

ارائه معماری FFT راستای سمت در روش تصویربرداری برد-داپلر در رادارهای SAR

سعیدرضا محسنی^{۱*}، محمد مهدی نایی^۲، رضا محسنی^۳، بهروز ابراهیمی

۱- کارشناس ارشد ۲- استاد، دانشکده برق، دانشگاه صنعتی شریف ۳- استادیار، دانشکده برق و الکترونیک، دانشگاه صنعتی شیراز

۴- کارشناس ارشد، دانشکده برق، دانشگاه صنعتی شریف

(دریافت: ۱۴/۰۵/۹۳، پذیرش: ۰۳/۰۳/۹۴)

چکیده

یکی از بخش‌های اصلی روش تصویربرداری برد-داپلر در رادارهای SAR که منابع محاسباتی، حافظه زیاد و همچنین زمان حل طولانی نیاز دارد، FFT راستای سمت می‌باشد. در مطالعه کنونی ابتدا با تجزیه کردن رابطه FFT و سپس ساده‌سازی و تبدیل آن به FFTهای کوچک‌تر رابطه‌ای به دست می‌آید که عمل پیاده‌سازی را آسان‌تر می‌کند. با تبدیل به FFT در نهایت روش ارائه شده با استفاده از تکنیک حذف فاکتورهای فاز اضافی باعث کاهش بسیار منابع محاسبات، زمان حل و حافظه‌ها در سخت‌افزار می‌شود. بررسی عملکرد روش پیشنهادی نشان می‌دهد که یافتن بهترین معماری به پارامترهایی از جمله کیفیت تصویر، طبقات بعدی پردازش و چند نقطه‌ای بودن FFT بستگی دارد. با ارائه این معماری، پیاده‌سازی کل روش تصویربرداری برد-داپلر در سخت‌افزارهای کوچک‌تر نیز عملی خواهد شد.

واژگان کلیدی

رادارهای روزنه ترکیبی، راستای سمت، روش محاسبه سریع تبدیل فوریه، پیاده‌سازی، تک مسیر همراه با فیدبک.

۱. مقدمه

هنگام عبور از پردازنده وجود ندارد. طراحی معماری پردازنده خطلوله‌ای موضوع تحقیقات اصلی دهه ۱۹۷۰ میلادی بوده است [۴ و ۱۸]. در متون چاپ شده دو دهه اخیر، معماری خطلوله‌ای FFT معمولاً بر دو نوع اساسی یافت می‌شود که می‌توان به کموتاتور تأخیر^۱ و فیدبک تأخیر^۲ اشاره کرد. معماری خطلوله‌ای FFT بر اساس تعداد خطوط داده را می‌توان به دو دسته تک مسیره و چند مسیره تقسیم‌بندی کرد [۸-۶]. ایده‌ی مبنای r^k الگوریتم، برای دستیابی همزمان به ساختار پروانه‌ای ساده‌تر و کاهش عامل‌های ضرب کاملاً مناسب می‌باشد و تا حدودی معماری را تحت تأثیر قرار می‌دهد [۹-۱۰]. مبنای پایین‌تر، ساختار پروانه‌ای ساده‌تری دارد ولی مبنای بالاتر تعداد عامل‌های ضرب را کاهش می‌دهد. مرجع [۱۱] نشان می‌دهد که محصول کرونکر^۳ برای اجرا و

FFT یک روش بسیار کارآمد در بسیاری از شاخه‌ها مانند مخابرات، پردازش تصویر، پردازش سیگنال‌های راداری، سنسار، تحقیقات زیست پزشکی و ... می‌باشد. به عنوان یک تکنیک کلیدی در پردازش سیگنال دیجیتال رادار، روش‌های بسیاری برای تحقق FFT مورد بررسی قرار می‌گیرد. پردازنده خطلوله‌ای FFT یک کلاس خاص از پردازش، برای محاسبه FFT با استفاده از روش‌های محاسبه سریع می‌باشد. استفاده از پردازنده خطلوله‌ای FFT، یکی از راه‌های بالا بردن سرعت پردازش و کاهش زمان پردازش می‌باشد [۵-۴]. از خصوصیات پردازش زمان واقعی در پردازنده خطلوله‌ای این است که هیچ توقفی بین پردازش دنباله داده‌ها به

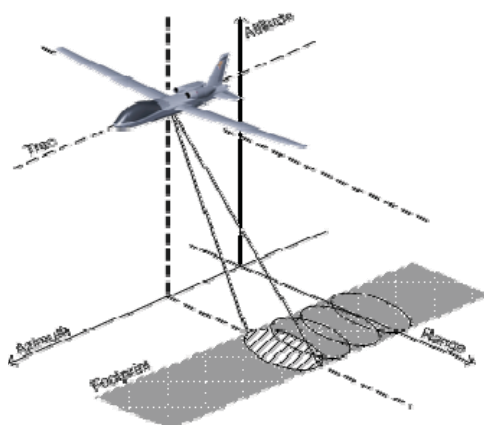
^۱ Delay Commutator

^۲ Delay Feedback

^۳ Kronecker

مهمترین و اصلی‌ترین مسئله در فرآیند تصویربرداری به کمک رادارهای روزنه ترکیبی تلقی می‌شود [۱-۲]. SAR به‌طور گسترده‌ای در بسیاری از زمینه‌ها از جمله، کشف منابع زیر زمینی و اهداف استتار شده، کشف آثار باستانی مدفون شده، سامانه فرود هوشمند هواپیما و ... مورد استفاده قرار می‌گیرد [۱].

مطابق شکل ۱، در SAR سطح زمین به سلول‌های کوچکی در بعد فاصله^۴ (که ارسال سیگنال در این جهت می‌باشد) و سمت^۵ (که جهت حرکت سکو در این راستا می‌باشد) تقسیم می‌گردد. توان دریافتی از هر سلول کوچک در پردازش تصویر SAR محاسبه شده و با توجه به شدت و ضعف توان فاصله هر سلول، RCS^۶ سلول مورد نظر تخمین زده می‌شود. این RCS‌های تخمینی تصویر نهایی SAR را تشکیل می‌دهند.



شکل ۱. نحوه تصویربرداری مد نواری در SAR

در خروجی پردازش SAR برای تشکیل یک تصویر مناسب و قابل قبول، به سیگنال به نویز در حدود ۱۰ db تا ۱۵ db نیاز است. یکی از مهمترین روش‌هایی که برای تشکیل تصویر استفاده می‌شود، روش برد-داپلر^۷ (RDA) می‌باشد. روش‌های متعددی جهت استخراج تصویر SAR وجود دارد که یکی از مهم‌ترین آن‌ها روش برد-داپلر است. یکی از مزیت‌های مهم این روش کیفیت تصویربرداری با وضوح نسبتاً عالی می‌باشد.

بلوک دیاگرام پردازشی این روش تصویربرداری برای زاویه کوچکی^۸ کم در شکل ۲ ارائه شده است [۲]. این مراحل عبارتند از:

- ❖ فشرده سازی سیگنال در راستای برد
- ❖ اعمال FFT در راستای سمت (بردن سیگنال به حوزه فرکانس)
- ❖ الگوریتم RCMC
- ❖ فشرده سازی سیگنال در راستای سمت
- ❖ اعمال IFFT در راستای سمت

پیاده‌سازی مدارات VLSI کاربرد دارد. ارتباط بین محصول کروئکر و پیاده‌سازی سخت‌افزاری برای تولید خودکار مبنای معماری FFT به‌کار گرفته می‌شود [۱۲].

یکی از بهترین روش‌های پیاده‌سازی معماری خطلوله‌ای که از لحاظ منابع مصرفی بسیار مفید می‌باشد روش $R2^2SDF$ است. هدف اصلی در این پژوهش، یافتن روشی برای FFT راستای سمت در الگوریتم تصویربرداری برد-داپلر یا RDA است که پیاده‌سازی آن در سخت‌افزار مربوطه عملی شود. پیاده‌سازی این FFT به‌صورت معمول (حتی با استفاده از روش $R2^2SDF$ در سخت‌افزار FPGA از نوع $virtex4^4$ عملی نمی‌باشد چون بزرگ بودن FFT راستای سمت حجم بالایی از منابع را اشغال می‌کند. با توجه به طراحی سیستمی موجود تعداد نقاط FFT در اینجا بسیار زیاد می‌باشد و این کار پیاده‌سازی را با مشکل مواجه می‌کند. با استفاده از تکنیکی که در بخش سوم به آن اشاره شده است پیاده‌سازی این FFT در این سخت‌افزار عملی می‌شود و حجم منابع و زمان اجرا را به شدت کاهش می‌دهد.

ابتدا در بخش دوم به تعریف صورت مساله روش برد-داپلر و علت بزرگ شدن تعداد نقاط FFT در راستای سمت پرداخته می‌شود. سپس در بخش سوم مروری بر روش‌های پیاده‌سازی خطلوله‌ای FFT خواهد شد. در بخش چهارم پیاده‌سازی FFT راستای سمت و معماری ارائه شده مورد بررسی قرار می‌گیرد. در بخش پنجم نتایجی که از حاصل شبیه‌سازی و پیاده‌سازی به‌دست می‌آید، بررسی می‌گردد. در بخش آخر نیز نتیجه‌گیری صورت می‌گیرد.

۲. تعریف مسأله

رادارهای روزنه ترکیبی یا SAR یکی از پرکاربردترین و مهمترین ابزارهای مورد استفاده در تصویربرداری می‌باشد. رادارهای روزنه ترکیبی به خوبی می‌تواند تصویری جالب از منطقه‌ای خاص در طول شبانه روز و تمام شرایط آب و هوایی را فراهم کند. بر اساس ایده‌ی کارل ویلی^۱، آنتن کوچکی توسط یک سیستم پرنده حمل می‌شود و در موقعیت‌های مشخص، منطقه مورد نظر را تحت تابش الکترومغناطیسی قرار داده و بازتاب آن را از محیط دریافت می‌کند [۱]. پس از ثبت داده‌های بازگشتی، پردازش‌های لازم بر روی داده‌های خام^۲ که حاوی اطلاعات پالس‌های متوالی بازگشتی است صورت گرفته و در آخر تصویر مربوطه تولید می‌گردد. به این ترتیب به کمک روش‌های پردازش سیگنال می‌توان از داده‌های خام راداری که قدرت تفکیک بسیار نامناسبی دارد، تصاویری با قدرت تفکیک مناسب استخراج کرد. از این رو است که پردازش سیگنال

^۵ Azimuth

^۶ Radar Cross Section

^۷ Range Doppler Algorithm

^۸ Squint angle

^۱ XC4VSX55-10FFG1148

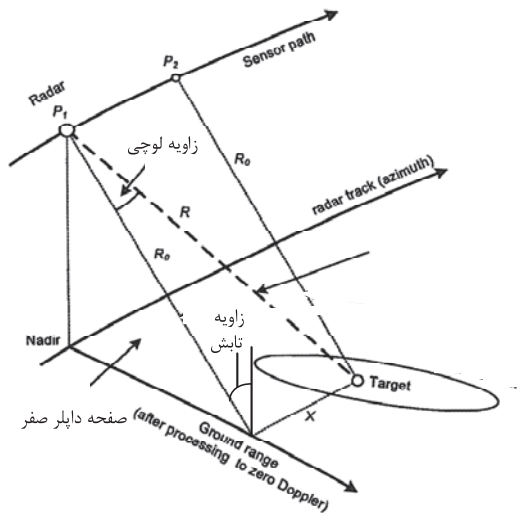
^۲ Carl A. Wiley

^۳ Raw Data

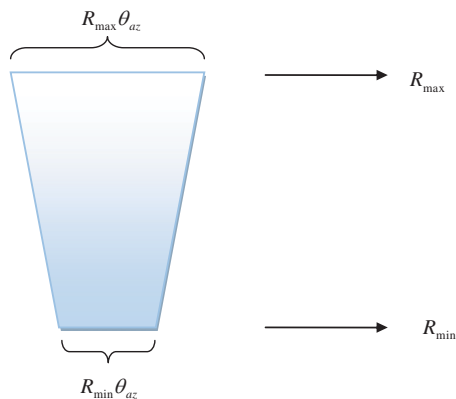
^۴ Range



شکل ۲. نمودار بلوکی روش تصویربرداری برد- دایر [۲]



شکل ۳. هندسه رادار روزنه ترکیبی



شکل ۴. کل ناحیه روشن شده در سطح زمین بین بازه R_{min} تا R_{max} مرتبه دو

اگر تعداد نمونه‌های سیگنال LFM در محدوده پهنای پرتو آنتن برابر با N_a باشد، طول فیلتر منطبق حوزه زمان نیز برابر با N_a در نظر گرفته می‌شود.

در سامانه رادارهای روزنه ترکیبی معمولاً از حوزه فرکانس برای پیاده سازی فیلتر منطبق سیگنال LFM استفاده می‌شود. لذا به اندازه طول فیلتر به کار رفته باید داده اضافی، برداشته شود تا اثر مخرب فیلتر روی داده‌های ابتدایی جبران گردد.

زمان مرور θ_{az} از هدف با فاصله R_0 طبق رابطه (۱) محاسبه می‌گردد [۲]. در اینجا R_0 کوتاه‌ترین فاصله‌ای است که رادار در مسیر حرکت خود با هدف مشخص شده خواهد داشت و V_r سرعت مؤثر

در این مطالعه، تمرکز بیشتری روی مرحله دوم است که در مورد نحوه پیاده‌سازی و مشکلاتی که در آن وجود دارد، توضیح داده می‌شود. هدف از مرحله دوم بردن سیگنال به حوزه فرکانس می‌باشد تا عملیات اصلاح منحنی RCM راحت‌تر و با دقت بالاتری صورت گیرد. با توجه به دلایلی که در ادامه بررسی می‌شود، تعداد نقاط FFT در این راستا بسیار زیاد می‌شود و کار پیاده‌سازی را با مشکلاتی روبرو می‌کند.

در رادارهای روزنه ترکیبی که با استفاده از روش مدنواری^۱ تصویربرداری می‌کند، باید یک رویه مشخص برای تصویربرداری از سطح زمین تعیین شود. بر اساس این رویه مشخص می‌شود که در حین حرکت پرنده و نمونه‌برداری با نرخ PRF در بعد سمت، هر قطعه از تصویر بر اساس چندین سمت از نمونه‌های دریافتی باید تشکیل گردد. همچنین باید مشخص شود که چگونه زمان بین دو تصمیم متوالی و تشکیل دو تصویر مجاور انتخاب گردد. پارامترهای موثر در تعیین این رویه عبارتند از:

- محدوده روشن شده در سطح زمین توسط پهنای پرتو آنتن در بعد سمت (که خود آن به زاویه راستای سمت، زاویه تابش و زاویه لوچی و همچنین ارتفاع پرواز وابسته است)
- سرعت حرکت هواپیما که معادل سرعت حرکت ناحیه روشن شده در سطح زمین می‌باشد.
- مقدار PRF انتخاب شده و در نتیجه تعداد نمونه‌های دریافتی از یک نقطه در حین مرور

محدودیت‌ها و قابلیت‌های پردازشی

با توجه به این‌که در طراحی ارائه شده زاویه لوچی وجود نداشت، این زاویه برابر صفر فرض شده است (شکل ۳). بر این اساس و با فرض مدل خطی برای حرکت پرنده مساحت ناحیه روشن شده در سطح زمین تقریباً برابر با $R \cdot \theta_{az}$ می‌باشد (در اینجا θ_{az} پهنای پرتو آنتن در راستای سمت می‌باشد). لذا در بازه بین R_{min} تا R_{max} این محدوده متغیر است و کل ناحیه‌ی روشن شده روی سطح زمین به صورت شکل ۴ خواهد بود.

بر این اساس تصویر نهایی از کنار هم قرار گرفتن تصویر نواحی دوزنقه‌ای مجاور تشکیل خواهد شد. در این زمینه چند نکته مهم باید در نظر گرفته شود که در ادامه به آنها اشاره می‌شود:

¹ Strip map

رادار می‌باشد.

۳. مروری بر روش‌های پیاده‌سازی خط‌لوله‌ای FFT

FFT، یک الگوریتم محاسباتی سریع و کارآمد برای محاسبه تبدیل فوریه گسسته یا DFT^1 (که تعداد نمونه‌های آن عدد مثبت توان دوم است) می‌باشد. تبدیل فوریه گسسته برای دنباله $x(n)$ به صورت زیر تعریف می‌شود [۳].

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N}, \quad k=0,1,2,\dots,N-1 \quad (4)$$

روش‌های متفاوتی برای پیاده‌سازی خط‌لوله‌ای پردازنده FFT وجود دارد که در زیر به طور مختصر چندین روش پر کاربرد بیان می‌شود:

• روش کموتاتور تأخیر چند مسیره مبنای ۲ یا R2MDC^۲

این روش ساده‌ترین و سرراست‌ترین روش برای پیاده‌سازی خط‌لوله‌ای از مبنای ۲ الگوریتم FFT می‌باشد. دنباله ورودی به دو جریان داده روبه جلو به صورت موازی تقسیم می‌شود و المان‌های ورودی با تأخیرات لازم به ساختار پروانه‌ای وارد می‌شوند [۱۳]. این روش اولین بار در سال ۱۹۷۵ شناخته شد ولی از آن برای FFT‌های بزرگ نمی‌توانستند استفاده کنند چون حجم بالایی از سخت‌افزار را اشغال می‌کند.

• روش فیدبک تأخیر تک مسیره مبنای ۲ یا R2SDF^۳

در این روش از رجیسترها یا حافظه‌ها به طور مؤثری برای ذخیره‌ی خروجی ساختار پروانه‌ای استفاده می‌شود. در هر مرحله فقط یک مسیر جریان داده وجود دارد [۱۴]. این روش برای اولین بار در سال ۱۹۸۴ مورد استفاده قرار گرفت. مزیت این روش نسبت به روش قبل، منابع مصرفی کمتر و همچنین کاهش زمان انجام پردازش می‌باشد.

$$T_a = \theta_{az} R(\eta_c) \frac{1}{V_r \cos(\theta_{sq})} \xrightarrow{\theta_{sq}=0} T_{az} = \frac{\theta_{az} R_0}{V_r} \quad (1)$$

با فرض نرخ نمونه‌برداری F_a (یا همان PRF)، تعداد نمونه‌ها در این محدوده برابر N_a خواهد بود.

$$N_a = T_a \times F_a = \frac{\theta_{az} R_0}{V_r} \times F_a \quad (2)$$

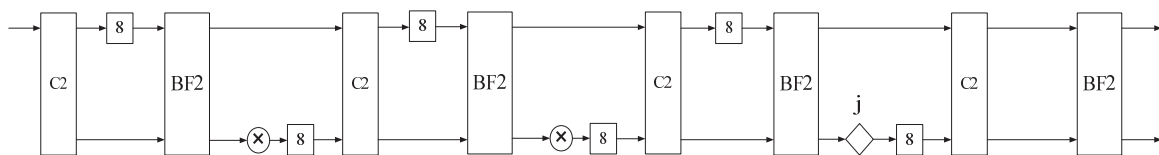
به دلیل این که مقدار PRF بیشتر از پهنای طیفی سیگنال است، طیف سیگنال در کل محدوده فرکانسی گسترده نمی‌شود. لذا اگر تعداد نقاط FFT برای تبدیل فوریه از سیگنال برابر N_a باشد، تعداد بین‌های حاوی سیگنال در طیف تقریباً از رابطه زیر به دست می‌آید.

$$N_s = \frac{\Delta f_{dopp}}{PRF} N_a \quad (3)$$

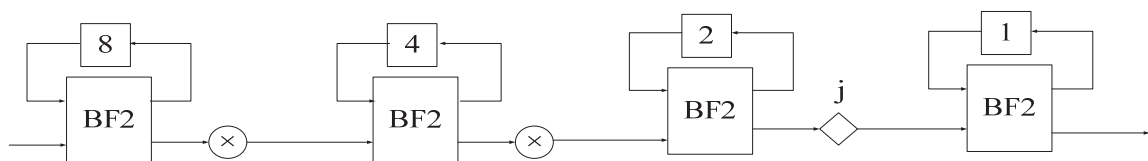
که در اینجا Δf_{dopp} طیف سیگنال در کل محدوده فرکانسی است. حال با توجه به رابطه‌های بالا و پارامترهایی که در جدول ۱ آمده است، تعداد نقاط FFT برای این طرح مورد نظر برابر با ۶۵۵۳۶ نقطه یا ۶۴ کیلو نقطه می‌شود. از این تعداد نقطه تقریباً مقدار ۸۱۹۲ نقطه یا ۸ کیلو نقطه حاوی اطلاعات است و برای ادامه پردازش به آن نیاز است.

جدول ۱. پارامترهای مورد نیاز مسأله

نماد	واحد	مقدار مورد نظر	پارامتر
V_r	m/s	≈ 50	سرعت سکو
R_0	Km	۴ تا ۸	فاصله رادار تا هدف
F_a	KHz	۳	مقدار PRF
θ_{az}	°	تقریباً ۶	پرتو آنتن در راستای سمت



شکل ۵. معماری R2MDC برای ۱۶ نقطه‌ای [۱۳]



شکل ۵. معماری R2SDF برای ۱۶ نقطه‌ای [۱۴]

² Radix-2 Multi-path Delay Commutator

³ Radix-2 Single-path Delay Feedback

¹ Discrete Fourier Transform

فقط مقادیر ۸۱۹۲ داده از ۶۴ کیلو خروجی را انتخاب کرده و برای ادامه پردازش از آن استفاده کرد. ولی پیاده‌سازی FFT با روش ذکر شده غیر عملی می‌باشد زیرا مقدار زیادی منابع محاسباتی و حافظه اشغال می‌کند که در سخت‌افزار مورد نظر^۱ موجود نمی‌باشد. یکی از دلایل دیگری که این روش مناسب نمی‌باشد این است که برای تعداد ۵۷،۳۴۴ داده محاسبات اضافی انجام می‌شود.

با توجه به ۸۱۹۲ داده خروجی مورد نیاز و هم‌چنین ملاحظات دیگر مانند منابع محاسباتی، بهترین روش این است که FFT به صورت ۸۱۹۲ نقطه‌ای پیاده‌سازی شود و عملیات محاسباتی فقط بر روی همان ۸۱۹۲ نقطه مورد نظر انجام شود. این روش تحت عنوان پیاده‌سازی FFT به روش جزئی شناخته می‌شود.

۴-۱. پیاده‌سازی FFT به روش جزئی

برای این که بتوان از این جزء کردن FFT استفاده کرد، ابتدا باید معادله FFT ۶۴ کیلو نقطه‌ای را به صورت ۸۱۹۲ نقطه‌ای درآورد. در روابط ذکر شده، مقدار N همان تعداد نقاط FFT در راستای سمت را نشان می‌دهد که برابر با ۶۵۵۳۶ یا ۶۴ کیلو نقطه است.

حال با توجه به رابطه (۴) باید FFT ۶۴ کیلو نقطه‌ای را به صورت FFT‌های ۸۱۹۲ نقطه‌ای تجزیه کرد.

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N} \quad (5)$$

$$= \sum_{n=0}^{N/8-1} x(8n)e^{-j2\pi k 8n/N} + \sum_{n=0}^{N/8-1} x(8n+1)e^{-j2\pi k(8n+1)/N}$$

$$+ \sum_{n=0}^{N/8-1} x(8n+2)e^{-j2\pi k(8n+2)/N} + \sum_{n=0}^{N/8-1} x(8n+3)e^{-j2\pi k(8n+3)/N}$$

$$+ \sum_{n=0}^{N/8-1} x(8n+4)e^{-j2\pi k(8n+4)/N} + \sum_{n=0}^{N/8-1} x(8n+5)e^{-j2\pi k(8n+5)/N}$$

$$+ \sum_{n=0}^{N/8-1} x(8n+6)e^{-j2\pi k(8n+6)/N} + \sum_{n=0}^{N/8-1} x(8n+7)e^{-j2\pi k(8n+7)/N}$$

• روش تاخیر فیدبک تک مسیره مبنای ۲ یا R2²SDF

این روش از ترکیب مبنای ۲ و مبنای ۴ الگوریتم FFT به دست می‌آید که به نام مبنای ۲۲ شناخته می‌شود. همان‌طور که بیان شد، استفاده از این روش باعث کاهش محاسبات می‌شود و هم‌چنین ساختار ساده پروانه‌ای شکل آن نیز راحت‌تر قابل پیاده‌سازی است [۱۵]. این روش اولین بار در سال ۱۹۹۶ مورد استفاده قرار گرفت و به دلیل این که حجم سخت‌افزاری مورد استفاده در آن را کاهش داده به سرعت مورد استفاده قرار گرفت. در سال ۲۰۰۴ یک FFT ۱۰۲۴ نقطه‌ای بر روی یک نوع FPGA شرکت پیاده‌سازی شد. این ساختار با توجه به آنچه در جدول ۲ آمده است، کمترین حجم محاسبات را دارد و در نتیجه برای پیاده‌سازی از این روش استفاده شده است.

در شکل ۷ دو ساختار پروانه‌ای وجود دارد که در بخش بعدی، این روش با ذکر جزئیات مورد بررسی قرار می‌گیرد. با توجه به روش‌هایی که در بالا بررسی شد همچنان پیاده‌سازی FFT و حتی زمانی که از روش R2²SDF استفاده گردد غیر عملی است چون منابع مصرفی در آن بسیار زیاد می‌باشد.

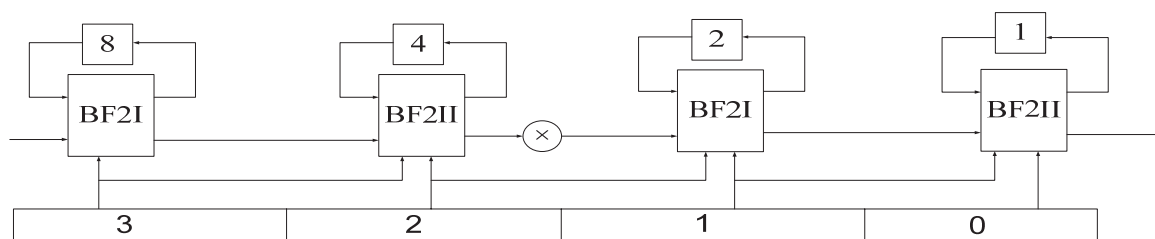
۴. طراحی و پیاده‌سازی خطلوله‌ای FFT راستای

سمت

یک روش برای پیاده‌سازی FFT راستای سمت، به صورت همان ۶۴ کیلو نقطه‌ای است. در این حالت می‌توان پس از انجام پردازش

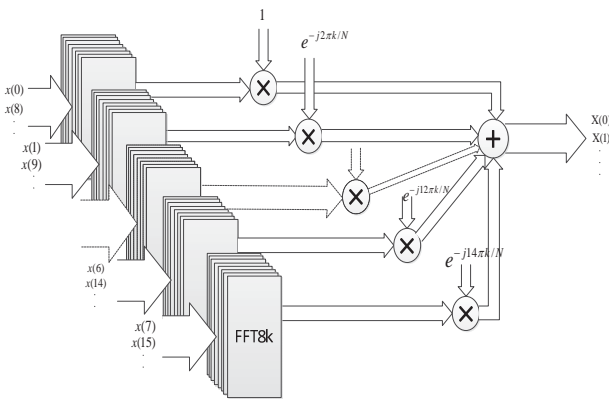
جدول ۲. مقایسه حجم محاسباتی برای روش‌های مختلف FFT [۱۵]

روش	ضرب مختلط	جمع	حجم حافظه	پیچیدگی کنترل
R2MDC	$2(\log_4 N - 1)$	$4\log_4 N$	$3N/2 - 2$	ساده
R2SDF	$2(\log_4 N - 1)$	$4\log_4 N$	$N - 1$	ساده
R22SDF	$\log_4 N - 1$	$4\log_4 N$	$N - 1$	ساده



شکل ۷. معماری R2²SDF برای FFT ۱۶ نقطه‌ای [۱۵]

^۱سخت‌افزار مورد استفاده در این طراحی از FPGAهای شرکت زایلینکس و نوع Virtex4(XC4VSX55) می‌باشد



شکل ۸. نمایش FFT راستای سمت به صورت FFT های ۸۱۹۲ نقطه‌ای

نیز بهینه شد. همان‌طور که در بخش یک نیز به آن اشاره گردید، برای پیاده‌سازی هسته FFT ۸۱۹۲ نقطه‌ای می‌توان از روش‌های موجود استفاده کرد که در اینجا از روش R^2SDF استفاده شده است. در بخش بعدی به توضیح این روش پرداخته شده است.

۲-۴. پیاده‌سازی FFT ۸۱۹۲ نقطه‌ای با روش R^2SDF

برای استخراج این روش جدید، ابتدا باید دو گام تجربه را به صورت زیر تعریف کرد [۱۵ و ۱۶]:

$$n = \frac{N}{2}n_1 + \frac{N}{4}n_2 + n_3 \quad \left(n_1 = n_2 = 0, 1; n_3 = 0, 1, \dots, \frac{N}{4} - 1 \right) \quad (9)$$

$$k = k_1 + 2k_2 + 4k_3 \quad \left(k_1 = k_2 = 0, 1; k_3 = 0, 1, \dots, \frac{N}{4} - 1 \right)$$

$$X(k_1 + 2k_2 + 4k_3) = \sum_{n_3=0}^{N/4-1} \sum_{n_2=0}^1 \sum_{n_1=0}^1 x\left(\frac{N}{2}n_1 + \frac{N}{4}n_2 + n_3\right) W_N^{\left(\frac{N}{2}n_1 + \frac{N}{4}n_2 + n_3\right)(k_1 + 2k_2 + 4k_3)}$$

$$= \sum_{n_3=0}^{N/4-1} \sum_{n_2=0}^1 \left\{ x\left(\frac{N}{4}n_2 + n_3\right) + (-1)^{k_1} x\left(\frac{N}{4}n_2 + n_3 + \frac{N}{2}\right) \right\} W_N^{\left(\frac{N}{4}n_2 + n_3\right)(k_1 + 2k_2 + 4k_3)} \quad (10)$$

حال با ساده‌سازی رابطه (۱۰) در نهایت رابطه زیر به دست خواهد آمد [۱۵ و ۱۶].

$$X(k_1 + 2k_2 + 4k_3) = \sum_{n_3=0}^{N/4-1} [H(k_1, k_2, n_3) W_N^{n_3(k_1 + 2k_2)}] W_N^{n_3 k_3} \quad (11)$$

در رابطه (۱۱) مقدار $H(k_1, k_2, n_3)$ به صورت رابطه (۱۲) به دست می‌آید: در رابطه (۱۲) دو ساختار پروانه‌ای به دست می‌آید که

حال اگر قسمت‌های $m = 0, 1, \dots, 7$ از سیگماها بیرون کشیده شود (چون ثابت هستند)، نتیجه زیر حاصل می‌شود.

$$X(k) = \sum_{n=0}^{N/8-1} x(8n) e^{-j2\pi k 8n/N} + e^{-j2\pi k/N} \sum_{n=0}^{N/8-1} x(8n+1) e^{-j2\pi k 8n/N} \quad (6)$$

$$+ e^{-j4\pi k/N} \sum_{n=0}^{N/8-1} x(8n+2) e^{-j2\pi k 8n/N} + e^{-j6\pi k/N} \sum_{n=0}^{N/8-1} x(8n+3) e^{-j2\pi k 8n/N} +$$

$$e^{-j8\pi k/N} \sum_{n=0}^{N/8-1} x(8n+4) e^{-j2\pi k 8n/N} + e^{-j10\pi k/N} \sum_{n=0}^{N/8-1} x(8n+5) e^{-j2\pi k 8n/N} +$$

$$e^{-j12\pi k/N} \sum_{n=0}^{N/8-1} x(8n+6) e^{-j2\pi k 8n/N} + e^{-j14\pi k/N} \sum_{n=0}^{N/8-1} x(8n+7) e^{-j2\pi k 8n/N}$$

این رابطه در نهایت به صورت زیر به دست می‌آید.

$$X(k) = \sum_{n=0}^{N/8-1} x(8n) e^{-j2\pi k n/(N/8)} + e^{-j2\pi k/N} \sum_{n=0}^{N/8-1} x(8n+1) e^{-j2\pi k n/(N/8)} \quad (7)$$

$$+ e^{-j4\pi k/N} \sum_{n=0}^{N/8-1} x(8n+2) e^{-j2\pi k n/(N/8)} + e^{-j6\pi k/N} \sum_{n=0}^{N/8-1} x(8n+3) e^{-j2\pi k n/(N/8)}$$

$$+ e^{-j8\pi k/N} \sum_{n=0}^{N/8-1} x(8n+4) e^{-j2\pi k n/(N/8)} + e^{-j10\pi k/N} \sum_{n=0}^{N/8-1} x(8n+5) e^{-j2\pi k n/(N/8)}$$

$$+ e^{-j12\pi k/N} \sum_{n=0}^{N/8-1} x(8n+6) e^{-j2\pi k n/(N/8)} + e^{-j14\pi k/N} \sum_{n=0}^{N/8-1} x(8n+7) e^{-j2\pi k n/(N/8)}$$

در حقیقت رابطه (۷) به صورت مجموعی از FFT های ۸۱۹۲ نقطه‌ای است با این تفاوت که ورودی هر یک از این FFT ها متفاوت است. تعداد ضرایبی که در سیگماها ضرب شده‌اند ۶۵۵۳۶ یا به عبارت دیگر به اندازه تعداد خروجی‌هاست. رابطه (۷) را می‌توان مانند شکل ۸ نیز نمایش داد.

نکته جالب این رابطه این است که به راحتی می‌توان به اندازه تعداد خروجی‌های مورد نیاز این رابطه را اجرا کرد و سپس هشت مرتبه این عمل را تکرار کرده تا رابطه (۸) به دست آید.

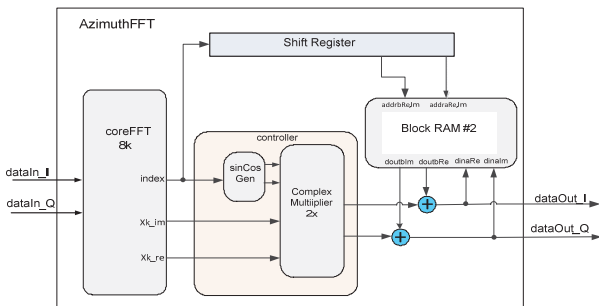
$$X(k) = \sum_{m=0}^7 e^{-j2\pi m k/N} \sum_{n=0}^{N/8-1} x(8n+m) e^{-j2\pi k 8n/N} \quad (8)$$

شکل ۸ کاملاً رابطه (۸) را تفسیر می‌کند که اگر هر یک از FFT های ۸۱۹۲ نقطه‌ای، هشت مرتبه تکرار شود و سپس با ضرب در فازهای، در نهایت خروجی‌های FFT راستای سمت آماده خواهد شد. بنابراین برای داشتن ۸۱۹۲ داده خروجی، کفایست که خروجی FFT های ۸۱۹۲ نقطه‌ای را در ۸ ضرایب ضرب کرده تا در نهایت ۸۱۹۲ خروجی به دست آید. با این روند از محاسبات اضافی صرف نظر خواهد شد و به همین ترتیب حافظه و منابع محاسباتی به کار رفته در آن کمتر خواهد شد.

برای به دست آوردن ۸۱۹۲ خروجی کافی است که مطابق شکل ۸ عملیات ضرب فازها به اندازه ۸۱۹۲ تا باشد. در این پژوهش همچنین برای کاهش منابع محاسباتی، هسته FFT ۸۱۹۲ نقطه‌ای

۳-۴. معماری ارائه شده برای FFT راستای سمت

با توجه به آنچه در بخش های قبلی ذکر شد ۸۱۹۲ ضریب برای پیاده سازی این FFT نیاز می باشد. برای پیاده سازی این ضرایب نیز روش های مختلفی وجود دارد که یکی همان ذخیره کردن ۸۱۹۲ ضریب در حافظه است، دیگری استفاده از روش های ^۱CORDIC و فرمول های مثلثاتی است. در اینجا برای پیاده سازی این ضرایب از فرمول های مثلثاتی استفاده شده است. در شکل ۱۱ این ضرایب توسط ماژول SinCosGen انجام خواهد شد.



شکل ۱۱. معماری کلی برای پیاده سازی FFT در راستای سمت

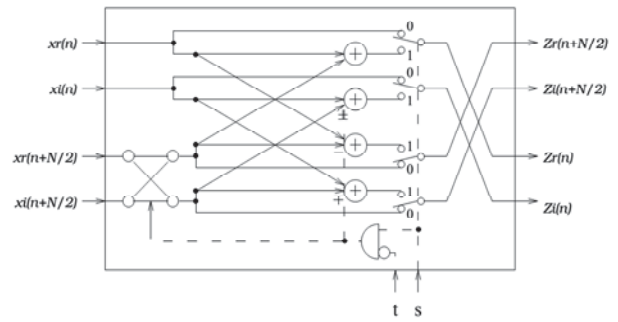
وظیفه کنترلر که در شکل ۱۱ نشان داده شده است به صورت زیر می باشد:

- محاسبه ضرایب به وسیله ماژول SinCosGen
- ضرب خروجی هر یک از FFT های ۸۱۹۲ نقطه ای در ضرایب مورد نظر بوسیله ماژول ComplexMultiplier2x

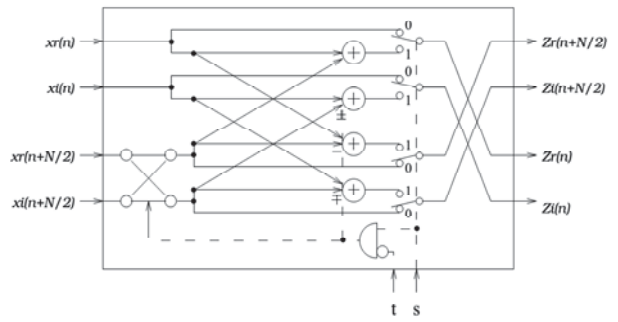
بعد از انجام محاسبه FFT ۸۱۹۲ نقطه ای و سپس ضرب در ضرایب مورد نظر توسط کنترلر باید داده های ذخیره شده در حافظه (یا خروجی کنترلر که در مرحله قبلی در حافظه ذخیره شده است) با داده های خروجی از کنترلر در مرحله فعلی جمع شود. ولی بعد از این که هشت مرحله این عمل تکرار شد باید داده ها به خروجی تحویل داده شود و مقادیر داخل حافظه پاک شود.

در طرح معماری شکل ۱۱ ورودی های هر بسته داده (یا ۶۵۵۳۶ با ۶۴ کیلو داده) به صورت خط لوله وارد می شوند. بعد از گذشت مدت زمانی برای انجام محاسبات، ۸۱۹۲ داده را به صورت کاملاً پشت

$$H(k_1, k_2, n_3) = \underbrace{\left[x(n_3) + (-1)^{k_1} x\left(n_3 + \frac{N}{2}\right) \right]}_{\text{BFI}} + \underbrace{\left[(-j)^{(k_1+2k_2)} \left[x\left(n_3 + \frac{N}{4}\right) + (-1)^{k_1} x\left(n_3 + \frac{3N}{4}\right) \right] \right]}_{\text{BFII}} \quad (12)$$



الف) BFI



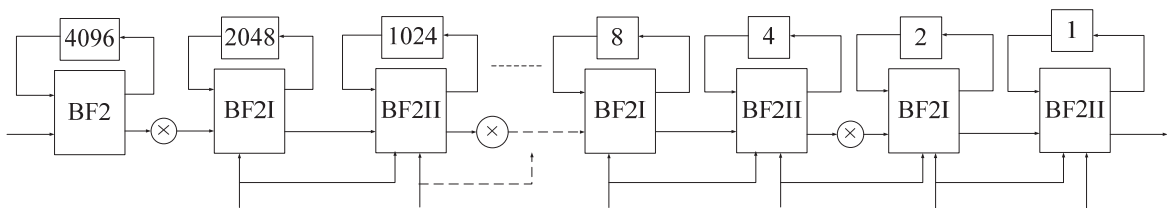
ب) BFII

شکل ۹. ساختار پروانه ای برای روش R2²SDF

به صورت شکل ۹: در عمل قابل پیاده سازی می باشد. ساختاری که برای روش R2²SDF ارائه شده برای FFT هایی که توانی از ۴ می باشند قابل استفاده است.

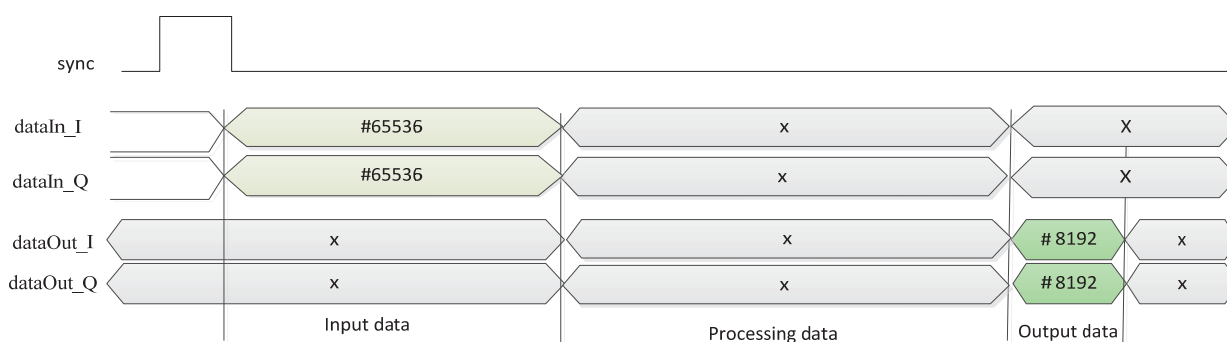
برای حل این مشکل باید در انتهای ساختار پروانه ای، روش R2SDF را اضافه کرد تا بتوان FFT های با توان ۲ نیز ساخت [۱۷].

با توجه به ساختار پروانه ای که در شکل ۹ نشان داده شد و رابطه هایی که در بالا به آن پرداخته شد در نهایت معماری به صورت شکل ۱۰ برای پیاده سازی FFT ۸۱۹۲ نقطه ای به دست می آید.

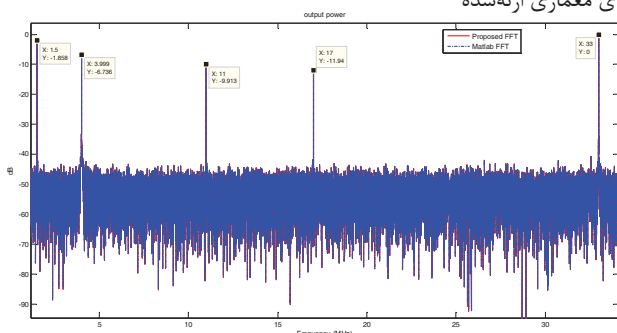


شکل ۱۰. ساختار هسته ای FFT ۸۱۹۲ نقطه ای

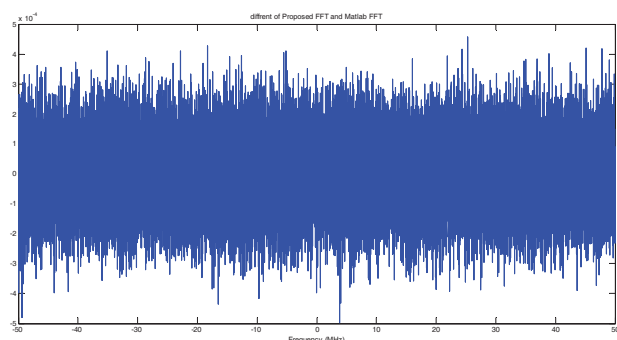
^۱ Coordinate Rotation Digital Computer



شکل ۱۲. نمودار زمانی برای معماری ارائه شده



شکل ۱۳. مقایسه فرکانس و توان بین روش ارائه شده و نرم افزار متلب



شکل ۱۴. اختلاف بین‌های مختلف روش ارائه شده و نرم افزار متلب

روش ارائه شده بهتر عمل کرده است مقایسه‌ای با FFT شرکت زایلینکس در جدول ۳ انجام شده است.

در جدول ۳ نشان داده شد که با استفاده از روش ارائه شده تقریباً ۶ برابر حجم حافظه و ۱/۵ برابر بلوک‌های DSP کاهش یافته است و از لحاظ زمانی نیز حدود ۶۰ هزار کلاک بهبود یافته است.

FFT پیاده‌سازی شده نیز در یک رادار تصویربرداری روزنه ترکیبی (با

سرهم تحویل خروجی می‌دهد ولی فاصله بین هر بسته ۶۵۵۳۶ تایی لازم است که ۸۱۹۲ کلاک فاصله باشد. در انتها ۸۱۹۲ داده‌ای که برای ادامه پردازش به آن نیاز است بعد از طی مدت زمانی مشخص تحویل داده می‌شود. در این طرح داده‌های ورودی آن دارای دقت ۱۶ بیت است و انجام محاسبات دارای دقت محدود نیست یعنی برای محاسبات ضرب و جمع، رقم یا بیتی دور ریخته نشده است. شکل ۱۲ این مطلب را نشان می‌دهد. فاصله بین هر دو بسته ۶۵۵۳۶ تایی با آمدن sync مشخص می‌شود.

۵. نتایج پیاده‌سازی و شبیه‌سازی

در ابتدا برای اثبات عملکرد روش ارائه شده، سیگنال‌هایی با فرکانس و توان‌های متفاوت به‌عنوان ورودی به الگوریتم ارائه‌شده و FFT نرم‌افزار متلب داده شده است و سپس خروجی آنها با یکدیگر مقایسه می‌شوند. به‌عنوان نمونه شکل ۱۳ در یک سناریو نشان می‌دهد که دامنه و فرکانس خروجی روش ارائه‌شده با دقت بسیار خوبی بر خروجی FFT نرم‌افزار متلب منطبق می‌باشد. در این سناریو سیگنال خروجی شامل ۵ مؤلفه فرکانسی با فرکانس‌های ۱.۵، ۴، ۱۱، ۱۷ و ۳۳ مگاهرتز می‌باشد که دارای توان نرمالیزه بر حسب db می‌باشد.

در شکل ۱۴ برای سناریوی پیشین اختلاف بین تمام بین‌های FFT برای روش ارائه‌شده و نرم‌افزار متلب نیز با یکدیگر مقایسه شده است. در شکل ۱۴ نیز اختلاف دو روش کمتر از ۰/۰۰۵ درصد می‌باشد.

حال برای این که مشخص شود که از لحاظ منابع مصرفی چقدر این

جدول ۳. مقایسه بین روش‌های پیاده‌سازی معماری ارائه‌شده و پیاده‌سازی بوسیله هسته FFT شرکت زایلینکس

روش‌ها	مدت زمان انجام محاسبات	مقدار حافظه ۱۸ کیلو بیتی	بلوک‌های DSP	fully used LUT-FF pair	occupied Slice	Slice LUT
روش پیاده‌سازی شده	۱۳۹۲۶۴ کلاک	۸۷	۱۲۸	۵۹۸۹	۵۹۹۹	۶۶۴۲
FFT ۶۴ کیلو نقطه‌ای شرکت زایلینکس ^۱	۱۹۶۷۷۵ کلاک	۴۹۸	۱۸۲	گزارش نشده	گزارش نشده	گزارش نشده

^۱ این مقادیر توسط نرم‌افزار شرکت زایلینکس تخمین زده است.

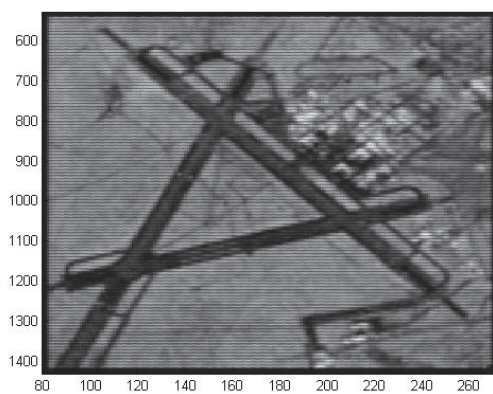
۶. مراجع

- [1] I. G. Cumming, F. H. Wong, U. o. B. Columbia, and M. Dettwiler, "Digital signal processing of synthetic aperture radar data: algorithms and implementation," Artech House Norwood, 2004.
- [2] Z. Fenghui, W. Yanfei, and S. Hongmei, "A new real time Range-Doppler imaging algorithm," in 2010 3rd International Symposium on Systems and Control in Aeronautics and Astronautics, 2010, pp. 119-122.
- [3] I. LogiCORE, "fast fourier transform v8. 0," ed, 2012. (2010-09-21).[2011-06-20]. http://china.xilinx.com/support/documentation/ip_documentation/ds808_xfft.pdf (2012).
- [4] M. Jahangir, D. Coe, A. P. Blake, P. G. Kealey, and C. P. Moate, "PodSAR: A versatile real-time SAR GMTI surveillance and targeting system," in Radar Conference, 2008. RADAR '08. IEEE, 2008, pp. 1-6.
- [5] M. Jahangir, D. Coe, A. P. Blake, P. G. Kealey, and C. P. Moate, "PodSAR: A versatile real-time SAR GMTI surveillance and targeting system," in Radar Conference, 2008. RADAR '08. IEEE, pp. 1-6, 2008.
- [6] Y. Jung, H. Yoon, and J. Kim, "New efficient FFT algorithm and pipeline implementation results for OFDM/DMT applications," Consumer Electronics, IEEE Transactions on, Vol. 49, pp. 14-20, 2003.
- [7] C. Cheng and K. K. Parhi, "High-throughput VLSI architecture for FFT computation," Circuits and Systems II: Express Briefs, IEEE Transactions on, Vol. 54, pp. 863-867, 2007.
- [8] M. Shin and H. Lee, "A high-speed four-parallel radix-2 4 FFT/IFFT processor for UWB applications," in Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on, pp. 960-963, 2008.
- [9] O. Jung-Yeol and L. Myoung-Seob, "New radix-2 to the 4th power pipeline FFT processor," IEICE transactions on electronics, Vol. 88, pp. 1740-1746, 2005.
- [10] S. He and M. Torkelson, "Designing pipeline FFT processor for OFDM (de) modulation," in Signals, Systems, and Electronics, 1998. ISSSE 98. 1998 URSI International Symposium on, pp. 257-262, 1998.
- [11] D. Rodriguez, "Tensor product algebra as a tool for VLSI implementation of the discrete Fourier transform," in Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference on, 1991, pp. 1025-1028, 1991.
- [12] P. A. Milder, F. Franchetti, J. C. Hoe, and M. Püschel, "Formal datapath representation and manipulation for implementing DSP transforms," in Proceedings of the 45th annual Design Automation Conference, pp. 385-390, 2008.
- [13] L. R. Rabiner and B. Gold, "Theory and application of digital signal processing," Englewood Cliffs, NJ, Prentice-Hall, Inc., vol. 1, p. 777, 1975.

الگوریتم برد-داپلر) مورد استفاده قرار گرفته و تصویر حاصله از آن به صورت شکل ۱۵ می شود. داده خام تصویر شکل ۱۵ (قسمت الف) به الگوریتم برد-داپلر داده شده و در نهایت خروجی تصویر به صورت شکل ۱۵ (قسمت ب) به دست آمده است.



الف) تصویر خام



ب) تصویر حاصله در الگوریتم برد-داپلر

شکل ۱۵. اختلاف بین‌های مختلف روش ارائه شده و نرم‌افزار متلب

۵. نتیجه‌گیری

در این مقاله بحث پیاده‌سازی الگوریتم FFT در راستای سمت در رادار SAR مد نظر قرار داده شد. در این الگوریتم معمولاً نیاز به پیاده‌سازی FFT با طول بسیار بلند می‌شود که به روش مستقیم پیاده‌سازی آن بر سخت افزارهای معمول امکان پذیر نیست.

در این مقاله با بهره‌گیری از فرم ریاضی مسئله و با ساده‌سازی فرمول‌ها و حذف عوامل اضافی روشی برای پیاده‌سازی الگوریتم FFT پیشنهاد شد که منابع محاسباتی و حجم حافظه را به شدت کاهش داده و پیاده‌سازی آن در سخت‌افزارهای معمول را ممکن می‌سازد. همچنین امکان پیاده‌سازی بقیه بلوک‌های پردازشی تصویربرداری رادار روزنه ترکیبی را در سخت‌افزار مذکور ممکن می‌کند.

- [14] E. H. Wold and A. M. Despain, "Pipeline and parallel-pipeline FFT processors for VLSI implementations," Computers, IEEE Transactions on, Vol. 100, pp. 414-426, 1984.
- [15] S. He and M. Torkelson, "A new approach to pipeline FFT processor," in Parallel Processing Symposium, 1996., Proceedings of IPPS'96, The 10th International, 1996, pp. 766-770.
- [16] S. Sukhsawas and K. Benkrid, "A high-level implementation of a high performance pipeline FFT on Virtex-E FPGAs," in VLSI, Proceedings.IEEE Computer society Annual Symposium on, pp. 229-232, 2004.
- [17] K. V. S. Reddy and K. Bala, "Implementation of 64-point FFT/IFFT by using radix-8 algorithm," 2013 IJEETC. Vol. 2, No. 4, October 2013
- [18] X. Sun, D. Qiu, H. Chen, and D. Chen, "An Implementation of FFT Processor," IET International Radar Conference 2013,p. 0479, 2013.

Introducing Azimuth FFT Architecture in Range-Doppler Imaging Algorithm in SAR Systems

S. R. Mohseni*, M. M. Nayebi, R. Mohseni, B. Ebrahimi

*Sharif University of Technology

(Received: 06/10/2014, Accepted: 24/05/2015)

Abstract

An important section of range-Doppler imaging algorithm in SAR systems which needs large number of resources, memories and solution time, is FFT on the azimuth dimension. In this paper, by decomposing the FFT relation, and converting to smaller dimension FFTs, implementation has been simplified. The proposed method, based on removing the extra phase factors reduces the resources, memories and solution time in hardware. Evaluation of the proposed method show that optimum architecture depends on image resolution, next processing stage and the number of FFT points. Using the proposed architecture leads to implement the whole range-Doppler imaging algorithm in smaller hardware.

Keywords: Synthetic Aperture Radar, Azimuth, Fast Fourier Transform, Implementation, Single-path Delay Feedback.