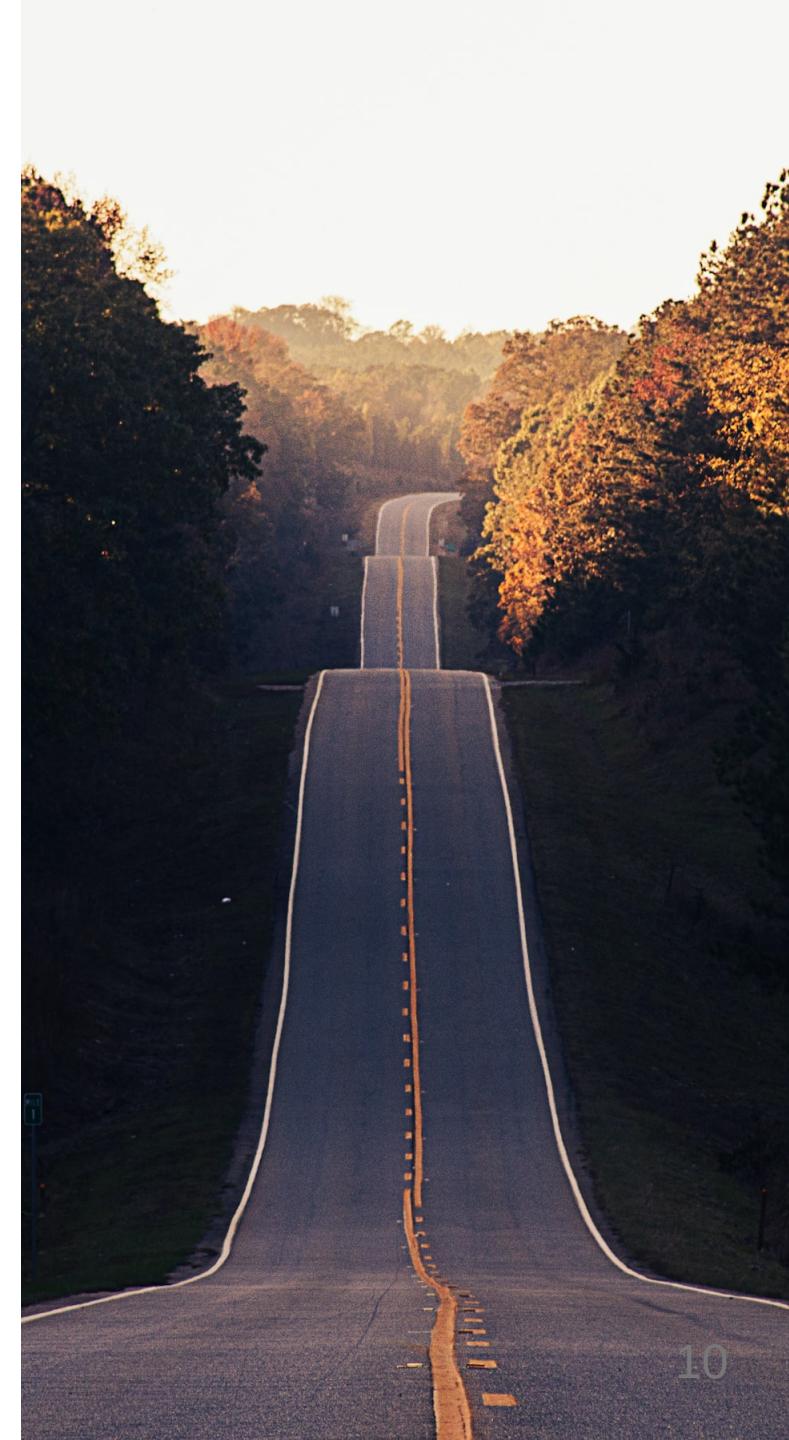
Timing Breakdown

Whole Design



Roadmap

- Continued support for FB engineering efforts
 - New contract for 6 months (with plans to extend)
- **Broaden scope of magma's use at Facebook**
 - Targeting SoC assembly level
- Engage more companies
 - Share contributions to open-source
- Engage universities
 - Develop learning material to increase talent pool
 - Improve experience for researchers

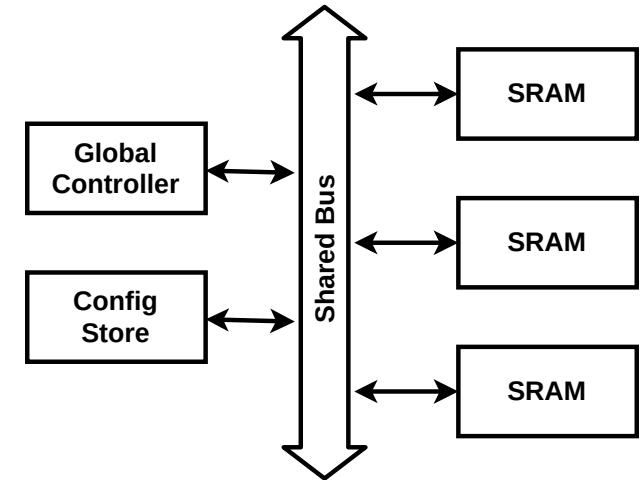


Using Magma at FB for SoCs

Presenting *Smart Components* to Ofer on Aug 6th

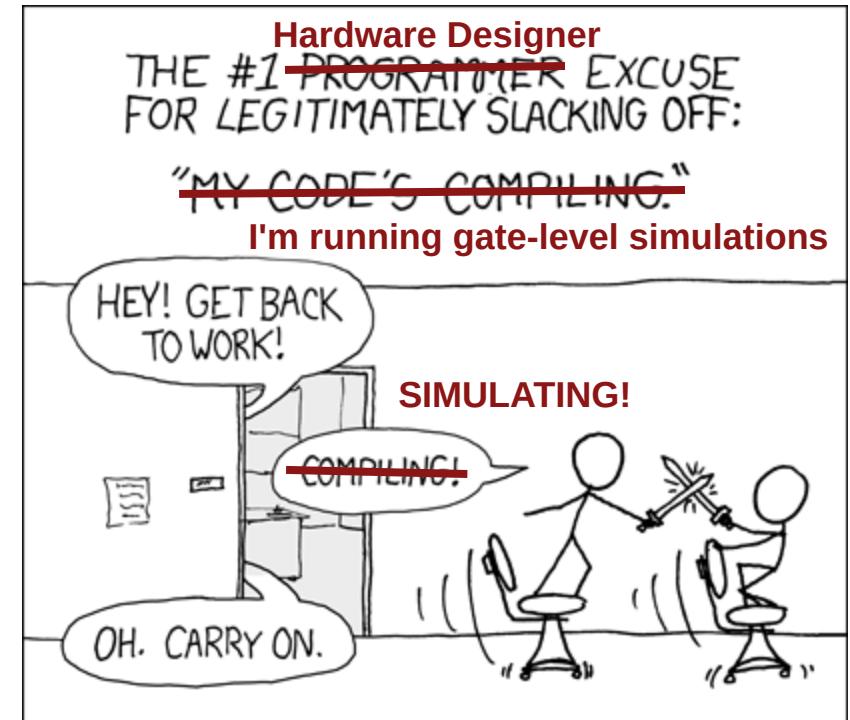
- Methodology targets major complaint from Ofer
- Avoids late stage bugs using advanced static typing techniques (session types)
- Adds new technology to Magma's type system and verification infrastructure

Goal: Convince an engineering team lead to drive a new Magma project (SoC level)



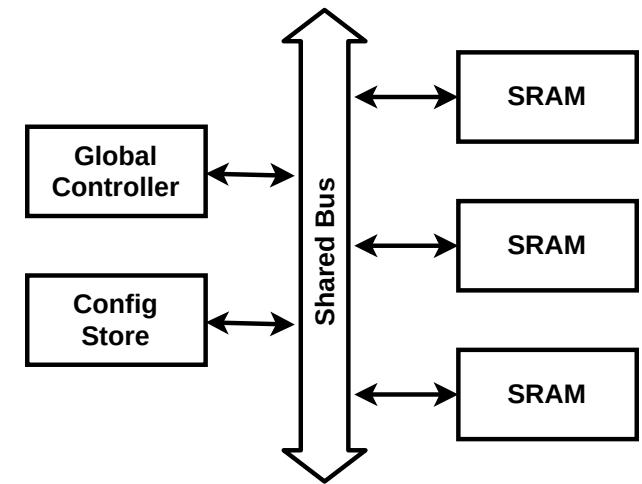
Avoiding Late Stage Design Bugs at Facebook

- Major complaint from Ofer about dependence on gate-level simulations
 - Used to find logical problems
 - **Can take a week to run**
- **Goal:** Gate level sims are an extra check
- **Approach:** Surface these logical problems earlier in the design process



Errors in Gate Level Sims?

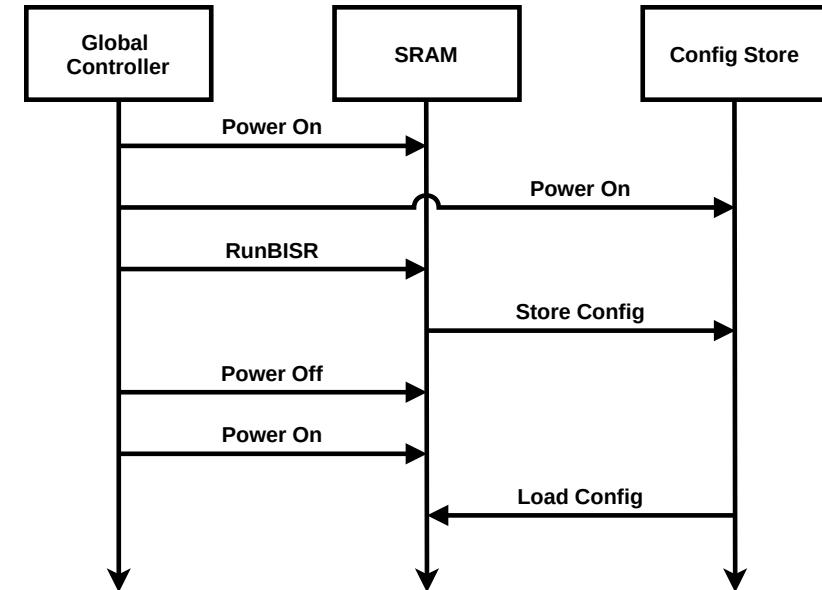
- Logic of the final netlist may not match the RTL
 - Some logical details added late in the process
 - **Example:** Power domains added after synthesis
- Final logic may have unexpected interactions
 - **Example:** SRAM with power domains must manage redundancy config using test interface
- **Summary:** Issues related to redundancy, power, and testing appear after mapping to physical instances



Smart Components

Methodology: Use abstract interfaces to surface the component interactions earlier

- Abstraction avoids technology details
- Resolve bugs at RTL rather than gate-level
- IP blocks lower actions to state changes
 - Reuse interface for different IP blocks
- **Examples:** power, boot, restart, configure, test



Magma as a Platform for Smart Components

New Technology: Extend the Magma type system with session types

- Specify smart component interfaces as a type (sequence of abstract actions)

```
SRAMInit = Sequence[
    Require[HasPowerState(ConfigStore, ON)],
    (GC > SRAM, RunBISR()),
    (SRAM > ConfigStore, StoreRedundancyConfig(id)),
]
```

- Type checking ensures components interact in a correct (type-safe) manner
- Types are convertible to FSMs for unit testing, assertions, and model checking

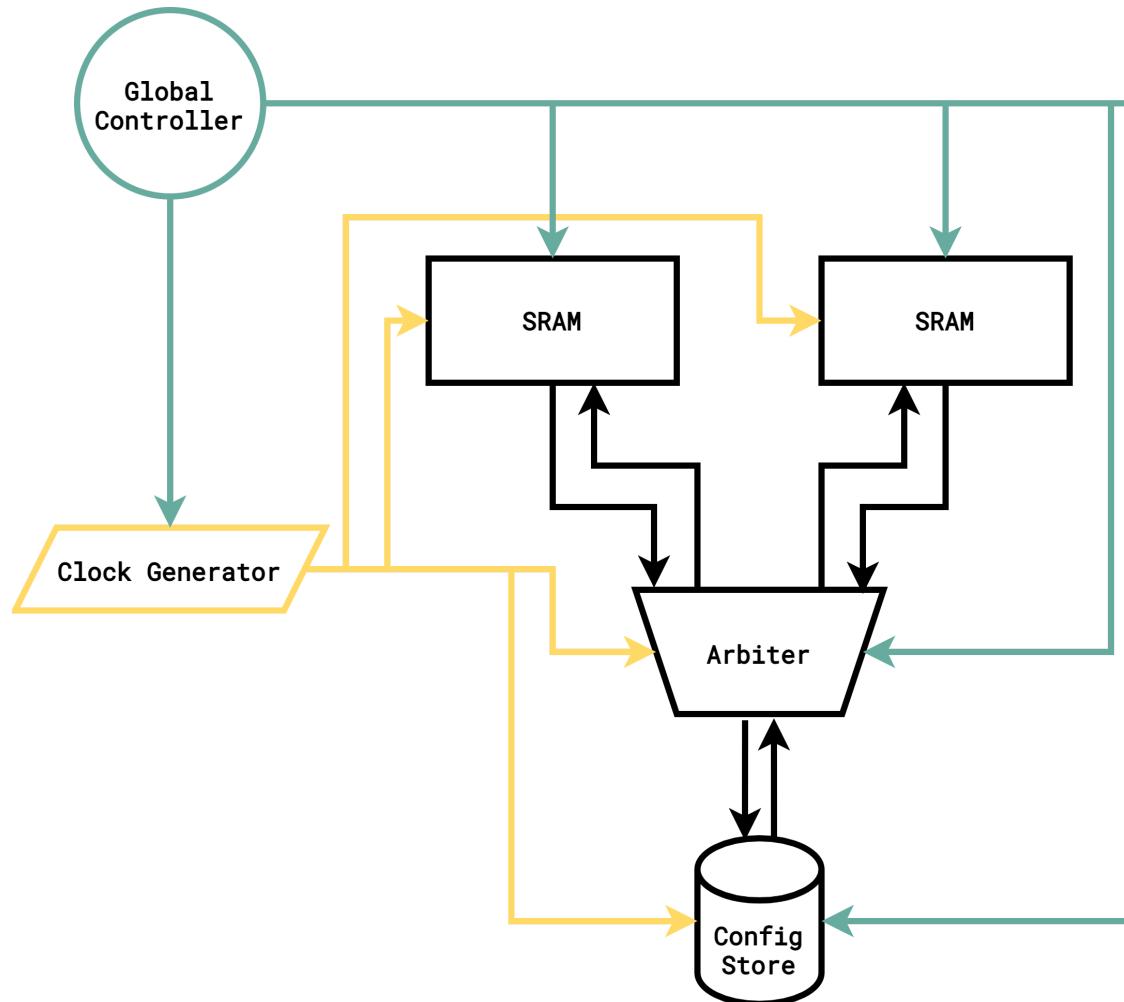
Design Requirement: Integrate with existing components (e.g. FB internal IP blocks) using Magma's Verilog interoperability to lower cost of adoption

Smart Components Intellectual Challenges

- Avoiding global specification (promote modularity)
- Handling parallelism and concurrency (interrupt events)
- Compiling abstract interfaces (encoding as ports, bus interfaces)
- Verifying implementations (static types, simulation, model checking)



SRAM Example



Avoiding Global Specification

Goal: Describe the required interactions from the local perspective of each component

- Avoid complexity of maintaining a global sequence (e.g. introducing a new module)

```
# Run BISR and store configuration
SRAMInit = Sequence[
    # Assertion on state of another component
    Require[HasPowerState(ConfigStore, ON)],
    # Run self repair
    (GC > SRAM, RunBISR()),
    # Store redundancy config bits
    (SRAM > ConfigStore, StoreRedundancyConfig(id)),
]
```

Handling Parallelism and Concurrency

Goal: Allow `ClockGenerator` to be turned off any time (clock gating)

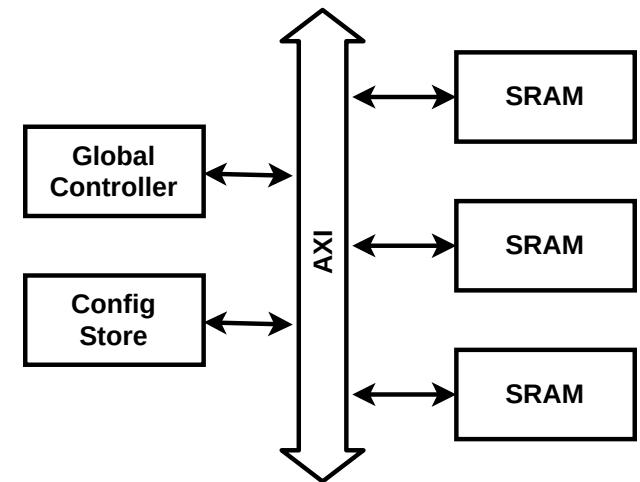
```
SRAM = Sequence[
    (GC > ClockGenerator, SetPowerState(ON)), SRAMInit, SRAMRun
]
SRAM.on_interrupt(
    Event[(GC > ClockGenerator, SetPowerState(OFF))],
    lambda curr_seq: Sequence[
        WaitUntil[(GC > ClockGenerator, SetPowerState(ON))],
        curr_seq
    ]
)
```

Approach: Use the concept of an *interrupts* to describe event-based logic

- Draws off exception theory from software

Compiling Abstract Interfaces

- Abstract interfaces can use ready/valid interfaces
 - Tag bits used to encode branch label selection
 - Reuse port for sending/receiving data and tags
- **Challenge:** Support mapping to busses or NoCs
 - Likely required for adoption in a production SoC



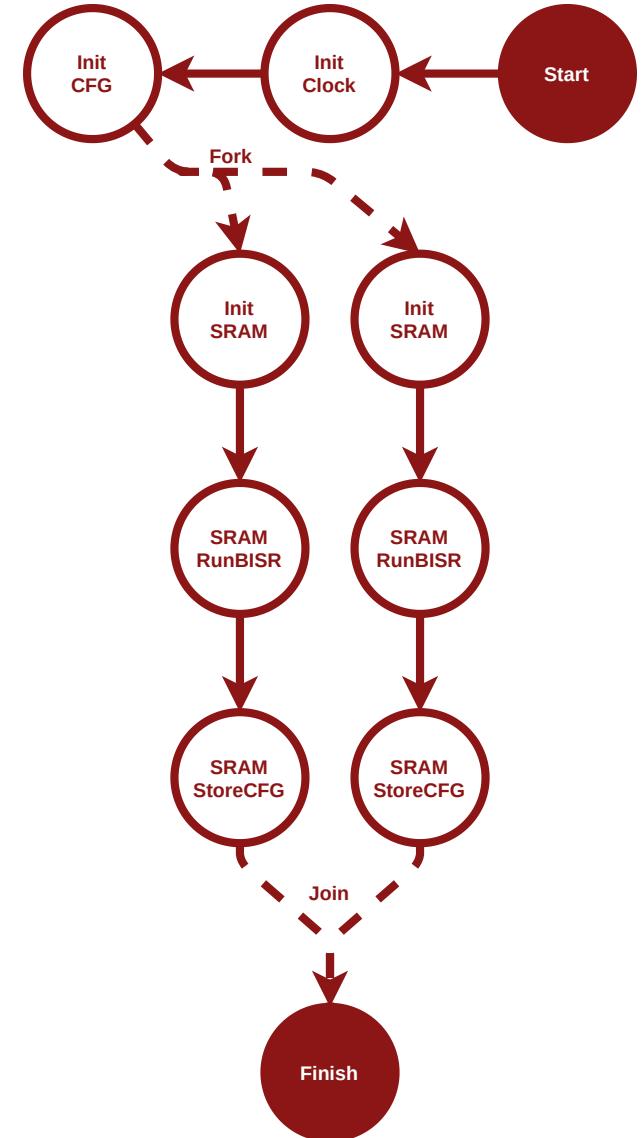
Verifying Implementations

Compile session types to an FSM

- Verify arbitrary components (e.g. Verilog IP)
 - Unit tests to check state transitions
 - Assertions for simulation and model checking

Statically type check magma coroutines

- Use traditional syntax checks of imperative FSMs
- Promotes adoption of more magma technology!



Magma: Tech Transfer Platform

Magma's adoption provides an opportunity to introduce our higher-level research technologies

- New technology broadens Magma's utility
- Existing infrastructure and engineering talent
 - Software available on development server
 - Knowledgeable developers and documentation

Magma provides a pathway for new research



Conclusion

- Major success with magma adoption at Facebook
 - Designed and verified a major block (custom accelerator) in a production SoC
 - Block is being actively developed and team desires continued support
 - Three years of productive engagement with a recently renewed contract
- Leveraging magma as a platform for introducing new research technology
 - Engaging with FB SoC team to introduce smart components methodology
- Expanding our technology transfer efforts to other companies and universities

