Semiconductors

AN462

硬件和软件验证流程

Rev. _1 — 7 August 1987

应用规格书

文件信息

信息	内容
关键词	UART，软硬件验证
摘要	这个流程是用于检验主机(控制处理器)、UART 和印制电路板之间的信号、总线、电气连接和时序的正确性。以下流程的执行和结果的验证不需要任何其它的测试设备（逻辑分析仪、协议分析仪，示波器等），它是基于处理器能够读写ＵＡＲＴ，并且所得到的结果能显示给操作者的假设之上的。如果无法完全确认简单的读写操被正确的执行，那么针对ＵＡＲＴ连接的其它检验都是不可靠的。

PHILIPS

1. 概述

这个流程是用于检验主机(控制处理器)、UART 和印制电路板之间的信号、总线、电气连接和时序的正确性。以下流程的执行和结果的验证不需要任何其它的测试设备（逻辑分析仪、协议分析仪，示波器等），它是基于处理器能够读写ＵＡＲＴ，并且所得到的结果能显示给操作者的假设之上的。如果无法完全确认简单的读写操被正确的执行，那么针对ＵＡＲＴ连接的其它检验都是不可靠的。

2. 总流程

首先，对相关的寄存器进行写和读的操作，在这些操作中与时钟有关的有片选、读、写信号；其次，通过读取状态寄存器的值来观察所写入的几个控制寄存器的结果；接下来建议的几个流程用于验证总线数据流并采用“本地循环回送” 的模式来验证接收器和发送器的运行，“本地循环回送”模式（所有的数据的发送和接收发生在 UART 内部）用于产生处理器中断或查询状态。这些流程的正确执行将表明内部寄存器、总线接口、时钟发生器、计数器和振荡器的正常运行，没有得到验证的是振荡器的频率、TxD 和 RxD 与外部端口及普通输入输出管脚的连接。

3. 注释

在硬件和软件的验证模式中，读取状态寄存器的状态是非常有效的，它对于检测那些在"已验证"过的硬件和软件上出现的“随机”或“少见”的错误也很有效。状态的读取可以在对设备进行操作之前和之后进行，寄存器的内容将反映异常状况发生的时间地点，这将使得 UART 的内部状态与外部连接、时序、软件一样具有了可见性。例一：在硬件复位后读取状态将显示发送器空标志位置位，这表示发送器已经使能（在硬件复位后将立即返回 0x00），然而软件并没有使能发送器，因此有人会把原因归结于：特殊的干扰，复位信号下降沿太缓慢等等，所有这些就像执行了一个发送器使能的命令，如果这种情况确实发生，采取进一步的验证操作来修正错误的意义并不大。例二：在对 UART 复位和执行几个操作之后，并且在使能接收器之前，发现有一些接收器数据状态位置位（如奇偶校验出错），这表示接收器已经使能，并且收到数据（虽然我们并没在软件上对接收器执行使能的命令），这可能是由于时序的冲突(可能是地址总线上的)移动了接收器 FIFO 的读取指针而导致了上电后接收器 FIFO 的随机状态的上报。经常还出现软件的不同模块都控制 UART，并且各控制是独立的情况，这将导致更多的情况出现。

操作流程	结果及注释
1．执行硬件复位（在没使用硬件复位的系统中，可通过手工断开复位管脚连接来临时产生硬件复位）	ＵＡＲＴ的正常工作并不要求有硬件复位，但是在本流程中我们强烈推荐硬件复位，即便采用手工复位。软件复位可随时执行，但软件复位在某种程度上来说是隐性的，所以在本流程中没有使用软件复位，如果没有完整正确地进行后续的操作是很难检验软件复位是否生效的。
2．向每个 MR 寄存器写入两个不同的数据（0xaa，0x55），不必考虑 MR 指针。	这个操作将把第二个字节写入 MR2 中，（本例是 0x55）这将表明（在第 4 步）你至少已经可以对设备进行读写并且回送数据到控制系统，不需要其它任何测试设备，系统本身就可以完成。这个测试甚至在没有 Vss 和 Vdd 的情况下都可以实现，尽管逻辑低电平将接近+0.7V
3．在对 MR 寄存器操作之前和之后读每个 UART 的状态寄存器。将返回 0x 00	返回的结果一定是0x 00，如果不是0x 00，那么相应的MR寄存器将无法执行其它操作。状态寄存器反映了MR的情况。
4．对 MR 寄存器执行一个单独的读操作，不必考虑 MR 指针将返回第二个写入 MR 寄存器的数据，按照上面的数据将返回 0x55 注意：这一步非常关键，它反映了基本控制的正确性，对于芯片内部逻辑来说这是很简单，即使Vss和Vdd不连接的情况下也起作用。	经过之前的 2 个写操作（第 2 步）后，MR 指针指向 MR2，这时读取 MR 将返回第 2 步中第 2 个写入 MR 的数据。 这一步表明至少一部分的地址线已经连接并且工作正常，CEN，RDN 和 WRN 信号正确。同时表明数据总线正在运行但并不意味着数据总线是正确连接的。 在理论设计中，大部分的 CMOS 器件可通过输入管脚供电，假设至少一个 Vcc 和 Vss。这将给内部逻辑提供近似 3.6V 的电压(在 Vss 上浮一个二极管电压,在 Vcc 上下降一个二极管电压).但它很有可能并不运行在一个特定的速度下，这将容易倒致（在 Vss 和 Vdd 不连接）ＣＭＯＳ的闭锁而损坏器件。新的ＣＭＯＳ电路设计充分考虑了ＣＭＯＳ闭锁的潜在危害，改进了电路设计和执行流程以防止闭锁现象。

操作流程	结果及注释
5．对地址为 0x02 的命令寄存器写入 0x10。 重复第 2 步中对 MR 寄存器的写操作。设置 MR 指针指回 MR1 并执行 2 次对 MR 寄存器的读操作，对所有 MR 寄存器执行该操作，返回值将是先前写入的 2 个字节。 注意：分别置MR1、MR2为00和87对于步骤8来说将是比较方便的，但在这一步中所有的MR寄存器必须写入不同的数据。	这表明 1 条以上的地址线在工作，1 条数据线是正确的，晶振起振，UART 不是工作在掉电模式，Vss 和 Vdd 已连接。 按这顺序完成所有操作之后，读取状态寄存器，返回值都是 0x00。
6．再次对命令寄存器进行写操作并使能发送器，对每个 UART 的命令寄存器写入 0x15。	使能发送器将使状态寄存器中的发送器状态位立即置位,并使 MR 指针指向 MR1
7．读状态寄存器。发送器准备就绪标志位和发送器空标志位都处于置位状态，返回值是 0x0C。 回到上面提到的：如果 MR1 和 MR2 分别是 00 和 87，那么接收器和发送器将处于“本地循环回送“模式。设置 MR1 和 MR2 为以上数值。（在上一步中 MR 指针已指向 MR1）	
8．对时钟选择寄存器写入分频字节来选择一个时钟，如写入字节 BB 表示每个接收器和发送器波特率为 9600bps。	
9．对命令寄存器进行写操作来使能发送器和接收器。？？？	
10．读取状态，结果仍为 0C	
11．对发送器写入一个字节，通过读状态寄存器可看到发生的几个事件。	
12．循环读取状态寄存器，状态为 0x04 和 0x0D 在读取事件中十分重要。 A.在写入到发送器后将立即返回状态值 0x00 B.状态将变为 0x04，这表明起始位的结束及 TxFIFO 已经准备好接收下一个字节。 C.下一个状态值将为 0x05，它将维持小于 7/16 个位时长，通常在总线周期是观察不到这一状态的	寄存器将按顺序报告状态：0x0C，0x00，0x04，0x0C，0x0D，鉴于波特率为 9600 读时序的异步特性， 你可能看不到以上所有数据的，有的维持 1/16 个位的时间，有的仅为 270 纳秒，但是你肯定能观察到 0x04 和 0x0D。

操作流程	结果及注释
D.状态将变为 0x0D	A．这表明发送器已经收到字节，正把它传送到移位寄存器中并开始发送起始位。 B．出现 0x04 的时间取决于相关的 Tx 16x 时钟，Tx 1x 时钟和总线周期，大约在 1/16 到 17/16 个位时长之间。 C．表明接收器收到字节（一过停止位时长的中间点接收器就把字节传送到 Rx FIFO 中），并且发送器还没完成停止位的发送。 D．表明接收器已经把字节传送到Rx FIFO 并且发送器是空的。
13．读接收器 FIFO	你将读到在步骤 11 中送入到发送器的字节
14．读取状态，显示为 0x0C。	显示 0x0C 表明接收器 FIFO 是空的，发送器 FIFO 是空，并且 FIFO 已准备就绪等待数据传送。
15．到此就完成了对总线接口的基本验证，并从状态寄存器中得到相应的数值。	以上流程必须确保正常运行，时序的冲突、错误的的连线或者振荡器不工作都将导致异常结果的出现， 当然芯片的损坏也可能导致异常结果。以上流程的失败是败是“第一类”的影响造成的。
如果设置 MR2 寄存器为 03，便可以将发送器输出管脚连接到接受器输入管脚，并重复步骤 11 到步骤 13 的流程。	在大多数情况下这实现起来很困难，因为电路板上与 UART 的连接已经固定了。但如果可以在发送器和接收器之间建立一个像 “导线”的连接，就证明发送器输出端口和接收器的输入端口并没有损坏。 另一方面，可通过让 UART 与其它 UART 或 DUMB 终端”通话”来验证其功能。这要求你可以针对”另类设备”来对 UART 进行编程,而且”另类设备”已经正确编程并且所有涉及的硬件都正常运行，

操作流程	结果及注释
	另外需要提及的一点：它很有可能导致解决问题的调试过程相当曲折。 在这一点上，与真实的外部的发送器和接收器相连接将涉及到很多方面，故障发生的原因可能是：外部硬件，通信设备之间的差异，以及对通信协议的误解。通信协议包括：波特率，字符长度，RTS/CTS 的握手信号，单工或全双工等等。
请切记所有这些操作是在系统本身实现。不必使用外接的逻辑分析仪，发送器，接收器等等，在这里只用到 3 个资源：UART，UART 与控制设备相连接的接口，控制/显示设备。 那么下来就是 UART 能否与外部设备通信的问题。例如：在这里没有用到 ACR（辅助控制寄存器），MR0 也没有编程，这将导致波特率超过 9600，但发送器和接收器仍将以同样的波特率同时运行。 在循环回送模式中，所有 UART 到处理器的控制和中断配置都可得到验证。这一点很有用，它使得我们在分析 UART 的运行和主机的中断动作中排除了外部的影响因素。	

基本代码用于对C100器件包　在接收发送波特率为9600bps情况下的初始化设置，适用器件：28L198，26C198，26C194，28L194。

本代码的编写是基于“本地循环回送”模式或者把发送器A和接收器A相连接之上的，只有在这一前提下，接收器相应的状态位和中断才会生效。

步骤	操作	寄存器	地址	数据	注释
1	WR	GCCR	F8	04	异步总线周期，修改中断向量使其在向量低3位显示通道值
2	WR	CRa	81	F8	复位芯片
3	WR	MR0a	00	C0	没有流控制，没有Xon/Xoff，Tx FIFO中断条件是FIFO 全空
4	WR	MR1a	01	13	没有RTS/CTS，ISR 不屏蔽，字符模式为：无奇偶校验，8个数据位
5	WR	MR2a	10	00	普通模式，接收器中断准备就绪，1个停止位
6	WR	Rx CSR	0C	1E	9600 波特率
7	WR	Tx CSR	0E	1E	9600 波特率
8	WR	CRa	81	28	清停止状态改变标志位，禁止Tx和RX（在这里是冗余的，但对于重启动上电周期是有利的。）
9	RD	SRa	81	00	读到的状态值应该为00，否则表明硬件有问题
10	RD	ISRa	82	00	
11	WR	IMRa	82	03	使能通道A的Tx的Rx中断
12	WR	ICR	1B	00	设置中断门限为00
13	WR	IVR	1F	00	中断向量设置为00（见步骤1-----已修改的低3位）
14	WR	CRa	81	03	使能Tx的Rx，这将使到一个中断立即产生。这个中断表明发送器A正在发生中断，通过读SR9（状态寄存器）或者让中断向量控制服务路径可以证明这一点

步骤	操作	寄存器	地址	数据	注释
15	RD	SRa	81	0C	发送器准备就绪和并且是空的
16	WR	TXa	83	nun	写入任意数据,Tx中断将结束
17	WR	TXa	83	tt	写入任意数值
18	RD	SDa	81	04	04----发送器不是空的并且准备接受更多的数据，中断关闭因为 Tx FIFO 不是空的。
					在一个字符的时长（10.4ms）后接收器中断将产生，由步骤14来决定中断服务类型。
19	RD	SRa	81	05	接收器有一个字节，发送器仍然忙碌。
20	RD	SRa	81	0B	Tx准备就绪并且是空的，接收器的FIFO准备就绪
21	RD	RXa	83	nn	读取由发送器发过来的第1个字节
22	RD	RXa	83	tt	读取由发送器发过来的第2个字节
23	RD	SRa	81	0C	接收器是空的；发送器是空的并且准备就绪，见步骤15

以下编程是基于UART 的SCCxxx，SCxxx，SCNxxx，SCCxxx系列器件。设置为1个起始位，8个数据位，没有奇偶校验，1个停止位。对于MR寄存器的访问是通过MR指针来实现的。TxA可以在外部与RxA相连接，或者采用本地循环回送模式。

步骤	操作	寄存器	地址	数据	注释
1					执行硬件复位
2					
3	WR	MR0a	0	00	没有流控制，没有Xon/Xoff，Tx FIFO中断条件是FIFO 全空（只有SCxxx器件有MR0）
4	WR	MR1a	0	13	没有RTS/CTS，ISR 不屏蔽，字符模式为：无奇偶校验，8个数据位
5	WR	MR2a	0	07	普通模式，接收器中断准备就绪，1个停止位（Tx在外部与Rx相连接）
6	WR	CSRa	1	BB	Rx 为9600 波特率，Tx 为9600 波特率
7	WR	CRa	2	28	清错误标志位，禁止Tx和RX（在这里是冗余的，但对于重启动的上电周期是有利的。）
8	WR	ACR	4	00	普通模式的波特率，C/T为计数器模式，外部C/T 时钟，禁止状态的变化
9	WR	IMR	5	03	使能RxA和TxA中断
10	RD	SRa	1	00	读到的状态寄存器的值应该为00，任何其他数值都表明硬件有问题
11	WR	CRa	2	15	复位MR指针使其指向MR1，使能Rx和Tx，中断将立即产生
12	RD	SRa	2	0C	表明发送器是空的并且准备就绪
13	WR	TXa	3	55	送第1个数据到发送器，中断信号大约在1到2个位的时长以后产生。
14	WR	TxA	3	AA	送第下个数据发送器，中断信号将在大约一个字符时长后结束。
15	RD	ISR	5	00	循环执行这个读操作或等待中断。下一个中断将是发送器中断，它紧跟在接收器中断的后面
16	RD	SRa	1	0D	发送器是空的并且准备就绪，接收器的FIFO中有一个字节

4. 定义

生命保障——这些产品在设计时并没有考虑到可以用于生命保障器具、装置、或系统；在此类场合，这些产品的故障能够明显地导致人员伤害。对于使用或销售这些产品的飞利浦半导体公司的用户，如果他们想在此类应用中使用这些产品，则他们必须自行承担风险，并同意在由于此类应用而导致任何损坏时全额向飞利浦半导体公司进行赔偿。

进行修改的权利——飞利浦半导体公司保留对此处描述或包含的产品进行修改的权利，其中包括电路、标准单元、和/或软件，以便能够改善产品的设计和/或性能。当产品已经投入批量生产时（状态“生产”），有关的修改将会通过一个《用户产品/过程修改通知书（CPCN）》进行公告。如未另行规定，飞利浦半导体公司不会对任何一个这些产品的使用承担任何责任或义务，不向这些产品转让任何受专利、版权、或掩码著作权保护的许可权或所有权，也不会做出任何表述或担保、说明这些产品没有侵犯任何专利、版权、或掩码著作权。

应用信息——对于任何一件此类产品，此处描述的应用情况仅仅是为了演示性目的。在没有进行进一步的试验或变更之前，飞利浦半导体公司并没有做出任何表示或担保，声明此类应用将会适应于特定的用途。

5. 许可

飞利浦 I^2C 零件的购买

飞利浦 I^2C 零件的购买转让一个飞利浦 I^2C 专利保护的许可在 I^2C 系统中使用零件从而与飞利浦制定的规范一致。 这个规范可以用代码 9398 393 40011 命令。

飞利浦 RC5 零件的购买

飞利浦 RC5 零件的购买转让一个飞利浦 RC5 专利保护的许可在 RC5 系统中使用零件从而与飞利浦制定的详细的控制命令 RC5 标准 UATM-5000 的分配规范一致。

6. 专利

同此通告主要器件使用一个或多个下列专利每个专利可能就其它权限有相应的专利。

<专利号> — <专利所有者>

7. 商标

<商标名称> — 使所有者的商标

<注册商标名称> —是被所有者注册的商标

8. 目录

continued >>