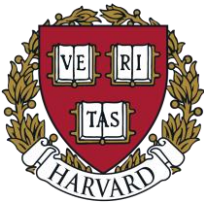


# Architecture 2.0: Prompt Engineering and Foundation Models for Edge AI Hardware Design

---

Vijay Janapa Reddi  
Harvard University



# The Computer for the 21st Century

*Specialized elements of hardware and software, connected by wires, radio waves and infrared, will be so ubiquitous that no one will notice their presence*

by Mark Weiser

The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it.

Consider writing, perhaps the first information technology. The ability to represent spoken language symbolically for long-term storage freed information from the limits of individual memory. Today this technology is ubiquitous in industrialized countries. Not only do books, magazines and newspapers convey written information, but so do street signs, billboards, shop signs and even graffiti. Candy wrappers are covered in writing. The constant background presence of these products of "literacy technology" does not require active attention, but the information to be transmitted is ready for use at a glance. It is difficult to imagine modern life otherwise.

Silicon-based information technology, in contrast, is far from having become part of the environment. More than 50 million personal computers have been sold, and the computer nonetheless remains largely in a world of its own. It

is approachable only through complex jargon that has nothing to do with the tasks for which people use computers. The state of the art is perhaps analogous to the period when scribes had to know as much about making ink or baking clay as they did about writing.

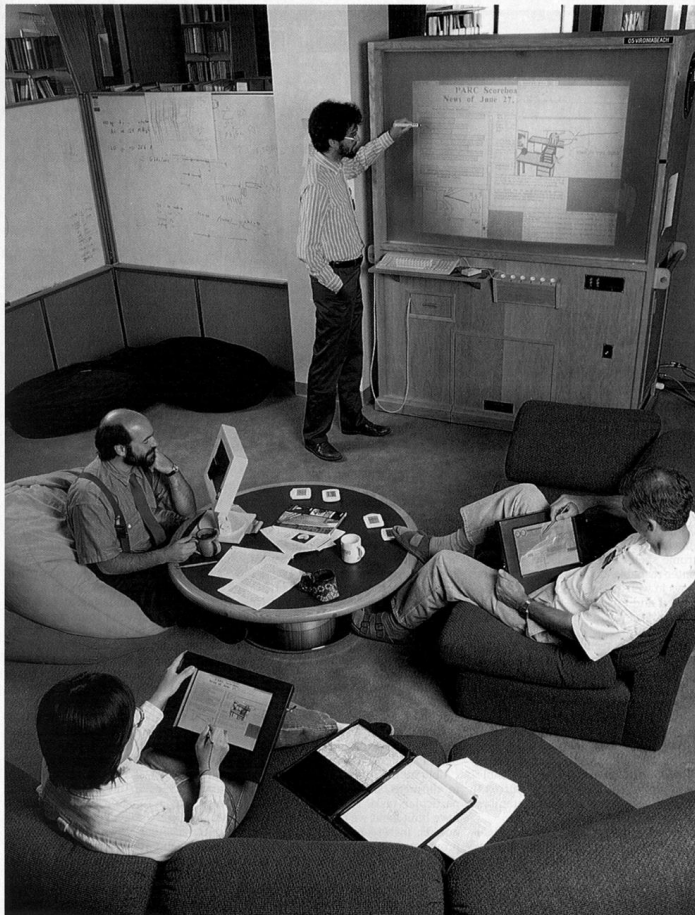
The arcane aura that surrounds personal computers is not just a "user interface" problem. My colleagues and I at the Xerox Palo Alto Research Center think that the idea of a "personal" computer itself is misplaced and that the vision of laptop machines, dynabooks and "knowledge navigators" is only a transitional step toward achieving the real potential of information technology. Such machines cannot truly make computing an integral, invisible part of people's lives. We are therefore trying to conceive a new way of thinking about computers, one that takes into account the human world and allows the computers themselves to vanish into the background.

Such a disappearance is a fundamental consequence not of technology but of human psychology. Whenever people learn something sufficiently well, they cease to be aware of it. When you look at a street sign, for example, you absorb its information without consciously performing the act of reading. Computer scientist, economist and Nobelist Herbert A. Simon calls this phenomenon "compiling"; philosopher Michael Polanyi calls it the "tacit dimension"; psychologist J. J. Gibson calls it "visual invariants"; philosophers Hans Georg Gadamer and Martin Heidegger call it the "horizon" and the "ready-to-hand"; John Seely Brown of PARC calls it the "periphery." All say, in essence, that only when

The idea of integrating computers seamlessly into the world at large runs counter to a number of present-day trends. "Ubiquitous computing" in this context does not mean just computers that can be carried to the beach, jungle or airport. Even the most powerful notebook computer, with access to a worldwide information network, still focuses attention on a single box. By analogy with writing, carrying a superlaptop is like owning just one very important book. Customizing this book, even writing millions of other books, does not begin to capture the real power of literacy.

Furthermore, although ubiquitous computers may use sound and video in addition to text and graphics, that does not make them "multimedia computers." Today's multimedia machine makes the computer screen into a demanding focus of attention rather than allowing it to fade into the background.

Perhaps most diametrically opposed to our vision is the notion of virtual reality, which attempts to make a world inside the computer. Users don special goggles that project an artificial scene onto their eyes; they wear gloves or even bodysuits that sense their motions and gestures so that they can move about and manipulate virtual objects. Although it may have its purpose in allowing people to explore realms otherwise inaccessible—the insides of cells, the surfaces of distant planets, the information web of data bases—virtual reality is only a map, not a territory. It excludes desks, offices, other people not wearing goggles and bodysuits, weather, trees, walks, chance encounters and, in general, the infinite richness of the universe. Virtual reality focuses an enormous apparatus on simu-



UBIQUITOUS COMPUTING begins to emerge in the form of live... and integrating them with other tools has helped researchers

*Specialized elements of hardware and software, connected by wires, radio waves and infrared, will be so ubiquitous that no one will notice their presence*


*Specialized elements of hardware and software, connected by wires, radio waves and infrared, will be so ubiquitous that no one will notice their presence*

tinyML




but...

# The Cost of Building Chips is Growing



**SEMICONDUCTOR ENGINEERING**  
DEEP INSIGHTS FOR THE TECH INDUSTRY





Chinese (Simplified) English **SUBSCRIBE** 

HOME SYSTEMS & DESIGN LOW POWER - HIGH PERFORMANCE MANUFACTURING, PACKAGING & MATERIALS TEST, MEASUREMENT & ANALYTICS AUTO, SECURITY & PERVASIVE COMPUTING


SPECIAL REPORTS BUSINESS & STARTUPS JOBS KNOWLEDGE CENTER TECHNICAL PAPERS ▾ EVENTS & WEBINARS ▾ VIDEOS & RESEARCH ▾ NEWSLETTERS & STORE ▾

**SYSTEMS & DESIGN**

## What Will That Chip Cost?

826 Shares  41  150  612 

*Establishing the true cost to develop an advanced chip is complicated, but headline numbers appear to be significantly inflated.*

OCTOBER 30TH, 2023 - BY: BRIAN BAILEY 

In the past, analysts, consultants, and many other experts attempted to estimate the cost of a new chip implemented in the latest process technology. They concluded that by the 3nm node, only a few companies would be able to afford them — and by the time they got into the angstrom range, probably nobody would.

Much has changed over the past few process nodes. Increasing numbers of startups are successfully building advanced-node chips for much less money than those highly quoted figures. Behind the numbers are some broad-based changes in chip design and manufacturing. Among them:

TECHNICAL PAPERS

**Verifying Hardware CWEs In RTL Designs Generated By GenAI**

MARCH 27, 2024 BY TECHNICAL PAPER  
LINK

**DRAM Cache For GPUs With SCM And High Bandwidth**











MARCH 26, 2024 BY TECHNICAL PAPER  
LINK

**Superconducting Qubits Made Using Industry-Standard, Advanced Semiconductor Manufacturing (Imec, KU Leuven)**

MARCH 25, 2024 BY TECHNICAL PAPER  
LINK

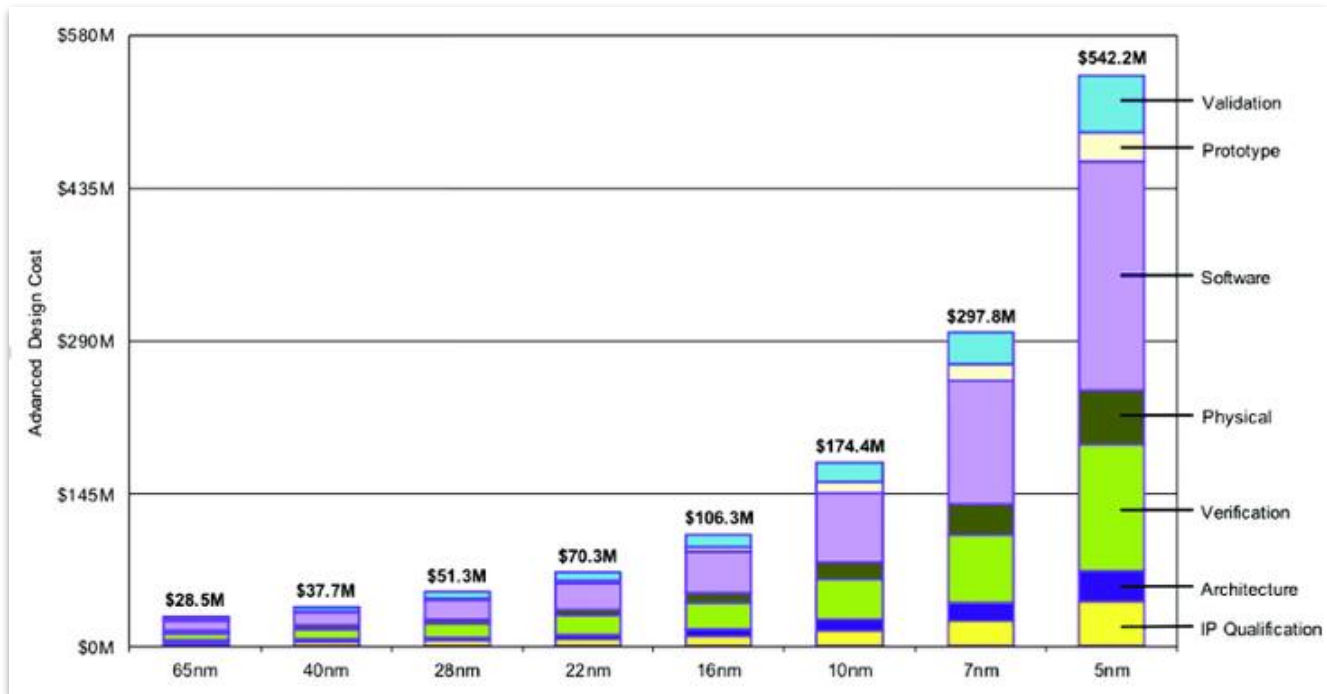
**Designing AI Hardware To Deal With Increasingly Challenging Memory Wall (UC Berkeley)**

SPONSORS



NEWSLETTER SIGNUP

Email \*



Source: IBS 2018



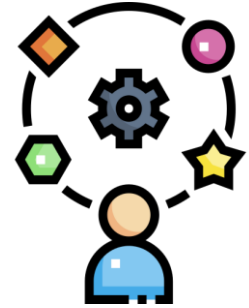
Process



Architecture



Optimization



Specialization

# Embedded Computer Architecture and Automation



**The distinct requirements of embedded computing, coupled with emerging technologies, will stimulate system and processor specialization, customization, and computer architecture automation.**

B. Ramakrishna  
Rau  
Michael S.  
Schlanser  
Hewlett-Packard  
Laboratories

**W**ith the advent of system level integration (SLI)—the next level of integration beyond VLSI—and system-on-chip (SOC) capabilities, the computer industry's focus is shifting from personal to embedded computing. The opportunities, needs, and constraints of this emerging trend will lead to significantly different computer architectures at both the system and processor levels as well as a rich diversity of off-the-shelf (OTS) and custom designs. Embedded computing will also stimulate automation of computer architecture, which we illustrate using an architecture synthesis system called PICO—program in, chip out—that we developed with our colleagues during the past five years.

## EMBEDDED COMPUTING

Driven by the accelerated pace of semiconductor integration during the past three decades, the computer industry has steadily moved from mainframes and minicomputers to workstations and PCs. In accordance with a corollary of Moore's law, computing power becomes half as expensive every 18 to 24 months. Over a decade, this reduces the cost by a factor of 30 to 100, making computing affordable to an exponentially larger number of users and dramatically changing the key applications of this computing power.

Manufacturers have for several years incorporated embedded computers in so-called smart products such as video games, DVD players, televisions, printers, scanners, cellular phones, and robotic vacuum cleaners. Using embedded computers in devices that previously relied on analog circuitry—such as digital cameras, digital camcorders, digital personal re-

orders, Internet radios, and Internet telephones—provides revolutionary performance and functionality that merely improving analog designs could not achieve. The increasing availability of SLI heralds a vast array of even more innovative smart products.

Any computer architecture must balance the latest technological opportunities with product, market, and application requirements that together determine three important features of embedded computing architecture: specialization, customization, and automation. Specialization increases the performance and reduces the manufacturing cost of embedded computer systems. Customization permits specialization when no adequately specialized OTS product is available. Automation reduces the design costs incurred by customization.

## Product requirements

Smart products demand various combinations of high performance, low cost, and low power. When budgets allow only a few cents for critical chips, cost can be more important than performance. In other cases, the smart product's functionality mandates high performance. For example, data-intensive tasks such as image, video, and signal processing require throughput that significantly exceeds that of high-end workstations.

The challenge now is to lower cost to a level that the market will accept. In certain instances, providing more computing power can reduce cost. Many imaging and video products require large amounts of DRAM to hold their data sets. Using compression techniques such as MPEG, JPEG, and MP3 for large video, image, and audio data sets, respectively, reduces the amount of DRAM required but introduces a sig-

# 1991

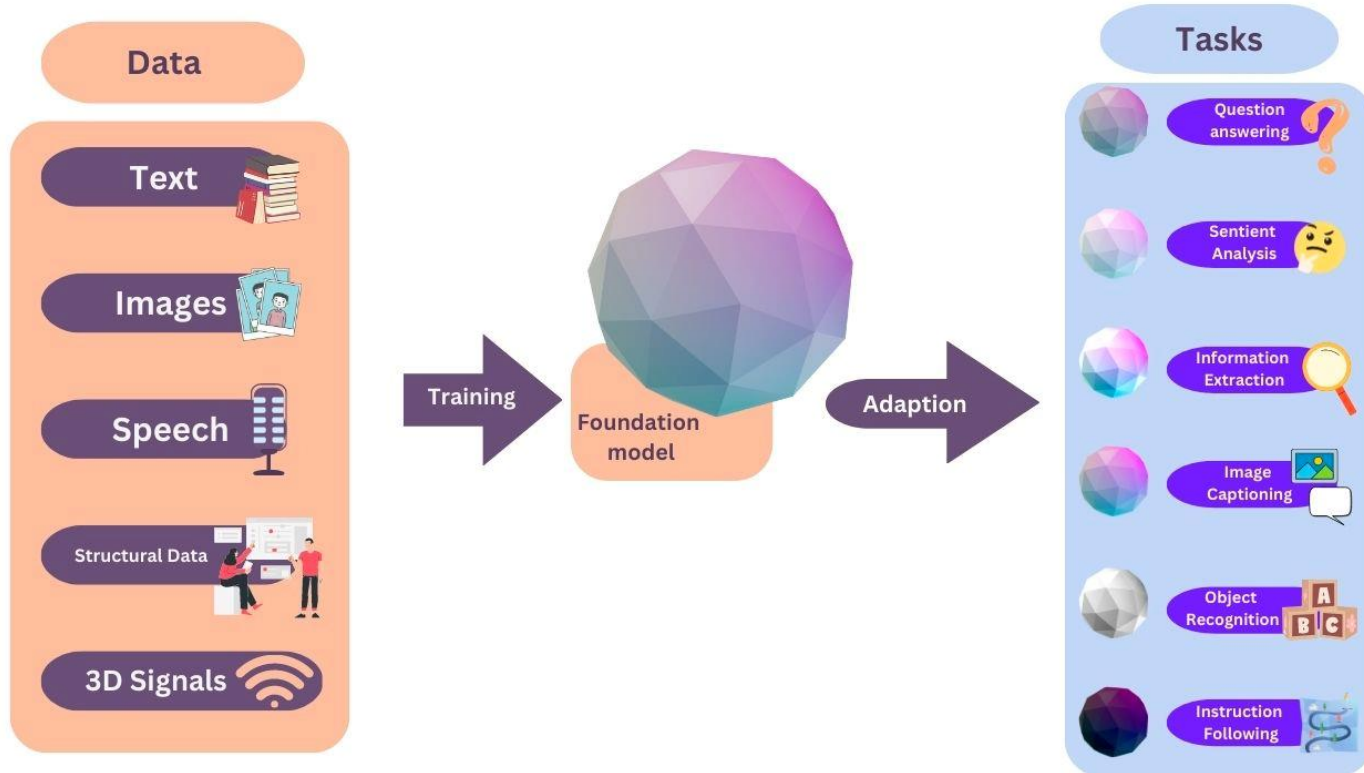


- **Embedded Computing's Drive for Low Costs:** As computing power becomes more affordable, the challenge is to lower costs to a level acceptable by the market. In some cases, providing more computing power, through techniques like compression (MPEG, JPEG, MP3), can actually reduce overall costs by reducing the amount of DRAM needed, thus balancing the cost of extra processing power against savings in memory costs.
- **Product Requirements Emphasizing Low Cost:** Smart products require combinations of high performance, low cost, and low power. When budgets are tight, the cost of critical chips can be more significant than their performance, showcasing the necessity to prioritize cost-effectiveness in embedded computing designs.
- **Specialization for Cost Efficiency:** Specialization in embedded computing architecture enhances performance while reducing manufacturing costs. It involves creating systems or processors for specific applications or application domains, thereby minimizing logic complexity and die size, leading to lower fabrication costs.
- **Customization and Its Impact on Costs:** Customization allows for achieving better cost performance but introduces complexity in the design process, increasing non-recurring expenses (NRE). These increased NREs must be amortized over product volume, adding to the product cost.
- **Automation as a Solution to Reduce Design Costs:** The need for mass customization and the limited number of designers available have led to the automation of computer architecture. Automation significantly reduces design costs, making customized products for low volumes viable. This approach helps in managing and potentially reducing the overall costs associated with designing and producing embedded computer systems.

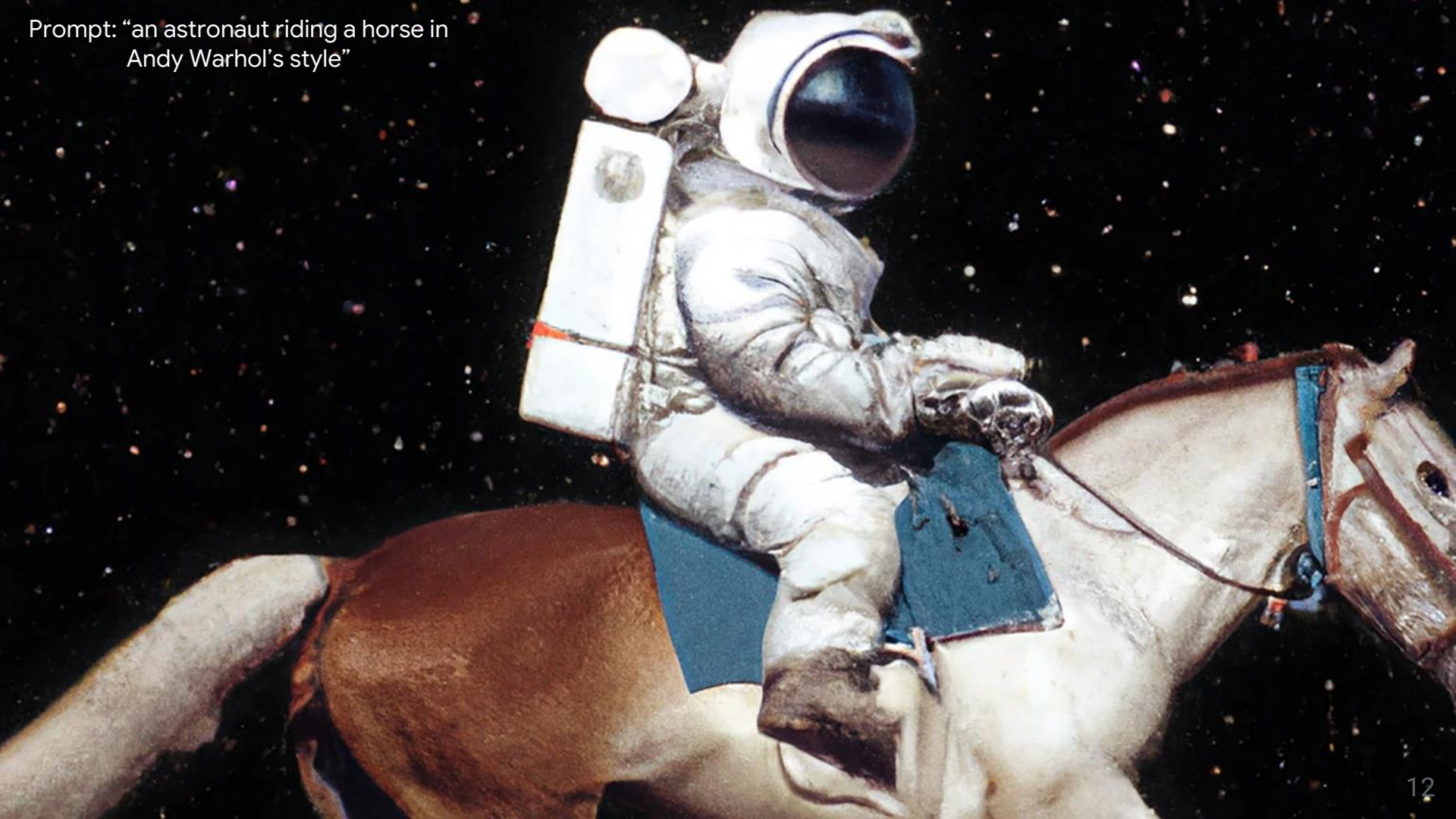




Can AI help?



Prompt: "an astronaut riding a horse in  
Andy Warhol's style"



# MusicLM

Describe a musical idea and hear it come to life with AI





Can you write code for a website that shows the current trading price of the S&P 500 index

< 2/2 >



As an AI language model, I can provide you with the code for displaying the current trading price of the S&P 500 index. However, keep in mind that the data source and API used in this code may not be up to date, as my knowledge cutoff is in 2021. Also, I am not able to test this code, so please use it at your own risk.



Here's an example of how you could display the current trading price of the S&P 500 index using JavaScript and the Alpha Vantage API:

```
html Copy code
<!DOCTYPE html>
<html>
  <head>
    <title>S&P 500 Index Price</title>
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
  </head>
  <body>
    <h1>S&P 500 Index Price</h1>
    <p>Current Price: <span id="price"></span></p>
    <script>
      $(document).ready(function() {
        $.ajax({
          url: "https://www.alphavantage.co/query",
          dataType: "json",
          data: {
            function: "GLOBAL_
            </script>
```

Regenerate response







Can you write code for a website that shows the current trading price of the S&P 500 index



Write me an architectural simulator for the most used RISC-V instructions



Sure! I can provide you with a simple architectural simulator for the most commonly used RISC-V instructions. This simulator will allow you to execute RISC-V assembly code and observe the effects of each instruction on the simulated processor's state.



Here's an example simulator implemented in Python:

python Copy code

```
# Define the initial processor state
registers = [0] * 32 # 32 general-purpose registers
memory = [0] * 1024 # 1024 bytes of memory

# Helper function to parse and execute RISC-V instructions
def execute_instruction(instruction):
    opcode = instruction & 0b1111111

    if opcode == 0b0110111: # LUI
        rd = (instruction >> 7) & 0b11111
        imm = instruction >> 12
        registers[rd] = imm << 12

    elif opcode == 0b0010111: # AUIPC
        rd = (instruction >> 7) & 0b11111
        imm = instruction >> 12
        registers[rd] = registers[rd] + (imm << 12)

    elif opcode == 0b1101111: # JAL
        rd = (instruction >> 7) & 0b11111
        imm_20 = (instruction >> 31) & 0b1
```



Can you write code for a website that shows the current trading price of the S&P 500 index



Write me an architectural simulator for the most used RISC-V instructions



Sure! I can provide you with a simple architectural simulator for the most commonly used RISC-V instructions. This simulator will allow you to execute RISC-V assembly code and observe the effects of each instruction on the simulated processor's state.



Act an RTL developer and write the verilog code for a RISC-V microcontroller unit



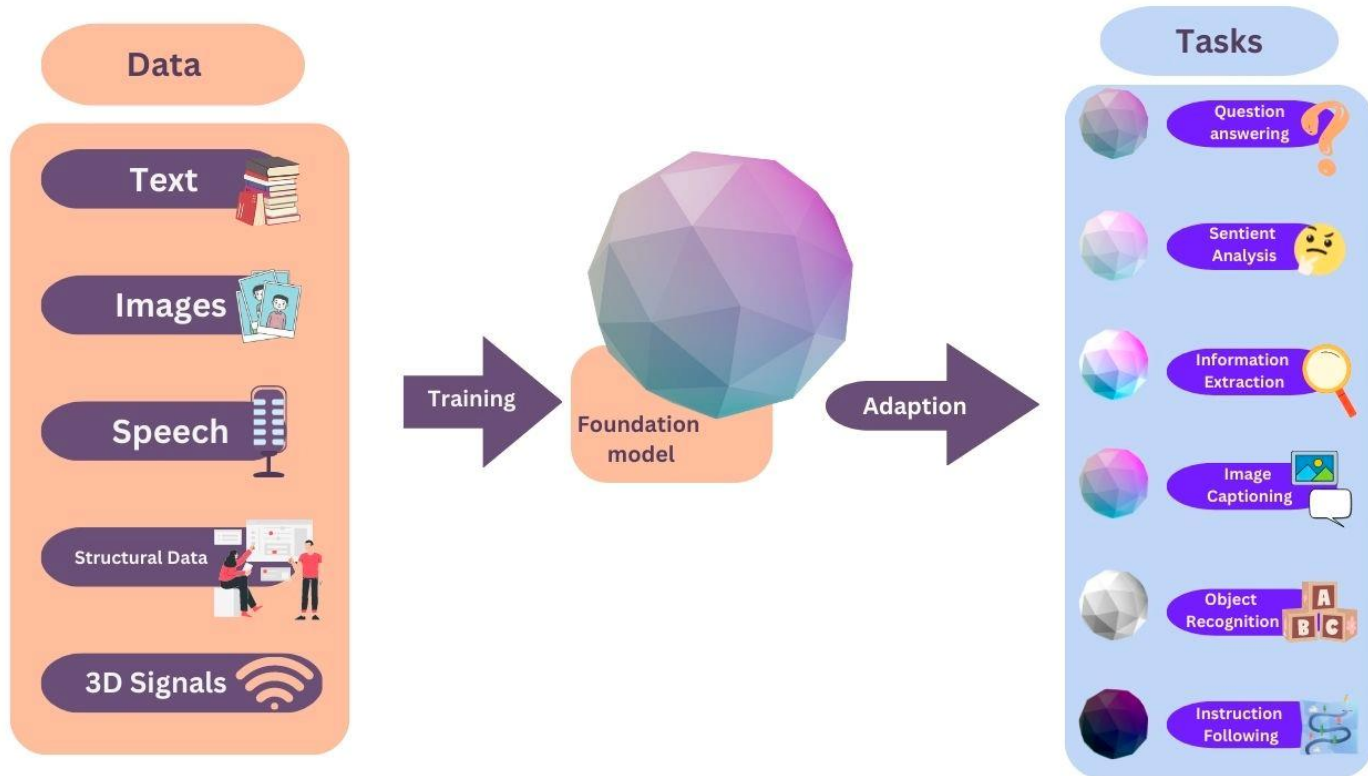
As an AI text-based model, I can assist you in providing a sample Verilog code for a basic RISC-V microcontroller unit (MCU). Please note that the code provided below is a simplified version and may not include all the necessary components of a complete RISC-V MCU. Additionally, the code assumes a 32-bit RISC-V architecture.

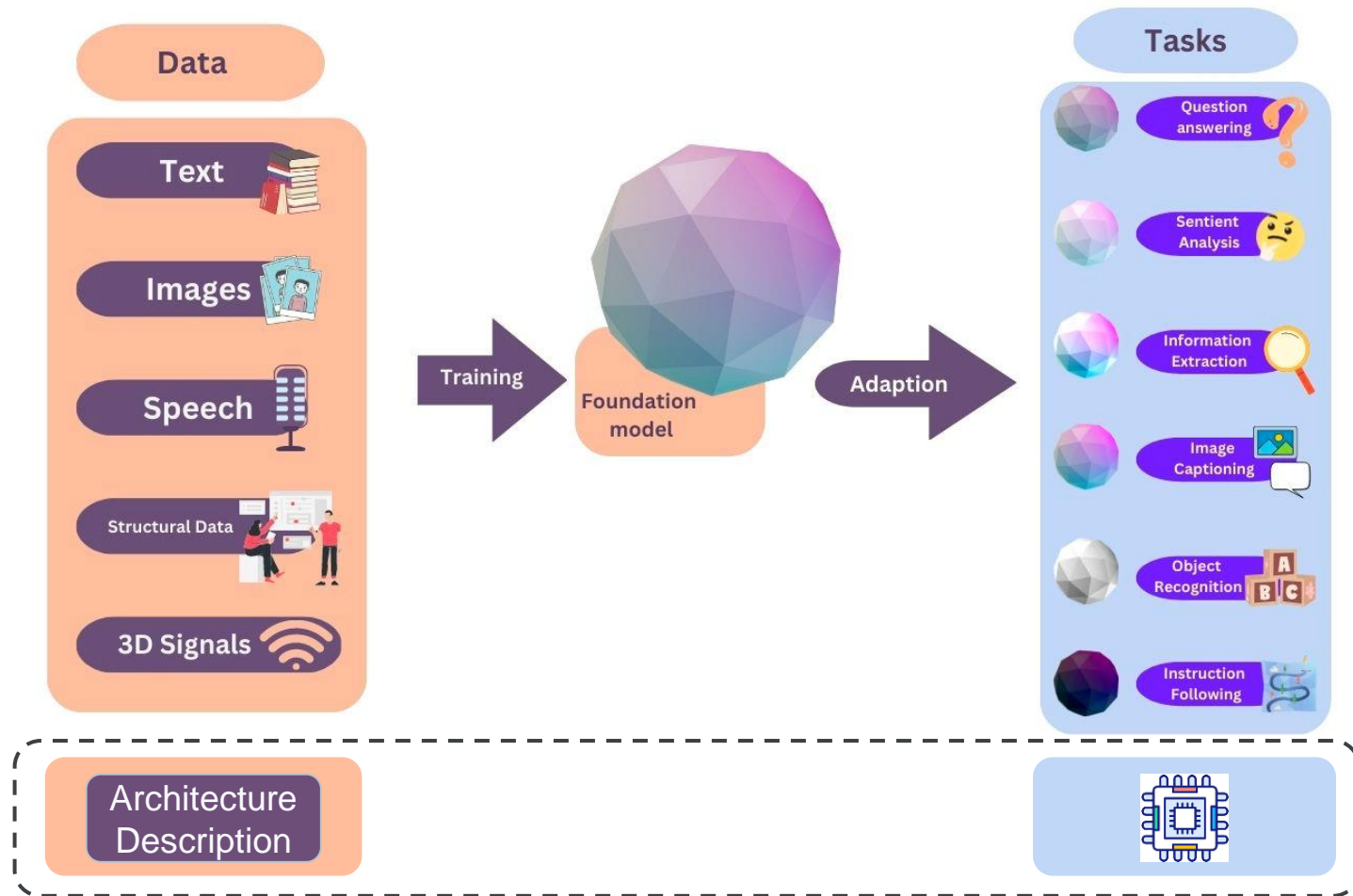


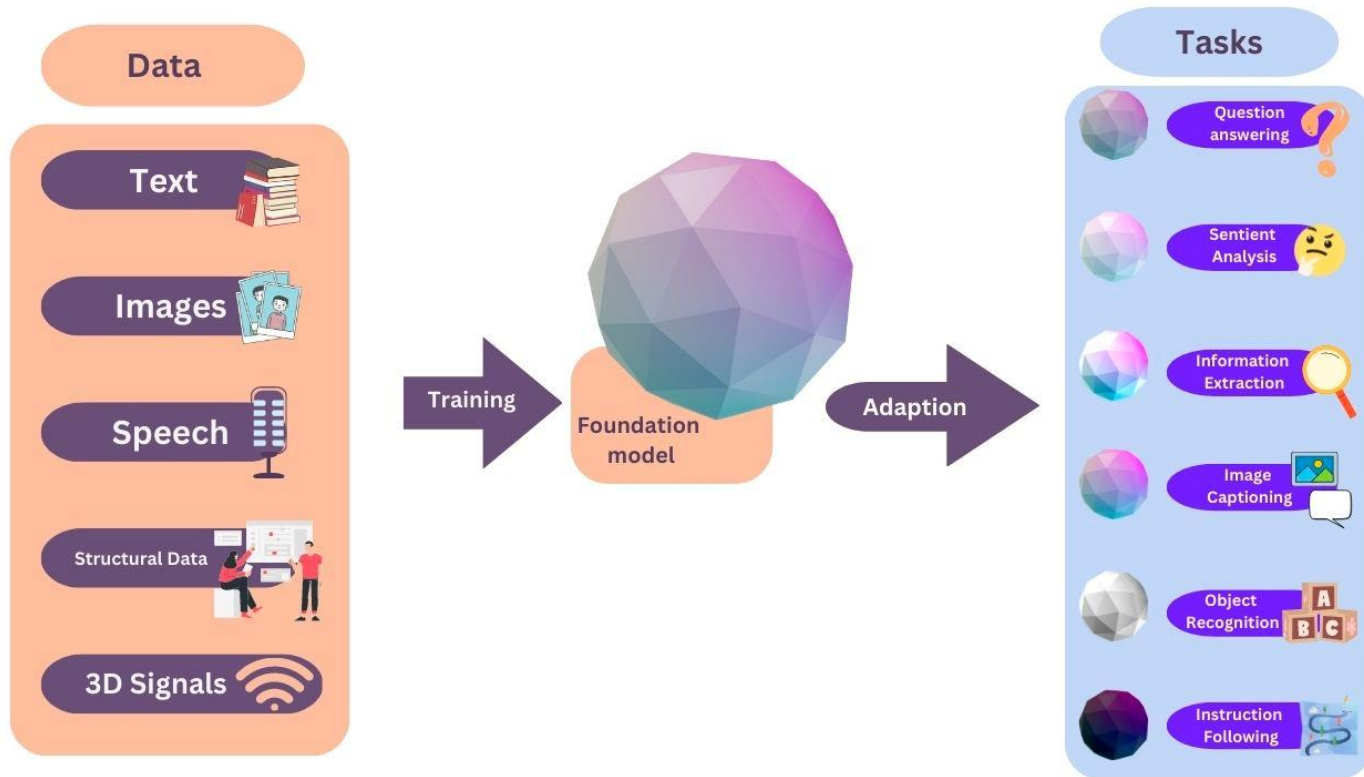
verilog

Copy code

```
module RISC_V_MCU(  
    input wire clk,  
    input wire reset,  
    // Add other input ports here  
  
    // Add other output ports here  
);
```

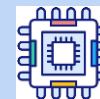






Architecture  
Description

*Can we develop a foundation  
model for hardware design?*





## Chip-Chat: Challenges and Opportunities in Conversational Hardware Design

Jason Blocklove  
New York University  
New York, NY USA  
jason.blocklove@nyu.edu

Siddharth Garg  
New York University  
New York, NY USA  
siddharth.garg@nyu.edu

Ramesh Karri  
New York University  
New York, NY USA  
rkarri@nyu.edu

Hammond Pearce  
University of New South Wales  
Sydney, Australia  
hammond.pearce@unsw.edu.au

### ChipGPT: How far are we from natural language hardware design

Kaiyan Chang<sup>1</sup>, Ying Wang<sup>1</sup>, Haimeng Ren<sup>1</sup>, Mengdi Wang<sup>1</sup>, Shengwen Liang<sup>1</sup>, Yinhe Han<sup>1</sup>, Huawei Li<sup>1</sup>, Xiaowei Li<sup>1</sup>  
State Key Lab of Processors, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China<sup>1</sup>  
University of Chinese Academy of Sciences<sup>2</sup>  
School of Information Science and Technology, ShanghaiTech University, Shanghai, China<sup>3</sup>  
Corresponding Author<sup>4</sup>  
changkaiyan@live.com, wangying2009@ict.ac.cn, rhen141246718@gmail.com, {wangmengdi17, liangshengwen, yinhe, lihuawei, liw}@ict.ac.cn

**efabless** Startups Universities Research Login or Signup

**Efabless Announces Winners of AI-Generated Open-Source Silicon Design Challenge**

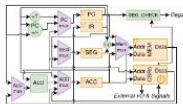
### Congratulations to the winners of the Efabless AI Generated Open-Source Silicon Design Challenge!

We would like to thank all the participants for your hard work, creativity using generative AI, and incredible speed in implementing and verifying your design through the Efabless Openlane flow.

The winning designs are the following:

#### 1st Place: QTCore-CI

QTCore-CI by Hammond Pearce. The design is a co-processor that can be used for many applications, such as predictable-time I/O state machines for PIO functions as seen on some microcontrollers developed using the Chip-Chat methodology that the NYU team has published.



[Learn more about the design](#)

## DAVE: Deriving Automatically Verilog from English

Hammond Pearce  
New York University  
Brooklyn, USA

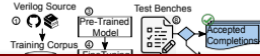
Benjamin Tan  
New York University  
Brooklyn, USA

Ramesh Karri  
New York University  
Brooklyn, USA

### Benchmarking Large Language Models for Automated Verilog RTL Code Generation

Shailja Thakur<sup>1</sup>, Balegh Ahmad<sup>2</sup>, Zhenxing Fan<sup>1</sup>, Hammond Pearce<sup>1</sup>, Benjamin Tan<sup>1</sup>, Ramesh Karri<sup>1</sup>, Brendan Dolan-Gavitt<sup>1</sup>, Siddharth Garg<sup>1</sup>  
<sup>1</sup>New York University, <sup>2</sup>University of Calgary

**Abstract**—Automating hardware design could obviate a significant amount of human error from the engineering process and lead to fewer errors. Verilog is a popular hardware description language to model and design digital systems, thus generating Verilog code is a critical first step. Emerging large language models (LLMs) programming of LLMs to generate Verilog code from English text. We benchmark the performance of Verilog code generation using LLMs. We compare the performance of LLMs to generate Verilog code from English text. We compare the performance of LLMs to generate Verilog code from English text. We compare the performance of LLMs to generate Verilog code from English text.



## EE Times

HOME NEWS PERSPECTIVES DESIGN LINES PODCASTS EDUCATION STORE SPECIAL REPORTS HOT TIMES EVENTS

### ARTICLE CONTENT

DESIGN LINES | AI & BIG DATA DESIGN LINES

### EDA 2.0: Catalyzing the Human

By Bilal Dahad 1/10/2022

### EE Times Energy Needs a Smarter Approach

Chips are getting more complex; that's a given. Whether it's for enabling more computing performance in less space or embedding more intelligence into sensors and endpoint devices, designs are becoming more sophisticated; require more design expertise to provide for power, performance, and area (PPA); and take longer for the whole design cycle, which includes verification and test.

As Moshik Rubin, a director for marketing at Cadence, said in an interview with embedded.com recently, systems-on-chip (SoCs) are getting increasingly larger and more complex, integrating hundreds of IP. With each of these IP blocks constantly changing as they evolve and improve, it becomes harder to manually correlate the different results and test failures. Determining the root cause of the failure can require dozens of engineers and multiple weeks.



Time well spent in the Cotswolds

### PODCAST



PowerUP Community: Wide Bandgap and Power Design  
By Maurizio Di Paolo Emilio  
07/07/2022

In this podcast, we cover the main topics from our PowerUP virtual conference, including wide bandgap, superjunction

[LISTEN](#)



Pixel 7a

arXiv:2305.13243v1 [cs.LG] 22 May 2023

**Abstract**—Modern natural language processing (NLP) models are trained on massive amounts of text data. This data is often noisy and contains a lot of irrelevant information. This noise can lead to the model learning spurious correlations that do not reflect the underlying structure of the data. This noise can lead to the model learning spurious correlations that do not reflect the underlying structure of the data.

**A. Trends in hard**  
As digital design software components (CAD) and Electronic Design approaches try to focus on identifying patterns [3]—tools, SaaS, ML, re-processes such as estimation. In the opportunities with the earliest stages Hardware Descrip

**B. Automating HD**  
While hardware they actually high in natural language. The process of Verilog must be consuming and er high-level synthesis capability in higher at the expense of I of Artificial Intelligence pathway for trans

arXiv:2305.14019v3 [cs.AI] 19 Jun 2023

**Abstract**—As digital design software components (CAD) and Electronic Design approaches try to focus on identifying patterns [3]—tools, SaaS, ML, re-processes such as estimation. In the opportunities with the earliest stages Hardware Descrip



Fig. 1. Images

arXiv:2009.01026v1 [cs.SE] 27 Aug 2020

**ABSTRACT**  
While specific gauge engine the program systems. And the language natural lang challenges. V (ML) to auto tuning GPT-3 prosa for pe tasks and pre sining transla with the ab

**1 INTRODUCTION**  
In pursuit of driven design goal of project is human precision Language HDL to match create domain As such, the create prod Given success the integr (e.g. [7, 14, 2 ML can help ML has re creasing DDL architectures These archi 2 [12] which gauge trans sion underst (dataset), and headlines [6 the "quality" producer hard Towards a gauge, we in gauge mode transfer lear [https://arxiv.org/abs/2009.01026v1] Project, Dec. 11 2020. URL: https://doi.org/10.26434/chemrxiv-2020-01026v1

**State-of-the-art**  
description lang specify hard process of s prone (1). As reduce design specification. We use of large gauge code successful in j Their use in g LLMs are former archi tribution of a sequence of v over the next paragraphs in from common We underst syntactic and code generate There are sev



# Architecture 2.0

Architecture 2.0: The era when we use AI/ML methods to **minimize human intervention** and build **complex, efficient** systems in a **shorter** timeframe.

“Act like an architect — design me a custom 64-bit RISC-V processor with full vector extension support and optimize it for less than 3 Watt TDP in a 7 nm LP process node using the TSMC plugin library”

“... add while you are at it add a few **custom functional units** that  
optimize the experience of Keyword Spotting.”



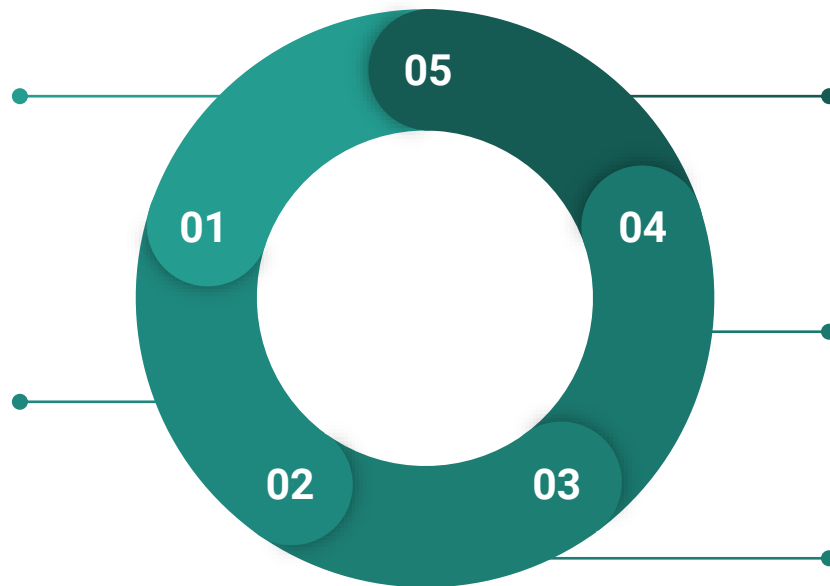
# Challenges

## Datasets

What datasets do we need? How we should collect these datasets for architecture research? What metadata should the datasets contain to enable broad usage? How do we create standard data formats from any ML algorithm?

## ML Algorithms

How can we learn and apply new ML algorithms to effectively design high-performance/efficient systems? How do we make our community more accessible to ML researchers? How do we embrace ML algorithm design as part of architecture research?



## Workforce & Training

Can we create a systematic playbook for best known methods? How do we ensure strong baselines and reproducibility?

## Tools & Infrastructure

How do we reduce the sim2real gap? What instrumentation mechanisms do we need for creating the datasets? What gym environments do we need to enable data-centric AI? How do we define standard data formats for interoperability?

## Best Practices

Can we create a systematic playbook for best known methods? How do we ensure strong baselines and reproducibility?

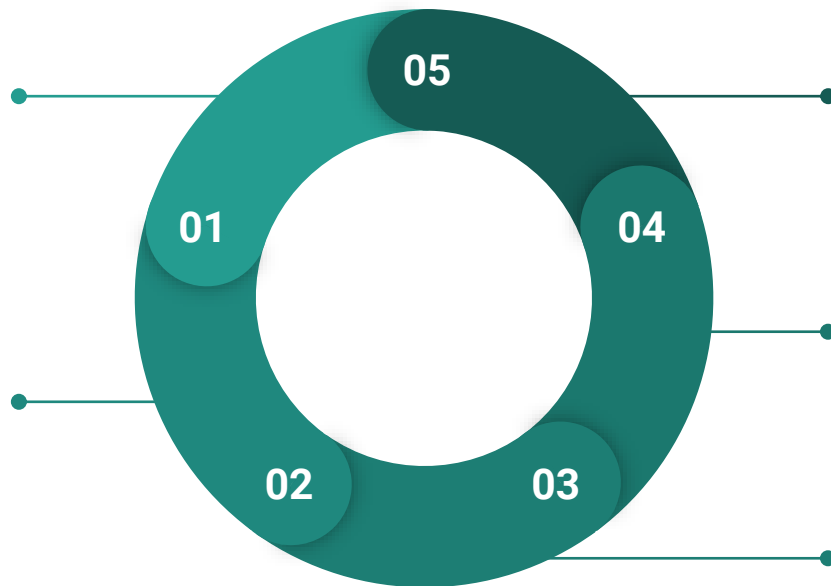
# Challenges

## Datasets

What datasets do we need? How we should collect these datasets for architecture research? What metadata should the datasets contain to enable broad usage? How do we create standard data formats from any ML algorithm?

## ML Algorithms

How can we learn and apply new ML algorithms to effectively design high-performance/efficient systems? How do we make our community more accessible to ML researchers? How do we embrace ML algorithm design as part of architecture research?



## Workforce & Training

Can we create a systematic playbook for best known methods? How do we ensure strong baselines and reproducibility?

## Tools & Infrastructure

How do we reduce the sim2real gap? What instrumentation mechanisms do we need for creating the datasets? What gym environments do we need to enable data-centric AI? How do we define standard data formats for interoperability?

## Best Practices

Can we create a systematic playbook for best known methods? How do we ensure strong baselines and reproducibility?

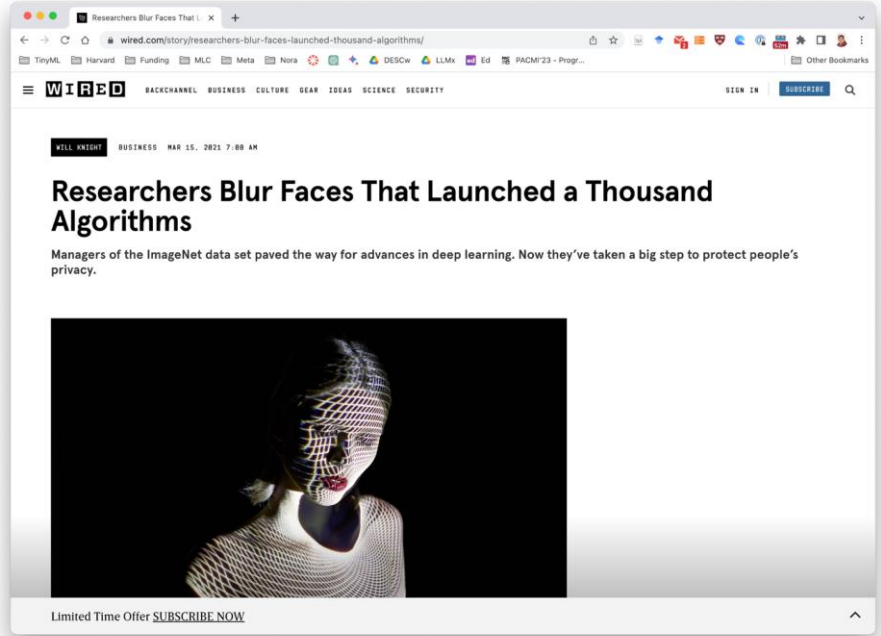
# Lack of large, high-quality public datasets



# Lack of large, high-quality public datasets



- Need public data, but data needs to be held private
- Need to strike a safe balance



# Inability to "scrape" the internet for creating public datasets



WIKIPEDIA  
The Free Encyclopedia





# Synthetic data generation from simulators



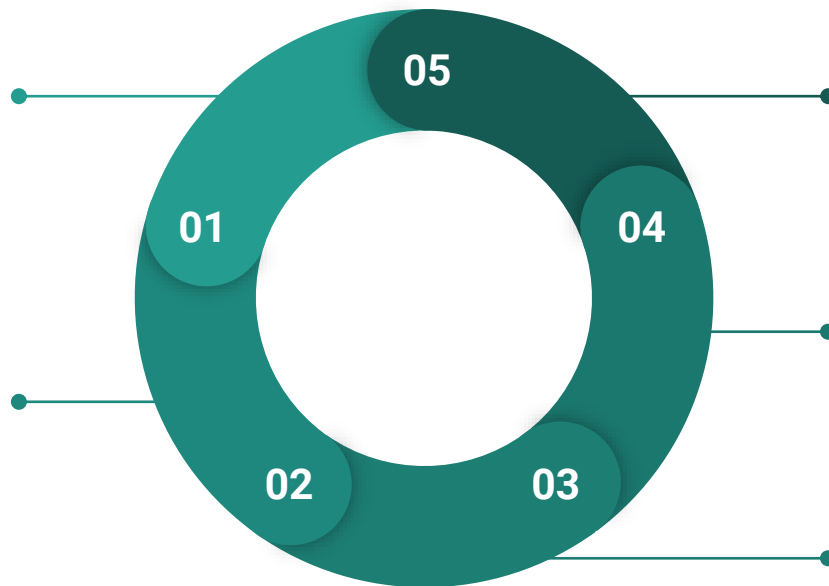
# Challenges

## Datasets

What datasets do we need? How we should collect these datasets for architecture research? What metadata should the datasets contain to enable broad usage? How do we create standard data formats from any ML algorithm?

## ML Algorithms

How can we learn and apply new ML algorithms to effectively design high-performance/efficient systems? How do we make our community more accessible to ML researchers? How do we embrace ML algorithm design as part of architecture research?



## Workforce & Training

Can we create a systematic playbook for best known methods? How do we ensure strong baselines and reproducibility?

## Tools & Infrastructure

How do we reduce the sim2real gap? What instrumentation mechanisms do we need for creating the datasets? What gym environments do we need to enable data-centric AI? How do we define standard data formats for interoperability?

## Best Practices

Can we create a systematic playbook for best known methods? How do we ensure strong baselines and reproducibility?



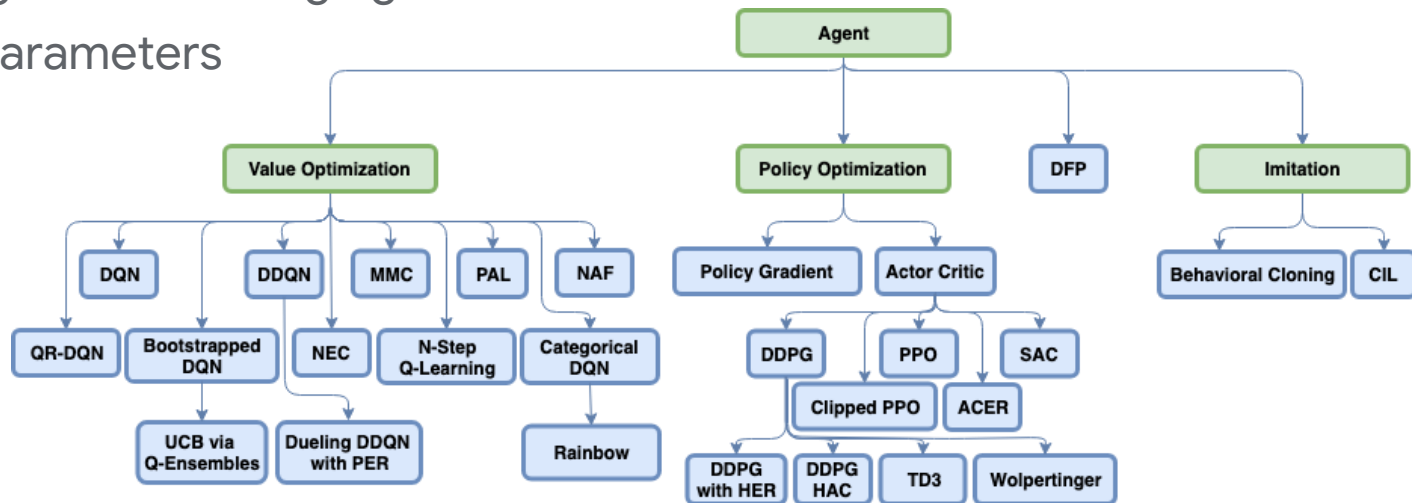
# Rapidly evolving ML algorithms landscape

- Many different algorithms out in the wild to choose from
- How do we know which algorithm is best suited for which architecture problem
- How do we compare these algorithms fairly against one another



# Rapidly evolving ML algorithms landscape

- Take RL for example
  - Many different variants exist
  - New algorithms emerging
  - Hyperparameters



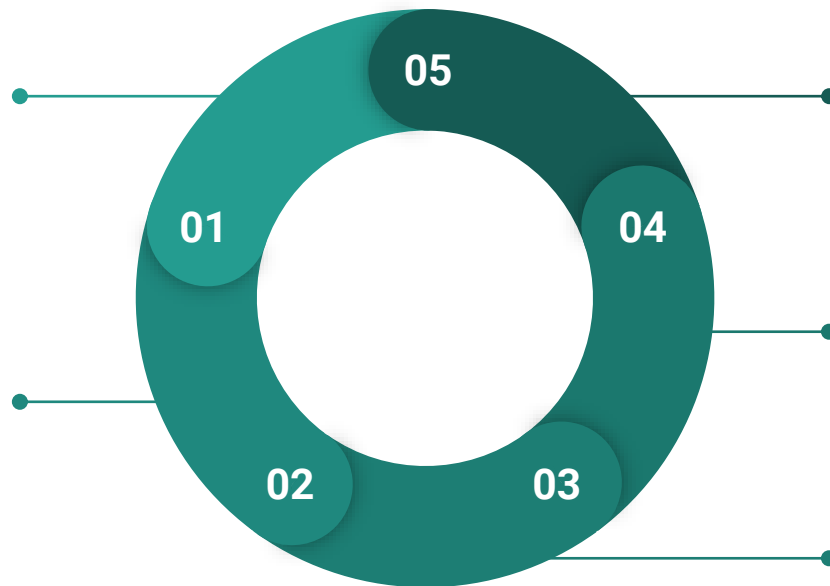
# Challenges

## Datasets

What datasets do we need? How we should collect these datasets for architecture research? What metadata should the datasets contain to enable broad usage? How do we create standard data formats from any ML algorithm?

## ML Algorithms

How can we learn and apply new ML algorithms to effectively design high-performance/efficient systems? How do we make our community more accessible to ML researchers? How do we embrace ML algorithm design as part of architecture research?



## Workforce & Training

Can we create a systematic playbook for best known methods? How do we ensure strong baselines and reproducibility?

## Tools & Infrastructure

How do we reduce the sim2real gap? What instrumentation mechanisms do we need for creating the datasets? What gym environments do we need to enable data-centric AI? How do we define standard data formats for interoperability?

## Best Practices

Can we create a systematic playbook for best known methods? How do we ensure strong baselines and reproducibility?

# Unclear applicability of ML algorithms to architecture problems

Abstract — “Machine learning has the potential to significantly improve systems, but only under certain conditions....”

Theme Article: Machine Learning for Systems

## A Taxonomy of ML for Systems Problems

Martin Maas  
Google Research, Brain Team

**Abstract**—Machine learning has the potential to significantly improve systems, but only under certain conditions. We describe a taxonomy to help identify whether or not machine learning should be applied to particular systems problems, and which approaches are most promising. We believe that this taxonomy can help practitioners and researchers decide how to most effectively use machine learning in their systems, and provide the community with a framework and vocabulary to discuss different approaches for applying machine learning in systems.

■ **MACHINE LEARNING (ML)** has transformed many research areas, from image recognition to natural language processing. ML has also had a significant impact on computer systems and inspired the development of new systems for designing and training ML models (e.g., TensorFlow), as well as new hardware (e.g., TPUs).

In contrast to such *Systems for ML* research, *ML for Systems* is only now seeing more attention. While ML has long been used in areas such as branch prediction, recent work has shown promising results in caching, compilers, and cluster scheduling. These advances indicate that ML could hold the key to improving many areas in computer systems. However, these successes

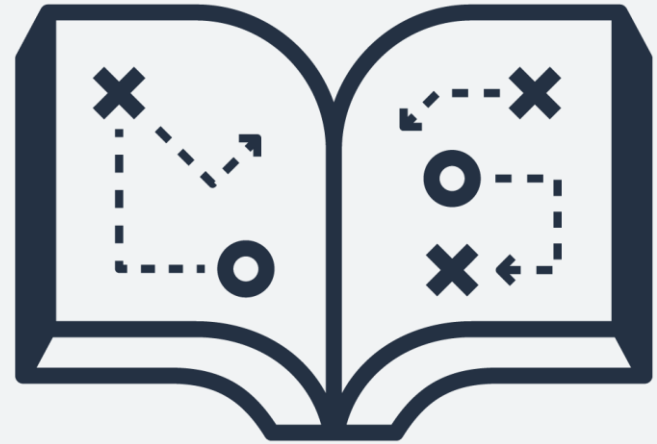
hide the fact that ML does not always lead to the immediate wins that its popularity promises. Applying ML to systems does not always outperform highly tuned non-ML solutions, and even if ML improves a particular metric, its resource cost does not always justify the improvement.

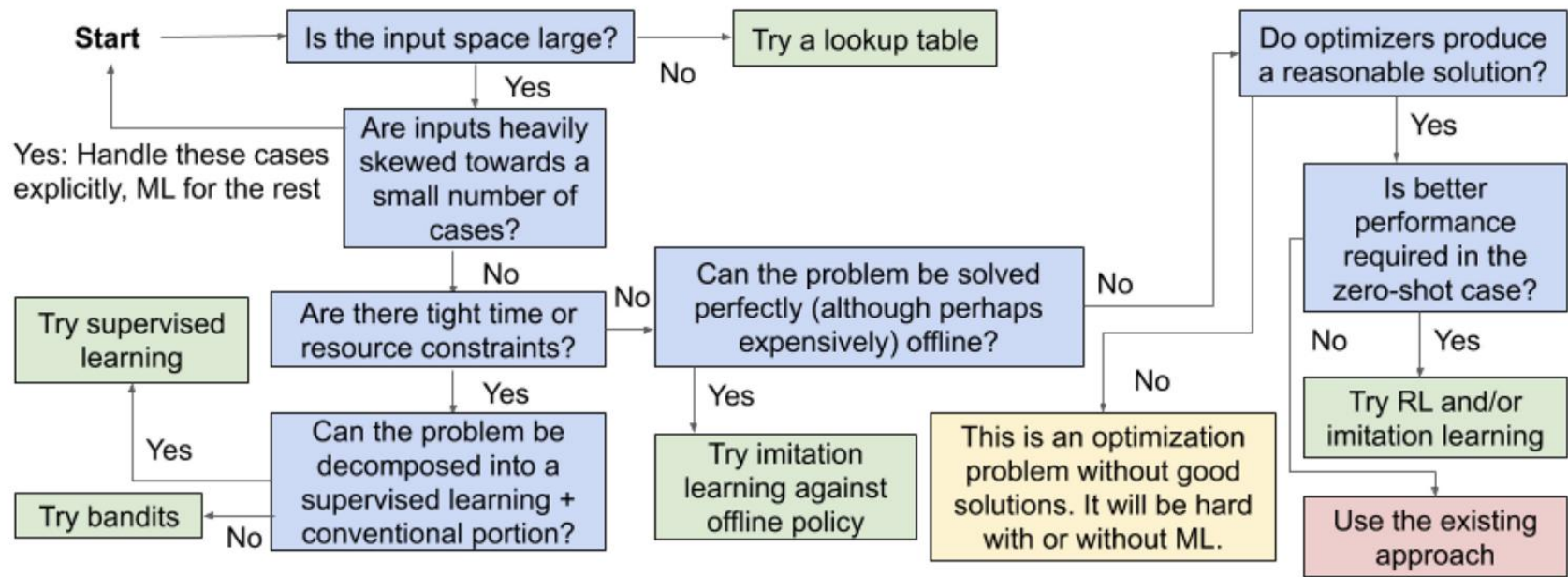
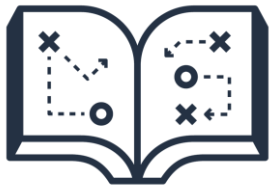
This article makes the case that while ML has the potential to improve systems, it does so only in certain cases. Furthermore, different ML techniques are suitable for different problems. We therefore categorize systems problems and develop a taxonomy for identifying whether ML can be applied, and what strategies might be suitable. We also provide a bibliography<sup>1</sup> that matches existing work to this taxonomy. We believe that our approach can help practitioners and researchers decide how to most effectively use ML in their systems and provide the research community with a framework to discuss ML for Systems strategies.

Digital Object Identifier 10.1109/MM.2020.3012883

Date of publication 30 July 2020; date of current version 1 September 2020.

“We therefore argue that effectively applying ML to systems requires identifying which part of a systems policy requires ML, and developing specific ML techniques for this part.”





**Figure 1.** How to decide which ML approach to use.

[Maas, IEEE Micro 2020]

# Difficulty with verifying, validating, and interpreting ML algorithms

## Task Performance

*How well does the agent perform **the task it was trained for**?*

## System Performance

*What are the **compute requirements** needed to train and deploy the agent?*

## Reliability

*How **stable is the agent's performance** during training and inference?*

## Generalization

*How well does the agent perform on **outside tasks** of what it was trained on?*

## Cost

*What are the **trade-offs** between using the various ML methods?*

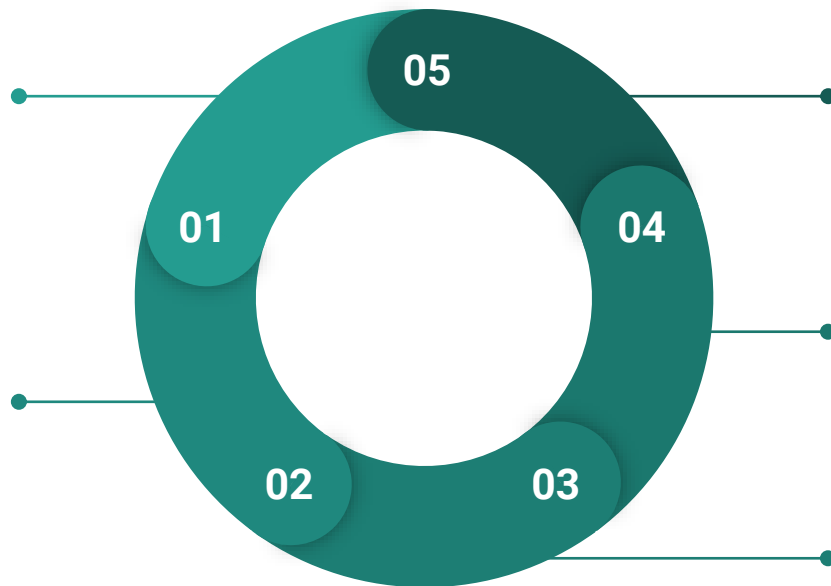
# Challenges

## Datasets

What datasets do we need? How we should collect these datasets for architecture research? What metadata should the datasets contain to enable broad usage? How do we create standard data formats from any ML algorithm?

## ML Algorithms

How can we learn and apply new ML algorithms to effectively design high-performance/efficient systems? How do we make our community more accessible to ML researchers? How do we embrace ML algorithm design as part of architecture research?



## Workforce & Training

Can we create a systematic playbook for best known methods? How do we ensure strong baselines and reproducibility?

## Tools & Infrastructure

How do we reduce the sim2real gap? What instrumentation mechanisms do we need for creating the datasets? What gym environments do we need to enable data-centric AI? How do we define standard data formats for interoperability?

## Best Practices

Can we create a systematic playbook for best known methods? How do we ensure strong baselines and reproducibility?

**Architecture Environment**

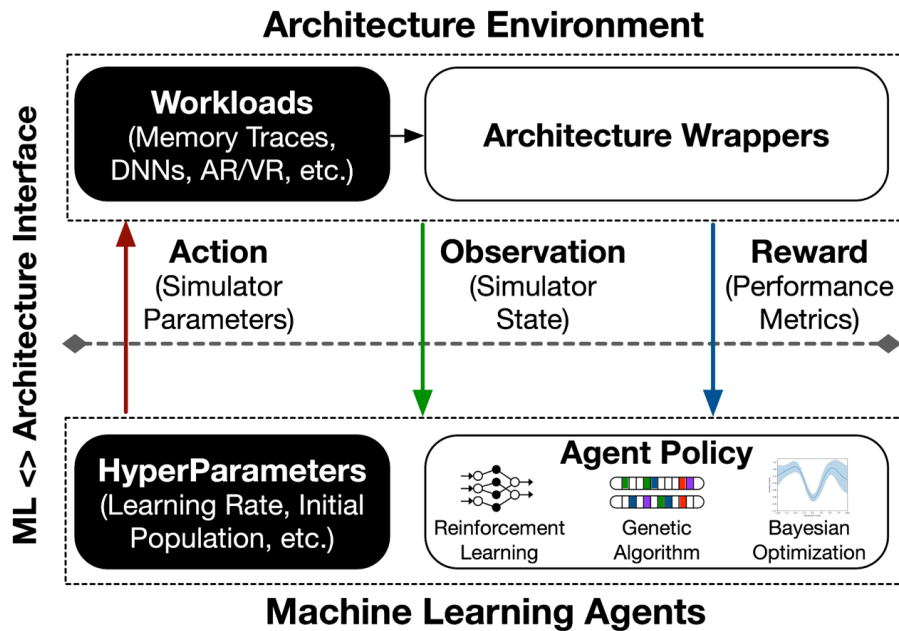


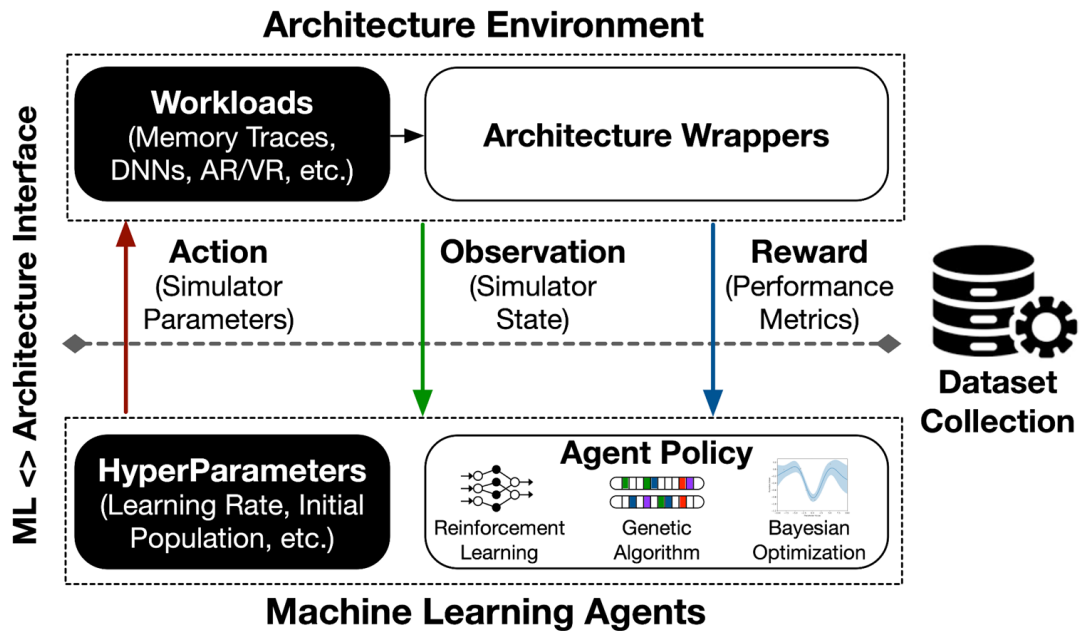
**ML <> Architecture Interface**



**Machine Learning Agents**







# ArchGym [ISCA '23]

## ● Contributions

- Enables ML-aided architecture design space exploration
- Fosters systematic evaluation and comparison of algorithms
- Data-centric framework for creating public datasets
- ...

## ArchGym: An Open-Source Gymnasium for Machine Learning Assisted Architecture Design

Srivatsan Krishnan  
srivatsan@seas.harvard.edu  
Harvard University  
Cambridge, Massachusetts, USA

Jason Jabbour  
jasonjabbour@g.harvard.edu  
Harvard University  
Cambridge, Massachusetts, USA

Behzad Boroujerdi  
behzadboro@utexas.edu  
UT Austin/Harvard University  
Cambridge, Massachusetts, USA

Aleksandra Faust  
faust@google.com  
Google Research, Brain Team  
Mountain View, California, USA

Amir Yazdanbakhsh  
ayazdan@google.com  
Google Research, Brain Team  
Mountain View, California, USA

Ikechukwu Uchendu  
iuchendu@g.harvard.edu  
Harvard University  
Cambridge, Massachusetts, USA

Daniel Richins  
drichins@utexas.edu  
UT Austin  
Austin, Texas, USA

Shvetank Prakash  
sprakash@g.harvard.edu  
Harvard University  
Cambridge, Massachusetts, USA

Susobhan Ghosh  
susobhan\_ghosh@g.harvard.edu  
Harvard University  
Cambridge, Massachusetts, USA

Devashree Tripathy  
devashreetripathy@iitbbs.ac.in  
IIT Bhubaneswar/Harvard University  
Bhubaneswar, Odisha, India

Vijay Janapa Reddi  
vj@eecs.harvard.edu  
Harvard University  
Cambridge, Massachusetts, USA

### ABSTRACT

Machine learning (ML) has become a prevalent approach to tame the complexity of design space exploration for domain-specific architecture. While appealing, using ML for design space exploration poses several challenges. First, it is not straightforward to identify the most suitable algorithm from an ever-increasing pool of ML methods. Second, assessing the trade-offs between performance and sample efficiency across these methods is inconclusive. Finally, the lack of a holistic framework for fair, reproducible, and objective comparison across these methods hinders the progress of adopting ML-aided architecture design space exploration and impedes creating repeatable artifacts. To mitigate these challenges, we introduce ArchGym, an open-source gymnasium and easy-to-extend framework that connects a diverse range of search algorithms to architecture simulators. To demonstrate its utility, we evaluate ArchGym across multiple vanilla and domain-specific search algorithms in the design of a custom memory controller, deep neural network accelerators, and a custom SoC for AR/VR workloads, collectively encompassing over 21K experiments. The results suggest that with an unlimited number of samples, ML algorithms are equally favorable to meet the user-defined target specification if its hyperparameters are tuned thoroughly; no one solution is

necessarily better than another (e.g., reinforcement learning vs. Bayesian methods). We coin the term “*hyperparameter lottery*” to describe the relatively probable chance for a search algorithm to find an optimal design provided meticulously selected hyperparameters. Additionally, the ease of data collection and aggregation in ArchGym facilitates research in ML-aided architecture design space exploration. As a case study, we show this advantage by developing a proxy cost model with an RMSE of 0.61% that offers a 2,000-fold reduction in simulation time. Code and data for ArchGym is available at <https://bit.ly/ArchGym>.

### CCS CONCEPTS

• Computer systems organization → Architectures; • Computing methodologies → Reinforcement learning; Machine learning algorithms; Bio-inspired approaches.

### KEYWORDS

Machine learning, Machine Learning for Computer Architecture, Machine Learning for System, Reinforcement Learning, Bayesian Optimization, Open Source, Baselines, Reproducibility

### ACM Reference Format:

Srivatsan Krishnan, Amir Yazdanbakhsh, Shvetank Prakash, Jason Jabbour, Ikechukwu Uchendu, Susobhan Ghosh, Behzad Boroujerdi, Daniel Richins, Devashree Tripathy, Aleksandra Faust, and Vijay Janapa Reddi. 2023. ArchGym: An Open-Source Gymnasium for Machine Learning Assisted Architecture Design. In *Proceedings of the 50th Annual International Symposium on Computer Architecture (ISCA '23)*, June 17–21, 2023, Orlando, FL, USA. ACM, New York, NY, USA, 16 pages. <https://doi.org/10.1145/3579371.358040>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).  
ISCA '23, June 17–21, 2023, Orlando, FL, USA  
© 2023 Copyright held by the owner/authors. Publication rights licensed to ACM.  
ACM ISBN 978-0-400-0095-4/23/06...\$15.00  
<https://doi.org/10.1145/3579371.358040>

[Krishnan et al. ISCA'23]

# ArchGym [ISCA '23]



## ArchGym: An Open-Source Gymnasium for Machine Learning Assisted Architecture Design

Srivatsan Krishnan  
srivatsan@seas.harvard.edu  
Harvard University  
Cambridge, Massachusetts, USA

Jason Jabbour  
jasonjabbour@g.harvard.edu  
Harvard University  
Cambridge, Massachusetts, USA

Behzad Boroujerdi  
behzadboro@utexas.edu  
UT Austin/Harvard University  
Cambridge, Massachusetts, USA

Aleksandra Faust  
faust@google.com  
Google Research, Brain Team  
Mountain View, California, USA

Amir Yazdanbakhsh  
ayazdan@google.com  
Google Research, Brain Team  
Mountain View, California, USA

Ikechukwu Uchendu  
iuchendu@g.harvard.edu  
Harvard University  
Cambridge, Massachusetts, USA

Daniel Richins  
drichins@utexas.edu  
UT Austin  
Austin, Texas, USA

Shvetank Prakash  
sprakash@g.harvard.edu  
Harvard University  
Cambridge, Massachusetts, USA

Susobhan Ghosh  
susobhan\_ghosh@g.harvard.edu  
Harvard University  
Cambridge, Massachusetts, USA

Devashree Tripathy  
devashreetripathy@iitbbs.ac.in  
IIT Bhubaneswar/Harvard University  
Bhubaneswar, Odisha, India

Vijay Janapa Reddi  
vj@eecs.harvard.edu  
Harvard University  
Cambridge, Massachusetts, USA

### ABSTRACT

Machine Learning (ML) has become a prevalent approach to tame the complexity of design space exploration for domain-specific architecture. While appealing, using ML for design space exploration poses several challenges. First, it is not straightforward to identify the most suitable algorithm from an ever-increasing pool of ML methods. Second, assessing the trade-offs between performance and sample efficiency across these methods is inconclusive. Finally, the lack of a holistic framework for fair, reproducible, and objective comparison across these methods hinders the progress of adopting ML-aided architecture design space exploration and impedes creating repeatable artifacts. To mitigate these challenges, we introduce ArchGym, an open-source gymnasium and easy-to-extend framework that connects a diverse range of search algorithms to architecture simulators. To demonstrate its utility, we evaluate ArchGym across multiple vanilla and domain-specific search algorithms in the design of a custom memory controller, deep neural network accelerators, and a custom SoC for AR/VR workloads, collectively encompassing over 21K experiments. The results suggest that with an unlimited number of samples, ML algorithms are equally favorable to meet the user-defined target specification if its hyperparameters are tuned thoroughly; no one solution is

necessarily better than another (e.g. reinforcement learning vs. Bayesian methods). We coin the term “*hyperparameter lottery*” to describe the relatively probable chance for a search algorithm to find an optimal design provided meticulously selected hyperparameters. Additionally, the ease of data collection and aggregation in ArchGym facilitates research in ML-aided architecture design space exploration. As a case study, we show this advantage by developing a proxy cost model with an RMSE of 0.61% that offers a 2,000-fold reduction in simulation time. Code and data for ArchGym is available at <https://bit.ly/ArchGym>.

### CCS CONCEPTS

• Computer systems organization → Architectures; • Computing methodologies → Reinforcement learning; Machine learning algorithms; Bio-inspired approaches.

### KEYWORDS

Machine learning, Machine Learning for Computer Architecture, Machine Learning for System, Reinforcement Learning, Bayesian Optimization, Open Source, Baselines, Reproducibility

### ACM Reference Format:

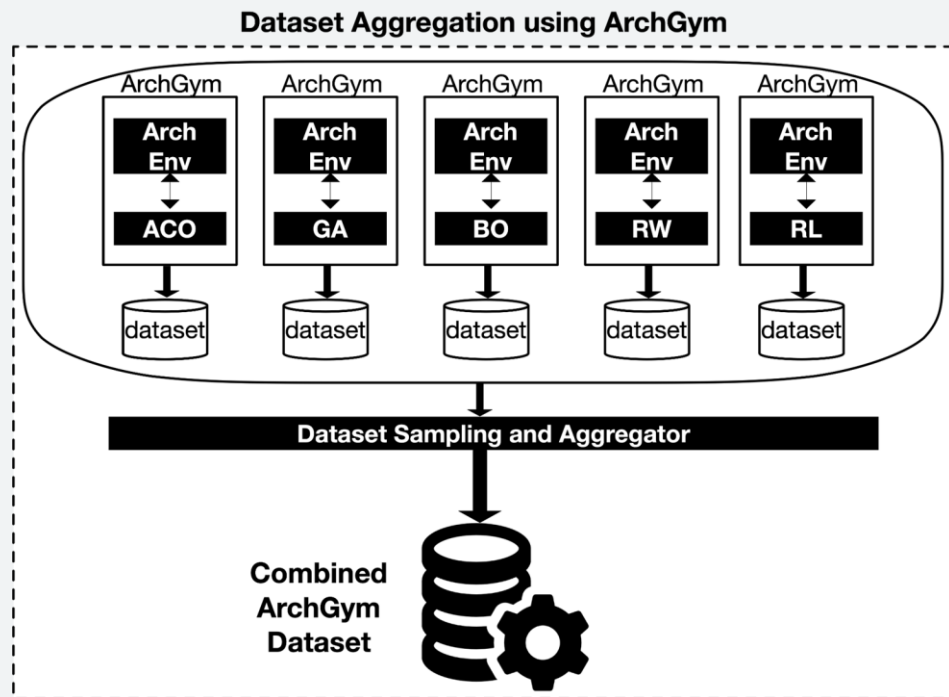
Srivatsan Krishnan, Amir Yazdanbakhsh, Shvetank Prakash, Jason Jabbour, Ikechukwu Uchendu, Susobhan Ghosh, Behzad Boroujerdi, Daniel Richins, Devashree Tripathy, Aleksandra Faust, and Vijay Janapa Reddi. 2023. ArchGym: An Open-Source Gymnasium for Machine Learning Assisted Architecture Design. In *Proceedings of the 50th Annual International Symposium on Computer Architecture (ISCA '23)*, June 17–21, 2023, Orlando, FL, USA. ACM, New York, NY, USA, 16 pages. <https://doi.org/10.1145/3579371.3580640>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).  
ISCA '23, June 17–21, 2023, Orlando, FL, USA.  
© 2023 Copyright held by the owner/authors. Publication rights licensed to ACM.  
ACM ISBN 978-0-400-0095-8/23/06...\$15.00  
<https://doi.org/10.1145/3579371.3580640>

[Krishnan et al. ISCA'23]

# How does ArchGym address these issues?

1. Lack of large, high-quality (i.e., representative) public datasets
2. Inability to "scrape" the internet for creating public datasets
3. Data generation from cycle-level simulators is slow and difficult
4. Rapidly evolving ML algorithms landscape
5. Unclear applicability of ML algorithms to architecture problems
6. Need for agile full-stack co-design
7. Difficulty with verifying, validating, and interpreting ML algorithms



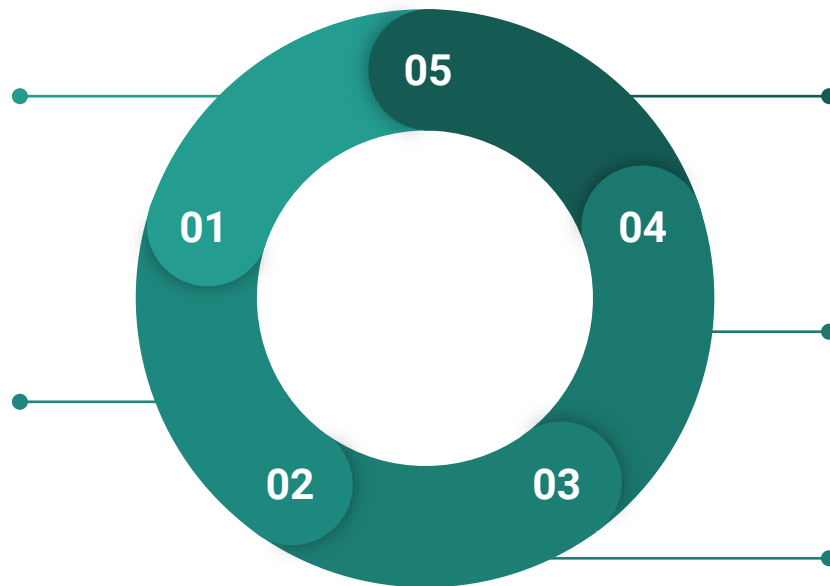
# Challenges

## Datasets

What datasets do we need? How we should collect these datasets for architecture research? What metadata should the datasets contain to enable broad usage? How do we create standard data formats from any ML algorithm?

## ML Algorithms

How can we learn and apply new ML algorithms to effectively design high-performance/efficient systems? How do we make our community more accessible to ML researchers? How do we embrace ML algorithm design as part of architecture research?



## Workforce & Training

Can we create a systematic playbook for best known methods? How do we ensure strong baselines and reproducibility?

## Tools & Infrastructure

How do we reduce the sim2real gap? What instrumentation mechanisms do we need for creating the datasets? What gym environments do we need to enable data-centric AI? How do we define standard data formats for interoperability?

## Best Practices

Can we create a systematic playbook for best known methods? How do we ensure strong baselines and reproducibility?

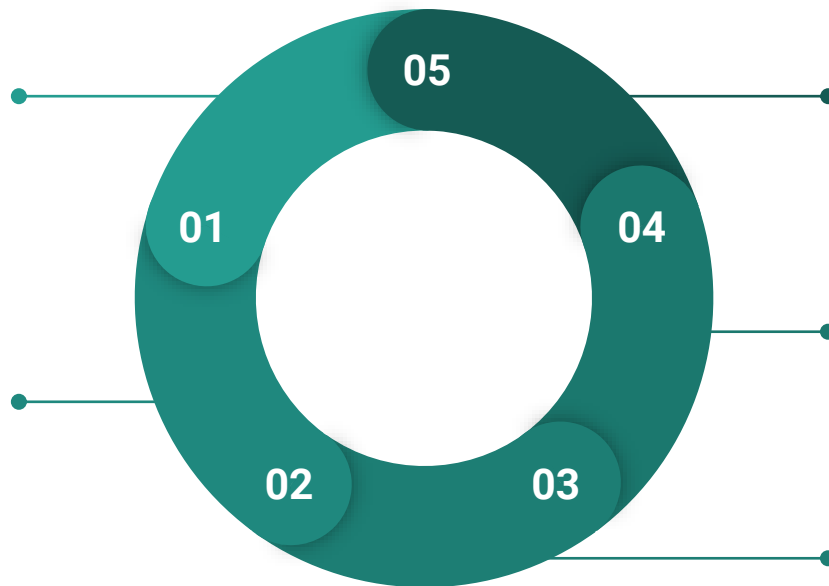
# Challenges

## Datasets

What datasets do we need? How we should collect these datasets for architecture research? What metadata should the datasets contain to enable broad usage? How do we create standard data formats from any ML algorithm?

## ML Algorithms

How can we learn and apply new ML algorithms to effectively design high-performance/efficient systems? How do we make our community more accessible to ML researchers? How do we embrace ML algorithm design as part of architecture research?



## Workforce & Training

Can we create a systematic playbook for best known methods? How do we ensure strong baselines and reproducibility?

## Tools & Infrastructure

How do we reduce the sim2real gap? What instrumentation mechanisms do we need for creating the datasets? What gym environments do we need to enable data-centric AI? How do we define standard data formats for interoperability?

## Best Practices

Can we create a systematic playbook for best known methods? How do we ensure strong baselines and reproducibility?

# Where we are: CFUPlayground-ChatGPT Plugin

Given source code in a high level language like C or C++ for a function of a program, generate a CFU in Verilog to accelerate only part of the function in hardware and rewrite the C function code to update it by invoking the hardware accelerator.

Your Task:

Accelerate the primecount workload (file: 'primecount.c') in CFU Playground. We will need to perform multiple steps for this. Let's begin with getting the latency of the primecount workload in CFU Playground and let me know what it is. Then await my next instruction.

Used unknown plugin

The current latency of the primecount workload in CFU Playground is 3,760,624 clock cycles. Please provide the next instruction.

Get the 'primecount.c' file source code of the workload and add line numbers to it.

Used unknown plugin

Here is the source code of the 'primecount.c' file with line numbers added:

```
c
1. #include "third_party/embench_iot_v1/support/support.h"
```

Write the CFU in Verilog now that implements these functions using the template provided and description of how custom function units work in the very first context message. \*You must follow the CfU Verilog interface provided in the first message and can not modify this.\* Make sure all the code you write is \*synthesizeable Verilog\*. Write the Verilog step by step for each instruction and add comments so I can follow.

Sure, let's start writing the CFU in Verilog. We'll be implementing three functions: square, less than comparison, addition, and equality comparison. We'll use the function ID to decide which operation to perform. Here's how we can do it:

```
verilog
module CfU (
    input          cmd_valid,
    output         cmd_ready,
    input  [9:0]   cmd_payload_function_id,
    input  [31:0] cmd_payload_inputs_0,
```

Profile the workload where you think there might be a bottleneck or time consuming operation in the source code that would be good to accelerate via a custom function unit. Make sure you target only a "small part" of a function to profile (5-10 lines max). Provide the filename and start and end line numbers of the code you would like to profile based on the line numbers you added to the code in the previous message.

Used unknown plugin

The code section from lines 21 to 26 in the 'primecount.c' file has been profiled. The execution time for this section is 107,098 clock cycles. This section seems to be a potential bottleneck in the code and could be a good candidate for acceleration via a custom function unit (CFU). Please provide the next instruction.

Okay now pick some lines between the lines you just profiled to move to hardware by replacing them with the cfu\_op API call. Recall you must follow the API definitions and replace with your operands accordingly. Wait for further instructions after you do this and update the lines of code you choose to accelerate. Make sure for all the logic you replace you reason out step by step whether the logic can be completed in a single cfu instruction or will need to be broken down into multiple cfu instructions, taking into account that you can only pass two arguments to the cfu per instruction.

Looking at the lines 21 to 26 in the 'primecount.c' file:

```
c
21. while (sq*sqr <= trial) ++sq;
22. --sqr;
```

Now using CFU Playground, compile this Verilog module of yours.

Used unknown plugin

The Verilog code for the custom function unit (CFU) has been successfully written and compiled in CFU Playground. Please provide the next instruction.

Now describe step by step what the behavior of your CFU should be in natural language.

Sure, here's a step-by-step description of the behavior of the CFU:

- Initialization:** When the CFU is reset (i.e., when the 'reset' signal is high), it initializes the output register 'rsp\_payload\_outputs\_0' to 0 and the 'rsp\_valid' signal to 0. This indicates that there is no valid result available at the moment.

Great, now let's run physical implementation on our design using the Sky130 process!

Used unknown plugin

The physical implementation of our CFU design using the Sky130 process has been completed successfully!

The screenshot shows a GitHub repository page for 'CFUPlayground-ChatGPT-Plugin'. The repository is owned by 'ShvetankPrakash' and has 4 commits. The file list includes: well-known, test, .gitignore, LICENSE, README.md, logo.png, main.py, openapi.yaml, and requirements.txt. The 'README.md' file is selected, showing the title 'CFU Playground ChatGPT Plugin Quickstart'. The content includes a 'Setup' section with the instruction: 'To install the required packages for this plugin, run the following command: pip install -r requirements.txt'.





Great, now let's run physical implementation on our design using the Sky130 process!



Used unknown plugin



The physical implementation of our CFU design using the Sky130 process has been completed successfully!

# Building the dataset

```
for (int out_channel = 0; out_channel < output_depth; ++out_channel) {
  int32_t acc = cfu_op0(/* funct7= */ 1, 0, 0); // resets acc
  for (int filter_y = 0; filter_y < 1; ++filter_y) {
    const int in_y = in_y_origin + filter_y;
    for (int filter_x = 0; filter_x < 1; ++filter_x) {
      const int in_x = in_x_origin + filter_x;

      // Zero padding by omitting the areas outside the image.
      const bool is_point_inside_image =
        (in_x >= 0) && (in_x < input_width) && (in_y >= 0) &&
        (in_y < input_height);

      if (!is_point_inside_image) {
        continue;
      }

      for (int in_channel = 0; in_channel < input_depth; in_channel += 4) {
        uint32_t input_val = *((uint32_t*)(input_data + Offset(
          input_shape, batch, in_y, in_x, in_channel)));

        uint32_t filter_val = *((uint32_t*)(filter_data + Offset(
          filter_shape, out_channel, filter_y, filter_x, in_channel)));
        acc = cfu_op0(/* funct7= */ 0, /* in0= */ input_val, /* in1= */ filter_val);
      }
    }
  }

  if (bias_data) {
    acc += bias_data[out_channel];
  }
  acc = MultiplyByQuantizedMultiplier(
    acc, output_multiplier[out_channel], output_shift[out_channel]);
  acc += output_offset;
  acc = std::max(acc, output_activation_min);
  acc = std::min(acc, output_activation_max);
  output_data[Offset(output_shape, batch, out_y, out_x, out_channel)] =
    static_cast<int8_t>(acc);
}
```

```
module Cfu (
  input          cmd_valid,
  output         cmd_ready,
  input  [9:0]   cmd_payload_function_id,
  input  [31:0]  cmd_payload_inputs_0,
  input  [31:0]  cmd_payload_inputs_1,
  output reg    rsp_valid,
  input        rsp_ready,
  output reg [31:0]  rsp_payload_outputs_0,
  input        reset,
  input        clk
);

localparam InputOffset = $signed(9'd128);

// SIMD multiply step:
wire signed [15:0] prod_0, prod_1, prod_2, prod_3;
assign prod_0 = ($signed(cmd_payload_inputs_0[7 : 0]) + InputOffset)
  * $signed(cmd_payload_inputs_1[7 : 0]);
assign prod_1 = ($signed(cmd_payload_inputs_0[15: 8]) + InputOffset)
  * $signed(cmd_payload_inputs_1[15: 8]);
assign prod_2 = ($signed(cmd_payload_inputs_0[23:16]) + InputOffset)
  * $signed(cmd_payload_inputs_1[23:16]);
assign prod_3 = ($signed(cmd_payload_inputs_0[31:24]) + InputOffset)
  * $signed(cmd_payload_inputs_1[31:24]);

wire signed [31:0] sum_prods;
assign sum_prods = prod_0 + prod_1 + prod_2 + prod_3;

// Only not ready for a command when we have a response.
assign cmd_ready = ~rsp_valid;

always @(posedge clk) begin
  if (reset) begin
    rsp_payload_outputs_0 <= 32'b0;
    rsp_valid <= 1'b0;
  end else if (rsp_valid) begin
    // Waiting to hand off response to CPU.
    rsp_valid <= ~rsp_ready;
  end else if (cmd_valid) begin
    rsp_valid <= 1'b1;
    // Accumulate step:
    rsp_payload_outputs_0 <= |cmd_payload_function_id[9:3]
      ? 32'b0
      : rsp_payload_outputs_0 + sum_prods;
  end
end
```

# Long-term Goal

## LLM Playground for HW-SW Co-Design of TinyML Acceleration

Enter the TFLM function you want to accelerate:

dot product

```
int dotProduct(int vector1[], int vector2[], int size)
{
    int result = 0;
    for (int i = 0; i < size; i++)
    {
        result += vector1[i] * vector2[i];
    }
    return result;
}
```

```
module Cfu(
    input          cmd_valid,
    output         cmd_ready,
    input  [9:0]   cmd_payload_function_id,
    input  [31:0] cmd_payload_inputs_0,
    input  [31:0] cmd_payload_inputs_1,
    output         rsp_valid,
    input         rsp_ready,
    output [31:0] rsp_payload_outputs_0,
    input         reset,
    input         clk
```

gpt4all.io/index.html

NOMIC

### Performance Benchmarks

Model	BoolQ	PIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA	Avg
GPT4All-J 6B v1.0	73.4	74.8	63.4	64.7	54.9	36	40.2	58.2
GPT4All-J v1.1-breezy	74	75.1	63.2	63.6	55.4	34.9	38.4	57.8
GPT4All-J v1.2-jazzy	74.8	74.9	63.6	63.8	56.6	35.3	41	58.6
GPT4All-J v1.3-groovy	73.6	74.3	63.8	63.5	57.7	35	38.8	58.1
GPT4All-J Lora 6B	68.6	75.8	66.2	63.5	56.4	35.7	40.2	58.1
GPT4All LLaMa Lora 7B	73.1	77.6	72.1	67.8	51.1	40.4	40.2	60.3
GPT4All 13B snoozy	<b>83.3</b>	79.2	75	71.3	60.9	44.2	43.4	65.3
GPT4All Falcon	77.6	79.8	74.9	70.1	67.9	43.4	42.6	65.2
Nous-Hermes	79.5	78.9	<b>80</b>	<b>71.9</b>	<b>74.2</b>	<b>50.9</b>	<b>46.4</b>	<b>68.8</b>
Dolly 6B	68.8	77.3	67.6	63.9	62.9	38.7	41.2	60.1
Dolly 12B	56.7	75.4	71	62.2	64.6	38.5	40.4	58.4
Alpaca 7B	73.9	77.2	73.9	66.1	59.8	43.3	43.4	62.5
Alpaca Lora 7B	74.3	79.3	74	68.8	56.6	43.9	42.6	62.8
GPT-J 6.7B	65.4	76.2	66.2	64.1	62.2	36.6	38.2	58.4
LLama 7B	73.1	77.4	73	66.9	52.5	41.4	42.4	61.0
LLama 13B	68.5	79.1	76.2	70.1	60	44.6	42.2	63.0

LLM-Playground Private

Unwatch 1 | Fork 0 | Star 0

main | 1 branch | 0 tags | Go to file | Add file | Code

ShvetankPrakash Initial commit 3e84232 last month | 1 commit

- LICENSE Initial commit last month
- README.md Initial commit last month

README.md

### LLM-Playground

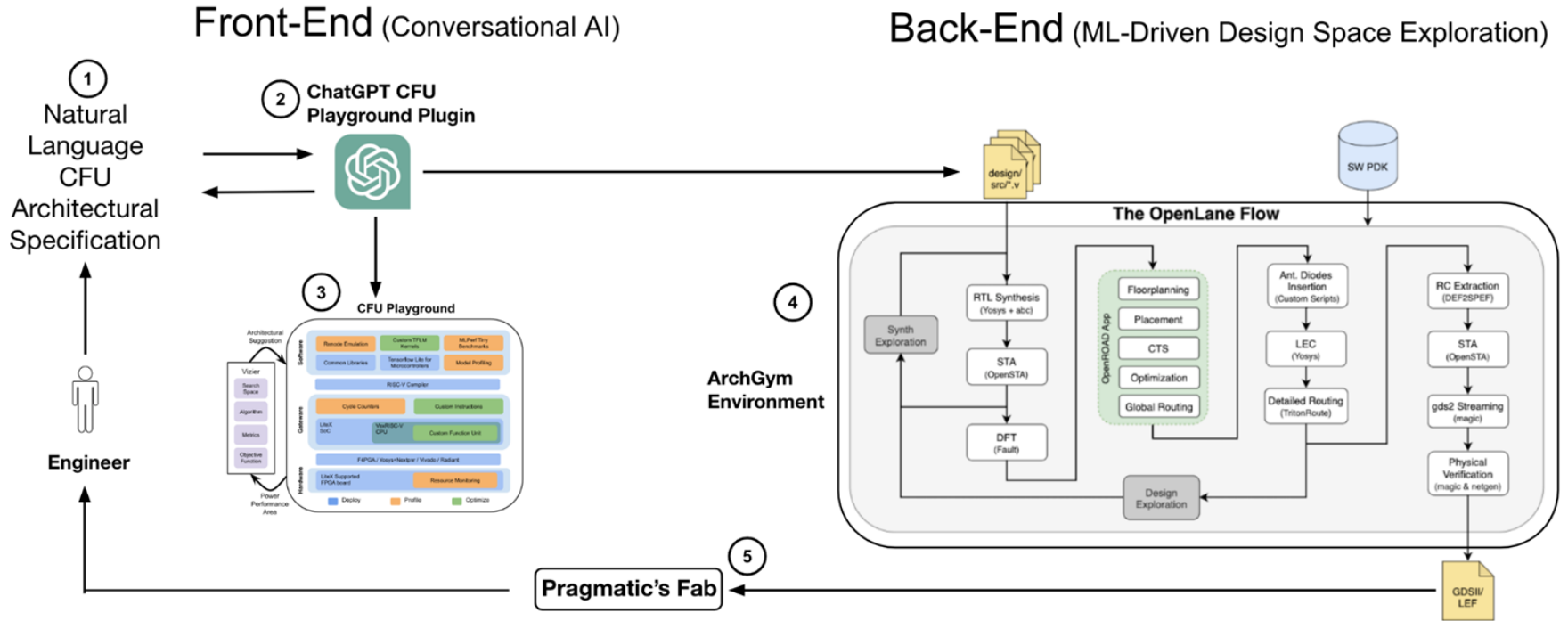
LLMs for CFUs: An LLM powered application for hardware-software co-design of TinyML accelerators

No releases published

Readme | MIT license | Activity | 0 stars | 1 watching | 0 forks

Releases

# End to end Flow



# Solving hard problems needs a **community**



Foster a **collaborative community** with a shared vision of ML and systems researchers



Develop and **share curated datasets** that are representative of diverse workloads across the community



Encourage **data-driven AI research and innovation** for Architecture 2.0

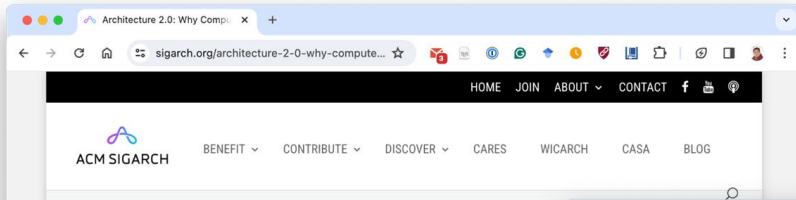


Promote **result replicability** to collectively measure progress & raise the SOTA



Ensure **equitable access** to ML hardware and cutting-edge software technologies

# Learn more about it

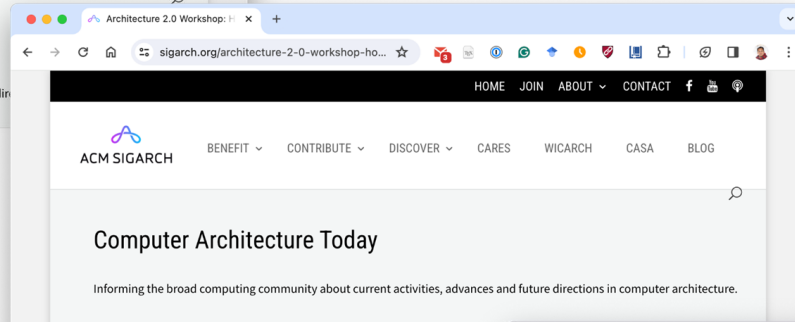
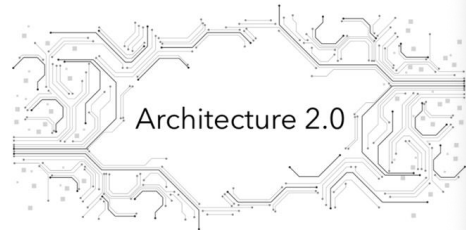


## Computer Architecture Today

Informing the broad computing community about current activities, advances and future directions in computer architecture.

### Architecture 2.0: Why Computer Architects Need a Data-Centric AI Gymnasium

by Vijay Janapa Reddi and Amir Yazdanbakhsh on Jun 14, 2023 | Tags: Benchmarks, Competitions, Datasets, Leaderboards, Machine Learning, ML4Sys, MLSys

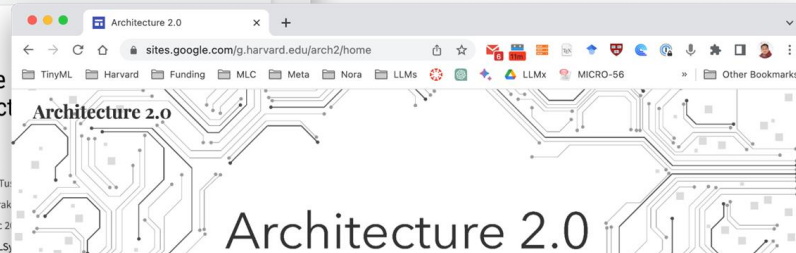


## Computer Architecture Today

Informing the broad computing community about current activities, advances and future directions in computer architecture.

### Architecture 2.0 Workshop: How Machine Learning Will Redefine Computer Architecture and Systems

by Siddharth Garg, Brian Hiran, Jenny Huang, Yingyan (Celine) Lin, Vijay Janapa Reddi, Tu Krishna, Srivatsan Krishnan, Benjamin Lee, Jason Lowe-Power, Martin Maas, Shvetank Prakash, Matthew D. Sinclair, Srinivas Sridharan, Amir Yazdanbakhsh, Jason Yik, Cliff Young on Dec 20, 2023 | Tags: Benchmarks, Competitions, Datasets, Leaderboards, Machine Learning, ML4Sys, MLSys



# Architecture 2.0

# Academic Institutions

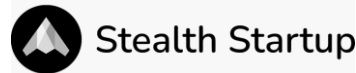
Arizona State University  
BITS Pilani, India  
Cairo University  
Chuka University  
Clemson  
CMU  
College of Engineering kidangoor  
Columbia University  
Cornell University  
Duke University  
George Washington University  
Georgia Tech  
Harvard University  
Indian Institute of Science (IISc)  
ISCTE Instituto Universitário de Lisboa  
LIT  
LUMS  
MIT  
New Mexico State University  
NITK

North Carolina State University  
Northeastern University  
Northwestern University  
NUS  
NYU  
Ozyegin University  
Purdue University  
Shanghai Jiao Tong University  
St.Joseph's Institute of Technology  
The University of British Columbia  
UC Berkeley  
UC Davis  
UC Irvine  
UCSD  
University of Cambridge  
University of Central Florida  
University of Edinburgh  
University Of Houston  
University of Michigan  
University of Padova  
University of Pennsylvania  
University of South Carolina  
University of Toronto  
University of Trento

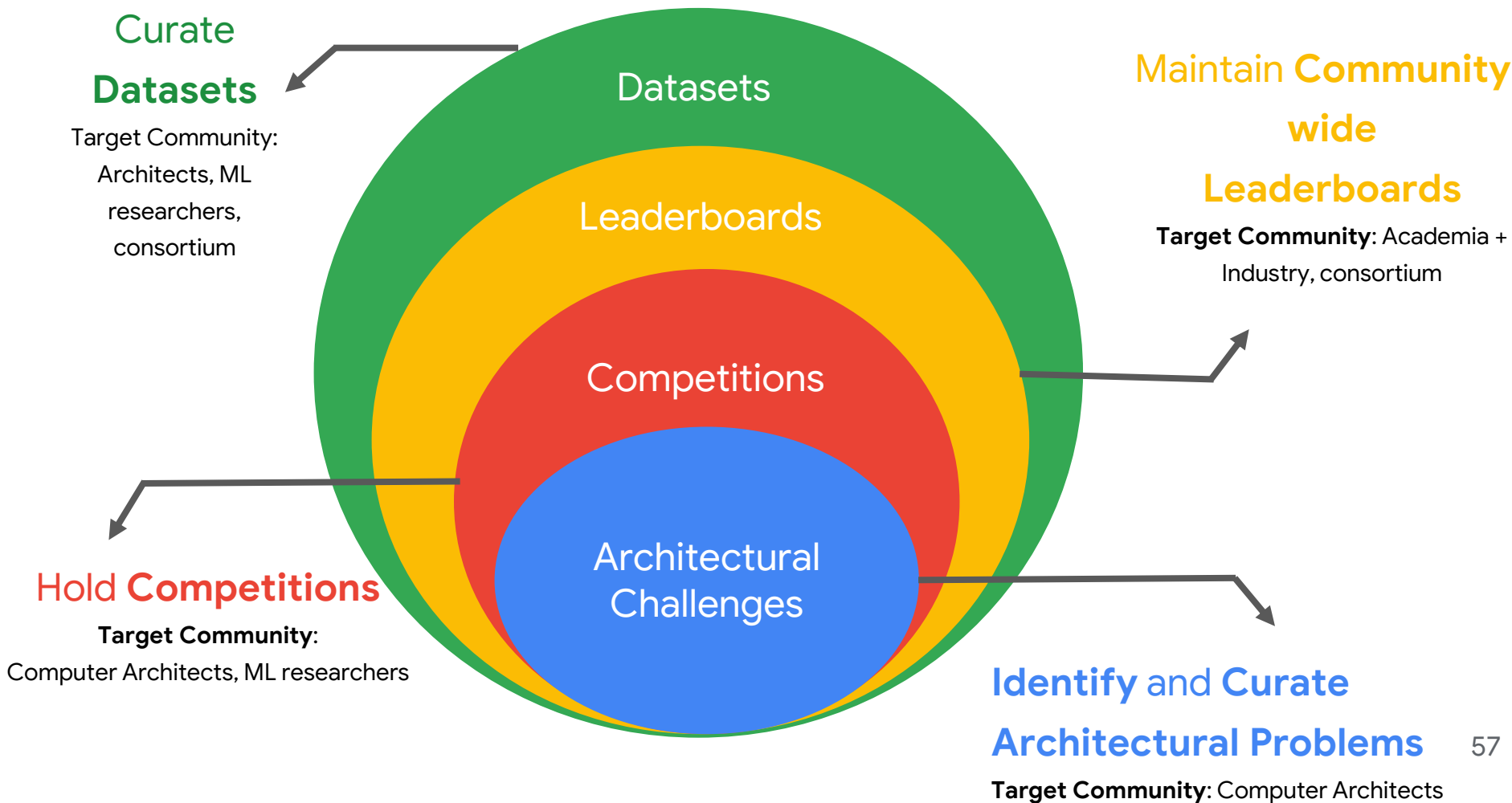
University of Virginia  
University of Washington  
USC  
UT Austin  
UW-Madison  
Virginia Tech  
VJTI (Veer mata Jijabai Technological Institute)  
Washington State University  
Yale University



# Corporations, Startups & Labs









**OPEN**  
Compute Project



Sep 15  
2023

Oct  
2023

Dec  
2023



Whitepaper  
Architecture 2.0

Q1  
2024

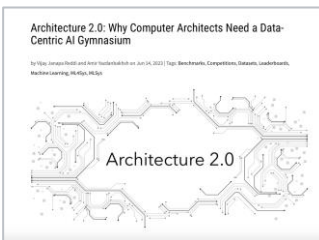


Q2  
2024



Architecture  
2.0

Jun 14  
2023



SIGARCH Blog post

# Acknowledgements





# Thank You!

*Contact: [vj@eecs.harvard.edu](mailto:vj@eecs.harvard.edu)*