دانشگاه صنعتی اصفهان

دانشکده برق و کامپیوتر

ارائه ی یک معماری مبتنی بر ترکیب باس و شبکه روی تراشه جهت سیستم های روی تراشه و توسعه ی الگوریتم نگاشت و مسیریابی درآن

پایان‌نامه کارشناسی ارشد مهندسی کامپیوتر-معماری

مهسا موسوی

اساتید راهنما

دکتر پژمان خدیوی- دکتر کیارش بازرگان

۱۳۹۰

به نام هستی بخش یکتا

دانشگاه صنعتی اصفهان

دانشکده برق و کامپیوتر

ارائه ی یک معماری مبتنی بر ترکیب باس و شبکه روی تراشه جهت سیستم های روی تراشه و توسعه ی الگوریتم نگاشت و مسیریابی درآن

پایان‌نامه کارشناسی ارشد مهندسی کامپیوتر-معماری

مهسا موسوی

اساتید راهنما

دکتر پژمان خدیوی- دکتر کیارش بازرگان

۱۳۹۰

دانشگاه صنعتی اصفهان

دانشکده برق و کامپیوتر

پایان‌نامه‌ی کارشناسی ارشد رشته‌ی مهندسی کامپیوتر – معماری خانم مهسا موسوی

تحت عنوان

ارائه ی یک معماری مبتنی بر ترکیب باس و شبکه روی تراشه جهت سیستم های روی تراشه و توسعه ی الگوریتم نگاشت و مسیریابی درآن

در تاریخ ۹۰/۲/۲۶ توسط کمیته‌ی تخصصی زیر مورد بررسی و تصویب نهایی قرار گرفت.

۱- استاد راهنمای پایان‌نامه	دکتر پژمان خدیوی
۲- استاد راهنمای پایان‌نامه	دکتر کیارش بازرگان
۳-استاد داور	دکتر محمد علی منتظری
سرپرست تحصیلات تکمیلی دانشکده	دکتر سید محمود مدرس هاشمی

بر خود لازم می‌دانم که از تمام کسانی که مرا برای انجام این پایان نامه یاری کردند تشکر کنم. به ویژه جناب آقای دکتر پژمان خدیوی و دکتر کیارش بازرگان که راهنماییهایشان بسیار ارزنده بود. از آقایان دکتر محمد علی منتظری و دکتر شادرخ سماوی که در جلسه دفاعیه اینجانب شرکت نمودند و آقای دکتر سید محمود مدرس هاشمی سرپرست تحصیلات تکمیلی دانشکده برق و کامپیوتر کمال تشکر را دارم. از همسر عزیزم، آقای مهندس محمد تحقیقی،که در این راه همراه و دلسوز من بودند و همچنین از پدر و مادر عزیزم به خاطر حمایت های معنویشان، قدردانی و تشکر فراوان می نمایم.

کلیه‌ی حقوق مادی مترتب بر نتایج مطالعات، ابتکارات و نوآوری‌های ناشی از تحقیق موضوع این پایان‌نامه (رساله) متعلق به دانشگاه صنعتی اصفهان است.

تقدیم به همسر مهربانم

و

پدر و مادر عزیزم

فهرست مطالب

عنوان **صفحه**

فهرست مطالب ..... هشت

چکیده ۱

فصل اول: مقدمه

فصل دوم: شبکه روی تراشه

۲-۱ تاریخچه پیدایش شبکه روی تراش ۶

۲-۲ شبکه روی تراشه ۷

۲-۲-۱ نکات مهم در طراحی سیستم های بزرگ ۷

۲-۲-۲ گرایش ها و چالش ها ۹

۲-۲-۳ شبکه روی یک تراشه ۱۱

۲-۲-۴ راه کار های شبکه روی تراشه ۱۲

۲-۳ شبکه روی تراشه در سیستم های روی تراشه ۱۵

۲-۴ اصول شبکه روی یک تراشه ۱۷

۲-۴-۱ مشخصه های معماری ۱۸

۲-۴-۲ چکیده ای از شبکه ۲۰

۲-۴-۳ آداپتور شبکه ۲۲

۲-۴-۴ شبکه ۲۴

۲-۴-۵ سطح ارتباط ۳۷

۲-۵ نمونه هایی از شبکه روی یک تراشه ۳۹

۲-۵-۱ نمونه ی EATHEREAL ۳۹

۲-۵-۲ نمونه ی NOSTRUM ۴۰

۲-۵-۳ نمونه ی SPIN ۴۰

۲-۵-۴ نمونه ی CHAIN ۴۰

۲-۵-۵ نمونه ی MANGO ۴۰

۲-۵-۶ نمونه ی XPIPES ۴۱

۲-۶ نتیجه گیری ۴۱

فصل سوم: توصیف مساله ی توپولوژی و نگاشت

۳-۱ روند طراحی شبکه ی روی یک تراشه ۴۲

۳-۲ هدف ها در طراحی شبکه روی تراشه ۴۴

۳-۳ ابزارهای طراحی شبکه روی تراشه ۴۵

۳-۴ موضوعات مرتبط با گام های نگاشت و زمانبدی ۴۵

۳-۵ مثالی از مساله نگاشت و زما ن بندی ۴۷

۳-۶ روش مدل کردن مراحل طراحی شبکه روی تراشه ۴۷

۳-۷ بیان مساله طراحی شبکه روی تراشه به صورت ریاضی ۵۱
۳-۸ ترکیب توپولوژی و مسائل نگاشت ۵۲
۳-۹ الگوریتم نگاشت NMAP ۵۶
۳-۱۰ نتیجه گیری ۵۷
فصل چهارم: توپولوژی و نگاشت پیشنهادی
۴-۱ ترکیب باس و شبکه روی تراشه ۵۹
۴-۱-۱ مقایسه ی توپولوژی ترکیبی با دیگر توپولوژی ها ۶۰
۴-۱-۲ تعیین حداکثر تعداد هسته های پردازشی متصل به یک باس ۶۳
۴-۱-۳ تعداد هسته های پردازشی که یه یک باس متصل هستند ۶۶
۴-۲ طراحی توپولوژی پیشنهادی ۶۷
۴-۲-۱ حداکثر تعداد هسته های پردازشی متصل به یک باس ۶۷
۴-۲-۲ تعداد هسته های پردازشی متصل به یک باس ۶۷
۴-۲-۳ ارائه ی توپولوژی پیشنهادی برای مش ۸×۸ ..... ۶۸
۴-۳ توصیف مسائل چند نمونه ای ۷۰
۴-۴ دریافت داده های تصادفی ۷۲
۴-۴-۱ فرمان های مربوط به تولید گراف مشخصه ی کاربرد ۷۳
۴-۴-۲ فرمان های مربوط به ایجاد جدول ۷۴
۴-۴-۳ فرمان های تولید خروجی ۷۴
۴-۵ تبدیل گراف مشخصه کاربرد به گراف ارتباط ۷۶
۴-۶ ایجاد گراف متوسط ۷۸
۴-۶-۱ روش ایجاد گراف متوسط ۷۸
۴-۷ دسته بندی ۸۰
۴-۷-۱ پیاده سازی تابع افراز ۸۱
۴-۷-۲ نحوه ی تقسیم بندی پهنای باند باس ۸۵
۴-۸ کیفیت سرویس ۸۶
۴-۹ روش مسیریابی ۸۷
۴-۱۰ نگاشت ۸۷
۴-۱۰-۱ مرحله ی اول : نگاشت هسته های پردازشی به باس ها ۸۸
۴-۱۰-۲ مرحله ی دوم : نگاشت هسته های پردازشی به خانه های شبکه روی تراشه ۸۹
۴-۱۰-۳ تابع هزینه ی کل ۹۰
۴-۱۱ حل یک مساله نمونه ۹۱
۴-۱۱-۱ دسته بندی ۹۱
۴-۱۱-۲ نگاشت ۹۴

۴-۱۲ نتیجه گیری ۹۶
فصل پنجم: بررسی الگوریتم و توپولوژی پیشنهادی
۵-۱ پارامترهای مشخص کننده ی نحوه عملکرد ۹۷
۵-۲ توپولوژی های متفاوت ۱۰۰
۵-۳ پهنای باند متفاوت ۱۰۲
۵-۴ نسبت تاخیر باس به تاخیر شبکه روی تراشه ۱۰۴
۵-۵ تغییر تعداد وظیفه های گراف مشخصه ی کاربرد ۱۰۵
۵-۶ مقایسه ی روش نگاشت پیشنهادی و روش NMAP ۱۰۷
۵-۷ وابستگی به کاربرد ۱۰۸
فصل ششم: نتیجه گیری و پیشنهادات
۶-۱ پیشنهادات ۱۱۲
مراجع ۱۱۳

چکیده

با پیشرفت تکنولوژی ساخت تراشه های نیمه هادی، امکان قرار گرفتن اجزای مختلف یک سیستم با صد ها پردازنده بر روی یک تراشه به وجود آمده است. شبکه ی روی تراشه، به عنوان یک شبکه ی ارتباطی موثر برای چنین سیستم هایی به کار می رود. تاخیر سیستم های روی تراشه، به میزان زیادی وابسته به زیربنای ارتباطی آن هاست. به همین دلیل، توجه به ساختار ارتباطی با تاخیر کم، اجتناب ناپذیر است. توپولوژی شبکه ی روی تراشه و چگونگی نگاشت هسته های پردازشی بر روی آن، دو فاکتور بسیار مهم هستند، که تاثیر زیادی در تاخیر شبکه های روی تراشه دارند. در این پایان نامه، یک توپولوژی جدید، با هدف کاهش تاخیر شبکه روی تراشه، ارائه شده است. این توپولوژی، ترکیبی از شبکه ی روی تراشه و باس است و توانسته مزیت های هر کدام را با هم داشته باشد. همچنین، روش جدیدی برای نگاشت هسته های پردازشی بر روی توپولوژی ترکیبی پیشنهادی، ارائه شده است. در این روش، ابتدا هسته های پردازشی که با هم ارتباط بیشتری دارند، به کمک تابع تصمیم گیری ارائه شده در این پایان نامه، دسته بندی می شوند. سپس، با توجه به توپولوژی و با هدف کاهش تاخیر شبکه روی تراشه، نگاشت می شوند. مقایسه ی نتایج با الگوریتم های متداول نگاشت بر روی توپولوژی مش، بهبود قابل توجهی را نشان می دهد.

کلمات کلیدی: ۱- شبکه روی تراشه ۲- توپولوژی و نگاشت ۳-سیستم روی تراشه

فصل اول

مقدمه

سیستم های تعبیه شده[1] امروزی از اجزای مختلفی تشکیل شده اند. با پیشرفت چشمگیر تکنولوژی ساخت سیلیکون، قرار دادن اجزای مختلف یک سیستم با صد ها پردازنده ی متفاوت بر روی یک تراشه امکان پذیر شده است. مفهوم شبکه روی تراشه[2] هنگامی پدید آمد که طراحان با سیستم های مجتمع خیلی بزرگ[3] و طراحی آنها بر روی یک تراشه کوچک مواجه شدند. این سیستم ها که تحت عنوان سیستم روی تراشه[4] نامیده می شوند، دارای مزیت های کاهش حجم، کاهش طول مسیر های انتقال، کاهش مصرف انرژی وغیره هستند[۱]. همچنین، با کوچک شدن مقیاس تکنولوژی ساخت، تاخیر سیم های ارتباطی در مقابل تاخیر گیت ها قابل ملاحظه شد و طراحی ارتباطات را به یک چالش مهم در تکنولوژی جدید تبدیل کرد[۲]. ساختار شبکه ی روی تراشه به عنوان یک راه حل مناسب برای طراحی ارتباطات ارائه شده است. این ساختار، ارتباط بین اجزای سیستم روی تراشه از قبیل هسته های پردازشی وحافظه را تشکیل می دهد. این ساختار از ارتباطات نقطه به نقطه که از طریق سوییچ (یا مسیریاب) به هم متصل شده اند، تشکیل شده است. به کارگیری شبکه ی روی تراشه به جای گذرگاه منجر به کاهش حجم، کاهش پیچیدگی طراحی سیستم، کاهش مصرف توان وافزایش سرعت می شود. همچنین، شبکه ی روی تراشه قابلیت طراحی پیمانه ای، استفاده از طرح های قبلی و توسعه پذیری سیستم را در اختیار طراحان قرار می دهد. این ویژگی های برجسته

---

[1] Embedded systems
[2] Network on Chip(NoC)
[3] VLSI
[4] System on Chip(SoC)

باعث شد که شبکه ی روی تراشه با استقبال زیادی روبرو شود و از سال ۲۰۰۳ به بعد سالانه تعداد زیادی از مقالات به این موضوع اختصاص داده شود. درحدود سال ۱۹۹۹ گروه های مختلف تحقیقاتی به بررسی روش های سیستماتیک برای طراحی اتصالات درون سیستم های روی یک تراشه پرداختند [۳]. این گروه ها به زودی به این نتیجه رسیدند که مشکلات باید در قسمت های مختلف از قسمت فیزیکی و معماری سیستم تا سیستم عامل و برنامه های کاربری حل شود. در نتیجه مفهوم گسترده شبکه روی یک تراشه ایجاد شد. در سال ۲۰۰۱، ویلیام دالی[۵] و برایان تولز[۶] عضو آزمایشگاه سیستم های کامپیوتری از دانشگاه استنفورد[۷] مفاهیم اولیه شبکه های روی یک تراشه را پایه گذاری کردند[۱]. بررسی های تانهونن[۸] دلایل برتری شبکه روی تراشه نسبت به معماری های دیگر را قابلیت طراحی پیمانه[۹]، توسعه پذیری، طراحی و تست آسان و قابلیت پیش بینی می داند[۴]. ساختار و اجزای معماری شبکه روی تراشه، توپولوژی ها و مفاهیم و موضوعات مربوط به آن به طور کامل و دقیق در مقالات[۲]، [۳]، [۵] و [۶] بیان شده است. در حال حاضر شبکه روی تراشه با مسائل زیادی که نیاز به مطالعه و تحقیق دارند، روبرو است. این مسائل از نقطه نظرطراحی در سه دسته انتخاب توپولوژی[۱۰]، انتخاب استراتژی مسیریابی[۱۱] و الگوریتم نگاشت متناسب[۱۲] قرار می گیرند. دسته اول به مباحث معماری NOC، اندازه بافرسوییچ ها وطول کانال های انتقال و . . . می پردازد. در دسته دوم الگوریتم های مسیریابی و مسائل مربوط به سوییچینگ مورد بررسی قرار می گیرد. دسته سوم مسائل مربوط به الگوریتم های زمان بندی و نگاشت هسته های پردازشی را در بر می گیرد[۷]، [۸]. مساله ی نگاشت هسته های پردازشی به چگونگی قرار دادن هسته های پردازشی روی خانه های شبکه روی تراشه می پردازد و اهداف آن تخصیص عادلانه ی پهنای باند، کاهش مصرف توان، افزایش سرعت سیستم وکاهش دمای سیستم است. مساله نگاشت با ظهور شبکه روی تراشه به عنوان یک مساله ی جدید ایجاد شد. برای اولین بار رادو مارکلسکو [۱۳] مساله ی نگاشت را مطرح کرد و یک راه حل شاخه وحرص کردن[۱۴] برای آن ارائه داد. وی نشان داد که نگاشت با استفاده از روش های الگوریتمیک در کاهش مصرف توان و افزایش سرعت سیستم تاثیر چشم گیری دارد[۹]. روش های حل مساله نگاشت را می توان به روش جستجوی کامل، روش جستجوی تصادفی، روش های اقتضایی [۱۵] و یا ترکیبی از این روش ها دسته بندی کرد. روش جستجوی کامل تمام جایگشت های ممکن هسته های پردازشی را روی شبکه

---

[5] William J. Dally
[6] Brian Towles
[7] Stanford
[8] Tenhunen
[9] Module
[10] Topology selection
[11] Routing Strategy Selection
[12] Application Mapping
[13] Radu Marculescue
[14] Branch & Bound
[15] heuristics

روی تراشه مورد بررسی قرار می دهد. روش تصادفی از روش هایی مانند الگوریتم ژنتیک، الگوریتم گرم کردن مرحله ای[16]، الگوریتم تابو[17] و برنامه ریزی خطی برای حل مساله نگاشت استفاده می کند. الگوریتم های اقتضایی از روش های حریصانه و نگاشت بر اساس مهمترین فاکتور بهره می گیرند[۱۰]. مسائل اولیه نگاشت با فرض داشتن یک نمونه و توپولوژی منظم و ارتباطات سیمی انجام می شدند. با این حال طی چند سال گذشته روش های نگاشت برای چند نمونه، نگاشت با توپولوژی نا منظم و ویژه ی کاربرد و نگاشت با استفاده از فیبر نوری نیز ارائه شده اند[۱۱]، [۱۲]، [۱۳]. روش های نگاشت تاثیر زیادی در کم کردن تاخیر شبکه، افزایش سرعت آن و مصرف توان دارند. مساله ی انتخاب توپولوژی به چگونگی چیدمان عناصر سیستم روی تراشه می پردازد. انتخاب یک توپولوژی مناسب در کاهش تاخیر و توان مصرفی تاثیر به سزایی دارد. توپولوژی ترکیبی باس و مش در میان دیگر توپولوژی ها توانسته است تاخیر کمتر و توان مصرفی کمتری را ارائه دهد.

در این پایان نامه، هدف، ارائه ی یک توپولوژی و روش نگاشت جدید برای کاربرد های چند نمونه ای است به گونه ای که تاخیر و توان مصرفی کل سیستم کاهش پیدا کند. کاربرد هایی که در آن چند برنامه متفاوت بر روی یک سیستم اجرا می شوند، کاربرد های چند نمونه ای نامیده می شوند. در سیستم های روی تراشه تاخیر و توان مصرفی از فا کتورهای مهم هستند. با توجه به اینکه کوچک شدن مقیاس ساخت نیمه هادی ها باعث شده است که تاخیر ارتباطات خود را بیشتر نشان دهند، طراحی زیر بنای ارتباطی با تاخیرکم، اهمیت زیادی پیدا کرده است. همچنین، داشتن تاخیر کم در دسترسی به حافظه های مشترک روی تراشه بسیار حیاتی است. در سیستم های تعبیه شده قابل حمل طول عمر سیستم متناسب با طول عمر باتری است و هر چه توان مصرفی کمتر باشد، طول عمر سیستم افزایش می یابد. همچنین، در سیستم های روی تراشه امروزی معمولا تراشه ها در چند لایه ایجاد می شوند و مصرف توان بیشتر موجب افزایش دمای تراشه و افت عملکرد آن می شود. توپولوژی ترکیبی به خاطر در برداشتن مزیت های باس و شبکه ی روی تراشه، تاخیر و توان کم به عنوان توپولوژی پایه در نظر گرفته شد. انتظار می رود در توپولوژی پیشنهادی معایب توپولوژی های ترکیبی قبلی حذف شود. همچنین، یک روش نگاشت برای کاربرد های چند نمونه ای و بر روی توپولوژی ترکیبی پیشنهادی ارائه شده است. در روش نگاشت پیشنهادی ابتدا هسته های پردازشی براساس میزان وابستگی، دسته بندی می شوند. برای انتخاب هسته های پردازشی که در یک دسته قرار می گیرند یک تابع هدف پیشنهاد می شود و به هر دسته از پردازنده ها یک باس اختصاص داده می شود. بعد از دسته بندی، هسته های پردازشی با استفاده از الگوریتم NMAP بر روی توپولوژی پیشنهادی نگاشت می شوند. نتایج نشان می دهند که توپولوژی و روش پیشنهادی توانسته است تا حد زیادی تاخیر و مصرف توان را بهبود ببخشد.

[16] Simulated Anealing
[17] Tabu Algorithm

ساختار پایان نامه به ترتیب زیر است :

- در فصل دوم، ابتدا تاریخچه ای از چگونگی پیدایش شبکه روی تراشه بیان می شود. در ادامه چالش هایی که با کوچک شدن تکنولوژی ساخت ایجاد می شود، بیان می شود. شبکه روی تراشه به عنوان راه حلی برای پاسخ به این چالش ها مورد ارزیابی قرار می گیرد. سپس، اهمیت شبکه روی تراشه و برتری آن نسبت به ارتباطات گذرگاه بررسی می شود. هم چنین، اجزای شبکه روی تراشه و مفاهیم مربوط به آن بیان می شود.
- در فصل سوم، روند طراحی شبکه روی تراشه بیان می شود. با توجه به اینکه در گام های طراحی شبکه روی تراشه، از تئوری گراف ها بهره گرفته می شود، مفاهیم اولیه طراحی شبکه ی روی تراشه در قالب مدل های گرافی توصیف می شوند، به کمک این مدل ها، توصیفی از مساله مورد بررسی در این پایان نامه ارائه می شود.
- در فصل چهارم، نکاتی که در ارائه توپولوژی و روش نگاشت پیشنهادی مورد توجه قرار گرفته است، بیان می شوند. به کمک این نکات چند توپولوژی ترکیبی باس و شبکه روی تراشه، مطرح می شوند و از میان آن ها، به وسیله تعریف معیار مقایسه، توپولوژی که در بین دیگر توپولوژی ها، منجر به ارتباطاتی با تاخیر کمتر می شود، برگزیده می شود. سپس روشی برای نگاشت هسته های پردازشی بر روی توپولوژی پیشنهادی ارائه می شود. در این روش ابتدا پردازنده ها دسته بندی و سپس نگاشت می گردند.
- در فصل پنجم، توپولوژی و روش نگاشت پیشنهاد ی مورد ارزیابی قرار می گیرند و به کمک نمودار ها نحوه ی عملکرد روش پیشنهادی ارزیابی می شود.
- فصل ششم، به تحلیل نتایج و ارائه پیشنهادات اختصاص دارد.

فصل دوم
شبکه روی تراشه

در این فصل، هدف، آشنایی با ساختار و مفاهیم شبکه روی تراشه است. به این منظور تاریخچه ی پیدایش شبکه روی تراشه بیان می شود و همچنین، دلیل پذیرش و برتری شبکه روی تراشه به عنوان ساختار ارتباطی سیستم های روی تراشه مورد بررسی قرار می گیرد، سپس ساختار کلی از معماری شبکه روی تراشه بیان می شود و در ادامه جزییات و مفاهیم مربوط به این ساختار بیان می شود. برای درک واقعی از شبکه روی تراشه در انتهای این فصل نمونه هایی از معماری شبکه روی تراشه آورده می شود.

۲-۱ تاریچه پیدایش شبکه روی یک تراشه

درطی سال ۱۹۹۰ به بعد تعداد بیشتری از هسته های پردازشی و مولفه های قابل استفاده مجدد در یک تراشه سیلیکون در کنار هم قرار گرفتند، که تحت واژه سیستم روی یک تراشه [18] شناخته می شدند. استاندارد کردن واسط های مولفه ها و تعیین صحت عملکرد کل سیستم با توجه به مشخصه های فیزیکی و عملکردی، از جمله مهمترین مشکلاتی بود که در این عصر با آن مواجه بودند. باس ها و اتصالات نقطه به نقطه عناصر اصلی ایجاد ارتباط بین مولفه ها بودند. باس ها طرفداران بسیاری داشتند، زیرا ارتباطاتی با کارایی بالا را فراهم می کردند و در مقابل می توانستند در بین چند مولفه به اشتراک گذاشته شوند و با هزینه کم، بازدهی قابل قبولی را در اختیار طراحان بگذارند[۳].

[18] System on Chip

با پیشرفت تکنولوژی ساخت سیلیکون، مشکلات متعددی پیرامون استفاده از باس ایجاد شد. باس ها می توانستند به طور موثر بین ۱۰-۳ مولفه به اشتراک گذاشته شوند، اما قابلیت توسعه پذیری به تعداد بالاتری از مولفه ها را نداشتند. همچنین، باس ها از دید مولفه ها، رفتاری غیر قابل پیش بینی داشتند (به دلیل اینکه بین تعداد زیادی از مولفه ها به اشتراک گذاشته می شدند). مشکل دیگر باس ها از فیزیک و تکنولوژی زیر میکرون [۱۹] ناشی می شد. سیم های بلند و عمومی به علت کارایی کم و غیر قابل پیشبینی، ناخوشایند شدند. مشکل دیگر باس ها از دید برنامه های کاربری ناشی می شود. طراحی و تعیین صحت عملکرد ارتباطات بین وظیفه [۲۰] ها در یک برنامه کاربری یک مساله سخت است. تعیین صحت عملکرد یک برنامه کاربری و تعیین منابع مورد نیاز آن به خاطر ماهیت غیر قابل پیش بینی باس کار مشکلی است. علاوه بر آن هر برنامه کاربری ساختار های ارتباطی منحصر به فردی دارد، بنابراین، استفاده مجدد از طرح های قبلی با مشکل مواجه می شد.

در نتیجه درحدود سال ۱۹۹۹ گروه های مختلف تحقیقاتی به بررسی روش های سیستماتیک برای طراحی اتصالات درون سیستم های روی یک تراشه پرداختند[۳]. این گروه ها به زودی به این نتیجه رسیدند که مشکلات باید در قسمت های مختلف از قسمت فیزیکی و معماری سیستم تا سیستم عامل و برنامه های کاربری حل شود. در نتیجه مفهوم گسترده شبکه روی یک تراشه ایجاد شد. در سال ۲۰۰۱، ویلیام دالی و برایان تولز[۲۱] عضو آزمایشگاه سیستم های کامپیوتری از دانشگاه استنفورد[۲۲] مفاهیم اولیه شبکه های روی یک تراشه را پایه گذاری کردند [۱]. شبکه روی یک تراشه امروزه شامل مفاهیم زیربنای سخت افزاری ارتباطات، سرویس های سیستم عامل، متدولوژی های طراحی و ابزارهای نگاشت برنامه های کاربردی به شبکه روی یک تراشه می شود.

۲-۲ شبکه روی یک تراشه

در این قسمت دلایل انتخاب شبکه روی تراشه به عنوان یک زیربنای موثر ارتباطی بیان می شود. ابتدا نکاتی که در طراحی یک سیستم مهم است، ذکر می شود، سپس گرایش ها وچالش هایی که در طراحی های امروزی وجود دارد، بررسی می شود و در آخر راه کار ها ی شبکه روی تراشه، برای حل این گرایش ها وچالش ها آورده می شود.

۲-۲-۱ نکات مهم در طراحی سیستم های بزرگ

در طراحی یک سیستم، نکته ی مهم برای بالا بردن سودمندی طراحی این است که، هزینه ی اضافه کردن یک قسمت جدید به سیستم، تابعی خطی از اندازه ی قسمت جدید باشد و وابسته به اندازه ی کل سیستم نباشد. در این صورت می توان بخش بزرگی از بلاک های طراحی قبلی را دوباره استفاده کرد و هزینه ی طراحی، محدود به

[19] Deep submicron technology
[20] task
[21] William J. Dally & Brian Towles
[22] Stanford

طراحی قسمت جدید می شود. علاوه بر آن این ویژگی، یک شرط لازم برای تولید یک معماری، متودولوژی و یا یک زیر سیستم نیرومند است، که در مقابل تغییرات نسل های مختلف تکنولوژی تحمل پذیر است. موضوع اصلی که در این بخش مورد بررسی قرار می گیرد، این است که، آیا شبکه روی تراشه پتانسیل ایجاد چنین سکوی[۲۳] نیرومند و تحمل پذیری را داراست؟ در صورت مثبت بودن جواب این سوال، شبکه روی تراشه می تواند تحول بزرگی در معماری سیستم- روی- تراشه و فرایند های طراحی ایجاد کند، در غیر این صورت شبکه روی تراشه مانند ده ها معماری موجود برای طراحی سیستم های تعبیه شده است[۴].

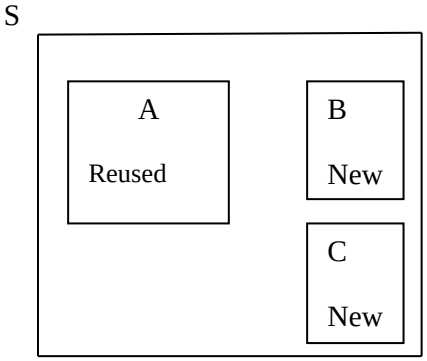
خاصیت ترکیب پذیری دلخواه: فرض کنید، مجموعه ای از مولفه ها و عملگر های ترکیبی که این مولفه ها را به هم مرتبط می کنند و یک مولفه ی جدید را ایجاد می کنند، داریم. مولفه ها و عملگر ها دارای ویژگی ترکیب پذیری دلخواه هستند، به شرطی که هر مجموعه از مولفه ها مانند A، بتواند با مولفه ی دیگر گسترش پیدا کند و در عملکرد آن تغییری ایجاد نشود.

برای مثال گیت های منطقی NAND,NOR ,INV دارای چنین خاصیتی هستند. شبکه جدید گیت ها می تواند به شبکه ی گیت های موجود اضافه شود، بدون این که در عملکرد قسمت های قبلی مدار، تغییری ایجاد کند. این ویژگی مبنای توانایی ما در ایجاد طرح هایی شامل میلیون ها گیت و استفاده دوباره از بلاک های بزرگ محسوب می شود. در مقابل عدم وجود این ویژگی، هزینه ی بالایی در طراحی را تحمیل می کند. به عنوان مثال عدم وجود خاصیت ترکیب پذیری دلخواه، مسئله ی توصیف سیستم را به یک مسئله ی دشوار تبدیل کرده است، زیرا تغییری کوچک در توصیف یک قسمت سیستم، می تواند تاثیرات غیرقابل پیش بینی در قسمت های دیگر سیستم داشته باشد و هزینه ی طراحی بالایی را تحمیل کند. یکی از نتایج خاصیت ترکیب پذیری دلخواه، خاصیت هزینه ی خطی است.

خاصیت هزینه ی خطی : فرض کنید، مجموعه ای از مولفه ها و عملگر های ترکیبی که این مولفه ها را به هم مرتبط می کنند و یک مولفه ی جدید را ایجاد می کنند، داریم. یک فرایند طراحی که در آن یک سیستم جدید با استفاده از عملگر های ترکیبی و مولفه ها ایجاد می شود دارای خاصیت هزینه ی خطی است، به شرطی که هزینه ی ایجاد سیستم خطی S که از ترکیب n مولفه مانند $A_1, A_2, \ldots, A_n$ ایجاد می شود، تابعی خطی از تعداد مولفه ها(n) باشد (Ieffort(n)) و وابسته به اندازه مولفه ها نباشد. بنابراین، هزینه ی کلی طراحی برای سیستم S برابر است با :

$$\text{Deffort(s)} = \text{Deffort}(A_1) + \text{Deffort}(A_2) + \ldots\ldots + \text{Deffort}(A_n) + \text{Ieffort}(n) \quad (۱\text{-}۲)$$

[23] platform



شکل ۲-۱ مثالی از خاصیت ترکیب پذیری و خاصیت هزینه ی خطی.

برای مثال با توجه به دو ویژگی که گفته شد، هزینه ی ایجاد سیستم S در شکل ۲-۱ برابر است با هزینه ی ایجاد دو جزء جدید B ,C، به اضافه ی هزینه ی متصل کردن مولفه های A,B,C به یک دیگر. به علت اینکه از مولفه A دوباره در طرح استفاده می شود، هزینه طراحی آن حذف می شود. بنابراین، هزینه ی طراحی سیستم S برابر است با

$$Deffort(s) = Deffort(B) + Deffort(C) + Ieffort \quad (3)$$

این ویژگی به طور ضمنی حاوی این مفهوم است که، پیچیدگی متصل کردن چند مجموعه به اندازه ی آن ها وابسته نیست. بدیهی است که در عمل پیچیدگی اتصال چند مجموعه، تا حدی وابسته به اندازه ی آن هاست و هرچه بتوانیم این وابستگی را کمتر کنیم، توانایی ما در ایجاد سیستم های بزرگتر، بیشتر می شود. محققان عقیده دارند که سیستم های مبتنی بر شبکه روی تراشه، پتانسیل فراهم کردن ویژگی های ترکیب پذیری دلخواه و هزینه ی خطی را در سطوح بالایی دارا هستند[۳]. در بخش بعدی گرایش ها و چالش هایی که منجر به ایجاد شبکه روی تراشه و معماری های مانند آن شد، را مورد مطالعه قرار می دهیم.

۲-۲-۲ گرایش ها و چالش ها

تکنولوژی ساخت مدارهای مجتمع، امکان ایجاد تراشه هایی با بیلیون ها ترانزیستور را، فراهم کرده است. چگونگی سازمان دهی این تراشه ها با چالش هایی روبرو است، که پیدا کردن راه حلی برای آن ها، می تواند، تحولی شگرف در طراحی و معماری مدارهای مجتمع ایجاد کند. در ادامه به بررسی تعدادی از این چالش ها می پردازیم.

ارتباطات در مقابل محاسبات: در حالی که سیم های روی تراشه نسبت به سیم ها ی عادی ارزان تر هستند، اما ارتباطات روی یک تراشه به علت توان مصرفی و سرعت بالا، گران تر از ارتباطات بین تراشه ها است. با کوچک شدن مقیاس ساخت، مقاومت سیم ها در هر میلیمتر افزایش می یابد، در حالی که ظرفیت سیم ها تغییر چندانی نمی