

## 在嵌入式系统中增加硬件加速器，降低功耗

*并不是所有的功能都同样适用于解决电路频率问题。当硬件能够同时执行几个操作时，并行工作的功能会运行的更快一些，在一定时钟速率下，性能会更好，而在性能相同时，可以降低时钟速率。在芯片设计中增加硬件能够降低功耗要求，同时保持性能不变。*

### 引言

人们在嵌入式系统上的设计经验一直是认为增加硬件会提高功耗。但是，仔细的使用硬件加速器打破了这一经验：增加硬件会降低功耗。通过分析算法，在可编程逻辑中实现合适的加速器，开发人员不但提升了嵌入式计算系统的设计性能，而且同时降低了功耗。测试结果表明，加速器扩展了综合考虑选项，从相同功耗下性能提高 200 倍到相同性能时功耗降低 90%。

由于历史原因，可编程逻辑一直背负了高功耗逻辑设计方法这一名声。经验认为，在一定的工艺技术条件下，集成电路的功耗大致与芯片面积成正比，可编程逻辑实现的设计规模一般要大于硬线逻辑。这虽然是暗示，但实际是一种误导。

对于集成电路，比面积相关功耗更重要的是频率相关功耗。当晶体管切换状态时，CMOS 电路吸收大部分电流，因此电路工作频率对功耗的影响要远远大于芯片面积的影响。频率越高，功耗要求就越大。这样，设计人员有可能通过增加电路来降低功耗，前提条件是增加硬件能够显著降低时钟速率。

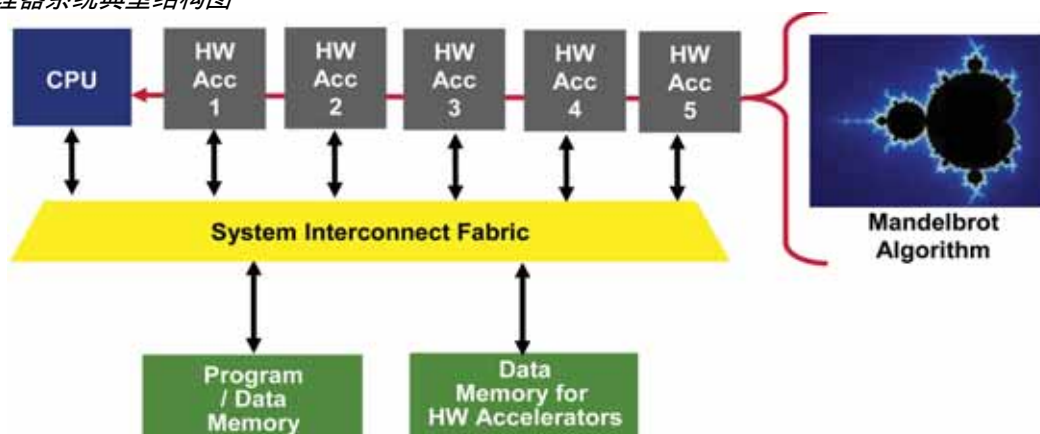
多年以来，嵌入式处理器依靠定制硬件来加速常用的算法，例如，图像或者信号处理等，在单位时钟周期中完成更多的工作。这一方法虽然提高了系统性能，但是没有降低系统时钟或者动态功耗。如果采用硬件能够加速软件算法同时降低时钟频率，那么，不但能降低功耗还同时满足了系统性能要求。

然而，并不是所有的功能都同样适用于解决电路频率问题。对于顺序处理，在开始下一步骤之前必须完成本步骤，增加电路带来的好处不大。另一方面，当硬件能够同时执行几个操作时，并行工作的功能会运行的更快一些。这意味着，在一定时钟速率下，性能会更好，而在性能相同时，可以降低时钟速率在芯片设计中增加硬件能够降低功耗要求，同时保持性能不变。

### Mandelbrot 实例

为演示设计人员取得的这类低功耗效果，使用 50-MHz Altera EP3C25F324 开发了基于低成本 FPGA 的设计实例，它具有 25K 逻辑单元 (LE)、66 个 M9K 存储器模块 (0.6 Mbits)、16 个 18x18 乘法器模块以及 4 个 PLL。设计使用 Altera® Nios® II 嵌入式处理器执行 Mandelbrot 算法，计算不规则碎片。虽然使用了规模相对较小的 FPGA，微处理器也只占用了部分 FPGA 资源。这为增加其他硬件来加速算法执行留下了空间 (图 1 所示)。

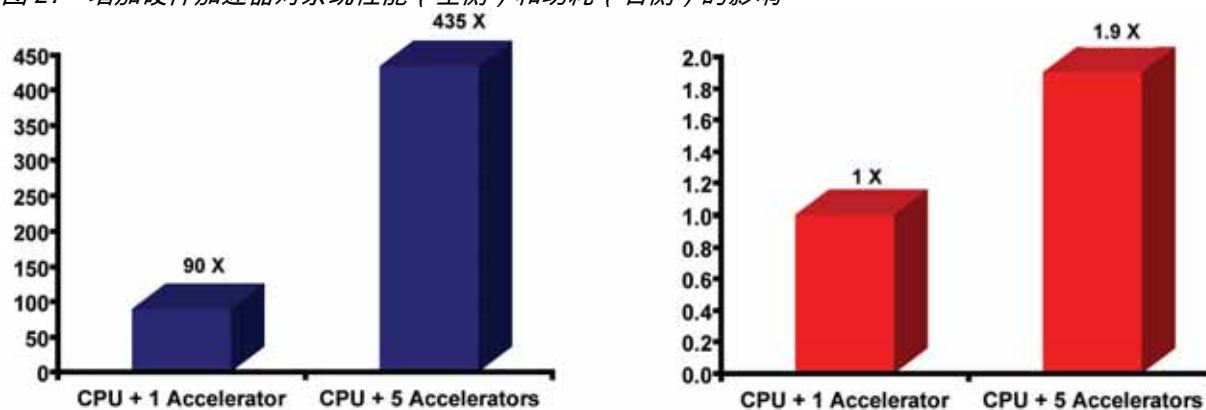
图 1. 处理器系统典型结构图



采用一个处理器，以及一个处理器和 5 个硬件加速器分别进行评估。Altera Cyclone® III 和 Stratix® III 产品系列容量远远大于测试器件，设计人员可以更全面的进行综合考虑。

基线测试表明，Nios II 处理器单独工作时需要 4.35 亿时钟周期来能够完成一帧 Mandelbrot 的计算。增加一个硬件加速器后，运算要求降到 4 百 90 万时钟周期——性能提高了近 90 倍（图 2，左侧），而功耗没有明显增加。与一个处理器相比，再增加四个硬件加速器后，性能提高了 435 倍。增加的加速器功耗仅比 CPU 多出 90%（图 2，右侧）。

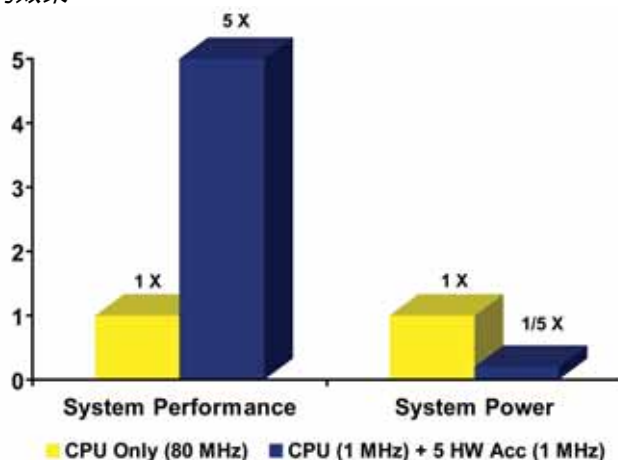
图 2. 增加硬件加速器对系统性能（左侧）和功耗（右侧）的影响



### 降低系统时钟频率

性能提高 435 倍后，大大提高了计算余量，从而能够进一步降低功耗。一种实现方法是降低整个设计的时钟。如图 3 所示，即使采用一个加速器，整个设计可以运行在 1 MHz，而性能仍然优于仅采用一个 80 MHz CPU 的情况。

图 3. 降低系统时钟频率的效果



同时，大大降低了功耗。与运行在 80 MHz 的一个 CPU 功耗 132 mW 相比，设计采用 CPU 和一个加速器，运行在 1 MHz 时，功耗只有 12 mW，而性能提高了近两倍。如果考虑 5 个加速器的设计，功耗能够降低不到一个 CPU 的五分之一，性能提高 5 倍以上。

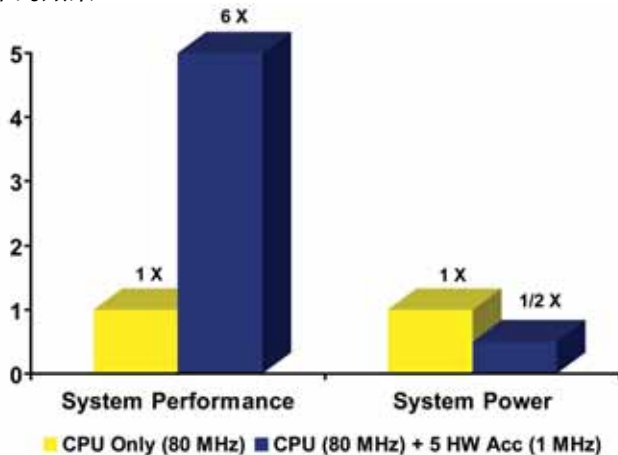
### 降低加速器时钟频率

然而，在很多应用中，加速硬件只对部分算法有效果。在这种情况下，降低整个设计的时钟频率会对其他功能有不利影响。而实际情况是应用软件有可能要求处理器采用更高的时钟频率来运行。此时，仍然可以通过降低加速器时钟频率进一步降低功耗。

开发人员可以评估各种硬件模块在不同时钟频率下对性能和功耗的影响。这一设计实例中使用的 FPGA 支持多时钟域，因此，CPU 及其加速硬件可以分别采用自己的最佳速度进行工作。通过调整每个域的时钟速率，开发人员可以确定需要的最小功耗，同时获得满意的性能。

考虑这样一个实例，嵌入式设计人员希望处理器以 80 MHz 执行程序，将大计算量的算法卸载到时钟频率较低的硬件中。在测试中，嵌入式处理器以 80 MHz 运行应用程序代码，5 个硬件加速器以 1 MHz 运行，系统性能提高了 6 倍，同时系统功耗降低了 55% (图 4)。

图 4. 降低加速器时钟频率的效果

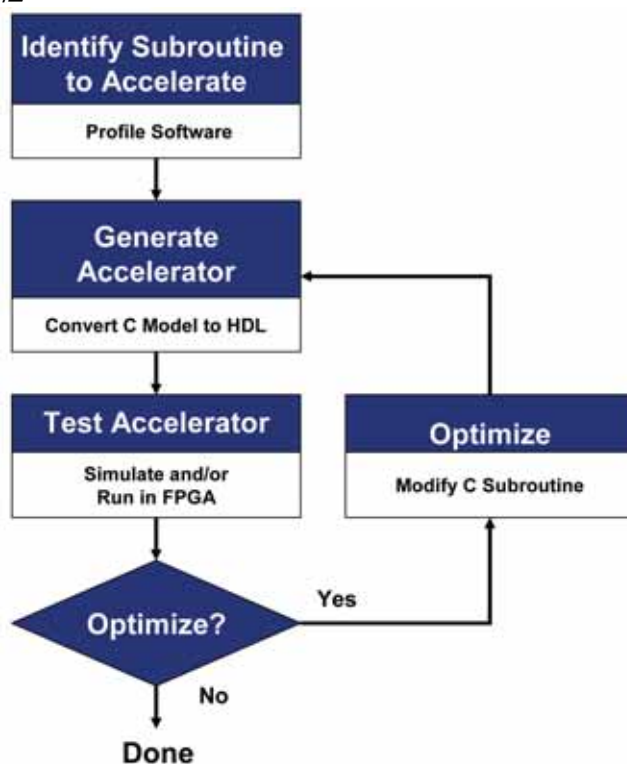


## 开发硬件加速器

第一步是检查子程序，找到计算或者状态机算法。这最有可能受益于硬件加速。开发人员确定目标子程序后，标准软件分析工具会提供加速处理的详细信息。执行时间较长的循环比较适合进行加速处理。

建立硬件加速器是硬件开发人员比较熟悉的工作，而对采用 C 语言进行算法建模的软件开发人员而言却是繁琐的工作。在实现高效的加速硬件时，将 ANSI C 算法转换为相应的逻辑功能是比较关键的步骤（图 5）。

图 5. 硬件加速器设计流程



使用高级工具包来实现应用实例，包括嵌入式处理器 C 语言至硬件加速编译器 (C2H) 以及 SOPC Builder 系统开发工具等，这都包含在 Altera 的 Quartus® II 设计软件中。工具将嵌入式处理器 C 源代码转换为在 HDL 中实现的硬件加速器。

同时将加速器集成到处理系统中。在系统中建立 HDL 后，可以使用 HDL 仿真工具进行评估，或者直接在 FPGA 硬件中运行。哪种方式都可以对结果进行评估；反复修改代码，进行迭代，直至获得系统所需要的性能和功耗。

### 发挥并行处理的优势

除了在关键算法步骤中开发加速器硬件设计，开发人员还应该考虑复制使用这类加速器。开发了 HDL 代码后，比较容易在设计中复制多个加速器，这样，评估时间会显著减少。如果算法的并行度很高，例如 Mandelbrot 实例，那么，并行加速器能够大大降低功耗，正如实例所演示的。

### 结论

通过使用硬件加速器，设计人员改变了“电路越多，功耗越大”的老看法。他们能够在很多设计中全面考虑性能和功耗，在需要小外形封装和电池供电的新应用中采用低功耗 FPGA。

## 详细信息

- AN 351 : 采用硬件加速器来降低功耗 :  
[www.altera.com/literature/an/an531.pdf](http://www.altera.com/literature/an/an531.pdf)
- Nios II 低功耗设计实例 :  
[www.altera.com/literature/an/power.zip](http://www.altera.com/literature/an/power.zip)

## 致谢

- Rodney Frazer , 嵌入式专家 FAE , 销售 , Altera 公司。



101 Innovation Drive  
San Jose, CA 95134  
[www.altera.com](http://www.altera.com)

版权 © 2009 Altera 公司。保留所有版权。Altera、可编程解决方案公司、程式化 Altera 标识、专用器件名称和其他所有其他专有商标或服务标记，除非特别声明，均为 Altera 公司在美国和其他国家的商标和服务标记。所有其他产品或服务名称的所有权属于其各自持有人。Altera 产品受美国和其他国家多种专利、未决应用、模板著作权和版权的保护。Altera 保证当前规范下的半导体产品性能与 Altera 标准质保一致，但是保留对产品或服务在没有事先通知时的升级变更权利。除非与 Altera 公司的书面条款完全一致，否则 Altera 不承担由此处所述信息、产品或服务导致的责任。Altera 建议客户在决定购买产品或服务，以及确信任何公开信息之前，阅读 Altera 最新版的器件规范说明。