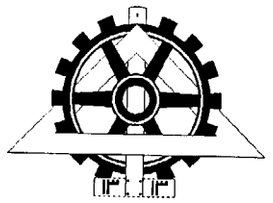
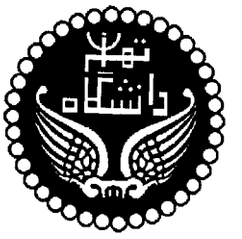
مرکز اطلاعات و مدارک علمی ایران
تهیه مدارک





بنام خدا

دانشگاه تهران، دانشکده فنی

گروه مهندسی برق و کامپیوتر

۲۰ /۱۱/ ۱۳۷۲

پایان‌نامه برای دریافت درجه کارشناسی ارشد

رشته مهندسی کامپیوتر، گرایش معماری کامپیوتر

شبیه‌سازی خطای مدارهای دیجیتال با استفاده از منطق فازی

امیررضا عطارها

استاد راهنما: دکتر مهرداد نورانی

استاد مشاور: دکتر کارولوکس

دی ۱۳۷۷

1320/2

۲۳۰۲۸

شبیه‌سازی خطای مدارهای دیجیتال با استفاده از منطق فازی

توسط

امیررضا عطارها

پایان‌نامه برای دریافت درجه کارشناسی ارشد

رشته مهندسی کامپیوتر، گرایش معماری کامپیوتر

از این پایان‌نامه در تاریخ ۱۳۷۷/۱۰/۲۷ در مقابل هیئت داوران

دفاع بعمل آمده و مورد تصویب قرار گرفت

محل امضاء

سرپرست تحصیلات تکمیلی دانشکده فنی: دکتر محمدعلی بنی‌هاشمی

مدیر گروه آموزشی: دکتر محمود کمره‌ای

سرپرست تحصیلات تکمیلی: دکتر محمود کمره‌ای

استاد راهنما: دکتر مهرداد نورانی

استاد مشاور: دکتر کارولوکس

عضوهیئت داوران: دکتر زین‌العابدین نوابی

عضوهیئت داوران: دکتر سید مهدی فخرایی

عضوهیئت داوران: دکتر کامبیز بدیع

۲۳۰۲۸

چکیده

در این پایان‌نامه، روش جدیدی برای شبیه‌سازی خطاهای غیرمرسوم در سطح گیت عرضه می‌شود. روش مورد نظر علاوه بر خطاهای تک-اتصال ایده‌آل قادر به آشکارسازی خطاهای غیرمرسوم تک-اتصال واقعی و پل نیز می‌باشد. این خطاهای غیرمرسوم در مدارهای CMOS منجر به ولتاژهای میانی می‌گردند. برای بررسی دقیق اثر این خطاها در مدار، در نظرگیری ولتاژهای آستانه متفاوت گیت‌های منطقی و عدم تقارن گیت‌های منطقی ضروری است. برای دستیابی به چنین ملزوماتی، از منطق فازی برای مدلسازی و شبیه‌سازی بهره گرفته شده است. برای خطاهای پل و تک-اتصال واقعی مدل خطای مقاومتی در نظر گرفته شده است. در این پایان‌نامه ابتدا با مدلسازی گیتهای منطقی بصورت بلوکهای فازی، یک شبیه‌ساز شبه‌آنالوگ عرضه می‌گردد. برای طراحی، شبیه‌ساز شبه‌آنالوگ ابتدا اطلاعات مربوط به گیت‌های موجود در کتابخانه مورد نظر با استفاده از شبیه‌ساز SPICE استخراج می‌شود و سپس با بهره‌گیری از دانش تجربی و الگوریتم بهینه‌سازی حداقل مربعات خطا بلوکهای فازی طراحی می‌گردند. سپس یک شبیه‌ساز برای بررسی خطاهای غیرمرسوم ارایه می‌شود. شبیه‌سازی خطا در سطح گیت عمل می‌نماید و بر اساس روش انتشار تک-خطا می‌باشد. لازم بذکر است بعد از تدارک دیدن بلوکهای فازی هیچ شبیه‌سازی در سطح ترانزیستور انجام نمی‌پذیرد. بنابراین سرعت بالای شبیه‌سازی و دقت بالا بطور توام در اختیار قرار می‌گیرد. سپس یک تولیدکننده جدید بردار آزمون بر اساس الگوریتم PODEM ارایه می‌گردد.بالاخره پوشش خطا برای مدارهای ISCAS Benchmark و یک ضرب‌کننده ۱۷×۱۷ بیتی عرضه می‌شود. نتایج بیانگر کارآیی امیدوارکننده روش مورد نظر می‌باشد.

فهرست مطالب

۱ مقدمه ۱

۱-۱ دیباچه ۱

۱-۲ مراحل تکوین یک تراشه با استفاده از ابزار CAD ۲

۱-۳ چرا شبیه‌ساز خطا با استفاده از منطق فازی؟ ۱۳

۲ آزمون سیستم‌های دیجیتال ۱۶

۲-۱ مقدمه ۱۶

۲-۲ آزمون و تشخیص عیب ۱۶

۲-۳ انواع روش‌های آزمون ۱۷

۲-۳-۱ روش آزمون پویش ۱۷

۲-۳-۲ روش آزمون خودآزمایی ۱۷

۲-۳-۳ روش آزمون IDDQ ۱۸

۲-۳-۴ روش آزمون براساس هسته‌های قابل جایدهی در سیستم ۱۸

۲-۴ مدلسازی خطا ۱۹

۲-۵ مدل‌های خطا ۱۹

۲-۵-۱ مدل خطای تک-اتصال ۲۰

۲-۵-۲ مدل خطای پل ۲۰

۲-۵-۳ مدل خطای باز ۲۷

۲-۵-۴ مدل خطای تاخیر ۲۸

۲-۶ آشکارسازی خطا ۲۹

۲-۷ ابزارهای مرسوم آزمون ۳۰

۲-۷-۱ تولیدکننده بردار آزمون ۳۰

۲-۷-۲ شبیه‌ساز خطا ۳۱

۲-۸ خلاصه و مطالعات بیشتر ۳۲

۳ مفاهیم منطق فازی ۳۳

۳-۱ مقدمه ۳۳

۳-۲ مجموعه‌های فازی و توابع عضویت ۳۵

۳-۳ عملگرهای مجموعه‌های فازی ۳۵
۳-۳-۱ تعریف اجتماع فازی ۳۵
۳-۳-۲ تعریف اشتراک فازی ۳۶
۳-۴ روابط فازی ۳۷
۳-۴-۱ رابطه فازی دودویی ۳۷
۳-۴-۲ ترکیب روابط فازی ۳۸
۳-۵ متغیرهای زبانی ۳۸
۳-۶ قواعد فازی IF-THEN ۴۰
۳-۷ منطق فازی و استدلال تقریبی ۴۰
۳-۷-۱ قاعده استنتاج وضع مقدم ۴۰
۳-۷-۲ قاعده استنتاج وضع مقدم تعمیم یافته ۴۱
۳-۸ سیستم‌های استنتاجی فازی ۴۱
۳-۸-۱ پایگاه قواعد فازی ۴۲
۳-۸-۲ موتور استنتاج فازی ۴۳
۳-۸-۳ فازی کننده ۴۵
۳-۸-۴ دی‌فازی کننده ۴۵
۳-۹ انواع سیستم‌های استنتاجی فازی ۴۶
۳-۹-۱ مدل فازی ممدانی ۴۶
۳-۹-۲ مدل فازی TSK (سوگنا) ۴۸
۳-۱۰ خلاصه و نتیجه‌گیری ۴۹

۴ شبیه‌سازی شبه آنالوگ مدارهای دیجیتال با استفاده از منطق فازی ۵۰
۴-۱ مقدمه ۵۰
۴-۲ شبیه‌سازی دقیق گیتهای منطقی با استفاده از منطق فازی ۵۱
۴-۲-۱ تهیه پایگاه داده ۵۲
۴-۲-۲ طراحی گیتهای منطقی با بکارگیری روش حداقل مربعات غیرخطی ۵۳
۴-۳ شبیه‌سازی شبه‌آنالوگ مدارهای دیجیتال ۵۸
۴-۴ نتایج تجربی ۵۹
۴-۵ خلاصه و نتیجه‌گیری ۶۲

۵ شبیه سازی خطاهای غیرمرسوم در مدارهای VLSI ۶۳
۵-۱ مقدمه ۶۳
۵-۲ مدلسازی خطا ۶۵
۵-۲-۱ محاسبه مقادیر ولتاژ دو سر خطاهای پل ۶۶
۵-۲-۲ تفسیر مقادیر ولتاژ دو سر خطاهای پل ۶۶
۵-۳ شبیه‌سازی خطا ۶۸
۵-۴ الگوریتم تولید بردار آزمون برای خطاهای پل ۷۱
۵-۵ نمایش نحوه عملکرد الگوریتم اصلاح‌شده PODEM ۷۳
۵-۵-۱ تولید بردار آزمون برای خطای پل رخ داده در ورودی گیت دو ورودی ۷۳
۵-۵-۲ تولید بردار آزمون برای خطای پل رخ‌داده در ورودی گیت با چندین ورودی ۷۶
۵-۶ نتایج تجربی ۷۷
۵-۷ خلاصه و نتیجه‌گیری ۸۱

۶ مطالعه موردی: انجام شبیه‌سازی خطا روی مدار ضرب کننده ۱۷×۱۷ بیتی با علامت ۸۲
۶-۱ مقدمه ۸۲
۶-۲ ساختار ضرب کننده ۱۷×۱۷ بیتی با علامت مورد استفاده در پردازنده‌های DSP
۶-۲-۱ مقدمه ۸۲
۶-۳ نتایج شبیه‌سازی خطا روی مدار ضرب کننده ۸۴
۶-۴ خلاصه و نتیجه‌گیری ۸۷

۷ نتیجه‌گیری و پیشنهادها ۸۸
۷-۱ خلاصه و نتیجه‌گیری ۸۸
۷-۲ پیشنهادها و قابلیتهای گسترش ۹۰

ضمیمه الف- پیاده سازی ضرب کننده ۱۷×۱۷ بیتی ۹۳
الف-۱ مقدمه ۹۳
الف-۲ معماری ضرب کننده ۹۴

الف-۳ ساختار جمع کننده ۹۹
الف-۴ راهکار پیاده‌سازی ۱۰۰
الف-۵ نتایج تجربی ۱۰۲

مراجع ۱۰۴

The most beautiful thing we can experience is the mysterious,
It is the source of all true art and science

Albert Einstein
What I Believe, 1930

۱ مقدمه

۱ -۱ دیباچه

اولین مدار مجتمع در سال ۱۹۶۱ چهارده سال بعد از اختراع اولین ترانزیستور به بازار عرضه گردید. این مدار شامل دو ترانزیستور و هشت مقاومت بود که بصورت فلیپ‌فلاپ ساماندهی گردیده بود [RiHF97]. امروزه همان مدار با نرخ ساعتی ۵۰۰ برابر سریعتر کار می‌کند در حالیکه با بیش از یک میلیون عنصر دیگر در یک تراشه قرار گرفته است و در حدود یک میلیونیم قیمت اولین مدار مجتمع (که تقریبا یکصد دلار بفروش رفت) می‌تواند به بازار عرضه گردد [DuER97].

تکامل قابل ملاحظه‌ی تکنولوژی سیلیکن طراحی سیستم‌های الکترونیکی‌ای را که چهل سال پیش قابل تصور هم نبودند ممکن ساخته است. ادامه رشد این تکنولوژی حداقل تا دهه‌ی آینده مسلم است و انتظار می‌رود قادر به جایدهی ۹۰ میلیون ترانزیستور با نرخ ساعت در حدود ۹۰۰ مگاهرتز برروی یک تراشه گردد [YoSh97].

اگرچه فشردگی مدارهای مجتمع موجب کاهش هزینه ساخت می‌گردد اما مشکلاتی را نیز در زمینه آزمون تراشه‌ها بوجود می‌آورد. برای نمونه در سطح پیچیدگی‌ای که قرار است ده میلیون ترانزیستور برروی یک تراشه تحقق یابند، ایجاد، جایگذاری و برقراری ارتباط مربوط به بیش از صدمیلیون الگو باید بدون خطا انجام گردد.

سازندگان تراشه‌های VLSI[1] سعی بسیار در تولید محصولاتی با درجه‌ی yield بالا دارند اما هنوز هیچ کارخانه‌ای نمی‌تواند بعد از تولید تراشه از کارکرد صحیح تراشه اطمینان کامل داشته باشد. بنابراین پس از تولید تراشه باید فرآیند پیچیده‌ای جهت آزمون تراشه انجام پذیرد تا محصول کاملا سالم به بازار عرضه گردد.

بعلت پیچیدگی فرآیند آزمون تراشه، ابزارهای CAD[2] متعددی جهت تسریع و پیشرفت این روند تهیه گردیده است که در مراحل مختلف طراحی تراشه، نقش موثری ایفا می‌نمایند. یکی از این ابزارها شبیه‌سازهای خطا می‌باشند.

در ادامه، مراحل تکوین یک مدار مجتمع مورد بررسی قرار می‌گیرد و جایگاه آزمون در مراحل مختلف طراحی نمایش داده می‌شود.

۱ -۲ مراحل تکوین یک تراشه با استفاده از ابزار CAD

پیشرفت‌های شگرف در ابزارهای CAD منجر به سهولت طراحی و خودکارسازی بسیاری از مراحل طراحی و تولید تراشه گردیده است. این ابزارها در سطوح مختلفی جهت کاهش زمان طراحی، اطمینان از درستی طرح و پیاده‌سازی مدار مجتمع بکار گرفته می‌شوند. بدون ابزار CAD، طراحی مدارهای مجتمع پیچیده بعلت طولانی شدن زمان طراحی، عدم بهینه بودن طرح و افزایش هزینه‌های مراحل مختلف طراحی غیرممکن است.

در این بخش راهبرد چند مرحله‌ای تکوین یک تراشه تشریح می‌گردد (نمودار ۱-۱) ونقش فرآیند آزمون تراشه در مراحل مربوطه به تفصیل بیان می‌گردد.

مرحله ۱: مساله‌سازی[3]

در بسیاری از موارد، تعریف دقیق و همه جانبه‌ی مساله (مدار، سیستم و ...) روشن نیست. مساله‌سازی عبارت از اخذ اطلاعات مهم و بحرانی از مساله می‌باشد. از جمله این اطلاعات مهم می‌توان به شناسایی

1- Very Large Scale Integrated Circuits
2- Computer Aided Design
3- Abstraction

الگوریتم‌های لازم که سیستم باید قادر به اجرای آنها باشد، محدودیت‌های سیستم مورد نظر، قابلیت‌های گسترش، توانایی برقراری ارتباط با سیستم‌های دیگر و ... را مورد نظر قرار داد.

مرحله ۲: تعریف مشخصات

•پس از مرحله ۱ تقریبا مشخصات عمومی سیستم پیدا شده است و اینک می‌توان برای تعیین جهت طراحی تصمیم‌های لازم را اتخاذ نمود. برای مثال در مورد یک پردازنده تعریف مشخصات زیر الزامی می‌باشد:

•فلسفه‌ی معماری (RISC یا CISC)

•طول کلمه

•نوع عملیات محاسباتی (اعشاری یا صحیح)

•سبک معماری (pipeline یا bit-slice)

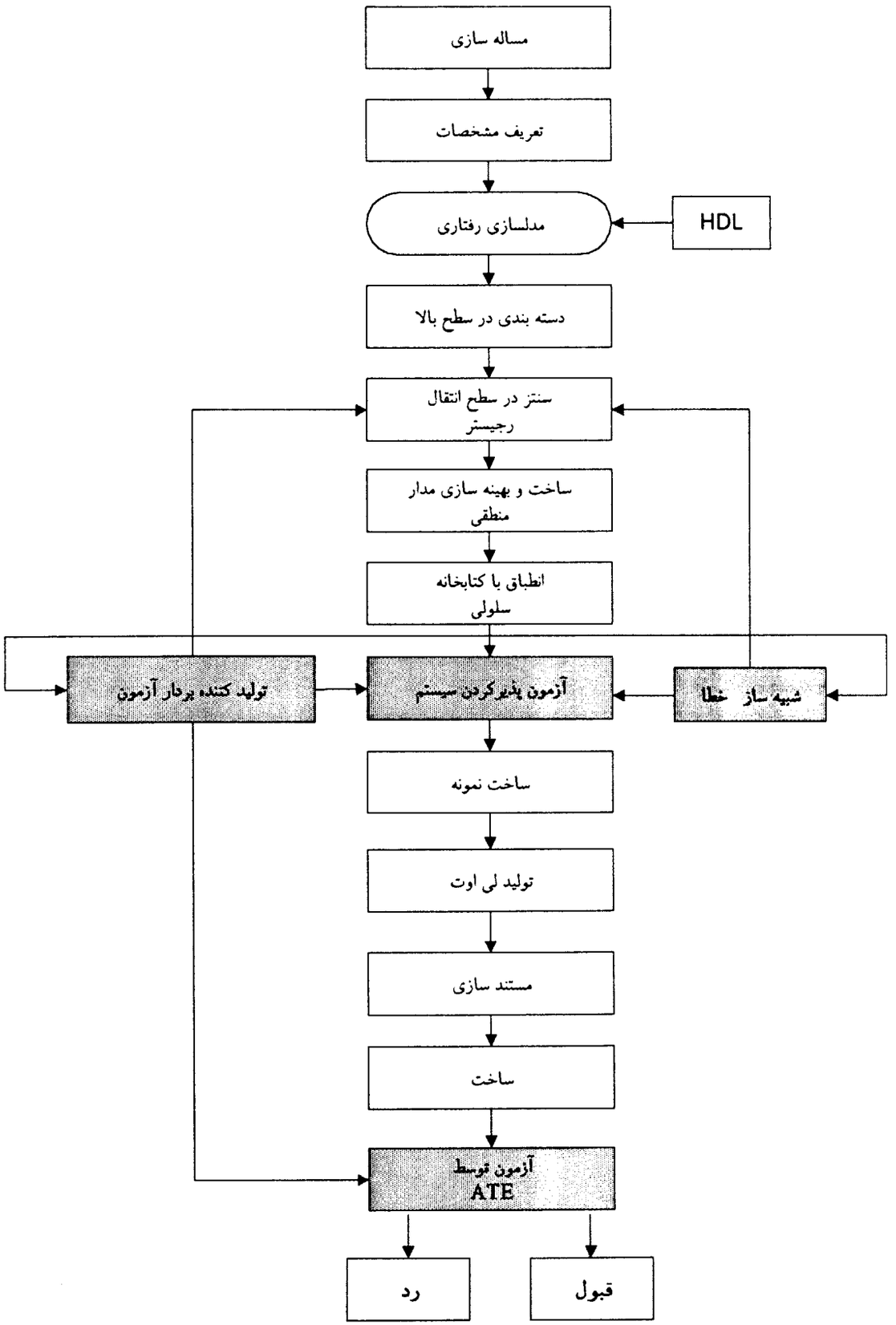
مرحله ۳: مدلسازی رفتاری

در این مرحله بدون آنکه به فکر محدودیت‌ها و مشخصات ساختاری باشیم باید بتوانیم رفتار سیستم را مدلسازی کنیم. در این مرحله از زبانهای توصیف سخت‌افزار مانند VHDL و Verilog استفاده می‌گردد. این مرحله دو نتیجه مهم بدنبال خواهد داشت:

الف) در ضمن مدلسازی رفتاری کاستی‌ها و افزودنی‌های رفتاری روشن خواهدشد و لذا عملا نوعی بهینه‌سازی رفتاری انجام خواهد گرفت.

ب) با اجرای شبیه‌ساز زبان سخت‌افزاری بکارگرفته شده خواهیم توانست رفتار بیان شده را بیازماییم و توالی عملیات و درست بودن رفتار را تایید نماییم.

در این مرحله مساله بطور کامل تفهیم گردیده و ابهامات احتمالی باید برطرف گردد.



نمودار ۱-۱: مراحل تکوین یک تراشه

مثال ۱-۱

رفتار یک سیستم ساده با استفاده از زبان سخت‌افزاری VHDL بصورت زیر مدل گردیده است.

```
ENTITY example IS
    PORT (a,b,c : IN BIT_VECTOR(3 DOWNTO 0);
          p     : OUT BIT_VECTOR(3 DOWNTO 0);
END example;

ARCHITECTURE behavioral OF example IS
SIGNAL i: INTEGER;
 PROCESS
  BEGIN
     p <= 0;
     FOR  i IN ( 1 TO c )
          p <= p + ( a * b );
     END;
 END PROCESS;
END  behavioral;
```

مرحله ٤: دسته‌بندی در سطح بالا

از آنجا که اصول حاکم به سنتز رفتارهای متفاوت، ملزومات مختلفی را نیاز دارد (مثلا سنتز مسیرداده با کنترلر متفاوت است) در این مرحله بر اساس رفتار مدل گردیده در مرحله‌ی قبل، رفتارها باید دسته‌بندی گردند و هر کدام با در نظرگیری اصول مناسب محور طراحی، بهینه سازی و سنتز شوند.

مثال ۱-۲

رفتار یک پردازنده که توسط یک زبان سخت‌افزاری مدل گردیده می‌تواند طبق شکل ۱-۱ دسته‌بندی شود.

مرحله ٥: سنتز در سطح انتقال رجیستر

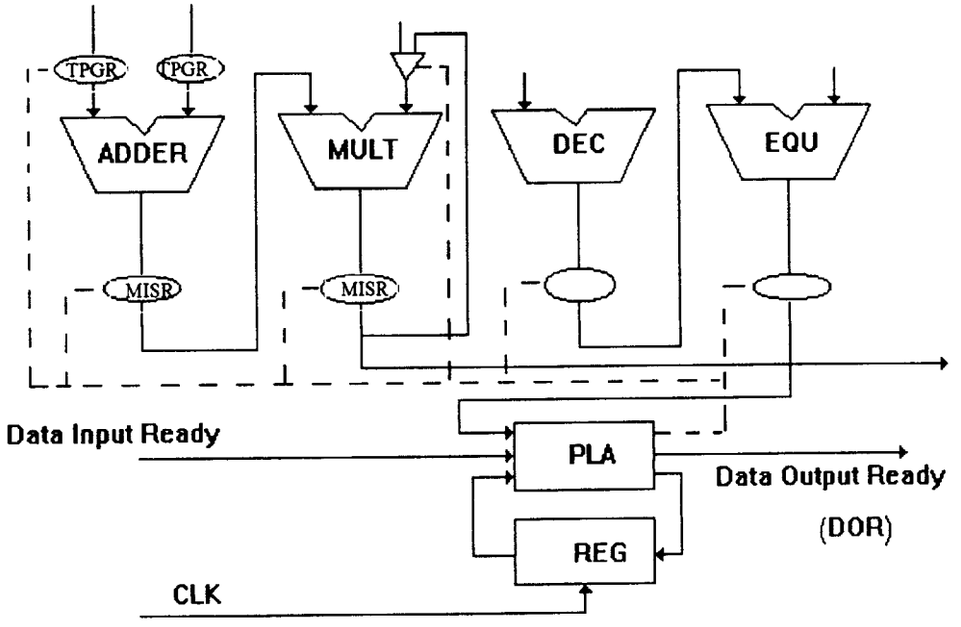
در سطح انتقال رجیستر به جزییات منطقی مدارها توجهی نمی‌شود. اجزا در این سطح عبارتند از: عناصر محاسباتی مانند ضرب‌کننده، جمع‌کننده، عناصر هدایت‌گر مانند باس‌ها و مالتی‌پلکسرها و عناصر ذخیره‌کننده مانند رجیسترها. بدلیل کم بودن تعداد عناصر امکان بهینه‌سازی (از دیدگاه هزینه یا تاخیر) نیز فراهم می‌گردد. از طرف دیگر این سطح برای بررسی تعادل بین پارامترهای مهم، سطح مطلوبی است بدین دلیل که از یکطرف

اطلاعات برای ارزیابی بحد کافی در دسترس قراردارد و ازطرف دیگر اطلاعات و تعداد عناصر بسیار زیاد نیست که جستجو در فضای طرح یا بهینه‌سازی وقت فوق‌العاده زیادی ببرد [Demi94].

یکی از ویژگی‌های این سطح امکان بررسی آزمون‌پذیری سیستم می‌باشد. در این سطح با روشهای تقریبی می‌توان معیاری از وضعیت آزمون‌پذیری سیستم را تخمین‌زد و درصورت نیاز، به اصلاحاتی در طراحی مبادرت نمود ویا عناصر خاصی جهت آزمون‌پذیر نمودن سیستم اضافه نمود. از این عناصر می توان به TPGR[1] و MISR[2] اشاره کرد.

مثال ۱-۳

در مدار نشان داده شده در شکل ۱-۱ نحوه جایگزینی بعضی از رجیسترها با TPGR و MISR نمایش داده شده است.



شکل ۱-۱: نمایش دسته‌بندی رفتار یک پردازنده

1-Test Pattern Generator Register
2-Multiple Input Signature Register

مرحله ٦: ساخت و بهینه‌سازی مدارمنطقی

عناصر سطح انتقال رجیستر بصورت کلی درمرحله ٥ طرح و به یکدیگر متصل می‌شوند. در این مرحله برای هر یک از این عناصر یک مدارمنطقی در نظر گرفته می‌شود. مثلا برای یک جمع کننده ٤ بیتی یک مدار متشکل از چهار عدد full-adder که بدنبال هم می‌آیند یعنی ripple-carry-adder در نظر گرفته می‌شود. بخاطر تنوع فراوان در انتخاب سلولها و همچنین مشخصات مختلف هزینه و سرعت سلول‌ها انتخاب‌های متعددی ممکن است وجود داشته باشد که در این مرحله برای این موارد تصمیم گرفته خواهد شد.

هنگامی که مدار منطقی کامل برای سیستم بدست آمد ابزارهای بهینه‌سازی مناسب باید مورد استفاده قرار گیرند تا این مدارها از نظر تعداد گیتها و گاهی تعداد سطوح (متناظر با تاخیر) بهینه شوند. این مرحله عموما با یک شبیه‌سازی مجدد کامل می‌شود تا از صحت روند بهینه‌سازی نیز اطمینان حاصل نماییم.

مثال ۱-٤

برای اجزای معرفی شده در شکل ۱-۱ در (مرحله‌ی قبل) ساختارهایی در سطح گیت منطقی معرفی می‌نماییم.

- ضرب کننده: ضرب کننده آرایه‌ای
- جمع کننده: Ripple carry adder
- DEC: DEC ساده با استفاده از گیتهای ساده منطقی
- EQU: مقایسه کننده ساده با کمک XOR
- PLA (با گیتهای ساده) برای تحقق Control Table مربوط به کنترلر
- رجیسترها با استفاده از D-flipflop و Gated Clock برای داشتن کنترل روی نوشته شدن آنها و کاهش توان مصرفی
- مالتی پلکسر معمولی ۲×۱ به کمک گیت‌های منطقی

•تغییر رجیسترهای ورودی جمع‌کننده به عنوان TPGR و رجیستر خروجی ضرب‌کننده و جمع‌کننده بعنوان MISR برای آزمون‌پذیر کردن مدار.

مرحله ۷: انطباق با کتابخانه سلولی

پس از بدست آوردن مدارمنطقی و بهینه‌سازی آن برای تسریع در امر طراحی و تکمیل مدار معمولا از کتابخانه‌های سلولی که شامل مجموعه‌ی نسبتا کاملی از سلول‌های منطقی (با اطلاعات مناسب برای تولید و بهینه‌سازی لی‌اوت) هستند استفاده می‌شود. انتخاب این سلولها بر اساس پیچیدگی مدار، محدودیت‌های سرعت، هزینه و درنظرگیری کاهش اتصالات صورت می‌گیرد. این مرحله را گاهی اتصال بر اساس کتابخانه[1] هم می‌گویند که تحقق مدار منطقی حاصل را بر اساس سلول‌های آزمایش‌شده یک کتابخانه انجام می‌دهد.

مثال ۱-۵

فرض کنید در این مرحله، کتابخانه خاصی از شرکت VLSI Technology با مشخصات ویژه سرعتی استفاده می‌کنیم و بر اساس آن تصمیم می‌گیریم که همه‌ی ANDها را به NAND و همه‌ی ORها را به NOR تبدیل کنیم. در ضمن بخاطر مشخصات خاص این کتابخانه فلیپ‌فلاپ D هم بر اساس فلیپ‌فلاپ J-K ساخته می‌شود.

مرحله ۸: آزمون‌پذیرکردن سیستم

تا این مرحله مدار بطور کامل در سطح گیت پیاده‌سازی گردیده است. حال باید ابتدا با بکارگیری ابزار CAD میزان آزمون‌پذیر بودن مدار، ارزشیابی گردد و سپس در صورت مشاهده‌ی کاستی در معیارهای آزمون‌پذیری بدست‌آمده، مدار باید با استفاده از روشهای مختلف مانند روش پویش[2]، روش خودآزمایی[3]، روش پویش مرزی[4] و ... آزمون‌پذیر گردد. بین این مرحله و مرحله‌ی ۵ ممکن است چندین بار انتقال انجام پذیرد تا در نهایت طرحی آزمون‌پذیر عرضه گردد.

1- Library binding
2- Scan
3- Biult In Self Test
4- Boundary-Scan