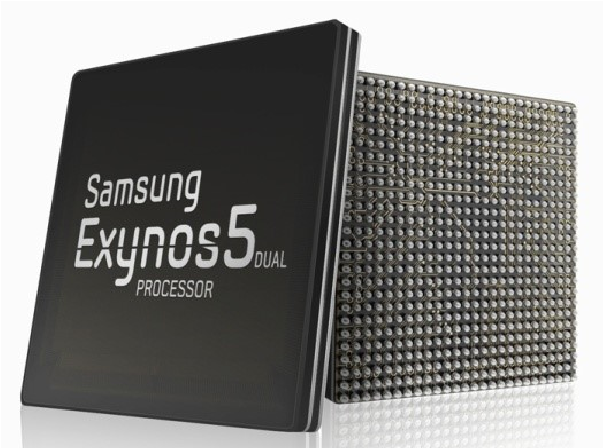
章节 2

**概览OPENCL**

# 这是一个异构的世界

现代的计算平台包括 • 一个或多个 CPU

* 一个或多个 GPU
* DSP 处理器

例：三星® Exynos 5:

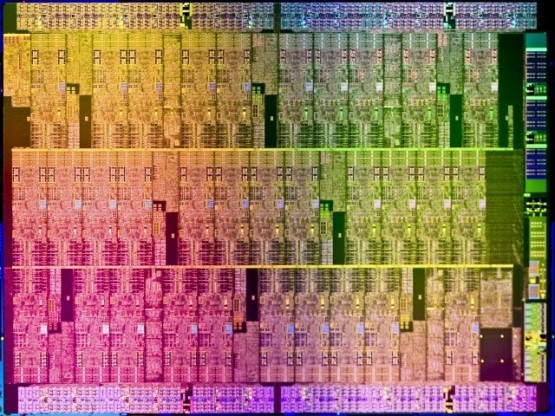
* 加速器(accelerator)
* 双核 ARM A15 1.7GHz,
* … 其他的

Mali T604 GPU 例： Intel XXX with IRIS

**OpenCL 允许开发者写一份可以移植的代码，在不同平台上都可以使用所有的资源。**

微处理器的趋势单个处理器的核心数量越来越多，这些核心还很可能是异构的。

ATI™ RV770

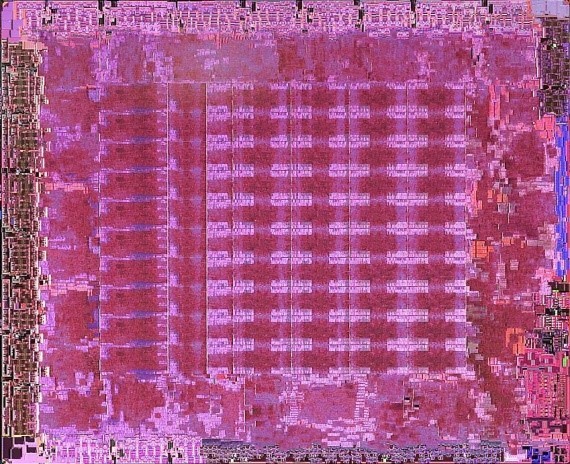


**61**

**cores**

**16**

**wide SIMD**

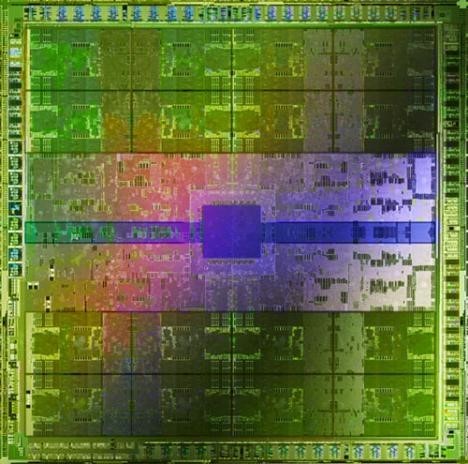


**10**

**cores**

**16**

**wide SIMD**



**16**

**cores**

**32**

**wide SIMD**

Intel® Xeon Phi™协

处理器 NVIDIA® Tesla®

## C2090

异构（Heterogeneous）和众核（many-core）带来的挑战:

如何为这些异构众核设备构建一个软件生态系统？

## 异构平台开发的工业标准

**CPUs**

多核心

**驱动**

性能提升

**GPUs**

通用数据并行

**计**

算日益增加

**图**

形

**API**

和

着色器

**语**

言

多

**处**

理器

**编**

程

例：

**OpenMP**

新兴交叉领域

异构计算



OpenCL – 开放（Open）计算语言（Computing

Language）

针对异构并行计算的CPU，GPU 等设备的可移植的并行计算的开放标准，免受专利限制

OpenCL 的起源

**AMD** 两家公司合并后需

**ARM**

**Nokia**

**IBM**

**Sony**

**Qualcomm**

**Imagination**

**TI**

**+**

**many**

**more**



**ATI**

**NVIDIA**

**Intel**

要对产品的通用支

持

GPU

厂商想要抢占

CPU

市场

CPU

厂商想要抢占

GPU

市场

**写一个草稿，稻草人**

**（straw man）API**

**Khronos 计算组织成立**

厌倦了对不同的众核设备和 GPU 等重复

**Apple** 开发，推动厂商们标准化

Khronos 组织内的 OpenCL 开发团队

* 多种厂商参与

– 处理器厂商，系统设备制造商, 中间件制造商, 应用开发者。

* 由于这些参与者的市场覆盖范围和影响力巨

大，OpenCL 一发布就成了一个重要的标准。



第三方名称是其公司所有财产

# OpenCL 时间线

* 2008年6月正式发布… 从草稿的“稻草人

（StarMan）API” 到 OpenCL 1.0 发布，用了6

个月

* 迅速迭代以紧跟硬件发展的步伐

–从 1.0 到 1.1 ，从 1.1 到 1.2 18 个月

–既定目标: 每隔18-24 个月发布一个新版本

–致力于向后兼容以保障软件开发

**2009年**

多个平台实现多个同样实现

**2008**

**年**

**12**

**月**

**2010**

**年**

**7**

**月**

**2011**

**年**

**11**

**月**

Khronos 公布了 Khronos 公布了正式版 发布了

**OpenCL 1.0** 细节 **OpenCL 1.1** 细节参数，不 **OpenCL 1.2** 参数 久后就可以得到实现

# OpenCL 时间线

* 2008年6月正式发布… 从草稿的“稻草人

（StarMan）API” 到 OpenCL 1.0 发布，用了6

个月

* 迅速迭代以紧跟硬件发展的步伐

–从 1.0 到 1.1 ，从 1.1 到 1.2 18 个月

–既定目标: 每隔18-24 个月发布一个新版本

–致力于向后兼容以保障软件开发

**OpenCL 1.1 发布并 OpenCL 2.0 进行测试 预览版发布**

## 2013年11月

**2008**

**年**

**12**

**月**

**2010**

**年**

**6**

**月**

**2011**

**年**

**11**

**月**

**2013**

**年**

**7**

**月**

**OpenCL 1.0 发布并 OpenCL 1.2 OpenCL 2.0**

**进行测试 发布并进行测试 最终发布并进行测试**

# OpenCL: 从手机到超级计算机

* OpenCL 嵌入式版本针对移动和嵌入式设备设计

–放宽数据类型和精度要求

–避免需要单独的“嵌入式ES”规范

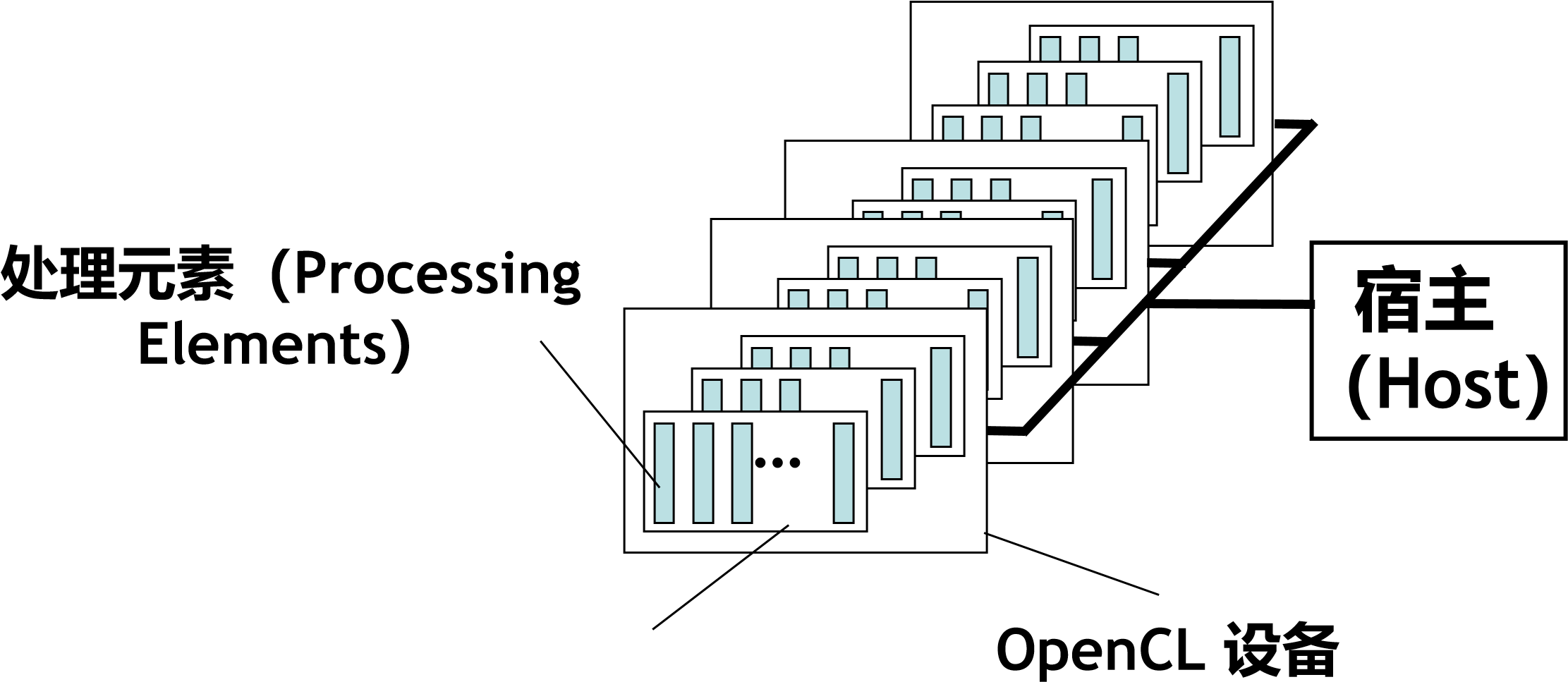
* Khronos API 为图形和图像提供计算支持

–支持高级应用，比如增强现实

– OpenCL 使得新市场领域 建筑物然后标记出来，并从互带有GPS的照相手机可以识别中的并行计算成为可能 联网上获取相关信息

–手机，汽车，航空电子设备

## OpenCL 平台模型



**计算单元（Compute Unites）**

* 一个宿主（***Host***）以及一个或者 *enCL* 设备（*Device*）

–每个 OpenCL 设备都由一个或者多个计算单元（***Compute Units***）组成，计算单元后文常缩写为 ***CU***

* 每个计算单元都可以分成一个或者多个处理元素（***Processing Elements***），
* 内存y分为宿主内存（***host memory***）和设备内存

（***device memory***）

OpenCL 平台样例 (一个节点，双路 CPU，双 GPU)

**CPU:**

* 看做一个 OpenCL 设备(device)
  + 每个核心（core）一个计算单元

（CU）

* + 每个计算单元一个处理元素

（PE），或者如果 PE 映射到单指令多数据流（SIMD lanes），则每个 CU 有*n*个，这里的*n* 等同于单指令多数据流的带宽

（ SIMD width ）

* 记住：
  + CPU本身也是自己的宿主（Host）

**GPU:**

* 每个 GPU 是一个单独的

OpenCL 设备（device）

* 通过 OpenCL 可以同时使用 CPU 和所有 GPU 设备

**CU = Compute Unit 计算单元 PE = Processing Element 处理元素**

练习 1: 获取平台信息

* 目标:

–验证一下你能够运行一个简单的 OpenCL 程序。

* 流程:

–利用已经提供的**DeviceInfo**程序，选用一个编辑器来阅读一下代码，然后构建并运行。

* 预期输出:

–应该会输出你所安装的 OpenCL 平台信息以及所有可见设备的信息。

* 扩展:

–运行命令**clinfo**，这个命令来自于 AMD 公司提供的 SDK，但可以运行于所有 OpenCL 平台之上。这个命令运行后你会看到 OpenCL 运行环境能找到的所有设备和平台的相关信息。