

目录

目录	1
概述	1
CCU简介	2
CCU功能实现	2
硬件及软件环境的准备	2
Loader程序的实现	2
用户FP逻辑的实现	4
用户MCU程序实现	7
CCU功能实现	9
结束语	11
关于Capital Microelectronics	12

概述

为了使用户可以清楚的了解京微雅格（Capital Microelectronics）可配置片上系统（CSoC）之 Astro系列芯片的应用方法，在文档的开头，将简要介绍Astro系列芯片的主要内部资源和运行模式。

Astro系列芯片的主要内部资源：

- 1K FP Logic Cells
- 64KB FP配置数据存储器（OTP型，可分为3块）
- 2块9Kb可配置双端口EMB
- 1 FP JTAG配置接口
- 100MIPS高速8051 MCU
- 16KB 8051数据存储器
- 64KB 8051代码存储器（OTP型，可分为2块）
- 1 8051 片上JTAG调试接口（OCDS）
- 512KB FP和8051公用SPI Flash（可分为8块）

Astro系列芯片的主要运行模式：

- 小模式
不外挂存储器，FP配置数据和8051代码（小于2KB）存储于SPI Flash中，系统上电FP配置数据先从SPI Flash中导出配置FP包括EMB的初始化数据，完成后8051代码直接EMB中开始运行（参考文档：[AN-Astro系列之EMB用做8051的代码存储器](#)）。
主要用于8051代码较少时，调试生产两便，成本也比较低。
- 调试模式
外挂SRAM存储器，FP Bootloader配置数据和FP用户程序配置数据和8051用户代码分段存储于SPI Flash中，系统上电先从SPI Flash中导出FP BootLoder配置数据（内含Loader部分）配置FP包括EMB的初始化数据，完成后SRAMLoader程序将搬运8051用户代码到外挂

SRAM中，之后再从SPI Flash中导入FP用户程序并重新配置FP，8051代码在外挂SRAM中开始运行（参考文档：[AN-Astro系列之SRAM用做8051的代码存储器](#)，即本文档要介绍的内容）。

主要用于8051代码超过2KB调试时用，当然也可用于生产，只是外挂SRAM增加了成本。

- 最终生产模式
不外挂存储器，8051代码存储于OTP存储器中，FP配置数据存于OTP或SPI FLASH中。因为OTP存储器是一次性的，所以不能用于调试阶段，一般用于产品最终定型后，使用OTP存储器可以获得最优的性能和最低的成本。

本文档主要介绍Capital Microelectronics Astro系列CSoC芯片之SRAM用做8051的代码存储器的方法，用于系统开发调试阶段或最终产品。现在就通过一个简单的CCU（Compare Capture Unit）例子来使用户对Astro系列芯片外挂SRAM模式的开发有一个直观的了解。

CCU简介

Capital Microelectronics Astro系列CSoC芯片内置CCU并不像传统MCU那样具有模拟输入引脚，通过比较两个输入引脚来产生输出信号，这是由Astro芯片本身的特性（在Astro芯片的内置8051和外部引脚之间是数字逻辑层来让用户实现自由逻辑）所决定，在Astro芯片中，CCU是Timer2的一个附属功能模块，通过对比CCU和Timer2寄存器中的值，用来直接控制P1[3..0]输出PWM信号，用来实现电机、音频、灯光类控制。

CCU功能实现

下面将分部介绍 CCU 的实现方法，BootLoader 是此实现方法中的一个步骤，程序分为 3 部分：Loader、用户 FP、用户 MCU (8051)，下面就将介绍此 3 部分的实现过程。

硬件及软件环境的准备

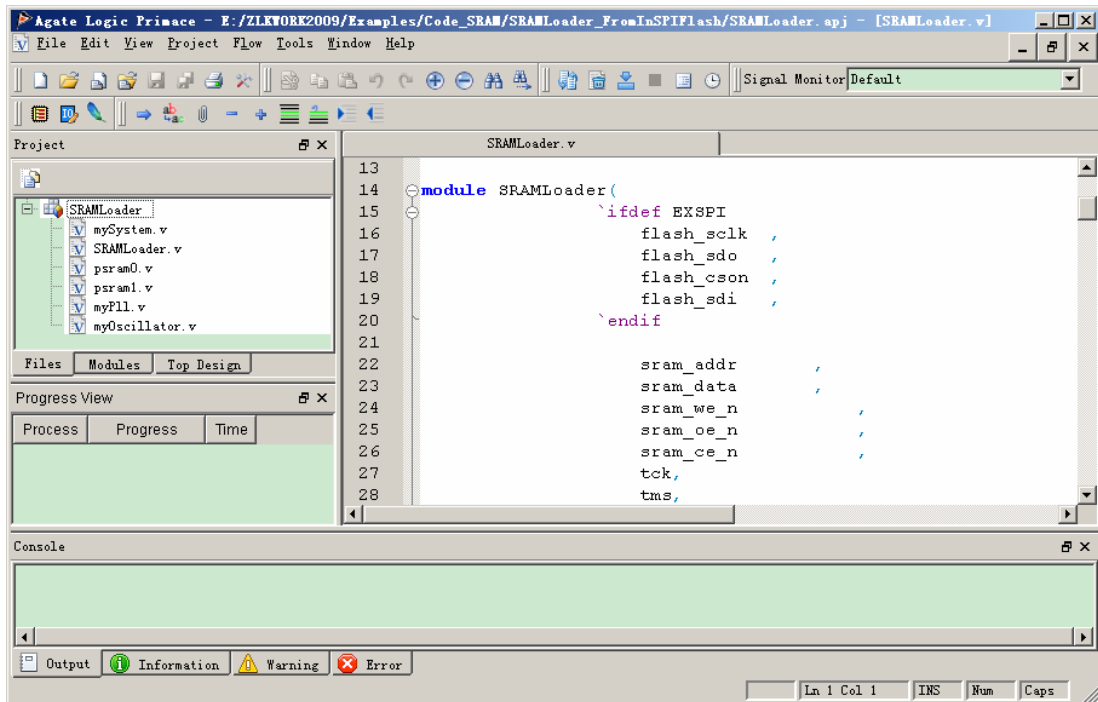
1. CME Astro-EVB-L144 开发板；
2. CME JTAG 调试器及其相应驱动；
3. CME Primace3.0（或其以上版本）EDA 开发环境；
4. Keil uVision2（或其以上版本）；
5. 所有以上硬件的获得及软件环境的安装及使用方法请参考相应的文档，在此不再赘述。

Loader程序的实现

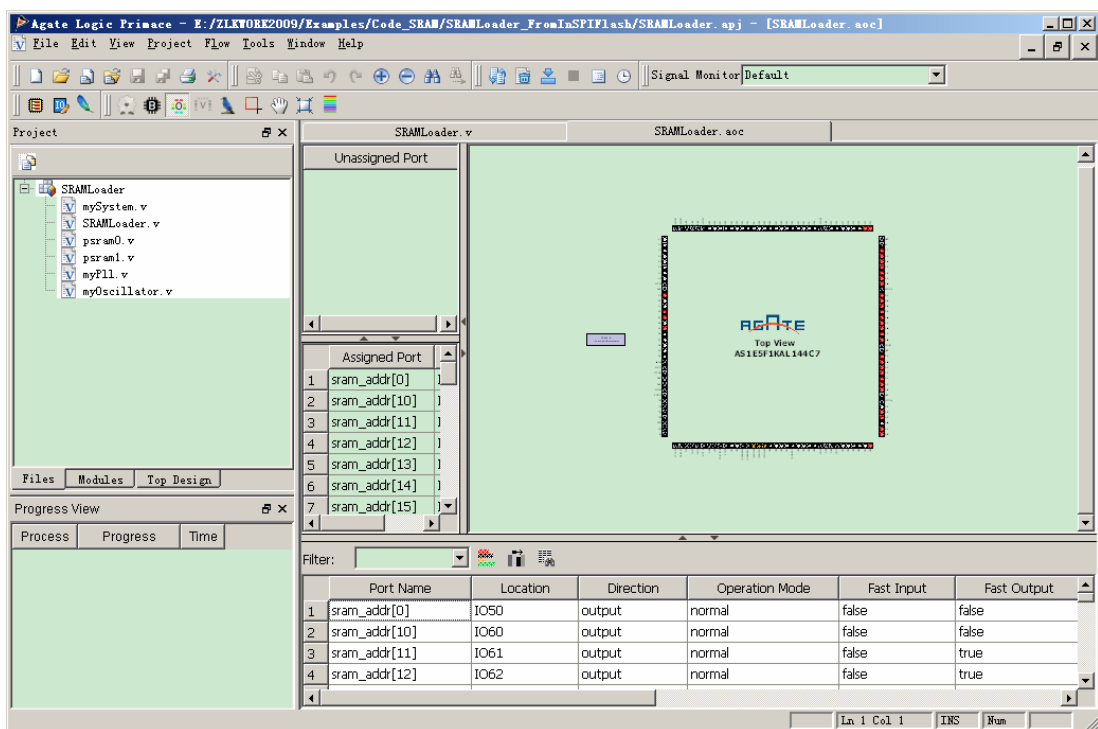
1. 在 Primace3.0 EDA 环境中直接打开 SRAMLoader_FPInSPIFlash 工程（此工程已经比较固定，只需视情况做一下修改）。

- 1) 若是 Astro 芯片带有内部 SPI FLASH 就注释掉 `define EXSPI 如下: `//define EXSPI if 否则就要定义, 如: `define EXSPI
- 2) 设置 SRAM_ADDR_WIDTH 值与外挂的 SRAM 总线宽度一致。

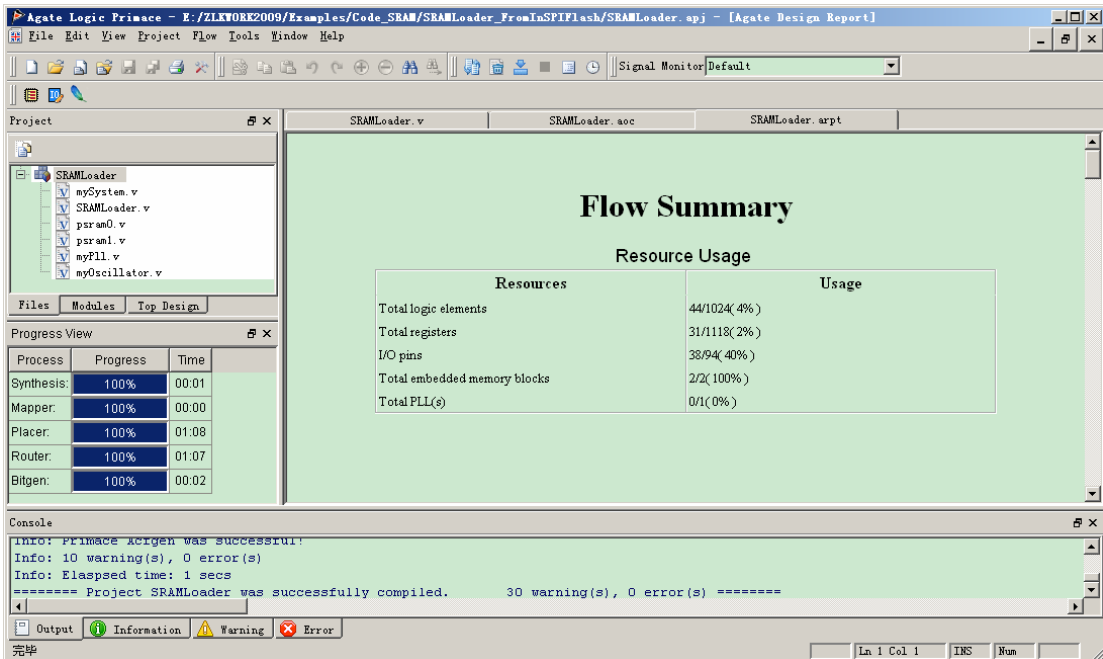
一般用户可在此基础上经过修改IO分配后直接作为Loader使用, 请参照[AN-Astro 系列之ISC应用指南](#)文档) 如下图:



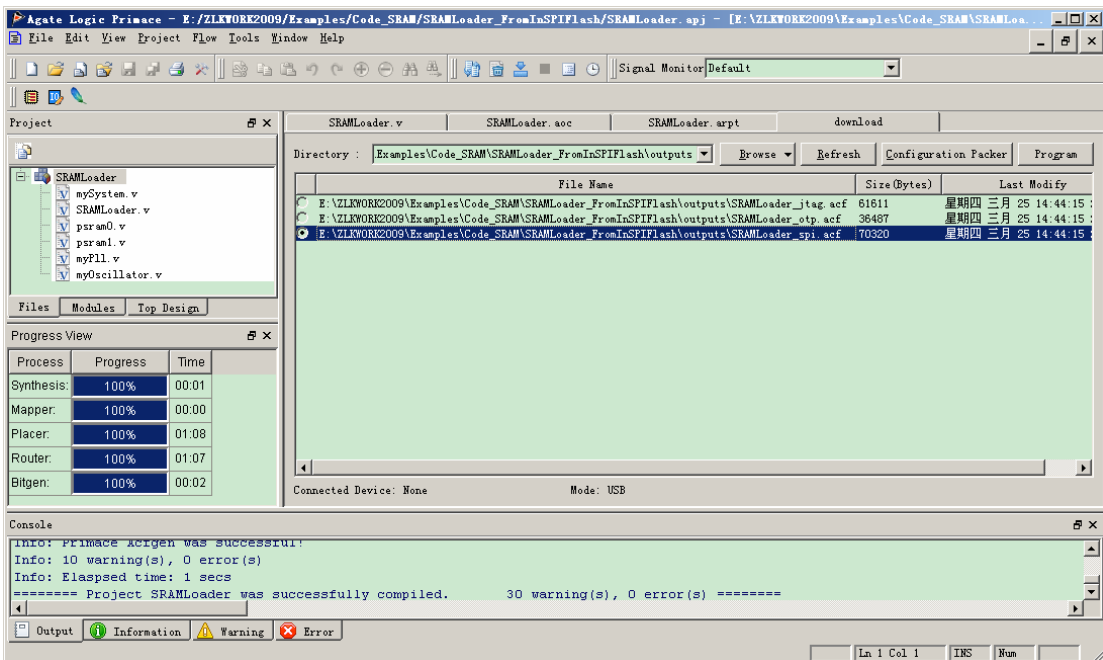
2. 选【Tools】菜单【IO Editor】打开 IO 编辑器, 把 SRAM IO 重新分配到用户 PCB 板的对应 IO 上, 如下图:



3. 选【Flow】菜单【Rerun Project】重新编译此工程（注意 Loader 中初始化 EMB 所用 load0.dat 和 load1.dat 已放入此工程目录的 src 目录），如下图：



4. 编译产生烧写文件 SRAMLoader_spi.acf 备用，如下图：

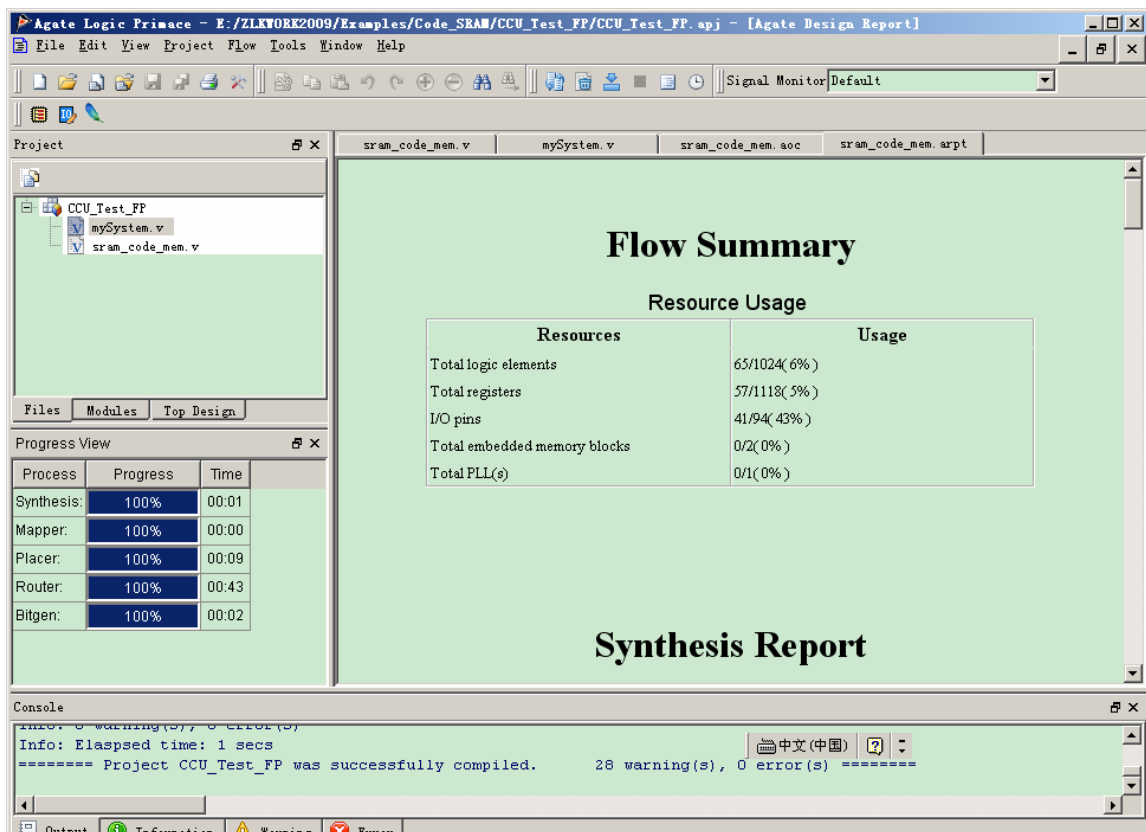
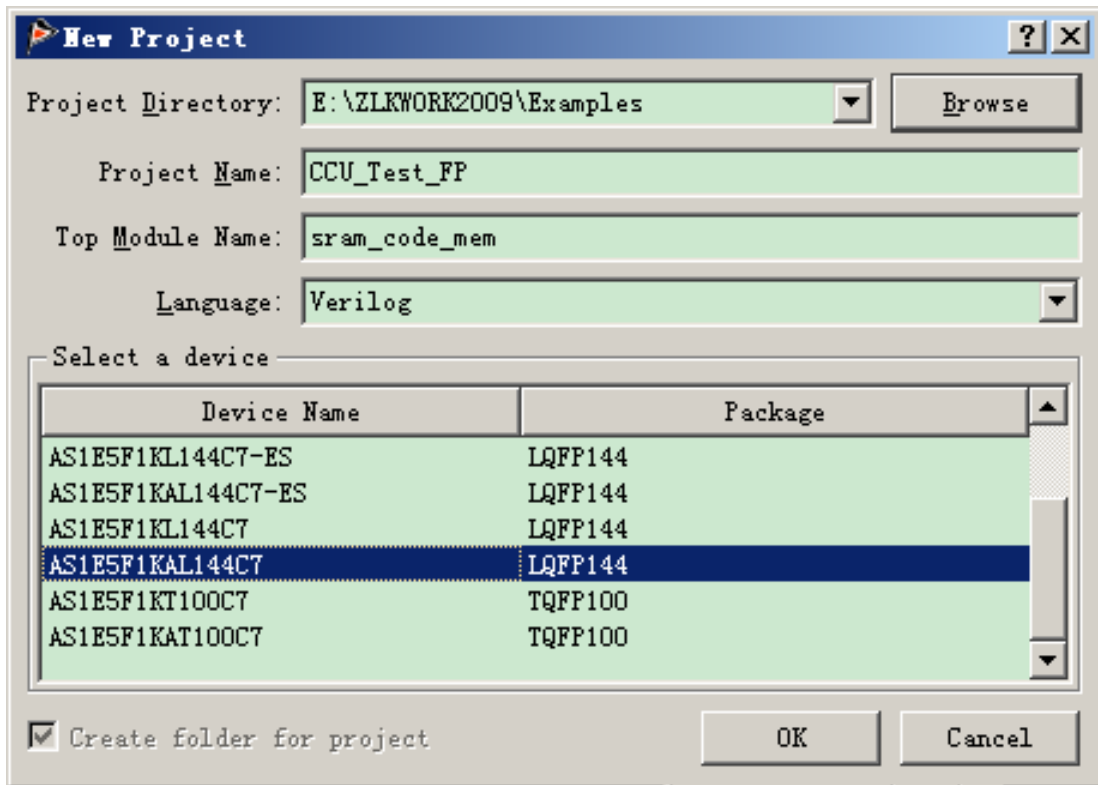


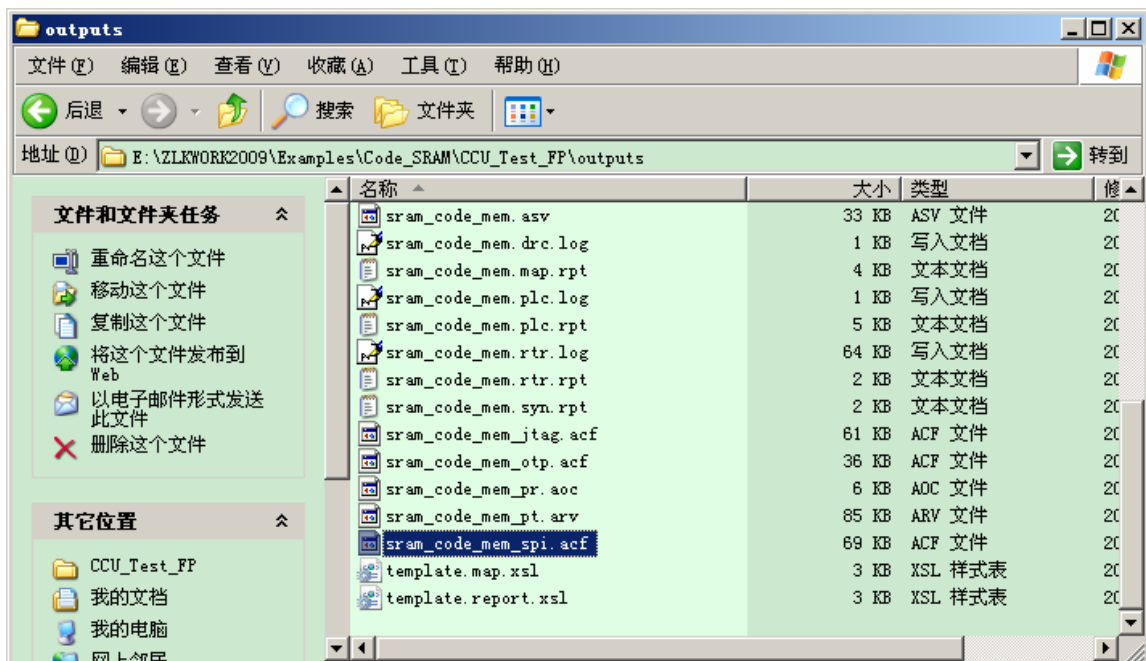
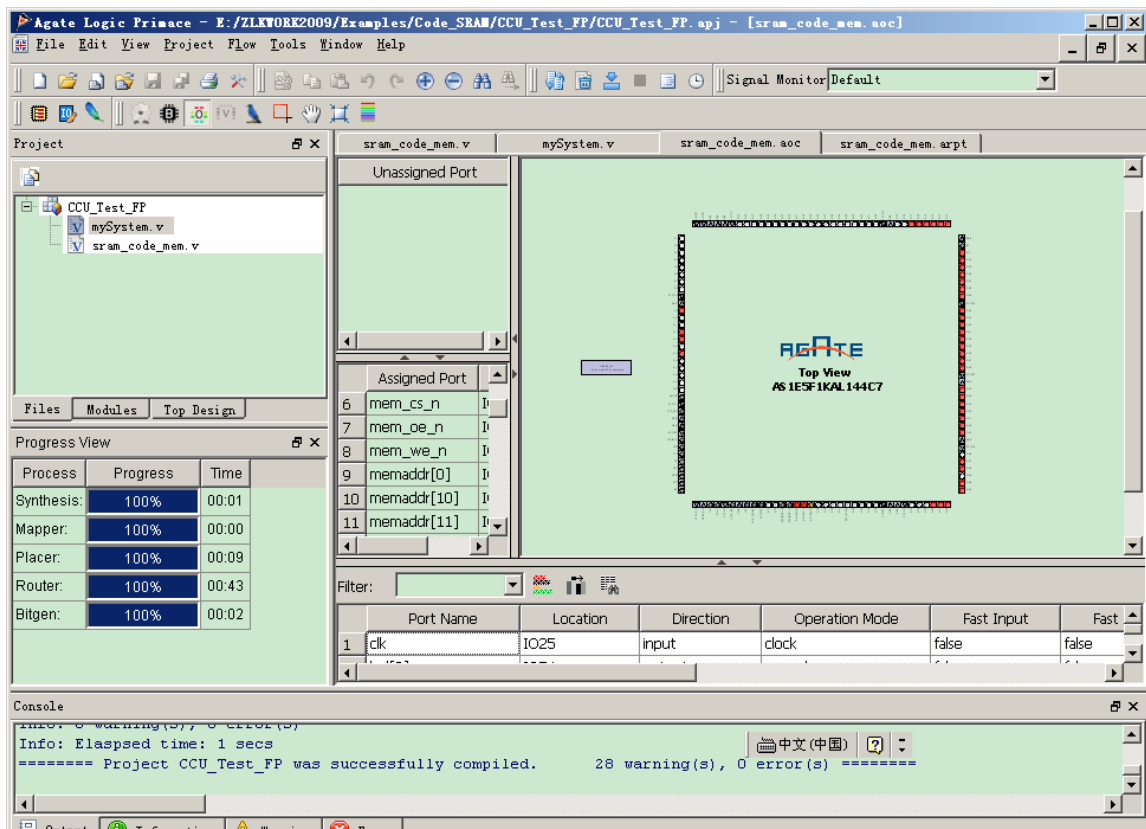
5. 至此Loader程序实现成功结束，更加详细的Loader介绍及应用请参考[AN-Astro系列之ISC应用指南](#)文档。

用户FP逻辑的实现

1. 用户 FP 逻辑的实现步骤和 Loader 的实现类似，同样经历在 Primace3.0 EDA 环境中建立 CCU_Test_FP 工程，书写 Verilog 源代码（调用 system wizard，根据 wizard 的指导产生扩展外部 SRAM 的源代码），编译工程，分配引脚（IO Editor 步骤也可以在 Verilog 源代码书写

【Run Mapper】后就进行), 重编译产生.acf 烧写文件的过程, 如下图:

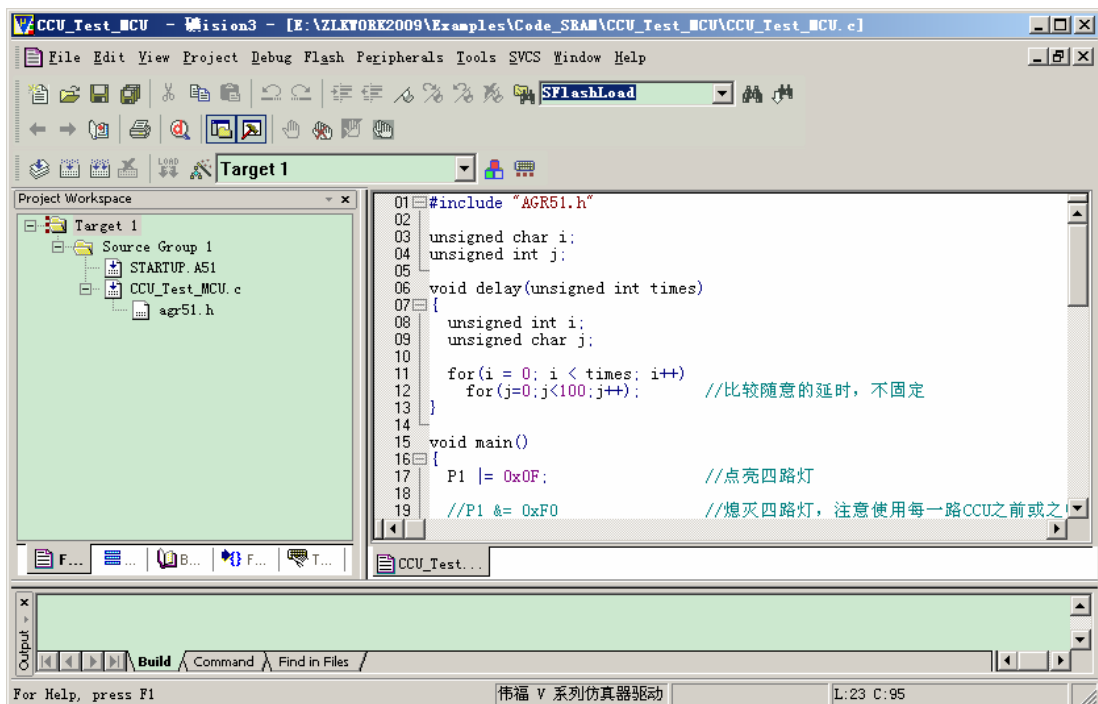
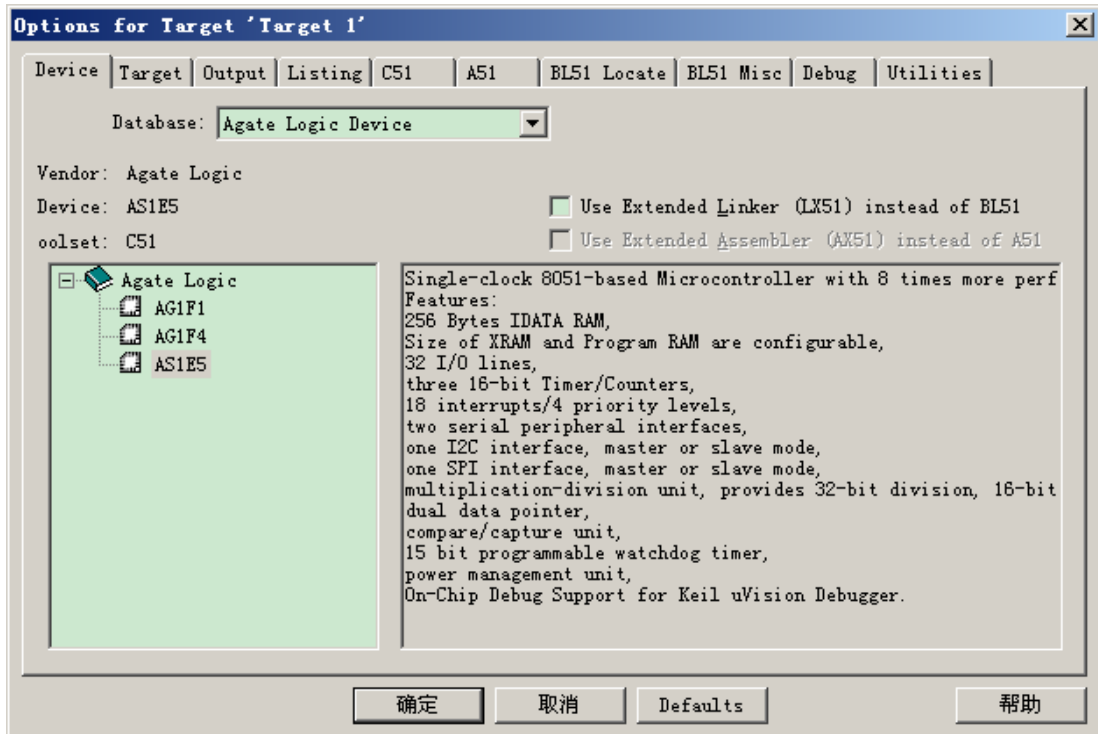




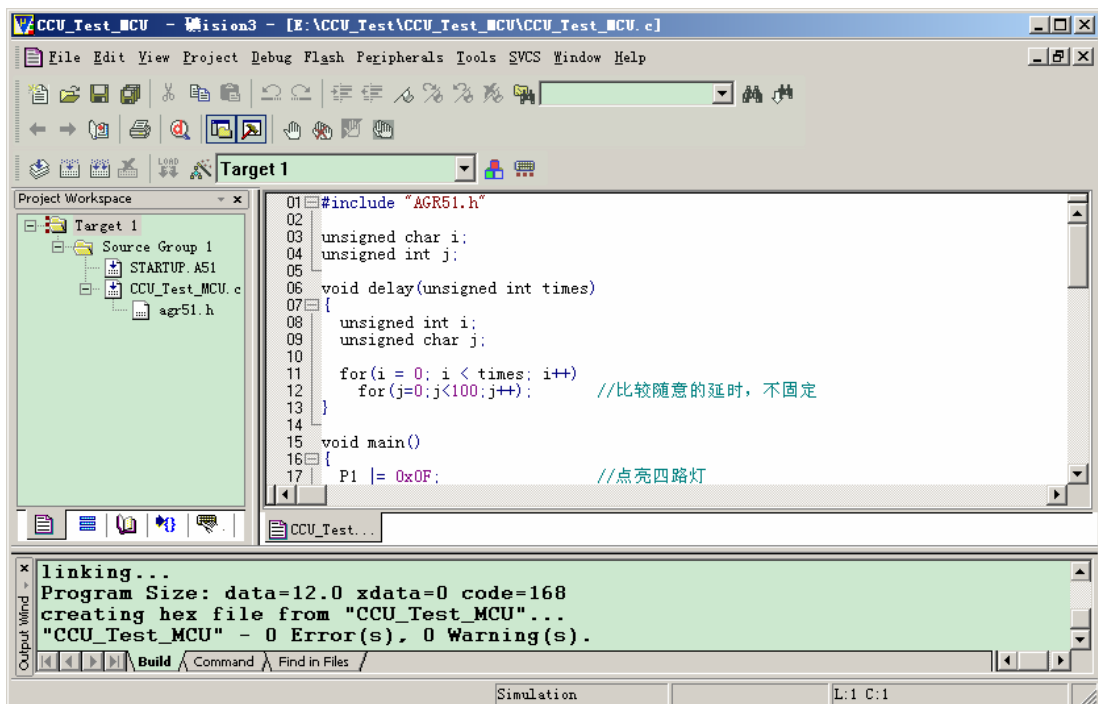
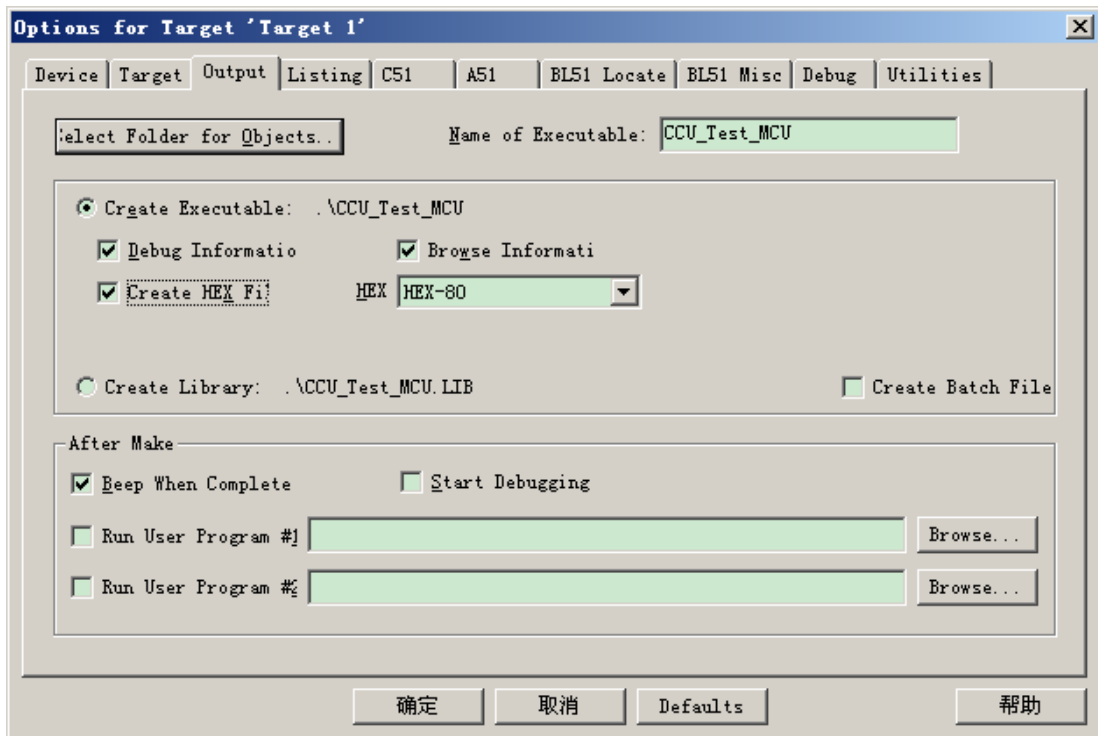
- 此工程主要是依据 Astro-EVB-L144 上的 IO 实际分配位置将 SRAM 引脚和 8051 代码控制线互联，并将内部 8051 引脚对应到 LED 上面的简单逻辑，具体请看相应的.v 文件。用户 FP 逻辑的实现至此成功结束。

用户MCU程序实现

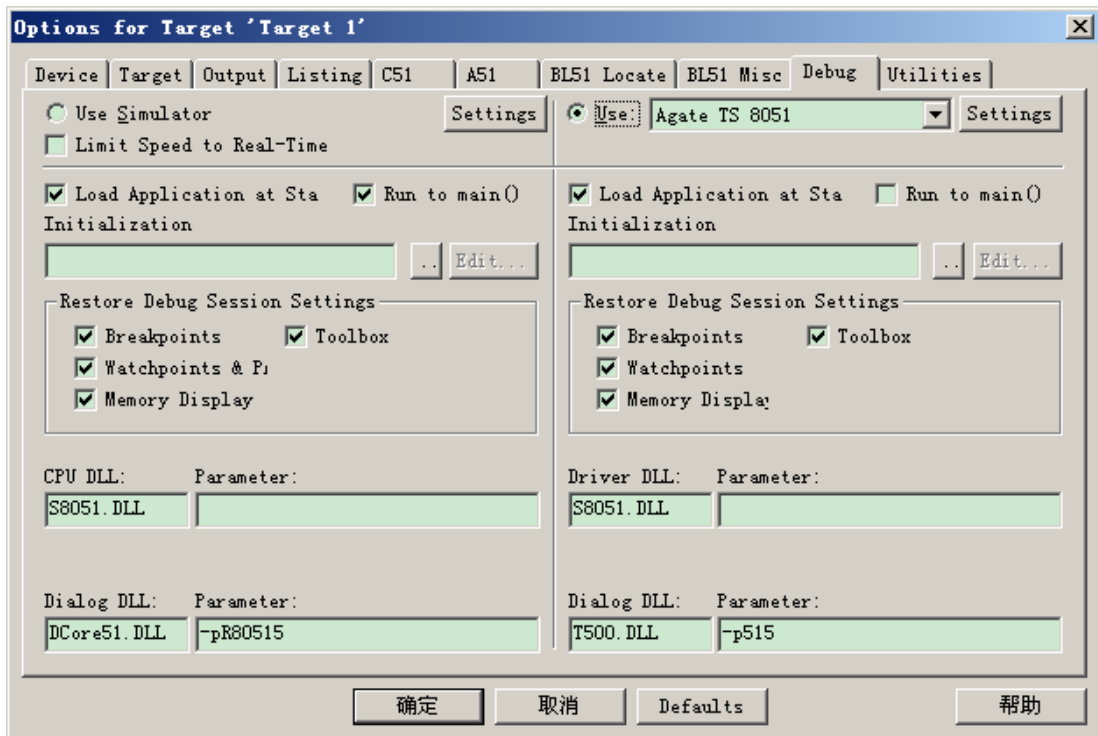
1. Astor 系列芯片内置 8051 MCU 程序的实现和普通 8051 程序的实现并无差别，在 Keil 中新建工程（如 *CCU_Test_MCU*），Database 选择 Agate Logic Device（在 *Primace* 软件安装完成后会出现这个选项），元件选 AS1E5，如下图：



工程配置中选择【Create HEX File】后成功编译，即可产生相应的.Hex 文件。



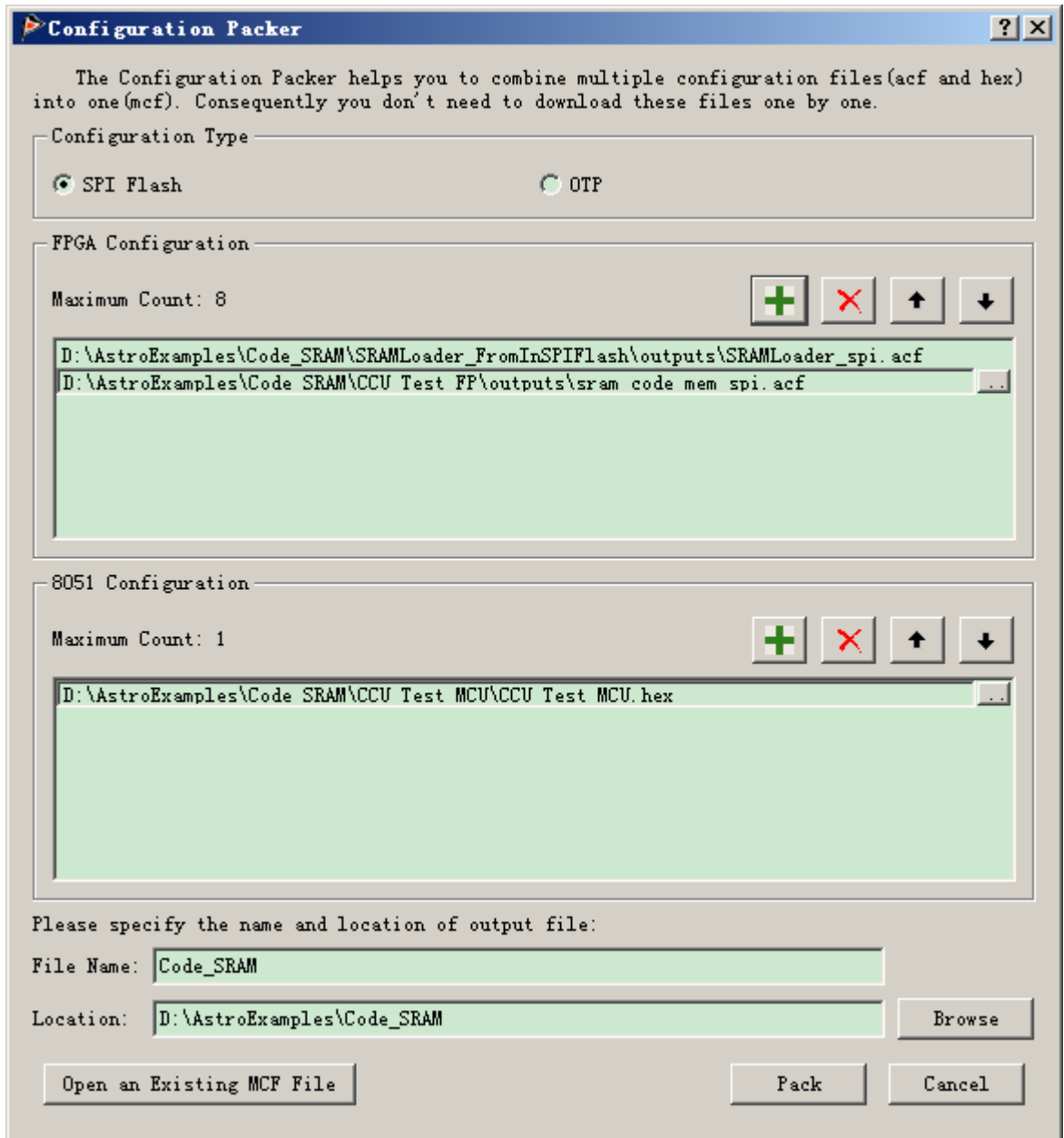
- 在此说明一下，Astro 系列芯片内置 8051 支持 JTAG Debug 方式，此时需要相应的硬件逻辑和 CME 8051 agdi 驱动支持，具体请参照压缩包中 agdi 文件夹中的使用说明进行配置使用，具体文档请咨询 Capital Microelectronics 相关人员。Astro-EVB-L144 支持 JTAG Debug，使用时应将 JTAG 下载线插入 8051-FPGA 插槽，同时应该注意的是 JTAG Debug 时序和实际运行时序是不一致的，实际中运行的要快，此点对时序要求严格的应用需要关注，如下图：



3. 成功产生.Hex 文件后，用户 MCU 程序即成功实现，CCU MCU 源代码请参考压缩包中的 CCU_Test_MCU.c 文件。

CCU功能实现

1. 有了需要的 Loader、FP、MCU 烧写文件后，我们只需要把这 3 个文件下载到片内 SPI Flash 中即可，在此我们使用配置向导来将要下载的文件打包到一个文件：点【Tools】菜单选【Configuration Packer】启动打包，选择要烧写的文件，如下图：



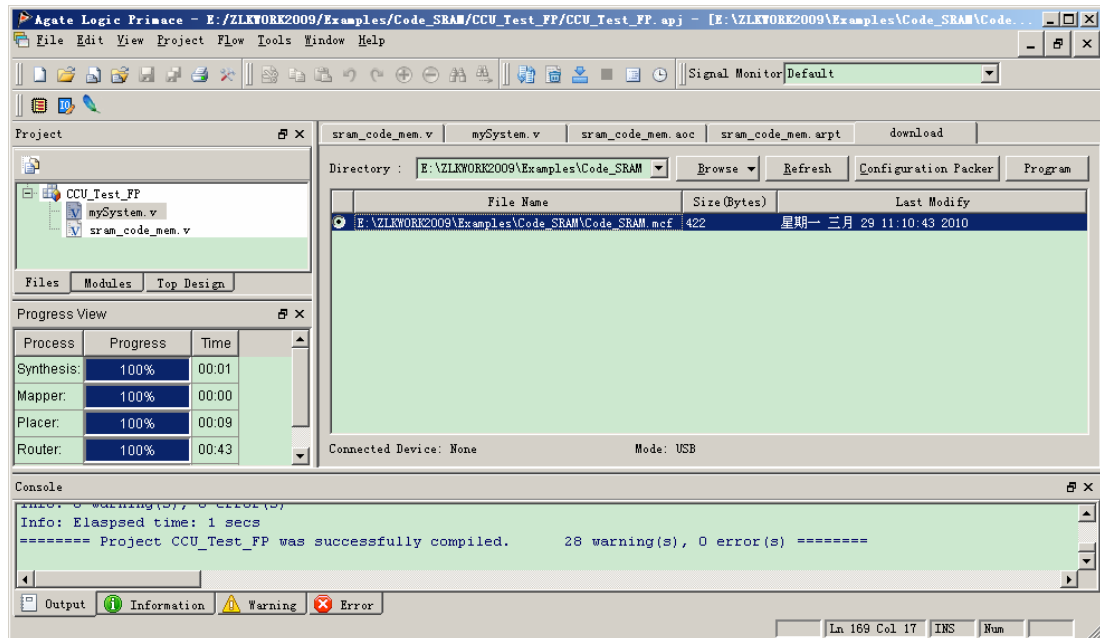
点击【Pack】即可在相应目录产生 Code_SRAM.mcf 文件，内容如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<multiConfig device="AS1E5F1KAL144C7" storage="SPI" bootAddr="0">
<acf addrIdx="0" ref="D:\AstroExamples\Code_SRAM\SRAMLoader_FromInSPIFlash\outputs\SRAMLoader_spi.acf"/>
<acf addrIdx="1" ref="D:\AstroExamples\Code_SRAM\CCU_Test_FP\outputs\sram_code_mem_spi.acf"/>
<hex addrIdx="0" ref="D:\AstroExamples\Code_SRAM\CCU_Test_MCU\CCU_Test_MCU.hex"/>
</multiConfig>
```

首行和最后一行不用管，第二行指出了使用的实际芯片型号和存储地址为 SPI Flash，第三行开头的“addrIdx=0”指出了 Loader 程序在 SPI 中的地址，第四行开头的“addrIdx=1”指出了用户 FP 程序在 SPI 中的地址，第五行开头的“addrIdx=0”指出了用户 MCU 程序在 SPI 中的地址，这几个地址是不可以变化的，如要变化，需要更改 Loader 程序的 8051 代码

部分，详细方法仍然参考 [AN-Astro 系列之 ISC 应用指南](#) 文档。

2. 在 Primace EDA 环境下【Flow】菜单选【Download】进入下载界面，点【Browse】到 Code_RAM.mcf 所在的目录选择它，连接 FP JTAG 连接线，点【Program】即可把代码写入芯片内部 SPI Flash 中，如下图：



3. 重启系统，CCU_Test 程序将运行，正确现象应该为顶板 D2 LED 灯由暗闪烁变亮，再由亮闪烁变暗。

结束语

本文主要通过一个 CCU_Test 例程来介绍 Astro 系列芯片在存储器模式内部 SPI Flash+外部 SRAM 状况下的开发过程，希望籍此能让读者对 Astro 系列芯片的开发有一个直观的了解。

Capital Microelectronics Astro系列CSoc芯片同时拥有 8051 内核和FPGA结构，因此可以有多种灵活和复杂的应用，更多信息请参阅Capital Microelectronics公司网站www.capital-micro.com或直接咨询相关市场人员。

关于Capital Microelectronics

Capital Microelectronics 是全球 APGA 技术的首创者和领导者，致力于提供可编程成逻辑器件、可配置片上系统（CSoC）、集成 EDA 工具、IP 及 IC 设计服务，目标定位于通信设备、工业控制系统、消费类电子产品等多种应用类市场领域。

技术支持

电话: +86 10 82150100

E-Mail: support@capital-micro.com

网址: www.capital-micro.com

©2010-2011 Capital Microelectronics, Inc. 版权所有。未经本公司书面许可，任何单位及个人不得以任何形式或任何方式（电子、机械、电磁、光学、化学、手工或其他任何方式）对本文档的任何部分进行复制、传播、转录、存放于其他公开检索系统或者翻译成任何一种语言或者计算机语言。本文档所提及的所有其它商标均为其各自所有人持有。