

ارائه یک الگوریتم جدید همگرایی برای محاسبه سخت افزاری جذر اعداد

شادرخ سماوی

استادیار دانشکده مهندسی برق و کامپیوتر - دانشگاه صنعتی اصفهان

علیرضا رضایی صدرآبادی

فارغ التحصیل کارشناسی ارشد - دانشکده مهندسی برق و کامپیوتر - دانشگاه صنعتی اصفهان

(تاریخ دریافت ۷۹/۷/۹، تاریخ تصویب ۸۱/۱/۲۴)

چکیده

بسیاری از الگوریتمهای موجود جذر نیاز به تعیین تخمین اولیه جذر برای شروع محاسبه جذر دارند. برای به دست آوردن تخمین اولیه معمولاً از ROM استفاده می‌شود، که پیاده سازی سخت افزاری آن فضای زیادی را اشغال می‌کند. در این مقاله ابتدا روشی بهبود یافته‌ای برای محاسبه تخمین اولیه جذر کلیه اعداد حقیقی و صحیح با خطای کمتر از ۶ درصد ارائه شده است. سپس یک روش پیاده سازی سخت افزاری جذر با دقت مناسب برای اعداد ممیز شناور پیشنهاد گردیده است.

واژه های کلیدی: جذر، ممیز شناور، الگوریتم همگرایی، خط لوله، FPGA

مقدمه

ثابتی رقم جذر را محاسبه می‌کنند به همین دلیل دارای همگرایی خطی است. دقت جذر محاسبه شده، بستگی مستقیم به تعداد تکرارها دارد. یکی از این روشها SRT است که در ریزپردازنده‌های زیادی استفاده شده است. دسته دوم، روشهای تقریبی مبتنی بر تکرار در حل معادلات غیر خطی نظیر روش نیوتن رافسون است. در این روشها عمل جذر با یک تخمین اولیه از جذر یا معکوس آن شروع می‌شود و با کمک رابطه‌ای بازگشتی این تخمین بهبود می‌یابد تا در نهایت جذر عدد مورد نظر حاصل شود. همگرایی این روشها می‌تواند غیرخطی باشد. در این روشها با استفاده از عمل ضرب به عنوان عمل اصلی در هر تکرار تعداد بیت‌های صحیح حاصل را دو برابر می‌کنند. به این ترتیب همگرایی "توان دو" حاصل می‌گردد [۹-۲].

الف) روش نیوتن رافسون: این روش از جمله روشهای تکرار ضرب است و قابلیت تصحیح خود را دارد [۵، ۲]. یعنی خطای تخمین اولیه در تکرارهای بعدی تصحیح می‌شود. در این روش رابطه بازگشتی به صورت $X_{m+1} = X_m - f(X_m)/f'$

عمل جذر پس از چهار عمل اصلی جمع، تفریق، ضرب و تقسیم، مهمترین و پرکاربردترین عمل در محاسبات علمی است. محاسبات عددهای مختلط، محاسبات آماری، گرافیک و پردازش تصویر برخی از زمینه های کاربرد عمل جذر است. اهمیت این عمل باعث شد که در سال 1985 استاندارد IEEE754 برای محاسبات باینری با ممیز شناور عمل جذر را در مجموعه دستورالعملهایی قرار دهد که باید به صورت مجزا به وسیله کلیه نرم افزارها و سخت افزارهای سازگار با این استاندارد پیاده‌سازی شود [۱].

در گذشته پیاده سازی جذر به صورت نرم افزاری انجام می‌شد و با سرعت پایین اجرا می‌شد. با پیشرفت تکنولوژی و امکان قرار دادن مدارهای بسیار بزرگ در داخل یک IC، به منظور افزایش سرعت، عمل جذر می‌تواند به صورت سخت افزاری پیاده سازی شود.

روشهای متنوعی برای عمل جذر موجود است. این روشها را می‌توان به دو دسته تقسیم کرد: دسته اول روشهای محاسبه رقم به رقم جذر که این روشها در هر مرحله تعداد

در این رابطه R_i باقی مانده جزئی مرحله i ام، Y_i جذر جزئی و جذر جزئی و y_i بیت i ام جذر تحت شرایطی است که $Y_1=0.1$ و $y_1=1$. اگر $R_{i+1} \geq 0$ آنگاه $Y_{i+1}=1$ در غیر این صورت $Y_{i+1}=-1$ می شود. همچنین R_{i+1} را می توان به صورت زیر تبدیل کرد:

$$X_{i+1} = 2X_i - 2Y_i y_i + y_i^2 2^i$$

که به این ترتیب از عمل مجذورگیری خودداری می شود. در روشهای بازگشتی نیاز به مداری برای تولید تخمین اولیه است. در اکثر پیاده سازی های سخت افزاری از یک ROM به این منظور استفاده می شود که حجم زیادی از مدار را اشغال می کند [۵]. در مرجع [۲] روش ساده ای برای تخمین اولیه جذر اعداد صحیح با تعداد ارقام زوج و اعداد حقیقی ممیز شناور^۴ با نمای زوج پیشنهاد شده است. همچنین رابطه ای بازگشتی برای محاسبه دقیق جذر ارائه گردیده است. در این مقاله پس از بررسی این روش، روش جدیدی برای محاسبه تخمین اولیه جذر کلیه اعداد با تقریب حداکثر 6% ارائه می شود. سپس پیاده سازی سخت افزاری این روش جدید جذر برای اعداد ممیز شناور به صورت خط لوله معرفی می گردد.

تقریب اولیه جذر

تقریب اولیه برای اعداد صحیح و حقیقی متفاوت است. در این بخش به توضیح هر دو مورد و ارائه روش بهبود یافته برای هر مورد می پردازیم.

اعداد حقیقی (ممیز شناور)

برای محاسبه جذر عدد ممیز شناور نرمالیزه $X * 2^e$ که در آن مانتیس در محدوده $1/2 \leq X < 1$ است، لازم است جذر هر دو قسمت مانتیس و نما محاسبه شود. اگر نما زوج باشد محاسبه نمای جذر با تقسیم e بر دو انجام می شود. اما اگر فرد باشد ابتدا یک واحد به نما افزوده می شود تا نما زوج شده و محاسبه جذر آن ممکن گردد. برای جبران این نمای اضافی، مانتیس بر دو تقسیم می شود. در این صورت مانتیس در محدوده $1/4 \leq X < 1/2$

است. جذر یا هر فرآیند محاسباتی دیگر که محاسبه آن مد نظر است ریشه معادله $f(x)=0$ می باشد. اگر X_0 تقریب اولیه باشد با n بار به کار بردن رابطه بازگشتی، جواب مورد نظر خواهد بود. برای عمل جذر دو راه وجود دارد. اگر $f(x)=x^2-A$ باشد، رابطه بازگشتی به صورت $X_{m+1}=(X_m+A/X_m)/2$ است. اجرای این روش نیاز به عمل تقسیم دارد که پیاده سازی سخت افزاری آن مشکل است. انتخاب دیگری به صورت $f(x)=1/x^2-A$ است که رابطه بازگشتی برای این حالت $X_{m+1} = X_m * (3 - A * X_m^2) / 2$ است. در این روش برای هر تکرار نیاز به سه عمل ضرب و یک عمل جمع یا تفریق است. X_0 در این حالت تقریب اولیه معکوس جذر می باشد. در نهایت پس از n بار تکرار، جذر از رابطه $Y=X_n * A$ حاصل می شود. این روش برای پیاده سازی سخت افزاری مناسب تر است. ولی به دلیل اینکه تعداد گیت های مورد نیاز برای ضرب کننده زیاد است، پیاده سازی سخت افزاری روش نیوتن رافسون به صورت خط لوله^۲ اگر غیر ممکن نباشد نیاز به سخت افزار زیادی دارد. روش گلداشمیت [۱] با تغییر ترتیب عملیات نیوتن رافسون توازی عملیات و کاهش تاخیر را ایجاد می کند ولی نتیجه از حاصل ضرب تعدادی عبارت مستقل از هم تشکیل می شود و خطای یک عبارت توسط عبارات دیگر تصحیح نمی گردد. (ب) روش **SRT**: این روش بر اساس عبارت تکرار زیر عمل می کند:

$$X_{i+1} = 2X_i - 2Y_i y_{i+1} - y_{i+1}^2 2^{-(i+1)}$$

$$Y_{i+1} = Y_i + y_{i+1} 2^{-(i+1)}$$

که X_i باقی مانده جزئی و X_0 عدد اولیه است. Y_i جذر حاصل شده تا مرحله i ام و y_i رقم جذر در i امین مرحله است که $y_i \in \{-1, 0, 1\}$. برای به دست آوردن y_{i+1} باید "تابع انتخاب رقم" به کار گرفته شود.

(ج) روشهای غیر مجدد^۳: این روشها شبیه SRT هستند ولی از مکمل ۲ برای نمایش جذر استفاده می کنند. در یک الگوریتم کلاسیک رابطه تکرار به صورت زیر خواهد بود:

$$R_{i+1} = X - Y_i^2$$

$$Y_{i+1} = Y_i + y_{i+1} 2^{-(i+1)}$$

(۵) در این حالت بیشترین خطا برای $X=1/2$ و برابر با 6.07% می‌باشد. در شکل (۱) نمودار تابع \sqrt{X} به همراه تقریب حاصل از روابط فوق رسم شده است.

جذر اعداد صحیح

حال به بررسی جذر اعداد صحیح می‌پردازیم. برای تقریب اولیه اعداد صحیح $2m$ بی‌تی، مرجع [۲] الگوریتم زیر را پیشنهاد کرده است.

الگوریتم ۱:

الف- از $m+1$ بیت سمت راست (ارقام کم ارزش) صرف نظر کنید. در واقع عمل تقسیم بر 2^{m+1} انجام شود.
ب- یک بیت "1" به سمت چپ (رقم پر ارزش) اضافه کنید. در واقع عمل جمع عدد حاصل از الف با 2^{m-1} انجام گیرد.
عدد m بی‌تی حاصل از دو قدم فوق با خطای حداکثر 6.07% ، جذر عدد $2m$ بی‌تی اولیه است. در اینجا پیشنهاد می‌کنیم که برای اعداد صحیح $2m-1$ بی‌تی الگوریتم زیر را می‌توان انجام داد.

الگوریتم ۲:

الف- از m بیت سمت راست (ارقام کم ارزش) صرف نظر کنید. در واقع عمل تقسیم بر 2^m انجام شود.
ب- بیت سمت چپ (رقم پر ارزش) را حذف کرده، دو بیت "10" را به سمت چپ عدد حاصل اضافه کنید. در واقع عمل جمع عدد حاصل از الف با 2^{m-2} انجام گیرد.
عدد m بی‌تی حاصل از دو قدم فوق با خطای حد اکثر 6.07% جذر عدد $2m-1$ بی‌تی اولیه است. برای به دست آوردن الگوریتم‌های ۱ و ۲ فوق از رابطه (۱) به ترتیب برای $a=2^{2m}$ و $a=2^{2m-2}$ استفاده شده است.

قرار می‌گیرد. تخمین اولیه جذر اعداد در فاصله $1/2 \leq X < 1$ را می‌توان از رابطه $T=(1+X)/2$ به دست آورد [۲]. در رابطه مذکور حداکثر خطا حدود 6% است. دلیل استفاده از رابطه فوق که در منابع و منتشرات ذکر نشده است را در ادامه بحث می‌آوریم.

دو جمله اول سری تیلور برای تابع $F(x)$ به صورت زیر است:

$$F(x) \approx F(a) + F'(a) * (x - a) \quad (۱)$$

برای تابع جذر و $a=1$ رابطه فوق به صورت زیر حاصل می‌شود:

$$\sqrt{x} \approx 1 + \frac{1}{2\sqrt{1}} * (x - 1) = (1 + x) / 2 \quad (۲)$$

و خطای نسبی حاصل از این تقریب عبارت است از:

$$\text{err}(x) = \left| \sqrt{x} - (1 + x) / 2 \right| / \sqrt{x} \quad (۳)$$

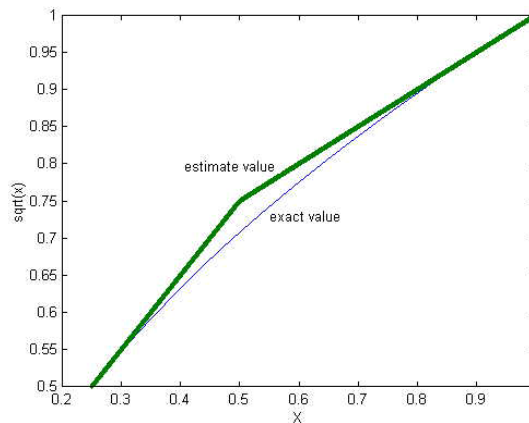
که بیشترین خطا برای $x=1/2$ و برابر با 6.07% است. روش فوق زمانی کارایی دارد که نمای عدد مقداری زوج باشد. حال به ارائه روش جدیدی می‌پردازیم که تخمین اولیه جذر اعداد با نمای فرد را نیز امکان پذیر می‌کند.

برای محاسبه تخمین اولیه جذر برای $1/4 \leq X < 1/2$ رابطه (۱) را برای $a=1/4$ و $F(x) = \sqrt{X}$ به کار می‌بریم که با استفاده از بسط تیلور عبارت زیر نتیجه می‌شود:

$$\sqrt{x} \approx 1/2 + \frac{1}{2\sqrt{1/4}} * (x - 1/4) = x + 1/4 \quad (۴)$$

و خطای نسبی حاصل از این تقریب از رابطه زیر به دست می‌آید:

$$\text{err}(x) = \left| \sqrt{x} - (1/4 + x) \right| / \sqrt{x}$$



شکل ۱: نمودار تابع جذر و تقریب آن با استفاده از روابط (۲) و (۴)

الگوریتم تکرار برای جذر اعداد ممیز شناور

الگوریتم مطرح شده در مرجع [۲] به صورت زیر است:

```

square root(input: X*2n,output:Y*2m)
1: if (n is even) then (m:=n/2)
   else m:=(n+1)/2           (*evaluation of exponent*)
2: Z:=(1-X)/2                (* initial estimate of square root *)
3: y0:=Z, i:=1              (* initialize *)
4: yi:=Z+(yi-1)2/2
5: d:=yi-yi-1, if d>ε then   (* ε is a specified precision *)
   i:=i+1 ,goto step 4
   else continue
6: Y=1-yi
7: if (n is odd) then ( Y=Y*0.7071067)
    
```

نیازی به اصلاح نتیجه در پایان نخواهد بود و آخرین عمل ضرب حذف می‌شود. الگوریتم فوق برای ورودیهای نرمالیزه خروجی نرمالیزه تولید می‌کند. پیشنهاد می‌شود که برای محاسبه نما چه فرد باشد و چه زوج از رابطه $m=(n+1)/2$ استفاده شود. عمل تقسیم بر دو با شیفت به راست اجرا می‌شود و در صورت زوج بودن n یک واحد اضافه شده به نما تأثیری در نتیجه نخواهد داشت. الگوریتم اصلاح شده به صورت زیر است:

الگوریتم فوق جذر اعداد با نمای فرد را مستقیماً محاسبه نمی‌کند و برای محاسبه جذر این اعداد در پایان حاصل را در $1/\sqrt{2}$ ضرب کرده و نتیجه را اصلاح می‌کند. در ادامه به اصلاح این الگوریتم پرداخته می‌شود. اگر نمای عدد ممیز شناور فرد باشد، چنانچه در بخش قبل ذکر شد با افزایش یک واحد زوج می‌شود و مانیتیس بر دو تقسیم می‌شود (شیفت به راست). در این صورت مانیتیس مقداری بین $1/4$ و $1/2$ خواهد بود (بافرض این که عدد در ابتدا نرمالیزه بوده است) و برای تقریب اولیه می‌توان از رابطه $T=1/4+X$ استفاده کرد. در این صورت

که در آن X_i و Y_i به ترتیب بیت‌های مانتیس عملوند جذر (N) و حاصل جذر (H) هستند و e_X و e_Y نمای عملوند جذر و نمای حاصل جذر می‌باشند.

پیاده سازی بخش نما

برای محاسبه نمای حاصل جذر داریم:

$$n=e_X-b, m=e_Y-b, m=(n+1)/2$$

(۷)

با توجه به روابط فوق رابطه زیر حاصل می‌شود.

$$e_Y=e_X/2+b/2+e_X \text{ MOD}(2)$$

(۸)

پیاده سازی جذر برای اعداد با ممیز شناور

در این بخش روشی برای پیاده‌سازی الگوریتم پیشنهادی ارائه می‌کنیم. فرض می‌شود عدد با روشی مشابه با استاندارد IEEE754 نمایش داده می‌شود. یعنی مانتیس عدد به صورت اندازه و علامت^۵ بیان شده و نما به صورت مکمل دوی بایاس شده با ثابت بایاس b است. در این صورت اگر نمایش عملوند و نتیجه جذر به صورت نشان داده شده در شکل (۲) باشد مقدار آنها از روابط زیر به دست می‌آید:

$$N = (-1)^s * (1.X_2X_3K X_n) * 2^{e_X-b}$$

$$H = (-1)^s * (1.Y_2Y_3K Y_n) * 2^{e_Y-b}$$

(۹)

square root (input: $X*2^n$, output: $Y*2^m$)

1: $m := (n+1)/2$ (*evaluation of exponent *)

2: if n is odd $X := X/2$ (* shift left one bit if exponent is odd*)

else continue

3: if $1/4 \leq X < 1/2$ then $T := 1/4 + X$

if $1/2 \leq X < 1$ then $T := (1+X)/2$ (*initial estimate *)

4: $z_0 = (1+X)/2, Z = 1 - z_0, y_0 = 1 - T, i = 1$ (* initialization*)

5: $y_i = Z + (y_{i-1})^2 / 2$

6: $i := i + 1$

7: if $i \leq k$ go to step 5 else continue

8: $Y := 1 - y_k$

ورودی الگوریتم فوق X و n است که به ترتیب مانتیس و نمای عددی است که می‌خواهیم جذر آن را بگیریم. در پایان اجرای الگوریتم Y و m به ترتیب مانتیس و نمای حاصل جذر خواهند بود. چنانچه مشاهده می‌شود نیاز به اصلاح نهایی وجود ندارد. یعنی عمل ضرب در مقدار ثابت 0.7071067 حذف شده است.

مقدار اولیه جذر به دست آمده را با انجام چهار تکرار بهبود می‌دهد که این چهار مرحله برای جذر اعداد هشت بیتی کافی می‌باشد. با افزایش تعداد بیت‌های مورد محاسبه جذر، تعداد تکرارها باید افزایش یابد. این مدار به صورت خط لوله عمل می‌کند و قادر است در هر سیکل ساعت یک ورودی جدید دریافت کند.

در شکل (۴) چگونگی به دست آوردن مقادیر اولیه T و Z_0 نشان داده شده است. با توجه به شکل می‌توان دید که پیاده سازی T (مقدار اولیه جذر) در دو حالت $e_0=0$ و

در رابطه فوق $\text{MOD}(2)$ به معنی باقیمانده تقسیم بر دو است که همان بیت کم ارزش e_X یعنی e_0 است. رابطه فوق با یک جمع کننده پیاده سازی می‌شود.

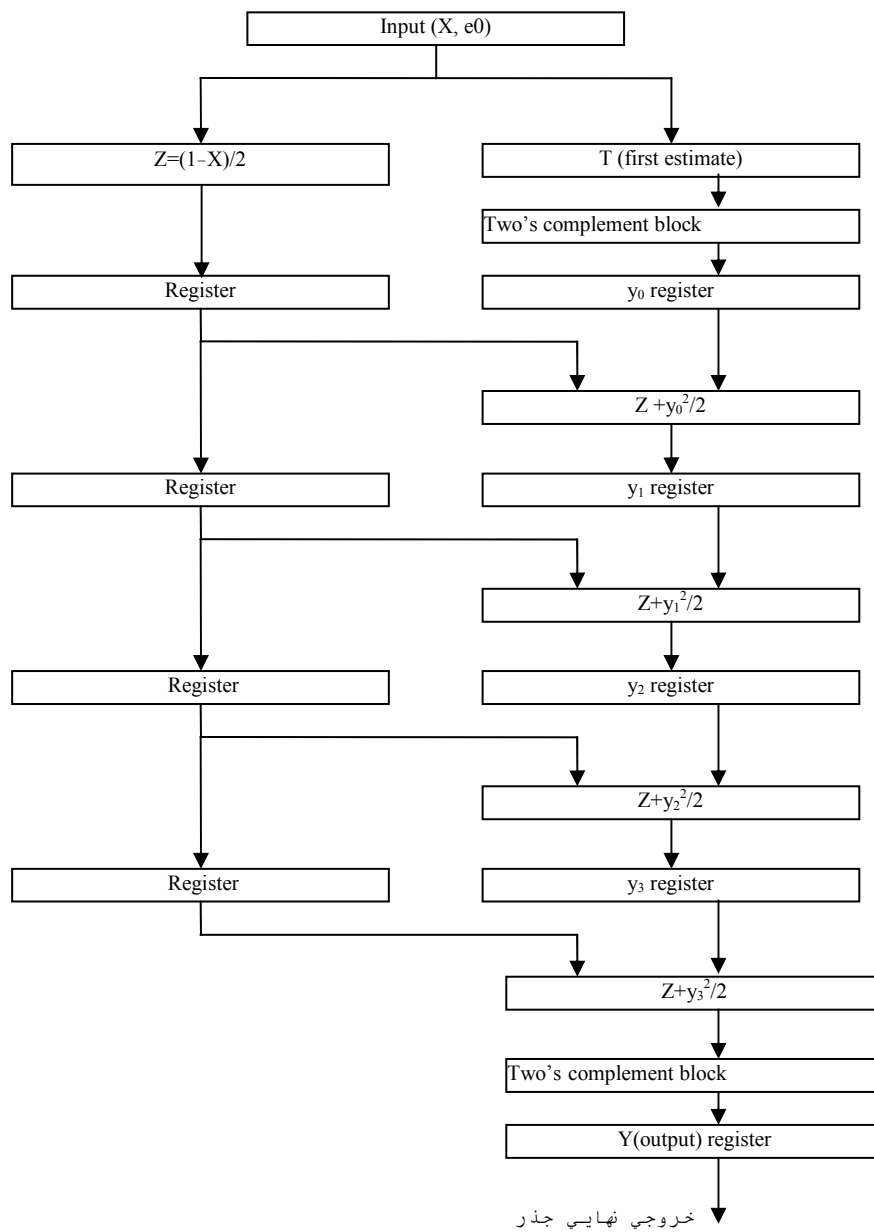
محاسبه مانتیس

نمودار بلوکی محاسبه مانتیس در شکل (۳) رسم شده است. مانتیس عدد ممیز شناور به همراه بیت کم ارزش نما را مدار، به عنوان ورودی دریافت کرده و مانتیس جذر را به عنوان خروجی تولید می‌کند. این مدار دقت

برای محاسبه ضرب کاهش می‌یابد. برای مثال استفاده از ضرب کننده آرایه‌ای براون^۱ برای محاسبه ضرب دو عدد هشت بیتی نیاز به 56 عدد FA و 64 عدد گیت AND دارد [۷] ولی در این تحقیق این کار با استفاده از 28 عدد FA و 28 گیت AND انجام شده است. به منظور افزایش سرعت، نتیجه نهایی مجذور کننده به صورت CSA باقی مانده و پس از شیفتم (تقسیم بر دو) و جمع با Z از CPA برای محاسبه نهایی استفاده می‌شود. در شکل (۶) مدار این بلوک برای محاسبات هشت بیتی نشان داده شده است. به دلیل اینکه در محاسبات هشت بیتی، تنها به خروجیهای هشت بیت پردازش مجذور نیاز است مدار قابل ساده سازی می‌باشد. علاوه بر این به جای برخی از FA ها که یکی از

$e_0=1$ با استفاده از یک گیت not قابل انجام است (شکل ۵-الف). Z_0 مقدار تعیین کننده Z است که ثابت رابطه تکرار می‌باشد. برای انتخاب مقادیر Z_0 در دو حالت $e_0=1$ و $e_0=0$ از یک تسهیم کننده^۲ n بیتی استفاده شده است (شکل ۵-ب). ورودی‌های این تسهیم کننده عبارتند از $[2:n-2]X_{101}$ که همان بردار X می‌باشد با سه بیت پردازش 101 و دیگر ورودی تسهیم کننده $[2:n-1]X_{11}$ است.

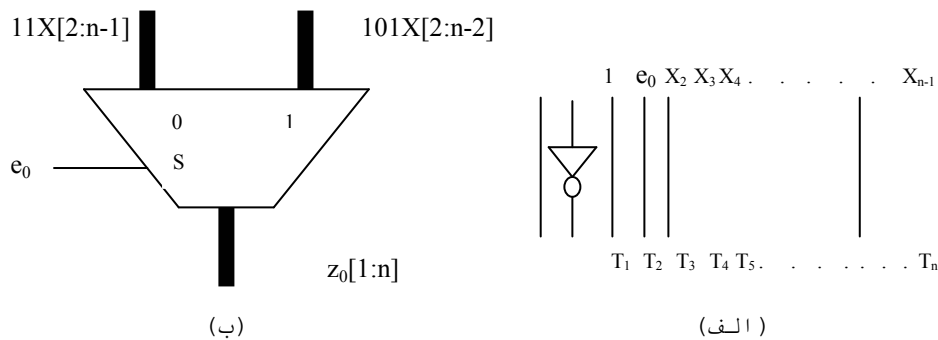
برای محاسبه عبارت تکرار $y_i = Z + y_{i-1}^2 / 2$ نیاز به یک ضرب کننده و یک جمع کننده است. با توجه به اینکه دو ورودی ضرب کننده یکسان است مدار ضرب کننده قابل ساده سازی است و تعداد گیت‌های And و FA های لازم



شکل ۳: نمودار بلوکی مدار پیشنهادی جذر (بخش محاسبه مانتیس).

حالت $e_0=0$							حالت $e_0=1$						
$X=$	0.	1	X_2	X_3	.	X_n	$X=$	0.	1	X_2	X_3	.	X_n
$X+1=$	1.	1	X_2	X_3	.	X_n	Shifted $X=$	0.	0	1	X_2	.	X_{n-1}
$z_0=(X+1)/2=$	0.	1	1	X_2	.	X_{n-1}	$X+1=$	1.	0	1	X_2	.	X_{n-1}
$T=(X+1)/2=$	0.	1	1	X_2	.	X_{n-1}	$z_0=(X+1)/2=$	0.	1	0	1	X_2	X_{n-2}
							$T=1/4+X=$	0.	1	0	X_2	X_3	X_{n-1}

شکل ۴: چگونگی محاسبه T و Z از روی مقادیر X و e_0 .



شکل ۵: روش پیشنهادی برای پیاده سازی سخت افزاری محاسبه مقادیر T و z_0 ، الف- اجرای T (تخمین اولیه جذر) ب- اجرای $z_0=(1+X)/2$.

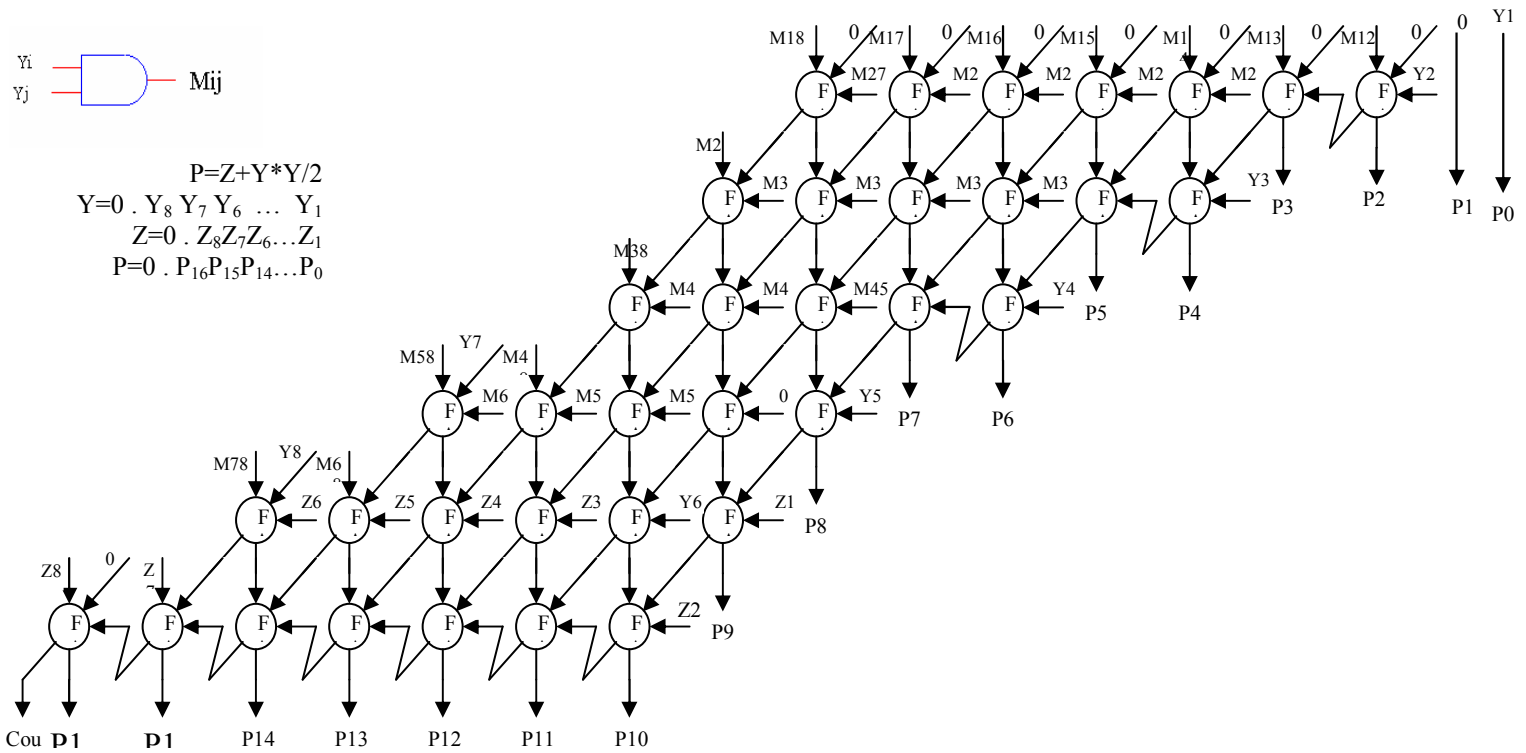
مجذور کننده هشت بیتی ارائه شده در این مقاله و یک ضرب کننده هشت بیتی پیاده سازی کرده و نتایج را با یکدیگر جمع کرد. در شکل (۳) برای محاسبه $(1-T), (1-Z_0)$ و $(1-y_i)$ از مدار مکمل دو استفاده شده است. برای سادگی پیاده سازی سخت افزاری می توان به جای مکمل دو از مدار مکمل یک (گیت های not) استفاده کرد. در این صورت خطای حاصل از این تقریب در مقدار نهایی جذر از رابطه زیر به دست می آید:

$$\text{error}(x) = (\sqrt{x + 2^{-n+1}} - 2^{-n}) - \sqrt{x} < 2^{-n} \quad (10)$$

ورودیهای آن صفر است از مدار 9HA استفاده می شود. در شکل (۷) مدار ساده شده نشان داده شده است. در این طراحی از مدار جدیدی به نام ^{10}CG استفاده شده است که پیچیدگی و سطح مدار را کاهش می دهد. در واقع این مدار وضعیت انتشار نقلی را تشخیص می دهد. تحت شرایطی که تولید حاصل جمع در یک FA مورد نظر نباشد و فقط تولید بیت نقلی اهمیت داشته باشد از این مدار استفاده شده است. برای گسترش مجذور کننده به ۱۶ بیتی با فرض اینکه A یک عدد ۱۶ بیتی باشد داریم:

$$\begin{aligned} (A_{15-0})^2 &= (A_{15-8}) * 2^8 + (A_{7-0})^2 \\ &= (A_{15-8})^2 * 2^{16} + (A_{15-8}) * (A_{7-0}) * 2^9 + (A_{7-0})^2 \end{aligned} \quad (9)$$

بنابراین می توان مجذور کننده ۱۶ بیتی را با استفاده از دو



شکل ۶ : مدار محاسبه مجذور - شیف - جمع با استفاده از بلوکهای تمام جمع کننده .

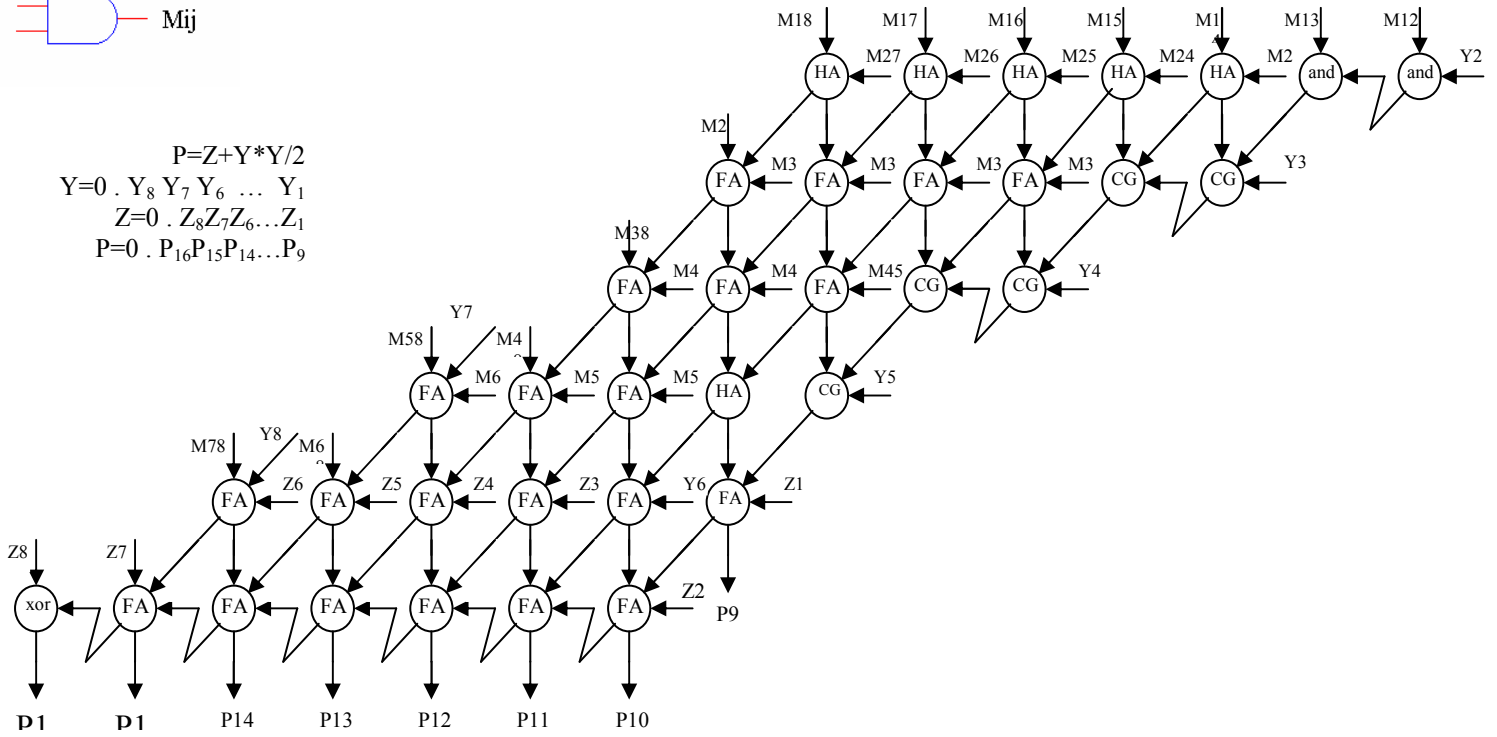


$$P = Z + Y * Y / 2$$

$$Y = 0 . Y_8 Y_7 Y_6 \dots Y_1$$

$$Z = 0 . Z_8 Z_7 Z_6 \dots Z_1$$

$$P = 0 . P_{16} P_{15} P_{14} \dots P_9$$



شکل ۷: مدار مجذور- شیفتم - جمع بهبود داده شده با کمک بلوکهای تمام جمع کننده (FA) و نیم جمع کننده (HA) و تولید کننده نقلی (CG)

جدول ۱: نتایج شبیه سازی جذر برای چند ورودی (داده‌ها به هگزادسیمال هستند).

clock	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	t11	t12	t13	t14
e0(input)	0	0	0	1	1	1	0	1						
x(input)	8F	F1	D8	D8	85	98	AA	AA						
T		C7	F8	EC	AC	82	8C	D5	95					
y0			39	08	14	54	7E	74	2B	6B				
y1				3F	08	14	57	7E	74	2E	6C			
y2					40	08	14	58	7E	74	2F	6C		
y3						41	08	14	59	7E	74	2F	6C	
Y							BF	F8	EC	A7	82	8C	D1	94

جدول ۲: مقایسه طرح پیشنهادی و دو پردازنده تجاری.

تعداد سلول محاسباتی	تاخیر (t_{FA})	پردازنده	دقت
۲۸۰۹ عدد CSA ^{۱۱}	۲۰۴	IBM-RS6000	مضاعف
۴۰۹۶ عدد CSA	۱۳۶	WEITEK-W4146/4363	مضاعف
۳۵۱۹ عدد FA	۹۵	طرح پیشنهادی	مضاعف

نشان داده شده است. برای برخی از ورودی‌ها بیت آخر ممکن است دقیق نباشد. برای رسیدن به دقت تا آخرین رقم مورد نظر برای داده‌های n بیتی، باید محاسبات برای تعداد اندکی بیش از n بیت انجام شود.

برای مقایسه بین طرح ارائه شده و دیگر روشهای موجود، یک مدار کلاسیک غیربازیافتی نیر شبیه‌سازی شد [۷]. این مدار اعداد ممیز ثابت 16 بیتی را به عنوان ورودی دریافت و جذر 8 بیتی آنها را تولید می‌کند. شبیه‌سازی مدار برای پیاده‌سازی آن روی تراشه XC4002XLPQ100 انجام گرفت و نشان داد که 62 عدد از 64 عدد CLB تراشه را اشغال می‌کند. حداکثر تأخیر این مدار 176.36ns ثبت گردید.

به طور کلی اگر از مدار مجذور-شیفت-جمع شکل (۶) برای اعداد استاندارد IEEE-754 با دقت مضاعف^{۱۲} استفاده شود مانتیس‌های ۵۳ بیتی وجود خواهد داشت. با شیفت "۱" نهم این استاندارد به داخل مانتیس باید عملیات ۵۴ بیتی صورت بگیرد. حجم سخت افزار برای این کار برای انجام ۳ مرحله تکرار برابر با ۳۵۱۹ عدد FA

چون خطای فوق کمتر از کم ارزش‌ترین بیت محاسبه شده جذر است، قابل صرف نظر است.

نتیجه‌گیری

مدار جذر گیرنده برای مانتیس‌های هشت بیتی معرفی شده در این مقاله با کمک نرم‌افزار Foundation شرکت XILINX شبیه‌سازی شد و آماده پیاده‌سازی روی تراشه FPGA مدل 4005XLPC84 گردید. از مجموعه 196 عدد CLB این تراشه 160 عدد یعنی 81% مورد استفاده قرار گرفت. حداکثر فرکانس کار این مدار 24.462MHz ثبت شد. یعنی مدار قادر است در هر 40.88ns یک ورودی جدید دریافت کند.

روش ارائه شده در این مقاله در ازاء افزایش سخت افزار، بهبود چشم‌گیری در زمان اجراء خطلوله‌ای جذر اعداد ایجاد نمود. در این طراحی از چهار تکرار برای بهبود تقریب اولیه استفاده شده است. ولی نتایج شبیه‌سازی نشان می‌دهد که برای داده‌های هشت بیتی در بدترین حالت با سه تکرار، جذر با دقت تا آخرین بیت محاسبه می‌شود. نتایج شبیه‌سازی برای چند ورودی در جدول (۱)

۵۳ بیتی این تاخیر برابر با ۹۵ تاخیر تمام جمع‌کننده می‌شود.

پردازنده‌های IBM-RS6000 و WEITEK-4146/4363 [۹] در جدول (۲) با طرح پیشنهادی مقایسه شده‌اند.

در محاسبه حجم سخت‌افزار برای طرح پیشنهادی سه ردیف رجیستر ۱۱۸ بیتی برای انتقال نمای عملوند، Z و Y_i نیز باید در نظر گرفته شود. همچنانکه پردازنده‌های دیگر نیاز به ROM برای یافتن تخمین اولیه دارند. چنانچه دقت نتایج حاصله کافی نباشد با افزایش تعداد مراحل تکرار می‌توان دقت را افزایش داد. به این ترتیب حجم سخت‌افزار اضافه خواهد شد ولی تاخیر در تولید حاصل تغییری نخواهد کرد و این از مزایای طرح ارائه شده است.

خواهد بود که از رابطه پیچیدگی زیر می‌توان آنرا استخراج کرد:

$$\text{تعداد FA های یک مرحله تکرار} = \left\{ \sum_{i=1}^{n-1} (n-i) \right\} + \frac{n^2}{8} + n + 2 \quad (11)$$

در رابطه فوق n تعداد بیت‌های مانتیس است. همچنین تاخیر مدار فوق در بدترین شرایط از رابطه زیر قابل محاسبه است:

$$t_{FA} = \left(\frac{7}{4}n + 1 \right) t_{FA} \quad (12)$$

مقدار تاخیر برحسب تاخیر یک تمام جمع‌کننده یعنی t_{FA} بیان شده است. برای مدار دقت مضاعف و مانتیس

مراجع

- 1 - IEEE Standard for Binary Floating Point Arithmetic .ANSI/IEEE Std 754-1985.
- 2 - R. Hashemian, R. (1990). "Square rooting algorithms for integer and floating point numbers." *IEEE Transactions on Computers*, Vol. 39, No. 8, PP. 1025-1029.
- 3 - Li, Y. and Chu, W. (1996). "A new nonrestoring square root algorithm and its VLSI implementations." *Proceedings of the IEEE International Conference on Computer Design: VLSI in Computers and Processors*, PP. 538-544.
- 4 - Majerski, S. (1985). "Square rooting algorithms for high speed digital circuits." *IEEE Transactions on Computers*, Vol. C-34, No. 8, PP. 724-733.
- 5 - Ito, M., Takagi, N. and Yajima, S. (1995). "Efficient initial approximation and fast converging methods for division and square root." *Proceedings of the 12th Symposium on Computer Arithmetic*, PP. 2 -9 .
- 6 - Ciminiera, L. and Montuschi, P. (1990). "Higher radix square rooting." *IEEE Transactions on Computers*, Vol. 39, No. 10, PP. 1220-1231.
- 7 - Parhami, B. (2000). *Computer arithmetic: algorithms and hardware designs*. Oxford University Press.
- 8 - Tchoumatchenco, V., Vassileva, T. and Gurov, P. (1996). "A FPGA based square-root coprocessor." *Proceedings of the 22nd EUROMICRO Conference* , PP. 520 – 525.
- 9 - Chu, W. and Li, Y. (1999). "Cost/performance tradeoff of n-select square root implementations." *5th Australasian Computer Architecture Conference*, PP. 9-16.

واژه های انگلیسی به ترتیب استفاده در متن

- | | |
|--------------------|-----------------------|
| 1 – Quadratic | 10 – Carry Generate |
| 2 – Pipeline | 11 – Carry Save Adder |
| 3 – Non Redundant | 12 – Double Precision |
| 4 – Floating Point | |
| 5 – Sign Magnitude | |

-
- 6 – Multiplexer
 - 7 – Full Adder
 - 8 – Broun
 - 9 – Half Add