

RT1050 采用不同存储器的启动时间

原文: <https://community.nxp.com/docs/DOC-341316>

我曾做过一个小项目来了解当 RT1050 从不同类型的存储器来启动时,性能如何。我使用 SDK 中的 LED_Blinky 代码作为基础,并在 EVKB 板上运行了一些测试。下面将介绍我收集的数据,以及更多的测试过程的细节。

测试过程

启动时间的定义是从处理器刚开始上电到它执行 main() 函数中的第一行代码的时间。

使用示波器(Tektronix TDS, 2014)测量了从 POR_B*信号的上升沿到以下两点之间的时间:

- 1、FlexSPI_CS 发生(ROM 首次读取 FlexSPI)**
- 2、应用程序代码中的 GPIO 口切换(代码执行信号的开始)。***

*POR_B 信号可通过 J26-1 接插头进行检测。

**FlexSPI_CS 信号可通过电路板上的一个小上拉电阻 R356 检测到。在这个电阻的旁边焊接了一小根电线,并连接到示波器的探针。

***所使用的 GPIO 引脚与连接到 USER_LED(低电平有效)的引脚相同。这个引脚可以通过 J22-5 接插头来检测。

TP 2、3、4 和 5 可以用于将示波器的探头接地。所有这些都是 EVKB 评估板上做的。

下面是关于本次测试过程中的几个值得注意的点:

- 1、本报告主要强调 POR_B 信号的上升沿与执行第一行代码之间的时间。然而,在电路板接通电源和 POR_B 系统上电之间仍有一段时间。这是电源器件的问题,且会根据用户的应用和设计而变化。有鉴于此,这份报告不会对此给予太大的重视。
- 2、应用程序的第一部分代码其实是配置处理器的几个引脚。只有在这些引脚被处理之后,GPIO 口才会切换到低电平并且在示波器上显示时间。然而,这几行配置代码的执行速度非常快,所以这段时间在测试中可以被忽略不计。

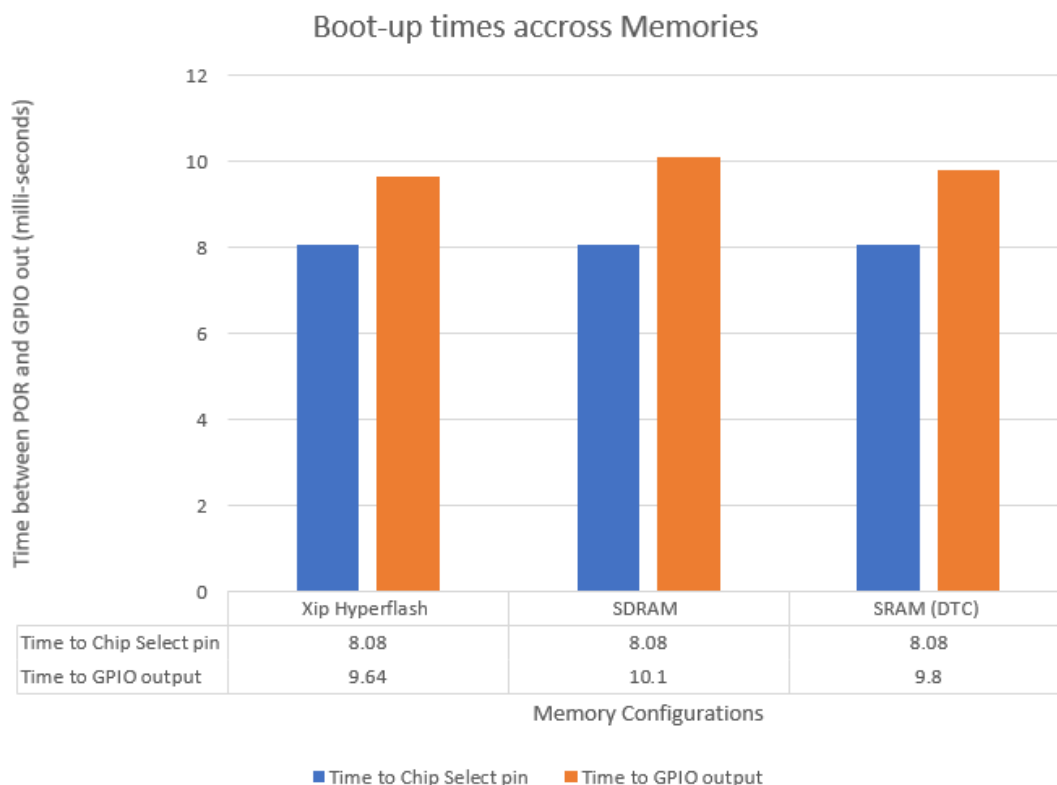
时钟配置

在所有三种情况下,启动映像文件都被写入 RT1050 的闪存。之后,在 MCUXpresso 中,把调试器的“Attach Only”配置设置为真。然后启动调试进程,处理器执行完代码后,停下来读取寄存器的数值,参见 RT1050 参考手册第 18 章中的 CCM 方框图。

启动配置	内核时钟 (MHz)*	FlexSPI 时钟 (MHz)	SEMC 时钟 (MHz)
FlexSPI		130	99
SDRAM	396	130	99
SRAM	396	130	99

*内核时钟的速度也通过将 c1ko1 配置为输出进行验证，此时钟速度将除以 8。用示波器测量这个频率，可验证其为 396 兆赫。

结果



片选引脚信号出现的时间代表的是 RT1050 处理器进行第一次闪存读取的时刻。GPIO 输出的时间表示启动时间。

正如预期的那样，XiP Hyperflash 比其他存储器的启动速度更快。SRAM 和 SDRAM 存储器必须在执行前复制到可执行内存，这将花费更多的时间，因此启动得更慢。在下面的部分，我们将对这些测试是如何运行的，以及为什么 Hyperflash XiP 启动最快的原因提供更全面的解释。

Hyperflash XiP 启动

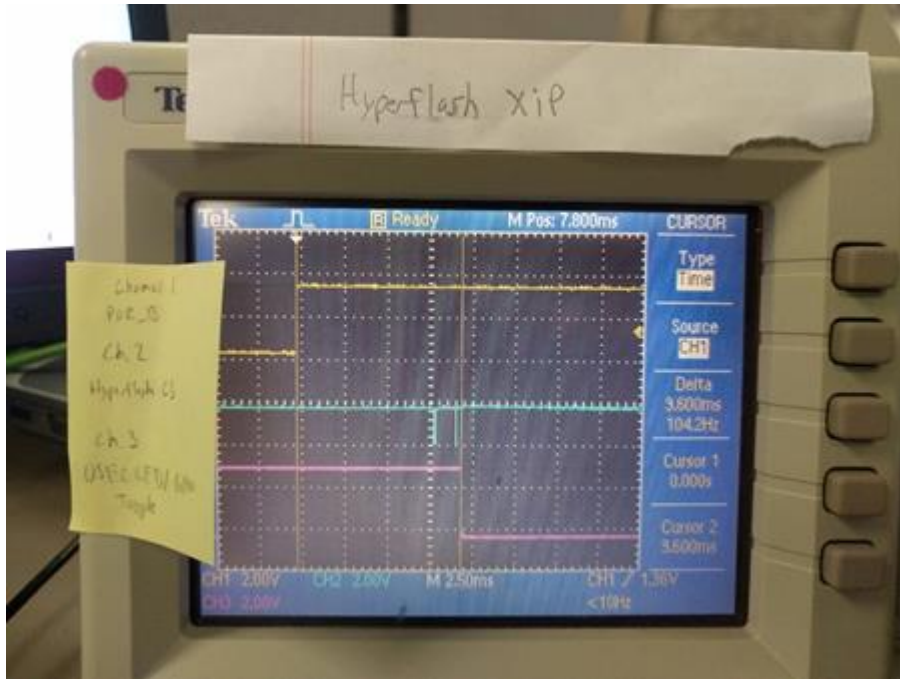
以下是我们预计的 Hyperflash XiP 的启动过程的概述：

- 1、上电复位 (J26-1)
- 2、开始访问闪存 (FlexSPI_SS0)
- 3、在闪存中执行 (XiP)
- 4、第一行代码被执行 (USER_LED)

在 MCUXpresso 中，map 文件显示了以下内容：

Memory region	Used Size	Region Size	%age Used
BOARD_FLASH:	15512 B	64 MB	0.02%
SRAM_DTC:	8428 B	128 KB	6.43%
SRAM_ITC:	0 GB	128 KB	0.00%
SRAM_OC:	0 GB	256 KB	0.00%
BOARD_SDRAM:	0 GB	32 MB	0.00%

示波器图像如下



SDRAM 的启动

处理器从 ROM 启动,ROM 会把应用程序映像从 NOR Flash 复制到 SDRAM (NOR Flash 使用 HyperFlash 通信)。我用 RT 的 flashloader 工具将应用程序加载到闪存中,并将其配置为复制到 SDRAM 中执行。

可以设想复制到 SDRAM 比在 HyperFlash 中直接执行要慢,因为这样必须执行整个复制操作。

SDRAM 的启动过程如下所示:

- 1、上电复位 (J26-1)
- 2、开始访问闪存 (FlexSPI_SS0)
- 3、将代码复制到 SDRAM
- 4、在 SDRAM 中执行 (FlexSPI_SS0)
- 5、执行第一行代码 (USER_LED)

在 MCUXpresso 中, map 文件显示了以下内容:

Memory region	Used Size	Region Size	%age Used
BOARD_FLASH:	0 GB	64 MB	0.00%
BOARD_SDRAM:	15744 B	32760 KB	0.05%
SRAM_DTC:	0 GB	128 KB	0.00%
SRAM_ITC:	0 GB	128 KB	0.00%
SRAM_OC:	0 GB	256 KB	0.00%

Finished building target: SDRAM2_led_blinky.axf

为了运行这个测试,我遵循了以下说明:<https://community.nxp.com/docs/DOC-340655>.



SRAM 的启动

对于 SRAM, 预期会出现类似 SDRAM 的过程。处理器将首先从内部 ROM 启动, 然后转到 HyperFlash。然后, 它将所有内容从 HyperFlash 复制到内部 SRAM DTC 内存, 然后从那里开始执行。

SRAM 启动过程如下:

- 1、上电复位(j26-1)
- 2、开始访问闪存(FlexSPI_SS0)
- 3、将代码复制到 SRAM
- 4、在 SRAM 中执行(FlexSPI_SS0)
- 5、执行第一行代码(USER_LED)

在 MCUXpresso 中, map 文件显示了以下内容:

```
Memory region      Used Size  Region Size  %age Used
BOARD_FLASH:       0 GB      64 MB       0.00%
SRAM_DTC:          15744 B   120 KB      12.81%
BOARD_SDRAM:       0 GB     32760 KB    0.00%
SRAM_ITC:          0 GB      128 KB      0.00%
SRAM_OC:           0 GB      256 KB      0.00%
Finished building target: SRAM1_led_blinky.axf
```

