

# طراحی ترکیبی مبدل‌های مستقیم و معکوس: شیوه‌ای نو برای کاهش پیچیدگی سخت‌افزاری سیستم اعداد مانده‌ای

آزاده السادات عمرانی زرنندی<sup>۱</sup>، استادیار؛ امیر صباغ ملاحسینی<sup>۲</sup>، استادیار

۱- گروه مهندسی کامپیوتر- دانشگاه شهید باهنر کرمان- کرمان- ایران - a.emrani@uk.ac.ir

۲- گروه مهندسی کامپیوتر- دانشگاه آزاد اسلامی واحد کرمان- کرمان- ایران - amir@iauk.ac.ir

**چکیده:** ساختار سخت‌افزاری سیستم اعداد مانده‌ای متشکل از چندین واحد شامل مبدل مستقیم، واحدهای محاسباتی مجزا برای انجام جمع و ضرب پیمانه‌ای و مبدل معکوس است. مبدل‌های مستقیم و معکوس که برای ارتباط سیستم اعداد مانده‌ای با دیگر مدارهای دیجیتال نیاز است، در واقع سربار سیستم می‌باشند زیرا باعث افزایش سطح تراشه و توان مصرفی می‌شوند. این مقاله، برای اولین بار، یک مبدل ترکیبی برای سیستم اعداد مانده‌ای پیشنهاد می‌دهد که مبدل‌های مستقیم و معکوس را از طریق اشتراک سخت‌افزار، یک‌پارچه می‌کند. برای رسیدن به این هدف، از الگوریتم تبدیل درهم مبنا استفاده شده است تا روابط حسابی تبدیل معکوس در یک قالب مشابه با روابط حسابی تبدیل مستقیم قرار گیرند. سپس با استفاده از مالتی پلکسرها و تنظیم ورودی‌ها، از سخت‌افزار مبدل معکوس، برای انجام تبدیل مستقیم استفاده شده است. نتایج حاصل از پیاده‌سازی VLSI مبدل ترکیبی پیشنهادی مبتنی بر تکنولوژی TSMC-65nm، برای مجموعه پیمانه  $\{2^n-1, 2^{2^n}, 2^{2^n+1}-1\}$ ، نشانگر کاهش حداکثر ۱۹ درصدی سطح تراشه در مقایسه با مجموع مبدل‌های مستقیم و معکوس است. این در حالی است که تأخیر مبدل ترکیبی پیشنهادی حداکثر ۱۰ درصد از تأخیر مبدل معکوس مجزا بیشتر شده است.

**واژه‌های کلیدی:** حساب کامپیوتری، مدارهای حسابی دیجیتال، سیستم اعداد مانده‌ای، مبدل مستقیم، مبدل معکوس، جمع‌کننده پیمانه‌ای.

## Hybrid Design of Forward and Reverse Converters: A New Approach to Reduce Hardware Complexity of Residue Number System

Azadeh Alsadat Emrani Zarandi<sup>1</sup>, Assistant professor; Amir Sabbagh Molahosseini<sup>2</sup>, Assistant professor

1- Department of Computer Engineering, Shahid Bahonar University of Kerman, Kerman, Iran, Email: a.emrani@uk.ac.ir

2- Department of Computer Engineering, Islamic Azad University Kerman Branch, Kerman, Iran, Email: amir@iauk.ac.ir

**Abstract:** The Residue Number System (RNS) hardware structure consists of several components, including forward converter, separate arithmetic units for performing modular addition and multiplication, and reverse converter. Forward and reverse converters, essential in any RNS to interface with other digital circuits, represent overhead, resulting in larger chip-area and power-consumption. This work, for the first time, proposes a hybrid converter for RNS, which unifies forward and reverse converters by re-using hardware. To achieve this aim, the mixed-radix conversion (MRC) algorithm has been used for putting up the reverse conversion formulas in a similar format to forward conversion formulas. The VLSI implementation results of the proposed hybrid converter based on TSMC-65nm technology for the moduli set  $\{2^n-1, 2^{2^n}, 2^{2^n+1}-1\}$  show a reduction up to 19% of the required area in comparison to the total area of the forward and reverse converters. However, the delay of the proposed hybrid converter is just 10% higher than individual reverse converter delay.

**Keywords:** Computer arithmetic, digital arithmetic circuits, residue number system, forward converter, reverse converter, modular adder.

تاریخ ارسال مقاله: ۱۳۹۷/۱۰/۰۶

تاریخ اصلاح مقاله: ۱۳۹۸/۰۲/۰۹

تاریخ پذیرش مقاله: ۱۳۹۸/۰۳/۲۳

نام نویسنده مسئول: آزاده السادات عمرانی زرنندی

نشانی نویسنده مسئول: گروه مهندسی کامپیوتر- دانشگاه شهید باهنر کرمان- کرمان- ایران.

## ۱- مقدمه

سیستم‌های پردازش سیگنال دیجیتال نهفته امروزی [۱] نیاز به ساختارهای محاسباتی با انرژی بسیار کم دارند تا بتوان از آن‌ها در پیاده‌سازی سخت‌افزاری کاربردهای مدرن مانند یادگیری عمیق [۲] و رمزنگاری [۳] استفاده کرد. به عبارتی پیاده‌سازی سخت‌افزاری کاربردهایی مانند شبکه‌های عصبی کانولوشن [۲] با تعداد بسیار زیاد جمع و ضرب، نیاز به طراحی ساختارهای ویژه با انرژی مصرفی کم دارند. از این رو، استفاده از روش‌های جدیدی مانند سیستم‌های عددی خاص غیرمتداول جهت طراحی کارآمد سیستم‌های نهفته جهت کاهش انرژی مصرفی محاسبات مورد توجه محققین قرار گرفته‌است [۱]. سیستم‌های عددی مانده‌ای<sup>۱</sup> [۴]، مرسوم‌ترین سیستم عددی خاص منظوره با قابلیت انجام محاسبات جمع، تفریق و ضرب با انتشار محدود و درون پیمانهای رقم نقلی است که این امر موجب کاهش توان مصرفی و افزایش سرعت مدارهای محاسباتی مبتنی بر این سیستم عددی شده‌است [۵]. از این رو، سیستم‌های عددی مانده‌ای در سال‌های اخیر کاربرد فراوانی در پیاده‌سازی سخت‌افزاری الگوریتم‌های محاسباتی مبتنی بر جمع و ضرب مانند پردازش سیگنال‌های دیجیتال [۵]، رمزنگاری [۶] و شبکه‌های عصبی کانولوشن عمیق [۷] پیدا کرده‌است.

در سیستم‌های عددی مانده‌ای ابتدا توسط مبدل مستقیم<sup>۲</sup>، اعداد وزنی دودوئی عادی بر تعدادی عدد دو به دو نسبت به هم اول که مجموعه پیمانهای<sup>۳</sup> سیستم نامیده می‌شوند، تقسیم می‌شوند و باقی‌مانده‌های حاصل، نمایش مانده‌ای آن عدد را درون سیستم اعداد مانده‌ای شکل می‌دهند. سپس اعمال حسابی جمع، تفریق یا ضرب به صورت پیمانهای بر روی نمایش مانده‌ای عمل‌وندها انجام می‌شوند. مزیت اصلی سیستم اعداد مانده‌ای در این است که مابین مدارهای حسابی که به‌ازای هر پیمانهای وجود دارند، انتشار رقم نقلی وجود ندارد و فقط درون مدارهای پیمانهای انتشار محدود رقم نقلی وجود دارد. از این رو، امکان پیاده‌سازی موازی اعمال ضرب و جمع پیمانهای بر روی باقی‌مانده‌ها به وجود می‌آید که منجر به بالا رفتن سرعت و کاهش توان مصرفی می‌شود. همچنین، امکان تحمل‌پذیری خطا در صورت استفاده از پیمانهای افزونه، فراهم می‌شود. در انتها نتیجه محاسبه، توسط مبدل معکوس<sup>۴</sup> به حالت وزنی دودوئی عادی تبدیل می‌شود تا توسط بقیه اجزای سیستم قابل فهم و پردازش باشد [۸]. در نتیجه مبدل‌های مستقیم و معکوس سربار سیستم می‌باشند و اگر ناکارآمد و با سخت‌افزار و تأخیر زیاد طراحی شوند، منجر به کاهش بهره سرعت ناشی از موازی‌سازی مدارهای حسابی پیمانهای داخلی سیستم می‌شود. لذا تحقیق بسیار زیادی بر روی طراحی کارآمد این مبدل‌ها و به‌خصوص مبدل معکوس که پیچیدگی بسیار زیادی دارد انجام شده‌است.

طراحی مبدل مستقیم یا همان مبدل دودوئی به سیستم اعداد مانده‌ای، مبتنی بر جمع‌کننده‌های ذخیره رقم نقلی<sup>۵</sup>، برای اولین بار در [۹] ارائه شد. از آن سال تاکنون، ساختار درختی مبتنی بر جمع‌کننده‌های ذخیره رقم نقلی که بعد از آن یک جمع‌کننده پیمانهای<sup>۶</sup>

قرار دارد، معماری رایج طراحی مبدل مستقیم شده‌است [۹، ۱۰-۱۲]. برخلاف مبدل مستقیم که ساختاری موازی دارد، مبدل معکوس بسیار پیچیده و وابسته به نوع الگوریتم تبدیل و مجموعه پیمانهای است. به همین دلیل، محققان از جنبه‌های مختلف، طراحی‌های متفاوتی برای مبدل معکوس ارائه داده‌اند تا کارایی این واحد محاسباتی که در واقع گلوگاه سیستم اعداد مانده‌ای است، افزایش یابد. برای اولین بار، استفاده از جمع‌کننده ذخیره رقم نقلی در مبدل معکوس در [۱۳] ارائه شد و این مقاله آغازگر سبکی جدید در طراحی مبدل‌های معکوس شد. سپس جمع‌کننده‌های ذخیره رقم نقلی در [۱۴-۱۶] برای طراحی مبدل‌های معکوس کارآمد برای مجموعه پیمانهای سه و چهار پیمانهای مبتنی بر تئوری باقی‌مانده چینی استفاده شدند. همچنین، در [۱۷] از تئوری باقی‌مانده چینی جدید و نیز جمع‌کننده‌های ذخیره رقم نقلی جهت طراحی مبدل معکوس برای دو مجموعه پیمانهای چهار پیمانهای استفاده شد. علاوه بر تئوری‌های باقی‌مانده چینی کلاسیک و جدید که در [۱۴-۱۷] مورد توجه قرار گرفته‌اند، تبدیل درهم مینا<sup>۷</sup> نیز برای طراحی مبدل معکوس می‌تواند استفاده شود که نمونه‌هایی از استفاده از آن برای طراحی مبدل معکوس در [۱۸، ۱۹] ارائه شده‌است. تئوری‌های باقیمانده چینی ساختار موازی دارند ولی در انتها نیاز به یک جمع پیمانهای بزرگ دارند ولی تبدیل درهم مینا نیاز به چندین جمع پیمانهای کوچک با ساختاری تریبی دارد. در مجموع، در تمامی این کارها شیوه رسیدن به مبدل معکوس بهتر بر پایه، (۱) انتخاب مجموعه پیمانهای جدید، (۲) انتخاب الگوریتم تبدیل مناسب و نیز (۳) ساده‌سازی نوآورانه فرمول‌های تبدیل معکوس است.

از طرفی دیگر، اخیراً شیوه‌ای جدید برای بالا بردن کارایی مبدل‌های معکوس در [۲۰-۲۲] مورد توجه قرار گرفته‌است. به این صورت که به جای تغییر مجموعه پیمانهای یا الگوریتم تبدیل، اجزای سخت‌افزاری حسابی ویژه استفاده در مبدل معکوس طراحی شود تا این واحدهای محاسباتی جدید که مبتنی بر نیازهای مبدل معکوس ساخته شده‌اند، در معماری هر نوع مبدل معکوسی قرار گیرند تا کارایی افزایش یابد. مرجع [۲۰] آغازگر این تحول اساسی در طراحی مبدل معکوس شد به این صورت که یک واحد جمع پیمانهای ترکیبی مبتنی بر جمع‌کننده پیشوندی موازی ویژه مبدل‌های معکوس طراحی کرد. در همین راستا در [۲۱]، یک جمع‌کننده پیمانهای جدید ویژه علامت‌دار کردن خروجی مبدل‌های معکوس طراحی شد. این سبک طراحی، در [۲۲] به کل سیستم اعداد مانده‌ای به‌جز مبدل مستقیم، بسط داده شد و یک واحد سخت‌افزاری چندمنظوره طراحی شد که توانایی انجام تبدیل معکوس، تشخیص علامت، تقسیم بر یک عدد ثابت و مقایسه اعداد سیستم اعداد مانده‌ای را دارد. زیرا روند و الگوریتم طراحی مبدل مستقیم و معکوس متفاوت است. از طرفی فارغ از عدم اشتراک الگوریتمی، باتوجه به این که هر دو مبدل مستقیم و معکوس مبتنی بر جمع‌کننده‌های ذخیره رقم نقلی و جمع‌کننده پیمانهای ساخته می‌شوند، این سؤال در ذهن تداعی می‌شود

سیس اعمال حسابی جمع، تفریق یا ضرب بر روی باقی‌مانده‌های اعدادی که وارد سیستم اعداد مانده‌ای شده‌اند، به صورت موازی و بدون انتشار رقم نقلی مابین ارقام مانده‌ای به صورت زیر انجام می‌شود:

$$X \otimes Y \xrightarrow{RNS} (|x_1 \otimes y_1|_{m_1}, |x_2 \otimes y_2|_{m_2}, \dots, |x_n \otimes y_n|_{m_n}) \quad (3)$$

که  $\otimes$  نمایانگر یکی از عملیات جمع، تفریق یا ضرب است:

$$\otimes \in \{+, -, \times\} \quad (4)$$

سپس بعد از انجام محاسبات، حاصل که به صورت مجموعه‌ای از ارقام مانده‌ای است توسط یک الگوریتم تبدیل عکس مانند تئوری باقی‌مانده چینی یا تبدیل درهم مینا به معادل وزنی تبدیل می‌شود [۸]. لازم به ذکر است که چندین عملیات حسابی بر روی باقی‌مانده‌ها می‌تواند انجام شود. به عنوان مثال، در پیاده‌سازی فیلترهای دیجیتال مورد نیاز در پردازش سیگنال دیجیتال مبتنی بر سیستم اعداد مانده‌ای [۱]، تعداد زیادی عملیات ضرب و جمع بر روی باقی‌مانده‌ها انجام می‌شود و سپس حاصل نهایی فیلتر، توسط مبدل معکوس از سیستم خارج می‌شود. لازم به ذکر است هرچقدر تعداد محاسبات جمع و ضرب پیمان‌های درون سیستمی نسبت به تعداد انجام تبدیل مستقیم یا عکس بیش‌تر باشد، کارایی سیستم بالاتر خواهد رفت زیرا بهره سرعت ناشی از پیاده‌سازی موازی محاسبات جمع و ضرب، بر سر بار ناشی از مبدل غالب خواهد شد.

الگوریتم تبدیل در هم مینا مبتنی بر مجموعه  $\{m_1, m_2, \dots, m_n\}$  و روابط حسابی زیر، تبدیل معکوس را انجام می‌دهد. به عبارتی یک عدد مانده‌ای مانند  $\{x_1, x_2, \dots, x_n\}$  را به معادل عادی وزنی  $X$  تبدیل می‌کند [۱۹، ۲۳]:

$$X = v_n \prod_{i=1}^{n-1} m_i + \dots + v_3 m_2 m_1 + v_2 m_1 + v_1 \quad (5)$$

رابطه (۵)، فرمول پایه‌ای الگوریتم تبدیل درهم مینا است که ورودی‌های آن، باقی‌مانده‌ها و نیز مقدار پیمان‌ها است و خروجی آن مقدار محاسبه‌شده عدد وزنی است. تعداد جمله‌های حاصل ضرب میانی، بستگی به تعداد پیمان‌های سیستم دارد (درواقع مقدار  $n$  در این فرمول، بیانگر تعداد پیمان‌ها است). به عنوان مثال، برای ۳ تا ۵ پیمان، فرمول (۵) به صورت (۶) تا (۸)، به ترتیب، در می‌آید.

$$X = v_3 m_2 m_1 + v_2 m_1 + v_1 \quad (6)$$

$$X = v_4 m_3 m_2 m_1 + v_3 m_2 m_1 + v_2 m_1 + v_1 \quad (7)$$

$$X = v_5 m_4 m_3 m_2 m_1 + v_4 m_3 m_2 m_1 + v_3 m_2 m_1 + v_2 m_1 + v_1 \quad (8)$$

ضرایب  $v_i$  نیز به صورت زیر بر حسب باقی‌مانده‌ها و نیز معکوس‌های ضربی به دست می‌آیند:

$$v_1 = x_1 \quad (9)$$

که آیا می‌توان دو واحد مجزا ولی با اجزای سخت‌افزاری یکسان را در هم ادغام کرد که عمل کرد هر دو واحد را داشته باشد ولی با سخت‌افزار کمتر؟ در این مقاله، برای اولین بار در طی چندین دهه عمر سیستم اعداد مانده‌ای، یک واحد محاسباتی جدید که توانایی انجام هر دو عملیات تبدیل مستقیم و معکوس را دارد، پیشنهاد شده است. مهم‌ترین مزیت طرح پیشنهادی، کاهش چشم‌گیر سر بار ناشی از داشتن دو طراحی مجزا برای مبدل‌های مستقیم و معکوس است. در مبدل ترکیبی ارائه‌شده، هم خروجی تبدیل مستقیم (باقی‌مانده‌ها) و هم خروجی تبدیل معکوس (عدد وزنی دودویی)، مبتنی بر یک سخت‌افزار یکسان که توسط یک سیگنال کنترلی هدایت می‌شود، تولید می‌شوند. جهت نیل به این هدف، به شیوه‌ای نو، از الگوریتم تبدیل درهم مینا استفاده شده است که روابط محاسباتی تبدیل معکوس به فرمت روابط محاسباتی تبدیل مستقیم درآید. سپس مبتنی بر تعدادی مالتی‌پلکسر از سخت‌افزار به صورت اشتراکی استفاده می‌شود؛ به این ترتیب که ورودی صحیح مبتنی بر سیگنال کنترلی انتخاب و به جمع‌کننده‌های درونی مدار، وارد می‌شود. نتایج تجربی مبتنی بر پیاده‌سازی VLSI نشان از کاهش چشم‌گیر سطح تراشه در صورت استفاده از مدار پیشنهادی دارد زیرا به جای دو مبدل مجزا، یک مبدل دو منظوره به کار گرفته شده است.

در ادامه، در بخش ۲، مروری بر مفاهیم پایه سیستم اعداد مانده‌ای انجام خواهد شد. سپس در بخش ۳، یک بازنویسی از فرمول‌های تبدیل مستقیم و معکوس جهت ادغام ارائه خواهد شد. در بخش ۴ نیز ایده پیشنهادی به صورت کلی ارائه و ساختار سیستم اعداد مانده‌ای پیشنهادی رسم می‌شود. یک حالت خاص از طرح پیشنهادی برای مجموعه پیمان  $\{2^{n-1}, 2^{2n}, 2^{n+1}-1\}$  در بخش ۵ ارائه و در بخش ۶، کارایی آن مورد ارزیابی قرار خواهد گرفت. در انتها، نتیجه‌گیری در بخش ۷ ارائه خواهد شد.

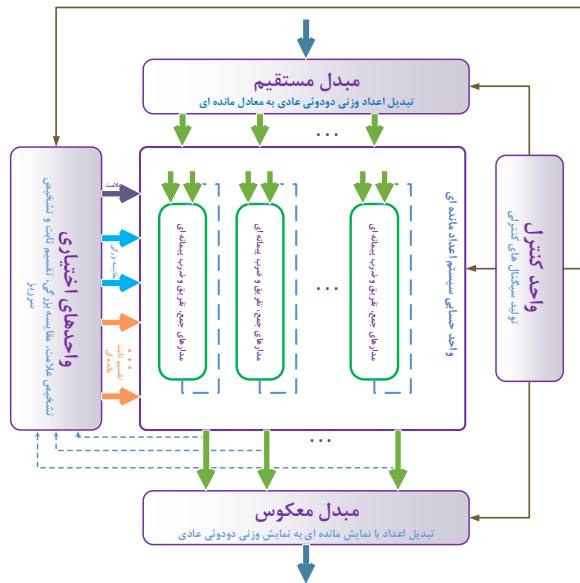
## ۲- سیستم اعداد مانده‌ای

پایه و اساس سیستم اعداد مانده‌ای، تعدادی عدد دو به دو نسبت به هم اول است که تشکیل مجموعه پیمان سیستم را می‌دهند و به صورت  $\{m_1, m_2, \dots, m_n\}$  نمایش داده می‌شوند. محدوده دینامیکی سیستم اعداد مانده‌ای بیانگر تعداد و محدوده اعدادی است که قابل نمایش در سیستم هستند و قابلیت ورود به سیستم توسط مبدل مستقیم را دارند. هم‌چنین مبتنی بر این تعریف، خروجی مبدل معکوس نیز در این محدوده قرار می‌گیرد. محدوده دینامیکی توسط رابطه زیر محاسبه می‌شود:

$$M = m_1 \times m_2 \times \dots \times m_n \quad (1)$$

به عبارتی حاصل ضرب پیمان‌ها، محدوده اعداد قابل نمایش در سیستم را می‌سازد که به صورت  $[0, M-1]$  نشان داده می‌شود [۲۳]. اعداد در این محدوده که به صورت عادی و وزنی هستند، توسط مبدل مستقیم بر اعداد درون مجموعه پیمان تقسیم و سپس باقی‌مانده‌ها، نمایش مانده‌ای آن عدد وزنی را در سیستم اعداد مانده‌ای به صورت زیر تشکیل می‌دهند:

$$X \xrightarrow{RNS} (x_1, x_2, \dots, x_n) \text{ where } x_i = |X|_{m_i} \text{ for } i=1, \dots, n \quad (2)$$



شکل ۱: ساختار مرسوم سیستم اعداد مانده‌ای [۵]

از این مرحله به بعد، فقط در سیستم اعداد مانده‌ای با نمایش مانده‌ای اعداد کار می‌شود و نمایش وزنی استفاده نمی‌شود. حال عملیات ضرب بر روی این دو عدد که وارد سیستم شده‌اند به صورت زیر انجام می‌شود:

$$(1,0,4) \times (2,3,1) \xrightarrow{RNS} (|1 \times 2|_3, |0 \times 3|_4, |4 \times 1|_5) = (2,0,4) \quad (16)$$

حال از الگوریتم تبدیل درهم مینا مبتنی بر (۶) می‌توان استفاده کرد تا این عدد حاصل ضرب از سیستم خارج و تبدیل به معادل وزنی عادی شود. در این راستا، ابتدا باید معکوس‌های ضربی مورد نیاز در (۱۰) و (۱۱) محاسبه شوند که مبتنی بر روش ارائه‌شده در [۸]، به صورت زیر به دست می‌آیند:

$$|m_1^{-1}|_{m_2} = |3^{-1}|_4 = 3 \quad (|3 \times 3|_4 = 1) \quad (17)$$

$$|m_1^{-1}|_{m_3} = |3^{-1}|_5 = 2 \quad (|3 \times 2|_5 = 1) \quad (18)$$

$$|m_2^{-1}|_{m_3} = |4^{-1}|_5 = 4 \quad (|4 \times 4|_5 = 1) \quad (19)$$

سپس مقدار باقی‌مانده‌ها و نیز معکوس‌های ضربی در (۹) تا (۱۱) قرار داده می‌شود تا مقدار ضرایب مورد نیاز در الگوریتم تبدیل درهم مینا به دست آید.

$$v_1 = x_1 = 2 \quad (20)$$

$$v_2 = \left| (x_2 - v_1) |m_1^{-1}|_{m_2} \right|_{m_2} = \left| (0 - 2) \times 3 \right|_4 = 2 \quad (21)$$

$$v_2 = \left| (x_2 - v_1) |m_1^{-1}|_{m_2} \right|_{m_2} \quad (10)$$

$$v_3 = \left| ((x_3 - v_1) |m_1^{-1}|_{m_3} - v_2) |m_2^{-1}|_{m_3} \right|_{m_3} \quad (11)$$

$$v_4 = \left| (((x_4 - v_1) |m_1^{-1}|_{m_4} - v_2) |m_2^{-1}|_{m_4} - v_3) |m_3^{-1}|_{m_4} \right|_{m_4} \quad (12)$$

بنابراین جمله اول رابطه فقط یک متغیر و باقی جمله‌ها، به صورت حاصل ضربی هستند که مقدار همه آن‌ها باید با هم جمع شود تا عدد وزنی نهایی به دست آید. البته توسط انتخاب پیمانه‌ها به فرم‌های  $2^n \pm 1$  یا  $2^n$ ، این ضرب‌ها تبدیل به شیفت می‌شوند و لذا دیگر نیازی به استفاده از ضرب‌کننده در ساختار مبدل معکوس، نیست. فرمول کلی به دست آوردن ضریب برای هر تعداد پیمانه‌ای که استفاده شود، نیز به صورت زیر است [۲۳]:

$$v_n = \left| \left( \left( (x_n - v_1) |m_1^{-1}|_{m_n} - v_2 \right) |m_2^{-1}|_{m_n} - \dots - v_{n-1} \right) |m_{n-1}^{-1}|_{m_n} \right|_{m_n} \quad (13)$$

عبارت  $|m_i^{-1}|_{m_j}$  نشان‌دهنده معکوس ضربی پیمانه  $m_i$  مبتنی بر پیمانه  $m_j$  است. معکوس‌های ضربی جداگانه محاسبه می‌شوند و در روابط بالا قرار می‌گیرند تا ضرایب الگوریتم تبدیل درهم مینا به دست آیند و سپس با داشتن ضرایب، مقدار  $X$  از رابطه (۵) محاسبه می‌شود.

علاوه بر مبدل‌های مستقیم و معکوس، و نیز واحدهای حسابی جمع و ضرب، باتوجه به کاربرد هدف، واحدهای تشخیص علامت، مقایسه بزرگی، تقسیم ثابت و تشخیص سرریز نیز ممکن است در سیستم طراحی شوند. زیرا به دلیل خاصیت غیروزنی بودن سیستم اعداد مانده‌ای و برخلاف سیستم اعداد وزنی، امکان تشخیص علامت با یک بیت و یا مقایسه بزرگی از طریق مقایسه بیت‌های متناظر وجود ندارد. یک بلوک دیاگرام از ساختار کلی سیستم اعداد مانده‌ای در شکل ۱ نشان داده شده است.

مثال ۱: مجموعه پیمانه  $\{3, 4, 5\}$  را در نظر بگیرید. محدوده دینامیکی مبتنی بر (۱) می‌شود ۶۰، به عبارتی اعداد ۰ تا ۵۹ را می‌توان وارد این سیستم کرد و سپس بر روی آن‌ها محاسبات موازی و سریع انجام داد. حال فرض کنید اعداد ۴ و ۱۱ توسط مبدل مستقیم مبتنی بر (۲) به صورت زیر وارد سیستم شوند (این اعداد بر تک تک اعداد درون مجموعه پیمانه تقسیم و سپس باقی‌مانده‌ها نگه داشته می‌شوند):

$$4 \xrightarrow{RNS} (|4|_3, |4|_4, |4|_5) \rightarrow (1,0,4) \quad (14)$$

$$11 \xrightarrow{RNS} (|11|_3, |11|_4, |11|_5) \rightarrow (2,3,1) \quad (15)$$

$$x_i = |X|_{m_i} = \left| \begin{array}{c} 2^{2n} (X_{3n-1} \cdots X_{2n}) + 2^n (X_{2n-1} \cdots X_n) \\ + (X_{n-1} \cdots X_0) \end{array} \right|_{m_i} \quad (28)$$

باتوجه به این که  $|2^{2n}|_{2^n} = |2^n|_{2^n} = 0$  است،  $x_1$  به صورت زیر به دست می آید:

$$x_1 = |X|_{2^n} = X_{n-1} \cdots X_0 \quad (29)$$

از طرفی، باتوجه به این که  $|2^{2n}|_{2^{n-1}} = |2^n|_{2^{n-1}} = 1$  است،  $x_2$  به صورت زیر به دست می آید:

$$x_2 = |X|_{2^{n-1}} = \left| \begin{array}{c} X_{3n-1} \cdots X_{2n} \\ \underbrace{\hspace{10em}}_{f_2} \\ + X_{2n-1} \cdots X_n \\ \underbrace{\hspace{10em}}_{f_1} \\ + X_{n-1} \cdots X_0 \\ \underbrace{\hspace{10em}}_{f_0} \end{array} \right|_{2^{n-1}} = |f_2 + f_1 + f_0|_{2^{n-1}} \quad (30)$$

در نهایت، باتوجه به این که  $|2^{2n}|_{2^{n+1}} = 1$  و  $|2^n|_{2^{n+1}} = -1$  است،  $x_3$  به صورت زیر به دست می آید:

$$x_3 = |X|_{2^{n+1}} = \left| \begin{array}{c} X_{3n-1} \cdots X_{2n} \\ \underbrace{\hspace{10em}}_{f_2} \\ - X_{2n-1} \cdots X_n \\ \underbrace{\hspace{10em}}_{f_1} \\ + X_{n-1} \cdots X_0 \\ \underbrace{\hspace{10em}}_{f_0} \end{array} \right|_{2^{n+1}} = |f_2 - f_1 + f_0|_{2^{n+1}} \quad (31)$$

هم چنین، به روابط حسابی مربوط به تبدیل معکوس در (۵) تا (۱۳) اشاره شد. حال، ضریب های مربوط به تبدیل درهم مینا (فرمول های (۱۰) تا (۱۳))، به صورت زیر بازنویسی می شوند. لازم به ذکر است منظور از  $k_i$  معکوس ضریبی است:

$$v_2 = \left| \begin{array}{c} k_1 x_2 + (-k_1 v_1) \\ \underbrace{\hspace{10em}}_{v_{21}} \quad \underbrace{\hspace{10em}}_{v_{22}} \end{array} \right|_{m_2} = |v_{21} + v_{22}|_{m_2} = \left| \sum_{i=1}^2 v_{2i} \right|_{m_2} \quad (32)$$

$$v_3 = \left| \begin{array}{c} k_3 k_2 x_3 + (-k_3 k_2 v_1) + (-k_3 v_2) \\ \underbrace{\hspace{10em}}_{v_{31}} \quad \underbrace{\hspace{10em}}_{v_{32}} \quad \underbrace{\hspace{10em}}_{v_{33}} \end{array} \right|_{m_3} \quad (33)$$

$$= |v_{31} + v_{32} + v_{33}|_{m_3} = \left| \sum_{i=1}^3 v_{3i} \right|_{m_3}$$

$$v_4 = \left| \begin{array}{c} k_4 k_3 k_2 x_3 + (-k_4 k_3 k_2 v_1) + (-k_4 k_3 v_2) + (-k_4 v_3) \\ \underbrace{\hspace{10em}}_{v_{41}} \quad \underbrace{\hspace{10em}}_{v_{42}} \quad \underbrace{\hspace{10em}}_{v_{43}} \quad \underbrace{\hspace{10em}}_{v_{44}} \end{array} \right|_{m_4} \quad (34)$$

$$= |v_{41} + v_{42} + v_{43} + v_{44}|_{m_4} = \left| \sum_{i=1}^4 v_{4i} \right|_{m_4}$$

$$v_3 = \left| \begin{array}{c} ((x_3 - v_1) m_1^{-1} |_{m_3} - v_2) |_{m_3} m_2^{-1} |_{m_3} \\ = |((4-2)2-2) \times 4|_5 = 3 \end{array} \right|_{m_3} \quad (22)$$

در نهایت با قراردادن مقدار پیمانها و ضرایب در (۶)، رابطه زیر به دست می آید:

$$X = v_3 m_2 m_1 + v_2 m_1 + v_1 = 3 \times 4 \times 3 + 2 \times 3 + 2 = 44 \quad (23)$$

به عبارتی ۴۴ برابر با حاصل تبدیل معکوس عدد مانده ای (۴، ۰، ۲) است. در نتیجه به جای این که بر روی ۴ و ۱۱ مستقیم ضرب را انجام دهیم، بر روی باقی مانده های آنها محاسبه انجام شد و صحت آن نشان می دهد که اطلاعات طلای نهفته در باقی مانده ها، می تواند به طور منحصربه فرد بیان گر نمایش اعداد باشد.

### ۳- بازنویسی روابط حسابی تبدیل مستقیم و معکوس

در این بخش، روابط حسابی مربوط به تبدیل مستقیم و نیز تبدیل معکوس به صورت جداگانه ارائه و سپس در بخش بعدی، نحوه ادغام آنها جهت رسیدن به یک طراحی یک پارچه ترکیبی شرح داده می شود.

روند طراحی مبدا مستقیم به این صورت است که باید باقی مانده تقسیم عمل وند وزنی ورودی بر هر یک از پیمانها به دست آید؛ لذا مبدا مستقیم، ساختاری موازی و مستقل برای هر پیمانها دارد. حال جهت تبدیل عدد وزنی  $X$  مبتنی بر مجموعه پیمانها  $\{m_1, m_2, \dots, m_n\}$  به باقی مانده ها (به عبارتی  $x_i$  ها)، از رابطه زیر می توان استفاده کرد:

$$x_i = |X|_{m_i} = |X_{MB-1} \cdots X_1 X_0|_{m_i} \quad \text{for } i=1, \dots, n \quad (24)$$

که  $MB$  تعداد بیت های محدوده دینامیکی (یعنی  $M$ ) است و  $m_i$  نشان گر پیمانها مدنظر است [۱]. لازم به ذکر است که  $X_i$  نشان دهنده بیت  $i$  ام از عدد دودویی  $X$  است. باتوجه به تعداد بیت های پیمانها  $m_i$  با  $m_{i+1}$  نشان داده می شود، بیت های عمل وند به دسته های  $m_{ib}$  بیتی تقسیم می شوند:

$$x_i = |X|_{m_i} = \left| \begin{array}{c} 2^{MB} (X_{2MB-1} \cdots X_{km_b}) + \cdots \\ + 2^{m_b} (X_{2m_b-1} \cdots X_{m_n}) + (X_{m_b-1} \cdots X_0) \end{array} \right|_{m_i} \quad (25)$$

حال دیگر نیازی به انجام عمل تقسیم برای محاسبه باقی مانده نیست و توسط تعدادی جمع و ضرب، باقی مانده به دست می آید. به عبارتی (۲۵) را می توان به صورت زیر بازنویسی کرد:

$$x_i = \left| \sum_{j=0}^k f_j \right|_{m_i} \quad (26)$$

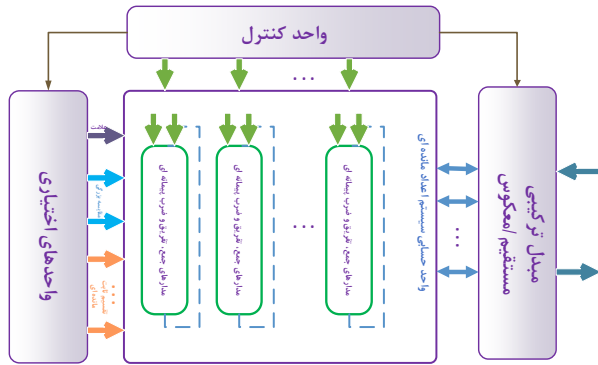
که بردارهای دودویی  $f_i$  به صورت زیر تعریف می شوند:

$$f_j = 2^{j m_b} (X_{(j+1)m_b-1} \cdots X_{j m_b}) \quad (27)$$

البته لازم به ذکر است که معمولاً پیمانها به صورت توانی از ۲ (مانند  $2^n \pm 1$ ) هستند و لذا ضرب های مورد نیاز در (۲۵) حذف (تبدیل به شیف) می شوند. به عنوان مثال، برای ۳ نوع پیمانها مشهور سیستم اعداد مانده ای یعنی  $2^n$ ،  $2^n - 1$  و  $2^n + 1$  مطابق [۸]، باقی مانده ها به صورت زیر به دست می آیند:

نیست و یک واحد ترکیبی جایگزین آن‌ها می‌شود که در شکل ۲ نشان داده شده‌است.

مقایسه ساختار مرسوم سیستم اعداد مانده‌ای (شکل ۱) و ساختار پیشنهادی (شکل ۲)، آشکار می‌سازد که پیچیدگی سخت‌افزاری کلی سیستم، کاهش پیدا کرده‌است. به عبارتی، با حذف مبدل‌های مستقیم و معکوس جداگانه، و جای‌گزینی مبدل ترکیبی، سخت‌افزار و نیز سیگنال‌های کنترلی موردنیاز، کاهش پیدا می‌کند که منجر به افزایش کارایی کلی سیستم می‌شود.



شکل ۲: ساختار سیستم اعداد مانده‌ای پیشنهادی

#### ۵- روش پیشنهادی: طراحی موردی

در این بخش جهت مشخص شدن ابعاد کامل روش پیشنهادی، پیاده‌سازی آن مبتنی بر مجموعه پیمانه  $\{2^{2n}, 2^{2n-1}, 2^{2n+1}-1\}$ ، شرح داده می‌شود. این مجموعه پیمانه در [۱۹] معرفی شده‌است و به دلیل محدوده دینامیکی بزرگ  $4n+1$  بیتی و نیز استفاده از پیمانه‌های خوش‌فرم و نیز عدم استفاده از پیمانه به فرم  $2^n+1$  (که نسبت به بقیه پیمانه‌ها پیچیده‌تر است)، به سیستم اعداد مانده‌ای با کارایی بالا منتج می‌شود. در ابتدا طراحی مبدل مستقیم و معکوس برای این مجموعه پیمانه شرح داده می‌شود و سپس طرح ترکیبی پیشنهادی مبتنی بر (۳۶) تا (۳۸) ارائه می‌شود.

مبدل مستقیم برای مجموعه پیمانه  $\{2^{2n}, 2^{2n-1}, 2^{2n+1}-1\}$  بر طبق (۲۴) تا (۲۷) به دست می‌آید. در ابتدا، اولین باقی‌مانده مبتنی بر پیمانه  $2^{2n}$  به صورت زیر به دست می‌آید:

$$x_1 = |X|_{2^{2n}} = |X_{4n} \cdots X_1 X_0|_{2^{2n}} = \left| \begin{matrix} 2^{2n} (X_{4n} \cdots X_{2n+1} X_{2n}) \\ + X_{2n-1} \cdots X_1 X_0 \end{matrix} \right|_{2^{2n}} = \underbrace{X_{2n-1} \cdots X_1 X_0}_{2n \text{ bits}} \quad (39)$$

به عبارتی مبدل مستقیم برای پیمانه  $2^{2n}$  نیازی به مدار محاسباتی ویژه ندارد و فقط کافی است که  $2n$  بیت کم ارزش ورودی، انتخاب شود. باقی مانده دوم مبتنی بر پیمانه  $2^n-1$  به صورت زیر محاسبه می‌شود:

$$v_n = \left| \begin{matrix} (k_n \cdots k_2) x_n + (-k_n \cdots k_2) v_1 + \cdots \\ + (-k_n v_{n-1}) \end{matrix} \right|_{m_n} = \left| \sum_{i=1}^n v_{ni} \right|_{m_n} \quad (35)$$

#### ۴- روش پیشنهادی: الگوریتم کلی

همان‌طور که در بخش قبلی توضیح داده شد، هر دو عملیات تبدیل مستقیم و معکوس نیاز به چندین جمع پیمانه‌ای دارند و نکته جالب این است که دقیقاً هر دو عملیات نیاز به جمع‌های پیمانه‌ای مجزا به‌ازای هر پیمانه دارند. لذا از این اشتراک می‌توان استفاده کرد و یک طراحی ترکیبی جهت انجام هر دو عملیات طراحی کرد. به عبارتی هدف، طراحی واحدی است که یک عدد وزنی دودوئی مانند  $X$ ، تعدادی باقی‌مانده مانند  $(y_1, y_2, \dots, y_n)$  و سیگنال کنترلی  $Control$  وارد آن شود و سپس باقی مانده‌های عدد وزنی  $X$  یعنی  $(x_1, x_2, \dots, x_n)$  و نیز معادل وزنی باقی مانده‌های ورودی  $y_i$  یعنی عدد عادی وزنی  $Y$  از آن خارج شود که در جدول ۱ نشان داده شده‌است.

جدول ۱: ورودی و خروجی‌های واحد ترکیبی تبدیل

مستقیم/معکوس			
خروجی	ورودی	مد کاری	سیگنال کنترلی
$Y$	$y_1, y_2, \dots, y_n$	مبدل معکوس	۰
$x_1, x_2, \dots, x_n$	$X$	مبدل مستقیم	۱

حال با توجه به (۵) تا (۱۳) و (۳۲) تا (۳۵) برای مبدل معکوس، و (۲۴) تا (۲۷) برای مبدل مستقیم، می‌توان به فرمول‌های ادغامی زیر برای طراحی واحد ترکیبی تبدیل مستقیم/معکوس رسید:

$$Y = h_n \prod_{i=1}^{n-1} m_i + \cdots + h_3 m_2 m_1 + h_2 m_1 + h_1 \quad (36)$$

$$x_i = h_i \quad \text{for } i=1, \dots, n \quad (37)$$

که ضرایب  $h_i$  این‌گونه محاسبه می‌شوند:

$$h_i = \begin{cases} \left| \sum_{j=1}^i v_{ij} \right|_{m_i} & \text{if Control} = 0 \\ \left| \sum_{j=0}^k f_j \right|_{m_i} & \text{if Control} = 1 \end{cases} \quad (38)$$

درحالتی که سیگنال کنترلی  $Control$  توسط واحد کنترل، صفر شود، مد کاری مبدل معکوس انتخاب می‌شود و مقدار ضرایب تبدیل درهم مینا که در (۹) تا (۱۳) ارائه و در (۳۲) تا (۳۵) به صورت جمع چند عمل‌وندی بازنویسی شدند، در  $h_i$  یعنی (۳۸) قرار می‌گیرد و سپس مقدار  $Y$  بر طبق (۳۶) محاسبه می‌شود. هم‌چنین با یک شدن سیگنال  $Control$ ، مدار در حالت عمل‌کردی مبدل مستقیم قرار می‌گیرد و منطبق بر روابط (۲۴) تا (۲۷)، عملوند مناسب در  $h_i$  قرار می‌گیرد تا تبدیل مستقیم مبتنی بر (۳۷) انجام شود و باقی‌مانده‌ها به دست آیند. حال دیگر نیازی به مدارهای مجزا برای مبدل‌های مستقیم و معکوس

مبدل معکوس برای مجموعه پیمانه  $\{2^{2n}, 2^n-1, 2^{n+1}-1\}$  از طریق قراردادن مقدار پیمانه‌ها و نیز معکوس‌های ضربی در (۵) تا (۱۳) به دست می‌آید و سپس مبتنی بر (۳۲) تا (۳۵) به صورت جمع برداری پیمانه‌ای، ساده می‌شوند. از آنجایی که طراحی مبدل معکوس برای مجموعه پیمانه  $\{2^{2n}, 2^n-1, 2^{n+1}-1\}$  مبتنی بر تبدیل درهم مینا به صورت کامل در [۱۹] انجام شده است، در اینجا فقط روابط نهایی تبدیل معکوس که در [۱۹] به صورت کامل محاسبه و اثبات شده‌اند، استفاده می‌شود. از این رو، برای تبدیل باقی‌مانده‌ها به معادل وزنی داریم (رابطه‌های (۱۵) تا (۱۷) در [۱۹]):

$$Y = 2^{2n}(2^n - 1)v_3 + 2^{2n}v_2 + v_1 \quad (46)$$

که ضرایب آن عبارتند از:

$$v_1 = x_1 \quad (47)$$

$$v_2 = |x_2 - x_1|_{2^n-1} \quad (48)$$

$$v_3 = |((x_3 - x_1)^2 - v_2)(-2)|_{2^{n+1}-1} \quad (49)$$

روابط فوق بعد از ساده‌سازی به صورت زیر در می‌آیند (رابطه‌های (۳۰) و (۳۷) در [۱۹]):

$$Y = 2^{2n}(v_2 + 2^n v_3 - v_3) + x_1 \quad (50)$$

$$v_2 = |w_1 + w_{21} + w_{22}|_{2^n-1} \quad (51)$$

$$v_3 = |w_3 + w_4 + w_{51} + w_{52}|_{2^{n+1}-1} \quad (52)$$

که بردارهای  $w_i$  به صورت زیر هستند (رابطه‌های (۳۲) تا (۳۶) در [۱۹]):

$$w_1 = \underbrace{x_{2,n-1} \cdots x_{2,1} x_{2,0}}_{n \text{ bits}} \quad (53)$$

$$w_{21} = \underbrace{\bar{x}_{1,n-1} \cdots \bar{x}_{1,1} \bar{x}_{1,0}}_{n \text{ bits}} \quad (54)$$

$$w_{22} = \underbrace{\bar{x}_{1,2n-1} \cdots \bar{x}_{1,n+1} \bar{x}_{1,n}}_{n \text{ bits}} \quad (55)$$

$$w_3 = \underbrace{v_{2,n-1} \cdots v_{2,1} v_{2,0}}_{n \text{ bits}} 0 \quad (56)$$

$$w_4 = \underbrace{\bar{x}_{3,n-3} \cdots \bar{x}_{3,1} \bar{x}_{3,0} \bar{x}_{3,n} \bar{x}_{3,n-1} \bar{x}_{3,n-2}}_{n-2 \text{ bits}} \quad (57)$$

$$w_{51} = \underbrace{x_{1,n-3} \cdots x_{1,1} x_{1,0} x_{1,n} x_{1,n-1} x_{1,n-2}}_{n-2 \text{ bits}} \quad (58)$$

$$w_{52} = \underbrace{x_{1,2n-2} \cdots x_{1,n+2} x_{1,n+1}}_{n-2 \text{ bits}} 00 x_{1,2n-1} \quad (59)$$

ساختار سخت‌افزاری مبدل معکوس برای مجموعه پیمانه سه‌تایی  $\{2^{2n}, 2^n-1, 2^{n+1}-1\}$  مبتنی بر روابط حسابی فوق در شکل ۴ نشان داده شده است. همان‌طور که مشاهده می‌شود از جمع‌کننده‌های ذخیره رقم نقلی و انتشار رقم نقلی چرخشی مطابق شیوه ارائه‌شده در [۱۳]، استفاده شده است.

بررسی شکل‌های ۳ و ۴، شباهت و اشتراک سخت‌افزارهای مبدل‌های مستقیم و معکوس را مشخص می‌کند. برای پیاده‌سازی (۴۰) و (۴۳)

$$\begin{aligned} x_2 &= |X|_{2^n-1} = |X_{4n} \cdots X_1 X_0|_{2^n-1} \\ &= \left| \begin{array}{l} 2^{4n} X_{4n} + 2^{3n} \underbrace{(X_{4n-1} \cdots X_{3n+1} X_{3n})}_{n \text{ bits}} \\ + 2^{2n} \underbrace{(X_{3n-1} \cdots X_{2n+1} X_{2n})}_{n \text{ bits}} \\ + 2^n \underbrace{(X_{2n-1} \cdots X_{n+1} X_n)}_{n \text{ bits}} + \underbrace{X_{n-1} \cdots X_1 X_0}_{n \text{ bits}} \end{array} \right|_{2^n-1} \quad (40) \\ &= \left| \sum_{i=0}^4 f_i \right|_{2^n-1} \end{aligned}$$

که بردارهای دودویی  $f_i$ ، این‌گونه می‌باشند:

$$f_i = \underbrace{X_{(i+1)n-1} \cdots X_{in+1} X_{in}}_{n \text{ bits}} \text{ for } i = 0 \text{ to } 3 \quad (41)$$

$$f_4 = |X_{4n} \times 2^{4n}|_{2^n-1} = 0 \cdots 00 X_{4n} \quad (42)$$

در نهایت، باقی‌مانده سوم مبتنی بر پیمانه  $2^{n+1}-1$  به صورت زیر محاسبه می‌شود:

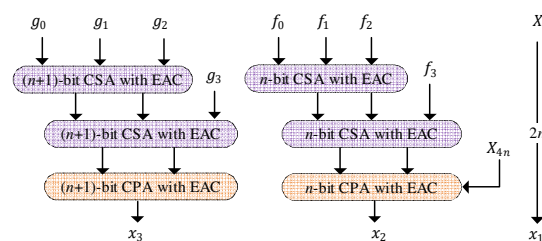
$$\begin{aligned} x_3 &= |X|_{2^{n+1}-1} = |X_{4n} \cdots X_1 X_0|_{2^{n+1}-1} \\ &= \left| \begin{array}{l} 2^{3n+3} \underbrace{(000 X_{4n} \cdots X_{3n+4} X_{3n+3})}_{n-2 \text{ bits}} \\ + 2^{2n+2} \underbrace{(X_{3n+2} \cdots X_{2n+3} X_{2n+2})}_{n+1 \text{ bits}} \\ + 2^{n+1} \underbrace{(X_{2n+1} \cdots X_{n+2} X_{n+1})}_{n+1 \text{ bits}} + \underbrace{X_n \cdots X_1 X_0}_{n+1 \text{ bits}} \end{array} \right|_{2^{n+1}-1} \quad (43) \\ &= \left| \sum_{i=0}^3 g_i \right|_{2^{n+1}-1} \end{aligned}$$

باتوجه به این‌که به‌ازای  $k \geq 1$  رابطه  $|2^{k(n+1)}|_{2^{n+1}-1} = 1$  برقرار است، بردارهای دودویی  $g_i$  به صورت زیر می‌توانند محاسبه شوند:

$$g_i = \underbrace{X_{(i+1)n+i} \cdots X_{in+i+1} X_{in+i}}_{n+1 \text{ bits}} \text{ for } i = 0 \text{ to } 2 \quad (44)$$

$$g_3 = \underbrace{000 X_{4n} \cdots X_{3n+4} X_{3n+3}}_{3 \text{ bits}} \quad (45)$$

همان‌طور که از (۴۰) و (۴۳) مشاهده می‌شود، برای به‌دست‌آوردن باقی‌مانده‌ها، فقط کافی است که تعدادی بردار دودویی با هم جمع شوند. این جمع می‌تواند مبتنی بر روش [۹] توسط جمع‌کننده‌های ذخیره رقم نقلی و یک جمع‌کننده انتشار رقم نقلی با قابلیت چرخش رقم نقلی انتهایی انجام شود که در شکل ۳ نشان داده شده است.



شکل ۳: مبدل مستقیم برای مجموعه پیمانه  $\{2^{2n}, 2^n-1, 2^{n+1}-1\}$

همان‌طور که در شکل ۵ مشاهده می‌شود، مالتی پلکس‌هایی که در ورودی جمع‌کننده‌های ذخیره رقم نقلی اضافه شده به همراه سیگنال کنترلی *Control*، عمل کرد مدار را مشخص می‌کنند. اگر *Control* صفر باشد، حالت عمل‌کردی تبدیل معکوس انتخاب می‌شود و  $(y_1, y_2, y_3)$  ورودی‌های مدار و *Y* خروجی خواهد بود. هم‌چنین اگر *Control* یک باشد، حالت عمل‌کردی تبدیل مستقیم انتخاب می‌شود و *X* ورودی و  $(x_1, x_2, x_3)$  خروجی‌های مدار خواهند بود.

تعداد و نوع جمع‌کننده‌های استفاده‌شده در مدار ترکیبی پیشنهادی در شکل ۵، دقیقاً همانند [۱۹] یعنی شکل ۴ است و فقط یک جمع‌کننده ذخیره رقم نقلی اضافه شده است تا چهارمین عمل‌وند مورد نیاز جهت تبدیل مستقیم در (۴۰) را پوشش دهد. هم‌چنین، پنجمین عمل‌وند مورد نیاز در (۴۰) یعنی  $f_4$  در (۴۲)، فقط توسط اتصال  $X_{4n}$  به ورودی رقم نقلی جمع‌کننده پیمان‌های، در جمع دخالت داده شده است. این نکته باعث شد تا از یک جمع‌کننده ذخیره رقم نقلی اضافی صرف‌نظر شود. لازم به ذکر است که آخرین جمع‌کننده انتشار رقم نقلی استفاده‌شده در شکل ۵، فقط جهت انجام تبدیل معکوس نیاز است. به عبارتی در حالت کاری تبدیل مستقیم، این جمع‌کننده نقشی ندارد.

مقایسه مبدا ترکیبی پیشنهادی (شکل ۵) و مبدا‌های مستقیم و معکوس عادی (شکل‌های ۳ و ۴)، نشان می‌دهد که چگونه مالتی پلکس‌ها امکان اشتراک سخت افزار را فراهم کرده‌اند. همان‌گونه که مشاهده می‌شود، مالتی پلکس‌ها با استفاده از سیگنال کنترلی، ورودی‌های مناسب را به جمع‌کننده‌های ذخیره رقم نقلی هدایت می‌کنند.

دقیقاً همانند (۵۱) و (۵۲) نیاز به جمع‌کننده‌های ذخیره رقم نقلی و جمع‌کننده‌های پیمان‌های است. لذا، باتوجه به (۳۶) تا (۳۸) می‌توان به فرمول‌های ادغامی جهت طراحی مبدا ترکیبی مستقیم/معکوس دست یافت. از این‌رو، با قراردادن (۳۹) تا (۴۵) و (۵۰) تا (۵۲) در (۳۶) تا (۳۸) و باتوجه به جدول ۱، رابطه (۶۰) به دست می‌آید.

$$Y = 2^{2n}(2^n - 1)h_3 + 2^{2n}h_2 + h_1 \quad (60)$$

$$= 2^{2n}(h_2 + 2^n h_3 - h_3) + y_1$$

$$x_1 = h_1 = \underbrace{X_{2n-1} \cdots X_1 X_0}_{2n \text{ bits}} \quad (61)$$

$$x_2 = h_2 \quad (62)$$

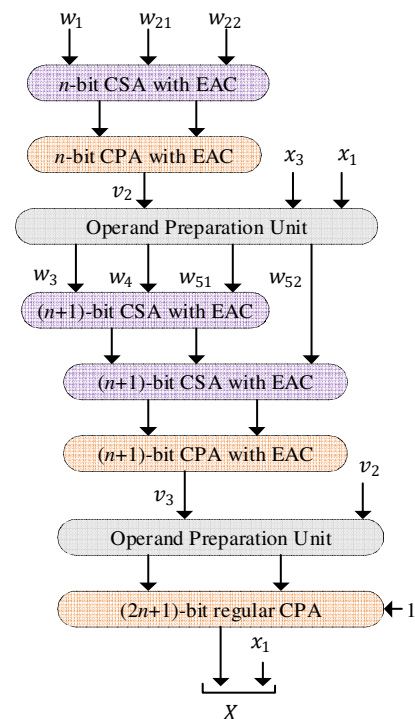
$$x_3 = h_3 \quad (63)$$

که ضرایب  $h_2$  و  $h_3$  این‌گونه محاسبه می‌شوند:

$$h_2 = \begin{cases} |w_1 + w_{21} + w_{22}|_{2^n-1} & \text{if Control} = 0 \\ |f_0 + f_1 + f_2 + f_3 + f_4|_{2^n-1} & \text{if Control} = 1 \end{cases} \quad (64)$$

$$h_3 = \begin{cases} |w_3 + w_4 + w_{51} + w_{52}|_{2^{n+1}-1} & \text{if Control} = 0 \\ |g_0 + g_1 + g_2 + g_3|_{2^{n+1}-1} & \text{if Control} = 1 \end{cases} \quad (65)$$

لازم به ذکر است که نحوه محاسبه و تنظیم بیت‌های بردارهای مورد نیاز در (۶۴) و (۶۵) در (۴۰) تا (۴۵) و (۵۳) تا (۵۹) مشخص شده است. حال باتوجه به (۶۰) تا (۶۵)، معماری سخت‌افزاری مبدا ترکیبی مستقیم/معکوس پیشنهادی در شکل ۵ نشان داده شده است. ورودی و خروجی‌های این واحد ترکیبی در جدول ۱ مشخص شده است.



شکل ۴: مبدا معکوس برای مجموعه پیمان { $2^{2n}, 2^{2n}-1, 2^{n+1}-1$ } [۱۹]



$$x_1 = |X|_{2^{2n}} = |120|_{16} = |(001111000)_2|_{2^4} = (1000)_2 = 8 \quad (۶۷)$$

$$x_2 = |X|_{2^{n-1}} = |120|_3 = |(001111000)_2|_{2^{2-1}} = |(00)_2 + (01)_2 + (11)_2 + (10)_2 + (00)_2|_{2^{2-1}} = |6|_3 = 0 \quad (۶۸)$$

$$x_3 = |X|_{2^{n+1-1}} = |120|_7 = |(001111000)_2|_{2^{3-1}} = |(000)_2 + (001)_2 + (111)_2 + (000)_2|_{2^{3-1}} = |8|_7 = 1 \quad (۶۹)$$

به عبارتی (۱، ۰، ۸) نمایش مانده‌های عدد ۱۲۰ است. همچنین، تبدیل معکوس عدد مانده‌ای (۵، ۲، ۹) بر حسب مجموعه پیمانه  $\{۱۶، ۳، ۷\}$  و مبتنی بر (۵۰) تا (۵۹) به صورت زیر انجام می‌شود:

$$x_1 = 9 = (1001)_2 \quad (۷۰)$$

$$x_2 = 2 = (10)_2 \quad (۷۱)$$

$$x_3 = 5 = (101)_2 \quad (۷۲)$$

$$w_1 = \underbrace{x_{2,n-1} \cdots x_{2,1} x_{2,0}}_{n \text{ bits}} = (10)_2 \quad (۷۳)$$

$$w_{21} = \underbrace{\bar{x}_{1,n-1} \cdots \bar{x}_{1,1} \bar{x}_{1,0}}_{n \text{ bits}} = (10)_2 \quad (۷۴)$$

$$w_{22} = \underbrace{\bar{x}_{1,2n-1} \cdots \bar{x}_{1,n+1} \bar{x}_{1,n}}_{n \text{ bits}} = (01)_2 \quad (۷۵)$$

$$v_2 = |w_1 + w_{21} + w_{22}|_{2^{n-1}} = |2 + 2 + 1|_3 = 2 \quad (۷۶)$$

$$w_3 = \underbrace{v_{2,n-1} \cdots v_{2,1} v_{2,0}}_{n \text{ bits}} 0 = (100)_2 = 4 \quad (۷۷)$$

$$w_4 = \underbrace{\bar{x}_{3,n-3} \cdots \bar{x}_{3,1} \bar{x}_{3,0} \bar{x}_{3,n} \bar{x}_{3,n-1} \bar{x}_{3,n-2}}_{n-2 \text{ bits}} = (010)_2 = 2 \quad (۷۸)$$

$$w_{51} = \underbrace{x_{1,n-3} \cdots x_{1,1} x_{1,0} x_{1,n} x_{1,n-1} x_{1,n-2}}_{n-2 \text{ bits}} = (001)_2 = 1 \quad (۷۹)$$

$$w_{52} = \underbrace{x_{1,2n-2} \cdots x_{1,n+2} x_{1,n+1}}_{n-2 \text{ bits}} 00 x_{1,2n-1} = (001)_2 = 1 \quad (۸۰)$$

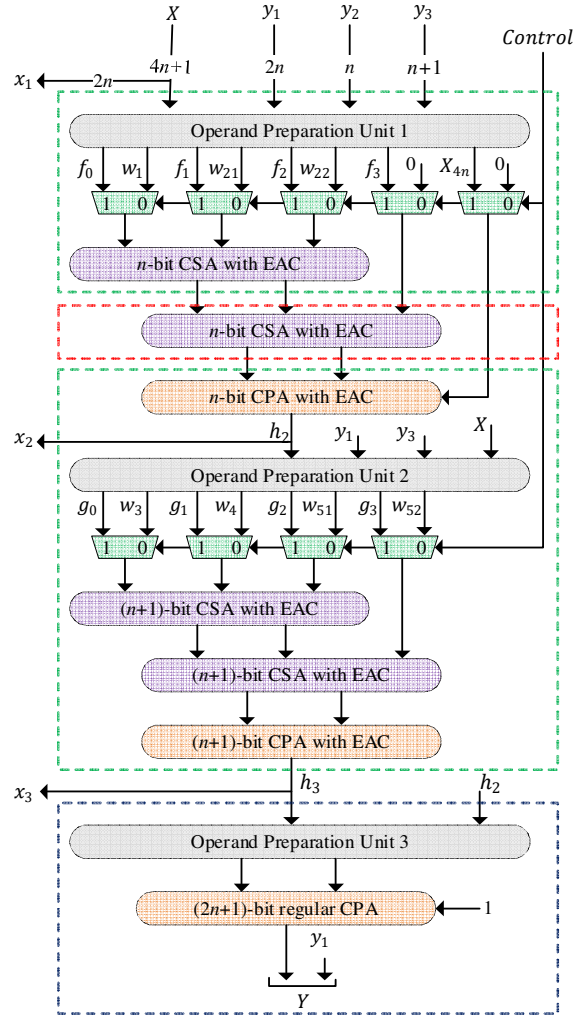
$$v_3 = |w_3 + w_4 + w_{51} + w_{52}|_{2^{n+1-1}} = |4 + 2 + 1 + 1|_7 = 1 \quad (۸۱)$$

$$Y = 2^{2n}(v_2 + 2^n v_3 - v_3) + x_1 = (v_3 \& v_2 - v_3) \& x_1 = (00110 - 001)_2 \& (1001)_2 = (001011001)_2 = 89 \quad (۸۲)$$

به عبارتی ۸۹ برابر با حاصل تبدیل معکوس عدد مانده‌ای (۹، ۲، ۵) است. لازم به ذکر است & بیان‌گر عمل‌گر الحاق بیتی است. در نهایت، توسط مبدل ترکیبی پیشنهادی بر حسب سیگنال کنترلی *Control*، یا عدد وزنی ۱۲۰ به باقی‌مانده‌هایش (تبدیل مستقیم) و یا عدد مانده‌ای (۵، ۲، ۹) به معادل وزنی‌اش (تبدیل معکوس) تبدیل می‌شود. از این‌رو، بر پایه روابط (۶۰) تا (۶۵) و با در نظر گرفتن (۶۸) و (۶۹)، و نیز (۷۶) و (۸۱)، رابطه‌های (۸۳) و (۸۴) به دست می‌آیند.

$$h_2 = \begin{cases} |w_1 + w_{21} + w_{22}|_{2^{n-1}} = 2 & \text{if } Control = 0 \\ |f_0 + f_1 + f_2 + f_3 + f_4|_{2^{n-1}} = 0 & \text{if } Control = 1 \end{cases} \quad (۸۳)$$

$$h_3 = \begin{cases} |w_3 + w_4 + w_{51} + w_{52}|_{2^{n+1-1}} = 1 & \text{if } Control = 0 \\ |g_0 + g_1 + g_2 + g_3|_{2^{n+1-1}} = 1 & \text{if } Control = 1 \end{cases} \quad (۸۴)$$



شکل ۵: مبدل ترکیبی مستقیم/معکوس پیشنهادی برای مجموعه پیمانه  $\{2^{2n}, 2^{n-1}, 2^{n+1}-1\}$

مثال ۲: مجموعه پیمانه  $\{2^{2n}, 2^{n-1}, 2^{n+1}-1\}$  به‌ازای  $n=2$

به صورت  $\{۱۶، ۳، ۷\}$  در می‌آید. محدوده دینامیکی این مجموعه پیمانه ۹ بیتی است و می‌توان اعداد ۰ تا ۳۳۵ را وارد به سیستم یا از آن خارج کرد. حال فرض کنید که عدد وزنی  $X=۱۲۰$  وارد سیستم شود و باید توسط مبدل مستقیم به باقی‌مانده‌های متناظر یعنی  $(x_1, x_2, x_3)$  تبدیل شود. همچنین فرض کنید عدد مانده‌ای (۹، ۲، ۵) وارد مبدل معکوس شود و این مبدل باید معادل وزنی آن، یعنی  $Y$  را به دست بیاورد. در ابتدا نحوه انجام این اعمال را توسط مبدل‌های مستقیم و معکوس عادی شرح داده می‌شود و سپس نحوه انجام آن را توسط مبدل ترکیبی پیشنهادی بررسی خواهد شد.

تبدیل مستقیم عدد  $X=۱۲۰$  بر حسب مجموعه پیمانه  $\{۱۶، ۳، ۷\}$  و مبتنی بر (۳۹) تا (۴۵) به صورت زیر انجام می‌شود:

$$X = 120 = (001111000)_2 \quad (۶۶)$$

شده است که رقم نقلی انتهایی، مجدد وارد می شود تا رقم نقلی را با عملوندها جمع کند تا جمع پیمانه ای انجام شود. به عبارتی جمع در دو مرحله انجام می شود. لذا سخت افزار مورد نیاز و نیز تأخیر آن معادل سطح و تأخیر  $k$  و  $2k$  تمام جمع کننده، به ترتیب، در نظر گرفته می شود. علاوه بر این، سخت افزار مورد نیاز برای جمع کننده ذخیره رقم نقلی  $k$  بیته معادل  $k$  تمام جمع کننده و تأخیر آن برابر یک تمام جمع کننده در نظر گرفته می شود [۱۳]. از طرفی دیگر، تمام جمع کننده های با یک ورودی بیته ثابت می توانند به نیم جمع کننده ها ساده شوند، ولی برای مقایسه یکسان بین همه طراحی ها، هیچ گونه ساده سازی تمام جمع کننده ها در جدول ۲ در نظر گرفته نشده است. در مجموع، طراحی پیشنهادی در شکل ۵، نیاز به  $8n+4$  تمام جمع کننده و ۹ مالتی پلکسر ۲ به ۱ دارد که ۴ تای آن ها  $n$  بیته، ۴ تای دیگر  $n+1$  بیته و یکی هم ۱ بیته است. بنابراین  $8n+5$  مالتی پلکسر ۲ به ۱ نیاز است. طبق مدل گیت واحد [۱]، هر تمام جمع کننده، سخت افزار معادل ۷ گیت و هر مالتی پلکسر ۲ به ۱ تک بیته، سخت افزار معادل ۳ گیت واحد دارد. لذا برای ساده تر شدن مقایسه، می توان هر مالتی پلکسر را معادل  $0.42$  یک تمام جمع کننده در نظر گرفت. بنابراین در مجموع، سخت افزار مورد نیاز مدار مبدل ترکیبی پیشنهادی می تواند معادل  $11.4n+6.14$  تا تمام جمع کننده در نظر گرفته شده است.

همان طور که در جدول ۲ مشاهده می شود، مبدل ترکیبی مستقیم/معکوس پیشنهادی نسبت به حالت داشتن هر دو مبدل به طور هم زمان در سیستم اعداد مانده ای، سخت افزار کم تری نیاز دارد زیرا تمام جمع کننده های پیمانه ای مورد نیاز در هر دو مبدل، در ساختار ترکیبی به صورت اشتراکی استفاده می شود. در مقایسه با مبدل معکوس عادی [۱۹]، فقط یک جمع کننده ذخیره رقم نقلی و تعدادی مالتی پلکسر در مدار ترکیبی اضافه شده است. علاوه بر این، در حالت کاری تبدیل مستقیم، محاسبه باقی مانده های خروجی یعنی  $h_2$  و  $h_3$  کاملاً موازی انجام می شود، زیرا (۴۰) و (۴۳) مستقل از یکدیگر هستند و فقط مالتی پلکسرها نسبت به مبدل مستقیم عادی به مسیر بحرانی تأخیر اضافه شده است. لذا تأخیر مدار ترکیبی پیشنهادی در حالت تبدیل مستقیم،  $T_{MUX} + T_{FA} (2n+4)$  است که در جدول ۲، تأخیر بدترین حالت، یعنی حالت کاری تبدیل معکوس نوشته شده است.

جهت ارزیابی دقیق تأخیر، سطح تراشه و توان مصرفی مبدل های مختلف، همه طراحی ها توسط کدهای VHDL شرح داده شدند و بعد از آزمایش درستی توسط ابزار Modelsim، توسط تکنولوژی TSMC-65nm و ابزارهای Cadence RTL compiler و Cadence Encounter پیاده سازی شدند که نتایج حاصل در جدول ۳ نشان داده شده اند. لازم به ذکر است، چون در حالت کاری مبدل معکوس، مدار محاسبات بیش تری باید انجام دهد، توان مصرفی آن بیشتر از حالت کاری مبدل مستقیم می شود. لذا در جدول ۳، توان مصرفی در بدترین حالت، یعنی در حالت کاری مبدل معکوس درج شده است. طبق جدول ۳، به ازای  $n=8$  مبدل ترکیبی پیشنهادی منجر به حداکثر ۱۹ درصد کاهش سطح تراشه در مقایسه با

حال اگر سیگنال کنترلی برابر با یک باشد، مدار در حالت عمل کردی تبدیل مستقیم قرار می گیرد و باقی مانده ها به صورت زیر محاسبه می شوند:

جدول ۲: سخت افزار مورد نیاز و تأخیر مبدل های مختلف

تأخیر	سخت افزار	مدار
$(6n+6) T_{FA}$	$(7n+4) A_{FA}$	مبدل معکوس [۱۹]
$(2n+4) T_{FA}$	$(6n+3) A_{FA}$	مبدل مستقیم
$(6n+6) T_{FA}$	$(13n+7) A_{FA}$	مجموع مبدل های مستقیم و معکوس
$(6n+7) T_{FA} + 2 T_{MUX}$	$(11.4n+6.14) A_{FA}$	مبدل ترکیبی مستقیم/معکوس پیشنهادی

$$x_1 = h_1 = \underbrace{X_{2n-1} \cdots X_1 X_0}_{2n \text{ bits}} = (1000)_2 = 8 \quad (85)$$

$$x_2 = h_2 = 0 \quad (86)$$

$$x_3 = h_3 = 1 \quad (87)$$

در صورتی که سیگنال کنترلی برابر با صفر باشد، مدار در حالت عمل کردی تبدیل معکوس قرار می گیرد و عدد وزنی به صورت زیر محاسبه می شود:

$$\begin{aligned} Y &= 2^{2n} (2^n - 1) h_3 + 2^{2n} h_2 + h_1 \\ &= 2^{2n} (h_2 + 2^n h_3 - h_3) + x_1 = (h_3 \& h_2 - h_3) \& x_1 \\ &= (00110 - 001)_2 \& (1001) = (001011001)_2 = 89 \end{aligned} \quad (88)$$

## ۶- ارزیابی کارایی

باتوجه به این که برای اولین بار است یک طراحی ترکیبی از مبدل مستقیم و معکوس ارائه می شود، برای ارزیابی کارایی آن، می توان مقایسه را با طراحی های مستقل مبدل مستقیم و معکوس انجام داد. از این رو، مدارهای مربوط به مبدل مستقیم (شکل ۳)، مبدل معکوس (شکل ۴) و مبدل ترکیبی مستقیم/معکوس پیشنهادی (شکل ۵) مبتنی بر مجموعه پیمانه  $\{2^{2n}, 2^n - 1, 2^{n+1} - 1\}$ ، از لحاظ کارایی، مقایسه خواهند شد. در ابتدا تحلیل مستقل از تکنولوژی و بر پایه گیت های منطقی ارائه می شود و سپس ارزیابی مبتنی بر نتایج تجربی پیاده سازی VLSI و نیز FPGA، انجام خواهد شد.

جدول ۲ سخت افزار مورد نیاز و تأخیر مدارهای ذکر شده را بر حسب تعداد تمام جمع کننده ها مقایسه می کند. لازم به ذکر است  $A_{FA}$  و  $T_{FA}$  به ترتیب نشان دهنده سطح و تأخیر یک تمام جمع کننده است. همچنین مشابه [۱۹]، جمع کننده انتشار رقم نقلی با چرخش رقم نقلی انتهایی  $k$  بیته مانند یک جمع کننده انتشار رقم نقلی عادی در نظر گرفته

## سياست گزاري

نويسندگان مقاله تمايل دارند از آقاي لئونل سوسا و آزمايشگاه پردازش سيگنالهاي ديجيتال دانشگاه ليسبون کشور پرتغال جهت همکاري در پياده سازي VLSI مبدلها، تشکر کنند.

## پيوستها

در اين بخش RTL Schematic هاي مربوط به مبدل ترکيبی پيشنهادی

جدول ۳: پارامترهاي مداري حاصل از پياده سازي VLSI مبدلهاي

### مختلف

مدار	$n$	سطح تراشه ( $\mu\text{m}^2$ )	تأخير (ns)	توان مصرفي (mW)
مبدل معکوس [۱۹]	۴	۲۷۵۶	۱/۱۴۶	۱/۸۸۷
	۸	۴۷۳۳	۱/۸۱۰	۲/۲۷۴
	۱۶	۸۳۲۸	۳/۱۱۷	۲/۴۹۲
مبدل مستقيم	۴	۲۱۹۰	۰/۸۴۰	۰/۹۱۵
	۸	۴۱۹۹	۰/۸۸۶	۲/۵۱۱
	۱۶	۷۷۰۹	۱/۴۵۲	۳/۰۱۱
مجموع مبدلهاي مستقيم و معکوس	۴	۴۹۴۶	۱/۱۴۶	۲/۸۰۲
	۸	۸۹۳۲	۱/۸۱۰	۴/۷۸۵
	۱۶	۱۶۰۳۷	۳/۱۱۷	۵/۵۰۳
مبدل ترکيبی مستقيم/معکوس پيشنهادی	۴	۳۹۸۲	۱/۲۸۵	۲/۳۵۷
	۸	۷۱۹۳	۱/۹۹۴	۳/۱۷۲
	۱۶	۱۳۳۸۵	۳/۲۵۹	۳/۸۲۷

برای مجموعه پيمانه  $\{2^{2n}, 2^n-1, 2^{n+1}-1\}$  به ازای  $n=4$  که از ابزار ISE به دست آمده است، رسم شده است. شکل ۶ بلوک دياگرام کلی و ورودی/خروجیهاي مبدل ترکيبی پيشنهادی را نشان می دهد. همان طور که مشاهده می شود، ورودیهاي مدار شامل سيگنال  $Control$  (یک بيتی)، باقي مانده هاي  $x_1, x_2, x_3$  و عدد وزنی  $Y$  می باشند. هم چنین، خروجیهاي مدار عدد وزنی  $X$  و نیز باقي مانده هاي  $y_1, y_2, y_3$  می باشند. باتوجه به اين که  $n=4$  است، باقي مانده هاي متناظر با پيمانه هاي  $2^{2n}-1$  و  $2^{n+1}-1$  به ترتيب ۸، ۴ و ۵ بيتی و نیز اعداد وزنی ورودی و خروجی ۱۷ بيتی می باشند.



شکل ۶: بلوک دياگرام کلی مبدل ترکيبی مستقيم/معکوس

پيشنهادی برای مجموعه پيمانه  $\{2^{2n}, 2^n-1, 2^{n+1}-1\}$  به ازای  $n=4$

داشتن هر دو مبدل مستقيم و معکوس به طور مستقل، شده است. علاوه بر اين، تأخير مدار ترکيبی مبدل مستقيم/معکوس پيشنهادی نسبت به مبدل معکوس مستقل، کمی بيش تر شده است، زیرا مالتی پلکسرها و یک جمع کننده ذخيره رقم نقلی به مسير بحرانی تأخير مبدل اضافه شده است. به خصوص هر چه  $n$  افزايش پيدا می کند، تفاضل تأخير مبدل ترکيبی پيشنهادی و مبدل معکوس مستقل، کم تر می شود (۱۰ درصد و ۴ درصد به ازای  $n=4$  و  $n=16$  به ترتيب).

علاوه بر پياده سازي ASIC مدارها، تمامی طراحیهاي مبتنی بر ابزار ISE بروی تراشه FPGA Virtex5 مدل XC5VLX85 پياده سازي شدند که نتايج حاصل در جدول ۴ نشان داده شده است. همان گونه که مشاهده می شود، مبدل ترکيبی پيشنهادی نسبت به مجموع مبدلهاي مستقيم و معکوس نیاز به تعداد LUT هاي کمتر ولی تأخير بيشتر دارد. هم چنین اگر چه، تأخير طرح پيشنهادی از مبدلهاي مستقيم و معکوس عادی بيش تر است، ولی درصد کاهش سخت افزار نسبت به حالت مجموع هر دو مبدل، چشمگیر است. به عنوان مثال، به ازای مقدار ۱۶ برای  $n$  طرح پيشنهادی، ۲۰ درصد سخت افزار (تعداد LUT هاي مورد نیاز) را کاهش داده، ولی فقط ۱۰ درصد تأخير را در مقايسه با مجموع مبدل مستقيم/معکوس افزايش داده است.

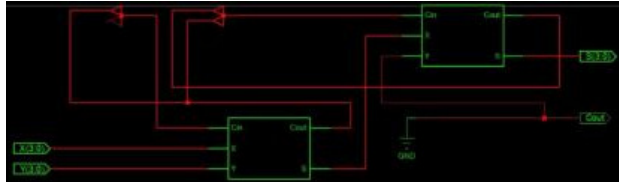
جدول ۴: پارامترهاي مداري حاصل از پياده سازي FPGA مبدلهاي

### مختلف

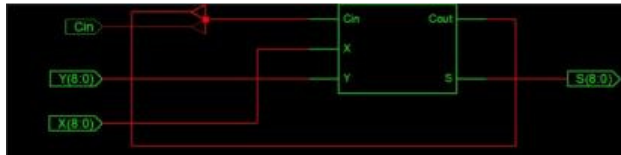
مدار	$n$	حجم مصرفي (تعداد LUT ها)	تأخير (ns)
مبدل معکوس [۱۹]	۴	۶۰	۵/۹۴۰
	۸	۱۶۱	۷/۷۳۲
	۱۶	۴۱۳	۹/۴۸۲
مبدل مستقيم	۴	۴۹	۳
	۸	۱۰۳	۴/۰۷۴
	۱۶	۲۶۳	۴/۹۷۶
مجموع مبدلهاي مستقيم و معکوس	۴	۱۰۹	۵/۹۴۰
	۸	۲۶۴	۷/۷۳۲
	۱۶	۶۷۶	۹/۴۸۲
مبدل ترکيبی مستقيم/معکوس پيشنهادی	۴	۱۰۱	۶/۹۳۸
	۸	۲۳۵	۹/۳۸۲
	۱۶	۵۳۵	۱۰/۵۵۹

## ۷- نتیجه

در اين مقاله، برای اولین بار، یک طراحی ترکيبی از مبدلهاي مستقيم و معکوس ارائه شد که توانست سربار بالای ناشی از سخت افزار مورد نیاز برای مبدلهاي مستقيم و معکوس مستقل را کاهش دهد. لذا با داشتن چنین مبدل ترکيبی، پيچيدگی سخت افزاری کلی سيستم اعداد مانده ای کاهش می يابد و امکان استفاده از آن در کاربردهايی که نیاز به سخت افزار کم دارند مانند سيستمهاي نهفته فراهم می آيد.



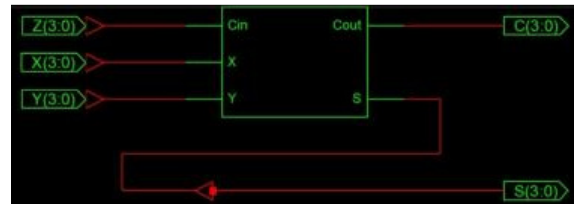
شکل ۹: ساختار کلی جمع کننده انتشار رقم نقلی پیمانهای



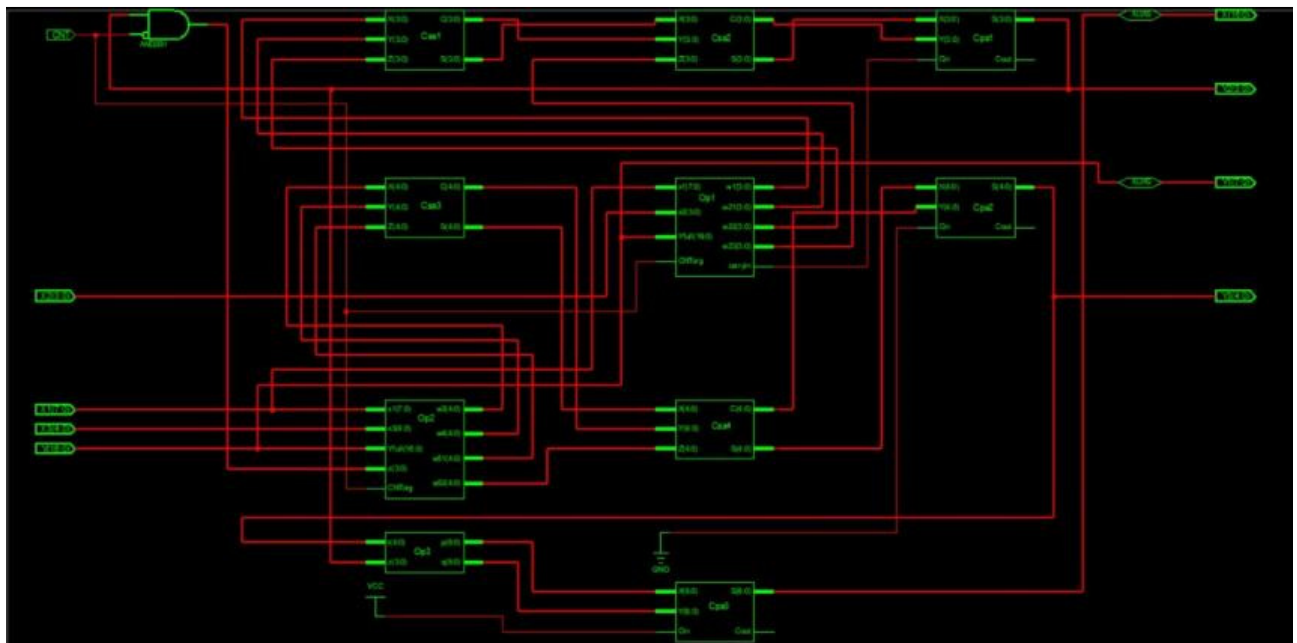
شکل ۱۰: ساختار کلی جمع کننده انتشار رقم نقلی عادی

همچنین در شکل ۱۱، معماری کلی واحد آماده سازی عملوندهای (OPU1) نشان داده شده است. در واحدهای OPU اول و دوم، مالتی پلکسرها قرار دارند که ورودی‌های مناسب جهت جمع کننده‌های ذخیره رقم نقلی را فراهم می‌کنند. در OPU سوم، از طریق سیم بندی و تعدادی گیت معکوس کننده، ورودی‌های مناسب جمع کننده انتشار رقم نقلی انتهایی مدار آماده می‌شوند.

در شکل ۷ ساختار داخلی مبدل ترکیبی نشان داده شده است که از جمع کننده‌های ذخیره رقم نقلی (CSA ها)، جمع کننده‌های انتشار رقم نقلی (CPA ها) و واحدهای آماده سازی عملوندها (OPU ها) تشکیل شده است. در شکل ۸ بلوک دیاگرام جمع کننده ذخیره رقم نقلی ۱، و در شکل ۹ ساختار جمع کننده انتشار رقم نقلی پیمانهای که رقم نقلی خروجی را مجدداً با حاصل جمع مرحله اول جمع می‌کند، نشان داده شده است. همچنین ساختار جمع کننده انتشار رقم نقلی عادی که در انتهای مبدل نیاز است، در شکل ۱۰ رسم شده است.

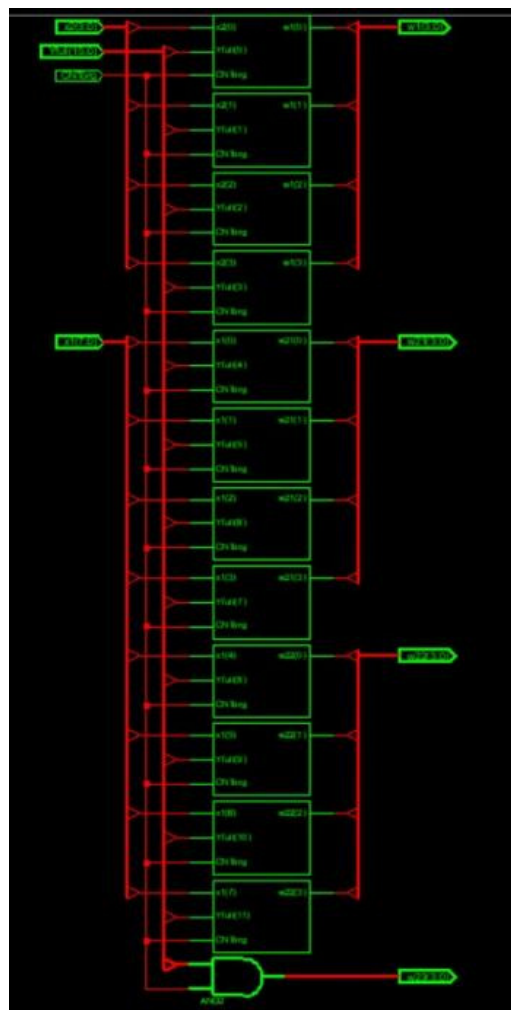


شکل ۸: بلوک دیاگرام کلی جمع کننده ذخیره رقم نقلی



شکل ۷: ساختار مبدل ترکیبی مستقیم/معکوس پیشنهادی برای مجموعه پیمانهای  $\{2^{2n}, 2^n - 1, 2^{n+1} - 1\}$  به ازای  $n=4$

- Circuits and Systems Magazine, vol. 16, no. 4, pp. 6-32, Nov. 2016.
- [7] V. Arrigoni, B. Rossi, P. Fragneto and G. Desoli, "Approximate operations in convolutional neural networks with RNS data representation," In Proc. of 25th European Symposium on Artificial Neural Networks, Bruges, Belgium, Apr. 26-28 2017.
- [8] K. Navi, A.S. Molahosseini and M. Esmaildoust, "How to teach residue number system to computer scientists and engineers," IEEE Transactions on Education, vol. 54, no. 1, pp. 156-163, Feb. 2011.
- [9] S. J. Piestrak, "Design of residue generators and multioperand modular adders using carry-save adders," IEEE Transactions on Computers, vol. 43, no. 1, pp. 68-77, Jan. 1994.
- [10] A. A. Hiasat, "Arithmetic binary to residue encoders for moduli  $(2n \pm 2k + 1)$ ," IEEE Proceedings - Computers and Digital Techniques, vol. 150, no. 6, pp. 369-374, Nov. 2003.
- [11] C. Efstathiou, N. Moschopoulos, K. Tsoumanis and K. Pekmezzi, "On the design of configurable modulo  $2n \pm 1$  residue generators," In Proc. of 15th Euromicro Conference on Digital System Design, Izmir, Turkey, pp. 50-56, 2012.
- [12] P. M. Matutino, R. Chaves and L. Sousa, "Arithmetic-based binary-to-RNS converter modulo  $\{2n \pm k\}$   $j$ n-bit dynamic range," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 23, no. 3, pp. 603-607, Mar. 2015.
- [13] S.J. Piestrak, "A high speed realization of a residue to binary converter," IEEE Transactions on Circuits and Systems-II, vol. 42, no. 10, pp. 661-663, Oct. 1995.
- [14] Y. Wang, X. Song, M. Aboulhamid and H. Shen, "Adder based residue to binary numbers converters for  $(2n-1, 2n, 2n+1)$ ," IEEE Transactions Signal Processing, vol. 50, no. 7, pp. 1772-1779, Jul. 2002.
- [15] B. Cao, C. H. Chang and T. Srikanthan, "An efficient reverse converter for the 4-moduli set  $\{2n-1, 2n, 2n+1, 22n+1\}$  based on the new Chinese remainder theorem," IEEE Transaction on Circuits and Systems- I, vol. 50, no. 10, pp. 1296-1303, Oct. 2003.
- [16] A. Hariri, K. Navi, and R. Rastegar, "A new high dynamic range moduli set with efficient reverse converter," Journal of Computers and Mathematics with Applications, vol. 55, no. 4, pp. 660-668, Feb. 2008.
- [17] A.S. Molahosseini, K. Navi, C. Dadkhah, O. Kavehei, S. Timarchi, "Efficient reverse converter designs for the new 4-moduli sets  $\{2n-1, 2n, 2n+1, 22n+1-1\}$  and  $\{2n-1, 2n+1, 22n, 22n+1\}$  based on new CRTs," IEEE Transactions on Circuits and Systems-I, vol. 57, no. 4, pp. 823-835, Apr. 2010.
- [18] L. Sousa and S. Antao, "MRC-based RNS reverse converters for the four-moduli sets  $\{2n+1, 2n-1, 2n, 22n+1-1\}$  and  $\{2n+1, 2n-1, 22n, 22n+1-1\}$ ," IEEE Transactions on Circuits and Systems II, vol. 59, no. 4, pp. 244-248, Apr. 2012.
- [19] A.S. Molahosseini, K. Navi, C. Dadkhah, M. Eshghi, "Efficient MRC-based residue to binary converters for the new moduli sets  $\{22n, 2n-1, 2n+1-1\}$  and  $\{22n, 2n-1, 2n-1-1\}$ ," IEICE Transactions on Information and Systems, vol. 92, no. 9, pp. 1628-1638, Sep. 2009.
- [20] A.A.E. Zarandi, A.S. Molahosseini, M. Hosseinzadeh, S. Sorouri, S.F. Antão and L. Sousa, "Reverse converter design via parallel-prefix adders: novel components, methodology and implementations," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 2, no. 374-378, p. 23, Feb. 2015.
- [21] A.A.E. Zarandi, A.S. Molahosseini, L. Sousa and M. Hosseinzadeh, "An efficient component for designing signed reverse converters for a class of RNS moduli sets with composite form  $\{2K, 2P-1\}$ ," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 25, no. 1, pp. 48-59, Jan. 2017.
- [22] A. S. Molahosseini, A. A. E. Zarandi, P. Martins and L. Sousa, "A Multifunctional Unit for Designing Efficient RNS-Based Datapaths," IEEE Access, vol. 5, pp. 25972-25986, Dec. 2017.
- [23] P.V.A. Mohan, Residue Number Systems: Theory and Applications, Springer International Publishing, 2016.



شکل ۱: بلوک دیاگرام کلی واحد آماده‌سازی عمل‌وندها ۱

## مراجع

- [1] A. S. Molahosseini, L. Sousa and C.H. Chang (Eds.), Embedded Systems Design with Special Arithmetic and Number Systems, Springer International Publishing, 2017.
- [2] امیر سزاوار، حسن فرسی، سجاد محمدزاده، «بازیابی تصویر مبتنی بر محتوا با استفاده از شبکه‌های عصبی کانولوشن عمیق»، مجله مهندسی برق دانشگاه تبریز، دوره ۴۸، شماره ۴، صفحه ۱۵۹۵-۱۶۰۳، زمستان ۱۳۹۷.
- [3] پرهام درّی، علی قیاسیان، حسین سعیدی، «طراحی و پیاده‌سازی رمزنگار AES در بستر FPGA برای خطوط پرسرعت»، مجله مهندسی برق دانشگاه تبریز، دوره ۴۶، شماره ۱، صفحه ۱۵۳-۱۶۷، بهار ۱۳۹۵.
- [4] H.L. Garner, "The residue number system," IRE Transactions on Electronic Computers, vol. 8, no. 2, pp. 140-147, Jun. 1959.
- [5] C.H. Chang, A.S. Molahosseini, A.A. Emrani Zarandi and T.F. Tay, "Residue number systems: a new paradigm to datapath optimization for low-power and high-performance digital signal processing applications," IEEE Circuits and Systems Magazine, vol. 15, no. 4, pp. 26-44, Nov. 2015.
- [6] L. Sousa, S. Antão and P. Martins, "Combining residue arithmetic to design efficient cryptographic circuits and systems," IEEE

## زيرنويس ها

---

- <sup>1</sup> Residue Number System (RNS)
- <sup>2</sup> Forward Converter
- <sup>3</sup> Moduli Set
- <sup>4</sup> Reverse Converter
- <sup>5</sup> Carry-Save Adder (CSA)
- <sup>6</sup> Modulo Adder
- <sup>7</sup> Mixed-Radix Conversion (MRC)