

مقایسه کارایی پیاده‌سازی صحیح و اعشاری فیلترهای مختلف تبدیل موجک گسسته دوبعدی

عبدالصیر تیباش^۱، اسدا... شاه بهرامی^۲

^۱ کارشناسی ارشد، مهندسی کامپیوتر، دانشگاه گیلان، رشت، BasirTibash@msc.guilan.ac.ir

^۲ دانشیار، مهندسی کامپیوتر، دانشگاه گیلان، رشت، Shahbahrami@guilan.ac.ir

چکیده

تبدیل موجک گسسته یکی از تبدیلات مهم در زمینه‌ی پردازش سیگنال‌های دیجیتال است و در کاربردهای مختلف از جمله فشرده‌سازی داده‌های چندرسانه‌ای مورد استفاده قرار می‌گیرد. برای مثال استانداردهای JPEG۲۰۰۰ و MPEG-۴ بر مبنای این تبدیل عمل می‌کنند. الگوریتم‌های مختلفی برای پیاده‌سازی این تبدیل وجود دارد که رویکرد پیچش و روش پلکانی از این جمله‌اند. این الگوریتم‌ها با فیلترهای مختلف به صورت اعشاری و صحیح در بستر کارهای نرم افزاری و سخت افزاری پیاده‌سازی می‌شوند. هدف این مقاله ارزیابی کارایی فیلترهای مختلف پیاده‌سازی تبدیل دوبعدی موجک به صورت اعشاری و صحیح است. فیلترهایی که انتخاب شده عبارتند از سه فیلتر Daub-۴، CDF-۵/۳، CDF-۹/۷ با رویکرد پیچش و دو فیلتر CDF-۵/۳ و CDF-۹/۷ به روش پلکانی. نتایج پیاده‌سازی این فیلترها بر روی تصاویر مختلف به ازای اندازه‌های متفاوت نشان داده که پیاده‌سازی صحیح نسبت به اعشاری به ازای خطای کمتر از یک، میزان حافظه‌ی مصرفی ۵۰ درصد بهبود، تعداد فقدان‌های کش کاهش و میزان کارایی تا دو برابر افزایش یافته است.

واژه‌های کلیدی

تبدیل موجک گسسته دوبعدی، روش پیچش، روش پلکانی، پیاده‌سازی صحیح، پیاده‌سازی اعشاری.

۱- مقدمه

با توجه به پیشرفت فناوری ریزپردازنده‌ها و وجود ریز معماری Haswell که سال ۲۰۱۳ به جای Ivy Bridge توسط شرکت اینتل ارائه شده، پیاده‌سازی اعشاری بسیار بهبود یافته است [۵].

در این مقاله، ابتدا به دو رویکرد پیچش و روش پلکانی ارائه شده برای تبدیل موجک گسسته پرداخته شده است. سپس، پیاده‌سازی صحیحی برای پنج فیلتر پرکاربرد تبدیل موجک گسسته دوبعدی که عبارت‌اند از سه فیلتر Daub-۴، CDF-۵/۳، CDF-۹/۷ با رویکرد پیچش و دو فیلتر CDF-۵/۳ و CDF-۹/۷ به روش پلکانی ارائه گردیده که با میزان خطای بسیار کم، میزان حافظه‌ی مصرفی را ۵۰ درصد کاهش و در نتیجه تعداد فقدان‌های کش^۱ را نیز بهبود و کارایی تا دو برابر افزایش داده است.

این مقاله در ادامه به صورت زیر ساختار بندی شده است. مفاهیم اولیه مربوط به دو الگوریتم پیاده‌سازی تبدیل موجک گسسته از جمله رویکرد پیچش و روش پلکانی در بخش دوم بیان شده است. در قسمت سوم پیاده‌سازی صحیح پیشنهادی برای سه فیلتر Daub-۴، CDF-۵/۳، CDF-۹/۷ با رویکرد پیچش و دو فیلتر CDF-۵/۳ و CDF-۹/۷ به روش پلکانی مطرح گردیده است. نتایج حاصل از مقایسه پیاده‌سازی اعشاری و صحیح پیشنهادی برای این فیلترها بر

امروزه تبدیل موجک گسسته^۱ در پردازش سیگنال، مخصوصاً در الگوریتم‌های فشرده‌سازی^۲، نشانه‌گذاری^۳ و تشخیص الگو^۴ به صورت گسترده‌ای مورد استفاده قرار می‌گیرد [۱]. همچنین، تبدیل موجک گسسته دوبعدی در استانداردهای فشرده‌سازی JPEG۲۰۰۰ و MPEG-۴ بکار رفته است [۲]. مزیت تبدیل موجک نسبت به تبدیل‌های مرسوم مانند تبدیل فوریه^۵ در حال حاضر به خوبی شناخته شده و در بسیاری از حوزه‌های نرم‌افزار، تبدیل موجک در نمایندگی از ویژگی سیگنال که در هر دو حوزه زمان و فرکانس محلی شده کارآمدتر است؛ بنابراین، پیاده‌سازی‌های صحیحی جهت بهبود کارایی برای فیلترهای مختلف تبدیل موجک گسسته پیشنهاد گردیده است [۳-۴].

پیاده‌سازی صحیحی برای فیلتر CDF-۹/۷ با رویکرد پیچش^۶ و روش پلکانی^۷ جهت تصاویر توان پایین ارائه شده است [۳]. با توجه به استفاده از هشت بیت برای عملیات ضرب صحیح، میزان خطای آن نسبت به پیاده‌سازی اعشاری زیاد است. فیلتر CDF-۵/۳ اعشاری با استفاده از ضرب صحیح و گردش^۸ راست توسط دستورالعمل‌های MMX پیاده‌سازی شده است [۴]. نتایج حاصل از این پیاده‌سازی برای پردازنده پنتیوم ۴ افزایش کارایی بالایی را نشان می‌دهد؛ اما

فیلترهای مختلفی برای پیاده‌سازی تبدیل موجک گسسته از جمله فیلترهای دابچیز^{۱۳} و CDF که دو تا از مهم‌ترین این فیلتر هستند، ارائه گردیده‌اند. همچنین، در استانداردهای فشرده‌سازی JPEG۲۰۰۰ و MPEG-۴ از این فیلترها استفاده شده است.

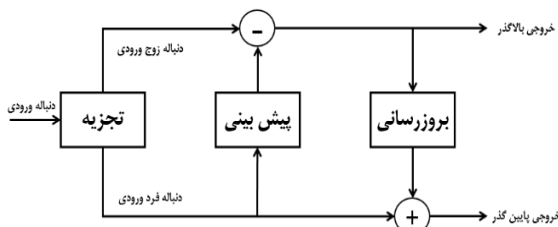
تصویر ورودی برای اعمال تبدیل موجک گسسته دارای مقادیر صحیح برای پیکسل‌هاست و نتیجه حاصل از تبدیل نیز به صورت صحیح ذخیره می‌گردد؛ اما به دلیل اعشاری بودن ضرایب فیلترها عملیات میانی برای تبدیل موجک گسسته به صورت اعشاری انجام می‌گردد. ضرایب پایین گذر و بالا گذر برای فیلترهای Daub-۴، CDF-۵/۳ و CDF-۹/۷ با رویکرد پیچش در جدول ۱ نشان داده شده است [۹-۷].

جدول ۱: ضرایب پایین گذر و بالا گذر برای فیلترها Daub-۴، CDF-۵/۳ و CDF-۹/۷ با رویکرد پیچش.

منبع	نام فیلتر	ضرایب پایین گذر
[۷]	Daub-۴	{-۰.۱۲۹۴۱, ۰.۲۲۴۱۴, ۰.۸۳۶۵۲, ۰.۴۸۲۹۶}
		{-۰.۴۸۲۹۶, ۰.۸۳۶۵۲, -۰.۲۲۴۱۴, -۰.۱۲۹۴۱}
[۸]	CDF-۵/۳	{-۰.۱۲۵, ۰.۲۵, ۰.۷۵, ۰.۲۵, -۰.۱۲۵}
		{-۰.۵, ۱, -۰.۵}
[۹]	CDF-۹/۷	{۰.۰۲۶۷۵, -۰.۰۱۶۸۶, -۰.۰۷۸۲۲, ۰.۲۶۶۸۶, ۰.۰۶۰۲۹۵, ۰.۲۶۶۸۶, -۰.۰۷۸۲۲, -۰.۰۱۶۸۶, ۰.۰۲۶۷۵}
		{۰.۰۹۱۲۷, -۰.۰۵۷۵۵, -۰.۰۵۹۱۲۷, ۱.۱۱۵۰۹, -۰.۵۹۱۲۷, -۰.۰۵۷۵۴, ۰.۰۹۱۲۷}

۲-۲- روش پلکانی

این طرح برای پیاده‌سازی کارای تبدیل موجک گسسته دوبعدی توسط هرلی^{۱۴} و سولدن^{۱۵} به صورت مستقل پیشنهاد گردیده است. روش پلکانی یک الگوریتم بسیار کارا برای محاسبه تبدیل موجک گسسته فراهم می‌کند. ایده اصلی روش پلکانی این است که از ارتباط مقادیر پیکسل‌های تصویر برای حذف افزونگی استفاده می‌کند. روش پلکانی شامل سه قسمت تجزیه^{۱۶}، پیش‌بینی^{۱۷} و بروزسانی^{۱۸} است که در شکل ۲ نشان داده شده است [۱۰].



شکل ۲: روش پلکانی برای پیاده‌سازی کارای تبدیل موجک گسسته [۱۰].

سیگنال ورودی $C_{j,k}$ ابتدا به نمونه‌های زوج و فرد تجزیه می‌شود. سپس ضرایب بالا گذر توسط کم کردن نمونه‌های خروجی تابع پیش‌بینی P (که توسط نمونه‌های فرد تولید شده) از نمونه‌های زوج تولید می‌شود. ضرایب پایین گذر توسط جمع کردن نمونه‌های فرد با خروجی تابع آپدیت U (که توسط خروجی بالا گذر تولید شده) (ت)ید می‌گردد. محاسبه‌ی هر دو خروجی بالا گذر و پایین گذر یک مرحله پلکان نامیده می‌شود. در این روش پیچیدگی محاسباتی متناظر با تبدیل موجک گسسته می‌تواند بیشتر از ۵۰ درصد کاهش

روی تصاویر مختلف با ابعاد در بخش چهارم نشان داده شده است. در نهایت، بخش پنج شامل نتیجه‌گیری است.

۲- مفاهیم اولیه

تبدیل موجک گسسته برخلاف تبدیل فوریه، مجموعه‌های ممکن زیادی از توابع پایه‌ای دارد؛ که مصالحه مختلفی را بین انتخاب توابع پایه‌ای و پیچیدگی پیاده‌سازی‌های سخت‌افزاری مربوطه ایجاد کرده است. روش متفاوتی از جمله روش پیچش و روش پلکانی برای تبدیل موجک گسسته ارائه شده‌اند [۱]. این دو روش به ترتیب در قسمت‌های ۲-۱ و ۲-۲ آورده شده‌اند.

۲-۱- رویکرد پیچش

تبدیل موجک گسسته با استفاده از روشی که به عنوان الگوریتم ملات^{۱۱} شناخته شده، می‌تواند محاسبه شود. در رویکرد پیچش ضرایب موجک رزولوشن‌های پایین از هر مرحله تبدیل موجک گسسته به صورت بازگشتی بر طبق معادلات (۱) و (۲) محاسبه می‌گردد.

$$c_{j-1,k} = \sum_m c_{j,m} \cdot h[m - 2k] \quad (1)$$

$$d_{j-1,k} = \sum_m c_{j,m} \cdot g[m - 2k] \quad (2)$$

که در آن:

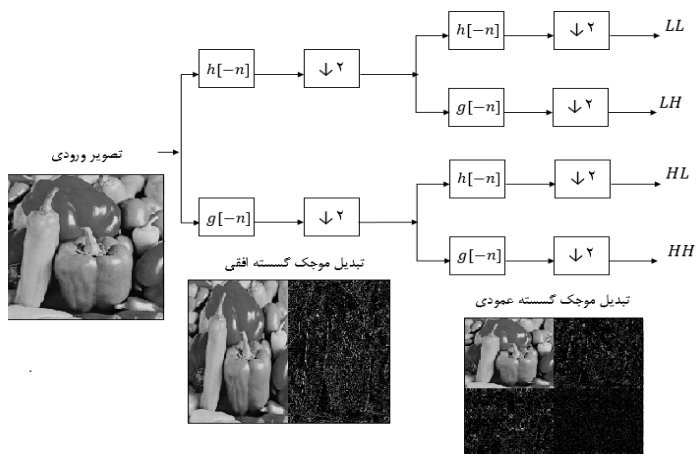
$c_{p,q}$ ضریب پایین گذر q در رزولوشن p است.

$d_{p,q}$ ضریب بالا گذر q در رزولوشن p است.

$h[n]$ فیلتر موجک پایین گذر مربوط به موجک است.

$g[n]$ فیلتر موجک بالا گذر مربوط به موجک است.

ساختار مربوط به الگوریتم تبدیل موجک گسسته دوبعدی در شکل ۱ نشان داده شده که G و H به ترتیب فیلترهای زیر باند پایین گذر^{۱۱} و بالا گذر^{۱۲} را نشان می‌دهند. تصویر ورودی ابتدا به صورت افقی تجزیه شده و سپس نتیجه خروجی به صورت عمودی درون چهار زیر باند که معمولاً توسط LL, LH, HL, HH و LL ترکیب شده قرار می‌گیرد. زیرمجموعه LL که بیشترین اطلاعات را در خود دارد را می‌توان به یک سطح دیگر نیز تجزیه کرد. این فرایند می‌تواند در سطوح دیگر ادامه یافته و در هر سطح فقط بخش LL به عنوان داده مورد استفاده قرار می‌گیرد [۶].



شکل ۱: ساختار مربوط به تبدیل موجک گسسته دوبعدی [۶].

جدول ۴: ضرایب صحیح پیشنهادی برای مراحل پلکان جهت ۵/۳-CDF و ۹/۷-CDF به روش پلکانی.

نام فیلتر	ضریب	مقدار	
CDF ۵/۳	P ₁	-2×2^{-2}	
	U ₁	1×2^{-2}	
	مقیاس	دنباله زوج	23170×2^{-14}
دنباله فرد		23171×2^{-15}	
CDF ۹/۷	P ₁	-25987×2^{-14}	
	U ₁	-862×2^{-14}	
	P ₂	28931×2^{-15}	
	U ₂	14523×2^{-15}	
	مقیاس	دنباله زوج	18835×2^{-14}
		دنباله فرد	28504×2^{-15}

جدول ۴: مشخصات محیط پیاده‌سازی.

واحد پردازنده	Intel® Core™ i۷-۴۵۰۰U CPU @ ۱.۸۰ GHz
کش L ₁	۱۲۸ KB
کش L ₂	۵۱۲ KB
کش L ₃	۴ MB
حافظه اصلی	۶ GB
سیستم‌عامل	Linux Mint xfce ۱۷.۳ x64

۲-۴- نتایج ارزیابی کارایی پیاده‌سازی اعشاری و صحیح

پردازنده Intel® Core™ i۷-۴۵۰۰U CPU @ ۱.۸۰ GHz یک پردازنده با ریز معماری Haswell است که محاسبات اعشاری را با استفاده از ثبات‌های MMX و دستورالعمل‌های FMA^۳ و محاسبات صحیح را با استفاده از دستورالعمل‌های عادی و ثبات‌های عمومی انجام می‌دهد. در نتیجه، پیاده‌سازی اعشاری در این پردازنده بسیار کارآمد است [۵]. نتایج پیاده‌سازی ۵/۳-CDF با رویکرد پیچش به صورت اعشاری و صحیح که از ۳۲ بیت برای انجام عملیات میانی استفاده می‌کنند در جدول ۶ نشان داده شده است. به همین دلیل در ادامه این مقاله، این پیاده‌سازی را پیاده‌سازی صحیح ۳۲ بیتی می‌نامیم. نتایج حاصل از کارایی^{۲۱} صحیح ۳۲ بیتی نسبت به اعشاری و نسبت فقدان کش^{۲۲} اعشاری نسبت به صحیح ۳۲ بیتی بر روی شش تصویر معرفی در شکل ۳ با اندازه‌های متفاوت عملکرد بهتر پیاده‌سازی اعشاری نسبت به صحیح ۳۲ بیتی را بر روی این نوع پردازنده نشان می‌دهد.

جدول ۶: میانگین کارایی پیاده‌سازی صحیح ۳۲ بیتی نسبت به اعشاری و نسبت فقدان کش اعشاری نسبت به صحیح ۳۲ بیتی برای فیلتر ۵/۳-CDF با رویکرد پیچش بر روی ۶ تصویر نشان داده شده در شکل ۳ با ابعاد مختلف.

اندازه‌های تصویر	۲۵۶	۵۱۲	۱۰۲۴	۲۰۴۸	۴۰۹۶
کارایی	۰.۹۹۶	۱.۰۲۱	۰.۹۸۹	۰.۹۶۳	۰.۹۷۳
نسبت فقدان کش	۱.۰۳۲	۰.۸۸۸	۰.۸۲۳	۰.۹۸۸	۰.۹۹۲

برای بهبود کارایی پیاده‌سازی تبدیل موجک گسسته دوبعدی، با توجه به ضرایب صحیح پیشنهادی در جداول ۳ و ۴ عملیات میانی را می‌توان با استفاده

یابد [۱۰]. ضرایب پایین گذر و بالا گذر برای فیلترهای ۵/۳-CDF و ۹/۷-CDF به روش پلکانی در جدول ۲ نشان داده شده است [۱۱-۱۲].

جدول ۲: ضرایب مراحل پلکان برای ۵/۳-CDF و ۹/۷-CDF

منبع	نام فیلتر	ضریب	مقدار
[۱۱]	CDF-۵/۳	P ₁	-۰.۵
		U ₁	۰.۲۵
	مقیاس ^{۱۹}	دنباله زوج	۱.۴۱۴۲۱
		دنباله فرد	۰.۷۰۷۱۱
[۱۲]	CDF-۹/۷	P ₁	-۱.۵۸۶۱۳
		U ₁	-۰.۰۵۲۹۸
	مقیاس	P ₂	۰.۸۸۲۹۱
		U ₂	۰.۴۴۳۵۱
	مقیاس	دنباله زوج	۱.۱۴۹۶۰
		دنباله فرد	۰.۸۶۹۸۶

۳- پیاده‌سازی فیلترها با ضرایب اعشاری و صحیح

ضرایب اعشاری فیلترهای ۴-Daub، ۵/۳-CDF، ۹/۷-CDF با رویکرد پیچش و ۵/۳-CDF و ۹/۷-CDF به روش پلکانی جهت انجام شدن عملیات میانی به صورت صحیح درآورده و در جدول ۳ و ۴ ارائه گردیده است. با توجه به صحیح شدن ضرایب، عملیات میانی شامل ضرب و جمع اعشاری را می‌توان با استفاده از ضرب و جمع صحیح و گردش به راست انجام داد.

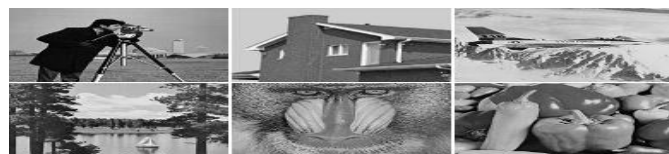
جدول ۳: ضرایب بالا و پایین گذر صحیح پیشنهادی برای ۴-Daub، ۵/۳-CDF با رویکرد پیچش.

نام فیلتر	ضرایب پایین گذر
Daub-۴	{ $4241 \times 2^{-15}, 7345 \times 2^{-15}, 27411 \times 2^{-15}, 15826 \times 2^{-15}$ }
	{ $-15826 \times 2^{-15}, 27411 \times 2^{-15}, 7345 \times 2^{-15}, -4241 \times 2^{-15}$ }
CDF-۵/۳	{ $-1 \times 2^{-2}, 2 \times 2^{-2}, 6 \times 2^{-2}, 2 \times 2^{-2}, -1 \times 2^{-2}$ }
	{ $-4 \times 2^{-2}, 8 \times 2^{-2}, -4 \times 2^{-2}$ }
CDF-۹/۷	{ $877 \times 2^{-15}, -52 \times 2^{-15}, -2563 \times 2^{-15}, 8744 \times 2^{-15}, 19757 \times 2^{-15}, 8744 \times 2^{-15}, -2563 \times 2^{-15}, -52 \times 2^{-15}, 877 \times 2^{-15}$ }
	{ $1495 \times 2^{-14}, -943 \times 2^{-14}, -9687 \times 2^{-14}, 18270 \times 2^{-14}, -9687 \times 2^{-14}, -943 \times 2^{-14}, 1495 \times 2^{-14}$ }

۴- ارزیابی و نتایج پیاده‌سازی پیشنهادی

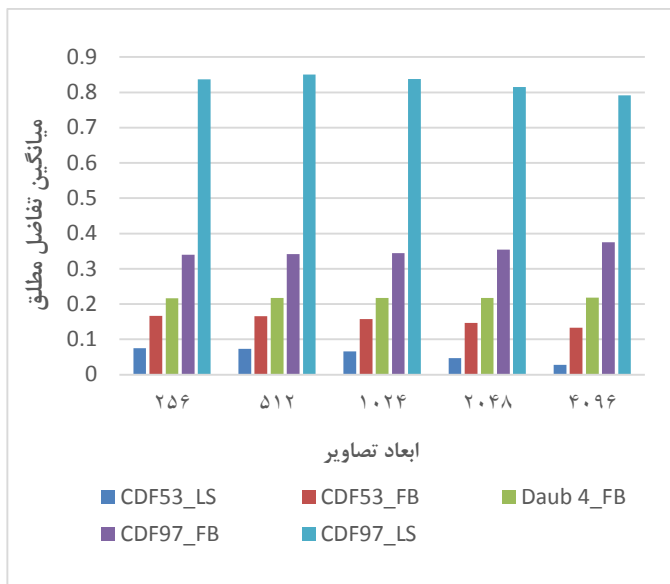
۴-۱- محک‌ها و محیط پیاده‌سازی

شش تصویر معروف از تصاویر محک در پردازش تصویر که در شکل ۳ نشان داده شده‌اند برای ارزیابی کارایی پیاده‌سازی اعشاری و صحیح پیشنهادی مورد استفاده قرار گرفته‌اند. پیاده‌سازی‌ها و نتایج حاصل از آن بر روی سیستم کامپیوتری با مشخصات جدول ۴ و زبان برنامه‌نویسی C انجام شده است.



شکل ۳: تصاویر محک استفاده شده برای ارزیابی کارایی.

پیشنهادی با توجه به شکل ۵ استفاده از معیار میانگین تفاضل مطلق 3^3 برای همه‌ی فیلترها کمتر از یک است.



شکل ۵: میانگین تفاضل مطلق برای پیکسل‌های پیاده‌سازی صحیح ۱۶ بیتی نسبت به پیاده‌سازی اعشاری در سه فیلتر Daub-۴، CDF-۵/۳ و CDF-۹/۷ به روش پیچش و دو فیلتر CDF-۵/۳ و CDF-۹/۷ به روش پلکانی.

۵- نتیجه‌گیری

با توجه به بالا بودن پیچیدگی محاسباتی و میزان حافظه مصرفی تبدیل موجک گسسته در این مقاله پیاده‌سازی صحیح ۱۶ بیتی برای این تبدیل ارائه گردیده است. نتایج حاصل از پیاده‌سازی پیشنهادی برای سه فیلتر Daub-۴، CDF-۵/۳ و CDF-۹/۷ با رویکرد پیچش و دو فیلتر CDF-۵/۳ و CDF-۹/۷ به روش پلکانی بر روی تصاویر محک با ابعاد مختلف، کاهش ۵۰ درصدی حافظه‌ی مصرفی را نشان می‌دهد. با توجه به میزان کاهش حافظه‌ی مصرفی تعداد فقدان‌های کش نیز در این روش بهبود یافته و نسبت فقدان کش پیاده‌سازی صحیح ۱۶ بیتی برای فیلترهای مختلف با ابعاد متفاوت نسبت به پیاده‌سازی اعشاری تا حداکثر ۱۹ برابر کاهش یافته است. در نهایت، با میزان خطای کمتر از یک، کارایی پیاده‌سازی پیشنهادی نسبت به پیاده‌سازی اعشاری تا دو برابر افزایش یافته است.

مراجع

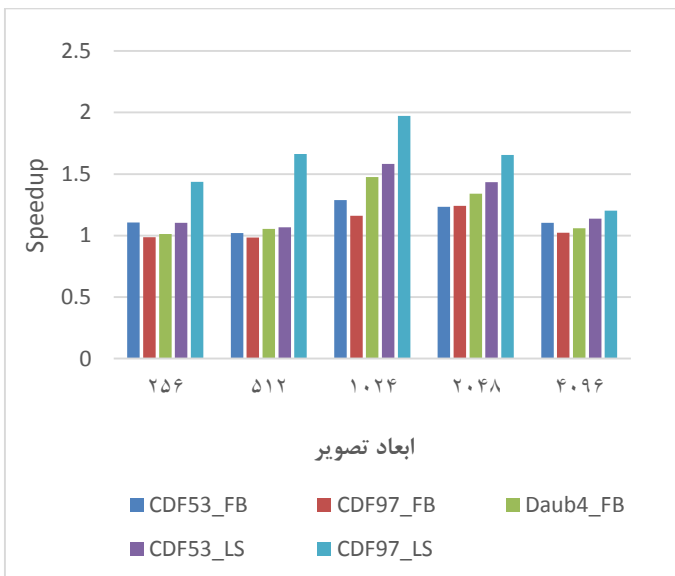
- [۱] I. Daubechies, "The wavelet transform time-frequency localization and signal analysis," IEEE Trans. Inform. Theory, vol. 36, pp. 961-1004, Sept. 1990.
- [۲] JPEG2000 Part II Final Committee Draft (2000, Dec.). [Online]. Available: <http://www.jpeg.org>
- [۳] P.P. Dang and P.M. Chau, "Integer fast wavelet transform and its VLSI implementation for low power applications," in Proc. IEEE Workshop Signal Processing Systems (SIPS), pp. 93-98, Oct 2002.
- [۴] A. Shahbahrami, B.H.H. Juurlink, S. Vassiliadis, "Performance Comparison of SIMD Implementations of the Discrete Wavelet Transform," Proceedings of the 11th IEEE Int. Conf. on Application-Specific Systems Architectures and Processors (ASAP), Samos, Greece, pp. 393-398, July 2000.
- [۵] P. Hammarlund, "4th generation Intel™ Core processor, codenamed Haswell," 2013 IEEE Hot Chips 25 Symposium (HCS), pp. 1-20, Aug. 2013.

از متغیر صحیحی که از ۱۶ بیت برای انجام عملیات استفاده می‌کند، انجام داد. به همین دلیل، در ادامه‌ی مقاله، این پیاده‌سازی را پیاده‌سازی صحیح ۱۶ بیتی می‌نامیم. در نتیجه میزان حافظه مصرفی برای این روش ۵۰ درصد نسبت به روش اعشاری کاهش و در نتیجه فقدان کش پیاده‌سازی صحیح ۱۶ بیتی برای فیلترهای مختلف با ابعاد متفاوت نسبت به پیاده‌سازی اعشاری، همان‌طور که در شکل ۷ نشان داده شده، بهبود یافته است.

جدول ۷: میانگین نسبت فقدان کش پیاده‌سازی اعشاری به صحیح ۱۶ بیتی برای فیلترهای Daub-۴، CDF-۵/۳، CDF-۹/۷ با رویکرد پیچش و CDF-۵/۳ و CDF-۹/۷ به روش پلکانی بر روی ۶ تصویر نشان داده شده در شکل ۳ با ابعاد مختلف.

اندازه‌های تصویر		۴۰۹۶	۲۰۴۸	۱۰۲۴	۵۱۲	۲۵۶	
رویکرد پیچش	CDF-۵/۳	۱,۱۰۷	۱,۴۵۸	۱۰,۲۳۰	۱,۵۶۰	۳,۴۱۱	
	CDF-۹/۷	۱,۰۹۲	۱,۶۷۴	۱۷,۳۱۷	۱,۱۳۸	۱,۸۰۴	
	Daub-۴	۱,۱۱۲	۱,۶۱۶	۱۷,۶۶۷	۱,۲۸۴	۱,۲۱۹	
روش پلکانی	CDF-۵/۳	۱,۰۹۶	۱,۴۹۱	۱۴,۱۷۰	۱,۱۴۶	۱,۶۴۸	
	CDF-۹/۷	۱,۰۷۶	۱,۶۷۶	۱۹,۰۲۵	۱,۹۵۰	۳,۵۷۳	

با توجه به بیشتر بودن نسبت فقدان کش در پیاده‌سازی اعشاری به پیاده‌سازی صحیح ۱۶ بیتی، همان‌طور که در نمودار شکل ۴ نشان داده شده، میانگین کارایی پیاده‌سازی صحیح ۱۶ بیتی نسبت به پیاده‌سازی اعشاری بر روی تصاویر محک نشان داده شده در شکل ۳ برای ابعاد مختلف بهبود یافته است. همان‌طور که در شکل ۴ دیده می‌شود این میزان بهبود برای فیلترهای CDF-۵/۳ و CDF-۹/۷ به روش پلکانی نسبت به فیلترهای Daub-۴، CDF-۵/۳، CDF-۹/۷ با رویکرد پیچش بیشتر است و بیشترین میزان کارایی از نظر ابعاد تصویر مربوط به تصاویر ۱۰۲۴ * ۱۰۲۴ است که به علت زیاد بودن فقدان کش اعشاری در این سایز تصویر نسبت به صحیح ۱۶ بیتی است. چراکه، میزان حافظه کش L3 محیط پیاده‌سازی برابر ۴ MB است.



شکل ۴: میانگین کارایی پیاده‌سازی صحیح ۱۶ بیتی نسبت به پیاده‌سازی اعشاری برای سه فیلتر Daub-۴، CDF-۵/۳ و CDF-۹/۷ با رویکرد پیچش و دو فیلتر CDF-۵/۳ و CDF-۹/۷ به روش پلکانی بر روی تصاویر نشان داده شده در شکل ۳ با ابعاد مختلف.

همواره پیاده‌سازی صحیح نسبت به اعشاری به دلیل گرد شدن ضرایب و عملیات میانی دارای خطا است؛ اما میزان خطابه ازای هر پیکسل برای روش

- [۶] H. Liao, M.K. Mandal, "Efficient Architectures for ۱-D and ۲-D Lifting-Based Wavelet Transforms", IEEE Trans. Signal Processing, Vol. ۵۲, No. ۵, May ۲۰۰۴.
- [۷] I. Daubechies, "Orthonormal bases of compactly supported wavelets," Comm. Pure Appl. Math, pp. ۹۰۹_۹۹۶.
- [۸] M.E. Angelopoulou, K. Masselos, P.Y.K. Cheung and Y. Andereopoulos, "Implementation and Comparison of the Δ^3 Lifting ۲D Discrete Wavelet Transform Computation Schedules on FPGAs", Journal of Signal Processing Systems, Vol. ۲۵, pp ۳-۲۱, April ۲۰۰۸.
- [۹] I. Daubechies, "Ten Lectures on Wavelets," CBMS-NSF Regional Conf. Series in Appl. Math., Society for Industrial and Applied Mathematics, Philadelphia, PA. vol. ۶۱.
- [۱۰] A. Shahbahrami, B.H.H. Juurlink, and S. Vassiliadis, "Implementing the ۲-D Wavelet Transform on SIMD-Enhanced General-Purpose Processors" IEEE Trans., Multimedia, Vol. ۱۰, No. ۱, pp. ۴۳-۵۱, Jan ۲۰۰۸.
- [۱۱] M. Mohan, B.K. Sujatha, "Efficient Security for Data Transmission Using CDF- σ/τ Lift-DWT and LSB Technique," IOSR Journal of VLSI and Signal Processing (IOSR-JVSP), Vol. ۶, Issue ۲, pp. ۰۱-۰۶, Apr ۲۰۱۶.
- [۱۲] I. Daubechies and W. Sweldens, "Factoring Wavelet Transforms into Lifting Steps." The Journal of Fourier Analysis and Applications, Vol. ۴, Issue ۳, ۱۹۹۸.

زیرنویس ها

-
۱. Discrete Wavelet Transform
 ۲. Compression
 ۳. Watermarking
 ۴. Pattern Rrecognitions
 ۵. Discrete Fourier Transform
 ۶. Convolution Approach
 ۷. Lifting Scheme
 ۸. Cache Misses
 ۹. Tradeoff
 ۱۰. Mallat Tree Algorithm
 ۱۱. Low Pass Filter
 ۱۲. High Pass Filter
 ۱۳. Daubechies
 ۱۴. Harley
 ۱۵. Solden
 ۱۶. Split
 ۱۷. Predict
 ۱۸. Update
 ۱۹. Benchmark
 ۲۰. Scale
 ۲۱. Speedup
 ۲۲. Cache Miss Ratio
 ۲۳. Mean Absolute Difference