در مورد کاربرد جمع کننده های تقریبی با قابلیت ذخیره رقم نقلی در انباره های ضرب کننده

چکیده:

رایانش تقریبی با تخفیف الزام انجام محاسبات دقیق، عملکرد مدار دیجیتالی را بهبود می بخشد. ما در این مقاله به بررسی کاربرد جمع کننده های تقریبی در مرحلهٔ نهایی انباشتگر مولتی پلایر(MAC) با قابلیت ذخیرهٔ رقم نقلی که برای اپلیکیشن فیلترینگ عکس طراحی شده می پردازیم. ما یک جریان طراحی مبتنی بر ابزارهای سنتزی پیشنهاد می دهیم که از توصیف HDL آغاز می گردد.پس از اولین مرحله ای که در آن از جمع کنندهٔ انتشار رقم نقلی استفاده می شود، نت لیست سنتزی جهت استخراج آمار و ارقام موارد جمع شده در جمع کنندهٔ انتشار رقم نقلی شبیه سازی می گردد. سپس به طراحی جمع کنندهٔ تقریبی جهت مواجه با مشخصه های خطایی مورد نیاز که در آمار و ارقام ورودی داده شده می پردازیم. این نت لیست در نهایت با جایگزینی جمع کنندهٔ دقیق با جمع کنندهٔ تقریبی اصلاح می گردد و سنتر و بهینه سازی نهایی انجام می گیرد. نمونهٔ طرح ارائه شده در 28nm CMOS نشان می دهد که به بهرهٔ توان 14درصدی با کاهش محدود کیفیت عکس می توان دست یافت.

کلیدواژه ها: رایانش ابری، جمع کننده تقریبی، انباشتگر مولتی پلایر؛ مدارهای محاسباتی، سخت افزار غیردقیق

1. مقدمه

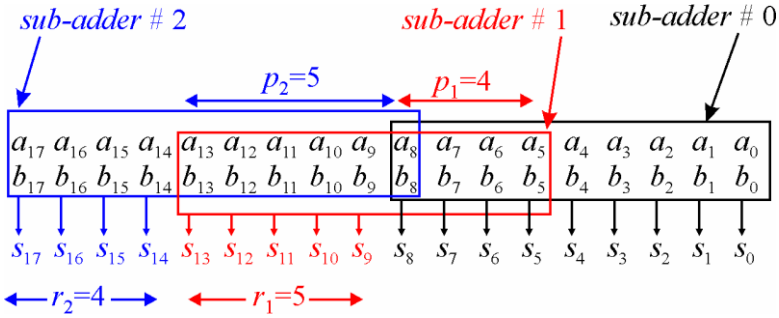
محاسبهٔ تقریبی یکی از روش های نویدبخش در بهبود عملکرد مدار دیجیتال از طریق تخفیف لزوم اجرای محاسبات دقیق می باشد. این تکنیک برای اپلیکیشن هایی که می توانند برخی خطاهای موجود در نتایج محاسبه شده مثل پردازش چندرسانه ای، داده کاوی و تشخیص داده ها را تحمل کنند بسیار مثمر ثمر است.

در بین اپراتورهای دیتاپد تقریبی، جمع کننده ها به عنوان یک زیرسیستمی که کاربرد بیشتری در اپلیکیشن های تحمل پذیر خطا دارند بیشتری توجهات را به خود جلب کرده اند. در یک جمع کنندۀ n بیتی، بدترین وضعیتی که در آن رقم نقلی در سرتا سر (یا اکثر) n بیتها منتشر می شود به ندرت زمانی رخ می دهد که ورودی ها به صورت یکنواخت توزیع شده باشند. جمع کننده های تقریبی از این مشاهده استفاده می کنند و برونداد خود را فرض اینکه هر رقم نقلی از هیچ p بیتی عبور نمی کند محاسبه می کنند. p<n یک پارامتر طراحی است.با کاهش p، جمع کنندۀ تقریبی سریعتر می شود اما سرعت خطا افزایش میابد. جمع کننده های تقریبی با استفاده از زیرجمع کننده های کوچک متعدد که به موازات هم عمل می کنند به بخش های مختلف تقسیم می شوند. جمع کنندۀ فرعی iام مجموع بیت های ri را تولید می کند که در نتیجۀ نهایی نقش دارد و از بیت های pi مورد استفاده جهت پیش بینی رقم نقلی (تو جه داشته باشید که جمع کنندۀ فرعی 0# یک استثنا است که با تمام توان خروجی خود به نتیجۀ نهایی کمک می کند) بهره می گیرد. اندازۀ هر جمع کنندۀ فرعی ri+pi می باشد.

ما در این مقاله به بررسی کاربرد جمع کننده های تقریبی در مرحلۀ نهایی یک انباشتگر مولتی (MAC) ذخیرۀ رقم نقلی خواهیم پرداخت.اهمیت این اپلیکیشن از آن جهت است کهMACها از بلوک های ساتمانی رایج در ایجاد پیچیدگی هستند. پیچیدگی یک عملیات پایه در پردازش سیگنال مولتی مدیاست. به طور ویژه، ما بر MAC طراحی شده برای اپلیکیشن فیلترینگ عکس تمرکز خواهیم کرد. سایر مقالات به عملکرد جمع کننده های تقریبی اپلیکیشن فیلترینگ عکس خواهند پرداخت. اما چندین فرضیۀ ساده سازی شده که در اپلیکیشن های عملی به سختی یافت می شوند را اجرا خواهند کرد. مثلا در پژوهش (11) از جمع کننده های تقریبی برای پردازش عکس استفاده می شود اما این فرض نیز در نظر گرفته می شود که تفریق ها و ضرب ها با واحدهای تابعی دقیق اجرا می شوند. از فرضیۀ مشابهی در (9) استفاده شده است.

ما در این پژوهش یک معماری واقع گرایانه از MAC که توسط درخت Wallace ذخیرۀ رقم نقلی برای تراکم محصولات جزئی درست شده پس از یک جمع کنندۀ منتشر کنندۀ رقم نقلی نهایی (تقریبی) لحاظ خواهیم کرد. مدار با کمک ابزارهای سنتر اتوماتیک طراحی می شود که از توصیف HDL اغاز می گردد. در گام اول، یک جمع کنندۀ انتشار رقم نقلی دقیق مورد استفاده قرار می گیرد. نت لیست سنتزشده سپس برای

استخراج آمار و ارقام موارد افزوده شده شبیه سازی می گردد. سپس جمع کنندۀ تقریبی را جهت یافتن مشخصه های خطایی مناسب با توجه به آمار ورودی ها طراحی خواهیم کرد. نت لیست سنتزشده در نهایت با جایگزینی جمع کنندۀ دقیق با جمع کنندۀ تقریبی اصلاح می گردد و یک سنتز نهایی و بهینه سازی انجام می شود.



شکل 1: تقسیم بندی جمع کنندۀ Speculative. یک جمع کنندۀ n=18 با استفاده از سه جمع کننده فرعی به قسمت های مختلف تقسیم می شود. هر جمه کنندۀ فرعی بیت های مجموع r را تولید می کند که در نتیجۀ نهایی نقش دارد و از بیت های p برای پیش بینی رقم نقلی بهره می گردد.

علاوه بر جریان طراحی جدید، دیگر یافته های پژوهش حاضر به شرح زیر خلاصه می شود.اولا،ما نشان می دهیم که در یک اپلیکیشن پردازش نوعی عکس، ورودی های جمع کنندۀ امتشار رقم نقلی از وضعیت انتشار یکنواخت و گاوسی دور هستند. در نتیجه، فرمول های موجود برای پیش بینی خطا [5]،[12] , [11] برای قضاوت در مورد عملکرد واقعی mAC کفایت نمی کنند. دوما، ما مشاهده می کنیم که در خصوص برخی فیلتهای شالوده ای، خطاهای ایجاد شده توسط جمع کنندۀ تقریبی منجر به نویز تصویری محسوس می گردد و ما یک رویکرد ساده برای کاهش این پدیده پیشنهاد می دهیم. سوما، ما نشان می دهیم که توزیع خاص زمان ورود ورودی های جمع کنندۀ انتشار رقم نقلی، پیشرفت های عملکردی مربوط به کاربرد جمع کنندۀ تقریبی را محدود می کند.

مقاله حاضر شامل بخش های زیر می باشد: بخش 2: ارائۀ جریان طرح پیشنهادی و mAC مورد استفاده به عنوان نمونۀ آزمایش. بخش 3: گزارش عملکرد خطای پیکربندی جمع کنندۀ تقریبی و شالوده های مختلف فیلتر.بخش 4: ارائه و تشریح نتایج پیاده سازی سخت افزار در تکنولوژی 28nm.

2. جریان طرح

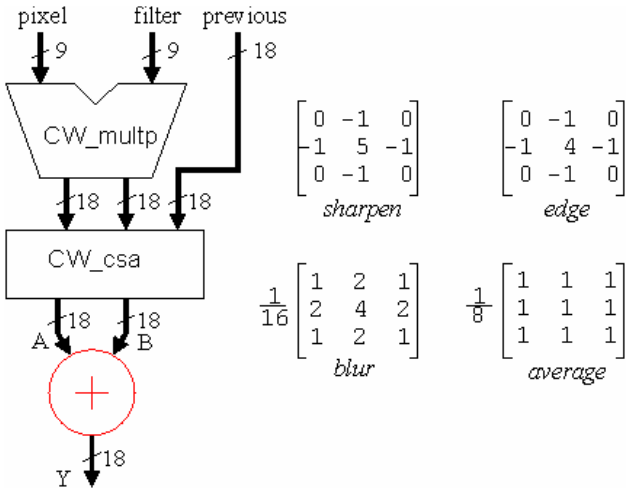
ساختار انباشتگر مولتی پلایر بررسی شده در شکل 2 نشان داده شده است. ما از تشریح HDL مدار آغاز می کنیم که بعدا توسط سنتسایزر با جزئیات ساخته می شود. سنتسایزر از استخراج دیتاپد برای تغییر عملگرهای حسابی (در این مورد افزودن و ضرب کردن) به بلوک های بهینه با استفاده از حسابی ذخیرهٔ عدد نقلی جهت بهبود عملکرد بهره می گیرد. جمع کنندهٔ انتشار رقم نقلی برای جمع کردن خروجی های آنی مرحلهٔ ذخیرهٔ عدد نقلی که نتیجه نهایی را می دهد استفاده می شود. متاسفانه طراح در هنگام سنتز یک سازهٔ سطح بالا مثل y<=A+B*C در VHDL، هیچ دسترسی به سیگنال های آنی ذخیرهٔ عدد نقلی ندارد.از اینرو برای توصیف MAC به استفاده از اجزاء حسابی IP موجود در ابزار سنتز متوسل می شویم. به ویژه از اجزاء IP ChipWare متعلق به Cadence Encounter RTL Compiler [13] (شبیه IPهای موجود در سایر ابزارهای سنتز مثل DesignWare Building Block IP در Synopsys Design Compiler [14])استفاده کردیم.

MAC جهت استفاده از تصاویر 8بیتی بهینه سازی می شود. ما از یک مولتی پلایر علامت دار استفاده کردیم. به گونه ای که ورودی پیکسل در شکل 2 به صورت مقدار 9 بیتی (MSB آن صفر است) نشان داده شده است. ضرایب فیلتر علامت دار می شوند و از 4بیت کسری (LSB=2□4) برای ارائهٔ راحت شالوده های 3x3 استفاده می شود (برخی از این شالوده ها در شکل نیز گزارش شده اند). بلوک CW_multp در شکل 2 مولتی پلایر محصول نیمه تمام است در حالی که CW_csa جمع کنندهٔ ذخیرهٔ رقم نقلی است که برای جمع زدن افزوده شوندهٔ 18 بیتی به خروجی های مولتی پلایر ذخیرهٔ رقم نقلی استفاده می شود. لطفا توجه داشته باشید که ابعاد دیتاپد برای جلوگیری از سرریز حین اجرای فیلترینگ تصویر مشخص می گردد. جمع کنندهٔ انتشار عدد نقلی که با رنگ قرمز در شکل 2 مشخص شده است خروجی MAC یعنی Y را تولید می کند.

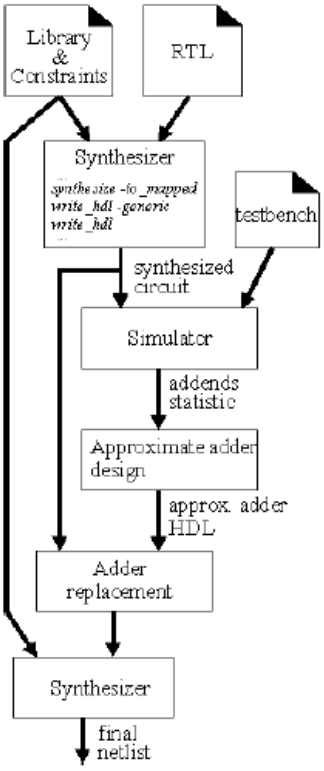
جریان طرح پیشنهادی در شکل 3 نشان داده شده است. ما با توصیف ساختاری سادهٔ MAC HDL شروع می کنیم.سپس محدودیت ها را اعمال می کنیم و مدار را سنتز می کنیم. در اینجا نت لیست سنتز شده را جهت استخراج امار دو جمع شونده A و B که جمع کنندهٔ انتشار عدد نقلی نهایی افزوده شدند شبیه سازی می کنیم. لطفا توجه داشته باشید که این شبیه سازی را نمی توان با توصیف اولیهٔ HDL انجام داد. سنتسایزر یا ترکیب کننده در حقیقت معماری متناسب را برای مولتی پلایر محوصول نسبی بسته به محدودیت های اعمال شده

انتخاب می کند. از اینرو مقدار A و B با مقدار بدست آمده از مدل های شبیه سازی تفات دارد (هرچند مجموع آنها یکی است).

در شکل 4 گزارشی از توزیع A و Bدر هنگام اجرا فیلرینگ حاشیه ای روی تصویر آزمایشی Lena ارائه شده است. همانطور که مشاهده می کنید توزیع نه به صورت یکنواخت و نه گاوسی است. در نتیجه فرمول های ارائه شده در پژوهش های موجود برای پیش بینی احتمالا خطا، جهت قضاوت عملکرد واقعی MAC کفایت نمی کنند.



شکل 2. ساختار MAC. CW_multp مولتی پلایر محصول نسبی است. در حالی که CW_csa جمع کنندهٔ ذخیرهٔ عدد نقلی. جمع کنندهٔ نهایی انتشار عدد نقلی با رنگ قرمز مشخص شده است.



شکل 3: جریان طرح پیشنهادی

برای طراحی جمع کنندۀ تقریبی، ما دو مجموعه مقدار را از داده های شبیه سازی شده به صورت عددی کسر می کنیم: احتمال داشتن یک انتشار عدد نقلی از بیت i به بیت j (به صورت pi,j نشان داده می شود) و احتمال داشتن یک تولید عدد نقلی از بیت i به بیت j (بصورت gi,j نشان داده می شود) . از روی این مقادیر می توانیم احتمالا خطای جمع کنندۀ تقریبی را از لحاظ عدد ی محاسبه کنیم. به جمع کنندۀ شکل 1 توجه کنید. جمع کنندۀ فرعی #0 (با محاسبۀ LSBs 9) دقیق است. در جمع کنندۀ فرعی #1 یک خطا در صورتی رخ می دهد که عدد نقلی خارج از جمع کنندۀ فرعی #0 بالا باشد. در حالی که رقم نقلی c9 که توسط جمع کنندۀ فرعی #1 محاسبه می شود پایین باشد. در این وضعیت، یک carry-out حاصل از موقعیت بیت #4 وجود دارد که در سرتاسر #5,6,7,8 منتشر می گردد. این وضعیت دارای احتمال P(E1)=g0,4 p5,8 است. در جمع کنندۀ فرعی #2، خطا زمانی رخ می دهد که رقم نقلی جمع کنندۀ فرعی#1 بالا باشد. در حالی که رقم نقلی c14 که توسط جمع کنندۀ فرعی #2 محاسبه شده پایین باشد. از اینرو، ما زمانی خطا داریم که یم عدد نقلی از موقعیت بیت #7 حاصل گردد که در سرتاسر بیت های #8 تا #13 منتشر گردد و این با احتمال P(E2)=g0,7 p8,13 رخ می دهد.

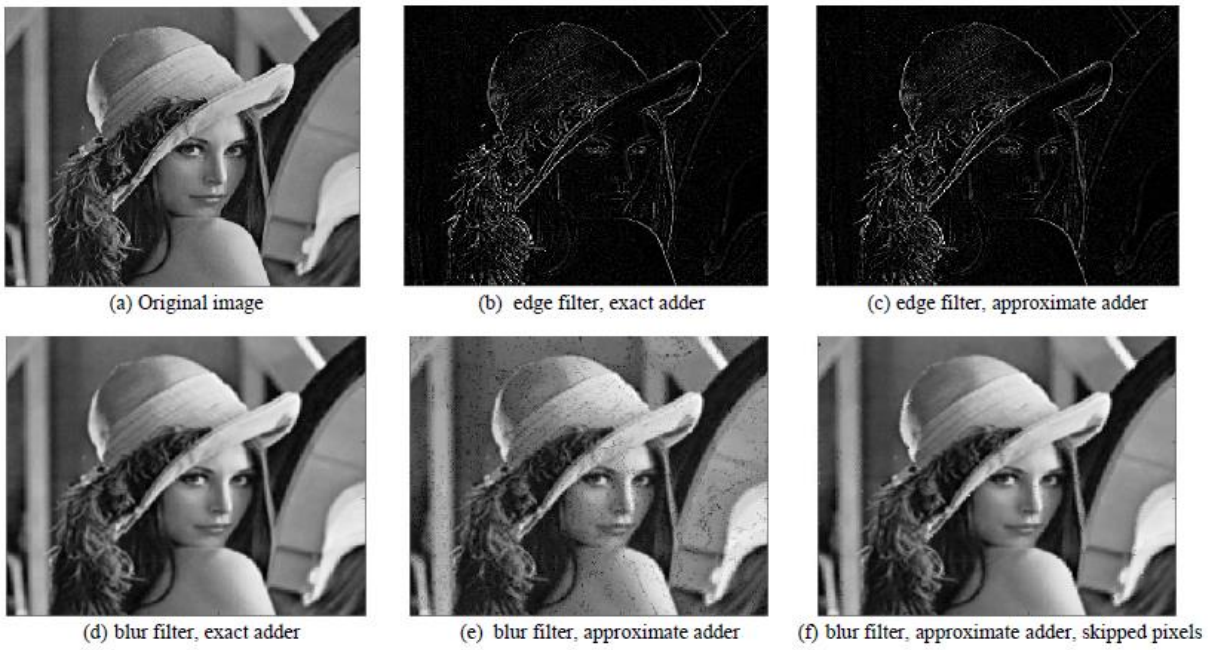
پیرو [6]، احتمال خطای کلی به صورت زیر می باشد:

$$P(E)=P(E1 \cup E2)=P(E1)+P(E2)-P(E1 \cap E2)$$

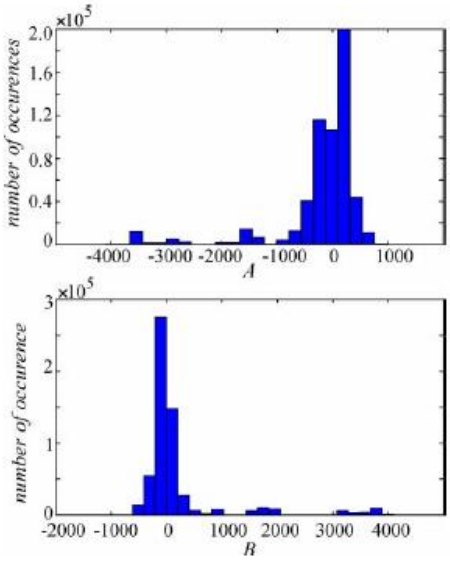
محسابات ساده در این مثال منجر به $P(E)=g_{0,4}\ p_{5,8}+g_{5,7}\ p_{8,13}$ می گردد. ما با تعمیم دهی این این روش، می توانیم پیکربندی های مختلف جمع کنندۀ تقریبی را پس از تعیین مشخصات توزیع سیگنال های A و B به سرعت بررسی کنیم. در ادامه، پیچیدگی با شالودۀ 3×3، اجرای 9 ضرب یا افزایش برای هر پیکسل تصویر فیلترشده را لحاظ می کنیم. از اینرو احتمال محاسبۀ پیکسل خطادار را می توان به صورت $9P(E)$ محاسبه کرد. قابل ذکر است که پس از پیچیدگی مقدار هر پیکسل محدود به یک مقدار صحیح 8 بیتی می گردد. یعنی مقدار منفی محدود به صفر می شود و مقدار مثبت تا 225 اشباع می گردد. در نتیجه تمام خطاهای موجود در پیچیدگی منجر به پیکسل خطادار در تصویر خروجی نمی شود.

در شکل 5 برخی نتایج حاصل از اجرای فیلترینگ تصویر با استفاده از جمع کنندۀ تقریبی شکل 1 به عنوان جمع کنندۀ انتشار عدد نقلی MAC نشان داده شده است. جمع کنندۀ تقریبی در خصوص فیلتر حاشیه به خوبی عمل می کند. کیفیت تصویر را می تواند با شاخص شباهت ساختاری (Structural Similarity Index) یا SSI تعین کرد. مقدار SSIM=1 یعنی شباهت کامل بین دو تصویر و SSIM=0.98 برای دو تصویر موجود در شکل 5(b) و 5(c). با فیلتر blur، استفاده از جمع کنندۀ تقریبی در عوض منجر به پارازیت تصویری مشهود به شکل پیکسل های سیاه ساختگی با SSIM=0.81 برای هر دو تصویر موجود در شکل 5(d) و 5(e) می گردد. ما یک شیوۀ ساده اما کارامد برای کاهش این اثر تصنعی پیشنهاد می دهیم. ما جمع کنندۀ تقریبی را با یک مداربندی ساده جهت یافتن وضعیت خطا تقویت می کنیم و در این مسیر عدد نقلی جمع کننده های فرعی #1 و #2 و عدد نقلی c9 و c14 (را همانطورکه قبلا تضیح دادیم) بررسی خواهیم کرد. بعد از اینکه خطا پیدا شد، پیچیدگی نادیده گرفته می شود و پیکسل فیلتر شدۀ قبلی خروجی کار می شود. همانطور که در شکل 5(f) نشان داده شده این تکنیک ساده با SSIM=0.98 برای هر دو تصویر موجود در شکل 5(d) و 5(f) کارآمد است.

پس از تکمیل طرح جمع کنندهٔ تقریبی طبق آنچه در شکل 3 نشان داده شده، ما نت لیست را با جایگزینی جمع کنندهٔ دقیق با جمع کنندهٔ تقریبی اصلاح می کنیم. دیگر مرحلهٔ بهینه سازی و سنتز منجر به نت لیست نهایی MAC می گردد.



شکل 5: نتایج فیلترینگ عکس



شکل 4: توزیع ورودی های A و B جمع کنندهٔ انتشار عدد نقلی در حین اجرای یک فیلترینگ حاشیه ای روی تصویر آزمایشی لنا

3. نتایج پیاده سازی VLSI

ما MAC را با جمع کنندهٔ دقیق و تقریبی به کمک تکنولوژی STM 28nm، استاندارد VT سنتز کردیم. محدودیت هایی با هدف دستیابی به یک مساحت حداقلی، طراحی کم قدرت پیشنهاد دادیم.

در جدول ض، نتایج سنتز آورده شده است. همانطور که مشاهده می شود، سرعت 19٪ افزایش داشته که با افزایش 4-5 درصدی در مساحت و قدرت مواجه بوده. ما می توانیم سرعت را به خاطر قدرت معاوضه کنیم. به ردیف آخر جدول 1 مراجعه کنید. با کاهش ولتاژ منبع به 0.91V، طرحی که از جمع کنندهٔ تقریبی استفاده کرده دارای مشابه با MAC با جمع کنندهٔ دقیق با بهرهٔ توان 14% می باشد. این بهبود عملکرد احتمالا کمتر از آنچه انتظار می رفت می باشد.

برای بررسی بهتر این رفتار، شکل 6 زمان ورود ورودی های هر جمع کنندهٔ انتشار رقم نقلی را نشان می دهد. همینطور که مشاهده می شود، علائم متناظر با بیت های میانی عدد افزوده دیرتر از سایر موارد می رسند. این وضعیت تا حدودی بر مزایای سرعت مربوط به تجزیهٔ جمع کنندهٔ فرعی جمع کننده های تقریبی غلبه می کند.

4 نتیجه گیری

ما کاربرد افزایشگرهای تقریبی در مرحلهٔ نهایی MAC ذخیرهٔ عدد نقلی طراحی شده برای اپلیکیشن فیلترینگ عکس را بررسی کردیم. یک بهرهٔ نیروی 14٪ در ولتاژ برای MAC کوچک مد نظر در این مقاله بدست آمد. برای دستیابی به حداکثر بهبود عملکرد افزایشگر تقریبی با لحاظ توزیع دفعات ورود ورودی ها به بررسی های بیشتری نیاز است.

REFERENCES

[1] Shih-Lien Lu, "Speeding up processing with approximation circuits," Computer , vol.37, no.3, pp.67,73, Mar 2004.

[2] H. Jiang, J. Han, F. Lombardi, "A comparative review and evaluation of approximate adders", Proc. of Great Lakes Symposium on VLSI, pp. 343-348, Pittsburgh, 2015.

[3] D. Esposito, D. De Caro, E. Napoli, N.Petra, A.G.M. Strollo, "Variable Latency Speculative Han-Carlson Adder", IEEE Trans. on Circuits and Systems I: Regular Papers, vol. 62, n. 5, pp. 1353-1361, May 2015.

[4] D. Esposito, D. De Caro, A.G.M. Strollo, "Variable Latency Speculative Parallel Prefix Adders for Unsigned and Signed Operands", IEEE Trans. on Circuits and Systems I: Regular Papers, vol. 63, n. 8, Aug. 2016.

[5] V. Gupta, D. Mohapatra, A. Raghunathan, K. Roy, "Low-Power Digital Signal Processing Using Approximate Adders", IEEE Trans. on CAD, vol. 32, no. 1, pp. 124-137, jan. 2013

[6] M. Shafique, W. Ahmad, R. Hafiz, J. Henkel, "A low-latency generic accuracy configurable adder", proc. of Design Automation Conf., DAC'15, pp. 86:1-86:6, June 2015, San Francisco.

[7] A. K. Verma, P. Brisk, P. Ienne, "Variable Latency Speculative Addition: A New Paradigm for Arithmetic Circuit Design," proc. of Design, Automation and Test in Europe, DATE '08, pp.1250-1255, March 2008.

[8] N. Zhu, W. L. Goh, G. Wang, K. S. Yeo, "An enhanced low-power high-speed adder for error tolerant applications", proc. of International Symp. on Integrated Circuits, (ISIC), pp. 67-69, 2009.

[9] A. B. Kahng, S. Kang, "Accuracy configurable adder for approximate arithmetic design", proc. of Design Automation Conf., DAC'12, pp.820-825, June 2012, San Francisco.

[10] K. Du, P. Varman, K. Mohanram, "High performance reliable variable latency carry select addition," Design, Automation and Test in Europe DATE'12, pp.1257-1262, March 2012.

[11] C. Liu, J. Han, F. Lombardi, "An analytical framework for evaluating the error characteristics of approximate adders", IEEE Trans. on Computer, vol. 64, n. 5, pp.1268-1281, May 2015.

[12] D. Esposito, G. Castellano, D. De Caro, E. Napoli, N. Petra and A. G. M. Strollo, "Approximate adder with output correction for error tolerant applications and Gaussian distributed inputs," 2016 IEEE ISCAS Conf., Montreal, QC, 2016, pp. 1970-1973.

[13] ChipWare IP Components in Encounter® RTL Compiler, Cadence, Product Version 14.2, August 2015.

[14] Synopsys DesignWare Building Block IP User Guide, 2009.

[15] Zhou Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," in IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600-612, April 2004.

[16] V. G. Oklobdzija, D. Villeger, S. S. Liu, "A Method for Speed Optimized Partial Product Reduction and Generation of Fast Parallel Multipliers Using an Algorithmic Approach", IEEE Transactions on Computers, Vol. 45 n. 3, March 1996, pp.294-306