

第1章 FPGA 中 IO 口时序分析

作者： 屋檐下的龙卷风

博客地址：<http://www.cnblogs.com/linjie-swust/>

日期： 2012.3.1

1.1 概述

在高速系统中 FPGA 时序约束不止包括内部时钟约束，还应包括完整的 IO 时序约束和时序例外约束才能实现 PCB 板级的时序收敛。因此，FPGA 时序约束中 IO 口时序约束也是一个重点。只有约束正确才能在高速情况下保证 FPGA 和外部器件通信正确。

1.2 FPGA 整体概念

由于 IO 口时序约束分析是针对整个电路板整个系统进行时序分析，所以 FPGA 需要作为一个整体分析，其中包括 FPGA 的建立时间、保持时间以及传输延时。传统的建立时间、保持时间以及传输延时都是针对寄存器形式的分析。但是针对整个系统 FPGA 的建立时间保持时间可以简化。

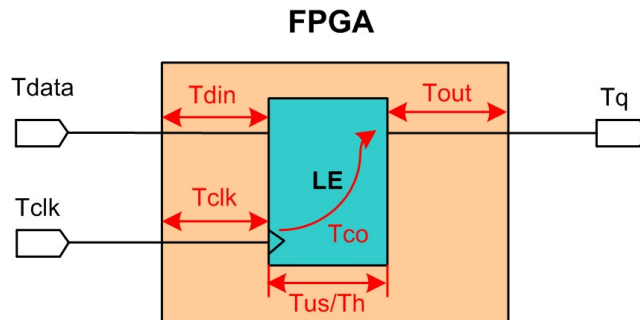


图 1.1 FPGA 整体时序图

如图 1.1 所示，为分解的 FPGA 内部寄存器的性能参数：

- (1) Tdin 为从 FPGA 的 IO 口到 FPGA 内部寄存器输入端的延时；
- (2) Tclk 为从 FPGA 的 IO 口到 FPGA 内部寄存器时钟端的延时；
- (3) Tus/Th 为 FPGA 内部寄存器的建立时间和保持时间；
- (4) Tco 为 FPGA 内部寄存器传输时间；
- (5) Tout 为从 FPGA 寄存器输出到 IO 口输出的延时；

对于整个 FPGA 系统分析，可以重新定义这些参数：FPGA 建立时间可以定义为：

- (1) FPGA 建立时间： $FT_{su} = T_{din} + T_{su} - T_{clk}$ ；
- (2) FPGA 保持时间： $FTh = Th + T_{clk}$ ；
- (3) FPGA 数据传输时间： $FT_{co} = T_{clk} + T_{co} + T_{out}$ ；

由上分析当 FPGA 成为一个系统后即可进行 IO 时序分析了。FPGA 模型变为如图 1.2 所示。

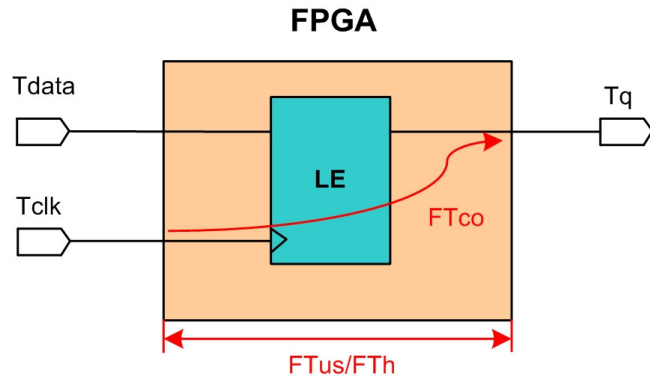


图 1.2 FPGA 系统参数

1.3 输入最大最小延时

外部器件发送数据到 FPGA 系统模型如图 1.3 所示。对 FPGA 的 IO 口进行输入最大最小延时约束是为了让 FPGA 设计工具能够尽可能的优化从输入端口到第一级寄存器之间的路径延迟，使其能够保证系统时钟可靠的采到从外部芯片到 FPGA 的信号。

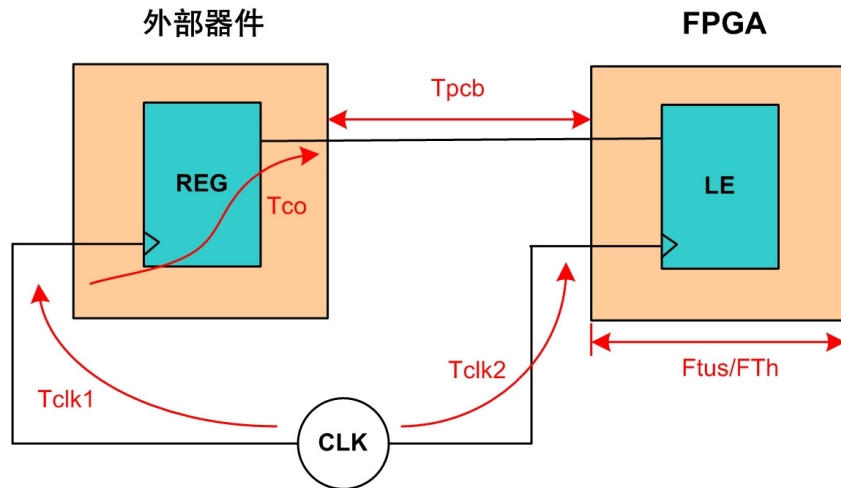


图 1.3 FPGA 数据输入模型

输入延时即为从外部器件发出数据到 FPGA 输入端口的延时时间。其中包括时钟源到 FPGA 延时和到外部器件延时之差、经过外部器件的数据发送 Tco，再加上 PCB 板上的走线延时。如图 1.4 所示，为外部器件和 FPGA 接口时序。

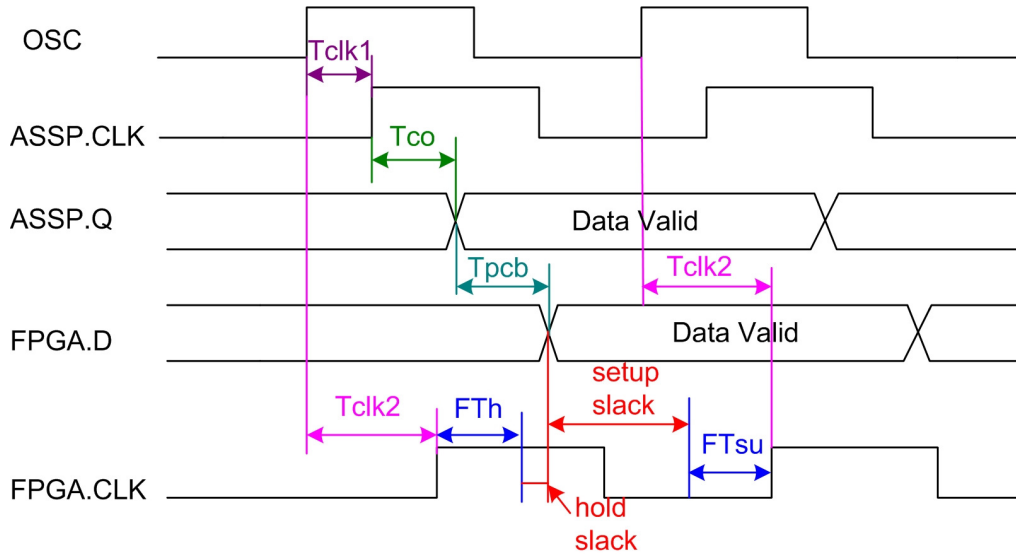


图 1.4 外部器件和 FPGA 接口时序

1. 最大输入延时

最大输入延时 (input delay max) 为当从数据发送时钟沿 (lanuch edge) 经过最大外部器件时钟偏斜(Tclk1),最大的器件数据输出延时(Tco),再加上最大的 PCB 走线延时(Tpcb),减去最小的 FPGA 时钟偏移(FTsu)的情况下还能保证时序满足的延时。这样才能保证 FPGA 的建立时间,准确采集到本次数据值,即为 setup slack 必须为正,如图 1.1 的所示,计算公式如下式所示:

$$\text{Setup slack} = (T_{\text{clk}} + T_{\text{clk2}(\text{min})}) - (T_{\text{clk1}(\text{max})} + T_{\text{co}(\text{max})} + T_{\text{pcb}(\text{max})} + FT_{\text{su}}) \geq 0 \quad (\text{公式 1})$$

推出如下公式:

$$T_{\text{clk1}(\text{max})} + T_{\text{co}(\text{max})} + T_{\text{pcb}(\text{max})} - T_{\text{clk2}(\text{min})} \leq T_{\text{clk}} + FT_{\text{su}} \quad (\text{公式 2})$$

由 Altera 官方数据手册得知:

$$\text{input delay max} = \text{Board Delay}_{(\text{max})} + \text{Board clock skew}_{(\text{min})} + T_{\text{co}(\text{max})} \quad (\text{公式 3})$$

结合本系统参数公式为

$$\text{input delay max} = T_{\text{pcb}(\text{max})} - (T_{\text{clk2}(\text{min})} - T_{\text{clk1}(\text{max})}) + T_{\text{co}(\text{max})} \quad (\text{公式 4})$$

2. 最小输入延时

最小输入延时 (input delay min) 为当从数据发送时钟沿 (lanuch edge) 经过最小外部器件时钟偏斜 (Tclk1), 最小器件数据输出延时 (Tco), 再加上最小 PCB 走线延时 (Tpcb), 此时的时间总延时值一定要大于 FPGA 的最大时钟延时和建立时间之和, 这样才能不破坏 FPGA 上一次数据的保持时间, 即为 hold slack 必须为正, 如图 1.1 的所示, 计算公式如下式所示:

$$\text{Hold slack} = (T_{\text{clk1}(\text{min})} + T_{\text{co}(\text{min})} + T_{\text{pcb}(\text{min})}) - (FTh + T_{\text{clk2}(\text{max})}) \geq 0 \quad (\text{公式 5})$$

推出公式:

$$T_{clk1(min)} + T_{co(min)} + T_{pcb(min)} - T_{clk2(max)} \geq FTh \quad (\text{公式 6})$$

由 Altera 官方数据手册得知:

$$\text{input delay max} = \text{Board Delay}_{(min)} - \text{Board clock skew}_{(min)} + T_{co(min)} \quad (\text{公式 7})$$

结合本系统参数公式为

$$\text{input delay max} = T_{pcb(min)} - (T_{clk2(max)} - T_{clk1(min)}) + T_{co(min)} \quad (\text{公式 8})$$

由公式 4 和公式 8 得知, 进行输入最大最小延时的计算, 我们需要估算 4 个值:

- (1) 外部器件输出数据通过 PCB 板到达 FPGA 端口的最大值和最小值 T_{pcb} , PCB 延时经验值为 600mil/ns , $1\text{mm} = 39.37\text{mil}$;
- (2) 外部器件接收到时钟信号后输出数据延时的最大值和最小值 T_{co} ;
- (3) 时钟源到达外部器件的最大、最小时钟偏斜 T_{clk1} ;
- (4) 时钟源到达 FPGA 的最大、最小时钟偏斜 T_{clk2} ;

当外部器件时钟为 FPGA 提供的时候, T_{clk1} 和 T_{clk2} 即合成 T_{skew} , 如图 1.5 所示:

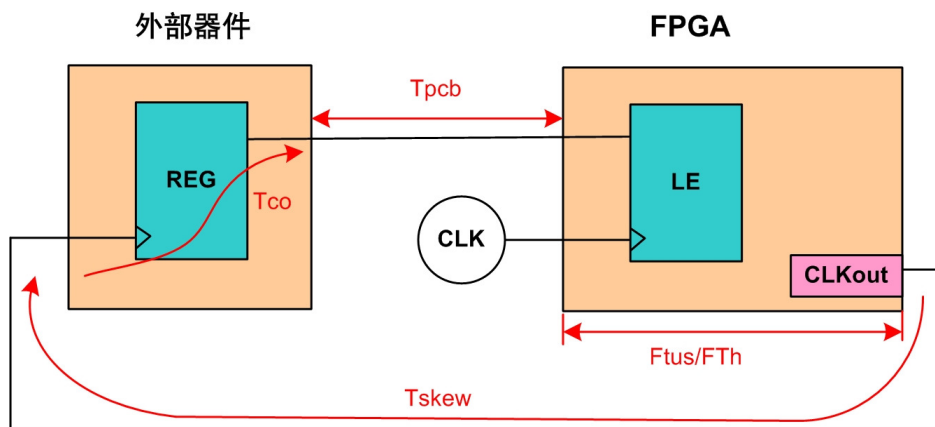


图 1.5 FPGA 输出时钟模型

1.4 输出最大最小延时

FPGA 输出数据给外部器件模型如图 1.6 所示。对 FPGA 的 IO 口进行输出最大最小延时约束是为了让 FPGA 设计工具能够尽可能的优化从第一级寄存器到输出端口之间的路径延迟, 使其能够保证让外部器件能准确的采集到 FPGA 的输出数据。

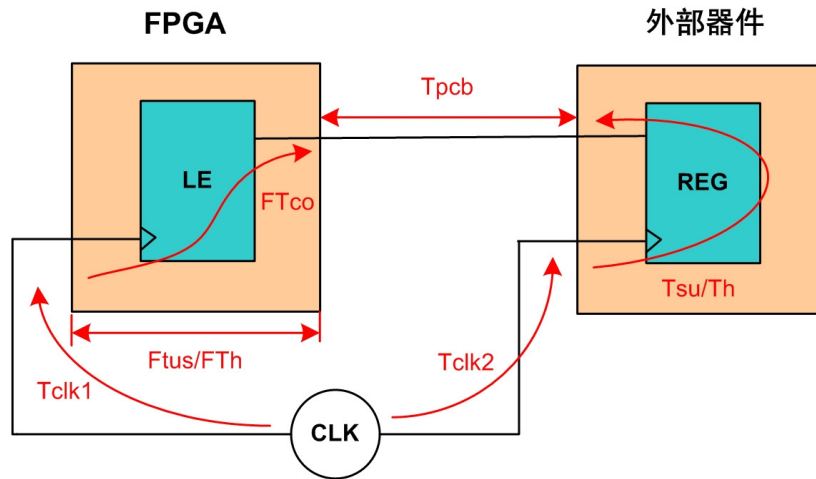


图 1.6 FPGA 输出延时模型

输出延时即为从 FPGA 输出数据后到达外部器件的延时时间。其中包括时钟源到 FPGA 延时和到外部器件延时之差、PCB 板上的走线延时以及外部器件的数据建立和保持时间。如所示，为 FPGA 和外部器件接口时序图。

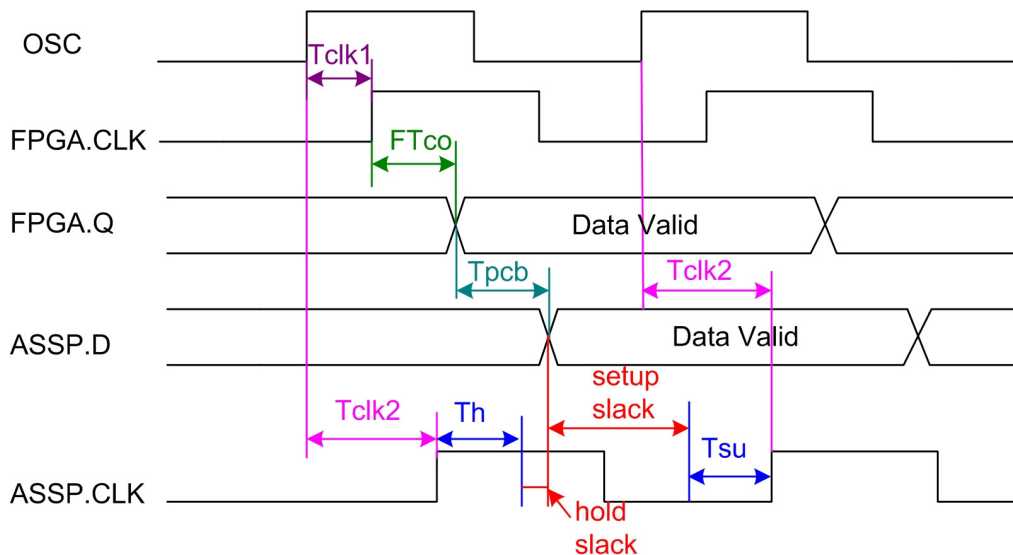


图 1.7 FPGA 输出延时

1. 最大输出延时

由 Altera 官方数据手册得知：

$$\text{Output delay max} = \text{Board Delay}_{(\text{max})} - \text{Board clock skew}_{(\text{min})} + \text{Tsu} \quad (\text{公式 9})$$

由公式得知，最大输出延时（output delay max）为当从 FPGA 数据发出后经过最大的 PCB 延时、最小的 FPGA 和器件时钟偏斜，再加上外部器件的建立时间。约束最大输出延时，是为了约束 IO 口输出，从而使外部器件的数据建立时间，即为 setup slack 必须为正，计算公式如下式所示：

$$\text{Setup slack} = (\text{T}_{\text{clk}} + \text{T}_{\text{clk2}(\text{min})}) - (\text{T}_{\text{clk1}(\text{max})} + \text{FT}_{\text{co}} + \text{T}_{\text{pcb}(\text{max})} + \text{T}_{\text{su}}) \geq 0 \quad (\text{公式 10})$$

推导出如下公式:

$$FT_{co(max)} + T_{pcb(max)} - (T_{clk2(min)} - T_{clk1(max)}) + T_{su} \leq T_{clk} \quad (\text{公式 11})$$

再次推导, 得到如下公式:

$$FT_{co(max)} + \text{Output delay max} \leq T_{clk} \quad (\text{公式 12})$$

由此可见, 约束输出最大延时, 即为通知编译器 FPGA 的 FTco 最大值为+多少, 根据这个值做出正确的综合结果。

1.4.2 输出最小延时

由 Altera 官方数据手册得知:

$$\text{Output delay min} = \text{Board Delay}_{(min)} - \text{Board clock skew}_{(max)} - T_h \quad (\text{公式 13})$$

由公式得知, 最小输出延时 (output delay min) 为当从 FPGA 数据发出后经过最小的 PCB 延时、最大的 FPGA 和器件时钟偏斜, 再减去外部器件的建立时间。约束最小输出延时, 是为了约束 IO 口输出, 从而使 IO 口输出 s 有个最小延时值, 防止输出过快, 破坏了外部器件上一个时钟的数据保持时间, 导致 hold slack 为负值, 不能正确的锁存到数据, 最小输出延时的推导计算公式如下式所示:

$$\text{Hold slack} = (T_{clk1(min)} + FT_{co(min)} + T_{pcb(min)}) - (T_h + T_{clk2(max)}) \geq 0 \quad (\text{公式 14})$$

推导出如下公式:

$$FT_{co(min)} + T_{pcb(min)} - (T_{clk2(max)} - T_{clk1(min)}) - T_h \geq 0 \quad (\text{公式 15})$$

再次推导, 得出如下公式:

$$FT_{co(min)} + \text{Output delay min} \geq 0 \quad (\text{公式 17})$$

由公式得知, 约束输出最大延时, 即为通知编译器 FPGA 的 FTco 最小值为多少, 根据这个值做出正确的综合结果。

由公式 10 和公式 14 得知, 进行输出最大最小延时的计算, 我们需要估算 4 个值:

- (1) FPGA 输出数据通过 PCB 板到达外部器件输入端口的最大值和最小值 Tpcb, PCB 延时经验值为 600mil/ns, 1mm = 39.37mil;;
- (2) 时钟源到达外部器件的最大、最小时钟偏斜 Tclk2;
- (3) 时钟源到达 FPGA 的最大、最小时钟偏斜 Tclk1;
- (4) 外部器件的建立时间 Tsu 和保持时间 Th;

当外部器件时钟为 FPGA 提供的时候, Tclk1 和 Tclk2 即合成 Tshew, 如图 1.8 所示:

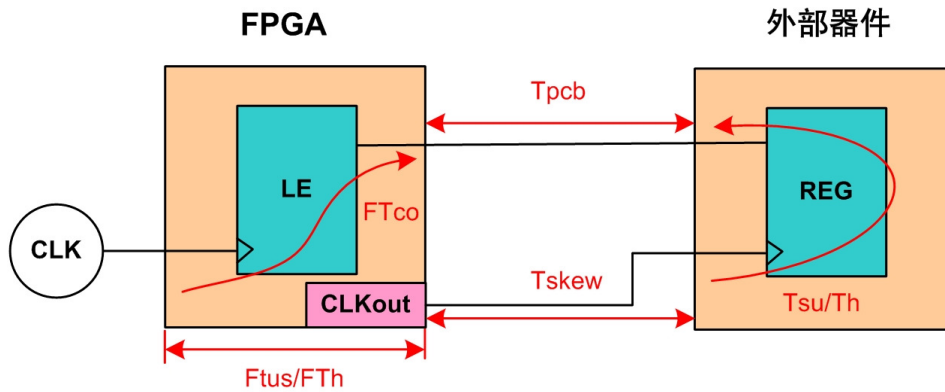


图 1.8 FPGA 提供时钟模型

1.5 使用范围

通过作者使用总结情况，IO 口时序约束主要使用在以下情况：

1. 数据交换频率较高

由于 IO 时序约束一般计算值都是在几纳秒，当 FPGA 和外部数据交换频率较低，如 FPGA 操作 640*480 的 TFT 液晶进行刷屏，数据传输频率仅仅 24Mhz，一个数据时钟都有 41.666ns，完全不用约束都能满足时序要求。但是当操作 SDRAM 运行到 120M 时候，由于一个数据变换周期才 8ns，因此 IO 口的少量延时都会影响到 SDRAM 数据，因此这种情况下需要对输入输出进行完整的 IO 口时序约束，并且分析正确，才能消除数据传输不稳定过的情况。

2. 代码已经比较优化

当数据交换频率较高，但是时序约束还是不满足时序要求的时候，我们都需要对代码进行分析，好的时序都是设计出来的，不是约束出来的。如程序清单 1.1 所示，首先 hcount_r 和 vcount_r 都为 10 位计数器，这样的代码 TFT 的三色输出的端口就会有大的延时，因为 dat_act 的胶合逻辑太多，输出路径太长导致。这种情况下应该不是首先做时序约束，应该修改代码，尽量做到寄存器直接输出。只有当代码比较优化的情况，再做时序约束这样才能得到较好的结果。

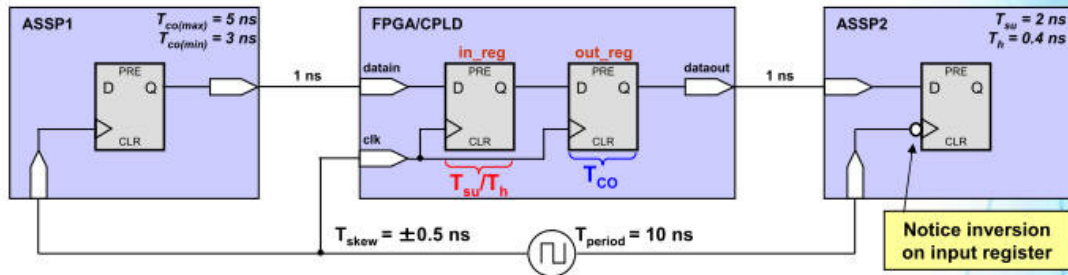
程序清单 1.1 示例程序

```
assign dat_act = ((hcount_r >= hdat_begin) && (hcount_r < hdat_end))
                && ((vcount_r >= vdat_begin) && (vcount_r < vdat_end));
assign tft_r   = (dat_act) ? {rgb16_dat[15:11], 3'b111} : 8'h00;
assign tft_g   = (dat_act) ? {rgb16_dat[10:5],  3'b111} : 8'h00;
assign tft_b   = (dat_act) ? {rgb16_dat[4:0],   3'b111} : 8'h00;
```

1.6 总结

本文档主要是对 FPGA 的 IO 口时序约束进行相应的分析，并未做实际的使用分析，在后续文档中将会结合软件，以及实际案例对 IO 口时序约束进行详细的使用介绍。最后附上一个 Altera 官方的 IO 时序约束分析例子，如图 1.9 所示。

Synchronous I/O Example



```
create_clock -period 10 -name clk [get_ports clk]
```

```
set_input_delay -clock clk -max [expr 1 - (-0.5) + 5] [get_ports datain]
```

```
set_input_delay -clock clk -min [expr 1 - 0.5 + 3] [get_ports datain]
```

```
set_output_delay -clock clk -max [expr 1 - (-0.5) + 2] \
    -clock_fall [get_ports dataout]
```

```
set_output_delay -clock clk -min [expr 1 - 0.5 - 0.4] \
    -clock_fall [get_ports dataout]
```

Note: expr in these constraints is used to simply calculate the value of the equation broken down into the 3 parts defined by the input/output delay equations

© 2009 Altera Corporation
 Altera, Stratix, Arria, Cyclone, MAX, HardCopy, Nios, Quartus, and MegaCore are trademarks of Altera Corporation
 69



图 1.9 Altera 官方例程