

32 位微控制器

高级定时器 ADT 的输入信号处理

适用对象

系列	产品型号	系列	产品型号	系列	产品型号
HC32L110	HC32L110C6UA HC32L110C6PA HC32L110C4UA HC32L110C4PA HC32L110B6PA HC32L110B4PA HC32L110B6YA	HC32F00	HC32F003C4UA HC32F003C4PA HC32F005C6UA HC32F005C6PA HC32F005D6UA	HC32L13	HC32L130E8PA HC32L130F8UA HC32L130J8TA HC32L130J8UA HC32L136J8TA HC32L136K8TA
HC32F03	HC32F030E8PA HC32F030F8UA HC32F030F8TA HC32F030H8TA HC32F030J8TA HC32F030K8TA	HC32L07	HC32L072PATA HC32L072KATA HC32L072JATA HC32L073PATA HC32L073KATA HC32L073JATA HC32L072FAUA	HC32F07	HC32F072PATA HC32F072KATA HC32F072JATA HC32F072FAUA
HC32L17	HC32L176PATA HC32L176MATA HC32L176KATA HC32L176JATA HC32L170JATA HC32L170FAUA	HC32F17	HC32F176PATA HC32F176MATA HC32F176KATA HC32F176JATA HC32F170JATA HC32F170FAUA HC32F170LATA	HC32L19	HC32L196PCTA HC32L196MCTA HC32L196KCTA HC32L196JCTA HC32L190JCTA HC32L190FCUA
HC32F19	HC32F196PCTA HC32F196MCTA HC32F196KCTA HC32F196JCTA HC32F190JCTA HC32F190FCUA				

目 录

1	摘要	3
2	功能介绍	3
3	高级定时器的捕获功能	4
3.1	高级定时器 ADT 的硬件事件	4
3.2	输入电平的捕获.....	4
3.3	AOS 事件的捕获	7
3.4	VC 输出结果的捕获	9
4	高级定时器的外部输入计数功能	10
4.1	单路脉冲信号计数	10
4.2	双路正交信号计数	11
5	总结	14
6	参考样例及驱动	14
7	其他信息	14
8	版本信息 & 联系方式	15

1 摘要

本篇应用笔记主要介绍华大半导体 MCU*的高级定时器 Advanced Timer（以下简称 ADT）的输入信号处理。

本应用笔记主要包括：

- ADT 高级定时器的硬件事件
- 输入电平捕获
- AOS 事件捕获
- VC 输出结果的捕获
- 单路脉冲信号计数
- 双路正交信号计数

注意：

- 本应用笔记为华大半导体 MCU*的应用补充材料，不能代替用户手册，具体功能及寄存器的操作等相关事项请以用户手册为准。

2 功能介绍

ADT 除了可以调制输出多种 PWM，还可以用来对输入信号进行多种方式的处理，包括捕获和外部输入计数等。

***支持型号见封面。**

3 高级定时器的捕获功能

3.1 高级定时器 ADT 的硬件事件

ADT 一共有 3 类硬件事件：CHA/B 端口事件，TRI_x 端口（x=A/B/C/D）事件，AOS 事件：

1. CHA/B 端口事件是由 ADT 的通道 IO 输入的电平或者电平边沿跳变事件，每个 ADT 的定时器单元都有一对 CHA/B 引脚，由 GPIO 的功能配置寄存器 P_{xyy}_SEL（x=A/B/C/D，yy=01~15）寄存器设置。
2. TRI_x 事件是 ADT 三个单元共用的外部输入触发 IO 产生的电平边沿跳变事件，该端口由 ADT 寄存器 TTRIG 设置。
3. AOS 事件是内部指定的模块包括 ADT 自身产生的事件请求（对于 L110/F005/F003 系列而言，是指中断请求，而不是事件请求），三个单元共用。可以同时设置 4 个 AOS 事件，由 ADT 寄存器 ITRIG 设置。

3.2 输入电平的捕获

ADT 的每一个定时器单元，都具有两路捕获通道，这两路捕获通道既可以分别捕获两个 IO 口（CHA/B 或 TRI_x 引脚）的输入信号，也可以用于捕获同一个 IO 的输入信号。

捕获通道 A 的硬件捕获条件由寄存器 HCPAR 设置，捕获通道 B 的硬件捕获条件由 HCPBR 寄存器设置。CHA/B 或 TRI_x 引脚输入的电平跳变符合所设置的条件，就会产生捕获动作，将计数器 CNT_{ER} 的值保存到对应的 GCMAR 或 GCMBR 寄存器中。

如果在设置捕获条件的同时，设置与硬件捕获条件对应的硬件启动（HSTAR）、硬件停止（HSTPR）或者硬件清零（HCELR）的条件，就可以在捕获的同时执行相应的硬件启动（定时器）、停止（定时器）或者清零（定时器计数值）的动作。

例 1：用户需要捕获一个输入脉冲的占空比：

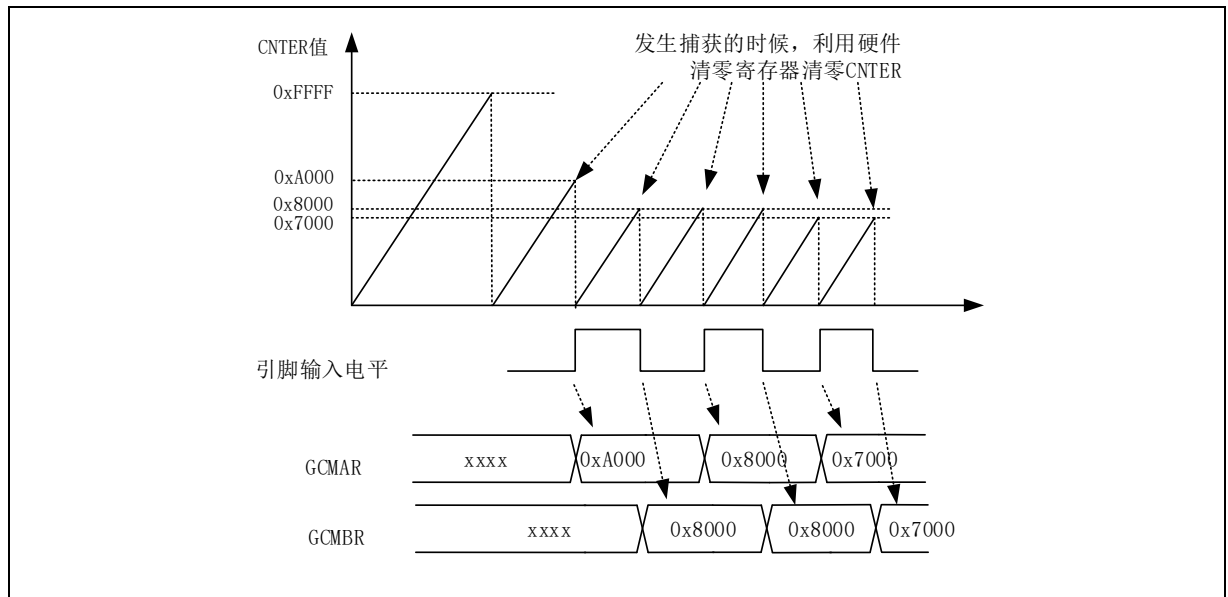


图 1 高低电平捕获示意图

如图 1 所示，使用捕获通道 A 和捕获通道 B 对通过 CHA/B 或者 TRIx 的某一个 IO 引脚输入的信号进行捕获，在电平的上升沿和下降沿的时候都会清零 CNTER 计数器，所以捕获通道 A/B 的捕获值都是两次电平跳变之间的值：

HCPAR 捕获条件：CHA/B 或者 TRIx 的 IO 端口采样到上升沿

HCPBR 捕获条件：CHA/B 或者 TRIx 的 IO 端口采样到下降沿

HCELR 的清零条件：CHA/B 或者 TRIx 的 IO 端口采样到上升沿和下降沿

如上述条件设置，当有连续的脉冲信号输入：在符合设定的捕获件时候，通道 A 的通道 B 就会发生捕获；在符合设定的清零件时候，定时器的计数器也会被自动清零（先捕获再清零）。通道 A 的捕获结果送入寄存器 GCMAR，通道 B 的捕获结果送入寄存器 GCMBR：

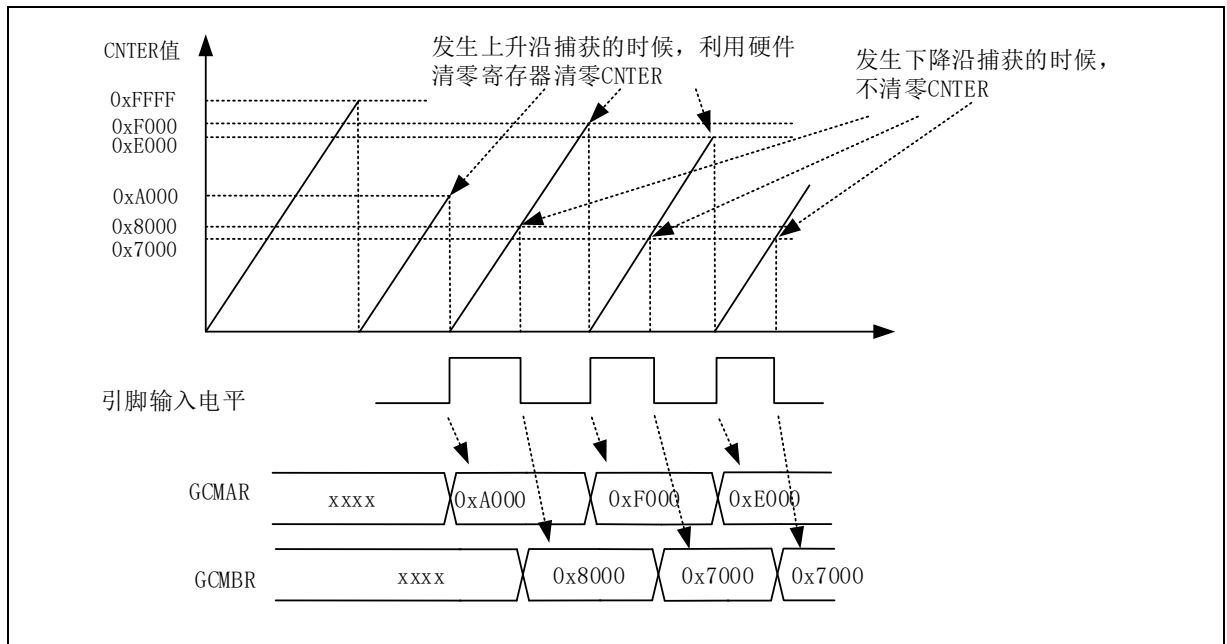
捕获通道 A → 捕获低电平的计数值 → GCMAR

捕获通道 B → 捕获低高平的计数值 → GCMBR

如果 ICONR.INTENA 和 ICONR.INTENB 使能，在发送上述捕获的时候，将分别产生计数比较匹配中断 A（其实为捕获中断 A）和计数比较匹配中断 B（其实为捕获中断 B）。

因此，该串脉冲的占空比：GCMBR 值 / (GCMBR 值 + GCMAR 值)。

例 2：用户需要捕获一串输入脉冲的周期：



如图 2 所示，硬件捕获条件和硬件清零设置如下：

HCPAR 捕获条件：CHA/B 或者 TRIx 的 IO 端口采样到上升沿

HCPBR 捕获条件：CHA/B 或者 TRIx 的 IO 端口采样到下降沿

HCELR 的清零条件：CHA/B 或者 TRIx 的 IO 端口采样到上升

在每个电平的上升沿的时候，捕获 CNTER 的值到 GCMAR，并且设置硬件自动清零 CNTER，这样 GCMAR 存储的值就是该串脉冲的周期值（除去第一个上升沿捕获的值可能不是该串脉冲的周期值）。

如果在捕获周期的条件设置之外，再用捕获通道 B 设置条件，捕获电平下降沿时候的值，这时候 GCMBR 的值就是该串脉冲的高电平时间值，如此占空比就是 GCMBR 值 / GCMAR 值。

操作步骤（以 ADT 的 Timer4 为例）：

1. 使能 ADT 和 GPIO 外设时钟。
2. 设置 IO：Timer4 的 CHA/B 功能在 GPIO 寄存器内设置；TRIx 的 IO，在 GPIO 内设置为输入引脚，在 ADT 的寄存器 TTRIG 内选择对应的引脚。
3. Timer4 设置为锯齿波计数模式，计数方向为向上计数，根据所需捕获的电平大概周期范围选择合适的 Timer4 的分频值。

4. Timer4 的周期设置为 0xFFFF。
5. Timer4 的通道 A 和通道 B 设置为捕获输入（PCONR.CAPCA 和 CAPCB 设置为 1）。
6. Timer4 的 CHA/B 或者 TRIx 输入滤波时间设置并且滤波使能（FCONR，该寄存器只在 Timer4 设置，即使使用 Timer5/6 作引脚的捕获输入功能）。
7. 设置通道 A 的硬件捕获条件（HCPAR）和通道 B 的硬件捕获条件（HCPBR）为 CHA/B 或者 TRIx 端口采样条件。
8. 设置硬件清零条件（CHA/B 或者 TRIx 端口采样条件）并且使能硬件清零功能（如有其他捕获需求，也可以设置硬件启动和停止条件）。
9. 清零 Timer4 的中断标志，使能计数比较匹配中断 A（ICONR.INTENA）和计数比较匹配中断 B（ICONR.INTENB），并且使能 Timer4 的 NVIC 中断。
10. 启动 Timer4。
11. 在中断内读取捕获值，并清零对应的中断标志。

硬件捕获事件选择寄存器 HCPAR/HCPBR 配合硬件启动、硬件停止和硬件清零寄存器 (HSTAR、HSTPR 和 HCELR)，可以组合出很多种用户需要的捕获功能。上述例子，仅作为抛砖引玉，用户可以结合手册，进一步拓展 ADT 的使用。

3.3 AOS 事件的捕获

在 AOS 触发控制寄存器 ITRIG 选择了 AOS 事件之后，硬件捕获事件选择寄存器 HCPAR 或 HCPBR 可以设置 AOS0~3 事件作为捕获条件。

例 3：捕获 VC0 的中断事件 VC0_INT：

HCPA/BR 捕获条件：AOS0 触发事件有效

HCELR 的清零条件：AOS0 触发事件有效

ITRIG 的 AOS 事件选择：AOS0 触发源选择 VC0_INT

VC0 配置上升沿或者下降沿中断使能，但是对应序号的 NVIC 中断可以不需要使能（如果不使能 VC0 的 NVIC 中断，需要在 ADT 的捕获中断内将 VC 的中断标志清零），上述的 ITRIG 的 AOS0 选择 VC0_INT，捕获寄存器选择 AOS0 捕获。再配置 VC 和 ADT 的其他基本设置，这样当 VC 输出结果电平边沿跳变的时候，就可以捕获 COUNTER 的值到 GCMAR/GCMBR 寄存器。

利用硬件清零寄存器，使能 AOS0 事件匹配的时候清零 CNTER，这样每一次捕获发生后，会自动清零 CNTER。

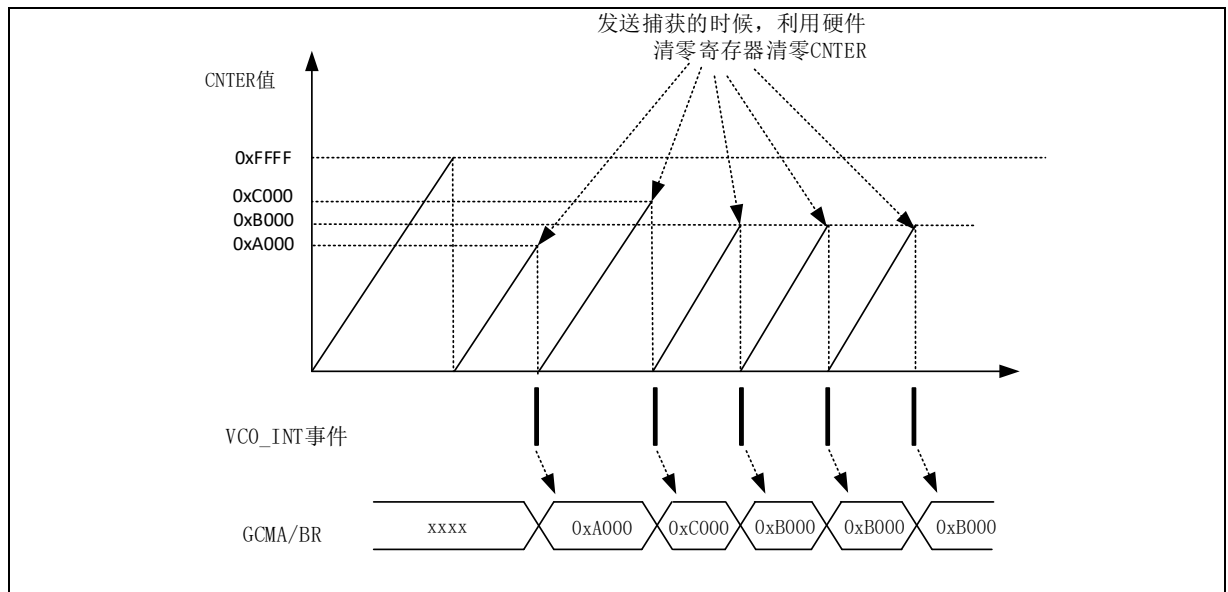


图 3 AOS 事件捕获示意图

操作步骤（以 ADT 的 Timer4 为例）：

1. 使能 ADT 和 GPIO 外设时钟。
2. 设置 IO：配置 VC 的相应引脚。
3. Timer4 设置为锯齿波计数模式，计数方向为向上计数，根据所需捕获的事件的大概周期范围选择合适的 Timer4 的分频值。
4. Timer4 的周期设置为 0xFFFF。
5. Timer4 的通道 A 设置为捕获输入（PCONR.CAPCA 设置为 1）。
6. 设置通道 A 的硬件捕获条件为 AOS 触发事件（HCPAR）。
7. 设置硬件清零条件（AOS 触发事件）并且使能硬件清零功能（如有其他捕获需求，也可以设置硬件启动和停止条件）。
8. 选择 ADT 的 AOS 触发源（ITRIG）。
9. 清零 Timer4 的中断标志，使能计数比较匹配中断 A（ICONR.INTENA）并且使能 Timer4 的 NVIC 中断。
10. 启动 Timer4。
11. 在中断内读取捕获值，并清零对应的中断标志（也要清零 VC 的中断标志）。

利用 AOS 的触发，也可以把两个 ADT 单元，例如 Timer4 的上溢中断，输出给 Timer5 做硬件递加计数，可以组成一个 32bit 的定时器功能。注意，此时 Timer4 除了需要使能 ICONR.INTENOVF（不需要使能 NVIC 中断）以外，还需要使能 CR.OVFE（上溢匹配输出 AOS trig 使能控制位）。

3.4 VC 输出结果的捕获

上述 3.3 描述的是通过 AOS 事件触发（AOS 请求为 VC 中断事件）的捕获。但是通过 AOS 事件的捕获，捕获只能发生在 VC 中断请求发生的时候，ADT 不能自动区分 VC 的中断请求为上升沿还是下降沿，所以除此之外，ADT 也可以通过 CHA/CHB 端口，对 VC 输出结果进行捕获（不需引脚外接引线），以达到测量 VC 输出结果的高电平或低电平的宽度。

操作步骤如下（以 VC1 输出结果输入到 Timer4 的 CHB 为例）：

1. 使能 ADT、VC 和 GPIO 外设时钟。
2. 设置 IO：配置 VC 的相应引脚。
3. Timer4 设置为锯齿波计数模式，计数方向为向上计数，根据所需捕获的事件的大概周期范围选择合适的 Timer4 的分频值。
4. Timer4 的周期设置为 0xFFFF。
5. Timer4 的通道 A 和通道 B 设置为捕获输入（PCONR.CAPCA 和 CAPCB 设置为 1）。
6. Timer4 的 CHB 输入滤波时间设置并且滤波使能（FCONR，该寄存器只在 Timer4 设置，即使使用 Timer5/6 作引脚的捕获输入功能，用户视需求而决定是否使用）。
7. 设置通道 A 的硬件捕获条件（HCPAR）为 CHB 端口采样到下降沿，通道 B 的硬件捕获条件（HCPBR）为 CHB 端口采样到上升沿。
8. 设置硬件清零条件（CHB 端口采样到下降沿和上升沿）并且使能硬件清零功能（如有其他捕获需求，也可以设置硬件启动和停止条件）。
9. 配置 VC1，并使能输出结果到 ADT CHB，不需要使能中断。
10. 清零 Timer4 的中断标志，使能计数比较匹配中断 A（ICONR.INTENA）和计数比较匹配中断 B（ICONR.INTENB），并且使能 Timer4 的 NVIC 中断。
11. 启动 Timer4。
12. 在中断内读取捕获值，并清零对应的中断标志。

4 高级定时器的外部输入计数功能

4.1 单路脉冲信号计数

硬件递加事件选择寄存器 HCUPR 或者硬件递减事件选择寄存器 HCDOR 设置硬件递加或者递减条件，单路脉冲信号通过 TRIx 引脚输入，当输入信号电平边沿跳变满足该条件的时候，ADT 的计数器 COUNTER 会执行递加或者递减计数的操作。

例 4：从 TRIA 引脚输入一串脉冲波形，计数这个脉冲的周期个数：

HCUPR.HCUP8 =1, 设置采样到 TRIA 端口的上升沿电平后，ADT 的计数器执行递加计数。

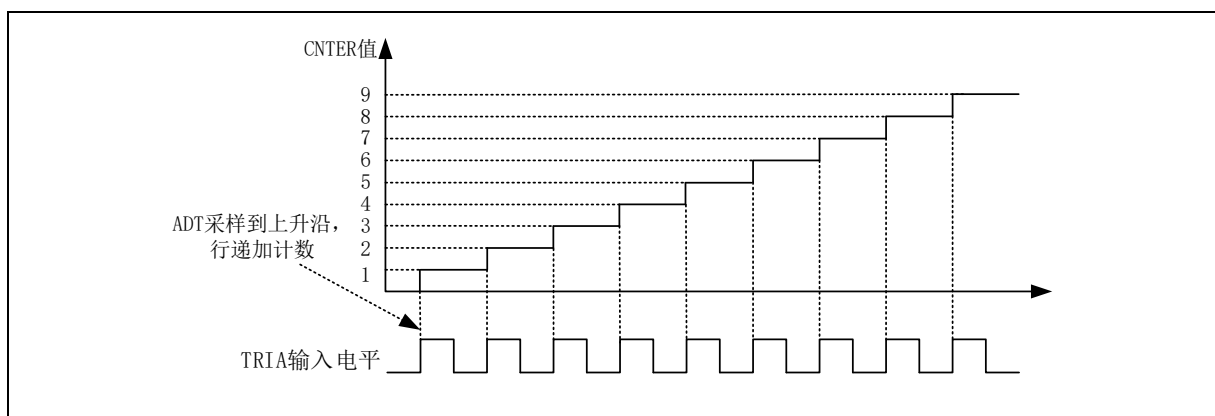


图 4 单路脉冲的单边沿递加计数示意图

如图 4，每次检测到输入电平的上升沿，COUNTER 都会向上计数 1 次。

如果对脉冲的上升沿和下降沿同时计数，需要设置硬件递加计数器的 HCUPR.HCUP8 =1 和 HCUPR.HCUP9 =1，TRIA 端口上采样到上升沿和下降沿都会进行硬件递加计数。那么结果将如图 5 所示：

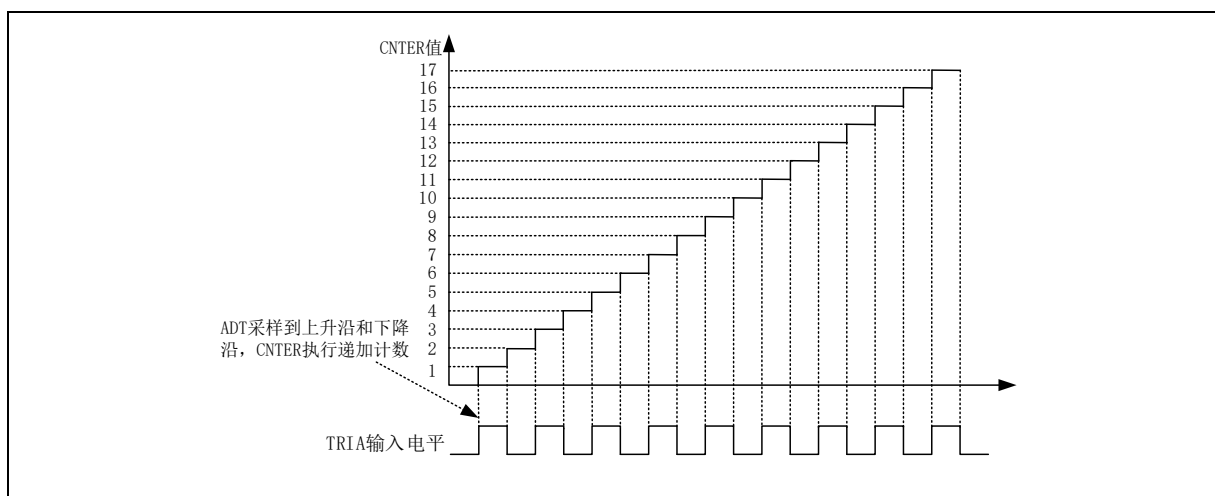


图 5 单路脉冲的双边沿递加计数示意图

在硬件递加、递减计数功能时，内部的计数时钟不起作用，ADT 的通用控制寄存器的 GCONR 的波形模式、计数方式和分频都不用设置。

操作步骤（以 ADT 的 Timer4 和 TRIA(PA15)引脚为例）：

1. 使能 ADT 和 GPIO 外设时钟。
2. 设置 IO：PA15 配置为输入，ADT 的 TTRIG.TRIGAS 设置为 0xC，选择 PA15。
3. Timer4 的周期设置为 0x6000（用户由需求决定）。
4. Timer4 的 TRIA 输入滤波时间设置并且滤波使能（FCONR，该寄存器只在 Timer4 设置，即使使用 Timer5/6 作引脚的捕获输入功能）。
5. 设置 Timer 的硬件递加事件选择寄存器 HCUPR.HCUP8 或者 HCUPR.HCUP9 为 1，在 TRIA 引脚采样到上升沿或者下降沿的时候执行递加计数，也可以两种条件都设置为 1，在输入信号上升沿或者下降沿的时候都执行递加计数。
6. 清零 Timer4 的中断标志，使能上溢中断（ICONR.INTENOVF）和计数比较匹配中断 A（ICONR.INTENA，用户视需求决定是否使用该中断），并且使能 Timer4 的 NVIC 中断。
7. 启动 Timer4。
8. 硬件递加计数到上溢出匹配，产生上溢中断。需要清除对应标志位。

4.2 双路正交信号计数

两路正交脉冲信号通过 CHA 和 CHB 端口输入，ADT 可以做正交计数。设置硬件递加事件选择寄存器 HCUPR 和硬件递减选择寄存器 HCDOR，当 CHA 和 CHB 引脚的电平满足所设置的条件时，ADT 的计数器 CNTER 会执行递加或者递减计数的操作。

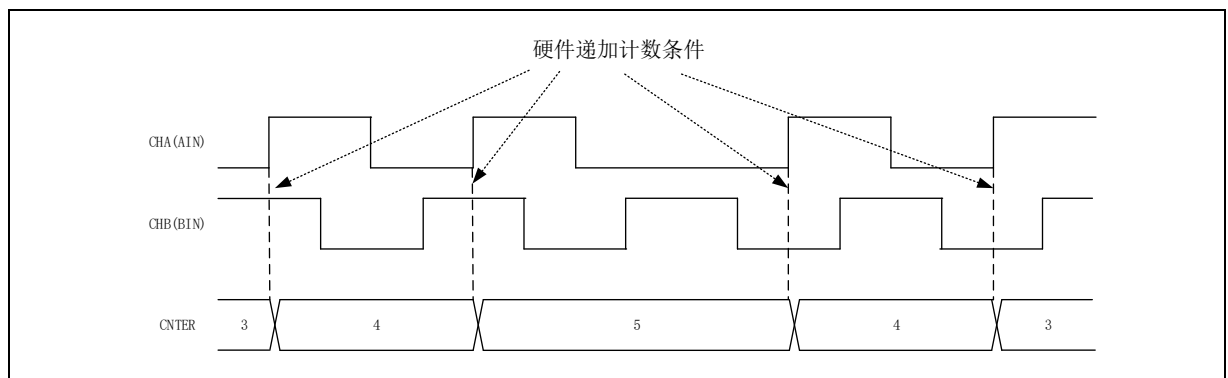


图 6 双路正交信号硬件递加计数示意图

如图 6 所示，硬件递加计数条件为：

HCUPR.HCUP6 =1，当 CHB 端口为高电平时，CHA 端口采样到上升沿电平。

硬件递减计数条件为：

HCDOR.HCDO4 =1，当 CHB 端口为低电平时，CHA 端口采样到上升沿电平。

操作步骤（以 ADT 的 Timer4 和 CHA/B 引脚为例）：

1. 使能 ADT 和 GPIO 外设时钟。
2. 设置 IO：在 GPIO 寄存器内设置 Timer4 的 CHA 和 CHB 引脚。
3. Timer4 的周期设置为 2000（用户由需求决定）。
4. Timer4 的通道 A 和通道 B 设置为输入（寄存器 PCONR）。
5. Timer4 的通道 A 和 B 输入滤波时间设置并且滤波使能（FCONR，该寄存器只在 Timer4 设置，即使使用 Timer5/6 作引脚的捕获输入功能）。
6. 设置 Timer4 的硬件递加事件选择寄存器 HCUPR.HCUP6 为 1，在 CHB 为高电平时 CHA 端口采样到上升沿的时候执行递加计数。
7. 设置 Timer4 的硬件递减事件选择寄存器 HCDOR.HCDO4 为 1，在 CHB 为低电平时 CHA 端口采样到上升沿的时候执行递减计数。
8. 清零 Timer4 的中断标志，使能上溢中断（ICONR.INTENOVF）和下溢中断（ICONR.INTENUDF）并且使能 Timer4 的 NVIC 中断。
9. 启动 Timer4。
10. 硬件递加计数到上溢出值或者 0 匹配，产生上溢或者下溢中断。需要清除对应标志位。

上述仅为最简单的正交信号计数的例子，通过设置 HCUPR 和 HCDOR 条件的组合，可以实现更复杂的正交信号计数功能。

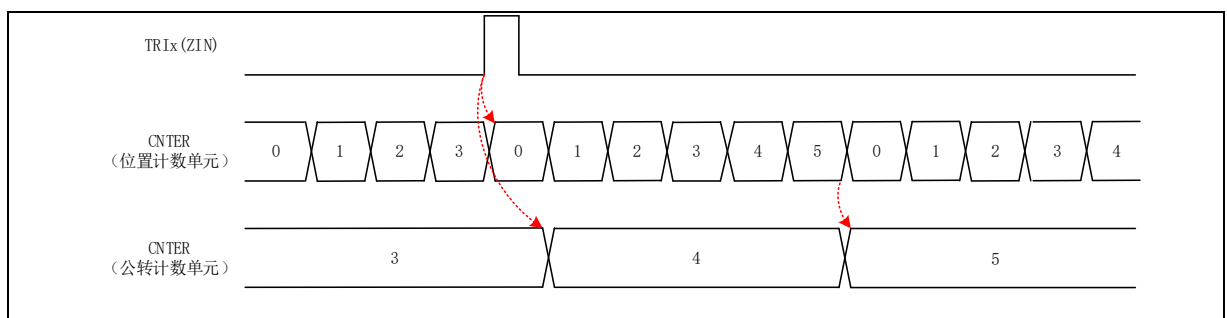


图 7 ABZ 信号数示意图

如图 7 所示，位置计数单元（比如 Timer4）的 CHA 和 CHB 通道计数原理同上，加上另一个 ADT 定时器（比如 Timer5），配合一个 TRIx 引脚，可以完成正交编码的 ABZ 输入信号计数。

1. Timer4 的 COUNTER 的计数由 CHA 和 CHB 的正交信号决定。
2. Timer4 的上溢可以通过 AOS 触发事件，通过 Timer5 的 HCUPR（或 DCDOR）配置，使 Timer5 执行硬件递增（递减）计数。
3. TRIx 引脚的电平信号，可以清零 Timer4 的计数器，也可以通过 Timer5 的 HCUPR 配置，使 Timer5 执行硬件递增计数。

当然，Timer5 可以仅由 ZIN 信号触发硬件递增计数，或仅由 Timer4 的上溢或下溢的 AOS 触发硬件递增（递减）计数，都是由用户视其需求而选择。

5 总结

以上仅仅是对输入信号的一些处理的描述和举例，用户应用中可能会有更多种组合和实践，使用时请以用户手册为参考。结合本笔记，可以加深对该高级定时器 ADT 的理解和运用。

6 参考样例及驱动

华大半导体（HDSC）官方同时提供了该模块的应用样例及驱动库，用户可通过打开样例的工程进一步直观地熟悉该模块以及驱动库的应用，在实际开发中也可以直接参考样例和使用驱动库来快速实现对该模块的操作。

- 样例参考：~/HC32L17X_DDL/example/adt
- 驱动库参考：~/HC32L17X_DDL/driver/.../adt

7 其他信息

技术支持信息：www.hdsc.com.cn

8 版本信息 & 联系方式

日期	版本	修改记录
2020/1/20	Rev1.0	初版发布。



如果您在购买与使用过程中有任何意见或建议，请随时与我们联系。

Email: mcu@hdsc.com.cn

网址: www.hdsc.com.cn

通信地址: 上海市浦东新区中科路 1867 号 A 座 10 层

邮编: 201203

