NeuroBaseband: An Integrated AI and Wireless Baseband Architecture

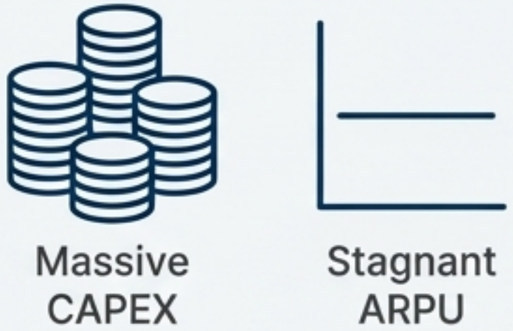
A Technical Report on the Future of Sustainable, Software-Defined Wireless Networks

The Mobile Industry Faces an Economic Impasse

The predictable cycle of generational network upgrades is yielding diminishing returns, creating a crisis of investment and purpose.

User & Operator Disconnect

5G has not delivered the expected returns for consumers, while CAPEX remains massive. Some estimates suggest a 20-year recovery period for 5G investments.



Industry Voice

> *'Nobody needs 6G.'*
>
> — Santiago Tenorio, Director of Network Strategy, Vodafone Group

The 'Odd-Generation Curse'

Without a revolutionary technological driver, there is growing industry skepticism about the need for 6G.

5G investment or **20 years** to recover?

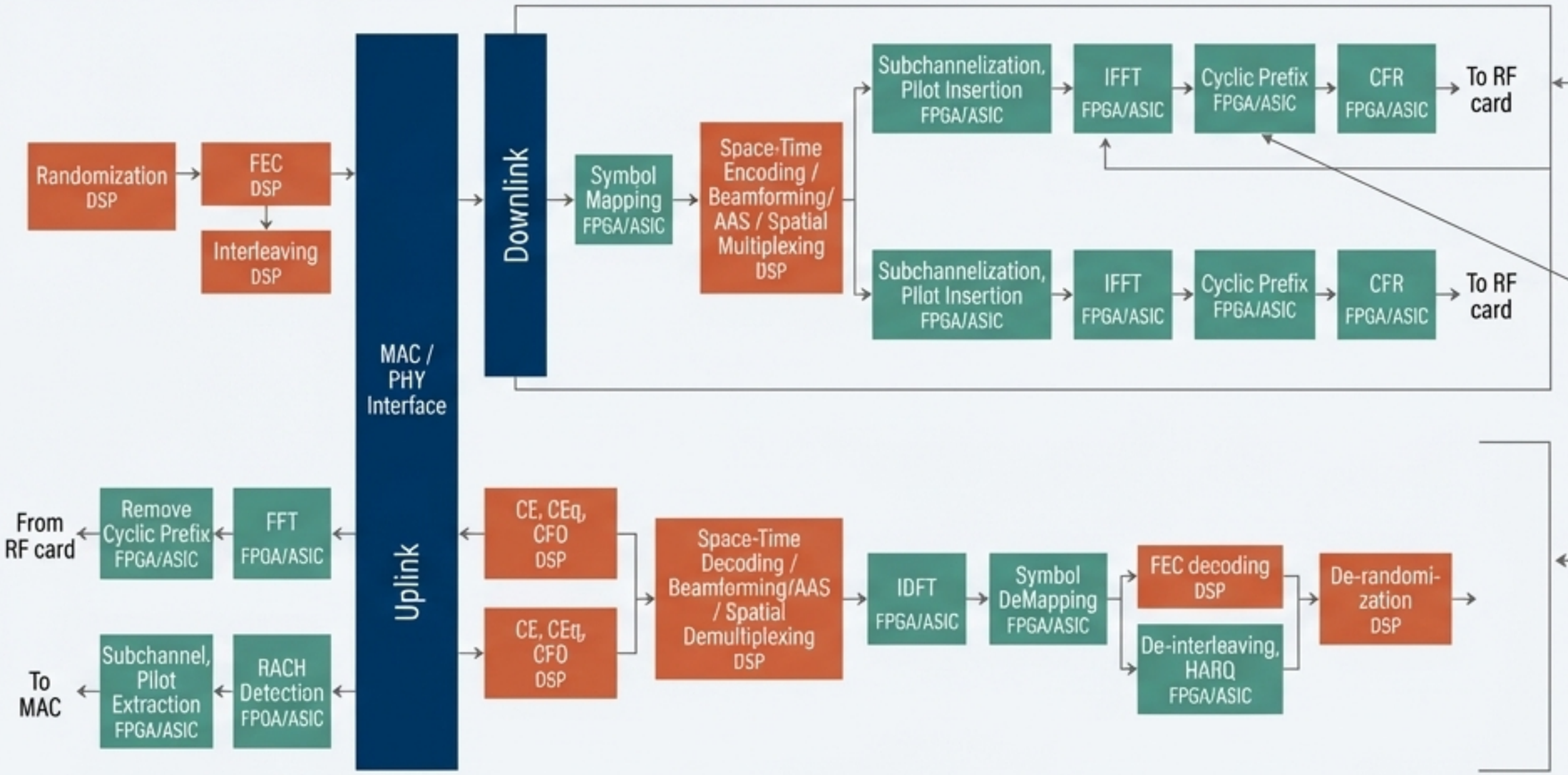
5G costs not yet recovered, is it too hasty to develop 6G?

> *'I hope 5G becomes the best and last generation of mobile communication technology ever, and that we don't need 6G.'*
>
> — Neil McRae, Chief Architect, BT Group

The Core Bottleneck: A Rigid Architecture for a Dynamic Future

The conventional baseband architecture, built on a fixed-function Digital Signal Processor (DSP) and Application-Specific Integrated Circuit (ASIC) / FPGA accelerators, is ill-equipped for the future.

Inherent Limitations: This model lacks the flexibility needed for rapid evolution and cross-generational upgrades, leading to costly and repetitive investment cycles.



Key Weaknesses

Poor Scalability

Tightly coupled hardware cannot scale efficiently with the demands of future standards like 6G.

Limited Software Definition

Protocols are largely baked into silicon, making updates slow and expensive.

Inefficient AI Integration

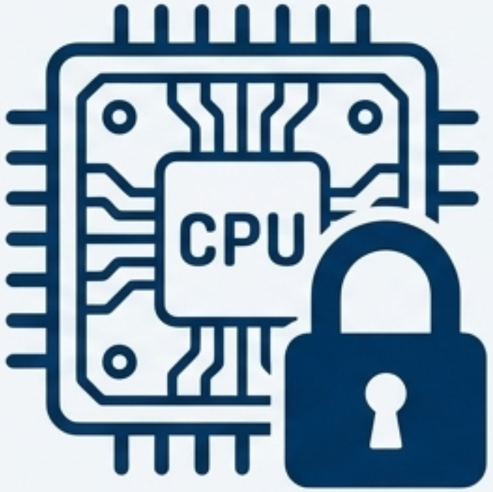
Simply adding an external Neural Processing Unit (NPU) leads to resource waste and data movement bottlenecks. AI needs to be a native, integrated component.

A Confluence of Crises: Supply Chain and Ecosystem Constraints

On top of internal challenges, the entire semiconductor value chain is facing unprecedented constraints, creating strategic vulnerabilities for high-performance chip development.

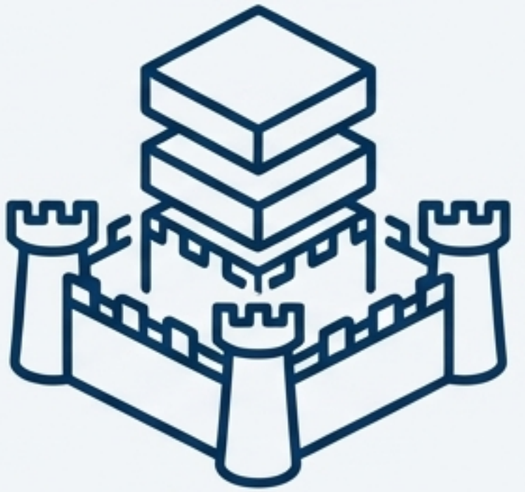
Advanced Manufacturing Restrictions

Access to sub-7nm fabrication processes is increasingly controlled, limiting the ability to produce state-of-the-art chips.



Processor IP Lock-in

Reliance on proprietary processor IP (e.g., ARM) is vulnerable to licensing restrictions and geopolitical pressures.

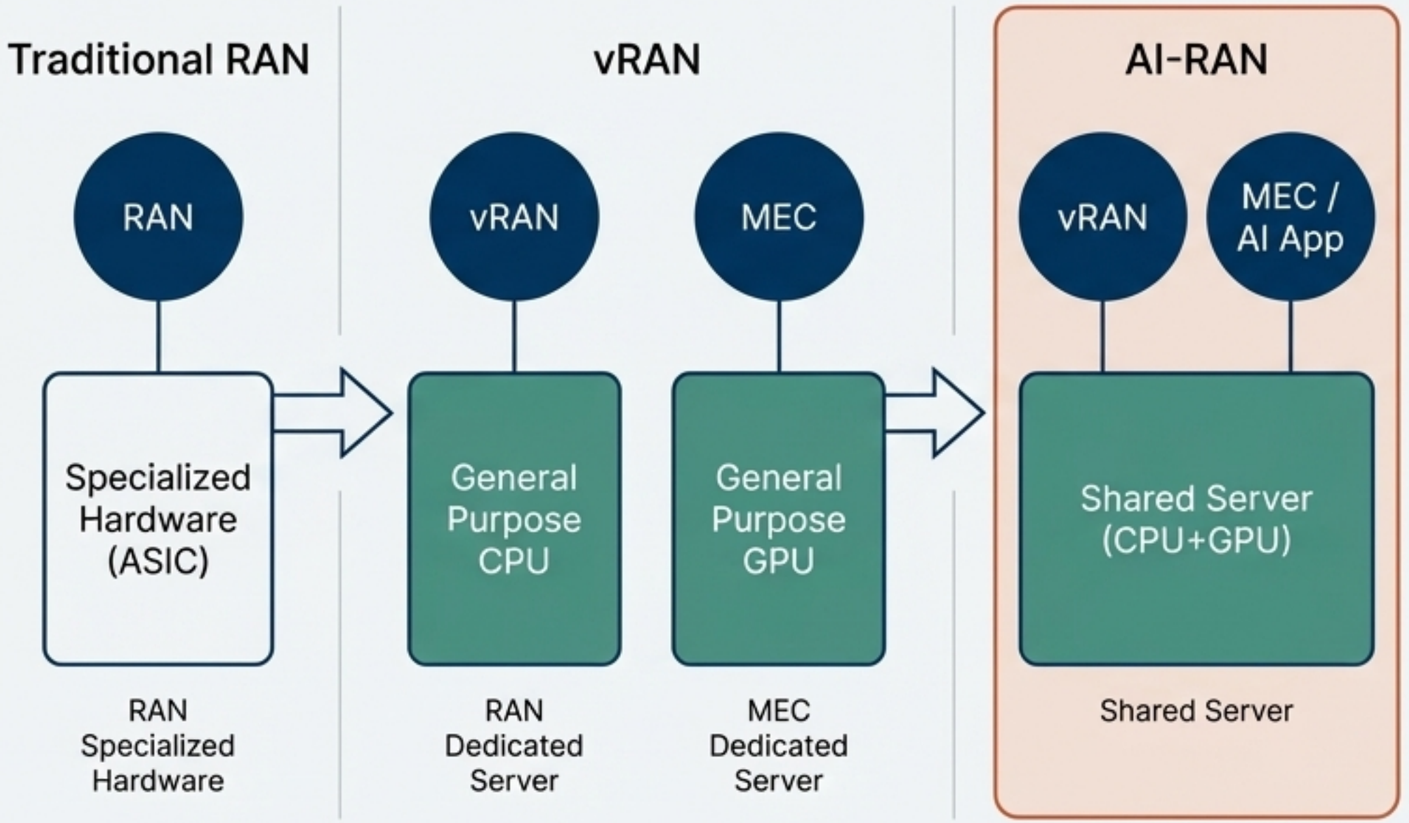


Software Ecosystem Lock-in

Dominant software ecosystems (e.g., NVIDIA's CUDA) create a powerful moat, but also a critical dependency that stifles competition and innovation.

A Fork in the Road: The Rise of the GPU-Based AI-RAN

A new paradigm, AI-RAN, proposes to unify AI and wireless baseband processing on general-purpose GPUs. While innovative, this approach presents a significant strategic risk.



The AI-RAN Proposition

By leveraging the massive compute power of GPUs, this architecture aims to replace specialized hardware with a software-defined solution.

The Hidden Cost

This model replaces one set of dependencies with another, more concentrated one. It consolidates the future of wireless infrastructure on a proprietary, control-flow architecture and its closed software ecosystem.

The Risk of a New Lock-In

The formation of industry alliances around this model threatens to create a new “choke point,” potentially excluding and marginalizing alternative approaches.

The Third Way: A GPU-Like Architecture, Not a GPU

NeuroBaseband is a fundamental re-imagination of the baseband processor. It delivers the high parallelism and software-defined nature of a GPU, but is built from the ground up as a domain-specific architecture (DSA) for integrated AI and communications on an open RISC-V foundation.

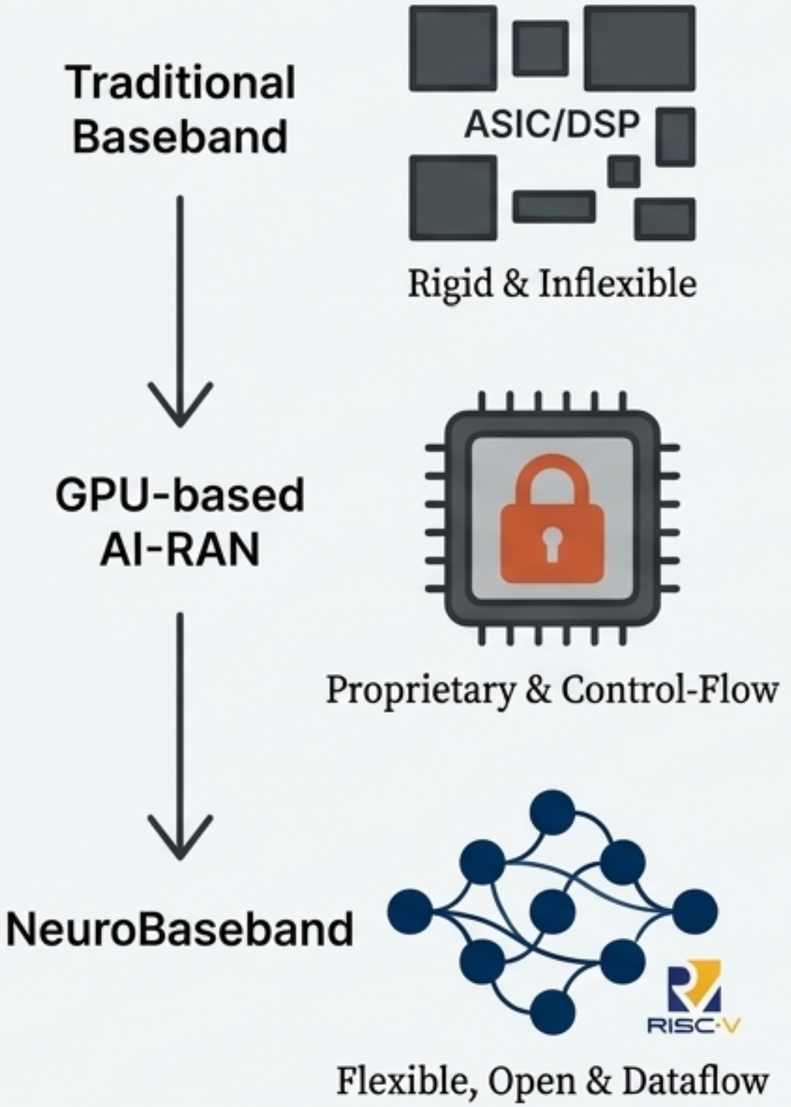
Our Goal

To create a sustainable, evolvable, and open architecture that breaks free from the limitations of both legacy systems and new proprietary ecosystems.

The Core Idea

Reconstruct the baseband processor by replacing the closed DSP + custom accelerator model with an architecture that is:

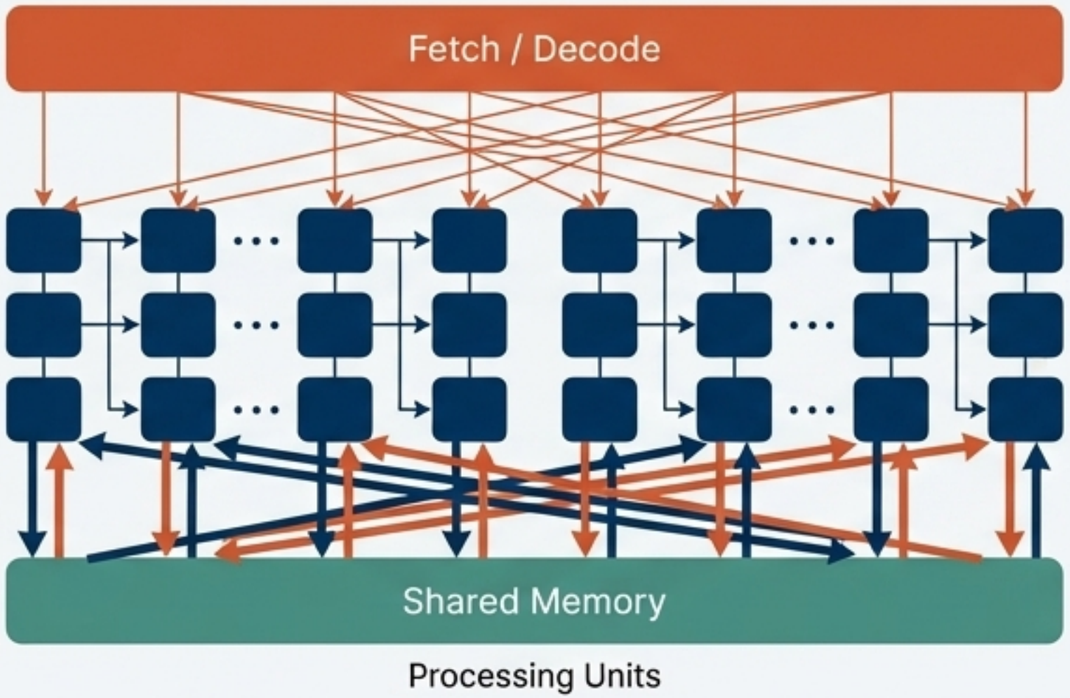
1. **GPU-Like:** High parallelism, domain-specific, and supports an open ecosystem.
2. **Not a GPU:** Employs a communications-first DSA and is based on the open RISC-V standard.



Unlocking Efficiency with a Dataflow-Driven Paradigm

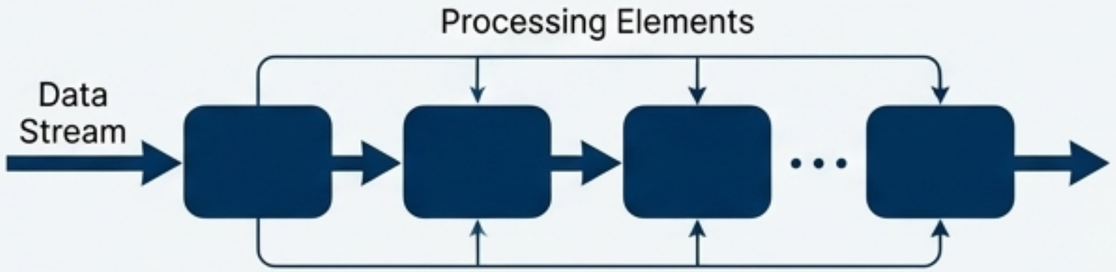
The core inefficiency of using general-purpose architectures like GPUs for baseband processing lies in their control-flow model. NeuroBaseband adopts a fundamentally more efficient dataflow-driven approach.

Control-Flow (GPU Model)



Complex front-end units fetch, decode, and schedule instructions. Data is constantly moved between memory and processing units, creating an energy and latency bottleneck.

Dataflow (NeuroBaseband Model)



Computation happens where the data resides. The architecture is organized around the natural flow of data, minimizing movement, reducing control overhead, and dramatically improving power efficiency.

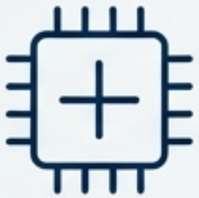
Built on an Open and Sovereign Foundation: RISC-V

To ensure long-term sustainability and avoid ecosystem lock-in, the NeuroBaseband architecture is built entirely on the open RISC-V instruction set architecture (ISA).

RISC-V®

Freedom from Licensing

RISC-V is a free and open standard, eliminating licensing fees and restrictions associated with proprietary ISAs like ARM or x86.



Full Extensibility for Domain Specialization

The modular nature of RISC-V allows for the creation of custom instruction extensions. We have developed specific extensions tailored for the core operators in both wireless communications and AI.

A Sovereign and Collaborative Ecosystem

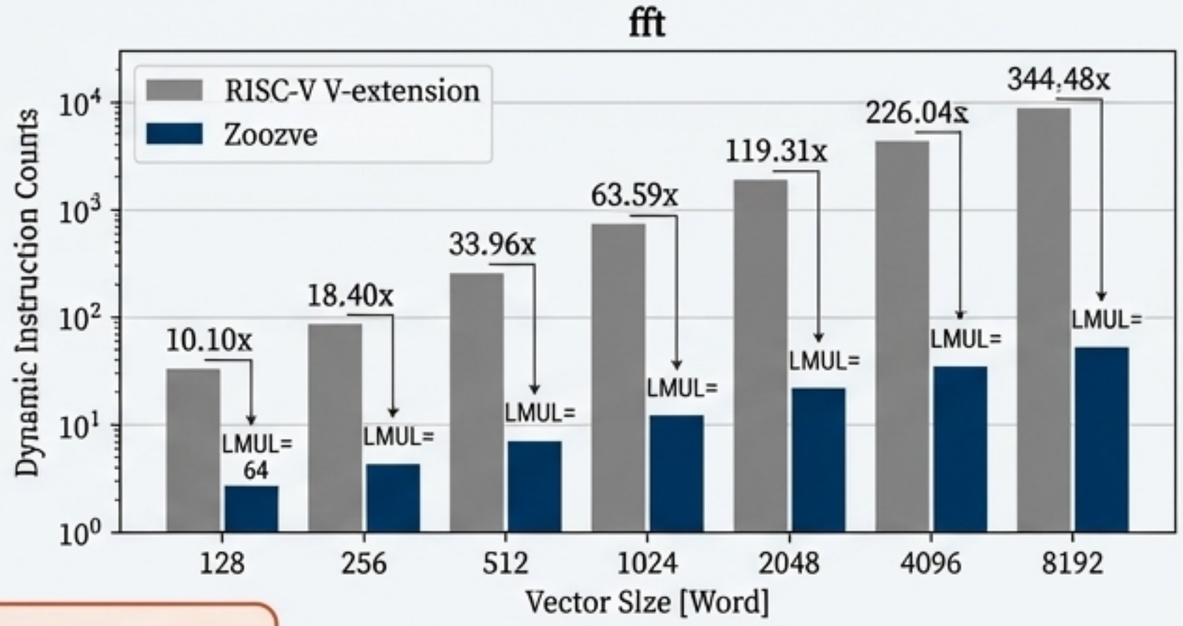
Building on RISC-V fosters a domestic and global ecosystem of tools, talent, and shared innovation, mitigating supply chain risks.

Beyond Limits: Eliminating the “Strip-Mining” Bottleneck

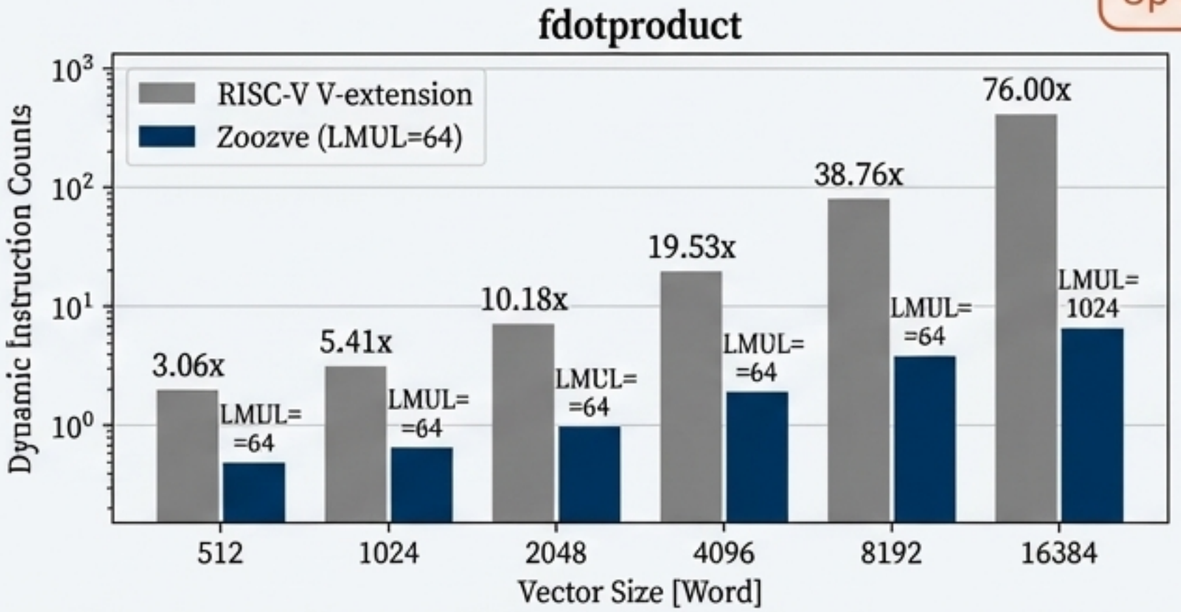
Traditional vector processors have a fixed maximum vector length. When processing data streams larger than this limit, the compiler must insert extra code to break the stream into chunks—a process called “strip-mining” which introduces significant overhead.

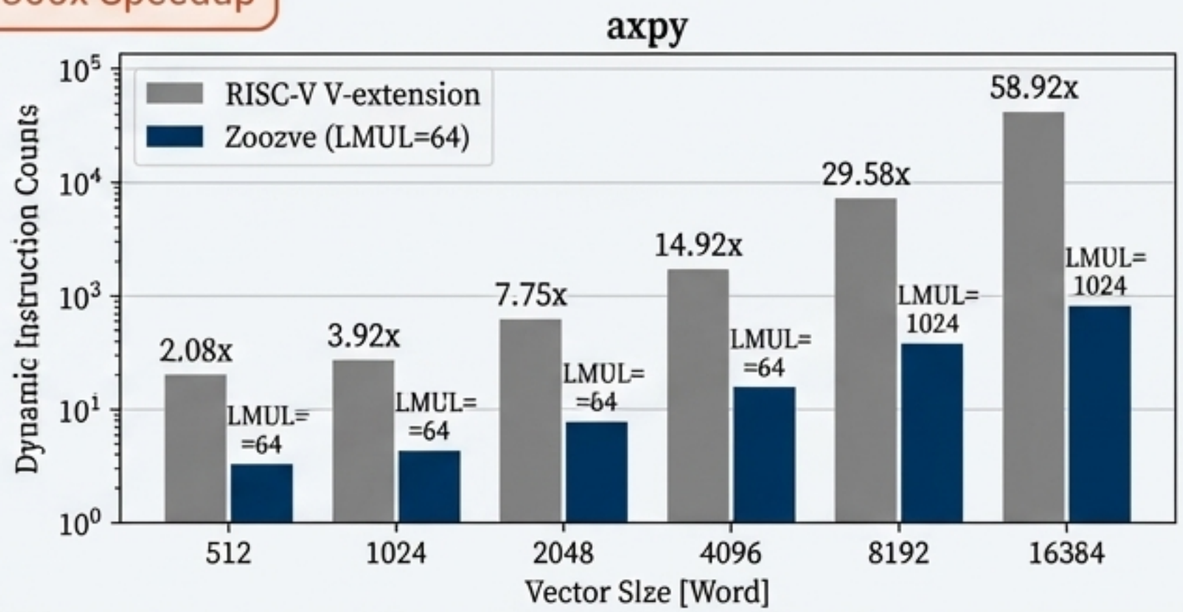
The NeuroBaseband Advantage (Zoozve Extension)

Our architecture supports variable-length dataflow with no upper limit on stream length, completely eliminating the need for strip-mining and massively reducing dynamic instruction counts.



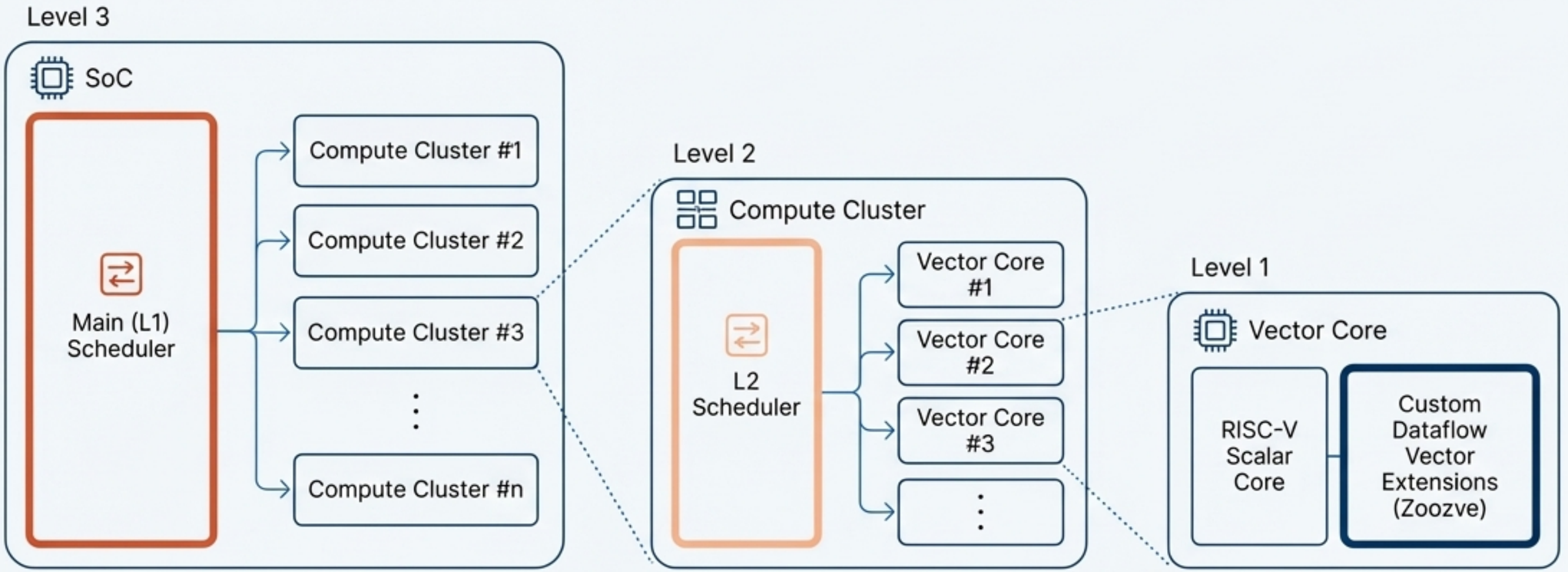
Performance Gain:
Up to 300x Speedup





An Architecture of Systems: From Core to Cluster to Chip

The NeuroBaseband architecture is a hierarchical system designed for massive parallelism and efficient data distribution, managed by a two-level dataflow scheduling mechanism.



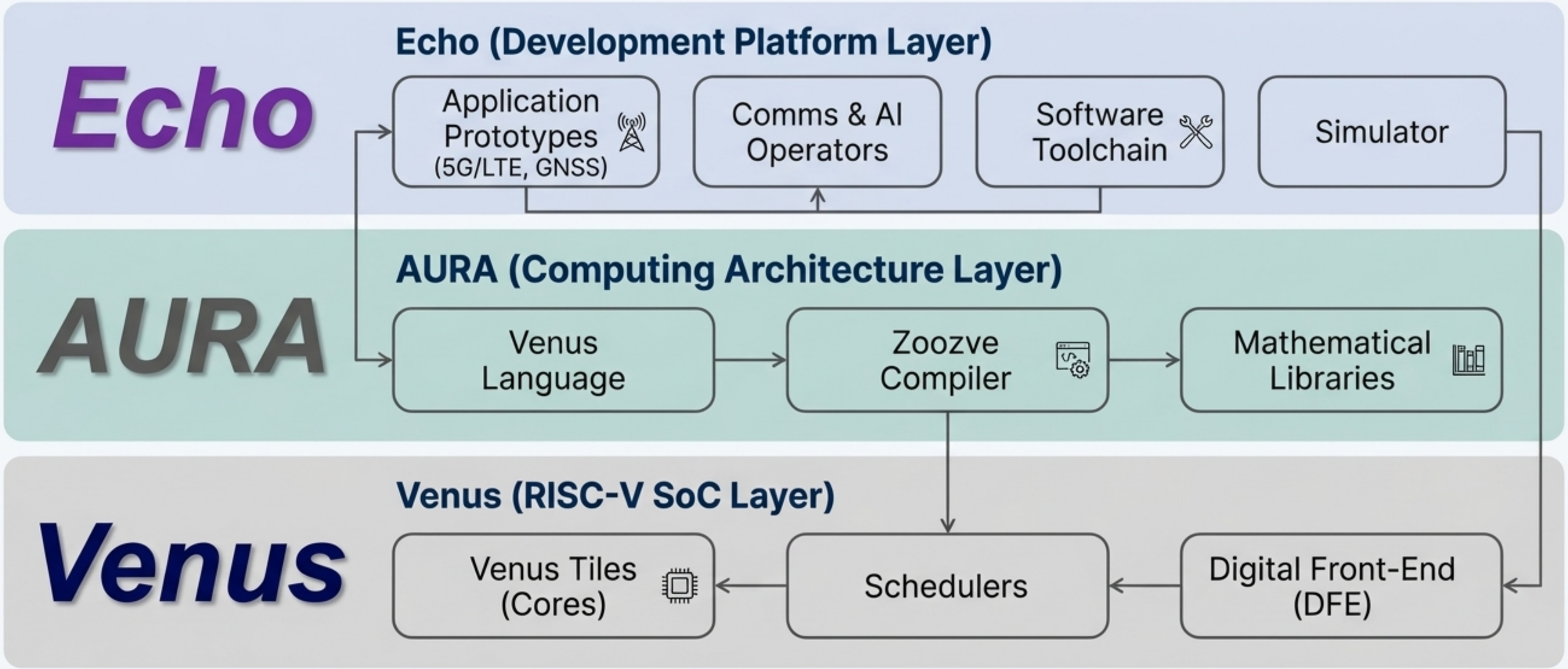
The full chip with a main (L1) scheduler managing coarse-grained dataflow between clusters.

A cluster of cores managed by a task-level L2 scheduler.

The fundamental building block, combining a standard scalar core with our custom extensions.

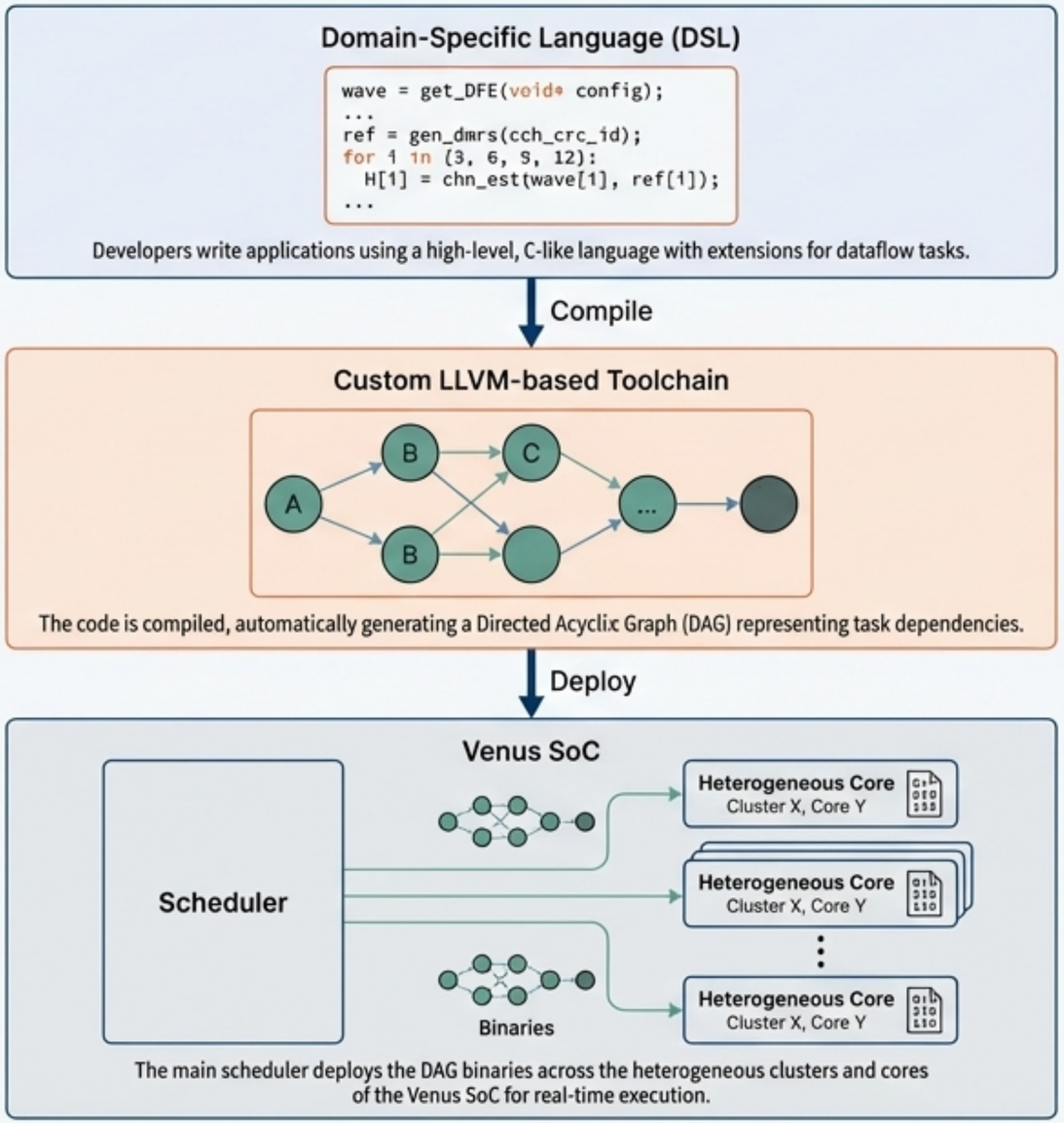
The Echo Platform: A Full Stack for Integrated AI & Comms

A powerful hardware architecture requires a comprehensive and accessible software platform. We have built the Echo ecosystem, an open-source development platform for Communication-AI (ComAI) applications.



A Unified Workflow: From High-Level Code to Hardware Execution

The Echo platform provides a streamlined workflow that allows developers to describe complex signal processing chains in a high-level language and compile them efficiently to our heterogeneous hardware.



Validated Performance: 51x Speedup in 5G NR Processing

To validate the architecture, we implemented a 5G New Radio (NR) physical layer processing chain and benchmarked it against an ARM Cortex-A9 baseline on the same FPGA platform. The results demonstrate a dramatic improvement in both performance and latency.

51.57x
Speedup vs. Baseline

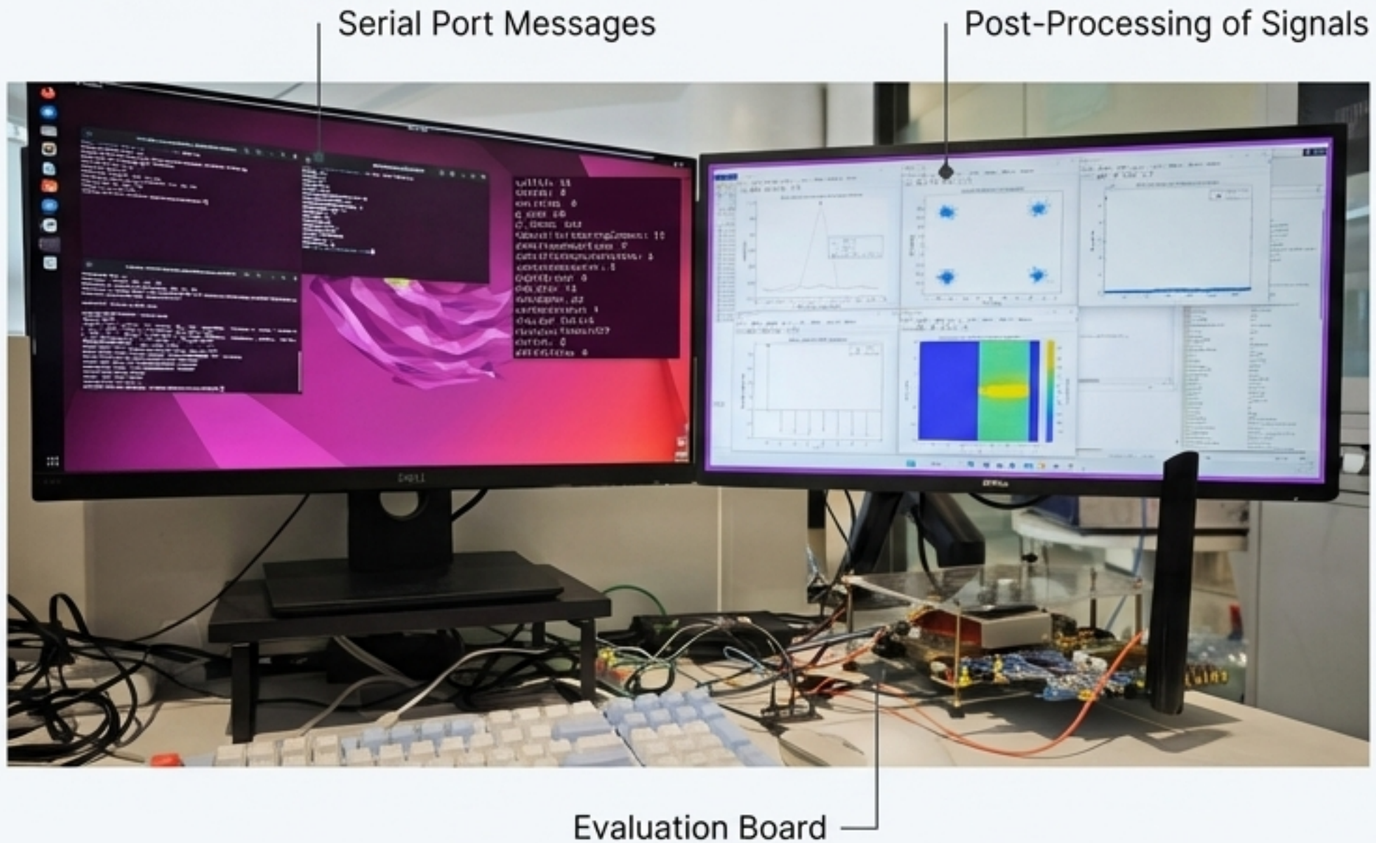
Work	Platform	Standard	SW Friendly	Latency (ms)	Speedup
Baseline	Zynq	NR	No	284.15	1.00x
Venusian	**FPGA**	**NR**	**Yes**	**5.51**	**51.57x**
[13]	Intel CPU	LTE	No	184.34	1.54x
[14]	Nvidia GPU	NR	Yes	1612.28	0.18x

284ms → 5.51ms
Latency Reduction

Unlike many hardware-accelerated solutions, our architecture is fully programmable and software-friendly. Compared to a high-performance Intel CPU or NVIDIA GPU, our solution provides a superior combination of performance, efficiency, and programmability for this domain.

From Theory to Reality: Deployed Prototypes and Applications

The NeuroBaseband architecture and Echo development platform have been successfully implemented and deployed in multiple real-world communication systems.



Flagship Demo: End-to-End 5G Cell Search

A complete 5G cell search demonstration running on our Venus hardware, successfully receiving off-air signals, performing synchronization, and decoding the Master Information Block (MIB).

Proven in the Field

- LTE-V2X vehicle-to-everything communications
- Private networks for power grid management
- Low-altitude sensing and communications

NeuroBaseband: A Sustainable, Open, and High-Performance Future for Wireless

We have presented a new path forward for wireless evolution, designed to address the core technical, economic, and strategic challenges facing the industry.

The Challenge

The mobile industry is at an impasse, facing unsustainable economics and a rigid baseband architecture. This is compounded by strategic supply chain risks and the threat of a new, proprietary ecosystem lock-in.

The Solution

A fundamental re-architecture of the baseband processor. NeuroBaseband is a dataflow-driven, domain-specific architecture that natively integrates AI and wireless processing on an open RISC-V foundation.

The Impact

This approach enables a sustainable path for network evolution with drastically reduced development cycles (from 18 to 6 months). The open-source Echo platform provides the foundation for a sovereign and collaborative technology ecosystem.

Join the Open-Source Community

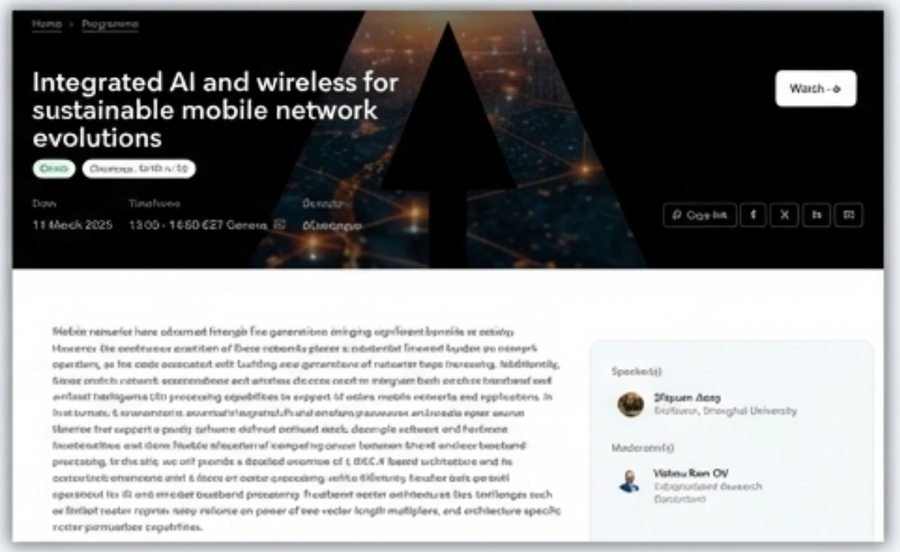
We invite you to explore, contribute to, and build upon the Echo platform.

GitHub: github.com/ACELab-SHU/ACE-Echo Website: acelab-shu.github.io/ACE-Echo

An Open Architecture for a Sustainable 6G, Recognized Globally

International Recognition

Our work on a sustainable, open-source 6G architecture and domain-specific instruction set was selected as a global best-use case at the **ITU's AI for Good Global Summit**.



Summary

The Challenge

The mobile industry faces a dual crisis of unsustainable economics and strategic supply chain risks. Legacy architectures are the bottleneck.

The Solution

NeuroBaseband offers a new paradigm: a scalable, software-defined, AI-native baseband architecture built on an open RISC-V foundation.

The Enabler

The Echo open-source platform provides the tools for the entire community to build upon this architecture, creating a collaborative and sustainable ecosystem for 6G and beyond.

Appendix: Programming on the Venus Architecture

Application development is a three-step process: writing tasks, defining dependencies, and scheduling execution.

Step 1: Write tasks using the C-like Venus Language.

Introduces a new `vector` variable type and custom vector instructions (e.g., `vmul`, `vadd`, `vshuffle`).

```
#include "riscv_printf.h"
#include "venus.h"

short rxSignalLength = 432;
short softBitLength = 864;

int Task_nrPBCHDemodulate(__v4096i8 inSignal_real, __v40
  /*--------------------QPSK Demodulate----------------
  __v4096i8 softbit;
  __v2048i16 softbit_shuffle_index_tmp;
  vclaim(softbit_shuffle_index_tmp);
  vclaim(softbit);
  vrange(softbit_shuffle_index_tmp, rxSignalLength);
  softbit_shuffle_index_tmp = vmul(softbit_shuffle_index
  vshuffle(softbit, softbit_shuffle_index_tmp, inSignal_
  softbit_shuffle_index_tmp = vsadd(softbit_shuffle_inde
  softbit_shuffle_index_tmp = vsadd(softbit_shuffle_inde
  vshuffle(softbit, softbit_shuffle_index_tmp, inSignal_
  vreturn(softbit, softBit1Length);
}
```

Step 2: Connect tasks into a Directed Acyclic Graph (DAG).

A simple text file (`.bas`) defines the tasks and their data dependencies.

```
dag dag1 = {
  [ncellid] = Task_SSS_Search(d_SSS0, d_SSS1, rxData_rea
  [dmrs_index] = Task_nrPBCHDMRSIndices(init_dmrs_index,
  [iBar_SSB] = Task_iBar_SSB_Search(seq1, seq2_init_tabl
  seq2_init_table_3,  seq2_init_table_4, seq2_init_table
  seq2_init_table_9,  seq2_init_table_10, seq2_init_tabl
  seq2_init_table_14, seq2_init_table_15, seq2_init_tabl
  seq2_init_table_19, seq2_trans_table_0, seq2_trans_tab
  seq2_trans_table_4, seq2_trans_table_5, seq2_trans_tab
  rxData_shuffle_index)
  [dmrs_real, dmrs_imag] = Task_nrPBCHDMRS(seq1, seq2_in
  seq2_init_table_3, seq2_init_table_4, seq2_init_table_
  seq2_init_table_9, seq2_init_table_10, seq2_init_table
  seq2_init_table_14, seq2_init_table_15, seq2_init_tabl
  seq2_init_table_19, seq2_trans_table_0, seq2_trans_tab
  seq2_trans_table_4, seq2_trans_table_5, seq2_trans_tab
  [h_real, h_imag] = Task_nrPBCHChannelEstimate(rxData_r
  global_coef_h, rxData_shuffle_index)
  [pbch_index] = Task_nrPBCHIndices(init_dmrs_index, nce
  [pbchEq_real, pbchEq_imag, csi] = Task_nrPBCHEqualize(
  rxData_shuffle_index)
```

Step 3: Write the L1 scheduler to manage DAG execution.

The L1 scheduler controls task triggering, hardware interrupts, and data flow with the Digital Front-End (DFE).

```
dfe_init();

// BO帧号
if (fifo_size(&rfdata_init) > 1) {
  fire_dag(pbch, 2, 2, &rfdata_init, &cellid_s, &ssSlot_
};
dag_fence();

// 调整DFE时偏
timer_offset = FRONT_READ(DFE_REG(slot_timer)) + symbol;
CONFIG_DFE_REG(slot_timer, TINER_CONFIG(timer_offset));

// CO帧号
rfdata_cch.frame[0] = RULES;
rfdata_cch.frame[1] = IS_EVEN(pdcchFrame) ? EVEN_ID; FRA
rfdata_cch.slot[0] = SPEC_NUM;
rfdata_cch.slot[1] = ssSlot + NSlot;
if (fifo_size(&rfdata_cch) > 2) {
  fire_dag(pdcch, 2, 1, &rfdata_cch, &rfdata_cch, &crc_s
}
dag_fence();
```