RGHT-Q: Reconfigurable GEMM Unit for Heterogeneous-Homogeneous Tensor Quantization

Seungho Lee*, Donghyun Nam*, and Jeongwoo Park[†]
Dept. of Semiconductor Convergence Engineering, Dept. of Electrical and Computer Engineering*[†]*,*
Sungkyunkwan University
Email: *otuloom@g.skku.edu, *nam3918@skku.edu, [†]jeffjw@g.skku.edu

***Abstract*—The high computational demands of large language models (LLMs) are limited by the lack of GPU hardware support for heterogeneous quantization, which mixes integers and floating points. To address this limitation, we propose an LLM processing element (PE), RGHT-Q, which features reconfigurable general-matrix multiplication (GEMM) operations for both heterogeneous and homogeneous tensor quantization. The RGHT-Q introduces a novel design that leverages butterfly routing and multi-precision multipliers. As a result, we achieve significant performance improvements, offering 3.14× higher energy efficiency, and 1.56× better area efficiency compared to prior designs.**

I. Introduction

Traditional approaches to address the computational demands of large language models (LLMs) have mainly focused on homogeneous quantization (e.g., FP16-FP16, FP8-FP8), which applies uniform precision scaling to both weights and activations. Recently, heterogeneous quantization, particularly weight-only quantization (e.g., FP16-INT4, BF16-INT4), has emerged as a promising alternative. However, the absence of dedicated FP-INT arithmetic units on GPUs forces these methods to rely on FP-FP units, thereby missing optimization opportunities in terms of area and power efficiency. To overcome these challenges, we introduce RGHT-Q, a processing element (PE) designed to support reconfigurable General-Matrix Multiplication (GEMM) operations for both heterogeneous and homogeneous quantization.

II. Reconfigurable Multiply-Shift Unit with Buffered Routing

Minimizing hardware overhead while supporting multiple quantization levels is a critical design challenge in reconfigurable computing units. For instance, in a unit supporting FP16-FP16 and FP8-FP8 precisions, achieving twice the throughput for 8-bit precision compared to 16-bit precision requires additional hardware resources. This challenge becomes even more pronounced when supporting heterogeneous quantization, where operands have differing bit-widths. To tackle this, we first propose the Reconfigurable Multiply-Shift Unit (RMS), the smallest computational element of the RGHT-Q. The core of the RMS lies in the 11-bit × 4-bit multiplier-shifter and the buffer routing mechanism (Fig. 1(a)).

When the RMS operates under case 1 of Table I, the four active multipliers and shifters handle the multiplication of the operands and align the results by the difference between each exponent value and the global maximum exponent value of the PE. The aligned results are then used to generate a 4-way dot product ($d_{out} = a_0b_0 + a_1b_1 + a_2b_2 + a_3b_3$) (Fig. 1(a-b)).

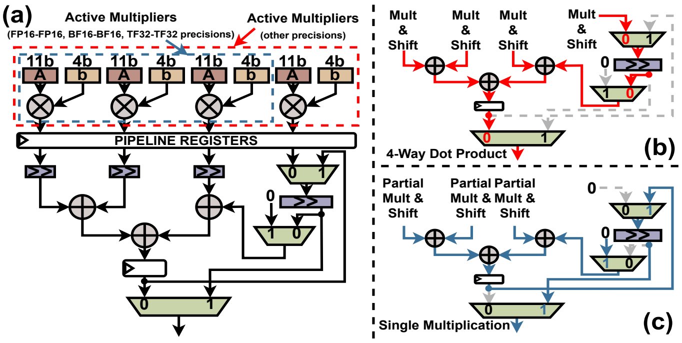

Fig. 1. (a) Block diagram of reconfigurable multiply-shift unit. (b) Normal routing scheme for generating a 4-way dot product. (c) Buffered routing scheme for generating a single multiplication result.

TABLE I
Performance Metrics for Supported Precisions

Case	Supported Precision	Outputs	Cycle (w/o latency)	Throughput (outputs/cycle)
Case 1-1	FP16-INT4, BF16-INT4	4×32	1	128
Case 1-2	FP8-FP8 (E5M2, E4M3)	4×32	1	128
	FP16-INT8, BF16-INT8	4×32	2	64
	INT8-INT8	4×32	2	64
Case 2	FP16-FP16, BF16-BF16, TF32-TF32	1×32	1	32

When the RMS operates at precision corresponding to case 2 in Table I, three active multipliers handle a single multiplication of a_0 and b_0. For example, in FP16-FP16 operations, the mantissa of a_0 is broadcast to the three active multipliers, while the mantissa of b_0 is divided into 4-bit chunks and fed into these multipliers. The three partial products are aligned considering the differences in significant bits and then accumulated, ensuring accurate multiplication results. Subsequently, the buffered routing mechanism (Fig. 1(c)) efficiently utilizes the fourth shifting unit without introducing additional shifting units. This mechanism aligns the results by the difference between the operand's exponents and the global maximum exponent of the PE, thereby generating a single multiplication result.

III. Grid-Based Structure with Butterfly Router

The distinguishing feature of the RGHT-Q lies in its capability to efficiently support both heterogeneous and homogeneous quantization schemes. This is achieved through the computing grid and the following datapaths responsible for processing its outputs. The computing grid consists of four grouped RMS units and a Local Butterfly Router(LBR) that manages the input routing for the RMS units (Fig. 2(a)).

For homogeneous quantization under case 2 precisions in Table I, each computing grid generates four multiplication results.

978-3-9826741-0-0/DATE25/© 2025 EDAA

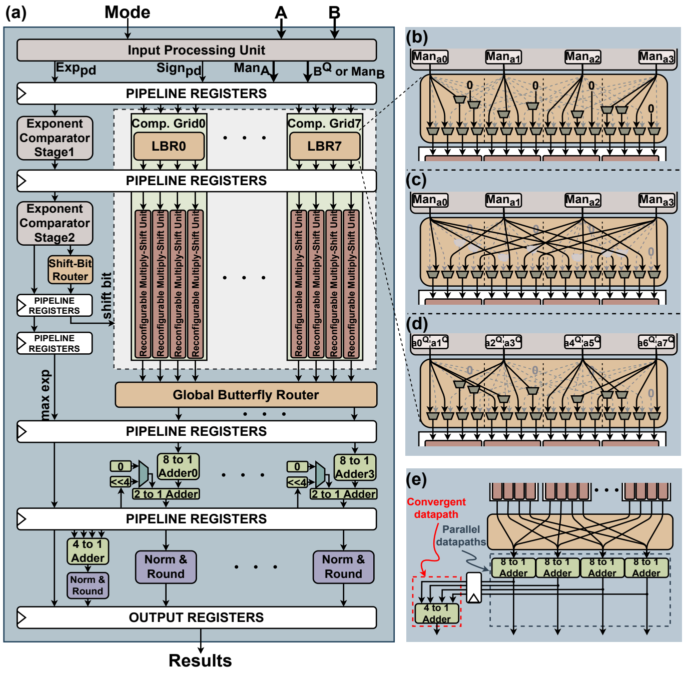

Fig. 2. (a) Block diagram of RGHT-Q. (b) Local butterfly routing scheme for generating a single multiplication per RMS. (c) Local butterfly routing scheme for generating a 4-way dot product per RMS. (d) Local butterfly routing scheme for generating a 4-way dot product per RMS in homogeneous precision (FP8, INT8). (e) Details of the global butterfly router and the following datapaths.

The LBR broadcasts the same mantissa A to all RMS units, enabling operations with three chunked mantissa B values (Fig. 2(b)). This is achieved by configuring the RMS units according to the Fig. 1(c)'s routing scheme. Across eight computing grids, a total of 32 multiplication results are generated. These results are aggregated through a convergent datapath consisting of an 8-to-1 Adder and a 4-to-1 Adder, ultimately producing a single 32-way dot product result (Fig. 2(e)).

For heterogeneous quantization under case 1-1 precisions in Table I, each computing grid generates four 4-way dot products. The LBR supplies four distinct mantissa A values to the RMS units, while mantissa B values are similarly routed and supplied (Fig. 2(c)). This is achieved by configuring the RMS units according to the Fig. 1(b)'s routing scheme. Across the eight computing grids, a total of 32 independent 4-way dot products are generated. These results are mapped into groups of eight through parallel datapaths in the 8-to-1 adders, ultimately producing four 32-way dot products (Fig. 2(e)).

Exceptionally, for case 1-2 in Table I, where operand B is INT8, a 2-to-1 adder is used. In this case, the computation results for high 4 bits and low 4 bits of INT8 are accumulated over two cycles, ultimately producing four 32-way dot product results (Fig. 2(a, d)). For FP8-FP8 precision, the design skips a 2-to-1 adder and directly generates four 32-way dot product results (Fig. 2(a, d)). The RGHT-Q design is implemented with an N-way configuration of 32. Furthermore, by selecting N as a multiple of 4, the architecture ensures scalability.

IV. Measurement Results

The RGHT-Q was synthesized using a 28nm process at 1000 MHz. The measured area is 75313µm², with a measured power consumption of 33.966mW obtained through PrimeTime.

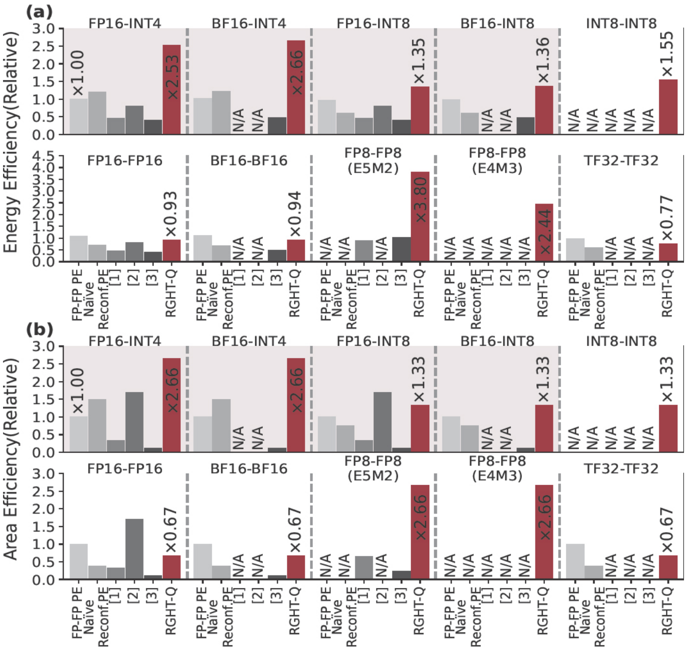

Fig. 3. (a) Relative energy efficiency of RGHT-Q and prior works. (b) Relative area efficiency of RGHT-Q and prior works.

The relative energy efficiency of the RGHT-Q is compared to FP-FP only PE (standard in GPUs), naïve reconfigurable PE for heterogeneous-homogeneous quantization, and prior reconfigurable MAC units that support various floating-point quantization schemes [1], [2], [3]. For our primary target of heterogeneous precision (highlighted in red in Fig. 3), RGHT-Q shows great improvements, with energy efficiency of 2.53× to 2.66× and area efficiency of 2.66×, compared to 1.20× to 1.22× and 1.50× of the naïve reconfigurable PE. Furthermore, compared to [1], [2], [3], RGHT-Q demonstrates over ×3.14 higher energy efficiency and over ×1.56 higher area efficiency.

The RGHT-Q achieved 3.07 to 3.22 TFLOPS/W for the case 1-1 in Table I, and 1.13 to 1.14 TFLOPS/W for the case 2 in Table I. The area efficiency was measured at 3.40 TFLOPS/mm² for case 1-1 and 0.85 TFLOPS/mm² for case 2.

V. Acknowledgement

This work was supported by the National Research Fund (NRF) granted by the Korean government (MSIT) (No. 00355678) and Institute for Information & Communication Technology Promotion (IITP) grant funded by the Korean government (MSIP) (No. 1711195788). The EDA tool was supported by the IC Design Education Center (IDEC), Korea.

References

[1] H. Zhang, D. Chen, and S.-B. Ko, "New flexible multiple-precision multiply-accumulate unit for deep neural network training and inference," *IEEE Transactions on Computers*, vol. 69, no. 1, pp. 26–38, 2020.

[2] W. Mao, K. Li, Q. Cheng, L. Dai, B. Li, X. Xie, H. Li, L. Lin, and H. Yu, "A configurable floating-point multiple-precision processing element for hpc and ai converged computing," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 30, no. 2, pp. 213–226, 2022.

[3] S. Mach, F. Schuiki, F. Zaruba, and L. Benini, "Fpnew: An open-source multiformat floating-point unit architecture for energy-proportional transprecision computing," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 29, no. 4, pp. 774–787, 2021.