



# **Nuclei™ N100 系列处 理器内核配套 FPGA 实现**

## 版权声明

版权所有 © 2018–2020 芯来科技（Nuclei System Technology）有限公司。保留所有权利。

Nuclei™是芯来科技公司拥有的商标。本文件使用的所有其他商标为各持有公司所有。

本文件包含芯来科技公司的机密信息。使用此版权声明为预防作用，并不意味着公布或披露。未经芯来科技公司书面许可，不得以任何形式将本文的全部或部分信息进行复制、传播、转录、存储在检索系统中或翻译成任何语言。

本文文件描述的产品将不断发展和完善；此处的信息由芯来科技提供，但不做任何担保。

本文件仅用于帮助读者使用该产品。对于因采用本文件的任何信息或错误使用产品造成的任何损失或损害，芯来科技概不负责。

## 联系我们

若您有任何疑问，请通过电子邮件 [support@nucleisys.com](mailto:support@nucleisys.com) 联系芯来科技。

## 修订历史

版本号	修订日期	修订的章节	修订的内容
1.0	2019/11/20	N/A	1. 初始版本

## 目录

版权声明.....	0
联系我们.....	0
修订历史.....	1
<b>1. N100 内核配套 SOC.....</b>	<b>3</b>
<b>2. 蜂鸟 FPGA 评估板和 JTAG 调试器 .....</b>	<b>3</b>
2.1. 简介.....	3
2.2. 连接蜂鸟调试器与 FPGA 评估板.....	3
2.3. 安装蜂鸟调试器的驱动 .....	3
<b>3. 搭建 FPGA 原型平台.....</b>	<b>4</b>
3.1. FPGA 项目介绍 .....	4
3.2. 生成 MCS 文件 .....	5
3.3. 将 MCS 文件烧写入 FPGA.....	6
<b>4. 基于 FPGA 平台运行和调试软件示例.....</b>	<b>7</b>
4.1. 配套基于 LINUX 的 SDK .....	7
4.2. 配套基于 WINDOWS 的 IDE.....	7

## 1. N100 内核配套 SoC

对于 N100 系列处理器内核而言，还需要为其配备原型 SoC 才能展示一个典型 SoC 的功能。Nuclei N100 系列内核搭配完整的原型 SoC 平台，请参见《Nuclei\_N100 系列配套 SoC 介绍》了解更多 SoC 的介绍与信息。

## 2. 蜂鸟 FPGA 评估板和 JTAG 调试器

### 2.1. 简介

芯来科技定制了专用的 FPGA 评估板（HBird Eval Kit），本文中称为蜂鸟 FPGA 评估板。上节描述的配套 SoC 原型（包含 N100 系列处理器内核）可以被整体实现在 HBird Eval Kit 上成为 SoC 的 FPGA 原型开发板。

芯来科技还定制了专用的 JTAG 调试器（HBird Debugger Kit），本文中称为蜂鸟调试器，用于在 FPGA 评估板上对 N100 系列进行调试。

有关 FPGA 评估板和专用 JTAG 调试器的总体介绍请参见芯来科技官网开发板页面：

<http://www.nucleisys.com/developboard.php>

详细介绍请参见芯来科技官网开发板页面中的详细介绍文档链接：

[http://www.nucleisys.com/theme/package/Nuclei\\_FPGA\\_DebugKit\\_Intro.pdf](http://www.nucleisys.com/theme/package/Nuclei_FPGA_DebugKit_Intro.pdf)

### 2.2. 连接蜂鸟调试器与 FPGA 评估板

关于如何将蜂鸟调试器与 FPGA 评估板进行连接，请参见芯来科技官网开发板页面中的详细介绍文档链接：[http://www.nucleisys.com/theme/package/Nuclei\\_FPGA\\_DebugKit\\_Intro.pdf](http://www.nucleisys.com/theme/package/Nuclei_FPGA_DebugKit_Intro.pdf)

### 2.3. 安装蜂鸟调试器的驱动

由于蜂鸟调试器是基于 U 盘与主机 PC 连接，因此需要在 PC 上安装驱动，关于如何为蜂鸟调

试器安装驱动，请参见芯来科技官网开发板页面中的详细介绍文档链接：

[http://www.nucleisys.com/theme/package/Nuclei\\_FPGA\\_DebugKit\\_Intro.pdf](http://www.nucleisys.com/theme/package/Nuclei_FPGA_DebugKit_Intro.pdf)

## 3. 搭建 FPGA 原型平台

### 3.1. FPGA 项目介绍

N100 系列内核配套 FPGA 项目在 n100\_rls\_pkg 文件包的文件层次结构如下所示：

```
N100_rls_pkg
|n100_cct
|----fpga                                // 存放 FPGA 项目和脚本的目录
|----hbirdkit                            // 蜂鸟 FPGA 评估板的项目文件夹
|----constrs                             // 存放约束文件的文件夹
|----nuclei-master.xdc                   // 主约束文件
|----Makefile                             // Makefile 脚本
|----script                             // 存放运行 TCL 脚本的文件夹
|----src                                 // 存放 Verilog 源代码的文件夹
|----system.org                          // FPGA 系统的顶层模块
|----Makefile                             // Makefile 脚本
```

FPGA 项目通过 Makefile（fpga/common.mk 文件）将添加一个特殊的宏 FPGA\_SOURCE 至 Core 的宏文件中，如下图所示。所以最终用于编译 FPGA 比特流的 RTL 源代码必须包含此宏（FPGA\_SOURCE）。

```
20 install:
21     mkdir -p ${PWD}/install
22     cp ${PWD}/../rtl/${CORE} ${INSTALL_RTL} -rf
23     cp ${PWD}/artydevkit/src/system.org ${INSTALL_RTL}/system.v -rf
24     sed -i 's/n200/${CORE}/g' ${INSTALL_RTL}/system.v
25     sed -i 'li\`define FPGA_SOURCE\`' ${INSTALL_RTL}/core/${CORE}_defines.v
```

图 3-1 FPGA 项目宏定义文件中添加 FPGA\_SOURCE

在 FPGA 的顶层模块（system.org）中除了例化 SoC 的顶层（n100\_soc\_top）之外，主要是使用 Xilinx 的 I/O Pad 单元例化顶层的 Pad。另外使用 Xilinx 的 MMCM 单元生成时钟。注意：SoC 的 Main Domain 使用的 MMCM 产生的高速时钟连接到 SoC 的 hfextclk, Always-On Domain 使用的是开发板上的低速实时时钟（32.768KHz）。

N100 系列内核配套 SoC 的顶层 I/O Pad 经过 FPGA 的约束文件（nuclei-master.xdc）进行约束使之连接到 FPGA 芯片外部的引脚上面（譬如将 JTAG I/O 约束到开发板的 MCU\_JTAG 插口引脚上）。

## 3.2. 生成 mcs 文件

在前文中介绍了 N100 系列内核项目的 SoC 整体架构和 Verilog RTL 代码，为了使得该 SoC 能够真正运行在 FPGA 硬件上，需要将其编译成为 bitstream 文件（用于烧录到 FPGA 中去）。可以使用如下步骤进行编译。

**// 步骤一：准备好自己的电脑环境，可以在公司的服务器环境中运行，如果是个人用户，推荐如下配置：**

**(1) 使用 VMware 虚拟机在个人电脑上安装虚拟的 Linux 操作系统。**

**(2) 由于 Linux 操作系统的版本众多，推荐使用 Ubuntu 16.04 版本的 Linux 操作系统**  
**有关如何安装 VMware 以及 Ubuntu 操作系统本文不做介绍，有关 Linux 的基本使用本文也不做介绍，请用户自行查阅资料学习。**

**// 步骤二：安装 Xilinx Vivado 软件(包括 XILINX USB JTAG 下载器驱动程序)至此虚拟机 Linux 操作系统中。有关如何安装 Xilinx Vivado 软件和下载器驱动本文不做介绍，请用户自行查阅资料了解。**

**// 步骤三：将 n100\_rls\_pkg 解压至本机 Linux 环境中。**

**// 步骤四：设置需要编译的 Core 的具体型号，使用如下命令：**

```
cd n100_rls_pkg/n100_cct/fpga
```

```
// 进入到 n100_rls_pkg/n100_cct 目录文件夹下面的 fpga 目录。
```

```
make install CORE=n100 FPGA_NAME=hbirdkit
```

```
// 运行该命令指明需要为 n100 内核进行编译，该命令会在 fpga 目录下生成一个
```

```
// install 子文件夹，在其中放置 Vivado 所需的脚本，且将脚本中的关键字设置为 n100。
```

**// 步骤五：生成 bitstream 文件或者 mcs 文件（推荐使用 mcs 文件），使用如下命令：**

```
make bit
```

```
// 运行该命令将调用 Vivado 软件对 Verilog RTL 进行编译生成 bitstream 文件
// 生成的 bitstream 文件名和路径为
// n100_rls_pkg/n100_cct/fpga/hbirdkit/obj/system.bit
// 该 bitstream 文件则可以使用 Vivado 软件的 Hardware Manager 功能将
// system.bit 烧录至 FPGA 中去。

// 熟悉 Vivado 和 Xilinx FPGA 使用的用户应该了解, bitstream 文件烧录到 FPGA 中
// 去之后 FPGA 不能掉电, 因为一旦掉电之后 FPGA 烧录的内容即丢失, 需要重新使用
// Vivado 的 Hardware Manager 进行烧录方能使用。
// 为了方便用户使用, FPGA 评估板可以将需要烧录的内容写入开发板上的
// Flash 中, 然后在每次 FPGA 上电之后通过硬件电路自动将需要加载的内容从外部的
// Flash 中读出并加载到 FPGA 之中 (该过程非常的快, 不影响用户使用)。由于 Flash
// 是非易失性的内存, 具有掉电后仍可保存的特性, 因此意味着将需要载入的内容写入
// Flash 后, 每次掉电后无需使用 Hardware Manager 人工重新烧录 (而是硬件电路
// 快速自动完成), 即等效于, FPGA 上电即可使用。

// 为了能够将烧录 FPGA 的内容写入 Flash 中, 需要生成 mcs 文件, 使用如下命令:
make mcs
// 运行该命令将调用 Vivado 软件对 Verilog RTL 进行编译生成 mcs 文件
// 生成的 mcs 文件名和路径为
// n100_rls_pkg/n100_cct/fpga/hbirdkit/obj/system.mcs
// 该 mcs 文件则可以使用 Vivado 软件的 Hardware Manager 功能将
// system.mcs 烧录至 FPGA 评估板中的 Flash 中去。
```

### 3.3. 将 mcs 文件烧写入 FPGA

关于如何将上节生成的 mcs 文件烧写入 FPGA 评估板, 请参见芯来科技官网开发板页面中的详细介绍文档链接:

[http://www.nucleisys.com/theme/package/Nuclei\\_FPGA\\_DebugKit\\_Intro.pdf](http://www.nucleisys.com/theme/package/Nuclei_FPGA_DebugKit_Intro.pdf)



## 4. 基于 FPGA 平台运行和调试软件示例

### 4.1. 配套基于 Linux 的 SDK

基于配套的 SoC，Nuclei N100 系列处理器内核提供基于 Linux 的软件开发套件（SDK，Software Development Kit）。可以使用此 SDK 在 FPGA 原型平台上进行嵌入式软件的开发，详细介绍请参见单独文档《Nuclei\_N100 系列 SDK 使用说明》。

### 4.2. 配套基于 Windows 的 IDE

基于配套的 SoC，提供基于 Windows 的图形化集成开发环境（IDE，Integrated Development Environment）。可以使用此 IDE 在 FPGA 原型平台上进行嵌入式软件的开发，详细介绍请参见单独文档《Nuclei\_N100 系列 IDE 使用说明》。