

# FPGA 嵌入式系统 开发过程中的 XBD 文件设计

■ 厦门大学 闫铁铮 姚铭

摘要

利用 Xilinx 的嵌入式开发工具 EDK 快速建立 FPGA 嵌入式系统过程中, 板级描述(Xilinx Board Description, XBD) 文件起着至关重要的作用, 它是系统硬件配置文件 MHS 和约束文件 UCF 的基础。XBD 文件的设计过程中, 采用模块化的设计方法对物理电路板上 FPGA 外围各单元的信息进行描述, 从而通过 XBD 文件实现 IP 核和板上电路模块之间的连接。

关键词 FPGA 嵌入式系统 XBD 文件 EDK

随着可编程逻辑器件的不断进步和发展, FPGA 在嵌入式系统中发挥着越来越重要的作用, 已经开始被广泛应用于通信、航天、医疗、工控等领域。Xilinx 公司作为全球最大的可编程逻辑器件生产厂商, 为嵌入式系统设计人员提供了比较全面的解决方案。Xilinx 的嵌入式系统开发环境 EDK, 提供了一种通用的完全集成的硬件和软件开发环境, 使设计人员可以利用单个开发环境快速配置针对 PowerPC 硬处理器或 Microblaze 软处理器内核的平台。在 EDK 开发环境下, 利用用户向导可以准确快速地创建一个新的嵌入式系统, 而在这一过程中, XBD 文件决定了系统的硬件平台描述文件 MHS 以及约束文件 UCF 的正确与否, 在构建系统过程中起着至关重要的作用。

## 1 基于 FPGA 的嵌入式系统开发

利用 Xilinx 公司的嵌入式系统开发工具 EDK 就可以完成整个嵌入式系统的硬件和软件开发, EDK 由 XPS (Xilinx Platform Studio) 和 SDK (Software Development Kit) 组成。其中 XPS 是主设计程序平台, 可以实现嵌入式系统开发的所有步骤, 并且可以在其中调用 SDK。SDK 是软件开发工具, 支持 C 和 C++, 主要完成软件设计。在 XPS 开发环境下, 完整的开发流程如图 1 所示。

MHS 和 MSS 文件都是根据系统要求在 EDK 环境下生成的。MHS 文件包含了对整个嵌入式系统的定义, 包括处理器、总线、外围设备、地址空间等, 用于整个硬件平台的综合、实现; MSS 文件包含了操作系统、设备驱动等信息, 将其输入到库生成器 (Libgen), 产生应用程序中需要的驱动程序及 Xilinx 的调用库。

嵌入式开发软件 EDK 为设计人员提供了自动化的设

计向导——BSB (Base System Builder), 可以指引工程师快速完成整个设计过程。在利用 BSB 创建嵌入式系统过程中, 利用嵌入式硬件平台的 XBD 文件对板级各个功能电路的描述, 按步骤选择需要的电路模块添加到 MHS 文件中, 从而减少出现错误的可能, 降低学习难度。因此, 一个新的嵌入式 FPGA 硬件平台确定以后, 在 EDK 中针对开发板进行嵌入式系统开发, 可以利用相应的 XBD 文件, 快速建立一个基于 FPGA 的嵌入式系统。

## 2 板级描述文件 XBD

XBD (Xilinx Board Description) 文件定义了电路板的功能模块以及各个模块与 FPGA 芯片的接口情况, 利用 BSB 可以将 XBD 文件中包含的功能模块加入到要建立的嵌入式系统中。

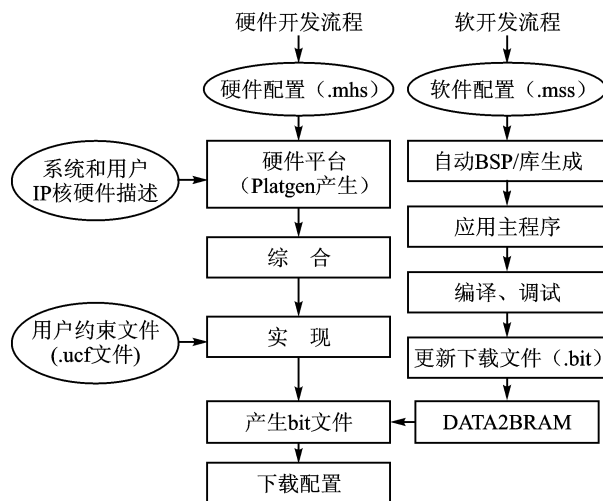


图 1 开发流程

在 EDK 的安装目录下, Xilinx 公司为设计者提供了一些开发板的 XBD 文件。但是, 一个新的嵌入式硬件平台建立后, 想要在 EDK 软件中利用 BSB 向导读取硬件电路的信息, 从而高效无误地产生 FPGA 嵌入式系统, 需要针对电路板上的各个单元电路设计新的 XBD 文件来描述硬件平台的信息。通常, 一个 XBD 文件包括如下信息:

- ◆ 电路板所支持的功能模块的 FPGA 接口;
- ◆ 每个模块的属性、参数、端口定义;
- ◆ 不同端口或模块间的连接信息;
- ◆ 每个 FPGA 引脚的 UCF 约束信息。

由于 XBD 文件是对硬件各功能模块的描述, 因此在设计 XBD 过程中也是以模块的形式来表示电路板信息的, 而且各个功能块的描述具备相似的结构, 使用相同的赋值命令进行具体参数定义。

## 2.1 模块的定义

一个完整的模块定义如下:

```
BEGIN < block_type_keyword>
:
END
```

关键字 BEGIN 表示一个新的模块开始, 后面是要定义模块的类型, 中间部分是描述 IP 核行为的 MPD 文件相对应的各种参数。当前, XBD 文件能够识别 3 种类型的模块定义:

① IO\_INTERFACE。IO\_INTERFACE 指定了一个电路板上的物理模块(不包括 FPGA 本身), 每一个 IO\_INTERFACE 在板上应该有一个在 FPGA 中使用的软 IP 核与之相对应。

② IO\_ADAPTER。IO\_ADAPTER 指定了连接 IO\_INTERFACE 引脚与相应软 IP 端口的软胶合逻辑。

③ FPGA。FPGA 模块代表 FPGA 本身。

## 2.2 赋值命令

每个 BEGIN-END 模块包括多个赋值命令。赋值命令至少包括一个 name value 对, 还可以加入多个 name value 子对。

赋值命令包括:

① ATTRIBUTE。ATTRIBUTE 命令是对属性命名的关键字。对于经过 ATTRIBUTE 赋值的对象, EDK 工具会执行某种操作或以特定的方式使用该对象。ATTRIBUTE 赋值命令可以在 BEGIN-END 模块内部或者外部使用。

② PARAMETER。PARAMETER 命令将 IP 核中的 PARAMETER 参数同 XBD 文件中的 IO\_INTERFACE 联系在一起, 且 PARAMETER 命令只能在 IO\_INTERFACE 模块内部使用。

③ PORT。PORT 命令用来指定电路板上各个模块(包括 FPGA)的连通性, 只能在 IO\_INTERFACE 和 IO\_ADAPTER 模块内使用。

PARAMETER 和 PORT 命令后能够跟随子属性, 每一个子属性也是 name value 对。子属性必须同 PARAMETER 和 PORT 命令在同一行, 并且用逗号隔开。

## 3 XBD 文件的设计实现

在基于 FPGA 的嵌入式系统中, 无论是硬核 PowerPC 还是软核 Microblaze 处理器, 都是通过 IP 核的方式与周围设备进行操作的, 因此, 在 XBD 中定义电路模块时要选择与实际电路相对应的 IP 核。在描述 IP 核行为的 MPD 文件中, 定义了 IP 核的各种参数, IO\_INTERFACE 的子属性 IOTYPE 决定了该 IP 核是否能够与电路板上的特定模块进行连接, 通过查看 MPD 文件的信息就可以知道该 IP 的功能及其各种参数。例如, 对于电路板上的 4 个 LED 显示单元, 为了能够利用 BSB 将其加入到嵌入式系统中, 需要在 XBD 文件中定义一个能够对 4 个 LED 进行操作的模块。在基于 FPGA 的嵌入式系统中, 处理器通过通用 I/O 接口 GPIO 对 LED 进行操作, 故 LED 在 XBD 中要定义一个 IOTYPE 为 GPIO 的 IO\_INTERFACE 模块, 表示可以通过这个模块与实际电路板上的 I/O 进行通信。在 GPIO 的 MPD 文件描述中, 对于 IO\_INTERFACE 进行了如下定义:

```
IO_INTERFACE IO_IF= gpio_0, IO_TYPE= XIL_GPIO_V1
```

这个 IO\_INTERFACE 表明 IP 核可以和 GPIO 进行通信。

与电路板上特定模块通信的 IP 核选定后, 根据实际的电路特点, 在 XBD 文件中指定模块的参数值, 这些参数值都要与 MPD 文件中的参数一一对应。需要指出的是, EDK 工具是利用 XBD 文件中的 IO\_IS 子属性将 IP 核的端口与电路板上模块连接在一起的。在 MPD 文件中, GPIO 的部分描述摘录如下:

```
BEGIN opb_gpio
## Peripheral Options
OPTION ...
IO_INTERFACE IO_IF= gpio_0, IO_TYPE= XIL_GPIO_V1
...
## Generics for VHDL or Parameters for Verilog
PARAMETER C_GPIO_WIDTH= 32, DT= integer, IO_IF= gpio_0, IO_IS= num_bits
PARAMETER C_ALL_INPUTS= 0, DT= integer, IO_IF= gpio_0, IO_IS= all_inputs
PARAMETER ...
```



```
## Ports
PORT GPIO_IO= "", DIR= INOUT, IO_IF= gpio_0, IO_
IS= gpio_io, ...
PORT ...
END
```

MPD 文件定义了 IOTYPE 类型为 XIL\_GPIO\_V1 的 I/O 接口, 并且有 C\_GPIO\_WIDTH 和 C\_ALL\_INPUTS 两个参数。这些参数跟随的 IO\_INTERFACE 是通过 IO\_IF 子属性来指定的。同样, PORT GPIO\_IO 也是通过 IO\_IF 子属性来表示属于 gpio\_0 接口的。通过 MPD 文件对 GPIO 的描述, 可以通过将电路板上的 4 个 LED 电路模块在 XBD 文件中定义一个 IO\_INTERFACE 模块来表示。

```
BEGIN IO_INTERFACE
ATTRIBUTE IOTYPE= XIL_GPIO_V1
ATTRIBUTE INSTANCE= LEDs_4Bit
PARAMETER num_bits = 4, IO_IS= num_bits
PARAMETER is_dual= 0, IO_IS= is_dual
PARAMETER bidir_data= 0, IO_IS= is_bidir# Non-bidir
data pins
PARAMETER all_inputs = 0, IO_IS= all_inputs # All out-
puts
PORT LED1= CONN_LED1, IO_IS= gpio_data_out[0], INI-
TIALVAL= VCC, UCF_NET_STRING= ("LOC= R20")
PORT LED2= CONN_LED2, IO_IS= gpio_data_out[1], INI-
TIALVAL= VCC, UCF_NET_STRING= ("LOC= T19")
PORT LED3= CONN_LED3, IO_IS= gpio_data_out[2], INI-
TIALVAL= VCC, UCF_NET_STRING= ("LOC= U20")
PORT LED4= CONN_LED4, IO_IS= gpio_data_out[3], INI-
TIALVAL= VCC, UCF_NET_STRING= ("LOC= U19")
END
```

可以看到, MPD 文件中 IOTYPE 类型为 XIL\_GPIO\_V1 的 IO\_INTERFACE 同 XBD 文件中 IOTYPE 类型同样为 XIL\_GPIO\_V1 的 LEDs\_4BIT 模块是匹配的。因此, 子属性 IO\_IS 决定了 IP 核中的端口同 XBD 文件中哪个端口连接。最终, 利用自己设计的 XBD 文件中的 LEDs\_4BIT 模块, 通过 BSB 建立一个基于 FPGA 的嵌入式系统。实际电路板中 4 个 LED 在构建的硬件配置文件 MHS 中的描述如下:

```
## Global Ports
PORT fpga_0_LEDs_4Bit_GPIO_d_out_pin= fpga_0_LEDs_
_4Bit_GPIO_d_out, DIR= 0, VEC= [0:3]
:
## GPIO instance
BEGIN opb_gpio
ATTRIBUTE INSTANCE= LEDs_4Bit
PARAMETER HW_VER= 3.01.b
PARAMETER C_GPIO_WIDTH= 4## IO_IS= num_bits
```

```
PARAMETER C_ALL_INPUTS= 0## IO_IS= all_inputs
PORT GPIO_d_out= fpga_0_LEDs_4Bit_GPIO_d_out ##
IO_IS= gpio_io
:
END
```

由设计的 XBD 文件生成的 UCF 文件对 LED 模块部分的描述:

```
Net fpga_0_LEDs_8Bit_GPIO_d_out_pin< 0> LOC= r20;
Net fpga_0_LEDs_8Bit_GPIO_d_out_pin< 1> LOC= t19;
Net fpga_0_LEDs_8Bit_GPIO_d_out_pin< 2> LOC= u20;
Net fpga_0_LEDs_8Bit_GPIO_d_out_pin< 3> LOC= u19;
```

类似于在 XBD 文件中设计一个与 4 个 LED 进行通信的 GPIO 模块, 利用同样的方法, 在 XBD 中能够完成电路板上其他模块的设计, 如 UART、按键、存储器等。在 XBD 文件中将电路板上所有的模块信息设计完成之后, 为了使 EDK 软件中的开发向导 BSB 能够读取到自己设计的 XBD 文件的信息, 要将 XBD 文件存放到 EDK 安装目录下的 < library\_name > /boards 中。应该注意的是, 开发板的目录名称必须同板本身的名称相同, 每个开发板目录下应该包括一个 /data 目录, XBD 文件必须以 < board\_name > \_v2\_2\_0. xbd 的形式命名, 并且存放在这个 /data 目录下, 例如 ../boards/myboard\_rev1/data/myboard\_rev1\_v2\_2\_0. xbd。这样, 在利用 BSB 创建嵌入式系统过程中, XPS 工具就会自动搜索 < library\_name > /boards 目录, 并将该目录下存在的描述开发板信息的 XBD 文件在创建向导 BSB 中显示出来。

## 4 总结

板级描述文件 XBD 在利用 BSB 快速构建基于 FPGA 的嵌入式系统中起着非常重要的作用, 本文针对新的硬件平台的 XBD 文件设计进行了介绍。在 XBD 文件设计时, 针对电路板上的各个实际电路模块, 通过对照相应描述 IP 行为的 MPD 文件进行各个电路模块的设计与实现。

## 参考文献

- [1] Platform Specification Format Reference Manual[OL]. [http://china.xilinx.com/ise/embedded/edk91i\\_docs/psf\\_rm.pdf](http://china.xilinx.com/ise/embedded/edk91i_docs/psf_rm.pdf).
- [2] 张燕, 冯永新. 基于 MicroBlaze 系统的 AD 数据采集与实现[J]. 电子技术, 2007(7-8).
- [3] Embedded System Tools Reference Manual[OL]. [http://china.xilinx.com/ise/embedded/edk91i\\_docs/est\\_rm.pdf](http://china.xilinx.com/ise/embedded/edk91i_docs/est_rm.pdf).
- [4] 王磊. 基于 MicroBlaze 软核的 FPGA 片上系统设计[J]. 新器件新技术, 2004(7).

(收稿日期: 2008-02-25)