A Technical Analysis of the System76 Lemur Pro: Architecture, Design Decisions, and Linux-Oriented Engineering

Corbin Hibler, *Ohio Northern University*

***Abstract*—In this report, I analyze the internal architecture and engineering decisions behind the System76 Lemur Pro, a lightweight Linux laptop built around open-source firmware and power-efficient hardware. I examined each subsystem—CPU, memory, storage, firmware, and power delivery—to understand how they interact and why certain trade-offs were made. What I found is that the Lemur Pro's design centers around long battery life, maintainability, and deep Linux integration, supported by Coreboot-based firmware and user-replaceable components. Through hands-on evaluation and comparison with manufacturer documentation, I show how the Lemur Pro achieves a balance between portability and performance while emphasizing transparency and sustainability. This analysis highlights the advantages of open firmware and modular design in modern mobile computing systems.**

***Index Terms*—open firmware, Coreboot, embedded controller, Linux, power-efficient computing, NVMe storage, Intel U-series processors, system architecture, modular hardware design.**

I. Introduction

When I started working with the System76 Lemur Pro, I was immediately interested not just in how lightweight and efficient it felt, but *why* it behaved the way it did. I've always enjoyed understanding hardware at a deeper level, so I took this assignment as an opportunity to analyze the system from the inside out. What I found is that the Lemur Pro is designed around a very deliberate set of engineering choices: open firmware, power-efficient Intel processors, user-replaceable components, and a battery system tuned for long runtime on Linux. In this report, I walk through the device's architecture, the components that drive it, and the reasoning behind its design.

II. Device Overview

The Lemur Pro is System76's ultralight Linux laptop, optimized for their own in-house Linux distribution, Pop!_OS. Unlike many consumer laptops, the Lemur Pro uses an open firmware stack built on Coreboot, along with System76's own embedded controller firmware [1]. This immediately caught my attention because it shifts control from proprietary BIOS code to something far more transparent and configurable.

The laptop weighs under 2.2 pounds, yet routinely reaches over 14 hours of real-world battery life in my experience. Internally, it relies on Intel's U-series processors, NVMe storage, and upgradable RAM. As someone who works heavily with Linux, development tools, and open-source frameworks, this type of system aligns closely with my workflow [2].

III. CPU and Processing Architecture

The first thing I analyzed was the processor family used in the Lemur Pro. System76 typically pairs it with Intel i5 or i7 U-series mobile processors. These CPUs operate at around **10–15W TDP**, which explains the impressive battery life I've observed [3].

The integrated Iris Xe graphics reinforce this efficiency-driven approach. Instead of dedicating power and thermal budget to a discrete GPU, the system leans on Intel's unified graphics architecture. For many of my development tasks—compilation, virtualization, database work—the graphics subsystem isn't the bottleneck, so this trade-off makes sense [3].

What stood out to me was how quickly the CPU scales frequency while staying cool. Intel's SpeedShift technology allows near-instantaneous clock adjustments, which I noticed when alternating between idle tasks and heavier workloads like building software or running containers [4].

IV. Memory System

Something I really appreciated was that the Lemur Pro keeps its **RAM user-replaceable**. After spending a lot of time dealing with soldered laptop memory that locks you into a single configuration, seeing SO-DIMM slots is a fantastic choice.

The model I examined supported up to **40GB** or **64GB** of RAM depending on the generation. This extendable memory system is extremely helpful when I work on projects involving dense graph datasets or multiple virtual machines. Dual-channel memory also bumps overall bandwidth, which I noticed especially when multitasking during development sessions [2].

V. Storage Architecture

For storage, the Lemur Pro uses an M.2 2280 NVMe SSD on a PCIe x4 interface. NVMe's parallel command queues make a noticeable difference during tasks like compiling large projects or running containerized workflows. The SSD in my unit consistently produced fast load times and low latency, which improved the responsiveness of database workloads and large file operations [2].

I also appreciated that the SSD is user-replaceable. This lets me expand storage or opt for a higher-end drive without replacing the entire machine—an approach that extends the laptop's lifespan significantly.

VI. Display and Input System

The Lemur Pro's **14-inch 1080p IPS display** is tuned more for battery life than for absolute color performance, but it performs well for everyday development work [2]. IPS technology provides consistent viewing angles, and I never ran into visual clarity issues while coding or doing academic writing.

The keyboard and trackpad work smoothly with Linux's libinput driver. I've used many Linux laptops where input devices felt like afterthoughts, so the out-of-box reliability here stood out.

VII. Battery and Power Efficiency

One of the most impressive components is the **73 Wh battery** found in many Lemur Pro revisions [2]. Combined with the low-power CPU, the system delivers all-day battery life under realistic workloads—not just idealized spec sheet claims.

I ran several personal tests during normal usage: writing, browsing, and running lightweight development tasks. The battery consistently outperformed mainstream Windows ultrabooks I had used previously. Power management worked smoothly under Pop!_OS, which isn't surprising given System76's tight integration with Linux.

VIII. Firmware and Embedded Controller

This was the part of the device I spent the most time investigating.

System76 replaces proprietary BIOS code with a **Coreboot-based open firmware**, paired with their custom open firmware, embedded controller firmware, and firmware manager [1], [2].

I found this architecture especially interesting because it breaks away from the opaque, vendor-locked firmware stacks I usually see on laptops. Coreboot initializes hardware much faster than a traditional BIOS, and the embedded controller handles everything from fan control to keyboard input to battery calibration.

What this means in practice is better transparency and finer control over the system's behavior. I also appreciate that firmware updates are delivered directly through Pop!_OS with fully auditable source code—something almost no mainstream manufacturer provides.

IX. System Integration and Data Flow

Walking through the boot process helped me understand how the components interact. First, the embedded controller powers on and performs early hardware checks. Then, Coreboot initializes the CPU, memory, and storage extremely quickly. After that the firmware hands control off to the Linux bootloader. During operation, the OS communicates with hardware through primarily open-source drivers [1].

From a system design perspective, the laptop prioritizes *coherence and maintainability* over unnecessary complexity. The CPU, RAM, and SSD form a tightly integrated low-latency pipeline, with the firmware acting as a streamlined facilitator rather than a roadblock.

X. Engineering Trade-offs

A. Portability vs. GPU performance

The absence of a discrete GPU is intentional. The laptop wasn't built for AAA gaming or ML workloads, and including a GPU would undermine its runtime and size. I made this a personal choice, as I've found that integrated graphics are more than adequate for everything I do in my workflow [4].

B. Repairability vs. Chassis thinness

Component accessibility slightly increases internal volume, but I view it as a positive trade-off. Being able to replace RAM, SSDs, and the battery helps keep the device viable for years. This is a design philosophy I wish more manufacturers followed [4].

C. Open firmware vs. proprietary features

By using open firmware, System76 sacrifices some proprietary Intel features, but gains transparency, security, and user control. Personally, I prefer this direction, especially for a Linux-focused machine.

D. Use Cases and Practical Value

Throughout my research and real-world usage, I've found that the Lemur Pro excels in software development, with a fast NVMe, large RAM capacity, strong Linux integration. It also excels in mobile college work, with long battery life for writing, reading, and coding on the go. Lastly, it excels in open-source and privacy-focused workflows, with full firmware visibility and customization.

The laptop's design consistently reinforces these goals, and I've seen tangible benefits in my daily development tasks.

XI. Conclusion

Analyzing the System76 Lemur Pro showed me how intentional design can create a balanced, efficient, and open machine. The device combines low-power Intel processors, modular components, Coreboot-based firmware, and excellent Linux compatibility. The engineering trade-offs—especially around battery efficiency, repairability, and openness—align with the needs of developers, students, and professionals who rely on Linux.

From my experience and research, the Lemur Pro stands out as one of the few laptops where the internal architecture genuinely reflects the philosophy behind it: transparent, customizable, and optimized for long-term usability. It is absolutely my favorite device that I own, and hope to own more devices that follow this philosophy.

[1] "Coreboot for end users," Accessed: Dec. 12, 2025. [Online]. Available: https://www.coreboot.org/end_users.html.

[2] "Lemur Pro (lemp12) - System76 Technical Documentation," Accessed: Dec. 12, 2025. [Online]. Available: https://tech-docs.system76.com/models/lemp12/README.html.

[3] “Intel® Core™ Ultra 7 Processor 155U (12M Cache, up to 4.80 GHz) - Product Specifications,” Intel, Accessed: Dec. 12, 2025. [Online]. Available: https://www.intel.com/content/www/us/en/products/sku/237327/intel-core-ultra-7-processor-155u-12m-cache-up-to-4-80-ghz/specifications.html.

[4] OpenAI. “Analysis of Intel 155U Processors,” ChatGPT.