UNIVERSIDADE DE VIGO

Departamento de Ingeniería de Sistemas y Automática

TEMPORIZADORES , CONTADORES Y ACUMULADORES

UNIVERSIDADE DE VIGO

Acumuladores

- Los acumuladores son registros auxiliares en la CPU que se utilizan en el intercambio de datos y para operaciones de comparación y matemáticas. El S7-300 tiene dos acumuladores de 32 bits cada uno y el S7-400 cuatro.

ACU	Bit
ACUx (x = 1 a 2)	Bits 0 a 31
ACUx-L	Bits 0 a 15
ACUx-H	Bits 16 a 31
ACUx-LL	Bits 0 a 7
ACUx-LH	Bits 8 a 15
ACUx-HL	Bits 16 a 23
ACUx-HH	Bits 24 a 31

UNIVERSIDADE DE VIGO

Acumuladores

- Las siguientes instrucciones están disponibles para intercambiar y desplazar el contenido de los acumuladores:
  - **TAK** intercambia el contenido de ACCU 1 con el contenido de ACCU 2
  - **PUSH** desplaza el contenido de ACCU 1 a ACCU 2
  - **POP** desplaza el contenido de ACCU 2 a ACCU 1

UNIVERSIDADE DE VIGO

Acumuladores : Operaciones de carga y transferencia

- No dependen del valor del RLO

L - Carga T - Transferencia
(Todos los tipos de datos con 8, 16, 32 bits)

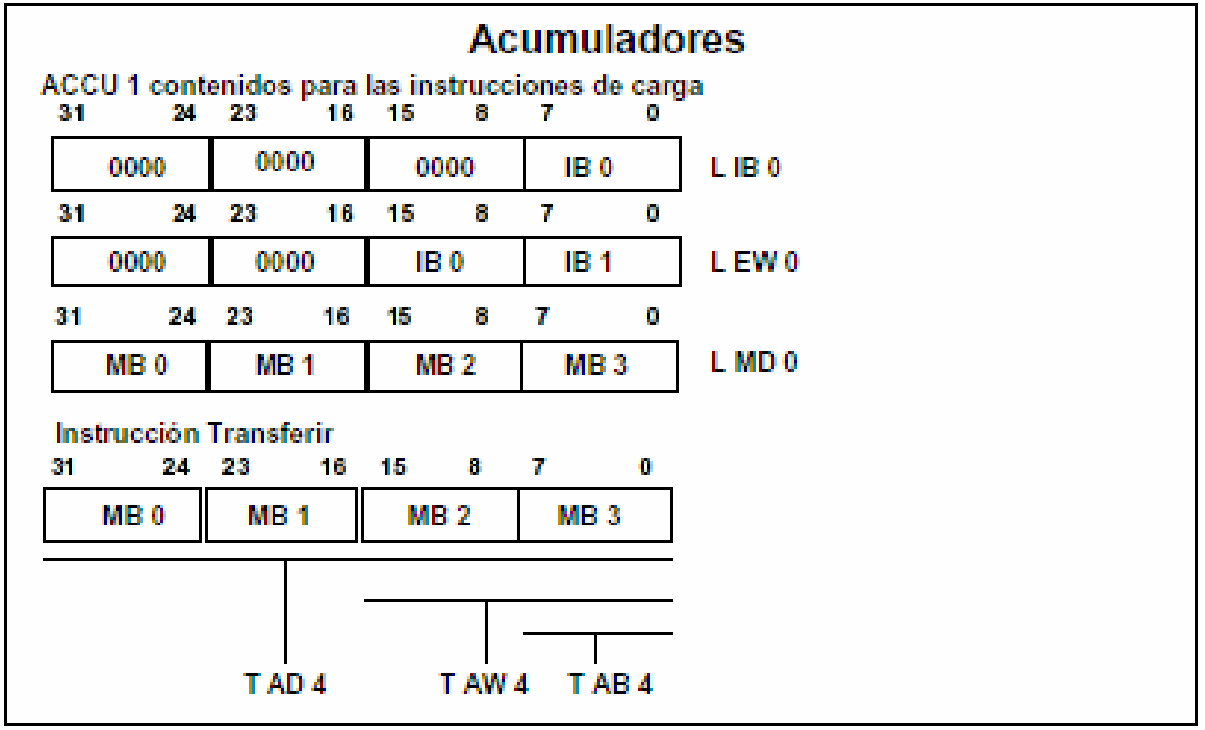
Ejemplos:

```
L +5              // Carga un entero de 16-bit
L L#s23123        // Carga un entero de 32-bit
L B#16#EF         // Carga un hexadecimal de 8-bit
L 2#0001_0110_1110_0011
                  // Carga un binario de 16-bit
L TOD#1:10:3.3
                  // Carga un tiempo de 32-bit

T MB0             // Transferir un valor al byte de
                     marcas 0
T AD256           // Transferir un valor a la doble
                     doble palabra de salida 256
```

Acumuladores

- **Operación de Carga**
  - La operación de carga siempre afecta al ACCU 1. Las posiciones ro utilizadas se ponen a 0. El valor actual del ACCU 1 pasa al ACCU 2 durante la carga.
- **Operación de Transferencia**
  - Durante una transferencia, el contenido de ACCU 1 se retiene y se usa para transferir la información a varias áreas de memoria. Si sólo se transfiere un byte se usan los ocho bits de la derecha.



UNIVERSIDADE DE VIGO

Temporizadores

- La señal de disparo se activa por flanco. Independiente de la duración.
- La señal generada se mantiene activa mientras dura la temporización.
- Pueden ser por hardware o por software (variables o alarmas)
- Admiten el redisparo de la temporización.
- Las variables tipo temporizadores admiten operaciones específicas.
  - Modos de funcionamiento (SI, SV, SE, SS, SA)
  - Borrado (R)
  - Consulta de estado (U, UN, O, ON, X, XN)
  - Cargar un valor dado (L, LC)
  - Habilitar un temporizador (FR)

Modos de temporización

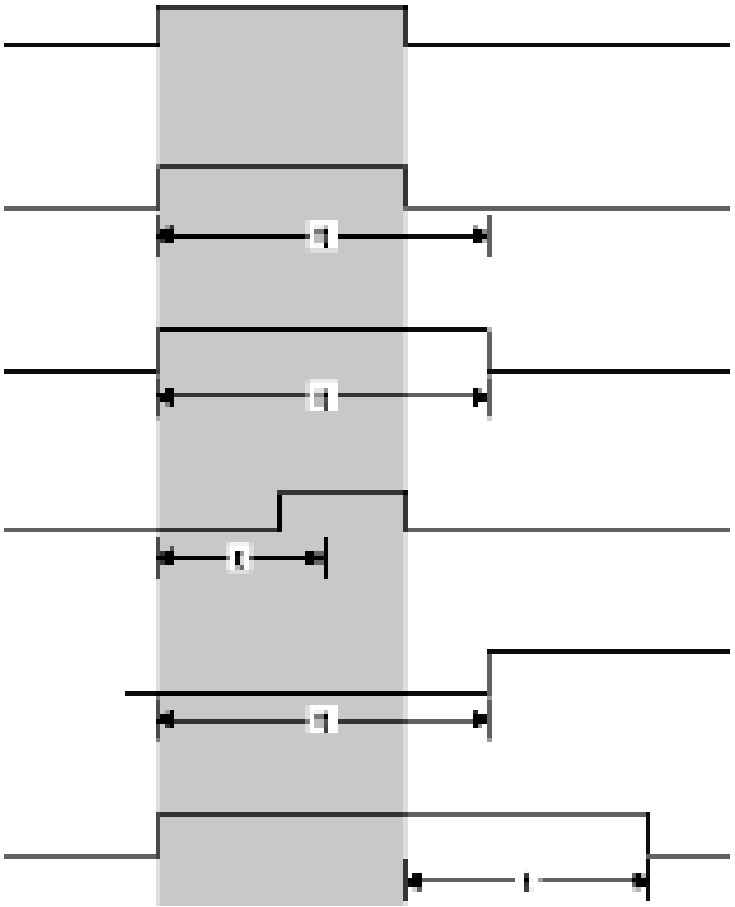
SI – Por impulso

SV – Por impulso prolongado

SE– Por retardo a la conexión

SS– Por retardo a la conexión memorizada

SA– Por retardo a la desconexión



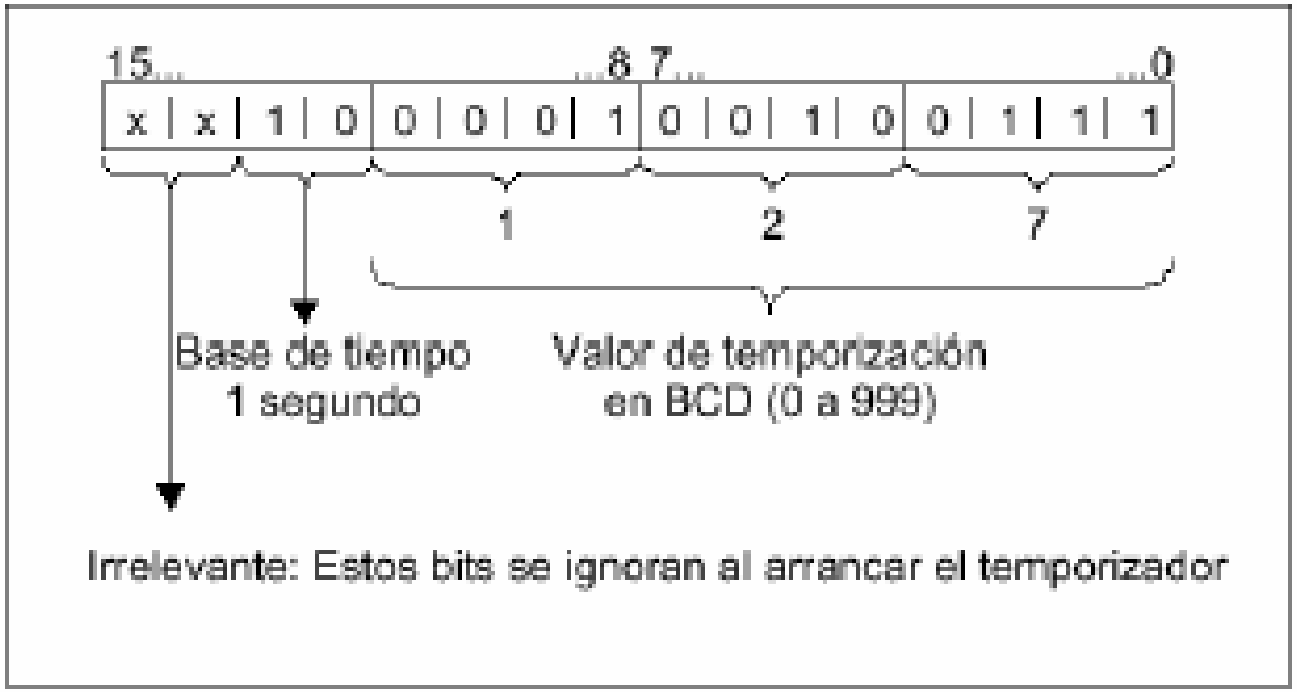
UNIVERSIDADE DE VIGO

Temporizadores

- ✓ Ocupan una zona específica de la memoria de la CPU
- ✓ Ocupan 16 bits cada uno (palabra de temporización)
- ✓ Máximo 256 en Step7.
- ✓ Componentes de la palabra de temporización
  - Cantidad de tiempo. 10 bits (0–9) Máximo 1024, real 999
  - Memoria de habilitación. Un bit (10) Frult memoriza el RLO del FR anterior
  - Memoria de disparo. Un bit (11) Sult memoriza el RLO del S anterior
  - Retícula de tiempo. Dos bits (12-13) 10 mseg, 100 mseg, 1 seg, 10 seg
  - Bit de control. Dos bits (14-15) Utilizados por el sistema operativo

Temporizadores

PALABRA DE TEMPORIZACIÓN



UNIVERSIDADE DE VIGO

Temporizadores

✓ Arranque o disparo

Solamente se dispara si existe un cambio de 0 a 1 en el Sult.(Copia del RLO)

CLR . Pone a 0 el RLO

SV T1 Guarda el RLO (0) en el Sult

U E 0.0 Primera consulta. Pone el valor de E 0.0 en el RLO

SV T1 Guarda el RLO en el Sult. Al detectar el cambio de Sult, arranca T1

✓ Valor de temporización (Carga temporización en el acumulador ACU1)

L W#16#rxyz

siendo r : retícula de tiempo

xyz : valor de temporización, tres cifras en formato BCD (000:999)

UNIVERSIDADE DE VIGO

Temporizadores

✓ Valor de temporización (Carga temporización en el acumulador ACU1)

L S5T#aH_bM_cS_dMS

siendo a : Horas b: Minutos c: Segundos d: Milisegundos

la retícula de tiempo de asigna automáticamente. Busca mayor precisión.

✓ Habilitación de un temporizador **FR T1**

Si el bit FRult=0 y el RLO=1 escribe un 0 en FRult.(Habilitación)

✓ Borrado o anulación de un temporizador : **R T1**

Si el RLO=1 la temporización se anula

✓ Consulta de estado **U T3**

Para el SV 1: Temporizador funcionando 0: No temporizando

UNIVERSIDADE DE VIGO

Temporizadores

✓ Carga / Lectura de un temporizador

Cargar un temporizador es copiar su contenido en el ACU1

L T1 (binario) o **LC T1** (BCD)

Temporizadores

- También es posible resetear el temporizador mediante una entrada, con lo cual el valor del temporizador pasa a 0 y el bit del mismo se deshabilita automáticamente.
  - U E 0.1 // SI ESTA LA ENTRADA
  - R T 0 // EL TEMPORIZADOR SE RESETEA

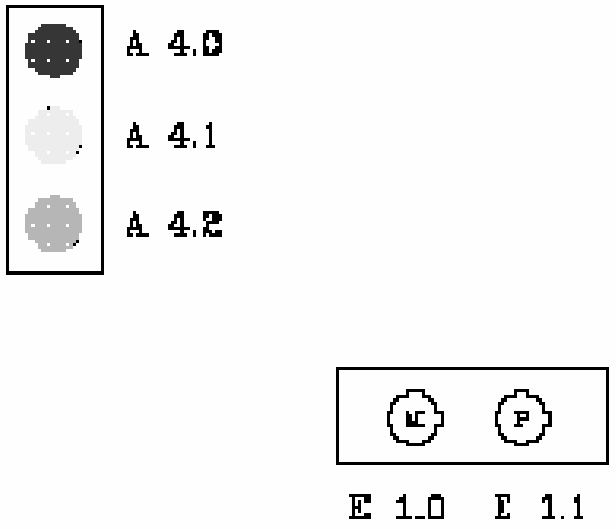
- Otra posibilidad es relanzar el contaje del temporizador, mediante la función FR de liberación de temporización. Cuando se active la entrada, el contador comienza de nuevo su proceso de contaje desde el último valor que se le había asignado como valor preseleccionado.
  - U E 0.3 // SI ESTA LA ENTRADA
  - FR T 0 // COMIENZA DE NUEVO EL CONTAJE

- Por último nos puede ser interesante conocer el estado actual del temporizador (cuanto tiempo le resta por contar). Para ello, únicamente debemos de cargar el valor de la palabra del temporizador. Esta carga se puede realizar de dos modos: normal en formato decimal (para comparaciones), o codificada en formato BCD (utilizada en displays).
  - L T 0 // CARGA EL VALOR ACTUAL DEL TEMPORIZADOR
  - T MW 0 // TRANSFIERELO EN DECIMAL

  - LC T 0 // CARGA CODIFICADO EL VALOR EL TEMPORIZADOR
  - T MW 2 // TRANSFIERELO EN FORMATO BCD

Ejemplo: Semáforo.

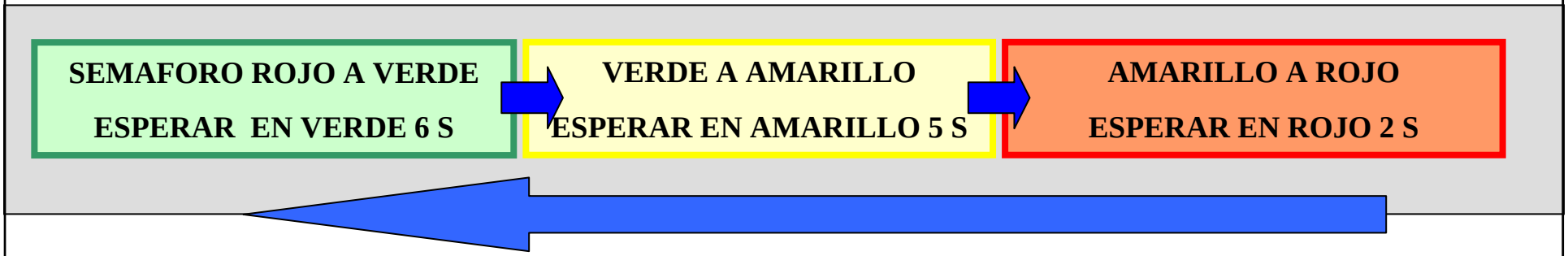
- Tenemos un semáforo con las tres luces verde, amarillo y rojo. Tenemos dos pulsadores de mando: un pulsador de marcha y un pulsador de paro.
- Con el pulsador de marcha quiero que comience el ciclo. El ciclo de funcionamiento es el siguiente:
- 1º/ Verde durante 5 seg.
- 2º/ Verde + Amarillo durante 2 seg.
- 3º/ Rojo durante 6 seg.
- El ciclo es repetitivo hasta que se pulse el pulsador



Temporizadores

Ejemplo: Semáforo.

PROGRAMACION ESTRUCTURADA



Ejemplo: Semáforo.

```
U E 1.0
R A 4.0   Apaga el ROJO
R A 4.1   Apaga el AMARILLO
S A 4.2   Enciende el VERDE
S M 0.0   Marca R-->V
```

```
CLR       0 → RLO
SV T1     RLO –>  Sult
U M 0.0   M 0.0 → RLO
L S5T#6s  Carga 6s  en ACU1
SV T1     Carga y Arranca T1
R M 0.0   Borra marca R → V
S M 0.1   Activa marca VERDE
```

SEMAFORO ROJO A VERDE

ESPERAR EN VERDE 6 S

Temporizadores

Ejemplo: Semáforo.

VERDE A AMARILLO

ESPERAR EN AMARILLO 5 S

```
U M 0.1
UN T1
R A 4.2   Apaga el VERDE
S A 4.1  Enciende el
AMARILLO
R M 0.1 Borra marca VERDE
S M 0.2   Marca V-->A
```

```
CLR        0  → RLO
SV T2      RLO –>  Sult
U M 0.2      M 0.2 → RLO
L S5T#5s    Carga 5s  en Acu 1
SV T2       Carga y Arranca T2
R M 0.2      Borra marca V → A
S M 0.3      Activa marca AMARILLO
```

Temporizadores

Ejemplo: Semáforo.

AMARILLO A ROJO

ESPERAR EN ROJO 2 S

```
U M 0 3   M 0.3 → RLO
UN T2  No T1 y RLO → RLO
R A 4.1   Apaga el AMARILLO
S A 4.0  Enciende el  ROJO
R M 0.3 Borra marca AMARILLO
S M 0.4   Marca A--> R
```

```
CLR        0  → RLO
SV T3      RLO –>  Sult
U M 0.4      M 0.2 → RLO
L S5T#2s    Carga 2s  en Acu 1
SV T3       Carga y Arranca T3
R M 0.4      Borra marca A → R
S M 0.5      Activa marca ROJO
```

Temporizadores

Ejemplo: Semáforo.

```
U E 1.0
R A 4.0   Apaga el ROJO
R A 4.1   Apaga el AMARILLO
S A 4.2   Enciende el VERDE
S M 0.0   Marca R-->V
```

```
CLR       0 → RLO
SV T1     RLO –>  Sult
U M 0.0   M 0.0 → RLO
L S5T#5s  Carga 5s  en ACU1
SV T1     Carga y Arranca T1
R M 0.0   Borra marca R → V
S M 0.1   Activa marca VERDE
```

```
U M 0.1
UN T1
R A 4.2   Apaga el VERDE
S A 4.1   Enciende el
AMARILLO
R M 0.1 Borra marca VERDE
S M 0.2   Marca V-->A
```

```
CLR       0 → RLO
SV T2     RLO –>  Sult
U M 0.2   M 0.2 → RLO
L S5T#6s  Carga 6s  en Acu 1
SV T2     Carga y Arranca T2
R M 0.2   Borra marca V → A
S M 0.3   Activa marca AMARILLO
```

```
U M 3   M 0.3 → RLO
UN T2   No T1 y RLO → RLO
R A 4.1   Apaga el AMARILLO
S A 4.0   Enciende el  ROJO
R M 0.3 Borra marca AMARILLO
S M 0.4   Marca A--> R
```

```
CLR       0 → RLO
SV T3     RLO –>  Sult
U M 0.4   M 0.2 → RLO
L S5T#2s  Carga 2s  en Acu 1
SV T3     Carga y Arranca T3
R M 0.4   Borra marca A → R
S M 0.5   Activa marca ROJO
```

UNIVERSIDADE DE VIGO

Temporizadores

¿ Como se programaría la transición del ROJO al VERDE ?

Temporizadores

Ejemplo: Semáforo

```
U M 0.5   M 0.5 → RLO
UN T3  No T3 y RLO → RLO
R M 0.5 Borra marca ROJO
R A 4.0   Apaga el ROJO
S A 4.2   Enciende el VERDE
R M 0.5 Borra marca ROJO
S M 0.0   Marca R --> V
```

```
CLR        0  → RLO
SV T1      RLO –>  Sult
U M 0.0    M 0.0 → RLO
L S5T#5s   Carga 5s  en ACU1
SV T1      Carga y Arranca T1
R M 0.0    Borra marca R → V
S M 0.1    Activa marca VERDE
```

UNIVERSIDADE DE VIGO

Contadores

- Utilizamos los contadores integrados en la CPU
- Contaje ascendente o descendente entre 0 y 999.
- Las operaciones específicas sobre contadores son.
  - Inicialización (S)
  - Incrementar una unidad (ZV)
  - Decrementar una unidad (ZR)
  - Borrado (R)
  - Consulta de estado (U, UN, O, ON, X, XN)
  - Cargar un valor dado (L, LC)
  - Habilitar un contador (FR)

UNIVERSIDADE DE VIGO

Contadores

- ✓ Ocupan una zona específica de la memoria de la CPU
- ✓ Ocupan 16 bits cada uno (palabra de contador)
- ✓ Máximo 256 en Step7.
- ✓ Componentes de la palabra de contador
  - Valor de contador. 10 bits (0–9) Máximo 1024, real 999
  - Memoria de habilitación. Un bit (10) FRult memoriza el RLO del FR anterior
  - Memoria de disparo. Un bit (11) Sult memoriza el RLO del S anterior
  - Memoria de ZR. Un bit (12). ZRult memoria el RLO del ZR anterior
  - Memoria de ZV. Un bit (13). ZRult memoria el RLO del ZV anterior
  - Salida del contador. Un bit (15) 0: Contador a cero 1:Resto de casos

UNIVERSIDADE DE VIGO

Contadores

✓ Carga inicial

L C#<valor> Carga el valor en el ACU1

S Z1 Transfiere el acumulador en el contador Z1

✓ Contaje ascendente

Solamente se dispara si existe un cambio de 0 a 1 en el ZVult.(Copia del RLO)

CLR . Pone a 0 el RLO

ZV Z1 Guarda el RLO (0) en el ZVult

U E 0.0 Primera consulta. Pone el valor de E 0.0 en el RLO

ZV Z1 Guarda el RLO en el ZVult. Al detectar el cambio de ZVult, incrementa una unidad Z1

UNIVERSIDADE DE VIGO

Contadores

✓ Contaje descendente

Solamente se dispara si existe un cambio de 0 a 1 en el ZRult.(Copia del RLO)

CLR . Pone a 0 el RLO

ZR Z1 Guarda el RLO (0) en el ZRult

U E 0.0 Primera consulta. Pone el valor de E 0.0 en el RLO

ZR Z1 Guarda el RLO en el ZRult. Al detectar el cambio de ZRult, decrementa una unidad Z1

UNIVERSIDADE DE VIGO

Contadores

✓Habilitación de un contador **FR Z1**

Si el bit FRult=0 y el RLO=1 escribe un 0 en FRult.(Habilitación)

✓ Borrado de un contador : **R Z1**

Si el RLO=1 el contador se pone a cero

✓ Consulta de estado **U Z3**

0: Contador a cero y 1: Cualquier otro valor del contador

✓ Carga / Lectura de un contador

Cargar un contador es copiar su contenido en el ACU1

L Z1 (binario) o **LC Z1** (BCD)

UNIVERSIDADE DE VIGO

Contadores

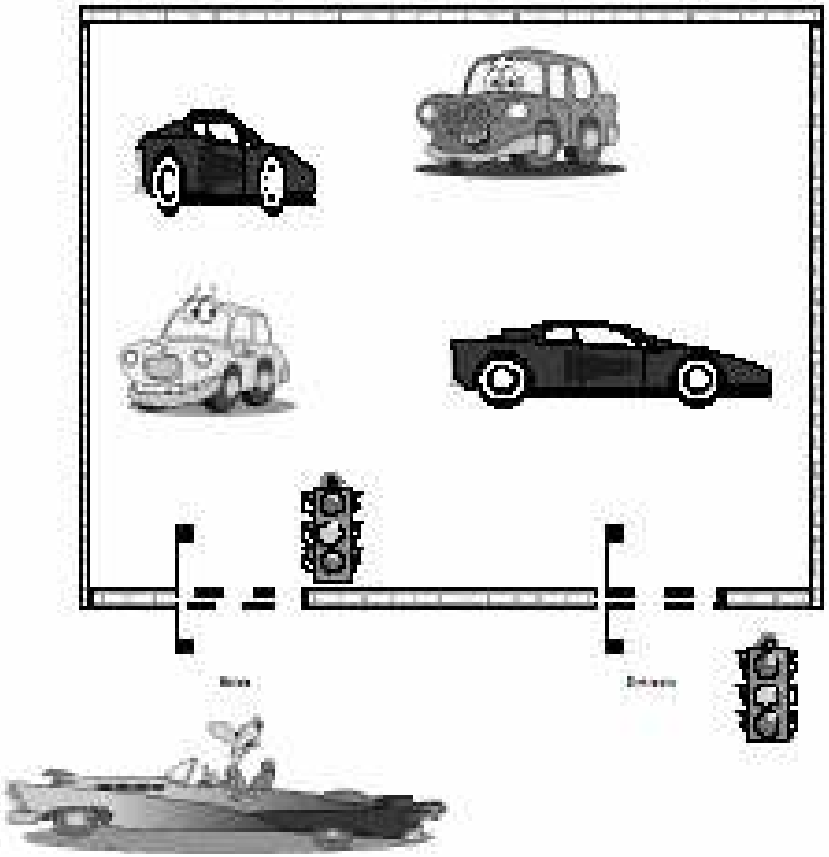
- Para meter los valores en los acumuladores, tenemos la instrucción de carga. (L).
  - Cuando cargamos un valor, siempre se carga en el acumulador 1. Cuando volvemos a cargar otro valor, también se guarda en acumulador 1. Lo que tenía en el acumulador 1 pasa al acumulador 2, y lo que tenía en el acumulador 2 lo pierde.
- En nuestro caso, cargaremos el valor de Z1 y a continuación cargaremos el valor con el que queremos comparar.
- Una vez tengamos los valores en el acumulador, tendremos que compararlos. Para ello tenemos las siguientes instrucciones:

>	>	>=	<=	==	<>
Mayor	Menor	Mayor o igual	Menor o igual	Igua	Dist.

- A continuación del símbolo de comparación pondremos una I si lo que estamos comparando son dos números enteros. Pondremos una R si lo que estamos comparando son números reales.

Contadores

Ejemplo: Parking



Automatizar un garaje de 5 plazas de tal forma que si éste se encuentra lleno se encienda una luz indicándolo y no suba la barrera. En caso contrario deberá estar encendida otra luz indicando "LIBRE".

El garaje consta de 5 plazas.

Disponemos de una célula fotoeléctrica y una barrera en la entrada y lo mismo en la salida.

Asignación de variables	
E0.0	Célula fotoeléctrica de entrada
E0.1	Célula fotoeléctrica de salida
A4.0	Barrera de entrada
A4.1	Barrera de salida
A4.2	Luz de señalización de "LIBRE"
A4.3	Luz de señalización de "LLENO"

UNIVERSIDADE DE VIGO

Contadores

Inicio del sistema

```
CLR
S Z1
UN M 0.0
L C#5
S Z1        Asigna 5 al contador Z1



UN M 0.0

S M 0.0    Marca iniciado sistema
```

Contadores

Entrada de vehículos

```
CLR      0 → RLO
ZR Z1    RLO -> ZRult

U  Z1    Garaje vacío
U E 0.0  Coche en entrada
ZR Z1    Decrementa contado
S  A 4.0  Abre barrera entrada
S M 0.1  Coche pasando entrada

U Z1
S A 4.2   Enciende Luz LIBRE
R A 4.3   Apaga luz LLENO
```

UNIVERSIDADE DE VIGO

Contadores

Salida de vehículos

```
CLR      0 → RLO
ZV Z1    RLO -> ZVult

U E 0.1   Coche en salida
ZV Z1     Incrementa contador
S A 4.1   Abre barrera salida
S M 0.2   Coche pasando salida

UN Z1
S A 4.3   Enciende luz LLENO
R A 4.2   Apaga luz LIBRE
```

Contadores

Bajar la barrera

```
UN M 0.1     Coche pasando entrada
UN E 0.0       Ya no hay coche
R A 4.0      Baja barrera entrada
R M 0.1        Borrado marca


UN M 0.2     Coche pasando salida
UN E 0.1       Ya no hay coche
R A 4.1      Baja barrera salida
R M 0.2        Borrado marca
```

Contadores

Solución.

```
CLR
S Z1
UN M 0.0
L C#5
S Z1        Asigna 5 al contador Z1
UN M 0.0
S M 0.0     Marca iniciado sistema
```

```
UN M 0.1    Coche pasando entrada
UN E 0.0    Ya no hay coche
R A 4.1     Baja barrera entrada
R M 0.1

UN M 0.2    Coche pasando salida
UN E 0.1    Ya no hay coche
R A 4.0     Baja barrera salida
R M 0.2
```

```
CLR     0 → RLO
ZR Z1   RLO -> ZRult

U  Z1   Garaje vacío
U E 0.0 Coche en entrada
ZR Z1   Decrementa contado
S  A 4.1 Abre barrera entrada
S M 0.1 Coche pasando entrada

U Z1
S A 4.2  Enciende Luz Verde
R A 4.3  Apaga luz Roja
```

```
CLR     0 → RLO
ZV Z1   RLO -> ZVult

U E 0.1 Coche en salida
ZV Z1   Incrementa contador
S A 4.0  Abre barrera salida
S M 0.2 Coche pasando salida

UN Z1
S A 4.3 Enciende luz Roja
R A 4.2 Apaga luz Verde
```

UNIVERSIDADE DE VIGO

Temporizadores

¿ Si el parking se amplia a 50 plazas, que tendríamos que modificar en el programa STEP 7 ?

UNIVERSIDADE DE VIGO

Contadores

Inicio del sistema

```
UN M 0.0
L C#50
S Z1        Asigna 50 al contador
Z1



UN M 0.0

S M 0.0    Marca iniciado sistema
```

Contadores