

FPGA 三国志-第一篇/不可不看的故事

CPLD 的时代

我在 12 年前，偶然接触 PLD，没有想到自己居然就在这个行当里安身下来。可是这个行业也的确是个飞速发展的行业，十多年过去后，从当初的接近十家主要供应商，到今天已经激烈搏杀后，只有差不多如论坛题目一样的，成为了今天三足鼎立的局面。想来想去，决定以这个名字作为论坛的主题。同时也和大家分享我多年来的一些经历和感受。

全局布线，ISP,PLD,宏单元机构，成为 PLD 市场必备的武器。

CPLD 时代，进入我国最早的供应商是 Lattice，那个时候，也不是每个行业都用的了这种产品。首先，软件是需要收费的。这个和今天你可以轻松下载到免费的版本有很大的不同。另外，不同的授权，也决定你能使用不同的产品规模和设计语言。

90 年代中，是电信行业大发展的年代。同时也是专用应用领域大发展的时代。当时 pld 是解决一个逻辑粘连的功能。同时由于 Lattice 很早进入市场，推广很成功。全局布线池的结构，对于布局布线要求不高。一度时间，很多大的通信企业，研究所，都很快成为 Lattice 的用户。但是，有句话说得好。“长江后浪推前浪，前浪死在沙滩上。”太早的成功也孕育着。安则危！

94 年 Altera 已经有了一些用户。但是相对来说。还是很有局限性。另外过去的信息远远不可以与今天同日而语。但是用过 altera 的工程师，已经为他的界面和功能留下了深刻的印象。96 年是 ALTERA 在中国发力追赶的开始。当然，乱世出英雄。当时的代理商是一家香港的公司。他们很快找出这种集成电路最好的销售和支持模式。并且这种从大洋那边继承过来的方式，经过适当的改良，的确收到很好的效果。代理商有专门负责的现场应用工程师。这种方式极大方便了设计者与供应商之间的信息交流。在推广初期是否有技术支持，变得非常重要。这个时期在整个中国市场上涌现出很多非常优秀的现场技术工程师。今天已经有很多人成为这个行业的领军人物。

Lattice 首开 ISP 技术先河，也就是今天常说的在线可编程，给所有设计者带来很大的方便。芯片在电路板上，可以直接编程调试。不用每次拔插芯片，再通过紫外线来擦写要方便得多。这个时候，Lattice 还是明显占有上风的。毕竟先入为主嘛。当时主要的型号集中在 Lattice ispLSI1032, 1016, Altera 的 EPM7128E，不过 Altera 已经做好了准备，因为，Altera 毕竟是 pld 的发明者。而且也是最早采用 Windows 平台的开发工具。在美国市场上占有先机。他不会轻易放弃这样一个市场

当时的应用在电信领域主要是将之前的 74 系列的一些单元进行集成。同时加入一些控制功能。不过这个时期的产品结构都是采用 mc 单元结构。每个 mc 实际上就是相当于 32 到 36 个与非门，以 Altera EPM3032 为例就是认为有 32*30 到 256*30，大致就是 7000 门左右，因此产品命名为 MAX7000，但是当时的设计很多还在大量应用异步设计。因此，电路的结构如果能导致利用率的上升，将是更加有竞争力的表现。MC 的结构就是采用先组合，后时序多个时钟输入结构。Lattice 是用 4 个宏单元一组。altera 采用 8 个一组。而且，altera 在利用率上，稍微占有上风。同时 Altera 当时的 maxplusII 的良好界面。在 97，98 年的两年时间里，已经奠定在中国的基础。更深层次，Altera 已经看到未来市场的需求，前面说到，成功太早有时候也不是好事。在 MAX7000 的铺垫下，Altera 已经有了进攻 Xilinx 的武器，可 Altera 一致宣称那不是 FPGA，换以一个更加中性的名字---CPLD（复杂可编程逻辑器件）。那什么是 Altera 的武器呢。FLEX8000！他的出现是 Altera 奠定今天可以和 xilinx 平分秋色的基础。

在 1996，97 年，成都，西安多家做专业领域的公司和研究单位，也使用了一些 Actel，Actel 的产品和那个时候的 Quicklogic 来说，都是属于 Antifuse 的技术。使用他的最大好处在当时就是有防止辐射，就是说在航空产品中可以用。但是需要你认真的仿真。如果你烧入进去设计，就只能换下一片了。而且为了烧断里面的熔丝，第三方的编程器支持的也不多。但是他们有些军用温度的产品。还是在这个领域有不错的口碑。

Actel 当时的策略结构是，基本上和他们现在的论调也非常一致，就是精细颗粒，所有的 DFF，还是可以靠独立的门来搭建。这个在他早先的 A1020 等产品系列上可以看到。而且芯片上有一部分是组合逻辑区，有一部分是时序逻辑区。另外他们的软件也是多家 EDA 工具的组合。特别是库的一致性不是特别通用。

Quicklogic 的产品是 FPGA 公司中最早嵌入 Synplify 的商家，而且他们的 LE 结构是基于 Mux 的，底层的 layout 也可以清晰看到路由，资源消耗。输入法和库的建立很特别。但是也存在上面的问题。产品好像是 QLxxx 的，我还去应聘过一次这家公司，因为他们最早采用 Synplify。而且，当时我已经感觉 Synplify 和雷昂纳多（拼写忘记了），以及 exampler?? 这个拼写也有问题。我在做 FAE 的时候，总是用这 3 个工具都综合一次，看他们哪个强。后来证明是对的。Sy 是最平均的，而且简单的优点傻瓜。遗憾他也在今年被 Synopsys 收购了。现在 Quicklogic 也专注一些细分市场。

为什么要这样的结构-先组合，后时序多个时钟输入

随着科技的进步，有很多人已经忘记了很多细节，当然，我们也要遵循一个原则。难事做易，大事做细！就像朱熹说的--“问渠哪得清如许，为有源头活水来”，如果你不知道原理，碰到重要的问题，或者是设计的时候，**即使成功，也是不知不觉成功，同样呢，也是不知不觉失败。**

拉回话题，当你设计一个扫描电路的时候，例如，你可以采用一个时钟，输入一个计数器，然后计数器的输出，再驱动一个译码器，这样你的电路就出来了。当然事情完成了一半。功能实现了，但是，有没有更好的办法。后者换句话说。有没有适合 PLD 的方法。实际上用另外一个角度看，这个设计是典型的，先时序模块，才是组合电路。这个实际上，不是很和 PLD 本身的结构相符。还有一个问题时，随着电路速度的提高，每个译码输出之间的抖动也成为问题。就是输出的不一样的。从资源的角度来说，一个计数器，要 3 各单元。一个译码器，由于有 8 个输出，还是需要 8 各单元。但是如果用一个移位寄存器的方式，可以只需要 8 个单元。或者用状态机的方式实现，也是消耗同样的资源。可是，输出都是由于同一个时钟锁定，每个输出的延迟也比较一致。另外，当时对总线数据的译码是经常有的事情，因此，每个单元的扇入数量都是很高的。都有 30 个以上。因此，你设计的时候，**要有两个思想：用同步设计模式，用先组合后时序的模式。**后来的 Xilinx 的 95 系列更是将扇入系数扩大到 90 个。也是这个原因。当时 PLD 厂商比较的也是这些指标。

Xilinx, FPGA 的发明者

Xilinx 实际上成立的日期，比 altera 还晚了一年。但是他走的道路，从开始就注定了有今天的成就。在 1996-1998,pld 的高端市场，也就是 FPGA 市场，都是 Xilinx 把持，主要是 xc3000/4000，当然 Xilinx 开了一个新的结构，不过这种结构还是能够找到一些共同点，也是先组合，后时序。不过很重要的闪亮点是：**细颗粒查找表结构，丰富的寄存器资源，以及分段式路由布线结构，电路上电加载。**当然如果这个也算是的话，那就是，他的**内部甚至直接集成了三态门。**这个特点是那个时代 Xilinx 拥趸攻击 Altera,Lattice 的重要工具。甚至有人说，没有这个不能实现某些电路设计的障碍。当然了，Altera,Lattice 的 pld 也是具有三态门，不过是在输入输出管脚上。的确是不如 Xilinx 的来的强大和直接。有的人问了，这种结构有何好处。

1. 电路的规模得到空前的提高。
2. 适合做加加減減的，counter,comparter 密集的设计。
3. 无限制的更新电路。

在那个时代，衡量一个 PLD 很重要的指标，就是比谁的规模更大。从这个角度来看，Xilinx 无疑走到了最前面。当然，也有人说，这样的设计不可靠。这样的设计指标难以预测。还有地说这样的设计无法保密。但是市场的成功，应用需求的推动，证明这些都是杂音。

Device	Max Logic Gates	Typical Gate Range	CLBs	Array	User I/Os Max	Flip-Flops	Horizontal Longlines	Configuration Data Bits
XC3020A, 3020L, 3120A	1,500	1,000 - 1,500	64	8 x 8	64	256	16	14,776
XC3030A, 3030L, 3130A	2,000	1,500 - 2,000	100	10 x 10	80	360	20	22,176
XC3040A, 3040L, 3140A, 3142L	3,000	2,000 - 3,000	144	12 x 12	96	480	24	30,784
XC3060A, 3060L, 3160A	4,500	3,500 - 4,500	224	16 x 14	120	688	32	46,064
XC3090A, 3090L, 3190A, 3190L	6,000	5,000 - 6,000	320	16 x 20	144	928	40	64,160
XC3195A	7,500	6,500 - 7,500	484	22 x 22	176	1,320	44	94,984

November 9, 1998 (Version 3.1)

7-3

我们可以从当时 xc3000 的选型指南上看到。最大规模的产品，也就是今天 xilinx 最小的 xc3s50a 规模相当的东西。可是这个和当时 PLD 流行的规模，实在是一个飞跃。

Xilinx FPGA 的架构。

Xilinx 的成功，激励了 Altera 的新产品的推出。同时也有了革命性的突破。世界就是这么奇怪。中国古语。塞翁失马，焉知非福。实际上，如下所言，这款产品因该 8282,具有 282 个逻辑单元的。但是在中国基本上只是在 1995 年有个别公司使用。

1992	FLEX® 8000 FPGA	Altera 的第一款现场可编程门阵列(FPGA)
------	-----------------	---------------------------

我本人经历我们开始在 1997 年开始大规模推广 Flex10K, 同年 5 月，位于西安的邮电部 4 所，是第一个采用 EPF10K50 的用户，当时电子杂志的广告，也有表明,Altera 最先推出最大的 FPGA。

1996	FLEX 10K® FPGA	带有集成锁相环(PLL)的 FPGA
1995	FLEX 10K® FPGA	带有嵌入式模块 RAM 的 FPGA

实际上，这个时间就是 Lattice 噩梦的开始，由于设计规模的迅速攀升，规模已经是决定一个产品的关

键因素。但是 Lattice 一直没有代表作。市场份额已经开始逐步被 Xilinx, Altera 蚕食。Altera 的结构到底是什么革新呢？

Altera FPGA 粗颗粒结构，嵌入式存储器，长联线结构。

Flex10K 的推出，已经表明，altera 的思维已经非常清晰，大规模的 fpga 将会是将来的王道！而且，高速的嵌入式块 RAM 是一个开创新的思维。当时，Xilinx 凭借分布式 RAM 以及内置三态的结构，的确大有横扫千军的架势。但是 Flex10K 的出现，有效提升了 Altera 的 FPGA 产品的竞争力。在 RAM 需求应用不断上升的市场中。Altera 迎来了市场的春天。

1. 采用多选一，或者一对多的 MUX 结构，Altera 推出应用指南。直到用户有效利用这种方式，实现了三态门的功能。
2. ESB 的 memory 结构，是构建 FIFO, DPRAM，大型查找表，都非常方便。
3. 布线资源相对来说，以长联线资源为主。编译速度快。

Xilinx 的分布式 RAM 尽管很灵活，但是随着 RAM 容量的增加，访问的速度也是递减。同时也要消耗大量的逻辑资源。这个时候，应该说 Xilinx 和 Altera 的产品已经是势均力敌。因为中国是新兴的市场。这一点表现得很明显。

规模制胜的产物

Feature	EPF10K10 EPF10K10A	EPF10K20	EPF10K30 EPF10K30A	EPF10K40	EPF10K50 EPF10K50V
Typical gates (logic and RAM) (1)	10,000	20,000	30,000	40,000	50,000
Maximum system gates	31,000	63,000	69,000	93,000	116,000
Logic elements (LEs)	576	1,152	1,728	2,304	2,880
Logic array blocks (LABs)	72	144	216	288	360
Embedded array blocks (EABs)	3	6	6	8	10
Total RAM bits	6,144	12,288	12,288	16,384	20,480
Maximum user I/O pins	150	189	246	189	310

世界是运动的，粗颗粒，也是好结构！

这里又回到了我们的主题，产品的内在结构。先说细颗粒结构。当时，Xilinx 的 3000 是以 CLB 为基本机构。

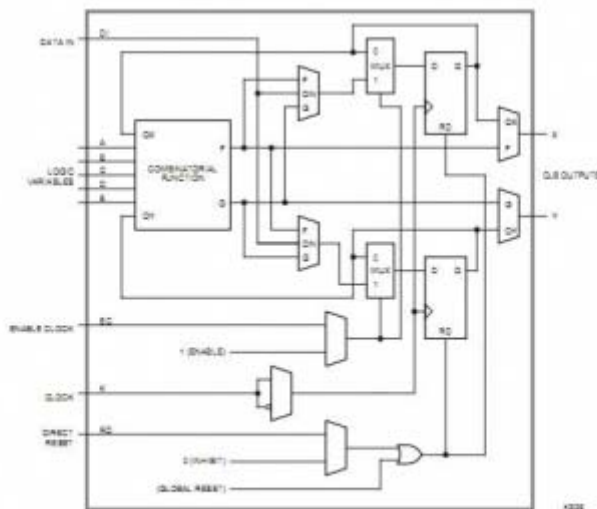
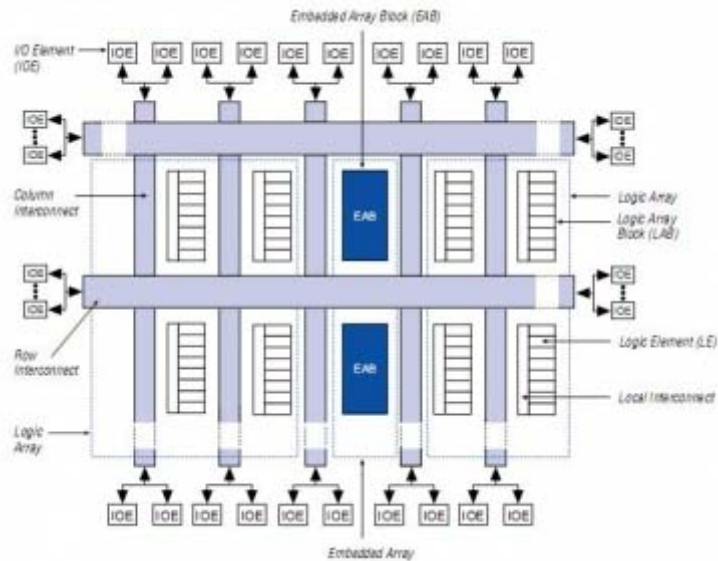


Figure 5: Configurable Logic Block. Each CLB includes a combinatorial logic section, two flip-flops and a program memory controlled multiplexer selection of function. It has the following:
 - Five logic variable inputs A, B, C, D, and E

实际上，这个输入的个数是很有讲究的。当时，也有一种激烈的争论，就是 5 输入的好，还是 4 输入的好。在实际应用中，应该说，最小的逻辑单元是这样的规模是合理的。但是要注意一点。Xilinx 和 Lattice 后面推出的 fpga 一样，两个寄存器是具有同一个时钟输入的。

Figure 1. FLEX 10K Device Block Diagram



粗颗粒结构，Altera 还是依托自己以前在 PLD 的经验积累，他的 fpga 有很多他的 pld 产品的血统。世界是基于以权制利的。因为这个时候如果已经有了 Wintel 的名字一样。数字电路以及 CPU 的发展，都和 8 以及 8 的倍数有关系。因此。这种大颗粒的结构确保：

1. 8 个逻辑单元 LE 为一组 LAB。
2. 每个 LAB 内部有独立的布线资源。确保可以实现 8bit adder, shift reg, 8bit counter。
3. ESB, 实现了存储器的资源和速度同时提高的可能。
4. 布局布线算法比细颗粒机构的算法简单。
5. 采用了非对称的结构，实现速度需求和控制需求的折衷。

Cyclone, Stratix, Vertex

Cyclone 的出现和 FPGA 的市场定位和产品定位

好的技术是需要，但是有效的市场定位和销售策略也要跟上。在 2002 年左右，为了更好地服务各种不同市场的需要。而且当时的 DSP 市场仍然不太明朗，有的观望，也有大力投入。而且也有两种不同的策略。

一种是同一个器件平台，但是不同的应用区隔，例如 Vertex 和 Spartan, 有一段时间，他们可能是同一个产品平台，只是 Spartan 的东西就是 Vertex 屏蔽了一部分高级的性能，对于不使用的地方，也不需要检测，同时依靠同一个平台来维持成本优势。

另外一种，就是 Altera 的策略，完全依赖客人市场的需要，简化不需要的特性，推出简化的结构，面向主要的逻辑集成和 LVDS 信号链集成。推出 Cyclone，市场反应非常强力，是 Altera 有史以来，市场响应最快的产品。

FPGA 三国志-第二篇/结构篇

Altera FPGA 的速度没有 Xilinx 的速度快，错！当然这种非对称的结构，你必须有一定的了解，才可以更好的利用。也就是要遵循：

- ◆ 大的数据吞吐通道应该采用横向放置规划！
- ◆ 控制通路采用纵向放置规划！

Altera 至此以后，一直沿袭这种结构规划。因此，如果你想有效利用好 Altera 的产品，就应该遵守这个规则。但是要说明的，真正能体现性能的东西，也许就是你意识不到的一种小东西，就是简单的就是最好的。Altera 正是凭借这个简单而高效的布局结构实现了今天的王道！昨天说到这里今天继续开始。又有很多时候没有提 Lattice，在 1998 年的时候，Lattice 和 Altera 同时都有成为 PLD 霸主地位的意图。什么可以证明呢，那就是谁最先推出可以 ISP 的宏单元超过 1000 个的 PLD。当然在这个游戏过程中，Altera 有一些变化，他有效的将他的 Flex8000 的布局结构和他的 MAX7000 进行结合。从而实现了在 PLD 规模扩大的同时可以实现：

1. 规模的迅速扩大,可以比肩 Xilinx 的 FPGA
2. 局部的快速布线,和 ISP,使其在获得规模优势的同时,保持布线延迟的稳定
3. 沿用过去 MAX7000 的适配结构和 FPGA 的路由,实现两者有效的统一。

Altera 在推出他的最大的 9560,具有 560 个宏单元的 PLD,登上了无可争议的 PLD 冠军奖台。

Feature	EPM9320 EPM9320A	EPM9400	EPM9480	EPM9560 EPM9560A
Usable gates	6,000	8,000	10,000	12,000
Flipflops	484	580	676	772
Macrocells	320	400	480	560
Logic array blocks (LABs)	20	25	30	35
Maximum user I/O pins	168	159	175	216
t_{PD1} (ns)	10	15	10	10
t_{FSU} (ns)	3.0	5	3.0	3.0
t_{FCO} (ns)	4.5	7	4.8	4.8
f_{CNT} (MHz)	144	118	144	144

Altera Corporation

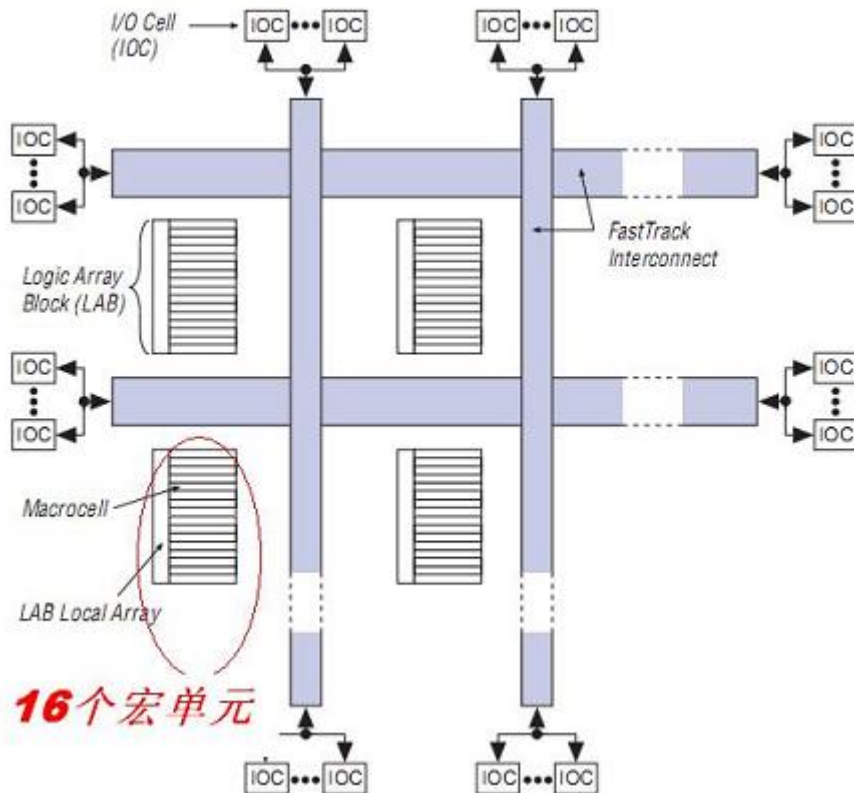
A-DS-M9000-06.01

1

当然有的人要说,Lattice 有推出 1000 个左右,怎么不提呢. 正是因为这点,导致 Lattice 步入歧途. 实际上,FPGA 世界的游戏规则已经改变了.

MAX9000 的成功得益于以下的细微结构.这个时候 Altera 又一次将自己的颗粒度进行了扩容. 有 16 个宏聚集在一起,在实现更多位的加法,控制,超前进位,大的多选一的应用中,可以将这些模块一次性放入一个 LAB,同时在 LAB 内完成路由. 现在已经露出一一种迹象. 大的规模要有,但是速度的需求已经开始了. 所以在 FPGA,CPLD 的应用中,又有了新的裁判规则,你不仅要够大,还有够快.

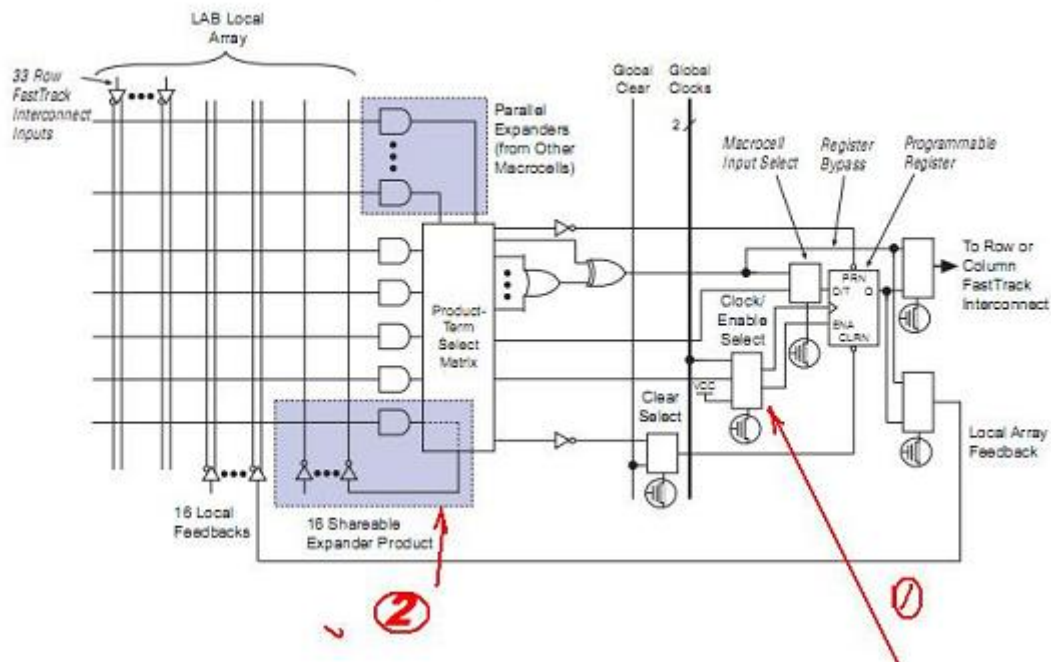
MAX 9000 Device Block Diagram



这个时候的异步设计还是非常的多,而且板上芯片间信号的互联也多起来了,能够有效缩短 T_{su} 已经成为一个重要的话题. 实际上,就是在 IOC 上要有 DFF,来进行快速锁存,同时也为所有进入 CPLD 的总线信号

进行第一次整理. 怎么实现很多的异步设计,看了下面,你就明白了. 第1点,就可以用所有信号的函数输出作为 clk,第2点,有效的将没有用完的资源很好借用给其他的宏,来用对称的结构实现非对称的应用!用简单的结构,应变不断的变化.

Figure 3. MAX 9000 Macrocell & Local Array



纵观当时其他的 PLD,在结构上就落后很多了. 你想,让你和姚明来争篮板球,如果你没有人家的身材,赢他恐怕也是嘴上的功夫了.

这个时代的强者就是---谁有最多的逻辑资源,或者 memory,谁就是老大.

上面说到 Lattice 已经在极力扩大自己的身材,但是他不是靠结构上的改变,而是 Lattice 收购了 Vantis,也就是 AMD 的一个做 PLD 的小部门. 当然在当时,AMD 的 Mach 就这样并入了 Lattice 的家族. 新的问题就出现了. 好比我们现在有人用什么大灵通,小灵通,GSM,CDMA,是有百花齐放的感觉,但是 Lattice 就像变成了解放前的蒋介石,没有办法很好的用一套工具来统一使用不同的器件. 而且本身 Lattice 自己当时的工具也是 3 个独立的工具拼凑在一起的. 那个年代,用过 Tango,后者 Orcad 的人都知道,他们的图形输入是第一名的,但是和 MaxplusII 比起来,自动识别对象链接,以及和 Office 95 类似的快捷键,用过 MaxplusII 的人,让他们转用 Viewlogic 等 workoffice 等,简直简直就是抹他们的脖子. 还有当时 Lattice 的销售团队,总是宣扬他们是最好的 PLD,有些人竟然有 "我认为 64KROM,就可以应付未来所有的软件需求" 这样的论调,认为 PLD 必将击败 FPGA,事实上,市场的残酷,告诉他们那是个很冷的冬天. 于是他们又一次在 2000 年左右,如同水淹七军一样的结局,又急忙掉转船头,收购了 ORCA,可惜了 ORCA 是出自 Lucent 的一条好汉,由于没有良好的软件支撑,使得每个工程师必须像哪吒一样. 对了,怎么会像哪吒? 因为你必须有三头六臂,如果你公司有些产品需要从 32 个宏单元到 2000 个 LE 的 FPGA 的应用,你就必须学习 3 种工具来适应它. 你想想,你是不是一定要像哪吒呢!

再次谈结构以及方法学!

这里开始讲些看来与我们主题有点不搭界的东西.

- 1--两个人相遇, 只握 1 次手
- 2--三个人相遇, 每人都握一次, 握 3 次手
- 3--4 个呢?就是 2 的结果加 3 次, 6 次手
- 4--5 个呢? 10 次手.
- 5--16 个呢, 天, 不少于...

PLD 就是这样的产物, 当逻辑.....你会发现, 路由的面积都超过了有效的逻辑面积. 而且, 越大越糟糕. 可以下载这个简单的 PPT 来发现一些小问题!

什么样的数目比较好, 对于 PLD 的宏单元数目

实践是检验真理的唯一标准. 有人说了, 256 个是最好的结构, 为什么呢, 不然怎么那么多厂商都是在这

个范围呢。实际上，他是由经济规律决定的，就是当时用 256 个的宏单元结构的芯片的面积，和他们卖出的价钱，比较符合当时这些上市公司的利润要求，所以.....就以这个最流行了。

但是 Altera 进行了很好的变通。中国有句话叫“玄之又玄，妙之又妙”，什么是玄？玄就是变通的意思，也是变得意思。实际上万事万物都是相通的。那 Altera 就像我们的学校一样，每个年级分不同楼层，每个楼层分不同班级，每个班级上不同的课。但是用行列块的方式，达到既有规模，也有位置相关性。而且路由的面积也不会大到赚不到钱。实际上你自己观察，近 10 年，Altera 的 FPGA 的主要框架是没有变化的！

Xilinx 的 FPGA 结构，实际上，有一个 5200 系列，很向 Altera 的 Flex6000，但是没有多久这个东西就不见了。总之，Xilinx 的结构属于称为“孤岛式”结构，就是 CLB 在中间，路由围着这个孤岛。在一定的密度的时候，这种结构也还是不错的，当然有一个很重要的结构就是，他是全对称的。就是 Xilinx 的芯片的逻辑上资源的密集度是上小左右对称的。这个有好处，但是也有坏处。Altera 的呢，是横向资源丰富，纵向资源相对较少，但是，在局部的横向上，又可以级联 LE, DSP, Carry chain 等等。好了，给大家举个简单的例子。

孤岛式的结构

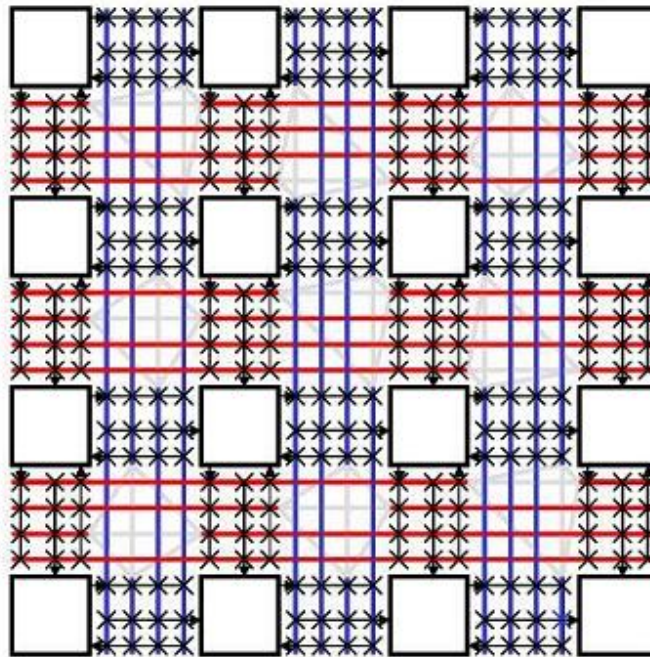
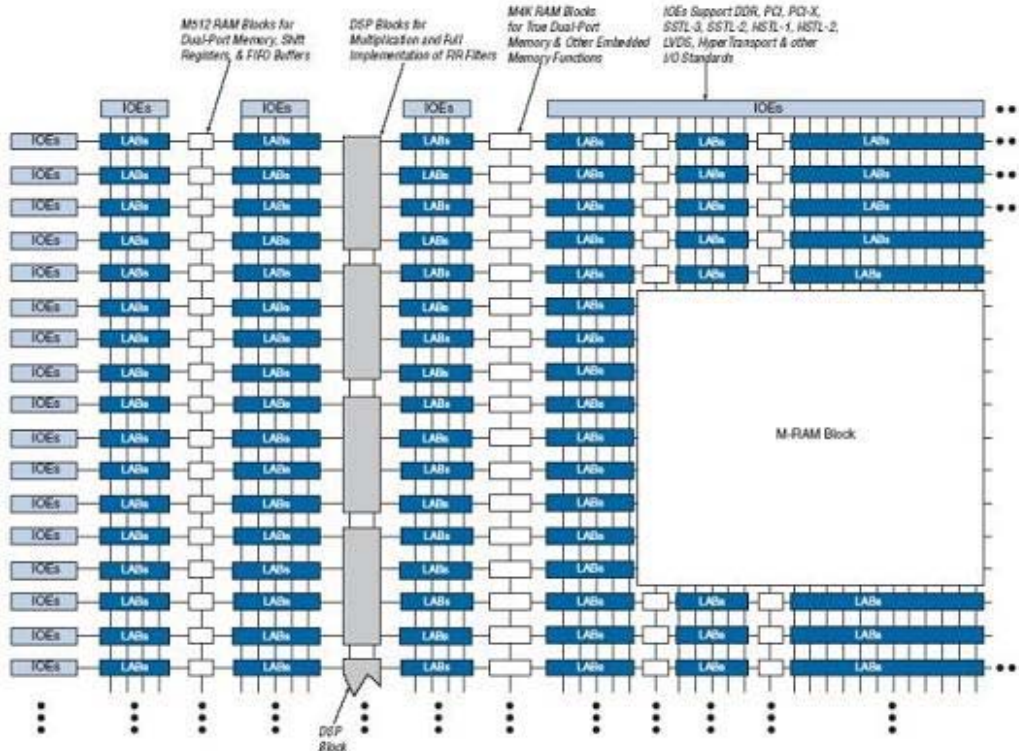


Figure 2-1: An illustration of an island-style FPGA. The white boxes represent logic blocks. The horizontal (red) and vertical (blue) intersecting lines represent routing wires. The logic blocks connect to surrounding wires using programmable connection-points (shown as crosses), and individual wires connect to each other by means of programmable routing switches (shown as gray lines).

Altera 的类似的结构，但是颗粒度大，从这张图，应该可以看出 altera 在横向资源是很丰富的。就是同一行的资源远多于同一列的。输入输出就更是了。当然，这个和管脚的封装脚的出位不是绝对一一对应的。



Stratix 的出现

在 2001 年, Altera 推出了他们最伟大的产品, Stratix. 当时 FPGA 的竞争规则又发生了改变

1. Altera 用 TRAM 的形式和 Xilinx 的分布式 RAM 和 blockRAM 竞争
2. Altera 的 PLL 性能超越对手
3. 布通率, 利用率, 表现突出.

但是上面这三点, 都不是决定性的. 这个时候, 数据通信对背板走线和背板总线要求已经很高了. 实际上 FPGA 也摇身变为系统级芯片了.

你不仅要有大的逻辑规模, 合理的 memory 尺寸, 相对丰富的时钟资源, 还有就是要有高速的 Serdes, 缺少一项, 你都会在系统级的应用中只能是亚军!

刚才说了系统级的应用, 已经成了 FPGA 最残酷的竞争市场. 那么 PLD 呢, 怎么样了, 实际上自然总是物竞天择! PLD 已经变为这样的几种应用了

1. 输入输出的扩展!
2. 简单总线或者接口的协议转换
3. 对系统级的模块进行配置, 或者控制.
4. 上电初期的一些管理

就像一个国家的海军一样, PLD 已经成为一些简单的驱逐舰, 驰骋大洋的, 可以跨海作战的, 绝对不是这样的产品可以涵盖的. 所以, 你今天喝可乐的时候, 不会有太多的选择. 偶尔的一些牌子如同过眼烟云, 很快就弹出你的视线了.

是的, 十年前我的很多朋友, 有在 Quicklogic, 有在 Cypress, 现在还有一些在不断出产品的公司, 当然, 只能是剑走偏锋. 做些细分市场还是可以继续的. 但是三国鼎立的形式已经是不可撼动的事实了.

北京和深圳的差别?

你到过我们的首都北京吗? 到过我们的特区深圳吗? 这样打比喻是因为我在深圳和北京都旅游或者工作过. 实际上我也只是想借用这两个城市的布局来做个比喻.

北京是比较对称的城市, 有东直门, 西直门. 有东单, 西单等等. 总之, 他的布局就像 Xilinx, 无数个胡同就是像 Xilinx 围绕在 CLB 的路线. 这些资源在 Xilinx 的数据手册中有:

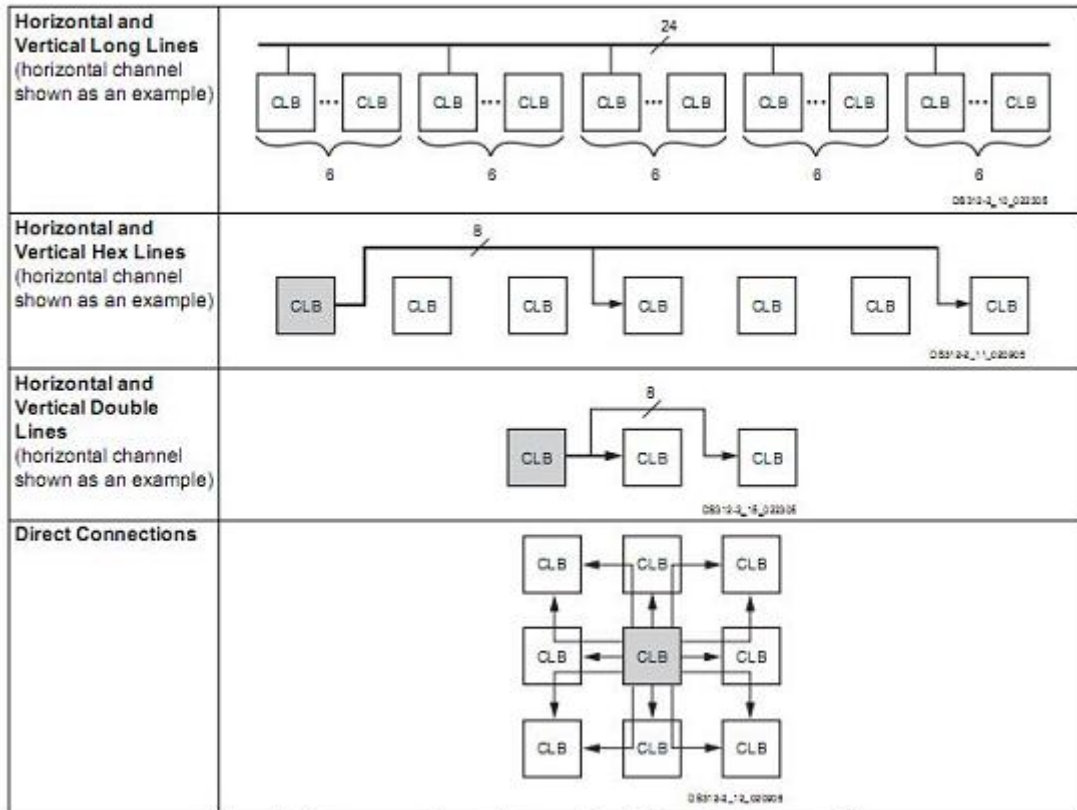


Figure 51: Interconnect Types between Two Adjacent Interconnect Tiles

但是这些胡同间的联线并不是十分充足。特别是到了规模很大的时候。Altera 的呢?在深圳的人,如果你不认路,很简单,只要你走到“深南大道,滨河大道,北环大道”上的任意一条,你就可以再从这些大道到你要去的地方。但是前提是,这三条大道的宽度要够。提示一点,这三条大道也是东西走向比较平行的。而且整个深圳也是一个东西走向的城市,地下再有一条地铁,在同样资源的情况下,布通率,和平均车速是非常好的。而北京的地铁,是环形的。资源上不如深圳的利用率高,这里无意于评价城市规划和道路设计。只是比喻。

Altera 的 FPGA 就是如同深圳的道路,他可以让你从列上很自如的转到宽阔的行上,然后再到达你的目的地。所以,可以告诉你一个经验。

1. Altera FPGA, 布通率基本在 95%左右,没有太大问题,Stratix 最大的产品,有人有 99%的布通率
2. Lattice 的 FPGA, 规模超过 20KLEs, 布通效率... 有愿意透漏的吗?
3. Xilinx, 规模越大,有所恶化。

另外,这个地方也有颗粒度大带来的好处。所以,有些原来看似不精细的地方,却在规则改变后,就变得反而犀利无比。

运筹学也是很重要的!

经常有这样的情况,大家选择 FPGA 的时候,就开始翻看每种 FPGA 的选型手册,然后对资源表。有的甚至直接说:我这个是 10 万门的。你的那个是 6 万门的,实际上,这些都不是很正确的评估。如同有人说,我家的房子 4 室两厅,你家的房子 3 室 1 厅双卫生间。到底哪个更大呢?到底哪个更节能呢。厨房热水器之间离得很远又是问题。

所以说,学会正确的评价资源是很重要的。这个时候还要参照他的结构!!板式的,还是砖混的!!!

可能你还听说过一个使用面积的问题。实际上,有的房子看起来大,走道,不规则的布局,导致很多的地方都不能用!下面看个例子:

■ Reach more logic faster

	Number of Reachable LEs (Logic)		Ratio
Hops	Stratix III	Virtex-5	Stratix III to Virtex-5
1	850	132	6.4
2	2400	1056	2.3
3	4000	1980	2.0
Total	7250	3168	2.3

左边的 Hops 代表跨一步的意思, 就是路由转换一次的意思. 这个表什么意思呢. 就是说在 S3 的路由过程中, 每个路由在第一次, 就可以覆盖 850LEs 中的一个, V5 是 132. 如果你的逻辑, 经过 4 跨, 同样的路径覆盖的区域 A 的是 X 的 2 倍. 代表什么呢?

1. 编译速度更快, 因为概率提高, 路由成功率提高
2. 布通率更高
3. 速度更快! 为什么, 90nm 以后, LE 内部延迟已经不如路由延迟的时间长了. 所以经过的路由多, 就会严重降低系统速度.

当然还有一些好事者, 例如 (<http://www.opencores.org>) 有很多公开的 opencore, 大家可以将它们同时尝试放在 A, X, L, 看谁放的多, 放的快, 系统延迟更小. 这方便数据就不多说了, 可能都成为一门学科都不一定, 因为不同的比对都有 benchmark 的不同标准. 最终我们应该很清楚的看到.

结构真的是很重要, 我们能干, 也要看是否站在巨人...

不过呢, 很多时候, 我们的朋友基本上都说: 结构和我无关, 我要学好 VHDL, 我要... codingstyle. 这个设计属于系统工程!

FPGA 的真正命门和 Know How 实际上, 能做 FPGA 的公司太多了, 但是能将我们的设计通过算法成功放到这个芯片上, 而且算出正确的时间和你的仿真要求的. 就不多了, 说的难听点. 有些领域甚至是没有亚军的竞争! 选择小规模 PLD, 那些, 不是太重要的问题. 本身就不够养活一票人的产品.

Cyclone III 与 Spartan3 的对决—苹果对苹果?

目前, 很多人都基本上对于新的中档设计都会集中在这个系列的竞争中, 换言之, 在中国目前以成本为导向的第一要求下, 实际上中小公司的产品选用 80% 是集中在这里.

经常有人说, 为什么你们的 LEs 数目相同, 价格很不同. 做些解释, 一家之言.

1. 在今天, 尽管是可以编程的器件, 还是有不灵活的地方. 例如, 你的应用决定你对什么资源敏感
2. 不同市场也有不同的关注. 没有哪个东西绝对适合, 只能是系列之间互相交叉来完成.

有一点要说, 那些所谓我的是多少系统门的比较方式, 是典型的大忽悠模式

比较产品有很多 benchmark. 这里列举一些. 不全的地方, 可以大家补充.

1. 工艺

Cyclone III, 65nm

Spartan3, 90nm

眼前看, 90nm 是主流工艺, 但是未来降价空间在 2009 年中达到轨点, 另外, 65nm 的功耗不用说, 35% 的优势轻轻松松.

2. 设计学

CycloneIII ---LP 工艺, 有很多人理解这个, 同样的设计采用 LP 和不采用就很大不同.

Spartan3---没有采用

3. 规模

Spartan3, 3e, 3a, 3an, 覆盖区域不同, 从 1K 到 40KLE 左右

CycloneIII: 典型长中焦距镜头, 5K 到 120K,

40K 以上, 基本上 Spartan3 没有产品, 可以用 V5, V5 定义为 Highend, S3 定义为 Lowercost 无线, DSLAM, 医疗, 平均规模在 25KLE 到 80KLEs 为最多, CycloneIII 解决了有无问题

4. Memory

这个是 CycloneIII 的幸运之处, 当时可能设计上没有这么大吧.

9K 块, 总容量绝对平均高出 S 为 30%, 块数也. Sp3dsp 例外, 但是他只有两个孩子. 定焦镜头. 要符合你的品味.

memory 多影响到: CPU 速度的提高, DSP 应用, DUC, DDC, FIR 等, 级数上可以做更多. 速度高, 还可以提高复用.

5. 乘法器, 一个 18x18 的乘法器, 相当于 350-450 个 LEs, 当然流水的话要另外算.
6. PLL: Altera 是模拟的, X 是数字的. 恢复性和收敛速度那毫无疑问, 地球人都知道模拟的好. A 的时钟树更是多
7. 布局: 从左到有, 基本符合多时钟域交叉,
如: LVDS 入, 经过第一级 FIFO, 进行时钟域交叉, 或者数据交叉, 第一次处理, 然后经过中间逻辑加工, 参数重加载等, 然后进入乘法器, 可以级联, 然后有通过通用逻辑池进行加工, 再次通过 FIFO 或者 RAM 来对接下一级.
其他行可以独立构成 NiosII 等, 布局收敛性一流. 我尽量找个照片来显示.
所以不是简单说我的苹果和你的苹果一样. S3 只有销价处理才是真正的出路.

如果你知道了结构, 你会发现什么呢?

实际上, Altera 一直在横向布线资源上浓墨重彩, 因此, 在 Cyclone 一代系列的时候, 如果你的设计模块放在一个比较长的区域就更加容易跑出好性能.

如: 5 行, 5 列的一个资源放置, 不如在一个 4 行 7 列的区域中更好发挥性能.

到了 CycloneII, 可能就是接近 1:1.2 的样子.

Statix, 基本上接近正方形.

其他的你可能需要实验一下. 毕竟这个是动手的科学. 而且, 每家 FPGA 厂商, 在关键布线资源方面, 都是秘而不宣. 这个也是为什么 Synplicity 要另嫁豪门的原因! 因为自己没有办法得到这些资源. 而综合技术已经被 FPGA 厂商步步紧逼了!

Cyclone III 的巧合.

Cyclone III 的诞生, 可以说也是有划时代意义的. 但是赢得偶然, 其中已有些必然. 下面来说说这个东西.

前面有人说过 Xilinx 的 V5 不错, 但是如果说 V5 和 StatixIII 带有 Serdes 的产品同时间面世的话. 作为设计者, 可能问题就来了. **太多选择就是难以选择**

StratixIII 速度快, 布线好, 但是没有 Serdes

V5 速度布线都不错, 出来的早, 快人一步也是卖点.

但是呢? 这两个产品都很贵.

由于 Xilinx 很重视高端用户, 因此他们也认为 Spartan3 可以解决目前很多需要, 这个也对, 因此他们计划在 45nm 左右推出 Spartan 的升级产品.

Cyclone III, 正好赶上高端 DSP 处理市场的繁荣, 以前 Xilinx 的 website 上面也有这方面的迹象. 号称 Xilinx 也是一个 DSP 的公司. 这个也要得益于其他生态链软件系统的发展, Matlab 的 simulink,

另外, 目前的有线系统中对 Memory 的需求也非常高, 作为有效的缓冲也需要更多的空间.

举例来说, 一个 Video 的应用. 需要一些滤波或者其他的应用, 那需要的 FIR 的 Tap 数目实际上是可大可小的. 但是以前他们认为逻辑资源的比例太多于 DSP 的建立资源. 导致很多设计用 30 万个 LE 的资源的 FPGA, 实际上, Logic 资源利用在 30% 左右, 而 Memory 仍旧显得很局促. 还有一个就是用 DDR2 的设计也有增多的趋势,

还有就是中型设计的比重已经上升很快, 就是在 3 年前, 大家很多集中在 6000 个 LE 左右的资源. 而紧接着的趋势是 200 个 500 个左右的资源和 10000 个 LE 资源的迅速两级化. 另外一个增长就是 50000 个 LE 左右的区间, 而这部分长期以来都是 Stratix 和 v4, v5 的传统空间. 可是现在由于 memory, multiplier 的增加, 导致系统性能也可以用 CycloneIII 实现资源换取速度和效率的方案. 因此 CycloneIII 的资源也能利用低价格来和高端 FPGA 分一杯羹.

但是 CycloneIII 尽管有价格优势, 逻辑和 memory 的优势, 可是在 Serdes 的应用上, 还是一片空白. 这也是大家觉得他的缺憾所在.

总之, CycloneIII 和 Spartan3 DSPA 系列的推广, 已经有一个暗示的信号, 就是

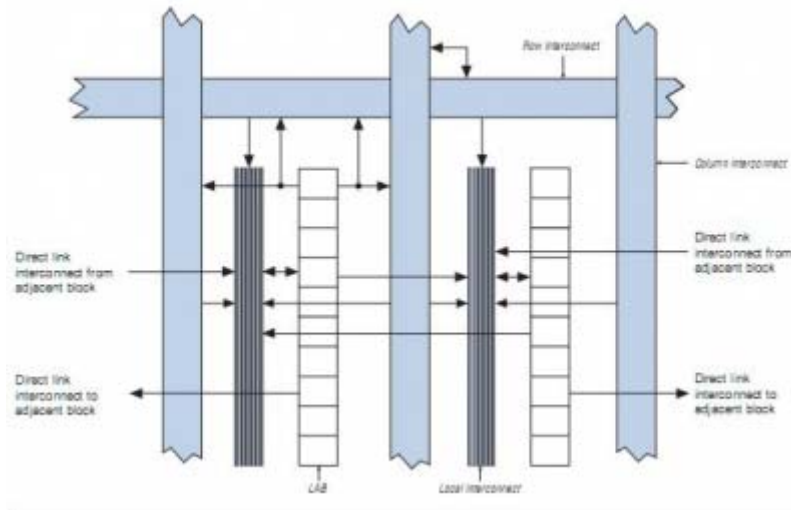
以多块, 大容量 memory, Serdes 可选的特点将成为新的中级 FPGA 市场的标杆了.

为什么 Cyclone, 或者说 Altera 的粗颗粒有一定的好处?

大家都知道, Altera 的结构可以说看起来是大开大合, 实际上是粗中有细.

以前一个 LAB 有 8 个 LE, 大家都可以理解, 后来又发展成有 10 个 LE, 有 16 个 Les

Figure 2-2. Cyclone LAB Structure



如果你有一个计数器,假定在 Cyclone 里和 Spartan 里面跑,Cyclone 和 Spartan 在做 8 位计数器方面应该是不相上下,但是,当在 16 位计数器还要跑同样的速度,而且保证路由资源最简单的时候,Cyclone 的优势,或者说 A 家的优势就来了。

大家都知道,计数器就是进位翻转的传递链路是他性能的关键路径,换句话说,16 位的计数器,就是两个 8 位计数器的级联,唯一区别的复杂度就是 8 位的传递时间如果是 8x,那 16 位的就是 16x 了。用另外一种方式来思考:

8 位的计数器,在到达 FE 这个数字的时候,就用一个 DFF 进行一次隔离,提前一个时钟节拍将进位准备好,这样就将一个 16 位的计数器的复杂度降低到了 8 位 一个样子。可是原本 Cyclone 的 LAB 就有 10 个 LE,因此为了防止毛刺的问题以及刚才需要一个插入的 DFF,就刚好放在一个 LAB 里面,LAB 里面的路由是最快的,而且编译基本不太花时间。同时也为设计流水线的译码技术,提供两级的 DFF 延迟,但是这些全部做到了放在一个 LAB。

大家回顾一下我们的设计,不就是计数器,加多选一,什么 FSM 就是那几个玩意来回的组合。然后中间加流水,再平衡流水寄存器之间的路由。没有新的发明,只有新的组合。但是 Altera 这样的结构就相对来说..哈哈又要吹一下牛皮了。

过去 10 年 FPGA 产业的发展和 FPGA 厂商的挑战?

过去 10 年, FPGA 产业发生了些重要变化, 简要 3 点:

- 逻辑数量超过 10 年前 50 倍,
- 存储容量超过 100 倍
- Serdes 速度接近 10G(Xilinx6.5G)
- 消耗功耗只提高 10 倍多些.

- =====
- 编译算法和综合算法的提高
 - 和结构结合的物理综合
 - 编译平台多 CPU 的支持,例如 4 核 CPU 的 PC
 - 递增编译和编译约束的更好支持

FPGA 三国志-第三篇/SOPC 篇

PLD 的重新定义 PLD 是什么, Programmable Logic Device. 当然今天应该换个称呼了: Processor+Logic+DSP 了, 世界就是这样不停的变化来维持他的稳定。

表 1 列出了 Altera®可编程解决方案的主要历史创新发展过程., 可以看到在 2000 年的时候, FPGA 的厂商已经开始在嵌入式微处理器方面开始崭露头脚。当然起初引来的也有一片哗然。说效率低下, 难以开发。新鲜事物的诞生总是要经过这样的阵痛期。

在 2002 年, 就有众多厂家出来声称都支持嵌入式软核 CPU。甚至也和当时流行的 8051 turbo 以及一些 ARM 进行比较。

表 1. 在 PLD 业界的率先创新

年度	技术发展里程碑	在 PLD 业界的率先创新
2008	Stratix® IV FPGA	业界的首款 40-nm FPGA，具有最高的密度、最好的性能、最低的功耗、最大的收发器带宽(高达 8.5-Gbps 的收发器)，并为 PCI Express Gen 1/2 提供硬核知识产权(IP)模块。
2007	Arria™ GX FPGAs	业界的首款低成本，基于收发器的中端，协议优化的 FPGA。
2007	Cyclone® III FPGA	业界的首款低成本 65nm FPGA，前所未有地同时实现了低功耗、低成本和高性能。
2006	Stratix III FPGA	65nm FPGA 以更高的密度和性能支持高级集成，实现更复杂的产品。
2006	Quartus® II 软件	自然支持 SDC 设计约束
2006	C2H 编译器	嵌入式处理器的首款自动 ANSI C 至寄存器级(RTL)生成工具
2006	Stratix II GX FPGA	速度最快，密度最大的 90nm FPGA 架构，含有工作在 622Mbps 至 6.375Gbps 的 20 个低功耗收发器。
2005	HardCopy® II 结构化 ASIC	精细粒度体系结构；从 90nm Stratix II FGPA 原型无缝移植。
2005	Cyclone II FPGA	90nm FPGA，业界首款低成本 FPGA，成本降低 30%，密度提高 3 倍。
2004	Stratix II FPGA	90nm FPGA，含有名为 ALM 的 8 输入“分段式”查找表(LUT)，以此替代了 4 输入 LUT 体系结构。
2003	Quartus II 软件	可编程逻辑软件包，提供 Tcl 脚本支持。
2003	Stratix GX FPGA	0.13μm FPGA，分块收发器体系结构。
2003	HardCopy 结构化 ASIC	业界唯一原型至量产 0.13μm 的完整解决方案
2003	Stratix FPGA	0.13μm、300mm、高速高密度 FPGA
2002	Quartus II 软件	可编程逻辑设计工具包，支持 Linux。
2002	Cyclone FPGA	世界上成本最低的 FPGA (0.13μm)
2002	SOPC Builder	第一款 FPGA 自动 系统生成工具
2002	Stratix FPGA	世界上第一款带有嵌入式 DSP 模块的 FPGA
2001	HardCopy APEX™ 结构化 ASIC	FPGA 首次无缝移植至低成本结构化 ASIC
2001	系统互联架构	第一款自动生成的 互联架构 ，支持同时主机/从机操作。
2001	Quartus II 软件	推出 Quartus II 设计软件
2001	Mercury™ FPGA	世界上第一款带有嵌入式收发器的 0.18μm FPGA
2000	基于 ARM® 的 Excalibur™ 器件	世界上第一款带有硬件嵌入式处理器的 FPGA
2000	Nios® 嵌入式处理器	世界上第一款嵌入式处理器，适合可编程逻辑应用。

处理器嵌入在 FPGA 中，需要什么样的温床？

的确，在推出 Nios soft processor 两年之后，在我们周边也开始陆续有开始吃螃蟹的了。但是，还是回到结构的老话题上，来谈论一下为什么这个时候出现了由 FPGA 制造商推出的 CPU，实际上在 1997 年，就已经有什么 FPGACPU 的说法甚嚣尘上。而且也有人在 Xilinx 和 Altera 上面分别实现了原形。

开始在 FPGA 嵌入大块 RAM 的时候，很多人抨击这种结构，只是一个动物园里面的老虎，实际上就是和猫差不多的杀伤力。不可否认，它不如分布式 RAM 来的灵活，但是它可以作为一些 Cache，或者是紧耦合的 RAM，还有就是它的规模在容量增加的时候，仍旧有很好的速度，那就可以作为 CPU 的 Cache 了。Altera 的 RAM 都是做在同一列的，很方便级联。另外总线宽度也很容易更改。

还有就是 CPU 里面总是有很多总线，内部总线。由于丰富的行联线的存在，在几列行资源里，就可以轻松集成 RAM，CPU 要的逻辑资源，以及总线的结构。这样，CPU 也就顺利的在 FPGA 的世界里粉墨登场了。虽然起初只是 16bit 的。

2002 年，Altera 的开发软件已经趋于稳定，可以说已经很适合自己产品的结构，而且布线效率非常高。同时 Altera 的 Nios 也上升为 NiosII，这个时候 NiosII 已经蜕变为美丽蝴蝶——一个 32bit 的 RISC CPU。而且 Altera 又采用了非对称的机构实现了革命性的 Avlon 互联体系。关于这个体系的由非对称可以覆盖对称的，或不对称性的应用，我们以后详细聊。总而言之。如果再有一款更好结构的低成本的 FPGA，那无异于在 FPGA 市场上刮起一场飓风！

Xilinx 有 Microblaze，其他的有 ARM 的核，实际上 FPGA 上实现的 CPU，不是由一个 CPU 的内核来决定他的应用，而是谁有最好的互联体系，能和外围设备，和 FPGA 中的 RAM，乘法器，以及其他定制逻辑更好桥接，粘合成一个 SOPC，那才是真正意思上的软核。那究竟什么样子才好呢？

1. 可以增加多个核来弥补自己速度上的缺陷，提高更多的并行处理
2. 灵活的总线结构，可以连接高速，慢速，master，salve 的模块

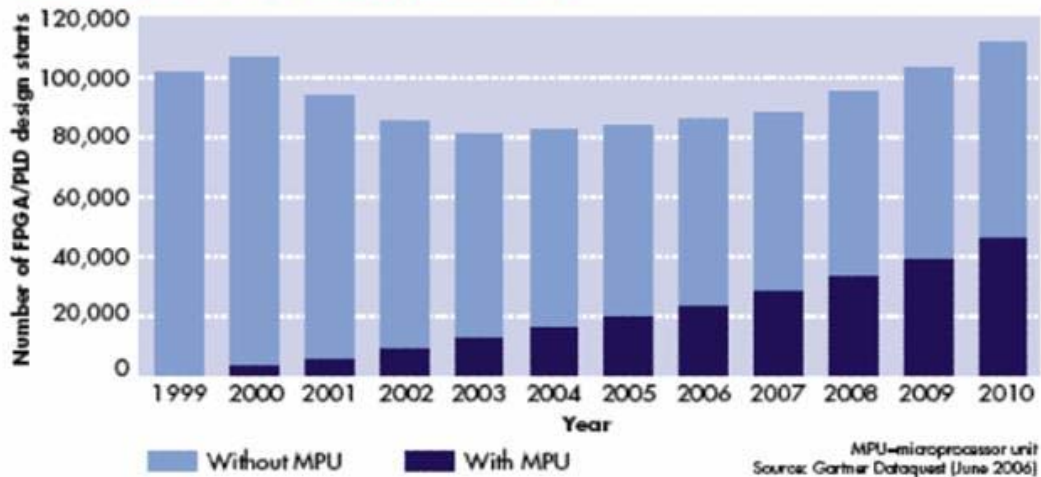
3. 可以将定制化的逻辑，有效集成到 CPU 的指令体系。
 4. 有面向不同应用，可以进行裁剪的 CPU core
 5. 方便片上调试的环境
 6. 可以快速自动实现外设资源冲突仲裁的机制
 7. 可以实现以软代硬，或者以硬代软，在软硬间转化速度与资源的偏重
- 如果有以上的特性，那相信留给设计者的只有是无限的遐想空间。

FPGA 上的 CPU，有太多的元素，你知道吗？

我们从哪里开始讲呢，准备从算盘开始讲 CPU, 很多人说，我是不是太无聊了呢？听我明天说给你听！看似简单的算盘，绝对孕育着无比的大智慧。看是简单的 Avalon，也同样有很多玄机。那就是变变变。来自 Xilinx 的消息，关于 CPU 在 FPGA 设计中的一些预测

FPGA-Based Processing

Growth of FPGA-based processing.



神奇的算盘

上回说到算盘。实际上大家应该都见过这个了不起的发明，只是他出现在你的记忆中的次数越来越少，就忘记了，实际上，很多伟大之处在很早就有历史可以证明的。看看你平时见到的算盘。



算盘的妙处和 Altera 的 LE 排列布局非常相似，或许是巧合吧。

但是这里的偶然也有些必然。或许我们有一天说，FPGA 的结构最早是源于我国的。哈哈，玩笑啊。

1. 可以增加多个核来弥补自己速度上的缺陷，提高更多的并行处理
一个算盘是算盘，级联起来还是算盘，可以横向，也可以纵向
2. 灵活的总线结构，可以连接高速，慢速，master，salve 的模块
算盘是平行也是并行结构，上面的有两个珠子，下面 5 个，2 也可以代表 10，5 也可以代表 10，同时，进位时也可以暂时存储在高档位，也可以留在本档。进位链和 Altera LE 的进位布局也是一样。

看一个更久的照片，当然这个也可以代表 Cyclone 的 layout 了。

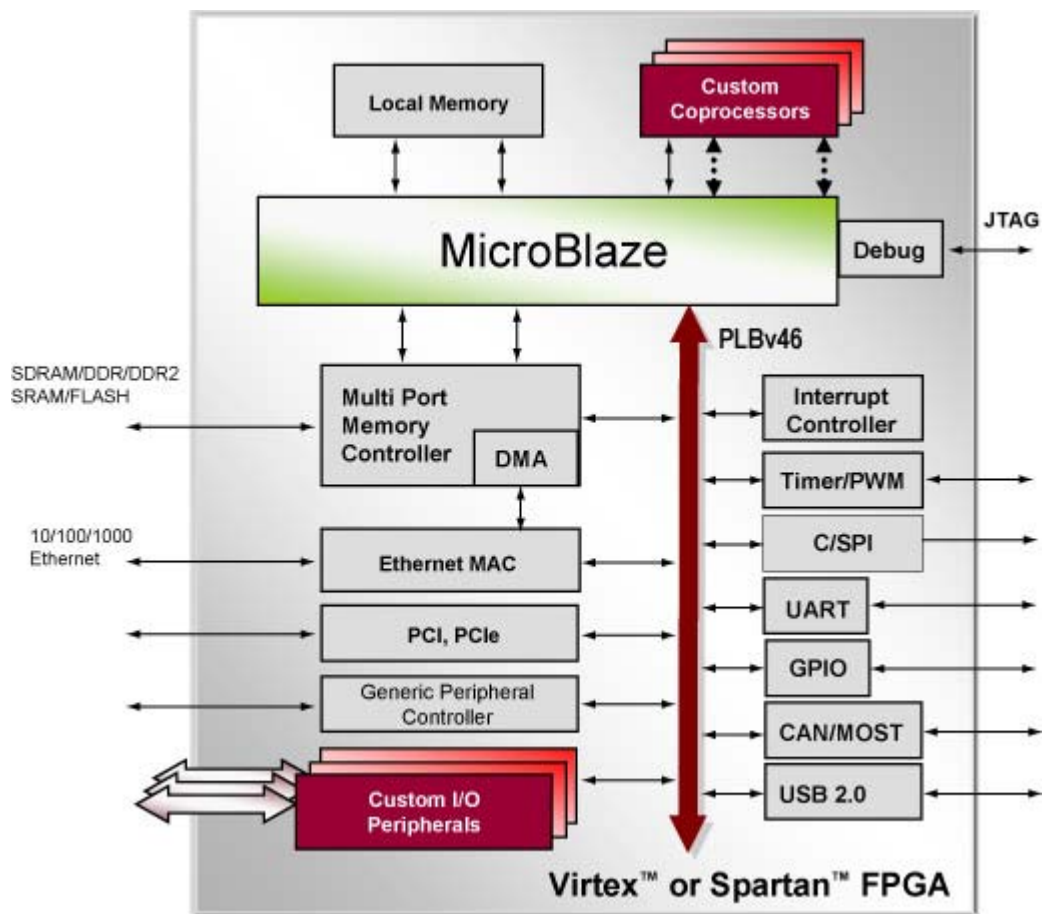


Avalon 的奥秘

总线结构：

全交叉，部分交叉型（适合 FPGA 中的嵌入总线！）

Xilinx 的是共享总线型。仍旧属于传统型设计，只是将其搬移到了 FPGA 中。

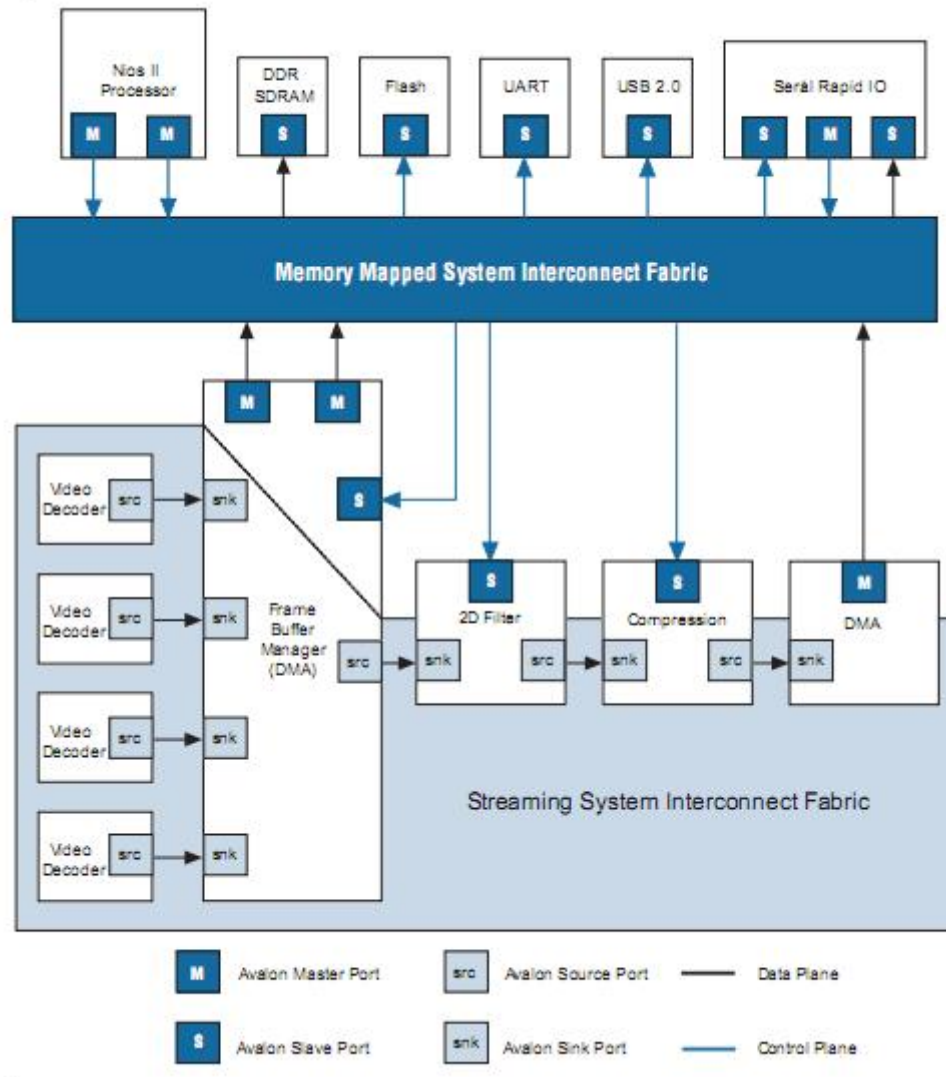


处理器接口主要抽象为：

1. 共享寄存器
2. RAMs
3. FIFOs

回头看 Avalon：

Figure 6-4. Video Data and Control Planes



上面看起来，这两个 softcore 都差不多，看了下面的特点，你就有答案了。

Selecting Hardware Architecture

Bus

- achieve relatively high clock frequencies
- expense of little
- no concurrency
- common arbitration unit.
- all masters compete for access to

Full Crossbar Switch-NiosII Avalon

- concurrent transactions
- flexible
- high throughput
- large multiplexers
- more masters and slaves are added
- avoid large crossbar switches

Partial Crossbar Switch--Avalon is okay

- connectivity to a subset of the slaves
- provides the optimal connectivity

- operates at higher clock frequencies
- interconnect fabric consumes fewer resources.
- ideal for ASIC or
- FPGA interconnect structures
- slave side arbitration.

Streaming-Avalon Streaming Map

- high speed data transfers
- point-to-point connections between source and sink
- eliminating arbitration

Dynamic Bus Sizing

Understanding Concurrency

Create Multiple Masters

- General purpose processors, such as Nios II
- DMA engines
- Communication interfaces, such as PCI Express

Create Separate Datapaths

Use DMA Engines

- data width
- clock frequency
- more DMA engines

Include Multiple Master or Slave Ports

- increases the concurrency

Create Separate Sub-Systems

Increasing Transfer Throughput

increasing the transfer efficiency

lower frequency devices can be used

Using Pipelined Transfers

- Maximum Pending Reads
- Selecting the Maximum Pending Reads Value

Pipelined Read Masters

Increasing System Frequency

strategy

- introduce bridges to reduce the amount of logic
- increase the clock frequency.

Tatic

Use Pipeline Bridges

- Master-to-Slave Pipelining
- Slave-to-Master Pipelining
- waitrequest Pipelining

Use a Clock Crossing Bridge

Consequences of Using Bridges

- Increased Latency
- Limited Concurrency
- Address Space Translation

Reducing Logic Utilization

Reducing Power Utilization

Reduce Clock Speeds of Non-Critical Logic

Clock Crossing Bridge

Avalon-MM master ports operating at a higher frequency to slave ports running a lower frequency

- PIOs
- UARTs (JTAG or RS-232)
- System identification (SysID)
- Timers
- PLL (instantiated within SOPC Builder)
- Serial peripheral interface (SPI)
- EPCS controller

- Tristate bridge and the components connected to the bridge

总线结构是 SOPC 的最重要的战场，也是区分 SOPC 是否高效的关键因素！

一说到 SOPC，最基本的集成首先是 CPU 的内置，但是 CPU 的有下列因素决定总线结构。

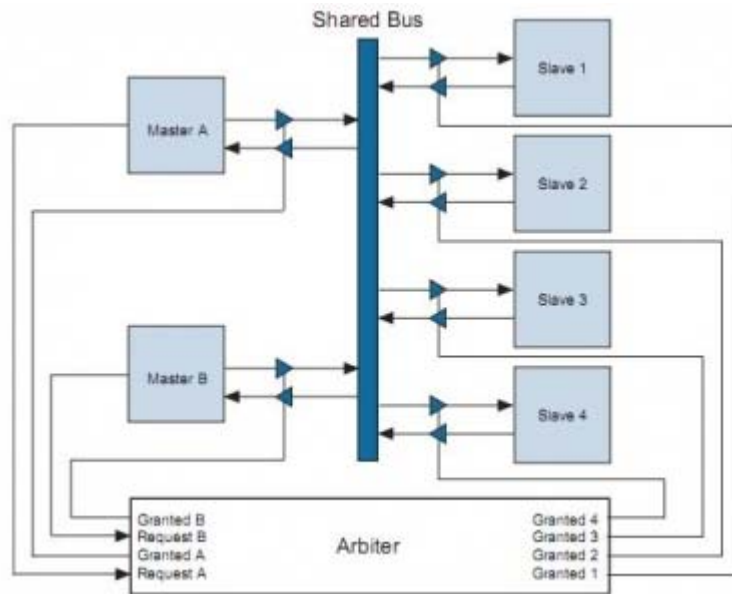
- CPU—冯诺依曼结构，CISC 体系多采用这种方式，特点，就是指令功能强大，总线结构复杂，堆栈结构设计庞大。因为要解决所有 Master 和 Slave 之间的运算和重写，另外寻址模式非常丰富。缺点是面积偏大导致这类 CPU 的频率不能太高！
- CPU—哈佛结构，总线独立，能够提高并行，基本上以数据流加工为主。所以 DSP，以及一些 RISC 的体系也很多。

这里声明，RISC，CISC 和 CPU 结构无必然联系。

简单说来：RISC 是以总线结构越少越好，通路尽可能简单。例如大部分指令面向寄存器，然后数据操作多数在寄存器完成。有很多甚至基于堆栈。

共享总线的速度设计与性能总结：

1. 完成相对高的总线频率
2. 相对低廉的成本
3. 不能并行！（这个是缺点）
4. 共用的仲裁结构（缺点，反应速度慢）
5. 所有主设备对从设备的存取都存在竞争关系（同时访问时候）

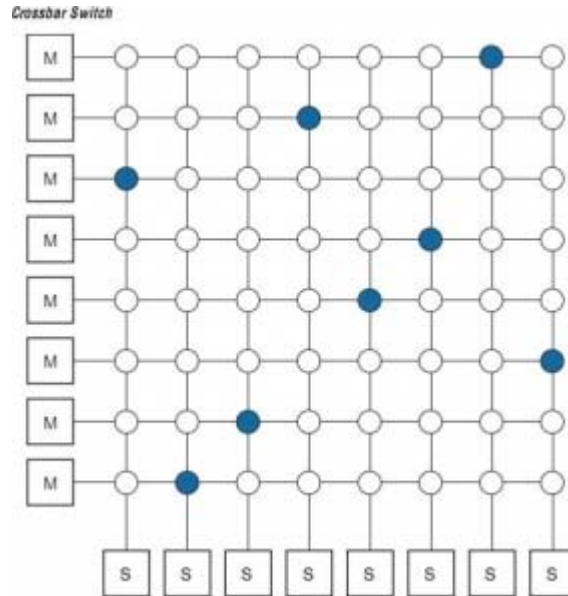


当然在 FPGA 中，双向三态的实现是依靠多选一和 DFF 的隔离来实现的，而且，多数为同步设计。当然在现代分离系统的设计中，已经很早有这样的先例。

例如一个 ARM 和 TI 的 DSP 进行协同的系统，TI DSP 中有对总线进行释放的功能机制，释放的时候，可以由 ARM 来对 DSP 中的数据结果进行交换，或者是操作参数的更新。然后再将总线控制权返回给主控的 ARM. 在 FPGA 中，也是同样的道理。

NIOS II 的总线结构 Avalon 是具有全数字交叉的总线结构，支持多主并发的结构。

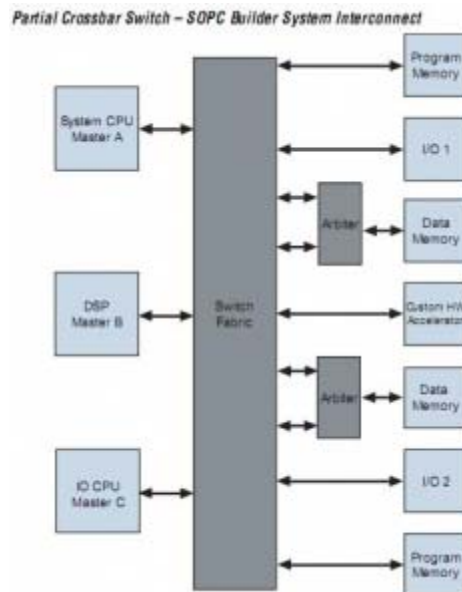
FPGA 中，由于 FPGA 中丰富的互联结构，以及 FPGA 中本身的逻辑胶合的本身设计定位，决定了在 FPGA 中的 SOPC 的总线结构：全数字交叉的互联结构是 SOPC 在 FPGA 中性能发挥的调节棒！



很多人对此有以下担忧：

1. 仲裁结构复杂，
2. 规模过于庞大
3. 对设计工具要求高（每次互联设计，可能要修改太多东西）

Altera 采用了一个折中的方式就是**部分数字交叉结构**。



而且将仲裁机制放在了从端，这样做的好处，**就是有冲突访问可能的从设备，就对他增加仲裁**，好处：

1. 降低了总线规模
2. 提高并行性
3. 仲裁效率高

没有优良的总线结构，一切都是空谈，有效的机制，可以弥补 CPU 的性能的低能化！

总线接口的抽象！Xilinx 和 Altera 志同道合

尽管总线接口的抽象不同，但是，不代表总线结构一致。X 是总线共享型，A 是数字交叉型。

例如 Avalon-MM, Avalon-SM

Xilinx 是 PBv, 和 Fastlink 来对应

MM 是意思采用存储地址映射抽象，就是将所有端口定义为可以用内存地址来寻址操作。而 SM 是代表流媒体数据型，也就是一般和 DMA 都有一定的联系，当然，也是一种要不就是不断接收数据流，要不就

是发送数据流的接口。例如，VGA 的数据缓冲，DSP 信号数据的输入和输出，例如 AD 数据的输入，以及 DA 的输出，同时具有 FIFO 功能。

并行，并行，再并行。

FPGA 中用的设计语言，VHDL, Verilog HDL 等，这些都是并行的，FPGA 之所以能在很多 DSP 领域进行大显身手，也是因为他良好的并行扩展性。那同样。在 SOPC 中，要提高系统的整体性能，还是要并行，并行再并行！

上面就结构已经进行了简单的说明。那么在数据处理的单元上如何设置呢。

1—动态总线宽度的适应。

当你要将 32bit NIOS 和 8bit，或者 16bit 的系统进行连接的时候，他应该能自动适应！

2—增加多个主控单元，对一些控制进行有效并发。

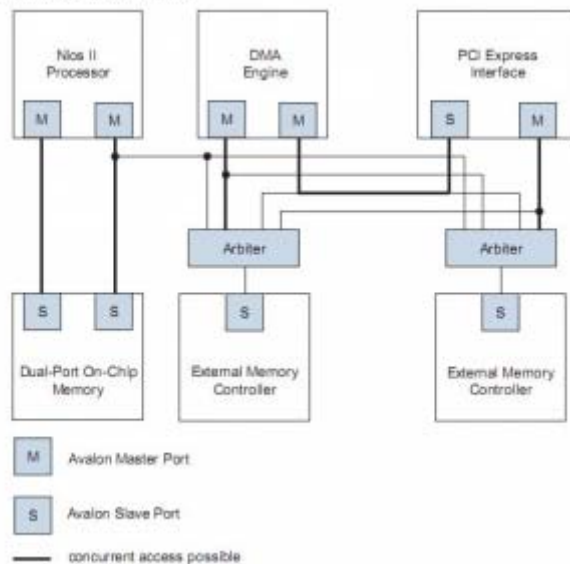
归纳为三点：

多增加 CPU 的个数，SOPC 支持多 CPU，而且现在很多 CPU 也在发展多核。商用 PC 已经很成熟了。

多增加 DMA 的通道个数，这样 CPU 可以干别的事情，至少数据吞吐加大

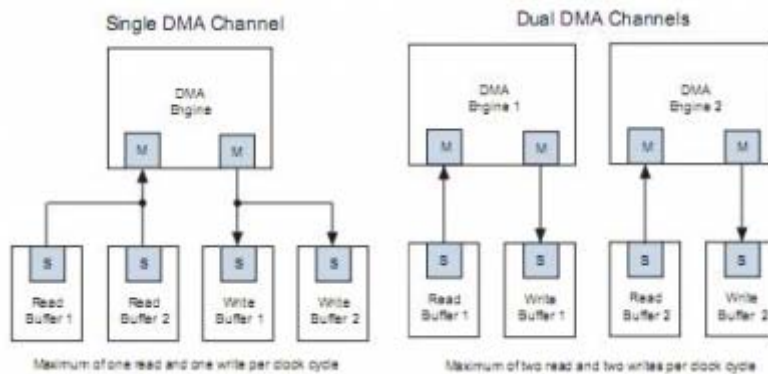
增加 PCI-e 的设备，这样，NIOS, FPGA 不方便处理的，可以通过这样的接口进行功能上的折中。当然以后也可以增加 RapidIO 来和 TIDSP 进行数据的互换。

6-5. Multi Master Parallel Access



3—创建分离的数据通路

4—用多 DMA 进行辅助。 例如



没有新的发明，只是新的组合！

实际上今天很多所谓的创新，也就是是一些不同设计元素的合并和组合。TSMC，在20年前开始代工的时候，有很多人质疑这种方式，实际上这个是一种商业模式的创新。Altera是前期一些Fabless的厂商之一，很多人也怀疑没有自己的工厂？怎么可以做好产品。但是回顾今天，大家不都是走这样的路吗。当然也有一些自己领域的佼佼者，内存，CPU还是自己开灶。

话题拉回，当你一个系统已经不方便的时候，或者是某种功能要求有很突出表现的时候，两个或者多个CPU的系统，或者是拆解再增加必要的子系统。也是一种创新。

分割为多个子系统的好处：

1. 方便管理
2. 由于分割后，从设备可以分在不同的有限个主设备中，提供并行，降低冲突。
3. 从系统之间耦合较松，就是说可以
依靠共享内存，消息传递，信号灯互锁，FIFO等机制进行同步和信息交换。

一生二，二生三，三生万物.....

现在的工具都很方便复制和例化多个功能单元。当你设计好一个上面的系统的时候，你也可以将他们又构成一个模块，然后复制这些模块来达到更大的系统。这种在DSLAM的设计，或者是多路以太网复用到光设备等经常应用。

因此，

一生二，二生三。。。。

当然这种思想是建立在大批量复制功能单元的策略。

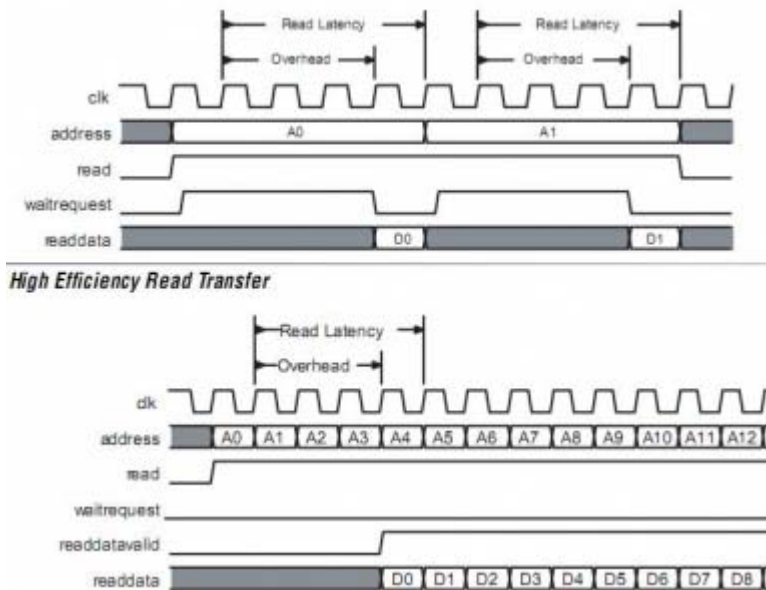
另外，还有一种大家熟悉的方式，

就是提高局部的流水动作以提高系统吞吐。你不仅要有火尖枪，同时也要有混天绫。互相配合才是自然之道。

流水的数据操作分为两类：

- 读流水
- 写流水

读流水和写流水主要是针对目前的一些存储外设来设计的，就是一个主设备可以在第一读的动作发生，但是没有数据返回的前提下，连续发出多个读的动作，然后在适当的时钟延迟后，连续得到由连续发出读的指令而产生的返回数据序列。好处是，用个图来表达就明白了。



当然了，你可能说，这个说易行难，设计经验不足。没有关系，SOPCbuilder的master和Slave接口，已经将这些都做好了。如果你想自己设计一个，实际知道这样的时序，你也可以的。不外乎就是要做一个读和返回地址的计数器，计数器的差值是在Readdatavalid的返回时候发出的地址数目，当然你也要设计好数据的宽度

ARM Vs Nios II

大家对ARM和NiosII喜欢比较，今天就说上几句。

最快的NiosII (250Mhz)比ARM7快，
最小的NiosII比最小的ARM7要小

另外 NiosII 已经有了 MMU, 而且对 Linux 进行了支持

ARM7 是把西瓜刀

NiosII 是把瑞士军刀。NiosII 做一个分型算法借助 C2H, 速度是 ARM7 的 400 倍, 不算慢。但是 ARM7 的生态资源广, 这两个应用实际上是相辅相成。

FPGA 三国志-第四篇/工具篇

FPGA 的公司实际上换而言之, 也可以成为 EDA 的公司了, 只是他们在这方面的能力是否强大而已。这 3 家公司中, 拥有自己 HDL 的公司只有 Altera。可能很多人不太清楚, 就是 AHDL, 当然这种语言的辉煌期已经过去了。这里说这个, 绝对不是鼓励大家用这个语言。

这 3 家主要的开发软件大家都知道, 在中国应该以 Quartus II 用的最广。ISE 次之。但是现在这两大软件的外观都已经有些一致的地方了。但是他们近乎同样外观的软件, 却还是有地方非常不同。

Altera 在 2000 年前, 在日本市场独领风骚。大家知道日本是漫画的国度, 他们总是喜欢用图形表达, 这可以证明 Altera 的图形输入做的是炉火纯青。但是, 现在都是语言输入为主导了, 因此 Altera 的这个优势才渐渐淡化。但是其他图形界面仍旧有其鲜明的特色, 简单容易。

Altera 的软件像 Canon 的相机, 很傻瓜, 对着景物按个快门。当然也有些地方, 你也可以仔细微调。

Xilinx 的软件洋洋洒洒, 很专业, 像部哈苏相机, 你在照相前, 先成为相机的专家, 才能成为设计的专家。有人开玩笑了, 这是拿东西来说事。那在这里引用 Altera 公司一些思想, 给大家介绍一下。实际上, 这个特点非常容易记忆。大家都知道 TIPS 这个词吧, 就是提示的意思。那在这里就给你一些提示, 让你可以过目不忘!

TIPS——提高设计效能的秘方!

这几天准备一下, 尽快让大家看到下面的章节!

从 TIPS 中的 T 开始说, FPGA 的趋势如下



1——首先说一下眼前的大环境是怎么影响你的设计的。过去的设计规模应该还是比较小, 一个 FPGA 工程师, 写个状态机, 写个好的代码, 这个就是相对来说的高手了。但是你要知道这种高手都是关起门来自称老大的。实际出门走多远很难说。因为现在器件的发展速度是很快的。

2——规模的变化

1998——一位当时的 PLD 的厂商说 世界上有 1000 个宏单元的 PLD 就可以应付 95% 的设计了。今天你认为怎样?

1998——最大的 FPGA 内含 10000LEs, 今天的 CycloneIII 的第二小的家伙就是这个身段。

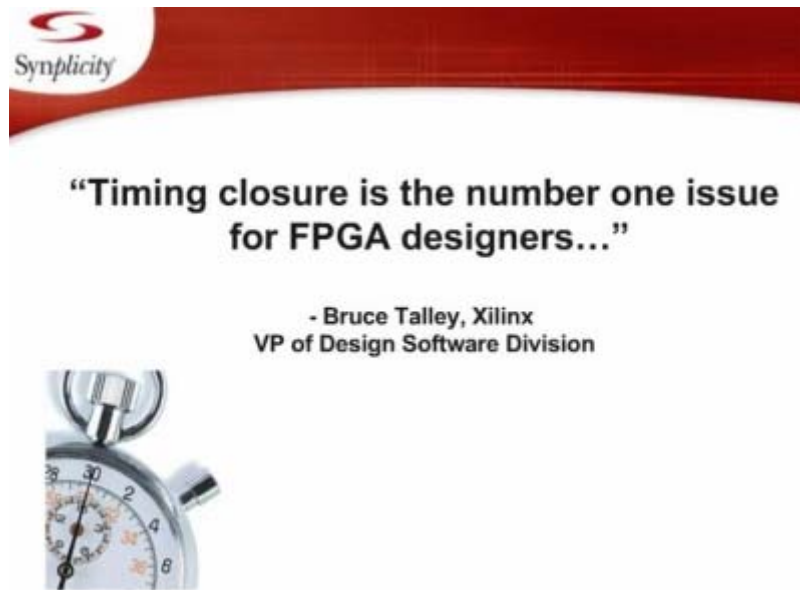
2008 年——现今最大的 FPGA 内含 530, 000LEs, 比 1998 年增大 50 倍! 你还能用以前的设计方式来应付今天这个庞然大物吗?

3——规模的变化, 带来**复杂度的提高**, 调试也成为关键?

芯片规模大的时候, 你就会要有些测试工程师的需求。因为人自己总是看不清自己的缺点。

4——一个模块跑个固定的速度, 是否可以和其他模块都配合起来, 还能保持性能?

目前EDA厂商众口一词的话：20KLEs 规模以上的设计，时序收敛是第一目标
当有问道 Xilinx 的软件设计者时候，他们也承认了这样的事实：



好了，第一个题目诞生了。T-IPS 中的 T 就是 TimeQuest，以前 FPGA 的规模和设计 ASIC 不能相比，今天已经可以了。在 FPGA 中集成很多 IP 已经不是什么难题了。TimeQuest 是依照 SDC 来进行工作的，SDC，是 ASIC 中设置约束的工具，换言之，也算是你如何指挥你的软件，将你设计的模块让他们统一协调好时间，然后保证最终的结构是你想要的。你不仅要看到设计，还要跑到那样的速度的时候，他还是你要的功能，有人说了，以前没有这样的工具吗？有，但是那不是针对大的设计的。当然大也是相对的，到了一定的规模，就必须使用另外的标准了！什么能帮助你达到时序收敛，什么可以加速你找到哪些地方不满足，TimeQuest！

目前支持这个 SDC 的有以下工具，当然 Quartus 支持的最完整。潜在的秘密是，Altera 的器件的规模也是最大的了。

- Altera - TimeQuest à SDC based Timing analysis
- Actel - Offers preliminary SDC support
- Xilinx & Lattice - No SDC support

可能你会问，能不能详细点说一下：

1. TimeQuest 是 Altera 第二代的时序分析工具
2. 你不需要记忆什么语法，完全的 GUI 界面，而且也支持模板输入功能。
3. 他生成的约束是基于 Synopsys 的工业标准的。

TIPS 中的 I 是指“递增编译”，就是英文“Incremental compile”的第一个字母是 I。

刚才说了，当你有时候改动了一部分设计的时候，你以前好不容易调好的时序，可能因为重新改动的影响，而在编译路由后，将没有改动的设计在布局 and 速度上发生了不期望的改变。那这个递增编译就是干这个的。他可以向制导武器一样，让你指哪里，打哪里。同时缩短编译时间，提高你的工作效率！当然这里绝非这么一句话就可以涵盖了。你也要有设计分离，规划布局等工作。

现在经常有提到我的产品是最好的 Price，但是，现在由于系统规模增大，Power 也是一个很重要的制约，功耗在某些设计中甚至是第一位的。Quartus 中有关于根据 Power 的要求来智能编译设计的约束。就是 TIPS 中的 P，也就是 PowerPlay。

最后一个 TIPS 中的 S 就是指 Systemlevel，或者是 SOPC，因为你的设计可能要有各种接口，外部的，内部的，系统的互联。总线的对接，时钟域的交叉，这些，都可以用 SOPC 的工具 SOPC builder 来实现。不用内置处理器也可以。同时要求你在系统级的高度进行设计，这样就提高设计速度。在第三篇中有详细讲。

生产力发展的标志-生产工具，你现在和 15 年前设计方式的根本改变！

不管 A, X, L, 哪一家 FPGA 玩家，这三年来都在开发工具上投入了重兵。此前有网友争论——

Lattice 用的是 Synplify 进行的综合，效果如何如何高。这个论调 10 年前是天经地义，颠覆不破的真理，但是今天再这样说，绝对偏颇！

Synplicity 公司不少人后来都转投 Altera 门下。当时 Sy 最先得到 Xilinx 的结构的时候，大力发展所谓物理综合等一系列亮点，但是却对 Altera 提及甚少，大家都知道 Sy 的发家史，失去了 FPGA 厂商的支持，基本上也就失去温床。在 2004 年 Quartus II 4 以后，Quartus 的综合能力就已经与 S 家的不相伯仲了。Xilinx 在 10 年前收购多家 EDA 厂商，例如 StateCAD，当时用流程图生成状态机的 VHDL 和其他的一些 厂商的时

候, Quartus 都是建立在自己的固有体系下. 如果单独从产品器件上, 只有工艺和功耗是未来需要争斗的地方, 那么, 综合能力和相关路由(就是紧耦合的模块放在一起)能力, 以及 EDA 工具的全面性才是真正 FPGA 厂商的绝对命门. S 家一共有 250 个软件开发人员. A 家可能就远远不止了.

另外 Cadence, Synopsys, Magma, Mentor 等大哥, 也就是卖个综合和核对工具, 价值点相信大家已经悟到几分吧.

QuartusII 在今天, 作为 FPGA 摄影大师的工具来说, 不仅是精密的哈苏, 更是 Canon 门下 Eos MarkII 的便捷傻瓜的代表. QuartusII 就是今天 FPGA 的第一生产力

