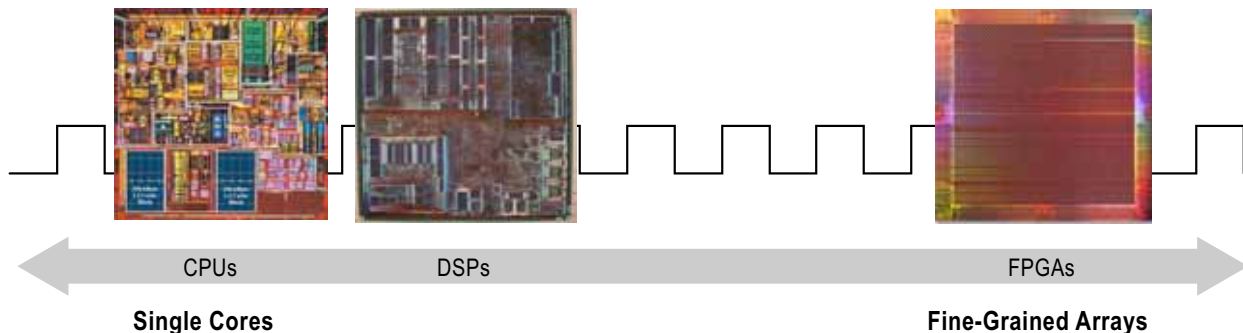


利用 FPGA 上的 Khronos 集团 OpenCL™ 标准，与目前的 CPU、图形处理单元 (GPU) 和数字信号处理 (DSP) 单元等硬件体系结构相比，能够大幅度提高性能，同时降低了功耗。此外，与使用 Verilog 或者 VHDL 等底层硬件描述语言 (HDL) 的传统 FPGA 开发方法相比，使用 OpenCL 标准、基于 FPGA 的混合系统 (CPU + FPGA) 具有明显的产品及市场优势。

引言

在可编程技术发展的最初阶段，可编程能力出现了两个极端。如图 1 所示，一个极端的代表是单核 CPU 和数字信号处理 (DSP) 单元。这些器件使用含有一系列可执行指令的软件来进行编程。对于编程人员，以概念上连续的方式来开发这些指令，而高级处理器能够对指令重新排序，在运行时从这些连续程序中提取出指令级并行处理操作。作为对比，可编程技术另一极端的代表是 FPGA。通过开发可配置硬件电路对这些器件编程，完全并行执行。使用 FPGA 的设计人员实际上是开发粒度非常精细的并行应用。多年以来，这两个极端同时存在，每一类型的可编程功能适用于不同的应用领域。但是，最近的技术发展趋势表明，有更好的技术同时实现了可编程和并行处理操作。

图 1. 可编程技术的早期状况

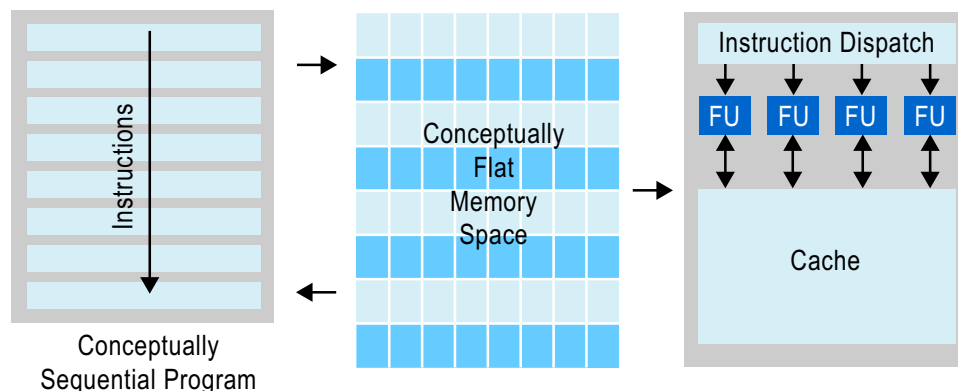


随着对性能需求的增长，执行连续程序的软件可编程器件越来越需要依靠两种基本趋势来提高其性能。第一种是随着工艺代的发展而调整工作频率。出于各种原因考虑，不可能持续的降低工作电压，也不可能提高工作频率同时维持合理的功率密度。这一现象被称为“功率墙”，对所有类型可编程器件的体系结构都会产生很大的影响。

软件可编程器件依靠的第二种趋势是复杂硬件的出现，从连续程序中提取出指令级并行处理操作。如图 2 所示，单核体系结构输入指令流，在器件中执行它们，这些器件会有很多并行功能单元。处理器硬件的很大一部分必须专门用于从连续代码中动态提取出并行处理操作。

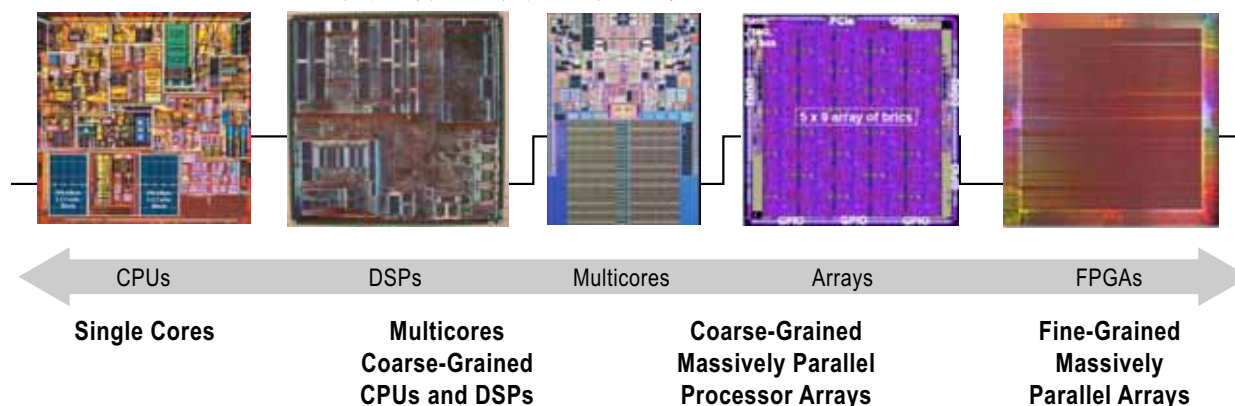
此外，硬件还会尝试去补偿存储器延时。一般而言，编程人员开发程序时没有考虑处理器的底层存储器结构，好像只有大规模的统一快速存储器。相比较而言，处理器必须处理实际延时，以及与外部存储器的有限带宽链接。为保持功能单元能够传送数据，处理器必须从外部存储器中预先获取数据，放入片内高速缓存中，这样，数据更接近要进行计算的地方。使用这些技术，性能经过多年的提高后，这类体系结构的改动已经不大。

图 2. 单核体系结构



在传统处理器体系结构上，这两种趋势的优势日益减小，我们开始寻找各种软件可编程器件，这些器件的发展非常快，如图 3 所示。重点是从运行时自动提取指令级并行处理操作，发展到在编码时明确的找到线程级并行处理操作。开始出现高度并行的多核器件，一般趋势是含有多个简单处理器，很多晶体管专门用于计算，而不是采用高速缓存，提取并行处理操作。这些器件包括一般含有 2、4 或者 8 个内核的多核 CPU，以及含有数百个适用于数据并行计算的简单内核的 GPU 等。为能够在这些多核器件上实现高性能，编程人员必须以并行方式清晰的对实际应用进行编程。每一内核都必须分配一定的工作，这样，所有内核能够协同工作，执行某一计算。这也是 FPGA 设计人员在开发其高级系统体系结构时所做的工作。

图 3. 可编程和并行技术最近的发展趋势



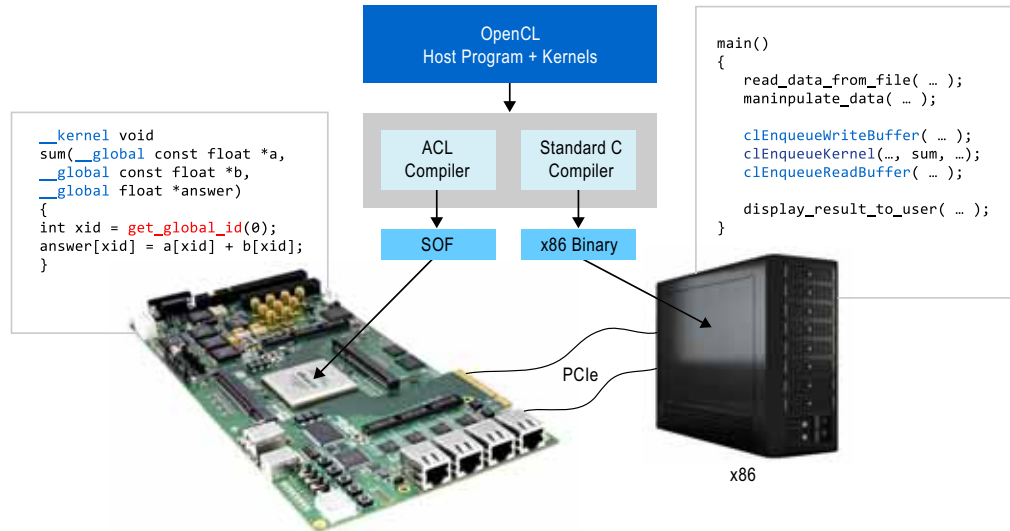
考虑到在多核新时代开发并行程序的需求，应该有标准模型来开发程序，能够在所有这些完全不同的器件上执行。缺乏能够适用于这些不同可编程技术的标准，这一一直困扰着编程人员。在 2008 年夏天，Apple 向 Khronos 集团提出了 OpenCL（开放计算语言）草案规范建议，努力开发跨平台并行编程标准。Khronos 集团包括 Apple、IBM、Intel、AMD、NVIDIA、Altera 等很多其他业界联盟成员。这一集团负责制定 OpenCL 1.0、1.1 和 1.2 规范。OpenCL 标准支持实现并行算法，可以在不同平台之间导入导出，对代码的改动很小。这一语言基于 C 编程语言，进行了扩展，支持并行规范。

除了提供可移植模型，OpenCL 标准还能够自然的描述在 FPGA 中实现的并行算法，其抽象级要比 VHDL 或者 Verilog 等硬件描述语言 (HDL) 高得多。虽然有很多高级综合工具能够实现高等级的抽象功能，但是都存在同样的基本问题。这些工具会采用连续 C 程序，产生并行 HDL 实现。在开发 HDL 时，困难还不是很明显，但是，提取出线程级并行处理操作在 FPGA 中实现以提高性能时，困难却非常大。而 FPGA 的并行功能非常强大，与其他器件相比，在尽可能提取并行功能时出现任何失败的后果都非常严重。OpenCL 标准能够解决很多这类问题，它支持编程人员明确的设定并控制并行处理操作。与纯 C 语言描述连续程序相比，OpenCL 标准能够更自然的匹配 FPGA 的高度并行特性。

OpenCL 标准简介

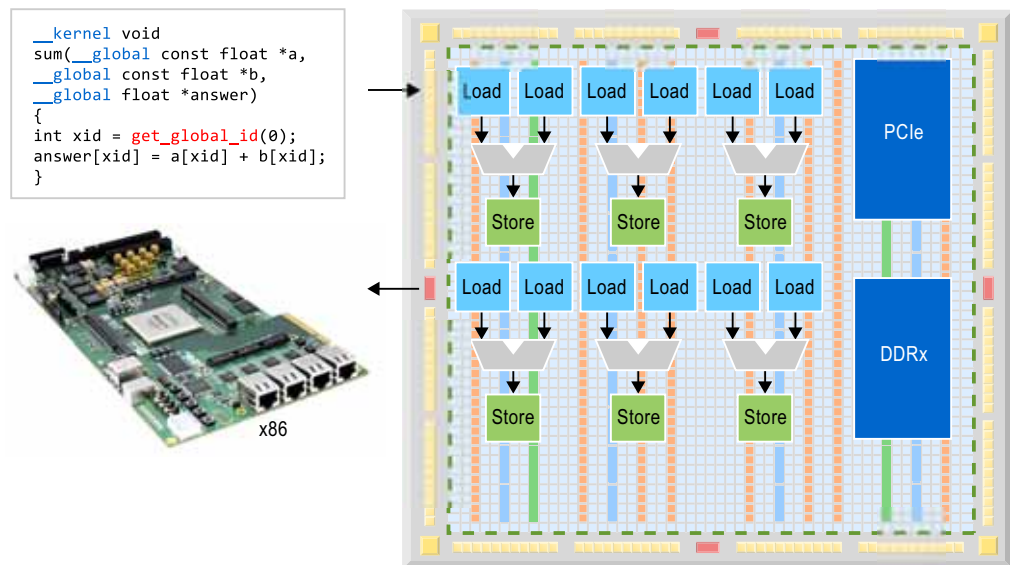
OpenCL 应用程序含有两部分。OpenCL 主程序是纯软件例程，以标准 C/C++ 编写，可以运行在任何类型的微处理器上。例如，这类处理器可以是 FPGA 中的嵌入式软核处理器、硬核 ARM 处理器或者外置 x86 处理器，如图 4 所示。

图 4. OpenCL 简介



在这一主软件例程执行期间的某一点，某一功能有可能需要很大的计算量，这就可以受益于并行器件的高度并行加速功能，例如 CPU、GPU、FPGA 等器件。要加速的功能被称为 OpenCL 内核。采用标准 C 编写这些内核；但是，采用结构对其进行注释，以设定并行处理操作和存储器等级。图 5 中的例子对两个数组 a 和 b 进行矢量加法，将结果写回输出数组应答中。矢量的每一元素都采用了并行线程，当采用像 FPGA 这类具有大量精细粒度并行单元的器件进行加速时，能够很快的计算出结果。主程序使用标准 OpenCL 应用程序接口 (API)，支持将数据传送至 FPGA，调用 FPGA 内核，传回得到的数据。

图 5. 在 FPGA 上实现的 OpenCL 例子



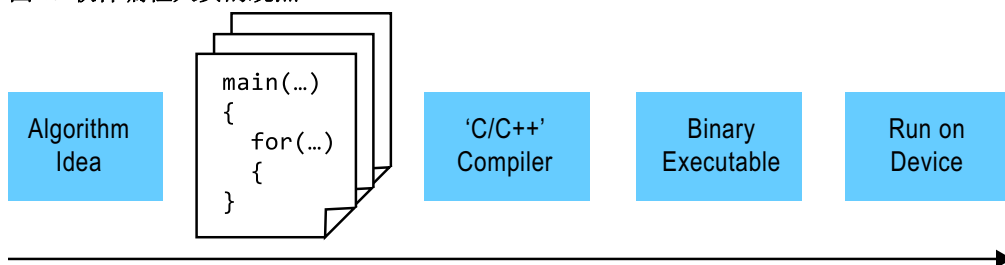
 Khronos 集团网站 (www.khronos.org/opencl/) 对 OpenCL 标准进行了详细的介绍。

与 CPU 和 GPU 不同，其并行线程可以在不同的内核中执行，而 FPGA 能够提供不同的策略。可以把内核功能传送到专用深度流水线硬件电路中，它使用了流水线并行处理概念，在本质上就是多线程的。这些流水线的每一条都可以复制多次，与一条流水线相比，提供更强的并行处理功能。如图 5 所示，可以通过级联功能单元实现矢量加法内核，在 OpenCL 描述中实现每一操作，进行复制以满足实际应用的吞吐量和延时要求。虽然所显示的只是一个简单表征，但每个功能单元都可以是深度流水线，以保证最终电路的工作频率足够高。此外，编译器可以建立电路来管理与外部系统的通信。在这个例子中，DDR_x 控制器和 PHY 连接至内核，使其能够高效访问片外阵列。类似的，PCI Express[®] (PCIe[®]) IP 自动例化，连接至内核，这样，x86 主机能够通过 OpenCL API 与 FPGA 加速器进行通信。

在 FPGA 上实现 OpenCL 标准的优势

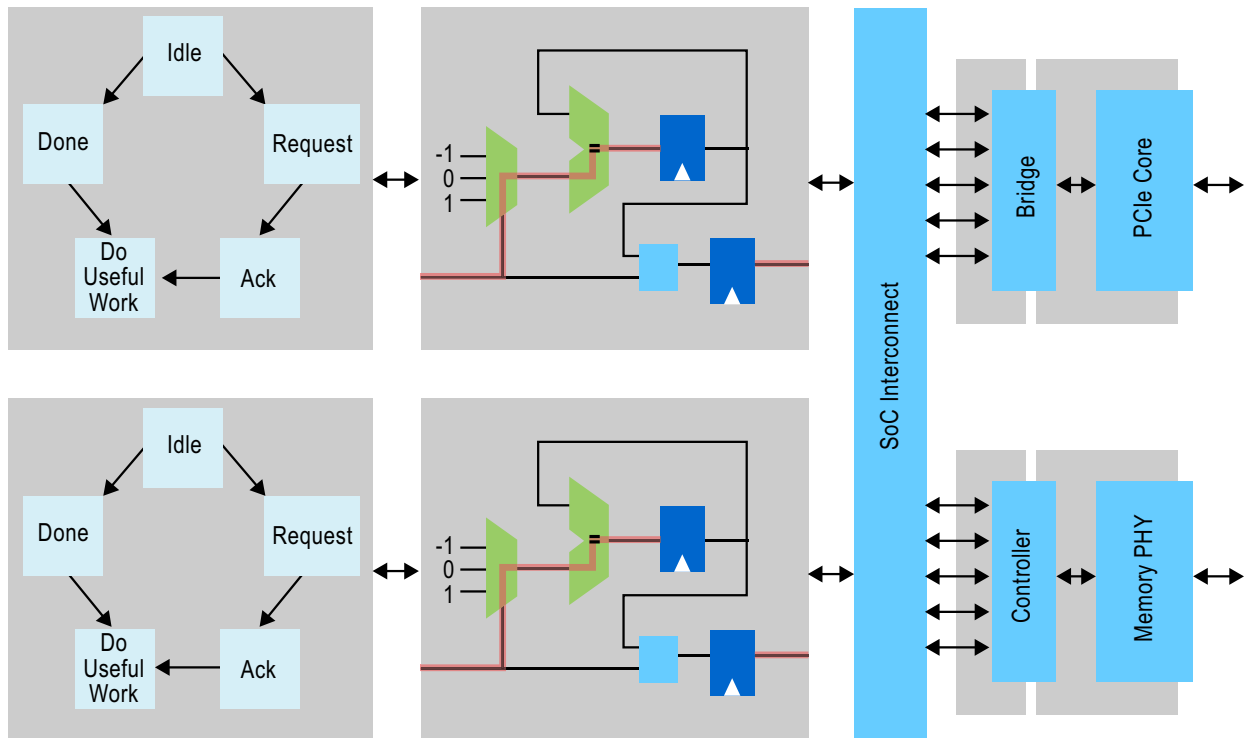
使用 OpenCL 描述来开发 FPGA 设计，与基于 HDL 设计的传统方法相比，具有很多优势。最显著的优势如图 6 所示。开发软件可编程器件的流程一般包括进行构思、在 C 等高级语言中对算法编程，然后使用自动编译器来建立指令流。

图 6. 软件编程人员的观点



这一方法可以与传统基于 FPGA 的设计方法相比。这里，设计人员的主要工作是对硬件按照每个周期进行描述，用于实现其算法。传统流程涉及到建立数据通路，如图 7 所示，通过状态机来控制这些数据通路，使用系统级工具（例如，SOPC Builder、Platform Studio）连接至底层 IP 内核，由于必须要满足外部接口带来的约束，因此，需要处理时序收敛问题。OpenCL 编译器的目的是帮助设计人员自动完成所有这些步骤，使他们能够集中精力定义算法，而不是重点关注乏味的硬件设计。以这种方式进行设计，设计人员很容易移植到新 FPGA，性能更好，功能更强，这是因为 OpenCL 编译器将相同的高级描述转换为流水线，从而发挥了 FPGA 新器件的优势。

图 7. FPGA 设计方法



案例：Monte Carlo Black-Scholes 方法

在金融市场上最重要的一个基准测试方法是通过 Monte Carlo Black-Scholes 方法计算期权价格。该方法基于对底层股票价格的随机仿真，以及数百万不同路径上的平均预期收益。图 8 以图形化的方式显示了这类仿真的一个例子。

图 8. Monte Carlo 仿真

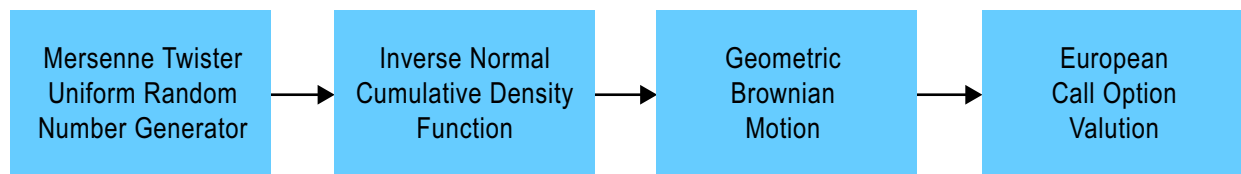
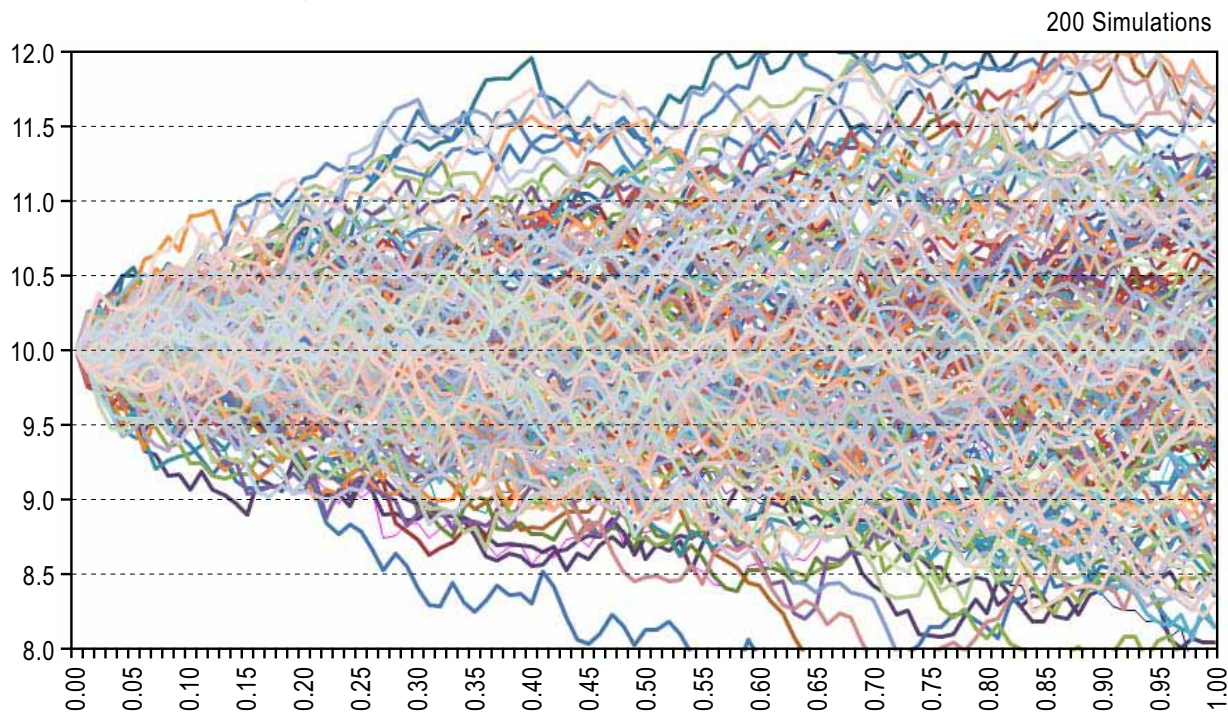


图 9 显示了进行这一计算的高级算法结构。首先采用 Mersenne 旋转随机数发生器来创建均匀分布的数值。将随机数序列送入逆正态累积密度函数，以产生正态分布序列。然后，使用几何布朗运动，这些随机数用于仿真股票价格的变化。在每一仿真通路的最后，记录看涨期权的收益，进行平均来产生收益预期值。整个算法通过大约 300 行的 OpenCL 代码来实现，可以从 FPGA 移植到 CPU、GPU。

图 9. 算法结构



~100,000 simulations may be required to achieve a result that is accurate enough

利用针对 Altera® FPGA 开发的 OpenCL 工作台，可以产生很好的基准测试结果，如表 1 所示。与相应的 GPU 相比，面向 Stratix® IV FPGA EP4SGX530 的 OpenCL 工作台在吞吐量上超过了 CPU 和 GPU。与相应的 GPU 相比，在执行相同的代码时，FPGA 解决方案不但提高了吞吐量，保守估计，功耗也只有其五分之一。速率和高功耗相结合，降低了大计算量应用的功耗需求。

表 1. Monte Carlo Black-Scholes 结果

OpenCL Monte Carlo Black-Scholes	四核 Core Xeon	相应的 GPU	Stratix IV 530
仿真 / 秒	240M	950M	2,200M
器件峰值 GFLOPS	160	500	200

结论

利用 FPGA 上的 OpenCL 标准，与目前的硬件体系结构（CPU、GPU，等）相比，能够大幅度提高性能，同时降低了功耗。此外，与使用 Verilog 或者 VHDL 等底层硬件描述语言（HDL）的传统 FPGA 开发方法相比，使用 OpenCL 标准、基于 FPGA 的混合系统（CPU + FPGA）具有明显的产品及及时面市优势。Altera 于 2010 年加入 Khronos 集团，为标准建设做出了积极贡献。请在 www.altera.com/opencl 上进行注册，了解 Altera OpenCL 在 FPGA 开发上的最新信息。

详细信息

- Altera 的 OpenCL 计划：
www.altera.com/opencl

- Khronos 集团——OpenCL 标准：
www.khronos.org/opencl/

致谢

- Deshanand Singh, 首席督导工程师, 软件和 IP 工程, Altera 公司。

文档修订历史

表 2 列出了本文档的修订历史。

表 2. 文档修订历史

日期	版本	进行的修改
2011 年 11 月	1.0	初次发布。