

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



دانشکده برق و روباتیک

گروه الکترونیک

طراحی و شبیه‌سازی یک شبکه روی تراشه‌ی ترکیبی با زبان توصیف سخت افزار VHDL

سید منصور شریفی مهارلویی

استاد راهنما:

دکتر امیدرضا معروضی

استاد مشاور:

دکتر عباس رضانی

پایان نامه ارشد جهت اخذ درجه کارشناسی ارشد

بهمن ماه ۱۳۹۱

دانشگاه صنعتی شاهرود

دانشکده :

گروه :

پایان نامه کارشناسی ارشد آقای / خانم

تحت عنوان:

در تاریخ توسط کمیته تخصصی زیر جهت اخذ مدرک کارشناسی ارشد
مورد ارزیابی و با درجه مورد پذیرش قرار گرفت.

امضاء	اساتید مشاور	امضاء	اساتید راهنما
	نام و نام خانوادگی :		نام و نام خانوادگی :
	نام و نام خانوادگی :		نام و نام خانوادگی :

امضاء	نماینده تحصیلات تکمیلی	امضاء	اساتید داور
	نام و نام خانوادگی :		نام و نام خانوادگی :
			نام و نام خانوادگی :
			نام و نام خانوادگی :
			نام و نام خانوادگی :

تقدیم به مادر عزیزم که در تمام مراحل زندگی دلسوزانه مشوق و راهنمایم بوده است.

تقدیم به پدر کرامی که راه درست زیستن را به من آموخت.

تقدیم به خواهران و برادرانم که همواره در کنارم بوده اند.

باسپاس از

اساتید محترم

دکتر امیدرضا معروضی از دانشگاه صنعتی شاهرود

و

دکتر عباس رمضانی از دانشگاه صنعتی بهمان

که با بلندای اندیشه و صفای وجودی خود مرا از راهمائی های بی دریغ و ارزشمندشان بهره مند نمودند.

تعهد نامه

اینجانب دانشجوی دوره کارشناسی ارشد رشته
دانشکده دانشگاه صنعتی شاهرود نویسنده پایان نامه
..... تحت راهنمایی.....متعهد می شوم .

- تحقیقات در این پایان نامه توسط اینجانب انجام شده است و از صحت و اصالت برخوردار است .
- در استفاده از نتایج پژوهشهای محققان دیگر به مرجع مورد استفاده استناد شده است .
- مطالب مندرج در پایان نامه تاکنون توسط خود یا فرد دیگری برای دریافت هیچ نوع مدرک یا امتیازی در هیچ جا ارائه نشده است .
- کلیه حقوق معنوی این اثر متعلق به دانشگاه صنعتی شاهرود می باشد و مقالات مستخرج با نام « دانشگاه صنعتی شاهرود » و یا « Shahrood University of Technology » به چاپ خواهد رسید .
- حقوق معنوی تمام افرادی که در به دست آمدن نتایج اصلی پایان نامه تأثیرگذار بوده اند در مقالات مستخرج از پایان نامه رعایت می گردد.
- در کلیه مراحل انجام این پایان نامه ، در مواردی که از موجود زنده (یا بافتهای آنها) استفاده شده است ضوابط و اصول اخلاقی رعایت شده است .
- در کلیه مراحل انجام این پایان نامه، در مواردی که به حوزه اطلاعات شخصی افراد دسترسی یافته یا استفاده شده است اصل رازداری ، ضوابط و اصول اخلاق انسانی رعایت شده است .

تاریخ

مالکیت نتایج و حق نشر

- کلیه حقوق معنوی این اثر و محصولات آن (مقالات مستخرج ، کتاب ، برنامه های رایانه ای ، نرم افزار ها و تجهیزات ساخته شده است) متعلق به دانشگاه صنعتی شاهرود می باشد . این مطلب باید به نحو مقتضی در تولیدات علمی مربوطه ذکر شود .
- استفاده از اطلاعات و نتایج موجود در پایان نامه بدون ذکر مرجع مجاز نمی باشد.

چکیده: نیاز روز افزون به تراشه هایی با کارایی بیشتر، افزایش پیچیدگی در طراحی مدارهای مجتمع را به دنبال داشته است. بخشی از مشکلات با کوچکتر شدن تکنولوژی ساخت ترانزیستور برطرف شد ولی کوچک تر شدن تکنولوژی ساخت خود باعث برهم خوردن توازن بین تأخیر سیم و تأخیر گیت می شد. همچنین با افزایش فرکانس کار تراشه توان مصرفی نیز افزایش می یافت. برای غلبه بر این چالش ها طراحان مدار های مجتمع تمرکز بر افزایش کارایی را جایگزین تمرکز بر افزایش سرعت نمودند و این تغییر نگرش باعث شد که دو پردازنده مجزا را در یک تراشه قرار دهند و با یک باس ارتباط بین آن ها را برقرار کنند. نتیجه کار بسیار رضایت بخش بود و بگونه ای که در مدت کوتاهی سیستم های تشکیل شده از چندین بخش مختلف که بر روی یک برد اجرا می شدند نیز به درون یک تراشه انتقال یافتند. این معماری ساخت پردازنده ها به سیستم روی تراشه یا SOC معروف شد. اما این پایان کار نبود، سیستم روی تراشه نیز دارای مشکلاتی بود که با گذشت زمان مشخص شد. با افزایش تعداد بخش های مجزا که به واحدهای IP معروف شدند سیستم روی تراشه دیگر پاسخگو نبود. مشکلاتی از قبیل عدم مقیاس پذیری و مصرف توان قابل توجه در باس بار دیگر تلاش جدیدی را در طراحی مدارهای مجتمع برانگیخت، این بار نظرها به سمت معماری شبکه روی تراشه معطوف شد.

در معماری شبکه روی تراشه که با ایده گرفتن از شبکه های کامپیوتری مقیاس بزرگ مطرح شده بود لایه هایی را بر اساس استاندارد OSI پیشنهاد می شود. این معماری بسیار کارآمد بود و به سرعت جایگزین دیگر معماری ها شد و شبکه های مختلفی با ویژگی ها و توانایی های گوناگون موضوع بسیاری از تحقیقات شد.

در این پایان نامه تلاش شده است یکی از شبکه های معرفی شده با معماری شبکه روی تراشه به نام شبکه Hermes شبیه سازی شود و برخی از ویژگی های آن ارتقاء یابد. یکی از معایب شبکه Hermes این است که با افزایش تعداد IPها در آن کارایی آن کاهش می یابد. از طرف دیگر ممکن است چند IP با یکدیگر تبادل داده و همکاری زیادی داشته باشند اگر آن ها با یکدیگر یک شبکه محلی تشکیل دهند راحت تر و سریع تر می توانند با

یکدیگر ارتباط داشته باشند. بنابراین در شبکه طراحی شده به جای یک IP تنها شبکه ای از چهار IP که با یکدیگر تبادل اطلاعات زیادی دارند به کار رفته است که ارتباط بین آن ها با یک باس برقرار شده است. این طراحی باعث شده است ترکیبی از SOC و NOC داشته باشیم. عبارت " ترکیبی " در عنوان پایان نامه نیز به همین واقعیت اشاره دارد. همچنین بسته ها با فرکانس کار شبکه ارسال می شوند و زمان مسیریابی کاهش یافته است در نتیجه سرعت انتقال بسته ها نیز در مقایسه با کارهای مشابه انجام شده بیشتر شده است. دیگر ویژگی کار انجام شده این است که با طراحی سخت افزار مناسب اثر طول بسته بر میزان تأخیر شبکه از بین رفته است یعنی با افزایش طول بسته میزان تأخیر برای انتقال یک بسته بین دو سوئیچ مشخص ثابت می ماند.

کلمات کلیدی: سیستم روی تراشه، شبکه روی تراشه، بلوک های IP

لیست مقالات استخراج شده :

- ۱- عنوان مقاله: افزایش سرعت انتقال داده ها در شبکه روی تراشه با استفاده از الگوریتم مسیریابی XY تعمیم یافته و کاهش زمان مسیریابی. بیست و یکمین کنفرانس برق ایران، مشهد، دانشگاه فردوسی

فهرست

۱۷.....	۱ مقدمه
۱۹.....	۱-۱ ۱-۱ مروری بر کارهای انجام شده.....
۲۰.....	۲-۱ اهداف پایان نامه.....
۲۱.....	۳-۱ ساختار پایان نامه.....
۲۳.....	۲ مقدمه ای بر شبکه روی تراشه.....
۲۶.....	۱-۲ استانداردهای معماری ارتباط روی تراشه.....
۲۷.....	۱-۱-۲ استاندارد AMBA 2.0.....
۲۹.....	۱-۱-۱-۲ ۱-۱-۱-۲ باس پیشرفته با کارایی بسیار بالا.....
۳۲.....	۲-۱-۱-۲ ۲-۱-۱-۲ باس جانبی پیشرفته.....
۳۴.....	۲-۱-۲ استاندارد ABM 3.0.....
۳۴.....	۱-۲-۱-۲ ارتباط قابل گسترش پیشرفته (AXI).....
۴۱.....	۴-۱-۲ استاندارد ارتباط بخش مجازی (VCI).....
۴۲.....	۵-۱-۲ استاندارد ارتباطی DTL فیلیپس.....
۴۴.....	۲-۲ معایب استفاده از باس.....
۴۴.....	۱-۲-۲ عدم مقیاسپذیری.....

- ۴۵..... ۱-۱-۲-۲ ارتباط همزمان
- ۴۵..... ۲-۱-۲-۲ کیفیت خدمات
- ۴۷..... ۲-۲-۲ مشکلات فیزیکی
- ۴۷..... ۱-۲-۲-۲ تأخیر سیم
- ۴۹..... ۲-۲-۲-۲ مصرف توان
- ۵۰..... ۳-۲ مبانی شبکه روی تراشه
- ۵۴..... ۳ ویژگی های شبکه روی تراشه
- ۵۵..... ۱-۳ توپولوژی
- ۵۷..... ۱-۱-۳ توپولوژی های بر پایه درخت:
- ۶۱..... ۲-۱-۳ توپولوژیهای مستقیم
- ۶۲..... ۱-۲-۱-۳ توپولوژی حلقه وتری
- ۶۳..... ۲-۲-۱-۳ تولوژی های میس
- ۶۶..... ۲-۳ مسیریابی
- ۶۸..... ۱-۲-۳ الگوریتم های مسیریابی
- ۶۸..... ۱-۱-۲-۳ الگوریتم های مسیریابی مستقیم
- ۷۱..... ۲-۱-۲-۳ الگوریتم های مسیریابی چرخشی
- ۷۲..... ۳-۳ روش ارسال بسته

۷۳ ۱-۳-۳ ذخیره و ارسال
۷۳ ۲-۳-۳ برشمیان مجازی
۷۴ ۳-۳-۳ خزیدنی
۷۶ ۴-۳ کنترل جریان شبکه
۷۸ ۴ شبیه سازی و تحلیل نتایج
۸۱ ۱-۴ توپولوژی شبکه
۸۲ ۲-۴ اجزای یک گره
۸۲ ۱-۲-۴ واحد IP
۸۴ ۲-۲-۴ سوئیچ
۸۸ ۳-۲-۴ مسیریاب (Router)
۸۹ ۴-۲-۴ ناظر (Arbiter)
۸۹ ۳-۴ روند کلی ارسال یک بسته
۹۰ ۳-۴ شبکه با گذرگاه اشتراکی
۹۲ ۴-۴ تحلیل نتایج
۹۲ ۱-۴-۴ فرکانس سیگنال های TX و IX
۹۴ ۲-۴-۴ کاهش زمان مسیریابی
۹۵ ۳-۴-۴ میزان تأخیر

۹۶ سنتر ۴-۴-۴
۹۷ پیشنهادهایی برای کار در آینده ۵-۴-۴
۹۹ مراجع

فهرست تصاویر

۱۸ شکل ۱-۱: منحنی نشان دهنده قانون مور
۲۵ شکل ۱-۲: نمایش ساختار داخلی تراشه core 2Due
۲۵ شکل ۲-۲: نمایش افزایش تعداد IPها در سال های مختلف
۲۸ شکل ۳-۲: نمایش استاندارد AMBA 2.0
۳۰ شکل ۴-۲: طرح اتصال داخلی استاندارد AHB با مالتیپلکسر
۳۱ شکل ۵-۲: یک مثال با دو ارباب و چهار نوکر استاندارد AHB: توپولوژی ماتریس گذرگاه کامل
۳۲ شکل ۶-۲: توپولوژی ماتریس باس غیرکامل
۳۳ شکل ۷-۲: دیاگرام حالت استاندارد APB
۳۵ شکل ۸-۲: کانال های خواندن آدرس و خواندن داده در معماری AXI
۳۵ شکل ۹-۲: کانال های نوشتن داده و پاسخ نوشتن
۳۷ شکل ۱۰-۲: انتقال داده های پیوسته در استاندارد AHB
۳۷ شکل ۱۱-۲: انتقال دادهای پیوسته در AXI

- شکل ۲-۱۲: انجام همزمان عمل خواندن و نوشتن در استاندارد AXI..... ۳۸
- شکل ۲-۱۳: مثالی از پیکره بندی سیستم با استاندارد ارتباطی OCP بر پایه سوکت..... ۴۰
- شکل ۲-۱۴: نمایش استاندارد OCP..... ۴۲
- شکل ۲-۱۵: نمایش تأخیر ایجاد شده در برابر افزایش IPها..... ۴۴
- شکل ۲-۱۶: نمایش برهم خوردن توازن تأخیر سیمها و تأخیر گیت..... ۴۸
- شکل ۲-۱۷: نمایش لایه‌ها در شبکه روی تراشه..... ۵۲
- شکل ۳-۱: درخت باینری..... ۵۷
- شکل ۳-۲: آرایش بسط یافته یک درخت باینری..... ۵۸
- شکل ۳-۳: درخت - فربه پروانه ای..... ۵۹
- شکل ۳-۴: درخت - فربه باینری (۲- آرایه ۴- بعدی)..... ۵۹
- شکل ۳-۵: نمایش بسط یافته درخت فربه- باینری..... ۵۹
- شکل ۳-۶: شبکه SPIN ۴-آرایه ۳- بعدی..... ۶۰
- شکل ۳-۷: نمایش دیگر شبکه SPIN ۴- آرایه ۳- بعدی..... ۶۰
- شکل ۳-۸: نمایش یک توپولوژی آرایه ای..... ۶۱
- شکل ۳-۹: نمایش یک توپولوژی حلقه ای..... ۶۲
- شکل ۳-۱۰: نمایش یک توپولوژی حلقه وتری با ۱۶ گره..... ۶۲
- شکل ۳-۱۱: توپولوژی حلقه وتری سلسله مراتبی..... ۶۳
- شکل ۳-۱۲: نمایش توپولوژی مش..... ۶۴
- شکل ۳-۱۳: نمایش توپولوژی طاقچه ای..... ۶۵
- شکل ۳-۱۴: پیچاندن توپولوژی طاقچه‌های..... ۶۵

- شکل ۳-۱۵: نمایش هرزگردی در شبکه ۶۷
- شکل ۳-۱۶: نمایش بنبست در شبکه ۶۸
- شکل ۳-۱۷: نمایش الگوریتم مسیریابی XY ۶۹
- شکل ۳-۱۸: مسیریابی به روش احاطه کننده افقی ۷۰
- شکل ۳-۱۹: مسیریابی به روش احاطه کننده عمودی ۷۱
- شکل ۳-۲۰: نمایش مسیریابی های چرخشی ۷۲
- شکل ۳-۲۱: ارسال بسته ها در روش ذخیره و ارسال ۷۳
- شکل ۳-۲۲: نمایش روش برش - میانی مجازی ۷۴
- شکل ۳-۲۳: نمایش روش خزیدنی ۷۵
- شکل ۳-۲۴: نمایش حرکت فلیت ها در روش خزیدنی ۷۵
- شکل ۳-۲۵: نمایش حرکت فلیت ها در روش خزیدنی نسبت به زمان ۷۶
- شکل ۴-۱: نمایش توپولوژی شبکه ۸۱
- شکل ۴-۲: نمایش شماتیک سوئیچ طراحی شده ۸۲
- شکل ۴-۳: نتیجه سنتز یک IP ۸۴
- شکل ۴-۴: ارسال فلیت های یک بسته به پایه خروجی توسط یک IP ۸۴
- شکل ۴-۵: نمایش مشخص شدن مسیر برای ارسال فلیت ها ۸۶
- شکل ۴-۶: نمایش وضعیت ثبات های reg1 و reg2 ۸۶
- شکل ۴-۷: ذخیره فلیت ها در بافر ۸۷
- شکل ۴-۸: ارسال فلیت ها به خارج از سوئیچ ۸۷
- شکل ۴-۹: نمایش استفاده از سیگنال کمکی clk2 ۸۸

- شکل ۴-۱۰: ارسال یک بسته از پورت Local به پورت East ۹۰
- شکل ۴-۱۱: ارسال فلیت ها از پورت East به خارج از سوئیچ ۹۰
- شکل ۴-۱۲: ارتباط چهار IP با یک گذرگاه ارتباطی با سوئیچ ۹۱
- شکل ۴-۱۳: ارتباط چند IP توسط یک ناظر با سوئیچ ۹۲
- شکل ۴-۱۴: سیگنال های rx و tx ۹۳
- شکل ۴-۱۵: سیگنال های tx و clk ۹۳
- شکل ۴-۱۶: مقایسه فرکانس سیگنال های rx و tx با فرکانس پالس ساعت ورودی در شبکه Hermes ۹۳
- شکل ۴-۱۷: نمایش مدت زمان مسریابی در شبکه Hermes ۹۴

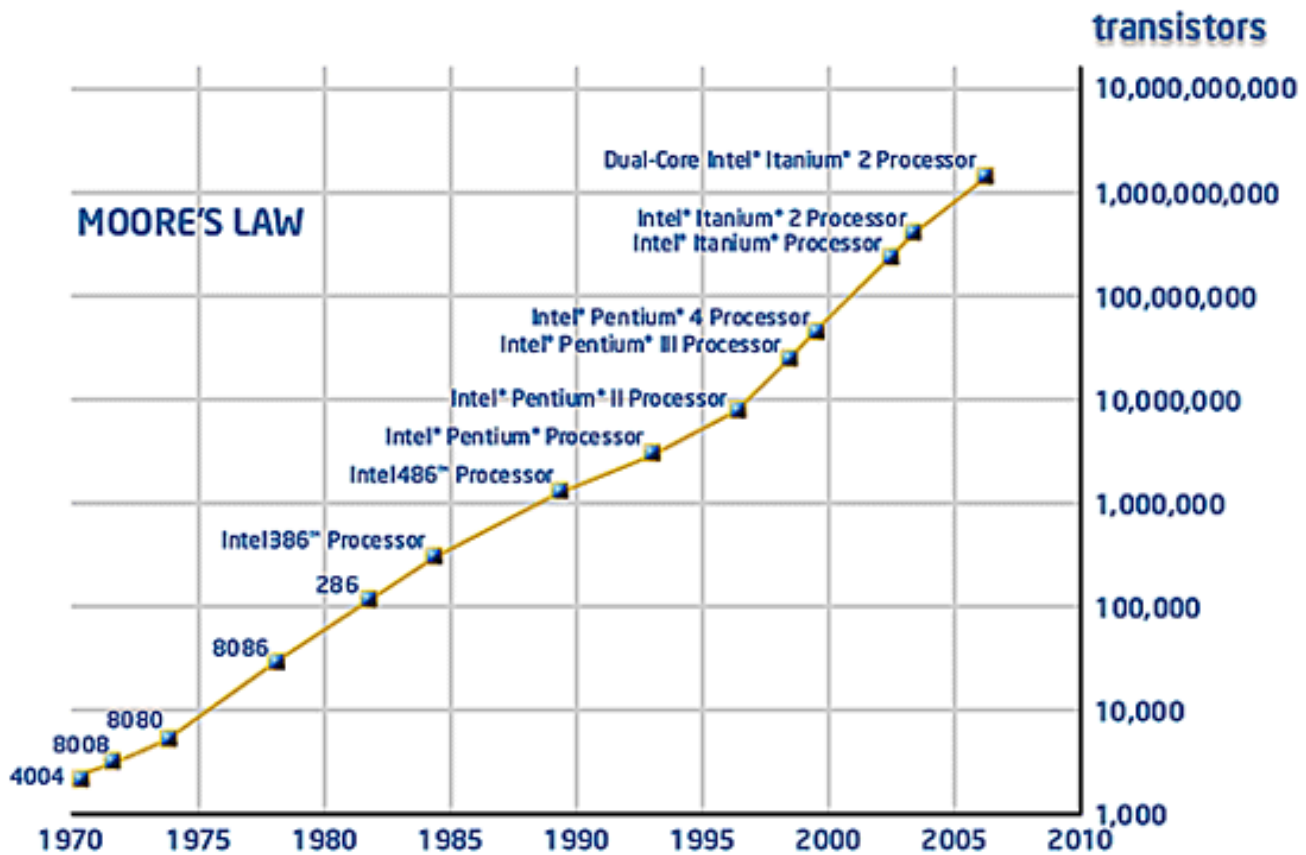
فهرست جدول ها

- جدول ۲-۱: ویژگی های چند توپولوژی در طراحی شبکه روی تراشه ۵۶
- جدول ۳-۱: تعداد پالس های ساعت مورد نیاز برای بسته های مختلف ۹۵

فصل اول :

مقدمه

نیاز روزافزون به تراشه‌هایی با فرکانس کار بالاتر و مصرف توان کمتر به سرعت رو به رشد است به گونه‌ای که طبق رابطه‌ای که به قانون مور معروف است پیش‌بینی می‌شود که در هر بازه‌ی زمانی ۱۸ ماه باید تعداد ترانزیستورهای روی یک سطح مشخص از مدارهای مجتمع دو برابر شود تا پاسخگوی نیازهای پیش رو باشد. شکل (۱-۱) این میزان رشد را نشان می‌دهد.



شکل ۱-۱: منحنی نشان دهنده قانون مور

انتظار می‌رود اندازه ترانزیستور نیز این روند را ادامه دهد. یک کاهش ۷۰ درصدی در ابعاد ترانزیستور با حرکت به سمت یک تکنولوژی ساخت جدیدتر (به عنوان مثال از تکنولوژی ساخت ۶۵ نانومتر به ۴۵ نانومتر) اجازه می‌دهد که چگالی ترانزیستور در سطح تراشه به دو برابر افزایش یابد. تکنولوژی‌های جدیدتر از قبیل برش‌های سه بعدی سطح سلیکون ممکن است حتی باعث بیش از دو برابر شدن چگالی ترانزیستورها شود. مؤسسه ولی مشکل

تنها کوچک‌تر شدن ابعاد تراشه نیست بلکه با افزایش فرکانس کار تراشه توان مصرفی به شدت افزایش می‌یابد به طوری که طبق مقاله‌ای که اخیراً از سوی شرکت اینتل منتشر شده است از توان مصرفی و گرمای ناشی از آن به عنوان عامل محدود کننده افزایش سرعت تراشه‌ها یاد شده است. اگرچه تا ۲۰ سال آینده افزایش فرکانس مشکل خاصی را ایجاد نمی‌کند ولی بعد از ۲۰ سال نیاز به تحقیقات جدید وجود خواهد داشت.

امروزه در طراحی مدارهای مجتمع برای افزایش سرعت و کاهش توان مصرفی از چندین واحد پردازنده که توانایی کار به صورت همزمان را دارند استفاده می‌شود. با افزایش تعداد این واحدهای پردازنده که به IP^۱ها معروف هستند ارتباط داخلی آن‌ها حائز اهمیت می‌شود. در گذشته سعی در استفاده از باس برای ارتباط داخلی بین IPها بود. این معماری که در آن از گذرگاه برای ارتباط بین IPها استفاده می‌شود به سیستم روی تراشه (SOC^۲) معروف شد ولی استفاده از گذرگاه مشکلات زیادی داشت، برای غلبه بر این مشکلات ایده‌ی شبکه روی تراشه (NOC^۳) مطرح شد که در آن با ایده گرفتن از شبکه‌های کامپیوتری مقیاس بزرگ لایه‌هایی مطابق با استاندارد OSI تعریف می‌شود

۱-۱ مروری بر کارهای انجام شده

تا کنون شبکه‌های بسیار زیادی با ویژگی‌های مختلف طراحی و ارائه شده است که از میان آن‌ها می‌توان به موارد زیر اشاره کرد. شبکه X pipe [۳۵] که توسط D. Bertozzi معرفی شد و دارای توپولوژی غیرمنظم می‌باشد و بسته‌ها در آن به روش بسته‌ای و نوع خزیدنی انجام می‌شود. شبکه‌های aSOC [۳۶] (معرفی شده توسط Liang) و SoCBUS [۳۷] (معرفی شده توسط Wikl, Henr)، که دارای توپولوژی میس می‌باشند و ارسال بسته‌ها را به روش مداری انجام می‌دهند. شبکه Proteo [۳۷] که توسط Sigue, Ahon معرفی شده

^۱ - intellectual property

^۲ -Systems on Chip

^۳ -Network on Chip

است دارای توپولوژی حلقه ای می باشد و از روش برش میان مجازی برای ارسال بسته ها استفاده می کند و اندازه فلیت ها در آن ۸ بیت می باشد، در این شبکه پروتکل ارتباطی^۱ VCI برای اتباط IPها و سوئیچ ها مورد استفاده قرار گرفته است. شبکه Spin [۳۸] که توسط Guerr., Andr معرفی شد، توپولوژی آن درخت-فریه باینری و ارسال بسته ها در آن به روش خزیدنی انجام می شود در این شبکه نیز از پروتکل ارتباطی VCI استفاده شده است. شبکه Aethereal [۳۹] که توسط Rijpkema معرفی شد دارای توپولوژی میش می باشد و از روش خزیدنی برای ارسال بسته ها استفاده می کند، اندازه فلیت ها^۲ (کوچکترین بخش از یک بسته که ارسال می شود) در این شبکه ۳۲ بیت تعیین شده است. شبکه Octagon [۴۰] که توسط Karim معرفی شد دارای توپولوژی حلقه وتری بوده و بسته ها در آن به روش برش میان مجازی ارسال می شوند، در این شبکه نیز اندازه فلیت ها ۳۲ بیت تعیین شده است. شبکه Nostrum [۴۱] که توسط Millberg, Pen معرفی شد دارای توپولوژی میش می باشد و بسته ها به روش ذخیره و ارسال منتقل می شوند. شبکه Qnoc [۴۲] که توسط Bolotin, Rost معرفی شد دارای توپولوژی میش بوده و بسته ها در آن به روش خزیدنی ارسال می شوند. شبکه Hermes [۳۴] که توسط Moraes معرفی شد نیز دارای توپولوژی میش و ارسال بسته ها به روش خزیدنی در آن انجام می شود.

۱-۲ اهداف پایان نامه

در این پایان نامه شبکه Hermes که در بخش قبل معرفی شد مورد توجه قرار گرفته است. یکی از مواردی که در معماری NOC و همچنین در شبکه Hermes مورد توجه قرار نمی گیرد این است که ممکن است در شبکه چند IP تبادل اطلاعاتی زیادی با یکدیگر داشته باشند و همواره نیاز باشد که داده ها بین آن ها مبادله شود، (مانند ارتباط یک حافظه و یک پردازنده که نیاز است داده ها از پردازنده به حافظه و از پردازنده به حافظه

^۱ virtual component interface

^۲ flow control digits(flit)

منتقل شود). برای این دسته از IPها می توان یک شبکه محلی تعریف کرد تا اینکه تبادل داده ها راحت تر و با سرعت بیشتری انجام شود. به علاوه این شبکه محلی باعث کوچکتر شدن اندازه شبکه اصلی خواهد شد.

در شبکه Hermes برای انتقال هر فلیت دو پالس ساعت مصرف می شود که برطرف کردن این مشکل نیز از اهداف پایان نامه می باشد. چون مصرف پالس ساعت بیشتر، به معنی افزایش توان مصرفی و بیشتر شدن تأخیر در ارسال بسته ها می باشد. بر طرف شدن این مشکل در افزایش سرعت انتقال داده ها بسیار مؤثر است. توجه داشته باشید فرکانس کار شبکه از مرتبه گیگا هرتز می باشد.

مدت زمان زیاد برای مسیریابی در شبکه Hermes (۱۰ پالس ساعت) از دیگر معایب این شبکه می باشد. اگر طول بسته کم باشد راندمان شبکه به شدت کاهش می یابد به عنوان مثال اگر طول بسته ها ۵ فلیت باشد در هر سوئیچ دو برابر تعداد فلیت ها پالس ساعت برای مسیریابی مصرف می شود. بنابراین تلاش شده است که این مدت زمان کاهش یابد.

الگوریتم مسیریابی در شبکه Hermes مسیریابی XY می باشد. مزیت این مسیریابی سادگی آن است ولی در حرکت بسته ها وقفه های زیادی رخ می دهد و باعث می شود شبکه در تراکم بالا به بن بست برسد برای برطرف کردن این معایب مسیریابی XY احاطه کننده مورد استفاده قرار گرفته است.

۱-۳ ساختار پایان نامه

در فصل دوم پایان نامه به معرفی سیستم روی تراشه به عنوان مقدمه ای بر شبکه روی تراشه می پردازیم. چون ایده ی شبکه روی تراشه در راستای تکامل معماری سیستم روی تراشه مطرح شده است و ضرورت شبکه روی تراشه بدون بیان معایب سیستم روی تراشه درک نخواهد شد. در پایان این فصل شبکه روی تراشه و لایه های آن تعریف می شود.

در فصل سوم ویژگی ها و نظریه های مختلف در معماری شبکه روی تراشه بیان خواهد شد. توپولوژی ها، الگوریتم های مسیریابی و نظریه های مطرح شده در چگونگی ارسال بسته در این فصل معرفی شده است.

در فصل چهارم به تحلیل نتایج شبیه سازی و مقایسه شبکه طراحی شده با شبکه Hermes پرداخته شده است.

فصل دوم:

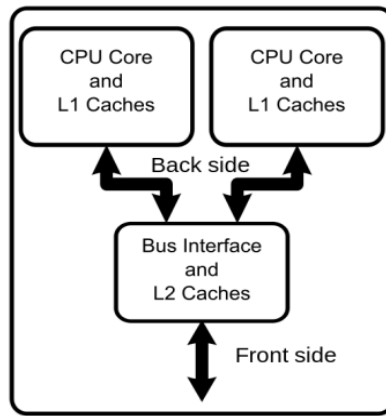
مقدمه ای بر شبکه روی تراشه

وقتی طراحان مدارهای مجتمع به این نتیجه رسیدند که با روش های معمول نمی توانند بر مشکلات دما و توان مصرفی و در نتیجه کاهش کارایی تراشه غلبه کنند ایده ی سیستم روی تراشه شکل گرفت که باعث تحول بزرگی در معماری و ساخت مدارهای مجتمع شد. افزایش دما برای ساخت تراشه هایی با سرعت بیشتر از آنچه که تا سال های ۲۰۰۲ و ۲۰۰۳ ساخته شده بودند آنقدر زیاد بود که شرکت اینتل در سال ۲۰۰۴ مجبور شد مطالعه برای ساخت تراشه های ۴ گیگا هرتزی از خانواده pentium را متوقف کند و دما به عنوان مهمترین عامل محدود کننده افزایش سرعت پردازنده ها شناخته شد [۱].

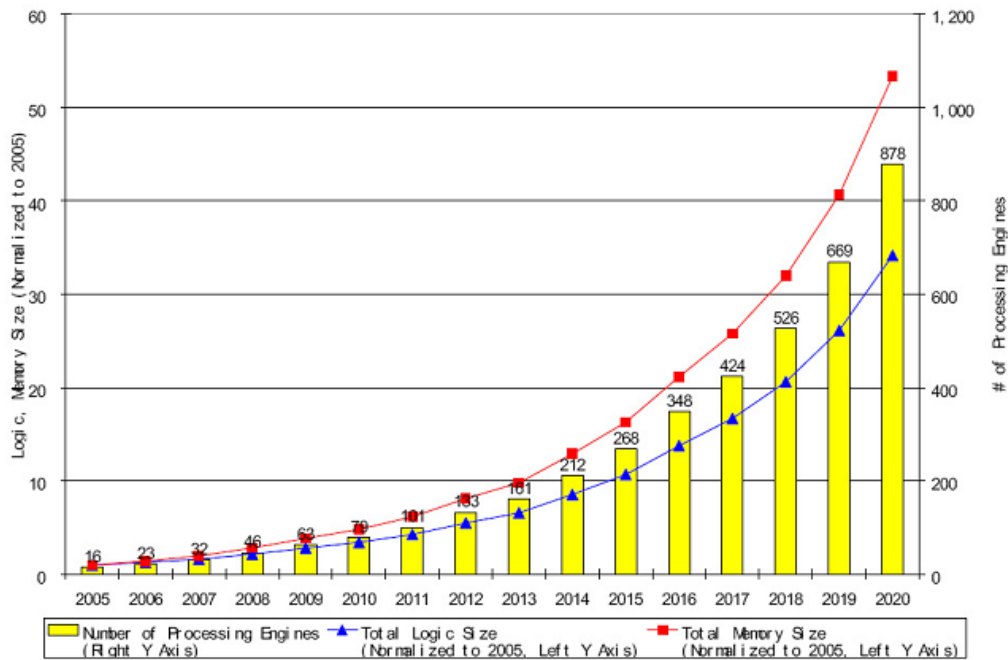
اولین بار شرکت اینتل در سال ۲۰۰۶ تراشه های خانواده Xeon و Core2 Due که در آن ها دو پردازنده در تراشه استفاده شده بود را روانه بازار کرد، که کارایی بسیار بهتر و در عین حال مصرف توان کمتری نسبت به آخرین خانواده های تک پردازنده ای یعنی پنتیوم داشتند. همانطور که در شکل (۲-۱) مشاهده می شود در این تراشه ها از یک گذرگاه برای ارتباط بین دو پردازنده استفاده می شد. بعد از آن تعداد پردازنده های داخل یک تراشه رو به افزایش گذاشت و نتیجه کار آنقدر رضایت بخش بود که تلاش شد که حتی سیستم هایی که تا آن زمان بر روی یک برد ساخته می شدند نیز در یک تراشه قرار داده شوند. این معماری به مرور زمان به نام سیستم روی تراشه یا SOC^۱ [۳] معروف شد. طبق پیش بینی که مؤسسه ITRS^۲ انجام داد نتیجه گرفت که تا سال ۲۰۲۰ تعداد بخش های مختلفی که در یک تراشه قرار خواهد گرفت به بیش از ۸۰۰ عدد خواهد رسید، نمودار این پیش بینی در شکل (۲-۲) نمایش داده شده است.

^۱ System On Chip

^۲ International Technology Roadmap for Semiconductors



شکل ۲-۱: نمایش ساختار داخلی تراشه core 2Due [۲]



شکل ۲-۲: نمایش افزایش تعداد IPها در سال‌های مختلف [۴]

بنابراین در کاربرد چند پردازنده‌ای نحوه ارتباط داخلی آن‌ها مسلماً یک کلید اساسی در کارایی کل شبکه می‌باشد و به دلیل نیاز به افزایش کارایی، یک ارتباط داخلی مناسب همواره امری ضروری می‌باشد. در سال‌های اخیر ارتباط داخلی تراشه‌ها موضوع بسیاری از تحقیقات بوده است. اکنون ارتباط و محاسبات دو جنبه

کاملاً مجزا در تحقیقات مربوط به طراحی SOC می باشند. شبکه روی تراشه یا NOC [۵] به عنوان یک نمونه جدید از SOC یک راه کار برجسته برای ارتباط داخلی تراشه شناخته می شود. در حقیقت NOC شکل مترقی و تکامل یافته ارتباط با باس اشتراکی قدیمی می باشد.

۲-۱ استانداردهای معماری ارتباط روی تراشه

طراحی سیستم روی تراشه شامل بخش‌های مختلف از قبیل پردازنده ها، حافظه ها، سخت افزارهای معمول، دستگاه های جانبی و رابط های خارجی (که بلوک های IP^۱ نامیده می شوند) تشکیل شده است، که نیاز به ارتباط هریک از آن ها با یکدیگر می باشد.

از حدود سال ۱۹۹۰ چندین استاندارد معماری ارتباط برپایه باس برای انجام ارتباط مورد نیاز در طراحی SOC پیشنهاد شده است. از استانداردهای محبوب در معماری های برپایه باس می توان از استانداردهای میکروکنترلرهای ARM (AMBA^۲) نسخه های ۲.۰ و ۳.۰، IBM CoreConnect، STMicroelectronics، STBus، Sonics SMART Interconnect، OpenCores Wishbone و Altera Avalon نام برد [۶]. در ادامه چند استاندارد پرکاربرد ارتباطی برپایه باس معرفی می شود. از آنجا که این استانداردها دائماً در حال تغییر هستند توصیف هایی که درباره این استانداردها بیان می شود به عنوان مثال هایی از ویژگی ها و توانایی ها و طرح های مورد نیاز معماری ارتباط روی تراشه برای پشتیبانی از نیازهای کاربرد مختلف SOC می باشد.

^۱ intellectual property

^۲ ARM Microcontroller Bus Architecture

۲-۱-۱ استاندارد AMBA 2.0

نسخه شماره 2.0 استاندارد AMBA [۷] امروزه برای ارتباط روی تراشه بسیار زیاد استفاده می‌شود. هدف این استاندارد یک معماری باس ویژه انعطاف پذیر است که کارایی بسیار بالایی داشته باشد در عین حال مستقل از تکنولوژی باشد سطح بسیار کمی از سلیکون را اشغال کند و استفاده مجدد از IPها را تقویت کند.

این معماری سه استاندارد باس مجزا را توصیف می‌کند:

۱. باس پیشرفته با کارایی بالا (AHB^1): که از اتصال با پهنای باند بالا، بخش‌های با فرکانس کاری بالا مانند میکروپروسور ها، کنترل کننده‌های دسترسی مستقیم به حافظه (DMA)، رابط حافظه خارجی و بلوک‌های حافظه با پهنای باند بسیار بزرگ پشتیبانی می‌کند.
۲. باس سیستم پیشرفته (ASB^2): این استاندارد کمی از AHB پیشرفته تر است به این صورت که به اجزای با فرکانس کار بسیار بالا نیز متصل می‌شود در حالیکه به طرح‌های پروتکل‌های پیشرفته استاندارد AHB نیاز ندارد.
۳. باس جانبی پیشرفته (APB^3): این استاندارد یک باس بهینه شده برای کاربردهای کم مصرف می‌باشد. این استاندارد برای کاربردهای نیازمند به تأخیر زیاد، اجزاء جانبی با پهنای باند کم مانند تایمرها، فرستنده/گیرنده‌های غیرهمزمان همه منظوره ($UART^4$)، کنترل کننده رابط کاربر (مانند صفحه کلید) طراحی شده است.

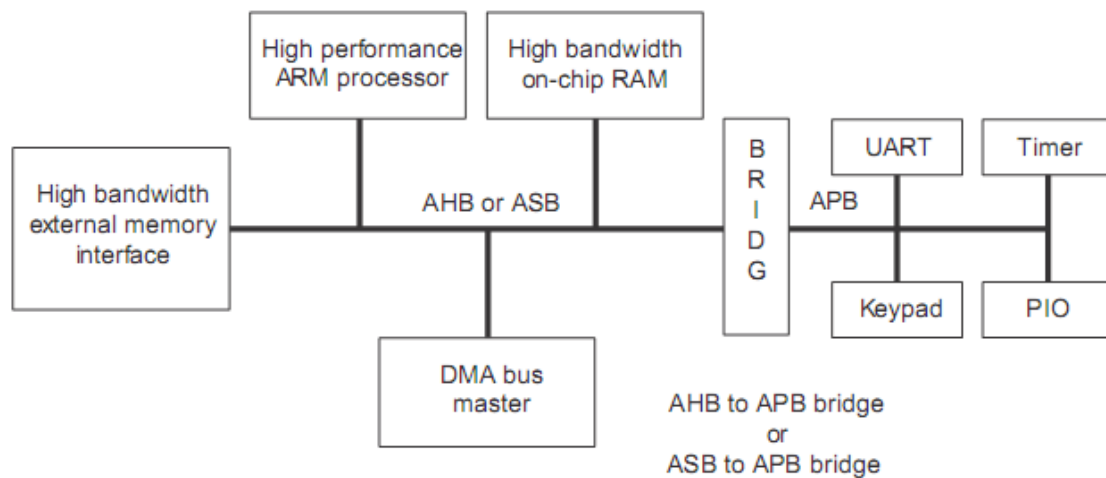
¹ Advanced high performance bus

² Advanced system bus

³ Advanced peripheral bus

⁴ universal asynchronous receivers/transmitters

شکل (۳-۲) یک نوع سیستم با معماری AMBA را نمایش می‌دهد که در آن باس‌ها در یک ساختار سلسه‌مراتبی قرار گرفته‌اند. استاندارد نوع ASB یک ساختار باس مستحکم دارد که یک ارتباط با پهنای باند بزرگ را بین بخش‌های اصلی انتقال، برقرار می‌کند. بخش پل در نوع باس با کارایی بالا برای ارتباط با نوع باس جنبی (APB) که پهنای باند کمی دارد و اغلب برای ارتباط با اجزاء جانبی با پهنای باند کم استفاده می‌شود قرار داده شده است. توجه کنید که شکل شماره (۳-۲) تنها یکی از توپولوژی‌های ممکن برای AMBA 2.0 را نشان می‌دهد.



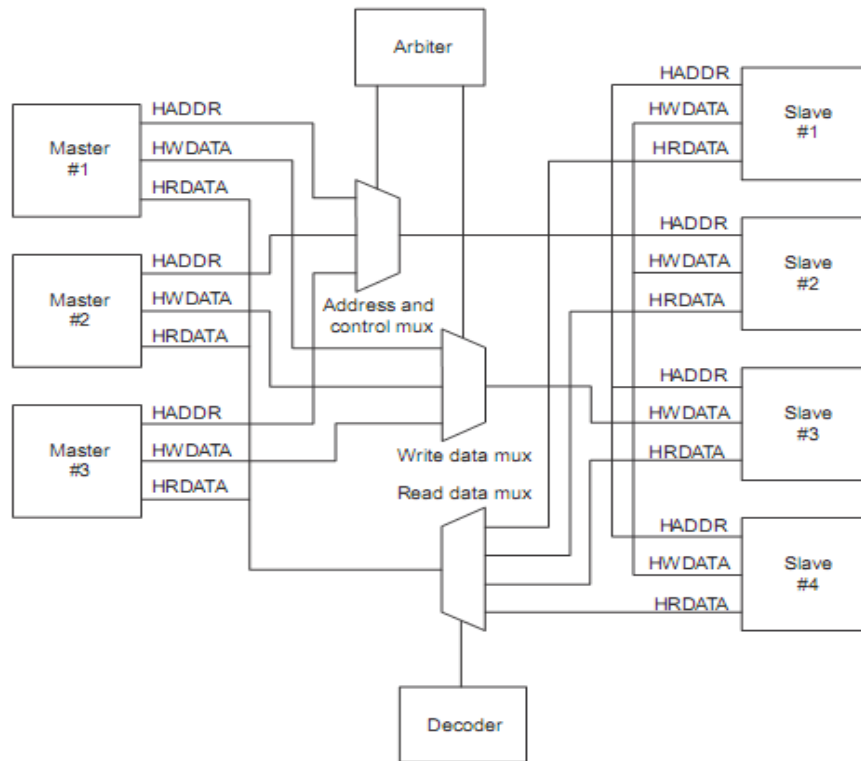
شکل ۳-۲: نمایش استاندارد AMBA 2.0 [۷]

پیکره بندی‌های دیگری از قبیل باس مشترک تنها، AHB چند لایه (یا ماتریس باس) نیز ممکن هستند که در ادامه مورد بحث قرار می‌گیرند. همچنین توجه داشته باشید که ویژگی‌های AMBA تنها در سطح معماری توصیف می‌شود و درباره ویژگی‌های الکتریکی باس که وابسته به تکنولوژی ساخت است اطلاعاتی ارائه نمی‌شود.

۲-۱-۱-۱-۱-۱ باس پیشرفته با کارایی بسیار بالا

استاندارد AHB یک باس با کارایی بالا را توصیف می‌کند که از طرح‌های پیشرفته برای انتقال با پهنای باند زیاد و تأخیر کم پشتیبانی می‌کند. استاندارد AHB می‌تواند برای اتصال چندین بخش ارباب^۱ مورد استفاده قرار گیرد همچنین می‌تواند از طرح‌های پیشرفته انتقال داده مانند عملیات پردازش موازی و حالت انتقال متوالی نیز پشتیبانی می‌کند. در این معماری یک آدرس ۳۲ بیتی توسط بخش ارباب برای ارسال آدرس بخش نوکر^۲ مورد نیاز، برای کامل کردن عمل خواندن یا نوشتن مورد استفاده قرار می‌گیرد. همه باس‌های روی تراشه به پیاده‌سازی غیر سه‌حالتی محدود می‌شوند بنابراین AHB برای خواندن و نوشتن داده‌ها، باس‌های مجزایی دارد. باس‌های داده با حداقل ۳۲ بیت سفارشی شده است اما می‌تواند هر یک از مقادیر ۸، ۱۶، ۳۲، ۶۴، ۱۲۸، ۲۵۶، ۵۱۲ یا ۱۰۲۴ (وابسته به پهنای باند کاربرد مورد نیاز، محدودیت پایه‌های بخش رابط و عرض بیت کلمه در دسترس از واحد حافظه) را می‌تواند داشته باشد. شکل (۲-۴) یک ساختار ساده با ۳ واحد ارباب و ۴ واحد نوکر را نمایش می‌دهد [۷].

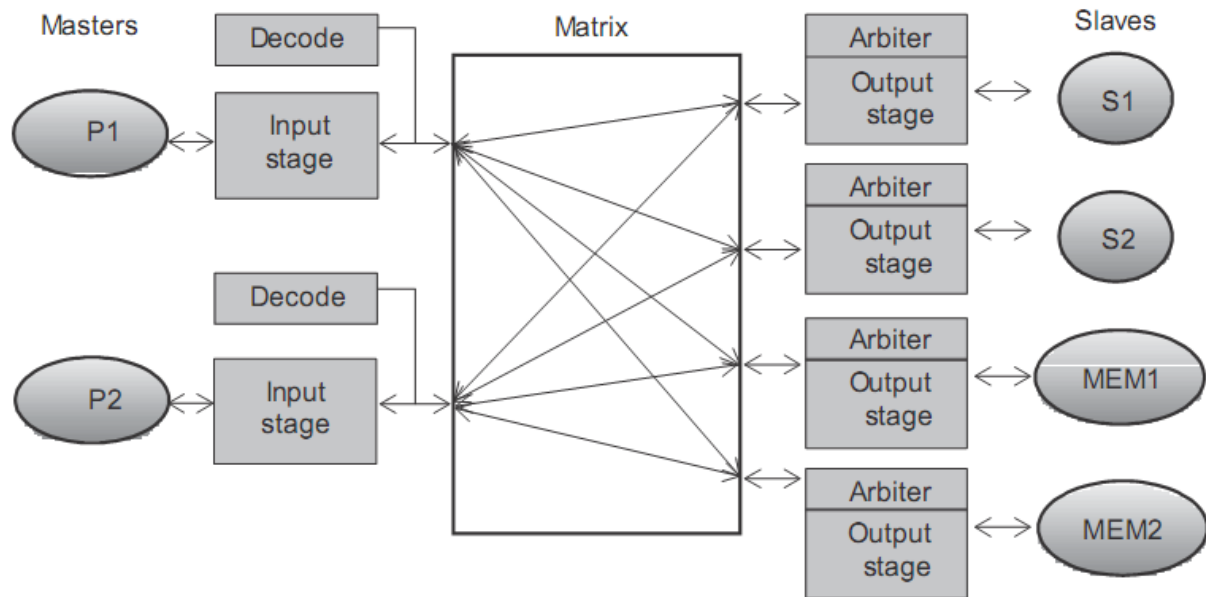
^۱Master^۲Slave



شکل ۲-۴: طرح اتصال داخلی استاندارد AHB با مالتی پالکسر

- توپولوژی ماتریسی استاندارد AHB

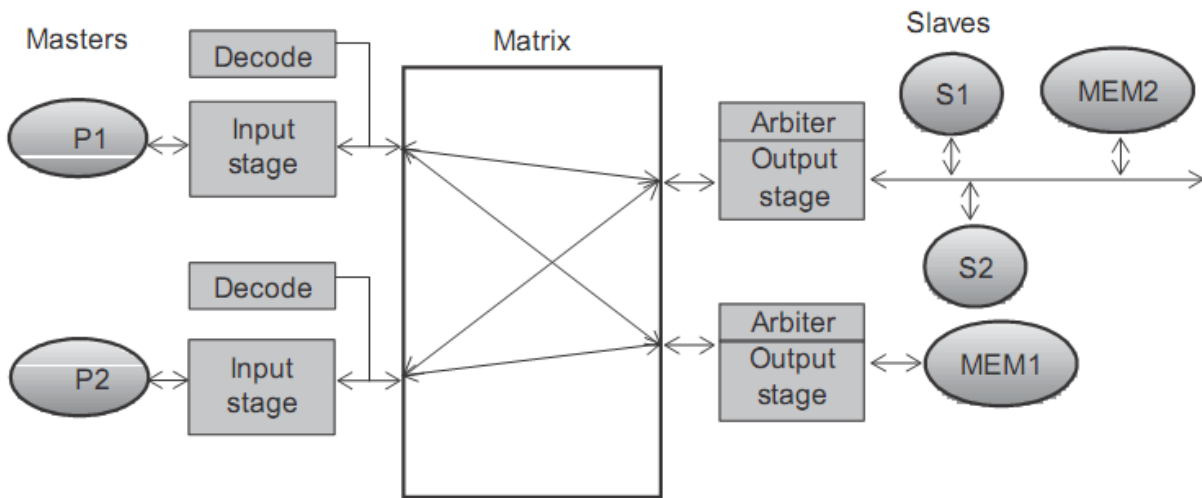
علاوه بر توپولوژی باس سلسله مراتبی پایه که در آن برای ارتباط با استاندارد APB از یک پل ارتباطی استفاده می کرد، معماری ارتباطی برپایه باس AHB می تواند توپولوژی های دیگری از قبیل توپولوژی باس سلسله مراتبی با چندین معماری AHB (همچنین APB) که از باس برای ارتباط بین آنها در میان پل ها استفاده می شود، نیز داشته باشد. برای طرح های SOC که نیاز به پهنای باند بسیار بالا و انتقال همزمان چندین داده به صورت همزمان دارند ممکن است استفاده از طرح های سلسله مراتبی مطلوب نباشد. برای این قبیل طرح ها توپولوژی باس ماتریسی AHB اغلب ساختار ارتباطی مناسب تری می باشد. شکل (۲-۵) مثالی از یک توپولوژی ماتریسی باس کامل با ۳ واحد ارباب و ۴ واحد نوکر که در آن چندین باس برای پشتیبانی از انتقال همزمان و پهنای باند بزرگ موازی شده اند را نمایش می دهد.



شکل ۲-۵: یک مثال با دو ارباب و چهار نوکر استاندارد AHB: توپولوژی ماتریس گذرگاه کامل [۷]

مرحله ورودی (Input stage) برای اجرای وقفه‌های پشت سرهم و ذخیره داده‌های ارسال شده از یک ارباب در ثبات در زمانی که نوکر مقصد از پذیرفتن آن‌ها معذور باشد مورد استفاده قرار می‌گیرد. قسمت رمزگشا (Decoder) برای تولید سیگنال انتخاب‌کننده نوکر و همچنین کنترل و خواندن داده‌های ارسال شده توسط ارباب می‌باشد. مرحله خروجی (Output Stage) آدرس را انتخاب می‌کند و داده‌های ارسال شده برای نوشتن در یک نوکر را کنترل می‌کند. این بخش تنظیم‌کننده نامیده می‌شود که یک طرح نظارت را برای زمانیکه چندین ارباب قصد دسترسی به یک نوکر را داشته باشند اجرا می‌کند. برخلاف معماری باس‌های سلسله‌مراتبی قدیمی در ماتریس باس تنظیم‌کننده در یک جا متمرکز نیست و هر ارباب یک تنظیم‌کننده متعلق به خود دارد. یک بازنگری در طرح ماتریس باس کامل نشان می‌دهد که هر ارباب و نوکر به یکدیگر متصل شده‌اند در نتیجه حجم بسیار زیادی از باس‌ها و سیم‌ها را خواهد داشت. بنابراین در این پیکره بندی، کارایی بالا در قبال هزینه بالا، مصرف توان بالا و اشغال شدن سطح بزرگی از تراشه خواهد بود. برای سیستمی که کارایی کمتری مورد نیاز است توپولوژی ماتریس باس غیرکامل می‌تواند به کار رود. شکل (۲-۶) یک نمونه از این این

توپولوژی را که در آن برای کاهش تعداد باس ها برای دسته ای از بخش ها یک باس اشتراکی استفاده شده است را نمایش می دهد.



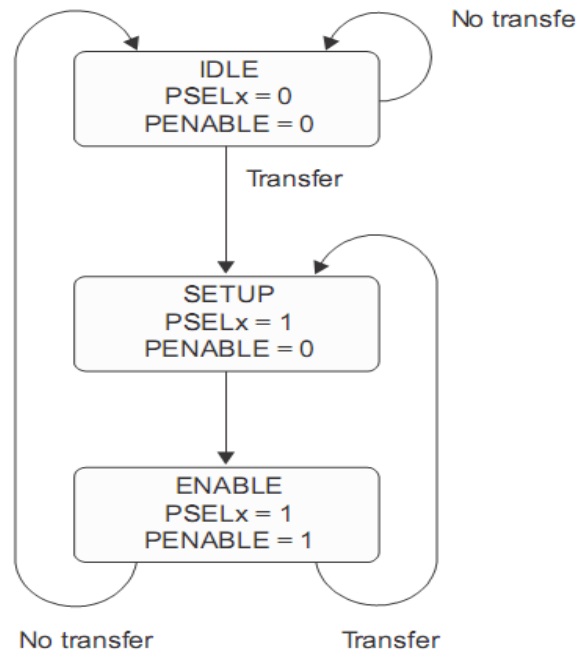
شکل ۲-۶: توپولوژی ماتریس باس غیر کامل [۷]

توپولوژی ماتریس باس غیر کامل در مقایسه با ماتریس باس کامل به علت برخورد داده ها در باس اشتراکی پهنای باند کمتری خواهد داشت. پیکره بندی ماتریس باس غیر کامل مصرف توان کمتری دارد و سطح کمتری هم بر روی تراشه اشغال می کند که ویژگی های مورد نظر در معماری ارتباط . در مرجع شماره [۸] نشان داده شده است در طراحی یک SOC چگونه یک ماتریس باس کامل AHB می تواند با کاهش دادن تعداد باس ها به یک توپولوژی غیر کامل ساده شود در حالی که سایر قیدهای مربوط به کارایی مطلوب و مناسب پابرجا باشد.

۲-۱-۱-۲ باس جانبی پیشرفته

استاندارد باس APB یک باس بهینه برای کاهش پیچیدگی ارتباط و مصرف توان کم می باشد. این باس به منظور ارتباط با AHB (با یک پل) برای اتصال با وسایل جانبی با پهنای باند کم (یا تأخیر زیاد) که نیاز به طرح های پیشرفته ارتباط ندارند استفاده می شود. استاندارد APB تنها اجازه یک ارسال خط لوله ای را می دهد. یک

پل استاندارد APB را به AHB اتصال می دهد. استاندارد APB با فرکانس کار کمتری عمل می کند که این خود به مصرف بیشتر توان منجر می شود. شکل (۷-۲) دیاگرام فعالیت استاندارد APB را نمایش می دهد.



شکل ۷-۲: دیاگرام حالت استاندارد APB [۷]

مرحله IDLE مرحله پیش فرض می باشد که وقتی در خواست ارسال از AHB وجود نداشته باشد، باس APB در این حالت می ماند. وقتی یک درخواست از باس AHB از طریق پل رابط به باس APB می رسد، باس APB به حالت State رفته و در آنجا سیگنال مناسب برای انتخاب واحد نوکری که باید در انتقال شرکت کند فعال می شود. باس APB برای یک سیکل در حالت SETUP می ماند و در این مدت آدرس بخش جانبی مقصد رمزگشایی می شود سپس باس APB به حالت ENABLE رفته و در لبه بالا رونده پالس ساعت بعدی سیگنال PENABLE را فعال کرده تا نشان دهد شرایط برای انتقال فراهم می باشد. معمولاً این مرحله بیش از یک پالس ساعت طول می کشد، بعد از اتمام کار اگر درخواست انتقال دیگری باشد به حالت SETUP برگشته تا آن را بررسی کند در غیر اینصورت به حالت IDLE برگشته و منتظر درخواست از باس AHB می ماند.

۲-۱-۲ استاندارد ABM 3.0

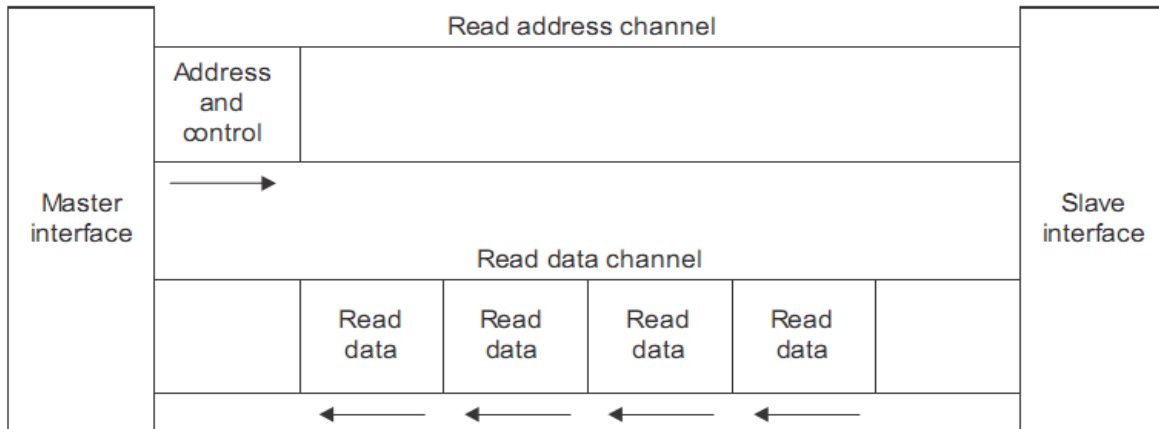
ویژگی‌های استاندارد AMBA 3.0 یک ارتباط قابل گسترش پیشرفته (AXI^۱) را معرفی می‌کند [۷]. این استاندارد، استاندارد باس AHB را با ویژگی‌های پیشرفته‌ای توسعه می‌دهد تا بتواند از طرح‌هایی که در تولیدهای بعدی MPSOC برای کارایی بسیار بالا وجود داشت پشتیبانی کند. هدف از پروتکل باس AXI دستیابی به فرکانس کار بسیار بالا بدون استفاده از پل‌های ارتباطی پیچیده، انعطاف پذیری در واحدهای ارتباطی و سازگاری با استاندارد AHB و APB از پروتکل AMBA2.0 می‌باشد.

۲-۱-۲-۱ ارتباط قابل گسترش پیشرفته (AXI)

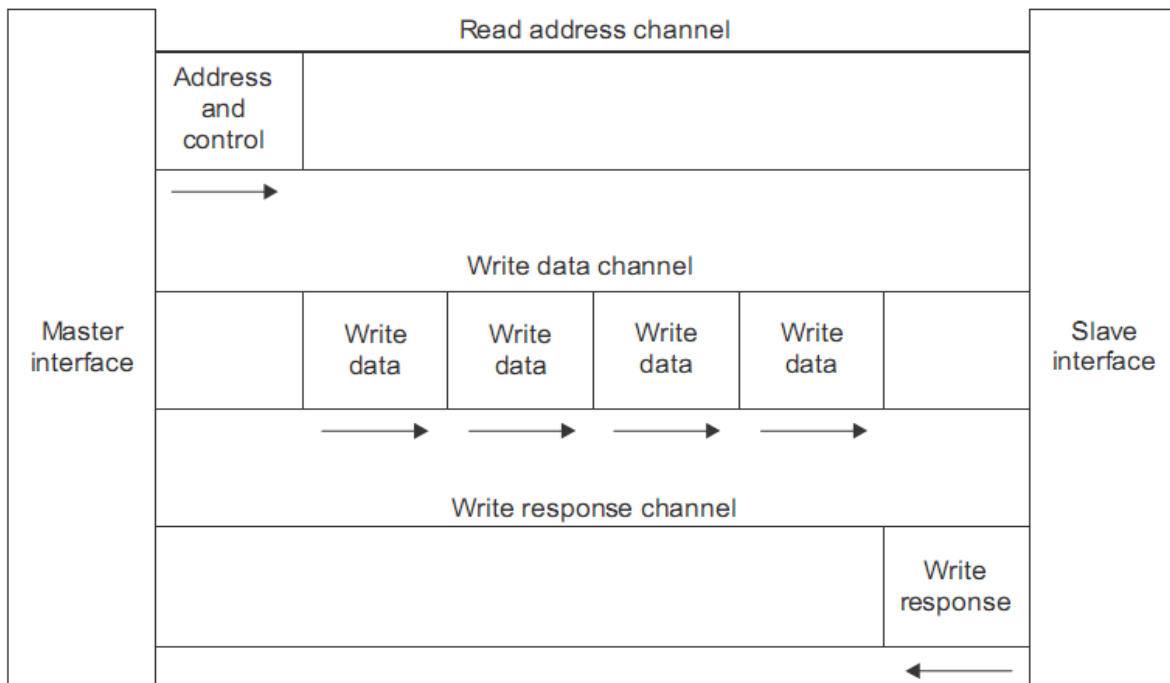
استاندارد AXI یک انتقال خط لوله‌ای داده مانند استاندارد باس AHB پیشنهاد می‌کند اما با ویژگی‌های پیشرفته‌تر و کامل‌تر.

استاندارد AXI یک معماری برپایه باس سطح بالا برای ارتباط بین ارباب‌ها و نوکرها بر روی یک باس را توصیف می‌کند. پنج کانال مجزا تعریف شده‌اند که عبارتند از کانال خواندن آدرس (read address)، خواندن داده (read data)، آدرس نوشتن (write address)، نوشتن داده (write address) و نوشتن پاسخ (write response). دقیقاً مشابه باس داده در استاندارد AHB عرض کانال در AXI می‌تواند بین ۸ تا ۱۰۲۴ بیت باشد. شکل (۲-۷) کانال‌های خواندن را نشان می‌دهد. آدرس و اطلاعات کنترلی برای انجام عمل خواندن در انتقال توسط واحد ارباب به کانال خواندن ارسال می‌شود در حالی که داده‌های مورد نظر و اطلاعات پاسخ از واحد نوکر در کانال خواندن داده دریافت می‌شود. شکل (۲-۸) کانال‌های نوشتن داده را نشان می‌دهد.

^۱ Advanced Extensible Interface



شکل ۲-۸: کانال های خواندن آدرس و خواندن داده در معماری AXI [۷]



شکل ۲-۹: کانال های نوشتن داده و پاسخ نوشتن [۷]

آدرس و اطلاعات کنترلی برای انجام عمل نوشتن به کانال آدرس نوشتن ارسال می شود در حالی که داده هایی که باید نوشته شوند به کانال نوشتن آدرس منتقل می شوند. یک سیگنال یک بیتی به ازای هر ۸ بیت نوشته شدن داده ها فعال می شود و نشان دهنده این است که کدام داده ها بر روی باس داده معتبر هستند. این قابلیت برای زمان هایی که بین داده های منتقل شده به باس و عرض باس ناهماهنگی وجود دارد مفید است.

یک کانال مجزا برای پاسخ نوشتن باعث می شود که نوکر بتواند به عمل نوشتن پاخ دهد. سیگنالی که نشان دهنده کامل انجام شدن عمل نوشتن می باشد فقط یک بار بعد از انتقال همه داده ها (نه بعد از انتقال هر داده) فعال می شود. وجود پنج کانال مجزا موجب انعطاف پذیری در پیکره بندی طرح می شود و می تواند با یکی از روش های زیر پیاده سازی شود:

۱- باس آدرس مشترک و باس داده مشترک ($SASD^1$): یک باس آدرس مشترک با یک باس داده دو طرفه که هم عمل خواندن را انجام می دهد و هم عمل نوشتن را جفت می شود. این قبیل پیکره بندی برای سیستم های کوچک و پیچیدگی کم مفید است.

۲- باس آدرس مشترک و چند باس داده ($SAMD^2$): یک باس آدرس مشترک با باس های مجزای یک طرفه که فقط عمل خواندن یا نوشتن را انجام می دهند جفت می شود. از آنجا که پهنای باند (عرض باس) معمولاً کمتر از باس های داده می باشد (چون فقط یک آدرس برای انتقال متوالی داده نیاز است) پیچیدگی اتصالات داخلی کمتر می شود در حالی که همان کارایی را دارد.

۳- چند باس آدرس و چند باس مشترک ($MAMD^3$): استفاده از چند باس آدرس مجزا برای انجام عمل خواندن یا انجام نوشتن که با چند باس مجزایی که آن ها نیز فقط برای انجام عمل خواندن یا نوشتن هستند جفت شده اند. این پیکره بندی در اتصالات داخلی پیچیدگی زیادی دارد ولی بهترین کارایی را خواهد داشت.

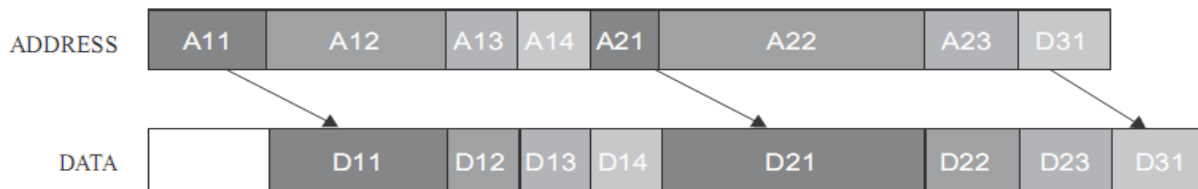
¹ Shared address bus and shared data buses

² Shared address bus and multiple data buses

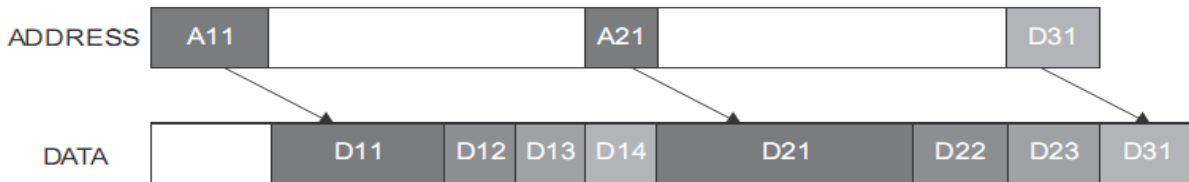
³ Multiple address buses, multiple data buses

استفاده از چند باس آدرس و چند باس داده اجازه می‌دهد که چند عمل خواندن و نوشتن به طور مستقل از یکدیگر انجام شوند که برای طرح‌های SOC با کارایی بالا بسیار مفید خواهد بود. در حالی که استاندارد باس AHB تنها از حالت یک باس آدرس و چند باس داده پشتیبانی می‌کند.

یک تفاوت مهم بین AXI و AHB روش آدرس دهی در هنگام انتقال متوالی داده‌ها می‌باشد. در AHB هر انتقال داده پشت سرهم نیاز به یک آدرس دارد تا به آن آدرس منتقل شود در حالی که در AXI تنها نیاز به آدرس اولین داده می‌باشد تا به صورت پیوسته و پشت سرهم ارسال شوند. شکل (۲-۱۰) انتقال داده‌های پیوسته را نشان می‌دهد که در آن برای ارسال هر داده یک آدرس نیز مورد نیاز است. شکل (۲-۱۱) انتقال داده‌های پیوسته را در استاندارد AXI نشان می‌دهد که در آن فقط نیاز به ارسال آدرس اولین داده می‌باشد. محاسبه آدرس داده‌های بعدی برعهده واحد نوکر می‌باشد.



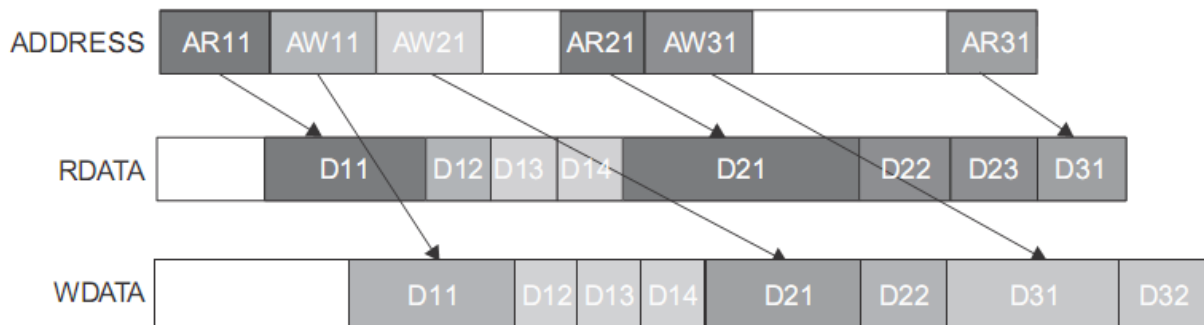
شکل ۲-۱۰: انتقال داده‌های پیوسته در استاندارد AHB [۷]



شکل ۲-۱۱: انتقال داده‌های پیوسته در AXI [۷]

چون در استاندارد AXI فقط یک آدرس برای هر مجموعه داده پیوسته ارسال می‌شود باس آدرس برای انجام تراکنش‌های دیگر آزاد است. شکل (۲-۱۲) نشان می‌دهد که در یک پیکره بندی یک باس آدرس مشترک و

چند باس داده در استاندارد AXI چگونه عمل های خواندن و نوشتن به طور همزمان می توانند انجام شوند اما در استاندارد AHB باس آدرس باید منتظر بماند که یک انتقال به طور کامل انجام شود و انتقال بعدی را شروع کند. یک پیشرفت دیگر در استاندارد AXI در مقایسه با استاندارد AHB پشتیبانی از انواع دیگری از نوع داده های پیوسته است. استاندارد AXI نه تنها از انواع افزایشی که در استاندارد AHB ارائه شده است پشتیبانی می کند بلکه از حالت پیوسته ثابت نیز پشتیبانی می کند. در این حالت آدرس هر داده نیز به صورت پیوسته و پشت سرهم باقی می ماند. این نوع برای دسترسی تکراری به یک مکان مانند انتقال داده به یک ورودی/خروجی جانبی FIFO بسیار مفید است.



شکل ۲-۱۲: انجام همزمان عمل خواندن و نوشتن در استاندارد AXI [۷]

یک ویژگی مهم دیگر استاندارد AXI پشتیبانی از اجرای تراکنش خارج از نوبت یا تراکنش^۱ OO می باشد این ویژگی یک ویژگی پیشرفته است که قابلیت عبور دهی داده ها را حداکثر کرده و راندمان سیستم را افزایش می دهد. واحدهای ارباب توانایی ارسال چند آدرس مهم را دارند، این بدین معنی است که آدرس یک عمل انتقال بدون اینکه منتظر شود آخرین تراکنش به طور کامل انجام شود می تواند ارسال شود و عملیات جدید آغاز شود. عمل خواندن و نوشتن با فعال کردن ID ها توسط واحد ارباب کامل می شود. مشخصات AXI توسط یک

¹ out-of-order

رهنمون مطمئن برای کنترل ترتیب تراکنش راهنمایی می شود. مرتب سازی تراکنش های مورد نظر از واحدهای ارباب مختلف هیچ محدودیتی ندارد و به هر طریقی می تواند انجام شود.

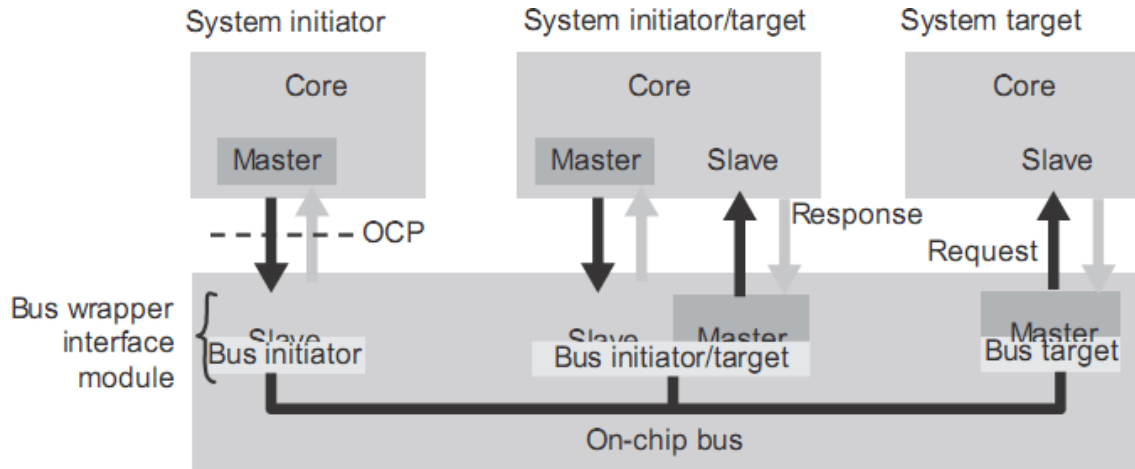
۳-۱-۲ استاندارد OCP^۱

استاندارد OCP نسخه ۲.۰ که برپایه استاندارد ارتباطی سوکت می باشد بین هسته های IP یک ارتباط با کارایی بالا، همزمان و مستقل از معماری باس را برقرار می کند [۹]. این استاندارد با اجازه دادن مستقل بودن IP از معماری و طراحی سیستم، قابلیت استفاده مجدد از IP را ترویج می دهد. استاندارد OCP بسیار انعطاف پذیر بوده و می توان برای استفاده از ویژگی های لازم برای برقراری ارتباط بین دو جزء، آن را بهینه سازی کرد که باعث صرفه جویی در سطح تراشه می شود. اساساً OCP یک ارتباط نقطه به نقطه را بین دو جزء تعریف می کند، یکی از دو جزء باید به عنوان ارباب و دیگری به عنوان نوکر باشد. مهمترین ویژگی های این استاندارد عبارتند از [۹]:

- ارتباط نقطه به نقطه همزمان
- معماری مستقل از باس
- پشتیبانی از ارسال موازی
- پشتیبانی از ارسال متوالی و پشت سرهم
- سیگنال های تنظیم جریان داده (آدرس، داده، کنترل) برای پیاده سازی کارآمد
- سیگنال های جانبی برای پشتیبانی از ارتباطات اضافی

^۱ Open Core Protocol

شکل (۲-۱۳) یک مثال ساده از یک سیستم SOC با سه هسته IP با استاندارد OCP را نمایش می دهد که به باس روی تراشه متصل شده اند. باس روی تراشه می تواند به هریک از معماری های ارتباط مبتنی بر باس روی تراشه از قبیل AMBA 2.0/3.0، CoreConnect یا STBus باشد [۱۰].



شکل ۲-۱۳: مثالی از پیکره بندی سیستم با استاندارد ارتباطی OCP برپایه سوکت [۹]

ماژول ارتباطی برای ترجمه و تطبیق سیگنال های ارتباطی IP برای باس روی تراشه مورد نیاز است. واحد ارتباطی باید به عنوان طرف مکمل اتصال OCP نقطه به نقطه برای هر IP که به باس متصل می شود عمل کند به اینصورت که برای IP ارباب باید به عنوان نوکر و برای IP نوکر باید به عنوان ارباب عمل کند. انتقال داده در چنین سیستمی به صورت زیر انجام می شود:

ابتدا IP ارباب (آغاز کننده) اطلاعات کنترلی، داده و آدرس را با سیگنال های ارتباطی OCP به واحد ارتباطی IP نوکر (هدف) ارسال می کند. واحد ارتباطی درخواست OCP را به یک درخواست باس روی تراشه تبدیل می کند و به مقصد انتقال می دهد. درخواست توسط واحد ارتباطی مقصد دریافت می شود و آن را از درخواست باس روی تراشه به یک درخواست OCP تبدیل می کند.

۲-۱-۴ استاندارد ارتباط بخش مجازی (VCI)^۱

VCI یکی دیگر از استانداردهای نقطه به نقطه، همزمان و برپایه استاندارد سوکت می باشد. این استاندارد سه نوع ارتباط مختلف که در میزان پیچیدگی با یکدیگر تفاوت دارند را تعریف می کند [۹]:

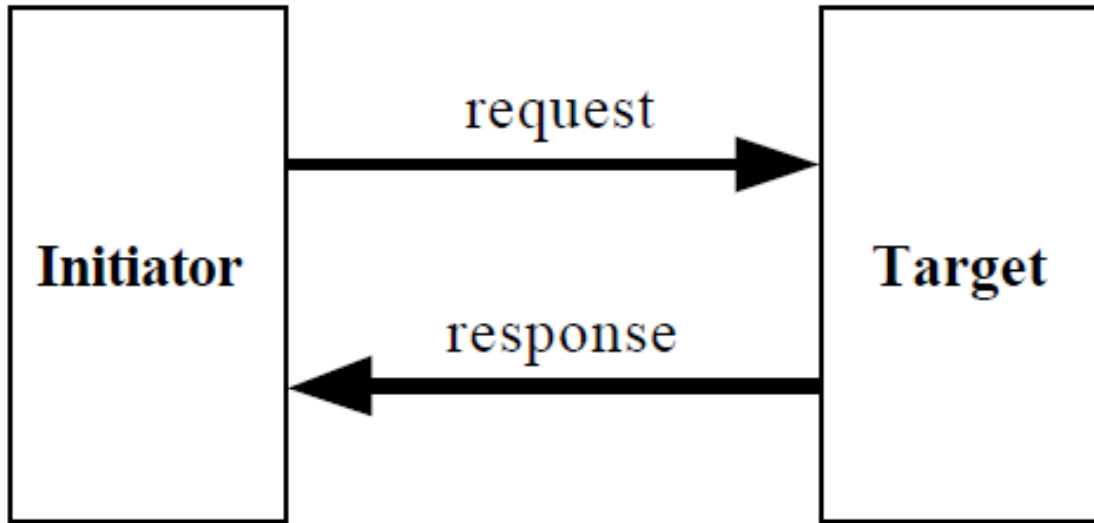
۱. VCI جنبی: این استاندارد یک ارتباط ساده دست تکان دهی برای انتقال داده، با پشتیبانی از انتقال متوالی و پشت سرهم، باس آدرس ۶۴ بیتی و باس آدرس ۳۲ بیتی.

۲. VCI پایه: این نوع علاوه بر پشتیبانی از VCI جنبی از حالت های متوالی و پشت سرهم اضافی، حالت های انتقال داده اضافی و باس داده ۱۲۸ بیتی نیز پشتیبانی می کند.

۳. VCI پیشرفته: این نوع نیز علاوه بر پشتیبانی از VCI پایه دارای حالت های انتقال داده اضافی می باشد و از طرح های پیشرفته از قبیل تراکنش OO به طور کامل پشتیبانی می کند.

همانطور که می توان مشاهده کرد هر یک از انواع ارتباطی که در بالا معرفی شد نسبت به نوع بالاتر خود کامل تر و پیشرفته تر می باشد. ارتباط هایی که توسط استانداردهای معماری باس AMBAAP ، AHB و AXI توصیف می شود را می توان متعلق به یکی از سه نوع دانست. در واقع VCI تشابه زیادی با استاندارد ارتباطی OCP 2.0 دارد ولی برخلاف آن تنها دارای سیگنال های جریان داده می باشد، درباره مسائل مربوط به تست و کنترل آن بحث نمی کند.

^۱ Virtual Component Interface



شکل ۲-۱۴: نمایش استاندارد OCP

۲-۱-۵ استاندارد ارتباطی^۱ DTL فیلیپس

استاندارد DTL فیلیپس نوع دیگری از پروتکل انتقال داده همزمان نقطه به نقطه را تعریف می کند [۱۲]. این استاندارد ارتباطی، از همه سیگنال های اصلی ارسال داده ها به صورت تکی یا توده ای و سیگنال های خطا پشتیبانی می کند. علاوه بر آن یک پروتکل ارتباطی DTL توسعه یافته نسبت به پروتکل اصلی یک پروتکل برای کاربرد خاص را توصیف می کند. سیگنال هایی که برای توسعه به کار می روند عبارتند از:

- حالت های آدرس دهی: آدرس دهی مخفی، ثابت یا کاهشی برای انتقال توده ای.
- عملیات بلوکی دو بعدی: این عملیات وقتی مفید است که داده های ذخیره شده در حافظه یک فضای دوبعدی بزرگ را نمایش دهند.
- عملیات ایمن: برای نشان دادن اینکه یک تراکنش خاص با موفقیت انجام شده است یا خیر.

¹ Device Transaction Level Protocol

• مدیریت بافر: این سیگنال برای اینکه به یک بخش اجازه دهد که داده‌ها را در بافر قرار دهد یا از آن خارج کند و یا برای اطمینان از اینکه داده‌ها به مقصد رسیده‌اند یا خیر به کار می‌رود.

چهار کاربرد اصلی متناسب با نوع ترافیک توسط DTL پشتیبانی می‌شود. هر یک از این چهار نوع نیاز به سیگنال‌هایی دارد که باید در بخش رابط پیاده‌سازی شود. این چهار کاربرد و نوع اصلی عبارتند از:

۱. نگاشت حافظه ورودی/خروجی (MMBD): این نوع برای پهنای باند کم و کنترل تأخیر بحرانی شبکه.
۲. نگاشت حافظه داده‌های بلوکی (MMBD): برای انتقال یک بلوک از داده بین CPU (یا هر بخش دیگری) و حافظه به کار می‌رود. این نوع ترافیک هر دو نوع پهنای باند و تأخیر باشد.
۳. نگاشت حافظه داده‌های رشته (MMSD): برای انتقال متوالی داده‌ها بین یک جزء و حافظه. در این مدل ترافیک معمولاً پهنای باند بحرانی می‌باشد در حالی که ممکن است تأخیر از اهمیت کمتری برخوردار باشد.
۴. داده‌های رشته‌ای جفت-جفت (PPSD): برای انتقال دنباله‌ای از داده‌ها بین دو قسمت می‌باشد. همانند MMSD ترافیک ناشی از پهنای باند بحرانی می‌باشد در حالی که تأخیر ممکن است از اهمیت کمتری برخوردار باشد.

استاندارد DTL (بر خلاف OCP 2.0 و VCI که آزاد هستند) توسط شرکت فیلیپس ایجاد شده است معمولاً در پلت فرم Nextperia فیلیپس استفاده می‌شود.

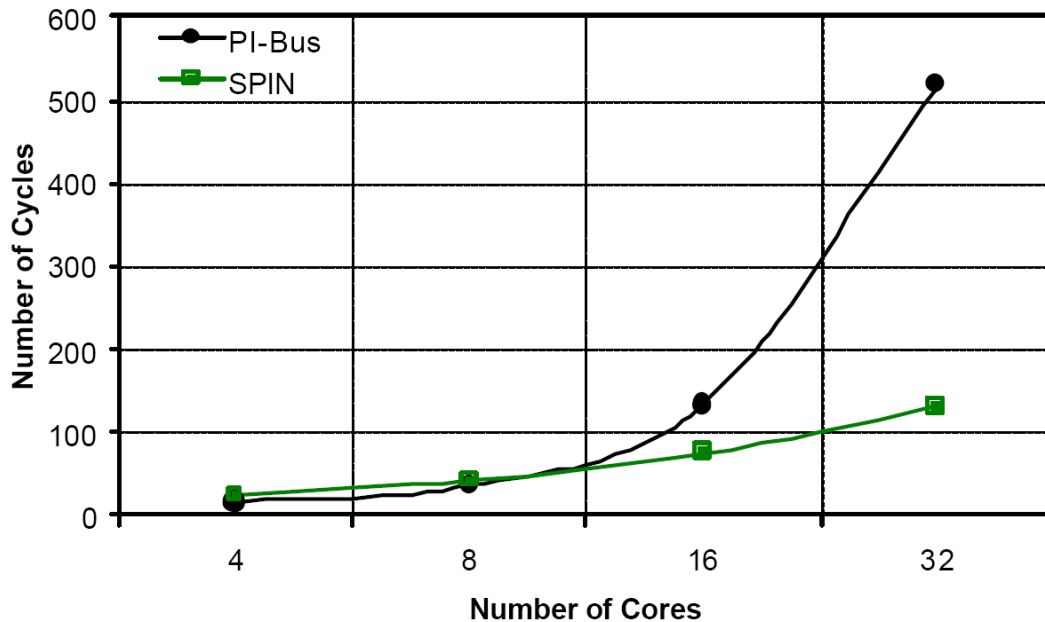
اگرچه در حال حاضر در این استاندارد از بیشتر طرح‌های پیشرفته مانند تراکنش OO، مدیریت حافظه کش و خواندن بافر پشتیبانی نمی‌شود، از طرح‌های آینده است که در نسخه‌های جدید DTL از این موارد نیز پشتیبانی شود.

۲-۲ معایب استفاده از باس

استفاده از باس دارای معایبی می باشد که استفاده از آن به خصوص در مواردی که تعداد IPها زیاد باشد مناسب نیست. در این بخش به بررسی این معایب می پردازیم.

۱-۲-۲ عدم مقیاس پذیری

در معماری شبکه روی تراشه برای دست یابی به کارایی بالاتر باید تعداد IPها را در یک تراشه افزایش داد. ارتباط‌های داخلی قدیمی فقط برای تعداد محدودی از IPها مناسب هستند و با افزایش تعداد IP دیگر پاسخگو نیستند. این عدم کارایی برای تعداد بیشتری از IPها در شکل زیر نمایش داده شده است. شکل (۲-۱۵) میزان تأخیر را در برابر افزایش تعداد IPها برای دو نوع متفاوت از اتصالات داخلی نشان می دهد. معماری نمودار مشکی رنگ از نوع سیستم روی تراشه و نمودار سبز رنگ از نوع شبکه روی تراشه می باشند (شبکه SPIN).



شکل ۲-۱۵: نمایش تأخیر ایجاد شده در برابر افزایش IPها [۴]

همانطور که مشاهده می شود در سیستم روی تراشه با افزایش تعداد IPها میزان تأخیر به صورت نمایی افزایش می یابد در حالی که در شبکه SPIN تغییر زیادی رخ نمی دهد.

مواردی که در مقیاس پذیری تراشه اثر گذار معرفی شده اند عبارتند از: ارتباط همزمان و کیفیت خدمات.

۲-۱-۲-۱ ارتباط همزمان

عملکرد ارتباط برای کاربردهایی که تراشه های مقیاس بزرگ با کارایی بالا مورد نیاز است یک نقش کلیدی در کارایی کل تراشه ایفا می کند. ممکن است تراشه نیاز به یک اتصال داخلی متراکم با چند صد گیگا بایت در ثانیه برای انتقال داده ها نیاز داشته باشد. معماری بر پایه NOC با تعداد بسیار زیادی ارتباط داخلی همزمان می تواند به طور کامل این نیاز را بر طرف نماید ولی در معماری های قدیمی مانند باس اشتراکی، باس توسط همه اجزاء اضافه شده به اشتراک گذاشته می شود و ارتباط همزمان امکان ندارد.

در حقیقت شبکه روی تراشه نتیجه پیشرفت تدریجی در افزایش قابلیت استفاده مجدد و مقیاس پذیری در معماری اتصالات داخلی گذرگاه سلسله مراتبی است. سیم ها موازی می شوند و پل ها به مسیریاب تبدیل می شوند.

۲-۱-۲-۲ کیفیت خدمات

توانایی یک شبکه در فراهم کردن خدمات تضمین شده در اتصالات ویژه اغلب کیفیت خدمات یا QoS¹ نامیده می شود. هدف اصلی QoS این است که وقتی شبکه به نقطه اشباع می رسد عملکرد و تأخیر شبکه به محدوده خاصی تضمین شود. برای مثال یک رشته از داده های متعلق به یک ویدئو که از یک دوربین با کیفیت بسیار بالا به یک انکودر MPEG ارسال می شود نیاز به یک عملکرد با یک تأخیر کم، پایدار و قابل پیش بینی دارد. اگر در

¹ Quality Of Service

زمان بین فریم‌ها، یک محدودیت قطعی وجود داشته باشد کیفیت خدمات تضمین شده نیست. کیفیت خدمات یا QoS سطح التزام برای تحویل بسته را مشخص می‌کند، چنین تعهدی می‌تواند نتیجه درستی برای کامل بودن تراکنش دریافت و ارسال و مرزهای کارایی باشد. QoS پیوند مستقیمی با پهنای باند، تأخیر و حرکت نامنظم اتفاقی دارد چون درستی و تعهد دو نیاز اساسی برای تحویل پیام بر روی تراشه می‌باشد بنابراین بر مقیاس‌پذیری نیز اثر گذار خواهد بود.

QoS به عنوان تعیین کارها و خدماتی که توسط شبکه در رابطه با درخواست‌های هسته‌ها انجام می‌شود نیز تعریف می‌شود [۱۳] که شامل دو بخش می‌شود: (۱) تعریف خدمات مورد نظر با یک مشخصات مطمئن، خدمات می‌تواند تأخیر کم، قدرت اجرای بالا، توان مصرفی کم و ... باشد. (۲) برقراری تعادل بین کار درخواستی هسته و خدمات در دسترس و قابل ارائه از شبکه است. در [۱۴] دو دسته اساسی برای QoS تعریف شده است، یکی خدمات بهترین تلاش (BE^1) و که تعهد و تضمین نداشت و دیگری خدمات تضمین شده (GS^2). در استفاده از ترکیب BE و GS مطرح می‌شود. اساساً GS باعث افزایش قابلیت پیش‌بینی‌پذیری وضعیتی می‌شود که اغلب مطلوب است.

اگر بخواهیم دقیق‌تر صحبت کنیم BE در مواردی که به هیچ وجه نمی‌توان به تضمینی دست یافت ارجحیت دارد. در بیشتر کارهای انجام شده در زمینه NOC، در ترافیک شبکه BE فقط درستی و انجام کامل تراکنش تضمین می‌شوند در حالی که در GS یک تضمین اضافی نیز به دست می‌آید: تضمین کارایی تراکنش. به منظور دست‌یابی به ضمانت بالا، ارتباط GS باید از دیگر ترافیک‌ها در شبکه مستقل باشد. این نیازمند مسیریابی اتصال جهت‌دار می‌باشد. ارتباط می‌تواند به عنوان مدار مجازی معرفی شود که مستقل از واحدهای پردازنده به کار می‌روند بنابراین با یکدیگر درگیری ندارند. مدار مجازی می‌تواند با کانال مجازی، شکاف زمانی، ساخت سوئیچ‌های موازی و ... پیاده

¹ Best-effort

² guaranteed services

سازی شوند. با افزایش پیچیدگی سیستم نیاز به GS نیز افزایش می‌یابد بنابراین تعدادی از مدارهای مجازی و واحدهای پردازنده نیاز به حمایت دارند.

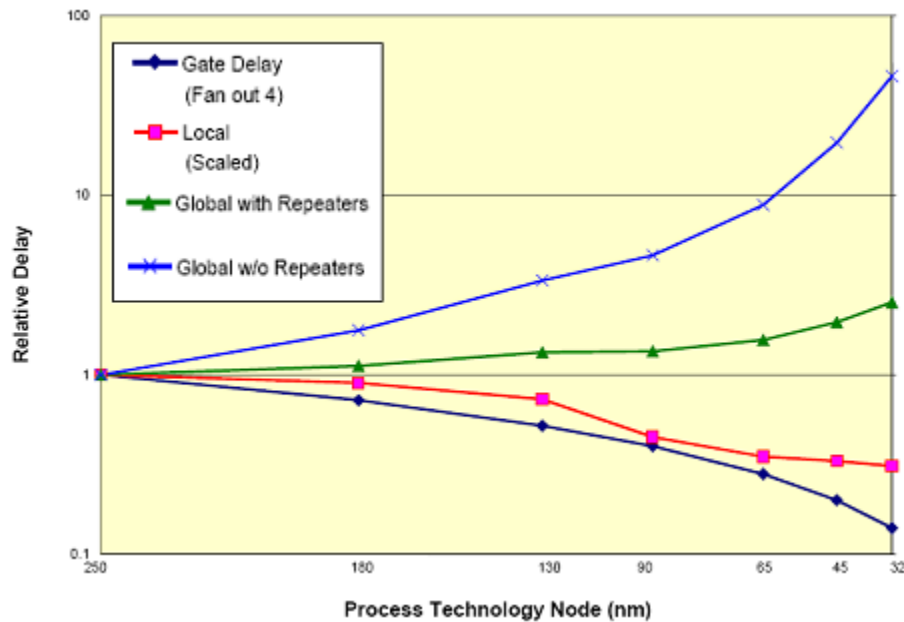
۲-۲-۲ مشکلات فیزیکی

ساخت یک تراشه در مقیاس بزرگ و پیچیده با معماری SOC اگر غیر ممکن نباشد، بسیار مشکل است که با یک مدل کاملاً عملی ساخته شود. گذشته از عامل دما، تغییر ویژگی‌های فیزیکی و الکتریکی نیز عامل دیگری است که در تکنولوژی SOC محدودیت ایجاد می‌کند. یک سیستم ممکن است مدل‌های مختلفی برای ساخت داشته باشد. ویژگی‌های تفکیک‌کننده کلید مدیریت و سازمان‌دهی میزان پیچیدگی در مدل‌سازی و طراحی هستند. معماری به عنوان یک سطح انتزاع و تفکیک‌کننده بالا، اکنون یک عامل مهم برای طراحی به شمار می‌رود. در حالیکه مدل انتزاعی و سنتز خودکار طراحی سیستم‌های پیچیده را ممکن کرده‌اند اما تغییرات پارامترهای الکتریکی و فیزیکی افزایش یافته است.

۲-۲-۲-۱ تأخیر سیم

تکنولوژی کوچکتر کردن ابعاد پردازشگرها در اندازه میکرون تعادل بین تأخیر گیت‌ها و تأخیر سیم‌ها را برهم زده است [۱۴]. درحالی‌که تأخیر گیت‌ها کاهش می‌یابد ولی تأخیر سیم‌ها به علت افزایش مقاومت آن‌ها افزایش می‌یابد. از آنجا که طول سیم‌های محلی معمولاً نسبت به مقیاس گذشته کوچکتر می‌شود اثر تأخیر آن‌ها بر عملکرد ناچیز است. برعکس، طول سیم‌های سراسری کمتر نمی‌شوند، سیم‌های سراسری واحد‌های عملیاتی مختلف یک سیستم را به یکدیگر متصل کرده و در سرتاسر تراشه منتشر می‌شوند. اکنون بخش اعظم تأخیر وابسته به سیم‌های سراسری می‌باشد. شکل (۲-۱۶) میزان تأخیر را برای سیم‌های محلی، سیم‌های سراسری و گیت‌ها نسبت به اندازه تکنولوژی ساخت را نمایش می‌دهد. در حالی‌که فرکانس کار و چگالی

ترانزیستور نیاز به ادامه افزایش دارند سیم‌های سراسری تأخیر انتشاری دارند که از یک پالس ساعت مورد نیاز بیشتر است.



شکل ۲-۱۶: نمایش برهم خوردن توازن تأخیر سیم‌ها و تأخیر گیت [۴]

به علاوه افزایش تعداد اجزاء متصل شده به سیم اشتراکی، ظرفیت خازنی در نتیجه تأخیر انتشار سیگنال را افزایش می‌دهد و فرکانس کار کاهش می‌یابد. با این محدودیت‌ها طراحی فیزیکی دقیق تبدیل به یک چالش در طراحی پردازنده‌ها شد. به علت اثرات کوچک تر شدن سیم فیزیکی فرق گذاشتن بین اتصالات داخلی محلی و سراسری کاملاً ضروری است و نیاز به پشتیبانی از معماری ارتباط سراسری در SOC نیازی بود که پدیدار گشت. در روش‌های طراحی NOC سیم‌های سراسری با سیم‌های قطعه‌ای خط لوله‌ای (پیوند‌های نقطه‌به‌نقطه) که گره‌های شبکه را به یکدیگر متصل می‌کنند جایگزین شده‌اند. هر واحد IP اغلب در SOC به یک گره در شبکه متصل می‌شود.

۲-۲-۲-۲ مصرف توان

مسئله ای که امروزه قطعات الکترونیک با آن روبرو هستند این است که به یک کارایی بسیار بالا همراه با مصرف توان کم دست یابند. مصرف توان کم برای رسیدن به یک استقلال قابل قبول برای سیستم های مجهز به باتری مانند دستگاه تلفن همراه و دیگر وسایل همراه شخصی مورد نیاز است. علاوه بر این مصرف توان کم اثرات زیست محیطی (مانند اتلاف گرما، نویز القایی خنک سازی) و هزینه بهره‌برداری از سیستم را کمتر می کند. به عبارت دیگر دستیابی به بالاترین راندمان توان اصلی ترین چالش در طراحی الکترونیک می باشد.

بنابراین طرح های مدارهای مجتمع بر خارج کردن گرما و مصرف توان می باشد و این هدف باعث شد در SOC تکنیک مدیریت توان پویا یا ¹DPM مطرح و استفاده شود [۱۵]. DPM یک مجموعه تکنیک هایی برای دستیابی به راندمان توان بالا با خاموش کردن یا کاهش میزان کارایی قسمت هایی از سیستم در زمانی که آن قسمت کار خاصی انجام نمی دهد، می باشد. در این روش ها نیاز به فاصله فیزیکی بین قسمت هایی که تغذیه شبکه را تأمین می کنند پدیدار می شود. در NOCها با دسته بندی واحدهای مختلف داخل تراشه سعی می شود که مصرف توان به طور مستقل به ایده آل نزدیک شود.

اما در سال های اخیر عامل دما مانع بسیار بزرگی برای ساخت تراشه هایی با کارایی بیشتر بوده است به طوری که شرکت اینتل مجبور شد از ساخت تراشه های ۴ گیگا هرتزی که تحقیقات درباره آن را شروع کرد چشم پوشی کند چون مصرف توان و افزایش دما آنقدر زیاد بود که با روش های معمول قابل خنک کردن نبود.

به دلایل مشکلات استفاده از باس، در [۵] استفاده از لایه هایی مانند لایه های OSI² که طبق استاندارد در شبکه های کامپیوتری مقیاس بزرگ استفاده می شود پیشنهاد شد.

¹ Dynamic Power Management

² Open System Interconnection

۲-۳ مبانی شبکه روی تراشه

در مقاله های مربوط به NOC روش های طراحی جنبه های گسترده ای از لایه ها را در برمی گیرند که با حرکت از یک سطح طراحی به سطح دیگر مشکلات متفاوتی را مشاهده می کنیم. به عنوان مثال آنچه در نرم افزار سیستم پدیدار می شود شامل مسائل فیزیکی و سخت افزار مانند انحراف پالس ساعت یا تأخیر سیم های بلند نمی شود، همچنین لایه فیزیکی درباره اجرای حجم بزرگی از پردازش های موازی نظری ندارد.

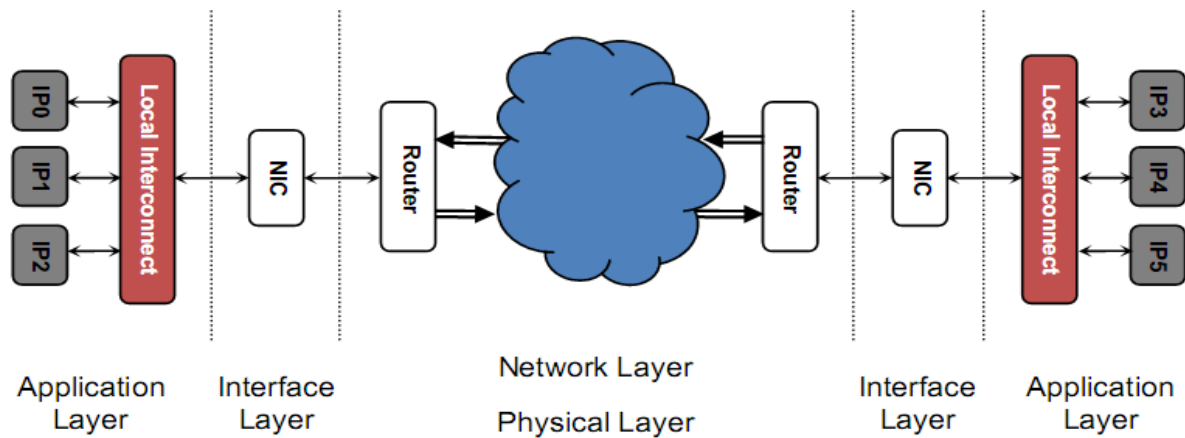
در نتیجه لایه های ارتباطی مدل OSI می تواند برای کاربردهای NOC تنظیم شود. هدف این است که مسائل مربوط به هر سطح نسبت سطوح دیگر مجزا شود تا ارتباط بین لایه ها مستقل از یکدیگر توسعه یابد. توجه داشته باشید که سطوح پایین تر در صورتی می توانند مفید باشند که باعث افزایش کارایی شوند. در یک سیستم مبتنی بر NOC لایه ها نسبت به یک شبکه کامپیوتری مقیاس بزرگ به یکدیگر بیشتر وابسته هستند و اغلب یک ویژگی فیزیکی مرتبط دارند. در [۴] بیان شده است که طراحی یک سیستم با استفاده از معماری شبکه روی تراشه را می توان در چهار لایه به شرح زیر بررسی کرد:

- لایه فیزیکی (لایه معادل در OSI : لایه فیزیکی): این لایه مربوط به جزئیات پایین ترین سطح انتقال داده ها روی سیم است. این لایه زمان بندی سیگنال و همه مسائل مربوط به آن را مانند حداکثر تأخیرهای سیم، میزان انحراف پالس ساعت و خطای همزمانی (که همه آن ها با کوچکتر شدن تکنولوژی ساخت بدتر می شوند) را تعریف می کند. مصرف توان یکی از مهم ترین مسائل در این سطح می باشد. پیاده سازی یک سیستم به عنوان یک مدار همزمان یا غیرهمزمان یک تصمیم طراحی است که به علت مواجه شدن با مسائل لایه فیزیکی باید گرفته شود. این لایه ها عبارتند از:
- لایه شبکه (لایه های معادل در OSI: لایه پیوند و شبکه): لایه شبکه ساختار کلی یا توپولوژی ارتباط را مشخص می کند. مهمترین کار این لایه این است که تعیین می کند داده ها از مبدأ به مقصد چگونه

مسیریابی شوند. این لایه می تواند با انتخاب الگوریتم مسیریابی، سوئیچینگ و کنترل جریان مناسب که همه آن ها اثر مهمی بر کارایی دارند، بهینه شود. بن بست، هرزگردی و اشباع برخی از موارد بحرانی هستند که در این لایه معرفی می شوند.

- لایه واسط (لایه معادل OSI: لایه انتقال): تجزیه یک پیام به چند بسته در مبدأ و اتصال مجدد آن ها مهم ترین عمل لایه واسط در یک NOC می باشد. مانند بسیاری از شبکه ها نحوه کنترل جریان که وابسته به اندازه بسته و نحوه بسته بندی می باشد تصمیم مهمی در این لایه به شمار می رود.
- لایه کاربرد (لایه های معادل در OSI: لایه های کاربرد، جلسه و نمایش): لایه کاربرد بالاترین سطح انتزاعی در معماری ارتباط است. این لایه توابع خاصی را برای بهره برداری از قابلیت های اجزاء سطح پائین تر فراهم می کند و سیستم می تواند آن عملکردهای ارتباط انتزاعی را بدون هیچ نگرانی نسبت به جزئیات شبکه مورد استفاده قرار دهد. تمام نرم افزار سیستم مربوط به لایه کاربردی طراحی شبکه می باشند. از دیگر وظایف این لایه برقراری یک ارتباط منطقی قابل اعتماد بین سوئیچ ها می باشد. برای این منظور اغلب کنترل جریان و خطایابی انجام می شود.

شکل (۲-۱۷) اجزاء لایه های طراحی شده یک شبکه روی تراشه را نشان می دهد. به طور نمونه در یک MP-SoC با حافظه اشتراکی هر زیر سیستم، که با یکدیگر یک سیستم کلی را تشکیل می دهند، شامل یک یا چند پردازنده، یک یا چند بانک حافظه فیزیکی، IPهای اختصاصی (از قبیل پردازنده های کمکی، کنترل کننده های ورودی/خروجی و...) و موارد دیگری باشد که ممکن است با اتصالات داخلی محلی با یکدیگر ارتباط داشته باشند. همه این اجزاء می توانند در لایه کاربردی سیستم مورد بررسی قرار گیرند.



شکل ۲-۱۷: نمایش لایه‌ها در شبکه روی تراشه [۴]

زیر سیستم‌ها با یک کنترل‌کننده واسط شبکه^۱ NIC (که تنها راه دسترسی است) به شبکه متصل می‌شوند. NIC درگیر مسائلی است در لایه واسط رخ می‌دهد. در نتیجه این بخش، حتی اگر از نظر فیزیکی تراشه به قسمت‌های مختلف خوشه بندی شده باشد، تمام پردازنده‌ها در تمام خوشه‌ها ممکن است فضای آدرس را به اشتراک بگذارند و همه پردازنده‌ها در سیستم می‌توانند هر مقصد یا IP جانبی را آدرس دهی کنند.

یک شبکه از تعدادی سوئیچ تشکیل شده است و مشکلات لایه شبکه عبارتند از طراحی سوئیچ و اتصالات آن‌ها. دقیقاً در سطح شبکه مفهوم سوئیچ و تعیین توپولوژی هندسی از سیستم تعریف می‌شود. واحدی که مشخص می‌کند که کدام سوئیچ باید فعال شود اغلب مسیریاب نامیده می‌شود و باید با توجه به مسائل فیزیکی پیاده‌سازی شود. بنابراین یک مسیریاب ممکن است به هر دو لایه فیزیکی و لایه شبکه تعلق داشته باشد. تمایز قائل شدن بین این دو لایه به عنوان دو لایه مجزا شده ساده و روشن نیست.

تمرکز اصلی در این بخش بر ترکیب و آمیختن ایده‌های پیاده‌سازی معماری شبکه می‌باشد. طبیعتاً تعریف اصول معماری مبتنی بر NOC نهفته در روش طراحی مسیریاب است، یعنی طراحی لایه‌ی شبکه و لایه‌فیزیکی

^۱ Network Interface Controller

که معماری شبکه وابسته به آن است. گاهی، در صورت نیاز به تشکیل یک رابط بین معماری شبکه و اجزای لایه کاربرد، لایه واسط نیز می‌تواند با اهمیت باشد.

فصل سوم

ویژگی های شبکه روی تراشه

چهار ویژگی اصلی که به وسیله آن‌ها روش‌های مختلف NOC را دسته‌بندی کرد عبارتند از: توپولوژی، الگوریتم مسیریابی، روش ارسال بسته^۱ و جریان داده^۲.

۳-۱ توپولوژی

توپولوژی شبکه مطالعه درباره تنظیم و اتصال گره‌هاست. نحوه تنظیم گره‌ها شکل هندسی شبکه را مشخص می‌کند و چگونگی اتصال آن‌ها قسمت فیزیکی توپولوژی شبکه را مشخص می‌کند.

انتخاب توپولوژی شبکه بر اساس سه جنبه که دوجه دو با یکدیگر در تعارض هستند انجام می‌شود: کارایی، هزینه و مقیاس‌پذیری [۴]. توپولوژی‌هایی که اغلب برای اتصالات داخلی شبکه به کار می‌روند عبارتند از: باس اشتراکی [۱۶]، درخت باینری [۱۷]، درخت-فره [۱۷]، درخت-فره پروانه‌ای [۱۷]، حلقه‌ای [۱۸]، حلقه دوجه‌ای (طاقچه‌ای) [۱۹]، آرایه‌ای، آرایه مربعی (مش) [۴]. براساس نیازمندی‌های سیستم‌های برپایه NOC و تئوری گراف، تعداد کل سوئیچ‌ها، درجه هر سوئیچ، قطر شبکه، تعداد کل کانال‌ها و تعداد کانال‌های دوجه‌ای مهمترین ویژگی‌های برای مقایسه بین توپولوژی‌های مختلف می‌باشد.

- **تعداد سوئیچ‌ها:** تعداد کل سوئیچ‌ها مورد نیاز است تا توپولوژی شبکه درک شود. تعداد کمتر سوئیچ‌ها به معنی تعداد عملیات کمتر و سرباری کمتر شبکه می‌باشد.
- **درجه سوئیچ:** تعداد پورت‌های ورودی/خروجی می‌باشد. درجه کمتر سوئیچ به معنی پیچیدگی کمتر طراحی می‌باشد. مدار ساده‌تر سطح سلیکون و توان کمتری مصرف می‌شود.
- **تعداد کانال‌ها:** حداکثر ارتباط همزمان در شبکه را تعیین می‌کند. تعداد کانال‌های بیشتر به معنی حجم عملیات فشرده‌تر می‌باشد که باعث می‌شود شبکه زودتر اشباع شود.

^۱Switching

^۲ Data Flow

- **تعداد کانال‌های دو بخشی:** کمترین تعداد کانال‌هایی که اگر برداشته شوند، شبکه به دو قسمت تقریباً مساوی تقسیم می‌شود. کانال‌های دو بخشی یک اندازه‌گیری تنگنای کانال‌ها هستند که می‌تواند در ارتباط اعضاء یک زیرشبکه با یک زیرشبکه دیگر مورد استفاده قرار گیرد. تعداد بیشترکانال‌های دو بخشی به معنی امکان بیشتر ارتباط سراسری همزمان و دیرتر اشباع شدن شبکه می‌باشد.

در جدول ۱-۲ نتایج تحلیل برخی از توپولوژی‌ها ارائه شده است.

جدول ۱-۳: ویژگی‌های چند توپولوژی در طراحی شبکه روی تراشه [۴]

	Number of Nodes	Number of Switches	Switch Degree	Network Diameter	Number of Channels	Number of Bisection Channels
Shared-Bus	N	1	N	1	1	1
Crossbar	N	1	N	1	N	N
Binary Tree	$N = 2^n$	$N - 1$	3	$2 \log_2 N$	$4N - 4$	2
Fat-Tree (2-ary)	$N = 2^n$	$\frac{1}{2} N \times \log_2 N$	4	$2 \log_2 N$	$2N \times \log_2 N$	N
Fat-Tree (4-ary)	$N = 4^n$	$\frac{1}{4} N \times \log_4 N$	8	$2 \log_4 N$	$2N \times \log_4 N$	N
Butterfly Fat-Tree	$N = 2^n$	$\frac{1}{2} N \times \log_2 N$	2	$\log_2 N$	$N \times \log_2 N$	$\frac{1}{2} N$
Ring	N	N	3	$\frac{1}{2} N$	4N	4
2D-Ring	$N = n^2$	N	5	\sqrt{N}	6N	$4\sqrt{N}$
Chordal Ring	$N = 2n$	N	4	$\frac{1}{4} N$	5N	8
Array	N	N	3	N	$4N - 2$	2
2D-Array	$N = n^2$	N	5	$2\sqrt{N}$	$6N - 4\sqrt{N}$	$2\sqrt{N}$

در یک تقسیم بندی توپولوژی‌ها را می‌توان به دو دسته مستقیم و غیرمستقیم تقسیم کرد. شبکه‌های غیرمستقیم، شبکه‌هایی هستند که در آن‌ها زیر مجموعه‌ای از گره‌ها وجود داشته باشد که هیچ IP به آن‌ها متصل نشده باشد و شبکه‌های مستقیم آن‌هایی هستند که حداقل یک IP به یک سوئیچ گره متصل شده است

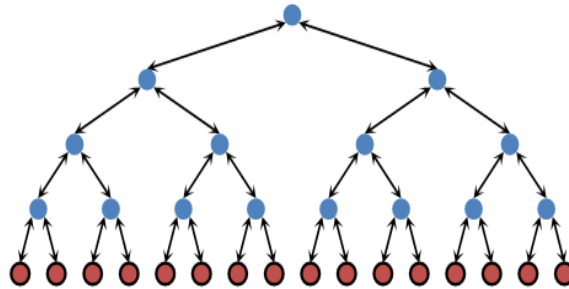
بنابراین سوئیچ‌ها ممکن است به صورت منظم بین IPها پخش شده باشند. استفاده از توپولوژی‌های مستقیم کمک می‌کند که پیاده‌سازی فیزیکی ساده‌تر انجام شود.

معروفترین توپولوژی غیرمستقیم توپولوژی‌های برپایه درخت می‌باشد که در ادامه به بررسی آن‌ها می‌پردازیم.

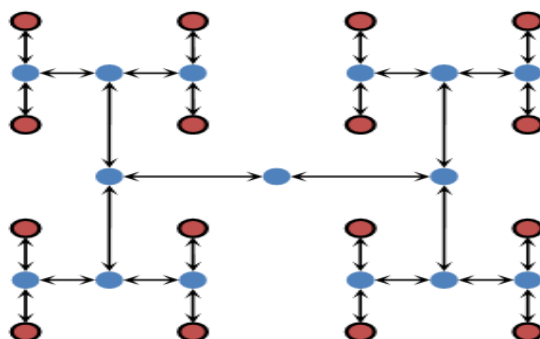
۳-۱-۱ توپولوژی‌های برپایه درخت:

توپولوژی‌های برپایه درخت را می‌توان به سه کلی تقسیم کرد که عبارتند از درخت باینری، درخت-فره باینری و درخت پروانه‌ای.

درخت باینری یک نمونه از توپولوژی شبکه می‌باشد که در آن بین دو گره تنها یک مسیر وجود دارد و در شکل (۳-۱) نمایش داده شده است. این توپولوژی $N-1$ سوئیچ از درجه ۳ به کار می‌برد (N تعداد گره‌ها می‌باشد) تعداد کل کانال‌ها $4 \times (N-1)$ و قطر شبکه $2 \times \log_2^N$ می‌باشد. در حالی که پارامترهای این توپولوژی نشان دهنده یک کارایی رضایت‌بخش دارد، تعداد محدود کانال‌های دوبخشی در آن مشخص می‌کند که تعداد ارتباط‌های همزمان سراسری محدودی دارد. شکل (۳-۱) بسط یافته این توپولوژی را نمایش می‌دهد.



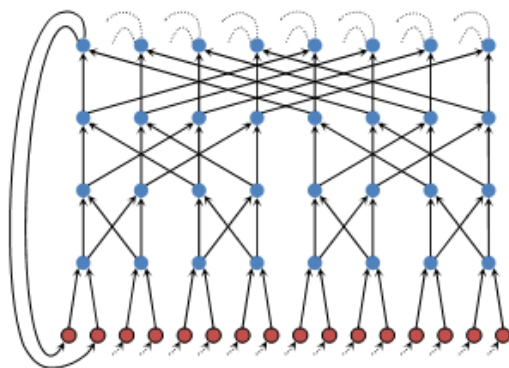
شکل ۳-۱: درخت باینری [۴]



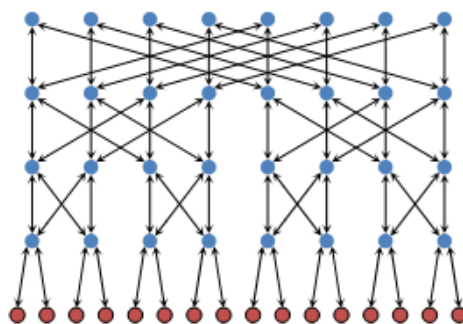
شکل ۳-۲: آرایش بسط یافته یک درخت باینری [۴]

درخت-فربه باینری و درخت-فربه پروانه ای دو راه حل پیشنهادی برای افزایش تعداد کانال‌های دوبخشی می باشد. همانطور که جدول (۱-۲) نشان می‌دهد تعداد کانال‌ها و کانال‌های دوبخشی در توپولوژی فربه-درخت نسبت به پروانه ای دو برابر است و هزینه انجام این کار نیز دوبرابر شدن قطر شبکه و درجه سوئیچ‌ها می باشد. پیوندها در توپولوژی درخت-فربه دوجهته هستند در حالی که در توپولوژی پروانه‌ای یک جهته می باشند. همانطور که در شکل (۳-۳) مشاهده می شود در یک شبکه پروانه ای برای هر جفت مبدا و مقصد فقط یک مسیر وجود دارد و باعث می شود الگوریتم مسیریابی ساده باشد. از طرف دیگر بهره از تنوع مسیر در توپولوژی درخت-فربه که در شکل (۴-۳) نشان داده شده است یک کارایی شگفت‌انگیز را پیش بینی می کند، نیازمند یک الگوریتم مسیریابی افقی است که باعث افزایش پیچیدگی پیاده سازی می شود.

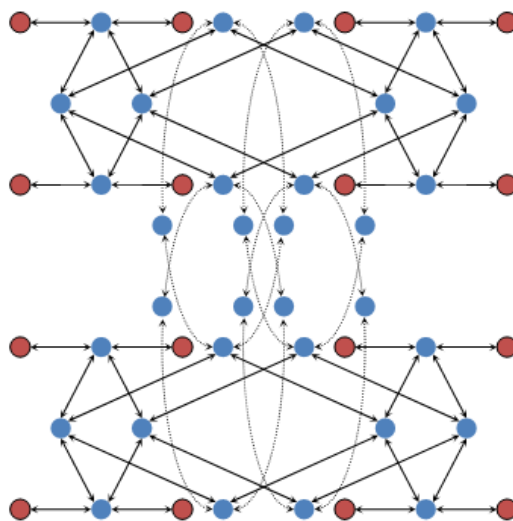
برخلاف درخت باینری که ممکن است که یک آرایش دوبعدی منظم داشته باشد توپولوژی درخت-فربه باینری (و همچنین درخت باینری پروانه ای) را به سختی می توان به صورت منظم در یک سطح دوبعدی تراشه پخش کرد و در نتیجه برای پیاده سازی تعداد سیم‌های با طول بلند بسیار افزایش می یابد. یک توپولوژی درخت-فربه می تواند به عنوان یک گراف چند بعدی منظور شود. شکل (۵-۳) یک آرایش ۴ بعدی از توپولوژی درخت-فربه را نمایش می دهد.



شکل ۳-۳: درخت - فرجه پروانه ای [۴]

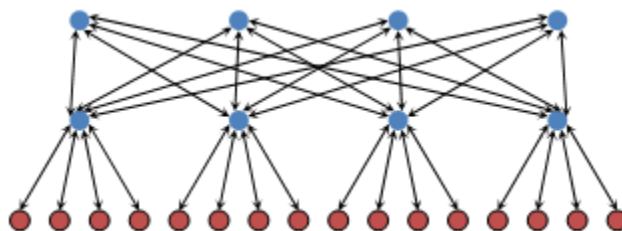


شکل ۴-۳: درخت - فرجه باینری (۲- آرایه ۴- بعدی) [۴]

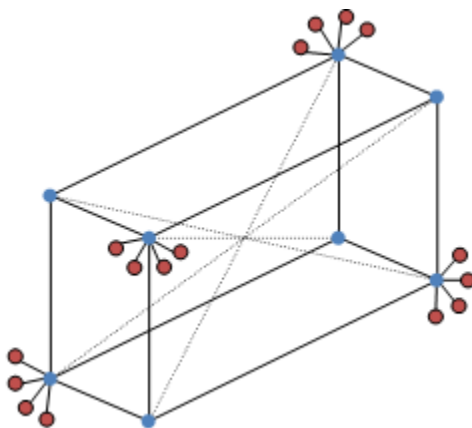


شکل ۵-۳: نمایش بسط یافته درخت فرجه - باینری [۴]

عبارت ۲-آرایه و n -بعد در توپولوژی درخت-فریه نشان دهنده ساختار توپولوژی می‌باشد. در اصطلاح درخت، عبارت k -آرایه، k تعداد برگ‌های متصل شده به یک گره را نشان می‌دهد و عبارت k -آرایه n -بعد در توپولوژی درخت-فریه نیز به یک توپولوژی مشابه اشاره دارد. شبکه SPIN [۲۰] یک شاهکار ۴-آرایه n -بعدی درخت-فریه می‌باشد که اثبات شده است نسبت به هر نوع دیگر، سخت‌افزار کارآمدتری دارد. به عنوان نمونه یک توپولوژی ۴-آرایه ۳-بعدی در شکل (۳-۶) [۴] نمایش داده شده است. این گراف ۳-بعدی مانند شکل هندسی یک مکعب می‌باشد. شکل (۳-۷) [۴] مشاهده شود.



شکل ۳-۶: شبکه SPIN ۴-آرایه ۳-بعدی [۴]



شکل ۳-۷: نمایش دیگر شبکه SPIN ۴-آرایه ۳-بعدی [۴]

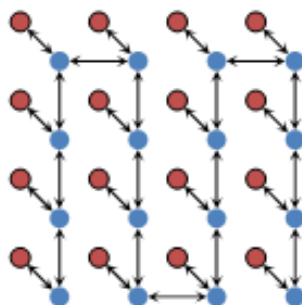
به علت قطر کم شبکه که ضریبی از \log_k^N می‌باشد، بیشترین تعداد کانال با ضریب $N \times \log_k^N$ و تعداد زیاد کانال‌های دوبرخی به تعداد N ، توپولوژی k -آرایه n -بعدی درخت-فریه از نظر اتصال و گستره کاربرد در معماری کامپیوترهای موازی بهترین انتخاب است. اما پیاده‌سازی این نوع از شبکه متصل شده به سیم کشی

بسیار پیچیده نیاز دارد تا تعداد بسیار زیادی سیم را در لایه های مختلف استفاده کند. سیم کشی یک عامل محدود کننده و یک نگرانی مهم در طراحی NOC می باشد.

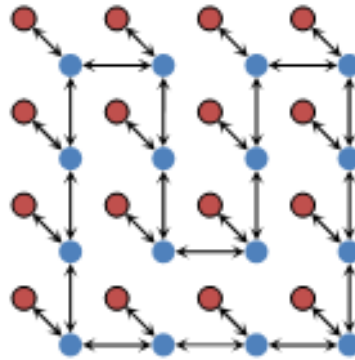
همچنین توپولوژی k -آرایه n -بعدی اجازه می دهد که تعداد سوئیچ ها از تعداد گره ها بیشتر باشد، وقتی تعداد گره ها بیشتر از k^k باشد تعداد سوئیچ ها برابر $(N \times \log_k N)/k$ می شود. افزایش k تعداد سوئیچ ها را کاهش می دهد و افزایش درجه سوئیچ که اثری مستقیم بر طراحی و پیچیدگی سیم کشی دارد. این رویداد در سربار شدن شبکه (به معنی مقدار پردازش مورد نیاز برای اتمام یک کار) مهم است. برای ارتباط داخلی تراشه سرباری شبکه کاملاً بحرانی تر از شبکه های خارج کامپیوتر هستند و مقیاس پذیری طرح نیز ضروری است.

۳-۱-۲ توپولوژی‌های مستقیم

دو نوع ساده از توپولوژی‌های ساده را که می‌توان نام برد عبارتند از آرایه‌ای و حلقه‌ای. در این توپولوژی میزان سرباری شبکه با درجه N افزایش می یابد که در شکل زیر نمایش داده شده است. هر دو توپولوژی به راحتی می توانند بر روی یک صفحه دوبعدی پیاده‌سازی شوند. اما تعداد کم کانال های دوبخشی و قطر بزرگ شبکه دو عامل محدود کننده برای مقیاس‌پذیری شبکه می باشند. شکل (۳-۸) یک توپولوژی آرایه‌ای و شکل (۳-۹) یک توپولوژی حلقه ای را نشان می دهند.



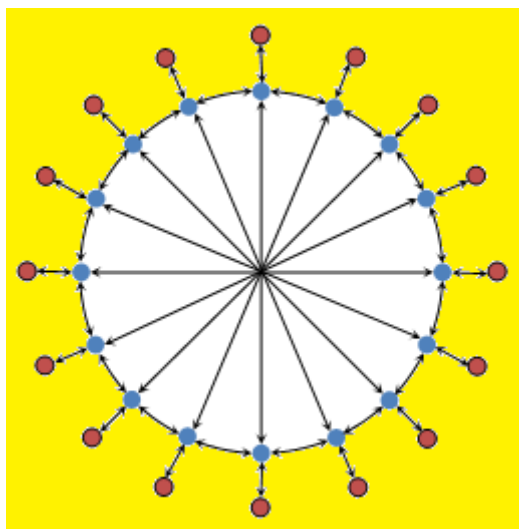
شکل ۳-۸: نمایش یک توپولوژی آرایه ای [۴]



شکل ۳-۹: نمایش یک توپولوژی حلقه ای [۴]

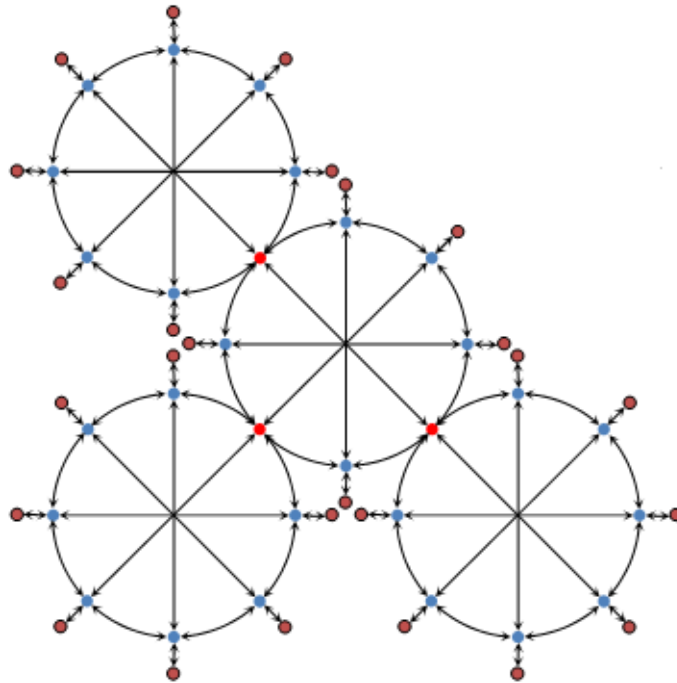
۳-۱-۲-۱ توپولوژی حلقه وتری

حلقه وتری یک توپولوژی حلقه‌ای توسعه یافته است که در آن هر گره در حلقه، علاوه بر دو گره مجاور در حلقه، توسط قطر حلقه به گره مخالف متصل شده‌اند. شکل (۳-۱۰) یک نمونه از این توپولوژی را با ۱۶ گره نمایش می‌دهد. توپولوژی حلقه‌ای وتری مسیر بین گره‌ها را نصف می‌کند که برای تعداد محدودی IP در مقایسه با توپولوژی‌های دیگر کوتاهترین مسیر می‌باشد. به عنوان مثال یک حلقه وتری با ۸ گره تضمین می‌کند که بین ۲ گره حداکثر ۴ گره بین قرار می‌گیرد.



شکل ۳-۱۰: نمایش یک توپولوژی حلقه وتری با ۱۶ گره [۴]

به هر حال توپولوژی حلقه وتری نیز مقیاس‌پذیر نیست. تعداد کانال‌های دوبخشی ۸ عدد می‌باشد و قطر شبکه با ضریب N افزایش می‌یابد. برای دستیابی به یک توپولوژی سلسله‌مراتبی با ۸ گره که Octagon نامیده می‌شود در [۲۱] معرفی شده است. شبکه شکل (۳-۱۱) پیشنهاد شده است. این طراحی بر اساس استفاده از پل‌های ویژه در برخی از گره‌ها (گره‌های قرمز رنگ در شکل شماره ۳-۱۱) می‌باشد. به خاطر بیاورید که روش سلسله‌مراتبی یکی از انواع معماری‌های برپایه گذرگاه (مانند گذرگاه‌های محلی) می‌باشد که برای ارتباط محلی سودمند بود. امکان ارتباط‌های سراسری همزمان کاهش می‌یابد.



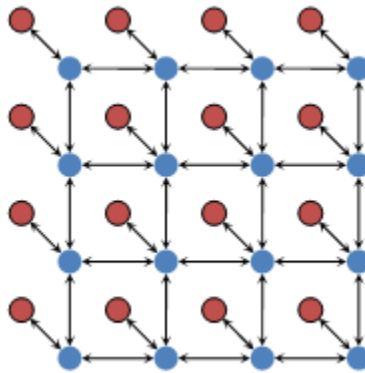
شکل ۳-۱۱: توپولوژی حلقه وتری سلسله‌مراتبی [۴]

۳-۱-۲-۲-۲ تولوژی‌های مش

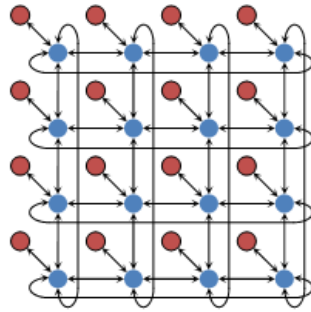
شبکه‌های با توپولوژی منظم که اتصال قابل قبول و استفاده عملی از منابع را پیشنهاد می‌کنند نخستین بار در [۳۲] معرفی شد، عبارت k -آرایه و n -مربع معرفی شد تا ایده‌های ساده‌ای از توپولوژی‌های چند بعدی مربعی ارائه

فصل سوم ویژگی‌های شبکه‌روی تراشه

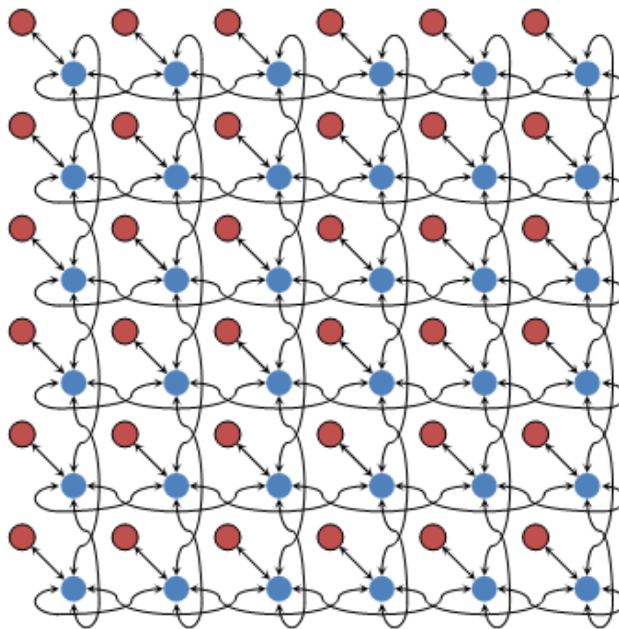
دهد. واقعیت این است که به علت ساختار مربعی و دوبعدی بودن و همچنین کاهش پیچیدگی سیم‌کشی این توپولوژی می‌تواند به صورت سراسر تراشه پیاده‌سازی شود. به علاوه مسیریابی در این روش آسان است. از توپولوژی‌های مربعی می‌توان به توپولوژی M_{SH} و طاقچه‌ای اشاره کرد. در توپولوژی طاقچه‌ای گره سیم‌هایی که ابتدا و انتها را در هر سطر و ستون را به یکدیگر متصل می‌کنند طول زیادی دارند ولی طول بقیه سیم‌ها در توپولوژی طاقچه‌ای و همه سیم‌ها در توپولوژی M_{SH} به فاصله بین دو IP مجاور محدود می‌شوند. برای متعادل کردن طول سیم‌ها در یک توپولوژی طاقچه‌ای می‌توان ساختار آن را آن را به صورت شکل (۳-۱۴) پیچاند. میانگین طول سیم‌ها در یک توپولوژی M_{SH} در مقایسه با توپولوژی‌های طاقچه‌ای یا طاقچه‌ای به هم تابیده کمتر است. از طرف دیگر تعداد کانال‌های دو بخشی توپولوژی طاقچه‌ای نسبت به مربعی دو برابر و قطر آن نصف توپولوژی مربعی است.



شکل ۳-۱۲: نمایش توپولوژی M_{SH} [۴]



شکل ۳-۱۳: نمایش توپولوژی طاقچه ای [۴]



شکل ۳-۱۴: پیچاندن توپولوژی طاقچه‌ای [۴]

انتخاب بین توپولوژی‌های طاقچه ای و میس قابل بحث است. هر دو شبکه کاملاً مقیاس‌پذیر هستند، اما در حالی که توپولوژی طاقچه ای کارایی بهتری دارد، در نظم و استفاده بهتر از لینک‌ها و در نتیجه کاهش سرباری شبکه اولویت با میس است. به عنوان مثال، سیم‌های بلند در توپولوژی طاقچه ای باعث شده است که توپولوژی میس محبوبیت بیشتری داشته باشد با این استدلال که تأخیر بین گره‌ها زیاد است و بر اساس آن مصرف توان در آن بیشتر است. همچنین در [۲۱] بیان شده است که میس از نظر اقتصادی مزیت دارد چون درجه سوئیچ می‌تواند در مرزها به ۴ و در گوشه‌ها به ۳ کاهش یابد. با توجه به نیازهای فعلی و آینده و روند تکنولوژی‌های

جدید میکرونی به نظر می‌رسد که انتخاب یک توپولوژی همه منظوره برای شبکه روی تراشه توپولوژی می‌ش مناسب تر است چون منظم تر است، طول سیم‌ها در آن کوتاه تر است و راندمان پیوندهای اشتراکی در آن بیشتر است.

۲-۳ مسیریابی

با توجه به توپولوژی الگوریتم‌های مسیریابی متفاوتی ارائه شده است. هر الگوریتم در حداقل کردن بعضی موارد تضمین‌های مختلفی ارائه دهد.

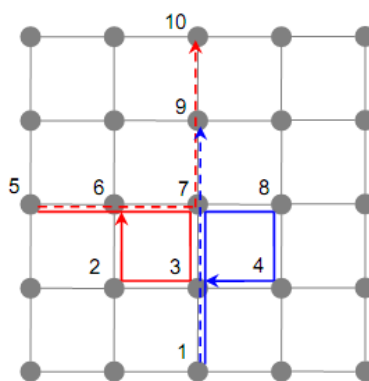
الگوریتم مسیریابی می‌تواند مسیریابی را در مبدأ یا به صورت توزیع شده در طول مسیر انجام شود. مسیریابی در مبدأ به این صورت می‌باشد که مسیری که بسته باید طی کند در همان گره مبدأ با توجه به توپولوژی شبکه تعیین می‌شود ولی در مسیریابی توزیع شده اطلاعاتی در بسته قرار داده می‌شود و در هر گره با توجه به موقعیت گره فعلی و اطلاعات موجود در بسته که آدرس مقصد را نیز شامل می‌شود مسیر مناسب مشخص می‌شود [۴]. مسیریابی توزیع شده به دو دسته قطعی و وفقی تقسیم می‌شود. در مسیریابی قطعی مسیر تنها با توجه به موقعیت گره‌های مبدأ و مقصد مشخص می‌شود و اغلب بین دو گره، تنها یک مسیر وجود دارد. در مسیریابی وفقی سعی می‌شود که مسیر با توجه به شرایط ترافیک در شبکه تعیین شود.

مسیریابی بر اساس الگوریتم قطعی برای شبکه‌های با توپولوژی منظم و یا شبکه‌هایی که الگوی ترافیک در آن‌ها منظم است مناسب می‌باشد در حالی که الگوریتم‌های وفقی برای شبکه‌های غیرمنظم و یا شبکه‌هایی که گره‌ها و پیوندها زیاد قابل اعتماد نیستند مورد استفاده قرار می‌گیرند. به علت سازوکار نظارتی پویایی که در الگوریتم‌های وفقی وجود دارد، این روش مسیریابی می‌تواند بسته‌هایی را که در گره‌ها یا کانال‌ها دچار بن-بست شده‌اند را نیز مسیریابی کند. این توانایی برای افزایش کارایی با اهمیت است ولی ممکن است باعث پیچیدگی بیشتر طراحی و کاهش سرعت شبکه شود.

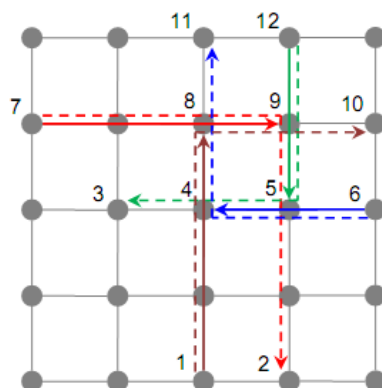
به هر حال مسیریاب باید بگونه ای باشد که از بروز حالت‌هایی که باعث از بین رفتن بسته‌ها می‌شود جلوگیری کند. این حالت‌ها را می‌توان به سه دسته تقسیم کرد که عبارتند از بن‌بست، هرزگردی و قحطی.

هرزگردی: این وضعیت وقتی رخ می‌دهد که یک بسته در یک مسیرحلقه ای گرفتار شده و بدون اینکه به سمت مقصد پیشروی کند در یک حلقه حرکت می‌کند بدون اینکه از آن خارج شود. این وضعیت در شکل (۳-۱۵) نمایش داده شده است.

بن‌بست: در این وضعیت دو بسته منتظر یکدیگر هستند تا دیگری مسیریابی و ارسال شود. شکل (۳-۱۶) بن‌بست را نشان می‌دهد، همانطور که مشاهده می‌شود بسته قرمز قصد دارد که از گره ۷ به گره ۲ برود (مسیری که با خط چین قرمز مشخص شده است) و در گره ۹ متوقف می‌شود (خط قرمز) چون توسط بسته سبز رنگ اشغال شده است، بسته سبز نیاز دارد که به از طریق گره ۹ به گره ۳ برود (خط سبز رنگ را ببینید) بسته سبز خودش در گره ۵ متوقف شده است و منتظر است کانال بین گره‌های ۴ و ۵ آزاد شود، گره ۵ به بسته آبی اختصاص داده شده است و بسته آبی در گره ۴ متوقف شده است چون گره ۴ توسط بسته قهوه‌ای که قصد دارد از طریق گره‌های ۸ و ۹ به گره ۱۰ برسد اشغال شده است (خط چین قهوه‌ای). بسته قهوه‌ای خود به علت اینکه گره ۸ قبلاً به بسته قرمز اختصاص یافته است نمی‌تواند پیشروی کند [۴].



شکل ۳-۱۵: نمایش هرزگردی در شبکه [۴]



شکل ۳-۱۶: نمایش بن‌بست در شبکه [۴]

قحطی^۱: وقتی بسته‌های با اولویت پایین تر هرگز به مقصد نرسند وضعیت قحطی رخ می‌دهد و علت آن این است که بسته‌های با اولویت بالاتر مسیرها را در همه زمان‌ها برای خود نگه می‌دارند. با اختصاص کانال‌هایی برای بسته‌هایی با اولویت پایین تر یا استفاده از یک مسیریاب نسبتاً خوب از این وضعیت جلوگیری کرد.

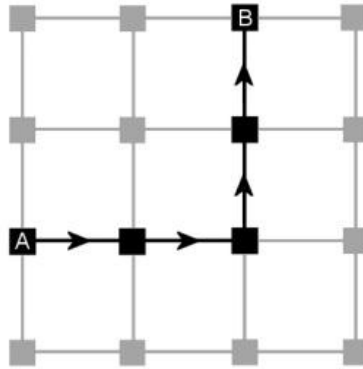
۳-۲-۱ الگوریتم‌های مسیریابی

الگوریتم‌های مسیریابی را به طور کلی می‌توان به دو دسته کلی مستقیم و چرخشی تقسیم کرد.

۳-۲-۱-۱ الگوریتم‌های مسیریابی مستقیم

مسیریابی XY: مسیریابی XY یک مسیریابی مستقیم می‌باشد که مسیریاب‌ها را ابتدا در جهت محور X (محور افقی) مشخص کرده، سپس جهت Y را مشخص می‌کند. مسیریابی XY در شبکه‌هایی که توپولوژی mesh یا torus دارند، استفاده می‌شود. این مسیریابی هرگز گرفتار بن‌بست یا هرزگردی نمی‌شود. شکل (۳-۱۷) این مسیریابی را نشان می‌دهد [۲۲].

¹ starvation



شکل ۳-۱۷: نمایش الگوریتم مسیریابی XY [۲۲]

در مسیریابی‌های قدیمی XY ترافیک به صورت منظم در تمام شبکه گسترش نمی‌یابد زیرا الگوریتم باعث می‌شود که بیشترین بار در میانه شبکه باشد. بنابراین نیاز به الگوریتم‌هایی است که بار ترافیک شبکه را یکنواخت کند.

مسیریابی شبه افقی XY: مسیریابی شبه افقی XY در روش‌های قطعی یا افقی که وابسته به حالت شبکه هستند به کار می‌رود. وقتی شبکه دچار انسداد می‌شود الگوریتم به روش افقی سوئیچ کرده و جستجوی مسیری که گرفتگی نداشته باشد را آغاز می‌کند.

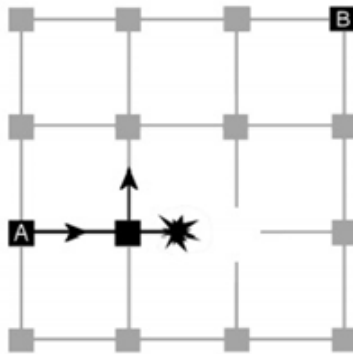
مسیریابی شبه افقی XY در شبکه‌های مش به کار می‌رود. هر مسیریاب پنج پورت دوجبهته دارند که عبارتند از north, south, east, west و local. پورت local مسیریاب را به IP آن مسیریاب متصل می‌کند در حالی که بقیه پورت‌ها به سوئیچ‌های همسایه متصل می‌شوند. هر پورت یک حافظه موقت کوچک و دو بیت معرف وضعیت دارد. بیت وضعیت به دیگر مسیریاب‌ها اعلام می‌کند که این مسیریاب در حال حاضر نمی‌تواند بسته جدیدی را قبول کند [۲۳].

وقتی چند بسته به طور همزمان وارد می‌شوند، مسیریاب اولویت بسته‌ها را تعیین می‌کند. بسته‌های وارد شده از پورت north دارای بالاترین اولویت هستند، پورت‌های south, east و west در اولویت بعدی قرار دارند.

در حالی که مسیریاب‌های XY قدیمی باعث می‌شوند که بار شبکه بیشتر در میانه شبکه باشد الگوریتم شبه تقریبی ترافیک شبکه را در کل شبکه تقسیم می‌کند.

مسیریابی XY احاطه کننده: مسیریابی XY احاطه کننده یا $S-XY^1$ سه روش متفاوت دارد که عبارتند از: روش XY نرمال $N-XY^2$ که دقیقاً مشابه مسیریابی XY معمولی می‌باشد، ابتدا مسیر بسته در راستای محور X سپس در جهت محور Y مشخص می‌شود. روش احاطه کننده افقی $SH-XY^3$ وقتی به کار می‌رود که مسیریاب‌هایی که در همسایگی سمت چپ یا راست قرار گرفته باشند غیرفعال باشند. به طور مشابه احاطه کننده عمودی $SV-XY^4$ وقتی به کار می‌رود که مسیریاب‌های که در همسایگی بالاتر یا پایین تر قرار گرفته اند غیرفعال باشند [۲۴].

روش SH-XY ستون مربوط به مسیر بسته‌ها را بر اساس هماهنگی با مقصد مشخص می‌کند. این الگوریتم در کوتاه‌ترین مسیر ممکن بسته‌ها را از مسیریاب‌های غیرفعال می‌گذراند مانند شکل‌های (۳-۱۸) و (۳-۱۹) [۲۲].



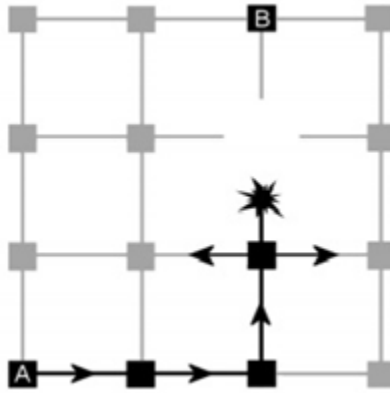
شکل ۳-۱۸: مسیریابی به روش احاطه کننده افقی [۲۲]

¹ XY Surround

² Normal XY

³ Surround horizontal XY

⁴ Surround vertical XY



شکل ۳-۱۹: مسیریابی به روش احاطه کننده عمودی [۲۲]

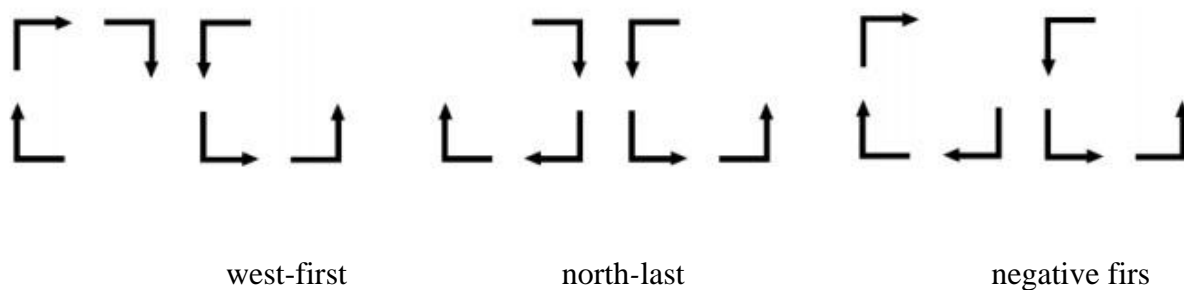
۳-۲-۱-۲ الگوریتم های مسیریابی چرخشی

الگوریتم های مدل چرخشی یک یا چند حلقه را مشخص می کنند. در این مدل ها هرزگردی رخ نمی دهد.

مسیریابی west-first: یک الگوریتم مسیریابی west-first از تمام چرخش ها به پورت west جلوگیری می کند. بسته ای که تصمیم دارد به شرق برود ابتدا تا حد امکان به سمت west منتقل می شود تا دیگر نیاز نباشد به سمت west چرخش پیدا کند.

مسیریابی North-last: در این روش چرخش به سمت پورت north امکان پذیر نیست. ابتدا بسته در جهت سایر پورت ها مسیریابی شده و در پایان به سمت پورت north چرخش می یابد.

مسیریابی negative-first: الگوریتم این مسیریابی به همه چرخش ها اجازه می دهد که انجام شوند و بجز چرخش هایی که از جهت مثبت به جهت منفی می باشند. چرخش های در جهت منفی قبل از هرکار دیگری باید انجام شوند [۲۲].



شکل ۳-۲۰: نمایش مسیریابی‌های چرخشی [۲۲]

۳-۳ روش ارسال بسته

الگوریتم‌های مسیریابی، مسیر بین مبدأ و مقصد را تعیین می‌کنند ولی روش‌های ارسال بسته چگونگی حرکت بسته در مسیر را تعریف می‌کنند. دو روش کلی در ارسال بسته استفاده می‌شود که عبارتند از روش بسته‌ای و روش مدار [۲۴].

در روش مدار قبل از ارسال بسته مسیر بین مبدأ و مقصد برقرار می‌شود و تا زمانی که بسته به طور کامل ارسال شود، مسیر برقرار می‌ماند ولی در روش بسته‌ای هر بسته محتوی اطلاعات مسیری است که داده‌ها به آنجا منتقل خواهند شد و از روی آن در هر گره، گره مناسب بعدی برای ادامه حرکت بسته مشخص می‌شود.

چون در روش مدار به حجم زیادی از بافرها نیاز است، در معماری شبکه‌روی تراشه برای انتقال بسته‌ها بین سوئیچ‌ها از روش بسته‌ای استفاده می‌شود و سه روش اصلی آن عبارتند از [۲۵]:

(۱) ذخیره و ارسال^۱

(۲) خزیدنی^۲

(۳) برش - میان مجازی^۳

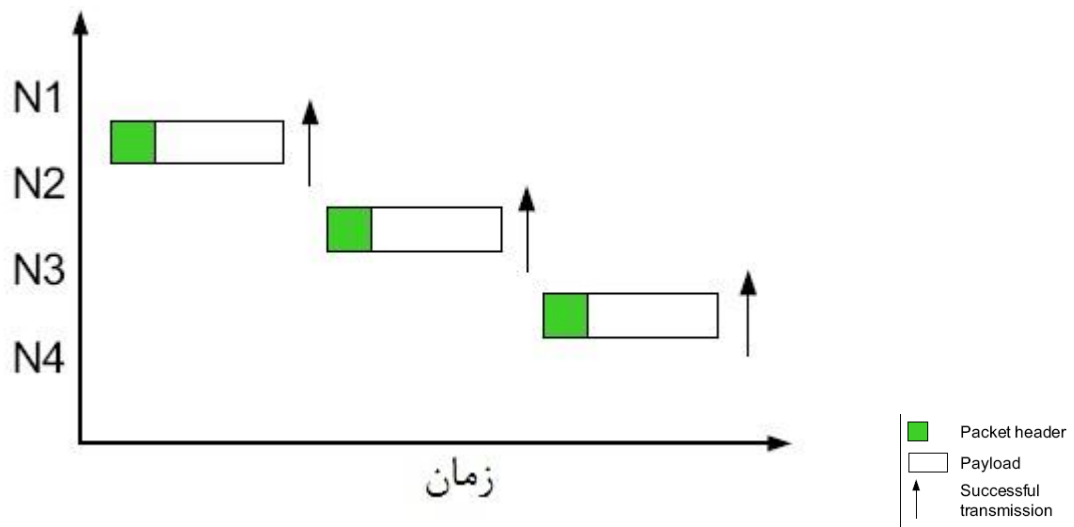
^۱ Store and forward

^۲ wormhole

^۳ virtual cut through

۳-۳-۱ ذخیره و ارسال

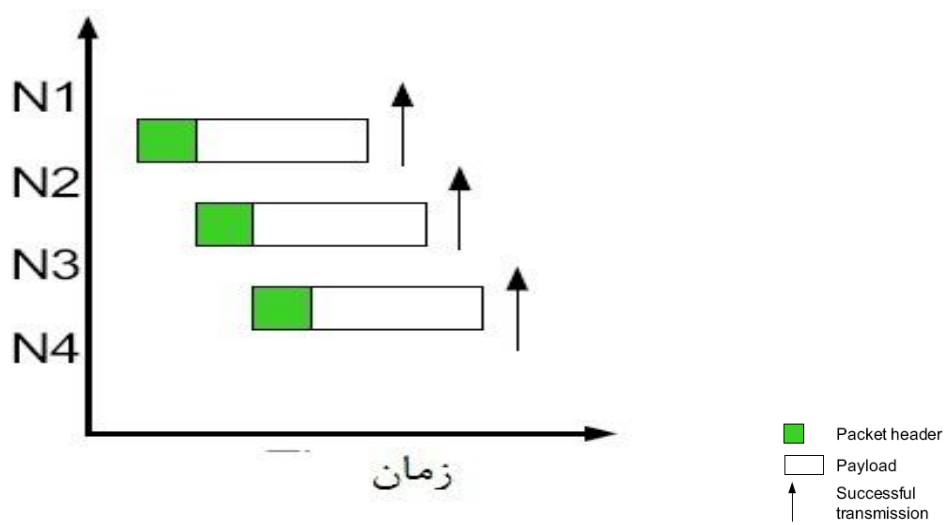
در روش ذخیره و ارسال یک بسته به طور کامل ذخیره شده سپس براساس اطلاعاتی که در سرآیند قرار دارد ارسال می‌شود. این روش به حافظه زیادی برای ذخیره یکباره بسته نیاز دارد بنابراین اگر در مسیر ارسال حافظه کافی نباشد در انتقال بسته مشکل ایجاد شده و بسته گیرمی افتد. شکل (۳-۲۱) این مسیریابی را نشان می‌دهد



شکل ۳-۲۱: ارسال بسته‌ها در روش ذخیره و ارسال [۲۶]

۳-۳-۲ برش میان مجازی

برش میان مجازی بهبودیافته مسیریابی ذخیره و ارسال می‌باشد. یک مسیریاب به محض اینکه مسیریاب بعدی اجازه داد بسته را می‌فرستد یعنی قبل از دریافت و ذخیره شدن تمام بسته نیز ارسال می‌توان صورت گیرد. تأخیر این روش از ذخیره و ارسال کمتر است. مانند شکل (۳-۲۲).



شکل ۳-۲۲: نمایش روش برش-میانی مجازی [۲۶]

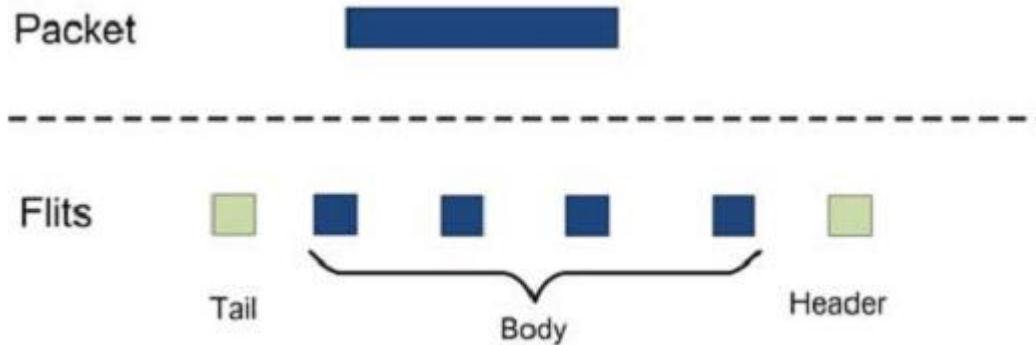
۳-۳-۳ خزیدنی

در روش خزیدنی بسته‌ها به واحدهای کوچکی با اندازه برابر که فلیت نامیده می‌شوند تقسیم می‌شوند. یک بسته بین مسیریاب‌ها به صورت فلیت به فلیت منتقل می‌شود. هر بسته به سه قسمت تجزیه می‌شود: سرآیند، بدنه و یک فلیت دیگر که ممکن است نشان دهنده اتمام یک بسته باشد که در اینصورت فلیت پی‌آیند نامیده می‌شود و آخرین فلیت بسته خواهد بود و یا فلیتی است که تعداد فلیت‌های یک بسته را در خود ذخیره دارد و فلیت دوم ارسالی خواهد بود. اشکال روش اول این است که بیت‌های اختصاص یافته به فلیت‌های سرآیند و پی‌آیند دیگر نمی‌توانند در فلیت بدنه نیز وجود داشته باشند و از این نظر دارای محدودیت می‌باشد ولی روش دوم این محدودیت را ندارد و توسط یک شمارنده مشخص می‌شود که فلیت متعلق به کدام دسته می‌باشد، به این ترتیب که اگر مقدار شمارنده ۱ بود فلیت وارد شده فلیت سرآیند می‌باشد و اگر ۲ بود، فلیت وارد شده تعداد فلیت‌های بسته را در خود ذخیره خواهد داشت و باید ذخیره شود و برای مقادیر بزرگ‌تر ۲ فلیت‌های وارد شده فلیت بدنه خواهند بود و باید به

فصل سوم ویژگی‌های بک‌روی تراشه

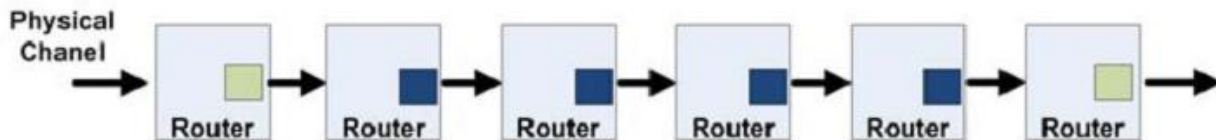
مسیری که توسط فلیت سرآیند مشخص شده است ارسال شوند. در این پایان نامه از روش دوم استفاده شده است.

فقط فلیت سرآیند حاوی اطلاعات مربوط به مسیریابی می باشد. بنابراین بقیه فلیت های باقی مانده که بسته را تشکیل می دهند از همان مسیر فلیت سرآیند عبور می کنند. شکل (۳-۲۳) نحوه شکسته شدن یک بسته به تعدادی فلیت را نشان میدهد.



شکل ۳-۲۳: نمایش روش خزیدنی

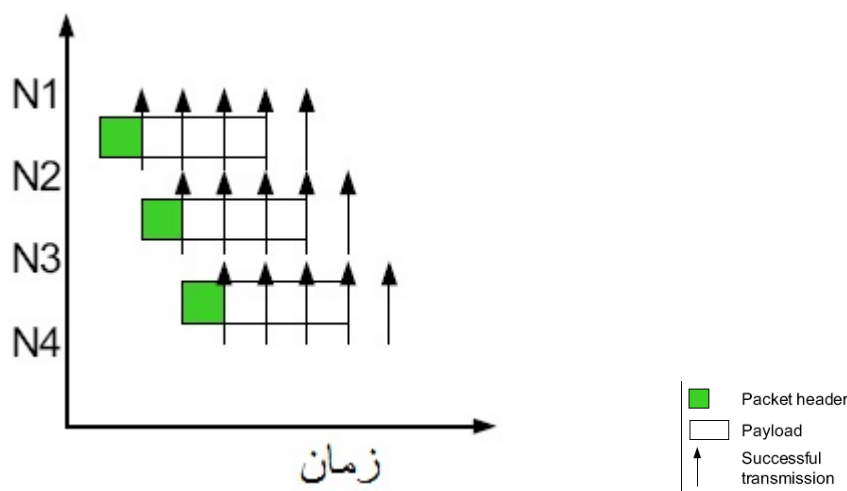
بعد از تجزیه بسته فلیت ها به صورتی که در شکل (۳-۲۴) مشاهده می شود به صورت موازی نیز می توانند انتقال پیدا کنند.



شکل ۳-۲۴: نمایش حرکت فلیت ها در روش خزیدنی

این روش نسبت به دوروش دیگر به حافظه کمتری نیاز دارد زیرا فقط یک فلیت نیاز به ذخیره شدن و ارسال دارد. همچنین تأخیر این مسیریابی کمتر است ولی احتمال وقوع بن بست در آن بیشتر است.

شکل (۳-۲۵) انتقال بسته‌ها با روش خزیدنی نسبت به زمان را نشان می‌دهد. همانطور که مشاهده می‌شود در این روش چند بسته به صورت همزمان می‌توانند ارسال شوند و نسبت به روش‌های دیگر سرعت بیشتری دارد و به همین علت در بیشتر طراحی‌های مربوط به شبکه روی تراشه از این روش استفاده می‌شود.



شکل ۳-۲۵: نمایش حرکت فلیت‌ها در روش خزیدنی نسبت به زمان [۲۶]

۴-۳ کنترل جریان شبکه

در کنترل جریان شبکه چگونگی ارسال بسته‌ها در شبکه، چگونگی به اشتراک گذاشتن بافرها و لینک‌های اتصال (کانال‌های فیزیکی) و جزئیات اتصال تعیین می‌شود. اساساً مکانیسم یک کنترل جریان شبکه تبادل داده با هماهنگی ارسال و دریافت بسته‌های اطلاعاتی برای تحویل صحیح بسته می‌باشد. وقتی که دو یا تعداد بیشتری بسته تلاش می‌کنند از گذرگاه یکسانی استفاده کنند. به علت محدودیت عرض باند بافرها و لینک‌ها بسته‌ها ممکن است به بن بست برسند. در شبکه‌های با روش ارسال بسته‌ای دو نوع کنترل جریان شبکه استفاده می‌شوند که عبارتند از: کنترل جریان بدون بافر و کنترل جریان با بافر [۲۷].

کنترل جریان داده بدون بافر ساده‌ترین نوع کنترل جریان می‌باشد. از آنجا که سوئیچ‌ها بافر ندارند منبع اختصاص داده شده لینک‌ها می‌باشند. این روش متکی بر یک ناظر برای تصمیم‌گیری در مواقعی که ارسال

بسته‌ها با مشکل مواجه می‌شود، می‌باشد. در غیر اینصورت بعد از پیشرفت یک بسته بر روی لینک چون بافری موجود نیست بسته‌های دیگر دچار آسیب دیدگی یا از دست رفتن مسیریابی شوند ولی مصرف توان در آن کمتر است. در [۲۸] کنترل جریان بدون بافر مورد ارزیابی قرار گرفته است.

در کنترل جریان با بافر بسته‌های به بن‌بست رسیده ذخیره می‌شوند. روش‌های ارسال داده که در بخش قبل مورد بررسی قرار گرفت جزء این دسته به شمار می‌روند. اختصاص دادن بافر و مدیریت آن نقش مهمی در بن‌بست شبکه دارد. در [۲۹] سه تکنیک به نام های STALL/GO, T-Error, ACK/NACK برای اختصاص بافر و پهنای باند در ارتباط خط لوله ای بین سوئیچ‌ها بررسی شده‌اند. این تکنیک‌ها در نتیجه توازن‌های مختلف بین میزان سطح اشغال شده و توان مصرفی نتایج متفاوت دارند.

فصل چهارم

شبیه سازی و تحلیل نتایج

شبکه‌ی Hermes یکی از شبکه‌های با معماری NOC می‌باشد که در [۳۴] معرفی شده است که در این پایان نامه مورد توجه قرار گرفته است. این شبکه دارای توپولوژی می‌ش می‌باشد و از الگوریتم مسیریابی XY برای مسیریابی استفاده می‌کند. یکی از اشکالات شبکه Hermes این است که با افزایش تعداد IPها، کارایی آن کاهش می‌یابد. از طرف دیگر نکته‌ای که در معماری NOC در نظر گرفته نمی‌شود این است که ممکن است بعضی از IPها در شبکه ارتباط و تبادل داده‌های زیادی با یکدیگر داشته باشند، بنابراین اگر آن‌ها با یکدیگر یک شبکه محلی تشکیل دهند راحت‌تر و سریع‌تر می‌توانند داده‌ها را مبادله کنند. به همین دلیل در شبکه طراحی شده به جای یک IP تنها، در هر سوئیچ چهار IP قرار گرفته اند که با یک باس اشتراکی بین آن‌ها ارتباط برقرار شده است و یک شبکه محلی تشکیل داده اند بنابراین ترکیبی از NOC و SOC را خواهیم داشت. عبارت ترکیبی در عنوان پایان نامه به همین واقعیت اشاره دارد.

از دیگر معایب شبکه Hermes این است که فرکانس انتقال فلیت‌ها در آن نصف فرکانس کار شبکه می‌باشد و همچنین زمان زیادی برای مسیریابی مصرف می‌کند. که باعث هدر رفتن بخشی از توان ورودی و کند شدن انتقال بسته‌ها می‌شوند، به علاوه صرف زمان زیاد برای مسیریابی باعث پایین آمدن راندمان و کارایی شبکه بخصوص در مواردی که طول بسته‌ها کم می‌باشد، می‌شود. در مسیر در شبکه طراحی شده این معایب برطرف شده است و با استفاده از لبه پایین رونده پالس ساعت ورودی فلیت‌ها با همان فرکانس کار شبکه ارسال می‌شوند و زمان مسیریابی نیز کاهش یافته است.

ثابت بودن میزان تأخیر در انتقال بسته‌های با طول‌های مختلف بین دو سوئیچ مشخص یکی دیگر از ویژگی‌های کار انجام شده است. به این معنی که میزان تأخیر برای ارسال بسته‌های با طول‌های مختلف بین دو سوئیچ مشخص مقداری ثابت است.

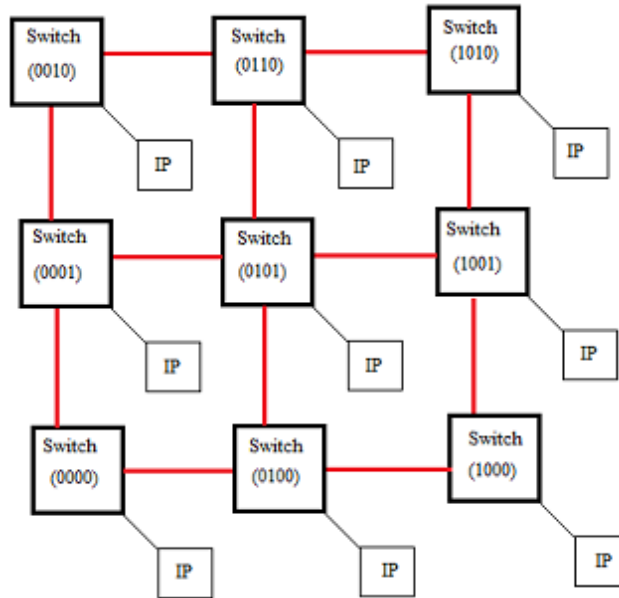
در واقع دلیل اصلی بهبود شبکه طراحی شده استفاده از لبه پایین رونده پالس ساعت ورودی و سخت افزار طراحی شده است که در کنار یکدیگر به بهبود عملکرد شبکه منجر شده است.

در ادامه شبکه طراحی شده و عملکرد بخش های مختلف توضیح داده می شود و پس از آن به نتایج شبیه سازی و تحلیل نتایج به دست آمده در شبکه طراحی شده می پردازیم.

این شبکه با زبان توصیف سخت افزار VHDL در نرم افزار Xilinx ISE نسخه 5 (رایگان) طراحی شده است. مزیت این نسخه نسبت به نسخه های جدیدتر این است که دارای دو حالت سنتز می باشد یک حالت تنها خطا های مربوط به زبان VHDL را بررسی می کند و برای شبیه سازی به کار می رود و حالت دیگر علاوه بر بررسی قواعد زبان VHDL، کد نوشته شده را از نظر قابلیت سنتز شدن بر روی تراشه های شرکت Xilinx نیز بررسی می کند. ابزار سنتز کننده نرم افزار ISE بسیار قدرتمند می باشد بگونه ای که می توان گفت اگر در سنتز خطایی از برنامه نوشته شده گرفته نشود، کد نوشته شده بدون مشکل بر روی FPGA یا CPLD قابل پیاده سازی است، ولی برخلاف توانایی سنتز، شبیه سازی آن ضعیف بوده و همواره برای شبیه سازی از نرم افزار modelsim استفاده می شود. در نسخه های مختلف Xilinx ISE می توان نرم افزارهای دیگری را برای شبیه سازی و سنتز به آن افزود. در نسخه 5 علاوه بر modelsim که برای شبیه سازی استفاده می شود، برای سنتز نرم افزار Leonardo Spectrum که به اختصار LS نامیده می شود نیز قابلیت افزوده شدن به Xilinx ISE را دارد. سنتز کننده نرم افزار ISE تنها از FPGA ها و CPLD های شرکت Xilinx پشتیبانی می کند در حالی که ممکن است کاربر مایل باشد نتیجه سنتز را به صورت ASIC یا تولیدات شرکت Altera نیز مشاهده کند در این حالت با فراخونی نرم افزار LS به راحتی می تواند این کار انجام دهد چون این نرم افزار از همه موارد فوق پشتیبانی می کند. البته در صورت استفاده از نسخه های کرک شده نرم افزارهای معرفی شده ممکن است نتیجه مورد نظر حاصل نشود و با یک پیغام warning کاربر را متوجه غیر معتبر بودن خروجی کنند.

۴-۱ توپولوژی شبکه

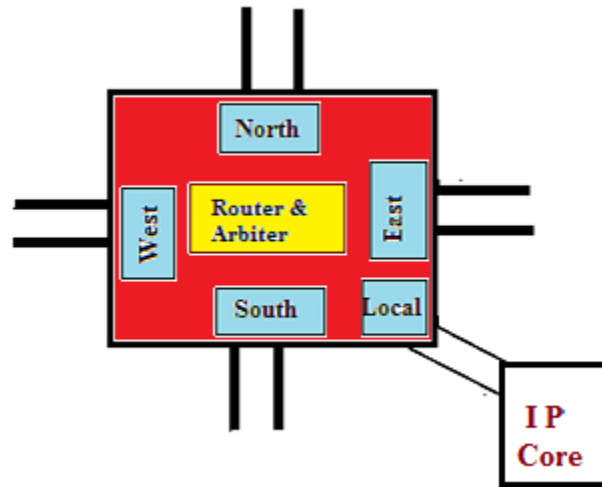
در اینجا یک شبکه 3×3 با توپولوژی مش طراحی شده است که می توان طرح کلی آن را در شکل (۴-۱) مشاهده کرد.



شکل ۴-۱: نمایش توپولوژی شبکه

هر سوئیچ شبکه مانند سوئیچ های شبکه Hermes از پنج پورت تشکیل شده است که با نام های Local, North, South, East و West مشخص شده اند، شکل (۴-۲) یک سوئیچ این شبکه را به صورت شماتیک نمایش

می دهد.



شکل ۴-۲: نمایش شماتیک سوئیچ طراحی شده

ابتدا شبیه سازی را به ازای یک IP انجام می دهیم و در ادامه آن را به چهار IP تعمیم می دهیم.

در ابتدای کد VHDL نوشته شده بسته ی ListConstants قرار داده شده که در آن یک ثابت با نام width معرفی شده است. این ثابت عرض گذرگاهها را که برابر با اندازه هرفلیت می باشد را مشخص می کند و به سادگی قابل تغییر می باشد.

واحدهای IPCore استفاده شده در این پروژه عبارتند از تولید کننده سیگنال تصادفی که با هر پالس ساعت یک عدد باینری به طول width در خروجی قرار می دهد.

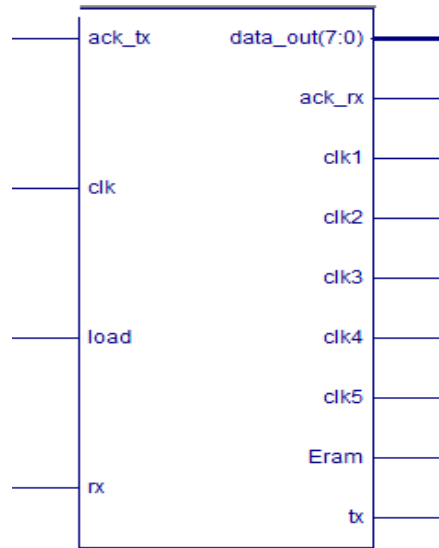
۴-۲ اجزای یک گره

۴-۲-۱ واحد IP

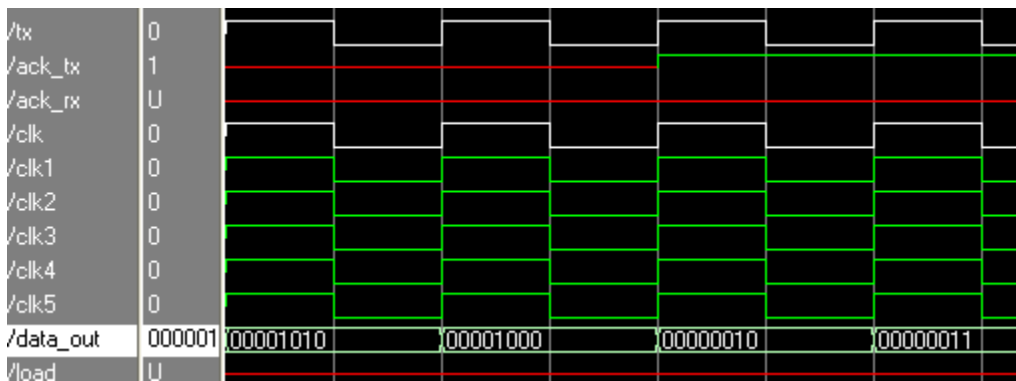
معماری این بخش به این صورت است که با لبه بالا رونده پالس ساعت یک عدد باینری به طول width در خروجی قرار داده می شود، سیگنال کنترلی tx برابر "1" می شود و اگر پایه ack_tx برابر "1" بود آن را

در خروجی قرار می‌دهد در غیر اینصورت tx برابر "0" می‌شود. در واقع همان کنترل جریان بر اساس ارتباط دست تکان دهی در حال اجرا شدن می‌باشد. با هر لبه بالارونده پالس ساعتی که به پایه clk اعمال می‌شود یک داده ۸ بیتی را در صورتی که سیگنال کنترلی ack_rx برابر "1" باشد را بر روی پایه data_out قرار می‌گیرد. در اینجا با کمک یک شمارنده بر فلیت‌های اول و دوم نظارت می‌کنیم و مقداری مناسب به آن‌ها اختصاص می‌دهیم. همانطور که گفته شد روش خزیدنی نوع دوم برای ارسال بسته‌ها به کار رفته است بنابراین فلیت‌های اول دو دوم به ترتیب نشان دهنده آدرس مقصد و تعداد فلیت‌های بسته می‌باشند و باید مقادیری مناسب بگیرند به همین دلیل وقتی مقدار شمارنده برابر یک (معادل باینری "00000001") می‌شود به خروجی مقدار "00001010" و وقتی برابر ۲ می‌شود مقدار "00001000" را به خروجی اختصاص می‌دهیم و معنی آن این است که یک بسته ۸ فلیتی باید به سوئیچ "1010" ارسال شود، بعد از اینکه ۸ فلیت دیگر به خروجی منتقل شدند مقدار شمارنده صفر می‌شود و روند فوق دو باره تکرار می‌شود.

با هر فلیتی که به خروجی منتقل می‌شود سیگنال کنترلی tx نیز "1" می‌شود و در زمان‌هایی که داده‌ای برای ارسال وجود ندارد یعنی در زمان‌هایی که پالس ساعت در لبه پایین رونده می‌باشد "0" می‌شود. شماتیک این بخش را در شکل (۳-۴) و نتیجه شبه‌سازی در شکل (۴-۴) قابل مشاهده می‌باشد.



شکل ۳-۴: نتیجه سنتز یک IP



شکل ۴-۴: ارسال فلیت‌های یک بسته به پایه خروجی توسط یک IP

۴-۲-۲ سوئیچ

همانطور که گفته شد هر سوئیچ از پنج پورت تشکیل شده است. پورت‌ها دو به دو با یکدیگر ارتباط داشته و می‌توانند به یکدیگر داده ارسال کنند و به طور همزمان از پورت‌های دیگر داده‌ها را دریافت کنند، به ازای هر پورت پنج پایه برای ورود داده‌ها و پنج پورت نیز برای خروج داده‌ها اختصاص یافته است، چهار پایه ورودی به همراه چهار پایه خروجی امکان ارتباط پورت‌ها در داخل سوئیچ را فراهم می‌کنند و یک ورودی و یک خروجی باقی‌مانده ارتباط سوئیچ با سوئیچ‌های همسایه از سمت آن پورت را برقرار می‌کنند. به ازای هر ورودی و

خروجی یک جفت سیگنال کنترلی وجود دارد که برای پایه ورودی عبارتند از rx و ack_rx و برای پایه خروجی عبارتند از tx و ack_tx. هر بار که rx فعال شود پورت متوجه می شود که یک فلیت جدید بر روی پایه ورودی قرار گرفته است و وقتی آن را ذخیره کرد با فعال کردن ack_rx به واحد ارسال کننده اطلاع می دهد که فلیت ارسالی را دریافت و با موفقیت ذخیره کرده است. در حالت ارسال یک فلیت نیز اگر سیگنال ack_rx برابر "1" باشد به معنی این است که پورت دریافت کننده فلیت قبلی را با موفقیت دریافت و ذخیره کرده است و آمادگی دریافت فلیت جدید را دارد، و هنگامی که فلیت جدید بر روی پایه خروجی قرار گرفت سیگنال tx را فعال می کند تا به پورت دریافت کننده اطلاع دهد که یک فلیت جدید بر روی پایه ورودی آن قرار گرفته است.

با هر بار فعال شدن سیگنال tx، سیگنال rx در پورت Local سوئیچ متناظر نیز فعال می شود و یک شمارنده در آن شروع به شمارش می کند. همانطور که گفته شد فلیت اول آدرس مقصد را مشخص می کند، بنابراین وقتی مقدار این شمارنده برابر "1" می باشد از فلیت وارد شده ۴ بیت سمت راست آن را به یک پایه خروجی به نام incoming به واحد router منقل کرده تا مسیر مناسب برای بسته مشخص شود. در اینجا به این دلیل فقط از چهار بیت سمت راست فلیت اول استفاده می شود که ما در اینجا ۹ سوئیچ بیشتر نداریم و برای آدرس دهی این تعداد سوئیچ چهار بیت کافی می باشد، از این چهار بیت دوییت سمت راست مختصات Y و دوییت سمت چپ مختصات X آدرس مقصد را مشخص می کنند. همزمان که آدرس مقصد بر روی پایه incoming قرار می گیرد سیگنال h از سوی واحد پورت فعال می شود تا به واحد router اطلاع دهد که به یک آدرس جدید برای ارسال بسته وارد شده نیاز دارد. با فعال شدن سیگنال h از طریق واحد Arbiter سیگنال req_rout فعال می شود. بلافاصله واحد Router یک مقایسه انجام داده و مسیر مناسب را طبق الگوریتمی که برای آن تعریف شده است (در اینجا الگوریتم مسیریابی XY احاطه کننده اجرا شده است) پیدا می کند. با پیدا شدن مسیر، مقدار مناسب را به پایه select_port اختصاص می دهد (مقادیری که پایه select_port می گیرد عبارتند از "000" برای ارسال به پورت North، "001" برای ارسال به پورت South، "010" برای ارسال به پورت East، "011" برای ارسال به

پورت West و "100" برای ذخیره در بافر) و همزمان سیگنال acq_rout را فعال کرده تا مشخص شدن مسیر را اطلاع دهد. با فعال شدن ack_rout واحد Arbiter سیگنال ack_h را فعال کرده و به پورت اجازه می دهد که داده ها را به مسیر مشخص شده ارسال کند. همانطور که در شکل (۴-۵) مشاهده می شود مقدار "010" به select_port اختصاص یافته است.

/h	1						
/ack_h	1						
/select_port	010	UUU	010				
/incoming	1010	UUUU	1010				

شکل ۴-۵: نمایش مشخص شدن مسیر برای ارسال فلیت ها

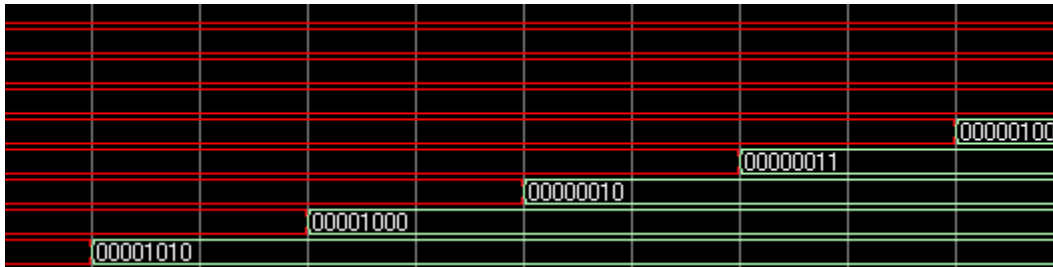
نکته ای که در اینجا باید به آن توجه داشت این است که در فاصله ورود اولین فلیت تا تعیین مسیر مناسب که دو پالس ساعت طول می کشد باید فلیت های وارد شده را ذخیره کرد. برای این منظور از دو ثبات استفاده می شود که نسبت به ورودی تأخیر دارند. داده ها را از ثبات دوم به خروجی مناسب ارسال میشوند بنابراین بسته ها نیز با دو پالس ساعت تأخیر، ارسال خواهند شد. شکل (۴-۶) وضعیت ثبات های reg1 و reg2 را نشان می دهد.

00001010	00001000	00000010	00000011	00000100	00000101
	00001010	00001000	00000010	00000011	00000100

شکل ۴-۶: نمایش وضعیت ثبات های reg1 و reg2

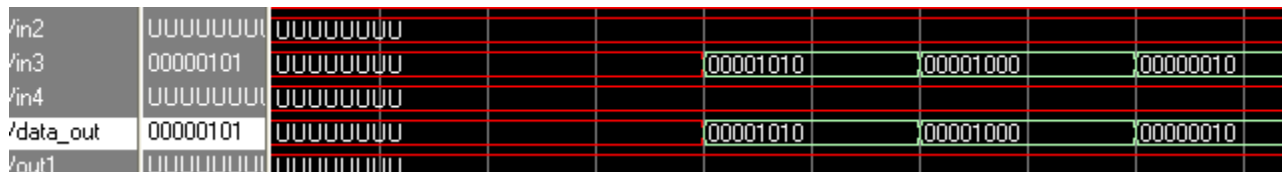
یک بافر نیز برای ذخیره موقت بسته ها در هر پورت قرار داده شده است و در مواردی که مسیر مناسب برای بسته وجود نداشته باشد بسته را در بافر ذخیره می کنیم و به محض اینکه مسیر مناسب آزاد شد فلیت ها از بافر ارسال می شوند. شکل زیر وضعیتی را نشان می دهد که ابتدا مسیر مورد نظر بسته بوده است و فلیت ها در بافر ذخیره می شوند و به محض آزاد شدن مسیر فلیت ها از بافر به بیرون منتقل می شوند. مسئله ای که در اینجا بسیار اهمیت دارد این است که بافر باید بتواند به صورت همزمان از یک طرف فلیت ها را ذخیره کند و از

طرف دیگر آن‌ها را ارسال کند، چون ممکن است وضعیتی پیش بیاید که در آن هنگام ذخیره فلیت‌ها مسیر مناسب باز شود. شکل (۷-۴) وضعیتی را نشان می‌دهد که فلیت‌ها در حال ذخیره شدن در بافر هستند.



شکل ۷-۴: ذخیره فلیت‌ها در بافر

بعد از اینکه پورت مناسب برای خروج یک بسته پیدا شد فلیت‌ها به ترتیبی که وارد می‌شوند از ثبات reg2 به خروجی منتقل می‌شوند. با هر فلیتی که به خروجی منتقل می‌شود باید سیگنال tx را نیز فعال کرد تا سوئیچ بعدی که فلیت‌ها دریافت می‌کند متوجه شود که یک فلیت جدید بر روی پایه ورودی آن قرار گرفته است (توجه داشته باشید که سیگنال‌های tx و tx دو سوئیچ همسایه به یکدیگر متصل هستند). در این مرحله کد VHDL نوشته شده حداکثر کارایی را دارد و با هر پالس ساعت فلیت‌ها به پایه خروجی منتقل می‌شوند. یکی از مهمترین ویژگی‌های کار انجام شده که در ادامه بیشتر توضیح داده خواهد شد این است که فلیت‌ها با همان فرکانس پالس ساعت ورودی منتقل می‌شوند. شکل (۸-۴) ارسال فلیت‌ها به خارج از سوئیچ را نشان می‌دهد.

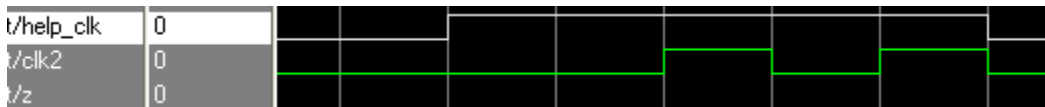


شکل ۸-۴: ارسال فلیت‌ها به خارج از سوئیچ

که اگر تعداد پاس‌های سیگنال tx دقیقاً برابر با تعداد فلیت‌ها باشد سوئیچ بعدی که فلیت‌ها را دریافت می‌کند با مشکل مواجه خواهد شد چون دو پالس اول را برای مسیر یابی مصرف می‌کند و بعد از آن شروع به

ارسال فلیت ها می کند بنابراین دو پالس tx کم خواهد آمد که باعث می شود در هر سوئیچ دو فلیت از دست برود. برای رفع این مشکل از یک پالس کمکی استفاده شده است. به این صورت که بعد از تمام شدن ارسال یک بسته یک پالس کمکی فراخوانی می شود تا دو پالس ساعت تولید کند. برای این منظور وقتی همه فلیت ها به خارج از سوئیچ ارسال شد یک سیگنال به نام help_clk فعال می شود و به دنبال آن سیگنال دیگری به clk2 مقدار گرفته باعث می شود پروسس مربوط به tx بار دیگر اجرا شود. وقتی tx دقیقاً ۲ بار دیگر فعال شد سیگنال help_clk صفر شده و ارسال بسته پایان می یابد و برای نشان دادن تمام شدن ارسال بسته و آزاد بودن پایه خروجی یک سیگنال کنترلی به نام free را که در زمان ارسال فلیت ها صفر شده بود را فعال می کند.

شکل (۹-۴) فعال شدن و غیرفعال شدن سیگنال help_clk را نمایش می دهد.



شکل ۹-۴: نمایش استفاده از سیگنال کمکی clk2

۳-۲-۴ مسیریاب (Router)

همانطور که از نام این واحد مشخص است وظیفه این واحد مشخص کردن مسیر مناسب برای ادامه حرکت یک بسته می باشد. برای مسیریابی هر یک از پنج پورت، پنج پروسس مجزا اختصاص یافته است و با توجه به الگوریتم مسیریابی، سمتی که داد ها وارد شده اند و مقصد بسته و نیز وضعیت شبکه (از قبیل اینکه کدام پورت مشغول است و وضعیت بافر) مسیر مناسب را انتخاب می کند و با فعال کردن ack_rout اطلاع می دهد که مقدار اختصاص یافته به پایه select_port معتبر می باشد.

برای خارج کردن بسته ها از بافر هر پورت نیز پنج پروسس اختصاص یافته است. این پروسس ها به محض اینکه یک پورت آزاد شد بررسی می کند که آیا بافری هست که منتظر باز شدن آن پورت باشد یا خیر، اگر بود یک سیگنال کنترلی به نام `buffer_E` را فعال کرده و فلیت ها را از بافر به پورت مناسب ارسال می کند.

۴-۲-۴ ناظر (Arbiter)

این قسمت وظیفه هماهنگی بین واحد های Router و Switch را بر عهده دارد. به این صورت که در خواست های مسیریابی واحد های Switch را به Router اطلاع میدهد و به دنبال آن پیدا شدن مسیر را به سوئیچ اطلاع می دهد. تمام درخواست ها از سوی پنج پورت و نیز وضعیت هر پورت (اینکه پورت آزاد است یا مشغول) و بافر آن توسط این بخش در اختیار Router قرار می گیرد و مسیریابی بر اساس آن صورت می گیرد.

۴-۳ روند کلی ارسال یک بسته

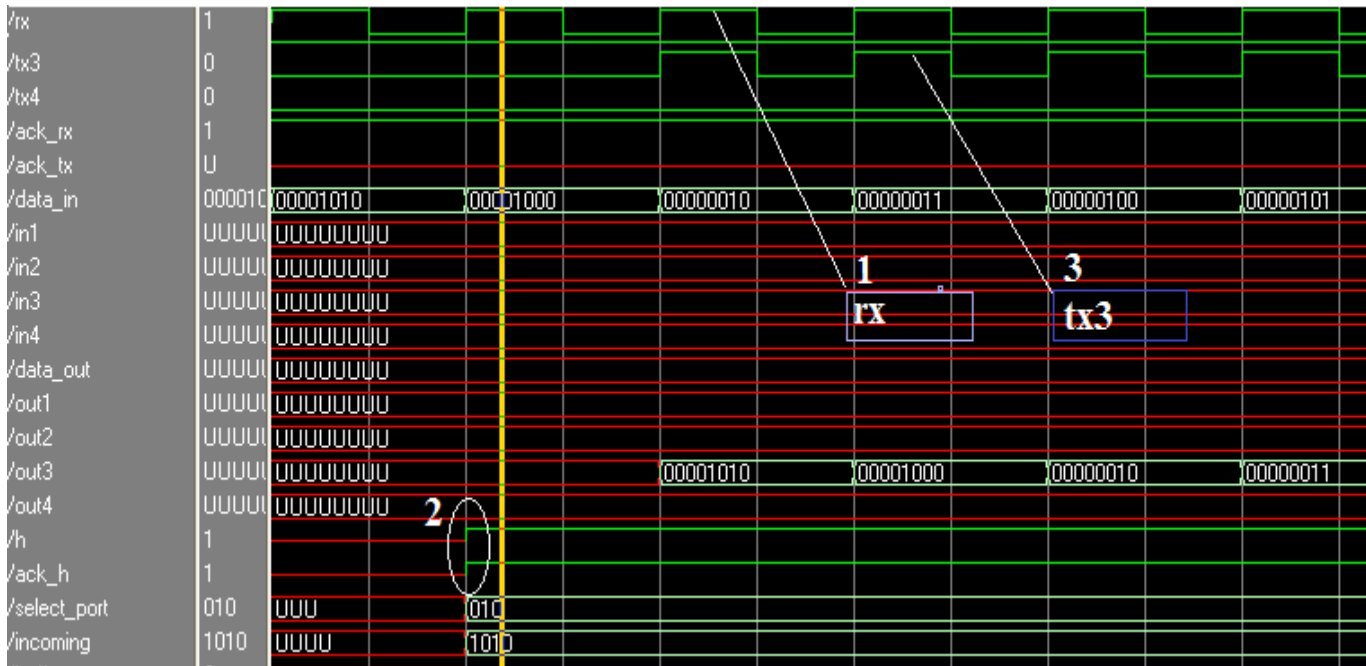
به طور خلاصه ارسال یک بسته را می توان به سه مرحله تقسیم کرد:

مرحله اول: ورود فلیت ها و فعال شدن سیگنال tx

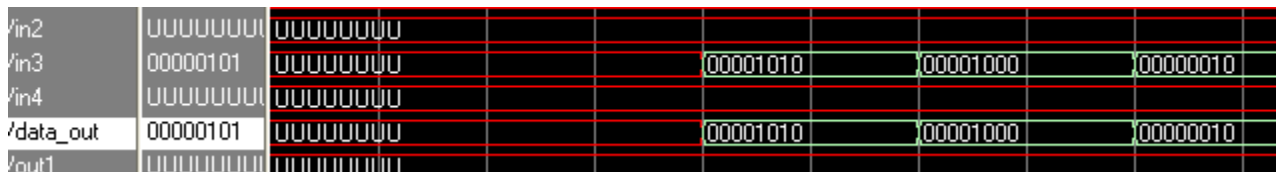
مرحله دوم: در خواست مسیر و مشخص شدن مسیر

مرحله سه: انتقال فلیت ها به خروجی مشخص شده و فعال شدن سیگنال tx

این مراحل در شکل های (۴-۱۰) و (۴-۱۱) نمایش داده شده است.



شکل ۴-۱۰: ارسال یک بسته از پورت Local به پورت East

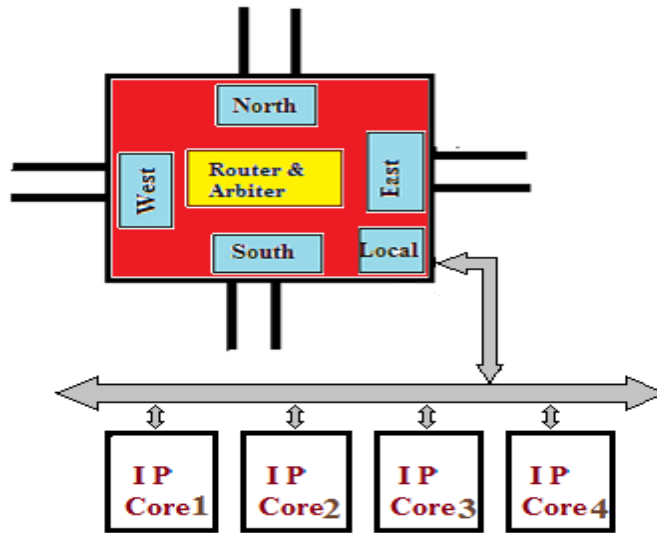


شکل ۴-۱۱: ارسال فلیت ها از پورت East به خارج از سوئیچ

۴-۳ شبکه با گذرگاه اشتراکی

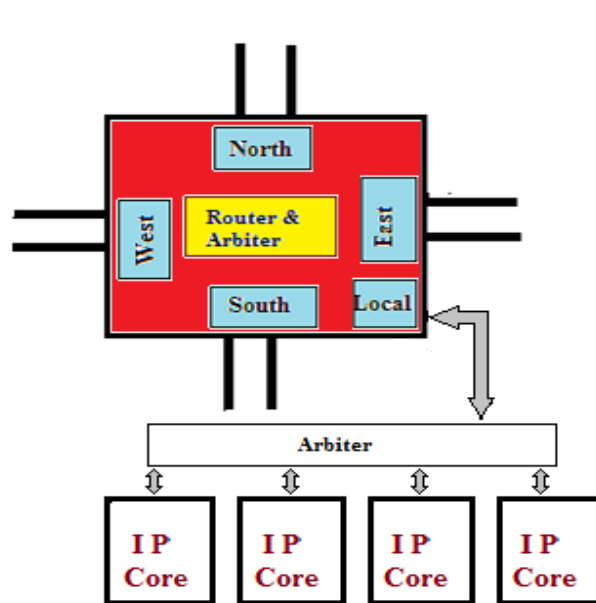
در این مرحله در جهت تکمیل کردن پایان نامه به جای یک IP مجموعه ای از IP ها که با یک گذرگاه اشتراکی با یکدیگر ارتباط دارند و یک شبکه محلی را تشکیل داده اند را به کار می بریم. IPها توسط یک گذرگاه اشتراکی با یکدیگر در ارتباط بوده و توسط یک واحد Arbiter سازماندهی می شوند به این صورت که هرگاه یک IP قصد ارسال یک بسته را به پورت Local داشته باشد یک سیگنال کنترلی به نام req_bus را به واحد Arbiter فرستاده و واحد Arbiter با توجه به اینکه گذرگاه ارتباطی با پورت Local آزاد است یا خیر، با

درخواست آن موافقت می‌کند یا اینکه آنرا در صف انتظار قرار می‌دهد. شکل (۴-۱۲) ارتباط سوئیچ با چند IP را نمایش می‌دهد.



شکل ۴-۱۲: ارتباط چهار IP با یک گذرگاه ارتباطی با سوئیچ

ارتباط نشان داده شده در شکل (۴-۱۲) قابل سنتز شدن و پیاده بر روی تراشه FPGA نیست چون به یک سیم (شی سیگنال) نمی‌توان بیش از یک IP وصل کرد. برای اینکه ملاحظات سخت افزار نیز در نظر گرفته باشیم بدون اینکه به کلیت طرح خدشه وارد شود سوئیچ شبکه محلی را به صورت نمایش داده شده در شکل (۴-۱۳) تغییر می‌دهیم. همانطور که مشاهده می‌شود از یک تنظیم کننده دیگر استفاده شده است و وظیفه آن این است که درخواست های ارسال IPها را سازماندهی کند. این واحد که در کد VHDL به نام Arbiter_bus نامگذاری شده است با توجه به اینکه گذرگاه آزاد است یا خیر به واحدهای IP که قصد استفاده از گذرگاه را دارند، اجازه ارسال یک بسته را صادر می‌کند یا آن را در صف انتظار قرار می‌دهد. در مواردی که دویا چند IP درخواست استفاده از گذرگاه را داشته باشند اولویت را به IPای که شماره کمتری دارد اختصاص می‌یابد.



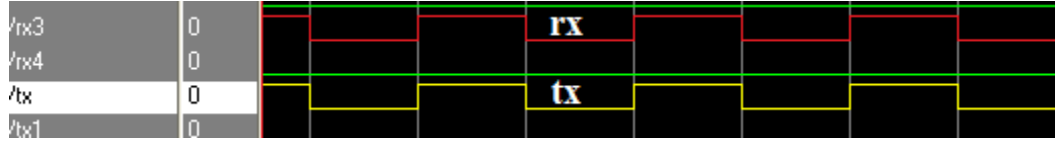
شکل ۴-۱۳: ارتباط چند IP توسط یک ناظر با سوئیچ

۴-۴ تحلیل نتایج

در این بخش به تحلیل نتایج و مقایسه کار انجام شده و شبکه Hermes می پردازیم.

۴-۴-۱ فرکانس سیگنال های rx و tx

یکی از مهمترین ویژگی های کار انجام شده برابر بودن فرکانس سیگنال های کنترلی rx و tx با فرکانس پالس ساعت ورودی می باشد، در حالی که در شبکه Hermes فرکانس این سیگنال ها نصف فرکانس پالس ساعت ورودی است. این امر به معنی آن است که سرعت انتقال فلیت ها در شبکه طراحی شده دوبرابر شبکه Hermes می باشد. این نکته را می توان در شکل های (۴-۱۴) و (۴-۱۵) مشاهده کرد.

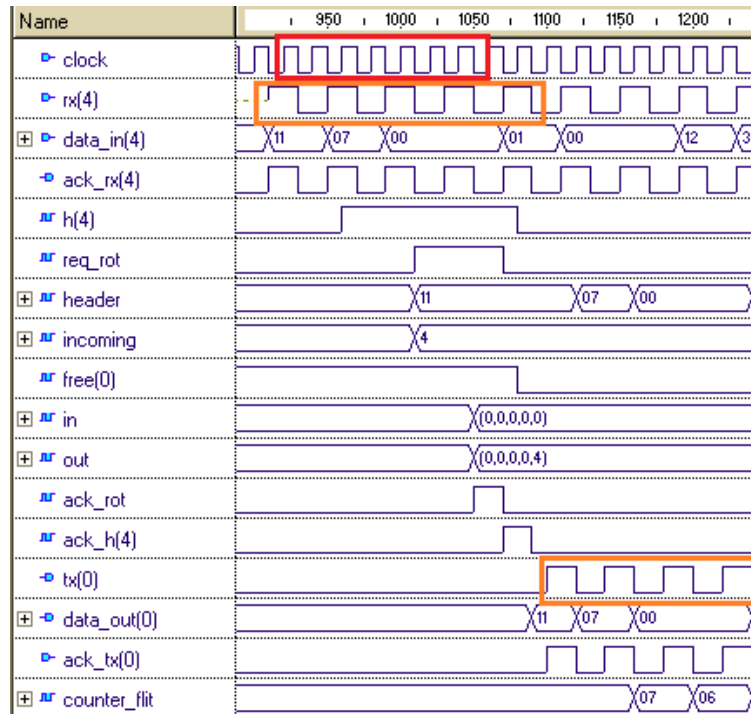


شکل ۴-۱۴: سیگنال های rx و tx



شکل ۴-۱۵: سیگنال های tx و clk

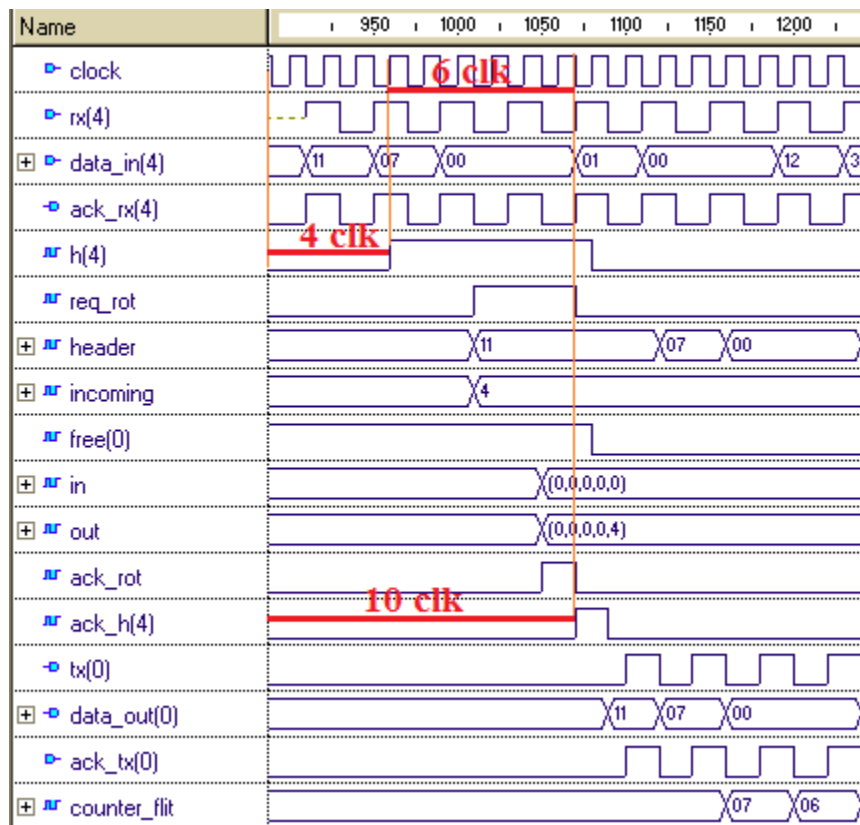
شکل (۴-۱۶) فرکانس سیگنال های rx و tx را در شبکه Hermes نشان می دهد.



شکل ۴-۱۶: مقایسه فرکانس سیگنال های rx و tx با فرکانس پالس ساعت ورودی در شبکه Hermes

۴-۴-۲ کاهش زمان مسیریابی

مزیت دیگری که شبکه طراحی شده نسبت به شبکه Hermes دارد کاهش یافتن زمان مسیریابی است. همانطور که در شکل (۴-۱۷) نشان داده شده است در شبکه Hermes از زمان ورود اولین فلیت یک بسته تا زمان مشخص شدن مسیر مناسب برای آن بسته ۱۰ پالس ساعت طول می کشد یعنی بعد از ۱۰ پالس ساعت فلیت ها به خروجی منتقل می شوند ولی همانطور که قبلاً نیز ذکر شد در شبکه طراحی شده این مدت زمان تنها ۲ پالس ساعت می باشد.



شکل ۴-۱۷: نمایش مدت زمان مسیریابی در شبکه Hermes

۳-۴-۴ میزان تأخیر

یکی دیگر از ویژگی های شبکه طراحی شده در مستقل بودن میزان تأخیر نسبت طول بسته می باشد. به این معنی که میزان تأخیری که برای ارسال یک بسته از یک سوئیچ مشخص به یک سوئیچ مشخص دیگر ایجاد می شود برای بسته های با طول مختلف ثابت می باشد و از رابطه زیر قابل محاسبه می باشد.

$$\text{میزان تأخیر} = 2 \times \sum_{i=1}^d s_i \quad (۱-۴)$$

که در آن d تعداد سوئیچ هایی می باشد که در مسیر حرکت بسته برای رسیدن به مقصد قرار دارد. برای نشان دادن درستی این رابطه بسته هایی با طول های مختلف از سوئیچ "0000" به سوئیچ "1010" ارسال می شوند مورد شمارش قرار گرفته و نتیجه را ثبت کردیم. چون الگوریتم مسیریابی بر اساس الگوریتم XY می باشد با توجه به شکل (۱-۴) بسته مسیر زیر را برای رسیدن به مقصد طی می کند:

"0000" → "0100" → "1000" → "1001" → "1010"

بنابراین چون پنج سوئیچ در مسیر بسته قرار انتظار داریم که بسته با ۱۰ پالس ساعت تأخیر به خروجی منتقل شود. نتایج را می توان در جدول (۱-۴) مشاهده کرد.

جدول ۱-۴: تعداد پالس های ساعت مورد نیاز برای بسته های مختلف

تعداد پالس ساعت مصرفی	طول بسته (فلیت)	تعداد فلیت ها
۲۰	۸	۱۰
۲۸	۱۶	۱۸
۳۶	۲۴	۲۶

همانطور که در جدول (۳-۱) مشاهده می شود میزان تأخیر دقیقاً برابر ۱۰ پالس ساعت می باشد که درستی رابطه (۴-۱) را تأیید میکند.

۴-۴-۴ سنتز

مهمترین ویژگی یک کد VHDL قابل سنتز بودن آن است. چون در غیر اینصورت کد نوشته شده کاربردی نخواهد داشت. در این پایان نامه همانند کار آقای Moraes در [۳۴] نتیجه سنتز یک شبکه ۲×۲ مورد توجه قرار گرفت. در ابتدا تلاش بر مشاهده سنتز بر تراشه Virtex2- xc2v1000fg456 (تراشه ای که شبکه Hermes بر روی آن پیاده شده است) تا بتوانیم از نظر سخت افزاری و حداکثر فرکانس کاری شبکه طراحی شده را با شبکه Hermes نیز مقایسه کنیم ولی به علت در دسترس نبودن نسخه های اصلی نرم افزار های Xilinx ISE و LeonardoSpectrum موفق به انجام این کار نشدیم چون نسخه های کرک شده و رایگان از این تراشه پشتیبانی نمی کنند. شماره یکی از تراشه هایی که توسط نسخه ISE موجود پشتیبانی می شود شباهت زیادی به شماره تراشه مورد نظر دارد. این تراشه از خانواده Spartan3 بوده و شماره آن xc3s1000fg456 می باشد. سنتز را بر روی این تراشه انجام دادیم، نتیجه رضایت بخش بود. میزان بافر اختصاص یافته به هر پورت یک کیلو بایت با واحد های هشت بیتی می باشد (۱۰۲۴ واحد هشت بیتی). میزان سخت افزار اشغال شده در شکل (۴-۱۸) که خروجی نرم افزار ISE می باشد قابل مشاهده است.

Network Project Status (01/17/2013 - 21:20:06)			
Project File:	bistoseAbansaate Ishab.xise	Parser Errors:	No Errors
Module Name:	Network	Implementation State:	Synthesized
Target Device:	xc3s1000-4fg456	• Errors:	No Errors
Product Version:	ISE 12.1	• Warnings:	13518 Warnings (12111 new)
Design Goal:	Balanced	• Routing Results:	
Design Strategy:	Xilinx Default (unlocked)	• Timing Constraints:	
Environment:	System Settings	• Final Timing Score:	

Device Utilization Summary (estimated values)				
Logic Utilization	Used	Available	Utilization	
Number of Slices	400	7680		5%
Number of Slice Flip Flops	256	15360		1%
Number of 4 input LUTs	740	15360		4%
Number of bonded IOBs	168	333		50%
Number of GCLKs	8	8		100%

شکل ۴-۱۸: نتیجه سنتز

همانطور که مشاهده می شود سخت افزار قابل توجهی اشغال نشده است. همچنین توسط نرم افزار حداکثر فرکانس کار شبکه طراحی شده ۱۰۵ مگاهرتز تعیین شد. در برگی مشخصات^۱ حداکثر فرکانس کاری این تراشه ۲۸۰ مگا هرتز مشخص شده است. بنابراین می توان گفت شبکه طراحی شده تقریباً با ۳۷ درصد حداکثر فرکانس تراشه، کار می کند. این مقدار تقریباً چهار برابر فرکانس کار شبکه Hermes می باشد. اگرچه از این نظر نمی توان مقایسه درستی انجام داد چون بر روی دو تراشه متفاوت سنتز شده اند.

۴-۴-۵ پیشنهادهایی برای کار در آینده

پیشنهاد اول:

از یک دیدگاه می توان سوئیچ طراحی شده را مانند یک چهارراه و واحد مسیریاب را به عنوان چراغ راهنما در نظر گرفت. در دنیای واقعی در بعضی از چهارراه ها که تردد اتومبیل ها در آن ها زیاد است با ایجاد یک زیرگذر بار ترافیک را کم می کنند. در اینجا نیز می توان با یک زیرگذر در برخی از سوئیچ ها که بار ترافیک شبکه در آن زیاد است مانع از به بن بست رسیدن شبکه شد.

¹ Data Sheet

پیشنهاد دوم:

از دیدگاه دیگر حرکت بسته ها در شبکه را می توان مانند حرکت مهره رُخ در بازی شطرنج در نظر گرفت چون حرکت آن ها به حرکت در راستای عمودی و افقی محدود می شود. دلیل این محدودیت ماهیت سوئیچ ها می باشد که تنها پنج پورت دارند ولی اگر سوئیچ ها ۸ وجهی باشند آنگاه بسته ها را می توان مانند مهره وزیر حرکت داد که قدرت مانور بیشتری دارد و بسته ها مسلماً زودتر به مقصد می رسند.

مراجع

- [1]. *From a Few Cores to Many: A Tera-scale Computing Research Overview* **J. Held, J. Bautista, and S. Koehl**.2006 ,Intel White Paper.
- [2]. <http://www.xbitlabs.com/articles/mobile/display/asus-v6x00j.html> .
- [3]. *A Formal Language to Model SOC For Synthesis And Verification* **Palaniveloo, Vinitha A .** Sydney : School of Computer Science Engineering, September 24, 2007.
- [4]. *Asynchronous Implementation of a Distributed Network-on-Chip* **SHEIBANYRAD, Abbas .** Paris : in the Computer Science Laboratory of Paris 6, 2008.
- [5]. *Networks on Chips:A New SoC Paradigm* **Luca Benini, Giovanni De Micheli** .2002 .
17.00\$/.2/9162-00IEEE.
- [6]. *An Overview of On-Chip Buses* **Stojčev, Milica Mitić and Mil.** FACTA UNIVERSITATIS, 2006.
- [7]. <http://www.arm.com> **ARM AMBA Specification and Multi layer AHB Specification rev2.0**.(2001).

- [8]. *System-level power-performance trade-offs in bus matrix communication architecture synthesis* **S. Pasricha, Y. Park, F. Kurdahi and N. Dutt** .Seoul :International Conference on Hardware, 2006.
- [9]. *Extending Open Core Protocol to* . **Konstantinos Aisopos, Chien-Chun Chou, Li-Shiuan Peh** .Atlanta 2008.
- [10]. *OCP Datasheet (OCP-IP), Open Core Protocol International Partnershi* :
<http://www.ocpip.org>., Release Version.
- [11]. *Virtual Component Interface Group, VSI Alliance™ On-Chip Bus Development Working* .April 2001.
- [12]. *Device Transaction Level (DTL) Protocol Specification* **Semiconductors, Philips** : Version 2.4, February 2005.
- [13]. *A Survey of Research and Practices of Network-on-Chip* **MAHADEVAN, TOBIAS BJERREGAARD AND SHANKAR** .ACM Computing Surveys ,Technical University of Denmark, 2006.
- [14]. *A network-on-chip architecture and design methodology* **KUMAR, S., JANTSCH, A., SOININEN, J.-P., FORSELL, M., MILLBERG, M., OBERG, J., TIENSYRJÄ, K., AND** .
IEEE Computer Society, Proceedings of the Computer Society Annual Symposium on VLSI (ISVLSI), 2002.
- [15]. *A survey of design techniques for systemlevel dynamic power management* **L. Benini, A. Bogliolo, and G. De Micheli** .IEEE Trans. Very Large Scale Integr, 2000 .pp. 299-316.

- [16]. *BENoC: A Bus-Enhanced Network on-Chip for a Power Efficient CMP* **Isask'har Walter¹, Israel Cidon², and Avinoam Kolodny** :Electrical Engineering Department, Technion – Israel Institute of Technology, Israel, 2008. جلد ٧ ,
- [17]. *Communication Performance in Network-on-Chips* **Jantsch, Axel** .Stockholm :Royal Institute of Technology, 2004.
- [18]. *A Tree-Based Topology Synthesis for On-Chip Network* **Jason Cong, Yuhui Huang, and Bo Yuan** .Los Angeles for On-Chip Network : Computer Science Department.
- [19]. *Simulation and Analysis of Network on Chip Architectures: Ring, Spidergon and 2D Mesh* . **Luciano Bononi, Nicola Concer** .Bologna, Italy : Dipartimento di Scienze dell'Informazione, Università degli Studi di Bologna,, 2006.
- [20]. *SPIN: a Scalable, Packet Switched, On-chip Micro-network* **Adrijean Adriahtenaina, Hervé Charlery, Alain Greiner** .IEEE Computer Society.Proceedings of the conference on Design, Automation and Test in Europe, 2003.
- [21]. *An Interconnect Architecture for Networking Systems on Chips* **F. Karim, A. Nguyen, and S. Dey** .IEEE Micro .pp. 36-45.
- [22]. *Route packets, not wires: on-chip inteconnection networks* .**Towles, J. W. Dally and B** . Proceedings of the 38th conference on Design automation, 2001.
- [23]. *Network on Chip Routing Algorithms* .**Ville Rantala, Teijo Lehtonen, Juha Plosila** . University of Turku, Department of Information Technology, 2006.

- [24]. *Evaluation of Pseudo Adaptive XY Routing Using an Object Oriented Model for NOC* **M. Dehyadgari, M. Nickray, A. Afzali-kusha, Z. Navabi** : The 17th International Conference on Microelectronics, 2005.
- [25]. *An Energy-Efficient Reconfigurable Circuit-Switched Network-on-Chip* **Pascal T. Wolkotte, Gerard J.M. Smit, Gerard K. Rauwerda, Lodewijk T. Smit**.2005.
- [26]. *A New Router Architecture for High-Performance Intrachip Networks* **Everton Carara, Ney Calazans, Fernando Moraes** .Porto Alegre : 2008.
- [27]. *Design of a High-Performance Buffered Crossbar Switch Fabric Using Network on Chip* . **Sen'm, Iria Varela** .Netherlands : 2008 .CE-MS-2008-19.
- [28]. *HW-SW COMPONENTS FOR PARALLEL EMBEDDED COMPUTING ON NOC-BASED MP-SOCs* **Murillo, Jaume Joven** .Bellaterra, Spain: 2009
- [29]. *Real-Time Communication Services for Networks on Chip* **Shi, Zheng** .Computer Science November 2009
- [30]. *Evaluating Bufferless Flow Control for On-Chip Networks* **George Micheliogiannakis, Daniel Sanchez, William J. Dally, Christos Kozyrakis** .Appears in the Proceedings of the 4th ACM/IEEE International Symposium on Networks-on-Chip, 2010.
- [31]. *Fault Tolerance Overhead in NetworkonChip Schemes* **Antonio Pullini, Federico Angiolini, Davide Bertozzi** .Florianópolis, Brazil .2005
- [32]. *Deadlock-free message routing in multiprocessor interconnection networks* **Seitz, W. J. Dally and C. L.** IEEE Trans. Comput .pp. 547-553, 1987.

- [33]. *Fault Tolerant Source Routing for Network-on-chip* . **Young Bok Kim, Yong-Bin Ki** .
Northeastern University, Boston, MA ,USA : Dept. of Electrical and Computer Engineering.
- [34]. *HERMES: an Infrastructure for Low Area Overhead Packet-switching Networks on Chip* .
**HERMES: an Infrastructure for Low Area Overhead Mello, Leandro Heleno Möller,
Luciano Copello Ost** .October, 2003.
- [35] D. Bertozzi *et al.*, “NoC **synthesis flow for customized domain specific multiprocessor systems-on-chip**,” *IEEE Trans. Parallel and Distributed Systems*, vol. 16, no. 2, pp. 113–129, Feb. 2005.
- [36] J. Liang *et al.*, **An architecture and compiler for scalable on-chip communication** *IEEE Trans. VLSI Syst.*, vol. 12, no. 7, pp. 711–726, Jul. 2004.
- [37] T. Henriksson, D. Wiklund, and D. Liu, **VLSI implementation of a switch for on-chip networks**,in *DDECS*, 2003
- [38] D. Sigüenza-Tortosa, T. Ahonen, and J. Nurmi, **Issues in the development of a practical NoC: the Proteo concept**, *Integration, the VLSI Journal*, vol. 38, no. 1, pp. 95–105, Oct. 2004
- [39] A. Andriahantenaina and A. Greiner, **Micro-network for SoC: Implementation of a 32-port SPIN network**, in *DATE*, Mar. 2003, pp. 1128–1129.
- [40] E. Rijpkema *et al.*, **Trade offs in the design of a router with both guaranteed and best-effort services for network on chip (extended version)**, *IEE Proc. Computers and Digital Techniques*, vol. 150, no. 5, pp. 294–302, Sep. 2003.
- [41] F. Karim, A. Nguyen, and S. Dey, **An interconnect architecture for networking systems on chips**, *IEEE Micro*, vol. 22, no. 5, pp. 36–45, Sep.-Oct. 2002
- [42] M. Millberg, R. T. E. Nilsson, and A. Jantsch, **Guaranteed bandwidth using looped containers in temporally disjoint networks within the Nostrum network on chip**, in *DATE*, Feb. 2004, pp. 890–895.
- [43] E. Bolotin *et al.*,**QNoC: QoS architecture and design process for network on chip**, *Journal of Systems Architecture*, vol. 50, no. 2–3, pp. 105–128, Feb. 2004.

Abstract

Increasing demands for chips with higher efficiency cause the complexity of integrated circuit design to increase. Although the problem has been fixed mostly by making smaller chips, the technology of having smaller disturbs the tradeoff between wires and gates delays. Moreover, as the speed was increased, consume power of chip was increased subsequently. To tackle these challenges, designers of integrated circuits focus on increasing the performance rather than speed. Thus, two separated processors were placed in a single chip and a bus was used to establish connections between them. The result was satisfactory so the systems consisting of several parts played on a board moved to be in a single chip. The architect of making processors like this is generally known as System on Chip (SOC). However, this was not the end of story, as the SOC has also problems that were recognized during the time of usages. For instance, increasing the number of separate parts (also known as IP) causes the system to not respond to any other chips. Problems such as lack of scalability and significant power consumption in bus lead to new attempts carried out to design of integrated circuits. The main concentration this time was on the architect of Network on Chip.

The idea of utilizing Network on chip based on large scale computer network concepts causes the OSI standard layers offers to be used. Since this architect was very efficient and fast, it was replaced with other architects. Hence, architects having various features and capabilities were considered for more investigation.

In this thesis, a network with NOC architect known as Hermes was simulated in order to increase its capabilities. One of the disadvantages of Hermes network is that its efficiency is reduced with increasing the number of IPs. On the other hand, IPs may have information exchanges so if they make a local network, their communications will be faster. Thus, rather than utilizing a single IP, each switch was connected to four IPs linked by a bus. Therefore, we have a combination of NOS and SOC. The term "hybrid" in the title of the thesis is referring to this fact. In addition, flits were transmitted with the frequency of network so routing time was reduced and speed was increased as well.



Shahrood University of Technology

Faculty of Electrical and Robotic Engineering

**Design and Simulation a Hybrid Network On Chip With Hardware
Description Language VHDL**

Seyed Mansoor Sharifi

Supervisors:

Dr.Omidreza Ma'rozi

Advisor:

Dr.Abbas Ramezani

FOR THE DEGREE OF MASTER OF SCIENCE

February 2013