

电 子 科 技 大 学

嵌入式智能计算研究团队

珊瑚-II 入门引导

ACORAL-II INTRODUCTION



版本号 1.0

修订历史

版本号	内容	日期	负责人
0.1	开始编写，修改 latex 模板，确定大纲	2022.05.15	王彬浩
1.0	第一版 aCoral-II 入门引导编写完成	2022.07.10	胡博文

目 录

第一章 概述.....	1
1.1 珊瑚（aCoral）简介	1
1.2 aCoral 项目成员	1
1.2.1 aCoral 早期版本项目成员	1
1.2.2 aCoral-II 项目成员	3
第二章 aCoral-II 结构	4
2.1 aCoral-II 文件结构	4
第三章 快速上手	6
3.1 环境搭建.....	6
3.2 软件操作.....	6
3.2.1 新建工程.....	6
3.2.2 配置 vivado 工程.....	7
3.2.3 配置 SDK 工程	10

第一章 概述

1.1 珊瑚（aCoral）简介

珊瑚（aCoral）是电子科技大学信息与软件工程学院嵌入式智能计算研究团队开发的一款嵌入式实时操作系统,具有开源、高可配、高扩展性的特点，其主要平台有 mini2440、pb11mpcore、stm32 和 zynq7020 等等。

珊瑚（aCoral）目前拥有单核（aCoral-I）和多核（aCoral-II）两个版本。本文档将介绍珊瑚操作系统的多核版本 aCoral-II。

aCoral-II 基于正点原子的领航者平台，平台所使用的 Soc 为 zynq7020。zynq7020 是 xilinx 公司推出的一款集成了双 Cortex-A9 及单 FPGA 的拥有强大性能的 Soc。正点原子使用此 Soc 推出的领航者平台涵盖了众多资源，包括但不限于 DDR3、UART、以太网、QSPI、SD 卡、音频接口、USB 等等。xilinx 公司还为 zynq7020 提供了充分的支持，其中包括功能强大的 IDE(vivado、sdk)、完善的底层硬件库以及完善的接口化的交叉编译调试链。

1.2 aCoral 项目成员

1.2.1 aCoral 早期版本项目成员

成员姓名	主要贡献
廖勇	项目创建人及总负责人
申建晶	内核框架设计及实现，GUI 系统
闫志强	内核线程交互开发及 TCP/IP 协议栈
孔帅帅	多核支持及中断系统
高攀	文件系统开发，H.264 在 ARM11 的多核优化
陈旭东	多核调度、实时性确保、多核实时控制
刘晓翔	开发环境及驱动模块
杨茂林	强实时调度算法研究及实现
张国梁	操作系统移植
王小溪	系统测试及性能确保、应用程序

魏守峰	系统测试及性能确保、应用程序
任艳伟	内核实时性确保
程潜	内核实时性确保
程勇明	内核实时性确保
汪琳玫	功耗管理
周强	驱动模型设计及自有图形系统开发
许斌	操作系统配置工具开发
袁霞	调度算法
江维	可信调度、功耗管理
李波	操作系统移植
张海斌	多核中断支持
钟太聪	多核调试器
彭东脉	多核调试器
Mugundhan balaji	H.264 多核支持
Subhajit Banerjee Purnapatra	多核调试器
孙康	内存管理
王云飞	内存管理
郭治蛟	操作系统移植
韩炫	操作系统移植
文秀春	操作系统移植
刘坚	操作系统移植
龚俊儒	操作系统移植
李天华	操作系统移植
郭文生	多核形式化验证
刘洋	应用程序
熊光泽	技术指导
桑楠	技术指导
雷航	技术指导
罗蕾	技术指导

李允	技术指导
陈丽蓉	技术指导

1.2.2 aCoral-II 项目成员

成员姓名	主要贡献
杨茂林	项目负责人
胡博文	代码重构、文档编写、多核开发

第二章 aCoral-II 结构

2.1 aCoral-II 文件结构

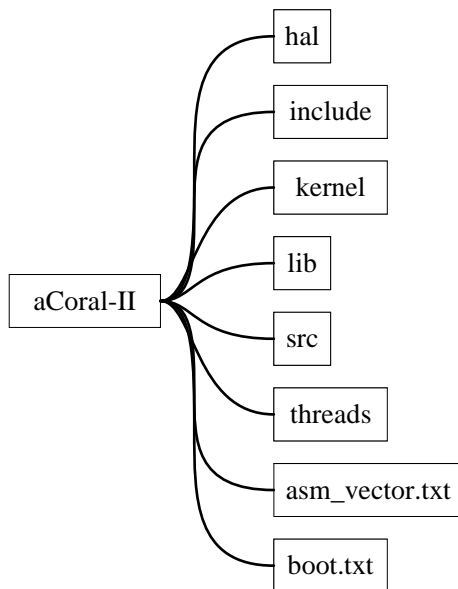


图 2-1 aCoral-II 文件结构

aCoral-II 由于使用了 Xilinx 官方的 IDE 以及库文件，所以除了基本的六个文件夹之外，还有两个基于官方库文件的更改，下面是具体文件结构介绍：

(1) hal 文件夹

hal 文件夹主要包含了基于底层硬件的代码，是 aCoral-II 的抽象层，把底层硬件细节抽象给内核层以及更上层使用。

(2) include 文件夹

include 文件夹里面包含了数据类型文件，aCoral-II 配置文件，以及给上层使用的总头文件。

(3) kernel 文件夹

kernel 文件夹包含了 aCoral-II 内核的代码，其中包括基本的内存、线程、线程通信、链表等等机制。

(4) lib 文件夹

lib 文件夹主要包含了与 aCoral-II 内核内容较无关的机制代码，比如 print。

(5) src 文件夹

src 文件夹里面主要就是 c 语言入口文件 main.c，然后还包括了 aCoral-II 的链

接脚本等。

(6) threads 文件夹

threads 文件夹包含了一些具体线程的文件。

(7) asm_vector.txt 文件

asm_vector.txt 文件是更改的官方库文件，具体使用时需要粘贴替换 acoral_bsp/ps7_cortexa9_0/libsrc/standalone_v6_8/src/asm_vector.S 的内容。

(8) boot.txt 文件

boot.txt 文件是更改的官方库文件，具体使用时需要粘贴替换 acoral_bsp/ps7_cortexa9_0/libsrc/standalone_v6_8/src/boot.S 的内容。

第三章 快速上手

3.1 环境搭建

由于使用的开发平台为 Xilinx 公司推出的 zynq-7020，所以具体环境就是 Xilinx 公司的 vivado 以及 SDK。操作系统方面 windows 和 linux 都可以，安装过程网上都有资料，也比较简单，在此不再赘述。目前使用到的版本为 vivado2018.3，安装 vivado 的时候会同时安装 SDK。

3.2 软件操作

3.2.1 新建工程

首先打开 vivado 软件，点击 create project 后设置工程路径（自行定义），然后勾选见图 3-1。

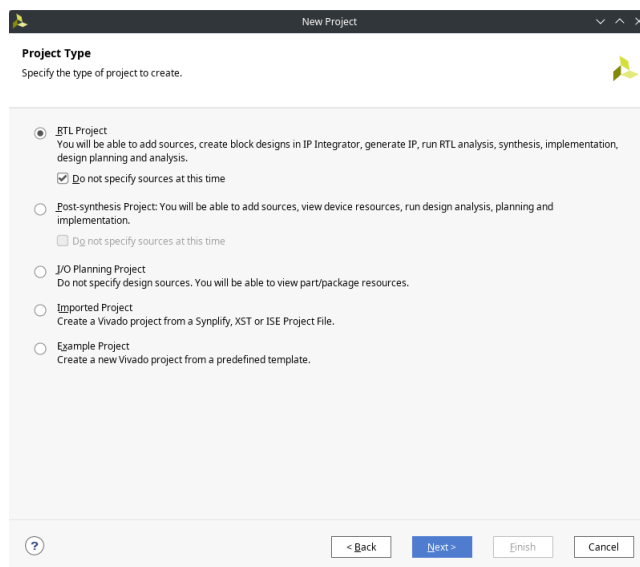


图 3-1 新建工程 1

然后搜索型号为 xc7z020clg400-2，选中后 next 应该显示与图 3-2 相符界面，接下来点击 finish 完成创建。

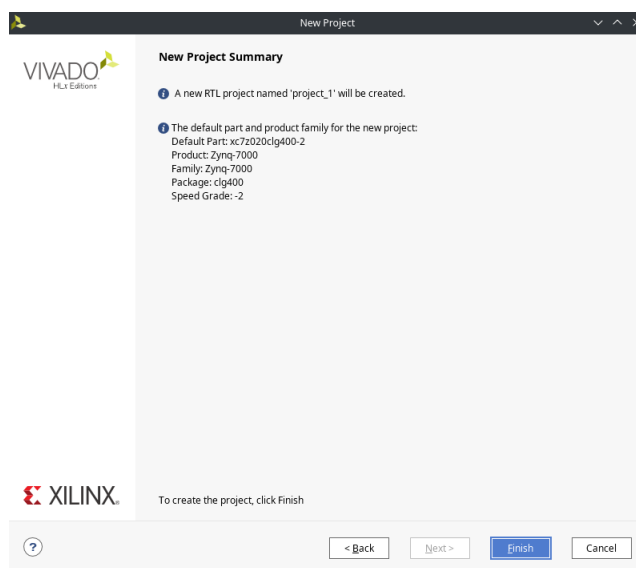


图 3-2 新建工程 2

3.2.2 配置 vivado 工程

之后就来到了 vivado 工程页面，左边就是按照操作流程来安排的 Flow Navigator，首先点击 IP INTEGRATOR 中的 Create Block Design，然后给 design 文件命名。

命名完成后右边出现了 Diagram，点击 Diagram 中间的 + 号，然后如图 3-3 搜索 zynq 添加 ZYNQ7 Processing System。

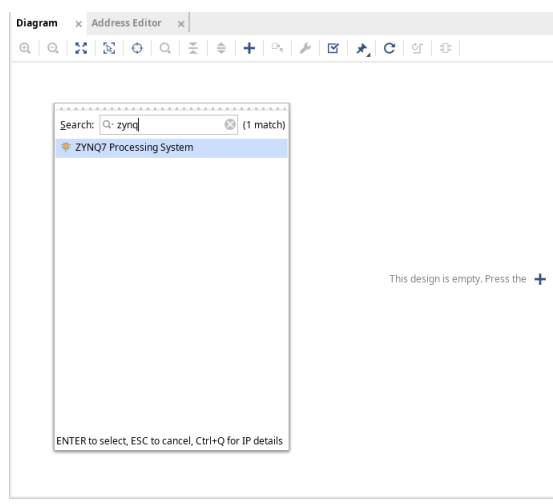


图 3-3 配置 vivado 工程 1

接下来双击添加的 ZYNQ7 Processing System IP，按照图 3-4 来配置引脚，图

中没包括的部分还需要勾选 GPIO MIO。

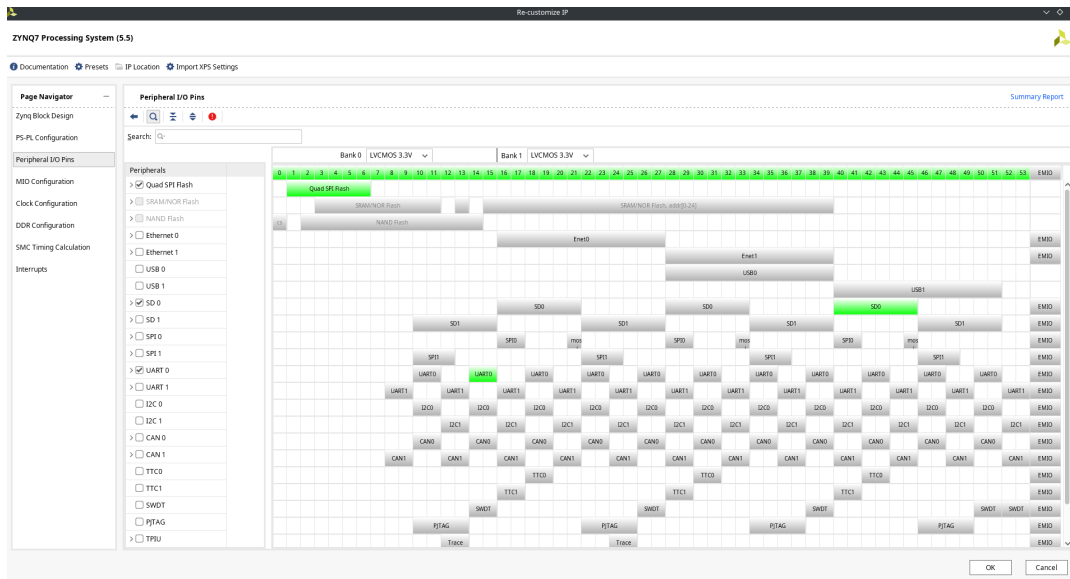


图 3-4 配置 vivado 工程 2

由于操作系统部分和 PL 也就是 FPGA 部分无关，所以去掉 PL 的时钟和复位线等等，具体操作如图 3-5 和图 3-6 以及图 3-7，然后点击右下角的 ok。

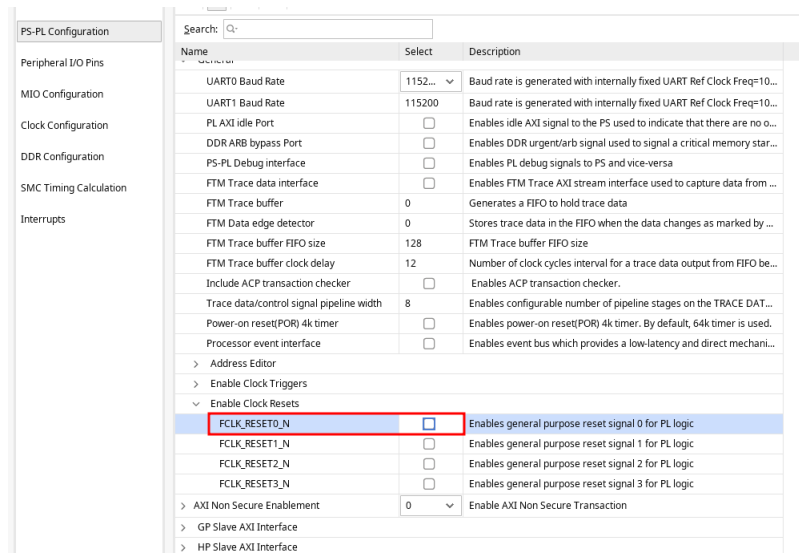


图 3-5 配置 vivado 工程 3

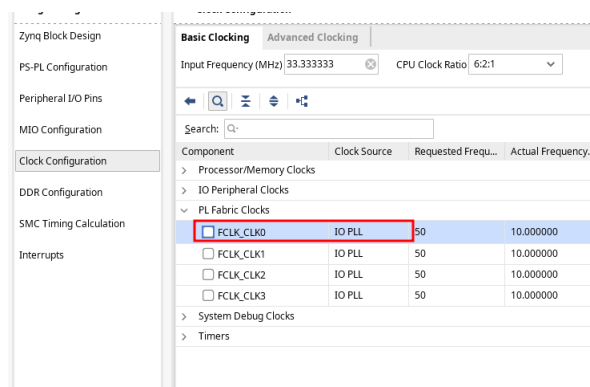


图 3-6 配置 vivado 工程 4

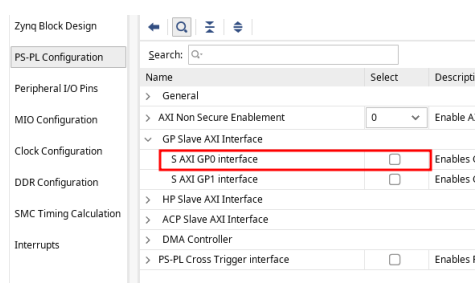


图 3-7 配置 vivado 工程 5

接下来点击 Run Block Automation, OK, 如图 3-8。右键 source 中的 design (应该为你自己之前的命名), 如图 3-9, 点击 Generate Output Products..., ok 后还是右键 design, 点击 Create HDL Wrapper..., 勾选让 vivado 控制, 然后 ok。

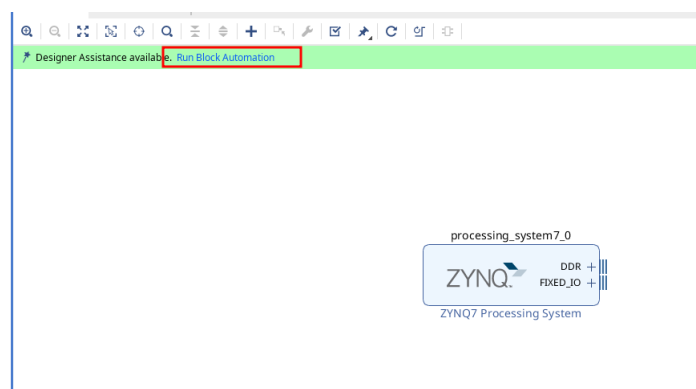


图 3-8 配置 vivado 工程 6



图 3-9 配置 vivado 工程 7

然后点击左边 Flow Navigator 中的 Run Implementation，一直点 ok。接下来点击左上角 File->Export->Export Hardware，不勾选 include bitstream，最后点击左上角 File->Launch SDK，ok。

3.2.3 配置 SDK 工程

点击左上角 File->New->Application Project，命名然后直接 finish，如图 3-10。

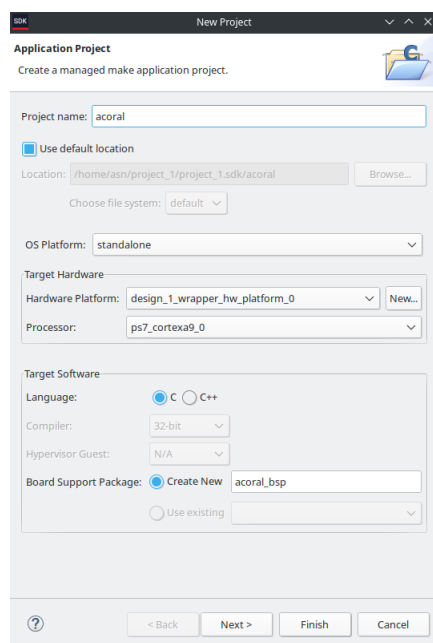


图 3-10 配置 SDK 工程 1

然后把 aCoral-II 的六个文件夹覆盖放到 SDK 的 acoral 工程文件夹中，目录应该为 you_project_name/you_project_name.sdk/acoral，还需要把两个 txt 文件的内容

复制粘贴到 `you_project_name/you_project_name.sdk/acoral_bsp` 中，具体路径前面有。

F5 刷新下工程，然后在更改下 SDK 的默认编码为 utf-8，设置 tab 被空格替换，而且为 4 个空格，改动方法见<http://xinzero.com/vivado-tab-to-space.html>，最后右键 acoral 选择属性 Properties，添加头文件路径。具体做法可以自行搜索，不再赘述。

至此可以进行编译下载了。

以上只是基本步骤，详情可以参考正点原子的 FPGA 领航者 V2 的嵌入式开发视频BV1Hq4y1u7qH。