

# 西门子S7-300授权总经销商 6ES7321-1BH50-0AA0 数字输入 SM 321

产品名称	西门子S7-300授权总经销商 6ES7321-1BH50-0AA0 数字输入 SM 321
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 6ES7321-1BH50-0AA0 数字输入 SM 321

6ES7321-1BH50-0AA0

SIMATIC S7-300，数字输入 SM 321，电位隔离  
16数字输入，24V DC，M 读取，1个 20针

本公司销售的一律为原装\*\*，假一罚十，可签正规的产品购销合同，可开增值税发票，税点另外算，24  
小时销售热线：15801997124

浔之漫智控技术(上海)有限公司

本公司专业经销合信/CO-TRUST科思创西门子PLC；S7-200S7-300 S7-400 S7-1200 触摸屏，变频器

6FC，6SNS120 V10 V60 V80伺服数控备件：原装进口电机，电线，电缆，希望能跟您有更多的合作机会2  
4小时销售热线：15801997124王工

我公司经营西门子全新原装现货PLC；S7-200S7-300 S7-400 S7-1200 触摸屏，变频器，

6FC，6SNS120 V10 V60 V80伺服数控备件：原装进口电机（1LA7、1LG4、1LA9、1LE1），国产电机（1L  
G0，1LE0）大型电机（1LA8，1LA4，1PQ8）

伺服电机 ( 1PH , 1PM , 1FT , 1FK , 1FS ) 西门子保内全新原装产品 ‘ 质保一年。

一年内因产品质量问题免费更换新产品 ; 不收取任何费。 欢迎致电咨询。

XOD ( 双字异或运算 ) 根据布尔逻辑 XOR ( 异或 ) 运算 , 将 ACCU 1 的内容与 ACCU 2 的内容或与 32 位常数逐位组合。 当在逻辑运算中组合的两个双字的相应位中仅有一位是 "1" 时 , 结果双字中的位才为 "1"。 该结果存储在 ACCU 1 中。 ACCU 2 保持不变。 状态位 CC 1 被置为运算的结果 ( 如结果不等于零 , 则 CC 1 = 1 )。 复位状态字的位 CC 0 和 OV 为 0。

可多次使用异或运算函数。 如果所选中地址有奇数个 "1" , 则逻辑运算结果为 "1"。 XOD : 将 ACCU 1 与 ACCU 2 组合。 XODTAK ( 将 ACCU 1 与 ACCU 2 互换 ) 将把 ACCU 1 的内容与 ACCU 2 的内容交换。 执行该指令时不考虑状态位 , 也不会影响状态位。 对具有四个 ACCU 的 CPU , ACCU 3 和 ACCU 4 的内容保持不变。 状态字 BR CC 1 CC 0 OV OS OR STA RLO /FC 写 : -----

实例 : 从较大值中减去较小值 STL 解释 L MW10 // 将 MW10 的内容载入 ACCU 1-L。 L MW12 // 将 ACCU 1-L 的内容装载到 ACCU 2-L 中。 将 MW12 // 的内容载入 ACCU 1-L。 > I // 检查 ACCU 2-L ( MW10 ) 是否大于 ACCU 1-L ( MW12 )。 SPB NEXT // 如果 ACCU 2 ( MW10 ) 大于 ACCU 1 ( MW12 ) , 则跳转到 NEXT 跳转标签。 TAK // 将 ACCU 1 的内容与 ACCU 2 的内容交换 NEXT: - I

// 从 ACCU 1-L 的内容中减去 ACCU 2-L 的内容。 T MW14 // 将结果 (= 较大值减较小值) 传送到 MW14。

目录 ACCU 1 ACCU 2 执行 TAK 指令之前执行 TAK 指令之后 S7-300 和 S7-400 编程的语句表 (STL) 236

参考手册, 05/2017, A5E41525031-AA 累加器指令 14.3 POP 具有两个 ACCU 的 CPU 14.3 POP

具有两个 ACCU 的 CPU 格式 POP 描述 POP ( 具有两个 ACCU 的 CPU ) 将 ACCU

2 的全部内容复制到 ACCU 1。 ACCU 2 保持不变。 执行该指令时不考虑状态位 , 也不会影响状态位。

状态字 BR CC 1 CC 0 OV OS OR STA RLO /FC 写 : ----- 实例 STL 解释 T MD10 // 将 ACCU

1 的内容 (= 值 A) 传送到 MD10 POP // 将 ACCU 2 的整个内容复制到 ACCU 1 T MD14 // 将 ACCU 1 的内容 (= 值 B) 传送到 MD14 目录 ACCU 1 ACCU 2 执行 POP 指令之前 值 A 值 B 执行 POP 指令之后 值 B 值 B

S7-300 和 S7-400 编程的语句表 (STL) 参考手册, 05/2017, A5E41525031-AA 237 累加器指令 14.4 POP

具有四个 ACCU 的 CPU 14.4 POP 具有四个 ACCU 的 CPU 格式 POP 描述 POP

( 具有四个 ACCU 的 CPU ) 将 ACCU 2 的全部内容复制到 ACCU 1 , 将 ACCU 3 的内容复制到 ACCU

2 , 并将 ACCU 4 的内容复制到 ACCU 3。 ACCU 4 保持不变。 执行该指令时不考虑状态位 ,

也不会影响状态位。 状态字 BR CC 1 CC 0 OV OS OR STA RLO /FC 写 : ----- 实例 STL 解释

T MD10 // 将 ACCU 1 的内容 (= 值 A) 传送到 MD10 POP // 将 ACCU 2 的整个内容复制到 ACCU 1 T MD14

// 将 ACCU 1 的内容 (= 值 B) 传送到 MD14 目录 ACCU 1 ACCU 2 ACCU 3 ACCU 4 执行 POP 指令之前 值 A 值 B

值 C 值 D 执行 POP 指令之后 值 B 值 C 值 D 值 D S7-300 和 S7-400 编程的语句表 (STL) 238 参考手册, 05/2017,

A5E41525031-AA 累加器指令 14.5 PUSH 具有两个 ACCU 的 CPU 14.5 PUSH

具有两个 ACCU 的 CPU 格式 PUSH 描述 PUSH ( ACCU 1 到 ACCU 2 ) 将 ACCU 1 的整个内容复制到 ACCU

2。 ACCU 1 保持不变。 执行该指令时不考虑状态位 , 也不会影响状态位。 状态字 BR CC 1 CC 0 OV

OS OR STA RLO /FC 写 : ----- 实例 STL 解释 L MW10 // 将 MW10 的内容载入 ACCU 1。 PUSH

// 将 ACCU 1 的整个内容复制到 ACCU 2 目录 ACCU 1 ACCU 2

执行 PUSH 指令之前执行 PUSH 指令之后 S7-300 和 S7-400 编程的语句表 (STL)

参考手册, 05/2017, A5E41525031-AA 239 累加器指令 14.6 PUSH 具有四个 ACCU 的 CPU 14.6 PUSH

具有四个 ACCU 的 CPU 格式 PUSH 描述 PUSH ( 具有四个 ACCU 的 CPU ) 将 ACCU 3 的内容复制到 ACCU

4 , 将 ACCU 2 的内容复制到 ACCU 3 , 并将 ACCU 1 的内容复制到 ACCU 2。 ACCU

1 保持不变。 执行该指令时不考虑状态位 , 也不会影响状态位。 状态字 BR CC 1 CC 0 OV OS OR STA

RLO /FC 写 : ----- 实例 STL 解释 L MW10 // 将 MW10 的内容载入 ACCU 1。 PUSH

// 将 ACCU 1 的整个内容复制到 ACCU 2 , 将 ACCU 2 的内容复制到 ACCU 3 , // 并将 ACCU

3 的内容复制到 ACCU 4。 目录 ACCU 1 ACCU 2 ACCU 3 ACCU 4 执行 PUSH 指令之前 值 A 值 B 值 C 值 D

执行 PUSH 指令之后 值 A 值 A 值 B 值 C S7-300 和 S7-400 编程的语句表 (STL) 240 参考手册, 05/2017,

A5E41525031-AA 累加器指令 14.7 ENT 进入 ACCU 堆栈 14.7 ENT 进入 ACCU 堆栈 格式 ENT 描述

ENT ( 进入累加器堆栈 ) 将把 ACCU 3 的内容复制到 ACCU 4 , 并将 ACCU 2 的内容复制到 ACCU 3。

如果在装载指令前面直接编程 ENT 指令 , 则可将中间结果保存到 ACCU 3 中。 实例 STL 解释 L DBD0

// 从数据双字 DBD0 中将值载入 ACCU 1。 //( 该值必须为浮点数格式。 ) L DBD4 // 将值从 ACCU

1复制到ACCU 2。 //从数据双字DBD4中将值载入ACCU 1。(该值必须为浮点数格式)。 +R //将ACCU 1和ACCU 2的内容作为浮点数(32位, IEEE 754)相加, 并将结果保存到ACCU 1中。 L DBD8 //将值从ACCU 1复制到ACCU 2, 并从数据双字DBD8中将值载入ACCU 1。 ENT //将ACCU 3的内容复制到ACCU 4。 将ACCU 2的内容(中间结果) //复制到ACCU 3。 L DBD12 //从数据双字DBD12中将值载入ACCU 1。 - R //从ACCU 2的内容中减去ACCU 1的内容, 并将结果保存在ACCU 1中。 //将ACCU 3的内容复制到ACCU 2。 //将ACCU 4的内容复制到ACCU 3。 /R //将ACCU 2 (DBD0 + DBD4)的内容除以ACCU 1 (DBD8 - DBD12)的内容。 //将结果保存在ACCU 1中。 T DBD16 //将结果(ACCU 1)传送到数据双字DBD16。