بنام خداوند جان و خرد

هم‌طراحی سخت‌افزار نرم‌افزار

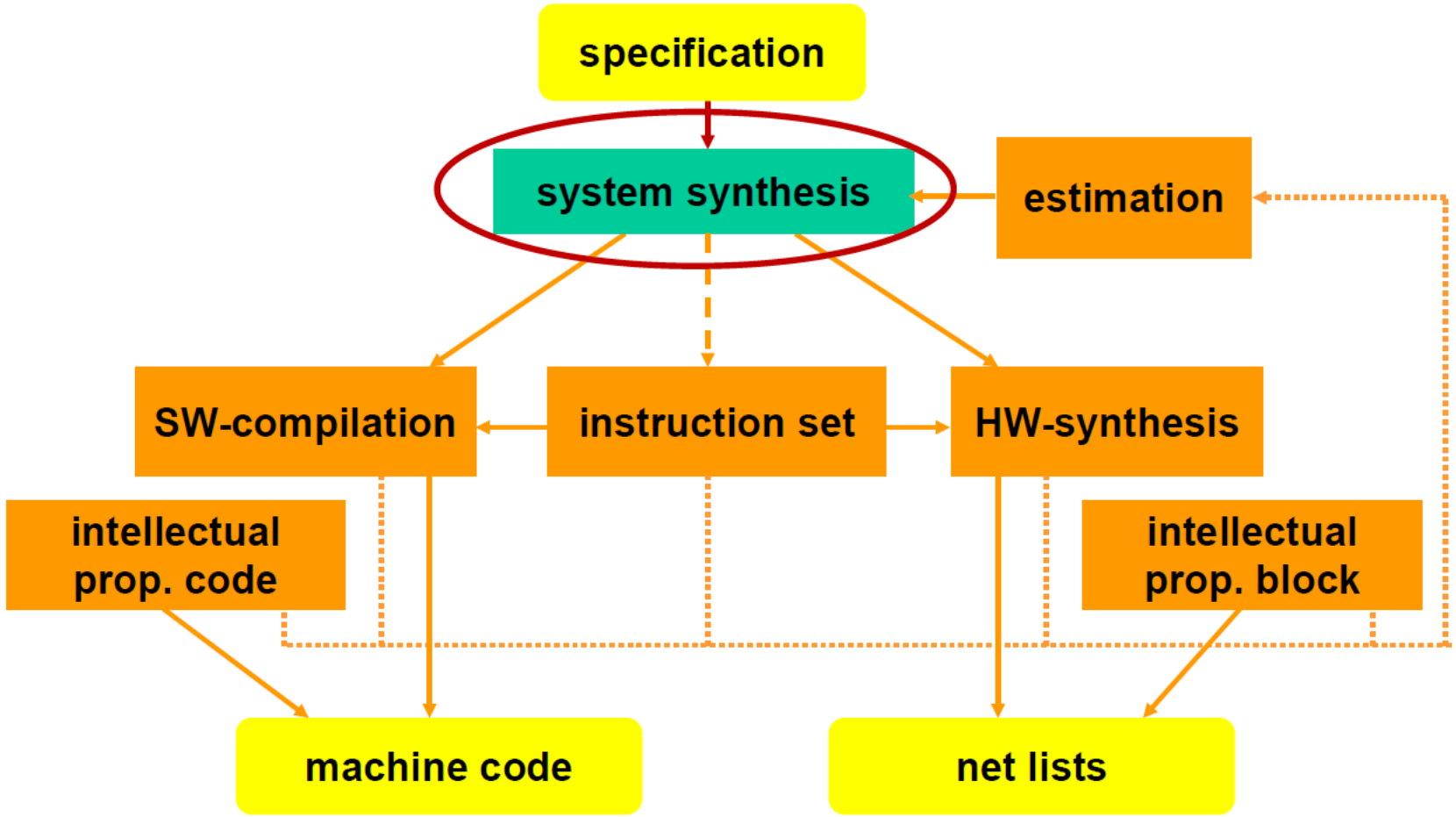
جلسه چهاردهم: الگوریتم‌های Partitioning

ارائه‌دهنده: آتنا عبدی
a_abdi@kntu.ac.ir

مباحث این جلسه

- سنتز توام در روال هم‌طراحی سخت‌افزار و نرم‌افزار
  - فرایند نگاشت و بخش‌بندی (Partitioning)
  - الگوریتم‌های پایه ارائه شده

فرایند هم‌طراحی



فرایند سنتز توأم

- در سه مرحله اصلی انجام می‌شود:
  - تخصیص (Allocation):
    - مشخص کردن اجزای پردازشی که محاسبات روی آن‌ها اجرا می‌شوند
  - بخش‌بندی و نگاشت (Partitioning/ Mapping):
    - تقسیم‌بندی کارکردهای سیستم به واحدهای محاسباتی
    - انتخاب اجزای مناسب برای واحدهای تخصیص داده شده
  - زمان‌بندی (Scheduling):
    - مشخص کردن زمان اجرای محاسبات

فرایند بخش‌بندی در سنتز توأم

- مهم‌ترین گام در مرحله سنتز توأم است:
  - بیشترین تاثیر را بر برقراری مبادله بین کارایی و هزینه سیستم طراحی شده دارد
- در طی بخش‌بندی مشخص می‌شود که:
  - چه بخش‌هایی از مدل بهتر است روی سخت‌افزار و چه بخش‌هایی روی نرم‌افزار پیاده‌سازی شوند
  - معیار: رسیدن به کارایی مدنظر و تحقق تمام محدودیت‌های مشخص شده سیستم
    - محدودیت‌ها: توان مصرفی، هزینه و....
  - فرایند جستجوی فضای طراحی (DSE)

Optimization Strategy -روش‌های بخش‌بندی

- دسته‌بندی روش‌ها براساس استراتژی که در رعایت محدودیت‌ها (کارایی-هزینه) دارند:
  - هدف اولیه: کارایی
    - Primal Strategy
      - **سیستم Vulcan:** تخصیص همه وظایف به ASIC و انتقال تدریجی توابع غیربحرانی به پردازنده با هدف کاهش هزینه
  - هدف اولیه: هزینه
    - Dual Strategy
      - **سیستم Cosyma:** تخصیص همه وظایف به پردازنده و انتقال تدریجی توابع بحرانی به سمت ASIC با هدف افزایش کارایی

Vulcan

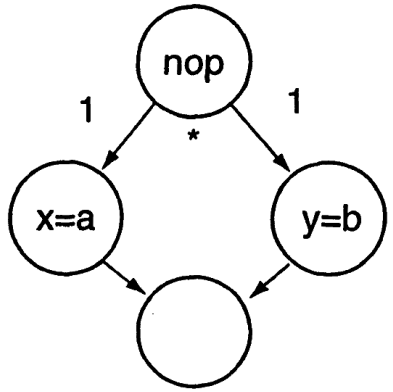
- توسط Gupta و De Micheli در دانشگاه استنفورد ارائه شد
  - شروع بخش‌بندی از یک راهکار مبتنی بر کارایی و تماما سخت‌افزاری
  - حرکت تکراری به سمت جواب‌های نرم‌افزاری با هدف تعدیل کردن هزینه
    - با حذف هم‌روندی‌های نالازم، به سمت نرم‌افزار حرکت می‌کند
  - زبان توصیف سطح سیستم در این ابزار: HardwareC
    - به صورت گراف جریان (مشابه گراف وظیفه) کامپایل می‌شود
    - مجموعه‌ای از توابع سطح پایین مانند ضرب (گره‌ها) و وابستگی داده‌ای بین آن‌ها (یال‌ها)
    - روی هر یال شرط باینری با هدف کنترل موازی‌سازی و قابلیت تعریف محدودیت زمانی روی هر گره

Vulcan-Flow Graph

Two Parallel Assignments

```
x = a; y = b;
```

HardwareC

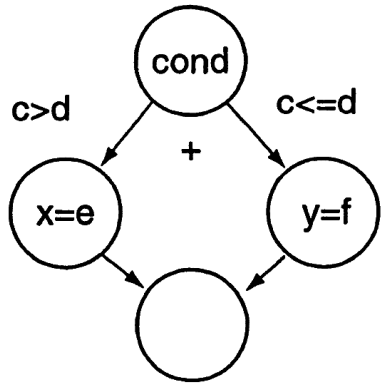


flow graph

Disjunctive Execution

```
if (c > d)
  x = e;
else y = f;
```

HardwareC



flow graph

الگوریتم بخش‌بندی Vulcan

- در این سیستم، گراف جریان به مجموعه‌ای از نخ‌ها (thread) تقسیم می‌شود
- نخ‌ها به عناصر پردازشی تخصیص داده می‌شوند
- تقسیم گراف جریان به نخ‌ها توسط:
  - مکانی که گره wait به‌منظور ایجاد تاخیر برای رخداد خارجی قرار دارد
  - یا سایر نقاط گراف جریان برحسب انتخاب
- معماری هدف:
  - یک پردازنده همراه با چندین عنصر پردازشی ASIC

الگوریتم بخش‌بندی Vulcan (ادامه)

- هدف

- تخصیص نخ‌های گراف جریان به عناصر پردازشی به یکی از دو بخش زیر:
  - بخش CPU (ϕ_S)
  - بخش پردازنده‌های جانبی ASIC (ϕ_H)
- رعایت محدودیت‌های کارایی و هزینه سیستم
- تابع هزینه این روش:

$$f(\omega) = c_1 S_h(\Phi_H) - c_2 S_s(\Phi_S) + c_3 \mathcal{B} - c_4 \mathcal{P} + c_5 |m|$$

سایز HW	سایز SW	نرخ گذردهی باس(<۱)	نرخ گذردهی پردازنده (<۱)	تعداد کل متغیرها

مراحل الگوریتم بخش‌بندی Vulcan

۱- ایجاد بخش اول (initial partition) با تخصیص تمامی نخ‌ها به مجموعه Φ_H

۲- سپس به‌صورت بازگشتی دو مرحله زیر انجام می‌گیرد

۲-۱- انتخاب برخی از عملیات جهت انتقال به Φ_S،

- محاسبه کارایی در بخش‌بندی جدید و انتقال در صورت رعایت محدودیت
- محاسبه کارایی از طریق تخمین بدترین حالت تاخیر در گراف جریان با تخصیص جدید

۲-۲- بروزرسانی تدریجی تابع هدف تعریف شده به‌منظور مشخص شدن تاثیر بخش‌بندی جدید

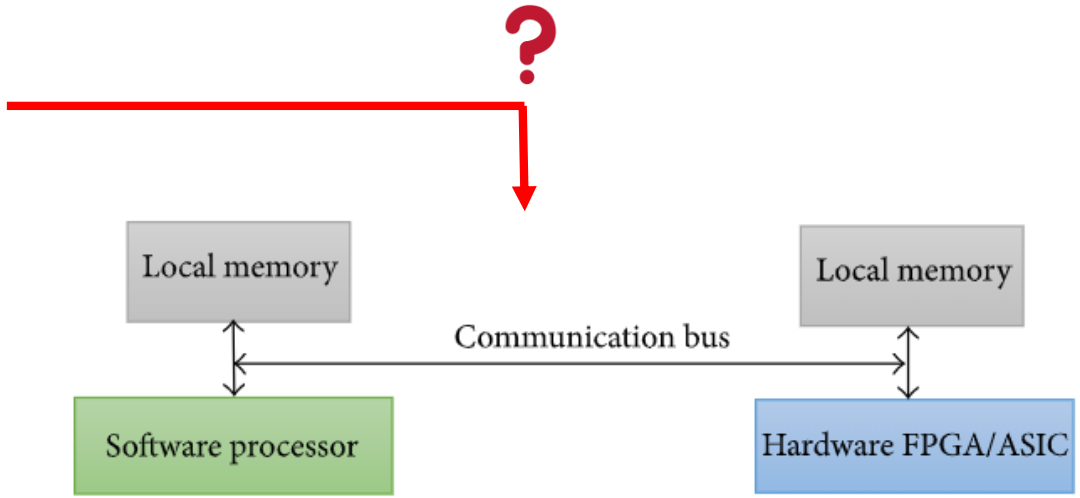
جزئیات الگوریتم بخش‌بندی Vulcan

- با انتقال هر نخ به Φ_S گره‌متصل بعدی آن برای ارزیابی در لیست بعدی قرار می‌گیرند
  - کاهش هزینه ارتباط بین دو بخش
- امکان بازگشت به عقب و انتقال از مجموعه Φ_S به Φ_H وجود ندارد
- آزمایش‌های تجربی انجام شده توسط این الگوریتم نشان داده‌اند که نتیجه
  - بسیار سریع‌تر از تخصیص همه نخ‌ها به نرم‌افزار و تخصیص همه نخ‌ها به سخت‌افزار بوده است

مثالی از الگوریتم بخش‌بندی Vulcan

- سیستم زیر را درنظر می‌گیریم:
  - بخش‌بندی اجرای اجزای سیستم روی CPU یا ASIC با فرض محدودیت هزینه ۱۰۰ واحد ساده ASIC

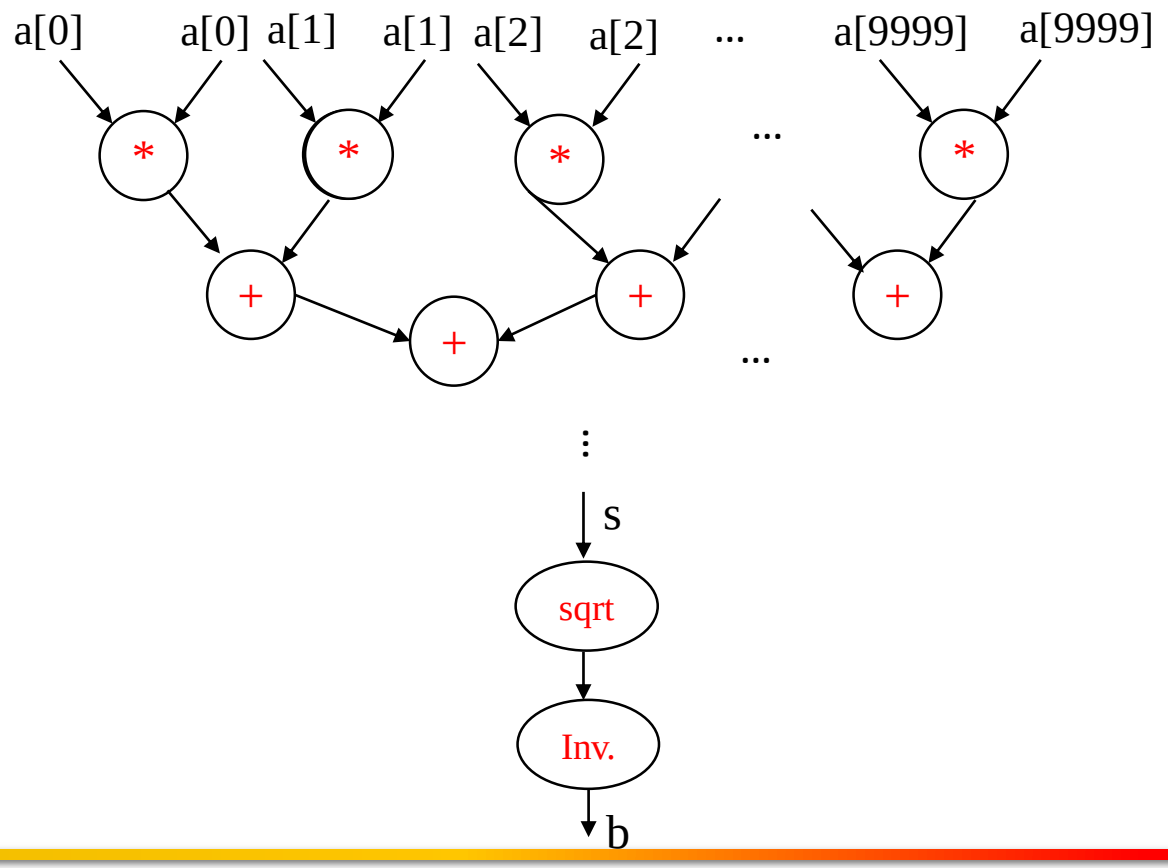
```
for (i=0; i<10000;i++)
      s=s+a[i]*a[i];
b=1/sqrt(s);
```



مثالی از الگوریتم بخش‌بندی Vulcan

- مرحله اول: گراف جریان سیستم

```
for (i=0; i<10000;i++)
        s=s+a[i]*a[i];
b=1/sqrt(s);
```

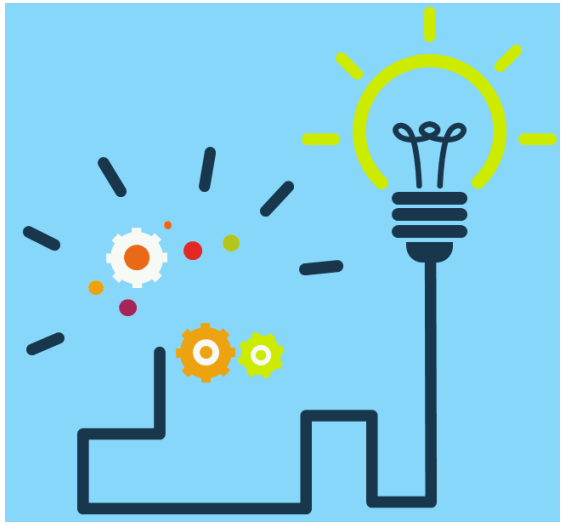


مثالی از الگوریتم بخش‌بندی Vulcan (ادامه)

- شروع با تخصیص تمام نخ‌ها به ASIC
  - نخ‌ها: عملیات ضرب، جمع و عملیات نهایی برای تولید b
- بررسی تابع هزینه و محدودیت‌های سیستم در تصمیم فعلی
  - کارایی = بیشینه حالت ممکن و هزینه = ۱۰۰۰۰ واحد ASIC ✗
- انتقال مجموعه‌ای از نخ‌ها به پردازنده با هدف کاهش واحدهای موازی ASIC
- ادامه و تکرار از مرحله قبل تا برقراری محدودیت

مباحثی که این جلسه آموختیم

- فرایند سنتز توأم
- بخش‌بندی و نگاشت
- الگوریتم‌های مکاشفه‌ای با استراتژی بهینه‌سازی
- Vulcan



مباحث جلسه آینده

- فرایند سنتز توأم
- بخش‌بندی و نگاشت
- الگوریتم‌های بخش‌بندی