White Paper

Rebellions' Software Stack

: Silent Support

rebellions_

Contents

Introduction

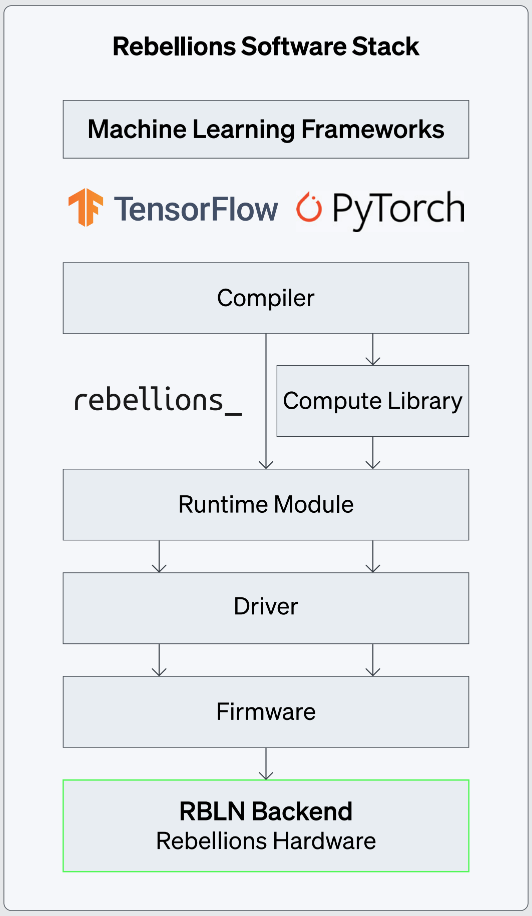
Amidst the surge in Artificial Intelligence (AI) spanning diverse fields of application, the demand for specialized hardware optimized for low latency and high energy efficiency is expected to grow. AI accelerators, also known as AI chips, are engineered to accelerate AI algorithms, executing complex computations with greater speed and lower power consumption than traditional CPUs and GPUs.

Rebellions' ATOM™ is a SoC (System-on-Chip) specifically tailored to optimize the processing of deep learning models. With eight powerful Neural Engines executing computations within an elaborately designed memory architecture, ATOM™ is optimized for high performance and low latency. However, the potential of this advanced hardware remains unrealized without equally optimized software to elevate its performance to the highest possible level.

Software is the driving force that ensures our hardware operates at its peak to deliver maximum performance. Software plays a crucial role in managing resources, optimizing data flow, and executing algorithms efficiently. For users migrating from GPUs, our software provides a seamless experience, minimizing compatibility issues and ensuring smooth integration. Our Software Stack puts special emphasis on ease of use and reliability, providing a comprehensive user documentation and a robust SDK, making it accessible and user-friendly for developers and engineers.

In this document, we introduce the key components and features of Rebellions' Software Stack. Highlighting how it supports ATOM™ in achieving high performance with drastically reduced power consumption. Additionally, we present a performance comparison of the YOLOv6-Large object detection model against NVIDIA's RTX A5000, demonstrating our commitment to delivering powerful, yet more energy-efficient, full-stack solutions.

Rebellions' Software Stack



[Figure 1. Rebellions' Software Stack]

While ATOM™ provides the raw computational power for the complex AI workloads, Rebellions' Software Stack helps optimize the chip's architectural advantages, tailoring the execution of models to harness the full potential of ATOM™.

RBLN SDK includes Rebellions' proprietary Compiler, Compute Library, Runtime, Drivers, and Firmware, engineered to facilitate seamless integration of pre-trained models from frameworks such as TensorFlow, PyTorch or the Hugging Face ecosystem into Rebellions' chip-based serving environment. All the components in the Software Stack serve to render the lowest latency possible.

Framework Support

With a growing list of over 200 reference models and operations for popular frameworks such as TensorFlow, PyTorch and optimized support of Hugging Face models, RBLN SDK guarantees seamless experience for developers to migrate to Rebellions' chips, allowing prominent large language and diffusion models, as well as popular vision and speech models to run smoothly.

RBLN Compiler

The RBLN Compiler transforms models into executable instructions for ATOM^{TM}. It comprises two main components: the Frontend Compiler and the Backend Compiler. The Frontend Compiler abstracts deep learning models into Intermediate Representations (IRs), optimizing them before handing them off to the Backend Compiler. The Backend Compiler further optimizes these IRs and produces the Command Stream, the Program Binary for the hardware to execute the tasks, and serialized weights.

- Frontend Compiler

The Frontend Compiler converts the models into unified graph IRs. The nodes in the IRs are then optimized before being offloaded to the Backend Compiler, which involves removing redundant functions and nodes in the IRs, binding the weights to matched nodes, merging the nodes to reduce data/kernel load/store overhead, annotating the nodes that are ready to be transferred, and partitioning the graphs for efficient parallelism.

- Backend Compiler

The Backend Compiler ensures that the high-level code can be efficiently executed on the hardware. It takes the IR operations and employs a suite of techniques to optimize resources between compute and memory operations to ensure maximum utilization. Key techniques include:

· **Partitioning:** For configurations using multiple cards, the model is broken down into smaller components, which are distributed across multiple devices for efficient parallelism.

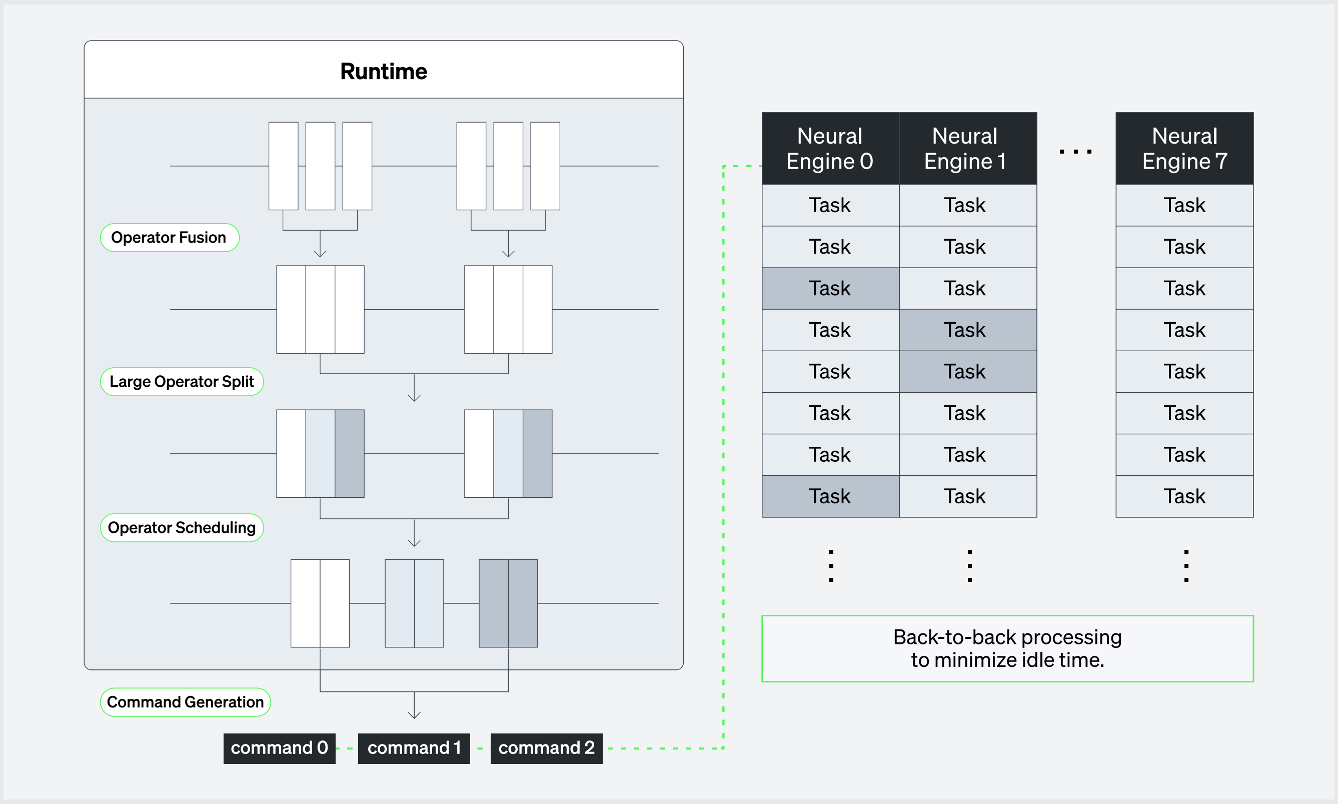
· **Fusion:** Merging operators to reduce unnecessary intermediate activation transfers between computations, leading to optimized computing in the Neural Engines.

· **Splitting:** Within the device, operations are divided to maximize performance while optimizing for pipelining and scheduling.

· **Tiling:** The split operations are further divided across the Neural Engines to achieve the most efficient performance.

The Backend Compiler produces three main outputs:

1. **Command Stream:** Instructions for the Command Processor to manage workload execution across the chip's layers.
2. **Program Binary:** Highly optimized instructions for the chip's Neural Engines, based on the program stream from the Compute Library.
3. **Kernel Serialization:** Converting FP32 weights to FP16, the most optimal format for our Neural Engines.



[Figure 2. Model Compilation]

Compute Library

The Compute Library includes a comprehensive suite of highly optimized low-level operations, which are essential for model inference. These low-level operations form the programmable components of the arithmetic logic units within the Neural Engines. The Compute Library prepares the Program Binary at the Compiler's command.
The RBLN SDK supports low-level operations for both traditional Convolutional Neural Networks (CNNs) and state-of-the-art GenAI models. This includes hundreds of General Matrix Multiply (GEMM), normalization, and nonlinear activation functions. Thanks to the flexibility of the Neural Engines, the list of supported low-level operations continues to expand, enabling acceleration across a wide range of AI applications.

Runtime Module

The Runtime Module acts as the intermediary between the compiled model and the hardware, managing the actual execution of programs. It prepares executable instructions generated by the Compiler, manages data transfer between memory and the Neural Engines, and monitors performance to optimize the execution process.

Driver

The Driver, consisting of the Kernel-Mode Driver (KMD) and User-Mode Driver (UMD), provides efficient, safe, and flexible access to the hardware. The KMD allows the operating system to recognize the hardware and exposes APIs to the UMD. It also delivers the Command Stream from the Compiler stack to the device. The UMD, running in user space, intermediates between the application software and the hardware, managing their interactions.

Firmware

The Firmware is the lowest-level software component on ATOM™, serving as the final interface between software and hardware. It controls the tasks of the Command Processor, which orchestrates ATOM™'s operations. Located on the SoC, the Command Processor manages the Command Stream (the actual AI workloads) across multiple layers of the memory architecture and monitors the hardware's health status.

Flexible Architecture, Maximized by Software

ATOM™ embodies flexibility, allowing adaptation to various workloads and applications.

Future-Proofing

As new models and algorithms are developed, they may require new or specialized low-level operations. With continued support from our software, our chip can flexibly support different computational operations, efficiently running any of these architectures without needing specialized hardware for each type.

Optimization

Models often require different types of computations at different stages (e.g., convolution, matrix multiplication, pooling, activation functions). Our compiler allows it to adaptively allocate resources and optimize performance for each stage of the model.

Multi-device Scalability

To ensure stable performance even with larger GenAI workloads, our advanced Driver and Firmware support robust inter-device communication. Performance is optimized through strategic partitioning designed to minimize inter-device communication overhead.

Dynamic Configuration

Different applications may prioritize different aspects, such as speed, or energy efficiency. Our advanced Dynamic Voltage Frequency Scaling (DVFS) allows for flexible configurations that can be optimized for minimum power consumptions based on the specific requirements of the application.

YOLOv6-Large Performance Comparison

ATOM™ is engineered for energy-efficient performance and exhibits flexibility in supporting a diverse array of models. To illustrate its capabilities with prominent AI models, inference was performed using YOLOv6. YOLOv6, a state-of-the-art CNN-based object detection model, simultaneously classifies and detects objects. Building upon the enhancements and optimizations of preceding You Only Look Once (YOLO) models, YOLOv6 achieves accurate and rapid real-time object detection.

In the test, YOLOv6-Large was run on Rebellions' RBLN-CA12, a PCIe card containing the AI accelerator ATOM™. Its performance was compared against NVIDIA's RTX A5000 GPU, measuring the following performance indicators:

· **Watts:** The power consumption, which is especially relevant for large-scale deployments, as it affects operational costs and feasibility.

· **Joules per frame:** The performance measure which represents the overall energy efficiency into a comparable figure.

	ATOM™	**A5000**
Power Consumption (W)	avg. 40	avg. 110
Energy Consumption (J/Frame)	avg. 0.64	avg. 2.82

* Input resolution: 640x640

[Table 1. Test Result]

The results in Table 1 clearly highlight ATOM™'s superior performance in running the YOLOv6-Large model in all of the measurements. ATOM™ achieves up to 2.1x performance and 4.5x energy efficiency compared to RTX A5000. This was possible thanks to the architecture, and the RBLN Software Stack that brought it to full utilization.

Efficiency and Performance: Scaling Benefits

Efficient resource utilization and workload balancing at both the hardware and software levels prevent excessive power draw from any single part of the chip and avoid thermal hotspots that can increase power usage.

By effectively managing concurrency, our Software Stack allows impressive parallel processing that takes full advantage of the chip's Neural Engines. The data flow and execution pipeline are also optimized to ensure that the chip's processing units are continuously fed with data. On top of this, the efficient implementation of low-level operations reduces the computational load and latency.

This optimized efficiency and performance has important business implications: while a 1,500W server can host 16 RBLN-CA12, it can only accommodate four graphic cards. Therefore, the significant savings achieved at the individual card level suggest an even more substantial cumulative impact for businesses operating one or more servers.

Conclusion

As AI use cases continue to surge across various industries, designing the right hardware is crucial. Equally important—and often overshadowed—is the need for optimized software. Rebellions' Software Stack is the shadow support, providing the power to maximize the flexibility and performance of the hardware. RBLN-CA12's performance results against NVIDIA's RTX A5000 in running the YOLOv6-Large model clearly demonstrate the significant performance advantages of integrating well-optimized software with advanced hardware, solidifying Rebellions' commitment to delivering cutting-edge AI solutions.