مروری بر معماری ها و فناوری های پیشرفته FPGA

چکیده

آرایه گیت قابل برنامه نویسی میدانی (FPGA)، یک پلت فرم مداری کارآمد و قابل پیکربندی مجدد است و در دهه گذشته، به یک دستگاه ریزتراشه پردازش سیگنال هسته ای برای سیستم های دیجیتال تبدیل شده است. با توسعه سریع تکنولوژی نیمه هادی، یکپارچه سازی سازی عملکرد و سیستم ادوات FPGA به طور قابل توجهی پیشرفت نموده است، و در عین حال چالش های جدیدی بوجود آمده است. طراحی معماری FPGA به منظور تکامل و برآورده سازی این چالش ها مورد نیاز است و در عین حال باید از چگالی فزاینده ریزتراشه بیش از هر زمان دیگری استفاده نمود. این مطالعه به بررسی توسعه های اخیر معماری FPGA پیشرفته، از جمله بهبود فن آوری های برنامه نویسی، بلوک های منطقی، اتصالات، و منابع تعبیه شده می پردازد. علاوه بر این، برخی از مسائل مهم طراحی در حال ظهور از معماری FPGA، مانند FPGA های مبتنی بر حافظه جدید و FPGA سه بعدی نیز به منظور ارائه چشم اندازی برای توسعه آینده FPGA ارائه می شود.

کلمات کلیدی: آرایه گیت قابل برنامه نویسی میدانی (FPGA)؛ معماری ریزتراشه؛ وسیله منطقی قابل برنامه نویسی؛ سیستم بر روی تراشه (SOC)

شاخصCLC : TN47

DOI: 10.1007/s11767-014-4090-x

1. مقدمه

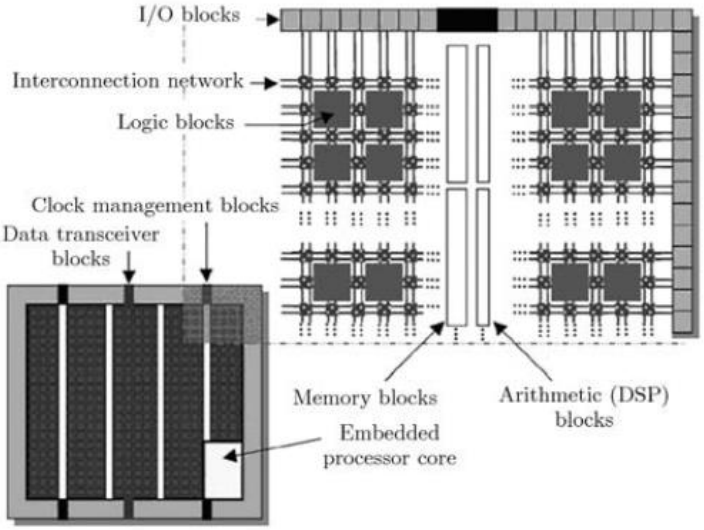
آرایه های گیت قابل برنامه نویسی میدانی (FPGA ها)، ادوات سیلیکونی پیش ساخته هستند که می توانند به لحاظ الکتریکی برای اجرای تقریبا هر نوع مدار و یا سیستم دیجیتال برنامه نویسی شوند. همانطور که در شکل 1 نشان داده شده است، ساختار اساسی FPGA شامل یک دریا بلوک های منطقی (LBS)، یک شبکه اتصال داخلی، و بلوک های I / O قابل پیکربندی می شود. به علت یکپارچگی در سطح بسیار بالا، ادوات FPGA اخیر نیز شامل بلوک های حافظه، بلوک های DSP سیم بندی شده، بلوک های مدیریت ساعت، و بلوک های فرستنده-گیرنده داده ها با سرعت بالا می شوند که تماماً به صورت یکپارچه سازی [1-3] تعبیه شده اند.

بلوک های منطقی منابع اصلی پردازش دیجیتال هستند و بسته به عملکرد مورد نیاز برای پیاده سازی، هر یک از آنها برای انجام ترکیبی و عملیات پی در پی پیکربندی می شوند.

برای عملیات های ترکیبی، مجموعه ای از جداول مراجعه (LUTS) به عنوان تولیدکننده های تابع منطقی مستقل به کار گرفته می شوند و برای عملیات های پی در پی، مجموعه ای از D-فلیپ فلاپ ها گنجانده می شوند. علاوه بر این، برخی از اشکال تکامل یافته LB برای حمایت از توابع اضافی، مانند ذخیره سازی محلی (حافظه RAM توزیع)، شیفت رجیستر (SR)، عملیات مالتی پلکسر ، و عملیات های جمع کننده / کم کننده بهینه سازی می شوند. شبکه اتصال توسط کاربر به منظور پیوند بسیاری از LBها در صورت لزوم قابل برنامه نویسی می باشد [4-7].

به منظور بهینه سازی عملکرد FPGA، بلوک های DSP متصل شده گنجانده می شوند، که عملیات های ریاضی پیچیده را میسر می سازند. حافظه های داخلی، مانند RAM، ROM، فلشRAM ، و شیفت رجیسترها به صورت اختیاری برای افزایش سرعت پردازش و طراحی ساده سطح برد سیستم مجتمع می شوند. بلوک های مدیریت ساعت مجتمع شده برای ارائه هممسیر سازی سیستم قابل ترکیب استفاده می شوند. آنها معمولا بر اساس حلقه های-قفل-فاز (PLLs) هستند که از ویژگی های از قبیل ضرب و تقسیم فرکانسی، جبران تاخیر انتشار و اصلاح تغییر فاز پشتیبانی می نمایند. ادوات FPGA فعلی نیز شامل بلوک های فرستنده و گیرنده داده سرعت بالا می شوند که به طور کلی از بافر های ارسال و دریافت تشکیل شده اند. پروتکل های ارتباطی مختلف شده داده ها، از جمله USB،

اترنت، CAN، PCI، SPI، I2C، و غیره پشتیبانی می شوند. علاوه بر این، معماری های FPGA پیچیده شامل ریزپردازنده تعبیه شده و لوازم جانبی برای ویژگی های پشتیبانی SOC اضافی [8-10] می شوند.



شکل. 1 ساختار کلی یک FPGA [11]

به منظور ارائه تراکم مجتمع سازی بالا، سرعت بالا و مصرف کم انرژی، FPGA ها تحت پیشرفت قابل توجهی از نظر تکنولوژی فرآیند نیمه هادی پیشرفته قرار گرفته اند. ادوات اخیر FPGA تجاری، مانند Arria 10 و Xilinx 'Virtex UltraScale به پردازی در حد 20 نانومتر [12،13] رسیده اند. علاوه بر این، هدف Xilinx ارائه Virtex UltraScale تمام ادوات قابل برنامه نویسی ساخته شده بر روی تکنولوژی فرآیند 16 نانومتری FINFET TSMC است و شرکت Altera و اینتل به طور مشترک اعلام کرده اند که نسل بعدی محصولاتی با بالاترین FPGA Altera با عملکرد بالا (Stratix 10) با استفاده از فن آوری ترانزیستوری Tri-Gate اینتل سه بعدی 14 نانومتری اینتل تولید خواهد شد [14،15]. توسعه FPGA تا حد زیادی توسط پیاده سازی فرآیندهای پیشرفته افزایش یافته است. بنابراین سطوح بی سابقه ای از عملکرد، یکپارچه سازی سازی سیستم و پهنای باند را می توان در نسل های بعدی FPGA انتظار داشت.

هدف از این مطالعه، ارائه آخرین وضعیت معماری FPGA و پیش بینی روند آینده در پیشرفت طراحی FPGA است. بقیه این مقاله به شرح زیر است. بخش دوم روش های اصلی برای فن آوری های برنامه نویسی FPGA را معرفی می

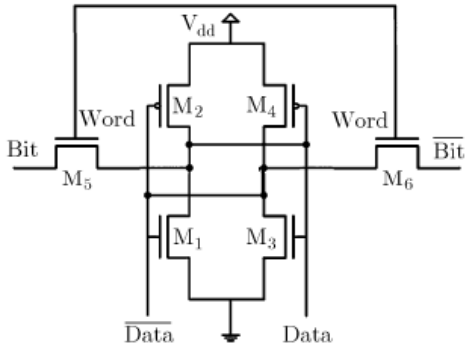
کند. بخش سوم به مسائل طراحی با توجه به معماری بلوک منطقی می پردازد، در حالی که معماری مسیریابی برنامه نویسی شده در بخش چهارم توصیف می شود. بخش پنجم آرایش منابع سخت افزاری تعبیه شده در FPGA ها را مورد بحث قرار می دهد و در مورد همگرایی به سمت FPGA های سیستم روی تراشه (SOC) توضیح می دهد. بخش ششم برای کشف روندهای تکنولوژی آینده FPGA ها تلاش می کند و نتیجه گیری مقاله در بخش هفتم ارائه شده است.

2. فن آوری های برنامه نویسی

یک FPGA، با استفاده از سوئیچ های قابل برنامه نویسی الکتریکی برنامه نویسی می شود. خصوصیات این سوئیچ های قابل برنامه نویسی، از قبیل اندازه، بر روی مقاومت و خازن، تاثیر قابل توجهی بر روی معماری های منطقی برنامه نویسی دارند. روش های مورد استفاده گسترده در FPGA های مدرن شامل حافظه استاتیک با دسترسی تصادفی (SRAM)، حافظه های فلش، و ضد فیوز می شوند. از این روش ها، FPGA های مبتنی بر SRAM ها بیشتر مورد استفاده قرار می گیرند و عمدتا به دلیل مقیاس پذیری آن با تکنولوژی فرآیند ، موقعیت غالب را در بازار MOS دارند. در این بخش، تمام این روش های فن آوری های برنامه نویسی را مرور خواهیم کرد، و مزایا و معایب آنها را نسبت به درک کامل از فن آوری های برنامه نویسی مقایسه می نماییم.

1. برنامه نویسی مبتنی بر SRAM

اساس و مبنای تکنولوژی برنامه نویسی SRAM، سلول حافظه استاتیک، در شکل 2 نشان داده شده است. دسترسی به سلول توسط خط کلمه فعال می شود که دو ترانزیستور دسترسی M5 و M6 را برای هر دو عملیات خواندن و نوشتن کنترل می کند. اگر خط کلمه اظهار نشود، ترانزیستورهای دسترسی M5 و M6 ارتباط سلول را از خطوط بیت قطع می نمایند و دو اینورتر تزویج شده-متقابل تشکیل شده توسط M4 ~ M1 به تقویت یکدیگر ادامه خواهند داد تا زمانی که آنها به منبع متصل شوند [متصل 3].

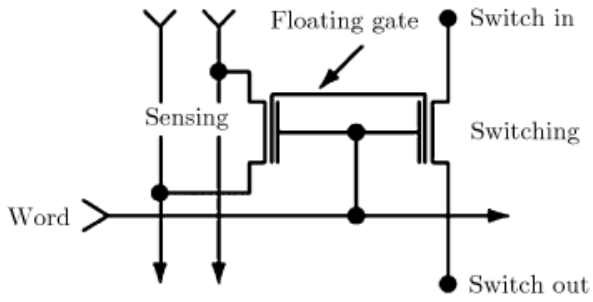


شکل. 2 سلول حافظه استاتیک

FPGA های مبتنی بر SRAM از SRAM به طور معمول از طریق استفاده از LUTs و مالتی پلکسرها، برای ارائه قابلیت پیکربندی برای توابع مسیریابی و محاسباتی استفاده می شوند. دو مزیت اصلی مرتبط، یعنی قابلیت برنامه نویسی دوباره توسط طراحی و سازگاری با تکنولوژی فرآیند استاندارد CMOS وجود دارد. در نتیجه، FPGA های مبتنی بر SRAM می توانند با آخرین فن آوری CMOS در دسترس پیاده سازی شوند و بنابراین، از افزایش سطح یکپارچگی، سرعت بالاتر و مصرف توان دینامیک کمتر با هندسه حداقل کوچکتر بهره مند هستند [16].

2. برنامه نویسی مبتنی بر حافظه فلش / E2PROM

در مورد فن آوری حافظه فلش، پیکربندی بر اساس اتصالات فلش است که زمانی که دستگاه دارای توان پایین است، حالت پیکربندی را حفظ می کند. هر اتصال شامل دو ترانزیستور می شود که دارای یک گیت شناور مشترک هستند و اطلاعات بیتی برنامه نویسی را ذخیره می نماید، همانطور که شکل 3 نشان می دهد. رسانایی ترانزیستور دسترسی را می توان با تزریق بار بر روی گیت شناور کنترل نمود. از آنجا که گیت شناور از لحاظ الکتریکی توسط لایه عایق آن جدا شده است، هر الکترون قرار داده شده بر روی آن در آنجا به دام می افتد و تحت شرایط عادی، برای سال های طولانی تخلیه نخواهد شد.



شکل. 3 ترانزیستور با گیت شناور [17]

بر خلاف SRAM مبتنی بر CMOS، عدم نوسان یکی از مهمترین مزایای تکنولوژی برنامه نویسی مبتنی بر حافظه فلش است. در نتیجه، FPGA های مبتنی بر حافظه فلش نباید از منابع خارجی برای ذخیره و داده های پیکربندی بار استفاده نمایند. علاوه بر این، زمانی که تمام داده های پیکربندی و مسیریابی، حالات خود را در هنگام خاموش بودن دستگاه حفظ کنند، یک دستگاه مبتنی بر فلش می تواند بلافاصله پس از تغذیه توان به جای انتظار برای پیکربندی دوباره عمل نماید. روش فلش ار لحاظ سطح کاری کارآمدتر از فن آوری مبتنی بر SRAM است و برای پیاده سازی ذخیره بیتی قابل برنامه نویسی نیاز به 5 یا 6 ترانزیستور دارد.

معایب اصلی ادوات مبتنی بر فلش، تعداد محدود اجراهای برنامه نویسی مجدد و نیاز به یک فرایند CMOS غیر استاندارد هستند. ادوات کنونی مانند Microsemi IGLOO2 تنها برای 500 ~ 1000 چرخه برنامه نویسی [18] رتبه بندی می شود.

3. برنامه نویسی زمانی بر اساس ضد فیوز

یک جایگزین برای فن آوری های مبتنی بر گیت شناور و SRAM، تکنولوژی برنامه نویسی ضد فیوز است. این تکنولوژی بر اساس ساختارهایی است که مقاومت بسیار بالا را در شرایط عادی نشان می دهند اما می توانند برای ایجاد یک لینک با مقاومت کم "دمیده" (در واقع، متصل شوند) شوند. در مقابل فن آوری های برنامه نویسی مبتنی

بر گیت شناور و SRAM، این لینک دائمی است. بنابراین، FPGA ها مبتنی بر تکنولوژی برنامه نویسی ضد فیوز می توانند فقط یک بار برنامه نویسی شوند.

مزیت اصلی تکنولوژی برنامه نویسی ضد فیوز، رد پای کوچک آن [16] است. با ضد فیوزهای فلز به فلز (شکل 4)، هیچ مساحت سیلیکونی برای ایجاد ارتباط مورد نیاز است که مساحت مورد نیاز برای برنامه نویسی را کاهش می دهد. به همین دلیل، ضد فیوزها دارای مقاومتهای و خازن های پارازیتی کمتر از فن آوری های برنامه نویسی دیگر هستند. علاوه بر این، همانطور که برنامه نویسی FPGA های مبتنی بر ضد فیوز تنها یک بار باید اجرا شوند، امنیت برای جریان بیت توصیف سخت افزاری دانلود شده به FPGA ها تا حد زیادی بهبود یافته اند.

شکل. 4 عناصر اتصال ضد فیوز (فلز به فلز) [19]

با این حال، برخی معایب وجود دارند. به طور خاص، چون FPGA های مبتنی بر ضد فیوز- به یک فرایند CMOS غیر استاندارد نیاز دارند، آنها در مقایسه با FPGA های مبتنی بر SRAM، به طور معمول چندین نسل پشت فرآیندهای تولید هستند (جدول 1 را در زیر مشاهده کنید). علاوه بر این، مکانیسم اساسی برنامه نویسی، که شامل تغییرات قابل توجهی برای خواص مواد در فیوز می شوند، زمانی که فرآیندهای ساخت IC های جدید فازبندی می شوند، منجر به چالش هایی در مقیاس بندی می شوند. عدم توانایی برای برنامه نویسی مجدد نیز دامنه کاربردها را محدود می کند. بر خلاف دیگر فن آوری ها، برنامه نویسی درون-سیستم که قبل از نصب بر روی محصول نهایی، به برنامه نویسی سیستم های خاص مورد استفاده برای برنامه نویسی یک دستگاه نیاز دارند، امکان پذیر نمی باشد. در

نهایت، برنامه نویسی یکباره ضد فیوزها، آزمایش های تولید برای تشخیص تمام خطاهای ممکن [16] را غیر ممکن می سازد.

جدول. 1 مقایسه فن آوری های برنامه نویسی مختلف

Parameters	Programming type		
	SRAM	Flash	Anti-fuse
Nonvolatile	×	√	√
Reprogrammable	√	√	×
Cell components	5 or 6 transistors	1 or 2 transistors	Metal-to-metal
Read/Write speed	1 ns/1 ns	100 ns/200 ns[20]	-
State-of-the-art process	20 nm	65 nm[18]	130 nm
Endurance (Cycles)	Infinite	500–1000[18]	1
Programming yield	100%	100%	>90%

4. خلاصه

سه فن آوری برنامه نویسی بازنگری شده در این بخش باید حوزه کاربردهای مختلف را در بازارهای FPGA مدرن داشته باشند. تکنولوژی برنامه نویسی مبتنی بر SRAM به طور گسترده به دلیل سازگاری آن ببرایه فرآیند استاندارد CMOS استفاده می شود. FPGA های مبتنی بر حافظه فلش به طور عمده در برخی کاربردهای خاص استفاده می شوند که در آن هر دو تکنولوژی غیر فرار و قابل برنامه نویسی مجدد مورد نیاز هستند. با توجه به بالاترین قابلیت اطمینان در هزینه های انعطاف پذیری، FPGA های ضد فیوز عمدتاً در سیستم های حمل و نقل هوایی و هوا فضا استفاده می شوند. جدول 1، خلاصه جوانب مثبت و منفی در میان فن آوری های برنامه نویسی مختلف را نشان داده است.

3. معماری بلوک منطقی

FPGAها از بلوک های منطقی برای پیاده سازی توابع منطقی، مسیریابی قابل برنامه نویسی برای اتصال این توابع و بلوک های I / O برای اتصالات خارج از تراشه تشکیل شده اند. معماری بلوک منطقی بسیار مهم است، زیرا با توجه به طراحی FPGA ها برای عملکرد بهینه و تراکم منطقی، در قلب این کار نهفته است [21-23]. در این بخش، ما

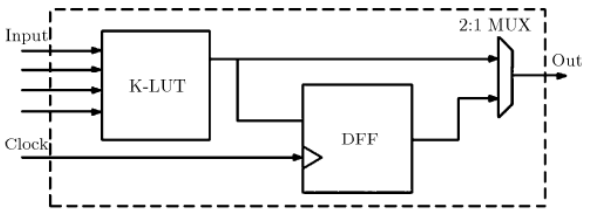
مسائل اساسی و سبک سنگین کردن در طراحی معماری بلوک منطقی را با بازدید از تکامل معماری های FPGA تجاری مورد بحث قرار می دهیم.

1. اصول طراحی

هدف از یک بلوک منطقی در FPGA، ارائه عناصر محاسبات و ذخیره سازی اساسی مورد استفاده در سیستم های دیجیتال است. در این بخش، سه بلوک های منطقی که عمدتاً اشاره می شوند، و به ترتیب در LUT، مالتی پلکسر، و And-Inverter Cone (AIC) استفاده می شوند، بررسی خواهد شد.

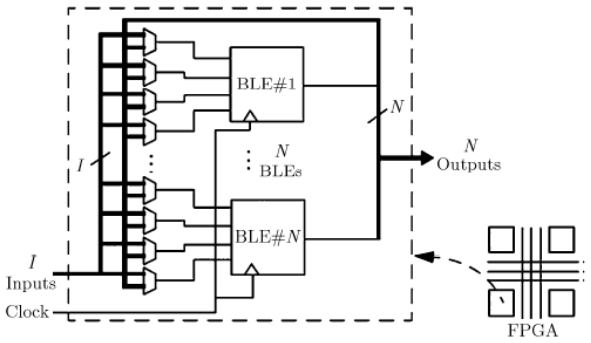
LUT (1)

LUT به عنوان آرایه ای از حافظه ها در نظر گرفته می شود که تنها 1 بیت خروجی دارند. معماری بلوک منطقی با یک LUT و DFF در شکل 5 نشان داده شده است. در اغلب موارد، یک جدول درستی برای یک تابع منطقی K-ورودی در SRAM ($K \times 12$) ذخیره می شود. خطوط آدرس SRAM به عنوان ورودی ها عمل می کنند و خروجی SRAM، مقدار تابع منطقی را فراهم می کند. با این حال، قابلیت بالای LUT آزاد نیست. آنها تقریبا بزرگ و آهسته هستند، زیرا مساحت آنها به طور نمایی رشد می کند و تاخیر یه طور خطی با تعداد ورودی رشد می کند. همچنین، تعداد خروجی ها ذاتا تنها یک است و انعطاف پذیری آنها را محدود می نمایند [24،25].



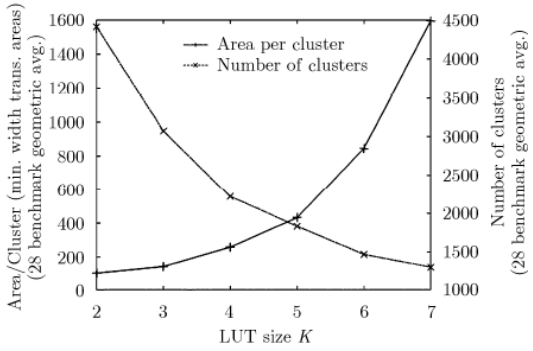
شکل. 5 معماری LB با یک LUT و DFF

FPGA هاص تجاری به طور عمده استفاده شده همیشه بر اساس خوشه ها، از جمله Altera Stratix، Cyclone ، و سری Xilinx Virtex [35-26] هستند. خوشه، گروهی از عناصر منطقی پایه (BLEها) است که به طور کامل توسط یک نوار متقابل مبتنی بر MUX [36] متصل می شوند، همانطور که در شکل 6 نشان داده شده است.

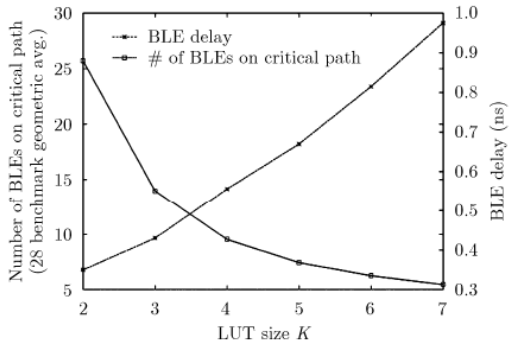


شکل. 6 ساختار خوشه منطقی [36]

در اغلب FPGA های صنعتی، ذره ذرعه بودن بلوک منطقی با استفاده از LUTS و فلیپ فلاپ قابل تنظیم است. اندازه LUT (K، تعداد ورودی) و اندازه خوشه (N، تعداد LUTS در هر خوشه) اثر مهمی در مساحت و سرعت عملکرد FPGA ها دارد. مرجع [37] برای اولین بار، اثر K و N بر معماری FPGA را مورد کاوش قرار داده است، همانطور که در شکل 7 نشان داده است.



(a) Number of clusters and cluster area versus K [37]



(b) Number of BLEs on critical path and BLE delay versus K [37]

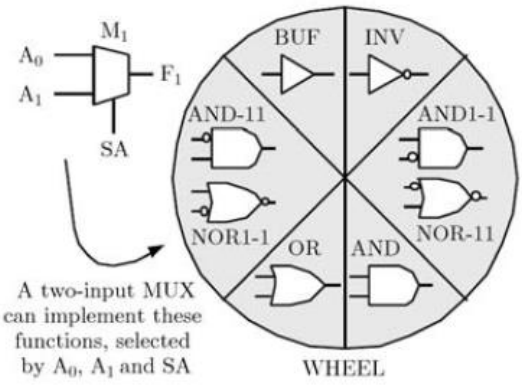
شکل. 7 تصویر اثر ویژگی های LUT و BLE

شکل 7 (الف) اثر اندازه LUT را روی تعداد خوشه ها و منطقه خوشه را نشان می دهد و شکل 7 (ب) سبک سنگین کردن بین تاخیر BLE و تعداد BLEs را در مسیر بحرانی نشان می دهد. این قابل درک است که با افزایش LUT و اندازه خوشه، تعداد کل بلوک های منطقی مورد نیاز برای اجرای یک تابع معین کاهش می یابد، و تعداد چنین بلوک در مسیر بحرانی نیز کاهش می یابد. با این حال، اندازه و تاخیر بلوک منطقی با K و N افزایش می یابد. علاوه بر این، مساحت اختصاص داده شده به مسیر یابی خارج از بلوک به عنوان یک تابعی از K و N تغییر خواهد کرد، و این یک اثر قوی بر نتایج خواهد داشت. انتخاب بلوک منطق دانه دانه، که بهترین محصول تاخیر-مساحت را تولید می کند، در جایی بین این دو کران قرار می گیرد [38-40].

MUX (2)

توابعی که می توانند توسط یک MUX دو ورودی تک پیاده سازی شوند، در شکل 8 نشان داده شده است. در FPGA های مبتنی بر MUX، MUXs با اتصال هر یک از ورودی های خود به یک ثابت یا به یک سیگنال، دو ورودی برای پیاده سازی توابع مختلف منطقی استفاده می شوند. با استفاده از این روش ها، یک بلوک منطقی توانمند در پیاده سازی تعداد زیادی از توابع را می توان با دسته بندی تعدادی از MUXs و گیت های منطقی پایه ساخت.

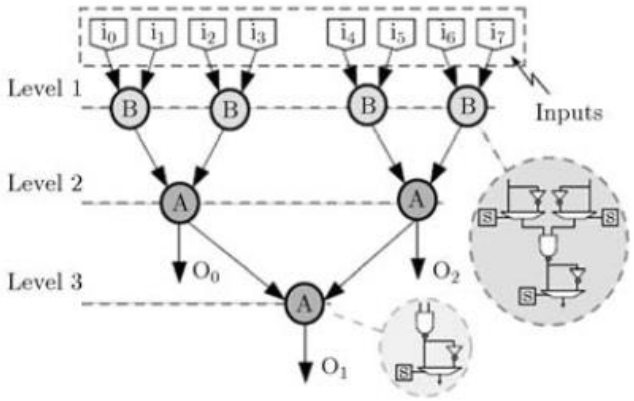
بلوک های منطقی مبتنی بر مالتی پلکسر از مزیت ارائه درجه بالایی از قابلیت برای تعداد نسبتا کمی از ترانزیستورها استفاده کرده اند. با این حال، این مورد با تعداد زیادی از ورودی ها به دست می آید که نیاز بالایی به منابع مسیریابی دارد. در نتیجه، چنین بلوک هایی بیشتر مناسب FPGAهای هستند که از سوئیچ های قابل برنامه نویسی با اندازه کوچک مانند فلش و ضد فیوز استفاده می نمایند.



شکل. 8 توابع پیاده سازی شده توسط MUX دو ورودی [41]

AIC (3)

The And-Inverter Cone (AIC) یک درخت دودویی کامل از سلول ها است که می تواند به عنوان گیت های NAND یا AND 2-ورودی پیکربندی شود. معماری AICs از]And-Inverter Graphs (AIGs) [42]، الهام گرفته است که در آن تمام گره ها، گیت های AND 2 ورودی با وارونگی اختیاری در خروجی هستند. با این حال، در این معماری اصلی AIC، اینورتر در ورودی سلول ها در دسترس نیست. بنابراین، گره هایی که گنجایش خروجی آنها شامل هر دو لبه های معکوس و غیر معکوس می شود را نمی توان به عنوان نماینده معرفی نمود. برای جای دادن این موارد، برخی از تحولات AIG مورد نیاز است، مانند تکثیر گره که دارای گنجایش خروجی و یا اضافه کردن یک گره اینورتر در خروجی آن است. این کار انعطاف پذیری بیشتر را فراهم می کند و اجازه می دهد تا AIC زیر مجموعه بزرگتری از توابع را نگاشت نماید و معماری AIC در شکل 9 نشان داده شده است.



شکل. 9 معماری AIC [43]

اگر چه بلوک منطقی AIC قادر به پیاده سازی تمام توابع احتمالی ورودی آن نیست، کمتر همه کاره در نظر گرفته می شود، اما در مقایسه با بلوک مبتنی بر LUT-کارآمد تر است. AICs می توانند از نظر پهنای باند ورودی و خروجی غنی تر باشند، چرا که مساحت آنها به طور خطی با تعداد ورودی ها رشد می کند. همچنین تاخیر آنها به صورت لگاریتمی با ورودی های گسترده رشد می کند و خروجی متوسط، آسان تر پیاده سازی می شود.

2. نمونه های محصول تجاری

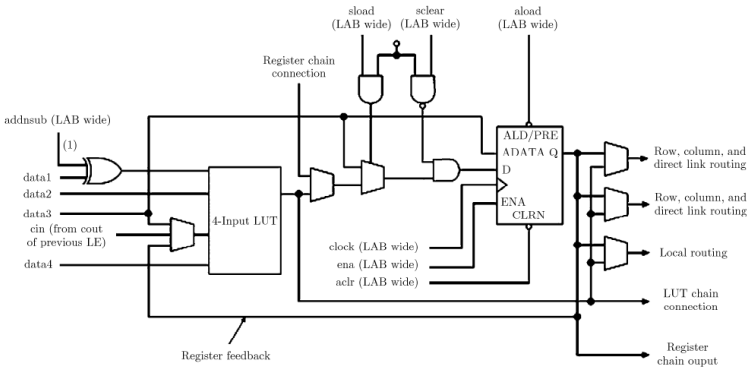
به طور کلی، کار تحقیقاتی منتشر شده در مورد معماری بلوک منطقی تمایل به مدلسازی و کشف عناصر نسبتا ساده منطق پایه، مانند LUT خالص با K-ورودی دارند. در مقابل، بلوک های منطقی تجاری تحت تکاملی قرار گرفته اند که در تلاش برای به دست آوردن قابلیت های بیشتری، به طور معمول به توسعه بلوک های پیچیده تر منجر شده است.

Altera (1)

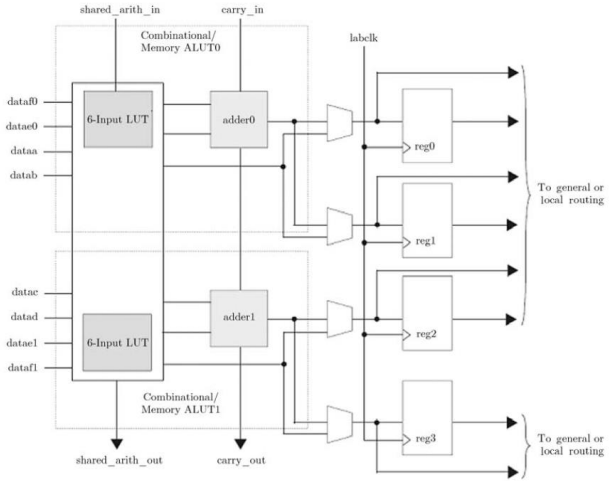
در خانواده های محصول Altera FPGA، معماری های مبتنی بر Stratix و مبتنی بر Cyclone از عناصر منطقی (LES) از انواع مختلف مرتب شده در بلوک های آرایه منطقی (LABS) استفاده می نمایند. در، I FPGA، هر LAB شامل تعدادی از LEها می شود، که از یک LUT 4- ورودی، یک ثبات قابل برنامه نویسی، زنجیره حمل با قابلیت انتخاب حمل، و برخی از منطق های کنترل دیگر تشکیل شده است، همانطور که شکل. 10 نشان می دهد. با LUT 4-ورودی، هر LE می تواند هر تابع از چهار متغیر را پیاده سازی نماید و از حالت اضافه نمودن و یا تفریق بیت تک پویا و به صورت قابل انتخاب توسط یک سیگنال کنترل در گستره-LAB نیز پشتیبانی می کند. این دستگاه دارای یک معماری ساده است که به راحتی قابل نگاشت است و در نرم افزار سنتز می شود و برای FPGA های کم هزینه، مناسب در نظر گرفته می شود. این معماری همچنان در خانواده Cyclone 1-4 FPGA [27-30] استفاده می شود.

برای دریافت ویژگی های پیشرفته با استفاده از منطق کارآمد، معماری ها در Stratix II در حد زیادی تکامل یافته اند. ماژول منطقی تطبیقی (ALM) به جای LE به کوچکترین واحد منطقی تبدیل شده است، و دارای ویژگی اصلی LABها در خانواده های دستگاه Stratix زیر است. به عنوان مثال، ALM در دستگاه های Stratix V متشکل از مدارهای ترکیبی، چهار ثبات، و دو جمع کننده می شوند، همانطور که در شکل 11 نشان داده شده است. بخش منطقی ترکیبی دارای هشت ورودی از جمله دو LUTS تطبیقی با استفاده از تکنولوژی LUT ثبت شده Altera می شوند. یک ALM کلی برای پیاده سازی یک تابع با شش ورودی دلخواه مورد نیاز است. از آنجا که ALM دارای هشت ورودی برای بلوک منطقی ترکیبی است، یک ALM از پیاده سازی ترکیباتی از دو تابع منفرد حمایت می کند. علاوه بر پیاده سازی کامل LUT 6-ورودی، ALM می تواند برای مثال، 2 تابع 4-ورودی مستقل با یک تابع 5 ورودی و توابع 3-ورودی را با ورودی های مستقل پیاده سازی نماید [31]. از آنجا که 4 ثبات و 2 جمع کننده در دسترس هستند، ALM دارای انعطاف پذیری برای پیاده سازی 2.5 عناصر منطق (LES) از معماری LUT کلاسیک 4-ورودی (LUT-4) ، متشکل از 4 LUT، منطق حامل، و یک ثبات است [32]. در مقایسه با LEs با 4 LUTS 4-ورودی ثابت، معماری ALM نمایشگر قابلیت بیشتر و انعطاف پذیری است که برتری بالاتر عملکرد LUTS و بازده مساحت LUTS کوچکتر را ارائه می دهد [44].

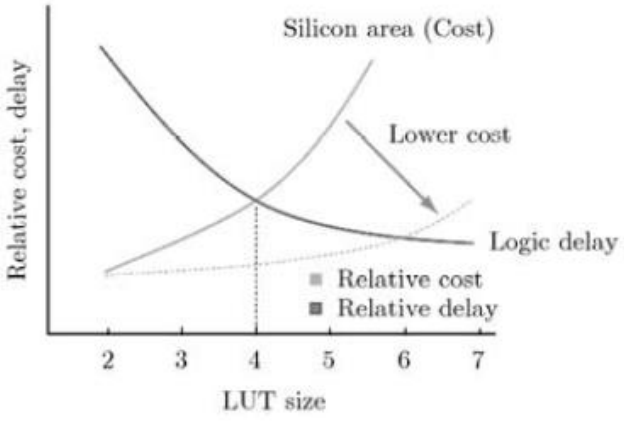
طراحی ALM نشان دهنده سبک سنگین کردن های ممکن بین عملکرد سرعت FPGA و هزینه مساحت است. نتایج تحقیقات Altera نشان داد که LUT-6 پایه می تواند بهبود عملکرد 14٪ را با کاهش تعداد سطوح عناصر منطقی در مسیر بحرانی مدارها حاصل نماید، در حالی که این افزایش عملکرد دارای یک جریمه مساحت بزرگ است، 17٪ افزایش مساحت منجر به یک ماسک LUT بزرگتر و ورودی بیشتر برای LUT می شود [32]. شکل. 12 نشان دهنده موازنه بین مساحت و تاخیر برای اندازه های مختلف LUTS است. رویکرد اساسی در طراحی ALM نه تنها برای بررسی ساخت LUT بزرگتر به منظور کاهش سطوح منطقی و از این رو افزایش عملکرد، بلکه برای جلوگیری از افزایش مساحت موثر با تقسیم LUT بزرگتر بر LUTS کوچکتر در صورت اقتضا است، همانطور که توسط خطوط تکه تکه نشان داده شده است. توانایی تقسیم LUT چیزی است که آن را "تطبیقی" می سازد.



شکل 10 معماری LE در Stratix I [26]



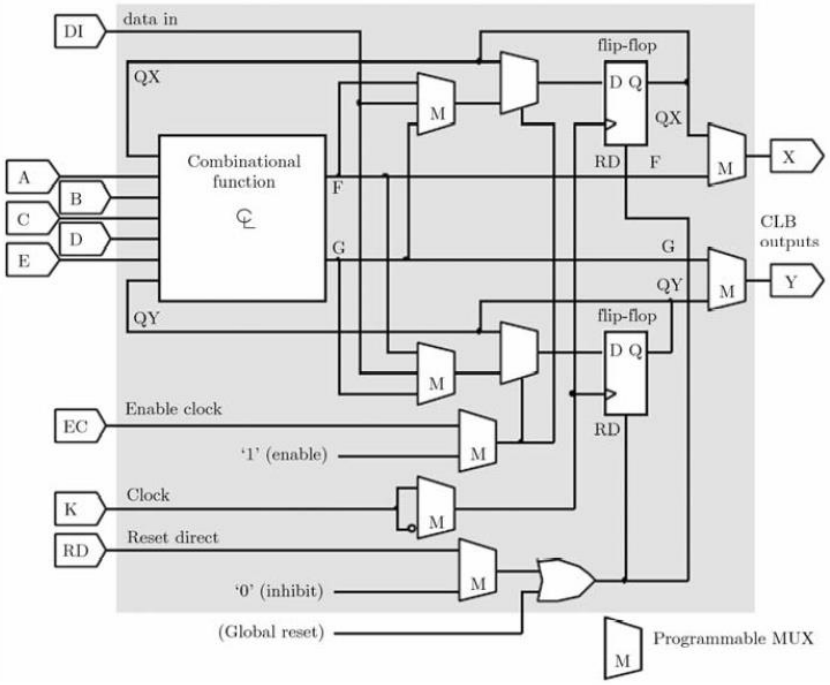
شکل 11 بلوک دیاگرام ALM در Stratix V [31]



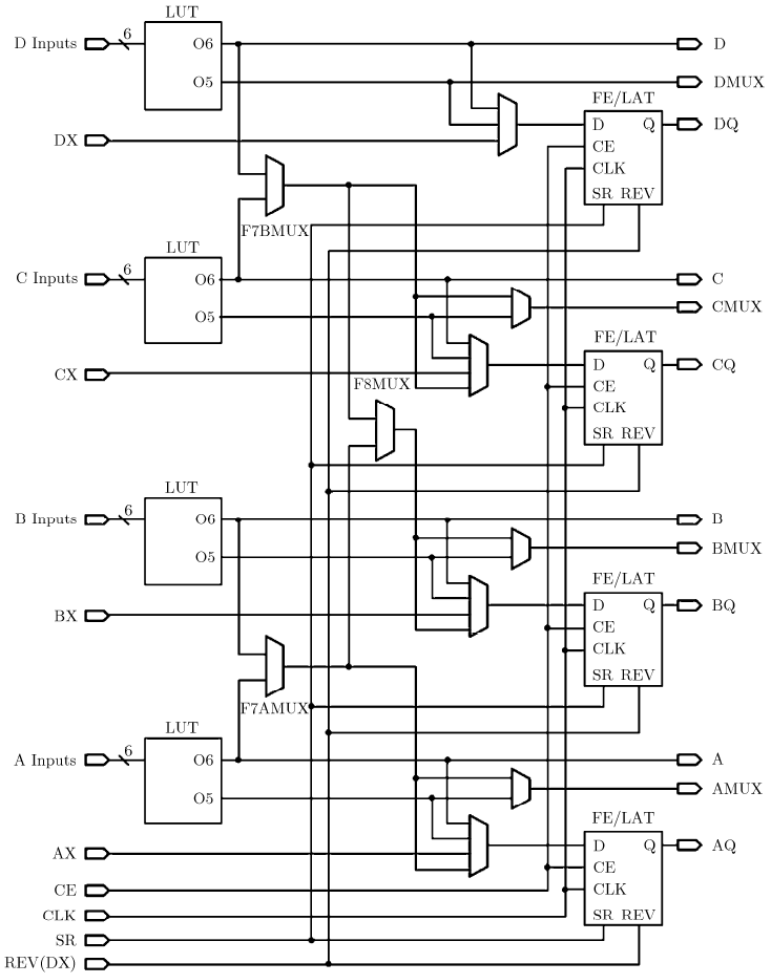
شکل 12 موازنه تاخیر-هزینه با اندازه LUT در طراحی ALM [32]

Xinlix (2)

در اغلب ادوات Xinlix FPGA، عنصر منطق پایه (BLE) مبتنی بر LUTS با ورودی های-تثبیت شده است. به عنوان مثال به دستگاه اولیه XC3000 (شکل 13) نگاهی می اندازیم، هر BLE شامل یک 5 LUT -ورودی است که می تواند هر تابع بولی 5-ورودی مستقل تعریف شده یا دو توابع بولی 4-ورودی مستقل تعریف شده را پیاده سازی نماید، تا زمانی که این دو توابع اشتراک ورودی های مشترک داشته باشند. با این حال، در آن زمان (حدود سال 1995 ~ 2000) ابزار EDA برای سنتز بسیار کارآمد و قرار دادن این تولیدکننده تابع مختلط مناسب نبود، و با معماری هایی بر اساس 4 LUTS -ورودی در Virtex 1-4 FPGA جایگزین شد. این تغییر می تواند به عنوان یک مصالحه بین مساحت و عملکرد دیده شود.



شکل. 13 نمودار عنصر منطق اساسی XC3000 [41]



شکل. 14 نمودار برش در Virtex-5 FPGA [33]

Xilinx، Virtex-5 FPGA را در سال 2007 پیشنهاد نمود، که در آن بلوک های قابل پیکربندی مستقیم (CLBs) بر اساس 6 LUTS -ورودی به عنوان عناصر اساسی مورد استفاده قرار گرفتند. CLB شامل دو برش می شود، که هر یک به عنوان یک ستون سازماندهی شده است، و شامل چهار 6 LUTS -ورودی، چهار عنصر ذخیره سازی، تعدادی از مالتی پلکسرها، و منطق کری می شود. این عناصر توسط تمام برش ها برای ارائه منطق، حساب، و توابع ROM مورد استفاده قرار می گیرند. مولدهای تابع به عنوان LUTS شش ورودی پیاده سازی می شوند و هر یک می توانند به عنوان 6 LUT -ورودی با یک خروجی ، و یا به عنوان دو 5 LUTS -ورودی با خروجی جداگانه اما آدرس مشترک و یا ورودی های منطقی پیکربندی شوند. هر یک از خروجی های LUT 5-ورودی به طور اختیاری می توانند در

یک فلیپ فلاپ ثبت شود و 4 فلیپ فلاپ در هر برش به صورت اختیاری می تواند به عنوان لچ پیکربندی شده است. نمودار برش در Virtex-5 در شکل 14 نشان داده شده است.

علاوه بر این، برخی از برش ها در Virtex-5 از توابع اضافی مانند ذخیره سازی داده ها با استفاده از LUTS و تغییر داده ها با ثبات 32 بیتی پشتیبانی می نمایند [33]. این برش ها SLICEM نامیده می شوند که در آن مولدهای تابع (LUTS) می توانند به عنوان منابع RAM همزمان به نام عنصر RAM توزیع شده پیاده سازی شوند. LUTS چندگانه در یک SLICEM می توانند برای ذخیره مقدار بیشتری از اطلاعات، تا 256 بیت ترکیب شوند. به طور کلی، RAM توزیع شده از نظر منابع و عملکرد برای پیاده سازی حافظه هایی که از 64 بیت یا کمتر تشکیل شده اند کارآمد تر است و در نتیجه یک موازنه را بین استفاده از عناصر ذخیره سازی برای آرایه های بسیار کوچک و RAM بلوک برای آرایه های بزرگتر [34] ارائه می دهند. یک تابع مشابه را می توان در آزمایشگاه حافظه (MLAB) از، Stratix IV، یافت که در آن هر ALM می تواند به عنوان بلوک حافظه 64 بیت پیکربندی شود [7].

مقایسه بین معماری های مختلف LUT از دو فروشندگان FPGA اصلی موجب یک سری از استدلالات شده است. این مورد توسط Altera اعلام شده است که معماری ALM انعطاف پذیر نسبت به معماری LUT ثابت 6-ورودی از دستگاه های Xilinx Virtex-5 پیشرفته تر است، و هر ALM می تواند به عملکرد معادل 1.8 عنصر منطقی بر اساس LUT 6-ورودی [45] برسد. با این حال، Xilinx اعلام نمود که عملکرد ALM تنها 1.2 برابر معماری مبتنی بر LUT ثابت با 6-ورودی است، در حالی که بیشتر از مساحت، از نظر هزینه گران قیمت است. در نتیجه، دستگاه Virtex-5 هنوز هم حاوی ظرفیت بالاتر از منابع منطقی دستگاه همتای خود، Stratix III است [46]. حقیقت این استدلال حال حاضر این بود که اتخاذ LUTS با ورودی های متعدد به عنوان واحد اصلی FPGA تبدیل به جریان اصلی در طراحی معماری FPGA شد.

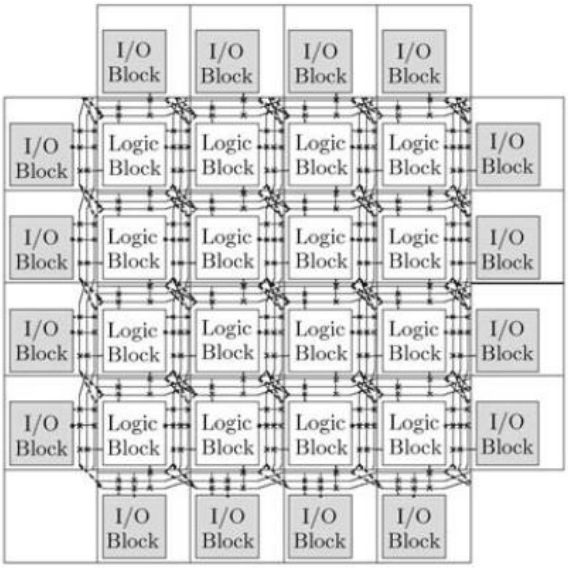
4. معماری مسیریابی

معماری FPGA شامل عناصر قابل برنامه نویسی منطقی و سبک مسیریابی قابل برنامه نویسی. مسیریابی برنامه نویسی ارتباط میان بلوک های منطقی و بلوک های I / O را برای تکمیل مدار طراحی شده کاربر فراهم می کند.

این مسیریابی از سیم ها و سوئیچ های قابل برنامه نویسی به صورت اتصالات مورد نظر تشکیل شده است. به جای طیف گسترده ای از مدارات، ساختار اتصال باید به اندازه کافی برای حمایت از خواسته مسیریابی محلی به طور گسترده متفاوت و خواسته های مسیریابی از راه دور همراه با اهداف طراحی عملکرد سرعت و مصرف برق انعطاف پذیر باشد. در معماری های تجاری، مسیریابی بیشترین مساحت تراشه را مصرف می کند و مسئول بسیاری از تاخیرات مداری است. همانطور که FPGA ها در حال حرکت به فن آوری های پیشرفته تر هستند، سبک مسیریابی مهم تر شده است. بنابراین، علاقه زیادی به توسعه معماری کارآمد مسیریابی FPGA وجود دارد. در این بخش، بحث ما به طور انحصاری روی هدف کلی مسیریابی FPGA تمرکز خواهد کرد.

1. مسیریابی کلی

موضوع اساسی در طراحی FPGA، سازماندهی معماری یونیورسال است، که تخصیص ماکروسکوپی سیم هاست. بر اساس ترتیب منابع منطقی و اتصال، معماری مسیریابی FPGA ها به طور گسترده به سه نوع اصلی زیر دسته بندی می شود: معماری مسیریابی به سبک جزیره ، معماری مسیریابی به سبک کانال و معماری مسیریابی سلسله مراتبی. امروزه، سبک جزیره، جریان اصلی معماری مسیریابی مورد استفاده در FPGA های تجاری است. سبک کانال و معماری مسیریابی سلسله مراتبی را می توان در برخی از ادوات FPGA اولیه، مانند Microsemi ACT، SX، و خانواده MX FPGA ها برای سبک کانال [47]، و Flex10K Altera، Apex و آپکس II FPGA ها برای معماری مسیریابی سلسله مراتبی [48-50] یافت. برای بقیه این بخش، ما در مسائل طراحی معماری مسیریابی به سبک جزیره تمرکز می کنیم.



شکل. 15 مثالی از سبک جزیره FPGA [51]

معماری مسیریابی به سبک جزیره شامل آرایه ای از بلوک های منطقی قابل برنامه نویسی متصل از طریق کانال های مسیر یابی عمودی و افقی قابل برنامه نویسی می شود، همانطور که شکل. 15 نشان می دهد [51]. این معماری معمولاً دارای کانال های مسیر یابی در هر چهار طرف از بلوک های منطقی است و ورودی یا خروجی یک بلوک منطقی می تواند به کانال های مسیریابی با بلوک های اتصال متصل شود. معماری های مسیریابی به سبک جزیره به طور کلی برای تقسیم بندی سیم با طول های مختلف برای برنامه نویسی اتصال داخلی انعطاف پذیر به کار گرفته می شود و کانال های مسیر یابی افقی و عمودی در هر تقاطع با بلوک سوئیچ قابل برنامه نویسی متصل می شود.

2. مسیریابی مفصل

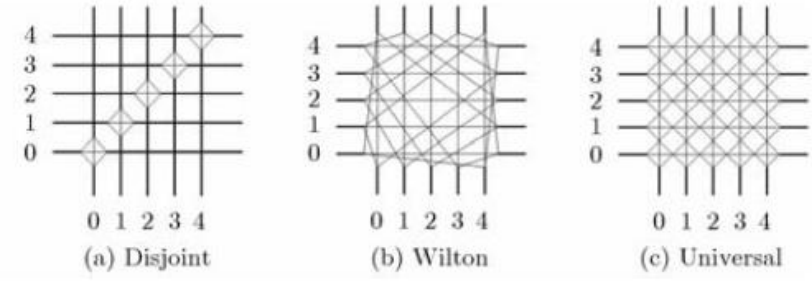
معماری مفصل مسیریابی FPGA، ساختار منطقی اتصال بین هر یک از بخش های دیگر سیم و بلوک های I / O را تعریف می کند. بلوک های سوئیچ برای ایجاد ارتباط بین این بخش سیم در هر تقاطع کانال ها استفاده می شوند. در ادامه این بخش، روی الگوهای بهینه سازی سوئیچ، تقسیم بندی کانال مسیریابی و طراحی مسیر یابی دو جهته در FPGA ها تمرکز می کنیم.

(1) جعبه سوئیچ

بلوک سوئیچ یک بلوک اتصال برنامه نویسی در تقاطع هر یک از کانال های افقی و عمودی است، که از نظر قابل برنامه نویسی بودن، مسیر ورودی را به تعدادی از مسیرهای خروجی متصل می کند. واضح است که انعطاف پذیری هر بلوک سوئیچ برای انعطاف پذیری کلی و قابلیت مسیریابی دستگاه ضروری است. از آنجا که ترانزیستورها در بلوک سوئیچ، بارگذاری خازن را به هر مسیر می افزایند، بلوک سوئیچ دارای اثر قابل توجهی در سرعت هر اتصال مسیریابی، و از این رو سرعت FPGA است. علاوه بر این، از آنجا که چنین بخش بزرگی از FPGA به مسیریابی اختصاص داده شده است، سطح تراشه های مورد نیاز در هر بلوک سوئیچ دارای یک اثر بزرگ در چگالی منطقی دست یافتنی از دستگاه خواهد بود. بنابراین، طراحی یک بلوک سوئیچ خوب از بالا ترین اهمیت برخوردار است.

شکل. 16، سه معماری بلوک سوئیچ قبلی که پیشنهاد شده است را نشان می دهد [52-55]. در هر بلوک، مسیر ورودی را می توان به سه مسیر خروجی متصل نمود، اما توپولوژی هر بلوک متفاوت است. جعبه سوییچ منفصل در تعدادی از FPGA های تجاری، مانند خانواده Xilinx XC4000 [41] استفاده می شود. همانطور که در شکل 16 (الف) دیده می شود. الگوی اتصال در بلوک منفصل "متقارن" است، که به معنی یک سیم ورودی به یک بلوک سوئیچ منفصل است که تنها می تواند به سیم های دیگر با همان طرح عددی از طریق سوئیچ های قابل برنامه نویسی اتصال یابد. در نتیجه، مسیرها در FPGA ها به حوزه های مسیریابی متمایز جدا می شوند و انعطاف پذیری مسیریابی محدود می شود. بلوک سوئیچ Wilton در شکل. 16 (ب) شبیه به بلوک سوئیچ منفصل است، با این تفاوت که هر اتصال مورب بوده توسط یک مسیر "چرخش" یافته است [56]. این مورد، مشکل حوزه های بلوک سوئیچ منفصل را حذف می کند و به بسیاری از گزینه های بیشتر مسیریابی برای هر اتصال منجر می شود. علاوه بر Wilton و بلوک های سوئیچ منفصل، تعدادی از طرح های جایگزین، مانند بلوک سوئیچ جهانی نشان داده شده در شکل 16 (ج)، نیز پیشنهاد شده است [57]. تمرکز طراحی بلوک جهانی در به حداکثر رساندن تعداد اتصالات همزمان است که می تواند با استفاده از این بلوک ساخته شود، و فعل و انفعالات بین بلوک های سوئیچ همسایه را در نظر نمی گیرد. یک بررسی کامل از بلوک های سوئیچ اضافی بهینه سازی شده به سبک جزیره برای بخش های طول 1 سیم را می توان در مرجع [58] یافت.

هر یک از بلوک ها در شکل 16 توسعه داده شدند و با فرض اینکه معماری تنها با سیم های طول-تک است، مورد بررسی قرار گرفتند (یعنی سیم هایی که تنها بلوک های سوئیچ همسایه را اتصال می دهند). با این حال، در واقع، FPGA ها، معمولا دارای سیم های طویل تری هستند که بلوک های سوئیچ فاصله دار را اتصال می دهد. چنین معماری مسیریابی، معماری سگمنتال نامیده می شود و مشخص شده است که چنین معماری هایی منجر به تراکم و سرعت بالاتر از معماری تنها با سیم های تک-طول می شود [59].

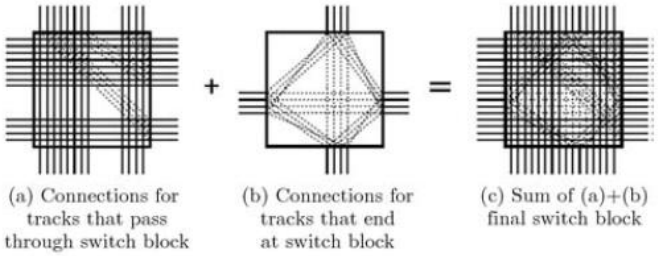


شکل. 16 بلوک های سوئیچ قبلی [52-55]

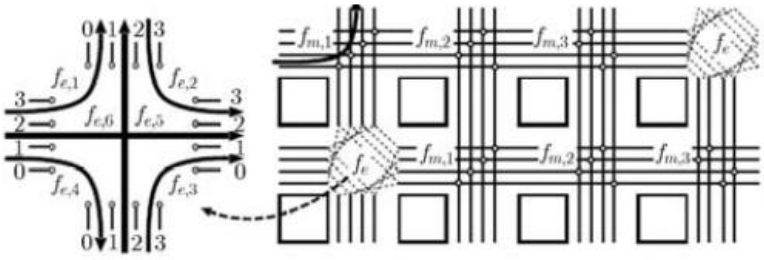
اکثر طرح های بلوک سوئیچ اخیر تنها اتصالات سوئیچ بین نقاط انتهایی سیم و یا بین نقاط میانی سیم را میسر می سازند، اما نه بین نقاط انتهایی و نقاط میانی. اتصالات نقطه میانی به نقطه میانی با استفاده از اتصالات منفصل تک ساخته شده است. مثالها عبارتند از: بلوک های سوئیچ Imran [56] و shifty [60]، همانطور که در شکل 17 و شکل18 نشان داده شده است. بلوک سوییچ Imran از یک بلوک سوئیچ Wilton برای اتصال نقاط پایانی سیم ها و اتصالات منفصل تک ترانزیستور برای اتصال نقاط میانی استفاده می کند. این بلوک سوئیچ از نظر مساحت کارآمدتر از بلوک سوئیچ های منفصل، یونیورسال، و یا Wilton نشان داده است [56]. بلوک سوئیچ shifty به طور مشابه، تغییرات دامنه مسیریابی را در دورهای نقطه پایانی و اتصالات منفصل را در نقاط میانی میسر می سازد. آزمایش نشان داده است که بلوک های سوئیچ shifty و Imran نتایج مشابهی در مساحت و تاخیر را نشان می دهند [60]. به علت توانایی خود برای مسیرهای متنوع مسیریابی، هر دو بلوک سوییچ نسبت به بلوک های سوئیچ منفصل در مساحت و عملکرد تاخیر برتر هستند.

(2) تقسیم بندی کانال

در طراحی معماری مسیریابیFPGA ، یک بخش سیم به عنوان تعدادی از بلوک های منطقی تعریف می شود که یک سیم مسیریابی قبل از خاتمه گسترش می یابد. همه FPGA ها از تقسیم بندی کانال مسیریابی استفاده می نمایند که به موجب آن هر مسیر مسیریابی به بخش های سیم با طول های مختلف تقسیم می شود. مطالعات نشان داده است که ترکیبی از طول های بخش مورد استفاده در مسیر های مختلف مسیریابی می توانند تاثیر قابل توجهی در عملکرد اتصال [6،58]، و از این رو عملکرد کلی FPGA داشته باشند.



شکل. 17 بلوک سوئیچ Imran [60]



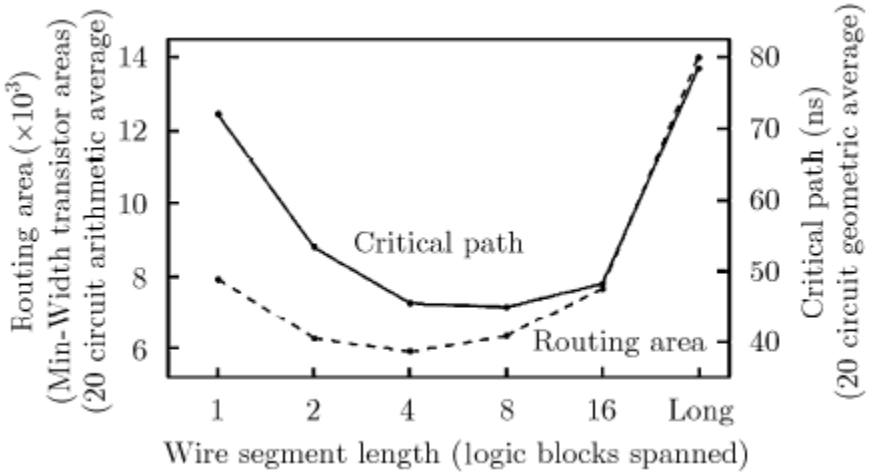
شکل. 18 بلوک سوئیچ shifty [60]

از نظر مفهومی، معماری مسیریابی با بخش های کوتاهتر به قابلیت مسیریابی بهتر و بارگذاری خالص بیش از حد پایین تر منجر می شود، که به معنی تراکم منطقی بالاتر و مصرف برق پایین تر است. با این حال، در FPGA با بیش سیم های کوتاه زیاد، برخی از اتصالات طولانی باید استفاده از چند بخش سیم کوتاه متصل به صورت سری ساخته شود که به تاخیر طولانی تر منجر می شود. از سوی دیگر، عملکرد سرعت معماری مسیریابی را می توان با افزایش بخش های دیگر بهبود داد، اما با تراکم منطق پایین تر و مصرف برق بالاتر است. علاوه بر این، در مرجع [61] مشاهده می نماییم که با مقیاس بندی تکنولوژی CMOS، طول متوسط بخش سیم باید به دلیل افزایش در

پارازیتی های سیم نسبت به پارازیت ادوات کاهش یابد. با توجه به این سبک سنگین کردن ها و مشاهدات، تقسیم بندی کانال مسیریابی باید برای بهینه سازی عملکرد کلی FPGA به دقت انتخاب شود.

در مرجع. [62]، طراحی کانال مسیریابی سگمنتال برای اولین بار برای FPGA های مبتنی بر ردیف مورد بحث قرار گرفت. روش ساخت یک کانال مسیریابی سگمنتال برای دریافت قابلیت مسیریابی بالا نشان داده شده است، با فرض اینکه هر دو نقاط منشاء تصادفی و طول اتصال هندسی توزیع شده. در مرجع [63]، این رویکرد آماری به FPGA ها به سبک جزیره گسترش داده شود و که در آن توزیع های تجربی برای طول های قطعه برای اولین بار توسطتجزیه و تحلیل آماری طرح های قرار داده شده و مسیریابی شده مشخص شدند و سپس با توجه به تقاضا برای هر طول بخش، بخش های جدای کانال افقی و عمودی یافت شد.

روش های تجربی نیز اقدام شدند. در مرجع [59]، Betz و همکاران. یک مسیریاب FPGA معاصر که ترکیبی از یونیورسال و مفصل در یک مرحله برای ارزیابی تقسیم بندی استفاده نمود. در شکل 19 نشان داده شده است که در میان کانال هایی از بخش های با طول برابر، یک کانال تنها با بخش 4 طول به کمترین مساحت مسیریابی و کوتاه ترین تاخیر مسیر بحرانی دستیابی پیدا می کند. علاوه بر این، یک کانال مسیریابی با مخلوطی از بخش هایی با طول 4 و 8 می توانند یک آرایش کانال را تنها با طول 4 بخش بهتر عمل نماید. به طور مشابه در مرجع [64]، تقسیم بندی یکنواخت بهینه از نظر تجربی برای FPGA فرآیند 100 نانومتر مورد بررسی قرار گرفت، و نتایج آنها نشان داده است که استفاده از بخش هایی با طول 3 منجر به پایین ترین مصرف انرژی و همچنین محصول انرژی تاخیر مساحت می شود. این مطالعات اهمیت گنجاندن بخش هایی با طول متوسط را که بین چند بلوک منطقی در یک معماری مسیریابی به سبک جزیره گسترش می یابد، تایید نمود و در طول توسعه معماری، Stratix [65] اعتبارسنجی شده است که شامل بخش هایی با طول قابل توجه 4 و 8 می شود.

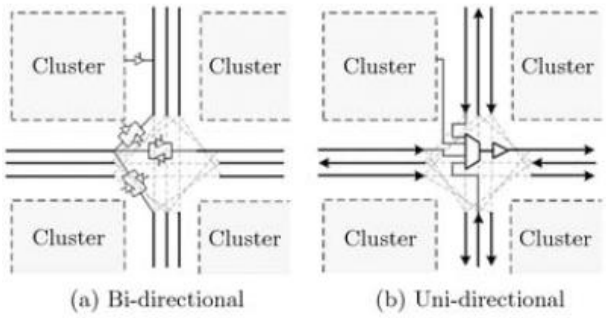


شکل. 19 سرعت و مساحت FPGA ها در مقابل طول مسیر یابی بخش سیم [59]

علاوه بر الگوی اتصال و پارامترهای کمی، عملکرد معماری مسیریابی FPGA مفصل توسط انواع سوئیچ استفاده شده برای ایجاد اتصالات، اندازه ترانزیستورهای استفاده شده برای ساخت سوئیچ برنامه نویسی و عرض فلزی و فاصله از سیم اتصال اداره می شود [59]. سوئیچ های مسیریابی معمولاً از مجموعه ساختارهای ترانزیستور پایه از جمله ترانزیستورهای پاس، بافرها ، و مالتی پلکسرها ساخته شده اند.

(3) دو جهته یا یک سویه

شکل. 20 نشان دهنده ساختارهای اساسی مسیریابی دو جهته و یک سویه است. در مسیریابی دو جهته، خروجی کاشی خوشه از طریق یک بافر متصل می شود، و به طریقی دیگر، در مسیریابی یک سویه، خروجی به مالتی پلکسر با سیم های دیگر (از هر دو خروجی خوشه و سیم های دیگر در کانال) متصل می شود، و سپس یک بافر، خروجی این مالتی پلکسر را تامین می کند.



شکل 20 ساختار پایه دو جهته و تک جهته [66]

بر اساس ساختار مسیریابی مورد بحث، برخی تضادها بین معماری های مسیریابی تک-جهته و دو جهته قابل مشاهده است. اول، از نظر مساحت، مسیریابی تک-جهته با توجه به بافر به اشتراک گذاری تسهیل شده با بهره گیری از سوئیچ مسیریابی مالتی پلکسر / درایور، مساحت کمتری را نسبت به مسیریابی دو جهته مصرف می کند. این اثر توسط Lemieux [67] و Lewis [68] نشان داده شده است، و در آثار آنها نشان داده شده است که مساحت کلی مورد نیاز برای پیاده سازی معماری مسیریابی تک-جهته نسبت به معادل دو طرفه، حدود 20٪ کمتر کاهش می یابد. ثانیاً، شبکه های تک-جهته مسیریابی به نقاط سوئیچینگ کمتری اتصال پیدا می کنند و در نتیجه بار خازنی کمتر است. اما برای وضعیتی که یک مسیر مسیریابی حداقل از طریق یک جعبه سوئیچ حرکت می کند، معماری مسیریابی دو جهته دارای سوئیچ های قابل برنامه نویسی کمتر در مقایسه با مسیریابی تک-جهته است. در طراحی FPGA مسیریابی مدرن، مسیر یابی تک-جهته برای بسیاری از مسیر ها مورد استفاده قرار می گیرد، و مسیر یابی دو جهته فقط برای اتصالات طولانی در سراسر چندین بلوک های سوئیچ برای مهار گسترش با توجه به تعداد چنین اتصالات طولانی استفاده می شود.

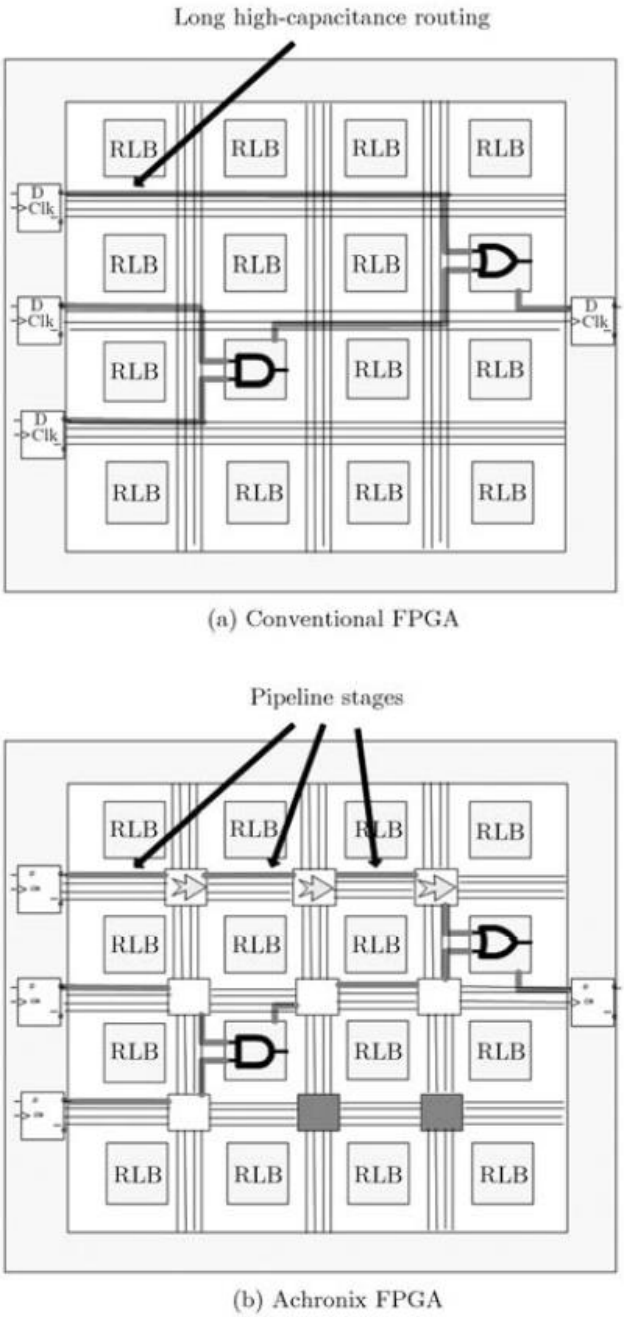
3. اتصال آسنکرون

زمانی که اندازه منطقی FPGA رشد می کند، برخی از چالش های معماری FPGA معمولی وجود دارد. اول، تاخیر سیم های اتصال طولانی به راحتی می توانند بر همه تاخیرهای دیگر تسلط پیدا کنند. دوم، توزیع مساوی سیگنال های ساعت جهانی در سراسر مساحت FPGA به دلیل انحراف ساعت به تلاش های طراحی بزرگ نیاز دارد. سوم، FPGAها به احتمال زیاد شامل بسیاری از ماژول های اجرایی در فرکانسهای مختلف ساعت می شوند، زیرا آنها به اندازه های مناسب قالب رشد می کنند. به نظر می رسد سیگنال های داده در دامنه جدید ساعت ناهمزمان هستند، هنگامی که داده ها در سراسر ماژول حرکت می کنند [69].

معرفی مفهوم ناهمزمان بر معماری FPGA، یک راه حل ممکن برای چالش های نامبرده شده است. از لحاظ تاخیرات اتصال، عملکرد توسط متوسط تاخیرات اتصال دیکته می شود نه تاخیر بدترین حالت. با اتخاذ طراحی ناهمزمان،

FPGA ها می توانند حمایت های معماری را برای ارتباطات در سراسر حوزه های مختلف ساعت فراهم کنند. ماژول های مختلف در حال اجرا در فرکانسهای مختلف ساعت می توانند به راحتی با هم چسب زده شوند.

FPGA Achronix به طور معمول با اتصال ناهمزمان پیاده سازی می شود و شکل 21 اصل معماری مسیریابی ناهمزمان Achronix FPGA را نشان می دهد.

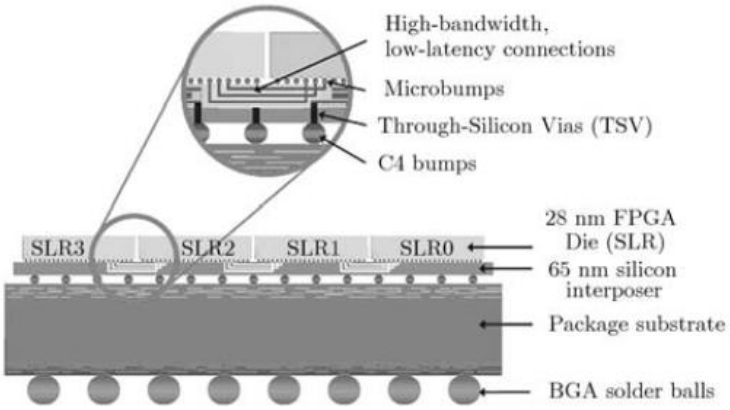


شکل. 21 پیاده سازی متعارف در مقابل پیاده سازی ناهمزمان [71]

در عرض یک FPGA سنتی، سیگنال هایی که در مسیرهای طولانی مسیریابی حرکت می کنند، از یک بار خازنی بالا رنج می برند، و زمانی که FPGA در اندازه بزرگتر شود، همه چیز بدتر و بدتر می شود. در FPGA های ناهمزمان، یک خط لوله داخلی تضمین می کند که سیگنال ها تنها به سفر در مسیر های کوتاه مسیریابی نیاز دارند که این موجب کاهش بار خازنی در هر مرحله می شود. علاوه بر این، معماری ناهمزمان تضمین می نماید که تنها یک سطح منطقی در هر مرحله خط لوله وجود دارد که نرخ بیشتری از نشانه های داده ها [70] را میسر می سازد.

4. 2.5 اتصال بعدی

فن آوری The Xilinx Stacked Silicon Interconnect (SSI) یک مجتمع سازی 2.5 بعدی از چند ادوات فعال متعدد روی یک واسطه پسیو برای تشکیل یک دستگاه واحد است، همانطور که در شکل 22 نشان داده شده است. این فناوری برای پاسخگویی به چالش هایی پیاده سازی شده است که برای ترکیب منطق اتصال دو یا چند FPGA ها و ایجاد یک FPGA بزرگتر برای پیاده سازی یک طراحی پیچیده تلاش می نمایند. این چالش ها به طور عمده عبارتند از: مقدار I / O در دسترس برای اتصال کافی نیست؛ زمان تاخیر سیگنال های عبوری بین عملکرد محدودیت FPGA ها؛ و مصرف برق با استفاده از دستگاه استاندارد I / O برای ایجاد اتصالات منطقی بین چند FPGA افزایش می یابد.



شکل. 22 سطح مقطع ها Stacked Silicon Interconnect [72]

با افزایش چالش ها، فن آوری SSI از واسطه های سیلیکونی پسیو با میکرو ضربه ها و Through-Silicon Vias (TSVs) برای ترکیب چند قالب FPGA در یک بسته واحد استفاده می کند و اتصال با پهنای باند بالا بین قالب

های مختلف را با ارائه تعداد زیادی از اتصالات میسر می سازد. این کار به زمان تاخیر کمتر و مصرف برق پایین تر از رویکرد FPGA چندگانه نیاز دارد، در حالی که امکان مجتمع سازی مقادیر عظیمی از منابع منطق اتصال، فرستنده و گیرنده، و بر روی چیپ وجود دارد. علاوه بر این، این مقوله در مراحل اولیه یک گره فرآیند جدید، از اهمیت فوق العاده برخوردار است، زمانی که تراکم نقایص هنوز هم بسیار بالا می باشد و عملکرد قالب به طور چشمگیری با افزایش اندازه کاهش می یابد. این نیز به حداکثر رساندن عملکرد هر یک از تراشه ها کمک می کند.

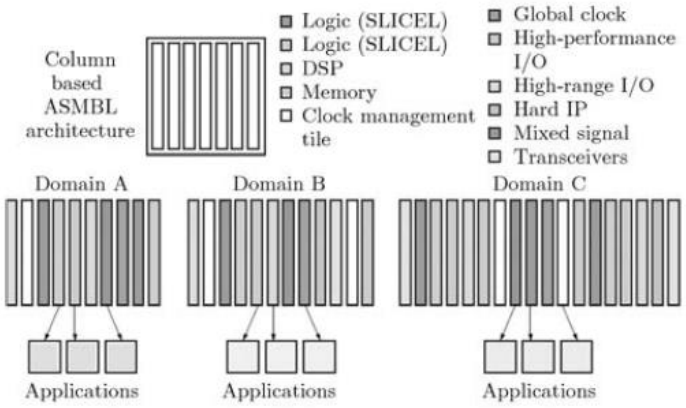
شکل. 22 دیدگاه جانبی پشته قالب را با چهار برش FPGA، واسطه سیلیکون، و بستر بسته نشان می دهد. واسطه سیلیکون به عنوان یک وسیله نقلیه اتصال بر اساس فرآیند تولید سیلیکون عمل می کند که در آن قالب های متعدد در کنار تنظیم و به هم پیوسته می شوند. این نوآوری کلیدی، استاندارد I / Os را با هزاران اتصال قالب-به-قالب از طریق آثار پسیو ساخته شده در واسطه سیلیکون افزایش می دهد و اتصال بالا و زمان تاخیر کم را بدون تحمیل جریمه توان بر ساختارهای I / O سنتی فراهم می کند. علاوه بر این، فن آوری SSI از موضوعات قدرت و قابلیت اطمینان ناشی از انباشته شدن چند قالب FPGA در بالای یکدیگر است اجتناب می نماید. در مقایسه با بسترهای آلی یا سرامیک، واسطه های سیلیکون در نظر گرفته شده برای ارائه هندسه بهتر اتصال (حدود X20 سیم زمین متراکم تر) به منظور ارائه سلسله مراتب اتصال مقیاس-دستگاهی در نظر گرفته می شوند که از نظر تئوری، تا 10000 اتصال قالب-به-قالب را فراهم می کند [73].

5. منابع جاسازی شده

با افزایش میزان و بهبود عملکرد ماژول های تابعی، مانند DSP، حافظه، مدیر ساعت، فرستنده و گیرنده با سرعت بالا، و غیره، FPGA به یک دستگاه سیستم-فعال ضروری تبدیل شده است. گنجاندن این منابع جاسازی شده دارای تاثیر زیادی بر عملکرد FPGA های مدرن است. علاوه بر این، فروشندگان تجاری اصلی به جای یک محصول واحد FPGA، تمایل به ارائه یک سری از سیستم عامل های FPGA ، با ترکیبی متغیر از ویژگی های بهینه سازی شده برای حوزه های کاربردهای مختلف دارند [74]. چگونگی دستیابی به یک مونتاژ سریع و مقرون به صرفه از سیستم عامل های FPGA با ماژول های مختلف عملکردی، به یک مسئله طراحی FPGA تبدیل شده است. در این بخش ما

این مشکل را مورد بحث قرار خواهیم داد و نمایش گسترده را از ویژگی های طراحی Altera و Xilinx ارائه می دهیم.

Xilinx، معماری پیشرفته Advanced Silicon Modular Block (ASMBL) را برای میسر نمودن مونتاژ سریع و مقرون به صرفه اشکال نقشه کشی FPGA با ویژگی های مختلف خلق نمود [75]. شرح سطح بالایی از این معماری در شکل 23 نشان داده شده است. در این معماری، منابع منطقی، مانند بلوک های قابل پیکربندی مستقیم (CLB)، DSP، حافظه و غیره، در ستون مرتب شده اند و می توانند برای برنامه های کاربردی هدف انتخاب شوند. مشتریان می توانند سیستم عامل FPGA را با ترکیب مناسبی از امکانات و قابلیت های طراحی خاص خود را انتخاب کنند. مزیت اصلی معماری ASMBL از بین بردن محدودیت های طرح هندسی است که به معنی هر نوع از بلوک های IP سخت است که می توانند مستقل از منابع پیرامون مقیاس بندی شوند و توان الکتریکی و زمین را می توان در هر نقطه بر روی تراشه [76] قرار داد.

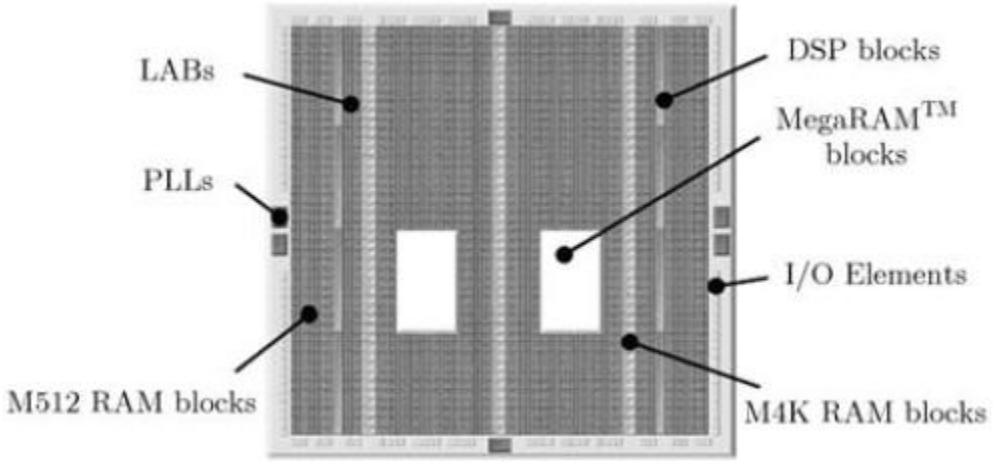


شکل. 23 منابع مبتنی بر ستون در معماری ASMBL [77]

ادوات Altera شامل آرایه ای از LABهای متصل شده توسط سطرها و ستون های سیم های مسیریابی می شوند. معماری LAB به طور طبیعی یک طرح بلند و باریک را ایجاد می کند. همانطور که شکل 24 نشان می دهد، ماژول های کاربردی، مانند MegaRAMs، به عنوان بلوک های مستطیل سازمان یافته اند و دارای عرض بسیار بزرگتری نسبت به ستون ها در Virtex هستند. این ماژول ها نمی توانند مونتاژ شوند و به طور مستقیم مانند معماری ASMBL برداشته می شوند، و راه اندازی مجدد سنتز و جایگذاری برای عملکرد سیستم عامل FPGA های مختلف

مورد نیاز است. برای مشتریان، این جریان طراحی به معنای انتخاب گسترده با سیستم عامل های "شخصی" FPGA است، در حالی که برای طراحان FPGA، تجربه بیشتر از سفارشی سازی، و همچنین نرم افزار و ابزار EDA بسیار قوی تر برای حمایت از سنتز و جایگذاری سریع مورد نیاز است [65].

یکی دیگر از معماری های مهم منابع جاسازی شده را می توان در SOC FPGA ها یافت که در آن ریزپردازنده و FPGA در یک دستگاه واحد مجتمع می شوند. آمیختن این دو تکنولوژی، انواع مزایا از جمله ادغام بالاتر، انرژی کمتر، اندازه برد کوچکتر، و ارتباطات پهنای باند بالاتر بین آنها را فراهم می کند. علاوه بر این، سیستم SOC FPGA انعطاف پذیری کاملی برای انتخاب هر ترکیبی از لوازم جانبی و کنترل کننده ها را دارد و توانایی خوبی برای موازنه ها بین سخت افزار و نرم افزار را به منظور به حداکثر رساندن بهره وری و عملکرد نشان می دهد. ادوات خانواده Arria Altera و ادوات خانواده Zynq Xilinx نمونه معمولی از این معماری [8،9] هستند.



شکل. 24 بررسی اجمالی از طبقه-طرح Stratix [65]

6. روندهای آینده از FPGA

در زمینه توسعه FPGA، نگرانی های مداوم در مورد مساحت، سرعت و قدرت باید در مسائل طراحی FPGA ها در نظر گرفته شوند. علاوه بر این، برخی از فن آوری های دیگر در حال ظهور وجود دارند که ممکن است بر روند آینده FPGA تاثیر داشته باشند، مانند FPGA های جدید مبتنی بر حافظه و 3D FPGA ها. در این بخش به معرفی مختصر آنها ارائه می پردازیم.

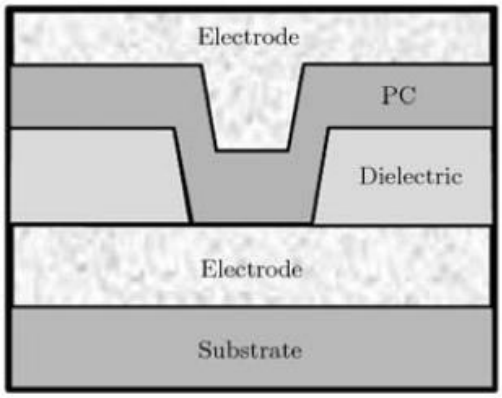
1. فن آوری های حافظه جدید

در طول دهه گذشته، صنعت نیمه هادی یک تجدید حیات در علاقه برای جستجو در فن آوری های حافظه بسیار مقیاس پذیر را تجربه کرده است. با پیشرفت فن آوری های جدید حافظه غیر فرار به سرعت در حال رشد، علاقه رو به رشدی در بررسی استفاده از آنها در FPGA ها در آینده به وجود آمده است.

بسیاری از تلاش های تحقیقاتی به منظور توسعه یک تکنولوژی حافظه جدید برای FPGAها انجام شده است که ترکیبی از شایستگی های مورد نظر از عدم نوسانات و کارایی بالا می باشند. Kryder 13 فن آوری جایگزین (NVM) را معرفی نمود که با توجه به تراکم، عملکرد دستگاه، و احتمال موفقیت در سال 2020 [20] مورد بررسی قرار گرفتند. در میان این فن آوری های حافظه جدید، Phase-Change RAM (PCRAM) Spin-Torque , Transfer RAM(STTRAM) و Resistive RAM (RRAM) امیدوار کننده ترین نامزدها هستند و از این رو بسیاری از توجهات را [78-83] دریافت کرده اند. در این بخش، این فن آوری حافظه جایگزین و همچنین چشم انداز آنها برای FPGA ها در آینده مورد بحث قرار خواهد گرفت.

PCRAM (1)

PCRAMs از یک تغییر فاز قابل برگشت بین حالات بی شکل و بلوری از یک لیوان chalcogenide برای تولید یک تغییر مقاومت قابل برگشت در سلول استفاده می نمایند [84]. شکل 25 دیدگاه مقطعی شماتیک از سلول های PCRAM را نشان می دهد. این ساختار متشکل از الکترودهای بالا و پایین ساخته شده از ماده TiW، لایه دی الکتریک SiO2 به عنوان جداساز، و مواد تغییر فاز Ge2Sb2Te5 به عنوان لایه فعال می باشد. حالات کریستالی و بی شکل به ترتیب، دارای مقاومت کم و زیاد، برای داده های 0 و 1 است و یک عنصر گرمایش برای تولید گرما به منظور تعویض بین حالات استفاده می شود.

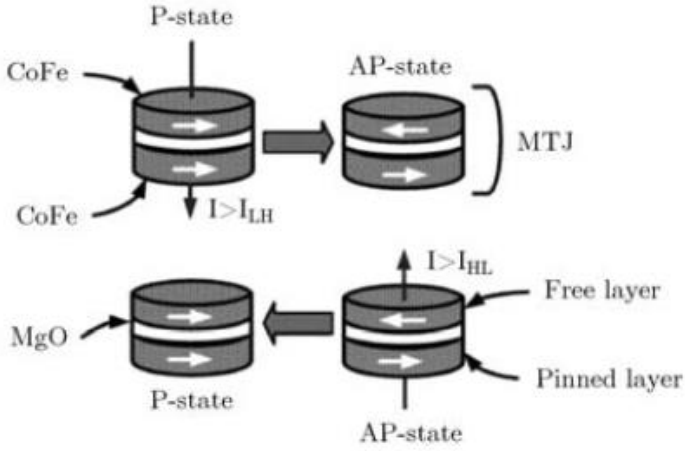


شکل. 25 مقطع شماتیک سلول PCRAM [85]

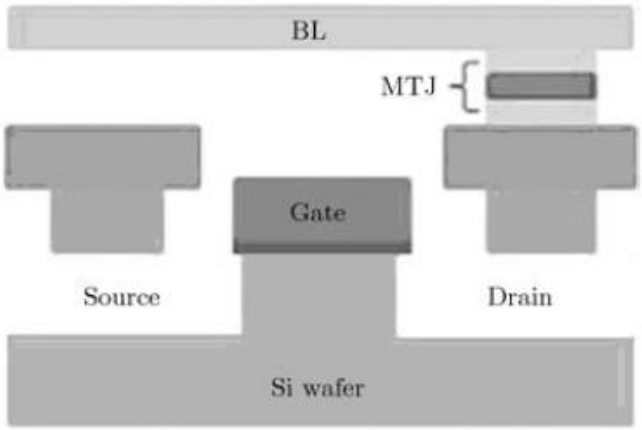
PCRAM یک تکنولوژی حافظه غیر فرار با سرعت قابل مقایسه با SRAM، ظرفیت قابل مقایسه با DRAM، استقامت تقریبا نامحدود، و سرعت نوشتن سریع تر از فلش تا چندین مرتبه است و در عین حال به فقط سه الی چهار ماسک اضافی نیاز دارد [78]. این به عنوان یکی از امیدوار کننده ترین نامزدها برای FPGA ها در آینده در نظر گرفته می شود. چالش اصلی در PCRAMs، کنترل تغییرات مقاومت در طول ساخت و پس از چرخه های خواندن / نوشتن است.

STTRAM (2)

بلوک ساختمان اساسی یک سلول STTRAM، تقاطع تونل مغناطیسی (میانبر) است (شکل 26 را ببینید). هر MTJ شامل دو لایه فرومغناطیسی از هم جدا شده توسط یک فیلم دی الکتریک تونل زنی بسیار نازک می شود. حالات سوئیچینگ MTJ از آنتی پارالل و یا "1" تا موازی و یا "0" و بالعکس با پیاده سازی یک جریان الکترونی قطبی از بالا به پایین MTJ و بالعکس [86] انجام می شود. جهت مغناطش لایه آزاد را می توان با تزریق الکترون-چرخش قطبی کنترل نمود. از این رو، MTJ را می توان بین دو حالت مغناطیسی پایدار با مقاومت بالا و یا پایین روشن نمود.



(a) Change in magnetization states for an MTJ



(b) Cell structures of STTRAM[87]

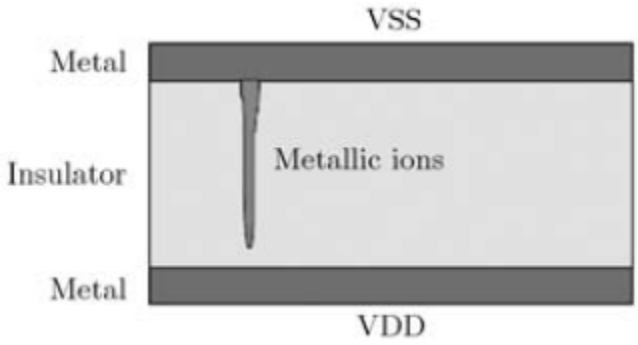
شکل. 26 شماتیک STTRAM

یک طرح برای سنجش مقاومت MTJs در مورد FPGA ها در Refs ارائه شده است. [88] و [89]. ایده اصلی، استفاده از دو عنصر MTJ برای ذخیره سازی اطلاعات یک بیتی، و پس از آن، به کارگیری تقویت کننده حس برای خواندن پیکربندی ذخیره شده است. درست مثل PCRAM، STTRAM همچنین دارای مزیت عدم نوسانات و اندازه سلول کوچک است. علاوه بر این، در مقایسه با PCRAM، STTRAM دارای نرخ خواندن/ نوشتن سریعتر و تعداد زیادی از چرخه های نوشتن است. چالش اصلی در چارچوب مجدد مبتنی بر STTRAM، چگونگی احساس حالت مقاومت برای هر بیت پیکربندی در طول عملکرد طبیعی FPGA است. مکانیسم های خواندن داده های مبتنی بر-

تقویت کننده-حس، که معمولا در حافظه های STTRAM جاسازی شده به کار گرفته می شود، زمانی که در یک چارچوب FPGA استفاده می شود، سربار زیادی را متحمل می شود.

RRAM (3)

سلول های RRAM ساختارهای شبیه به-خازن هستند که نشان دهنده پدیده سوئیچینگ مقاومتی در اکسیدهای فلزات واسطه است. این مفهوم حافظه مبتنی بر اثر تغییر مقاومت قالب برگشت ناشی از پالس و دارای یک نوع از معماری ساندویچ فلز و فلز-عایق است که در شکل 27 نشان داده شده است. یونهای فلزی در عایق، الکترولیت حالت جامد هستند که می توانند در هنگامی که بایاس الکتریکی فراهم شده است، در فیلامان ها جمع آوری شوند و مقاومت بسیار کمی را القا می نمایند. هنگامی که بایاس الکترود معکوس می شود، فیلامان ها کوچک و یا ناپدید می شوند، و مقاومت بالایی را نشان می دهند. مقدار مقاومت بالا می تواند بیش از 1000 برابر مقاومت کم به دست آید. علاوه بر این، مقدار مقاومت را می توان با برداشتن ولتاژ اعمال شده ثابت حفظ نمود.



شکل. 27 ساختار RRAM

معماری RRAM برای تحقق بخشیدن به FPGA مطرح شده است. M.Liu یک مفهوم FPGA جدید به نام rFPGA [90] را پیشنهاد نمود. در rFPGA، معماری های مرسوم، مانند بلوک سوئیچ (SB)، بلوک اتصال (CB) و حافظه بلوک، توسط RRAM جایگزین شده اند. همانطور که در شکل 28نشان داده شده است، T1R2 (دو ترانزیستور و یک مقاومت) RRAM برای جایگزینی با سوئیچ های معمولی قابل برنامه نویسی در CB و SB استفاده می شوند و T1R RRAM1 به جای سلول T SRAM6 در حافظه بلوک استفاده می شود. نتایج شبیه سازی نشان

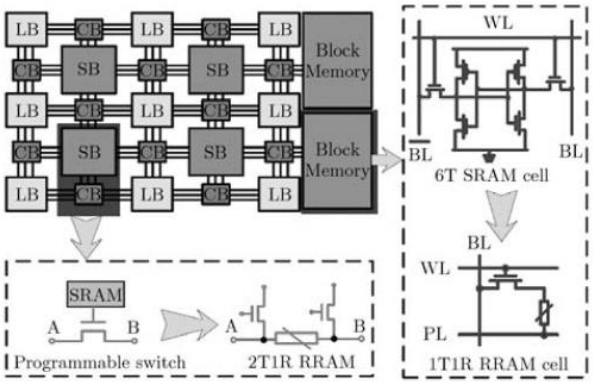
می دهد که پیچیدگی منابع مسیریابی می تواند کاهش یابد و با استفاده از RRAM برای تحقق بخشیدن به CB و SB، تاخیر کانال 2 ~ 3 برابر کاهش می یابد. مساحت بلوک حافظه 5 ~ 6 برابر با استفاده از RRAM کاهش خواهد یافت.

جدول. 2 مقایسه فن آوری های حافظه جاسازی شده [20،78،87،93]

Device type	Cell structure	Cell components*	Minimum cell size (F^2, F=feature size)	Incremental masks	Read/write speed (ns)	Endurance (Cycles)
SRAM	Latch	6T	24	0	1/1	Infinite
Flash	Floating	1T-2T	8-12	6-8	25/200	10^4~10^5
PCRAM	Phase	1T1R	6-10	3-4	20/50	$>10^{12}$
STTRAM	Magneto	1T1MTJ	6-10	3-4	10/10	$>10^{15}$
RRAM	Memristor	1T1R	4	3-4	10/20	10^5~10^6

*T = transistor; C = capacitor; R = resistance/phase-change element

معماری بهبود یافته به دلیل به کارگیری تعداد کمتر از ترانزیستورها، نیز می تواند مصرف برق را کاهش دهد. پژوهش های دیگر در مورد بهبود سطح ادغام بر روی جایگزینی های RRAM 2-D با 3D RRAM [91] متمرکز شده اند.



شکل. 28 FPGA با استفاده از اجزای RRAM به عنوان عناصر حافظه و مسیریابی [92]

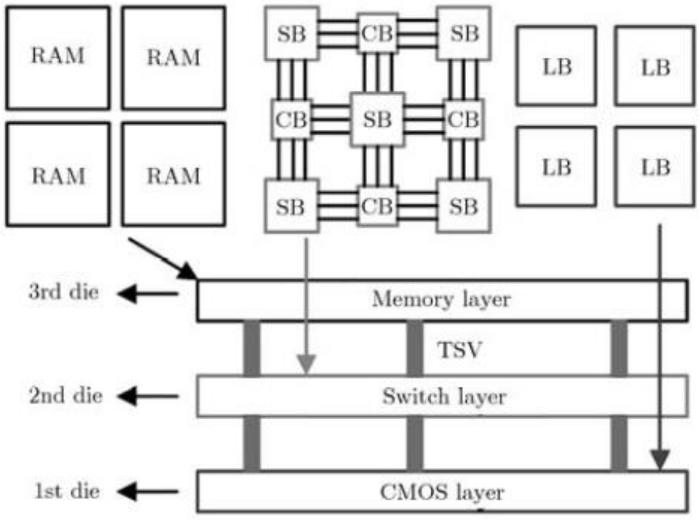
(4) خلاصه

جدول 2، خلاصه ویژگی های ارائه شده از فن آوری های حافظه در حال ظهور را همراه با SRAM و فلش برای اهداف مقایسه گنجانده است. اندازه های سلول از تمام فن آوری های حافظه در واحدهای حداقل در اندازه ویژگی F بر اساس فصل دستگاه های پژوهشی در حال ظهور (ERD) نقشه راه تکنولوژی بین المللی سال 2007 برای نیمه هادی ها (ITRS) پیش بینی شد که حاوی جدولبندی مقادیر تجربی اخیر گزارش شده در نوشته هاست.

3D FPGA .2

3D FPGA، که نشان دهنده روند مهم FPGA در حال توسعه دیگری است، بر اساس فن آوری های مداری سه بعدی است. زمانی که پیچیدگی تراشه FPGA افزایش یابد، مشکلاتی مانند تاخیر اتصال، سنکرون سازی ساعت و افت IR در شبکه برق شدید تر از همیشه شده است. فن آوری 3D بر اساس Through Silicon Vias (TSVs) به عنوان امیدوار کننده ترین نامزد برای حل چنین مشکلاتی در نظر گرفته می شود.

M.Lin از دانشگاه استنفورد یک ساختار بهینه از 3D FPGA بر اساس تراشه های پشته شده و Through Silicone Vias (TSVs) به صورت کانال های اتصال داخلی پیشنهاد کرده است. شکل. 29 ساختار معمولی 3D FPGA را نشان می دهد که FPGA سنتی دوبعدی را به سه لایه گسترش می دهد: حافظه در بالا، سوئیچ در وسط و منطق در لایه بستر. مزیت این ساختار، مجتمع سازی آسان برای منطق و SRAM IP است. در مقایسه با Virtex-II، این 3D FPGA پشته شده دارای 3.2 برابر مجتمع سازی تراکم منطقی با 31٪ مساحت کاهش یافته است.

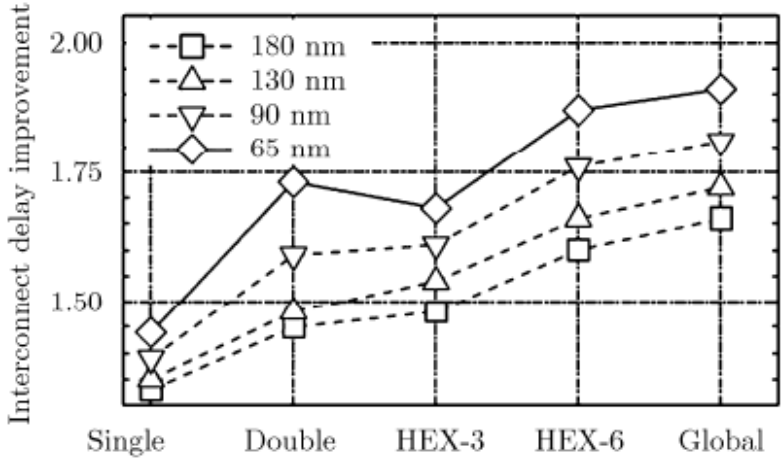


شکل. 29 3D-FPGA پشته بندی شده یکپارچه

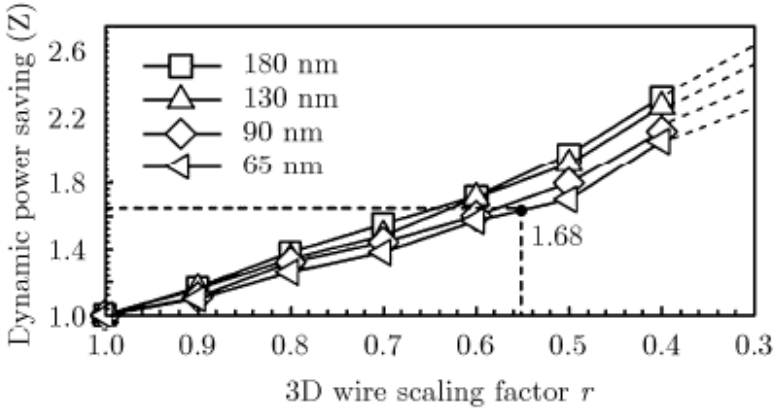
برآورد 3D FPGA برای هر یک از چهار گره فن آوری (180 نانومتر، 130 نانومتر، 90 نانومتر، و 65 نانومتر) در ITRS [93] با استفاده از (VPR) همه کاره ، شکل 30 را ببینید، نشان می دهد که متوسط هندسی تاخیر پین به

پین در محدوده بهبود یافته بین 1.7 و 2.05 قرار دارد، و برای تاخیر مسیر بحرانی بین 1.31 و 2.14 است و 1.68 بیش از یک D FPGA2 معمولی، در مصرف برق دینامیک صرفه جویی می نماید.

اگر چه 3D FPGA دارای مزیت قریب به اتفاق در مقایسه با FPGA دوبعدی معمولی است، پایداری کمتر در روند ساخت 3D و عدم ابزار EDA حمایتی، مانع این کاربرد می شود. خوشبختانه، همراه با تکنیک 3D در حال توسعه و ابزار مکان یابی و مسیریابی گزارش شده در 3D FPGA [94]، پیش بینی شده است که 3D FPGA احتمالا می تواند در 5 سال آینده صنعتی شود.



(a) Interconnect delay (One (referred to as Single), two (Double), three (HEX-3), and six (HEX-6) are sets of FPGA tile width segments in addition to interconnects that span the entire array width (Global))



(b) Relative power saving (Factor $0<r<1$ represents the ratio of the 3D-FPGA tile width to the baseline 2D-FPGA)

شکل. 30 بهبودهایی برای گره های مختلف فن آوری زیر میکرون [95]

7. بحث کوتاهی در مورد توسعه حال حاضر تحقیقات FPGA در چین

به طور کلی، تحقیق و توسعه FPGA در چین در فاز آغازین قرار دارد [96-102]. با توجه به طراحی معماری، برخی تلاش ها شایسته و در خور ذکر است. به عنوان مثال، دانشگاه Fudan یک سیستم ارزیابی برای معماری FPGA را بر اساس ابزار VPR، به منظور برآورد اطلاعات قدرت، مساحت و زمان بندی توسعه داده است [103]. یک روش تاخیر-تخمینی بر اساس مدل آماری نیز پیشنهاد شده است که به بهینه سازی معماری اتصال FPGA کمک می کند [104]. به منظور بهبود انعطاف پذیری منابع مسیریابی، یک جعبه سوئیچ جدید، به نام جعبه سوییچ Minimum-Loop Maximization (MLM) با حداثر نمودن اندازه حلقه مینیمم در گراف منبع-مسیریابی توسعه داده شد.

FPGA های سریال "Comet" [106 107] توسط System on Programmable Chip Research Department of Institute of Electronics, Chinese Academy of Sciences توسعه یافته است و پژوهش جامع برای معماری های FPGA بر روی این پلت فرم انجام شد. از طریق روش های تجربی، رابطه بین انعطاف پذیری اتصال (FC) یک FPGA عمومی و عملکرد آن، مورد بررسی قرار گرفتکه موجب بهینه سازی انعطاف پذیری معماری FPGA [108] می شود. علاوه بر این، قابلیت مسیریابی جعبه سوئیچ و جعبه اتصال، مدلسازی شد و روش بازپخت شبیه سازی شده برای به حداکثر رساندن آنتروپی اطلاعات توزیع سوئیچ به منظور بهبود قابلیت مسیریابی [109] استفاده شد. این دستاوردهای پژوهشی با موفقیت برای تراشه FPGA نگا-گیت "Comet 02" به کار برده شده اند که اولین محصول داخلی FPGA در مقیاس بزرگ برای هوا فضا یا ماهواره است.

8. نتیجه گیری

هدف از این مقاله، ارائه بینشی جامع به طراحی معماری پیشرفته FPGA است. فن آوری FPGA تا حد زیادی توسط توسعه سریع تکنولوژی نیمه هادی پیشرفت داده شده است، در حالی که در همان زمان با توجه به اثرات نامطلوب ناشی از فرآیند زیر میکرون و نانو متر عمیق، طراحان باچالش های جدید روبرو هستند. همگام با خواسته های بازار امروز، ادوات FPGA به سمت سطح بالایی از عملکرد، یکپارچه سازی سازی سیستم و پهنای باند توسعه

یافته است. علاوه بر این، با افزایش میزان و بهبود عملکرد ماژول های تابعی، FPGA به یک سیستم فعال دستگاه ضروری تبدیل شده است و ریزپردازنده ها در ادوات FPGA برای ارائه یک پایه پلت فرم جامع برای سیستم های نسل بعدی ادغام شد اند. برای چشم اندازهای آینده، کاملا امیدوار کننده است که طراحی FPGA ها فن آوری هیجان انگیز و پویا باقی خواهد ماند ، و فن آوری های در حال ظهور می توانند به صورت ترکیبی با هم فرصت های فوق العاده ای را به منظور توسعه آینده سیستم عامل FPGA بر روی تراشه ارائه دهند.

References

[1] J. J. Rodriguez-Andina, M. J. Moure, and M. D. Valdes. Features, design tools, and application domains of FPGAs. *IEEE Transactions on Industrial Electronics*, **54**(2007)4, 1810–1823.

[2] M. Slimane-Kadi, D. Brasen, and G. Saucier. A fast-FPGA prototyping system that uses inexpensive high-performance FPIC. In Proceeding of 2nd Annual Workshop on FPGAs, Berkeley, CA, USA, 1994, 1–6.

[3] S. M. Trimberger. Field-Programmable Gate Array Technology. USA, Kluwer, 1994, Chapters 1–4.

[4] E. Ahmed and J. Rose. The effect of LUT and cluster size on deep-submicron FPGA performance and density. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, **12**(2004)3, 288–298.

[5] Xilinx. Virtex-6 Family Overview. DS150 (v1.0), February 2, 2009.

[6] V. Betz, J. Rose, and A. Marquardt. Architecture and CAD for Deep-Submicron FPGAs. Berlin, Kluwer Academic Publishers, 1999, Chapters 1–5.

[7] Altera Corporation. Stratix IV Device Handbook. March 2009.

[8] Altera Corporation. Architecture Matters: Choosing the Right SoC FPGA for Your Application. November 2013.

[9] Xilinx. Zynq-7000 All Programmable SoC Overview. March 2013.

[10] Microsemi. SmartFusion2 System-on-Chip FPGAs. January 2014, Revision 5.

[11] E. Monmasson, L. Idkhajine, and M. Cirstea, *et al.*. FPGAs in industrial control applications. *IEEE Transactions on Industrial Informatics*, **7**(2011)2, 224–243.

[12] Altera Corporation. Arria 10 Device Handbook. June 2013.

[13] Xilinx. Xilinx UltraScale Architecture for High-Performance, Smarter Systems. December 2013.

[14] Xilinx. Xilinx UltraScale: The Next-Generation Architecture for Your Next-Generation Architecture. July 2013.

[15] Altera Corporation. Expect a Breakthrough Advantage In Next Generation FPGAs. June 2013.

[16] I. Kuon, R. Tessier, and J. Rose. FPGA architecture: survey and challenges. *Electronic Design Automation*, **2**(2007)2, 135–253.

[17] Microsemi. ProASIC3 Flash Family FPGAs with Optional Soft ARM Support. January 2013, Revision 13.

[18] Microsemi. IGLOO2 FPGAs. June 2013.

[19] Microsemi. Axcelerator Family FPGAs. Revision 18, March 2012.

[20] I. Kuon and J. Rose. Measuring the gap between FPGAs and ASICs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **26**(2007)2, 203–215.

[21] Yang Hai-gang, Sun Jia-bin, and Wang Wei. An overview to FPGA device design technologies. *Journal of Electronics and Information Technology*, **32**(2010)3, 714–727.

[22] A. M. Smith, G. A. Constantinides, and P. Y. K. Cheung. FPGA architecture optimization using geometric programming. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **29**(2010)8, 1163–1176.

[23] F. Barranco, M. Tomasi, J. Diaz, et al.. Architecture for hierarchical optical flow estimation based on FPGA. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 20(2012)6, 1058–1067.

[24] E. Ahmed and J. Rose. The effect of logic block granularity on deep-submicron FPGA performance and density. Master's thesis, University of Toronto, Department of Electrical and Computer Engineering, 2001.

[25] R. Tessier, V. Betz, D. Neto, and T. Gopalsamy. Power-aware RAM mapping for FPGA embedded memory blocks. In Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays, Monterey, CA, USA, 2006, 189–198.

[26] Altera Corporation. Stratix Device Handbook. January 2006.

[27] Altera Corporation. Cyclone Device Handbook. May 2008, ver. 3.1.

[28] Altera Corporation. Cyclone II Device Handbook. February 2008, ver. 4.0.

[29] Altera Corporation. Cyclone III Device Handbook. August 2012, ver. 4.2.

[30] Altera Corporation. Cyclone IV Device Handbook. February 2013, ver. 1.8.

[31] Altera Corporation. Stratix V Device Handbook. June 2013.

[32] Altera Corporation. FPGA Architecture. July 2006, ver. 1.0.

FPGA soft core. 19th International Conference on VLSI Design and Held jointly with 5th International Conference on Embedded Systems and Design, India, 2006, 1–5.

[40] I. Kuon and J. Rose. Measuring the gap between FPGAs and ASICs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **26**(2007)2, 203–215.

[41] Michael John and Sebastian Smith. Application-specific integrated circuits. US, Electronic Industry Press, January 2003.

[42] A. Mishchenko, S. Chatterjee, and R. Brayton. DAG-aware AIG rewriting: a fresh look at combinational logic synthesis. In Proceedings of the 43rd Design Automation Conference, San Francisco, CA, USA, July 2006, 532–536.

[43] Hadi Parandeh-Afshar, Hind Benbihi, *et al.*. Rethinking FPGAs: elude the flexibility excess of LUTs with and-inverter cones. ACM/SIGDA 20th International Symposium on Field Programmable Gate Arrays, Monterey, CA, US, February 2012, 119–128.

[44] D. Lewis, D. Cashman, Mark Chan, *et al.*. Architectural enhancements in Stratix-V™. ACM/SIGDA 21st International Symposium on Field Programmable Gate Arrays, Monterey, CA, US, February 2013, 147– 156.

[45] Altera Corporation. Stratix III FPGAs vs. Xilinx Virtex-5 Devices: Architecture and Performance

[33] Xilinx. Virtex-5 FPGA User Guide. March 2012, ver. 5.4.

[34] Xilinx. UltraScale Architecture Configurable Logic Block. December, 2013, ver. 1.0.

[35] Xilinx. 7 Series FPGAs Configurable Logic Block. August 2013, ver. 1.5.

[36] V. Betz and J. Rose. Cluster-based logic blocks for FPGAs: area-efficiency vs. input sharing and size. IEEE Custom Integrated Circuits Conference, Santa Clara, CA, USA, 1997, 551–554.

[37] E. Ahmed and J. Rose. The effect of LUT and cluster size on deep-submicron FPGA performance and density. In Proceedings of the 2000 ACM/SIGDA Eighth International Symposium on Field Programmable Gate Arrays, USA, ACM Press, 2000, 3–12.

[38] R. Lysecky and F. Vahid. A study of the speedups and competitiveness of FPGA soft processor cores using dynamic hardware/software partitioning. Design, Automation and Test in Europe, Munich, Germany, 2005, 18–23.

[39] P. Biswas, S. Banerjee, and N. Dutt. Performance and energy benefits of instruction set extensions in an

Comparison. September 2007, ver. 2.0

[46] Xilinx. Advantages of the Virtex-5 FPGA 6-Input LUT Architecture. December 2007, ver. 1.0.

[47] V. George and J. M. Rabaey. Low-Energy FPGAs: Architecture and Design.US, Kluwer Academic Publishers, US, 2001.

[48] Altera Corporation. FLEX 10K embedded programmable logic device family. DS-F10K-4.2 January 2003.

[49] Altera Corporation. APEX 20K programmable logic device family data sheet. DS-APEX20K-5.1, March 2004.

[50] Altera Corporation. APEX II programmable logic device family. DSAPEXII-3.0, August 2002.

[51] V. Betz and J. Rose. FPGA routing architecture: Segmentation and buffering to optimize speed and density. ACM/SIGDA International Symposium on Field Programmable Gate Arrays, Monterey, CA, US, February 1999, 140–149.

[52] Xilinx Inc.. The Programmable Logic Data Book. 1994.

[53] G. G. Lemieux and S. D. Brown. A detailed router for

allocating wire segments in field-programmable gate arrays. ACM Physical Design Workshop, Reston Sheraton, US, April 1993, 1–8.

[54] Y. W. Chang, D. Wong, and C.Wong. Universal switch modules for FPGA design. *ACM Transactions on Design Automation of Electronic Systems*, **1**(1996)1, 80–101.

[55] S. J. E. Wilton. Architectures and algorithms for field-programmable gate arrays with embedded memory. Ph.D. thesis, University of Toronto, 1997.

[56] M. I. Masud and S. Wilton. A new switch block for segmented FPGAs. International Workshop on Field Programmable Logic and Applications, US, August 1999, 274–281.

[57] H. Fan, J. Liu, Y. L. Wu, and C. C. Cheung. On optimal hyperuniversal and rearrangeable switch box designs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **22**(2003)12, 1637–1649.

[58] G. Lemieux and D. Lewis. Design and Interconnection Networks for Programmable Logic. Boston, MA, US, Kluwer Academic Publishers, 2004.

[59] D. Marple and L. Cooke. An MPGA compatible FPGA architecture. IEEE Custom Integrated Circuits Conference, Boston, MA, US, 1992, 421–424.

[60] G. Lemieux and D. Lewis. Analytical framework for switch block design. International Symposium on Field Programmable Logic and Applications, Mon-

[66] P. Jamieson, W. Luk, *et al.*. An energy and power consumption analysis of FPGA routing architectures. International Conference on Field-Programmable Technology (FPT), Sydney, Australia, December 2009, 324–327.

[67] G. Lemieux and D. Lewis. Directional and single-driver wires in FPGA interconnect. ACM/SIGDA International Symposium on Field Programmable Gate Arrays, Monterey, CA, US, December 2004, 41–48.

[68] D. Lewis, *et al.*. The Stratix II logic and routing architecture. In Proceeding: ACM/SIGDA International Symposium on Field Programmable Gate Arrays, Monterey, CA, US, Februery 2005, 14–20.

[69] John Teifel and Rajit Manohar. An Asynchronous Dataflow FPGA Architecture. *IEEE Transactions on Computers*, **53**(2004)11, 1376–1392.

[70] Achronix Corporation. ACE User Guide. December 2012, ver. 5.0.

[71] Achronix Corporation. Introduction to Achronix FPGAs. August 2008, Rev. 1.6.

[72] Xilinx. Xilinx Stacked Silicon Interconnect Technology Delivers Breakthrough FPGA Capacity, Bandwidth, and Power Efficiency. December 2012, ver. 1.2.

[73] L. Madden, E. Wu, *et al.*. Advancing high performance heterogeneous integration through die stacking. European Solid-State Circuits Conference (ESSCIRC), Bordeaux, France, September 2012, 18–24.

terey, CA, US, September 2002, 122–131.

[61] M. Lin, A. El Gamal, Y. C. Lu, and S. Wong. Performance benefits of monolithically stacked 3D FPGA. *IEEE Transaction on Computer -Aided Design Integrated Circuits and System*, **26**(2007)2, 216–229.

[62] A. El Gamal, J. Greene, V. Roychowdhury, *et al.*. Segmented channel routing in nearly as efficient as channel routing (and just as hard). University of California/Santa Cruz Conference, Berkeley, CA, US, 1991, 192–211.

[63] W. K. Mak and D. F. Wong. Channel segmentation design for symmetrical FPGAs. International Conference on Computer-aided Design, Kasuga, Japan, 1997, 496–501.

[64] C. G. Wong, A. J. Martin, and P. Thomas. An architecture for asynchronous FPGAs. IEEE International Conference on Field-Programmable Technology (FPT), HK SAR China, 2003, 170–177.

[65] David Lewis, Vaughn Betz, *et al.*. The StratixTM routing and logic architecture. ACM/SIGDA International Symposium on Field Programmable Gate Arrays, Monterey, CA, US, February 2003, 41–48.

[74] Kurt Keutzer, Sharad Malik, A. Richard Newton, *et al.*. System-level design: orthogonalization of concerns and platform-based design. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **19**(2000)12, 1523–1543.

[75] Xilinx. Virtex-4 FPGA User Guide. UG070 (v2.6), December 2008.

[76] Xilinx. Virtex-4: Breakthrough Performance at the Lowest Cost. December 2004.

[77] Xilinx. Virtex-5 Platform FPGA Family Technical Backgrounder. May 2006.

[78] S. Natarajan, S. Chung, L. Paris, and A. Keshavarzi. Searching for the dream embedded memory. *IEEE Solid-State Circuits Magazine*, **1**(2009)3, 34–44.

[79] Y. Guillemenet, L. Torres, and G. Sassatelli. Non-volatile run-time field-programmable gate arrays structures using thermally assisted switching magnetic random access memories. *IET Computer Digital Technology*, **4**(2010) 3, 211–226.

[80] S. Paul, S. Mukhopadhyay, and S. Bhunia. Circuit and architecture co-design approach for hybrid CMOS-STTRAM non-volatile FPGA. *IEEE Transactions on. Nanotechnology*, **10**(2011)3, 385–394.

[81] Jason Cong and Bingjun Xiao. FPGA-RPI: A novel FPGA architecture with rram-based programmable interconnects. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, **22**(2014)4, 864–877.

[82] S. Tanachutiwat, M. Liu, and W. Wang. FPGA based on integration of CMOS and RRAM. *IEEE Transactions on Very Large Scale Integrity System*, **19** (2011)11, 2023–2032.

[83] O. Turkyilmaz, J. Figueras, and Y. Zorian. RRAM-based FPGA for ACM normally off, instantly on ACM applications. International Symposium on Nanoscale Architecture, US, 2012, 101–108.

[84] A. Pirovano. Electronic switching effect in phase-change memory cells. IEEE International Electron Devices Meeting, Washington, D.C., US, 2005, 923–926.

[85] R. Zhao and L.P. Shi. Study of phase change random access memory (PCRAM) at the nano-scale. Non-Volatile Memory Technology Symposium, Albuquerque, New Mexico, US, 2007, 36–39.

[86] M. Hosomi. A novel nonvolatile memory with spin torque transfer magnetization switching: Spin-RAM. International Electron Device Meeting, Washington, D.C., US, 2005, 473–476.

[87] Somnath Paul and Swarup Bhunia. A circuit and architecture codesign approach for a hybrid CMOS-STTRAM nonvolatile FPGA. *IEEE Transactions on Nanotechnology*, **10**(2011)3, 385–394.

[88] W. Zhao, E. Belhaire, Q. Mistral, E. Nicolle, T. De-

national Symposium on Circuits and Systems (ISCAS), France, July 2010, 1428–1431.

[93] G. Hariharan, R. Chaware, L. Yip. *et al.*. Assembly process qualification and reliability evaluations for heterogeneous 2.5D FPGA with HiCTE ceramic. Electronic Components & Technology Conference, Las Vegas, NV, USA, November 2013, 904–908.

[94] C. Ababei, H. Mogal, and K. Bazargan. Three-dimensional place and route for FPGAs. IEEE Asia and South Pacific Design Automation Conference, Shanghai, China, January 2005, 773–778.

[95] Mingjie Lin, A. El Gamal, Yi-Chang, *et al.*. Performance benefits of monolithically stacked 3D FPGA. *IEEE Transactions on Computer Aided Design of Integrated Circuit and Systems*, **26**(2007)2, 216–229.

[96] Xie Ding, Lai Jinmei, and Tong Jiarong. Research of efficient utilization routing algorithm for Current FPGA. *Chinese Journal of Electronics*, **19**(2010)1, 48–52.

[97] Zhu Limin, Bian Jinian, Zhou Qiang, *et al.*. Integer programming based routing algorithm fro hierarchical FPGAs. *Journal of Computer-Aided Design & Computer Graphics*, **22**(2010)10, 1687–1694.

[98] Zhang Kun, Zhou Huabing, Stanley L. Chen, *et al.*. Technology mapping for FPGA with multi-mode logic cell. *Journal of Computer-Aided Design & Computer Graphics*, **21**(2009)10, 1375–1380.

[99] Zhang Huiguo, Tang Yulan, Yu Zongguang, and Tao Yufeng. High performance FPGA LUT design and implementation. *Research & Progress of SSE*, **29** (2009)4, 584–588.

volder, and C. Chappert. Integration of spin-RAM technology in FPGA circuits. International Conference Solid-State Integrated Circuit. Technology, US, 2006, 799–802.

[89] N. Bruchon, L. Torres, G. Sassatelli, *et al.*. New non-volatile FPGA concept using magnetic tunneling junction. IEEE Computing Society Annie Symposium Emery on VLSI Technology Architecture, US, 2006, 269–276.

[90] M. Liu and W. Wang. Rfpga: CMOS-Nano hybrid FPGA using RRAM components. IEEE International Symposium on Nanoscale Architectures, Anaheim, CA, US, June 2008, 93–98.

[91] J. Zhang, Y. Q. Ding, *et al.*. A 3D RRAM using stackable 1TXR memory all for high density application. International Conference on Communications, Circuits and Systems, CA, US, July 2009, 917–920.

[92] Hai-Gang Yang. Overview: Emerging technologies on giga-scale FPGA implementation, 2010 IEEE International Conference on Field-Programmable Logic and Applications, Greece, 2011, 447–452.

[100] Xu Hanyang, Wang Jian, and Lai Jinmei. A FPGA prototype design emphasis on low power technique. In Proceedings of the 2014 ACM/SIGDA 22nd International Symposium on Field Programmable Gate Arrays, Monterey, CA, US, February 2014, 147–150.

[101] Gao Haixia, Yang Yintang, and Dong Gang. Theoretical analysis of effect of LUT size on area and delay of FPGA. *Chinese Journal of Semiconductors*, **26** (2005)5, 893–898.

[102] Gao Haixia, Ma Xiaohua, and Yang Yintang. Accurate interconnection length and routing channel width estimates for FPGAs. *Chinese Journal of Semiconductors*, **27**(2006)7, 1196–1200.

[103] Chen Yuanfeng, Tang Pushan, Lai Jinmei, *et al.*. Evaluation System for FPGA. *Journal of Fudan University* (*Natural Science*), **45**(2006)4, 523–528.

[104] Zhen Wang, Ding Xie, Jinmei Lai, *et al.*. FPGA interconnect architecture exploration based on a statistical model. International Conference on Field-Pro-

[105] Yu Jiande, Xie Ding, Shao Yun, *et al.*. Mini-loop maximization method on FPGA routing resources architecture design. *Journal of Computer-Aided Design & Computer Graphics*, **22**(2010)6, 934–942.

[106] System on Programmable Chip Research Department of Institute of Electronics, Chinese Academy of Science. COMET02 Device Handbook. January 2010, Ver. 1.1.

[107] System on Programmable Chip Research Department of Institute of Electronics, Chinese Academy of Science. ER2C1000-G Handbook. June 2013, Ver. 1.0.

[108] Xingzheng Li, Haigang Yang, and Hua Zhong. Use of VPR in design of FPGA architecture. International Conference on Solid-State and Integrated Circuit Technology, Shanghai, China, 2006, 1880–1882.

[109] Li Wei, Yang Haigang, and Gong Xiao. Optimal design of topological structure for FPGA connection box based on information entropy. *Journal of Computer-Aided Design & Computer Graphics*, **21**(2009)2, 203–208.