

تک فرم به اسم General Estrin یک عاری ارائه داد به اسم Fix-plus
این عاری در سال ۱۹۵۹ ارائه شد و عاری که ارائه داد یک عاری مثل باز سیم بندی بود یعنی حساسیت آن را
ساختار را بصورت رانکس داد.

صاحب کار تصویر را میسوزاند.
 - ایده اش این بود که تلفت من یعنی دارم به یک بخش Fine & part (بخش UL «F» در شکل
 general persons -
 میت راست این اسلاید (که چون تصویر من میسوزانست) و از IBM main frame استفاده می‌کنه

- یک بخش variable part دارد که در اصل بیت راست صفت یا سبج «ن» است. یک بقایای

مادرول های طبیعی با سرعیت بالا برای منظورهای خاص فراهمی کرده بود (سخت افزای خاص منظور ۱۳۸۵)

و چون به ملاحظه برای خون کار ایاد شده سرعش بالاست و در شکل جسم می بینیم که از حفره های سرع و external

backup ایسا دوسرا

به دنبال می‌نماید. بسیاری از داروهای ترکیبی، بسیاری از حفظه‌های خارجی، داخلی دارو

۱- کنترل نظارتی (supervisory control) SC

speed gain این سیستم درآسیب IBM 7090 ایجاد می کرد ، ۲۱۵ برابر یا ۱۰۰۰ برابر سرعت

چون برای کار و صفا و انکسار حای خدای استفاده هست
این رنج هر چه که است

این رنج بر عودا صحت است

اسلامی مرکز

IBM 7090 Fixed part ۸۵٪

special purpose substructure (variable part)

variable part -
صداحت، توافق، مطابقت، تابع، کثرت، به توان، رساندن، جذب، مرکب، محاسبات، اعداد، محلول، توافق، hyperbolic

و علیات ماسرینج و ... بود برای هر کدام از این علیات یک نرد جدا طراحی می کرد و بسته
در جدا طراحی گرفت و با سبک و بالایی نسبت به 90 I Baud می نوشت انجام دده

سپیس Fix-plus Estrin و هر دو دستبند این Reconfiguration انجام میدهد و قابل جابجایی
 اصل هر دو یک نرد را وارد می کند و یکی در یک خارج می کند از مدار و آن وقت آن functional unit
 Year: _____ Month: _____ Day: _____ Subject: _____
 می تواند برای آن عملیات را می داد و نرد تا عملیات شروع می انجام می دهد. این چیزی است که در
 قابلیت بازنگری برای سیستم ایجاد می کند

بعد از آن در سال ۱۹۷۷ Rammig machine توسط Rammig ارائه شد که نخستین این نور بدون
 دخلت دست کار برنامه ریزی انجام میدهد یعنی به جای این که دست انجام می دهد سیستمی باشد که هر دو استر و
 کار برنامه ریزی انجام می دهد که این نداشتن Hardware Editor

Hardware Editor (نیمه انفرای که می تواند Edit شود).
 برای این که این کار انجام می شود مطمئن باین این اسلايد صحبت می کند. general purposes کامپیوتر است
 و اما Fix است که بخش Fix-part داشت اینها هم یک کامپیوتر تمام منظوره است که از PDP11
 استفاده می کرد برای این کار که این کامپیوتر کارهای عمومی را انجام می دهد و برای کارهای خصوصی باز
 library board وجود دارد که هر یکی از کارهای خاص منظوره را انجام می دهد و یک بخش دارد
 برنامه ریزی crossbar که هست که قابل برنامه ریزی است و اتصالات را می تواند برنامه ریزی
 می کند و در ردی نام می برد و می تواند نرد در مدارها را کامپیوتر قرار دهد
 این تفاوت این سیستم با سیستم Fixed-plus این است که بخش crossbar و هر دو استر و قابل
 برنامه ریزی است و برای این که از باینری را تغییر می کند که رشته ای از ۱ و ۰ ها است که به معنی
 اتصال می تواند قرار داشته و معنی اتصال یا بر قطع باشد که یک دنباله ای از ۱ ها و ۰ ها می تواند
 تا crossbar را تغییر برنامه ریزی کند (bit stream این دنباله ای از ۱ ها و ۰ ها است که از
 یک برنامه ریزی می کند و از این طریق selectable می تواند ارتباط قرار می دهد)

یک نمونه از نرد سیستم Rammig machine که هر دو یک عملیات کامپیوتری انجام می دهد
 این Rammig برای کارهای emulation خطی استفاده می کند (یعنی می خواهم یک انفرایب کامپیوتر را شبیه سازم
 CACTUS

در سطحی simulate, emulate کنیم

تفاوت simulate و emulation را با هم از نظر انجام کار می بینیم و می بینیم که در هر دو

Year: ... Month: ... Day: ... Subject: ...

emulation یعنی simulation به صورت سخت افزاری یعنی سخت افزار ورودی می شود و خروجی می دهد و

FPGA را با emulation انجام میدهند

granularity (دانه بندی)

granularity یعنی مقدار بلوک های reconfigurable چقدر درشت یا ریز باشد یعنی سیستم درشت دانه

است یا ریز دانه یا سبک دانه

اسلاید ۲۰

تقسیم بندی granularity

Boolean level - (ریز دانه یا Fine)

instruction level -

function level -

process level - (درشت دانه یا coarse grain)

از ریز به درشت

به عبارت دیگر granularity دانه بندی سطح abstraction نشان می دهد که برای configure کردن device از چه اندازه ای استفاده می کنیم مثلاً در boolean level سطح abstraction در درجه boolean است یا در instruction level سطح abstraction در درجه instruction است و ...

اسلاید ۲۱

Boolean level - منظور همین FPGA های امروزی است که آن دانه ای که عموماً را انجام میدهد look up table (LUT) ها هستند. این LUT ها کارشون اینست که کارهای منطقی را با داده سازی می کنند مثلاً برای عملیات ضرب یک تعدادی از این LUT ها با هم وصل کنیم که عملیات ضرب را انجام دهد. کارهای اساسی منطقی همین توابع انجام بدن این boolean level و کارهایی که می توان انجام داد سبک و ساده هستند

این Architecture ها fine grain ترین کسب ویش برای کارهای boolean ساده محسوب می شوند

CACTUS خاص هستند و با هم LUT انجام می دهند و اگر تعداد زیادی ورودی داشته باشند LUT ها را به هم وصل می کنند و به صورت efficient کار می کنند ولی برای عملیات های پیچیده مثل multiplier

efficient نیست

Instruction level : در سطح instruction (دستورالعمل) عمل می کنند. بنابراین واحدهای کامپیوتری دارند که هر واحد فایده خاصی در instruction operation را انجام می دهد. این واحدها در ردی های خاص و در صورت **word-width** است یعنی ۸ بیتی ۳۲ بیتی و... حتماً یعنی **data path** باشد زیاد انجام می دهند. عملیات بیتی و برای عملیات Boolean مناسب نیستند.

Functional level : عملیاتی که انجام می دهند عملیات **multi cycle** است. عملیات جمع و تفریق را به واسطه یک برنامه انجام می دهد. مثلاً یک سری processor ها هستند اسم **Extensible process** که شروع اینها به بیان اینست که ما از آنها استفاده می کنیم و مثلاً می بینیم که اینها یک حلقه دارند که ما می توانیم از آنها استفاده کنیم. در اینجا ما دستورالعمل **Add, compare, mult** و **Add** است که خیلی تر از آنست که برای اینها **instruction** درست می کنند و بعد یک **Functional unit** برای این **instruction** قرار می دهند که این عملیات را انجام دهد و در نتیجه عملیاتی که در چند سیکل انجام می شد در یک سیکل انجام می شود.

process level : عملیات در چند سیکل یا چند سیکل انجام می شود و سیستم **reconfigurable** که ما می توانیم از آنها استفاده کنیم و می توانیم از آنها استفاده کنیم. مثلاً ما می بینیم که اینها یک حلقه دارند که ما می توانیم از آنها استفاده کنیم. قابلیت بازیکردن هم دارد و عملیاتی که در اینها انجام می شود برای انجام در واحدهای دیگر مناسب نیستند.

اسلاید ۷ تا ۹

در **Fine grain (boolean level)** کار **Configuration** زیاد است مثلاً چند میلیون و ۱ دارند در حالی که در **Coarse grain (process level)** کار کمتر است مثلاً باریت قابل انجام است. بنابراین اگر ما می خواهیم یک سیستم بریم سراغ **Fine grain** ها یا **Coarse grain** ها یا چاشنی؟ (در عمل کدام جواب تر هست؟)

برای این که تصمیم بگیریم کدام یکی از اینها را استفاده کنیم نگاه می کنیم به اینست که ما می بینیم که اینها یک حلقه دارند که ما می توانیم از آنها استفاده کنیم. آن اینست که ما می بینیم که اینها یک حلقه دارند که ما می توانیم از آنها استفاده کنیم. **Instruction level** حتماً بهتر است بریم سراغ **boolean level** و **word size** و **Boolean level** و... مثلاً اینها هم **DSP** ها و **word size** و **Boolean level** و... **Instruction level** حتماً بهتر است بریم سراغ **boolean level** و **word size** و **Boolean level** و...

CACTUS

"Hybrid Device"

اسلام کا

« Hybrid Device »
 : Jot, min boolean level \rightarrow FPGAs Device
 : Lut -

6 Lut -

Subject:

Year: Month: Day:

adder/multiplier

Loop -

microprocessor-

این امر در device ها hybrid است یعنی در granularity کمتر عمل می کنند
مثلا DSP هست یک بخش است و ... چون امروزه خیلی از اینها را دارند و بزرگای آنها

اسلام 11

FPGA و CGRA به سبب دانه ریز بودن fine grain و به سبب دانه درشت بودن coarse grain

جواب C GRA حاصلی سے برآز Fp GA حاصہ ہے۔

این دو طول Configuration خطی می باشد دارند و چون CGRA ها طول برنامه ریزی نمی کنند
دارند سرعت تستی دارند FPGA ها که طول برنامه ریزی می کنند نمی دارند

312 ۱۱۱

8. ASIC & FPGA, CARA و

صفحه ۱۱
۱۱ FPGAs حاوی منطق مورد نیاز برای یک مدار است و در مقابل تراشه‌های ASIC که برای یک کاربرد خاص طراحی شده‌اند، به دلیل انعطاف‌پذیری و قابلیت برنامه‌ریزی مجدد، در بسیاری از موارد، به دلیل هزینه کمتر و زمان عرضه کوتاه‌تر، گزینه مناسب‌تری هستند. همچنین، FPGAs به دلیل قابلیت برنامه‌ریزی مجدد، می‌توانند برای تست و توسعه سریع مدارها استفاده شوند. در حالی که ASICها معمولاً برای تولید انبوه و در مواردی که نیاز به عملکرد بالا و بهینه‌سازی دقیق است، استفاده می‌شوند. بنابراین، انتخاب بین این دو فناوری بستگی به نیازهای پروژه دارد.

CCRA حاصل می شود: ASIC هزینه و از لحاظ Area efficiency خرابی دارد. هزینه در هر چون
آمد تا بلیت برنامه ریزی شده باشد. یک سیکل با ASIC قابل استفاده می شود.

این اسلاید فقط chip را به عنوان قابل بازبینی نمی بینیم هزینه داره و این هزینه ها یکبار در تولیدهای

قابل برنامه ریزی و اوان کدهای که در خود configuration نام خودش نه دارد هست. هر چه طول configuration کمتر باشد فضای مورد نیاز برای بیت های configuration کمتر است و این قیمت های کم برای configuration استفاده می شود برای logic که نمی شه استفاده کرد و تعداد زیادی را داریم بعد می کشیم بدون این که می کشیم از کارایی logic که امکان استفاده نمی

تراشه های منطقی برنامه پذیر logic blocks

اسلاید ۸

تراشه های که در دنیا وجود دارند به دو دسته تقسیم می شوند
۱) IC های که مورد تمام سازش های طراحی می شوند که microprocessors ها و معمولی ها... این چون به تعداد خیلی زیادی قرار می گیرند و تولید می شوند Full custom طراحی می کنند خیلی efficient باشه و خیلی طراحی درستی که انجام میدن که efficient باشه. طراحی خیلی داره و ساخت از نظر RE خیلی بالاست

۲) نیمه برنامه ریزی می شوند ASIC ها هستند برای کار در مشخصی طراحی می کنند و اینها به صورت سلول است و با یک سری ای آی تی ها با ASIC ساخت یافته هستند
سلول استاندارد یعنی همان آن الکتریکال و ریلک می بینن و تعداد طراحی یک library از سلول ها داره که این library از سلول ها که هم تراشه و یک chip طراحی می کنن. در واقع که یوا را با استفاده از اینها طراحی می کنن و در یک دستگاه می توانیم برنامه ریزی کنیم
single programmable logic device

۳) برنامه پذیر: آن هایی که در این IC ها رایج است در SPLD (این PAL و PLA) و که ما همون آشنایتم که تعداد کمی می کشیم که باطون ساده سازی کرد و ساختار ساده ای دارن، دسته CPLD که device های قابل برنامه ریزی می کشیم که باطون ساختار پیچیده تر است
PAL است. ساختار الکتریکال های ترانزستور می کشیم باطون (D) های ساده سازی کرد. دسته آخر Field programmable gate array
FPGA ها هستند (gate array) قابل برنامه ریزی که برنامه ریزی آن داخل یک دستگاه انجام می شود مثل آی تی ها که آی تی از بازار می خریم خودمون در میان می کشیم و تو هم برنامه ریزی می کشیم

اسلاید ۹

۱) EPLD

CACTUS

device های برنامه پذیر باطون Filed programmable logic device های که در دستگاه های مختلف می کشیم
CPLD و FPGA که اینها FPGAs داخلی هستند و در می کشن

دقیقه می‌خواهم یک الیغش را سخت از این بزرگش تواند کشیم و در محصول به کار ببریم پس در حالت نیمه مناسبتی و برنامه‌نویس باید انتخاب کنیم یعنی بین FPLD ها و ASIC ها

Year: Month: Day:

Subject: استاد ۴

مقایسه بین FPGAs و ASIC ها

ASIC ها

• چندین ماه طول می‌کشد تا طرح مناسبی به دست بیاید این در صورتی که می‌خواهم زود به محصول برسم و به بازار عرضه کنم ASIC ها اصلاً مناسب نیستند، اصلاً وقتی ساخت و تست کردیم یک وسیله به دست می‌آوریم دوباره چند ماه صبر کنیم تا آن وسیله را برطرف کنیم

• در ASIC ها هزینه خیلی بالاست به خاطر اینکه چند هزار دلار یا چندین میلیون دلار طول می‌کشد تا اولین نمونه‌ای که سفارش می‌دهیم را تحویل بگیریم. در مورد محصولاتی که مشتریان زیاد هستند و خیلی‌ها می‌خواهند هزینه‌های زیادی را صرف کنند و هزینه‌های دیگر

FPGA ها

• در مدت زمان کوتاهی می‌توانیم برنامه‌ریزی کنیم و در کسری از ثانیه می‌توانیم به محصول برسم یعنی طایفه FPGA را عملاً از بازار عرضه می‌کنیم و در کسری از ثانیه می‌توانیم به chip برنامه‌ریزی شده برسم

• هزینه‌های برای خرید IC (هزینه‌های خرید) در هر چند دلار (chip هایی که خیلی امکانات داخلشون نیست) اما در هر چند هزار دلار (مثلاً chip جدید versal ACAP حدود ۱۲۰۰۰ دلار قیمت یک chip است) اگر تعداد chip هایی که می‌خواهم عرضه کنم زیاد باشد باید انتخاب مناسبی انجام بدهیم یعنی FPGA ها مناسب انتخاب کنیم که هزینه خیلی بالانویس (در ASIC هزینه‌های گزافی می‌آید و غیره گزافی زیاد و کسی در اینجا برعکس است) از لحاظ Area هزینه‌های حدود ۲۰ برابر تا ۳۵ برابر از ASIC بدتر است

• از لحاظ سرعت FPGA ها از ASIC ها کندتر هستند به این معنی حدود ۳ تا ۴ برابر است

• از لحاظ توان مصرفی یا پاشیدگی FPGA ها حدوداً ۵ برابر بیشتر از ASIC ها هستند (توان مصرفی است) بالایی دارند چون برانرژی‌های زیادی هستند که از شول استفاده نمی‌کنند (یعنی همان‌طور که توان مصرفی FPGA ها از ASIC ها خیلی بدتر است)

استاد ۵
CAETUS

chip هایی که نیاز دارند تست کنند باید این chip ها را عرضه می‌کنند

Altera و Xilinx حدود ۷۵ تا ۸۵ درصد بازار را در اختیار دارند

Year: _____ Month: _____ Day: _____ Subject: اسلاید ۸۷
 محاسبه قیمت سخت افزار intel و xilinx به کمک سیستم مدار

میزان رشد intel و xilinx در سال قبل از intel، رشد منفی داشته اما به مرور در رشدش زیاد شده و میزان رشدش امروزه از xilinx بیشتر است اما در مجموع به کمک xilinx بیشتر است

«FPGA Market»

بازار FPGA به عنوان در سال های اخیر خیلی رشد کرده و دو سه دلیل دارد که :

۱) AI (هوش مصنوعی) ۲) صنعت اینترنت اشیا ۳) Data Center

رشد صنعت FPGA در سال ۲۰۲۰ حدود ۲ میلیارد دلار بوده در سال ۲۰۲۵ حدود ۴۶ میلیارد دلار است و این رشد ۷۱٪ در ۵ ساله است

«FPLDs»

FPLD ها از بین های مختلف با هم متفاوت هستند و بهای دلخواهی ندارند

۱) از لحاظ معماری program کردن با هم متفاوت هستند

۲) از لحاظ معایر logic block ها با هم متفاوت هستند

۳) از لحاظ block های I/O ها متفاوت هستند

۴) از لحاظ معایر اتصال با هم متفاوت هستند

۵) از لحاظ core های خاص متفاوت هستند و به خاطر این تفاوت هستند

CPLD

CACTUS

اولین تراشه هایی که طرفدار بالایی داشتند و سیستم های بزرگ می توانستند آنها را به کار ببرند CPLD ها و

FPGA ها هستند

در مدارهای دیجیتال به عنوان پورت خروجی یک سری سلول های بزرگ دارند و این سلول ها از طریق connection wire به هم متصل می شوند و از یک سری از این سلول ها استفاده می کنند که این موضوع قابل برنامه ریزی می باشد و همچنین می توان که بیان کرد هر یکی از این سلول ها در مدار با یکدیگر اتصال می یابند و به هم متصل می شوند.

اسلاید ۱۵
 این شکل داخل سلول های cpld نشان می دهد (سلول های بزرگ به بالا تقسیم). داخل اینها یک سری ساختار PAL وجود دارد که یک سری and قابل برنامه ریزی و or غیر قابل برنامه ریزی دارند و این تابع منطقی است و صورت SOP باشد می تواند در داخل این سلول ها پیاده شود و قرار گیرد.
 در این ساختار PAL یک xor وجود دارد که یکی از درگاه های آن قابل برنامه ریزی است (بگذار xor یک xor وجود دارد) که در تابع xor همانطور که یک not قابل برنامه ریزی را همراه با این not قابل برنامه ریزی می تواند این که به جای پیاده سازی F و F' را می تواند پیاده کنیم و خوب است این که به جای پیاده سازی F' راحت تر پیاده تر از پیاده سازی F باشد.

این ساختار هم در cpld ها و هم FPGA ها استفاده می شود.
 در cpld های امروزی به جای این که ساختار PAL باشد ساختار PLA است یعنی فقط xor هم قابل برنامه ریزی است.

اسلاید ۱۳
 ساختار FPGA ها
 ساختار سلول های این نوع در پایه ارتباطات وجود دارد (بگذار زیادی از رسم ها و سلول ها) و این هر یک از این logic block ها هستند که می توان پیاده سازی می کنند.

اسلاید ۱۴
 (شکل های منطقی)
 این شکل های منطقی logic block (ها) اسم های مختلفی دارند: LE, CLB, LC, LUT
 logic element logic cell
 adaptable logic module
 ALM, LUT
 برای طراحی یک مدار منطقی در آسان و در نظر می گیرند.
 این امکانات نیز می تواند باشد به یاد یک طرف رسمی از یک طرف می تواند پیاده سازی می شود اما این اعتبار پذیری را max کنیم که از پیاده سازی
 ۲۸
 ۲۹

یکی از دستورات FPGA های امروزی این است که امکان پیکربندی را در زمان کار و اجرای برنامه می دهد
 که امروزه بعضی از دستورات پیکربندی را می توان در زمان کار با مدار و اجرای برنامه پیکربندی کرد

سه مورد بلاک بندی در Device های reconfigurable وجود دارد

- ۱) قطعه SOP (که در اینجا به کار می رود)
- ۲) قطعه LUT (یعنی FPGA های این مورد هستند)
- ۳) قطعه Block RAM

اسلاید ۱۴

در بلوک های اختصاصی SOP

- FPGA های که هم در CPLD ها می توان یافت و نام پیکربندی آنها مشخص نمی شود
- T-FF یا D-FF که هم در D-FF هستند
- یک ورودی که می تواند پیکربندی را از طریق reset کردن یا تغییر کردن
- یک راه دیگر این است که با فعال شدن نام پیکربندی در زمان کار و اجرای برنامه

اسلاید ۱۷ و ۱۸

در بلوک های اختصاصی LUT

- LUT یک حافظه متغییر از تعدادی ورودی و خروجی است
- حافظه ۲^K حافظه تک سیتی دارد
- ایمان میانه سازی خروجی را می تواند داشته باشد
- ایمان میانه سازی shift register را دارد
- ایمان میانه سازی حافظه RAM را دارد و خروجی توزیع شده دارد

اسلاید ۱۹

- یک حافظه ۸ ورودی RAM است چون LUT ها می توانند به عنوان حافظه پیکربندی برای برنامه های دیگر استفاده کنند
- توان هر یکی از حافظه SRAM را می توان تغییر داد
- در LUT یک حافظه است که حرف آخر این LUT ها این است که می تواند به عنوان حافظه پیکربندی برای برنامه های دیگر استفاده کند
- و تابع های boolean را می توان به آن میانه سازی نمود

SYNOPSIS

Year: 2000 Month: 1

خاص

Subject

۲×۴ ترانزیستور از این مدار اما این ترانزیستورهای NAND با ورودی را (SCG) static CMOS gate

از لحاظ Delay : SCG : اخیر می باشد و Lnt : اخیر است

- از لحاظ توان مصرفی: SRAM خوان توان مصرفی بالایی دارد پس گنجا خوان مصرفی بالایی دارد

8 21 22 23

س E حساب دارم در آن بر وجهه از است خای CMOS یعنی هر یک Area واحد و توان مصرفی بالایی

دارند چون خطر در طراحی 5 است یعنی بر 1 است یعنی خطر طراحی بر 1 است یعنی خطر طراحی بر 1 است

322

المادة السادسة

حرف تابع مشخصه را با `max` بیان می‌کنیم و برای هر حرفی که در جدول داده‌ها وجود دارد، یک `select` می‌نویسیم.

اسماء ۲۳

باسمہ تعالیٰ الرحمن الرحیم

* چیزی که بیشتر در FPGAs کاربرد دارد بلوک های مشخصه 100k است

۱۰- تراش های مقطعی ریاضه بندبر ۱۱

۱. اسلوب F_{12} بدون اختیار $FPCA, CPLD$ ، اختیار x

82 NLL 1

اثر اعم تراست دعای بر نامه نذر

از خنثی های خالص: صمغ عربی، گلاب

۲۰

Hard core (D)

اسلامیاد ۳۶

1- منور سوزی صغای اصلی 8

EpRom (Read Only Memory) یا (کرام) ← در این نوع سوئیچ های ثابت
برنامه ریزی با ترانزیستورهای Floating gate می کنند که ترانزیستورهای
خاصی دارد و این باشد

- EEpRom و Flash ← EPrOm حافظه دسر وجود دارد و دستی بر EEpRom و Flash هست چون EPrOm قابل پاک کردن است و در صورت اشتباهی این امکان وجود داشته و یک برنامه حافظه EEpRom و Flash، صورت اشتباهی قابل پاک شدن هست.

- Ant Fuse → وقتی برنامه ریزی می کنیم یک صورت دائمی وجود ندارد و قابل برنامه ریزی نیست

3V 4441

8 SRAM, 1MB

هدار SRAM ؟

از دو تا ۱۱ که صورت نیست به شیب به هم وصل شده تا سلسله شده است که به صورت مثل یک است این به صورت

این ترانزیستور آن را باید سازی کرده که هر دو inverter به ورودی دیگری وصل است

اسلام 8

استفاده از SRM در مدیریت منابع

[illegible]

SRPM احیاء دارند که کارشون میره بای سنانل حاصرت
سکام حکمت حیدر مدد انرا از شورای ممانع قرار نسیان سرحد

سر 810
یک مشتری دهنه SRAM ها LUT خاصه
یک LUT 4 ورودی از 14 تا سلول SRAM تبدیل شده است
Subject:

Year: Month: Day:

اسلاید ۱۱

علاوه بر سلول SRAM یکسری تراشه‌ها هم می‌تواند که آن سلول خاص را در خود درج می‌کند و به مثابه یک LUT 4 ورودی 4 تا سلول SRAM می‌تواند و یک چنین ساختاری که در درجی آن هم A و B است (چون در آن ورودی LUT)

اسلاید ۱۲

FPGA های که با SRAM برنامه‌ریزی می‌شوند مزایای دارند که باعث می‌شود مشتری زیادی را به دست آورده و در صنعت خیلی پرکار بوده است
مزایا:

۱) برنامه‌ریزی مجدد سریع انجام می‌شود یعنی بازنگری آن سریع در خود لکسری و باقی می‌ماند
برنامه‌ریزی کرد البته کاملاً بسته به نوع Recentization یعنی اگر چنانچه زیاد باشد زمان برنامه‌ریزی هم زیاد می‌شود اما در مقابل با SRAM base FPGA های دهنه SRAM base خیلی بهتر و سریع‌تر برنامه‌ریزی می‌شوند

۲) می‌توانست در تعداد ناچند در chip برنامه‌ریزی کنیم اما تکنولوژی ساخت دهنه SRAM base با تعداد در تعداد برنامه‌ریزی آنجا کم می‌شود یعنی ده‌ها هزار می‌شود مقدار برنامه‌ریزی کرد برای کاربرد های که داریم می‌خواهیم برنامه‌ریزی کنیم مناسب نیستند

اسلاید ۱۳

مقالب ۳

۱) صحبت زیادی را سلول های SRAM اشغال می‌کنند و برای پیاده‌سازی logic را کم می‌کنند
چون هر سلول SRAM ۵ تا ۶ تراشه‌ها را اشغال می‌کند

۲) چون SRAM است وقتی برق می‌خوریم باید قطع قطع می‌شود آن برنامه‌ریزی هم از بین می‌رود و دوباره باید اطلاعات را که داشتیم به یک حافظه می‌زنیم و اطلاعات برنامه‌ریزی را در آن تراشه‌ها هم می‌زنیم
ما هر بار که مدار روشن می‌شود اولی کار می‌کند که ما می‌خواهیم یک مدار جیسور روشن شود و این را می‌تواند به ما بگوید که ما می‌خواهیم در SRAM و FPGA تراشه‌ها ما می‌توانیم

CXCTUS

برای FPCA های قبلی بر SRAM به یک حافظه خطی خاص احتیاج داریم که این باعث می‌شود حافظه
حدار را زیاده‌تر

۳) امنیت در برابر سرقت طرح یا این است برای مالدون امنیت برای زیر کار خود اطلاعات رمز شده
Year: Month: Day:
اطلاعات را می‌فرستند و دنیا های رمز شده را می‌فرستند و اطلاعات رمز شده را می‌فرستند و
حدار decade شده و خود دارد که رمز نشانی می‌کند و اطلاعات را در SRAM را می‌دهد

۴) توان رمز نهایی سلول های SRAM