



更多关于 ADI 公司的 DSP、处理器以及开发工具的技术资料，
 请访问网站：<http://www.analog.com/ee-note> 和 <http://www.analog.com/processor>
 如需技术支持，请发邮件至 processor.support@analog.com 或 processor.tools.support@analog.com

ADSP-BF54x Blackfin® 处理器的启动

作者: Andreas Pellkofer

Rev 1 – June 25, 2008

引言

ADSP-BF54x Blackfin® 处理器提供了多种启动方式，有些是已知的 Blackfin 处理器所具有的方式，有些是经过一定的更新得到，而其它的则是全新的启动方式。

处理器启动模式的基本概念和说明，请参照处理器的 *硬件参考*^[1] 中的启动章节。本文档重点提供以下几方面的参考，以帮助您入门：

- 从启动模式：诸如 SPI 从模式，TWI 从模式和 UART 从模式
- 一个用于主机设备的软件范例(主机是另一个用于示范目的的 ADSP-BF548 处理器)，包括装载文件
- ala 格式表示的启动过程的记录，需要的软件请参照 [4]
- 初始化代码例子中特殊函数的描述 [5]

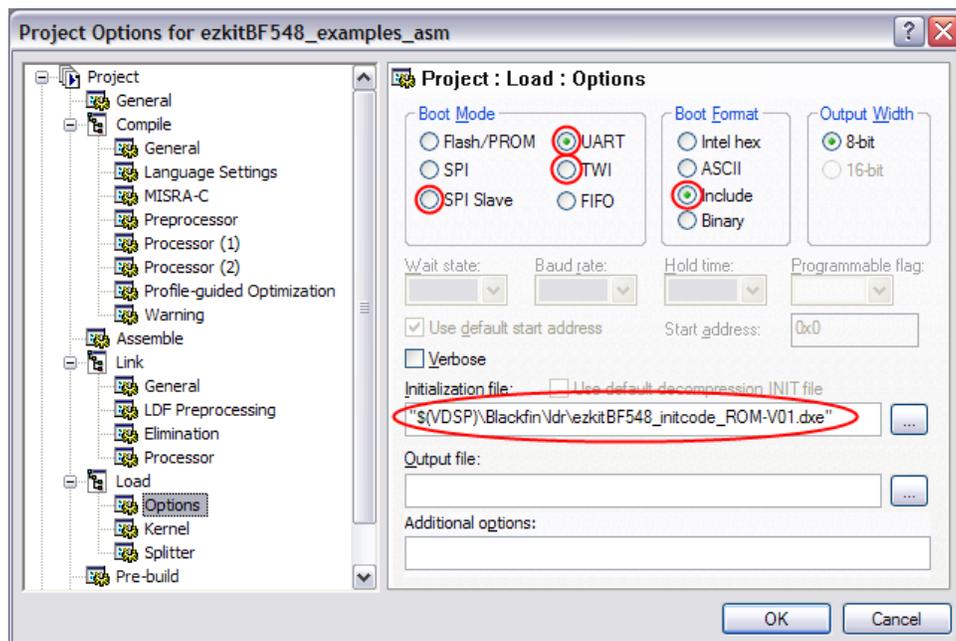


图1 Visual DSP++ 工程选项(工程->工程选项->装载->选项)

设备怎样使主机停止发送后续数据(为示范目的, 在启动过程, 初始化代码里执行了一个延时循环操作)。

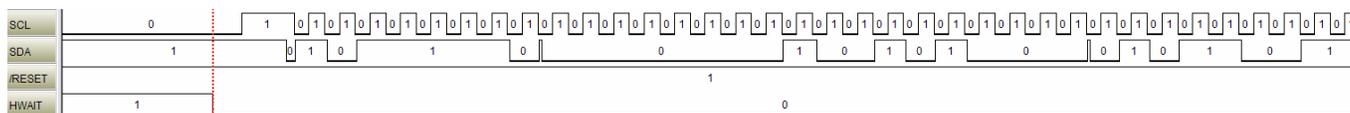


图3 TWI从启动的起始情况



图4 带有延迟的初始化代码的执行

UART 从启动模式(BMODE = 0111)

UART从模式启动中, Blackfin处理器从通过UART连接的外部主设备接收启动数据。更多一般启动和配置的细节请参考[1]。在UART主机启动过程中, 几个关键时刻, 必须妥善处理。

比特率: 处理系统时钟变化

首先要采取的行动是自动波特率序列——根据系统时钟速率设置UART比特率。跟其他从启动模式(如SPI, TWI)不同, 该从设备要求正确地设置UART比特率。图5显示了在一个传输开始时的自动波特率检测序列。一旦接收到@ (0x40)自动波特率检测字节, 启动内核就在它的传输输出上返回一个0xBF, DLL, DLH, 0x00序列。最后, 释放HWAIT。

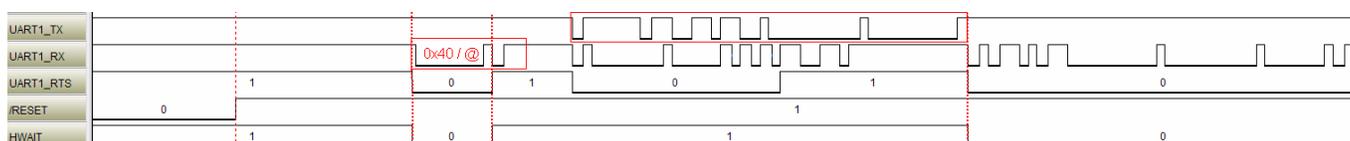


图5 UART: 自动波特率检测

流控

UART从设备向主机发出信号, 以暂停正在进行的传输, 有两种不同的方式: HWAIT和UART硬件流控机制(UART RTS)。

图6显示了初始代码通过向主设备传回状态字的方式延迟启动过程的行为, 这种行为依赖HWAIT信号。一旦置起了HWAIT信号, 主机完成当前字的传输后, 停止发送后续数据而处于等待状态(查看UART1_TX)。这种方式与前面描述的其他从启动模式相似。图5也显示了在HWAIT释放前的一些传输行上的活动以及主机的复位操作。

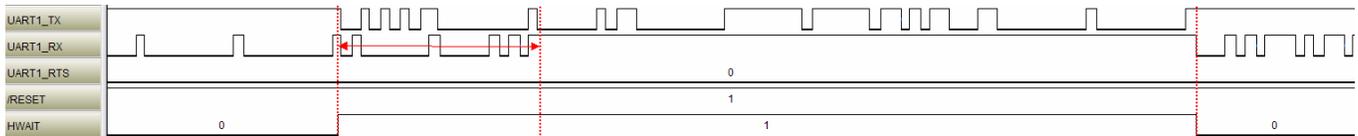


图6 UART:HWAIT信号传输

另一种方式是通过UART的硬件流控机制来暂停一个正在进行的传输。该功能是首次在ADSP-BF54x UART上得以实现，其他Blackfin系列处理器目前还不具备此功能。这种方式采用UART RTS信号。如果主机设备支持这种功能，则不需要连接额外的信号。该功能对主机端来说是相似的，如果置起了RTS信号，主机在完成当前字的传输后立即中止。图7描述了这种行为：主机在UART1-RTS信号置高逻辑(RTS是低逻辑有效)后停止传输后续数据(参照UART1_TX)，这种情况只在内核接收到缓冲中一个特殊的填满状态时才触发。

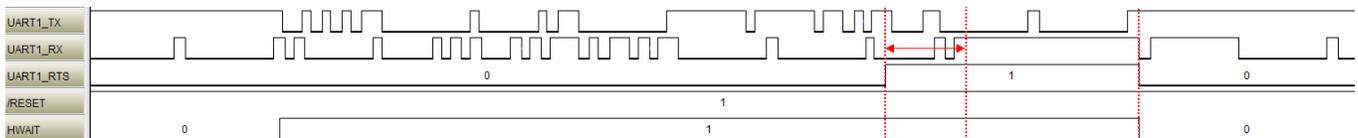


图7 UART: RTS信号传输

图6和图7显示的是从设备正向主机发送数据的过程。对于全双工连接，时间重叠没有问题；而对于半双工的连接，数据线是共享的，程序员必须考虑到这种重叠的情况(具体参照VisualDSP++[®] *initcode* 的例子)。

为了适当地解决复位问题，建议在RTS输出端增加一个上拉电阻。

Visual DSP++[®]启动代码例子

Visual DSP++ 5.0(更新3)安装中的启动代码范例，改变了运行中的PLL锁相环频率，需要密切关注以下两个挑战：

- 当系统时钟在启动过程中改变时，更新UART比特率分配器
- 半双工传输要求由UART RTS发送信号来控制

如果PLL在启动过程(初始化代码)中改变了，程序员必须特别注意这点；否则，启动过程可能失败。UART控制器要求在UARTx_DLL及UARTx_DLH寄存器中设置正确的比特率，来恰当地分析输入进来的数据。如果系统时钟改变了，比特率分配器必须也做相应的调整。在Visual DSP++ 5.0(更新3)或以后的开发工具中，启动代码范例使用两个函数来处理这种情况。这两个函数是u32 `uart1_get_bitrate(void)` 和 `void uart1_set_bitrate(u32)`，分别在PLL改变前后执行，如果检测到BMODE=0111，将被自动调用。它们保存 (u32 `uart1_get_bitrate(void)`) 当前的UART比特率，根据该比特率，计算新的UART分配器的值，并将其存放到UARTx _DLL和UARTx_DLH寄存器中(`void uart1_set_bitrate(u32)`)。

函数 `void uart1_set_bitrate(u32)` 通过发送一些字节向主机提供反馈(类似于在自动波特率检测后的启动内核所做的)，这些字节是：

- 0xAD
- UART1_DLL
- UART1_DLH
- 0x00

在图6到图9中，您能在UART1_TX信号上看到这些字节信息。

此外，关键的部分是PLL改编序列。在PLL改变和更新UART分配器中存在一个时间延迟，你必须确保在改编时间内，UART主机不会发送任何后续数据。

函数u16 uart1_check_buffer(u32)通过使硬件流控无效以及手工设置RTS来强行置起UART的RTS信号。这是通过UARTx_MCR寄存器来完成的。然后你可能必须等待主机发送最后的字节。当该函数正在向主机发送一个4字节的反馈序列，存在一个十分安全的时隙：

- 0xAA
- UART1_MSR
- UART1_LSR
- 0x00

当PLL改编序列运行完成和设置了新的UART分配器的值后，对RTS信号的控制被传回给UART控制器(重新使能硬件流控机制)，然后可恢复UARTx_MCR的原始设置。同时，也能发现HWAIT信号被释放。

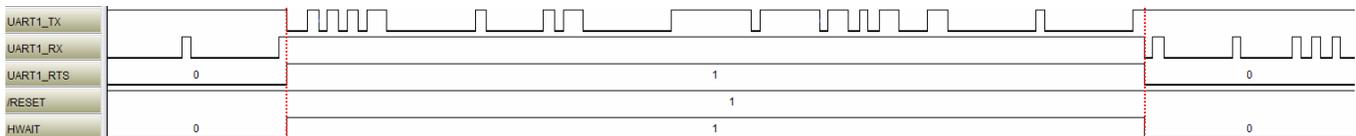


图8 硬件流控的操作

对于半双工的连接，可能仍然存在一个重叠。因此，可以插入一个延迟(例如，一个数据字的长度)，来确保在线路上不会出现冲突。图9显示的序列跟图8相同，只是插入了延迟。



图9 带有额外延迟的硬件流控的操作



注意：如果主机延迟发送最后一个字，您必须调整启动过程来适应所有可能的情况。

附录 A

与该文档相关的有一个ZIP文件，它包含一个适合ADSP-BF548 EZ-KIT Lite[®]开发板的例子代码工程，作为主机设备，支持以下启动模式：

- SPI从启动
- TWI从启动
- UART从启动

该工程在VisualDSP++ 5.0 (更新3)开发工具中编程和测试，基本配置在configurations.h文件中设置，更多信息可参考README.txt文件。

附录 B

相关的一个ZIP文件，它包含特殊的启动过程的一些记录。该文件是安捷伦技术的逻辑分析的应用软件[4]。使用附录A的代码例子，更多信息参考README.txt文件。

参考文献

- [1] *ADSP-BF548 Blackfin Processor Hardware Reference*. Rev 0.3, May 2008. Analog Devices, Inc.
- [2] *ADSP-BF548 Blackfin Processor Peripheral Hardware Reference Manual*. Rev 0.1, March 2007. Analog Devices, Inc.
- [3] *ADSP-BF548 Blackfin Processor Evaluation System Manual*. Rev 1.2, April 2008. Analog Devices, Inc.
- [4] [Agilent Technologies 16900, 16800, and 1680/90 Series Application Software](#).
- [5] VisualDSP++ 5.0 (Update 3): Initialization code examples (<install_path>\Blackfin\ldr\init_code\)

读物

- [6] *ADSP-BF53x/ADSP-BF56x Programming Reference*. Rev 1.2. February 2007. Analog Devices, Inc.
- [7] *EE-240: ADSP-BF533 Blackfin[®] Booting Process*. Rev 3, January 2005. Analog Devices, Inc.
- [8] *EE-331: UART Enhancements on ADSP-BF54x Blackfin[®] Processors*. Rev 1, November 2007. Analog Devices, Inc.

文档记录

Revision	Description
Rev 1 – June 25, 2008 by Andreas Pellkofer	Initial release.