



DIVISIÓN DE INGENIERÍA DE SISTEMAS Y AUTOMÁTICA

AUTÓMATAS PROGRAMABLES

Programación del S7-200

Operaciones básicas



Autómatas Programables
ISA-UMH © TDOC-2000

1

Indice

- Estructura interna de la familia de autómatas S7-200
 - Distribución de la memoria
 - Direccionamiento directo de la memoria de la CPU
 - Entradas y salidas integradas y ampliadas mediante módulos de expansión.
 - Ciclo de Programa
- Lenguajes de programación
 - SIMATIC/IEC 1131-1
 - KOP, AWL
- Operaciones básicas(KOP)
 - Operaciones con contactos
 - Operaciones con salidas

Autómatas Programables
ISA-UMH © TDOC-2000

2

Indice

- Operaciones con temporizadores
- Operaciones con contadores
- Operaciones de comparación

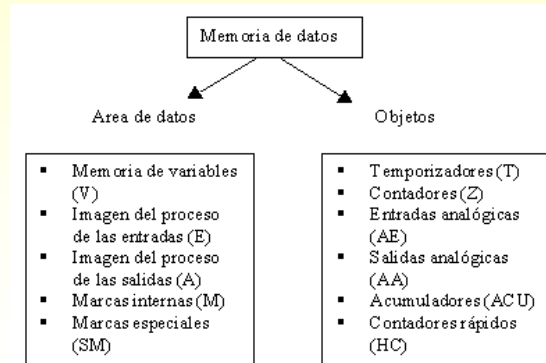
Distribución de la memoria

- Memoria de programa
 - La memoria de programa contiene las operaciones de esquema de contactos (KOP) o de lista de asignación (AWL), que ejecuta el autómata programable para la aplicación deseada.
- Memoria de parámetros
 - La memoria de parámetros permite almacenar determinados parámetros configurables, tales como contraseñas, direcciones de estaciones e informaciones sobre las áreas remanentes

Distribución de la memoria

■ Memoria de datos

- La memoria de datos es el área de trabajo a la que accede el programa de aplicación (también denominado programa de usuario).

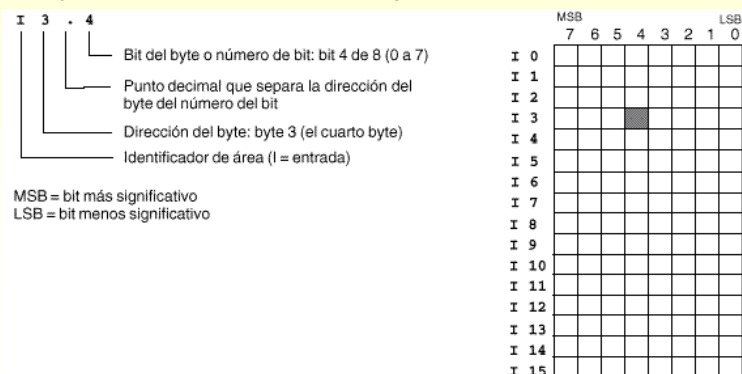


Direccionamiento directo de la memoria de la CPU

■ Acceso a un bit

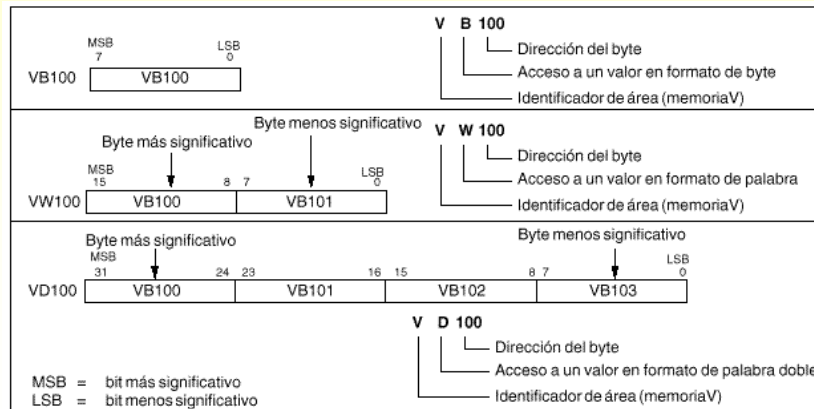
- “Identificador de área” “dirección del byte” . “nº del bit”

Ejemplo I 0.0 el bit 0 del byte 0 de las entradas



Direccionamiento directo de la memoria de la CPU

- Se puede acceder a diversas áreas de la memoria de la CPU (V, I, Q, M, SM) en formato byte, palabra y palabra doble

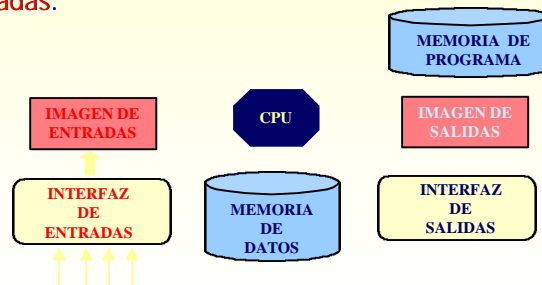


Autómatas Programables
ISA-UMH © TDOC-2000

7

Direccionamiento directo de la memoria de la CPU Ciclo de tratamiento de las señales de entrada/salida

- Ciclo de tratamiento de las señales de entrada/salida a través de las memorias imagen:
 - Antes de la ejecución del programa de usuario, la CPU consulta los estados de las **entradas físicas** y carga con ellos la **memoria imagen de entradas**.



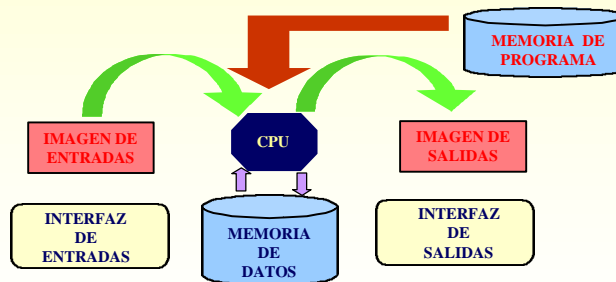
Autómatas Programables
ISA-UMH © TDOC-2000

8

Direccionamiento directo de la memoria de la CPU

Ciclo de tratamiento de las señales de entrada/salida

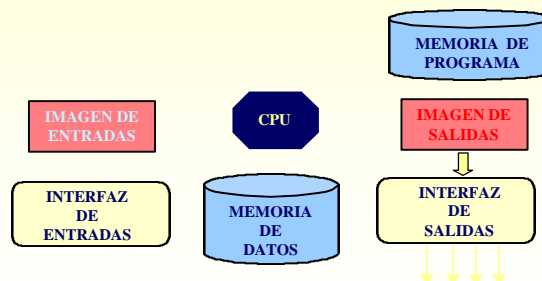
- Durante la ejecución del programa de usuario, la CPU realiza los cálculos a partir de los datos de la memoria imagen y del estado de los temporizadores, contadores y relés internos. El resultado de estos cálculos queda depositado en la memoria **imagen de salidas**.



Direccionamiento directo de la memoria de la CPU

Ciclo de tratamiento de las señales de entrada/salida

- Finalizada la ejecución, la CPU transfiere a las **interfaces de salida** los estados de las señales contenidos en la **memoria imagen de salidas**, quedando el sistema preparado para comenzar un nuevo ciclo



Direccionamiento directo de la memoria de la CPU

- Direccionamiento de la imagen del proceso de las entradas (**I**)
 - Formato:
 - Bit I [direcc. del byte].[direcc. del bit] I0.1
 - Byte, palabra, palabra doble I [tamaño][direcc. del byte inicial] IB4
- Direccionamiento de la imagen del proceso de las salidas (**Q**)
 - Formato:
 - Bit Q [direcc. del byte].[direcc. del bit] Q1.1
 - Byte, palabra, p. doble Q [tamaño][direcc. del byte inicial] QB5
- Direccionamiento de la memoria de variables (**V**)
 - Formato:
 - Bit V [direcc. del byte].[direcc. del bit] V10.2
 - Byte, palabra, p. Doble V [tamaño][direcc. del byte inicial] VW100

Direccionamiento directo de la memoria de la CPU

- Direccionamiento del área de marcas (**M**)
 - Las marcas internas (área de marcas M) se pueden utilizar como relés de control para almacenar el estado intermedio de una operación u otras informaciones de control
 - Formato:
 - Bit M [direcc. del byte].[direcc. del bit] M26.7
 - Byte, palabra, p. Doble M [tamaño][direcc. del byte inicial] MD20

Direccionamiento directo de la memoria de la CPU

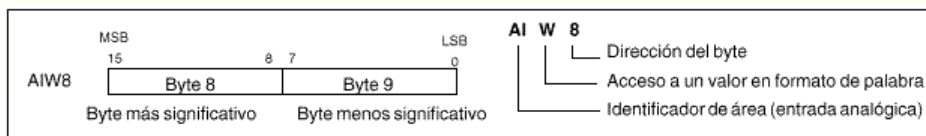
■ Direccionamiento de las marcas especiales (**SM**)

- Las marcas especiales permiten intercambiar datos entre la CPU y el programa. Dichas marcas se puede utilizar para seleccionar y controlar algunas funciones especiales de la CPU S7-200, tales como:
 - Un bit que se activa sólo en el primer ciclo.
 - Bits que se activan y se desactivan en determinados intervalos.
 - Bits que muestran el estado de operaciones matemáticas y de otras operaciones.
- Aunque el área de las marcas especiales se basa en bits, es posible acceder a los datos en formato de bit, byte, palabra o palabra doble.
- Formato:
 - Bit **SM [direcc. del byte].[direcc. del bit] M0.1**
 - Byte, palabra, p. Doble **SM [tamaño][direcc. del byte inicial] SMB86**

Direccionamiento directo de la memoria de la CPU

■ Direccionamiento de las entradas analógicas (**AI**)

- La CPU S7-200 convierte valores reales analógicos (p.ej. temperatura, tensión, etc). en valores digitales en formato de palabra (de 16 bits).
 - Puesto que las entradas analógicas son palabras que comienzan siempre en bytes pares (p.ej. 0, 2, 4, etc)., es preciso utilizar direcciones con bytes pares (p.ej. AIW0, AIW2, AIW4, etc)
- Formato:
 - AIW [dirección del byte inicial] AIW4



Direccionamiento directo de la memoria de la CPU

- Direccionamiento de las salidas analógicas (AQ)
 - La CPU S7-200 convierte valores digitales en formato de palabra (de 16 bits) en valores reales analógicos (p.ej. corriente o voltaje), proporcionales al valor digital.
 - Puesto que las salidas analógicas son palabras que comienzan siempre en bytes pares (p.ej. 0, 2, 4, etc)., es preciso utilizar direcciones con bytes pares (p.ej. AQW0, AQW2, AQW4, etc). para acceder a las mismas.
 - Formato:
 - AQW [dirección del byte inicial] AQW4



Autómatas Programables
ISA-UMH © TDOC-2000

15

Entradas y salidas integradas y ampliadas mediante módulos de expansión.

- entradas y salidas integradas (en la CPU), así como de E/S adicionales (en los módulos de ampliación).
- Direccionar las E/S integradas y adicionales
 - Las entradas y salidas integradas en la unidad central de procesamiento (CPU) tienen direcciones fijas.
 - Las direcciones de las E/S de cada módulo vienen determinadas por
 - el tipo de E/S
 - la posición del módulo en la cadena, con respecto al anterior módulo de entradas o de salidas del mismo tipo
 - Por ejemplo, un módulo de salidas no afecta las direcciones de un módulo de entradas y viceversa.
 - los módulos analógicos no afectan al direccionamiento de los módulos digitales y viceversa.

Autómatas Programables
ISA-UMH © TDOC-2000

16

Entradas y salidas integradas y ampliadas mediante módulos de expansión.

- Los módulos de ampliación digitales reservan siempre un espacio de la imagen del proceso en incrementos de ocho bits (un byte).
 - Si un módulo no dispone de un punto físico para cada bit de cada byte reservado, se pierden estos bits no utilizados y no se pueden asignar a los módulos siguientes en la cadena de E/S.
 - En cuanto a los módulos de entradas, los bits no utilizados en los bytes reservados se ponen a cero cada vez que se actualizan las entradas.
- Las direcciones de los módulos de ampliación analógicos se asignan siempre en incrementos de dos puntos.
 - Si un módulo no ofrece E/S físicas para cada uno de dichos puntos, se pierden los mismos y no se pueden asignar a los módulos siguientes en la cadena de E/S.

Entradas y salidas integradas y ampliadas mediante módulos de expansión.

■ EJEMPLOS

CPU 221	CPU 224	4 En / 4 Sal	8 Entradas	8 Salidas
I0.0 Q0.0	I0.0 Q0.0	I2.0 Q2.0	I3.0	Q3.0
I0.1 Q0.1	I0.1 Q0.1	I2.1 Q2.1	I3.1	Q3.1
I0.2 Q0.2	I0.2 Q0.2	I2.2 Q2.2	I3.2	Q3.2
I0.3 Q0.3	I0.3 Q0.3	I2.3 Q2.3	I3.3	Q3.3
I0.4	I0.4 Q0.4		I3.4	Q3.4
I0.5	I0.5 Q0.5		I3.5	Q3.5
	I0.6 Q0.6		I3.6	Q3.6
	I0.7 Q0.7		I3.7	Q3.7
	I1.0 Q1.0			
	I1.1 Q1.1			
	I1.2			
	I1.3			
	I1.4			
	I1.5			

Ciclo de Programa

- A la ejecución cíclica de tareas que realiza la CPU del autómata cuando se pone este en el modo RUN se le denomina "Ciclo" de programa.
 - Leer las entradas digitales
 - Ejecutar el programa de usuario
 - Procesar las peticiones de comunicación
 - Efectuar un autodiagnóstico
 - Escribir en las salidas digitales



SIMATIC/IEC 1131-1

- SIMATIC
 - juego de instrucciones ofrecido por Siemens
- IEC 1131-3
 - estándar en la programación de autómatas, la Comisión Electrotecnia Internacional (CEI) o International Electrotechnical Commission (IEC)
 - Sólo en KOP o en FUP
- INCONVENIENTES DE USAR IEC 1131-3
 - se dispone de un menor número de operaciones que con el juego de operaciones SIMATIC
 - el tiempo de ejecución de las operaciones IEC 1131-3 es más largo
 - el funcionamiento de algunas operaciones es diferente al de su equivalente en SIMATIC.

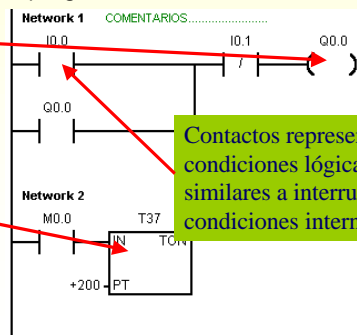
KOP-AWL

■ KOP

- la lógica se divide en unidades pequeñas y de fácil comprensión llamadas "segmentos" o "networks"
- El programa se ejecuta segmento por segmento, de izquierda a derecha y luego de arriba a abajo.
- Tras alcanzar la CPU el final del programa, comienza nuevamente en la primera operación del mismo

Bobinas representan condiciones lógicas de "salida" similares a lámparas, arrancadores de motor, relés interpuestos, condiciones internas de salida, etc.

Cuadros representan operaciones adicionales tales como temporizadores, contadores u operaciones aritméticas.



Contactos representan condiciones lógicas de "entrada" similares a interruptores, botones, condiciones internas, etc.

KOP-AWL

■ Ventajas del KOP

- El lenguaje KOP les facilita el trabajo a los programadores principiantes.
- La representación gráfica es a menudo fácil de comprender, siendo popular en el mundo entero.
- El editor KOP se puede utilizar con los juegos de operaciones SIMATIC e IEC 1131-3.
- El editor AWL siempre se puede utilizar para visualizar un programa creado en KOP.

KOP-AWL

■ AWL

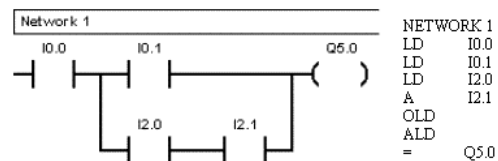
- permite crear programas de control introduciendo la nemotécnica de las operaciones.
- El editor AWL también permite crear ciertos programas que, de otra forma, no se podrían programar con los editores KOP ni FUP.
- Ello se debe a que AWL es el lenguaje nativo de la CPU, a diferencia de los editores gráficos en los que son aplicables ciertas restricciones para poder dibujar los diagramas correctamente.

```

NETWORK 1
LD  I0.0
LD  I0.1
LD  I2.0
A   I2.1
OLD
ALD
=   Q5.0
    
```

KOP-AWL

- Se utiliza una pila lógica de 9 bits de profundidad y uno de ancho.



```

NETWORK 1
LD  I0.0
LD  I0.1
LD  I2.0
A   I2.1
OLD
ALD
=   Q5.0
    
```

Pila	LD I0.0	LD I0.1	LD I2.0	A	OLD	ALD
S0	I0.0	I0.1	I2.0	I2.0 & I2.1	(I2.0 & I2.1) OR I0.1	I0.0 AND [(I2.0 & I2.1) OR I0.1]
S1		I0.0	I0.1	I0.1	I0.0	
S2			I0.0	I0.0		
S3						
S4						
S5						
S6						
S7						
S8						

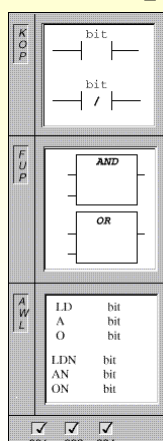
KOP-AWL

■ AWL. Consideraciones

- Más apropiado para los programadores expertos
- En algunos casos AWL permite solucionar problemas que no podrían solucionarse con KOP o FUP
- El juego de operaciones SIMATIC sólo se puede utilizar con el editor AWL. Para AWL no se dispone de un juego de operaciones IEC.
- En tanto que el editor AWL se puede utilizar siempre para ver o editar un programa creado con los editores KOP o FUP SIMATIC, lo contrario no es posible en todos los casos.

Operaciones con contactos

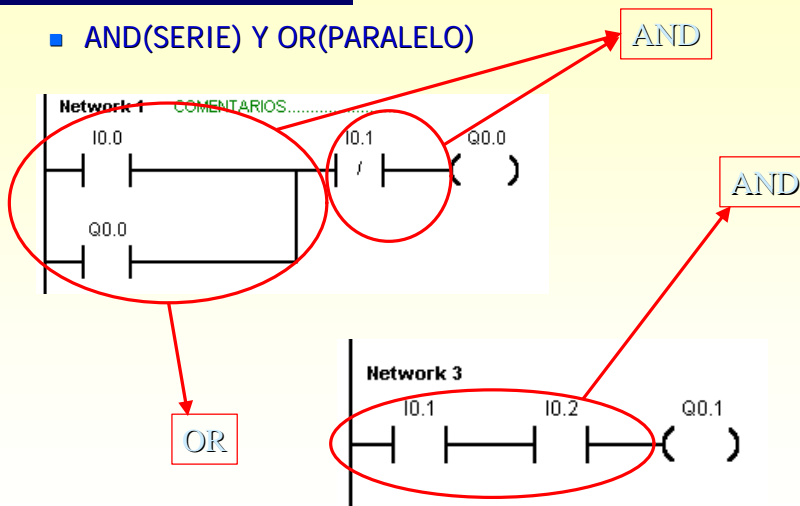
■ Contactos estándar



- El **contacto abierto** (-| |-) se cierra (se activa) si el valor binario de la dirección $n = 1$.
 - En AWL, **Cargar** (LD), **Y** (AND) y **O** (OR). Dichas operaciones cargan el valor binario de la dirección n en el nivel superior de la pila.
- El **contacto cerrado** (-| / |-) se cierra (se activa) si el valor binario de la dirección $n = 0$.
 - En AWL, **Cargar valor negado** (LDN), **Y-NO** (AN) y **O-NO** (ON). Dichas operaciones cargan el valor binario invertido de la dirección n en el nivel superior de la pila.
- Operandos:
 - n : I, Q, M, SM, T, C, V, S

Operaciones con contactos

■ AND(SERIE) Y OR(PARALELO)

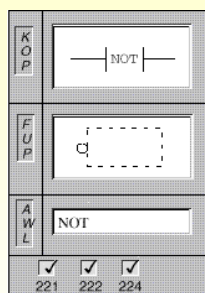


Autómatas Programables
ISA-UMH © TDOC-2000

27

Operaciones con contactos

■ NOT



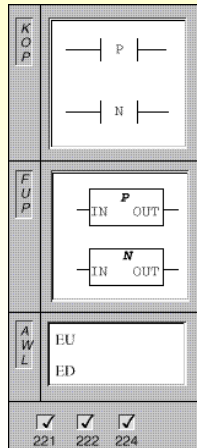
- El contacto **NOT** invierte el sentido de circulación de la corriente. La corriente se detiene al alcanzar el contacto NOT. Si no logra alcanzar el contacto, entonces hace circular la corriente.
 - En AWL, la operación **Invertir primer valor** (NOT) invierte el primer valor de la pila de 0 a 1, o bien de 1 a 0.
- Operandos:
 - ninguno

Autómatas Programables
ISA-UMH © TDOC-2000

28

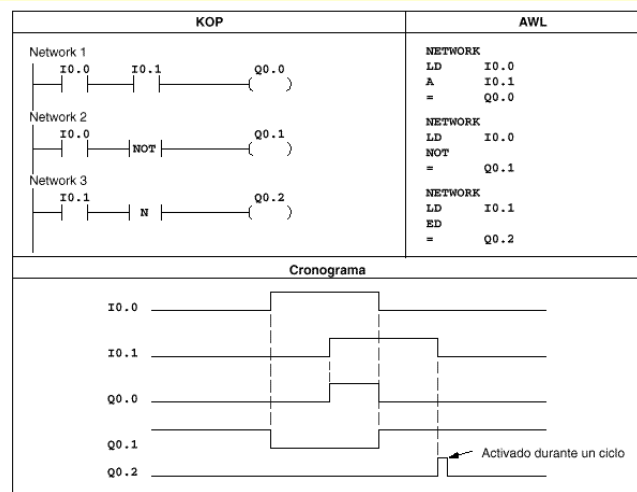
Operaciones con contactos

■ Detectar flanco positivo y negativo



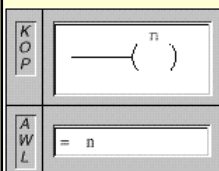
- El contacto **Detectar flanco positivo** permite que fluya la corriente durante un ciclo cada vez que se produce un cambio de 0 a 1 (de "off" a "on").
 - En AWL, **Detectar flanco positivo** (EU). Cuando se detecta un cambio de señal de 0 a 1 en el primer valor de la pila, éste se pone a 1. En caso contrario, se pone a 0.
- El contacto **Detectar flanco negativo** permite que fluya la corriente durante un ciclo cada vez que se produce un cambio de 1 a 0 (de "on" a "off").
 - En AWL, **Detectar flanco negativo** (ED). Cuando se detecta un cambio de señal de 1 a 0 en el primer valor de la pila, éste se pone a 1. En caso contrario, se pone a 0.
- Operandos:
 - ninguno

Operaciones con contactos



Operaciones con salidas

■ Asignar

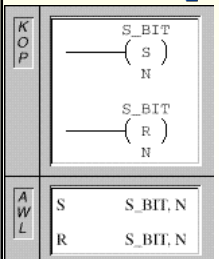


– Al ejecutar la operación Asignar se activa el parámetro indicado (n). En AWL, la operación Asignar (=) copia el primer valor de la pila en el parámetro indicado (n).

– Operandos:

- n: I, Q, M, SM, T, C, V, S

■ Poner a 1, Poner a cero



– Al ejecutar las operaciones **Poner a 1** y **Poner a 0**, se activa (se pone a 1) o se desactiva (se pone a 0) el número indicado de entradas y/o salidas (N) a partir de S_BIT, respectivamente.

– Operandos:

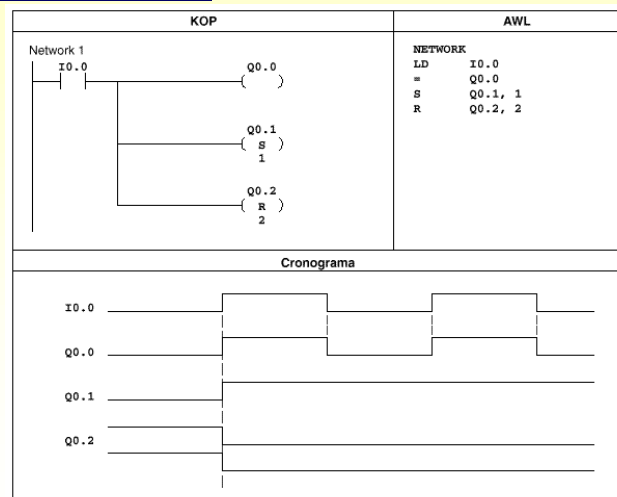
- S_BIT: I, Q, M, SM, T, C, V, S
- N: IB, QB, MB, SMB, VB, AC, constante, *VD, *AC, SB

– N entre 1 y 255

Autómatas Programables
ISA-UMH © TDOC-2000

31

Operaciones con salidas

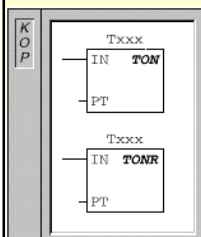


Autómatas Programables
ISA-UMH © TDOC-2000

32

Operaciones con temporizadores

- Temporizador de retardo a la conexión, Temporizador de retardo a la conexión memorizado

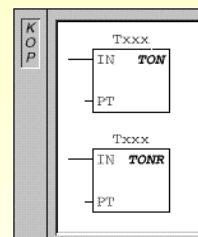


- Las operaciones **Temporizador de retardo a la conexión** y **Temporizador de retardo a la conexión memorizado** empiezan a contar hasta el valor máximo al ser habilitadas. Si el valor actual (Txxx) es mayor o igual al valor de preselección (PT), se activa el bit de temporización.
- Cuando se inhibe la operación, el temporizador de retardo a la conexión se pone a 0, en tanto que el temporizador de retardo a la conexión memorizado se detiene. Ambos temporizadores se detienen al alcanzar el valor máximo.

Operaciones con temporizadores

- Operandos:

Txxx:	TON	TONR
1 ms	T32, T96	T0, T64
10 ms	T33 a T36	T1 a T4
	T97 a T100	T65 a T68
100 ms	T37 a T63	T5 a T31
	T101 a T255	T69 a T95
PT:	VW, T, C, IW, QW, MW, SMW, AC, AIW, constante, *VD, *AC, SW	



- Por ejemplo, el valor de conteo 50 en un temporizador de 10 milisegundos (ms) equivale a 500 ms.

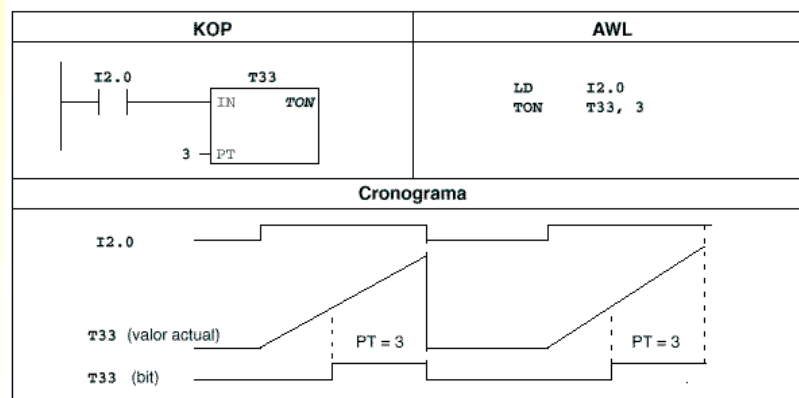
Operaciones con temporizadores

■ Actualización

- Dependiendo de sus resoluciones, también los temporizadores son actualizados en diferentes momentos
 - 1ms: mediante una interrupción !Prolongan el ciclo un 5 0 6% más)
 - 10ms: al inicio del ciclo
 - 100ms: durante la ejecución de la función de temporización
- Valor actual y el bit T de un temporizador
 - 1ms pueden actualizarse varias veces durante un solo ciclo
 - 100ms en una subrutina que no se ejecuta todos los ciclos no se actualizará correctamente ni el valor actual ni el bit T.
 - 100ms que se habilita varias veces en un ciclo hace que el valor de conteo de un mismo ciclo se sume repetidas veces. Sólo cuando se ejecuten una vez por ciclo

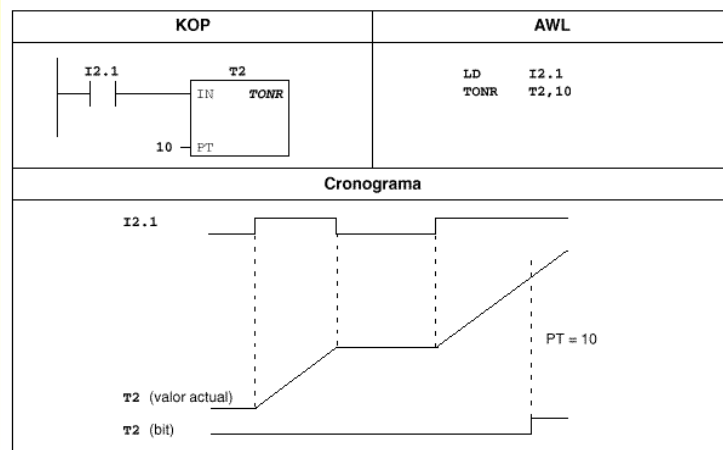
Operaciones con temporizadores

■ TON



Operaciones con temporizadores

■ TONR



Autómatas Programables
ISA-UMH © TDOC-2000

37

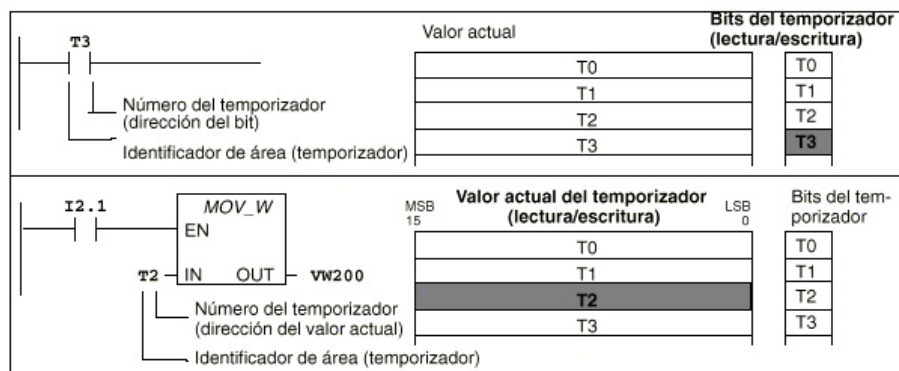
Operaciones con temporizadores

- Direccionamiento del área de temporizadores (T)
- Hay dos variables asociadas a los temporizadores:
 - Valor actual: En este número entero de 16 bits con signo se deposita el valor de tiempo contado por el temporizador.
 - Bit del temporizador (bit T): Este bit se activa (se pone a 1) cuando el valor actual del temporizador es mayor o igual al valor predeterminado. (Éste último se introduce como parte de la operación).
- A estas dos variables se accede
 - Formato: T [número del temporizador] T24
- Las operaciones con operandos en formato de bit acceden al bit del temporizador, en tanto que las operaciones con operandos en formato de palabra acceden al valor actual.

Autómatas Programables
ISA-UMH © TDOC-2000

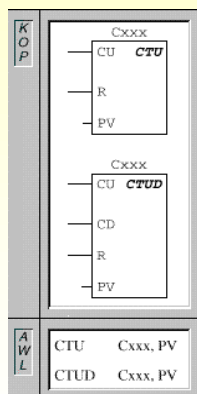
38

Operaciones con temporizadores



Operaciones con contadores

■ Contar adelante, Contar adelante/atrás



– La operación **Contar adelante**

- empieza a contar hasta el valor máximo cuando se produce un flanco positivo en la entrada de conteo adelante (CU).
- Si el valor actual (Cxxx) es mayor o igual al valor de preselección (PV), se activa el bit de conteo (Cxxx).
- El contador se inicializa al activarse la entrada de desactivación (R).

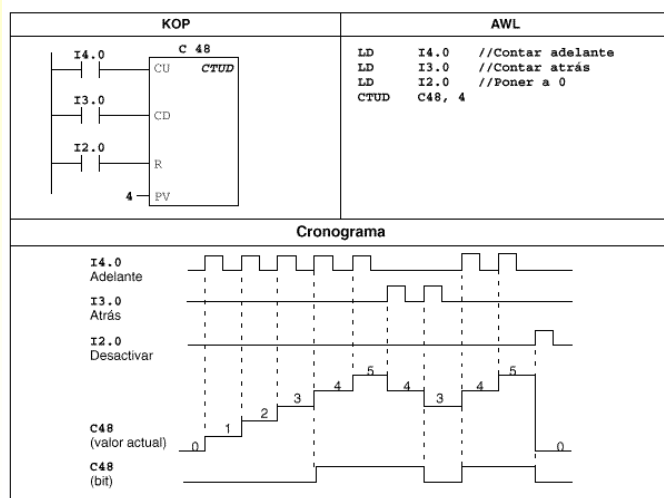
– La operación **Contar adelante/atrás**

- empieza a contar atrás cuando se produce un flanco positivo en la entrada de conteo atrás (CD).

– Operandos:

- Cxxx: 0 a 255
- PV: VW, T, C, IW, QW, MW, SMW, AC, AIW, constante, *VD, *AC, SW

Operaciones con contadores



Autómatas Programables
ISA-UMH © TDOC-2000

41

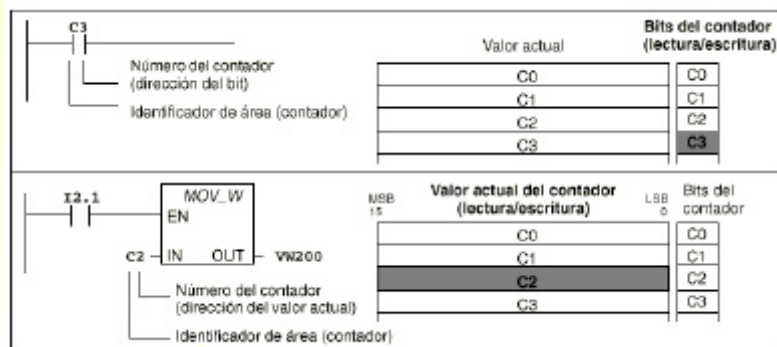
Operaciones con contadores

- Direccionamiento de los contadores (C)
 - Hay dos variables asociadas a los contadores:
 - Valor actual: En este número entero de 16 bits con signo se deposita el valor de conteo acumulado.
 - Bit del contador (bit C): Este bit se activa (se pone a 1) cuando el valor actual del contador es mayor o igual al valor predeterminado. (Éste último se introduce como parte de la operación).
 - A estas dos variables se accede utilizando la dirección del contador (C + número del contador).
 - Dependiendo de la operación utilizada, se accede al bit del contador o al valor actual.
 - Formato: C [número del contador] C20

Autómatas Programables
ISA-UMH © TDOC-2000

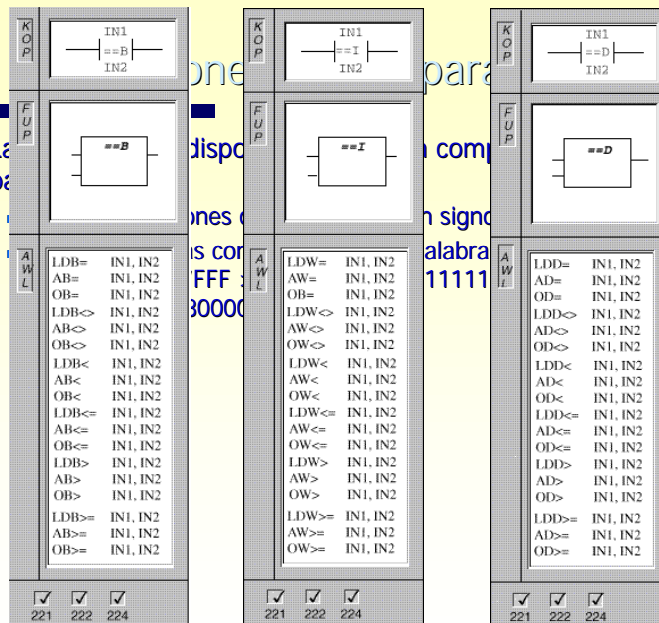
42

Operaciones con contadores



Autómatas Programables
 ISA-UMH © TDOC-2000

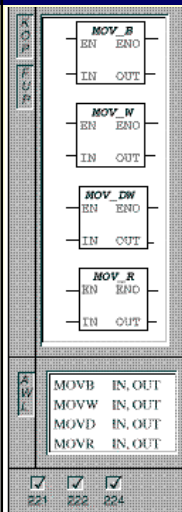
43



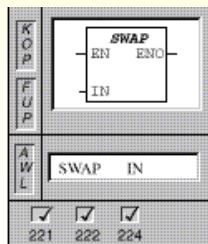
Autómatas Programables
 ISA-UMH © TDOC-2000

44

Operaciones de transferencia



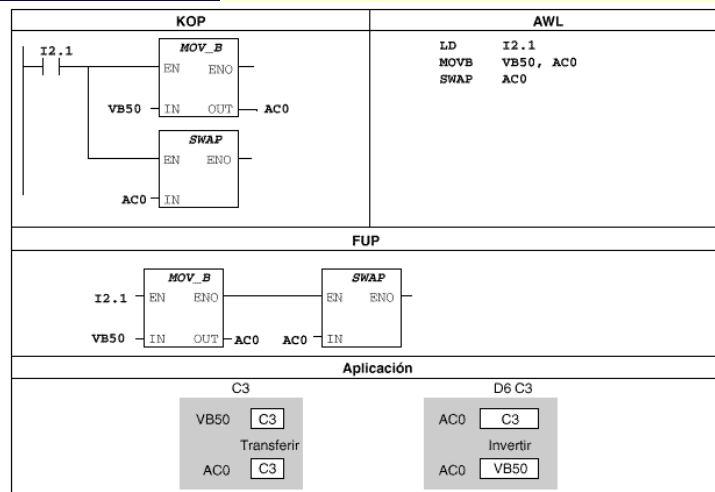
- Transferir byte, Transferir palabra, Transferir palabra doble y Transferir real
 - Las operaciones de transferencia se utilizan para transferir datos de una dirección a otra.
- La operación Invertir bytes de una palabra intercambia el byte más significativo y el byte menos significativo de una palabra (IN)



Autómatas Programables
ISA-UMH © TDOC-2000

45

Operaciones de transferencia



Autómatas Programables
ISA-UMH © TDOC-2000

46