دانشگاه صنعتی اصفهان

دانشکده برق و کامپیوتر

ارائه روش های پیشنهادی به منظور موازی سازی الگوریتم چینش VPR

پایان نامه کارشناسی ارشد مهندسی کامپیوتر- معماری

میلاد قربانی مقدم

استاد راهنما

دکتر کیارش بازرگان

1390

بسم الله
الرحمن الرحيم

دانشگاه صنعتی اصفهان

دانشکده برق و کامپیوتر

ارائه روش های پیشنهادی به منظور موازی سازی الگوریتم چینش VPR

پایان نامه کارشناسی ارشد مهندسی کامپیوتر- معماری

میلاد قربانی مقدم

استاد راهنما

دکتر کیارش بازرگان

1390

دانشگاه صنعتی اصفهان
دانشکده برق و کامپیوتر

پایان نامه کارشناسی ارشد رشتهٔ مهندسی کامپیوتر آقای میلاد قربانی مقدم

تحت عنوان

ارائه روش های پیشنهادی به منظور موازی سازی الگوریتم چینش VPR

در تاریخ 1390/3/29 توسط کمیته تخصصی زیر مورد بررسی و تصویب نهایی قرار گرفت.

1- استاد راهنمای پایان نامه — دکتر کیارش بازرگان

2- استاد داور پایان نامه — دکتر شادرخ سماوی

3- استاد داور پایان نامه — دکتر محمود اشرفی زاده

سرپرست تحصیلات تکمیلی دانشکده — دکتر سید محمود مدرس هاشمی

تشکر و قدردانی

با سپاس فراوان از:

پدر و **مادر** عزیزم که محبت‌های ایشان را هیچگاه نمی‌توانم جبران کنم.

استاد و راهنمای گرامی، **دکتر کیارش بازرگان،** که همواره با راهنمایی‌ها و نظرات ارزشمندشان در طول این تحقیق گره‌گشا بوده‌اند.

اساتید ارجمند، **دکتر شادرخ سماوی** و **دکتر محمود اشرفی زاده** که با دقت و ظرافت، بازخوانی و داوری این تز را برعهده گرفتند و از راهنمایی های ارزنده خود نسبت به اینجانب دریغ نکردند.

مدیر گروه کامپیوتر، **دکتر پژمان خدیوی** و سرپرست تحصیلات تکمیلی دانشکده، **دکتر محمود مدرس هاشمی** که در طول تحصیل اینجانب، کمال همکاری و مساعدت را داشتند.

مسئول تحصیلات تکمیلی دانشکده، سرکار خانم ها **نکویی** و **مظاهری** که با تجربه ارزنده خویش، همواره دلسوزانه و مسئولانه، اینجانب را راهنمایی کردند.

کلیه حقوق مادی مترتب بر نتایج مطالعات، ابتکارات و

نوآوریهای ناشی از تحقیق موضوع این پایان نامه متعلق به

دانشگاه صنعتی اصفهان است.

تقدیم به

پدر مادر عزیزم

آنان که راستی قامتم در شکستگی قامتشان تجلی یافت. در برابر وجود گرامیشان زانوی ادب بر زمین می نهم و با دلی مملو از عشق و محبت بر دست پر مهرشان بوسه می زنم.

برادران عزیزم

که همواره حامی و پشتیبان من بوده اند.

عنوان **صفحه**

فهرست مطالب .....هشت

چکیده 1

فصل اول: مقدمه

1-1- مقدمه 2

2-1- فرآیند طراحی FPGA 3

3-1- معرفی گام چینش 4

3-1-کاستی ها و روش های موجود 5

5-1- ساختار پایان نامه 8

فصل دوم: کارهای انجام شده

2-1- ساختار FPGA 10

2-1-1- تکنولوژی برنامه پذیری 11

2-1-2- معماری بلوک های منطق 12

2-1-3- معماری مسیریابی 14

2-2- چینش VPR استاندارد 17

2-2-1- الگوریتم چینش VPR سریال 18

2-3- موازی سازی چینش VPR 22

2-3-1 شتاب دادن به حرکات 23

2-3-2 جابجایی های موازی در سطح دانه ریز 24

2-3-3 جابجایی های موازی در سطح دانه درشت 30

فصل سوم: روش های پیشنهادی به منظور موازی سازی چینش VPR

3-1- مقدمه 33

3-2- روش حرکات موازی 34

3-3- روش محاسبات براساس تأمل و پیش بینی 41

3-4- روش میانگین مختصات بلوک ها 46

3-5- روش پارتیشن بندی چینش 48

فصل چهارم: ارزیابی نتایج

4-1- مقدمه 54

4-2- ارزیابی نتایج روش حرکات موازی 55

4-3- ارزیابی نتایج محاسبات براساس تأمل و پیش بینی 57

4-4- ارزیابی نتایج روش میانگین مختصات بلوک ها 61

4-5- ارزیابی نتایج روش پارتیشن بندی 63

فصل پنجم: نتیجه گیری

5-1- مقدمه 70

5-1-نتیجه گیری 71

5-1- پیشنهادات 72

مراجع 74

چکیده:

پیشرفت تکنولوژی، افزایش پیچیدگی در مدارات و نیز افزایش تعداد ترانزیستورهای بکار رفته در تراشه ها و به دنبال آن گسترش سایز FPGA ها را به دنبال داشته است. از طرفی فرآیند طراحی و پیاده سازی مدارات روی FPGA، فرآیندی زمانگیر بوده که با گسترش پیچیدگی مدار و سایز FPGA ها، به صورت نمایی افزایش می یابد. بر این اساس، از آنجایی برنامه های کاربردی که از FPGA استفاده می کنند نیاز به زمان ارائه به بازار محدودی دارند، یکی از مهمترین چالش های پیش روی طراحان، به خصوص در طراحی سیستم های با قابلیت پیکربندی مجدد به طورپویا، کاستن این زمان طراحی، بدون کاهش در کیفیت نهایی راه حل مورد نظر می باشد. یکی از گام های اصلی و بحرانی در فرآیند طراحی FPGA به کمک کامپیوتر (CAD)، گام چینش است که وظیفه مکان دهی بلوک های منطقی را با هدف کاهش طول سیم بکار رفته و تأخیر زمانی مدار، برعهده دارد. در اکثر ابزارهای طراحی، از میان روش هایی که برای چینش ابداع شده، روش های مبتنی بر شبیه سازی گداختگی فلزات که بر اساس سرد شدن تدریجی فلزات بنا نهاده شده، به خاطر ارائه نتایج و راه حل های با کیفیت بالا و قدرت عملیاتی مناسب، به صورت گسترده ای مورد توجه قرار گرفته است که در آن میلیون ها حرکت تصادفی، سعی در تغییر مکان بلوک های منطقی به منظور بهبود پارامترهای کیفی دارند.

در سال های اخیر موازی سازی به دلیل تمایل به افزایش نمایی تعداد هسته ها، می تواند راه حل تضمین شده ای برای کنترل افت سرعت طراحی با افزایش نمایی زمان محاسباتی با گسترش سایز FPGA ها محسوب شود. یکی از امکانات موازی سازی که اخیرا رایج شده و مورد استقبال طراحان قرار گرفته، استفاده از قدرت محاسباتی واحدهای پردازشی گرافیکی همه منظوره است که به خاطر داشتن سخت افزار خاص و تعبیه صدها هسته پردازشی بر اساس معماری اتصالی و ارتباطی مناسب، راه حل موازی سازی تضمین شده ای را برای بهبود زمان اجرایی کاربردهای مورد نظر ارائه می کند.

در این پایان نامه، VPR که یکی از محبوب ترین ابزارهای چینش و مسیریابی در طراحی FPGA با کمک کامپیوتر محسوب می شود، بحث شده و الگوریتم و کد در دسترس آن به خصوص برای گام چینش مورد مطالعه و ارزیابی قرار گرفته است. از آنجایی که الگوریتم چینش در VPR براساس روش گداختگی شبیه سازی شده عمل می کند و ذات و عملکردی سریال گونه دارد، بدین صورت که عملیات بعدی منوط به استفاده از نتایج عملیات فعلی می باشد، انتخاب روش مناسب برای موازی سازی آن با استفاده از تعداد زیادی ترد، کاری دشوار می باشد و ممکن است راه حل های بدست آمده از لحاظ کیفیت، بسیار دورتر از راه حل نهایی حاصل از اجرای سریال چینش از طریق الگوریتم بکار گرفته شده توسط VPR باشد. بنابراین در این پایان نامه سعی شده با در نظر قرار دادن معماری واحدهای پردازشی گرافیکی همه منظوره، به ارائه روشی مناسب برای غلبه بر ذات سریال گونه چینش VPR و استفاده همزمان از تعداد زیادی ترد برای دستیافتن به افزایش سرعتی درخور، بدون کاهش معنی دار در کیفیت نهایی راه حل پرداخته شود. بدین منظور چهار روش جهت پیاده سازی با استفاده از تنوعی از تعداد تردها مورد تحلیل و شبیه سازی قرار گرفتند. روش حرکات موازی در صورت پیاده سازی مناسب می تواند به ازای تعداد تردهای محدود، افزایش سرعتی تا حدی مناسب را نتیجه دهد. این در حالی که در روش ارائه شده دوم یعنی روش مبتنی بر محاسبات براساس تأمل و پیش بینی دستیابی به افزایش سرعتی محدود بدست آمد. روش میانگین مختصات بلوک ها نهایتا به شکست انجامید و روش آخر که بر اساس پارتیشن بندی مسأله چینش استوار بود به ازای تعداد تردهای بالا افزایش سرعت محدودی را نتیجه داد.

کلمات کلیدی: موازی سازی، FPGA، مکان دهی، VPR، گداختگی شبیه سازی شده

فصل اول

مقدمه

1-1 مقدمه

FPGA[1] ها، تراشه های منطقی برنامه پذیری هستند که انقلابی در طراحی سیستم های دیجیتالی در دو دهه اخیر به وجود آورده اند. قابلیت برنامه پذیری، هزینه مناسب سخت افزاری و نیز زمان کوتاه مورد نیاز برای ارائه به بازار[2] موجب محبوبیت FPGA ها برای طراحان سیستم ها شده است. FPGA ها در حقیقت از مجموعه ای از بلوک های منطقی قابل پیکر بندی و منابع مسیریابی تشکیل شده اند که بر این اساس امکان برنامه پذیری روی آن ها ممکن می شود. قابلیت برنامه پذیری در FPGA ها از طریق فرآیند طراحی انجام می گیرد و مدار کاربرد مورد نظر که از طریق زبان های توصیف سخت افزار به صورت کد توصیف شده، به صورت بهینه روی بلوک های منطقی سطح FPGA چیده می شود و اتصالات میانی آن ها برقرار می گردد. یکی از مهمترین گام ها در فرآیند طراحی، گام چینش است که وظیفه چیدن این بلوک ها را با هدف کاهش طول سیم بکار رفته و کاهش تأخیر در مدار پیاده سازی شده، بر عهده دارد.

ظهور FPGA ها با چندین میلیون گیت و در اختیار بودن تنوع در ویژگی های سطح سیستمی در آنها، FPGA ها را به عنوان یک انتخاب طراحی مناسب برای طراحان سیستم های پیچیده، مطرح کرده است. این محبوبیت رو به

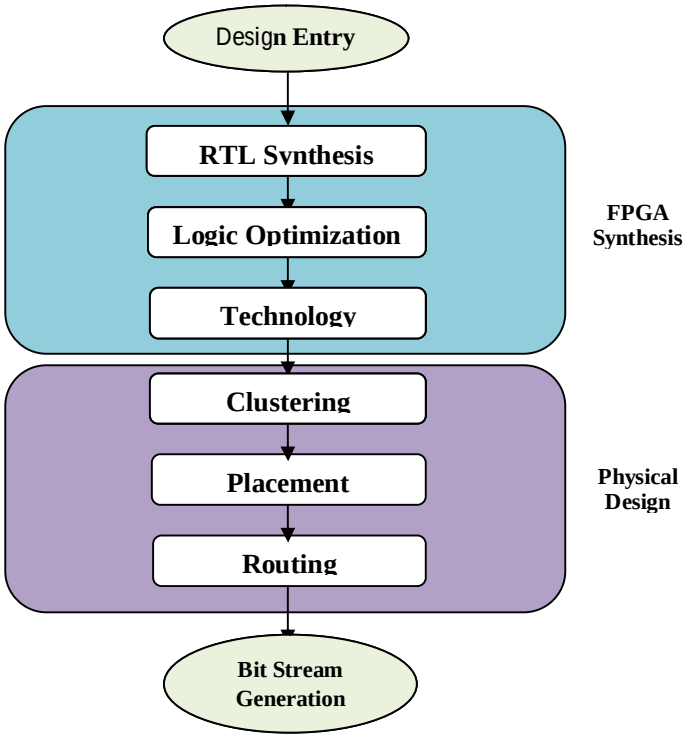
---

[1] Field Programmable Gate Array
[2] Time-to-market

افزایش، منجر به ایجاد علاقه بیشتر برای مطالعات الگوریتمیک و پیشرفت ابزارها برای رفع موانع و مشکلات طراحی و نیز بهینه سازی پیاده سازی ها شده است.

2-1 فرآیند طراحی FPGA

فرآیند طراحی FPGA [1]به کمک کامپیوتر[2]، مطابق شکل 1-1 قابل ملاحظه است. مشابه با [1]، پس از آنکه طرح و مدار مورد نظر توسط زبان های توصیف سخت افزار[3] مانند Verilog و VHDL آماده شد، این توصیف به همراه محدودیت های طراحی و نوع FPGA مورد نظر، به عنوان ورودی در اختیار فرآیند طراحی قرار می گیرد. در ابتدا توصیف سخت افزاری مورد نظر در گام سنتز سطح ثبات و ترانزیستور[4] از سطح ترانزیستور به سطح گیت، مانند جمع کننده ها، ضرب کننده ها, ثبات ها، مالتی پلکسرها، ماشین های وضعیت و بلوک های حافظه تبدیل می شوند. اکثر FPGA های مورد استفاده معمولا سخت افزاری بر اساس جدول درستی[5] دارند و در آن ها، عنصر منطقی پایه[6] شامل یک جدول درستی با m ورودی است که توانایی پیاده سازی هر تابعی را با m متغیر داراست. علاوه بر این جهت افزودن قابلیت پیاده سازی مدارات ترتیبی در هر عنصر منطقی پایه در FPGA، علاوه بر جدول



شکل 1-1: فرآیند طراحی FPGA، باکمک از کامپیوتر (مشابه [1])

[1] FPGA CAD Flow
[2] Computer Aided Design (CAD)
[3] Hardware Description Language (HDL)
[4] Register-Transistor Level Synthesis (RTL Synthesis)
[5] Look-Up Table (LUT)
[6] Basic Logic Element (BLE)

درستی یک فلیپ فلاپ هم در نظر گرفته شده است. حال با در نظر گرفتن معماری FPGA، در گام بعدی یعنی بهینه سازی منطقی[1]، برای عناصر ایجاد شده از مرحله قبل، جدول درستی و تابع دودویی متناظر محاسبه می گردد و در گام نگاشت بر اساس تکنولوژی[2]، بر اساس نوع FPGA مورد استفاده و تعداد ورودی های جدول درستی های بکار رفته در آن، جداول درستی عناصر سطح گیت مورد نظر به عناصر منطقی پایه، نگاشت می شوند. جهت بهبود تأخیر در مدار در حال پیاده سازی، در مرحله خوشه بندی[3]، چندین عنصر منطقی پایه که مجاور یکدیگر قرار می گیرند، به بلوکهای منطقی قابل پیکربندی[4]، بسته بندی می شوند. سپس در گام چینش[5]، موقعیت مکانی بلوک های های منطقی قابل پیکربندی برای دستیابی به حداقل طول سیم بکار رفته و نیز حداقل تأخیر، مشخص می شود و در مرحله مسیریابی[6] با توجه به معماری مسیریابی در FPGA مورد نظر، با تخصیص مناسب منابع مسیریابی، اتصالات بین بلوک های منطقی قابل پیکربندی برقرار می گردد.

3-1- معرفی گام چینش

همانگونه که ذکر شد در فرآیند طراحی FPGA به کمک کامپیوتر، گام های متفاوتی وجود دارد. در این میان گام چینش یکی از مراحل است که از بیشترین شدت عملیات محاسباتی برخوردار می باشد. از میان روش هایی که برای چینش ابداع شده، روش های مبتنی بر شبیه سازی گداختگی فلزات[7]، که بر اساس سرد شدن تدریجی فلزات بنا نهاده شده، به خاطر ارائه نتایج و راه حل های با کیفیت بالا و قدرت عملیاتی مناسب، به صورت گسترده ای برای گام چینش، در قلمرو VLSI و FPGA مورد توجه قرار گرفته است [2و3]. در اینگونه روش ها که در اکثر ابزارهای تحقیقاتی و نیز صنعتی بکار گرفته می شود، با هدف بهینه سازی و کاهش طول سیم و تأخیر مدار مورد نظر، بلوک های منطقی که هریک قسمتی از مدار را در بر دارند، روی سطح FPGA چیده می شوند.

به عنوان مثال پیاده سازی یک پردازنده را روی FPGA در نظر بگیریم. فرض شود که مدار یک پردازنده پس از پیمودن گام های طراحی به مجموعه ای از بلوک های منطقی قابل فهم توسط FPGA مورد نظر تقسیم شده است. در گام چینش این بلوک ها باید بگونه ای در داخل سطح تراشه چیده شوند که نهایتا در انتهای طراحی، در اتصالات مجموعه بلوک ها که با ارتباط با هم پردازنده مورد نظر را تشکیل می دهند، تا حد امکان کوتاهترین طول سیم مورد نظر به کار رفته شده باشد و از طرفی هم انتقال نتیجه حاصل از ورودی مورد نظر در پین های وروی FPGA، به پین های خروجی آن، از لحاظ زمانی، تأخیر کمتری داشته باشد.

فرآیند چینش بر اساس روش گداختگی شبیه سازی شده به طور مختصر مطابق زیر است:

---

١ Logic Optimization
٢ Technology Mapping
٣ Clustering
٤ Configurable Logic Block (CLB)
٥ Placement
٦ Routing
٧ Simulated Annealing (SA)

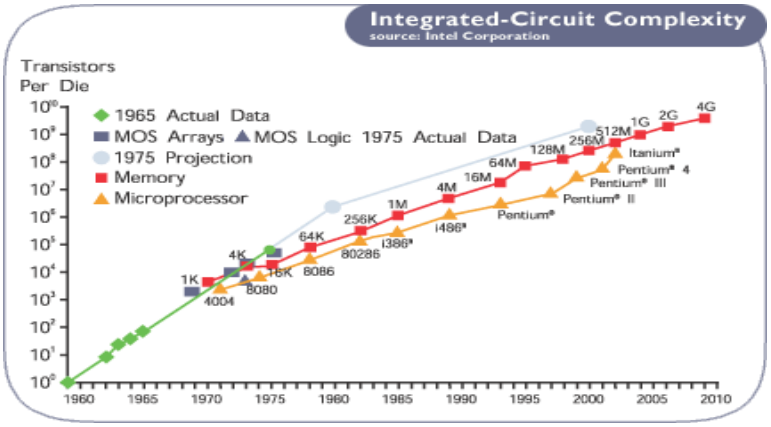
در ابتدا بلوک های منطقی به طور تصادفی روی سطح FPGA توزیع می شوند. این الگوریتم از دو حلقه اصلی تشکیل شده است. حلقه خارجی که پارامتری برای تعیین احتمال پذیرش حرکات نامناسب را تنظیم می کند و حلقه داخلی که بر اساس این پارامتر، اجرای مجموعه ای از عملیات را برعهده دارد. در هر اجرا از حلقه داخلی ابتدا پیشنهاد جابجایی دو بلوک یا انتقال یک بلوک به مکان خالی ارائه می شود؛ سپس با در نظر گرفتن پذیرش آن حرکت، تابع هزینه فراخوانی می شود. این تابع هزینه، چینش جدید را از لحاظ میزان طول سیم به کار رفته و نیز تأخیر مدار، مورد ارزیابی قرار می دهد. در صورتی که چینش جدید، چینش گذشته را بهبود داده باشد، حرکت پیشنهاد شده پذیرفته می شود. در غیر این صورت یعنی در حالتی که حرکت نامناسب تشخیص داده شود، این حرکت براساس احتمالی که به پارامتر احتمال پذیرش حرکات نامناسب در حلقه خارجی وابسته است، پذیرفته و یا رد می شود. درانتهای حلقه داخلی یعنی زمانی که مجموعه ای از حرکات ارزیابی شدند، درصد پذیرش حرکات محاسبه می شود و بر اساس نرخ پذیرش بدست آمده، حلقه خارجی احتمال پذیرش حرکات نامناسب را با تنظیم مجدد پارامتر مورد نظر، کاهش داده و به ازای پارامتر احتمال جدید، حلقه داخلی مجدداً تکرار می شود. این فرآیند تا جایی که تنها حرکات مناسب امکان پذیرش دارند، ادامه یافته و نهایتا چینش بهینه در انتهای فرآیند بدست می آید. دلیل پذیرش برخی حرکات نامناسب درطول فرآیند، عبور از حداکثرهای محلی است که نهایتا به بهبود چینش حاصل کمک می کند.

4-1 کاستی ها و روش های موجود

در دهه های اخیر تعداد ترانزیستورها در یک مدار مجتمع، براساس قانون مور[1] [4] مطابق شکل 1-2، حدوداً در هر دو سال، دو برابر شده است که در نتیجه آن سایز و قابلیت FPGA ها روز به روز در حال افزایش است. از آنجایی که فرآیند طراحی FPGA با کمک گرفتن از کامپیوتر، فرآیندی زمانگیر است و با افزایش سایز FPGA ها، این زمان به صورت نمایی افزایش می یابد، با توجه به محدودیت شدیدی که در زمان ارائه به بازار برای کاربردهای مورد نظر برای پیاده سازی روی FPGA وجود دارد، همواره یکی از چالش های پیش روی طراحان، به خصوص در طراحی سیستم های با قابلیت پیکربندی مجدد به صورت پویا[2]، کاستن این زمان طراحی بدون وقوع کاهش معنی داری در کیفیت نهایی راه حل، یعنی افزایش طول سیم استفاده شده و نیز افزایش تأخیر مدار پیاده سازی شده، می باشد. بنابراین هدفی که طراحان به دنبالش هستند، کاهش این زمان طراحی در حد و اندازه زمان کامپایل برنامه های نوشته شده به زبان هایی از جمله C، است.

---

۱ Moore’s law
۲ Dynamic Reconfigurable System



شکل 1-2: قانون مور براساس [5]

درگام چینش مشابه سایر گام های فرآیند طراحی FPGA به کمک کامپیوتر، فاصله زمانی زیادی بین کارهای محاسباتی مورد نیاز و قدرت پردازشی در دسترس وجود دارد. بنابراین نیاز فراوانی برای بهبود کارایی چینش های مبتنی بر گداختگی شبیه سازی شده احساس می شود[6و7]. موازی سازی در سال های اخیر به دلیل تمایل به افزایش نمایی تعداد هسته ها، می تواند راه حل تضمین شده ای برای کنترل افت سرعت کامپایل طرح با افزایش نمایی زمان محاسباتی با گسترش سایز FPGA ها محسوب شود.

باتوجه به [8] الگوریتم های موجود را می توان به الگوریتم های مبتنی بر روش های پارتیشن بندی بازگشتی[1] [9و10]، روش های تحلیلی[2] [11]، روش مبتنی بر الگوریتم ژنتیک[3][12]، و نیز روش های مبتنی بر شبیه سازی گداختگی و سردکردن تدریجی فلزات [2و3و13] طبقه بندی کرد. پیاده سازی های فعلی اکثراً منطبق بر اجراهای سریال بوده و همانطور که اشاره شد با بزرگ شدن سایز FPGA ها و پیچیده تر شدن مدارات، متأثر از پیشرفت تکنولوژی، و متعاقبا افزایش نمایی زمان طراحی، اینگونه پیاده سازی ها قادر نخواهند بود سرعت طراحی مطلوبی را باتوجه به محدودیت زمانی برای تحویل به بازار، ارائه دهند.

براساس [8] استفاده از پردازنده های موازی، نقطه عطفی در تاریخچه محاسبات بوده [14] و تلاش های فعلی نیز بر افزایش مداوم تعداد هسته ها در یک تراشه دلالت می کند [15-20]. با وجود این حقیقت که محاسبات موازی برای مدت زیادی است که مورد بحث قرار گرفته[21و22]، انتخاب بهترین و مناسب ترین روش موازی سازی برای انتقال کاربرد از حالت سریال به حالت موازی که مزیت موازی سازی را به حداکثر برساند، هنوز یکی از بزرگ ترین چالش ها می باشد. همزمانی[4]، اکنون یکی از گسترده ترین چالش ها در این زمینه بوده [23] و نیز وجود تعداد روش های موازی بیش از حد نیز [24] انتخاب روش مناسب را با سختی ممکن می سازد. یکی از روش های موازی سازی که اخیرا رایج شده و مورد استقبال طراحان قرار گرفته، استفاده از قدرت محاسباتی واحدهای پردازشی

---

[1] Recursive partitioning
[2] Analytical
[3] Genetic algorithm
[4] Synchronization

گرافیکی همه منظوره[1] [25] است که به خاطر داشتن سخت افزار خاص و تعبیه صدها هسته پردازشی بر اساس معماری اتصالی و ارتباطی مناسب، راه حل موازی سازی تضمین شده ای برای بهبود زمان اجرایی کاربردهای مورد نظر را ارائه می کند.

بر این اساس کارهایی نیز در جهت پیاده سازی موازی روش های موجود انجام گرفته، از این قبیل می توان به پیاده سازی های موازی روش مبتنی بر گداختگی شبیه سازی شده به دلیل کیفیت بالای نتایج اشاره کرد. این دسته از پیاده سازی های موازی معمولا در سه قالب انجام گرفتند. برخی سعی در تسریع الگوریتم با ایده شتاب دادن به حرکات دارند [26و6] و سعی می کنند گام ارزیابی پذیرش یا رد حرکت را توسط چندین پردازنده به طور موازی اجرا کنند که مقیاس پذیری بالایی ندارد و با محدودیت در افزایش سرعت مواجه بوده اند. دسته ای دیگر سعی در موازی سازی دانه ریز جابجایی ها دارند[27] [28] [29] [30] [32] [33]، مانند روش های مبتنی بر پارتیشن بندی و نیز روش های مبتنی بر ایده پیش بینی حرکات آینده و همچنین توزیع ارزیابی های حرکات بدون تداخل روی پردازنده ها که در کارهای گذشته نتوانستند افزایش سرعت مناسبی را با افزایش تعداد پردازنده های موازی گزارش کنند. دسته آخر هم مبتنی بر موازی سازی دانه درشت جابجایی ها شکل گرفته [34] [35] [27] و براساس اجرای مستقل هر پردازنده روی صورت مسأله اصلی عمل می کنند و نهایتا میان نتایج بدست آمده از پردازنده های متفاوت بر اساس نظر سنجی و همه پرسی چینش نهایی تشکیل می شود. این روش ها از افزایش سرعت مناسبی برخوردارند اما در سطوح بالای موازی سازی با افت شدید و غیر قابل چشم پوشی کیفیت مواجهند.

در این پایان نامه، VPR[2] که یکی از محبوب ترین ابزارهای چینش و مسیریابی در طراحی FPGA با کمک کامپیوتر محسوب می شود، بحث شده و الگوریتم و کد در دسترس آن به خصوص برای گام چینش مورد مطالعه و ارزیابی قرار گرفته است. از آنجایی که الگوریتم چینش در VPR براساس روش گداختگی شبیه سازی شده عمل می کند و ذات و عملکردی سریال گونه دارد، بدین صورت که عملیات بعدی منوط به استفاده از نتایج عملیات فعلی می باشد، انتخاب روش مناسب برای موازی سازی آن با استفاده از تعداد زیادی ترد، کاری دشوار بوده و ممکن است راه حل های بدست آمده از لحاظ کیفیت، بسیار دورتر از راه حل نهایی حاصل از اجرای سریال چینش از طریق الگوریتم VPR باشد. بنابراین در این پایان نامه سعی شده قابلیت های موازی سازی چینش VPR که طبیعتی سریالی دارد، مشخص شده و بر این اساس به ارائه روشی مناسب برای استفاده همزمان از تعداد زیادی ترد و دست یافتن به افزایش سرعتی درخور، بدون کاهش معنی دار در کیفیت نهایی راه حل پرداخته شود. بدین منظور در این تز چهار روش پیشنهاد و تحلیل شده اند و افزایش سرعت هریک از روش های ارائه شده، نسبت به آنچه در چینش VPR سریال استاندارد انجام می گیرد مورد ارزیابی قرار گرفته است. در روش اول یعنی روش حرکات موازی، تکرار زیاد در فراخوانی یکی از توابع در اجرای سریال چینش VPR منجر به بررسی امکان تقسیم این تکرارها میان پردازنده هایی شد که می توانند پردازش همزمان و موازی داشته باشند. در روش دوم که مبتنی بر تأمل

[1] General Purpose Graphics Processing Units (GPGPU)
[2] Versatile Place and Route

و پیش بینی بود، سعی شد تا روشی بر اساس ایده شروع عملیات بعدی با فرض نتیجه عملیات پیشین و ارزیابی نتیجه عملیات پیشین توسط پردازنده های موازی، ارائه شود و افزایش سرعت حاصل از این روش مشخص گردد. روش سوم با نام میانگین مختصات بلوک ها، به دنبال یافتن موقعیت مکانی چینش جدید با استفاده از همه پرسی و میانگین گیری از نتیجه چینش هر ترد که کاملا مستقل از سایر تردها روی یک چینش مشترک اولیه عمل می کند، می باشد و در روش آخر تلاش می شود مسأله از لحاظ فیزیکی به چندین پارتیشن شکسته شود و هر پارتیشن به صورت موازی با سایر پارتیشن ها، توسط تردها مرتب شده و چینش یکپارچه را تشکیل دهند. نتایج شبیه سازی با استفاده از یک پردازنده بیانگر این است که روش حرکات موازی در صورت پیاده سازی مناسب می تواند به ازای تعداد تردهای محدود، افزایش سرعتی تا حدی مناسب را نتیجه دهد. این در حالی که در روش ارائه شده دوم یعنی روش مبتنی بر محاسبات براساس تأمل و پیش بینی دستیابی به افزایش سرعتی محدود بدست آمد. روش میانگین مختصات بلوک ها نهایتا به شکست انجامید و روش آخر که بر اساس پارتیشن بندی مسأله چینش استوار بود به ازای تعداد تردهای بالا افزایش سرعت محدودی را نتیجه داد.

1-5 ساختار پایان نامه

ساختار فصل ها در این پایان نامه به صورت زیر است:

فصل دوم اختصاص به مفاهیم پایه و نیز کارهای انجام شده دارد. در ابتدای این فصل، به منظور آشنایی با معماری FPGA، به بیان ساختار اینگونه تراشه ها پرداخته می شود. FPGA ها از مجموعه ای از عناصر منطقی و مسیریابی تشکیل شده اند که بر اساس نوع FPGA مورد نظر، به گونه ای در سطح تراشه چیده شده اند. هریک از بلوک های منطقی قادر به پیاده سازی مداری منطقی هستند که در صورت اتصال به همدیگر بر اساس معماری مسیریابی، می توانند مدارات پیچیده ای را پیاده سازی کنند. در ادامه این فصل، VPR که از محبوب ترین ابزار های چینش و مسیریابی FPGA ها در صنعت و تحقیقات به شمار می آید، مورد بحث قرار می گیرد و الگوریتم چینش آن که بر اساس ایده گداختگی شبیه سازی شده استوار است، به طور کامل بررسی می شود. همانگونه که اشاره شد در گام چینش مجموعه ای از حرکات پیشنهاد شده، ارزیابی می شوند و آن دسته که بهبودی در کیفیت چینش ایجاد کرده اند به طور مستقیم پذیرفته می شوند و در غیر این صورت حرکات پیشنهادی بر اساس احتمالی پذیرفته و یا رد می شوند. نهایتا بر این اساس، چینش حاصل از لحاظ میزان طول سیم بکار رفته در سطح تراشه و نیز میزان تأخیر مدار پیاده سازی شده بهینه می شود. در انتهای فصل دوم، بطور مختصر به مرور کارهای انجام گرفته در زمینه موازی سازی چینش پرداخته شده و کارهای گذشته بر اساس نوع و روش موازی سازی پیشنهادی طبقه بندی گردیده است. سپس به ازای هر دسته، نمونه های موجود بیان و پس از توضیح روش، افزایش سرعت های بدست آمده ارائه گشته است.

فصل سوم اشاره به روش های پیشنهاد شده به منظور موازی سازی چینش انجام گرفته توسط الگوریتم VPR دارد. در این فصل با توجه به مشکلات و دشواری هایی که در موازی سازی الگوریتم چینش مبتنی بر گداختگی

شبیه سازی وجود دارد، سعی شده قابلیت های موازی سازی این الگوریتم یافته شود تا بر این اساس روش هایی برای تسریع چینش VPR سریال به منظور موازی سازی با استفاده از تعداد زیادی ترد، ارائه گردد. در این فصل چهار روش پیشنهاد شده و موانع ممکن در پیاده سازی هریک مورد بحث قرار گرفته است.

در فصل چهارم، روش های پیشنهادی مورد بررسی و ارزیابی قرار گرفته و نتایج حاصل از تحلیل و شبیه سازی هر یک در غالب نتیجه گیری ها و جداول و نمودارها بیان شده و درصورت ناکارامدی روشی دلیل آن نیز ذکر شده است. نتایج حاصل از روش های حرکات محاسبات بر اساس تأمل و پیش بینی و نیز روش پارتیشن بندی دلالت بر محدودیت سرعت به ازای افزایش تعداد تردها دارند. این در حالیست که روش حرکات موازی در صورت اجرا روی تعداد تردهای زیاد، با افت کیفیت شدید مواجه می شود. روش میانگین مختصات بلوک ها نیز به دلیل تداخل بسیار زیاد در چینش بلوک های منطقی، به شکست انجامید.

نهایتا فصل پنجم، نتیجه گیری نهایی از آنچه در پایان نامه به آن اشاره شده را در بر دارد و پیشنهاداتی را برای آن دسته از محققین که علاقه به کار در اینگونه موضوعات و زمینه ها دارند ارائه می دهد.

فصل دوم

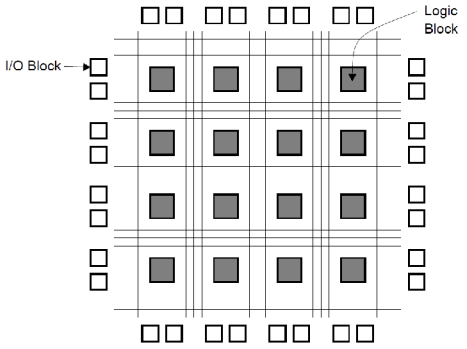
مروری بر کارهای انجام گرفته

2-1 ساختار FPGA

FPGA ها همانگونه که در شکل 2-1 نشان داده شده اند از آرایه ای از بلوک های منطقی برنامه پذیر[1] در گونه های متفاوت شامل منطق[2]، حافظه[3] و بلوک های ضرب کننده[4] تشکیل شده است که آنها را معماری های مسیریابی برنامه پذیر که به بلوک ها اجازه اتصالات مورد نظر را می دهند، احاطه نموده اند. اطراف آرایه را بلوک های برنامه پذیر خروجی و ورودی که در شکل، با I/O نشان داده شده اند، پوشش داده اند که تراشه را به دنیای خارج متصل می سازند. موضوع برنامه پذیری در FPGA، این را نشان می دهد که پس از اینکه فرآیند ساخت سیلیکونی تکمیل شد، تابع قابل برنامه ریزی به درون تراشه خواهد بود.براساس [36] این امر از طریق تکنولوژی برنامه پذیری[5] که روشی است که توان تغییر در رفتار تراشه ساخته شده با استفاده از طرح کاربران را دارد، امکان پذیر است.

---

۱ Programmable logic block
۲ logic
۳ memory
٤ multiplier
٥ Programming Technology

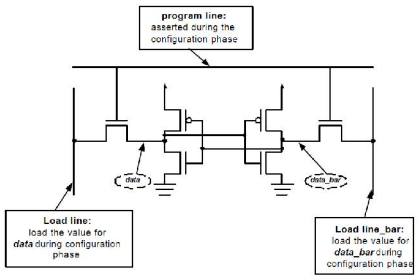


شکل2-1: ساختار FPGA (مطابق [37])

2-1-1 تکنولوژی برنامه پذیری:

مدارهای پیاده سازی شده در یک FPGA بر اساس مجموعه ای از بیت های قابل پیکربندی، در FPGA ذخیره می شوند. این بیت ها به طرق مختلفی از جمله بر اساس روش های مبتنی بر حافظه دسترسی تصادفی ایستا[1]، فلش[2] و ضد فیوز[3]، قابل ساخت هستند.

رایج ترین طراحی برای بیت های قابل پیکربندی در یک FPGA، استفاده از سلول های حافظه دسترسی تصادفی ایستا است. تکنولوژی حافظه ایستا سریع است و امکان پیکربندی و برنامه پذیری مجدد را دارد. علاوه بر این بیت های این حافظه قابل پیاده سازی با استفاده از ترانزیستورهای CMOS استاندارد می باشند، این بدین معنی است که FPGA هایی که از حافظه دسترسی تصادفی ایستا استفاده می کنند می توانند در مراحل لبه مقدم[4] پیاده سازی شوند.



شکل2-2: سلول حافظه دسترسی تصادفی ایستا 6 ترانزیستوری (مطابق [36])

شکل2-2، یک سلول حافظه دسترسی تصادفی ایستا 6 تراتزیستوری را نشان می دهد. این حافظه از بیت داده به صورت درست و نیز متمم استفاده می کند که امکان نوشتن و خواندن سریع را مهیا می سازد. اگرچه یک سلول

---

۱ SRAM
۲ FLASH
۳ Anti-fused
٤ Leading-edge