دانشکده مهندسی برق و کامپیوتر-گروه الکترونیک

پایان نامه

برای دریافت درجه کارشناسی ارشد در رشته الکترونیک

عنوان

پیاده‌سازی توابع اولیه بر روی FPGA

استاد راهنما

دکتر قادر کریمیان

استاد مشاور

دکتر غلامرضا حجتی

پژوهشگر

رحمت مینایی

شهریور ۱۳۹۱

بسم الله الرحمن الرحيم

تشکر

به نام آنکه قلم را بیافرید تا بنویسد آنچه را که باید نوشته شود و همچنین فهم و تفکر را مایه رستگاری بشر نمود. و سلام بر فرستاده‌اش (ص) که رحمتی برای عالمیان شد و انسانیت را به انسان نمایاند و درود بر آل و یارانش، که این پیغام عظیم را برسانیدند.

ابر و باد و مه و خورشید و فلک در کارند تا تو نانی به کف آری و به غفلت نخوری

قدردانی وسیله‌ای است که باعث می‌گردد انسان غِرّه نشود و آنچه را که به دست آورده فقط از خود نداند، با این حال از **مادر** مهربانِ فهیم و **پدر** دلسوزِ زحمت کشم سپاس گذارم. همچنین از استاد بزرگوارم، **دکتر قادر کریمیان** که در طی این مسیر، علاوه بر اشراف بر مسائل علمی از لحاظ اخلاقی نیز از مناعت طبع بلندی برخوردارند، بسیار سپاس گذارم. همچنین، از استاد مشاور گرانقدر، **دکتر غلامرضا حجتی**، تشکر دارم که خوش رویی و زنده دلی ایشان چراغ راه من هست. در انتها، از داور محترم جناب **دکتر ضیاءالدین کوزه‌کنانی** کمال قدردانی را دارم که وقت گرانبهای خود را صرف خواندن این پایان نامه کرده‌اند.

نام خانوادگی دانشجو: مینایی نام: رحمت

استاد راهنما: دکتر قادر کریمیان

استاد مشاور: دکتر غلامرضا حجتی

مقطع تحصیلی: کارشناسی ارشد رشته: مهندسی برق گرایش: الکترونیک دانشگاه: تبریز

دانشکده: مهندسی برق و کامپیوتر تاریخ فارغ التحصیلی: شهریور ۱۳۹۱ تعداد صفحه: ۸۹

کلید واژه: توابع پایه، تقریب چندجمله‌ای، پیاده‌سازی سخت‌افزاری، بازه‌بندی، FPGA، الگوریتم، تقریب minimax.

چکیده: محاسبات توابع پایه همچون تابع لگاریتمی، نمایی، مثلثاتی، 1/x و ...، که در DSP، پردازنده-های گرافیکی، سیستم‌های مخابراتی و ... به‌کار برده می‌شود، بسیار مورد استفاده قرار می‌گیرد. از اینرو محاسبه سریع و دقیق این توابع تاثیر زیادی بر روی عملکرد این سیستم‌ها دارد. اگر چه این توابع با استفاده از نرم‌افزارهایی با دقت بالا قابل محاسبه هستند ولی در بیشتر کاربردهایی که به محاسبه متعدد این توابع نیاز است این نرم‌افزارها بسیار کند عمل می کنند، از اینرو پیاده‌سازی این توابع و حتی توابع ترکیبی به صورت سخت‌افزاری نسبت به تحلیل نرم‌افزاری بویژه با پیشرفت تکنولوژی VLSI مورد توجه قرار گرفته است. پیاده‌سازی سخت‌افزاری بر حسب پارامترهای مورد نیاز با الگوریتم‌های مختلف صورت می‌گیرد. در این پایان‌نامه سعی می‌شود الگوریتم‌ها و روش‌های محاسبه توابع به صورت سخت‌افزار بررسی و مقایسه شوند سپس با انتخاب الگوریتم (بهینه) تقریب چندجمله‌ای تکه‌ای و با اعمال شرط پیوستگی در نقاط مرزی بین دو بازه مجاور بعضی از ضرایب را برای دو بازه مجاور یکسان در نظر می‌گیریم و این باعث می‌شود میزان حافظه مورد نیاز برای ذخیره ضرایب کاهش یابد، در آخر بلوک‌هایی برای محاسبه مستقیم توابع پایه بر اساس این تکنیک بر روی FPGA پیاده‌سازی می‌نماییم.

اختصارات

LUT	*look-up table*
FPGA	*Field-programmable Gate Array*
ROM	*Read Only Memory*
ULP	*Unit in the Last Place*
ASIC	*Application Specific Integrated Circuit*
DSP	*Digital Signal Processor*
CORDIC	*Coordinate Rotation Digital Computer*
SBTM	*Symmetric Bipartite Table Method*
STAM	*Symmetric Table Addition Method*
VLSI	*Very Large Scale Integration*
IC	*Integrated Circuit*
FB	*Fractional Bit*
IB	*Integer Bit*
MSB	*Most Significant Bit*
LSB	*Least Significant Bit*
DRR	*Dynamic Range Reduction*
CPM	*Composite Polynomial Method*
US	*Uniform segmentation*
MIP	*Mixed-Integer programming*

فهرست

۱) فصل اول: مقدمه ۱

۱-۱) مقدمه ۲

۲) فصل دوم: روش‌ها و الگوریتم تقریب توابع ۶

۲-۱) آشنایی با FPGA: ۷

۲-۱-۱) مقایسه FPGA و ASIC ۹

۲-۲) مراحل محاسبه توابع ۱۰

۲-۳) الگوریتم‌های تقریب تابع ۱۶

۲-۳-۱) الگوریتم جدول یا مقدار دهی مستقیم ۱۷

۲-۳-۲) روش‌های دو جدولی و چند جدولی ۱۸

۲-۳-۳) الگوریتم CORDIC ۲۱

۲-۳-۴) تقریب چندجمله‌ای ۲۳

۲-۳-۵) تقریب نسبت یا تقسیم ۲۵

۲-۳-۶) تقریب چندجمله‌ای تکه‌ای ۲۵

۲-۴) روش انتخاب الگوریتم مناسب برای طراحی ۲۷

۳) فصل سوم: بررسی روش‌های پیاده‌سازی با استفاده از تقریب چندجمله‌ای ۳۰

۳-۱) روش‌های بازه‌بندی ۳۱

۳-۲) تقریب تابع چندجمله‌ای تکه‌ای با استفاده از بازه‌بندی طبقه‌بندی شده ۳۴

۳-۲-۱) الگوریتم بازه‌بندی ۳۴

۳-۲-۲) سخت‌افزار لازم برای پیاده‌سازی بازه‌بندی طبقه‌بندی شده ۳۶

۳-۳) محاسبه تقریب تابع با استفاده از LUTهای سری شده ۳۹

۳-۳-۱) الگوریتم بازه‌بندی ۳۹

۳-۳-۲) سخت‌افزار مرحله محاسبه تابع ۴۱

۳-۳-۳) سخت‌افزار تعیین آدرس با استفاده از کسکد کردن جدول‌ها ۴۲

۳-۳-۴) مقایسه حافظه برای دو روش بازه‌بندی یکسان و کسکد کردن جداول ۴۳

۳-۳-۵) نتایج حاصل از پیاده‌سازی بر روی FPGA برای دو روش ۴۴

۳-۴) پیاده‌سازی با استفاده از چندجمله‌ای درونیاب ۴۵

۳-۴-۱) چندجمله‌ای درونیاب ۴۶

۳-۴-۲) سخت‌افزار برای پیاده‌سازی چندجمله‌ای درونیاب ۴۷

۳-۴-۳) نتایج پیاده‌سازی سخت‌افزاری چندجمله‌ای درونیاب ۴۹

۳-۵) بررسی سایر روش‌ها برای پیاده‌سازی چندجمله‌ای تکه‌ای ۵۱

۴) فصل چهارم: پیاده‌سازی سخت‌افزاری توابع پایه با استفاده از تقریب چندجمله‌ای تکه‌ای محدود شده ۵۴

۴-۱) تعریفات ۵۴

۴-۲) تقریب چندجمله‌ای محدود شده ۵۶

۴-۳) خطای ذاتی در چندجمله‌ای محدود شده ۵۸

۴-۳-۱) تقریب minimax ۵۹
۴-۳-۲) پیوستگی تابع تقریب ۶۱
۴-۳-۲-۱) درجه چندجمله‌ای N فرد ۶۱
۴-۳-۲-۲) درجه چندجمله‌ای N زوج ۶۳
۴-۳-۳) پیوستگی مشتقات تابع ۶۵
۴-۴) محاسبه ضرایب چندجمله‌ای ۶۷
۴-۴-۱) کوانتزاسیون ضرایب ۷۰
۴-۴-۲) انتخاب طول بیت بهینه برای ضرایب ۷۵
۴-۵) پیاده‌سازی سخت‌افزاری ۷۹
۴-۵-۱) آنالیز خطا ۸۳
۴-۶) نتایج پیاده‌سازی ۸۳

فهرست اشکال

شکل ۲-۱: ساختار کلی داخلی یک FPGA. [۳] ۸

شکل ۲-۲: یک نمای کلی از سلول منطقی (LC) در Xilinx FPGA. [۳] ۹

شکل ۲-۳: توصیف نرم‌افزاری سه مرحله محاسبه تابع $\sin(x)$. [۳] ۱۳

شکل ۲-۴: سخت‌افزار مربوط به سه مرحله محاسبه تابع $\sin(x)$. [۳] ۱۴

شکل ۲-۵: ساختار روش مقداردهی مستقیم. [۲] ۱۷

شکل ۲-۶: بلوک دیاگرام روش دوجدولی. [۲] ۱۹

شکل ۲-۷: بلوک دیاگرام الگوریتم CORDIC. [۲] ۲۲

شکل ۲-۸: بعضی از روش‌های تقریب برای یک پارامتر مشخص در دقت‌های مختلف، بهتر می‌باشند. [۸] ۲۷

شکل ۲-۹: مقایسه روش‌های مختلف تقریب تابع نسبت به بلوک‌های CLB در FPGA. [۲۵] ۲۸

شکل ۳-۱: بازه‌بندی یکسان برای تابع $\sqrt{x}$ در بازه [1 , 0]. [۶] ۳۲

شکل ۳-۲: بازه‌بندی غیریکسان برای تابع تابع $\sqrt{x}$ در بازه [1 , 0]. [۶] ۳۳

شکل ۳-۳: چهار طرح بازه‌بندی در بازه‌بندی طبقه‌بندی شده. [۴] ۳۴

شکل ۳-۴: بازه‌بندی طبقه‌بندی شده بر روی توابع f_1 تا f_6 برای دقت 2^{-14}. [۴] ۳۵

شکل ۳-۵: ساختار سخت‌افزار برای محاسبه توابع بازه‌بندی شده به روش طبقه‌بندی شده. [۴] ۳۷

شکل ۳-۶: مقایسه مساحت تقریب درجه ۲ برای شش تابع. [۴] ۳۷

شکل ۳-۷: مقایسه تاخیر تقریب درجه ۲ برای شش تابع. [۴] ۳۸

شکل ۳-۸: مقایسه مساحت بازه‌بندی یکسان و طبقه‌بندی شده برای $f_5 = \ln(1+x)$. [۴] ۳۸

شکل ۳-۹: مراحل الگوریتم بازه‌بندی غیر یکسان برای تابع $\sqrt{x}$. [۶] ۴۱

شکل ۳-۱۰: دو ساختار برای محاسبه توابع. (a) ساختار بازه‌بندی یکسان. (b) ساختار بازه‌بندی غیریکسان. [۶] ۴۲

شکل ۳-۱۱: مدار تعیین شماره بازه براساس کسکد کردن جدول‌ها. [۶] ۴۳

شکل ۳-۱۲: نمایی کلی از مراحل محاسبات در چندجمله‌ای درونیاب و تقریب. [۱۵] ۴۵

شکل ۳-۱۳: سخت‌افزار استخراج مقادیر تابع در چندجمله‌ای درونیاب درجه ۱ و ۲ با استفاده از ROM تک پورتی. [۱۵] ۴۸

شکل ۳-۱۴: سخت‌افزار استخراج مقادیر تابع در چندجمله‌ای درونیاب درجه ۱ و ۲ با استفاده از ROM دوپورتی. [۱۵] ۴۸

شکل ۳-۱۵: مدار محاسبه ضرایب و چندجمله‌ای در روش درونیاب. [۱۵] ۴۹

شکل ۳-۱۶: تغییرات مساحت در روش‌های تقریب/درونیابی. [۱۵] ۵۱

شکل ۳-۱۷: تغییرات تاخیر در روش‌های تقریب/درونیابی. [۱۵] ۵۱

شکل ۴-۱: در تقریب چندجمله‌ای تکه‌ای محدود شده دو بازه مجاور یک جفت بازه را تشکیل می‌دهند. ..۵۷

شکل ۴-۲: خطای برای تقریب چندجمله‌ای درجه ۳ استاندارد برای $f(x) = 2^x$ در بازه $[0,1]$. ۶۲

شکل ۴-۳: خطای تقریب چندجمله‌ای درجه ۳ برای تابع $f(x) = 2^x$ در بازه $[0,1]$ با اعمال شرط پیوستگی در نقطه وسط بازه. ۶۳

شکل ۴-۴: خطا برای تقریب چندجمله‌ای درجه دوم استاندارد برای تابع $f(x) = 2^x$ در بازه $[0,1]$. ۶۴

شکل ۴-۵: خطا در تقریب چندجمله‌ای درجه ۲ برای تابع $f(x)=2^x$ در بازه $[0,1]$ با شرط پیوستگی در نقطه وسط جفت بازه. ۶۵

شکل ۴-۶: : خطا در تقریب چندجمله‌ای درجه ۲ برای تابع $f(x)=2^x$ در بازه $[0,1]$ با شرط پیوستگی مشتق اول در نقطه وسط جفت بازه. ۶۶

شکل ۴-۷: خطا در تقریب چندجمله‌ای درجه ۳ برای تابع $f(x)=2^x$ در بازه $[0,1]$ با شرط پیوستگی مشتق اول. ۶۷

شکل ۴-۸: خطا در تقریب چندجمله‌ای درجه ۳ برای تابع $f(x)=2^x$ در بازه $[0,1]$ با شرط پیوستگی مشتق دوم. ۶۷

شکل ۴-۹: الگوریتم تعیین بهینه طول بیت‌های کسری ضرایب چندجمله‌ایها. [۱۰] ۷۶

شکل ۴-۱۰: مثال برای جواب‌های بدست آمده از الگوریتم انتخاب طول بیت بهینه ضرایب. [۱۰] ۷۷

شکل ۴-۱۱: پیاده‌سازی تقریب چندجمله‌ای تکه‌ای درجه ۲ محدود شده به روش مستقیم. ۸۱

شکل ۴-۱۲: پیاده‌سازی تقریب چندجمله‌ای تکه‌ای درجه ۲ محدود شده به روش هورنر. ۸۲

شکل ۴-۱۳: سخت‌افزار تقریب تکه‌ای خطی محدود شده. ۸۳

فهرست جداول

جدول ۲-۱: بازه‌های ورودی و خروجی برای چند تابع پایه. ۱۵

جدول ۲-۲: ماکزیمم و متوسط خطا چند نوع تقریب درجه اول تابع e^x در بازه $[-1\ ,\ 1]$. ۲۴

جدول ۲-۳: میزان خطا و درجه چندجمله‌ای برای تقریب تابع $\sin(x)$ در کل بازه با یک چندجمله‌ای. [۱] ۲۶

جدول ۲-۴: : میزان خطا و درجه چندجمله‌ای برای تقریب تابع $\sin(x)$ باتقسیم بازه اصلی به ۲ زیربازه مساوی. [۱] ۲۶

جدول ۲-۵: میزان خطا و درجه چندجمله‌ای برای تقریب تابع $\sin(x)$ باتقسیم بازه اصلی به ۴ زیربازه مساوی. [۱] ۲۶

جدول ۲-۶: مقایسه تعداد کلاک‌ها برای همه روش‌های تقریب در دقت‌های مختلف. [۲۵] ۲۹

جدول ۳-۱: مقایسه تعداد بازه‌ها برای دو روش بازه‌بندی یکسان و طبقه‌بندی شده. [۴] ۳۶

جدول ۳-۲: تابع بازه‌ها برای تعیین شماره بازه‌ها. ۴۲

جدول ۳-۳: مقایسه حافظه مورد نیاز برای بازه‌بندی به روش کسکد کردن جداول و یکسان. [۶] ۴۴

جدول ۳-۴: نتایج حاصل از پیاده‌سازی در FPGA در دو روش بازه‌بندی با کسکد کردن جداول و بازه‌بندی یکسان. [۶] ۴۴

جدول ۳-۵: مقایسه تعداد بازه‌ها و حافظه لازم در دو روش درونیابی و تقریب. [۱۵] ۵۰

جدول ۴-۱: نتایج حاصل از تکنیک کوانتیزاسیون ضرایب. [۱۰] ۷۳

جدول ۴-۲: اندازه Rom برای تقریب تکه‌ای خطی. ۷۷

جدول ۴-۳: میزان ROM برای تقریب تکه‌ای درجه ۲. ۷۸

جدول ۴-۴: نتایج حاصل از پیاده‌سازی بر روی FPGA برای تقریب درجه ۲. ۸۴

جدول ۴-۵: نتیجه حاصل از پیاده‌سازی برای تقریب خطی تابع e^x. ۸۴

جدول ۴-۶: مقایسه نتیجه حاصل از پیاده‌سازی تقریب درجه ۲ به روش هورنر. ۸۵

۱) فصل اول

مقدمه

۱-۱) مقدمه

امروزه همه ما تا حدودی با توابع ریاضی آشنایی داریم و به طور معمول در محاسبات و در کارهای روزمره به عنوان یک کاربر حرفه‌ای که با روابط ریاضی سر و کار دارد یا چه به عنوان یک فرد معمولی در بسیاری از موارد، استفاده می‌نماییم. در ریاضیات تابع عملکردی است که برای هر ورودی داده شده یک خروجی منحصر بفرد تولید می‌کند. توابع را می‌توان به دو دسته تقسیم نمود، توابع پایه[1] و توابع ترکیبی[2]. از توابع پایه می‌توان به تابع لگاریتم، تابع نمایی، تابع وارون، توابع مثلثاتی و ..... اشاره کرد. توابع ترکیبی از ترکیب چند تابع پایه تشکیل می‌شوند مانند $\sqrt{\ln(x)}$ که از دو تابع $\ln(x)$ و $\sqrt{x}$، تشکیل شده است. بسیاری از ما روزانه به طور متعدد این توابع را با استفاده از ماشین‌حساب، کامپیوتر یا با استفاده از نرم‌افزارهای MATLAB، C و .... محاسبه نموده‌ایم ولی شاید تا به حال به چگونگی محاسبه این توابع توسط این نرم‌افزارها یا وسایل فکر نکرده باشیم. جالب این است که بدانیم محاسبه‌ی توابع پایه در کاربردهای حرفه‌ای‌تر همچون پردازنده‌های DSP[3]، پردازنده‌های گرافیکی و سیستم‌های مخابراتی و ..... بسیار مورد استفاده قرار می‌گیرد.

نرم‌افزارهای متلب، C و سایر نرم‌افزارهای دیگر ریاضی کتابخانه‌های برای محاسبه این توابع با دقت بالا فراهم می‌نمایند. محاسبه توابع با استفاده از نرم‌افزار، اگرچه با دقت بالایی قابل محاسبه است ولی از آنجاییکه محاسبه این توابع در نرم‌افزارها با استفاده از پردازنده‌هایی که براساس یکسری دستورالعمل که برای آنها تعریف شده است، انجام می‌شود در مقایسه با پیاده‌سازی سخت‌افزار به طور معمول بسیار کند

[1] Elementary functions
[2] Compound functions
[3] Digital Signal Processor

عمل می‌کنند در حالیکه در بعضی از کاربردها لازم است این توابع برای دفعات متعدد محاسبه شوند. برای مثال ۶۰٪ از کل زمان اجرا یک برنامه در موتور جت صرف عملیات محاسبه تابع می‌شود [11]. یک مثال خوب دیگر برای مقایسه سخت‌افزار و نرم‌افزار این است که اگر ما در کامپیوتر خود یک ویدیو کارت خوب نداشته باشیم و از یک SVGA [1]معمولی استفاده نماییم خواهیم دید زمانی که بخواهیم یک پنجره را drag کنیم چقدر کند عمل خواهد نمود و این بدین خاطر است که محاسبه پارامترهای (میزان نور، مکان و ...) پیکسل‌ها توسط CPU انجام می‌شوند و این در حالی است هنگامی که از یک ویدیو کارت خوب استفاده نماییم محاسبه پارامترهای مربوط به پیکسل‌ها توسط پردازنده‌های گرافیکی که در آنها بلوک‌هایی جهت محاسبه توابع به صورت سخت‌افزار تعبیه شده است، صورت می‌گیرد که بسیار سریعتر است و همچنین باعث می‌شود که CPU آزاد باشد. همچنین پردازنده‌های گرافیکی بایستی بتوانند میلیاردها بار محاسبه توابع را در هر ثانیه اجرا کنند. پردازنده‌های گرافیکی پیشرفته‌تر نرخ rendring بالا با انعطاف‌پذیری بالا را با هم می‌پیوندند و نیاز می‌شود تا الگوریتم‌های سه بعدی پیشرفته و متعددی را پشتیبانی کند با این هدف واحد پردازش گرافیکی پردازنده‌های قدرتمند قابل برنامه‌ریزی دارد که به عنوان shader شناخته می‌شوند. این Shaderها نه‌تنها عملیات جمع و ضرب را به صورت سخت‌افزاری انجام می‌دهد بلکه بلوک‌هایی برای محاسبه توابع مقدماتی که در بالا ذکر شده‌اند و در بسیاری از کارهای پردازش تصویر همچون نرمالیزه‌کردن بردارها، محاسبه فاصله، میزان نور و رنگ پیکسل‌ها مورد استفاده قرار می‌گیرند، در خود دارد. از اینرو با پیشرفت تکنولوژی VLSI[2] و نیاز به سرعت‌های بالاتر، تلاش جهت طراحی بلوک‌هایی برای محاسبه مستقیم توابع به صورت سخت‌افزاری بشدت مورد توجه قرار گرفته است. پیاده‌سازی بصورت سخت‌افزار بدین معنی است که یک ماشین واحد یک تابع را به ازای ورودی داده شده محاسبه می‌نماید.

توابع پایه‌ای همچون sin، cos، تابع نمایی و تابع لگاریتمی در الگوریتم‌های پردازنده‌های DSP مورد استفاده قرار می‌گیرند. در یک تولید کننده اعداد تصادفی، با یک توزیع گوسی در صورتی که از روش Box-

[1] Super Video Graphics Array
[2] Very Large Scale Integration

Muller ساخته شده باشد به توابعی همچون لگاریتم، ریشه دوم، و sin و cos نیاز دارد. همچنین توابع ترکیبی $f(x) = \log(1 + 2^x)$ و سایر توابع پیچیده‌تر در LNS[1] مورد استفاده قرار می‌گیرند.

همانطور که اشاره شد اگرچه پیاده‌سازی سخت‌افزاری توابع به محاسبه سریعتر توابع منجر می‌شود ولی از آنجاییکه در سخت افزار محدودیت نشان دادن اعداد (عرض بیت) داریم و دقت‌های بسیار بالا مساحتی بزرگی از سخت‌افزار را می‌طلبد بنابراین دست یافتن به دقت‌های بالا به اندازه نرم‌افزار بسیار پرهزینه و کم بازده خواهد بود. همچنین پیاده‌سازی سخت‌افزاری باعث اشغال شدن بخشی از مساحت آی سی خواهد شد ولی این مشکل با پیشرفت تکنولوژی VLSI که پیاده‌سازی مدارهای بسیار پیچیده و ساخت حافظه‌های بزرگ را بر روی یک آی سی امکان ساخته است، می‌تواند در قبال سرعت بالایی که بدست می‌دهد قابل چشم‌پوشی باشد. در طراحی مدارهای سخت‌افزاری باید تعادل بین پارامترهای مورد نیاز مساله برقرار شود. از جمله این پارامترها می‌توان به: ۱. سرعت (تاخیر) ۲. دقت ۳. حافظه و مساحت اشغال شده بر روی IC و در بعضی موارد توان مصرفی اشاره کرد.

محاسبه توابع به صورت نرم‌افزار و سخت‌افزار با استفاده از الگوریتم‌های متعددی انجام می‌شود ولی از آنجاییکه که هدف ما پیاده‌سازی به‌صورت سخت‌افزار می‌باشد تنها در رابطه با آن دسته از الگوریتم‌ها که بیشتر در پیاده‌سازی سخت‌افزار مورد استفاده قرار می‌گیرند بحث خواهیم نمود.

پیاده‌سازی سخت‌افزاری برای مدارهای دیجیتال بیشتر بر روی آی سی‌های FPGA[2] یا بصورت ASIC[3] صورت می‌پذیرد. هدف ما این است که پیاده‌سازی را بر روی FPGA انجام دهیم بنابراین در بخش اول فصل دوم مقدمه‌ای بر این آی سی و مقایسه آن با آی سی‌های ASIC خواهیم داشت . به طور کلی محاسبه تابع در سه مرحله صورت می‌گیرد ولی تمرکز ما بیشتر بر روی مرحله تقریب که اساسی‌ترین و مهمترین مرحله محاسبه توابع است، خواهد بود و بررسی با تفصیل کلیه مراحل از حوصله ای یک پایان‌نامه خارج است بنابراین در بخش ۲ فصل دوم شرح مختصری بر روی مرحله اول و سوم خواهیم داشت. در بخش

[1] Logarithm Number System
[2] Field Programmable Gate Array
[3] Application Specific Integrated Circuit

۳ فصل دوم که مهمترین بخش این فصل می‌باشد مروری کلی بر الگوریتم‌های تقریب (محاسبه) تابع و بررسی اجمالی سخت‌افزار مربوط به این الگوریتم‌ها خواهیم داشت. در فصل سوم این پایان‌نامه مروری بر کارهای که قبلا صورت گرفته است خواهیم داشت. لازم به ذکر است از آنجاییکه ما الگوریتم تقریب با استفاده از چندجمله‌ای تکه‌ای را انتخاب خواهیم نمود تنها به بررسی منابعی که از این الگوریتم برای تقریب تابع استفاده نموده‌اند خواهیم پرداخت. و البته فصل سوم نه تنها بررسی چند منبع بلکه بررسی چند روش کلی از تقریب می‌باشد که محور اصلی سایر منابع دیگر نیز می‌باشند بنابراین در این فصل به تفصیل به ارائه این سه روش و نتایج حاصل از آنها می‌پردازیم. در فصل چهارم روش پیشنهادی پایان‌نامه توضیح داده شده است. بخش اول این فصل به ارائه چند تعریف مهم که در طول فصل با آن روبه‌رو خواهیم شد اختصاص داده شده است. در بخش ۲ چگونگی اعمال تکنیک بکارگرفته شده در پایان‌نامه توصیف می‌شود. در بخش ۳ میزان خطای تقریب ناشی از اعمال تکنیک بحث خواهد شد. در بخش ۴ روش محاسبه ضرایب و کوانتیزاسیون ضرایب چندجمله‌ایها ارائه شده است. در بخش آخر نیز پیاده‌سازی سخت‌افزاری بر روی FPGA و بررسی نتایج گزارش می‌شود.

۲) فصل دوم

روش‌ها و الگوریتم‌های تقریب توابع

پیاده‌سازی به صورت سخت‌افزاری مدارها و المان‌هایی را می‌طلبد که بایستی طراحی شوند. خوشبختانه امروزه آی‌سی‌هایی وجود دارد که بیشتر بلوک‌های مورد نیاز به طور بهینه در آنها تعبیه شده‌اند. از جمله‌ی این آی‌سی‌ها FPGA می‌باشد. در بخش اول این فصل به بررسی این آی‌سی می‌پردازیم. در بخش دوم از سه مرحله اصلی محاسبه یک تابع دو مرحله را با شرح مثال توضیح خواهیم داد. بخش سوم این فصل را نیز به الگوریتم‌های تقریب یک تابع که مرحله دوم محاسبه یک تابع و اساس‌ترین آن نیز می‌باشد اختصاص داده‌ایم.

۲-۱) آشنایی با FPGA:

FPGA یک آی‌سی کاملا قبل برنامه‌ریزی می‌باشد. این آی‌سی را به هر مدار منطقی دلخواهی که بخواهیم می‌توان تبدیل کرد. با FPGA می‌توان یک میکروپروسسور یا یک فیلتر FIR[1] ساخت. می‌توان برای محاسبه FFT[2] با سرعت زیاد از آن استفاده کرد. ساختار داخلی FPGA به نحوی است که می‌توان بیشتر مدارهای دیجیتال را بر روی آن پیاده‌سازی کرد. عملیاتی که روی FPGA انجام می‌شود تا تبدیل به یک مدار خاص شود را پیکربندی[3] می‌گویند. FPGA بسیار انعطاف‌پذیر است و به راحتی می‌توان جایگزین بسیاری از مدارها شود. در واقع تمام اجزایی که زمانی یک بورد بزرگ را می‌ساختند اکنون با پیشرفت صنعت نیمه‌هادی در یک آی‌سی کوچک خلاصه می‌شوند.

[1] Finite Impulse Response
[2] Fast Fourier Transform
[3] Configuration