

Introducción a las PLC

Grafcets

1.GRAFCET Introducción

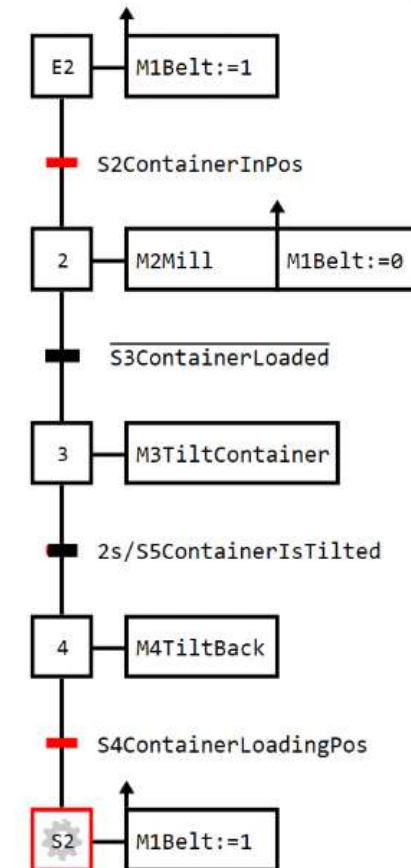
En 1975 la Asociación Francesa para Cibernética Económica y Técnica (AFCET) crea una comisión para comparar e investigar los diferentes métodos y modelos de programación de controladores lógicos secuenciales que se estaban utilizando hasta entonces.

En el año 1977, a partir de las conclusiones de la comisión, fue creado el lenguaje GRAFCET y posteriormente en 1988 incluido en la norma IEC-848 (fundamentos) y en 2002 en la IEC-60848 (estructuración)

Los lenguajes secuenciales (SFC, Sequential Function Chart) están basados en GRAFCET y forman parte de la norma IEC-61131-3 (por ejemplo, lenguaje LADDER –LD-)

Un **GRAFCET** (Gráfico Funcional de Control de Etapas y Transiciones) es una herramienta gráfica cuya finalidad es describir procesos mediante gráficos funcionales que puedan ser interpretados por personal no experto en automatización. Es un gráfico interpretable independiente a la tecnología empleada.

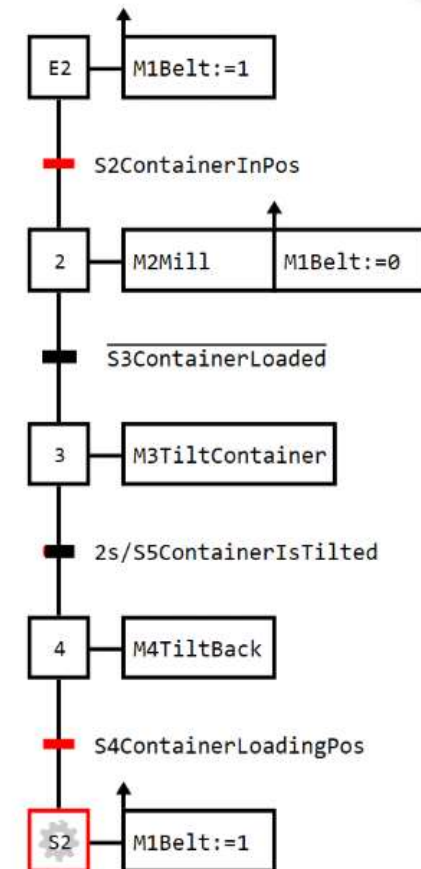
Actualmente el uso del GRAFCET se ha extendido mayoritariamente para la estructuración de programas de PLC's. Esta representación puede servir como herramienta que describe el funcionamiento de los sistemas de control.



2.GRAFCET Diseño

Los conceptos básicos de una GRAFCET son:

- ETAPAS: Acciones que realiza el controlador lógico
- TRANSICIONES: condiciones que se deben cumplir para realizar un cambio de etapa, están relacionadas con las receptividades del controlador
- CADENA SECUENCIAL: conjunto de etapas y transiciones adecuadamente enlazadas mediante determinadas reglas de evolución.

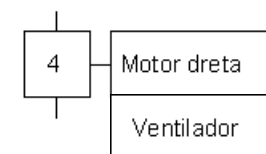
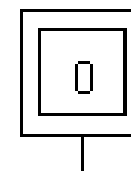
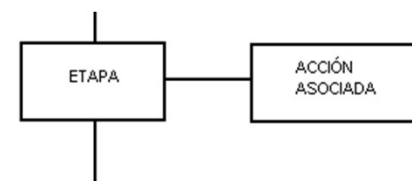


2.GRAFCET Diseño

- **Etapas:**

La etapa refleja una situación en la cual el comportamiento de todo o una parte de controlador lógico secuencial permanece invariable (las condiciones no varían). Las etapas están asociadas a acciones que equivalen a la activación o desactivación de determinadas variables lógicas. Las acciones pueden estar condicionadas por otras variables lógicas o temporales, o depender de la situación de otras etapas.

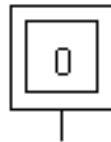
- ✓ El controlador siempre está en una etapa. Y puede permanecer en esta etapa hasta que las condiciones no cambien.
- ✓ La etapa inicial se representa por un rectángulo doble
- ✓ El resto de las etapas se representan mediante un **rectángulo**
- ✓ Cada etapa tendrá asociado un número **único** cuya asignación será **secuencial**
- ✓ Cada etapa tendrá instrucciones o acciones asociadas a esa etapa. Se representan en otro rectángulo al lado derecho de la etapa
- ✓ Puede haber etapas que no tengan órdenes asociadas



2.GRAFCET Diseño

El GRAFCETS debe mostrar cada una de las etapas y como va evolucionando el sistema a lo largo del tiempo.

- ✓ **En el sistema sólo puede estar en una etapa en una misma rama.** A esta etapa se le llama etapa activa.
- ✓ **El sistema tiene que partir siempre de condiciones iniciales** (estado inicial, **etapa 0**). La máquina se encuentra en reposo y, tras un orden, ir evolucionando por etapas a lo largo del tiempo, **secuencialmente**. Se representa mediante un doble cuadrado y no tiene orden asociada.



- ✓ **Para pasar de una etapa a la siguiente se tienen que cumplir una serie de condiciones**

Todo ello se expresará de forma gráfica en el GRAFCET.

2.GRAFCET Diseño



- **Transiciones**

Las transiciones entre dos etapas se representan mediante una línea vertical que las une.

Para que se realice la transición debe cumplirse una **condición lógica**:

- ✓ Se simboliza mediante una pequeña línea perpendicular a la unión entre dos etapas y a la derecha se indica una descripción de la condición.
- ✓ Suelen ser combinaciones lógicas de señales de detectores, *reeds*, resultados de temporizadores, contadores, etc.

- **Evolución**

La evolución es una secuencia de situaciones y está ligada a una serie de condiciones en las que se producen las transiciones entre etapas.

Si en la etapa actual (activa), **se cumple la condición lógica** de transición:

- ✓ **Se desactiva la etapa actual**
- ✓ **Se activa la siguiente etapa**

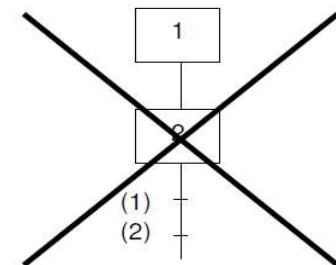
3.GRAFCET Normas

Para crear un GRAFCET deberán definirse, en primer lugar, **los sensores y actuadores** con la correspondiente lista de entradas/salidas. El GRAFCET deberá incluir:

- a. **La definición de un diagrama de estados** del sistema junto con las **condiciones** que deben darse para que el sistema evolucione de un estado a otro.
- b. **La definición de un mando de máquina** que sirva de interficie con el operario y que le permita interactuar con ésta.

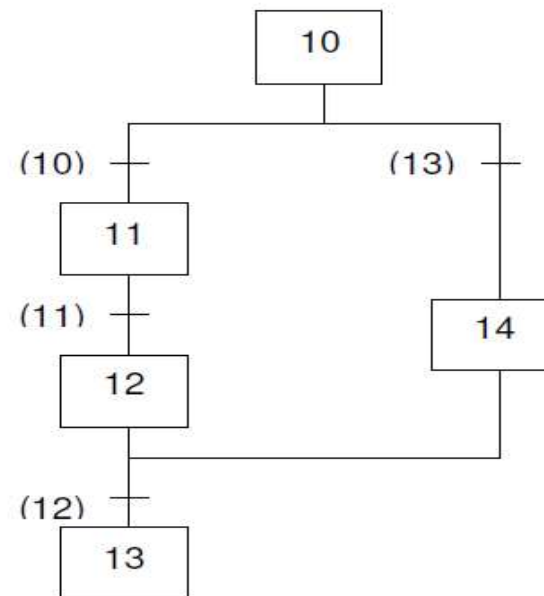
En la elaboración de un GRAFCET deben seguirse una serie de reglas:

- Siempre debe existir **una única etapa** activa en cada GRAFCET.
- Se activará una etapa cuando:
 - ✓ La etapa anterior este activa
 - ✓ **Sea cierta la transición**
- La última etapa del GRAFCET debe conectarse a la primera o a alguna etapa: el **GRAFCET queda cerrado**
- **No pueden existir dos etapas seguidas sin condición de transición**
- **No pueden existir dos transiciones seguidas**



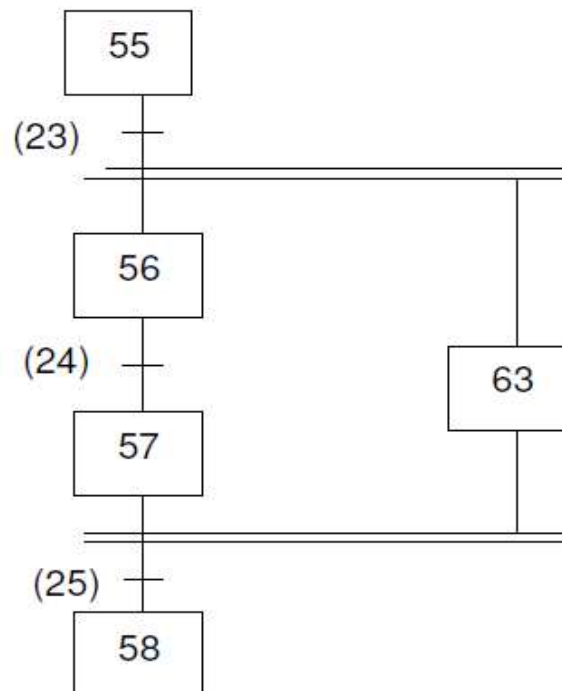
3.GRAFCET Normas

- Es posible bifurcar una secuencia en transiciones alternativas, donde:
 - ✓ debe haber una transición para cada rama
 - ✓ se cumplirá **una de las condiciones**
 - ✓ se ejecutará la rama de la condición que se cumpla (solo una)



3.GRAFCET Normas

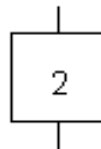
- Es posible indicar acciones paralelas.
 - ✓ El inicio de etapas paralelas se indica con una línea horizontal doble después de la transición
 - ✓ Se ejecutan “simultáneamente” en ramas diferentes
 - ✓ El final de las etapas paralelas se indica con una línea horizontal doble antes de la transición
 - ✓ Caso excepcional de coexistencia de más de una etapa activa en un mismo GRAFCET.



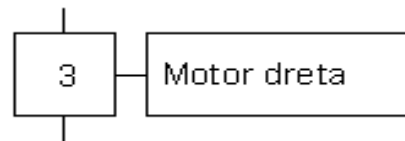
3.GRAFCET Normas

- En las acciones asociadas a una etapa deben tenerse en cuenta las siguientes prescripciones:

- ✓ Una etapa suele tener una acción asociada. No obstante pueden haber etapas que no tenga acción asociada. Las condiciones de transición si que deben de estar presentes.



- ✓ Una acción asociada indica que la acción se ejecuta al estar activa la etapa .



- ✓ En una etapa puede haber múltiples acciones asociadas



3.GRAFCET Normas

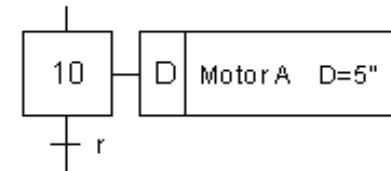
- **Tipos de acciones asociadas a etapas**

La norma de representación de GRAFCET IEC-848 propone una serie de representaciones gráficas sobre las acciones asociadas. Reflejan condicionantes adicionales a la ejecución de la acción cuando la etapa está activada.

Estas situaciones son las siguientes:

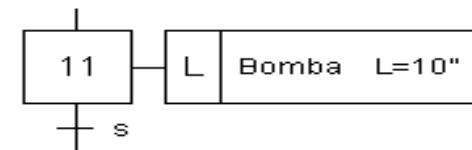
- ✓ **Acción retardada en el tiempo:** la acción asociada a la etapa empieza a ejecutarse al cabo de un tiempo.

Por ejemplo, el motor A se pondrá en marcha 5 segundos después de activarse la etapa 10; si la transición r se activa antes de ese tiempo el motor no llegara a ponerse en marcha.



- ✓ **Acción limitada en el tiempo:** la acción asociada se ejecutará durante un tiempo limitado.

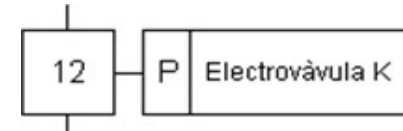
Por ejemplo, la bomba se pondrá en funcionamiento durante 10" después de haberse activado la etapa 11. Pasado este tiempo, aunque no se active la transición s, la bomba dejará de funcionar.



3.GRAFCET Normas

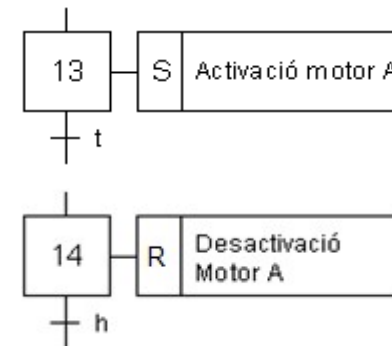
- ✓ **Acción de Pulso:** la acción asociada a la etapa se ejecutará mediante un pulso (una fracción de tiempo muy corta).

Por ejemplo, en la siguiente acción se activará la electroválvula K con un pulso como señal.



- ✓ **Acción Memorizada:** Se utiliza para representar la actuación memorizada de una señal con un SET o la desactivación con un RESET.

Por ejemplo, cuando se activa la etapa 13, el motor A se pondrá en marcha de forma memorizada (SET) y al salir de la etapa continuará funcionando. Al activarse la etapa 14, el motor A se detendrá debido al *reset*.



3.GRAFCET Ejemplo

Suponemos, como ejemplo, un manipulador de tres cilindros neumáticos :

1- Sensores y actuadores

Actuadores (3): 2 cilindros doble efecto (horizontal, vertical) y 1 cilindro efecto simple (pinza)

Sensores: Fococélula detección de pieza, *reeds* para cada posición de cilindro horizontal y vertical, pinza abierta.

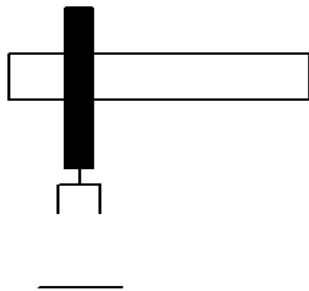
ENTRADAS	DESCRIPCION	TIPO DE DETECCION
1.00	Cilindro Horizontal Atrás	REED
1.01	Cilindro Horizontal Adelante	REED
1.02	Cilindro Vertical Alto	REED
1.03	Cilindro Vertical Bajo	REED
1.04	Pinza Abierta	REED
1.05	Detección Pieza en Línea	Fotocélula Barrera
SALIDAS	DESCRIPCION	TIPO SALIDA
10.00	EV Cilindro Horizontal Adelante	EV BIESTABLE
10.01	EV Cilindro Horizontal Atrás	
10.02	EV Cilindro Vertical Alto	EV BIESTABLE
10.03	EV Cilindro Vertical Abajo	
10.04	EV Cerrar Pinza	EV MONOESTABLE



1.00 - Cilindro Horizontal Atrás
1.02 - Cilindro Vertical alto
1.04 - Deteccion Pinza Abierto.

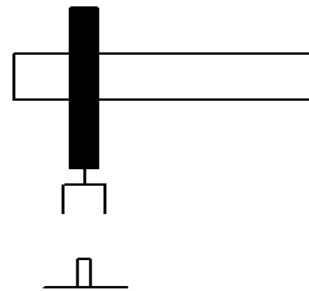
3.GRAFCET Ejemplo

2. Definición de etapas



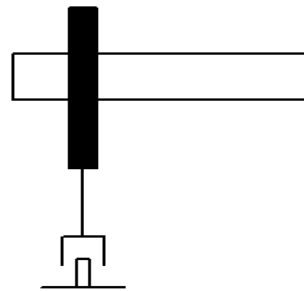
Condiciones iniciales:

- Cilindro vertical posición arriba
- Cilindro horizontal posición atrás
- Pinza abierta



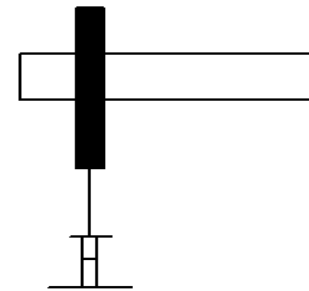
Etapa 1:

- Espera pieza



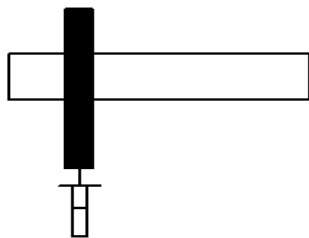
Etapa 2:

- Baja manipulador:
Cilindro vertical posición abajo



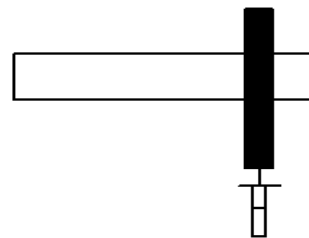
Etapa 3:

- Cierra pinza



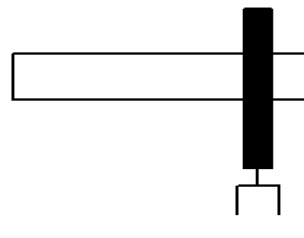
Etapa 4:

- Sube manipulador:
Cilindro vertical posición arriba



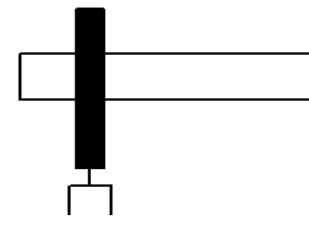
Etapa 5:

- Adelante manipulador:
Cilindro horizontal posición adelante



Etapa 6:

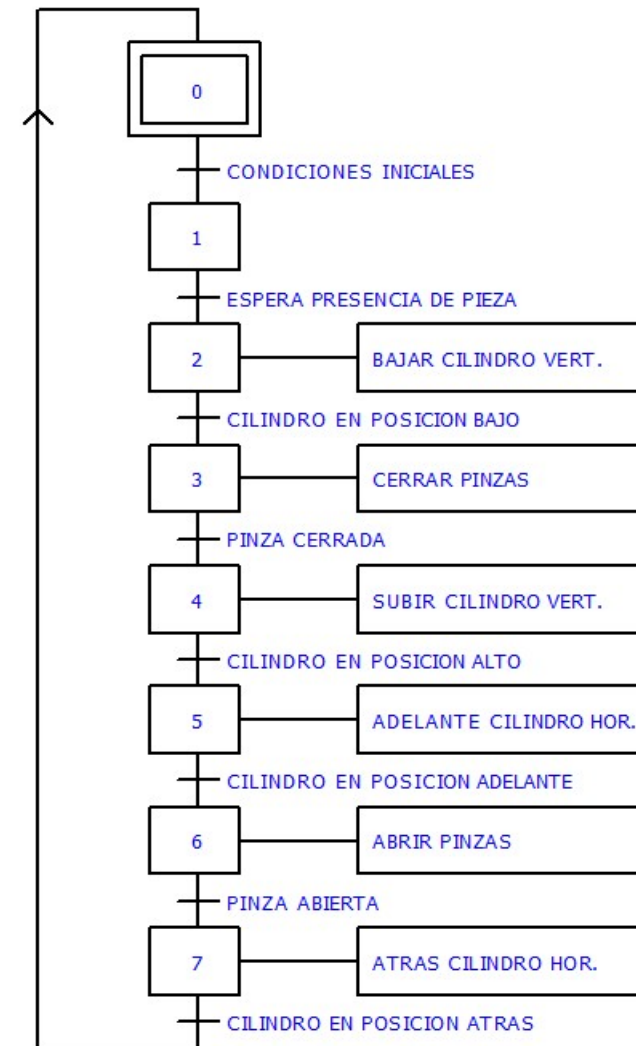
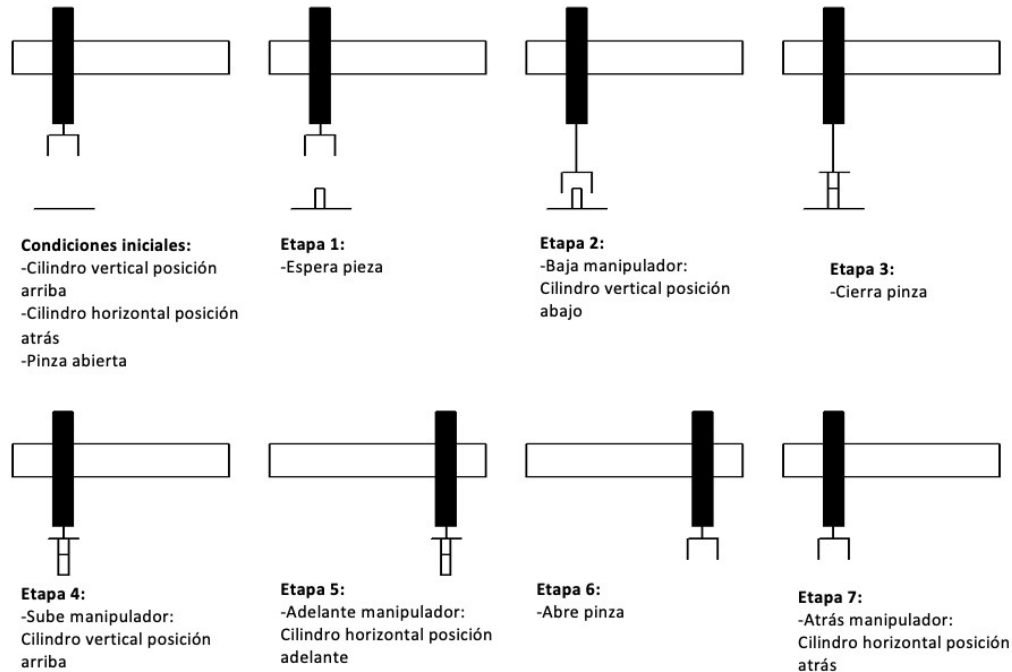
- Abre pinza



Etapa 7:

- Atrás manipulador:
Cilindro horizontal posición atrás

3.GRAFCET Ejemplo

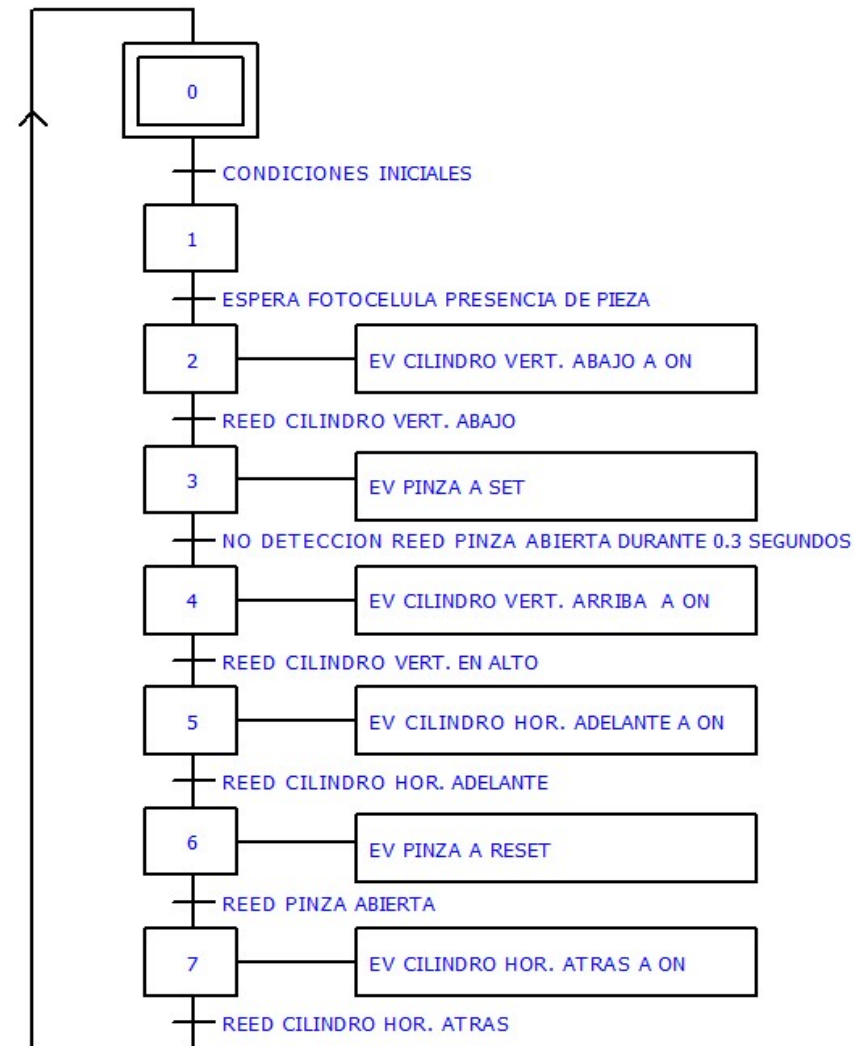


3.GRAFCET Ejemplo

ENTRADAS	DESCRIPCION	TIPO DE DETECCION
1.00	Cilindro Horizontal Atrás	REED
1.01	Cilindro Horizontal Adelante	REED
1.02	Cilindro Vertical Alto	REED
1.03	Cilindro Vertical Bajo	REED
1.04	Pinza Abierta	REED
1.05	Detección Pieza en Línea	Fotocélula Barrera
SALIDAS	DESCRIPCION	TIPO SALIDA
10.00	EV Cilindro Horizontal Adelante	EV BIESTABLE
10.01	EV Cilindro Horizontal Atrás	
10.02	EV Cilindro Vertical Alto	EV BIESTABLE
10.03	EV Cilindro Vertical Abajo	
10.04	EV Cerrar Pinza	EV MONOESTABLE



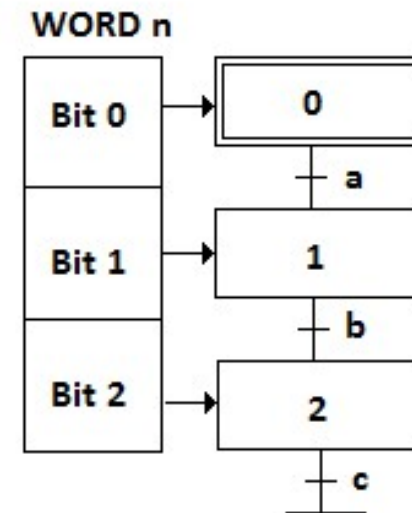
1.00 - Cilindro Horizontal Atrás
1.02 - Cilindro Vertical alto
1.04 - Deteccion Pinza Abierto.



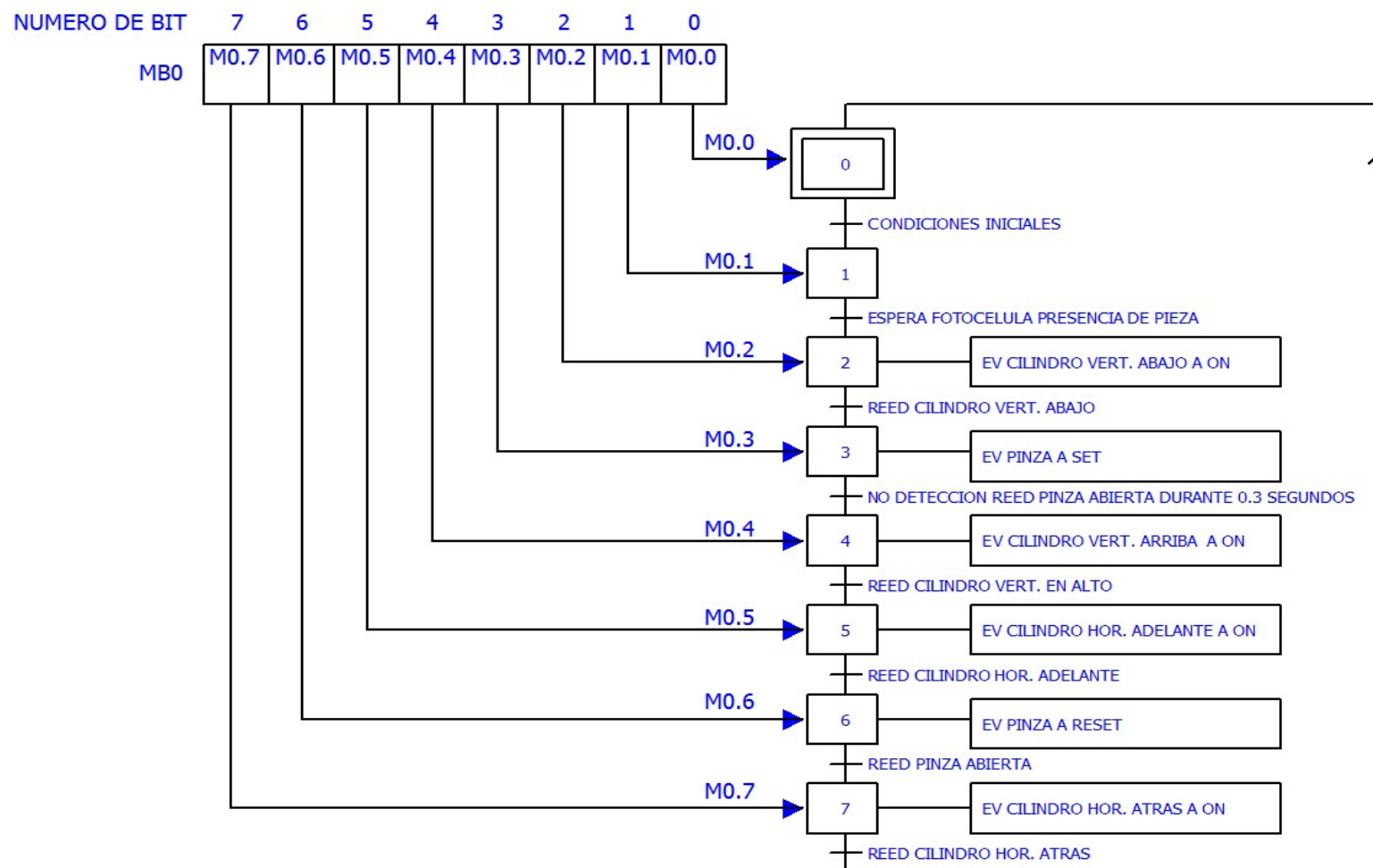
4.GRAFCET Programación

A partir del GRAFCET ya finalizado se debe diseñar un esquema de contactos:

- **A cada etapa se asigna un bit.** Si el bit está activo, la etapa está activa.
- Los bits de cada etapa de un GRAFCET deben formar parte de un **mismo word**, de tal manera que para poner a 0 el GRAFCET sea suficiente escribir un cero en el *word*.
- **Los bits deben asignarse a la etapa en orden secuencial**, de tal manera que el bit 0 corresponde a la etapa inicial, el bit 1 a la etapa 1, etc.



4.GRAFCET Programación



4.GRAFCET Programación



- La programación se realiza en dos bloques o dos partes diferenciadas en el programa *Ladder*:

1. Bloque de transiciones:

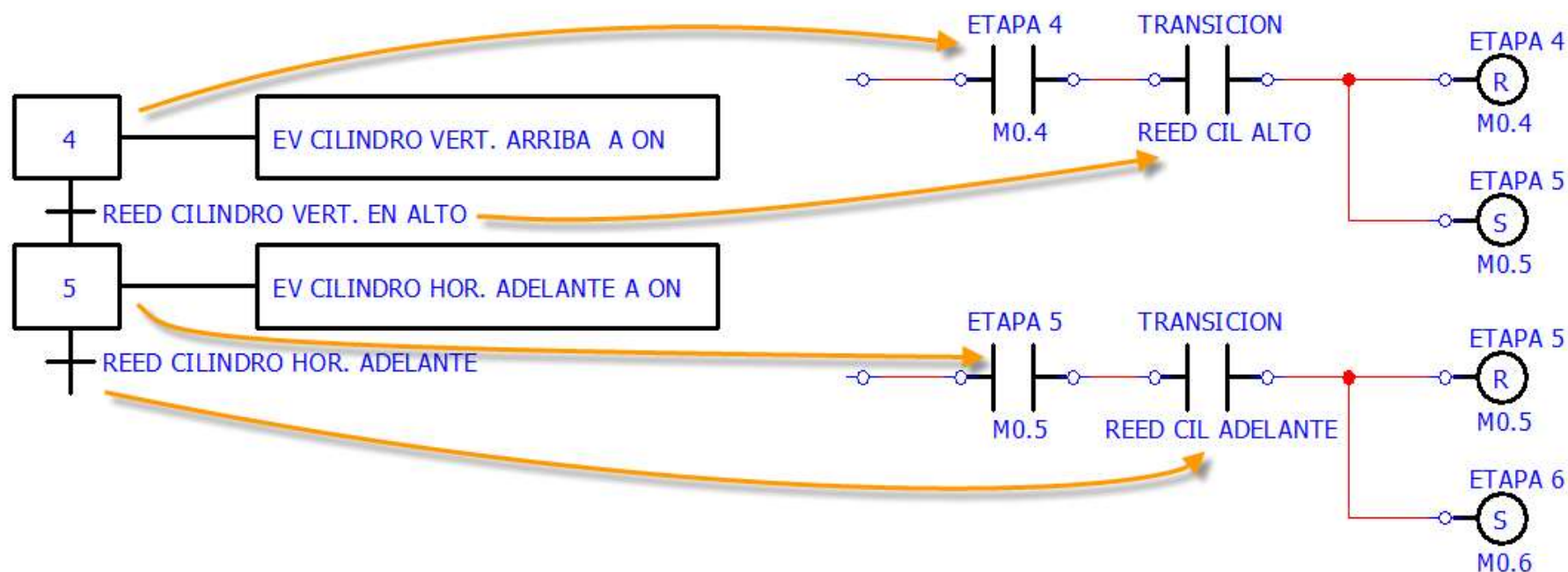
Se activan y desactivan las etapas de dos en dos, como resultado de la lógica de transición entre ellas. Si se cumple:

- ✓ El bit de la etapa activa a estado *true*
- ✓ Se cumplen las condiciones de transición

Entonces se activa la siguiente etapa:

- ✓ El bit de la etapa activa pasa a estado *false* (*reset*)
- ✓ El bit de la etapa siguiente pasa a estado *true* (*set*)

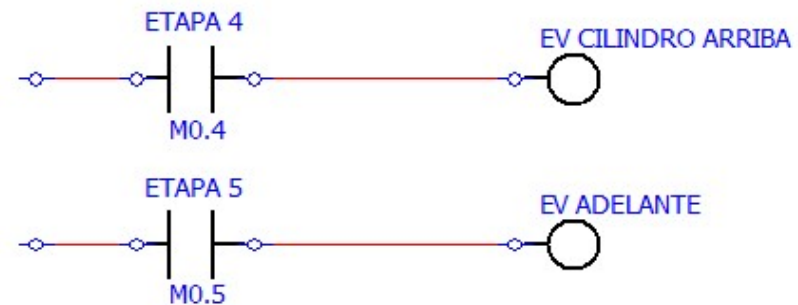
4.GRAFCET Programación



4.GRAFCET Programación

2. Bloque de ejecución de instrucciones asociadas

- ✓ Solo existe un bit de etapa en estado *true*, (bit de etapa activa)
- ✓ El bit de etapa activa permite ejecutar las instrucciones asociadas
- ✓ Las instrucciones asociadas pueden ser, por ejemplo, escritura de bits mediante bobinas o bobinas enclavadas (SET, RESET)

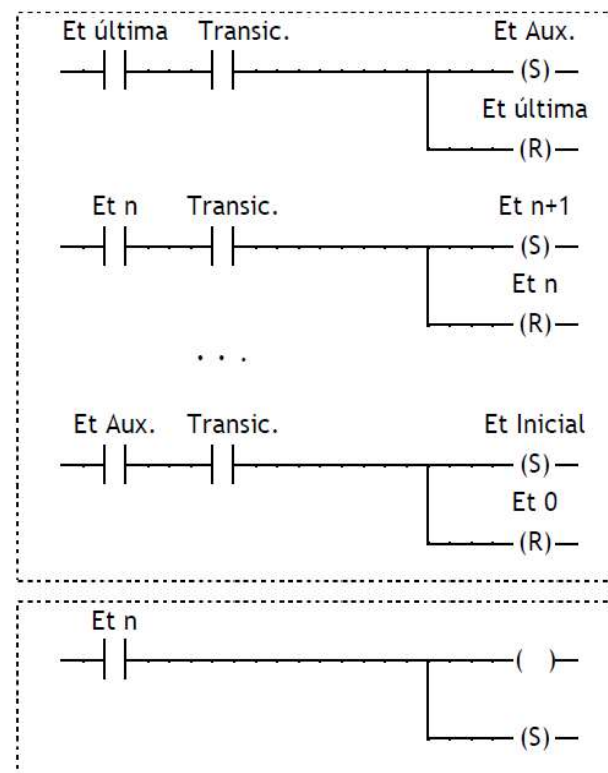


4.GRAFCET Programación

Es conveniente programa de forma ordenada. Primero se programarán las transiciones de etapa para posteriormente programar las acciones asociadas a cada etapa, en el mismo o en diferentes bloques.



TRANSICIONES



POSTERIOR

4.GRAFCET Programación

Cuando se realiza la programación de una máquina compleja que se divide en numerosas unidades funcionales, conviene no complicar en exceso el proceso de programación:

- **Dividir la máquina en unidades funcionales sencillas.**

Cada una de las unidades realizará un proceso secuencial predefinido y algunas de estas unidades pueden interactuar entre ellas. Por ejemplo, una estación formada por un manipulador y un alimentador tendría dos unidades funcionales, que interactúan entre ellas: el manipulador y el alimentador.

- Obtener el listado de entradas salidas de cada una de las unidades funcionales así como sus condiciones iniciales.
- Desarrollar un GRAFCET para cada una de estas unidades. El GRAFCET debe ser lo más lineal posible evitando en lo posible que se generen ramificaciones.

