

PCI8100 任意波形发生器

硬件使用说明书



阿尔泰科技发展有限公司

产品研发部修订

目 录

目 录	1
第一章 功能概述	2
第一节、产品应用	2
第二节、总线及制作工艺特点	2
第三节、DA 波形输出功能	2
第四节、DI 数字量输入功能	3
第五节、DO 数字量输出功能	3
第六节、其他指标	3
第七节、板卡外形尺寸	3
第八节、产品安装核对表	3
第九节、安装指导	3
第二章 元件布局图及简要说明	4
第一节、主要元件布局图	4
第二节、主要元件功能说明	4
第三章 信号输入输出连接器定义	6
第四章 各种信号的连接方法	7
第一节、DA 模拟量输出的信号连接方式	7
第二节、外时钟输入信号连接方式	7
第三节、时钟输出信号连接方式	7
第四节、ATR 模拟量外触发信号连接方式	7
第五节、DI 数字量输入的信号连接方法	8
第六节、DO 数字量输出的信号连接方法	8
第五章 数据格式、排放顺序及换算关系	9
第六章 各种功能的使用方法	10
第一节、触发功能	10
第二节、DA 触发功能的使用方法	12
第三节、DA 内时钟与外时钟功能的使用方法	13
第七章 产品的应用注意事项、校准、保修	15
第一节、注意事项	15
第二节、DA 模拟量输出的校准	15
第三节、保修	15
附录 A: 各种标识、概念的命名约定	16

第一章 功能概述

信息社会的发展,在很大程度上取决于信息与信号处理技术的先进性。数字信号处理技术的出现改变了信息与信号处理技术的整个面貌,而数据采集作为数字信号处理的必不可少的前期工作在整个数字系统中起到关键性、乃至决定性的作用,其应用已经深入到信号处理的各个领域。实时信号处理、数字图像处理等领域对高速度、高精度数据采集卡的需求越来越大。ISA 总线由于其传输速度的限制而逐渐被淘汰。我公司推出的基于 PCI 总线、USB 总线等数据采集卡综合了国内外众多同类产品的优点,以其使用的便捷、稳定的性能、极高的性价比,获得多家客户的一致好评,是一系列真正具有可比性的产品,也是您理想的选择。

第一节、产品应用

PCI8100 卡是一种基于 PCI 总线的高速任意波形发生器,可直接插在 IBM-PC/AT 或与之兼容的计算机内的任一 PCI 插槽中,作为实验室、产品质量检测中心等各种领域的信号发生源使用。它的主要应用场合为:

- ◆ 电子产品质量检测
- ◆ 信号采集
- ◆ 过程控制
- ◆ 伺服控制

第二节、总线及制作工艺特点

- ◆ 32 位 PCI 总线,支持 PCI2.2 协议,真正实现即插即用
- ◆ FPGA 接口芯片设计,具有极高的保密性,特别适合 OEM 合作

第三节、DA 波形输出功能

- ◆ 转换器类型: DAC9765
- ◆ 输出量程: $\pm 5V$ 、 $\pm 10V$
- ◆ 转换精度: 12 位
- ◆ 点速率: $1Hz \sim 80MHz$ (12.5ns/点),软件可调
- 分频公式: 采样频率 = 主频 / 分频数,其中主频 = $80MHz$,32 位分频,分频数的取值范围:最低为 1,最高为 2^{32}
- ◆ 分为高速模式(80M,只支持单段连续输出模式)和低速模式(5M 以下)
- ◆ 通道数: 2 路,可同步
- ◆ 触发方式: 软件内触发、硬件外触发
- ◆ 触发模式(TriggerMode): 单次、连续、单步和紧急触发
- ◆ 触发方向(TriggerDir): 负向触发、正向触发、正负向触发
- ◆ 触发源 (TriggerSource): ATR(模拟触发信号)和软件触发
- ◆ 触发源 ATR 输入范围: $-10V \sim 10V$
- ◆ 触发电平: 软件可调,触发电平软件可调($-10V \sim 10V$)
- ◆ 时钟源: 内时钟和外时钟源软件可选
- ◆ 存储器深度: 每路 512K 字(点)RAM 存储器,可分为 666 段
- ◆ 每一路可指定任意一段 RAM 做任意循环输出的数据
- ◆ 数据传输方式: 程序方式

- ◆ 系统测量精度：0.1%
- ◆ 工作温度范围：0℃ ~ +50℃
- ◆ 存储温度范围：-20℃ ~ +70℃

第四节、DI 数字量输入功能

- ◆ 通道数：8 路
- ◆ 电气标准：TTL 兼容
- ◆ 高电平的最低电压：2V
- ◆ 低电平的最低电压：0.8V

第五节、DO 数字量输出功能

- ◆ 通道数：8 路
- ◆ 电气标准：TTL 兼容
- ◆ 高电平的最低电压：3.8V
- ◆ 低电平的最低电压：0.44V
- ◆ 上电输出：低电平

第六节、其他指标

- ◆ 板载时钟振荡器：80MHz

第七节、板卡外形尺寸

130mm(长) x 99.5mm(宽) x 16mm(高)

第八节、产品安装核对表

打开 PCI8100 板卡包装后，你将会发现如下物品：

- 1、PCI8100 板卡一个
- 2、ART 软件光盘一张，该光盘包括如下内容：
 - a) 本公司所有产品驱动程序，用户可在 PCI 目录下找到 PCI8100 驱动程序；
 - b) 用户手册（pdf 格式电子文档）；

第九节、安装指导

一、软件安装指导

在不同操作系统下安装PCI8100板卡的方法一致，在本公司提供的光盘中含有安装程序Setup.exe，用户双击此安装程序按界面提示即可完成安装。

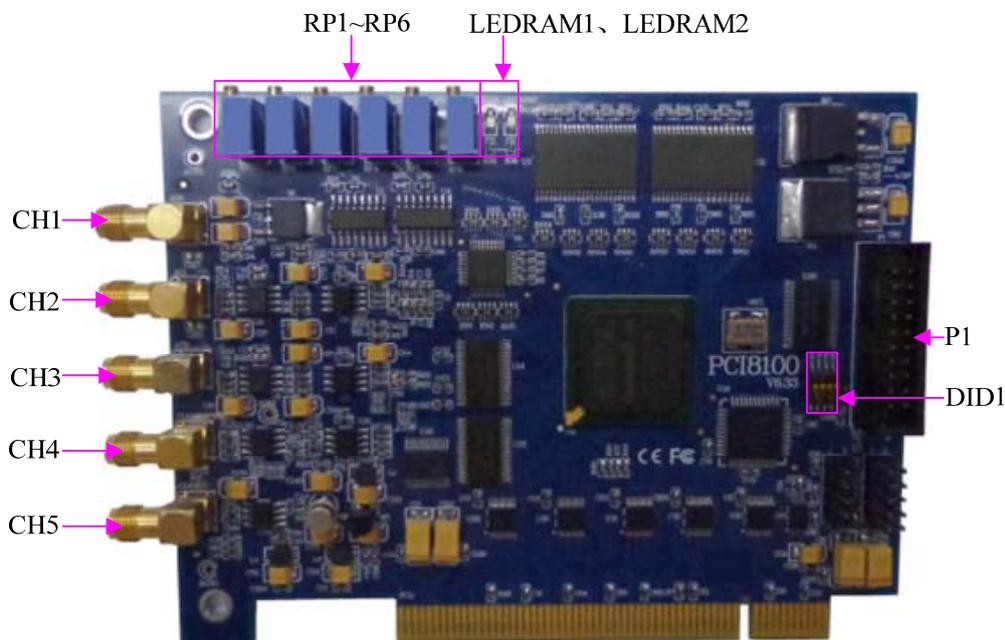
二、硬件安装指导

在硬件安装前首先关闭系统电源，待板卡固定后开机，开机后系统会自动弹出硬件安装向导，用户可选择系统自动安装或手动安装。

注意：不可带电插拔板卡。

第二章 元件布局图及简要说明

第一节、主要元件布局图



第二节、主要元件功能说明

请参考第二节中的布局图，了解下面各主要元件的大体功能。

一、信号输入输出连接器

CH1: AO0 模拟通道 0 信号输出接口

CH2: AO1 模拟通道 1 信号输出接口

CH3: ATR 外触发信号输入接口

CH4: CLKIN 外时钟输入接口

CH5: CLKOUT 内时钟输出接口

P1: 数字量输入输出连接器

以上接口的连接方法请参考《[信号输入输出连接器及连接方法](#)》。

二、电位器

RP1: AO0 模拟量输出满度调节

RP2: AO1 模拟量输出满度调节

RP3: ATR 触发电平零点调节

RP4: ATR 触发电平满度调节

RP5: AO0 模拟量输出零点调节

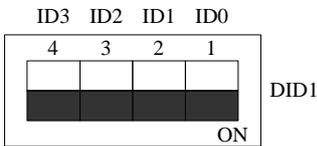
RP6: AO1 模拟量输出零点调节

以上电位器的详细说明请参考《[产品的应用注意事项、校准、保修](#)》章节。

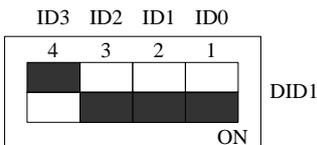
三、物理 ID 拨码开关

DID1: 设置物理ID号，当PC机中安装的多块PCI8100时，可以用此拨码开关设置每一块板卡的物理ID号，这样使得用户很方便的在硬件配置和软件编程过程中区分和访问每块板卡。下面四位均以二进制表示，拨码开关拨向“ON”，表示“1”，拨向另一侧表示“0”。如下列图中所示：位置“ID3”为高位，“ID0”

为低位，图中黑色的位置表示开关的位置。（出厂的测试软件通常使用逻辑ID号管理设备，此时物理ID拨码开关无效。若您想在同一个系统中同时使用多个相同设备时，请尽可能使用物理ID。关于逻辑ID与物理ID的区别请参考软件说明书《PCI8100S》的《设备对象管理函数原型说明》章节中“CreateDevice”函数说明部分）。



上图表示“1111”，则表示的物理ID号为15



上图表示“0111”，则代表的物理ID号为7



上图表示“0101”，则代表的物理ID号为5

下面以表格形式说明物理ID号的设置：

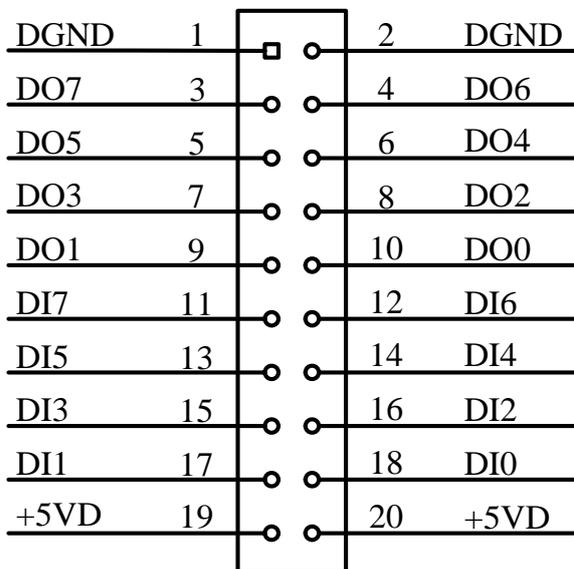
ID3	ID2	ID1	ID0	物理ID (Hex)	物理ID (Dec)
OFF (0)	OFF (0)	OFF (0)	OFF (0)	0	0
OFF (0)	OFF (0)	OFF (0)	ON (1)	1	1
OFF (0)	OFF (0)	ON (1)	OFF (0)	2	2
OFF (0)	OFF (0)	ON (1)	ON (1)	3	3
OFF (0)	ON (1)	OFF (0)	OFF (0)	4	4
OFF (0)	ON (1)	OFF (0)	ON (1)	5	5
OFF (0)	ON (1)	ON (1)	OFF (0)	6	6
OFF (0)	ON (1)	ON (1)	ON (1)	7	7
ON (1)	OFF (0)	OFF (0)	OFF (0)	8	8
ON (1)	OFF (0)	OFF (0)	ON (1)	9	9
ON (1)	OFF (0)	ON (1)	OFF (0)	A	10
ON (1)	OFF (0)	ON (1)	ON (1)	B	11
ON (1)	ON (1)	OFF (0)	OFF (0)	C	12
ON (1)	ON (1)	OFF (0)	ON (1)	D	13
ON (1)	ON (1)	ON (1)	OFF (0)	E	14
ON (1)	ON (1)	ON (1)	ON (1)	F	15

四、状态指示灯

LED RAM1~ LED RAM2: RAM 指示灯

第三章 信号输入输出连接器定义

关于 20 芯插头数字量信号输入输出连接器 P1 的管脚定义（图形方式）



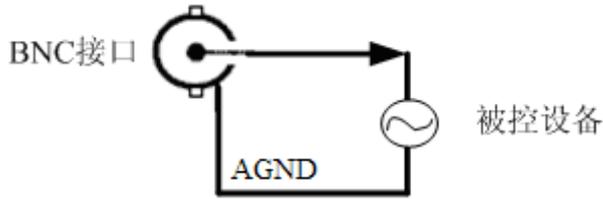
关于 20 芯插头 P1 的管脚定义（表格方式）

管脚信号名称	管脚特性	管脚功能定义
DI0~DI7	Input	数字量输入，其参考地请使用本连接器上的DGND
DO0~DO7	Output	数字量输出，其参考地请使用本连接器上的DGND
+5VD	Output	输出5V电源
DGND	GND	数字地

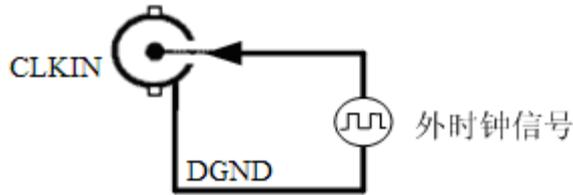
注明：关于DI数字量信号的输入连接方法请参考《[DI数字量输入的信号连接方法](#)》章节；
关于DO数字量信号的输入连接方法请参考《[DO数字量输出的信号连接方法](#)》章节。

第四章 各种信号的连接方法

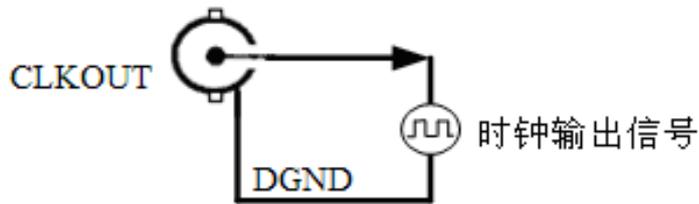
第一节、DA 模拟量输出的信号连接方式



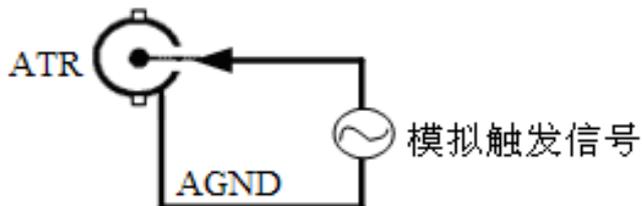
第二节、外时钟输入信号连接方式



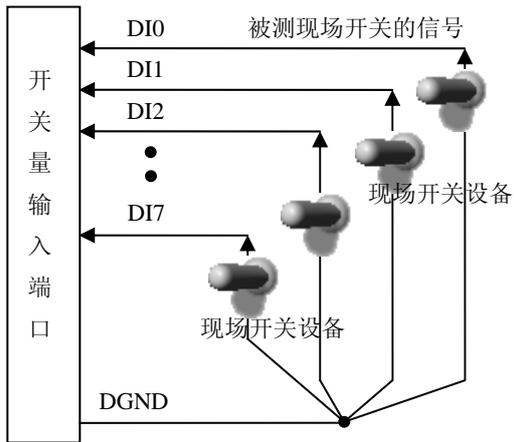
第三节、时钟输出信号连接方式



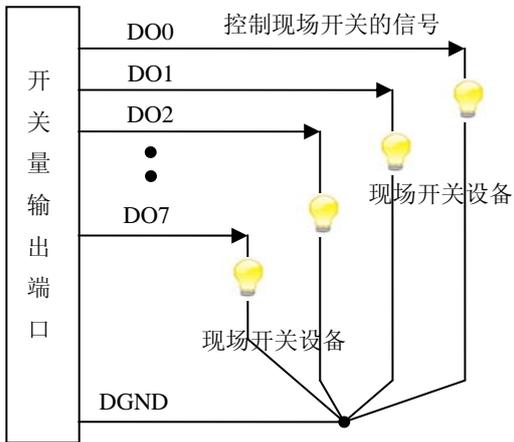
第四节、ATR 模拟量外触发信号连接方式



第五节、DI 数字量输入的信号连接方法



第六节、DO 数字量输出的信号连接方法



第五章 数据格式、排放顺序及换算关系

DA 模拟量输出数据格式及码值换算，如下表所示：

输入	DA原始码(二进制)	DA原始码(十六进制)	DA原始码(十进制)
正满度	1111 1111 1111	FFF	4095
正满度-1LSB	1111 1111 1110	FFE	4094
中间值+1LSB	1000 0000 0001	801	2049
中间值(零点)	1000 0000 0000	800	2048
中间值-1LSB	0111 1111 1111	7FF	2047
负满度+1LSB	0000 0000 0001	001	1
负满度	0000 0000 0000	000	0

注明：当输出量程为±5V、±10V 时，即为双极性输出。假定输出的电压值为 Volt(单位为 mV)，写向设备的 DA 原始码为 nDADData，则换算关系如下：（注意上限不能超过 4095）

±5V量程时： $nDADData = Volt / (10000.00/4096) + 2048$;

±10V量程时： $nDADData = Volt / (20000.00/4096) + 2048$;

第六章 各种功能的使用方法

第一节、触发功能

一、触发源(TriggerSource)

触发源包括软件触发、ATR 模拟量外触发，可以通过软件选择。它的作用是决定由什么信号产生触发事件。注意软件触发源是由软件指令实现，即调用一次 SetDevTrigLevelDA()即产生一次触发事件。而 ATR 需要由板外输入硬件信号。

二、触发方向(TriggerDir)

触发方向包括负向触发、正向触发、正负向触发。具体定义：

负向触发指的是触发源信号产生一个由高到低的瞬间跳变信号（即下边沿信号）时产生触发事件。比如触发源选择 ATR 时表示 ATR 输入信号由大于触发电平状态瞬间变化到小于触发电平状态时产生触发事件。而软件触发源则不必关心触发方向。

正向触发指的是触发源信号产生一个由低到高的瞬间跳变信号（即上边沿信号）时产生触发事件。比如触发源选择 ATR 时表示 ATR 输入信号由小于触发电平状态瞬间变化到大于触发电平状态时产生触发事件。而软件触发源则不必关心触发方向。

正负向触发指的是一旦发生以上两种情况中的一种即可产生触发事件。

三、触发模式 (TriggerMode)

触发模式包括单次触发(Single Trigger Mode)、连续触发 (Continuous Trigger Mode)、单步触发 (Stepped Trigger Mode)、紧急触发(Burst Trigger Mode)，具体定义：

1、单次触发(Single)

总体上是指定的段序列被依次循环输出一次。具体是当调用 EnableDeviceDA ()时 DA 不输出波形，只保持在原始状态下，只有当产生一个触发事件时设备才开始从段 0 输出波形。当段 0 循环指定次数后，自动转入下一段继续循环输出，待循环结束后再继续下一段。当输出到最后一段时，则最后一段自动进入无限循环，直到用户强行停止，停止后保持在被停止时的当前数据点状态下。如图 6.1，使用段 0、1、2 去创建单次触发的任意波形。当信号发生器收到一个触发事件，则开始产生段 0 数据波形，待段 0 数据按照指定次数循环结束后，紧接着循环段 1，以至到段 2。此时段 2 总是作无穷循环直到您停止它。因此，其总循环次数和最后段的循环次数不再有效。一旦触发后，不再接受新的触发事件。

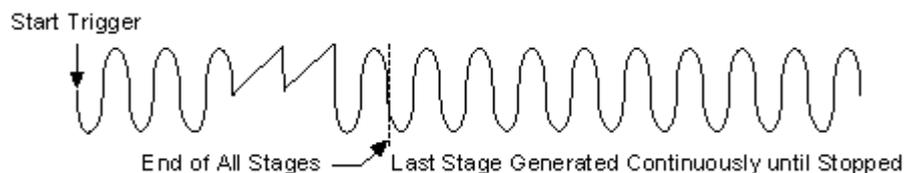


图 6.1

2、连续触发(Continuous)

总体上是指定的段序列被依次整个有限或无限循环输出。具体是当调用 EnableDeviceDA ()时 DA 不输出波形，只保持在原始状态下，只有当产生一个触发事件时设备才开始从段 0 输出波形。当段 0 循环指定次数后，自动转入下一段继续循环输出，待循环结束后再继续下一段。当输出到最后一段时，则最后一段不像单次触发那样，而是循环指定次数自动转入段 0 继续循环，当再次循环完最后段时再回到段 0。若总循环次数设为 0 时，则表示总循环为无穷循环，一旦到最后段结束，便总是回到段 0 继续输出，直到用户强行停止。若总循环次数 (LoopCount) 不等于 0，则表示总循环为有限循环，即从最后段循环完后回到段 0 的次数。当回到段 0 的次数达到 LoopCount 指定的值，则输出到最后段时会自动停止。停止后的状态保持在停止当时的点数据状态下。如图 6.2，使用段 0、1、2 去创建连续触发的任意波形。

当信号发生器收到一个触发事件，则开始产生段 0 数据波形，待段 0 数据按照指定次数循环结束后，紧接着循环段 1，以至到最后段 2。待段 2 结束后自动回到段 0。因此，其总循环次数和段循环次数均有效。一旦触发后，不再接受新的触发事件。

当收到触发事件后，发生器从段 0 开始循环，直至最后一段，然后回到段 0，继续输出，除非用户停止，否则一直循环到指定的总循环次数。在输出过程中，不再接受新的触发事件。当一旦触发时，置触发标志为 1，而触发之前则为 0。且用户停止输出时自动复位为 0。

Continuous Trigger Mode for Arbitrary Waveform Generation Mode

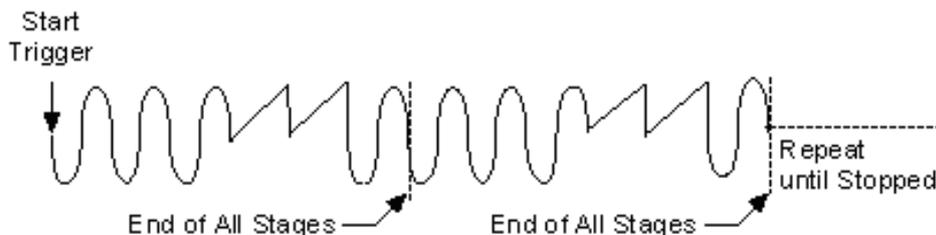
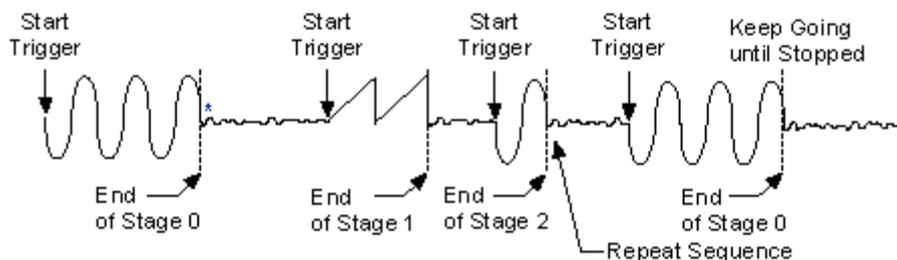


图 6.2

3、单步触发(Stepped)

总体上是指定的段序列中每个段均要接受一个独立触发事件才被依次有限循环输出。具体是当调用 EnableDeviceDA ()时 DA 不输出波形，只保持在原始状态下，只有当产生第一个触发事件时设备才开始从段 0 输出波形，待指定次数循环完后，自动停止，且保持在该段最后一个点状态下。此时若再产生一个触发事件，则自动转入下一段继续循环输出，待循环结束后又自动停止，且保持在该段最后一个点状态下，直到最后一个结束后又再产生触发事件，则又自动回到段 0。注意在各个段循环输出过程中新的触发事件无效，只有在某个段循环结束处于最后点保持状态下时产生的触发事件才有效。如图 6.3，使用段 0、1、2 去创建单步触发的任意波形。当信号发生器收到一个触发事件，则开始产生段 0 数据波形，待段 0 数据按照指定次数循环结束后，自动停止，且保持在最一个点状态下，当再来一个触发事件时，则自动循环输出段 1，以至到最后段 2。待段 2 结束后自动再来新的触发则又回到段 0。因此，只有段循环次数有效。



*The first eight samples of the next stage are generated repeatedly.

图 6.3

4、紧急触发(Burst)

总体上是指定的段序列中每个段均要接受一个独立触发事件才被依次无限循环输出。具体是当调用 EnableDeviceDA ()时 DA 不输出波形，只保持在原始状态下，只有当产生第一个触发事件时设备才开始从段 0 输出波形，此时段 0 作无限循环输出，此时若再产生一个触发事件，则自动转入下一段继续作无限循环输出，直至到最后段作无限循环输出，若再来一个触发事件，则又自动回到段 0 继续作无限循环输出。注意在各个段循环输出过程中新的触发事件总是立即有效。待用户停止后保持在最后一个点上。

如图 6.4, 使用段 0、1、2 去创建紧急触发的任意波形。当信号发生器收到一个触发事件, 则开始产生段 0 数据波形, 此时段 0 则作无限循环输出, 若此时再来一个触发事件, 则自动开始段 1 的无限循环输出, 当再来触发事件, 则自动开始段 2 的无限循环输出, 当再来触发事件时则又回到段 0。因此, 总循环次数和段循环次数均无效。

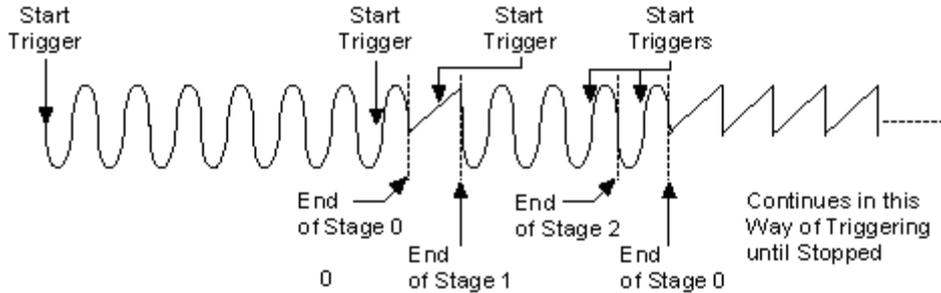


图 6.4

注: 所谓原始状态即指上电最初状态或输出停止后保持的状态也可指软件复位后的状态。

第二节、DA 触发功能的使用方法

一、DA 软件内触发功能

在初始化DA时, 若DA硬件参数DAPara. TriggerMode = PCI8100_TRIGSRC_SOFT_DA时, 则可实现内触发采集。在内触发回放功能下, 调用EnableDeviceDA函数启动DA时, DA即刻进入转换过程, 不等待其他任何外部硬件条件。也可理解为内触发。

具体过程请参考以下图例, 图中DA工作脉冲的周期由设定的采样频率(Frequency)决定。DA启动脉冲由软件接口函数EnableDeviceDA函数产生。

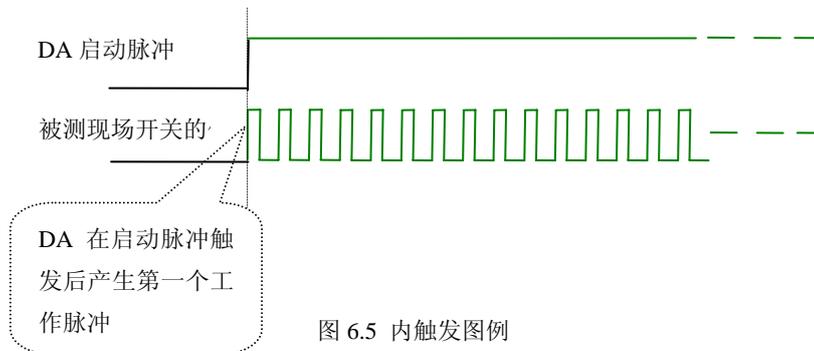


图 6.5 内触发图例

二、DA 硬件外触发功能

在初始化DA时, 若DA硬件参数DAPara. TriggerMode = PCI8100_TRIGSRC_ATR_DA时, 则可实现外触发采集。在外触发采集功能下, 调用EnableDeviceDA函数启动DA时, DA并不立即进入转换过程, 而是要等待外部硬件触发源信号符合指定条件后才开始转换DA数据。其外部硬件触发源信号由命名为“ATR”的BNC接头输入提供。ATR都可提供边缘触发及电平触发。关于在什么条件下触发DA, 由用户选择的触发类型(TriggerMode)、触发方向 (TriggerDir)、触发电平共同决定。各种条件的具体实现请参考以下两小节内容。

以DA硬件参数DAPara. TriggerMode = PCI8100_TRIGMODE_SINGLE, 为单次触发为例, 说明触发功能。

(一)、ATR 模拟触发功能

模拟量触发是将一定范围内变化的模拟量信号作为触发源。由ATR提供的触发源信号与模拟触发电平信号同时进入模拟比较器进行高速模拟比较，产生一个预期的比较结果(Result)来触发DA转换（如下图）。模拟触发电平信号由AO0的输出电压决定。其模拟量触发源信号的有效变化范围为±10V，具体实现方法是：

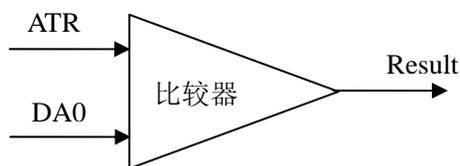


图 6.6 模拟量比较原理

当DAPara.TriggerDir = PCI8100_TRIGDIR_NEGATIVE时，为正向触发，即当ATR触发源信号从小于触发电平变化至大于触发电平时，AD即刻进入转换过程，在此情况下，ATR的后续状态变化并不会影响AD采集，除非用户重新初始化AD。

当DAPara.TriggerDir = PCI8100_TRIGDIR_POSITIVE时，即为负向触发，它与正向触发方向相反，其他方面同理。见图6.7：

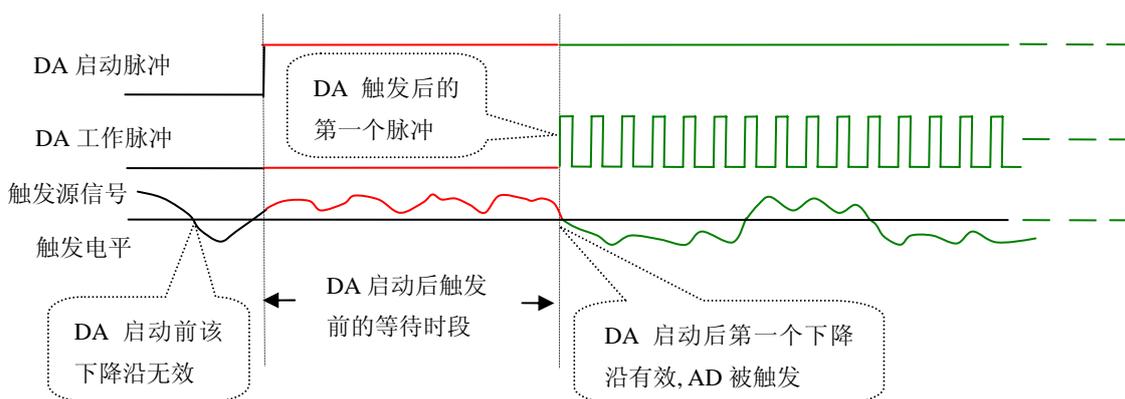


图 6.7 下降沿触发图例

当ADPara.TriggerDir = PCI8100_TRIGDIR_POSIT_NEGAT时，即选择触发方向为上边沿或下边沿触发。它的特点是只要触发源信号的变化跨越触发电平便立即触发AD转换。后续变化对AD采集无影响。此项功能可应用在只要外界的某一信号变化时就采集的场合。

第三节、DA 内时钟与外时钟功能的使用方法

一、DA 内时钟功能

内时钟功能是指使用板载时钟振荡器产生的时钟信号去触发DA定时转换（即DA的刷新时钟）。而读取SRAM中存放的波形数据的时钟则是经板载逻辑控制电路根据用户指定的分频数分频后产生的，该时钟最大与触发DA 的频率相同，最小可以达到DA触发频率的65535倍分频，这样可以实现慢速回放SRAM中存放的波形。要使用内时钟功能应在软件中置硬件参数 DAPara.ClockSouce = PCI8100_CLOCKSRC_IN。该时钟的频率在软件中由硬件参数DAPara.Frequency决定。如Frequency = 100000，则表示读取波形的速度为100KHz，但是DA的刷新时钟不变，仍然是40MHz。

二、DA 外时钟功能

外时钟功能是指使用板外的时钟信号来定时触发DA进行转换。该时钟信号由连接器BNC的CLKIN脚输入提供。要使用外时钟功能应在软件中置硬件参数 `DAPara.ClockSouce = PCI8100_CLOCKSRC_OUT`。该时钟的频率主要取决于外时钟的频率。在外时钟模式下,无论是DA的刷新时钟还是读取波形数据的时钟均与外时钟相同。

第七章 产品的应用注意事项、校准、保修

第一节、注意事项

在公司售出的产品包装中，用户将会找到这本硬件说明书、软件说明书和PCI8100板，同时还有产品质保卡。产品质保卡请用户务必妥善保存，当该产品出现问题需要维修时，请用户将产品质保卡同产品一起，寄回本公司，以便我们能尽快的帮用户解决问题。

在使用该产品时，应注意产品正面的IC芯片不要用手去摸，防止芯片受到静电的危害。万不得已时，请先将手触摸一下主机机箱(确保主机电源三线插头中的地线与大地相接)，以提前放掉身体上的静电。

第二节、DA 模拟量输出的校准

在以 $\pm 10V$ 为例：

- 1) 将数字电压表的地线与任意模拟地(AGND)相接，电压表的输入端与需要校准的DA通道相连接。
- 2) 在Windows下运行PCI8100高级程序，选择菜单文件操作下的D/A输出检测，根据需要校准的通道选择相应的手动单点输出。
- 3) 将D/A输出设置为2048，通过调整电位器RP5和RP6，使相应的D/A输出为0.000V。
- 4) 将D/A输出设置为4095，通过调整电位器RP1和RP2，使相应的D/A输出为10.000V。
- 5) 重复以上3)、4)步骤，直到满足要求为止。

第三节、保修

PCI8100自出厂之日起，两年内凡用户遵守运输，贮存和使用规则，而质量低于产品标准者公司免费修理。

附录 A：各种标识、概念的命名约定

AO0、AO1：表示模拟量输出通道，0、1 为模拟量输出通道编号(Number).

CLKIN：表示外时钟输入管脚

TTL：表示方波输出管脚

ATR：表示触发源信号输入管脚

DAPara：指的是 DA 初始化函数中的 DAPara 参数，它的实际类型为结构体 PCI8100_PARA_DA