Archive of SID

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

آزاده السادات عمرانی زرندی[۱]، استادیار؛ امیر صباغ ملاحسینی[۲]، 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 شده‌است 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 شده‌است. نتایج حاصل از پیاده‌سازی VLSI مبدل ترکیبی پیشنهادی مبتنی بر تکنولوژی TSMC-65nm، برای مجموعه پیمانه $\{2^n\text{-}1, 2^{2n}, 2^{n+1}\text{-}1\}$، 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 شده‌است.

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[1], Assistant professor; Amir Sabbagh Molahosseini[2], 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 $\{2^n-1, 2^{2n}, 2^{n+1}-1\}$ 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 زرندی
نشانی نویسنده مسئول: گروه مهندسی کامپیوتر- دانشگاه شهید باهنر کرمان- کرمان- ایران.

www.SID.ir

Archive of SID

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. به‌عبارتی 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 کارامد سیستم‌های نهفته جهت کاهش انرژی مصرفی محاسبات موردتوجه محققین قرار گرفته‌است [۱]. سیستم اعداد مانده‌ای[۱] [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]. ازاین‌رو، سیستم اعداد 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 سیستم اعداد مانده‌ای ابتدا توسط مبدل مستقیم[۲]، اعداد وزنی دودوئی عادی بر تعدادی عدد دو به دو نسبت به هم اول که مجموعه پیمانه[۳] 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 به‌صورت پیمانه‌ای برروی نمایش مانده‌ای عمل‌وندها انجام می‌شوند. مزیت اصلی سیستم اعداد مانده‌ای دراین است که مابین مدارهای حسابی که به‌ازای هر پیمانه وجود دارند، انتشار رقم نقلی وجود ندارد و فقط درون مدارهای پیمانه‌ای انتشار محدود رقم نقلی وجود دارد. ازاین‌رو، امکان پیاده‌سازی موازی اعمال ضرب و جمع پیمانه‌ای برروی باقی‌مانده‌ها به وجود می‌آید که منجر به بالارفتن سرعت و کاهش توان مصرفی می‌شود. هم‌چنین، امکان تحمل‌پذیری خطا درصورت استفاده از پیمانه افزونه، فراهم می‌شود. درانتها نتیجه محاسبه، توسط مبدل معکوس[۴] 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 می باشند و اگر ناکارامد و با سخت افزار و تأخیر زیاد طراحی شوند، منجر به کاهش بهره سرعت ناشی از موازی‌سازی مدارهای حسابی پیمانه‌ای داخل سیستم می‌شود. لذا تحقیق بسیار زیادی برروی طراحی کارامد 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[۵]، 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[۶] 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 پیمانه است. به‌همین‌دلیل، محققان از جنبه‌های مختلف، طراحی‌های متفاوتی برای مبدل معکوس ارائه داده‌اند تا کارایی این واحد محاسباتی که درواقع 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 کارامد برای مجموعه پیمانه‌های سه و چهار پیمانه‌ای مبتنی بر تئوری باقی‌مانده چینی استفاده شدند. همچنین، در [۱۷] از تئوری باقی مانده چینی جدید و نیز جمع‌کننده‌های ذخیره رقم نقلی جهت طراحی مبدل معکوس برای دو مجموعه پیمانه جدید چهار پیمانه‌ای استفاده شد. علاوه‌بر تئوری‌های باقی‌مانده چینی کلاسیک و جدید که در [۱۴-۱۷] مورد توجه قرار گرفته‌اند، تبدیل درهم مبنا[۷] نیز برای طراحی مبدل معکوس می‌تواند استفاده شود که نمونه‌هایی از استفاده از آن برای طراحی مبدل معکوس در [۱۸، ۱۹] ارائه شده است. تئوری‌های باقیمانده چینی ساختار موازی دارند ولی در انتها نیاز به یک جمع پیمانه‌ای بزرگ دارند ولی تبدیل درهم مبنا نیاز به چندین جمع پیمانه‌ای کوچک با ساختاری ترتیبی دارد. در مجموع، در تمامی این کارها شیوه رسیدن به مبدل معکوس بهتر بر پایه، ۱) انتخاب مجموعه پیمانه جدید، ۲) انتخاب الگوریتم تبدیل مناسب و نیز ۳) ساده‌سازی نوآورانه فرمول‌های تبدیل معکوس است.

از طرفی دیگر، اخیراً شیوه‌ای جدید برای بالابردن کارایی مبدل‌های معکوس در [۲۰-۲۲] مورد توجه قرار گرفته‌است. به‌این‌صورت که به‌جای تغییر مجموعه پیمانه 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 به‌این‌صورت 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 علامت‌دارکردن خروجی مبدل‌های معکوس طراحی شد. این سبک طراحی، در [۲۲] به کل سیستم اعداد مانده‌ای به‌جزء 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. ازطرفی فارغ از عدم اشتراک الگوریتمی، باتوجه‌به این‌که هر دو مبدل مستقیم و معکوس مبتنی‌بر جمع‌کننده‌های ذخیره رقم نقلی و جمع‌کننده پیمانه‌ای ساخته می‌شوند، این سؤال در ذهن تداعی می‌شود

www.SID.ir

Archive of SID

که آیا می‌توان دو واحد مجزا ولی با اجزای سخت‌افزاری یکسان را در هم ادغام کرد که عمل‌کرد هر دو واحد را داشته‌باشد ولی با سخت‌افزار کمتر؟

در این مقاله، برای اولین‌بار 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 شده‌است. مهم‌ترین مزیت طرح پیشنهادی، کاهش چشم‌گیر سربار 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 دودوئی)، مبتنی‌بر یک سخت‌افزار یکسان که توسط یک سیگنال کنترلی هدایت می‌شود، تولید می‌شوند. جهت نیل به این هدف، به شیوه‌ای نو، از الگوریتم تبدیل درهم مبنا استفاده شده‌است که روابط محاسباتی تبدیل معکوس به فرمت روابط محاسباتی تبدیل مستقیم درآید. سپس مبتنی‌بر تعدادی مالتی‌پلکسر از سخت‌افزار به‌صورت اشتراکی استفاده می‌شود؛ به این ترتیب که ورودی صحیح مبتنی‌بر 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 به‌جای دو مبدل مجزا، یک مبدل دو منظوره به کارگرفته شده‌است.

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

۲- سیستم اعداد مانده‌ای

پایه و اساس سیستم اعداد مانده‌ای، تعدادی عدد دو به دو نسبت به‌هم اول است که تشکیل مجموعه پیمانه سیستم را می‌دهند و به‌صورت $\{m_1, m_2, \ldots, m_n\}$ 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 مبتنی‌براین تعریف، خروجی مبدل معکوس نیز در این محدوده قرار می‌گیرد. محدوده دینامیکی توسط رابطه زیر محاسبه می‌شود:

$$M = m_1 \times m_2 \times \ldots \times m_n \qquad (۱)$$

به‌عبارتی حاصل‌ضرب پیمانه‌ها، محدوده اعداد قابل نمایش در سیستم را می‌سازد که به‌صورت $[0, M\text{-}1]$ نشان داده می‌شود [۲۳]. اعداد در این محدوده که به‌صورت عادی و وزنی هستند، توسط مبدل مستقیم بر اعداد درون مجموعه پیمانه تقسیم و سپس باقی‌مانده‌ها، نمایش مانده‌ای آن عدد وزنی را در سیستم اعداد مانده‌ای به‌صورت زیر تشکیل می‌دهند:

$$X \xrightarrow{RNS} (x_1, x_2, \cdots, x_n) \; where \; x_i = |X|_{m_i} \; for \; i = 1, \cdots, n \qquad (۲)$$

سپس اعمال حسابی جمع، تفریق یا ضرب برروی باقی‌مانده‌های اعدادی که وارد سیستم اعداد مانده‌ای شده‌اند، به‌صورت موازی و بدون انتشار رقم نقلی مابین ارقام مانده‌ای به‌صورت زیر انجام می‌شود:

$$X \otimes Y \xrightarrow{RNS} (|x_1 \otimes y_1|_{m_1}, |x_2 \otimes y_2|_{m_2}, \cdots, |x_n \otimes y_n|_{m_n}) \qquad (۳)$$

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

$$\otimes \in \{+, -, \times\} \qquad (۴)$$

سپس بعد از انجام محاسبات، حاصل که به‌صورت مجموعه‌ای از ارقام مانده‌ای است توسط یک الگوریتم تبدیل عکس مانند تئوری باقی‌مانده چینی یا تبدیل درهم مبنا به معادل وزنی تبدیل می‌شود [۸]. لازم به‌ذکر است که چندین عملیات حسابی برروی باقی‌مانده‌ها می‌تواند انجام شود. به‌عنوان مثال، در پیاده‌سازی فیلترهای دیجیتال موردنیاز در پردازش سیگنال دیجیتال مبتنی بر سیستم اعداد مانده‌ای [۱]، تعداد زیادی عملیات ضرب و جمع برروی باقی‌مانده‌ها انجام می‌شود و سپس حاصل نهایی فیلتر، توسط مبدل معکوس از سیستم خارج می‌شود. لازم به‌ذکر است هرچقدر تعداد محاسبات جمع و ضرب پیمانه‌ای درون سیستمی نسبت به تعداد انجام تبدیل مستقیم یا عکس بیش‌تر باشد، کارایی سیستم بالاتر خواهدرفت زیرا بهره سرعت ناشی از پیاده‌سازی موازی محاسبات جمع و ضرب، بر سربار ناشی از مبدل غالب خواهدشد.

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

$$X = v_n \prod_{i=1}^{n-1} m_i + \ldots + v_3 m_2 m_1 + v_2 m_1 + v_1 \qquad (۵)$$

رابطه (۵)، فرمول پایه‌ای الگوریتم تبدیل درهم‌مبنا است که ورودی‌های آن، باقی‌مانده‌ها و نیز مقدار پیمانه‌ها است و خروجی آن مقدار محاسبه‌شده عدد وزنی است. تعداد جمله‌های حاصل‌ضرب میانی، بستگی به‌تعداد پیمانه‌های سیستم دارد (درواقع مقدار n در این فرمول، بیان‌گر تعداد پیمانه‌ها است). به‌عنوان مثال، برای ۳ تا ۵ پیمانه، فرمول (۵) به‌صورت (۶) تا (۸)، به‌ترتیب، در می‌آید.

$$X = v_3 m_2 m_1 + v_2 m_1 + v_1 \qquad (۶)$$

$$X = v_4 m_3 m_2 m_1 + v_3 m_2 m_1 + v_2 m_1 + v_1 \qquad (۷)$$

$$X = v_5 m_4 m_3 m_2 m_1 + v_4 m_3 m_2 m_1 + v_3 m_2 m_1 + v_2 m_1 + v_1 \qquad (۸)$$

ضرایب v_i نیز به‌صورت زیر برحسب باقی‌مانده‌ها و نیز معکوس‌های ضربی به‌دست می‌آیند:

$$v_1 = x_1 \qquad (۹)$$

www.SID.ir

Archive of SID

$$v_2 = \left| (x_2 - v_1) \left| m_1^{-1} \right|_{m_2} \right|_{m_2} \quad (۱۰)$$

$$v_3 = \left| ((x_3 - v_1) \left| m_1^{-1} \right|_{m_3} - v_2) \left| m_2^{-1} \right|_{m_3} \right|_{m_3} \quad (۱۱)$$

$$v_4 = \left| \begin{matrix} (((x_4 - v_1) \left| m_1^{-1} \right|_{m_4} - v_2) \left| m_2^{-1} \right|_{m_4} \\ -v_3) \left| m_3^{-1} \right|_{m_4} \end{matrix} \right|_{m_4} \quad (۱۲)$$

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

$$v_n = \left| \begin{matrix} (((x_n - v_1) \left| m_1^{-1} \right|_{m_n} - v_2) \left| m_2^{-1} \right|_{m_n} - \cdots \\ -v_{n-1}) \left| m_{n-1}^{-1} \right|_{m_n} \end{matrix} \right|_{m_n} \quad (۱۳)$$

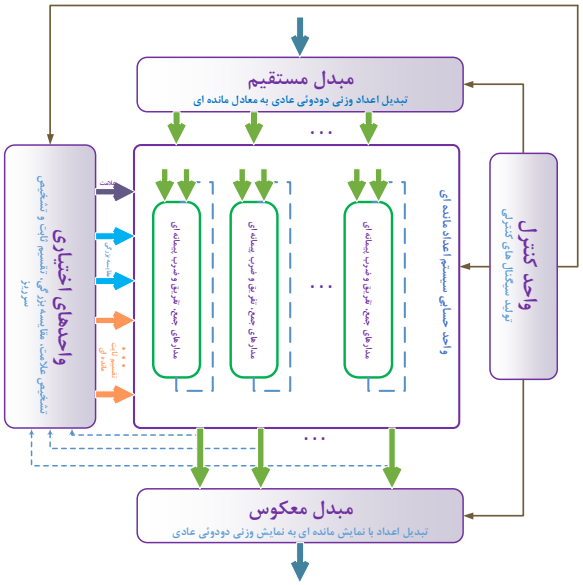
عبارت $\left| m_i^{-1} \right|_{m_j}$ نشان‌دهنده معکوس ضربی پیمانه m_i مبتنی بر پیمانه m_j است. معکوس‌های ضربی جداگانه محاسبه می‌شوند و در روابط بالا قرار می‌گیرند تا ضرایب الگوریتم تبدیل درهم مبنا به‌دست آیند و سپس با داشتن ضرایب، مقدار X از رابطه (۵) محاسبه می‌شود.

علاوه‌بر مبدل‌های مستقیم و معکوس، و نیز واحدهای حسابی جمع و ضرب، باتوجه‌به کاربرد هدف، واحدهای تشخیص علامت، مقایسه بزرگی، تقسیم ثابت و تشخیص سرریز نیز ممکن است در سیستم طراحی شوند. زیرا به‌دلیل خاصیت غیروزنی‌بودن سیستم اعداد مانده‌ای و برخلاف سیستم اعداد وزنی، امکان تشخیص علامت با یک بیت و یا مقایسه بزرگی ازطریق 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 $\{۵، ۴، ۳\}$ 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 برروی آن‌ها محاسبات موازی و سریع انجام داد. حال فرض کنید اعداد ۴ و ۱۱ توسط مبدل مستقیم مبتنی‌بر (۲) به‌صورت زیر وارد سیستم شوند (این اعداد بر تک تک اعداد درون مجموعه پیمانه تقسیم و سپس باقی‌مانده‌ها نگه داشته‌می‌شوند):

$$4 \xrightarrow{RNS} (|4|_3, |4|_4, |4|_5) \rightarrow (1,0,4) \quad (۱۴)$$

$$11 \xrightarrow{RNS} (|11|_3, |11|_4, |11|_5) \rightarrow (2,3,1) \quad (۱۵)$$



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 برروی این دو عدد که وارد سیستم شده‌اند به‌صورت زیر انجام می شود:

$$(1,0,4) \times (2,3,1) \xrightarrow{RNS} (|1 \times 2|_3, |0 \times 3|_4, |4 \times 1|_5) = (2,0,4) \quad (۱۶)$$

حال از الگوریتم تبدیل درهم مبنا 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 موردنیاز در (۱۰) و (۱۱) محاسبه شوند که مبتنی بر روش ارائه‌شده در [۸]، به‌صورت زیر به‌دست می‌آیند:

$$\left| m_1^{-1} \right|_{m_2} = \left| 3^{-1} \right|_4 = 3 \quad (|3 \times 3|_4 = 1) \quad (۱۷)$$

$$\left| m_1^{-1} \right|_{m_3} = \left| 3^{-1} \right|_5 = 2 \quad (|3 \times 2|_5 = 1) \quad (۱۸)$$

$$\left| m_2^{-1} \right|_{m_3} = \left| 4^{-1} \right|_5 = 4 \quad (|4 \times 4|_5 = 1) \quad (۱۹)$$

سپس مقدار باقی‌مانده‌ها و نیز معکوس‌های ضربی در (۹) تا (۱۱) قرار داده‌می‌شود تا مقدار ضرایب مورد نیاز در الگوریتم تبدیل درهم مبنا به‌دست آید.

$$v_1 = x_1 = 2 \quad (۲۰)$$

$$v_2 = \left| (x_2 - v_1) \left| m_1^{-1} \right|_{m_2} \right|_{m_2} = |(0-2) \times 3|_4 = 2 \quad (۲۱)$$

www.SID.ir

Archive of SID

$$v_3 = \left|((x_3 - v_1)\left|m_1^{-1}\right|_{m_3} - v_2)\left|m_2^{-1}\right|_{m_3}\right|_{m_3} = \left|((4-2)2-2)\times 4\right|_5 = 3 \quad (۲۲)$$

درنهایت با قراردادن مقدار پیمانه‌ها و ضرایب در (۶)، رابطه زیر به‌دست می‌آید:

$$X = v_3 m_2 m_1 + v_2 m_1 + v_1 = 3\times 4\times 3 + 2\times 3 + 2 = 44 \quad (۲۳)$$

به عبارتی ۴۴ برابر با حاصل تبدیل معکوس عدد مانده‌ای (۴، ۰، ۲) است. درنتیجه به‌جای این‌که برروی ۴ و ۱۱ مستقیم ضرب را انجام دهیم، برروی باقی‌مانده‌های آن‌ها محاسبه انجام شد و صحت آن نشان می‌دهد که اطلاعات طلائی نهفته در باقی‌مانده‌ها، می‌تواند به‌طور منحصربه‌فرد بیان‌گر نمایش اعداد باشد.

۳- بازنویسی روابط حسابی تبدیل مستقیم و معکوس

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

روند طراحی *مبدل مستقیم* به‌این‌صورت است که باید باقی‌مانده تقسیم عمل‌وند وزنی ورودی بر هر یک از پیمانه‌ها به‌دست آید؛ لذا مبدل مستقیم، ساختاری موازی و مستقل برای هر پیمانه دارد. حال جهت تبدیل عدد وزنی X مبتنی‌بر مجموعه پیمانه $\{m_1, m_2, \ldots, m_n\}$ به باقی مانده‌ها (به عبارتی x_iها)، ازرابطه زیر می‌توان استفاده‌کرد:

$$x_i = |X|_{m_i} = |X_{MB-1}\cdots X_1 X_0|_{m_i} \quad for\ \ i = 1,\cdots,n \quad (۲۴)$$

که MB تعداد بیت‌های محدوده دینامیکی (یعنی M) است و m_i نشان‌گر پیمانه مدنظر است [۱]. لازم به‌ذکر است که X_i نشان‌دهنده بیت i ام از عدد دودوئی X است. باتوجه‌به تعداد بیت‌های پیمانه m_i که با m_{ib} نشان داده می‌شود، بیت‌های عمل‌وند به دسته‌های m_{ib} بیتی تقسیم می‌شوند:

$$x_i = |X|_{m_i} = \left|\begin{matrix}2^{MB}(X_{2MB-1}\cdots X_{km_{ib}})+\cdots \\ +2^{m_{ib}}(X_{2m_{ib-1}}\cdots X_{m_{ib}}) + (X_{m_{ib-1}}\cdots X_0)\end{matrix}\right|_{m_i} \quad (۲۵)$$

حال دیگر نیازی به انجام عمل تقسیم برای محاسبه باقی‌مانده نیست و توسط تعدادی جمع و ضرب، باقی‌مانده به‌دست می‌آید. به‌عبارتی (۲۵) را می‌توان به‌صورت زیر بازنویسی کرد:

$$x_i = \left|\sum_{j=0}^{k} f_j\right|_{m_i} \quad (۲۶)$$

که بردارهای دودوئی f_i به‌صورت زیر تعریف می‌شوند:

$$f_j = 2^{jm_{ib}}(X_{(j+1)m_{ib-1}}\cdots X_{jm_{ib}}) \quad (۲۷)$$

البته لازم به‌ذکر است که معمولاً پیمانه‌ها به‌صورت توانی از ۲ (مانند $2^n\pm1$) هستند و لذا ضرب‌های موردنیاز در (۲۵) حذف (تبدیل به شیفت) می‌شوند. به‌عنوان مثال، برای ۳ نوع پیمانه مشهور سیستم اعداد مانده‌ای یعنی 2^n، 2^n-1 و 2^n+1 مطابق [۸]، باقی‌مانده‌ها به‌صورت زیر به‌دست می‌آیند:

$$x_i = |X|_{m_i} = \left|\begin{matrix}2^{2n}(X_{3n-1}\cdots X_{2n}) + 2^n(X_{2n-1}\cdots X_n) \\ +(X_{n-1}\cdots X_0)\end{matrix}\right|_{m_i} \quad (۲۸)$$

باتوجه به این‌که $\left|2^{2n}\right|_{2^n} = \left|2^n\right|_{2^n} = 0$ است، x_1 به‌صورت زیر به‌دست می‌آید:

$$x_1 = |X|_{2^n} = X_{n-1}\cdots X_0 \quad (۲۹)$$

ازطرفی، باتوجه‌به این‌که $\left|2^{2n}\right|_{2^n-1} = \left|2^n\right|_{2^n-1} = 1$ است، x_2 به‌صورت زیر به‌دست می‌آید:

$$x_2 = |X|_{2^n-1} = \left|\begin{matrix}\underbrace{X_{3n-1}\cdots X_{2n}}_{f_2} \\ +\underbrace{X_{2n-1}\cdots X_n}_{f_1} \\ +\underbrace{X_{n-1}\cdots X_0}_{f_0}\end{matrix}\right|_{2^n-1} = |f_2 + f_1 + f_0|_{2^n-1} \quad (۳۰)$$

درنهایت، باتوجه‌به این‌که $\left|2^{2n}\right|_{2^n+1} = 1$ و $\left|2^n\right|_{2^n+1} = -1$ است، x_3 به‌صورت زیر به‌دست می‌آید:

$$x_3 = |X|_{2^n+1} = \left|\begin{matrix}\underbrace{X_{3n-1}\cdots X_{2n}}_{f_2} \\ -\underbrace{X_{2n-1}\cdots X_n}_{f_1} \\ +\underbrace{X_{n-1}\cdots X_0}_{f_0}\end{matrix}\right|_{2^n-1} = |f_2 - f_1 + f_0|_{2^n+1} \quad (۳۱)$$

هم‌چنین، به‌روابط حسابی مربوط به *تبدیل معکوس* در (۵) تا (۱۳) اشاره شد. حال، ضریب‌های مربوط به تبدیل درهم مبنا (فرمول‌های (۱۰) تا (۱۳))، به‌صورت زیر بازنویسی می‌شوند. لازم به‌ذکر است منظور از k_i، معکوس ضربی است:

$$v_2 = \left|\underbrace{k_1 x_2}_{v_{21}} + \underbrace{(-k_1 v_1)}_{v_{22}}\right|_{m_2} = |v_{21} + v_{22}|_{m_2} = \left|\sum_{i=1}^{2} v_{2i}\right|_{m_2} \quad (۳۲)$$

$$\begin{aligned} v_3 &= \left|\underbrace{k_3 k_2 x_3}_{v_{31}} + \underbrace{(-k_3 k_2 v_1)}_{v_{32}} + \underbrace{(-k_3 v_2)}_{v_{33}}\right|_{m_3} \\ &= |v_{31} + v_{32} + v_{33}|_{m_3} = \left|\sum_{i=1}^{3} v_{3i}\right|_{m_3} \end{aligned} \quad (۳۳)$$

$$\begin{aligned} v_4 &= \left|\underbrace{k_4 k_3 k_2 x_3}_{v_{41}} + \underbrace{(-k_4 k_3 k_2 v_1)}_{v_{42}} + \underbrace{(-k_4 k_3 v_2)}_{v_{43}} + \underbrace{(-k_4 v_3)}_{v_{44}}\right|_{m_3} \\ &= |v_{41} + v_{42} + v_{43} + v_{44}|_{m_3} = \left|\sum_{i=1}^{4} v_{4i}\right|_{m_4} \end{aligned} \quad (۳۴)$$

www.SID.ir

Archive of SID

$$v_n = \left| \underbrace{(k_n \cdots k_2)x_n}_{v_{n1}} + \underbrace{(-(k_n \cdots k_2)v_1)}_{v_{n2}} + \cdots + \underbrace{(-k_n v_{n-1})}_{v_{nn}} \right|_{m_n} = \left| \sum_{i=1}^{n} v_{ni} \right|_{m_n} \quad (۳۵)$$

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 این‌است 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. به‌عبارتی هدف، طراحی واحدی است که یک عدد وزنی دودوئی مانند X، تعدادی باقی‌مانده مانند $(y_1, y_2, \ldots, y_n)$ و سیگنال کنترلی *Control* وارد آن شود و سپس باقی مانده‌های عدد وزنی X، یعنی $(x_1, x_2, \ldots, x_n)$ 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	$y_1, y_2, \ldots, y_n$	مبدل معکوس	۰
$x_1, x_2, \ldots, x_n$	X	مبدل مستقیم	۱

حال باتوجه‌به (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:

$$Y = h_n \prod_{i=1}^{n-1} m_i + \cdots + h_3 m_2 m_1 + h_2 m_1 + h_1 \quad (۳۶)$$

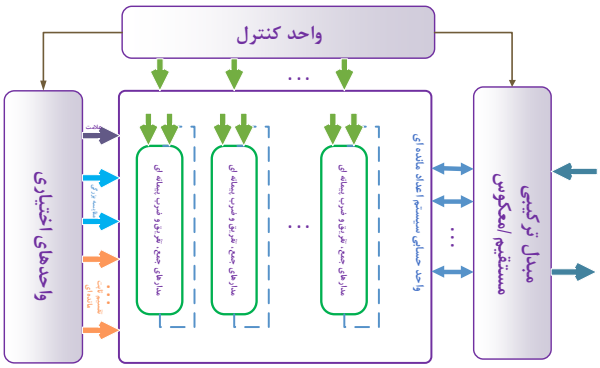
$$x_i = h_i \quad for \;\; i = 1, \cdots, n \quad (۳۷)$$

که ضرایب h_i این‌گونه محاسبه می‌شوند:

$$h_i = \begin{cases} \left| \sum_{j=1}^{i} v_{ij} \right|_{m_i} & if \;\; Control = 0 \\ \left| \sum_{j=0}^{k} f_j \right|_{m_i} & if \;\; Control = 1 \end{cases} \quad (۳۸)$$

درحالتی که سیگنال کنترلی *Control* توسط واحد کنترل، صفر شود، مد کاری مبدل معکوس انتخاب می‌شود و مقدار ضرایب تبدیل درهم مبنا که در (۹) تا (۱۳) ارائه و در (۳۲) تا (۳۵) به‌صورت 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 به‌دست آیند. حال دیگر نیازی به مدارهای مجزا برای مبدل‌های مستقیم و معکوس 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. به‌عبارتی، 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 مشخص‌شدن ابعاد کامل روش پیشنهادی، پیاده‌سازی آن مبتنی‌بر مجموعه پیمانه $\{2^{2n}, 2^n-1, 2^{n+1}-1\}$، شرح داده می‌شود. این مجموعه پیمانه در [۱۹] معرفی شده‌است و به‌دلیل 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 $\{2^{2n}, 2^n-1, 2^{n+1}-1\}$ بر طبق (۲۴) تا (۲۷) به دست می‌آید. در ابتدا، اولین باقی‌مانده مبتنی‌بر پیمانه 2^{2n} به‌صورت زیر به‌دست می‌آید:

$$x_1 = |X|_{2^{2n}} = |X_{4n} \cdots X_1 X_0|_{2^{2n}} = \left| \underbrace{2^{2n}(X_{4n} \cdots X_{2n+1} X_{2n})}_{2n+1 \; bits} + \underbrace{X_{2n-1} \cdots X_1 X_0}_{2n \; bits} \right|_{2^{2n}} = \underbrace{X_{2n-1} \cdots X_1 X_0}_{2n \; bits} \quad (۳۹)$$

به‌عبارتی 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 به‌صورت زیر محاسبه می‌شود:

www.SID.ir

Archive of SID

$$x_2=\left|X\right|_{2^n-1}=\left|X_{4n}\cdots X_1X_0\right|_{2^n-1}$$
$$=\left|\begin{array}{l}2^{4n}X_{4n}+2^{3n}(\underbrace{X_{4n-1}\cdots X_{3n+1}X_{3n}}_{n\ bits})\\+2^{2n}(\underbrace{X_{3n-1}\cdots X_{2n+1}X_{2n}}_{n\ bits})\\+2^{n}(\underbrace{X_{2n-1}\cdots X_{n+1}X_{n}}_{n\ bits})+\underbrace{X_{n-1}\cdots X_1X_0}_{n\ bits}\end{array}\right|_{2^n-1} \quad (۴۰)$$
$$=\left|\sum_{i=0}^{4}f_i\right|_{2^n-1}$$

که بردارهای دودوئی f_i، این‌گونه می‌باشند:

$$f_i=\underbrace{X_{(i+1)n-1}\cdots X_{in+1}X_{in}}_{n\ bits}\quad for\ \ i=0\ to\ 3 \quad (۴۱)$$

$$f_4=\left|X_{4n}\times 2^{4n}\right|_{2^n-1}=\underbrace{0\cdots 00}_{n-1\ bits}X_{4n} \quad (۴۲)$$

درنهایت، باقی‌مانده سوم مبتنی‌بر پیمانه 2^{n+1}-1 به‌صورت زیر محاسبه می‌شود:

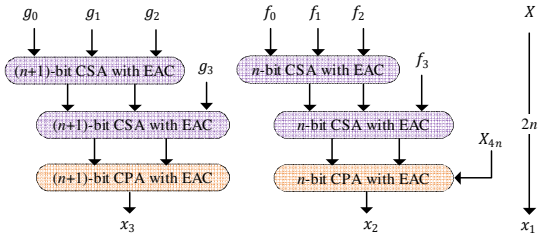
$$x_3=\left|X\right|_{2^{n+1}-1}=\left|X_{4n}\cdots X_1X_0\right|_{2^{n+1}-1}$$
$$=\left|\begin{array}{l}2^{3n+3}(000\underbrace{X_{4n}\cdots X_{3n+4}X_{3n+3}}_{n-2\ bits})\\+2^{2n+2}(\underbrace{X_{3n+2}\cdots X_{2n+3}X_{2n+2}}_{n+1\ bits})\\+2^{n+1}(\underbrace{X_{2n+1}\cdots X_{n+2}X_{n+1}}_{n+1\ bits})+\underbrace{X_{n}\cdots X_1X_0}_{n+1\ bits}\end{array}\right|_{2^n+1-1} \quad (۴۳)$$
$$=\left|\sum_{i=0}^{3}g_i\right|_{2^{n+1}-1}$$

باتوجه‌به این‌که به‌ازای $k\geq 1$ رابطه $\left|2^{k(n+1)}\right|_{2^{n+1}-1}=1$ برقرار است، بردارهای دودوئی g_i به‌صورت زیر می‌توانند محاسبه شوند:

$$g_i=\underbrace{X_{(i+1)n+i}\cdots X_{in+i+1}X_{in+i}}_{n+1\ bits}\quad for\ \ i=0\ to\ 2 \quad (۴۴)$$

$$g_3=\underbrace{000}_{3\ bits}\underbrace{X_{4n}\cdots X_{3n+4}X_{3n+3}}_{n-2\ bits} \quad (۴۵)$$

همان‌طورکه از (۴۰) و (۴۳) مشاهده می‌شود، برای به‌دست‌آوردن باقی مانده‌ها، فقط کافی است که تعدادی بردار دودوئی 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 شده‌است.



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

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) به‌صورت جمع برداری پیمانه‌ای، ساده می‌شوند. ازآنجایی‌که طراحی مبدل معکوس برای مجموعه پیمانه $\{2^{2n}, 2^n-1, 2^{n+1}-1\}$ مبتنی بر تبدیل درهم مبنا به‌صورت 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. ازاین‌رو، برای تبدیل باقی‌مانده‌ها به معادل وزنی داریم (رابطه‌های (۱۵) تا (۱۷) در [۱۹]):

$$Y=2^{2n}(2^n-1)v_3+2^{2n}v_2+v_1 \quad (۴۶)$$

که ضرایب آن عبارتند از:

$$v_1=x_1 \quad (۴۷)$$

$$v_2=\left|x_2-x_1\right|_{2^n-1} \quad (۴۸)$$

$$v_3=\left|((x_3-x_1)2^2-v_2)(-2)\right|_{2^{n+1}-1} \quad (۴۹)$$

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

$$Y=2^{2n}(v_2+2^nv_3-v_3)+x_1 \quad (۵۰)$$

$$v_2=\left|w_1+w_{21}+w_{22}\right|_{2^n-1} \quad (۵۱)$$

$$v_3=\left|w_3+w_4+w_{51}+w_{52}\right|_{2^{n+1}-1} \quad (۵۲)$$

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

$$w_1=\underbrace{x_{2,n-1}\cdots x_{2,1}x_{2,0}}_{n\ bits} \quad (۵۳)$$

$$w_{21}=\underbrace{\overline{x}_{1,n-1}\cdots\overline{x}_{1,1}\overline{x}_{1,0}}_{n\ bits} \quad (۵۴)$$

$$w_{22}=\underbrace{\overline{x}_{1,2n-1}\cdots\overline{x}_{1,n+1}\overline{x}_{1,n}}_{n\ bits} \quad (۵۵)$$

$$w_3=\underbrace{v_{2,n-1}\cdots v_{2,1}v_{2,0}}_{n\ bits}0 \quad (۵۶)$$

$$w_4=\underbrace{\overline{x}_{3,n-3}\cdots\overline{x}_{3,1}\overline{x}_{3,0}}_{n-2\ bits}\overline{x}_{3,n}\overline{x}_{3,n-1}\overline{x}_{3,n-2} \quad (۵۷)$$

$$w_{51}=\underbrace{x_{1,n-3}\cdots x_{1,1}x_{1,0}}_{n-2\ bits}x_{1,n}x_{1,n-1}x_{1,n-2} \quad (۵۸)$$

$$w_{52}=\underbrace{x_{1,2n-2}\cdots x_{1,n+2}x_{1,n+1}}_{n-2\ bits}00x_{1,2n-1} \quad (۵۹)$$

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 شده‌است. همان‌طور که مشاهده می‌شود از جمع‌کننده‌های ذخیره رقم نقلی و انتشار رقم نقلی چرخشی مطابق شیوه ارائه‌شده در [۱۳]، استفاده شده‌است.

بررسی شکل‌های ۳ و۴، شباهت و اشتراک سخت‌افزارهای مبدل‌های مستقیم و معکوس را مشخص می‌کند. برای پیاده‌سازی (۴۰) و (۴۳)

www.SID.ir

Archive of SID

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

$$\begin{aligned} Y &= 2^{2n}(2^n-1)h_3 + 2^{2n}h_2 + h_1 \\ &= 2^{2n}(h_2 + 2^n h_3 - h_3) + y_1 \end{aligned} \quad (۶۰)$$

$$x_1 = h_1 = \underbrace{X_{2n-1}\cdots X_1 X_0}_{2n\ bits} \quad (۶۱)$$

$$x_2 = h_2 \quad (۶۲)$$

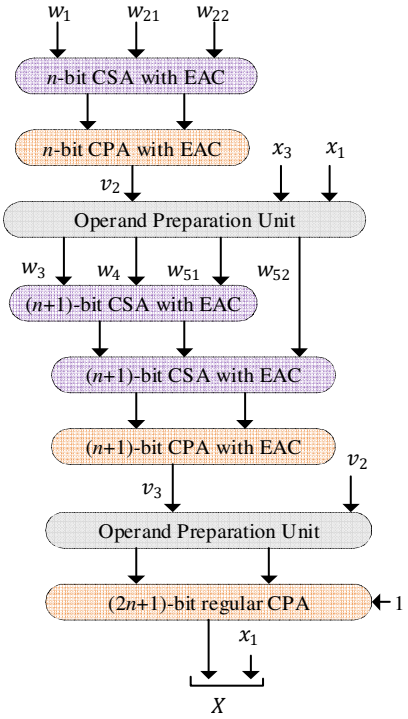
$$x_3 = h_3 \quad (۶۳)$$

که ضرایب h_2 و h_3 این‌گونه محاسبه می‌شوند:

$$h_2 = \begin{cases} \left|w_1 + w_{21} + w_{22}\right|_{2^n-1} & if\ \ Control = 0 \\ \left|f_0 + f_1 + f_2 + f_3 + f_4\right|_{2^n-1} & if\ \ Control = 1 \end{cases} \quad (۶۴)$$

$$h_3 = \begin{cases} \left|w_3 + w_4 + w_{51} + w_{52}\right|_{2^{n+1}-1} & if\ \ Control = 0 \\ \left|g_0 + g_1 + g_2 + g_3\right|_{2^{n+1}-1} & if\ \ Control = 1 \end{cases} \quad (۶۵)$$

لازم به‌ذکر است که نحوه محاسبه و تنظیم بیت‌های بردارهای موردنیاز در (۶۴) و (۶۵) در (۴۰) تا (۴۵) و (۵۳) تا (۵۹) مشخص شده‌است. حال باتوجه‌به (۶۰) تا (۶۵)، معماری سخت‌افزاری مبدل ترکیبی مستقیم/معکوس پیشنهادی در شکل ۵ نشان داده شده‌است. ورودی و خروجی‌های این واحد ترکیبی در جدول ۱ مشخص شده‌است.



شکل ۴: مبدل معکوس برای مجموعه پیمانه $\{2^{2n}, 2^n-1, 2^{n+1}-1\}$ [۱۹]

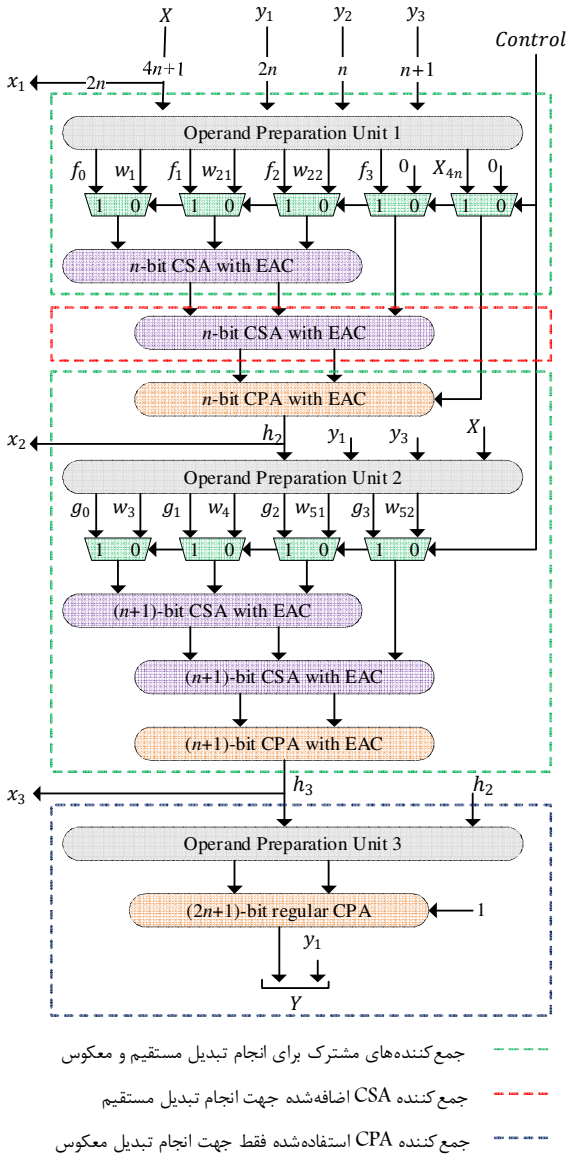
همان‌طورکه در شکل ۵ مشاهده می‌شود، مالتی پلکسرهایی که در ورودی جمع کننده‌های ذخیره رقم نقلی اضافه‌شده به‌همراه سیگنال کنترلی *Control*، عمل‌کرد مدار را مشخص می‌کنند. اگر *Control* صفر باشد، حالت عمل‌کردی تبدیل معکوس انتخاب می‌شود و (y_1, y_2, y_3) ورودی‌های مدار و Y خروجی خواهدبود. هم‌چنین اگر *Control* یک باشد، حالت عمل‌کردی تبدیل مستقیم انتخاب می‌شود و X ورودی و (x_1, x_2, x_3) خروجی‌های مدار خواهندبود.

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 شده‌است تا چهارمین عمل‌وند موردنیاز جهت تبدیل مستقیم در (۴۰) را پوشش دهد. هم‌چنین، پنجمین عمل‌وند موردنیاز در (۴۰) یعنی f_4در (۴۲)، فقط توسط اتصال X_{4n} به ورودی رقم نقلی جمع‌کننده پیمانه‌ای، در جمع دخالت داده شده‌است. این نکته باعث شد تا از یک جمع‌کننده ذخیره رقم نقلی اضافی صرف‌نظر شود. لازم به‌ذکر است که آخرین جمع‌کننده انتشار رقم نقلی استفاده‌شده در شکل ۵، فقط جهت انجام تبدیل معکوس نیاز است. به‌عبارتی در حالت کاری تبدیل مستقیم، این جمع‌کننده نقشی ندارد.

مقایسه مبدل ترکیبی پیشنهادی (شکل ۵) و مبدل‌های مستقیم و معکوس عادی (شکل‌های ۳ و ۴)، نشان می‌دهد که چگونه مالتی‌پلکسرها امکان اشتراک سخت افزار را فراهم کرده‌اند. همان‌گونه‌که مشاهده می شود، مالتی پلکسرها با استفاده از سیگنال کنترلی، ورودی‌های مناسب را به جمع‌کننده‌های ذخیره رقم نقلی هدایت می‌کنند.

www.SID.ir

Archive of SID



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 $n=2$ به‌صورت {۷، ۳، ۱۶} 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 (۵، ۲، ۹) وارد مبدل معکوس شود و این مبدل باید معادل وزنی آن، یعنی Y را به‌دست 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 خواهدشد.

تبدیل مستقیم عدد X=۱۲۰ برحسب مجموعه پیمانه {۷، ۳، ۱۶} و مبتنی بر (۳۹) تا (۴۵) به‌صورت زیر انجام می‌شود:

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

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

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

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

به‌عبارتی (۱، ۰، ۸) نمایش مانده‌ای عدد ۱۲۰ است. هم‌چنین، تبدیل معکوس عدد مانده‌ای (۵، ۲، ۹)=(x_1, x_2, x_3) برحسب مجموعه پیمانه {۷، ۳، ۱۶} و مبتنی بر (۵۰) تا (۵۹) به‌صورت زیر انجام می‌شود:

$$x_1 = 9 = (1001)_2 \quad (۷۰)$$

$$x_2 = 2 = (10)_2 \quad (۷۱)$$

$$x_3 = 5 = (101)_2 \quad (۷۲)$$

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

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

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

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

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

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

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

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

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

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

به‌عبارتی ۸۹ برابر با حاصل تبدیل معکوس عدد مانده‌ای (۵، ۲، ۹) است. لازم به‌ذکر است & بیان‌گر عمل‌گر الحاق بیتی است. درنهایت، توسط مبدل ترکیبی پیشنهادی برحسب 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. ازاین‌رو، برپایه روابط (۶۰) تا (۶۵) و با درنظرگرفتن (۶۸) و (۶۹)، و نیز (۷۶) و (۸۱)، رابطه‌های (۸۳) و (۸۴) به‌دست می‌آیند.

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

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

www.SID.ir

Archive of SID

حال اگر سیگنال کنترلی برابر با یک باشد، مدار در حالت عمل‌کردی تبدیل مستقیم قرار می‌گیرد و باقی‌مانده‌ها به‌صورت زیر محاسبه می شوند:

جدول ۲: سخت‌افزار موردنیاز و تأخیر مبدل‌های مختلف

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

$$x_1 = h_1 = \underbrace{X_{2n-1}\cdots X_1 X_0}_{2n\ bits} = (1000)_2 = 8 \quad (۸۵)$$

$$x_2 = h_2 = 0 \quad (۸۶)$$

$$x_3 = h_3 = 1 \quad (۸۷)$$

درصورتی که سیگنال کنترلی برابر با صفر باشد، مدار در حالت عمل‌کردی تبدیل معکوس قرار می‌گیرد و عدد وزنی به‌صورت زیر محاسبه می‌شود:

$$\begin{aligned} Y &= 2^{2n}(2^n-1)h_3 + 2^{2n}h_2 + h_1 \\ &= 2^{2n}(h_2 + 2^n h_3 - h_3) + x_1 = (h_3 \,\&\, h_2 - h_3)\,\&\, x_1 \\ &= (00110-001)_2 \,\&\, (1001) = (001011001)_2 = 89 \end{aligned} \quad (۸۸)$$

۶- ارزیابی کارایی

باتوجه‌به این‌که برای اولین بار است یک طراحی ترکیبی از مبدل مستقیم و معکوس ارائه می‌شود، برای ارزیابی کارایی آن، می‌توان مقایسه را با طراحی‌های مستقل مبدل مستقیم و معکوس انجام داد. ازاین‌رو، 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 خواهندشد. در ابتدا تحلیل مستقل از تکنولوژی و بر پایه گیت‌های منطقی ارائه می‌شود و سپس ارزیابی مبتنی بر نتایج تجربی پیاده‌سازی VLSI و نیز FPGA، انجام خواهدشد.

جدول ۲ سخت‌افزار موردنیاز و تأخیر مدارهای ذکرشده را برحسب تعداد تمام جمع‌کننده‌ها مقایسه می‌کند. لازم به‌ذکر است A_{FA} و T_{FA} به ترتیب نشان دهنده سطح و تأخیر یک تمام جمع‌کننده است. همچنین مشابه [۱۹]، جمع‌کننده انتشار رقم نقلی با چرخش رقم نقلی انتهایی k بیتی مانند یک جمع‌کننده انتشار رقم نقلی عادی در نظر گرفته شده‌است که رقم نقلی انتهایی، مجدد وارد می‌شود تا رقم نقلی را با عمل‌وندها جمع کند تا جمع پیمانه‌ای انجام شود. به‌عبارتی جمع در دو مرحله انجام می‌شود. لذا سخت‌افزار موردنیاز و نیز تأخیر آن معادل سطح و تأخیر k و $2k$ تمام جمع‌کننده، به‌ترتیب، درنظر گرفته می‌شود. علاوه‌براین، سخت افزار موردنیاز برای جمع‌کننده ذخیره رقم نقلی k بیتی معادل k تمام جمع‌کننده و تأخیر آن برابر یک تمام جمع‌کننده درنظر گرفته می‌شود [۱۳]. ازطرفی دیگر، تمام جمع‌کننده‌های با یک ورودی بیتی ثابت می‌توانند به نیم جمع‌کننده‌ها ساده شوند، ولی برای مقایسه یکسان بین همه طراحی‌ها، هیچ‌گونه ساده‌سازی تمام جمع‌کننده‌ها در جدول ۲ در نظر گرفته نشده‌است. در مجموع، طراحی پیشنهادی در شکل ۵، نیاز به $8n+4$ تمام جمع‌کننده و ۹ مالتی پلکسر ۲ به ۱ دارد که ۴ تای آن‌ها n بیتی، ۴ تای دیگر $n+1$ بیتی و یکی هم ۱ بیتی است. بنابراین $8n+5$ مالتی پلکسر ۲ به ۱ نیاز است. طبق مدل گیت واحد [۱]، هر تمام جمع‌کننده، سخت‌افزار معادل ۷ گیت و هر مالتی پلکسر ۲ به ۱ تک بیتی، سخت‌افزار معادل ۳ گیت واحد دارد. لذا برای ساده‌ترشدن مقایسه، می‌توان هر مالتی پلکسر را معادل ۰/۴۲ یک تمام جمع‌کننده در نظر گرفت. بنابراین درمجموع، سخت افزار مورد نیاز مدار مبدل ترکیبی پیشنهادی می‌تواند معادل $11.4n+6.14$ تا تمام جمع‌کننده در نظر گرفته شده‌است.

همان‌طورکه در جدول ۲ مشاهده می‌شود، مبدل ترکیبی مستقیم/معکوس پیشنهادی نسبت به حالت داشتن هر دو مبدل به طور هم‌زمان در سیستم اعداد مانده‌ای، سخت‌افزار کم‌تری نیاز دارد زیرا تمام جمع‌کننده‌های پیمانه‌ای موردنیاز در هر دو مبدل، در ساختار ترکیبی به‌صورت اشتراکی استفاده می‌شود. در مقایسه با مبدل معکوس عادی [۱۹]، فقط یک جمع‌کننده ذخیره رقم نقلی و تعدادی مالتی پلکسر در مدار ترکیبی اضافه شده‌است. علاوه‌براین، در حالت کاری تبدیل مستقیم، محاسبه باقی‌مانده‌های خروجی یعنی h_2 و h_3 کاملاً موازی انجام می‌شود، زیرا (۴۰) و (۴۳) مستقل از یک‌دیگر هستند و فقط مالتی پلکسرها نسبت به مبدل مستقیم عادی به مسیر بحرانی تأخیر اضافه شده‌است. لذا تأخیر مدار ترکیبی پیشنهادی در حالت تبدیل مستقیم، $(2n+4)\ T_{FA} + T_{MUX}$ است که در جدول ۲، تأخیر بدترین حالت، یعنی حالت کاری تبدیل معکوس نوشته شده‌است.

جهت ارزیابی دقیق تأخیر، سطح تراشه و توان مصرفی مبدل‌های مختلف، همه طراحی‌ها 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 پیاده‌سازی شدند که نتایج حاصل در جدول ۳ نشان داده شده‌اند. لازم به‌ذکر است، چون در حالت کاری مبدل معکوس، مدار محاسبات بیش‌تری باید انجام دهد، توان مصرفی آن بیشتر از حالت کاری مبدل مستقیم می‌شود. لذا در جدول ۳، توان مصرفی در بدترین حالت، یعنی در حالت کاری مبدل معکوس درج شده‌است. طبق جدول ۳، به‌ازای $n=8$، مبدل ترکیبی پیشنهادی منجر به حداکثر ۱۹ درصد کاهش سطح تراشه در مقایسه با

www.SID.ir

Archive of SID

داشتن هر دو مبدل مستقیم و معکوس به طور مستقل، شده‌است. علاوه‌براین، تأخیر مدار ترکیبی مبدل مستقیم/معکوس پیشنهادی نسبت به مبدل معکوس مستقل، کمی بیش‌تر شده‌است، زیرا مالتی پلکسرها و یک جمع‌کننده ذخیره رقم نقلی به مسیر بحرانی تأخیر مبدل اضافه شده‌است. به‌خصوص هر چه n افزایش پیدا می‌کند، تفاضل تأخیر مبدل ترکیبی پیشنهادی و مبدل معکوس مستقل، کم‌تر می‌شود (۱۰ درصد و ۴ درصد به‌ازای n=4 و n=16 به‌ترتیب).

علاوه‌بر پیاده‌سازی ASIC مدارها، تمامی طراحی‌ها مبتنی بر ابزار ISE برروی تراشه FPGA Virtex5 مدل XC5VLX85 پیاده‌سازی شدند که نتایج حاصل در جدول ۴ نشان داده شده‌است. همان‌گونه‌که مشاهده می‌شود، مبدل ترکیبی پیشنهادی نسبت به مجموع مبدل‌های مستقیم و معکوس نیاز به تعداد LUT های کمتر ولی تأخیر بیشتر دارد. هم‌چنین اگرچه، تأخیر طرح پیشنهادی از مبدل‌های مستقیم و معکوس عادی بیش‌تر است، ولی درصد کاهش سخت‌افزار نسبت به حالت مجموع هر دو مبدل، چشمگیر است. به‌عنوان مثال، به‌ازای مقدار ۱۶ برای n، طرح پیشنهادی، ۲۰ درصد سخت‌افزار (تعداد LUT های موردنیاز) را کاهش داده، ولی فقط ۱۰ درصد تأخیر را در مقایسه با مجموع مبدل مستقیم/معکوس افزایش داده‌است.

جدول ۴: پارامترهای مداری حاصل از پیاده‌سازی FPGA مبدل‌های مختلف

مدار	n	حجم مصرفی (تعداد LUT ها)	تأخیر (ns)
مبدل معکوس [۱۹]	۴	۶۰	۵/۹۴۰
	۸	۱۶۱	۷/۷۳۲
	۱۶	۴۱۳	۹/۴۸۲
مبدل مستقیم	۴	۴۹	۳
	۸	۱۰۳	۴/۰۷۴
	۱۶	۲۶۳	۴/۹۷۶
مجموع مبدل‌های مستقیم و معکوس	۴	۱۰۹	۵/۹۴۰
	۸	۲۶۴	۷/۷۳۲
	۱۶	۶۷۶	۹/۴۸۲
مبدل ترکیبی مستقیم/معکوس پیشنهادی	۴	۱۰۱	۶/۹۳۸
	۸	۲۳۵	۹/۳۸۲
	۱۶	۵۳۵	۱۰/۵۵۹

۷- نتیجه

در این مقاله، برای اولین بار، یک طراحی ترکیبی از مبدل‌های مستقیم و معکوس ارائه شد که توانست سربار بالای ناشی از سخت‌افزار موردنیاز برای مبدل‌های مستقیم و معکوس مستقل را کاهش دهد. لذا با داشتن چنین مبدل ترکیبی، پیچیدگی سخت‌افزاری کلی سیستم اعداد مانده‌ای کاهش می‌یابد و امکان استفاده از آن در کاربردهایی که نیاز به سخت‌افزار کم دارند مانند سیستم‌های نهفته فراهم می‌آید.

سپاس‌گزاری

نویسندگان مقاله تمایل دارند از آقای لئونل سوسا و آزمایشگاه پردازش سیگنال‌های دیجیتال دانشگاه لیسبون کشور پرتغال جهت همکاری در پیاده‌سازی VLSI مبدل‌ها، تشکر کنند.

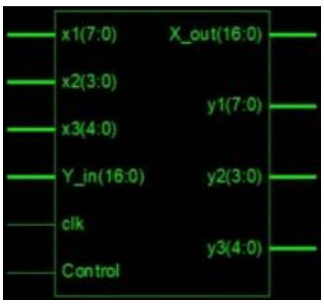
پیوست‌ها

در این بخش RTL Schematic های مربوط به مبدل ترکیبی پیشنهادی

جدول ۳: پارامترهای مداری حاصل از پیاده‌سازی VLSI مبدل‌های مختلف

مدار	n	سطح تراشه (μm^2)	تأخیر (ns)	توان مصرفی (mW)
مبدل معکوس [۱۹]	۴	۲۷۵۶	۱/۱۴۶	۱/۸۸۷
	۸	۴۷۳۳	۱/۸۱۰	۲/۲۷۴
	۱۶	۸۳۲۸	۳/۱۱۷	۲/۴۹۲
مبدل مستقیم	۴	۲۱۹۰	۰/۸۴۰	۰/۹۱۵
	۸	۴۱۹۹	۰/۸۸۶	۲/۵۱۱
	۱۶	۷۷۰۹	۱/۴۵۲	۳/۰۱۱
مجموع مبدل‌های مستقیم و معکوس	۴	۴۹۴۶	۱/۱۴۶	۲/۸۰۲
	۸	۸۹۳۲	۱/۸۱۰	۴/۷۸۵
	۱۶	۱۶۰۳۷	۳/۱۱۷	۵/۵۰۳
مبدل ترکیبی مستقیم/معکوس پیشنهادی	۴	۳۹۸۲	۱/۲۸۵	۲/۳۵۷
	۸	۷۱۹۳	۱/۹۹۴	۳/۱۷۲
	۱۶	۱۳۳۸۵	۳/۲۵۹	۳/۸۳۷

برای مجموعه پیمانه $\{2^{2n}, 2^n-1, 2^{n+1}-1\}$ به‌ازای n=4، که از ابزار ISE به‌دست آمده‌است، رسم شده‌است. شکل ۶ بلوک دیاگرام کلی و ورودی/خروجی‌های مبدل ترکیبی پیشنهادی را نشان می‌دهد. همان‌طورکه مشاهده می‌شود، ورودی‌های مدار شامل سیگنال *Control* (یک بیتی)، باقی‌مانده‌های x_1، x_2، x_3 و عدد وزنی Y می‌باشند. هم‌چنین، خروجی‌های مدار عدد وزنی X و نیز باقی‌مانده‌های y_1، y_2 و y_3 می‌باشند. باتوجه‌به این‌که n=4 است، باقی‌مانده‌های متناظر با پیمانه‌های 2^{2n}، 2^n-1 و $2^{n+1}-1$ به‌ترتیب ۸، ۴ و ۵ بیتی و نیز اعداد وزنی ورودی و خروجی ۱۷ بیتی می‌باشند.

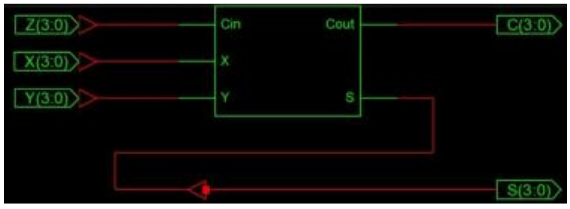


شکل ۶: بلوک دیاگرام کلی مبدل ترکیبی مستقیم/معکوس پیشنهادی برای مجموعه پیمانه $\{2^{2n}, 2^n-1, 2^{n+1}-1\}$ به‌ازای n=4

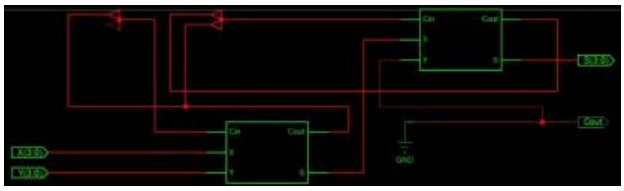
www.SID.ir

Archive of SID

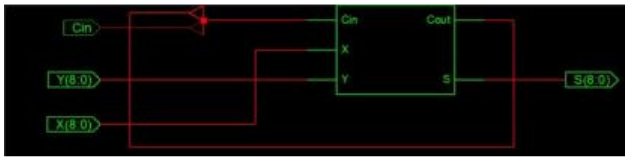
در شکل ۷ ساختار داخلی مبدل ترکیبی نشان داده شده‌است که از جمع‌کننده‌های ذخیره رقم نقلی (CSA ها)، جمع‌کننده‌های انتشار رقم نقلی (CPA ها) و واحدهای آماده‌سازی عمل‌وندها (OPU ها) تشکیل شده‌است. در شکل ۸ بلوک دیاگرام جمع‌کننده ذخیره رقم نقلی ۱، و در شکل ۹ ساختار جمع‌کننده انتشار رقم نقلی پیمانه‌ای که رقم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



شکل ۸: بلوک دیاگرام کلی جمع‌کننده ذخیره رقم نقلی

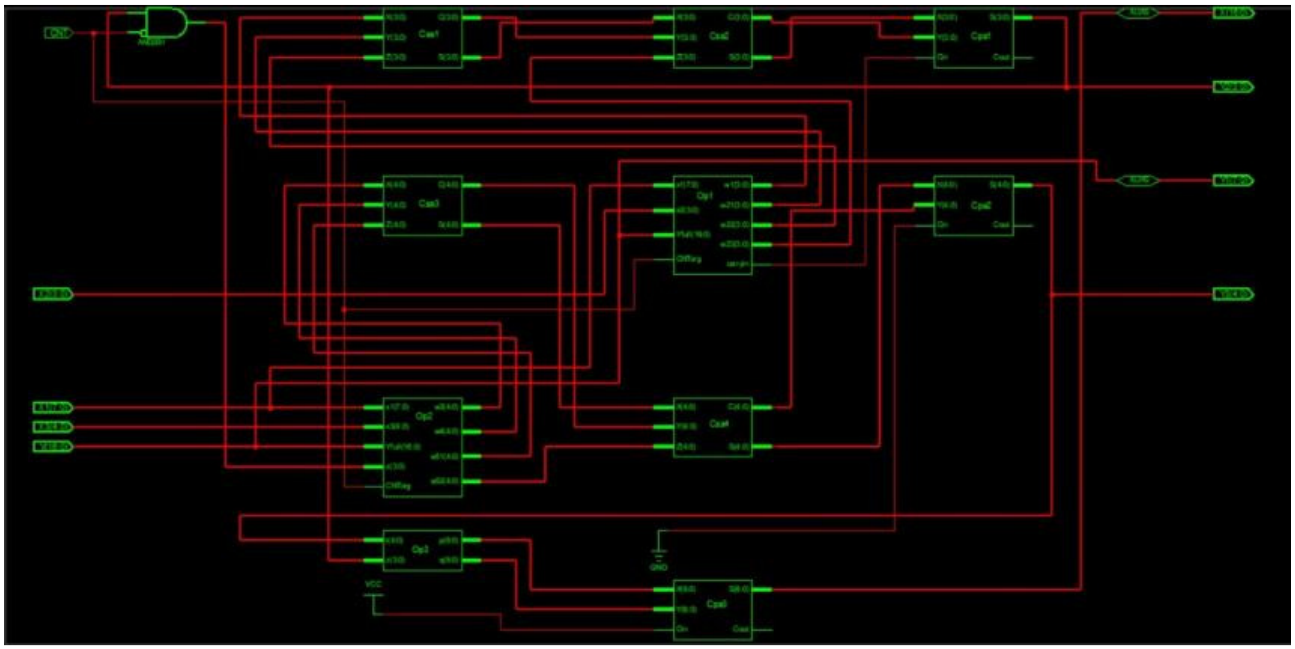


شکل ۹: ساختار کلی جمع‌کننده انتشار رقم نقلی پیمانه‌ای



شکل ۱۰: ساختار کلی جمع‌کننده انتشار رقم نقلی عادی

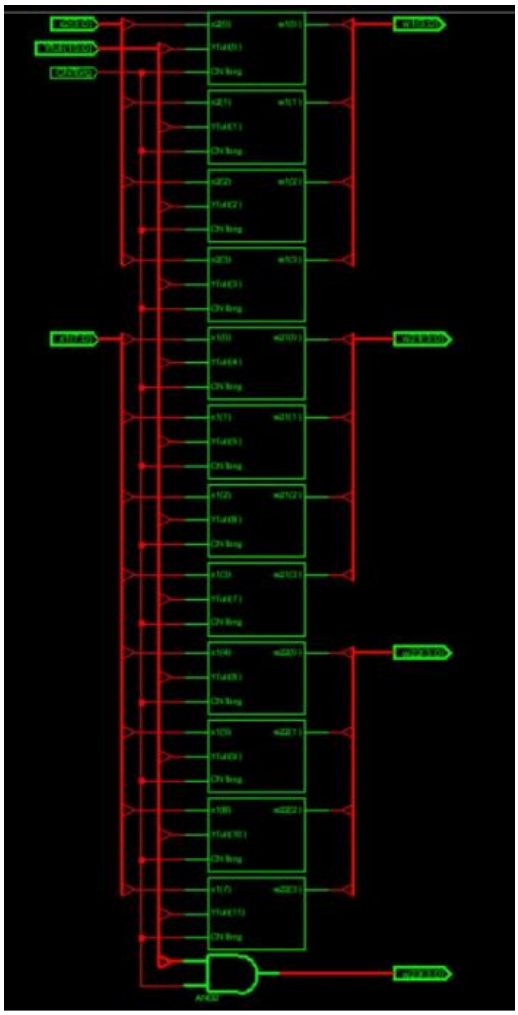
همچنین در شکل ۱۱، معماری کلی واحد آماده‌سازی عملوندهای ۱ (OPU1) نشان داده شده‌است. در واحدهای OPU اول و دوم، مالتی پلکسرها قرار دارند که ورودی‌های مناسب جهت جمع‌کننده‌های ذخیره رقم نقلی را فراهم می‌کنند. در OPU سوم، ازطریق سیم بندی و تعدادی گیت معکوس‌کننده، ورودی‌های مناسب جمع‌کننده انتشار رقم نقلی انتهایی مدار آماده می‌شوند.



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

www.SID.ir

Archive of SID



شکل ۱۱: 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.

[۲] 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.

[۳] 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 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. Esmaeildoust, "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±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. Pekmestzi, "On the design of configurable modulo 2n±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±k} jn-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.

www.SID.ir

Archive of SID

زیرنویس‌ها

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

www.SID.ir