

欢迎来到控制中文网

www.cechinamag.com

可编程逻辑器件应用设计技巧百问

1. 什么是.scf?

答: SCF 文件是 MAXPLUSII 的仿真文件, 可以在 MP2 中新建.

2. 用 Altera_Cpld 作了一个 186(主 CPU)控制 sdram 的控制接口, 发现问题: 要使得 sdram 读写正确, 必须把 186(主 CPU)的 clk 送给 sdram, 而不能把 clk 经 cpld 的延时送给 sdram. 两者相差仅仅 4ns. 而时序通过逻辑分析仪测试没有问题. 此程序在 xilinx 器件上没有问题. 这是怎么回事?

答: 建议将所有控制和时钟信号都从 PLD 输出, 因为 SDRAM 对时钟偏移(clock skew)很敏感, 而 Altera 的器件 PLL 允许对时钟频率和相位都进行完全控制. 因此, 对于所有使用 SDRAM 的设计, Altera 的器件 PLL 必须生成 SDRAM 时钟信号.

要利用 SDRAM 作为数据或程序存储地址来完成设计, 是采用 MegaWizard 还是 Plug-In Manager 来将一个 PLL 在采用 Quartus II 软件的设计中的顶层示例? 可以选择创建一个新的 megafuntion 变量, 然后在 Plug-In manager 中创建 ALTCLKLOCK(I/P 菜单)变量. 可以将 PLL 设置成多个, 或是将输入划分开来, 以适应设计需求. 一旦软件生成 PLL, 将其在设计中示例, 并使用 PLL 的 "Clock" 输出以驱动 CPU 时钟输入和输出 IP 引脚.

3. 在 max7000 系列中, 只允许有两个输出使能信号, 可在设计中却存在三个, 每次编译时出现 "device need too many [3/2] output enable signal". 如果不更换器件 (使用的是 max7064lc68). 如何解决这个问题?

答: Each of these unique output enables may control a large number of tri-stated signals. For example, you may have 16 bidirectional I/O pins. Each of these pins require an output enable signal. If you group the signals into a 16-bit bus, you can use one output enable to control all of the signals instead of an individual output enable for each signal. (参考译文: 这两个独特的输出使能中每个都可能控制大量三相信号. 例如, 可能有 16 个双向 I/O 引脚. 每个引脚需要一个输出使能信号. 如果将这些信号一起分组到一个 16 位总线, 就可以使用一个输出使能控制所有信号, 而不用每个信号一个输出使能.)

4. 关于 vhdl 的问题: process(a, b, c) begin... end process; 如果 a、b、c 同时改变, 该进程是否同时执行三次?

答: PROCESS STATEMENTS 中的执行跟逻辑有关系, 假如是同步逻辑, 则在每次时钟的触发沿根据 A, B, C 的条件来执行一次; 假如是异步逻辑, 则根据判断 A、B、C 的条件来执行. 一般我们都推荐使用同步逻辑设计

5. 在设计最初, 由于没有将时钟信号定义在全局时钟引脚上, 导致 MAXPLUS II 在时间分析时提示错误: (时钟偏斜加上信号延迟时间超过输入信号建立时间). 全局时钟引脚的时钟信号到各个触发器的延时最小, 有没有可能通过编译软件设置, 将普通 I/O 脚上的

时钟信号也经过芯片内部的快速通道以最小的延迟送到每个触发器时钟引脚？

答：you can register that signal and assign it as the global signal, by the step flow: assign->logic option->Individual logic options->Global signal. But you'd better input the clock signal through the dedicated input pin. (参考译文：可以寄存这个信号，并将它指定为全局信号，步骤如下：指定—>逻辑选项—>个别逻辑选项—>全局信号。但是，最好通过专用输入引脚输入时钟信号。)

6. 用 MaxplusII 软件设计完后，用 Delay Matrix 查看延迟时间。由于内部触发器的时钟信号用了一个输出引脚的信号，譬如将一引脚 ClkOut 定义为 Buffer, Clkout 是一时钟信号，然后反馈到内部逻辑，内部逻辑用此信号作为时钟信号，但用 Delay Matrix, 却查看不到一些信号相应于 ClkOut 的延迟，因为 ClkOut 是一 Output 引脚，在 Delay Matrix source 一栏中没有 ClkOut 信号，如何解决这个问题？

答：这种做法在逻辑设计中称为 GATE CLOCK，所谓 GATE CLOCK 就是将设计中的组合逻辑结果拿来作时钟信号，这是一种异步逻辑设计。

现在都推荐使用同步逻辑设计方法。可以将该信号(CLKOUT)拿来作使能信号，即 ENABLE 信号，而时钟信号还是采用原来的统一时钟，使设计用尽量少的同步时钟，这样一来就还是用 DELAY MATRIX 来分析原有的时钟。

7. 我是一个 epld 的初学者，目前看到 xilinx 的 Virtex-II 中嵌入大量的资源如：powerpc、ram 等，究竟如何在 fpga 中使用这些资源？

答：Xilinx Virtex-II 中嵌入的资源非常丰富，如 BlockRAM、Digital Clock Manager、On-chip termination 等等。ISE 4.2i 软件完全支持这些资源。可以举出单元库中相应基本数据的实例。Xilinx Core Generator 中也还支持 BlockRAM 等特性。至于 PowerPC 和 MGT 设计，可以使用 Virtex-II Pro 开发者套件。

8. 在设计中，往往需要对某个信号做一定（任意长）的延时，有没有好的方法来实现？而不是采用类似移位寄存器的方法来延时。

答：使用移位寄存器在 FPGA 中对信号进行延时是一种好方法。Xilinx Virtex 架构中每个对照表(LUT)都能够设置成为具有可编程深度(最多为 16)的移位寄存器。这就提供了一种高效的途径来在 FPGA 中实现移位寄存器。无须使用触发器就可以实现一个 16 位寄存器。作为一个好的设计习惯，任何情况下都不要通过闸延迟来实现延迟逻辑。

9. ISE 中的 PAD TO PAD CONSTRAINT 是否是包括输入输出的 pad 时延之和再加上输入输出之间的组合逻辑的时延？还是只是输入输出之间的组合逻辑的时延？

答：Xilinx PAD-to-PAD constraint 的确涉及到输入输出 PAD 时延。这从布局后时序报告中可以看出。

10. 由于现在的设计基本上都是同步设计，那么 PAD TO PAD CONSTRAINT 在什么情况下使用？

答：虽然现今多数设计都是完全同步，但仍有一些情况需要从一个输入引脚到另一个输出引脚的纯粹组合路径。因此，仍然需要 PAD-to-PAD constraint 控制这些路径的时延。

11. 如何在 ISE 中看到 PAD TO PAD 的布线情况？

答：通常不必在意信号在 FPGA 内的路线，只要它涉及到时序问题。这种工具将对以优化的

方式对设计进行路由。如果希望检查具体路由, 可以使用 Xilinx FPGA Editor, 它包含在 ISE4.2i 软件中。

12. 在 Xilinx Foundation 3.1i 下用 JTAG PROGRAMER 下载程序到芯片中, 可是总是出现如下错误: If the security flag is turned on in the bitstream, programming status can not be confirmed; others, programming terminated due to error. 测量电路信号, 没有相应的波形, 显然下载没有成功。所用的芯片是: Xilinx Spartan2 XC2S50TQ144. 怎么解决?

答: This is a security feature. By disabling readback, the configuration data cannot be read back from the FPGA. This prevents others from pirating your intellectual properties. You can enable or disable this feature during bitstream generation.

The proper way to determine if the configuration is finished without error is to check the status of the DONE pin on the FPGA. DONE pin should go high if the bitstream is received correctly. Also, since you are using JTAG configuration, please make sure you have selected JTAG clock (not CClk) as your Startup clock during bitstream generation. (参考译文: 这是保密功能。通过禁用回读, 配置数据不能从 FPGA 回读。这可以防止其他人盗用你的成果。在生成位元流过程中, 可以启用或禁用这个功能。

确定配置是否准确无误地完成, 适合的方法就是检查 FPGA 上 DONE 引脚的状态。如果正确地接收了位元流, 则 DONE 引脚将会升高。而且, 既然使用 JTAG 配置, 就要确保在生成位元流过程中, 已经将 JTAG 时钟(而不是 CClk)选作了 Startup 时钟。)

13. Xilinx Virtex 架构中每个对照表(LUT)都能够设置成为具有可编程深度(最多为 16)的移位寄存器。可否理解为, 在写设计的时候如果设计了一个深度不大于 16 位的移位寄存器, ISE 综合时就会用一个 LUT 来替代它?

答: Most synthesis tools (e. g. Synplify Pro, Xilinx XST) are able to infer LUT based shift register (SRL16E) from your source code. Even for depth greater than 16, the tool is smart enough to infer multiple SRL16E to realize the shift register. Another way to utilize this feature is to instantiate an SRL16E in the source code. You can refer to the Library Guide in the Xilinx ISE software package for more details. (参考译文: 大多数综合工具, 例如 Synplify Pro 和 Xilinx XST, 都能根据源代码中的移位寄存器 SRL16E 来推断 LUT。即使是深度大于 16 的情况, 此类工具也能够推断出多 SRL16E, 从而实现移位寄存器。利用此功能的另一种途径是在原代码中例示一个 SRL16E。详细说明可以参考 Xilinx ISE 软件包中的库指南。)

14. LUT 是实现组合逻辑的 SRAM, 怎样实现一个时序的移位寄存器, 是不是必须加一个触发器来配合 LUT?

答: The LUTs in Xilinx Virtex architecture are not simply combinational logic. When it is configured as 16x1 RAM, the write operation is synchronous. When it is configured as shift register, there is no need to consume any flip-flop resource. In fact the internal circuitry of a Virtex LUT is more complicated than what it looks like. (参考译文: Xilinx Virtex 结构中的 LUT 不是简单的组合逻辑。当它被配置为 16x1 RAM 时, 写操作是同步的。当它被配置为移位寄存器时, 则无需消耗任何 flip-flop 资源。事实上 Virtex LUT 的内部电路比看起来更复杂。)

15. 在 foundation 3.1 环境里怎么找不到启动 testbench.vhd 的程式?

答: 伴随 Foundation 3.1i 出现的仿真器为门极仿真器, 因此你不能在这种设计环境下以

VHDL 级运行仿真. vhd1 代码必须在你运行任何仿真之前进行综合. 因此, 在 Foundation 3.1i 环境下并不能使用 vhd1 testbench. 作为替代方式, 你可以编写仿真 script.

实际上, Foundation 3.1i 是一款相对较老的软件. Xilinx ISE 软件中支持 HDL testbench, 它的最新版本为 4.2i.

16. 关于双向口的仿真, 如果双向口用作输入口, 输出口该怎么设置?

答: 做仿真时, 软件会自动地将 IO 口(包括双向口)的引脚本加入到. SCF 文件中. 先新建一个 SCF 文件, 然后在 NODE->ENTER NODES FROM SNF->LIST, 将列出的所有 IO 引脚(包括了双向口)都加入仿真文件中, 就可以进行仿真了.

17. 关于 ACEX1K 的 I/O 脚驱动能力. ALTERA 计算功耗的 datasheet 中: 对 ACEX1K 器件, PDCOUT (power of steady-state outputs) 的计算就是根据 IOH, IOL 来计算的, 能否告诉我 ACEX1K 芯片的 IOH, IOL 分别是多少?

答: 关于 ACEX1K 的 IO 驱动能力, IOH&IOL 的大小可以从 ACEX1K 的数据手册中查到 (ACEX. PDF PAGE 50/86).

18. 设计中 Vccio=3. 3V, 假如 IOH=20mA, IOL=20mA, n=10 (Total number of DC output with steady-state outputs), 如何计算 PDCOUT?

答: 关于功耗的计算可以参照 AN74(P2)中的功耗计算公式.

19. 当 Vccio=3. 3V 时, 对于输入脚, 它兼容 TTL, CMOS 电平; 对输出脚, 它是否也兼容 TTL 和 CMOS 电平? 对 CMOS 电平, 是否需要用 OpenDrain 加上下拉电阻来实现?

答: ACEX1K 器件引脚兼容 TTL 与 CMOS 电平. COMS 输出是否要加上拉电阻要看外部接的 CMOS 电平, 假如说接 5V COMS 则需要上拉. 详细情况可以参照 AN117.

20. 将 EPC2 与 EPF10K30A 连接成 JTAG 菊花链的形式, 在调试阶段可以跳过 EPC2 直接配置 EPF10K30A, 而在配置通过验证以后再利用 EPC2 的 JTAG 口将 EPF10K30A 的配置信息固化到 EPC2 中去. EPC2 的专用配置端与 EPF10K30A 配置端连接, 当系统脱离 JTAG 电缆上电配置时, 由 EPC2 完成对 EPF10K30A 的配置. 这个过程中有一个疑问, EPF10K30A 相当于有两个配置通道(一个通过 JTAG, 一个通过 EPC2), 当其中一个配置通道工作时, 另外一个配置通道的存在是否会影响到配置过程的正常进行呢? 如果相互影响, 怎样才能做到两种方式同时存在又互不影响呢?

答: 可以使 10K30A 拥有两个不同的下载方式, 在板子上做一个跳线开关即可. 也可以从当地的代理得到该模块的参考设计.

21. ALTERA 是建议直接使用 MAXPLUSII 或 QUARTUS 编译 HDL 源代码, 还是使用第三方 EDA 工具 (如 SYNPLIFY、LeonrdoSpectrum 或 SYNOPSYS) 先把 HDL 源代码编译为 edf 文件后再使用 ALTERA 的工具编译?

答: ALTERA 建议用第三方的工具将 HDL 源代码编译为 edf 文件后再使用 ALTERA 的工具进行布局布线. ALTERA 的 MAXPLUSII 和 QUARTUS 也都自带有 HDL 的综合器, 一些简单的设计可以直接在 MP2 或 QII 中编译即可. 而且可以直接在软件中后台调用第三方的 EDA 工具.

22. 用 MAXPLUSII 或 QUARTUS 多次编译同一设计生成的带延时的网表文件中的延时是否一样?

答：用 MP2 或 QII 多次编译同一设计成的带延时的网表文件中的延时是一样，但要保证该网表文件没有修改过。

23. 在编译前设定一个模块的 Synthesis Style 为 FAST 是否一定比不设定（NONE）要节省 LC 资源？

答：在布局布线过程中，Synthesis Style 的设置会影响到资源的利用率和速度的快慢，一般情况下：设置为 FAST 主要是为了提高设计的速度。在软件中除了综合类型的设置，还有一项是选择优化的目的：OPTIMIZE->AREA OR SPEED. 选择 AREA 可以节省设计所占用的资源。

24. Altera 公司对芯片热设计有哪些资料 and 工具？

答：ALTERA 提供了许多计算功耗的资料和工具。数据手册中的 AN74 就是关于计算 ALTERA 器件功耗的专门文档。

25. 如何在设计前期分析芯片的功耗？

计算功耗的工具：ALTERA 提供的 QUARTUS 软件就有计算功耗的功能，它可以根据你不同的激励项量来计算功耗；在 ALTERA 的网网上就有专门计算功耗的运算器，请点击相关文章 它就给出了 APEXII 的功耗计算方法。

26. 看过“FLEX PCI Development Board”的参考设计原理图，它利用了条线开关选择配置方式。既然两种配置方式管脚并没有公用，为何需要这个跳线开关呢？

答：EPC2 不会影响到用下载电缆通过 JTAG 口配置 EPF10K30A。使用跳线开关是在选择给 10K30A 下载的方式，是通过下载电缆还是 EPC2。

因为将 EPC2 与 10K30A 连接成 JTAG 菊花链的形式通过下载电缆下载和用 EPC2 对 10K30A 下载的连接方式是不同的，所以要将这两者分开，详细的连接方法应该在参考设计原理图中已经描绘的很清楚了。

27. 为了保证设计可靠性，需要重点关注哪些方面？

答：Here are a few guidelines for reliable FPGA design（关于可靠性 FPGA 设计的几点建议）

- a) Use fully synchronous design. Asynchronous design is very sensitive to path delay and is therefore not robust. An example of asynchronous circuit is the SR latch which uses combinational feedback.（使用完全同步设计。异步设计对路径延迟非常敏感，因此不很可靠。异步电路的一个例子是使用组合反馈的 SR 闭锁。）
- b) Never gate your clock signal with combinational logic. Glitches may occur on any gated clock signals, which results in false triggering of flip-flops.（绝不要使用组合逻辑控制时钟信号。因为在任何门控制时钟信号上可能产生短时脉冲干扰，最终导致错误触发 flip-flop.）
- c) Never rely on gate delay.（绝不要依靠门延迟。）
- d) Enough bypass capacitors should be placed close to the power and ground pins of FPGA. Use capacitors with good high frequency response.（FPGA 的电源和接地引脚附近应该放置足够多的旁路电容器。使用优质高频响应电容器。）
- e) Always use the global clock buffers on the FPGA to drive internal clock signals. These clock buffers and the associated clock distribution network have been carefully designed to minimize skew.（在 FPGA 上始终使用全局时钟缓冲来驱动内部时钟信号。并且

已经仔细设计了这些时钟缓冲和关联时钟配电网，以将畸变减至最小。）

28. You said "An example of asynchronous circuit is the SR latch which uses combinational feedback". How do I learn SR latch ? What is the difference between SR latch and SRFF? (“异步电路的一个例子是使用组合反馈的 SR 闭锁”。请问如何理解 SR 闭锁？SR 闭锁与 SRFF 有什么区别？）

答：A latch changes states whenever the gate signal is active. A FF changes states only at clock edges. (参考译文：只要门信号是活动的，闭锁就会更改状态。而 FF 只有在时钟边沿才更改状态。)

29. Xilinx 公司的芯片在热设计方面可以提供哪些工具和资料？

答：For thermal consideration, you need to know the power consumption of your FPGA and the thermal resistivity of the device package you are using. The power consumption can be estimated by the XPower tool included in Xilinx ISE software. The thermal resistivity of the device package can be found in Xilinx databook. The junction temperature can then be calculated by the following formula (至于散热问题，需要了解所用 FPGA 的功耗和正在使用的器件封装的热阻系数。Xilinx ISE 软件中所包含的 Xpower 工具可以估计功耗。在 Xilinx 数据手册中可以找到器件封装的热阻系数。然后利用下面的公式计算接合温度。)

$P = (T_j - T_a) / \theta_{JA}$ 其中, P=功耗; T_j =接合温度; T_a =环境温度; θ_{JA} = 封装的热阻系数

You can then determine if the junction temperature falls within the acceptable region. The maximum acceptable junction temperature is different for C and I grade devices. If it is higher than the max. acceptable temperature, you may consider adding a heatsink or cooling fan. (之后确定接合温度是否在允许范围内。C 和 I 级器件允许的最高接合温度不同。如果温度高于允许的最高值，可能需要增加散热片或风扇。)

30. 如果时钟进入 FPGA 后经过一段组合逻辑才上时钟网络，会存在一定的延时。综合布线后会出现信号输入延时为负值，意味着信号比时钟先到达触发器。那么，怎样通过约束文件增加输入信号的延时呢？我试过对"NET"加上"MEDELAY"的约束，但是没效果。

答：Gating the clock signal with combinational logic is not recommended in modern high speed digital design since it may create glitches on the gated clock signal, which results in false triggering of flip-flops. This results in a less reliable design. A common technique to remove gated clock is to make use of the clock enable pin of the flip-flop. For example, if you have a signal $clk_o = clk_i \& a \& b$ driving the clock pin of a flip-flop, you can eliminate the gated clock by feeding clk_i directly to the FF clock pin, and have another signal $en = a \& b$ connected to the clock enable pin of the FF. (参考译文：在现代高速数字设计当中，不建议使用组合逻辑门控时钟信号，因为这将会在选通的时钟信号上产生短时脉冲波形干扰，导致错误触发 flip-flop。这是缺乏可靠性的设计。移除门控时钟通常所采用的技巧是使用 flip-flop 的时钟使能引脚。例如，如果有一个信号 $clk_o = clk_i \& a \& b$ 正在驱动 flip-flop 的时钟引脚，则可以通过直接将 clk_i 传递给 FF 时钟引脚，并将另一个信号 $en = a \& b$ 与 FF 的时钟使能引脚连接，来消除门控时钟。)

By removing the gated clock, you no longer have the problem of clock delay. Also the design is more robust. (移除门控时钟后，就不再有有时钟延迟的问题了。而且这种设计也比

较可靠.)

31. FPGA 生产厂商提供了 IP, 如何用第三方软件, 如 Advantage 或 ACTIVE vhd, 调用并进行仿真?

答: The IPs provided by Xilinx, e. g. PCI, come with simulation models which can be processed by 3rd part simulation tools like Modelsim. So there is no problem for functional simulation. Timing simulation can be done by exporting the post-layout vhd/verilog model from Xilinx ISE software. In some cases, sample testbenches are also included. (参考译文: Xilinx 提供的 IP, 例如 PCI, 是与仿真模型一同提供的, 这种模型可由第三方仿真工具, 如 Modelsim 来处理. 因此对功能仿真来说, 没有问题. 通过从 Xilinx ISE 软件中导出 post-layout vhd/verilog 可以执行定时仿真. 在某些情况下, 也包括样本测试平台.)

32. “As a good design practice, never use gate delay to implement your delay logic under all circumstances”. Please tell me what does "gate delay"(闸) mean? (“一个好的设计, 在所有情况下都决不会使用门延迟来实现延迟逻辑.” 此处“门延迟”是什么意思?)

答: “By "gate delay" I mean using a series of logic gates to introduce certain amount of delay in the design. This is highly undesirable since gate delay changes with factors like temperature and process technology. The design may fail as temperature changes or using a different version silicon. Also designs relying on gate delay are not portable, meaning that you need to re-design the whole circuit whenever you want to change to another product series or part number, simply because the gate delay changes as well. (参考译文: “门延迟”指得是使用一系列逻辑门将一定数量的延迟导入到设计中. 既然门延迟更改像温度和处理技术这样的因素, 所以, 这是很不合适的. 由于温度的改变或使用不同版本的芯片, 设计可能会失败. 依赖门延迟的设计也不是可移植的, 也就是说, 要更改另一产品系列或部件号时, 需要重新设计整个电路, 只因为更改了门延迟.)

Always use fully synchronous design. You never need to reply on gate delay if your design is fully synchronous. (始终使用完全同步设计. 如果设计是全同步的, 则无需回应门延迟.)

33. This time I download another program to another chip SpartanII XC2S50PQ208 in another circuit, while it fails, and show the following message: ". . . Checking boundary-scan chain integrity. . . ERROR:JTag - Boundary-scan chain test failed at bit position '3' on instance '****'(a substitute for the real name of file). A problem may exist in the hardware configuration. Check that the cable, scan chain, and power connections are intact, that the specified scan chain configuration matches the actual hardware, and that the power supply is adequate and delivering the correct voltage. ERROR:JTag - Boundary scan chain has been improperly specified. Please check your configuration and re-enter the boundary-scan chain information. Boundary-scan chain validated unsuccessfully. ERROR:JTag - : The boundary-scan chain has not been declared correctly. Verify the syntax and correctness of the device BSDL files, correct the files, reset the cable and retry this command." With so many messages, I don't know what to do! I try many times but only fail, and doubt whether there is something wrong with the circuit? But the powers checked in circuit are right. Would you please give me some advice to crack the problem? (有一次, 将程序下载至 SpartanII XC2S50PQ208 芯片电路, 结果发生了故障, 并显示以下消息: "... Checking boundary-scan chain integrity. . . ERROR:JTag - Boundary-scan chain test failed at

bit position '3' on instance '***'(实际的文件名)”。问题可能在硬件配置。检查了连线、扫描链路和电源接头都没有问题。特定的扫描链路配置与实际的硬件相匹配，电源充足且电压正常。“ERROR:JTag - Boundary scan chain has been improperly specified. Please check your configuration and re-enter the boundary-scan chain information. Boundary-scan chain validated unsuccessfully. ERROR:JTag - : The boundary-scan chain has not been declared correctly. Verify the syntax and correctness of the device BSDL files, correct the files, reset the cable and retry this command.”这么多出错消息，什么原因，怎么办？）

答: Usually it is the result of a broken JTAG chain or noisy chain. Most commonly, the cable is not connected properly, a trace is not correct on the board, other devices in the chain are causing a problem, or a noisy parallel port exists. Try using a different PC. You can also add a 4.7K pullup on the PROG pin of the FPGA and see if it helps. (通常，这是由于中断的 JTAG 链或噪声链。最常见的原因是，连线不正确，板子上的迹线不正确，链路中的其它器件导致问题，或者存在噪声并口。试一下使用不同的 PC。也可以在 FPGA 的 PROG 引脚上增加一个 4.7K 的上拉电阻，看看是否有帮助。)

34. 在 VHDL 中，定义为 SIGNAL 的量起到什么作用？什么时候需要定义这个量？下面的程序

```
ARCHITECTURE EXER2_ARCH OF EXERCISE2 IS
```

```
SIGNAL TEM: STD_LOGIC;
```

```
BEGIN
```

```
TEM<=PIN50 AND PIN51;
```

```
PIN8 <=TEM;
```

```
END EXER2_ARCH;
```

和如下的程序有何区别？

```
ARCHITECTURE EXER2_ARCH OF EXERCISE2 IS
```

```
BEGIN
```

```
PIN8<=PIN50 AND PIN51;
```

```
END EXER2_ARCH;
```

答: If PIN8 is declared in your port list, the 2 examples are identical. From a hardware design's perspective, you can think of a vhdl "signal" as an electrical signal. So basically you can declare every object as "signal". >From a simulation's perspective, there is a fundamental difference between "signal" and "variable" in vhdl. A variable is nothing more than an object that holds a value. A variable assignment occurs instantly in a vhdl simulation. Also, a variable can only exist within a process, so it cannot transfer values across processes. A signal, on the other hand, has a history of values. Whenever a signal assignment occurs, the vhdl simulator schedules an event to update the signal value after a certain period of simulation time - the signal does not get this new value instantly in the simulation time domain. Also, a signal can exist outside processes. Sounds complicated, but for most of the time you can simply use vhdl "signal" in your hardware design. (参考译文：如果在端口表中声明了 PIN8，这两个示例是一样的。从硬件设计的角度看，可以将 vhdl "signal" 视为电子信号。因此，基本上可以将每个对象声明为“signal”。从仿真角度看，vhdl 中的 "signal" 与 "variable" 是根本不同的。变量只不过是拥有值的对象。变量分配即时出现在 vhdl 仿真中。而且，变量只能存在于一个过程内，因此它不能通过过程来传递值。另一方面，信号有多个值。不论何时分配信号，vhdl 仿真都会在某个仿真时段安排一个事件来更新信号的值。在仿真时域里，信号不会立即获得这

个新的值. 而且信号可以存在于过程之外. 听起来好象有点复杂, 但大多数时候, 在硬件设计中可以只使用 `vhdl` "信号".)

35. 如果输入时钟必须经过一段组合逻辑(比如需要进行时钟选择, 可选外部或内部时钟), 那么在 DFF 使能端加控制是无法解决的, 有什么更好的方法?

答: A simple answer is to use the BUFGMUX resource in Xilinx VirtexII devices. The BUFGMUX is actually a global clock buffer in VirtexII which incorporates a smart mux to switch between 2 clock sources. More importantly, the BUFGMUX guarantees glitch-free switching between these 2 clocks, even though the select signal changes asynchronously. (参考译文: 简单的方法是使用 Xilinx VirtexII 器件上的 BUFGMUX 资源. BUFGMUX 实际上是 VirtexII 中的全局时钟缓冲, VirtexII 将智能 mux 与 2 个时钟源之间的切换相结合. 更为重要的是, 即使选择信号更改不同步, BUFGMUX 也能保证这两个时钟之间的无干扰切换.)

36. 用 Altera 器件设计一个电路, 外挂一同步存储器件. 逻辑设计和存储器件的时钟是相同的, 但由于时钟信号带负载能力较差, 只能接一个负载, 所以将时钟信号接在可编程器件上, 而用内部赋值语句将时钟信号赋值给某一引脚, 此引脚信号再接到存储器件的时钟引脚, 但这样存在一个问题: 存储器件的读写信号相对于 Altera 器件上的时钟信号有延迟, 存储器件的时钟信号相对于 Altera 器件上的时钟信号也有延迟, 这样存储器件的时钟信号与读写信号之间的时间差难以控制, 极有可能不满足存储器件的 Setup/hold 时间, 有什么好的方法解决此问题?

答: 可以利用 ALTERA 器件中的 PLL 来精确控制延时. ALTERA 中的 PLL 是一个真正的模拟锁相环, 它可以提供精确的时钟频率合成与相位延时的细微调整. 可以将时钟信号按照你的要求进行延迟调整.

假如采用的器件没有 PLL 的话, 那么可能需要在逻辑内部来做一些延时逻辑, 不过这样会导致异步逻辑设计, 我们一般情况下不推荐这样用. 还有一种办法就是在 PCB 板上调整时延. 关于 PLL 的详细资料可以参考 ALTERA 的相关文档.

37. 利用 ACEX1K 系列片内 EAB 单元创建 RAM 时, 每创建一个容量较小的 RAM 时就要占用一个 EAB 单元, 能否将多个 RAM 整合由一个 EAB 实现, 以节约片内的 EAB 单元?

答: 在 ACEX1K 系列的器件中, 一个 EAB 单元目前只能做一个应用, 我们现在的软件还不支持将多个小的 RAM 集成到一个 EAB 当中.

38. Synthesis Style 设为 FAST 后, 发现速度有所提升, 同时使用的资源也减少了, 资源和速度似乎兼得了, 那么是否所有的模块都可以设定为 FAST 呢?

答: 将 SYNTHESIS STYLE 设置为 FAST 主要是为了提高系统性能. 但是有一点要记住的是, 软件的设置不是在任何情况下对所有的设计都表现出相同的结果. 针对这个设计模块, 将 SYNTHESIS STYLE 设置为 FAST 可能对资源和速度都有了优化, 但这并不说明对所有的模块都有相同的效果, 但是可以试一试. 设计优化是一个原则与经验, 技巧相结合的过程, 我们只掌握一定的原则与方法, 根据我们自己的经验, 运用一定的技巧, 才能将一个设计做到最优化.

39. I am now going to design a gray coded 16 bits counter , any efficient way to implement it in VHDL description? (要设计一个灰色编码 16 位的计数器, 怎么以 VHDL 描述来实

现它？)

答: You can just use megawizard(lpm_counter) in the software to generate the counter for you, select the output language with VHDL. That should be the efficient way to implement a counter, and you don't to make additional optimize.

Because the lpm function code is the best way to fit the structure of device, which is designed by altera factory specialist who is very familiar with our device structure. (参考译文: 可以使用软件里的 megawizard(lpm_counter)来生成计数器, 选择带有 VHDL 的输出语言. 这应该是实现计数器的有效方法, 而且不用进行额外的优化.

因为 lpm 功能代码最适合此设计结构, 这种结构是非常熟悉我们的设计结构的 altera 专家设计的.)

40. 一般情况下用 Synplify Pro 综合后生成的 edf 文件经 MP2 编译后与用 MP2 综合及编译相比较, 占用资源较少, 但在使用层次化设计中, 使用 Synplify Pro 综合顶层文件后得到的 edf 文件经 MP2 编译后与用 MP2 综合及编译该顶层文件相比较却大大的占用资源, 请问, 在使用 Synplify Pro 综合层次化设计中如何才能节省资源?

答: 在使用软件做优化时存在这样一个平衡关系: 资源利用率与速度的平衡. 资源利用率提高了, 也就是节省了资源, 但整个设计的性能可能会降低了.

同样尽力去优化系统性能, 提高速度, 那资源的利用也可能会增加. 当在使用层次化设计中, 如何来优化整体设计, 最关键的就是层次与模块的划分.

在划分层次和模块是有几点建议:

1)以功能来划分;功能模块与层次的设计可以帮助你清楚的定义边界, 在模块框图中, Data paths、tri-state signals、state machines、register blocks、large macrofunctions、memory elements、control blocks 和一些重复使用的模块都具有其本身的自然边界.

2)划分模块时要尽量减少模块间的 IO 连接, 过多的 IO 接口会使系统变得复杂, 软件需要交叉布线, 降低性能和提高资源利用.

3)在可能的情况下, 尽可能多的给模块的输出加 Resister. 尽可能地优化模块的划分与接口, 是提高层次化设计性能的关键. 更详细的方法与建议, 可以参照我们的应用文档 AN101.

41. 以前的问题提到, 用 EPM7064LC68 进行编译, 会出现编译错误. 如果编译时, 让系统自动选择器件, 则选中的器件是 EPM7064SLC84, 编译通过. 我查遍手头的资料, 并没有发现后者有三个输出使能, 这是怎么回事? 望解答.

答: 事实上在 MAX7000S 上有 6 个输出使能控制信号, 你可以在 MAX7000 的数据手册 (M7000. PDF)第二页看到有这样的说明.

Enhanced features available in MAX 7000E and MAX 7000S devices – Six pin- or logic-driven output enable signals

42. 在 FPGA 中是以何种形式实现 VHDL 的变量类型的?

答: There is no definite answer to this. It depends on how you write your codes. A variable in vhdl may be synthesized into a physical net, or it may not exist at all in the resulting circuit. 文: 没有明确的答案. 它取决于所编写的代码. Vhdl 中的变量可能同步到物理网络中, 或者根本不可能存在于结果电路中.)

43. 在布线后生成的时序报告文件中, 可以看到延时的一些报告, 对于某条时序报告,

如何定位其对应的语句呢？特别是当完全使用语言方式输入时，生成的网表中大量的为 N**形式，无法看懂其含义。

答：Most synthesizers do preserve signal names to a certain extent, usually a string is concatenated to the end of the original name. So you can still correlate the names in many cases. For those strange net names like N***, they are signals generated by the synthesizer and may not have a counterpart in the original source code. （参考译文：大多数合成器是会以某种扩展名来保存信号名称，这些扩展名通常是连接到最初的名称末尾的字符串，使这些名称在很多情况下仍然相关。至于那些像 N***一样奇怪的网表名称，是由合成器生成的信号，而且可能不会在最初的源代码中有副本。）

44. 布线后时序仿真与实际电路板上测试一般都不一样，特别对于高速信号，几个 ns 的差别是很大的，到底应该以哪一个为标准进行设计呢？

答：The timing information you get from the post-layout simulation is based on worst case parameter. So you usually have better results on silicon than in simulation. For robust designs, always consider the worst case. （参考译文：从时序仿真中获得的时序信息是基于最坏情况参数的。因此，通常在硅片上实际操作的结果要比仿真中的好。对于可靠的设计而言，始终都要考虑最坏的情况。）

45. ISE4.2 和 ISE4.1 相比有何改进？

答：Here's a brief list of new features in ISE4.2i （以下是 ISE4.2i 新特性的要点）

- Device support for VirtexII Pro and CoolRunnerII （设备支持 VirtexII Pro 和 CoolRunnerII）
- Provides 2 new source types, BMM files and ELF files, for embedded VirtexII Pro PowerPC and Microblaze processor support. BMM file is the Block RAM Memory Map file that describes the organization of Block RAM memory. ELF file is the Executable and Linkable Format file contains the executable CPU code image to be stored in Block RAM as specified in the BMM file. （提供 2 个新的源类型：BMM 文件和 ELF 文件，以支持嵌入式 VirtexII Pro PowerPC 和 Microblaze 处理器。BMM 文件是“块 RAM 内存图”文件，它描述了块 RAM 内存的结构。ELF 文件是“可执行和可链接格式”文件，它包含存储在 BMM 文件中指定的块 RAM 的可执行 CPU 代码图。）
- Improved PAD file for easier to import into a spreadsheet program for viewing, sorting and printing. （改进 PAD 文件，以便导入到电子表格程序中，供查看、存储和打印）
- iMPACT now incorporates the functionality of the PROM File Formatter and Xilinx System ACE software. （iMPACT 与 PROM 文件格式程序和 Xilinx 系统 ACE 软件的功能相结合）
- XST enhancement for better language support and preservation of internal signal names. （XST 增强了语言支持，并能保存内部信号名称。）

For more information regarding Xilinx ISE4.2i, please visit our website www.xilinx.com （更多有关 Xilinx ISE4.2i 的信息，请访问网站 www.xilinx.com）。

46. 经常看到 gate 这个词。能够具体解释一下它的含义，例举其用法以及如何避免问题？

答：Here're a couple of examples : （举例说明）

- Never use gated clock. By gated clock we mean the clock signal comes out from combinational logic. It is well known that any signal coming out of combinational logic is prone to glitch. The result is fatal if there is a glitch on your clock signal since it will cause false triggering of FFs.

A common technique to avoid gated clock is to utilize the clock enable pin on the FF. (从不使用 gated clock. 这个词表示时钟信号出自组合逻辑. 众所周知, 任何出自组合逻辑的信号都容易发生故障. 由于时钟信号上的故障将导致错误触发 FF, 其结果是致命的. 避免 gated clock 常用的技巧是利用 FF 上的时钟使能引脚.)

- Never design a circuit that relies on gate delay to function. It was a common practise in the past to introduce a delay in the design by inserting a series of logic gates. This is not a recommended style in modern high speed digital design since the delay changes as new devices coming out from more advance process technologies. Also, the amount of delay changes as temperature and voltage as well. So it is not a good design practice to have circuits which relies in gate delay to function. (绝不设计依赖 gate delay 工作的电路. 通过插入一系列逻辑门在设计中引入延迟, 这是以前常见的作法. 而在现代高速数字设计中, 建议不要使用这种作法, 因为延迟会随采用更先进的工艺技术所制造的新器件而改变. 而且, 延迟的总量也会随温度和电压而改变. 因此依赖 gate delay 而工作的电路不是很好的设计.)

47. 用 FLEX6016 设计了一个频率测试卡, 用的是 ISA 总线和计算机相连, 不把卡插在 ISA 槽上时, 由外部提供电源时, 下载就能成功, 一但插上去, 下载就出现 “SRAM load unsuccessful”, 这是怎么回事呢?

答: 导致 “SRAM load unsuccessful” 可能有各方面的原因. 基于板子由外部供电是可以下载成功, 说明下载电路是正确的; 而插入 ISA 槽中则出现问题, 可能是 ISA 槽供电有问题, 可以检查一下芯片的电源信号, ISA 与外部供电是否采用一致的电路, 检查 ISA 供电环境, 是否存在毛刺, 电源纹波的大小.

48. 想把 EPM712 和 TMS320F240 做在一块实验板上, 但是不知道怎么设计仿真口对 EPM7128 编程? 还是必须买厂家的 EPM7128 开发板?

答: 可以参考 AN116 的下载数据手册, 按照其中的下载原理图来连接 DSP 与 PLD 的下载口. 下载电路其实非常的简单, 只需按照 7128 的下载波形, 从 DSP 中将 PLD 的下载文件依此输入即可. 当然也可以从当地代理商那里获得支持, 参考一些典型下载电路.

49. 想自己设计一块 TMS320F240 试验电路板, 包括 A/D、D/A、键盘显示接口, 其中译码、键盘显示部分想用 EPM7128 来做, 不知道具体怎么与 TMS320F240 接口?

答: 所谓 PLD 为可编程器件, 其 IO 口的连接非常地灵活, IO 口的功能可以按照自己的定义来设定. 只需将希望的 TMS320F240 连接口连接到 PLD 的任意 IO 口上, 然后在 PLD 中编写相应的控制逻辑, 即可进行数据传输与控制.

50. 变频器盘中使用一芯片是 ALTERA EP330PC-12 烧坏, 请问如何处理?

答: 这是一款旧型号的芯片. 可以利用 ALTERA 或第三方提供的编成器将下载文件重新下载到一个好的器件中去, 或者采用 MAXPLUSII 软件通过 BYTEBLAST (MV) 连接到板子上将编成文件读出再下载到新的器件中去.

51. SRFF = SR flipflop SRFF 和 SR latch 有何区别?

答: FUNCTION SRFF (S, R, CLK, CLRN, PRN) RETURNS (Q);

//VHDL Component Declaration:

COMPONENT SRFF

```

PORT (s : IN STD_LOGIC;
r : IN STD_LOGIC;
clk : IN STD_LOGIC;
clm: IN STD_LOGIC;
prn : IN STD_LOGIC;
q : OUT STD_LOGIC);
END COMPONENT;
FUNCTION LATCH (D, ENA)
RETURNS (Q);
//VHDL Component Declaration:
COMPONENT LATCH
PORT (d : IN STD_LOGIC;
ena: IN STD_LOGIC;
q : OUT STD_LOGIC);
END COMPONENT;

```

不同点在于 SRFF 是一个触发器，而 LATCH 只是一个锁存器，更详细的真值表可以从软件的 HELP 文档中可以查到。

52. 想在内部上拉输入信号，所使用的设备是 FLEX6016. 怎么做？

答：可以在 MAXPLUSII 中选定该信号，然后选择 assign-> logic option->Individual logic options -> Enable pull-up resistor. 然后重新编译一下就可以了。

53. 有关输入信号的上拉问题（前题），按照专家的回答做过，但是失败了，不知道是什么原因导致了此法不可行？还有没有别的办法？

答：Altera 的 FLEX6000 系列在 I/O 管脚上是没有上下拉电阻的，所以加了约束也没有作用。

54. 使用 AHDL 语言编写的程序，在 Quartus II 1.0 下编译，使用的是 20K400EBC652-3 的片子。将编译产生的 pof 文件下载到 EPROM 里，但是在程序没有多大修改的情况下（仅仅改变一些测试管脚），程序运行结果不一样。具体表现在 DSP 芯片启动 FPGA 里的一根控制线不稳。

答：逻辑功能仿真结果如何？在修改前后有没有改变？假如说功能仿真是对的，请确认设计 Timing 是否满足要求，尤其是 IO 的 Timing 要求是否达到。在可能的情况下进行后仿真，其仿真结果能够确保你的逻辑在 PCB 板上正常地工作。假如仿真结果与 Timing 要求都没有问题，其逻辑一定能在板子上正常地工作。

55. 当一个输入信号不满足触发器的 Setup/Hold 时间时，触发器的输出信号是不是一稳定状态（或为 0，或为 1，当下一轮的输入信号满足 Setup/Hold 时间时，触发器能正确地输出）？由于此时触发器处于亚稳态，以前看过一些资料，某些器件的输出可能是振荡状态，即此时、将来的输出信号不可预测，与时钟信号、输入信号无关。我想问的是 Altera 器件对此情况是如何处理？因为某些情况下，当输入信号超过 1 个 Clk 时间，只是在第一个 Clk 周期内，不满足 Setup/Hold，但是其他的 Clk 周期内，满足 Setup/Hold。

答：关于这个问题，建议参考一下 ALTERA 的文档 AN42。该文档详细地讨论了 ALTERA 器件的亚稳态性。网上的地址是 <http://www.altera.com/literature/an/an042.pdf>。

56. 在中国市场上, 可以容易买到使用 Altera 公司的软件 MAX+PlusII 进行 VHDL 和 FPGA 设计的教程书籍, 但是却鲜有使用 Xilinx foundation 软件平台的书籍, Xilinx 是否考虑增强这方面内容?

答: Thank you for your input. In fact there are a number of books available in the market on Xilinx FPGA and development tools. A good example is the title "XILINX 数字系统集成技术" by Professor 朱明程, published by Southeast University Press. We will work closely with local publishers to bring out more titles on Xilinx products. (市场上还是有几本 Xilinx FPGA 和开发工具的书. 比较好的有朱明程教授编的《XILINX 数字系统集成技术》, 东南大学出版社出版. Xilinx 公司也将会与本地出版商密切合作, 推出更多针对 Xilinx 产品的书籍.)

57. 在 ISE4.1 环境下编写一个包结构, 里面有几个函数, 编译通过, 而 MODELSIM 仿真出错, 提示: 没有找到此 PACKAGE, 为什么?

答: After you have created the package, you need to add it to your project. In the source window, right click and select "Add source", pick the source file for your vhdl package, and then select "Vhdl package". The package will then be added to your ISE project. (参考译文: 创建了这个包之后, 需要将它添加到项目中. 在源代码窗口, 右键单击并选择 "Add source", 为 vhdl 包选择源文件, 然后选 "Vhdl package". 这个包就添加到 ISE 项目中了.)

58. 布板时, 时钟信号没有接全局时钟, 如何处理时钟使之可以更好 (不需飞线)?

答: You can bring the signal back to the global clock network by inserting a BUFG. (参考译文: 可以通过插入一个 BUFG 将一个信号返回全局时钟网络.)

59. 准备使用 XCV50 FPGA, 前面有 16 个模块, 本来每个模块都须要一个 27M 的时钟, 为了达到同步, 初步设想外部接一个 27M 时钟, 最后由 FPGA 产生 16 个 27M 的时钟输出, 这样做驱动会不会有问题?

答: A possible way is to feed your 27MHz input clock to a DLL. The output of the DLL drives multiple OBUF. In this way you can create multiple copies of your input clock to drive other components on your board. You don't actually need 16 OBUFs since each OBUF can drive several loads. Note that although the DLL output can drive multiple OBUF, only the one which provides feedback to the DLL can be completely deskewed. You can also add a MAXSKEW constraint on the output net of the DLL to minimize skew among the OBUFs. This may not be a big issue since 27MHz is relatively slow. (参考译文: 可能的方法是将 27MHz 输入时钟传递给 DLL. DLL 输出驱动多重 OBUF. 用这种方法可以创建输入时钟的多个副本以驱动板子上的其它组件. 实际上, 不需要 16 个 OBUF, 因为每个 OBUF 都可以驱动几个负载. 请注意, 虽然 DLL 输出能驱动多个 OBUF, 但只有那个向 DLL 提供反馈的 OBUF 可以完全对称(deskewed)的. 也可以在 DLL 的输出量上添加 MAXSKEW 限制, 以最小化 OBUF 间的偏斜. 因为 27MHz 相对比较慢, 所以问题不大.)

For more information regarding the use of DLL, please refer to the application note XAPP132 which can be downloaded from the xilinx website www.xilinx.com (关于 DLL 的使用, 请参考 XAPP132 的应用手册.)

60. IP CAPTION 就是核发生器, 它运行在核发生器目录下吗?

答: I think you mean "IP Capture". The IP Capture tool provides designers with an automated method to identify, capture, and document a core. The core can exist in the form of

synthesizable VHDL or Verilog code, or a fixed function netlist. Once the new module has been captured, it can be installed into and distributed from a user's local copy of the Xilinx CORE Generator system. (参考译文: 所指的应该是"IP Capture"吧. IP Capture 工具为设计者提供了自动识别、捕获和归档核的方法. 核可以综合 VHDL 或 Verilog 码, 或固定功能连接表的形式存在. 一旦捕获了新模块, 就能够将核装入, 并从用户的本机 Xilinx CORE Generator 副本中分配此核.)

Details of the IP Capture tools can be found from Xilinx website . http://www.xilinx.com/page_moved/ipcenter_e.htm?url=/ipcenter/designreuse/ipic.htm (有关IP Capture工具的详细资料, 请访问相关网站.)

61. 制作了一块试验板, CPLD 使用 EPF10K20TI144-4. 用 max+plus II 10.1 编程, 完成后, 下载至片子内, 经过几次下载尝试 (一直提示 configuration failure: SRAM load unsuccessful), 直至提示 "configuration complete" ——这应该是表示下载成功吧, 可是片子却无法实现任何功能 (连最基本的一个或门都无法实现) ——不起任何作用, 我使用 TQFP-144 的适配座放置 EPF10K20TI144. 现在有几个问题:

- a) 编译前选择的是 EPF10K20TC144-3 芯片 (max+plus II 的器件库中没有 EPF10K20TI144-4), 有没有关系? 也用 EPF10K10TC144-4 试过, 也是可以下载但无法实现功能.
- b) 为什么要下载很多次才能成功, 下载线是自己制作的?
- c) 将 MSEL0、MSEL1、nCE、专用输入脚、专用时钟脚接地, 其它非 I/O 引脚悬空, 是否还有些引脚需要接地或 VCC?
- d) 如果因为 TQFP-144 适配座一些触点与 CPLD 引脚接触不好导致一些 VCCIO, VCCINT、GNDINT、GNDIO 或其它一些非 I/O 引脚悬空, 会出现什么结果?
- e) 如何才能使 CPLD 实现功能?

答: 根据描述, 很有可能是加载电路有问题, 请参照 Altera 的 AN116 文件的加载电路图仔细核对, 特别注意和 CPU 相连的加载控制管脚:

- ① 编译芯片选择与速度等级无关;
- ② 请检测 conf_done 信号是否拉高;
- ③ 在常用的 PS 和 JTAG 加载模式时除了 MSEL0, MSEL1, nCE, 没用的专用输入脚, 专用时钟脚应该接地, data0, nconfig, dclk, nstatus, conf_done 都应该上拉, 这些管脚不上拉很可能是导致你加载不成功的原因;
- ④ 如果因为 TQFP-144 适配座一些触点与 CPLD 引脚接触不好导致一些 VCCIO, VCCINT, GNDINT, GNDIO 或其它一些非 I/O 引脚悬空, 会出现会出现加载链路不通或加载不成功的现象;
- ⑤ 加载成功后 CPLD 就能实现功能.

62. 目前采用 SRAM 技术的 LUT-based 的 FPGA 仍然占绝对主流地位, 但是这种 FPGA 有其天生的缺陷, 请问下一代 FPGA 的构架发展趋势怎样? Altera 在这方面有没有什么新的尝试?

答: 下一代 FPGA 主要朝着 SOPC 的方向发展:

Altera 的下一代产品 Stratix, 基本单元 LE 仍然基于 LUT 结构, 但有些大的改变,

- ① 去掉了传统的级联链 (Cascade Chain);
- ② 进位链 (Carry Chain) 由以前的单一 1 条变为 LAB Carry-in、Carry-in1、Carry-in0 三条;
- ③ LE 的扇出也由 2 个变为了 4 个;

④ 触发器的控制信号也更多.

其它方面:

① RAM: 一改以前单一 2K BITS 或 4K BITS 的 Memory 为小 RAM: 512BITS; 中 RAM: 4KBITS; 大 RAM: 512 KBITS;

② 第一次在 FPGA 中嵌入 DSP;

③ 时钟链路也由全局时钟变为全局时钟和区域全局时钟;

④

所有的这些变化都是通过大量的用户反馈信息作出的慎重改进, 使得 Stratix 器件更加适合用户的设计需求

63. 能否提供 7128 的下载波形、DSP 与 PLD 的典型下载电路?

答: 7128 的下载波形与 PLD 的典型下载电路在 Altera 应用文档 AN95 里有详细的说明.

64. "Always use fully synchronous design. You never need to rely on gate delay if your design is fully synchronous." 如果设计是完全同步的, 意思是就可以不用依靠门延时. 但在设计时, 在对和数据一同进来的控制信号进行处理后所产生的输出控制信号会滞后数据数个时钟周期, 这时就不得不对数据进行门延时, 已达到同步. 现在看来, 这是不合适的. 但是, 要用什么办法来处理这个问题而不应用门延时呢?

答: In a "fully synchronous" design, you only have 3 kinds of paths (在完全同步设计中, 只有三种途径:)

a) From I/O to synchronous element. (从 I/O 到同步元素)

b) From synchronous element to synchronous element. (从同步元素到同步元素).

c) From synchronous element to I/O. (从同步元素到 I/O)

A fully synchronous design will work as long as the delays of all these paths do not exceed the clock period. So you never need to introduce any delay. If you derive any control signal from the input signals, that control signal only goes to the 'D' or 'EN' pin of a FF, but not the 'CLK' pin. So there is no need to introduce any delay to your data. Simply speaking, you only need to design the "data" path but not the "clock" path in a fully synchronous design.

On the other hand, you can remove clock delay with the DLL in Xilinx SpartanII/Virtex devices or the DCM in VirtexII devices. (完全同步设计将工作到所有这些路径的延迟不超过时钟期间. 因此从不需要传入任何延迟. 如果从输入信号中得到任何控制信号, 则该控制信号只能传递至 FF 的 D 或 EN 引脚, 而不是 CLK 引脚. 因此, 不需要将任何延迟传入数据. 简单讲, 在完全同步设计中, 只需设计“数据”路径, 而不是“时钟”路径.)

65. ISE4. 2I 可对某些特殊管脚设置内部上拉, 那么外部是不是就可以不必再加上拉电阻了? 另外, 配置过程中, 所有的用户 I/O 是否均为三态? VIRTEX2 的 HSWAP_EN 应接何电平? 为什么?

答: Whether you need to add external pullup resistors depends on your board design. Usually external pullup resistors are required under the following conditions ((参考译文: 是否需要添加外部上拉电阻取决于电路板设计. 通常在以下条件下需要外部上拉电阻:)

- You need precise pullup values. The pullup built into the IOB is not specified to a precise resistive value since it varies from batch to batch. (需要精确的上拉值. 因为它随批次而变化, 所以然内置入 IOB 的上拉未指定精确电阻值.)

- You need strong pull-up strength. The IOB pullup is relatively weak - in the order of several tens of KOhms. (需要强大的上拉动力. IOB 上拉相对较弱, 只有数十千欧)

If you only want to prevent the board signal from floating, external pullup is usually not required.

You need to refer to the datasheet of the corresponding FPGA series to determine the state of the I/O during configuration. As an example, I/O pins of Virtex-E and Spartan-IIIE behave as tri-state before configuration. The HSWAP_EN pin on Xilinx Virtex-II is for enabling/disabling pullups on the user I/O pins during configuration. By default, HSWAP_EN is tied high (internal pullup resistor) which turn off pullup resistors on user I/O during configuration. (如果只想阻止板信号浮动, 通常不需要外部上拉. 需要参考相应的 FPGA 系列的参数表, 以确定配置过程中 I/O 的状态. 例如, Virtex-E 和 Spartan-IIIE 的 I/O 引脚在配置以前是三态的. 配置过程中, Xilinx Virtex-II 上的 HSWAP_EN 引脚用于启用/禁用用户 I/O 引脚上的上拉. 缺省情况下, HSWAP_EN 系有较高内部上拉电阻, 用于在配置期间关闭用户 I/O 上的上拉电阻).

66. 自顶向下的设计过程是什么样的一种过程? 我的理解是在设计开始时, 利用 vhdl 或 verilog 中的延迟描述语言在功能和波形上进行仿真描述, 而后进行 rtl 设计... 是否是这样的一个过程?

答: By top down design we usually mean you partition the top level design into a number of modules first, without going into details of each of those modules. Then you start to design each module, which may involves defining more underlying submodules. The process stops when you have designed all the underlying submodules. The basic idea is to build a hierarchy which starts with less details on the top and more details as you go down the hierarchy. (参考译文: 我们所谓的自顶向下设计通常是指: 先将顶级设计划分为多个模块, 而不涉及这些模块的细节. 然后开始设计每个模块, 可能包括定义更多底层子模块. 当设计完成所有底层子模块后, 这个过程即完成. 基本思路是建立一个层次结构, 从顶部粗略划分开始, 越向下层越详细.)

67. HSWAP_EN 推荐接高还是低, 为什么?

答: There is no definite answer to this - it depends on whether you want the user I/Os to have pull up resistors during configurations. Therefore it is design dependent. (参考译文: 没有定论. 高还是低取决于配置过程中是否要使用者 I/O 具有上拉电阻. 因此因设计而定.)

68. 用 ISE 做过很多的设计, 但是 ISE 中的 help 对设计说的比较详细, 但是对 Navigator 的功能的说明不是很好. ISE 中有一个 Library_view. 当你把 work 库中的设计 move 到 VHDL 或 VERILOG 中, 这时就会在 library_view 的窗口中看到. 但是有什么用也没有呀. VHDL 里用 library ****; use *** 不行. verilghdl 里用`include 也不行. 还不如直接在 work 中的一个 package. 什么时候用到 library_view 中的功能?

答: The proper method to create and use library within ISE is (正确的方法是创建并使用 ISE 中的库, 过程如下:)

- a) Click Project->New Source and select VHDL library. (单击 Project->New Source 然后选择 VHDL 库)
- b) Enter a name for the new library and click Next, then Finish. The new library is now visible in the Library View tab. (为新的库输入名称, 然后单击 Next-> Finish. 新的库即出现在 Library View 标签上.)

- c) In the library view tab, right click on the new library and then click Add Sources.
(在库查看标签中, 右键单击新库, 然后单击 Add Sources)
- d) Choose the sources you want to add to the library. (选择要添加到库的源)
- e) You can then use this library in your source codes. For example (然后可以在源代码中使用这个库. 例如:)

```
library MyLib;
use MyLib. MyPkg. all;
.....
```

Please refer to p. 5-14 to 5-17 of the ISE User Guide which is included in the ISE Documentation CD for more details. (详情请参阅 ISE 文档说明光盘中的 ISE User Guide 5-14 至 5-17)

69. 利用器件的 IBIS 模型能否对器件的功能进行仿真?

答: IBIS models provide information about I/O driver and receiver characteristics without disclosing proprietary knowledge of the IC design (as unencrypted SPICE models do). You need to have an IBIS simulator in order to use the IBIS model. It is not for logic simulation like vhdl or verilog models do. (参考译文: IBIS 模型提供有关 I/O 驱动器和接收器特征的信息, 而不会透露 IC 设计的所有权信息, 这和未加密的 SPICE 模型一样. 为了使用 IBIS 模型, 需要有一个 IBIS 模拟器. 像 vhdl 或 verilog 模型一样, 它不是适用于逻辑仿真.)

70. 用 modelsim 后仿真时需要用到 XILINX 的库, 下载了 XILINX 网址提供的 TCL 文件, 按照要求在 AUTOEXEC. BAT 中设置了 SET MODELSIM=****, 重启了机器, 在 MODELSIM 下运行了 TCL 文件, 选项为 VERILOG, simprim, 可是 MODELSIM 还是报找不到库, 为什么? 所用 ISE 版本为 ISE4. 11.

答: The tcl script is for compiling the Xilinx libraries with Modelsim. After you've compile the libraries, you need to vmap them when you simulate your design. For example, if the physical path to your compiled simprim library is d:/xilinx_sim_model/simprim, you need to map simprim to the full path by typing "vmap simprim d:/xilinx_sim_model/simprim" at the Modelsim command prompt. Then you can simulate your design by the following command "vsim -L simprim " (参考译文件: tcl 脚本是为使用 Modelsim 编译 Xilinx 库的. 在编译完库之后, 仿真设计时需要 vmap 这些库. 例如, 如果指向已编译的 simprim 库的物理路径是 d:/xilinx_sim_model/simprim, 则需要通过在 Modelsim 命令提示处输入 "vmap simprim d:/xilinx_sim_model/simprim" 来将 simprim 映射到完整路径. 然后可以使用命令 "vsim -L simprim " 仿真设计.)

71. PIN HSWAP_EN 的作用是使能或关闭配置时 I/O 管脚上的上拉电阻, 这两种情况具体有什么不同呢? PCB 设计时 HSWAP_EN 具体如何处理?

答: The Virtex-II HSWAP-EN pin has an internal pull-up by default to turn off the pull-up resistors at the I/O pins during configuration. You need to tie it low externally if you need to activate the pull-up at I/O pins during configuration. (参考译文: 缺省情况下, Virtex-II HSWAP-EN 管脚有一个内部上拉电阻, 以在配置期间关闭 I/O 管脚上的上拉电阻. 如果在配置过程中需要激活 I/O 管脚的上拉电阻, 需要将这个上拉电阻在表面上约束到较低.)

72. PWRDWN_B 是 power down pin, 可不可以不连接或上拉?

答: The PWRDWN_B pin was provided to set a Virtex-II device into a power-down mode: a

low-power, inactive state. However, this feature is no longer supported in Virtex-II. To set it as inactive, the PWRDWN_B should be left at its default value, which is pulled up. (The PWRDWN_B does not require an external pull-up or pull-down.). Please refer to Xilinx website for more details. (参考译文: PWRDWN_B 管脚用于将 Virtex-II 器件设定为功率下降模式: 低功率和不活动状态. 然而 Virtex-II 中已不再支持这种功能. 要在非活动状态下设定它, PWRDWN_B 应该保留缺省值, 即已上拉. PWRDWN_B 不要求外部上拉或下拉. 详情讲参阅 Xilinx 网站) .

73. spartanII 不能下载, 显示 done 不能拉高, done 上上拉电阻已接, 是何原因?

答: Here are my suggestions (有几点建议:)

-Check if the mode pin is selected correctly. (检查是否正确选择了方式管脚) .

- Check if you have selected the correct startup clock. If you are using serial mode, you need to use Cclk. If you are using jtag, you have to use Jtagclk. This can be done by right clicking on Generate Programming Files -> Properties, then go to the Startup option tab, and set the startup clock properly. (检查是否已经选择了正确的启动时钟. 如果正在使用串联方式, 则需要使用 Cclk. 如果使用 jtag, 则必须使用 Jtagclk. 具体做法是右键单击 Generate Programming Files -> Properties, 然后转至“启动”选项卡, 并正确设定启动时钟.)

- If you are downloading from cable, the parallel port of you PC may be noisy. Change to another PC. (如果正从电缆下载, 则 PC 的并口可能会有嘈杂声. 更换到另一台 PC.)

- There may be noise or ringing on the Cclk signal. Try to insert a series termination of around 100 ohm, or a small capacitor to gnd, and see if it helps (Cclk 信号上可能会有噪音或响铃. 试着在 gnd 引脚上插入一个约 100 欧的串联端接或者小电容, 看看是否有用.)

74. 98 年使用某器件设计了一个电路, 用了四年, 共发出几千个芯片, 一直未出现问题. 后来由于器件更新换代, 将该设计原封不动移植到同型号器件改进型上, 发现工作不稳定, 将近 5% 的芯片不能用. 厂家设计工程师告诉我们, 设计没有任何问题, 信号模拟全部通过. 原因是新型器件速度提高, 造成毛刺增加引起的. 这种说法有无理论根据?

Altera 公司器件是否也会有类似问题? (注: 我们正在考虑将该设计移植到 Altera 相应器件上, 但担心问题仍在) .

答: 这个问题与前面讨论的同步设计问题有延续性, 我们一直都推荐客户采用同步设计的办法, 而不是异步设计, 不要采用一些门阵列来确定延时, 而应当采用同步时钟来触发逻辑运行. 因为在异步设计中, 用门阵列得到的延时是与器件的性能有很大的相关性, 当采用的器件更新后, 速度提高后, 每个门阵列的延时参数都发生改变了, 所以整个设计的结果就有可能与原来的不一样, 这对于任何 PLD 厂商来说都是一样的. 所以需要确定设计是否同步设计, 假如是同步设计那对器件的升级不会造成任何问题, 假如说不是同步设计, 更新器件时则要对不同的设计作出可能需要的相应修改.

75. TMS320F240 需要哪些信号线与 EPM7128 连接, 实现 4×4 键盘、8 位数码管显示?

设计 PCB 板时应该注意什么问题? EPM7128 是 5V 供电吗? 能否提供相应的典型电路?

答: Altera 器件的任何用户管脚都可以用于和 TMS320F240 相连; 实现 4×4 键盘、8 位数码管显示时要注意我们的管脚的典型驱动电流是 20MA, 如果超出, 需要加驱动电路; PCB 没有什么特别需要注意的地方; 不同的型号有不同的供电电压.

76. AN95 中 TCK 通过 1K 电阻接地, 而且 TDO 直接与目标芯片连接了. 但是 Altera5V

下载电缆不是这样连接的, TCK、TDO、TMS、TDI 都是通过上拉电阻接 VCC. 通过 ByteBlaster 下载电缆应该怎么连接?

答: 正确连接 TMS、TDI 通过上拉电阻接 VCC; TCK 通过电阻接地; TDO 直接与目标芯片连接.

77. 基于 maxplus2 的仿真是不是就是所谓的后仿真? 答: 基于 Max+PlusII 的仿真有两种: (1) 功能仿真; (2) 时序仿真 (后仿真).

78. 如何使用 modelsim 对 maxplus2 适配后的下载文件进行后仿真?

答: 在 Max+PlusII 中可以产生用于在 modelsim 仿真的文件.

79. VIRTEX2 的 DCI 特性的使用问题: IO 电平标准为 LVTTTL, 当该 BANK 的 IO 为输出时, 可通过 VRN、VRP 接匹配参考电阻来匹配该 BANK 的输出信号, 而当 IO 为输入时呢, 是否应在与之相接的驱动端加匹配电阻? 如果不加, VIRTEX2 能自行调整阻抗匹配吗?

答: For LVTTTL input, you should use the LVDCI input standard. LVDCI inputs do not require reference resistors on VRN and VRP pins. In fact there are no termination on these inputs. Therefore, termination at the driving end is recommended. Visit the Xilinx website for more details. (参考译文: 对于 LVTTTL 输入, 应该使用 LVDCI 输入标准. LVDCI 输入不需要在 VRN 和 VRP 引脚上接参考电阻. 事实上这些输入上也没有端接. 因此建议使用驱动端上的端接. 详情请访问 Xilinx 网站).

80. "Enable Pull-up Resistor"好像只对 MAX7000B 有效, 它对 Flex6k, Flex10k, Acex1k 也有效吗?

答: "Enable Pull-up Resistor"只对 MAX7000 和 3000 有效, 对 Flex6k, Flex10k, Acex1k 无效.

81. 有关 ACEX1K 的 PCB 注意事项有哪些?

答: 关于 PCB 的注意事项可以参考 Altera 的应用文档 AN75 中关于电源模块部分, 该文档中有详细的描述.

82. 批量生产时, 是否要一个一个对 ACEX1K 的 PCB 的配置器件如 EPC1 进行下载?

答: 在批量生产时, 可以利用 ALTERA 或第三方提供的下载器进行下载, 不需要对每一块板子单独地下载, 只需将下载好的 EPC 直接插入板上即可.

83. 如何将 FPGA 设计转为 ASIC 设计?

答: 可以将 FPGA 设计转为 ASIC 设计, 市面上有很多这样的 ASIC 设计厂商. 但是转为 ASIC 需要很大的量很一大笔开模费用, 而现在的 FPGA 器件已经非常便宜了.

84. 在用 MAX+plusII 设计一个产品时发现仿真结果不正确, 一个三分频的电路却变成 4 分频. 最后发现删除所有的 cnf 文件就行了. cnf 文件有什么用?能否自行删除?

答: The MAX+PLUS II software creates .cnf files to reduce compile times. With the .cnf files, the compiler does not have to re-extract the design files every time the software recompiles a design.

The MAX+PLUS II Database Builder can determine if a given entity, with a given set of

parameters and port usage, has already been extracted. If so, it saves time by just reading the .cnf file information from the disk.

If the .cnf files are deleted after each compile, the MAX+PLUS II software will call the extractors for every source file, which may cause an increase in the compile time.

The number of .cnf files generated depends on the design size and parameters. Once the .cnf files are created, they are not overwritten by the MAX+PLUS II software unless there is a design/parameter change. This could affect all the .cnf files initially generated by the MAX+PLUS II software. (参考译文: MAX+PLUS II 软件创建 .cnf 文件以缩短编译时间. 有了 .cnf 文件, 编译器就不必在每次软件重新编译设计时重新抽取设计文件了.)

MAX+PLUS II 数据库构建器可以确定某个带有一组指定参数和端口使用的给定实体是否已被抽取. 如果是, 那么只要从盘中读取 .cnf 文件信息就可以了, 因而节省了时间.

如果在每次编译结束后都删除 .cnf 文件, 则 MAX+PLUS II 软件将为每个源文件调用抽取程序, 这可能导致编译时间的增加.

生成的 .cnf 文件的数量取决于设计的大小和参数. 一旦创建了 .cnf 文件, MAX+PLUS II 软件就不会覆盖这些文件, 除非设计或参数发生改变. 这会影响所有最初由 MAX+PLUS II 软件生成的 .cnf 文件.)

85. EPM7128 与数码管、键盘接口用不用驱动?

答: EPM7128 与数码管、键盘接口可以不用驱动, 但要注意连接的个数. 具体的电流参数可以参照 7000 的数据手册.

86. MAX+PLUS II 9.6 中怎么不支持 EP1K10 器件? 它里面怎么只有 EP1K30, EP1K50, EP1K100? 如果我需要用 EP1K10, 该如何解决?

答: 可以从 ALTERA 网站上下载最新的 MAXPLUSII 软件, MP2 10.1 就支持 1K10. http://www.altera.com/products/software/pld/products/max/sof-emax_baseline.html

87. 1 个 EAB 等于多少个 LE? 手册中 LE 的数量 (如 EP1K10 有 576 个 LE) 有没有包括 EAB 折算的 LE 在内?

答: 每个不同的器件中的 EAB 完成不同的功能相当与 LE 的个数也不等, 可以从 12 个到 36 个不等. 1K10 中的 576 个 LE 没有包括 EAB 折算的 LE 在内, 并且所有 ALTERA 的器件中所指的 LE 个数都不包含 EAB 的折算.

88. 设计中用了 2 个 EAB 来实现 FIFO, 如果不用 EAB 来实现, 要多少个 LE 才能完成?

答: 假如说将 2 个 EAB 实现的 FIFO 用 LE 来实现可能要占用很多的 LE 个数来完成, 因为一个 EAB 中有 4Kbit, 假如说全部用 LE 来完成那将是一个很大的浪费, 也不推荐这样做. 但是该 FIFO 要占用多少个 LE 的资源, 要看实际的 FIFO 设计而定, FIFO 的大小, 算法的优劣都决定了 LE 资源的占用.

89. 在使用 Altera 的器件进行设计时, AHDL 是否会优于 VHDL, MAX+PLUS II 对 VHDL 的支持是否欠佳, 怎样才能在设计时更好的使用 VHDL?

答: AHDL 是 ALTERA 在 VHDL 基础上自己定义的一种设计语言, 特别适合做 ALTERA 的器件编成, 但是不通用于别的厂商. 假如用的是 VHDL, 推荐使用 ALTERA 软件包中自带的第三方综合工具对 VHDL 设计进行编译.

90. EPM7128STC100-15 供电电压是多少？是否用外扩 EEPROM？

答：EPM7128STC100-15 供电电压是 5V 的，它不需要外扩 EEPROM. 它本身就是 EEPROM 结构的。

91. 在实际应用中，用到 8 路输入的时钟和输据（每路时钟都是不同频/相的），要求任选 4 路采样。这时候时钟/数据是经过 4 个 8 选 1 产生的，所以不得不使用了“gated clock”，由于 FPGA 内部的延时的不确定，会造成采样错误。是否能提供一个解决方案？

答：可以在设计中引入一个更快的时钟信号，要比该 8 路信号都快，将该时钟信号作为系统时钟，而将其选择出的原时钟信号用来做使能信号，这样就可以将设计改成一个同步时钟触发逻辑，就避免了原来各个不同时钟之间的异步信号。当然这只是一个可能的解决方案，其实现结果要根据具体设计而定。

92. 输入管脚带有上拉电阻，有时还需要外加上拉电阻，他们的作用是什么？

答：The pull-up resistors built into the FPGA I/O pin are relatively weak - usually in the order of several tens of Kohm to a hundred Kohm. It will be too slow if you rely on these weak pull-up to charge up an open-drain signal to logic '1'. You can speed it up by adding a stronger external pull up resistor. （参考译文：置入 FPGA I/O 引脚的上拉电阻相对较弱，通常为几十千欧至几百千欧。如果依靠这些弱电阻来将开漏信号填充至逻辑“1”，则速度会很慢。通过添加外部上拉电阻可以提高速度。）