
بسم الله الرحمن الرحيم

دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر و فنآوری اطلاعات

ارائه یک هسته سخت‌افزاری جهت تشخیص دستورالعمل‌های سفارشی و باز پیکربندی معماری به صورت پویا

پایان‌نامه برای دریافت درجه کارشناسی ارشد

در رشته مهندسی کامپیوتر گرایش معماری کامپیوتر

نگارش:

حسن اصغریان

استاد:

دکتر مرتضی صاحب‌الزمانی

فروردین ۱۳۸۸

بسمه تعالی

تاریخ:
شماره:

دانشگاه صنعتی امیرکبیر
(پلی‌تکنیک تهران)

فرم اطلاعات پایان‌نامه
کارشناسی- ارشد و دکترا

معاونت پژوهشی
فرم پروژه تحصیلات تکمیلی ۷

مشخصات دانشجو:

نام و نام خانوادگی: **حسن اصغریان** دانشجوی آزاد ● بورسیه ○ معادل ○

شماره دانشجوئی: **۸۵۱۳۱۰۳۷** دانشکده: **مهندسی کامپیوتر** رشته تحصیلی: مهندسی کامپیوتر گروه: **معماری کامپیوتر**

مشخصات استاد راهنما:

نام و نام خانوادگی: **مرتضی صاحب‌الزمانی** درجه و رتبه: **دانشیار دانشگاه صنعتی امیر کبیر**

نام و نام خانوادگی: درجه و رتبه:

مشخصات استاد مشاور:

نام و نام خانوادگی: **فرهاد مهدی‌پور** درجه و رتبه: **دانشیار دانشگاه کیوشو (ژاپن)**

نام و نام خانوادگی: درجه و رتبه:

عنوان پایان‌نامه به فارسی :

ارائه یک هسته سخت‌افزاری جهت تشخیص دستورالعمل‌های سفارشی و باز پیکربندی معماری به صورت پویا

عنوان پایان‌نامه به انگلیسی:

A HARDWARE CORE FOR DETECTING CUSTOM INSTRUCTIONS AND RECONFIGURING ARCHITECTURE DYNAMICALLY

نوع پروژه: کارشناسی○ ارشد● دکتر ا ○ سال تحصیلی: **۸۸-۱۳۸۷**

کاربردی ● بنیادی○ توسعه‌ای● نظری○

تاریخ شروع : **آبان ۱۳۸۶** تاریخ خاتمه : **فروردین ۱۳۸۸** تعداد واحد : **۶** سازمان تأمین کننده اعتبار : --

واژه‌های کلیدی به فارسی: پردازنده قابل توسعه، محاسبات قابل باز پیکربندی، دستورالعمل های سفارشی.

واژه‌های کلیدی به انگلیسی : Extensible Processor, Reconfigurable Computing, Custom Instruction

مشخصات ظاهری	تعداد صفحات **۱۰۵**	تصویر●	جدول●	نمودار●	نقشه●	واژه‌نامه●	تعداد مراجع **۴۶**	تعداد صفحات ضمائم -
زبان متن	فارسی●	انگلیسی ○		چکیده	فارسی●	انگلیسی●		
یادداشت								

نظرها و پیشنهادها به منظور بهبود فعالیت‌های پژوهشی دانشگاه

استاد:

دانشجو:

امضاء استاد راهنما: **تاریخ:**

۱: ارائه به معاونت پژوهشی به همراه یک نسخه الکترونیکی از پایان‌نامه و فرم اطلاعات پایان‌نامه بصورت PDF همراه چاپ چکیده (فارسی انگلیسی) و فرم اطلاعات پایان‌نامه

۲: ارائه به کتابخانه دانشکده (شامل دو جلد پایان نامه به همراه نسخه الکترونیکی فرم در لوح فشرده طبق نمونه اعلام شده در صفحه خانگی کتابخانه مرکزی)

مرکزی)

تعهدنامه دانشجویان دوره تحصیلات تکمیلی

بدینوسیله؛ اینجانب **حسن اصغریان** تعهد می‌نمایم که مطالب ارائه شده در این پایان‌نامه حاصل کار پژوهشی و تحقیق اینجانب می‌باشد و قبلا برای احراز هیچ مدرک دیگری ارائه نشده است. رجوع به دست‌آوردهای پژوهشی دیگران که در این پایان‌نامه از آنها استفاده شده، مطابق مقررات ارجاع داده شده است. چنانچه در هر شرایطی این موارد بدرستی رعایت نگردد؛ دانشگاه مجاز به ابطال پایان‌نامه خواهد بود. کلیه حقوق مادی و معنوی این اثر متعلق به دانشکده مهندسی کامپیوتر دانشگاه صنعتی امیرکبیر می‌باشد.

نام و نام خانوادگی دانشجو: حسن اصغریان

تاریخ: امضا:

چکیده

یکی از راه‌کارهای افزایش کارایی پردازنده‌های همه منظوره، استفاده از پردازنده‌ها با مجموعه دستورالعمل‌های قابل توسعه است. در این نوع پردازنده‌ها امکان اضافه کردن دستورالعمل‌های جدید برای پردازنده وجود دارد. مشکلی که برای توسعه برنامه بر روی این پردازنده‌ها وجود دارد، پیچیدگی فرایند طراحی در آنها است. انتخاب خودکار دستورالعمل‌های سفارشی و تشخیص آنها هزینه بالایی دارد. شناسایی دستورالعمل‌های سفارشی بر مبنای برنامه کاربردی و با توجه به ورودی‌های آن انجام می‌شود. عموما برای شناسایی دستورالعمل‌های سفارشی و اجرای آنها از روش‌های نرم‌افزاری و شبیه‌سازی با کمک داده‌های واقعی استفاده می‌شود.

در این پایان‌نامه یک بستر محاسباتی قابل باز پیکربندی برای اجرای برنامه‌های کاربردی ارائه داده شده است که در آن تشخیص دستورالعمل‌های سفارشی هم‌زمان با اجرای برنامه کاربردی صورت می‌گیرد. نقطه قوت استفاده از بستر محاسباتی پیشنهادی برای تشخیص دستورالعمل‌های سفارشی، عدم نیاز آن به شبیه‌سازی با داده‌های زمان اجرا است. اجرای دستورالعمل‌های سفارشی بر روی این بستر محاسباتی به صورت مخفی از دیدگاه کاربر نهایی سیستم انجام می‌شود. برای اجرای برنامه‌های کاربردی بر روی این بستر محاسباتی نیازی به انجام تغییرات بر روی کد باینری نیست و به همین دلیل فرایند طراحی برنامه برای آن کاملا مشابه به فرایند طراحی برنامه برای یک پردازنده همه منظوره است.

حجم سخت‌افزار اضافی برای تشخیص دستورالعمل‌های سفارشی که بر روی FPGA پیاده‌سازی شده است، کمتر از ۲۰٪ حجم پردازنده پایه استفاده شده در این بستر محاسباتی پیشنهادی است. نتایج آزمایش‌های انجام شده بر روی این بستر محاسباتی حاکی از افزایش سرعت برنامه‌های کاربردی تا حداکثر ٤۰٪ و به طور میانگین حدود ۲۰٪ است.

کلمات کلیدی: پردازنده قابل توسعه، محاسبات قابل باز پیکربندی، دستورالعمل‌های سفارشی.

فهرست مطالب

فهرست جدول‌ها

فهرست شکل‌ها

فصل اول

مقدمه

۱ مقدمه

امروزه سه رهیافت کلی زیر برای انجام محاسبات در حوزه کامپیوتر و الکترونیک وجود دارد:

۱- پردازنده‌های همه منظوره

۲- مدارهای تمام سفارشی

۳- محاسبات قابل باز پیکربندی

پردازنده‌های همه منظوره بیشترین انعطاف‌پذیری و مدارهای تمام سفارشی کمترین قابلیت انعطاف را در بین روش‌های نام برده شده دارند. از نظر کارایی محاسباتی و مصرف توان، مدارهای تمام سفارشی بهترین گزینه و پردازنده‌های همه منظوره بدترین انتخاب هستند.

با معرفی محاسبات قابل باز پیکربندی در دو دهه گذشته شاهد ترکیب مزایای دو روش طراحی همه منظوره و تمام سفارشی بوده‌ایم. در این نوع محاسبات، تلاش شده است تا کارایی بالای سخت‌افزار در کنار انعطاف‌پذیری نرم‌افزار به دست آید. ارائه پردازنده‌هایی با عنوان پردازنده‌های قابل توسعه را می‌توان نتیجه این ترکیب دانست. در

این پردازنده‌ها امکان افزودن تعدادی دستورالعمل جدید به مجموعه دستورالعمل‌های پایه پردازنده وجود دارد. مساله اصلی در استفاده از پردازنده‌های قابل توسعه مربوط به تعریف دستورالعمل‌های جدید و سیاست استفاده از این دستورالعمل‌ها در زمان اجرای برنامه است. مساله اصلی موجود در استفاده از محاسبات قابل باز پیکربندی برای پیاده‌سازی کاربردهای مختلف، در پیچیدگی فرایند طراحی برنامه برای آنها است.

در این پایان‌نامه یک هسته سخت‌افزاری جهت تشخیص دستورالعمل‌های سفارشی ارائه داده شده است که از آن در یک بستر محاسباتی قابل باز پیکربندی پویا استفاده شده است. در بستر محاسباتی پویای پیشنهادی در این پایان‌نامه، هسته سخت‌افزاری در کنار یک پردازنده همه منظوره قرار گرفته و با تشخیص هر دستورالعمل سفارشی، آن را با کمک واحدهای محاسباتی قابل برنامه‌ریزی خود اجرا می‌کند.

۱-۱. معرفی

در دهه‌های گذشته با توسعه صنعت ریز پردازنده‌ها، همواره تلاش شده است تا کارایی پردازنده‌ها با بالا بردن فرکانس کاری آنها افزایش پیدا کند. امروزه با افزایش کارایی پردازنده‌ها امکان پردازش بی‌درنگ بسیاری از کاربردهایی که در گذشته ناممکن به‌نظر می‌رسید، فراهم شده است. با این وجود هنوز نیازهای پردازشی بسیاری از برنامه‌های کاربردی بیش از توان پردازشی پردازنده‌ها است و این موضوع باعث شده است که موضوع افزایش کارایی پردازنده‌ها کماکان یکی از اهداف مهم پژوهشی به‌شمار آید.

افزایش فرکانس کاری پردازنده‌ها یکی از راه‌کارهای اصلی افزایش کارایی آنها است. یک از مشکلات اصلی افزایش فرکانس کاری پردازنده‌ها افزایش توان مصرفی آنها است. در چند سال اخیر[۱]، تلاش برای بهبود کارایی پردازنده‌ها با استفاده از افزایش موازی‌سازی در اجرای برنامه‌های کاربردی مورد نظر قرار گرفته است. معرفی پردازنده‌های چند هسته‌ای مختلف نیز با هدف افزایش کارایی اجرای برنامه‌های کاربردی از طریق اجرای موازی آنها انجام شده است.

با وجود توسعه پردازنده‌های چند هسته‌ای و با فرض حل مسایل نرم‌افزاری مربوط به توسعه برنامه کاربردی برای آنها، بسیاری از برنامه‌های کاربردی برای اجرا در محیط‌های چند هسته‌ای مناسب نیستند. در چنین کاربردهایی

[۱] پس از سال ۲۰۰۰

استفاده از سخت‌افزارهای خاص منظوره پردازشی به عنوان شتاب‌دهنده[۲] در کنار پردازنده اصلی باعث افزایش کارایی اجرای برنامه کاربردی می‌شود. افزودن شتاب‌دهنده سخت‌افزاری به یک طرح عمومی نه تنها باعث افزایش کارایی آن می‌شود بلکه باعث کاهش قابل ملاحظه توان مصرفی آن نیز می‌گردد [۱].

سه چالش اصلی که در طراحی و به کارگیری شتاب‌دهنده‌های سخت‌افزاری از آنها نام برده می‌شود، عبارتند از [۱]:

۱- طراحی شتاب‌دهنده

۲- ارتباط شتاب‌دهنده با پردازنده همه منظوره

۳- توسعه برنامه کاربردی

۱-۱-۱. طراحی شتاب‌دهنده‌های محاسباتی

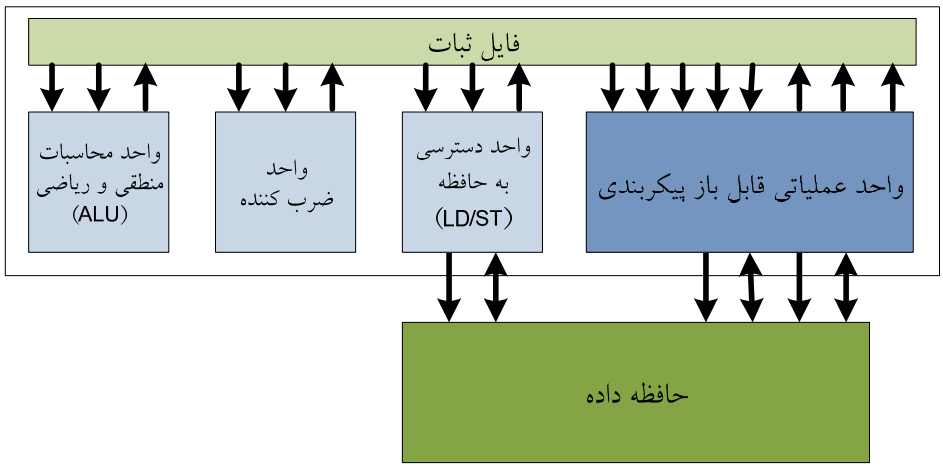
مصالحه[۳] اصلی در طراحی یک شتاب‌دهنده محاسباتی بین کارایی و انعطاف‌پذیری در آن است. یک شتاب‌دهنده سخت‌افزاری با انجام حجم بالاتری از محاسبات خود به صورت موازی، کارایی بیشتری نسبت به پردازنده‌های عمومی ارائه می‌دهد. هر چه معماری شتاب‌دهنده برای یک کاربرد خاص بهتر تنظیم شود، افزایش کارایی بهتری را به همراه خواهد داشت ولی استفاده از آن تنها محدود به همان کاربرد می‌شود. به همین دلیل در طراحی یک شتاب‌دهنده باید موضوع کارایی و انعطاف‌پذیری در کنار یکدیگر مورد نظر قرار بگیرند.

در حال حاضر مشهورترین روش برای استفاده از شتاب‌دهنده‌های سخت‌افزاری، به‌کارگیری مدارهای تمام سفارشی[۴] به عنوان شتاب‌دهنده در کنار پردازنده‌های عمومی است. اجرای بخش‌های بزرگی از برنامه کاربردی که سرعت اجرای آنها در پردازنده عمومی کم است، توسط این مدارهای تمام سفارشی انجام می‌شود. از آنجایی‌که استفاده از مدارهای تمام سفارشی قابلیت انعطاف مناسب و نیز قابلیت برنامه‌ریزی پس از تولید را ندارد، جز در موارد خاص استفاده نمی‌شود.

روش دیگر برای طراحی و استفاده از شتاب‌دهنده سخت‌افزاری، تجهیز پردازنده پایه به یک سخت‌افزار خاص با قابلیت برنامه‌ریزی برای انجام محاسبات موازی است. در این روش مجموعه دستورالعمل‌های پردازنده پایه برای

[2] ACCELERATOR
[3] TRADE-OFF
[4] ASICS

پشتیبانی از مجموعه عملیات جدیدی که توسط شتاب‌دهنده قابل انجام است، توسعه داده می‌شود. به چنین پردازنده‌هایی قابل توسعه یا قابل پیکربندی نرم‌افزاری می‌گویند. نمونه‌ای از چنین پردازنده‌هایی در [۲] معرفی شده است. در این پردازنده‌ها، تعدادی واحد محاسباتی قابل بازپیکربندی در کنار واحدهای اجرایی پردازنده قرار داده می‌شوند و امکانات نرم‌افزاری لازم برای استفاده از این واحدهای اجرایی جدید نیز طراحی می‌شوند. ساختار کلی پردازنده‌های قابل توسعه در شکل ۱-۱ نمایش داده شده است.



شکل ۱-۱ نمونه‌ای از یک پردازنده قابل توسعه

پردازنده‌ها با مجموعه دستورالعمل‌های سفارشی[۵] با روشی مشابه، کارایی پردازنده‌های عمومی را افزایش می‌دهند. اما از آنجایی‌که طراحی چنین پردازنده‌هایی برای یک کاربرد هدف انجام می‌شود و امکان تغییر کاربرد پس از تولید آنها وجود ندارد، قابلیت انعطاف کمتری نسبت به پردازنده‌های قابل توسعه دارند. از طرف دیگر فرایند طراحی یک پردازنده با مجموعه دستورالعمل‌های سفارشی نیز فرایندی پر هزینه و زمان‌بر است [۱][۲].

عموما در پردازنده‌های قابل توسعه، امکان تعریف تعداد محدودی دستورالعمل جدید در پردازنده پیش‌بینی می‌شود. پس از تولید نهایی پردازنده قابل توسعه، امکان تعریف این دستورالعمل‌ها برای آن وجود دارد. نوع، تعداد و چگونگی تعریف دستورالعمل‌های جدید با توجه به پردازنده قابل توسعه مورد استفاده متفاوت است ولی روند شناسایی و تعریف دستورالعمل‌های جدید برای پردازنده‌های قابل توسعه یکسان است. برای افزایش کارایی برنامه‌های کاربردی در پردازنده‌های قابل توسعه، همواره تلاش می‌شود تا دستورالعمل‌های جدید طوری تعریف شوند که بخش‌های محاسباتی برنامه کاربردی مورد نظر را در واحدهای محاسباتی سفارشی اجرا کنند. بخش‌های

[5] Application Specific Instruction set Processors (ASIPs)

محاسباتی برنامه‌های کاربردی عموما شامل زنجیره‌ای از دستورالعمل‌های محاسباتی هستند که اجرای متوالی و یا موازی آنها بر روی سخت‌افزار شتاب‌دهنده امکان‌پذیر است. اجرای چنین زنجیره‌ای از دستورالعمل‌های پردازنده به‌صورت نرم‌افزاری زمان زیادی نیاز دارد. پردازنده‌های مشهوری مانند ALTERA NIOS [۵]، XTENSA [۶] و XILINX MICROBLAZE نمونه‌هایی از پردازنده‌های قابل توسعه هستند که با استفاده از تعریف دستورالعمل‌های جدید تلاش در بهبود کارایی پردازنده دارند.

افزودن قابلیت توسعه به مجموعه دستورالعمل‌های پردازنده پایه با استفاده از یک شتاب‌دهنده سخت‌افزاری دارای مزایای متعددی است. در زیر تعدادی از این مزیت‌ها نام برده شده است:

قابلیت برنامه‌ریزی شتاب‌دهنده سخت‌افزاری، امکان تغییر دستورالعمل‌های جدید را در زمان اجرا فراهم می‌کند. در پردازنده‌های قابل توسعه موجود عموما دستورالعمل‌های جدید در مرحله پیکربندی به پردازنده اضافه شده و در زمان اجرا، برنامه کاربردی بر اساس این واحدهای اجرایی جدید کامپایل می‌شود. به عبارت دیگر بر اساس برنامه کاربردی مورد نظر، می‌توان برنامه‌ریزی شتاب‌دهنده را حتی در زمان اجرا تغییر داد.

بخش‌های محاسباتی برنامه‌های کاربردی حوزه‌های مختلف به‌طور مثال پردازش سیگنال، رمزنگاری و غیره عموما شامل ساختارهای مشابه هستند که با توسعه مجموعه دستورالعمل‌های پردازنده برای بخشی از این کاربردها، می‌توان از آن در کاربردهای مشابه نیز استفاده کرد.

زمان توسعه یک پردازنده قابل توسعه برای کاربردی خاص در مقایسه با زمان توسعه یک پردازنده با مجموعه دستورالعمل‌های سفارشی بسیار کمتر است [۱]. علت اصلی این موضوع این است که عموما پردازنده‌ها با مجموعه دستورالعمل‌های سفارشی را به صورت تمام سفارشی تولید می‌کنند.

مهم‌ترین موضوع در فرایند توسعه مجموعه دستورالعمل‌های یک پردازنده در غیر خودکار بودن این فرایند و وابستگی ابزاری زیادی است که در آن وجود دارد. اجرای مناسب این فرایند اغلب طولانی‌تر و پر هزینه‌تر از طراحی تمام سفارشی است [۱]. در سیستم موجود XTENSA بخش عمده‌ای از مسئولیت فرایند توسعه مجموعه دستورالعمل‌های پردازنده بر عهده کاربران قرار گرفته است [۱][۴].

خودکار سازی این فرایند طراحی موضوعی است که موجب موفقیت چنین فرایند طراحی می‌شود.

برای رفع بخشی از مشکلات نام‌برده شده در طراحی پردازنده‌های قابل توسعه، در این پایان‌نامه یک هسته سخت‌افزاری ارائه داده شده است که به‌طور خودکار و هم‌زمان با اجرای برنامه کاربردی در پردازنده پایه قابلیت تشخیص دستورالعمل‌های سفارشی تعریف شده برای پردازنده را دارا است. برای اجرای دستورالعمل‌های سفارشی تشخیص داده شده در ضمن اجرای برنامه کاربردی نیز از یک شتاب‌دهنده با معماری دانه درشت[۶] استفاده شده است.

۱-۱-۲. یک‌پارچه سازی شتاب‌دهنده با پردازنده عمومی

همان‌طور که در بخش قبل عنوان شد، شتاب‌دهنده‌های سخت‌افزاری باعث افزایش کارایی برنامه‌های کاربردی می‌شوند ولی فرایند طراحی آنها مسائلی را به همراه دارد. مهم‌ترین مسئله موجود مربوط به هزینه‌های برگشت‌ناپذیر مهندسی[۷] فرایند طراحی شتاب‌دهنده است. فرایند افزودن یک شتاب‌دهنده سخت‌افزاری به یک پردازنده پایه بسیاری از مسائل مربوط به طراحی یک پردازنده جدید را دارد. در صورتی‌که طراحی پردازنده به صورت تمام سفارشی مورد نظر باشد، یک مجموعه ماسک جدید برای ساخت تراشه باید ساخته شود، تراشه باید از نظر ساختار، عملکرد و زمانی باید مورد آزمایش قرار گیرد و دستورالعمل‌های جدید نیز باید در مدل خط لوله پردازنده به درستی مدل شوند. از طرف دیگر برنامه کاربردی نیز باید بر اساس پردازنده جدید باز نویسی شود و یا با استفاده از کامپایلر جدید دوباره کامپایل شود.

برای کاهش هزینه‌های برگشت‌ناپذیر مهندسی موجود در طراحی پردازنده‌های قابل توسعه، در این پایان‌نامه یک بستر محاسباتی قابل باز پیکربندی پویا پیشنهاد شده است. این بستر محاسباتی، پس از تشخیص دستورالعمل‌های سفارشی، با استفاده از شتاب‌دهنده‌های قابل باز پیکربندی آنها را اجرا می‌کند. برای اجرای دستورالعمل سفارشی نیاز به دسترسی به اطلاعات موجود در پردازنده پایه است. در بستر محاسباتی پیشنهاد شده در این پایان‌نامه، تلاش شده است حداقل وابستگی بین شتاب‌دهنده و پردازنده پایه وجود داشته باشد. نکته دیگر موجود در بستر محاسباتی پیشنهادی این است که فرایند تشخیص و اجرای دستورالعمل‌های سفارشی به صورت کاملا مخفی از دید کاربر انجام می‌شود. به همین دلیل نیازی به تغییر روند توسعه برنامه کاربردی برای بستر محاسباتی پیشنهادی

[6] Coarse Grain

[7] Non-Recurring Engineering Costs

وجود ندارد و کاربران نهایی سیستم بدون نیاز به تغییر کدهای اجرایی خود توانایی اجرای آنها را بر روی بستر محاسباتی جدید دارند. از طرف دیگر نیازی به توسعه مجموعه دستورالعمل‌های پردازنده به صورت واقعی نیز در این بستر محاسباتی نیست و پردازنده استفاده شده در بستر محاسباتی پیشنهادی یک پردازنده معمولی است.

در چارچوب کاری پیشنهاد شده در این پایان‌نامه، شناسایی الگوهای مربوط به دستورالعمل‌های جدید به صورت غیر هم‌زمان با اجرای برنامه کاربردی انجام می‌شود. الگوهای مورد استفاده در این پایان‌نامه، نتایج ارائه شده در [۱] هستند که در زمان اجرای برنامه کاربردی بر روی پردازنده پایه، تشخیص داده شده و برای اجرا بر روی شتاب‌دهنده نگاشته می‌شوند. با استفاده از این چارچوب کاری، می‌توان از الگوریتم‌های پیچیده زمان کامپایل برای تعریف بهتر دستورالعمل‌های جدید برای سیستم بهره گرفت و با استفاده از اطلاعات موجود در زمان اجرا نیز فرایند تشخیص دستورالعمل‌های جدید را به‌صورت بهتری انجام داد.

۱-۱-۳. توسعه برنامه کاربردی

از دیگر موضوعات مهم در تجهیز یک پردازنده پایه به شتاب‌دهنده سخت‌افزاری، پشتیبانی نرم‌افزاری از دستورالعمل‌های جدید تعریف شده برای پردازنده پایه است. به عبارت دیگر پس از تعریف دستورالعمل‌های جدید برای پردازنده، باید کاربر نهایی سیستم قابلیت استفاده از این دستورالعمل‌ها را دارا باشد. روش‌های متفاوتی برای این منظور پیشنهاد شده است. در بهترین حالت، استفاده از دستورالعمل‌های جدید وظیفه کامپایلر است. در واقع کامپایلر مسئولیت تولید کد باینری جدید را با استفاده از مجموعه جدید دستورالعمل‌های پردازنده بر عهده دارد. ولی از آنجایی‌که طراحی کامپایلر جدید و یا تغییر کامپایلر برای پشتیبانی از دستورالعمل‌های جدید پیچیدگی‌های فراوانی را به همراه دارد، این کار نیز تا حدودی به‌صورت غیر خودکار انجام می‌شود.

یکی از اهداف بستر محاسباتی قابل باز پیکربندی پیشنهادی در این پایان‌نامه، مخفی کردن فرایند سفارشی شدن پردازنده از دید کاربر نهایی است. به همین دلیل در آن تلاش شده است تا نیازی به تغییر زنجیره نرم‌افزاری توسعه برنامه برای پردازنده وجود نداشته باشد. در این بستر محاسباتی پیشنهادی، برای اجرای برنامه‌های کاربردی تنها کافی است کد باینری مربوطه را بر روی سیستم بار گذاری کرد. زبان برنامه‌نویسی کد برنامه کاربردی، کامپایلر مورد استفاده و دیگر موارد مربوط به توسعه برنامه کاربردی از مواردی است که محدودیتی در آنها وجود ندارد.

۱-۲. ساختار پایان‌نامه

این پایان‌نامه در چهار فصل ارائه شده است. فصل اول مربوط به معرفی موضوع و چالش‌های موجود است. در فصل دوم مرور کارهای پژوهشی مرتبط انجام شده است و تلاش شده است تا مفاهیم اصلی موجود در محاسبات قابل باز پیکربندی نیز معرفی گردد. فصل سوم اختصاص به معرفی هسته تشخیص دستورالعمل سفارشی و نیز بستر محاسباتی قابل باز پیکربندی پویا دارد. در این فصل در مورد الگوریتم تشخیص دستورالعمل‌های جدید، معماری هسته سخت‌افزاری، معماری شتاب‌دهنده سخت‌افزاری و نیز نحوه ارتباط آن با پردازنده پایه بحث شده است. در فصل چهارم نتایج سنتز بخش‌های مختلف سخت‌افزار و نیز نتایج شبیه‌سازی تعدادی از برنامه‌های محک آورده شده است. در فصل پنجم نیز جمع‌بندی کوتاهی از کارهای انجام شده در این پایان‌نامه انجام شده است و تعدادی زمینه پژوهشی مرتبط به عنوان کارهای آینده معرفی شده است.