

MIPSfpga

by Imagination

Lab9 移植 MIPSfpga 到其他 FPGA 板



These materials produced in association with Imagination.
Join our University community for more resources.

[**community.imgtec.com/university**](https://community.imgtec.com/university)

Lab9 移植 MIPSfpga 到其他 FPGA 板

1. 介绍

这个实验描述了如何把 MIPSfpga 系统移植到除 Nexys4 DDR 板以外的其他 FPGA 板，例如当你想要更低的成本或者你正有一个其他电路板。

我们将会描述如何把 MIPSfpga 系统移植到 Digilent 的 Basys3 板和作为 Nexys4 DDR 板前身的 Nexys4 板。你可以按照下面相似的步骤把 MIPSfpga 系统移植到其他基于 Xilinx FPGAs 的板子上。表 1 综述了 Nexys4 和 Basys3 板以及在之前实验中我们一直使用的 Nexys4 DDR 板的特点。如果需要，之前的实验也可以在 Nexys4 或者 Basys3 或者其他板上完成。

表 1. FPGA 板

板	整体规格	网络链接	花费
Nexys4 DDR	FPGA: Artix-7 (XC7A100T-CSG324C) block RAM 容量: 607 KB LEs 数量: 101 K 7-段数码管显示器数量: 8 拨码开关数量: 16 按钮数量: 5 LEDs 数量: 16 PMOD 连接器数量: 5	http://www.digilent.com.cn/products/product-nexys-4-ddr-artix-7-fpga-trainer-board.html	<2000RMB
Nexys4	和 Nexys4 DDR 相似		<2000RMB
Basys3	FPGA: Artix-7 (XC7A35T-CPG236C) block RAM 容量: 225 KB LEs 数量: 33K 7-段数码管显示器数量: 4 拨码开关数量: 16 按钮数量: 5 LEDs 数量: 16 PMOD 连接器数量: 4	http://www.digilent.com.cn/products/product-basys3-artix-7-fpga-board.html	<1000RMB

2. 移植 MIPSfpga 到 Digilent 的 Basys3 板

在这部分我们向你展示了如何把 MIPSfpga 系统移植到 Digilent 的 Basys3 板上，如图 1 所示。

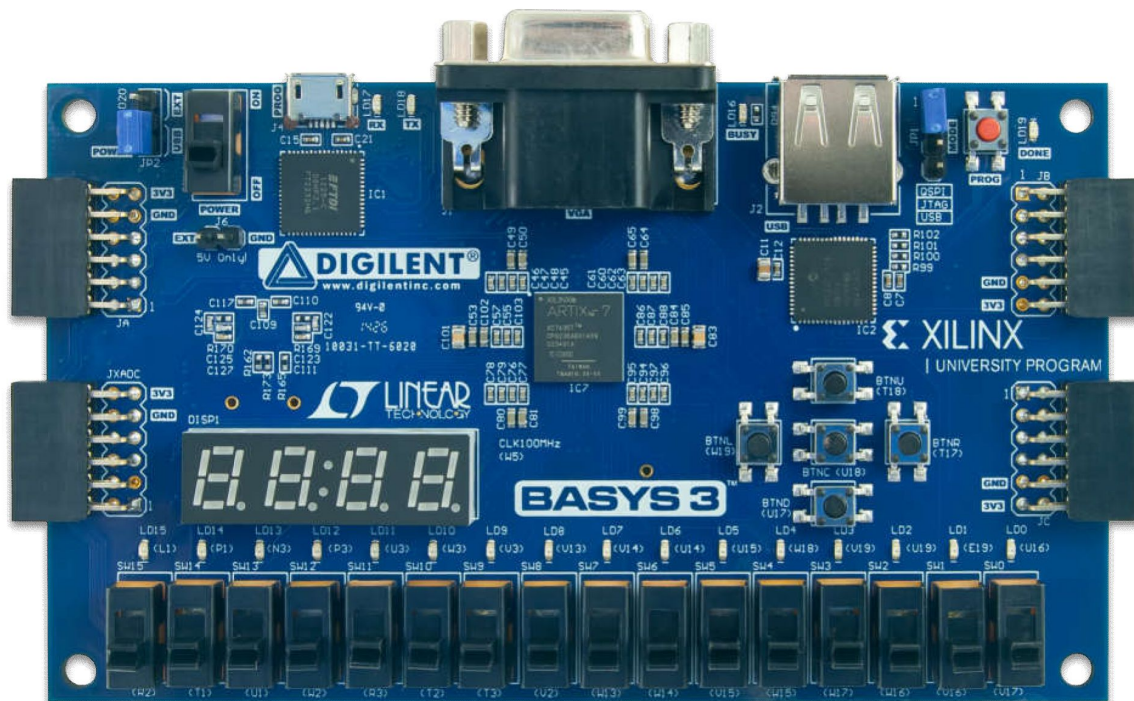


图 1. Digilent 的 Basys3 板

正如在表 1 所列出的那样，Basys3 和 Nexys4 DDR 板相似，它们都是建立在 Xilinx 的 Artix7 FPGA 之上的，但是 Basys3 是 Nexys4 DDR 板的更低版本的 FPGA。Basys3 的 Artix7 FPGA 的逻辑单元数量和储存单元容量大约是 Nexys4 DDR 板的三分之一（Basys3 上有 225KB 的 block RAM 和 33K 可配置的逻辑块(CLBs)，而 Nexys4 DDR 上有 607KB 的 block RAM 和 102K 的 CLB）。MIPSfpga 硬件容易适合任一块板，但它的 block RAM 大小需要减少以来适应 Basys3 板。

3. 对 Basys3 板上的 MIPSfpga 的修改

为了把 MIPSfpga 系统移植到 Basys3 板上，我们必须：

- 步骤 1：写一个封装模块来把 MIPSfpga I/O 映射到 Basys3 板 I/O 上。
- 步骤 2：减少 MIPSfpga 系统存储单元的大小来适应 Basys3 板。
- 步骤 3：添加约束文件把板 I/O 映射到正确的 FPGA 引脚上。

我们详细地描述了每一步，然后展示了如何把 MIPSfpga 系统下载到 Basys3 板上。我们也描述了如何构建一个 Vivado 工程并且为 Basys3 板编译 MIPSfpga 系统。

步骤 1: Basys3 封装模块

首先，我们写了一个封装模块来把 mipsfpga I/O 映射到 Basys3 板的 I/O 上。进入 Lab09_PortingMIPSfpga\Basys3 文件夹，然后打开 mipsfpga_basys3.v 文件。

首先，注意 mipsfpga_basys3 模块的输入和输出。这些信号是 Basys3 板的接口。

```
module mipsfpga_basys3( input      clk,
                      input      btnU, btnD, btnL, btnR, btnC,
                      input  [15:0] sw,
                      output [15:0] led,
                      inout  [ 5:0] JB,
                      );
```

clk 是板载 100MHz 时钟。btnU, btnD, btnL, btnR 和 btnC 是 Basys3 板上的上、下、左、右和中按钮的名字。我们使用 btnC 来复位处理器。

输入信号 sw[15:0] 是 16 个开关，led[15:0] 是 16 个发光二极管等等。PMODB 连接器引脚上的 JB 信号连接到 Bus Blaster 所需的 EJTAG 信号。注意到 JB 连接器在 Basys3 板的右上方（见图 1），在 Nexys4 DDR 板的右下方。

接下来，实例化锁相环 clk_wiz_0，在板载 100MHz 时钟下产生一个 50MHz 的 MIPSfpga 系统时钟。

```
clk_wiz_0 clk_wiz_0(.clk_in1(clk), .clk_out1(clk_out));
```

接下来的数行手动地连接 tck 引脚（JB[3]）到输入缓冲器（IBUF），然后连接到时钟缓冲器（BUFG）来解决 JB[3] 引脚携带了一个时钟信号但没有连接到一个时钟缓冲器上的问题。

```
IBUF IBUF1(.O(tck_in), .I(JB[3]));
BUFG BUFG1(.O(tck), .I(tck_in));
```

最后，这个模块的核心是实例化 MIPSfpga 系统（mipsfpga_sys）并且连接到 Basys3 I/O。

```
mipsfpga_sysmipsfpga_sys(
    .SI_Reset_N(~btnC),
    .SI_ClkIn(clk_out),
    .HADDR(),
    .HRDATA(),
```

```

        .HWDATA(),
        .HWRITE(),
        .EJ_TRST_N_probe(JB[4]),
        .EJ_TDI(JB[1]),
        .EJ_TDO(JB[2]),
        .EJ_TMS(JB[0]),
        .EJ_TCK(tck),
        .SI_ColdReset_N(JB[5]),
        .EJ_DINT(1'b0),
        .IO_Switch({2'b0,sw}),
        .IO_PB({btnU, btnD, btnL, 1'b0, btnR}),
        .IO_LEDR(led),
        .IO_LEDG()
    );

```

步骤 2: 减少内存

Basys3 板仅有 225KB 的 block RAM，而不是 Nexys4 DDR 板上的 607KB。因此，这两个内存块（boot RAM 有 128KB，program RAM 有 256KB）不适合 Basys3 板。幸运的是，启动代码可以适合在 32 KB，我们可以限制我们的程序内存为 64 KB。所以，Basys3 板满足总共的内存需要（32KB+64KB=96KB）。最后剩余的 225-96=129KB block RAM 能够用在 MIPSfpga 系统的其他内存需求，例如高速缓存。

我们通过修改声明在 Verilog 头文件中的内存大小来减少内存的数量。再一次，进入到 Lab09_PortingMIPSfpga\Basys3 目录，打开 mipsfpga_ahb_const.vh 文件。Reset（boot）RAM 地址的大小是 13 位，因此 boot RAM 有 2^{13} 32-bit 字 = 2^{15} 字节 = 32KB。

```
`define H_RAM_RESET_ADDR_WIDTH (13)
```

Program RAM 地址的大小是 14 位，因此 program RAM 有 2^{14} 32-bit 字 = 2^{16} 字节 = 64KB。

```
`define H_RAM_ADDR_WIDTH (14)
```

步骤 3: Basys3 约束文件

作为最后一步，我们增加了一个约束文件来把封装模块（mipsfpga_basys3）的 I/O 信号名称映射到 Basys3 板上正确的 FPGA 引脚。再一次，进入到 Lab09_PortingMIPSfpga\Basys3 目录并打开 mipsfpga_basys3.xdc Xilinx 设计约束文件。这个文件把 FPGA 引脚映射到 mipsfpga_basys3 封装模块的输入和输出上。例如下面一行映射输入 clk 到 FPGA 的引脚 W5 上，这个引脚输出 Basys3 板的 100MHz（周期 10ns）时钟。

```
set_property PACKAGE_PIN W5 [get_ports clk]
```



```

set_property IOSTANDARD LVCMOS33 [get_ports clk]
create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5}
[get_ports clk]

```

下面一行解决了 EJTAG 接口的 tck 引脚没有连接到有时钟缓冲器 FPGA 引脚上的问题。

```

set_property CLOCK_DEDICATED_ROUTE FALSE [get_nets tck_in]

```

下面的几行把拨码开关输入（sw[15:0]）映射到的 FPGA 引脚上，这些引脚通过物理连线连接到 Basys3 板上的拨码开关上。例如 sw[0]连接到 Artix7 封装的引脚 V17 上，sw[1]连接到引脚 W16 上等等。它们使用了 LVCMOS3.3V 信号级电平。

```

## Switches
set_property PACKAGE_PIN V17 [get_ports {sw[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {sw[0]}]
set_property PACKAGE_PIN V16 [get_ports {sw[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {sw[1]}]
set_property PACKAGE_PIN W16 [get_ports {sw[2]}]
...

```

关于 FPGA I/O 的时序约束在文件的底部定义。

4. 把 MIPSfpga 系统加载到 Basys3 板上

一个预编译的比特流文件 mipsfpga_basys3.bit 包含了以 Basys3 板为目标的 MIPSfpga 系统。这个文件在下列文件夹中：Lab09_PortingMIPSfpga\Basys3

打开 Vivado 并且加载比特流文件到 Basys3 板上，按下 Basys3 板上的中央按钮（BTNC）来复位处理器，你现在应该可以看到发光二极管显示增加的数值。为了使用 Codescape 和 Bus Blaster 连接器来编程 MIPSfpga 核，连接 Bus Blaster 连接器到在板右上部（标记为 JB）的 PMOD B 连接器上。按照在之前实验（例如，实验 2）中提供的相同的说明来下载和调试 MIPSfpga 系统。

5. 为 Basys3 板上的 MIPSfpga 建立一个 Vivado 工程

使用与在实验 1 中描述的相似的步骤来建立一个以 Basys3 板上的 Artix7 为目标的 Vivado 工程。在建立工程之前，先复制 MIPSfpga_rtl_up 目录并且命名这个复制的目录为 MIPSfpga_rtl_up_basys3。复制 mipsfpga_basys3.v 和 mipsfpga_ahb_const.vh 文件到 MIPSfpga_rtl_up_basys3 文件夹中。

现在打开 Vivado 并且创建一个新工程。增加 MIPSfpga_rtl_up_basys3 文件夹中的所有文件作为源文件，增加 Lab09_PortingMIPSfpga\Basys3 目录中 mipsfpga_basys3.xdc 文件作为约束文件。你也需要增加一个 50Mhz 的 PLL（锁相环，见实验 1 的说明）。这个工程关于 Basys3 板的目标设备是 xc7a35tcpg236-1 Artix7 FPGA。在工程建立之后，编译，综合并且通过生成一个比特流文件来布局布线这个设计，这正如实验 1 所描述的那样。

6. 在 Nexys4 板上的 MIPSfpga

一些大学或实验室可能有传统的 Nexys4 板，它是 Nexys4 DDR 板的前身。为了方便大家，我们提供了 Nexys4 板的封装文件和 Xilinx 设计约束文件。它们在下面的目录中：

```
MIPSfpga_Wuhan_v12\Lab\Lab09_PortingMIPSfpga\Nexys4
```

这个目标设备 Artix-7 FPGA 和 Nexys4 DDR 板相同都是：xc7a100tcsg324-1。我们也提供了一个已经编译好的比特流文件（mipsfpga_nexys4.bit）放在上述目录下。

7. Xilinx 其他的 FPGA 和 FPGA 板

你也可以按照这里描述的相同的方法来下载到不同的 FPGA 板或者 Xilinx FPGA。显然，你需要写一个封装模块，包括特定板卡的 Xilinx 设计约束文件（.xdc）并可能修改 MIPSfpga 系统使用的内存数量。对于 MIPSfpga 系统来说，一些 FPGA 可能太小而不适合。