

# 西门子S7-300授权总经销商 6ES7370-0AA01-0AA0 假体模块 DM 370

产品名称	西门子S7-300授权总经销商 6ES7370-0AA01-0AA0 假体模块 DM 370
公司名称	浔之漫智控技术(上海)有限公司
价格	.00/件
规格参数	西门子:现货 S7-300:正品 德国:全新
公司地址	上海市松江区石湖荡镇塔汇路755弄29号1幢一层A区213室
联系电话	15801997124 15801997124

## 产品详情

西门子S7-300授权总经销商 6ES7370-0AA01-0AA0 假体模块 DM 370

6ES7370-0AA01-0AA0

SIMATIC S7-300，假体模块 DM 370，占位模块  
更换模块时应用

算术指令组合累加器1和累加器2的内容。对于带有两个累加器的CPU，累加器2的内容保持不变。对于带4个累加器的CPU，将累加器3的内容复制到累加器2中，将累加器4的内容复制到累加器3中。累加器4的旧内容保持不变。使用整数运算，您可以对两个整数(16和32位)执行以下运算：+I ACCU 1 + ACCU 2，整型(16位) -I \*I /I ACCU 2 - ACCU 1，整型(16位) ACCU 1 \* ACCU 2，整型(16位) ACCU 2 / ACCU 1，整型(16位) + 整型常数相加(16、32位) +D ACCU 1 + ACCU 2，长整型(32位) -D ACCU 2 - ACCU 1，长整型(32位) \*D ACCU 1 \* ACCU 2，长整型(32位) /D ACCU 2 / ACCU 1，长整型(32位) MOD 除法余数，长整型(32位)

参见使用整数算术指令时得出状态字的位数值。S7-300 和S7-400 编程的语句表(STL) 参考手册, 05/2017, A5E41525031-AA 101 整型数学运算指令 7.2 使用整数算术指令时得出状态字的位数值 7.2

使用整数算术指令时得出状态字的位数值 描述

整数算术指令影响状态字中的下列位：CC1和CC0、OV和OS。

下表显示指令结果为整数(16位和32位)时状态字中各位的信号状态：结果的有效范围 CC 1 CC 0 OV OS 0 (零) 0 0 0 \* 16位：-32 768 <= 结果 < 0 (负数) 32位：-2 147 483 648 <=结果 < 0 (负数) 0 1 0 \* 16位：32 767 >= 结果 > 0 (正数) 32位：2 147 483 647 >= 结果 > 0 (正数) 1 0 0 \* \* 指令结果不影响OS位。结果的无效范围 A1 A0 OV OS 下溢(加法) 16位：结果 = -65536 32位：结果 = -4 294 967 296 0 0 1 1 下溢(乘法) 16位：结果 < -32 768 (负数) 32位：结果 < -2 147 483 648 (负数) 0 1 1 1 溢出(加法、减法) 16位：结果 > 32 767 (正数) 32位：结果 > 2 147 483 647 (正数) 0 1 1 1 溢出(乘法、除法) 16位：结果 > 32 767 (正数) 32位：结果 > 2 147

483 647 (正数) 1 0 1 1 下溢(加法、减法) 16位: 结果 < -32.768 (负数) 32位: 结果 < -2 147 483 648 (负数) 1 0 1 1 被0除 1 1 1 1 操作 A1 A0 OV OS +D: 结果 = -4 294 967 296 0 0 1 1 /D或MOD: 除以0 1 1 1 1

S7-300和S7-400编程的语句表(STL) 102 参考手册, 05/2017, A5E41525031-AA 整型数学运算指令 7.3 +I  
ACCU 1 + ACCU 2, 整型(16位) 7.3 +I ACCU 1 + ACCU 2, 整型(16位) 格式 +I 描述 +I  
(16位整数相加)将ACCU 1-L的内容与ACCU 2-L中的内容相加, 并将结果存储在ACCU 1-L中。将ACCU 1-L和ACCU 2-L的内容解释为16位整数。执行该指令时不涉及RLO, 也不会影响RLO。  
作为指令运算结果的一个功能, 将对状态字的位CC 1、CC 0、OS和OV进行设置。在发生溢出/下溢时, 该指令生成一个16位整数, 而不是一个32位整数。  
对于带2个累加器的CPU而言, 累加器2的结果保持不变。  
对于带4个累加器的CPU, 还将累加器3的内容复制到累加器2中, 将累加器4的内容复制到累加器3中。累加器4的内容保持不变。参见使用整数算术指令时得出状态字的位数值。状态字 BR CC 1 CC 0 OV OS OR STA RLO /FC 写: -xxxx- - - - 生成状态位 CC 1 CC 0 OV OS 总和 = 0 0 0 0 - -32768 <= 总和 < 0 0 1 0 - 32767 >= 总和 > 0 1 0 0 - 总和 = -65536 0 0 1 1 65534 >= 总和 > 32767 0 1 1 1 -65535 <= 总和 < -32768 1 0 1 1 实例 STL 解释 L IW10 //将IW10的数值装载到ACCU 1-L中。L MW14 //将ACCU 1-L的内容装载到ACCU 2-L中。//将MW14的数值装载到 //ACCU 1-L中。+I //ACCU 2-L和ACCU 1-L相加; 将结果存储在ACCU 1-L中。T DB1.DBW25 //将ACCU 1-L(结果)的内容传送到DB1的DBW25中。S7-300和S7-400编程的语句表(STL)  
参考手册, 05/2017, A5E41525031-AA 103 整型数学运算指令 7.4 -I ACCU 2 - ACCU 1, 整型(16位)  
7.4 -I ACCU 2 - ACCU 1, 整型(16位) 格式 -I 描述 I (16位整数相减)从ACCU 2-L的内容中减去ACCU 1-L的内容, 并将结果存储在ACCU 1-L中。将 ACCU 1-L和ACCU 2-L的内容解释为16位整数。执行该指令时不涉及RLO, 也不会影响RLO。作为指令运算结果的一个功能, 将对状态字的位CC 1、CC 0、OS和OV进行设置。在发生溢出/下溢时, 该指令生成一个16位整数, 而不是一个32位整数。  
对于带2个累加器的CPU而言, 累加器2的结果保持不变。  
对于带4个累加器的CPU, 还将累加器3的内容复制到累加器2中, 将累加器4的内容复制到累加器3中。累加器4的内容保持不变。参见使用整数算术指令时得出状态字的位数值。状态字 BR CC 1 CC 0 OV OS OR STA RLO /FC 写: -xxxx- - - - 生成状态位 CC 1 CC 0 OV OS 差 = 0 0 0 0 - -32768 <= 差 < 0 0 1 0 - 32767 >= 差 > 0 1 0 0 - 65535 >= 差 > 32767 0 1 1 1 -65535 <= 差 < -32768 1 0 1 1 实例 STL 解释 L IW10 //将IW10的数值装载到ACCU 1-L中。L MW14 //将ACCU 1-L的内容装载到ACCU 2-L中。将MW14的数值装载到 //ACCU 1-L中。-I //从ACCU 2-L中减去ACCU 1-L; 将结果存储在ACCU 1-L中。T DB1.DBW25 //将ACCU 1-L(结果)的内容传送到DB1的DBW25中。S7-300和S7-400编程的语句表(STL) 104 参考手册, 05/2017, A5E41525031-AA 整型数学运算指令 7.5 \*I  
ACCU 1 \* ACCU 2, 整型(16位) 7.5 \*I ACCU 1 \* ACCU 2, 整型(16位) 格式 \*I 描述  
\*I(乘以16位整数)ACCU 2-L的内容乘以ACCU 1-L的内容。将ACCU 1-L和ACCU 2-L的内容解释为16位整数。结果作为一个32位整数存储在ACCU 1中。当状态字的位为OV1 = 1和OS = 1时, 表示结果超出16位整数范围。  
执行该指令时不涉及RLO, 也不会影响RLO。作为指令运算结果的一个功能, 将对状态字的位CC 1、CC 0、OS和OV进行设置。对于带2个累加器的CPU而言, 累加器2的结果保持不变。  
对于带4个累加器的CPU, 还将累加器3的内容复制到累加器2中, 将累加器4的内容复制到累加器3中。参见使用整数算术指令时得出状态字的位数值。状态字 BR CC 1 CC 0 OV OS OR STA RLO /FC 写: -xxxx- - - - 生成状态位 CC 1 CC 0 OV OS 乘积 = 0 0 0 0 - -32768 <= 乘积 < 0 0 1 0 - 32767 >= 乘积 > 0 1 0 0 - 1073741824 >= 乘积 > 32767 1 0 1 1 -1073709056 <= 乘积 < -32768 0 1 1 1 实例 STL 解释 L IW10 //将IW10的数值装载到ACCU 1-L中。L MW14 //将ACCU 1-L的内容装载到ACCU 2-L中。//将MW14的内容载入ACCU 1-L。\*I //ACCU 2-L和ACCU 1-L相乘; 结果存储在ACCU 1中。T DB1.DBW25 //将ACCU 1(结果)的内容传送到DB1的DBW25中。S7-300和S7-400编程的语句表(STL)  
参考手册, 05/2017, A5E41525031-AA 105 整型数学运算指令 7.6 /I ACCU 2 / ACCU 1, 整型(16位)  
7.6 /I ACCU 2 / ACCU 1, 整型(16位) 格式 /I 描述 /I (16位整数相除)ACCU 2-L的内容除以ACCU 1-L的内容。将ACCU 1-L和ACCU 2-L的内容解释为16位整数。结果存储在ACCU 1中, 包含两个16位整数, 即商和余数。在ACCU 1-L中存储商, 在ACCU 1-H中存储余数。执行该指令时不涉及RLO, 也不会影响RLO。作为指令运算结果的一个功能, 将对状态字的位CC 1、CC 0、OS和OV进行设置。

对于带2个累加器的CPU而言，累加器2的结果保持不变。

对于带4个累加器的CPU，还将累加器3的内容复制到累加器2中，将累加器4的内容复制到累加器3中。累加器4的内容保持不变。参见使用整数算术指令时得出状态字的位数值。状态字 BR CC 1 CC 0 OV OS OR STA RLO /FC 写：-xxxx- - - - 生成状态位 CC 1 CC 0 OV OS 商 = 0000 - -32768 <= 商 < 0010 - 32767 >= 商 > 0100 - 商 = 32768 1011 被零除 1111 实例 STL 解释 L IW10

//将IW10的数值装载到ACCU 1-L中。L MW14 //将ACCU 1-L的内容装载到ACCU

2-L中。将MW14的数值装载到 //ACCU 1-L中。/I //ACCU 2-L除以ACCU 1-L; 将结果存储在ACCU 1中。ACCU 1-L：//商、ACCU 1-H: 余数 T MD20 //将ACCU 1(结果)的内容传送到MD20中。

S7-300和S7-400编程的语句表(STL) 106 参考手册, 05/2017, A5E41525031-AA 整型数学运算指令 7.6 /I

ACCU 2 / ACCU 1, 整型(16 位) 实例：13 除以 4 执行指令(IW10)前，ACCU 2-L的内容：

执行指令(MW14)前，ACCU 1-L的内容：指令 /I (ACCU 2-L / ACCU 1-L)：执行指令后，ACCU 1-L的内容(商): 执行指令后，ACCU 1-H的内容(余数)：S7-300 和S7-400 编程的语句表(STL)

参考手册, 05/2017, A5E41525031-AA "13" "4" "13/4" "3" "1" 107 整型数学运算指令 7.7 + + 整型常数(16位、32位) 7.7 + + 整型常数(16位、32位) 格式 + <整型常数> 地址 数据类型 描述

<整型常数> (16位或32位整数) 要相加的整数 描述 + <整型常数>将整型常数加到ACCU

1的内容中，并将结果存储在ACCU 1中。该指令的执行与状态

字的位无关，也不影响状态字的位。执行该指令时不涉及，也不会影响状态字的位。+

<16位整型常数>：将一个16位整型常数(范围为-32768至+32767)加到ACCU 1-L的内容中，然后将结果存储在ACCU 1-L中。+

<32位整型常数>：将一个32位整型常数(范围为-2,147,483,648至2,147,483,647)加到ACCU 1

的内容中，然后将结果存储在ACCU 1中。状态字 BR CC 1 CC 0 OV OS OR STA RLO /FC 写：- - - - -

- - - 实例1 STL 解释 L IW10 //将IW10的数值装载到ACCU 1-L中。L MW14 //将ACCU

1-L的内容装载到ACCU 2-L中。将MW14的数值装载到 //ACCU 1-L中。+I //ACCU 2-L和ACCU

1-L相加；将结果存储在ACCU 1-L中。+ 25 //ACCU 1-L与25相加；将结果存储在ACCU 1-L中。T

DB1.DBW25 //将ACCU 1-L(结果)的内容传送到DB1的DBW25中。S7-300和S7-400编程的语句表(STL) 108

参考手册, 05/2017, A5E41525031-AA 整型数学运算指令 7.7 + + 整型常数(16位、32位) 实例2 STL

解释 L IW12 L IW14 + 100 //ACCU 1-L与100相加；将结果存储在ACCU 1-L中。> I //当ACCU 2

> ACCU 1或IW12 > (IW14 + 100)时 JC NEXT //则条件跳转到跳转标签NEXT。实例3 STL 解释 L MD20

L MD24 +D //ACCU 1和ACCU 2相加；将结果存储在ACCU 1中。+I# -200 //ACCU

1与-200相加；将结果存储在ACCU 1中。T MD28 S7-300和S7-400编程的语句表(STL)

参考手册, 05/2017, A5E41525031-AA 109 整型数学运算指令 7.8 +D ACCU 1 + ACCU

2, 长整型(32位) 7.8 +D ACCU 1 + ACCU 2, 长整型(32位) 格式 +D 描述 +D

(32位整数相加)将ACCU 1的内容与ACCU 2中的内容相加，并将结果存储在ACCU 1中。将 ACCU

1的内容和ACCU 2的内容解释为32位整数。执行该指令时不涉及RLO，也不会影响RLO。

作为指令运算结果的一个功能，将对状态字的位CC 1、CC 0、OS和OV进行设置。

对于带2个累加器的CPU而言，累加器2的结果保持不变。

对于带4个累加器的CPU，还将累加器3的内容复制到累加器2中，将累加器4的内容复制到累加器3中。累加器4的内容保持不变。参见使用整数算术指令时得出状态字的位数值。状态字 BR CC 1

CC 0 OV OS OR STA RLO /FC 写：-xxxx- - - - 生成状态位 CC 1 CC 0 OV OS 总和 = 0000 - -2147483648 <= 总和 < 0010 - 2147483647 >= 总和 > 0100 - 总和 = -4294967296 0011 4294967294 >= 总和 >

2147483647 0111 -4294967295 <= 总和 < -2147483648 1011 实例 STL 解释 L ID10

//将ID10的值装载到ACCU 1中。L MD14 //将ACCU 1的内容装载到ACCU 2中。将MD14的值装载到

//ACCU 1中。+D //ACCU 2和ACCU 1相加；将结果存储在ACCU 1中。T DB1.DBD25 //将ACCU

1(结果)的内容传送到DB1的DBD25中。