دوره ۷، شماره ۲

مجله فناوری اطلاعات در طراحی مهندسی
Information Technology on Engineering Design
http://ited.sinaweb.net

طراحی مدار شناساگر علامت جدید برای سیستم اعداد مانده‌ای در مجموعه چهار پیمانه‌ای $\left(2^n-1,\ 2^n,\ 2^n+1,\ 2^{2n}+1\right)$

شیوا قرقانی* (۱)، مهدی حسین‌زاده(۲)، امیر صباغ ملاحسینی(۳)

(۱) دانشجوی کارشناسی ارشد، دانشگاه آزاد اسلامی، واحد اراک، گروه کامپیوتر، اراک، ایران.

(۲) استادیار، دانشگاه آزاد اسلامی، واحد علوم و تحقیقات تهران، گروه کامپیوتر، تهران، ایران.

(۳) استادیار، دانشگاه آزاد اسلامی، واحد کرمان، گروه کامپیوتر، کرمان، ایران.

چکیده:

کاربردهای امروزی نیازمند محدوده‌ی دینامیکی بالاتر و موازی سازی بیشتر می‌باشند. بنابراین در این مقاله از مجموعه چهار پیمانه‌ای جدید $\left(2^n-1,\ 2^n,\ 2^n+1,\ 2^{2n}+1\right)$ که در مقایسه با اغلب مجموعه‌های سه پیمانه‌ای دارای محدوده‌ی دینامیکی بالاتر و موازی سازی بیشتری می‌باشد، استفاده شده است. سپس برای اولین‌بار طراحی مدار شناساگر علامت جدید و کارآمد مخصوص سیستم اعداد مانده‌ای چهار پیمانه‌ای $\left(2^n-1,\ 2^n,\ 2^n+1,\ 2^{2n}+1\right)$ ارائه شده است تا در پی آن استفاده عملی از این مجموعه پیمانه در سیستم‌های محاسباتی مبتنی بر سیستم اعداد مانده‌ای امکان پذیر گردد. محدوده‌ی دینامیکی این مجموعه پیمانه 5n بیتی می‌باشد و از مزایای مدار پیشنهادی این است که تنها با محاسبه n بیت شناسایی علامت صورت می‌گیرد. هم‌چنین در ساختار سخت افزاری مدار پیشنهادی بر خلاف روش پیشین از مدار مقایسه‌گر استفاده نمی‌شود و تنها با استفاده از جمع کننده‌های ذخیره رقم نقلی و جمع کننده‌های انتشار رقم نقلی، علامت تشخیص داده می‌شود. بنابراین مدار پیشنهادی در مقایسه با روش پیشین دارای سرعت بیشتر و هزینه سخت افزاری کمتری می‌باشد.

واژه‌های کلیدی:

حساب کامپیوتری، سیستم اعداد مانده‌ای، مدار شناساگر علامت، مجموعه پیمانه‌ی کارآمد.

* عهده‌دار مکاتبات

نشانی: دانشگاه آزاد اسلامی، واحد اراک، گروه کامپیوتر، اراک، ایران.

تلفن: ۳۳۶۶۳۰۴۹-۳۳۶۶۳۰۴۱-۰۸۶ پست الکترونیکی:gharghani3754@yahoo.com

۱- مقدمه

سیستم اعداد مانده‌ای[۱] در دهه‌های اخیر به عنوان یک زمینه‌ی تحقیقاتی بسیار مهم در حساب کامپیوتری به شمار می‌رود. علت این امر آن است که این سیستم یک سیستم عددی بدون انتشار رقم نقلی است که منجر به معماری‌های محاسباتی با کارآیی بالا و توان مصرفی پایین و تاخیر کم می‌گردد. در سیستم اعداد مانده‌ای یک عدد وزن‌دار به مجموعه‌ای از مانده‌های کوچک تبدیل می‌گردد و محاسباتی نظیر جمع، تفریق و ضرب بر روی مانده‌ها به صورت موازی و بدون انتشار رقم نقلی مابین ارقام مانده‌ای صورت می‌گیرد. بنابراین در این سیستم عملیات جمع، تفریق و ضرب با سرعت بسیار زیادی انجام می‌شوند[۱].

با توجه به خواص سیستم اعداد مانده‌ای، این سیستم عددی در کاربردهای محاسباتی مانند سیستم‌های پردازش تصویر [۳-۲]، سیستم رمزRSA [۵-۴]، سیستم پردازش سیگنال‌های دیجیتال [۷-۶] و فیلترهای دیجیتال [۱۱-۸] به میزان قابل توجهی مورد استفاده قرار می‌گیرد و به طور کلی این سیستم، به دلیل خاصیت عدم انتشار رقم نقلی در اعمال حسابی، برای کاربردهایی که در محدوده‌ای از اعداد اعمال جمع، تفریق و ضرب تکرار می‌شوند به میزان قابل توجهی مورد استفاده قرار می‌گیرد. علاوه بر این، به طور گسترده‌ای از سیستم اعداد مانده‌ای افزونه برای تشخیص و تصحیح خطا استفاده می‌شود [۱۴-۱۲].

در سیسستم اعداد مانده‌ای عملیاتی نظیر تقسیم، مقایسه‌ی بزرگی، تشخیص علامت اعداد مانده‌ای و مبدل مانده‌ای به باینری (تبدیل معکوس) بسیار پیچیده و مشکل می‌باشند. یکی از مهم‌ترین مسائل مطرح در سیستم اعداد مانده‌ای، طراحی شناساگر علامت است زیرا به دلیل بی وزن بودن باقیمانده‌ها در این سیستم، تشخیص علامت اعداد مانده‌ای در مقایسه با سیستم‌های عددی وزن‌دار بسیار پیچیده است وهمین امر مانعی اساسی برای استفاده‌ی گسترده از این سیستم در بسیاری از کاربردها می‌گردد. علاوه بر این شناساگر علامت به عنوان یک عنصر اساسی در الگوریتم‌های پیچیده نظیر تقسیم و مقایسه‌ی اعداد به شمار می‌رود و در کاربردهای هندسه‌ی محاسباتی مورد استفاده قرار می‌گیرد که از آن جمله می‌توان به الگوریتم‌های مثلث بندی دلونی، دیاگرام‌های ورونوی یا آزمون‌های تقاطع اشاره کرد [۱۵].

طراحی شناساگر علامت توسط محققان زیادی مورد بررسی قرار گرفته است و همواره تحقیقات آن‌ها به طور جدی در مراکز تحقیقاتی دنیا دنبال می‌شود. یکی از روش‌های پیشنهادی به منظور تشخیص علامت اعداد مانده‌ای، تبدیل سیستم اعداد مانده‌ای به سیستم اعداد وزنی و سپس مقایسه نتیجه‌ی حاصله با عدد مناسبی است که معمولاً $\frac{1}{2}$ محدوده‌ی دینامیکی سیستم اعداد مانده‌ای می‌باشد. در [۱۶] الگوریتمی برای تبدیل سیستم اعداد مانده‌ای به سیستم اعداد وزن‌دار بر اساس تبدیل درهم مبنا[۲] ارائه شده است که با کمک آن می‌توان پیاده سازی مقایسه بزرگی اعداد، تشخیص سرریز و تشخیص علامت را انجام داد. در این الگوریتم از look up table ها نیز استفاده می‌شود و همین امر منجر به افزایش تاخیر در پیاده‌سازی این روش می‌گردد. در [۱۷] روشی برای شناسایی علامت اعداد مانده‌ای ارائه شده است که بر اساس الگوریتم تبدیل درهم مبنای ارائه شده در [۱۶] می‌باشد. در این روش از مقایسه‌گر استفاده می‌شود و تنها برای یک مجموعه پیمانه‌ی خاص از look up table استفاده نمی‌شود اما برای مجموعه پیمانه‌های دیگر نیازمند به استفاده از look up table هستیم و همین امر منجر به افزایش تاخیر در پیاده‌سازی این روش می‌گردد. متد جدیدی در [۱۸] ارائه شده است که از طریق آن شناسایی علامت و مقایسه بزرگی اعداد صورت می‌گیرد. این متد تکنیکی را بر اساس تبدیل درهم مبنا برای انتخاب مجموعه پیمانه ارائه می‌دهد به طوری که یک مجموعه پیمانه خاص با هر تعداد پیمانه‌ای انتخاب می‌گردد. بنابراین این روش به دلیل آن که محدودیت‌هایی را بر روی مجموعه پیمانه‌ها اعمال می‌کند، کمتر مورد استفاده قرار می‌گیرد. روش دیگری در [۱۹] برای تشخیص علامت اعداد مانده‌ای پیشنهاد شده است به طوری که بر اساس تئوری باقیمانده‌ی چینی جدید II[۳] می‌باشد. در این روش بایستی مجموعه پیمانه به صورت صعودی مرتب شده باشد و یکی از

---

[۱] Residue Number System(RNS)

[۲] Mixed Radix Conversion(MRC)

[۳] New Chinese Remainder Theorem II(New CRT II)

پیمانه‌های نیمه اول آن عددی زوج باشد. علاوه بر این، برای شناسایی علامت به یک مقایسه‌گر نیاز داریم. بنابراین این روش به دلیل استفاده از مقایسه‌گر و محدودیت‌هایی که بر روی پیمانه‌ها ایجاد می‌کند، روش مناسبی نمی‌باشد.

مدارهای شناساگر علامتی که تا کنون برای سیستم اعداد مانده‌ای معرفی شده‌اند اغلب برای یک مجموعه پیمانه‌ی سه تایی طراحی شده‌اند. ولی امروزه به دلیل نیاز به محدوده‌ی دینامیکی بالا و موازی سازی بیشتر، مجموعه پیمانه‌های چهارتایی مورد توجه قرار می‌گیرند. به همین دلیل در سال‌های اخیر تحقیقات زیادی بر روی شناسایی و معرفی مجموعه پیمانه‌های جدید و کارآمد برای سیستم اعداد مانده‌ای انجام شده است، که در پی آن مجموعه پیمانه‌ی چهارتایی جدید $\left(2^n-1,\ 2^n,\ 2^n+1,\ 2^{2n}+1\right)$برای این سیستم معرفی شده است. اخیراً عمده تحقیقات بر روی طراحی تبدیل معکوس برای این مجموعه پیمانه چهارتایی جدید بوده است در حالی که در صورتی که بخواهیم به صورت عملی از این مجموعه پیمانه در یک سیستم محاسباتی مبتنی بر سیستم اعداد مانده‌ای استفاده کنیم به مدارشناساگر علامت نیاز خواهیم داشت.

در این مقاله برای اولین بار طراحی مدار شناساگر علامت جدید و کارآمد برای سیستم اعداد مانده‌ای در مجموعه چهار پیمانه‌ای $\left(2^n-1,\ 2^n,\ 2^n+1,\ 2^{2n}+1\right)$با استفاده از قضیه‌ی باقیمانده‌ی چینی جدید II ارائه شده است تا در پی آن تشخیص علامت اعداد در این مجموعه پیمانه که منجر به محدوده دینامیکی بالا و موازی سازی بیشتر می‌گردد، امکان پذیر شود. از مزایای مدارپیشنهادی این است که بدون نیاز به مدار مقایسه‌گر و تنها با محاسبه n بیت تشخیص علامت صورت می‌گیرد. بنابراین مدار شناساگر علامت پیشنهادی دارای سرعت بالاتر و هزینه‌ی سخت افزاری پایین‌تری نسبت به روش پیشین می‌باشد.

در ادامه‌ی مقاله، پیش زمینه‌ی لازم در بخش ۲ بررسی می‌شود. سپس در بخش ۳ الگوریتم پیشنهادی تشخیص علامت اعداد مانده‌ای در مجموعه پیمانه $\left(2^n-1,\ 2^n,\ 2^n+1,\ 2^{2n}+1\right)$ارائه می‌شود. پیاده‌سازی سخت افزاری مدار شناساگر علامت پیشنهادی در بخش ۴ صورت می‌گیرد. در بخش ۵ ارزیابی کارآیی مدار شناساگر علامت پیشنهادی بر حسب تاخیر و هزینه‌ی سخت افزاری آن بررسی می‌شود و سپس مدار پیشنهادی با روش پیشین بر حسب گیت واحد از نظر تاخیر و هزینه‌ی سخت افزاری مقایسه می‌شود و در نهایت بخش ۶ نتیجه‌گیری می‌باشد.

۲- پیش زمینه

سیستم اعداد مانده‌ای یک سیستم عددی صحیح است که بر حسب تعدادی اعداد دو به دو نسبت به هم اول که مجموعه پیمانه را تشکیل می‌دهند تعریف می‌شود. در سیستم اعداد مانده‌ای با مجموعه پیمانه $\{P_1,P_2,....,P_n\}$، عدد وزنی X به صورت $X=\{x_1,x_2,...x_n\}$ نمایش داده می‌شود به طوری که[۱]:

$$x_i = X \bmod P_i = |X|_{P_i},\ \ 0 \le x_i < P_i \qquad (۱)$$

برای اعداد صحیح بدون علامت، هر عدد صحیح X در محدوده $[0, M-1]$ نمایش مانده‌ای منحصر به فردی دارد. به طوری که $M = P_1P_2 \dots P_n$ به عنوان محدوده دینامیکی مجموعه پیمانه $\{P_1,P_2,....,P_n\}$ نامیده می‌شود و برای اعداد صحیح علامت‌دار، عدد صحیح X زمانی که $0 \le X \le (M-1)/2$ باشد، مثبت می‌باشد و زمانی که $(M-1)/2 < X \le M-1$ باشد، منفی می‌باشد. بنابراین بر طبق تعاریف بالا تابع تشخیص علامت را به صورت زیر نمایش می‌دهیم [۶].

$$\mathrm{sgn}(X) = \begin{cases} 0 & \text{for} \quad \mathrm{X} \in \left[0,\ \left\lceil \frac{M}{2} \right\rceil\right) \\ 1 & \text{for} \quad \mathrm{X} \in \left[\left\lceil \frac{M}{2} \right\rceil,\ \mathrm{M}\right) \end{cases} \qquad (۲)$$

تبدیل یک عدد صحیح وزنی دودوئی به نمایش مانده‌ای، تبدیل مستقیم و عکس آن، یعنی تبدیل از نمایش مانده‌ای به نمایش وزنی دودوئی تبدیل معکوس (مبدل معکوس) نامیده می‌شود. الگوریتم‌های تبدیل معکوس عمدتاً بر پایه تئوری باقیمانده چینی، تبدیل درهم مبنا و تئوری باقیمانده چینی جدیدI و II هستند.

تئوری باقیمانده چینی جدید II: برای یک سیستم دو پیمانه‌ای (P_1, P_2) عدد صحیح X از طریق مانده‌ای‌های

(x_1, x_2) با استفاده از تئوری باقیمانده چینی جدید II به صورت فرمول (۳) محاسبه می‌شود:

(۳)

$$X = x_2 + \left| k_{2,1}(x_1 - x_2) \right|_{P_1} \cdot P_2$$

به طوری که $k_{2,1}$ معکوس ضربی می‌باشد و برابر $k_{2,1} = \left| P_2^{-1} \right|_{P_1}$ است و $X \in \left[0, P_1 \cdot P_2\right)$ می‌باشد.

۳- الگوریتم پیشنهادی تشخیص علامت اعداد مانده‌ای در $\left(2^n - 1,\ 2^n,\ 2^n + 1,\ 2^{2n} + 1\right)$:

مجموعه چهار پیمانه‌ای $\left(2^n - 1,\ 2^n,\ 2^n + 1,\ 2^{2n} + 1\right)$ را در اختیار داریم. محدوده دینامیکی آن برابر است با:

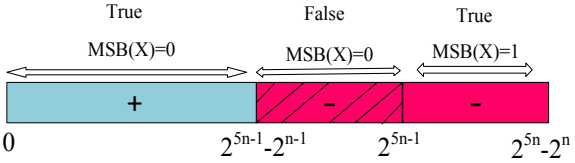
$$M = \left(2^n - 1\right) \times 2^n \times \left(2^n + 1\right) \times \left(2^{2n} + 1\right) = 2^{5n} - 2^n \quad (۴)$$

بنابراین همان گونه که در معادله (۴) مشاهده می‌کنید اعداد در این سیستم به صورت 5n بیتی نمایش داده می‌شوند (محدوده‌ی دینامیکی این مجموعه پیمانه 5n بیتی می‌باشد). هم اکنون بر طبق معادله (۲) تابع تشخیص علامت برای مجموعه پیمانه مورد نظر برابر با معادله (۵) است:

(۵)

$$\text{sgn}(X) = \begin{cases} 0, & \text{for} \quad X \in \left[0, 2^{5n-1} - 2^{n-1}\right) \\ 1, & \text{for} \quad X \in \left[2^{5n-1} - 2^{n-1}, 2^{5n} - 2^n\right) \end{cases}$$

با توجه به نمایش بایزی X، می‌توان محدوده دینامیکی این مجموعه پیمانه را به صورت شکل (۱) توصیف کرد:



شکل (۱) : نمایش بازه اعداد درمحدوده دینامیکی معرفی شده و چگونگی تغییر با ارزش ترین بیت اعداد با توجه به علامت آن‌ها

همان‌گونه که در شکل (۱) مشاهده می‌کنید در بازه مثبت (True) با ارزش‌ترین بیت در نمایش بایزی عددX نشان دهنده‌ی علامت X است اما در بازه منفی این‌گونه نیست. این بازه به دو زیر بازه‌۱- $X \in \left[2^{5n-1} - 2^{n-1}, 2^{5n-1}\right)$، $X \in \left[2^{5n-1}, 2^{5n} - 2^n\right)$-۲، تقسیم می‌شود:

(۶)

$$1)\ X \in \left[2^{5n-1} - 2^{n-1}, 2^{5n-1}\right) \Rightarrow \begin{bmatrix} {\scriptstyle 5n-1} & {\scriptstyle 5n-1} \\ 0X..X & ,10..00 \\ & {\scriptstyle 2^{5n-1}} \end{bmatrix}$$

(۷)

$$2)\ X \in \left[2^{5n-1}, 2^{5n} - 2^n\right) \Rightarrow \begin{bmatrix} {\scriptstyle 5n-1} & \\ 10...0,1\overbrace{XX...X} & \\ {\scriptstyle 2^{5n-1}} & \end{bmatrix}$$

در بازه ۱ (False) با ارزش‌ترین بیت اعدادی که در رنج $X \in \left[2^{5n-1} - 2^{n-1}, 2^{5n-1}\right)$ قرار گرفته‌اند برابر با "0" می‌باشد در حالی که اعداد ما منفی هستند. بنابراین در این بازه با ارزش‌ترین بیت در نمایش بایزی عددX، نشان دهنده‌ی علامت X نمی‌باشد.

در بازه ۲ (True) با ارزش‌ترین بیت اعدادی که در رنج $X \in \left[2^{5n-1}, 2^{5n} - 2^n\right)$ قرار گرفته‌اند برابر با "۱"می‌باشد که نشان دهنده‌ی علامت X است.

بنابراین از طریق MSB(X) نمی‌توانیم علامت اعداد را تشخیص دهیم. پس بایستی به دنبال راه حلی برای رفع این مشکل باشیم.

برای مجموعه پیمانه $M = \left(2^n - 1,\ 2^n,\ 2^n + 1,\ 2^{2n} + 1\right)$، به گونه‌ای که $x_i = \left| X \right|_{P_i}$ می‌باشد، ابتدا پیمانه‌های جدید زیر را ایجاد می‌کنیم:

(۸)

$$Q_1 = P_1 \cdot P_3 \cdot P_4 = (2^n - 1) \cdot (2^n + 1) \cdot (2^{2n} + 1) = 2^{4n} - 1$$

(۹)

$$Q_2 = P_2 = 2^n$$

ابتدا با کمک سه پیمانه‌ی P_1, P_3, P_4 پیمانه‌های زیر را تعریف می‌کنیم:

(۱۰)

$$Q_1 = \begin{cases} M_1 = P_1 \cdot P_3 = (2^{2n} - 1) \\ M_2 = P_4 = (2^{2n} + 1) \end{cases}$$

هم اکنون مجموعه دو پیمانه‌ای $Q_1=(\mathrm{M}_1,\mathrm{M}_2)$ را در نظر می‌گیریم. با توجه به معادله (۳) مانده‌های $m_i=|X|_{M_i}$ می‌باشند که به صورت زیر هستند:

(۱۱)

$$m_1=|X|_{M_1}=|X|_{P_1\cdot P_3}$$
$$=x_3+\left|k_{3,1}\cdot(x_1-x_3)\right|_{P_1}\cdot P_3$$
$$m_2=|X|_{M_2}=|X|_{P_4}=x_4$$

به طوری که:

$$k_{3,1}=\left|P_3^{-1}\right|_{P_1}=\left|(2^n+1)^{-1}\right|_{2^n-1}=2^{n-1}$$

برای به دست آوردن تبدیل معکوس مجموعه پیمانه Q_1 می‌توانیم از معادله (۳) استفاده کنیم و داریم:

(۱۲)

$X=m_2+\left|l_{2,1}(m_1-m_2)\right|_{M_1}\cdot M_2$ بنابراین:

$$l_{2,1}=\left|M_2^{-1}\right|_{M_1}=\left|(2^{2n}+1)^{-1}\right|_{2^{2n}-1}=2^{2n-1}$$

با جایگذاری مقادیر (۱۰) و (۱۱) در (۱۲) تبدیل معکوس Q_1، را محاسبه می‌کنیم:

(۱۳)

$$X_{Q_1}=x_4+\left|2^{2n-1}\left(x_3-x_4+\left|2^{n-1}(x_1-x_3)\right|_{2^n-1}\right)\right|_{2^{2n}-1}\cdot(2^{2n}+1)$$

هم اکنون بایستی تبدیل معکوس Q_1 را همراه با Q_2 محاسبه کنیم.با توجه به معادله (۳) مانده‌های $mm_i=|X|_{Q_i}$ می‌باشند. بنابراین داریم:

(۱۴)

$$mm_1=|X|_{Q_1}=x_4+\left|2^{2n-1}\left(x_3-x_4+\left|2^{n-1}(x_1-x_3)\right|_{2^n-1}\cdot(2^n+1)\right)\right|_{2^{2n}-1}\cdot(2^{2n}+1)$$

$$mm_2=|X|_{Q_2}=|X|_{P_2}=x_2$$

حال برای به دست آوردن تبدیل معکوس مجموعه پیمانه $M=\left(2^{\mathrm{n}}-1,\ 2^{\mathrm{n}},\ 2^{\mathrm{n}}+1,\ 2^{2\mathrm{n}}+1\right)$ از معادله (۳) استفاده می‌کنیم و داریم:

(۱۵)

$$X=mm_2+\left|I_{2,1}(mm_1-mm_2)\right|_{Q_1}\cdot Q_2$$

به طوری که: $I_{2,1}=\left|Q_2^{-1}\right|_{Q_1}=\left|(2^n)^{-1}\right|_{2^{4n}-1}=2^{3n}$

مقادیر (۸) و (۹) و (۱۴) را در (۱۵) جایگذاری می‌کنیم بنابراین:

(۱۶)

$$X=x_2+\left|2^{3n}\left(x_4-x_2+\left|2^{2n-1}\left(x_3-x_4+\left|2^{n-1}(x_1-x_3)\right|_{2^n-1}\cdot(2^n+1)\right)\right|_{2^{2n}-1}\cdot(2^{2n}+1)\right)\right|_{2^{4n}-1}\cdot 2^n$$

برای سهولت کار سمبل‌های زیر را معرفی می‌کنیم:

(۱۷)

$$Y=\left|2^{2n-1}\left(x_3-x_4+\left|2^{n-1}(x_1-x_3)\right|_{2^n-1}\cdot(2^n+1)\right)\right|_{2^{2n}-1}$$

$$S=x_4-x_2+Y\cdot(2^{2n}+1) \qquad (۱۸)$$

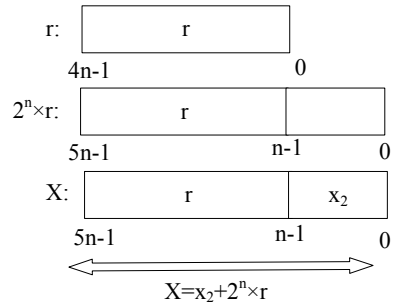
(۱۹)

$$r=\left|2^{3n}\cdot S\right|_{2^{4n}-1}$$

(۲۰)

$$X=x_2+2^n\cdot r$$

همان‌گونه که در شکل (۲) مشاهده می‌کنید، بر طبق $X=x_2+2^n\cdot r$، 4n بیت پر ارزش X به r بستگی دارد.



شکل(۲): نمایش بیتی معادله X

بنابراین:

$$\mathrm{MSB(X)=MSB(r)}$$

از مزایای الگوریتم پیشنهادی این است که نیازی نیست تمام 4n بیت r محاسبه شود بلکه تنها با محاسبه n بیت تشخیص علامت امکان پذیر است.

معادله (۱۹) را می‌توان به صورت زیر بازنویسی کرد:

(۲۱)

$$r = \left|2^{3n}\cdot S\right|_{2^{4n}-1} = \left|2^{3n}\cdot \left|S\right|_{2^{4n}-1}\right|_{2^{4n}-1}$$

(۲۲)

$$Z = \left|S\right|_{2^{4n}-1} = \mathrm{S+c}$$

c ضریبی از $(2^{4n}-1)$ می‌باشد و برابر است با:

(۲۳)

$$\mathrm{c} = -(2^{4n}-1)\left\lfloor \frac{S}{(2^{4n}-1)} \right\rfloor$$

از آن جایی که Z یک عدد مانده‌ای به پیمانه $(2^{4n}-1)$ می‌باشد، بنابراین به صورت یک عدد باینری 4n بیتی $Z_{4n-1}, Z_{4n-2}, \ldots\ldots, Z_1, Z_0$ نمایش داده می‌شود. حال معادله (۲۱) از طریق 3n بیت شیفت چرخشی به چپ Z قابل محاسبه می‌باشد. بنابراین بر طبق (۲۱) و (۲۲)، $\mathrm{MSB(r)} = \mathrm{Z_{n-1}}$ می‌گردد. در ادامه معادله (۱۸) و (۲۳) را در (۲۲) جایگذاری کرده و داریم:

(۲۴)

$$Z = S + c = x_4 - x_2 + Y\cdot(2^{2n}+1) + c$$
$$= x_4 - x_2 + Y\cdot 2^{2n} + Y + c$$

حال n بیت کم ارزش Z از طریق فرمول زیر محاسبه می‌شود:

(۲۵)

$$t = x_4 - x_2 + Y + \left|c\right|_{2^n} = t' + \left|c\right|_{2^n}$$

اگر c وجود داشته باشد بیت $t_{n-1} = Z_{n-1}$ با کمک عملیات به پهنای n بیت می‌تواند محاسبه گردد، بنابراین داریم:

(۲۶)

$$MSB(X) = MSB(r) = MSB(\left|t\right|_{2^n})$$

در ادامه خواهیم دیدکه تابع تشخیص علامت پیشنهادی از طریق معادله (۲۶) با جایگذاری t' به جای t استنتاج می‌گردد:

$$t' = x_4 - x_2 + Y \qquad (۲۷)$$

ابتدا بر طبق معادله (۱۷) مقدار Y در بازه‌ی $\left[0, 2^{2n}-1\right)$ می‌باشد. حال با کمک Y، بازه‌ی اختصاص یافته به معادله (۱۸) یعنی S را می‌یابیم، بنابراین داریم:

1)if $0<Y<2^{2n}-1$ then $S\in\left[2^{2n}-2^n+2,\ 2^{4n}-2\right]$

$$\begin{aligned} S &\le x_{4,\max} - x_{2,\min} + Y_{\max}(2^{2n}+1) \\ &\le 2^{2n} - 0 + (2^{2n}-2)(2^{2n}+1) \\ &\le 2^{4n} - 2 \end{aligned}$$

$$\begin{aligned} S &\ge x_{4,\min} - x_{2,\max} + Y_{\min}(2^{2n}+1) \\ &\ge 0 - (2^n-1) + 1\times(2^{2n}+1) \\ &\ge 2^{2n} - 2^n + 2 \end{aligned}$$

2)if $Y = 0$ then $S\in\left(-2^n, 2^{2n}\right]$

a) $S \le x_{4,\max} - x_{2,\min}$

$$\le 2^{2n} - 0 \ \rightarrow 0 \le S \le 2^{2n}$$

b) $S \ge x_{4,\min} - x_{2,\max}$

$$\ge 0 - (2^n - 1)$$
$$\ge -2^n + 1 \ \rightarrow\ -2^n < S < 0$$

در حالت 1 و در حالت 2a مقدار c بر طبق معادله (۲۳) برابر با صفر می‌باشد. بنابراین در این حالات طبق فرمول (۲۵)، $t = t'$ می‌گردد و در نهایت بر طبق فرمول (۲۶)، $MSB(X) = MSB\left(\left|t'\right|_{2^n}\right)$ می‌باشد و علامت حاصل می‌شود. اما در حالت 2b (بازه False) مقدار c مخالف با صفر می‌باشد و برابر است با :

(۲۸)

$$\mathrm{c} = -(2^{4n}-1)\underbrace{\left\lfloor \frac{S}{(2^{4n}-1)} \right\rfloor}_{-1} = (2^{4n}-1)$$

از آن جایی که $\left|c\right|_{2^n} = \left|2^{4n}-1\right|_{2^n} = \left|-1\right|_{2^n}$ است، بنابراین در حالت 2b بر طبق فرمول(۲۵)، $t = t' - 1$ می‌گردد.

یک تفاوت بین فرم‌های باینری t و t' در یک دنباله‌ای از صفرهای کم ارزش و اولین یک از راست می‌باشد یعنی $\mathrm{xxx1}\underbrace{00\ldots00}_{m}$ و $\underbrace{011\ldots11}_{m}$ است.

بنابراین بر طبق مطالب بیان شده شرط $MSB\left(\left|t\right|_{2^n}\right) \neq MSB\left(\left|t'\right|_{2^n}\right)$ تنها زمانی رخ خواهد داد که

$t' = x_4 - x_2 = 1..100..0$ یا $t' = x_4 - x_2 = 1..1100..0$
n n
باشد.

اولین حالت غیر ممکن است یعنی:

(۲۹)

$$x_4 \geq 0 \ \cap \ x_2 < 2^n \Rightarrow x_4 - x_2 \neq -2^n$$

دومین حالت نتیجه‌ی زیر را می‌دهد:

(۳۰)

$$x_4 - x_2 = -2^{n-1} \quad \Rightarrow x_2 \in \left[2^{n-1}, 2^n - 1\right]$$

با توجه به مطالب بیان شده نتیجه می‌گیریم برای تمام X ها، $MSB(X) = MSB\left(|t'|_{2^n}\right)$ می‌باشد به جز برای حالتی که $Y = 0$ و $c \neq 0$ و $x_4 - x_2 = -2^{n-1}$ باشد (حالت 2b)، در این حالت $MSB(X) \neq MSB\left(|t'|_{2^n}\right)$ است. در ادامه ثابت می‌کنیم با توجه به بازه x_2 در فرمول (۳۰) و هم‌چنین مقدار $x_4 - x_2 = -2^{n-1}$ و $Y = 0$، بازه‌ی X هایی که در آن‌ها $MSB(X) \neq MSB\left(|t'|_{2^n}\right)$ می‌باشد، با جایگذاری معادلات (۱۹) و (۱۸) در (۲۰) به صورت زیر محاسبه می‌شود:

(۳۱)

$$\begin{aligned} X &= x_2 + 2^n \cdot r = x_2 + 2^n \cdot \left|2^{3n} \cdot S\right|_{2^{4n}-1} \\ &= x_2 + 2^n \cdot \left|2^{3n} \cdot (x_4 - x_2 + \underbrace{Y \times (2^{2n} - 1)}_{0})\right|_{2^{4n}-1} \\ &= x_2 + 2^n \cdot \left|2^{3n} \cdot (x_4 - x_2)\right|_{2^{4n}-1} \\ &= x_2 + 2^n \cdot \left|2^{3n} \cdot (-2)^{n-1}\right|_{2^{4n}-1} \\ &= x_2 + 2^{5n-1} - 2^n \end{aligned}$$

حال بازه‌ی x_2 به دست آمده از معادله (۳۰) را در فرمول بالاجایگذاری کرده وبازه X را محاسبه می‌کنیم:

$$X \in \left[2^{5n-1} - 2^{n-1}, \ 2^{5n-1} - 1\right] \qquad (۳۲)$$

X هایی که در بازه بالا صدق می‌کنند دقیقاً منطبق با بازه‌ی (۶) یعنی بازه False ما می‌باشند و همان‌گونه که بیان شد در این X ها $MSB(X) \neq MSB\left(|t'|_{2^n}\right)$ می‌باشد. بنابراین همین امر سبب می‌شود از $MSB(|t'|_{2^n})$ برای تشخیص علامت استفاده شود تا بدین ترتیب مشکل بازه‌ی False نیز برطرف شود.

در نتیجه تابع تشخیص علامت پیشنهادی برای مجموعه پیمانه $M = \left(2^n - 1, \ 2^n, \ 2^n + 1, \ 2^{2n} + 1\right)$ به صورت زیر تعریف می‌شود:

$$\text{n detection}\left(x_1, x_2, x_3, x_4\right) = MSB\left(|t'|_{2^n}\right) \qquad (۳۳)$$

به عنوان مثال مجموعه $\left(2^n - 1, \ 2^n, \ 2^n + 1, \ 2^{2n} + 1\right)$ را در نظر می‌گیریم و فرض می‌کنیم $n = 2$ باشد. بنابراین مجموعه پیمانه ما به صورت $\{3, 4, 5, 17\}$ می‌باشد. حال برای مجموعه پیمانه مذکور تابع تشخیص علامت پیشنهادی را محاسبه می‌کنیم.

$$M = 3 \times 4 \times 5 \times 17 = 1020$$

$$\text{sgn}(X) = \begin{cases} 0 & \text{for} \quad X \in \left[0, \ \left\lceil \frac{1020}{2} \right\rceil\right) \\ 1 & \text{for} \quad X \in \left[\left\lceil \frac{1020}{2} \right\rceil, \ 1020\right) \end{cases}$$

جدول (۱): محاسبه تابع تشخیص علامت پیشنهادی برای مجموعه پیمانه‌ی $\{3, 4, 5, 17\}$

X	x_{RNS}	Y	t'	t'_{bit}
0	(0,0,0,0)	0	0	0000
509	(2,1,4,16)	14	29	1110
510	(0,2,0,0)	0	-2	1111
511	(1,3,1,1)	0	-2	1111
512	(2,0,2,2)	0	2	0001
514	(1,2,4,4)	0	2	0001
1017	(0,1,2,14)	14	27	1101
1018	(1,2,3,15)	14	27	1101
1019	(2,3,4,16)	14	27	1101

همان‌گونه که در مثال بالا نشان داده شده است تابع تشخیص علامت اعداد مانده‌ای پیشنهادی در مجموعه پیمانه $\{3, 4, 5, 17\}$ برابر با یکمین بیت t' با شروع از صفر می‌باشد.

۴- پیاده‌سازی سخت افزاری مدار شناساگر علامت پیشنهادی:

تابع تشخیص علامت پیشنهادی $(n-1)$امین بیت t' (با شروع از صفر) است. مقدار t' که توسط فرمول (۲۷) به دست می‌آید $(2n+1)$بیتی است و برابر با مجموع $-x_2$و x_4 و Yاست. بنابراین چون ما به $(n-1)$امین بیت t'نیاز داریم

لم۱: ضرب یک عدد مانده‌ای v در عدد 2^p و کاهش آن در پیمانه (2^k-1) به طوری که p و k اعداد صحیح و مثبتی باشند، برابر با p بیت شیفت چرخشی به چپ v می‌باشد [۲۰].

لم۲: کاهش پیمانه‌ای یک عدد منفی در پیمانه (2^k-1) برابر با مکمل یک آن عدد می‌باشد [۲۰].

مجموعه پیمانه $M=\left(2^n-1,2^n,2^n+1,2^{2n}+1\right)$را در اختیار داریم. اعداد مانده‌ای نظیر به نظیر عدد Xبا (x_1,x_2,x_3,x_4)نمایش داده می‌شوند. به طوری که x_1، nبیتی و x_2، nبیتی و x_3، n+1بیتی و x_4، 2n+1بیتی می‌باشد.

در ابتدا فرمول (۱۷) را به صورت زیر ساده سازی می‌کنیم.

(۳۵)

$$Y=\left|2^{2n-1}\left(\underbrace{\left|2^{n-1}(x_1-x_2)\right|}_{K}\cdot(2^n+1)\right)\right|_{2^{2n}-1}$$

$$\mathrm{K}=\left|2^{n-1}(x_1-x_3)\right|_{2^n-1}=\left|v_1+v_2\right|_{2^n-1} \qquad (۳۶)$$

(۳۷)

$$v_1=\left|2^{n-1}x_1\right|_{2^n-1}=\left|2^{n-1}(\underbrace{x_{\cdot}\ \ x_{\cdot}}_{n})\right|_{2^n-1}$$

$$=\underbrace{x_{\cdot}\ x_{\cdot}\ \ x_{\cdot}\ \ x_{\cdot}}_{n}$$

از وزن‌های بالاتر از 2^{n-1}چشم پوشی می‌کنیم. در نتیجه t'، n بیتی برای تشخیص علامت کافی می‌باشد. در ادامه nبیت کم ارزش t'راپیاده سازی می‌کنیم.

(۳۴)

$$t'=\left|x_4\right|_{2^n}-x_2+\left|Y\right|_{2^n}=x_4^*-x_2+Y^*$$

از طریق دو لم شناخته شده‌ی زیر می‌توانیم معادله (۳۴) را ساده سازی نماییم.

(۳۸)

$$v_2=\left|-2^{n-1}x_3\right|_{2^n-1}=\left|-2^{n-1}(\underbrace{x_{\cdot}\ \ x_{\cdot}}_{n+1})\right|_{2^n-1}$$

$$=\left|-2^{n-1}(x_{3,n}\times 2^n+\underbrace{x_{\cdot}\ \ x_{\cdot}\ \ x_{\cdot}}_{n})\right|_{2^n-1}$$

به دلیل این که x_3عددی است که از 2^n+1 کوچکتر می‌باشد، می‌توانیم دو حالت برای آن در نظر بگیریم . حالت اول زمانی است که x_3کوچکتر از 2^nاست و دومین حالت زمانی که برابر با 2^nاست.

اگر $x_{3,n}=0$ باشد بنابراین داریم:

(۳۹)

$$v_{21}=\left|-2^{n-1}(\underbrace{x_{\cdot}\ \ x_{\cdot}\ \ x_{\cdot}}_{n})\right|_{2^n-1}$$

$$=\underbrace{\overline{x_{\cdot}}\ \overline{x_{\cdot}}\ \ \overline{x_{\cdot}}}_{n}$$

در غیر این صورت اگر $x_{3,n}=1$ باشد خواهیم داشت:

(۴۰)

$$v_{22}=\left|-2^{n-1}\times 2^n(\underset{n-1}{0..0}\ \ x_{3,n})\right|_{2^n-1}=\underset{n-1}{01...1}$$

بنابراین v_2به صورت زیر محاسبه می‌شود:

(۴۱)

$$v_2=\begin{cases}v_{21} & \text{if} \quad x_{3,n}=0\\ v_{22} & \text{if} \quad x_{3,n}=1\end{cases}$$

در نهایت (۳۵) به صورت زیر ساده سازی می‌شود:

(۴۲)

$$Y = \left| 2^{2n-1}(x_3 - x_4 + k(2^n + 1)) \right|_{2^{2n}-1}$$

معادله بالا به صورت زیر مورد ارزیابی قرار می‌گیرد:

(۴۳)

$$v_3 = \left| 2^{2n-1} x_3 \right|_{2^{2n}-1} = \left| 2^{2n-1}(0\ldots0\underbrace{x_{\cdot} \quad x_{\cdot\cdot}}_{n+1}) \right|_{2^{2n}-1}$$

(۴۴)

$$= x_{3,0}\, 0\ldots00\underbrace{x_{\cdot} \quad x_{\cdot\cdot}}_{n}$$

$$v_4 = \left| -2^{2n-1} x_4 \right|_{2^{2n}-1} = \left| -2^{2n-1}(\underbrace{x_{\cdot\cdot} \quad x_{\cdot\cdot}}_{2n+1}) \right|_{2^{2n}-1}$$

$$= \left| -2^{2n-1}(x_{4,2n} \times 2^{2n} + \underbrace{x_{\cdot\cdot} \quad x_{\cdot\cdot}}_{2n}) \right|_{2^{2n}-1}$$

(۴۷)

$$v_4 = \begin{cases} v_{41} & \text{if} \quad x_{4,2n} = 0 \\ v_{42} & \text{if} \quad x_{4,2n} = 1 \end{cases}$$

(۴۸)

$$v_5 = \left| \begin{matrix} 2^{2n-1} \times \\ (2^n + 1)k \end{matrix} \right|_{2^{2n}-1} = \left| \begin{matrix} 2^{2n-1} \times \\ (2^n + 1)(\underbrace{1 \quad 1}_{n}) \end{matrix} \right|_{2^{2n}-1}$$

$$= k_0 \underbrace{1 \quad 1 \quad 1 \quad 1}_{n} \quad \underbrace{1}_{n-1}$$

در نتیجه معادله (۴۲) به صورت زیر محاسبه می‌شود:

(۴۹)

$$Y = \left| v_3 + v_4 + v_5 \right|_{2^{2n}-1}$$

پیاده سازی سخت افزاری معادله (۴۹) در شکل (۳) نشان داده شده است.

به دلیل این که x_4 عددی است که از $2^{2n}+1$ کوچکتر می‌باشد، می‌توانیم دو حالت برای آن در نظر بگیریم. حالت اول زمانی است که x_4 کوچکتر از 2^{2n} است و دومین حالت زمانی که برابر با 2^{2n} است.

اگر $x_{4,2n} = 0$ باشد آن‌گاه داریم:

(۴۵)

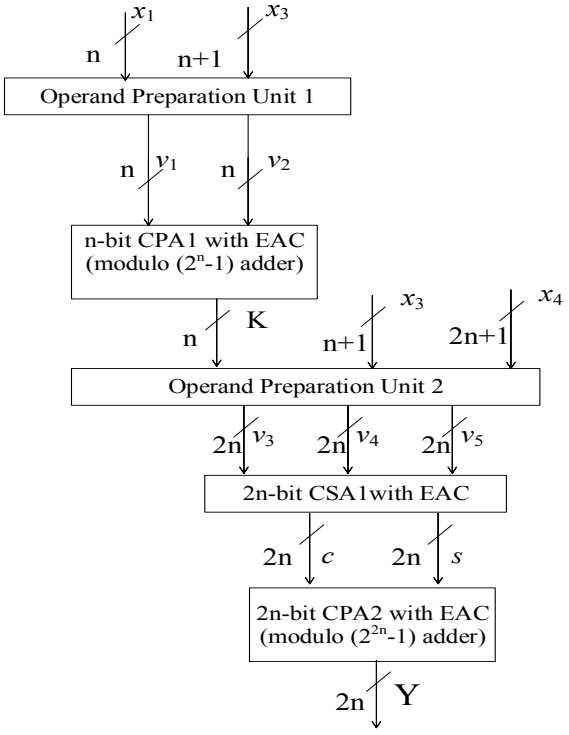
$$v_{41} = \underbrace{\overline{x_{\cdot\cdot}} \quad \overline{x_{\cdot\cdot}} \quad \overline{x_{\cdot\cdot}}}_{2n}$$

درغیر این صورت اگر $x_{4,2n} = 1$ باشد خواهیم داشت:

(۴۶)

$$v_{42} = \left| -2^{2n-1} \times 2^{2n}(\underbrace{\cdot \quad \cdot\cdot}_{2n-1} \quad \cdot \right|_{2^{2n}-1} = \underbrace{01..11}_{2n-1}$$

بنابراین v_4 به صورت زیر محاسبه می‌شود:



شکل(۳): پیاده سازی Y

پیاده سازی سخت افزاری مدار شناساگر علامت پیشنهادی در مجموعه چهار پیمانه‌ای $\left(2^n - 1, 2^n, 2^n + 1, 2^{2n} + 1\right)$ با مانده‌ای‌های نظیر به نظیر (x_1, x_2, x_3, x_4) در شکل (۴) نشان داده شده است. به طوری که داریم:

(۵۰)

$$t' = x_4^* - x_2 + Y^*$$

(۵۱)

$$x_4^* = |x_4|_{2^n} = \left| \underbrace{(x_{2n} \; x_{2n-1} \; \cdots \; x_0)}_{2n+1} \right|_{2^n}$$

$$= \left| \left(\underbrace{x_{2n} \cdots x_n}_{n+1} \times 2^n + \underbrace{x_{n-1} \cdots x_1 x_0}_{n} \right) \right|_{2^n} = \underbrace{x_{n-1} \cdots x_0}_{n}$$

(۵۲)

$$Y^* = |Y|_{2^n} = \left|(Y_{2n-1}Y_{2n-2}\ldots Y_0)\right|_{2^n}$$

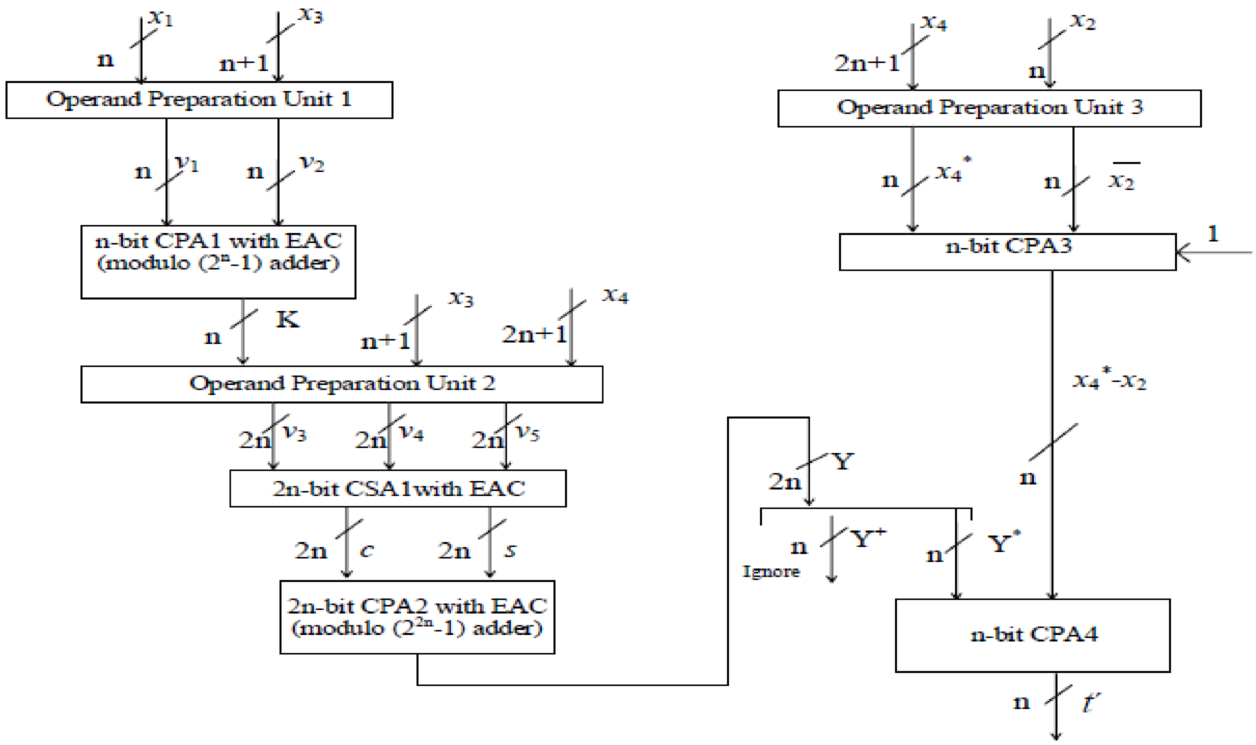
$$= \left| \left(Y_{2n-1} \ldots Y_n \times 2^n + \underbrace{Y_{n-1} \quad Y_0}_{n} \right) \right|_{2^n} = Y_{n-1} \ldots Y_1 Y_0$$

(۵۳)

$$-x_2 = \overline{x_2} + 1$$

(۵۴)

$$\overline{x_2} = \underbrace{\overline{x_{n-1}} \; \overline{x_{n-2}} \; \cdots \; \overline{x_1} \; \overline{x_0}}_{n \text{ bit}}$$



شکل (۴): پیاده سازی سخت افزار شناساگر علامت پیشنهادی برای مجموعه پیمانه $\left(2^n-1, 2^n, 2^n+1, 2^{2n}+1\right)$

هزینه سخت افزاری و تاخیری که برای هر قسمت از مدار شناساگر علامت پیشنهادی شکل (۴) اختصاص می‌یابد در جدول (۲) نشان داده شده است.

جدول (۲): ویژگی هر قسمت از شناساگر علامت پیشنهادی در مجموعه پیمانه $\left(2^n-1, 2^n, 2^n+1, 2^{2n}+1\right)$

Parts	FA	NOT	XOR/AND Pairs	MUX 2×1	Delay
OPU1	–	n	–	1(n)-bit	$t_{NOT}+t_{MUX}$
CPA1	n	–	–	–	$(2n)t_{FA}$
OPU2	–	2n	–	1(2n)-bit	$t_{NOT}+t_{MUX}$
CSA1	n+1	–	n-1	–	t_{FA}
CPA2	2n	–	–	–	$(4n)t_{FA}$
OPU3	–	n	–	–	t_{NOT}
CPA3	n	–	–	–	$(n)t_{FA}$
CPA4	n	–	–	–	$(n)t_{FA}$

باید در نظر داشته باشیم که برای محاسبه تاخیرکلی، OPU3 و CPA3 موازی با OPU1 و CPA1 اجرا می‌شوند.

پیاده سازی سخت افزار مدار شناساگر علامت اعداد مانده‌ای برای مجموعه پیمانه

$\left(2^n-1, 2^n, 2^n+1, 2^{2n}+1\right)$ بر اساس (۳۶)، (۴۹)، (۵۱)، (۵۲)، (۵۳) و (۵۴) می‌باشد. در ابتدا واحد آماده سازی عملوندهای ۱[1] عملوندهای مورد نیاز (۳۷) و (۳۸) را آماده می‌کند. این آماده سازی از طریق چرخش بیت‌های اعداد مانده‌ای صورت می‌گیرد. هم‌چنین n گیت NOT برای اجرای معکوس‌های بیتی مورد نیاز در (۳۹) در نظر گرفته می‌شود. علاوه بر این به یک مالتی پلکسر ۱×۲، n بیتی برای (۴۱) نیاز داریم. محاسبه (۳۶) نیاز به یک جمع کننده انتشار رقم نقلی n بیتی با گردش رقم نقلی انتهایی[2] دارد. این جمع کننده‌های پیمانه‌ای با شیوه‌های متفاوتی پیاده سازی می‌شوند. در این مقاله CPA with EAC، [۲۱] را در نظر می‌گیریم. تاخیر CPA with EAC دو برابر تاخیر یک CPA عادی می‌باشد. اما از نظر پیچیدگی سخت افزاری هر دو جمع کننده با یکدیگر مشابه می‌باشند. بنابراین CPA1 with EAC دارای پیچیدگی سخت افزاری n عدد تمام جمع کننده و تاخیر انتشار $(2n)t_{FA}$ است که t_{FA} نشان دهنده تاخیر یک تمام جمع کننده است. علاوه بر این واحد آماده سازی عملوندهای ۲ عملوندهای مورد نیاز (۴۳)، (۴۷) و (۴۸) را آماده می‌کند. این آماده سازی توسط چرخش بیت‌های اعداد مانده‌ای انجام می‌شود و علاوه بر این 2n گیت NOT برای اجرای معکوس‌های بیتی در (۴۵) مورد نیاز است. هم‌چنین به یک مالتی پلکسر ۱×۲، 2n بیتی برای (۴۷) نیاز داریم. محاسبه (۴۹) نیاز به یک جمع کننده ذخیره رقم نقلی 2n بیتی با گردش رقم نقلی انتهایی[3] و یک 2n-bit CPA2 with EAC دارد. از آن جایی که (۴۳) دارای (n-1) بیت صفر می‌باشد، بنابراین (n-1) تمام جمع کننده در CSA1 به گیت‌های XOR/AND کاهش می‌یابد. معادله (۵۲) برابر با n بیت کم ارزش Y است و یکی از ورودی‌های n-bit CPA4 می‌باشد. Y^+ نیز برابر با n بیت پر ارزش Y است که از آن چشم پوشی می‌گردد. سپس برای محاسبه $x_4^* - x_2$ به یک تفریق کننده باینری n بیتی نیاز داریم. این تفریق کننده از طریق n گیت NOT و n عدد تمام جمع کننده به دست می‌آید. در نهایت برای محاسبه (۵۰) به یک n-bit CPA4 که دارای پیچیدگی سخت افزاری n عدد تمام جمع کننده و هم‌چنین دارای تاخیر انتشار $(n)t_{FA}$ است نیاز داریم.

۵- ارزیابی کارآیی

در این مقاله مدار شناساگر علامت اعداد مانده‌ای برای مجموعه پیمانه $\left(2^n-1, 2^n, 2^n+1, 2^{2n}+1\right)$ طراحی شده است. برای بررسی کارآیی مدار پیشنهادی، آن را از نظر تاخیر و هزینه سخت افزاری با روش پیشین مقایسه کرده‌ایم.

در [۱۷] الگوریتمی عمومی برای شناسایی علامت اعداد مانده‌ای ارائه شده است به طوری که از مدار مقایسه‌گر برای تشخیص علامت استفاده می‌کند. پیاده‌سازی سخت افزاری مجموعه $\left(2^n-1, 2^n, 2^n+1, 2^{2n}+1\right)$ از طریق این روش به فرم 2^k+1 می‌باشد و همین امر منجر به بالا رفتن هزینه سخت افزاری و تاخیر می‌گردد. در [۱۸] مدار دیگری معرفی شده به طوری که در این طرح، پیمانه‌ی آخر مجموعه پیمانه، بایستی حتماً برابر با ۲ باشد در نتیجه نمی‌توانیم به شناسایی علامت مجموعه پیمانه پیشنهادی از طریق طرح ارائه شده در [۱۸] بپردازیم. الگوریتمی که در [۱۹] معرفی شده است برای شناسایی علامت به مدار مقایسه‌گر نیاز دارد. در نتیجه شناسایی علامت مجموعه‌ی چهار پیمانه‌ای $\left(2^n-1, 2^n, 2^n+1, 2^{2n}+1\right)$ از طریق روش مذکور دارای تاخیر و سخت افزار بیشتری می‌باشد.

۵-۱ مقایسه سخت افزار مورد نیاز و تاخیر مدار شناساگر علامت پیشنهادی با روش پیشین

مدار شناساگر علامت $\left(2^n-1, 2^n, 2^n+1, 2^{2n}+1\right)$ را با [۲۲] مقایسه می‌کنیم. در [۲۲] مبدل مانده‌ای به باینری برای مجموعه پیمانه $\left(2^n-1, 2^n, 2^n+1, 2^{2n}+1\right)$ با استفاده از قضیه باقیمانده چینی جدید I ارائه شده است. سپس برای تشخیص علامت، خروجی حاصل از این مبدل از طریق یک مدار مقایسه‌گر، با $\frac{1}{2}$ محدوده‌ی دینامیکی مقایسه شده و سپس علامت اعداد مانده‌ای مشخص می‌شود. روش پیشین همان گونه که در جدول (۳) نشان داده شده است

---

[1] Operand Preparation Unit 1 (OPU1)

[2] n-bit Carry Propagate Adder 1 with End-Around Carry(n-bit CPA1with EAC)

[3] 2n-bit Carry Save Adder 1 with End-Around Carry(2n-bit CSA1 with EAC)

دارای سخت افزار و تاخیر بسیار زیادی در مقایسه با روش پیشنهادی مقاله می باشد.

۵-۲ مقایسه هزینه‌ی سخت افزاری و تاخیر گیت واحد مدار شناساگر علامت پیشنهادی با روش پیشین

مدل گیت واحد [۲۳] برای تخمین هزینه سخت افزاری و تاخیر مورد استفاده قرار می‌گیرد. در این مدل هر گیت دو ورودی به عنوان یک گیت در هزینه سخت افزاری و تاخیر لحاظ می‌شود. هم‌چنین یک گیت XOR/XNOR به عنوان دو گیت در هزینه سخت افزاری و تاخیر در نظر گرفته می‌شود و $\left(2^n-1,\ 2^n,\ 2^n+1,\ 2^{2n}+1\right)$ دارای سرعت بالاتر و هزینه سخت افزاری بسیار کمتری در مقایسه با [۲۲] می‌باشد. در جدول (۵) محدوده‌های دینامیکی متفاوت را از طریق مقداردهی‌های مختلف n به دست آوردیم. در این جدول محدوده‌ی دینامیکی $\left(2^n-1,\ 2^n,\ 2^n+1,\ 2^{2n}+1\right)$ ، 5n بیتی می‌باشد. بنابراین با استفاده از مقادیر متفاوت $n=2,4,7,...$ محدوده‌های دینامیکی مورد نظر را ایجاد یک تمام جمع کننده دارای هفت گیت هزینه سخت افزاری و چهار گیت تاخیر می‌باشد و نهایتاً مالتی پلکسر ۱×۲ به ترتیب به عنوان سه گیت و دو گیت در هزینه سخت افزاری و تاخیر محسوب می‌گردد. در ادامه در جدول (۴) تخمین کلی هزینه سخت افزاری و تاخیر گیت واحد برای مدار شناساگر علامت پیشنهادی و روش پیشین ارائه شده است. نتایج ارائه شده در این جدول بر اساس مقادیر به دست آمده در جدول (۳) و بر مبنای مدل گیت واحد می‌باشند. همان‌گونه که در جدول (۴) مشاهده می‌کنید مدار شناساگر علامت پیشنهادی می‌کنیم. حال با استفاده از nهای به دست آمده هزینه سخت افزاری و تاخیر گیت واحد مدار شناساگر علامت پیشنهادی و روش پیشین را در محدوده‌های دینامیکی متفاوت محاسبه می‌کنیم. هم اکنون با کمک مقادیر به دست آمده در جدول (۵) هزینه سخت افزاری و تاخیر گیت واحد مدار شناساگر علامت پیشنهادی و روش پیشین را به ترتیب در شکل (۵) و (۶) به صورت نمودار نشان می‌دهیم.

جدول (۳): مقایسه سخت افزار مورد نیاز و تاخیر مدار شناساگر علامت پیشنهادی با روش پیشین

Sign detector	Moduli set	Area	comparator	Delay
Proposed	$(2^n-1,\ 2^n,\ 2^n+1,\ 2^{2n}+1)$	$(6n+1)A_{FA}+(n-1)A_{XOR}+(n-1)A_{AND}+(4n)A_{NOT}+(n)A_{MUX2\times1}+(2n)A_{MUX2\times1}$	*no*	$(7n+1)t_{FA}+2t_{NOT}+2t_{MUX}$
[22]	$(2^n-1,\ 2^n,\ 2^n+1,\ 2^{2n}+1)$	$(11n+6)A_{FA}+(2n-1)A_{XOR}+(2n-1)A_{AND}+(4n)A_{XNOR}+(4n)A_{OR}+(5n+3)A_{NOT}+A_{Comparator}$	*yes*	$(8n+3)t_{FA}+t_{NOT}+t_{comparator}$

جدول (۴): مقایسه هزینه سخت افزاری و تاخیر گیت واحد شناساگرعلامت پیشنهادی با روش پیشین

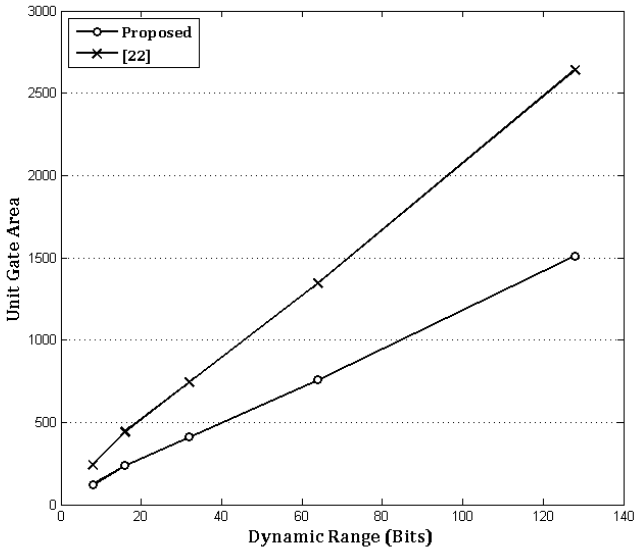
Sign detector	Proposed	[22]
Moduli set	$(2^n-1,\ 2^n,\ 2^n+1,\ 2^{2n}+1)$	$(2^n-1,\ 2^n,\ 2^n+1,\ 2^{2n}+1)$
Unit gate area	$58n+4$	$100n+42+A^*_{comparator}$
Unit gate delay	$28n+10$	$32n+13+t^*_{comparator}$

* نشان دهنده هزینه سخت افزاری و تاخیر مقایسه کننده می‌باشد که در جدول بالا در نظر گرفته نشده است.

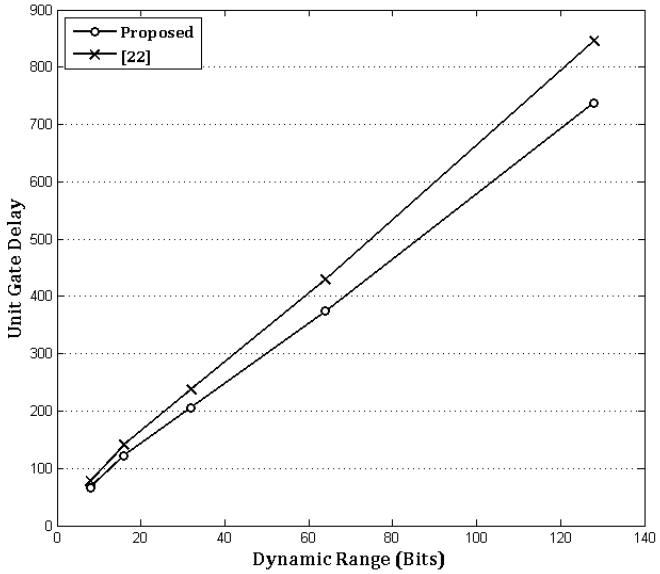
جدول (۵): جدول کلی مقایسه هزینه‌ی سخت افزاری و تاخیر گیت واحد شناساگرعلامت پیشنهادیبا روش پیشین

Sign Detector	Moduli set	Dynamic Range: 8 bit, n=2		16 bit, n=4		32 bit, n=7		64 bit, n=13		128 bit, n=26	
		Area	Delay	Area	Delay	Area	Delay	Area	Delay	Area	Delay
Proposed	$(2^n-1, 2^n, 2^n+1, 2^{2n}+1)$	120	66	236	122	410	206	758	374	1512	738
[22]	$(2^n-1, 2^n, 2^n+1, 2^{2n}+1)$	2*42	77*	4*42	14*1	7*42	23*7	13*42	42*9	26*42	84*5

* نشان دهنده هزینه سخت افزاری و تاخیر مقایسه کننده می‌باشد که در جدول بالا در نظر گرفته نشده است.



شکل (۵): مقایسه هزینه سخت افزاری مدار شناساگر علامت پیشنهادی در مجموعه پیمانه $\left(2^n-1, 2^n, 2^n+1, 2^{2n}+1\right)$ با روش پیشین



شکل (۶): مقایسه تاخیر مدار شناساگر علامت پیشنهادی در مجموعه پیمانه $\left(2^n-1,2^n,2^n+1,2^{2n}+1\right)$ با روش پیشین

۶- نتیجه گیری

بی وزن بودن سیستم اعداد مانده‌ای سبب سرعت بالا در انجام محاسبات ریاضی می‌گردد. از طرف دیگر بی وزن بودن اعداد در این سیستم تشخیص علامت را مشکل می‌سازد.

در این مقاله برای اولین بار به طراحی مدار شناساگر علامت جدید و کارآمد برای سیستم اعداد مانده‌ای چهار پیمانه‌ای $\left(2^n-1,\ 2^n,\ 2^n+1,\ 2^{2n}+1\right)$ با کمک تئوری باقیمانده چینی جدید II پرداختیم تا در پی آن استفاده از سیستم‌های اعداد مانده‌ای چهارپیمانه‌ای در کاربردهای امروزی که نیاز به محدوده دینامیکی بالا و نیز شناسایی علامت دارند، امکان پذیر شود. مدار پیشنهادی در مقایسه با روش پیشین از مدار مقایسه‌گر استفاده نمی‌کند و تنها به n بیت برای تشخیص علامت نیاز دارد و همین امر سبب می‌شود دارای سرعت بیشتر و هزینه سخت افزاری کمتری نسبت به روش پیشین گردد. از آن جایی که شناساگر علامت به عنوان یک عنصر اساسی در عملیات مشکل و پیچیده RNS نظیر تقسیم و مقایسه اعداد به شمار می‌رود، در نتیجه مدار پیشنهادی می‌تواند منجر به تسریع عملیات مذکور نیز گردد.

۷- مراجع

[1]Molahosseini Amir Sabbagh, Navi Keivan, Dadkhah Chitra, Kavehei Omid and Timarchi Somayeh, "Efficient reverse converterdesigns for the new 4- moduli sets (2^n-1,2^n,2^n+1,2^{2n+1}-1) and (2^n-1,2^n+1,2^{2n},2^{2n}+1) based on new CRTs," IEEE Transactions on Circuits and Systems I: Regular Papers , Apr. 2010, vol. 57, no. 4, pp. 823-835.

on Communications, Mar. 2008, vol. 56, no. 3, pp. 325-330.

[15]Tomczak Tadeusz, "Fast Sign Detection for RNS $(2^n-1,2^n,2^n+1)$,"IEEE Transactions on Circuits and Systems I:Regular Papers, Jul. 2008,vol. 55, no. 6, pp. 1502-1511.

[16] Akkal M. and Siy P., "A new Mixed Radix Conversion algorithm MRC-II, " Elsevier Journal of Systems Architecture, Sep. 2007, vol. 53, no. 9, pp. 577-586.

[17] Akkal M. and Siy P., "Optimum RNS sign detection algorithm using MRC-II with special moduli set, " Elsevier Journal of Systems Architecture, Oct. 2008, vol. 54, no. 10, pp. 911-918.

[18]Setiaarif Eep and Siy Pepe, "A new moduli set selection technique to improve sign detection and number comparison in Residue Number System (RNS)," IEEE Proceedings, Fuzzy Information Processing Society, 2005. NAFIPS 2005. Annual Meeting of the North American, Jun. 2005, pp. 766- 768.

[19] Radadi E.AL and Siy P., "RNS Sign Detector Based on Chinese Remainder Theorem II (CRT II)," Elsevier Journal Computers and Mathematics with Applications, Nov. 2003, vol. 46, no. 10-11, pp. 1559-1570.

[20]Hariri Arash, Navi Keivan and Rastegar Reza, "A new high dynamic range moduli set with efficient reverse converter," International Elsevier Journal of Computers and Mathematics with Applications, 2008, vol. 55, no. 4, pp. 660–668.

[21]Piestrak Stanislaw J., "A high speed realization of a residue to binary number system converter," IEEE Transaction on Circuits and Systems II, Analog and Digital Signal Processing, Oct. 1995, vol. 42, no. 10, pp. 661–663.

[22]Cao Bin, Chang Chip-Hong and Srikanthan Thambipillai, "An efficient reverse converter for the 4-moduli set $(2^n-1,2^n,2^n+1,2^{2n}+1)$ based on the new Chinese remainder theorem," IEEE Transaction on Circuits and Systems I: Regular Papers, Oct. 2003, vol. 50, no. 10, pp. 1296–1303.

[23]Molahosseini Amir Sabbagh, Navi Keivan, Hashemipour Omid and Jalali Ali, "An efficient architecture for designing reverse converters based on a general three-moduli set," Elsevier Journal of Systems Architecture, Oct. 2008, Vol. 54 , Issue 10, Pages 929-934.

[2] Wang Wei, Swamy M. N. S and Ahmad M. O, "RNS Application for Digital Image Processing," in Proceedings of the 4th IEEE International Workshop on System-on-Chip for Real-Time Applications, Jul. 2004, pp. 77-80.

[3] Taylor Fred J, "A Single Modulus Complex ALU for Signal Processing," IEEE Transactions on Acoustics, Speech, Signal Processing, Oct. 1985, vol. 33, no. 4, pp. 1302-1315.

[4] Hu Jingwei, Guo Wei, Wei Jizeng, Chang Yisong and Sun Dazhi, "A Novel Architecture for Fast RSA Key Generation Based on RNS," in Proceedings of the 4th International Symposium on Parallel Architectures, Algorithms and Programming(paap), 2011, pp. 345-349.

[5] Bajard Jean-Claude and Imbert Laurent, "Brief contributions: A Full RNS Implementation of RSA," IEEE Transactions on Computer, vol. 53, no. 6, pp. 769-774, Jun. 2004.

[6] Chen Jienan and Hu Jianhao, "Energy-Efficient Digital Signal Processing via Voltage-Overscaling-Based Residue Number System," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2012, Issue 99, pp. 1-11.

[7] Cardarilli Gian Carlo, Nannarelli Alberto and Re Marco, "Residue number system for low-power DSP applications," in Proceedings of the 41th Asilomar Conference on Signals, Systems, Computers, 2007, pp. 1412–1416.

[8] Stamenkovic Negovan and Stojanovic Vladica, "Constant-Coefficient FIR Filters Based on Residue Number System Arithmetic," Serbian Journal Of Electrical Engineering, Oct. 2012, vol. 9, no. 3, pp. 325-342.

[9] Pontarelli Salvatore, Cardarilli Gian Carlo, Re Marco and Salsano Adelio, "Optimized Implementation of RNS FIR Filters Based on FPGAs," Journal of Signal Processing Systems, Sep. 2010.

[10] Maji Pallab and Rath Girija Sankar, "A Novel Design Approach for Low Pass Finite Impulse Response Filter Based onResidue Number System," International Conference on Electronics Computer Technology (ICECT), 2011, vol. 3, pp. 74-78.

[11] Conway Richard and Nelson John, "Improved RNS FIR filter architectures," IEEE Transactions on Circuits and Systems II: Express Briefs, Jan. 2004, vol. 51, no. 1, pp. 26–28.

[12] Jassbi Somayyeh Jafarali, Hosseinzade Mehdi and Navi Keivan, "Redundant Multi-Level one-hot Residue Number System Based Error Correction Codes," East-West Design & Test Symposium(EWDTS), 2010, PP. 491-494.

[13] Timarchi Somayeh and Navi Keivan, "Efficient class of redundant residue number system," in Proceedings IEEE International Symposium on Intelligent Signal Processing, 2007.

[14] Goh Vik Tor and Siddiqi Mohammad Umar, "Multiple error detection and correction based on redundant residue number systems," IEEE Transactions