

运动控制芯片

MCX314

使用手册

南京顺康数码科技有限公司

www.novaelec.com.cn

南京市鼓楼区长江新村 3 幢 504 室

邮编 210037

电话 025-3881012

前 言

■ 在此说明书上使用的特殊用语

激活 — 对某一个信号，此信号持有的功能处于有效状态。

定速 — 固定速度。

定量驱动 — 只输出指定的脉冲量的驱动。

Jerk/变化率 — 单位时间内的加速度/减速度的增加率/减少率。

2 的补码 — 2 进制负数的表示方法（例）16 位长的数据，-1 是 FFFFh，-2 是 FFFEh，-3 是 FFFDh……。
-32768 是 8,000h，这样表示。

■ 在此说明书上使用的特别文字记号

n ○○○○把 X, Y, Z, U 的每个轴信号名写为 n ○○○○，这个“n”表示 X, Y, Z, 及 U。

↑	信号从	Low	电平变至	Hi	电平时的	上升沿
↓		Hi		Low		下降沿

目 录

1	概要	5
2	功能说明	8
2.1	定量驱动和连续驱动	8
2.1.1	定量驱动	8
2.1.2	连续驱动	9
2.2	速度曲线	9
2.2.1	定速驱动	9
2.2.2	直线加/减速驱动	10
2.2.3	S 曲线加/减速驱动	11
2.2.4	驱动脉冲宽度和速度精确度	13
2.3	位置管理	15
2.3.1	逻辑位置计数器和实位计数器	15
2.3.2	比较寄存器和软件限制	15
2.4	插补	16
2.4.1	2 轴/3 轴直线插补	16
2.4.2	圆弧插补	18
2.4.3	位模式插补	20
2.4.4	固定线速度	24
2.4.5	连续插补	25
2.4.6	加减速驱动的插补	27
2.4.7	步进插补（命令，外部信号）	30
2.5	中断	31
2.6	其它功能	32
2.6.1	外部信号控制的驱动操作	32
2.6.2	脉冲输出方式的选择	33
2.6.3	脉冲输入方式的选择	33
2.6.4	硬件限制信号	33
2.6.5	伺服马达对应的信号	34
2.6.6	紧急停止	34
2.6.7	驱动状态的输出	34
2.6.8	通用输出	34
3	引脚配置和信号说明	35
4	读/写寄存器	40
4.1	16 位数据总线的寄存器地址	40
4.2	8 位数据总线的寄存器地址	42
4.3	WR0 命令寄存器	43
4.4	WR1 模式寄存器 1	43
4.5	WR2 模式寄存器 2	44
4.6	WR3 模式寄存器 3	46
4.7	WR4 输出寄存器	47
4.8	WR5 插补模式寄存器	47
4.9	WR6, 7 写数据寄存器 1, 2	48
4.10	RR0 主状态寄存器	48
4.11	RR1 状态寄存器 1	49

4.12	RR2 状态寄存器 2	50
4.13	RR3 状态寄存器 3	51
4.14	RR4,5 输入寄存器 1, 2	51
4.15	RR6,7 读数据寄存器 1, 2	51
5	命令列表	52
6	数据写入命令	54
6.1	范围设定	54
6.2	加/减速度的变化率设定	54
6.3	加速度设定	55
6.4	减速度设定	55
6.5	初始速度设定	55
6.6	驱动速度设定	56
6.7	输出脉冲数/插补终点设定	56
6.8	手动减速点设定	56
6.9	圆弧中心点设定	57
6.10	逻辑位置计数器设定	57
6.11	实位计数器设定	57
6.12	COMP+寄存器设定	57
6.13	COMP-寄存器设定	57
6.14	加速计数器偏移设定	58
6.15	NOP (轴切换用)	58
7	读数据命令	59
7.1	读逻辑位置计数器	59
7.2	读实位计数器	59
7.3	读当前驱动速度	59
7.4	读当前加/减速度	59
8	驱动命令	60
8.1	正方向定量驱动	60
8.2	负方向定量驱动	60
8.3	正方向连续驱动	60
8.4	负方向连续驱动	61
8.5	驱动开始暂停	61
8.6	解除暂停驱动开始状态/结束状态清除	61
8.7	驱动减速停止	61
8.8	驱动立即停止	61
9	插补命令	62
9.1	2 轴直线插补驱动	62
9.2	3 轴直线插补驱动	62
9.3	CW 圆弧插补驱动	62
9.4	CCW 圆弧插补驱动	62
9.5	2 轴位模式插补驱动	63
9.6	3 轴位模式插补驱动	63
9.7	BP 寄存器写入允许	63
9.8	BP 寄存器写入不允许	63
9.9	BP 数据堆栈	63
9.10	BP 清除	63

9.11	插补单步	64
9.12	减速有效	64
9.13	禁止减速	64
9.14	插补中断清除	64
10	I/O 信号连接例子	65
10.1	与 68000CPU 的连接例子	65
10.2	与 Z80CPU 连接的例子	65
10.3	运动系统构成例子	66
10.4	驱动脉冲输出电路	66
10.5	限位等输入信号的连接例子	67
10.6	编码输入信号的连接例子	67
11	控制程序的例子	68
12	电气的特性	71
12.1	DC 特性	71
12.2	AC 延迟特性	71
12.2.1	时钟	71
12.2.2	CPU 读/写周期	72
12.2.3	BUSYN 信号	73
12.2.4	SCLK/输出信号延迟	73
12.2.5	输入脉冲	73
12.2.6	通用 I/O 信号	74
13	I/O 信号时序	74
13.1	上电时序	74
13.2	驱动开始/结束时序	75
13.3	插补驱动时序	75
13.4	驱动开始释放时序	75
13.5	立即停止时序	76
13.6	驱动减速停止时序	76
14	外形尺寸	77
15	规格	78
附录 A:	速度曲线图型	80
附录 B:	指定圆弧插补结束点应注意	84

1 概要

MCX314 是一款能够同时控制 4 个伺服马达或步进马达的运动控制芯片。它以脉冲串形式输出，能对伺服马达和步进马达进行位置控制、插补驱动、速度控制等。以下是它的功能：

■ 独立 4 轴驱动

一个芯片可以分别控制 4 个马达驱动轴的运动。每个轴都可以进行定速驱动，直线加/减速驱动，S 曲线加/减速驱动等。4 轴的性能相同。

■ 速度控制

输出的驱动速度范围是从 1PPS 到 4MPPS。可以运行固定速度驱动，直线加/减速驱动，S 曲线加/减速驱动。加/减速驱动可以使用自动和手动 2 种操作方法。脉冲输出的频率精确度小于 $\pm 0.1\%$ (在 CLK=16MHZ 时) 驱动脉冲输出的速度可以在驱动中自由变更。

■ S-曲线加/减速驱动

每个轴可以用 S-曲线进行加/减速设定，使用 S-曲线命令还可以对抛物线加/减速驱动输出脉冲进行设定。此外，对于定量驱动，我们使用独特的方法避免在 S-曲线加/减速中发生三角波形。

■ 2 轴/3 轴直线插补

可以选择 4 轴中的任何 2 个或 3 个轴进行 2 轴/3 轴的直线插补驱动。插补坐标范围是从当前位置到 -8,388,607 ~ +8,388,607 之间。在整个指定的直线插补范围内，插补精度是 $\pm 0.5\text{LSB}$ 。插补速度范围是从 1PPS 至 4MPPS。

■ 圆弧插补

可以选择 4 轴中的任何 2 个轴进行圆弧插补驱动。插补坐标范围是从当前位置到 -8,388,608 ~ +8,388,607 之间。在整个指定的圆弧曲线插补范围内，插补精度是 $\pm 0.5\text{LSB}$ ，插补速度范围是从 1PPS 至 4MPPS。

■ 2 轴/3 轴位模式插补

收到在高位 CPU 上计算的位模式插补数据后，可以用指定的驱动速度连续输出插补脉冲，用这种方式可以产生任何插补曲线。

■ 连续插补

直线插补→圆弧插补→直线插补→... 这样可以不停地运行每个插补节点的插补驱动，连续插补的最大驱动速度是 2MHz。

■ 固定线速度控制

这是一种在插补驱动中保持插补轴合成速度的功能。2 轴同时输出脉冲时，第 2 轴可以设定为 1.414 倍脉冲周期，3 轴同时输出脉冲时，第 3 轴可以设定为 1.732 倍脉冲周期。

■ 位置控制

每轴都有 2 个 32 位位置计数器，一个是在芯片内部管理驱动脉冲输出的逻辑位置计算器。另一个是管理从外部编码器来的脉冲的实际位置计数器。

■ 比较寄存器和软件限制功能

每轴都有 2 个 32 位比较寄存器，用于逻辑位置计数器或者实际位置计数器的位置大小比较。在驱动时，可以从状态寄存器读出比较寄存器和逻辑/实际位置计数器之间的大小关系。大小关系有变化时，可产生中断。并且，可以启动这 2 个比较寄存器作为软件限位。

■ 由外部信号驱动

每个轴都可以用外部信号进行 +/- 方向运行的定量驱动和连续驱动。这功能使各轴在手动操作时，可以减轻 CPU 的负担，并且可以顺利运行。

■ 搜寻原位输入

每轴有 4 个输入信号，以在驱动中完成减速停止。这些输入信号可以运行原位靠近高速搜寻、原位搜寻、编码器 Z 相搜寻。

■ 伺服马达各种信号

MCX314 接受来自伺服马达驱动器的信号，如 2 相编码器信号，在位信号，报警信号等。

■ 中断发生功能

可以由这些原因产生中断，如：加/减速驱动的定速开始时、定速完毕时、驱动完毕时、位置计数器和比较器之间的大小关系有变化时，等等。此外，连续插补、位模式插补发生下一个数据请求时的中断。

■ 实时监控功能

在驱动中，可以实时读出逻辑位置、实际位置、驱动速度、加速度、加/减速状态（加速中、定速中、减速中）。

■ 8/16 位数据总线

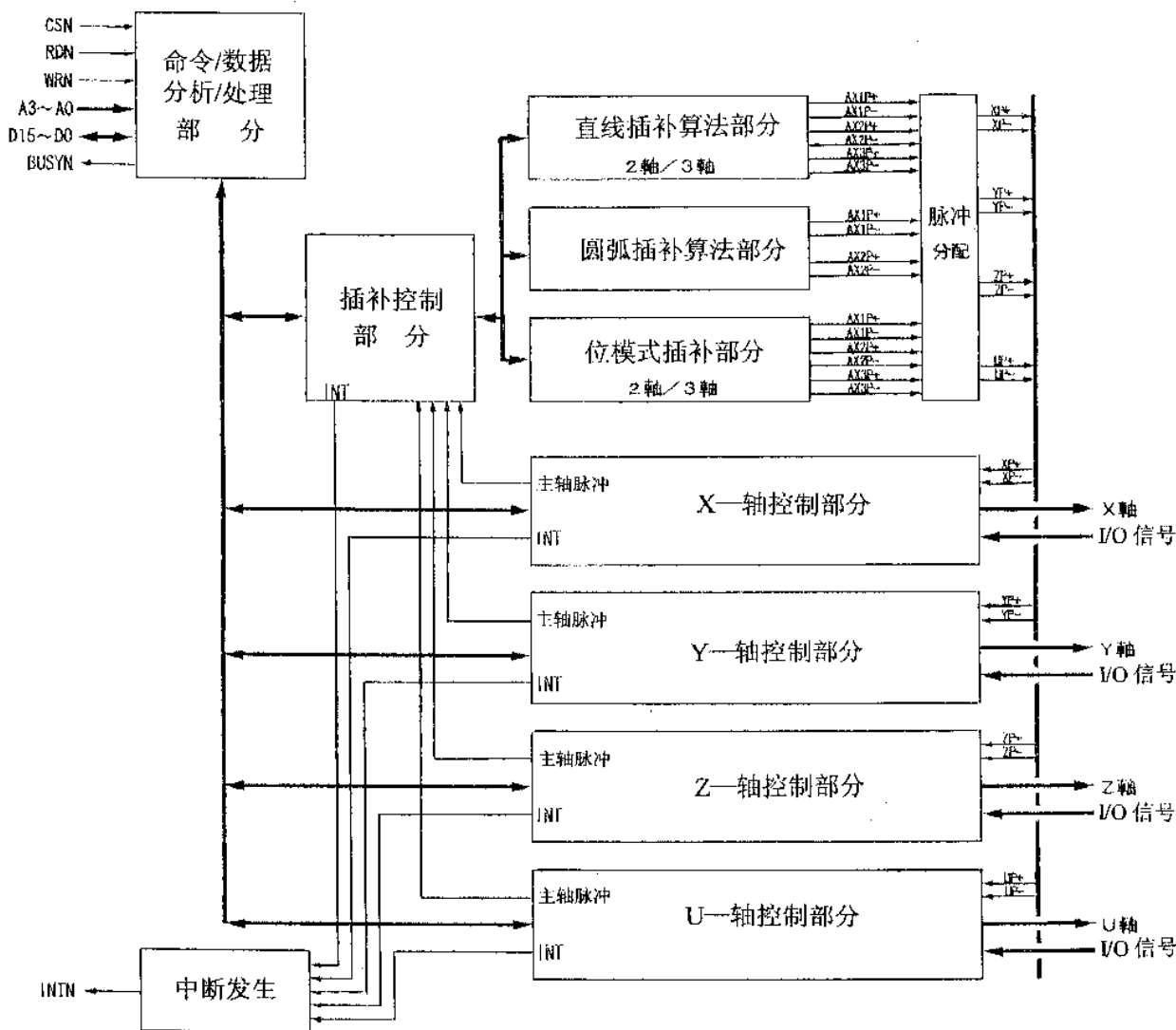
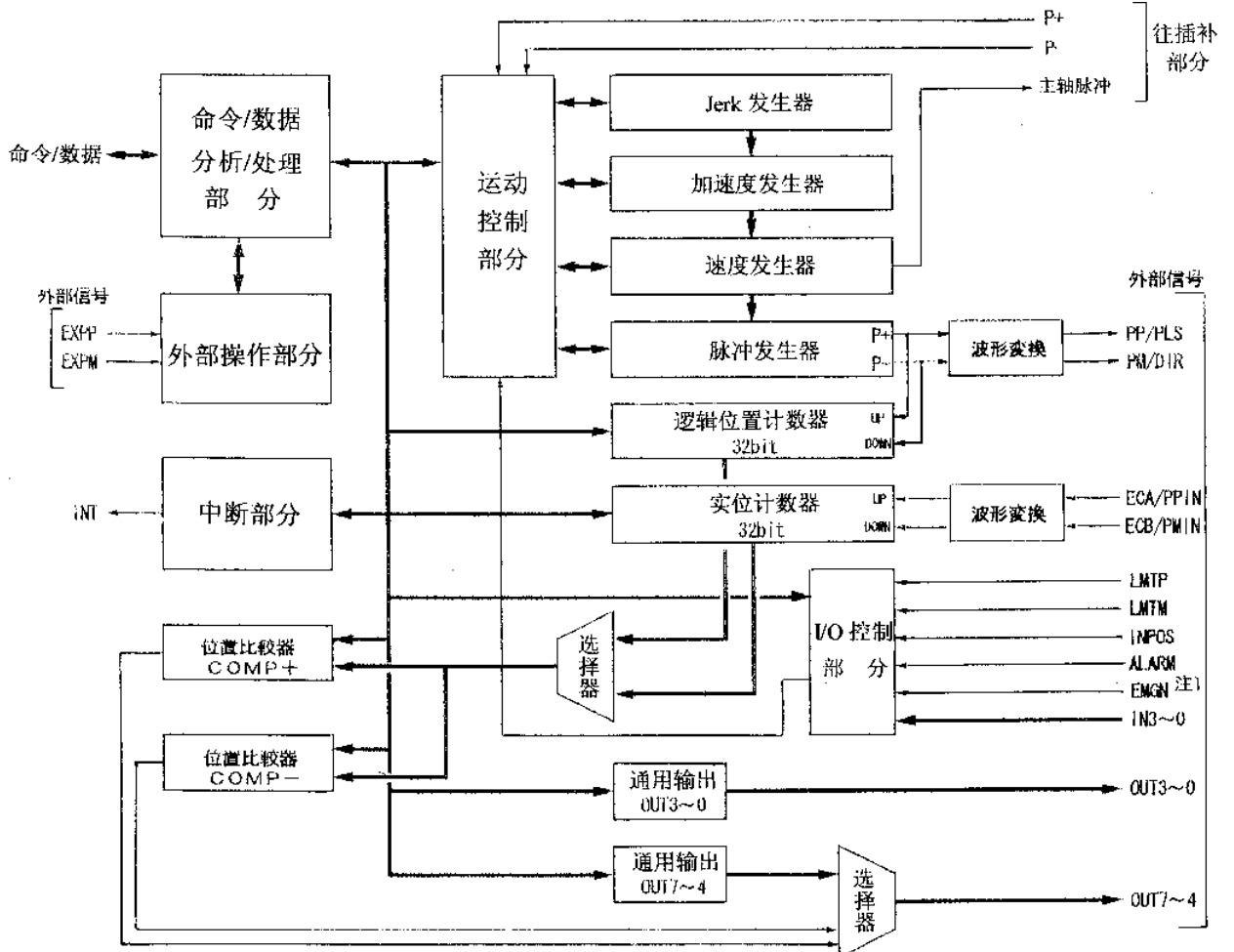


图 1.1 MCX314 功能方框图

8 位/16 位的数据总线都可以和 CPU 连接。

图 1.1 是这 IC 的功能方框图，它包括相同功能的 X, Y, Z, U 轴控制部分和插补算术部分，插补驱动时，在主轴（AX1）上产生的基本脉冲振荡的时序，进行插补算术运算。也可进行定速驱动，加/减速驱动。图 1.2 是每个轴控制部分的功能方框图。



注 1. EMGN 信号各轴共用

图 1.2 每个轴的控制功能方框图

2 功能说明

2.1 定量驱动和连续驱动

各轴的驱动脉冲输出一般使用正方向或负方向的定量驱动命令或者连续驱动命令。

2.1.1 定量驱动

定量驱动的意思是以固定速度或加/减速度输出指定数量的脉冲。需要移动到确定的位置或进行确定的动作时，使用此功能。加/减速的定量驱动如图 2.1 所示，输出脉冲的剩余数比加速累计的脉冲数少时就开始减速，输出指定的脉冲数后，驱动也结束。

进行加/减速的定量驱动，需要设定下列参数：

- 范围 R
- 加/减速 A/D
- 初始速度 SV
- 驱动速度 V
- 输出脉冲数 P

■ 驱动中变更输出脉冲数

定量驱动中可以变更输出脉冲数。加/减速驱动中，开始减速时，如果输出脉冲数有变更的话，重新开始加速（图 2.3）。此外，如果变更的输出脉冲数比已经输出的脉冲数要少的话，立即停止。（图 2.4）。

S 曲线减速如图 2.3 所示，减速时有变更的话，不能正确地运行 S 曲线动作。

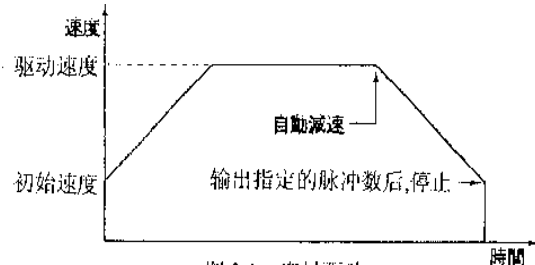


图 2.1 定量驱动

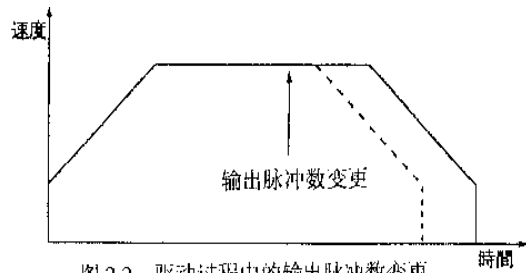


图 2.2 驱动过程中的输出脉冲数变更

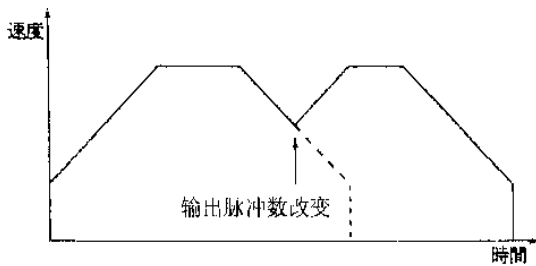


图 2.3 减速时的输入脉冲数变更

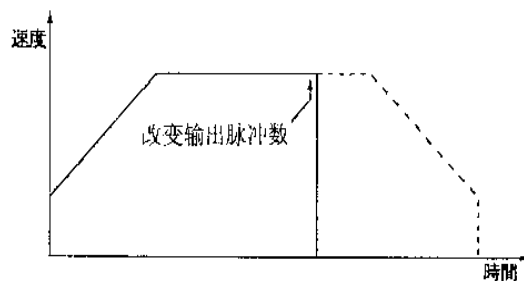


图 2.4 要改变的脉冲数比已输出的脉冲数少的情况

■ 加/减速定量驱动的手动减速

加/减速定量驱动是一般如图 2.1 所示，从芯片计算的减速点开始自动减速。此外，也可以用手动减速。在下列的情况下，芯片不能正确地计算自动减速点或无法算出此减速点，所以需要手动地计算减速点。

- 直线加/减速定量驱动中，需要经常变更速度
- 用加/减速运行圆弧插补、位模式插补、连续插补

需要改为手动减速模式的话，把 WR3 寄存器的 D0 置为 1。根据手动减速点设定命令（07h）设定减速点。其它操作和一般的定量驱动一样。

■ 加/减速定量驱动的加速计数器偏移

对于加/减速定量驱动，加速时，用加速计数器计算加速中累计的脉冲数。当设定的脉冲输出剩余数少于加速计数器的值时，就开始减速。减速时，输出的脉冲数和加速中累计的脉冲数一致时，加速计数器偏移是在加速计数器上加上指定的偏移数值；如右图 2.5 所示，设定偏移数值越大，自动减数点越往前移动。因此，减速完毕的初始速度要延迟。

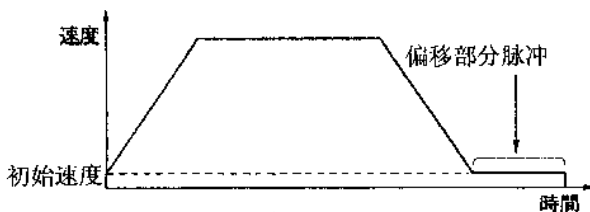


图 2.5 偏移加速计数器

加速计数器偏移在复位时，设定为 8。运行一般的直线加/减速驱动几乎不需要重新设定这个参数。S 曲线加/减速定量驱动中，如果驱动完毕速度降不到初始速度的话，要把加速计数器偏移数值设定为适当的数值，以修正它的速度。

2.1.2 连续驱动

在连续驱动中，连续输出驱动脉冲直至高位的停止命令或外部的停止信号有效。需要运行原点搜寻、扫描操作、控制马达旋转速度时，使用此功能。

有两种停止命令，一个是减速停止，另一个是立即停止。每个轴都有用于减速/立即停止的 IN3~IN0 的外部信号。每个信号都可以设定有效/无效电平。

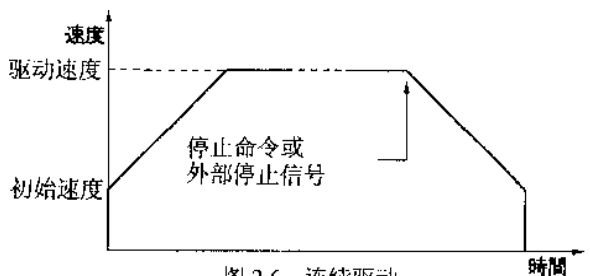


图 2.6 连续驱动

■ 连续驱动的原点搜寻动作

把原点接近信号、原点信号、编码器 Z 相信号等安排在 nIN3~0。在各轴的 WR1 寄存器上设定各信号的有效/无效和逻辑电平。高速搜寻时，用加/减速连续驱动。当设定的有效信号处于激活电平时就减速停止。低速搜寻时，用定速连续驱动。当设定的有效信号处于激活电平时就立即停止。为了以加/减速连续驱动，除了输出脉冲数以外，都要设定和定量驱动一样的参数。

2.2 速度曲线

各轴的驱动脉冲输出一般使用正/负方向的定量驱动命令或连续驱动命令。此外，以设定模式或设定参数来产生定速、直线加/减速、S 曲线加/减速的速度曲线。

2.2.1 定速驱动

定速驱动就是以一成不变的速度输出驱动脉冲。如果设定驱动速度小于初始速度，就没有加/减速驱动，而是定速驱动。使用搜寻原点、编码器 Z 相信号时，找到信号后马上要立即停止的话，不必进行加/减速驱动，而是一开始就运行低速的定速驱动。

为了定速驱动，下列参数应相应预先设定：

- 范围 R
- 初速速度 SV
- 驱动速度 V
- 输出脉冲数 P

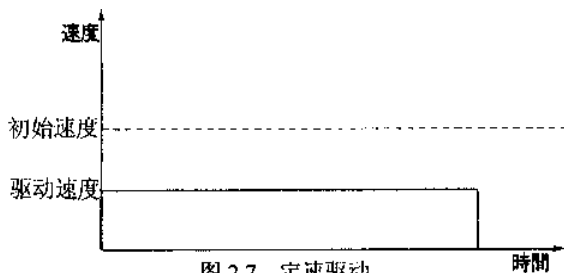


图 2.7 定速驱动

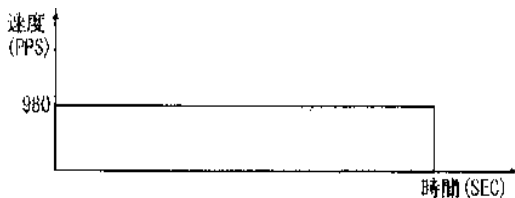
■ 设定参数例子

如右图所示，设定 980PPS 运行定速驱动。

范围 R=8,000,000 ; 倍率=1

初始速度 SV=980 ; 初始速度 ≥ 驱动速度

驱动速度 V=980



有关参数，请参考第 6 章

2.2.2 直线加/减速驱动

直线加/减速驱动是线性地从驱动开始的初始速度加速到指定的驱动速度。

定量驱动时，由于加速的计数器记录加速所累计的脉冲数。当剩余输出脉冲数少于加速脉冲后就开始减速（自动减速）。减速时，将用指定的减速度线性地减速至初始速度。

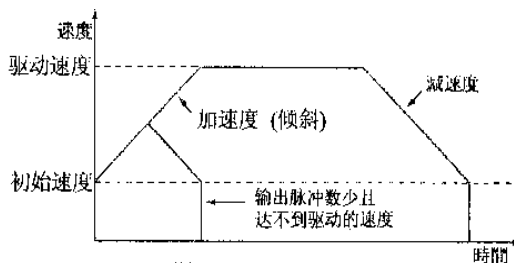


图 2.8 直线加/减速驱动

加速中命令减速停止或在定量驱动中输出脉冲数少于加速至驱动速度所要的脉冲数，如图 2.8 所示，就要在加速过程中开始减速。（有关三角阻止，请参考附录 A-A3）。

通常加速率和减速率基本相同。但是，可以个别设定减速率。个别设定时，不能启用定量驱动的智能减速，而要手动减速。个别设定减速率需要把 WR3 寄存器的 D1 位置为 1，根据减速设定命令（03h）设定减速。

为了直线加/减速驱动，下列参数需预先设定。○记号：需要时，设定。

- 范围 R
- 加速度 A ; 加速度和减速度
- 减速度 D ; 加/减速度个别设定时的减速度
- 初始速度 SV
- 驱动速度 V
- 输出脉冲数 P ; 定量驱动时使用

■ 设定参数的例子

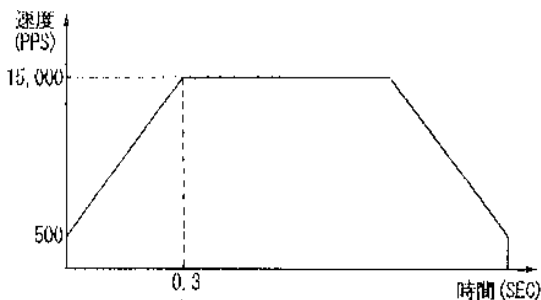
如右图所示，在 0.3 秒之内从初始速度：500PPS 至驱动速度：15,000PPS 直线地加/减速度。

范围 R=4,000,000 ; 倍率=2

加速度 A=193 ; (15000-500)/0.3=48,333PPS/SEC ; (48333/125)/2=193

初始速度 SV=250 ; 500/2=250

驱动速度 V=7,500 ; 15000/2=7,500



有关各参数，请参考第 6 章

2.2.3 S 曲线加/减速驱动

驱动速度加/减速时，此 IC 线性地增加/减少加速度/减速度，以产生 S 型速度曲线。S 曲线加/减速驱动如图 2.9 所示运行。

驱动开始加速时，加速度以指定的加速度的增加率 (K) 从 0 线性增加至指定的数值 (A)。因此，这个速度曲线成为二次级抛物线 (a 区间)。加速度达到指定数值 (A) 后保持此数值。这时速度曲线是直线型的，速度在加速中 (b 区间)。目标速度 (V) 和当前速度的差值比相应时间增加所增加的速度少时，加速度趋向 0。减少率和增加率一样，指定的减速度的减少率 (K) 线性地减少。这时速度曲线成为二次抛物线 (c 区间)。本书定义这种具有部分固定加速度的加速为部分 S 曲线加速。

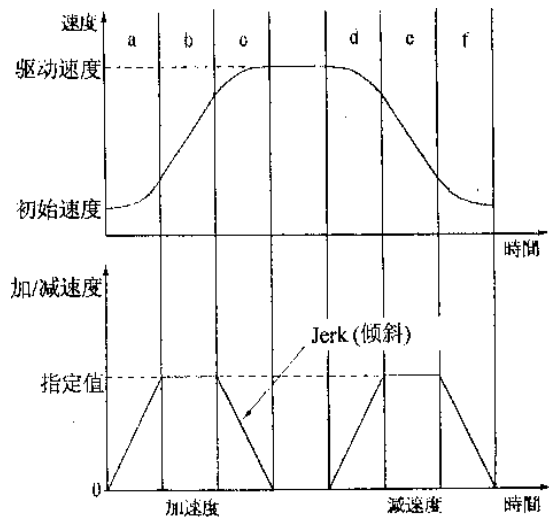


图 2.9 S 曲线加/减速驱动

另一方面，在 a 区间若在加速度达到指定数值 (A) 前，目标速度 (V) 和当前速度的差值比相应时间增加所增加的速度少时，b 区间就消失，只有 a 和 c 区间。这种没有固定加速度的加速称为完全 S 曲线加速。有关部分 S 曲线加速，完全 S 曲线加速的速度曲线例子，请参考之后的参数设定例子及附录 A。速度减速时和加速的内容相同，线性地增加/减少减速度，产生 S 速度曲线。此外，在连续脉冲驱动中，在加速/减速时变更速度也有相同的动作。

要执行 S 曲线加/减速，用户必须将寄存器 nWR3 的 D2 置 1，然后设定下列参数。

- 范围 R
- 加速度/减速度的变化率 K
- 加速度 A ; 加/减速从 0 线性增加至此数值
- 减速度 D ; 分别设定加/减速时，此为减速度指定数值
- 初始速度 SV
- 驱动速度 V
- 输出脉冲数 P ; 定量驱动时使用

■ 定量驱动的三角波形防止功能

就直线加/减速驱动而言，当定量驱动过程中输出脉冲数少于加速至驱动速度所需的脉冲数时或者在加速过程中启动减速停止时，速度曲线会出现三角波形。对于 S 曲线加/减速驱动，即使发生这种情况，为了尽量保持速度曲线的形状可采取以下方式。

初速度为 0 时，以某加速率增加加速度至时间 t，该时间 (t) 的速度由下式表达。

$$V(t) = at^2$$

因此，从 0 至时间 t 累计的脉冲数是把从 0 至时间 t 的速度积分的数值，以下式表述。

$$P(t) = \frac{1}{3}at^3$$

此数值和加速率无关，它表示 $at^2 \times t$ (图 2.10 阴影部分表示脉冲数占单位时间内的脉冲比例) 的 1/3。

对于定量驱动，以某个加速率从 0 至时间 t 增加加速度，以同样的变化率从时间 t 减少加速度，使加速度变为 0，减速时也一样的，以同样

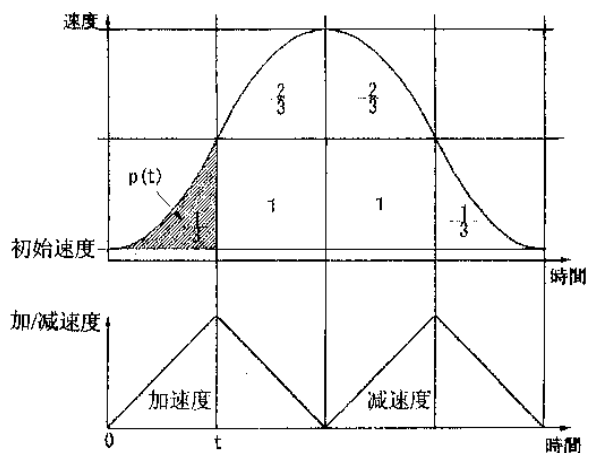


图 2.10 抛物线加/减速的 1/12 法则

的变化率进行增加/减少，在整个过程中累计的脉冲数量。如图 2.10 所示。

$$(1/3 + 2/3 + 1 + 2/3 + 1 + 1/3) = 4 \text{ 个时间单位的脉冲数}$$

因此，从 0 开始至时间 t 的脉冲数 (1/3) 是脉冲总数的 1/12。

对于 S 曲线加减速的定量驱动，加速度增加时的脉冲数大于输出总脉冲数的 1/12 就开始减少加速度。产生如图 2.10 所示的速度曲线。[1/12 法则]

这个方式，若能够达到初始速度 0 的话，产生很理想的曲线，不过实际上绝对达不到 0，所以如图 2.10 所示，剩下从速度 0 至初始速度的脉冲数。这部分脉冲数以最高速度输出。

此外，在加速度的固定加速区域上，加速时的输出脉冲数等于输出总脉冲的 1/4 后，就开始减少加速度。[1/4 法则]

■ 减速停止的三角形防止功能

在直线加/减速驱动上加速时，启动减速停止的话，速度曲线会是三角形。

对于 S 曲线加/减速驱动，应重视速度曲线的形状。正如图 2.11 所示，加速时产生减速停止的话，加速不会马上变减速。加速度减少至 0 后，再开始减速。

■ 运行 S 曲线加/减速驱动时的注意点

- 运行 S 曲线加/减速定量驱动时，驱动中不能变更驱动速度
- 运行 S 曲线加/减速定量驱动时，如果在减速时变更输出脉冲数的话，不能正确地产生 S 曲线
- 运行 S 曲线加/减速时，不能驱动圆弧插补，位模式插补，连续插补。
- 对于 S 曲线加/减速的定量驱动，由于设定参数的原因，在减速时没达到初始速度驱动就结束(①)或出现初始速度拖曳驱动(②)。发生①的情况，用加速计数器偏移来改正。发生②的情况，要改变加/减速的变化率(K)，驱动速度(V)等参数，使情况得以改善。

■ 参数设定例子 1 (完全 S 曲线加/减速)

如右图所示，是在 0.4 秒内用完全 S 曲线加速增加至 40KPPS 的例子。

首先，在计算上不考虑初始速度(把它作为 0)。因为是完全 S 曲线加速，所以在 0.4 秒的 1/2 (0.2 秒)把速度增加至 40KPPS 的 1/2 (20KPPS)，在剩下的 0.2 秒增加至 40KPPS。这时加速度线性地增加直至 0.2 秒。这个积分数值(斜线的面积-右图)等于增加速度 20KPPS。因此，0.2 秒的加速度是 $20000 \times 2 / 0.2 = 200\text{KPPS/SEC}$ ，加速度的增加率是 $200\text{K} / 0.2 = 1,000\text{KPPS/SEC}^2$ 。

运行完全 S 曲线加减速时，速度取决于加/减速度的变化率，所以为了避免产生部分 S 曲线，加/减速度要设定 200KPPS/SEC 以上的数值。

范围	R=800,000	； 倍率=10
加/减速度的增加/减少率	K=625	； $(62.5 \times 10^6) / 625 \times 10 = 1,000 \times 10^3 \text{ PPS/SEC}^2$
加速度	A=160	； $125 \times 160 \times 10 = 200 \times 10^3 \text{ PPS/SEC}$
初始速度	SV=100	； $100 \times 10 = 1,000 \text{ PPS}$
驱动速度	V=4,000	； $4000 \times 10 = 40,000 \text{ PPS}$ 有关各参数，请参考第 6 章

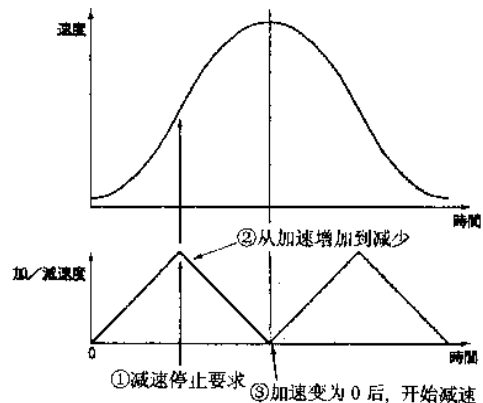
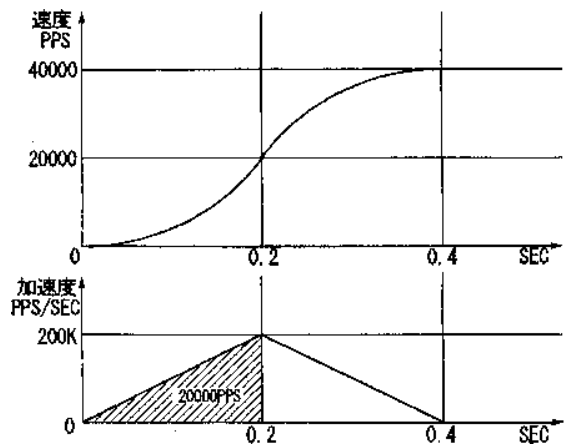


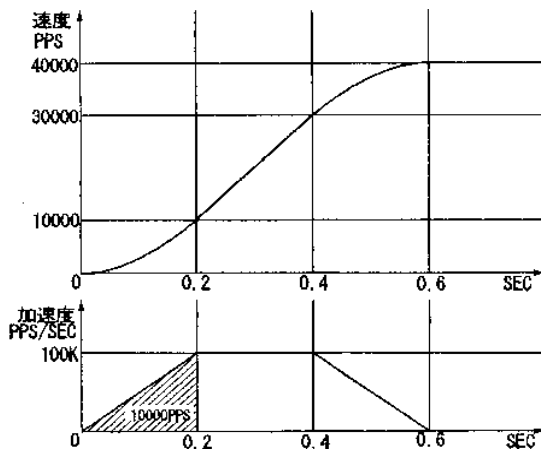
图 2.11 在 S 曲线加/减速的线性加速中减速停止



■ 参数设定例子 2（部分 S 曲线加/减速）

如右图所示，用 0.2 秒抛物线地加速至 10KPPS，再用 0.2 秒从 10KPPS 直线地加速至 30KPPS，再用剩下的 0.2 秒从 30KPPS 抛物线地加速至 40KPPS。这是部分 S 曲线加速例子。

和例子 1 一样，不用考虑初始速度。在 0.2 秒内加速度线性增加至 10KPPS。这时的积分数值（右图的斜线面积）相当于起动速度 10KPPS。因此，0.2 秒的加速度是 $10000 \times 2 / 0.2 = 100\text{KPPS/SEC}$ 。加速度的增加率是 $100\text{K} / 0.2 = 500\text{KPPS/SEC}^2$ 。



范围	R=800,000; 倍率=10
初始速度	SV=100 ; $100 \times 10 = 1,000 \text{ PPS}$
加/减速度的变化率	K=1,250 ; $(62.5 \times 10^6) / 1250 \times 10 = 500 \times 10^3 \text{ PPS/SEC}^2$
加速度	A=80 ; $125 \times 80 \times 10 = 100 \times 10^3 \text{ PPS/SEC}$
驱动速度	V=4,000 ; $4000 \times 10 = 40,000 \text{ PPS}$

2.2.4 驱动脉冲宽度和速度精确度

■ 驱动脉冲的脉冲比率

对于各轴正/负方向上的驱动脉冲，由驱动速度决定的脉冲周期时间有算术上的 $\pm 1\text{SCLK}$ (CLK=16MHz 时为 $\pm 125\text{nSEC}$) 的误差，并且基本上分布在高电平 50%和在低电平 50%。举例如下图所示，设定为 R=8,000,000、V=1,000（倍率=1，驱动速度=1,000PPS），驱动脉冲输出高电平宽度=500 μS ，低电平宽度=500 μS ，周期=1.00mS 的脉冲。

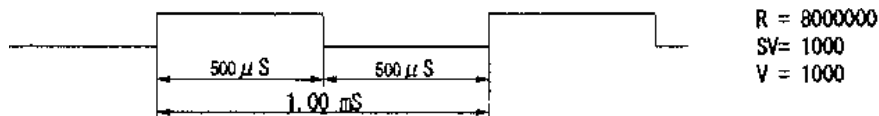


图 2.12 驱动脉冲输出 (1000PPS)

但在加/减速驱动的加速度过程中，由于驱动速度在上升，导致输出脉冲低电平的宽度比高电平的窄。在减速时，低电平的脉冲宽度比高电平的宽。

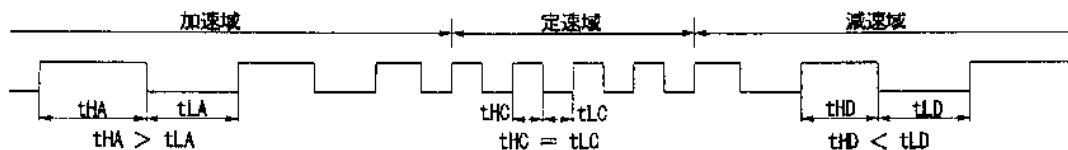


图 2.13 加/减速驱动时的脉宽比较

■ 驱动速度的精确度

产生驱动脉冲的电路是由 IC 内 SCLK 来驱动的。SCLK 由时钟信号 (CLK) 2 分频而得。若 CLK 输入是标准的 16MHz，SCLK 就是 8MHz。要产生某个频率的驱动脉冲，并且是没有抖动的均匀频率驱动脉冲，它的频率周期只能是下图所示 SCLK 周期的整倍数。

	驱动速度(PPS)		驱动速度(PPS)		驱动速度 (PPS)		驱动速度(PPS)
		11	727K	95	84,211	995	8,040
2	4,000M	12	667K	96	83,333	996	8,032
3	2,667M	13	615K	97	82,474	997	8,024
4	2,000M	14	571K	98	81,632	998	8,016
5	1,600M	15	533K	99	80,808	999	8,008
6	1,333M	16	500K	100	80,000	1,000	8,000
7	1,143M	17	471K	101	79,208	1,001	7,992
8	1,000M	18	444K	102	78,431	1,002	7,984
9	889K	19	421K	103	77,670	1,003	7,976
10	800K	20	400K	104	76,923	1,004	7,968

这样就不能随意设定驱动速度。因此，此 IC 采用下述的方式输出任选的驱动速度。

范围设定数值：80,000（倍率：100）、驱动速度设定值：V=4,900、这样设定的话，驱动脉冲输出是 $4900 \times 100 = 490\text{KPPS}$ ，如上表所示，不能用均匀的频率输出 490KPPS 。因此，如下图 2.14 所示，组合 16SCLK 整数倍的 500KPP 频率和 17SCLK 整数倍的 471KPPS 频率输出。 490KPP 的周期是 SCLK（8MHz）周期的 16.326 倍，所以用 674:326 的比率输出 SCLK16 倍周期的脉冲和 17 倍周期的脉冲，每一个单位时间的平均周期做成 16.326。

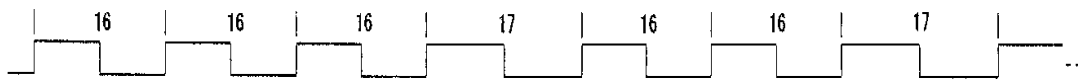


图2.14 对 SCLK 周期的 490KPPS 驱动脉冲周期

依靠这种方法可以非常精确地输出指定速度的驱动脉冲。一般速度倍率越大，可以指定的驱动速度越不精确。不过对于 MCX314，哪怕速度倍率再大，驱动脉冲速度的精确度也在 $\pm 0.1\%$ 以下。用示波器观测驱动脉冲时，若驱动脉冲周期不是时钟周期整倍数的话，如上图所示，在脉冲周期上有 1SCLK（125nSEC）的时差，因此看起来是抖动的，不过此 IC 正是用这个 1SCLK 的时差产生了正确的驱动速度。这个 1SCLK 的时差在运转马达时，会由负荷的惯性吸收。所以这不成问题。

2.3 位置管理

图 2.15 是 1 轴位置控制的方框图。每个轴，有 2 个为了控制当前位置的 32 位可逆计数器和 2 个用以比较当前位置大小的比较器。

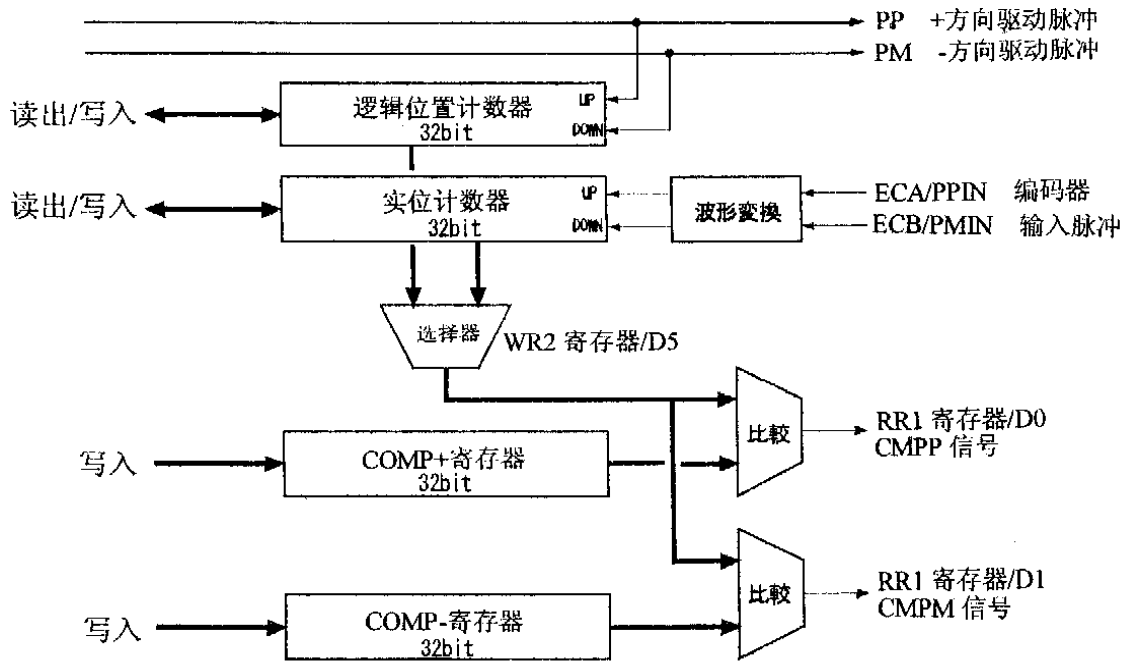


图 2.15 位置管理部分方框图

2.3.1 逻辑位置计数器和实位计数器

逻辑位置计数器是计数 MCX314 中的正/负方向输出脉冲。输出一个正向脉冲时，向上计 1，输出一个负方向脉冲时，向下计 1。

实位计数器计数来自外部编码器的输入脉冲，可以用命令选择输入脉冲的类型，2 相信号或者独立 2 脉冲（向上/下计数）信号。请参考 2.6.3 节。

CPU2 可以在任何时候写入或读出 2 个计数器的数据，计数范围在 $-2,147,483,648 \sim +2,147,483,647$ 之间。负数是 2 的补码格式。在复位时计数值是随意的。

2.3.2 比较寄存器和软件限制

如图 2.15 中所示，每轴有 2 个 32 位寄存器（COMP+，COMP-），能与逻辑位置计数器和实位计数器比较大小。把 2 个比较寄存器的比较对象定为逻辑计数器还是实位计数器，由寄存器 WR2 的 D5 位（CMPSL）选择。COMP+ 寄存器主要用来检测逻辑/实位计数器某个范围的上限。当逻辑/实位计数器的值大于 COMP+ 寄存器的值时，RR1 寄存器的 D0（CMP+）位就置 1。另一方面，COMP- 寄存器用来检测逻辑/实位计数器某个范围的下限。当逻辑/实位计数器数值小于 COMP- 寄存器的数值时，RR1 寄存器的 D1（CMP-）位就置 1。图 2.16 是设定 COMP+=10,000，COMP-=-10,000 的例子。

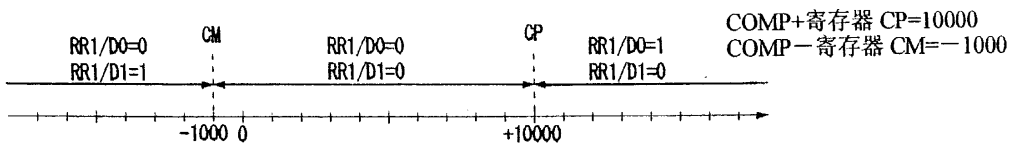


图 2.16 COMP+/-寄存器设定例子

可以把 COMP+寄存器和 COMP-寄存器可用作+/-方向的软件限制来运行。把 WR2 寄存器的 D0 和 D1 (SLMT+, SLMT-) 位设定为 1, 软件限位就有效。在驱动中, 如果逻辑/实位计数器的值大于 COMP+的值就执行减速停止, 并且 RR2 寄存器的 D0 (SLMT+) 位为 1。当执行负方向驱动命令并且逻辑/实位计数器的值小于 COMP+寄存器后, 就会清除这个状态。同样情况适用于负方向的 COMP-。

可以在任何时候写 COMP+寄存器和 COMP-寄存器。复位时, 寄存器的值是任意的。

2.4 插补

本 IC 可在 4 轴中任选 2 轴或 3 轴, 进行直线插补、圆弧插补、模式位插补驱动。指定插补轴是用轴编码方式设定 WR5 寄存器的 D0, D1 (ax1); D2, D3 (ax2); D4, D5 (ax3)。

在插补驱动过程中, 插补运算是在指定主轴 (ax1) 的基本脉冲时序下运行的。因此, 进行插补命令之前, 先要设定指定 ax1 的初始速度, 驱动速度等参数。主轴由 ax1 指定, 在直线插补时, 并不一定是长轴。

设定每个插补命令所需要的参数后, 在 WR0 命令寄存器写入插补驱动命令, 插补驱动立即开始。在插补驱动中, RR0 主状态寄存器的 D8 (I-DRV) 位为 1, 驱动结束后为 0。另外, 在插补驱动中, 在运行插补轴的 n-DRV 位也为 1。

对直线插补、圆弧插补、位模式插补, 可以插补运算最高达 4MPPS。连续插补时, 最高达 2MPPS。

■ 插补时的越限错误

插补驱动时, 每个驱动轴都能进行硬件限制和软件限制。在插补驱动中, 任何轴的限制有动作, 插补驱动就停止。由发生错误而停止, 就读指定插补轴的 RR0 (主状态寄存器) 的错误位。若为 1, 就读其轴的 RR2 (错误寄存器)。

【注意】运行圆弧插补及位模式插补时, 任何方向 (+方向/-方向) 的硬件限制, 软件限制有效, 插补都会停止。因此, 使用圆弧插补及位模式插补要非常小心, 不能脱离限制区域。

■ 伺服马达的在位信号

在插补驱动中, 各轴在位信号 (nINPOS) 一旦有效插补驱动就结束。结束后, 所有轴的 nINPOS 信号处于有效电平, RR0 寄存器的 D8 (I-DRV) 位返回 0。

2.4.1 2 轴/3 轴直线插补

任选 4 轴中的 2 轴或 3 轴, 进行直线插补。

设定相对于当前位置的终点坐标, 写入 2 轴或 3 轴直线插补命令后, 就开始进行线性插补。如图 2.17 所示, 从当前坐标至终点坐标执行直线插补, 终点坐标由针对当前位置的相对数值设定。精确设定每个轴的输出脉冲数。在每个轴独立运行时, 输出脉冲数设定为没有符号的数值。但是, 在插补驱动时, 用相对数值设定对当前位置的终点坐标。

如图 2.17 所示, 对指定直线的位置精确度, 在整个插补范围内有 $\pm 0.5\text{LSB}$ 。右图 2.18 是直线插补的驱动脉冲输出例子, 在设定的终点数值中绝对值最大的轴是长轴。在插补驱动中, 此轴一直输出脉冲, 其它的轴是短轴, 根据直线插补算术的结果, 有时输出脉冲, 有时不输出脉冲。

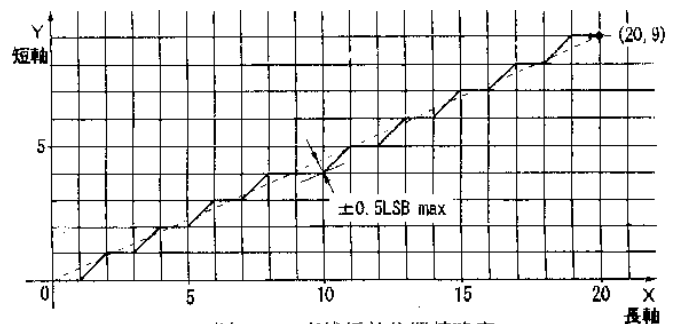


图 2.17 直线插补位置精确度

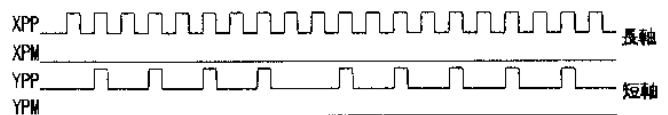


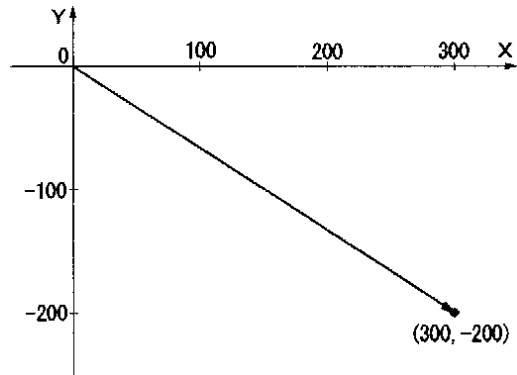
图 2.18 终点(X: 20, Y: 9)驱动脉冲输出例子

直线插补的坐标范围是带符号的 24 位字长。插补范围为从各轴当前位置到-8,388,607~+8,388,607 之间（注意：不能设定-8,388,608）

■ 2 轴直线插补驱动的例子

在 X、Y 轴上，从当前位置至终点坐标（X: +300, Y: -200）进行线性插补，插补驱动速度是 1,000PPS 的定速驱动。

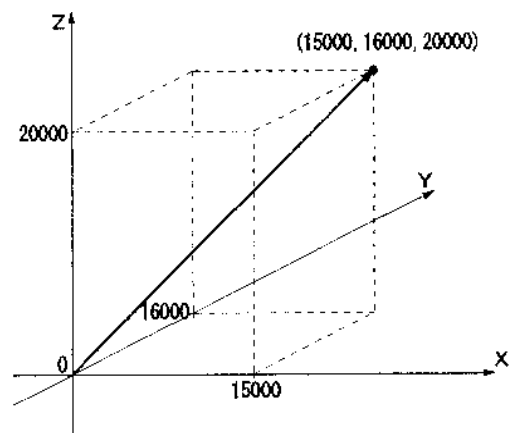
- WR5 ← 0004h 写入； ax1: X 轴、ax2: Y 轴指定
- WR6 ← 1200h 写入； 范围： 8,000,000（倍率=1）
- WR7 ← 007Ah 写入
- WR0 ← 0100h 写入
- WR6 ← 03E8h 写入； 初始速度： 1,000PPS
- WR0 ← 0104h 写入
- WR6 ← 03E8h 写入； 驱动速度： 1,000PPS
- WR0 ← 0105h 写入
- WR6 ← 012Ch 写入； 终点 X 轴： 300
- WR7 ← 0000h 写入
- WR0 ← 0106h 写入
- WR6 ← FF38h 写入； 终点 Y 轴： -200
- WR7 ← FFFFh 写入
- WR0 ← 0206h 写入
- WR0 ← 0030h 写入； 2 轴直线插补驱动



■ 3 轴直线插补驱动的例子

在 X、Y、Z 轴上从当前位置至终点坐标（X: 15,000, Y: 16,000, Z: 20,000）进行 3 轴线性插补。在插补驱动中，初始速度为 500PPS，加速度为 40,000PPS/SEC，驱动速度为 5,000PPS，执行直线加/减速驱动。

- WR5 ← 0024h 写入； ax1: X 轴， ax2: Y 轴， ax3: Z 轴指定
- WR6 ← 1200h 写入； 范围: 8,000,000（倍率=1）
- WR7 ← 007Ah 写入
- WR0 ← 0100h 写入
- WR6 ← 0140h 写入； 加/减速: 40,000PPS/SEC
- WR0 ← 0102h 写入； 40,000/125/1=320
- WR6 ← 01F4h 写入； 初始速度: 500PPS
- WR0 ← 0104h 写入
- WR6 ← 1388h 写入； 驱动速度: 5,000PPS
- WR0 ← 0105h 写入
- WR6 ← 3A98h 写入； 终点 X: 15,000
- WR7 ← 0000h 写入
- WR0 ← 0106h 写入
- WR6 ← 3E80h 写入； 终点 Y: 16,000
- WR7 ← 0000h 写入
- WR0 ← 0206h 写入
- WR6 ← 4E20h 写入； 终点 Z: 20,000
- WR7 ← 0000h 写入
- WR0 ← 0406h 写入
- WR0 ← 003Bh 写入； 减速有效
- WR0 ← 0031h 写入； 3 轴直线插补驱动



2.4.2 圆弧插补

任选 4 轴中的 2 轴进行圆弧插补驱动。

设定相对当前位置（始点）的圆弧中心坐标及终点坐标，在写入 CW 圆弧插补命令或 CCW 圆弧插补命令后执行圆弧插补。用当前坐标（始点）的相对值设定中心坐标及终点坐标。

CW 圆弧插补从当前坐标至终点坐标以顺时针方向绕中心坐标画圆弧，CCW 圆弧插补以逆时针方向绕中心坐标画圆弧。如果终点设为 (0, 0)，能画整个圆。

至于圆弧插补的算法，如图 2.20 所示，由第 1 轴 (ax1) 和第 2 轴 (ax2) 定义一个平面，绕中心坐标把它分为 0~7 的 8 个象限。如图所示，在 0 象限的插补坐标 (ax1, ax2) 上，ax2 绝对值一直比 ax1 的绝对值小，绝对值小的轴为短轴，1、2、5、6 象限是第 1 轴 (ax1) 的短轴，0、3、4、7 象限是第 2 轴 (ax2) 的短轴，短轴在这些象限之间一直输出驱动脉冲，长轴根据圆弧插补运算结果，有时输出脉冲，有时不输出脉冲。

图 2.21 是在当前坐标上指定中心 (-11, 0)，终点 (0, 0)，画半径 11 完整圆的例子。此外，图 2.22 是那个时候输出脉冲的例子。

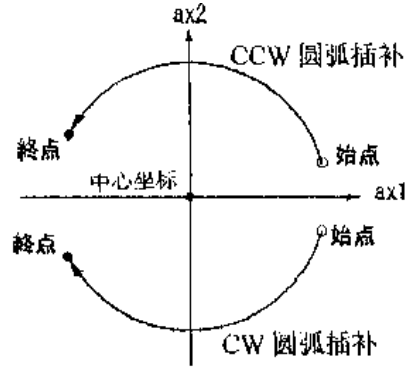


图 2.19 CW/CCW 圆弧插补

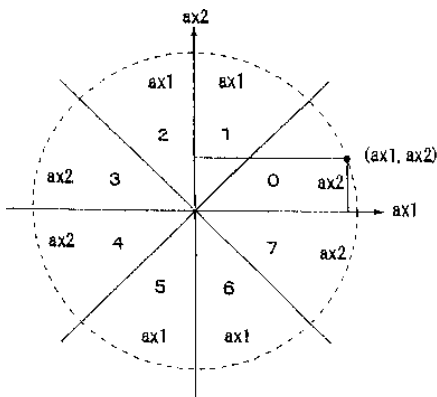


图 2.20 圆弧插补算法的 0~7 象限和短轴

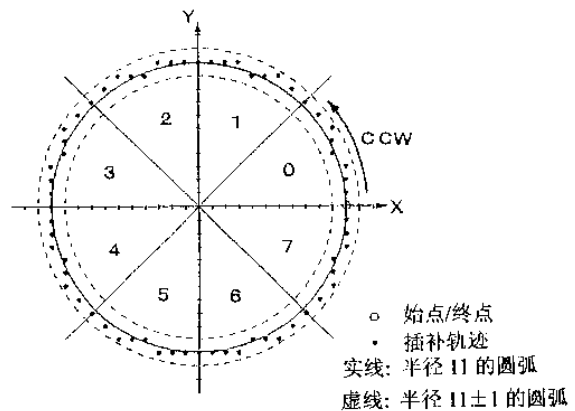


图 2.21 圆弧插补例子

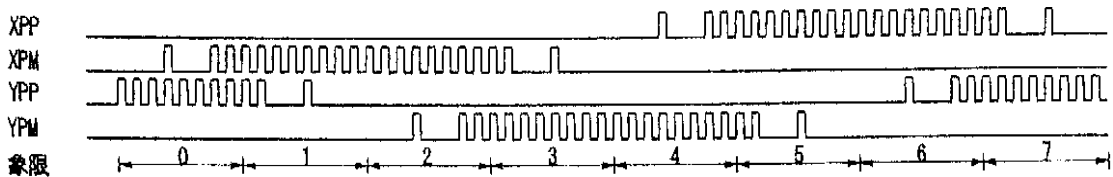


图 2.22 圆弧插补驱动脉冲输出例子

■ 终点判断

对于圆弧插补，在插补驱动开始前，把当前坐标设为 (0, 0)，根据中心坐标的数值决定半径画圆。圆弧算法的误差在插补驱动范围内有 ±1LSB。因此，指定的终点可能不在圆的轨迹上。圆弧插补进入终点所在的象限时，只要结束点值与终点的短轴数值一致，圆弧插补就结束。

图 2.23 是在中心 (-200, 500)，终点 (-702, 099) 从当前位置 (0, 0) 进行 CCW 圆弧插补的例子。由当前位置 (0, 0) 和中心 (-200, 500) 决定半径后，以 CCW 方向进行插补。由于和中心点的位置关系，指定的终点 (-702, 299) 在 4 象限上。插补运行到 4 象限，第 2 轴(ax2)为短轴，第 2 轴的数值达到终点(-702, 299) 的 299 后插补就结束。插补坐标范围是从当前位置至-8,388,608~+8,388,607。对指定圆弧曲线的位置误差在整个插补范围有 ±1LSB。插补速度从 1PPS 到 4MPPS。

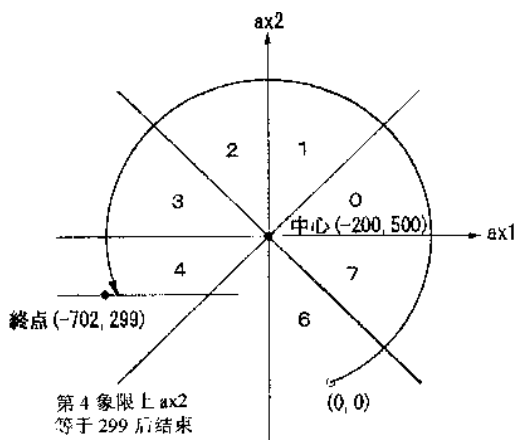
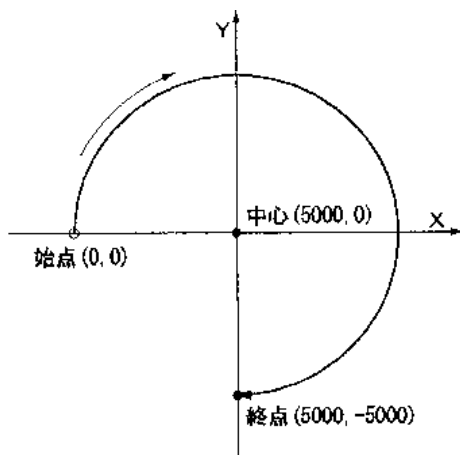


图 2.23 圆弧插补结束判断例子

■ CW 圆弧插补驱动的例子

在 X、Y 轴的中心 (X: 5,000, Y: 0)，终点 (X: 5,000, Y: -5,000) 上，从当前位置 (始点) 进行 CW 圆弧插补。插补驱动速度设定为 1,000PPS 的定速驱动，用固定线速模式进行插补。

- WR5 ← 0104h 写入; ax1: X 轴指定, ax2: Y 轴指定, 固定线速
- WR6 ← 0900h 写入; 范围: 4,000,000(倍率=2)
- WR7 ← 003Dh 写入
- WR0 ← 0100h 写入
- WR6 ← 4DC0h 写入; 2 轴固定线速度的范围:
- WR7 ← 0056h 写入; $4000000 \times 1.414 = 5,656,000$
- WR0 ← 0200h 写入
- WR6 ← 01F4h 写入; 初始速度: $500 \times 2 = 1,000\text{PPS}$
- WR0 ← 0104h 写入
- WR6 ← 01F4h 写入; 驱动速度: $500 \times 2 = 1,000\text{PPS}$
- WR0 ← 0105h 写入
- WR6 ← 1388h 写入; 中心 X: 5,000
- WR7 ← 0000h 写入
- WR0 ← 0108h 写入
- WR6 ← 0000h 写入; 中心 Y: 0
- WR7 ← 0000h 写入
- WR0 ← 0208h 写入
- WR6 ← 1388h 写入; 终点 X: 5,000
- WR7 ← 0000h 写入
- WR0 ← 0106h 写入
- WR6 ← EC78h 写入; 终点 Y: -5,000
- WR7 ← FFFFh 写入
- WR0 ← 0206h 写入
- WR0 ← 0032h 写入; CW 圆弧插补驱动



2.4.3 位模式插补

位模式插补驱动是把高位 CPU 计算的插补数据以数据包的方式接收后，以指定的驱动速度连续输出插补脉冲。

为了进行位模式插补，在每个寄存器上把 2 轴或 3 轴的 +/- 方向驱动脉冲设定为 1 位 1 脉冲，输出驱动脉冲时，设定为 1，不输出脉冲时，设定为 0。

若要运行图 2.24 所示的轨迹的话，输出 X +/- 方向，Y +/- 方向的每个驱动脉冲时，设定为 1，不输出时，设定为 0。设定这样的话，位模式插补数据如下所示。

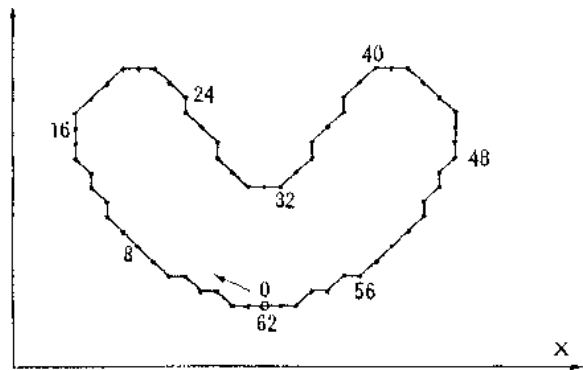


图 2.24 位模式插补例子

```

    ←56←  ←48←  ←40←  ←32←  ←24←  ←16←  ←8←  ←0
01000000 00000000 00011111 11011011 11110110 11111110 00000000 00000000 :XPP(X+方向)
01111111 11110101 00000000 00000000 00000000 00000000 00101011 11111111 :XPM(X-方向)
00000000 00000000 00000000 11111111 00000000 00001111 11111111 11010100 :YPP(Y+方向)
00001010 11111111 11111100 00000000 00111111 11000000 00000000 00000000 :YPM(Y-方向)
  
```

图 2.25 表示位模式插补的第 1 轴寄存器构成。BP1P 寄存器，BP1M 寄存器是从高位 CPU 写入位模式数据的 16 位寄存器（8 位总线时，分别写 H 字节和 L 字节），+ 方向的 16 位的位模式写入在 BP1P 寄存器，- 方向的数据写入在 BP1M 寄存器，位模式插补开始后，从 D0 依次输出驱动脉冲。

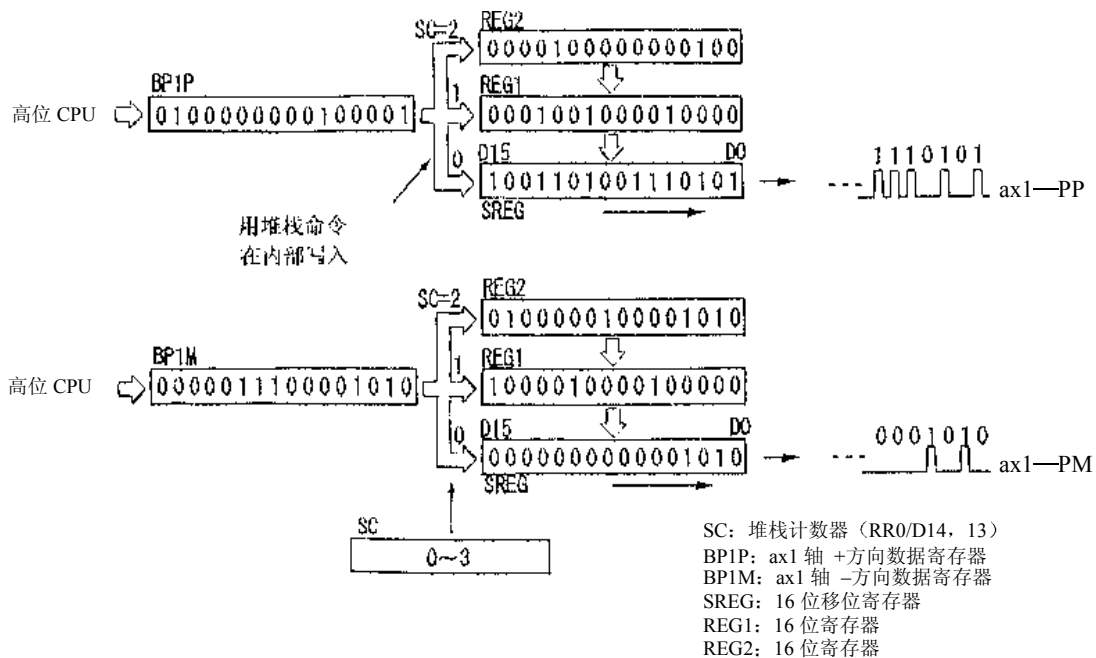


图 2.25 位模式插补寄存器结构图 (ax1 轴分)

堆栈计数器 (SC) 是计算位模式数据储存量的计数器，它能从 0 至 3 变化。RR0 寄存器的 D14, 13 位表示堆栈计数器的数值，没有写入数据时，堆栈计数器 (SC) 表示 0。从高位 CPU 来的 BP 数据堆栈命令使之逐个增加。用 BP 数据堆栈命令将设定在 BP1P, BP1M 寄存器上的数据写入内部的 16 位移位寄

寄存器 (SREG) 或 2 个 16 位寄存器 (REG1, REG2)。堆栈计数器 SC=0 时, 写入在 SREG 上。SC=1 时, 写入在 REG1, SC=2 时, 写入在 REG2, 数据写入完毕后堆栈计数器 (SC) 增加 1。

位模式插补被 2 轴或 3 轴位模式插补命令启动后, 所有轴跟从主轴来的基本脉冲同步, 根据 16 位移寄存器 (SREG) D0 位的值输出驱动脉冲。D0 值为 1 时, 驱动脉冲输出, 为 0 时, 不输出。位移寄存器的 16 位全部输出结束后, 寄存器 REG1 的数据移到移位寄存器, 寄存器 REG2 的数据移到 REG1, 堆栈计数器 (SC) 减 1。

堆栈计数器变为 3 后, 高位 CPU 再也不能把位模式数据压栈到内部。但是, 插补驱动开始后, 随着驱动脉冲输出, 堆栈计数器 (SC) 的数值以 3→2→1 减少。所以, 可以重新写数据, 堆栈计数器 (SC)=0 意味着插补驱动的结束, 所以连续运行位模式插补的话, SC=2 或 1 时, 要设定下一个数据。SC 数值从 2 变到 1 时, 也可以要求中断高位 CPU, 以写入数据。

■ 插补驱动速度的限制

此 IC 的位模式插补驱动速度最高达 4MHz。但是, 数据超过 48 位的话, CPU 要在插补驱动中补充数据, 所以插补驱动速度将受到 CPU 建立模式数据所需时间的影响。

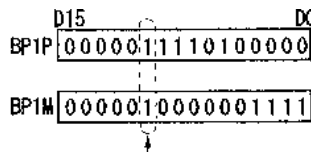
比如, 对于 2 轴位模式插补, 如果 CPU 花费在 4×16 位数据运算和数据建立的时间以及执行 BP 数据堆栈命令的时间是 $100 \mu \text{ SEC}$, 则插补驱动速度在 $1 / (100 \mu \text{ SEC} / 16) = 160 \text{ KPPS}$ 以下。

■ 位模式插补的结束

位模式插补以下述 2 种方法结束

① 在第 1 轴数据内写结束码

当主轴 (ax1) 正和负方向的位数据都置 1 时, 位模式插补就结束。收到结束码后, 堆栈计数器 (SC) 被强制为 0。所有遗留的位模式数据都无效。



主轴的 +方向, -方向变为 1 后结束

② 停止数据写入

停止以 BP 数据堆栈命令写入位模式数据到内部寄存器, 当所有的位模式数据作为驱动脉冲输出后, SC 为 0, 结束插补驱动。

■ 用停止命令中断插补驱动

对运行位模式插补驱动的主轴(ax1)写入立即停止命令或减速停止命令, 插补驱动就停止。重新写入位模式插补命令, 就继续位模式插补。如果以停止命令停止驱动而不需要再进行插补的话, 请用 BP 数据清除命令把所有遗留的数据都清除。

■ 用硬件限制和软件限制的停止

在插补驱动中任何轴有硬件限制, 软件限制动作, 插补驱动就停止。若不需要再运行插补, 就一定要用 BP 数据清除命令把所有留下数据清除。在位模式插补中, 如果在正/负方向上有硬件限制和软件限制动作, 插补就会终止。因此, 位模式插补不能脱离限定区域。

■ 位模式数据的写入寄存器

对于 16 位总线或 8 位总线，从 ax1 轴至 ax3 轴的位模式数据写入寄存器的地址如下图所示。

16 位数据总线的位模式数据写入寄存器的地址

地 址			寄存器名	内 容	在一般情况下的同一个地址寄存器
A2	A1	A0			
0	0	0			WR0
0	0	1			nWR1
0	1	0	BP1P	ax1 轴+方向数据寄存器	nWR2
0	1	1	BP1M	ax1 轴-方向数据寄存器	nWR3
1	0	0	BP2P	ax2 轴+方向数据寄存器	WR4
1	0	1	BP2M	ax2 轴-方向数据寄存器	WR5
1	1	0	BP3P (注 1)	ax3 轴+方向数据寄存器	WR6
1	1	1	BP3M(注 1)	ax3 轴-方向数据寄存器	WR7

注 1: BP3P, BP3M 与 WR6, 7 寄存器共享

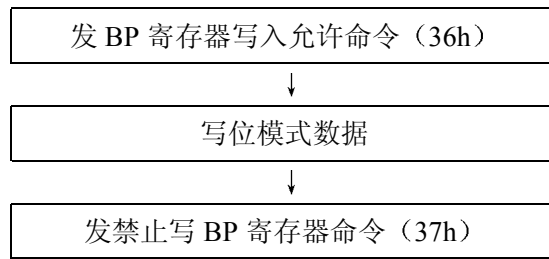
8 位数据总线的位模式数据写入寄存器的地址

地 址				寄存器名	地 址				寄存器名
A3	A2	A1	A0		A3	A2	A1	A0	
0	0	0	0		1	0	0	0	BP2PL
0	0	0	1		1	0	0	1	BP2PH
0	0	1	0		1	0	1	0	BP2ML
0	0	1	1		1	0	1	1	BP2MH
0	1	0	0	BP1PL	1	1	0	0	BP3PL
0	1	0	1	BP1PH	1	1	0	1	BP3PH
0	1	1	0	BP1ML	1	1	1	0	BP3ML
0	1	1	1	BP1MH	1	1	1	1	BP3MH

BPmPL, BPmPH, BPmML, BPmMH 分别表示下述的字节 (m: 1~3)

- BPmPL : BPmP 的低字节 (D7~D0)
- BPmPH : BPmP 的高字节 (D15~D8)
- BPmML : BPmM 的低字节 (D7~D0)
- BPmMH : BPmM 的高字节 (D15~D8)

位模式数据写入寄存器和 nWR2~WR7 寄存器的地址一样，此 IC 复位时，不能写数据到位模式数据寄存器。而要下述步骤写入数据



【注意】在结束写入位模式数据后，如果不发禁止写 BP 寄存器命令 (37h)，就不能写 nWR2~nWR5 寄存器。因此一定要发此命令。

■ 位模式插补驱动的例子

如图 2.24 位模式例子，所示主轴 (ax1) =X 轴，第 2 轴 (ax2) =Y 轴，以 1,000PPS 定速驱动，固定线速度方式进行插补

<p>WR5←0104h 写入 ax1: X 轴 ax2: Y 轴指定，固定线速</p> <p>WR6←0900h 写入； 主轴速度参数设定</p> <p>WR7←003Dh 写入； 范围：4,000,000 (倍率=2)</p> <p>WR0←0100h 写入</p> <p>WR6←4DC0h 写入； 2 轴固定线速的范围</p> <p>WR7←0056h 写入； 4000000×1.414=5,656,000</p> <p>WR0←0200h 写入</p> <p>WR6←01F4h 写入； 初始速度:500×2=1,000PPS</p> <p>WR0←0104h 写入</p> <p>WR6←01F4h 写入； 驱动速度:500×2=1,000PPS</p> <p>WR0←0105h 写入</p> <p>WR0←0039h 写入； BP 数据清除</p> <p>WR0←0036h 写入； BP 寄存器写入允许</p> <p>BP1P←0000h 写入； 0~15 X 轴 +方向</p> <p>BP1M←2BFFh 写入； X 轴 -方向</p> <p>BP2P←FFD4h 写入； Y 轴 +方向</p> <p>BP2M←0000h 写入； Y 轴 -方向</p> <p>WR0 ←0038h 写入； BP 数据堆栈</p> <p>BP1P←F6FEh 写入； 16~31 X 轴 +方向</p> <p>BP1M←0000h 写入； X 轴 -方向</p> <p>BP2P←000Fh 写入； Y 轴 +方向</p>	<p>BP2M←3FC0h 写入； Y 轴 +方向</p> <p>WR0 ←0038h 写入； BP 数据堆栈</p> <p>BP1P←1FDBh 写入； 32~47 X 轴 +方向</p> <p>BP1M←0000h 写入； X 轴 -方向</p> <p>BP2P←00FFh 写入； Y 轴 +方向</p> <p>BP2M←FC00h 写入； Y 轴 -方向</p> <p>WR0 ←0038h 写入； BP 数据清除</p> <p>WR0 ←0034h 写入； 2 轴驱动开始轴位模式插补</p> <p>J1 RR0/D14, 13 读出； 直等到堆栈计数器 2 以下 如果 D14=D13=1 跳到 J1</p> <p>BP1P←4000h 写入； 48~61 X 轴 +方向</p> <p>BP1M←7FF5h 写入； X 轴 -方向</p> <p>BP2P←0000h 写入； Y 轴 +方向</p> <p>BP2M←0AFFh 写入； Y 轴 -方向</p> <p>WR0 ←0038h 写入； BP 数据清除</p> <p>WR0 ←0037h 写入； 禁止 BP 寄存器写入</p> <p>J2 RR0/D8 读出； 等到插补驱动结束 如果 D8=1 跳到 J2</p> <p>注 1： 如果 BP 数据持续下去，重复此过程。</p>	<p>(注 1)</p>
--	--	--------------

■ 用中断的位模式插补驱动

在运行位模式插补驱动中，堆栈计数器（SC）的数值从 2 变到 1 时，可以对高位 CPU 发生中断，并且要求数据写入。为了产生中断，置 WR5 寄存器的 D15 位为 1。如此启动的位模式插补驱动，当堆栈计数器（SC）数值从 2 变 1 时，INTN 输出信号下降到低电平。由高位 CPU 的中断处理子程序来确认堆栈计数器（SC）的数值。如果需要位模式插补的数据，就写入 16 位或 32 位的模式数据。在写入 BP 数据堆栈命令后，中断解除。

对于在插补驱动中发生的中断可以通过写入插补中断清除命令（3Dh）来解除。即使 INTN 输出信号一直处于低电平状态，当插补驱动结束中断也就解除，回到高阻态。

2.4.4 固定线速度

固定线速度控制是始终保持插补轴以合成速度运行的功能。图 2.26 显示了 2 轴插补的轨迹，每个轴根据主轴的基本脉冲输出驱动脉冲。如图所示，当在 X，Y 的 2 个轴上输出驱动脉冲时，移动的距离比只用 1 个轴的长 1.414 倍。因此，若需要保持 2 个轴始终一致的合成速度，其中 1 个轴的速度要设定为另 1 个轴当它作为单轴输出驱动脉冲速度的 1/1.414。

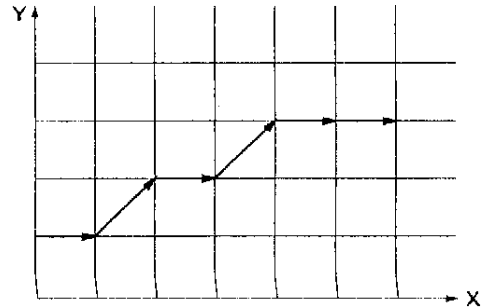


图 2.26 2 轴插补例子

■ 2 轴固定线速度

运行 2 轴固定线速度，先设定 WR5 寄存器的 D9，D8 位为 0，1。把第 2 插补轴的范围参数设定为主轴范围参数的 1.414 倍。此后，只用 1 个轴输出驱动脉冲时，主轴的范围参数就在使用中，用 2 个轴输出驱动脉冲时，第 2 轴的范围参数自动得到使用，脉冲周期延长到 1.414 倍。

■ 3 轴固定线速度

3 轴固定线速度也一样，先设定 WR5 寄存器的 D9，D8 位为 1 和 1。在第 2 轴的范围参数上设定为主轴范围数值的 1.414 倍。在第 3 轴范围参数上设定为主轴范围数值的 1.732 倍。插补驱动开始后只用 3 轴中的 1 个轴来输出驱动脉冲时，主轴的范围参数被使用。用 2 个轴输出驱动脉冲时，第 2 轴的范围参数被使用。用 3 个轴输出驱动脉冲时，第 3 个轴的范围参数被使用，请参考图 2.28。就是 3 轴插补方式，也可以运行由主轴和第 2 轴合成的固定线速度。这时，要设定 WR5 寄存器的 D9，D8 位为 0，1。

■ 固定线速度的插补驱动例子

下图所示，在主轴（ax1）=X 轴、第 2 轴（ax2）=Y 轴上，以 1,000PPS 定速驱动，以固定线速度方式进行直线插补，输出如图 2.27 所示的驱动脉冲。

WR5←0104h 写入；ax1: X、ax2: Y 轴指定、固定线速
WR6←0900h 写入；主轴速度参数设定
WR6←003Dh 写入；范围: 4,000,000(倍率:2)
WR0←0100h 写入
WR6←4DC0h 写入；2 轴固定线速的范围
WR7←0056h 写入； $4000000 \times 1.414 = 5,656,000$
WR0←0200h 写入
WR6←01F4h 写入；初始速度: $500 \times 2 = 1,000\text{PPS}$
WR0←0104h 写入
WR6←01F4h 写入；驱动速度: $500 \times 2 = 1,000\text{ PPS}$
WR0←0105h 写入
WR6←03E8h 写入；终点 X 轴
WR7←0000h 写入；
WR0←0106h 写入

WR6←0190h 写入；终点 Y 轴
WR7←0000h 写入；
WR0←0206h 写入
WR0←0030h 写入；2 轴直线插补开始

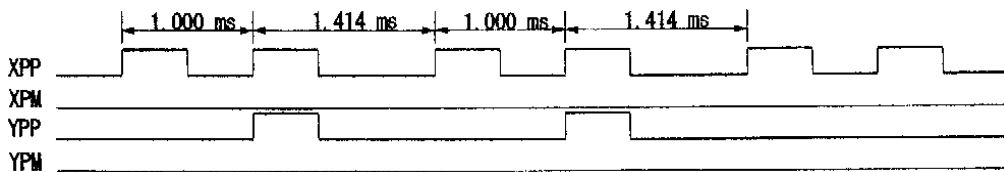


图 2.27 2 轴插补固定线速的例子(线速度:1000PPS)

【注意】2 个轴输出驱动脉冲时，脉冲周期延长到原来的 1.414 倍。驱动脉冲的高电平宽度是不变的，只有低电平在一个周期内比原有的宽 1.414 倍（驱动脉冲时，要设定为正逻辑）。对于 3 个轴固定线速度，其周期是原来的 1.732 倍，同样只是低电平变宽。

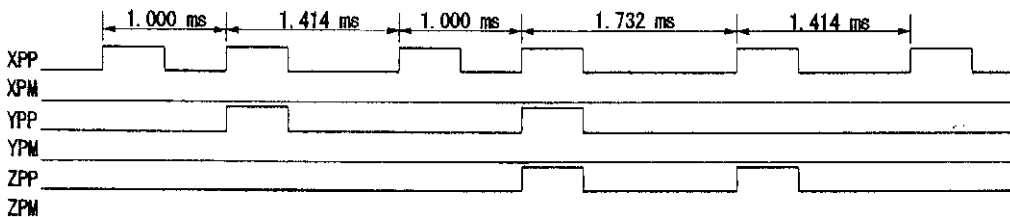


图 2.28 3 轴插补固定线速的例子(线速度:1000PPS)

2.4.5 连续插补

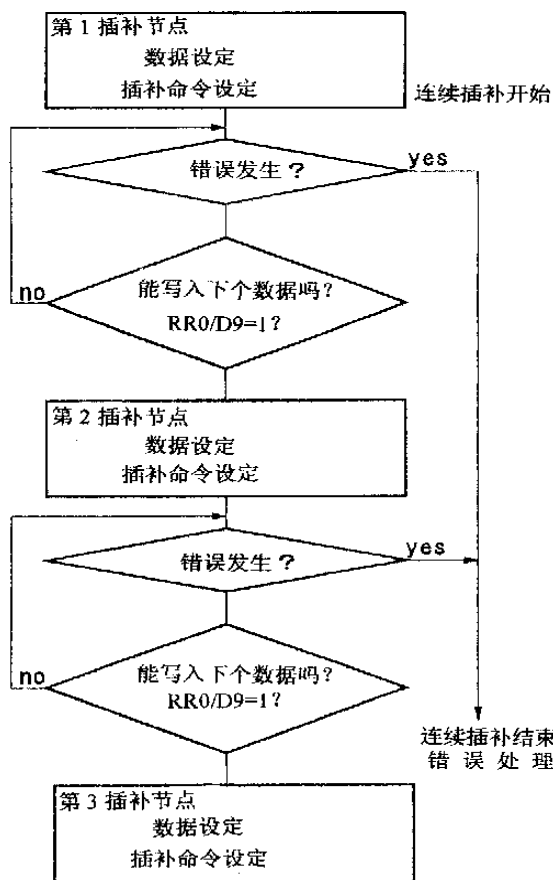
连续插补是直线→圆弧插补→直线插补→…这样在每个插补节点之间不停地驱动，连续插补。在连续插补驱动中，如果设定了下一个插补参数并写入插补命令，就能执行连续的插补驱动。因此，在所有的插补节点中，从连续插补驱动开始至结束的时间必须长于设定下一个插补节点的数据和发命令的时间。

右图是连续插补的操作程序。

RR0 寄存器的 D9 (CNEXT) 位用于连续插补。在插补驱动中这个位表示是否写入下一个插补节点的数据及插补命令,1 表示可以写入,0 表示不可以写入。当驱动停止时,该位为 0;插补驱动开始就变为 1,可以写入下一个插补节点的数据及插补命令。写入下一个插补节点的插补命令后,变为 0(不可以写入状态)。下一个插补节点开始驱动后,变为 1,可以写入再下一个插补节点的数据及插补命令。

■ 用中断的连续插补

WR5 寄存器的 D14 位是设定连续插补的中断允许/禁止位。当该位为 1 并且 RR0 寄存器的 D9 (CNEXT) 位为 1(可以写入)时,INTN 输出信号下降到低电平,在高位 CPU 的中断处理程序中确认 RR0 寄存器的 D9 (CNEXT) 位,如果是 1(可以写入),写入下一个插补节点的数据及插补命令。运行连续插补中断时,写入下一个插补命令后,INTN 信号就回复到高阻态,写入下一个插补节点的数据前,可以用插补中断清除命令解除中断。此外,插补驱动结束后插补中断被强制解除,INTN 信号回复到高阻态。



■ 连续插补中发生的错误

在连续插补驱动过程中，若发生越限等错误，驱动就立即在当前插补节点上停止。在停止的插补节点上下一节点的数据和命令虽在，但命令是无效的。此外，在设定每个插补节点数据及插补命令前必须检查错误。若没有检查，当发生错误停止驱动后，这些数据 and 命令将无效，而从下面第 2 个插补节点开始运行。一定要进行检查，若发现错误的话，要脱离连续插补的循环。

■ 连续插补的注意事项

- 在每个插补节点上要在设定需要的数据后，发插补命令，请不要先送插补命令后送数据。
- 连续插补的驱动速度最高达 2MHz。
- 所有驱动插补节点的时间必须长于插补轴检错和设定下一个插补节点的数据及命令的时间。如果在设定下一个插补节点时驱动停止，RR0 寄存器的 D9 (CNEXT) 位为 0，但是写入下一个插补节点的数据后就从暂停处继续插补。
- 连续插补中有圆弧插补时，圆弧插补终点的短轴数值也许会比真值偏差 ±1LSB，因此为了避免累积每个节点的误差，事先要确认每个圆弧插补的终点，然后考虑怎么运行连续插补。
- 不能运行这样的连续插补，即：2 轴插补 → 3 轴插补 或 3 轴插补 → 2 轴插补。
- 在运行连续插补中不能变更指定的插补轴。

■ 连续插补的例子

图 2.29 是把 (0, 0) 作为始点，从节点 1 至 2, 3... 8 运行连续插补的例子。节点 1, 3, 5, 7 是直线插补，节点 2, 4, 6, 8 是半径 1,500 的 1/4 圆插补，速度是 1,000PPS 的定速驱动，线速不变。

WR5 ← 0104h 写入； ax1: X 轴、ax2: Y 轴指定、线速常数

WR6 ← 0900h 写入； 主轴固定速度设定

WR7 ← 003Dh 写入； 范围: 4,000,000 (倍率:2)

WR0 ← 0100h 写入

WR6 ← 4DC0h 写入； 2 轴固定线速范围

WR7 ← 0056h 写入； $4000000 \times 1.414 = 5,656,000$

WR0 ← 0200h 写入

WR6 ← 01F4h 写入； 初始速度: $500 \times 2 = 1,000\text{PPS}$

WR0 ← 0104h 写入

WR6 ← 01F4h 写入； 驱动速度: $500 \times 2 = 1,000\text{PPS}$

WR0 ← 0105h 写入

WR6 ← 1194h 写入； 终点 X 值: 4,500

WR7 ← 0000h 写入；

WR0 ← 0106h 写入

WR6 ← 0000h 写入； 终点 Y 值: 0

WR7 ← 0000h 写入；

WR0 ← 0206h 写入

WR0 ← 0030h 写入； 2 轴直线插补

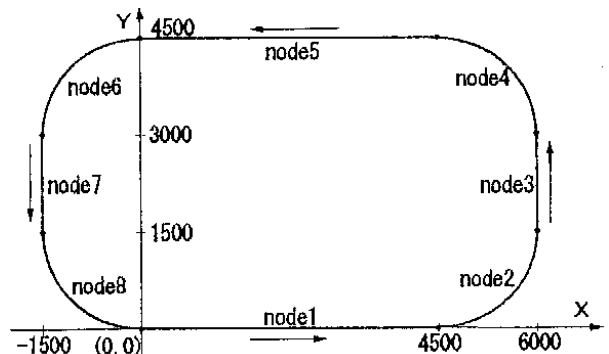


图 2.29 连续插补轨迹的例子

J1 RR0/D5,4 读出； 在 X,Y 轴有错误的话
D5orD4=1 的话,转到 ERROR: 做错误处理
RR0/D9 读出； 下一个节点数据
D9=0 的话,转到 J1； 等写入允许

WR6←0000h 写入； 中心 X 值： 0
WR7←0000h 写入；
WR0←0108h 写入
WR6←05DCh 写入； 中心 Y 值： 1,500
WR7←0000h 写入；
WR0←0208h 写入
WR6←05DCh 写入； 终点 X 值： 1,500 node2
WR7←0000h 写入；
WR0←0106h 写入
WR6←05DCh 写入； 终点 Y 值： 1,500
WR7←0000h 写入；
WR0←0206h 写入
WR0←0033h 写入； CCW 圆弧插补

A 处理

WR6←0000h 写入； 终点 X 值: 0
WR7←0000h 写入；
WR0←0106h 写入
WR6←05DCh 写入； 终点 Y 值: 1,500
WR7←0000h 写入；
WR0←0206h 写入
WR0←0030h 写入； 2 轴直线插补

A 处理

•
•
•

以下 node4~8 继续以同样方法处理第 4 节点到第 8 节点。

2.4.6 加减速驱动的插补

插补一般用定速驱动，不过此 IC 可以用直线加/减速驱动或 S 曲线加/减速驱动（只可做直线插补）运行插补。

在连续插补时为了实现加/减速驱动，使用减速有效命令（3Bh）和减速无效命令（3Ch）。在插补驱动时减速有效命令是使自动减速或手动减速变为有效，减速无效命令是使它变为无效。复位时，都是无效状态。用加/减速单独运行插补驱动时，驱动开始之前一定要设定成减速有效状态。在驱动中写入减速有效命令，也不能变为有效。

■ 2 轴/3 轴直线插补的加/减速驱动

在 2 轴/3 轴直线插补中可以运行直线加/减速驱动及 S 曲线加/减速驱动，这时，自动减速和手动减速都可以用。使用手动减速时，把在终点坐标的各轴数值中绝对值最大的数值设定为主轴的手动减速点。比如：在主轴为 X 轴、第 2 轴为 Y、第 3 轴为 Z 轴上，运行 3 轴直线插补到终点（X: -20,000, Y: 30,000, Z: -50,000）。假定减速时需要的脉冲数量为 5,000，Z 轴的终点绝对数值最大，所以把 50000-5000=45,000 作为手动减速点设定在主轴 X 轴。

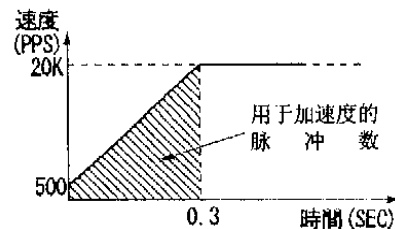
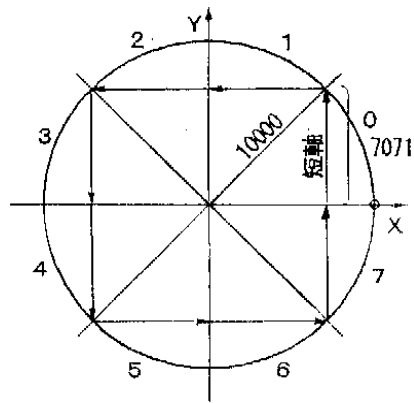
直线插补的加/减速驱动的例子，请参考 2.4.1 的 3 轴直线插补驱动的例子。

■ 圆弧插补，位模式插补的加/减速驱动

在圆弧插补，位模式插补中只能用手动减速的直线加/减速驱动，不能使用 S 曲线加/减速驱动及自动减速。

右图是用直线加减速驱动运行半径 10,000 完整圆轨迹的例子。在圆弧插补中不能用自动减速，所以事先要设定手动减速点。半径 10,000 的圆通过从 0 至 7 象限，在每个象限上短轴一直输出脉冲，所以短轴每 1 个象限输出 $10000/\sqrt{2}=7,071$ 脉冲。因此在整个圆上从主轴输出的基本脉冲数是 $7071 \times 8=56,568$ 。

此外，把初始速度设定为 500PPS，在 0.3 秒之内用直线加速把驱动速度增加至 20,000PPS 的话，加速度是 $(20000-500)/0.3=65,000\text{PPS/SEC}$ 。加速时化费的脉冲数是右图的斜线部面积 $(500+20000) \times 0.3/2=3,075$ 。因此，如果减速度和加速度一样，手动减速点就设定为 $56568-3075=53,493$ 。



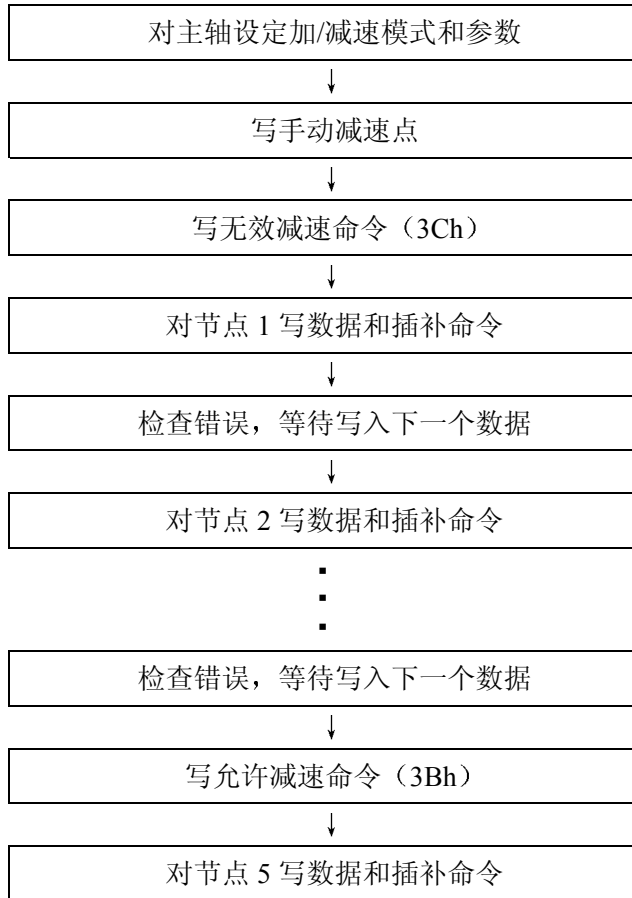
【注意】 在固定线速模式上，不能用这个算法。

- WR3←0001h 写入；减速开始点：手动
- WR5←0004h 写入；插补 ax1: X 轴、ax2: Y 轴指定
- WR6←8480h 写入；范围: 2,000,000(倍率: 4)
- WR7←001Eh 写入
- WR0←0100h 写入
- WR6←0082h 写入；加速度
- WR0←0102h 写入； $130 \times 125 \times 4=65,000\text{PPS/SEC}$
- WR6←007Dh 写入；初始速度: $125 \times 4=500\text{PPS}$
- WR0←0104h 写入
- WR6←1388h 写入；驱动速度
- WR0←0105h 写入； $5000 \times 4=20,000\text{PPS}$
- WR6←D8F0h 写入；中心 X: -10,000
- WR7←FFFFh 写入
- WR0←0108h 写入
- WR6←0000h 写入；中心 Y: 0
- WR7←0000h 写入
- WR0←0208h 写入
- WR6←0000h 写入；终点 X: 0
- WR7←0000h 写入
- WR0←0106h 写入
- WR6←0000h 写入；终点 Y: 0
- WR7←0000h 写入
- WR0←0206h 写入
- WR6←D0F5h 写入；手动减速点: 53,493
- WR7←0000h 写入
- WR0←0107h 写入
- WR0←003Bh 写入；减速有效
- WR0←0033h 写入；CCW 圆弧插补驱动

■ 连续插补的加/减速驱动

在连续插补中只能用手动减速的直线加/减速驱动，不能用 S 曲线加/减速驱动及自动减速。在连续插补中要事先设定手动减速点。这个手动减速点设定在运行减速的最终节点上，并设定从主轴输出的基本脉冲的数值。连续插补时先把减速设定为无效，然后开始插补驱动。在要减速的最终插补节点上写入插补命令之前，写入允许减速命令。开始最终插补节点的驱动时，减速就有效。从最终插补节点的开始计数主轴输出的基本脉冲数大于手动减速点的数值时，减速就开始。

比如，从插补节点 1 至 5 的连续插补中，在最终节点 5 上用手动减速的话，有下述的程序。



- 由从节点 5 开始的主轴基本脉冲数的数值来设定手动减速点。比如，假定减速化费 2,000 脉冲，在节点 5 上输出的基本脉冲的总脉冲数是 5,000 的话，手动减速点就设定为 $5000-2000=3,000$ 。
- 开始至停止一定要在 1 个节点内运行减速，减速停止的最终插补节点需要从其主轴输出的基本脉冲总数要大于在减速中化费的脉冲数。

2.4.7 步进插补（命令，外部信号）

逐步插补驱动的功能是以逐个脉冲的方式执行的。有根据命令和根据外部信号这 2 种方法。如果使用外部信号，除了根据主轴的基本脉冲以外还可以运行跟外部信号同步的插补驱动。步进插补时，插补主轴设定为定速驱动。从每个轴输出的驱动脉冲高电平宽度是 1/2 脉冲周期，这周期由在主轴设定的驱动速度而定。低电平宽度的增加直到下一个命令或外部信号的到来。图 2.30 是由外部信号驱动的步进插补的例子。假定把主轴的初始速度设定为 500PPS，把驱动速度设定为 500PPS 的定速驱动，那么，输出的驱动脉冲的高电平宽度是 1mSEC（驱动脉冲在正逻辑时）。

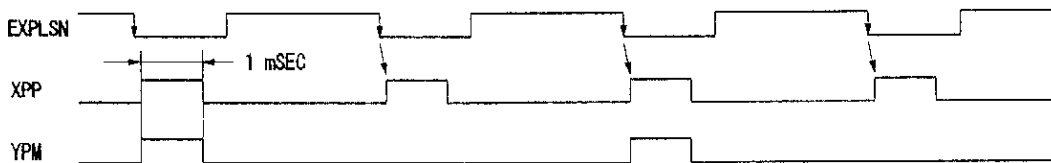


图 2.30 根据外部信号(EXPLSN)的步进插补例子(驱动速度: 500PPS)

■ 根据命令的步进插补

单步插补命令（3Ah）用于插补驱动的步进传送，置 WR5 寄存器的 D12 位为 1，可以由命令进行步进插补，以下是操作程序。

①置 WR5 寄存器的 D12 位为 1

这将成为由命令控制的步进插补模式

②用相同数值设定插补主轴的初始速度和驱动速度

用相同数值设定初始速度和驱动速度将执行定速驱动，这个速度值要设定得比写入单步命令周期快。比如，要用最高 1mSEC 周期写入单步命令，就要把初始速度和驱动速度设定得比 1,000PPS 快的数值

③设定插补数据（终点、中心点等）

④写入插补命令

就是写入插补命令，它也是命令控制的步进插补模式，各轴的驱动脉冲还不被输出

⑤写入单步插补（3Ah）命令

根据插补运算的结果，从各轴输出驱动脉冲。写入单步命令（3Ah），直到插补驱动结束为止。若要在步进插补中停止，可给主轴写立即停止命令（27h），等待 1 个以上驱动速度的脉冲周期后，再写单步插补命令，驱动就停止。插补驱动结束后，写入的单步插补命令都无效

● 外部信号控制的步进插补

引脚（29）EXPLSN 用作步进插补驱动的外部输入信号。置 WR5 寄存器的 D11 位为 1，就可以用外部信号进行步进插补。EXPLSN 输入信号通常为高电平。在插补步进模式时，步进插补由外部信号的下降沿启动。以下是操作程序。

①置 WR5 寄存器的 D11 位为 1

它将成为由外部信号控制的步进插补模式

②用相同数值设定插补主轴的初始速度和驱动速度

用相同数值设定初始速度和驱动速度将执行定速驱动，这个速度值和上述的命令控制一样设定比 EXPLSN 的 Low 脉冲周期快的速度

③设定插补数据（终点、中心点等）

④写入插补命令

就是写入插补命令，它也是命令控制的步进插补模式，各轴的驱动脉冲还不被输出

⑤给 EXPLSN 输入 Low 电平脉冲

脉冲下降 2~5CLK 后，插补驱动脉冲从各轴输出。EXPLSN 的低电平脉冲宽度需要 4CLK 以上。此外，EXPLSN 的脉冲周期一定要比在主轴设定的驱动速度周期长。插补驱动结束前，要反复向 EXPLSN 端输入低电平脉冲。若要在步进插补中停止，就要给主轴写立即停止命令（27h），等待 1 个以上驱动速度的脉冲周期后，再输入 EXPLSN 的低电平脉冲，驱动就停止（用软件复位的方法更简便）

插补驱动结束后，向 EXPLSN 端输入的低脉冲无作用。

【注意】如果由机械接点产生低电平脉冲，不要使 EXPLSN 信号发生颤动。

2.5 中断

发生中断有 2 种情况，其一来自 X、Y、Z、U 各个轴的中断，其二是位模式插补及连接插补时的中断。对 CPU 的中断信号只有 1 个 INTN 信号。因此，下图所示每个轴的中断信号及位模式插补，连续插补的中断信号在 IC 内做逻辑。

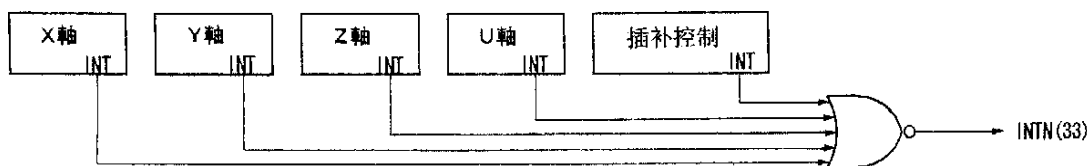


图 2.31 器件内部中断信号的路径

每个轴的中断及插补中断可以分别允许或禁止。复位时，都是禁止状态。

■ X、Y、Z、U 轴的中断

允许/禁止设定 nWR1 寄存器	状态 nRR3 寄存器	中断发生的因素
D8 (PULSE)	D0 (PULSE)	输出了 1 个驱动脉冲(正逻辑脉冲时,由于脉冲上升沿的原因发生)
D9 (P≥C-)	D1 (P≥C-)	逻辑/实位计数器的值大于 COMP-寄存器 (CM) 的值
D10 (P<C-)	D2 (P<C-)	逻辑/实位计数器的值小于 COMP-寄存器 (CM) 的值
D11 (P<C+)	D3 (P<C+)	逻辑/实位计数器的值小于 COMP+寄存器 (CM) 的值
D12 (P≥C+)	D4 (P≥C+)	逻辑/实位计数器的值大于 COMP+寄存器 (CM) 的值
D13 (C-END)	D5 (C-END)	加/减速驱动中，结束在定速区域输出脉冲
D14 (C-STA)	D6 (C-STA)	加/减速驱动中，开始在定速区域输出脉冲
D15 (D-END)	D7 (D-END)	驱动结束时

上图是由 X、Y、Z、U 轴发生中断的因素。

在 nWR1 寄存器内每个中断因素都可被设定成允许（1）或禁止（0）。在驱动中，如果中断因素变成有效，nRR3 寄存器的相应位就为 1，中断输出信号（INTN）变为低电平。高位 CPU 读出被中断轴的 RR3 寄存器后，nRR3 寄存器的状态位被清为 0，中断输出信号（INTN）返回到高阻态。

■ 插补驱动中断

允许/禁止设定 WR5 寄存器	RR0 寄存器 状态位	中断发生的因素 () 内中断清除的方法
D14(CIINT)	D9(CNEXT)	在连续插补驱动中, 可以写下一个插补节点的数据和插补驱动命令 (写入下一个插补驱动命令后中断被清除)
D15(BPINT)	D14,13(BPS1,0)	在位模式插补中堆栈计数值(SC)的数值从 2 变为 1, 可以压入下一个 BP 数据。(压栈 BP 数据后中断被清除)

写入插补中断清除命令 (3Dh), 可以解除在插补驱动中发生的中断。此外, 如果 INTN 输出信号停留在低电平状态, 插补驱动结束后将返回到高阻状态。有关插补驱动中断的使用方法, 请参考位模式插补, 连续插补的相关段落。

2.6 其它功能

2.6.1 外部信号控制的驱动操作

此功能不是用命令, 而是用外部信号来运行定量驱动, 连续驱动。当系统内所控制马达的轴增加很多, 1 个 CPU 还要进行每个轴的手动操作, 这时 CPU 过于繁忙, 不能对每个事件作出应答。此 IC 用外部信号控制的驱动操作功能可以减轻 CPU 的负担。

每个轴都有 nEXPP 和 nEXPM 的 2 个信号输入, nEXPP 信号用于+方向的驱动操作, nEXPM 信号用于-方向的驱动操作。在 WR3 寄存器的 D4 和 D3 位上设定定量驱动或连续驱动。要事先设定好定量驱动或连续驱动所需的参数。nEXPP 和 nEXPM 信号要固定为高电平。

■ 定量驱动模式

把 WR3 寄存器的 D4 和 D3 位设为 1 和 0, 设定驱动所需的速度参数和输出脉冲数。当 nEXPP 信号从高电平降到低电平时, 由信号的下降沿启动正方向的定量驱动。nEXPM 信号从高电平降到低电平时, 由信号的下降沿启动负方向的定量驱动。每个输入操作信号的低电平宽度必须大于 4CLK 周期。驱动结束前, 再来输入信号也无效。

【注意】若输入连接在简单的机械接点上, 在信号上会出现颤动。若输出脉冲数不多, 由于颤动的原因驱动将被再次启动, 所以, 要正确操作就需要防颤电路。

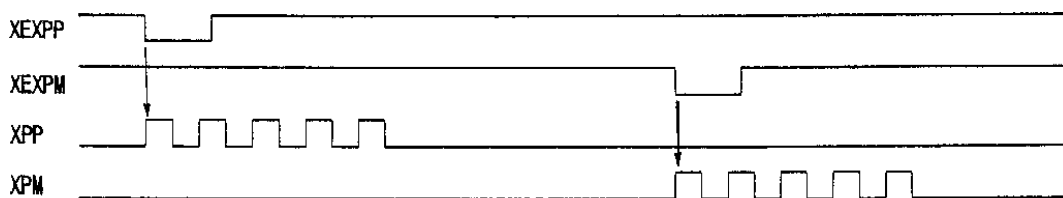


图 2.32 根据外部操作信号输出脉冲 5 的定量驱动例子

■ 连续驱动模式

把 WR3 寄存器的 D4 和 D3 位设为 0 和 1, 设定驱动所需的速度参数, 当 nEXPP 信号从高电平降到低电

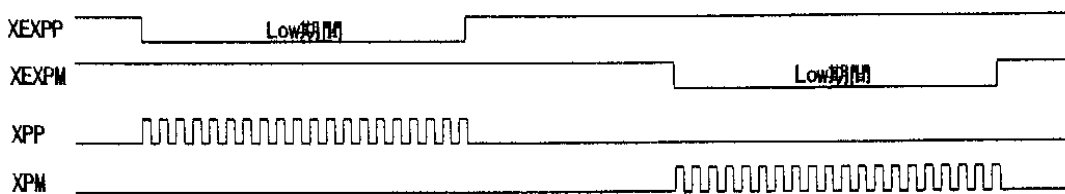


图 2.33 根据外部操作信号连续驱动的例子

平后，在低电平期间在正方向连续输出驱动脉冲。若 nEXPP 信号从低电平返回高电平，对于加/减速驱动就进行减速停止，对于定速驱动就立即停止。nEXPM 信号也如此，连续输出负方向的驱动脉冲。

2.6.2 脉冲输出方式的选择

驱动输出脉冲有下图所示的 2 种脉冲输出方式。以独立 2 脉冲方式正方向驱动时，由 nPP/PLS 输出驱动脉冲，负方向驱动时，由 nPM/DIR 输出驱动脉冲。采用 1 脉冲方式则由 nPP/PLS 输出驱动脉冲，由 nPM/DIR 输出方向信号。WR2 寄存器的 D6 位 (PLSMD) 用于选择脉冲输出方式。脉冲输出、方向输出和逻辑电平都可以由 WR2 寄存器的 D7 (PLS-L)、D8 (DIR-L) 位来选择。

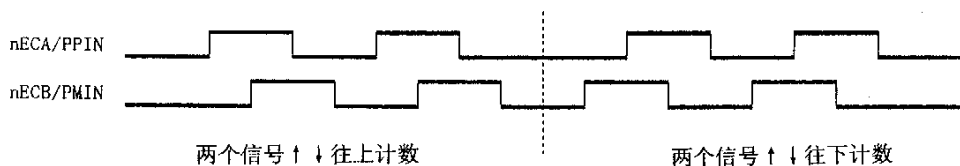
脉冲/方向都是正逻辑设定时

脉冲输出方式	驱动方向	输出信号波形	
		nPP/PLS信号	nPM/DIR信号
独立 2 脉冲方式	+ 方向驱动输出		Low 电平
	- 方向驱动输出	Low 电平	
1 脉冲方式	+ 方向驱动输出		Low 电平
	- 方向驱动输出		Hi 电平

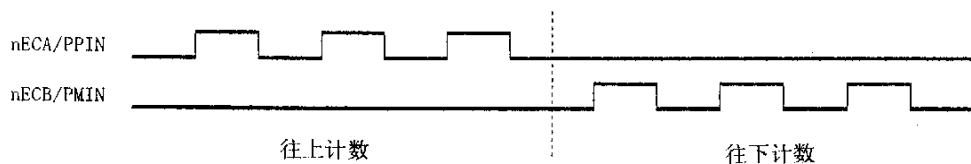
【注意】有关 1 脉冲方式的输出脉冲信号 (nPLS) 和方向信号 (nDIR) 的时序，请参考 13.2、13.3。

2.6.3 脉冲输入方式的选择

当向上/向下计数的实位计数器输入是编码器脉冲时，其输入可以选择 2 相脉冲输入或上/下脉冲输入。设定为 2 相脉冲输入模式后，对正逻辑而言，当 A 相超前 B 相时，向上计数；B 相超前 A 相时，往下计数；由 2 个信号的“↓”、“↑”分别向上和向下计数。此外，对于 2 相脉冲输入模式，可以把输入脉冲分为 1/2 和 1/4。【注意】使用上/下脉冲输入模式时，给 SCLK 周期 nPPIN, nPMIN 信号后要输入，请参考 P67。



设定为上/下脉冲输入方式后，nECA/PPIN 作为向上计数输入，nECB/PMIN 作为向下计数输入。由脉冲的“↑”分别计数。



脉冲输入方式由 WR2 寄存器的 D9 位 (PINMD) 选择，WR2 寄存器的 D11 和 D10 位 (PIND1 和 PIND0) 用来设定编码器 2 相脉冲输入的分周比。

2.6.4 硬件限制信号

硬件限制信号 (nLMTP, nLMTM) 是限定正方向和负方向驱动脉冲的输入信号。当限制信号的逻辑电平和限制信号有效时，可以由命令选择减速停止或立即停止。其设定由 WR2 寄存器的 D3、D4 (HLMT+, HLMT-) 和 D2 (LMTMD) 位。

2.6.5 伺服马达对应的信号

与伺服马达驱动器连接的输入信号有到位信号（nINPOS）和警报信号（nALARM）。每个信号的有效/无效及逻辑电平都可以设定，由 WR2 寄存器的 D15~12 位设定。nINPOS 输入信号与伺服马达驱动的到位（定位完毕）信号对应。设定模式为有效时，一个驱动结束后，为等待 nINPOS 输入信号有效，RR0 主状态寄存器的 n-DRV 位返回 0。nALARM 输入信号接受从伺服马达驱动器的警报信号。设定为有效时，一直监视 nALARM 输入信号，若信号有效，RR2 寄存器的 D4（ALARM）位为 1。若在驱动中，立即停止驱动。这些用于伺服马达驱动器的输入信号可以一直在 RR5, 6 寄存器读其状态。通用输出信号 nOUT7~4 或 nOUT3~0 可用于偏置计数器清除、警报复位等，对于伺服马达驱动器，是警报复位等。

2.6.6 紧急停止

为了紧急停止所有 4 轴的驱动，此 IC 有 EMGN 信号。EMGM 信号通常处于高电平。下降到低电平后，驱动中的全部轴立即停止，各轴的 RR2 寄存器的 D5（EMG）位为 1。EMGN 信号不能选择逻辑电平。

有下述的方法使 CPU 紧急停止 4 轴的运行。

①同时对 4 轴发立即停止命令

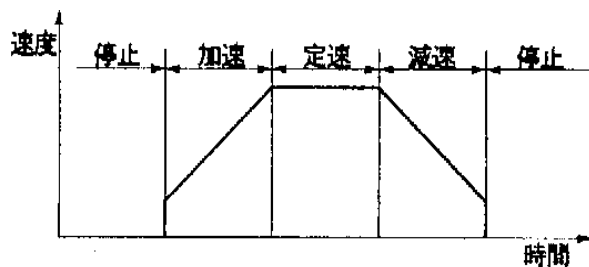
对 WR0 寄存器，指定所有 4 轴写立即停止命令（27h）

②进行软件复位

对 WR0 寄存器写 8,000h 进行软件复位。

2.6.7 驱动状态的输出

每个轴的驱动/停止状态反映在 RR0 寄存器 D3~0（n-DRV）位并由 nDRIVE 信号输出。在驱动过程中，每个轴驱动速度的加速/定速/减速状态反映在每个轴 RR1 寄存器输出 D2（ASND），D3（CNST），D4（DSND）位，并由 nOUT6/ASND，nOUT7/DSND 信号输出。因为输出信号和通用输出信号共享引脚端子，所以当输出驱动状态时，置 WR3 寄存器的 D7（OUTSL）位为 1。



驱动状态	状态寄存器				输出信号		
	RR0/n-DRV	nRR1/ASND	nRR1/CNST	nRR1/DSND	nDRIVE	nOUT6/ASND	nOUT7/DSND
停止	0	0	0	0	Low	Low	Low
加速	1	1	0	0	Hi	Hi	Low
定速	1	0	1	0	Hi	Low	Low
减速	1	0	0	1	Hi	Low	Hi

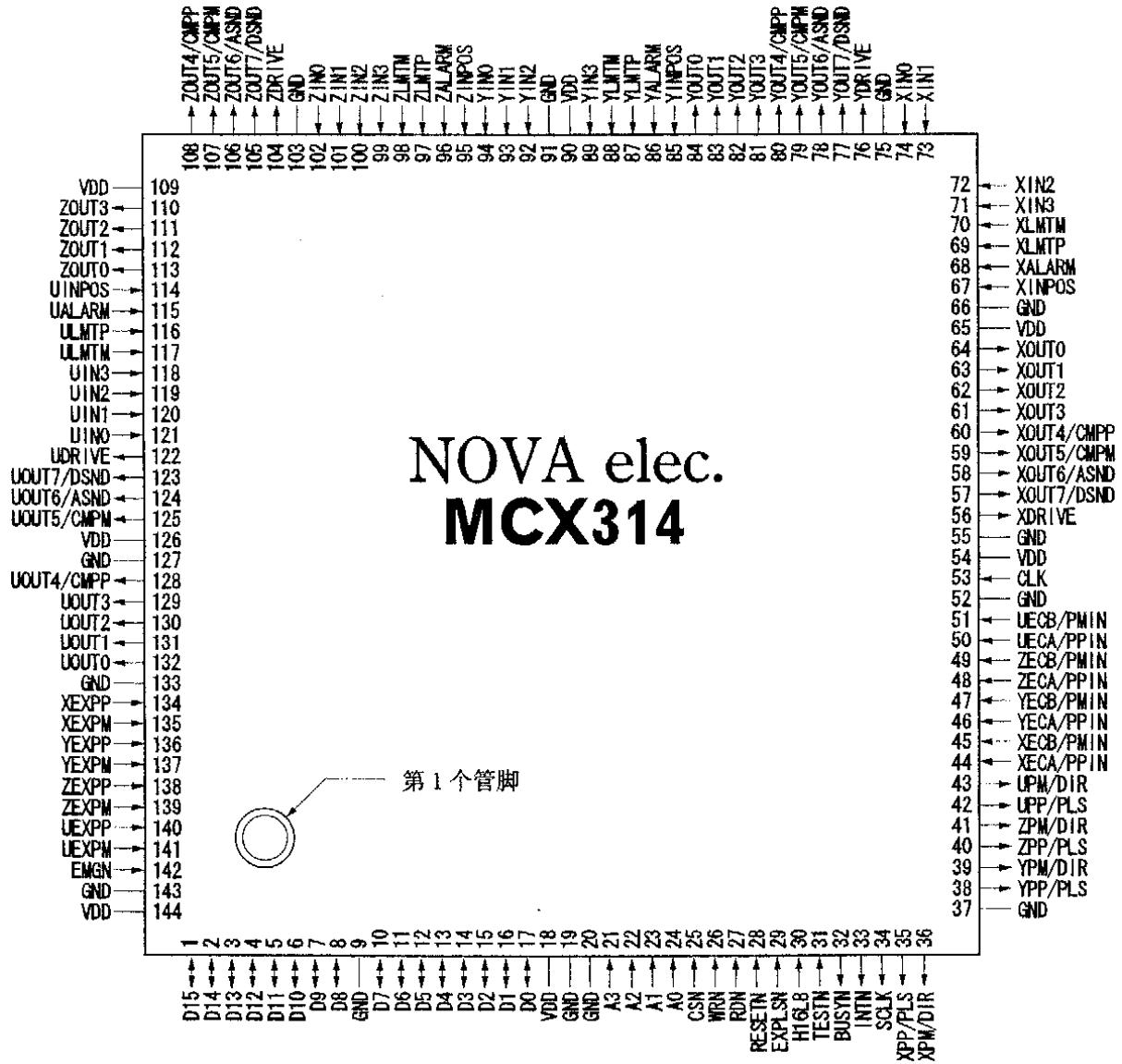
此外，S 曲线加减速驱动中的加速度，减速度的增加/恒定/减少的状态也反映在 RR1 寄存器的 D5（AASND），D6（ACNST），D7（ADSND）位。

2.6.8 通用输出

此 IC 的每个轴都有 nOUT3~0，nOUT7~4 的 8 个通用输出信号。不过，nOUT7~4 引脚和位置比较输出及驱动状态输出的引脚合用，所以使用这些输出时，就不能使用那些信号。

在设定好 WR4 寄存器各位输出电平的值后，由 nOUT3~0 信号输出。使用 nOUT7~4 信号时，通过 WR3 寄存器的 D7 (OUTSL) 设定为使用通用输出的模式。对 WR3 寄存器的 D11~8 (OUT7~4) 设定输出电平的数值后，就输出。通用输出信号可以作为马达驱动器的励磁 OFF，偏差计数器清除，报警复位等。复位时，WR4 寄存器，nWR3 寄存器的每个位被清除，所有的输出为低电平。

3 引脚配置和信号说明



144pin QFP 外形 30.9×30.9mm 管脚距离 0.65mm

■ 信号说明

信号名的 X000, Y000, Z000, U000 是 X 轴, Y 轴, Z 轴, U 轴的输入输出信号, n000 的 “n” 表示 X, Y, Z, U。

信号名	引脚编号	输入/输出	信号说明
CLK	53	输入 A	时钟: 启动此 IC 的内部同步电路的主时钟信号, 其频率为 16.000MHz。驱动速度, 加/减速度, 加速度/减速度的变化率都根据此频率。输入 16MHz 以外的频率时, 速度设定数值, 加减速设定数值都跟上述的不一样。
D15~D0	1~8,10~17	双方向 A	数据总线: 3 态双向的 16 位数据总线。连接在系统的数据总线。当 CSN=Low, RDN=Low 时, 处于输出状态, 除此以外都是高阻的输入状态。 用于 8 位数据总线时, 不使用高位 D15~D8, 所以用高电阻 (100K Ω 左右) 把 D15~D8 上拉到+5V。
A3~A0	21,22,23,24	输入 A	地址: 高位 CPU 选择此 IC 的读/写寄存器的地址信号。用 16 位数据总线时, 不使用 A3。
CSN	25	输入 A	片选: 把此 IC 作为 I/O 器件来选择的输入信号, 读/写此 IC 时, 设定为 Low 电平。
WRN	26	输入 A	写选通: 写此 IC 的写入寄存器时, 设定为 Low 电平, WRN 在低电平时, CSN 及 A3~A0 要确定, WRN 的上升沿将数据总线的内容锁入写寄存器, 所以 D15~D0 的数值在 WRN 上升沿前后要确定。
RDN	27	输入 A	读选通: 从此 IC 的读寄存器读出数据时, 设定为低电平。CSN 和 RDN 都要为低电平。只有在 RDN 处于低电平时, 由 A3~A0 地址选择的读寄存器数据就被输出至数据总线。
RESETN	28	输入 A	复位: 复位 (初始化) 此 IC 的信号。把 RESETN 置为低电平大于 4 周期以上时, 就执行复位。接通电源后一定要用 RESETN 信号复位此 IC。(注意) 如果不输入 CLK, 即使把 RESETN 置为低电平, 也不能复位此 IC。
EXPLSN	29	输入 A	外部脉冲: 外部插补脉冲模式的脉冲输入。一般处于高电平。在外部插补脉冲模式的插补驱动中, 随着 EXPLSN 的下降沿, 启动 1 脉冲的插补算术, 并每个轴的插补脉冲被输出 1 脉冲。EXPLSN 的低电平最小脉宽必须在 4CLK 以上。
H16L8	30	输入 A	H=16 位, L=8 位: 用于 16 位/8 位数据总线选择。高电平时, 为 16 位数据总线, 用 16 位数据总线存取 IC 内的读/写寄存器。而设为 Low 电平后, 只使用数据总线的 D7~D0 8 位, 用 8 位数据总线存取内部读/写寄存器。
TESTN	31	输入 A	测试: 此引脚用于内部电路的测试。如果处于低电平, 会发生意外的动作, 所以要开路或上拉到 5V。
BUSYN	32	输出 B	忙: 表示正处理当前写入的命令。写入命令后正在处理 (命令解析) 其命令, 处于低电平时间在最小 2CLK 与最大 4CLK 之间。BUSYN 处于低电平时, 不能执行写入命令。在命令写入后的 4CLK (CLK=16MHz 的 250nSEC) 之内, 要写下一个命令的高速 CPU 时, 使用此信号。
INTN	33	输出 B	中断: 供高位 CPU 使用的中断请求信号。由任何中断因素引发中断, INTN 都为低电平。中断解除后, 返回高阻态。

信号名	引脚编号	输入/输出	信号说明
SCLK	34	输出 A	系统时钟：输入时钟信号 CLK 二分频后的输出时钟信号。此 IC 内所有的同步电路与此时钟同步。把每个轴的输出信号向外锁存时，可以用它。（注意）在 RESETN 信号处于低电平时，SCLK 无输出。
XPP/PLS YPP/PLS ZPP/PLS UPP/PLS	35 38 40 42	输出 A	脉冲+/脉冲：输出+方向的驱动脉冲。复位时处于低电平。开动驱动后，输出 50%占空比（定速时）的正脉冲。能选择正/负脉冲方式。选择 1 脉冲方式时，从此引脚输出驱动脉冲。
XPM/DIR YPM/DIR ZPM/DIR UPM/DIR	36 39 41 43	输出 A	脉冲-/方向：输出-方向的驱动脉冲。复位时处于低电平。开动驱动后，输出 50%占空比（定速时）的正脉冲。可以用模式选择正/负脉冲。在模式选择上选择 1 脉冲方式时，此引脚是换向信号。
XECA/PPIN YECA/PPIN ZECA/PPIN UECA/PPIN	44 46 48 50	输入 A	编码器 A/脉冲+入：编码器 A 相信号的输入。在 IC 内部与 B 相信号一起变换为上/下脉冲，成为实位计数器的计数输入。用模式选择上/下脉冲输入后，此引脚为上脉冲输入，由输入脉冲的上升沿使实位计数器向上计数。（注意）作上脉冲输入使用时，请参考 P67 图。
XECB/PMIN YECB/PMIN ZECB/PMIN UECB/PMIN	45 47 49 51	输入 A	编码器 B/脉冲-入：编码器 B 相信号的输入。在 IC 内部与 A 相信号一起变换为上/下脉冲，成为实位计数器的计数输入。用模式选择上/下脉冲输入后，此引脚为下脉冲输入。由输入脉冲的下降沿使实位计数器向下计数。（注意）作下脉冲输入使用时，请参考 P67 图。
XDRIVE YDRIVE ZDRIVE UDRIVE	56 76 104 122	输出 A	驱动：表示驱动中的输出信号。执行驱动命令。在输出+方向/-方向的驱动脉冲时此引脚处于高电平。插补驱动指定的轴在插补驱动时也为高电平。通过模式选择使伺服马达用的 nINPOS 信号有效，在 nINPOS 有效前，DRIVE 一直保持高电平。
XOUT7/DSND YOUT7/DSND ZOUT7/DSND UOUT7/DSND	57 77 105 123	输出 A	通用输出 7/下降：通用输出信号 nOUT7~4 的输出端。在 WR0 寄存器指定轴后，在 WR3 寄存器的 D11~D8 写入 I/O 数据。nOUT7~4 依数据输出高或低电平。复位时为低电平。在驱动状态输出模式时，此引脚表示减速驱动状态的输出信号，当驱动处于减速状态时，就为高电平。
XOUT6/ASND YOUT6/ASND ZOUT6/ASND UOUT6/ASND	58 78 106 124	输出 A	通用输出 6/上升：通用输出信号。操作与 nOUT7 一样。在驱动状态输出模式时，此引脚表示加速驱动状态的输出信号。当驱动处于加速状态时，就为高电平。
XOUT5/CMPM YOUT5/CMPM ZOUT5/CMPM UOUT5/CMPM	59 79 107 125	输出 A	通用输出 5/比较-：通用输出信号，操作与 nOUT7 一样。在处于驱动状态输出模式时，逻辑/实位计数器小于 COMP-寄存器就输出高电平，大于的话，输出低电平。
XOUT4/CMPP YOUT4/CMPP ZOUT4/CMPP UOUT4/CMPP	60 80 108 128	输出 A	通用输出 4/比较+：通用输出信号，操作与 nOUT7 一样。在处于驱动状态输出模式时，逻辑/实位计数器大于 COMP+寄存器就输出高电平，小于的话，输出低电平。
XOUT3~0 YOUT3~0 ZOUT3~0 UOUT3~0	61~64 81~84 110~113 129~132	输出 A	通用输出 3~0：每个轴有 4 个通用输出信号。根据 WR4 寄存器 D15~D0 的“1”或“0”数据，nOUT3~0 输出高或低电平。复位时为低电平。因为不用指定轴，所以设定比 nOUT7~4 简单。

信号名	引脚编号	输入/输出	信号说明
XINPOS YINPOS ZINPOS UINPOS	67 85 95 114	输入 A	到位：处理伺服马达驱动输出到位（定位完毕）的输入信号。用命令设定有效/无效和逻辑电平。设定为有效，当驱动结束后待此信号成为有效后，主状态寄存器的 n-DRV 位返回 0。
XALARM YALARM ZALARM UALARM	68 86 96 115	输入 A	伺服报警：处理伺服马达驱动警报输出的输入信号。可以选择有效/无效模式和逻辑电平。设定为有效，当此信号有效，RR2 寄存器的 ALARM 位为 1。
XLMT+ YLMT+ ZLMT+ ULMT+	69 87 97 116	输入 A	越限+：正方向的越限信号。在正方向驱动脉冲输出过程中此信号有效，就执行减速停止或立即停止。有效脉冲宽度需大于 2CLK 脉冲。在方式选择中，可设定减速停止或立即停止和逻辑电平。此信号有效时，RR2 寄存器的 HLMT+ 位为 1。
XLMT- YLMT- ZLMT- ULMT-	70 88 98 117	输入 A	越限-：负方向的越限信号。在负方向驱动脉冲输出过程中此信号有效，就执行减速停止或立即停止。有效脉冲宽度需大于 2CLK 脉冲。在方式选择中，可设定减速停止或立即停止和逻辑电平。此信号有效时，RR2 寄存器的 HLMT- 位为 1。
XIN3~0 YIN3~0 ZIN3~0 UIN3~0	71~74 89.92~94 99~102 118~121	输入 A	输入 3~0：4 输入信号用于对各驱动轴执行减速停止或立即停止。还用于位置搜索的输入信号。有效脉冲宽度需要 2CLK 以上。可以对每个 IN3~IN0 设定有效/无效和逻辑电平。信号状态可以一直在 RR4/RR5 寄存器读出。
XEXPP+ YEXPP+ ZEXPP+ UEXPP+	134 136 138 140	输入 A	外部操作+：从外部启动正方向驱动的输入信号。在外部定量驱动模式时，由此信号的下降沿启动定量脉冲驱动。此外，在外部连续模式时，此信号处于低电平时，运行正方向连续驱动。
XEXPM- YEXPM- ZEXPM- UEXPM-	135 137 139 141	输入 A	外部操作-：从外部启动负方向驱动的输入信号。在外部定量驱动模式时，由此信号的下降沿启动定量脉冲驱动。此外，在外部连续模式时，此信号处于低电平时，运行负方向连续驱动。
EMGN	142	输入 A	紧急停止：紧急停止各轴驱动的输入信号。此信号处于低电平，包括插补驱动在内的各轴驱动被立即停止，每个轴的 RR2 寄存器 EMG 位就为 1。低电平脉冲宽度需要 2CLK 以上。（注意）此信号不能选择逻辑电平。
GND	9.19.20.37 52.55.66 75.91.103. 127.133.143		接地端（0V），必须将所有 13 个引脚连接到地。
VDD	18.54.65 90.109.126. 144		+5V 电源端，必须将所有 7 个引脚连接到+5V 电源。

■ 输入/输出电路

输入 A	这是 TTL 电平的史密特触发器输入，内部有高阻（几十 $K\Omega$ ~几百 $K\Omega$ ）上拉至 VDD。可与 CMOS, TTL 连接。不用时，请开路或上拉至+5V。
输出 A	CMOS 电平的输出。因为是 4mA 驱动缓冲器（若高电平输出电流 $I_{OH} = -4mA$, $V_{OH}=2.4V_{min}$, 若低电平输出电流 $I_{OL}=4mA$, $V_{OL}=0.4V_{max}$ ），所以它能驱动多达 10 个 LSTTL。
输出 B	开漏输出，4mA 驱动缓冲器（若低电平输出电流 $I_{OL}=4mA$, $V_{OL}=0.4V_{max}$ ）使用时，用高电阻上拉至+5V。
双方向 A	输入侧是 TTL 电平的史密特触发器输入，在 IC 内部没有用高电阻上拉，有高的输入电阻。为避免信号线处于高阻态，在整个系统中用高电阻上拉数据总线。不用 D15~D8 时，请用高电阻（约 $100K\Omega$ ）上拉至+5V。因为是双方向，所以最好不要直接上拉，用高电阻上拉安全些。 输出侧是 CMOS 电平输出，8mA 驱动缓冲器（若高电平输出电流 $I_{OH} = -8mA$, $V_{OH}=2.4V_{min}$, 若低电平输出电流 $I_{OL}=8mA$, $V_{OL}=0.4V_{max}$ ）。

电路设计上的考虑

(1) 去耦电容器

在此 IC 的 VDD 和 GND 之间请放 1~2 个高频特性好的 $0.1\mu F$ 左右的去耦电容器。

(2) 引脚电感引起的振铃噪音

由于输出引脚端带有的电感和连接在输出的负载电容共振，使输出信号在上升和下降时，可能出现振铃噪音。振铃噪音太大使电路无法正常工作时，要连接 $10\sim 100PF$ 左右的负载电容，以降低振铃噪音。

(3) 传送通路上的反射

以输出 A、B 及双方向 A 形式输出时，假设负载电容为 $20\sim 50PF$ ，信号的上升/下降时间在 $3\sim 4ns$ ，配线长度 $60cm$ 左右开始反射的影响很明显。为减少反射，布线的长度越短越好。

4 读/写寄存器

本章详细叙述 CPU 为控制每个轴而存取的读/写寄存器。有关位模式插补寄存器 (BP1P/M, BP2P/M, BP3P/M) 请参考 2.4.3 节的位模式插补。

4.1 16 位数据总线的寄存器地址

如下表所示, 使用 16 位数据总线时, 存取 16 位读/写的地址有 8 个。

■ 16 位数据总线的写入寄存器

地址 A2 A1 A0	寄存器符号	寄存器名称	内 容
0 0 0	WR0	命令寄存器	轴指定, 命令编码的设定
0 0 1	XWR1 YWR1 ZWR1 UWR1	X 轴方式寄存器 1 Y 轴方式寄存器 1 Z 轴方式寄存器 1 U 轴方式寄存器 1	设定各轴的外部减速停止信号的逻辑电平和有效/无效。 设定各轴中断的允许/禁止。
0 1 0	XWR2 YWR2 ZWR2 UWR2	X 轴方式寄存器 2 Y 轴方式寄存器 2 Z 轴方式寄存器 2 U 轴方式寄存器 2	设定各轴限制信号的模式。设定驱动脉冲的模式。设定编码器输入信号的模式。设定用于伺服马达信号的逻辑电平和有效/无效。
	BP1P	BP1P 寄存器	设定正方向位模式插补第 1 轴的位数据。
0 1 1	XWR3 YWR3 ZWR3 UWR3	X 轴方式寄存器 3 Y 轴方式寄存器 3 Z 轴方式寄存器 3 U 轴方式寄存器 3	设定各轴的手动减速, 单独减速度, S 曲线加减速模式。 设定外部操作模式。 设定通用输出 OUT7~4。
	BP1M	BP1M 寄存器	设定负方向位模式插补第 1 轴的位数据。
1 0 0	WR4	输出寄存器	设定通用输出 OUT3~0。
	BP2P	BP2P 寄存器	设定正方向位模式插补第 2 轴的位数据。
1 0 1	WR5	插补方式寄存器	轴指定。设定固定线速度模式、步进模式和中断。
	BP2M	BP2M 寄存器	设定负方向位模式插补第 2 轴的位数据。
1 1 0	WR6 BP3P	写数据寄存器 1 BP3P 寄存器	设定写入数据的低 16 位 (D15~D0)。 设定正方向位模式插补第 3 轴的位数据。
	WR7 BP3M	写数据寄存器 2 BP3M 寄存器	设定写入数据的高 16 位 (D31~D16)。 设定负方向设定位模式插补第 3 轴设定的位数据。

- 如上表所示, 每个轴都有 WR1, WR2, WR3 (模式寄存器 1, 2, 3), 用统一地址写这些寄存器。由事先写入轴指定命令来决定写入哪一个轴的模式寄存器。或者, 事先写入轴指定的 NOP 命令, 选择要写入的轴。
- 复位后位模式插补用的位数据寄存器 BP1~3P、BP1~3M 不能马上写, 发 BP 寄存器允许命令(36h)后, 才能写这些寄存器。发 BP 寄存器写入允许命令后, 不能写 nWR2~3, 所以在位模式插补中写完位数据后, 需要发禁止写 BP 寄存器的命令 (37h)。
- WR6 寄存器和 BP3P 寄存器, WR7 寄存器和 BP3M 寄存器共享同一个寄存器。使用时, 请注意这点。
- 复位时, nWR1, nWR2, nWR3, WR4, WR5 寄存器所有的位都被清除为 0。其它寄存器不确定。

■ 16 位数据总线的读寄存器

地址 A2 A1 A0	寄存器符号	寄存器名称	内 容
0 0 0	RR0	主状态寄存器	表示每一个轴的驱动和出错状态。 表示插补驱动、连续插补下一个数据允许、圆弧插补的象限和 BP 插补堆栈计数器的状态。
0 0 1	XRR1 YRR1 ZRR1 URR1	X 轴状态寄存器 1 Y 轴状态寄存器 1 Z 轴状态寄存器 1 U 轴状态寄存器 1	位置：表示 COMP 比较器的比较，加速度和加速度变化率的状态。 表示结束状态。
0 1 0	XRR2 YRR2 ZRR2 URR2	X 轴状态寄存器 2 Y 轴状态寄存器 2 Z 轴状态寄存器 2 U 轴状态寄存器 2	表示出错原因。
0 1 1	XRR3 YRR3 ZRR3 URR3	X 轴状态寄存器 3 Y 轴状态寄存器 3 Z 轴状态寄存器 3 U 轴状态寄存器 3	表示发生中断的原因。
1 0 0	RR4	输入寄存器 1	表示 X 轴、Y 轴输入信号的状态。
1 0 1	RR5	输入寄存器 2	表示 Z 轴、U 轴输入信号的状态。
1 1 0	RR6	读数据寄存器 1	表示读出数据的低 16 位 (D15~D0)。
1 1 1	RR7	读数据寄存器 2	表示读出数据的高 16 位 (D31~D16)。

- 每一个轴都有 RR1、RR2、RR3（每一个轴状态寄存器 1，2，3）用同一个地址读出这些寄存器，由事先写入轴指定的命令来决定写入哪一个轴的状态寄存器，或者事先写入轴指定的 NOP 命令，选择要读出的轴。

4.2 8 位数据总线的寄存器地址

用 8 位数据总线存取时，16 位寄存器分为高位字节和低位字节。

下表所示，****L 是 16 位寄存器****的低位字节（D7~D0），****H 是 16 位寄存器****的高位字节（D15~D8）。只有命令寄存器（WR0L，WR0H）一定要先写高位字节（WR0H）后写低位字节（WR0L）。

■ 8 位数据总线的写寄存器

地址 A3 A2 A1 A0	写寄存器
0 0 0 0	WR0L
0 0 0 1	WR0H
0 0 1 0	XWR1L,YWR1L,ZWR1L,UWR1L
0 0 1 1	XWR1H,YWR1H,ZWR1H,UWR1H
0 1 0 0	XWR2L,YWR2L,ZWR2L,UWR2L,BP1PL
0 1 0 1	XWR2H,YWR2H,ZWR2H,UWR2H,BP1PH
0 1 1 0	XWR3L,YWR3L,ZWR3L,UWR3L,BP1ML
0 1 1 1	XWR3H,YWR3H,ZWR3H,UWR3H,BP1MH
1 0 0 0	WR4L,BP2PL
1 0 0 1	WR4H,BP2PH
1 0 1 0	WR5L,BP2ML
1 0 1 1	WR5H,BP2MH
1 1 0 0	WR6L,BP3PL
1 1 0 1	WR6H,BP3PH
1 1 1 0	WR7L,BP3ML
1 1 1 1	WR7H,BP3MH

■ 8 位数据总线的读寄存器

地址 A3 A2 A1 A0	读寄存器
0 0 0 0	RR0L
0 0 0 1	RR0H
0 0 1 0	XRR1L,YRR1L,ZRR1L,URR1L
0 0 1 1	XRR1H,YRR1H,ZRR1H,URR1H
0 1 0 0	XRR2L,YRR2L,ZRR2L,URR2L
0 1 0 1	XRR2H,YRR2H,ZRR2H,URR2H
0 1 1 0	XRR3L,YRR3L,ZRR3L,URR3L
0 1 1 1	XRR3H,YRR3H,ZRR3H,URR3H
1 0 0 0	RR4L
1 0 0 1	RR4H
1 0 1 0	RR5L
1 0 1 1	RR5H
1 1 0 0	RR6L
1 1 0 1	RR6H
1 1 1 0	RR7L
1 1 1 1	RR7H

4.3 WR0 命令寄存器

此寄存器用于给 IC 内的各轴指定轴和写入命令。寄存器包括指定轴的位，设定命令编码的位和复位命令位。在此寄存器写入轴指定和命令编码后，该命令马上就执行。事先一定要在 WR6, 7 寄存器里写好数据，然后写命令数据，譬如命令可能执行是驱动速度的设定等。命令写入此命令寄存器后，由数据读命令从内部电路将数据设定在 RR6, 7 寄存器。

8 位数据总线时，先写高位字节（H）后写低位字节（L）。写入低位字节后，针对先指定轴的命令马上执行。

所有命令编码的命令处理时间需要最大 250nSEC（CLK=16MHz 时）在这期间请不要写入下一个命令，输出信号 BUSYN 在这个命令处理期间保持低电平。



D5~0 设定命令编码，有关命令编码请参考 5 章以后的各命令说明。

D11~8 指定执行命令的轴。那个轴的位为 1，就被指定。轴指定不限于 1 个轴，可以 1 个轴以上。可以同时对各轴发同样的命令，并且写入同样的参数。但是读数据命令只对 1 个指定轴。命令涉及到插补，轴指定的位都要为 0。

D15 RESET 用命令复位此 IC 的位。把这个位设定为 1，其它位设定为 0，写入命令，此 IC 将被复位。写入命令后，BUSYN 信号处于低电平最大 875nSEC（CLK=16MHz 时），在这期间不能对此 IC 的寄存器存取。

8 位数据总线时，写入 WR0H (=80h) 后复位。

写入一般命令时，RESET 位一定要为 0。

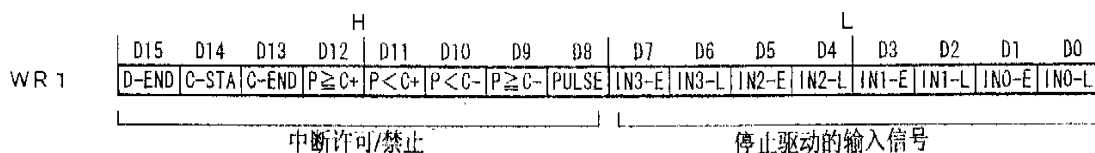
其它位一定要为 0，若设定为 1 的话，IC 内部电路的测试命令将被启动，然后将发生意外的动作。

4.4 WR1 模式寄存器 1

4 个轴都有模式寄存器 1。根据事先写入轴指定的命令决定写哪个轴的模式寄存器，或者事先写入轴指定的 NOP 命令，选择要写入的轴。

模式寄存器 1 有 2 种设定的位。一种是在驱动中设定减速停止/立即停止的输入信号 IN3~IN0 的有效/无效和逻辑电平的位，另一种是对每一个中断源设定允许/禁止的位。

把 IN3~IN0 设定为有效，在定量驱动或连续驱动开始驱动后，指定的 IN 信号达到设定的逻辑电平，驱动将减速停止或立即停止。使用加减速驱动的话，执行减速停止。使用定速驱动的话，执行立即停止。



D7,5,3,1 INm-E 设定驱动停止输入信号 INm 的有效/无效的位。0: 无效 1: 有效

D6,4,2,0 INm-L 设定输入信号 INm 的有效逻辑电平的位。0: 低电平停止 1: 高电平停止

以下是中断允许/禁止位，设定为 1 为中断允许，设定为 0 为中断禁止。

- D8 PULSE 每一个驱动脉冲上升沿，中断将发生（驱动脉冲用正逻辑）
 - D9 $P \geq C-$ 逻辑/实位计数器的数值比 COMP-寄存器的数值变大时，中断将发生。
 - D10 $P < C-$ 逻辑/实位计数器的数值比 COMP-寄存器的数值变小时，中断将发生。
 - D11 $P < C+$ 逻辑/实位计数器的数值比 COMP+寄存器的数值变小时，中断将发生。
 - D12 $P \geq C+$ 逻辑/实位计数器的数值比 COMP+寄存器的数值变大时，中断将发生。
 - D13 C-END 加减速驱动时，在定速域结束脉冲输出后中断发生。
 - D14 C-STA 加减速驱动时，在定速域开始脉冲输出后中断发生。
 - D15 D-END 驱动结束时，中断将发生。
- 复位时，D15~D0 为 0。

4.5 WR2 模式寄存器 2

4 个轴都有模式寄存器 2。根据写入轴指定的命令来决定写入哪个轴的模式寄存器，或者预先写入轴指定的 NOP 命令，选择要写的轴。

模式寄存器 2 执行限制输入信号的模式设定、驱动脉冲的模式设定、编码器输入信号的模式设定及伺服马达用信号的逻辑电平和有效/无效的设定。

	H										L					
WR 2	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
	INP-E	INP-L	ALM-E	ALM-L	PINDI	PINDO	PINMD	DIR-L	PLS-L	PLSMD	CMPSL	HLMT-	HLMT+	LMTMD	SLMT-	SLMT+

- D0 SLMT+ 把 COMP+寄存器作为软件限制设定为有效/无效，设定 1 为有效；设定 0 为无效。有效后，在正方向的驱动中，逻辑/实位计数器的值大于 COMP+寄存器的值就减速停止。另外，当 RR2 寄存器的 D0（SLMT+）位为 1，即使在此模式下写入正方向的驱动命令，也不会执行。
- D1 SLMT- 把 COMP-寄存器作为软件限制设定为有效/无效，设定 1 为有效；设定 0 为无效。有效后，在负方向的驱动中，逻辑/实位计数器的值小于 COMP-寄存器的值，就减速停止。另外，当 RR2 寄存器的 D1（SLMT-）位为 1，即使在此模式下写入负方向的驱动命令，也不会执行。
- D2 LMTMD 设定当硬件限制（nLMTP，nLMTM 输入信号）为有效时就停止的驱动停止方式。设定为 0，立即停止；设定为 1，减速停止。
- D3 HLMT+ 设定正方向限制输入信号（nLMTP）的逻辑电平，0：低电平有效，1：高电平有效。
- D4 HLMT- 设定负方向限制输入信号（nLMTP）的逻辑电平，0：低电平有效，1：高电平有效。
- D5 CMPSL 把 COMP+/-寄存器的比较对象设定为逻辑位置计数器或实际位置计数器。0：逻辑位置计数器；1：实位计数器。
- D6 PLSMD 设定驱动脉冲的输出方式。0：独立 2 脉冲方式，1：1 脉冲方式。设定为独立 2 脉冲方式后正方向脉冲通过输出信号 nPP/PLS 输出，负方向脉冲通过输出信号 nPM/DIR 输出。设定为 1 脉冲方式后，正负 2 个方向的驱动脉冲通过输出信号 nPP/PLS 输出，脉冲方向信号通过输出信号 nPM/DIR 输出。

【注意】 1 脉冲方式时，有关输出脉冲信号（nPLS）和方向信号（nDIR）的时序，请参考 13.2，13.3 节。

D7 PLS-L 设定驱动脉冲的逻辑电平。0: 正逻辑脉冲 1: 负逻辑脉冲



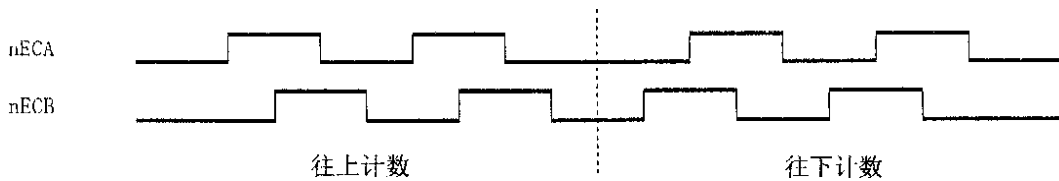
D8 DIR-L 设定驱动脉冲的方向输出信号逻辑电平。
根据这个位数值，如下表所示 nPM/DIR 输出信号的电平。

DIR-L	正方向脉冲输出时	负方向脉冲输出时
0	Low	Hi
1	Hi	Low

D9 PINMD 把编码器输入信号 (nECA/PPIN, nECB/PMIN) 设定为 2 相脉冲输入或上/下脉冲输入。编码器输入信号对实位计数器向上/下计数。

0: 2 相脉冲输入 1: 上/下脉冲输入

把该位设定为 2 相脉冲输入后，按正逻辑当 A 相超前时向上计数，当 B 相超前就向下计数。由 2 个信号的上升沿和下降沿进行向上/下计数。



把该位设定为上/下脉冲输入的模式后，nECA/PPIN 就为向上计数输入，nECB/PMIN 为向下计数输入，由正脉冲的上升沿计数。

D11,10 PIND1,0 设定编码器 2 相脉冲输入的分周比

D11	D10	2 相脉冲输入的分周比
0	0	1/1
0	1	1/2
1	0	1/4
1	1	无效

上/下脉冲输入不被分周

D12 ALM-L 设定 nALARM 输入信号的逻辑电平，0: 低电平有效 1: 高电平有效。

D13 ALM-E 设定伺服马达警报用输入信号 nALARM 的有效/无效，0: 无效 1: 有效
定为有效，一直监视 nALARM 输入信号，激活状态时，RR2 寄存器的 D14 (ALARM) 位为 1，在驱动中变为有效电平后，驱动就立即停止。

D14 INP-L 设定 nINPOS 输入信号的逻辑电平。0: 低电平有效 1: 高电平有效。

D15 INP-E 设定伺服马达定位完毕用输入信号 nINPOS 的有效/无效，0: 无效 1: 有效。
设定为有效，当驱动结束后或 nINPOS 信号起作用时，RR0 (主状态) 寄存器的 nDRV 位返回 0。

复位时，D15~D0 都为 0。

4.6 WR3 模式寄存器 3

4 个轴都有模式寄存器 3。根据事先写入轴指定的命令来决定写入哪个轴的模式寄存器，或者事先写入轴指定的 NOP 命令，选择要写入的轴。

模式寄存器 3 执行手动减速、独立减速度、S 曲线减速模式、外部操作模式的设定及通用输出 OUT7~4 的设定。

WR 3	H								L							
	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
	0	0	0	0	OUT7	OUT6	OUT5	OUT4	OUTSL	0	0	EXOP1	EXOP0	SACC	DSNDE	MANLD

- D0 MANLD** 把加/减速定量驱动的减速设定为自动减速或手动减速。
0: 自动减速 **1:** 手动减速
 设定为手动减速模式时，事先要设定好手动减速点。
- D1 DSNDE** 把加/减速驱动减速时的减速度设定为加速度的数值或独立的减速度数值。
0: 使用加速度的数值 **1:** 使用减速度的数值。
 设定为 0，加/减速驱动的加速度和减速度使用同一个加速度数值。自动减速时，一定要设定为 0。设定为 1 的话，加速时，使用加速度数值，减速时，使用减速度。设定为 1 的话，只好使用手动减速。
- D2 SACC** 设定直线加/减速和 S 曲线加/减速。**0:** 直线加/减速 **1:** S 曲线加/减速。S 曲线加/减速时，一定要设定好加/减速度的变化率。
- D4,3 EXOP1,0** 根据外部输入信号（nEXPP，nEXPM）设定驱动操作。

D4	D3	
0	0	根据外部输入信号驱动操作的无效
0	1	连续脉冲驱动
1	0	定量脉冲驱动
1	1	根据外部输入信号驱动操作的无效

在连续驱动模式时，在 nEXPP 信号保持低电平期间，连续输出正方向的驱动脉冲。在 nEXPM 信号保持低电平期间，连续输出负方向的驱动脉冲。
 在定量驱动模式时，把 nEXPP 信号从高电平降到低电平后，由其下降沿启动正方向的定量驱动，nEXPM 信号也一样，启动负方向的定量驱动。

- D7 OUTSL** 选择输出信号 nOUT7~4 为通用输出还是为输出驱动状态。
0: 为通用输出。D11~D8 的内容输出至 nOUT7~4 端子。
1: 把驱动状态输出至下图所示的 nOUT7~4。

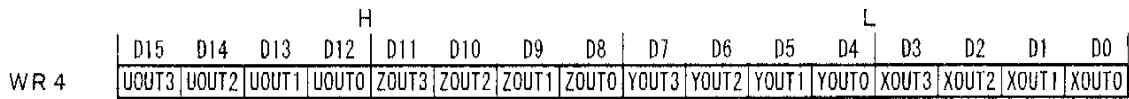
信号名	输出内容
nOUT4/CMPP	当逻辑/实际寄存器的值大于 COMP+寄存器时，输出高电平；当小于时输出低电平。
nOUT5/CMPM	当逻辑/实际寄存器的值大于 COMP-寄存器时，输出低电平；当小于时输出高电平。
nOUT6/ASND	驱动命令正在执行时，并处于加速状态，输出高电平。
nOUT7/DSND	驱动命令正在执行时，并处于减速状态，输出高电平。

- D11~8 OUTm** 设定把输出信号 nOUT7~4 作为通用输出使用时的数值。
0: 低电平输出 **1:** 高电平输出

复位时，D15~D0 都为 0，请一直在 D15~12、6，5 位上设定为 0。

4.7 WR4 输出寄存器

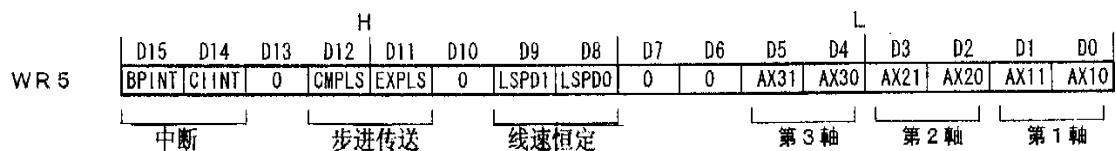
此寄存器用于设定通用输出 nOUT3~0。4 个轴的输出信号共享 1 个 16 位寄存器。寄存器可以作为真正的 16 位通用输出寄存器使用。任何位为 0 时，输出低电平，为 1 时，输出高电平。



复位时，D15~D0 都为 0，nOUT3~0 输出信号都为低电平。

4.8 WR5 插补模式寄存器

设定运行插补驱动的轴指定、固定线速度模式插补、步进插补模式和插补时中断。



D1,0 AX11,10 指定运行插补驱动的第 1 轴（主轴），下图所示轴编码。

轴	编码（2 进制）	
X	0	0
Y	0	1
Z	1	0
U	1	1

第 1 轴：X、第 2 轴：Y、第 3 轴：Z 的例子

D5	D4	D3	D2	D1	D0
1	0	0	1	0	0

指定为第 1 轴（主轴）的轴产生启动插补运算的基本脉冲，所以要设定在定速/加/减速驱动中所需要的速度参数。

D3,2 AX21,20 用上表所示的编码指定运行插补驱动的第 2 轴。

D5,4 AX31,30 用上表所示的编码指定运行 3 轴插补驱动的第 3 轴。

2 轴插补驱动时，设定任选的数值。

D9,8 LSPD1,0 设定插补驱动的固定线速度模式。

D9	D8	操作模式
0	0	线速恒定无效
0	1	2 轴线速恒定
1	0	（不能设定）
1	1	3 轴线速恒定

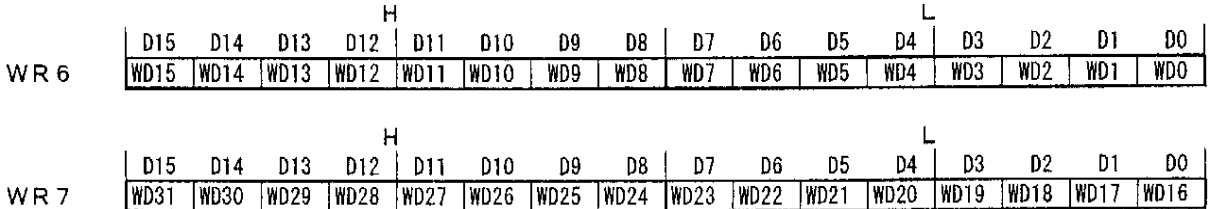
2 轴线速恒定模式时，用主轴 R 的 1.414 倍的数值设定第 2 轴的范围（R）。

3 轴线速恒定模式时，用主轴 R 的 1.414 倍的数值设定第 2 轴的范围（R），用主轴 R 的 1.732 倍的数值设定第 3 轴的范围（R）

- D11 EXPLS 设定为 1，为根据外部信号（EXPLSN）步进插补驱动的模式。
- D12 CMPLS 设定为 1，为根据命令执行步进插补驱动的模式。
- D14 CIINT 设定连续插补时中断的允许/禁止。 0: 禁止 1: 允许
- D15 BPINT 设定定位模式插补时中断的允许/禁止。 0: 禁止 1: 允许
- 复位时，D15~D0 都为 0。

4.9 WR6, 7 写数据寄存器 1, 2

设定写数据命令所要数据的寄存器。WR6 寄存器写入数据低 16 位（WD15~WD0），WR7 寄存器写入数据高 16 位（WD31~WD16）。



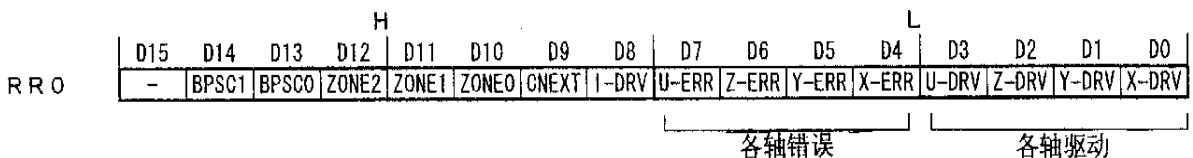
写数据命令首先在此数据寄存器写入根据每个命令指定的数据长度的数据。

写数据寄存器 WR6,7（8 位数据总线时，WR6L, WR6H, WR7L, WR7H）无论先后写入。此后，命令码写入命令寄存器后，写数据寄存器的内容就存入各个寄存器。写入数据的数值都是二进制。此外，负数以 2 进制补码设定。每一个命令的数据一定要用指定的数据长度设定。比如，圆弧插补的终点可用的算术数据范围为 -8,388,608~+8,388,607 的带符号 24 位，其数据长度是 4 字节，所以请用带符号的 32 位设定。

复位时，WR6, WR7 寄存器的内容不固定。

4.10 RR0 主状态寄存器

表示每一个轴的驱动、出错状态。此外，表示插补驱动、连续插补下一个数据允许、圆弧插补的象限、BP 插补的堆栈计数器。



- D3~0 n-DRV 表示每一个轴的驱动状态，该位为 1 时，表示此轴正在输出驱动脉冲。为 0 时，表示此轴已经结束驱动。
 伺服马达定位完毕所用的输入信号 nINPOS 有效时，输出驱动脉冲后，nINPOS 信号激活就返回 0。
- D7~4 n-ERR 表示每一个轴的所有出错状态。在每一个轴 RR2 寄存器的出错位（D5~D0）及 RR1 寄存器的错误结束位（D15~D12）中的任何位为 1，该位就为 1。
- D8 I-DRV 表示插补驱动状态。该位为 1 时，表示正在输出插补驱动脉冲。
- D9 CNEXT 表示可以写入连续插补的下一个数据。在连续插补驱动中，该位为 1 后，可以写入下一个节点的参数及插补命令。
- D12~10 ZONEm 在圆弧插补驱动中表示正在驱动所在的象限。

- D11~8 IN3~0 根据外部减速停止信号 (nIN3~0) 驱动停止时, 为 1。
- D12 LMT+ 根据正方向限制信号 (nLMPT) 驱动停止时, 为 1。
- D13 LMT- 根据负方向限制信号 (nLMTM) 驱动停止时, 为 1。
- D14 ALARM 根据伺服马达警报信号 (nALARM) 驱动停止时, 为 1。
- D5 EMG 根据紧急停止信号 (EMGN) 驱动停止时, 为 1。

■ 关于驱动结束状态位

驱动结束状态位保持驱动结束原因的信息, 下述的原因会结束定量驱动或连续驱动将结束。

- ① 在定量驱动中, 输出所有的输出脉冲时
- ② 减速停止或立即停止命令被写入时
- ③ 设定成有效的软件限制激活时
- ④ 在定量/连续驱动中, 设定成有效的减速停止外部信号 (nIN3, 2, 1, 0) 激活时
- ⑤ 限制输入信号 (nLMTP, nLMTM) 激活时
- ⑥ 设定成有效的 nLALRM 信号激活时
- ⑦ EMGN 信号为低电平时

- ①②因素 ——高位 CPU 可以管理它们
- ③因素 ——驱动结束后状态都不变, 可以在 RR2 寄存器确认它
- ④~⑦因素 ——直至驱动结束, 不一定保持激活状态

驱动结束状态位在结束驱动的④~⑦因素的位为 1, 然后信号变为不激活也保持位信息。

在驱动结束状态位里出错因素的 D15~D12 的位为 1 后, RR0 主状态寄存器的 n-ERR 位为 1。由于下一个驱动命令的写入, 驱动结束状态位自动清除, 结束状态清除命令 (25h) 也可以清除。

4.12 RR2 状态寄存器 2

4 个轴都有状态寄存器 2。根据事先写入的命令决定读出哪个轴的状态寄存器或者事先写入轴指定的 NOP 命令, 选择要读出的轴。状态寄存器 2 表示出错信息, 在每个位为 1 时, 表示此位指示的错误发生, RR2 寄存器 D5~D0 的任何位为 1 后, RR0 主状态寄存器的 n-ERR 位为 1。

RR2	H								L							
	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
	-	-	-	-	-	-	-	-	-	-	EMG	ALARM	HLMT-	HLMT+	SLMT-	SLMT+

- D0 SLMT+ 设定 COMP+寄存器作为有效软件限制后, 在正方向驱动中, 逻辑/实位计数器大于 COMP+寄存器值。
- D1 SLMT- 设定 COMP-寄存器作为有效软件限制后, 在负方向驱动中, 逻辑/实位计数器小于 COMP-寄存器值。
- D2 HLMT+ 正方向限制信号 (nLMTP) 处于有效电平
- D3 HLMT- 负方向限制信号 (nLMTP) 处于有效电平
- D4 ALARM 设定成有效的伺服马达报警信号 (nALARM) 处于有效电平
- D5 EMG 紧急停止信号 (EMGN) 处于低电平

在驱动中顺方向的硬件/软件限制启动后, 驱动减速停止或立即停止。停止后的同方向驱动命令不被执行, 在逆方向驱动时 SLMT+/-位达到那些条件, 也不变为 1。

4.13 RR3 状态寄存器 3

4 个轴都有状态寄存器 3。根据事先写入的命令决定读出哪个轴的状态寄存器或者事先写入轴指定的 NOP 命令，选择要读出的轴。状态寄存器 3 表示中断源。中断发生后，引发中断的位就为 1。为了产生中断，要在 WR1 寄存器中设定中断允许。

	H								L							
RR 3	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
	-	-	-	-	-	-	-	-	D-END	C-STA	C-END	P \geq C+	P<C+	P<C-	P \geq C-	PULSE

- D0 PULSE 驱动脉冲上升时（驱动脉冲以正逻辑方式）
- D1 P \geq C- 逻辑/实位计数器的值大于 COMP-寄存器的值时
- D2 P<C- 逻辑/实位计数器的值小于 COMP+寄存器的值时
- D3 P<C+ 逻辑/实位计数器的值小于 COMP+寄存器的值时
- D4 P \geq C+ 逻辑/实位计数器的值大于 COMP+寄存器的值时
- D5 C-END 加/减速驱动时，在定速域结束脉冲输出时
- D6 C-STA 加/减速驱动时，在定速域开始脉冲输出时
- D7 D-END 驱动结束

某个中断源引发中断后，此寄存器的位就为 1，中断输出信号（INTN）为低电平。CPU 读出发生中断的这个 RR3 寄存器后，RR3 寄存器的位被清除为 0，中断输出信号返回无效电平。8 位数据总线时，RR3L 寄存器的读出后被清除。

4.14 RR4,5 输入寄存器 1, 2

输入寄存器 1, 2 表示每一个轴的输入信号状态，输入信号在低电平时，表示 0，高电平时，表示 1。不使用这些输入信号的功能时，可以作为通用输入信号使用。

	H								L							
RR 4	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
	Y-ALM	Y-INP	Y-EX-	Y-EX+	Y-IN3	Y-IN2	Y-IN1	Y-INO	X-ALM	X-INP	X-EX-	X-EX+	X-IN3	X-IN2	X-IN1	X-INO

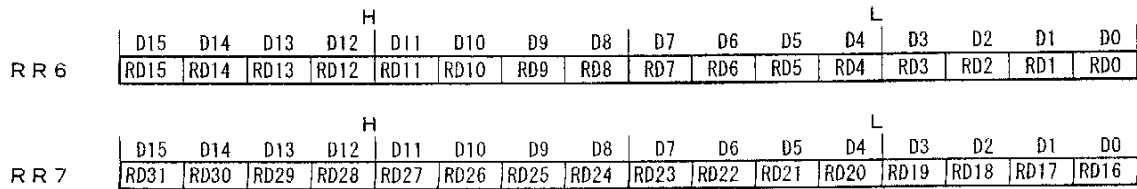
	H								L							
RR 5	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
	U-ALM	U-INP	U-EX-	U-EX+	U-IN3	U-IN2	U-IN1	U-INO	Z-ALM	Z-INP	Z-EX-	Z-EX+	Z-IN3	Z-IN2	Z-IN1	Z-INO

位名	输入信号	位名	输入信号
n-IN0	nIN0	n-EX+	nEXPP
n-IN1	nIN1	n-EX-	nEXPM
n-IN2	nIN2	n-INP	nINPOS
n-IN3	nIN3	n-ALM	nALARM

4.15 RR6,7 读数据寄存器 1, 2

由数据读出命令将内部寄存器的数据设定在这些寄存器。低 16 位（RD15~RD0）在 RR6 寄存器；高 16 位（RD31~RD16）在 RR7 寄存器。

数据都是二进制，负数是2的补码。



5 命令列表

■ 数据写命令

编码	命令	参数符号	数据范围	数据长度
00h	范围设定	R	8,000,000 (倍率: 1) ~16,000 (倍率: 500)	4 字节
01	加/减速度的变化率设定	K	1~65,535	2
02	加速度设定	A	1~8,000	2
03	减速度设定	D	1~8,000	2
04	初始速度	SV	1~8,000	2
05	驱动速度设定	V	1~8,000	2
06	输出脉冲数/插补终点设定	P	输出脉冲数: 0~268,435,455 插补终点: -8,388,608~+8,388,607	4
07	手动减速点设定	DP	0~268,435,455	4
08	圆弧中心点设定	C	-8,388,608~+8,388,607	4
09	逻辑位置计数器设定	LP	-2,147,483,648~+2,147,483,647	4
0A	实位计数器设定	EP	-2,147,483,648~+2,147,483,647	4
0B	COMP+寄存器设定	CP	-2,147,483,648~+2,147,483,647	4
0C	COMP-寄存器设定	CM	-2,147,483,648~+2,147,483,647	4
0D	加速计数器偏移设定	AO	0~65,535	4
0F	NOP (轴切换用)			2

【注意】 数据范围有些达不到上表的数据长度的参数，但是写入数据时一定要写入指定的数据长度。

[参数计算公式]

$$\text{倍率} = \frac{8,000,000}{R}$$

$$\text{Jerk (PPS/SEC}^2) = \frac{62.5 \times 10^6}{K} \times \underbrace{\frac{8,000,000}{R}}_{\text{倍率}}$$

$$\text{加速度 (PPS/SEC)} = A \times 125 \times \underbrace{\frac{8,000,000}{R}}_{\text{倍率}}$$

$$\text{减速度 (PPS/SEC)} = D \times 125 \times \underbrace{\frac{8,000,000}{R}}_{\text{倍率}}$$

$$\text{驱动速度 (PPS)} = V \times \underbrace{\frac{8,000,000}{R}}_{\text{倍率}}$$

$$\text{初始速度 (PPS)} = SV \times \underbrace{\frac{8,000,000}{R}}_{\text{倍率}}$$

■ 数据读出命令

编码	命令	参数符号	数据范围	数据长
10h	逻辑位置计数器读出	LP	-2,147,483,648~+2,147,483,647	4 字节
11	实位计数器读出	EP	-2,147,483,648~+2,147,483,647	4
12	当前驱动速度读出	CV	1~8,000	2
13	当前加减速度读出	CA	1~8,000	2

■ 驱动命令

编码	命令
20h	正方向定量驱动
21	负方向定量驱动
22	正方向连续驱动
23	负方向连续驱动
24	暂停驱动
25	解除暂停驱动状态/清除结束状态
26	驱动减速停止
27	驱动立即停止

■ 插补命令

编码	命令
30h	2 轴直线插补驱动
31	3 轴直线插补驱动
32	CW 圆弧插补驱动
33	CCW 圆弧插补驱动
34	2 轴位模式插补驱动
35	3 轴位模式插补驱动
36	BP 寄存器写入允许 (注 1)
37	BP 禁止寄存器写入不允许
38	BP 数据堆栈
39	BP 数据清除
3A	单步插补
3B	减速有效
3C	减速无效
3D	插补中断清除

(注1) BP: “位模式”的简称

【注意】 请不要在命令寄存器写入除上表之外的命令码。IC 内部电路的试验命令可能被启动，也许会发生意外的动作。

6 数据写入命令

数据写入命令是带有写入数据的命令，用于设定驱动参数，诸如加速度、驱动速度、输出脉冲数等。如果指定多个轴，可以同时把同样的数据设定在指定的所有轴。

指定的数据长度是 2 字节时，数据写入命令在 WR6 寄存器设定数据。指定的数据长度是 4 字节时，数据写入命令在 WR6, 7 寄存器设定数据。此后，在 WR0 寄存器写入轴指定和命令编码就执行此命令。

设定在 WR6, 7 写数据寄存器的数值都是 2 进制，负数为 2 的补码。每一个数据一定要是设定数据范围内的数值，设定范围以外的数值，不能进行正确的驱动。

【注意事项】

- 数据写入命令的命令处理时间需要最大 250nSEC (CLK=16MHz 时)，在命令处理中请不要写入下一个数据命令。
- 复位时，除加速计数器偏置 (AO) 外，其它所有的运行参数都不固定。在驱动前，驱动所需的参数要设定适当的数值。

6.1 范围设定

命令编码	命令	符号	数据范围	数据长 (字节)
00h	范围设定	R	8,000,000 (倍率=1)~16,000 (倍率=500)	4

范围是决定速度、加/减速度、加/减速度的变化率的倍率参数，假设把范围数值作为 R，倍率是下述的算式。

$$\text{倍率} = \frac{8,000,000}{R}$$

驱动速度、初始速度、加/减速度等参数的设定范围在 1~8,000。若设定为 8,000 以上的数值的话，要提高倍率。提高倍率后可以高速驱动，但是速度分辨率变粗。请在使用的速度范围内设定最小的数值。比如，如果需要 40KPPS 的速度，在速度范围内 1~8,000 倍率中最好是 5，即设定 R 为 1,600,000。在驱动中请不要变更范围 (R)，否则速度会变乱。

6.2 加/减速度的变化率设定

命令编码	命令	符号	数据范围	数据长 (字节)
01h	加/减速度的变化率设定	K	1~65,535	2

加/减速度的变化率设定值是决定 S 曲线加/减速的加速度及减速度在单位时间内增加/减少率的参数。把加/减速度的变化率的设定值作为 K 的话，加/减速度的变化率由下述算式表示。

$$\text{加减速度的变化率 (PPS / SEC}^2\text{)} = \frac{62.5 \times 10^6}{K} \times \underbrace{\frac{8,000,000}{R}}_{\text{倍率}}$$

加/减速度的变化率设定值 (K) 的设定范围是 1~65,535，所以加/减速度的变化率范围如下所示。

$$\begin{aligned} \text{倍率}=1 & \quad 954\text{PPS/SEC}^2 \sim 62.5 \times 10^6\text{PPS/SEC}^2 \\ \text{倍率}=500 & \quad 477 \times 10^3\text{PPS/SEC}^2 \sim 31.25 \times 10^9\text{PPS/SEC}^2 \end{aligned}$$

注 1: Jerk 的意思是指，加速度/减速度在单位时间内增加/减少率；Jerk 不仅是加速度的增加率，也是加速度的减少率，Jerk 还包含减速度的增加率和减速度的减少率。

6.3 加速度设定

命令编码	命令	参数符号	数据范围	数据长 (字节)
02h	加速度设定	A	1~8,000	2

复位后的常用模式中，它是作为直线加减速驱动中直线加速及减速的参数。在 S 曲线加/减速驱动中，加速度及减速度线性地从 0 增加至加速度的设定值。请参考图 2.9。加速度设定值为 A，加速度是下述算式。

$$\text{加速度 (PPS/SEC)} = A \times 125 \times \frac{8,000,000}{R} \times \text{倍率}$$

加速度设定值 (A) 的设定范围是 1~8,000，所以实际的加速度范围如下：

$$\begin{aligned} \text{倍率}=1 & \quad 125\text{PPS/SEC} \sim 1 \times 10^6\text{PPS/SEC} \\ \text{倍率}=500 & \quad 62.5 \times 10^3\text{PPS/SEC} \sim 500 \times 10^6\text{PPS/SEC} \end{aligned}$$

6.4 减速度设定

命令编码	命令	符号	数据范围	数据长 (字节)
03h	减速度设定	D	1~8,000	2

在加速度/减速度的分别设定模式 (WR3 寄存器 D1=1) 中，它是直线加/减速度驱动在减速时的减速度参数。

在这个模式的 S 曲线加/减速驱动中，减速度直线从 0 增加至减速度的设定值。把减速度设定值作为 D，减速度是下述算式。

$$\text{减速度 (PPS/SEC)} = D \times 125 \times \frac{8,000,000}{R} \times \text{倍率}$$

设定为加速度/减速度分别设定的模式 (WR3 寄存器 D1=1) 后，不能运行加/减速度的自动减速，用手动进行减速。

6.5 初始速度设定

命令编码	命令	参数符号	数据范围	数据长 (字节)
04h	初始速度设定	SV	1~8,000	2

它是加/减速驱动的加速开始时的速度和减速结束时的速度。初始速度设定数值为 SV 的话，初始速度是下述算式。

$$\text{初始速度 (PPS)} = SV \times \frac{8,000,000}{\underbrace{R}_{\text{倍率}}}$$

使用步进马达时，设定自启动频率内的数值。使用伺服马达时，设定太低的数值的话，在定量脉冲驱动的减速结束时，会出现以初始速度的拖曳驱动。必须取√(加速度)以上的数值。比如，加/减速度=125,000PPS/SEC 时，设定√(125,000)=354PPS 以上的数值。

6.6 驱动速度设定

命令编码	命令	参数符号	数据范围	数据长 (字节)
05h	驱动速度设定	V	1~8,000	2

它是加/减速驱动中达到定速区域的速度。定速驱动从该速度开始运行。把这个驱动速度设定在初始速度以下的话，不运行加/减速驱动，开始就运行定速驱动。在编码器的 Z 相搜寻等中采用低速驱动；当检出 Z 相后就立即停止，这时，驱动速度要设定在初始速度以下。

$$\text{驱动速度 (PPS)} = V \times \frac{8,000,000}{\underbrace{R}_{\text{倍率}}}$$

在驱动中可以随便变更驱动速度，在加/减速度驱动的定速区域中，如果重新设定驱动速度，就开始加速或减速到重新设定的速度。达到重新设定的速度后，开始运行定速驱动。

【注意事项】

- S 曲线加/减速的定量脉冲驱动在驱动中不能变更驱动速度。此外，S 曲线加/减速的连续驱动如果在加速中，减速中变更速度，就不能运行正确的 S 曲线，所以请在定速区域内变更。
- 运行直线加/减速的定量脉冲驱动时，在驱动中如果经常变更驱动速度，则在输出脉冲结束的减速时，出现以初始速度拖曳驱动的可能性较大。

6.7 输出脉冲数/插补终点设定

命令编码	命令	参数符号	数据范围	数据长 (字节)
06h	输出脉冲数/插补终点设定	P	输出脉冲数：0~268,435,455 插补终点：-8,388,608~+8,388,607	4

它是定量驱动的输出脉冲总数。运行直线插补圆弧插补时，设定每一个轴的终点，终点坐标设定用 24 位对当前位置的带符号相对数值。输出脉冲数不带符号、插补终点带符号、输出脉冲数、插补终点也用 4 字节长设定。在驱动中可以变更输出脉冲数。

6.8 手动减速点设定

命令编码	命令	参数符号	数据范围	数据长 (字节)
07h	手动减速点设定	DP	0~268,435,455	4

设定在手动减速模式的加减速定量驱动上的减速点。

在手动减速模式时，设定 WR3 寄存器的 D0 位为 1，减速点设定如下：

手动减速点 = 输出脉冲数 — 化费在减速上的脉冲数。

6.9 圆弧中心点设定

命令编码	命令	参数符号	数据范围	数据长（字节）
08h	圆弧中心点设定	C	-8,388,608~+8,388,607	4

设定圆弧插补驱动的中心点，以当前位置的带符号相对值设定中心坐标。

6.10 逻辑位置计数器设定

命令编码	命令	参数符号	数据范围	数据长（字节）
09h	逻辑位置计数器设定	LP	-2,147,483,648~+2,147,483,647	4

设定逻辑位置计数器的数值。

逻辑位置计数器向上/下计数+方向/-方向的驱动输出脉冲。

逻辑位置计数器任何时候都能写，任何时候都能用读出命令读。

6.11 实位计数器设定

命令编码	命令	参数符号	数据范围	数据长（字节）
0Ah	实位计数器设定		-2,147,483,648~+2,147,483,647	4

设定实位计数器的数值。

实位计数器向上/下计数编码器输入脉冲。

实位计数器任何时候都能写，任何时候都能用读出命令读。

6.12 COMP+寄存器设定

命令编码	命令	参数符号	数据范围	数据长（字节）
0Bh	COMP+寄存器设定	CP	-2,147,483,648~+2,147,483,647	4

设定 COMP+寄存器的数值。

COMP+寄存器用作与逻辑/实位计数器比较的寄存器，比较结果存在 RR1 寄存器的 D0 或以 nOUT4/CMPP 信号输出，也可以作为正方向的软件限位。

COMP+寄存器任何时候都能写。

6.13 COMP-寄存器设定

命令编码	命令	参数符号	数据范围	数据长（字节）
0Ch	COMP-寄存器设定	CM	-2,147,483,648~+2,147,483,647	4

设定 COMP-寄存器的数值。

COMP-寄存器用作与逻辑/实位计数器比较的寄存器，比较结果存在 RR1 寄存器的 D1 或以 nOUT5/CMPP 信号输出。可以作为负方向的软件限位。

COMP-寄存器任何时候都能写。

6.14 加速计数器偏移设定

命令编码	命令	参数符号	数据范围	数据长 (字节)
0Dh	加速计数器偏移设定	AO	0~65,535	2

设定加速计数器偏置值。

复位时，加速计数器的偏置值为 8。

6.15 NOP (轴切换用)

命令编码	命令	参数符号	数据范围	数据长 (字节)
0Fh	NOP (轴切换用)			

不被执行任何命令。用于选择任一个轴的 WR1~3 寄存器，RR1~3 寄存器的轴切换。

7 读数据命令

读数据命令是把每个轴的寄存器内容读出至读数据寄存器的命令。在 WR0 寄存器写入轴指定及读数据命令编码后，指定的数据就设定在 RR6, 7 寄存器。CPU 读 RR6, 7 寄存器就可以得到指定的数据，读出的数据都是 2 进制格式，负数值以 2 的补码表示。

【注意事项】

- 读命令的命令处理时间最多需要 250nSEC (CLK=16MHz)。请在命令处理结束后读出 RR6, RR7 寄存器。
- 轴指定一定要只指定 1 个轴，指定 2 个轴以上的话、将根据 X>Y>Z>U 的顺序读出各轴的数据。

7.1 读逻辑位置计数器

命令编码	命令	参数符号	数据范围	数据长 (字节)
10h	逻辑位置计数器读出	LP	-2,147,483,648~+2,147,483,647	4

逻辑位置计数器的当前数值被设定在 RR6, 7 读数据寄存器。

7.2 读实位计数器

命令编码	命令	参数符号	数据范围	数据长 (字节)
11h	实位计数器读出	EP	-2,147,483,648~+2,147,483,647	4

实位计数器的当前数值被设定在 RR6, 7 读出数据寄存器。

7.3 读当前驱动速度

命令编码	命令	参数符号	数据范围	数据长 (字节)
12h	当前驱动速度读出	CV	1~8,000	2

在驱动中的当前驱动速度被设定在 RR6, 7 读出数据寄存器，驱动停止时，被设定为 0。数据的单位和驱动设定数值 (V) 一样。

7.4 读当前加/减速度

命令编码	命令	参数符号	数据范围	数据长 (字节)
13h	当前加减速度读出	CA	1~8,000	2

在驱动中的当前加速度或减速度被设定在 RR6, 7 读出数据寄存器。驱动停止时，读出数据不固定。数据的单位和加速度设定数值 (A) 一样。

8 驱动命令

驱动命令是输出脉冲驱动每个轴的命令及其它相关的命令。在 WR0 命令寄存器写入了轴指定和命令编码，马上就能执行。如果指定多个轴，可以同时发一样的命令。主状态寄存器 RR0 的每个轴 n-DRV 位为 1，表示正在驱动中，驱动结束后 n-DRV 位返回为 0。

当允许使用伺服马达驱动器用的 nINPOS 信号时，等到 nINPOS 输入信号有效，主状态寄存器 RR0 的 n-DRV 位返回为 0。

【注意事项】

- 驱动命令的命令处理时间最多需要 250nSEC (CLK=16MHz)。请在命令处理结束后写入下一个命令。

8.1 正方向定量驱动

命令编码	命令
20h	正方向定量驱动

设定的输出脉冲数以 nPP 输出信号输出。在驱动中每当输出 1 个驱动脉冲，逻辑位置计数器将向上计 1。写入驱动命令之前，一定要正确地设定速度曲线所需的参数和输出脉冲数。

	范围 (R)	JerK (k)	加速度 (A)	初始速度 (SV)	驱动速度 (V)	输出脉冲 (P)
定速驱动	○			○	○	○
直线加减速驱动	○		○	○	○	○
S 曲线加减速驱动	○	○	○	○	○	○

8.2 负方向定量驱动

命令编码	命令
21h	负方向定量驱动

设定的输出脉冲数以 nPM 输出信号输出。在驱动中每当输出 1 个驱动脉冲，逻辑位置计数器将向下计 1。写入驱动命令之前，一定要正确地设定速度曲线所需的参数和输出脉冲数。

8.3 正方向连续驱动

命令编码	命令
22h	正方向连续驱动

直至停止命令或指定的外部信号激活为止，连续以 nPP 输出信号输出脉冲。

在驱动中每当输出 1 个驱动脉冲，逻辑位置计数器将向上计 1。

写入驱动命令之前，一定要正确地设定速度曲线所需的参数及输出脉冲数。

8.4 负方向连续驱动

命令编码	命令
23h	负方向连续驱动

直至停止命令或指定的外部信号激活为止，连续以 nPM 输出信号输出脉冲。在驱动中每当输出 1 个驱动脉冲，逻辑位置计数器将向下计 1。写入驱动命令之前，一定要正确地设定速度曲线所需的参数及输出脉冲数。

8.5 驱动开始暂停

命令编码	命令
24h	驱动开始暂停

暂时停止驱动的开始。同时启动多轴驱动时，使用此功能先把本命令发至要同时启动的各轴后，对各个的轴写驱动命令，然后同时对那些轴写驱动开始释放命令（25h），各轴就同时开始驱动。

8.6 解除暂停驱动开始状态/结束状态清除

命令编码	命令
25h	驱动开始释放/结束状态清除

- ① 驱动开始释放命令（24h）用于解除暂停驱动开始的状态。
- ② 清除 RR1 寄存器的驱动结束状态位 D15~8。

8.7 驱动减速停止

命令编码	命令
26h	驱动减速停止

在驱动脉冲输出过程中，此命令作出减速停止。驱动速度比初始速度慢的时候，也可以用本命令立即停止。在插补驱动中，对主轴写入本命令或立即停止命令，插补驱动就停止。驱动停止时，写入命令也不处理。

8.8 驱动立即停止

命令编码	命令
27h	驱动立即停止

立即停止正在驱动中的脉冲输出。在加/减速驱动中也立即停止。驱动停止时，写入命令也不处理。

9 插补命令

插补命令是有关 2 轴/3 轴直线插补、CW/CCW 圆弧插补，2 轴/3 轴位模式插补及插补驱动的命令。插补命令不需要 WR0 命令寄存器的 D11~8 位的轴指定，请设定为 0。

进行任何一个插补，在插补驱动之前必须具备 2 个条件。

- ① 指定进行插补的轴（设定 WR5 寄存器的 D5~D0）。
- ② 对主轴设定指定的速度参数。

在进行插补驱动中，RR0 主状态寄存器的 D8 (I-DRV) 位为 1，驱动结束后返回至 0。在进行插补驱动中，插补轴的运行位 n-DRV 为 1。

【注意事项】

- 插补命令的命令处理时间最多要 250nSEC(CLK=16MHz 时)。请在命令处理结束后写入下一个命令。

9.1 2 轴直线插补驱动

命令编码	命令
30h	2 轴直线插补驱动

从当前坐标至终点坐标进行 2 轴直线插补。驱动前对运行插补的 2 个轴都要用相对数值对终点设定输出脉冲 (P)。

9.2 3 轴直线插补驱动

命令编码	命令
31h	3 轴直线插补驱动

从当前坐标至终点坐标进行 3 轴直线插补。驱动前对运行插补的 3 个轴都要用相对数值对终点设定输出脉冲 (P)。

9.3 CW 圆弧插补驱动

命令编码	命令
32h	CW 圆弧插补驱动

绕着指定的中点坐标，从当前坐标至终点坐标顺时针方向进行圆弧插补。驱动前，对运行插补的 2 个轴都要把圆弧中心点 (C) 和终点与当前位置的相对数值设定成输出脉冲 (P)。终点坐标设定为 (0, 0)，就可画出真圆弧。

9.4 CCW 圆弧插补驱动

命令编码	命令
33h	CCW 圆弧插补驱动

绕着指定的中点坐标，从当前坐标至终点坐标逆时针方向进行圆弧插补。驱动前，对运行插补的 2 个轴都要把圆弧中心点 (C) 和终点与当前位置的相对数值设定成输出脉冲 (P)。终点坐标设定为 (0, 0)，就可运行真圆弧。

9.5 2 轴位模式插补驱动

命令编码	命令
34h	2 轴位模式插补驱动

进行 2 轴位模式插补。驱动前，设定 2 个插补轴的正方向/负方向的位数据。每个轴在每个方向能设定的位数据最多为 $16 \times 3 = 48$ 位。数据超过 48 位，则在驱动中补充超过部分。

9.6 3 轴位模式插补驱动

命令编码	命令
35h	3 轴位模式插补驱动

进行 3 轴位模式插补。驱动前，设定 3 个插补轴的正方向/负方向的位数据。每个轴在每个方向能设定的位数据最多为 $16 \times 3 = 48$ 位。数据超过 48 位，则在驱动中补充超过部分。

9.7 BP 寄存器写入允许

命令编码	命令
36h	BP 寄存器写入允许

允许写入位模式插补的位模式数据寄存器 (BP1P/M, BP2P/M, BP3P/M)。发出此命令后，就不能写 nWR2~nWR5 寄存器。复位时，不能对位模式寄存器写数据。

9.8 BP 寄存器写入不允许

命令编码	命令
37h	BP 寄存器写入不允许

不允许写入位模式插补的位模式数据寄存器 (BP1P/M, BP2P/M, BP3P/M)。此命令发出后就能写 nWR2~nWR5 寄存器。

9.9 BP 数据堆栈

命令编码	命令
38h	BP 数据堆栈

把写入寄存器 (BP1P/M, BP2P/M, BP3P/M) 的位模式数据移动至内部寄存器存放。发出 BP 数据堆栈命令后堆栈计数器 (SC) 就增加 1。堆栈计数器 (SC) 计到 3 后，就不能发此命令。

9.10 BP 清除

命令编码	命令
39h	BP 清除

清除在内部寄存器存积的所有位模式数据，堆栈计数器 (SC) 为 0。

9.11 插补单步

命令编码	命令
3Ah	插补单步

在插补驱动时，步进传送每一个脉冲。WR5 寄存器的 D12 位置 1，用命令设定为插补步进模式，发插补命令后，就执行单步插补。

9.12 减速有效

命令编码	命令
3Bh	减速有效

允许在加/减速插补驱动中使用自动或手动减速。运行单个加/减速插补驱动时，驱动前一定要发此命令。连续插补驱动时先把减速设定为无效，然后才开始插补驱动，在写入最终插补节点的插补命令前，写入此命令允许减速。

复位后，处于禁止减速状态。此命令使之进入允许减速状态，直到写入禁止命令（3C）或复位为止。减速有效/无效模式只用于插补驱动。每个轴单独驱动时，自动减速或手动减速一直保持有效状态。

9.13 禁止减速

命令编码	命令
3Ch	减速无效

禁止在加/减速插补驱动中使用自动或手动减速。

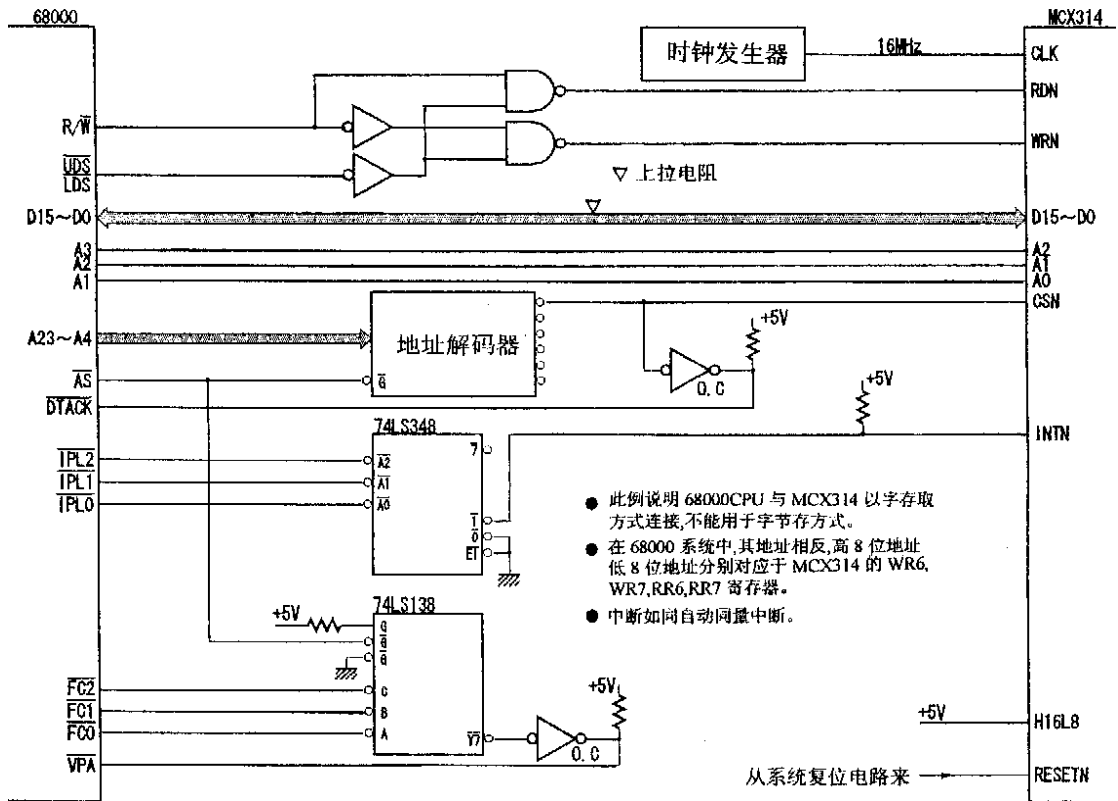
9.14 插补中断清除

命令编码	命令
3Dh	插补中断清除

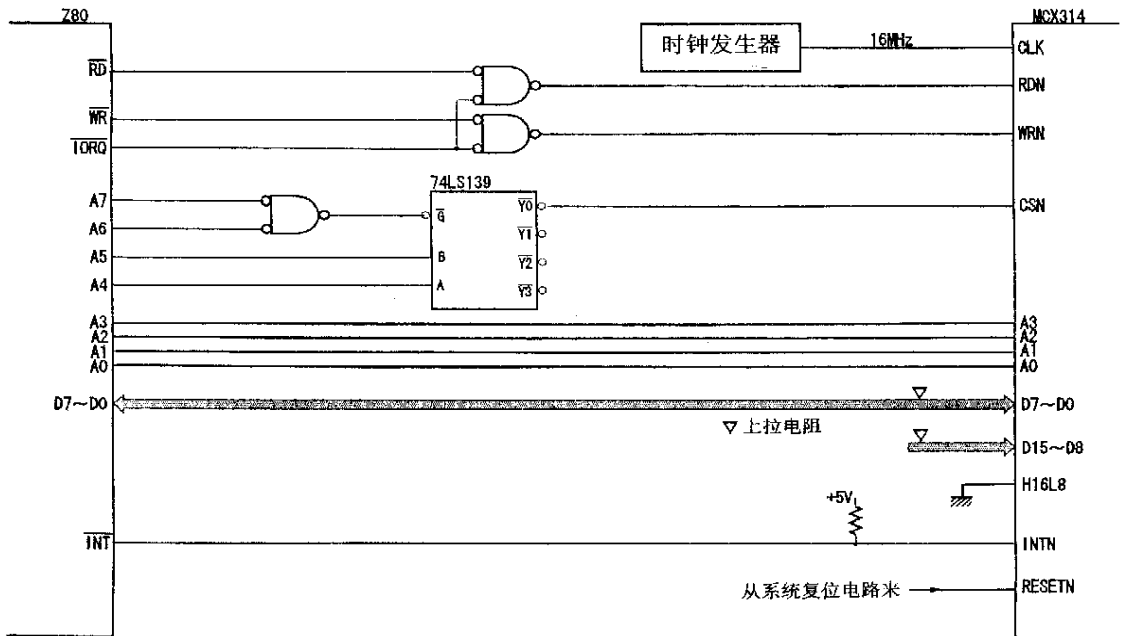
清除在位模式插补或连续插补中发生的中断。在位模式插补中置 WR5 寄存器的 D15 位为 1，当堆栈计数器（SC）从 2 变为 1 时，中断发生。此外，在连线插补中置 WR5 寄存器的 D14 位为 1，则在允许写入下一个插补节点的数据及插补驱动命令时，就发生中断。

10 I/O 信号连接例子

10.1 与 68000CPU 的连接例子

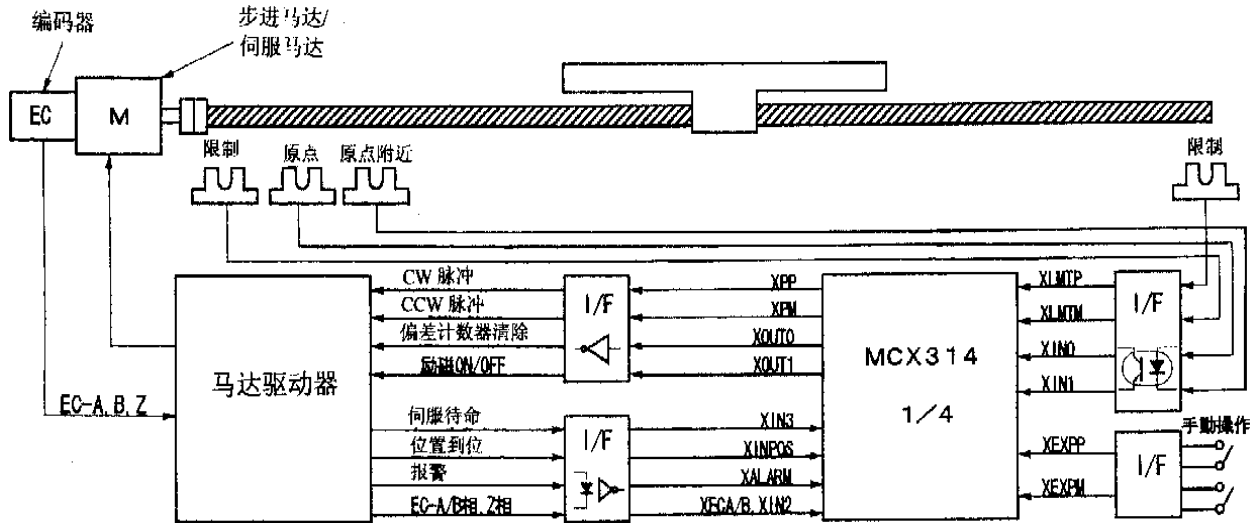


10.2 与 Z80CPU 连接的例子



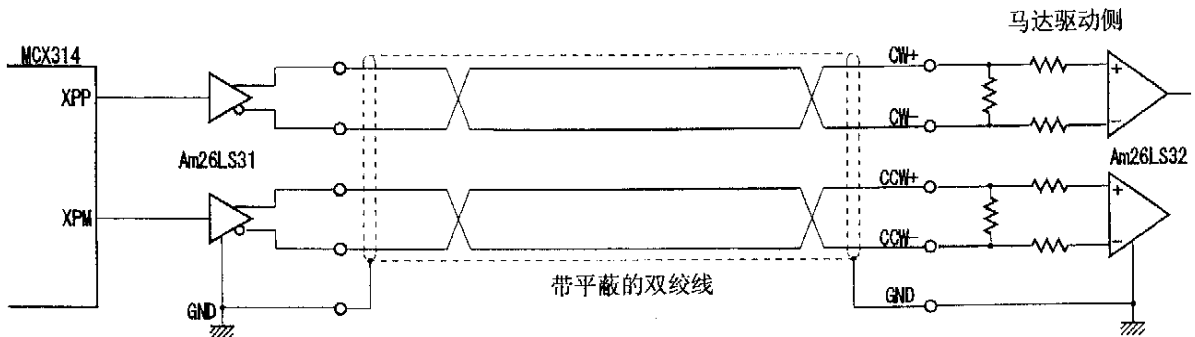
10.3 运动系统构成例子

运动系统的 1 个轴部分的例子如下图所示。4 个轴可有同样的结构。

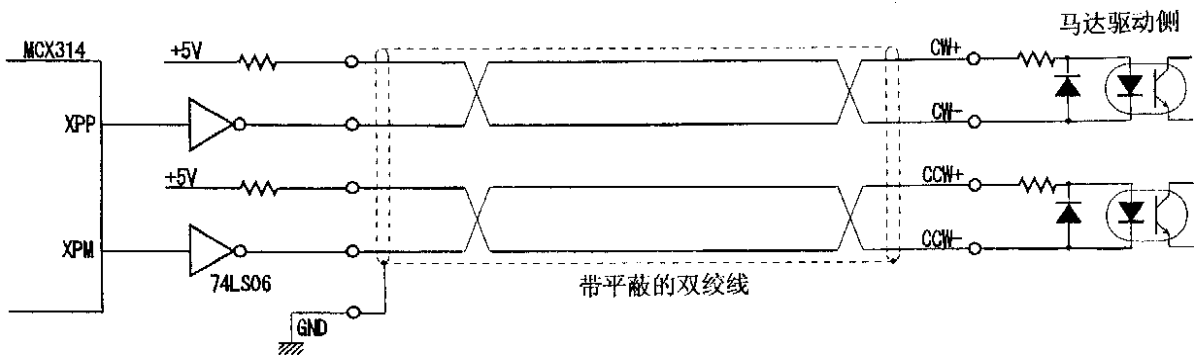


10.4 驱动脉冲输出电路

■ 差动线驱动输出



■ 集电极开路 TTL 输出

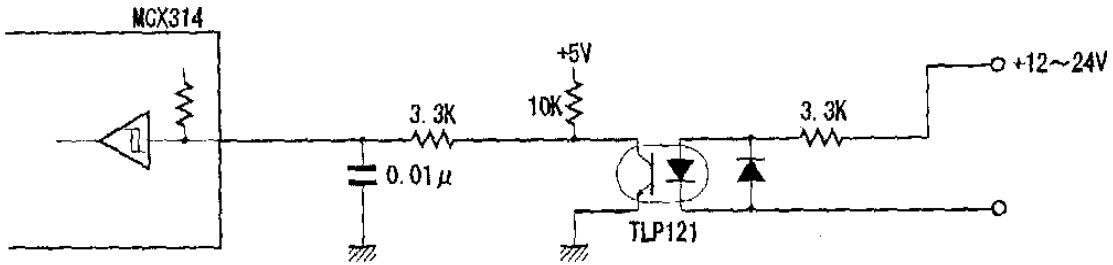


考虑到 EMC，驱动脉冲输出信号推荐使用双绞屏蔽线。

10.5 限位等输入信号的连接例子

除了数据信号 (D15~D0) 以外, 此 IC 的所有输入信号都是在 IC 内部用高阻上拉的史密特触发器输入, 如下图所示。

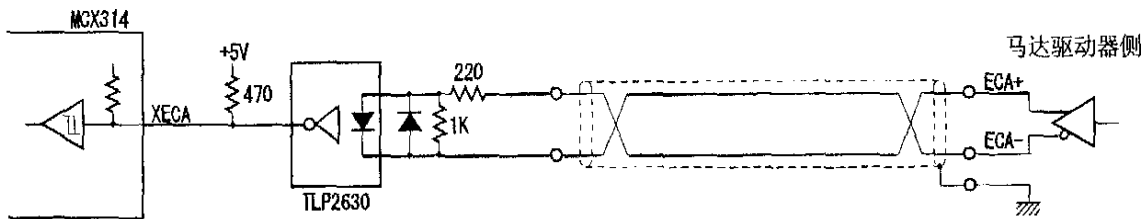
限位信号等信号容易受干扰, 所以只用光偶不能吸收噪音。如下图所示, 用 CR 积分型的滤波器有更好的效果。



此电路的响应时间有 0.2~0.4mSEC 左右。

10.6 编码输入信号的连接例子

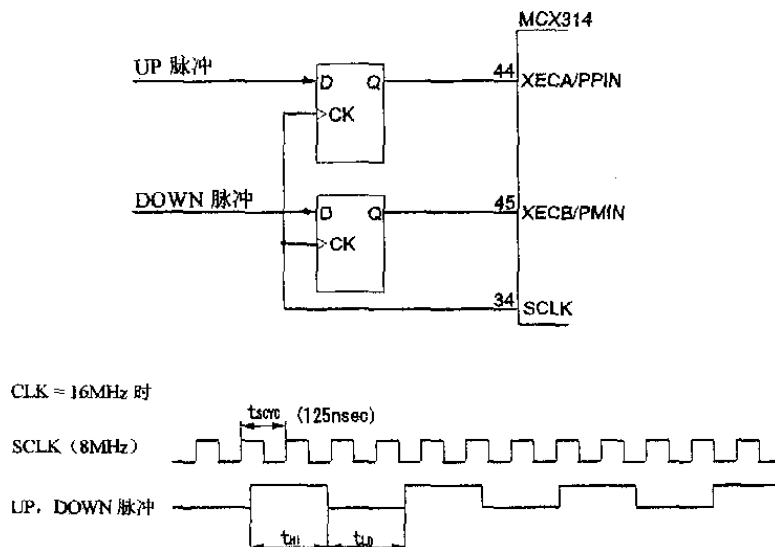
下图所示是 MCX314 输入电路的例子, 它接收由高速光偶差动线驱动输出的编码器信号。



UP/DOWN 脉冲输入模式使用时应注意

用 UP/DOWN 脉冲输入模式使用编码器输入信号 (nECA/PPIN, nECB/PMIN) 时, 如下图所示, 把 UP 脉冲和 DOWN 脉冲由 SCLK 同步输入。直接输入的话, 将计错数据。因此 UP、DOWN 脉冲的 Hi 电平脉冲宽度及 Low 电平脉冲宽度都要大于 SCLK 的同步时间 (CLK=16MHz 输入时, SCLK=1/2CLK, 所以是 125nSEC)。

使用 A/B 2 相脉冲输入模式时, 可以直接输入。



11 控制程序的例子

用 C 语言编写的 MCX314 控制程序的例子, 编译器采用 Baudland 公司的 Turbo C++4.0。基于 DOS-5 个人电脑 ISA 总线上的电路。

```
#include <stdio.h>
#include <conio.h>
//----mcx314 寄存器地址定义
#define adr 0x2a0 //基本地址

#define wr0 0x0 //命令寄存器
#define wr1 0x2 //模式寄存器 1
#define wr2 0x4 //模式寄存器 2
#define wr3 0x6 //模式寄存器 3
#define wr4 0x8 //输出寄存器
#define wr5 0xa //插补模式寄存器
#define wr6 0xc //低位写入数据寄存器

#define wr7 0xe //高位写入数据寄存器

#define rr0 0x0 //主状态寄存器
#define rr1 0x2 //状态寄存器 1
#define rr2 0x4 //状态寄存器 2
#define rr3 0x6 //状态寄存器 3
#define rr4 0x8 //输入寄存器 1
#define rr5 0xa //输入寄存器 2

#define rr6 0xc //低位读出数据寄存器
#define rr7 0xe //高位读出数据寄存器

#define bp1p 0x4 //BP第 1 轴+方向数据寄存器
#define bp1m 0x6 //BP第 1 轴-方向数据寄存器
#define bp2p 0x8 //BP第 2 轴+方向数据寄存器
#define bp2m 0xa //BP第 2 轴-方向数据寄存器
#define bp3p 0xc //BP第 3 轴+方向数据寄存器
#define bp3m 0xe //BP第 3 轴-方向数据寄存器

// wreg1(轴指定,数据)----- 写入寄存器 1 设定
void wreg1(int axis, int wdata)
{
    outpw(adr+wr0,(axis<<8) + 0xf); //轴指定
    outpw(adr+wr1, wdata);
}

// wreg2(轴指定,数据)----- 写入寄存器 2 设定
void wreg2(int axis, int wdata)
{
    outpw(adr+wr0,(axis<<8) + 0xf); //轴指定
    outpw(adr+wr2,wdata);
}

// wreg3(轴指定,数据)----- 写入寄存器 3 设定
void wreg3(int axis, int wdata)
{
    outpw(adr+wr0,(axis<<8) + 0xf); //轴指定
    outpw(adr+wr3, wdata);
}

// command(轴指定,命令编码)---- 命令写入
void command(int axis, int cmd)
{
    outpw(adr+wr0,(axis<<8) + cmd);
}

// range(轴指定,数据)----- 范围 (R) 设定
void range1(int axis, long wdata)
{
    outpw(adr+wr7,(wdata>>16)&0xffff);
    outpw(adr+wr6,wdata & 0xffff);
    outpw(adr+wr0,(axis<<8) + 0x00);
}

// acac(轴指定,数据)----- 变化率(K)设定
void acac(int axis, int wdata)
{
    outpw(adr+wr6,wdata);
    outpw(adr+wr0,(axis<<8) + 0x01);
}

// acc(轴指定,数据)----- 加/减速度(A)设定
void acc(int axis, int wdata)
{
    outpw(adr+wr6,wdata);
    outpw(adr+wr0,(axis<<8) + 0x02);
}

}

// dec(轴指定,数据)----- 减速度 (D) 设定
void dec(int axis, int wdata)
{
    outpw(adr+wr6, wdata);
    outpw(adr+wr0,(axis<<8) + 0x03);
}

// startv(轴指定,数据)----- 初始速度 (SV) 设定
void startv(int axis, int wdata)
{
    outpw(adr+wr6,wdata);
    outpw(adr+wr0,(axis<<8) + 0x04);
}

// speed(轴指定,数据)----- 驱动速度 (V) 设定
void speed(int axis, int wdata)
{
    outpw(adr+wr6, wdata);
    outpw(adr+wr0,(axis<<8) + 0x05);
}

// pulse(轴指定,数据)----- 输出脉冲数/终点 (P) 设定
void pulse(int axis, long wdata)
{
    outpw(adr+wr7,(wdata>>16)&0xffff);
    outpw(adr+wr6,wdata & 0xffff);
    outpw(adr+wr0,(axis<<8) + 0x06);
}

// decp(轴指定,数据)----- 手动减速度 (DP) 设定
void decp(int axis, long wdata)
{
    outpw(adr+wr7,(wdata>>16)&0xffff);
    outpw(adr+wr6,wdata & 0xffff);
    outpw(adr+wr0,(axis<<8) + 0x07);
}

// center(轴指定,数据)----- 圆弧中心点 (C) 设定
void center(int axis, long wdata)
{
    outpw(adr+wr7,(wdata>>16)&0xffff);
    outpw(adr+wr6,wdata & 0xffff);
    outpw(adr+wr0,(axis<<8) + 0x08);
}

// lp(轴指定,数据)----- 逻辑位置计数器(LP)设定
void lp(int axis, long wdata)
{
    outpw(adr+wr7,(wdata>>16)&0xffff);
    outpw(adr+wr6,wdata & 0xffff);
    outpw(adr+wr0,(axis<<8) + 0x09);
}

// ep(轴指定,数据)----- 逻辑位置计数器(EP)设定
void ep(int axis, long wdata)
{
    outpw(adr+wr7,(wdata>>16)&0xffff);
    outpw(adr+wr6,wdata & 0xffff);
    outpw(adr+wr0,(axis<<8) + 0x0a);
}

// compp(轴指定,数据)----- COMP + (CP) 设定
void compp(int axis, long wdata)
{
    outpw(adr+wr7,(wdata>>16)&0xffff);
    outpw(adr+wr6,wdata & 0xffff);
    outpw(adr+wr0,(axis<<8) + 0x0b);
}

// compm(轴指定,数据)----- COMP - (CM) 设定
void compm(int axis, long wdata)
{
    outpw(adr+wr7,(wdata>>16)&0xffff);
    outpw(adr+wr6,wdata & 0xffff);
    outpw(adr+wr0,(axis<<8) + 0x0c);
}

// accofst(轴指定,数据)----- 加速计数器偏移(AO)设定
```

```

void accofst(int axis, long wdata)
{
    outpw(adr+wr7,(wdata>>16)& 0xffff);
    outpw(adr+wr6,wdata & 0xffff);
    outpw(adr+wr0,(axis<<8) + 0x0d);
}
// readlp(轴指定)----- 逻辑位置计数器数值(LP)读出

long readlp(int axis)
{
    long a; long d6; long d7;
    outpw(adr+wr0,(axis<<8) + 0x10);
    d6 = inpw(adr+rr6);d7 = inpw(adr+rr7);
    a = d6 + (d7<<16);
    return(a);
}
// readep(轴指定)----- 逻辑位置计数器数值(EP)读出

long readep(int axis)
{
    long a; long d6; long d7;
    outpw(adr+wr0,(axis<<8) + 0x11);
    d6 = inpw(adr+rr6);d7 = inpw(adr+rr7);
    a = d6 + (d7<<16);
    return(a);
}
// wait(轴指定)----- 等驱动结束

void wait(int axis)
{
    while(inpw(adr+rr0) & axis);
}

// next_wait()----- 等连续插补下一个数据设定
//
void next_wait(void)
{
    while((inpw(adr+rr0) & 0x0200) == 0;
}

// bp_wait()----- 等 BP 插补下一个数据设定

void bp_wait(void)
{
    while((inpw(adr+rr0) & 0x6000) == 0x6000);
}

// homesrch()----- 全轴原点检出
void homesrch(void)
{
    // [动作] (1)~(3)各轴都相同
    // (1)原点信号(IN1)关闭时
    speed(0xf, 2,000); // 用速度 20,000PPS-连续驱动。
    if((inpw(adr+rr4) & 0x2) == 0x2) // IN1 信号 ON 后,减速停止。
    {
        command(0x1, 0x23);
    }
    if((inpw(adr+rr4) & 0x200) == 0x200)
    {
        command(0x2, 0x23);
    }
    if((inpw(adr+rr5) & 0x2) == 0x2)
    {
        command(0x4, 0x23);
    }
    if((inpw(adr+rr5) & 0x200) == 0x200)
    {
        command(0x8, 0x23);
    }
    wait(0xf);
    wreg1(0xf, 0x000c); // (2)用速度 500PPS+连续驱动。
    speed(0xf, 50); // IN1 信号 OFF 后,立即停止。
    command(0xf, 0x22);
    wait(0xf);
    wreg1(0xf, 0x0000); // (3)用速度 40,000PPS 向-方向。
    speed(0xf, 4,000); // 100 脉冲偏移移动
    pulse(0xf, 100);
    command(0xf, 0x21);
    wait(0xf);
    lp(0xf, 0) // (4) X, Y, Z, U 轴 LP = 0
    wreg2(0x7, 0x0003); // X, Y, Z 轴软件限制:ON
    compmp(0x1, 100,000); // X: -1,000 ~ +100,000
    compmp(0x1, -1,000); // Y: -500 ~ +50,000
    compmp(0x2, 50,000); // Y: -500 ~ +50,000
    compmp(0x2, -500);
}

```

```

compmp(0x4, 10,000); // Z: -100 ~ +10,000
compmp(0x4, -100);
}

//*****<<主要>>*****
void main(void)
{
    int count;

    outpw(adr+wr0, 0x8,000); //软件复位
    for(count = 0; count < 2; ++count);

    command(0xf, 0xf); //-----全轴模式设定-----
    outpw(adr+wr1, 0x0000); //模式寄存器 1: 00000000 00000000
    outpw(adr+wr2, 0x0000); //模式寄存器 2: 00000000 00000000
    outpw(adr+wr3, 0x0000); //模式寄存器 3: 00000000 00000000
    outpw(adr+wr4, 0x0000); //通用输出寄存器: 00000000 00000000
    outpw(adr+wr5, 0x0024); //插补模式寄存器: 00000000 00100100

    //-----全轴操作参数初期设定-----
    //A0 = 0
    //R = 800,000(倍率 = 10)
    //K = 1,010(变化率 = 619KPPS/SEC)
    //A = 100(加/减速度 = 125KPPS/SEC)
    //D = 100(减速度 = 125KPPS/SEC)
    //SV = 100(初始速度 = 1,000PPS)
    //V = 4,000(驱动速度 = 40,000PPS)
    //P = 100,000(输出脉冲数 = 100,000)
    //LP = 0(逻辑位置计数器 = 0)

    homesrch(); //-----全轴原点检出-----

    //-----X、Y 轴直线加减速驱动----
    //A=200(加/减速度= 250KPPS/SEC)
    //V = 4,000(驱动速度 = 40,000PPS)
    //xP = 80,000
    //yP = 40,000
    //+定量驱动
    wait(0x3); //等驱动结束

    wreg3(0x3, 0x0,004); //-----X、Y 轴 S 曲线加减速驱动--
    acac(0x3, 1,010); //K= 1,010(变化率= 619KPPS/SEC)
    acc(0x3, 200); //A=200(加/减速度= 250KPPS/SEC)
    speed(0x3, 4,000); //V = 4,000(驱动速度 = 40,000PPS)
    pulse(0x1, 50,000); xP = 50,000
    pulse(0x2, 25,000); yP = 25,000
    command(0x3, 0x21); // -定量驱动
    wait(0x3); //S 曲线加/减速模式解除

    outpw(adr+wr5, 0x0124); //-----X、Y 轴直线插补驱动----
    range(0x1, 800,000); //a×1 = x, a×2 = y, a×3 = z 线速固定
    range(0x2, 1,131,371); //a×1/R = 800,000(倍率 = 10)
    speed(0x1, 100); //a×2/R = 800,000×1.414
    pulse(0x1, 5,000); //驱动速度 = 1,000PPS 定速
    pulse(0x2, -2,000); //xP = +5,000(终点 X = +5,000)
    command(0x0, 0x30); //yP = -2,000(终点 Y = -2,000)
    //2 轴直线插补

    wait(0x3);

    //-----X、Y 轴圆弧插补驱动---
    outpw(adr+wr5, 0x0124); //a×1 = x, a×2 = y, a×3 = z 线速常数
    range(0x1, 800,000); //a×1/R = 800,000(倍率 = 10)
    range(0x2, 1,131,371); //a×2/R = 800000×1.414
    speed(0x1, 100); //驱动速度 = 1,000PPS 定速
    center(0x1, -5,000); //xC = -5,000(中心 X = -5,000)
    center(0x2, 0); //yC = 0 (中心 Y = 0)
    pulse(0x1, 0); //xP = 0 (终点 X = 0)真圆
    pulse(0x2, 0); //yP = 0 (终点 Y = 0)
    command(0x0, 0x33); //CCW 圆弧插补
    wait(0x3);

    //-----X、Y 轴位模式插补---
    // (图 2.24 例子)
    speed(0x1, 1); //驱动速度 = 10PPS 定速
    command(0, 0x36); //位模式数据写入
    outpw(adr+bp1p,0x0000); //0~15 位模式数据写入
    outpw(adr+bp1m, 0x2bff);
    outpw(adr+bp2p, 0xffd4);
    outpw(adr+bp2m,0x0000);
    command(0, 0x38); //堆栈
    outpw(adr+bp1p, 0xf6fe); //16~31 位数据写入
    outpw(adr+bp1m,0x0000);
    outpw(adr+bp2p,0x000f);
    outpw(adr+bp2m,0x3fc0);
    command(0, 0x38);
}

```

```

outpw(adr+bp1p, 0x1fdb); //32 ~ 47 位数据写入
outpw(adr+bp1m, 0x0000);
outpw(adr+bp2p, 0x00ff);
outpw(adr+bp2m, 0xfc00);
command(0, 0x38);

command(0, 0x34); //2 轴 BP 插补驱动开始

bp_wait(); //数据等待写入

outpw(adr+bp1p, 0x4000); //48 ~ 63 位数据写入

outpw(adr+bp1m, 0x7ff5);
outpw(adr+bp2p, 0x0000);
outpw(adr+bp2m, 0x0aff);
command(0, 0x38);
command(0, 0x37); //不允许位模式数据写入

wait(0x3); //等待驱动结束

speed(0x1, 100); //-----X, Y 轴连续插补 (图 2.29 例子)
//驱动速度 = 1,000PPS 定速

pulse(0x1, 4,500); //node1
pulse(0x2, 0);
command(0, 0x30);

next_wait(); //次数据等待
center(0x1, 0); //node2
center(0x2, 1,500);
pulse(0x1, 1,500);
pulse(0x2, 1,500);
command(0, 0x33);

next_wait();
pulse(0x1, 0); //node3
pulse(0x2, 1,500);
command(0, 0x30);
next_wait();
center(0x1, -1,500); //node4
center(0x2, 0);
pulse(0x1, -1,500);
pulse(0x2, 1,500);
command(0, 0x33);

next_wait();
pulse(0x1, -4,500); //node5
pulse(0x2, 0);
command(0, 0x30);
next_wait();
center(0x1, 0); //node6
center(0x2, -1,500);
pulse(0x1, -1,500);
pulse(0x2, -1,500);
command(0, 0x33);
next_wait();
pulse(0x1, 0); //node7
pulse(0x2, -1,500);
command(0, 0x30);
next_wait();
center(0x1, 1,500); //node8
center(0x2, 0);
pulse(0x1, 1,500);
pulse(0x2, -1,500);
command(0, 0x33);
wait(0x3);
}

```

12 电气的特性

12.1 DC 特性

■ 绝对最大定额

项目	符号	定额	单位
电源电压	V _{DD}	-0.3 ~ +7.0	V
输入电压	V _{IN}	-0.3 ~ V _{DD} +0.3	V
输入电流	I _{IN}	±10	mA
保存温度	T _{STG}	-40 ~ +125	°C

项目	符号	定额	单位
电源电压	V _{DD}	4.75 ~ 5.25	V
周围温度	T _a	0 ~ +85	°C

如果你想 0°C 下使用，请与我们技术支持联系。

■ DC 特性

(T_a = 0 ~ +85°C, V_{DD} 5V±5%)

项目	符号	条件	最小值	典型值	最大值	单位	备注
高电平输入电压	V _{IH}		2.2			V	
低电平输入电压	V _{IL}				0.8	V	
高电平输入电流	I _{IH}	V _{IN} = V _{DD}	-10		10	μA	
低电平输入电流	I _{IL}	V _{IN} = OV	-10		10	μA	D15 ~ D0 输入信号
		V _{IN} = OV	-200		-10	μA	除 D15 ~ D0 之外的输入信号
高电平输出电压	V _{OH}	I _{OH} = -1 μA	V _{DD} -0.05			V	注 1
		I _{OH} = -4mA	2.4			V	除 D15 ~ D0 之外的输出信号
		I _{OH} = -8mA	2.4			V	D15 ~ D0 输出信号
低电平输出电压	V _{OL}	I _{OL} = 1 μA			0.05	V	
		I _{OL} = 4mA			0.4	V	除 D15 ~ D0 之外的输出信号
		I _{OL} = 8mA			0.4	V	D15 ~ D0 输出信号
输出泄漏电流	I _{OZ}	V _{OUT} = V _{DD} or OV	-10		10	μA	D15 ~ D0, BUSYN, INTN
斯密特触发器泄漏电流引起的电压	V _H			0.3		V	
消耗电流	I _{DD}	I _{I/O} = 0 mA, CLK = 16MHz		52	90	mA	

注 1: 因为 BUSYN, INTN 输出信号是开漏输出，所以它们没有高电平输出。

■ 引脚电容

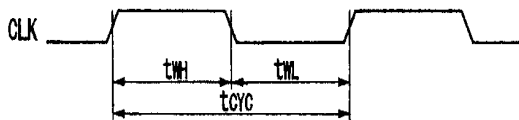
项目	符号	条件	最小值	典型值	最大值	单位	备注
I/O 电容	C _{IO}	T _a = 25°C, f = 1MHz			10	pF	D15 ~ D0
输入电容	C _I				10	pF	其它输入引脚

12.2 AC 延迟特性

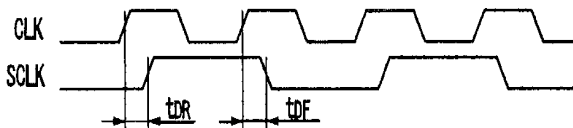
(T_a = 0~85°C, V_{DD} = +5V±5% 输出负荷的条件: 85pF+1TTL)

12.2.1 时钟

■ CLK 输入信号



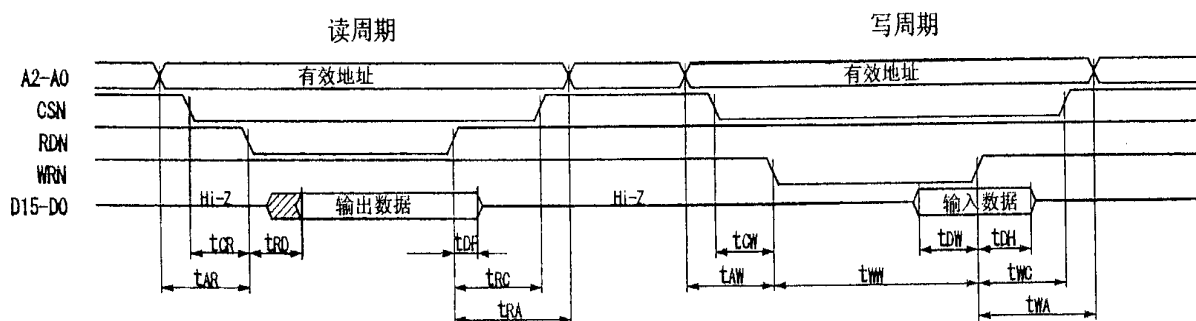
■ SCLK 输出信号



- RESETN 在 Low 期间，SCLK 不被输出。

符号	项 目	最小值	最大值	单位
tCYC	CLK 周期	62.5		nS
tWH	CLK 高电平宽度	20		nS
tWL	CLK 低电平宽度	20		nS
tDR	CLK ↑ → SCLK ↑ 延迟时间		21	nS
tDF	CLK ↑ → SCLK ↓ 延迟时间		23	nS

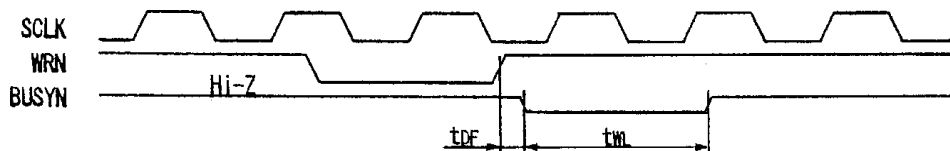
12.2.2 CPU 读/写周期



- 上图是 16 位数据总线 (H16L8=Hi) 的信号, 8 位数据总线 () 的时候, 图中地址信号是 A3~A0, 数据信号是 D7~D0。
- RDN 和 CSN 都变为低电平之后, 读出周期的数据信号 (D15~D0) 处于输出状态, RDN 返回到高电平后, 它仍然在 tDF 期间保持输出状态。因此, 请注意发生冲突。

符号	项 目	最小值	最大值	单 位
tAR	地址建立时间 (to RDN ↓)	0		nS
tCR	CSN 建立时间 (to RDN ↓)	0		nS
tRD	输出数据延迟时间 (from RDN ↓)		29	nS
tDF	输出数据保持时间 (from RDN ↑)	0	30	nS
tRC	CSN 保持时间 (from RDN ↑)	0		nS
tRA	地址保持时间 (from RDN ↑)	0		nS
tAW	地址建立时间 (to WRN ↓)	0		nS
tCW	CSN 建立时间 (to WRN ↓)	0		nS
tWW	WRN Low 电平脉冲宽度	50		nS
tDW	输入数据设定时间 (to WRN ↑)	30		nS
tDH	输入数据保持时间 (from WRN ↑)	10		nS
tWC	CSN 保持时间 (from WRN ↑)	5		nS
tWA	地址保持时间 (from WRN ↑)	5		nS

12.2.3 BUSYN 信号



- BUSYN 输出信号在 WRN 的上升沿后最大 $SCLK \times 2$ 周期期间保持 Low 激活，这期间不能读/写此 IC。

符号	项 目	最小值	最大值	单 位
tDF	WRN $\uparrow \rightarrow$ BUSYN \downarrow 延迟时间		32	nS
tWL	BUSYN 低电平宽度		tCYC \times 4 + 30	nS

tCYC 是 CLK 的周期。

12.2.4 SCLK/输出信号延迟

下面的输出信号一直与 SCLK 输出信号同步，电平随 SCLK 的上升沿变化。

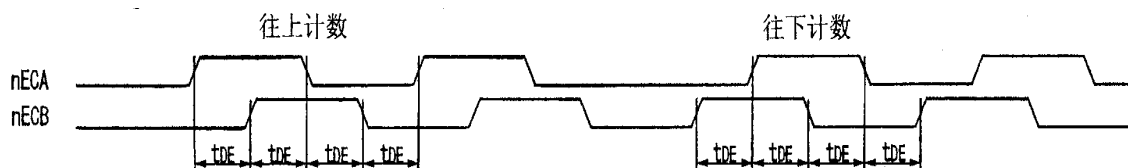
输出信号：nPP/PLS, nPM/DIR, nDRIVE, nASND, nDSND, nCMPP, nCMPM



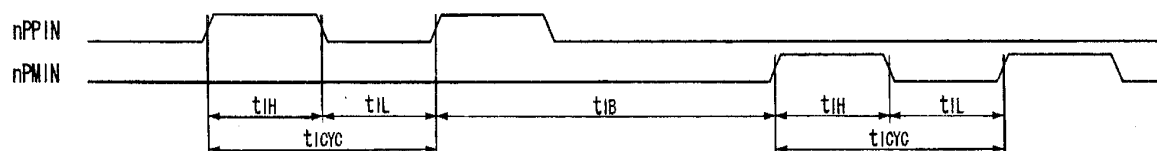
符号	项 目	最小值	最大值	单 位
tDD	SCLK $\uparrow \rightarrow$ 输出信号 $\uparrow \downarrow$ 延迟时间	0	20	nS

12.2.5 输入脉冲

- 2 相脉冲输入模式



- 上下脉冲输入模式



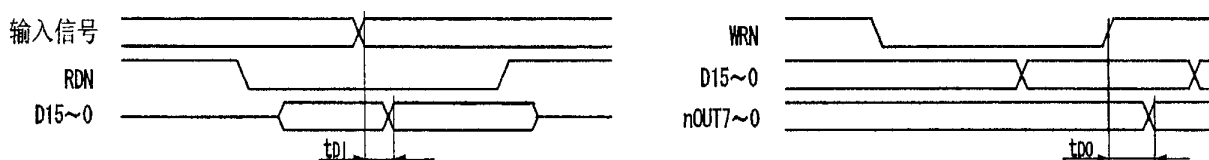
- 使用上下脉冲输入模式时，用 SCLK 把 nPPIN, nPMIN 信号同步化后输入。（参考 P67 图）
- 使用 2 相脉冲输入模式时，nECA, nECB 输入变化后，实位计数器在最大 4 个 SCLK 周期后变为改变后的数值。
- 使用上下脉冲输入模式时，从 nPPIN, nPMIN 输入的 \uparrow 起最大 4 个 SCLK 周期之后，实位计数器将变为改变后的数值。

符号	项 目	最小值	最大值	单 位
tDE	nECA,nECB 位相差时间	tCYC×2+20		nS
tIH	nPPIN,nPMIN 高电平宽度	tCYC×2+20		nS
tIL	nPPIN,nPMIN 低电平宽度	tCYC×2+20		nS
tICYC	nPPIN,nPMIN 周期	tCYC×4+20		nS
tIB	nPPIN ↑ ←→nPMIN ↑ 时间	tCYC×4+20		nS

12.2.6 通用 I/O 信号

左下图表示在 RR4, RR5 寄存器读出输入信号 (nIN3~0, nEXPP, nEXPM, nINPOS, nALARM) 的延迟时间。

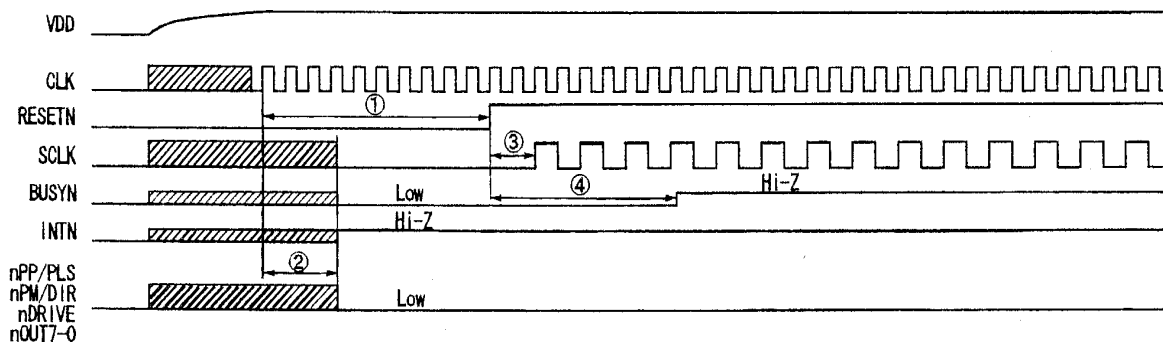
右下图表示在 nWR3, WR4 寄存器写入通用输出信号数据的延迟时间。



符号	项 目	最小值	最大值	单 位
tDI	输入信号 → 数据延迟时间		32	nS
tDO	WRN ↑ →nOUT7~0 设定结束时间		32	nS

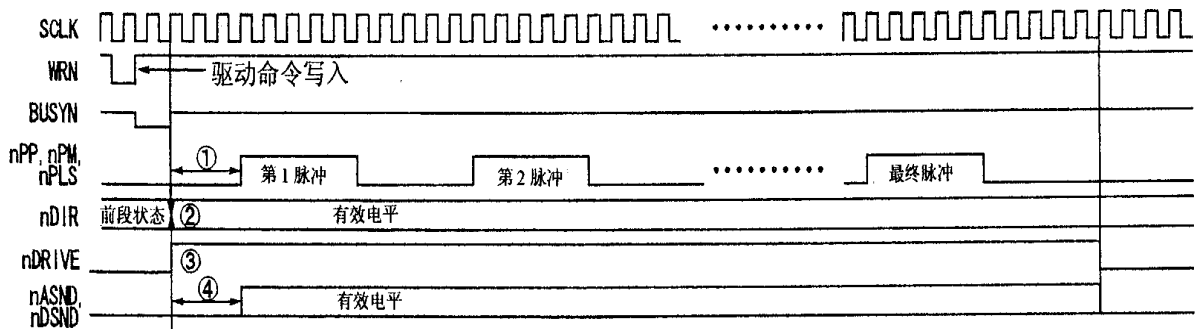
13 I/O 信号时序

13.1 上电时序



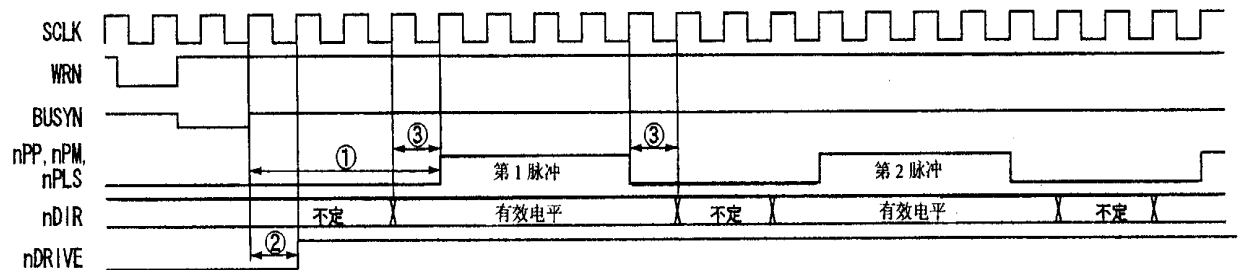
- ①复位输入信号 RESETN: 第一要 CLK 输入后; 第二要 CLK×4 周期以上低电平。
- ②打开电源时的输入信号: 第一要 RESETN 在低电平; 第二要 CLK 处于输入状态; 第三要在最大 CLK×4 周期后, 确定如上图所示的电平。
- ③RESETN 达到高电平后, SCLK 在最大 CLK×2 周期后输出。
- ④RESETN 达到高电平后, BUSYN 在最大 CLK×8 周期保持低电平。在这期间不能读/写此 IC。

13.2 驱动开始/结束时序



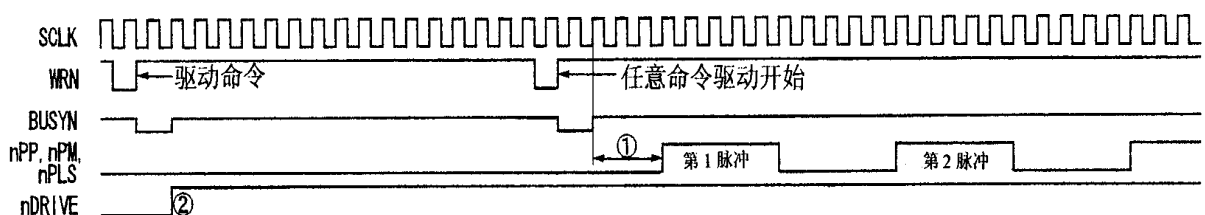
- ① 上图所示的驱动脉冲（nPP，nPM，nPLS）是正脉冲，从 BUSYN 的 ↑ 起 3 个 SCLK 周期后，第一脉冲将被输出。
- ② 驱动输出脉冲方式设定为 1 脉冲方式时，由 BUSYN 的 ↑ 使 nDIR（方向）信号有效。驱动结束后，仍保持此电平直到写入下一个驱动命令。不过，插补驱动时不是这样。
- ③ 由 BUSYN 的 ↑ 使 nDRIVE 变为高电平，在最终脉冲的低电平之后返回低电平。
- ④ 从 BUSYN 的 ↑ 起 3 个 SCLK 周期后 nASND，nDSND 变为有效电平，在最终脉冲的低电平后返回至低电平。

13.3 插补驱动时序



- ① 插补时，从 BUSYN 的 ↑ 起 4 个 SCLK 周期后输出第一驱动脉冲。
- ② 从 BUSYN 的 ↑ 起 1 个 SCLK 周期后，nDRIVE 变为高电平。
- ③ 驱动输出脉冲方式设定为 1 脉冲方式时，nDIR（方向）信号在驱动脉冲的高电平宽度与其前后 1 个 SCLK 周期后变为有效电平。（驱动脉冲：正逻辑脉冲时）

13.4 驱动开始释放时序

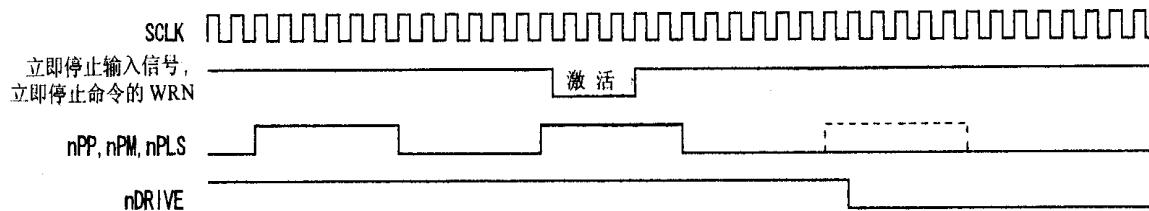


- ① 各个轴的驱动脉冲（nPP，nPM，nPLS）从驱动开始释放命令写入的 BUSYN ↑ 起 3 个 SCLK 周期后同时输出第一脉冲。
- ② 由每个轴驱动命令写入的 BUSYN ↑ 使 nDRIVE 变为高电平。

13.5 立即停止时序

这是立即停止输入信号和立即停止命令的动作时序。立即停止输入信号是 EMGN，nLMTP/M（设定为立即停止模式时）和 nALARM。

立即停止输入信号有效或者写入立即停止命令后，输出完当前驱动脉冲后，就停止驱动脉冲输出。

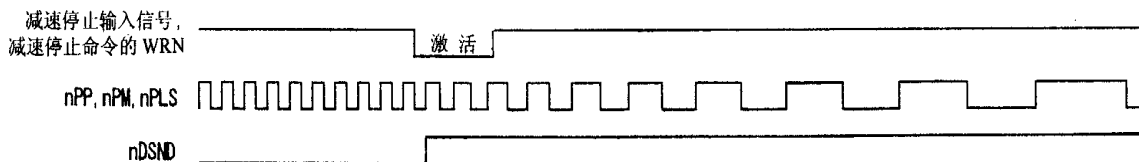


立即停止输入信号由 SCLK 的 \uparrow 存入 IC 内部，所以比 1 个 SCLK 周期短的作用脉冲有可能不被接受。

13.6 驱动减速停止时序

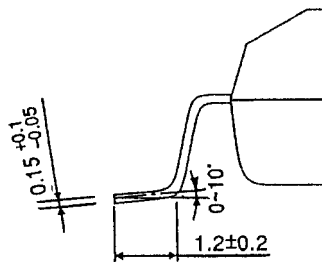
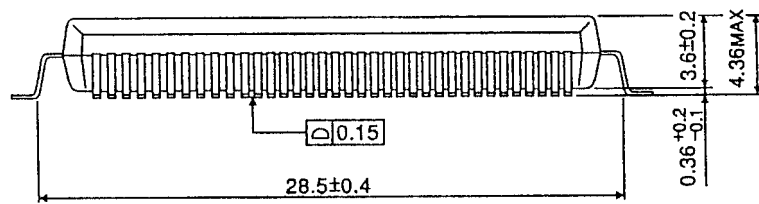
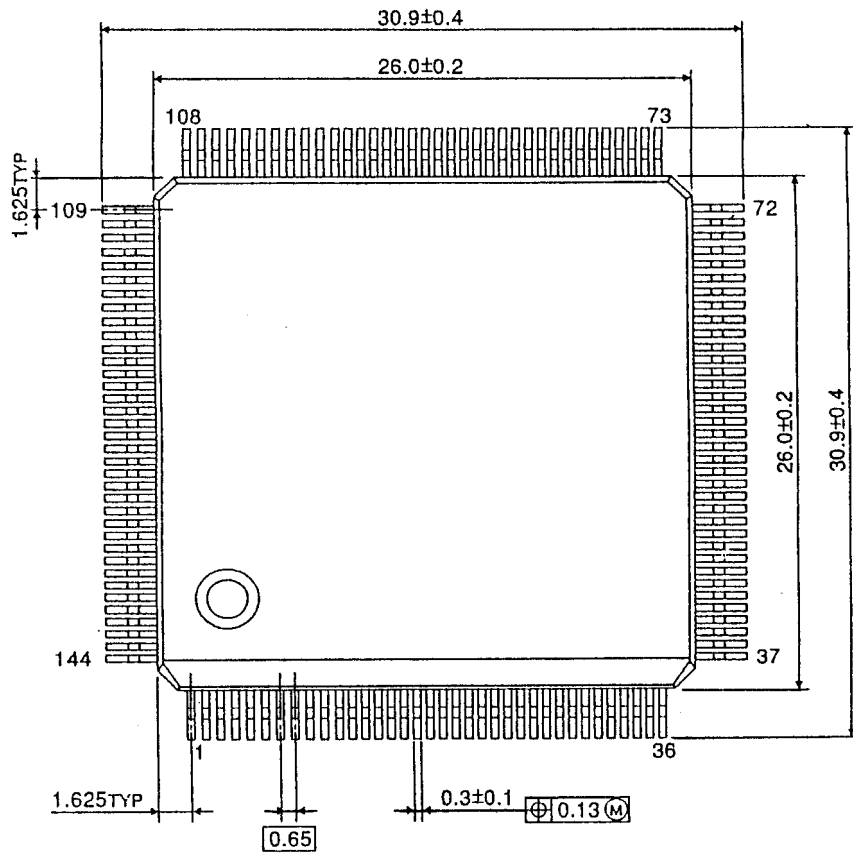
这是减速停止输入信号和减速停止命令的动作时序。减速停止输入信号是 nIN3~0、nLMTP/M（设定为减速停止模式时）。

当减速停止输入信号有效或者减速停止命令被写入时，在完成当前驱动脉冲输出后，就开始减速。



14 外形尺寸

單位：mm



15 规格

- 控制轴 4 轴
- CPU 数据总线长度 可选 8 位/16 位

插补功能

- 2 轴/3 轴直线插补
 - 插补范围 各个轴-8,388,607 ~ +8,388,607
 - 插补速度 1~4 MPPS
 - 插补位置精密度 ± 0.5 LSB 以下 (在全插补范围内)
- 圆弧插补
 - 插补范围 各个轴-8,388,607 ~ +8,388,607
 - 插补速度 1~4 MPPS
 - 插补位置精密度 ± 1 LSB 以下 (在全插补范围内)
- 2 轴/3 轴位模式插补
 - 插补速度 1~4 MPPS (但依靠 CPU 数据设定时间)
- 其他插补功能
 - 线速常数
 - 连续插补
 - 插补步进传送 (命令/外部信号)
 - 可任选轴

各轴共通规格

- 驱动输出脉冲 (CLK=16MHz 时)
 - 输出速度范围 1 PPS ~ 4 MPPS
 - 输出速度精密度 $\pm 0.1\%$ 以下 (对设定数值)
 - 速度倍率 1 ~ 500
 - S 曲线用 Jerk (加/减速度的增/减率)
 - 954 $\sim 62.5 \times 10^6$ PPS/SEC² (倍率=1 的时候)
 - 477×10^3 $\sim 31.25 \times 10^9$ PPS/SEC² (倍率=500 的时候)
 - 加/减速度
 - 125 $\sim 1 \times 10^6$ PPS/SEC (倍率=1 的时候)
 - $62.5 \times 10^3 \sim 500 \times 10^6$ PPS/SEC (倍率=500 的时候)
 - 初始速度
 - 1 $\sim 8,000$ PPS (倍率=1 的时候)
 - 500 PPS $\sim 4 \times 10^6$ PPS (倍率=500 的时候)
 - 驱动速度
 - 1 $\sim 8,000$ PPS (倍率=1 的时候)
 - 500 PPS $\sim 4 \times 10^6$ PPS (倍率=500 的时候)
 - 输出脉冲数 0 $\sim 268,435,455$ (定量驱动)
 - 速度曲线 定速/直线加减速/抛物线 S 曲线加减速驱动
 - 定量驱动的减速模式 自动减速/手动减速
 - 可以变更在驱动中的输出脉冲数、驱动速度
 - 可以选择独立 2 脉冲/1 脉冲·方向 方式
 - 可以选择脉冲逻辑电平
- 编码器输入脉冲
 - 可以选择 2 相脉冲/上下脉冲输入
 - 可以选择 2 相脉冲 1, 2, 4, 的平方
- 位置计数器
 - 逻辑位置计数器 (输出脉冲用) 计数范围 -2,147,483,648 ~ +2,147,483,647
 - 实位计数器 (输入脉冲用) 计数范围 -2,147,483,648 ~ +2,147,483,647可以一直写入、读出

- 比较寄存器
 - COMP+寄存器 位置比较范围 -2,147,483,648 ~ +2,147,483,647
 - COMP-寄存器 位置比较范围 -2,147,483,648 ~ +2,147,483,647
 - 把和位置计数器的大小状态输出, 信号输出
 - 做为软件限制可以动作
- 中断功能 (除插补外)
 - 中断发生因素 1 脉冲输出, 位置计数器 \geq COMP-变化时、位置计数器 $<$ COMP-变化时、位置计数器 $<$ COMP+变化时、位置计数器 \geq COMP+变化时、加减速驱动中的定速开始时、加减速驱动中的定速结束时、驱动结束时。
可以对任何因素选择有效/无效。
- 根据外部信号的驱动操作
 - 根据 EXPP、EXPM 信号, 可以运行+/-方向的定量/连续驱动
- 外部减速停止/立即停止信号
 - IN0 ~ 3 每一个轴 4 个
任何信号都可以选择有效/无效, 逻辑电平。可以做为通用输入使用。
- 伺服马达用输入信号
 - ALARM (警报), INPOS (定位完毕)
任何信号都可以选择有效/无效, 逻辑电平。
- 通用输出/入信号
 - OUT0 ~ 7 每一个轴 8 个 (和驱动状态输出信号共用端子)
- 驱动状态信号输出
 - DRIVE (驱动脉冲输出中), ASND (加速中), DSND (减速中), CMPP (位置 \geq COMP+), CMPM (位置 $<$ COMP-)。
驱动状态也可以在状态寄存器读出。
- 超越限制信号输入
 - +方向, -方向每一个
可选逻辑电平。激活时, 可选立即停止/减速停止。
- 紧急停止信号输入
 - 全轴只有 1 个 EMGN。在 Low 电平立即停止全轴的驱动脉冲。
- 电气的特性
 - 动作温度范围 0 ~ +85°C
 - 动作电源电压 +5V \pm 5% (消费电流 50mA max)
 - 输出/入信号电平 CMOS、TTL 可以连接
 - 输入时钟 16.000 MHz (标准)
- 封装 144 个脚 QFP, 0.65 间距
外形尺寸: 30.9 \times 30.9 \times 4.36 mm

附录 A：速度曲线图型

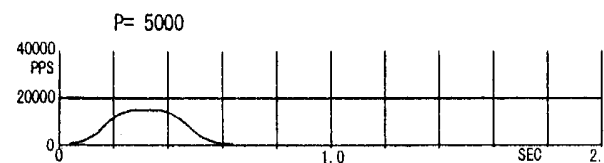
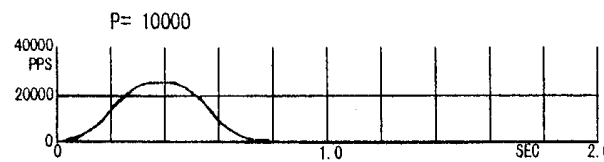
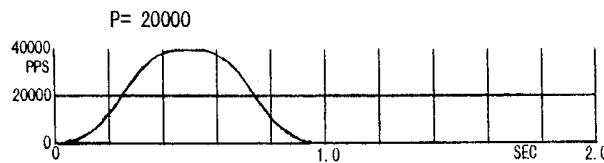
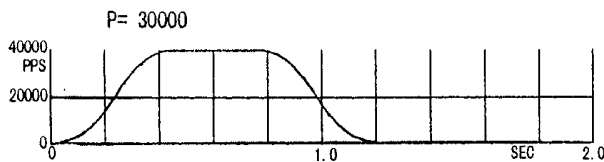
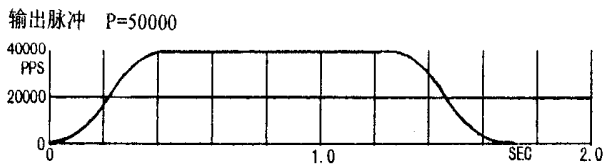
以下所示的速度曲线是观测带有速度跟踪器的 MCX314 输出驱动脉冲而得的图型。

加/减速时，完全 S 曲线加/减速是在达到目标速度前一直运行抛物线加速的 S 曲线加/减速（不包括加速度/减速度恒定域）。部分 S 曲线加/减速是包括加速度/减速度恒定域（加/减速区域是线性的）的 S 曲线加/减速。

■ 40KPPS 完全 S 曲线加/减速

R=800,000(倍率:10), K=700, (A=D=200), SV=100, V=4,000, A0=50
自动减速模式

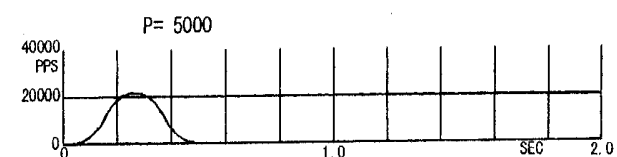
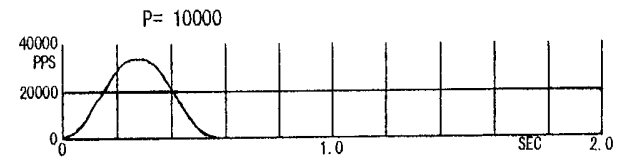
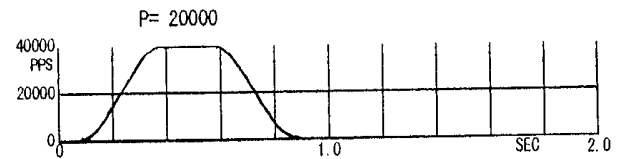
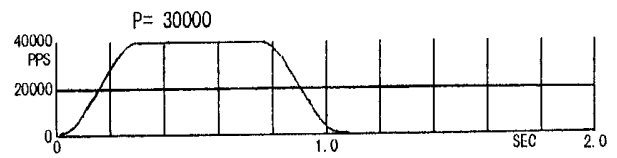
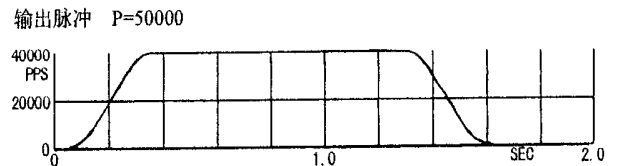
Jerk=893K PPS/SEC²
(加/减速度=250K PPS/SEC)
初始速度=1,000 PPS
驱动速度=40 PPS



■ 40KPPS 部分 S 曲线加/减速

R=800,000(倍率:10), K=300, A=D=150, SV=100, V=4,000, A0=20
自动减速模式

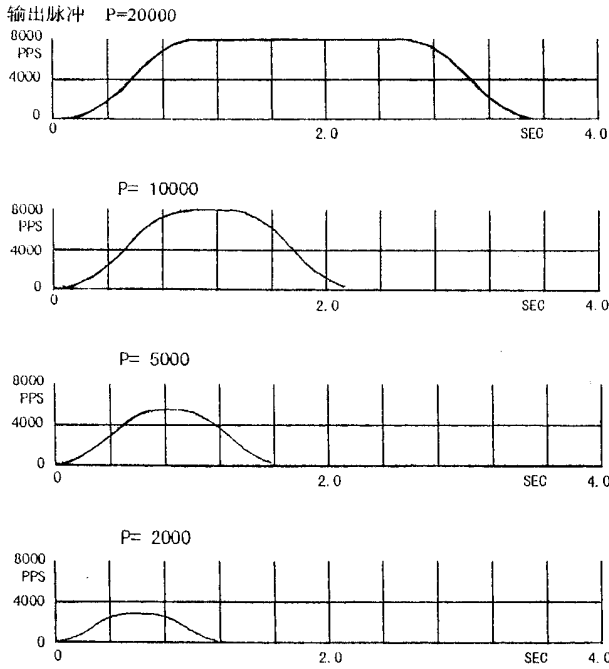
Jerk=2083K PPS/SEC²
加/减速度=188K PPS/SEC
初始速度=1,000 PPS
驱动速度=40K PPS



■ 8,000KPPS 完全 S 曲线加/减速

R=8,000,000(倍率:1), K=2,000, (A=D=500), SV=100, V=8,000, A0=0
自动减速模式

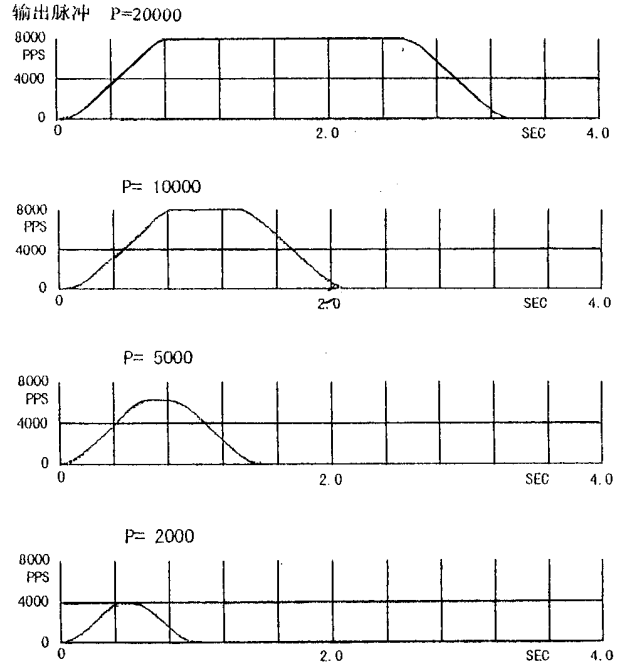
Jerk=31K PPS/SEC²
(加/减速度=62.5K PPS/SEC)
初始速度=100 PPS
驱动速度=8,000 PPS



■ 8,000KPPS 部分 S 曲线加/减速

R=8,000,000(倍率:1), K=1,000, A=D=100, SV=100, V=8,000, A0=0
自动减速模式

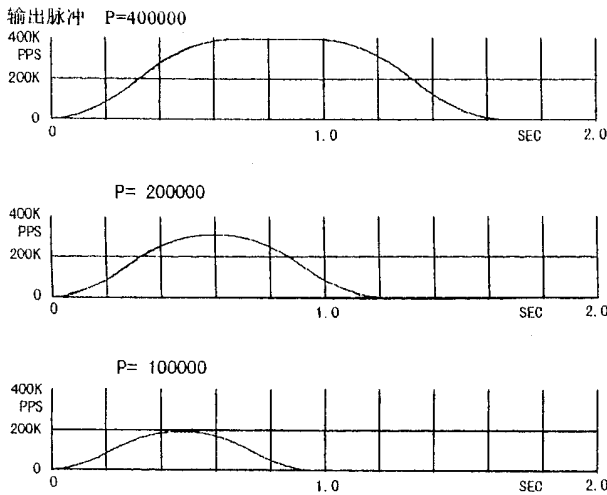
Jerk=62.5K PPS/SEC²
加/减速度=12.5K PPS/SEC
初始速度=100 PPS
驱动速度=8,000 PPS



■ 400KPPS 完全 S 曲线加/减速

R=80,000(倍率:100), K=2,000, (A=D=100), SV=10, V=4,000, A0=1,000
自动减速模式

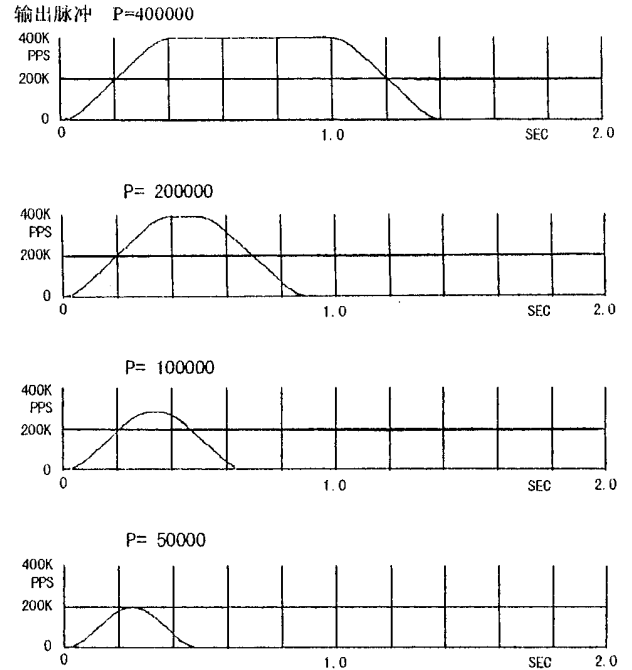
Jerk=3.13M PPS/SEC²
(加/减速度=12.5M PPS/SEC)
初始速度=1,000 PPS
驱动速度=400K PPS



■ 400KPPS 部分 S 曲线加/减速

R=80,000(倍率:100), K=500, A=D=100, SV=10, V=4,000, A0=1,000
自动减速模式

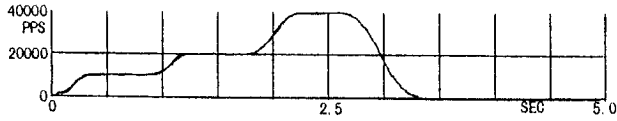
Jerk=12.5M PPS/SEC²
加/减速度=1.25M PPS/SEC
初始速度=1,000 PPS
驱动速度=400K PPS



■ S 曲线加/减速连续驱动的速度变更

R=800,000(倍率:10), K=2,000, (A=D=100), SV=100, V=1,000→2,000→4,000 减速停止

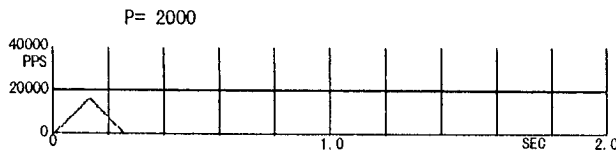
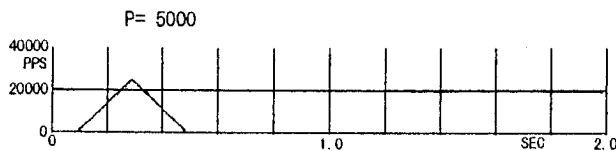
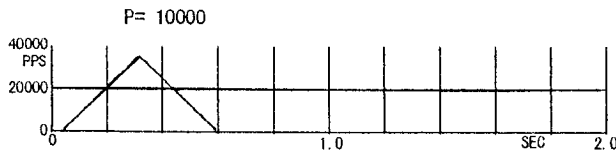
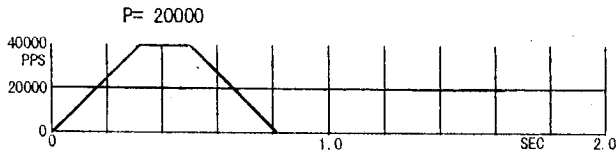
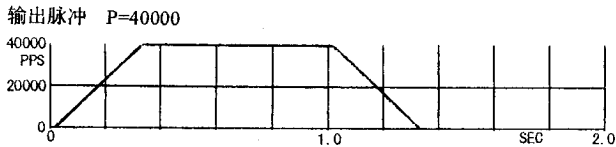
Jerk=312.5K PPS/SEC²
 加/减速度=125K PPS/SEC
 初始速度=1,000 PPS
 驱动速度=10K→20K→40K PPS



■ 40KPPS 直线加/减速

R=800,000(倍率:10), A=D=100, SV=100, V=4,000
 WR3/D2, D1, D0=0, 0, 0 直线加/减速、自动减速模式

加/减速度=125K PPS/SEC
 初始速度=1,000 PPS
 驱动速度=40,000 PPS

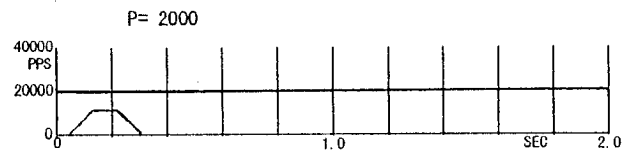
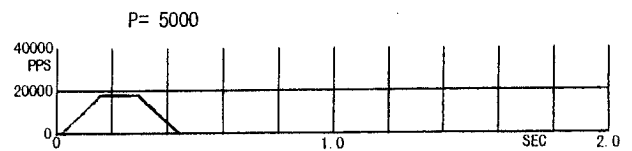
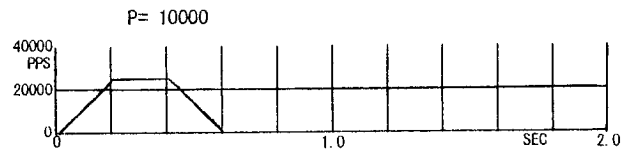
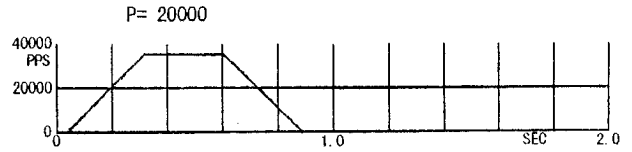
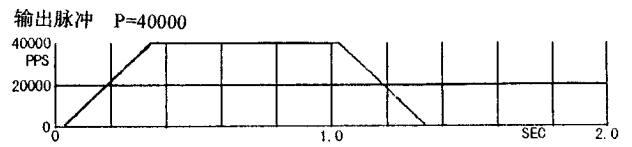


■ 直线加/减速的三角防止方法

当 K=1 (Jerk: 最高值) 并且处于 S 曲线加/减速模式时, 假如输出脉冲很少, 根据 1/4 法则, 就会产生三角波形

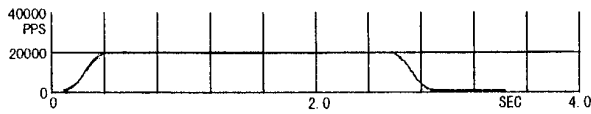
R=800,000(倍率:10), K=1, A=D=100, SV=100, V=4,000
 WR3/D2, D1, D0=1, 0, 0 S 曲线加/减速、自动减速模式

Jerk=625M PPS/SEC²
 加/减速度=125K PPS/SEC
 初始速度=1,000 PPS
 驱动速度=40,000 PPS

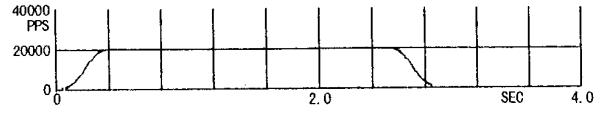


- 在定量脉冲驱动中通过 S 曲线加/减速后，仍落后初始速度，对此情况需调整参数。
变更点 K、V 等参数，就改善点。

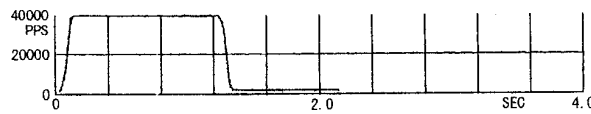
R=800,000(倍率:10), K=1,000, A=D=8,000, SV=100, V=2,000
自动减速模式 P=50,000



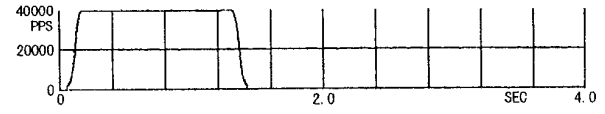
K=1010



R=800,000(倍率:10), K=50, A=D=8,000, SV=200, V=4,000, A0=0
自动减速模式 P=50,000

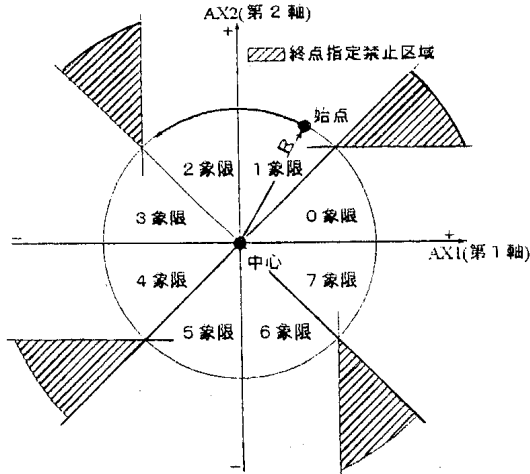


K=53

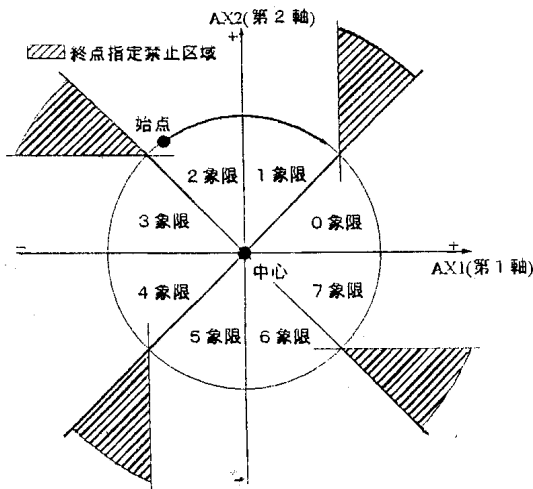


附录 B: 指定圆弧插补结束点应注意

CCW 圆弧插补



CW 圆弧插补



在 CW/CCW 圆弧插补时，根据中心点和已确定的半径 R 画圆弧，一般终点是在这个圆弧上。但是由于算术误差等原因使终点的指定脱离圆弧线，并指定在左图的斜线内的话，MCX314 不能检出终点，发生不停地在圆弧上运行的不正常现象。

有关圆弧插补的终结判定，请再参考 P15，把终点放在斜线内的话，在达到终点的短轴数值之前象限将变化，所以不能检出终点。

【避免方法】

如图所示，终点的位置在圆弧外而在 45° 象限切换附近时，很有可能发生这种情况。因此，要把终点放在圆弧上或者由于算术误差的原因不能放在圆弧上的话，可放在圆弧线的内侧。

比如 AX1（第一轴）设定为 X、AX2（第二轴）设定为 Y、中心点设定为 (CX, CY)、终点设定为 (EX, EY) 的话，圆弧的半径 R 为：

$$R = \sqrt{CX^2 + CY^2}$$

离圆弧中心至终点的距离 EL 为：

$$EL = \sqrt{(EX - CX)^2 + (EY - CY)^2}$$

EL > R 的时候，终点在圆弧外侧

EL = R 的时候，终点在圆弧线上

EL < R 的时候，终点在圆弧内侧

EL > R 的时候把 EX, EY 的数值增或减 1，变为 EL ≤ R 的状态。