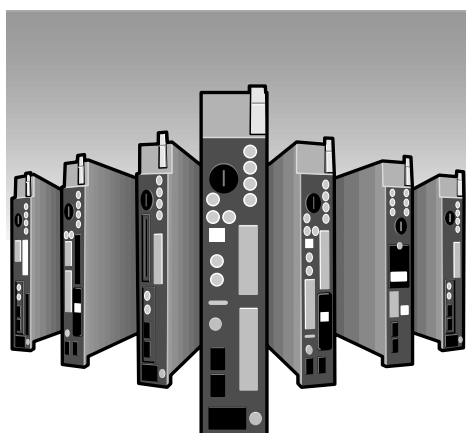




Allen-Bradley

***Controladores
programables
PLC-5***

Referencia del conjunto de instrucciones



Información importante para el usuario

El equipo de estado sólido tiene características de operación diferentes a las del equipo electromecánico. La publicación "Safety Guidelines for the Application, Installation and Maintenance of Solid State Controls" (publicación SGI-1.1) describe algunas diferencias importantes entre equipos de estado sólido y dispositivos electromecánicos cableados. Debido a estas diferencias y debido también a la amplia variedad de usos para los equipos de estado sólido, todas las personas responsables de la aplicación de este equipo deben asegurarse de que cada aplicación sea la correcta.

En ningún caso será Allen-Bradley Company responsable de daños indirectos o como consecuencia del uso o aplicación de este equipo.

Los ejemplos y diagramas mostrados en este manual tienen la única intención de ilustrar el texto. Debido a las muchas variables y requisitos asociados con cualquier instalación particular, Allen-Bradley Company no puede asumir responsabilidad u obligación por el uso real basado en los ejemplos y diagramas mostrados.

Allen-Bradley Company no asume responsabilidad por violación de patente alguna, con respecto al uso de información, circuitos, equipos o programas de software descritos en este manual.

Está prohibida la reproducción total o parcial del contenido de este manual sin el permiso escrito de Allen-Bradley Company.

En este manual hacemos anotaciones para advertirle sobre consideraciones de seguridad:



ATENCIÓN: Identifica información o prácticas o circunstancias que pueden conducir a lesiones personales o la muerte, daños materiales o pérdidas económicas.

Las notas de "Atención" le ayudan a :

- identificar un peligro
- evitar un peligro
- reconocer las consecuencias

Importante: Identifica información crítica para una correcta aplicación y entendimiento del producto.

Sírvase tomar nota de que en esta publicación se usa el punto decimal para separar la parte entera de la decimal de todos los números

Ethernet es una marca registrada de Intel Corporation, Xerox Corporation y Digital Equipment Corporation.

Data Highway Plus, DH+, PLC, PLC-5, PLC-5/11, -5/20, -5/26, -5/30, -5/40, -5/46, -5/40L, -5/60, -5/60L, -5/80, -5/86, -5/20E, -5/40E y -5/80E son marcas comerciales de Rockwell Automation.

Allen-Bradley es una marca comercial de Rockwell Automation, una entidad principal de Rockwell International Corporation.

Lista alfabética del conjunto de instrucciones PLC-5

Para esta instrucción	Vea la página	Para esta instrucción	Vea la página	Para esta instrucción	Vea la página	Para esta instrucción	Vea la página
ABL	17-5 ¹	CMP	3-3	JSR	13-12	RES	2-25
ACB	17-7 ¹	COP	9-20	LBL	13-5	RET	13-12
ACI	17-9 ¹	COS	4-21 ¹	LEQ	3-9	RTO	2-13
ACN	17-10 ¹	CPT	4-5	LES	3-10	SBR	13-12
ACS	4-13 ¹	CTD	2-20	LFL	11-5 ¹	SDS	18-2
ADD	4-14	CTU	2-18	LFU	11-5 ¹	SFR	13-23 ¹
AEX	17-11 ¹	DDT	10-2	LIM	3-11	SIN	4-27 ¹
AFI	13-19	DEG	6-5 ¹	LN	4-23 ¹	SQI	12-2
AHL	17-12 ¹	DFA	18-3	LOG	4-24 ¹	SQL	12-2
AIC	17-14 ¹	DIV	4-22	MCR	13-3	SQO	12-2
AND	5-2	DTR	10-8	MEQ	3-13	SQR	4-28
ARD	17-15 ¹	EOT	13-24	MOV	7-4	SRT	4-29 ¹
ARL	17-18 ¹	EQU	3-6	MSG	16-2	STD	4-31 ¹
ASC	17-21 ¹	FAL	9-2	MUL	4-25	SUB	4-34
ASN	4-15 ¹	FBC	10-2	MVM	7-5	TAN	4-35 ¹
ASR	17-22 ¹	FFL	11-5	NEG	4-26	TND	13-19
ATN	4-16 ¹	FFU	11-5	NEQ	3-15	TOD	6-3
AVE	4-17 ¹	FLL	9-21	NOT	5-4	TOF	2-9
AWA	17-23 ¹	FOR	13-8	NXT	13-8	TON	2-5
AWT	17-26 ¹	FRD	6-4	ONS	13-20	UID	13-25 ¹
BRK	13-8	FSC	9-15	OR	5-6	UIE	13-26 ¹
BSL	11-2	GEQ	3-7	OSF	13-22 ¹	XIC	1-3
BSR	11-2	GRT	3-8	OSR	13-21 ¹	XIO	1-4
BTB	7-2	IDI	1-10 ²	OTE	1-5	XOR	5-8
BTR	15-4	IDO	1-11 ²	OTL	1-6	XPY	4-36 ¹
BTW	15-4	IIN	1-8	OTU	1-7		
CIO	15-25 ²	IOT	1-9	PID	NO TAG		
CLR	4-20	JMP	13-5	RAD	6-6 ¹		

¹ Procesadores PLC-5 con características mejoradas solamente

² Software de programación 6200 con procesadores PLC-5 solamente

Vea la Tabla A para obtener pautas acerca de cómo seleccionar la instrucción apropiada para la operación que desea realizar. La Tabla B indica algunos ejemplos.

Tabla A
Selección de una categoría de instrucción

Si desea realizar esta operación:	Use esta categoría de instrucción:	
examen, verificación o control	dispositivo o condición de 2 estados múltiples dispositivos o condiciones de 2 estados	nivel de bit varios bits
mover, copiar, cambiar, calcular, comparar	valores analógicos, códigos varios conjuntos de valores	nivel de elemento instrucciones de archivo
convertir	instrucciones de conversión	
tiempo o retardo	temporizador	
conteo	contador	
desplazamiento o seguimiento	desplazamiento de bit	
secuencia	secuenciador	
PID	PID	
envío/recepción de mensajes	mensaje	
transferencia de datos hacia/desde módulos	transferencia en bloques o transferencia ControlNet	
manejo de diagnósticos y fallos	diagnósticos	
control de flujo del programa	control de programa	

Tabla B
Ejemplos de operaciones

Si la aplicación requiere las operaciones siguientes:	Use:
detección cuándo se cierra un interruptor de final de carrera	nivel de bit
cambio del valor preseleccionado de la temperatura	nivel de elemento
transferencia de datos analógicos	transferencia en bloques
arranque de un motor 10 segundos después de activarse una bomba	temporización
movimiento de 1 de 3 recetas en una zona de trabajo	varios elementos
seguimiento de piezas cuando se mueven de una estación a otra	desplazamiento
seguimiento del total de piezas en un recipiente	conteo

Resumen de los cambios

Nueva información que aparece en este manual

La lista siguiente resume los cambios efectuados en este manual desde la impresión más reciente:

Para obtener la siguiente información actualizada:	Vea el capítulo
Cómo convertir los números no decimales con la instrucción FRD	6
Cómo las direcciones indirectas no existentes afectan las instrucciones COP y FLL	9
Cómo el valor .POS funciona en las instrucciones de secuenciador	12
Cómo usar una instrucción RET	13
Cómo usar el término de polarización PID	14
Cómo usar las características “paso por cero” (.NOZC) y “calcular sin datos originales” (.NOBC) en el bloque de control PD	14
Explicación del código de error 89 para la instrucción MSG	16
Los procesadores son actualmente compatibles con las instrucciones MSG SLC Typed Read y SLC Typed Write	16
Cómo configurar una instrucción MSG de saltos múltiples mediante Ethernet o ControlNet	16
Cómo monitorear el estado del bit .EN en una instrucción MSG continua	16

Resumen de los cambios

Notas:

Prefacio

Convenciones

Este manual usa las convenciones siguientes:

- Si no se indica lo contrario:

Referencias acerca de:	Incluyen estos procesadores de Allen-Bradley:
Procesadores PLC-5 clásicos	Procesadores PLC-5/10™, -5/12™, -5/15™, -5/25™ y -5/VME™.
Procesadores PLC-5 con características mejoradas	Procesadores PLC-5/11™, -5/20™, -5/30™, -5/40™, -5/40L™, -5/60™, -5/60L™ y -5/80™. Nota: si no se indica lo contrario, los procesadores PLC-5 con características mejoradas incluyen los procesadores PLC-5 Ethernet, PLC-5 ControlNet, PLC-5 protegidos y PLC-5 VME.
Procesadores PLC-5 Ethernet	Procesadores PLC-5/20E™, -5/40E™ y -5/80E™.
Procesadores PLC-5 ControlNet	Procesadores PLC-5/20C™, -5/40C™, -5/46C™ y -5/80C™.
Procesadores PLC-5 protegidos ¹	Procesadores LC-5/26™, -5/46™ y -5/86™.
Procesadores PLC-5 VME	Procesadores PLC-5/V30™, -5/V40™, -5/V40L™ y -5/V80™. Vea el Manual del usuario de los controladores programables PLC-5/VME VMEbus para obtener más información.

¹ Los procesadores PLC-5 protegidos por sí solos no aseguran la seguridad del sistema PLC-5. La seguridad del sistema es una combinación del procesador PLC-5 protegido, el software y su experiencia en la aplicación.

- Las palabras que aparecen entre corchetes representan las teclas que se presionan. Por ejemplo:
[Enter]; [F1] – Online Programming/Documentation
- Las palabras que describen información que usted tiene que proporcionar se muestran en caracteres itálicos. Por ejemplo, si usted tiene que escribir el nombre de un archivo, aparece así:
filename
- Los mensajes e instrucciones que se muestran en la pantalla aparecen así:
Press a function key

Prefacio

Notas:

Instrucciones tipo relé**XIC, XIO, OTE, OTL, OTU, IIN, IOT,
IDI, IDO****Capítulo 1**

Cómo usar las instrucciones tipo relé	1-1
Archivos de imagen de E/S en el almacenamiento de datos	1-2
Lógica de renglón	1-2
Examine On (XIC)	1-3
Examine Off (XIO)	1-3
Energize (OTE)	1-4
Latch (OTL)	1-4
Unlatch (OTU)	1-5
Immediate Input (IIN)	1-6
Immediate Output (IOT)	1-7
Immediate Data Input (IDI)	1-8
Immediate Data Output (IDO)	1-8
Cómo usar las instrucciones IDI e IDO	1-9

Instrucciones de temporizador**TON, TOF, RTO****Instrucciones de contador****CTU, CTD****Restablecimiento RES****Capítulo 2**

Cómo usar los temporizadores y contadores	2-1
Cómo usar los temporizadores	2-1
Introducción de parámetros	2-2
Precisión del temporizador	2-3
Timer On Delay (TON)	2-4
Cómo usar los bits de estado	2-4
Timer Off Delay (TOF)	2-7
Cómo usar los bits de estado	2-7
Retentive Timer On (RTO)	2-10
Cómo usar los bits de estado	2-10
Using Counters	2-13
Introducción de parámetros	2-13
Count Up (CTU)	2-15
Cómo usar los bits de estado	2-15
Count Down (CTD)	2-17
Cómo usar los bits de estado	2-17
Timer and Counter Reset (RES)	2-20

Instrucciones de comparación**CMP, EQU, GEQ, GRT, LEQ, LES, LIM,
MEQ, NEQ****Capítulo 3**

Cómo usar las instrucciones de comparación	3-1
Cómo usar los indicadores de estado aritméticos	3-2
Compare (CMP)	3-2
Cómo introducir una expresión CMP	3-2
Cómo determinar la longitud de una expresión	3-3
Equal to (EQU)	3-5
Greater than or Equal to (GEQ)	3-5
Greater than (GRT)	3-6
Less than or Equal to (LEQ)	3-6
Less than (LES)	3-7
Limit Test (LIM)	3-7
Introducción de parámetros	3-7
Mask Compare Equal to (MEQ)	3-9
Introducción de parámetros	3-9
Not Equal to (NEQ)	3-10

Instrucciones de cálculo**CPT, ACS, ADD, ASN, ATN, AVE,
CLR, COS, DIV, LN, LOG, MUL, NEG,
SIN, SRT, SQR, STD, SUB, TAN, XPY****Capítulo 4**

Cómo usar las instrucciones de cálculo	4-1
Cómo usar los indicadores de estado aritméticos	4-2
tipos de datos y la instrucción de cálculo	4-3
Cómo usar los tipos de datos de	
punto flotante (coma flotante)	4-4
Compute (CPT)	4-5
Cómo introducir la expresión CPT	4-5
Cómo determinar la longitud de una expresión	4-7
Cómo determinar el orden de operación	4-8
Ejemplos de expresiones	4-8
Cómo introducir el destino	4-9
Cómo usar las funciones CPT	4-9
Arc Cosine (ACS)	4-11
Addition (ADD)	4-12
Arc Sine (ASN)	4-13
Arc Tangent (ATN)	4-14
Average File (AVE)	4-15
Introducción de parámetros	4-15
Cómo usar los bits de estado	4-16
Clear (CLR)	4-17
Cosine (COS)	4-18
Divide (DIV)	4-19
Natural Log (LN)	4-20
Log to the Base 10 (LOG)	4-21
Multiply (MUL)	4-22
Negate (NEG)	4-23
Sine (SIN)	4-24
Square Root (SQR)	4-25

Sort File (SRT)	4-26
Introducción de parámetros	4-26
Cómo usar los bits de estado	4-27
Standard Deviation (STD)	4-28
Introducción de parámetros	4-29
Cómo usar los bits de estado	4-29
Subtract (SUB)	4-31
Tangent (TAN)	4-32
X to the Power of Y (XPY)	4-33

Instrucciones lógicas AND, NOT, OR, XOR

Capítulo 5

Cómo usar las instrucciones lógicas	5-1
Cómo usar los indicadores de estado aritméticos	5-1
AND Operation (AND)	5-2
NOT Operation (NOT)	5-3
OR Operation (OR)	5-4
Exclusive OR Operation (XOR)	5-5

Instrucciones de conversión FRD y TOD, DEG y RAD

Capítulo 6

Cómo usar las instrucciones de conversión	6-1
Cómo usar los indicadores de estado aritméticos	6-1
Convert to BCD (TOD)	6-2
Convert from BCD (FRD)	6-2
Degree (DEG) (procesadores PLC-5 con características mejoradas solamente)	6-3
Radian (RAD) (procesadores PLC-5 con características mejoradas solamente)	6-4

Instrucciones para modificar y mover bits BTD, MOV, MVM

Capítulo 7

Cómo usar las instrucciones para modificar y mover bits . . .	7-1
Bit Distribute (BTD)	7-2
Introducción de parámetros	7-2
Move (MOV)	7-3
Masked Move (MVM)	7-4
Introducción de parámetros	7-4

Conceptos de instrucciones de archivo

Capítulo 8

Conceptos de operación de archivo	8-1
Cómo introducir los parámetros	8-1
Cómo usar la estructura de control	8-2
Cómo manipular los datos de archivo	8-3
Cómo usar los modos de operación de bloque	8-5
Modo Todo	8-5
Modo Numérico	8-6
Modo Incremento	8-7
Caso especial - modo numérico con palabras por escán = 1	8-8

Instrucciones de archivo**FAL, FSC, COP, FLL****Capítulo 9**

Cómo usar las instrucciones de archivo.	9-1
File Arithmetic and Logic (FAL)	9-2
Cómo usar los bits de estado.	9-4
FAL Copy Operations.	9-5
FAL Arithmetic Operations.	9-7
Límites superior e inferior	9-7
FAL Logic Operations	9-12
FAL Convert Operations.	9-14
File Search and Compare (FSC)	9-14
Cómo usar los bits de estado.	9-15
FSC Search and Compare Operations	9-17
Operaciones de búsqueda y comparación FSC.	9-17
Operación de búsqueda de archivo	9-17
File Copy (COP)	9-19
Cómo introducir los parámetros.	9-19
File Fill (FLL)	9-20
Cómo introducir los parámetros.	9-20

instrucciones de diagnóstico**FBC, DDT, DTR****Capítulo 10**

Cómo usar las instrucciones de diagnóstico	10-1
File Bit Comparison (FBC) y Diagnostic Detect (DDT)	10-2
Cómo seleccionar el modo búsqueda	10-2
Una desigualdad a la vez.	10-2
Todas las desigualdades durante un escán	10-3
Cómo introducir los parámetros.	10-4
Cómo usar los bits de estado.	10-5
Data Transitional (DTR)	10-8
Cómo introducir los parámetros.	10-8

Instrucciones de registro de desplazamiento BSL, BSR, FFL, FFU, LFL, LFU**Capítulo 11**

Cómo aplicar los registros de desplazamiento.	11-1
Cómo usar las instrucciones de desplazamiento de bit	11-2
Cómo introducir los parámetros.	11-2
Cómo usar los bits de estado.	11-3
Cómo usar las instrucciones FIFO y LIFO	11-5
Cómo introducir los parámetros.	11-5
Cómo usar los bits de estado.	11-6

Instrucciones de secuenciador SQO, SQI, SQL**Capítulo 12**

Cómo aplicar los secuenciadores	12-1
Cómo usar las instrucciones de secuenciador	12-2
Cómo introducir los parámetros.	12-2
Cómo usar los bits de estado.	12-4
Cómo restablecer la posición de SQO	12-6
Cómo usar SQI sin SQO	12-7

Instrucciones de control de programas MCR, JMP, LBL, FOR, NXT, BRK, JSR, SBR, RET, TND, AFI, ONS, OSR, OSF, SFR, EOT, UIE, UID

Capítulo 13

Cómo seleccionar las instrucciones de flujo de programas .	13-1
Master Control Reset (MCR)	13-2
Jump (JMP) and Label (LBL)	13-3
Cómo usar JMP	13-4
Cómo usar LBL	13-4
For Next Loop (FOR, NXT), Break (BRK)	13-5
Cómo introducir los parámetros	13-6
Cómo usar FOR	13-6
Cómo usar BRK	13-7
Cómo usar NXT	13-7
Jump to Subroutine (JSR), Subroutine (SBR) y Return (RET)	13-8
Cómo pasar parámetros	13-8
Cómo introducir los parámetros	13-10
Cómo anidar los archivos de subrutina	13-10
Cómo usar JSR	13-11
Cómo usar SBR	13-11
Cómo usar RET	13-12
Temporary End (TND)	13-13
Always False (AFI)	13-13
One Shot (ONS)	13-14
One Shot Rising (OSR)	13-15
Cómo introducir los parámetros	13-15
One Shot Falling (OSF)	13-16
Cómo introducir los parámetros	13-16
Sequential Function Chart Reset (SFR)	13-17
Cómo introducir los parámetros	13-17
End of Transition (EOT)	13-18
User Interrupt Disable (UID)	13-19
User Interrupt Enable (UIE)	13-20

Instrucción de control de proceso PID

Capítulo 14

Cómo usar PID	14-1
Características PID	14-2
Cómo usar las ecuaciones PID	14-2
Conversión de constante de ganancia	14-3
Implementación de término integral	14-3
Término de derivada	14-4
Cómo establecer los rangos de entrada/salida	14-5
Cómo implementar el escalado a unidades de ingeniería - tipo de archivo de número entero	14-5
Cómo establecer la banda muerta	14-6
Cómo usar el paso por cero	14-6
Cómo usar la característica sin paso por cero	14-7

Cómo seleccionar el término de derivada	
(acciones en PV o error)	14-7
Cómo establecer las alarmas de salida	14-7
Cómo usar el límite de salida	14-7
Bloqueo de acción integral	14-8
Cómo usar una operación de modo manual	
(con transferencia sin perturbaciones.	14-8
Cómo establecer la salida	14-8
Prealimentación o polarización de salida	14-9
Cómo reanudar el último estado	14-9
Instrucción PID	14-10
Cómo usar la característica de no calcular en	
base a datos originales.	14-11
Bits de estado de operación	14-11
Bloque de números enteros.	14-11
Bloque PD	14-12
Cómo introducir los parámetros.	14-12
Cómo usar un tipo de archivo de datos de números	
enteros para el bloque de control	14-14
Cómo usar los valores del bloque de control	14-16
Cómo usar un tipo de archivo PD para el	
bloque de control	14-18
Cómo usar los valores del bloque de control	14-23
Consideraciones de programación.	14-25
Errores de tiempo de ejecución	14-25
Cómo transferir datos a la instrucción PID	14-25
Consideraciones del lazo.	14-26
Número de lazos PID	14-26
Tiempo de actualización del lazo	14-26
Cómo desactivar el escalado de las entradas	14-27
Ejemplos de PID	14-29
Ejemplos de bloques de números entero (N)	14-29
Archivo de programa principal.	14-29
Archivo de programa STI	14-30
Archivo de programa RTS	14-32
Ejemplos del bloque PD.	14-33
Archivo de programa principal.	14-33
Archivo de programa STI	14-34
Archivo de programa RTS	14-36
Simulación de lógica de escalera de una	
estación de control manual.	14-37
Lazos en cascada	14-38
Control de relación	14-38
Seguimiento de la variable de proceso	14-39
Teoría de PID	14-40

Instrucciones de transferencia en bloques BTR y BTW e instrucción de transferencia de E/S ControlNet CIO

Capítulo 15

Cómo usar las instrucciones de transferencia en bloques y transferencia de E/S ControlNet.	15-1
Cómo usar las instrucciones de transferencia en bloques	15-1
Block-Transfer Read (BTR) y Block-Transfer Write (BTW).	15-3
Cola de solicitud de transferencia en bloques	15-3
Cómo introducir los parámetros.	15-4
Cómo usar los bits de estado	15-6
Cómo usar el bloque de control	15-8
Conteo de palabras solicitado (.RLEN)	15-8
Número de palabras transmitido (.DLEN)	15-8
Número de archivo (.FILE)	15-9
Número de elemento (.ELEM)	15-9
Cómo seleccionar la operación continua	15-10
Cómo seleccionar la operación no continua.	15-12
Temporización de transferencias en bloques -	
Procesadores PLC-5 clásicos	15-13
Tiempo de ejecución de la instrucción	15-13
Tiempo de espera en la cola	15-13
Tiempo de transferencia	15-13
Temporización de transferencias en bloques -	
Procesadores PLC-5 con características mejoradas	15-14
Tiempo de ejecución de la instrucción	15-14
Tiempo de espera en la zona de espera.	15-14
Tiempo de transferencia	15-14
Ejemplos de programación	15-15
Ejemplo de transferencia en bloques	
alternante bidireccional	15-16
ejemplo de transferencia en bloques	
de repetición alternante bidireccional	15-17
Ejemplo de transferencia en bloques	
continua bidireccional	15-18
Ejemplo de transferencia en bloques	
no continua bidireccional	15-19
Ejemplo de transferencia en bloques	
de repetición direccional	15-19
Ejemplo de transferencia en bloques	
continua bidireccional	15-20
Ejemplo de búfer de datos de transferencia en bloques	15-21
Instrucción de transferencia de E/S ControlNet (CIO)	15-22
Dirección del bloque de control	15-22
Cómo usar la instrucción CIO	15-23
Cómo usar los bits de estado	15-24
Cómo usar el bloque de control CT	15-25

Instrucción de mensaje MSG

Capítulo 16

Cómo usar la instrucción de mensaje	16-1
Message (MSG)	16-1
Cómo introducir los parámetros	16-2
Dirección del bloque de control	16-2
Pantalla para la introducción de datos MSG	16-3
Cómo usar la instrucción de mensaje para las	
comunicaciones Ethernet	16-5
Cómo introducir los parámetros	16-5
Cómo usar la instrucción de mensaje para las	
comunicaciones del módulo de interface PLC-5 Ethernet	16-7
Cómo introducir los parámetros	16-7
Cómo configurar una instrucción MSG de	
multisaltos Ethernet	16-9
Cómo usar la instrucción de mensaje para las	
comunicaciones ControlNet	16-10
Dirección del bloque de control	16-10
Cómo configurar una instrucción MSG de	
multisaltos ControlNet	16-11
Cómo usar los bits de estado	16-12
Cómo usar el bloque de control	16-13
Código de error (.ERR)	16-13
Longitud solicitada (.RLEN)	16-13
Longitud transmitida(.DLEN)	16-13
Cómo introducir los parámetros	16-14
Comando de comunicación	16-14
Direcciones externas de la tabla de datos	16-15
Archivos de compatibilidad de PLC-2 a PLC-5	16-15
Cómo enviar los comandos SLC Typed Logical Read	
y SLC Typed Logical Write	16-16
Cómo monitorear una instrucción de mensaje	16-17
Cómo seleccionar la operación continua	16-18
Cómo seleccionar la operación no continua	16-19
Temporización MSG	16-20
Códigos de error	16-22

Instrucciones ASCII

**ABL, ACB, ACI, ACN, AEX, AIC, AHL,
ARD, ARL, ASC, ASR, AWA, AWT**

Capítulo 17

Cómo usar las instrucciones ASCII para los procesadores

PLC-5 con características mejoradas solamente	17-1
Cómo usar los bits de estado	17-2
Cómo usar el bloque de control	17-3
Longitud (.LEN)	17-3
Posición (.POS)	17-3
Cómo usar las cadenas	17-3
Test Buffer for Line (ABL)	17-4
Cómo introducir los parámetros	17-4
Number of Characters in Buffer (ACB)	17-5
Cómo introducir los parámetros	17-5
ASCII String to Integer (ACI)	17-6
ASCII String Concatenate (ACN)	17-7
ASCII String Extract (AEX)	17-7
Cómo introducir los parámetros	17-7
ASCII Set or Reset Handshake Lines (AHL)	17-8
Cómo introducir los parámetros	17-8
ASCII Integer to String (AIC)	17-9
ASCII Read Characters (ARD)	17-10
Cómo introducir los parámetros	17-10
ASCII Read Line (ARL)	17-12
Cómo introducir los parámetros	17-12
ASCII String Search (ASC)	17-14
Cómo introducir los parámetros	17-14
ASCII String Compare (ASR)	17-15
ASCII Write with Append (AWA)	17-15
Cómo introducir los parámetros	17-15
ASCII Write (AWT)	17-17
Cómo introducir los parámetros	17-17

**instrucciones de rutina de
aplicación personalizada SDS, DFA**

Capítulo 18

Objetivos del capítulo	18-1
Smart Directed Sequencer (SDS) - Descripción general . . .	18-2
Cómo programar la instrucción SDS	18-2
Diagnostic Fault Annunciator (DFA) - Descripción general . .	18-3
Cómo programar la instrucción DFA	18-3

Requisitos de temporización y memoria de la instrucción

Apéndice A-1

Requisitos de temporización y memoria de la instrucción . . .	A-1
Temporización para los procesadores PLC-5 con	
características mejoradas	A-2
Instrucciones de bit y palabra	A-2
instrucciones de archivo	A-5
Temporización para los procesadores PLC-5 clásicos	A-10
Instrucciones de bit y palabra	A-10
instrucciones de archivo	A-13
Constantes de programa	A-17
Elementos directos o indirectos - Procesadores PLC-5	
con características mejoradas.	A-17
Elementos directos o indirectos -	
Procesadores PLC-5 clásicos	A-18
Direcciones indirectas de bits o elementos -	
Procesadores PLC-5 clásicos	A-19
Otras consideraciones asociadas con la temporización -	
Procesadores PLC-5 clásicos	A-20

Referencia SFC

Apéndice B-1

Objetivos del apéndice	B-1
Información de estado SFC en el archivo de estado	
del procesador	B-1
Asignación de memoria.	B-3
Restricciones dinámicas - Procesadores PLC-5	
clásicos solamente	B-5
Secuencias de escán	B-7
Escán de pasos y transiciones.	B-7
Escán de bifurcaciones seleccionadas.	B-8
Escán de bifurcaciones simultáneas	B-9
Ejemplo de SFC y secuencia de escán.	B-11
Tiempo de ejecución - Procesadores PLC-5 clásicos.	B-12
Cómo usar los diagramas de secuencia para determinar	
el tiempo de ejecución	B-13
Cómo usar las ecuaciones para determinar el	
tiempo de ejecución	B-14

Tipos de datos válidos para los operandos de instrucciones

Apéndice C-1

Objetivos del apéndice	C-1
Operandos de instrucciones y tipos de datos válidos.	C-1

Cómo usar las instrucciones tipo relé

Instrucciones tipo relé XIC, XIO, OTE, OTL, OTU, IIN, IOT, IDI, IDO

Use las instrucciones tipo relé para monitorear y controlar el estado de bits en la tabla de datos, tales como los bits de entrada o los bits de la palabra de control del temporizador. Las instrucciones de relé le permiten:

Si desea:	Use esta instrucción:	Que se encuentra en la página:
Examinar un bit para ver si una condición está ACTIVADA	XIC	1-3
Examinar un bit ara ver si una condición está DESACTIVADA	XIO	1-3
Retener un bit ACTIVADO o DESACTIVADO (no retentivo)	OTE	1-4
Enclavar un bit a ACTIVADO (retentivo)	OTL	1-4
Desenclavar un bit a DESACTIVADO (retentivo)	OTU	1-5
Actualizar inmediatamente los bits de imagen de entrada	IIN	1-6
Actualizar inmediatamente las salidas	IOT	1-7
Realizar inmediatamente una actualización del archivo de entrada de datos ControlNet™ desde los búferes de memoria ControlNet.	IDI	1-8
Realizar inmediatamente una actualización de los búferes de memoria ControlNet desde el archivo de origen antes de la próxima actualización de imagen de salida.	IDO	1-8

Estas instrucciones le permiten direccionar bits en todas las secciones del almacenamiento de datos, pero los ejemplos que aparecen en este capítulo indican solamente cómo direccionar los bits en los archivos de imagen de salida.

Vea el Apéndice C para obtener más información acerca de los operandos (y los tipos/valores de datos válidos de cada operando) usados por las instrucciones tratadas en este capítulo.

Si se usa una instrucción tipo relé (OTE, OTL u OTU) con direcciones indirectas para establecer o restablecer un bit en el archivo de control de una instrucción de transferencia en bloques o mensaje, es posible que se consigan resultados incompatibles. Aunque la instrucción de bit se ejecute para establecer o restablecer un bit, es posible que la operación de transferencia en bloques o mensaje que establece o restablece dicho bit pueda sobrescribir el resultado de la instrucción. Estas son operaciones asíncronas. La última operación que establece o restablece el bit es el valor que se guarda en la tabla de datos.

Archivos de imagen de E/S en el almacenamiento de datos

El archivo de imagen de entrada en el procesador almacena el estado de los detectores de entrada conectados a los terminales del módulo de entrada.

Si el detector de entrada está:	El bit de imagen de entrada correspondiente está:
cerrado (activado)	activado (1)
abierto (desactivado)	desactivado (0)

Las instrucciones se programan en la lógica de escalera para monitorear los bits. Use una dirección lógica para el bit.

El archivo de imagen de salida controla el estado de los accionadores cableados a los terminales del módulo de salida.

Si el bit de imagen de salida está:	La salida correspondiente está:
activado (1)	activada
desactivado (0)	desactivada

Las instrucciones se programan en la lógica de escalera para controlar los bits.

Lógica de renglón

Durante la ejecución de cada instrucción de acondicionamiento, el bit direccionado se examina para verificar su coincidencia con una condición determinada (activada o desactivada). Si se encuentra una ruta completa de condiciones verdaderas examinadas, el renglón se establece en verdadero. El renglón debe contener una ruta continua de instrucciones verdaderas desde el inicio del renglón hasta la salida para que la salida se habilite.

Examine On (XIC)

Ejemplo:

Si se encuentra una condición ACTIVADA en el bit I:012/07 en la tabla de datos, establezca la instrucción como verdadera

Este bit corresponde al terminal de entrada 7 de un módulo en el grupo de E/S 2 del rack de E/S 1. Si el circuito de entrada es verdadero, la instrucción es verdadera.

Descripción:

Cuando un dispositivo cierra su circuito, el módulo cuyo terminal está cableado al dispositivo detecta el circuito cerrado. El procesador refleja este estado ACTIVADO en la tabla de datos. Cuando el procesador encuentra una instrucción XIC que direcciona el bit correspondiente al terminal de entrada, el procesador determina si el dispositivo está ACTIVADO (cerrado). Si el procesador encuentra un estado ACTIVADO, establece la lógica de escalera como verdadera para esta instrucción. Si el procesador encuentra un estado DESACTIVADO, establece la lógica de escalera como no verdadera para dicha instrucción.

Si la instrucción XIC es la única instrucción de acondicionamiento en el renglón, el procesador habilita la instrucción de salida cuando la instrucción XIC es verdadera (entrada cerrada). El procesador inhabilita una instrucción de salida cuando la instrucción XIC es falsa (entrada abierta).

La instrucción Examen On es verdadera o falsa dependiendo de que el procesador encuentre una condición DESACTIVADA o ACTIVADA en el bit direccionado.

Si el bit está:	La instrucción es:	Estado lógico del bit:
activado	verdadera	1
desactivado	falsa	0

Examine Off (XIO)

Ejemplo:

Si se encuentra una condición DESACTIVADA en el bit I:012/07 en la tabla de datos, establezca la instrucción como verdadera

Este bit corresponde al terminal de entrada 7 de un módulo en el grupo de E/S 2 del rack de E/S 1. Si el circuito de entrada es falso, la instrucción es verdadera.

Descripción:

Cuando un dispositivo abre su circuito, el módulo cuyo terminal de entrada está cableado al dispositivo detecta un circuito abierto. El procesador refleja este estado DESACTIVADO en la tabla de datos. Cuando el procesador encuentra una instrucción XIO que direcciona el bit correspondiente al terminal de entrada, el procesador determina si el dispositivo está DESACTIVADO (abierto). Si el procesador encuentra un estado DESACTIVADO, establece la lógica de escalera como verdadera para esta instrucción. Si el procesador encuentra un estado ACTIVADO, establece la instrucción XIO como falsa.

Si la instrucción XIO es la única instrucción de acondicionamiento en el renglón, el procesador habilita la instrucción de salida cuando la instrucción XIO es verdadera (entrada abierta).

La instrucción Examine Off es verdadera o falsa dependiendo de que el procesador encuentre una condición DESACTIVADA o ACTIVADA en el bit direccionado.

Si el bit está:	La instrucción es:	Estado lógico del bit:
desactivado	verdadera	0
activado	falsa	1

Energize (OTE)

—————()|

Ejemplo:

0:013
—————()|
01

ACTIVE el bit 0:013/01 de la tabla de imagen de salida si el renglón es verdadero. DESACTIVE el bit si el renglón es falso.

Este bit corresponde al terminal de salida 01 de un módulo en el grupo de E/S 3 del rack de E/S 1.

Descripción:

Use la instrucción OTE para controlar un bit en la memoria. Si el bit corresponde a un terminal del módulo de salida, el dispositivo cableado a este terminal se activa cuando la instrucción se habilita y se desactiva cuando la instrucción se inhabilita. Si las condiciones de entrada que preceden la instrucción OTE son verdaderas, el procesador habilita la instrucción OTE. Si las condiciones de entrada que preceden la instrucción OTE son falsas, el procesador inhabilita la instrucción OTE. Cuando las condiciones de renglón se hacen falsas, el dispositivo correspondiente se desactiva.

Una instrucción OTE es similar a una bobina de relé. La instrucción OTE está controlada por las instrucciones de entrada precedentes. La bobina de relé está controlada por los contactos en el renglón cableado.

La instrucción OTE le indica al procesador que controle el bit direccionado según la condición del renglón:

Si el renglón es:	El procesador:	Estado lógico del bit:
verdadero	activa el bit	1
falso	desactiva el bit	0

Latch (OTL)

—————(L)|

Ejemplo:

0:013
—————(L)|
01

ACTIVE el bit 0:013/01 de la tabla de imagen de salida si el renglón es verdadero.

Este bit corresponde al terminal de salida 1 de un módulo en el grupo de E/S 3 del rack de E/S 1.

Descripción:

La instrucción OTL es una instrucción de salida retentiva que solamente activa un bit (no puede desactivar un bit). Esta instrucción normalmente se usa en parejas con una instrucción OTU (desenclavamiento) donde ambas instrucciones direccionan el mismo bit.

Cuando se asigna una dirección a una instrucción OTL que corresponde a un terminal de un módulo de salida, el dispositivo de salida conectado a dicho terminal se activa cuando el procesador establece (habilita) el bit en la memoria del procesador. Si las condiciones de entrada que preceden la instrucción son verdaderas, el procesador habilita la instrucción OTL. Cuando las condiciones del renglón se hacen falsas (después de ser verdaderas), el bit permanece establecido y el dispositivo de salida correspondiente permanece activado. Use la instrucción OTU para DESACTIVAR el bit que se enclavó con la instrucción OTL.

Una vez habilitada, la instrucción de enclavamiento le indica al procesador que active el bit direccionado. De ahí en adelante, el bit permanece activado independientemente de la condición del renglón hasta que se desactiva el bit, típicamente mediante una instrucción de desenclavamiento (OTU) en otro renglón.

Si el renglón es:	El procesador:
verdadero	activa el bit
falso	no cambia el bit

Cuando el procesador cambia del modo Marcha a Programación o cuando el procesador pierde energía (y hay una batería de respaldo), la última instrucción OTL verdadera continúa controlando el bit en memoria. El dispositivo de salida enclavado se activa aunque las condiciones de renglón que controlan la instrucción puedan haberse hecho falsas.

Importante: La instrucción OTL es retentiva. Cuando el procesador pierde energía, cambia al modo de Programación o Prueba o detecta un fallo mayor, las salidas se desactivan. Pero los estados de las salidas retentivas se retienen en memoria. Cuando el procesador reanuda la operación en el modo Marcha, las salidas retentivas inmediatamente retornan a sus estados anteriores. Las salidas no retentivas, tales como las salidas OTE, se restablecen.

Unlatch (OTU)

Descripción:

— (U) —

Ejemplo:

0:013
— (U) —
01

DEACTIVE el bit 0:013/01 de la tabla de imagen de salida si el renglón es verdadero.

Este bit corresponde al terminal de salida 1 de un módulo en el grupo de E/S 3 del rack de E/S 1.

La instrucción OTU es una instrucción de salida retentiva que solamente desactiva un bit (no puede activar un bit). Esta instrucción normalmente se usa en parejas con una instrucción OTL (enclavamiento de salida) donde ambas instrucciones direccionan el mismo bit. La instrucción DESACTIVA el bit que la instrucción OTL ACTIVO (enclavó).

Cuando el procesador cambia del modo Marcha a Programación o cuando el procesador pierde energía (y hay una batería de respaldo), el bit se retiene en el estado establecido por el último renglón de la pareja de enclavamiento/desenclavamiento que fue verdadero.

La instrucción de desenclavamiento le indica al procesador que desactive el bit direccionado según la condición del renglón. De ahí en adelante, el bit permanece desactivado independientemente de la condición del renglón hasta que se activa, típicamente mediante otra instrucción OTL en otro renglón.

Si el renglón es:	El procesador:
verdadero	activa el bit
falso	no cambia el bit

Immediate Input (IIN)

Descripción:

La instrucción IIN es una instrucción de salida que, cuando se habilita, actualiza una palabra de los bits de imagen de entrada antes de la próxima actualización normal de la imagen de entrada.

— (IIN) —

Ejemplo:

RRG
— (IIN) —

Donde:

RR = el número del rack de E/S

00-03 PLC-5/10, -5/11, -5/12, -5/15, -5/20

00-07 PLC-5/25, -5/30

000-177 PLC-5/40, -5/40L

000-277 PLC-5/60, -5/60L, -5/80

G = el número de grupo de E/S (0 - 7)

001
— (IIN) —

Cuando las condiciones de entrada sean verdaderas, actualice la palabra de imagen de entrada correspondiente al rack de E/S 0, grupo 1.

Para las entradas en el **chasis local**, el escán del programa se interrumpe mientras que las entradas del grupo de E/S direccionadas se examinan. Esto establece los bits de imagen de entrada en los estados actuales de las entradas antes de que continúe el escán de programa. Si el programa llega a una instrucción IIN habilitada a la vez que se está realizando una transferencia en bloques con el chasis local, el procesador completa la transferencia en bloques antes de ejecutar la instrucción IIN.

Para las entradas en un **chasis remoto**, el escán del programa se interrumpe solamente para actualizar la imagen de entrada con los estados más recientes de las entradas que se encuentran en el búfer de E/S remotas (desde el escán de E/S remotas más reciente). Las entradas no se escanean antes de que el escán del programa continúe.

Posicione el renglón con la instrucción IIN inmediatamente antes de los renglones que examinan los bits de entrada crítica actualizados por la instrucción IIN.

Para la instrucción IIN, es necesario introducir solamente el número del rack de E/S y el número de grupo de E/S. No es necesario introducir un número de archivo.



ATENCION: No introduzca una dirección que incluya un número de archivo tal como I:027. El procesador interpreta el arreglo de bits encontrado en dicha dirección como el rack de E/S y el número de grupo de E/S de las entradas que se deben actualizar. Esto resultará en la operación inesperada, daños al equipo y lesiones corporales eventuales.

Vea el capítulo 15 para obtener más información acerca del escán de E/S y transferencias en bloques.

Immediate Output (IOT)

———(IOT)———

Ejemplo:

RRG
———(IOT)———

Donde:

RR = el número del rack de E/S

00-03 PLC-5/10, -5/11, -5/12, -5/15, -5/20

00-07 PLC-5/25, -5/30

000-177 PLC-5/40, -5/40L

000-277 PLC-5/60, -5/60L, -5/80

G = el número de grupo de E/S (0 - 7)

001
———(IOT)———

Cuando las condiciones de entrada sean verdaderas, actualice la palabra de imagen de entrada correspondiente al rack de E/S 0, grupo 1.

Descripción:

La instrucción IOT es una instrucción de salida que, cuando se habilita, actualiza un grupo de E/S de salidas antes de la próxima actualización normal de la imagen de salida.

Para las salidas en el **chasis local**, el escán del programa se interrumpe mientras que las salidas del grupo de E/S direccionadas se examinan. Esto establece los circuitos de salida en los estados actuales de los bits de salida en la tabla de imagen de salida antes de que continúe el escán de programa. Si el programa llega a una instrucción IOT habilitada a la vez que se está realizando una transferencia en bloques con el chasis local, el procesador completa la transferencia en bloques antes de ejecutar la instrucción IOT.

Para las salidas en un **chasis remoto**, el escán del programa se interrumpe solamente para actualizar el búfer de E/S remotas con los estados actuales de los bits de imagen de salida. Esto inmediatamente hace disponibles estos estados para el próximo escán normal de E/S remotas a la vez que continúa el escán de programa. Las salidas no se escanean antes de que el escán del programa continúe.

Posicione el renglón con la instrucción IOT inmediatamente después de los renglones que examinan los bits de imagen de salida crítica que van a ser actualizados por la instrucción IOT.

Para la instrucción IOT, es necesario introducir solamente el número del rack de E/S y el número de grupo de E/S. No es necesario introducir un número de archivo.



ATENCION: No introduzca una dirección que incluya un número de archivo tal como O:027. El procesador interpreta el arreglo de bits encontrado en dicha dirección como el rack de E/S y el número de grupo de E/S de las salidas que se deben actualizar. Esto resultará en la operación inesperada, daños al equipo y lesiones corporales eventuales.

Vea el capítulo 15 para obtener más información acerca del escán de E/S y transferencias en bloques.

Immediate Data Input (IDI)

Descripción:

IDI IMMEDIATE DATA INPUT	
Data file offset	232
Length	10
Destination	N10:232

Cuando el renglón se hace verdadero, la instrucción IDI realiza una actualización inmediata del archivo de entrada de datos ControlNet desde los búferes de memoria ControlNet antes de la próxima actualización normal de la imagen de entrada (la cual ocurre al final del escán del programa).

Para programar una instrucción IDI, es necesario proporcionar al procesador la información siguiente que éste almacena en su bloque de control:

- **Data file offset** especifica el offset en el archivo de entrada de datos (DIF) donde se leen las palabras. Puede ser un valor inmediato (0-999) o una dirección lógica que especifica el offset del archivo de imagen de datos.
- **Length** especifica el número de palabras que se deben transferir. Un valor inmediato (0-64) o una dirección lógica que especifica el número de palabras que se deben transferir.
- **Destination** especifica una dirección de la tabla de datos que se debe usar como destino de las palabras a transferir.

Importante: El destino debe ser la dirección coincidente de la tabla de datos en el archivo de entrada de datos (DIF) excepto cuando se usa la instrucción para asegurar la integridad del bloque de datos en el caso de interrupciones temporizadas seleccionables (STI). Vea la página 1-9 para obtener más información.

Immediate Data Output (IDO)

Descripción:

IDO IMMEDIATE DATA OUTPUT	
Data file offset	232
Length	10
Source	N7:232

Cuando el renglón se hace verdadero, la instrucción IDO realiza una actualización inmediata de los búferes de memoria ControlNet desde el archivo de origen antes de la próxima actualización de imagen de salida. Esto transmite la información actualizada del archivo de salida de datos mediante la red ControlNet al dispositivo ControlNet apropiado.

Para programar una instrucción IDO, es necesario proporcionar al procesador la información siguiente que éste almacena en su bloque de control:

- **Data file offset** especifica el offset en el archivo de salida de datos (DOF) donde se escriben las palabras. Puede ser un valor inmediato (0-999) o una dirección lógica que especifica el offset del archivo de imagen de datos.
- **Length** especifica el número de palabras que se deben transferir. Un valor inmediato (0-64) o una dirección lógica que especifica el número de palabras que se deben transferir.
- **Destination** especifica una dirección de la tabla de datos que se debe usar como el origen de las palabras a transferir.

Importante: El destino debe ser la dirección coincidente de la tabla de datos en el archivo de salida de datos (DOF) excepto cuando se usa la instrucción para asegurar la integridad del bloque de datos en el caso de interrupciones temporizadas seleccionables (STI). Vea la página 1-9 para obtener más información..

Cómo usar las instrucciones IDI e IDO

Puede usar las instrucciones IDI e IDO para la entrada y salida de datos inmediatas en ControlNet.

Vea el manual de programación para obtener información más detallada acerca de cómo escribir los programas de lógica de escalera.

Importante: Tenga cuidado al usar las interrupciones temporizadas seleccionables (STI) con un programa en una red ControlNet.

Una interrupción temporizada seleccionable (STI) interrumpe periódicamente la ejecución primaria de programas para completar la ejecución de un subprograma. Si una STI ocurre durante la realización de una transferencia normal de E/S no discretas ControlNet o una instrucción de E/S de datos inmediatos ControlNet (IDI o IDO), las cuales operan en el mismo conjunto de datos, la integridad del bloque de datos se pone en peligro.

Para proteger la integridad del bloque de datos, escriba la rutina STI para que opere en su propia copia del bloque de datos que necesite. Use las instrucciones de E/S de datos inmediatos (IDI e IDO) dentro de la STI para copiar el bloque de datos necesario hacia y desde una ubicación temporal que sea diferente que la usada por la tabla de datos normal.

Vea el manual del usuario del software para obtener información más detallada acerca de las STI.

Notas:

Instrucciones de temporizador TON, TOF, RTO

Instrucciones de contador CTU, CTD Restablecimiento RES

Cómo usar los temporizadores y contadores

Los temporizadores y contadores le permiten controlar operaciones según el tiempo o el número de eventos. La tabla 2.A indica las instrucciones de temporizador y contador disponibles.

Tabla 2.A
Instrucciones de temporizador y contador disponibles

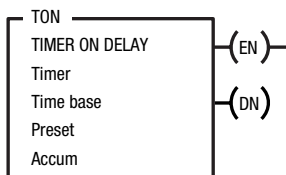
Si desea:	Use esta instrucción:	Que se encuentra en la página:
Retardar la activación de una salida	TON	2-4
Retardar la desactivación de una salida	TOF	2-7
Temporizar un evento retentivamente	RTO	2-10
Contar progresivamente	CTU	2-15
Contar regresivamente	CTD	2-17
Restablecer un contador, temporizador o instrucción de contador	RE	2-20

Vea el Apéndice C para obtener más información acerca de los operandos (y los tipos/valores válidos de cada operando) usados por las instrucciones descritas en este capítulo.

Cómo usar los temporizadores

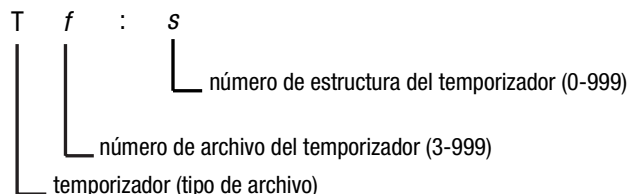
Antes de programar las instrucciones de temporizador, usted debe entender los parámetros que introduce para las instrucciones de temporizador y así como la precisión del temporizador.

Introducción de parámetros



Para programar una instrucción de temporizador, proporcione al procesador la siguiente información:

- **Timer** es la dirección de control del temporizador en la zona del temporizador (T) del almacenamiento de datos. Use el siguiente formato de dirección:



Importante: Puede usar cualquier número de archivo del temporizador de 3 a 999. No obstante, el número predeterminado del archivo del temporizador es 4. Si desea especificar un número de archivo del temporizador como cualquier archivo entre 3 y 8 (con la excepción del número 4 predeterminado), primero debe eliminar todo el archivo predeterminado para dicho número y crear el archivo del temporizador. Por ejemplo, si desea un número de archivo del temporizador como archivo 3, primero debe eliminar todo el archivo binario predeterminado y crear el archivo del temporizador como archivo 3.

Use el siguiente formato de dirección para obtener acceso a un bit de estado del temporizador, valor preseleccionado o acumulado almacenado en la dirección de control del temporizador:

Bit de estado	Valor preseleccionado	Valor acumulado
<i>Tf:s.sb</i>	<i>Tf:s.PRE</i>	<i>Tf:s.ACC</i>

Las letras *sb* especifican un mnemónico del bit de estado, tal como .DN.

Importante: El procesador almacena los bits de estado del temporizador y los valores preseleccionados y acumulados en una estructura de almacenamiento de 48 bits (tres palabras de 16 bits) en un archivo del temporizador (T).

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	
T4:0	EN	TT	DN						uso interno solamente								Palabra de control para T4:0
	valor preseleccionado (16 bits)																
	valor acumulado (16 bits)																
T4:1	EN	TT	DN						uso interno solamente								Palabra de control para T4:1
	valor preseleccionado (16 bits)																
	valor acumulado (16 bits)																
T4:2																	:

- **Time base** determina cómo funciona el temporizador. La tabla 2.B indica las bases de tiempo posibles.

Tabla 2.B**Valores disponibles de la base de tiempo**

Introduzca esta base de tiempo:	El rango del valor acumulado es:
1 segundo	hasta 32,767 intervalos de base de tiempo (hasta 9.1 horas)
0.01 segundo (10 ms)	hasta 32,767 intervalos de base de tiempo (hasta 5.5 minutos)

- **Preset** especifica el valor que el temporizador debe alcanzar antes de que el procesador establezca el bit de efectuado (.DN). Es necesario introducir un valor preseleccionado de 0-32,767. El procesador almacena el valor preseleccionado como un valor entero de 16 bits.

Importante: El valor preseleccionado funciona de otra forma si se usa una instrucción TOF. Vea la página 2-7 para obtener más información.

- **El valor acumulado** es el número de incrementos de tiempo que la instrucción ha contado. Cuando se habilita, el temporizador actualiza este valor continuamente. Es usual introducir cero cuando se programa la instrucción. Si se introduce un valor, la instrucción comienza a contar los intervalos de la base de tiempo a partir de dicho valor. El rango para el valor acumulado es 0-32,767. El procesador almacena el valor acumulado como un número entero de 16 bits.

Importante: El valor acumulado funciona de otra forma si se usa una instrucción TOF. Vea la página 2-7 para obtener más información.

Precisión del temporizador

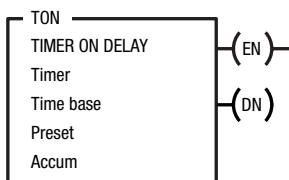
La precisión del temporizador significa el plazo de tiempo entre el momento en que el procesador habilita una instrucción del temporizador y el momento en que el procesador completa el intervalo temporizado. La precisión del temporizador depende de la tolerancia de reloj del temporizador y la base de tiempo. La tolerancia de reloj es $\pm 0.02\%$. Esto quiere decir que un temporizador puede sobrepasar el tiempo de espera en un valor temprano o tarde de 0.01 segundo (10 ms) para una base de tiempo de 0.01 ó de 1 segundo para una base de tiempo de 1 segundo.

El temporizador de 0.01 segundo mantiene la precisión con un escán de programa de hasta 2.5 segundos. El temporizador de 1 segundo mantiene la precisión con un escán de programa de hasta 1.5 segundos. Si los programas exceden 1.5 ó 2.5 segundos, repita el renglón de la instrucción de temporizador para que el renglón se escanee dentro de estos límites.

El valor acumulado mostrado en pantalla de un temporizador indica el tiempo real pero depende del tiempo de actualización CRT. El valor acumulado puede parecer menor que el valor preseleccionado cuando el valor del bit efectuado se establece.

Timer On Delay (TON)

Descripción:



Use la instrucción TON para activar y desactivar una salida después que el temporizador ha funcionado durante un intervalo de tiempo preseleccionado. La instrucción TON comienza a acumular el tiempo cuando el renglón se hace verdadero y continúa hasta que ocurre cualquiera de los siguientes eventos:

- el valor acumulado es igual al valor preseleccionado
- el renglón se hace falso
- una instrucción de restablecimiento restablece el temporizador
- el paso SFC se desactiva
- el procesador restablece el valor acumulado cuando las condiciones del renglón se hacen falsas independientemente de que el temporizador haya sobrepasado o no el tiempo de espera.

Cómo usar los bits de estado

Examine los bits de estado en el programa de lógica de escalera para activar un evento. El procesador cambia los estados de los bits de estado cuando el procesador ejecuta esta instrucción. Los bits de estado se direccionan mediante mnemónicos.

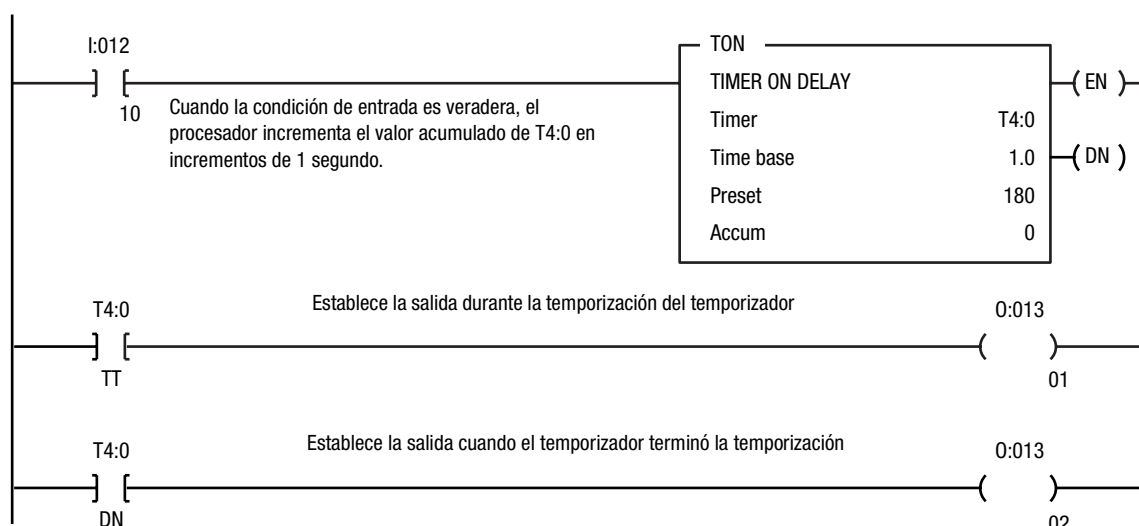
Este bit:	Se establece cuando:	Indica:	Y permanece establecido hasta que ocurre cualquiera de los eventos siguientes:
Habilitación del temporizador.EN (bit 15)	el renglón se hace verdadero	que el temporizador está habilitado	• el renglón se hace falso • una instrucción de restablecimiento restablece el temporizador • el paso SFC se desactiva
Bit de temporización del temporizador.TT (bit 14)	el renglón se hace verdadero	que se está realizando una operación de temporización	• el renglón se hace falso • el bit .DN se establece (.ACC = .PRE) • una instrucción de restablecimiento restablece el temporizador • el paso SFC asociado se desactiva
Bit de efectuado del temporizador.DN (bit 13)	el valor acumulado es igual al valor preseleccionado	que se completó una operación de temporización	• el renglón se hace falso • una instrucción de restablecimiento restablece el temporizador • el paso SFC asociado se desactiva

Si usted establece el bit de efectuado .DN mediante una instrucción OTE, por ejemplo, puede poner en pausa el temporizador. Los bits .EN y .TT permanecen establecidos, pero el valor acumulado no incrementa. La temporización se reanuda cuando restablece el bit .DN. Si el renglón se hace falso durante la pausa del temporizador, el temporizador se restablece normalmente.

1. Si cambia al modo de Programación o si el procesador pierde energía antes de que la instrucción alcance el valor preseleccionado, ocurre lo siguiente:
 - el bit de habilitación del temporizador (.EN) permanece establecido
 - el bit de temporización del temporizador (.TT) permanece establecido
 - el valor acumulado (.ACC) permanece igual
2. Cuando cambia nuevamente al modo Marcha o Prueba o cuando la alimentación eléctrica se vuelve a conectar, ocurre lo siguiente:

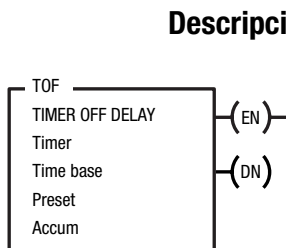
Condición:	Resultado:
Si el renglón es verdadero:	El bit .EN permanece establecido El bit .TT permanece establecido El bit .DN permanece restablecido El valor .ACC se restablece y comienza a contar progresivamente
Si el renglón es falso:	El bit .EN se restablece El bit .TT se restablece El bit .DN se restablece El valor .ACC se restablece

Figura 2.1
Ejemplo de un diagrama de lógica de escalera TON



Cuando el bit I:012/10 se establece, el procesador inicia T4:0. El valor acumulado incrementa en intervalos de 1 segundo. T4:0.TT se establece y el bit de salida :013/01 se establece (el dispositivo de salida asociado se activa) durante la temporización del temporizador. Cuando el temporizador termina la temporización (.ACC = .PRE), T4:0.TT se restablece (para que 0:013/01 y el dispositivo de salida asociado se desactiven) y T4:0.DN se establece (para que 0:013/02 se establezca y el dispositivo de salida asociado se active). Cuando el valor acumulado llega a 180, el bit .DN se establece. Si el renglón se hace falso, el temporizador se restablece.

Timer Off Delay (TOF)



Descripción:

Use la instrucción TOF para activar y desactivar una salida después que el renglón ha estado desactivado durante un intervalo de tiempo preseleccionado. La instrucción TOF comienza a acumular el tiempo cuando el renglón se hace falso y continúa temporizando hasta que ocurre cualquiera de las siguientes condiciones:

- el valor acumulado es igual al valor preseleccionado
- el renglón se hace verdadero
- una instrucción de restablecimiento restablece el temporizador
- el paso SFC se desactiva

El procesador restablece el valor acumulado cuando las condiciones del renglón se hacen verdaderas independientemente de que el temporizador haya sobrepasado o no el tiempo de espera.

Cómo usar los bits de estado

Examine los bits de estado en el programa de lógica de escalera para activar un evento. El procesador cambia los estados de los bits de estado cuando el procesador ejecuta esta instrucción. Los bits de estado se direccionan mediante mnemónicos.

Este bit:	Se establece cuando:	Y permanece establecido hasta que ocurre cualquiera de los eventos siguientes:
Habilitación del temporizador.EN (bit 15)	el renglón se hace verdadero	• el renglón se hace falso • una instrucción de restablecimiento restablece el temporizador • el paso SFC se desactiva
Bit de temporización del temporizador.TT (bit 14)	el renglón se hace falso y el valor acumulado es menor que el valor preseleccionado	• el renglón se hace falso • el bit .DN se establece (.ACC = .PRE) • una instrucción de restablecimiento restablece el temporizador • el paso SFC asociado se desactiva
Bit de efectuado del temporizador.DN (bit 13)	el renglón se hace verdadero	• el valor acumulado es igual al valor preseleccionado

Si usted establece el bit de efectuado .DN mediante una instrucción OTE, por ejemplo, puede poner en pausa el temporizador. Los bits .EN y .TT permanecen establecidos, pero el valor acumulado no incrementa. La temporización se reanuda cuando restablece el bit .DN. Si el renglón se hace falso durante la pausa del temporizador, el temporizador se restablece normalmente.

1. Si cambia al modo de Programación o si el procesador pierde energía o si el fallo del procesador interrumpe la instrucción TOF antes de que alcance el valor preseleccionado, ocurre lo siguiente:
 - el bit de habilitación del temporizador (.EN) permanece establecido
 - el bit de temporización del temporizador (.TT) permanece establecido
 - el bit de efectuado del temporizador (.DN) permanece establecido
 - el valor acumulado (.ACC) permanece igual
2. Cuando cambia nuevamente al modo Marcha o Prueba, ocurre lo siguiente:

Condición:	Resultado:
Si el renglón es verdadero:	El bit .EN se establece El bit .TT se restablece El bit .DN permanece restablecido El valor .ACC se borra
Si el renglón es falso:	El bit .EN se restablece El bit .TT se restablece El bit .DN se restablece El valor .ACC es igual al valor PRE (el temporizador no comienza a temporizar)

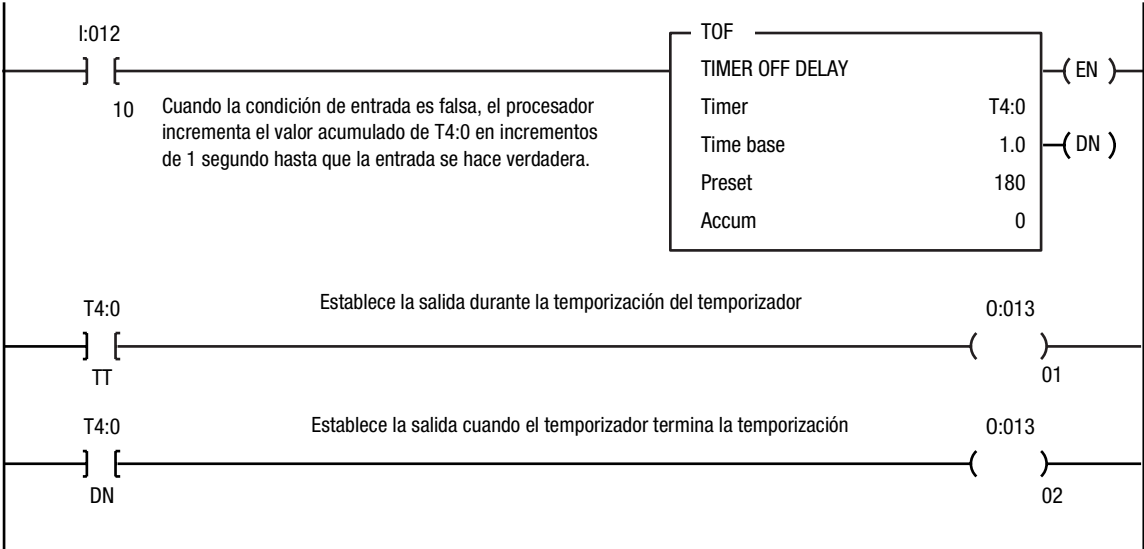


ATENCION: Puesto que la instrucción REST restablece el valor acumulado, el bit de efectuado y los bits de temporización de una instrucción de temporización, no use la instrucción RES para restablecer un temporizador TOF.

Lo siguiente ocurre durante un preescán:

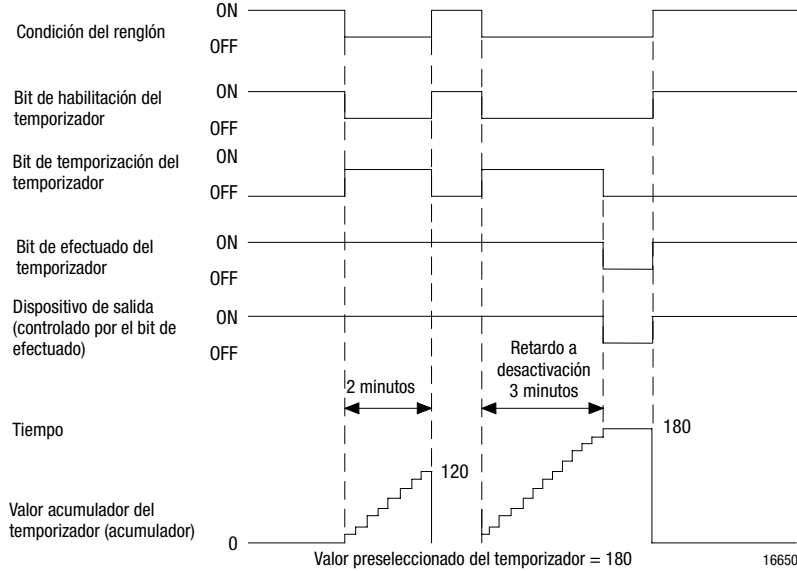
- se restablece el bit de temporización del temporizador (.TT)
- el valor acumulado (.ACC) es igual al valor preseleccionado

Figura 2.3
Ejemplo de un diagrama de lógica de escalera TOF



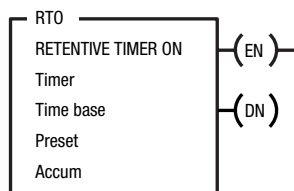
Cuando el bit I:012/10 se restablece, el procesador inicia T4:0. El valor acumulado incrementa en intervalos de 1 segundo siempre que el renglón permanezca falso. T4:0.TT se establece y el bit de salida 0:013/01 se establece (el dispositivo de salida asociado se activa) durante la temporización del temporizador. Cuando el temporizador termina la temporización (.ACC = .PRE), T4:0.TT se restablece (para que 0:013/01 y el dispositivo de salida asociado se desactiven) y T4:0.DN se establece (para que 0:013/02 se establezca y el dispositivo de salida asociado se desactive). Cuando el valor acumulado alcanza 180 ó cuando las condiciones del renglón se hacen verdaderas, el temporizador se detiene.

Figura 2.4
Ejemplo de un diagrama de temporización TOF



Retentive Timer On (RTO)

Descripción:



Use la instrucción RTO para activar o desactivar una salida después que el temporizador de la misma ha estado activado durante un intervalo de tiempo preseleccionado. La instrucción RTO permite que el temporizador se detenga y arranque sin la necesidad de restablecer el valor acumulado.

La instrucción RTO comienza a temporizar cuando el renglón se hace verdadero. Siempre que el renglón permanezca verdadero, el temporizador actualiza el valor acumulado durante cada escán de programa hasta que alcanza el valor preseleccionado. La instrucción RTO retiene su valor acumulado aun cuando ocurra cualquiera de las condiciones siguientes:

- el renglón se hace falso
- se cambia al modo de Programación
- el procesador entra en fallo o pierde energía
- el paso SFC se desactiva

Cuando el procesador reanuda la operación o cuando el renglón se hace verdadero, la temporización continúa a partir del valor acumulado retenido. Puesto que los temporizadores retentivos retienen su valor acumulado, miden a su vez el período acumulativo durante el cual el renglón es verdadero.

Importante: Para restablecer el valor acumulado del temporizador y los bits de estado después de que el renglón RTO se hace falso, es necesario programar una instrucción de restablecimiento RES con la misma dirección en otro renglón.

Cómo usar los bits de estado

Examine los bits de estado en el programa de lógica de escalera para activar un evento. El procesador cambia el estado de los bits de estado cuando el procesador ejecuta esta instrucción. Los bits de estado se direccionan mediante mnemónicos.

Este bit:	Se establece cuando:	Indica:	Y permanece establecido hasta que ocurre cualquiera de los eventos siguientes:
Bit de habilitación del temporizador.EN (bit 15)	el renglón se hace verdadero	que se realiza una operación de temporización	• el renglón se hace falso • una instrucción de restablecimiento restablece el temporizador
Bit de temporización del temporizador.TT (bit 14)	el renglón se hace verdadero	que se realiza una operación de temporización	• el renglón se hace falso • el bit .DN se establece • el valor acumulado es igual al valor preseleccionado (.ACC=.PRE) • una instrucción de restablecimiento restablece el temporizador
Bit de efectuado del temporizador.DN (bit 13)	el valor acumulado es igual al valor preseleccionado	que se ha completado una operación de temporización	• el bit .DN se restablece mediante la instrucción RES

Si establece el bit de efectuado .DN con el uso de una instrucción OTE, por ejemplo, puede poner en pausa el temporizador. Los bits .EN y .TT permanecen establecidos, pero el valor acumulado no incrementa. La temporización se reanuda cuando se restablece el bit .DN. Si el renglón se hace falso durante la pausa del temporizador, el temporizador se restablece normalmente.

1. Si cambia al modo de Programación o si el procesador pierde energía o si el fallo del procesador interrumpe la instrucción RTO antes de que alcance el valor preseleccionado, ocurre lo siguiente:
 - el bit de habilitación del temporizador (.EN) permanece establecido
 - el bit de temporización del temporizador (.TT) permanece establecido
 - el valor acumulado (.ACC) permanece igual
2. Cuando cambia nuevamente al modo Marcha o Prueba, ocurre lo siguiente:

Condición:	Resultado:
Si el renglón es verdadero:	El bit .EN permanece establecido El bit .TT permanece establecido El valor .ACC continúa temporizando
Si el renglón es falso:	El bit .EN se restablece El bit .TT se restablece El bit .DN permanece igual El valor .ACC permanece igual

Figura 2.5
Ejemplo de un diagrama de lógica de escalera RTO

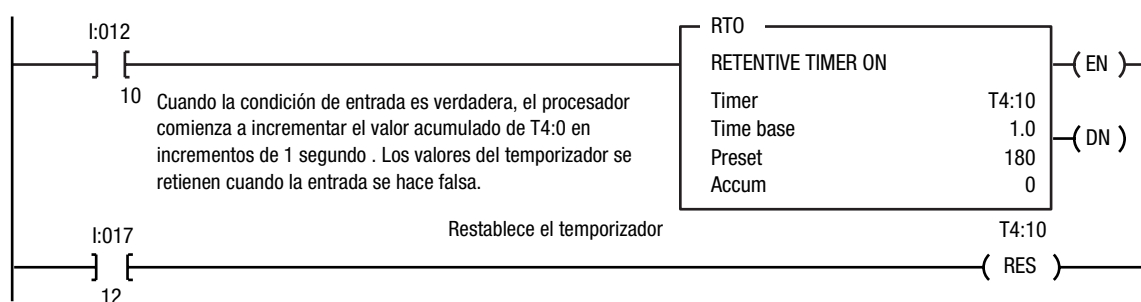
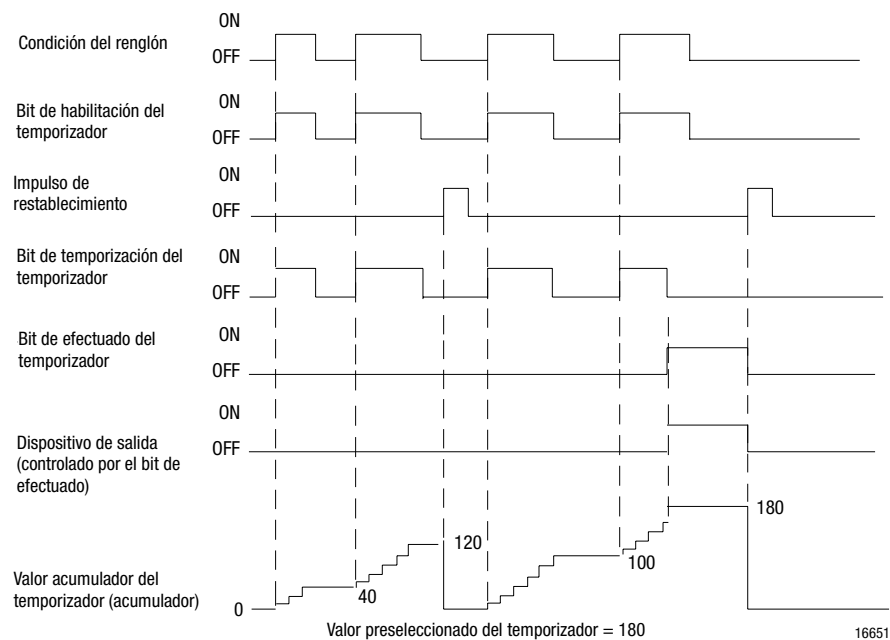
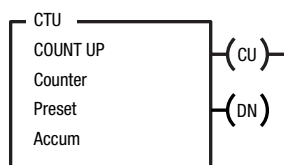


Figura 2.6
Diagrama de temporización del temporizador retentivo



Cómo usar los contadores

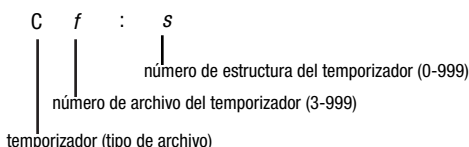


Antes de usar las instrucciones del contador, es necesario entender los parámetros que se introducen.

Cómo introducir los parámetros

Proporcione al procesador la información siguiente para programar una instrucción del contador:

- **Counter** es la dirección de control del contrador en la zona del contador (C) del almacenamiento de datos. Use el siguiente formato de dirección:



Importante: Puede usar cualquier número de archivo del temporizador de 3 a 999. No obstante, el número predeterminado del archivo del temporizador es 5. Si desea especificar un número de archivo del temporizador como cualquier archivo entre 3 y 8 (con la excepción del número 5 predeterminado), primero debe eliminar todo el archivo predeterminado para dicho número y crear el archivo del temporizador. Por ejemplo, si desea un número de archivo del temporizador como archivo 3, primero debe eliminar todo el archivo binario predeterminado y crear el archivo del temporizador como archivo 3.

Use el siguiente formato de dirección para obtener acceso al bit de estado, valor preseleccionado o valor acumulado del contador:

Bit de estado	Valor preseleccionado	Valor acumulado
<i>Cf.s.bb</i>	<i>Cf.s.PRE</i>	<i>Cf.s.ACC</i>

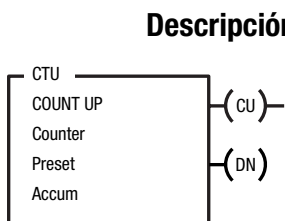
Las letras *bb* son un mnemónico del bit de estado, tal como .DN.

Importante: El procesador almacena los bits de estado del contador y los valores preseleccionados y acumulados en una estructura de almacenamiento (48 bits – tres palabras de 16 bits) en un archivo de contador (C) en la tabla de datos..

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	
C5:0	CU	CD	DN	OV	UN						uso interno solamente						Palabra de control para T4:0
	valor preseleccionado (16 bits)																
	valor acumulado (16 bits)																
C5:1	CU	CD	DN	OV	UN						uso interno solamente						Palabra de control para T4:1
	valor preseleccionado (16 bits)																
C5:2	valor acumulado (16 bits)																
	:																

- **Preset** especifica el valor que el contador debe alcanzar antes de establecer el bit de efectuado .DN. Introduzca un valor preseleccionado de -32,768 hasta +32,767. El valor preseleccionado se almacena como valor de número entero de 16 bits. Los valores negativos se almacenan en el formato de complemento a 2.
- **Accumulated Value** es el conteo actual según el número de veces que el renglón cambia de falso a verdadero. El valor acumulado se almacena como valor de número entero de 16 bits. Los valores negativos se almacenan en el formato de complemento a 2. El rango del valor acumulado es -32,768 a +32,767. Es usual introducir un valor de cero cuando se programan las instrucciones del contador. Si se establece un valor que no sea cero, la instrucción comienza a contar a partir de dicho valor. Si el contador se restablece, el valor acumulado se establece en cero.

Count Up (CTU)



La instrucción CTU cuenta progresivamente para abarcar un rango de -32,768 a +32,767. Cada vez que el renglón cambia de falso a verdadero, la instrucción CTU incrementa el valor acumulador por un conteo. Cuando el valor acumulado es igual o excede el valor preseleccionado, la instrucción CTU establece un bit de efectuado .DN, el cual el programa de lógica de escalera puede usar para iniciar una acción tal como controlar de un bit de almacenamiento o un dispositivo de salida.

El valor acumulador de un contador es retentivo. El conteo se retiene hasta que una instrucción de restablecimiento (RES) lo restablece. Esta instrucción debe tener la misma dirección que el contador.

Cómo usar los bits de estado

Examine los bits de estado en el programa de lógica de escalera para activar un evento. El procesador cambia los estados de los bits de estado cuando el procesador ejecuta la instrucción CTU. Los bits de estado se direcciona mediante mnemónicos.

Este bit:	Se establece:	Y permanece establecido hasta que ocurre cualquiera de los eventos siguientes:
Bit de habilitación de conteo progresivo .CU (bit 15)	cuando el renglón se hace verdadero para indicar que la instrucción ha incrementado el conteo Nota: este bit se establece anteriormente al escán para evitar un conteo falso cuando comienza el escán del programa durante el preescán.	- el renglón se hace falso - una instrucción RES restablece el bit .DN
Bit de efectuado de conteo progresivo .DN (bit 13)	cuando el valor acumulador es mayor o igual al valor preseleccionado	- el valor acumulado cuenta por debajo del valor preseleccionado mediante una instrucción CTD para contar regresivamente o un cambio del valor acumulado - una instrucción RES restablece el bit .DN
Bit de overflow de conteo progresivo .OV (bit 12)	cuando el contador ha excedido el límite superior de +32,767 y regresado a -32,768. El CTU cuenta progresivamente a partir de este número.	- una instrucción RES restablece el bit .DN - se cuenta regresivamente hasta 32,767 mediante una instrucción CTD con la misma dirección



ATENCION: Coloque los contadores críticos fuera de la zona MCR o secciones saltadas del programa de lógica de escalera para evitar resultados no válidos que pueden conducir a daños del equipo o lesiones corporales.

Figura 2.7
Ejemplo de un diagrama de lógica de escalera CTU

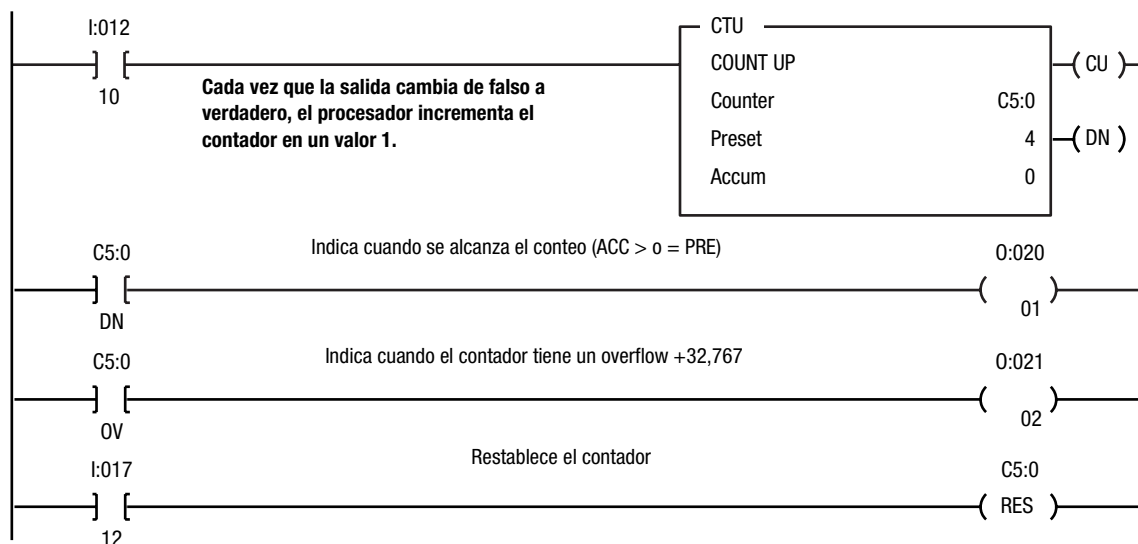
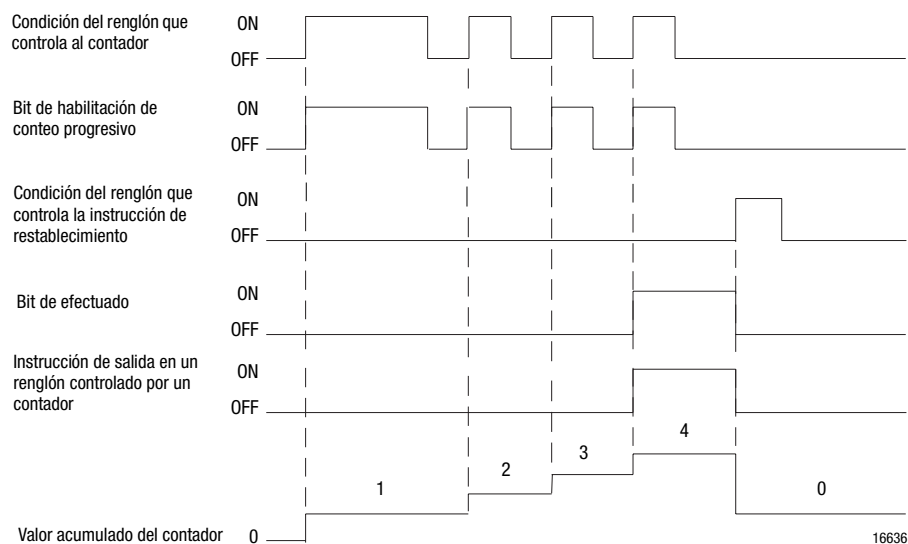


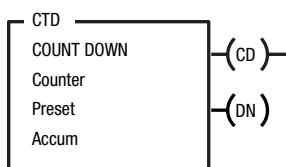
Figura 2.8
Ejemplo de un diagrama de temporización CTU

Valor preseleccionado del contador = 4 conteos



Count Down (CTD)

Descripción:



La instrucción CTD cuenta regresivamente para abarcar un rango de +32,767 a -32,768. Cada vez que el renglón cambia de falso a verdadero, la instrucción CTD decrementa el valor acumulado por un conteo. El bit de efectuado .DN se establece siempre que el valor acumulado sea mayor o igual al valor preseleccionado. Cuando el valor acumulado es menos que el valor preseleccionado, el bit de efectuado .DN se establece. El programa de lógica de escalera puede usar este bit para iniciar una acción, tal como controlar de un bit de almacenamiento o un dispositivo de salida.

El valor acumulado de un contador es retentivo. El conteo se retiene hasta que una instrucción de restablecimiento (RES) lo restablece. Esta instrucción debe tener la misma dirección que la instrucción CTD.

Cómo usar los bits de estado

Examine los bits de estado en el programa de lógica de escalera para activar un evento. El procesador cambia los estados de los bits de estado cuando el procesador ejecuta esta instrucción. Los bits de estado se direccionan mediante mnemónicos.

Este bit:	Se establece:	Y permanece establecido hasta que ocurre cualquiera de los eventos siguientes:
Bit de habilitación de conteo regresivo .CU (bit 15)	cuando el renglón se hace verdadero para indicar que el contador está habilitado como un contador regresivo Nota: este bit está establecido para evitar un conteo falso cuando se comienza el escán de programa durante el preescán.	- el renglón se hace falso - una instrucción RES restablece el bit .DN
Bit de efectuado de conteo regresivo .DN (bit 13)	cuando el valor acumulador es mayor o igual al valor preseleccionado	- el valor acumulado cuenta por debajo del valor preseleccionado - otra instrucción cambia el valor acumulado - una instrucción RES restablece el bit .DN
Bit de underflow de conteo regresivo .UN (bit 11)	por el procesador para indicar que el contador regresivo pasó por debajo del límite inferior de -32,768 y ha regresado a +32,767. La instrucción CTD cuenta a partir de este número.	- una instrucción RES restablece el bit .DN - se cuenta regresivamente hasta -32,768 mediante una instrucción CTD



ATENCION: Coloque los contadores críticos fuera de la zona MCR o secciones saltadas del programa de lógica de escalera para evitar resultados no válidos que pueden conducir a daños del equipo o lesiones corporales.

Figura 2.9
Ejemplo de un diagrama de lógica de escalera CTD

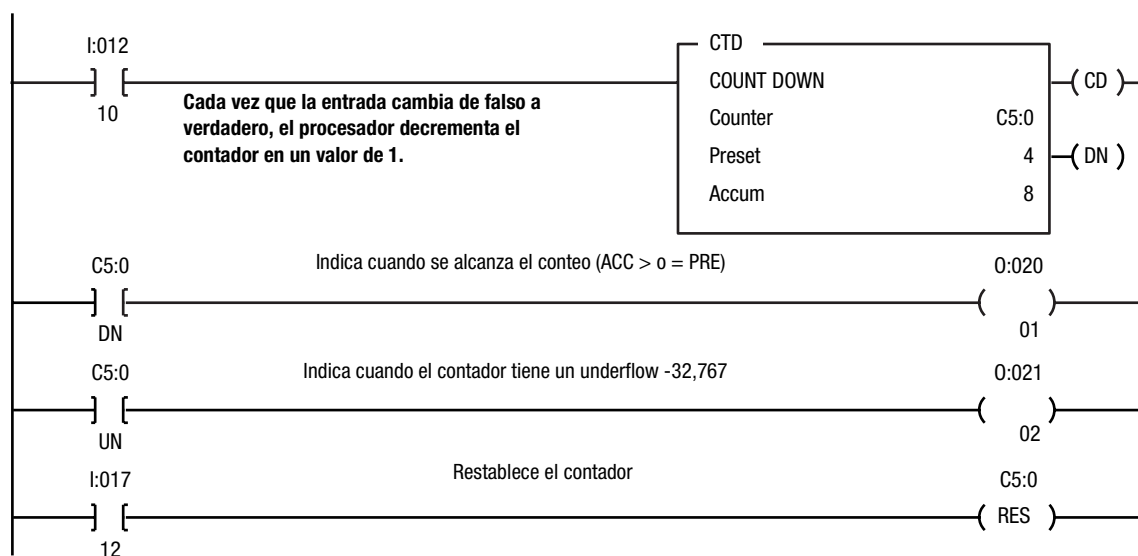


Figura 2.10
Ejemplo de un diagrama de temporización CTD

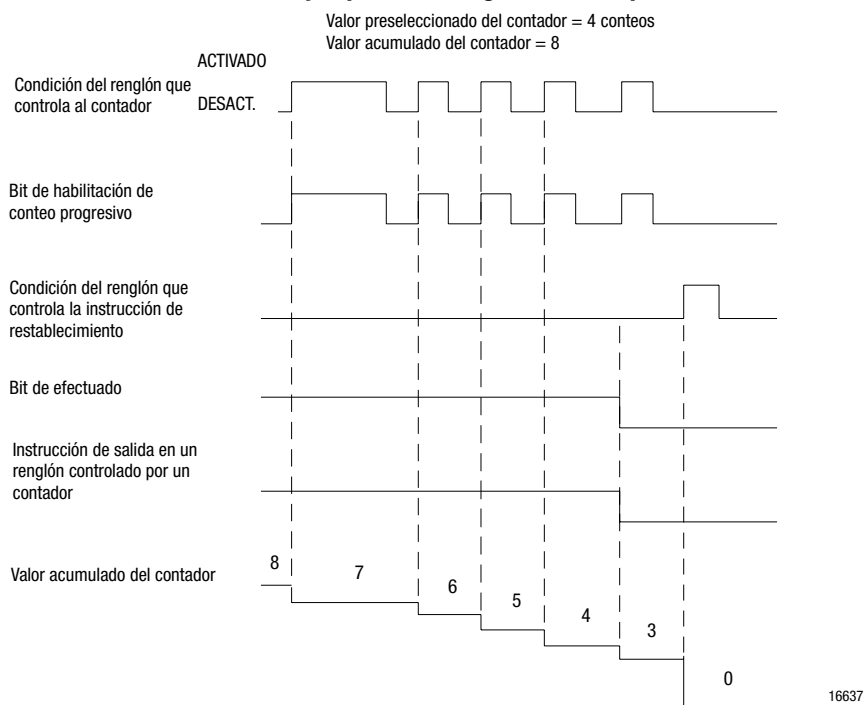


Figura 2.11
Ejemplo de un diagrama de lógica de escalera CTU y CTD

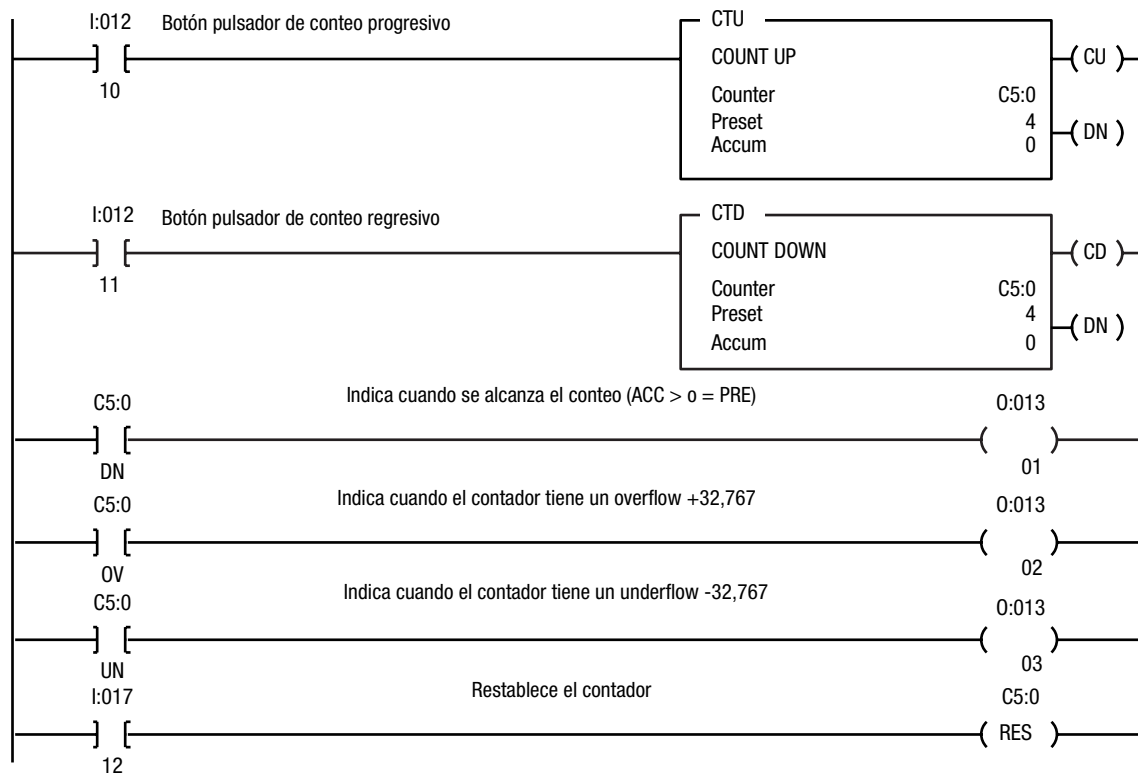
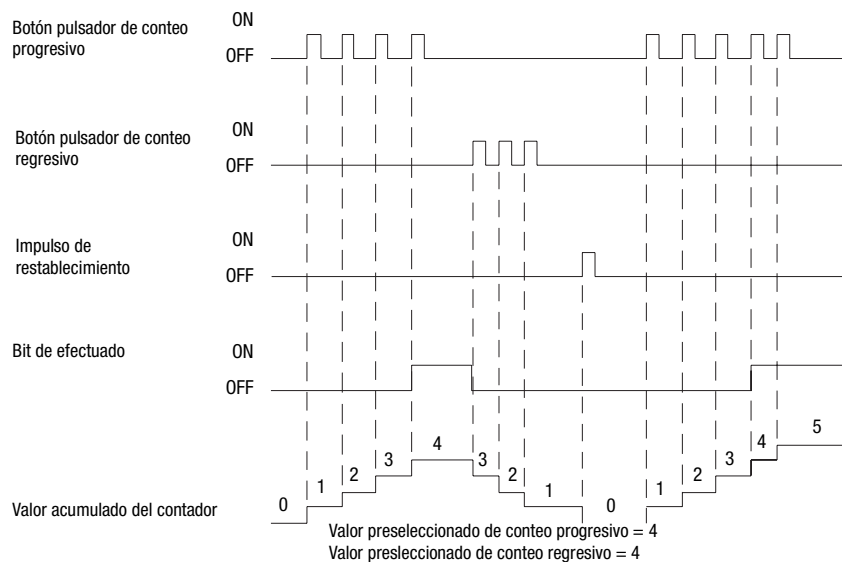


Figura 2.12
Ejemplo de un diagrama de temporización CTU y CTD



Timer and Counter Reset (RES)

Descripción:


La instrucción RES es una instrucción de salida que restablece un temporizador o contador. La instrucción RES se ejecuta cuando su renglón es verdadero.

Cuando se usa una instrucción RES para: **El procesador restablece el:**

El temporizador (no use una instrucción RES para .TOF)	Valor .ACC
	Bit .EN
	Bit .TT
	Bit .DN
El contador	Valor .ACC
	Bit .EN
	bit .OV o .UN
	Bit .DN

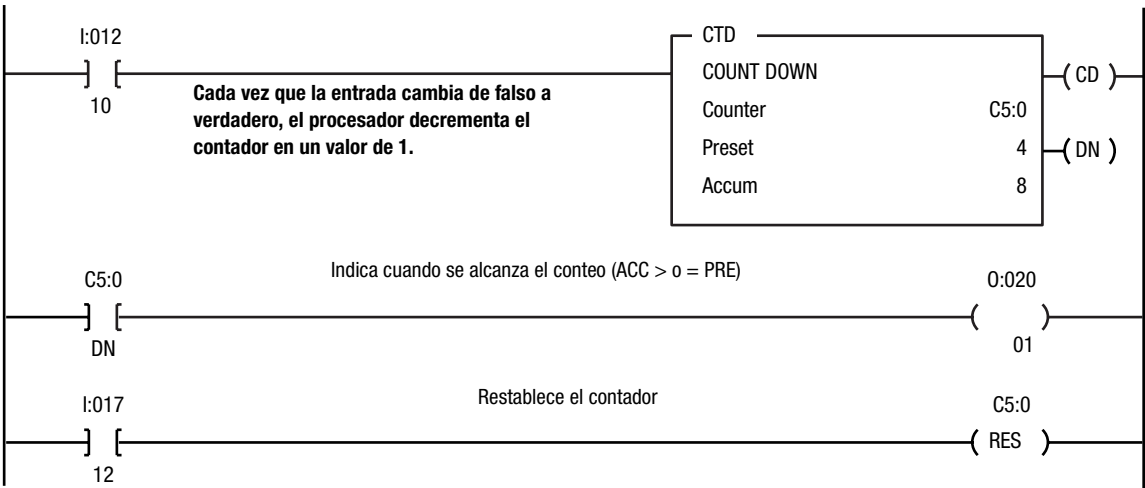
Si el renglón del contador esté habilitado, el bit CU o CD se restablecerá siempre que la instrucción RES esté habilitada.

Importante: Usted puede usar un valor preseleccionado negativo en una instrucción CTU o CTD si va a usar la instrucción RES. No obstante, tome en cuenta de que la instrucción RES pone el valor acumulado en cero, lo cual puede establecer el bit .DN y evitar que la instrucción CTU o CTD funcione la próxima vez que se habilite.



ATENCION: Puesto que la instrucción RES restablece el valor acumulado, el bit .DN y el bit .TT de una instrucción de temporización, no use la instrucción RES para restablecer una instrucción TOF. Puede resultar en la operación inesperada de la máquina o lesiones corporales.

Figura 2.13
Ejemplo de un diagrama de lógica de escalera RES



Instrucciones de comparación CMP, EQU, GEQ, GRT, LEQ, LES, LIM, MEQ, NEQ

Cómo usar las instrucciones de comparación

Las instrucciones de comparación le permiten comparar los valores mediante una expresión o una instrucción de comparación específica. La tabla 3.A indica las instrucciones de comparación disponibles.

Tabla 3.A
Instrucciones de comparación disponibles

Si desea:	Use la instrucción:	Que se encuentra en la pág.:
Comparar valores en base a una expresión	CMP	3-2
Probar si dos valores son iguales	EQU	3-5
Probar si un valor es mayor o igual al segundo valor	GEQ	3-5
Probar si un valor es mayor que un segundo valor	GRT	3-6
Probar si un valor es menor o igual a un valor segundo	LEQ	3-6
Probar si un valor es menos que un valor segundo	LES	3-7
Probar si un valor se encuentra entre dos otros valores	LIM	3-7
Pasar dos valores por una máscara y probar si son iguales	MEQ	3-9
Probar si un valor no es igual a un valor segundo	NEQ	3-10

Importante: Puede comparar los valores de diferentes tipos de datos, tales como punto flotante (coma flotante) y números enteros. Debe usar los valores BCD y ASCII para mostrar los valores en pantalla. Si introduce valores BCD o ASCII, el procesador considera dichos valores como números enteros. Por ejemplo, si el valor en N7:2 es 10 (decimal) y el valor en D9:3 es 10 (BCD), la comparación de N7:2 - D9:3 se evalúa como falsa. El número 10 en BCD sale como 0000 0000 0001 0000. El número 10 en decimal sale como 0000 0000 0000 1010.

Los parámetros que se introducen son constantes de programa o direcciones lógicas de los valores que se desean comparar.

Vea el Apéndice C para obtener más información acerca de los operandos (y los tipos/valores de datos válidos de cada operando) usados por las instrucciones descritas en este capítulo.

Cómo usar los indicadores de estado aritméticos

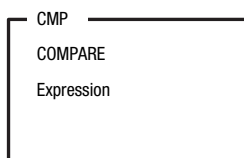
Los indicadores de estado aritméticos se encuentran en los bits 0-3 de la palabra 0 en el archivo de estado del procesador (S). Monitoree estos bits si realiza una función aritmética con la instrucción CMP. La tabla 3.B indica los bits de estado:

Tabla 3.B
Bits de estado aritméticos

Este bit:	Descripción:
S:0/0	Acarreo (C)
S:0/1	Overflow (V)
S:0/2	Cero (Z)
S:0/3	Signo (S)

Compare (CMP)

Descripción:



La instrucción CMP compara los valores y realiza comparaciones lógicas.

La instrucción CMP es una instrucción de entrada que realiza una comparación en operaciones aritméticas que usted especifica en la expresión. Cuando el procesador determina que la expresión es verdadera, el renglón se hace verdadero. De lo contrario, el renglón es falso. Los procesadores PLC-5 con características mejoradas le permiten introducir varios operandos (expresión compleja).

El tiempo de ejecución de una instrucción CMP es más largo que el tiempo de ejecución de una de las otras instrucciones de comparación (por ejemplo GRT, LEQ, etc.). Una instrucción CMP también usa más palabras en el archivo de programa que la instrucción de comparación correspondiente.

Cómo introducir una expresión CMP

La expresión define las operaciones que desea realizar. Defina la expresión con operadores y direcciones o constantes de programa. Los procesadores PLC-5 con características mejoradas le permiten introducir expresiones complejas. La tabla 3.C indica las operaciones válidas para una expresión. La lista siguiente proporciona las pautas para escribir las expresiones.

- Los operadores (símbolos) definen las operaciones
- Las direcciones pueden ser directas, indirectas o indexadas (deben estar al nivel de palabra)
- Los procesadores PLC-5 con características mejoradas le permiten que las constantes de programa sean números enteros o de punto flotante (coma flotante) (si se introducen valores octales, hay que usar un &O precedente; si se introducen valor hexadecimales, hay que usar un &H precedente; si se introducen valores binarios, hay que usar un &B precedente).

Tabla 3.C
Operaciones válidas para uso en una expresión CMP

Tipo	Operador	Descripción	Ejemplo de operación
Comparación	=	igual a	si A = B, entonces
	<>	no igual a	si A <> B, entonces ...
	<	menor que	si A < B, entonces ...
	<=	menor o igual a	si A <= B, entonces ...
	>	mayor que	si A > B, entonces ...
	>=	mayor o igual a	si A >= B, entonces ...
Aritmético	+	sumar	2 + 3 procesador con características mejoradas: 2 + 3 + 7
	-	restar	12 - 5
	*	multiplicar	5 * 2 PLC-5/30, -5/40, -5/60, -5/80: 6 * (5 * 2)
	(barra vertical)	dividir	24 6
	-	cambiar signo	- N7:0
	SQR	raíz cuadrada	SQR N7:0
	**	exponencial (x a la potencia de y)	10**3 (procesadores con características mejoradas solamente)
Conversión	FRD	convertir de BCD en binario	FRD N7:0
	TOD	convertir de binario en BCD	TOD N7:0

Cómo determinar la longitud de una expresión

Los procesadores PLC-5 con características mejoradas son compatibles con instrucciones complejas (hasta un total de 80 caracteres, incluso espacios y paréntesis). Dependiendo del operador, el procesador inserta caracteres antes o después del operador en la expresión para formatear la expresión a fin de facilitar la interpretación. Use la tabla 3.D para determinar el número de caracteres que usa cada operador en una expresión.

Importante: No puede introducir números flotantes (comas flotantes) en una notación científica con exponentes negativos en expresiones complejas. En lugar de ello, use la equivalencia decimal o ponga el número en un archivo de punto flotante (coma flotante) y use la dirección de datos en la expresión compleja.

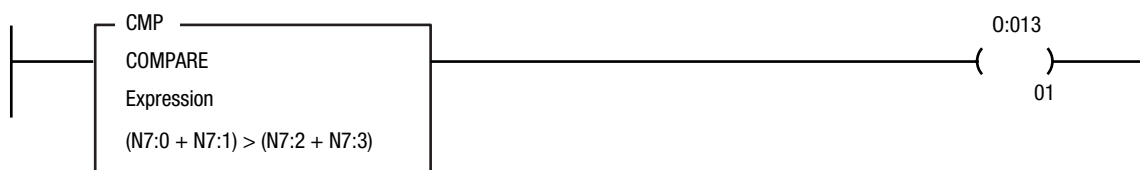
La instrucción CMP permite que se muestre en pantalla un máximo de 80 caracteres de la expresión. Si la expresión que se introduce está muy cerca del máximo de 80 caracteres, cuando usted acepta el renglón que contiene la instrucción, el procesador puede ampliarlo a más de los 80 caracteres. Cuando intenta editar la expresión, solamente los 80 primeros caracteres aparecen en pantalla y el renglón se muestra como un renglón de error. Sin embargo, el procesador contiene la expresión completa y la instrucción funciona correctamente.

Para evitar este problema de visualizar en pantalla, exporte el archivo de memoria del procesador y efectúe las ediciones en el archivo de texto PC5. Luego, importe este archivo de texto. Vea el manual de programación para obtener más información acerca de cómo importar y exportar los archivos de memoria del procesador.

Tabla 3.D
Longitudes de caracteres para los operadores

Esta operación:	Que usa este operador:	Usa este número de caracteres:
binaria matemática	+, -, *, /	3
	OR, **	4
	AND, XOR	5
matemática de un dígito	- (cambiar signo)	2
	LN	3
	FRD, TOD, DEG, RAD, SQR, NOT, LOG, SIN, COS, TAN, ASN, ACS, ATN	4
comparativa	=, <, >	3
	<>, <=, >=	4

Ejemplo:



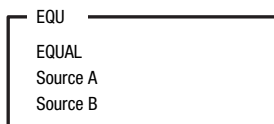
La instrucción CMP indica a un procesador PLC-5 con características mejoradas: si la suma de los valores en N7:0 y N7:1 son mayores que la suma de los valores en N7:2 y N7:3, establezca el bit de salida O:013/01. (El número total de caracteres usados en esta expresión es 3.)

Vea el capítulo 4 para obtener más información acerca de cómo introducir las expresiones complejas.

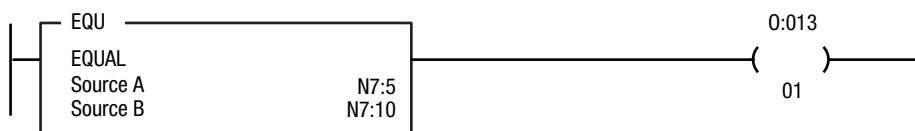
Equal to (EQU)

Descripción:

Use la sintrucción EQU para probar si dos valores son iguales. La fuente A y la fuente B pueden ser valores o direcciones que contienen valores.



Ejemplo:



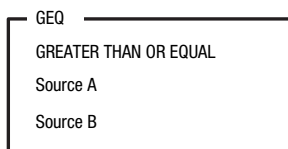
Si el valor en N7:5 es igual al valor en N7:10, establezca el bit de salida 0:013/01.

Los valores de punto flotante (coma flotante) raramente son absolutamente iguales. Si es necesario determinar la igualdad de los valores de punto flotante (coma flotante), use la instrucción LIM (en vez de EQU). Vea la página 3-7 para obtener información acerca de la instrucción LIM.

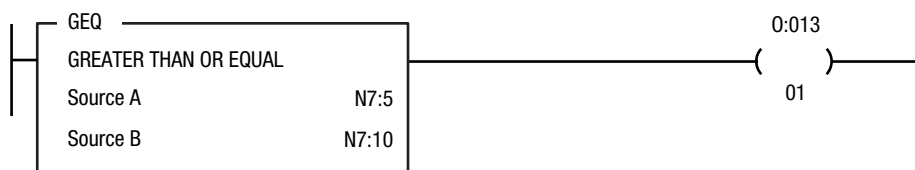
Greater than or Equal to (GEQ)

Descripción:

Use la instrucción GEQ para probar si un valor (fuente A) es mayor o igual a otro valor (fuente B). La fuente A y la fuente B pueden ser valores o direcciones que contienen valores.



Ejemplo:



Si el valor en N7:5 es mayor o igual al valor en N7:10, establezca el bit de salida 0:013/01.

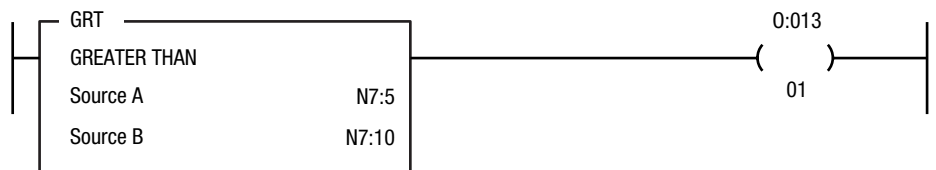
Greater than (GRT)

Descripción:

GRT
GREATER THAN OR EQUAL
Source A
Source B

Use la instrucción GRT para probar si un valor (fuente A) es mayor que otro valor (fuente B). La fuente A y la fuente B pueden ser valores o direcciones que contienen valores.

Ejemplo:



Si el valor en N7:5 es mayor que el valor en N7:10, establezca el bit de salida 0:013/01.

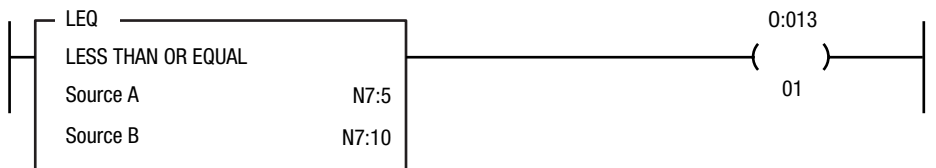
Less than or Equal to (LEQ)

Descripción:

LEQ
LESS THAN OR EQUAL
Source A
Source B

Use la instrucción LEQ para probar si un valor (fuente A) es menor o igual a otro valor (fuente B). La fuente A y la fuente B pueden ser valores o direcciones que contienen valores.

Ejemplo:



Si el valor en N7:5 es menor o igual al valor en N7:10, establezca el bit de salida 0:013/01.

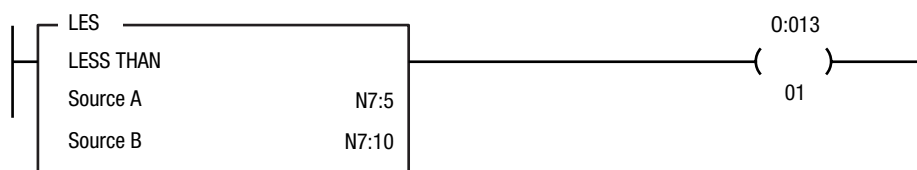
Less than (LES)

Descripción:

LES
LESS THAN
Source A
Source B

Use la instrucción LES para probar si un valor (fuente A) es menor que otro valor (fuente B). La fuente A y la fuente B pueden ser valores o direcciones que contienen valores.

Ejemplo:



Si el valor en N7:5 es menor que el valor en N7:10, establezca el bit de salida 0:013/01.

Limit Test (LIM)

Descripción:

LIM
LIMIT TEST (CIRC)
Low limit
Test
High limit

La instrucción LIM es una instrucción de entrada que prueba si hay valores dentro o fuera de un rango especificado. La instrucción es falsa hasta que detecta que el valor de prueba se encuentra dentro de límites determinados. entonces la instrucción se hace verdadera. Cuando la instrucción detecta que el valor de prueba sale fuera de los límites determinados, se hace falsa.

Usted puede usar la instrucción LIM para probar si un valor de entrada analógica se encuentra dentro de límites especificados.

Cómo introducir los parámetros

Hay que proporcionar al procesador la información siguiente para programar la instrucción LIM:

Parámetro:	Definición:
Límite bajo	una constante o una dirección desde la cual la instrucción lee el rango inferior del rango de límite especificado. La dirección contiene un valor entero o de punto flotante (coma flotante).
Valor de prueba	la dirección que contiene el valor entero o de punto flotante (coma flotante) que se examina para determinar si el valor se encuentra dentro o fuera del rango de límite especificado.
Límite alto	una constante o dirección desde la cual la instrucción lee el rango superior del rango de límite especificado. La dirección contiene un valor entero o de punto flotante (coma flotante).

Ejemplo de LIM con el uso de un número entero:

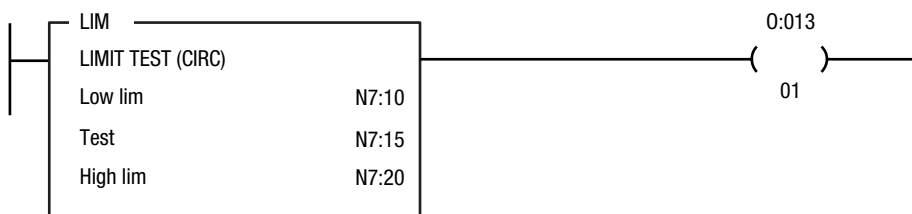
- **Si el valor del límite inferior $\leq$ el valor del límite superior:**
Cuando el procesador detecta que el valor de B (prueba) es igual o se encuentra entre los límites, la instrucción es verdadera. Si la prueba del valor se encuentra fuera de los límites, la instrucción es falsa.

falso	<	-----verdadero----- A C <valor B>	>	falso
de -32,768				a +32,767

- **Si el valor del límite inferior $\geq$ el valor del límite superior:**
Cuando el procesador detecta que el valor de la prueba es igual o se encuentra fuera de los límites, la instrucción es verdadera. Si la prueba del valor se encuentra entre pero no es igual a cualquiera de los límites, la instrucción es falsa.

verdadero	<	----- falso ----- A C valor B	>	verdadero
de -32,768				a +32,767

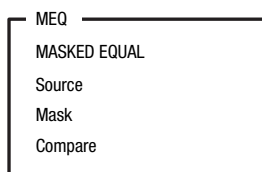
Ejemplo (cuando el límite inferior es menor que el límite superior):



Si el valor en N7:15 es mayor o igual al valor en N7:10 y menor o igual al valor en N7:20, establezca el bit de salida :013/01.

Mask Compare Equal to (MEQ)

Descripción:



La instrucción MEQ es una instrucción de entrada que compara un valor de una dirección de origen con datos en una dirección de comparación y permite que se enmascaren porciones de los datos. Si los datos en la dirección de origen coinciden con los datos en la dirección de comparación bit a bit (menos los bits con máscara), la instrucción es verdadera. La instrucción se hace falsa cuando detecta un desigualdad.

Puede usar la instrucción MEQ para extraer datos de bits (para comparación) tales como los bits de estado o control desde un elemento que contiene datos de bits y palabras.

Cómo introducir los parámetros

Es necesario proporcionar al procesador la información siguiente para programar la instrucción MEQ:

Parámetro:	Definición:
Source	una constante de programa o dirección de datos desde la cual la instrucción lee una imagen del valor. El origen no cambia.
Mask	especifica cuáles bits se deben pasar o bloquear. Una máscara pasa los datos cuando los bits de máscara están establecidos (1). Una máscara bloquea los datos cuando los bits de máscara están restablecidos (0). La máscara debe tener el mismo tamaño de elemento (16 bits) que la dirección de origen o comparación. Para comparar los bits, es necesario establecer (1) bits de máscara. Los bits en la dirección de comparación correspondientes a ceros (0) en la máscara no se comparan. Si se desea que el programa de lógica de escalera cambie el valor de la máscara, hay que almacenar la máscara en una dirección de datos. De lo contrario, es necesario introducir un valor hexadecimal para un valor de máscara de constante. Si se introduce un valor hexadecimal que comienza con una letra (tal como F800), hay que introducir el valor con un cero inicial. Por ejemplo: 0F800
Compare	especifica si se desea que el programa de lógica de escalera varíe el valor de comparación o una constante de programa para una referencia fija. Use los elementos de 16 bits que son los mismos que la fuente.

Ejemplo:

```

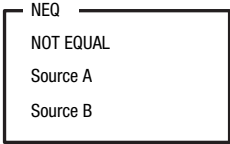
Source      01010101 01011111
Mask        11111111 11110000
Compare     01010101 0101xxxx
Result      The instruction is true because
              reference bits xxxx are not compared.
  
```



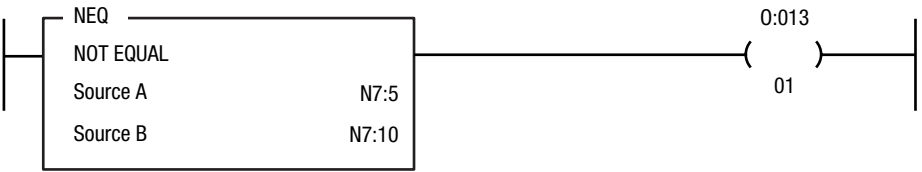
El procesador pasa el valor en N7:5 por la máscara en N7:6. Luego, pasa el valor en N7:10 por la máscara en N7:6. Si los dos valores enmascarados son iguales, establezca el bit de salida 0:013/01.

Not Equal to (NEQ)

Descripción: Use la instrucción NEQ para probar si dos valores no son iguales. La fuente A y la fuente B pueden ser valores o direcciones.



Ejemplo:



Si el valor en N7:5 no es igual al valor en N7:10, establezca el bit 0:013/01.

Instrucciones de cálculo CPT, ACS, ADD, ASN, ATN, AVE, CLR, COS, DIV, LN, LOG, MUL, NEG, SIN, SRT, SQR, STD, SUB, TAN, XPY

Cómo usar las instrucciones de cálculo

Las instrucciones de cálculo evalúan las operaciones aritméticas mediante una expresión o una instrucción aritmética específica. La tabla 4.A indica las instrucciones de cálculo disponibles.

Tabla 4.A
Instrucciones de cálculo disponibles

Si desea:	Use esta instrucción:	Que se encuentra en la página
Evaluar una expresión	CPT	4-5
Hallar el arco coseno de un número	ACS*	4-11
Sumar dos valores	ADD	4-12
Hallar el arco seno de un número	ASN*	4-13
Hallar la arco tangente de un número	ATN*	4-14
Calcular el promedio para un conjunto de valores	AVE*	4-15
Borrar una palabra de dirección (poner todos los bits en cero)	CLR	4-17
Hallar el coseno de un número	COS*	4-18
Dividir dos valores	DIV	4-19
Hallar el logaritmo natural de un número	LN*	4-20
Hallar el logaritmo de un número	LOG*	4-21
* Solamente los procesadores PLC-5 con características mejoradas son compatibles con esta instrucción		

(Continúa)

Si desea:	Use esta instrucción:	Que se encuentra en la página
Multiplicar dos valores	MUL	4-22
Hallar el signo opuesto de un valor	NEG	4-23
Hallar el seno de un número	SIN*	4-24
Hallar la raíz cuadrada de un valor	SQR	4-25
Organizar un conjunto de valores en orden ascendente	SRT*	4-26
Calcular la desviación estándar para un conjunto de valores	STD*	4-28
Restar dos valores	SUB	4-31
Hallar la tangente de un número	TAN*	4-32
Elevar un número a una potencia	XPY*	4-33
* Solamente los procesadores PLC-5 con características mejoradas son compatibles con esta instrucción		

Vea el Apéndice C para obtener más información acerca de los operandos (y los tipos/valores válidos de cada operando) usados por las instrucciones instrucciones descritas en este capítulo.

Cómo usar los indicadores de estado aritméticos

Los indicadores de estado aritméticos se encuentran en los bits 0-3 de la palabra 0 en el archivo de estado del procesador (S). La tabla 4.B indica los bits de estado:

Tabla 4.B
Bits de estado aritméticos

Este bit:	Descripción:
S:0/0	Acarreo (C)
S:0/1	Overflow (V)
S:0/2	Cero (Z)
S:0/3	Signo (S)

Tipos de datos y la instrucción de cálculo

Usted puede calcular los valores de diferentes tipos de datos tales como punto flotante (coma flotante) y número entero. Si usa un valor de punto flotante (coma flotante) como el origen, use un valor de punto flotante (coma flotante) como destino. De lo contrario, el valor de destino será redondeado.

Debe usar los valores BCD y ASCII para fines de visualización. Si introduce los valores BCD o ASCII, el procesador considera dichos valores como enteros.

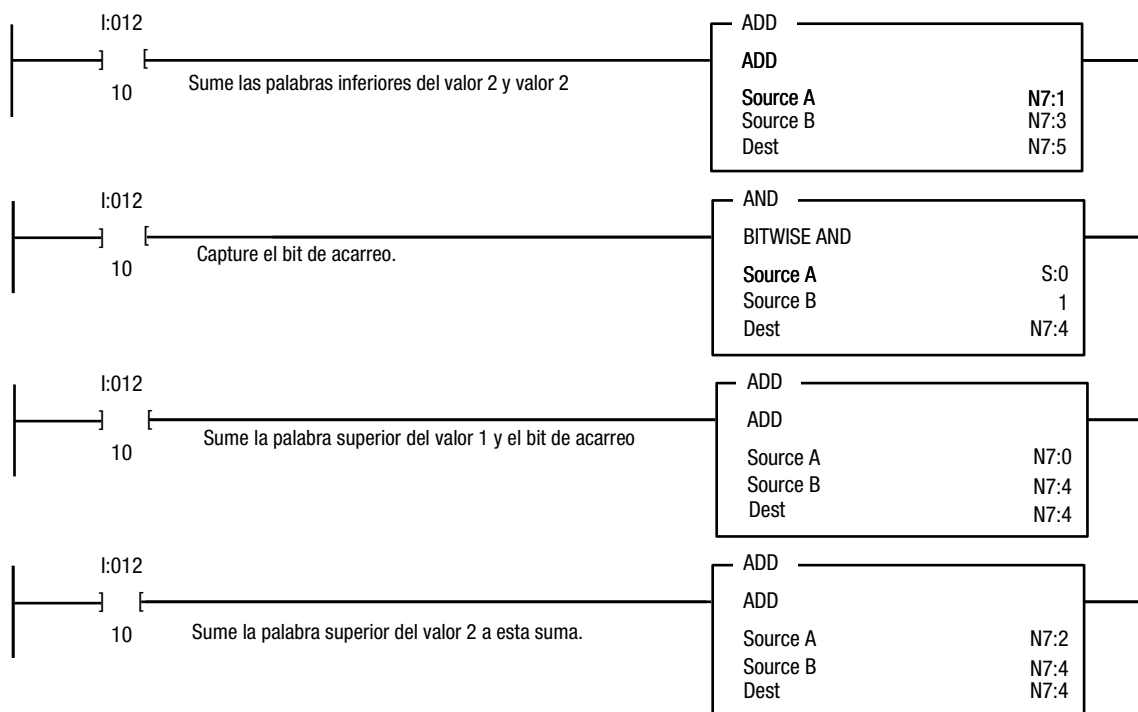
Los parámetros que se introducen son constantes de programa o direcciones lógicas de los valores deseados.

Si usa este procesador:	El procesador redondea:
PLC-5 clásico	el valor final de una operación matemática antes de almacenar el resultado final. El procesador redondea al número entero más cercano. El procesador redondea los valores de 0.5-0.9 al próximo número entero. El procesador redondea los valores de 0.1-0.4 hacia abajo al número entero más cercano. Si este valor es mayor que 32,767 ó menor que -32,768, el bit de estado de overflow se establece.
PLC-5 con características mejoradas	hacia abajo si el valor es <0.5, hacia arriba si el valor es >0.5 y al número par más cercano si el valor es = 0.5. Si este valor es mayor que 32,767 ó menor que -32,768, el procesador regresa al valor negativo (32,767, -32,768, -32,767, -327,66, etc.). Por ejemplo, si tiene una instrucción ADD con un resultado mayor que 32,767, el bit de overflow se establece, el bit de signo se establece y el resultado es negativo: $32,767 + 5 = -32,764$.

Importante: Si usa un procesador PLC-5 con características mejoradas y una operación aritmética genera un overflow, los bits superiores se pierden, pero los bits inferiores son correctos. Si realiza una operación lógica en la palabra inferior (Y u O), puede obtener el resultado correcto. Además, el uso del bit de acarreo le permite hacer aritmética de palabras múltiples (por ejemplo, sumar dos palabras de 32 bits).

Por ejemplo, si: el valor 1 = N7:0 y N7:1
 el valor 2 = N7:2 y N7:3
 el resultado = N7:4 y N7:5

y desea sumar el valor 1 al valor 2, el programa de lógica de escalera es:



Cómo usar los tipos de datos de punto flotante (coma flotante)

Si usa los tipos de datos de punto flotante (coma flotante) para un procesador PLC-5 con características mejoradas y el resultado es demasiado grande o si no se define (por ejemplo, un logaritmo natural de 0), el procesador establece el bit de overflow.

Si el resultado es demasiado grande, se muestra **!+INF!**. Si el resultado es demasiado pequeño, se muestra **!-INF!**. Si el valor no es un número, se muestra **!NAN!**.

Importante: Si usa el punto flotante (coma flotante) y el número es mayor que 32,767 ó menor que -32,768, debe usar un punto decimal. Si no usa un punto decimal, aparece el error **INVALID OPERAND**.

Cuando usa expresiones complejas y cualquier operando es un punto flotante (coma flotante), toda la expresión se evalúa como un punto flotante (coma flotante). Vea el ejemplo en la sección de “Ejemplos de expresiones” en la página 4-10 para obtener más información.

Compute (CPT)

Descripción:

CPT
COMPUTE
Destination
Expression

La instrucción CPT realiza operaciones de copiar, aritméticas, lógicas y de conversión.

La instrucción CPT es una instrucción de salida que realiza las operaciones que usted define en la expresión y escribe el resultado en la dirección de destino. La instrucción CPT también puede copiar datos de una dirección a otra y convierte automáticamente el tipo de datos ubicado en la dirección de origen en el tipo de datos que usted especifica en la dirección de destino.

El tiempo de ejecución de una instrucción CPT es más largo que el tiempo de ejecución de una instrucción aritmética, lógica o de mover (por ejemplo, ADD, AND, MOV, etc.). La instrucción CPT también usa más palabras en su archivo de programa.

Después de realizarse cada instrucción CPT, los bits de estado aritméticos en el archivo de datos de la tabla de datos se actualizan de la misma manera que las instrucciones correspondientes aritméticas, lógicas o de mover. Por ejemplo, vea la descripción de la instrucción ADD para saber cómo los bits de estado se actualizan después de la ejecución de una instrucción (CPT) (sumar).

Cómo introducir la expresión CPT

La expresión define las operaciones que desea realizar. La expresión se define mediante operadores o direcciones o constantes de programa. Los procesadores PLC-5 con características mejoradas le permiten introducir expresiones complejas. La tabla 4.C indica las operaciones válidas para una expresión. La lista siguiente proporciona pautas para escribir las expresiones:

- Los operadores (símbolos) definen las operaciones
- Las direcciones pueden ser direcciones lógicas directas o indirectas (deben ser elementos o nivel de bit)
- Los procesadores PLC-5 con características mejoradas permiten que las constantes de programa sean números enteros o de punto flotante (coma flotante) (si se introducen valores octales, hay que usar un &O precedente. Si se introducen valores hexadecimales, hay que usar un &H precedente)
- Las expresiones pueden tener un total de sólo 80 caracteres, incluso espacios y paréntesis

Tabla 4.C
Operaciones válidas para uso en una expresión CPT

Tipo	Operador	Descripción	Ejemplo de operación
Copiar	ninguno	copiar de A a B	enter source address in the expression enter destination address in destination
Borrar	ninguno	poner un valor en cero	0 (enter 0 for the expression)
Aritmético	+	sumar	2 + 3 2 + 3 + 7 (Enhanced PLC-5 processors)
	–	restar	12 – 5 (12 – 5) – 7 (Enhanced PLC-5 processors)
	*	multiplicar	5 * 2 6 * (5 * 2) (Enhanced PLC-5 processors)
	I (barra vertical)	dividir	24 I 6 (24 I 6) * 2 (Enhanced PLC-5 processors)
	–	cambiar signo	– N7:0
	SQR	raíz cuadrada	SQR N7:0
	**	exponencial* (x a la potencia de y)	10**3
	LN	logaritmo natural*	LN F8:20
	LOG	logaritmo a la base 10	LOG F8:3
Trigono- métrico	ACS	arco coseno*	ACS F8:18
	ASN	arco seno*	ASN F8:20
	ATN	arco tangente*	ATN F8:22
	COS	coseno*	COS F8:14
	SIN	seno*	SIN F8:12
	TAN	tangente*	TAN F8:16
Bit a bit	AND	función Y	D9:3 AND D10:4
	OR	función O	D10:4 OR D10:5
	XOR	función O exclusivo	D9:5 XOR D10:4
	NOT	complemento bit a bit	NOT D9:3
Conversión	FRD	convertir de BCD en binario	FRD N7:0
	TOD	convertir de binario en BCD	TOD N7:0
	DEG	convertir radianes en grados*	DEG F8:8
	RAD	convertir grados en radianes*	RAD F8:10
* Disponible en los procesadores PLC-5 con características mejoradas solamente.			

Cómo determinar la longitud de una expresión

Los procesadores PLC-5 con características mejoradas le permiten introducir instrucciones complejas (hasta un total de 80 caracteres, incluso espacios y paréntesis). Según el operador, el procesador inserta caracteres antes o después del operador en la expresión para formatear la expresión a fin de facilitar la interpretación. Use la tabla 4.D a continuación para determinar el número de caracteres que usa cada operador en una expresión.

La instrucción CPT permite mostrar en pantalla un máximo de 80 caracteres de la expresión. Si la expresión introducida se acerca a este máximo de 80 caracteres, el procesador puede ampliar dicho máximo cuando se acepta el renglón que contiene la instrucción. Cuando usted intenta editar la expresión, se muestran en pantalla solamente los 80 primeros caracteres y el renglón se muestra como un renglón de error. Sin embargo, el procesador contiene la expresión completa y la instrucción funciona correctamente.

Para evitar este problema de visualización en pantalla, exporte el archivo de memoria del procesador y efectúe las ediciones en el archivo de texto PC5. Luego importe este archivo de texto.

Importante: No puede introducir números de punto flotante (coma flotante) en una notación científica con exponentes negativos en expresiones complejas. En lugar de ello, use la equivalencia decimal o ponga el número en un archivo de punto flotante (coma flotante) y use la dirección de datos en la expresión compleja.

Tabla 4.D
Longitudes de caracteres para los operadores

Esta operación	Con el uso de este operador:	Usa este número de caracteres:
binaria matemática	+, -, *,	3
	OR, **	4
	AND, XOR	5
matemática de un dígito	- (cambiar signo)	2
	LN *	3
	FRD, TOD, DEG*, RAD*, SQR, NOT, LOG*, SIN*, COS*, TAN*, ASN*, ACS*, ATN*	4
comparativa	=, <, >	3
	<>, <=, >=	4
* Disponible en los procesadores PLC-5 con características mejoradas solamente.		

Cómo determinar el orden de operación

El procesador realiza en un orden indicado las operaciones que usted escribe en la expresión y no según el orden en que las escribió. Puede anular el orden de operación si coloca los términos entre paréntesis, lo cual fuerza que el procesador realice la operación entre paréntesis antes de las otras operaciones.

Las operaciones de orden igual se realizan desde la izquierda hacia la derecha. La expresión usada debe incluir un operador. La tabla 4.E muestra el orden de operación.

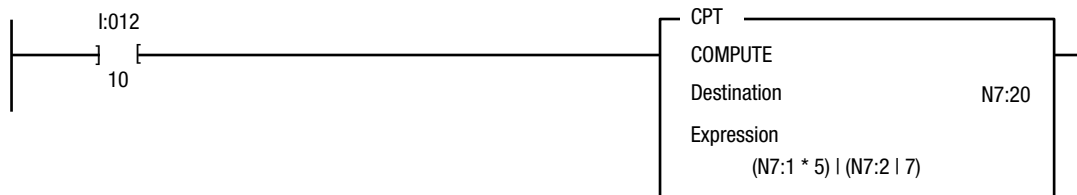
Tabla 4.E
Orden de operación de expresiones CPT

Orden	Operación	Descripción
1	**	exponencial (X^Y) procesadores PLC-5 con características mejoradas
2	–	cambiar signo
	NOT	complemento bit a bit
3	*	multiplicar
		dividir
4	+	sumar
	–	restar
5	AND	función Y
6	XOR	función O exclusivo
7	OR	función O

Ejemplos de expresiones

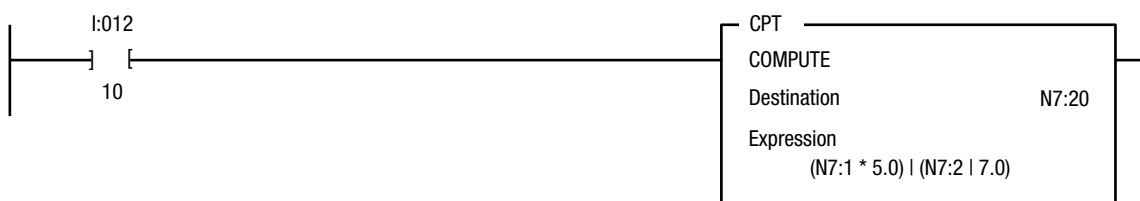
Valor único: la expresión SQR (N7:4) con el destino N7:20 indica al procesador que halle la raíz cuadrada del valor almacenado en N7:4 y almacene el resultado en N7:20.

Valores múltiples: los procesadores PLC-5 con características mejoradas le permiten usar funciones para operar uno o más valores en la expresión (expresiones complejas) para las operaciones de cálculo y comparación. Las expresiones complejas pueden tener una longitud de hasta 80 caracteres (los espacios y paréntesis se consideran como caracteres). Por ejemplo, se puede introducir una expresión tal como:

Ejemplo:

Si el bit 10 de la palabra de entrada 12 está establecido, multiplique el valor de N7:1 por 5. Divida este resultado entre el cociente de N7:2 dividido entre 7. Si N7:1 = 5 y N7:2 = 9, el resultado es 25. (El resultado se redondea al número entero más cercano puesto que las constantes 5 y 7 se especificaron como números enteros.)

Cuando se usan expresiones complejas, si cualquier operando es un valor de punto flotante (coma flotante), toda la expresión se evalúa como un valor de punto flotante (coma flotante):

Ejemplo:

Si el bit 10 de la palabra de entrada 12 está establecido, multiplique el valor de N7:1 por 5. Divida este resultado entre el cociente de N7:2 dividido entre 7. Si N7:1 = 5 y N7:2 = 9, el resultado es 25. (El resultado se redondea diferentemente puesto que las constantes 5.0 y 7.0 se especificaron a 1 lugar decimal.)

Cómo introducir el destino

Introduzca una dirección lógica directa o indirecta para el destino. La instrucción almacena el resultado de la operación en la dirección de destino.

Importante: El procesador convierte automáticamente el tipo de datos especificado por la dirección de origen al tipo de datos especificado por la dirección de destino. El procesador usa BCD para la muestra en pantalla o compatibilidad con los procesadores de la familia PLC-2. Tiene que programar las conversiones BCD.

Cómo usar las funciones CPT

Use las funciones para operar uno o más valores en la expresión de una instrucción CPT para realizar estos tipos de operaciones:

- convertir de un formato numérico en otro
- manipular los números
- realizar funciones trigonométricas

La instrucción realiza la función que usted especifica según un mnemónico. Cuando introduzca esta expresión, introduzca el mnemónico como prefijo de la dirección del valor que desea operar o como prefijo del valor mismo cuando se introduzca como constante de programa.

Importante: Los números de punto flotante (coma flotante) son valores de 32 bits. Los números enteros son valores de 16 bits. La instrucción convierte automáticamente los tipos de datos encontrados en la expresión al tipo de datos especificado por la dirección de origen.



ATENCION: Si la expresión o las direcciones de destino requieren la conversión de datos de 32 bits en 16 bits y el valor es demasiado grande, el procesador establece un bit de overflow en S:0/1 y establece un fallo menor (S10:14). El valor erróneo resultante puede conducir a una situación peligrosa. Monitoree este bit en el programa de lógica de escalera.

La tabla 4.F indica las funciones CPT que puede usar.

Tabla 4.F
Funciones CPT para la conversión de números

Mnemónico	Título	Descripción
RAD *	radianes	Convierte de grados en radianes
DEG *	grados	Convierte de radianes en grados
TOD	en BCD	Convierte de número entero en BCD (compatible con los números BCD de 4 dígitos)
FRD	de BCD	Convierte de BCD en número entero (compatible con los números BCD de 4 dígitos)
SQR	raíz cuadrada	Halla la raíz cuadrada del número. Preciso a 6 dígitos significativos
LOG *	—	Logaritmo a la base 10. Preciso a 6 dígitos significativos
LN *	—	Algoritmo natural. Preciso a 6 dígitos significativos
SIN *	seno; manipulado en radianes.	Preciso a 6 dígitos significativos
COS *	coseno; manipulado en radianes.	Preciso a 6 dígitos significativos
TAN *	tangente; manipulado en radianes.	Preciso a 6 dígitos significativos
ASN *	arco seno; manipulado en radianes.	Preciso a 6 dígitos significativos
ACS *	arco coseno; manipulado en radianes.	Preciso a 6 dígitos significativos
ATN *	arco tangente; manipulado en radianes.	Preciso a 6 dígitos significativos

* Disponible en los procesadores PLC-5 con características mejoradas solamente.

Puede usar las funciones aritméticas CPT anteriores dentro de las expresiones o como instrucciones autónomas. Vea las instrucciones individuales descritas en este capítulo.

Arc Cosine (ACS) (procesadores PLC-5 con características solamente)

Descripción:

ACS
ARCCOSINE
Source
Destination

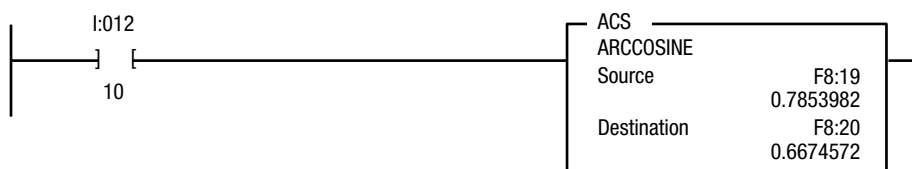
Use la instrucción ACS para hallar el arco coseno del origen (en radianes) y almacenar el resultado (en radianes) en el destino. La tabla 4.G muestran los indicadores de estado para la instrucción ACS.

El origen debe ser mayor o igual a -1 y menor o igual a 1. Si no se encuentra en este rango, el procesador devuelve un resultado !NAN! en el destino. El valor resultante en el destino siempre es mayor o igual a 0 y menor o igual a π (donde $\pi = 3.141592$).

Tabla 4.G
Cómo actualizar los indicadores aritméticos para una instrucción ACS

Con este bit:	El procesador:
Acarreo (C)	siempre se restablece
Overflow (V)	se establece si se genera un overflow; de lo contrario, se restablece
Cero (Z)	se establece si el resultado es cero; de lo contrario, se restablece
Signo (S)	siempre se restablece

Ejemplo:



Si el bit 10 de la palabra de entrada 12 está establecido, halle el arco coseno del valor en F8:19 y almacene el resultado en F8:20.

Addition (ADD)

Descripción:

ADD

ADD

Source A

Source B

Destination

Use la instrucción ADD para sumar un valor (origen A) y otro valor (origen B) y colocar el resultado en el destino. El origen A y el origen B pueden ser valores o direcciones que contienen valores. La tabla 4.H muestra los indicadores de estado para la instrucción ADD.

Importante: La instrucción ADD se ejecuta una vez durante cada escán siempre que el renglón sea verdadero. Si prefiere sumar los valores una sola vez, incluya el comando ONS (vea el capítulo 13).

Tabla 4.H
Actualización de los indicadores aritméticos para una instrucción ADD

Con este bit:	El procesador:
Acarreo (C)	se establece si se genera el acarreo; de lo contrario, se restablece
Overflow (V)	se establece si se genera un overflow; de lo contrario, se restablece
Cero (Z)	se establece si el resultado es cero; de lo contrario, se restablece
Signo (S)	se establece si el resultado es negativo; de lo contrario, se restablece

Ejemplo:



Si el bit 10 de la palabra de entrada 12 está establecido, sume el valor en N7:3 y el valor en N7:4 y almacene el resultado en N7:20.

Arc Sine (ASN) (procesadores PLC-5 con características mejoradas solamente)

Descripción:

ASN
ARCSINE
Source
Destination

Use la instrucción ASN para hallar el arco seno del origen (en radianes) y almacenar el resultado (en radianes) en el destino. La tabla 4.I muestra los indicadores de estado para la instrucción ASN.

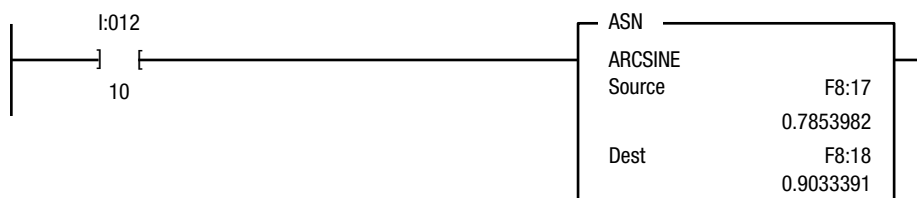
El origen debe ser mayor o igual a -1 y menos o igual a 1. Si no se encuentra dentro de este rango, el procesador devuelve un resultado !NAN! en el destino. El valor resultante en el destino siempre es mayor o igual a $-\pi/2$ y menos o igual a $\pi/2$ (donde $\pi = 3.141592$).

Tabla 4.I

Actualización de los indicadores de estado aritméticos para una instrucción ASN

Con este bit:	El procesador:
Acarreo (C)	siempre se restablece
Overflow (V)	se establece si se genera un overflow; de lo contrario, se restablece
Cero (Z)	se establece si el resultado es cero; de lo contrario, se restablece
Signo (S)	siempre se establece

Ejemplo:



Si el bit 10 de la palabra de entrada 12 está establecido, halle el arco seno del valor en F8:17 y almacene el resultado en F8:18.

Arc Tangent (ATN) (procesadores PLC-5 con características mejoradas solamente)

Descripción:

ATN
ARCTANGENT
Source
Destination

Use la instrucción ATN para hallar la arco tangente del origen (en radianes) y almacenar el resultado (en radianes) en el destino. El valor resultante en el destino siempre es mayor o igual a $-\pi/2$ y menor o igual a $\pi/2$ (donde $\pi = 3.141592$). La tabla 4.J muestra los indicadores de estado para la instrucción ATN.

Tabla 4.J
Actualización de los indicadores de estado aritméticos para una instrucción ATN

Con este bit:	El procesador:
Acarreo (C)	siempre se restablece
Overflow (V)	se establece si se genera un overflow; de lo contrario, se restablece
Cero (Z)	se establece si el resultado es cero; de lo contrario, se restablece
Signo (S)	se establece si el resultado es negativo; de lo contrario, se restablece

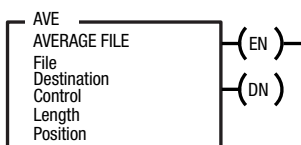
Ejemplo:



Si el bit 10 de la palabra de entrada 12 está establecido, halle la arco tangente del valor F8:21 y almacene el resultado en F8:22.

Average File (AVE) (procesadores PLC-5 con características mejoradas solamente)

Descripción:



La instrucción AVE calcula el promedio de un conjunto de valores. Cuando el renglón va de falso a verdadero, el valor en la posición actual se suma al próximo valor, el cual se suma al próximo valor, y así sucesivamente. La tabla 4.K muestra los indicadores de estado para la instrucción AVE.

Cada vez que otro valor se suma, el campo de posición y la palabra de estado (S:24) se incrementan. La suma final se divide entre el número de valores sumados y el resultado se almacena en el destino.

Tabla 4.K

Actualización de los indicadores de estado aritméticos para una instrucción AVE

Con este bit:	El procesador:
Acarreo (C)	siempre se restablece
Overflow (V)	se establece si se genera un overflow; de lo contrario, se restablece
Cero (Z)	se establece si el resultado es cero; de lo contrario, se restablece
Signo (S)	se establece si el resultado es negativo; de lo contrario, se restablece

Un overflow puede ocurrir si:

- la suma intermedia excede el valor máximo del punto flotante (coma flotante)
- el destino es una dirección entera y el valor final es mayor que 32,767 ó menor que -32,768

Si un overflow ocurre, el procesador detiene el cálculo, establece el bit .ER y el destino no cambia. La posición identifica el elemento que causó el overflow. Cuando se restablece el bit .ER, la posición se pone en cero y el promedio se vuelve a calcular.

Importante: Use la instrucción RES para borrar los indicadores de estado.

Cómo introducir los parámetros

Es necesario proporcionar al procesador la información siguiente para programar la instrucción AVE:

- File** es la dirección que contiene el primer valor a sumar. Esta dirección puede ser un valor de punto flotante (coma flotante) o un número entero.
- Destination** es la dirección donde se almacena el resultado de la instrucción. Esta dirección puede ser un valor de punto flotante (coma flotante) o un número entero.
- Control** es la dirección de la estructura de control en la zona de control (R) de la memoria del procesador. El procesador almacena información tal como la longitud, la posición y el estado y usa esta información para ejecutar la instrucción.
- Length** es el número de palabras en el archivo (1-1000).
- Position** indica la palabra que la instrucción usa actualmente.

Cómo usar los bits de estado

Para usar la instrucción AVE correctamente, examine los bits de estado en la estructura de control. Direccione estos bits mediante mnemónicos.

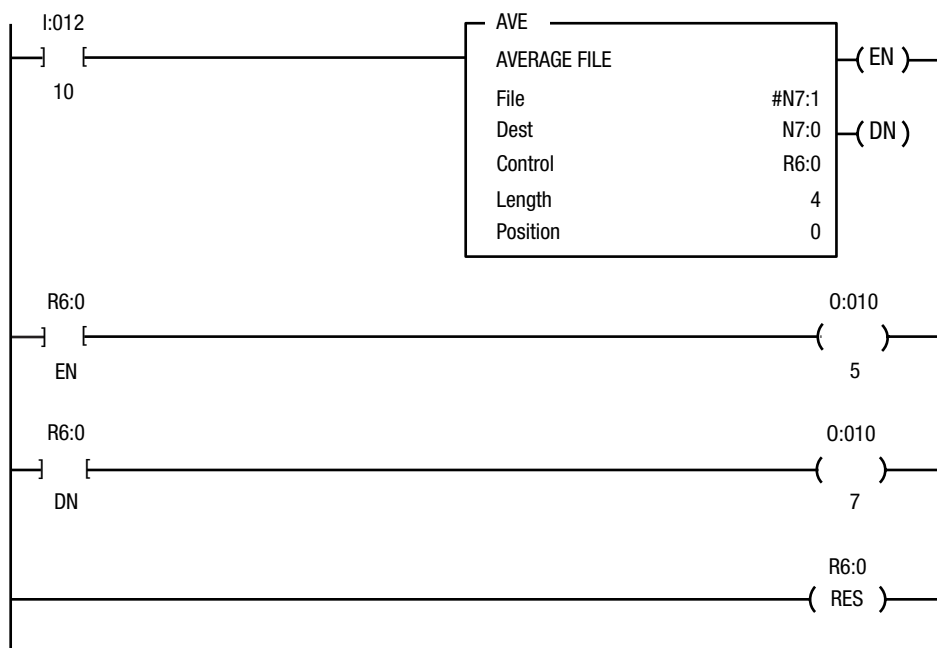
Este bit:	Se establece:
Habilitación .EN (bit 15)	durante una transición de falso a verdadero del renglón para indicar que la instrucción está habilitada. La instrucción sigue a la condición del renglón.
Efectuado .DN (bit 13)	después de que la instrucción termina la operación. Después de que el renglón se hace falso, el procesador restablece el bit de efectuado .DN durante la próxima transición de falso a verdadero del renglón.
Error .ER (bit 11)	cuando la operación genera un overflow. La instrucción se detiene hasta que el programa de lógica de escalera restablece el bit .ER.

Importante: La instrucción AVE calcula el promedio usando punto flotante (coma flotante) independientemente del tipo especificado para los parámetros de archivo o destino.



ATENCION: La instrucción AVE incrementa el valor de offset almacenado en S:24. Asegúrese de monitorear o cargar el valor de offset que desea antes de usar una dirección indexada. De lo contrario, es posible que ocurra una operación inesperada de la máquina, lo cual puede causar daños al equipo y/o lesiones corporales.

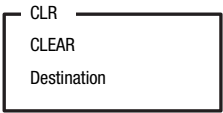
Ejemplo:



Si el bit 10 de la palabra de entrada 12 está establecido, la instrucción AVE está habilitada. Los valores en N7:1, N7:2, N7:3 y N7:4 se suman y dividen entre 4. El resultado se almacena en N7:0. Cuando el cálculo se completa, el bit de salida 7 de la palabra 10 se establece. Luego la instrucción RES restablece los bits de estado del archivo de control R6:0.

Clear (CLR)

Descripción:

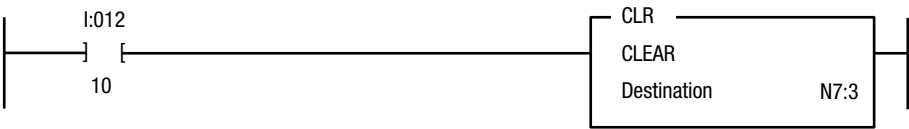


Use la instrucción CLR para poner a cero todos los bits de una palabra. El destino debe ser una dirección de palabra. La tabla 4.L muestra los indicadores de estado para la instrucción CLR.

Tabla 4.L
Actualización de los indicadores de estado aritméticos para una instrucción CLR

Con este bit:	El procesador:
Acarreo (C)	siempre se restablece
Overflow (V)	siempre se restablece
Cero (Z)	siempre se restablece
Signo (S)	siempre se restablece

Ejemplo:



Si el bit 10 de la palabra de entrada 12 está establecido, ponga a cero todos los bits en N7:3.

Cosine (COS) (procesadores PLC-5 con características mejoradas solamente)

Descripción:

COS
COSINE
Source
Destination

Use la instrucción COS para hallar el coseno de un número (origen en radianes) y almacenar el resultado en el destino. La tabla 4.M muestra los indicadores de estado para la instrucción COS.

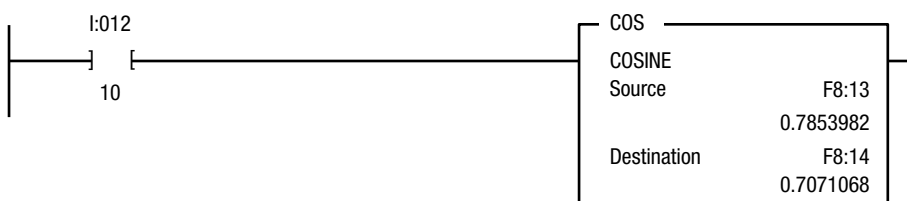
El origen debe ser mayor o igual a -205887.4 y menor o igual a 205887.4. Si no se encuentra dentro de este rango, el procesador devuelve un resultado ! INF ! en el destino. El valor resultante en el destino siempre es mayor o igual a -1 ó menor o igual a 1.

Importante: Para obtener mejor precisión, el dato del origen debe ser mayor o igual a -2π y menos o igual a 2π .

Tabla 4.M
Actualización de los indicadores de estado aritméticos
para una instrucción COS

Con este bit:	El procesador:
Acarreo (C)	siempre se restablece
Overflow (V)	se establece si se genera un overflow; de lo contrario, se restablece
Cero (Z)	se establece si el resultado es cero; de lo contrario, se restablece
Signo (S)	se establece si el resultado es negativo; de lo contrario, se restablece

Ejemplo:



Si el bit 10 de la palabra de entrada 12 está establecido, halle el coseno del valor en F8:13 y almacene el resultado en F8:14.

Divide (DIV)

Descripción:

DIV
DIVIDE
Source A
Source B
Destination

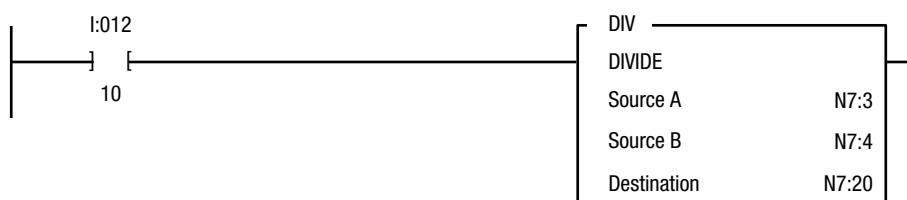
Use la instrucción DIV para dividir un valor (origen A) en otro valor (origen B) y colocar el resultado en el destino. El origen A y el origen B pueden ser valores o direcciones que contienen valores. La tabla 4.N muestra los indicadores de estado para la instrucción DIV.

Importante: Las instrucciones de cálculo se ejecutan para cada escán siempre que el renglón sea verdadero. Si desea que los valores se calculen una sola vez, incluya el comando ONS (vea el capítulo 13).

Tabla 4.N
Actualización de los indicadores de estado aritméticos para una instrucción DIV

Con este bit:	El procesador:
Acarreo (C)	siempre se restablece
Overflow (V)	se establece si se trata de división entre 0 ó si se genera un overflow; de lo contrario, se restablece
Cero (Z)	se establece si el resultado es cero; de lo contrario, se restablece; está no definido si el overflow está establecido
Signo (S)	se establece si el resultado es negativo; de lo contrario, se restablece; no está definido si el overflow está establecido

Ejemplo:



Si el bit 10 de la palabra de entrada 12 está establecido, divida el valor en N7:3 entre el valor en N7:4 y almacene el resultado en N7:20.

Natural Log (LN) (procesadores PLC-5 con características mejoradas solamente)

Descripción:

LN
NATURAL LOG
Source
Destination

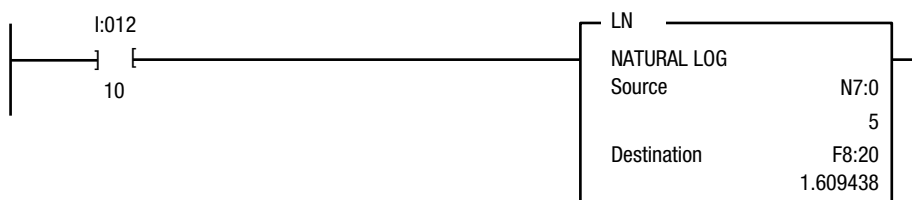
Use la instrucción LN para hallar el algoritmo del valor en el origen y almacenar el resultado en el destino. La tabla 4.O muestra los indicadores de estado para la instrucción LN.

Si el origen es igual a 0, el resultado en el destino será ! - INF !. Si el valor en el origen es menor de 0, el resultado en el destino será ! NAN !. El valor resultante en el destino siempre es mayor o igual a -87.33655 y menor o igual a 88.72284.

Tabla 4.0
Actualización de los indicadores de estado aritméticos
para una instrucción LN

Con este bit:	El procesador:
Acarreo (C)	siempre se restablece
Overflow (V)	se establece si se genera un overflow; de lo contrario, se restablece
Cero (Z)	se establece si el resultado es cero; de lo contrario, se restablece
Signo (S)	se establece si el resultado es negativo; de lo contrario, se restablece

Ejemplo:



Si el bit 10 de la palabra de entrada 12 está establecido, halle el algoritmo natural del valor en N7:0 y almacene el resultado en F8:20.

Log to the Base 10 (LOG) (procesadores PLC-5 con características mejoradas solamente)

Descripción:

LOG
LOG BASE 10
Source
Destination

Use la instrucción LOG para hallar la base 10 del algoritmo de valor en el origen y almacene el resultado en el destino. La tabla 4.P muestra los indicadores de estado para la instrucción LOG.

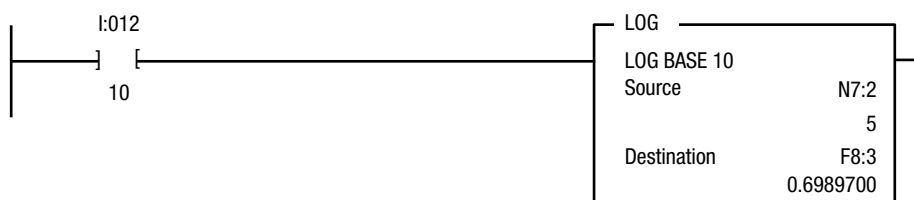
Si el origen es igual a 0, el resultado en el destino será ! - INF !. Si el valor en el origen es menor que 0, el resultado en el destino será ! NAN !. El valor resultante en el destino siempre es mayor o igual a -37.92978 y menor o igual a 38.53184.

Tabla 4.P

Actualización de los indicadores de estado aritméticos para una instrucción LOG

Con este bit:	El procesador:
Acarreo (C)	siempre se restablece
Overflow (V)	se establece si se genera un overflow; de lo contrario, se restablece
Cero (Z)	se establece si el resultado es cero; de lo contrario, se restablece
Signo (S)	se establece si el resultado es negativo; de lo contrario, se restablece

Ejemplo:



Si el bit 10 de la palabra de entrada 12 está establecido, halle la base 10 de algoritmo del valor en N7:2 y almacene el resultado en F8:3.

Multiply (MUL)

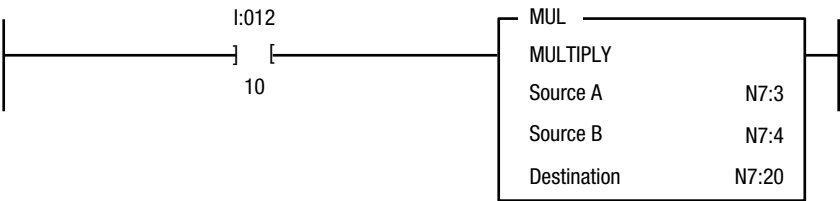
Descripción: Use la instrucción MUL para multiplicar un valor (origen A) por otro valor (origen B) y colocar el resultado en el destino. El origen A y el origen B pueden ser valores o direcciones. La tabla 4.Q muestra los indicadores de estado para la instrucción MUL.

MUL
MULTIPLY
Source A
Source B
Destination

Tabla 4.Q
Actualización de los indicadores de estado aritméticos para una instrucción MUL

Con este bit:	El procesador:
Acarreo (C)	siempre se restablece
Overflow (V)	se establece si se genera un overflow; de lo contrario, se restablece
Cero (Z)	se establece si el resultado es cero; de lo contrario, se restablece
Signo (S)	se establece si el resultado es negativo; de lo contrario, se restablece

Ejemplo:



Si el bit 10 de la palabra de entrada 12 está establecido, multiplique el valor en N7:3 por el valor en N7:4 y almacene el resultado en N7:20.

Negate (NEG)

Descripción:

NEG
NEGATE
Source
Destination

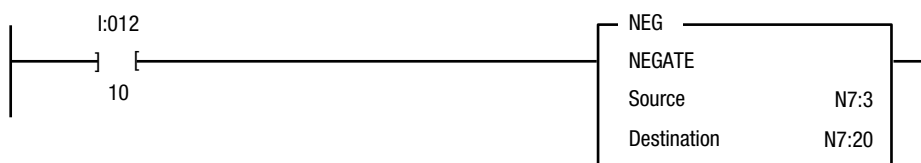
Use la instrucción NEG para cambiar el signo de un valor. Si usted cambia el signo de un valor negativo, el resultado es positivo. Si cambia el signo de un valor positivo, el resultado es negativo. La tabla 4.R muestra los indicadores de estado para la instrucción NEG.

Importante: Las instrucciones de cálculo se ejecutan para cada escán siempre que el renglón sea verdadero. Si desea que los valores se calculen una sola vez, incluya el comando ONS (vea el capítulo 13).

Tabla 4.R
Actualización de los indicadores de estado aritméticos
para una instrucción MUL

Con este bit:	El procesador:
Acarreo (C)	siempre se restablece
Overflow (V)	se establece si se genera un overflow; de lo contrario, se restablece
Cero (Z)	se establece si el resultado es cero; de lo contrario, se restablece
Signo (S)	se establece si el resultado es negativo; de lo contrario, se restablece

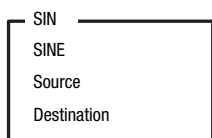
Ejemplo:



Si el bit 10 de la palabra de entrada 12 está establecido, halle el valor opuesto del valor en N7:3 y almacene el resultado en N7:20.

Sine (SIN) (procesadores PLC-5 con características mejoradas solamente)

Descripción:



Use la instrucción SIN para hallar el seno de un número (origen en radianes) y almacenar el resultado en el destino. La tabla 4.S muestra los indicadores de estado para la instrucción SIN.

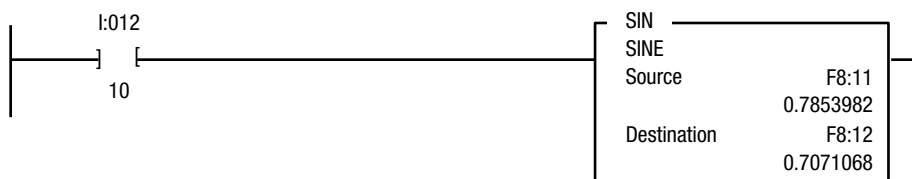
El origen debe ser mayor o igual a -205887.4 y menor o igual a 205887.4. Si no se encuentra dentro de este rango, el procesador devuelve un resultado ! INF ! en el destino. El valor resultante en el destino siempre es mayor o igual a -1 y menor o igual a 1.

Importante: Para obtener mejor precisión, el dato del origen debe ser mayor o igual a -2π y menor o igual a 2π .

Tabla 4.S
Actualización de los indicadores de estado aritméticos
para una instrucción SIN

Con este bit:	El procesador:
Acarreo (C)	siempre se restablece
Overflow (V)	se establece si se genera un overflow; de lo contrario, se restablece
Cero (Z)	se establece si el resultado es cero; de lo contrario, se restablece
Signo (S)	se establece si el resultado es negativo; de lo contrario, se restablece

Ejemplo:



Si el bit 10 de la palabra de entrada 12 está establecido, halle el seno de F8:11 y almacene el resultado en F8:12.

Square Root (SQR)

Descripción:

SQR
SQUARE ROOT
Source
Destination

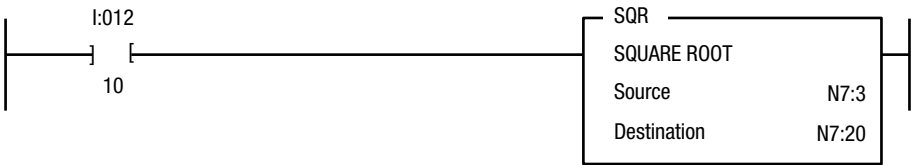
Use la instrucción SQR para extraer la raíz cuadrada de un valor y almacenar el resultado en el destino. El origen puede ser un valor o una dirección. Si el valor del origen es negativo, el procesador halla su valor absoluto para realizar la función de raíz cuadrada. La tabla 4.T muestra los indicadores de estado para la instrucción SQR.

Importante: La instrucción SQR se ejecuta una vez durante cada escán siempre que el renglón sea verdadero. Si desea que los valores se calculen una sola vez, incluya el comando ONS (vea el capítulo 13).

Tabla 4.T
Actualización de los indicadores de estado aritméticos para una instrucción SQR

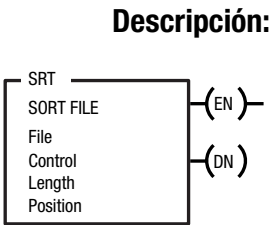
Con este bit:	El procesador:
Acarreo (C)	siempre se restablece
Overflow (V)	se establece si se genera un overflow durante la conversión de punto flotante (coma flotante) en número entero; de lo contrario, se restablece
Cero (Z)	se establece si el resultado es cero; de lo contrario, se restablece
Signo (S)	siempre se restablece

Ejemplo:



Si el bit 10 de la palabra de entrada 12 está establecido, halle la raíz cuadrada del valor en N7:3 y almacene el resultado en N7:20.

Sort File (SRT)
(procesadores PLC-5 con
características mejoradas
solamente)



La instrucción SRT clasifica un conjunto de valores en orden ascendente. Esta instrucción se ejecuta durante una transición de falso a verdadero.

Importante: Asegúrese de que el valor de longitud del archivo especificado en la instrucción no cause que la dirección indexada exceda los límites del archivo. El procesador no verifica esto a no ser que se exceda la zona de memoria del archivo de datos. Si la dirección indexada excede la zona del archivo de datos, el procesador inicia un error de tiempo de ejecución y establece un fallo mayor. El procesador no determina si la dirección indexada cruza los tipos de archivo, tales como N7 a N8.

Cómo introducir los parámetros

Es necesario proporcionar al procesador la siguiente información para programar la instrucción SRT:

Parámetro:	Definición:
archivo	la dirección que contiene el primer valor a clasificar. Esta dirección puede ser valor de un punto flotante (coma flotante) o un número entero.
control	la dirección de la estructura de control en la zona de control (R) de la memoria del procesador. El procesador almacena información, tal como la longitud, la posición y el estado, y usa esta información para ejecutar la instrucción.
longitud	el número de palabras en el archivo (1-1000)
posición	indica el elemento que la instrucción usa actualmente.

Cómo usar los bits de estado

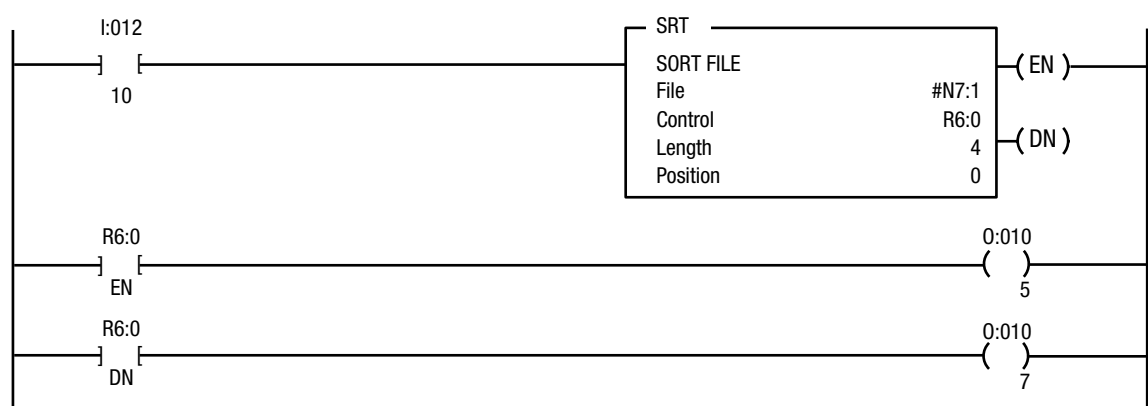
Para usar la instrucción SRT correctamente, el programa de lógica de escalera debe examinar los bits de estado en la estructura de control. Estos bits se direccionan mediante mnemónicos.

Este bit:	Se establece:
Habilitación .EN (bit 15)	durante una transición de falso a verdadero del renglón para indicar que la instrucción está habilitada. La instrucción sigue a la condición del renglón.
Efectuado .DN (bit 13)	después de que la instrucción termina la operación. Después de que el renglón se hace falso, el procesador restablece el bit de efectuado .DN durante la próxima transición de falso a verdadero del renglón.
Error .ER (bit 11)	cuando la operación genera un overflow. La instrucción se detiene hasta que el programa de lógica de escalera restablece el bit .ER.



ATENCION: La instrucción SRT manipula el valor de offset almacenado en S:24. Asegúrese de monitorear o cargar el valor de offset que desea antes de usar una dirección indexada. De lo contrario, es posible que ocurra una operación inesperada de la máquina, lo cual puede causar daños al equipo o lesiones corporales.

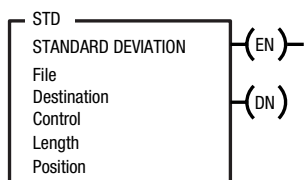
Ejemplo:



Si el bit 10 de la palabra de entrada 12 está establecido, la instrucción SRT está habilitada. Los elementos N7:1, N7:2, N7:3 y N7:4 se clasifican en orden ascendente. Cuando la operación de clasificación se completa, el bit 7 de la palabra de salida 10 se establece.

Standard Derivation (STD) (procesadores PLC-5 con características mejoradas solamente)

Descripción:



La instrucción STD calcula la desviación estándar de un conjunto de valores y almacena el resultado en el destino. Esta instrucción se ejecuta durante una transición de falso a verdadero. La tabla 4.U muestra los indicadores de estado para la instrucción STD.

La desviación estándar se calcula según la fórmula siguiente:

$$\text{Desviación estándar} = \sqrt{\left(\frac{\text{SUM}((x_i - \text{AVE}(x_i))^2)}{(N - 1)} \right)}$$

Donde:

- SUM() – función de suma de las variables encerradas
- AVE () – función de promedio de las variables encerradas
- x_i – elementos variables del archivo de datos
- N – número de elementos en el archivo de datos

Importante: Asegúrese de que el valor de longitud del archivo especificado en la instrucción no cause que la dirección indexada exceda los límites del archivo. El procesador no verifica esto a no ser que se exceda la zona de memoria del archivo de datos. Si la dirección indexada excede la zona del archivo de datos, el procesador inicia un error de tiempo de ejecución y establece un fallo mayor. El procesador no determina si la dirección indexada cruza los tipos de archivo, tales como N7 a N8.

Tabla 4.U
Actualización de los indicadores de estado aritméticos
para una instrucción STD

Con este bit:	El procesador:
Acarreo (C)	siempre se restablece
Overflow (V)	se establece si se genera un overflow; de lo contrario, se restablece
Cero (Z)	se establece si el resultado es cero; de lo contrario, se restablece
Signo (S)	siempre se restablece

Un overflow puede ocurrir de dos maneras:

- la suma intermedia excede el valor del punto flotante (coma flotante) máximo (los valores del punto flotante [coma flotante] son: $\pm 1.1754944^{e-38}$ a $\pm 3.4028237^{e+38}$)
- el destino es una dirección entera y el valor final es mayor que 32,767

Si ocurre un overflow, el procesador detiene el cálculo, establece el bit .ER y no cambia el destino. La posición identifica el elemento que causó el overflow. Cuando se restablece el bit .ER, la posición se pone a 0 y la desviación estándar vuelve a calcularse.

Importante: Use la instrucción RES para restablecer los bits de estado.

Cómo introducir los parámetros

Es necesario proporcionar al procesador la siguiente información para programar la instrucción STD:

Parámetro:	Definición:
archivo	la dirección que contiene el primer valor a clasificar. Esta dirección puede ser un valor de punto flotante (coma flotante) o un número entero.
destino	la dirección donde se almacena el resultado de la instrucción. Esta dirección puede ser un valor de punto flotante (coma flotante) o un número entero.
control	la dirección de la estructura de control en la zona de control (R) de la memoria del procesador. El procesador almacena información, tal como la longitud, la posición y el estado, y usa esta información para ejecutar la instrucción.
longitud	el número de palabras en el archivo (1-1000)
posición	indica el elemento que la instrucción usa actualmente.

Cómo usar los bits de estado

Para usar la instrucción STD correctamente, el programa de lógica de escalera debe examinar los bits de estado en la estructura de control. Estos bits se direccionan mediante mnemónicos.

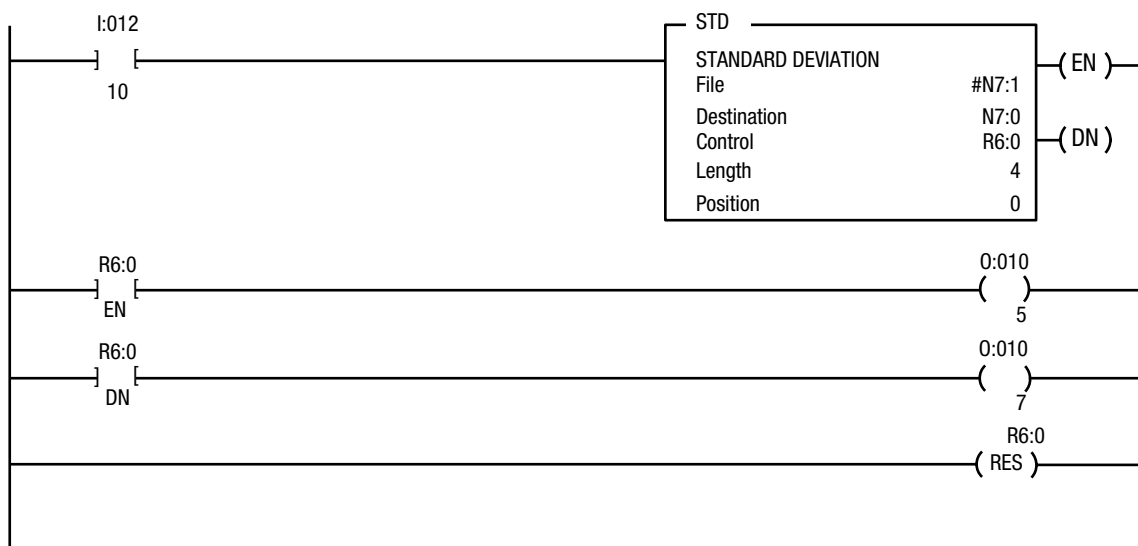
Este bit:	Se establece:
Habilitación .EN (bit 15)	durante una transición de falso a verdadero del renglón para indicar que la instrucción está habilitada. La instrucción sigue a la condición del renglón.
Efectuado .DN (bit 13)	después de que la instrucción termina la operación. Después de que el renglón se hace falso, el procesador restablece el bit de efectuado .DN durante la próxima transición de falso a verdadero del renglón.
Error .ER (bit 11)	cuando la operación genera un overflow. La instrucción se detiene hasta que el programa de lógica de escalera restablece el bit .ER.

Importante: La instrucción STD calcula la desviación estándar usando el punto flotante (coma flotante) independientemente del tipo especificado para los parámetros de archivo o destino.



ATENCION: La instrucción STD manipula el valor de offset almacenado en S:24. Asegúrese de monitorear o cargar el valor de offset que desea antes de usar una dirección indexada. De lo contrario, es posible que ocurra una operación inesperada de la máquina, lo cual puede causar daños al equipo o lesiones corporales.

Ejemplo:



Si el bit 10 de la palabra de entrada 12 está establecido, la instrucción STD está habilitada. Los elementos N7:1, N7:2, N7:3 y N7:4 se usan para calcular la desviación estándar. Cuando el cálculo se completa, el bit 7 de la palabra de salida 10 se establece. La instrucción RES restablece los bits de estado del archivo de control R6:0.

Subtract (SUB)

Descripción:

SUB
SUBTRACT
Source A
Source B
Destination

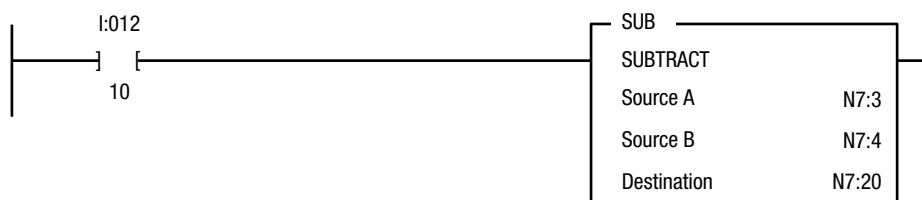
Use la instrucción SUB para restar un valor (origen A) de otro valor (origen B) y colocar el resultado en el destino. El origen A y el origen B pueden ser valores o direcciones que contienen valores. La tabla 4.V muestra los indicadores de estado para la instrucción SUB.

Importante: La instrucción SUB se ejecuta una vez durante cada escán siempre que el renglón sea verdadero. Si desea que los valores se resten una sola vez, incluya el comando ONS (vea el capítulo 13).

Tabla 4.V
Actualización de los indicadores de estado aritméticos para una instrucción SUB

Con este bit:	The Processor:
Acarreo (C)	se establece si se genera un acarreo negativo; de lo contrario, se restablece
Overflow (V)	se establece si se genera un underflow; de lo contrario, se restablece
Cero (Z)	se establece si el resultado es cero; de lo contrario, se restablece
Signo (S)	se establece si el resultado es negativo; de lo contrario, se restablece

Ejemplo:



Si el bit 10 de la palabra de entrada 12 está establecido, reste el valor en N7:4 del valor en N7:3 y almacene el resultado en N7:20.

Tangent (TAN) (procesadores PLC-5 con características mejoradas solamente)

Descripción:

TAN
TANGENT
Source
Destination

Use la instrucción TAN para hallar la tangente de un número (origen en radianes) y almacenar el resultado en el destino. La tabla 4.W muestra los indicadores de estado para la instrucción TAN.

El valor en el origen debe ser mayor o igual a -102943.7 y menor o igual a 102943.7 . Si no se encuentra dentro de este rango, el procesador devuelve un resultado ! INF ! en el destino. El valor resultante en el destino siempre es un número real.

Importante: Para obtener mejor precisión, el dato del origen debe ser mayor o igual a -2π y menor o igual a 2π .

Tabla 4.W
Actualización de los indicadores de estado aritméticos
para una instrucción TAN

Con este bit:	El procesador:
Acarreo (C)	siempre se restablece
Overflow (V)	se establece si se genera un overflow; de lo contrario, se restablece
Cero (Z)	se establece si el resultado es cero; de lo contrario, se restablece
Signo (S)	se establece si el resultado es negativo; de lo contrario, se restablece

Ejemplo:



Si el bit 10 de la palabra de entrada 12 está establecido, halle la tangente del valor en F8:15 y almacene el resultado en F8:16.

X to the Power of Y (XPY) (procesadores PLC-5 con características mejoradas solamente)

Descripción:

XPY
X TO POWER OF Y
Source A
Source B
Destination

Use la instrucción XPY para elevar un valor (origen A) a una potencia (origen B) y almacenar el resultado en el destino. Si el valor en el origen A es negativo, el exponente (fuente B) debe ser un valor entero. Si el exponente no es un número entero (por ejemplo, si es un valor de punto flotante [coma flotante]), el bit de overflow se establece y el valor absoluto de la base se usa en este cálculo. La tabla 4.X muestra los indicadores de estado para la instrucción XPY.

La instrucción XPY usa el algoritmo siguiente:

$$XPY = 10^{**} (Y * \text{algoritmo}(X))$$

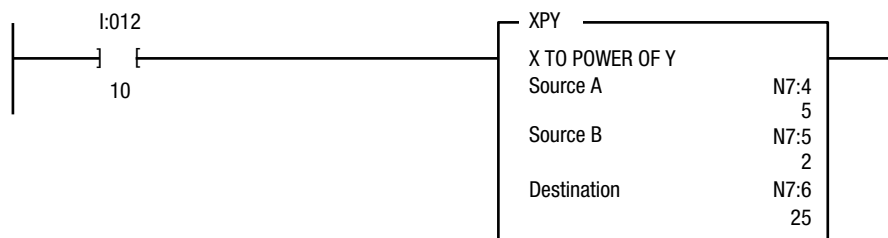
Si cualquiera de las operaciones intermedias en este algoritmo produce un overflow, el bit de fallo menor aritmético (S:10/14) se establece. El bit del indicador de estado aritmético se establece solamente si el resultado es un overflow.

Importante: Tome en cuenta de que x0 es igual a 1 y 0x es igual a 0. Para los números de punto flotante (coma flotante), 00 es igual a !NAN! (un valor matemático no válido) y para los números enteros, 00 es igual a -1.

Tabla 4.X
Actualización de los indicadores de estado aritméticos
para una instrucción XPY

Con este bit:	El procesador:
Acarreo (C)	siempre se restablece
Overflow (V)	se establece si se genera un overflow; de lo contrario, se restablece
Cero (Z)	se establece si el resultado es cero; de lo contrario, se restablece
Signo (S)	se establece si el resultado es negativo; de lo contrario, se restablece

Ejemplo:



Si el bit 10 de la palabra de entrada 12 está establecido, halle el valor en N7:4, elévelo a la potencia del valor en N7:5 y almacene el resultado en N7:6.

Notas:

Cómo usar las instrucciones lógicas

Instrucciones lógicas AND, NOT, OR, XOR

Estas instrucciones (tabla 5.A) realizan operaciones lógicas.

Tabla 5.A
Instrucciones lógicas disponibles

Si desea:	Use esta instrucción:	Que se encuentra en la página:
realizar una operación AND	AND	5-2
realizar una operación NOT	NOT	5-3
realizar una operación OR	OR	5-4
realizar una operación XOR	XOR	5-5

Los parámetros que usted introduce son constantes de programa o direcciones lógicas directas.

Vea el Apéndice C para obtener más información acerca de los operandos (y los tipos/valores de datos válidos de cada operando) usados por las instrucciones descritas en este capítulo.

Cómo usar los indicadores de estado aritméticos

Los bits de estado aritméticos se encuentran en los bits 0-3 de la palabra 0 del archivo de estado (S) del procesador. La tabla 5.B muestra los indicadores de estado:

Tabla 5.B
Indicadores de estado aritméticos

Este bit:	Descripción:
S:0/0	Acarreo (C)
S:0/1	Overflow (V)
S:0/2	Cero (Z)
S:0/3	Signo (S)

AND Operation (AND)

Descripción: Use la instrucción AND para realizar una operación AND usando los bits en las dos direcciones de origen.

AND
BITWISE AND
Source A
Source B
Destination

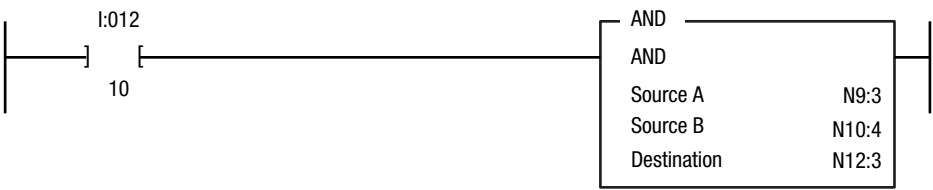
Tabla 5.C
Tabla de verdad para una operación AND

Origen A	Origen B	Resultado
0	0	0
1	0	0
0	1	0
1	1	1

Tabla 5.D
Actualización de los indicadores de estado aritméticos para una instrucción AND

Con este bit:	El procesador:
Acarreo (C)	siempre se restablece
Overflow (V)	siempre se restablece
Cero (Z)	se establece si el resultado es cero; de lo contrario, se restablece
Signo (S)	se establece si el bit más significativo está establecido; de lo contrario, se restablece

Ejemplo:

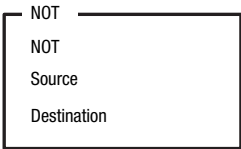


Si el bit 10 de la palabra de entrada 12 está establecido, el procesador realiza una operación AND en N9:3 y N10:4 y almacena el resultado en N12:3.

Origen A N9:3	0 0 0 0 0 0 0 0 1 0 1 0 1 0
Origen B N10:4	0 0 0 0 0 0 0 1 1 1 0 1 0 1
Destino N12:3	0 0 0 0 0 0 0 1 0 1 0 1 0

NOT Operation (NOT)

Descripción:



Use la instrucción NOT para realizar una operación NOT usando los bits en la dirección de origen. Esta operación también se conoce como una inversión de bit.

Importante: La instrucción NOT no está disponible en los procesadores PLC-5/15 de serie A.

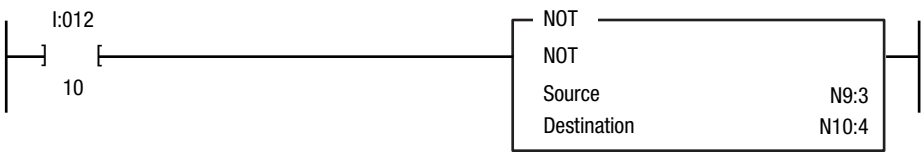
Tabla 5.E
Tabla de verdad para una operación NOT

Origen	Resultado
0	1
1	0

Tabla 5.F
Actualización de los indicadores de estado aritméticos para una instrucción NOT

Con este bit:	El procesador:
Acarreo (C)	siempre se restablece
Overflow (V)	siempre se restablece
Cero (Z)	se establece si el resultado es cero; de lo contrario, se restablece
Signo (S)	se establece si el bit más significativo está establecido; de lo contrario, se restablece

Ejemplo:



Si el bit 10 de la palabra 1 de entrada 2 está establecido, el procesador realiza una operación NOT en N9:3 y almacena el resultado en N10:4.

Origen N9:3	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0
Destino N10:4	1	1	1	1	1	1	1	0	1	0	1	0	1	0	1	1

OR Operation (OR)

Descripción:

OR
BITWISE INCLUSIVE OR
Source A
Source B
Destination

Use la instrucción OR para realizar una operación OR usando los bits en los dos orígenes (constantes o direcciones).

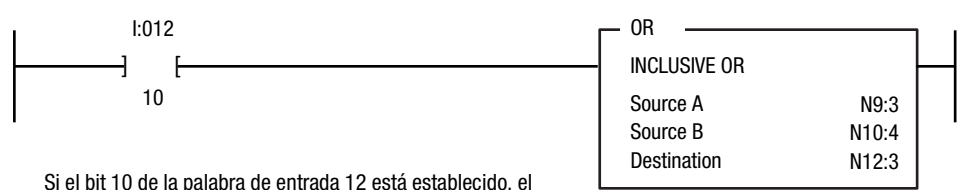
Tabla 5.G
Tabla de verdad para una operación OR

Origen A	Origen B	Resultado
0	0	0
1	0	1
0	1	1
1	1	1

Tabla 5.H
Actualización de los indicadores de estado aritméticos para una instrucción OR

Con este bit:	El procesador:
Acarreo (C)	siempre se restablece
Overflow (V)	siempre se restablece
Cero (Z)	se establece si el resultado es cero; de lo contrario, se restablece
Signo (S)	se establece si el bit más significativo está establecido; de lo contrario, se restablece

Ejemplo:



Si el bit 10 de la palabra de entrada 12 está establecido, el procesador realiza una operación OR en N9:3 y N10:4 y almacena el resultado en N12:3.

Origen A N9:3	0 0 0 0 0 0 0 1 0 1 0 1 0 1 0
Origen B N10:4	0 0 0 0 0 0 0 1 1 1 0 1 0 1 1
Destino N12:3	0 0 0 0 0 0 0 1 1 1 0 1 0 1 1

Exclusive OR Operation (XOR)

Descripción:

Use la instrucción XOR para realizar una operación O exclusivo con el uso de los bits en los dos orígenes (constantes o direcciones).

XOR
BITWISE EXCLUSIVE OR
Source A
Source B
Destination

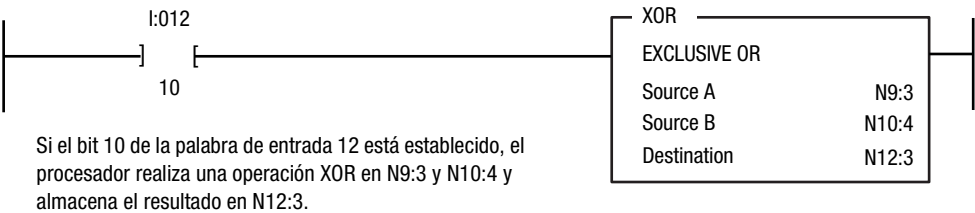
Tabla 5.I
Tabla de verdad para una operación XOR

Fuente A	Fuente B	Resultado
0	0	0
1	0	1
0	1	1
1	1	0

Tabla 5.J
Actualización de los indicadores de estado aritméticos para una instrucción XOR

Con este bit:	El procesador:
Acarreo (C)	siempre se restablece
Overflow (V)	siempre se restablece
Cero (Z)	se establece si el resultado es cero; de lo contrario, se restablece
Signo (S)	se establece si el bit más significativo está establecido; de los contrario, se restablece

Ejemplo:



Origen A N9:3	000000010101010
Origen B N10:4	000000011101011
Destino N12:3	000000001000001

Notas:

Instrucciones de conversión FRD y TOD, DEG y RAD

Cómo usar las instrucciones de conversión

Las instrucciones de conversión convierten los números enteros en BCD y convierte BCD en números enteros (con el uso de TOD y FRD). Por ejemplo, use TOD y FRD para señales hacia/desde los dispositivos de E/S BCD, para fines de visualización en pantalla o para la compatibilidad de números con los procesadores de la familia PLC-2. También puede convertir los radianes en grados y grados en radianes (con el uso de DEG y RAD). Por ejemplo, puede usar DEG y RAD con instrucciones trigonométricas (vea el capítulo 4).

La tabla 6.A indica las instrucciones de conversión disponibles.

Tabla 6.A
Instrucciones de conversión disponibles

Si desea:	Use esta instrucción:	Que se encuentra en la pág.:
Convertir de número entero en BCD	TOD	6-2
Convertir de BCD en número entero	FRD	6-2
Convertir radianes en grados	DEG*	6-3
Convertir grados en radianes	RAD*	6-4

* Estas instrucciones son compatibles con los procesadores PLC-5 con características mejoradas solamente.

Los parámetros que usted introduce son constantes de programa o direcciones lógicas de los valores que desea.

Vea el Apéndice C para obtener más información acerca de los operandos (y los tipos/valores de datos válidos de cada operando) usados por las instrucciones descritas en este capítulo.

Cómo usar los indicadores de estado aritméticos

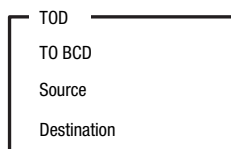
Los indicadores de estado aritméticos se encuentran en los bits 0-3 de la palabra 0 del archivo de estado (S2) del procesador. La tabla 6.B muestra los indicadores de estado:

Tabla 6.B
Indicadores de estado aritméticos

Este bit:	Descripción:
S:0/0	Acarreo (C)
S:0/1	Overflow (V)
S:0/2	Cero (Z)
S:0/3	Signo (S)

Convert to BCD (TOD)

Descripción:

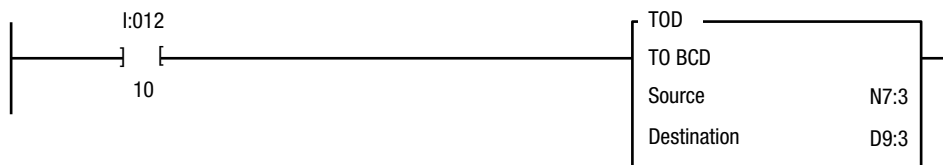


Use la instrucción TOD para convertir un valor entero en un valor BCD. Si el valor entero es mayor que 9999, el procesador almacena 9999 y establece el bit de overflow. Si el valor entero es negativo, el procesador almacena 0 en el destino y establece los bits de overflow y estado de cero.

Tabla 6.C
Actualización de los indicadores de estado aritméticos para una instrucción TOD

Con este bit:	El procesador:
Acarreo (C)	siempre se restablece
Overflow (V)	se establece si el valor entero se encuentra fuera del rango 0-9999; de lo contrario, se restablece
Cero (Z)	se establece si el valor de destino es negativo o cero; de lo contrario, se restablece
Signo (S)	siempre se restablece

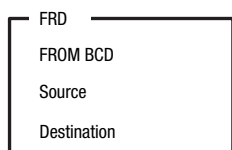
Ejemplo:



Si el bit 10 de la palabra de entrada 12 está establecido, convierta el valor en N7:3 a un valor BCD y almacene el resultado en D9:3.

Convert from BCD (FRD)

Descripción:



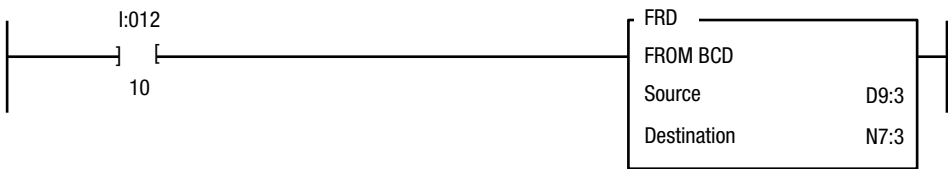
Use la instrucción FRD para convertir un valor BCD en un valor entero. Convierta los valores BCD en valores enteros antes de manipular los valores con lógica de escalera puesto que el procesador considera los valores BCD como valores enteros. El valor BCD real se puede perder o distorsionar.

Tabla 6.D
Actualización de los indicadores de estado aritméticos para una instrucción FRD

Con este bit:	El procesador:
Acarreo (C)	siempre se restablece
Overflow (V)	siempre se restablece
cero (Z)	se establece si el valor de destino es cero; de lo contrario, se restablece
Signo (S)	siempre se restablece

La instrucción FRD convertirá un número no decimal sin una condición de error. Por ejemplo, si “C” está en el origen, se convierte en “12”, aunque “C” no sea un número decimal no válido.

Ejemplo:



Si el bit 10 de la palabra de entrada 12 está establecido, convierta el valor en D9:3 en un valor entero y almacene el resultado en N7:3.

Degree (DEG)
(procesadores PLC-5 con
características mejoradas
solamente)

Descripción:

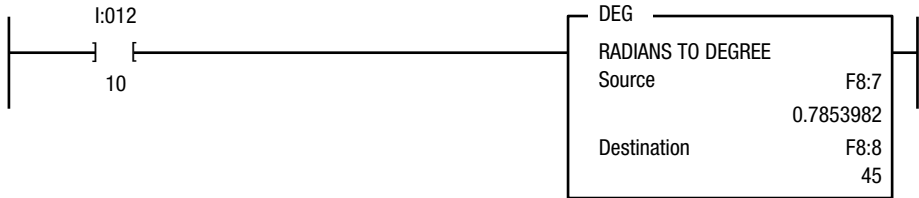


Use la instrucción DEG para convertir los radianes (origen) en grados y almacenar el resultado en el destino (origen multiplicado por $180/\pi$).

Tabla 6.E
Actualización de los indicadores de estado aritméticos
para una instrucción DEG

Con este bit:	El procesador:
Acarreo (C)	siempre se restablece
Overflow (V)	se establece si se genera un overflow; de lo contrario, se restablece
Cero (Z)	se establece si el resultado es cero; de lo contrario, se restablece
Signo (S)	se establece si el resultado es negativo; de lo contrario, se restablece

Ejemplo:



Si el bit 10 de la palabra de entrada 12 está establecido, convierta el valor en F8:7 a grados y almacene el resultado en F8:8.

Radian (RAD) (procesadores PLC-5 con características mejoradas solamente)

Descripción:

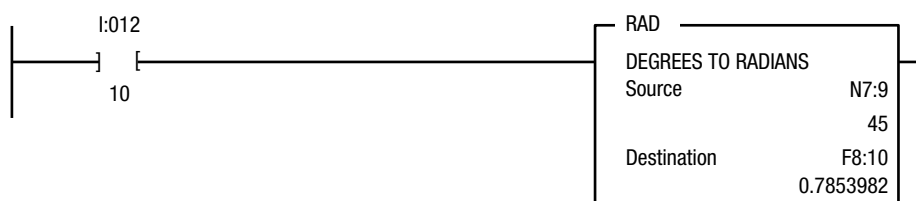
RAD
DEGREES TO RADIANS
Source
Destination

Use la instrucción RAD para convertir grados (origen) en radianes y almacenar el resultado en el destino (origen multiplicado por $\pi/180$).

Tabla 6.F
Actualización de los indicadores de estado aritméticos
para una instrucción RAD

Con este bit:	El procesador:
Acarreo (C)	siempre se restablece
Overflow (V)	se establece si se genera un overflow; de lo contrario, se restablece
Cero (Z)	se establece si el resultado es cero; de lo contrario, se restablece
Signo (S)	se establece si el resultado es negativo; de lo contrario, se restablece

Ejemplo:



Si el bit 10 de la palabra de entrada 12 está establecido, convierta el valor en N7:9 a radianes y almacene el resultado en F8:10.

Instrucciones para modificar y mover bits BTM, MOV, MVM

Cómo usar las instrucciones para modificar y mover bits

Las instrucciones para modificar y mover bits le permiten modificar y mover bits. La tabla 7.A indica las instrucciones de mover disponibles.

Tabla 7.A
Instrucciones disponibles para modificar y mover bits

Si desea:	Use esta instrucción:	Que se encuentra en la página:
Mover bits dentro de una palabra o entre palabras	BTM	7-2
Copiar el valor en una palabra a otra palabra	MOV	7-3
Copiar la porción deseada de un valor de 16 bits enmascarando el resto del valor	MVM	7-4

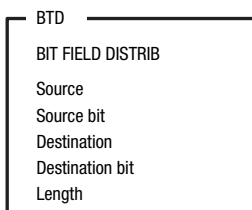
Estas instrucciones operan en números enteros, binarios o de punto flotante (coma flotante) de 16 bits para mover o copiar bits entre palabras. La instrucción MVM usa una máscara para pasar o bloquear los bits de datos del origen. Una máscara pasa los datos cuando los bits de máscara están establecidos (1). Una máscara bloquea los datos cuando los bits de máscara están restablecidos (0). La máscara debe tener el mismo tamaño de palabra que el origen y el destino.

El procesador no redondea correctamente los números menor que -1 cuando redondea los números de punto flotante (coma flotante) durante el movimiento a una palabra de número entero.

Vea el Apéndice C para obtener más información acerca de los operandos (y los tipos/valores de datos válidos de cada operando) usados por las instrucciones descritas en este capítulo.

Bit Distribute (BTD)

Descripción:



La instrucción BTD es una instrucción de salida que mueve hasta 16 bits de datos dentro de una palabra o entre palabras. No se cambia el origen. La instrucción sobrescribe el destino con los bits especificados. Si la longitud del campo de bits sobrepasa la palabra de destino, el procesador no guarda los bits de overflow. Estos bits de overflow se pierden. No continúan en la próxima palabra.

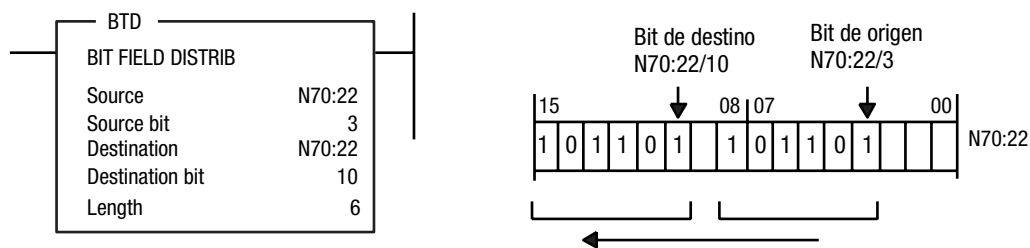
Durante cada escán, cuando el renglón que contiene la instrucción BTD es verdadero, el procesador mueve el campo de bits desde la palabra de origen a la palabra de destino. Para mover los datos dentro de una palabra, introduzca la misma dirección para el origen y destino.

Cómo introducir los parámetros

Es necesario proporcionar al procesador la información siguiente para programar la instrucción BTD:

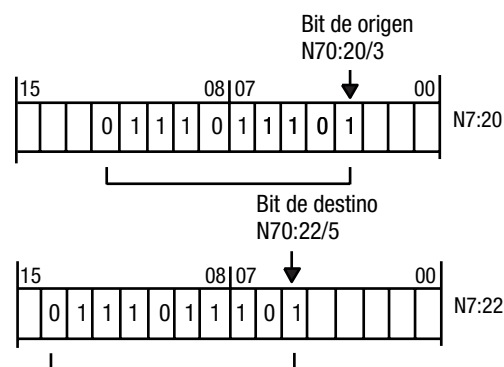
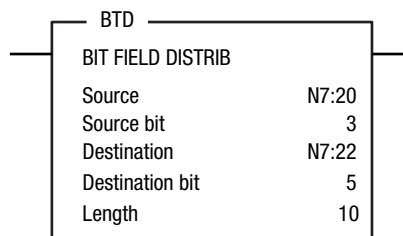
Parámetro:	Definición:
Origen	la dirección de la palabra de origen en un archivo binario o de número entero. El origen no se cambia.
Bit de origen	el número del bit (número de bit inferior) en la palabra de origen a partir del cual se inicia el movimiento.
Destino	la dirección de la palabra de destino en un archivo binario o de número entero. La instrucción sobrescribe los datos ya almacenados en el destino.
Bit de destino	el número del bit (número de bit inferior) en la palabra de destino a partir del cual el procesador comienza a copiar los bits desde la palabra de origen.
Longitud	el número de bits que se deben mover.

Ejemplo: Cómo mover bits dentro de una palabra



13384

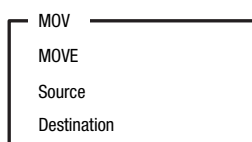
Ejemplo: Cómo mover bits entre palabras



Importante: Los bits se pierden si sobrepasan el final de la palabra de origen. Los bits **no** continúan en la próxima palabra superior.

Move (MOV)

Descripción:



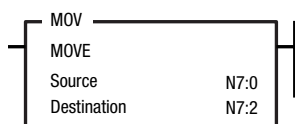
La instrucción MOV es una instrucción de salida que copia la dirección de origen a un destino. La instrucción mueve los datos durante cada escán siempre que el renglón permanezca verdadero.

La Tabla 7.B describe cómo el procesador actualiza los indicadores de estado aritméticos.

Tabla 7.B
Actualización de los indicadores de estado aritméticos para la instrucción MOV

Con este bit:	El procesador:
Acarreo (C)	siempre se restablece
Overflow (V)	se establece si se genera un overflow durante la conversión de punto flotante (coma flotante) en número entero; de lo contrario, se restablece
Cero (Z)	se establece si el resultado es cero; de lo contrario, se restablece
Signo (S)	se establece si el resultado es negativo; de lo contrario, se restablece

Ejemplo:

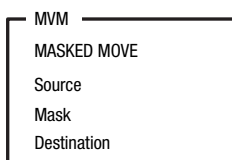


Es necesario proporcionar al procesador la información siguiente para programar esta instrucción:

Parámetro:	Definición:
origen	es una constante de programa o dirección de datos de la cual la instrucción lee una imagen del valor. También puede usar un símbolo siempre que el nombre del símbolo tenga más de 1 carácter. El origen no se cambia.
destino	la dirección de datos a la cual la instrucción escribe el resultado de la operación. La instrucción sobrescribe los datos almacenados en el destino.

Masked Move (MVM)

Descripción:



La instrucción MVM es una instrucción de salida que copia el origen a un destino y permite que porciones de los datos se enmascaren. La instrucción mueve los datos durante cada escán siempre que el renglón permanezca verdadero.

Usted puede usar la instrucción MVM para copiar valores de imagen de E/S, binarios o enteros. Por ejemplo, use la instrucción MVM para extraer los datos de bit tales como los bits de estado o control de un elemento que contiene datos de bit y palabra.

La Tabla 7.C describe cómo el procesador actualiza los indicadores de estado aritméticos.

Tabla 7.C
Actualización de los indicadores de estado aritméticos para la instrucción MVM

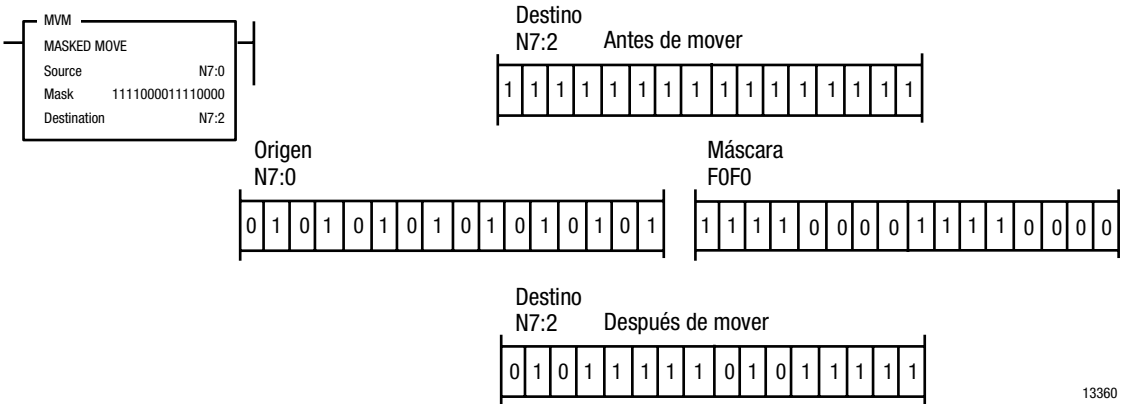
Con este bit:	El procesador:
Acarreo (C)	siempre se restablece
Overflow (V)	siempre se restablece
Cero (Z)	se establece si el resultado es cero; de lo contrario, se restablece
Signo (S)	se establece si el resultado es negativo; de lo contrario, se restablece

Cómo introducir los parámetros

Es necesario proporcionar al procesador la información siguiente para programar esta instrucción:

Parámetro:	Definición:
Origen	una constante de programa o dirección de datos de la cual la instrucción lee una imagen del valor. El origen no se cambia.
Máscara	una dirección o valor hexadecimal que especifica qué bits se deben pasar o bloquear. Es necesario establecer (1) los bits de máscara para mover datos. Los datos movidos sobrescriben los datos de destino. No se modifican los bits en el destino correspondientes a ceros en la máscara. Si desea que el programa de lógica de escalera cambie el valor de máscara, almacene la máscara en la dirección de datos. Cuando introduzca un valor en este campo, asegúrese de incluir el tipo de datos, el número de archivo y el número de palabra. Por ejemplo, escriba B100:0. De lo contrario, introduzca un valor hexadecimal para un valor de máscara constante. Por ejemplo, escriba F800.
Destino	la dirección de datos a la cual la instrucción escribe el resultado de la operación. La instrucción sobrescribe los datos almacenados en el destino.

Ejemplo:



Notas:

Conceptos de instrucciones de archivo

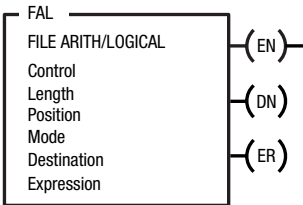
Conceptos de operación de archivo

Este capítulo presenta los conceptos de operación en bloques para las instrucciones de Aritmética y lógica de archivos (FAL) y Búsqueda y comparación de archivos (FSC).

La instrucción FAL realiza operaciones aritméticas y lógicas en bloques de palabras. La instrucción FSC realiza operaciones de comparación en bloques de palabras. Vea el capítulo 9 para obtener información específica acerca de las instrucciones FAL y FSC.

Vea el Apéndice C para obtener información acerca de los operandos (y los tipos/valores de datos válidos de cada operando) usados por la instrucción tratada en este capítulo.

Cómo introducir los parámetros



Es necesario proporcionar al procesador la información siguiente para introducir una instrucción de archivo:

Parámetro:	Definición:
Control	la dirección de la estructura de control en un archivo de tipo control (R). El procesador usa esta información para ejecutar la instrucción. Vea “Cómo usar la estructura de control” en la página 8-2.
Longitud	el número de palabras en el bloque de datos en el cual la instrucción de archivo opera. Introduzca cualquier número decimal entre 1-1000.
Posición	la palabra actual dentro del bloque de datos a la cual el procesador obtiene acceso. Generalmente se introduce un cero para comenzar al inicio de un bloque.
Modo	el número de palabras de archivo en que se opera durante cada escán del renglón en el programa. El modo permite que se distribuya la operación en el bloque completo de palabras. Especifique uno de los siguientes: • para el modo Todo, escriba A • para el modo Numérico, escriba un número decimal (1-1000) • para el modo Incremental, escriba I Vea “Cómo seleccionar los modos de operación de bloques” en la página 8-5 para obtener más información acerca de los diferentes modos.
Destino	la dirección en la cual el procesador almacena el resultado de la operación. La instrucción se convierte en el tipo de datos especificado por la dirección de destino.
Expresión	contiene direcciones, constantes de programa y operadores que especifican el origen de datos y las operaciones que se deben realizar. Si se introduce el prefijo de índice (numérico) para una dirección de destino o expresión, el procesador lo acepta como la dirección de la primera palabra de un bloque en que se debe operar. El procesador asigna y usa el valor de offset en el estado de módulo para procesar la dirección de bloque. Si se elimina el prefijo numérico, el procesador lo acepta como la dirección de una sola tarea en que se debe operar.

Importante: Asegúrese de que el valor del índice (positivo o negativo) no cause que la dirección indexada sobrepase el límite del tipo de archivo. El procesador no verifica esto a no ser que se use una dirección indirecta indexada o se exceda la zona de memoria de la tabla de datos. Si la dirección indexada sobrepasa la zona de la tabla de datos, el procesador inicia un error de tiempo de ejecución y establece un fallo mayor. El procesador no verifica si la dirección indexada cruza los tipos de archivo, tales como N7 a N8.



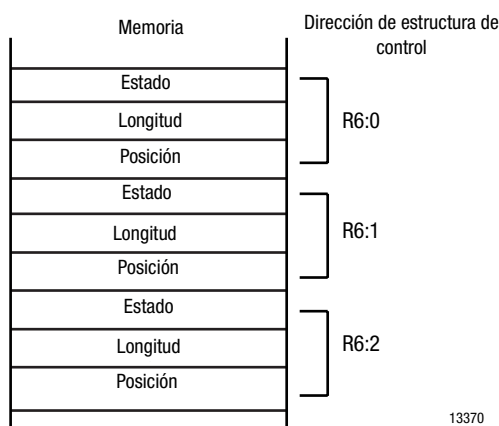
ATENCION: Las instrucciones con un símbolo # en una dirección manipulan el valor de offset almacenado en S:24. Asegúrese de monitorear o cargar el valor de offset que desea antes de usar una dirección indexada. De lo contrario, puede ocurrir una operación inesperada de la máquina, lo cual puede resultar en daños al equipo o lesiones corporales.

Vea el capítulo que describe el direccionamiento de los archivos de la tabla de datos en el manual del usuario del software para obtener más información acerca del direccionamiento indexado.

Cómo usar la estructura de control

La estructura de control (tipo de archivo R) controla la operación de la instrucción de archivo. Esta estructura es similar a un contador y controla el archivo según longitud y posición así como según los bits de estado y control (Figura 8.1). Usted introduce la dirección de estructura de control (por ejemplo, R6:0) en el campo de control cuando programa una instrucción FAL o FSC.

Figura 8.1
Ejemplo de archivo de control R6:0



ATENCION: No use la misma dirección de control para más de una instrucción. La duplicación de una dirección de control puede resultar en una operación inesperada con la posibilidad de causar daños al equipo y/o lesiones corporales.

Cómo manipular los datos de archivo

La estructura de control almacena la información siguiente:

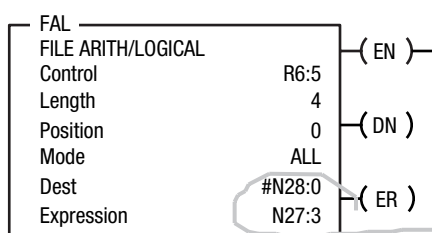
- Bits de estado
- Longitud (.LEN) del bloque (1-1000 palabras)
- Posición (.POS) de las palabras en que el procesador opera

La instrucción FAL y la instrucción FSC tienen su propio conjunto de bits de estado. El capítulo 9 presenta una descripción de estos bits de estado para las instrucciones FAL y FSC.

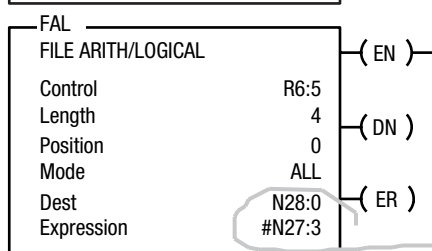
Las manipulaciones de datos típicas con las instrucciones de archivo incluyen:

- Copiar datos de
 - una palabra de origen a un bloque de destino
 - un bloque de origen a un bloque de destino
 - un bloque de origen a una palabra de destino
- Operar en datos desde múltiples orígenes tales como
 - palabras de origen
 - bloques de origen
- Almacenar el resultado en
 - un bloque de destino
 - una palabra de destino

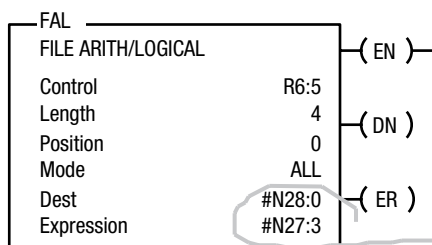
El prefijo # de una dirección de destino o expresión la establece como la primera palabra de un bloque en que se debe operar. La ausencia del prefijo # establece la dirección como la dirección de una sola palabra en que se debe operar.



El prefijo # de la dirección de destino y la ausencia de un prefijo # de para dirección de expresión define ésta como una operación de **palabra a bloque**.



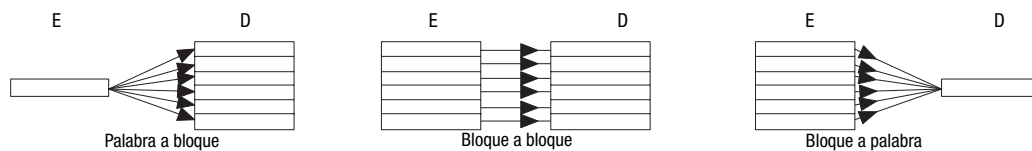
La ausencia de un prefijo # para la dirección de destino y el prefijo # para la dirección de expresión define ésta como una operación de **bloque a palabra**.



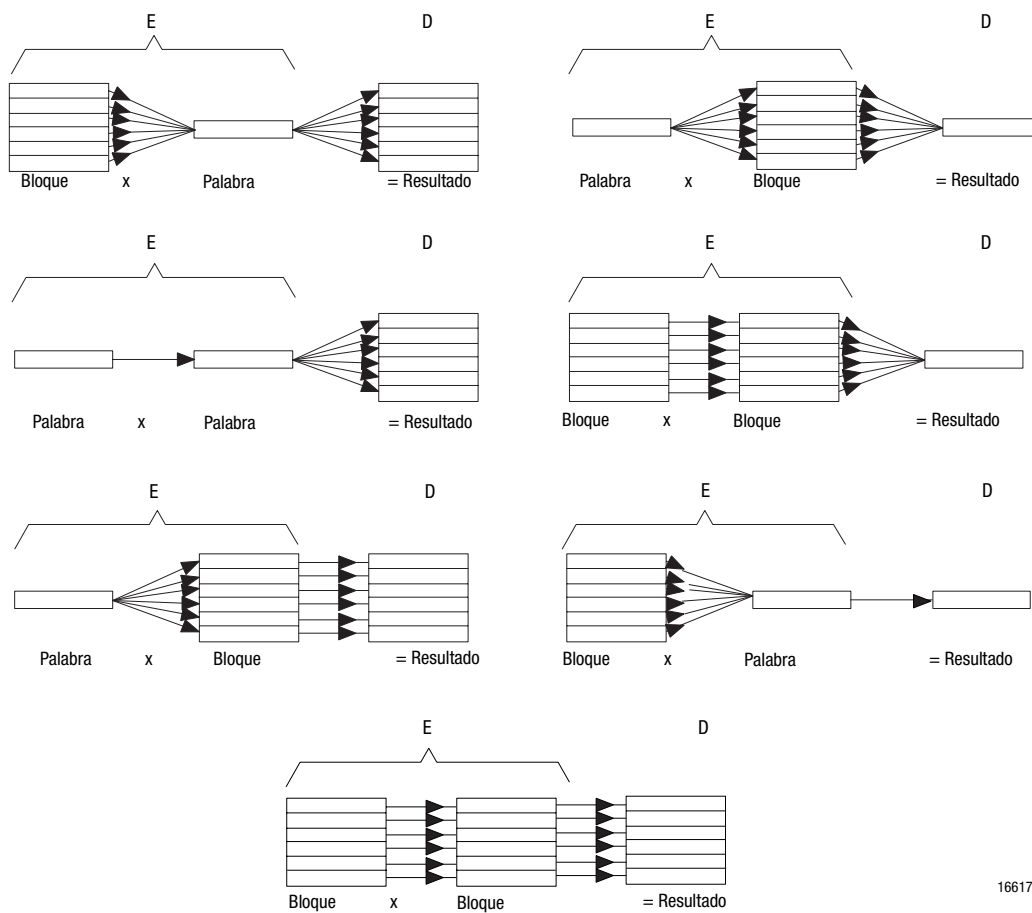
El prefijo # para la dirección de destino y el prefijo # para la dirección de expresión define ésta como una operación de **bloque a bloque**.

El ejemplo siguiente muestra las manipulaciones genéricas de datos usadas con las instrucciones de archivo (E = expresión, D = destino, x = operación).

Cómo mover datos



Cómo operar en datos



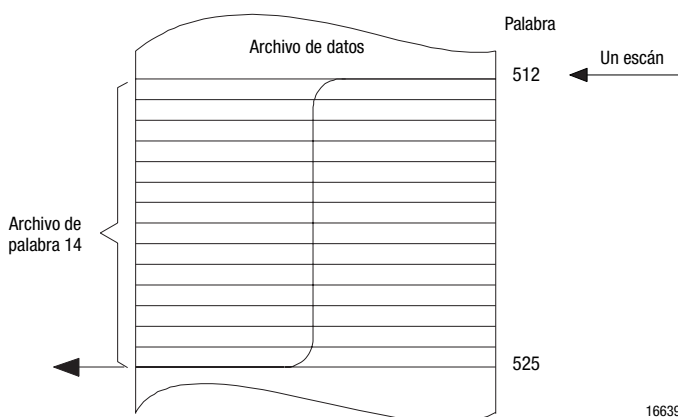
16617a

Cómo seleccionar los modos de operación de bloque

El modo bloque indica al procesador cómo distribuir la operación de bloque durante uno o más escanes de programa. Seleccione uno de los modos siguientes:

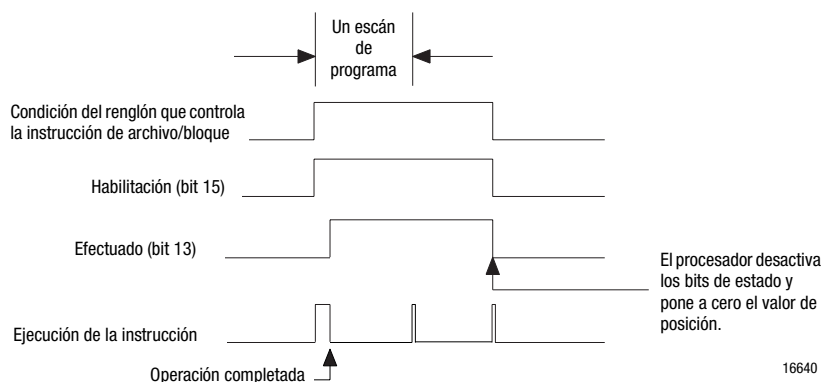
Modo Todo

En el modo **Todo**, se opera en el archivo entero antes de continuar al próximo renglón del programa. Escriba **A** para el parámetro de modo cuando introduzca la instrucción.



La operación comienza cuando el renglón va de no verdadero a verdadero. El valor de posición (.POS) en la estructura de control indica la palabra en el bloque de datos que la instrucción está usando actualmente. La operación se detiene cuando la función se completa o cuando el procesador detecta un error.

El siguiente diagrama de temporización muestra la relación entre los bits de estado y la operación de instrucción. Cuando la ejecución de la instrucción se completa, el bit de efectuado se activa. Los bits de efectuado y habilitación no se desactivan y el valor de posición no se pone en cero hasta que las condiciones de renglón ya no son verdaderas. Sólo entonces una transición de no verdadero a verdadero de las condiciones de renglón puede activar otra operación.

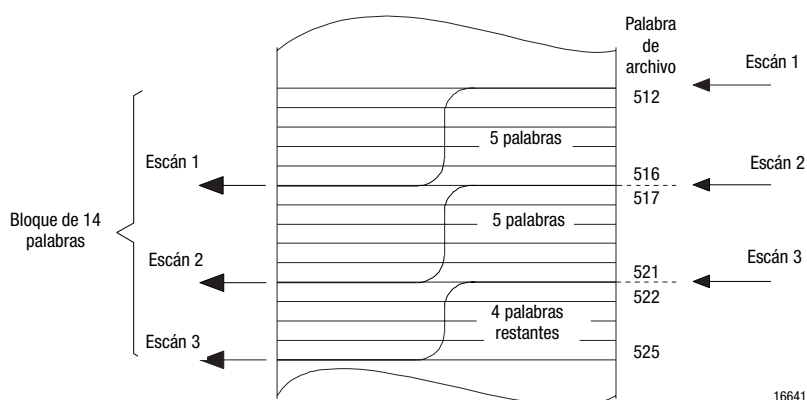


Modo numérico

El **modo numérico** distribuye la operación de archivo durante varios escanes de programa. Para seleccionar el modo numérico, introduzca el **número de palabras por escán** (1-1000) para el parámetro de modo cuando introduzca la instrucción de archivo. El número de palabras introducido debe ser menos o igual a la longitud del archivo.

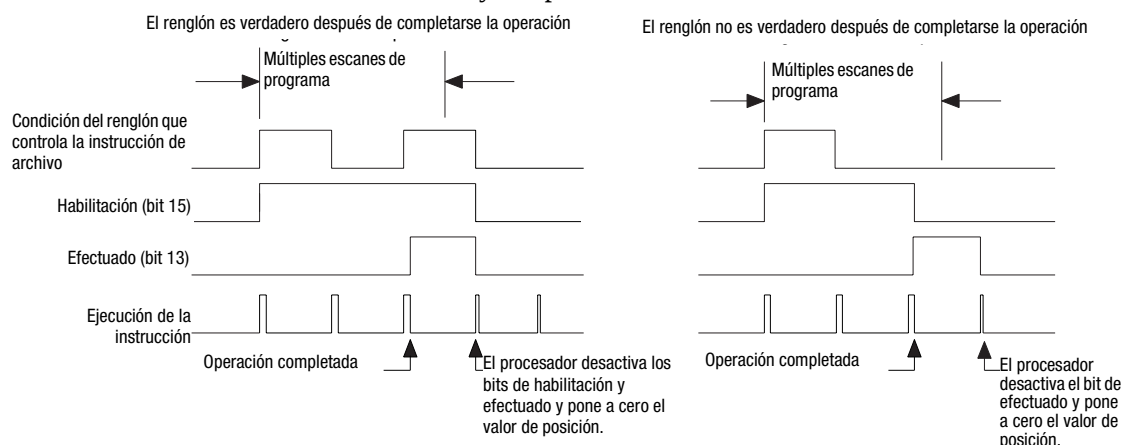
La ejecución se activa cuando las condiciones de renglón van de no verdadero a verdadero. Una vez activada, la instrucción se ejecuta continuamente cada vez que el renglón se escanea en el programa durante el número de escanes necesarios para completar la operación en todo el archivo. Una vez activada, la lógica de escalera del renglón puede cambiarse repetidamente sin interrumpir la ejecución de la instrucción.

Cada vez que el renglón se escanea, la instrucción opera en el número de palabras igual a la velocidad que usted introdujo para el valor de modo. La instrucción continúa hasta que ha operado en el número de palabras que especificó según el valor de longitud. Durante el último escán del renglón, es posible que el procesador opere en menos que el número de palabras introducidas.



Importante: Evite usar los resultados de una instrucción de archivo que opere en modo numérico hasta que el bit de efectuado se establezca puesto que los datos no estarán completos.

El siguiente diagrama de temporización muestra la relación entre los bits de estado y la operación de instrucción.



Cuando concluye la ejecución de la instrucción, el bit de efectuado se activa.

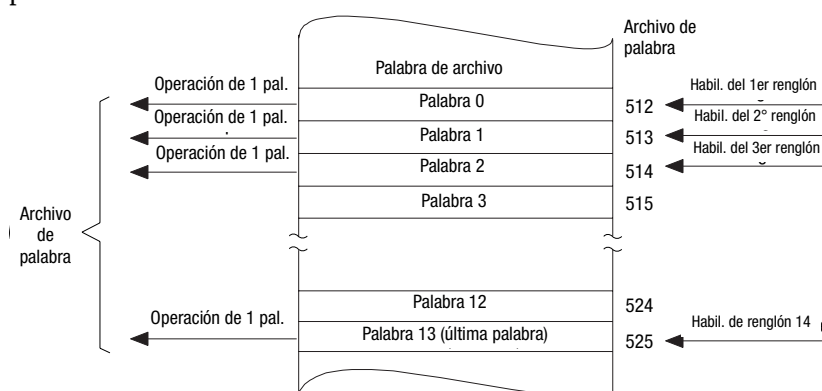
Si el renglón es verdadero después de completarse la operación, los bits de habilitación y efectuado no se desactivan hasta que el renglón ya no es verdadero. Cuando el renglón ya no es verdadero, estos bits se desactivan y el valor de posición se pone a cero.

Si el renglón no es verdadero después de completarse la operación, el bit de habilitación se desactiva inmediatamente y, un escán después de desactivarse el bit de habilitación, el bit de efectuado se desactiva y el valor de posición se pone a cero.

Sólo después de desactivarse los bits de habilitación y efectuado es posible que una transición de no verdadero a verdadero de las condiciones del renglón pueda activar otra operación.

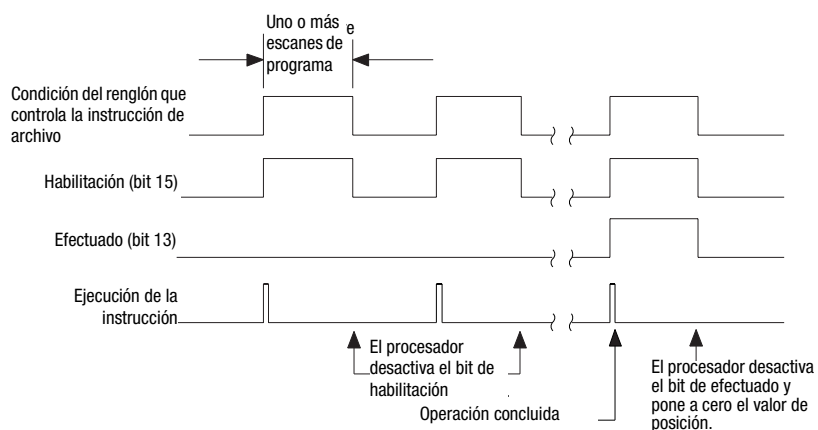
Modo incremental

El **modo incremental** manipula una palabra del archivo cada vez que el renglón va de no verdadero a verdadero. Escriba **I** para el parámetro de modo cuando introduzca la instrucción.



11

El siguiente diagrama de temporización muestra la relación entre los bits de estado y la operación de la instrucción.



16644

La ejecución ocurre solamente durante un escán del programa en que el renglón va de no verdadero a verdadero. Cada vez que esto ocurre, se opera en una palabra solamente. El bit de habilitación se activa cuando la lógica de escalera del renglón es verdadera. El bit de efectuado se activa cuando se ha operado en la última palabra en el archivo. Cuando se ha operado en la última palabra en el archivo y el renglón va de verdadero a no verdadero, los bits de habilitación y efectuado se desactivan y el valor de posición se pone a cero. Si el renglón permanece verdadero durante más de un escán de programa, el archivo de instrucción no se ejecuta en los escanes siguientes a la transición.

Importante: Si se opera en un archivo entero, evite usar los resultados de una instrucción de archivo/bloque mediante el modo incremental hasta que el bit de efectuado se active (los datos estarán incompletos).

Caso especial – modo numérico con palabras por escán = 1

La diferencia entre el modo numérico con una velocidad de 1 palabra por escán y el modo incremental es:

- El modo numérico con cualquier número de palabras por escán requiere solamente una transición de no verdadero a verdadero para la ejecución continua de la instrucción hasta completarse la operación en todo el archivo.
- El modo incremental requiere una transición de renglón de no verdadero a verdadero para cada palabra en el archivo.

Instrucciones de archivo FAL, FSC, COP, FLL

Cómo usar las instrucciones de archivo

Las instrucciones de archivo realizan operaciones en datos de archivo y datos de archivo de comparación. La Tabla 9.A indica las instrucciones de archivo disponibles.

Tabla 9.A
Instrucciones de archivo disponibles

Si desea:	Use esta operación:	Que se encuentra en la página:
Realizar operaciones aritméticas, lógicas, de desplazamiento y de función en los datos de archivo	FAL	9-2
Realizar operaciones de búsqueda y comparación en los datos de archivo	FSC	9-14
Copiar el contenido de un archivo a otro archivo	COP	9-19
Llenar un archivo con valores específicos	FLL	9-20

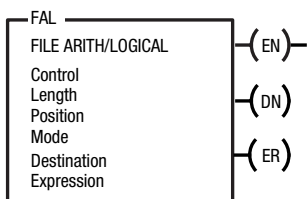
Si no lo ha hecho ya, repase los conceptos básicos de la operación de archivo en el capítulo anterior. Vea el manual del usuario del software para obtener más información acerca de cómo usar las direcciones indexadas.

Vea el Apéndice C para obtener más información acerca de los operandos (y los valores/tipos de datos válidos de cada operando) usados por las instrucciones descritas en este capítulo.

File Arithmetic and Logic (FAL)

La instrucción FAL realiza operaciones de copiar, aritméticas, lógicas y de función en los datos almacenados en los archivos. La instrucción FAL realiza las mismas operaciones que la instrucción CPT. La diferencia es que la instrucción FAL realiza operaciones en palabras múltiples y la instrucción CPT manipula palabras únicas.

Descripción:



La instrucción FAL es una instrucción de salida que realiza las operaciones definidas por direcciones de origen y operadores que usted escribe en la expresión. La instrucción escribe los resultados en una dirección de destino.

Seleccione cómo el procesador distribuye la operación durante uno o más escanes de programa mediante el modo de instrucción. Vea el capítulo 8 para obtener más información acerca de los modos de operación de archivos.

La instrucción FAL convierte automáticamente el tipo de datos en las direcciones de origen a el tipo de datos que usted especifica en la dirección de destino.

Puede usar esta instrucción para realizar operaciones como éstas:

- poner a cero un archivo
- copiar datos de un archivo a otro
- efectuar cálculos aritméticos o lógicos en datos almacenados en archivos
- descargar un archivo de códigos de error uno por uno para la visualización en pantalla



ATENCION: Las instrucciones con un signo # en una dirección manipulan el valor de offset almacenado en S:24. Asegúrese de monitorear o cargar el valor de offset que desea antes de usar una dirección indexada. De lo contrario, puede ocurrir una operación inesperada de la máquina, con posibles daños al equipo y/o lesiones corporales.

Tabla 9.B
Operaciones FAL

Tipo	Operador	Descripción	Ejemplo de operación
Copiar	ninguno	copiar de A a B	introduzca la dirección de origen en la expresión; introduzca la dirección de destino en el destino
Poner a cero	ninguno	poner a cero un valor	0 (introduzca 0 para la expresión)
Aritmético	+	sumar	2 + 3 2 + 3 + 7 (Procesadores PLC-5 con características mejoradas solamente)
	–	restar	12 – 5 (12 – 5) – 1 (Procesadores PLC-5 con características mejoradas solamente)
	*	multiplicar	5 * 2 6 * (5 * 2) (Procesadores PLC-5 con características mejoradas solamente)
		dividir	24 6 (24 6) * 2 (Procesadores PLC-5 con características mejoradas solamente)
	–	cambiar signo	– N7:0
	SQR	raíz cuadrada	SQR N7:0
	**	exponencial (x a la potencia de y)	10**3 (Procesadores PLC-5 con características mejoradas solamente)
Bit a bit	AND	función Y	D9:3 AND D10:4
	OR	función O	D9:4 OR D9:5
	XOR	función O exclusivo	D10:10 XOR D10:11
	NOT	complemento bit a bit	NOT D9:4
Conversión	FRD	convertir de BCD en binario	FRD D14:0
	TOD	convertir de binario en BCD	TOD N7:0

Cómo usar los bits de estado

Examine los bits de estado de control en el elemento de control para usar la instrucción FAL correctamente. Estos bits se direccionan mediante mnemónicos:

Este bit:	Se establece:
Habilitación .EN (bit 15)	mediante una transición de falso a verdadero e indica que la instrucción está habilitada. En el modo incremental, el bit .EN sigue a la condición del renglón. En los modos Numérico y Todo (ALL), el bit .EN permanece establecido hasta que la instrucción completa su operación, independientemente de la condición del renglón. El bit .EN se restablece cuando el renglón se hace falso y la instrucción completa su operación.
Efectuado .DN (bit 13)	después de que la instrucción ha operado en el último conjunto de palabras. Si la instrucción es falsa cuando se completa en el modo numérico, restablece el bit .DN un escán de programa después de completarse la operación. Si la instrucción es verdadera cuando se completa la operación, el bit .DN se restablece cuando la instrucción se hace falsa.
Error .ER (bit 11)	cuando la operación genera un overflow. La instrucción se detiene hasta que el programa de lógica de escalera restablece el bit .ER. Cuando el procesador detecta un error, el valor de posición almacena el número de la palabra que entró en fallo.

Con la instrucción FAL, se puede mostrar en pantalla un máximo de 80 caracteres de la expresión. Si la expresión introducida está cerca del máximo de 80 caracteres, es posible que el procesador amplíe este máximo para el renglón que contiene la instrucción cuando se acepte dicho renglón. Cuando usted intenta editar la expresión se muestran en pantalla solamente los 80 primeros caracteres y el renglón se muestra en pantalla como renglón de error. Sin embargo, el procesador contiene la expresión completa y la instrucción funciona correctamente.

Para evitar este problema de visualización en pantalla, exporte el archivo de memoria del procesador y efectúe las ediciones en el archivo de texto PC5. Luego importe este archivo de texto. Vea el manual de programación para obtener más información acerca de cómo importar/exportar los archivos de memoria del procesador.

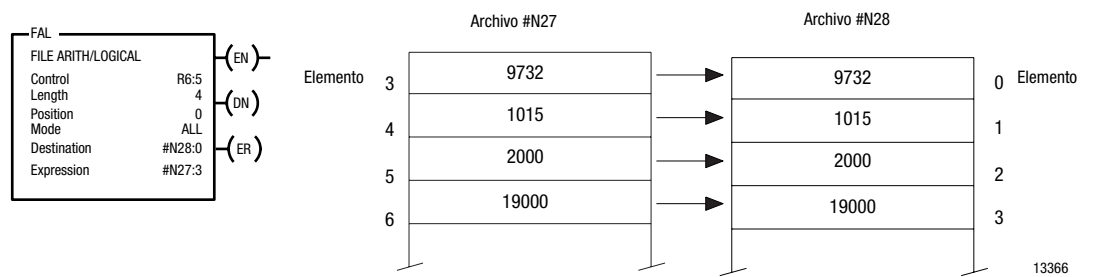
FAL Copy Operations

La operación de copiar FAL copia datos:

- entre archivos
- de una palabra a un archivo
- de un archivo a una palabra

Para copiar datos con la operación de copiar FAL, introduzca la dirección de origen o la constante de programa en la expresión y la dirección de destino en el destino.

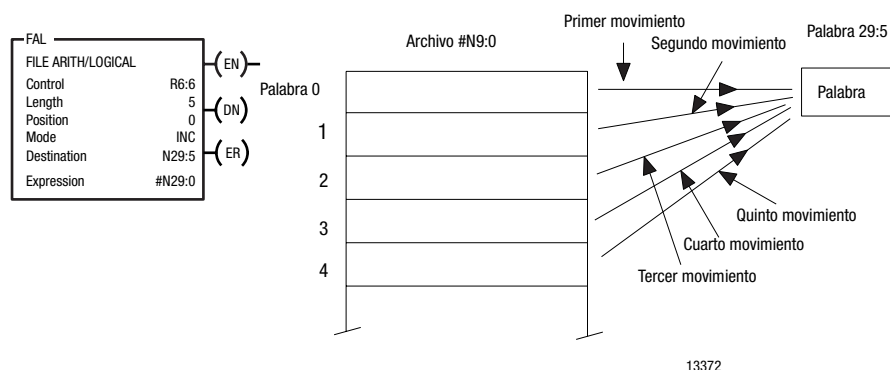
Ejemplo de copiar archivo a archivo:



Este parámetro:	Indica al procesador:
Control (R6:5)	Qué estructura de control controla la operación. Este parámetro está controlado por la condición del renglón, el estado de los bits .EN y .DN y el modo (Incremental, Numérico o Todo). Contiene la ubicación del último valor al cual la instrucción FAL escribió. Por ejemplo, en el modo incremental, si la posición = 0 y la longitud = 4, la última palabra a la cual la instrucción FAL escribió es la palabra 3 puesto que la instrucción comienza en la ubicación 0.
Longitud (4)	Mover cuatro palabras
Posición (0)	Comenzar en la dirección de origen
Modo (ALL)	Ejecutar la longitud durante un escán de programa
Destino (#N28:0)	Dónde escribir los datos (el signo # indica que la operación se debe realizar en un archivo)
Expresión (#N27:3)	Dónde leer los datos (el signo # indica que la operación se debe realizar en un archivo)

Cuando el renglón se hace verdadero, el procesador lee cuatro elementos del archivo entero N27 palabra por palabra a partir del elemento 3 y escribe la imagen al archivo entero N28 a partir del elemento 0. Sobrescribe los datos en el archivo de destino.

Ejemplo de copiar archivo a palabra:



Este parámetro:	Indica al procesador:
Control (R6:6)	Qué estructura de control controla la operación
Longitud (5)	Copiar cinco palabras
Posición (0)	Comenzar en la dirección de origen
Modo (incremental)	Copiar una palabra cada vez que el renglón se hace verdadero
Destino (N29:5)	Dónde escribir los datos (dirección de palabra)
Expresión (#N29:0)	Dónde leer los datos (el signo # indica que la operación se debe realizar en un archivo)

Durante cada transición de renglón de falso a verdadero el procesador lee un elemento del archivo entero N29 a partir del elemento 0 y escribe la imagen en el elemento 5 del archivo entero N29. La instrucción sobrescribe los datos en el destino.

Un movimiento de palabra a archivo es similar excepto que la instrucción copia datos de una dirección de palabra en un archivo. La dirección de palabra puede ser el mismo archivo o un archivo diferente.

FAL Arithmetic Operations

Usted puede realizar múltiples operaciones aritméticas en los datos de archivo (número entero o punto flotante [coma flotante]) mediante los operadores siguientes:

Operador:	Significa:	Operador:	Significa:
+	sumar		dividir
-	restar	-	cambiar signo
*	multiplicar	0	poner a cero

Vea el capítulo 4 para obtener más información acerca del orden de operación.

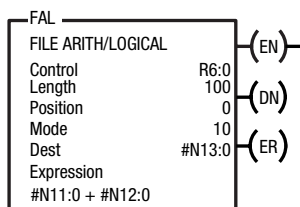
Límites superior e inferior

Los límites de los datos que se manipulan matemáticamente dependen del tipo de archivo en que los datos se almacenan. Las pautas siguientes son pertinentes:

- todos los datos, excepto el punto flotante (coma flotante), son números enteros con signo
- los valores negativos se almacenan en el complemento a dos
- los número de punto flotante (coma flotante) se formatean como subconjunto de punto flotante (coma flotante) de precisión única del IEEE

Tipo de archivo:	Rango almacenado en la palabra:
bit	-32,768 a +32,767 para números enteros
número entero	-32,768 a +32,767
temporizador	0 a +32,767
contador	-32,768 a +32,767
control	0 a +32,767
punto flotante (coma flotante)	$\pm 1.1754944e^{-38}$ a $\pm 3.4028237e^{+38}$

Un error ocurre cuando el resultado de una operación sobrepasa el límite superior o inferior de la palabra de destino en la cual se almacena. El bit de overflow se establece en el archivo de estado del procesador (S:0/1). La instrucción también establece el bit de error en el byte de estado de su palabra de control.

Ejemplo de sumar:

Cuando el renglón se hace verdadero, el procesador suma los 100 valores en el archivo #N11:0 y los valores correspondientes en el archivo #N12:0 mediante el modo numérico de 10 palabras por escán. La operación se realiza durante 10 escanes y la instrucción suma secuencialmente los valores en la expresión y almacena el resultado en el archivo #N13:0.

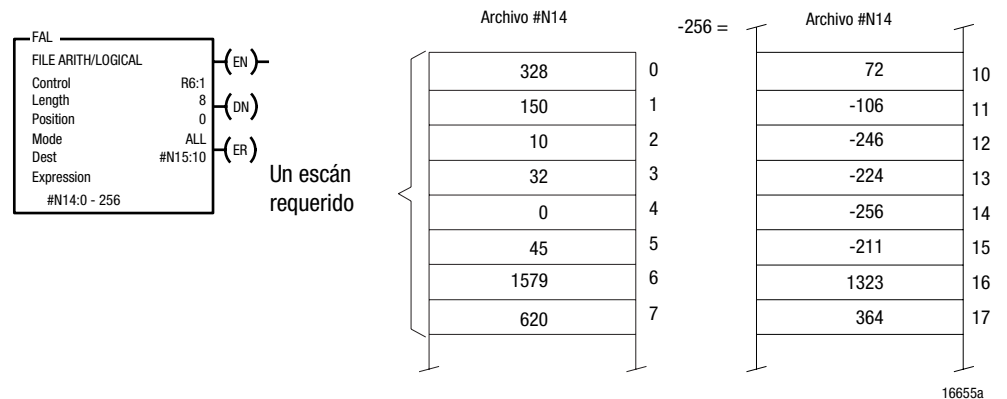
	Archivo #N11:0	+	Archivo #N12:0	=	Archivo #N13:0
1er escán	328 0		10 0		338 0
	150 1		32 1		182 1
	10 2		1 2		11 2
	32 3		147 3		179 3
	0 4		99 4		99 4
	45 5		572 5		617 5
	1579 6		300 6		1879 6
	620 7		42 7		662 7
	800 8		19 8		819 8
	1243 9		1000 9		2243 9
2° escán	las 10 palabras siguientes				
3er escán	las 10 palabras siguientes				
4° escán	las 10 palabras siguientes				
	...				
10° escán	últimos 10 elementos 99				

13386

Este parámetro:**Indica al procesador:**

Control (R6:0)	Qué estructura de control controla la operación
Longitud (100)	Operar en cien elementos
Posición (0)	Comenzar en la dirección de origen
Modo (10)	Ejecutar los datos en 10 palabras por escán
Destino (#N13:0)	Dónde escribir los datos del resultado
Expresión (#N11:0 + #N12:0)	Los operadores, constantes de programa y direcciones de origen

Ejemplo de restar:

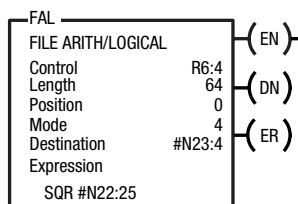


Este parámetro:

Indica al procesador:

Control (R6:1)	Qué estructura de control controla la operación
Longitud (8)	Operar en ocho palabras
Posición (0)	Comenzar en la dirección de origen
Modo (ALL)	Ejecutar los datos durante un escán de programa
Destino (#N15:10)	Dónde escribir los datos del resultado
Expresión (#N14:0 – 256)	Los operadores, constantes de programa y direcciones de origen

Cuando el renglón se hace verdadero, el procesador lee ocho elementos del archivo entero N14 palabra por palabra a partir del elemento 0, resta una constante de programa (256) de cada uno y escribe el resultado en el archivo de destino N15 a partir del elemento 10. Esta operación se realiza durante un escán.

Ejemplo de raíz cuadrada:

Cuando las condiciones se hacen verdaderas, la instrucción obtiene la raíz cuadrada positiva del valor en el origen. El modo seleccionado determina la velocidad. El resultado de cada operación de raíz cuadrada se almacena en la palabra correspondiente en el destino palabra por palabra.

El procesador halla la raíz cuadrada del valor absoluto (si el signo es negativo, el procesador ignora el signo).

Este parámetro:	Indica al procesador:
-----------------	-----------------------

Control (R6:4)	Qué estructura de control controla la operación
Longitud (64)	Hallar la raíz cuadrada de 64 palabras
Posición (0)	Comenzar en la dirección de origen
Modo (4)	Operar en 4 palabras durante cada escán
Destino (#N23:4)	Dónde escribir los datos del resultado
Expresión (SQR #N22:25)	El operador y la dirección de origen

Después de hacerse verdadero el renglón, se calcula la raíz cuadrada de las 4 primeras palabras en el archivo a partir de N22:25 y el resultado se escribe en el archivo de destino a partir de N23:4. De ahí en adelante, cada vez que el renglón se escanea, las cuatro próximas palabras se calculan y el resultado se escribe en el archivo de destino. El procesador requiere un total de 16 escanes (longitud = 16 / modo = 4) para completar la instrucción.

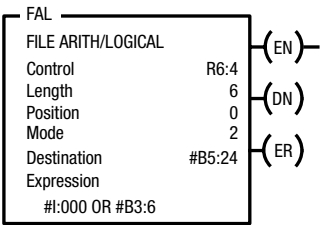
Operaciones lógicas FAL

Realice operaciones lógicas múltiples en datos de archivos binarios mediante los siguientes operadores lógicos bit a bit:

- AND
- OR
- XOR
- NOT

Para realizar operaciones lógicas múltiples, introduzca los operadores, las direcciones de origen o las constantes de programa en la expresión y la dirección del resultado en el destino.

Ejemplo de OR lógico:



	Archivo I:000	palabra	o	Archivo B3	palabra	o	Archivo B5	palabra
1er escán	0000000000000000	0		1010101010101010	6		1010101010101010	24
	1111111111111111	1		1111111100000000	7		1111111111111111	25
2° escán	1111000011110000	2		0000000000000000	8		1111000011110000	26
	1010101010101010	3		1100110011001100	9		1110111011101110	27
3er escán		4			10			28
		5			11			29

16618a

Este parámetro:	Indica al procesador:
Control (R6:4)	Qué estructura de control controla la operación
Longitud (6)	Realizar la operación OR en 6 palabras
Posición (0)	Comenzar en la dirección de origen
Modo (2)	Mover 2 palabras durante cada escán
Destino (#B5:24)	Dónde escribir los datos del resultado
Expresión (#I:000 OR #B3:6)	Los operadores y las direcciones de origen

Después de hacerse verdadero el renglón, el procesador realiza una operación OR lógica en dos palabras a partir de I:0 y B3:6. El resultado se escribe en el archivo de destino a partir de B5:24. De ahí en adelante, cada vez que el renglón se escanea, las dos próximas palabras se calculan y el resultado se escribe en el archivo de destino. El procesador requiere un total de 3 escanes (longitud = 6 / modo = 2) para completar la instrucción.

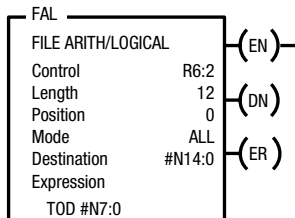
El procesador ejecuta los operadores lógicos en un orden predeterminado. Vea el capítulo 4 para obtener más información acerca del orden de operaciones.

Operaciones de conversión FAL

La instrucción FAL puede realizar estas operaciones de conversión:

- convertir de número entero en BCD (TOD)
- convertir de BCD en número entero (FRD)

Ejemplo: convertir en BCD



Cuando las condiciones del renglón se hacen verdaderas, el procesador convierte el valor en el origen de un número entero en BCD. El modo seleccionado determina la velocidad de conversión. El resultado de la operación se almacena en la palabra correspondiente en el destino.

Ejemplo: convertir de BCD

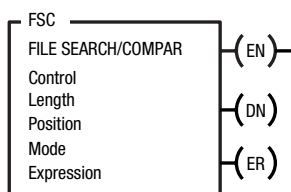
Cuando las condiciones del renglón se hacen verdaderas, el procesador convierte el valor en el origen de BCD en un número entero. El modo seleccionado determina la velocidad de conversión. El resultado de la operación se almacena en la palabra correspondiente en el destino.

Importante: Convierta los valores BCD en números enteros antes de manipularlos. Si no convierte los valores, el procesador los manipula como números enteros y el valor BCD se pierde.

File Search and Compare (FSC)

La instrucción FSC realiza las operaciones de búsqueda y comparación. Estas son las mismas operaciones de la instrucción CMP, incluso las expresiones complejas (procesadores PLC-5 con características mejoradas solamente). La diferencia es que la instrucción FSC realiza operaciones lógicas en los archivos y la instrucción CMP opera en una sola palabra. Además, la instrucción FSC es una instrucción de salida. En cambio, la instrucción CMP es una instrucción de entrada.

Descripción:



La instrucción FSC es una instrucción de salida que compara los valores en los archivos de origen, palabra por palabra, para las operaciones lógicas que usted especifica en la expresión. Cuando el procesador determina que la comparación especificada es verdadera, establece el bit de encontrado .FD y registra la posición .POS donde se encontró la comparación verdadera. El bit de inhibición .IN se establece para evitar más búsquedas de los archivos.

Es necesario que el programa de lógica de escalera examine el bit de encontrado .FD y la posición .POS para tomar la acción apropiada. Restablezca el bit de inhibición .IN para que la instrucción continúe.

Seleccione cómo el procesador distribuye la operación durante uno o más escanes de programa según el modo de instrucción. Vea el capítulo 8 para obtener más información acerca de los modos de operación de archivo.

Use esta instrucción para realizar operaciones tales como:

- establecer alarmas bajas y altas de proceso para entradas múltiples analógicas
- comparar variables de lotes con un archivo de referencia antes de comenzar una operación de lotes

Cómo usar los bits de estado

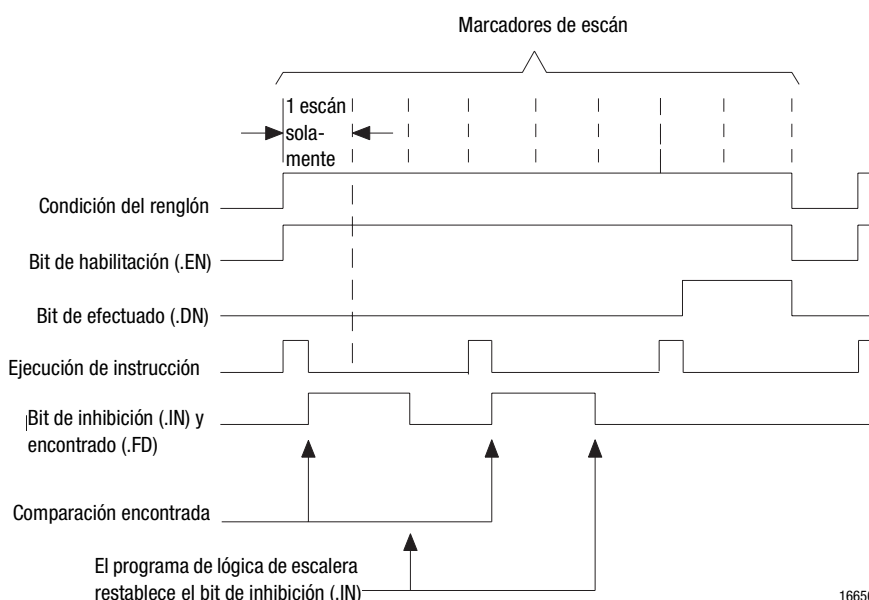
Para usar la instrucción FSC correctamente, el programa de lógica de escalera debe examinar y controlar los bits de estado en la estructura de control. Estos bits se deben direccionar mediante mnemónicos.

Este bit:	Se establece:
Habilitación .EN (bit 15)	mediante una transición de renglón de falso a verdadero e indica que la instrucción está habilitada. En el modo incremental, este bit sigue a la condición del renglón. En los modos Numérico y Todo, este bit permanece establecido hasta que la instrucción completa su operación, independientemente de la condición del renglón. El bit .EN se restablece cuando las condiciones del renglón se hacen falsas, pero solamente después de que la instrucción haya establecido el bit .DN.
Efectuado .DN (bit 13)	después de que la instrucción ha operado en el último conjunto de palabras. Si la instrucción es falsa al completarse la operación en el modo numérico, la instrucción restablece el bit .DN un escán de programa después de completarse la operación. Si la instrucción es verdadera al completarse la operación, el bit .DN se restablece cuando la instrucción se hace falsa.
Error .ER (bit 11)	cuando la operación genera un overflow. La instrucción se detiene hasta que el programa de lógica de escalera restablece este bit. Cuando el procesador detecta un error, el valor de posición almacena el número del elemento que falló.
Inhibición .IN (bit 9)	cuando el procesador detecta una comparación verdadera. El programa de lógica de escalera debe restablecer este bit para continuar la búsqueda después de tomar una acción iniciada por el examen del bit .FD. El programa de lógica de escalera debe restablecer este bit para continuar la operación.
Encontrado .FD (bit 8)	cuando el procesador detecta una comparación verdadera. El procesador bloquea la búsqueda y establece el bit de inhibición .IN. El bit .FD es la salida de la instrucción FSC.

Con la instrucción FSC, se puede mostrar en pantalla un máximo de 80 caracteres de la expresión. Si la expresión introducida está cerca del máximo de 80 caracteres, es posible que el procesador amplíe este máximo para el renglón que contiene la instrucción cuando se acepte dicho renglón. Cuando usted intenta editar la expresión se muestran en pantalla solamente los 80 primeros caracteres y el renglón se muestra en pantalla como renglón de error. Sin embargo, el procesador contiene la expresión completa y la instrucción funciona correctamente.

Para evitar este problema de visualización en pantalla, exporte el archivo de memoria del procesador y efectúe las ediciones en el archivo de texto PC5. Luego importe este archivo de texto. Vea el manual de programación para obtener más información acerca de cómo importar/exportar los archivos de memoria del procesador.

El siguiente diagrama de temporización para el modo Todo muestra las relaciones entre los bits de estado y la ejecución de la instrucción cuando la instrucción encuentra dos condiciones verdaderas.



16656

Vea los diagramas de temporización en el capítulo 8 para obtener más información acerca de cómo la instrucción FSC responde cuando no encuentra comparaciones verdaderas.

Operaciones de búsqueda y comparación FSC

La instrucción FSC realiza estas comparaciones de datos de archivo según la manera en que usted las especifica en la expresión. (Las expresiones complejas son válidas en los procesadores PLC-5 con características mejoradas solamente.)

Comparación:	Ejemplo de expresión:
Buscar igual	#N50:0 = #N51:0
Buscar diferente	#N52:0 <> N52:11
Buscar menor que	#B3:100 < #N53:0
Buscar menor o igual	#F60:0 <= F60:12
Buscar mayor que	#N54:0 > 256
Buscar mayor o igual	F60:10 >= #N61:0

Conversión de datos

El procesador compara archivos de diferentes tipos de datos convirtiendo internamente los datos en su equivalencia binaria antes de realizar la comparación. El procesador manipula los siguientes tipos de datos como números enteros:

Importante: Cuando compare los valores de punto flotante (coma flotante) y números enteros en la instrucción FSC, restrinja las comparaciones a “menor o igual” y “mayor o igual”.

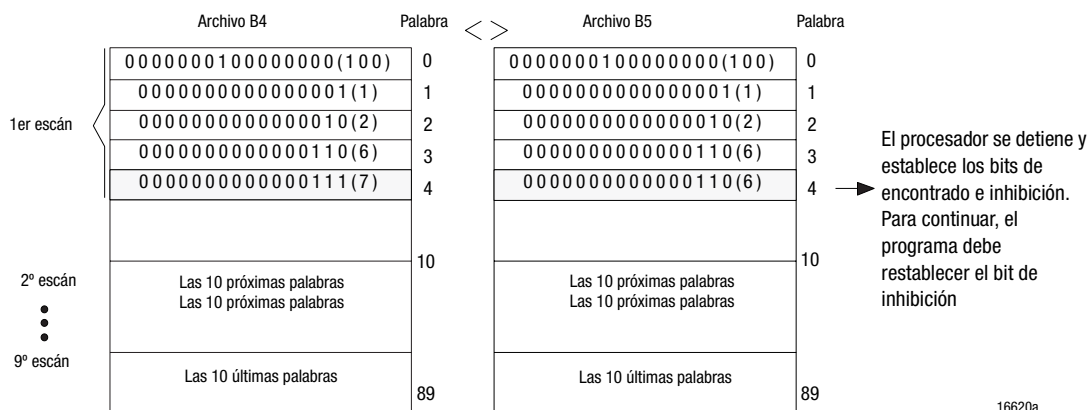
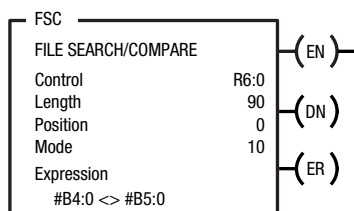
Importante: Use ASCII y BCD para la visualización en pantalla solamente y no como valores. Puesto que el procesador los interpreta como números enteros, es posible que pierdan su significado si se introducen como valores.

Vea la sección “Cómo determinar el orden de operación” en el capítulo 4 para obtener información acerca del orden en que la instrucción realiza las operaciones lógicas.

Operación de búsqueda de archivo

Cuando la condición del renglón se hace verdadera, la comparación deseada se realiza en los datos direccionados en la expresión. Las palabras se comparan en orden ascendente a partir del inicio. El modo de operación especificado determina la velocidad de comparación.

El bit de efectuado .DN (bit 13) se establece después de que el procesador ha comparado la última pareja. Si el renglón es verdadero al completarse la operación, el bit .DN se desactiva cuando el renglón ya no es verdadero. No obstante, en el modo numérico, si el renglón no es verdadero al completarse la operación, el bit .DN permanece activado durante un escán de programa después de completarse la operación.

Ejemplo de buscar diferente:

16620a

Este parámetro: Indica al procesador:

Control (R6:0)	Qué estructura de control controla la operación
Longitud (90)	Buscar en 90 palabras
Posición (0)	Comenzar en las direcciones de origen
Modo (10)	Buscar en 10 palabras por escán de programa
Expresión (#B4:0 <> #B5:0)	La comparación a realizar y las direcciones de origen

Cuando un renglón que contiene la instrucción FSC se hace verdadero, el procesador realiza la comparación de diferente a palabras a partir de B4:0 y B:50. El modo seleccionado determina el número de palabras comparadas por escán de programa (10 en este ejemplo).

Cuando el procesador determina que las palabras de origen correspondientes no son iguales (palabras B4:4 y B5:4 en este ejemplo), el procesador detiene la búsqueda y activa los bits de encontrado .FD e inhibición .IN para que el programa de lógica de escalera tome la acción apropiada. Hay que desactivar el bit .IN para continuar la comparación de búsquedas.

File Copy (COP)

Descripción:

COP
COPY FILE
Source
Destination
Length

La instrucción COP es una instrucción de salida que copia los valores en el archivo de origen al archivo de destino. No se cambia el origen. La instrucción COP no usa los bits de estado. Si es necesario habilitar un bit, programe una salida en paralelo que use una dirección de almacenamiento.

La instrucción COP no sobrescribe los límites del archivo. Se pierden los datos de overflow. Además, no ocurre una conversión de datos si los archivos de origen y destino tienen diferentes tipos de datos. Use archivos con el mismo tipo de datos para cada uno.

Si el destino se encuentra en un archivo de palabras (tal como un archivo de números enteros), especifique la longitud de las palabras. Si el destino se encuentra en un archivo de estructuras (tal como un archivo de contador), especifique la longitud en las estructuras. Por ejemplo, si el origen se encuentra en un archivo entero, el destino se encuentra en un archivo de contador y usted especifica una longitud de 5, hay 15 palabras enteras que se copian a 5 estructuras de contador.

Cómo introducir los parámetros

Es necesario proporcionar al procesador la información siguiente para programar la instrucción COP:

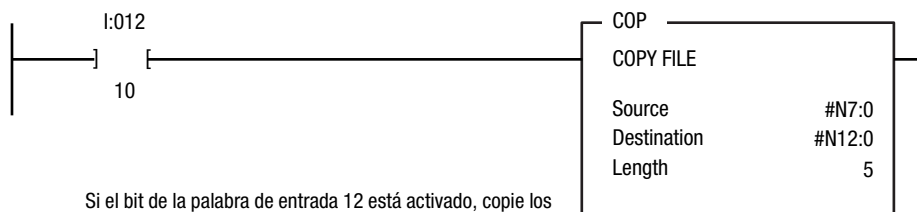
Parámetro:	Definición:
Origen	la dirección de inicio del archivo de origen. No se cambia el origen.
Destino	la dirección del archivo de destino. La instrucción sobrescribe los datos ya almacenados en el destino.
Longitud	el número de palabras/estructuras que se deben sobrescribir en el archivo de destino.



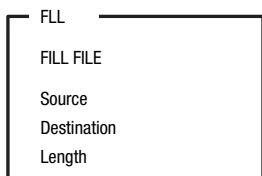
ATENCION: Si usa la instrucción COP con un procesador PLC-5 con características mejoradas, serie A-D, es posible que los límites de archivo se crucen si el parámetro de destino se direcciona indirectamente.

Si la dirección indirecta se escribe a la zona de programas, el procesador PLC-5 con características mejoradas, serie A-D, muestra el código de fallo mayor 11 (suma de comprobación incorrecta de programa del usuario). Si la dirección indirecta se escribe fuera de la zona de programas, pueden ocurrir resultados inesperados.

Si usa la instrucción COP con los procesadores PLC-5 con características mejoradas, serie E y posteriores, esta condición está identificada correctamente por el código de fallo mayor 20 (dirección indirecta fuera del rango alto) o el código de fallo mayor 21 (dirección indirecta fuera del rango bajo).

Ejemplo:

Si el bit de la palabra de entrada 12 está activado, copie los valores de las cinco primeras palabras a partir de N7:0 a las cinco primeras palabras de N12:0.

File Fill (FLL)**Descripción:**

La instrucción FLL es una instrucción de salida que llena las palabras de un archivo con un valor de origen. No se cambia el origen. La instrucción FLL no usa bits de estados. Si es necesario habilitar un bit, programe una salida en paralelo que use una dirección de almacenamiento.

La instrucción FLL no sobrescribe los límites del archivo. Se pierden los datos de overflow. Además, no ocurre una conversión de datos si los archivos de origen y destino tienen diferentes tipos de datos. Use archivos con el mismo tipo de datos para cada uno.

Si el destino se encuentra en un archivo de palabras (tal como un archivo de números enteros), especifique la longitud de las palabras. Si el destino se encuentra en un archivo de estructuras (tal como un archivo de contador), especifique la longitud en las estructuras. Por ejemplo, si el origen se encuentra en un archivo entero, el destino se encuentra en un archivo de contador y usted especifica una longitud de 5, la palabra de origen se copia 15 veces para llenar las 5 estructuras de contador.

La instrucción es sensible a los niveles.

Cómo introducir los parámetros

Es necesario proporcionar al procesador la información siguiente para programar la instrucción FLL:

Parámetro:	Definición:
Origen	la dirección de la palabra de origen o una constante de programa. No se cambia el origen.
Destino	la dirección de inicio del archivo de destino. La instrucción sobrescribe los datos ya almacenados en el destino.
Longitud	el número de palabras/estructuras que se deben llenar en el archivo de destino.

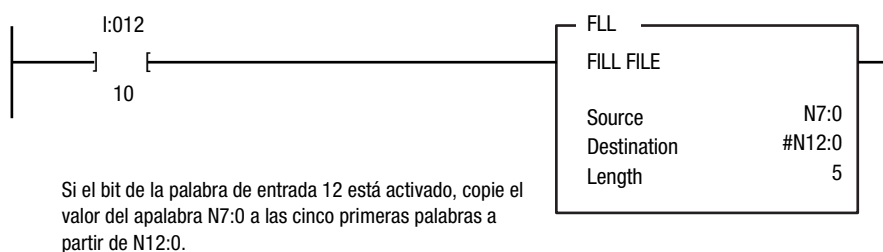


ATENCIÓN: Si usa la instrucción FLL con un procesador PLC-5 con características mejoradas, serie A-D, es posible que los límites de archivo se crucen si el parámetro de destino se direcciona indirectamente.

Si la dirección indirecta se escribe a la zona de programas, el procesador PLC-5 con características mejoradas, serie A-D, muestra el código de fallo mayor 11 (suma de comprobación incorrecta de programa del usuario). Si la dirección indirecta se escribe fuera de la zona de programas, pueden ocurrir resultados inesperados.

Si usa la instrucción FLL con los procesadores PLC-5 con características mejoradas, serie E y posteriores, esta condición está identificada correctamente por el código de fallo mayor 20 (dirección indirecta fuera del rango alto) o el código de fallo mayor 21 (dirección indirecta fuera del rango bajo).

Ejemplo:



Las palabras se copian del archivo de origen especificado al archivo de destino especificado durante cada escán en que el renglón es verdadero. Se copian (en orden ascendente sin transformación de datos) hasta el número especificado o hasta que se alcanza la última palabra del archivo de destino, cualquiera que ocurra primero.

Especifique precisamente la dirección de inicio y la longitud del bloque de datos que llena. La instrucción no sobrescribe un límite de archivo (tal como entre los archivos N16 y N17) en el destino. El overflow se perderá.

Notas:

Instrucciones de diagnóstico FBC, DDT, DTR

Cómo usar las instrucciones de diagnóstico

Las instrucciones de diagnóstico le permiten detectar problemas con los datos en los programas. La Tabla 10.A indica las instrucciones de diagnóstico disponibles.

Tabla 10.A
Instrucciones de diagnóstico disponibles

Si desea:	Use esta operación:	Que se encuentra en la página:
Comparar datos de E/S con una buena referencia y registrar las desigualdades	FBC	10-2
Comparar datos de E/S con una buena referencia, registrar las desigualdades y actualizar el archivo de referencia para que coincida con el archivo de origen	DDT	10-2
Pasar datos de origen por una máscara y comparar el resultado con datos de referencia y escribir la palabra de origen en la dirección de referencia de la próxima comparación.	DTR	10-8

Vea el Apéndice C para obtener más información acerca de los operandos (y los tipos/valores de datos válidos de cada operando) usados por las instrucciones descirtas en este capítulo.

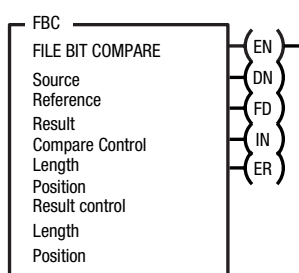
File Bit Comparison (FBC) y Diagnostic Detect (DDT)

Las instrucciones de diagnóstico FBC y DDT son instrucciones de salida que se usan para monitorear la operación de máquinas o procesos a fin de detectar malas funciones.

Tabla 10.B
Instrucciones de diagnóstico disponibles

Si desea detectar malas funciones de esta manera:	Use esta instrucción:
Comparando los bits en un archivo de entradas en tiempo real con un archivo de bit de referencia que representa la operación correcta	FBC
Diagnósticos de cambio de estado	DDT

Descripción:



Las instrucciones FBC y DDT comparan bits en un archivo de valores en tiempo real de máquinas o procesos (archivo de entrada) con bits en un archivo de referencia, detectan las desviaciones y registran los números de bits desiguales. Estas instrucciones registran la posición de cada desigualdad encontrada y colocan esta información en el archivo del resultado. Si no se encuentran desigualdades, se establece el bit .DN, pero no se cambia el archivo del resultado.

La diferencia entre la instrucción DDT y la instrucción FBC es que cada vez que la instrucción DDT encuentra una desigualdad, el procesador cambia el bit de referencia para que coincida con el bit de origen. La instrucción FBC no cambia el bit de referencia. Use la instrucción DDT para actualizar el archivo de referencia a fin de reflejar el cambio en las condiciones de las máquinas o procesos.

Cómo seleccionar el modo búsqueda

Haga una selección para determinar si la instrucción de diagnóstico busca una desigualdad una por una o si busca todas las desigualdades durante un escán de programa.

Una desigualdad a la vez

Durante cada transición del renglón de falso a verdadero, la instrucción busca la próxima desigualdad entre los archivos de entrada y referencia. Cuando encuentra una desigualdad, la instrucción se detiene y establece el bit de encontrado .FD. La instrucción introduce el número de posición de la desigualdad en el archivo de resultados.

La instrucción DDT también cambia el estado del bit de referencia para que coincida con el estado del bit de entrada correspondiente. La instrucción restablece el bit de encontrado cuando el renglón se hace falso.

Cuando la instrucción llega al fin del archivo, se establece el bit de efectuado (bit 13 DN del elemento de control de comparación).

Cuando el renglón se hace falso, la instrucción restablece:

- el bit de habilitación
- el bit de encontrado (si está establecido)
- el bit de efectuado de comparación
- el bit de efectuado de resultado (si está establecido)
- los dos contadores de control

Para habilitar este modo de operación, establezca el bit de inhibición (.IN = 1) mediante el programa de lógica de escalera o manualmente antes de la ejecución del programa.

Todas las desigualdades durante un escán

La instrucción busca todas las desigualdades entre los archivos de entrada y referencia durante un escán de programa. Cuando encuentra desigualdades, la instrucción introduce los números de posición de los bits desiguales en el archivo de resultados según el orden en que los encontró. Después de llegar al fin de los archivos de entrada y referencia, la instrucción establece el bit .FD si encuentra por lo menos una desigualdad. La instrucción establece el bit .DN.

Si usa un archivo de resultados que no puede contener todas las desigualdades detectadas (si el archivo de resultados se llena), la instrucción se detiene y requiere otra transición del renglón de falso a verdadero para continuar la operación. La instrucción continúa con las nuevas posiciones de bits desiguales en el inicio del archivo del resultado y sobrescribe las posiciones anteriores.

La instrucción establece lo siguiente después de completar la comparación y cuando el renglón se hace falso:

- el bit de habilitación
- el bit de encontrado (si está establecido)
- el bit de efectuado de comparación
- el bit de efectuado de resultado (si está establecido)
- los dos contadores de control

Para habilitar este modo de operación, restablezca el bit de inhibición (.IN = 0) mediante el programa de lógica de escalera o manualmente antes de la ejecución del programa.

Cómo introducir los parámetros

Es necesario proporcionar al procesador la información siguiente para programar estas instrucciones:

Parámetro:	Descripción:
Origen	la dirección indexada del archivo de entrada.
Referencia	la dirección indexada del archivo que contiene los datos con los cuales usted compara el archivo de entrada.
Resultado	la dirección indexada del archivo en que la instrucción almacena el número (bit) de posición de cada desigualdad detectada.
Control de comparación	la dirección de la estructura de control de comparación (R) que almacena los bits de estado, la longitud de los archivos de origen y referencia (los dos deben ser iguales) y la posición actual durante la operación. Use la dirección de control de comparación mediante mnemónicos cuando direcciona estos parámetros: Longitud (.LEN) es el número decimal de los bits que se deben comparar en los archivos de origen y referencia. Tome en cuenta de que los bits en los archivos de E/S se enumeran octalmente 00-17, pero los bits en todos los demás archivos se enumeran decimalmente 0-15. Posición (.POS) es la posición actual del bit que la instrucción indica. Introduzca un valor solamente si desea que la instrucción comience en un offset concurrente con un offset de archivo de control durante un escán.
Control de resultado	la dirección de la estructura de control del resultado (R) que almacena el número de posición del bit cada vez que la instrucción encuentra una desigualdad entre los archivos de origen y referencia.

Use la dirección de control del resultado con un mnemónico cuando direcciona estos parámetros:

- **La longitud (.LEN)** es el número decimal de elementos en el archivo de resultados. Cree una longitud suficiente para registrar el número máximo de desigualdades pronosticadas.
- **La posición (.POS)** es la posición actual en el archivo de resultados. Introduzca un valor solamente si desea que la instrucción comience en un offset concurrente con un offset de archivo de control durante un escán.



ATENCION: No use la misma dirección para más de una estructura de control. La duplicación de estas direcciones puede resultar en una operación inesperada, lo cual puede causar daños al equipo y/o lesiones corporales.

Cómo usar los bits de estado

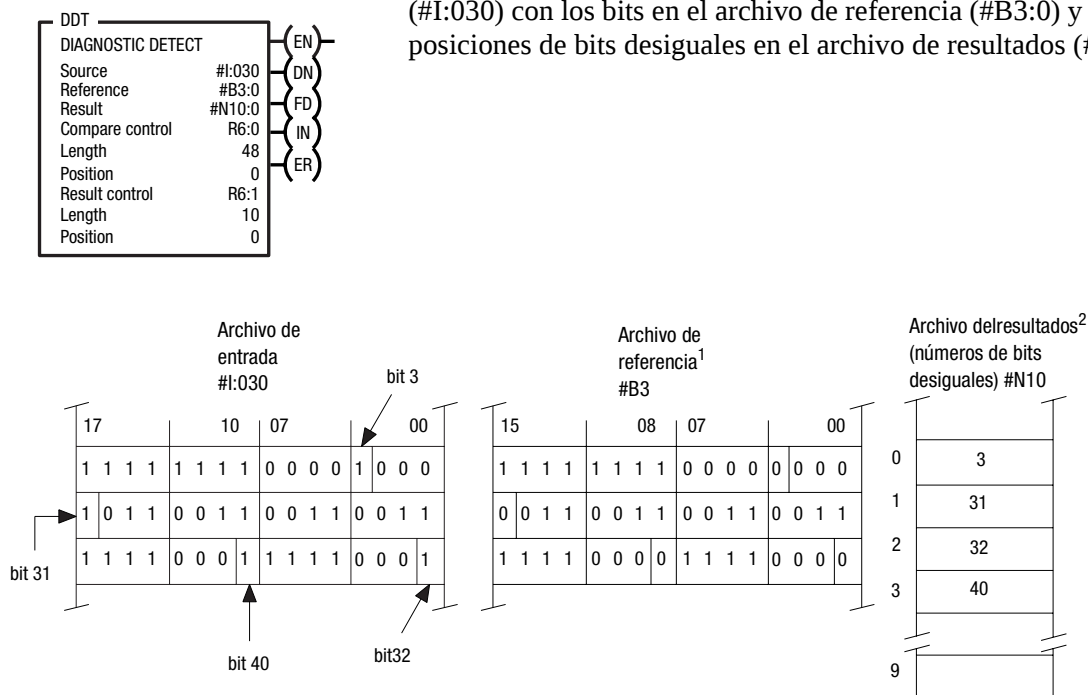
Para usar las instrucciones FBC y DDT correctamente, examine y controle los bits en los elementos de control de comparación y resultado. Estos bits se direccionan mediante mnemónicos.

Bit:	Función:	
Bits de control de comparación	Habilitación .EN (bit 15)	inicia la operación durante una transición del renglón de falso a verdadero Si el bit .IN está establecido para la operación individual, el programa de lógica de escalera debe alternar el bit .EN después de que la instrucción detecte cada desigualdad.
	Efectuado .DN (bit 13)	se establece cuando el procesador llega al fin de los archivos de origen y referencia
	Error .ER (bit 11)	se establece cuando el procesador detecta un error y detiene la operación de la instrucción Por ejemplo, ocurre un error si la longitud (.LEN) es menor o igual a cero o si la posición (.POS) es menor que cero. El programa de lógica de escalera debe restablecer el bit .ER si la instrucción detecta un error.
	Inhibición .IN (bit 09)	determina el modo de operación Cuando este bit está restablecido, el procesador detecta todas las desigualdades durante un escán. Cuando este bit está establecido, el procesador detiene la búsqueda en cada desigualdad y espera hasta que el programa de lógica de escalera vuelva a habilitar la instrucción antes de continuar la búsqueda.
Bits de control de resultado	Encontrado .FD (bit 08)	se establece cada vez que el procesador registra un número de bit desigual en el archivo del resultado (operación de una a la vez) o después de registrar todas las desigualdades (todas por escán).
	Efectuado .DN (bit 13)	se establece cuando se llena el archivo de resultados La instrucción se detiene y requiere otra transición de renglón de falso a verdadero para restablecer el bit .DN del resultado y, luego, continúa. Si la instrucción encuentra otra desigualdad, continúa con el nuevo número de posición al inicio del archivo y sobrescribe los números de posición anteriores.

Después de que la instrucción FBC o DDT establece el bit .DN de comparación, la instrucción se restablece cuando las condiciones del renglón se hacen falsas. La instrucción restablece sus bits de estado y los dos elementos de control.

Ejemplo:

La instrucción DDT anterior compara los bits en el archivo de origen (#I:030) con los bits en el archivo de referencia (#B3:0) y registra las posiciones de bits desiguales en el archivo de resultados (#N10:0).



Las instrucciones FBC y DDT detectan desigualdades y registran la ubicación de las mismas según el número de bit en un archivo del resultado.

¹ La instrucción DDT cambia el estado del bit correspondiente en el archivo de referencia para que coincida con el archivo de entrada cuando detecta una desigualdad.

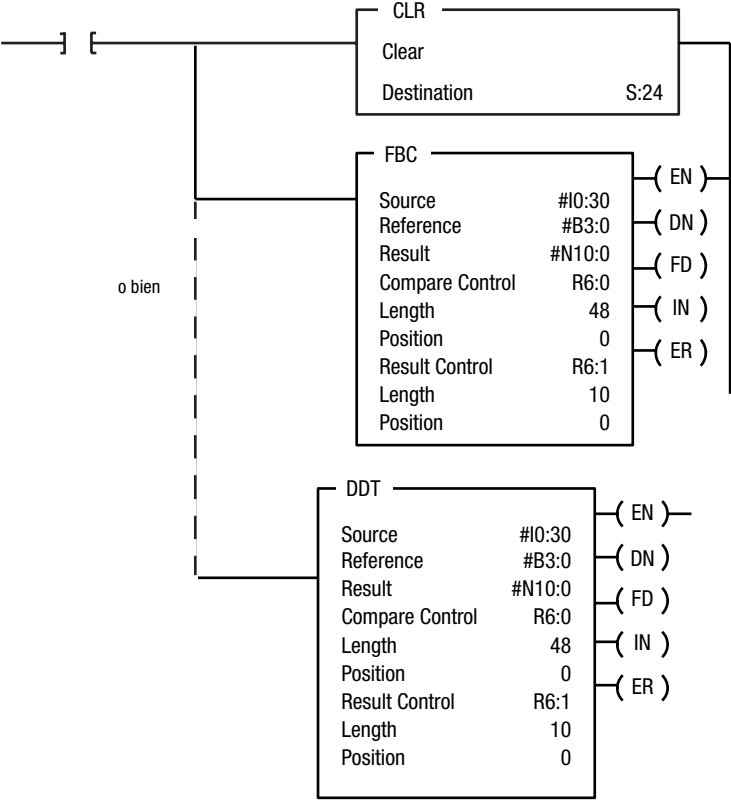
² La longitud del archivo del resultado es la longitud que introduce para el CONTROL DEL RESULTADO.

16657a

Este parámetro:**Indica al procesador:**

Origen (#I:030)	Dónde encontrar los datos de entrada para la comparación
Referencia (#B3:0)	Dónde encontrar el archivo de referencia
Resultado (#N10:0)	Dónde encontrar los números de bits desiguales
Control de comparación (R6:0)	Qué estructura de control controla la comparación
Longitud (48)	El número de bits que se deben comparar
Posición (0)	Comenzar en el inicio del archivo
Control del resultado (R6:1)	Qué estructura de control controla la comparación
Longitud (10)	El número de palabras reservadas para desigualdades
Posición (0)	Comenzar en el inicio del archivo

Importante: Las instrucciones FBC y DDT pueden causar el fallo de cualquier procesador PLC-5 con características mejoradas si el offset de direccionamiento indexado contiene un valor que excede los límites de la tabla de datos. Para evitar este problema, añada un renglón de escalera que borre S:24 (offset de direccionamiento indexado) justo antes de una instrucción FBC o DDT.



Data Transitional (DTR)

Descripción:

DTR
DATA TRANSITION
Source
Mask
Reference

La instrucción DTR es una instrucción de entrada que pasa un valor de origen por una máscara y compara el resultado con un valor de referencia. Use esta instrucción para detectar e identificar entradas no válidas y evitar que las entradas no válidas desactiven un procesador de lotes o la operación de máquinas.

La instrucción DTR compara una palabra de origen mediante una máscara con una palabra de referencia. La instrucción también escribe la palabra de origen en la dirección de referencia para la próxima comparación. La palabra de origen no se cambia.

Cuando el origen enmascarado difiere de la referencia, la instrucción se hace verdadera durante un solo escán. El procesador escribe el valor de origen enmascarado en la dirección de referencia. Cuando el origen enmascarado y la referencia son iguales, la instrucción permanece falsa.



ATENCIÓN: La programación en línea con esta instrucción puede ser peligrosa. Si el valor de destino es diferente del valor de origen, la instrucción se hace verdadera. Tenga mucho cuidado al insertar esta instrucción cuando el procesador esté en el modo marchado o marcha remota.

Cómo introducir los parámetros

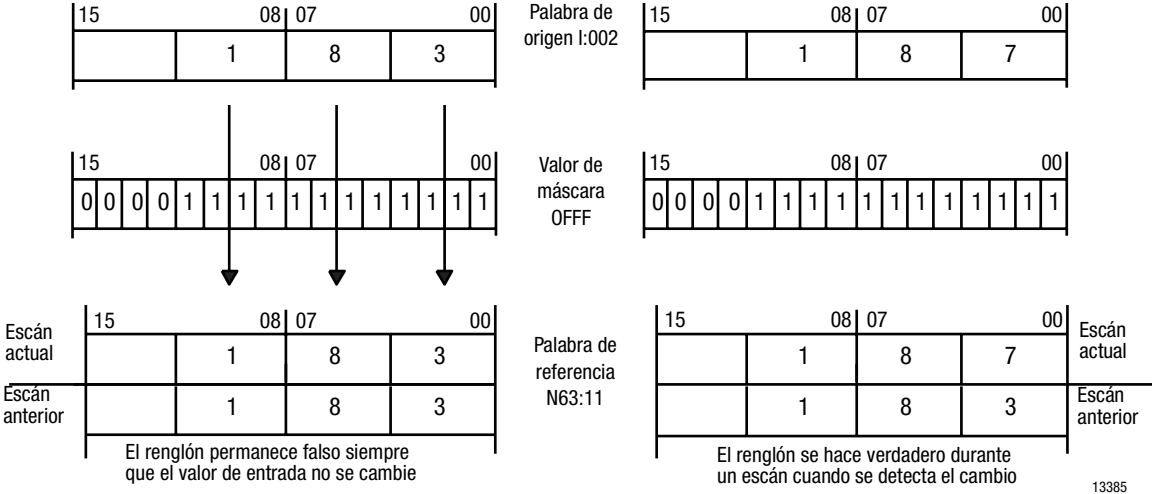
Es necesario proporcionar al procesador la información siguiente para programar la instrucción DTR:

Parámetro:	Definición:
Origen	la dirección de la palabra de entrada, típicamente entradas reales.
Máscara	el valor hexadecimal o la dirección que contiene el valor enmascarado
Referencia	la dirección de la palabra de referencia La referencia contiene los datos de origen del último escán DTR

Ejemplo:

DTR	
DATA TRANSITION	
Source	I:002
Mask	0FFF
Reference	N63:11

La instrucción DTR anterior pasa el origen (I:002) por una máscara de 0FFF y compara el resultado con la palabra de referencia (N63:11). La palabra de origen se escribe en la dirección de referencia para la próxima comparación (no se cambia el origen).



13385

Notas:

Instrucciones de registro de desplazamiento BSL, BSR, FFL, FFU, LFL, LFU

Cómo aplicar los registros de desplazamiento

Use la instrucción de registro de desplazamiento para simular el movimiento o el flujo de piezas e información.

Si usa un registro de desplazamiento para:	Los datos en el registro de desplazamiento pueden representar:
Seguir las piezas en una línea de ensamblaje	Tipos, calidad, tamaño y estado de las piezas
Controlar las operaciones de máquinas o procesos	El orden en que ocurren los eventos
El control de inventario	Los números de identificación o ubicaciones
Los diagnósticos de sistemas	Una condición de fallo que causó una interrupción

La Tabla 11.A indica las instrucciones de desplazamiento disponibles.

Tabla 11.A
Instrucciones de desplazamiento disponibles

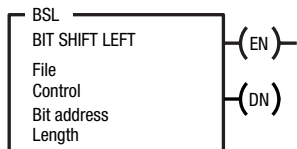
Si desea:	Use estas instrucciones:	Que se encuentra en la página:
Cargar bits, desplazar bits y descargar bits de un registro de bits individualmente, por ejemplo, para realizar el seguimiento de botellas a través de una línea de embotellado en que cada bit representa una botella	BSL, BSR	11-2
Cargar y descargar valores en el mismo orden, por ejemplo, para realizar el seguimiento de piezas a través de una línea de ensamblaje en que las piezas están representadas por valores que tienen un número de pieza y un código de ensamblaje	FFL, FFU	11-5
Cargar y descargar valores en orden inverso, por ejemplo, para realizar el seguimiento de el inventario apilado en un almacén donde las mercancías están representadas por números de serie y códigos de inventario	LFL, LFU *	11-8

* Estas instrucciones son compatibles solamente con los procesadores PLC-5 con características mejoradas.

Vea el Apéndice C para obtener más información acerca de los operandos (y los tipos/valores de datos válidos de cada operando) usados por las instrucciones descritas en este capítulo.

Cómo usar las instrucciones de desplazamiento de bit

Descripción:



Las instrucciones de desplazamiento de bits desplazan todos los bits dentro de la dirección especificada una posición de bit durante cada transición del renglón de falso a verdadero. Estas instrucciones son:

- Desplazamiento de bits a la izquierda (BSL)
- Desplazamiento de bits a la derecha (BSR)

Cómo introducir los parámetros

Es necesario proporcionar al procesador la información siguiente para programar una instrucción de desplazamiento de bits.

Parámetro	Definición:
Archivo	la dirección del registro de bit que desea manipular. Hay que iniciar el registro en un límite de palabras de 16 bits. Por ejemplo, use el bit 0 del número de palabra 1, 2, 3, etc. Puede terminar el registro en cualquier número de bit hasta 15,999. Sin embargo, no puede usar los bits restantes en ese elemento puesto que la instrucción los hace no válidos.
Control	La dirección de la estructura de control (48 bits – tres palabras de 16 bits) en la zona de control (R) de la memoria que almacena los bits de estado, el tamaño del registro (número de bits) y el indicador de bits de la instrucción.
Posición	la posición actual del bit que la instrucción indica. Introduzca un valor solamente si desea que la instrucción comience en un offset concurrente con un offset de archivo de control durante un escán. Use la dirección de control con un mnemónico cuando dirija este parámetro.
Dirección de bit	la dirección del bit de origen. La instrucción inserta el estado de este bit en la primera posición (inferior) del bit (para la instrucción BSL) o en la última posición (superior) del bit (para la instrucción BSR) en el registro.
Longitud	el número decimal de bits que se deben desplazar. Tome en cuenta que los bits en los archivos de E/S se enumeran octalmente 00-17, pero los bits en todos los otros archivos se enumeran decimalmente 0-15. Use la dirección de control con un mnemónico cuando dirija este parámetro.



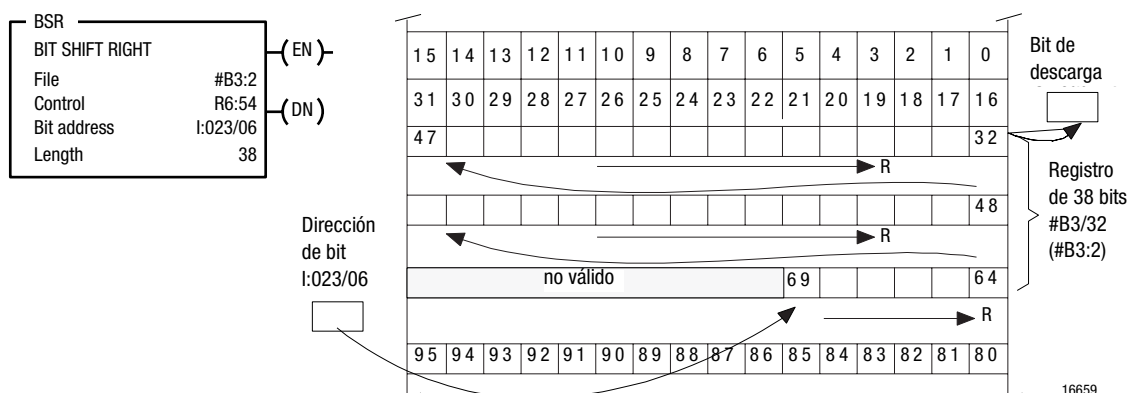
ATENCIÓN: No use la misma dirección de control para más de una instrucción. Puede resultar en una operación inesperada, lo cual podría causar daños al equipo y/o lesiones corporales.

Cuando un renglón que contiene la instrucción BSL va de falso a verdadero, el procesador establece el bit .EN. El procesador desplaza 58 bits en el archivo de bit B3, a partir del bit 16, a la izquierda (número de bit superior) una posición de bit. El último bit se desplaza fuera de la posición de bit 73 en el bit .UL. El bit de origen especificado, el bit 12 de la palabra de entrada 22, se desplaza a la primera posición de bit, el bit 16 del archivo de bit B3.

Después de que el procesador complete la operación de desplazamiento durante un escán de programa y cuando el renglón se hace falso, la instrucción restablece los bits .EN, .ER (si están establecidos) y .DN y restablece el indicador.

Para la operación de dar la vuelta, asegúrese de que la dirección de origen sea la misma que la dirección de bit superior (salida). No es necesario usar el bit .UL en la operación de dar la vuelta.

Ejemplo de desplazamiento de bit a la derecha (BSR):



Este parámetro:

Indica al procesador:

Archivo (#B3:2)	La ubicación del registro de bit
Control (R6:54)	La dirección y elemento de control de la instrucción
Dirección de bit (I:023/06)	La dirección de bit de origen (bit 06 en la palabra de entrada 23)
Longitud (38)	El número de bits en el registro de bit

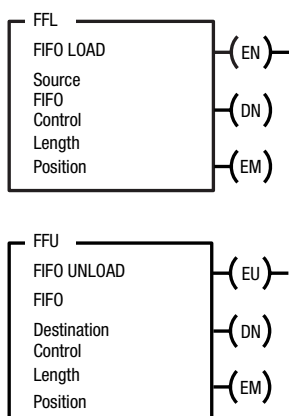
Cuando un renglón que contiene la instrucción de BSR va de falso a verdadero, el procesador establece el bit .EN. El procesador desplaza 38 bits en el archivo de bit B3 a la derecha (a un número de bit inferior) una posición de bit a partir de la posición de bit más alta 69. El bit inferior (bit 32) se desplaza fuera del registro de bit en el bit .UL. El origen especificado, el bit 06 de la palabra de entrada 23, se desplaza a la posición de bit más alta 69.

Después de que el procesador complete la operación de desplazamiento durante un escán de programa y cuando el renglón se hace falso, la instrucción restablece los bits .EN, .ER (si están establecidos) y .DN y restablece el puntero.

Para la operación de dar la vuelta, asegúrese de que la dirección de origen sea la misma que la dirección de bit inferior (salida). No es necesario usar el bit .UL en la operación de dar la vuelta.

Cómo usar las instrucciones FIFO y LIFO

Descripción:



Use las instrucciones FIFO (primero en entrar/primero en salir [FFL y FFU]) y las instrucciones LIFO (último en entrar/primero en salir [LFL y LFU]) en parejas para almacenar y recuperar datos en un orden determinado.

Estas instrucciones: Recuperan datos:

FFL y FFU	En el orden en que se almacenaron (primero en entrar/primero en salir)
LFL y LFU *	En orden inverso al orden en que se almacenaron (último en entrar/primero en salir)

* Disponible solamente en los procesadores PLC-5 con características mejoradas.

Estas instrucciones establecen un registro de desplazamiento asíncrono (pila) cuando se usan en parejas.

Cómo introducir los parámetros

Cuando programe una pila FIFO o LIFO, use las mismas direcciones de archivo y control, y los mismos valores de longitud y posición para las dos instrucciones en la pareja. Es necesario proporcionar al procesador la información siguiente:

- **El origen** es la dirección que almacena el próximo valor a entrar en la pila. Las instrucciones de carga FIFO y LIFO (FFL y LFL) recuperan el valor desde esta dirección y lo cargan en la próxima palabra en la pila.
- **El destino** es la dirección que almacena el valor que sale de la pila.

Esta instrucción: Descarga el valor desde:

FFU de FIFO	la palabra cero
LFU de LIFO	la última palabra introducida

- **El FIFO o LIFO** es una dirección indexada de la pila. Use la misma dirección FIFO para las instrucciones FFL y FFU asociadas. Use la misma dirección LIFO para las instrucciones LFL y LFU asociadas.
- **El control** es la dirección de la estructura de control (48 bits – tres palabras de 16 bits) en la zona de control (R) de la memoria. La estructura de control almacena los bits de estado, la longitud de pila y la próxima posición disponible (puntero) de la instrucción en la pila.

Use la dirección de control con un mnemónico cuando direcciona los parámetros siguientes:

- **La longitud (.LEN)** es el número máximo de elementos en la pila
- **La posición (.POS)** indica la próxima ubicación disponible donde la instrucción carga los datos en la pila
- **La longitud** especifica el número máximo de palabras en la pila. Direcciona el valor de longitud mediante el mnemónico .LEN.
- **La posición** indica la próxima ubicación disponible donde la instrucción carga los datos en la pila. Direcciona el valor de posición mediante el mnemónico .POS.

Introduzca un valor de posición solamente si desea que la instrucción comience en un offset en el momento del encendido. De lo contrario, introduzca 0. El programa de lógica de escalera puede cambiar la posición si fuera necesario.



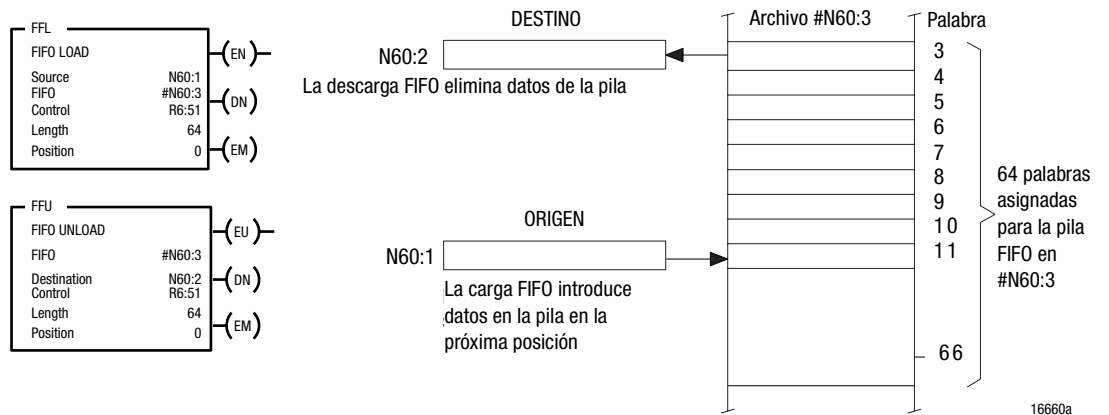
ATENCION: No use la misma dirección de control para cualquier otra instrucción a menos que empareje las instrucciones de pila. Puede resultar en una operación inesperada y causar daños al equipo y/o lesiones corporales.

Cómo usar los bits de estado

Para usar las instrucciones FIFO y LIFO correctamente, examine los bits de estado en la estructura de control. Estos bits se direccionan mnemónicamente.

Este bit:	Se establece:
Habilitación de carga .EN (bit 15)	cuando el renglón realiza una transición de falso a verdadero para indicar que la instrucción está habilitada (se usa en las instrucciones FFL y LFL). Nota: durante el preescán, este bit está establecido para evitar una carga falsa cuando se inicia el escán de programa.
Habilitación de descarga .EU (bit 14)	cuando las condiciones del renglón son verdaderas para indicar que la instrucción está habilitada (se usa en las instrucciones FFU y LFU). Nota: durante el preescán, este bit está establecido para evitar una descarga falsa cuando se inicia el escán de programa.
Efectuado .DN (bit 13)	mediante el procesador para indicar que la pila está llena. El bit .DN inhibe la carga de la pila hasta que haya espacio disponible.
Vacío .EM (bit 12)	mediante el procesador para indicar que la pila está vacía. No habilite los comandos de descarga FIFO o LIFO si el bit .EM está establecido.

Ejemplo de carga FIFO (FFL) y
descarga FIFO (FFU):



Este parámetro:	Indica el procesador:
Origen (N60:1)	La ubicación de la palabra de origen “próxima en entrar”
FIFO (#N60:3)	La ubicación de la pila (archivo FIFO)
Destino (N60:2)	La ubicación de la palabra de “salida”
Control (R6:51)	La dirección de la instrucción y la estructura de control
Longitud (64)	El número máximo de palabras que se pueden cargar
Posición (0)	Comenzar en la dirección de archivo FIFO

Descripción de la carga FIFO:

Cuando el renglón que contiene la instrucción FFL va de falso a verdadero, el procesador establece el bit .EN y carga el elemento de origen (N60:1) en el próximo elemento disponible en la pila tal como lo indica la posición de la estructura de control. El procesador carga un elemento cada vez que el renglón va de falso a verdadero hasta que llena la pila. Cuando la pila está llena, el procesador establece el bit .DN. El programa de lógica de escalera debe detectar que la pila está llena e inhibir la continuación de la carga de datos desde el origen.

Puede cargar la pila de antemano o habilitar la instrucción de carga a la vez que inhibe la instrucción de descarga hasta que la pila contenga los datos deseados.

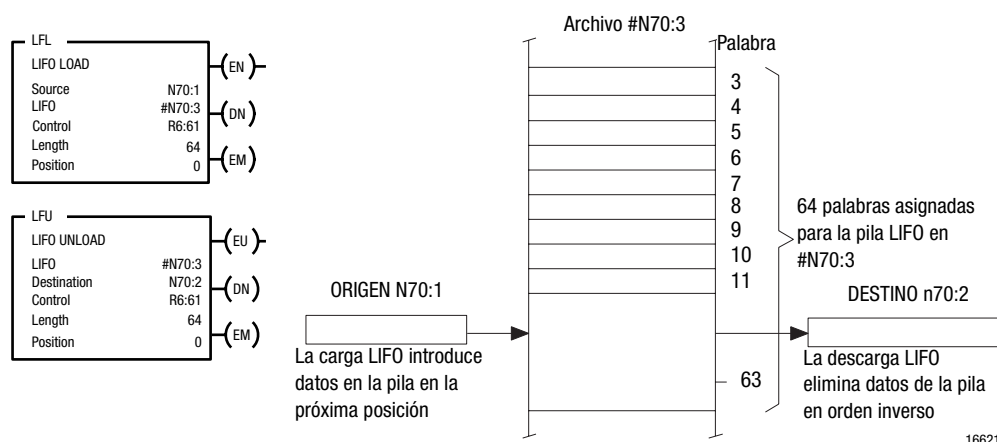
Descripción de la descarga FIFO:

Cuando el renglón que contiene la instrucción FFU va de falso a verdadero, el procesador establece el bit .EU y descarga los datos del primer elemento almacenado en la pila FIFO en la palabra de destino N60:2. Simultáneamente el procesador desplaza todos los datos en la pila una posición hacia la primera palabra. El procesador descarga una palabra cada vez que el renglón va de falso a verdadero hasta que la pila FIFO quede vacía.

Cuando la pila está vacía, el procesador establece el bit .EM. De ahí en adelante, el procesador transfiere un valor de cero para cada transición del renglón de falso a verdadero hasta que la instrucción FFL carga nuevos valores. El programa de lógica de escalera debe detectar que la pila está vacía e inhibir que otras instrucciones usen los valores de cero almacenados en el destino.

La instrucción FFU le permite descargar datos de una palabra que no sea la primera palabra de la pila si cambia la dirección FIFO a la dirección de la palabra deseada y cambiando la longitud como corresponde.

**Ejemplo de carga LIFO (LFL) y
descarga LIFO (LFU):
(Procesadores PLC-5 con
características mejoradas solamente)**



16621

Este parámetro:	Indica el procesador:
Origen (N70:1)	La ubicación de la palabra de origen “próxima en entrar”
LIFO (#N70:3)	La ubicación de la pila (archivo LIFO)
Destino (N70:2)	La ubicación de la palabra de “salida”
Control (R6:61)	La dirección de la instrucción y la estructura de control
Longitud (64)	El número máximo de palabras que se pueden cargar
Posición (0)	Comenzar en la dirección de archivo LIFO

Importante: La diferencia entre las operaciones de pila FIFO y LIFO es que la instrucción LFU elimina datos en el orden inverso en que se cargan (último en entrar/ primero en salir). Fuera de estos, las instrucciones LIFO y las instrucciones FIFO funcionan idénticamente.

Descripción de la carga LIFO:

Cuando el renglón que contiene la instrucción LFL va de falso a verdadero, el procesador establece el bit .EN y carga la palabra de origen (N70:1) en la próxima palabra disponible en la pila indicada por la posición de la estructura de control. El procesador carga un elemento cada vez que el renglón va de falso a verdadero hasta que llena la pila. Cuando la pila está llena, el procesador establece el bit .DN. El programa de lógica de escalera debe detectar que la pila está llena e inhibir la continuación de la carga de datos desde el origen.

Puede cargar la pila de antemano o habilitar la instrucción de carga a la vez que inhibe la instrucción de descarga hasta que la pila contenga los datos deseados.

Descripción de la descarga LIFO:

Cuando el renglón que contiene la instrucción LFU va de falso a verdadero, el procesador establece el bit .EU y descarga los datos a partir de la última palabra almacenada en la pila LIFO en la palabra de destino N70:2. El procesador descarga una palabra cada vez que el renglón va de falso a verdadero hasta que la pila LIFO quede vacía.

Cuando la pila está vacía, el procesador establece el bit .EM. De ahí en adelante, el procesador transfiere un valor de cero para cada transición del renglón de falso a verdadero hasta que la instrucción de carga cargue nuevos valores. El programa de lógica de escalera debe detectar que la pila está vacía e inhibir que otras instrucciones usen los valores de cero almacenados en el destino.

La instrucción de descarga LIFO le permite descargar datos de una palabra que no sea la primera palabra de la pila si cambia la dirección LIFO a la dirección de la palabra deseada y cambiando la longitud como corresponde.

Notas:

Instrucciones de secuenciador SQO, SQI, SQL

Cómo aplicar los secuenciadores

Las instrucciones de secuenciador se usan típicamente para controlar las máquinas de ensamblaje automáticas que tienen una operación uniforme y repetible. Use la instrucción de entrada de secuenciador para detectar cuando un paso se ha concluido. Use la instrucción de secuenciador de salidas para establecer las condiciones de salida para el próximo paso. Use la instrucción de carga secuenciador para cargar las condiciones de referencia en el archivo de secuenciador de entradas y salidas .

La Tabla 12.A indica las instrucciones de secuenciador disponibles.

Tabla 12.A
Instrucciones de secuenciador disponibles

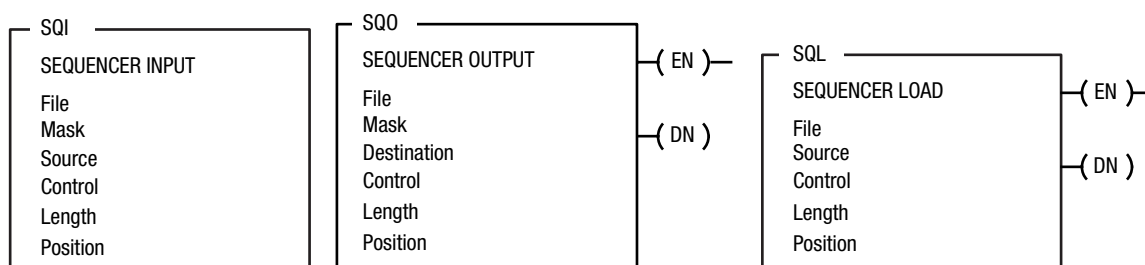
Si desea:	Use esta instrucción:	Que se encuentra en la página:
Controlar las operaciones secuenciales de máquina mediante la transferencia de datos de 16 bits a través de una máscara a direcciones de imagen de salida	SQO	12-5
Monitorear las condiciones de funcionamiento de máquinas para el propósito de diagnóstico mediante la comparación de datos de imagen de 16 bits (a través de una máscara) con datos en un archivo de referencia	SQI	12-7
Capturar condiciones de referencia activado manualmente de la máquina a través de sus secuencias de funcionamiento y la carga de datos de E/S y almacenamiento en los archivos de destino	SQL	12-8

Las instrucciones de secuenciador pueden conservar la memoria de programa. Estas instrucciones monitorean y controlan múltiplos de 16 salidas discretas simultáneamente en un solo renglón.

Vea el Apéndice C para obtener más información acerca de los operandos (y los tipos/valores de datos válidos de cada operando) usados por las instrucciones descritas en este capítulo.

Cómo usar las instrucciones de secuenciador

Descripción: Use las instrucciones SQI y SQO en parejas para monitorear y controlar una operación secuencial. Use la instrucción SQL para cargar datos en el archivo de secuenciador.



Estas instrucciones operan en múltiplos de 16 bits simultáneamente. Coloque las instrucciones SQI en serie y las instrucciones SQO en paralelo en el mismo renglón para las operaciones de 32, 48 y 64 bits u otras operaciones de bits.

Importante: Cada instrucción SQO incrementa la estructura de control. Por lo tanto, es posible que las instrucciones SQI correspondientes pasen por alto porciones del archivo de origen.

Cómo introducir los parámetros

Cuando programe las instrucciones SQI y SQO en parejas, use la misma dirección de control, valor de longitud y valor de posición en cada instrucción. Esto es también pertinente al uso de múltiples instrucciones en el mismo renglón para doblar, triplicar o aumentar más aún el número de bits.

Es necesario proporcionar al procesador la información siguiente para programar las instrucciones de secuenciador:

- **El archivo** es la dirección indexada del archivo de secuenciador al cual o desde el cual la instrucción transfiere datos. Su propósito depende de la instrucción:

En esta instrucción:	El archivo de secuenciador almacena datos para:
SQO	Controlar entradas
SQI	Referencia para detectar si un paso o una condición de fallo ha concluido
SQL	Crear el archivo SQO o SQI

- **Mask** (para SQO y SQI) es un código hexadecimal o la dirección del elemento o archivo de máscara a través del cual la instrucción mueve datos. Establezca los bits de máscara (1) para pasar datos. Restablezca los bits de máscara (0) para evitar que la instrucción opere en los bits de destino correspondientes. Especifique un valor hexadecimal para un valor de máscara constante. Almacene la máscara en un elemento o archivo si desea cambiar la máscara según los requisitos de la aplicación.
- **Source** (para SQI y SQL) es la dirección del elemento o archivo de entrada desde el cual la instrucción obtiene datos para su archivo de secuenciador.
- **Destination** (para SQO solamente) es la dirección de destino de la palabra o archivo de salida a la cual la instrucción mueve datos desde su archivo de secuenciador.

Importante: Si usa un archivo para el origen, máscara o destino de una instrucción de secuenciador, la instrucción determina automáticamente la longitud del archivo y se mueve por el archivo paso a paso a la vez que se mueve por el archivo de secuenciador.

- **El control** es la dirección de la estructura de control en la zona de (R) de la memoria (48 bits – tres palabras de 16 bits) que almacena los bits de estado de la instrucción, la longitud del archivo de secuenciador y la posición instantánea en el archivo.

Use la dirección de control con un mnemónico cuando direcciona los parámetros siguientes:

- **Length (.LEN)** es la longitud del archivo de secuenciador..
- **Position (.POS)** es la posición actual de la palabra en el archivo de secuenciador que el procesador está usando.

Para esta instrucción: **La estructura de control se aumenta:**

SQO y SQL	Mediante la instrucción
SQI	Externamente mediante el SQO emparejado con la misma dirección de control o por otra instrucción



ATENCION: No use la misma dirección de control para cualquier otro propósito a menos que empareje las instrucciones. La duplicación de un elemento de control puede resultar en una operación inesperada y causar daños al equipo y/o lesiones corporales.

- **Length** es el número de pasos del archivo de secuenciador a partir de la posición 1. La posición 0 es la posición de encendido. La instrucción se restablece a la posición 1 cada vez que concluye.

Importante: La dirección asignada para un archivo de secuenciador es paso cero. Las instrucciones de secuenciador usan (longitud + 1) palabras de datos para cada archivo referido en la instrucción. Esto también es pertinente a los valores de origen, máscara y destino si se direccionan como archivos.

- **Position** es la ubicación de la palabra en el archivo de secuenciador. El valor de posición se incrementa internamente mediante las instrucciones SQO y SQL.

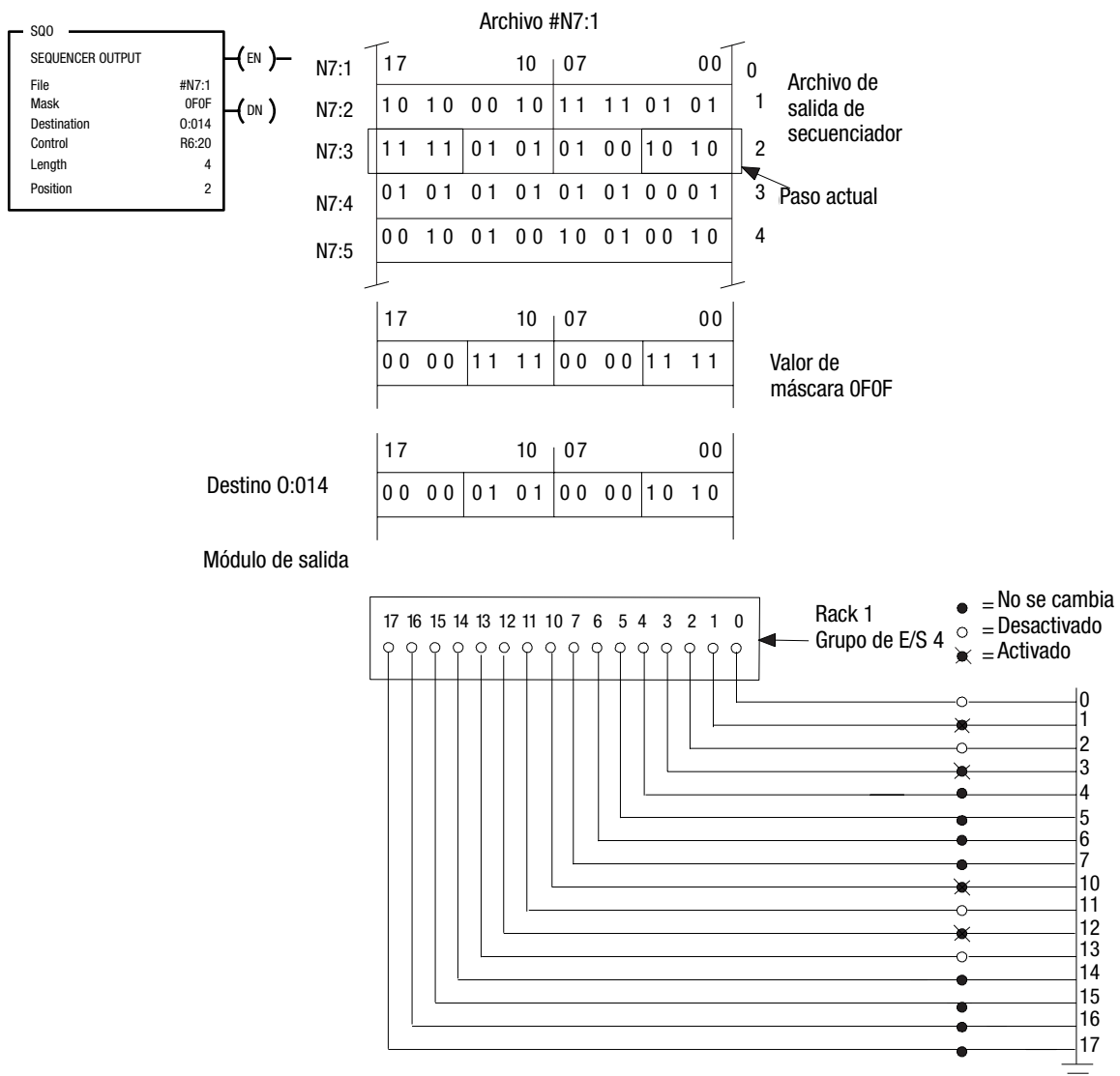
Importante: El programa de lógica de escalera puede incrementar externamente el valor de posición de la instrucción SQI. Uno de los métodos para hacer esto es empareja la instrucción SQI con la instrucción SQO y asignar la misma estructura de control a las dos instrucciones.

En los procesadores de series anteriores, si el valor .POS se encontró fuera del rango, el valor .POS se estableció automáticamente en 1, el cual es el primer paso en la secuencia. La ocurrencia de este evento no se indicó. En los procesadores de serie E y superiores, si el valor .POS excede el número de palabras en el archivo, el bit .ER se establece, no se escriben datos y el valor .POS no se cambia.

Cómo usar los bits de estado

Para usar las instrucciones de secuenciador correctamente, el programa de lógica de escalera debe examinar los bits de estado en el elemento de control. Estos bits se direccionan mediante mnemónicos.

Este bit:	Se establece:
Habilitación .EN (bit 15)	(SQO o SQL) durante una transición del renglón de falso a verdadero para indicar que la instrucción está habilitada. La instrucción sigue a la condición del renglón. Nota: durante el preescán, este bit está establecido para evitar el incremento falso del puntero de la tabla cuando se inicia el escán de programa.
Efectuado .DN (bit 13)	(SQO o SQL) después de que la instrucción termina la operación en la última palabra en el archivo de secuenciador. Después que el renglón se hace falso, el procesador restablece el bit .DIN durante la próxima transición del renglón de falso a verdadero.
Error .ER (bit 11)	cuando el valor de longitud es menor o igual a cero o cuando el valor de posición es menor que cero.

Ejemplo de Sequencer Output (SQ0):

La instrucción SQ0 mueve los datos del paso actual a través de una máscara a una palabra de salida para controlar múltiples salidas.

16645a

Este parámetro:	Indica al procesador:
Archivo (#N7:1)	La ubicación del archivo de secuenciador
Máscara (0F0F)	El valor hexadecimal fijo de la máscara
Destino (O:014)	La dirección de imagen de salida que se debe cambiar
Control (R6:20)	La estructura que controla la operación
Longitud (4)	El número de palabras que se deben procesar
Posición (2)	La posición actual

La instrucción SQO pasa por el archivo de secuenciador de palabras de salida de 16 bits cuyos bits han sido establecidos para controlar varios dispositivos de salida.

Cuando el renglón va de falso a verdadero, la instrucción se incrementa al paso siguiente (palabra) en el archivo de secuenciador #N7:1. Los datos en el archivo de secuenciador se transfieren a través de una máscara fija (0F0F) a la dirección de destino O:014. Los datos actuales se escriben al elemento de destino durante cada escán que el renglón permanece verdadero.

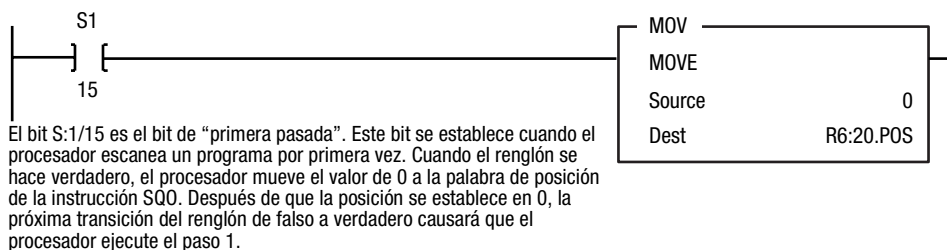
Cuando cambia el procesador del modo de programación al modo marcha en el momento del encendido, la operación de la instrucción depende de si el renglón es verdadero o falso durante el primer escán:

- Si el renglón es verdadero y POS = 0, la instrucción transfiere datos en el paso 0.
- Si el renglón es falso, la instrucción espera la primera transición del renglón de falso a verdadero y transfiere los datos en el paso 1.

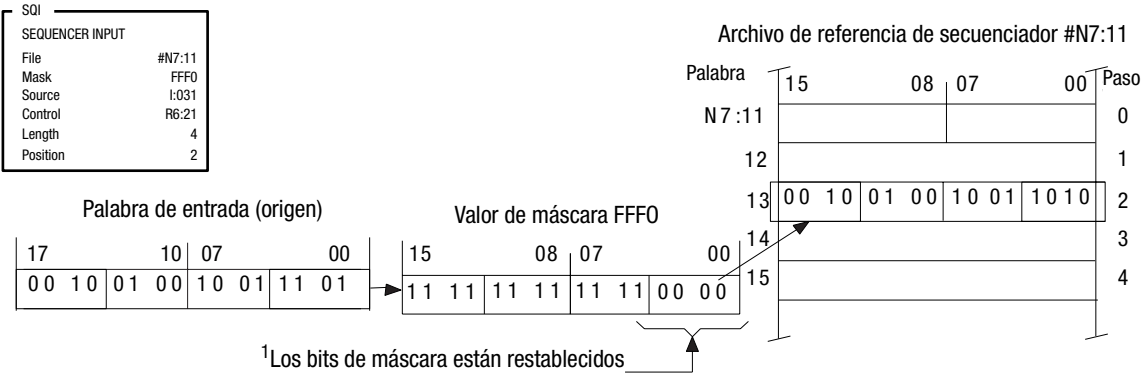
Después de transferir la última palabra del archivo de secuenciador, el procesador establece el bit .DN. Durante la próxima transición del renglón de falso a verdadero, el procesador restablece el bit .DN y establece la posición al paso 1.

Cómo restablecer la posición de SQO

Cada vez que el procesador va del modo de programación al modo marcha, usted debe restablecer la posición de cualquier instrucción SQO. Para hacerlo, use la lógica de escalera siguiente:



Ejemplo de Sequencer Input (SQI):



La instrucción SQI es verdadera cuando detecta que un palabra de entrada coincide (a través de una máscara) con su palabra de referencia correspondiente.

¹ Estos bits no se comparan. Por lo tanto, la instrucción es verdadera en este ejemplo.

16646a

Este parámetro:	Indica al procesador:
Archivo (#N7:11)	La ubicación del archivo de referencia
Máscara (FFF0)	El valor hexadecimal fijo de la máscara
Origen (#I:031)	La dirección de la imagen de entrada que se debe comparar
Control (R6:21)	El elemento que controla la operación
Longitud (4)	El número de elementos que se deben efectuar paso a paso
Posición (2)	La posición actual

La instrucción SQI compara un archivo de datos de la imagen de entrada (I:031) mediante una máscara (FFF0) a un archivo de datos de referencia (N7:11) para determinar su igualdad. Cuando el estado de todos los bits sin máscara de la palabra en el paso determinado coincide con los bits de la palabra de referencia correspondiente, la instrucción se hace verdadera. De lo contrario, la instrucción es falsa.

Importante: Puede usar la instrucción SQI con la estructura de control de la instrucción SQO. Programe la instrucción SQI como la instrucción condicional en el mismo renglón con la instrucción SQO. Asigne la misma dirección de control y longitud a las dos instrucciones de manera que vayan juntos.

Cómo usar SQI sin SQO

Otra aplicación de la instrucción SQI es el diagnóstico de máquinas donde usted carga el archivo de referencia con los datos que representan la secuencia deseada de funcionamiento de la máquina. Durante la operación, si la secuencia en tiempo real de la operación no coincide con la secuencia deseada de la operación almacenada en el archivo de referencia, habilite una señal de fallo. En este caso, el programa de lógica de escalera incrementa externamente la instrucción SQI.

Este parámetro:	Indica al procesador:
Archivo (#N7:20)	La ubicación del archivo de destino
Origen (I:002)	La dirección de la imagen de entrada que se debe leer
Control (R6:22)	La estructura que controla la operación
Longitud (5)	El número de palabras que se deben efectuar paso a paso
Posición (3)	El paso actual

Cuando el renglón va de falso a verdadero, la instrucción SQL se incrementa al próximo paso en el archivo de secuenciador y carga datos en dicho archivo, un paso por cada transición de renglón. La instrucción SQL carga los datos actuales durante cada escán que el renglón permanece verdadero. No se usa una máscara.

Cuando cambia el procesador del modo de bprogramación al modo marcha en el momento del encendido, la operación de la instrucción depende de si el renglón es verdadero o falso durante el primer escán:

- Si el renglón es verdadero, la instrucción carga datos en el paso 0.
- Si el renglón es falso, la instrucción espera la primera transición del renglón de falso a verdadero y carga los datos en el paso 1.

Después cargar el último paso, el procesador establece el bit .DN. Durante la próxima transición del renglón de falso a verdadero, el procesador restablece el bit .DN, restablece la posición al paso 1 y carga los datos en esa palabra.

Notas:

Instrucciones de control de programas MCR, JMP, LBL, FOR, NXT, BRK, JSR, SBR, RET, TND, AFI, ONS, OSR, OSF, SFR, EOT, UIE, UID

Cómo seleccionar las instrucciones de flujo de programas

Las instrucciones de flujo de programas cambian el flujo de la ejecución de programas de lógica de escalera. Use la Tabla 13.A para seleccionar la instrucción de control de programas o el grupo de instrucciones que satisfacen sus requisitos de programación.

Tabla 13.A
Instrucciones de control de programas disponibles

Si desea:	Use estas instrucciones:	Que se encuentra en la página:
Desactivar todas las salidas no retentivas en una sección de un programa de lógica de escalera	MCR	13-2
Saltar una sección de un programa que no necesita ejecutarse cada vez	JMP, LBL	13-3
Enlazar por un conjunto de renglones durante un número preseleccionado de veces	FOR, NXT, BRK	13-5
Saltar a un archivo de subrutina diferente, pasar datos a la subrutina, realizar una operación y devolver los resultados	JSR, SBR, RET	13-8
Marcar un fin temporal que bloquea la ejecución del programa	TND	13-13
Inhabilitar un renglón	AFI	13-13
Activar un evento de un impulso según un cambio de la condición del renglón	ONS, OSR,* OSF*	13-14 (ONS), 13-15 (OSR), 13-16 (OSF)
Restablecer un diagrama de función secuencial	SFR*	13-17
Terminar un archivo de transición	EOT	13-18
Habilitar o inhabilitar las interrupciones del usuario	UIE,* UID*	13-19 (UID), 13-20 (UIE)

*Estas instrucciones son compatibles solamente con los procesadores PLC-5 con características mejoradas.

Vea el Apéndice C para obtener más información acerca de los operandos (y los tipos/valores de datos válidos de cada operando) usados por las instrucciones descritas en este capítulo.

Master Control Reset (MCR)

Descripción:

—(MCR)—

Use las instrucciones MCR en parejas para crear zonas de programas que desactiven todas las salidas no retentivas en la zona. Todavía se escanean los renglones dentro de la zona MCR, pero el tiempo de escán se reduce debido al estado falso de las salidas no retentivas. Las salidas no retentivas se restablecen cuando su renglón se hace falso.

**Si el renglón MCR
que inicia la
zona es:**

El procesador:

verdadero	ejecuta el los renglones en la zona MCR según las condiciones de entrada individuales de cada renglón (como si la zona no existiera).
falso	restablece todas las instrucciones de salida no retentivas en la zona MCR independientemente de las condiciones de entrada individuales de cada renglón.

Las zonas MCR le permiten habilitar o inhibir segmentos del programa tales como aplicaciones de recetas.

Cuando programe las instrucciones MCR, observe lo siguiente:

- Es necesario terminar la zona con una instrucción MCR no condicional.
- No puede anidar una zona MCR con otra.
- No salte en una zona MCR. Si la zona es falsa, el saltar a la misma activa la zona.
- Si una zona MCR continúa hasta el fin del programa de lógica de escalera, no es necesario programar una instrucción MCR para terminar la zona.

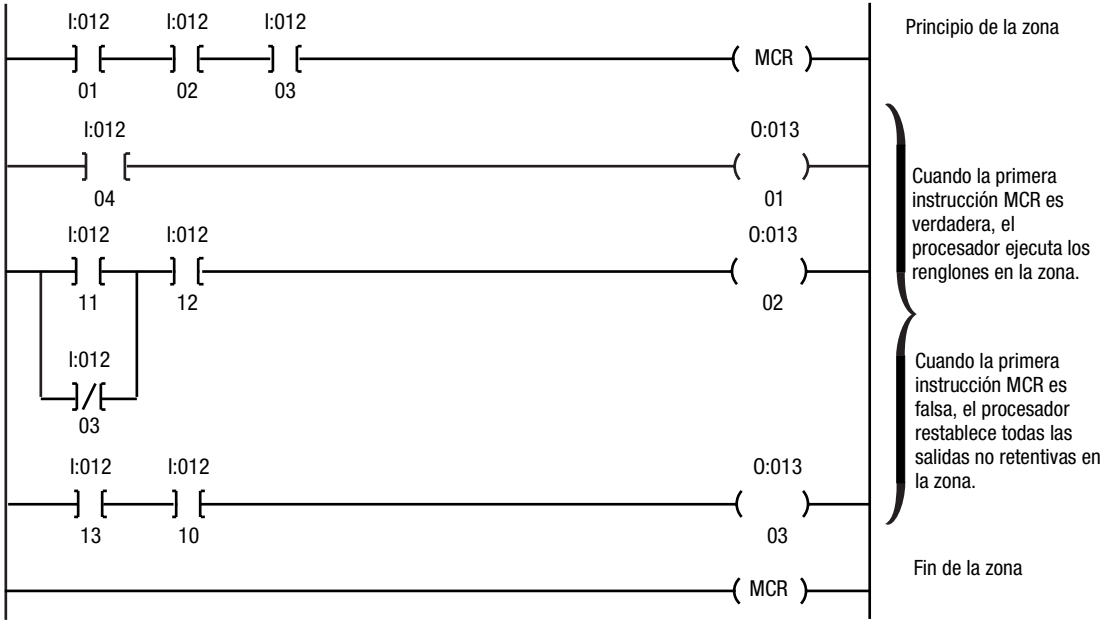
Importante: La instrucción MCR no es una sustitución para un relé de control maestro cableado que proporciona la capacidad de realizar de un paro de emergencia. Debe instalar un relé de control maestro cableado para proporcionar la desconexión de emergencia de la alimentación eléctrica de las E/S.



ATENCION: No sobreponga ni anide las zonas MCR. Cada zona MCR debe ser separada y completa. Si se sobreponen o se anidan, puede ocurrir una operación inesperada de la máquina, lo cual podría resultar en daños al equipo y/o lesiones corporales.

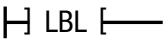
ATENCION: Si inicia instrucciones tales como los temporizadores o contadores en una zona MCR, la operación de la instrucción se detiene cuando la zona se inhabilita. Vuelva a programar las operaciones críticas fuera de la zona, si fuese necesario.

Ejemplo: Cuando el renglón que contiene la primera instrucción MCR es verdadero, el procesador ejecuta los renglones en la zona MCR según las condiciones de entrada del renglón. De lo contrario, el procesador restablece las instrucciones de salida no retentivas dentro de la zona MCR



Jump (JMP) y Label (LBL)

Descripción: Use las instrucciones JMP y LBL en parejas para saltar porciones del programa de lógica de escalera.



Si el renglón de salto es:	El procesador:
verdadero	salta del renglón JMP al renglón LBL y continúa ejecutando el programa. Se puede saltar hacia adelante o hacia atrás.
falso	ignora la instrucción JMP

El salto hacia adelante a una etiqueta ahorra el tiempo de escán de programa puesto que omite un segmento del programa hasta que se necesita. El salto hacia atrás permite que el procesador repita iteraciones a través de un segmento del programa hasta que lógica se completa.

Importante: Tenga cuidado de no saltar hacia atrás demasiadas veces. El temporizador de control (watchdog) puede sobrepasar el tiempo de espera, lo cual causa un fallo del procesador.

Cómo usar JMP

La instrucción JMP permite que el procesador salte renglones. Usted puede saltar a la misma etiqueta de una o más instrucciones JMP.



ATENCIÓN: Los temporizadores y contadores saltados no se escanean. Vuelva a programar las operaciones críticas fuera de la zona saltada.

Cómo usar LBL

La instrucción LBL es el receptor de la instrucción JMP que tiene el mismo número de etiqueta. Coloque la instrucción LBL primero en el renglón al cual desea que el procesador salte.

Importante: Asegúrese de que la instrucción LBL sea la primera instrucción en el renglón. (Actualmente el software le permite crear una bifurcación alrededor una instrucción LBL. Esto causa que el procesador funcione incorrectamente.)

Si tiene este procesador:	Números LBL válidos:	Cantidad válida por archivo de programa:
PLC-5 con características mejoradas	000-255	256
PLC-5 clásico	0-31	32

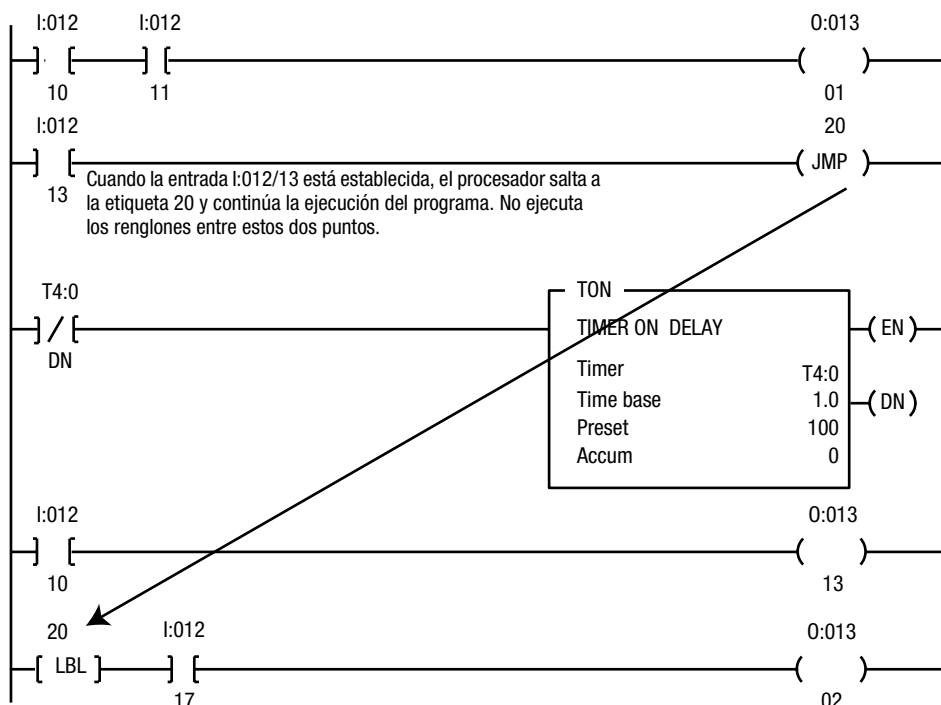
Si modifica y acepta un renglón que contiene una etiqueta mientras está en línea con el procesador en el modo marcha, el software crea una pareja I/R. Si modifica el renglón I antes de ensamblar las ediciones, el procesador entrará en fallo con un error de etiqueta duplicado.

Hay cuatro métodos para evitar este problema:

- Edite el renglón con el procesador en el modo de programación.
- Anule las ediciones y vuelva a editar el renglón.
- Permita que ocurra el fallo y luego borre el fallo después de ensamblar las ediciones.
- Ensamble la primera edición y modifique el renglón nuevamente para hacer el segundo cambio. Si edita en línea, es posible que el procesador ejecute el renglón con la primera edición y cause que el procesador entre en fallo o funcione incorrectamente.

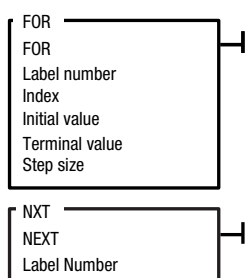
Ejemplo JMP y LBL:

Cuando el renglón que contiene la instrucción JMP se hace verdadero, el procesador salta los renglones sucesivos hasta llegar al renglón que contiene la instrucción LBL con el mismo número. El procesador reanuda la ejecución en el renglón LBL.



El temporizador (TON) no se actualizará siempre que I:012/13 sea verdadero.

For Next Loop (FOR, NXT), Break (BRK)

Descripción:

Use las instrucciones FOR, BRK y NXT para crear sus propias rutinas de programación donde usted controla el número de veces que se ejecuta el lazo.

Importante: Durante el preescán, las instrucciones de lógica de escalera dentro del lazo FOR/NXT no se preescanean (no se saltan).



ATENCION: El uso de las instrucciones FOR y NXT dentro de una bifurcación de salida puede causar una operación inesperada de la máquina.

Cuando se usan las instrucciones FOR y NXT dentro de una bifurcación de un programa de lógica de escalera, es posible que la ejecución del lazo FOR/NXT no ocurra de la manera usual. No use las instrucciones FOR ni NXT cuando programe dentro de una bifurcación en un programa de lógica de escalera.

Cómo introducir los parámetros

Es necesario proporcionar al procesador la información siguiente para programar la instrucción FOR:

Parámetro:	Definición:
Número de etiqueta	el número de etiqueta único que marca la ubicación de la instrucción FOR. Introduzca un número único. Los procesadores PLC-5 clásicos son compatibles con los números de etiqueta 0-31. Los procesadores PLC-5 con características mejoradas son compatibles con los números de etiqueta 0-255.
Índice	la dirección lógica en que la instrucción almacena el valor del índice que calcula. El valor del índice es la suma de valor inicial más los valores de paso acumulados. La instrucción FOR usa el valor del índice para determinar el número de veces que se ejecuta el lazo. Cuando habilita la instrucción FOR, el procesador establece el valor del índice igual al valor inicial. Luego, si el valor del índice es menor o igual al valor del terminal, el procesador entrega las instrucciones que siguen. Si el índice es mayor que el valor del terminal, el procesador salta a la instrucción NXT. Cuando el procesador encuentra una instrucción NXT, retorna a la instrucción FOR correspondiente y compara el índice con el valor del terminal. Si el índice es menor o igual al valor del terminal, el procesador salta nuevamente a la instrucción FOR. De lo contrario, pasa a la instrucción siguiente. Si el procesador encuentra una instrucción BRK en un renglón verdadero, salta a la instrucción que sigue a la instrucción NXT.
Valor inicial	(valor del índice) es un valor entero o dirección de número entero que representa el valor de inicio del lazo.
Valor de terminal	(valor de referencia) es un valor entero o dirección de número entero que representa el valor de fin del lazo.
Tamaño de paso	(constante) es un valor entero que especifica la cantidad según la cual el valor del índice se debe incrementar. Puede cambiar el valor del paso desde el programa de lógica de escalera.

Cómo usar FOR

Cuando el renglón es verdadero, la instrucción FOR ejecuta los renglones entre FOR y NXT repetidamente durante un escán de programa hasta que alcanza el número preseleccionado de lazos o hasta que una instrucción cancela la operación. La instrucción FOR repite esta operación durante cada escán que el renglón es verdadero. La instrucción FOR no requiere una transición para iniciar una operación.

Cuando el renglón es falso, el procesador salta al renglón que sigue a la instrucción NXT.

Importante: Tenga cuidado de no enlazar demasiadas veces durante un solo escán de programa. Una cantidad excesiva de llamadas causa que el temORIZADOR de control (watchdog) sobrepase el tiempo de espera, lo cual hace que el procesador entre en fallo.

Puede cambiar los valores iniciales y terminales desde el programa principal antes de ejecutar la instrucción FOR. No debe cambiar el valor del índice.



ATENCION: El cambiar el valor del índice puede causar que la instrucción ejecute el lazo el número de veces inesperado, lo cual puede resultar en daños al equipo y/o lesiones corporales.

Además, si edita una instrucción FOR/NXT en el modo marcha remota, asegúrese de hacer los cambios correspondientes en los dos renglones antes de ensamblar las ediciones. Por ejemplo, si desea cambiar el número de etiqueta para la pareja FOR/NXT, cambie la etiqueta en la instrucción FOR y en la instrucción NXT. Luego, ensamble las ediciones. Si ensambla las ediciones después de cambiar solamente una de las instrucciones de la pareja FOR/NXT, el procesador causa un error de tiempo de ejecución o sobrepasa el tiempo de espera del temporizador de control (watchdog).

Cómo usar BRK

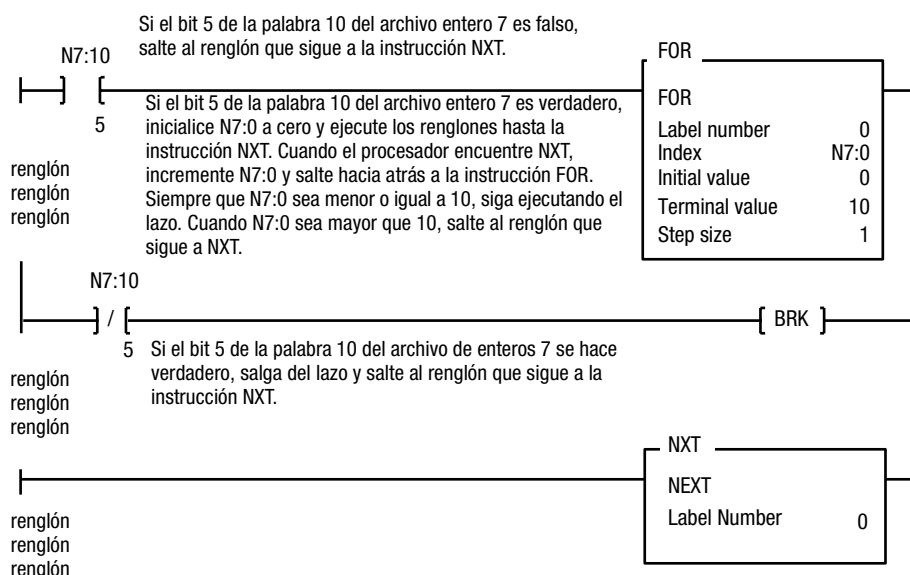
La instrucción BRK bloquea la operación de la instrucción FOR. Coloque el renglón BRK en cualquier lugar entre los renglones FOR y NXT. Cuando el renglón se hace verdadero, devuelve el procesador al lazo más alto siguiente (si se usan lazos anidados) o a la instrucción que sigue a la instrucción NXT correspondiente en el programa principal.

Use BRK para salir del lazo cuando el procesador detecte un error o para evitar lazos prolongados que pueden causar que el temporizador de control (watchdog) sobrepase el tiempo de espera, lo cual ocasionaría un fallo del procesador..

Cómo usar NXT

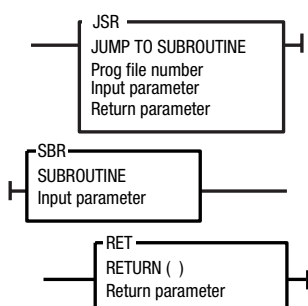
La instrucción NXT se debe programar en un renglón no condicional que es el último renglón repetido por el lazo For-Next. La instrucción NXT devuelve el procesador a la instrucción FOR correspondiente (identificada por el número de etiqueta especificada en la instrucción NXT).

Ejemplo de FOR, BRK y NXT:



Jump to Subroutine (JSR), Subroutine (SBR) y Return (RET)

Descripción:



Las instrucciones JSR, SBR y RET indican al procesador que vaya a un archivo de subrutina diferente dentro del programa de lógica de escalera que se escanee dicho archivo de subrutina una sola vez y que retorne al punto original.

La instrucción JSR dirige el procesador al archivo de subrutina especificado y, si fuera necesario, define los parámetros pasados hacia y recibidos desde la subrutina. La instrucción SBR opcional es la instrucción de encabezado que almacena los parámetros de entrada. Use SBR solamente si desea pasar parámetros. La instrucción RET termina la subrutina y, si fuera necesario, almacena los parámetros que deben retornar a la instrucción JSR en el programa principal.

Importante: Si usa la instrucción SBR, la instrucción SBR debe ser la primera instrucción en el primer renglón en el archivo de programa que contiene la subrutina.

Use una subrutina para almacenar secciones reiterativas de lógica de programa a las cuales se puede obtener acceso desde múltiples archivos de programa. Una subrutina ahorra memoria puesto que se programa una sola vez.

Actualice las E/S críticas dentro de las subrutinas mediante instrucciones de entrada/salida inmediatas (IIN, IOT), especialmente si la aplicación requiere subrutinas anidadas o bastante largas. De lo contrario, el procesador no actualiza las E/S hasta que llega al fin del programa principal (después de ejecutar todas las subrutinas). Las salidas en las subrutinas permanecen en su último estado.

Cómo pasar parámetros

Pase los valores seleccionados a una subrutina antes de la ejecución para que la subrutina pueda realizar operaciones matemáticas o lógicas en los datos y devolver los resultados al programa principal.

Por ejemplo, puede escribir una subrutina genérica para múltiples operaciones de recetas. Pase de antemano los valores preseleccionados para cada receta a la subrutina o haga que el programa principal especifique y pase los valores preseleccionados según los requisitos de la aplicación.

Puede pasar los siguientes tipos de parámetros:

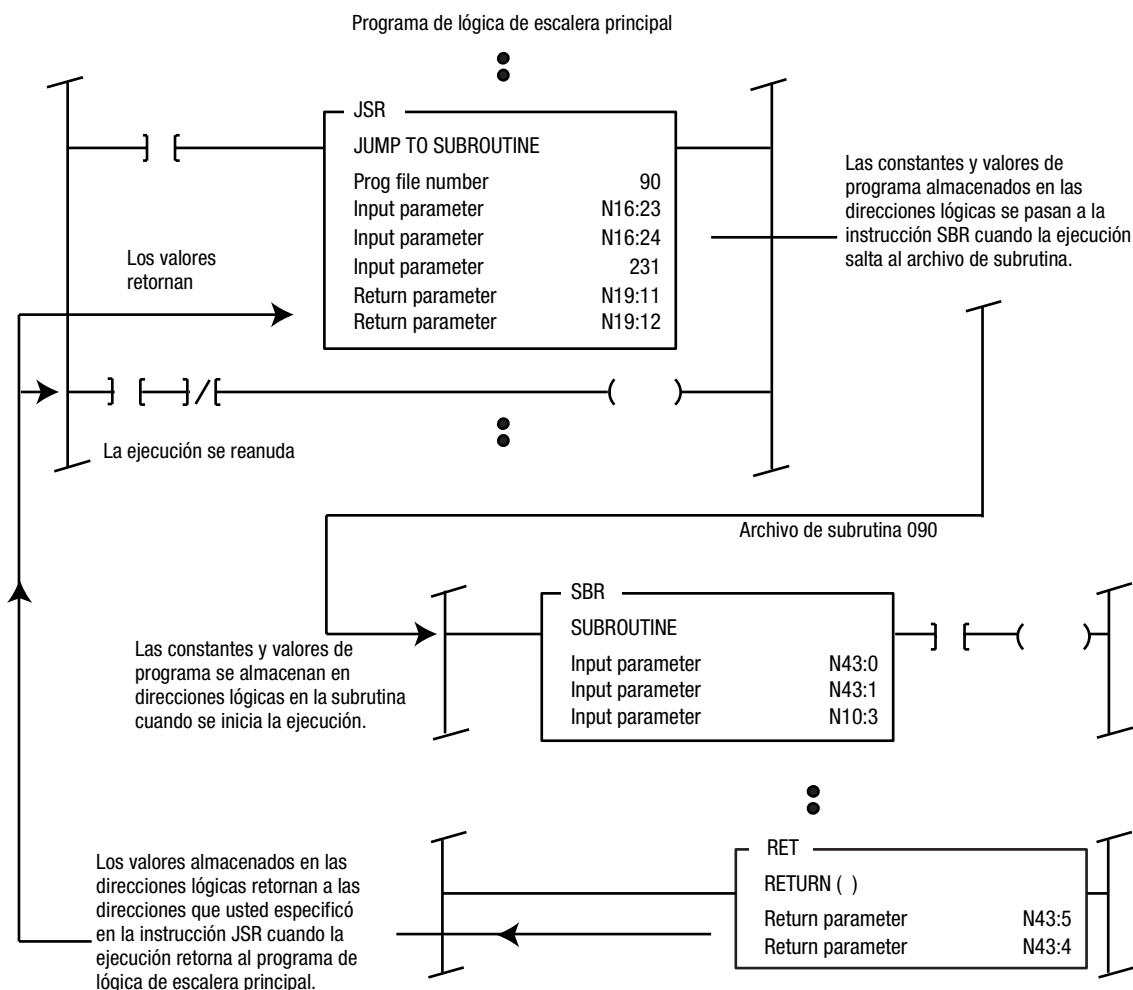
Tipo:	Ejemplo:
Constante de programa (número entero)	256
Constante de programa (punto flotante [coma flotante])	23.467
Dirección de elemento lógica	N7:0
Dirección de estructura lógica	C5:0.ACC

Si pasa los datos de punto flotante (coma flotante) a una dirección de número entero, la porción fraccionaria del valor se trunca (se pierde).

Importante: No combine los datos y las direcciones de punto flotante (coma flotante) y números enteros cuando pase datos porque esto disminuye la precisión.

Ejemplo de cómo pasar parámetros:

El diagrama siguiente muestra cómo pasar los parámetros entre un archivo de programa principal y un archivo de subrutina.



Cómo introducir los parámetros

Es necesario proporcionar al procesador la información siguiente para programar estas instrucciones:

Parámetro:	Definición:
Número de archivo del programa	el número de archivo del programa del archivo que contiene la subrutina
Parámetro de entrada (JSR)	una constante de programa o una dirección de un parámetro que se envía a la subrutina (opcional)
Parámetro de entrada (SBR)	una dirección en que la subrutina almacena los datos de entrada (opcional)
Parámetro de retorno (JSR)	una dirección que almacena los datos recibidos de la subrutina (opcional)
Parámetro de retorno (RET)	una constante de programa o una dirección de un parámetro que va a retornar a la instrucción JSR en el programa principal (opcional)

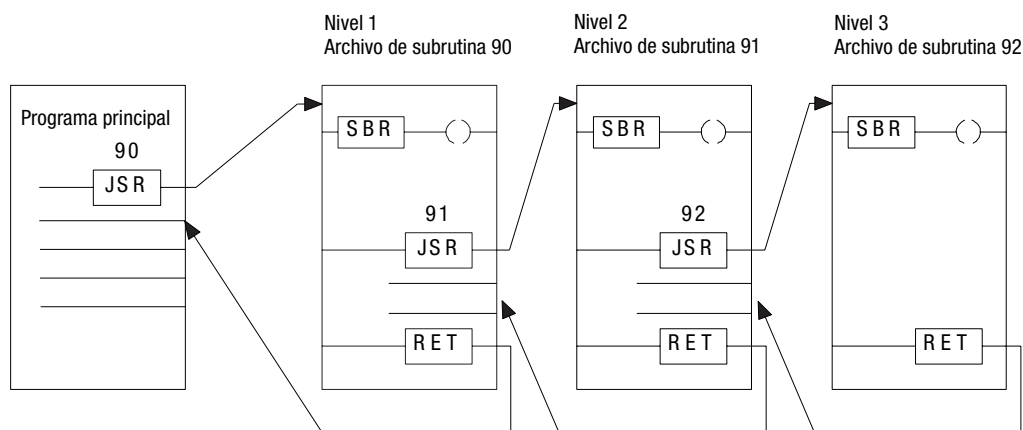
Al introducir los parámetros de entrada y retorno:

- Cuando introduce la instrucción JSR, el software de programación le invita a introducir los parámetros de entrada. Después de introducir un parámetro de entrada, presione **[Enter]**. El software le invita nuevamente a introducir otro parámetro de entrada. Cuando no tenga más parámetros de entrada que introducir, vuelva a presionar **[Enter]**. El software de programación le invita a introducir parámetros de retorno según el método usado para los parámetros de entrada. No puede introducir más de ocho parámetros de entrada y retorno combinados.
- Haga que el número de entradas JSR en la subrutina sea mayor o igual al número de direcciones de parámetros de entrada en la instrucción SBR. Si el número de entradas es menor que las direcciones que las reciben, se causa un error de tiempo de ejecución.
- Asegúrese de que el número de parámetros de retorno RET sea mayor o igual al número de direcciones de retorno JSR que los reciben. Si el número de salidas es menor que las direcciones que las reciben, se causa un error de tiempo de ejecución.

Cómo anidar los archivos de subrutina

Puede anidar hasta ocho subrutinas dentro de un archivo de programa. Esto significa que se puede dirigir el flujo de programas del programa principal a una subrutina y luego a otra subrutina, siempre que no haya más de 7 niveles de subrutinas.

La ruta de acceso de retorno es la inversa. Durante RET, el procesador retorna automáticamente a la próxima instrucción después de la instrucción JSR anterior. El procesador sigue este procedimiento hasta retornar al programa principal.



15294

Cómo usar JSR

La instrucción JSR dirige el procesador al archivo de subrutina especificado y, si fuera necesario, define los parámetros pasados hacia y recibidos desde la subrutina.

Tome en cuenta lo siguiente cuando programe la instrucción JSR:

- Cada subrutina fuera del archivo de programa principal debe tener su propio archivo identificado por una descripción de archivo única.
- No puede saltar en ninguna parte del archivo de subrutina excepto la primera instrucción (SBR) en dicho archivo.
- Puede anidar hasta ocho archivos de subrutina.

Cómo usar SBR

La instrucción SBR opcional es la instrucción de encabezado que almacena los parámetros de entrada. Use la instrucción SBR solamente si desea pasar parámetros. Cuando pasa parámetros, la instrucción SBR debe ser la primera instrucción en el primer renglón de la subrutina. Este renglón también debe tener una instrucción de salida. La instrucción SBR almacena las constantes de programa y los valores de la tabla de datos pasados desde la instrucción JSR.

Importante: Si usa la instrucción SBR, ésta debe ser la primera instrucción en el primer renglón en el archivo de programa que contiene la subrutina.

Cómo usar RET

La instrucción RET termina la subrutina y, si fuese necesario, almacena los parámetros que retornan a la instrucción JSR en el programa principal. La instrucción RET pone fin a la ejecución de la subrutina. La instrucción RET dirige el procesador nuevamente a la instrucción que sigue a la instrucción JSR correspondiente. La instrucción RET también devuelve datos a la subrutina o programa principal anteriores.

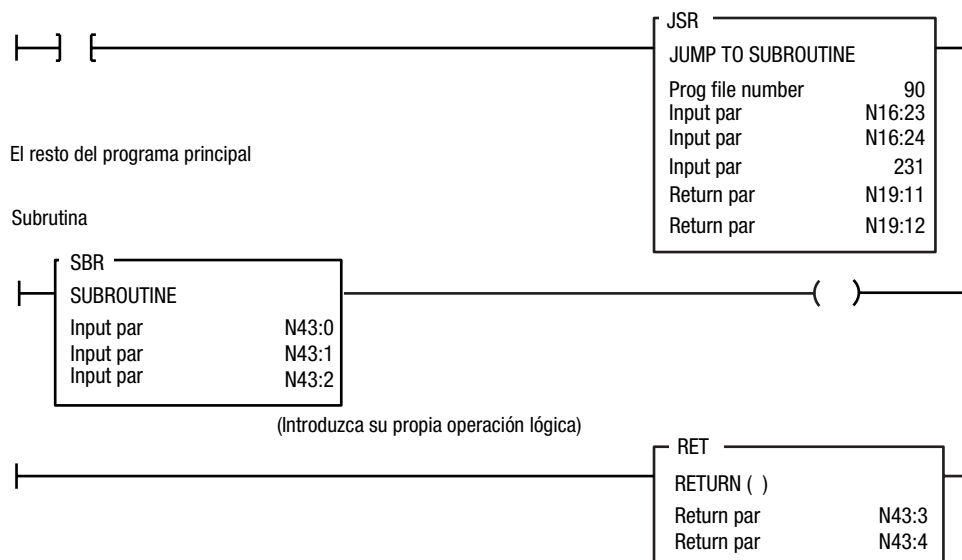
Cada subrutina debe contener una instrucción RET ejecutable si se desea devolver valores desde la subrutina. El renglón que contiene la instrucción RET puede ser condicional. Si usa este método, puede programar el procesador para que ejecute solamente una porción de la subrutina si condiciones determinadas son verdaderas. No obstante, asegúrese de programar otra instrucción RET en un renglón no condicional al fin de la subrutina para garantizar un retorno válido desde la subrutina cuando las condiciones en la primera instrucción RET sean falsas.

Importante: Para evitar el fallo del procesador, sólo use la instrucción RET en el programa cuando retorne parámetros. Si no retorna parámetros, permita que el comando final en la subrutina haga el retorno al programa principal.

Ejemplo de JSR, SBR y RET:

Cuando el renglón que contiene la instrucción JSR se hace verdadera, el procesador salta al archivo de subrutina especificado por la instrucción JSR. El procesador también pasa tres valores a la subrutina (el valor almacenado en N16:23, el valor almacenado en N16:24 y la constante 231). Luego el procesador ejecuta la lógica de subrutina.

Cuando el procesador ejecuta la instrucción RET en la subrutina, el procesador retorna a la instrucción que sigue a la instrucción JSR anterior en el programa principal. La subrutina devuelve dos valores al programa principal: el valor almacenado en N43:3 se transfiere a N19:11 y el valor almacenado en N43:4 se transfiere a N19:12.

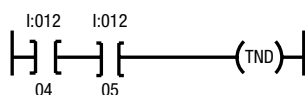


Temporary End (TND)

Descripción:

—(TND)—

Ejemplo:



Cuando el procesador encuentra la instrucción TND, el procesador restablece el temporizador de control (watchdog) (a cero), realiza una actualización de E/S y comienza a ejecutar el programa de lógica de escalera en la primera instrucción en el programa principal.

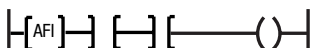
Inserte la instrucción TND cuando depure o resuelva problemas del programa de lógica de escalera. La instrucción TND permite que el programa se ejecute hasta esta instrucción solamente. Muévela progresivamente a medida que depura cada sección nueva. Use la instrucción TND también como barrera entre el programa principal y las subrutinas locales. Puede programar la instrucción TND no condicionalmente o condicionar el renglón según sus necesidades de depuración.

Importante: No confunda la instrucción TND con el símbolo de fin de programa (EOP). No se puede colocar instrucciones en el renglón que tiene el símbolo EOP.

Always False (AFI)

Descripción:

Ejemplo:



La instrucción AFI es una instrucción de entrada que hace que el renglón se haga falso cuando se inserta el lado en la condición del renglón. Puede usar la instrucción AFI para inhabilitar temporalmente un renglón cuando depura un nprograma.

One Shot (ONS)

Descripción: La instrucción ONS es una instrucción de entrada que hace que el renglón sea verdadero durante un escán de programa cuando ocurre una transición de falso a verdadero de las condiciones antes de la instrucción ONS en el renglón.

┌ ONS ─

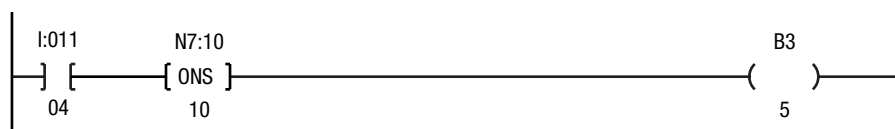
Use la instrucción ONS para comenzar eventos que se activan mediante un botón pulsador, tal como la extracción de valores de interruptores de regulación manual o la inmovilización rápida de los valores LED mostrados en pantalla. Debe introducir una dirección de bit para el bit. Use un archivo binario o una dirección de archivo entero. Un bit único se debe dedicar a cada ONS. Puede programar una dirección de salida para la instrucción ONS, pero tome en cuenta lo siguiente:



ATENCION: La programación en línea con esta instrucción puede ser peligrosa puesto que la salida puede activarse inmediatamente cuando el renglón se escanea. Establezca el valor de la dirección de bit a 1 antes de introducir la instrucción. El renglón debe ir de falso a verdadero antes de activar su salida.

Importante: Durante el preescán, la dirección de bit está establecida para inhibir la activación falsa cuando el escán de programa comienza.

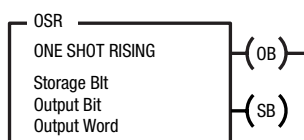
Ejemplo:



Cuando la condición de entrada va de falso a verdadero, la instrucción ONS condiciona el renglón para que la salida se active durante un escán. La salida se desactiva durante escanes sucesivos hasta que la entrada va nuevamente de falso a verdadero.

One Shot Rising (OSR) (procesadores PLC-5 con características mejoradas solamente)

Descripción:



La instrucción OSR es una instrucción de salida que activa la ocurrencia de un evento una sola vez. La instrucción OSR establece los bits siguientes:

Este bit:	Cambia el estado de la manera siguiente:
Salida .OB	Está establecido durante un escán de programa cuando el renglón va de falso a verdadero Nota: durante el preescán, este bit se restablece para inhibir la activación falsa cuando el escán de programa comienza.
Almacena - miento .SB	Sigue el estado del renglón Nota: durante el preescán, este bit se restablece para inhibir la activación falsa cuando el escán de programa comienza.

Use la instrucción OSR cuando sea necesario que un evento se inicie según el cambio de estado del renglón de falso a verdadero y no continuamente cuando el renglón es verdadero. Debe introducir una dirección de bit para el bit de salida y el bit de almacenamiento. Use una dirección de archivo binario o de número entero.

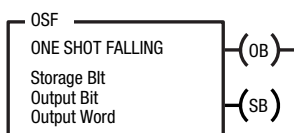
Cómo introducir los parámetros

Es necesario proporcionar al procesador la información siguiente para programar estas instrucciones:

Parámetro:	Definición:
Bit de almacenamiento	la dirección donde desea almacenar el estado del bit de almacenamiento. Por ejemplo, B3/17
Bit de salida	la posición de bit en la palabra de salida donde desea almacenar el estado del bit de salida. Por ejemplo, 5
Palabra de salida	la dirección de palabra en que desea almacenar el estado del bit de salida. Por ejemplo, N7:0

One Shot Falling (OSF) (procesadores PLC-5 con características mejoradas solamente)

Descripción:



La instrucción OSF es una instrucción de salida que activa la ocurrencia de un evento una sola vez durante la transición del renglón de falso a verdadero. La instrucción OSF establece los bits siguientes:

Este bit:	Cambia el estado de la manera siguiente:
Salida .OB	se establece durante un escán de programa cuando el renglón va de verdadero a falso
Almacena- miento .SB	Sigue el estado del renglón

Use la instrucción OSF cuando sea necesario que un evento se inicie según el cambio de estado del renglón de falso a verdadero y no según el estado del renglón resultante. Debe introducir una dirección de bit para el bit de salida y el bit de almacenamiento. Use una dirección de archivo binario o de número entero.

Cómo introducir los parámetros

Es necesario proporcionar al procesador la información siguiente para programar estas instrucciones:

Parámetro:	Definición:
Bit de almacena- miento	la dirección donde desea almacenar el estado del bit de almacenamiento. Por ejemplo, B3/17
Bit de salida	la posición de bit en la palabra de salida donde desea almacenar el estado del bit de salida. Por ejemplo, 5
Palabra de salida	la dirección de palabra en que desea almacenar el estado del bit de salida. Por ejemplo, N7:0

Sequential Function Chart Reset (SFR) (procesadores PLC-5 con características mejoradas solamente)

Descripción:

SFR	
SFC Reset	
Prog file number	
Restart step at	

La instrucción SFR restablece la lógica en un diagrama de función secuencial. Cuando una instrucción SFR se hace verdadera, el procesador realiza un post-escán/último escán en todos los pasos activos y acciones en el archivo seleccionado. Luego, restablece la lógica en la instrucción SFC durante el próximo escán de programa. El diagrama permanece en este estado restablecido hasta que la instrucción se hace falsa. La instrucción SFR también restablece todas las acciones retentivas que están activas actualmente.

Cómo introducir los parámetros

Ejemplo:

SFR	
SFC Reset	
Prog file number	2
Restart step at	N7:5

Es necesario proporcionar al procesador la información siguiente para programar esta instrucción:

Parámetro:	Definición:
Número del archivo de programa	un número válido del archivo de programa de SFC
Reiniciar el paso en	introduzca uno de los siguientes: • un número válido de referencia de paso, 0 a 32767 (la introducción de 0 causa el reinicio predeterminado en el paso inicial) • un nombre de paso válido • una dirección de número entero (que almacena un número de referencia de paso) • un símbolo de dirección (de una dirección de número entero que almacena un número de referencia de paso)

Importante: El parámetro de reinicio del paso está disponibles solamente en los procesadores PLC-5/11, -5/20 y -5/30 de serie A, PLC-5/40, -5/40L, -5/60 y 5/60L de serie B y todos los procesadores PLC-5 con características mejoradas de serie C. Si usa un procesador PLC-5/40 ó -5/60 de serie A, la instrucción SFC restablece el paso inicial.

Un número de paso es un número de referencia asignado por el software asociado con cada paso. Debe configurar la instrucción para que muestre en pantalla estos números. Vea el manual de programación para obtener información acerca de cómo configurar la pantalla.

Un nombre de paso es cualquier nombre que se asigna al paso. Vea la sección acerca de cómo asignar nombres de paso y transición en el manual de programación para obtener más información.

Importante: Asegúrese de que el paso es un paso y no una transición o macro. Estos causan el fallo del procesador. El software no los verifica. También asegúrese de que el paso no se encuentre dentro de una bifurcación simultánea. De lo contrario, el procesador entrará en fallo.

Importante: Use solamente una instrucción SFR para un solo diagrama. Múltiples instrucciones SFR en el mismo diagrama pueden causar resultados no deseados puesto que los escanes verdaderos y falsos de la instrucción SFR causan un comportamiento diferente de los programas.

Como analogía, se presenta el uso de múltiples instrucciones del temporizador TON con el mismo archivo de control. Si desea restablecer un diagrama a otras posiciones en el diagrama en base a condiciones diferentes, cargue el “paso para restablecer a” en una ubicación de tabla de datos enteros según la condición y active la instrucción SFR.

End of Transition (EOT)

Descripción:

Ejemplo:



La instrucción EOT debe ser la última instrucción en un archivo de transición. Si no coloca una instrucción EOT en un archivo de transición, el procesador siempre evalúa el archivo de transición como verdadero.

Importante: La instrucción EOT se salta durante el preescán para que todas las instrucciones de lógica de escalera se puedan preescanear.

User Interruptor Disable (UID) (procesadores PLC-5 con características mejoradas solamente)

Descripción:

—(UID)—

La instrucción UID se usa para inhabilitar temporalmente los programas de interrupción, tales como las interrupciones temporizadas seleccionables (STI) o interrupciones de entrada del procesador (PII).

Cuando el renglón es verdadero, la instrucción UID incrementa un contador de inhabilitación de interrupción interna. Siempre que este valor de contador no sea igual a cero, una instrucción STI o PII **no puede** interrumpir el programa que se ejecuta actualmente. Además, si tiene una llamada de subrutina dentro de una pareja UIE/UID, dicha subrutina se ejecuta sin interrupción.

La instrucción UID no inhabilita la rutina de fallo del usuario.

Importante: Puesto que la instrucción UID hace que un programa no se pueda interrumpir, es posible que se afecte el tiempo de respuesta del procesador a un evento STI o PII. La sección UID/UIE del programa debe ser tan corta como sea posible. Si deja las instrucciones STI y PII inhabilitadas durante largos plazos de tiempo, pueden ocurrir errores de sobreposición de las mismas.

Importante: Si tiene una transferencia en bloques en una instrucción STI o PII y dicha transferencia en bloques se encuentra dentro de la sección UID/UIE del programa, se detiene el escán del programa principal hasta concluye que la transferencia en bloques.

User Interrupt Enable (UIE) (procesadores PLC-5 con características mejoradas solamente)

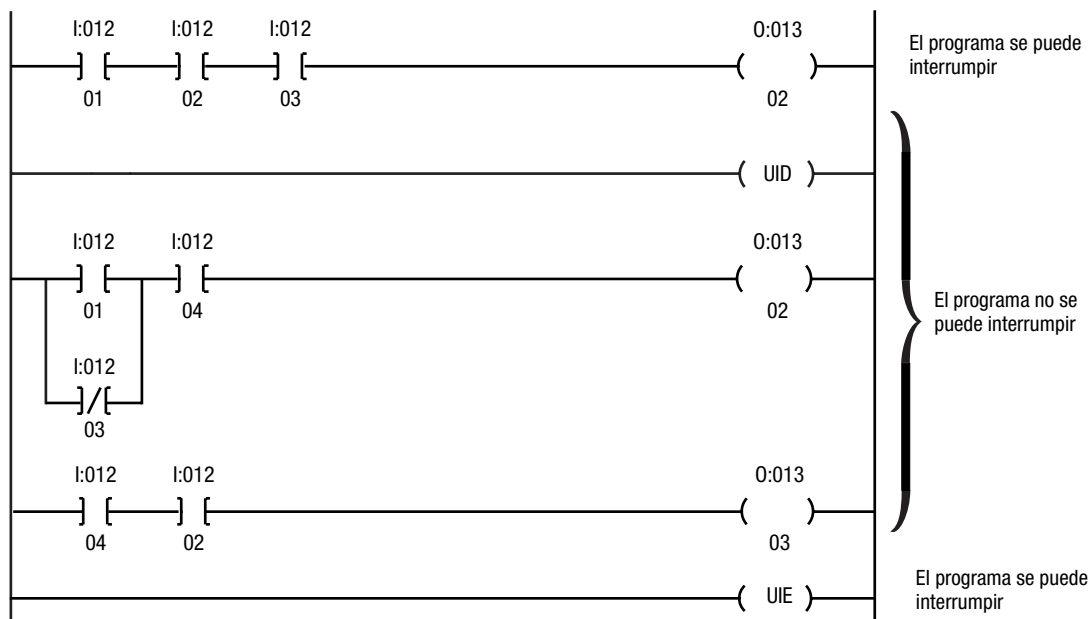
Descripción: La instrucción UIE vuelve a habilitar los programas de interrupción STI o PII.

—(UIE)—

Cuando el renglón es verdadero y el contador de habilitación de interrupción interna es mayor que cero, el contador de inhabilitación de interrupción disminuye.

Cuando el contador es igual a cero, el programa que se está ejecutando actualmente puede interrumpirse nuevamente. Si hay programas de interrupción pendientes, estos se ejecutarán ahora.

Ejemplo:

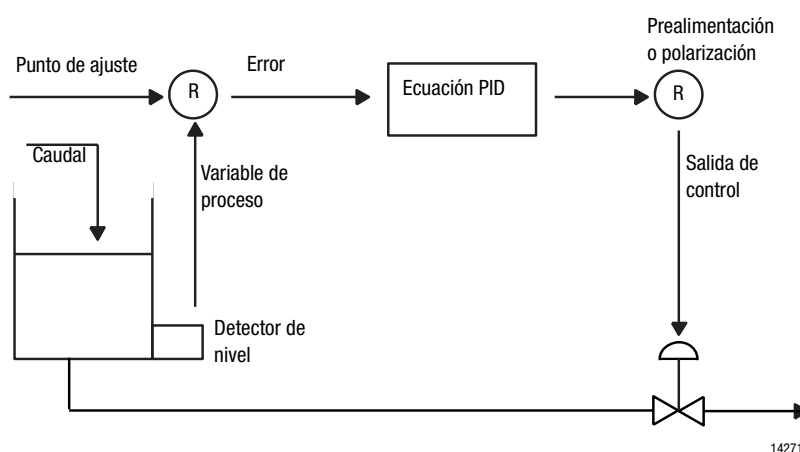


Instrucción de control de proceso PID

Cómo usar PID

El control de lazo cerrado PID mantiene una variable de proceso en un punto de ajuste deseado. La Figura 14.1 muestra un ejemplo de régimen de flujo/nivel de fluido.

Figura 14.1
Ejemplo de control PID



En el ejemplo anterior, la ecuación PID controla el proceso enviando una señal de salida a la válvula de control. Cuanto mayor es el error entre el punto de ajuste y la entrada de la variable de proceso, tanto mayor es la señal de salida, y vice versa. Se puede añadir un valor adicional (prealimentación o polarización) a la salida de control como offset. El propósito de los cálculos PID es mantener la variable de proceso que usted controla en el punto de ajuste.

Vea el fin de este capítulo para obtener información acerca de las consideraciones de programación.

Vea el Apéndice C para obtener información acerca de los operandos (y los tipos/valores de datos válidos de cada operando) usados por la instrucción PID.

Características PID

La instrucción PID permite que el proceso monitoree y controle lazo de control para cantidades tales como la presión, temperatura, régimen de flujo y nivel de fluido. Las características de la instrucción PID incluyen:

- ecuaciones PID expresadas en ISA o ganancias independientes
- rango de entrada y salida de 0-4095 (analógico de 12 bits)
- escaleado de entrada en unidades de ingeniería
- banda muerta de paso por cero
- término de derivada (puede afectar PV o error)
- control de acción directa o inversa
- alarmas de salida
- límite de salida con bloqueo de acción integral
- modo manual (con transferencia sin perturbaciones)
- prealimentación o polarización de salida
- visualización en pantalla y monitoreo de valores PID

Cómo usar las ecuaciones PID

La instrucción PID tiene dos formatos específicos: el tipo de bloque de control de número entero y el tipo de bloque de control PD. Los dos formatos usan la misma técnica de cálculo para la ecuación básica, pero tienen diferentes opciones y funciones matemáticas, específicamente, la matemática de números enteros y punto flotante (coma flotante).

La ecuación PID de base usada en los dos casos es el algoritmo PID de posición en paralelo con la opción de introducir ganancias como 'independientes' o 'dependientes'. La segunda opción es reconocida como el formato estándar ISA.

El procesador le ofrece las seis selecciones siguientes de algoritmos PID

La ecuación estándar con ganancias dependientes (estándar ISA):

Derivada de error:

$$= K_c \left[(E) + \frac{1}{T_i} \int_0^t (E) dt + T_d \frac{d(E)}{dt} \right] + Polarizacion$$

Derivada de PV:

$$CV = K_c \left[(E) + \frac{1}{T_i} \int_0^t (E) dt + T_d \frac{d(PV)}{dt} \right] + Polar(E = SP - PV)$$

$$CV = K_c \left[(E) + \frac{1}{T_i} \int_0^t (E) dt + T_d \frac{d(PV)}{dt} \right] + Polar(E = PV - SP)$$

Ecuación de ganancias independientes:

Derivada de error:

$$CV = K_p(E) + K_i \int_0^t (E) dt + K_d \frac{d(E)}{dt} + Polarizacion$$

Derivada de PV:

$$CV = K_p(E) + K_i \int_0^t (E) dt - K_d \frac{d(PV)}{dt} + Polar(E = SP - PV)$$

$$CV = K_p(E) + K_i \int_0^t (E) dt + K_d \frac{d(PV)}{dt} + Polar(E = PV - SP)$$

Donde:

K_p = Ganancia porporcional (sin unidades)	SP = Punto de ajuste
K_i = Ganancia integral (segundos ⁻¹)	PV = Variable del proceso
K_d = Ganancias derivada (segundos)	Error = (SP – PV) o (PV – SP)
$\frac{1}{T_1}$ = Ganancia de restablecimiento (repeticiones/minuto)	Polzarización = Prealimentación o polarización externa
T_d = Ganancia de velocidad (repeticiones/minuto)	CV = Variable de control de salida
	Δ_t = Tiempo de actualización del lazo

Conversión de constante de ganancia

Convierta de ganancias estándar en ganancias independientes sustituyendo los valores de de ganancia (Kc), restablecimiento (1/Ti) y velocidad del controlador en las fórmulas siguientes:

$$K_p = K_c \text{ sin unidades}$$

$$K_i = \frac{K_c}{60 T_i} \text{ segundos}$$

$$K_d = K_c (T_d) 60 \text{ segundos}$$

Implementación de término integral

Realice la integración manteniendo una suma acumulada, S_k .

En el caso de ganancias Independientes: $S_k = K_i(E_k)\Delta_t + S_{k-1}$

Con ganancias dependientes seleccionadas: $S_k = \frac{1}{T_i}(E_k)\Delta_t + S_{k-1}$

Si la ganancia integral o de restablecimiento es cero, la suma acumulada se pone a cero continuamente en el modo automático.

Puede evitar la acción integral si impide que la suma corriente se acumule cuando la salida (CV) alcance los valores máximo o mínimo. Estos valores son 0% ó 100% o los límites especificados por el usuario en el límite de salida. En este caso, es $S_k = S_{k-1}$.

La suma acumulada permanece inmovilizada hasta que la salida cae por debajo de su valor máximo o sobrepasa su valor mínimo. La acumulación normal se reanuda.

Cuando se ejecuta la instrucción PID en el modo manual, se puede realizar una transferencia sin perturbaciones nuevamente al modo automático con el uso de la suma acumulada para calcular el seguimiento de la salida manual:

$$S_k = CV_{Manual} - Polarización - K_p(E) - K_d \frac{d(E)}{dt}$$

Cuando se cambia otra vez al modo automático, el cálculo PID produce este valor de salida manual y no ocurre un salto en la salida como resultado del cambio de modo.

Término de derivada

Se usa la aproximación siguiente para calcular el término de derivada:

$$\frac{d(Q)}{dt} = \frac{Q_k - Q_{k-1}}{\Delta_t}$$

Donde Q representa un error o PV,
según los las selecciones.

El cálculo es más preciso si se usa un 'filtro de uniformidad de derivada'. Este filtro digital de paso bajo de primer orden elimina los 'picos' grandes del término de derivada causados por el ruido en la PV.

Si añade este filtro al término de derivada general, se produce lo siguiente:

$$D_k = (1 - \alpha) \left[K_d \frac{Q_k - Q_{k-1}}{\Delta_t} \right] + \alpha D_{k-1}$$

Donde:

K_d = la ganancia de derivada

D_k = el término de derivada actual

D_{k-1} = el término de derivada anterior

Q_k = (según lo definido anteriormente)

$$\alpha = \frac{1}{16 \frac{\Delta_t}{K_d} + 1}$$

Δ_t = Tiempo de actualización de lazo

Cómo establecer los rangos de entrada/salida

El módulo de entrada que mide la variable del procesador (PV) debe tener un rango binario de amplia escala de 0-4095. El procesador ignora los cuatro bits superiores más significativos de la variable de proceso de 16 bits (PID de número entero solamente).

La salida de control tiene el mismo rango de 0-4095. Usted puede establecer límites en la salida para restringir la salida calculada por la instrucción PID a cualquier valor en el rango de 0-4095.

La entrada del valor retenido (seguimiento de salida) desde una estación de control manual también debe tener un rango de 0-4095. La instrucción PID usa el resultado para calcular el valor acumulado de número entero, lo cual permite la transferencia sin perturbaciones del control manual al control automático.

La instrucción PID también copia el valor retenido a la ubicación de almacenamiento de salida de control cuando está en el modo manual. La entrada del valor retenido se usa solamente cuando se usa una estación de hardware automática/manual. De lo contrario, ponga el valor retenido a cero.

Cómo implementar el escalado a unidades de ingeniería – tipo de archivo de número entero

Puede escalar los valores de punto de ajuste y banda muerta de paso por cero a unidades de ingeniería para los tipos de archivo de número entero. También puede mostrar en pantalla la variable de proceso y los valores de error en estas misas unidades.

Cuando selecciona el escalado, la instrucción PID escala los valores de punto de ajuste, banda muerta, variable de proceso y error. También tiene que hacer lo siguiente:

1. Introduzca los valores máximo y mínimo $S_{\max}$ y $S_{\min}$ en el bloque de control PID (palabras 7 y 8). El valor $S_{\min}$ corresponde a un valor analógico de cero para la medición más baja de la variable de proceso. El valor $S_{\max}$ corresponde a un valor analógico de 4095 para la medición más alta de la variable de proceso. Estos valores representan los límites del proceso. Establezca $S_{\min}$ y $S_{\max}$ si no desea el escalado.

Por ejemplo, si mide una escala de temperatura de -73 (PV=0) a $+1156$ (PV=4095), introduzca -73 para $S_{\min}$ y 1156 para $S_{\max}$.

Si el módulo de entrada analógica no está configurado para devolver un valor en el rango de 0-4095, vea “Cómo desactivar el escalado de entradas” en la página 14-27 de este capítulo.

2. Restablezca el bit 5 de la palabra 0 en el bloque de control PID (tipo de archivo de número entero solamente). Establezca este bit solamente si desea inhibir el escalado del punto de ajuste. Debe inhibir el escalado del punto de ajuste de un lazo interno en cascada a la vez que escanea otras variables de lazo.

- Introduzca los valores de punto de ajuste, palabra 2, y banda muerta, palabra 9 (tipo de archivo de número entero solamente), en las mismas unidades de ingeniería escaladas. La salida de control (palabra 16) se muestra en pantalla como porcentaje del rango de 0-4095. La salida que el procesador transfiere al módulo de salida nunca está escalada.

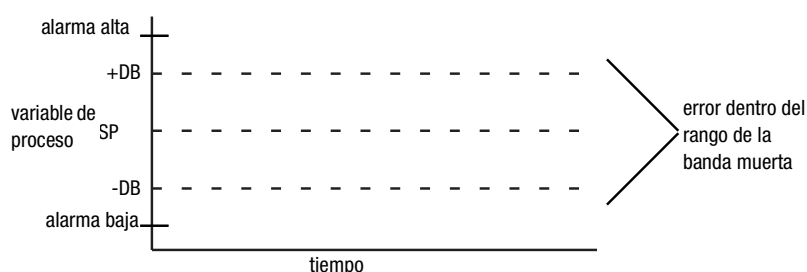


ATENCION: No cambie el escalado cuando el procesador se encuentre en el modo marcha. El procesador puede entrar en fallo y causar una respuesta de proceso no deseada, daños al equipo y lesiones corporales.

Cómo establecer la banda muerta

La banda muerta ajustable le permite seleccionar un rango de error por encima y por debajo del punto de ajuste donde la salida no cambia siempre que el error permanezca dentro de este rango.

Esta banda muerta le permite controlar la precisión de coincidencia entre la variable de proceso y el punto de ajuste sin cambiar la salida.



Cómo usar el paso por cero

El paso por cero es un control de banda muerta que permite que la instrucción use el error para propósitos de cálculo cuando la variable de proceso pasa a la banda muerta hasta que la variable de proceso pasa el punto de ajuste. Una vez que la variable de proceso pasa el punto de ajuste (el error pasa cero y cambia el signo) y siempre que la variable de proceso permanezca en la banda muerta, la instrucción considera el valor de error como cero.

Introduzca el valor de banda muerta en la palabra 9 del bloque de control (palabra .DB de un tipo de archivo de datos PD). La banda muerta se extiende por encima y por debajo del punto de ajuste según el valor especificado. Introduzca 0 para inhibir la banda muerta. Si se escala, la banda muerta tiene las mismas unidades de escala que el punto de ajuste.

Cómo usar la característica sin paso por cero

El procesador de serie E ahora tiene una característica de no paso por cero, lo cual es útil para las aplicaciones que ejecutan procesos de alta inercia que mueven lentamente masas grandes difíciles de detener. La característica sin paso por cero causa que la salida CV no cambie el valor siempre que la PV se encuentre dentro del rango y no solamente después de que la salida CV alcance el valor de punto de ajuste. Con los ajustes correctos, es posible hacer que la PV entre al valor del punto de ajuste.

Cómo seleccionar el término de derivada (acciones en PV o error)

La derivada es un cambio de la variable de estado. Se puede seleccionar si el término de derivada en cualquier ecuación PID actúa en cambios en la variable de procesador o valor de error. Use el bit 6 de la palabra 0 en el bloque de control (palabra .DO de un tipo de archivo de datos PD) para seleccionar el tipo de acción de derivada que desea.

Cómo establecer las alarmas de salida

Puede usar una alarma de salida en la salida de variable de control en un valor seleccionado por encima o por debajo del punto de ajuste. Cuando la instrucción detecta que la salida ha alcanzado cualquier valor, el procesador establece un bit de alarma (el bit 10 para el límite inferior y el bit 9 para el límite superior) en la palabra 0 del bloque de control (bits .OLH y .OLL de un tipo de archivo de datos PD). La instrucción restablece los bits de alarma cuando la salida se encuentra nuevamente dentro de los límites. La instrucción no impide que la salida exceda los valores de alarma a menos que seleccione el límite de salida.

Introduzca la alarma de salida superior en la palabra 11 (.MAXO) y la alarma de salida inferior en la palabra 12 (.MINO) del bloque de control. El procesador trata los valores de alarma de salida como porcentaje de la salida. Si no desea las alarmas, introduzca 0% para la alarma inferior y 100% para la alarma superior.

Cómo usar el límite de salida

Puede establecer un límite de salida (porcentaje de salida) en la salida de control. Cuando la instrucción detecta que la salida ha alcanzado un límite, establece un bit de alarma (el bit .OLL para el límite inferior y el bit .OLH para el límite superior) en la palabra 0 del bloque de control e impide que la salida exceda cualquiera de los valores. La instrucción restringe la salida a 0 y 4095 si no especifica un límite.

Para usar los límites de salida, establezca el bit de habilitación de límite (el bit 03 de la palabra 0) e introduzca el límite superior en la palabra 11 y el límite inferior en la palabra 12. Los valores de límite son un porcentaje (0-100%) de la salida.

Importante: Si usa el tipo de archivo de datos PD para el bloque de control, el procesador realiza esta función sin la necesidad de establecer los bits.

Bloqueo de acción integral

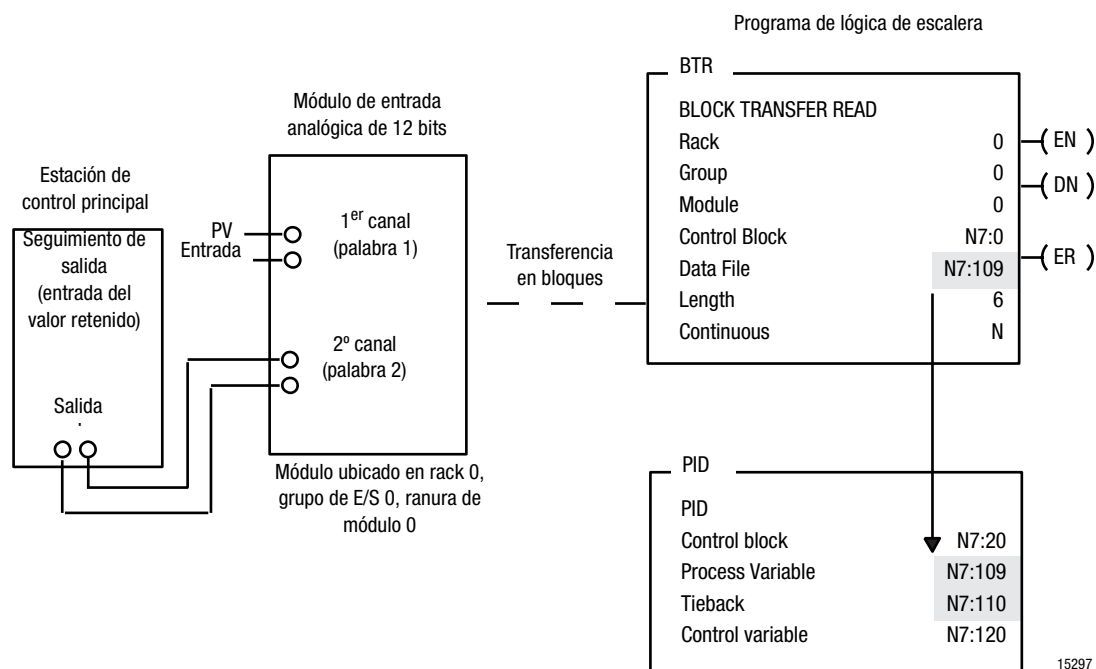
El bloqueo de acción integral es una característica que impide que el término integral se haga excesivo cuando las salidas alcancen un límite. Cuando la suma del PID y los término de polarización en la salida alcanzan un límite, la instrucción deja de calcular la salida integral hasta que la salida se encuentra nuevamente dentro del rango.

Cómo usar una operación de modo manual (con transferencia sin perturbaciones)

La operación manual permite que una salida de una estación de control manual o del programa de lógica de escalera anule la salida calculada de la instrucción PID.

Con una estación de control manual, puede controlar el dispositivo de salida directamente y anular la salida de la instrucción PID. Debe alimentar el valor de salida en la entrada de valor retenido de la instrucción PID (Figura 14.2). La instrucción PID usa este valor para calcular el valor del término integral requerido para obtener una transferencia sin perturbaciones cuando cambia del control manual al control automático.

Figura 14.2
Ejemplo de diagrama para mover las entradas analógicas a una instrucción PID



Cómo establecer la salida

Puede reemplazar una estación de control manual con un interruptor de regulación manual e interruptores de botones pulsadores y simular la función PID con la lógica de escalera.

Use el modo de establecimiento de salida para introducir un valor que representa un porcentaje de la salida de la variable de control.

Generalmente, se desea introducir un valor desde una interface de operador. La tabla siguiente indica el procedimiento si se desea usar el modo de establecimiento de salida.

Tabla 14.A
Procedimiento del modo de establecimiento de salida

	Bloque de control de número entero (N7:0)	Bloque de control (PD10:0)
Seleccione el modo automático	Modo:0 (0:automático/1:manual) (bit N7:0/1 = 0)	Modo de estación A/M = Automático (bit PD10:0.MO = 0)
Seleccione el modo para establecer la salida	MODO DE ESTABLECIMIENTO DE SALIDA: 1 (0:no/1sí) (bit N7:0/4 = 1)	Modo A/M de software = Manual (bit PD10:0.SWM = 1) Nota: en el monitor de datos, MODE-AUTO se cambia a MODE-SW MANUAL.
Introduzca el % en el valor de establecimiento de salida (0-100%)	PORCENTAJE DEL VALOR PARA ESTABLECER LA SALIDA (palabra N7:10 = valor de porcentaje)	% DE ESTABLECIMIENTO DE SALIDA (palabra PD10:0.S0 = valor de porcentaje)

Si el valor para establecer la salida es mayor que el límite CV superior o menor que el límite CV inferior y el límite de salida está habilitado y la instrucción está en el modo de establecimiento de salida, el procesador usa la salida real (no el valor de establecimiento de salida) para calcular el término acumulador integral para el cálculo de la transferencia sin perturbaciones.

Prealimentación o polarización de salida

Se puede prealimentar una perturbación desde la salida de sistema o polarización si se prealimenta cualquiera de estos valores en la palabra de prealimentación/polarización de la instrucción PID (palabra 6 PD.BIAS) del bloque de control. Uno de los valores debe tener un rango de -4095 a +4095 (números enteros) o -100% a +100% (punto flotante [coma flotante]).

El valor de prealimentación representa una perturbación alimentada en la instrucción PID antes de que la perturbación pueda cambiar la variable del proceso. La prealimentación se usa frecuentemente para controlar los procesos que poseen un retardo de transporte. Por ejemplo, un valor de prealimentación que representa “agua fría vertida en una mezcla caliente” puede acelerar la salida en vez de esperar hasta que la variable cambie como resultado del mezclado.

Un valor de polarización se puede usar para compensar una pérdida constante de energía del proceso controlado.

Cómo reanudar el último estado

La función de reanudar el último estado le permite aprovechar al máximo la función de retención del último estado del módulo de salida analógica. La función de reanudar el último estado permite que la instrucción reanude el cálculo del término integral del algoritmo PID a partir de su último valor de salida (en vez de cero) cuando retorna al modo marcha.

Si usa un archivo de datos de número entero para el bloque de control, establezca los bits según las pautas siguientes. Si usa un tipo de archivo de datos PD para el bloque de control, el procesador guarda el acumulador integral y lo usa cuando va del modo de programación al modo marcha.

Use esta función así:

- Establezca la palabra 0, bit 7 si configuró el módulo de salida analógica para retener el último estado si ocurre un fallo y cuando se cambia del modo marcha al marcha de programación
- Restablezca la palabra 0, bit 7 si configuró el módulo de salida analógica para desactivarse si ocurre un fallo y cuando se cambia del modo marcha al modo de programación



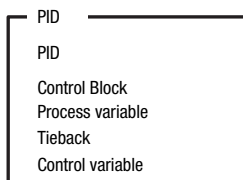
ATENCIÓN: Si desea usar esta función, establezca el bit 7 solamente después de que la instrucción se haya ejecutado por lo menos una vez (al momento del encendido) o cuando retorne al modo marcha. Si no permite que la instrucción PID se ejecute por lo menos una vez, puede ocurrir una operación inesperada de la máquina, lo cual puede causar daños al equipo y/o lesiones corporales.

La característica de reanudar el último estado está disponibles en los procesadores siguientes:

- Procesadores PLC-5 con características mejoradas de todas las series/revisiones
- PLC-5/12 de serie A/revisión C y posteriores
- PLC-5/15 de serie B/revisión H y posteriores
- PLC-5/25 de serie A/revisión D y posteriores

Instrucción PID

Descripción:



La instrucción PID es una instrucción de salida que controla las propiedades físicas, tales como la temperatura, presión, nivel de líquido o régimen de flujo, de los lazos de proceso.

La instrucción PID controla un lazo PID con entradas de un módulo de entrada analógica y una salida a un módulo de salida analógica. Para el control de temperatura, puede convertir la salida analógica en una salida de activación/desactivación de tiempo proporcional para hacer funcionar una unidad de calefacción o enfriamiento.

Ejecute la instrucción PID periódicamente a intervalos constantes con el uso de un temporizador, una interrupción temporizada seleccionable (STI) o el muestreo en tiempo real. El programa de lógica de escalera puede interactuar con el algoritmo PID cambiando las variables durante la operación, o usted puede cambiar las variables desde un terminal de programación o desde estaciones en una red de comunicación Data Highway™ o Data Highway Plus™.

La instrucción PID proporciona transferencias sin perturbaciones incluso cuando no usa la ganancia integral. Esto se logra porque la instrucción genera un término de polarización igual a la diferencia entre el término proporcional y la salida ajustada manualmente tal como sigue:

Si selecciona el modo manual con valor retenido:

$$\text{POLARIZACION} = (\text{VALOR RETENIDO} - \text{Pterm}) - \text{Dterm}$$

Si selecciona el modo manual para restablecer la salida:

$$\text{POLARIZACION} = (\text{modo de ESTABLECIMIENTO DE SALIDA} - \text{Pterm}) - \text{Dterm}$$

Normalmente, el procesador lee el valor del término de polarización que usted especifica en el bloque de configuración PID. Sin embargo, bajo una condición, el procesador escribirá un valor al término de polarización. Esto ocurre cuando la ganancia integral es igual a cero y el modo del lazo se cambia del modo manual al modo automático. El procesador calcula en base a datos originales el acumulador integral para intentar proporcionar una transferencia sin perturbaciones cuando va del modo manual al modo automático.

La función de transferencia sin perturbaciones está disponible con los niveles siguientes de revisión (o posteriores) de los procesadores:

- Procesadores PLC-5 con características mejoradas, todas las series y revisiones
- PLC-5/12 de serie A, revisión C
- PLC-5/15 de serie B, revisión H
- PLC-5/25 de serie A, revisión D

Los procesadores que tienen niveles de revisión anteriores proporcionan transferencias sin perturbaciones solamente cuando se incluyó un término integral en el algoritmo PID.

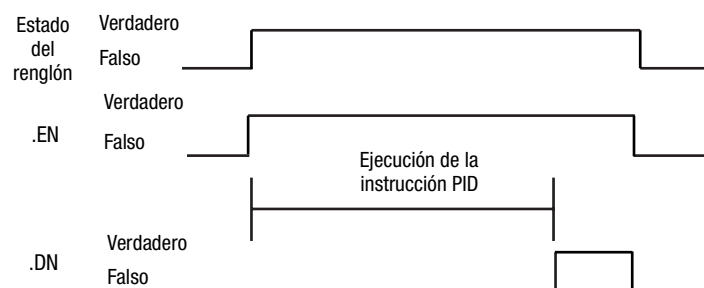
Cómo usar la característica de no calcular en base a datos originales

La característica de no calcular en base a datos originales es para las aplicaciones en que no desea que se sobrescriba el valor de polarización para la salida CV cuando está en el modo manual o para establecer la salida (manual de software). Cuando selecciona la característica de no calcular en base a datos originales y el modo es cualquiera de los modos manuales y la ganancia integral es cero, la instrucción PID no realiza el cálculo en base a datos originales en el término de polarización. Bajo esta condición, puede ocurrir una perturbación en la salida CV.

Bits de estado de operación

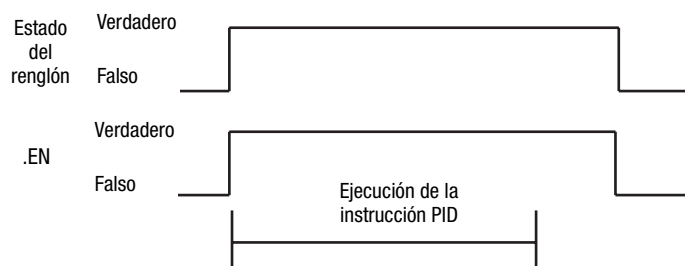
Bloque de números enteros

La instrucción PID de bloque de números enteros usa un bit de habilitación (.EN) para indicar que sus condiciones de renglón calificadas han realizado una transición de falso a verdadero. Las condiciones de renglón han permanecido verdaderas, lo cual indica que el bit de habilitación es verdadero. La única manera en que el bit de habilitación vuelve a hacerse falso es cuando las mismas condiciones calificadas se hacen falsas o si la lógica de escalera desenclava intencionalmente el bit de habilitación. El bit de efectuado (.DN) del bloque de números enteros se hace verdadero cuando la instrucción PID logra completar la ejecución y permanece verdadera hasta que las condiciones de renglón calificadas se hacen falsas.



Bloque PD

La instrucción PID del bloque PD tiene solamente un bit de habilitación (.EN) para indicar el estado de operación. Este bit indica que sus condiciones de renglón calificadas son verdaderas. En dicho caso, el bit de habilitación es verdadero (no se necesita una transición de falso a verdadero). La única manera en que el bit de habilitación se hace falso nuevamente es cuando estas mismas condiciones de renglón calificadas se hacen falsas. El bloque PD no usa un bit de efectuado.



Importante: A diferencia de la versión de bloque de números enteros, la instrucción PID del bloque PD vuelve a ejecutarse si el escán de programa encuentra este renglón nuevamente cuando el estado de renglón todavía es verdadero.

Cómo introducir los parámetros

Cuando introduce la instrucción, usted tiene que especificar las direcciones que son fundamentales para la operación de la instrucción. Después de que usted introduce estas direcciones, el software de programación muestra una pantalla mediante la cual se introducen los parámetros de operación de la instrucción.

El uso de bloques de control de números enteros en vez de bloques de control PD depende del procesador. Si usa un procesador PLC-5 clásico, el bloque de control PD no está disponible. En los procesadores PLC-5 con características mejoradas, los bloques de control de números enteros (N) y PD están disponibles. El bloque de control PD ofrece más flexibilidad (es decir, variables de punto flotante [coma flotante], mejor resolución – 12 bits en vez de 16 bits).

Las direcciones que se introducen son:

Parámetro:	Definición:
Bloque de control	un archivo que almacena los bits de estado y control, constantes, variables y parámetros para uso interno PID. Según el tipo de datos que use, una pantalla de configuración diferente aparece para que introduzca la información PID (vea las secciones siguientes para obtener más información). Si tiene un procesador PLC-5 con características mejoradas, puede usar un bloque de control de números enteros o un bloque de control PD. Si se usa un archivo PD, las palabras 0 y 1 son palabras de estado. Las palabras 2-80 almacenan los valores PID. Si usa un bloque de control de números enteros, los cálculos PID se realizan usando valores enteros. Si usa un bloque de control PD, los cálculos PID se realizan usando valores de punto flotante (coma flotante). Si tiene un procesador PLC-5 clásico, debe usar un archivo de números enteros (N) para el bloque de control. Si se usa un archivo de números enteros, la palabra 0 es la palabra de estado. Las palabras 1-22 almacenan los valores PID.
Variable del proceso	una dirección de palabra que almacena el valor de entrada del proceso.
Valor retenido	una dirección de palabra que se usa para implementar la transferencia sin perturbaciones cuando se usa una estación de control manual. El valor retenido es una salida de una instrucción BTR desde la estación.
Variable de control	una dirección de palabra a la cual la instrucción PID envía su valor de salida PID calculado. Nota: si un valor mayor que 4095 se escribe a la ubicación de la "variable de control" de la instrucción PID de tipo de números enteros, la salida de la instrucción PID obtiene un offset permanente el cual se puede eliminar si se escribe a la "variable de control" con un valor entre 0 y 4095. Esto ocurre cuando usted escribe a esta ubicación mediante la lógica de renglón o cuando escribe directamente a la ubicación de la tabla de datos. Nota: la instrucción PID de tipo de archivo PD no funciona así.

Cómo usar un tipo de archivo de datos de números enteros para el bloque de control

Cuando se usa un tipo de archivo de datos de números enteros para el bloque de control, la pantalla de monitoreo de datos para la instrucción PID muestra la información siguiente. Algunas porciones de estos datos se muestran para visualización solamente. Para otras porciones se especifican los valores (Tabla 14.B).

Tabla 14.B
Descripciones de parámetros de la instrucción PID (bloque de control de números enteros)

Parámetro:	Descripción:
Equation	Seleccione si desea usar las ganancias independiente (0) o dependiente (1). Se muestra uno de los siguientes: INDEPENDENT (0) – para ganancias independientes DEPENDENT (1) – para ganancias dependientes (ISA) Use las ganancias dependientes cuando desee usar los métodos de ajuste de lazo estándar. Use las ganancias independientes cuando desee que las tres constantes de ganancia (P, I y D) operen independientemente.
Modo	Muestra el modo de operación: AUTOMATIC (0) – control PID automático MANUAL (1) – control de una estación de control manual Establece el uso del parámetro de valor retenido para la operación manual
Error	Muestra en pantalla uno de los siguientes valores de error: Reverse acting: 0 = SP-PV Direct acting: 1 = PV-SP
Límite de salida	Muestra si la instrucción enclava o no enclava la salida en los valores de límites alto y bajo. Se muestra uno de los siguientes: NO (0) – no se enclava la salida YES (1) – se enclava la salida El algoritmo PID tiene una característica de bloqueo de acción integral que impide que el término integral se haga demasiado grande cuando la salida alcanza los límites de alarma alta o baja. Si se alcanzan los límites, el algoritmo deja de calcular el término integral hasta que la salida se encuentre nuevamente dentro del rango.
Modo de establecimiento de salida	Selecciona el uso del porcentaje del valor de salida para la operación manual
Escalado del punto de ajuste	Selecciona si el punto de ajuste se debe interpretar como valor en las unidades de ingeniería o un valor no escalado (0 a 4095)
Entrada de derivada	Selecciona un término de derivada según cambios en PV o cambios en error
Reanudación del último estado	Se selecciona para reanudar el último estado o para retener el último estado

(Continúa)

Parámetro:	Descripción:
Estado de banda muerta	Se establece si el PV se encuentra dentro del rango de banda muerta seleccionado. De lo contrario, se restablece
Alarma de límite CV superior	Se establece si el VC es mayor que el porcentaje de la palabra de límite CV superior
Alarma de límite CV inferior	Se establece si el VC es menor que el porcentaje de la palabra de límite CV inferior
Punto de ajuste fuera de rango	Muestra si el punto de ajuste se encuentra o no se encuentra fuera del rango de las unidades de ingeniería seleccionadas en la pantalla de configuración PID. Se muestra uno de los siguientes: NO (0) – punto de ajuste dentro de rango YES (1) – punto de ajuste fuera de rango Nota: ocurre un fallo mayor del procesador si el punto de ajuste se encuentra fuera de rango cuando la instrucción se habilita por primera vez.
PID efectuado	Muestra si la instrucción PID se ha completado (1 = efectuado; 0 = no efectuado)
PID habilitado	Muestra si la instrucción PID se ha habilitado (1 = habilitado; 0 = no habilitado)
Prealimentación	Introduzca un valor entre -4095 y 4095 para la cantidad de prealimentación. El programa de lógica de escalera puede introducir un valor de prealimentación para mover la salida anticipando de una perturbación. Este valor se usa frecuentemente para controlar un proceso que tiene un retardo de transporte.
Entrada escalada máxima	Introduzca el número entero (-32,768 a 32,767) que es el valor máximo disponible del módulo analógico. Por ejemplo, use 4095 para un módulo que tiene un rango de 0 a 4095.
Entrada escalada mínima	Introduzca el número que es el valor mínimo disponible del módulo analógico. Por ejemplo, use 0 para un módulo que tienen un rango de 0 a 4095.
Banda muerta	Para una banda muerta no escalada, introduzca un valor en las unidades de ingeniería que seleccionó en la pantalla de configuración PID. El rango válido es 0 a 4095 no escalado y -32,768 a +32,767 escalado. Nota: la banda muerta es paso por cero.
Porcentaje del valor de establecimiento de salida	Introduzca un porcentaje (0-100%) que se usa para como la salida CV cuando se selecciona el 'modo de establecimiento de salida'.
Porcentaje del límite superior CV	Introduzca un porcentaje (0-100%) por encima del cual el algoritmo enclava la salida.
Porcentaje del límite inferior CV	Introduzca un porcentaje (0-100%) por debajo del cual el algoritmo enclava la salida.
Valor PV escalado	Muestra datos del módulo de entrada analógica que la instrucción escala según las mismas unidades de ingeniería que seleccionó para el punto de ajuste.
Error escalado	Muestra el error actual en unidades de ingeniería escaladas
Porcentaje CV actual	Muestra el valor de salida actual de variable controlada como porcentaje
Punto de ajuste	Introduzca un número entero. El rango válido es 0 a 4095 (no escalado) o Smin-Smax (unidades de ingeniería escaladas)
Ganancia proporcional (K_p)	Introduzca un número entero. El rango de entrada válido es 0 a 32,767 (sin unidades) o K_p 0-32,767. El procesador divide el valor de entrada entre 100 para los cálculos.

(Continúa)

Parámetro:	Descripción:
Tiempo de restablecimiento (T_i) minutos/repetición	Introduzca un número entero. El rango de entrada válido para T_i es 0 a 32,767 (minutos multiplicados por 100). El procesador divide automáticamente el valor de entrada entre 100 para los cálculos. El rango de entrada válido para K_i es 0 a 32,767 (segundos inversos multiplicados por 100). El procesador divide automáticamente la entrada entre 1000 para los cálculos.
Velocidad de derivada (T_d)	Introduzca un número entero. El rango de entrada válido es 0 a 32,767 o KD 0 a 32,767. El procesador divide el valor de entrada entre 100 para los cálculos.
Tiempo de actualización del lazo	Introduzca un tiempo de actualización (mayor o igual a 0.01 segundo) a 1/5 a 1/10 veces el período natural de la carga (constante de tiempo de carga). El rango de entrada válido es 1 a 32,767 segundos. El procesador divide el valor de entrada entre 100 para los cálculos. La constante de tiempo de carga debe ser mayor que: 1 ms (algoritmo) + tiempo de transferencia en bloques (ms) Habilite periódicamente la instrucción PID a un intervalo constante igual al tiempo de actualización. Para los tiempos de actualización que son menores que 100 mseg, use una STI. Cuando los tiempos de actualización sean mayores que 100 mseg, use un temporizador o un muestreo en tiempo real. Nota: si elimina un tiempo de actualización o introduce un tiempo de actualización negativo, ocurre un fallo mayor la primera vez que el procesador ejecuta la instrucción PID.

Cómo usar los valores del bloque de control

La palabra 0 del bloque de control contiene los bits de estado y control. La Tabla 14.B muestra los valores almacenados en cada palabra del bloque de control.

Tabla 14.C
Bloque de control PID (bloque de control de números enteros)

Pal.:	Contiene:	Término:	Rango de entrada:
0	Bit 15 Habilitado (EN) Bit 13 Efectuado (DN) Bit 11 Punto de ajuste fuera de rango Bit 10 Alarma de salida, límite inferior Bit 9 Alarma de salida, límite superior Bit 8 banda muerta; se establece cuando el error está en la banda muerta Bit 7 Reanudación del último estado (0=sí; 1=retener el último estado) Bit 6 Acción de derivada (0=PV, 1=error) Bit 5 Desactivación de escalado del punto de ajuste (0=no, 1=sí) Bit 4 Establecimiento de la salida (0=no, 1=sí) Bit 3 Límite de salida (0=no, 1=sí) Bit 2 Control (0=retroceso, 1=directo) Bit 1 Modo (0=automático, 1>manual) Bit 0 Ecuación (0=independiente, 1=ISA) Nota: durante el preescán, los bits 8, 9 y 10, más los valores enteros de acumulador integral y error de derivada, se ponen en cero y el valor de registro del error de los escanes anteriores se establece en 32,767.		
1	Reservado		
2	Punto de ajuste	SP	0 a 4095 (sin escala) Smin–Smax escala)
Nota: los términos indicados por un asterisco (*) se introducen como $Y_y \times 100$. El término mismo es Y_y . El término indicado por dos asteriscos (**) se introduce como $Y_y \times 1000$. El término mismo es Y_y .			

(Continúa)

Pal.:	Contiene:	Término:	Rango de entrada:
3	Independiente: Ganancia proporcional x 100 (sin unidades)	K_p^*	0-32,767
	ISA: Ganancia de controlador x 100 (sin unidades)	K_c^*	0-32,767
4	Independiente: Ganancia integral x 1000 (1/seg)	K_i^{**}	0-32,767
	ISA: Término de restablecimiento x 100 (minutos por repetición)	T_i^*	0-32,767
5	Independiente: Ganancia de derivada x 100 (segundos)	K_d^*	0-32,767
	ISA: Término de velocidad x 100 (minutos)	T_d^*	0-32,767
6	Prealimentación o polarización	FF/polar.	-4095-+4095
7	Escalado máximo	Smax	-32,768-+32,767
8	Escalado mínimo	Smin	-32,768-+32,767
9	Banda muerta	DB	0-4095 (sin escala) $S_{min}-S_{max}$ (escala)
10	Establecimiento de la salida	SETOUT	0-100%
11	Límite de salida máximo (porcentaje de salida)	Lmax	0-100%
12	Límite de salida mínimo (porcentaje de salida)	Lmin	0-100%
13	Tiempo de actualización del lazo x 100 (segundos)	dt	0-32,767
14	Valor PV escalado (mostrado en pantalla)		$S_{min}-S_{max}$
15	Valor de error escalado (mostrado en pantalla)		$S_{min}-S_{max}$
16	Salida (porcentaje de 4095)	CV	0-100%
17-22	Almacenamiento interno; no se usa		
Nota: los términos indicados por un asterisco (*) se introducen como $Y_y \times 100$. El término mismo es Y_y . El término indicado por dos asteriscos (**) se introduce como $Y_y \times 1000$. El término mismo es Y_y .			

Cómo usar un tipo de archivo PD para el bloque de control (procesadores PLC-5 con características mejoradas solamente)

Cuando se usa un tipo de archivo PD para el bloque de control, la pantalla de monitoreo de datos para la instrucción PID muestra la información siguiente. Algunas porciones de esta información son para visualización solamente. Para otras porciones se especifican los valores (Tabla 14.B).

Tabla 14.D
Descripciones de parámetro PID (bloque de control PD)

Parámetro	Mnemónico de dirección:	Descripción:
Punto de ajuste	.SP	Introduzca un número de punto flotante (coma flotante) en las mismas unidades de ingeniería que aparecen en la pantalla de configuración PID. El rango válido es -3.4 E+38 a +3.4 E+38.
Variable del proceso	.PV	Muestra en pantalla los datos del módulo de entrada analógica que la instrucción escala a las mismas unidades de ingeniería que seleccionó para el punto de ajuste.
Error	.ERR	Muestra en pantalla uno de los siguientes: Acción inversora: Error = PV-SP Acción directa: Error = SP-PV
Porcentaje de salida	.OUT	Muestra en pantalla el valor de salida de control del algoritmo PID (0-100%)
Modo	.MO .MO=0 .MO=1 .SWM=1	Muestra en pantalla el modo de operación: AUTO – control PID automático MANUAL – control desde una estación de control manual SW MANUAL – control manual simulado desde el monitor de datos o el programa de lógica de escalera
Alarma PV	.PVHA=1 .PVLA=1	Muestra si el PV se encuentra dentro de o excede los límites de alarma alto o bajo que seleccionó en la pantalla de configuración PID. Muestra en pantalla uno de los siguientes: NONE – PV se encuentra dentro de los límites de alarma HIGH – PV excede el límite de alarma alto (se usa con la banda muerta) LOW – PV excede el límite de alarma bajo (se usa con la banda muerta)
Alarma de derivación	.DVPA=1 .DVNA=1	Muestra si el error se encuentra dentro de o excede las alarmas de derivación alta o baja que seleccionó en la pantalla de configuración PID. Muestra en pantalla uno de los siguientes: NONE – el error se encuentra dentro de los límites de alarma de derivación POSITIVE – el error excede la alarma alta (se usa con la banda muerta) NEGATIVE – el error excede la alarma baja (se usa con la banda muerta)
Límite de salida	.OLH=1 .OLL=1	Muestra si la instrucción enclava o no enclava la salida en los valores de límite alto y bajo (.MAXO y .MINO) que seleccionó en la pantalla de configuración PID. Muestra en pantalla uno de los siguientes: NONE – no se enclava la salida HIGH – se enclava la salida en el extremo superior (.MAXO) LOW – se enclava la salida en el extremo inferior (.MINO) El algoritmo PID tiene una característica de bloqueo de acción integral que impide que el término integral se haga demasiado grande cuando la salida alcanza los límites de alarma alta o baja. Si se alcanzan los límites, el algoritmo deja de calcular el término integral hasta que la salida se encuentra nuevamente dentro del rango.

(Continúa)

Parámetro	Mnemónico de dirección:	Descripción:
Punto de ajuste fuera de rango	.SPOR=0 .SPOR=1	Muestra si el punto de ajuste se encuentra o no se encuentra fuera del rango de las unidades de ingeniería que seleccionó en la pantalla de configuración PID. Muestra en pantalla uno de los siguientes: NO – punto de ajuste dentro de rango YES – punto de ajuste fuera de rango Nota: ocurre un fallo mayor del procesador si el punto de ajuste se encuentra fuera de rango cuando la instrucción se habilita por primera vez.
Error dentro de la banda muerta	.EWD=0 .EWD=1	Muestra si el error se encuentra dentro de o excede el valor de banda muerta que introduce en esta pantalla. La banda muerta es un paso por cero. Muestra en pantalla uno de los siguientes: RESET – el error sale de la zona de la banda muerta SET – el error pasa la línea central de la banda muerta
PID inicializado	.INI=0 .INI=1	Cada vez que cambia un valor en el bloque de control, la instrucción PID necesita más del doble del tiempo para ejecutarse (hasta que se inicializa) durante el primer escán. Muestra en pantalla uno de los siguientes: NO – la instrucción PID no se inicializó después de que se cambiaran los valores del bloque de control YES – la instrucción PID permanece inicializada porque no se cambiaron los valores del bloque de control Atención: no cambie el rango de entrada ni las unidades de ingeniería cuando ejecute la instrucción. Si un cambio es necesario, debe restablecer este bit para reiniciarse. De lo contrario, la instrucción tiene un malfuncionamiento, lo cual puede resultar en daños al equipo y lesiones corporales eventuales.
Modo de estación A/M	.MO=0 .MO=1	Seleccione el control PID automático (0) o manual (1). Muestra en pantalla uno de los siguientes: AUTO (0) – control PID automático MANUAL (1) – control PID manual El control manual especificó que una salida desde una estación de control manual anula la salida calculada del algoritmo PID. Nota: el control manual anula el modo de establecimiento de la salida.
Modo de software A/M	.SWM=0 .SWM=1	Seleccione el control PID automático (0) o el modo de establecimiento de salida (1) para el control de software simulado. Muestra en pantalla uno de los siguientes: AUTO (0) – control PID automático SW MANUAL (1) – control PID de software simulado Se puede simular una estación de control manual con el monitor de datos cuando se programa un solo lazo. Para hacerlo, establezca .SWM en SW MANUAL e introduzca un valor de porcentaje de establecimiento de salida. Se puede simular una estación de control manual con la lógica de escalera, preseletores rotativos e interruptores de botones pulsadores cuando programa varios lazos. Para hacerlo, establezca .SWM en SW MANUAL y mueva un valor en el elemento para de establecimiento de salida .SO.

(Continúa)

Parámetro	Mnemónico de dirección:	Descripción:
Habilitación de estado	.EN=0 .EN=1	Seleccione el uso (1) o la inhibición (0) de este bit que muestra en pantalla la condición de renglón para que usted pueda determinar si la instrucción PID funciona. Muestra en pantalla uno de los siguientes: 0 – la instrucción no se ejecuta 1 – la instrucción se ejecuta
Ganancia proporcional	.KP	Introduzca un valor de punto flotante (coma flotante). El rango válido para las ganancias independiente y estándar es 0 a 3.4 E^{+38} (sin unidades).
Ganancia integral	.KI	Introduzca un valor de punto flotante (coma flotante). El rango válido para las ganancias independiente y estándar es 0 a 3.4 E^{+38} segundos inversos; el rango válido para las ganancias estándar es 0 a 3.4 E^{+38} minutos por repetición.
Ganancia de derivada	.KD	Introduzca un valor de punto flotante (coma flotante). El rango válido para las ganancias independientes es 0 a 3.4 E^{+38} segundos; el rango válido para las ganancias estándar es 0 a 3.4 E^{+38} minutos.
% de polarización de salida	.BIAS	Introduzca un valor (-100 a +100) para representar el porcentaje de salida que desea prealimentar o usar como polarización a la salida. El valor de polarización puede compensar la pérdida constante de energía del sistema. El programa de lógica de escalera puede introducir un valor de prealimentación para mover la salida en anticipación de una perturbación. Este valor se usa frecuentemente para controlar un proceso que tiene un retardo de transporte.
Porcentaje del valor retenido	.TIE	Muestra en pantalla un número (0 a 100) que representa el porcentaje del valor retenido general (0 a 4095) desde la estación de control manual. El algoritmo PID usa este número para realizar una transferencia sin perturbaciones cuando cambia del modo manual al modo automático.
Porcentaje de establecimiento de salida	.SO	Introduzca un porcentaje (0 a 100) de esta pantalla o de un programa de lógica de escalera para representar la salida controlada manualmente por software. Cuando selecciona el control simulado por software (.SWM = 1), la instrucción PID anula el algoritmo con el valor de establecimiento de salida (0 a 4095) para la transferencia al módulo de salida y lo copia a .OUT para mostrarlo en pantalla como porcentaje. La transferencia al control simulado por software se efectúa sin perturbaciones porque .SO (bajo su control) comienza a partir de la última salida del algoritmo automática. Varíe .SO solamente después de la transferencia. Para realizar la transferencia sin perturbaciones al cambiar del control simulado por software al control automático, el algoritmo PID cambia el término integral para que la salida sea igual al valor de establecimiento de salida.

Cuando se usa un tipo de archivo PD para el bloque de control, la pantalla del monitor de datos para la instrucción PID proporciona acceso a una pantalla de configuración PID. Desde la pantalla de configuración PID puede definir las siguientes características de la instrucción PID (Tabla 14.B).

Tabla 14.E
Descripciones de la configuración PID (bloque de control PD)

Parámetro:	Mnemónico de dirección:	Descripción:
Ecuación PID	.PE=0 .PE=1	Seleccione si desea usar las ganancias independiente (0) o dependiente (1). Se muestra uno de los siguientes: INDEPENDENT (0) – para ganancias independientes DEPENDENT (1) – para ganancias dependientes (ISA) Use las ganancias dependientes cuando desee usar los métodos de ajuste de lazo estándar. Use las ganancias independientes cuando desee que las tres constantes de ganancia (P, I y D) operen independientemente.
Derivada de	.DO=0 .DO=1	Seleccione la derivada del PV (0) o el error (1). Muestra en pantalla uno de los siguientes: PV (0) – para derivada PV ERROR (1) – para derivada de error Seleccione la derivada PV para obtener un control más estable cuando no cambie el punto de ajuste frecuentemente. Seleccione la derivada de error para obtener respuestas rápidas a cambios del punto de ajuste cuando el algoritmo pueda tolerar los sobreimpulsos.
Acción de control	.CA=0 .CA=1	Seleccione la acción inversa (0) o directa (1). Muestra en pantalla uno de los siguientes: REVERSE (0) – para la acción inversa ($E = SP - PV$) DIRECT (1) – para la acción directa ($E = PV - SP$)
Seguimiento PV	.PVT=0 .PVT=1	Indique si desea (1) o no desea (0) el seguimiento PV. Muestra en pantalla uno de los siguientes: NO (0) – sin seguimiento YES (1) – para seguimiento PV Seleccione la opción sin seguimiento si el algoritmo puede tolerar una perturbación cuando se cambia del control manual al control automático. Seleccione el seguimiento PV si desea que el punto de ajuste siga la PV en el control manual para la transferencia sin perturbaciones al control automático.
Tiempo de actualización	.UPD	Introduzca un tiempo de actualización (mayor que o igual a 0.01 segundo) a 1/5 a 1/10 del período natural de la carga (constante de tiempo de carga). La constante de tiempo de carga debe ser mayor que: 3 ms (algoritmo) + tiempo de transferencia en bloques (ms) Habilite periódicamente la instrucción PID a un intervalo constante igual al tiempo de actualización. Cuando el escán de programa esté cerca del tiempo de actualización requerido, use una STI para asegurar un intervalo de actualización constante. Cuando el escán de programa sea más rápido que el tiempo de actualización requerido, use un temporizador. Atención: si elimina un tiempo de actualización o introduce un tiempo de actualización negativo, ocurre un fallo mayor del procesador la primera vez que el procesador ejecuta la instrucción PID.

(Continúa)

Parámetro:	Mnemónico de dirección:	Descripción:
Lazo en cascada	.CL=0 .CL=1	Seleccione si el lazo no se usa (0) o se usa (1) en una cascada de lazos. Muestra en pantalla uno de los siguientes: NO (0) – no se usa en cascada YES (1) – se usa en cascada
Tipo de cascada	.CT=0 .CT=1	Si este lazo es parte de una cascada de lazos, seleccione si el lazo es el maestro (1) o el esclavo (0). Muestra en pantalla uno de los siguientes: SLAVE (0) – para un lazo de esclavo MASTER (1) – para un lazo maestro
Maestro a este esclavo	.ADDR	Si este lazo es un lazo de esclavo en cascada, introduzca la dirección del bloque de control del maestro. Se ignora el valor retenido en el lazo maestro de una cascada. Cuando cambia los lazos en cascada al control manual, el esclavo fuerza el maestro en el control manual. Cuando se habilita el seguimiento PV, el orden de eventos es: Slave.SP > Master.TIE > Master.OUT > Slave.SP Cuando retorne al control automático, cambie el esclavo primero y luego el maestro.
Unidad de ingeniería máxima	.MAXS	Introduzca el valor de punto flotante (coma flotante) en unidades de ingeniería que corresponden a la salida analógica de escala total del módulo. El rango válido es $-3.4 \times 10^{+38}$ a $+3.4 \times 10^{+38}$. Atención: no cambie este valor durante la operación porque puede ocurrir un fallo del procesador.
Unidad de ingeniería mínima	.MINS	Introduzca el valor de punto flotante (coma flotante) en unidades de ingeniería que corresponden a la salida analógica de cero del módulo. El rango válido es $-3.4 \times 10^{+38}$ a $+3.4 \times 10^{+38}$ (número escalado posteriormente). Atención: no cambie el valor escalado máximo durante la operación porque puede ocurrir un fallo del procesador.
Rango de entrada máximo	.MAXI	Introduzca el número de punto flotante (coma flotante) $(-3.4 \times 10^{+38} \text{ to } +3.4 \times 10^{+38})$ que es el valor máximo no escalado disponible del módulo analógico. Por ejemplo, use 4095 para un módulo que tiene un rango de 0 a 4095.
Rango de entrada mínimo	.MINI	Introduzca el número de punto flotante (coma flotante) $(-3.4 \times 10^{+38} \text{ to } +3.4 \times 10^{+38})$ que es el valor mínimo no escalado disponible del módulo analógico. Por ejemplo, use 0 para un módulo que tiene un rango de 0 a 4095.
Porcentaje alto de límite de salida	.MAXO	Introduzca un porcentaje (0 a 100) por encima del cual el algoritmo enclava la salida.
Porcentaje bajo de límite de salida	.MINO	Introduzca un porcentaje (0 a 100) por debajo del cual el algoritmo enclava la salida.
Alarma alta PV	.PVH	Introduzca un número de punto flotante (coma flotante) $(-3.4 \times 10^{+38} \text{ a } +3.4 \times 10^{+38})$ que represente el valor más alto que el sistema puede tolerar.
Alarma baja PV	.PVL	Introduzca un número de punto flotante (coma flotante) $(-3.4 \times 10^{+38} \text{ a } +3.4 \times 10^{+38})$ que represente el valor más bajo que el sistema puede tolerar.
Banda muerta de alarma PV	.PVDB	Introduzca un número de punto flotante (coma flotante) $(0-3.4 \times 10^{+38})$ que sea suficiente para minimizar las alarmas de perturbaciones. Esta es una banda muerta unilateral. El bit de alarma (.PVH o .PVL) no se establece hasta que la PV pasa la banda muerta y alcanza el límite de alarma (punto de cero DB). El bit de alarma permanece establecido hasta que la PV pasa nuevamente por la banda muerta y sale de la misma.

(Continúa)

Parámetro:	Mnemónico de dirección:	Descripción:
Alarma de derivación (+)	.DVP	Introduzca un número de punto flotante (coma flotante) $(0-3.4 \text{ E}^{+38})$ que especifique la mayor derivación de error por encima del punto de ajuste que el sistema puede tolerar.
Alarma de derivación (-)	.DVN	Introduzca un número de punto flotante (coma flotante) $(-3.4 \text{ E}^{+38}-0)$ que especifique la mayor derivación de error por debajo del punto de ajuste que el sistema puede tolerar.
Banda muerta de la alarma de derivación	.DVDB	Introduzca un número de punto flotante (coma flotante) $(0-3.4 \text{ E}^{+38})$ que sea suficiente para minimizar las alarmas de perturbaciones. Esta es una banda muerta unilateral. El bit de alarma (.PVH o .PVL) no se establece hasta que la PV pasa la banda muerta y alcanza el límite de alarma (punto de cero DB). El bit de alarma permanece establecido hasta que la PV pasa nuevamente por la banda muerta y sale de la misma.
Sin paso por cero	.NOZC=0 .NOZC=1	Seleccione el uso (1) o la inhibición (0) de la característica sin paso por cero: 0 – sin paso por cero inhabilitado 1 – sin paso por cero habilitado
Sin cálculo en base a datos originales	.NOBC=0 .NOBC=1	Seleccione el uso (1) la inhibición (0) de la característica sin cálculo en base a datos originales: 0 – sin cálculo en base a datos originales inhabilitado 1 – sin cálculo en base a datos originales habilitado
Sin filtro de derivación	.NDF=0 .NDF=1	Seleccione el uso (1) la inhibición (0) del filtro en el cálculo de la derivada. 0 – no se usa un filtro en el cálculo de la derivada 1 – se usa un filtro en el cálculo de la derivada

Cómo usar los valores del bloque de control

Las palabras 0 y 1 del bloque de control contienen los bits de estado y control. La Tabla 14.B muestra los valores almacenados en cada palabra del bloque de control.

Tabla 14.F
Bloque de control PID

Palabra:	Contiene:	Rango:
0	Bits de control/estado Bit 15 Habilitado (EN) Bit 11 Sin cálculo en base a datos originales (0=inhabilitado, 1=habilitado) Bit 10 Sin paso por cero (0=inhabilitado, 1=habilitado) Bit 9 Selección de cascada (maestro, esclavo) Bit 8 Lazo en cascada (0=no, 1=sí) Bit 7 Seguimiento de variable del proceso (0=no, 1=sí) Bit 6 Acción de derivada (0=PV, 1=error) Bit 5 Sin filtro de derivada (0=inhabilitado, 1=habilitado) Bit 4 Establecimiento de la salida (0=no, 1=sí) Bit 2 Acción de control (0=SP-PV, 1=PV-SP) Bit 1 Modo (0=automático, 1>manual) Bit 0 Ecuación (0=independiente, 1=ISA)	

(Continúa)

Palabra:	Contiene:	Rango:
1	Bits de estado Bit 12 PID inicializada (0=no, 1=sí) Bit 11 Punto de ajuste fuera de rango Bit 10 Alarma de salida, límite inferior Bit 9 Alarma de salida, límite superior Bit 8 DB; se establece cuando el error está en la DB Bit 3 Alarma baja de error Bit 2 Alarma alta de error Bit 1 Alarma baja de la variable de proceso (PV) Bit 0 Alarma alta de la variable de proceso (PV) Nota: el bit 12 se pone a cero durante el preescán.	
2, 3	Punto de ajuste	-3.4 E^{+38} a $+3.4 \text{ E}^{+38}$
4, 5	Independiente: Ganancia proporcional (sin unidades)	0 a $+3.4 \text{ E}^{+38}$
	ISA: Ganancia de controlador (sin unidades)	0 a $+3.4 \text{ E}^{+38}$
6, 7	Independiente: Ganancia integral (1/seg)	0 a $+3.4 \text{ E}^{+38}$
	ISA: Término de restablecimiento (minutos por repetición)	
8, 9	Independiente: Ganancia de derivada (segundos)	0 a $+3.4 \text{ E}^{+38}$
	ISA: Término de velocidad (minutos)	0 a $+3.4 \text{ E}^{+38}$
10, 11	Prealimentación o polarización	-100 a +100%
12, 13	Escalado máximo	-3.4 E^{+38} a $+3.4 \text{ E}^{+38}$
14, 15	Escalado mínimo	-3.4 E^{+38} a $+3.4 \text{ E}^{+38}$
16, 17	Banda muerta	0 a $+3.4 \text{ E}^{+38}$
18, 19	Establecimiento de la salida	0 a 100%
20, 21	Límite de salida máximo (porcentaje de salida)	0 a 100%
22, 23	Límite de salida mínimo (porcentaje de salida)	0 a 100%
24, 25	Tiempo de actualización de lazo (segundos)	
26, 27	Valor PV escalado (mostrado en pantalla)	
28, 29	Valor de error escalado (mostrado en pantalla)	
30, 31	Salida (porcentaje de 4095)	0 a 100%
32, 33	Valor alto de alarma de la variable de proceso	-3.4 E^{+38} a $+3.4 \text{ E}^{+38}$
34, 35	Valor bajo de alarma de la variable de proceso	-3.4 E^{+38} a $+3.4 \text{ E}^{+38}$
36, 37	Valor alto de alarma de error	0 a $+3.4 \text{ E}^{+38}$
38, 39	Valor bajo de alarma de error	-3.4 E^{+38} a 0
40, 41	Banda muerta de alarma de la variable de proceso	0 a $+3.4 \text{ E}^{+38}$
42, 43	Banda muerta de la alarma de error	0 a $+3.4 \text{ E}^{+38}$

(Continúa)

Palabra:	Contiene:	Rango:
44, 45	Valor de entrada máximo	-3.4 E^{+38} a $+3.4 \text{ E}^{+38}$
46, 47	Valor de entrada mínimo	-3.4 E^{+38} a $+3.4 \text{ E}^{+38}$
48, 49	Valor retenido para el control manual (0 a 4095)	0 a 100%
51	Número de archivo PID maestro	0 a 999; 0 a 9999 para los procesadores PLC-5 con características mejoradas solamente
52	Número de elemento PID maestro	0 a 999; 0 a 9999 para los procesadores PLC-5 con características mejoradas solamente
54-80	Almacenamiento interno; no se usa	

Consideraciones de programación

Cuando programe una instrucción PID, no cambie los siguientes valores cuando el procesador esté en el modo marcha:

- la selección de la ecuación ISA o ganancias independientes porque las constantes de ganancias PID no se intercambian directamente
- los valores de escala $S_{\min}$ y $S_{\max}$ porque un cambio puede poner el punto de ajuste fuera de rango y cambiar el rango de la banda muerta
- la selección de la acción de derivada según cambios en la PV o cambios en el error porque se cambiarán los valores internos

Errores de tiempo de ejecución

Si el punto de ajuste (SP) se encuentra fuera de rango ($SP < S_{\min}$ o $SP > S_{\max}$), el procesador produce un error de tiempo de ejecución cuando ejecuta la instrucción.

Si cambia SP, $S_{\min}$, o $S_{\max}$ para crear la condición anterior, la instrucción PID primero intenta usar el punto de ajuste válido anteriormente, continúa el control PID y establece el bit de error de punto de ajuste fuera de rango. Si la instrucción no encuentra un punto de ajuste anteriormente válido, produce un error de tiempo de ejecución.

Si introduce valores negativos para K_p , K_I , K_D , K_C , T_I , o T_D , la instrucción PID sustituye un cero por el valor negativo. Esto inhibe dicho término en la ecuación sin producir un error de tiempo de ejecución.

Cómo transferir datos a la instrucción PID

Use las instrucciones de transferencia en bloques para transferir datos entre los módulos de E/S analógicas y la instrucción PID. Use una instrucción BTR para los valores de entrada (variable de proceso y valor retenido). Use una instrucción BTW para la salida de control.

Haga que cada dirección de archivo de transferencia en bloques (entrada de archivo de datos) sea la misma dirección en la PID para la variable de proceso, valor retenido y salida de control respectivamente.

Todos los módulos de entrada analógica de Allen-Bradley no introducen datos en el mismo formato. Es necesario determinar dónde almacenar los datos de canal. Por ejemplo, los módulos detectores de temperatura (tales como 1771-IR y 1771-IXE) colocan palabras de estado ante las palabras que contienen datos de canal. Vea los documentos del módulo para obtener información acerca de dónde un módulo analógico almacena los datos de canal.

Consideraciones del lazo

El número de lazos PID, el tiempo de actualización del lazo y la ubicación de los módulos de entrada analógica son consideraciones importantes para usar la instrucción PID.

Número de lazos PID

El número de lazos PID que el procesador puede manejar depende del tiempo de actualización requerido por los lazos. Cuanto más largo es el tiempo de actualización y cuanto menos sofisticado es el control de lazo, tanto más son los lazos que el procesador puede controlar.

La suma del tiempo de transferencia en bloques en el peor de los casos asociada con las entradas analógicas más el tiempo requerido para un escán de programa deben ser menor que el tiempo requerido por los lazos.

Tiempo de actualización del lazo

La instrucción PID calcula una nueva salida de control cuando su renglón cambia de falso a verdadero cuando se usa un archivo de datos de números enteros para el bloque de control. Una instrucción PID con un bloque de control PD se ejecutará en cada escán en que el renglón sea verdadero. Usted puede usar una instrucción de un impulso para forzar que la instrucción con un bloque de control PD se ejecute solamente durante una transición de falso a verdadero. Vea los ejemplos al fin de este capítulo. El tiempo de actualización debe ser igual a la velocidad a la cual el renglón PID cambia entre falso y verdadero para que la instrucción opere según lo esperado. Una derivación de la velocidad al alternar del tiempo de actualización menoscaba significativamente la precisión de los cálculos PID.

Debe programar lazos de respuesta rápida (tiempos de actualización de menos de 100 ms) en la interrupción temporizada seleccionable (STI) juntos con las instrucciones correspondientes de transferencia en bloques. Desenclave el bit de habilitación PID para forzar la ejecución durante cada escán STI (si usa un archivo de datos PD para el bloque de control, no es necesario desenclavar el bit de habilitación). Debe colocar los módulos de E/S analógicas correspondientes en el chasis local cuando vea esta configuración.

Programe los lazos de respuesta más lenta (tiempos de actualización mayor que 100 ms) en el programa de lógica de escalera principal y use temporizadores o el muestreo en tiempo real para controlar el tiempo de actualización.

Cómo desactivar el escalado de las entradas

La instrucción PID debe usar datos no escalados (0 a 4095) de módulos de entrada analógica. Es posible que los módulos de entrada analógica usados tengan rangos escalados o no escalados. Si es posible, seleccione el rango no escalado de 0 a 4095.

Sin embargo, algunos módulos, tales como los módulos detectores de temperatura 1771-IR y 1771-IXE, no pueden generar datos en un rango no escalado. Para estos módulos es necesario programar la lógica aritmética para convertir la salida escalada al rango no escalado para la instrucción PID. Si usa un archivo de datos PD para el bloque de control, el procesador realiza esta eliminación de escalado internamente (vea las descripciones de .MAXI y .MIN en las características de configuración PID, página 14-22).

Use esta ecuación para convertir las salidas escaladas:

$$M_2 = (M_1 - S_{\min 1}) \frac{4095}{(S_{\max 1} - S_{\min 1})}$$

Variable	Descripción
M2	salida calculada
M1	valor medido del módulo en unidades escaladas
S _{max1}	valor máximo escalado del módulo
S _{min1}	valor mínimo escalado del módulo
S _{max1} - S _{min1}	rango escalado del módulo

Por ejemplo, la temperatura de un módulo 1771-IXE para el termopar de tipo J es 170°. Para convertirla en un valor no escalado, use estos valores:

$$M_2 = [170 - (-200)] \frac{4095}{[1200 - (-200)]}$$

$$M_2 = 1082 \quad \text{no escalado}$$

Si está seguro de que la temperatura del proceso siempre permanecerá dentro de un rango especificado, puede establecer los límites $S_{\min 1}$ y $S_{\max 1}$ en vez de los valores mínimo y máximo para el módulo del termopar. Este método mejora la resolución de la variable de proceso.



ATENCIÓN: Si establece los límites en vez de usar los límites de temperatura inferior y superior del módulo de termopar o RTD, debe mantener el proceso dentro de los límites especificados. De lo contrario, puede causar una operación inesperada, daños al equipo o lesiones corporales.

La Figura 14.3 muestra la lógica de escalera que debe añadir al programa PID. La Tabla 14.G indica las variables en este ejemplo.

Figura 14.3
Ejemplo de cómo desactivar el escalado de valores PID

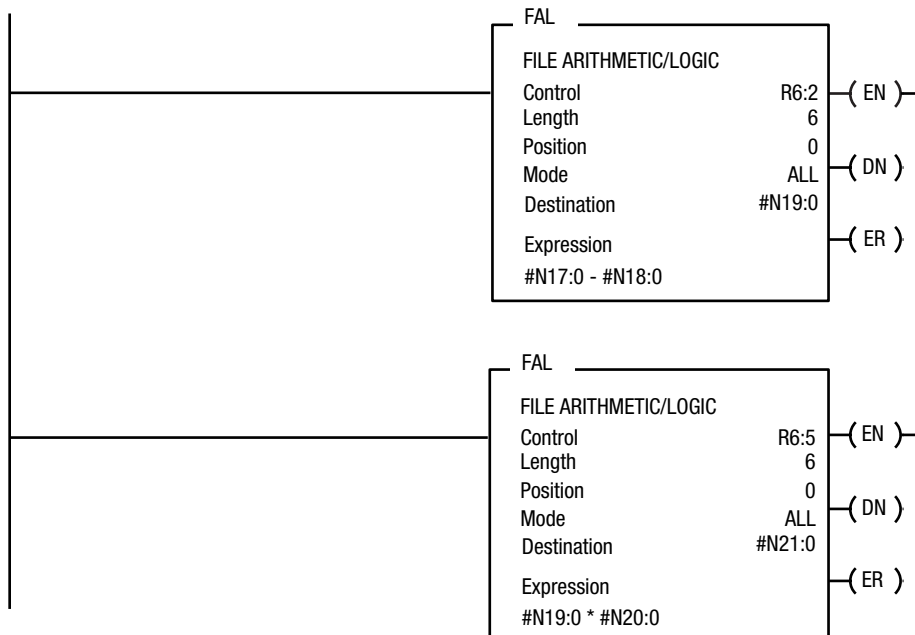


Tabla 14.G
Ejemplo de variables para desactivar el escalado de valores PID

Variable	Descripción
$S_{\max}$	valor de escalado máximo
$S_{\min}$	valor de escalado mínimo
$K = \frac{4095}{S_{\max} - S_{\min}}$	constante para cada canal
#N17:0	contiene valores M_1 para cada canal
#N18:0	contiene constantes $S_{\min}$ para cada canal
#N19:0	confirma el resultado de $M_1 - S_{\min}$ para cada canal
#N20:0	ubicación en que se almacena K para cada canal
#N21:0	contiene el valor no escalado resultante para cada canal

Ejemplos de PID

Los ejemplos siguientes suponen que los datos de canal se almacenan en el principio (primera palabra) del archivo de transferencia en bloques.

Ejemplos de bloques de números enteros (N)

Archivo de programa principal

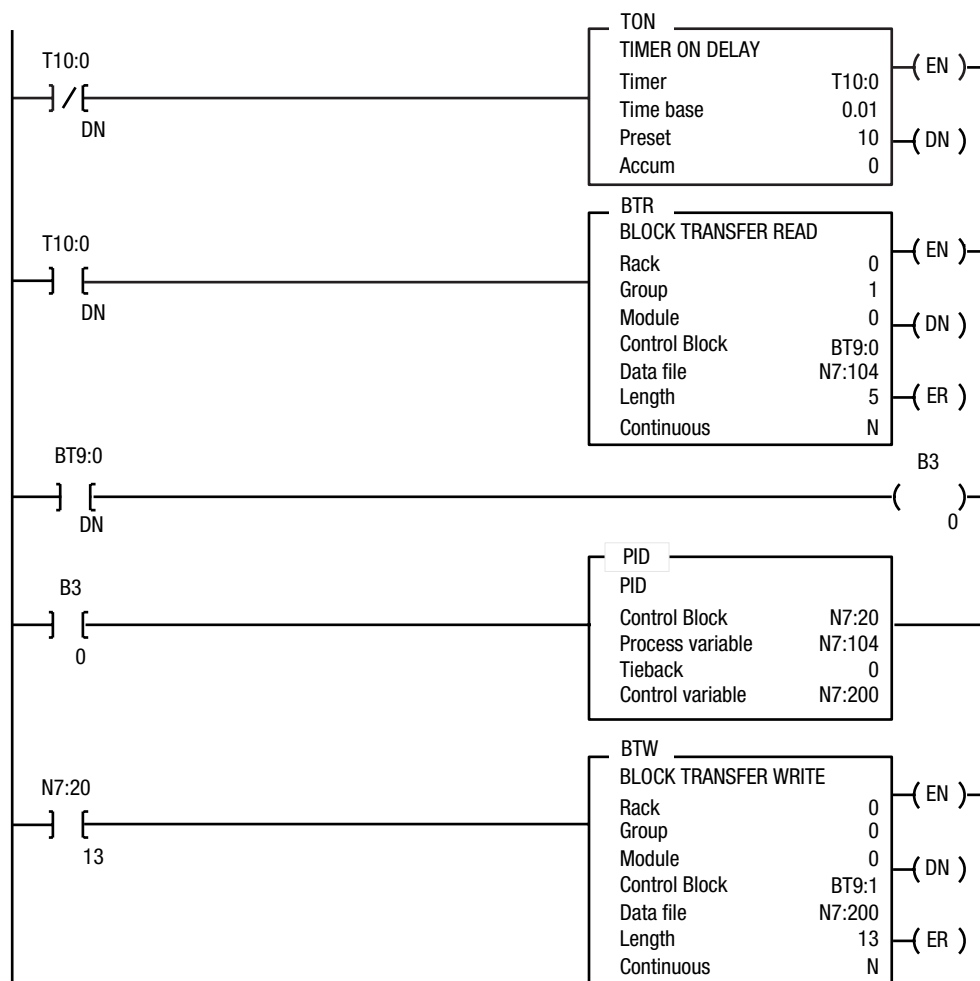
Cuando coloque la instrucción PID en el archivo del programa principal, controle el tiempo de muestreo con un temporizador donde el tiempo de actualización del lazo PID = valor preseleccionado del temporizador.

La ejecución basada en el temporizador usa un temporizador autónomo para la coordinación de eventos. Cuando el valor acumulador del temporizador alcanza su valor preseleccionado, activa la secuencia de actualización del lazo. El temporizador se restablece y se inicia inmediatamente para mantener un intervalo de actualización uniforme. Use la ejecución basada en el temporizador en aplicaciones de lazos “más lentas” o en aplicaciones donde hay pocos lazos. Vea la Figura 14.4 para obtener un ejemplo de programación.

La precisión del temporizador depende de la base de tiempo y el tiempo de escán total del procesador. Siempre seleccione la base de tiempo de 0.01 segundo para esta aplicación PID. Duplique la instrucción de temporizador en otra parte en el programa si el tiempo de escán del procesador (escán de E/S locales más escán de programa) es mayor que 2.5 segundos.

Puesto que las transferencias en bloques en el chasis local ocurren asíncronamente durante el escán de programa principal, es necesario un bit de almacenamiento para asegurar que el estado de la condición de entrada PID permanezca constante durante todo el escán de programa. Condicione todas las instrucciones usando este bit de almacenamiento.

Figura 14.4
Ejemplo de programación PID condicionada por un temporizador en el programa principal



Archivo de programa STI

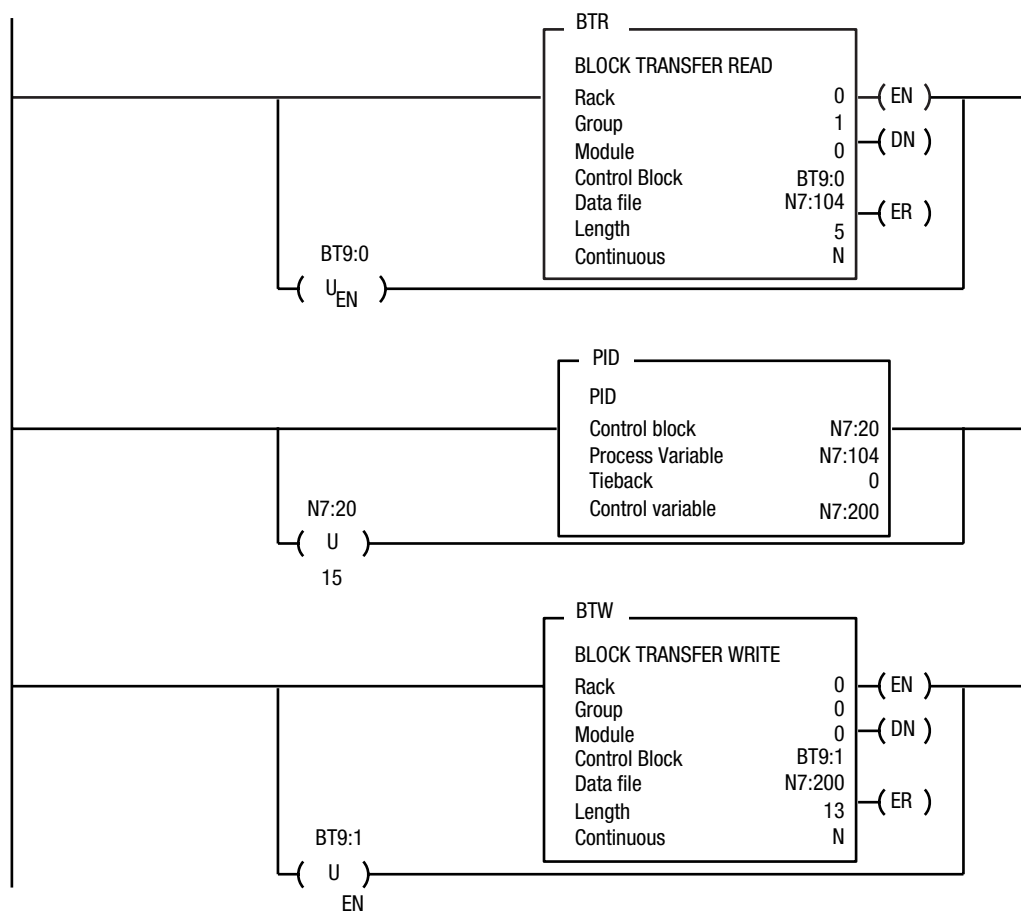
Cuando coloca la instrucción PID en un archivo de interrupción temporizada seleccionable (STI), la STI controla el tiempo de actualización del lazo (muestreo) donde el tiempo de actualización del lazo PID = intervalo STI.

En la STI hay un archivo de programa aparte que contiene toda la lógica necesaria para realizar la actualización del lazo. El procesador PLC-5 está configurado con una STI para ejecutar dicho archivo en el intervalo de actualización del usuario. Se prefiere la coordinación de lazos STI con lazos “más rápidos” o cuando se requiere más procesamiento de lazos en el intervalo de actualización especificado. Vea la Figura 14.5 para obtener ejemplos de programación.

La instrucción PID opera en los datos más recientes cuando las instrucciones de transferencia en bloques se incluyen en el archivo STI. Debe colocar los módulos de transferencia en bloques en el chasis local para esta aplicación PID. El desenclavamiento de los bits de habilitación PID y BT fuerza al procesador a ejecutar las instrucciones de transferencia en bloques y PID cada vez que la STI se habilita.

Importante: El escán de programa espera hasta que las instrucciones de transferencia en bloques en el archivo STI concluyan sus transferencias.

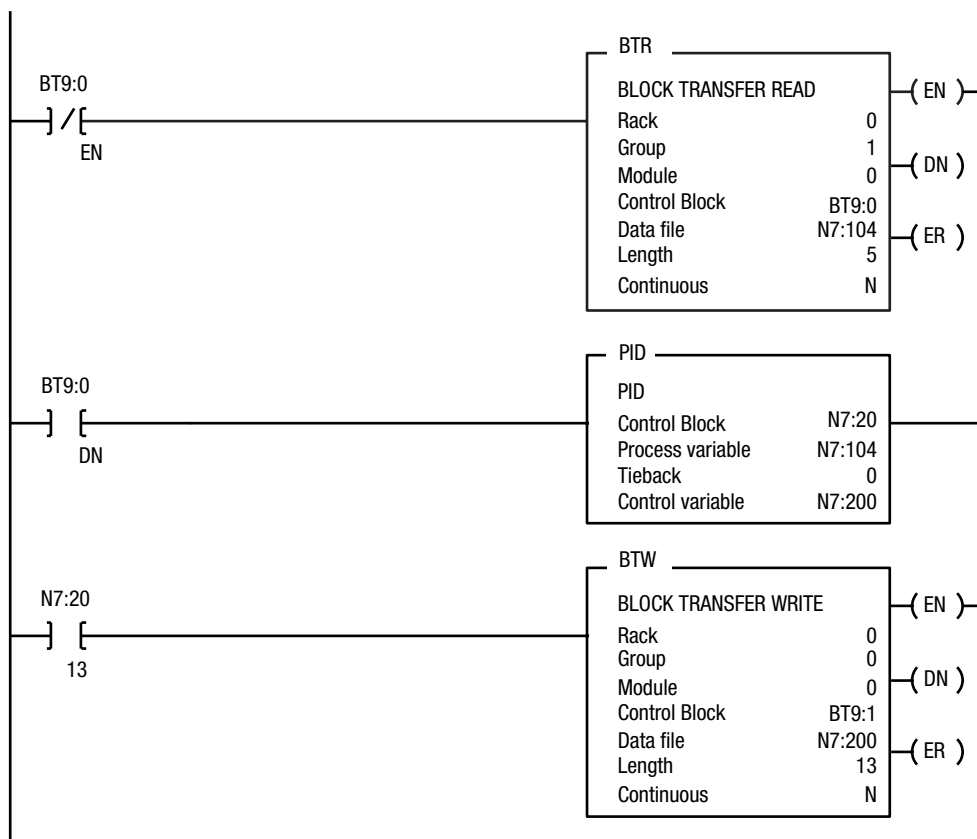
Figura 14.5
Ejemplo de programación PID en un archivo STI



Archivo de programa RTS

Con el muestreo en tiempo real (RTS), la disponibilidad de nuevos datos analógicos desde un origen de entrada analógica configurados para el muestreo en tiempo real activa la ejecución de la instrucción PID. Puesto que la configuración RTS de un módulo analógico no iniciará ni permitirá un BTR hasta que haya nuevos datos disponibles, el bit de efectuado del BTR puede condicionar el renglón de la instrucción PID. Esto asegura que la instrucción PID se ejecute solamente cuando haya nuevos datos analógicos disponibles en el intervalo RTS. Vea la Figura 14.6 para obtener ejemplos de programación donde el tiempo de actualización del lazo PID = intervalo RTS.

Figura 14.6
Ejemplo de programación PID en un archivo RTS



Ejemplos del bloque PD

Archivo de programa principal

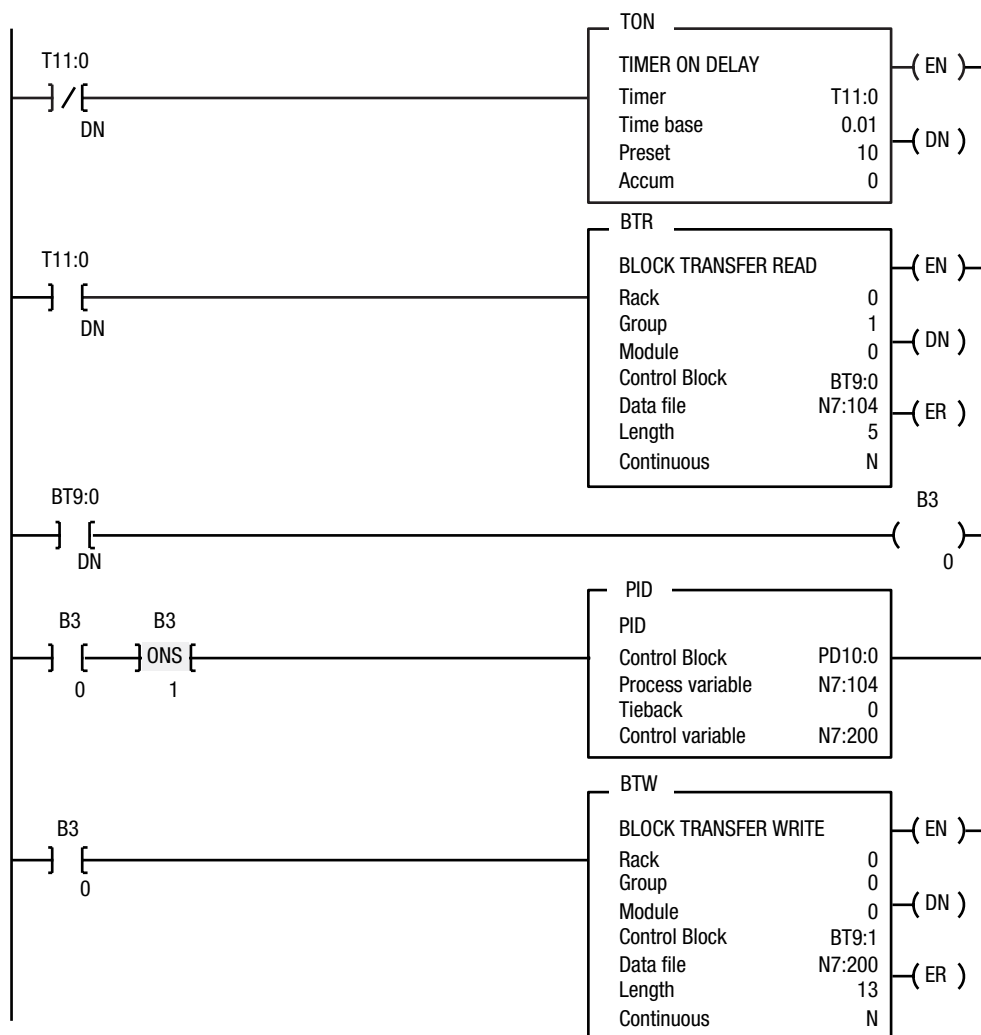
Cuando coloque la instrucción PID en el archivo de programa principal, controle el tiempo de muestreo con un temporizador donde el tiempo de actualización del lazo PID = valor preseleccionado del temporizador.

La ejecución basada en el temporizador usa un temporizador autónomo para la coordinación de eventos. Cuando el valor acumulador del temporizador alcanza su valor preseleccionado, activa la secuencia de actualización del lazo. El temporizador se restablece y se inicia inmediatamente para mantener un intervalo de actualización uniforme. Use la ejecución basada en el temporizador en aplicaciones de lazos “más lentas” o en aplicaciones donde hay pocos lazos. Vea la Figura 14.7 para obtener un ejemplo de programación.

La precisión del temporizador depende de la base de tiempo y el tiempo de escán total del procesador. Siempre seleccione la base de tiempo de 0.01 segundo para esta aplicación PID. Duplique la instrucción de temporizador en otra parte en el programa si el tiempo de escán del procesador (escán de E/S locales más escán de programa) es mayor que 2.5 segundos.

Puesto que las transferencias en bloques en el chasis local ocurren asíncronamente durante el escán de programa principal, es necesario un bit de almacenamiento para asegurar que el estado de la condición de entrada PID permanezca constante durante todo el escán del programa. Condicione todas las instrucciones usando este bit de almacenamiento.

Figura 14.7
Ejemplo de programación PID condicionada por un temporizador en el programa principal



Archivo de programa STI

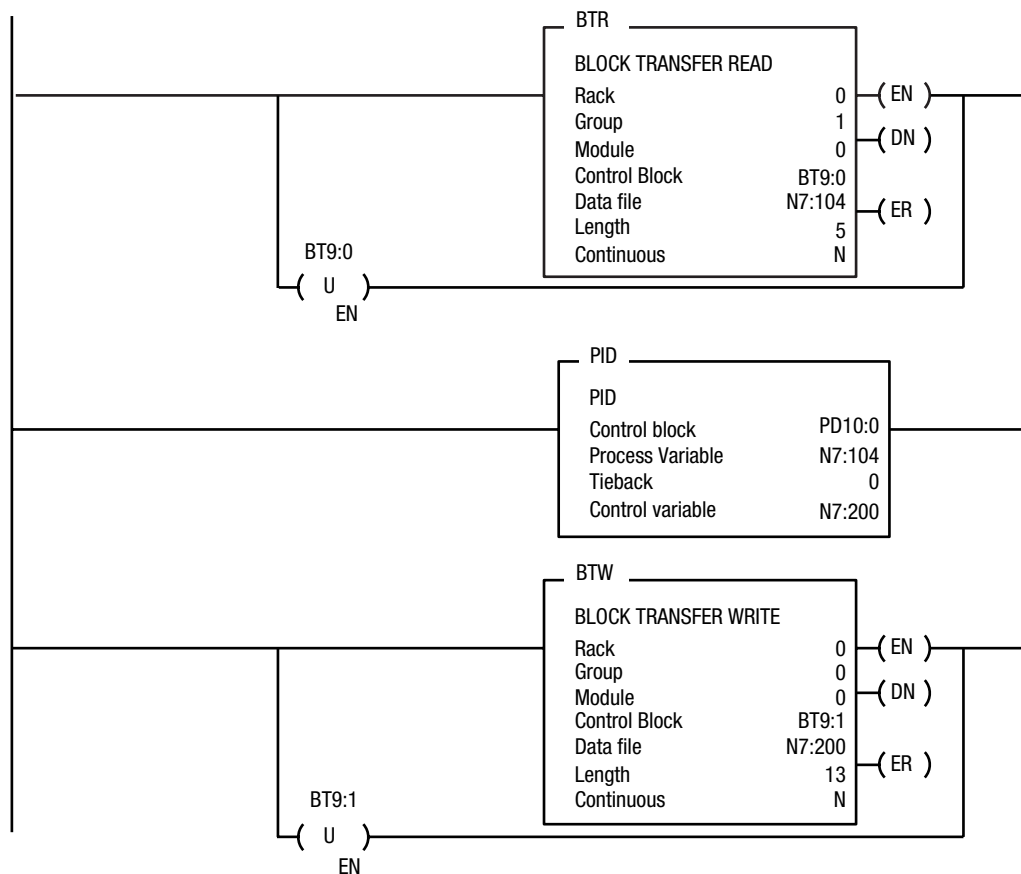
Cuando coloca la instrucción PID en un archivo de interrupción temporizada seleccionable (STI), la STI controla el tiempo de actualización del lazo (muestreo) donde el tiempo de actualización del lazo PID = intervalo STI.

En la STI hay un archivo de programa aparte que contiene toda la lógica necesaria para realizar la actualización del lazo. El procesador PLC-5 se configura con una STI para ejecutar dicho archivo en el intervalo de actualización del usuario. Se prefiere la coordinación de lazos STI con lazos “más rápidos” o cuando se requiere más procesamiento de lazos en el intervalo de actualización especificado. Vea la Figura 14.5 para obtener ejemplos de programación.

La instrucción PID opera en los datos más recientes cuando las instrucciones de transferencia en bloques se incluyen en el archivo STI. Debe colocar los módulos de transferencia en bloques en el chasis local para esta aplicación PID. El desenclavamiento de los bits de habilitación PID y BT fuerza al procesador a ejecutar las instrucciones de transferencia en bloques y PID cada vez que la STI se habilita.

Importante: El escán de programa espera hasta que las instrucciones de transferencia en bloques en el archivo STI concluyan sus transferencias.

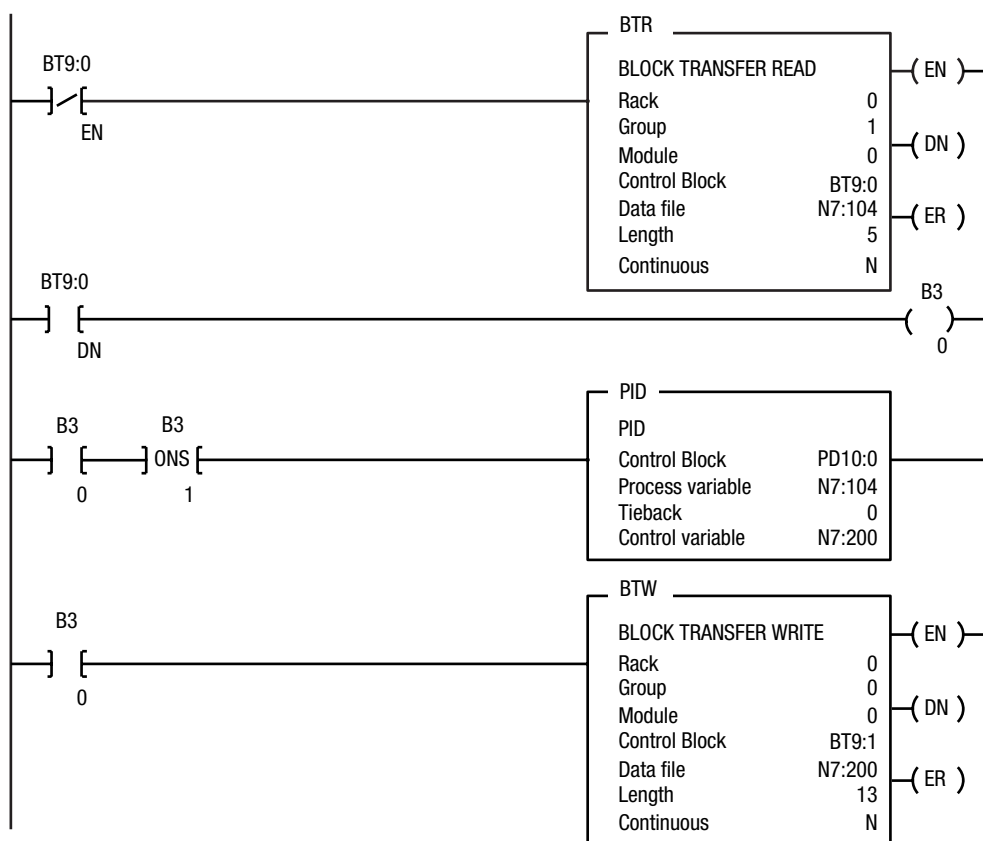
Figura 14.8
Ejemplo de programación PID en un archivo STI



Archivo de programa RTS

Con el muestreo en tiempo real (RTS), la disponibilidad de nuevos datos analógicos desde un origen de entrada analógica configurados para el muestreo en tiempo real activa la ejecución de la instrucción PID. Puesto que la configuración RTS de un módulo analógico no iniciará ni permitirá un BTR hasta que haya nuevos datos disponibles, el bit de efectuado del BTR puede condicionar el renglón de la instrucción PID. Esto asegura que la instrucción PID se ejecute solamente cuando haya nuevos datos analógicos disponibles en el intervalo RTS. Vea la Figura 14.9 para obtener ejemplos de programación donde el tiempo de actualización del lazo PID = intervalo RTS.

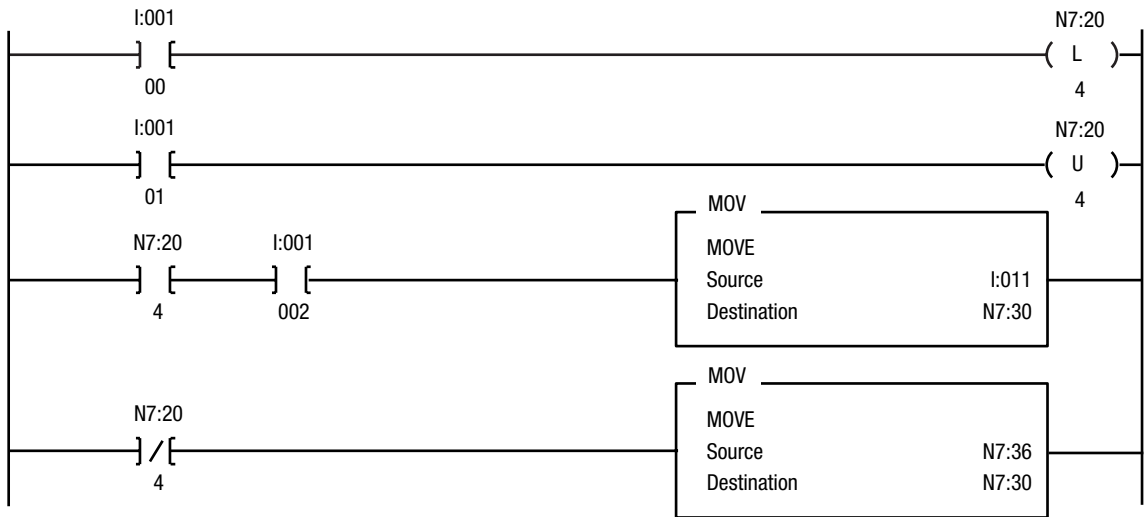
Figura 14.9
Ejemplo de programación PID en un archivo RTS



Simulación de lógica de escalera de una estación de control manual

Cuando programe la simulación de una estación de control manual, asegúrese de que no haya una estación de control manual de hardware conectada cuando el programa se habilite. Añada los renglones de la Figura 14.10 al programa PID en la Figura 14.4, Figura 14.5, Figura 14.7 ó Figura 14.8.

Figura 14.10
Ejemplo de programa para simular una estación de control manual



El último renglón en el ejemplo anterior es para el seguimiento de salida de la transferencia sin perturbaciones del modo automático al modo manual.

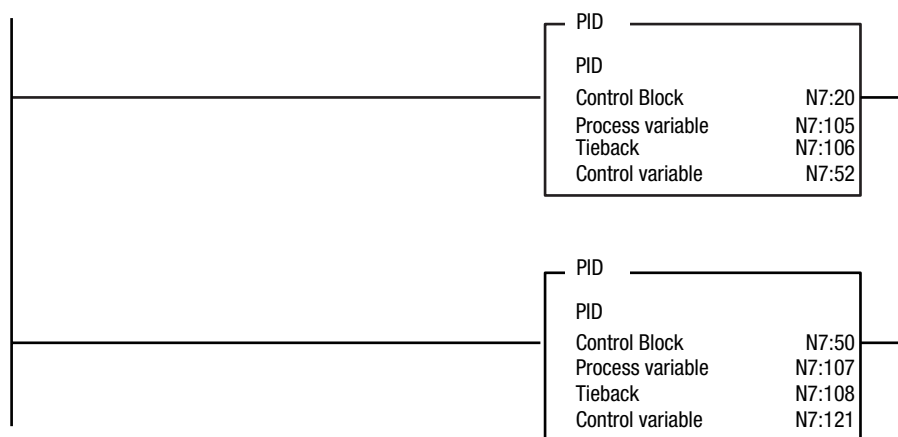
Dirección:	Descripción:
I:001/00	Interruptor de botón pulsador manual
I:001/01	Interruptor de botón pulsador automático
I:001/02	Introducción de interruptor de botón pulsador
I:011	Valor de salida manual
N7:20/4	Bit PID para establecer la salida
N7:30	Valor PID para establecer la salida
N7:36	Salida de control actual

Lazos en cascada

Puede poner dos lazos en cascada si asigna la salida de control del lazo exterior en el punto de ajuste del lazo interior. El punto de ajuste del lazo interior es la tercera palabra (palabra 2) del bloque de control de números enteros. Si el bloque de control del lazo interior es N7:50, direcciona la salida de control del lazo de salida en N7:52. Reemplace los renglones PID en la Figura 14.14 ó Figura 14.5 con los que aparecen en la Figura 14.11.

No debe escalar el punto de ajuste del lazo interior. Establezca el bit de escalado (palabra 0, bit 5) en 1 para inhibir el escalado del punto de ajuste.

Figura 14.11
Lazos en cascada



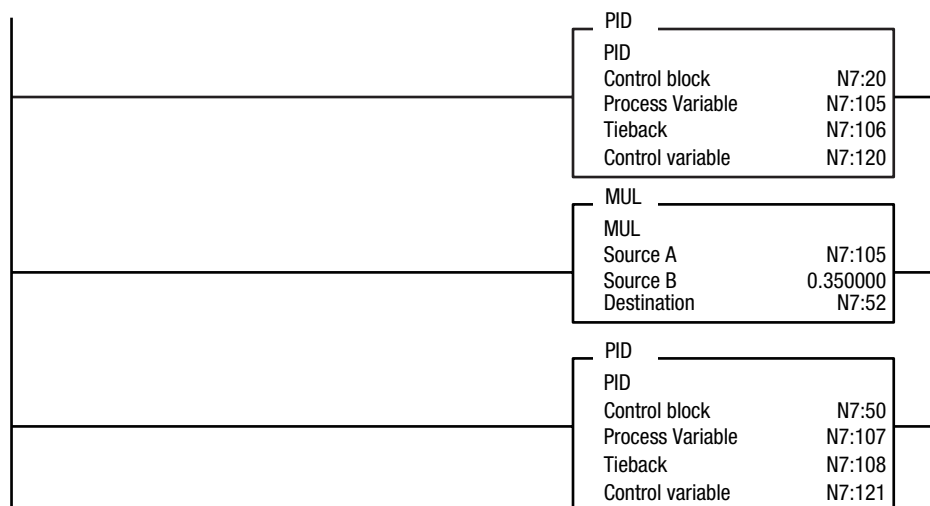
Control de relación

Puede mantener dos valores en una relación si usa una instrucción MUL. Se usan tres parámetros:

- el valor de comodín o no controlado
- el valor controlado
- la relación entre estos dos valores

Introduzca la dirección del valor controlado como destino. Introduzca la dirección del valor de comodín o no controlado como origen A. Introduzca la dirección del valor de relación o una constante de programa para la relación como origen B. Por ejemplo, añada los renglones de la Figura 14.12 al programa PID en la Figura 14.4 ó Figura 14.5.

Figura 14.12
Control de relación con una instrucción PID



Seguimiento de la variable de proceso

Cuando el programa está en el control manual, el programa puede forzar que el punto de ajuste sea igual a la variable de proceso (PV) moviendo la PV a la palabra de punto de ajuste (palabra 2 del bloque de control de números enteros) para realizar una transferencia sin perturbaciones del modo manual al modo automático. Si se escala el punto de ajuste, mueva la PV escalada del bloque de control PID directamente a la palabra de punto de ajuste. Si no se escala el punto de ajuste, mueva el valor no escalado de la dirección PV en la instrucción PID al punto de ajuste. Por ejemplo, añada los renglones de la Figura 14.13 al programa PID en la Figura 14.4 ó Figura 14.5.

Figura 14.13
Seguimiento de la variable de proceso

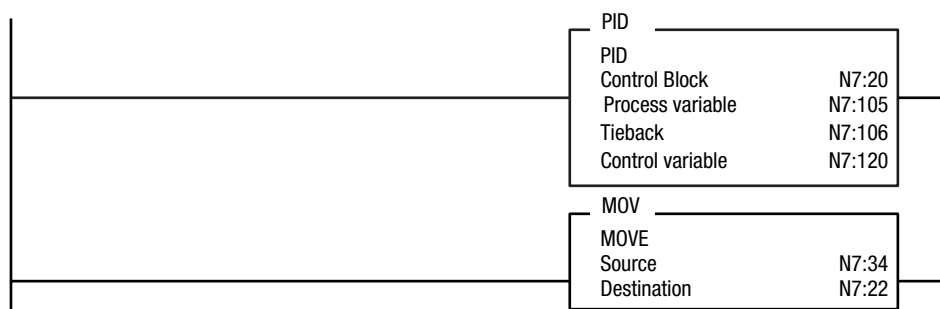


Figura 14.16
PID de PLC-5 (bloque PD) como lazos maestros/de esclavo

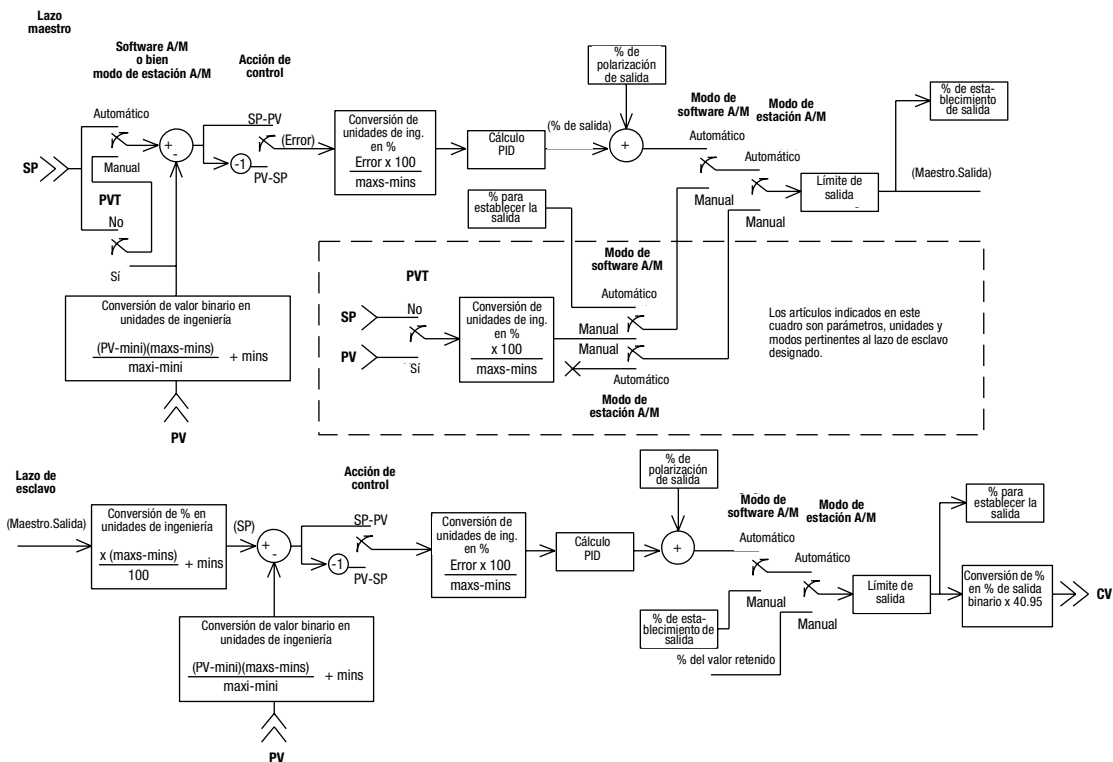
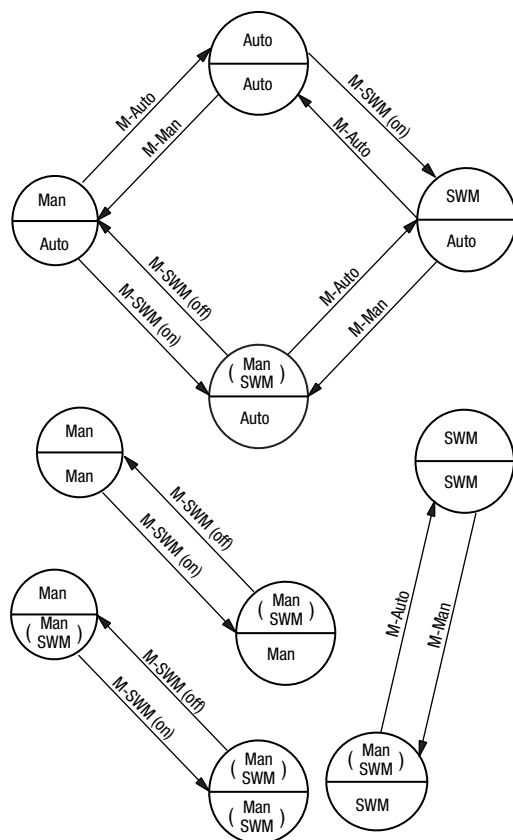


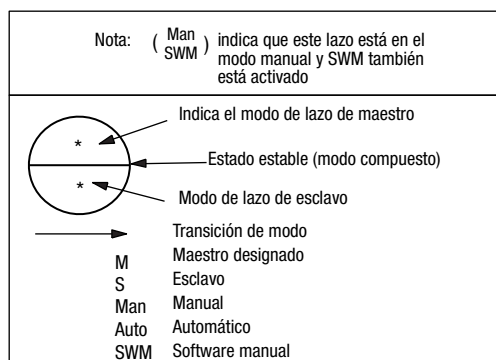
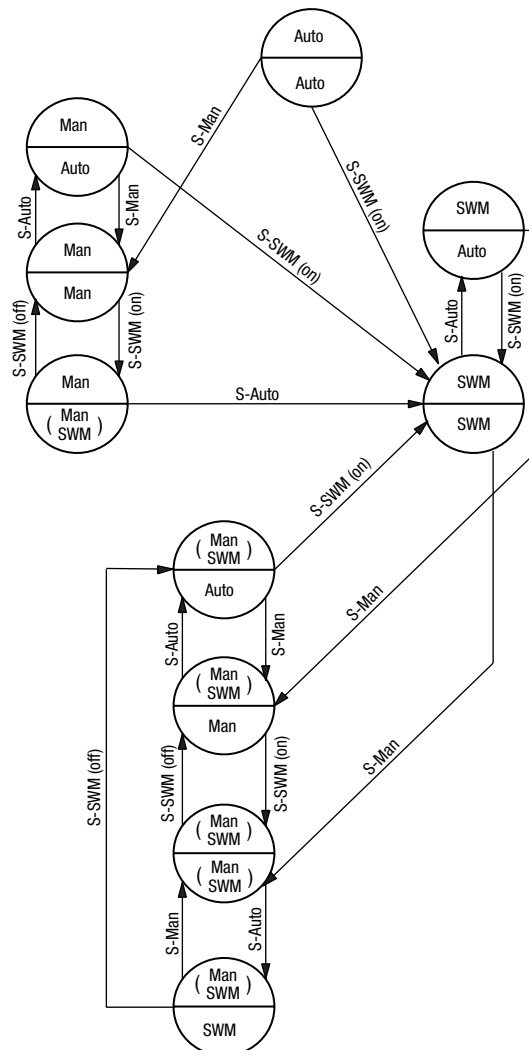
Figura 14.17

Transiciones de estado de enclavamiento de maestro/escalvo del bloque PD

Transiciones del lazo de maestro



Transiciones del lazo de esclavo



Cómo usar las instrucciones de transferencia en bloques y transferencia de E/S ControlNet

Instrucciones de transferencia en bloques BTR y BTW e instrucción de transferencia de E/S ControlNet CIO

Las instrucciones de transferencia en bloques le permiten transferir palabras hacia o desde un módulo de transferencia en bloques. Las instrucciones de E/S ControlNet le permite realizar transferencias no programadas hacia módulos de E/S en una red ControlNet™. La Tabla 15.A indica las instrucciones de transferencia en bloques y transferencia de E/S ControlNet disponibles.

Tabla 15.A
Instrucciones de transferencia en bloques y transferencia de E/S ControlNet disponibles

Si desea:	Use esta instrucción:	Que se encuentra en la página:
Transferir palabras hacia un módulo de transferencia en bloques	BTW	15-3
Transferir palabras desde un módulo de transferencia en bloques	BTR	15-3
Realizar transferencias no programadas hacia módulos de E/S en una red ControlNet	CIO	15-22

Cómo usar las instrucciones de transferencia en bloques

Vea el Apéndice C para obtener más información acerca de los operandos (y los tipos/valores de datos válidos de cada operando) usados por las instrucciones descritas en este capítulo.

Las instrucciones de transferencia en bloques le permiten transferir hasta 64 palabras simultáneamente hacia o desde un módulo de transferencia en bloques en un chasis de E/S locales o remotas. También puede transferir hasta 64 palabras simultáneamente entre un procesador supervisor (modo escáner) y un procesador configurado para el modo adaptador.

Los procesadores PLC-5 con características mejoradas tienen canales de comunicación configurables. Seleccione el escáner de E/S remotas, adaptador de E/S remotas o DH+. Las instrucciones de transferencia en bloques de la lógica de escalera no son necesarias cuando se usan los procesadores PLC-5 con características mejoradas en el modo adaptador.

La Tabla 15.A describe cómo transferir en bloques los datos hacia un rack local o remoto cuando el procesador está configurado para el modo escáner. La Figura 15.1 ilustra cómo ocurre la transferencia.

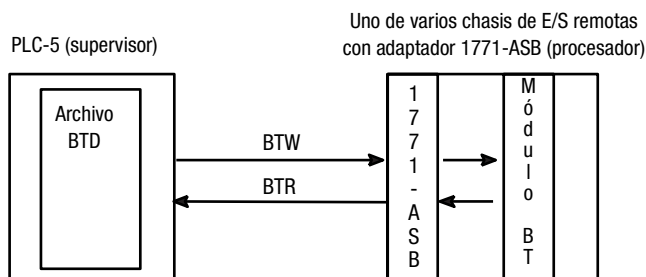
Tabla 15.B

Instrucciones de transferencia en bloques para los racks locales o remotos en el modo escáner

Si desea transferir datos:	Use:
Hacia el módulo de E/S BT	BTW (transferencia en bloques de escritura)
Desde el módulo de E/S BT	BTR (transferencia en bloques de lectura)

Figura 15.1

Operación de transferencia en bloques en el modo escáner



La Tabla 15.C describe cómo transferir en bloques los datos cuando el procesador está configurado para el modo adaptador. La Figura 15.2 ilustra cómo ocurre la transferencia.

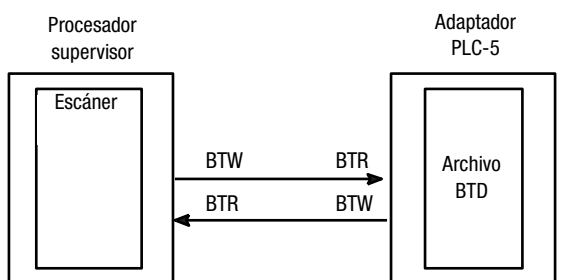
Tabla 15.C

Instrucciones de transferencia en bloques para el modo adaptador

Si desea transferir datos:	Use:
Desde el procesador supervisor	BTR (transferencia en bloques de lectura)
Hacia el procesador supervisor	BTW (transferencia en bloques de escritura)

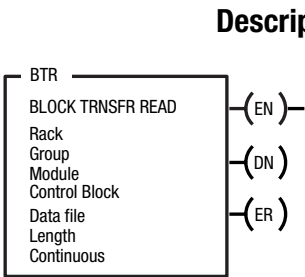
Figura 15.2

Operación de transferencia en bloques en el modo adaptador



Los dos procesadores ejecutan simultáneamente la instrucción opuesta de transferencia en bloques

Block Transfer Read (BTR) y Block Transfer Write (BTW)



Cuando el renglón se hace verdadero, la instrucción BTW indica al procesador que escriba los datos almacenados en el archivo de datos a la dirección de rack/grupo/módulo especificada. La instrucción BTR le indica al procesador que lea los datos de la dirección de rack/grupo/módulo y que los almacene en el archivo de datos.

Cola de solicitud de transferencia en bloques

Cuando una transición de falso a verdadero habilita una instrucción BTW o BTR, la solicitud de transferencia se pone en la cola:

Para este procesador:	La cola contiene hasta:
PLC-5 clásico	17 solicitudes de transferencia en bloques por rack lógico
PLC-5/11, 5/20, -5/30	64 solicitudes de transferencia en bloques a los racks remotos (máximo de 64 por pareja de canales – 1A/1B). No hay restricción de solicitudes a los racks locales.
PLC-5/40, -5/60, -5/80	128 solicitudes de transferencia en bloques a los racks remotos (máximo de 64 por par de canales – 1A/1B, 2A/2B). No hay restricción de solicitudes a los racks locales

El procesador ejecuta cada solicitud de transferencia en bloques según el orden en que se solicitó. Cuando el procesador cambia al modo programa, las transferencias en bloques se anulan.

Para los procesadores PLC-5 clásicos, cada número de rack tiene una cola de transferencia en bloques con un bit de cola llena correspondiente. La Tabla 15.D indica los bits de cola llena. Una vez que estos bits están establecidos, el programa de lógica de escalera debe ponerlos a cero. El programa debe monitorear continuamente estos bits de cola llena, los cuales se encuentran en el archivo de estado, palabra 7, bits 08 a 15. (Los procesadores PLC-5 con características mejoradas pueden tener transferencias en bloques ilimitadas en los racks locales, por lo tanto no tienen bits de cola llena.)

Tabla 15.D

Bits de cola llena para las solicitudes de transferencia en bloques (palabra 7)
– Procesadores PLC-5 clásicos

Bit	Descripción
S:7/8	La cola de transferencia en bloques para el rack 0 está llena
S:7/9	La cola de transferencia en bloques para el rack 1 está llena
S:7/10	La cola de transferencia en bloques para el rack 2 está llena
S:7/11	La cola de transferencia en bloques para el rack 3 está llena
S:7/12	La cola de transferencia en bloques para el rack 4 está llena
S:7/13	La cola de transferencia en bloques para el rack 5 está llena
S:7/14	La cola de transferencia en bloques para el rack 6 está llena
S:7/15	La cola de transferencia en bloques para el rack 7 está llena

El número de racks en el sistema depende del procesador que se use.

Una instrucción BTR o BTW escribe valores a su dirección de bloque de control (un archivo de números enteros de cinco palabras) cuando la instrucción se introduce. El procesador usa estos valores para ejecutar la transferencia.

Los procesadores PLC-5 con características mejoradas también tienen un tipo de archivo de transferencia en bloques (BT). Puede usar los programas existentes con los tipos de archivo de números enteros, pero el nuevo tipo de archivo BT facilita el direccionamiento. Por ejemplo, si necesita dos archivos de control, puede usar BT10:0 y BT10:1. Si usa los archivos de números enteros, deberá usar, por ejemplo, N7:0 y N7:5.

Cómo introducir los parámetros

Para programar una instrucción BTW o BTR, debe proporcionar al procesador la siguiente información que éste almacena en su bloque de control :

- **El rack** es el número del rack de E/S (00 a 27 octal) del cahsis de E/S en que se coloca el módulo de E/S receptor. La Tabla 15.E indica los rangos válidos para los números de racks.

Tabla 15.E

Rangos válidos para el número de rack en las instrucciones de transferencia en bloques

Procesador	Racks máximos	Rango válido para los números de rack (octal)
PLC-5/10, -5/11, -5/12, -5/15, -5/20, -5/VME	4	00-03
PLC-5/25, -5/30	8	00-07
PLC-5/40, -5/40L	16	00-17
PLC-5/60, -5/60L, -5/80	24	00-27

- **Group** es el número de grupo de E/S (0 a 7) que especifica la posición del módulo de E/S receptor en el chasis de E/S.
- **Module** es el número de ranura (0 a 1) dentro del grupo. Cuando se usa el direccionamiento a 2-slot, la ranura 0 es la ranura baja y la ranura 1 es la ranura alta. Debe usar 0 para el módulo cuando se usa el direccionamiento a 1 ó 1/2-slot.
- **El bloque de control** es un archivo de control de transferencia en bloques (BT) de seis palabras o un archivo de números enteros (N) de cinco palabras que controla la operación de la instrucción. Introduzca esta dirección de archivo sin el símbolo #. Este no es un archivo de control (tipo R).

Importante: Puede usar direcciones indirectas para la dirección de bloque de control en una instrucción BTR o BTW.

Importante: En un procesador PLC-5/40, -5/60 ó -5/80, el tipo de datos de transferencia en bloques (BT) se debe usar para las direcciones de rack mayores que 7.

El archivo de control de números enteros (N) de cinco palabras tiene la estructura siguiente:

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Palabra 0	EN	ST	DN	ER	CO	EW	NR	TO	RW	rack			group			slot
Palabra 1	conteo de palabras solicitado															
Palabra 2	conteo de palabras transmitido															
Palabra 3	número de tipo de archivo															
Palabra 4	número de elemento															

Para obtener información acerca de los bits de estado en la palabra 0, vea la página 15-8. Para obtener información acerca de las palabras 1 a 4, vea la página 15-10.

- **Data File** es la dirección del archivo de datos de entrada, salida, estado, número entero (N), punto flotante (coma flotante), binario, BCD o ASCII desde el cual (escritura) o hacia el cual (lectura) el procesador transfiere datos. Introduzca esta dirección de archivo sin el símbolo #.

Importante: No se puede usar las direcciones indirectas para las dirección de archivo de datos en una instrucción BTR o BTW.

- **Length** es el número de palabras del archivo de datos que se deben leer/escribir.

Si establece la longitud en:

El procesador:

0	Reserva 64 palabras para los datos de transferencia en bloques. El módulo de transferencia en bloques transfiere las palabras máximas que puede manejar.
1 a 64	Transfiere el número de palabras especificado.

Importante: Un elemento de punto flotante (coma flotante) consiste en dos palabras. Cuando especifica un valor en el campo de longitud para un archivo de datos de punto flotante (coma flotante), se lee/escribe solamente la mitad de dichos elementos de punto flotante (coma flotante). Por ejemplo, si especifica 64 para la longitud, en realidad se leen/escriben 32 elementos de punto flotante (coma flotante).

Importante: Las longitudes del archivo de datos de punto flotante (coma flotante) deben ser un número par.

- **Continuous** determina el modo de operación.

Si especifica:	La instrucción usa este modo:
----------------	-------------------------------

Sí	Continuo – una vez que el renglón se hace verdadero, la instrucción continúa transfiriendo datos hasta que el bit continuo (.CO) se restablece y el renglón es falso o usted edita la instrucción y especifica NO para el modo continuo.
----	--

No	No continuo – la instrucción se habilita cada vez que el renglón se hace verdadero y realiza solamente una transferencia de datos por transición de renglón.
----	--

Cómo usar los bits de estado

Para usar las instrucciones BTR y BTW correctamente, examine los bits de estado de la instrucción almacenados en el bloque de control. Estos bits están en la palabra 0 del bloque de control.



ATENCION: Aparte del bit continuo .CO (bit 11) y el bit de exceso de tiempo de espera .TO (bit 08), no modifique ningún bit de estado cuando la instrucción de transferencia en bloques esté habilitada. Puede ocurrir una operación inesperada, posibles daños al equipo y/o lesiones corporales.

Importante: Las etiquetas de bits (.EN, S.T, .CO, etc.) se pueden usar solamente con el tipo de archivo de transferencia en bloques (BT).

Este bit:	Se establece:
-----------	---------------

Habilitación .EN (bit 15)	cuando el renglón se hace verdadero. Este bit indica que la instrucción está habilitada (que la transferencia en bloques está en progreso). En el modo no continuo, el bit .EN permanece establecido hasta que la transferencia en bloques se completa o falla y el renglón se hace falso. En el modo continuo, una vez que el bit .EN está establecido, permanece establecido independientemente de la condición del renglón.
---------------------------	--

Inicio .ST (bit 14)	cuando el procesador comienza a transferir datos. El bit .ST se restablece durante la transición de falso a verdadero después de que se establece el bit .DN o .ER.
---------------------	---

Efectuado .DN (bit 13)	cuando se completa la transferencia en bloques, si los datos son válidos. El bit .DN se establece asincrónicamente al escán del programa para que el bit .DN se haga verdadero en cualquier momento después que se inicie la transferencia en bloques. El bit .DN se restablece la próxima vez que el renglón asociado va de falso a verdadero.
------------------------	---

Este bit:	Se establece:
Error .ER (bit 12)	cuando el procesador detecta el fallo de la transferencia en bloques. El bit .ER se restablece la próxima vez que el renglón asociado va de falso a verdadero.
Continuo .CO (bit 11)	cuando usted edita la instrucción para operación repetida de la transferencia en bloques después del primer escán, independientemente de que el procesador continúe escaneando el renglón del procesador. Restablezca el bit .CO si desea que la condición de renglón inicie las transferencias en bloques (retorno al modo no continuo). Si usa las transferencias en bloques continuas en un diagrama de función secuencial, vea el Apéndice B "Referencia SFC" de este manual.
Habilitado-esperando .EW (bit 10)	cuando la solicitud de transferencia en bloques entra a la cola. Si la cola está completa, este bit permanece restablecido hasta que haya espacio en la cola. El bit .EW se restablece cuando el renglón asociado va de falso a verdadero. En el modo continuo, una vez que el bit .EW se establece, permanece establecido. Use el bit .EW para verificar que una instrucción BTW o BTR esté en la cola antes de salir de un paso SFC.
No respuesta .NR (bit 09)	si el módulo de transferencia en bloques no responde a la primera solicitud de transferencia en bloques. El bit .NR se restablece cuando el renglón asociado va de falso a verdadero (no se usa con transferencias en bloques remotas).
Tiempo de espera .TO (bit 08)	si restablece el bit de tiempo de espera mediante la lógica de escalera o monitor de datos, el procesador intenta repetidamente enviar una solicitud de transferencia en bloques a un módulo sin respuesta durante cuatro segundos antes de establecer el bit .ER. Si establece el bit .TO mediante la lógica de escalera o monitor de datos, el procesador inhabilita el temporizador de cuatro segundos y solicita una transferencia en bloques una vez más antes de establecer el bit .ER.
Lectura-escritura .RW (bit 07)	controlado por la instrucción. Un 0 representa una operación de escritura. Un 1 representa una operación de lectura.



ATENCION: El procesador ejecuta las instrucciones de transferencia en bloques asincrónicamente con el escán de programa. El estado de estos bits se puede cambiar en cualquier punto en el escán de programa. Si examina estos bits en la lógica de escalera, copie el estado una vez a un bit de almacenamiento cuyo estado está sincronizado con el escán de programa. De lo contrario, los problemas de temporización pueden invalidar el programa, lo cual puede resultar en daños al equipo o lesiones corporales.

Importante: Cuando se usan tipos de archivo de números enteros (N) y transferencia en bloques (BT), se ponen a cero los bits .EN, .ST, .EN, .ER, .EW y .NR durante el preescán.

El programa de lógica de escalera debe condicionar el uso de los datos de transferencia en bloques en el estado del bit .DN.

Cómo usar el bloque de control

Además de los bits de estado, el bloque de control contiene otros parámetros que el procesador usa para controlar las instrucciones de transferencia en bloques. La Tabla 15.F indica estos valores.

Tabla 15.F
Valores en el bloque de control de transferencia en bloques

Palabra – Bloque de control de números enteros	Bloque de control BT	Descripción
0	.EN a .RW	Bits de estado
1	.RLEN	Conteo de palabras solicitado
2	.DLEN	Conteo de palabras / código de error transmitido (procesadores PLC-5 con características mejoradas)
3	.FILE	Tipo / número de archivo
4	.ELEM	Número de elemento

Conteo de palabras solicitado (.RLEN)

Este es el número de palabras que se deben transferir entre el procesador y el módulo (0 a 64 palabras). El procesador crea un archivo con la longitud que usted especifica, el cual comienza a partir de la dirección de datos que introduce. La longitud depende del módulo receptor o la aplicación. Por ejemplo, si especifica 30 en este campo, lo que especifica es una longitud de bloque de 30 y el procesador crea un archivo de 30 palabras. Si especifica 64, lo que especifica es una longitud de bloque de 64 y el procesador crea un archivo de 64 palabras. Si especifica 0 cuando introduce la instrucción de transferencia en bloques, el procesador permite que el módulo de transferencia en bloques determine el número de palabras que deben transferirse y crea un archivo predeterminado de 64 palabras.

Conteo de palabras transmitido (.DLEN)

Este es el número de palabras que el módulo transfiere después de que la instrucción concluye la ejecución. El procesador usa este número para verificar la transferencia. Este número debe coincidir con el conteo de palabras solicitado (a menos que el conteo de palabras transmitido sea cero). Si estos números no coinciden, el procesador establece el bit .ER (bit 12).

Los procesadores PLC-5 con características mejoradas también tienen códigos de error (palabra 2 del bloque de control de archivo de números enteros almacenado en la palabra .DLEN del bloque de control BT) que el procesador puede establecer durante la transferencia. Si ocurre un error de transferencia en bloques en un procesador PLC-5 con características mejoradas, el código de error se almacena en el conteo de palabras transmitido. Este error se identifica según su número negativo. Se almacena un solo error a la vez (un nuevo código de error sobrescribe cualquier código de error anterior). La Tabla 15.G indica estos códigos de error.

Tabla 15.G
Códigos de error de transferencia en bloques del procesador PLC-5 con características mejoradas

Número de error:	Descripción:
-1	No se usa
-2	No se usa
-3	El tamaño de la transferencia en bloques más el tamaño del índice en la tabla de datos de la transferencia en bloques fueron mayores que el tamaño del archivo de la tabla de datos de la transferencia en bloques.
-4	Ocurrió una transferencia no válida de datos de transferencia en bloques de escritura entre el módulo adaptador y el módulo de transferencia en bloques.
-5	Fue incorrecta la suma de comprobación de los datos de transferencia en bloques de lectura.
-6	El módulo de transferencia en bloques solicitó una longitud diferente de la instrucción de transferencia en bloques asociada. Esto puede ocurrir si una instrucción de transferencia en bloques de 64 palabras se ejecutó y la longitud predeterminada del módulo de transferencia en bloques no fue 64 palabras.
-7	Se perdieron los datos de transferencia en bloques debido a un canal de comunicación deficiente. Las causas posibles son ruido, conexiones deficientes y cables sueltos. Revise las resistencias.
-8	Error en el protocolo de transferencia en bloques – transferencia en bloques no solicitada.
-9	El tiempo de espera de transferencia en bloques establecido en la instrucción se sobrepasó antes de completarse la transferencia.
-10	No hay canales de comunicación configurados para las E/S remotas o el número de rack no aparece en la lista de racks.
-11	No hay canales de comunicación configurados para el rack o ranura solicitados.
-12	El adaptador tiene un fallo no está presente para el comando BT.
-13	Las colas para las transferencias en bloques remotas están llenas.

Número de archivo (.FILE)

Este número identifica el número de archivo del archivo de números enteros desde el cual se escriben los datos o hacia el cual se leen los datos. Por ejemplo, el número de archivo de N7:20 es 7.

Número de elemento (.ELEM)

Este número identifica la palabra inicial en la dirección de archivo de datos. Por ejemplo, en N7:20 el número de palabra es 20.

Cómo seleccionar la operación continua

La transferencia en bloques continua es similar a la transferencia de E/S porque las E/S se actualizan continuamente, pero la transferencia en bloques continua actualiza las E/S de transferencia en bloques, tales como los datos de entrada analógica y salida analógica.

El modo continuo le permite realizar múltiples transferencias en bloques mediante la programación de una sola instrucción de transferencia en bloques (sin condiciones de entrada en el renglón). Una vez iniciada la transferencia en bloques, la transferencia se ejecuta continuamente una vez por escán, independientemente de que el procesador continúe escaneando el renglón asociado e independientemente de la condición del renglón. Para habilitar la operación continua, seleccione Continuous (continuo) cuando introduzca la instrucción de transferencia en bloques.

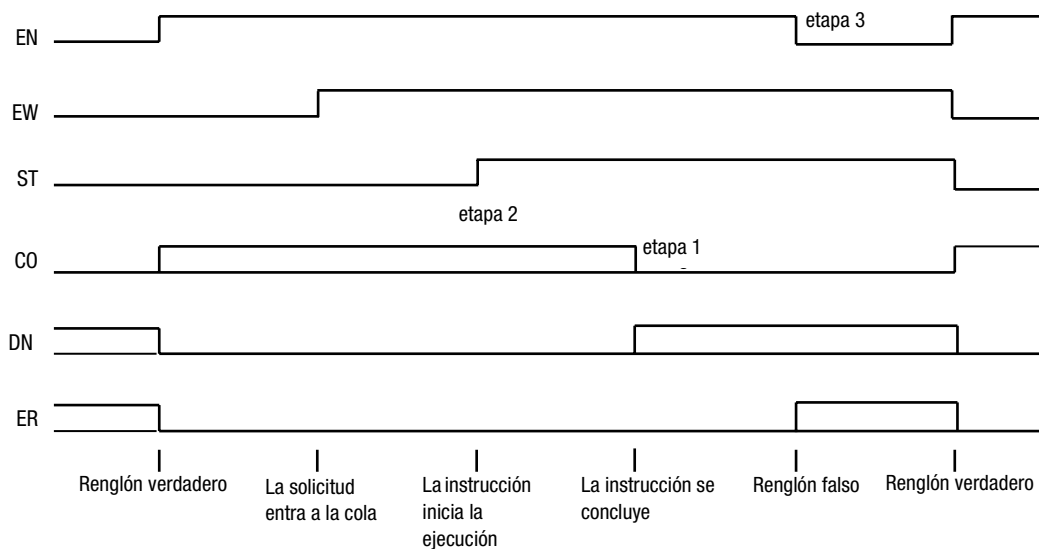
El modo continuo funciona así (Figura 15.3):

1. Cuando el renglón que contiene la instrucción de transferencia en bloques se hace verdadero, el procesador establece el bit .EN. El procesador también restablece los bits .DN, .ER, .ST, .EW y .NR.
2. El procesador pone en la cola la solicitud de transferencia en bloques. Cuando la solicitud de transferencia en bloques entra a la cola, el procesador establece el bit .EW.
3. Cuando el procesador comienza a procesar la solicitud de transferencia en bloques, el procesador establece el bit .ST.
4. Si no ocurre un error durante la transmisión, el procesador establece el bit .DN. El procesador copia el número de los elementos enviados o recibidos por la instrucción de transferencia en bloques en el conteo de palabras transmitido (palabra 2 del bloque de control).

Si ocurre un error, el procesador establece el bit .ER. Si ocurre un error en un procesador PLC-5 con características mejoradas, el procesador también coloca el código de error en la ubicación del conteo de palabras transmitido como número negativo.

5. Si no hay respuesta (y después de que el procesador establezca el bit .NR), el procesador intenta transmitir nuevamente la transferencia en bloques. Si el bit .TO se restablece, el procesador transmite continuamente la solicitud durante cuatro segundos. Si el bit .TO se establece, el procesador vuelve a intentar a transmitir la solicitud una vez solamente.
6. Si una transferencia en bloques continua tiene un error, se debe volver a iniciar para continuar. (Vea la Figura 15.7 en la página 15-18 para obtener un ejemplo de programa.)

Figura 15.3
Diagrama de temporización para los bits de estado en instrucciones
continuas BTR y BTW



Etapa 1 – Si .CO está establecido, retorne a la etapa 2; si .CO está restablecido, vaya a la etapa 3

Etapa 2 – Retorne acá para la operación continua

Etapa 3 – Vaya a esta etapa si .CO está restablecido

Una transferencia en bloques continua continúa siempre que el procesador permanezca en el modo marcha o prueba y la transferencia no tenga un error. Si cambia al modo de programación o si el procesador tiene un fallo, la transferencia en bloques se detiene y no volverá a iniciarse hasta que el procesador escanee el renglón que contiene la instrucción de transferencia en bloques. Vea el Apéndice B si se ejecutan transferencias en bloques continuas desde los diagramas de función secuenciales.

Para detener la operación continua, modifique la instrucción de transferencia en bloques y seleccione no continuo o restablezca el bit .CO.

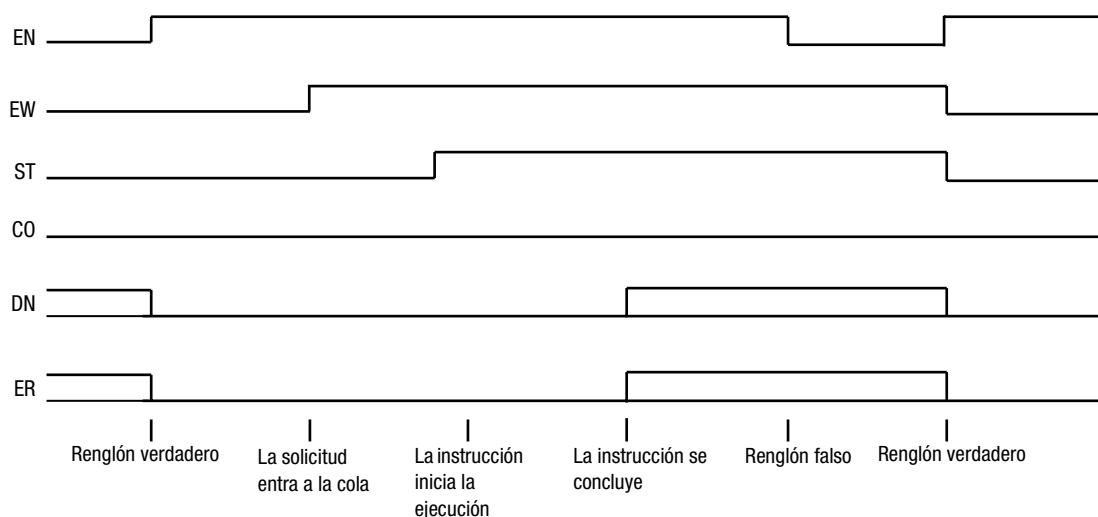
Cómo seleccionar la operación no continua

La transferencia en bloques no continua actualiza las E/S de transferencia en bloques una vez cuando el renglón se hace verdadero. Una transferencia en bloques no continua mantiene la integridad del bloque. Se actualiza todo el bloque de datos cada vez que el procesador ejecuta la instrucción de transferencia en bloques. Use el modo no continuo cuando desee controlar cuándo ocurre la transferencia en bloques o cuántas veces ocurre la transferencia en bloques.

El modo no continuo funciona así (Figura 15.4):

1. Cuando el renglón que contiene la instrucción de transferencia en bloques se hace verdadero, el procesador establece el bit .EN. El procesador también restablece los bits .DN, .ER, .ST, .EW y .NR.
2. El procesador pone en la cola la solicitud de transferencia en bloques. Cuando la solicitud de transferencia en bloques entra a la cola, el procesador establece el bit .EW.
3. Cuando el procesador comienza a procesar la solicitud de transferencia en bloques, el procesador establece el bit .ST.
4. Si no ocurre un error durante la transmisión, el procesador establece el bit .DN después de que la instrucción de transferencia en bloques se complete. Si ocurre un error, el procesador establece el bit .ER.
5. Esto significa que se completó una transferencia en bloques. La próxima vez que el renglón se hace falso, el procesador restablece el bit .EN.

Figura 15.4
Diagrama de temporización para los bits de estado en las instrucciones no continuas BTR y BTW



Temporización de transferencias en bloques – Procesadores PLC-5 clásicos

El tiempo necesario para completar una transferencia en bloques en un procesador PLC-5 clásico depende de:

- el tiempo de ejecución de la instrucción
- el tiempo de espera en la cola
- el tiempo de transferencia

Tiempo de ejecución de la instrucción

El tiempo en microsegundos que necesita el procesador para ejecutar una instrucción de transferencia en bloques se define según estas fórmulas:

Escritura:

$$310 + 11.2Q + 5.4W$$

Lectura:

$$250 + 11.2Q$$

Donde:

Representa:

Q

el número de solicitudes de transferencia en bloques al mismo chasis de E/S con el conjunto de bits continuos

W

el número de palabras que se deben transferir

Tiempo de espera en la cola

El tiempo de espera en la cola es la suma de los tiempos de transferencia por ocurrir antes de la solicitud de transferencia en bloques (para la cual se calcula el tiempo) al mismo chasis de E/S.

Tiempo de transferencia

El tiempo de transferencia en milisegundos entre el búfer activo y el módulo comienza cuando el procesador establece el bit de inicio y se termina cuando el procesador establece el bit de efectuado. El tiempo de transferencia se define según estas fórmulas:

Escritura:

$$\text{local} \quad 0.9 + 0.1W$$

$$\text{remoto (57.6 K baud)} \quad 13 + 30C + 0.3W$$

Lectura:

$$\text{local} \quad 0.9 + 0.1W$$

$$\text{remoto (57.6 K baud)} \quad 9 + 21.3C + 0.3W$$

Donde:

Representa:

C

el número de racks lógicos remotos completos

W

el número de palabras que se deben transferir

Temporización de transferencias en bloques – Procesadores PLC-5 con características mejoradas

El tiempo necesario para completar una transferencia en bloques en los procesadores PLC-5 con características mejoradas depende de:

- el tiempo de ejecución de la instrucción
- el tiempo de espera en la zona de espera (cola)
- el tiempo de transferencia

Tiempo de ejecución de la instrucción

El tiempo que el procesador necesita para ejecutar una instrucción de transferencia en bloques es el tiempo para una lectura o escritura: 450 microsegundos.

Tiempo de espera en la zona de espera

El tiempo de espera en la zona de espera es la suma de los tiempos de transferencia por ocurrir antes de la solicitud de transferencia en bloques (para la cual se calcula el tiempo) al mismo chasis de E/S.

Tiempo de transferencia

El tiempo de transferencia en milisegundos entre el búfer activo y el módulo comienza cuando el procesador establece el bit de inicio y se termina cuando el procesador establece el bit de efectuado. El tiempo de transferencia se define según esta fórmula (se usa la misma fórmula para la lectura o escritura):

local $600 \mu\text{seg} + x(w)$

remoto (57.6 K baudios) $4 + 8C + 0.3W$

remoto (115 K baudios) $4 + 4.6C + 0.15W$

remoto (230 K baudios) $4 + 3.2C + 0.075W$

Donde: **Representa:**

x	• 8 ó menos transferencias en bloques en la cola en el rack local = 86 μseg • más de 8 transferencias en bloques en la cola en el rack local = 300 μseg Nota: esta temporización supone que no hay otras transferencias en bloques en la cola a la misma ranura y que las transferencias en bloques sucesivas a la misma ranura se ejecutan cada 1000 μseg.
C	el número de racks lógicos remotos completos
W	el número de palabras que se deben transferir

Ejemplos de programación

Programa el procesador para la transferencia en bloques con el uso de uno de los siguientes métodos según los requisitos de la aplicación (Tabla 15.H):

Tabla 15.H
Métodos de programación de transferencia en bloques

Si desea:	Use este método:
Programar las transferencias en bloques hacia y desde el mismo módulo cuando desea que el orden de ejecución siga el mismo orden escaneado al programa	Alternante bidireccional
Repetir continuamente las transferencias en bloques alternantes bidireccionales y se escaneará el paso	Repetición alternante bidireccional
Programar transferencias en bloques hacia y desde el mismo módulo cuando desea que las transferencias continúen independientemente de cuáles pasos SFC están activos	Continuo bidireccional*
Programar una BTR desde o un BTW hacia un módulo cuando desea que la transferencia en bloques se ejecuten en base a un evento	Direccional no continuo
Repetir continuamente una transferencia en bloques y se escaneará el paso	Repetición direccional
Programar una BTR desde o una BTW hacia un módulo cuando desea que la transferencia continúe independientemente de cuáles pasos SFC están activos	Continuo direccional*
Asegurar la integridad del bloque	Almacenamiento de datos de transferencia en bloques en el búfer
* Use el modo continuo solamente cuando desee que una transferencia en bloques continúe ejecutándose aun cuando no se escanee la lógica que controla la transferencia.	

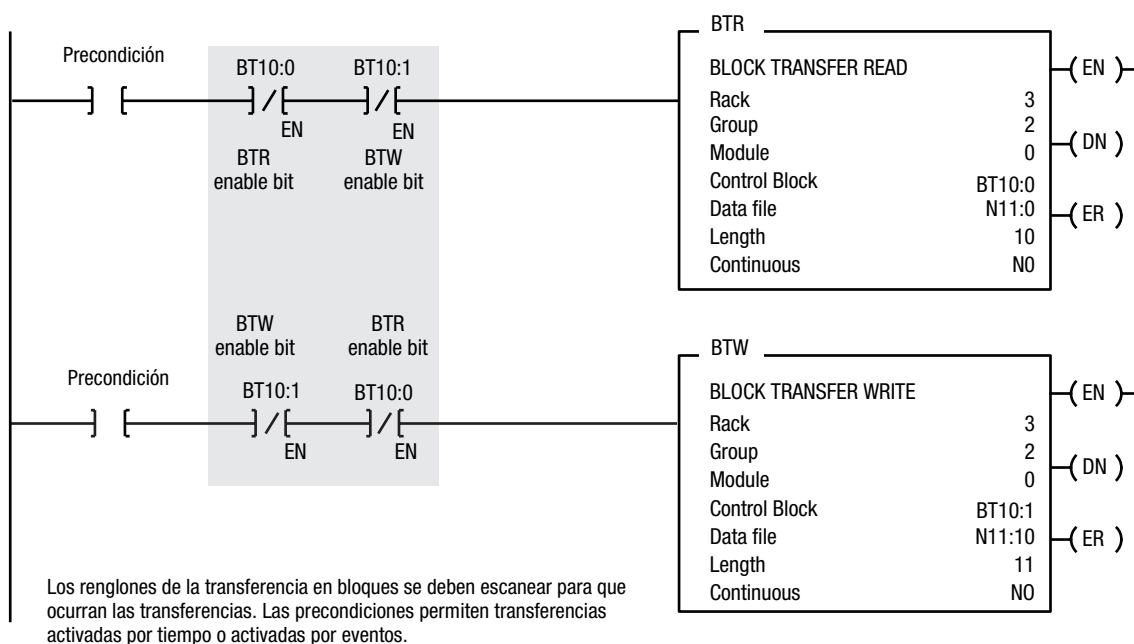
Importante: Estos ejemplos muestran un procesador PLC-5 con características mejoradas usando el tipo de archivo BT. Si usa un procesador PLC-5 clásico, sustituya un archivo de números enteros apropiado.

Ejemplo de transferencia en bloques alternante bidireccional

La Figura 15.5 muestra un ejemplo de una transferencia en bloques alternante bidireccional. El uso de renglones como los de este ejemplo asegura que las solicitudes de transferencia en bloques se ejecuten en el orden en que se enviaron a la cola. El procesador alterna entre las BTR y las BTW en el orden en que se escanearon según la condición XIO. La condición XIO impide que la transferencia en bloques de lectura y la transferencia en bloques de escritura se pongan en la cola simultáneamente. La transferencia en bloques continúa siempre que la condición de renglón sea verdadera.

En los renglones de lógica, puede incluir tantas condiciones opcionales como desee a la izquierda de la transición de condición del bit de habilitación requerida (XIO).

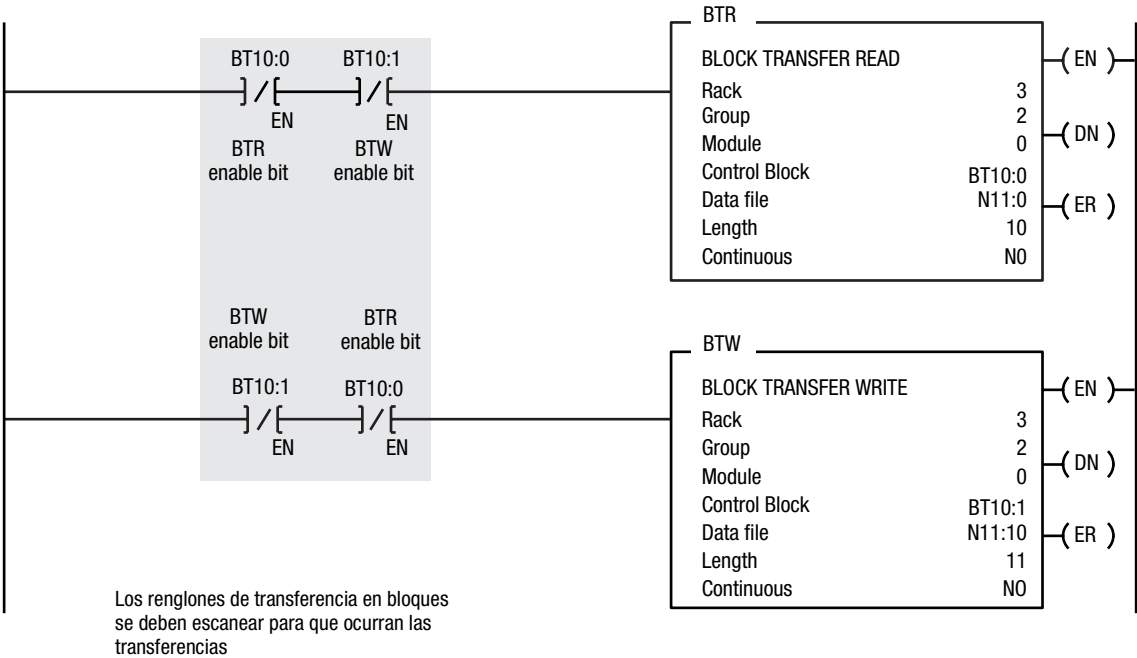
Figura 15.5
Ejemplo de transferencia en bloques alternante bidireccional



Ejemplo de transferencia en bloques de repetición alternante bidireccional

La Figura 15.6 muestra un ejemplo de una transferencia en bloques de reptición alternante bidireccional. El uso de renglones según este ejemplo asegura que las solicitudes de transferencia en bloques se ejecuten en el orden en que se enviaron a la cola. El procesaor alterna entre las BTR y las BTW en el orden en que se escanearon según la condición XIO. La condición XIO impide que la transferencia en bloques de lectura y la transferencia en bloques de escritura se pongan en la cola simultáneamente. La transferencia en bloques continúa siempre que la condición de renglón sea verdadera.

Figura 15.6
Ejemplo de transferencia en bloques de repetición alternante bidireccional

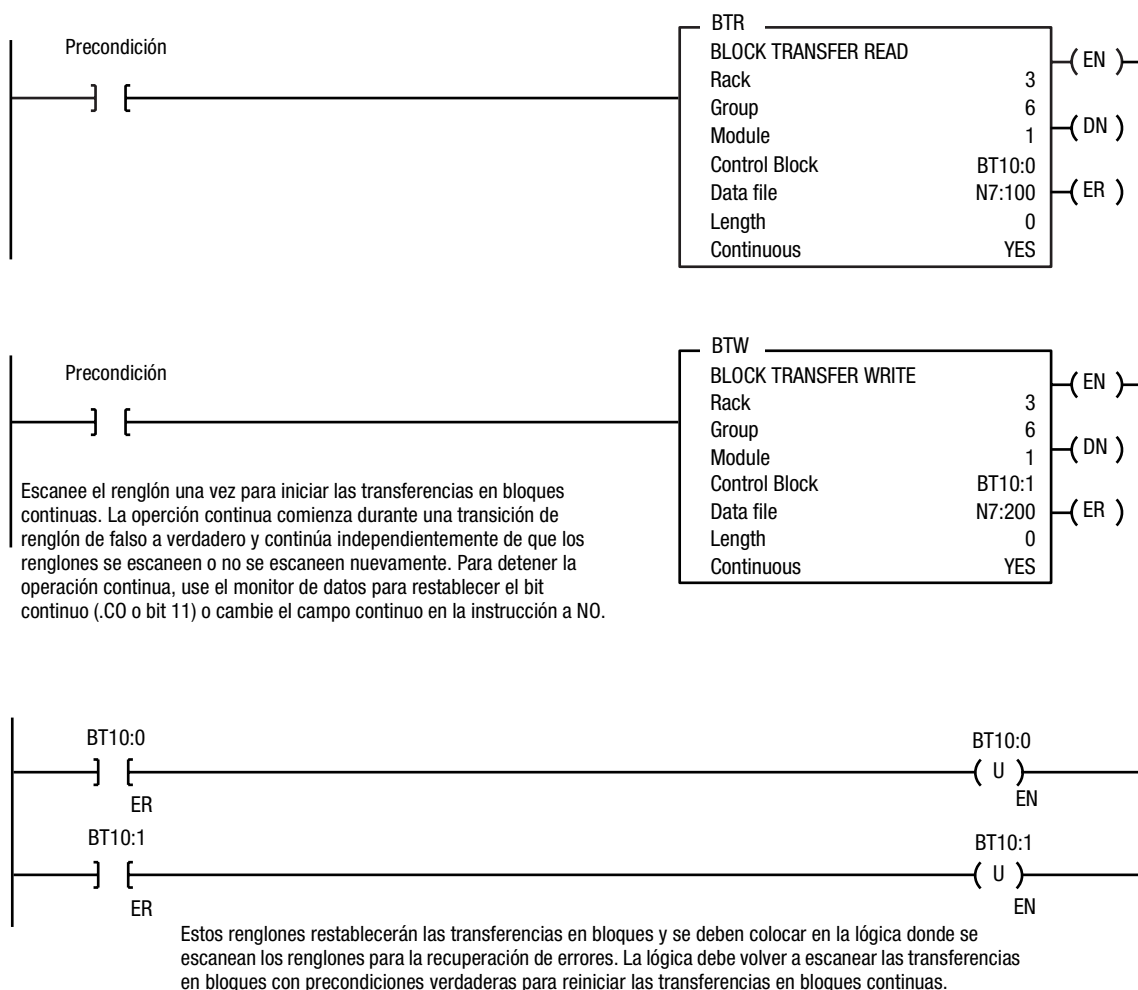


Ejemplo de transferencia en bloques continua bidireccional

La Figura 15.7 muestra un ejemplo de una transferencia en bloques continua bidireccional.

Figura 15.7

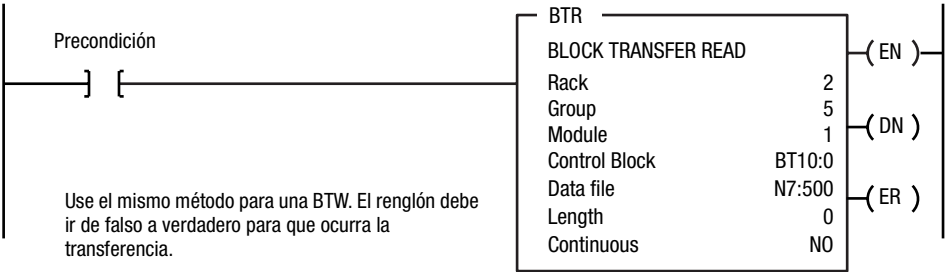
Ejemplo de transferencia en bloques continua bidireccional



Ejemplo de transferencia en bloques no continua bidireccional

La Figura 15.8 muestra un ejemplo de una transferencia en bloques no continua bidireccional. La transferencia en bloques se ejecuta una vez para cada transición de falso a verdadero de la precondition.

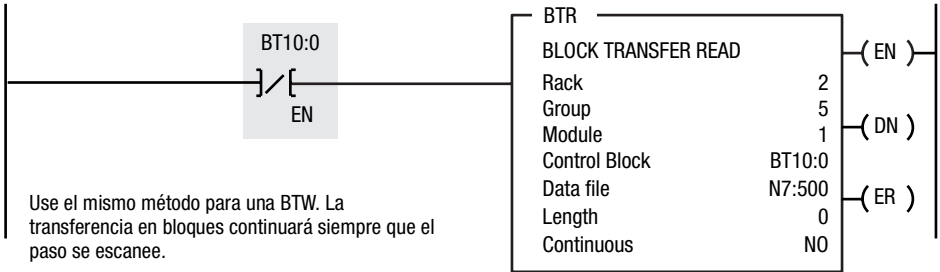
Figura 15.8
Ejemplo de transferencia en bloques no continua bidireccional



Ejemplo de transferencia en bloques de repetición direccional

La Figura 15.9 muestra un ejemplo de una transferencia en bloques de repetición direccional.

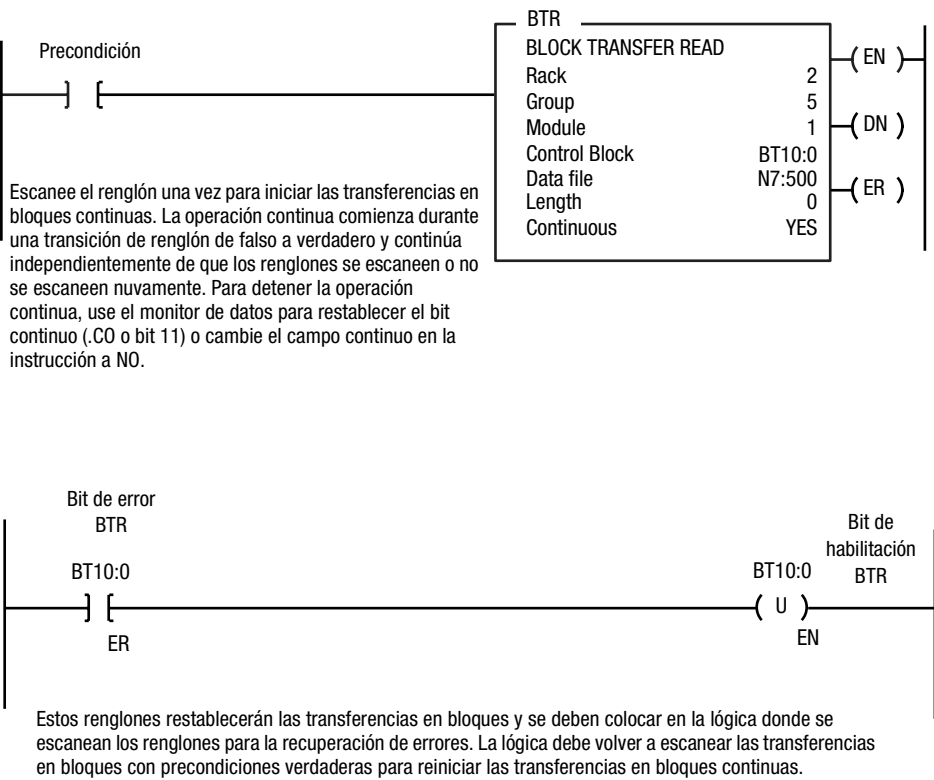
Figura 15.9
Ejemplo de transferencia en bloques de repetición direccional



Ejemplo de transferencia en bloques continua bidireccional

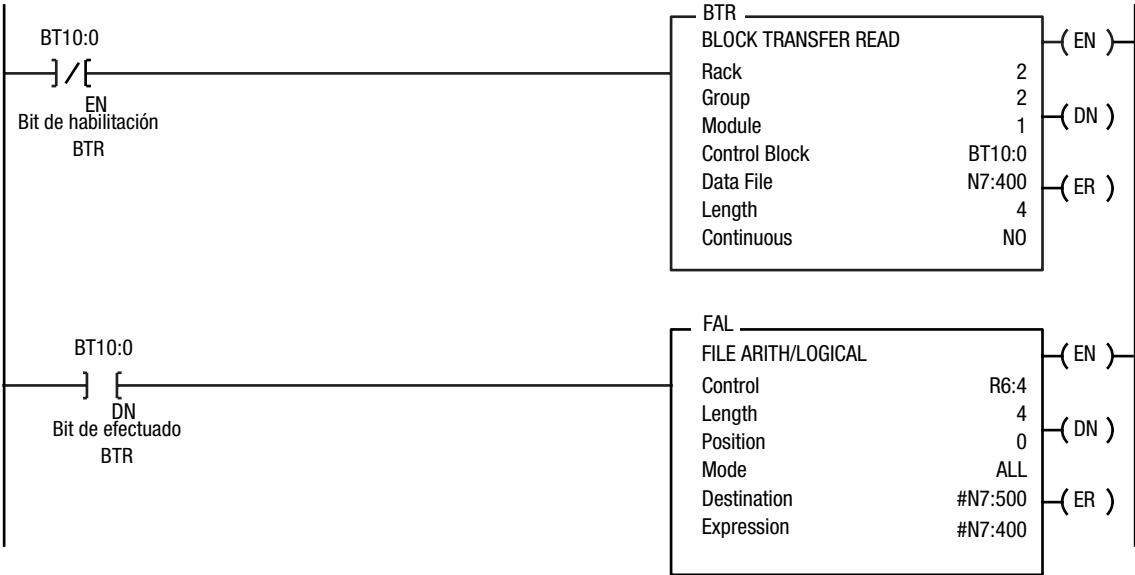
La Figura 15.10 muestra un ejemplo de una transferencia en bloques continua bidireccional.

Figura 15.10
Ejemplo de transferencia en bloques continua bidireccional

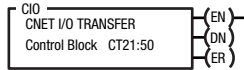


Ejemplo de búfer de datos de transferencia en bloques

Si transfiere datos en bloques repetidamente, almacene en un búfer el archivo examinando el bit de efectuado BTR y ejecutando un movimiento (o copia) de archivo a archivo cuando el bit de efectuado es verdadero. Esto asegura la integridad del archivo de datos de la transferencia en bloques de lectura.



Instrucción de transferencia de E/S ControlNet (CIO)



El uso de la instrucción CIO le permite realizar transferencias no programadas iniciadas por la lógica de escalera (hasta 64 elementos) a módulos de E/S (típicamente analógicos o inteligentes) en una red ControlNet. Vea el Manual del usuario de los controladores programables PLC-5 ControlNet para obtener más información acerca de las operaciones de E/S ControlNet.

Cuando las condiciones de entrada cambian de falso a verdadero, los datos se transfieren según los parámetros de instrucción establecidos cuando se introdujo la instrucción CIO.

Para programar la instrucción CIO, debe proporcionar al procesador PLC-5 ControlNet una dirección de bloque de control, la cual contiene los parámetros de estado e instrucción. Después de que usted introduce los parámetros del bloque de control, el terminal muestra una pantalla para la introducción de instrucciones mediante la cual introduce los parámetros de instrucción almacenados en la dirección del bloque de control.

Dirección del bloque de control

Con los procesadores PLC-5 ControlNet, use un tipo de archivo de transferencia ControlNet (CT) para el bloque de control. Por ejemplo, CT12:1 es una dirección de bloque válida CIO.

Importante: No puede usar direcciones indirectas para las dirección del bloque de control en una instrucción CIO.

Después de que introduzca la dirección del bloque de control para la instrucción CIO, el terminal de programación muestra una pantalla para la introducción de instrucciones.

Cómo usar la instrucción CIO

Puede usar la instrucción CIO para transferir hasta 64 elementos de datos (por instrucción CIO) mediante de una red ControlNet. La pantalla para la introducción de instrucciones CIO le permite configurar la información siguiente (Tabla 15.I).

Importante: El software de programación de texto estructurado PLC-5 no es compatible con la instrucción CIO.

Tabla 15.I
Configuración de la pantalla para la introducción de instrucciones CIO

Si desea:	Presione esta tecla:
Cambiar el tipo de comando. Alterne entre lo siguiente: • La lectura 1771 selecciona una transferencia en bloques de lectura. • La escritura 1771 selecciona una transferencia en bloques de escritura. • La acción de fallo 1794 selecciona la acción que el módulo realiza cuando el adaptador tiene un fallo y la conexión se termina. • La acción de inactividad 1794 selecciona la acción que el módulo toma cuando la conexión está en inactividad. • Los datos de configuración 1794 cambian la configuración para el módulo 1794. • Los datos de estado de seguridad 1794 cambian el valor de los datos de estado de seguridad para el módulo 1794.	[F1] – Command Type
Introducir una dirección de la tabla de datos PLC-5 del procesador ControlNet	[F2] – PLC-5 Address
Introducir el tamaño en elementos. Escriba el número de elementos y presione [Enter]. • 1 (acción de fallo 1794 y acción de inactividad 1794) • 1-15 (datos de configuración 1794 y datos de estado de seguridad 1794) • 0-64 (lectura 1771 y escritura 1771) Nota: si introduce 0 para la lectura 1771 y escritura 1771, hay 64 palabras reservadas para la transferencia en bloques.	[F3] – Size in Elements
Introduzca la dirección de red de destino. Escriba un número (1-99) y presione [Enter].	[F8] – Local Node
Introduzca un número de ranura de destino. Escriba un número y presione [Enter]. • 0-7 (tipos de comando 1794) • 0-15 (tipos de comando 1771) Nota: el número de ranura representa la ranura física en el chasis ocupado por el módulo. Para encontrar el número de ranura, cuente desde la ranura de E/S izquierda a partir de 0.	[F9] – Slot Number

Después de introducir y aceptar el renglón que contiene la instrucción CIO, la pantalla del monitor de datos para la instrucción CIO le permite mostrar los parámetros para el bloque de control de la instrucción CIO actual. La pantalla del monitor de datos le permite definir los siguientes parámetros (Tabla 15.J).

Tabla 15.J
Parámetros del bloque de control de la instrucción CIO

Si desea:	Presione esta tecla:
Alternar el bit de control en el cual se encuentra el cursor. Puede alternar entre los bits TO, EW, CO, ER, DN, ST y EN.	[F2] – Toggle Bit
Cambiar el tamaño del bloque de datos que se debe enviar o recibir	[F3] – Size in Elements
Cambiar la dirección para la cual se muestran los datos	[F5] – Specify Address
Mostrar en pantalla los valores de la tabla de datos para el próximo archivo.	[F7] – Next File
Mostrar en pantalla los valores de la tabla de datos para el archivo anterior.	[F8] – Previous File
Mostrar en pantalla los valores de la tabla de datos para el próximo elemento.	[F9] – Next Element
Mostrar en pantalla los valores de la tabla de datos para el elemento anterior.	[F10] – Previous Element

Cómo usar los bits de estado

La instrucción CIO usa los siguientes bits de estado:

Este bit:	Se establece:
Habilitación .EN (bit 15)	cuando el renglón se hace verdadero. El bit .EN se restablece cuando el bit .DN o el bit .ER se establece. Este bit indica que la instrucción está habilitada.
Inicio .ST (bit 14)	cuando el procesador comienza a ejecutar la instrucción CIO. El bit .ST se restablece cuando el bit .DN o el bit .ER se establece.
Efectuado .DN (bit 13)	cuando se transfirió la última palabra de la instrucción CIO. El bit .DN se restablece la próxima vez que el renglón asociado va de falso a verdadero. El bit .DN está activo solamente en el modo no continuo.
Error .ER (bit 12)	cuando el procesador detecta que falló la transferencia de mensaje. El bit .ER se restablece la próxima vez que el renglón asociado cambia de falso a verdadero.
Continuo .CO (bit 11)	manualmente para la operación repetida de la instrucción CIO después del primer escán independientemente de que el procesador continúe escaneando el renglón.
Habilitado-esperando .EW (bit 10)	cuando el procesador detecta que una solicitud de mensaje entró en la cola. El procesador restablece el bit .EW cuando se establece el bit .ST.
Tiempo de espera .TO (bit 08)	mediante la lógica de escalera para detener el procesamiento del mensaje. El procesador establece el bit .ER.



ATENCION: El procesador controla los bits .ST y .EW asincrónicamente con el escán de programa. Si examina estos bits en la lógica de escalera, copie el estado a un bit de almacenamiento cuyo estado está sincronizado con el escán de programa. De lo contrario, los problemas de temporización pueden anular el programa, con posibles daños al equipo o lesiones corporales.



ATENCION: Para que el modo continuo funcione correctamente, debe establecer el bit .CO (en la pantalla de configuración y mediante la lógica de escalera) **antes** de habilitar la instrucción CIO.

Cómo usar el bloque de control CT

Además de los bits de estado, el bloque de control CT contiene estos parámetros que usa el procesador PLC-5 ControlNet para controlar las instrucciones CIO.

Palabra:	Bloque de control CT:	Descripción:
0	.EN a .TO	Bits de estado Vea “Cómo usar los bits de estado”.
1	.ERR	Código de error El procesador almacena el código de error aquí si ocurre un problema durante la transmisión del mensaje.
2	.RLEN	Longitud solicitada Este es el número de elementos solicitado que desea transferir con la instrucción de mensaje.
3	.DLEN	Longitud de efectuado Este es el número de elementos que el módulo realmente transfirió después de que la instrucción concluye la ejecución. Este número debe coincidir con la longitud solicitada (a menos que la longitud solicitada sea 0).
4	.FILE	Número de archivo Este número identifica el número del archivo desde el cual se escriben los datos o hacia el cual se leen los datos. Por ejemplo, el número de archivo N12:1 es 12.
5	.ELEM	Número de elemento Este número identifica la palabra inicial en la dirección del archivo de datos. Por ejemplo, en N12:1, el número de palabra es 1.

Notas:

Instrucción de mensaje MSG

Cómo usar la instrucción de mensaje

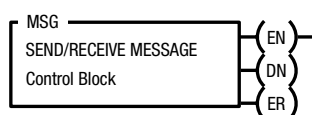
Esta instrucción de mensaje (MSG) se usa para leer o escribir un bloque de datos a otra estación en la red DH+, a un coprocesador de control conectado, al VMEbus que usa un procesador PLC-5 VME o a otro nodo en una red Ethernet. La instrucción MSG también se usa para crear mensajes no programados iniciados por un procesador PLC-5 ControlNet y enviadas a otro procesador PLC-5 ControlNet y para permitir que los procesadores PLC-5 con características mejoradas (que no sean PLC-5 Ethernet) programen mensajes no solicitados de carga/descarga mediante Ethernet por el módulo de interface Ethernet PLC-5. La instrucción MSG se programa en la lógica de escalera.

La instrucción MSG mediante DH+ puede comunicarse con los procesadores PLC-2[®], PLC-3[®], PLC-5[®], PLC-5/250[™], SLC 5/03[™] y SLC-5/04[™] en redes locales o remotas.

Vea el Apéndice C para obtener más información acerca de los operandos (y los tipos/valores de datos válidos de cada operando) usados por la instrucción MSG.

Message (MSG)

Descripción:



La instrucción MSG transfiere hasta 1000 elementos de datos (120 palabras usando un coprocesador de control). El tamaño de cada elemento depende de la sección de la tabla de datos que usted especifica y el tipo de comando de mensaje que usa. Por ejemplo, un elemento binario contiene una palabra de 16 bits y un elemento de punto flotante (coma flotante) contiene dos palabras de 16 bits.

La instrucción MSG transfiere datos en paquetes. Cada paquete de datos DH+ puede contener hasta 120 palabras. Si la transferencia de mensaje contiene demasiadas palabras para el paquete, la transferencia requiere más de un paquete de datos de transferencia. Cuantos más datos hay para la transferencia, tanto más tiempo se requiere para realizar la transferencia. Cada paquete puede contener hasta 709 palabras a través de la red Ethernet. Por lo tanto, ésta es una opción de conexión en red más eficaz.

La tabla siguiente indica cuáles procesadores PLC-5 con características mejoradas (serie o revisión) puede usar con la instrucción MSG para transferir datos desde/hacia un procesador o hacia/desde un procesador SLC 5/04 ó 5/04 en el modo nativo SLC.

Serie/revisión del procesador	Procesadores:
Serie A / revisión M	PLC-5/40, -5/40L, -5/60, -5/60L
Serie A / revisión J	PLC-5/30
Serie A / revisión H	PLC-5/11, -5/20
Serie B / revisión J	PLC-5/40, -5/40L, -5/60, -5/60L
Serie C / revisión G	Procesadores con características mejoradas, Ethernet y PLC-5 VME
Serie C / revisión H	PLC-5 ControlNet
Serie D / revisión A	Procesadores con características mejoradas, Ethernet, ControlNet y PLC-5 VME

Cómo introducir los parámetros

Especifique una dirección de bloque de control cuando introduzca la instrucción MSG por primera vez. El bloque de control almacenará toda la información relacionada con el mensaje. Después de que se introduzca el bloque de control, el terminal de programación muestra automáticamente una pantalla para la introducción de datos mediante la cual se introducen parámetros de instrucción almacenados en la dirección del bloque de control. También puede usar la pantalla del monitor para la instrucción MSG a fin de editar parámetros seleccionados.

Dirección del bloque de control

Use un archivo de números enteros (N) con los procesadores PLC-5 clásicos sin el símbolo # para el bloque de control de mensaje. Por ejemplo, N7:0 es una dirección de bloque de control MSG válida.

Si tiene este procesador:	Use esta dirección de bloque de control:
PLC-5 clásico	un archivo de números enteros (N) sin el símbolo # para el bloque de control de mensaje. Ejemplo: N7:0
PLC-5 con características mejoradas, PLC-5 Ethernet o PLC-5 VME	un archivo de números enteros (N) o el tipo de archivo de mensaje (MG) para obtener acceso al bloque de control de mensaje para transferencias DH+. Ejemplo: MG10:0 Cuando se usa el bloque de control MG, el tamaño del bloque está fijo en 56 palabras. Este tamaño se muestra en la pantalla en el campo BLOCK SIZE (tamaño del bloque). Debe usar el bloque de control MG si envía mensajes a un procesador SLC 500 que usa los comandos de lectura y escritura SLC o si envía mensajes por cualquier puerto que no sea el canal 1A.
PLC-5 Ethernet, PLC-5 ControlNet, PLC-5 VME	un tipo de archivo de mensaje (MG) para obtener acceso a la red VMEbus, Ethernet o ControlNet

No puede usar direcciones indirectas para la dirección del bloque de control en una instrucción MSG. Si tiene una instrucción MSG creada con la versión 3.21 ó anteriores que usa un bloque de control con una dirección indirecta, debe eliminar la instrucción y volver a introducirla sin una dirección indirecta.

Para que los procesadores PLC-5 VME realicen transferencias al VMEbus, es necesario programar la instrucción MSG con un bloque de control MG.

Para que los procesadores PLC-5 ControlNet realicen transferencias en la red ControlNet, es necesario programar la instrucción MSG con un tipo de datos MG en el bloque de control.

El tamaño del bloque de control varía según la longitud del mensaje. Si se comunica con un procesador PLC-2, el archivo de control tendrá una longitud aproximada de 11 ó 12 palabras. Si se comunica con un procesador PLC-3, PLC-5 ó PLC-5/250, el archivo de control tendrá una longitud aproximada de 11 a 15 palabras.

Puede usar un archivo de números enteros (excepto los procesadores PLC-5 ControlNet) o un tipo de archivo de mensaje (MG) para los procesadores PLC-5 con características mejoradas a fin de obtener acceso al bloque de control de mensaje para las transferencias DH+. Por ejemplo, MG10:0 es una dirección de bloque de control MSG válida para los procesadores PLC-5 con características mejoradas. El uso del tipo de archivo MG fija el tamaño del bloque de control en 56 palabras. Este tamaño se muestra en pantalla en el campo BLOCK SIZE.

Cuando se trata de los procesadores PLC-5 Ethernet, la instrucción MSG que pasa por el puerto 2, el puerto Ethernet, usa dos elementos de mensaje consecutivos (es decir, MG10:0 y MG10:1). El software de programación puede mostrar en pantalla una advertencia cuando selecciona el puerto 2.

Pantalla para la introducción de datos MSG

Después de que usted introduzca la dirección del bloque de control para una instrucción MSG, el software de programación muestra automáticamente una pantalla para la introducción de datos para la instrucción MSG con el uso del tipo de datos apropiado (números enteros o mensaje). Presione las teclas de función para los datos que desea modificar. Puede especificar los siguientes parámetros MSG mediante la pantalla de introducción:

Esta tecla de función:	Especifica esta información:
[F1] – Command Type	Si la instrucción MSG realiza una operación de lectura o escritura y a qué tipo de procesador se envía el mensaje
[F2] – PLC-5 Address	La dirección del archivo de datos del procesador que contiene la instrucción de mensaje. Si la operación MSG es de escritura, esta dirección es la palabra inicial del archivo de origen. Si la operación MSG es de lectura, esta dirección es la palabra inicial del archivo de destino.
[F3] – Size in Elements	El número de elementos (1-1000) que se deben transferir.

Esta tecla de función:	Especifica esta información:
[F4] – Local/Remote	LOCAL: el mensaje se envía a un dispositivo en la red DH+ local. REMOTE: el mensaje se envía por un puente (DH, DH II, etc.) a otra red DH+. Si selecciona REMOTO, están activas las teclas de función [F5] – Remote Station, [F6] – Link ID y [F7] – Remote Link.
[F5] – Remote Station	La dirección DH o DH II (1-376 octal) de la estación receptora. Los procesadores PLC-2 y PLC-3 requiere módulos adaptadores de comunicación (1771-KA2 y 1775-KA, respectivamente) cuando operan como estaciones en DataHighway. En estas configuraciones, la dirección de estación remota es la dirección del módulo adaptador de comunicaciones.
[F6] – Link ID	La red remota donde reside el procesador con el cual usted desea comunicarse. El valor predeterminado es 0.
[F7] – Remote Network	Alterna entre DH, DH II y otras selecciones para encontrar lo que conecta la red remota a la DH+ local.
[F8] – Local Node	La dirección de estación local en la DH+ (0-77) local. Si se comunica con otro procesador en la red local, esta dirección es la dirección de la estación receptora en la red local. Si se comunica con otra estación receptora en una red remota, esta dirección es el número de estación del módulo adaptador de comunicación que conecta DataHighway.
[F9] – Destination Address	La dirección inicial del archivo de origen o destino en el lprocesador receptor.
[F10] – Port Number	El canal para las comunicaciones de mensaje. Las opciones válidas con 0, 1A (predeterminado), 1B, 2A, 2B y 3A para el comando ASCII.

Si selecciona la opción ASCII usando la tecla **[F1] – Command Type**, (para uso con el PLC-5/V40 que realiza las lecturas/escrituras al VMEbus), el software muestra una nueva pantalla mediante la cual se introduce el texto para las comunicaciones ASCII. Vea el Manual del usuario de los controladores programables VMEbus PLC-5/V40 para obtener la sintaxis de texto de comando necesaria para realizar transferencias VMEbus.

Use las selecciones siguientes para las transferencias de datos del coprocesador de control usando la instrucción MSG:

- comando de comunicación – seleccione un rango de palabra PLC-3 de lectura o un rango de palabra PLC-3 de escritura
- dirección de destino de la tabla de datos – 00 a 31; coincide con el administrador de memoria de lectura/escritura correspondiente en el programa de aplicación del coprocesador
- Número de puerto – 3A

Cómo usar la instrucción de mensaje para las comunicaciones Ethernet

La instrucción de mensaje (MSG) transfiere hasta 1000 elementos de datos. El tamaño de cada elemento depende de la sección de la tabla de datos que usted especifica y el tipo de comando de mensaje que usa. Por ejemplo, un elemento binario contiene una palabra de 16 bits y un elemento de punto flotante (coma flotante) contiene dos palabras de 16 bits.

La instrucción MSG transfiere datos en paquetes. Cada paquete puede contener hasta 709 palabras para los procesadores Ethernet. Si la transferencia de mensaje contiene demasiadas palabras para el paquete, la transferencia requiere más de un paquete de datos de transferencia. Cuantos más datos hay para la transferencia, tanto más tiempo es necesario para realizar la transferencia.

Cómo introducir los parámetros

El bloque de control almacena toda la información relacionada con el mensaje. Las instrucciones de mensaje Ethernet usan dos elementos MSG consecutivos. El primero contiene información del mensaje y el segundo contiene la dirección de destino.

Importante: Puesto que los mensajes Ethernet necesita dos bloques de control consecutivos, el bloque de control de mensaje que usted especifique debe comenzar en un número par.



ATENCIÓN: Durante la configuración de las instrucciones MSG para las redes DH+ y en serie, tome en cuenta los archivos usados para los bloques de control MSG Ethernet.

Si selecciona un archivo que se usa como bloque de control Ethernet, el software de programación le invita a sobrescribir el archivo. Si selecciona sobrescribir el archivo, puede ocurrir una operación inesperada de la máquina.

Después de introducirse el bloque de control, el terminal de programación muestra automáticamente una pantalla para la introducción de datos mediante la cual se introducen los parámetros de instrucción que se almacenan en la dirección del bloque de control.

Debe introducir un número de puerto de 2 para habilitar una pantalla especial para las transferencias Ethernet.

Este campo:	Especifica esta información:
Command Type	Si la instrucción MSG realiza una operación de lectura o escritura. El software alterna entre: • PLC-5 Typed Read • PLC-5 Typed Write • PLC-5 Typed Write to SLC • PLC-5 Typed Read from SLC • SLC Typed Logical Read • SLC Typed Logical Write • PLC-2 Unprotected Read • PLC-2 Unprotected Write • PLC-3 Word Range Read • PLC-3 Word Range Write • ASCII
PLC-5 Address	La dirección del archivo de datos del procesador que contiene la instrucción de mensaje. Si la operación MSG es de escritura, esta dirección es la palabra inicial del archivo de origen. Si la operación MSG es de lectura, esta dirección es la palabra inicial del archivo de destino.
Size in Elements	El número de elementos (1-1000) que se deben transferir.
IP Address	El nodo de destino de la instrucción MSG. • Si el destino es otro PLC-5/20E, -5/40E o -5/80E, el destino debe ser una dirección Internet completa. • Si el destino es un programa de cliente INTERCHANGE™, introduzca la palabra "CLIENT" (cliente) como el nodo de destino. No introduzca una dirección IP. Nota: debe establecer [F10] – port number a 2 para obtener acceso a esta función.
Destination Address	La dirección inicial del archivo de origen o destino en el procesador receptor.
Port Number	El canal para las comunicaciones de mensaje. Las comunicaciones Ethernet usan el canal 2.
Multihop	Seleccione YES si desea enviar la instrucción MSG a un dispositivo ControlLogix. Use la ficha Multihop (multisaltos) para especificar la ruta de acceso de la instrucción MSG. Vea "Cómo configurar una instrucción MSG de multisaltos Ethernet" en la página 16-9.

Los procesadores PLC-5 Ethernet no son compatibles con **nombres de computadora principal** como medio para direccionar mensajes. Sin embargo, puede usar los nombres de computadora principal con el software de programación PLC-5 para conectarse a los procesadores PLC-5 Ethernet si un servidor de nombre se encuentra en la red o si un archivo de la computadora principal se mantiene en su estación de trabajo.

Cómo usar la instrucción de mensaje para las comunicaciones del módulo de interface PLC-5 Ethernet

Use la instrucción MSG para que los procesadores PLC-5 con características mejoradas puedan programar y cargar/descargar mensajes no solicitados (hasta 1000 elementos cada uno) mediante Ethernet por el módulo de interface Ethernet PLC-5. El tamaño de cada elemento depende del comando de mensaje que usa. Por ejemplo, un elemento binario contiene una palabra de 16 bits y un elemento de punto flotante (coma flotante) contiene dos palabras de 16 bits.

Para programar una instrucción MSG, debe proporcionar al módulo de interface Ethernet PLC-5 y al procesador PLC-5 con características mejoradas una dirección de control, la cual contiene los parámetros de estado e instrucción. Después de que se introduzcan los parámetros del bloque de control, el terminal de programación muestra una pantalla para la introducción de instrucciones mediante la cual se introducen los parámetros de instrucción almacenados en la dirección del bloque de control.

Cómo introducir los parámetros

El bloque de control almacena toda la información relacionada con el mensaje. Las instrucciones de mensaje Ethernet usan dos elementos MSG consecutivos. El primero contiene información de mensaje y el segundo contiene la dirección de destino.

Importante: Puesto que los mensajes Ethernet necesitan dos bloques de control consecutivos, el bloque de control de mensaje que usted especifique debe comenzar en un número par.

Después de introducirse el bloque de control, el software de programación PLC-5 muestra automáticamente una pantalla para la introducción de datos mediante la cual se introducen los parámetros de instrucción que se almacenan en la dirección del bloque de control.

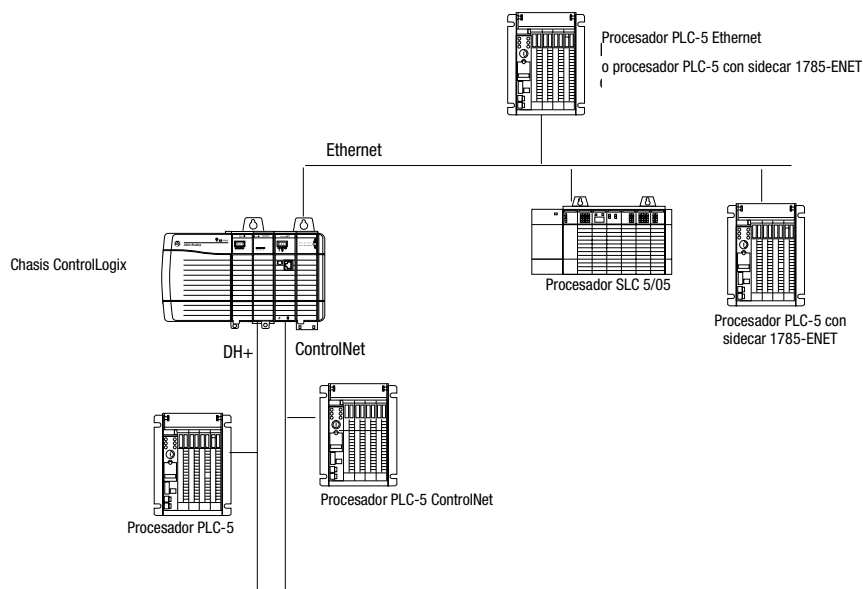
Debe introducir un número de puerto de 3A para habilitar una pantalla especial para las transferencias mediante Ethernet con el uso del módulo de interface Ethernet PLC-5.

Este campo:	Especifica esta información:
Command Type	Si la instrucción MSG realiza una operación de lectura o escritura. El software alterna entre: • PLC-5 Typed Read • PLC-5 Typed Write • PLC-5 Typed Write to SLC • PLC-5 Typed Read from SLC • SLC Typed Logical Read • SLC Typed Logical Write • PLC-2 Unprotected Read • PLC-2 Unprotected Write • PLC-3 Word Range Read • PLC-3 Word Range Write • ASCII
PLC-5 Address	La dirección del archivo de datos del procesador que contiene la instrucción de mensaje. Si la operación MSG es de escritura, esta dirección es la palabra inicial del archivo de origen. Si la operación MSG es de lectura, esta dirección es la palabra inicial del archivo de destino.
Size in Elements	El número de elementos (1-1000) que se deben transferir.
IP Address	El nodo de destino de la instrucción MSG. • Si el destino es un procesador PLC-5 con características mejoradas, el destino debe ser una dirección Internet completa. • Si el destino es un programa de cliente INTERCHANGE™, introduzca la palabra "CLIENT" (cliente) como el nodo de destino. No introduzca una dirección IP. Nota: debe establecer [F10] – port number a 2 para obtener acceso a esta función.
Destination Address	La dirección inicial del archivo de origen o destino en el procesador receptor.
Port Number	El canal para las comunicaciones de mensaje. Las comunicaciones del módulo de interface Ethernet PLC-5 usan el canal 3A.

El retiro del módulo de interface Ethernet PLC-5 no cambiará el formato de las instrucciones MSG definidas para el módulo.

Cómo configurar una instrucción MSG de multsaltos Ethernet

Los procesadores PLC-5 de serie E, revisión D y posteriores, pueden comunicarse mediante Ethernet con dispositivos ControlLogix o mediante un módulo Ethernet (1756-ENET) ControlLogix con otros procesadores PLC-5. Se necesita un procesador PLC-5 Ethernet o cualquier procesador PLC-5 que tenga el módulo sidecar 1785-ENET de serie A, revisión E. El diagrama siguiente muestra un procesador PLC-5 Ethernet y los otros procesadores PLC y SLC con los cuales se puede comunicar usando una instrucción MSG de multsaltos.



Para comunicarse a través de un módulo 1756-ENET ControlNet, se configura la característica de multsaltos de una instrucción MSG desde el procesador PLC-5 Ethernet (o procesador PLC-5 con módulo sidecar 1785-ENET) hacia el dispositivo receptor. Se necesita el software de programación RSLogix5. Habilite la opción de multsaltos cuando especifique el dispositivo receptor. Use la ficha Multihop (multsaltos) para especificar la ruta de acceso de la instrucción MSG.

Si desea pasar por el módulo 1756-ENET ControlLogix y a través del módulo 1756-DHRIO hacia el dispositivo receptor:

- use el software de configuración Gateway para configurar la tabla de encaminamiento del módulo 1756-DHRIO en el sistema ControlLogix.
- especifique un número de identificación de vínculo en las propiedades del canal para el canal 2/3A del procesador PLC-5 Ethernet (o procesador PLC-5 con un módulo sidecar 1785-ENET).

Vea los documentos del software de programación para obtener más información acerca de cómo configurar un canal PLC-5 y cómo especificar la ruta de acceso de la instrucción MSG.

Cómo usar la instrucción de mensaje para las comunicaciones ControlNet

Use la instrucción MSG para crear mensajes no programados (hasta 1000 elementos cada uno) iniciados por un procesador PLC-5 ControlNet y enviados a otro procesador PLC-5 ControlNet. Vea el Manual del usuario de los controladores programables PLC-5 ControlNet para obtener más información acerca de las operaciones de E/S ControlNet.

Cuando las condiciones de entrada cambian de falso a verdadero, los datos se transfieren según los parámetros de instrucción que estableció al introducir la instrucción MSG.

Para programar una instrucción MSG, debe proporcionar al procesador PLC-5 ControlNet una dirección de bloque de control que contenga los parámetros de estado e instrucción. Después de que se introducen los parámetros del bloque de control, el terminal de programación muestra una pantalla para la introducción de instrucciones mediante la cual se introducen los parámetros de instrucción almacenados en la dirección del bloque de control.

Dirección del bloque de control

Use un archivo de datos de mensaje (MG) con los procesadores PLC-5 ControlNet para el bloque de control de mensaje. Por ejemplo, MG20:50 es una dirección de bloque de control MSG válida.

Puede usar el tipo de archivo de mensaje (MG) y la instrucción MSG para enviar dos comandos mediante ControlNet dentro de la red ControlNet local:

- PLC-5 Typed Write
- PLC-5 Typed Read

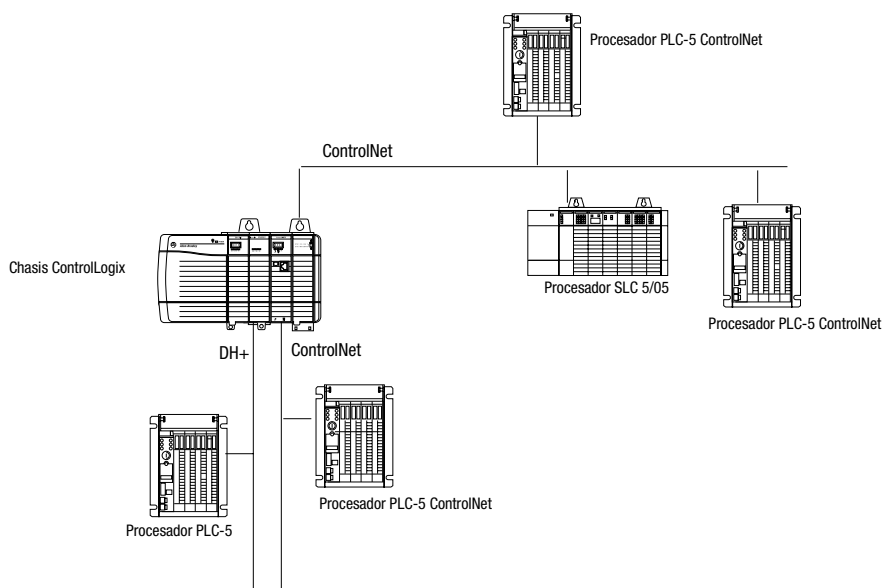
Después de que usted introduce la dirección del bloque de control para la instrucción MSG, el terminal de programación muestra una pantalla para la instrucción de instrucciones. Presione la tecla de función para los datos que desea modificar. Puede especificar lo siguiente mediante la pantalla de la introducción de instrucciones:

Este campo:	Especifica esta información:
Command Type	Cambie el tipo de comando. Alterne entre lo siguiente: • PLC-5 Typed Write selecciona una operación de escritura a un procesador PLC-5 ControlNet • PLC-5 Typed Read selecciona una operación de lectura de otro procesador PLC-5 ControlNet
PLC-5 Address	La dirección de la tabla de datos PLC-5 del procesador ControlNet. Si la operación MSG es de escritura, esta dirección es la palabra inicial del archivo de origen. Si la operación MSG es de lectura, esta dirección es la palabra inicial del archivo de destino.
Size in Elements	El número de elementos (1-1000) que se deben transferir.
Local Node	La dirección del nodo de destino (1-99).
Destination Address	La dirección inicial del archivo de origen o destino en el procesador receptor.
Port Number	El canal para las comunicaciones de mensaje. El número de puerto debe ser 2 para ControlNet.
Multihop	Selecione YES si desea enviar la instrucción MSG a un dispositivo ControlLogix. Use la ficha Multihop (multisaltos) para especificar la ruta de acceso de la instrucción MSG. Vea "Cómo configurar una instrucción MSG de multisaltos ControlNet" en la página 16-11.

Cómo configurar una instrucción MSG de multsaltos ControlNet

Los procesadores PLC-5 ControlNet de serie F, revisión A y posteriores, pueden comunicarse mediante ControlNet con dispositivos ControlLogix o por un módulo ControlNet (1756-CNB) ControlLogix con otros procesadores PLC-5 ControlNet en otras redes. Los procesadores PLC-5 ControlNet de series anteriores se pueden actualizar para ser compatibles con los mensajes de red de ControlNet a ControlNet y responder a mensajes de multsaltos mediante una red DH+. Los procesadores PLC-5 ControlNet de serie F, revisión A, añaden compatibilidad para mensajes de red de ControlNet a otros dispositivos.

El diagrama siguiente muestra un procesador PLC-5 ControlNet y los otros procesadores PLC y SLC con los cuales se puede comunicar usando una instrucción MSG de multsaltos.



Para comunicarse por un módulo 1756-CNB ControlLogix, usted configura la característica de multsaltos de una instrucción MSG desde el procesador PLC-5 ControlNet hacia el dispositivo receptor. Se necesita el software de programación RSLogix 5. Habilite la opción de multsaltos cuando especifique el dispositivo receptor. Use la ficha Multihop para especificar la ruta de acceso de la instrucción MSG.

Si desea pasar por el módulo 1756-ENET ControlLogix y a través del módulo 1756-DHRIO hacia el dispositivo receptor:

- use el software de configuración Gateway para configurar la tabla de encaminamiento del módulo 1756-DHRIO en el sistema ControlLogix.
- especifique un número de identificación de vínculo en las propiedades del canal para el canal 2/3A del procesador PLC-5 Ethernet (o procesador PLC-5 con un módulo sidecar 1785-ENET).

Vea los documentos del software de programación para obtener más información acerca de cómo configurar un canal PLC-5 y especificar la ruta de acceso de la instrucción MSG.

Cómo usar los bits de estado

La instrucción MSG usa los siguientes bits de estado:



ATENCION: No modifique ningún bit de estado cuando la instrucción esté habilitada. Puede resultar en una operación inesperada de la máquina, con posibles daños al equipo y lesiones corporales.

Importante: Los etiquetas de bit (.EN, .ST, .CO, etc.) se pueden usar solamente con el tipo de archivo de mensaje (MG).

Este bit:	Se establece:
Habilitación .EN (bit 15)	cuando el renglón se hace verdadero. Este bit indica que la instrucción está habilitada (que la instrucción se ejecuta). En el modo no continuo, el bit .EN permanece establecido hasta que el mensaje se completa y el renglón se hace falso. En el modo continuo, una vez establecido el bit .EN, permanece establecido independientemente de la condición del renglón.
Inicio .ST (bit 14)	cuando el procesador comienza a ejecutar la instrucción MSG. El bit .ST se restablece cuando el bit .DN o el bit .ER se establece.
Efectuado .DN (bit 13)	cuando se transfirió el último paquete de la instrucción MSG. El bit .DN se restablece la próxima vez que el renglón asociado cambia de falso a verdadero. El bit .DN está activo solamente en el modo no continuo.
Error .ER (bit 12)	cuando el procesador detecta que la transferencia de mensaje ha fallado. El bit .ER se restablece la próxima vez que el renglón asociado cambia de falso a verdadero.
Continuo .CO (bit 11)	manualmente para la operación repetida de la instrucción MSG después del primer escán independientemente de que el procesador continúe escaneando el renglón. Restablezca el bit .CO si desea que la condición de renglón inicie los mensajes (retornar al modo no continuo).
Habilitado-esperando .EW (bit 10)	cuando el procesador detecta que una solicitud de mensaje ha entrado a la cola. El procesador restablece el bit .EW cuando el bit .ST está establecido.
No respuesta .NR (bit 09)	si el procesador receptor no responde a la primera solicitud MSG. El bit .NR se restablece cuando el renglón asociado va de falso a verdadero.
Tiempo de espera .TO (bit 08)	si usted establece el bit .TO a través de la lógica de escalera, el procesador deja de procesar el mensaje y establece el bit .ER (error de tiempo de espera 55). Un tiempo de espera de mensaje DH+ de 30-60 segundos. Un mensaje ControlNet tendrá un tiempo de espera de 4 segundos.
Sin caché .NC (procesadores ControlNet solamente)	si usted establece el bit .NC, la conexión abierta se cierra cuando el MSG se efectúa. Si este bit permanece restablecido, la conexión permanece abierta aun cuando el MSG se ha efectuado.



ATENCION: El procesador controla los bits de estado .ST y .EW asincrónicamente con el escán de programa. Si examina estos bits en la lógica de escalera, copie el estado a un bit de almacenamiento cuyo estado está sincronizado con el escán de programa. De lo contrario, los problemas de temporización pueden invalidar el programa, con posibles daños al equipo y lesiones corporales.

Importante: Si los bits de reinicio SFC y .CO se ponen a cero, los bits .EN, .ST, .DN, .ER, .ET y .NR se ponen a cero durante el preescán.

Cómo usar el bloque de control

Además de los bits de estado, el bloque de control contiene otros parámetros que el procesador usa para controlar las instrucciones de mensaje. La Tabla 16.A indica dichos valores.

Tabla 16.A
Valores en el bloque de control

Palabra – Bloque de control de números enteros	Bloque de control de mensaje	Descripción
0	.EN a .RW	Bits de control
0 - byte bajo	.ERR	Código de error
2 - byte alto	.RLEN	Longitud solicitada
2 - byte bajo	.DLEN	Longitud de efectuado
3		Datos internos

Código de error (.ERR)

El procesador almacena el código de error aquí si ocurre un problema durante la transmisión del mensaje. Los códigos de error aparecen en la Tabla 16.E.

Longitud solicitada (.RLEN)

Esta es la cantidad solicitada de elementos que el usuario desea transferir con la instrucción de mensaje.

Longitud transmitida (.DLEN)

Este es el número de elementos que el módulo transfiere después de que la instrucción concluye la ejecución. Este número debe coincidir con la longitud solicitada.

Cómo introducir los parámetros

Comando de comunicación

La tabla siguiente describe los comandos de comunicación.

Si desea que la instrucción:	Seleccione el comando:
lea datos identificados por un código de tipo. Este comando lee las estructuras de datos sin la necesidad de especificar la longitud de palabra. Por ejemplo, si selecciona una lectura de tipo de la sección de datos del temporizador PLC-5 con un tamaño de datos solicitado de 5 elementos, la instrucción MSG lee 15 palabras (5 estructuras de temporizador de 3 palabras cada una).	PLC-5 Typed Read
escriba los datos identificados por un código de tipo. Este comando escribe las estructuras de datos sin la necesidad de especificar la longitud de palabra.	PLC-5 Typed Write
lea las palabras de 16 bits desde cualquier zona de la tabla de datos PLC-2 ó archivo de compatibilidad PLC-2.	PLC-2 Unprotected Read
escriba las palabras de 16 bits hacia cualquier zona de la tabla de datos PLC-2 ó archivo de compatibilidad PLC-2.	PLC-2 Unprotected Write
lea los datos identificados por un código de tipo. Este comando lee las estructuras de datos sin la necesidad de especificar la longitud de palabra. Este comando proporciona verificación adicional de datos para las comunicaciones entre un procesador PLC-5 y SLC 500. ¹	PLC-5 Typed Read from SLC ^{2, 3}
escriba los datos identificados por un código de tipo. Este comando escribe las estructuras de datos sin la necesidad de especificar la longitud de palabra. Este comando proporciona verificación adicional de datos para las comunicaciones entre un procesador PLC-5 y SLC 500. ¹	PLC-5 Typed Write from SLC ^{2, 3}
lea un rango de palabras, a partir de la dirección especificada para la dirección externa en el archivo de control MSG y leyendo secuencialmente el número de palabras especificado para el campo de tamaño solicitado en el archivo de control MSG. Los datos leídos se almacenan, a partir de la dirección especificada para la dirección interna en el archivo de control MSG. Esto se usa para la comunicación entre el procesador PLC-5 y SLC 500. ¹	SLC Typed Logical Read
escriba un rango de palabras, a partir de la dirección especificada para la dirección interna en el archivo de control MSG y escribiendo secuencialmente el número de palabras especificado para el campo de tamaño solicitado en el archivo de control MSG. Se escriben los datos de la dirección interna, a partir de la dirección especificada para la dirección externa en el archivo de control MSG. Esto se usa para la comunicación entre el procesador PLC-5 y SLC 500. ¹	SLC Typed Local Write ³
lea un rango de palabras, a partir de la dirección especificada para la dirección externa en el archivo de control MSG y leyendo secuencialmente el número de palabras especificado para el campo de tamaño solicitado en el archivo de control MSG. Los datos leídos se almacenan, a partir de la dirección especificada para la dirección interna en el archivo de control MSG.	PLC-3 Word Range Read
escriba un rango de palabras, a partir de la dirección especificada para la dirección interna en el archivo de control MSG y escribiendo secuencialmente el número de palabras especificado para el campo de tamaño solicitado en el archivo de control MSG. Se escriben los datos de la dirección interna, a partir de la dirección especificada para la dirección externa en el archivo de control MSG.	PLC-3 Word Range Write

¹El PLC-5 está limitado a un mensaje máximo de 103 palabras (206 bytes). El tamaño de mensaje máximo para los procesadores SLC 5/03™ y SLC 5/04™ es 103 palabras (206 bytes). La capacidad del tamaño de mensaje máximo de todos los demás procesadores SLC 500 es 41 palabras (82 bytes).

²Estos comandos son válidos solamente con cualquier procesador SLC 5/04 y SLC 5/03 de serie C y posteriores.

³Estos comandos son válidos solamente con los procesadores que aparecen en la lista en la página 16-2.

Puede usar los comandos de Typed Read y Typed Write para transferir secciones de la tabla de datos sin contar las palabras por elemento de la tabla de datos. Es necesario especificar solamente el número de elementos que desea transferir. Por ejemplo en la sección del temporizador de la tabla de datos, un elemento contiene 3 palabras, en cambio, en la sección binaria de la tabla de datos, un elemento contiene una palabra.

Direcciones externas de la tabla de datos

La tabla siguiente indica las direcciones externas válidas de la tabla de datos.

Este comando de comunicación:	A este dispositivo:	Requiere que introduzca:	Ejemplo de dirección:
PLC-5 Typed Read	PLC-5/250	la dirección entre comillas	"1N0:0"
PLC-5 Typed Write	PLC-5	la dirección	N7:0
	1775-S5	la dirección entre comillas con un carácter inicial de \$	"\$N7:0"
	1775-SR5		
PLC-2 Unprotected Read PLC-2 Unprotected Write	Compatible con PLC-2 PLC-2	número octal de offset de palabra de 16 bits	025
PLC-3 Word Range Read PLC-3 Word Range Write	PLC-5/250	la dirección entre comillas	"1N7:0"
	PLC-5	la dirección entre comillas con un carácter inicial de \$	"\$N7:0"
	1775-S5	la dirección entre comillas con un carácter inicial de \$ o la dirección solamente (esto es un poco más rápido)	"\$N7:0" N7:0
	1775-SR5		
	Coprocesadores de control 1771-DMC	la dirección entre comillas "00" a "31" para coincidir con el programa C	"01"
SLC Typed Logical Read SLC Typed Logical Write	Procesadores SLC 500	la dirección	N7:0
PLC-5 Typed Read to SLC PLC-5 Typed Write from SLC	Procesadores SLC 5/03 y 5/04	la dirección	N7:0

Archivos de compatibilidad de PLC-2 a PLC-5

Para enviar un mensaje entre un PLC-2 y un PLC-5, debe usar un archivo de compatibilidad PLC-2 dentro del procesador PLC-5. Este número de archivo debe ser la equivalencia decimal de la dirección octal del PLC-2. Recomendamos que la dirección octal del PLC-2 sea mayor que 10 para que no interfiera con los archivos de datos PLC-5 predeterminados.

Por ejemplo, si un PLC-2 se encuentra en la estación 12, cualquier mensaje que éste envíe retorna predeterminadamente al archivo 10 en el PLC-5 (la equivalencia decimal a 12 octal). Además, tome nota de que las direcciones PLC-2 son octales, si usted tiene una dirección PLC-2 como 024 en un comando de escritura, la escritura se realiza realmente en la palabra del PLC-5 (la equivalencia decimal a 24 octal).

Cómo enviar los comandos SLC Typed Logical Read y SLC Typed Logical Write

Siga estas pautas cuando programe los comandos SLC Typed Logical Read y SLC Typed Logical Write:

- Debe usar el tipo de datos MG para el bloque de control MSG..
- Los tipos de la dirección de la tabla de datos PLC-5 y la dirección de destino deben coincidir cuando el tipo de datos es compatible con los procesadores PLC 5/03 y 5/04. Si desea enviar un tipo de datos que no es compatible con los procesadores SLC 5/03 y SLC 5/04, los procesadores SLC interpretan dichos datos como números enteros. Esta tabla asigna los tipos de datos de los procesador PLC-5 a los procesadores SLC 5/03 y 5/04.

Este tipo de datos PLC-5:	Es interpretado por los procesadores SLC 5/03 y 5/04 como:
Binario (B)	Bit
Números enteros (N)	Números enteros
Salida (O)	Números enteros
Entrada (I)	Números enteros
Estado (S)	Números enteros
ASCII (A)	ASCII
BCD (D)	Números enteros
Escado SFC (SC)	Números enteros
Cadena (ST)	Cadena
Control BT (BT)	Números enteros
Transferencia ControlNet (CT)	Números enteros
Temporizador (T)	Temporizador
Contador (C)	Contador
Control (R)	Contador
Punto flotante (coma flotante) (F)	Punto flotante (coma flotante)
Control MSG (MG)	Números enteros
Control PID (PD)	Números enteros

- Para leer/escribir desde el archivo SLC de entrada, salida (lectura solamente) o estado, especifique una dirección de la tabla de datos PLC-5 de número entero y especifique la dirección del archivo SLC de entrada, salida o estado. Por ejemplo, S:37 para la palabra 37 del archivo de estado SLC. Especifique las direcciones SLC de entrada/salida según el formato lógico, es decir, O:001 hace referencia a la ranura 1.

- Los datos ASCII PLC-5 son datos de bytes (1/2 palabra). En cambio, un elemento de datos ASCII SLC es una palabra. Por lo tanto, si solicita una lectura de tipo PLC-5 de 10 elementos, el procesador SLC 500 envía un paquete con contiene 20 bytes (10 palabras)
- Los procesadores PLC-5 permiten 1000 elementos para la mayor parte de los tipos. En cambio, los procesadores SLC 500 permiten solamente 256 elementos.

Cómo monitorear una instrucción de mensaje

Para monitorear o editar los parámetros y bits de estado de la instrucción MSG después de introducir la instrucción MSG, muestre la pantalla del monitor de datos para la instrucción MSG y el tipo de archivo que usa.

Si usa este tipo de archivo:	Vea la:
Números enteros (N)	Tabla 16.A
Mensaje (MG)	Tabla 16.C

Si usa un tipo de archivo de números enteros (N), puede hacer lo siguiente mediante la pantalla del monitor de datos (Tabla 16.B):

Tabla 16.B
Pantalla del monitor de datos para la instrucción MSG – Tipo de archivo N

Se desea:	Presione esta tecla:
especificar el número de elementos (1-1000) que desea leer desde o escribir hacia la estación de la red	[F3] – Size in Elements
establecer y restablecer los bits de estado	[F9] – Toggle Bit

Si usa un tipo de archivo de mensaje (MG), puede hacer lo siguiente mediante la pantalla del monitor de datos (Tabla 16.C):

Tabla 16.C
Pantalla del monitor de datos para la instrucción MSG – Tipo de archivo MG

Se desea:	Presione esta tecla:
Alternar el bit de control en que se encuentra el cursor. Puede alternar entre los bits TO, NR, EW, CO, ER, DN, ST y EN	[F2] – Toggle Bit
Cambiar el tamaño del bloque de datos que desea enviar o recibir.	[F3] – Size in Elements
Cambiar la dirección para la cual los datos se muestran en pantalla.	[F5] –Specify Address
Mostrar en pantalla los valores de la tabla de datos para el próximo archivo.	[F7] – Next File
Mostrar en pantalla los valores de la tabla de datos para el archivo anterior.	[F8] –Previous File
Mostrar en pantalla los valores de la tabla de datos para el próximo elemento.	[F9] – Next Element
Mostrar en pantalla los valores de la tabla de datos para el elemento anterior.	[F10] – Previous Element

Cómo seleccionar la operación continua

El modo continuo le permite usar múltiples transferencias de mensaje porque programa solamente una instrucción MSG (sin condiciones de entrada en el renglón). Una vez que la transferencia de mensaje se inicia, la transferencia se ejecuta continuamente independientemente de que el procesador continúe escaneando el renglón asociado e independientemente de la condición del renglón. Establezca el bit .CO para habilitar la operación continua.



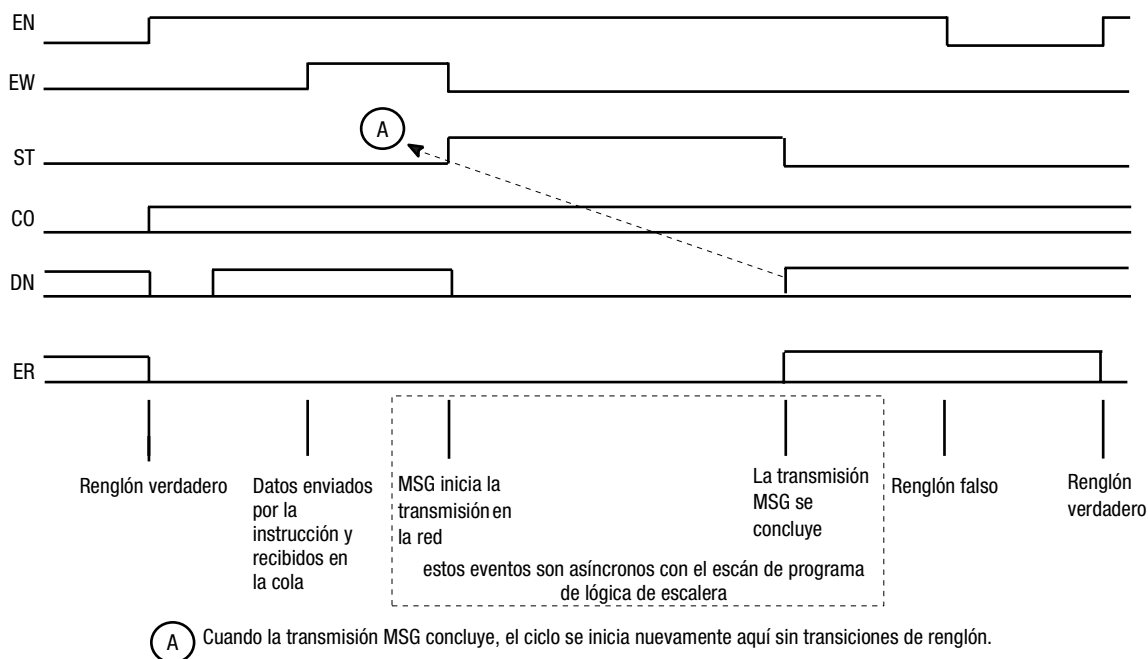
ATENCIÓN: Para que el modo continuo opere correctamente, se debe establecer el bit .CO (en la pantalla de configuración o mediante la lógica de escalera) **antes** de habilitar la instrucción MSG.

El modo continuo funciona así (Figura 16.1):

1. Cuando el renglón que contiene la instrucción MSG se hace verdadero, el procesador que inicia la instrucción MSG establece el bit .EN. El procesador también restablece los bits .ER y .DN.
2. El procesador pone en la cola la solicitud de mensaje. Cuando la solicitud de mensaje entra a la cola, el procesador establece el bit .EW.
3. Cuando el procesador comienza a procesar la solicitud de mensaje, el procesador establece el bit .ST. La próxima vez que el procesador recibe el control de red, el procesador transmite el mensaje.
4. Si ocurre un error, el procesador establece el bit .ER y almacena un código de error en el byte inferior de la palabra 0 del bloque de control para los procesadores PLC-5 clásicos y palabra 1 del bloque de control para los procesadores PLC-5 con características mejoradas.

Importante: La Figura 16.1 corresponde a los procesadores PLC-5 con características mejoradas solamente. Puede restablecer los procesadores PLC-5 clásicos si alterna los bits de error o habilitación.

Figura 16.1
Diagrama de temporización para los bits de estado en las
instrucciones MSG continuas



Una transferencia de mensaje continua continúa siempre que el procesador permanezca en el modo marcha o prueba. Si cambia al modo de programación o si el procesador falla, la transferencia de mensaje se detiene y no se reanuda hasta que el procesador escanea el renglón que contiene la instrucción MSG.

Restablezca el bit .CO para detener la operación continua.

Los procesadores PLC-5 anteriores a la serie E restablecen el bit .EN de un MSG continuo cuando el renglón se escanea como falso y el bit se establece .EN o .ER. Los procesadores de serie 5 y posteriores dejan el bit .EN establecido cuando el renglón es falso y el bit .DN está establecido. Esto indica el estado real de la instrucción MSG, la cual todavía está operando. Sin embargo, si el renglón es falso y el bit .ER está establecido, el bit .EN se restablece. Esto le permite reiniciar una instrucción MSG continua con error si alterna el estado del renglón.

Cómo seleccionar la operación no continua

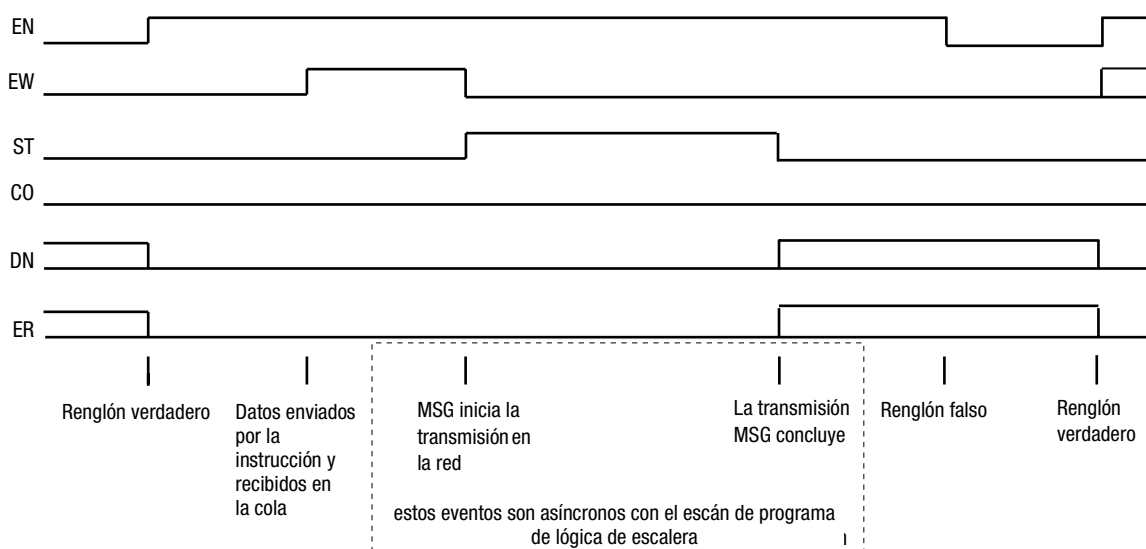
El modo no continuo realiza la transferencia de mensaje una vez por cada transición de falso a verdadero del renglón que contiene la instrucción MSG. La operación no continua ocurre siempre que el bit .CO permenezca restablecido. Use el modo no continuo cuando desea controlar cuándo ocurre la transferencia de mensaje o el número de veces que ocurre la transferencia de mensaje.

El modo no continua funciona así (Figura 16.2):

1. Cuando el renglón que contiene la instrucción MSG se hace verdadero, el procesador que inicia la instrucción MSG establece el bit .EN. El procesador también restablece los bits .EN y .ER.

2. El procesador pone en la cola la solicitud de mensaje. Cuando la solicitud de mensaje entra en la cola, el procesador establece el bit .EW.
3. Cuando el procesador comienza a procesar la solicitud de mensaje, el procesador establece el bit .ST. La próxima vez que el procesador recibe el control de red, el procesador transmite el mensaje.
4. Si no ocurre un error durante la transmisión, el procesador establece el bit .DN y restablece el bit .ST después de que se transfiere el último paquete en la primera ejecución de la instrucción MSG. Si ocurre un error, el procesador establece el bit .ER, restablece el bit .ST y almacena un código de error en el byte inferior de la palabra 0 del bloque de control para el PLC-5 clásico y palabra 1 del bloque de control para los procesadores PLC-5 con características mejoradas.
5. La próxima vez que el renglón se hace falso, el procesador restablece el bit .EN. Cuando el renglón asociado se hace verdadero nuevamente, el ciclo de transferencia de mensaje se vuelve a iniciar.

Figura 16.2
Diagrama de temporización para los bits de estado en las instrucciones MSG no continuas



Temporización MSG

El tiempo que requiere un procesador PLC-5 para enviar o recibir un mensaje hacia/desde otro procesador en la red DH+ depende del número de:

- estaciones en la red DH+
- mensajes transmitidos desde estaciones activas
- bytes de datos de todos los mensajes transmitidos
- solicitudes de mensaje que ya están en la cola

La temporización empieza con el establecimiento del bit de habilitación y termina con el establecimiento del bit de efectuado en el programa de lógica de escalera de la estación que inicia el mensaje. El orden de operación aparece en la Tabla 16.D.

Tabla 16.D
Operación de la instrucción de mensaje

Recepción de MSG (la estación A lee/recibe desde la estación B)	Transmisión de MSG (la estación A escribe/envía hacia la estación B)
la estación A habilita la instrucción de mensaje en el programa de lógica de escalera	la estación A habilita la instrucción de mensaje en el programa de lógica de escalera
la estación A obtiene el testigo y transmite el comando de lectura (la estación B conforma inmediatamente)	la estación A obtiene el testigo y transmite los datos (la estación B conforma inmediatamente)
la estación B obtiene el testigo y transmite los datos solicitados	la estación B almacena los datos en memoria
la estación A recibe los datos y confirma inmediatamente	la estación B obtiene el testigo y responde que la escritura se ha completado
la estación A establece el bit de efectuado	la estación A establece el bit de efectuado cuando recibe una respuesta

Se puede calcular el tiempo necesario (en milisegundos) para transmitir un paquete mediante DH+ con el uso de las siguientes fórmulas :

Tipo de procesador	Fórmula:
PLC-5 clásico	Tiempo de mensaje = TP + TT + OH + P + 8 (número de mensajes)
PLC-5 con características mejoradas	Tiempo de mensaje = TP + TT + OH + 8 (número de mensajes)

donde:

- TP = paso de testigo = (1.5) (1 + número de estaciones en la red DH+)
- TT = tiempo de transmisión = (0.28) (número de palabras de datos) Número de palabras de datos en todos los mensajes transmitidos para un paso de testigo alrededor de la red DH+.
- OH = tiempo de procesamiento interno DH+ = 20 ms
- P = el escán de programa más largo para cualquier procesador en la red DH+ (valor de aplicación en milisegundos)

Vea el Manual del usuario de los controladores programables PLC-5VMEbus y el Manual de los controladores programables PLC-5 Ethernet para obtener las cifras de rendimiento y los puntos de referencias.

Códigos de error

Cuando el procesador detecta un error durante la transferencia de datos de mensaje, el procesador establece el bit .ER e introduce un código de error que usted puede monitorear mediante el terminal de programación. Si el mensaje no es continuo, el procesador establece el bit .ER la primera vez que el procesador escanea la instrucción MSG.

Tabla 16.E
Errores detectados por el procesador

Código:			
PLC-5 ¹ con características mejoradas Tipo de datos MG	PLC-5 ² clásico Tipo de datos N	Ethernet solamente	Descripción (se muestra en la pantalla del monitor de datos)
0037	55	0037	message timed out in local processor
0083	131	0083	processor is disconnected
0089	137	0089	message buffer is full Si el MSG sale del canal 0, no hay disponibles suficientes búferes internos. Disminuye el número de instrucciones MSG a este puerto. De lo contrario, el nodo de destino devolvió un MSG que indicó que los búferes están completos. Disminuya el número de instrucciones MSG que van al nodo de destino.
0092	146	0092	no response (regardless of station type)
00D3	211	00D3	you formatted the control block incorrectly
00D5	213	00D5	incorrect address for the local data table
0200	2		link layer timed out or recieved a NAK
0300	3		duplica token holder detected by a link layer
0400	4		local port is disconnected
0500	5		application layer timed out waiting for a response
0600	6		duplicate node detected
0700	7		station is off line
0800	8		hardware fault
1000	129	1000	illegal command from local processor
2000	130	2000	communication module not working
3000	131		remote node is missing, disconnected, or shut down
4000	132	4000	processor connected but faulted (hardware)
5000	133	5000	you uted the wrong station number
6000	134	6000	requested function is not available
7000	135	7000	processor is in program mode

¹Hexadecimal – palabra 1 del bloque de control

²Decimal – byte inferior de la palabra 0 del bloque de control

Código:			
PLC-5 ¹ con características mejoradas Tipo de datos MG	PLC-5 ² clásico Tipo de datos N	Ethernet solamente	Descripción (se muestra en la pantalla del monitor de datos)
8000	136	8000	processor's compatibility file does not exist
9000	137	9000	remote node cannot buffer command
B000	139	B000	processor is downloading so it is inaccessible
F001	231	F001	processor incorrectly converted the address
F002	232	F002	incomplete address
F003	233	F003	incorrect address
F006	236	F006	addressed file does not exist in targ processor
F007	237	F007	destination file is too small for number of words requested
F00A	240	F00A	target processor cannot put requested information in packets
F00B	241	F00B	privilege error, access denied
F00C	242	F00C	requested function is not available
F00D	243	F00D	request is redundant
F011	247	F011	data type requested does not match data available
F012	248	F012	incorrect command parameters
0010 ³		0010	no IP address configured for the network
0011 ³		0011	already at maximum number of connections
0012 ³		0012	invalid internet address or host name
0013 ³		0013	no such host
0014 ³		0014	cannot communication with the name server
0015 ³		0015	connection not completed before user-specified timeout
0016 ³		0016	connection timed out by the network
0017 ³		0017	connection refused by destination host
0018 ³		0018	connection was broken
0019 ³		0019	reply not received before user-specified timeout
001A ³		001A	no network buffer space available
		F01A	file owner active – the file is being used

¹Hexadecimal – palabra 1 del bloque de control²Decimal – byte inferior de la palabra 0 del bloque de control³Errores detectados por un procesador PLC-5 con características mejoradas conectado a un módulo de interface Ethernet PLC-5 solamente.

Código:			
PLC-5 ¹ con características mejoradas Tipo de datos MG	PLC-5 ² clásico Tipo de datos N	Ethernet solamente	Descripción (se muestra en la pantalla del monitor de datos)
		F01B	program owner active – someone is downloading, online editing, or set the program owner with APS in the WHO Active Screen

¹Hexadecimal – palabra 1 del bloque de control
²Decimal – byte inferior de la palabra 0 del bloque de control
³Errores detectados por un procesador PLC-5 con características mejoradas conectado a un módulo de interface Ethernet PLC-5 solamente.

Tabla 16.F
Errores detectados por el procesador VME

PLC-5/40V (hexadecimal – palabra 1 del bloque de control)	Descripción (se muestra en la pantalla del monitor de datos)
0000	success
0001	invalid ASCII message format
0002	invalid file type
0003	invalid file number
0004	invalid file element
0005	invalid VME address
0006	invalid VME transfer width
0007	invalid number of elements requested for transfer
0008	invalid VME interruptor level
0009	invalid VME interrupt status-id level
000A	VMEbus transfer error (bus error)
000B	unable to assert requested interrupt (already pending)
000C	raw data transfer setup error
000D	raw data transfer crash (PLC switched out of run mode)
000E	unknown message type (message type not ASCII)

Instrucciones ASCII ABL, ACB, ACI, ACN, AEX, AIC, AHL, ARD, ARL, ASC, ASR, AWA, AWT

Cómo usar las instrucciones ASCII para los procesadores PLC-5 con características mejoradas solamente

Las instrucciones ASCII leen, escriben, comparan y convierten las cadenas ASCII. Estas instrucciones son compatibles solamente con los procesadores PLC-5 con características mejoradas. La Tabla 17.A indica las instrucciones ASCII disponibles.

Tabla 17.A
Instrucciones ASCII disponibles

Si desea:	Use esta instrucción:	Que se encuentra en la página:
ver cuántos caracteres hay en el búfer hasta el carácter de fin de línea inclusive	ABL	17-4
ver el total de caracteres en el búfer	ACB	17-5
convertir una cadena en un valor entero	ACI	17-6
concatenar dos cadenas en una	ACN	17-7
extraer una porción de una cadena para crear una nueva cadena	AEX	17-7
configurar las líneas de handshake del módem	AHL	17-8
convertir un valor entero en una cadena	AIC	17-9
leer los caracteres desde el búfer y ponerlos en una cadena	ARD	17-10
leer una línea de caracteres desde el búfer y ponerla en una cadena	ARL	17-12
buscar una cadena en otra cadena	ASC	17-14
comparar dos cadenas	ASR	17-15
escribir una cadena con caracteres añadidos configurados por el usuario	AWA	17-15
escribir una cadena	AWT	17-17

Vea el Apéndice C para obtener más información acerca de los operandos (y los valores/tipos de datos válidos de cada operando) usados por las instrucciones descritas en este capítulo.

Hay dos tipos de instrucciones ASCII:

Tipo de instrucción ASCII:	Descripción:
Control de puerto ASCII	lee, escribe, establece/restablece las líneas de handshaking, examina la longitud del búfer (ARD, ARL, AWT, AWA, AHL, ACB, ABL)
Cadena ASCII	manipula datos de cadena, tales como comparación, búsqueda, extracción, concatenación, conversión de/a número entero (ASR, ASC, AEX, ACN, ACI, AIC)

Las instrucciones ASCII dependen el una de otra. Por ejemplo, si tiene una ARD (instrucción de lectura ASCII) y luego una AWT (escritura ASCII), el bit de efectuado en la ARD se debe establecer para que la AWT pueda comenzar a ejecutarse (aun cuando la AWT se habilitó mientras el procesador ejecutaba la ARD). Una segunda instrucción ASCII no puede comenzar hasta que se complete la primera. Sin embargo, el procesador no espera hasta que la instrucción ASCII se complete para continuar ejecutando el programa de lógica de escalera (instrucciones no ASCII).

Cómo usar los bits de estado

Usted puede examinar los bits de estado en el programa de lógica de escalera para examinar un evento determinado. El procesador cambia los estados de los bits de estado a la vez que el procesador ejecuta la instrucción. Usted direcciona los bits de estado mediante mnemónicos (o por número de bit) en la dirección del elemento de control.

Las instrucciones ASCII usan los campos de longitud (.LEN) y posición (.POS) en algunas instrucciones así como los siguientes bits de estado:

Descripción:	Explicación del bit de estado:
Encontrado .FD (08)	Reservado
Descarga .UL (10)	El usuario puede usar este bit para cancelar una lectura o escritura ASCII que está en progreso. El exceso del tiempo de espera puede ocurrir inmediatamente o hasta 6 segundos más tarde.
Error.ER (11)	La instrucción no logró completarse. Nota: si este bit está establecido, el bit .EN se pone a cero y el bit .DN se establece durante el preescán.
Efectuado síncrono .EM (12)	El bit se establece durante el primer escán de la instrucción después que ésta concluye
Efectuado asíncrono .DN (13)	El bit se establece inmediatamente una vez que la instrucción concluye con éxito, asincrónamente con el escán de programa. Nota: si este bit está establecido, el bit .EN se pone a cero y el bit .DN se establece durante el preescán.
Cola .EU (14)	El bit se establece cuando la instrucción logra ponerse en la cola.
Habilitación .EN (15)	El bit se establece cuando el renglón se hace verdadero y se restablece cuando concluye la instrucción y el renglón se hace falso. Nota: si este bit está establecido y los bits .DN y .ER se ponen a cero, se borra la palabra de control durante el preescán.

Cómo usar el bloque de control

Además de los bits de estado, el bloque de control contiene otros parámetros que el procesador usa para controlar las instrucciones de transferencia ASCII. La Tabla 17.B indica estos valores.

Tabla 17.B
Valores en la palabra de control

Palabra – Bloque de control de número entero	Bloque de control ASCII	Descripción
0	.EN, .DN, etc	Bits de estado
1	.LEN	Longitud de palabra
2	.POS	Posición de carácter

Longitud (.LEN)

Este es el número de caracteres en que se realiza la operación.

Posición (.POS)

Este es el número actual de caracteres que la operación ha ejecutado.

Cómo usar las cadenas

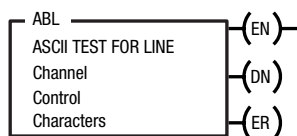
Puede direccionar las longitudes de cadenas agregando .LEN a cualquier dirección de cadena (por ejemplo, ST17:1.LEN).

Las longitudes de cadenas deben ser entre 0 y 82 bytes. Por lo general, las longitudes que se encuentran fuera de este rango causan que el procesador establezca un fallo menor (S:17/8) y la instrucción no se ejecuta.

Importante: Puede configurar caracteres añadidos o de fin de línea en la pantalla de configuración del canal. Los caracteres añadidos predeterminados son el retorno de carro y el salto de línea. El carácter de fin de línea (terminación) predeterminado es el retorno de carro. Vea el manual del usuario del software para obtener más información.

Test Buffer for Line (ABL)

Descripción:



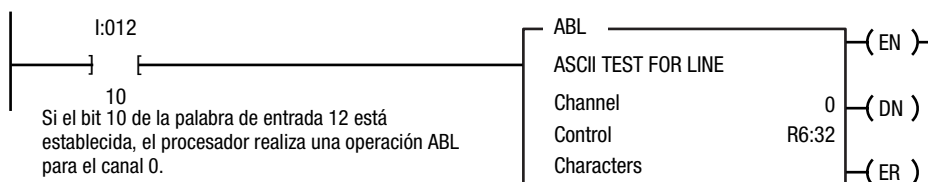
Use la instrucción ABL para averiguar cuántos caracteres hay en el búfer hasta los caracteres de fin de línea (terminación) inclusive. Durante una transición de falso a verdadero, el sistema indica el número de caracteres en el campo de posición y establece el bit de efectuado. El puerto en serie debe estar en el modo usuario.

Cómo introducir los parámetros

Debe proporcionar la siguiente información para usar la instrucción ABL:

Parámetro:	Definición:
Canal	el número del puerto RS-232. (El único valor válido es 0.)
Control	la dirección de un elemento de archivo de control usado para los bits de estado de control.
Caracteres	el número de caracteres en el búfer (incluso los caracteres de fin de línea/terminación) que el procesador encuentra. Este campo es de visualización solamente.

Ejemplo:



Cuando el renglón cambia de falso a verdadero, el bit de habilitación del elemento de control (.EN) se establece. La instrucción se pone en la cola de instrucciones ASCII, el bit .EU se establece y el escán de programa continúa. La instrucción se ejecuta en paralelo con el escán de programa.

El procesador determina el número de caracteres (hasta los caracteres de fin de línea/terminación inclusive) y coloca este valor en el campo de posición. El bit de efectuado se establece. Si aparece un cero en el campo de posición, no se encontró ningún carácter de fin de línea/terminación. El bit .FD se establece si el campo de posición se estableció en un valor que no sea cero.

Cuando el programa escanea la instrucción y encuentra el bit .DN efectuado, el procesador establece el bit .EM. El bit .EM sirve como bit de efectuado secundario para el escán del programa.

El bit de error (.ER) se establece durante la ejecución de la instrucción si:

- la instrucción se cancela – el puerto en serie no está en el modo usuario
- la instrucción se cancela debido a un cambio de modo del procesador

Number of Characters in Buffer (ACB)

Descripción:



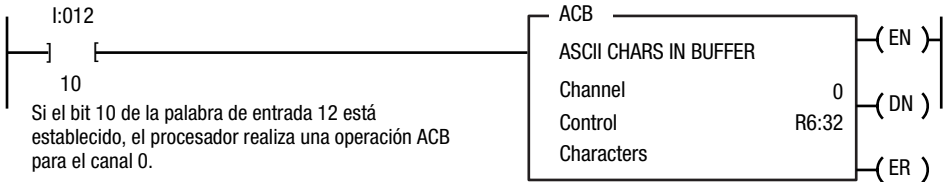
Use la instrucción ACB para averiguar el total de caracteres en el búfer. Durante una transición de falso a verdadero, el sistema determina el número total de caracteres y lo indica en el campo de caracteres. El puerto en serie debe estar en el modo usuario.

Cómo introducir los parámetros

Debe proporcionar la siguiente información para usar la instrucción ACB:

Parámetro:	Definición:
Canal	el número del puerto RS-232. (El único valor válido en este campo es 0.)
Control	la dirección de un elemento de archivo de control usado para los bits de estado.
Caracteres	el número de caracteres en el búfer que el procesador encuentra (0-256). Este campo es de visualización solamente.

Ejemplo:



Cuando el renglón cambia de falso a verdadero, el bit de habilitación del elemento de control (.EN) se establece. La instrucción se pone en la cola de instrucciones ASCII, el bit .EU se establece y el escán de programa continúa. La instrucción se ejecuta en paralelo con el escán de programa.

El procesador determina el número de caracteres en el búfer y coloca este valor en el campo de posición. El bit de efectuado se establece. Si aparece un cero en el campo de posición, no se encontró ningún carácter. El bit .FD se establece si el campo de posición se estableció en un valor que no sea cero.

Cuando el programa escanea la instrucción y encuentra el bit .DN efectuado, el procesador establece el bit .EM. El bit .EM sirve como bit de efectuado secundario para el escán de programa.

El bit de error (.ER) se establece durante la ejecución de la instrucción si:

- la instrucción se cancela – el puerto en serie no está en el modo usuario
- la instrucción se cancela debido a un cambio de modo del procesador

ASCII String to Integer (ACI)

Descripción:

ACI
STRING TO INTEGER CONVERSION
Source
Destination

Use la instrucción ACI para convertir una cadena ASCII en un valor de número entero entre -32,768 y 32,767.

El procesador busca en el origen (tipo de archivo ST) el primer carácter que sea entre 0 y 9. Todos los caracteres numéricos se extraen hasta que se alcanza un carácter no numérico o el fin de la cadena. Se permiten las comas y los signos (–, +) en la cadena.

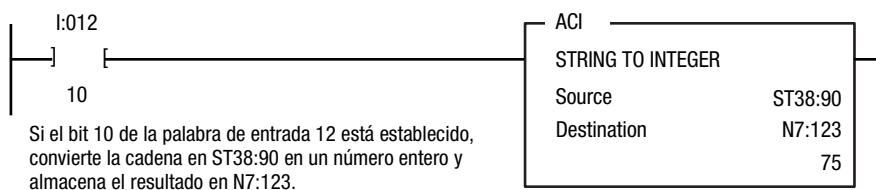
La cadena numérica extraída se convierte en un número entero entre -32,768 y 32,767.

Si no se encuentra ningún carácter numérico, una acción no es necesaria. Además, si la cadena tiene una longitud no válida (menor que cero o mayor que 82), el bit de fallo (S:17/8) se establece y la instrucción no se ejecuta.

Esta instrucción también establece los indicadores aritméticos (que se encuentran en la palabra 0, bits 0-3 en el archivo de estado del procesador S):

Bit:	Descripción:	Indica:
S:0/0	Acarreo (C)	que el acarreo se generó durante la conversión de la cadena en un número entero
S:0/1	Overflow (V)	que el número entero se encontró fuera del rango válido
S:0/2	Cero (Z)	que el valor entero es cero
S:0/3	Signo (S)	el valor entero es negativo

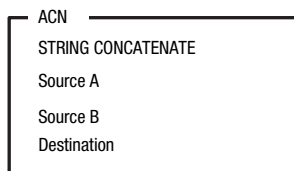
Ejemplo:



ASCII String Concatenate (ACN)

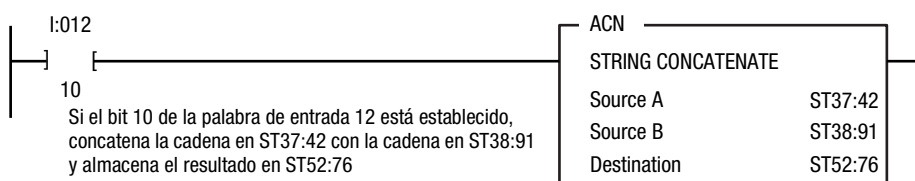
Descripción:

La instrucción ACN añade el origen B al fin del origen A y almacena el resultado en el destino.



Si el resultado es más largo que 82 caracteres, se escriben solamente los 82 primeros caracteres al archivo de destino y el bit de error (S:17/8) es establece. Además, si la longitud de las cadenas no es válida (menor que cero o mayor que 82), el bit de fallo se establece y la cadena en la dirección de destino no se cambia.

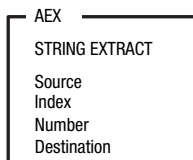
Ejemplo:



ASCII String Extract (AEX)

Descripción:

Use la instrucción AEX para crear una nueva cadena mediante la extracción de una porción de una cadena existente.

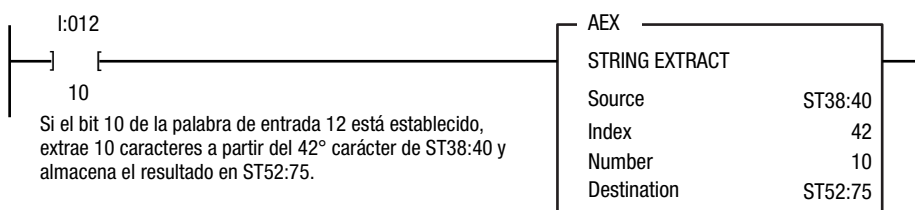


Cómo introducir los parámetros

Debe proporcionar la siguiente información para usar la instrucción AEX:

Parámetro:	Definición:
Origen	la cadena existente.
Índice	la posición inicial (de 1 a 82) de la porción de la cadena que desea extraer. (Un índice de 1 indica el carácter del extremo izquierdo de la cadena.)
Número	el número de caracteres (de 0 a 82) que desea extraer a partir de la posición indexada. Si el índice más el número son mayores que el total de caracteres en la cadena de origen, la cadena de destino será los caracteres desde el índice hasta el fin de la cadena de origen. Si introduce 0 para el número, la longitud de la cadena de destino se establece en cero.
Destino	el elemento de cadena (ST) donde desea almacenar la cadena extraída.

Ejemplo:



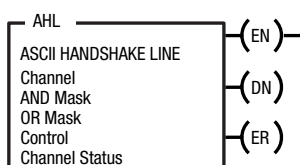
Las condiciones siguientes causan que el procesador establezca el bit de fallo (S:17/8):

- longitud de cadena no válida o longitud de cadena de cero
- valores de índice o número fuera del rango
- valor de índice mayor que la longitud de la cadena de origen

La cadena de destino no se cambiará en ninguna de las condiciones anteriores.

ASCII Set or Reset Handshake Lines (AHL)

Descripción:



Use la instrucción para establecer o restablecer las líneas de handshaking DTR y RTS RS-232 para el módem. Durante la transición de falso a verdadero, el sistema usa dos máscaras para determinar si debe establecer o restablecer las líneas DTR y RTS o dejarlas sin cambio.

Importante: Antes de usar la instrucción, asegúrese de no interferir con las líneas de control automáticas al módem.

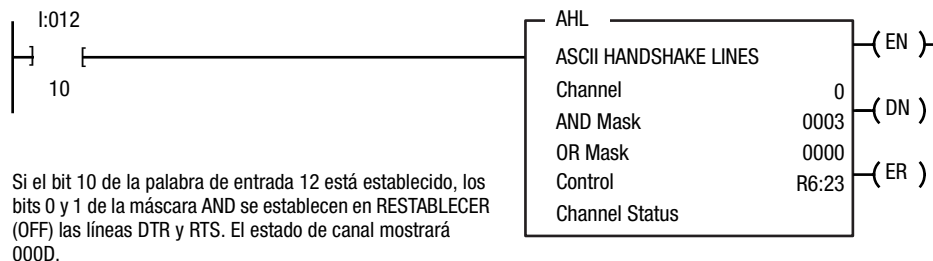
Cómo introducir los parámetros

Debe proporcionar la siguiente información para usar la instrucción AHL:

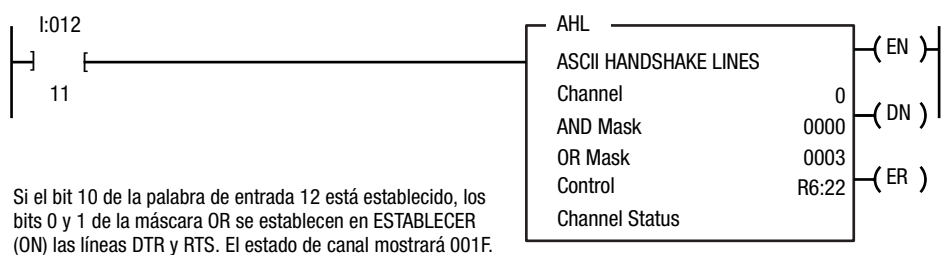
Parámetro:	Definición:
Canal	el número del puerto RS-232 que desea usar. Actualmente se puede establecer o restablecer el canal 0 solamente.
Máscara AND	la máscara para restablecer las líneas de control DTR y RTS. El bit 0 corresponde a la línea DTR y el bit 1 corresponde a la línea RTS. El número 1 en el bit de máscara causa que la línea se restablezca. El número 0 no cambia la línea.
Máscara OR	la máscara para establecer las líneas de control DTR y RTS. El bit 0 corresponde a la línea DTR y el bit 1 corresponde a la línea RTS. El número 1 en el bit de máscara causa que la línea se restablezca. El número 0 no cambia la línea.
Control	la dirección de la estructura de control de resultado en la zona de control de la memoria para el resultado.
Estado de canal	muestra en pantalla el estado actual (0000 a FFFF) de las líneas de handshaking para el canal especificado anteriormente. Este campo es para la muestra en pantalla solamente. Convierta el estado hexadecimal en binario y vea la tabla siguiente:

Bit	1	0
Línea	RTS	DTR

Ejemplo: (restablecimiento de las líneas DTR y RTS)



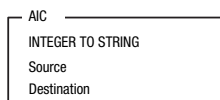
Ejemplo: (establecimiento de las líneas DTR y RTS)



El bit de error (.ER) se establece durante la ejecución de la instrucción si la instrucción se cancela debido al cambio de modo del procesador.

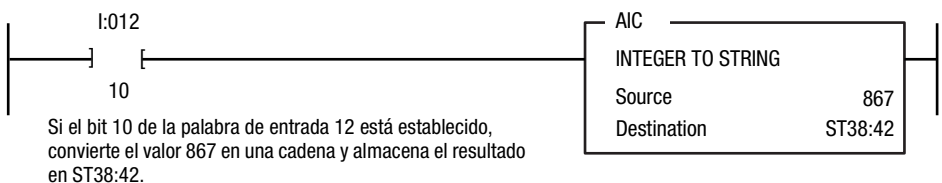
ASCII Integer to String (AIC)

Descripción:



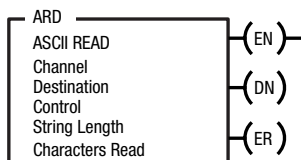
Use la instrucción AIC para convertir un valor entero (entre -32,768 y 32,767) en una cadena ASCII. El origen puede ser una constante o una dirección de número entero.

Ejemplo:



ASCII Read Characters (ARD)

Descripción:



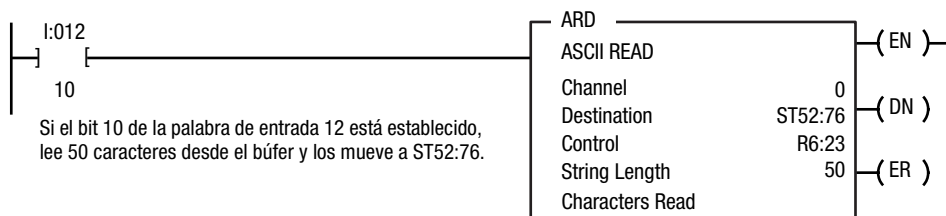
Use la instrucción ARD para leer caracteres desde el búfer y almacenarlos en una cadena. Para repetir la operación, el renglón debe ir de falso a verdadero. El puerto en serie debe estar en el modo usuario.

Cómo introducir los parámetros

Debe proporcionar la siguiente información para usar la instrucción ARD:

Parámetro:	Definición:
Canal	el número del puerto RS-232. (El único valor válido es 0.)
Control	el elemento del archivo de control usado para los bits de estado de control.
Destino	el elemento de cadena donde desea almacenar los caracteres.
Longitud de cadena	el número de caracteres que desea leer desde el búfer. El máximo es 82 caracteres. Si especifica una longitud mayor que 82, se leen solamente 82 caracteres. (Si especifica 0, la longitud de cadena tiene un valor predeterminado de 82.)
Caracteres leídos	el número de caracteres que el procesador movió del búfer a la cadena (0 a 82). Este campo es de visualización solamente.

Ejemplo:



Cuando el renglón cambia de falso a verdadero, el bit de habilitación del elemento de control (.EN) se establece. La instrucción se pone en la cola de instrucciones ASCII, el bit .EU se establece y el escán de programa continúa. La instrucción se ejecuta en paralelo con el escán de programa.

Una vez que el número de caracteres solicitado está en el búfer, los caracteres se mueven a la cadena de destino. El número de caracteres movidos se coloca en la palabra de posición del elemento de control y el bit de efectuado se establece.

Cuando el program escanea la instrucción y encuentra el bit .DN establecido, el procesador establece el bit .EM. El bit .EM sirve como un bit de efectuado secundario que corresponde al escán del programa.

Puede usar el bit .UL para terminar una instrucción ARD antes de que concluya (por ejemplo, es posible que desee terminar la instrucción si sabe que el dispositivo ASCII conectado al puerto no transmite datos o si la conexión se interrumpe después de que la instrucción comienza a ejecutarse). Establezca el bit .UL en la estructura de control (el bit .ER se establece).

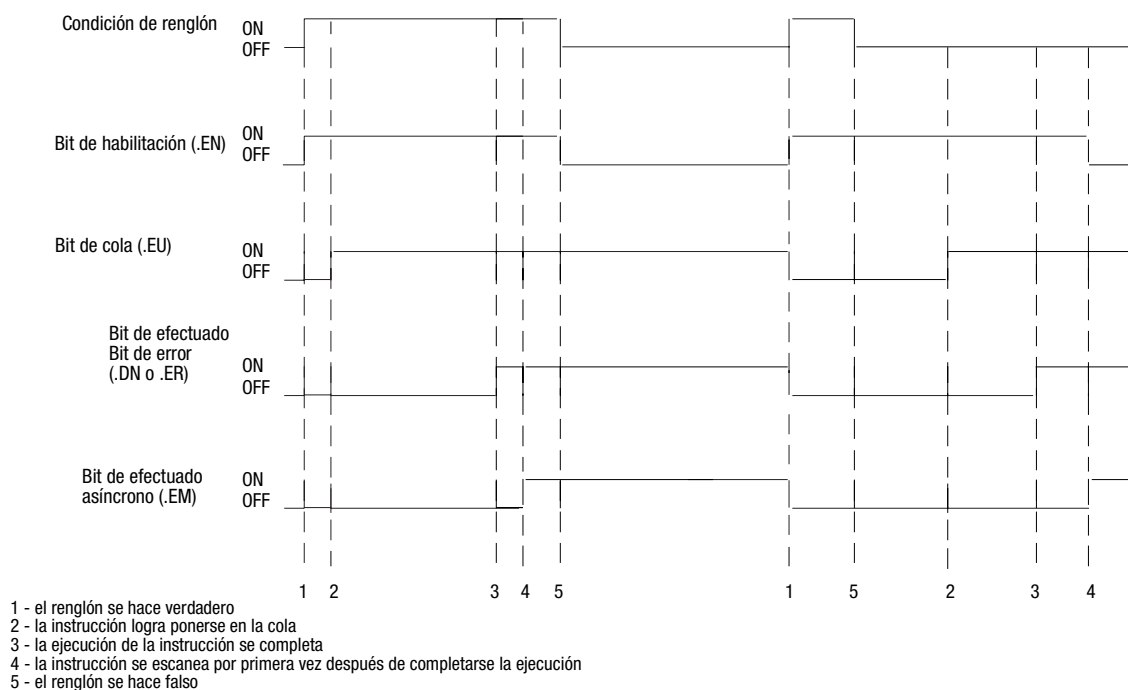
Importante: Cuando usted establece el bit .UL, la instrucción no se termina inmediatamente. El proceso puede tomar varios segundos.

Si una instrucción ARD comienza a ejecutarse con el bit .UL ya establecido y **no** hay caracteres en el búfer, la instrucción se termina. Si una instrucción comienza a ejecutarse con el bit .UL ya establecido y **hay** caracteres en el búfer, la instrucción se realiza normalmente.

El bit de error (.ER) se establece durante la ejecución de la instrucción si:

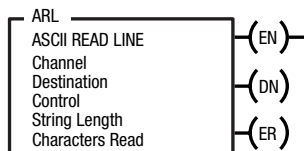
- la instrucción se cancela – el puerto en serie no está en el modo usuario
- la instrucción se cancela debido al cambio de modo del procesador
- cuando se usa un módem, el módem se desconecta

Figura 17.1
Ejemplo de un diagrama de temporización ARD



ASCII Read Line (ARL)

Descripción:



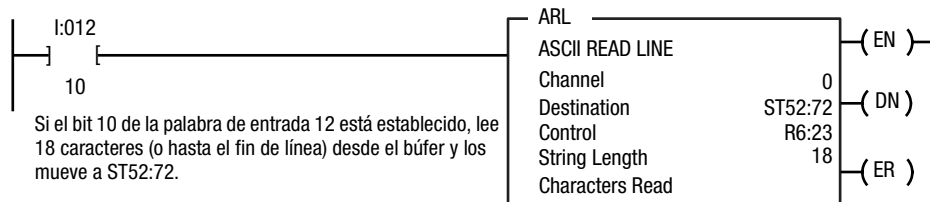
Use la instrucción ARL para leer caracteres desde el búfer hasta los caracteres de fin de línea (terminación) inclusive y almacenarlos en una cadena. Los caracteres de fin de línea se especifican en la pantalla de configuración del canal (la opción predeterminada es un retorno de carro). Vea el manual del usuario del software para obtener más información acerca de la configuración del canal.

Cómo introducir los parámetros

Debe proporcionar la siguiente información para usar la instrucción ARL:

Parámetro:	Definición:
Canal	el número del puerto RS-232. (El único valor válido es 0.)
Control	el elemento del archivo de control usado para los bits de estado de control.
Destino	el elemento de cadena donde desea almacenar los caracteres.
Longitud de cadena	el número de caracteres (máximo de 82) que desea leer desde el búfer. Si el procesador encuentra los caracteres de fin de línea antes de leer el número de caracteres especificado, solamente los caracteres leídos y el fin de línea se mueven al destino.
Caracteres leídos	el número de caracteres que el procesador movió del búfer a la cadena (0 a 82). Este campo es de visualización solamente.

Ejemplo:



Cuando el renglón cambia de falso a verdadero, el bit de habilitación del elemento de control (.EN) se establece. La instrucción se pone en la cola de instrucciones ASCII, el bit .EU se establece y el escán de programa continúa. La instrucción se ejecuta en paralelo con el escán de programa.

Una vez que el número de caracteres solicitado (o los caracteres de fin de línea) está en el búfer, todos los caracteres (incluso los caracteres de fin de línea) se mueven a la cadena de destino. El número de caracteres movidos se coloca en la palabra de posición del elemento de control y el bit de efectuado se establece.

Cuando el programa escanea la instrucción y encuentra el bit .DN establecido, el procesador establece el bit .EN. El bit .EM sirve como un bit de efectuado secundario que corresponde al escán del programa.

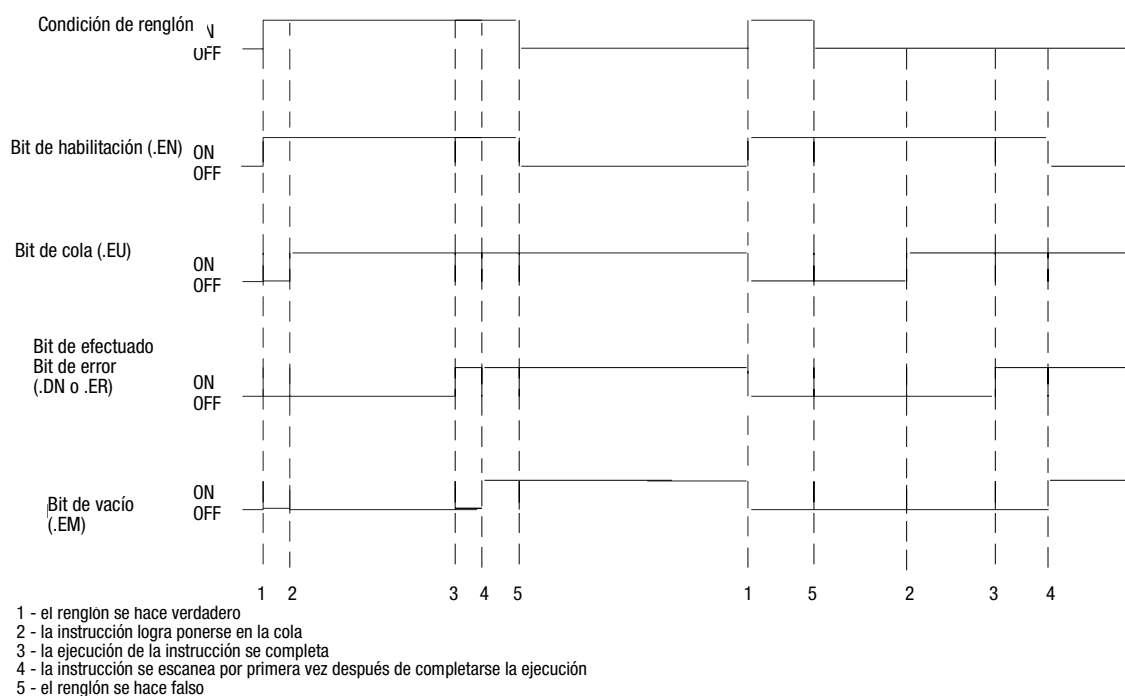
Puede usar el bit .UL para terminar una instrucción ARL antes de que concluye (por ejemplo, es posible que desee terminar la instrucción si sabe que el dispositivo ASCII conectado al puerto no transmite datos o si la conexión se interrumpe después de que la instrucción comienza a ejecutarse). Establezca el bit .UL en la estructura de control (el bit .ER se establece).

Importante: Cuando usted establece el bit .UL, la instrucción no se termina inmediatamente. Es posible que el proceso tome varios segundos.

Si una instrucción ARL comienza a ejecutarse con el bit .UL ya establecido y **no** hay caracteres en el búfer, la instrucción se termina. Si una instrucción comienza a ejecutarse con el bit .UL ya establecido y **hay** caracteres en el búfer, la instrucción se realiza normalmente.

El bit de error (.ER) se establece durante la ejecución de la instrucción si el canal está en el modo sistema (o si cambia al modo sistema), el procesador cambia al modo de programación/prueba o si el módem se pierde (cuando se usa el control de módem).

Figura 17.2
Ejemplo de un diagrama de temporización ARL



ASCII String Search (ASC)

Descripción:

ASC
STRING SEARCH
Source
Index
Search
Result

Use la instrucción ASC para buscar la ocurrencia de la cadena de origen en una cadena existente (cadena de búsqueda).

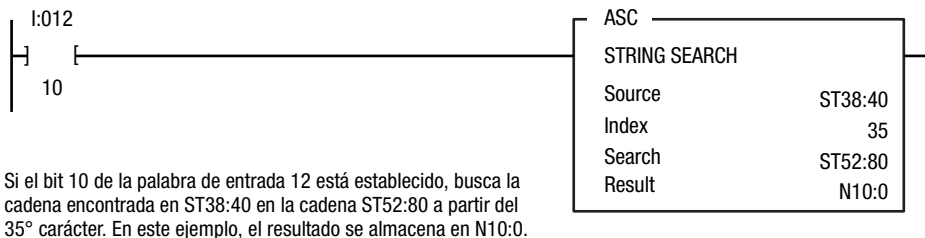
Cómo introducir los parámetros

Debe proporcionar la siguiente información para usar la instrucción ASC:

Parámetro: Definición:

Búsqueda	la cadena que desea examinar.
Origen	la cadena que desea encontrar cuando examina la cadena de búsqueda.
Índice	la posición inicial (de 1 a 82) de la porción de la cadena de búsqueda que desea buscar. Un índice de 1 indica el carácter del extremo izquierdo.
Resultado	una dirección de números enteros en que el procesador almacena la posición de la cadena de búsqueda donde comienza la cadena de origen. Si no hay coincidencias, 0 se almacena en el resultado.

Ejemplo:



Las condiciones siguientes causan que el procesador establezca el bit de fallo (S:17/8):

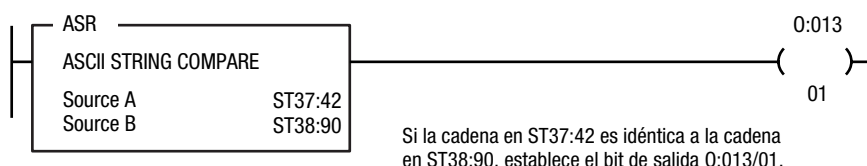
- longitud de cadena no válida o longitud de cadena de cero
- valores de índice fuera del rango
- valor de índice mayor que la longitud de la cadena de origen

El resultado se pone en cero en cualquiera de las condiciones anteriores.

ASCII String Compare (ASR)

Descripción: Use la instrucción ASR para comparar dos cadenas ASCII. El sistema busca una coincidencia de uso de longitud y mayúscula/minúscula. Si las dos cadenas son idénticas, el renglón es verdadero. Si hay diferencias, el renglón es falso.

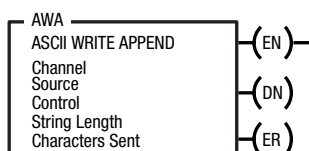
Ejemplo:



Una longitud de cadena no válida causa que el procesador establezca el bit de fallo (S:17/8) y el renglón es falso.

ASCII Write with Append (AWA)

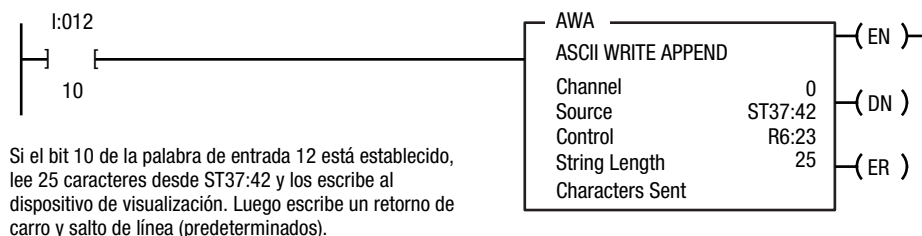
Descripción: Use la instrucción AWA para escribir caracteres desde el origen hacia un dispositivo de visualización. Esta instrucción de caracteres añadidos añade 1 ó 2 caracteres (los cuales usted configura en la configuración del canal). El valor predeterminado es un retorno de carro y salto de línea añadidos al fin de la cadena. Puede usar esta instrucción con el puerto en serie en el modo usuario o sistema.



Cómo introducir los parámetros

Debe proporcionar la siguiente información para usar la instrucción AWA:

Parámetro:	Definición:
Canal	el número del puerto RS-232. (El único valor válido es 0.)
Origen	la cadena que desea escribir.
Control	la dirección del elemento de archivo de control usado para los bits de estado de control.
Longitud de cadena	el número máximo de caracteres que desea escribir desde la cadena de origen (0 a 82). Si introduce 0, se escribe toda la cadena.
Caracteres transmitidos	el número de caracteres que el procesador transmitió a la zona de visualización (0 a 82). Este campo se actualiza solamente después de transmitirse toda la cadena (no se almacena un total corriente para cada carácter). Este campo es de visualización solamente.

Ejemplo:

Cuando el renglón cambia de falso a verdadero, el bit de habilitación del elemento de control (.EN) se establece. La instrucción se pone en la cola la instrucciones ASCII, el bit .EU se establece y el escán de programa continúa. La instrucción se ejecuta en paralelo con el escán de programa.

Veinticinco caracteres a partir del inicio de la cadena ST37:42 se transmiten al dispositivo de visualización y luego se transmiten los caracteres añadidos configurados por el usuario. El bit de efectuado se establece y un valor de 27 se envía a la palabra de posición.

Cuando el programa escanea la instrucción y encuentra el bit .DN establecido, el procesador establece el bit .EM para que sirva como bit de efectuado secundario correspondiente al escán de programa.

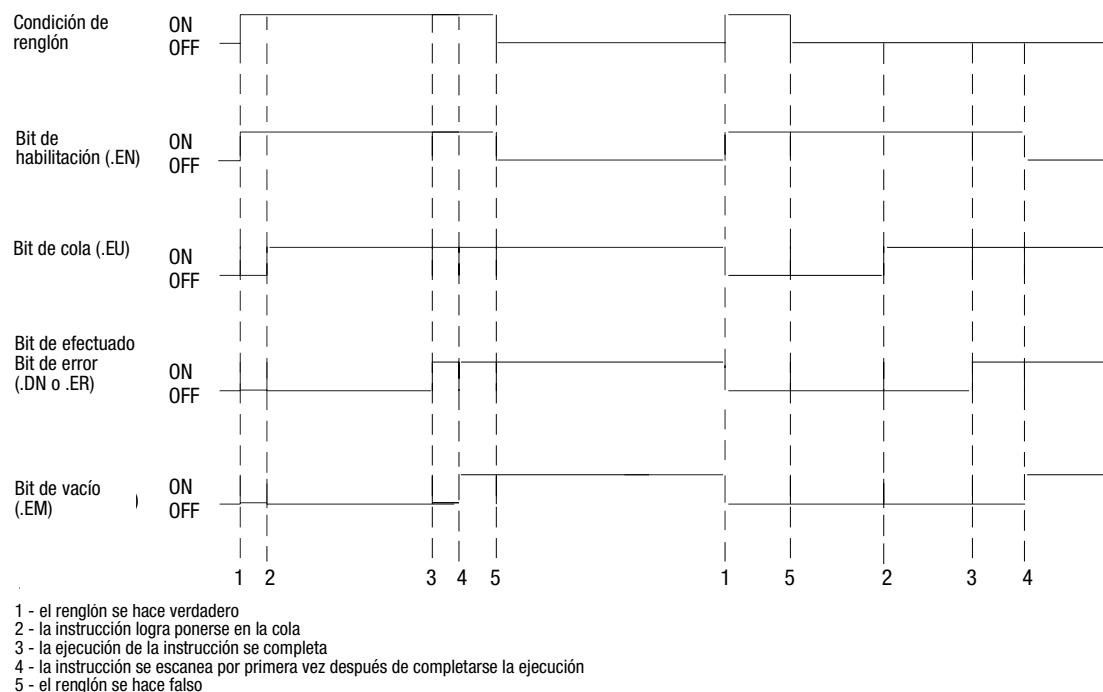
Puede usar el bit .UL para terminar una instrucción AWA antes de que concluye (por ejemplo, puede ser que desee terminar la instrucción si sabe que el dispositivo ASCII conectado al puerto no puede aceptar datos o si la conexión se interrumpe después de que la instrucción comienza a ejecutarse). Establezca el bit .UL en la estructura de control (el bit .ER se establece).

Importante: Cuando usted establece el bit .UL, la instrucción no se termina inmediatamente. El proceso puede tomar varios segundos.

Si una instrucción AWA comienza a ejecutarse con el bit .UL ya establecido, la instrucción se cancela inmediatamente.

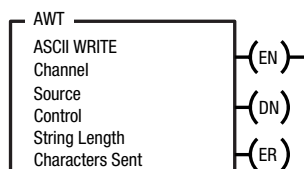
El bit de error (.ER) se establece durante la ejecución de la instrucción si la instrucción se cancela debido al cambio de modo del procesador o si el módem se pierde (cuando se usa el control de módem). Si el módem ya se había perdido, la instrucción continúa ejecutándose.

Figure 17.3
Ejemplo de un diagrama de temporización AWA



ASCII Write (AWT)

Descripción:

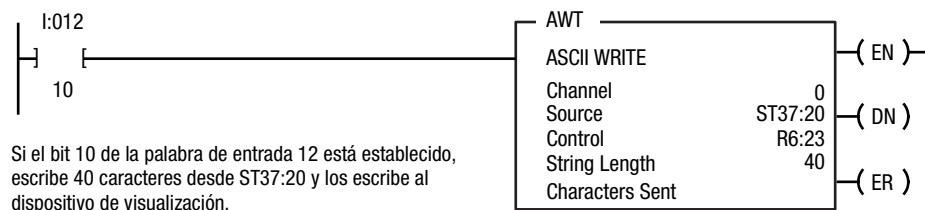


Use la instrucción AWT para escribir caracteres desde el origen hacia un dispositivo de visualización. Para repetir la instrucción, el renglón debe ir de falso a verdadero. Puede usar esta instrucción con el puerto en el modo sistema o usuario.

Cómo introducir los parámetros

Debe proporcionar la siguiente información para usar la instrucción AWT:

Parámetro:	Definición:
Canal	el número del puerto RS-232. (El único valor válido es 0.)
Origen	la cadena que desea escribir.
Control	la dirección del elemento de archivo de control usado para los bits de estado de control.
Longitud de cadena	el número máximo de caracteres que desea escribir desde la cadena de origen (0 a 82). Si introduce 0, se escribe toda la cadena.
Caracteres transmitidos	el número de caracteres que el procesador transmitió a la zona de visualización (0 a 82). Este campo se actualiza solamente después de transmitirse toda la cadena (no se almacena un total corriente para cada carácter). Este campo es de visualización solamente.

Ejemplo:

Cuando el renglón va de falso a verdadero, el bit de habilitación del elemento de control (.EN) se establece. La instrucción se pone en la cola de instrucciones ASCII, el bit .EU se establece y el escán del programa continúa. La instrucción se ejecuta en paralelo con el escán del programa.

Cuarenta caracteres de la cadena ST37:20 se envía a través del canal 0. El bit de efectuado se establece y un valor de 40 se envía a la palabra de posición.

Cuando el programa escanea la instrucción y encuentra el bit .DN establecido, el procesador establece el bit .EM para que sirva como bit de efectuado secundario correspondiente al escán de programa.

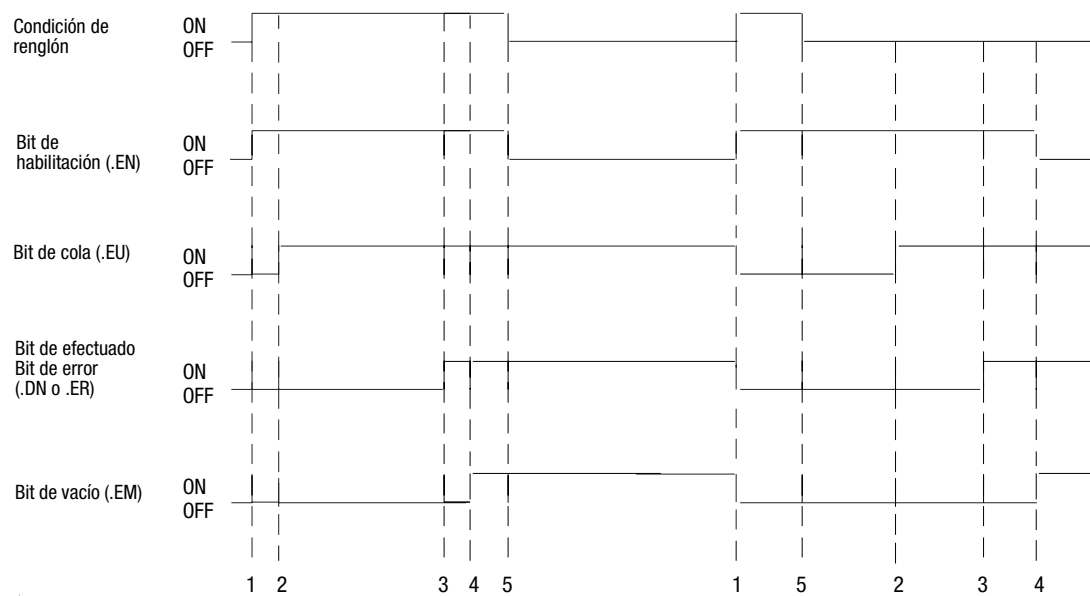
Puede usar el bit .UL para terminar una instrucción AWT antes de que concluye (por ejemplo, puede ser que desee terminar la instrucción si sabe que el dispositivo ASCII conectado al puerto no puede aceptar datos o si la conexión se interrumpe después de que la instrucción comienza a ejecutarse). Establezca el bit .UL en la estructura de control (el bit .ER se establece).

Importante: Cuando usted establece el bit .UL, la instrucción no se termina inmediatamente. Es posible que el proceso tome varios segundos.

Si una instrucción AWT comienza a ejecutarse con el bit .UL ya establecido, la instrucción se cancela inmediatamente.

El bit de error (.ER) se establece durante la ejecución de la instrucción si la instrucción se cancela debido al cambio de modo del procesador o si el módem se pierde (cuando se usa el control de módem). Si el módem ya se había perdido, la instrucción continúa ejecutándose.

Figura 17.4
Ejemplo de un diagrama de temporización AWT



- 1 - el renglón se hace verdadero
- 2 - la instrucción logra ponerse en la cola
- 3 - la ejecución de la instrucción se completa
- 4 - la instrucción se escanea por primera vez después de completarse la ejecución
- 5 - el renglón se hace falso

Notas:

Instrucciones de rutina de aplicación personalizada SDS, DFA

Objetivos del capítulo

Este capítulo presenta las instrucciones de rutina de aplicación personalizada (CAR) (SDS y DFA) para el software de programación PLC-5. Se necesita el software de rutina de aplicación personalizada (CAR) para usar estas instrucciones.

Para obtener información acerca de:

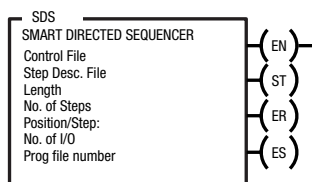
Vea:

Utilidades CAR SDS o DFA	Manual del usuario de control distribuido diagnóstico y de máquina
AGA3	Manual de programación de rutina de aplicación personalizada de flujo en masa AGA PLC-5
AGA7	Manual del usuario de CAR de flujo volumétrico PLC-5 para la medición de turbina y de desplazamiento
NX19	Manual del usuario de CAR de flujo volumétrico PLC-5 para la medición de orificios
API	Manual del usuario de CAR de flujo volumétrico PLC-5 para la medición de turbina y de desplazamiento

Vea el Apéndice C para obtener más información acerca de los operandos (y los tipos/valores de datos válidos de cada operando) usados por las instrucciones descritas en este capítulo.

Smart Directed Sequencer (SDS)

Descripción general



La instrucción del secuenciador dirigido inteligente (SDS) proporciona el control de estado que se puede usar para definir condiciones normales y anormales.

La instrucción SDS permite dos tipos básicos de ecuaciones lógicas:

- transicionales
- de combinación

Este tipo de ecuación lógica:

Hace lo siguiente:

Transicional	proporciona el control tradicional basado en el estado. Este tipo de instrucción SDS se construye alrededor del concepto de transición de estado en que cada transición de entrada dirige la instrucción a un próximo estado único usando una estructura lógica OR. Un cambio de entrada dirige la instrucción al paso A y la otra al paso B, etc.
De combinación	proporciona la función AND a las entradas además de la función OR usada en las ecuaciones de transición. Esto permite que las combinaciones complejas se procesen más fácilmente dentro de las estructuras SDS con un número mínimo de pasos.

Cómo programar la instrucción SDS

Es necesario hacer lo siguiente para programar la instrucción SDS:

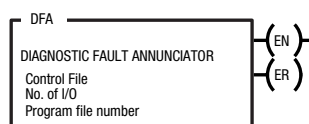
- descargar el CAR SDS
- introducir la instrucción SDS
- introducir la información de configuración
- introducir la información de E/S

Importante: No se puede usar los tipos de datos BT, PD, MG, ST o SC dentro de la lista de E/S de la instrucción SDS.

Importante: Cuando introduzca los operandos Control File y Step Desc. File, asegúrese de que los números de archivo (por ejemplo, 7, 10) no sean los mismos.

Vea el Manual del usuario de control distribuido diagnóstico y de máquinas para obtener más información acerca de la instrucción SDS.

Descripción general del anunciador de fallo diagnóstico (DFA)



La instrucción del anunciador de fallo diagnóstico (DFA) monitorea las entradas que usted define, pero no puede controlar las salidas. Las entradas válidas pueden ser:

- puntos de almacenamiento, tales como bits binarios
- bits de efectuado de contador / temporizador
- salidas (reales o lógicas)
- cualquier dirección de bit válida
- indicadores de nivel de lubricante
- alarmas
- bits de fallo establecidos por otro dispositivo (como un controlador de movimiento IMC) o por la lógica de escalera

Puede usar la instrucción DFA para generar mensajes cuando ocurre un fallo. Además, puede crear otros tipos de mensajes operacionales y diagnósticos con la instrucción DFA, tales como los mensajes de cambio de herramienta e instrucciones de operación.

Cómo programar la instrucción DFA

Es necesario hacer lo siguiente para programar la instrucción DFA:

- descargar el CAR DFA
- introducir la instrucción DFA
- introducir la información de configuración
- introducir la información de E/S

Vea el Manual del usuario de control distribuido diagnóstico y de máquinas para obtener más información acerca de la instrucción DFA.

Notas:

Requisitos de temporización y memoria de la instrucción

Requisitos de temporización y memoria de la instrucción

El tiempo que necesita un procesador para escanear una instrucción depende del tipo de instrucción, el tipo de direccionamiento, el tipo de datos, si la instrucción debe convertir datos y si la instrucción es verdadera o falsa.

Los cálculos de los requisitos de temporización y memoria en este capítulo presuponen lo siguiente:

- direccionamiento directo
- datos enteros (las excepciones se indican)
- ningunas conversiones del tipo de datos
- direcciones dentro de las 4098 primeras palabras de la tabla de datos para los procesadores PLC-5 clásicos; direcciones dentro de las 2048 primeras palabras para los procesadores PLC-5 con características mejoradas
- tiempos de ejecución indicados en μ s

Los requisitos de memoria se refieren al número de palabras que la instrucción usa. En algunos casos, una instrucción puede tener requisitos de rango de memoria. El rango de palabras existe porque la instrucción puede usar diferentes tipos de datos y modos de direccionamiento.

Las tablas se organizan según los requisitos de tiempos y memoria de la instrucción específicos a cada procesador.

Si usa este procesador:	Vea la página:
PLC-5 con características mejoradas, serie C:	A-2
Instrucciones de bit y palabra	A-5
Instrucciones de archivo	
PLC-5 clásico (todas las series):	
Instrucciones de bit y palabra	A-10
Instrucciones de archivo	A-13

Temporización para los procesadores PLC-5 con características mejoradas

Instrucciones de bit y palabra

La Tabla A.A muestra los requisitos de temporización y memoria para las instrucciones de bit y palabra de los procesadores PLC-5 con características mejoradas.

Tabla A.A
Requisitos de temporización y memoria para las instrucciones de bit y palabra (procesadores PLC-5 con características mejoradas solamente)

Categoría	Código	Título	Tiempo de ejecución No. entero (μs)		Tiempo de ejecución (μs) Punto flotante (coma flotante)		Palabras de memoria ¹
			Verdadero	Falso	Verdadero	Falso	
Relé	XIC	examine if closed	.32	.16			1 ²
	XIO	examine if open	.32	.16			1 ²
	OTL	output latch	.48	.16			1 ²
	OTU	output unlatch	.48	.16			1 ²
	OTE	output energize	.48	.48			1 ²
Bifurcación		branch end	.16	.16			1
		next branch	.16	.16			1
		branch start	.16	.16			1
Temporizador y contador	TON	timer on (0.01 base) (base 1.0)	3.8	2.6			2-3
			4.1	2.5			
	TOF	timer off (0.01 base) (base 1.0)	2.6	3.2			2-3
			2.6	3.2			
	RTO	retentive timer on (base 0.01) (base 1.0)	3.8	2.4			2-3
			4.1	2.3			
	CTU	count up	3.4	3.4			2-3
	CTD	count down	3.3	3.4			2-3
	RES	reset	2.2	1.0			2-3

(Continúa)

1. Use el número mayor para las direcciones que contienen más de 2048 palabras en la tabla de datos del procesador.

2. Para cada dirección de bit mayor que las 256 primeras palabras de memoria en la tabla de datos, añada 0.16 ms y 1 palabra de memoria.

Categoría	Código	Título	Tiempo de ejecución (μs) Número entero		Tiempo de ejecución (μs) Punto flotante (coma flotante)		Palabras de memoria ¹
			Verdadero	Falso	Verdadero	Falso	
Aritmético	ADD	add	6.1	1.4	14.9	1.4	4-7
	SUB	subtract	6.2	1.4	15.6	1.4	4-7
	MUL	multiply	9.9	1.4	18.2	1.4	4-7
	DIV	divide	12.2	1.4	23.4	1.4	4-7
	SQR	square root	9.9	1.3	35.6	1.3	3-5
	NEG	negate	4.8	1.3	6.0	1.3	3-5
	CLR	clear	3.4	1.1	3.9	1.1	2-3
	AVE	average file	152+E25.8	30	162+E22.9	36	4-7
	STD	standard deviation	321+E84.3	34	329+E77.5	34	4-7
	TOD	convert to BCD	7.8	1.3			3-5
	FRD	convert from BCD	8.1	1.3			3-5
	RAD	radian	57.4	1.4	50.1	1.4	3-5
	DEG	degree	55.9	1.4	50.7	1.4	3-5
	SIN	sine			414	1.4	3-5
	COS	cosine			404	1.4	3-5
	TAN	tangent			504	1.4	3-5
	ASN	inverse sine			426	1.4	3-5
	ACS	inverse cosine			436	1.4	3-5
	ATN	inverse tangent			375	1.4	3-5
	LN	natural log	409	1.4	403	1.4	3-5
	LOG	log	411	1.4	403	1.4	3-5
	XPY	X to the power of Y	897	1.5	897	1.5	4-7
	SRT	sort file					3-5
		(5/11, -5/20) (-5/30, -5/40, -5/60, -5/80)	276 + 12[E**1.34] 224 + 25[E**1.34]	227 189	278 + 16[E**1.35] 230 + 33[E**1.35]	227 189	

(Continúa)

1. Use el número mayor para las direcciones que contienen más de 2048 palabras en la tabla de datos del procesador.

E = Número de elementos procesados por escán.

SRT verdadero es una aproximación solamente. El tiempo real depende de la naturaleza aleatoria de los números.

Categoría	Código	Título	Tiempo de ejecución (μs) Número entero		Tiempo de ejecución (μs) Punto flotante (coma flotante)		Palabras de memoria ¹
			Verdadero	Falso	Verdadero	Falso	
Lógica	AND	and	5.9	1.4			4-7
	OR	or	5.9	1.4			4-7
	XOR	exclusive or	5.9	1.4			4-7
	NOT	no	4.6	1.3			3-5
Mover	MOV	move	4.5	1.3	5.6	1.3	3-5
	MVM	masked move	6.2	1.4			4-7
	BTD	bit distributor	10.0	1.7			6-9
Comparación	EQU	equal	3.8	1.0	4.6	1.0	3-5
	NEQ	not equal	3.8	1.0	4.5	1.0	3-5
	LES	less than	4.0	1.0	5.1	1.0	3-5
	LEQ	less than or equal	4.0	1.0	5.1	1.0	3-5
	GRT	greater than	4.0	1.0	5.1	1.0	3-5
	GEQ	greater than or equal	4.0	1.0	5.1	1.0	3-5
	LIM	limit test	6.1	1.1	8.4	1.1	4-7
	MEQ	mask compare if equal	5.1	1.1			4-7
Comparar	CMP	all	$2.48 + (\Sigma[0.8 + i])$	$2.16 + Wi[0.56]$	$2.48 + (\Sigma[0.8 + i])$	$2.16 + Wi[0.56]$	2+Wi
Calcular	CPT	all	$2.48 + (\Sigma[0.8 + i])$	$2.16 + Wi[0.56]$	$2.48 + (\Sigma[0.8 + i])$	$2.16 + Wi[0.56]$	2+Wi

1. Use el número mayor para las direcciones que contienen más de 2048 palabras en la tabla de datos del procesador.

i = El tiempo de ejecución de cada instrucción (operación, por ejemplo ADD, SUB, etc.) usada dentro de la expresión CMP o CPT

Wi = El número de palabras de memoria usadas por la instrucción (operación, por ejemplo, ADD, SUB, etc.) dentro de la expresión CMP o CPT.

Las instrucciones CMP o CPT se calculan con el direccionamiento directo corto.

Instrucciones de archivo

Vea la Tabla A.B para obtener la temporización de instrucción para las instrucciones de archivo.

Tabla A.B
Requisitos de temporización y memoria para las instrucciones de archivo, control de programa y ASCII (procesadores PLC-5 con características mejoradas solamente)

Categoría	Código	Título	Tiempo (μs)		Tiempo (μs)		Palabras de memoria ¹
			Número entero Verdadero	Falso	Punto flotante (coma flotante) Verdadero	Falso	
aritmética y lógica de archivos	FAL	all	$11 + (\Sigma[2.3 + i])E$	$6.16 + Wi[0.16]$	$11 + (\Sigma[2.3 + i])E$	$6.16 + Wi[0.16]$	3-5 + Wi
Búsqueda y comparación de archivos	FSC	all	$11 + (\Sigma[2.3 + i])E$	$6.16 + Wi[0.16]$	$11 + (\Sigma[2.3 + i])E$	$6.16 + Wi[0.16]$	3-5 + Wi
Archivo	COP	copy	$16.2 + E[0.72]$	1.4	$17.8 + E[1.44]$	1.4	4-6
		counter, timer, and control	$15.7 + E[2.16]$	1.4			
	FLL	fill	$15.7 + E[0.64]$	1.5	$18.1 + E[0.80]$	1.5	4-6
		counter, timer, and control	$15.1 + E[1.60]$	1.5			
Registro de desplazamiento	BSL	bit shift left	$10.6 + B[0.025]$	5.2			4-7
	BSR	bit shift right	$11.1 + B[0.025]$	5.2			4-7
	FFL	FIFO load	8.9	3.8			4-7
	FFU	FIFO unload	$10.0 + E[0.43]$	3.8			4-7
	LFL	LIFO load	9.1	3.7			4-7
	LFU	LIFO unload	10.6	3.8			4-7
Diagnóstico	FBC	0 mismatch	$15.4 + B[0.055]$	2.9			6-11
		1 mismatch	$22.4 + B[0.055]$	2.9			
		2 mismatches	$29.9 + B[0.055]$	2.9			
	DDT	0 mismatch	$15.4 + B[0.055]$	2.9			6-11
		1 mismatch	$24.5 + B[0.055]$	2.9			
		2 mismatches	$34.2 + B[0.055]$	2.9			
	DTR	data transitional	5.3	5.3			4-7

(Continúa)

1. Use el número mayor para las direcciones que contienen más de 2048 palabras en la tabla de datos del procesador.

i = El tiempo de ejecución de cada instrucción (operación, por ejemplo ADD, SUB, etc.) usada dentro de la expresión CMP o CPT

E = El número de elementos procesados por escán

B = El número de bits afectados por escán

Wi = El número de palabras de memoria usadas por la instrucción (operación, por ejemplo, ADD, SUB, etc.) dentro de la expresión CMP o CPT.

Las instrucciones FAL o FSC se calculan con el direccionamiento directo corto

Categoría	Código	Título	Tiempo (µs)		Tiempo (µs)		Palabras de memoria ¹
			Número entero Verdadero	Falso	Punto flotante (coma flotante) Verdadero	Falso	
Secuen- ciador	SQL	sequencer input	7.9	1.3			5-9
	SQL	sequencer load	7.9	3.5			4-7
	SQO	sequencer output	9.7	3.7			5-9
E/S inmediatas ²	IIN	immediate input (-5/11, -5/20) (-5/30, -5/40, -5/60, -5/80)	357 307	1.1			2
	IOT	immediate output (-5/11, -5/20) (-5/30, -5/40, -5/60, -5/80)	361 301	1.1			2
	IDI	immediate data input (-5/20C) (-5/40C, -5/60C y -5/80C)	200 + 1.4 (para cada palabra) 200 + 1.4 (para cada palabra)	1.1			4-7
	IDO	immediate data output (-5/20C) — (-5/40C, -5/60C y -5/80C)	230 + 1.4 (para cada palabra) 250 + 1.7 (para cada palabra)	1.1			4-7
Control de zona	MCR	master control	0.16	0.16			1
Control de programa	JMP	jump	8.9 + (número de archivo – 2) * 0.96	1.4 + (número de archivo – 2) * 0.96			2
	LBL	label	0.32	0.32			2+posición en la tabla de etiquetas
	JSR ³ / RET	jumb to subroutine/ return PLC-5/11, -5/20, -5/30, -5/40, -5/40L, -5/60, -5/60L, -5/20E, -5/40E					3+ parámetros/ JSR 1+ parámetros/ RET
		— 0 parameters	12.3	1.0	n/a	n/a	
		— 1 parameter	16.1	1.0	17.3	1.0	
		— increase/parameter	3.8	n/a	5.0	n/a	
		PLC-5/80					
		— 0 parameters	315	1.0			
		— 1 parameter	340	1.0	349		
		— increase/parameter	31	n/a	33	1.0	

(Continúa)

1. Use el número mayor para las direcciones que contienen más de 2048 palabras en la tabla de datos del procesador.

2. La temporización para las instrucciones de E/S inmediatas es el tiempo necesario para que la instrucción se ponga en la cola para el procesamiento

3. Calcule los tiempos de ejecución así: (tiempo) + (cantidad de parámetros adicionales)(tiempo/parámetro). Por ejemplo, si pasa 3 parámetros de número entero en una JSR dentro de un procesador PLC-5/11, el tiempo de ejecución = 16.1 + (2)(3.8) = 23.7 µs.

B = El número de bits afectados por escán

Categoría	Código	Título	Tiempo (μs) Número entero		Tiempo (μs) Punto flotante (coma flotante)		Palabras de memoria ¹
			Verdadero	Falso	Verdadero	Falso	
Control de programa	SBR	0 parameters	12.3	1.0			1+ parameters
		1 parameter	16.1	1.0	17.3	1.0	
		increase/parameter	3.8		5.0		
	END	end	negligible				1
	TND	temporary end					1
	EOT	end of transition					1
	AFI	always false	0.16	0.16			1
	ONS	one shot	3.0	3.0			2-3
	OSR	one shot rising	6.2	6.0			4-6
	OSF	one shot falling	6.2	5.8			4-6
	FOR/ NXT	for next loop (PLC-5/80)	8.1 + L[15.9] (151 + L[277])	5.3 + N[0.75] (152 + N[6.1])			FOR 5-9 NXT 2
	BRK	break	11.3 + N[0.75]	0.9			1
	UID	user interrupt disable		1.0			1
		(-5/11, -5/20)	175				
		(-5/30, -5/40, -5/60, -5/80)	119				
	UIE	user interrupt enable		1.0			1
		(-5/11, -5/20)	170				
		(-5/30, -5/40, -5/60, -5/80)	100				

(Continúa)

1. Use el número mayor para las direcciones que contienen más de 2048 palabras en la tabla de datos del procesador.

L = El número de lazos FOR/NXT

N = El número de palabras en la memoria entre FOR/NXT o BRK/NXT

Categoría	Código	Título	Tiempo (μs) Número entero Verdadero	Falso	Tiempo (μs) Punto flotante (coma flotante) Verdadero	Falso	Palabras de memoria ¹
Control de proceso	PID	PID loop control					5-9
Ganancias		Independent (-5/11, -5/20, -5/20E, -5/20C)	462	3.0	882	58	
		(-5/30, -5/40, -5/40E, -5/40C, -5/40L, -5/60, -5/60C, -5/60L, -5/80, -5/80E, -5/80C)	655				
		ISA (-5/11, -5/20, -5/20E, -5/20C)	560		1142		
		(-5/30, -5/40, -5/40E, -5/40C, -5/40L, -5/60, -5/60C, -5/60L, -5/80, -5/80E, -5/80C)	895				
Modos		Manual (-5/11, -5/20, -5/20E, -5/20C)	372		900		
		(-5/30, -5/40, -5/40E, -5/40C, -5/40L, -5/60, -5/60C, -5/60L, -5/80, -5/80E, -5/80C)	420				
		Set output (-5/11, -5/20, -5/20E, -5/20C)	380		882		
		(-5/30, -5/40, -5/40E, -5/40C, -5/40L, -5/60, -5/60C, -5/60L, -5/80, -5/80E, -5/80C)	440				
Cascada		Slave			1286		
		Master			840		
ASCII ²	ABL ²	test buffer for line (-5/11, -5/20)		214			3-5
		(-5/30, -5/40, -5/60, -5/80)	316 388	150			
	ACB ²	no. of characters in buffer (-5/11, -5/20)	316	214			3-5
		(-5/30, -5/40, -5/60, -5/80)	389	150			
	ACI	string to integer (-5/11, -5/20)	220 + C[11]	1.4			3-5
		(-5/30, -5/40, -5/60, -5/80)	140 + C[21.4]				

(Continúa)

1. Use el número mayor para las direcciones que contienen más de 2048 palabras en la tabla de datos del procesador.

2. La temporización para las instrucciones ASCII es el tiempo necesario para que la instrucción se ponga en la cola para el procesamiento en el canal 0.

Categoría	Código	Título	Tiempo (μs)		Tiempo (μs)		Palabras de memoria ¹
			Número entero Verdadero	Falso	Punto flotante (coma flotante) Verdadero	Falso	
ASCII ²	ACN	string concatenate (-5/11, -5/20) (-5/30, -5/40, -5/60, -5/80)	237 + C[2.6] 179 + C[5.5]	1.9			4-7
	AEX	string extract (-5/11, -5/20) (-5/30, -5/40, -5/60, -5/80)	226 + C[1.1] 159 + C[2.2]	1.9			5-9
	AHL ²	set or rest lines (-5/11, -5/20) (-5/30, -5/40, -5/60, -5/80)	318 526	213 157			5-9
	AIC	integer to string (-5/11, -5/20) (-5/30, -5/40, -5/60, -5/80)	260 270	1.4			3-5
	ARD ²	read characters (-5/11, -5/20) (-5/30, -5/40, -5/60, -5/80)	315 380	214 149			4-7
	ARL ²	read line (-5/11, -5/20) (-5/30, -5/40, -5/60, -5/80)	316 388	214 151			4-7
	ASC	string search (-5/11, -5/20) (-5/30, -5/40, -5/60, -5/80)	222 + C[1.7] 151 + C[3.0]	1.9			5-9
	ASR	string compare (-5/11, -5/20) (-5/30, -5/40, -5/60, -5/80)	234 + C[1.3] 169 + C[2.4]	202 119			3-5
	AWA ²	write with append (-5/11, -5/20) (-5/30, -5/40, -5/60, -5/80)	319 345	215 154			4-7
	AWT ²	write (-5/11, -5/20) (-5/30, -5/40, -5/60, -5/80)	318 344	215 151			4-7

1. Use el número mayor para las direcciones que contienen más de 2048 palabras en la tabla de datos del procesador.

2. La temporización para las instrucciones ASCII es el tiempo necesario para que la instrucción se ponga en la cola para el procesamiento en el canal 0.

C = Número de caracteres ASCII

Temporización para los procesadores PLC-5 clásicos

Instrucciones de bit y palabra

La Tabla A.C indica los requisitos de temporización y memoria para las instrucciones de bit y palabra (procesadores PLC-5 clásicos)

Tabla A.C
Requisitos de temporización y memoria para las instrucciones de bit y palabra (procesadores PLC-5 clásicos)

Categoría	Código	Título	Tiempo de ejecución (μs) Número entero		Tiempo de ejecución (μs) punto flotante (coma flotante)		Palabras de memoria ¹
			Verdadero	Falso	Verdadero	Falso	
Relé	XIC	examine if closed	1.3	0.8			1 ²
	XIO	examine if open	1.3	0.8			1 ²
	OTL	output latch	1.6	0.8			1 ²
	OTU	output unlatch	1.6	0.8			11
	OTE	output energize	1.6	1.6			1 ²
Bifurcación		branch end	0.8	0.8			1
		next branch	0.8	0.8			1
		branch start	0.8	0.8			1
Temporizador y contador	TON	timer on (base 0.01) (base 1.0)	39	27			2-3
			44	28			
	TOF	timer off (0.01 base) (1.0 base)	30	43			2-3
			30	51			
	RTO	retentive timer on (base 0.01) (base 1.0)	39	24			2-3
			44	24			
	CTU	count up	32	34			2-3
	CTD	count down	34	34			2-3
	RES	reset	30	14			2-3

¹ Use el número menor si todas las direcciones son menores que la palabra 4096. Use el número mayor si todas las direcciones son mayores que 4096.

² Para cada dirección de bit mayor que las 256 primeras palabras de memoria en la tabla de datos, añada 0.89 μs al tiempo de ejecución y 1 palabra de memoria a los requisitos.

(Continúa)

Categoría	Código	Título	Tiempo de ejecución (μs) Número entero		Tiempo de ejecución (μs) Punto flotante (coma flotante)		Palabras de memoria ¹
			Verdadero	Falso	Verdadero	Falso	
Aritmético	ADD	add	36	14	92	14	4-7
	SUB	subtract	36	14	92	14	4-7
	MUL	multiply	41	14	98	14	4-7
	DIV	divide	49	14	172	14	4-7
	SQR	square root	82	14	212	14	3-5
	NEG	negate	28	14	36	14	3-5
	CLE	clear	18	14	23	14	2-3
	TOD	convert to BCD	52	14			3-5
	FRD	convert from BCD	44	14			3-5
Lógico	AND	and	36	14			4-7
	OR	or	36	14			4-7
	XOR	exclusive or	36	14			4-7
	NOT	not	27	14			3-5
Mover	MOV	move	26	14	35	14	3-5
	MVM	masked move	55	14			6-9
Comparación	EQU	igual	32	14	42	14	3-5
	NEQ	not equal	32	14	42	14	3-5
	LES	less than	32	14	42	14	3-5
	LEQ	less than or equal	32	14	42	14	3-5
	GRT	greater than	32	14	42	14	3-5
	GEQ	greater than or equal	32	14	42	14	3-5
	LIM	limit test	42	14	60	14	4-7
	MEQ	mask compare if equal	41	14			4-7

¹ Use el número menor si todas las direcciones son menores que la palabra 4096. Use el número mayor si todas las direcciones son mayores que 4096.

(Continúa)

Categoría	Código	Título	Tiempo de ejecución (μs)		Tiempo de ejecución (μs)		Palabras de memoria ¹
			Número entero		Punto flotante (coma flotante)		
			Verdadero	Falso	Verdadero	Falso	
Cálculo	CPT	add	67	34	124	34	6-9
		subtract	67	34	124	34	6-9
		multiply	73	34	130	34	6-9
		divide	80	34	204	34	6-9
		square root	113	33	244	34	5-7
		negate	59	33	68	34	5-7
		clear	49	30	55	34	4-5
		move	58	33			5-7
		convert to BCD	84	33			5-7
		convert from BCD	75	33			5-7
		AND	68	34			6-9
		OR	68	34			6-9
		XOR	68	34			6-9
		NOT	59	34			5-7
Comparación	CMP	equal	63	34	73	34	5-7
		not equal	63	34	73	34	5-7
		less than	63	34	73	34	5-7
		less than or equal	63	34	73	34	5-7
		greater than	63	34	73	34	5-7
		greater than or equal	63	34	73	34	5-7

¹Use el número menor si todas las direcciones son menores que la palabra 4096. Use el número mayor si todas las direcciones son mayores que 4096.

Instrucciones de archivo

La temporización de instrucción para las instrucciones depende del tipo de datos, número de archivos procesados por escán, número de elementos procesadores por escán y si la instrucción convierte los datos entre los formatos de número entero y punto flotante (coma flotante).

- para la conversión de número entero en punto flotante (coma flotante) añadida:
8 μ s para cada dirección de elemento
10 μ s para cada dirección de archivo (prefijo de #)
- para la conversión de punto flotante (coma flotante) en número entero añadida:
33 μ s para cada dirección de elemento
44 μ s para cada dirección de archivo (prefijo de #)

Tabla A.D
Requisitos de temporización y memoria para las instrucciones de archivo
(procesadores PLC-5 clásicos)

Categoría	Código	Título	Tiempo (μ s) Número entero	Tiempo (μ s) Punto flotante (coma flotante)	Tiempo (μ s) No. entero o punto flotante (coma flotante)	Palabras de memoria ¹
			Verdadero	Verdadero	Falso	
Aritmética y lógica de archivos	FAL	add	$98 + W[36.7 + N]$	$98 + W[95.1 + N]$	54	7-12
		subtract	$98 + W[36.7 + N]$	$98 + W[95.1 + N]$	54	7-12
		multiply	$98 + W[42.5 + N]$	$98 + W[101.2 + N]$	54	7-12
		divide	$98 + W[51.1 + N]$	$98 + W[180.3 + N]$	54	7-12
		square root	$98 + W[84.7 + N]$	$98 + W[220.5 + N]$	54	6-10
		negate	$98 + W[29.2 + N]$	$98 + W[37.2 + N]$	54	6-10
		clear	$98 + W[18.4 + N]$	$98 + W[24.0 + N]$	54	5-8
		move	$98 + W[27.3 + N]$	$98 + W[36.2 + N]$	54	6-10
		convert to BCD	$98 + W[54.3 + N]$		54	6-10
		convert from BCD	$98 + W[45.4 + N]$		54	6-10

¹ Use el número menor si todas las direcciones son menores que la palabra 4096. Use el número mayor si todas las direcciones son mayores que 4096.

W = Número de elementos afectados por escán

N = 2 x (número de direcciones de archivo entero) + 8 x (número de direcciones de archivo de punto flotante [coma flotante]) + 6 x (número de direcciones de archivo de temporizador, contador o control) + (número de conversiones entre los formatos de número entero y punto flotante [coma flotante])

(Continúa)

Categoría	Código	Título	Tiempo (μs) Número entero	Tiempo (μs) Punto flotante (coma flotante)	Tiempo (μs) No. entero o punto flotante (coma flotante)	Palabras de memoria ¹
			Verdadero	Verdadero	Falso	
Aritmética y lógica de archivos		AND	$98 + W[37.2 + N]$		54	7-12
		OR	$98 + W[37.2 + N]$		54	7-12
		XOR	$98 + W[37.2 + N]$		54	7-12
		NOT	$98 + W[28.2 + N]$		54	6-10
Búsqueda y comapración de archivos	FSC	all comparisons	$93 + W[32.7 + N]$	$93 + W[43.3 + N]$	54	6-10
Archivo	COP	copy	$88 + 2.7W$	$104 + 3.8W$	20	4-7
		counter, timer, and control	$98 + 5.8W$			
	FLL	fill	$81 + 2/.1 W$	$100 + 3.1W$	15	4-7
		counter, timer, and control	$97 + 4.4W$			
Registro de desplazamiento	BSL	bit shift left	$74 + 3.4W$		57	4-7
	BSR	bit shift right	$78 + 3.0W$		57	4-7
	FFL	FIFO load	54		44	4-7
	FFU	FIFO unload	$68 + 3.2W$		46	4-7
Diagnóstico	FBC	file bit compare				6-11
		0 mismatch	$75 + 6W$		31	
		1 mismatch	$130 + 6W$		31	
		2 mismatches	$151 + 6W$		31	
	DDT	diagnostic detect				6-11
		0 mismatch	$71 + 6W$		31	
		0 mismatch	$150 + 6W$		31	
		2 mismatches	$161 + 6W$			

¹ Use el número menor si todas las direcciones son menores que la palabra 4096. Use el número mayor si todas las direcciones son mayores que 4096.

W = Número de elementos afectados por escán

N = 2 x (número de direcciones de archivo entero) + 8 x (número de direcciones de archivo de punto flotante [coma flotante]) + 6 x (número de direcciones de archivo de temporizador, contador o control) + (número de conversiones entre los formatos de número entero y punto flotante [coma flotante])

(Continúa)

Categoría	Código	Título	Tiempo (μs) Número entero	Tiempo (μs) Punto flotante (coma flotante)	Tiempo (μs) No. entero o punto flotante (coma flotante)	Palabras de memoria ¹
			Verdadero	Verdadero	Falso	
Control de zona	MCR	master control	12		18	1
E/S inmediatas	IIN	immediate input				2-3
		local	196		16	
		remote	204		16	
	IOT	immediate output				2-3
		local	202		16	
		remote	166		16	
Secuenciador	SQL	sequencer input	57		14	5-9
	SQL	sequencer load	55		42	4-7
	SQO	sequencer output	77		42	5-9
Salto y subrutina	JMP	jump	45		15	2-3
	JSR	jump to subroutine				
	SBR	0 parameters	56		15	2-3
		1 parameter	91		15	3-5
		add per parameter	21			
	RET	return from sub.				
		0 parameters	48		13	1
		1 parameter	70		13	2-3
		add per parameter	21			
	LBL	label	12		12	3

¹ Use el número menor si todas las direcciones son menores que la palabra 4096. Use el número mayor si todas las direcciones son mayores que 4096.

(Continúa)

Categoría	Código	Título	Tiempo (μs) Número entero	Tiempo (μs) Punto flotante (coma flotante)	Tiempo (μs) No. entero o punto flotante (coma flotante)	Palabras de memoria ¹
			Verdadero	Verdadero	Falso	
Varios	END	end	insignificante		insignificante	1
	TND	temporary end	insignificante		15	1
	AFI	always false	15		13	1
	ONS	one shot	28		30	2-3
	DTR	data transitional	41		41	4-7
	BTD	bit distributor	77		14	6-11
	PID	PID loop control	608		34	5-9
	BTR	block transfer read	Vea el capítulo 15			
	BTW	block transfer write				
	MSG	message	Vea el capítulo 16			

¹ Use el número menor si todas las direcciones son menores que la palabra 4096. Use el número mayor si todas las direcciones son mayores que 4096.

Constantes de programa

Use las constantes de programa en las instrucciones de comparación, cálculo y archivo para mejorar los tiempo de ejecución de la instrucción. Las constantes de números enteros y las constantes de punto flotante (coma flotante) se ejecutan en menos de 1 μ s.

Elementos directos o indirectos – Procesadores PLC-5 con características mejoradas

El tiempo de ejecución adicional para los elementos direccionados directa e indirectamente depende de la ubicación en la memoria, la referencia al inicio de todos los archivos de datos (archivo de salida, palabra 0), si los datos se almacenan en la dirección de origen o destino y si la instrucción convierte datos. La Tabla A.E indica los tiempos que se suman a los tiempos de ejecución de las estructuras.

Tabla A.E
Tiempo de ejecución adicional
(procesadores PLC-5 con características mejoradas)

Modo de direccionamiento	Tipo de datos	Modificador en μ seg (se añade para cada operando)
Directo	Número entero	0
	Punto flotante (coma flotante)	0
Índice	Número entero	1.1
	Punto flotante (coma flotante)	1.8
	Contador- temporizador-control	2.4
Inmediato	Número entero	0.24
	Punto flotante (coma flotante)	1.0
Indirecto		6.6 + W[0.09]
Punto flotante (coma flotante) a número entero		5.6
Número entero a punto flotante (coma flotante)		8.4

Elementos directos o indirectos – Procesadores PLC-5 clásicos

El tiempo de ejecución adicional para los elementos direccionados directamente depende de la ubicación en la memoria, la referencia al inicio de todos los archivos de datos (archivo de salida, palabra 0), si los datos se almacenan en la dirección de origen o destino y si la instrucción convierte datos. La Tabla A.F indica los tiempos que se suman a los tiempos de ejecución de las instrucciones.

Tabla A.F
Tiempo de ejecución adicional basado en las direcciones de origen y destino (procesadores PLC-5 clásicos)

Tipo de datos	Origen (número entero a punto flotante [coma flotante])			Destino (punto flotante [coma flotante] a número entero)		
	0-2 K	2-4 K	4 K+	0-2 K	2-4 K	4 K+
entero	0	1	2	0	1	2
punto flotante	0	3	4	0	3	4
conversión de datos	8	9	10	33	34	35

Añada lo siguiente cuando las direcciones de archivo (prefijo #) en la expresión o la dirección de destino contengan las direcciones indirectas para los números de archivo:

- 45 μ s cuando la dirección indirecta es de tipo de número entero
- 48 μ s cuando la dirección indirecta es de tipo de punto flotante (coma flotante)
- 48 μ s cuando la dirección indirecta es de tipo de temporizador, contador o control

Añada lo siguiente cuando las direcciones de archivo en la expresión o el destino contengan direcciones indirectas para los números de elemento:

- 45 μ s cuando la dirección directa es de tipo de número entero
- 46 μ s cuando la dirección directa es de tipo de punto flotante (coma flotante)
- 46 μ s cuando la dirección directa es de tipo de temporizador, contador o control

Si la dirección de archivo contiene dos direcciones indirectas, añada solamente un valor (el mayor). Por ejemplo, para #F[N7:20][N7:30], añada 48 μ s (dirección de archivo de punto flotante [coma flotante]).

Multiplique el tiempo adicional por el número de elementos en el archivo para cualquier tipo de archivo o dirección de archivo. Por ejemplo:

Expresión: #N[N7:100]:10 * F8:20
 añade 10 para convertir en punto flotante
 (coma flotante)
 añade 45 para la dirección indirecta

Destino: #N7:30
 añade 44 para convertir en número entero

multiplicación FAL:
 $98 + W[42.5 + N + \text{direccionamiento indirecto}]$
 $N = 2(2) + 8(1) + 6(0) + 10 + 44 = 66$
 $W = 16$

Tiempo de ejecución en el modo TODOS (ALL):
 $98 + 16[42.5 + 66 + 45]$
 2554 μs

Direcciones indirectas de bits o elementos – Procesadores PLC-5 clásicos

Los tiempos de ejecución adicionales para los bits y elementos direccionados indirectamente depende del número de direcciones indirectas en la dirección general. La Tabla A.G indica los tiempos adicionales.

Tabla A.G
Tiempo de ejecución adicionales para los bits y elementos direccionados indirectamente – Procesadores PLC-5 clásicos

Tipo de datos	Tiempo (μs) para archivo de variables o elemento	Tiempo (μs) para archivo de variables y elemento
Bit en archivo binario	57	60
Bit en archivo entero	60	63
Bit en archivo de temporizador, contador o control	64	66
Número entero (N)	42	42
Archivo de temporizador (T), contador (C) o control (R)	43	44
Punto flotante (coma flotante) (F)	61	64
Conversión de número entero en punto flotante (coma flotante)	71	77
Conversión de temporizador, contador o control en punto flotante (coma flotante)	85	81

Otras consideraciones asociadas con la temporización – Procesadores PLC-5 clásicos

La Tabla A.H indica otras consideraciones asociadas con la temporización.

Tabla A.H
Otras consideraciones asociadas con la temporización – Procesadores PLC-5 clásicos

Tareas	Tiempo (milisegundos)
Mantenimiento interno	4.5 máx
Escán de E/S locales residentes	1 por número de rack asignado
Escán de E/S remotas	10 por número de rack asignado a 57.6 Kb

Referencia SFC

Objetivos del apéndice

Use este apéndice para asegurarse de que el SFC satisface los requisitos del procesador y asegurarse de que el SFC funciona según las expectativas. Este apéndice describe lo siguiente:

- información de estado SFC en el archivo de estado del procesador
- asignación de memoria
- restricciones dinámicas
- secuencias de escán
- tiempos de ejecución

Información de estado SFC en el archivo de estado del procesador

La Tabla B.A indica las palabras y los bits en el archivo de estado (S) del procesador que contienen información SFC.

Tabla B.A
Palabras de estado SFC

Palabra:	Título:	Descripción:
S:1/15	Primera pasada	Establ.: El procesador inició el primer escán del próximo paso activo en la SFC Restabl.: El procesador completó el escán del paso activo actual
	Tiempo de escán actual de programa	El tiempo necesario para que el procesador escanee todos los pasos activos una sola vez Si usa múltiples programas de control principales en un procesador PLC-5 con características mejoradas, este tiempo es el total actual de un escán de todos los programas de control principales.
S:9	Tiempo de escán máximo de programa	El tiempo máximo necesario para que el procesador escanee todos los pasos activos una sola vez (palabras S:8) Si usa múltiples programas de control principales en un procesador PLC-5 con características mejoradas, este tiempo es el máximo de todos los totales anteriores. Este valor se mantiene hasta que el usuario lo restablece.
S:11/3	Fallo SFC	Establ.: El procesador detectó un fallo SFC y almacenó un código de fallo en la palabra 12 Restabl.: No hay fallo SFC
S:11/5	Fallo de encendido	Set: El procesador detectó un fallo de protección de encendido (vea la palabra 26, bit 1) Reset: No hay fallo. Se permite el encendido

(Continúa)

Palabra:	Título:	Descripción:	
S:12	Códigos de fallo	74	Fallo en el archivo SFC
		75	SFC tiene más de 24 pasos activos
		77	Falta archivo o es un archivo incorrecto para el tipo de paso, acción o transición
		78	La ejecución SFC no puede continuar después de la interrupción
		79	SFC no se puede ejecutar porque el PLC-5 no es compatible
S:13	Número de archivo con fallo	Contiene el número de archivo si ocurrió un fallo SFC	
S:14	Números de renglón con fallo	Contiene el número de renglón con fallo	
S:26/0 *	Reinicio/continuación	Establ.:	El procesador reinicia SFC en los pasos activos donde se detuvo debido a la pérdida de energía o cambio de modo del procesador
		Restabl.:	El procesador reinicia PLC en el primer paso
S:26/1 *	Protección de encendido después de pérdida de energía	Set:	La protección está habilitada. El procesador va a la rutina de fallo en el momento del encendido y el procesador establece la palabra 11, bit 5
		Restabl.:	La protección está inhabilitada. El procesador se enciende en el modo marcha
S:28 *	Punto de ajuste del temporizador de control (watchdog) de programa	El tiempo máximo (milisegundos) para escanear un solo paso a través de todos los pasos activos	
		Si usa múltiples programas de control principales en un procesador PLC-5 con características mejoradas, este tiempo es el total de un escán de todos los programas de control principales.	
S:79 * (excepto por el tiempo de escán) – S:127	Inhibición MCP, número de archivo y tiempo de escán	Información acerca de los múltiples programas de control principales individuales.	
		Procesadores PLC-5 con características mejoradas solamente.	

* Usted introduce valores para estas palabras/bits.

Asignación de memoria

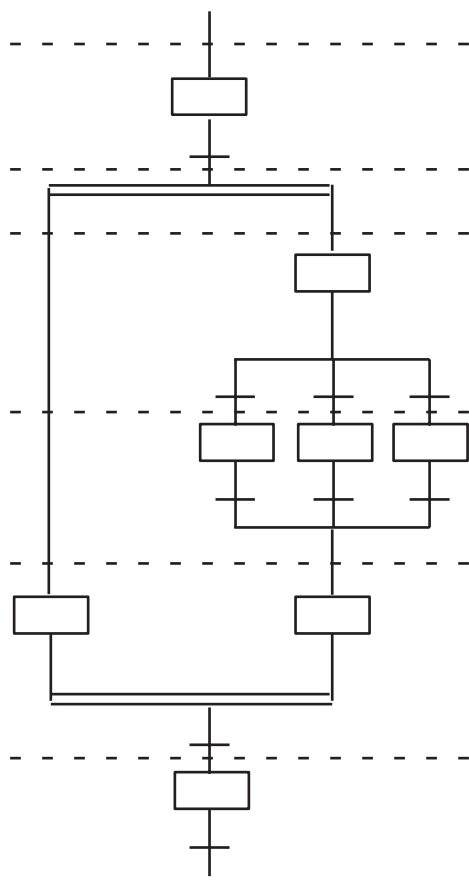
Los requisitos de memoria para la SFC dependen de las estructuras que usted use. La Tabla B.B indica el uso de palabras calculado para las estructuras SFC:

Tabla B.B
Uso de memoria SFC

Esta estructura:	Usa esta cantidad de memoria:	
	Procesador PLC-5 clásico	Procesador PLC-5 con características mejoradas
inicio y fin de programa	2 palabras	19 palabras
cada pareja de paso/transición	8 palabras	16 + 6a palabras a = número de acciones en el paso 6 palabras cada acción
cada bifurcación de selección	5n + 5 palabras n = número de bifurcaciones	11 + 6a + 7n a = número de acciones en el paso n = número de rutas de acceso
cada bifurcación simultánea con derivación	n + 1 palabra n = número de bifurcaciones	3n + 1 n = número de rutas de acceso
cada bifurcación simultánea convergente	n ² + 6n + 3 palabras n = número de bifurcaciones	5 + 11n + 6a a = número de acciones en todos los pasos convergentes para la bifurcación simultánea n = número de rutas de acceso
cada etiqueta o instrucción GOTO	1 palabra	1 palabra
cada compresión de diagrama	3 palabras	3 palabras

La Figura B.1 muestra un ejemplo de SFC y los requisitos de memoria calculados para la SFC.

Figura B.1
Ejemplo de SFC y requisitos de memoria



Procesadores PLC-5 clásicos

pareja de paso/transición
8 palabras

divergencia simultánea
 $n = 2$
 $n + 1 = 3$ palabras

bifurcación de selección
 $n = 3$
 $5n + 5 = 20$

3 parejas de paso/ transición
 $3 \times 8 = 24$ palabras

convergencia simultánea
 $n = 2$
 $n^2 + 6n + 3 = 19$ palabras

paso/transición
8 palabras

82 palabras (subtotal)
+ 2 palabras (inicio y fin de programa)

84 palabras en total para SFC

Procesadores PLC-5 con características mejoradas

una acción/paso
 $a=1$
 $16 + 6a=22$ palabras

divergencia simultánea
 $n = 2$
 $3n + 1 = 7$ palabras

bifurcación de selección
 $n = 3 \quad a = 1$
 $11 + 6a + 7n = 38$ palabras

3 parejas de paso/ transición $a = 1$
 $3 (16 + 6a) = 66$ palabras

convergencia simultánea
 $n = 2 \quad a = 2$
 $5 + 11n + 6a = 39$ palabras

una acción/paso $a = 1$
 $16 + 6a = 22$ palabras

194 palabras (subtotal)
+ 18 palabras (inicio y fin de programa)
(8 acciones * 6 palabras – supone
1 acción única por paso)

260 palabras en total para SFC

Restricciones dinámicas – Procesadores PLC-5 clásicos solamente

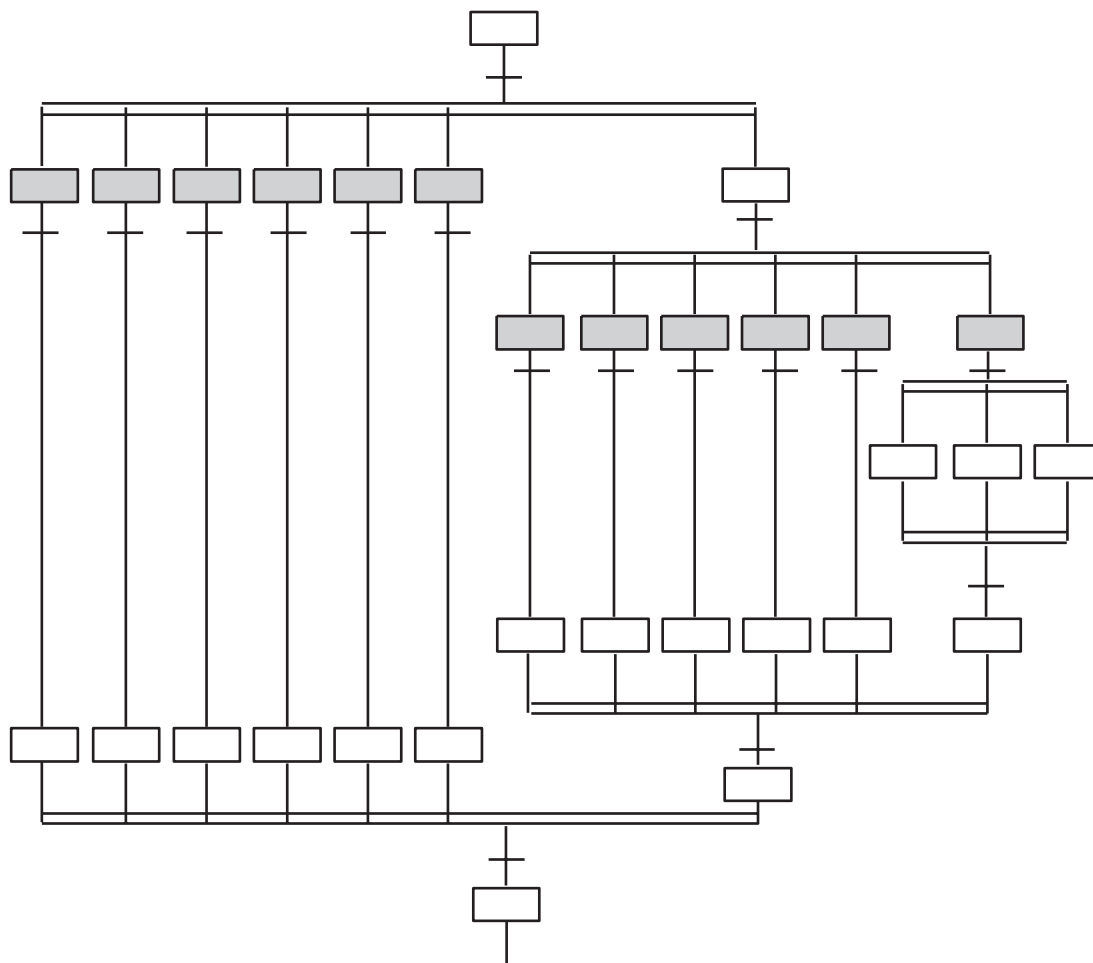
Si usa un procesador PLC-5 clásico y el SFC tiene más de 12 rutas de acceso en paralelo, es necesario determinar el número de rutas de acceso en paralelo que pueden estar activos a la vez. El límite dinámico es 24 rutas de acceso en paralelo simultáneas para un procesador PLC-5 clásico.

Cuando una transición se hace verdadera, los pasos anteriormente activos (que ahora esperan el post-escán) y los pasos nuevamente activos se encuentran momentáneamente en la cola de ejecución. Puede tener hasta 23 pasos activos en paralelo siempre que se asegure de que sólo una transición se haga verdadera a la vez.

Determine el número de pasos activos contando los pasos en cada lado de las transiciones que controlan la zona más amplia del SFC. Por ejemplo, 12 transiciones que son verdaderas a la vez significan por lo menos 24 pasos activos simultáneos. Si alguna divergencia nueva simultánea sigue una de estas transiciones, se excede el máximo de 24 rutas de acceso activas.

Si el diagrama de función en la Figura B.2 se encuentra en el punto donde los 12 pasos sombreados están activos y todas las transiciones que siguen a dichos pasos se hacen verdaderas simultáneamente, el sistema intenta tener 26 pasos activos (12 para el post-escán y 14 para el primer escán) y el procesador entrará en fallo.

Figura B.2
El límite dinámico de pasos activos se puede exceder
(Procesadores PLC-5 clásicos)



Secuencias de escán

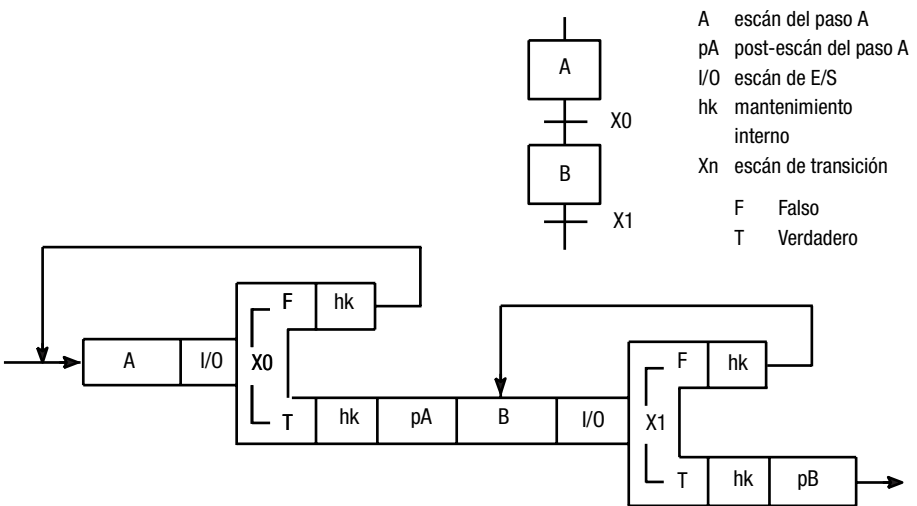
El procesador escanea el SFC desde la parter superior hasta la parte inferior y de la izquierda a la derecha. Cuando el escán encuentra pasos activos en paralelo, el procesador ejecuta la lógica de escalera primero en el paso del extremo izquierdo y mueve la lógica de escalera en el próximo paso en paralelo hasta que se ejecute todos los pasos activos. El procesador reconoce los pasos en paralelo según su posición con respecto a su divergencia común y no necesariamente según su posición en la pantalla.

Escán de pasos y transiciones

Por lo general, el procesador escanea un paso activo, luego escanea las E/S y continúa este ciclo hasta que la lógica de transición es verdadera. El escán del paso incluye la evaluación de todos los calificadores de acción del paso y el escán de todas las acciones apropiadas. Cuando la transición es verdadera, el procesador escanea el paso actual una vez más (post-escán). Durante el post-escán, el procesador fuerza que todos los renglones en el paso sean falsos y restablece la lógica de renglón. El procesador no actualiza las E/S entre un post-escán y el escán del próximo paso activo. La Figura B.3 indica la secuencia de escán para un paso, transición y post-escán. Si usa los procesadores PLC-5 con características mejoradas, puede configurar las operaciones de escán y post-escán. Vea el manual de programación para obtener más información.

Importante: Los subdiagramas activos por un diagrama se escanean antes del mantenimiento interno del sistema..

Figura B.3
Secuencia de escán para un paso, transición y postescán

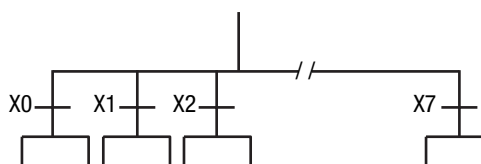


15556

Escán de bifurcaciones seleccionadas

El procesador selecciona una ruta de múltiples rutas en paralelo en una bifurcación seleccionada (Figura B.4). El procesador prueba las transiciones X0 a Xn desde la izquierda hacia la derecha hasta que una de las transiciones se hace verdadera. La ruta con la primera transición verdadera es la ruta activa.

Figura B.4
Bifurcación seleccionada – Divergencia

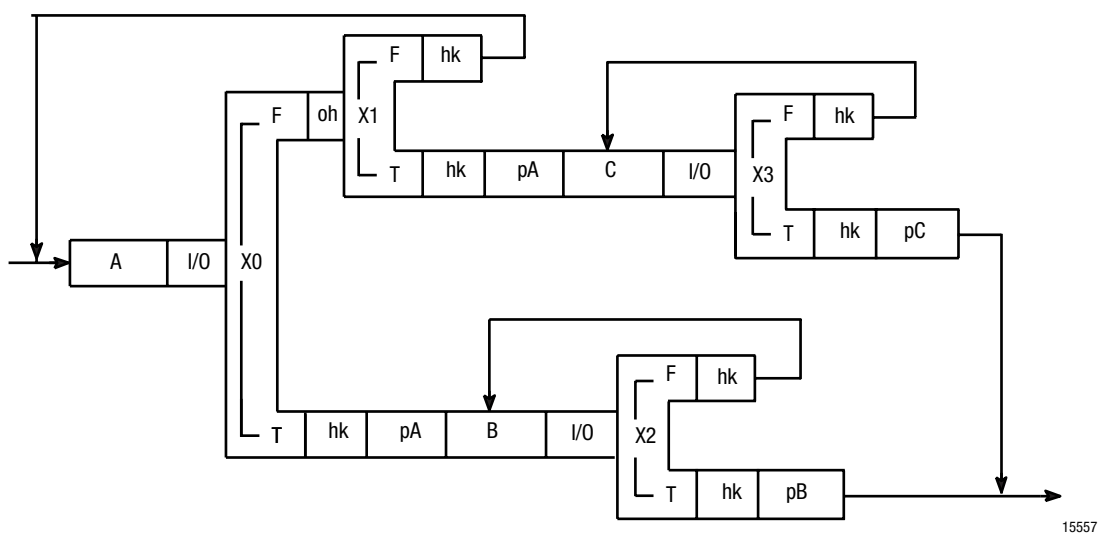
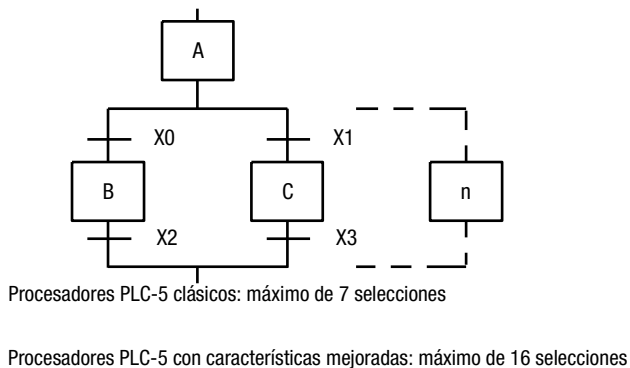


Puesto que hay solamente una ruta activa, la secuencia de escán para la convergencia es idéntica a un paso y transición. La Figura B.5 indica la secuencia de escán para la divergencia y la convergencia de una bifurcación seleccionada.

Figura B.5
Secuencia de escán para una bifurcación seleccionada –
Divergencia y convergencia

A escán del paso A
 pA post-escán del paso A
 I/O escán de E/S
 hk mantenimiento
 interno
 oh tiempo de procesamiento
 interno
 Xn escán de transición

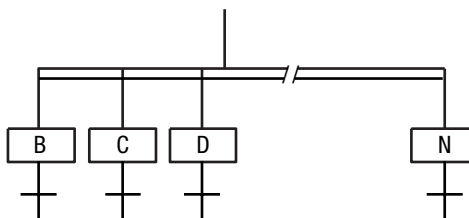
F Falso
 T Verdadero
 n número de transición



Escán de bifurcaciones simultáneas

El procesador escanea todas las rutas en paralelo en una bifurcación simultánea (Figura B.6). Durante el primer escán, el procesador escanea el paso B, luego el paso C, hasta que el procesador escanea todos los pasos en la divergencia.

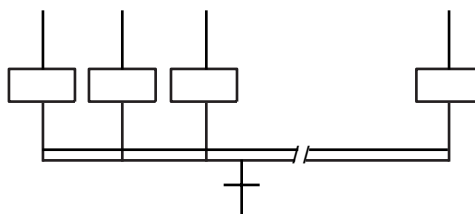
Figura B.6
Bifurcación simultánea – Divergencia



Durante los escaneos siguientes, el procesador escanea desde la izquierda según el orden de paso, E/S y transición para cada ruta.

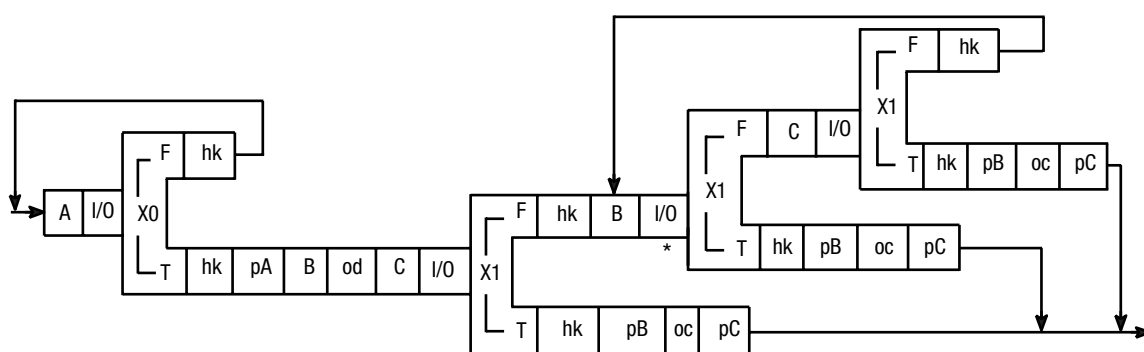
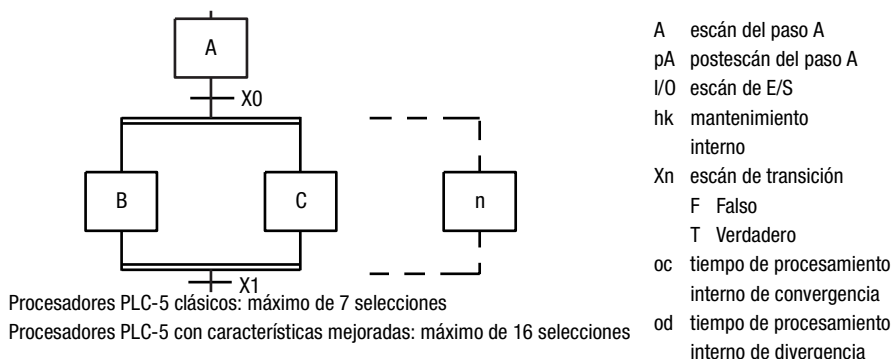
La progresión vertical de paso a paso es independiente de los pasos activos en otras rutas en paralelo (Figura B.7).

Figura B.7
Bifurcación simultánea – Convergencia



La transición común no puede hacerse verdadera hasta que el procesador escanea todos los pasos en la bifurcación por lo menos una vez. Una vez que la transición se hace verdadera, el procesador no escanea las demás rutas en la bifurcación. El procesador post-escanea cada paso en la bifurcación. La Figura B.8 indica la secuencia de escán para la divergencia y convergencia de una bifurcación seleccionada.

Figura B.8
Secuencia de escán para una bifurcación simultánea – Divergencia y convergencia



* En un procesador PLC-5 con características mejoradas, estos estados no ocurren si la configuración de escán se establece en el modo ADVANCED.

** Los subdiagramas conectados a este MCP se ejecutan ahora seguidos por la ejecución de MCP subsiguientes. Si este diagrama es MCP B y tiene acciones activas de subdiagrama pero los MCP A y C tienen programas de lógica de escalera, la secuencia es:

MCP A, diagrama en MCP B, subdiagramas de MCP B, MCP C

Ejemplo de SFC y secuencia de escán

La Figura B.9 muestra un ejemplo de SFC. La Figura B.10 muestra la secuencia de escán para el ejemplo de SFC. Use este ejemplo de SFC y la secuencia de escán como guía. Estas figuras pueden no aplicarse a su sistema.

Figura B.9
Ejemplo de SFC para ejemplo de secuencia de escán

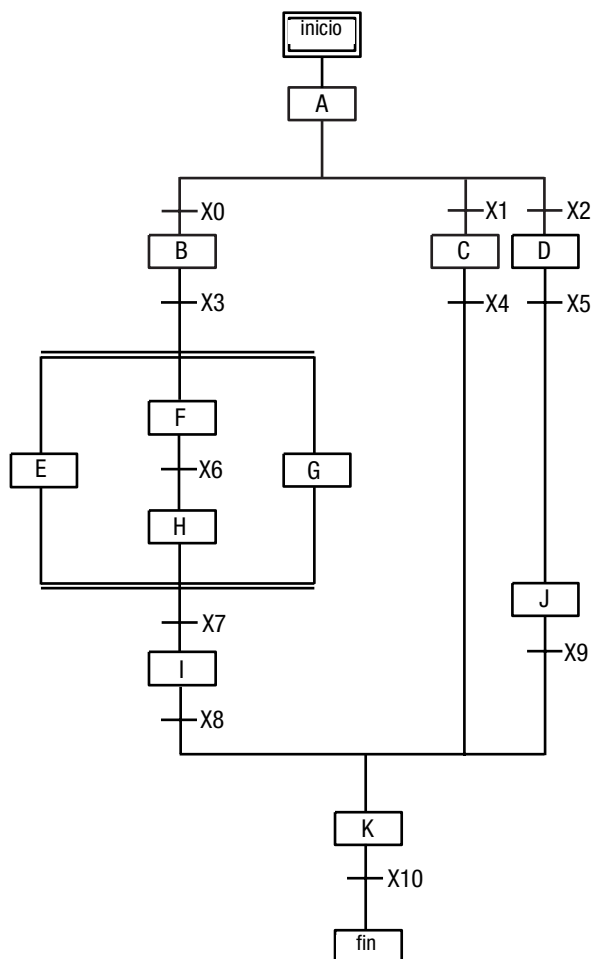
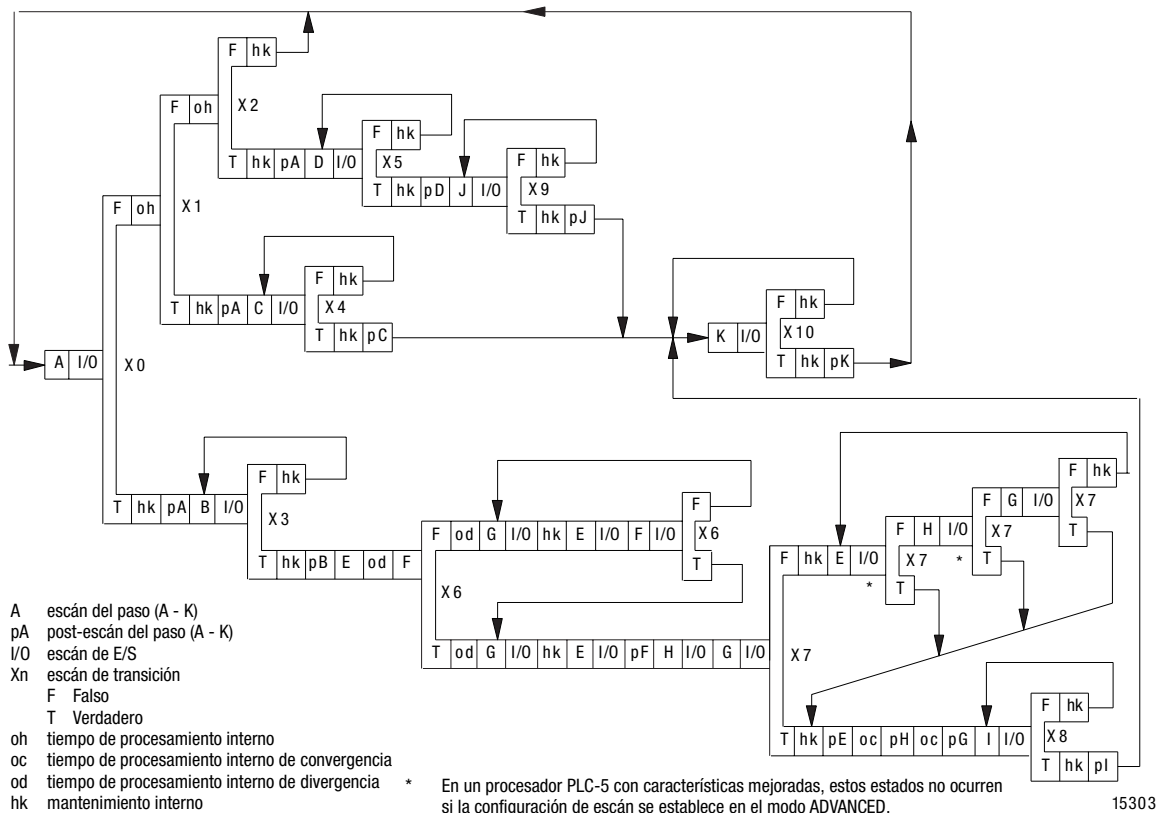


Figura B.10
Ejemplo de secuencia para el ejemplo de SFC



Tiempo de ejecución – Procesadores PLC-5 clásicos

Para determinar el tiempo de ejecución del archivo de memoria del procesador en un procesador PLC-5 clásico, sume el tiempo de ejecución para la lógica de escalera y el tiempo de ejecución para el SFC. Vea el Apéndice A para obtener información acerca de los tiempos de ejecución para la lógica de escalera. Use los diagramas de secuencia o ecuaciones para determinar el tiempo de ejecución de un SFC.

Cómo usar los diagramas de secuencia para determinar el tiempo de ejecución

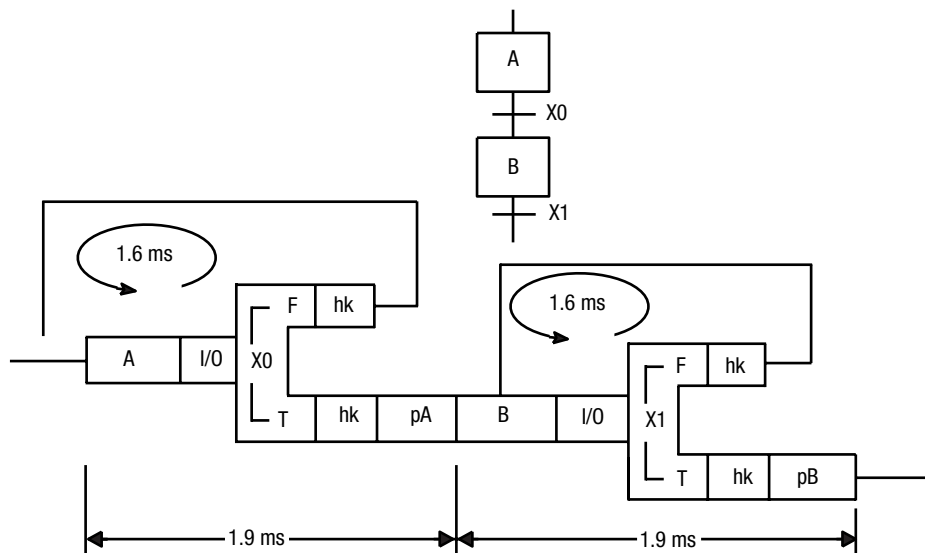
La Tabla B.C indica los tiempos de ejecución que se deben añadir según el diagrama de secuencia para el SFC.

Tabla B.C
Tiempos de ejecución para secciones del diagrama de secuencia –
Procesadores PLC-5 clásicos

Este evento:	Necesita esta cantidad de tiempo (en milisegundos):
A	tiempo para ejecutar la lógica de paso A + 0.1 ms
pA	tiempo para escanear la lógica de paso A con renglones falsos + 0.1 ms
XN	transición N falsa (F): tiempo para escanear la lógica + 0.1 ms transición N verdadera (T): tiempo para escanear la lógica + 0.25 ms
I/O (escán de E/S)	0.6 ms
hk (mantenimiento interno)	0.7 ms (se aumenta según el tráfico DH+ aumentado)
oh (tiempo de procesamiento interno)	0.02 ms
od (tiempo de procesamiento interno de divergencia)	0.3 ms
oc (tiempo de procesamiento interno de convergencia)	0.2 ms

Para determinar el tiempo de ejecución en el peor de los casos, suponga que una transición se hace verdadera después de un escán de E/S o después del escán de una transición. Esta suposición requiere una secuencia de escán adicional antes de que la transición se haga verdadera.

El tiempo de escán de un paso y transición es proporcional al número de renglones para el paso y transición. La Figura B.11 muestra el tiempo de escán mínimo que contiene una sola instrucción OTE y END y una transición que contiene una sola instrucción XIC y EOT.

Figura B.11**Tiempo de escán mínimo para una pareja de paso y transición**

14271

Cómo usar las ecuaciones para determinar el tiempo de ejecución

Las ecuaciones que usa dependen de si el escán es de estado constante (paso y transición sencillos) o divergente y convergente.

El tiempo de escán de estado estable significa que todas las transiciones que siguen a los pasos activos son falsas. Use esta ecuación (Tabla B.D):

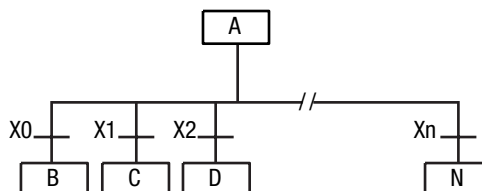
$$T_{\text{milisegundos}} = 0.8a + 0.7 + T_{\text{escán}}$$

Tabla B.D**Variables para el tiempo de escán de estado constante**

Donde:	Significa:
$T_{\text{milisegundos}}$	el tiempo de escán de estado constante en milisegundos
a	el número de pasos activos
$T_{\text{escán}}$	el tiempo total para escanear la lógica en todos los pasos activos y transiciones falsas asociadas

El tiempo de escán divergente se inicia cuando el procesador prueba una transición y se termina cuando el procesador escanea las E/S del próximo paso. El tiempo de escán divergente incluye el tiempo de escán de transición, tiempo de post-escán del paso anterior, tiempo de escán del paso nuevo, tiempo de procesamiento interno y tiempo de escán de cada paso activo en paralelo fuera de la divergencia.

Para una divergencia de ruta seleccionada, la mejor condición es cuando la transición se hace verdadera antes del escán de E/S. Use esta ecuación (Tabla B.E):

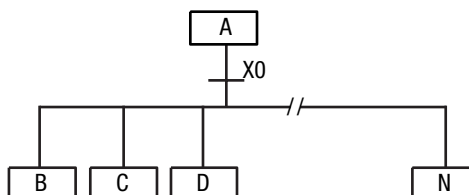


$$T_{\text{milisegundos}} = T_X + pA + T_S + 0.02(n-1) + 1.55 + 0.8a + T_0$$

Tabla B.E
Variables para el tiempo de escán divergente de ruta seleccionada

Donde:	Significa:
$T_{\text{milisegundos}}$	el tiempo de escán de transición en milisegundos del paso A al primer paso en la ruta seleccionada N
T_X	la suma de tiempos de escán de lógica de las transiciones X0, X1, ..., Xn en la divergencia hasta la transición seleccionada inclusive
pA	el tiempo de post-escán para el paso (paso A) que precede la divergencia
T_S	el tiempo de escán para la lógica en el paso nuevo (paso N)
n	el número de la ruta seleccionada (1-7 de izquierda a derecha)
a	el número de pasos activos fuera de la divergencia
T_0	la suma de los tiempos de escán de todos los otros pasos activos y transiciones en paralelo con la divergencia, pero fuera de la divergencia

Para una divergencia simultánea, la mejor condición es cuando la transición se hace verdadera antes del escán de E/S. Use esta ecuación (Tabla B.F):



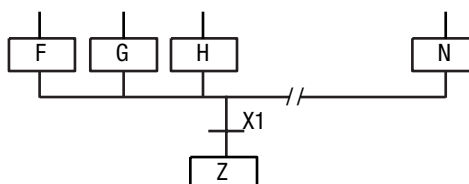
$$T_{\text{milisegundos}} = T_{X0} + pA + T_S + 0.3(n-1) + 1.97 + 0.8a + T_0$$

Tabla B.F
Variables para el tiempo de escán divergente de la ruta simultánea

Donde:	Significa:
$T_{\text{milisegundos}}$	el tiempo de transición en milisegundos a partir del momento en que la transición X0 se hace verdadera hasta que el procesador acaba por escanear el último paso simultáneo (paso N) en la divergencia
T_{X0}	el tiempo de escán de lógica en la transición X0
pA	el tiempo necesario para realizar un post-escán del paso A
T_S	la suma de los tiempos de escán de lógica en los pasos nuevos (paso B, paso C, . . . , paso N)
n	el número de pasos activos simultáneos en la divergencia
a	el número de pasos activos en paralelo fuera de la divergencia
T_0	la suma de los tiempos de escán de lógica en todos los demás pasos activos y transiciones en paralelo con la divergencia, pero fuera de la divergencia

En el pero de los casos, suponga que una transición se hace verdadera después de un escán de E/S o después del escán de una transición. Esta suposición requiere una secuencia de escán adicional antes de que la transición se haga verdadera.

El tiempo de escán convergente es cuando se termina una bifurcación simultánea. El mejor de los casos es cuando la transición se hace verdadera antes del escán de E/S. Use esta ecuación (Tabla B.G):



$$T_{\text{milisegundos}} = T_{X1} + T_p + T_Z + 0.2(n-1) + 1.5 + 0.8a + T_0$$

Tabla B.G
Variables para el tiempo de escán convergente de ruta simultánea

Donde:	Significa:
$T_{\text{milisegundos}}$	el tiempo de transición en milisegundos a partir del momento en que la transición X1 se hace verdadera hasta que el procesador acaba de escanear el paso Z
T_{X1}	el tiempo de escán de lógica en la transición X1
T_p	la suma de los tiempos de post-escán de los pasos F, G, . . . , N
T_Z	el tiempo de escán de lógica en el paso Z
n	el número de pasos activos simultáneos en la convergencia
a	el número de pasos activos en paralelo fuera de la convergencia
T_0	la suma de los tiempos de escán de lógica de todos los demás pasos y transiciones en paralelo con la convergencia, pero fuera de la convergencia

En el pero de los casos, suponga que una transición se hace verdadera después de un escán de E/S o después del escán de una transición. Esta suposición requiere una secuencia de escán adicional antes de que la transición se haga verdadera.

Notas:

Tipos de datos válidos para los operandos de instrucciones

Objetivos del apéndice

Este apéndice indica todas las instrucciones disponibles y los operandos de las mismas además de los valores/tipos de datos que son válidos para cada operando.

La tabla siguiente explica cada tipo/valor de datos válido:

Este tipo/valor de datos:	Acepta:
inmediato (constante de programa)	cualquier valor entre -32,768 y 32,767
número entero	cualquier tipo de datos de número entero: número entero, temporizador, contador, estado, bit, entrada, salida, ASCII, BCD, control (por ejemplo, N7:0, C4:0, etc.)
punto flotante (coma flotante)	cualquier tipo de datos de punto flotante (coma flotante) con precisión de 7 dígitos (el rango válido es $\pm 1.1754944e^{-38}$ a $\pm 3.4028237e^{+38}$).
transferencia en bloques	cualquier tipo de transferencia en bloques (por ejemplo, BT14:0)
transferencia ControlNet	cualquier tipo de datos CT (por ejemplo, CT14:0)
mensaje	cualquier tipo de datos de mensaje (por ejemplo, MG15:0)
PID	cualquier tipo de datos PID (por ejemplo, PD16:0) o tipo de datos de número entero (por ejemplo, N7:0)
cadena	cualquier tipo de datos de cadena (por ejemplo, ST12:0)
estado SFC	cualquier tipo de datos de estado SFC (por ejemplo, SC17:0)

Operandos de instrucciones y tipos de datos válidos

La Tabla C.A indica las instrucciones de programación que puede usar y los operandos para dichas instrucciones. También puede usar esta tabla para formatear las instrucciones en ASCII para la importación. Vea el manual de programación para obtener más información acerca de la importación.

Las instrucciones marcadas con un asterisco (*) son compatibles solamente con los procesadores PLC-5 con características mejoradas.

Para introducir la sintaxis de importación para cualquiera de las instrucciones indicadas en la Tabla C.A:

- ponga todos los operandos entre paréntesis
- separe cada operando con comas

Por ejemplo, la siguiente es la sintaxis de importación para la instrucción FAL:

FAL (R6:0, 10, 0, ALL, #N7:0, #N7:1+N7:2);

Tabla C.A
Instrucciones de programación y operandos

Instrucción	Descripción	Operando	Valor válido	Requisito de transición de falso a verdadero
ABL *	Prueba de línea búfer de ASCII	canal	inmediato, número entero 0-4	sí
		control	control	
ACB *	Número de caracteres ASCII en el búfer	canal	inmediato, número entero	sí
		control	control	
ACI *	Cadena ASCII a número entero	origen	cadena	no
		destino	número entero	
ACN *	Concatenado de cadenas ASCII	origen A	cadena	no
		origen B	cadena	
		destino	cadena	
ACS *	Arco coseno	origen	inmediato, punto flotante (coma flotante) (en radianes), número entero	no
		destino	punto flotante (coma flotante) (en radianes), número entero	
ACT *	Acción SFC (solamente para la importación/exportación ASCII)	número de acción	inmediato	N/A
		número de archivo	0 - 999	
		destino	cadena	
ADD	ADD	origen A	inmediato, número entero, punto flotante (coma flotante)	no
		origen B	inmediato, número entero, punto flotante (coma flotante)	
		destino	número entero, punto flotante (coma flotante)	
AEX *	Extracción de cadena	origen	cadena	no
		índice	inmediato, número entero 0-82	
		número	inmediato, número entero 0-82	
		destino	cadena	
AFI	Siempre falso	ninguno		no

Instrucción	Descripción	Operando	Valor válido	Requisito de transición de falso a verdadero
AHL *	Establecimiento/ restablecimiento de las líneas de handshaking ASCII	canal	inmediato, número entero 0-4	sí
		máscara AND de handshaking	inmediato, número entero hexadecimal	sí
		máscara OR de handshaking	inmediato, número entero hexadecimal	
		control	control	
AIC *	Número entero ASCII a cadena	origen	inmediato, número entero	no
		destino	cadena	
AND	Y lógico	origen A	número entero	no
		origen B	número entero	
		destino	número entero	
ARD *	Caracteres de lectura ASCII	canal	inmediato, número entero 0-4	sí
		destino	cadena	
		control	control	
		longitud de cadena	0 - 82	
ARL *	Línea de lectura ASCII	canal	inmediato, número entero 0-4	sí
		destino	cadena	
		control	control	
		longitud de cadena	0 - 82	
ASC *	Búsqueda de cadena ASCII	origen	cadena	no
		índice	inmediato, número entero 0-4	
		búsqueda	cadena	
		resultado	número entero	
ASN *	Arco seno	origen	inmediato, punto flotante (coma flotante) (en radianes)	no
		destino	punto flotante (coma flotante) (en radianes)	
ASR *	Comparación de cadenas ASCII	origen A	cadena	no
		origen B	cadena	
ATN *	Arco tangente	origen	punto flotante (coma flotante) (en radianes)	no
		destino	punto flotante (coma flotante) (en radianes)	

Instrucción	Descripción	Operando	Valor válido	Requisito de transición de falso a verdadero
AVE *	Archivo de promedio	archivo	número entero, punto flotante (coma flotante)	sí
		destino	número entero, punto flotante (coma flotante)	
		control	control	
		longitud	1 - 1000	
		posición	0 - 999	
AWA *	Escritura ASCII con anexo	canal	inmediato, número entero 0-4	sí
		origen	cadena	
		control	control	
		longitud de cadena	0 - 82	
AWT *	Escritura ASCII	canal	inmediato, número entero	sí
		origen	cadena	
		control	control	
		longitud	0 - 82	
BRK	Interrupción	ninguno		no
BSL	Desplazamiento de bit a la izquierda	archivo	binario	sí
		control	control	
		dirección de bit	bit	
		longitud	1 - 16000 (longitud en bits)	
BSR	Desplazamiento de bit a la derecha	archivo	binario	sí
		control	control	
		dirección de bit	bit	
		longitud	1 - 16000 (longitud en bits)	
BTD	Distribuidor de bits	origen	inmediato, número entero	no
		bit de origen	inmediato, (0 - 15) número entero	
		destino	número entero	
		bit de destino	inmediato (0 - 15)	
		longitud	inmediato (1 - 16)	

Instrucción	Descripción	Operando	Valor válido	Requisito de transición de falso a verdadero
BTR ¹	Transferencia en bloques de lectura	rack	00-277 octal	sí
		grupo	0-7	
		módulo	0-1	
		bloque de control	bloque, número entero	
		archivo de datos	número entero	
		longitud	0, 1-64	
		continuo	SI, NO	
BTW ¹	Transferencia en bloques de escritura	rack	00-277 octal	sí
		grupo	0-7	
		módulo	0-1	
		bloque de control	bloque, número entero	
		archivo de datos	número entero	
		longitud	0, 1-64	
		continuo	SI, NO	
CIO	Transferencia de E/S ControlNet	bloque de control	transferencia ControlNet (1 - 64)	sí
CIR	Rutina de entrada personalizada	número de archivo de programa	inmediato (2-999) para todos los procesadores	N/A
	(para uso con las aplicaciones CAR solamente)	lista de parámetros de entrada	inmediato, número entero, punto flotante (coma flotante)	
		lista de parámetros de retorno	número entero, punto flotante (coma flotante)	
CLR	Borrar	destino	número entero, punto flotante (coma flotante)	no
CMP	Comparar	expresión, expresión relativa, expresión	expresión que usa valores o direcciones con evaluadores (para obtener una lista, vea el capítulo 3 de este manual)	no
		mnemónico EXE (fin de expresión)	EXE	
		para la importación ASCII solamente		

¹ En el modo no continuo, las funciones de lógica de escalera BTR y BTW requieren que se ejecute una transición de falso a verdadero. En el modo continuo, una vez que el renglón se hace verdadero, las funciones BTR y BTW continúan ejecutándose independientemente de la condición del renglón. Vea la página 15-8 para obtener más información.

Instrucción	Descripción	Operando	Valor válido	Requisito de transición de falso a verdadero
COP	Copiar archivos	origen	arreglo	no
		destino	arreglo	
		longitud	inmediato (1 - 1000)	
COR	Rutina de salida personalizada	número de archivo de programa	inmediato (2-999) para todos los procesadores	no
	(para uso con las aplicaciones CAR solamente)	lista de parámetros de entrada	inmediato, número entero, punto flotante (coma flotante)	
		lista de parámetros de retorno	número entero, punto flotante (coma flotante)	
COS *	Coseno	origen	inmediato, punto flotante (coma flotante) (en radianes)	no
CPT	Calcular	expresión matemática	expresión que usa valores o direcciones inmediatas de punto flotante (coma flotante) de número entero (vea el capítulo 4 en este manual para obtener una lista)	no
		mnemónico EXE – para la importación ASCII solamente	EXE	
		expresión relativa	direcciones con evaluadores (vea el capítulo 4 en este manual para obtener una lista)	
		destino	número entero, punto flotante (coma flotante)	
CTD	Conteo regresivo	contador	contador	sí
		PRE	-32,768 - +32,767	
		ACC	-32,768 - +32,767	
CTU	Conteo progresivo	contador	contador	sí
		PRE	-32,768 - +32,767	
		ACC	-32,768 - +32,767	

Instrucción	Descripción	Operando	Valor válido	Requisito de transición de falso a verdadero
DDT	Detección diagnóstica	arreglo de origen	binario	sí
		arreglo de referencia	binario	
		arreglo de resultado	número entero	
		control de comparación	control	
		longitud	1 - 16000 (longitud en bits)	
		posición	0 - 15999	
		control de resultado	control	
		longitud	1 - 1000	
		posición	0 - 999	
DEG *	Grado (conversión de radianes en grados)	origen	inmediato, punto flotante (coma flotante) (en radianes)	no
		destino	inmediato, punto flotante (coma flotante) (en grados)	
DFA	Anunciador de fallo de diagnóstico	archivo de control	número entero	
		número de E/S	inmediato (8, 16, 32)	
		número de archivo de programa	inmediato (3-999)	
DIV	Dividir	origen A	inmediato, número entero, punto flotante (coma flotante)	no
		origen B	inmediato, número entero, punto flotante (coma flotante)	
		destino	número entero, punto flotante (coma flotante)	
DTR	Datos transicionales	origen	inmediato, número entero	no
		máscara	inmediato, número entero	
		referencia	número entero	
EOC	fin de compresión SFC (vea (SOC))	solamente para la importación/ exportación ASCII		N/A
EOR	fin de renglón	solamente para la importación/ exportación ASCII		N/A
EOT	fin de transición	ninguno		no
ESE	fin de la bifurcación simultánea SFC (vea SEL)	solamente para la importación/ exportación ASCII		N/A

Instrucción	Descripción	Operando	Valor válido	Requisito de transición de falso a verdadero
EQU	Igual	origen A	número entero, punto flotante (coma flotante)	no
		origen B	número entero, punto flotante (coma flotante)	
EOP	fin de programa SFC	solamente para la importación/exportación ASCII		N/A
ERI	error en una instrucción de entrada	solamente en los archivos de exportación ASCII		N/A
ERO	error en una instrucción de salida	solamente en los archivos de exportación ASCII		N/A
ESI	fin de la bifurcación simultánea SFC (vea SIM)	solamente para la importación/exportación ASCII		N/A
FAL	Aritmética y lógica de archivos	control	control	sí
		longitud	1 - 1000	
		posición	0 - 999	
		modo	(INC, 1-1000, ALL)	
		destino	número entero, punto flotante (coma flotante)	
		expresión matemática	instrucción matemática indexada	
FBC	Comparación de bits de archivo	arreglo de origen	binario	sí
		arreglo de referencia	binario	
		arreglo de resultado	número entero	
		control de comparación	control	
		longitud	1 - 16000 (longitud en bits)	
		posición	0 - 15999	
		control de resultado	control	
		longitud	1 - 1000	
		posición	0 - 999	

Instrucción	Descripción	Operando	Valor válido	Requisito de transición de falso a verdadero
FFL	Carga FIFO	operando de origen	inmediato, indexado, número entero	sí
		arreglo FIFO	indexado, número entero	
		control FIFO	control	
		longitud	1 - 1000	
		posición	0 - 999	
FFU	Descarga FIFO	arreglo FIFO	indexado, número entero	sí
		destino	indexado, número entero	
		control FIFO	control	
		longitud	1 - 1000	
		posición	0 - 999	
FLL	Llenar archivo	operando de origen	indexado, número entero, punto flotante (coma flotante)	no
		arreglo de destino	arreglo	no
		longitud	inmediato (1 - 1000)	
FOR	Lazo PARA	número LBL	número entero	no
		índice	número entero	
		valor inicial	inmediato, número entero	
		valor de terminal	inmediato, número entero	
		tamaño de paso	inmediato, número entero	
FRD	De BCD	origen	inmediato, número entero	no
		destino	número entero	
FSC	Búsqueda y comparación de archivos	control	control	sí
		longitud	1 - 1000	
		posición	0 - 999	
		modo	inmediato, número entero (0, INC, 1-1000, ALL)	
		expresión matemática	instrucción matemática indexada	
GEQ	Mayor o igual a	origen A	inmediato, número entero, punto flotante (coma flotante)	no
		origen B	inmediato, número entero, punto flotante (coma flotante)	

Instrucción	Descripción	Operando	Valor válido	Requisito de transición de falso a verdadero
GRT	Mayor que	origen A	inmediato, número entero, punto flotante (coma flotante)	no
		origen B	inmediato, número entero, punto flotante (coma flotante)	
IDI	Entrada inmediata de datos	offset de archivo de datos	inmediato (0-999), número entero	sí
		longitud	inmediato (1-64), número entero	
		destino	número entero	
IDO	Salida inmediata de datos	offset de archivo de datos	inmediato (0-999), número entero	sí
		longitud	inmediato (1-64), número entero	
		origen	número entero	
IIN	Entrada inmediata	1 palabra (entrada)	inmediato, número entero PLC-5/10, 11, 12, 15, 20, 25, 30: 000-077 PLC-5/40, 40L: 000-157 PLC-5/60, 60L, 80, :000-237	no
IOT	Salida inmediata	0 palabra (salida)	inmediato, número entero PLC-5/10, 11, 12, 15, 20, 25, 30: 000-077 PLC-5/40, 40L: 000-157 PLC-5/60, 60L, 80: 000-237	no
JMP	Salto	número de etiqueta	inmediato procesadores PLC-5 clásicos: 0-31 procesadores PLC-5 con características mejoradas: 0-255	no
JSR	Salto a subrutina	número de programa de lógica de escalera	inmediato (2 - 999)	no
		lista de parámetros de entrada	inmediato, número entero, punto flotante (coma flotante)	no
		lista de parámetros de retorno	número entero, punto flotante (coma flotante)	
LAB	Etiqueta SFC (importación/exportación solamente)	número de archivo	inmediato procesadores PLC-5 clásicos: 0-31 procesadores PLC-5 con características mejoradas: 0-255	N/A

Instrucción	Descripción	Operando	Valor válido	Requisito de transición de falso a verdadero
LBL	LBL (etiqueta de programa de lógica de escalera)	número de archivo	inmediato procesadores PLC-5 clásicos: 0-31 procesadores PLC-5 con características mejoradas: 0-255	no
LEQ	Menor o igual a	origen A	inmediato, número entero, punto flotante (coma flotante)	no
		origen B	inmediato, número entero, punto flotante (coma flotante)	
LES	Menor que	origen A	inmediato, número entero, punto flotante (coma flotante)	no
		origen B	inmediato, número entero, punto flotante (coma flotante)	
LFL *	Carga LIFO	operando de origen	inmediato, indexado, número entero	sí
		arreglo LIFO	indexado, número entero	
		control LIFO	control	
		longitud	1 - 1000	
		posición	0 - 999	
LFU *	Descarga LIFO	arreglo LIFO	indexado, número entero	sí
		destino	indexado, número entero	
		control LIFO	control	
		longitud	1 - 1000	
		posición	0 - 999	
LIM	Límite	límite bajo	inmediato, número entero, punto flotante (coma flotante)	no
		prueba	inmediato, número entero, punto flotante (coma flotante)	
		límite alto	inmediato, número entero, punto flotante (coma flotante)	
LN *	Algoritmo natural	origen	inmediato, número entero, punto flotante (coma flotante)	no
		destino	punto flotante (coma flotante)	
LOG *	Algoritmo a la base 10	origen	inmediato, número entero, punto flotante (coma flotante)	no
		destino	punto flotante (coma flotante)	no
MCR	Relé de control maestro			no

Instrucción	Descripción	Operando	Valor válido	Requisito de transición de falso a verdadero
MEQ	Comparación con máscara para igual	operando de origen	inmediato, número entero	no
		máscara de origen	inmediato, número entero	
		operando de comparación	inmediato, número entero	
MOV	Mover	origen	inmediato, número entero, punto flotante (coma flotante)	no
		destino	número entero, punto flotante (coma flotante)	
MSG	Mensaje	bloque de control	mensaje, número entero	sí
MUL	Multiplicar	origen A	inmediato, número entero, punto flotante (coma flotante)	no
		origen B	inmediato, número entero, punto flotante (coma flotante)	
		destino	número entero, punto flotante (coma flotante)	
MVM	Mover con máscara	operando de origen	inmediato, número entero	no
		máscara de origen	inmediato, número entero hexadecimal	
		destino	número entero	
NEG	Cambiar signo	origen	número entero, punto flotante (coma flotante)	no
		destino	número entero, punto flotante (coma flotante)	
NEQ	Diferente	origen A	inmediato, número entero, punto flotante (coma flotante)	no
		origen B	inmediato, número entero, punto flotante (coma flotante)	
NOT	NO lógico	origen	inmediato, número entero	no
		destino	número entero	
NSE	próxima bifurcación de selección SFC	solamente para la importación/exportación ASCII		N/A
NSI	próxima bifurcación simultánea SFC	solamente para la importación/exportación ASCII		N/A

Instrucción	Descripción	Operando	Valor válido	Requisito de transición de falso a verdadero
NXT	Siguiente (lazo FOR)	para los números de etiqueta	inmediato procesadores PLC-5 clásicos: 0-31 procesadores PLC-5 con características mejoradas: 0-255	no
OR	O lógico	origen A	inmediato, número entero de bits	sí
		origen B	inmediato, número entero de bits	
		destino	número entero	
OSF *	Un frente descendente	bit de almacenamiento	bit	sí; requiere la ejecución de una transición de falso a verdadero
		bit de salida	inmediato (0 - 15)	
		palabra de salida	número entero	
ONS	Un impulso	bit de origen	bit	sí
OSR *	Un frente ascendente	bit de almacenamiento	bit	sí
		bit de salida	inmediato (0 - 15)	
		palabra de salida	número entero	
OTE	Activación de salida	bit de destino	bit	no
OTL	Enclavamiento de salida	bit de destino	bit	no
OTU	Desenclavamiento de salida	bit de destino	bit	no
PID	PID	bloque de control	PD	no
		bloque de control	número entero	sí
		valor pv	número entero	
		valor retenido	inmediato, número entero	
		valor cv	número entero	
RAD *	Radián (conversión de grados en radianes)	origen	inmediato, punto flotante (coma flotante (en grados))	no
		destino	punto flotante (coma flotante) (en radianes)	
REF	referencia SFC (vea LAB) (importación/exportación ASCII solamente)	número de etiqueta	inmediato (0 - 255)	N/A
RES	Restablecimiento de temporizador/contador		temporizador, contador, control	no
RET	Retorno	lista de parámetros de retorno	inmediato, número entero, punto flotante (coma flotante)	no

Instrucción	Descripción	Operando	Valor válido	Requisito de transición de falso a verdadero
RTO ²	Temporizador retentivo a la conexión	temporizador	temporizador	sí
		base de tiempo	inmediato (0.01, 1.0)	
		PRE	0 - 32767	
		ACC	0 - 32767	
SBR	Subrutina	lista de parámetros de entrada	número entero, punto flotante (coma flotante)	no
SDS	Secuenciador dirigido inteligente	archivo de control	número entero	no
		número de E/S	inmediato (8, 16, 32)	
		número de archivo de programa	inmediato (3-999)	
SDZ	inicio de la zona de eliminación, ediciones no ensambladas	solamente en los archivos de exportación ASCII		N/A
SEL	bifurcación de selección SFC	solamente para la importación/exportación ASCII		N/A
SFR*	restablecimiento SFC	número de archivo SFC	inmediato (1 - 999)	no
		reinicio en el paso	inmediato, número entero	
SIM	bifurcación simultánea SFC	solamente para la importación ASCII		N/A
SIN *	Seno	origen	inmediato, punto flotante (coma flotante) (en radianes)	no
		destino	punto flotante (coma flotante) (en radianes)	
SIZ	inicio de la zona de inserción, ediciones no ensambladas	solamente en los archivos de exportación ASCII		N/A
SOC	inicio de compresión	solamente para la exportación/importación ASCII		N/A
SOP	inicio de programa SFC	solamente para la exportación/importación ASCII		N/A
SOR	inicio de renglón	solamente para la exportación/importación ASCII		N/A
² Esta instrucción requiere escaneos periódicos para que se actualice. Vea la página 2-13 en este manual o el manual del usuario de texto estructurado para obtener más información.				

Instrucción	Descripción	Operando	Valor válido	Requisito de transición de falso a verdadero
SQI	Secuenciador de entradas	archivo	número entero, indexado	no
		máscara	inmediato, indexado hexadecimal, número entero	
		origen	inmediato, indexado, número entero	
		control	control	
		longitud	1 - 1000	
		posición	0 - 999	
SQL	Carga secuenciador	archivo	número entero, indexado	sí
		origen	inmediato, indexado, número entero	
		control	control	
		longitud	1 - 1000	
		posición	0 - 999	
SQO	Secuenciador de salidas	archivo	número entero, indexado	sí
		máscara de destino	inmediato, indexado, número entero	
		destino	indexado, número entero	
		control	control	
		longitud	1 - 1000	
		posición	0 - 999	
SQR	Raíz cuadrada	origen	inmediato, número entero, punto flotante (coma flotante)	no
		destino	número entero, punto flotante (coma flotante)	
SRT *	Clasificar	archivo de clasificación	número entero, punto flotante (coma flotante)	sí
		control de archivo	control	
		longitud	1 - 1000	
		posición	0 - 999	
SRZ	inicio de la zona de reemplazo, ediciones no ensambladas	solamente en los archivos de exportación/ importación ASCII		N/A
STP	Paso SFC (procesadores PLC-5 clásicos) (importación/ exportación ASCII solamente)	número de archivo	2 - 999	N/A

Instrucción	Descripción	Operando	Valor válido	Requisito de transición de falso a verdadero
STP *	Paso SFC (procesadores PLC-5 con características mejoradas) (importación/exportación solamente)	número de archivo del temporizador de paso	2 - 9999	N/A
		base de tiempo	inmediato (0.01, 1.0)	
		cualificador	N, S, R, L, D, P1, P0, SL, SD, DS	
		número de acción (de ACT)	inmediato	
		número de archivo de temporizador	temporizador	
		base de tiempo	inmediato (0.01, 1.0)	
STD *	Desviación estándar	archivo de desviación estándar	número entero, punto flotante (coma flotante)	sí
		destino	número entero, punto flotante (coma flotante)	
		control de archivo	control	
		longitud	1 - 1000	
		posición	0 - 999	
SUB	Restar	origen A	inmediato, número entero, punto flotante (coma flotante)	no
		origen B	inmediato, número entero, punto flotante (coma flotante)	
		destino	número entero, punto flotante (coma flotante)	
TAN *	Tangente	origen	inmediato, punto flotante (coma flotante) (en radianes)	no
		destino	punto flotante (coma flotante) (en radianes)	
TID *	Identificación de testigo (importación/exportación ASCII solamente)	número de identificación de testigo (debe ser único por archivo SFC)	inmediato	N/A
TND	Fin temporal			no
TOD	A BCD	origen	inmediato, número entero	no
		destino	número entero	
TOF ²	Temporizador a la desconexión	temporizador	temporizador	sí: requiere la transición de verdadero a falso para ejecutarse

Instrucción	Descripción	Operando	Valor válido	Requisito de transición de falso a verdadero
TOF ²	Temporizador a la desconexión	base de tiempo	inmediato (0.01, 1.0)	sí: requiere la transición de verdadero a falso para ejecutarse
		PRE	0 - 32767	
		ACC	0 - 32767	
TON ³	Temporizador a la conexión	temporizador	temporizador	sí
		base de tiempo	inmediato (0.01, 1.0)	
		PRE	0 - 32767	
		ACC	0 - 32767	
TRC	Transición SFC (importación/exportación ASCII solamente)	número de archivo	2 - 999 para todos los procesadores	N/A
UID *	Inhabilitación de interrupción del usuario			no
UIE *	Habilitación de interrupción del usuario			no
XIC	Examina si cerrado	bit de origen	bit	no
XIO	Examina si abierto	bit de origen	bit	no
XOR	O exclusivo	origen A	inmediato, número entero de bits	no
		origen B	inmediato, número entero de bits	
		destino	número entero	
XPY *	X a la potencia de Y	origen A	inmediato, número entero	no
		origen B	inmediato, número entero	
		destino	número entero	

²Esta instrucción requiere escanes periódicos para que se actualice. Vea la página 2-13 en este manual o el manual del usuario de texto estructurado para obtener más información.

³Esta instrucción requiere escanes periódicos para que se actualice. Vea la página 1-14 en este manual o el manual del usuario de texto estructurado para obtener más información.

Notas:

Símbolos

&B 3-2, 4-5

&H 3-2, 4-5

&O 3-2, 4-5

A

almacenamiento de datos

archivos de imagen de E/S

1-2

archivo

operaciones de búsqueda y

comparación 9-17

Archivo de compatibilidad PLC2

16-15

archivo de control

ejemplo 8-2

archivo de datos

manipulación 8-3

archivos

funciones 9-14

instrucción COP 9-19

instrucciones

FLL 9-20

modos de operación 8-5

operaciones aritméticas 9-7

operaciones de copiar 9-5

operaciones lógicas 9-12

archivos de datos

rango de valores C-1

archivos de imagen de E/S 1-2

ASCII

ABL 17-4

ACB 17-5

ACI 17-6

ACN 17-7

AEX 17-7

AHL 17-8

AIC 17-9

ARD 17-10

ARL 17-12

ASC 17-14

ASR 17-15

AWA 17-15

AWT 17-17

Atención

cambiar el valor del índice

13-6

colocación de contadores

críticos 2-15, 2-17

conversión de 32 en 16 bits
4-10

dirección indexada AVE
4-16

dirección indexada FAL 9-2

dirección indexada SRT 4-27

dirección indexada STD 4-30

direccionamiento de

estructura de control

10-4

direccionamiento indexado

8-2

estado de los bits BTR/BTW

15-7

FOR y NXT con

bifurcaciones 13-5

FOR y NXT sin

bifurcaciones de

salida 13-5

introducción de direcciones

de entrada 1-6, 1-7

modificar los bits de estado

de BTR/BTW 15-6

MSG

bits de estado .ST y .EW

15-25

parejar las instrucciones de

pila 11-6

PID

cambiar entradas o

unidades 14-19

cambiar la unidad de

ingeniería

máxima 14-22

cambiar la unidad de

ingeniería

mínima 14-22

cambio del escalado 14-6

cómo establecer los

límites de

temperatura

14-28

reanudar el último estado

14-10

tiempo de actualización

14-21

programación en línea con

ONS 13-14

programación en línea DTR

10-8

- restablecimiento de TON y TOF 2-8, 2-20
- temporizadores y contadores saltados 13-4
- uso de direcciones de control 8-2
- uso de direcciones de control para instrucciones 11-2
- uso de una dirección de control 12-3
- zona MCR
 - superposición o anidado 13-2
- zonas MCR
 - temporizadores y contadores 13-2
- B**
- bifuración seleccionada
 - secuencia de escán B-8
- bifurcación simultánea
 - secuencia de escán B-9
- bits de estado
 - instrucción CIO 15-24
- C**
- calcular
 - ADD 4-12
 - funciones 4-9
 - longitud de expresiones 4-7
 - ONS 13-14
 - orden de operación 4-8
- cálculo
 - ACS 4-11
 - ASN 4-13
 - ATN 4-14
 - AVE 4-15
 - CLR 4-17
 - COS 4-18
 - DEG 6-3
 - DIV 4-19
 - EOT 13-18
 - FSC 9-14
 - IOT 1-7
 - LN 4-20
 - LOG 4-21
 - MUL 4-22
 - NEG 4-23
 - RAD 6-4
 - SIN 4-24
 - SQR 4-25
 - SRT 4-26
 - STD 4-28
 - SUB 4-31
 - TAN 4-32
 - XPY 4-33
- CMP
 - instrucción 3-2
- cómo conectar los procesadores PLC-5 Ethernet con el uso de nombres del sistema central 16-6
- comparación
 - EQU 3-5
 - GEQ 3-5, 3-6
 - LEQ 3-6
 - LES 3-7
 - NEQ 3-10
- comparar
 - expresión 3-2
 - instrucciones 3-2
 - longitud de expresiones 3-3
- Compute
 - CPT 4-5
 - expresión 4-5
- comunicación directa
 - transferencia en bloques 15-2
- conceptos de archivo
 - estructura de control 8-2
 - manipulación de datos 8-3
 - modos de operación 8-5
- consejo
 - conexión a los procesadores Ethernet PLC-5 usando nombres de computadora principal 16-6
- constante de programa
 - rango de valor válido C-1
- constante-estado
 - tiempo de escán B-14
- constantes de ganancia 14-3
- constantes de programa
 - &H, &O, &B 3-2, 4-5
- contador
 - CTD 2-17
 - CTU 2-15
 - RES 2-20
- contadores

instrucciones 2-13
 control de proceso
 como establecer las alarmas
 de salida 14-7
 constantes de ganancia 14-3
 ecuación 14-2
 ejemplos de PID 14-29
 ejemplos de PID entero
 14-29
 ejemplos de PID PD 14-33
 instrucción PID 14-1
 PID 14-10
 polarización 14-9
 selección del término de
 derivada 14-7
 uniformidad de derivada
 14-4
 uso del límite de salida 14-7
 ControlNet I/O Transfer
 instrucción 15-22
 convergente
 tiempo de escán B-14
 conversión
 BCD 6-2
 FRD 6-2

D

diagnóstico
 DDT 10-2
 DTR 10-8
 estado 10-5
 FBC 10-2
 modo búsqueda 10-2
 parámetros 10-4, 10-8
 divergente
 tiempo de escán B-14

E

Enhanced PLC5 processors 1
 Entrada inmediata de datos
 instrucción 1-8
 escalado a unidades de
 ingeniería 14-5
 expresión
 como determinar la longitud
 3-3
 cómo determinar la longitud
 4-7

F

flujo de programa
 AFI 13-13
 JMP y LBL 13-3
 JSR, SBR y RET 13-8
 UID 13-19
 UIE 13-20
 flujo de programas
 MCR 13-2

I

información de estado
 SFC B-1
 instrucción
 ControlNet I/O Transfer
 15-22
 entrada inmediata de datos
 1-8
 salida inmediata de datos 1-8
 instrucción (ACB) 17-5
 instrucción ABL 17-4
 instrucción ACI 17-6
 instrucción ACN 17-7
 instrucción ACS 4-11
 instrucción ADD 4-12
 instrucción Addition (ADD)
 4-12
 instrucción AEX 17-7
 instrucción AFI 13-13
 instrucción AHL 17-8
 instrucción AIC 17-9
 instrucción Always False 13-13
 instrucción AND 5-2
 instrucción AND Operation
 AND 5-2
 instrucción Arc Cosine
 ACS 4-11
 instrucción Arc Sine
 ASN 4-13
 instrucción Arc Tangent
 ATN 4-14
 instrucción ARD 17-10
 instrucción ARL 17-12
 instrucción ASC 17-14
 instrucción ASCII Integer to
 String 17-9
 instrucción ASCII Read Line
 17-12

instrucción ASCII Set Handshake Lines 17-8	instrucción Convert from BCD FRD 6-2
instrucción ASCII String Compare 17-15	instrucción Convert to BCD TOD 6-2
instrucción ASCII String Concatenate 17-7	instrucción COP 9-19
instrucción ASCII String Extract 17-7	instrucción COS 4-18
instrucción ASCII String Search 17-14	instrucción Cosine COS 4-18
instrucción ASCII String to Integer 17-6	instrucción Count Down 2-17
instrucción ASCII Write 17-17	instrucción Count Up (CTU) 2-15
instrucción ASCII Write Append 17-15	instrucción CPT 4-5
instrucción ASN 4-13	instrucción CTD 2-17
instrucción ASR 17-15	instrucción CTU 2-15
instrucción ATN 4-14	instrucción Data Transitional DTR 10-8
instrucción AVE 4-15	instrucción DDT 10-2
instrucción Average File AVE 4-15	instrucción de flujo de programa FOR, BRK, LBL y RET 13-5
instrucción AWA 17-15	OSF 13-16
instrucción AWT 17-17	OSR 13-15
instrucción Bit Distribute BTD 7-2	SFR 13-17
instrucción Bit Shift Left (BSL) 11-2	instrucción de registro de desplazamiento aplicación 11-1
instrucción Bit Shift Right (BSR) 11-2	BSL y BSR 11-2
instrucción Block Transfer Read BTR 15-3	FFL y FFU 11-5
instrucción Block Transfer Write BTW 15-3	LFL y LFU 11-5
instrucción Break (BRK) 13-5	Instrucción de restablecimiento 2-20
instrucción BRK 13-5	instrucción DEG 6-3
instrucción BSL 11-2	instrucción Degree DEG 6-3
instrucción BSR 11-2	instrucción DFA 18-1
instrucción BTD 7-2	instrucción Diagnostic Detect DDT 10-2
instrucción BTR 15-3	instrucción Diagnostic Fault Annunciator (DFA) 18-1
instrucción BTW 15-3	instrucción DIV 4-19
instrucción CIO 15-22	instrucción Divide DIV 4-19
bits de estado 15-24	instrucción DTR 10-8
monitoreo 15-24	instrucción End of Transition EOT 13-18
uso 15-23	instrucción Energize Output (OTE) 1-4
instrucción Clear CLR 4-17	instrucción EOT 13-18
instrucción CLR 4-17	instrucción EQU 3-5
instrucción Compute CPT 4-5	

-
- instrucción Equal to (EQU) 3-5
 - instrucción Examine Off (XIO) 1-3
 - instrucción Examine On (XIC) 1-3
 - instrucción FBC 10-2
 - instrucción FFL 11-5
 - instrucción FFU 11-5
 - instrucción FIFO Load (FFL) 11-5
 - instrucción FIFO Unload (FFU) 11-5
 - instrucción File Arithmetic and Logic FAL 9-2
 - instrucción File Bit Comparison FBC 10-2
 - instrucción File Copy COP 9-19
 - instrucción File Fill FLL 9-20
 - instrucción File Search and Compare FSC 9-14
 - instrucción FLL 9-20
 - instrucción FOR 13-5
 - instrucción For (FOR) 13-5
 - instrucción FRD 6-2
 - instrucción FSC 9-14
 - instrucción GEQ 3-5
 - instrucción Greater Than or Equal To 3-5, 3-6
 - instrucción GRT 3-6
 - instrucción IDI 1-8
uso 1-9
 - instrucción IDO 1-8
uso 1-9
 - instrucción IIN 1-6
 - instrucción Immediate Input (IIN) 1-6
 - instrucción Immediate Output IOT 1-7
 - instrucción IOT 1-7
 - instrucción JMP 13-3
 - instrucción JSR 13-8
 - instrucción Jump 13-3
 - instrucción Jump to Subroutine 13-8
 - instrucción Label 13-3
 - instrucción Label (LBL) 13-5
 - instrucción LBL 13-3, 13-5
 - instrucción LEQ 3-6
 - instrucción LES 3-7
 - instrucción Less Than 3-7
 - instrucción Less Than or Equal To 3-6
 - instrucción LFL 11-5
 - instrucción LFU 11-5
 - instrucción LIFO Load (LFL) 11-5
 - instrucción LIFO Unload (LFU) 11-5
 - instrucción LIM 3-7
 - instrucción Limit Test (LIM) 3-7
 - instrucción LN 4-20
 - instrucción Log to the base 10 LOG 4-21
 - instrucción lógica FAL 9-12
 - instrucción Masked Comparison Equal to (MEQ) 3-9
 - instrucción Masked Move 7-4
 - instrucción Master Control Reset 13-2
 - instrucción MCR 13-2
 - instrucción MEQ 3-9
 - instrucción MOV 7-3
 - instrucción Move MOV 7-3
 - instrucción MSG 16-1
uso 16-10
 - instrucción MUL 4-22
 - instrucción Multiply MUL 4-22
 - instrucción MVM 7-4
 - instrucción Natural Log LN 4-20
 - instrucción NEG 4-23
 - instrucción Negate NEG 4-23
 - instrucción NEQ 3-10
 - instrucción Next (NXT) 13-5
 - instrucción NOT 5-3
 - instrucción Not Equal To (NEQ) 3-10
 - instrucción NOT Operation NOT 5-3
 - instrucción Number of Char in Buffer 17-5
 - instrucción NXT 13-5

- instrucción One Shot
ONS 13-14
- instrucción One Shot Falling
(OSF) 13-16
- instrucción One Shot Rising
(OSR) 13-15
- instrucción ONS 13-14
- instrucción OR 5-4
- instrucción OR Operation
OR 5-4
- instrucción OSF 13-16
- instrucción OSR 13-15
- instrucción OTE 1-4
- instrucción OTL 1-4
- instrucción OTU 1-5
- instrucción Output Latch (OTL)
1-4
- instrucción Output Unlatch 1-5
- instrucción PID 14-1
- instrucción proporcional,
integral y de derivada
14-10
- instrucción RAD 6-4
- instrucción Radian
RAD 6-4
- instrucción RES 2-20
- instrucción RET 13-8
- instrucción Retentive Timer On
(RTO) 2-10
- instrucción Return 13-8
- instrucción RTO 2-10
- instrucción SBR 13-8
- instrucción SDS 18-1
- instrucción Sequencer Input 12-2
- instrucción Sequencer Load 12-2
- instrucción Sequencer Output
12-2
- instrucción Sequential Function
Chart Reset 13-17
- instrucción SFR 13-17
- instrucción SIN 4-24
- instrucción Sine
SIN 4-24
- instrucción Smart Directed
Sequencer 18-1
- instrucción Smart Directed
Sequencer (SDS)
programación 18-2
- instrucción Sort File
SRT 4-26
- instrucción SQI 12-2
- instrucción SQL 12-2
- instrucción SQO 12-2
- instrucción SQR 4-25
- instrucción Square Root
SQR 4-25
- instrucción SRT 4-26
- instrucción Standard Deviation
STD 4-28
- instrucción STD 4-28
- instrucción SUB 4-31
- instrucción Subroutine Header
13-8
- instrucción Subtract
SUB 4-31
- instrucción Tangent
TAN 4-32
- instrucción Temporary End
13-20
- instrucción Test Buffer For Line
17-4
- instrucción Timer Off Delay
(TOF) 2-7
- instrucción Timer On Delay
(TON) 2-4
- instrucción TND 13-19, 13-20
- instrucción TOD 6-2
- instrucción TOF 2-7
- instrucción TON 2-4
- instrucción X to the Power of Y
XPY 4-33
- Instrucción XIC 1-3
- instrucción XIO 1-3
- instrucción XOR 5-5
- instrucción XOR Operation
XOR 5-5
- instrucción XPY 4-33
- instrucciones
 - ASCII 17-1
 - CIO
 - monitoreo 15-24
 - comparar 3-2
 - diagnóstico 10-1
 - flujo de programas 13-1
 - mensaje 16-1
 - operandos C-1
 - registro de desplazamiento
11-1
 - requisitos de memoria A-1
 - secuenciador 12-1

- temporización A-1
- temporizador 2-1
- tipo relé 1-1, 2-1
- transferencia en bloques 15-1
- instrucciones ASCII
 - cadena 17-3
- instrucciones ASCII Read Characters 17-10
- instrucciones de archivo
 - lógicas 9-12
- instrucciones de diagnóstico
 - 10-1
- instrucciones de flujo de programa 13-1
- INVALID OPERAND
 - mensaje de error 4-4
- L**
- LOG
 - instrucción 4-21
- lógico
 - AND 5-2
 - NOT 5-3
 - OR 5-4
 - XOR 5-5
- M**
- manipulación
 - datos de archivo 8-3
- manipulación de elemento
 - MVM 7-4
- manipulación de elementos
 - LIM 3-7
 - MEQ 3-9
 - MOV 7-3
- memoria
 - requisitos de instrucción A-1
 - requisitos SFC B-3
- mensaje
 - instrucción 16-1
- mensaje de error
 - INVALID OPERAND 4-4
- modo de escán de E/S
 - transferencia en bloques 15-1
- modo escáner
 - configuración 15-13, 15-14
- modo incremental 8-7
- modos
 - operación de archivo 8-5
- monitoreo
 - instrucciones CIO 15-24
- MSG
 - introducción de instrucciones 16-10
- O**
- operandos
 - instrucciones C-1
- orden de operación 4-8
- P**
- paso
 - secuencia de escán B-7
- PID
 - cómo establecer las alarmas de salida 14-7
 - cómo seleccionar el término de derivada 14-7
 - cómo usar el límite de salida 14-7
 - cómo usar el modo manual 14-8
 - ecuaciones 14-2
 - ejemplos 14-29
 - ejemplos de número entero 14-29
 - ejemplos PD 14-33
 - instrucción 14-10
 - polarización 14-9
- procesador
 - PLC5/10 1
 - PLC5/60C 1
 - PLC5/80 1
 - PLC5/80E 1
 - PLC5/86 1
 - PLC5/V30 1
 - PLC5/VME 1
- Procesadores PLC-5 clásicos 1
- Procesadores PLC-5 ControlNet 1
- Procesadores PLC-5 Ethernet 1
- Procesadores PLC-5 protegidos 1
- Procesadores PLC-5 VME 1
- proceso de control
 - uso del modo manual 14-8
- processor
 - PLC5/11 1
 - PLC5/12 1
 - PLC5/15 1

PLC5/20 1	bifuración simultánea
PLC5/20C 1	B-9
PLC5/20E 1	ejemplo B-11
PLC5/25 1	secuencias de escán
PLC5/26 1	paso/transición B-7
PLC5/30 1	Smart Directed Sequencer (SDS)
PLC5/40 1	descripción general 18-2
PLC5/40C 1	
PLC5/40E 1	T
PLC5/40L 1	Temporary End
PLC5/46 1	instrucción 13-13
PLC5/60 1	temporización
PLC5/60L 1	instrucciones A-1
PLC5/80C 1	transferencia en bloques
PLC5/V40 1	15-13, 15-14
PLC5/V40L 1	temporizador
PLC5/V80 1	instrucción TON 2-4
PLC5/VME 1	parámetros de instrucción
programación	2-13
instrucción SDS 18-2	parámetros de la instrucción
programación	2-2
instrucciones	precisión 2-3
operandos C-1	RES 2-20
punto flotante (coma flotante)	RTO 2-10
rango de valor válido C-1	TOF 2-7
R	temporizadores 2-1
restricciones	tiempo de escán
SFC B-5	constante-estado B-14
rutina de aplicación	convergente B-14
personalizada 18-1	divergente B-14
S	tiempos de ejecución
salida inmediata de datos	determinación B-12
instrucciones 1-8	tipo relé
secuencia de escán	IIN 1-6
SFC B-7	OTE 1-4
secuenciador	OTL 1-4
aplicación 12-1	OTU 1-5
instrucciones 12-1	XIC 1-3
SQI 12-2	XIO 1-3
SQL 12-2	TND
SQO 12-2	instrucción 13-13
SFC	transferencia en bloques
ejemplo	ejemplos de programación
secuencia de escán B-11	15-15
requisitos de memoria B-3	instrucción BTR 15-3
secuencia de escán	instrucción BTW 15-3
bifuración seleccionada	instrucciones 15-1
B-8	modo de comunicación
	directa 15-2
	modo escán de E/S 15-1

temporización 15-13, 15-14
transición
secuencia de escán B-7

U

unidades de ingeniería
 escalado 14-5
unidades, ingeniería
 escalado 14-5
uniformidad de derivada 14-4
User Interrupt Disable
 UID 13-19
User Interrupt Enable
 UIE 13-20
uso
 instrucción CIO 15-23
 instrucción IDI 1-9
 instrucción IDO 1-9
 instrucción MSG 16-10
utilidad CAR 18-1

Servicio al cliente

Si usted necesita ayuda adicional relativa al software, Allen-Bradley ofrece soporte de productos por teléfono y en la planta mediante nuestros Centros de Servicio al Cliente ubicados en todo el mundo.

Si requiere ayuda técnica por teléfono, comuníquese primero con la oficina de ventas, distribuidor o integrador de sistemas regional. Si necesita ayuda adicional, le rogamos comunicarse con el Centro de Servicio al Cliente regional o los Servicios de Soporte de Sistemas.

En EE.UU. y Canadá

Si tiene un convenio SupportPlus o si su software está protegido por una garantía, puede comunicarse con los Servicios de Soporte de Sistemas al número siguiente: 1-800-289-2279. Tenga a mano el número de contrato o el número de registro del software.

Para obtener asistencia técnica in situ, comuníquese con la oficina de ventas, distribuidor o integrador de sistemas regionales. Puede llamar a la línea de ayuda de 24 horas de Allen-Bradley al 1-800-422-4913 fuera de las horas hábiles.

Otros países

Llame a su Centro de Servicio al Cliente regional al número siguiente:

Región o zona	Número de teléfono del Centro de Servicio al Cliente
Canadá (Cambridge, Ontario)	519-623-1810
América Latina (México)	52-5-259-0040
Reino Unido (Milton Keynes)	44-908 838800
Francia (París)	(33-1) 3067-7200
Alemania (Gruiten)	(49) 2104 6900
Italia (Milán)	(39-2) 939-721
Asia/Pacífico (Hong Kong)	(852) 887-4788
España (Barcelona)	(34-3) 331-7004

Para obtener asistencia técnica en la planta, comuníquese con la oficina de ventas, distribuidor o integrador de sistemas regionales. Puede comunicarse con el Centro de Servicio al Cliente regional fuera de las horas hábiles.

Argentina

Rockwell de Argentina, Av. Córdoba 4970, 1414 Buenos Aires, Argentina
Tel: (54-1) 776-1100, Fax: (54-1) 773-5175

Colombia

Rockwell Colombia, S.A., Muelle Industrial II, Bodega 4, Cr. 98, No. 42A-41, Santafé de Bogotá DF,
Tel: (57-1) 418-5902, Fax: (57-1) 418-5995

España

Rockwell Automation, S.A., Calle Doctor Trueta, 113-119, 08005 Barcelona,
Tel: (34-3) 295 90 00, Fax: (34-3) 295 90 01

Rockwell Automation, Villa de Plencia, 4, Urbanización Antiguo Golf, 48930 Las Arenas - Getxo,
Vizcaya, Tel: (34-4) 480 16 81, Fax: (34-4) 480 09 16

Rockwell Automation, Belmonte de Tajo, 31, 28019 Madrid, Tel: (34-1) 565 16 16, Fax: (34-1) 565 16 87

Rockwell Automation, Avda. San Francisco Javier, 9, Ed. Sevilla 2-Planta 5, Mod. 26A, 41018 Sevilla,
Tel: (34-5) 466 35 512, Fax: (34-5) 465 62 58

Rockwell Automation, Edificio Trevi, Fontanares, 51-4º D,E, 46014 Valencia,
Tel: (34-6) 377 06 12, Fax: (34-6) 377 07 61

México

Rockwell Automation de México, S.A. de C.V., Bosques de Ciruelos No. 160, Col. Bosques de
Las Lomas, C.P. 11700, México, DF., México, Tel: (52-5) 251-6161, Fax: (52-5) 251-1169

Rockwell Automation de México, S.A. de C.V., J. Sebastian Bach No. 4986 Esq. Av. Patria, Col. Prados
Guadalupe, C.P. 45030, Zapopan, Jalisco, México, Tel: (52-36) 732-997, Fax: (52-36) 732-957

Rockwell Automation de México, S.A. de C.V., Calle San Pedro No. 10, Fracc. Capistrano, 4a. Etapa,
C.P. 83240, Hermosillo, Son., México, Tel: (52-62) 60-40-79, Fax: (52-62) 60-40-79

Rockwell Automation de México, S.A. de C.V., 41 Oriente No. 2214, Col. El Mirador, C.P. 72530,
Puebla, Pue., México, Tel: (52-22) 455-329, Fax: (52-22) 455-548

Rockwell Automation de México, S.A. de C.V., Av. Pablo A. González 130 Pte., Col. San Jerónimo,
C.P. 64630, Monterrey, N.L., México, Tel: (52-83) 483-832, Fax: (52-83) 476-178

Rockwell Automation de México, S.A. de C.V., Av. Ramón Rodríguez Familiar 5, Col. Bosques de
Acueducto, C.P. 76020, Querétaro, Qro., México, Tel: (52-42) 134-884, Fax: (52-42) 135-798

Venezuela

Rockwell Automation de Venezuela, Edif. Allen-Bradley, Av. González Rincones, Zona. Ind. La
Trinidad, Caracas 1080, Venezuela, Tel: (58-2) 943-2311, Fax: (58-2) 943-3955

Rockwell Automation de Venezuela, Av. 3C con calle 67, Unicentro Virginia, Ofic. 2-4, Maracaibo, Edo.
Zulia, Venezuela, Tel: (58-61) 92-2813, Fax: (58-616) 92-2880

Rockwell Automation de Venezuela, Centro Comercial, Plaza Mayor, Sector 6, Ofic. 251-252, Prol.
Paseo Colón, Lecherías, Edo. Barcelona, Venezuela, Tel: (58-81) 81-0366, Fax: (58-81) 81-5677

Rockwell Automation de Venezuela, Urbanización Prebo, Residencias Avisá, Piso 9, Apto. 9A,
Valencia, Edo. Carabobo, Venezuela, Tel: (58-41) 22-3383, Fax: (58-41) 22-3383



Rockwell Automation ayuda a sus clientes a lograr mejores ganancias de sus inversiones integrando marcas líder de la automatización industrial y creando así una amplia gama de productos de integración fácil. Estos productos disponen del soporte de proveedores de soluciones de sistema además de los recursos de tecnología avanzada de Rockwell.



Con oficinas en las principales ciudades del mundo.

Alemania • Arabia Saudita • Argentina • Australia • Bahrein • Bélgica • Bolivia • Brasil • Bulgaria • Canadá • Chile • Chipre • Colombia • Corea del Sur • Costa Rica • Croacia
Dinamarca • Ecuador • Egipto • El Salvador • Emiratos Arabes Unidos • Eslovaquia • Eslovenia • España • Estados Unidos • Finlandia • Francia • Ghana • Grecia • Guatemala
Holanda • Honduras • Hong Kong • Hungría • India • Indonesia • Irlanda • Islandia • Israel • Italia • Jamaica • Japón • Jordania • Katar • Kenia • Kuwait • Las Filipinas • Líbano
Macao • Malasia • Malta • Mauricio • México • Marruecos • Nigeria • Noruega • Nueva Zelanda • Omán • Pakistán • Panamá • Perú • Polonia • Portugal • Puerto Rico • Reino
Unido • República Checa • República de Sudáfrica • República Dominicana • República Popular China • Rumania • Rusia • Singapur • Suecia • Suiza • Taiwan • Tailandia
Trinidad y Tobago • Turquía • Uruguay • Venezuela • Vietnam • Zimbabue

Sede central de Rockwell Automation: 1201 South Second Street, Milwaukee, WI 53204, USA, Tel: (1) 414-382-2000, Fax: (1) 414-382-4444

Sede central europea de Rockwell Automation: avenue Herrmann Debroux, 46, 1160 Bruselas, Bélgica, Tel: (32) 2 663 06 00, Fax: (32) 2 663 06 40

Publicación 1785-6.1ES Noviembre de 1998

Reemplaza la publicación 1785-6.1ES Febrero de 1996

PN 957280-51

© 1998 Rockwell International Corporation.