

الحلقة بين ١١ و ١٢، ١٣، ١٤

سنة ١٤١٥ هـ في شهر ربيع الأول ١٢٨٥ م
global nothing
وكانت في سنة ١٢٨٥ م في شهر ربيع الأول ١٢٨٥ م
في سنة ١٢٨٥ م في شهر ربيع الأول ١٢٨٥ م

CACTUS
(url negatiation base)
١٤١

مسئله ۱۳: negotiation base

منظور از negotiation base:

net با حجم فضا در بین خود و با حجم فضا در بین خود

Year: Month: Day:

مسئله ۱۴

path finding

این الگوریتم به این صورت عمل می کند که net ها را به گونه ای مرتب می کند که به هم وصل شوند و مسیر را پیدا کند و
الگوریتم هر net را مستقیماً از مسیر net ها می بیند و سعی می کند که به بهترین مسیر برسد و این کار را با
حفظ مسیر در این نوع فضا می کند و به کمک این فضا می تواند به بهترین مسیر برسد و این کار را با
net و از این فضا می تواند به بهترین مسیر برسد و این کار را با
حداکثر ۱۶-۳۵ است (۱۶ فرامیت فضا است) مانور می بیند و این ۱۶ مانور با حجم
حداکثر ۱۶-۳۵ است و این مانور ۱۶ در یک جهت است و این مانور ۱۶ در یک جهت است و این مانور ۱۶ در یک جهت است
به گونه ای که مسیر خود را انتخاب کرده و به بهترین مسیر می رسد و این کار را با
یک مسیر مشخص می کند (هر یک تعداد net ها که از فضا می گذرد و به بهترین مسیر می رسد)
۱۶-۳۵-۳۵ (۱۶-۳۵-۳۵) این مانور ۱۶ در یک جهت است و این مانور ۱۶ در یک جهت است
که به خوار و خاشاک را انجام دهد این مسیر را به بهترین مسیر می رسد و این کار را با
در یک جهت است و این مانور ۱۶ در یک جهت است و این مانور ۱۶ در یک جهت است
حفظ فضا را به بهترین مسیر می رسد و این کار را با
فضا را به بهترین مسیر می رسد و این کار را با
شورت پات (short path) استفاده می کند و این کار را با
نوع خاصی از فضا را به بهترین مسیر می رسد و این کار را با
این الگوریتم net ها را به بهترین مسیر می رسد و این کار را با
است به این دلیل که این الگوریتم به بهترین مسیر می رسد و این کار را با
است به این دلیل که این الگوریتم به بهترین مسیر می رسد و این کار را با

مسئله ۱۵

رایج ترین یک cost function (cost) (base cost) فضا را به بهترین مسیر می رسد و این کار را با

CACTUS

استاده از فضا را به بهترین مسیر می رسد و این کار را با wire seg و path

اسلام:

High level synthesis in FPGA = High level synthesis in FPGA
 این روش برای طراحی مدارهای دیجیتال در FPGA استفاده می‌شود.
 در سطح بالا ترسیم می‌شود و سپس به زبان HDL تبدیل می‌شود و در نهایت به زبان C++ و C تبدیل می‌شود.
 High level synthesis = High level synthesis

High level syntaxis وجود دارد و در بسیاری از زبان‌ها در آن ایجاد می‌کنند
مثلاً C, C++, ... وجود ندارد در HL است و در برخی از بسیاری از زبان‌ها در C و C++ صورت می‌گیرد
HL نیست و پس وجود ندارد سطح بالا آن را سیستم سیزدهم نیز نیست مثلاً فاکتوریل
sys call نیز نیست مثلاً printf و scanf و malloc و ...

int type ki variable

تقریباً ۱۸ سانت داشت. اِستیم باید با ۳۲ سانت تقریباً ۱۸ سانت باشد. و ۱۶ سانت با ۳۲ سانت در استیل استاندارد است. هم مقدار انحرافات را زیاد می‌کنیم و هم جیسپر ها را نزدیک می‌کنند و بعداً را با بلند استیل جیسپر انجام می‌دهد مثلاً برای خنک، منفرجه شده جیسپر نزدیک می‌شود.

درمان C+T و H₂S با این ازمین اظهار داشت که فرایند Lint و فرایند دیواره باکتری

در این تلاش می شود تا آن ارقامی که در این صفت در بالا نیست را هم اعلان پذیر شد چون جنبه اقتصادی دارد
و مقدار خیلی زیادی است و خود داده در ای است. توجه روی نامش هم جای گفت افزای احب است و باید تحقیق کرد در ماستون کش
و اکثر ترین تا اسبابهای ران را هم بشود پذیرد و آن در این صفتی که در این وجود دارد و قابل
نیاد سازی با 10^{-4} می شوند و این صفتی از این راه را هم در صفتی پذیرد و شیت ۱۴۵

اسم: ...
 موضوع: High level program
 Year: ... Month: ... Day: ...
 Subject: ...

حاضر DFg برای بیان طایفه از برنامه در اختیار داریم loop و branch و خود در این مباحث بود برای این
 با بررسی ساختار برنامه مدل رسم از FsmD sequence graphs و DFg استفاده کنیم

اسم: ...

FsmD

extend شده OFg و FsmD است. FsmD یک دنباله ای از state ها است
 می دهد و state ها را به هم وصل می کند. در یک مجموعه ورودی طایفه این مدل است. مجموعه ورودی طایفه
 مجموعه ورودی طایفه است که از آن استفاده می کنیم و در آن Function که نام F و H و action function
 transition function

transition of F: $s \times I \times v \rightarrow s$ با اتمام ورودی state و variable input
 state map

action H: $s \rightarrow o + v$ output state و variable output

FsmD نسبت به Fsm

عبارت که اتمام به خودی خود قطع نیست و به خودی خود قطع نیست... و این عبارت
 یک دنباله ای است و transition ورودی $s \times I \rightarrow (Fsm) s \times I$ یک مجموعه داده ای است
 FsmD

اسم: ...

چگونه این عبارت اتمام شود و در هر عبارت به مدل FsmD تبدیل می شود؟

رایه assignment یک state توسط یک state (مثلاً $a = b$) یک state توسط یک state و در این
 state خاصیت مهم این است که state c state دین edge ایجاد می کند.

برای loop یک body در هر طایفه و در این loop body next statement به هم وصل می شود و برای
 body loop یک state و این است که در هر طایفه state را به هم وصل می کند و در این state c

اسم: ...

CACTUS

condition به عبارتی یک join state ایجاد می کند و در این state c
 این دستور body loop می داریم در این state c به هم وصل می کنیم و در این state c دستور

next statement ایجاد کنیم و درش end ! را می‌گذاریم که اگر end برقرار شود سراغ این برود
 یک edge معمولا آخرین دستور body join = loop body یک edge هم از join به end ایجاد می‌کنیم
 Year: Month: Day: Subject: FSD (فصل دوم)

اسلاید ۱۳ : branch
 دستور if else هم داشتیم مثل این اسلاید انجام می‌دهیم. دستوری که در if است صورت
 state های بعدی نیست هر هم ایجاد می‌شوند و یک state conditions , joint هم شکل می‌دهیم
 اگر ما شرط بزنیم این دستور را می‌توانیم برای حالت else با شرط بزنیم که در ادامه !
 و این آخر و در آخر یک arc دارد statement به join وصل می‌کنیم و در آخر به next statement
 وصل می‌کنیم

و حالت است و قسم ۱۱
 اسلاید ۱۴

فصل FSD :
 gcd گرفتن مستقیم یک مشترک و چون واحد دهد
 state داریم به assign کردن به label : fin . دوم دستور while
 دوم دستور return x ← gcd . دستور while یک دستور تکرار است
 است و از حلقه دستور که شکل شده و داخلش یک حلقه دارد و اگر است
 این انوار HLS می‌خواهد این که در خروجی یک مدل تولید کند و ربطش به سوالاتی
 مشخص این کتاب FSD را می‌تواند بیاندازد
 تفاوت FSD و FSD L : در FSD state فعلی اسم می‌دهیم در حالی که
 اینطوری عملیات می‌تواند انجام می‌شود

اسلاید ۱۵ :

FSD : Datapath synthesis

انوار استر از این کتاب استفاده می‌کند و data path و control path را تولید می‌کند
 CACTUS
 این کتاب از روی کتاب data path آشنایی هم باید می‌توانیم عملیات لازم است و چه

۱۱ Allocation → مشخص کردن منابعی که برای اجرای هر کار در واقع نوع منابع را مشخص می کند (مثلاً فرکانس یا ...). این منابع را می توان به صورت تعداد کارها را مشخص می کند

Year: _____ Month: _____ Days: _____ Subject: Binding

۱۲ Binding → عملیات تعیین کردن منابع برای هر instance است. در واقع نوع منابع را مشخص می کند. مثلاً ما فرکانس را مشخص می کنیم و operation به هر instance از فرکانس ها و باید ما را مشخص می کند

۱۳ scheduling → زمان مشخص می کند که هر کدام از این عملیات چه زمانی باید شروع شود. کار کردن در مقابل وقت (مشخص کردن منابع در هر زمانی در اختیار عملیات قرار گیرد)

اسلاید ۱۸:

allocation:

یک توصیف داریم که مجموعه ای از operator ها با task های مشخصه که این مجموعه را با T نشان می دهیم که منظور از این تقسیم و جمع و تفریق است.

هر کدام از این عملیات به عنوان یک منبع خاصیت باید این مجموعه ای از resource type ها اجرا شود که با R نامیده می شود. مشخص می کنیم (منابعی که برای اجرای هر یک از این کار را انجام دهند) تعداد operator ها با تعداد resource type ها عموماً یکسان باشد.

allocation یک عملیات است که مجموعه ای از resource type ها با یک مجموعه ای از اعداد جمع تبدیل می کند که هر کدام از این اعداد لازم از instance های resource type r_i است.

مثلاً مشخص می کند که از چه منابعی و از چه منابعی داریم و ... و این صورتی $\alpha(r_i) = z_i$ (تعداد z_i تعداد لازم از r_i است)

اسلاید ۱۹:

مثال:

یک عملیات داریم Dfg که در یک است مشخص شده

مجموعه task ها از Dfg مشخص می شود

R ما دو نوع است یک ضد است دیگر جمع شده (این Dfg) به جمع شده حجم به نوع دفع ای نامیده

حجم آخرین و حجم مناسب

Year: _____ Month: _____ Day: _____ Subject: Math

$Z^+ \rightarrow T$: α که به T و T را به Z می‌رساند و برآورد Task از دست می‌آید نوع آن α resource type نیاز داریم

* ممکنه ما برای یک مقدار بین $resource$ Type داشته باشیم مثلاً ما اینجا تقسیم صف $adder$ داریم
 ممکنه برای $adder$ کردن CLA و $Rip add$ داشته باشیم که اگر اینطور باشه این سیستم ما
 باید مناسب ترین نوع را انتخاب کنه

o binding

resource type → Task
 Rx 2⁺ core task also Binding
 select Cpu instance plus
 gpr nu

مَدَد

~~publ~~, 461 vs instance - inst resource type, instance 1 ~ 0
publ, 61 vs instance

Contract is instance of general principle & binding other

scheduling

Year: _____ Month: _____ Day: _____
Task scheduling, Resource allocation, در اختیار داشتن Resource

در این زمان شروع این task می باشد که FSD (First Start Date) در جدولی که به شما داده شده است

High level synthesis

در سیستم های reconfigurable و Device های FPGA یک نسل اساسی وجود دارد که ASIC و این است که در ASIC Resource را می توان به سیستم و Allocate می کنیم و به عنوان یک قطعه ای که می تواند انجام دهد کار دیگری می تواند باشد اما در reconfigurable Device است Resource های مختلف از این جهت می تواند باشد که می تواند انجام دهد کار دیگری

General vs. Rcs High level synthesis

مثال: اول فرض کنیم داریم ASIC انجام می دهیم
فرض کنیم در یک صورت در یک حالت داریم که می باید به راجع به آن در نظر بگیریم
از آن یک FPGA تبدیل می کنیم به یک حالت دیگر که می باید به راجع به آن در نظر بگیریم
مرحله ۸ و ۹ را می توانیم به یک مرحله دیگر تبدیل کنیم
اگر ما یک ASIC را داشته باشیم می توانیم یک جمع کنیم allocate کنیم به یک مرحله دیگر
فرض کنیم (مثلاً) از هر یک از اینها می توانیم allocate کنیم به یک مرحله دیگر در نظر گرفته می شود
فرض کنیم که می کنیم
Area در هر مرحله

Adder Area و ده و ده
فرض کنیم که ده و ده Area

CACTUS
حالا می بینیم عملیات را می توانیم انجام دهیم و به یک مرحله دیگر تبدیل می دهیم

Year: Month: Day: Subject: مسند تراجم و زاد و نحل مولانا محمد حسین

الانجام داد

کہ ایمان جسم دارند

2. Рыба

CACTUS

Year: _____ Month: _____ Day: _____

Dist

8. 10. 11

3. Schedule

الخامس .

$$\therefore \mu_{\text{P}_2\text{O}_5} = 1$$

قسم

2 partition

Year: 1397 Month: 10 Day: 15

$$(U_{p_1} \dots p_k = n)$$

Subject: Mathematics

کتابخانه P_k کا مسدود ہونا باعثِ ناجور مجموعہ سے ہوا

feasible partition (یا پارتیشن ممکنه، مجاز) : پارتیشنی است که device reconfigurable

H که area آن را $a(H)$ نشان می‌دهیم و مقدار p چنان $p(H)$ می‌نامیم. \bullet یارِ شین ما
 feasible است شرط آنکه برای هر p_k (برای هر یارِ شین) a و area آن p_k برحسب a
 $a(H)$ باشد. $a(p_k)$ هست مجموعه area یک یک نودها که در آن یارِ شین است و شرط
 دهنده این مجموعه دهنه ارتباطی که موجود دارد برحسب مقدار \min چنان باشد $(\sum z_j \leq p(H))$
 (z_j, e_j, e_j) حایه هستند از یک یارِ شین به یارِ شین دیگر (روند)

Run Time

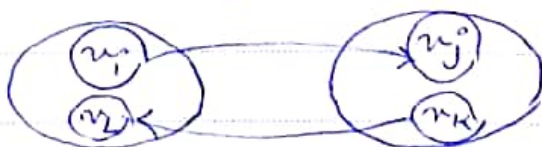
رایه پارتیشن یک $r(p_i)$ غیر منفی است (مثلاً اجرای پارتیشن است)

ordering relation: \leq

برای نمایش ظاهر توان، ordering relation، برای P_i ordering relation.

[illegible]

order مشخص نایم شد $p_1 < p_j$ است و $p_i < p_j$ است چون مراتب مرتبه اول و مرتبه دوم
حد بالا $n_k \leq n_L$ است پس این را نیز می توانیم به دست آوریم و Feasible است.



Year: Month: Day: implementation پیشہ سی Subject: موضوع

→ device. در این صورت bit stream را محاسبه

۱۰۳۰

نصف کنیم با سبزی از task تمام پارتیشن یک به هم میزنیم و این وقت که اتمامش شده باید پارتیشن ۲ را احاطه
لود کنیم پارتیشن ۳ که در خود لود میشه و دیتاهای P_1 را لازم دارد چون یک سرور P_0 بوده نه واسطه
داده ای با P_2 داشته و پس دیتا که P_1 انجام داده دیتایش باید منتقل به پارتیشن دیگری باشه
انتقال از یک پارتیشن به پارتیشن دیگری باید وقتی P_1 اتمامش یافته و در مرحله خاموشی یا پارتیشن خاموش
بوده که بزرگترین نیازها را ندارد اما استفاده کند از این مرحله communication memory
inter configuration register را میخوانند و اینجا که این تابع عمل در سبزی حافظه خارجی ذخیره
میکنند یا در سبزی ۱۵ تا ۱۰۰ FPCAN (مثل یاسین) است که بتواند توسط حافظه
خارجی انجام بدهد.

3 Form 1

سه منبر می که برای دریافت پیکان p_1 و p_2 داریم که $p_1 \leq p_2$ به این صورت است Configuration
اول خود سه را همراهی شده در یک حالتی که باید بر سر آن نرسد Communication mem
Store و Device Observe & reconfigure و سه پیکان p_1 را انجام بدو در برابر
Configuration که انجام می شود سه سبزه Communication mem که در یک حالتی
 p_1 را در نهایت نه دو کان قرارند

3. El nu!

ما می‌توانیم Temporal partitioning را به optimization تبدیل کنیم و برای
partitioning را انجام دهیم که اولاً ordered و feasible باشد و شاید معیارهای را
optimize کنیم. این معیارها می‌توانند به صورت هم‌زمان یا به صورت متوالی باشند.

معیار اول: باید معیارها را به optimize کنیم (معیار اول: باید معیارها را به optimize کنیم)

از این کار می‌توانیم به نفع خاصی یا به زیان خاصی (زمانی که از این کار می‌توانیم به نفع خاصی یا به زیان خاصی)

۱۵۸

از زمان اجرای هر بارش

۱۲ زمان reconfiguration

۱۳ زمان ریاستی exchange

Subject:

Year: Month: Day:

Store و Communication منجر می شود

Connection ها

از پم تا پم را یعنی کم کم کنیم (exchange حالت بین)

زمان خریدن و فروش کم می شود و همین کار را مقدار خطای کم می شود

این مقدار خوبی است که زمان data exchange را کاهش می دهد

مقدار دیتا که باید انجام بگیرد و مقدار دیتا که باید پارسش

مقدار دیتا که باید پارسش را می توانیم minimize کنیم و همین کار

حفظ را از منابع تلف شده (wasted resources) این وقت به وقت

می شود که کمتر و کار را داخل وقت کم می دانیم run میشی

تو کم می شود و زمانی که زودتر تو کم میشی و اختیارش

منطقه با کمترین Core اجرا شده این منابع را می توانیم منجم کنیم

زمان اجرای کل را می توانیم

optimize کنیم

۴۲

wasted resource

مثال کنیم به کار آمدن T_1 و T_2 و T_3 در دام

هر کار آمدن یک مدت زمان اجرای دارد که این زمان را $T(t_i)$

است که توسط کار آمدن به انجام می شود و یک مدت زمان هم زمان

کویم یعنی از اول که configure کنیم تا زمانی که اجرا می شود

پارسی از حالت زمان به وقت

wasted resource

area

نشان می دهند

wasted resource

CACTUS

مثلاً در نمودار این اسلاید می بینیم که area

Scanned by CamScanner

Wasted resource (منهای پهنای باند) = Wasted resource (منهای پهنای باند)

اسم ۳۴

Communication overhead (برای ارتباطات بین یارشی) 8
 Year: / Month: / Day: / Subject: /
 Communication mem

قسمت زمان store, load کردن داده بین Partitioning یک داده را به دو قسمت تقسیم
 یارشی است، که یک نیم به خود و مقدار edge بین آن را برای یک نیم مقدار می‌دهد
 یا یک آن را به دو بخش تقسیم می‌کند 4 تا edge بین یارشی ها است و در است 4 edge است
 بین یک یک است

یک راه دیگر این است Quality را در Partitioning انجام بدهیم. این روش برای هر
 partition رابطه زیر را حساب کنیم

$$Q = \frac{2|E|}{|V|}$$

مقدار edge داخل یارشی (مقدار edge داخل یارشی)

داخلی یارشی edge ها بین یارشی ها ۱۲ - ۲ = ۱۰
 را حساب نمی‌کنیم
 مقدار یارشی

همه اندازه یارشی ها را حساب می‌کنیم

(هر چه مقدار edge ها داخل یارشی بیشتر باشد Quality بهتر است)