

طراحی یک تسریع کننده با قابلیت پیکربندی مجدد

عبدالرضا سپیانی^۱، احسان اله کبیر^{۲*}، فرید به آذین^۳

۱- دانش آموخته کارشناسی ارشد مهندسی الکترونیک، دانشگاه تربیت مدرس- پژوهشگر مجتمع تحقیقاتی صنعتی یامهدی (عج)

۲- استاد بخش مهندسی برق، دانشکده فنی و مهندسی، دانشگاه تربیت مدرس

۳- استادیار پژوهشکده سیستمهای دفاعی، مجتمع تحقیقاتی صنعتی یامهدی (عج)

* تهران، صندوق پستی ۱۴۳-۱۴۱۱۵

kabir@modares.ac.ir

(دریافت مقاله: مرداد ۱۳۸۲، پذیرش مقاله: تیر ۱۳۸۴)

چکیده- اجرای بسیاری از الگوریتمهای پردازش تصاویر ویدئویی به صورت بی درنگ، به دلیل حجم زیاد محاسبات، خارج از توان کامپیوترهای معمولی است. برای این منظور از سخت افزارهای جانبی استفاده می شود که برای الگوریتمهای خاصی طراحی شده یا قابلیت پیکربندی مجدد را برای اجرای الگوریتمهای مختلف دارند. در این مقاله، گزارش طراحی یک برد تسریع کننده مبتنی بر گذرگاه PCI ارائه می شود که عناصر پردازشی آن را FPGAها تشکیل می دهند. برای ارزیابی عملکرد این تسریع کننده، یک الگوریتم نگاشت تصویر پیاده سازی و کارکرد آن بررسی شده است. این تسریع کننده، نگاشت های درجه یک و دو را برای تصویر ۵۱۲x۵۱۲، در مدت زمان ۷/۹ms انجام می دهد.

کلید واژگان: تسریع کننده، نگاشت تصویر، FPGA، پیاده سازی بی درنگ.

۱- مقدمه

سخت افزار، گستره وسیعی از الگوریتمها را پوشش دهد [۲]. در بسیاری از الگوریتمهای پردازش تصویر و سیگنال به محاسبات نقطه شناور نیاز نیست و در عوض لازم است محاسبات ساده ای با سرعت زیاد انجام شود. در چنین شرایطی FPGA انتخاب مناسبی برای پیاده سازی الگوریتم است [۳]. در کامپیوترها یا پردازشگرهای سیگنال، از تسریع کننده ها برای سرعت دادن به اجرای الگوریتمها استفاده می شود. معمولاً برای هر الگوریتم، برنامه مخصوصی برای FPGAهای تسریع کننده نوشته می شود که به صورت توابعی خاص در

امروزه FPGAها در پردازش دیجیتال سیگنالها و تصاویر کاربردهای فراوانی یافته اند. با این عناصر می توان به سطح بالایی از پردازش موازی و پردازش خط لوله ای دست یافت [۱]. بسیاری از تسریع کننده ها^۱ در ساختمان خود از این عناصر بهره می گیرند. تسریع کننده می تواند برای هر الگوریتم، پیکربندی خاصی را به خود بگیرد و ضمن صرفه جویی زیاد در میزان گیت و حجم

1. Field Programmable Gate Array
2. Accelerator

۲- ساختار تسریع کننده

شکل ۱ ساختار تسریع کننده را نشان می دهد. این تسریع کننده امکانات زیر را دارد:

- ارتباط با کامپیوتر میزبان از طریق باس PCI و انتقال اطلاعات با نرخ ۵۰ مگا بایت در ثانیه.
- تبادل اطلاعات با حافظه کامپیوتر به صورت DMA^۴
- یک درگاه کم سرعت برای ارتباط با کامپیوتر و ارتباط نرم افزار با سخت افزار. نحوه ارتباط این درگاه با پردازنده، قابل برنامه ریزی است.
- ۴ مگابایت حافظه دستیابی تصادفی سنکرون^۵ به صورت ۸ حافظه ۲۵۶ کیلوی ۸ بیتی و ۴ حافظه ۲۵۶ کیلوی ۱۶ بیتی. حافظه ها از نوع خطلوله ای همزمان و سرعت بالا هستند.
- دسترسی همزمان به تمام حافظه ها.
- ارتباط باس PCI با تمام حافظه ها.
- دو FPGA شرکت زایلینکس با شماره XC4020XLA-PQ240-7 که سرعت مناسبی داشته و امکان پیاده سازی بسیاری از الگوریتم های پردازش سیگنال و تصویر را فراهم می سازند.
- برنامه ریزی FPGA ها از طریق باس PCI بدون نیاز به سخت افزار مخصوص برنامه ریزی.

۲-۱ ارتباطات داخلی تسریع کننده

همانطور که در شکل ۱ دیده می شود چهار بانک حافظه برای تسریع کننده در نظر گرفته شده است. بانکهای ۱ و ۲، هر یک از چهار حافظه ۸ بیتی تشکیل شده و گذرگاه های آدرس و داده مستقل دارند. بانکهای ۳ و ۴ نیز هر یک از دو حافظه ۱۶ بیتی تشکیل شده که گذرگاه آدرس آنها مشترک اما گذرگاه داده آنها مستقل است. این مسئله باعث شده که به I/O های فراوانی نیاز باشد، لذا از دو FPGA استفاده شده است.

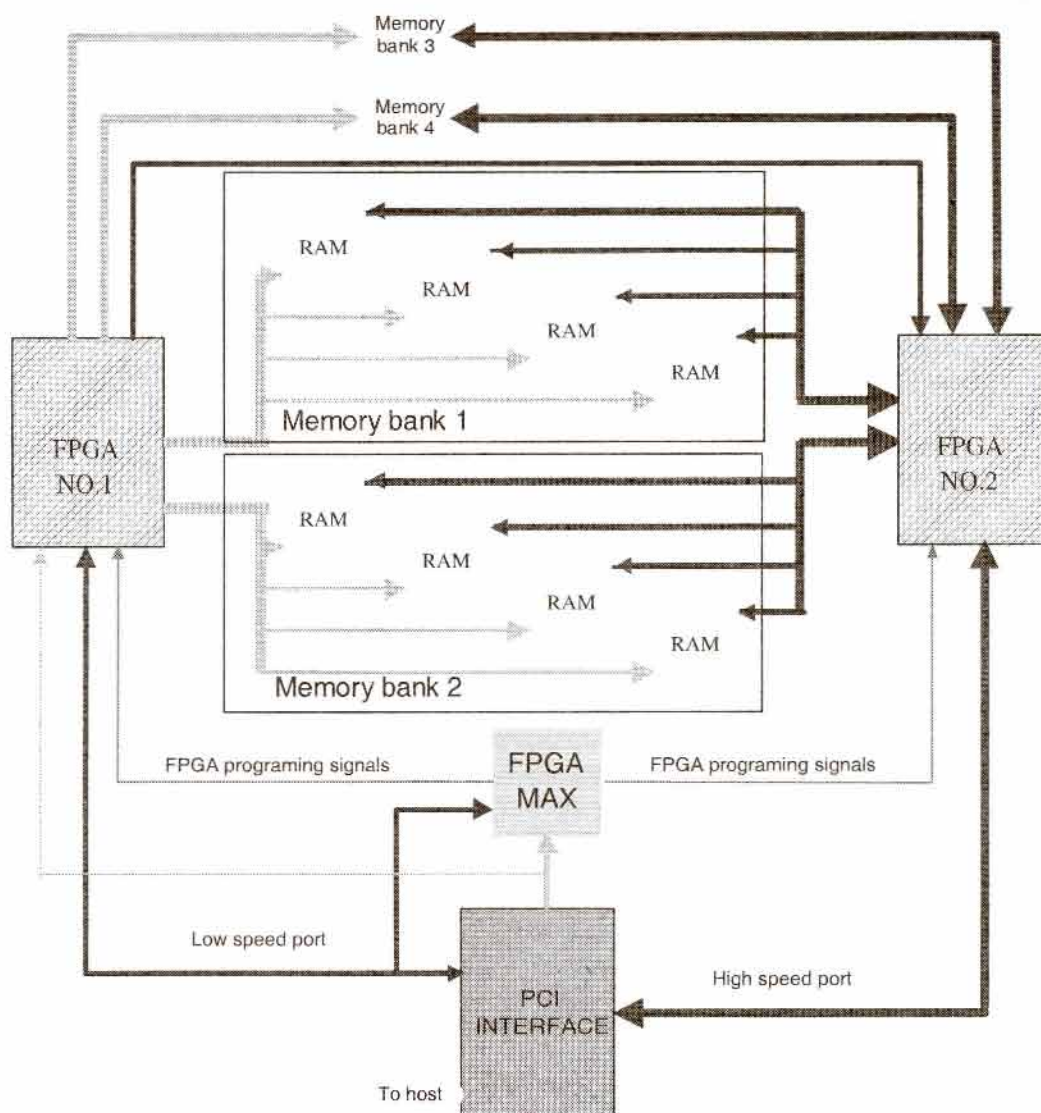
کتابخانه نرم افزار قرار گرفته و هرگاه کاربر نیاز به اجرای یکی از این الگوریتم ها داشته باشد، کافی است تابع مورد نظر را صدا بزند. در این حالت ابتدا تسریع کننده پیکربندی مخصوص آن الگوریتم را به خود می گیرد، سپس الگوریتم را اجرا می کند. از دید کاربر، الگوریتم به شکل نرم افزاری انجام شده است در حالی که این کار در تسریع کننده به صورت سخت افزاری انجام می شود [۴]. در مرجع [۵] تعدادی از سخت افزارهایی که قابلیت پیکربندی مجدد را دارند، معرفی شده است. برخی از این سخت افزارها به عنوان تسریع کننده، برخی به عنوان پردازنده کمکی یا هم پردازنده^۱ و برخی نیز به صورت مستقل استفاده می شوند.

امکان تغییر پیکربندی در زمان کوتاه به کمک کامپیوتر، مبادله سریع حجم زیادی از اطلاعات با حافظه کامپیوتر، امکان دسترسی سریع و همزمان به تعداد زیادی از خانه های حافظه محلی و قابلیت پردازش بالا، مسائلی است که طراحی این سخت افزارها را مشکل می سازد. در پیاده سازی های گزارش شده اغلب از FPGA های بزرگ، حافظه های دو درگاه و بردهای متعدد استفاده شده است. در طراحی انجام شده در این تحقیق از المانهای تجاری و ارزان قیمت استفاده شده و تسریع کننده ای با قابلیت بالا و قابل نصب در کامپیوتر ارائه شده است.

این مقاله به طراحی تسریع کننده ای با استفاده از FPGA های شرکت زایلینکس^۲ می پردازد. این تسریع کننده از طریق گذرگاه PCI^۳ با کامپیوتر میزبان در ارتباط بوده و برای پردازش دنباله تصاویر ویدئویی مناسب است. در بخش دوم، ساختار این تسریع کننده معرفی می شود. در بخش سوم پیاده سازی یک نمونه الگوریتم نگاشت تصویر بر روی این تسریع کننده شرح داده شده و نتایج به دست آمده با کارهای دیگران مقایسه می شود. در بخش چهارم جمع بندی مطالب ارائه می شود.

4. Direct Memory Access
5. SRAM

1. Coprocessor
2. Xilinx
3. Peripheral Component Interconnect/Interface



شکل ۱. ساختار کلی تسريع کننده طراحی شده

سیگنالهای متعددی برای ارتباط دو FPGA با یکدیگر در نظر گرفته شده است. این سیگنالها برای جابه جایی اطلاعات و فرمانهای کنترلی و همزمان سازی دو FPGA استفاده می شوند.

۲-۲- ارتباط با کامپیوتر میزبان

حجم داده ها در پردازش تصویر زیاد است، بنابراین نحوه ارتباط با کامپیوتر اهمیت بسیار زیادی دارد. درگاه های معمولی کامپیوتر مانند درگاه موازی و سریال توانایی

FPGA1 همواره وظیفه تولید آدرسها را بر عهده دارد و FPGA2 عملیات داده ها را انجام می دهد. استقلال گذرگاهها این مزیت را دارد که دسترسی همزمان و آزاد به تعداد بیشتری از پیکسل های تصویر میسر می شود. این مسأله در بعضی از الگوریتم های پردازش تصویر حیاتی است.

از بانکهای مختلف حافظه می توان به صورتهای مختلفی مانند بافر ورودی، بافر خروجی، بافر میانی یا جدول جستجو استفاده کرد.

برنامه‌ریزی این تراشه‌ها از نرم‌افزار فاندیشن سری ۲/۱^۴ استفاده شده است. کتابخانه این نرم‌افزار بسیاری از عناصری را که در پردازش تصویر و سیگنال استفاده می‌شوند، در بر دارد. برای نمونه می‌توان به جمع و تفریق‌کننده، ضرب‌کننده موازی، ضرب‌کننده در عدد ثابت، انباشتگر، تقسیم‌کننده، فیلتر FIR^۵، مکمل‌گیر، انتگرال‌گیر، جذرگیر، جدول جستجو برای سینوس و کسینوس، RAM^۶، ROM^۷ و FIFO^۸ اشاره کرد [۷].

۲-۴- توزیع پالس ساعت

از پالس ساعت گذرگاه PCI به‌عنوان منبع پالس ساعت استفاده شده است. فرکانس این منبع ۳۳ مگاهرتز است و کیفیت خوبی دارد. پالس ساعت بخشهای مختلف از طریق FPGA2 توزیع شده است. این کار باعث شده است که فرکانس بخشهای مختلف و زمانبندی آنها کاملاً برنامه‌ریزی شده و قابل تغییر باشد.

برای FPGAها ۵ ورودی پالس ساعت در نظر گرفته شده تا طراح بتواند به‌راحتی از چند پالس ساعت در محلهای مختلف FPGA استفاده کند و انحراف پالس ساعت^۹ را به حداقل برساند.

۳- مثال کاربردی: نگاشت تصویر^{۱۰}

نگاشت تصویر کاربردی است که به دلیل نامرتب بودن جریان داده و محاسبات فراوان، غالباً به عنوان پردازشی پیچیده در نظر گرفته می‌شود [۸-۱۰]؛ لذا نتایج پیاده‌سازی این الگوریتم، معیار مناسبی برای ارزیابی عملکرد تسریع‌کننده است.

عملیات نگاشت تصویر، نقاط واقع در تصویر مبدأ را به مختصات متفاوتی در تصویر مقصد منتقل می‌سازد

جابه‌جایی حجم انبوهی از اطلاعات را به‌صورت بی‌درنگ ندارند. به این دلیل از گذرگاه PCI برای انتقال اطلاعات استفاده شده است. به این منظور از کنترلگر گذرگاه PCI به شماره S5933 استفاده شده است. این کنترلگر قادر است گذرگاه PCI را به صورت فرمانده^۱، در اختیار گرفته و اطلاعات را به صورت DMA به حافظه کامپیوتر منتقل سازد [۶]. آدرس اولین داده در حافظه کامپیوتر و تعداد داده‌هایی که باید منتقل شوند، با تنظیم ثباتهای داخلی کنترلگر مشخص می‌شود. پس از این مرحله، کنترلگر به صورت خودکار داده‌ها را منتقل می‌کند. انتقال داده‌ها به شکل DMA این مزیت را دارد که هیچ زمانی از پروسسور صرف جابه‌جایی داده‌ها نمی‌شود. گذرگاه PCI داده‌ها را با حداکثر فرکانس ۳۳ مگاهرتز منتقل می‌کند. با توجه به این که این گذرگاه ۳۲ بیتی است، حداکثر نرخ جابه‌جایی اطلاعات برابر ۱۳۲ مگا بایت در ثانیه است. البته در عمل به دلیل اینکه بخشهای دیگری نیز از این گذرگاه استفاده می‌کنند، نرخ جابه‌جایی اطلاعات کمتر از این حد است [۴].

کنترلگر S5933 علاوه بر گذرگاه ۳۲ بیتی با سرعت زیاد، یک گذرگاه ۸ بیتی کم سرعت نیز دارد. از این گذرگاه برای برنامه‌ریزی FPGAها، انتقال اطلاعات کم حجم و کنترل سخت‌افزار از طریق نرم‌افزار استفاده می‌شود. از یک FPGA شرکت آلتر، که از نوع CPLD^۲ است، برای کدگذاری خطوط آدرس این گذرگاه و مالتی‌پلکس کردن خطوط داده‌های آن استفاده شده است.

۲-۳- عناصر پردازشی

در طراحی کارت تسریع‌کننده از دو FPGA شرکت زایلینکس به‌عنوان عناصر پردازشی استفاده شده است. هر یک از این FPGAها ۲۰۰۰۰ گیت داشته و سلولهای منطقی آنها از جدول جستجو^۳ تشکیل شده است. برای

4. Foundation Series 2.1i
5. Finite Impulse Response
6. Random Access Memory
7. Read Only Memory
8. First In First Out
9. Clock Skew
10. Image Warping

1. Master
2. Complex Programmable Logic Device
3. LUT

است. شکل ۲ دیاگرام بلوکی بخشهای مختلفی را که در FPGA1 پیاده‌سازی شده، نشان می‌دهد. همانطور که در این شکل دیده می‌شود، از معماری خط لوله‌ای استفاده شده است. بخش دریافت پارامترها، پارامترهای نگاشت و فرمانهای کنترلی را، از طریق درگاه کم سرعت واسط PCI، دریافت و در محل مناسبی ذخیره می‌کند. بخش تابع نگاشت، عبارت درجه اول یا دوم تابع نگاشت را به کمک انباشتگرها پیاده‌سازی می‌کند. تابع درجه ۲ با رابطه (۱) مشخص می‌شود. در این رابطه (x_s, y_s) مختصات محل نگاشت در تصویر مبدأ و (x_t, y_t) مختصات پیکسل تصویر مقصد است. در پیاده‌سازی این تابع از الگوریتم نشان داده شده در شکل ۳ استفاده شده است [۱۳].

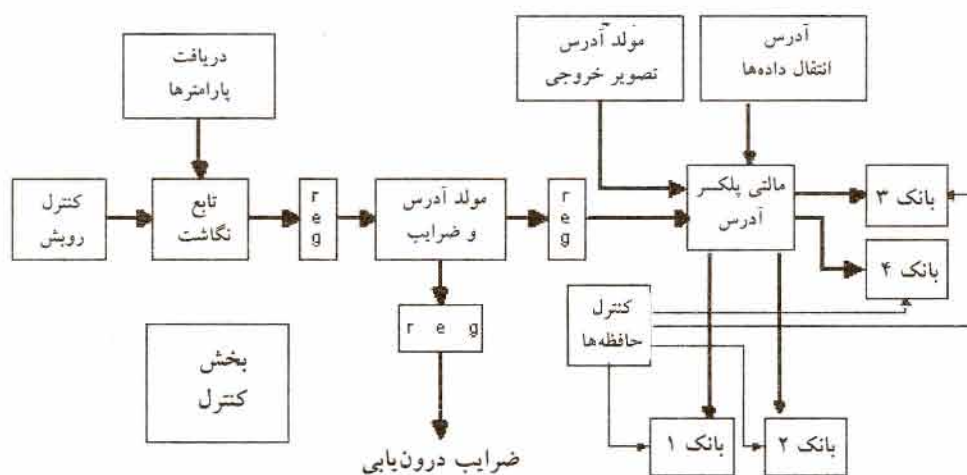
$$\begin{cases} x_s = m_4 x_t + m_5 y_t + m_6 + m_7 x_t^2 + m_8 y_t^2 \\ \quad + m_9 x_t y_t \\ y_s = m_{10} x_t + m_{11} y_t + m_{12} + m_{13} x_t^2 + m_{14} y_t^2 \\ \quad + m_{15} x_t y_t \end{cases} \quad (1)$$

این الگوریتم از توالی سطرها و ستونها برای محاسبه تابع درجه دو استفاده می‌کند. پیاده‌سازی این کد به پنج انباشتگر نیاز دارد. خروجی نهایی در انباشتگر f4 قرار می‌گیرد. خروجی بخش تابع نگاشت، مختصات هندسی یک نقطه در تصویر مبدأ است.

[۱۱]. برای تصاویری با ابعاد بزرگ، حجم محاسبات بسیار زیاد است. به‌ویژه زمانی که لازم است پردازش با دقت زیاد انجام شود. در این حالت علاوه بر محاسبه تابع نگاشت، نوعی درونیابی نیز باید انجام شود که به زمان زیادی نیاز دارد. در پیاده‌سازی سخت‌افزاری، درونیابی، منابع سخت‌افزاری زیادی را به خود اختصاص می‌دهد. درونیابی به دسترسی همزمان به چند پیکسل تصویر نیاز دارد که این مسأله خود محدودیتی را به حافظه تصویر و نوع ذخیره اطلاعات و فراخوانی آنها تحمیل می‌کند.

۳-۱- پیاده‌سازی الگوریتم نگاشت

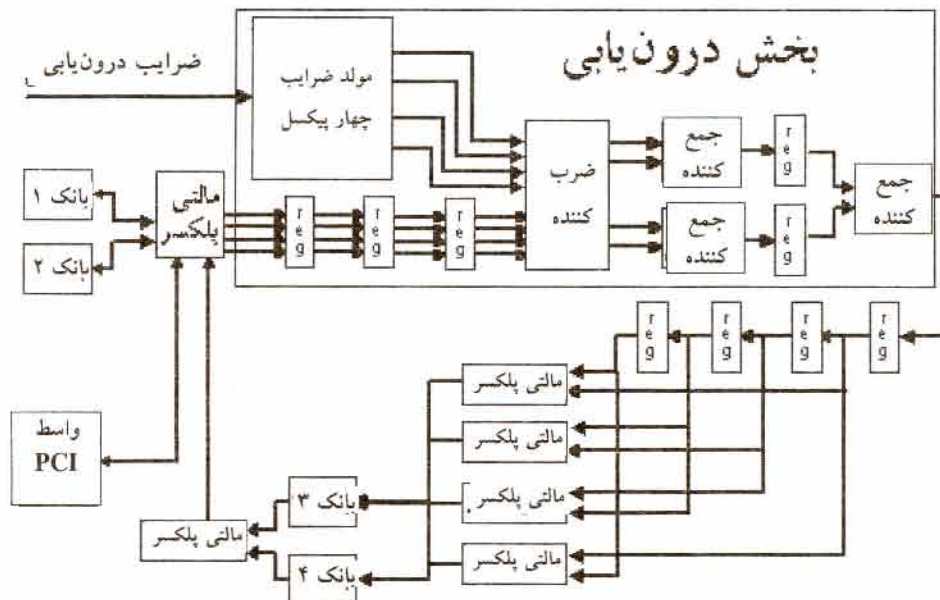
پیاده‌سازی نگاشت تصویر به سه بخش اصلی نیاز دارد: محاسبه تابع نگاشت، محاسبه آدرسها و ضرایب و درونیابی [۱۲]. در این تحقیق پیاده‌سازی نگاشتهای درجه یک و دو با درونیابی دوخطی، بر روی تسریع‌کننده مد نظر بوده است. همانطور که در توصیف ساختار تسریع‌کننده گفته شد، یکی از FPGAها همواره وظیفه تولید آدرسها را برعهده دارد و دیگری عملیات داده را انجام می‌دهد. در این کاربرد بخشهای محاسبه تابع نگاشت و محاسبه آدرسها و ضرایب در FPGA1 و بخش درونیابی و انتقال داده‌ها در FPGA2 پیاده‌سازی شده



شکل ۲. بلوک دیاگرام بخشهایی از الگوریتم نگاشت که در FPGA1 پیاده‌سازی شده است [۱۳].

کنترل
رویش

در بخش مولد آدرسها، با توجه به نوع ذخیره تصویر در حافظه‌ها، آدرسهای چهار پیکسل اطراف نقطه مذکور محاسبه می‌شود. تصویر در حافظه‌ها به گونه‌ای ذخیره شده که امکان فراخوانی چهار پیکسل مجاور وجود داشته باشد.



شکل ۴. بلوک دیاگرام بخشهایی از الگوریتم نگاشت که در FPGA2 پیاده‌سازی شده است [۱۳]

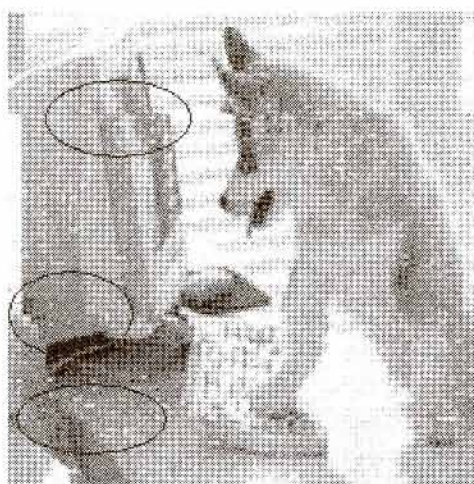
۳-۲- نتایج پیاده‌سازی الگوریتم نگاشت

کارت تسریع کننده می‌تواند نگاشتهای درجه یک و دو را با فرکانس ۳۳ مگاهرتز انجام دهد. نگاشت تصویر 512×512 ، $7/9 \text{ ms}$ طول می‌کشد. این کارت می‌تواند نگاشت تصویر رنگی با ابعاد 512×512 را در مدت زمان $23/1 \text{ ms}$ انجام دهد که برای پردازش بی‌درنگ کافی است.

نتایج اولیه پیاده‌سازی نشان داد که دو اشکال در کارکرد تسریع کننده وجود دارد: یکی ایجاد نویز در تصویر و دیگری شکست تصویر. نویز تصویر به دلیل انحراف پالس ساعت به وجود آمده بود و شکست تصویر به دلیل پدیده پرش زمین. پرش زمین موجب می‌شود که سطح منطقی بعضی از سیگنالهای خروجی از FPGA تغییر کند و از دید مدار بیرونی به غلط صفر یا یک تعبیر شود. وقتی تصویر از کامپیوتر به کارت تسریع کننده یا بعکس منتقل می‌شود، این پدیده موجب اضافه یا کم شدن چند داده به پیکسل‌های تصویر می‌شود. این باعث می‌شود که پیکسل‌های بعدی تصویر همگی شیفت مکانی پیدا کنند و لذا در تصویر شکستگی ایجاد شود. در شکل ۵-

ب نمونه‌ای از نتایج اولیه نشان داده شده است. در این تصویر نویز حاصل از انحراف پالس ساعت به وضوح دیده می‌شود.

شکستگی‌های تصویر نیز با علامت مشخص شده است. با همزمان‌سازی پالس ساعت بخشهای مختلف تسریع کننده، اثر انحراف پالس ساعت از بین رفت. این کار به وسیله اصلاحات مکرر و بررسی نتیجه، انجام شد تا پاسخ مطلوب به دست آمد. پرش زمین به چند عامل بستگی دارد که از آن جمله می‌توان به نرخ تغییر خروجی و تعداد خروجی‌هایی که با هم فعال می‌شوند، اشاره کرد. با کاهش نرخ تغییر خروجی‌ها و کاهش تعداد خروجی‌های فعال در هر لحظه، اثر پرش زمین نیز از بین رفت. شکل‌های ۶ و ۷ چند نمونه از نتایج اصلاح شده را نشان می‌دهند. اندازه تصاویر 240×240 پیکسل است و ۲۵۶ سطح خاکستری دارند. در شکل‌های ۸ دو نمونه از نتایج به دست آمده برای نگاشت درجه دوم نشان داده شده است.

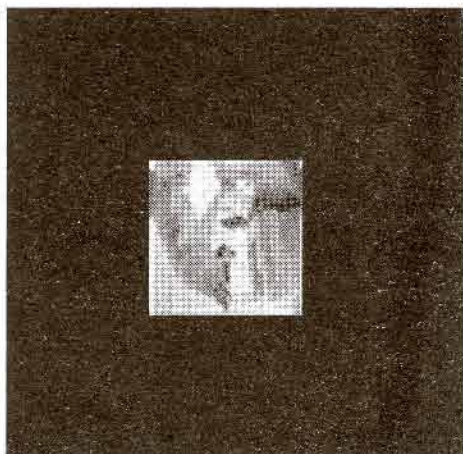


(ب)

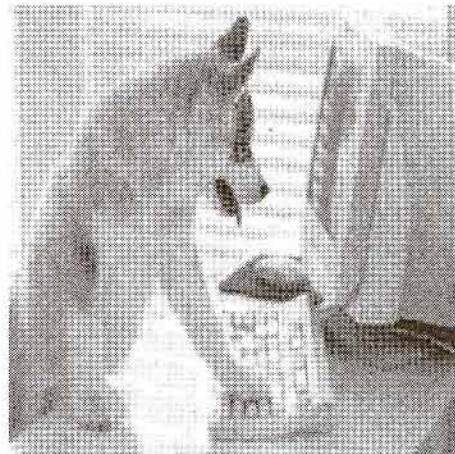


(الف)

شکل ۵ الف- تصویر سگ ب- نویز و شکستگی در تصویر نگاشت شده، پارامترهای نگاشت طوری تعیین شده است که تصویر ورودی و خروجی یکسان باشند تا مقایسه بهتر انجام شود

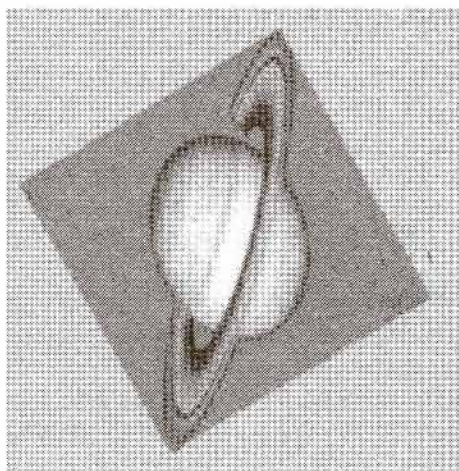


(ب)

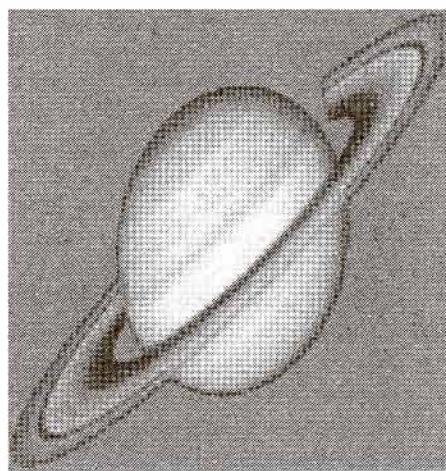


(الف)

شکل ۶ الف- انعکاس افقی شکل ۵- الف ب- دوران 180° درجه و تغییر مقیاس 0.33

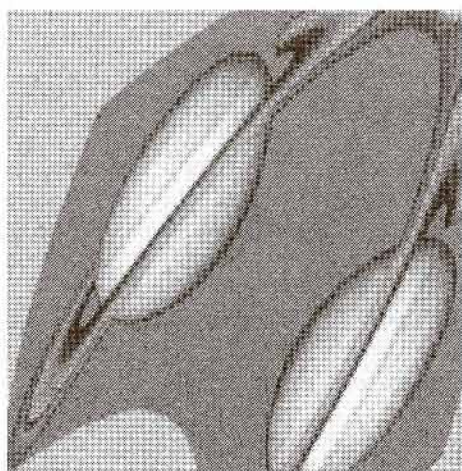


(ب)

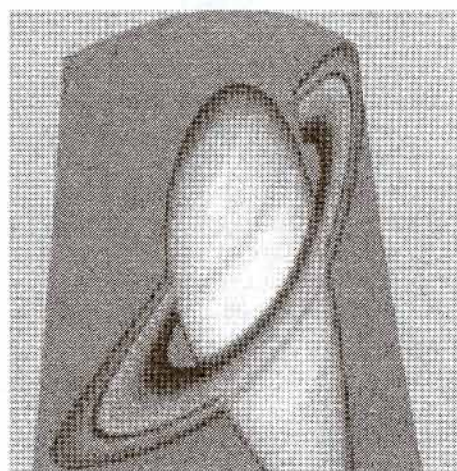


(الف)

شکل ۷ الف- تصویر کیوان ب- تصویر حاصل از دوران 30° درجه و تغییر مقیاس 0.66 و انتقال افقی ۸-



(ب)



(الف)

شکل ۸ دو نمونه از نگاشت درجه دوم تصویر کیوان

۳-۳- مقایسه نتایج

در زمینه پیاده‌سازی الگوریتم نگاشت، کارهای مختلفی انجام شده که در اینجا فقط شاخص‌ترین آنها آورده می‌شود. همه نتایج برای نگاشت تصویر 512×512 با 256 سطح خاکستری داده شده است.

یکی از نمونه‌ها سخت‌افزار شرکت سارنف است [۱۴]. این سخت‌افزار با استفاده از FPGAهای شرکت آلتر طراحی شده است. این سخت‌افزار نگاشت درجه یک و پرسپکتیو را با فرکانس ۳۳ مگاهرتز انجام می‌دهد. عیب اصلی این سیستم آن است که در آن از بیش از ۱۵۰ هزار دروازه منطقی استفاده شده است. این درحالی است که در پیاده‌سازی ارائه شده در این مقاله، کمتر از ۳۰ هزار دروازه منطقی به‌کار رفته است. علت اصلی این تفاوت را باید در روش پیاده‌سازی توابع ریاضی جستجو کرد. در این تحقیق با استفاده از الگوریتم ذکر شده در بخش ۳-۱ به‌جای ضرب‌کننده از انباشتگر استفاده شده و در نتیجه حجم پیاده‌سازی به‌شدت کاهش یافته است.

سخت‌افزار دیگری که با استفاده از FPGA ساخته شده است، می‌تواند نگاشت عمومی را با فرکانس ۱۲ مگاهرتز انجام دهد [۱۵].

شرکت دیتاکیوب سخت‌افزاری را با استفاده از تراشه‌های ASIC^۱ طراحی کرده که می‌تواند نگاشت درجه ۲ را در مدت زمان ۲۶ms انجام دهد. این سخت‌افزار در دو برد مجزا طراحی شده است [۱۶].

در [۱۷] پیاده‌سازی دیگری به کمک تراشه‌های ASIC ارائه شده که نگاشت درجه ۲ را در مدت زمان ۷/۹ms انجام می‌دهد. در فناوریهای مشابه معمولاً انتظار داریم که نتیجه طراحی به کمک تراشه‌های ASIC، ۵ تا ۱۰ برابر سریعتر از FPGA باشد [۱].

در طراحی به کمک سیگنال پروسسورها، DSP^۲، عمده‌ترین کارها در دانشگاه واشنگتن انجام شده است. در

یکی از این پیاده‌سازیها، نگاشت درجه ۱ در مدت زمان ۱۴/۱ms انجام شده است [۱۸]. در پیاده‌سازی دیگری، نگاشت عمومی در مدت زمان ۱۵/۱ms انجام شده است [۱۹]. در این پیاده‌سازیها برای فایق آمدن بر مشکل نامنظم بودن جریان داده، سعی شده از حافظه نهان DSP استفاده بهینه شود.

در جدول ۱ نتیجه پیاده‌سازی الگوریتم نگاشت تصویر به کمک تسریع‌کننده طراحی شده در این تحقیق، با پیاده‌سازیهای دیگر مقایسه شده است. زمانهای ذکر شده در این جدول، زمان نگاشت تصویر 512×512 با 256 سطح خاکستری است.

جدول ۱ مقایسه نتایج پیاده‌سازی الگوریتم نگاشت تصویر به کمک تسریع‌کننده این مقاله با پیاده‌سازیهای دیگر. زمانهای ذکر شده، زمان نگاشت تصویر 512×512 با 256 سطح خاکستری است.

نوع پیاده‌سازی	مرجع	نوع نگاشت	زمان (ms)
ASIC	[۱۳]	درجه ۲	۷/۹
ASIC	[۱۴]	درجه ۲	۲۶
TMS320C80	[۱۵]	آفین	۱۴/۱
TMS320C80	[۱۶]	عمومی	۱۵/۱
FPGA	[۱۱]	پرسپکتیو	۷/۹
FPGA	[۱۲]	عمومی	۲۱
FPGA	این مقاله	درجه یک و دو	۷/۹

۴- جمع‌بندی

در این مقاله طراحی یک تسریع‌کننده مبتنی بر گذرگاه PCI بیان شد. در ساختمان این تسریع‌کننده از FPGAهای شرکت زایلینکس به شماره XC4020XLA-PQ240-7 استفاده شده است. ساختار تسریع‌کننده طوری است که می‌تواند الگوریتم‌های پردازش دنباله تصاویر ویدئویی را به صورت بی‌درنگ انجام دهد. برای ارزیابی عملکرد این تسریع‌کننده،

1. Application Specific Integrated Circuit
2. Digital Signal Processor

1999.

[10] F. Planque, I. Kraljic and Y. Savaria, "Mapping Irregular Algorithms in a Custom Computing Image Processing Framework", *MAPLD2000*, 2000.

[11] Wolberg, G., *Digital Image warping*, IEEE Computer Society Press, 1993.

[۱۲] ع. سپیانی، ف. به آذین و ا. کبیر، "پیاده سازی بهینه الگوریتم نگاشت تصویر با استفاده از FPGA"، دومین کنفرانس علمی کاربردی سازمان صنایع هوافضا، دی ۱۳۸۲.

[۱۳] ع. سپیانی، "طراحی و پیاده سازی بخش نگاشت تصویر در یک تثبیت کننده الکترونیکی تصاویر"، پایان نامه کارشناسی ارشد الکترونیک، دانشکده فنی و مهندسی، دانشگاه تربیت مدرس، ۱۳۸۲.

[14] M. R. Piacentino, G. S. van der Wal and M. W. Hansen, "Reconfigurable Elements for a Video Pipeline Processor", *IEEE Symp. on Field-Programmable Custom Computing, FCCM99*, 1999.

[15] M. Dagtekin, S. Demarco, R. Ramanath and W. Snyder, "A High-speed Video Processing and Display System", *Proc. SPIE*, Vol. 3976, pp. 588-594, 2000.

[16] S. Siegel and B. Goetz-Greenwald, "VME Bords Perform High Speed Spatial Warping", *SPIE Proceedings*, 1027, 77-80, 1989.

[17] S. Lohmeyer, J. Burt and S. van der Wal "Quality Image Warper", U. S. Patent no. 6061447, 2000.

[18] O. D. Evans and Y. Kim, "Efficient Implementation of Image Warping on a Multimedia Processor", *Real-Time Imaging* 4, 417-428, 1998.

[19] P. Mattson and D. Kim and Y. Kim, "Generalized Image Warping Using Enhanced Lookup Tables", *Int. J. of Imaging Systems and Technology*, 9, pp. 475-483, 1998.

پیاده سازی الگوریتم نگاشت تصویر بر روی آن توضیح داده شد. نتایج عملی نشان می دهد که این تسریع کننده می تواند نگاشتهای درجه ۱ و ۲ تصویر 512×512 را در مدت زمان $7/9 \text{ ms}$ انجام دهد. نتایج به دست آمده با چند نمونه از کارهای دیگران مقایسه شد. نتایج به دست آمده، قابل مقایسه با سریعترین پیاده سازیهای گزارش شده است.

۵- منابع

[1] R. Tessier and W. Burleson, "Reconfigurable Computing for Digital Signal Processing: A Survey", *J. of VLSI Signal Processing*, Vol. 28, No. 1, pp. 7-27, 2000.

[2] M. Vasilko and D. Long, "Design of a Prototyping System for High-Speed Dynamically Reconfigurable Logic", *Proc. 8th Annual Advanced PLD & FPGA Conf. and Exhibition*, pp. 208-219, 1998.

[3] G. Knittel, "A PCI-Compatible FPGA-Coprocessor for 2D/3D Image Processing", *Proc. IEEE 1993 Symp. on FPGAs for Custom Computing Machines*, pp. 136-145, 1996.

[4] B. Bosi, G. Bosi and Y. Savaria, "Reconfigurable Pipelined 2-D Convolvers for Fast Digital Signal Processing", *IEEE Tran. VLSI Systems*, Vol. 7, No. 3, 1999.

[5] S. A. Guccione, "List of FPGA-Based Computing Machines", http://www.io.com/~guccione/HW_list.html.

[6] PCI Products Data Book, 1998. <http://www.amcc.com>

[7] The Programmable Logic Data Book, *Xilinx*, 1999.

[8] L. Zhang, B. Carter, C. Hsieh and A. Mckee, "Memory System Support for Image Processing", *IEEE Proc. 1999 Int. Conf. on Parallel Architectures and Compilation Techniques (PACT 99)*, pp. 98-107, 1999.

[9] S. Haynes, P. Cheung, W. Luk and J. Stone, "SONIC- A Plug-In Architecture for Video Processing", *Field-Programmable Logic and Application*, FPL 99, pp. 21-30,