

بهبود عملکرد محاسباتی الگوریتم HEVC به کمک سیستم اعداد مانده‌ای

زهرا جوادی مجلج<sup>۱</sup>، حمیدرضا احمدی فر<sup>۲</sup>، سیدمحمدحسین شکران<sup>۳</sup>

۱- زهرا جوادی مجلج، دانشگاه گیلان، دانشکده فنی، گروه مهندسی کامپیوتر

۲- حمیدرضا احمدی فر، دانشگاه گیلان، دانشکده فنی، گروه مهندسی کامپیوتر

۳- سیدمحمدحسین شکران، دانشگاه گیلان، دانشکده فنی، گروه مهندسی کامپیوتر

### چکیده

یکی از استانداردهای پرکاربرد ویدئو، HEVC/H.265 است که با هدف بهبود کارایی فشرده‌سازی در مقایسه با استاندارد قبلی یعنی H.264 ارائه شده است. در حال حاضر این استاندارد در بیش از ۴۷ درصد از ابزارهای فشرده‌سازی ویدئو مورد استفاده قرار گرفته و این میزان در حال گسترش است. فرآیند فشرده‌سازی در این استاندارد از بخش‌های مختلفی مانند تقریب تصویر داخلی (Intra-picture estimation)، پیشبینی تصویر داخلی (Intra-picture prediction)، مهار حرکت (Motion compensation) و تقریب حرکت (Motion estimation) تشکیل شده است. نتایج بررسی‌ها نشان می‌دهد که بین ۶۰ تا ۸۰ درصد از زمان محاسباتی این الگوریتم در بخش تقریب حرکت صرف می‌شود. در بخش تقریب حرکت، دو دسته الگوریتم جستجو قابل استفاده‌اند که عبارتند از الگوریتم‌های جستجوی کامل و الگوریتم‌های جستجوی سریع. برای پیاده‌سازی سخت‌افزاری HEVC موانع زیادی از قبیل پیچیدگی بخش‌های مربوط به فشرده‌سازی وجود دارند. الگوریتم جستجوی سریع (Test-Zone) نمونه‌ای از الگوریتم‌های ساده‌سازی جستجو با هدف کاهش پیچیدگی آن است. یکی از روش‌های بهبود عملکرد محاسباتی استفاده از سیستم اعداد مانده‌ای (RNS) است که قابلیت اجرای موازی عملیات جمع، تفریق و ضرب را دارد. در تحقیقات قبلی با هدف بهبود عملکرد محاسباتی از RNS در پیاده‌سازی بخشی از استاندارد HEVC استفاده شده است. در این مقاله، با هدف بهبود عملکرد محاسباتی الگوریتم HEVC از کمپرسورهای ۴ به ۲، به جای ۳ به ۲، برای کاهش عمق بخشی از مدار انجام عملیات جمع بلوکی داده‌ها استفاده شده است. مدار پیشنهادی پس از شبیه‌سازی توسط ابزار ModelSim توسط Synopsis Design Vision سنتز شد. نتایج سنتز نشان می‌دهد که مدار پیشنهادی در مقایسه با روش قبل بطور میانگین ۱۴/۱ درصد بهبود در تأخیر محاسبات داشته است. به همین ترتیب در توان مصرفی و مساحت تراشه نیز بطور میانگین ۱۴/۶ و ۱۵ درصد بهبود دارد.

\* Corresponding author: زهرا جوادی مجلج، دانشجوی کارشناسی ارشد دانشگاه گیلان

Email: Zahra.javadi7064@gmail.com

کلمات کلیدی: استاندارد ویدئو، استاندارد HEVC/H.265، عملکرد محاسباتی، سیستم اعداد مانده‌ای (RNS)

## ۱. مقدمه

با بالا رفتن کیفیت فیلم‌ها و استقبال بالای مردم از تصاویر و فیلم‌های UHD و 4k نیاز به استانداردهای بهتر، افزایش یافته است. کدگذاری ویدئویی با کارایی بالا (HEVC)<sup>1</sup> یک استاندارد جدید برای فشرده‌سازی ویدئو است که می‌تواند عملکرد بهتری نسبت به استانداردهای قبلی مانند AVC/H.264 ارائه دهد. ویدئوی اصلی متشکل از فریم‌های ویدئویی متوالی، توسط یک کدگذار ویدئویی HEVC، کدگذاری یا فشرده می‌شود تا یک جریان بیتی ویدئویی فشرده ایجاد کند. جریان بیت فشرده، ذخیره یا منتقل می‌شود. یک کدگشای ویدئویی جریان بیت را از حالت فشرده خارج می‌کند تا دنباله‌ای از فریم‌های کدگشایی شده ایجاد کند. الگوریتم‌های سرعت‌بخشی به محاسبات حرکتی در ویدئو، عمدتاً به دو بخش الگوریتم‌های جستجوی کامل و الگوریتم‌های جستجوی سریع طبقه‌بندی می‌شود؛ الگوریتم‌های جستجوی کامل همیشه نتایج بهتری نسبت به جستجوی سریع دارد، اما این الگوریتم‌ها بسیار زمان‌بر هستند [2].

در روش‌های الگوریتم جستجوی کامل یا سریع از SAD<sup>2</sup> به عنوان تابع هزینه استفاده می‌شود؛ چراکه هم ساده است و هم محاسبات کمتری دارد. SAD در واقع اختلاف دقیق بین بلوک‌های موجود و بلوک‌های مرجع را در نظر می‌گیرد و سپس جمع این اختلافات دقیق را بدست می‌آورد. ولی اشکالی که در آن وجود دارد این است که باعث افزایش پهنای باند می‌شود.

$$SAD = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |C(i,j) - R(i,j)| \quad (1)$$

جایی که C و R به ترتیب نشان دهنده پیکسل بلوک فعلی و بلوک مرجع هستند. در این تابع دو عمل تفریق و جمع به تعداد زیاد (به اندازه ابعاد بلوک‌ها که M و N هستند) انجام می‌شود. که برای بهبود عملکرد آن می‌توان از روش‌هایی تسریع کمک گرفت [2].

یکی از روش‌های پرکاربرد در تسریع عمل‌های حسابی جمع، تفریق و ضرب، بهره‌گیری از سیستم اعداد مانده‌ای<sup>3</sup> (RNS) است. محاسبات در چند مسیر موازی، متناظر با پیمان‌های RNS، انجام می‌شوند. عملوندهای دودویی، به چند عملوند کوچک‌تر متناظر با کانال‌های محاسباتی تبدیل می‌شوند و انجام عملیات حسابی روی دو عدد بزرگ به همان عملیات با اعدادی به مراتب کوچک‌تر تبدیل می‌شود. پس از انجام عملیات حسابی، نتایج با تبدیل معکوس به سیستم اعداد مانده‌ای برگردانده می‌شود. عمل تبدیل از دودویی به مانده‌ای و عملیات حسابی در کانال‌ها، موازی و سریع انجام می‌شوند. در این روش می‌توان سرعت محاسبات را به نحو چشمگیری افزایش داد. البته تنها مزیت این سیستم، افزایش سرعت نیست، بلکه مزایای دیگری مانند توان مصرفی پایین و کاهش مساحت تراشه را نیز می‌توان نام برد [1].

در این مقاله بهبود عملکرد محاسباتی در بخش SAD از فشرده‌ساز ویدئو HEVC/H.265 انجام شده است و تلاش اصلی بر این است که نسبت به نمونه‌های پیشین، این تاخیر توسط سیستم عددی مانده‌ای کاهش یابد. در واقع از قابلیت بالای RNS برای کاهش تاخیر زمان محاسبات SAD استفاده شده است.

<sup>1</sup> High efficiency video coding

<sup>2</sup> Sum of absolute differences

<sup>3</sup> Residue number system

در بخش دوم استانداردهای ویدئویی و بطور خاص استاندارد HEVC/H.265 شرح داده می‌شود. در بخش سوم نحوه‌ی عملکرد سیستم عددی مانده‌ای توضیح داده خواهد شد. در بخش چهارم مدار پیشنهادی برای انجام عمل SAD بصورت مانده‌ای ارائه شده و در ادامه نتایج شبیه‌سازی و سنتز آن مورد تجزیه و تحلیل قرار می‌گیرد. در انتها خلاصه و نتیجه‌گیری مقاله آورده شده است.

## ۲. ویدئوی دیجیتال و استانداردهای ویدئویی موجود:

ویدئوی دیجیتال نوعی از ذخیره‌سازی است که از سیگنال‌های دیجیتال بجای سیگنال‌های آنالوگ بهره می‌برد [8]. ویدئوی دیجیتال یک نمایشگر الکترونیکی از تصاویر متحرک (ویدئو) است که به صورت داده‌های دیجیتال کدگذاری شده است و شامل مجموعه‌ای از تصاویر دیجیتالی است که به صورت سریع نمایش داده می‌شود. ویدئو آنالوگ، نشان‌دهنده تصاویر متحرک در قالب سیگنال‌های آنالوگ است.

ویدئوی دیجیتال ابتدا در سال ۱۹۸۶ با فرمت سونی D معرفی شد، که یک سیگنال ویدئویی با مولفه‌ی استاندارد غیرفشرده را در فرم دیجیتالی ثبت کرد. امروزه علاوه بر فرمت‌های غیرفشرده، فرمت‌های ویدئویی دیجیتال فشرده محبوب شامل H.264 و MPEG-4<sup>1</sup> وجود دارند. رابط‌های استاندارد مدرن مورد استفاده برای پخش ویدئوی دیجیتال شامل HDMI<sup>2</sup>، DisplayPort، رابط بصری دیجیتال<sup>3</sup> (DVI) و رابط سریال دیجیتال<sup>4</sup> (SDI) است [8].

ویدئوی دیجیتال زنده، پهنای باند بیشتری از فیلم دیجیتال ضبط شده برای ذخیره‌سازی داده‌ها را مصرف می‌کند. مقدار پهنای باند یا ذخیره‌سازی مورد نیاز توسط اندازه قاب، عمق رنگ و نرخ فریم<sup>5</sup> تعیین می‌شود. داده‌های مورد نیاز برای نشان دادن یک فریم داده‌ها با ضرب در تعداد پیکسل‌ها در تصویر تعیین می‌شود. با این ضرب، پهنای باند مورد نیاز برای ذخیره‌سازی یک قاب با نرخ فریم تعیین می‌شود. سپس الزامات ذخیره‌سازی کلی برای یک برنامه می‌تواند با افزایش پهنای باند با طول مدت برنامه تعیین شود [8]. این محاسبات برای ویدئوی غیرفشرده نیز دقیق است، اما با توجه به نرخ بیت<sup>6</sup> نسبتاً بالای ویدئوی غیرفشرده، فشرده‌سازی ویدئو به طور گسترده‌ای مورد استفاده قرار می‌گیرد. در مورد ویدئوی فشرده، هر فریم نیاز به یک درصد کمی از بیت‌های اصلی دارد. البته لازم نیست که تمام فریم‌ها به همان اندازه با همان درصد فشرده شوند. در عوض، عامل متوسط فشرده‌سازی را برای تمام فریم‌های گرفته شده با هم در نظر می‌گیریم.

نرخ بیت نشان می‌دهد در مدت زمانی معین چه مقدار اطلاعات از جایی به جای دیگر منتقل می‌شود، در مورد ویدئو غیرفشرده، نرخ بیت به طور مستقیم به کیفیت ویدئو مربوط است. نرخ بیت یک بخش مهم در هنگام انتقال ویدئو است، زیرا لینک انتقال باید قادر به حمایت از این نرخ بیت باشد. تاخیر زمانی نرخ بیت نیز با ذخیره‌سازی ویدئو مهم است. زیرا، همانطور که در بالا نشان داده شد، اندازه ویدئو متناسب با نرخ بیت و مدت زمان است [9]. نرخ بیت ویدئوی غیرفشرده برای اکثر برنامه‌های کاربردی بسیار زیاد است. فشرده‌سازی ویدئو در واقع به این منظور است تا نرخ بیت را تا حد زیادی کاهش دهد. در ادامه به معرفی یکی از پرکاربردترین استانداردهای فشرده‌سازی ویدئو می‌پردازیم.

<sup>1</sup> Moving picture experts group

<sup>2</sup> Hyper text markup language

<sup>3</sup> Digital visual interface

<sup>4</sup> Spatial data infrastructure

<sup>5</sup> Frame Rate

<sup>6</sup> Bit Rate

## ۱-۲ استاندارد ویدئوی HEVC/H.265

کدگذاری ویدئو با کارایی بالا (HEVC) یک استاندارد جدید برای فشرده‌سازی ویدئو است که می‌تواند عملکرد بهتری نسبت به استانداردهای قبلی مانند AVC/H.264 ارائه دهد. ویدئوی اصلی متشکل از فریم‌های ویدئویی متوالی، توسط یک کدگذار ویدئویی HEVC، فشرده می‌شود تا یک جریان بیتی ویدئویی فشرده ایجاد کند. جریان بیت فشرده ذخیره یا منتقل می‌شود. یک کدگشای ویدئویی جریان بیت را از حالت فشرده خارج می‌کند تا دنباله‌ای از فریم‌های کدگشایی شده ایجاد کند. HEVC دارای ساختار اولیه مشابه استانداردهای قبلی مانند MPEG-2 و H.264/AVC است. اما با این تفاوت که، HEVC شامل بسیاری از بخش‌های کارآمد می‌باشد، مانند:

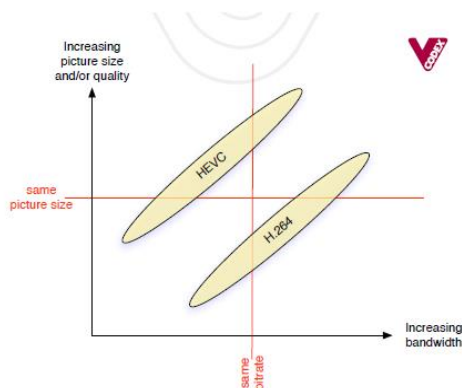
- پارتیشن‌بندی انعطاف‌پذیرتر، از پارتیشن با اندازه‌های بزرگ تا کوچک
- انعطاف‌پذیری بیشتر در حالت‌های پیش‌بینی و تغییر اندازه بلوک
- فیلترهای درون‌یابی و رفع خطای پیچیده‌تر
- پیش‌بینی پیچیده‌تر و سیگنالینگ حالت‌ها و بردارهای حرکت
- ویژگی‌هایی برای پشتیبانی از پردازش موازی کارآمد.

یک دنباله ویدئویی سازگار با HEVC باید (الف) مشخصات فریم ویدئوی فشرده را داشته‌باشد و (ب) با استفاده از روشی که در استاندارد شرح داده شده‌است به طور صحیح قابل کدگشایی باشد. دنباله‌های ویدئویی HEVC را می‌توان در فایل‌های رسانه‌ای ذخیره کرد، از طریق اینترنت پخش کرد و از طریق پخش منتقل کرد [10]. این به عهده طراح کدگذار است که تعدادی از روش‌ها یا ابزارهایی را که ممکن است توسط کدگذار فشرده‌سازی ویدئو استفاده شود را مشخص کند. وضوح بالا در حال حاضر برای بسیاری از دستگاه‌ها و برنامه‌ها امر مهمی است. HEVC برای تامین نیازهای زیر توسعه یافته است:

- استفاده گسترده از ویدئوهای دیجیتالی، با وضوح بالا، که به ظرفیت شبکه فشار زیادی وارد می‌کند.
- افزایش استفاده از وضوح ویدئویی فراتر از HD، که بار شبکه‌ها و ذخیره‌سازی را بیشتر می‌کند.
- بهبود مستمر در ظرفیت پردازش.

در HEVC، امکان ذخیره یا انتقال ویدئو با کارایی بیشتری نسبت به فناوری‌های قبلی مانند H.264 وجود دارد. این یعنی:

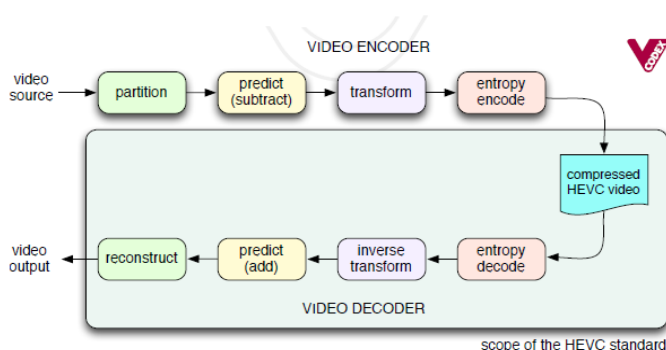
- با همان اندازه و کیفیت تصویر، یک دنباله ویدئویی HEVC باید نسبت به دنباله ویدئویی H.264 ظرفیت ذخیره‌سازی یا انتقال کمتری داشته باشد.
- در همان پهنای باند ذخیره یا انتقال، کیفیت وضوح دنباله ویدئویی HEVC باید از دنباله ویدئویی H.264 مربوطه بیشتر باشد [11].



شکل ۱- دستاوردهای بالقوه HEVC در مقابل AVC [11]

شکل ۱ تفاوت‌های کارایی دو مدل مذکور را نشان می‌دهد. در شکل ۲ مراحل اصلی فشرده‌سازی ویدئو و جایگاه استاندارد HEVC در آن نشان داده شده‌است. مراحل انجام شده توسط کدگذار و کدگشای ویدئو شامل موارد زیر است:

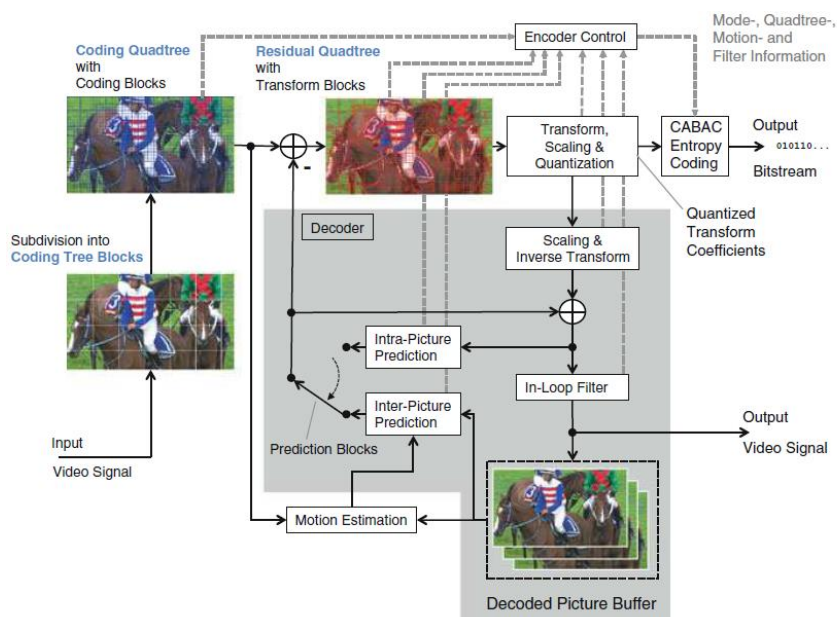
- تقسیم هر تصویر به چندین واحد
- پیش‌بینی هر واحد با استفاده از پیش‌بینی INTER یا INTRA و کسر پیش‌بینی از واحد
- تغییر و کمی کردن باقی‌مانده (تفاوت بین واحد تصویر اصلی و پیش‌بینی)
- کدگذاری آنتروپی خروجی تبدیل، اطلاعات پیش‌بینی، اطلاعات حالت و هدرها.
- عملیات فشرده‌سازی در HEVC
- کدگشایی آنتروپی و استخراج عناصر دنباله گذشته
- تغییر اندازه و وارونه‌شدن مرحله تبدیل
- پیش‌بینی هر واحد و افزودن پیش‌بینی به خروجی تبدیل معکوس
- بازسازی تصویر کدگشایی شده.



شکل ۲- مراحل کلی پردازش ویدئو [12]

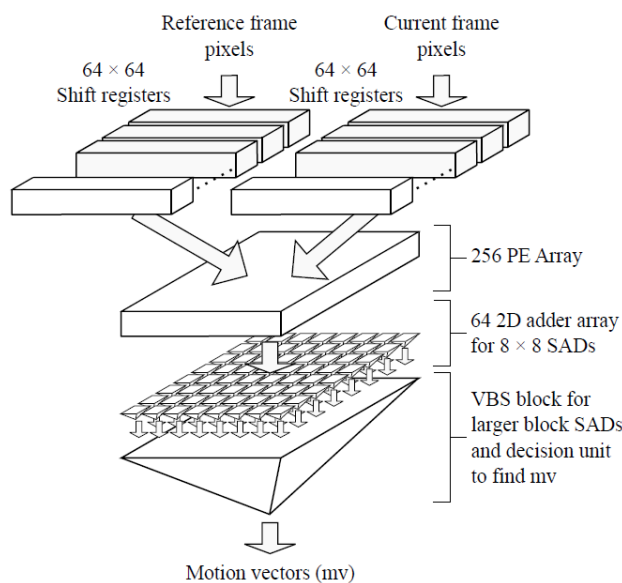
در استاندارد HEVC نیز بطور مجزا از واحدهای عملیاتی مختلفی استفاده می‌شود که شکل ۳ ساختار کلی آن‌ها و نحوه ارتباط بین واحدها را نشان می‌دهد.





شکل ۳- ساختار کلی یک کدگذار ویدئو بر اساس استاندارد HEVC [18]

شرح عملکرد هر یک از بخش‌های نامبرده شده در شکل ۳، بسیار طولانی و خارج از بحث مقاله است. اما روش پیشنهادی در این مقاله مربوط به واحد تقریب حرکت است. شکل ۴ معماری بخش تقریب حرکت را نشان می‌دهد.



شکل ۴- معماری واحد تقریب حرکت در استاندارد HEVC [2]

مطابق با شکل ۴، معماری شامل دو بخش اصلی است که عبارتند از یک آرایه‌ی دو بعدی از عناصر پردازشی ( $PE^2$ ) و ساختاری درخت‌گونه با اندازه‌های متغیر به نام  $VBS^3$  برای محاسبه‌ی SAD. معماری پایه در این استاندارد مشابه با استاندارد قبلی یعنی AVC/H.264 است [2]. البته با این تفاوت که تغییراتی در بخش تقریب حرکت

<sup>1</sup> Motion estimation

<sup>2</sup> Processing Element

<sup>3</sup> Variable Block Size

برای اجرای الگوریتم جستجوی سریع TZ منظور شده است. در معماری از دو آرایه برای ثبات‌های داده استفاده شده که داده‌های مربوط به بلوک مرجع و بلوک فعلی را نگهداری می‌کنند و به آرایه‌ی  $16 \times 16$  از PEها متصل شده‌اند. هر واحد PE یک تابع SAD را برای پیکسل‌های  $4 \times 4$  از بلوک‌های مرجع و فعلی محاسبه می‌کند. به دنبال آن، آرایه‌ای دو بعدی از ۶۴ جمع‌کننده، محاسبات  $8 \times 8$  توابع SAD را انجام می‌دهند و درخت جمع‌کننده VBS حداقل اندازه را از میان مقادیر به‌دست‌آمده از توابع SAD و در واقع بردار پیش‌بینی حرکت را انتخاب می‌کند. ستونی از ۶۴ پیکسل داده را می‌توان به ثبات‌های شیفت به راست در هر دو بلوک داده‌ی مرجع یا فعلی انتقال داد. آرایه‌ی ثبات داده‌ی مرجع قادر به به‌روزرسانی داده‌ها به ستون‌های سمت چپ یا راست را دارد. بخش VBS شامل یک سری از جمع‌کننده‌هایی است که می‌توانند مقدار خروجی توابع SAD را از بلوک‌های کوچک‌تر بگیرند و توابع SAD را برای ابعاد بزرگ‌تر محاسبه کنند. به همین دلیل پهنای خروجی جمع‌کننده هر بار بیشتر می‌شود و به دنبال آن زمان تاخیر انتشار رقم نقلی در جمع‌کننده‌های بزرگ‌تر نیز افزایش می‌یابد. یکی از روش‌های جلوگیری از انتشار رقم نقلی استفاده از سیستم اعداد مانده‌ای است که در ادامه شرح داده خواهد شد [2].

### ۳. سیستم اعداد مانده‌ای RNS :

یک سیستم عددی مانده‌ای توسط مجموعه‌ای از  $k$  عدد صحیح تعریف می‌شود  $\{m_1, m_2, m_3, \dots, m_k\}$  که به هر یک از اعداد پیمانه گفته می‌شود. یک عدد صحیح  $X$  در مجموعه پیمانه‌ای با مجموعه باقیمانده‌های آن نشان داده می‌شود  $\{x_1, x_2, x_3, \dots, x_k\}$ . به این معنا که:

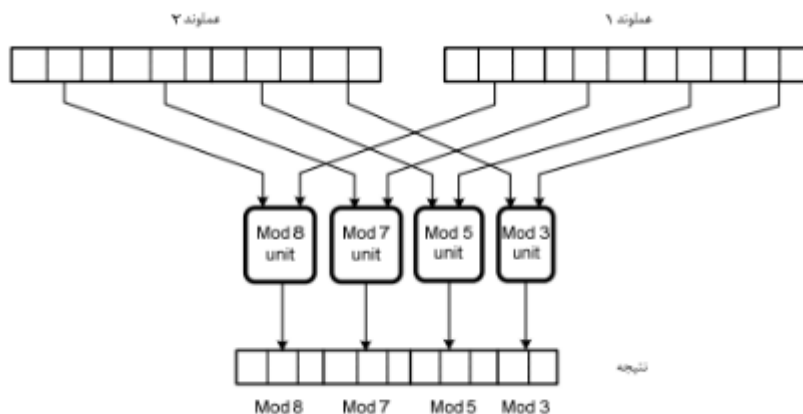
$$x_i = X \bmod m_i = |X|_{m_i} \quad 1 \leq x_i \leq m_i \quad (2)$$

محدوده نمایش سیستم اعداد مانده‌ای که با DR نشان داده می‌شود، بیانگر محدوده اعدادی است که قابل نمایش در سیستم هستند. با فرض اول بودن پیمانه‌ها نسبت به هم، حاصل ضرب آن‌ها  $M = m_1 \times m_2 \times \dots \times m_k$  نشان‌دهنده‌ی  $(DR_{m_k})$  است، یعنی بازه‌ی  $[0, M)$ . یکی از عوامل اصلی در تعیین میزان کارآمدی یک سیستم عددی، نحوه‌ی انجام محاسبات در آن است [1]. در این بخش عملیات حسابی مانده‌ای معرفی خواهد شد که به کمک آن می‌توانیم کارایی نظام را در عملیات حسابی مانده‌ای ارزیابی کنیم. برای عملیات جمع، تفریق و ضرب اعداد در یک سیستم اعداد مانده‌ای، انجام آن‌ها در پیمانه‌ی موجود، کافی است. به بیان دقیق‌تر، اگر  $\{m_1, \dots, m_k\}$  مجموعه پیمانه‌ها باشد، عددهای صحیح  $x$  و  $y$ ، به ترتیب با مجموعه باقی‌مانده‌های  $\{x_1, \dots, x_k\}$  و  $\{y_1, \dots, y_k\}$  نشان داده می‌شوند. عدد صحیح  $z$  که حاصل  $x + y$  است، بصورت  $\{z_1, \dots, z_k\}$  نشان داده شده، به طوری که:

$$z_i = (x_i + y_i) \bmod m_i \quad 1 \leq i \leq k \quad (3)$$

عملیات تفریق و ضرب نیز به طور مشابه انجام می‌شوند. با این حال، انجام کارهایی مانند مقایسه، محاسبه علامت، تشخیص سرریز، مقیاس‌گذاری و تقسیم در سیستم اعداد مانده‌ای دشوار است [1]. نتیجه‌ی مهمی که از رابطه (۳) به‌دست می‌آید؛ موازی بودن عملیات است، یعنی عملیات جمع، تفریق و یا ضرب می‌تواند در هر یک از پیمانه‌ها به‌طور مستقل انجام شود. به‌عنوان مثال، در عمل جمع مانده‌ای، انتشار رقم نقلی فقط داخل هر پیمانه و مستقل از سایر پیمانه‌ها صورت می‌گیرد. نحوه‌ی انجام عملیات در هر پیمانه مشابه عملیات حسابی معمولی است. شکل ۵ ساختار عمومی عملیات مانده‌ای را برای مجموعه‌ی چهار پیمانه‌ای  $\{3, 5, 7, 8\}$  نشان می‌دهد. مطابق شکل، تعداد بیت‌های لازم برای نمایش باقیمانده‌ها در پیمانه‌ی ۳، دو بیت و در سایر پیمانه‌ها سه بیت است. البته ذکر این نکته ضروری است که نتیجه‌ی به‌دست آمده باید در همان پیمانه نمایش داده شود (یعنی

ممکن است حاصل بزرگتر از پیمانانه باشد). در ادامه به بررسی نحوه انجام هر یک از عمل‌های حسابی می‌پردازیم [1].



شکل ۵- ساختار عمومی عملیات مانده‌ای در مجموعه چهار پیمانانه‌ای {3, 5, 7, 8} [15]

### ۳-۱ تبدیل از دودویی به مانده‌ای و برعکس:

اولین مرحله از استفاده‌ی سیستم اعداد مانده‌ای در یک کاربرد، تبدیل نمایش دودویی به این سیستم است که در اصطلاح تبدیل مستقیم نامیده می‌شود. پس از انجام محاسبات مانده‌ای، باید عمل تبدیل از مانده‌ای به دودویی و یا تبدیل معکوس صورت گیرد. همانند عملیات حسابی مانده‌ای که در بخش قبل به اختصار شرح داده شد، پیچیدگی تبدیل مستقیم و معکوس بستگی به نوع پیمانانه‌های انتخاب شده دارد، اگر پیمانانه‌ی انتخاب شده توانی از دو ( $2^n$ ) باشد در تبدیل مستقیم تنها کافی است که همه‌ی بیت‌هایی را که ارزش آنها بیشتر یا مساوی  $n$  هستند، حذف کنیم. وابستگی مدار تبدیل معکوس معمولاً بسیار بیشتر از تبدیل مستقیم است [1].

### ۳-۲ تبدیل از مانده‌ای به دودویی (تبدیل معکوس):

مسئله‌ی پیدا کردن معادل دودویی یک عدد مانده‌ای، به قرن‌ها پیش بازمی‌گردد. امروزه با به‌کارگیری سیستم اعداد مانده‌ای در برخی کاربردها، روش‌های حل این مسئله مورد بررسی دقیق‌تری قرار گرفته است. به‌طور کلی سه روش استفاده از قضیه‌ی باقیمانده‌های چینی<sup>۱</sup>، مبنای ترکیبی (MRC) و قضیه باقیمانده‌های چینی جدید<sup>۲</sup> در تبدیل معکوس به‌کار می‌روند. پایه و اساس همه‌ی روش‌ها، مبتنی بر یافتن معکوس ضربی است. معکوس ضربی فقط و فقط زمانی قابل تعریف است که شرط اول بودن  $x$  و  $m$  ( $\gcd(m, x) = 1$ ) برقرار باشد [13].

### ۳-۳ محاسبات مانده‌ای:

در محاسبات پیمانانه‌ای بسته به این‌که از چه پیمانانه‌هایی استفاده شود، ساختار انجام عملیات تغییر می‌کند. در حالت کلی می‌توان پیمانانه‌ها را به سه فرم در نظر گرفت که عبارتند از  $2^n$ ،  $2^n - \delta$  و  $2^n + \delta$ . محاسبات پیمانانه  $2^n$  تفاوتی با محاسبات  $n$  بیتی با حذف رقم نقلی ندارد. در دو پیمانانه‌ی دیگر نیز اگر  $\delta = 1$  باشد، معمولاً

<sup>1</sup> Chinese remainder theorem

<sup>2</sup> New CRT



محاسباتی ساده‌تر از سایر مقادیر دارد، به همین دلیل در مجموعه‌های سه پیمانه‌ای، اغلب این حالت گزینه‌ی مناسب‌تری است. در این بخش به معرفی محاسبات پیمانه‌ای در حالت  $\delta = -1$  می‌پردازیم.

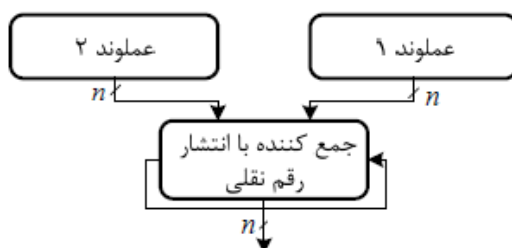
### ۳-۳-۱ جمع و تفریق در پیمانه $2^n - 1$

جمع دو عدد در پیمانه‌ی  $2^n - 1$  را می‌توان مطابق رابطه (۴) نشان داد. طبق رابطه (۵)، زمانی که حاصل عملیات بیشتر از پیمانه باشد یا در واقع رقم نقلی در خروجی حاصل شود، کافی است که مقدار ۱ به آن اضافه گردد.

$$|A + B|_{2^n - 1} = \begin{cases} A + B. & A + B < 2^n - 1 \\ A + B + 1. & A + B \geq 2^n - 1 \end{cases} \quad (4)$$

$$|2^n c_n|_{(2^n - 1)} = |(2^n - 1 + 1)c_n|_{(2^n - 1)} = c_n \quad (5)$$

بدین ترتیب هر رقم نقلی با ارزش  $2^n$  را باید به بیت کم‌ارزش‌تر در حاصل، اضافه کرد، شکل ۶ ساختار کلی عملیات جمع در این پیمانه را نشان می‌دهد [16].

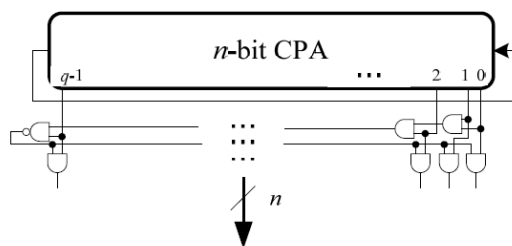


شکل ۶- حاصل به پیمانه  $2^n - 1$  [16]

در عمل تفریق مطابق با رابطه (۶) کافی است که مقدار مکمل یک، از عملوند دوم را با عملوند اول جمع کرد.

$$|A - B|_{2^n - 1} = |A + \bar{B} - 2^n + 1|_{2^n - 1} = |A + \bar{B}|_{2^n - 1} \quad (6)$$

یکی از معایب محاسبات در این پیمانه مشابه با عمل تفریق مکمل یک، وجود دو نمایش برای عدد صفر است که برای تشخیص آن از مدار شکل ۷ کمک گرفته شده است. مطابق شکل در صورتی که مقدار خروجی مدار جمع-کننده، عدد  $2^n - 1$  باشد، به کمک رشته‌ای از گیت‌های منطقی AND، آن را به صفر تبدیل می‌کنیم [17].



شکل ۷- مدار تصحیح‌کننده‌ی حاصل جمع در پیمانه‌ی  $2^n - 1$  [17]

#### ۴. طرح پیشنهادی:

در [2] از مجموعه سه پیمانه‌ای  $\{m_1, m_2, m_3\} = \{2^n - 1, 2^n, 2^{n+1} - 1\}$  برای استفاده در عملیات جمع پیمانه‌ای استفاده شده‌است. چراکه این پیمانه‌ها کمترین تاخیر را در محاسبات جمع دارند و علاوه بر این از مدار تبدیل معکوس مناسبی برخوردارند. تبدیل مستقیم در پیمانه‌های بالا کار سختی نیست و با چند جمع‌کننده پیمانه‌ای انجام می‌شود.

محدوده‌ی نمایش DR برای RNS، در مجموعه سه پیمانه‌ای  $\{2^n - 1, 2^n, 2^{n+1} - 1\}$ ،  $3n + 1$  بیت دارد. از طرفی در معماری برآورد حرکت، محاسبات پیمانه‌ای بعد از توابع  $8 \times 8$  SAD قرار می‌گیرد که فقط  $(n + 6)$  بیت پهنا دارد. بنابراین محدوده‌ی نمایش در این مجموعه برای نمایش ورودی  $X$  کافی است (یعنی  $3n > n + 6$ ). در اولین گام برای محاسبات مانده‌ای لازم است که عملیات تبدیل مستقیم به مانده‌ای انجام شود. با فرض نمایش عدد دودویی  $X$  مطابق با رابطه (۷) می‌توان نمایش مانده‌های  $x_1$ ،  $x_2$  و  $x_3$  را در این مجموعه طبق روابط (10)، (11) و (12) به دست آورد.

$$X = b_{3n-1}b_{3n-2} \dots b_{2n-1}b_{2n-2} \dots b_{n-1}b_{n-2} \dots b_1b_0 \quad (7)$$

عدد باینری  $X$  را به ۳ بلوک  $n$  بیتی تقسیم می‌کنیم.

$$B_{1,n} \triangleq \sum_{j=2n}^{3n-1} b_j 2^{j-2n}, B_{2,n} \triangleq \sum_{j=n}^{2n-1} b_j 2^{j-n}, B_{3,n} \triangleq \sum_{j=0}^{n-1} b_j 2^j \quad (8)$$

باقی‌مانده‌ی بدست آمده در پیمانه‌ی  $2^n - 1$  به صورت زیر است:

$$x_1 = |X|_{2^n-1} = |B_{1,n} \cdot 2^{2n} + B_{2,n} \cdot 2^n + B_{3,n}|_{2^n-1} \quad (9)$$

از نتایج رابطه‌ی (۵) و رابطه‌ی  $|X \pm Y|_m = ||X|_m \pm |Y|_m|_m$  که از ویژگی‌های RNS است، می‌توان باقیمانده‌ها را به صورت زیر به دست آورد:

$$x_1 = |B_{1,n} \cdot 1 + B_{2,n} \cdot 1 + B_{3,n}|_{2^n-1} \quad (10)$$

باقی‌مانده‌ی پیمانه‌ی  $2^{n+1} - 1$  دقیقاً از این راه بدست می‌آید با این تفاوت که، پهنای آن یک واحد اضافه می‌شود یعنی  $n + 1$  بیت:

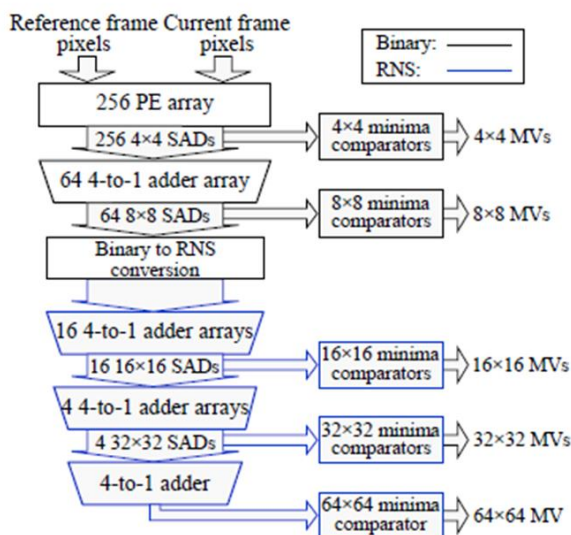
$$x_3 = |B_{1,n+1} + B_{2,n+1} + B_{3,n+1}|_{2^{n+1}-1} \quad (11)$$

باقی‌مانده‌ی پیمانه‌ی  $2^n$  هم به صورت زیر است:

$$x_2 = B_{3,n} \quad (12)$$

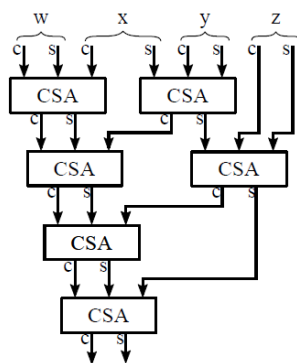
همانگونه که در بخش ۲ ذکر شد، هدف مقاله به کارگیری محاسبات مانده‌ای در پیاده‌سازی سخت‌افزاری توابع SAD است که در درخت VBS قرار دارد. ساختار دقیق VBS در شکل ۸ آمده‌است [۲]. مطابق با شکل، در درخت VBS از محاسبه‌ی توابع  $8 \times 8$  SAD، از محاسبات پیمانه‌ای استفاده می‌شود که در شکل به رنگ آبی نشان داده شده‌است. در این بخش نیاز به مداری داریم که ۸ بردار ۸ بیتی را با هم جمع نماید و حاصل را به مرحله‌ی بعد انتقال دهد. در [2] از مدار شکل ۹ برای کاهش عمق ورودی‌های ۸ بیتی به دو و سپس انجام عمل

جمع، استفاده شده است. بردارهای  $w, x, y$  و  $z$  هر کدام ۸ بیتی هستند که در شکل، مدار برای یک بیت نشان داده شده است. در این شکل، کاهش عمق به کمک مدارهای  $CSA^1$  انجام می‌شود که می‌تواند به طور مشابه بصورت مانده‌ای پیاده‌سازی شود. البته لازم است که رقم‌های نقلی خروجی به‌صورت بازگشتی<sup>۲</sup> (EAC) به بیت کم ارزش‌تر در مرحله‌ی بعد، اضافه شود (طبق رابطه (5)).



شکل ۸- ساختار VBS که با کمک RNS پیاده‌سازی شده [2].

در RNS عملیات جمع در سه کانال (مدار) موازی انجام می‌شود. داده‌ها از آرایه وارد جمع‌کننده شده و بر اساس پهنای داده ( $n = 8$ ) که در پیمان‌های ۲۵۵ و ۲۵۶، ۸ بیت و در پیمان‌های ۵۱۱ و ۵۱۲، ۹ بیت است، جمع می‌شوند. از آنجایی که محاسبات پیمان‌های هستند، لذا در دو پیمان‌های ۲۵۵ و ۵۱۱ از مدارهای CSA پیمان‌های با قابلیت EAC استفاده شده است. طولانی‌تری مسیر تاخیر مدار شامل چهار CSA و یا همان FA (Full Adder) است که هر FA به اندازه‌ی دو گیت XOR، تاخیر دارد [2].

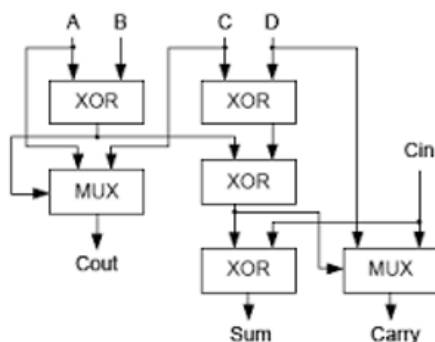


شکل ۹- معماری کاهش عمق داده‌ها از ۸ به ۲ در [2]

<sup>1</sup> Carry Save Adder

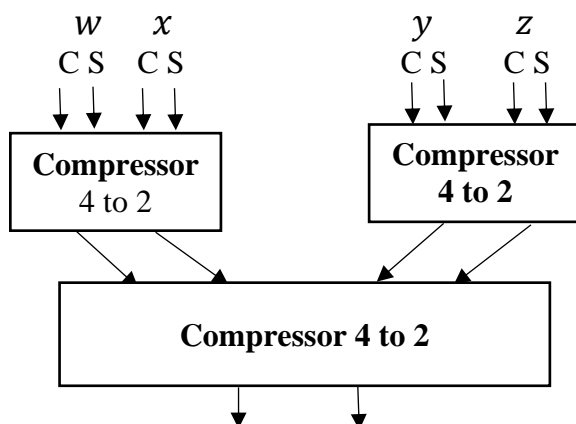
<sup>2</sup> End Around Carry

در عملیات کاهش عمق ورودی‌ها به دو می‌توان از مدارهای سریع‌تری نیز استفاده کرد. در واقع همه‌ی مدارهایی که در کاهش عمق استفاده می‌شوند، کمپرسور<sup>۱</sup> نام دارند و CSA را می‌توان یک کمپرسور ۳ به ۲ نامید. در روش پیشنهادی در این مقاله از کمپرسور ۴ به ۲ استفاده می‌شود که از عملکرد بهتری در مقایسه با CSA برخوردار است. شکل ۱۰ مدار یک کمپرسور ۴ به ۲ را نشان می‌دهد که به کمک گیت‌های XOR و MUX طراحی شده است.



شکل ۱۰- معماری کمپرسور ۴ به ۲ [14]

در فرآیند جمع ۸ ورودی ۸ بیتی نیاز به سه کمپرسور ۴ به ۲ برای هر بیت داریم که مطابق شکل ۱۱ عملیات کاهش عمق به ۲ را انجام می‌دهند. با این کار طولانی‌ترین مسیر تاخیر از دو کمپرسور تشکیل شده که تاخیر آن معادل ۶ گیت XOR است در حالی که در روش قبلی (کمپرسورهای ۳ به ۲، CSA) از ۸ گیت XOR استفاده می‌شود. بدین ترتیب سرعت انجام عملیات کاهش عمق به دو در کنار مساحت و توان مصرفی مدار کاهش می‌یابد که در بخش بعدی به کمک شبیه‌سازی و سنتز، دو روش با هم مقایسه شده‌اند.



شکل ۱۱- مدار کاهش عمق ۸ به ۲ در روش پیشنهادی به کمک کمپرسورهای ۴ به ۲

<sup>1</sup> Compressor

#### ۴-۱ شبیه‌سازی و سنتز مدار پیشنهادی:

طرح پیشنهادی به زبان توصیف سخت‌افزار VHDL در برنامه ModelSim نوشته و شبیه‌سازی شده است. بعد از بررسی صحت عملکرد مدار، به کمک کامپایلر Synopsys Design Vision در ماشین مجازی CentOS و با تکنولوژی 90nm سنتز شد. در جداول ۱ تا ۳ میزان تاخیر، توان مصرفی و مساحت مدار به ترتیب در سه کانال ۲۵۵، ۵۱۱ و ۲۵۶ برای روش پیشنهادی (یعنی استفاده از کمپرسور ۴ به ۲) و روش ارائه شده در [۲] نشان داده شده‌اند.

جدول ۱- نتایج سنتز روش پیشنهادی و [2] در کانال ۲۵۵

پیمانه‌ی $2^n - 1$	[2]	روش پیشنهادی
Area( $\mu m^2$ )	۳۱.۷	۲۸.۲
Power(mW)	۳.۱۰	۲.۷۸
Delay(ns)	۰.۰۷	۰.۰۶

جدول ۲- نتایج سنتز روش پیشنهادی و [2] در کانال ۵۱۱

پیمانه‌ی $2^{n+1} - 1$	[2]	روش پیشنهادی
Area( $\mu m^2$ )	۳۵.۲	۳۱.۷
Power(m W)	۳.۴۴	۳.۱۱
Delay(ns)	۰.۰۷۶	۰.۰۶۵

جدول ۳- نتایج سنتز روش پیشنهادی و [2] در کانال ۲۵۶

پیمانه‌ی $2^n$	[2]	روش پیشنهادی
Area( $\mu m^2$ )	۲۸.۲	۲۱.۲
Power(mW)	۲.۷۸	۲.۰۹
Delay(ns)	۰.۰۷	۰.۰۶

نتایج در جداول ۱ تا ۳ نشان‌دهنده‌ی کاهش ۱۴، ۱۴/۴ و ۱۴ درصدی در تاخیر ایجاد شده به ترتیب در سه کانال ۲۵۵، ۵۱۱ و ۲۵۶ است. در توان مصرفی نیز شاهد کاهش ۱۰/۵، ۹/۵ و ۲۴ درصدی در کانال‌های ۲۵۵، ۵۱۱ و ۲۵۶ هستیم و مساحت تراشه نیز به ترتیب ۱۱، ۹ و ۲۵ درصد کاهش داشته است.

#### ۵. نتیجه‌گیری

استاندارد HEVC/H.265 یکی از روش‌های پرکاربرد در پردازش ویدئو است که توانایی فشرده‌سازی بهتری را در مقایسه با استانداردهای قبلی مانند H.264 ارائه داده است. این استاندارد از مراحل مختلفی تشکیل شده و یکی از بخش‌های آن، واحد تقریب حرکت است. در پیاده‌سازی این بخش از الگوریتم‌های جستجوی مختلفی استفاده می‌شود که به دو دسته الگوریتم‌های جستجوی کامل و سریع تقسیم می‌شوند. الگوریتم جستجوی TZ



نمونه‌ای از دسته‌ی دوم است. یکی از مهم‌ترین توابعی که در چنین الگوریتم‌هایی محاسبه می‌شود، تابع SAD است که به کمک آن بردار حرکت بعدی در بخش تقریب حرکت انتخاب می‌شود. برای محاسبه SAD نیاز به انجام تعداد زیادی محاسبات جمع و تفریق است که باعث زمان‌بر شدن آن می‌شود. یکی از روش‌هایی که امکان اجرای موازی عملیات جمع و تفریق را فراهم می‌کند، سیستم اعداد مانده‌ای است. در روش ارائه شده در [2] از مجموعه سه پیمانهای  $\{2^n - 1, 2^n, 2^{n+1} - 1\}$  برای موازی‌سازی محاسبات استفاده شد. بخشی از این عملیات موازی نیاز به جمع بردارهای ۸ بیتی با تعداد زیاد دارد که در بخش VBS، برای محاسبه بردارهای ۸ تایی از RNS استفاده شده است. در عملیات کاهش عمق ۸ بردار ۸ بیتی، از مدارهای کمپرسور ۳ به ۲ کمک گرفته شده است. در این مقاله به کمک کمپرسورهای ۴ به ۲ نشان داده شد که می‌توان با هزینه‌ی کمتری همان عملیات را انجام داد. در پیاده‌سازی کمپرسور ۴ به ۲ از سریع‌ترین مدل‌های موجود استفاده شد که هر کمپرسور با تاخیر ۳ گیت XOR می‌تواند خروجی را تولید نماید. در مجموع برای کاهش عمق ۸ به ۲ از دو سری کمپرسور ۴ به ۲ استفاده شده که با تاخیر ۶ گیت XOR خروجی را تولید می‌نمایند. مدل پیشنهادی به کمک ابزارهای ModelSim و Synopsys Design Vision شبیه‌سازی و سنتز شد. در فرآیند سنتز از کتابخانه 90nm استفاده شد. نتایج سنتز نشان می‌دهد که بطور میانگین در تاخیر، توان مصرفی و مساحت مدار به ترتیب ۱۴/۱، ۱۴/۶ و ۱۵ درصد در مقایسه با روش موجود بهبود حاصل شده است.

## مراجع:

- [1] H. Ahmadifar, "Balanced Moduli Selection for Efficient Reverse Conversion in Residue Number System," PhD Thesis, *University of Shahid Beheshti*, 2013.
- [2] N.C. Vayalil, M. Paul, and Y. Kong "A Residue Number System Hardware Design of Fast-Search Variable-Motion-Estimation Accelerator for HEVC/H.265," *Transactions on Circuits and Systems for Video Technology IEEE*, 2017.
- [3] M.T. Pourazad, C. Doutre, M. Azimi, and P. Nasiopoulos, "HEVC: The new gold standard for video compression: How does HEVC compare with H.264/AVC?," *IEEE Consumer Electronics Magazine*, vol. 1, no. 3, pp. 36–46, July 2012.
- [4] N. Purnachand, L. Alves, and A. Navarro, "Improvements to TZ search motion estimation algorithm for multiview video coding," in *Systems, Signals and Image Processing (IWSSIP)*, 2012 19th International Conference on, April 2012, pp. 388–391.
- [5] J. Xiong, H. Li, F. Meng, Q. Wu, and K. N. Ngan, "Fast HEVC inter cu decision based on latent sad estimation," *IEEE Transactions on Multimedia*, vol. 17, no. 12, pp. 2147–2159, Dec. 2015.

- [6] Z. Chen, J. Xu, Y. He, and J. Zheng, "Fast integer-Pel and fractional-Pel motion estimation for H.264/AVC," *Journal of Visual Communication and Image Representation*, vol. 17, no. 2, pp. 264–290, 2006.
- [7] JVT of ISO/IEC MPEG, ITU-T VCEG, MVC Software Reference Manual-JMVC 8.2, Mar. 2010.
- [8] C. Rosewarne, K. Sharman, R. Sjöberg, G. Sullivan, "High Efficiency Video Coding (HEVC) Test Model 16 (HM 16) Improved Encoder Description Update 10," *35th Meeting: Geneva*, March 2019.
- [9] M. Yang, N. Bourbakis, "A High Bitrate Information Hiding Algorithm For Digital Video Content Under H.264/AVC Compression," *48th Midwest Symposium on Circuits and Systems*, 2005.
- [10] S. Kwong, J. Lee and K. Chung, "Half-Pixel Correction for MPEG-2/H.264 Transcoding," *Image Analysis and Processing*, 2005.
- [11] H. Wang, S. Kwong, C. Kok, "Efficient Prediction Algorithm of integer DCT coefficients for H.264/AVC optimization," *IEEE Transactions on Circuits and Systems for Video Technology*, 2006.
- [12] G. J. Sullivan, P. N. Topiwala, A. Luthra, "The H.264/AVC Advanced Video Coding Standard: Overview and Introduction to the Fidelity Range Extensions," (PDF). Retrieved July 30, 2004.
- [13] I. Koren, "Computer Arithmetic Algorithms," 2d Edition, A.K. Peters Ltd, 2002
- [14] C. Chang, J. Gu and M. Zhang, "Ultra Low-Voltage Low-Power CMOS 4-2 and 5-2 Compressors for Fast Arithmetic Circuits," *IEEE Transaction On Circuit System*, 2004.
- [15] B. Parhami, "Computer Arithmetic: Algorithms and Hardware Designs," *Oxford University Press, New York*, 2nd ed., 2010.
- [16] A.PV Mohan, "Residue Number Systems: Algorithms and Architectures," *New York, Kluwer Academic, Publishers*, 2002.
- [17] G. Jaberipur and H. Ahmadifar, "A ROM-less Reverse RNS Converter for Moduli Set  $\{2^q \pm 1, 2^q \pm 3\}$ ," *IET Computers & Digital Techniques*, to appear, 2013.
- [18] V. Sze, M. Budagavi and G. J Sullivan, "High Efficiency Video Coding, Algorithms and Architectures," *Springer Cham Heidelberg New York Dordrecht London*, 2014