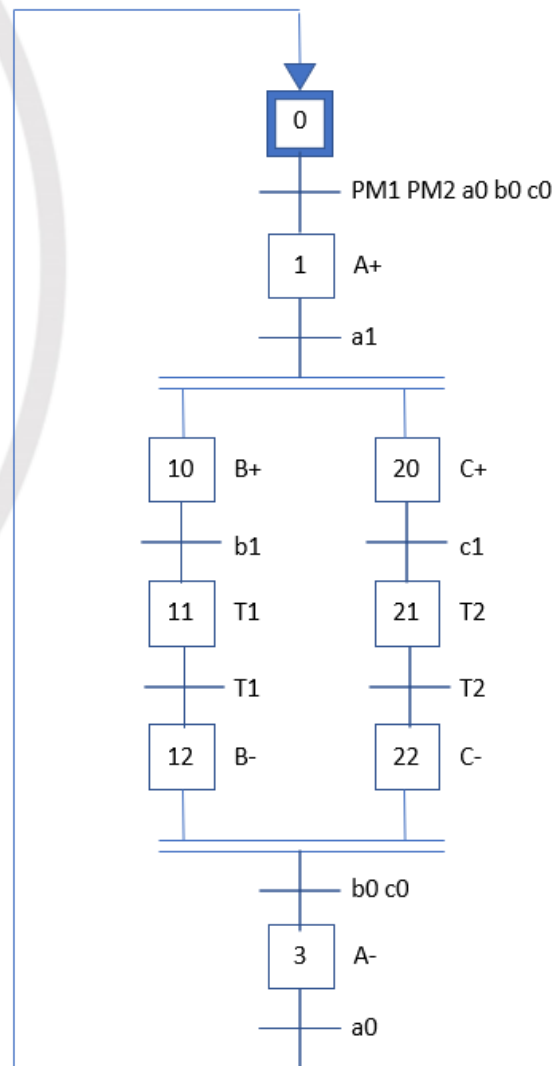


## MÉTODO DE PROGRAMACIÓN ESTRUCTURADA – DIVERGENCIAS CONVERGENCIAS

Hasta ahora hemos resuelto el método de programación estructurada en Grafcet puramente **lineales**. Veamos cómo aplicar el método cuando el Grafcet se ramifica en convergencias y divergencias del tipo and y or.

### CASO DE DIVERGENCIA Y CONVERGENCIA AND

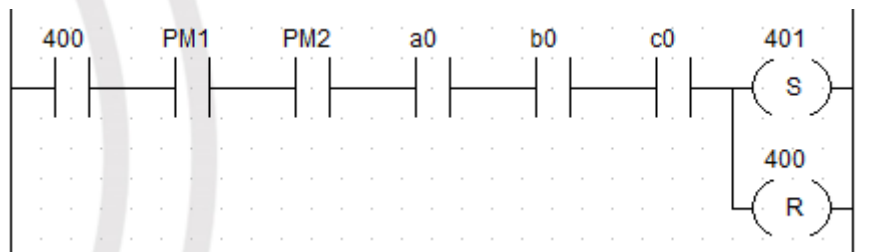
En este caso debemos tener en cuenta que, en el momento de la divergencia, una transición habilita varios caminos a la vez. Mientras que, al momento de la convergencia, hay varias etapas que se juntan en una sola. Tomemos el siguiente Grafcet como ejemplo



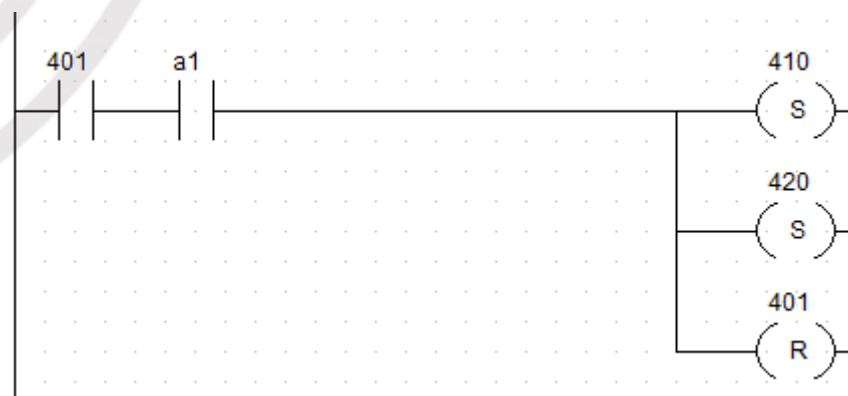
Para este caso tendremos una tabla de asociación de etapas con auxiliares como la siguiente:

Etapa	Auxiliar
0	400
1	401
10	410
11	411
12	412
20	420
21	421
22	422
3	403

La inicialización requerirá tener todos los bits auxiliares en serie y negados. Mientras que el accionamiento mostrará que salida, temporizador o contador activará cada auxiliar. Concentrémonos en ver como se arma el encadenamiento de este caso.

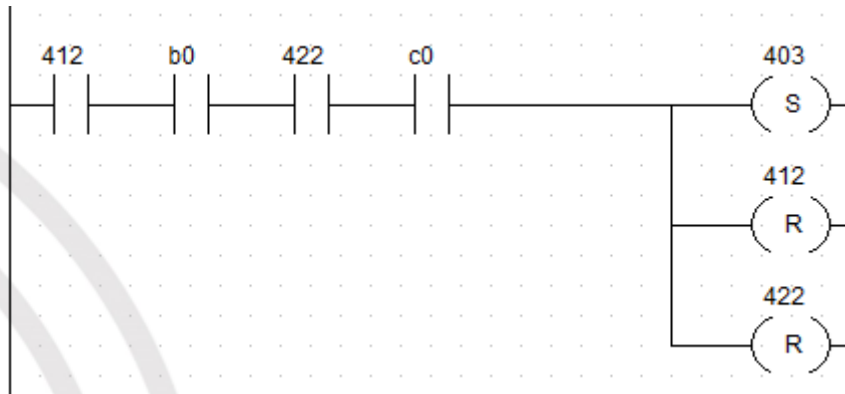


El pasaje de la etapa 0 a la etapa 1 es como lo hemos hecho hasta ahora. Con las condiciones de las transiciones se setea 401 y se resetea 400.



Cuando se convalida la transición de la etapa 1 (pisar el fin de carrera a1) se deben activar **simultáneamente** las etapas 10 y 20. Por ende, colocamos dos seteos, uno de 410 y otro de 420, para avanzar a ambas etapas. Luego apagamos la etapa activa 401.

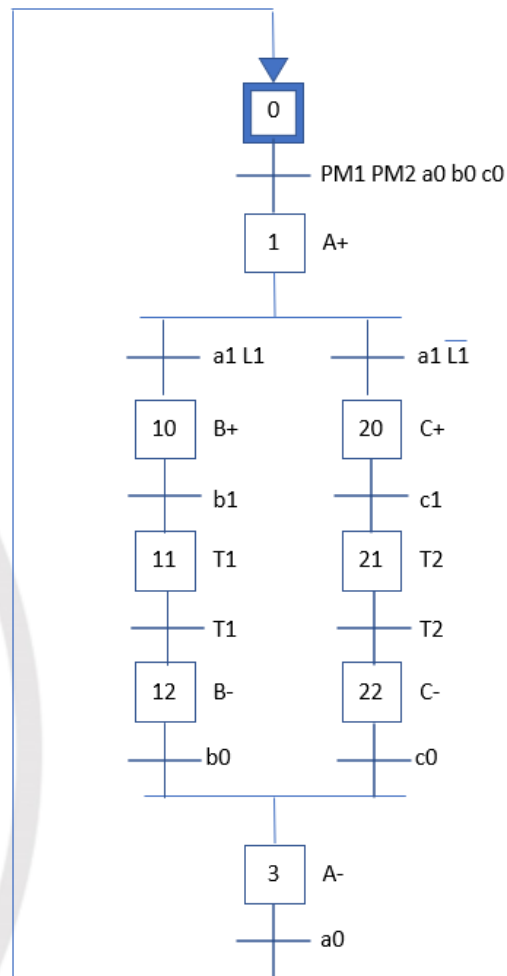
Con esto, resolvemos cada rama por separado, primero programamos todo el encadenamiento de las etapas 10, 11, 12 (como un Grafcet lineal) y luego todo el encadenamiento de las etapas 20, 21, 22 (como otro Grafcet lineal).



Al llegar a la convergencia, programamos el último auxiliar de una rama (412) con su transición (b0) en serie con el último auxiliar de la otra rama (422) con su transición (c0). Todo esto desemboca en el seteo de la etapa siguiente (etapa 3, auxiliar 403) y los dos resets de las etapas activas. De acá en adelante, el Grafcet sigue de manera lineal.

#### CASO DE DIVERGENCIA Y CONVERGENCIA OR

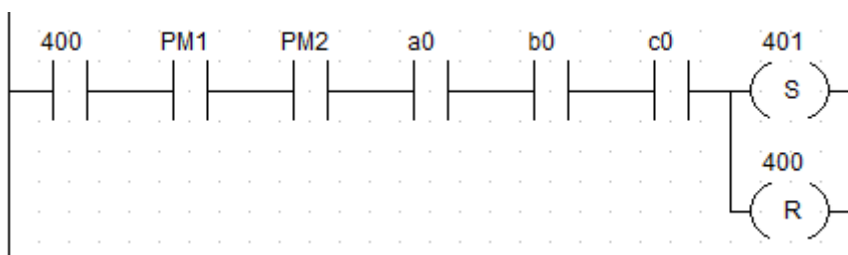
En este caso debemos tener en cuenta que, en el momento de la divergencia, solo se convalida una de todas las transiciones posibles y el resto no. Al momento de la convergencia, solo se puede verificar la rama validada en la divergencia para seguir por el camino lineal. Tomemos el siguiente Grafcet como ejemplo



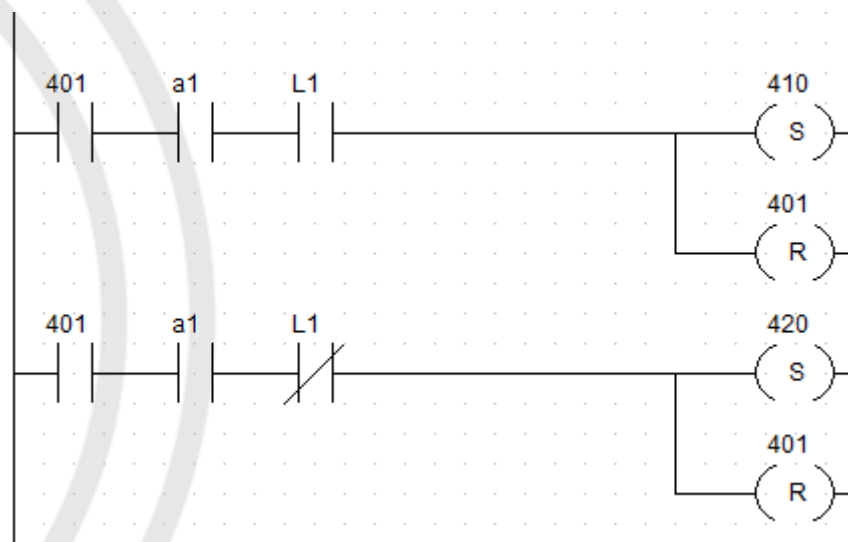
Para este caso tendremos una tabla de asociación de etapas con auxiliares como la siguiente:

<b>Etapas</b>	<b>Auxiliar</b>
<b>0</b>	400
<b>1</b>	401
<b>10</b>	410
<b>11</b>	411
<b>12</b>	412
<b>20</b>	420
<b>21</b>	421
<b>22</b>	422
<b>3</b>	403

La inicialización requerirá tener todos los bits auxiliares en serie y negados. Mientras que el accionamiento mostrará que salida, temporizador o contador activará cada auxiliar. Concentrémonos en ver como se arma el encadenamiento de este caso.

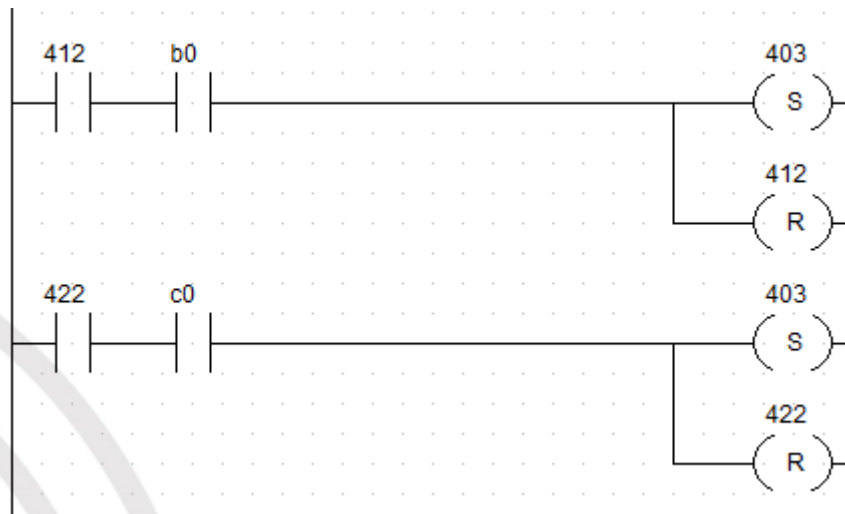


El pasaje de la etapa 0 a la etapa 1 es como lo hemos hecho hasta ahora. Con las condiciones de las transiciones se setea 401 y se resetea 400.



Al momento de activar 401 y pisar el fin de carrera a1, **solo una** de las dos líneas de programación puede activarse. Si L1 tiene señal, se setea 410 y se sigue ese camino. En cambio si L1 no tiene señal, se setea 420 y se sigue ese camino.

Con esto, resolvemos cada rama por separado, primero programamos todo el encadenamiento de las etapas 10, 11, 12 (como un Grafcet lineal) y luego todo el encadenamiento de las etapas 20, 21, 22 (como otro Grafcet lineal).



Al llegar a la convergencia, según cual camino hayamos tomado se verifica la condición que le corresponde. En ambos casos se termina seteando la señal de la etapa 3 (auxiliar 403) y se resetea el auxiliar que corresponda. Luego se sigue por el camino lineal del Grafcet.