

Design and Evaluation of a Novel Multifunctional Unit for a Class-F Five-Moduli Residue Number System with Hardware Resource Sharing

Mohamad Hosein AzamiRad^{1*}, Amir Sabagh Molahoseini^{1*}, Azadeh Alsadat Emrani Zarandi², Peyman Keshavarzian¹

¹ Department of Computer Engineering, Ke.C., Islamic Azad University, Kerman, Iran

² Department of Computer Engineering, Shahid Bahonar University of Kerman, Kerman Iran

* Corresponding author email address: amir@iauk.ac.ir

Article Info

Article type:

Original Research

How to cite this article:

AzamiRad, M. H., Sabagh Molahoseini, A., Alsadat Emrani Zarandi, A., & Keshavarzian, P. (2026). Design and Evaluation of a Novel Multifunctional Unit for a Class-F Five-Moduli Residue Number System with Hardware Resource Sharing. *Decision Science and Intelligent Systems*. 3(4), 1-22.



© 2026 the authors. Published by KMAN Publication Inc. (KMANPUB), Ontario, Canada. This is an open access article under the terms of the Creative Commons Attribution-NonCommercial 4.0 International (CC BY-NC 4.0) License.

ABSTRACT

The residue number system (RNS), by eliminating carry propagation among moduli, enables fast and parallel computation. However, inter-modular operations such as reverse conversion, sign detection, number comparison, and scaling can become major bottlenecks in RNS-based architectures. In this study, a novel multifunctional unit is designed and evaluated for a Class-F five-moduli set with the aim of reducing hardware redundancy and improving the efficiency of inter-modular computations. The employed moduli set is $\{2^n, 2^n - 1, 2^n + 1, 2^{n+1} - 1, 2^{n-1} - 1\}$, and the proposed architecture supports modular arithmetic operations, including addition, subtraction, and multiplication, together with reverse conversion, sign detection, signed-number comparison, and scaling within an integrated structure. In the proposed approach, the reverse-conversion core is designed using a staged architecture and the computation of intermediate information, while part of the same hardware resources is reused for sign detection and other inter-modular operations. Furthermore, hardware resource sharing, critical-path reduction, and optimization of the computational structure are employed to control circuit redundancy. The architecture was implemented at the register-transfer level (RTL) and, following functional validation, synthesized using 65-nm CMOS technology and evaluated in terms of area, critical-path delay, and area-delay product (ADP). The results for $(n=8)$, (12) , and (26) , corresponding to dynamic ranges of approximately 40, 60, and 130 bits, respectively, show that the proposed unit achieves areas of 5,592.9, 9,594.3, and 19,836.9 μm^2 and critical-path delays of 0.397, 0.439, and 0.526 ns, respectively. The corresponding ADP values are 2,220.3813, 4,211.8977, and 10,434.2094 $\mu\text{m}^2\cdot\text{ns}$. The results demonstrate the favorable efficiency of the proposed architecture and the effectiveness of hardware resource sharing in the design of a multifunctional unit for a Class-F five-moduli residue number system.

Keywords: residue number system; Class-F moduli set; multifunctional unit; hardware resource sharing; reverse conversion; sign detection.

Extended Abstract

Introduction

The residue number system (RNS) is a non-weighted numerical representation in which an integer is represented by its residues with respect to a set of pairwise coprime moduli. Because arithmetic operations are performed independently within each residue channel, RNS architectures eliminate carry propagation between channels and therefore provide an effective basis for parallel and high-speed computation. This property has made RNS particularly attractive for digital signal processing, cryptographic computation, modular arithmetic, and other high-performance hardware applications in which arithmetic throughput and critical-path reduction are important design objectives (Ahsan et al., 2022; Ming et al., 2011; Omondi & Premkumar, 2007). The theoretical foundations of residue arithmetic and its computational advantages have been established in classical works on computer arithmetic and RNS-based processing (Garner, 1959; Parhami, 2000; Szabo & Tanaka, 1967).

Despite the intrinsic parallelism of RNS, practical implementation is not limited to independent modular addition, subtraction, and multiplication. Many processing tasks require inter-modular operations, including reverse conversion from residue representation to conventional binary form, sign detection, signed-number comparison, scaling, and base-related transformations. These operations depend on information distributed across several residue channels and may therefore introduce substantial hardware complexity, logic depth, and latency. Consequently, the overall efficiency of an RNS processor is determined not only by the speed of its modular arithmetic units but also by the cost of these inter-modular operations. Reverse conversion is particularly important because it reconstructs the weighted representation required for output interpretation and for interaction with conventional digital systems. Several reverse-conversion methods have been developed using the Chinese remainder theorem, mixed-radix conversion, and specialized moduli-set relationships (Akkal, 2007; Cao et al., 2007; Kuchukov, 2022).

The selection of the moduli set has a direct effect on dynamic range, arithmetic regularity, converter complexity, and hardware implementation cost. Five-moduli systems are especially attractive because they can provide relatively large dynamic ranges while preserving structures that can be mapped efficiently onto modular adders, end-around-carry units, shifts, rotations, and constant-coefficient operations. Patronik and Piestrak (2017) examined a flexible five-moduli family and demonstrated that its algebraic structure can be exploited in the design of reverse converters. Kumar and Chang (2017) subsequently investigated signed integer comparison for a related balanced five-moduli RNS and showed that scaling operations can be integrated into the comparison process. These studies highlight the potential of structured five-moduli systems for inter-modular processing, but they also indicate that many functions have traditionally been developed as separate hardware blocks.

A major architectural issue therefore concerns hardware duplication. If reverse conversion, sign detection, comparison, and scaling are implemented independently, similar arithmetic elements may be instantiated repeatedly, even though some operations depend on common intermediate quantities. Previous work has shown that multifunctional RNS architectures can reduce such redundancy by organizing several complex operations around shared datapaths and reusable arithmetic resources. Molahosseini et al. (2017), for example, proposed a multifunctional unit for efficient RNS-based datapaths and demonstrated that common hardware resources can support multiple operations. Related research has

also explored integrated reverse conversion and sign detection, further indicating that intermediate conversion information can be reused instead of being recomputed by independent units (Mojahed et al., 2021).

Against this background, the present study develops and evaluates a new multifunctional architecture for the Class-F five-moduli set $\{2^n, 2^n - 1, 2^n + 1, 2^{n+1} - 1, 2^{n-1} - 1\}$. The principal objective is not to claim the first introduction of this five-moduli family, but rather to formulate it as a Class-F architecture and exploit its structural properties for hardware resource sharing. The proposed unit integrates modular addition, subtraction, and multiplication with reverse conversion, sign detection, signed-number comparison, and scaling. The central design hypothesis is that systematic reuse of intermediate information and computational resources can reduce hardware redundancy while maintaining favorable area, critical-path delay, and area-delay product (ADP). The source manuscript defines this research problem, architectural rationale, and evaluation framework in detail. 1_30517371061

Materials and Methods

The study adopted a hardware-design and performance-evaluation approach. The proposed architecture was constructed around the Class-F five-moduli set $\{2^n, 2^n - 1, 2^n + 1, 2^{n+1} - 1, 2^{n-1} - 1\}$, with the parameter n restricted to valid even values. The five residue channels were designed according to the arithmetic characteristics of their respective moduli. Operations associated with powers of two were implemented using direct binary structures, whereas channels based on moduli close to powers of two were mapped to end-around-carry addition, shifts, rotations, sign inversion, modular reduction, and constant-coefficient computation.

A multifunctional datapath was then designed to support modular addition, subtraction, and multiplication as well as reverse conversion, sign detection, signed comparison, and scaling. Instead of allocating a fully independent hardware path to each inter-modular function, the architecture used a common reverse-conversion core and shared intermediate data. Reverse conversion was divided into successive computational stages, and intermediate components produced during these stages were stored and reused. In particular, the more significant intermediate component generated by the reverse-conversion path was also used by the sign-detection logic, thereby avoiding the need for a completely separate conversion structure for sign determination.

Signed comparison was implemented by combining sign information with magnitude comparison. Two comparator configurations were considered: a parallel configuration using two reverse-conversion paths and a sequential configuration in which one reverse-conversion core was reused between operands by means of intermediate registers. The scaling unit was designed separately for the arithmetic characteristics of each residue channel while remaining integrated into the common datapath. A central control unit activated the appropriate resources according to the selected operation and routed outputs through a shared selection structure.

The complete architecture was described at the register-transfer level using Verilog HDL. Functional simulation was performed using ModelSim SE, while Vivado was employed for preliminary RTL inspection and hardware feasibility assessment. After functional verification, ASIC synthesis and timing analysis were conducted using Synopsys tools and a 65-nm CMOS standard-cell technology library. Validation was performed both at the subunit level and at the integrated multifunctional-unit level. Test vectors covered normal operating conditions, boundary values, signed and unsigned cases, scaling

conditions, and end-to-end multifunctional execution. Hardware performance was evaluated using circuit area, critical-path delay, and ADP. Three design scales were examined, with $n=8$, $n=12$, and $n=26$, producing dynamic ranges of approximately 40, 60, and 130 bits, respectively. The proposed Class-F unit was also compared with previously reported Class-A and Class-C multifunctional architectures.

Findings

Functional validation confirmed the correct operation of all major subunits, including the direct converter, reverse converter, sign detector, comparator, scaling unit, and complete multifunctional unit. The integrated design correctly routed shared intermediate information among reverse conversion, sign detection, comparison, and scaling. This result demonstrated that resource sharing did not alter the functional behavior expected from the individual operations.

The synthesis results showed a systematic increase in area and delay as the dynamic range increased. For $n=8$, corresponding to an actual dynamic range of approximately 39.99 bits, the proposed architecture occupied $5,592.9 \mu\text{m}^2$ and exhibited a critical-path delay of 0.397 ns. Its ADP was $2,220.3813 \mu\text{m}^2 \cdot \text{ns}$. For $n=12$, with an actual dynamic range of approximately 60 bits, the synthesized area increased to $9,594.3 \mu\text{m}^2$, the delay to 0.439 ns, and the ADP to $4,211.8977 \mu\text{m}^2 \cdot \text{ns}$. At the largest evaluated configuration, $n=26$, corresponding to an actual dynamic range of approximately 130 bits, the architecture required $19,836.9 \mu\text{m}^2$, achieved a critical delay of 0.526 ns, and produced an ADP of $10,434.2094 \mu\text{m}^2 \cdot \text{ns}$. These synthesis results are reported in the manuscript's hardware evaluation section. 1_30517371061

Comparison with the reference multifunctional architectures showed that the proposed Class-F design remained competitive at all three operating scales. At the smallest comparison point, the proposed unit reduced area by approximately 16.46% relative to the Class-C architecture and by 0.14% relative to Class-A. Critical-path delay was reduced by approximately 28.98% relative to Class-C and 1.00% relative to Class-A, whereas ADP decreased by approximately 40.67% and 1.13%, respectively.

At the approximately 60-bit operating point, where the dynamic ranges were almost directly comparable, the proposed Class-F design occupied $9,594.3 \mu\text{m}^2$ compared with $11,141.6 \mu\text{m}^2$ for Class-C and $9,668.9 \mu\text{m}^2$ for Class-A. These values correspond to area reductions of approximately 13.89% and 0.77%, respectively. The critical-path delay of 0.439 ns was approximately 31.73% lower than that of Class-C and 3.94% lower than that of Class-A. The ADP improvement was more pronounced, reaching approximately 41.21% relative to Class-C and 4.68% relative to Class-A.

At the largest scale, the proposed architecture showed further gains. Its area was approximately 24.89% lower than that of the Class-C reference and 5.12% lower than that of Class-A. Its delay was approximately 28.92% lower than Class-C and 8.20% lower than Class-A. The corresponding ADP reductions were approximately 46.61% and 12.90%. Therefore, the relative advantage of the proposed Class-F architecture became more apparent as the dynamic range increased, particularly for the combined area-delay metric.

Discussion and Conclusion

The results demonstrate that the main contribution of the proposed architecture lies in the organization of multiple RNS functions around shared computational and intermediate-data resources. Instead of treating reverse conversion, sign detection, comparison, and scaling as isolated subsystems, the architecture uses the reverse-conversion process as a shared computational backbone. This organization reduces duplication of arithmetic logic, intermediate registers, constant-coefficient operations, and parts of the comparison and scaling circuitry.

The synthesis results also indicate that resource sharing does not merely reduce circuit area; it contributes to a favorable timing structure. The consistently lower critical-path delays suggest that the staged organization of the reverse converter, together with appropriate reuse of intermediate quantities, prevented the multifunctional architecture from developing excessively deep logic paths. This effect is especially important because multifunctionality can otherwise increase routing complexity and control overhead.

The ADP results provide the clearest overall evidence of architectural efficiency. Across all evaluated scales, the proposed Class-F unit achieved a lower ADP than both reference classes, with the advantage increasing at the largest dynamic range. This finding indicates that the proposed architecture achieves an effective trade-off between silicon cost and execution speed rather than optimizing one measure at the expense of the other.

Another important outcome is scalability. Although area and delay naturally increased with dynamic range, the architecture retained competitive behavior from approximately 40 to 130 bits. The increasing relative improvement at the largest configuration suggests that hardware reuse becomes progressively more beneficial as operand width and inter-modular computational complexity grow. The design is therefore particularly relevant to implementations in which large dynamic range and multiple RNS operations must coexist within constrained hardware resources.

In conclusion, the study confirms that a general-purpose multifunctional unit can be developed for the Class-F five-moduli RNS by combining specialized modular channels, staged reverse conversion, intermediate-data reuse, and hardware resource sharing. The proposed unit successfully integrates modular arithmetic, reverse conversion, sign detection, signed comparison, and scaling within a unified and configurable RTL architecture. Functional verification confirmed the correctness of the integrated design, while 65-nm CMOS synthesis demonstrated favorable area, delay, and ADP characteristics. The results support the central premise that systematic reuse of hardware and intermediate information can reduce redundancy without sacrificing the principal computational advantages of RNS. Consequently, the proposed Class-F architecture provides a practical foundation for efficient, scalable, and multifunctional RNS-based datapaths.

طراحی و ارزیابی یک واحد چندمنظوره جدید برای سیستم اعداد مانده‌ای پنج‌پیمانه‌ای Class-F با اشتراک‌گذاری منابع سخت‌افزاری

محمدحسین اعظمی راد^۱، امیر صباغ ملاحسینی^{۱*}، آزاده السادات عمرانی زرنندی^۲، پیمان کشاورزبان^۱

۱. گروه مهندسی کامپیوتر، واحد کرمان، دانشگاه آزاد اسلامی، کرمان، ایران
۲. گروه مهندسی کامپیوتر، دانشگاه شهید باهنر کرمان، کرمان، ایران

*ایمیل نویسنده مسئول: amir@iauk.ac.ir

اطلاعات مقاله

چکیده

نوع مقاله

پژوهشی/اصیل

نحوه استناد به این مقاله:

اعظمی راد، محمدحسین، صباغ ملاحسینی، امیر، السادات عمرانی زرنندی، آزاده، و کشاورزبان، پیمان. (۱۴۰۵). طراحی و ارزیابی یک واحد چندمنظوره جدید برای سیستم اعداد مانده‌ای پنج‌پیمانه‌ای Class-F با اشتراک‌گذاری منابع سخت‌افزاری. علم تصمیم گیری و سیستم های هوشمند، ۳(۴)، ۱-۲۲.



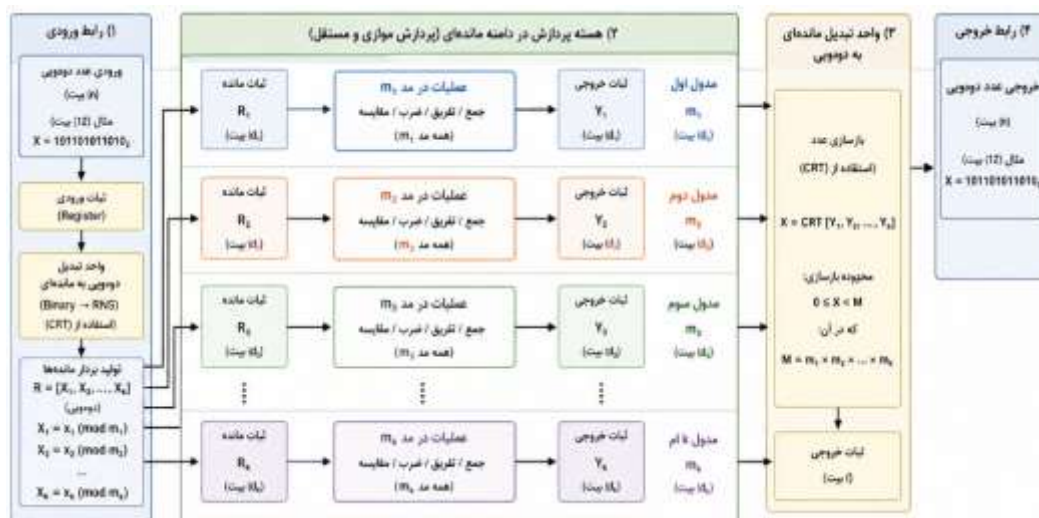
© ۱۴۰۵ تمامی حقوق انتشار این مقاله متعلق به نویسنده است. انتشار این مقاله به صورت دسترسی آزاد مطابق با گواهی (CC BY-NC 4.0) صورت گرفته است.

سیستم اعداد مانده‌ای با حذف انتشار نقلی میان پیمانه‌ها، امکان انجام محاسبات موازی و سریع را فراهم می‌کند؛ با این حال، عملیات بین‌پیمانه‌ای نظیر تبدیل معکوس، تشخیص علامت، مقایسه اعداد و Scaling می‌توانند به گلوگاه اصلی در معماری‌های مبتنی بر سیستم اعداد مانده‌ای تبدیل شوند. در این پژوهش، یک واحد چندمنظوره جدید برای مجموعه پنج‌پیمانه‌ای Class-F با هدف کاهش افزونگی سخت‌افزاری و بهبود کارایی محاسبات بین‌پیمانه‌ای طراحی و ارزیابی شده است. مجموعه پیمانه‌ای مورد استفاده شامل $\{2^n - 1, 2^{n+1} - 1, 2^n + 1, 2^n - 1, 2^n\}$ است و معماری پیشنهادی، عملیات حسابی جمع، تفریق و ضرب پیمانه‌ای را در کنار تبدیل معکوس، تشخیص علامت، مقایسه اعداد علامت‌دار و Scaling در یک ساختار یکپارچه پشتیبانی می‌کند. در روش پیشنهادی، هسته تبدیل معکوس با استفاده از ساختار مرحله‌ای و محاسبه اطلاعات میانی طراحی شده و بخشی از همین منابع در تشخیص علامت و سایر عملیات بین‌پیمانه‌ای مجدداً مورد استفاده قرار گرفته است. همچنین با اشتراک‌گذاری منابع سخت‌افزاری، کاهش مسیرهای بحرانی و بهینه‌سازی ساختار محاسباتی، افزونگی مدار کنترل شده است. معماری در سطح RTL پیاده‌سازی و پس از اعتبارسنجی عملکردی، با استفاده از فناوری CMOS ۶۵ نانومتر سنتز و از نظر مساحت، تأخیر بحرانی و حاصل ضرب مساحت-تأخیر (ADP) ارزیابی شده است. نتایج برای $n=8, 12, 26$ به ترتیب با دامنه دینامیکی تقریباً ۴۰، ۶۰ و ۱۳۰ بیت نشان می‌دهد که واحد پیشنهادی به ترتیب مساحت‌های ۵۵۹۲.۹، ۹۵۹۴.۳ و ۱۹۸۳۶.۹ میکرومترمربع و تأخیرهای ۰.۳۹۷، ۰.۴۳۹ و ۰.۵۲۶ نانوثانیه را به دست آورده است؛ ADP متناظر نیز برابر ۲۲۲۰.۳۸۱۳، ۲۲۱۱.۸۹۷۷ و ۱۰۴۳۴.۲۰۹۴ $\mu m^2 \cdot ns$ است. نتایج نشان‌دهنده کارایی مناسب معماری پیشنهادی و اثربخشی اشتراک‌گذاری منابع در طراحی یک واحد چندمنظوره برای سیستم اعداد مانده‌ای پنج‌پیمانه‌ای Class-F است.

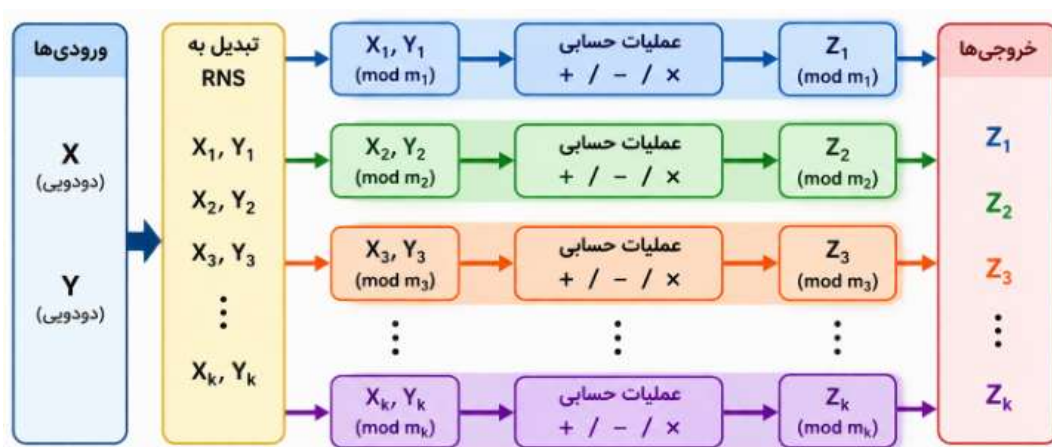
کلیدواژگان: سیستم اعداد مانده‌ای؛ مجموعه پیمانه‌ای Class-F؛ واحد چندمنظوره؛ اشتراک‌گذاری منابع سخت‌افزاری؛ تبدیل معکوس؛ تشخیص علامت.

۱. مقدمه

رشد روزافزون کاربردهای پردازش دیجیتال در حوزه‌هایی مانند پردازش سیگنال و تصویر، مخابرات، رمزنگاری و سامانه‌های محاسباتی با کارایی بالا، نیاز به معماری‌هایی را افزایش داده است که بتوانند عملیات حسابی را با سرعت بالا و هزینه سخت‌افزاری قابل قبول انجام دهند. یکی از رویکردهای مطرح در این زمینه، سیستم اعداد مانده‌ای است. در سیستم اعداد مانده‌ای، یک عدد صحیح به جای نمایش متعارف دودویی، به مجموعه‌ای از باقیمانده‌ها نسبت به پیمانه‌های مستقل تبدیل می‌شود. در نتیجه، عملیات جمع، تفریق و ضرب در هر کانال پیمانه‌ای به صورت مستقل انجام می‌گیرد و انتشار نقلی میان کانال‌ها وجود ندارد؛ ویژگی‌ای که امکان موازی‌سازی ذاتی محاسبات و کاهش وابستگی‌های موجود در مسیر بحرانی را فراهم می‌کند [۱، ۲]. در شکل (۱) معماری کلی سیستم اعداد مانده‌ای و در شکل (۲) استقلال کانال‌های پیمانه‌ای نمایش داده شده است.



شکل (۱): معماری کلی سیستم اعداد مانده‌ای



شکل (۲): استقلال کانال‌های پیمانه‌ای در عملیات حسابی RNS

با وجود این مزایا، استفاده عملی از RNS تنها به انجام محاسبات درون‌پیمانه‌ای محدود نیست. بسیاری از عملیات موردنیاز در یک پردازشگر واقعی، ماهیتی بین‌پیمانه‌ای دارند و مستلزم دسترسی هم‌زمان یا وابسته به چند کانال هستند. تبدیل عدد RNS به نمایش دودویی، تشخیص علامت، مقایسه اعداد، Scaling و برخی پردازش‌های وابسته به نمایش موقعیتی، از جمله عملیاتی هستند که پیچیدگی آن‌ها می‌تواند بخش مهمی از هزینه و تأخیر کل یک معماری RNS را تشکیل دهد. به همین دلیل، مزیت موازی‌سازی RNS در صورتی می‌تواند

به بهبود واقعی کارایی سامانه منجر شود که این عملیات بین‌پیمانه‌ای نیز با معماری مناسب و هزینه سخت‌افزاری کنترل‌شده پیاده‌سازی شوند. در ادبیات موجود، راهکارهای متعددی برای طراحی مبدل‌های معکوس، آشکارسازهای علامت، مقایسه‌گرها و واحدهای Scaling ارائه شده‌اند؛ با این حال، در بسیاری از این رویکردها هر عملیات به صورت یک بلوک مجزا طراحی می‌شود و در نتیجه بخشی از سخت‌افزار مورد استفاده در این عملیات با یکدیگر هم‌پوشانی دارد، بدون آنکه الزاماً این اشتراک در سطح یک معماری یکپارچه مورد بهره‌برداری قرار گیرد. در این میان، انتخاب مجموعه‌های پیمانه‌ای مناسب نقش اساسی در میزان پیچیدگی و کارایی معماری RNS دارد. پژوهش حاضر بر مجموعه پنج‌پیمانه‌ای

$$F(n) = \{2^n, 2^n - 1, 2^n + 1, 2^{n+1} - 1, 2^{n-1} - 1\}$$

با قیود لازم بر پارامتر n متمرکز است که در این مقاله با عنوان Class-F از آن یاد می‌شود. نکته مهم آن است که مجموعه فوق را نمی‌توان از نظر سابقه پژوهشی، یک مجموعه پیمانه‌ای کاملاً بی‌سابقه تلقی کرد؛ زیرا خانواده انعطاف‌پذیر پنج‌پیمانه‌ای شامل این حالت در [۱۸] نیز بررسی شده است. بنابراین، تمرکز نوآورانه این پژوهش نه بر ادعای معرفی این خانواده برای نخستین بار، بلکه بر صورت‌بندی معماری Class-F و توسعه یک واحد چندمنظوره عمومی مبتنی بر این مجموعه با تأکید بر اشتراک و استفاده مجدد از منابع سخت‌افزاری است. وجود پنج کانال با ساختارهای خاص $2^n, 2^n \pm 1, 2^{n+1} - 1, 2^{n-1} - 1$ از یک سو فرصت استفاده از روابط ریاضی و ساختارهای محاسباتی ساده‌تر را فراهم می‌سازد و از سوی دیگر، به دلیل افزایش تعداد کانال‌ها، طراحی عملیات بین‌پیمانه‌ای را به مسئله‌ای پیچیده‌تر تبدیل می‌کند. از این منظر، Class-F بستر مناسبی برای بررسی این پرسش است که آیا می‌توان میان دامنه دینامیکی، سرعت، مساحت و چندمنظوره بودن یک معماری RNS تعادل مناسبی برقرار کرد.

مطالعات پیشین در زمینه واحدهای چندمنظوره نشان داده‌اند که می‌توان عملیات دشوار RNS، از جمله تشخیص علامت، مقایسه بزرگی، تبدیل معکوس علامت‌دار و Scaling را با بهره‌گیری از سخت‌افزارهای قابل استفاده مجدد در یک معماری مشترک سازمان‌دهی کرد. برای نمونه، Molahosseini و همکاران یک واحد چندمنظوره را برای مجموعه‌های چهارپیمانه‌ای Class-A و Class-C ارائه کرده‌اند که هدف آن استفاده مؤثرتر از سطح مدار و ایجاد یک مسیر مشترک برای تعدادی از عملیات پیچیده RNS بود [۱۲]. از سوی دیگر، پژوهش Piestrak و Patronik بر طراحی مبدل‌های معکوس برای خانواده پنج‌پیمانه‌ای مورد نظر تمرکز داشته است [۱۸] و پژوهش Kumar و Chang نیز مسئله مقایسه صحیح علامت‌دار را برای همین خانواده پنج‌پیمانه‌ای، با استفاده از Scaling، مورد بررسی قرار داده است [25]. همچنین، در مطالعات مختلف، تشخیص علامت و Scaling برای مجموعه‌های مختلف RNS به صورت مستقل بررسی شده‌اند. این مطالعات هر یک بخشی از مسئله را حل می‌کنند، اما در مجموع نشان می‌دهند که هنوز نیاز به یک معماری یکپارچه و اختصاصی برای Class-F وجود دارد که در آن عملیات حسابی، تبدیل، تشخیص علامت، مقایسه و Scaling نه به صورت مجموعه‌ای از مدارهای مستقل، بلکه بر پایه منابع مشترک و اطلاعات میانی مشترک سازمان‌دهی شوند.

از این رو، مسئله محوری پژوهش حاضر آن است که چگونه می‌توان یک واحد چندمنظوره عمومی برای مجموعه پنج‌پیمانه‌ای Class-F طراحی کرد که ضمن پشتیبانی از عملیات حسابی پیمانه‌ای، تبدیل مستقیم و معکوس، تشخیص علامت، مقایسه اعداد علامت‌دار و Scaling، از اشتراک و استفاده مجدد از منابع سخت‌افزاری بهره برده و در عین حال پیچیدگی، مساحت و تأخیر معماری را در محدوده مناسب نگه دارد. در این مسئله، یک چالش مهم آن است که روابط ریاضی خاص میان پیمانه‌های Class-F باید به ساختارهای سخت‌افزاری قابل پیاده‌سازی تبدیل شوند؛ بنابراین، طراحی واحدهای CSA, CPA, EAC، مدارهای مقایسه، منطق کنترل و مسیرهای داده باید به گونه‌ای انجام شود که اطلاعات محاسبه‌شده در یک عملیات بتواند تا حد امکان در عملیات دیگر نیز مورد استفاده قرار گیرد.

در واقع، شکاف اصلی تحقیق را می‌توان در سه سطح بیان کرد: نخست، نبود یک چارچوب یکپارچه برای مجموعه پنج‌پیمانه‌ای Class-F؛ دوم، نبود بهره‌گیری سیستماتیک از اطلاعات میانی مبدل معکوس برای پشتیبانی از عملیات دیگری نظیر تشخیص علامت و پردازش‌های وابسته و سوم، نبود یک معماری واحد چندمنظوره اختصاصی برای Class-F که بتواند مجموعه متنوعی از عملیات RNS را در یک ساختار قابل کنترل و قابل پیاده‌سازی سخت‌افزاری ادغام کند. بنابراین، مسئله این پژوهش صرفاً به طراحی یک مدار منفرد محدود نمی‌شود، بلکه به تبدیل مجموعه پیمانه‌ای Class-F به یک زیرساخت پردازشی چندمنظوره مربوط است. انتخاب این موضوع از دو منظر علمی و مهندسی قابل توجیه است. از منظر علمی، مجموعه پنج‌پیمانه‌ای Class-F به دلیل ساختار ویژه پیمانه‌های خود، زمینه مناسبی برای استخراج روابط ساده و اختصاصی در محاسبات، تبدیل و پردازش‌های بین‌پیمانه‌ای فراهم می‌کند و در عین حال، ترکیب پنج پیمانه امکان دستیابی به دامنه دینامیکی قابل توجه را فراهم می‌سازد. از منظر مهندسی نیز مسئله اصلی در بسیاری از سامانه‌های RNS صرفاً سرعت عملیات جمع و ضرب نیست، بلکه هزینه سخت‌افزاری عملیات جانبی مانند تبدیل، تشخیص علامت و مقایسه است. بنابراین، طراحی یک واحد چندمنظوره که بتواند این عملیات را بر پایه سخت‌افزار مشترک انجام دهد، می‌تواند فاصله میان قابلیت‌های نظری RNS و نیازهای یک معماری پردازشی واقعی را کاهش دهد. افزون بر این، استفاده مجدد از منابع می‌تواند از تکرار مدارهای مشابه جلوگیری کرده و زمینه را برای دستیابی هم‌زمان به مساحت کمتر و تأخیر مناسب‌تر فراهم سازد. همین ویژگی سبب شده است که رویکرد Resource Sharing به عنوان یکی از مبانی اصلی معماری پیشنهادی انتخاب شود.

برای پاسخ به مسئله پژوهش، ابتدا ویژگی‌های ریاضی و ساختاری مجموعه Class-F تحلیل و روابط موردنیاز برای تبدیل و پردازش در این مجموعه استخراج و اثبات شده است. سپس، واحدهای محاسبات پیمانه‌ای و مبدل مستقیم متناسب با ساختار هر یک از پنج پیمانه طراحی شده‌اند. در ادامه، یک مبدل معکوس مرحله‌ای توسعه یافته است که در آن با تولید و نگهداری مؤلفه‌های میانی، مسیر تبدیل به بخش‌های قابل مدیریت تقسیم شده و امکان استفاده مجدد از بخشی از این اطلاعات در سایر عملیات فراهم شده است. یکی از محورهای اصلی روش پیشنهادی، طراحی آشکارساز علامت بر پایه اطلاعات میانی همین مبدل معکوس است؛ به گونه‌ای که برای تشخیص علامت، الزاماً نیاز به تولید کامل نمایش دودویی وجود نداشته باشد. همچنین، سازوکار مقایسه اعداد علامت‌دار و واحد Scaling با توجه به ویژگی‌های خاص پنج پیمانه طراحی شده و در نهایت تمامی اجزا در قالب یک واحد چندمنظوره عمومی تحت کنترل یک مسیر کنترلی مشترک یکپارچه شده‌اند. در مرحله پیاده‌سازی، معماری پیشنهادی در سطح RTL با استفاده از Verilog HDL توصیف شده و عملکرد منطقی آن با شبیه‌سازی و اعمال بردارهای آزمون بررسی شده است. سپس، برای ارزیابی سخت‌افزاری، طراحی با استفاده از فناوری CMOS ۶۵ نانومتر سنتز شده و شاخص‌هایی نظیر مساحت، تأخیر بحرانی و حاصل ضرب مساحت-تأخیر (ADP) استخراج و تحلیل شده‌اند. بدین ترتیب، ارزیابی نهایی صرفاً بر مبنای تحلیل نظری نبوده و قابلیت پیاده‌سازی و رفتار سخت‌افزاری معماری نیز بررسی شده است. بر مبنای مسئله و شکاف شناسایی شده، سؤال اصلی پژوهش به صورت زیر قابل طرح است: آیا می‌توان با بهره‌گیری از روابط اختصاصی مجموعه پنج‌پیمانه‌ای Class-F و اشتراک‌گذاری منابع سخت‌افزاری، یک واحد چندمنظوره عمومی طراحی کرد که عملیات حسابی و عملیات بین‌پیمانه‌ای اصلی RNS را با پیچیدگی، مساحت و تأخیر مناسب و قابل پیاده‌سازی پشتیبانی کند. فرضیه اصلی پژوهش نیز چنین صورت‌بندی می‌شود: انتظار می‌رود بهره‌گیری از روابط ساختاری Class-F، استفاده مجدد از اطلاعات و منابع مشترک و یکپارچه‌سازی عملیات حسابی، تبدیل معکوس، تشخیص علامت، مقایسه و Scaling در یک معماری واحد، بتواند افزونگی سخت‌افزاری را کاهش داده و ضمن حفظ صحت عملکرد، به عملکرد مناسبی از نظر مساحت، تأخیر و ADP در مقایسه با معماری‌های چندمنظوره و ساختارهای مرجع دست یابد.

۲) ساختار ریاضی و ویژگی‌های مجموعه پیمانه‌ای Class-F

مبنای طراحی واحد چندمنظوره پیشنهادی، مجموعه پنج‌پیمانه‌ای موسوم به Class-F است که به صورت رابطه (۱) تعریف می‌شود:

$$F(n) = \{2^n, 2^n - 1, 2^n + 1, 2^{n+1} - 1, 2^{n-1} - 1\} \quad (۱)$$

که در آن، برای طراحی موردنظر، n یک عدد صحیح زوج و دارای حداقل مقدار مجاز $n \geq 4$ است. در ادامه، پیمانه‌های مجموعه

به ترتیب با نمادهای رابطه (۲) نمایش داده می‌شوند:

$$m_1 = 2^n, m_2 = 2^n - 1, m_3 = 2^n + 1, m_4 = 2^{n+1} - 1, m_5 = 2^{n-1} - 1 \quad (۲)$$

یک عدد صحیح X در این سیستم به صورت بردار باقیمانده

$$X_R = (x_1, x_2, x_3, x_4, x_5) \quad (۳)$$

نمایش داده می‌شود که در آن $x_i = |X|_{m_i}$ است. مبنای انتخاب این مجموعه، وجود ساختارهای ویژه در پیمانه‌های آن است. در

این مجموعه، پیمانه $m_1 = 2^n$ با ساختار دودویی معمول سازگار است، در حالی که پیمانه‌های $2^n \pm 1, 2^{n+1} - 1, 2^{n-1} - 1$ امکان بهره‌گیری از جمع‌کننده‌های پیمانه‌ای ویژه، End-Around Carry، تاخوردگی بیت‌ها و ضرایب ثابت را فراهم می‌کنند. در نتیجه، می‌توان عملیات حسابی و تبدیل‌های موردنیاز را به ساختارهای سخت‌افزاری منظم‌تری نگاشت. معماری پنج‌پیمانه‌ای همچنین در مقایسه با مجموعه‌های چهارپیمانه‌ای، دامنه دینامیکی بیشتری را در اختیار قرار می‌دهد و در عین حال، وابستگی‌های جدیدی را در طراحی عملیات بین‌پیمانه‌ای ایجاد می‌کند.

نکته مهم قابل ذکر آن که؛ خانواده پیمانه‌ای مورد استفاده، پیش از این در قالب یک مجموعه پنج‌پیمانه‌ای انعطاف‌پذیر مورد مطالعه

قرار گرفته است [18]؛ بنابراین سهم این پژوهش در معرفی نخستین مجموعه پیمانه‌ای مذکور نیست، بلکه در صورت‌بندی آن به‌عنوان Class-F و توسعه یک معماری چندمنظوره اختصاصی برای بهره‌گیری از ویژگی‌های ساختاری آن است. به‌طور خاص، هدف آن است که روابط میان پیمانه‌ها نه فقط برای محاسبات درون‌پیمانه‌ای، بلکه برای طراحی مبدل معکوس، تشخیص علامت، مقایسه و Scaling نیز مورد استفاده قرار گیرد.

دامنه دینامیکی یک سیستم RNS برابر حاصل ضرب پیمانه‌های اول آن است. برای مجموعه Class-F، دامنه دینامیکی و مختصات

مربوط به آن، به صورت روابط (۴) تا (۱۱) به دست می‌آید:

$$M_F = \prod_{i=1}^5 m_i \quad (۴)$$

و در نتیجه:

$$M_F = (2^n - 1)2^n(2^n + 1)(2^{n+1} - 1)(2^{n-1} - 1) \quad (۵)$$

حال از آنجا که:

$$(2^n - 1)(2^n + 1) = 2^{2n} - 1 \quad (۶)$$

رابطه (۵) به شکل فشرده‌تر نوشته می‌شود:

$$M_F = 2^n(2^{2n} - 1)(2^{n+1} - 1)(2^{n-1} - 1) \quad (۷)$$

یا

$$M_F = (2^{2n} - 1)[2^n(2^{n+1} - 1)(2^{n-1} - 1)] \quad (۸)$$

بنابراین، تعداد بیت‌های موردنیاز برای نمایش دامنه دینامیکی را می‌توان از رابطه (۹) محاسبه کرد.

$$DR = \lfloor \log_2(M_F) \rfloor \quad (9)$$

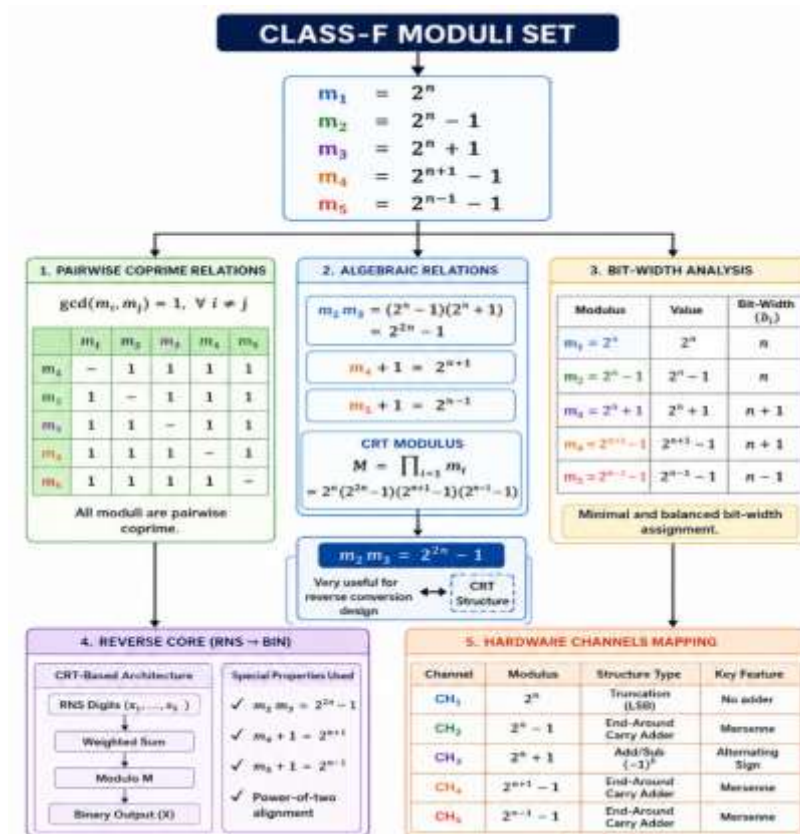
9

$$Q = (2^{2n} - 1)(2^{n+1} - 1)(2^{n-1} - 1) \quad (10)$$

بنابراین:

$$M_F = 2^n Q \quad (11)$$

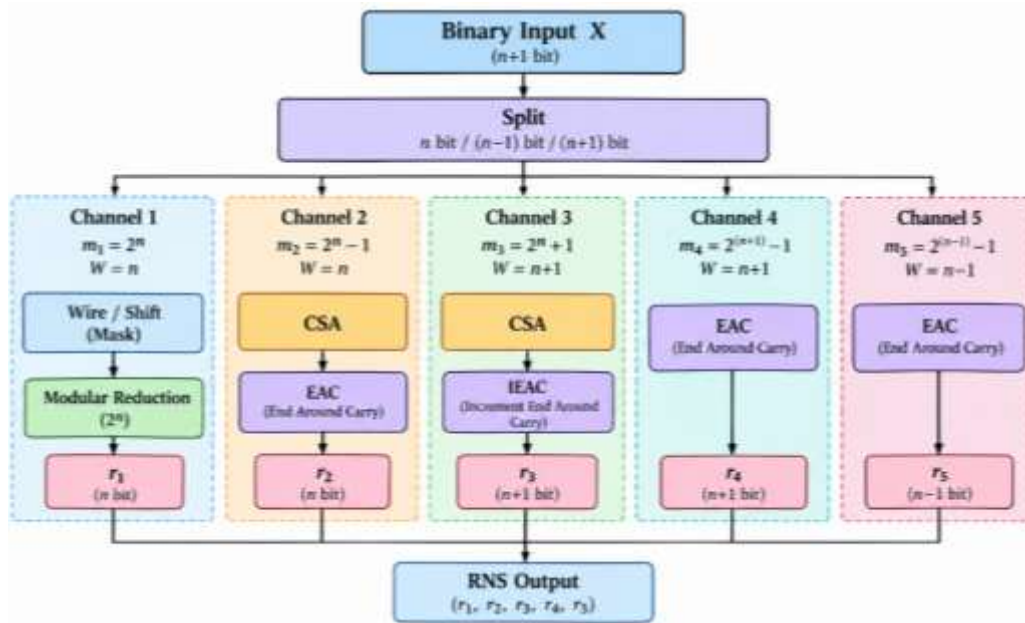
یعنی Class-F هم دارای یک جزء Mersenne بزرگ است و هم یک جزء مرکب چندعاملی و این دقیقاً همان ساختاری است که آن را از a-class و b-class متمایز می‌کند. علاوه بر آن، مطابق [18]، برای $n=4,6,8,\dots$ محدوده معادل به ترتیب ۱۹، ۲۹، ۳۹ و ... بیت است؛ این به آن معناست که مجموعه یک محدوده مؤثر $5n-1$ بیتی ایجاد می‌کند. نکته دیگر اینکه سه عامل حاصل، نسبت به یکدیگر مستقل و قابل ترکیب به وسیله CRT/MRC هستند. این تجزیه امکان طراحی Reverse Converter را در چند مرحله و با استفاده از سخت‌افزارهای قابل اشتراک فراهم می‌کند. با توجه به ساختار پیمانه‌ها، دامنه دینامیکی این مجموعه برای مقادیر مورد مطالعه تقریباً برابر n بیت است.



شکل (۳): دیگرام مشخصات معماری کلاس F

یکی از دلایل مهم انتخاب این مجموعه، قابلیت نگاشت مستقیم یا کم‌هزینه پیمانه‌ها به ساختارهای سخت‌افزاری ساده است. برای پیمانه $m_1 = 2^n$ عمل پیمانه‌ای اساساً با نگهداری n بیت کم‌ارزش ورودی انجام می‌شود. برای $m_2 = 2^n - 1$ خاصیت بازگشت نقلی به ابتدای کلمه امکان استفاده از End-Around Carry Adder (EAC) را فراهم می‌کند. در مقابل، پیمانه $m_3 = 2^n + 1$ به دلیل رابطه خاص آن با توان دوم، به ساختارهایی مبتنی بر شیف، جابه‌جایی و تغییر علامت نیاز دارد. دو پیمانه $m_4 = 2^{n+1} - 1$ و $m_5 = 2^{n-1} - 1$

1 نیز از خانواده پیمانه‌های $2^k - 1$ هستند و بنابراین می‌توانند از ساختارهای جمع پیمانه‌ای مبتنی بر End-Around Carry و فشرده‌سازی چند ورودی بهره ببرند. معماری سخت‌افزاری این کانال‌ها در شکل (۴) نمایش داده شده است.



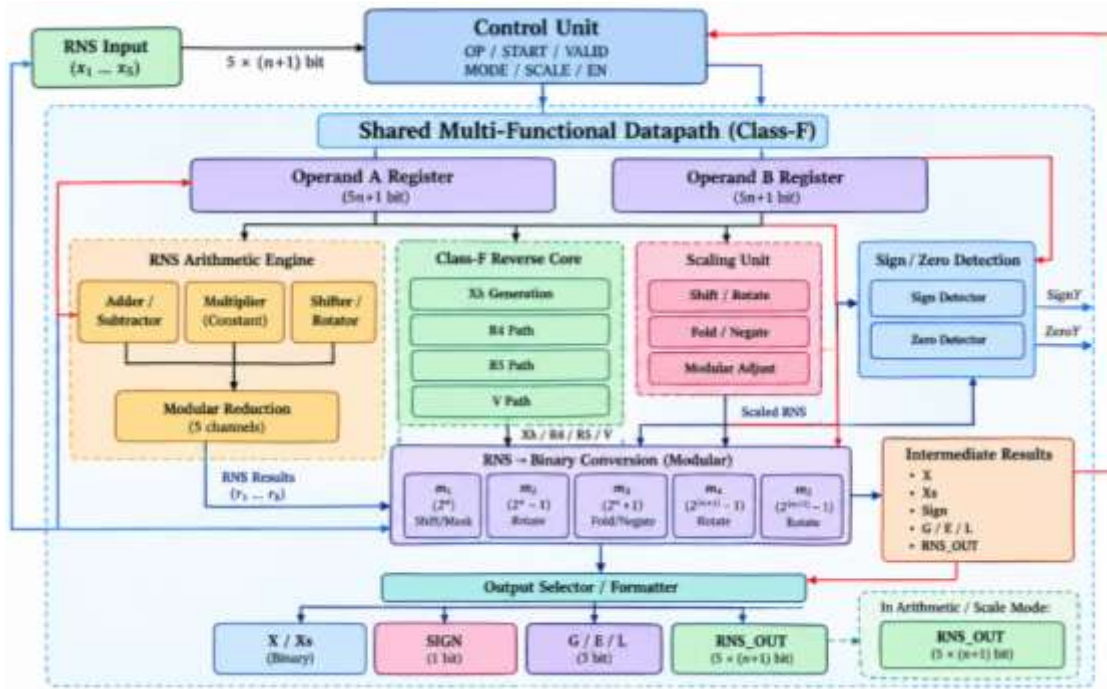
شکل(۴): معماری ترکیبی سخت‌افزاری کانالهای پیمانه‌ای

این ویژگی‌ها زمینه را برای طراحی یک مسیر داده مشترک فراهم می‌کنند؛ زیرا بسیاری از عملیات موردنیاز در معماری نهایی به جمع، تفریق، شیف، تغییر علامت، مقایسه و ضرب در ضرایب ثابت کاهش می‌یابند. از این رو، می‌توان بخشی از عناصر سخت‌افزاری تولیدشده برای یک عملیات را در عملیات دیگر نیز به کار گرفت. همین قابلیت، مبنای اصلی رویکرد اشتراک‌گذاری منابع سخت‌افزاری در واحد چندمنظوره پیشنهادی است.

ویژگی مهم دیگر مجموعه Class-F، وجود روابط منظم میان پیمانه‌هاست که در استخراج روابط تبدیل معکوس و پردازش‌های بین‌پیمانه‌ای مورد استفاده قرار می‌گیرد. در معماری پیشنهادی، به جای آنکه تبدیل RNS به دودویی به عنوان یک بلوک کاملاً مستقل از سایر عملیات در نظر گرفته شود، مسیر تبدیل به مجموعه‌ای از مؤلفه‌های میانی تقسیم شده است. مهم‌ترین این مؤلفه‌ها شامل X_h, R_4, L_5, R_5, V هستند و مسیر کلی تبدیل را می‌توان به صورت:

$$(x_1, x_2, x_3) = X_h \rightarrow R_4 \rightarrow L_5 \rightarrow R_5 \rightarrow V \rightarrow X \quad (12)$$

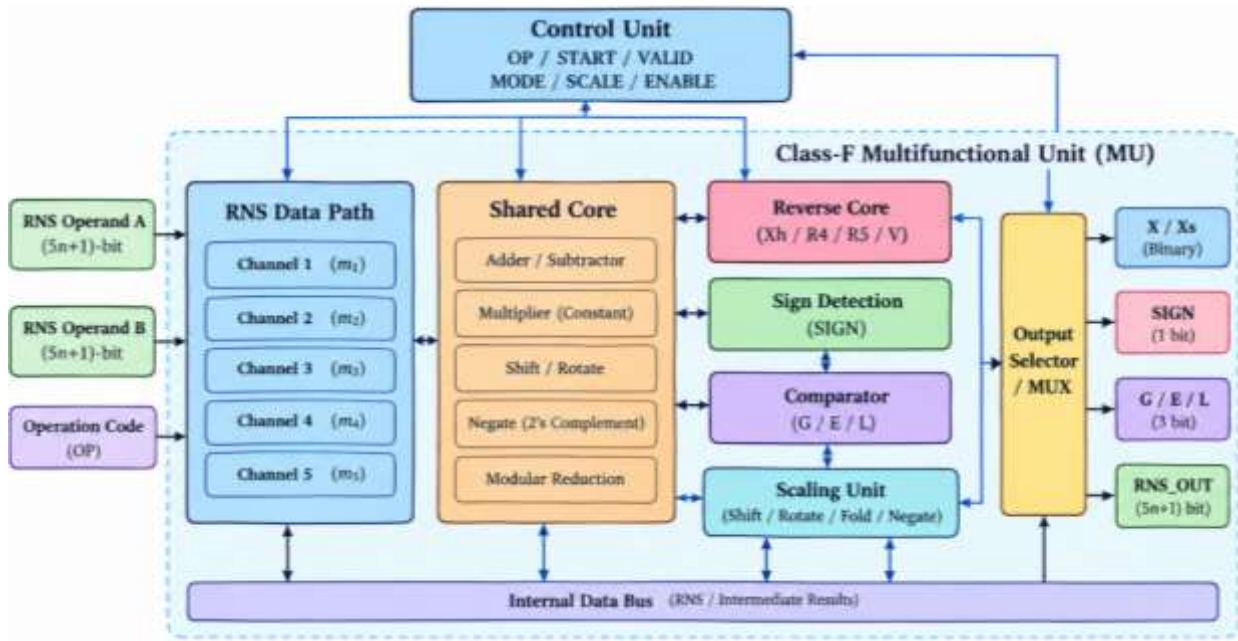
نمایش داد. اهمیت این ساختار برای پژوهش حاضر در آن است که تمام این مؤلفه‌ها صرفاً برای تولید خروجی دودویی مورد استفاده قرار نمی‌گیرند؛ بلکه بخشی از آن‌ها در تشخیص علامت و سایر عملیات بین‌پیمانه‌ای نیز قابل استفاده مجدد هستند. این ایده، مبنای انتقال از یک مجموعه‌ای از واحدهای مستقل به یک معماری چندمنظوره مشترک است. ساختار کامل این جریان داده در شکل(۵) نمایش داده شده است.



شکل (۵): ساختار کامل جریان داده مسیر تبدیل به مجموعه‌ای از مؤلفه‌های میانی

۳) معماری پیشنهادی واحد چندمنظوره Class-F

بر اساس ویژگی‌های ریاضی مجموعه Class-F و با هدف کاهش افزونگی ناشی از پیاده‌سازی مستقل عملیات بین‌پیمانه‌ای، در این پژوهش یک واحد چندمنظوره عمومی طراحی شده است. ایده اصلی معماری آن است که عملیات مختلف موردنیاز در یک پردازشگر مبتنی بر RNS، شامل محاسبات پیمانه‌ای، تبدیل معکوس، تشخیص علامت، مقایسه اعداد علامت‌دار و Scaling، بر روی مجموعه‌ای از منابع مشترک سازمان‌دهی شوند. در این ساختار، برخلاف رویکردی که برای هر عملیات یک مسیر سخت‌افزاری مستقل در نظر می‌گیرد، داده‌ها و برخی مؤلفه‌های محاسباتی در چند عملیات مورد استفاده مجدد قرار می‌گیرند. این رویکرد، ضمن حفظ استقلال عملکردی عملیات، امکان کنترل بهتر هزینه سخت‌افزاری و کاهش تکرار منابع را فراهم می‌کند. معماری پیشنهادی از یک واحد کنترل مرکزی، مسیر داده RNS، واحدهای محاسبات پیمانه‌ای، هسته مشترک تبدیل معکوس، منطق تشخیص علامت، واحد مقایسه و واحد Scaling تشکیل شده است. ورودی اصلی، دو عملوند در قالب نمایش RNS و یک کد عملیات است و واحد کنترل بر مبنای این کد، مسیر مناسب پردازش و منابع موردنیاز را فعال می‌کند. برای عملیات مستقل، مسیرهای پردازشی می‌توانند به صورت مجزا عمل کنند و در عملیاتی نظیر مقایسه دو عدد، از دو مسیر داده و یا یک هسته مشترک به صورت ترتیبی استفاده شود. این ساختار همچنین امکان حذف برخی زیرواحدهای غیرضروری در کاربردهای خاص را فراهم می‌سازد و در نتیجه، معماری از یک ساختار ثابت به یک ساختار قابل پیکربندی تبدیل می‌شود.



شکل (۶): معماری واحد چندمنظوره مبتنی بر اشتراک منابع سخت‌افزاری در سیستم Class-F

رفتار کلی واحد را می‌توان به صورت یک تابع چندحالتی از ورودی‌های RNS و کد عملیات بیان کرد:

$$F_{MFU}(A_R, B_R, OP, S) = \begin{cases} A_R + RB_R & OP = ADD \\ A_R - RB_R & OP = SUB \\ A_R \times RB_R & OP = MUL \\ Scale_{2^S}(A_R) & OP = Scale \\ RC_F(A_R) & OP = RC \\ SD_F(A_R) & OP = SIGN \\ CMP_F(A_R, B_R) & OP = CMP \\ SREV_F(A_R) & OP = SREV \end{cases} \quad (13)$$

در این رابطه، A_R و B_R نمایش‌های مانده‌ای دو عملوند، S ضریب Scaling، RC_F مبدل معکوس Class-F و SD_F آشکارساز علامت هستند. مزیت چنین تعریف عملکردی آن است که عملیات مختلف، از دید معماری در یک رابط یکسان قرار می‌گیرند و کنترل‌کننده می‌تواند با فعال‌سازی منابع لازم، تنها مسیر موردنیاز را وارد فرآیند محاسبه کند.

نخستین لایه معماری پیشنهادی، هسته محاسبات پیمانه‌ای است که عملیات پایه جمع، تفریق و ضرب را در پنج کانال Class-F انجام می‌دهد. از آنجا که کانال‌ها از نظر پیمانه‌ای مستقل هستند، محاسبات به صورت موازی انجام می‌شوند و مسیر نقلی میان کانال‌های مختلف ایجاد نمی‌شود. با این حال، ساختار هر کانال متناسب با ویژگی پیمانه خود انتخاب شده است. برای $m_1 = 2^n$ ، داده مستقیماً در قالب دودویی n بیتی پردازش می‌شود. برای $m_2 = 2^n - 1$ ، ساختار جمع پیمانه‌ای مبتنی بر End-Around Carry مورد استفاده قرار می‌گیرد. کانال $m_3 = 2^n + 1$ به دلیل ماهیت پیمانه، از منطق اختصاصی مبتنی بر شیفت، تغییر علامت و جمع پیمانه‌ای بهره می‌برد. دو کانال $m_4 = 2^{n+1} - 1$ و $m_5 = 2^{n-1} - 1$ (با توجه به فرم $2^k - 1$) قابلیت استفاده از فشرده‌سازی چندورودی و EAC را دارند. در پیاده‌سازی، این تفاوت‌ها باعث می‌شود که یک هسته محاسباتی یکسان برای همه کانال‌ها مناسب نباشد و معماری، از مجموعه‌ای از واحدهای تخصصی ولی هماهنگ استفاده کند. ساختار واحدهای حسابی در نهایت به مسیر داده مشترک واحد چندمنظوره متصل می‌شود و خروجی عملیات انتخاب‌شده از طریق منطق انتخاب و کنترل به مرحله بعد منتقل می‌گردد.

یکی از مهم‌ترین بخش‌های معماری پیشنهادی، هسته Reverse Converter است. تبدیل معکوس در RNS ذاتاً عملیاتی بین‌پیمانه‌ای است و در صورت طراحی مستقل می‌تواند موجب افزایش تعداد جمع‌کننده‌ها، عمق منطقی و هزینه سخت‌افزاری شود. در معماری Class-F پیشنهادی، این تبدیل بر اساس یک ساختار مرحله‌ای انجام می‌شود که در آن ابتدا مؤلفه X_h از سه کانال اول محاسبه شده و سپس با استفاده از اطلاعات کانال‌های چهارم و پنجم، مؤلفه‌های R_4 ، L_5 ، R_5 و در نهایت V تشکیل می‌شوند.

در مرحله نخست، رابطه مربوط به X_h بر اساس ضرایب ثابت حاصل از معکوس پیمانه‌های m_2 و m_3 محاسبه می‌شود:

$$X_h = \left| \left(m_3 x_2 \left| \frac{1}{m_3} \right|_{m_2} + m_2 x_3 \left| \frac{1}{m_2} \right|_{m_3} - x_1 \right) \left| \frac{1}{m_1} \right|_{m_2 m_3} \right|_{m_2 m_3} \quad (14)$$

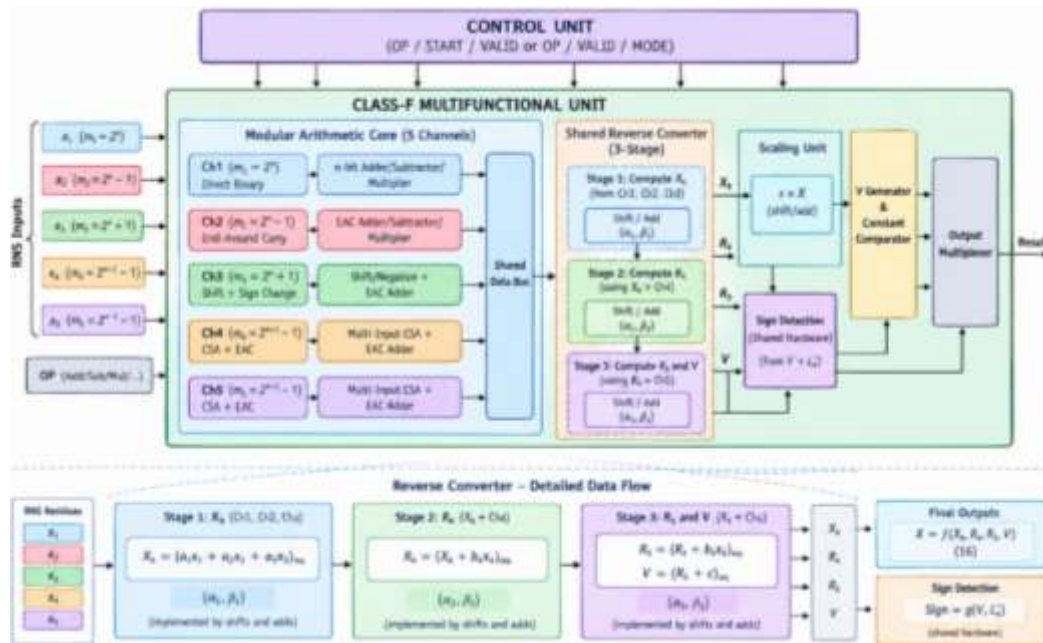
برای مجموعه Class-F، ضرایب فوق به ساختارهای ثابت مبتنی بر توان‌های دو تبدیل می‌شوند و بنابراین امکان پیاده‌سازی سخت‌افزاری با شیف و جمع را فراهم می‌کنند. در ادامه، با استفاده از اطلاعات X_h و باقی‌مانده کانال چهارم، R_4 محاسبه می‌شود و سپس مؤلفه مربوط به کانال پنجم برای تولید R_5 به‌دست می‌آید. در انتها:

$$V = X_h + m_2 m_3 R_4 + m_2 m_3 m_4 R_5 \quad (15)$$

و مقدار باینری متناظر از رابطه:

$$X = x_1 + 2^n V \quad (16)$$

تولید می‌شود. نکته کلیدی معماری پیشنهادی آن است که همه این اطلاعات فقط برای تولید X استفاده نمی‌شوند. بخش‌هایی از همین مسیر میانی، به‌ویژه V مستقیماً در تشخیص علامت و برخی پردازش‌های وابسته مورد استفاده قرار می‌گیرند. در نتیجه، به‌جای قرار دادن یک Sign Detector مستقل در کنار Reverse Converter، بخشی از سخت‌افزار تبدیل معکوس به‌صورت مشترک مورد استفاده قرار می‌گیرد. این همان نقطه‌ای است که ایده Hardware Reuse در معماری پیشنهادی از یک مفهوم کلی به یک سازوکار مشخص سخت‌افزاری تبدیل می‌شود.



شکل (۷): معماری پیشنهادی واحد چندمنظوره Class-F با اشتراک سخت‌افزار میان Reverse Converter، Scaling و

Sign Detection

تشخیص علامت در معماری پیشنهادی به صورت مستقل از تبدیل کامل انجام نمی‌شود. با توجه به نمایش عدد به صورت رابطه (۱۶)

$$X = x_1 + 2^n V \quad (16)$$

مقدار V بخش بارزتر عدد را مشخص می‌کند و می‌توان تصمیم مربوط به علامت را بدون تشکیل کامل X انجام داد. برای مجموعه Class-F، آستانه تصمیم از نصف دامنه دینامیکی استخراج می‌شود. با تعریف:

$$M = 2^n Q, \quad Q = 2q + 1 \quad (17)$$

9

$$q = 2^{4n-1} - 5 \times 2^{3n-2} + 5 \times 2^{n-2} - 1 \quad (18)$$

علامت به صورت زیر تعیین می‌شود:

$$Sign(X) = \begin{cases} 0. & V < q \\ 1. & V > q \\ x_{1,n-1}. & V = q \end{cases} \quad (19)$$

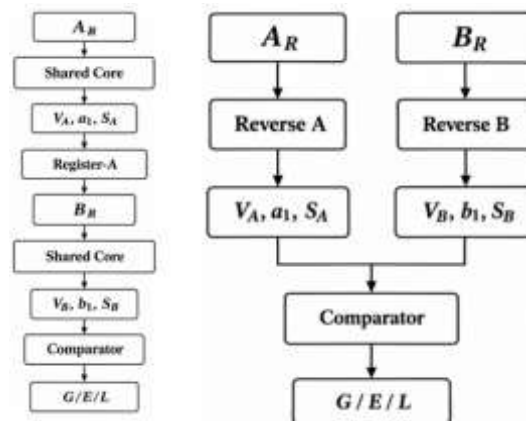
بنابراین، تشخیص علامت را می‌توان به یک مقایسه با ثابت و یک بررسی بیت مرزی کاهش داد:

$$Sign = (V > q) \vee [(V = q) \wedge x_{1,n-1}] \quad (20)$$

این ساختار باعث می‌شود که برای تولید بیت علامت، مسیر کامل CPA نهایی مبدل معکوس الزام‌آور نباشد و در نتیجه پاسخ علامت بتواند زودتر از خروجی کامل باینری تولید شود. علاوه بر این، سخت‌افزار تولیدکننده V در هر دو عملیات Reverse Conversion و Sign Detection استفاده می‌شود و از تکرار بخش مهمی از مسیر محاسبات جلوگیری می‌کند. این ایده به عنوان یکی از مهم‌ترین وجوه استفاده مجدد از منابع در این پژوهش، مطرح شده است. پس از تولید علامت، مقایسه دو عدد علامت‌دار در معماری Class-F بر پایه ترکیب علامت و بخش مقداری انجام می‌شود. اگر دو عدد دارای علامت متفاوت باشند، عدد مثبت بزرگ‌تر است؛ در غیر این صورت، مقایسه بخش مقداری آن‌ها انجام می‌شود. با نمایش

$$A = a_1 + 2^n V_A, \quad B = b_1 + 2^n V_B \quad (21)$$

و تعریف مقایسه بدون علامت، می‌توان روابط بزرگ‌تر، کوچک‌تر و مساوی را از ترکیب V_A, V_B, a_1 و b_1 به دست آورد. این ساختار سبب می‌شود که مقایسه نیز از داده‌های حاصل از هسته مشترک بهره‌بردار و نیاز به یک ساختار کاملاً مستقل کاهش یابد. در این پژوهش، برای این بخش دو پیکربندی موازی و ترتیبی در نظر گرفته شده است؛ در ساختار موازی، دو مسیر Reverse به صورت هم‌زمان فعال می‌شوند، در حالی که در ساختار ترتیبی، یک هسته Reverse با استفاده از رجیسترهای میانی بین دو عملوند به صورت اشتراکی مورد استفاده قرار می‌گیرد.



شکل (۸): نمایش پیکربندی موازی (راست) و پیکربندی ترتیبی (چپ) مقایسه‌کننده پیشنهادی

واحد Scaling نیز متناسب با ماهیت هر پیمانه طراحی شده است. برای پیمانه 2^n Scaling با شیفت و نگهداری بخش موردنیاز بیت‌ها انجام می‌شود؛ برای پیمانه‌های $1 - 2^{n-1}$ ، $1 - 2^{n+1}$ و $2^n - 1$ از ساختارهای مبتنی بر Rotation استفاده می‌شود؛ در حالی که پیمانه $2^n + 1$ به منطق تغییر علامت متناسب با شیفت نیاز دارد. در نتیجه، Scaling در Class-F یک مدار یکسان برای تمامی کانال‌ها نیست، بلکه مجموعه‌ای از عملیات ساده و متناسب با ساختار پیمانه‌ای هر کانال است. مهم‌تر آنکه واحد Scaling در معماری نهایی به‌عنوان یک بلوک قابل استفاده مجدد در کنار هسته Reverse قرار گرفته است.

اصل طراحی معماری پیشنهادی را می‌توان در مفهوم Reuse Data و Reuse Hardware خلاصه کرد. در این رویکرد، چند عملکرد مستقل الزاماً دارای سخت‌افزارهای مستقل نیستند، بلکه مجموعه‌ای از منابع محاسباتی و داده‌های میانی میان آن‌ها به اشتراک گذاشته می‌شود. مهم‌ترین منابع قابل اشتراک شامل مسیر تولید V ، بخشی از منطق جمع و فشرده‌سازی، رجیسترهای میانی، ضرایب ثابت و برخی اجزای مقایسه و Scaling هستند. بطور مثال، V که در تبدیل معکوس برای تولید خروجی باینری استفاده می‌شود، در تشخیص علامت نیز مستقیماً مورد استفاده قرار می‌گیرد. همچنین در Scaling در حالتی که خروجی باینری موردنیاز باشد، به‌جای طراحی یک مبدل مستقل، همان هسته تبدیل مشترک مورد استفاده قرار می‌گیرد. به این ترتیب، چند عملیات می‌توانند بر روی یک زیرساخت داده و سخت‌افزاری مشترک کار کنند. این رویکرد به‌عنوان مبنای یکپارچه‌سازی نهایی واحد چندمنظوره می‌باشد.

در سطح معماری، واحد کنترل مرکزی وظیفه دارد بر اساس کد عملیات، مسیر داده مناسب را فعال، رجیسترهای موردنیاز را کنترل و خروجی مناسب را انتخاب کند. این سازمان‌دهی اجازه می‌دهد که یک نسخه پایه از معماری برای چند کاربرد مختلف مورد استفاده قرار گیرد. همچنین، در کاربردهایی که برخی عملیات موردنیاز نیستند، زیرواحدهای متناظر می‌توانند در زمان سنتز حذف شوند؛ برای مثال، در کاربردی که فقط عملیات حسابی و تبدیل معکوس مورد نیاز است، بخش مقایسه‌کننده ضرورت ندارد. بنابراین، معماری پیشنهادی تنها یک واحد چندمنظوره ثابت نیست، بلکه یک چارچوب پارامتریک و قابل پیکربندی برای پیاده‌سازی پردازنده‌های RNS مبتنی بر Class-F فراهم می‌کند. در مجموع، معماری پیشنهادی با ترکیب عملیات حسابی و عملیات دشوار بین‌پیمانه‌ای در یک چارچوب مشترک، تلاش می‌کند سه هدف اصلی را به‌صورت هم‌زمان محقق کند: حفظ مزایای موازی‌سازی RNS، کاهش افزونگی سخت‌افزاری و کنترل هزینه زمانی و مساحتی معماری.

۴) پیاده‌سازی، اعتبارسنجی و ارزیابی عملکرد معماری پیشنهادی

پس از تکمیل طراحی ریاضی و معماری واحد چندمنظوره Class-F، ساختار پیشنهادی در سطح RTL پیاده‌سازی و برای ارزیابی عملی شبیه‌سازی شده است. در این مرحله، تمامی بلوک‌های اصلی شامل هسته محاسباتی، واحدهای تبدیل مستقیم و معکوس، کنترل‌کننده، مسیر داده و رجیسترهای موردنیاز به‌صورت ماژول‌های مستقل توسعه داده شده‌اند. توصیف سخت‌افزاری معماری با استفاده از Verilog HDL انجام شده و شبیه‌سازی عملکردی با ModelSim SE صورت گرفته است. همچنین Vivado برای بررسی اولیه ساختار RTL و قابلیت پیاده‌سازی سخت‌افزاری مورد استفاده قرار گرفته است. ارزیابی نهایی شاخص‌های ASIC نیز با استفاده از ابزارهای سنتز و تحلیل زمانی Synopsys و کتابخانه استاندارد سلول فناوری 65 nm CMOS انجام شده است.

در مرحله اعتبارسنجی، عملکرد واحدهای داخلی و نیز رفتار واحد چندمنظوره یکپارچه با اعمال بردارهای آزمون شامل حالت‌های عملیاتی مختلف و شرایط مرزی بررسی شده است. خروجی‌های حاصل از شبیه‌سازی با روابط ریاضی استخراج‌شده برای اجزای مختلف معماری تطبیق داده شده‌اند. بنابراین، ارزیابی عملکردی بر بررسی سازگاری پیاده‌سازی RTL با مدل ریاضی و معماری ارائه‌شده استوار بوده است. اعتبارسنجی در دو سطح انجام شده است. در سطح نخست، زیرواحدهای منفرد شامل واحدهای محاسبات پیمانه‌ای، مبدل مستقیم، مبدل معکوس، آشکارساز علامت، مقایسه‌گر و Scaling به‌صورت مستقل مورد بررسی قرار گرفته‌اند. در سطح دوم، تمامی این زیرواحدها در قالب واحد چندمنظوره یکپارچه شده و رفتار کلی معماری با توجه به کد عملیات مورد بررسی قرار گرفته است. در این فرایند، ورودی‌های RNS به واحد اعمال شده و خروجی هر عملیات با نتیجه حاصل از روابط ریاضی متناظر مقایسه شده است. اهمیت این مرحله در معماری حاضر بیش از یک بررسی صرفاً منطقی است؛ زیرا بسیاری از عملیات در MU بر پایه داده‌های میانی مشترک انجام می‌شوند. بنابراین، اعتبارسنجی باید علاوه بر صحت خروجی نهایی، سازگاری انتقال داده میان هسته Reverse، آشکارساز علامت، Comparator و Scaling را نیز نشان دهد. به‌ویژه در طراحی پیشنهادی، تشخیص علامت از اطلاعات تولیدشده در مسیر تبدیل معکوس استفاده می‌کند و Scaling نیز در برخی حالت‌ها به همان هسته تبدیل متکی است. در نتیجه، صحت این مسیرهای مشترک برای عملکرد صحیح چند عملیات وابسته اهمیت اساسی دارد.

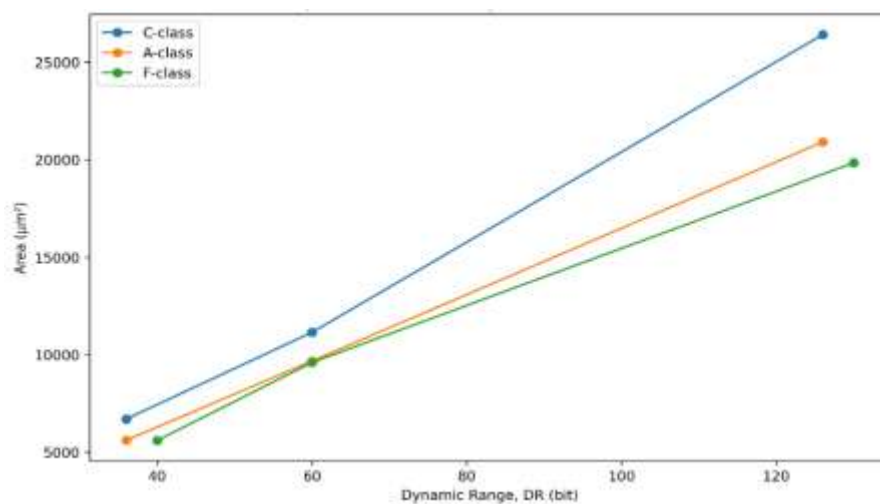
جدول (۱): اعتبارسنجی عملکردی هر یک از زیرواحدهای طراحی‌شده

بلوک	نوع آزمون	حالات مرزی	مرجع صحت
Direct Converter	Binary $\rightarrow$ RNS	$M-1, 0$ و مقادیر نزدیک پیمانه	مدل ریاضی
Reverse Converter	RNS $\rightarrow$ Binary	$0, 1, \frac{M}{2}-1, \frac{M}{2}, \frac{M}{2}+1, M-2, M-1$	CRT / محاسبات فصل ۴
Sign Detector	Sign	دو طرف $\frac{M}{2}$	روابط ریاضی
Comparator	G/E/L	برابر، مثبت/منفی، نزدیک‌ترین مقادیر	روابط ریاضی
Scaling	Signed/Unsigned	صفر، مرزها، مقادیر حداکثری	روابط Scaling
Complete MFU	End-to-End	ترکیب عملیات	Golden Model

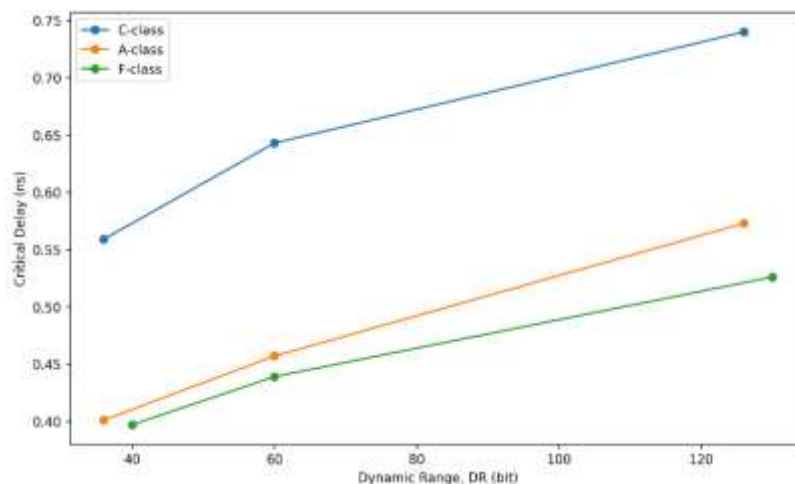
پس از اعتبارسنجی عملکردی، معماری پیشنهادی از نظر شاخص‌های سخت‌افزاری ارزیابی شده است. معیارهای اصلی شامل مساحت مدار (Area)، تأخیر بحرانی (Critical Delay) و حاصل ضرب مساحت-تأخیر (ADP) هستند. برای بررسی رفتار معماری در مقیاس‌های مختلف، سه مقدار $n=8$ ، $n=12$ و $n=26$ انتخاب شده‌اند که به‌ترتیب دامنه‌های دینامیکی حدود ۴۰، ۶۰ و ۱۳۰ بیت را ایجاد می‌کنند. لازم به تأکید است که به دلیل محدودیت زوج‌بودن n ، نقاط اول و سوم دقیقاً با دامنه‌های مرجع ۳۶ و ۱۲۶ بیت یکسان نیستند؛ از این‌رو مقایسه در این دو نقطه بر اساس نزدیک‌ترین پیکربندی مجاز انجام شده است، در حالی که $n=12$ تقریباً با نقطه ۶۰ بیتی منطبق است.

جدول (۲): نتایج سنتز واحد چندمنظوره پیشنهادی Class-F و معماری های مرجع

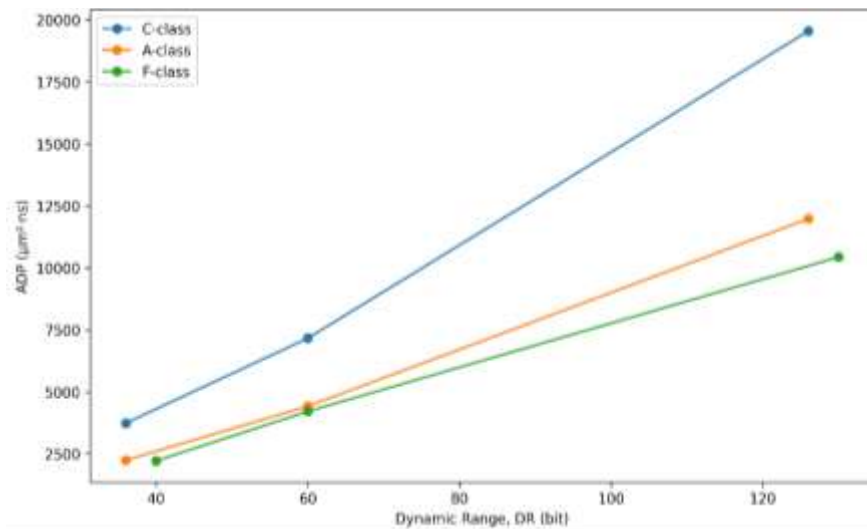
ADP ($\mu\text{m}^2 \cdot \text{ns}$)	Delay (ns)	Area (μm^2)	Circuit	DR	Set Class
۳۷۴۲.۴۴۹۱	۰.۵۵۹	۶۶۹۴.۹	Multifunctional unit [۱۲]	۳۶	C-class
۷۱۶۴.۰۴۸۸	۰.۶۴۳	۱۱۱۴۱.۶	Multifunctional unit [۱۲]	۶۰	C-class
۱۹۵۴۳.۹۱۸۰	۰.۷۴۰	۲۶۴۱۰.۷	Multifunctional unit [۱۲]	۱۲۶	C-class
۲۲۴۵.۸۰۰۵	۰.۴۰۱	۵۶۰۰.۵	Multifunctional unit [۱۲]	۳۶	A-class
۴۴۱۸.۶۸۷۳	۰.۴۵۷	۹۶۶۸.۹	Multifunctional unit [۱۲]	۶۰	A-class
۱۱۹۷۹.۹۴۰۲	۰.۵۷۳	۲۰۹۰۷.۴	Multifunctional unit [۱۲]	۱۲۶	A-class
۲۲۲۰.۳۸۱۳	۰.۳۹۷	۵۵۹۲.۹	Proposed Multifunctional unit	$39.985842 = 8F(n=$	F-class
۴۲۱۱.۸۹۷۷	۴۳۹۰.	۹۵۹۴.۳	Proposed Multifunctional unit	$59.999119F(n=12)=$	F-class
۱۰۴۳۴.۲۰۹۴	۰.۵۲۶	۱۹۸۳۶.۹	Proposed Multifunctional unit	$129.999999946 = 26F(n=$	F-class



شکل (۹): مقایسه Area بر حسب DR سه کلاس



شکل (۱۰): نمودار مقایسه Critical Delay سه کلاس بر حسب DR



شکل (۱۱): نمودار مقایسه ADP سه کلاس بر حسب DR

از نتایج مشخص است که:

$$T_F < T_A < T_C$$

$$ADP_8 < ADP_{12} < ADP_{26}$$

و علت این اتفاق را می‌توان به ساختار واقعی معماری Class F، عرض مسیرها، نوع CSA/CPA، اشتراک سخت‌افزار و نحوه استفاده

از اطلاعات میانی نسبت داد.

جدول (۳): اختلاف DR با مرجع [12]

اختلاف DR	DR واقعی F	پیکربندی F	DR مرجع [۱۲]
+۳.۹۸۵۸۴۲ bit	۳۹.۹۸۵۸۴۲	n=۸	۳۶
-۰.۰۰۰۸۸۱ bit	۵۹.۹۹۹۱۱۹	n=۱۲	۶۰
+۳.۹۹۹۹۹۹۴۶ bit	۱۲۹.۹۹۹۹۹۹۴۶	n=۲۶	۱۲۶

مطابق جدول (۳) مقایسه در نقاط اول و سوم بر مبنای نزدیک‌ترین مقدار مجاز دامنه دینامیکی برای n زوج انجام شده است و در نقطه میانی تقریباً انطباق کامل دامنه دینامیکی وجود دارد. نتایج نشان می‌دهد که با افزایش دامنه دینامیکی، مساحت و تأخیر هر سه معماری افزایش می‌یابد؛ با این حال، معماری پیشنهادی Class-F در هر سه نقطه بررسی‌شده رفتار رقابتی نشان می‌دهد. نتایج نشان می‌دهد که مزیت اصلی معماری پیشنهادی صرفاً در افزایش دامنه دینامیکی نیست، بلکه در نگاشت مناسب روابط Class-F به سخت‌افزار و سازمان‌دهی مشترک منابع نیز ریشه دارد.

۵) نتیجه‌گیری

مجموع نتایج پیاده‌سازی نشان می‌دهد که معماری پیشنهادی توانسته است یک مجموعه متنوع از عملیات RNS را در قالب یک ساختار واحد و قابل استفاده مجدد پیاده‌سازی کند. مزیت اصلی معماری، ترکیب چند عملیات مستقل در یک زیرساخت مشترک است؛ به‌ویژه، استفاده مجدد از داده‌های میانی هسته Reverse Converter در تشخیص علامت و بهره‌گیری از همان زیرساخت در مسیرهای وابسته به Scaling، از ایجاد سخت‌افزار اضافی جلوگیری می‌کند. در سطح کل واحد چندمنظوره نیز نتایج سنتز برای سه مقیاس مختلف نشان می‌دهد

که با افزایش دامنه دینامیکی، هزینه سخت‌افزاری افزایش می‌یابد، اما معماری پیشنهادی در مقایسه با معماری‌های مرجع Class-A و Class-C همچنان عملکرد رقابتی خود را حفظ می‌کند.

از این رو، نتایج تجربی، فرضیه اصلی پژوهش را پشتیبانی می‌کنند؛ به این معنا که اشتراک‌گذاری منابع و استفاده مجدد از اطلاعات میانی در یک واحد چندمنظوره Class-F می‌تواند بدون حذف قابلیت‌های اصلی RNS، افزونگی سخت‌افزاری را کنترل کرده و هم‌زمان بهبود مناسبی در شاخص‌های مساحت، تأخیر و ADP ایجاد کند.

تعارض منافع

در انجام مطالعه حاضر، هیچ‌گونه تضاد منافع وجود ندارد.

مشارکت نویسندگان

در نگارش این مقاله تمامی نویسندگان نقش یکسانی ایفا کردند.

موازن اخلاقی

در انجام این پژوهش تمامی موازن و اصول اخلاقی رعایت گردیده است.

شفافیت داده‌ها

داده‌ها و مآخذ پژوهش حاضر در صورت درخواست از نویسنده مسئول و ضمن رعایت اصول کپی رایت ارسال خواهد شد.

حامی مالی

این پژوهش حامی مالی نداشته است.

References

- [1] Ahsan, J., et al. (2022). Efficient FPGA implementation of RNS Montgomery multipliers. *VLSI Journal / ACM*.
- [2] Akkal, M. (2007). A new Mixed Radix Conversion algorithm MRC-II. *Computers & Mathematics with Applications*.
- [3] Alsuhli, G. (2023). Number Systems for Deep Neural Network Architectures. *arXiv*.
- [4] Alvani, D. (2024). Fast And Area-Efficient Reverse Converters For Five Moduli Set $\{2n+1, 2n-1, 2n, 2n+11, 2n-1\}$. *Educational Administration: Theory and Practice*, 30(5), 2385-2400.
- [5] Chakraborti, N. B., & Sinha, S. (1986). An implementation of mixed-radix conversion for residue number systems. *IEEE Transactions on Computers*.
- [6] Cao, B., Chang, C. H., & Srikanthan, T. (2007). A residue-to-binary converter for a new five-moduli set. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 54(5), 1041-1049.
- [7] Garner, H. (1959). The residue number system. *IRE Transactions on Electronic Computers*, EC-8(2), 140-147.
- [8] Jaberipur, G., et al. (2014). A ROM-less reverse RNS converter for moduli set $\{2^q \pm 1, 2^q, \dots\}$. *IET Computers & Digital Techniques*.
- [9] Kuchukov, V. (2022). Performance analysis of hardware implementations of RNS reverse conversion methods. *Applied Sciences*, 12(23), 12355.
- [10] Menezes, A., van Oorschot, P., & Vanstone, S. (1997). *Handbook of applied cryptography*. CRC Press.
- [11] Ming, X., Hu, S., & He, S. (2011). High-performance residue number system for digital signal processing. *IEEE Transactions on Signal Processing*, 59(9), 3771-3782.

- [12] Molahosseini, A. S., Emrani Zarandi, A. A., Martins, P., & Sousa, L. (2017). A multifunctional unit for designing efficient RNS-based datapaths. *IEEE Access*, 5, 25972–25983.
- [13] Nature / Research Intelligence. Residue number system architectures & applications (overview). *Nature*.
- [14] Nguyen, T. (2013). Fast modular exponentiation using residue domain (Shenoy–Kumaresan). *University report*.
- [15] Niras, C. V., & Kong, Y. (2016). Fast sign-detection algorithm for residue number system moduli set $\{2n-1, 2n, 2n+1-1\}$. *IET Computers & Digital Techniques*, 10(2), 54-58.
- [16] Omondi, A., & Premkumar, B. (2007). *Residue number systems: Theory and applications*. Imperial College Press.
- [17] Parhami, B. (2000). *Computer arithmetic: Algorithms and hardware designs*. Oxford University Press.
- [18] Patronik, P., & Piestrak, S. J. (2017). Design of Reverse Converters for a New Flexible RNS Five-Moduli Set $\{2^k, 2^{n-1}, 2^{n+1}, 2^{n+1-1}, 2^{n-1-1}\}$ (n Even). *Circuits, Systems, and Signal Processing*, 36(11), 4593-4614.
- [19] Salamat, S., et al. (2021). Residue-Net: Multiplication-free neural network by in-situ RNS. *ResidueNet Project, UCSD*.
- [20] Shenoy, A. P., & Kumaresan, R. (1989). Fast base extension using a redundant modulus in RNS. *IEEE Transactions on Computers*, 38(2), 292–297.
- [21] Szabo, N. S., & Tanaka, R. I. (1967). *Residue arithmetic and its applications to computer technology*. McGraw-Hill.
- [22] Various review articles (2020–2022). RNS reverse converters and base extension. *IJSTR (2020); MDPI Applied Sciences (2022)*.
- [23] Wang, Y. (1984). Residue number system arithmetic: Modern applications in digital signal processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2), 152–160.
- [24] Mojahed, M., Molahosseini, A. S., & Zarandi, A. A. E. (2021). Multifunctional unit for reverse conversion and sign detection based on five-moduli set $\{2^{2n}, 2^{n+1}, 2^{n-1}, 2^{n+3}, 2^{n-3}\}$. *Computer Science*, 22, 101-121.
- [25] Kumar, S., & Chang, C. H. (2017). A Scaling-Assisted Signed Integer Comparator for the Balanced Five-Moduli Set RNS $\{2^n-1, 2^n, 2^{n+1}, 2^{n+1}-1, 2^{n-1}-1\}$. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 25(12), 3521-3533.