افزایش امنیت ماژول رمزنگاری با قابلیت بازپیکربندی جزئی

منوچهر صادقی آهنگری[۱]، محمد امین امیری[۲]
[۱] کارشناسی ارشد، دانشگاه صنعتی مالک اشتر، مجتمع دانشگاهی برق و کامپیوتر
[۲] استادیار، دانشگاه صنعتی مالک اشتر، مجتمع دانشگاهی برق و کامپیوتر، maamiri@mut.ac.ir

چکیده

در این مقاله به ارائه روشی برای افزایش امنیت در سیستم‌های مخابراتی پرداخته شده است که شامل استفاده از ۴ الگوریتم رمزنگاری به جای یک الگوریتم رمزنگاری می‌باشد. این موضوع سبب می‌شود که دنباله به کارگیری شده از خروجی هر الگوریتم رمزنگاری به یک چهارم کاهش پیدا کند. برای پیاده‌سازی این روش از یک سیستم بر تراشه به نام ZYNQ از شرکت Xilinx استفاده می‌گردد و FPGA این تراشه برای پیاده‌سازی الگوریتم‌ها مدنظر است. استفاده از منابع سخت افزاری بیشتر (بدلیل استفاده از ۴ الگوریتم رمزنگاری) برای ما چالش به حساب آمده و محدودیت‌هایی را اضافه می‌کند که برای حل این مساله از تکنیک بازپیکربندی جزئی استفاده می‌گردد. با توجه به پیاده‌سازی سخت‌افزاری الگوریتم‌ها، پارامترهای زمانی مناسبی از این طرح بدست آمده است. نتایج نشان می‌دهد که علاوه بر بهبود پارامترهای زمانی، از نظر منابع سخت‌افزاری نیز سربار چندانی به طرح اضافه نشده است و از همه مهمتر امنیت بالاتری را در ارتباط به دست آورده‌ایم. در مقایسه با پیاده‌سازی کامل طرح، بهبود ۶۳ درصدی در میزان استفاده از منابع سخت‌افزاری انجام گرفته است.

کلیدواژه

ماژول رمز، افزایش امنیت، بازپیکربندی جزئی، سیستم بر تراشه

مقدمه

رمزنگاری دانش تغییردادن متن اطلاعات به کمک کلید رمز و با استفاده از یک الگوریتم رمز است، به صورتی که تنها شخصی که از کلید و الگوریتم رمزنگاری مطلع است قادر به استخراج اطلاعات اصلی از اطلاعات رمز شده باشد و شخصی که از حداقل یکی از آن‌ها اطلاع ندارد، نتواند به اطلاعات دسترسی پیدا کند [۱]. دانش رمزنگاری بر پایه مقدمات بسیاری از قبیل تئوری اطلاعات، نظریه اعداد و آمار بنا شده است و امروزه به طور خاص در علم مخابرات مورد بررسی و استفاده قرار می‌گیرد. با رشد روزافزون و سریع فناوری‌های وابسته به مخابرات در دهه‌های اخیر و تحولات ایجاد شده در پدیده ارتباطات، اهمیت مساله حفاظت اطلاعات و امنیت ارتباطات هر روز بیش از پیش جلوه می‌کند. سیستم‌های رمزنگاری وظیفه ایجاد امنیت در سیستم‌های مخابراتی را بر عهده دارند. عمل رمزنگاری معمولا به این صورت است که یک الگوریتم رمز با استفاده از کلید رمز در مد از پیش تعریف شده، اجرا شده و دنباله خروجی را تحویل می‌دهد.

تولید بیشتر دنباله‌ی خروجی رمز، همواره متناظر با کاهش امنیت دنباله‌ی خروجی می‌باشد. به این معنی که اگر طی دو اجرای الگوریتم رمزنگاری، m بیت و n بیت از خروجی الگوریتم رمز دریافت نماییم به ازای $n < m$ دنباله m بیتی امنیت کمتری از دنباله‌ی n بیتی خواهد داشت. به این ترتیب بهتر است که برای افزایش امنیت دنباله‌ی‌رمز، طول دنباله خروجی به کارگیری شده از الگوریتم را کاهش دهیم. در این راستا به کارگیری مثلا چهار الگوریتم رمزنگاری جهت تولید دنباله خروجی منجر به کاهش خروجی هر الگوریتم شده و امنیت بالاتری را تامین می‌کند. لازم بذکر است که استفاده از چهار الگوریتم به جای یک الگوریتم، سربار سخت‌افزاری زیادی را به سیستم اعمال می‌کند.

از طرف دیگر همانطور که می‌دانیم پیشرفت تکنولوژی VLSI طی چند دهه گذشته بدون تغییر در الگوریتم‌ها یا کدهای نرم‌افزارهای کاربردی، باعث افزایش کارایی بسیاری از آن‌ها شده است. اما در سال‌های اخیر به دلیل برخورد تکنولوژی VLSI با دو دیوار بزرگ حافظه و توان، شیب بهبود عملکرد پردازنده‌های همه منظوره کاهش یافته است و همچنین پیاده‌سازی تمام نرم‌افزاری طرح، در مقایسه با استفاده از سخت‌افزار بسیار کند می‌باشد [۲]. اما یکی از راه‌کارهایی که در این زمینه ارائه می‌گردد، استفاده از پردازنده‌های چند هسته‌ای است که با مطرح شدن پردازش موازی عملا پیچیدگی را از سطح سخت‌افزار به سطح دستورالعمل می‌برد [۳]. استفاده

تاریخ دریافت: ۱۳۹۶/۱۰/۱۶ تاریخ پذیرش: ۱۳۹۷/۱۱/۲۱

از پردازنده‌های گرافیکی که برای اعمال زمانبری نظیر پردازش تصویر کارآیی دارد نیز از دیگر راهکارهای موجود در این حوزه می‌باشد [۳]. اما راه کار بهتر و مفیدتر که در این مقاله بیشتر راجع به آن صحبت می‌کنیم و تقریبا در همه نوع طرح می‌توان از آن استفاده کرد، بکارگیری شتاب دهنده‌های سخت‌افزاری در کنار پردازنده‌ها است که فعالیت‌های زمانبر پردازنده را انجام می‌دهد [۳]. شتاب دهنده سخت‌افزاری که در کنار پردازنده قرار می‌گیرد یا بصورت ثابت است و یا بصورت متغیر. تراشه‌های DSP از نوع شتاب دهنده‌های ثابت هستند که با وجود داشتن سرعت بالاتر نسبت به نوع متغیر آن دارای ایراد عدم انعطاف پذیری هستند. به کارگیری شتاب دهنده‌های سخت‌افزاری متغیر مانند FPGA در کنار پردازنده‌ها، این محدودیت را از بین برده و طراح می‌تواند هر تابع زمانبر سیستم را در FPGA پیاده‌سازی نماید. مساله محدودیت منابع سخت‌افزاری، محدودیتی است که در این جا مطرح می‌شود و ایده بازپیکربندی جزئی تا حد قابل قبولی این محدودیت را از بین می‌برد.

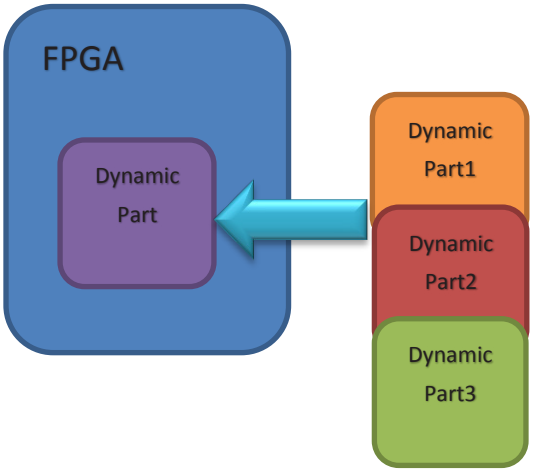
هرگاه پردازنده و حافظه‌های مربوطه، شتاب‌دهنده سخت‌افزاری و همچنین تجهیزات جانبی ارتباط پردازنده با دنیای بیرون در یک چیپ قرار گیرد، به این مجموعه سیستم بر تراشه می‌گوییم [۴]. در این طرح ما یک واحد رمز را با قابلیت بازپیکربندی جزئی که در تراشه ZYNQ [۴] که یک سیستم بر تراشه می‌باشد با استفاده از بسته نرم‌افزار VIVADO طراحی کردیم. از نرم افزار VIVADO جهت سنتز و پیاده‌سازی روی FPGA مربوطه و از نرم‌افزار SDK جهت برنامه‌نویسی روی میکروکنترلرهای آن سیستم بر تراشه استفاده کردیم. در این طرح ما با استفاده از چهار الگوریتم به جای یک الگوریتم استفاده از دنباله‌ی رمز خروجی الگوریتم را به یک چهارم کاهش دادیم. استفاده از قابلیت بازپیکربندی جزئی [۵] سبب صرفه جویی در منابع سخت‌افزاری موجود در FPGA می‌گردد و در عمل به ازای استفاده از هر تعداد الگوریتم رمزنگاری، تنها به اندازه منابع سخت‌افزاری بزرگترین الگوریتم رمزنگاری منابع سخت‌افزاری استفاده می‌گردد. به این ترتیب، در این مقاله واحد رمز در بستر یک سیستم بر تراشه که مزایای آن گفته شد با قابلیت بازپیکربندی‌جزئی ارائه می‌گردد. افزایش امنیت لینک های ارتباطی رادیویی بیسیم که مصرف توان و ابعاد مدارات در آنها اهمیت زیادی دارد، از مهمترین کاربردهای این طرح می باشد. لازم بذکر است که مد اصلی بکارگیری شده، مد ECB می باشد ولی امکان بکارگیری سایر مدها نظیر CNT و ... هم وجود دارد. امکان تغییر مد کاری الگوریتم توسط ماژول های بازپیکربندی جزئی وجود دارد. در ارتباط با تبادل کلید بین فرستنده و گیرنده، از مدل کلیدهای از پیش توزیع شده استفاده شده است و فرآیند توزیع کلید بخشی از پیاده سازی نمی باشد.

در ادامه به مروری بر کارهای قبلی که در این زمینه ارائه گردید می‌پردازیم. تاماس [۶] از بازپیکربندی جزئی در استفاده از ۲ فیلتر Median و FIR و همچنین محاسبه آستانه برای پردازش تصویر در تراشه XC5VSX50T از خانواده Virtex-5 شرکت Xilinx با قابلیت بازپیکربندی جزئی استفاده نمود. نتایج نشان می‌دهد که پیاده‌سازی سخت‌افزاری در مقایسه با پیاده‌سازی نرم‌افزاری برای فیلتر Median تا ۷۰ برابر سریعتر، برای فیلتر FIR تا ۳۰ برابر سریعتر و برای محاسبه آستانه بدون افزایش سرعت عمل کرده‌اند. لازم بذکر است که مجموع سه بخش فوق، حدود ۱۶۹٪ از LUTها و حدود ۱۶۰٪ از FlipFlopهای تراشه را به خود اختصاص داده‌اند و طبیعتاً امکان پیاده‌سازی همزمان آنها روی این تراشه وجود نداشت. وانخاده [۷] تمرکزش را بر روی الگوریتم رمزنگاری AES قرار داد و الگوریتم‌های AES با طول بلوک ۱۲۸ بیت، ۱۹۲ بیت و ۲۵۶ بیت را با استفاده از تکنیک بازپیکربندی جزئی و به عنوان قسمت پویا پیاده‌سازی نمود. نتایج بدست آمده از تحقیق ایشان به این صورت است که الگوریتم‌های ۱۲۸ بیتی، ۱۹۲ بیتی و ۲۵۶ بیتی به ترتیب به توان‌های عملیاتی ۲.۰۹ مگابیت بر ثانیه، ۱.۸۳ مگابیت بر ثانیه و ۱.۸۱ مگابیت بر ثانیه دست پیدا کرده‌اند. البته تراشه مورد استفاده ایشان، XC5VLX110T از خانواده Virtex-5 شرکت Xilinx بود که مشکلی از بابت تعداد و حجم منابع سخت‌افزاری طرح ایجاد نکرده است. رودریگز [۸] از تکنیک بازپیکربندی‌جزئی در بلوک کدگشای Jpeg برای کاهش منابع سخت‌افزاری استفاده کرد. نتایج تحقیقات ایشان نشان می‌دهد که با تقسیم کل فرآیند به پنج زیربخش، سخت‌افزار با استفاده از تکنیک بازپیکربندی جزئی حدود ۴۰٪ از سخت‌افزار اولیه را اشغال نمود ولی راه‌حل با استفاده از بازپیکربندی جزئی حدود ۹ برابر کندتر از راه‌حل ایستا عملیات کدگشایی را به انجام رساند. پیاده‌سازی این طرح روی تراشه‌ای از خانواده Zynq7000 شرکت Xilinx انجام شده است. مینال [۹] پیاده‌سازی سخت‌افزاری الگوریتم AES را روی تراشه XC3S400 از خانواده Spartan-3 شرکت Xilinx انجام داده است که با ماکزیمم فرکانسی کاری ۱۶۰ مگاهرتز، به توان عملیاتی ۲.۰۵۹ گیگابیت بر ثانیه رسیده است. از طرف دیگر ارکیت [۱۰] با پیاده‌سازی نرم‌افزاری که از الگوریتم AES روی تراشه ARM با عنوان Cortex-M4 انجام داد، به توان عملیاتی ۱۰۳۸۷۸ بیت بر ثانیه در رمزنگاری دست یافت.

بازپیکربندی جزئی

با پیچیده‌تر شدن هر چه بیشتر سیستم‌های دیجیتال، علی‌رغم تلاش‌های بسیار به منظور پیاده‌سازی هر چه بهینه‌تر این

سیستم‌ها، نیاز به سخت‌افزارهایی با منابع بیشتر، بیش از پیش احساس می‌شود. در گذشته در صورتی که نیاز به تغییر تنها قسمتی از تراشه FPGA بود، باید کل تراشه مجددا پیکربندی می‌شد. علاوه بر این، عملکرد کل مدار متوقف می‌شد تا این تراشه‌ها دوباره پیکربندی شوند و همچنین محتوای تمام ثبات‌های تراشه FPGA در حین این فرآیند از دست رفته و قابل بازیابی نبودند. شرکت Xilinx قابلیت بازپیکربندی‌جزئی را برای تراشه‌های FPGA خانواده Virtex-4 به بعد ارائه کرده است [۵]. همانطور که در شکل ۱ مشاهده می‌شود به واسطه این قابلیت می‌توان تنها بخشی از تراشه FPGA، موسوم به نواحی پویا را بدون ایجاد خلل در عملکرد نواحی موسوم به ایستا بازپیکربندی نمود.



شکل ۱. تسهیم زمانی واحد های باز پیکربندی

یکی از مهمترین مزایای این روش تسهیم زمانی منابع موجود در FPGA می‌باشد. این بدین معناست که بر خلاف روش متداول که تمامی ماژول‌ها پیاده‌سازی می‌گردند و منابع سخت‌افزاری FPGA را استفاده می‌کنند، در این روش با یکی از این ماژول‌ها که معمولا بزرگترین ماژول سخت‌افزاری FPGA می‌باشد، طرح را ایجاد کرده و مراحل بازپیکربندی را برای تمامی ماژول‌ها انجام می‌دهیم. در زمان اجرای طرح با ماژول پیش فرض، با انجام فرآیند بازپیکربندی‌جزئی ماژول‌های دیگر جایگزین می‌گردند و به این ترتیب، به جای پیاده‌سازی تمامی ماژول‌ها مانند حالت متداول، فقط به اندازه یک ماژول (ناحیه بازپیکربندی در حین عملیات بازپیکربندی تعیین می‌گردد) از منابع سخت‌افزاری FPGA استفاده می‌گردد.

پیاده سازی

همانطور که پیشتر نیز ذکر گردید، در این طرح بجای یک الگوریتم رمزنگاری از چهار الگوریتم رمزنگاری استفاده می‌گردد. نکته مهمی که باید مدنظر قرار گیرد، هماهنگی بین فرستنده و گیرنده برای رمزنگاری و رمزگشایی با یک الگوریتم رمز یکسان است. با توجه به یک طرفه بودن ارتباط، لازم است که فرستنده نقش تعیین‌کننده نوع الگوریتم رمزنگاری را برعهده بگیرد.

فرستنده با استفاده از دو بیت داده، تعیین می‌کند که با کدام الگوریتم، داده‌ها را رمز کرده است. گیرنده نیز بر اساس همان الگوریتم، داده‌ها را رمزگشایی می‌کند.

در ادامه باید بخش‌های ایستا و پویای طرح مشخص شود. بخش ایستا شامل مدارات ارتباطی با دنیای بیرون، اجراکننده الگوریتم و فرآیندهای‌کنترلی دیگر می‌باشد. بخش پویا نیز شامل هریک از ۴ الگوریتم‌های AES [۱۱]، Klein [۱۲]، Serpent [۱۳] و Simon [۱۴] می‌باشد. در این جا ۲ مساله برای قسمت ایستای طرح باید مدنظر قرار گیرد:

۱- برای جلوگیری از تبادل اطلاعات در زمان بازپیکربندی جزئی، یک واحد به نام کنترلر بازپیکربندی‌جزئی باید طراحی گردد تا زمان شروع و اتمام بازپیکربندی‌جزئی را نشان دهد و همچنین مساله جداسازی قسمت پویا از قسمت ایستا در حین فرآیند بازپیکربندی را برعهده بگیرد.

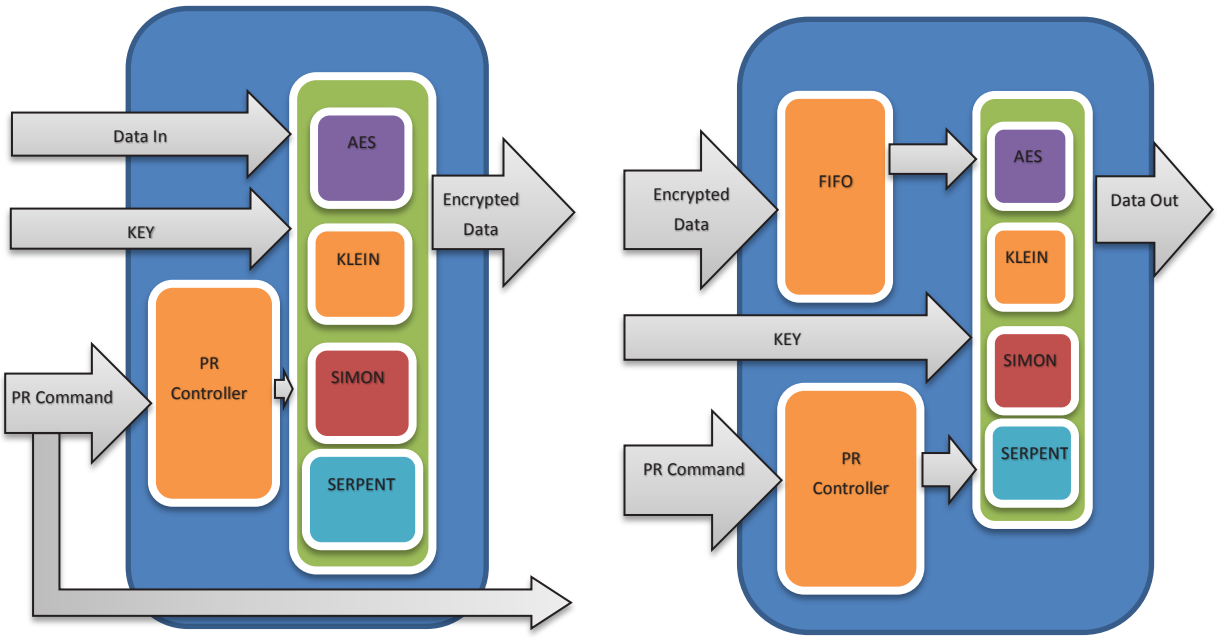
۲- از آنجا که عملیات بازپیکربندی‌جزئی در فرستنده و گیرنده در یک زمان انجام نمی شود، باید حافظه FIFO جهت نگهداری داده‌های دریافتی در گیرنده پیش‌بینی شود. کلیات عملیات به این صورت است که ابتدا فرستنده با انتخاب الگوریتم، فرآیند بازپیکربندی‌جزئی را انجام می‌دهد و سپس داده‌ها را رمز نموده و ارسال می‌کند. گیرنده پس از دریافت بخشی از داده‌ها، الگوریتم رمزنگاری را تشخیص داده و فرآیند بازپیکربندی‌جزئی را آغاز می‌کند. طی این فرآیند، داده‌های دریافتی در FIFO گیرنده ذخیره می‌گردد و گیرنده بعد از اتمام فرآیند بازپیکربندی‌جزئی شروع به خواندن داده‌ها از آن FIFO می‌کند. به این ترتیب با توجه به اینکه فرستنده منتظر اتمام فرآیند بازپیکربندی‌جزئی در گیرنده نمی‌ماند، سرعت ارتباط بالاتر می‌رود.

در شکل ۲ بلوک دیاگرام قسمت رمزکننده در فرستنده و قسمت رمزگشا در گیرنده را مشاهده می‌کنید. همانطور که در این شکل ملاحظه می‌کنید، بلوک کنترلر بازپیکربندی‌جزئی از طریق داده‌های دریافتی، دستور شروع عملیات بازپیکربندی، نوع الگوریتم انتخابی و اتمام فرآیند بازپیکربندی‌جزئی در سمت فرستنده را متوجه می‌شود.

وجود یک واحد FIFO برای جلوگیری از خراب شدن اطلاعات دریافتی در حین فرآیند بازپیکربندی‌جزئی الزامی می‌باشد.

بررسی نتایج

در این بخش بررسی نتایج را در دو قسمت می توان انجام داد. نخست بررسی توان عملیاتی با مراجع دیگر که در مقدمه ذکر گردید را انجام می‌دهیم. با توجه به نتایج ذکر شده در جدول ۱، توان عملیاتی سخت‌افزار این طرح از سایر طرح‌های مرجع بیشتر و بهتر می‌باشد. سپس از لحاظ صرفه‌جویی در منابع سخت‌افزاری نتایج را مورد ارزیابی قرار می‌دهیم. جدول ۲ میزان مصرف منابع سخت‌افزاری مورد نیاز برای هر یک از الگوریتم‌های ذکر شده را نشان می‌دهد. در این جدول همچنین میزان مصرف منابع سخت‌فزاری جهت پیکربندی متداول طرح بدون استفاده از بازپیکربندی جزئی ارائه شده است. مقایسه بین میزان صرفه‌جویی در منابع سخت‌افزاری این طرح و سایر طرح‌های مرجع در جدول ۳ ارائه شده است. میزان متوسط درصد صرفه‌جویی با توجه به کاهش مجموع منابع در حالت بازپیکربندی نسبت به مجموع منابع در حالت متداول محاسبه شده است.



شکل ۲. بلوک دیاگرام رمزکننده و رمزگشا

جدول ۱. مقایسه توان عملیاتی

توان عملیاتی	حداکثر فرکانس کار	نوع پیاده‌سازی	
۲.۰۵۶ گیگابیت بر ثانیه	۱۶۰ مگاهرتز	سخت‌افزاری	مرجع [۹]
۱۰۴ کیلوبیت بر ثانیه	۷۲ مگاهرتز	نرم‌افزاری	مرجع [۱۰]
۲.۶۶۲ گیگابیت بر ثانیه	۲۰۸ مگاهرتز	سخت‌افزاری	روش پیشنهادی

جدول ۲. میزان مصرف منابع سخت‌افزاری برای بخش‌های مختلف قابل پیکربندی بصورت بازپیکربندی‌جزئی و همچنین پیکربندی متداول طرح پیشنهادی

منابع سخت افزاری	منابع سخت افزاری AES	منابع سخت افزاری KLEIN	منابع سخت افزاری SERPENT	منابع سخت افزاری SIMON	منابع پیکربندی متداول	منابع بازپیکربندی جزئی
تعداد LUTها	۸۳۲۷	۲۵۲۴	۸۷۲۲	۸۵۷۲	۲۸۱۴۵	۸۷۲۲
تعداد FlipFlopها	۲۲۴۴	۹۶۴	۵۲۷۸	۱۰۹۲	۹۵۷۸	۵۲۷۸

جدول ۳. مقایسه میزان صرفه‌جویی در مصرف منابع سخت‌افزاری طرح پیشنهادی و سایر طرح‌ها در بخش قابل پیکربندی

	منابع پیکربندی متداول LUTها	منابع پیکربندی متداول FlipFlopها	منابع بازپیکربندی جزئی LUTها	منابع بازپیکربندی جزئی FlipFlopها	متوسط درصد صرفه‌جویی
طرح پیشنهادی	۲۸۱۴۵	۹۵۷۸	۸۷۲۲	۵۲۷۸	٪۶۳
مرجع [۶]	۱۶۱۲	۱۵۳۲	۶۶۴	۶۰۰	٪۵۹
مرجع [۸]	۱۶۳۲	۳۲۴۲	۱۳۸۹	۱۱۶۳	٪۴۷
مرجع [۷]	۶۴۰ Slices		۲۴۱ Slices		٪۶۲

همانطور که در این جدول دیده می‌شود، میزان صرفه جویی در منابع سخت‌افزاری این طرح از سایر طرح‌های مرجع بیشتر می‌باشد که نشاندهنده برتری طرح پیشنهادی است. لازم بذکر است که در کنار دو مقایسه ذکر شده، افزایش امنیت طرح نیز مطرح می‌باشد که پیش از این مطرح و مورد بررسی قرار گرفته است.

با عنایت به جداول ۱، ۲ و ۳ نکته قابل توجه دیگر این است که با استفاده از تکنیک بازپیکربندی‌جزئی، صرفه‌جویی قابل ملاحظه‌ای نسبت به پیکربندی متداول بدست می‌آید و این موضوع موجب انتخاب تراشه‌های ارزان‌تر و به همان نسبت صرفه‌جویی اقتصادی می‌گردد.

نتیجه گیری

در این مقاله نکاتی که مطرح شد را از دو جهت می‌توان نتیجه‌گیری نمود. در ابتدا برای افزایش امنیت ایده استفاده از چهار الگوریتم به جای یک الگوریتم و کاهش خروجی هر الگوریتم مطرح شد. این الگوریتم‌ها شامل الگوریتم AES، Serpent، Klein و Simon هستند. به دلیل پیاده‌سازی بیشتر از یک الگوریتم در این طرح، سربار سخت‌افزاری زیادی به طرح تحمیل می‌شود. در این مرحله برای حل این مشکل، استفاده از ایده بازپیکربندی جزئی برای این منظور پیشنهاد گردید. با استفاده از این ایده به هر تعداد الگوریتم که به طرحمان اضافه کنیم، تنها به میزان بزرگترین واحد بازپیکربندی منابع سخت‌افزاری استفاده می‌گردد و صرفه‌جویی خوبی در منابع سخت‌افزاری انجام می‌گیرد که این امر سربار سخت افزاری به وجود آمده در پیاده سازی متداول را از بین می برد. افزایش امنیت لینک های ارتباطی رادیویی بیسیم که مصرف توان و ابعاد مدارات در آنها اهمیت زیادی دارد، از مهمترین کاربردهای این طرح می باشد.

مراجع

[1] B. Koziel, R. Azarderakhsh, M. M. Kermani, D. Jao, “Post-quantum cryptography on FPGA based on isogenies on elliptic curves”, IEEE Trans. Circuits Syst., vol. 64-I, no. 1, pp. 86-99, Jan. 2017.

[2] Hoang Anh Du Nguyen, Lei Xie, Mottaqiallah Taouil, Razvan Nane, Said Hamdioui, and Koen Bertels, “On the Implementation of Computation-in-Memory Parallel Adder”, IEEE Trans. VLSI, vol. 25, Issue. 8, pp. 2206-2219, Aug. 2017.

[3] K. Gulati, S. P. Khatri, “Hardware Acceleration of EDA Algorithms, Custom ICs, FPGAs and GPUs”, Springer, e-ISBN 978-1-4419-0944-2, 2010.

[4] L. H. Crockett, R. A. Elliot, M. A. Enderwitz and R. W. Stewart, “The Zynq Book: Embedded Processing with the ARM Cortex-A9 on the Xilinx Zynq-7000 All Programmable SoC”, 1st Edition, Strathclyde Academic Media, 2014.

[5] Xilinx Inc., “Vivado Design Suite User Guide, Partial Reconfiguration”, UG909 (v2017.1), Apr. 2017.

[6] Raikovich, T., Feher, B., “Application of partial reconfiguration of FPGAs in image processing,” Conference on Ph.D. Research in Microelectronics and Electronics, pp.1-4, July 2010.

[7] S. Wankhade, R. Mahajan, “Dynamic Partial Reconfiguration Implementation of AES Algorithm”, International Journal of Computer Applications, Vol. 97, No. 3, pp.15-18, July 2014.

[8] T. Rodrigues, M. Vestias, “Using Dynamic Reconfiguration to Reduce the Area of a JPEG Decoder on FPGA”, Euromicro Conference on Digital System Design, 2015.

[9] Y. Minal, M. A. Sayyad, “Implementation of AES on FPGA”, IOSR Journal of VLSI and Signal Processing, Vol. 4, Issue 5, PP.65-69, Oct. 2014.

[10] Ukrit Arom-oon, “An AES Cryptosystem for Small Scale Network”, 3rd Asian Conference on Defence Technology, Jan. 2017.

[11] National Institute of Standards and Technology, Advanced Encryption Standard (AES), Federal Information Processing Standards Publications – FIPS 197

[12] Z. Gong, S. Nikova, Y. W. Law, “KLEIN: A New Family of Lightweight Block Ciphers”, International Workshop on Radio Frequency Identification: Security and Privacy Issues, June 2011.

[13] R. Anderson, E. Biham, L. Knudsen, “Serpent: A Proposal for the Advanced Encryption Standard”, 2001, http://www.cl.cam.ac.uk/rja14/serpent.html.

[14] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, L. Wingers, “The simon and speck of lightweight block ciphers”, National Security Agency 9800 Savage Road, Fort Meade, MD 20755, USA, June 2013.