

使用 CCS 的 Energy Trace 优化 msp430 的低功耗运行

技术有限公司

文档编号	RD-0011-A0
关键字	CCS, Energy Trace, 低功耗
摘要	本文简要介绍了使用 CCSv6.0 的低功耗测量评估工具 Energy Trace 对 msp430 芯片的低功耗运行进行优化。

目 录

1 概述.....	1
2 Energia 项目开发流程简介.....	2
2.1. Energy Trace 介绍.....	2
2.2. Energy Trace 的设置.....	2
2.3. Energy Trace 的运行.....	3
2.4. Energy Trace 的低功耗优化对比分析.....	6

插图索引

图 2-1	Energy Trace 设置界面.....	3
图 2-2	Project 的低功耗调试模式设置界面.....	3
图 2-3	设置 Energy Trace 的分析时间为 10s 截图.....	5
图 2-4	Energy Trace 运行 10s 截图.....	5
图 2-5	Energy Trace 对比分析低功耗 LPM3 模式与持续 Active 模式.....	6
图 2-6	Energy Trace 对比分析低功耗 Power/Energy 和 States.....	7
图 2-7	Energy Trace 对比分析 PortB 初始化的 Power/Energy 和 States.....	7

表格索引

错误！未找到目录项。

1 概述

本文简要介绍了使用 CCSv6.0 的低功耗测量评估工具 Energy Trace 对 msp430 芯片的低功耗运行进行优化。。

硬件平台：TI-Launchpad FR5969

软件平台：CCS V6.0.1

2 Energia 项目开发流程简介

本章介绍 CCS 的 msp430 低功耗测量评估工具——Energy Trace, 并简要记录采用 Energy Trace 对 msp430 的低功耗运行进行优化的流程。

2.1. Energy Trace 介绍

Energy Trace 是 TI 对 msp430 系列芯片开发的, 利用支持 Energy Trace 功能的 Debugger, 可以在 CCS 的调试界面显示芯片的运行功耗, 从而帮助开发者对芯片的低功耗运行进行优化, 以下是官方文档描述:

“The EnergyTrace™ Technology is an energy-based code analysis tool set that is useful for measuring and viewing the application’ s energy profile and optimizing it for ultra-low power consumption.

MSP430 devices with built-in EnergyTrace+[CPU State]+[Peripheral States] (or in short EnergyTrace++™) support allow real-time monitoring of many internal device states while normally executing user program code. EnergyTrace++™ is supported on selected MSP430 devices and debuggers.

The EnergyTrace mode (barring the “++”) is the base of EnergyTrace™ Technology and enables analog energy measurement to determine the energy consumption of an application, without the possibility to correlate it to internal device information. The EnergyTrace mode is available for all MSP430 devices with selected debuggers. ”

简而言之, Energy Trace 是利用支持该功能的 Debugger (EzFET 等) 对目标芯片供电, 并测量供电输出电流 (以及电压) 大小, 从而计算功耗, 进而确定功耗优化选择。

而 Energy Trace++则是通过调试器对支持该功能的芯片 (不是所有 430 的芯片都支持), 通过读取芯片内部功耗模式、时钟、外设使能等信息, 从而估算芯片的功耗, 进而确定功耗优化选择。

2.2. Energy Trace 的设置

Energy Trace 需要在菜单栏的 Window->Preferences->Code Composer Studio->Advanced Tools->EnergyTrace™ Technology.里面设置如下图:

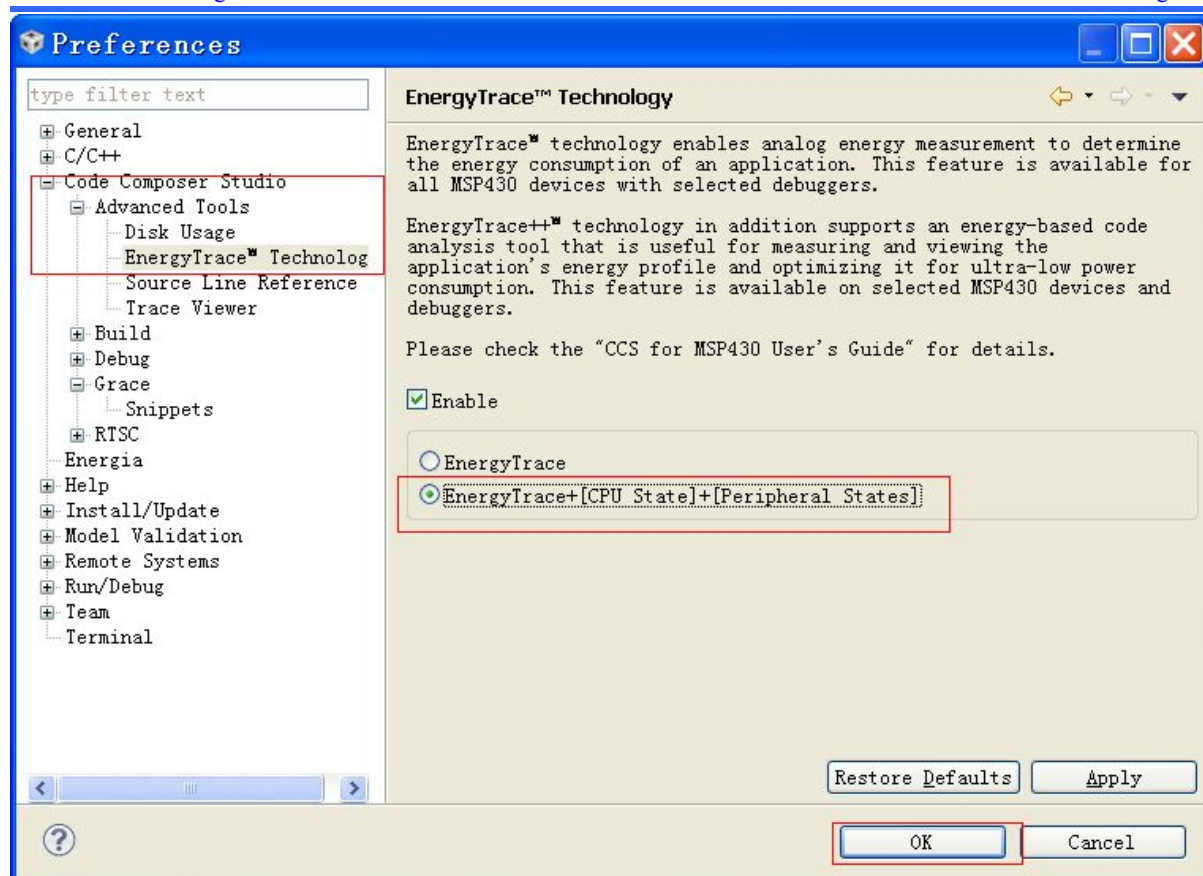


图 2-1 Energy Trace 设置界面

同时，对于建立的项目，需要勾选“超低功耗模式的调试”

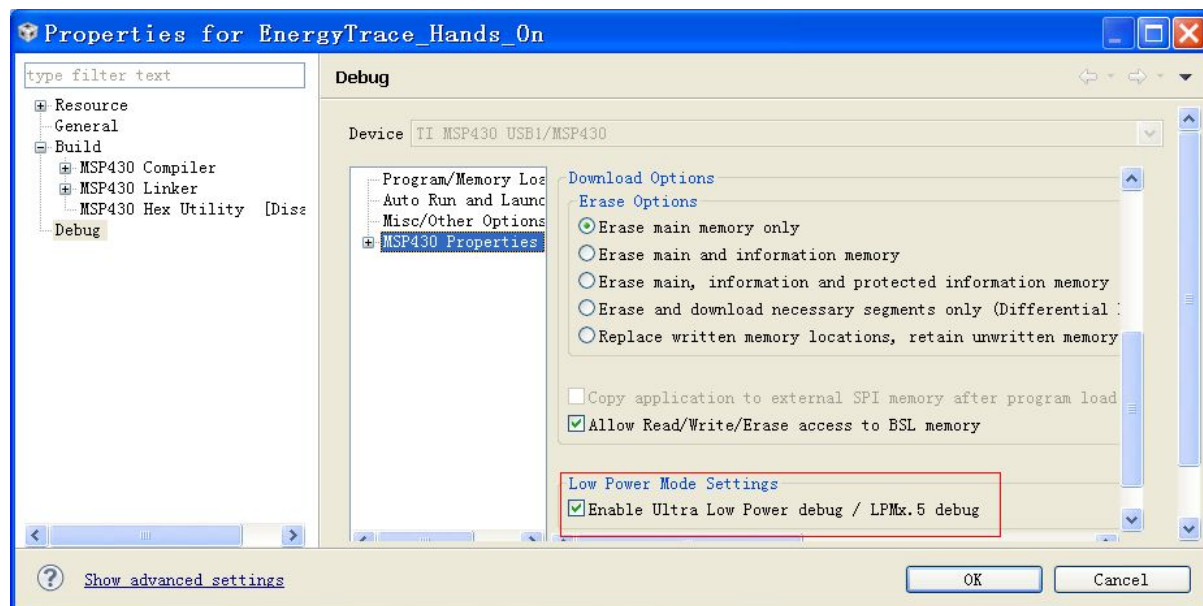


图 2-2 Project 的低功耗调试模式设置界面

2.3. Energy Trace 的运行

连接好 Launchpad，并安装好驱动（参考 Launchpad 用户手册，不再赘述），打开 EnergyTrace_Hands_On 的项目，右键项目，选择 Build Configurations->Set

Active->Insufficient, 则 Insufficient.c 导入项目, 代码如下:

```
/*
/
// MSP430FR59xx EnergyTrace Demo- High Energy Consumption Code
//
// Description: This code is intentionally written inefficiently in order
// to use an unnecessary amount of energy. The ULP Advisor and EnergyTrace
// are used to help identify the problem areas in the code to point out
// where changes can be made to increase efficiency.
//
// About every second, an ADC temperature sample is taken and the degrees
// Celsius and Fahrenheit are found using floating point calculations.
// The results are printed and transmitted through the UART.
// Some examples of the inefficient coding styles used are:
//     sprintf()
//     Floating point operations
//     Divide operations
//     Flag polling
//     Software delays
//     Floating port pins
//     No use of low-power modes
//     Counting up in loops
//
// B. Finch
// Texas Instruments Inc.
// June 2014
// Built with Code Composer Studio v6.0
#include <stdio.h>
#include <msp430.h>
#include <stdint.h>
.....
    UCA0CTLW0 &= ~UCSWRST;                // Enable eUSCI_A

    __no_operation();                      // SET A BREAKPOINT HERE

    while(1)
    {
        ....
        while(UCA0STATW & UCBUSY);
        __no_operation();                  // For debugger
    }
}
```

点击 Debug 后, 在调试界面的菜单栏选择 Windows->Show View->Others, 选择

Energy Trace，进入 Energy Trace 分析界面。选择测量记录时间为 10s，如下：

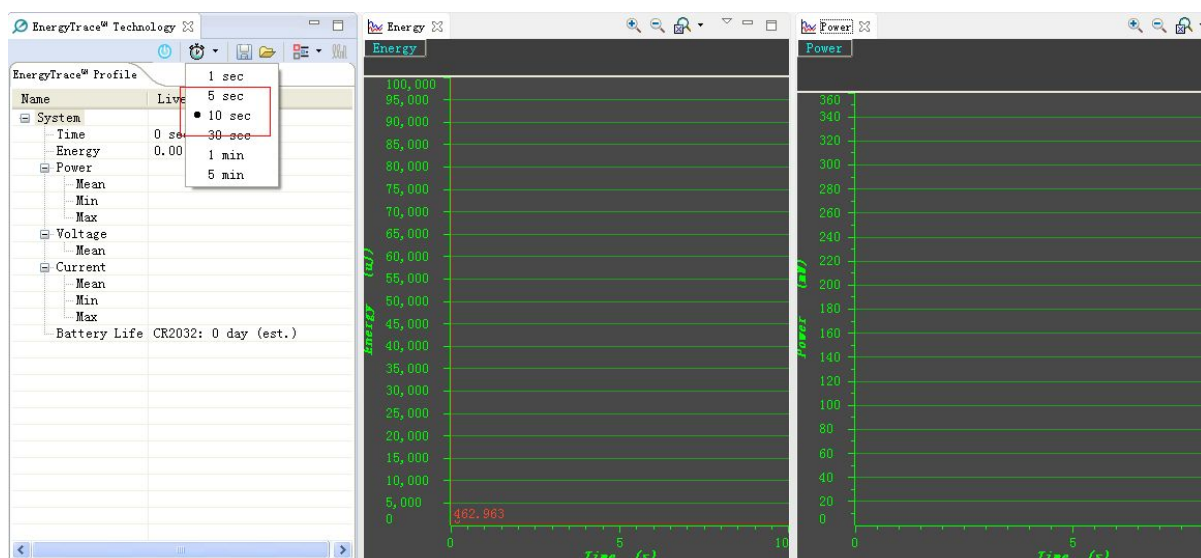


图 2-3 设置 Energy Trace 的分析时间为 10s 截图

Energy Trace 主要包括 Profile，以及 Energy Profile(Energy Trace++ Profile)、State 和 Power 四个标签页，以曲线形式表示功耗大小。Energy Trece Profile 对功率、电压、电流的最大值、平均值、最小值计算，同时估算采用 CR2032 纽扣电池供电的持续运行时间。而 Energy Trace++ Profile 则记录各个时间点 CPU 的运行模式、各个外设、时钟等信息。

点击 run，则 Energy Trace 自动记录芯片在 10s 内的功耗曲线。

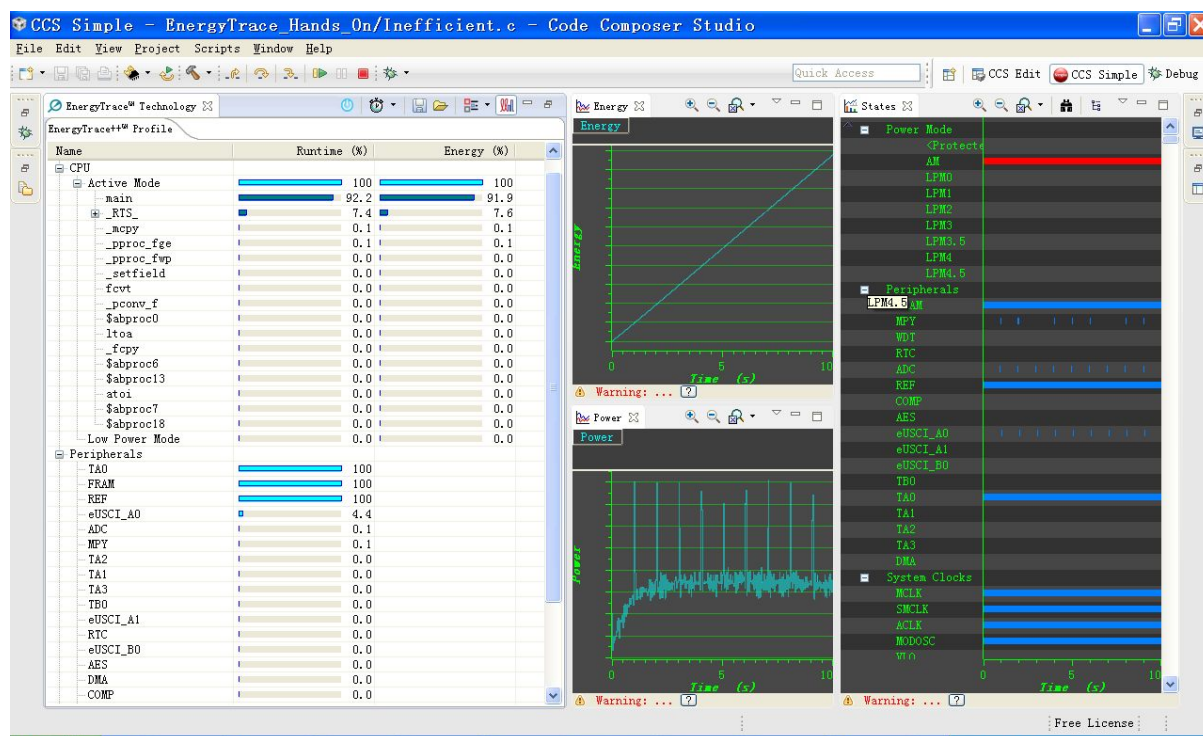




图 2-4 Energy Trace 运行 10s 截图

点击 Profile 标签页的保存按钮 ，将结果保存到一个文件。

2.4. Energy Trace 的低功耗优化对比分析

结束调试，在 Project Explorer 里选择项目的 Build Configuration 选择 Most Efficient，并编译调试，同样保存 10s 的 Energy Trace 记录，然后点击 Profile 标签页的导入按钮 ，导入刚才保存的 Inefficient 的记录，两者对比如下：

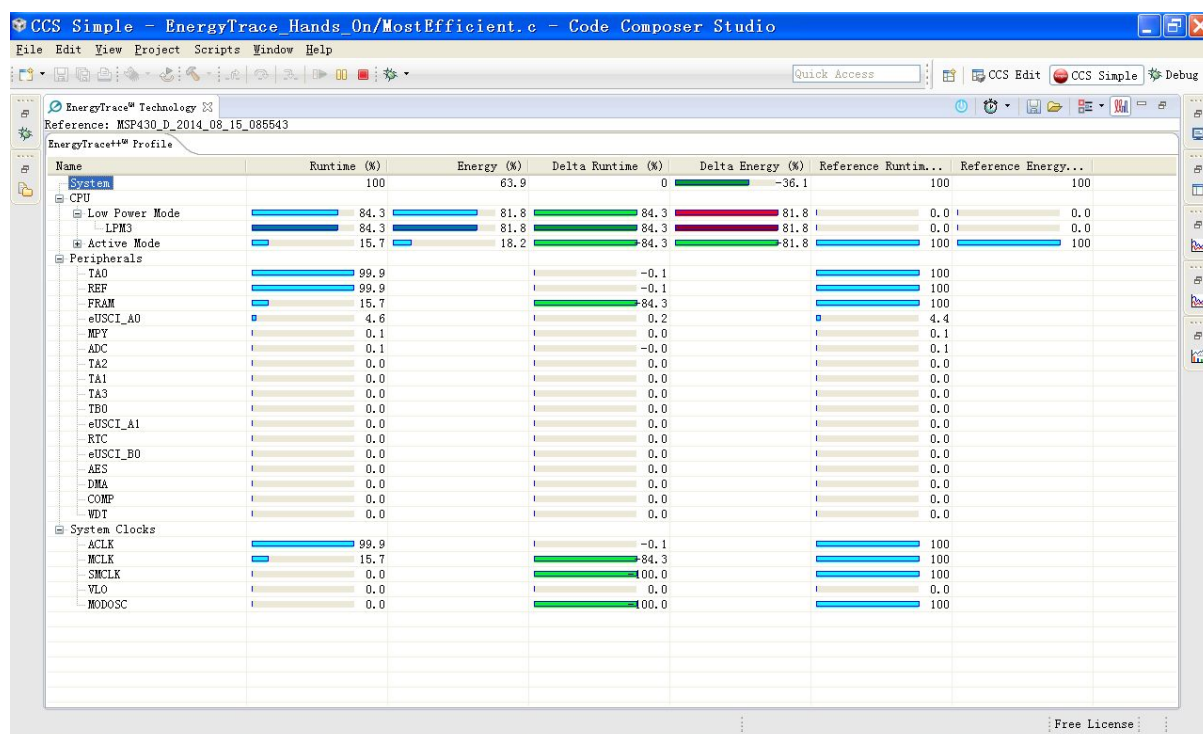


图 2-5 Energy Trace 对比分析低功耗 LPM3 模式与持续 Active 模式

在运行模式的对比中，当前的 Runtime 和 Energy 大部分集中在 LPM3 中，而且 FRAM、MCLK 都非常少，而 Reference Runtime 和 Reference Energy 都在 Active Mode 里面，可以直观地对比出不同低功耗运行模式下所占用的时间比例。

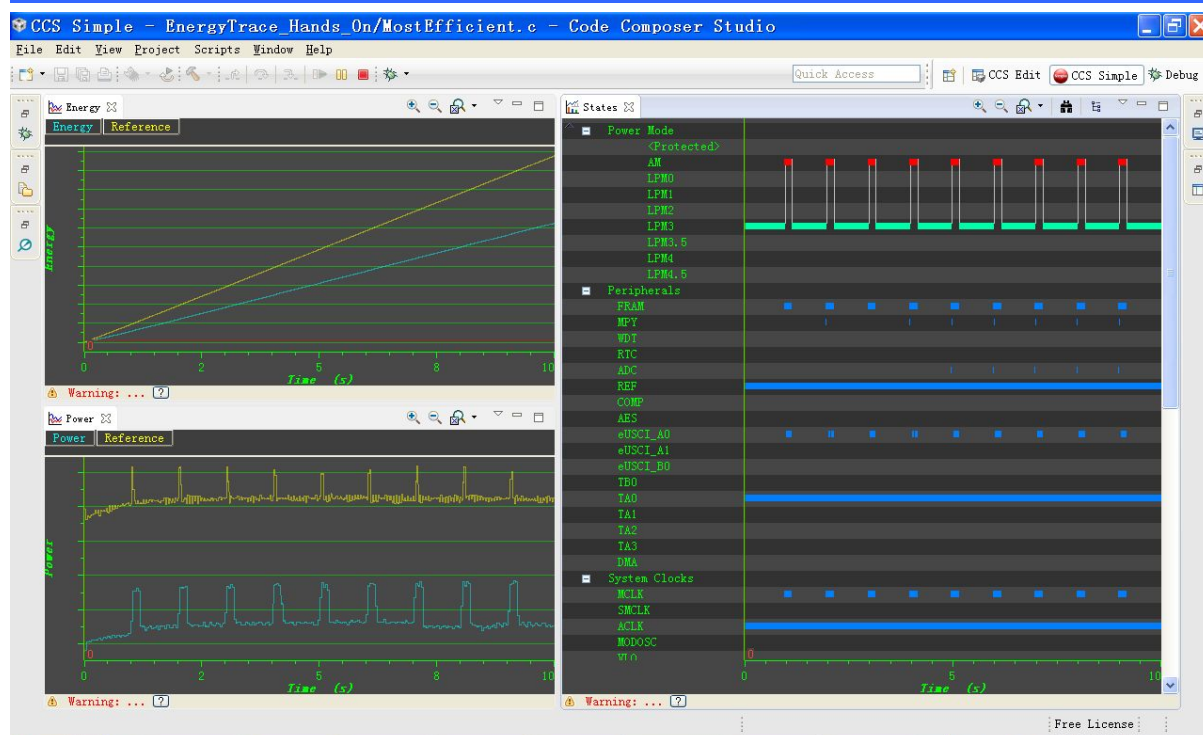


图 2-6 Energy Trace 对比分析低功耗 Power/Energy 和 States

上图则可以清楚地看到 Inefficeint 的 Reference Energy 和 Reference Power(黄色曲线)比 Most Efficient 的运行模式等状态在 LPM3 和 ActiveMode 之间切换,降低了功耗。

最后还有 Build Configuration 为 Most Efficient Solution 的运行对比如下:

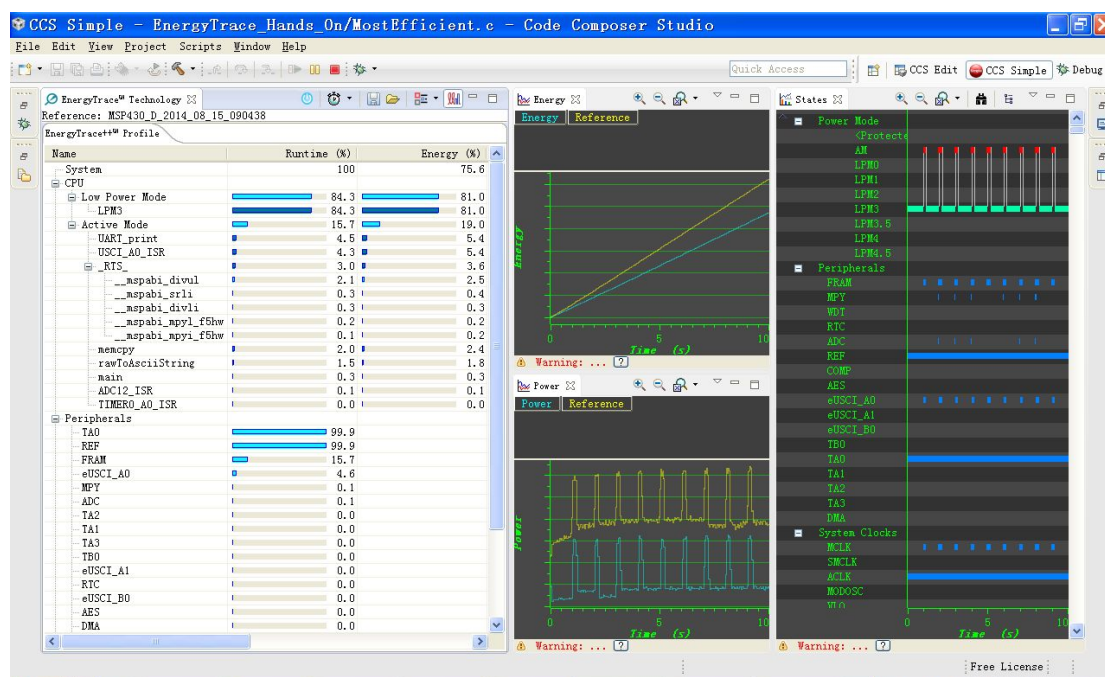


图 2-7 Energy Trace 对比分析 PortB 初始化的 Power/Energy 和 States

采用对 PortB 初始化后,可进一步降低功耗。

版本历史（Revision History）

版本号	发布时间	内容
A0	2014-08-17	初次发布。