

---

# FPGA 系统设计的三个原则

## 一. 面积与速度的平衡互换原则

这里的面积指的是 FPGA 的芯片资源，包括逻辑资源和 I/O 资源等；这里的速度指的是 FPGA 工作的最高频率（和 DSP 或者 ARM 不同，FPGA 设计的工作频率是不固定的，而是和设计本身的延迟紧密相连）。在实际设计中，使用最小的面积设计出最高的速度是每一个开发者追求的目标，但是“鱼和熊掌不可兼得”，取舍之间展示了一个开发者的智慧。

### 1. 速度换面积

速度优势可以换取面积的节约。面积越小，就意味着可以用更低的成本来实现产品的功能。速度换面积的原则在一些较复杂的算法设计中常常会用到。在这些算法设计中，流水线设计常常是必须用到的技术。在流水线的设计中，这些被重复使用但是使用次数不同的模块将会占用大量的 FPGA 资源。对 FPGA 的设计技术进行改造，将被重复使用的算法模块提炼出最小的复用单元，并利用这个最小的高速代替原设计中被重复使用但次数不同的模块。当然，在改造的过程中必然会增加一些其他的资源来实现这个代替的过程。但是只要速度具有优势，那么增加的这部分逻辑依然能够实现降低面积提高速度的目的。

可以看到，速度换面积的关键是高速基本单元的复用。

### 2. 面积换速度

在这种方法中面积的复制可以换取速度的提高。支持的速度越高，就意味着可以实现更高的产品性能。一些注重产品性能的应用领域可以采用并行处理技术，实现面积换速度。

## 二. 硬件可实现原则

FPGA 设计通常会使用 HDL 语言，比如 Verilog HDL 或者 VHDL。当采用 HDL 语言来描述一个硬件电路功能的时候，一定要确保代码描述的电路是硬件可实现的。

Verilog HDL 语言的语法与 C 语言很相似，但是它们之间有着本质的区别。C 语言是基于过程的高级语言，编译后可以在 CPU 上运行。而 Verilog HDL 语言描述的本身就是硬件结构，编译后是硬件电路。因此，有些语句在 C 语言的环境中应用是没有问题的，但是在 HDL 语言环境下就会导致结果不正确或者不理想。如：

```
for (i=0;i<16;i++)
```

```
DoSomething();
```

---

在 C 语言中运行没有任何问题，但是在 Verilog HDL 的环境下编译就会导致综合后的资源严重浪费。

### 三. 同步设计原则

同步电路和异步电路是 FPGA 设计的两种基本电路结构形式。

异步电路的最大缺点是会产生毛刺。同步设计的核心电路是由各种触发器构成的。这类电路的任何输出都是在某个时钟的边沿驱动触发器产生的。所以，同步设计可以很好地避免毛刺的产生。

OFweek 电子工程网