

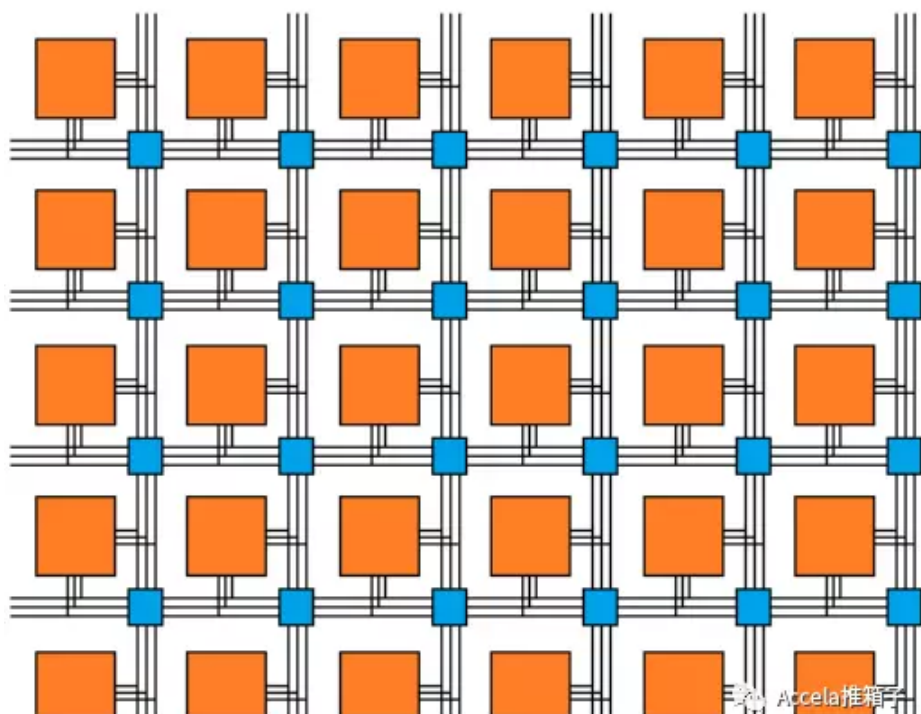
## 有意思的论文FPGA Catapult (P2)

Original 2018-04-03 Accela Zhao Accela推箱子

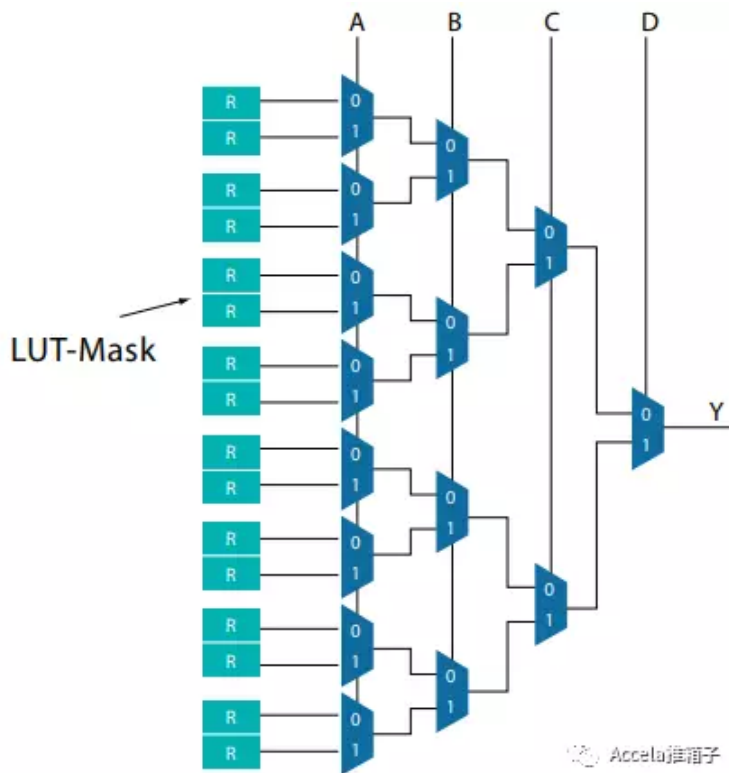
(续前文：有意思的论文FPGA Catapult (P1) )

### FPGA可编程原理

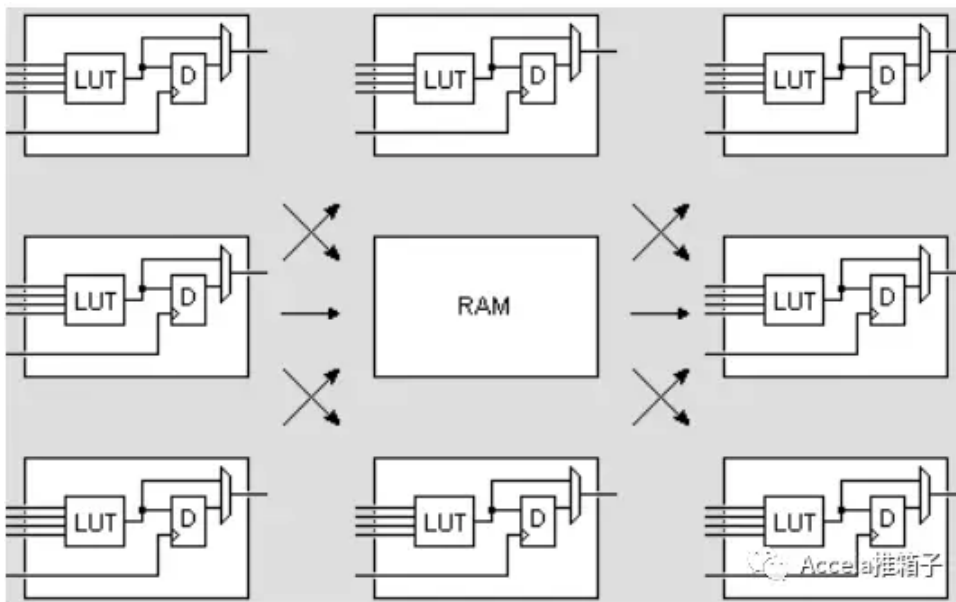
FPGA的基本组成单元是可编程**Configurable Logical Block (CLB)**。简单说，大量CLB排列组成FPGA板。



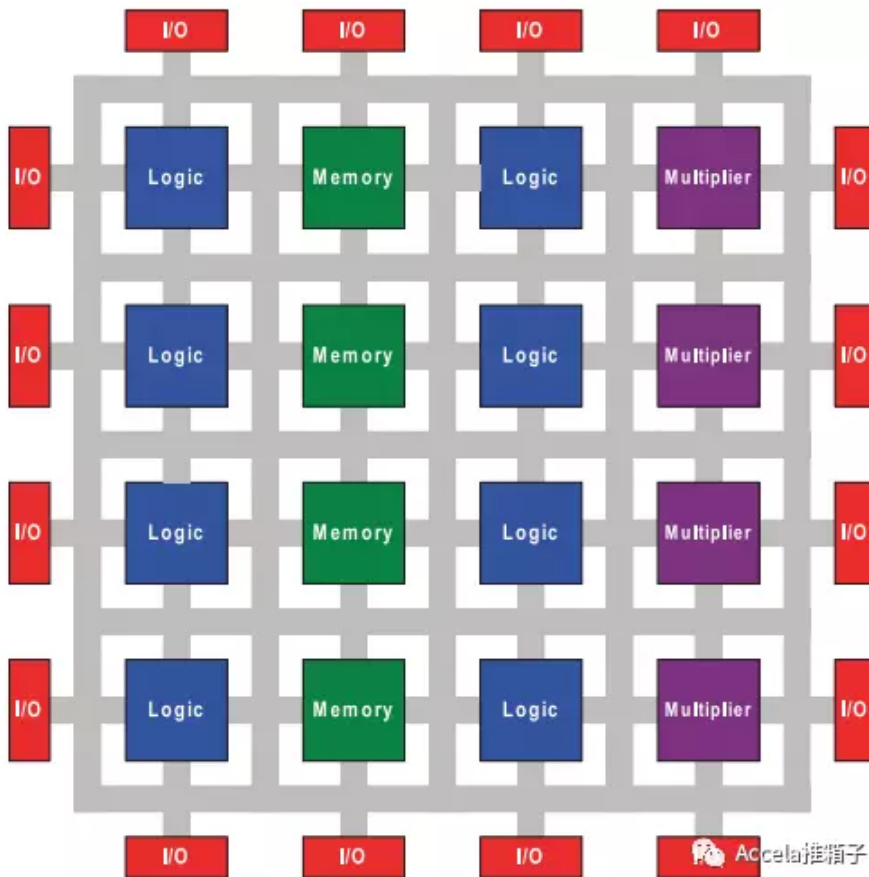
CLB是由门单元组成的组合逻辑电路，可编程为任意的真值表（Look-up-table (LUT) ）。换句话说，可编程为任意逻辑。编程方式如下图，A、B、C、D为输入，R指寄存器。LUT-Mask由寄存器组成，为其设定不同的0/1值，即可编程。图中4-输入LUT，需要 $2^4$ 个LUT-Mask寄存器来编程。



FPGA板上除CLB外，实际还排布了大量内存。CLB编程需要寄存器，可以是SRAM。另一方面，时序逻辑电路自身也需要内存。可以有独立的内存块，内存也可以分散混编入逻辑单元中。

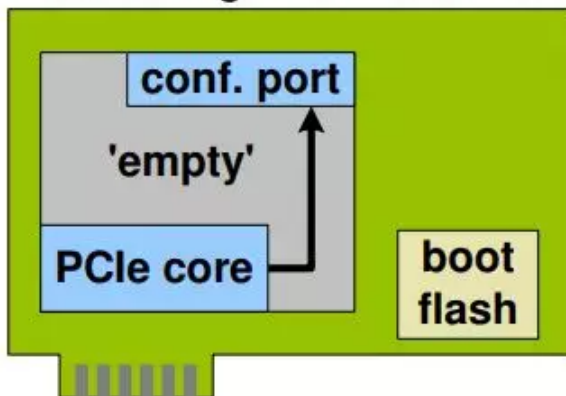


总体上，CLB、内存、其它附属混合排布，周围有IO引脚连出。

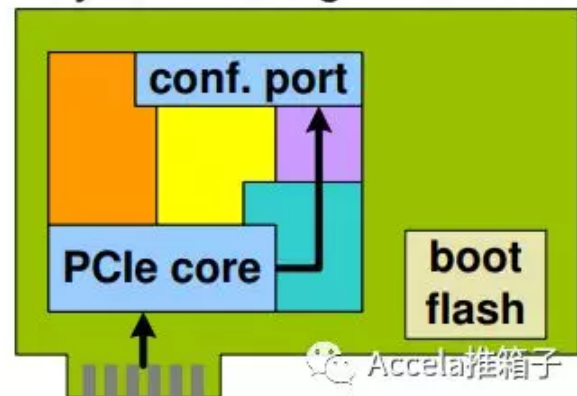


内存并不能持久保存编程配置。通常FPGA配有Boot Flash，以持久化编程配置。开机时读取Boot Flash重新编程。

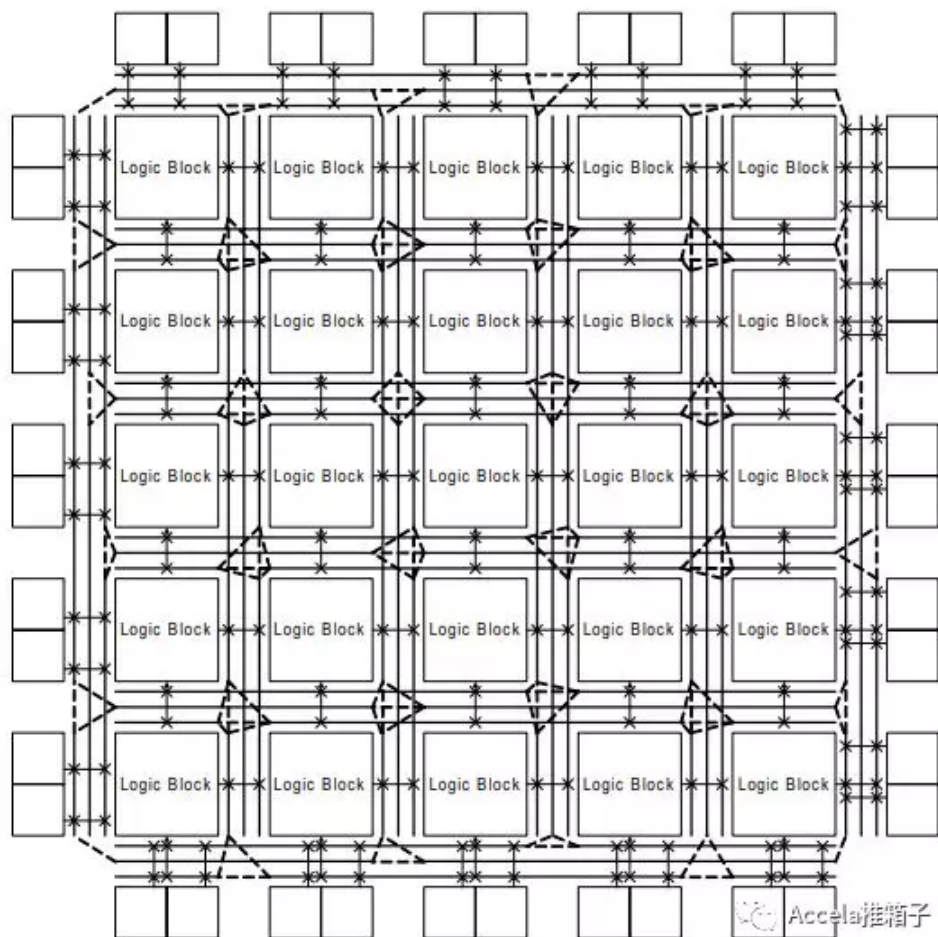
Initial config. from boot flash



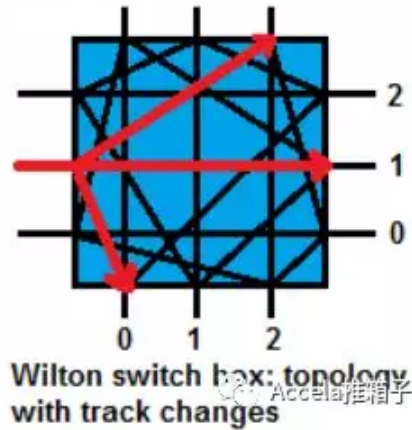
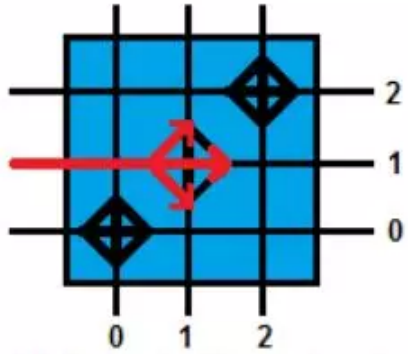
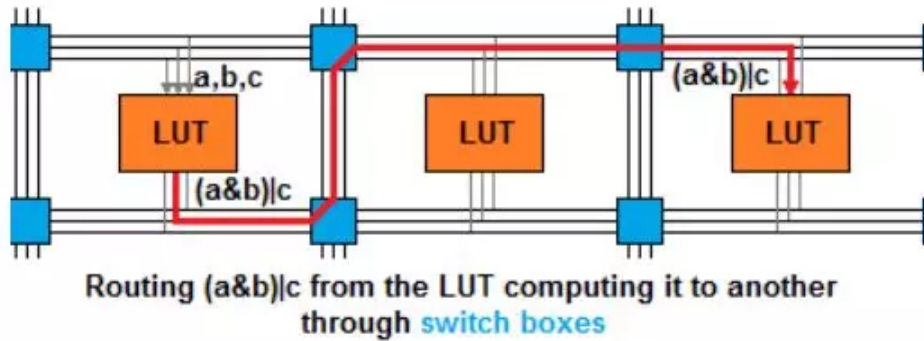
System config. via PCIe



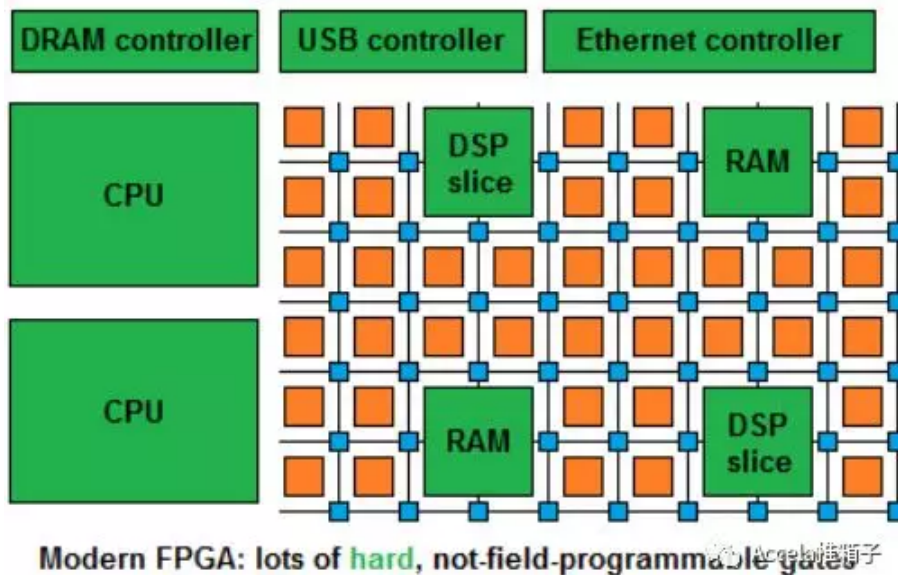
回到第一张图，除橘黄色方块代表的CLB外，还有大量蓝色方块和黑色连接线。它们构成FPGA的 **Routing Architecture**，即可编程的路由结构。Routing Architecture控制CLB间如何连接，通过对它们的编程，大量CLB连接成为复杂的功能逻辑。



连接线交汇点，即第一张图中的蓝色方块，通常称为Switch Box，名称由来类似交换机。它们支持多种连接结构，可编程其连接方式。Routing Architecture的可配置即在于对Switch Box的编程。



商用FPGA更进一步，常常在板上嵌入DSP核甚至ARM核，与CLB混排，达到更强的运算能力，并且与它们互补长短。与可编程的CLB相反，这些DSP核、ARM核不能再编程，它们被称为**Hard block**。



谈到时序逻辑电路，当然少不了时钟。FPGA板配有多个时钟，划分时钟区域**Clock Domain**。Clock Domain可配置不同的时钟频率，以平衡性能和省电需要。例如前述Catapult V2论文中，不同功能模块划分时钟区域和频率。



	<b>ALMs</b>	<b>MHz</b>
Role	55340 (32%)	175
40G MAC/PHY (TOR)	9785 (6%)	313
40G MAC/PHY (NIC)	13122 (8%)	313
Network Bridge / Bypass	4685 (3%)	313
DDR3 Memory Controller	13225 (8%)	200
Elastic Router	3449 (2%)	156
LTL Protocol Engine	11839 (7%)	156
LTL Packet Switch	4815 (3%)	-
PCIe DMA Engine	6817 (4%)	250
Other	8273 (5%)	-
<b>Total Area Used</b>	<b>131350 (76%)</b>	-
Total Area Available	172600	Accelerator 挂箱子

关于FPGA的可编程性，术语是Reconfiguration。现代FPGA通常支持**Partial-reconfiguration**，即部分重编程。Partial-reconfiguration可以在不中断FPGA其它部分运作的情况下，将指定区域重新编程。另一方面，Partial-reconfiguration速度更快，一般在~100ms左右，而全部重新编程可消耗1-2s。后文将会看到，Partial-reconfiguration对FPGA的云虚拟化至关重要。

上文图片的出处、关于FPGA的可编程原理的资料

[Altera FPGA Architecture White Paper]

([https://www.altera.com/en\\_US/pdfs/literature/wp/wp-01003.pdf](https://www.altera.com/en_US/pdfs/literature/wp/wp-01003.pdf))可了解CLB编程的原理。

Altera改进版CLB被其称为ALM。

[How FPGAs work, and why you'll buy one]

(<https://www.embeddedrelated.com/showarticle/195.php>)浅显易懂地介绍了FPGA架构的原理和各个概念。

[FPGA Architecture: Survey and Challenges](<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.153.3122>)深入介绍FPGA架构的原理、各个组成部分、不同实现方法、未来的发展以及挑战。

[FPGAs - How do they work?](<http://www.fpga4fun.com/FPGAinfo2.html>)中也深入介绍了FPGA的硬件组成和原理。这个网站FPGA4Fun有更多关于FPGA的信息。

关于Partial-reconfiguration的资料

[How to take advantage of partial reconfiguration in FPGA designs]

([https://www.eetimes.com/document.asp?doc\\_id=1274489](https://www.eetimes.com/document.asp?doc_id=1274489))

[Partial Reconfiguration on FPGAs]

([http://www.uio.no/studier/emner/matnat/ifi/INF5430/v11/undervisningsmateriale/lecture\\_slides\\_dirk/lecture\\_RC.pdf](http://www.uio.no/studier/emner/matnat/ifi/INF5430/v11/undervisningsmateriale/lecture_slides_dirk/lecture_RC.pdf))

从上述硬件原理，可以推论出FPGA相比GPU、CPU的诸多不同，以及优缺点。前文李博杰的文章中也有很好的总结。除了可编程和低能耗外，

- FPGA是一个可编程的集成电路，而GPU、CPU属于冯诺依曼架构。FPGA没有指令的概念，根据编程逻辑不同，可以构造任何想要的电路。GPU、CPU读取一条条指令，按照指令处理数据；GPU集成大量处理核，以SIMD方式提高吞吐量，而CPU则以复杂的指令流水和预测见长。
- 对于访问内存，FPGA的逻辑单元和任意连接、划分区域，各自控制逻辑不需要仲裁和缓存。而GPU、CPU等由于多核竞争访问内存，需要引入仲裁和缓存机制，增加的延迟。
- FPGA编程消耗CLB的数量；如果有更多的CLB，就可以烧制更多的并行流水逻辑；换句话说，板上面积就是计算能力。GPU、CPU可以通过增加核数提高吞吐量，但由于总体更加复杂，能做到批量处理，但无法做到FPGA的低延迟。
- 复杂的处理逻辑，在FPGA上可以烧制为更大面积的电路，但在一两个时钟周期内运算完毕。对于GPU、CPU，则需要多条指令，多个时钟处理周期，天然延迟更高。
- 对专用电路支持的复杂逻辑，FPGA需要编程实现，而GPU、CPU则有内置支持，可以是ASIC。FPGA这方面通常劣势，例如FPGA不擅长浮点数运算，这成为其机器学习应用的一大障碍。但作为补充，商用FPGA通常内嵌DSP处理器，以弥补之。
- FPGA的编程是完全面向硬件的，通常使用硬件开发语言，如Verilog。面向CPU的编程语言如C，虽然可以使用，但不利于完全发挥其硬件特性和并行。作为弥补，厂商大量提供可复用的功能设计，称为IP。相比GPU、CPU，后者面向广大软件开发者，直接兼容的大量语言和框架，开发易度和推广速度远超FPGA。

(未完待续……后面将讲解FPGA的云虚拟化方案、多种应用场景等。注：本文为个人观点总结，作者工作于微软)

