



استاد کیم غلامی در محوطه جای ۵ = محوطه ۱ بلوس محمد درم آورده و B کا احاطه کرده و B کا احاطه کرده و B کا احاطه کرده  
به عنوان محوطه استفاده کیم. آن B کا احاطه ۵ = محوطه ۱ استاد کیم غلامی درم آورده و B کا احاطه کرده و B کا احاطه کرده

ing distributed  $\frac{1}{n}$

823 2011

هر LAB از مالت ALM تشکیل شده که در هر ALM یک Cat و ورودی وجود دارد، بنابراین هر LAB

4F بیت SRAM، 40 بیت ALU، 40 بیت حافظه

از LAB به عنوان Shift Register لحاظ می‌کنیم FIFO و یک استادم

\* تارت LAB  $\leftarrow$  LAB  $\leftarrow$  استادان استادمه عنوان حافظه وجود ندارد LAB

این امکان وجود دارد چون علاوه بر Lut بودن مایه امکان نوشتن در حافظه shift Register را داشته باشند تا بتوان به عنوان حافظه از آنها استفاده کرد

\* انتخاب های رایج برای  $C_1$  و  $C_2$  فرم های  $\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$  و  $\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}$  به راحتی زیادای هم دارند

اسماء

در ترانس های شریک ultra scale سری بلای های درجه یک ام ultra RAM ایجاد کردند.

۲۸۸ بیت شعر هستند، اما تعداد آنها ثابت است و ۴۵۹۶ آلفا word ۷۲ بیت هستند.

توجه داشته باشید که این امکان را دارند ۸

ultra RAM type bit with 22, 25 & kintex ultra scale.

" " " " " zing wtra scale

نقطه کار می در حلی از اینک صاحب نیازند



تقسیم special persons block برای کاربر در خصوص استفاده شده مدارهای آن می شود. مثال استفاده از آن

Year: / Month: / Day:

حافظه زیادی را در مدارهای آن می حافظه

استفاده شده می توان برای منظور دیگری به کار برد. مثلا اگر خواستیم یک مدار دیگری با ترسیم پیاده سازی کنیم آن را  
همانند می توان از حافظه ها برای پیاده سازی توابع خاص یا استفاده کرد یا نه؟  
بله از حافظه ها می توان برای پیاده سازی توابع خاص یا استفاده کرد. مثلا برای تابع  $f(x) = x^2$  می توانیم  
مثلا در مدار ترسیم خواستیم پیاده کنیم و هم می توانیم با سه حجم کردن آن را با پیاده سازی کرد و هم از  
حافظه ها برای پیاده سازی آن استفاده کرد

حسابی پیاده سازی توابع خاص یا استفاده از حافظه :

- سرعت بالا در پیاده سازی با آن امکانی تواند باشد

- سرعت آن قابل پیش بینی است و به اندازه ی دسترس می حافظه است

برای خبر ترسیم  $f(x) = x^2$  می توانیم پیاده سازی کنیم که به عنوان  $x$  در مدار ترسیم و خبر را بخارا

در حافظه قرار دهیم

برای پیاده سازی تابع  $f(x) = x^2$  مثلا ضرب ۴ بیت در ۴ بیت که نتیجه ی ۸ بیت است حافظه ی ۸ بیتی می

۸ خط آدرس داشته باشد که ۴ بیت را به خطوط ۴ تا ۷ بستم و ۴ بیت دیگری را به خطوط ۰ تا ۳ بستم و اینجوری

تابع دو ضربه را پیاده می کنیم

در این اسلاید بخشی از حافظه را نشان داده است که اعداد را به صورت ضرب در ۴ بیت

نشان داده است. اگر خواستیم اعداد ترسیم را در هم ضرب کنیم مثلا ۸ بیت در ۸ بیت یا ۳۲ بیت در ۳۲ بیت و ... دوباره داریم

۱) با هم حافظه های کوچک را به هم میزنیم تا حافظه ی بزرگ تشکیل بشود

۲) از ضرب شده با مقدار بیت کمتر استفاده می کنیم و سه حجم میزنیم مثلا ضرب شده  $4 \times 4$  بیت را اینطوری میزنیم

CACTUS

$8 \times 8$  بیت میزنیم تا استفاده از ضرب شده های ۴ بیتی که با هم میزنیم این ۵ بیتی را با هم

یا میزنیم از حافظه های بزرگ استفاده می کنیم و خطوط آدرس به اندازه ی کافی داشته باشد



دریادہ سازی توابع حسابی یا منقسمہ تفریق و جمع

۱) دقت قاسمہ : خطوط یا تاخیر در می توانند در هر دقت به جا آورده شوند

مثلاً خیز در هر دقت از اعداد صحیح را می توانند به جا آورده و خیز عددی مثل ۵۰ را می توانند بدست بیاورند که برای بدست آوردن خیز از خطوط اعداد می توانیم از روش کوتاه کردن استفاده کنیم یعنی بنیم ۵۰ را برابر ۱۱ است (۵۰ را ۱۰ را خیز کنیم) و بعد خیز یک را بدست آوریم  
 روش دوم اینست که از روش کوتاه کردن استفاده کنیم بنیم ۲ به ۱۵ و خیز ۲ را بدست آوریم  
 راه دوم اینست که خیز بنیم یعنی برای بدست آوردن خیز ۵۰ را بنیم در خیز ۱ و ۲ را داریم  
 پس خیز ۵۰ را می یابیم این کار است یعنی ۱۱ برابر ۱۱ است با استفاده از تقریب خطی خیز در دقت تری نسبت به روش کوتاه کردن و کوتاه کردن داریم اما تحت احتراز بسیاری صرف می شود  
 منقسمه تحت دقت در این اسلام منقسمه تقریب خطی است و منقسمه تحت دقت برای روش کوتاه کردن و کوتاه کردن است

۲) دقت قاسم در خیز از اعداد صحیح و تفریق خیز از اعداد صحیح یا جمع و تفریق اعداد صحیح بدست آوریم که در این صورت دقت بالا می رود و روشی خطی و تقریبی برای بدست آوردن

یک نمونه از توابع حسابی پیچیده تابع سلیم و در صحت

از خواص اعداد این تابع را می توانیم تحت این جدول یک به یک نشان داده و احتیاج داریم یک جمع شده یک تقسیم شده و ...

اگرچه توابع با استفاده از خطوط آن را می یابیم و می توانیم تعدادی آن را در خطوط ذخیره کنیم یعنی تعدادی را از خط قاسم که از خطوط ذخیره کنیم و خطاهای خواص (اره) ۵ را حساب کنیم آن را در خطوط که پس دهی می کنیم و خطی را در آن را در خطوط بدست می آوریم



برای ضرب اعداد ثابت در مقیاس مثل  $3 \times 8$  در DSP ها و سیستم های ترکیبی استفاده می شود و می توان حاصل ضرب عدد ثابت را در حافظه را در حافظه را برای دسترسی به آن در حافظه می توانیم

Year: \_\_\_\_\_ Month: \_\_\_\_\_ Day: \_\_\_\_\_ Subject: \_\_\_\_\_

هر عددی در هر عددی را پیاده سازی کنیم

حالا اگر عدد ثابت بزرگتر از ۸ بیت باشد استفاده می کنیم منطق آن را در ۸ بیت می کنیم مثلا اگر بیتی  $3 \times 8$  می توانیم  $7 \times 8$  را انجام بدهیم از جدول حافظه ای که برای  $3 \times 8$  استفاده کردیم برای  $7 \times 8$  هم استفاده می کنیم منطق آن را در ۸ بیت می کنیم و بعد در جدول جدید را برای دسترسی داریم در اینجا اگر وقت بیشتر بخواهیم به حافظه دسترسی احتیاج داریم

فهرست در عدد ثابت در هر از می می بینیم استفاده می شود که در این اسلاید فرمول آن آمده است و این ها عدد های ثابت هستند که این یکی از کارهای ضرب در عدد ثابت است

توانیم در هر ای که کاربرد دارد می بینیم استفاده کرد تابع  $\cos$  و  $\sin$  و  $\logarithm$  ... است. هر طریقی خواستیم یک تابع خاص برای پیاده سازی کنیم آن در بعضی سیستم های دیجیتال ما می توانیم در آن استفاده کنیم که آن را پیاده سازی کنیم و با هر عددی که در آن استفاده می کنیم که آن را پیاده سازی کنیم یا با حافظه ای که کار را انجام می دهیم، که توانیم از حافظه پیاده سازی کنیم یا با حافظه پیاده سازی شود.

اگر خواستیم با هر مثل  $\sin(x)$  پیاده سازی کنیم  $x$  را به عنوان حافظه می باید بدهیم و  $\sin(x)$  در  $data$  پیاده می شود.

پیاده سازی state machine

اگر بخواهیم یک مدار دیجیتال پیاده سازی کنیم باید FSM پیاده سازی می شود و FSM state diagram

پیاده سازی می شود در این سیستم ها و مدارات FSM ها را به کار می آوریم و با هر عددی که در آن استفاده می کنیم که آن را پیاده سازی کنیم و با هر عددی که در آن استفاده می کنیم که آن را پیاده سازی کنیم



مقرون به هر سه مورد می توانیم با حفظ پیاده سازی کنیم

در این اسلاید یک مثال از state machine است و از روی این state diagram آن تبدیل می کنیم  
و حالت جاری و ورودی را به عنوان آدرس قرار می دهیم و  
Subject: Year: Month: Day:

حالت جاری و خروجی را به عنوان داده قرار می دهیم و در حافظه ذخیره می کنیم  
بنده ای که وجود دارد این است که این pin های خطه را باید به اجزا و به جی وصل کنیم

اسلاید ۳۹

برای پیاده سازی state machine با حفظ به این صورت است که ورودی ها و state جاری  
خطا در مسافت و خطا در next state و خروجی را می دهد (Door خروجی)  
است و در خطا در next state هست  
و حالت «وارده»

### computation-oriented tiles

اسلاید ۴۰

چون ضرب در ماتریس های مختلف خیلی مهم و رایج است داخل FPGA حاضر شده اند  
و تعداد زیادی ضرب شده اند که فراموش نمی شود که ضرب های موازی را انجام بدهند  
ضرب ماتریس می تواند به این صورت باشد که ضرب های مختلفی را می توان انجام دهد که می توان موازی انجام بدهیم به سببی  
که از ضرب شده های زیادی استفاده کنیم

اسلاید ۴۱

Layout cyclone ۲: قیمت های بسیار پایین و توان های ۱۰۰k هستند و توان های ۱۰۰k block DSP  
هستند و عملکردی که در DSP ها انجام می شود ضرب جمع و هستند (DSP بلوک ها را برای عملیات  
حسابی بسیار کارآمد هستند و قیمت های بسیار پایین logic fabric که چون بلوک های خاص هستند  
هستند و داخلون FF و LUT و MUX

اسلاید ۴۲

تعداد DSP بلوک ها در cyclone ۲: قیمت های مختلف را نشان می دهد و هر DSP بلوک می تواند  
یک ضرب ۱۸ بیت در ۱۸ بیت را نشان دهد که تا چهار دهه ۴۰۰ DSP بلوک وجود دارد  
میان cyclone ۱ و cyclone ۲: او هم که آن حدود ۲۰۰ بلوک DSP است اما امکان ضرب  
floating point ۴۰ بیت در ۴۰ بیت وجود دارد





Scanned by CamScanner



در جدول زیر اسلاید، در سطر اول نسبت  $\text{Dsp}$  به  $\text{logic cell}$  ها یا نسبت  $\text{soft/hard ratio}$  برای ۴۴۲ است (۹۸۰/۱۲۴۰) که برای اعلامی که بیشتر سیستمی معتد آرا می باشد

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

Subject: ..... ۱۰۹۰

## Embedded processors

در داخل  $\text{FPGA}$  های امروزی  $\text{processor}$  هم به صورت  $\text{hard block}$  قرار داده اند

اسلاید ۲۹  
در گذشته وقتی یک سیستم می خواستند بسازند آنرا یک  $\text{Dsp}$  می نیاز داشتند و بخش های سخت آنرا می ساختند. کارهای سخت آنرا می که می خواستند خیلی سریع انجام می دادند  $\text{FPGA}$  و این کارهای که خیلی از کارها سرعت هم نسبت به نرم افزار تراشه ها

در  $\text{FPGA}$  می بینیم که از سخت افزار نرم افزار استیل می شود و چون یک  $\text{Dsp}$  یک  $\text{FPGA}$  می دانسته اند کارهای سخت آنرا می و انجام دهد و یک  $\text{memory}$  خارجی و راسته های دیت و یک  $\text{bus}$  که آنها را هم ارتباط داده باشند

چون در خیلی از کارها نیاز به همین سختی وجود داشته و برای یک  $\text{board}$  هم فضای زیادی را می سر و در ارتباط با این سیستم می بینیم و از طرفی تکنولوژی ساخت  $\text{IC}$  بیشتر می شود و چون برای این کارها در تراشه ها در دسترس است و خیلی سریع خارج از تراشه  $\text{FPGA}$  بود و داخل  $\text{FPGA}$  آورده و حالا  $\text{FPGA}$   $\text{memory block}$  هست و برای این عمل است به طرف خارج می می باشد، تراشه های امروزی که برای دستگاه های خاصی بودند را

می توان داخل  $\text{FPGA}$  ساخت و خود  $\text{CPU}$  را داخل  $\text{FPGA}$  قرار دادند و امروزه در بعضی از  $\text{FPGA}$  ها هر چیزی که یک سیستم کامل احتیاج دارد، داخل آن هست و چون این داخل تراشه را می بینیم ارتباط خارج از تراشه کمتر است و برای این سرعت افزایش پیدا کرده است،  $\text{Area}$  روی یک برد کمتر کاهش پیدا کرده و توان مصرفی هم کمتر است. یک مزیت دیگر این است که باعث می شود  $\text{Reliability}$

$\text{reliability}$  بیشتر می شود زیرا ارتباط بودن بر روی یک تراشه در مقابل قرار می دهند با ارتباطات داخل تراشه در نتیجه سیستم بالاتر می رود

اسلاید ۳۰

CACTUS

SOPC

SOC: تراشه ای که  $\text{Functional element}$  های اصلی یک سیستم را در داخل یک تراشه قرار می دهند



تقسیم کردن تراشه‌های مختلف در داخل تراشه FPGA به CPU، memory و logic multiplier ...  
system on a programmable chip

Year: \_\_\_\_\_ Month: \_\_\_\_\_ Day: \_\_\_\_\_

Subject: اسلاید ۸۵

پایانه‌های CPU در یک سیستم یکپارچه (چه سیستم داخل تراشه چه بیرون):  
- باید به علایم انجام سیستم تقسیم بندی سیستم و دسترسی کارها را به بخش نرم افزار در یک سیستم دسترسی داریم  
- بخش hardware/software partitioning می‌تواند طراحی را به صورت دستی انجام می‌دهند  
و این‌ها برای وجود دارند در این سیستم یکپارچه می‌تواند

- برای این کار verification & simulation این سیستم یکپارچه باید به سیستم دسترسی داشته باشد از یک طرف  
و نرم افزار استیل شده باید یک co-verification وجود داشته باشد و هم بتواند یک طرف را داشته باشد  
که در هم نرم افزار

اسلاید ۸۶

### Microprocessor Core

اگر سیستمی از بخش‌های نرم افزار و سخت افزار استیل شده و در حقیقت از پرازنده داخلی FPGA استفاده کردن در نوع است

### Hard Core

پرازنده‌ای که علاوه بر آن فنونی در داخل FPGA قرار دارند

مثلاً چون می‌تواند layout که در داخل پرازنده ARM هست عموماً در داخل تراشه FPGA قرار داده اند

- در xilinx این hard core با استفاده از آرم power pc بود و بعد از آن آرم

- شرکت intel از پرازنده ARM استفاده می‌کنند

- شرکت quick logic از mips استفاده می‌کنند

### Soft Core

منطوق بر اساسی است که از سیستم کردن logic block ساخته می‌شود

برای این لازم باشد طراحی خود را خود را در پرازنده این را انجام دهد خود شرکت intel و xilinx

Quick logic و micropro. و soft و این می‌تواند در اختیار کارکنان قرار می‌دهند مثلاً

میکروپراسسوری که شرکت xilinx ارائه کرده تا اسم microblaze است ...



soft processor core

Subject: Maths

۱۔ سنیہ امت

۔ توان مصروفی مالیاتی دارد

-حرفی

82F null

soft processor core کا حصہ اور اسے دو حصوں میں تقسیم کیا گیا ہے۔

soft core (1)      firm core (1)

soft: هر دو RTL است که می‌تواند شناسایی شود (علی‌گذاشته و درایو)

Prim: علاء و رشتن place and route هم شده باشه و مکان مناسب برای قرار گیری را مشخص می کنه

- firm soft type intensity of peripheral

فصل interrupt controller, memory controller, timer counter, peripheral (و غیره) است و عموماً در بارنده ها به آن احتیاج دارند و این peripheral ها در بارنده های سخت (و به صورت نرم) (firm, soft) در دسترس می باشند و می توانیم

منویس: FPGC library (احتمالاً نام کتابخانه)

Logic cell 1000 و در هر ۳۲ بیت است

علاوه بر این یک soft core در این راه به نام Pico Blaze نیز دارد و این سورس ۸ بیت است و

CACTUS.

۱- برای ۱۰۰۰ cell و ۵۰ cell از استارک استفاده می شود -  
- تست الکترونیک بر رویه nips II به بر رویه ۳۲ می باشد

کامیابی و تنوعی در بازنده‌های soft core انجام دادند که instruction set را می‌توانند extend کنند یعنی طراح می‌تواند خودش یک دستورالعمل جدید را در آن اضافه کند و آن را در FPGA قرار می‌دهد

اسلاید ۵۶: Year: \_\_\_\_\_ Month: \_\_\_\_\_ Day: \_\_\_\_\_ Subject: \_\_\_\_\_

تاریخچه ای است که در دهه ۹۰ میلادی شروع شد. در دهه ۸۰ میلادی در حال آشنایی با لینوکس و در دهه ۷۰ میلادی در حال آشنایی با سیستم‌های امروزی قرار داشت. در دهه ۶۰ میلادی در حال آشنایی با سیستم‌های خاص و در دهه ۵۰ میلادی در حال آشنایی با سیستم‌های امروزی قرار داشت.

اسلاید ۵۸:

Recent Dedicated block  
IC های رایج در لینوکس و در دهه ۹۰ میلادی با اسم Versal ACAP که قدرت اصلی آن این بوده است. این سیستم‌های هوش مصنوعی بودند و صاحب ابزارهای هوش مصنوعی این device ها را می‌توانستند داشته باشند. در این راستا کارهای مهمی در آن زمان انجام می‌دادند و در دهه ۸۰ میلادی در حال آشنایی با لینوکس و در دهه ۷۰ میلادی در حال آشنایی با سیستم‌های امروزی قرار داشت. در دهه ۶۰ میلادی در حال آشنایی با سیستم‌های خاص و در دهه ۵۰ میلادی در حال آشنایی با سیستم‌های امروزی قرار داشت.

این کارها برای عملیات با کسب و کار زیاد، عمدتاً اینسین‌های AI و به سیستم ۵۰ می‌توانند کار کنند.

اسلاید ۵۹:

همکاری داخلی این کارها:

معمولاً در صورت (VERSAL ADAPTABLE) سیستم‌های عام است، داخلی آن

یک Dual core ARM و یک hard block

به کار می‌رود. بازنده‌های BCALAR Engines که برای سیستم‌های روز افزونی را انجام می‌دهد.

بخش intelligent Engines از بین Dsp Engine و AI Engine شامل شده است.

AI Engine کارها برای کارهای هوش مصنوعی انجام می‌دهد.

اسلاید ۶۰:

نمای کلی از ساختار داخلی

CACTUS

یک سیستم در یک بازه که در آن یک Engine Scalable را می‌تواند در یک FPGA قرار دهد.



Adaptable Engine یا LB دارد... شکل هندسی و یک جهت محاسبه vector processor ها هستند که intelligent Engine و دایکسون یک processing element است که با راس و می تواند مقایسه با پردازنده های دیگر انجام دهد

Year: \_\_\_\_\_ Month: \_\_\_\_\_ Day: \_\_\_\_\_

اسلاید ۴۱

یک تپایه از intelligent Engine است که Array از Engine AI ها را شامل می شود

اسلاید ۴۲

شکل داخل Engine AI را نشان می دهد که یک Array از Engine AI است که داخل هر ستون شکل پایی وجود دارد که عمده آن یک Fixed point SIMD است که قبل از یک پردازنده نوبت است و یک Floating point SIMD وجود دارد در کنار آن scalar ها و رجیسترهای لازم است برای عملیات که کارهای خاص ترنیک Deep learning نیاز دارند که در یک خطی ایجاد می شود و آن عملیات را به سرعت انجام دهد و یک رای ای از Engine AI ها را تولید می کند و سرعت را بالا می برد و این عملیات را به سرعت انجام دهد و سرعت را بالا می برد از آنجا استفاده می کنند

به صورت کلی این ACAP اینها CPU و FPGAs و vector processor در یک تراشه باشند و مشخص می شود که این MAP یک شبیه یک جهت هایی که برای CPU ها هستند روی CPU اینها یک جهت برای FPGAs و یک جهت برای vector processor اینها هستند