تعیین دامنه برای به‌کارگیری مجموعهٔ سه‌پیمانه‌ای $\{2^n-1, 2^n, 2^n+1\}$

زهرا حکیمی [1] و حمیدرضا احمدی‌فر [2*]
[1و2] دانشکدهٔ فنی، دانشگاه گیلان، رشت، ایران

چکیده

سیستم عددی مانده‌ای[1] به‌دلیل انجام عملیّات جمع، تفریق و ضرب در کانال‌های موازی باعث بهبود سرعت محاسبات می‌شود. برای استفاده از این سیستم به انجام عملیّات تبدیل از دودویی به مانده‌ای و مانده‌ای به دودویی نیاز است. وجود سربار محاسبات تبدیل می‌تواند باعث کاهش کارایی در به‌کارگیری این سیستم شود، مگراین‌که تعداد عملیّات مانده‌ای متوالی به‌ق-ری زیاد باشد که زمان سربار تبدیلات را پوشش دهد. در این مقاله با بررسی مجموعه سه‌پیمانه‌ای $\{2^n-1, 2^n, 2^n+1\}$ مشخص شد که به‌ازای چه تعداد عملیّات متوالی جمع یا ضرب، استفاده از عملیّات مانده‌ای منجر به سرعت بیشتر می‌شود. نتایج نشان‌می‌دهند که در صورت استفاده از جمع‌کننده با انتشاررقم نقلی[2]، در پیمانه‌های با عرض بیشتر از هشت بیت ($n \geq 8$) اگر تعداد عملیّات متوالی دست‌کم چهار باشد، باعث تسریع در محاسبات می‌شود. به همین ترتیب، در عمل ضرب و جمع‌کنندهٔ پیشوندی تعداد توالی به دو کاهش می‌یابد.

واژگان کلیدی: سیستم عددی مانده‌ای، عملیّات مانده‌ای، جمع‌کننده با انتشار رقم نقلی، جمع‌کنندهٔ پیشوندی موازی، ضرب‌کنندهٔ موازی بلوکی.

Set the Domain for Using 3-moduli Set $\{2^n-1, 2^n, 2^n+1\}$

Zahra Hakimi[1], HamidReza Ahmadifar[2*]
Computer Engineering Department, Faculty of Engineering,
University of Guilan, Rasht, Iran[1, 2]

Abstract

In special purpose circuits, the amount of energy consumed and the speed of operation are the main challenges. There are wide researches and methods to improve the performance of these types of circuits. One of these methods is to use a Residue Number System (RNS). In the RNS, there are a number of modules (channels) as a set to represent the number and perform parallel arithmetic operations. The most famous set is the 3-modlui set $\{2^n-1, 2^n, 2^n+1\}$. The form of modules to the power of 2 makes it easier to perform binary computational operations. To use this system, you need to perform conversion operations from binary to residue (forward conversion) and residue to binary (reverse conversion). The greater the number of modules (channels) in the set, the higher the degree of parallelism of computational operations. In contrast, more complex forward and reverse conversion circuits are required. The overhead of conversion computing can reduce the efficiency of using this system, unless the number of consecutive operations is large enough to cover the conversion overhead time.

In this paper, based on 3-moduli set $\{2^n-1, 2^n, 2^n+1\}$ evaluation, it was determined that for how many consecutive addition or multiplication operations, the use of RNS operations leads to greater

[1]Residue Number System [2]Ripple Adder

* Corresponding author
*نویسندهٔ عهده‌دار مکاتبات

• تاریخ ارسال مقاله: ۱۳۹۹/۱۲/۱۳ • تاریخ پذیرش: ۱۴۰۱/۱۲/۳ • تاریخ انتشار: ۱۴۰۲/۵/۲۰ • نوع مطالعه: پژوهشی

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-10]

[DOI: 10.61186/jsdp.20.1.123]

speed. In this paper, we evaluate the carry propagation adder as the most popular adder and parallel prefix adder as the high-speed adder. Also, the parallel block multiplier circuit was used to evaluate the multiplication operations. First, modular adder/multiplier, binary adder/multiplier, and forward/reverse conversion circuits were implemented and synthesized. We used Synopsys Design Compiler, K-2015.06 version and 45nm technology to evaluate the circuits. The results show that if the carry propagation adder is used, in modules with a width of more than 8 bits (n≥8), if the number of consecutive operations is at least 4, it will speed up the calculations. Likewise, in the multiplication operation and parallel prefix addition, the number of sequences is reduced to two.

Keywords: Residue Number System, Residue Operations, Carry Ripple Adder, Parallel Prefix Adder, Parallel Multiplier.

۱- مقدمه

سیستم عددی مانده‌ای یکی از سیستم‌های با قابلیّت انجام عملیّات جمع بدون انتشار رقم نقلی[۱] بین‌پیمانه‌ای است. عملیّات حسابی روی دو عدد بزرگ به همان عملیّات با اعداد کوچکتر تبدیل می‌شود که به‌طور موازی و در چند پیمانه انجام می‌شوند. در استفاده از این سیستم، نخست، باید تبدیل مستقیم انجام شود؛ یعنی عملیّات تبدیل اعداد از دودویی به مانده‌ای. پس از انجام عملیّات، برای نمایش نتایج نیاز به تبدل معکوس است که در این مرحله عدد از نمایش مانده‌ای به دودویی بازگردانده می‌شود[2].

انتشار بیت نقلی مهم‌ترین دلیل محدودیت سرعت در عملیّات حسابی است. امروزه سهم بزرگی از محاسبات در پردازنده‌های نهفته قراردارد، مانند دستگاه‌های تلفن همراه که برای این‌گونه سیستم‌ها سرعت بالا و مصرف توان کم از جمله مسائل بحرانی محسوب می‌شوند و فقدان انتشار بیت نقلی به تسهیل محاسبات در این شرایط کمک بسیار زیادی کرده‌است [3, 5, 6, 17].

کاربرد سیستم عددی مانده‌ای در مواردی است که نیاز به عملیّات تقسیم نداشته باشد. زیرا در این سیستم، روش خاصی برای انجام تقسیم وجود ندارد [2]. این دسته از کاربردها کم نیستند و نمونه‌هایی مانند پردازش سیگنال دیجیتال، امنیّت کامپیوتری (رمزنگاری)، سیستم‌های تحمل‌پذیر خطا و پردازش تصویر وجود دارد. در این روش، می‌توان سرعت محاسبات را به نحو چشمگیری افزایش داد [3, 5, 6, 17]. اگرچه جمع‌کننده و ضرب‌کننده‌های سریعی وجود دارند که می‌توانند برای افزایش کارایی در مدارات محاسباتی به کارگرفته شوند، اما مزیّت مهم سیستم عددی مانده‌ای را که عدم انتشار رقم نقلی بین پیمانه‌ای است، ندارند. از طرفی، در جمع‌کننده‌های سریعی مانند جمع‌کننده‌های پیشوندی موازی[۲] که دارای تأخیر لگاریتمی هستند، مدارهای معادل در سیستم عددی مانده‌ای طراحی شده‌اند که قابل رقابت با نمونه‌های دودویی خود هستند ولی در هر دو مدل فضای اشغال‌شده و توان مصرفی به‌شدت افزایش می‌یابد [14-16]. بنابراین، در هر دو سیستم مدارات جمع و ضرب سریع وجود دارند. از طرفی دیگر، به‌کارگیری سیستم عددی مانده‌ای مستلزم انجام عملیّات تبدیل مستقیم و معکوس است. بدیهی است که اگر عملیّات جمع یا ضرب از توالی کافی برخوردار نباشد، استفاده از این سیستم به‌صرفه نیست، چراکه سربار عملیّات تبدل مستقیم و معکوس غیرقابل چشم‌پوشی است. به‌ویژه این که افزایش تعداد پیمانه‌ها به بیش از سه، یا تغییر پیمانه‌ها باعث افزایش پیچیدگی عملیّات تبدیل معکوس می‌شود [8, 11]. باتوجّه‌به کاربردهای وسیع این سیستم و توجه به این نکته که کاربرد این روش بیشتر در سیستم‌های خاص‌منظوره است، اگر در یک کاربرد خاص بخواهیم از سیستم عددی مانده-ای استفاده کنیم، نخست باید بررسی شود که آیا انجام محاسبات در این سیستم به همراه تبدیل مستقیم و معکوس مقرون‌به‌صرفه است یا خیر. این مسئله را می‌توان با فرض استفاده از هر نوع مدار جمع و ضرب بررسی کرد. بنابراین، در این مقاله سعی شده تا به سؤال بالا پاسخ داده شود. ذکر این نکته ضروری است که در این مقاله هدف پیداکردن تعداد توالی عملیّات جمع یا ضرب است، به گونه‌ای که استفاده از سیستم عددی مانده‌ای در مقابل سیستم دودویی به صرفه باشد. در [1] این بررسی فقط برای جمع‌کنندهٔ با انتشار رقم نقلی و ضرب‌کننده موازی صورت گرفت. در این مقاله علاوه‌بر آن، جمع‌کنندهٔ پیشوندی موازی Kogge-Stone و ضرب‌کنندهٔ موازی با پیاده‌سازی بلوکی نیز در نظر گرفته و مدارهای تبدیل مستقیم و معکوس با تأخیر کمتری انتخاب شده‌اند، به گونه‌ای که تأخیر مجموع مدارات تبدیل و عملیّات مانده‌ای کمینه باشد.

از مهم‌ترین مجموعهٔ پیمانه‌ها که در کاربردهای مختلف استفاده شده است (و مجموعه پیمانه‌ای موردنظر در این مقاله نیز هست)، مجموعه سه‌پیمانه‌ای معروف

[۱] Carry free

[۲] Parallel Prefix Adder

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-10]

[DOI: 10.61186/jsdp.20.1.123]

$\{2^n-1, 2^n, 2^n+1\}$ است [2]. از جمله مزایای این مجموعه، توازن پیمانه‌های انتخابی است. منظور از متوازن بودن، نزدیک بودن زمان انجام عملیّات در پیمانه‌های مختلف است که باعث بهترشدن کارایی محاسبات می‌شود و وجود پیمانه‌هایی به فرم $2^n \pm 1$ باعث سادگی محاسبات مانده‌ای و تبدیل معکوس می‌شود.

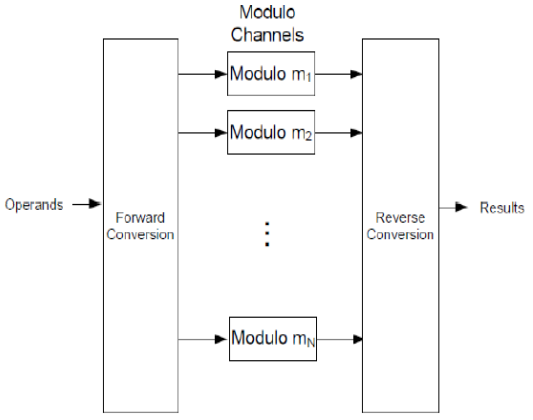
این مقاله شامل شش بخش است که در بخش دوم به معرفی سیستم عددی مانده‌ای پرداخته شده، در بخش سوم نحوهٔ انجام عملیّات مانده‌ای، در بخش چهارم به تعیین مرزبندی استفاده از سیستم مانده‌ای، بخش پنجم شبیه‌سازی و سنتز مدارها و در بخش ششم نتیجه‌گیری مقاله آورده شده است.

۲- تعریف سیستم عددی مانده‌ای

در سیستم عددی مانده‌ای، عدد X با مجموعه‌ای از باقی مانده‌ها نشان داده می‌شود که در kپیمانهٔ دوبه‌دو نسبت به هم، اول قرار دارند. اگر پیمانه‌ها با m_i نشان داده‌شود، دراین‌صورت، $x_i = |X|_{m_i}$ که باقی‌ماندهٔ عدد X است، متناسب با پیمانه m_i با x_i نشان‌داده‌شده است [2, 7].

تعداد اعداد قابل‌نمایش در یک مجموعه پیمانه‌ای که پیمانه‌ها نسبت به هم اول هستند، برابر است با حاصل‌ضرب پیمانه‌ها، که در اصطلاح DR [1] نامیده می‌شود [2, 7]. در مجموعه پیمانه‌ای موردنظر داریم $DR = (2^{2n}-1)2^n$، بنابراین، بازهٔ نمایش اعداد در این مجموعه حداکثر $3n$ بیتی است.

این سیستم از سه قسمت تبدیل مستقیم، عملیّات مانده‌ای و تبدیل معکوس، مطابق شکل (۱) تشکیل شده است، که در ادامه، عملیّات هر بخش شرح داده می‌شود.



(شکل-۱): اجزای سیستم مانده‌ای [12]

(Figure-1): Modular system components [12]

[1] Dynamic Range

۲-۱- تبدیل مستقیم

در تبدیل مستقیم، عمل نمایش عدد از دودویی به مانده‌ای انجام می‌شود. سخت‌افزار پیاده‌سازی مبدل مستقیم می‌تواند بر اساس جدول‌های جستجو[2]، مدارات منطقی ترکیبی، یا ترکیبی از هر دو باشد [2, 7].

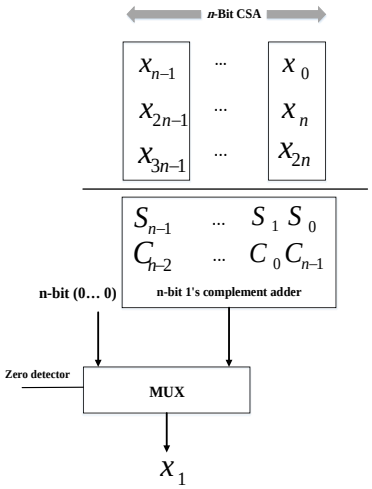
از مجموعهٔ سه‌پیمانه‌ای معروف به‌عنوان مجموعهٔ پیمانه‌ای کم‌هزینه یاد می‌شود، زیرا مبدل مستقیم و معکوس در این مجموعه نیاز به عملیّات پیچیده‌ای ندارد [12]. در این مجموعه به‌دلیل‌این‌که $0 \le DR \le (2^{2n}-1)2^n$ است، بنابراین، بازهٔ نمایش عدد X حداکثر $3n$ بیتی است (رابطه۱).

$$X = |x_{3n-1} \dots x_{n-1} x_{n-2} \dots x_0|_{DR} \qquad (۱)$$

در پیمانهٔ 2^n، برای تبدیل مستقیم تنها کافی است که همهٔ بیت‌هایی که ارزش آن‌ها بیشتر یا مساوی 2^n است را حذف کنیم تا عدد X در پیمانه نمایش داده شود. اگر باقی‌مانده متناظر با این پیمانه را X_2 بنامیم، داریم [10]:

$$X_2 = |X|_{2^n} = x_{n-1} x_{n-2} \dots x_0 \qquad (۲)$$

برای محاسبهٔ باقی‌مانده در پیمانهٔ 2^n-1، مطابق شکل۲، نخست، عدد به بلوک‌های n بیتی تقسیم می‌شود؛ سپس، به‌کمک مدارهای CSA[3] عمق آن به دو سطح کاهش یافته و در انتها با یک جمع‌کنندهٔ مکمل یک، حاصل در پیمانهٔ یادشده به دست می‌آید. بیت‌های s_i و c_i خروجی‌های مدارهای CSA هستند. قابل ذکر است که در این پیمانه به‌دلیل وجود جمع‌کنندهٔ مکمل یک ممکن است دو نمایش برای صفر حاصل شود؛ یکی مقدار تمام صفر و دیگری مقدار تمام یک (که دراین پیمانه معادل صفر است) که برای اصلاح آن از یک مالتی پلکسر استفاده شده است.



(شکل-۲): نمودار جعبه‌ای مبدل مستقیم در پیمانهٔ 2^n-1

(Figure-2): Modulo-(2^n-1) forward conversion diagram

[2] Look-Up Tables

[3] Carry Save Adder

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-10]

[DOI: 10.61186/jsdp.20.1.123]

در پیمانهٔ $2^n + 1$، مطابق شکل (۳) نخست، عدد به بلوک‌های n بیتی تقسیم می‌شود؛ با این تفاوت که بیت‌های x_n تا x_{2n-1} در بلوک دوم به‌صورت منفی ظاهر می‌شوند، زیرا $|2^n x_n|_{2^n+1} = -x_n$. به همین دلیل پیش‌از کاهش عمق به دو، نخست، باید بیت‌های مذکور به مقادیر مثبت تبدیل شوند. این کار به کمک رابطهٔ $-a = \bar{a} - 1$ انجام شده است. پس از عملیّات کاهش عمق به دو به کمک مدار جمع‌کنندهٔ n بیتی، می توان حاصل نهایی را به دست آورد. اما وجود مقادیر 1- در مرحلهٔ تبدیل بیت‌های منفی به مثبت، باعث می‌شود که حاصل-جمع نیاز به اصلاح داشته باشد. طبق جدول 1 نحوهٔ اضافه شدن مقادیر منفی به حاصل را نشان می‌دهد. البته مقدار اضافه‌شده بسته به مقدار رقم‌های نقلی تولیدشده در مرحلهٔ کاهش عمق (c_{n-1}) و جمع‌کنندهٔ n بیتی اول (c'_n)، یکی از دو مقدار یک، یا دو را دارد. دلیل مثبت بودن مقدار اضافه‌شده، جایگزینی مجموع مقادیر منفی با معادل مثبت در پیمانه‌ی $2^n + 1$ است. اگر هیچ‌کدام از مراحل کاهش عمق و جمع، رقم نقلی نداشته باشند دراین صورت مجموع مقادیر منفی برابر است با $-(2^n - 1)$ ، و اگر در یکی از این دو مرحله رقم نقلی داشته باشیم، برابر است با $-(2^n - 1) - 1$، و اگر هردو رقم نقلی داشته باشند، برابر است با $-(2^n - 1) - 2$. مطابق رابطهٔ (3) می‌توان این مقادیر منفی را به معادل مثبت آن در این پیمانه تبدیل کرد:

(۳)

$$|-(2^n-1)|_{2^n+1} = |2^n + 1 - (2^n - 1)|_{2^n+1} = 2$$

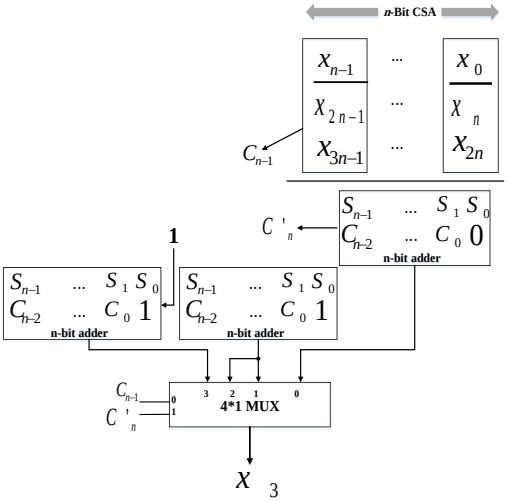
$$|-(2^n-1) - 1|_{2^n+1} = |2^n + 1 - (2^n - 1) - 1|_{2^n+1} = 1$$

$$|-(2^n-1) - 2|_{2^n+1} = |2^n + 1 - (2^n - 1) - 2|_{2^n+1} = 0$$

(جدول-۱): مقادیر عددی اضافه‌شده به حاصل بر مبنای رقم‌های نقلی
(Table-1): Added values based on carries out

c'_n	c_{n-1}	مقدار عددی مثبت
0	0	+2
0	1	+1
1	0	+1
1	1	+0

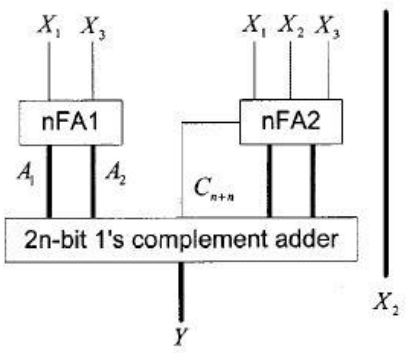
یکی از تفاوت‌های این مقاله با مدار ارائه‌شده در [1]، مدار تبدیل مستقیم در پیمانهٔ $2^n + 1$ است. این پیمانه بیشترین تأخیر محاسبات را در مقایسه با دو پیمانهٔ دیگر دارد، زیرا علاوه‌بر عملیّات کاهش عمق و جمع، نیاز است که مقدار ثابتی به حاصل اضافه شود. به همین دلیل به کمک مالتی‌پلکسر و جمع موازی مقادیر ثابت برای تشکیل حاصل نهایی، تأخیر مدار به کمینه رسیده است.



(شکل-۳): نمودار جعبه‌ای مبدل مستقیم در پیمانهٔ $2^n + 1$
rward conversion diagram (Figure-3): Modulo-$(2^n + 1)$ fo

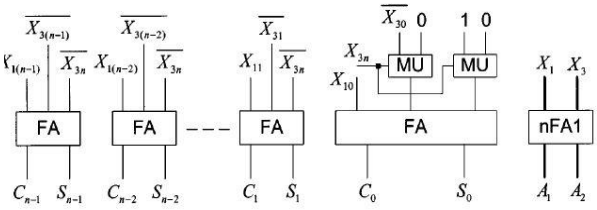
۲-۲-تبدیل معکوس

برای انجام تبدیل معکوس از روش ارائه‌شده در [12] مطابق شکل‌های(۴ تا ۶) استفاده شده‌است. در شکل 4، X_1باقی‌مانده در پیمانه $2^n - 1$، X_2باقی‌مانده در پیمانه 2^n و X_3باقی‌مانده در پیمانه $2^n + 1$ است. همان‌طورکه ذکر شد، X عددی $3n$ بیتی است؛ بنابراین، در شکل 4، n بیت کم‌ارزش‌تر به‌طور مستقیم از X_2گرفته شده و Y نیز $2n$ بیت پرارزش‌تر X است. شکل‌های ۵ و ۶ جزئیات پیاده‌سازی دو بخش چپ و راست از شکل ۴ را نشان می‌دهند [12].



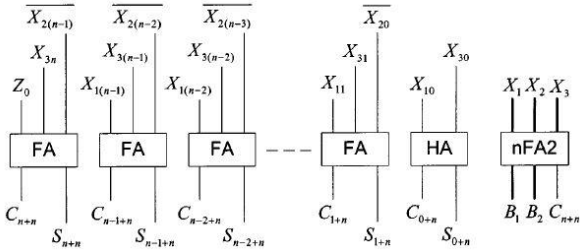
(شکل-۴): نمودار جعبه‌ای مبدل معکوس [12]
(Figure-4): Reverse conversion diagram [12]

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-10]
[DOI: 10.61186/jsdp.20.1.123]



(شکل-۵): محاسبه A_1 و A_2 در مبدل معکوس [12]

(Figure-5): A_1 and A_2 computations in reverse conversion [12]



(شکل-۶): محاسبه B_1 و B_2 در مبدل معکوس [12]

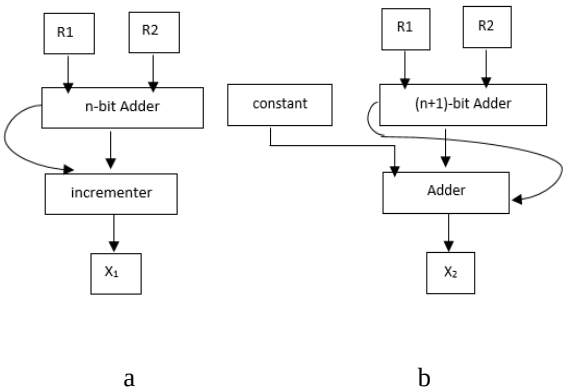
(Figure-6): B_1 and B_2 computations in reverse conversion [12]

۳-محاسبات مانده‌ای

در این بخش به بررسی نحوهٔ انجام عملیّات جمع و ضرب در سیستم عددی مانده‌ای پرداخته می‌شود. قابل ذکر است که عمل تفریق را می‌توان همانند سیستم عددی دودویی با کمک مکمّل دو به جمع تبدیل کرد.

۳-۱-جمع مانده‌ای

عمل جمع در پیمانه‌های این مجموعه از مراحلی مشابه تبدیل مستقیم برخوردار است. بلوک دیاگرام جمع به پیمانه‌های $2^n \pm 1$ در شکل (۷) نشان داده شده است. $R1$ و $R2$ باقی‌مانده‌های متناظر با دو عدد ورودی هستند [12].



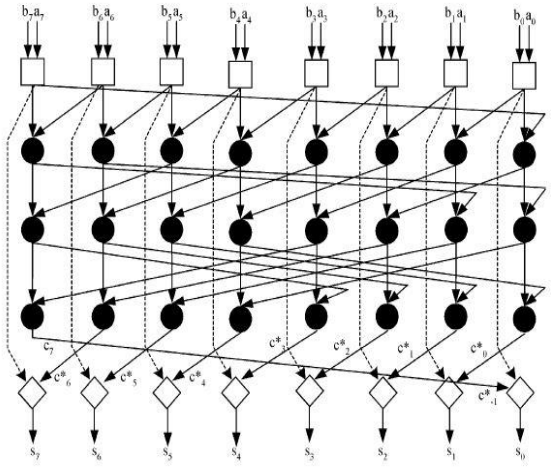
a b

(شکل-۷): نمودار جعبه‌ای جمع‌کننده مانده‌ای در پیمانه (a) 2^n-1 و (b) 2^n+1

(Figure-7): Modular adder diagram in modulo (a) 2^n-1 and (b) 2^n+1

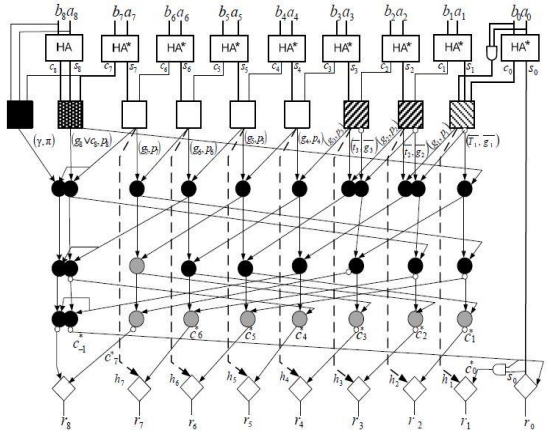
۳-۱-۱-جمع مانده‌ای پیاده‌سازی‌شده توسط جمع کننده پیشوندی موازی

همان‌گونه‌که در بخش مقدّمه اشاره شد، برای انجام جمع سریع در سیستم عددی دودویی مدارهای جمع‌کنندهٔ پیشوندی موازی وجود دارند، که با تأخیر لگاریتمی ($O(\log n)$) می‌توانند حاصل‌جمع را محاسبه کنند [2-7]. در سیستم عددی مانده‌ای نیز با تأخیری مشابه می‌توان جمع را انجام داد. برای محاسبهٔ حاصل‌جمع در پیمانه 2^n از روش معمول جمع‌کنندهٔ پیشوندی موازی Kogge-Stone استفاده شده است [7]. اما در پیمانه‌های 2^n-1 و 2^n+1 به روش‌های ارائه‌شده در [13-16] می‌توان اشاره کرد که در شکل‌های(۸ و ۹) نشان داده شده‌اند.



(شکل-۸): جمع‌کنندهٔ پیشوندی موازی در پیمانه 2^n-1 [14]

(Figure-8): Parallel prefix adder in modulo-2^n-1 [14]



(شکل-۹): جمع‌کنندهٔ پیشوندی موازی در پیمانه 2^n+1 [16]

(Figure-9): Parallel prefix adder in modulo-2^n+1 [16]

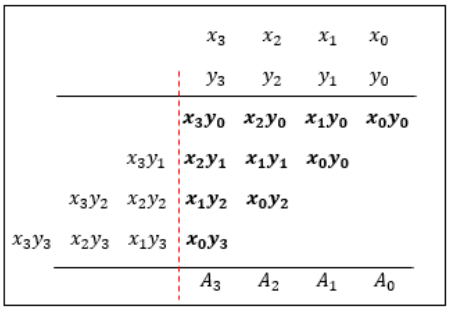
۳-۲-ضرب مانده‌ای

در محاسبهٔ باقی‌ماندهٔ حاصل‌ضرب دو عدد، نخست با تبدیل مستقیم، باقی‌ماندهٔ هر دو عدد محاسبه می‌شود.

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-10]

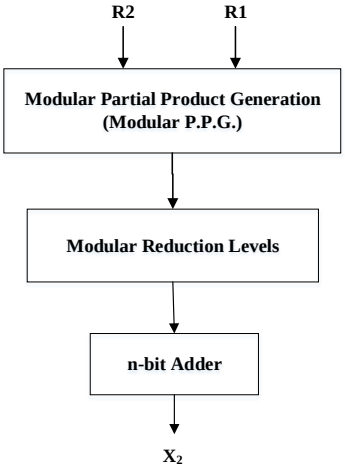
[DOI: 10.61186/jsdp.20.1.123]

سپس، برای محاسبه باقی‌مانده در پیمانه 2^n، تنها کافی است که حاصل‌ضرب‌های جزئی متناظر با دو باقی‌مانده، تنها برای n بیت محاسبه شود. سپس توسط مدارهای کاهنده‌ی عمق مانند CSA و جمع‌کننده، حاصل‌ضرب محاسبه شود. در شکل (۱۰) نحوهٔ قرارگیری حاصل‌ضرب‌های جزئی در پیمانه 2^n برای $n = 4$ نشان داده‌شده‌است. $(x_3 \dots x_0)$ و $(y_3 \dots y_0)$ به‌ترتیب، باقی‌ماندهٔ دو عدد X و Y به پیمانه 16 هستند و $(A_3 \dots A_0)$باقی‌ماندهٔ حاصل‌ضرب عدد X و Y در پیمانه ۱۶ است[9]. هم‌چنین، در شکل (۱۱) نمودار جعبه‌ای ضرب‌کننده در پیمانه 2^n نشان داده شده است.



(شکل-۱۰): نحوهٔ قرارگیری حاصل‌ضرب‌های جزئی در پیمانه ۱۶

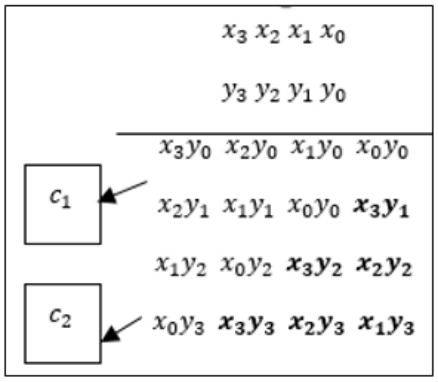
(Figure-10): Partial products bit map in modulo-16



(شکل-۱۱): نمودار جعبه‌ای ضرب‌کننده در پیمانه 2^n

(Figure-11): Multiplication diagram in modulo-2^n

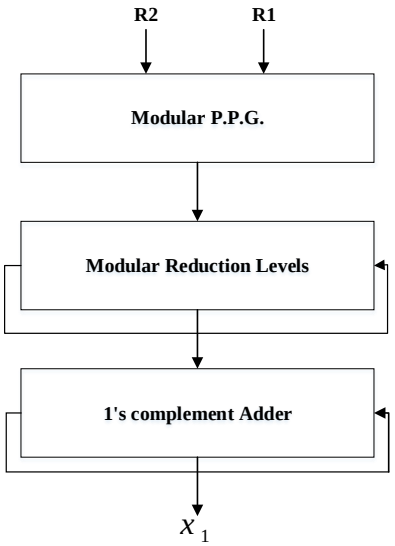
در پیمانه $2^n - 1$ نحوهٔ قرارگیری حاصل ضرب‌های جزئی مطابق شکل (۱۲) است (در پیمانه ۱۵). در این پیمانه مقادیر نقلی دوباره به چرخهٔ محاسبات باز می‌گردند. در شکل c_1 و c_2 رقم‌های نقلی حاصل از انجام عمل جمع برای کاهش عمق هستند. بلوک دیاگرام نحوهٔ انجام عمل ضرب در این پیمانه مشابه با پیمانه‌ی 2^n است، با این تفاوت که رقم‌های نقلی خروجی دوباره در محاسبات شرکت می‌کنند و هیچ‌کدام از بیت‌های مربوط به حاصل‌ضرب جزئی حذف نمی‌شوند.



(شکل-۱۲): نحوه قرارگیری حاصل‌ضرب‌های جزئی در پیمانه ۱۵

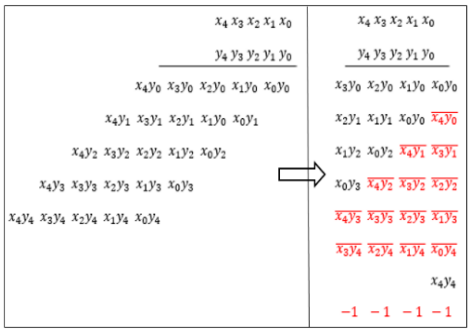
(Figure-12): Partial products bit map in modulo-15

در شکل (۱۴) نمونه‌ای از نحوهٔ قرارگیری حاصل ضرب‌های جزئی در پیمانه $2^n + 1$ (پیمانه ۱۷) نشان داده شده است. مانند حاصل‌جمع در این پیمانه، رقم‌های نقلی نیز به شکل منفی به چرخه محاسبات باز می‌گردند (بیت‌های به رنگ قرمز). در پایان با افزودن مقدار ثابت، حاصل ضرب نهایی به دست می‌آید. در شکل (۱۵) نمودار جعبه‌ای کلی ضرب‌کننده در پیمانه $2^n + 1$ نشان داده شده است.



(شکل-۱۳): نمودار جعبه‌ای ضرب‌کننده در پیمانه $2^n - 1$

(Figure-13): Multiplication diagram in modulo-$2^n - 1$

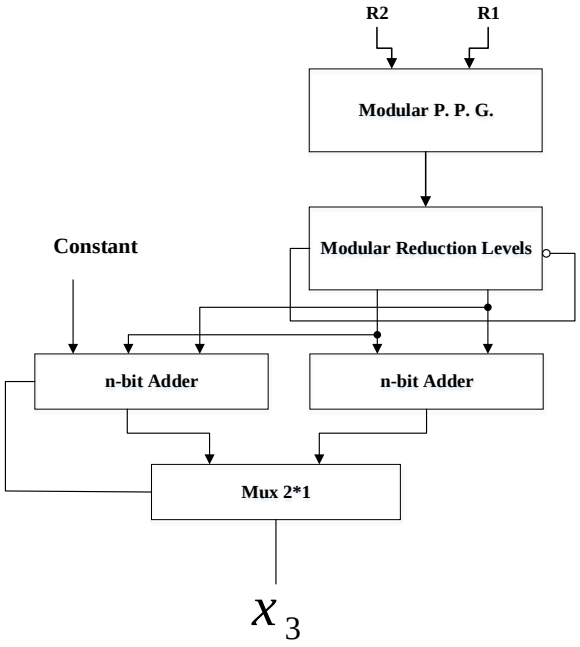


(شکل-۱۴): نحوهٔ قرارگیری حاصل‌ضرب‌های جزئی در پیمانه ۱۷

(Figure-14): Partial products bit map in modulo-17

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-10]

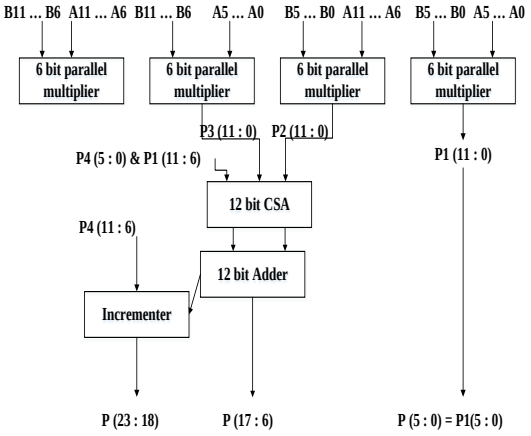
[DOI: 10.61186/jsdp.20.1.123]



(شکل-۱۵): بلوک دیاگرام ضرب‌کننده در پیمانه 2^n+1
(Figure-15): Multiplication diagram in modulo-2^n+1

۳-۲-۱-ضرب‌کننده موازی

در مقایسهٔ بین مدارهای ضرب‌کنندهٔ دودویی و مانده‌ای لازم است ساختاری برای پیاده‌سازی مدارهای ضرب‌کنندهٔ موازی دودویی در نظر گرفته شود. برای این کار در پیاده سازی ضرب‌کننده موازی، هر ضرب‌کننده را با چهار ضرب کنندهٔ قبلی خود تشکیل دادیم. این کار باعث سهولت پیاده‌سازی ضرب‌کننده با تعداد بیت‌های زیاد می‌شود. شکل (۱۶) بلوک دیاگرام مدار ضرب‌کنندهٔ دوازده‌بیتی را نشان می‌دهد که از جعبه‌های ضرب‌کنندهٔ شش‌بیتی تشکیل شده است.



(شکل-۱۶): نمودار جعبه‌ای ضرب‌کنندهٔ موازی ۱۲بیتی به کمک جعبه‌های شش‌بیتی
(Figure-16): 12-bit Parallel multiplication diagram based on 6-bit block

۴- تعیین دامنه برای به‌کارگیری مجموعهٔ سه‌پیمانه‌ای $\{2^n-1, 2^n, 2^n+1\}$

در این بخش در مدارهای طرّاحی‌شده برای مراحل مختلف انجام عملیّات حسابی مانده‌ای، یعنی تبدیل مستقیم، عملیّات مانده‌ای و تبدیل معکوس، تأخیر آن‌ها محاسبه شده تا بر اساس آن بتوان مقدار کمینهٔ توالی عملیّات را محاسبه کرد. در [1] تنها برای عملیّات جمع با انتشار رقم نقلی و ضرب‌کنندهٔ موازی مقایسه و تعیین کمینهٔ توالی انجام شد. در این بخش جمع‌کننده‌های پیشوندی موازی در هر دو حالت دودویی و مانده‌ای نیز در نظر گرفته شده‌اند. همچنین، مدار تبدل مستقیم در پیمانهٔ 2^n+1 در مقایسه با آنچه در [17] انتخاب شده بود، بهبود داده شد، که باعث کاهش تأخیر آن شود. در مدار ضرب‌کنندهٔ موازی نیز از ساختار بلوکی کمک گرفته شده که نمونه‌ای از آن در شکل (۱۶) نشان داده شده است.

بدیهی است که تنها انجام یک عمل جمع، یا ضرب به دلیل وجود مدارات تبدیل مستقیم و معکوس، مقرون به صرفه نیست.

در این مقاله برای مقایسهٔ عادلانه مدارات جمع و ضرب دودویی و پیمانه‌ای، ملاک مقایسه، برابر بودن تعداد بیت‌های نمایش‌دهنده مقدار خروجی است. بنابراین در جمع دودویی اگر حاصل‌جمع $3n+1$ بیتی باشد، در سیستم عددی مانده‌ای سه‌پیمانه‌ای، از دو جمع کننده n بیتی و یک جمع‌کنندهٔ $n+1$ بیتی استفاده شد تا خروجی $3n+1$بیتی باشد. در ضرب دودویی که پهنای خروجی دو برابر ورودی است، در سیستم مانده‌ای هر پیمانه $\lceil 2n/3 \rceil$بیتی خواهد بود.

رابطهٔ کلی تأخیر جمع و ضرب پیمانه‌ای برای C عمل متوالی جمع یا ضرب برابر است با تأخیر یک مدار تبدیل مستقیم به‌علاوهٔ تأخیر C عمل جمع یا ضرب به‌علاوهٔ تأخیر یک مدار تبدیل معکوس، مطابق رابطهٔ (۴). در اینجا Δ نشان‌دهندهٔ مقدار تأخیر است.

$$\Delta_{RNS} = \Delta_{\substack{Forward \\ Conversion}} + C \times \left(\Delta_{Add/Mul}\right) + \Delta_{\substack{Reverse \\ Conversion}} \quad (4)$$

اما تأخیر جمع و ضرب دودویی تنها وابسته به تأخیر مدار عمل مرتبط است، مطابق رابطهٔ (5):

$$\Delta_{Total(Add\,/\,Mul)} = C * \Delta_{Add/\,Mul} \quad (5)$$

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-10]
[DOI: 10.61186/jsdp.20.1.123]

۵- شبیه‌سازی و سنتز مدارها

جهت انجام آزمایش‌های و محاسبهٔ میزان تأخیر مدارات جمع و ضرب پس از نوشتن رمزهای VHDL و تست صحت عملکرد آن‌ها، کلیهٔ مدارات به‌کمک کامپایلر Synopsys Design Vision نسخهٔ K-2015.06 و تکنولوژی 45n سنتز شده‌اند. در جدول (۲) تأخیر مدارهای انجام عملیّات مانده‌ای شامل جمع با انتشار رقم نقلی، جمع پیشوندی موازی و ضرب‌کنندهٔ موازی نشان داده شده است. ستون نخست در این جدول نشان‌دهندهٔ عرض (پهنای) هریک از کانال‌ها (پیمانه‌ها) است که از ۴ الی ۱۲۸ بیت در نظرگرفته شده تا طیف کاملی از تغییرات دامنه را شامل شود.

(جدول-۲): تأخیر مدار جمع/ضرب‌کنندهٔ دودویی

(Table-2): Delay of binary Adder/Multiplier circuits

پهنای کانال (n)	جمع‌کننده با انتشار رقم نقلی مانده‌ای (ns)	جمع‌کننده پیشوندی موازی مانده‌ای (ns)	ضرب‌کننده مانده‌ای (ns)
4	0.59	0.57	1.35
8	1.07	0.57	1.98
16	2.14	0.71	3.05
32	4.26	0.83	4.96
64	8.50	0.90	8.25
128	16.99	1.06	14.09

در جدول (۳) تأخیر مدارات انجام عملیّات مانده‌ای نشان داده شده است. در مقایسه با جدول (۲) پهنای ورودی و خروجی محاسبات به گونه‌ای انتخاب شده که با پهنای خروجی واحد مانده‌ای هم‌اندازه باشد تا بتوان مقایسهٔ عادلانه‌ای انجام داد.

در جدول (۴) تأخیر مدارات طراحی‌شده در بخش پیش برای انجام عملیّات تبدیل مستقیم و معکوس نشان داده شده است. بدیهی است که پهنای انتخابی متناسب با پهنای عملیّات مانده‌ای است.

(جدول-۳): تأخیر مدار جمع/ضرب‌کننده مانده‌ای

(Table-3): Delay of modular Adder/Multiplier circuits

پهنای جمع/ضرب کننده‌دودویی (bit)	جمع‌کننده با انتشار رقم نقلی (ns)	جمع‌کننده پیشوندی موازی (ns)	ضرب‌کننده موازی(ns)
13	1.22	0.43	2.94
25	2.29	0.51	5.27
49	4.42	0.59	9.48
97	8.69	0.67	17.46
193	17.22	0.75	33.01
385	34.29	0.83	64.03

(جدول-۴): تأخیر مدار تبدیل مستقیم و معکوس

(Table-4): Delay of forward and reverse converters

n	تبدیل مستقیم (ns)	تبدیل معکوس (ns)
4	0.96	1.43
8	1.49	2.49
16	2.64	4.65
32	4.80	8.93
64	9.10	17.46
128	17.64	34.42

پس از جای‌گذاری تأخیر مدارات محاسبه‌شده در روابط (4) و (5)، مقدار حداقل C مطلوب جهت به‌کارگیری سیستم عددی مانده‌ای، مطابق جدول (۵) حاصل شده است. در ستون "جمع مانده‌ای" انجام متوالی عمل جمع سیزده‌بیتی با انتشار رقم نقلی با جمع مانده‌ای در مجموعهٔ سه پیمانه‌ای با پهنای چهار بیت، مقایسه شده است. در این شرایط خروجی مجموعهٔ سه پیمانه‌ای همانند جمع با انتشار رقم نقلی، سیزده‌بیتی خواهد بود. عدد پنج در این ستون نشان‌دهندهٔ این است که اگر پنج جمع متوالی (و یا بیشتر) وجود داشته باشد، هزینهٔ پیاده‌سازی آن به‌صورت سیستم عددی مانده‌ای مقرون‌به‌صرفه است. به‌ازای پهنای بیشتر جمع‌کننده، به تعداد توالی کمتری (یعنی ۴) نیاز است.

(جدول-۵): مقدار حداقل C در مقایسه با ضرب‌کنندهٔ موازی، جمع‌کننده با انتشار رقم نقلی و جمع‌کنندهٔ پیشوندی موازی

(Table-5): Minimum value of C in comparison with the carry propagation adder and parallel prefix adder

پهنای جمع-ضرب‌کننده دودویی (bit)	پهنای کانال (n)	مقدار C		
		جمع با انتشار رقم نقلی مانده‌ای	جمع پیشوندی موازی مانده‌ای	ضرب مانده‌ای
13	**4**	5	4	2
25	8	4	3	2
49	16	4	3	2
97	32	4	2	2
193	64	4	2	2
385	128	4	2	2

به همین ترتیب، درصورت استفاده از جمع‌کنندهٔ پیشوندی موازی، تعداد توالی به عدد چهار و به‌تدریج به دو کاهش می‌یابد. در مدار ضرب‌کنندهٔ مانده‌ای به‌دلیل افزایش پیچیدگی مدار، مقرون‌به‌صرفه بودن استفاده از سیستم عددی مانده‌ای زودتر به وقوع می‌پیوندد و از همان آغاز (برای $n = 4$) به عدد دو می‌رسد. آزمایش‌ها نشان می‌دهند که افزایش بیشتر تعداد بیت‌ها تأثیری در نتایج به‌دست‌آمده ندارند.

۶- نتیجه‌گیری

سیستم عددی مانده‌ای قابلیّت انجام محاسبات جمع، تفریق و ضرب را به‌صورت موازی و بدون انتشار رقم نقلی بین‌پیمانه‌ای دارد. به دلیل وجود مدارهای تبدیل مستقیم و معکوس، به‌کارگیری این سیستم تنها برای یک عمل جمع و یا ضرب به‌صرفه نیست؛ امّا این سیستم در کاربردهایی که در آنها عمل جمع یا ضرب، به‌صورت

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-10]

[DOI: 10.61186/jsdp.20.1.123]

متوالی تکرار می‌شوند، تأخیر بسیار کمتری نسبت به جمع یا ضرب‌کننده غیر مانده‌ای دارد. هدف این مقاله یافتن تعداد توالی عملیّات جمع و ضرب برای شروع به‌کارگیری سیستم عددی مانده‌ای بود. مجموعه پیمانه‌ای استفاده شده $\{2^n-1, 2^n, 2^n+1\}$ است. برای محاسبهٔ دقیق تعداد توالی برای استفاده از سیستم عددی مانده‌ای، مدارهای تبدیل مستقیم و معکوس طراحی شدند و بسته به نوع عمل و نحوهٔ انجام آن، بررسی بر اساس پیاده‌سازی و سنتز نتایج انجام شد. در این مقاله سه الگوی محاسباتی در نظرگرفته شد که عبارتند از عمل جمع با انتشار رقم نقلی، جمع پیشوندی و ضرب‌کنندهٔ موازی بلوکی. پس از شبیه‌سازی و سنتز مدارهای مرتبط مشخص شد که در عمل جمع با انتشار رقم نقلی با پهنای سیزده بیت حداقل پنج عمل متوالی ($C=5$)، در جمع پیشوندی موازی دست‌کم چهار ($C=4$) و در ضرب‌کنندهٔ موازی حداقل ۲ ($C=2$) عمل موردنیاز است تا در مقایسه با پیاده‌سازی آن‌ها به‌صورت دودویی، مقرون‌به‌صرفه باشد. با افزایش پهنای محاسبات، تعداد توالی کاهش می‌یابد و در نهایت، در جمع‌کنندهٔ با انتشار رقم نقلی به چهار و در جمع‌کنندهٔ پیشوندی و ضرب‌کنندهٔ موازی به دو می‌رسد. بر اساس این نتایج طرّاحان می‌توانند شرایط استفاده از سیستم عددی مانده‌ای را مشاهده کرده و بر اساس فرضیات تعریف‌شده، انتخاب درستی را انجام دهند.

۷- مراجع 7-Refrence

[۱] ز. حکیمی، ح. احمدی‌فر، مرزبندی برای به‌کارگیری سیستم‌های عددی مانده‌ای، چهارمین کنفرانس تکنولوژی در مهندسی برق و کامپیوتر، پژوهشگاه ارتباطات و فناوری اطلاعات، تهران، ایران، خرداد ۱۳۹۸.

[1] Hakimi, Z., Ahmadifar, H.,” Delimitation for Using Residue Number Systems”, 4th Conference on Electrical and Computer Engineering Technology, E-TECH2019, ITRC, Tehran, 1398.

[2] Omondi, A., Premkumar, B., “Residue Number Systems – Theory and Implementation”, Imperial College Press (ICP), 2007.

[3] Belghadr, A., Jaberipur, G., “FIR Filter Realization via Deferred End-Around Carry Modular Addition”, IEEE TCAS I, Vol. 65, pp. 2878 – 2888, 2018.

[4] Xiao, L., Xiang-Gen, X., “Robust Polynomial Reconstruction via Chinese Remainder Theorem in the Presence of Small Degree Residue Errors”, IEEE TCAS II: Vol. 65 , Issue: 11, pp. 1778 – 1782, 2018.

[5] Sousa, L., Antao, S., Martins, P., “Combining Residue Arithmetic to Design Efficient Cryptographic Circuits and Systems”, IEEE Circuits and Systems Magazine, Vol.16, Issue:4, pp. 6-32, 2016.

[6] Tay, T., Chang, C.-H., “A non-iterative multiple residue digit error detection and correction algorithm in RRNS”, IEEE Transactions on Computers, Vol. 65 , Issue: 2, pp. 396 – 408, 2016.

[7] Koren I, “Computer Arithmetic Algorithms”, 2d Edition, A.K. Peters Ltd, 2002.

[8] Hiasat, A, ”An Efficient Reverse Converter for the Three-Moduli Set $\{2^{n+1}-1, 2^n, 2^n-1\}$”, IEEE Transactions on Circuits and Systems II, Vol. 64, Issue: 8 pp. 962 – 966, 2017.

[9] Ahmadifar, A., and G. Jaberipur, “Improved modulo-$2^q \pm 3$ multipliers,” in Proc. Of the 17th CSI International Symposium on Computer Architecture and Digital Systems (CADS2013), Tehran, Iran, pp. 31-35.

[10] Patronik, P., Piestrak, S.J., “Hardware/Software Approach to Designing Low-Power RNS-Enhanced Arithmetic Units,” IEEE TCAS I, Vol. 64, 2017.

[11] Ahmadifar, H., Jaberipur, G., “A New Residue Number System with 5-Moduli Set: $\{2^{2q}, 2^q \pm 3, 2^q \pm 1\}$”, The Computer Journal, Vol. 58 , Issue: 7, pp.1548 – 1565, 2015.

[12] Wang Y, Song X, Aboulhamid M, Shen H, “ Adder Based Residue to Binary Number Converters for $\{2^n-1, 2^n, 2^n+1\}$”, IEEE Transactions on Signal Processing, Vol. 50, No. 7, July 2002.

[13] Jaberipur, G., B. Parhami, and S. Nejati, “On Building General Modular Adders from Standard Binary Arithmetic Components,” Proc. 45th Asilomar Conf. Signals, Systems, and Computers, 6-9 Nov., Pacific Grove, CA, USA, pp. 154-159, 2011.

[14] Kalamatianos L, Nikolos D, Efstathiou C, T. Vergos H, Kalamatianos J, “High-Speed Parallel-Prefix Modulo 2^n-1 Adders,” IEEE Trans. Computers, Vol. 49, No. 7, special issue on computer arithmetic, pp. 673-680, July 2000.

[15] Efstathiou C., H. T. Vergos, and D. Nikolos, “Fast Parallel-Prefix 2^n+1 Adder”, IEEE Trans. on Computers, Vol. 53, No. 9, pp. 1211–1216, September 2004.

[16] Jaberipur G, Alavi H, “Comment on “Fast Parallel Prefix Modulo 2^n+1 Adder”, IEEE Trans. on Computers, Vol. 64, No. 1, pp. 293-294, January 2015.

[17] Cardirilli, G.C., Nunzio, L.D., Fazzolari, R., Nannarelli, A., et al,” Design Space Exploration Based Methodology for Residue Number System Digital Filters Implementation”, IEEE Trans. On Emerging Topics in Computing, Early Access,25 May, 2020.

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-10]

[DOI: 10.61186/jsdp.20.1.123]



زهرا حکیمی، مدرک کارشناسی و کارشناسی‌ارشد خود را به‌ترتیب در سال‌های ۱۳۹۵ و ۱۳۹۸ از دانشگاه گیلان و در رشته مهندسی کامپیوتر دریافت کرده است. زمینه‌های پژوهشی موردعلاقهٔ ایشان حساب کامپیوتری، طرّاحی سیستم‌های نهفته و تحمل‌پذیر اشکال است.

نشانی رایانامهٔ ایشان عبارت است از:

zahra.hakimi.h@gmail.com



حمیدرضا احمدی‌فر تحصیلات خود را در مقطع کارشناسی و دکتری در دانشگاه شهید بهشتی تهران و در مقطع کارشناسی ارشد در دانشگاه صنعتی امیرکبیر در رشته مهندسی کامپیوتر با تخصص معماری سیستم‌های کامپیوتری به پایان رسانده است. در حال حاضر عضو هیئت علمی گروه مهندسی کامپیوتر دانشگاه گیلان است. سیستم اعداد مانده‌ای، حساب کامپیوتری و رایانش ابری زمینه‌های پژوهشی مورد علاقهٔ ایشان است.

نشانی رایانامهٔ ایشان عبارت است از:

ahmadifar@guilan.ac.ir

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-10]

[DOI: 10.61186/jsdp.20.1.123]