دانشگاه صنعتی امیرکبیر
(پلی‌تکنیک تهران)

دانشکده مهندسی کامپیوتر و فناوری
اطلاعات

پایان نامه کارشناسی ارشد

معماری کامپیوتر

بررسی الگوریتم‌های افراز زمانی در سیستمهای نهفته قابل بازپیکربندی پویا

توسط

محمد صادق صادقی

استاد راهنما: دکتر احمد کاردان

استاد مشاور: دکتر حسین پدرام

بهار ۱۳۸۶

بسمـه تعالـــی

دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)
معاونت پژوهشی

فرم اطلاعات پایان نامه
کارشناسی ارشد و دکترا

تاریخ: .....

پیوست: .....

نام و نام خانوادگی:	محمد صادق صادقی	**دانشجوی آزاد** ■	**بورسیه** □	**معادل** □
شماره دانشجوئی:	۸۳۱۳۱۲۱۹	**دانشکده:** مهندسی کامپیوتر	**رشته تحصیلی:** معماری کامپیوتر	
نام و نام خانوادگی استاد راهنما:	احمد کاردان			
عنوان پایان نامه به فارسی:	بررسی الگوریتم های افراز زمانی در سیستم های نهفته قابل بازپیکربندی پویا			
عنوان پایان نامه به انگلیسی:	Research in Temporal Partitioning Alorithms in Reconfigurable Embedded Systems			
نوع پروژه:	**کارشناسی ارشد** ■ / **دکترا** □	**کاربردی** □	**بنیادی** □	**توسعه ای** □ **نظری** ■
تاریخ شروع:	۸۵/۷/۱	**تاریخ خاتمه:** ۸۶/۱/۲۷	**تعداد واحد:** ۶	
سازمان تأمین کننده اعتبار:	--------			
واژه های کلیدی به فارسی:	افراز زمانی، سیستم‌های نهفته، بازپیکربندی پویا			
واژه های کلیدی به انگلیسی:	Temporal partitioning, embedded systems, Partial reconfiguration			

نظرها و پیشنهادها به منظور بهبود فعالیت های پژوهشی دانشگاه:

استاد راهنما:

دانشجو:

امضاء استاد راهنما: **تاریخ: ۸۶/۳/۱**

نسخه ۱: معاونت پژوهشی

نسخه ۲: کتابخانه و به انضمام دو جلد پایان نامه به منظور تسویه حساب با کتابخانه و مرکز اسناد و مدارک علمی

تقدیم به

پدر و ماد عزیزم

سپاسگذاری

از استاد بزرگوار جناب آقای دکتر کاردان برای راهنمایی‌های ارزشمندشان و تمامی دلگرمی‌هایی که امکان صورت گرفتن این کار را برای من فراهم آورد، کمال سپاسگذاری و قدردانی را دارم.

از استاد گرانقدر جناب آقای دکتر پدرام که زحمت مشاوره‌ی این پایان نامه را به عهده گرفتند و تجارب ارزنده‌ی خودرا در اختیار اینجانب قرار دادند، کمال تشکر را دارم.

همچنین از اساتید گرامی جناب آقای دکتر احمد اکبری و جناب آقای دکتر کاظم اکبری که داوری این پایان نامه را به عهده گرفتند، ممنون و سپاسگذار هستم.

در پایان از تمامی دوستان عزیزی که در انجام این کار از کمک و همکاری‌های آنها استفاده کرده‌ام، ممنون و سپاسگذارم.

چکیده

در این پایان‌نامه ما نوآوری خودمان را در زمینه سیستمهای نهفته قابل بازپیکربندی ارائه می‌دهیم. سیستم قابل بازپیکربندی از یک پردازنده همه منظوره و یک واحد پردازش کننده قابل بازپیکربندی ساخته شده است. در این کار یک کاربرد خاص جهت پیاده‌سازی به این معماری داده می‌شود، و سپس یک مرحله افراز نرم‌افزار/سخت‌افزار استفاده می‌شود تا بخش‌های قابل اجرا روی سخت‌افزار و نرم‌افزار را از هم جدا کند. بخش سخت‌افزاری کاربرد موردنظر روی واحد پردازش‌کننده قابل بازپیکربندی پیاده‌سازی می‌شود و بخش نرم‌افزاری کاربرد بر روی پردازنده همه منظوره پیاده‌سازی می‌شود. بخشی که روی واحد پردازش کننده قابل بازپیکربندی پیاده‌سازی می‌شود بصورت یک گراف جریان داده فراهم می‌گردد. این پایان‌نامه به بخشی از کاربرد می‌پردازد که روی واحد پردازش کننده قابل بازپیکربندی اجرا می‌شود. این واحد پردازش کننده قابل بازپیکربندی یک آرایه‌ای از دروازه‌های قابل برنامه‌ریزی(FPGA) است. چون گرافهای جریان داده‌ای که باید در این FPGA پیاده‌سازی شوند بسیار بزرگ هستند و در یک FPGA جا نمی-گیرند، باید به تکه‌های کوچکتری تقسیم شوند. سپس این تکه‌ها باید بطور متوالی داخل FPGA قرار گیرند و تابع موردنظر محاسبه شود. اگر FPGA نتواند بطور جزئی بازپیکربندی شود، این تکه‌ها برای پیکربندی کل قطعه استفاده می‌شوند. در مجموع این فرآیند تقسیم‌بندی را افراز زمانی می‌نامند. نوآوری ما پیشنهاد چندین الگوریتم بهبودیافته جهت حل مسأله افراز زمانی است. هدف الگوریتم‌های افراز، کاهش حجم داده انتقالی میان تکه‌ها جهت کاهش اندازه حافظه، استفاده از حداقل فضای سخت‌افزار قابل بازپیکربندی برای افزایش میزان بهره‌وری، کاهش تعداد تکه‌ها و در نتیجه کاهش کل زمان اجرای کاربرد است.

کلمات کلیدی

افراز زمانی، سیستم‌های نهفته، بازپیکربندی پویا

فهرست مطالب

۱ مقدمه ۲

۱-۱انگیزه ۵

۱-۲ اهداف ٦

۱-۳ تعریف های عمومی ٦

۲ محاسبات قابل بازپیکربندی ۱۰

۲-۱ فناوری ۱۵

۲-۲ FPGA های قدیمی ۱٦

۲-۳ سخت افزار ۱۹

۲-۳-۱ اتصال ریزپردازنده ۲۱

۲-۳-۲ فناوری برنامه ریزی ۲۳

۲-۳-۲-۱ SRAM ۲۳

۲-۳-۲-۲ آنتی فیوز ۲٤

۲-۳-۲-۳ EPROM, EEPROM, FLASH ۲٤

۲-۳-۳ مدل های بازپیکربندی ۲۵

۲-۳-۳-۱ بازپیکربندی ایستا ۲۵

۲-۳-۳-۲ بازپیکربندی پویا ۲۵

۲-۳-۳-۳ تک متنی ۲٦

۲-۳-۳-۴ چند متنی ۲۷

۲-۳-۳-۵ بازپیکربندی جزئی ۲۷

۲-۳-۳-۶ بازپیکربندی خط لوله ۲۸

۲-۴ نرم افزار ۲۸

۲-۵ پیکربندی سریع ۳۲

۲-۵-۱ پیش واکشی پیکربندی ۳۲

۲-۵-۲ فشرده سازی بازپیکربندی ۳۲

۲-۵-۳ جابجایی و یکپارچگی در سیستم های قابل بازپیکربندی جزئی ۳۳

۲-۵-۴ نهان سازی پیکربندی ۳۳

۲-۶ سیستم های نهفته ۳٤

۲-۶-۱ گرایش ها در مهندسی سیستم های نهفته ۳۵

۳ معماری سخت افزار قابل بازپیکربندی مقصد ۳۹

۳-۱ معماری FPGA ٤٠

۳-۱-۱ ارتباط بین پردازنده و FPGA ٤١

۳-۲ افراز سخت افزار/ نرم افزار ٤١

۳-۳ سخت افزار مقصد ٤٢

۴ الگوریتمهای مختلف موجود و پیشنهادی در زمینه افراز زمانی ٤٦

۴-۱ کارهای گذشته ٤٦

۴-۱-۱ زمانبندی لیست ٤٨

۴-۱-۲ جریان شبکه ٥٠

۴-۱-۳ برنامه ریزی خطی ٥٢

۴-۱-۴ روش احتمالی ٥٣

۴-۲ شباهتها، تفاوتها و نوآوریها ٥٣

۴-۳ الگوریتمهای پیشنهادی ٥٤

۴-۳-۱ الگوریتم افراز زمانی با زمانبندی لیست پویا در افزاره با پیکربندی جزئی ٥٥

۴-۳-۲ تعریف های عمومی درمعماری آرایه ای از دروازههای قابل برنامه ریزی تسهیم زمانی ٦٠

۴-۳-۳ الگوریتم زمانبندی لیست پویا در معماری آرایه ای از دروازههای قابل برنامه ریزی تسهیم زمانی ٦١

۴-۳-۴ الگوریتم افراز زمانی بر اساس روش برنامه ریزی خطی برای آرایه ای از دروازههای قابل برنامه ریزی تسهیم زمانی ٦٣

۴-۳-۴-۱ تعریفها و محدودیتها ٦٤

۴-۳-۴-۲ تعریف مسأله ٦٥

۴-۳-۴-۳ فرموله کردن برای مسأله برنامه ریزی خطی ٦٦

۴-۳-۴-۴ افراز زمانی با توجه به کارایی زمان ٧٢

۴-۴ جمع بندی و نتیجه گیری ٧٧

۵ نتایج حاصل از آزمایش الگوریتمهای پیاده سازی شده ۸۰

۵-۱ محیط آزمایش ۸۰

۵-۱-۱ ابزار محک زنی ۸۰

۵-۲ ارزیابی روش زمانبندی لیست پویا در مقابل ایستا ۸۱

۵-۳ ارزیابی الگوریتم افراز زمانی مبتنی بر روش برنامه ریزی خطی برای TMFPGA ۸۷

۶ نتیجه گیری ۹۲

۶-۱ کارهای آینده ۹۳

۷ مقاله های ارائه شده ۹۵

۸ فهرست منابع ۹۷

فهرست شکل‌ها

شکل۲-۱: موقعیت محاسبات قابل بازپیکربندی ۱۱

شکل۲-۲: بخشهای محاسباتی شدید از کد کاربردی ۱۲

شکل۲-۳: مؤلفه های یک بلوک منطقی پایه ۱۷

شکل۲-۴: معماری مسیریابی FPGA به سبک جزیره ای ۱۸

شکل۲-۵: سطوح مختلف اتصال در سیستم قابل بازپیکربندی [25]. ۲۱

شکل۲-۶: سه روش جریان طراحی برای پیاده سازی الگوریتم ۲۹

شکل۳-۱: ساختار یک FPGA [46] ٤٠

شکل۳-۲: مدل پیکربندی Xilinx TMFPGA [63] ٤٤

شکل۳-۳: FPGA تسهیم زمانی [63] ٤٤

شکل۴-۱: مدل جریان شبکه برای افراززمانی[76] ۵۱

شکل۴-۲: یک برش درمدل ساختگی جریان شبکه و ایجاد برش در نتها [76] ۵۱

شکل۴-۳: گراف جریان داده ٥٦

شکل۴-۴: زمانبندی الگوریتمASAP ۵۷

شکل۴-۵: حاصل زمانبندی الگوریتم ALAP ۵۷

شکل۴-۶: الگوریتم کشف تشابه ۵۹

شکل۴-۷: چهار مرحله در مسأله افراز با محدودیت اولویت از یک مدار ٦١

شکل۴-۸: نحوه فشرده سازی[62] ٦٣

شکل۴-۹: افراز با محدودیت اولویت، حداکثر وزن ۵ و حداکثر ارتباط ۵ ..... ٦٦

شکل۴-۱۰: افراز با محدودیت اولویت، حداکثر وزن ۳ و حداکثر ارتباط ۳ ..... ٦٦

شکل۴-۱۱: (الف) بخشی از مدار ترکیبی (ب) نتیجه TPG با D=2.....۷٤

شکل۵-۱: نحوه تولید نت لیست ۸۱

شکل۵-۲: بهبود زمان بازپیکربندی ۸٦

فهرست جدول‌ها

جدول ۵-۱: مشخصات گراف جریان داده (مدارهای انتخابی از ابزار MCNC) ۸۲

جدول ۵-۲: نتایج پیاده سازی گراف جریان داده بازپیکربندی با الگوریتم پویا ۸۲

جدول ۵-۳: نتایج پیاده سازی گراف جریان داده بازپیکربندی با الگوریتم ایستا..... ۸۳

جدول ۵-۴: نتایج پیاده سازی گراف جریان داده بازپیکربندی با الگوریتم پویا ۸۵

جدول۵-۵: تأثیر الگوریتم پویا بر شباهت گره و افزایش سرعت بازپیکربندی ۸۶

جدول۵-۶ : بهبود حاصل از پیاده سازی براساس الگوریتم پویا درحافظه..... ۸۷

جدول۵-۷: مشخصات گراف جریان داده..... ۸۸

جدول۵-۸: نتایج حاصل از پیاده سازی الگوریتم های مورد مقایسه..... ۸۹

جدول۵-۹: بهبود حاصل از کاهش حافظه مصرفی ۹۰

فصل اول

مقدمه

1

۱ مقدمه

پردازنده‌های همه منظوره امکان پیاده‌سازی انواع توابع را با استفاده از قطعه یکسان فراهم می‌کنند. اما انعطاف‌پذیری زیاد و قابلیت محاسباتی همه منظوره، منجر به نرخ بالای کلاک برای کارایی می‌شود. با توجه به معماری "وان نیومن" سربار قابل توجهی متحمل اجرای دستورالعمل می‌شود. ابتدا یک دستورالعمل واکشی و رمزگشایی می‌شود، سپس عملوندها خوانده و عملگرهای کد شده در دستورالعمل اجرا می‌شوند، نهایتاً نتیجه در حافظه ذخیره می‌شود.

زمانیکه اجرای یک تابع جریان داده[1] روی یک پردازنده همه منظوره بسیار زمانگیر می‌شود، پیاده‌سازی سخت‌افزای آن بهتر خواهد بود. بنابراین پیاده‌سازی آن معمولا در یک ASIC بگونه‌ای انجام می‌شود که

[1] Data flow

تابع بصورت سخت‌افزاری پیاده‌سازی می‌شود و دیگر قابل تغییر نیست. چون ASIC برای تنها یک تابع بهینه‌سازی شده است و سربار ناشی از واکشی و رمزگشایی، خواندن و ذخیره داده را حذف می‌کند، معمولا کارا ست. ASICها انعطاف‌پذیری ندارند و توابع پیاده‌سازی شده بر روی آنها در طول عمر قطعه غیرقابل تغییر هستند. بعلاوه پیاده‌سازی یک تابع در ASIC فرآیند طولانی و سختی است. هزینه آن نیز بالا است و تنها با تولید انبوه، قابل قبول است. بطور ایده‌آل، ما تمایل به داشتن انعطاف‌پذیری پردازنده همه منظوره و کارایی ASIC را در یک قطعه داریم، تا بتوانیم توابع مختلف را مستقل از حجم تولید، به منظور کاهش سربار هزینه پیاده‌سازی کنیم.

ترکیب انعطاف‌پذیری پردازنده همه منظوره و کارایی ASIC در یک قطعه یک راه‌حل مناسب است. یک نمونه جدیدی از واحد پردازش‌کننده بنام واحد پردازش‌کننده قابل بازپیکربندی ساخته شده و نشان داده که کارایی آن به مراتب بهتر از نمونه‌های پردازنده همه منظوره است [1, 2, 3]. بنابراین زمینه تحقیقاتی جدیدی به نام محاسبات قابل‌بازپیکربندی بوجود آمد.

قبلاً FPGA فقط برای ساخت نمونه اولیه از طراحی‌های ASIC به تعداد کمی تولید می‌شد، چون سرعت آن پایین، مصرف انرژی و هزینه آن زیاد بود. اما امروزه با رشد فناوری FPGAها، کاهش هزینه و کوتاه‌شدن زمان ارائه به بازار، انگیزه استفاده از FPGA بجای ASIC در طراحی سیستمهای نهفته[۲] افزایش یافته‌است[4].

تاکنون بیشتر طراحی FPGAها از طراحی سنتی ASIC پیروی می‌کرد و محدود به زمان بارگذاری پیکربندی مجدد می‌شد. بازپیکربندی پویا[۳] یا زمان اجرا یکی از مشخصه‌هایی است که کارایی و هزینه را در زمان بارگذاری بهبود می‌بخشد.

---

[2] Embedded system
[3] Dynamic reconfiguration

سیستمهای نهفته‌ای که از FPGAها استفاده می‌کنند با قابلیت بازپیکربندی مجدد در زمان اجرا هزینه بیشتری را صرفه‌جویی می‌کنند. بعنوان یک سیستم قابل برنامه‌ریزی، یک معماری قابل‌بازپیکربندی پویا تنها زمانی مؤثر واقع می‌شود که یک راه‌حل بهتری نسبت به سیستمهای دیگری مانند: DSP و پردازنده superscalar از جهت کارایی، هزینه و انرژی فراهم‌کند. در نظرگرفتن افزایش هرچه بیشتر کارایی پردازنده‌ها و یک روش خوب برای موفقیت در این روش، ضروری است. تعدادی از شرکت‌ها محصولاتی را تحت عنوان سیستمهای قابل‌بازپیکربندی پویا به بازار عرضه کرده‌اند[10]-[5]. اما تمام آنها روی طراحی معماری متمرکز شده‌اند و فرآیند طراحی را از یک دیدگاه سیستمی درنظر نگرفته‌اند. تحقیق در طراحی مشترک سخت‌افزار/نرم‌افزار به چگونگی طراحی سیستمهای غیرهمگن می‌پردازد. هدف اصلی، کوتاه‌کردن زمان ارائه به بازار است در حالیکه در جهت کاهش هزینه طراحی و فعالیت‌های طراحی تلاش می‌شود. جریان طراحی، شامل مشخص-سازی سیستم، تخمین هزینه، تقسیم‌بندی سخت‌افزار/نرم‌افزار، سنتز مشترک، شبیه‌سازی مشترک است. چون سیستمهای نهفته قابل بازپیکربندی پویا شامل پردازنده و FPGAها هستند، در این جریان طراحی مشترک، قرارمی‌گیرند. بنابراین بعلت قابلیت بازپیکربندی FPGAها، بعضی از تفاوتهای اساسی، مخصوصاً در افراز، زمانبندی و سنتز مشترک وجود دارد.

تلاشهایی در جهت افراز و زمانبندی خودکار برای معماری‌های قابل بازپیکربندی انجام شده است. معماری Garp [10] و کامپایلر آن توسط Hauster ارائه شد. معماری Garp، سخت‌افزار قابل پیکربندی را با یک پردازنده MIPS استاندارد روی یک دای مشترک ترکیب کرد. خصیصه‌های طراحی اجازه می‌دهد، سخت-افزار در میکروثانیه بازپیکربندی شود. کامپایلر Garp موازی‌سازی را در سطح دستورهای کد C انجام می-دهد و چرخه‌های انتخاب‌شده را مستقیماً به آرایه قابل بازپیکربندی کامپایل می‌کند.

در SARCS، یک ابزار سنتز سطح بالا برای تخمین منابع و تأخیر بکارگرفته شد. یک مدل برنامه‌ریزی خطی برای حل مسأله افراز زمانی ومکانی ارائه شد. این جریان، کامل و مناسب طراحی شد، اما بر اساس طراحی سخت‌افزارهای قدیمی بود و با درنظرگرفتن بازپیکربندی پویا پیچیدگی آن افزایش می‌یافت.

۱-۱ انگیزه

امکان نگاشت یک کاربرد با هر اندازه دلخواه به یک قطعه قابل بازپیکربندی با فضای سخت‌افزاری محدود، انگیزه‌ای برای روش مجازی‌سازی[4] است. روش افراز زمانی یکی از تکنیکهای مجازی‌سازی است. افراز زمانی، کاربرد را به بخش‌های کوچک تقسیم می‌کند تا داخل قطعه سخت‌افزاری قرارگیرد، سپس این بخش‌ها بطور پی‌درپی اجرا می‌شوند. زمانیکه قطعه‌های قابل بازپیکربندی برای کاربردهای بزرگ، بسیار کوچک باشند از این روش استفاده می‌شود. امروزه با توجه به FPGAهایی که در سیستمهای نهفته بکار گرفته می‌شوند صرفه‌جویی در فضا و هزینه، امر مهمی بشمار می‌رود.

بازپیکربندی پویا بطور وسیع زمینه کاربرد فنآوری FPGA را با وجود خصیصه‌هایی نظیر: افزایش چگالی عملیاتی، امکان پیاده‌سازی مدارهای تطبیق پذیر، گسترش داد. از طرف دیگر، این خصیصه‌ها به فضای طراحی افزوده شدند، بنابراین زمان طراحی مدارها نیاز به بررسی دارند. مسأله بسیار مهمی که در سبک طراحی قدیمی دیده نمی‌شد، وابستگی داخلی زمانی از بخش‌های مستقل است که بایستی بازپیکربندی شوند، همچنین جهت تغییر محتوای پیکربندی در بدترین حالت، زمان تحت تأثیر قرار می‌گیرد، بنابراین بدون در نظرگرفتن ابزارهای طراحی و روش‌های بهینه‌سازی، روند تولید با وجود این مسأله‌ها غیرممکن می‌شود.

[4] virtualization

۱-۲ اهداف

سهم ما در زمینه سیستمهای قابل بازپیکربندی در جهت بهبود و بررسی الگوریتم‌های مختلف برای حل مسأله افراز زمانی است. ابتدا مسأله افراز زمانی برای معماری قابل بازپیکربندی جزئی[5] مطرح خواهد شد که در حال حاضر افزاره‌هایی از شرکت‌های مختلف از این قابلیت پشتیبانی می‌کنند و در ادامه مسأله افراز زمانی برای آرایه‌ای از دروازه‌های قابل برنامه‌ریزی تسهیم زمانی[6] ارائه می‌شود. آرایه‌های قابل برنامه‌ریزی تسهیم زمانی کارایی را با استفاده مجدد از سخت‌افزار بصورت پویا بهبود می‌بخشند.

توابعی که برای قرارگرفتن در یک FPGA بسیار بزرگ هستند باید به بخشهای مختلفی تقسیم‌بندی شوند تا بطور پی‌درپی بر اساس اولویت‌بندی از قبل تعیین شده روی FPGA اجرا شوند. ما دو روش را برای حل مسأله افراز زمانی پیاده‌سازی کرده‌ایم و دلایل و انگیزه‌هایی را برای انتخاب آنها مطرح می‌کنیم و نتایج را با الگوریتم‌های دیگر مقایسه می‌کنیم. اولین الگوریتم یک نسخه بهبود یافته از الگوریتم معروف زمانبندی لیست است که روی معماری با افزاره قابل بازپیکربندی جزئی پیاده‌سازی شده و دومین الگوریتم نیز با استفاده از همین روش اما روی معماری تسهیم زمانی بکارگرفته شده است. سومین الگوریتم بر اساس مدل برنامه‌ریزی خطی است و هدف کاهش تعداد ارتباطها، با توجه به محدودیت اولویت است.

۱-۳ تعریف های عمومی

گراف جریان داده[7]: یک گراف جهت دار مانند $G(V, E)$ که هر گره آن معادل با یک عملگر است و کمان‌های گراف نیز بیانگر وابستگی (اتصالها) بین گره‌ها یا عملگرها هستند. اگر خروجی یک

---

[5] Partial Reconfiguration
[6] Time Multiplex Field Programmable Gate Array
[7] Data Flow Graph

گره v_i توسط یک گره دیگر v_j مورد استفاده قرار گیرد کمان $e_i \in E$ بین این دو گره وجود خواهد داشت.

پیکربندی: مجموعه‌ای از رشته بیت‌ها یا اطلاعات لازم برای پیکربندی تراشه برنامه‌پذیر.

پیکربندی جزئی: دسترسی انتخابی به حافظه پیکربندی را اجازه می‌دهد. تغییر پیکربندی در حالیکه مابقی مدار بطور کامل در حال اجرا است. سرعت بازپیکربندی پویا بطور مستقیم وابسته به تعداد مکانهای حافظه پیکربندی است که لازم به تغییر هستند.

حافظه پیکربندی چند متنی[8]: پیکربندی‌های متوالی را در چندین متن از حافظه پیکربندی قرار می‌دهد. هر لحظه فقط یک پیکربندی در متن فعال قرار می‌گیرد. جابجایی متن فعال می‌تواند به سرعت با پیکربندی کل آرایه انجام گیرد.

سخت‌افزار (افزاره) برنامه‌پذیر: شامل تراشه برنامه‌پذیر دانه‌ریز(مانند FPGA) است که دارای اجزا و معماری مشخص است. در روش‌های ارائه شده در فصل‌های آتی از دو نوع معماری استفاده می‌شود. معماری اول قابلیت پیکربندی بصورت جزئی دارد. معماری دوم از پیکربندی چند متنی پشتیبانی می‌کند.

این پایان نامه بصورت زیر سازماندهی شده‌است:

- بخش ۲: **محاسبات قابل بازپیکربندی** پیش‌زمینه اساسی را فراهم می‌کند، بحث پیرامون ساختار و عملکرد FPGA و انواع سخت‌افزار قابل بازپیکربندی و سیستم های نهفته.

[8] multi context

- بخش ۳: **معماری سخت‌افزار قابل بازپیکربندی مقصد** معماری مقصد را توصیف می‌کند. سخت‌افزار قابل بازپیکربندی بصورت جزئی، آرایه‌ای از دروازه‌های قابل برنامه‌ریزی تسهیم زمانی.
- بخش ۴: **الگوریتم‌های مختلف موجود و پیشنهادی در زمینه افراز زمانی** الگوریتم‌های مختلف در زمینه افراز زمانی را ارائه می‌کند و بررسی اجمالی روی کارهای قبلی. سه الگوریتم بحث می‌شود، افراز زمانی بر اساس زمانبندی لیست پویا مبتنی بر معماری قابل بازپیکربندی جزئی، زمانبندی لیست پویا مبتنی بر معماری تسهیم زمانی، افراز زمانی بر اساس برنامه‌ریزی خطی بر معماری تسهیم زمانی.
- بخش ۵: **نتایج حاصل از آزمایش الگوریتم‌های پیاده‌سازی شده** بررسی نتایج حاصل از آزمایش الگوریتم‌های پیاده‌سازی شده روی نمونه‌های آزمایشی گرفته شده از ابزار محک‌زنی MCNC Partitioning93 با سایر الگوریتم‌های موجود.
- بخش ۶ **نتیجه‌گیری** بیان خلاصه‌ای از سهم ما در زمینه افراز زمانی و معرفی زمینه‌های کار و تلاش در آینده.

فصل دوم

محاسبات قابل بازپیکربندی