

GENERACIÓN AUTOMÁTICA DE EXPRESIONES LÓGICAS DE UN PROGRAMA DE CONTROL BASADO EN GRAFCET. APLICACIÓN AL AUTÓMATA PROGRAMABLE SIMATIC S7-200

Jose Luis Pérez Quintas
jquntas@isa.uniovi.es

Felipe Mateos Martín
felipe@isa.uniovi.es

Victor M. González Suárez
victor@isa.uniovi.es

Área de Ingeniería de Sistemas y Automática. Dpto. de Ingeniería Eléctrica. Universidad de Oviedo. E.U.I.T.I.G. Avda Manuel Llaneza, 75. 33208-Gijón, Asturias

Tfno: 985182084 Fax: 985182068

Resumen

En esta comunicación se analizan y resuelven los problemas derivados de la implementación del diagrama funcional (Grafcet), como método de diseño de automatismos secuenciales, en equipos de control del tipo autómatas programables industriales.

En cuanto a la evolución de las etapas en el Grafcet, estos problemas se deben por un lado al modo de ejecución del programa de mando en el autómata, de carácter principalmente secuencial, por otro al franqueo instantáneo de transiciones, y adicionalmente al uso de estructuras basadas en repeticiones condicionales. También se presentan problemas cuando se utilizan en un Grafcet acciones mantenidas junto con acciones continuas sobre la misma señal. Las soluciones aportadas tienen aplicación general a cualquier autómata que incorpore las instrucciones combinacionales y de memoria básicas, temporizaciones y contajes. Se ofrecen dos alternativas posibles a las soluciones aportadas en la generación de ecuaciones: la asociación etapa-biestable y la asociación etapa-relé de memoria.

El estudio ha permitido el desarrollo de un módulo software para generación automática de ecuaciones lógicas asociadas a cualquier Grafcet y su traducción posterior a código de los autómatas de la serie Simatic S5 y S7-200 de Siemens, donde se ha verificado el correcto funcionamiento del método.

1. Introducción

Resulta comunmente aceptada la utilización de Grafcet como metodología de diseño de sistemas secuenciales. En ocasiones, los fabricantes de autómatas, disponen de herramientas de software de programación basadas en Grafcet que ofrecen una alternativa más o menos automatizada de implementación de aplicaciones de control. Sin embargo, muchos técnicos utilizan escasamente estas herramientas, debido a la no disponibilidad de

las mismas por parte del fabricante o porque se presenta como una herramienta opcional, generalmente de alto coste ó que sólo cubre en parte las necesidades. Esto hace que el Grafcet se utilice en muchos casos como medio para reflejar la evolución de etapas del automatismo, pero no para sistematizar la obtención de código en base al mismo, valiéndose de la experiencia por parte del técnico, no siempre adecuada, lo que conduce a errores de programación, aumentando los tiempos de desarrollo y puesta en marcha de automatismos.

2. Generación de expresiones

Si se generalizan la forma de obtener las expresiones lógicas que marcan la evolución de un grafcet que aparecen en [2] y [3], se tendría para la asociación etapa-relé de memoria

$$X_i = / Activables_i \cdot (\sum Activos_i \cdot Franqueo_i + X_i)$$

Mientras que para la asociación etapa-biestable se tiene

$$S(X_i) = \sum Activos_i \cdot Franqueo_i \\ R(X_i) = \sum Activables$$

donde *Activables* es el conjunto de estados activables una vez que esté activa la etapa *i*; *Activos* es el conjunto de etapas que estando activas hacen que la etapa *i* esté en estado activable; por último *Franqueo* son las receptividades que junto las etapas activas provocan la activación de la etapa *i*. A continuación se desglosarán los distintos problemas localizados, así como sus soluciones para cada una de las estructuras del grafcet.

3. Evolución secuencial

En la **Fig. 1** se muestra un Grafcet que describe una estructura en línea. Para este grafcet y siguiendo la asociación etapa-relé se tiene el siguiente conjunto de expresiones:

$$X_0 = / X_1 \cdot (X_2 \cdot T_3 + X_0) \\ X_1 = / X_2 \cdot (X_0 \cdot T_1 + X_1)$$

$$X_2 = / X_0 \cdot (X_1 \cdot T_2 + X_2)$$

En estas ecuaciones aparecen los primeros errores debidos a que la evaluación de las mismas en un PLC sigue una evolución secuencial y no concurrente, y por tanto primero se evalúa la expresión para la etapa 0, luego para la etapa 1, y finalmente la etapa 2.

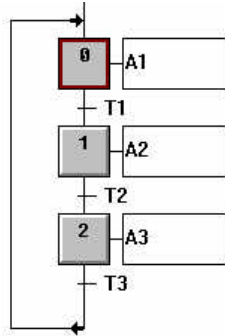


Fig. 1. Estructura Secuencial

Debido a esto si en un ciclo de programa esta activa la etapa 0, y se valida la receptividad T_1 se tendrían activas en el mismo ciclo las etapas 0 y uno, rompiendo la principal regla del grafcet. Esto se complicaría si se validarán a la vez las receptividades T_1 y T_2 estando la etapa 0 activa pues se tendrían activas las etapas 0, 1 y 2, además se activarían en un mismo ciclo dos etapas consecutivas. Este problema se resuelve con la simple inversión del orden de las ecuaciones, es decir en su implementación en el PLC primero se debe evaluar la etapa 2, luego la 1 y finalmente la 0. Si dichas expresiones tuvieran la forma de la asociación etapa-biestable las expresiones serían:

$$\begin{aligned} S(X_0) &= X_2 \cdot T_3 \\ R(X_0) &= X_1 \\ S(X_1) &= X_0 \cdot T_1 \\ R(X_1) &= X_2 \\ S(X_2) &= X_1 \cdot T_2 \\ R(X_2) &= X_0 \end{aligned}$$

En este tipo de ecuaciones aparecen los mismos problemas que en los anteriores, pero en este caso la solución no está sólo en la inversión del orden de las ecuaciones, pues si fuera así en el mismo ciclo pueden estar activas la primera y última etapa. Tras estudiar distintas alternativas en el orden de las etapas se encontró la solución invirtiendo el orden de activación de etapas y la desactivación de etapas se debe hacer inmediatamente después de que la consulta de una etapa, junto con una transición provoque la activación de otra etapa. Para el ejemplo mostrado se tendría:

$$\begin{aligned} S(X_2) &= X_1 \cdot T_2 \\ R(X_1) &= X_2 \\ S(X_1) &= X_0 \cdot T_1 \end{aligned}$$

$$R(X_0) = X_1$$

$$S(X_0) = X_2 \cdot T_3$$

$$R(X_2) = X_0$$

4. Divergencias en O

En la Fig. 2 se representa un ejemplo de grafcet con una estructura alternativa.

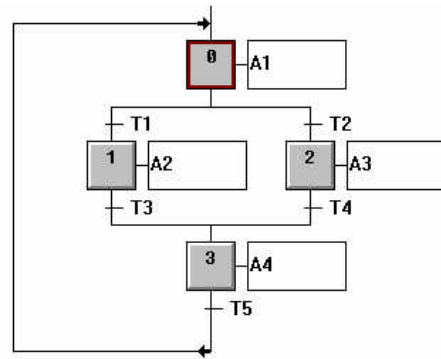


Fig. 2. Divergencia en O

El conjunto de expresiones que representarían este grafcet, teniendo ya en cuenta la inversión de expresiones, serían las siguientes:

$$\begin{aligned} X_3 &= / X_0 \cdot (X_1 \cdot T_3 + X_2 \cdot T_4 + X_3) \\ X_2 &= / X_3 \cdot (X_0 \cdot T_2 + X_2) \\ X_1 &= / X_3 \cdot (X_0 \cdot T_1 + X_1) \\ X_0 &= / X_1 \cdot / X_2 \cdot (X_2 \cdot T_5 + X_0) \end{aligned}$$

Si se generasen así las expresiones para el grafcet, basándose en la asociación etapa-relé de memoria en el caso de estar activa la etapa 0 y validarse las receptividades T_1 y T_2 se activarían las etapas 1 y 2, violando la regla del grafcet que indica que en una divergencia en O sólo puede haber una rama activa. Este error se corrige añadiendo una condición más a las etapas que inician las ramas de las divergencias en O, dicha condición será que para poder activarse una rama no puede estar activa ninguna otra rama de la divergencia.

Además existe otro problema pues por normativa grafcet recogida en [1] en el caso de que haya la posibilidad de entrar por varias ramas en una divergencia en O, siempre se debe seleccionar la rama de más a la izquierda. Esto se soluciona implementando primero la ecuación relativa a la etapa 1 y tras esta la relativa a la etapa 2.

Con esto las expresiones quedarían tal como siguen:

$$\begin{aligned} X_3 &= / X_0 \cdot (X_1 \cdot T_3 + X_2 \cdot T_4 + X_3) \\ X_1 &= / X_2 \cdot / X_3 \cdot (X_0 \cdot T_1 + X_1) \\ X_2 &= / X_1 \cdot / X_3 \cdot (X_0 \cdot T_2 + X_2) \end{aligned}$$

$$X_0 = / X_1 ./ X_2 \cdot (X_2 T_5 + X_0)$$

En el caso de la asociación etapa-biestable se tendrían las siguientes expresiones:

$$S(X_3) = X_1 T_3 + X_2 T_4$$

$$R(X_1) = X_3$$

$$R(X_2) = X_3$$

$$S(X_2) = X_0 T_2$$

$$R(X_0) = X_2$$

$$S(X_1) = X_0 T_1$$

$$R(X_0) = X_1$$

$$S(X_0) = X_2 T_5$$

$$R(X_2) = X_0$$

En este tipo de expresiones no aparece el problema de la activación de varias ramas de una divergencia en O, pues cuando se activa la etapa de inicio de una rama se desactiva inmediatamente la etapa que da origen a su activación y por tanto no se pueden activar otras ramas. En el ejemplo, una vez que se activa la etapa 2, inmediatamente se desactiva la etapa 0, y por tanto no se puede activar la etapa 1. No obstante presenta el problema de que no se da prioridad a la rama de más a la izquierda, solucionándose este problema con la alteración del orden de las ecuaciones, tal como se muestra a continuación:

$$S(X_3) = X_1 T_3 + X_2 T_4$$

$$R(X_1) = X_3$$

$$R(X_2) = X_3$$

$$S(X_1) = X_0 T_1$$

$$R(X_0) = X_1$$

$$S(X_2) = X_0 T_2$$

$$R(X_0) = X_2$$

$$S(X_0) = X_2 T_5$$

$$R(X_2) = X_0$$

5. Repeticiones Condicionales

Es en las repeticiones condicionales de un graficet donde se localizan los problemas mas graves en la implementación de las ecuaciones que representa un graficet. Para su estudio se verá un caso extremo, a partir del cual se podrá generalizar.

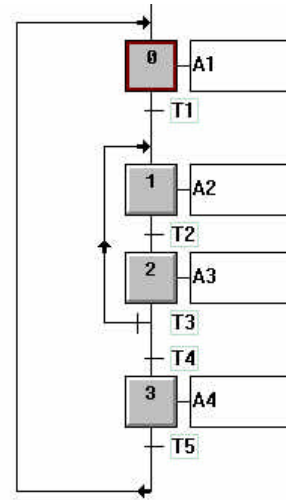


Fig. 3. Repetición condicional

Según lo comentado hasta ahora las ecuaciones para el graficet de la Fig. 3 deberían ser:

$$X_3 = / X_1 ./ X_0 \cdot (X_2 T_4 + X_3)$$

$$X_2 = / X_3 ./ X_1 \cdot (X_1 T_2 + X_2)$$

$$X_1 = / X_2 \cdot (X_0 T_1 + X_2 T_3 + X_1)$$

$$X_0 = / X_1 \cdot (X_3 T_5 + X_0)$$

En este tipo de estructura, que es un caso particular de una divergencia en O, el mayor problema es que la etapa 2 **nunca** se activa debido a que existe una contradicción en su expresión, pues por un lado para su activación se indica que no puede estar activa la etapa 1, pero viendo la evolución del graficet se ve que el estado 2 debe activarse cuando esté activa la etapa 1 y se franquee T_2 . También aparece una contradicción para la activación de la etapa 1, a través de la etapa 2. Este caso no se puede solucionar mediante la alteración del orden de las expresiones, pues el error está en las propias expresiones. Una solución a este problema es colocar, a la hora de generar las expresiones, una etapa y una transición cierta en el salto que va desde la etapa 2 a la uno, y la expresión necesaria para este estado situarla la primera. Gráficamente resulta lo siguiente:

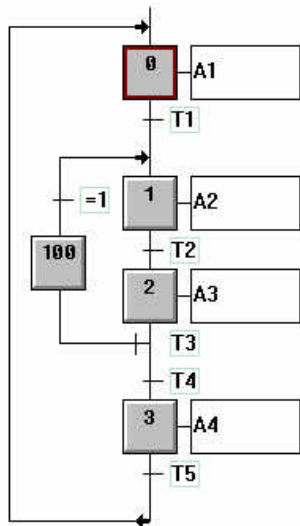


Fig. 4. Inserción de etapa intermedia

Con esto las expresiones quedan:

$$X_{100} = / X_1 \cdot (X_2 T_3 + X_{100})$$

$$X_3 = / X_0 \cdot (X_2 T_4 + X_3)$$

$$X_2 = / X_{100} ./ X_3 \cdot (X_1 T_2 + X_2)$$

$$X_1 = / X_2 \cdot (X_0 T_1 + X_{100} + X_1)$$

$$X_0 = / X_1 \cdot (X_3 T_5 + X_0)$$

De todas formas estas ecuaciones siguen presentando problemas ya que si se diera el caso de que estando activa la etapa 2 se verificasen las transiciones T_3 y T_4 se activarían las etapas 100 y 3. Esto se debe a que las repeticiones condicionales tienen el comportamiento de una divergencia en O, es decir, una alternativa y por tanto la solución será la misma que para dicha estructura. Otro problema esta en que la activación de la etapa inicial es una vuelta hacia atrás en el ciclo, lo cual pueden originar problemas, y para evitarlos se resuelve de forma similar a una repetición condicional. Con esto las ecuaciones quedan como siguen:

$$X_{200} = / X_0 \cdot (X_3 T_5 + X_{200})$$

$$X_{100} = / X_1 ./ X_3 \cdot (X_2 T_3 + X_{100})$$

$$X_3 = / X_{200} ./ X_{100} \cdot (X_2 T_4 + X_3)$$

$$X_2 = / X_3 ./ X_{100} \cdot (X_1 T_2 + X_2)$$

$$X_1 = / X_2 \cdot (X_0 T_1 + X_{100} + X_1)$$

$$X_0 = / X_1 \cdot (X_{200} + X_0)$$

El ejemplo anterior traducido a código S7-200 es el siguiente:

```
NETWORK
LDN "X0" // X200 = / X0 . (X3 T5 + X200)
LD "X3"
U "T5"
LD "X200"
```

```
OLD
ULD
= "X200"

NETWORK
LDN "X1" // X100 = / X1 ./ X3 . (X2 T3 + X100)
UN "X3"
LD "X2"
U "T3"
LD "X100"
OLD
ULD
= "X100"

NETWORK
LDN "X200" // X3 = / X200 ./ X100 . (X2 T4 + X3)
UN "X100"
LD "X2"
U "T4"
LD "X3"
OLD
ULD
= "X3"

NETWORK
LDN "X3" // X2 = / X3 ./ X100 . (X1 T2 + X2)
UN "X100"
LD "X1"
U "T2"
LD "X2"
OLD
ULD
= "X2"

NETWORK
LDN "X2" // X1 = / X2 . (X0 T1 + X100 + X1)
LD "X0"
U "T1"
LD "X100"
OLD
LD "X200"
OLD
ULD
= "X1"

NETWORK
LDN "X1" // X0 = / X1 . (X200 + X0)
LD "X200"
O "X0"
ULD
= "X0"
```

En cuanto a la asociación etapa-biestable se tienen las siguientes ecuaciones:

$$S(X_{200}) = X_3 \cdot T_5$$

$$R(X_3) = X_{200}$$

$$S(X_{100}) = X_2 \cdot T_3$$

$$R(X_2) = X_{100}$$

$$S(X_3) = X_2 \cdot T_4$$

$$R(X_2) = X_3$$

$$S(X_2) = X_1 \cdot T_2$$

$$R(X_1) = X_2$$

$$S(X_1) = X_0 \cdot T_1 + X_{100}$$

$$R(X_0) = X_1$$

$$R(X_{100}) = X_1$$

$$S(X_0) = X_{200}$$

$$R(X_{200}) = X_0$$

El código STEP-7 (Simatic S7-200) es el siguiente:

```
NETWORK
```

```

LD "X3" // S( X200 ) = X3 .T5
U "T5"
S "X200",1

NETWORK
LD "X200" // R( X3 ) = X200
R "X3",1

NETWORK
LD "X2" // S( X100 ) = X2 .T3
U "T3"
S "X100",1

NETWORK
LD "X100" // R( X2 ) = X100
R "X2",1

NETWORK
LD "X2" // S( X3 ) = X2 .T4
U "T4"
S "X3",1

NETWORK
LD "X3" // R( X2 ) = X3
R "X2",1

NETWORK
LD "X1" // S( X2 ) = X1 .T2
U "T2"
S "X2",1

NETWORK
LD "X2" // R( X1 ) = X2
R "X1",1

NETWORK
LD "X0" // S( X1 ) = X0 T1 + X100
U "T1"
LD "X100"
OLD
S "X1",1

NETWORK
LD "X1" // S( X1 ) = X0 T1 + X100
R "X0",1

NETWORK
LD "X1" // R( X100 ) = X1
R "X100",1

NETWORK
LD "X200" // S( X0 ) = X200
S "X0",1

NETWORK
LD "X0" // R( X200 ) = X0
R "X200",1

```

6. Activación de salidas

También existen problemas con la ejecución de acciones continuas y mantenidas mediante operaciones de Set-Reset sobre una misma señal, debido a la forma que tienen los PLC's de actualizar la memoria asociada a las acciones del sistema. Si se tiene el grafcet de la **Fig. 5**.

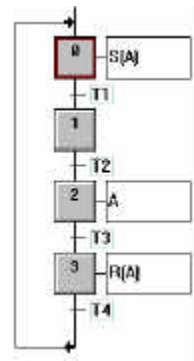


Fig. 5: Ejemplo de acciones

El conjunto de ecuaciones para activación de salidas, en formato Simatic S7-200, sería el siguiente:

```

NETWORK
LD "X0"
S "A",1

NETWORK
LD "X1"
= "A"

NETWORK
LD "X3"
R "A",1

```

Según estas ecuaciones la acción A sólo se lanzaría cuando estuviese activa la etapa 1, pues aunque se haga un Set sobre A en la etapa 0, al evaluar más tarde la etapa 1, es el estado de esta el que se pasa a la acción A.

Para evitar este problema se debe usar una marca auxiliar sobre la cual se hagan el Set y Reset, y por último la acción se compondría con la consulta a la etapa 1, más la combinación O con la marca auxiliar.

```

NETWORK
LD "X0"
S "TMP",1

NETWORK
LD "X3"
R "TMP",1

NETWORK
LD "X1"
O "TMP"
= "A"

```

7. Aplicación software

La generación automática de estas expresiones, siguiendo las reglas indicadas en esta comunicación, se puede realizar desde la aplicación **MEDISS (Metodología para el Diseño de Sistemas Secuenciales)**, desarrollada en el Área de Ingeniería de Sistemas y Automática de la Universidad de Oviedo:

MEDISS es una aplicación software que se ejecuta bajo entorno Windows y que permite el diseño de programas de control para los autómatas Simatic S5 y S7-200 basados en grafcet. Consta de dos componentes principales:

El Módulo de Edición: Basado en la normativa Grafcet se construye el diagrama funcional del

automatismo incluyendo las acciones y transiciones necesarias. A partir del mismo se generan automáticamente las ecuaciones secuenciales y de ejecución de acciones, así como el programa de control para Simatic S5 ó S7-200. Se incluyen también capacidades de documentación.

El Módulo VISGRAF: Esta aplicación permite por un lado la transmisión del código generado al autómeta programable y la supervisión del programa de control a través del Grafcet (evolución de etapas). Pueden monitorizarse los estados de las entradas y salidas del autómeta, temporizadores, contadores, etc...

Una presentación general de la aplicación se muestra en la **Fig. 6**. En la izquierda aparece una pantalla típica del módulo de Edición (Mediss), con un grafcet construido y la ventana auxiliar donde se indica la información obtenida a partir del mismo, en concreto, el programa para los autómetas de la gama Simatic S5. En la parte derecha se presenta un detalle del módulo de Visualización (Visgraf) con los valores de temporizadores y contadores, las señales de entrada y salida, y el grafcet en el que se representa la evolución del automatismo.

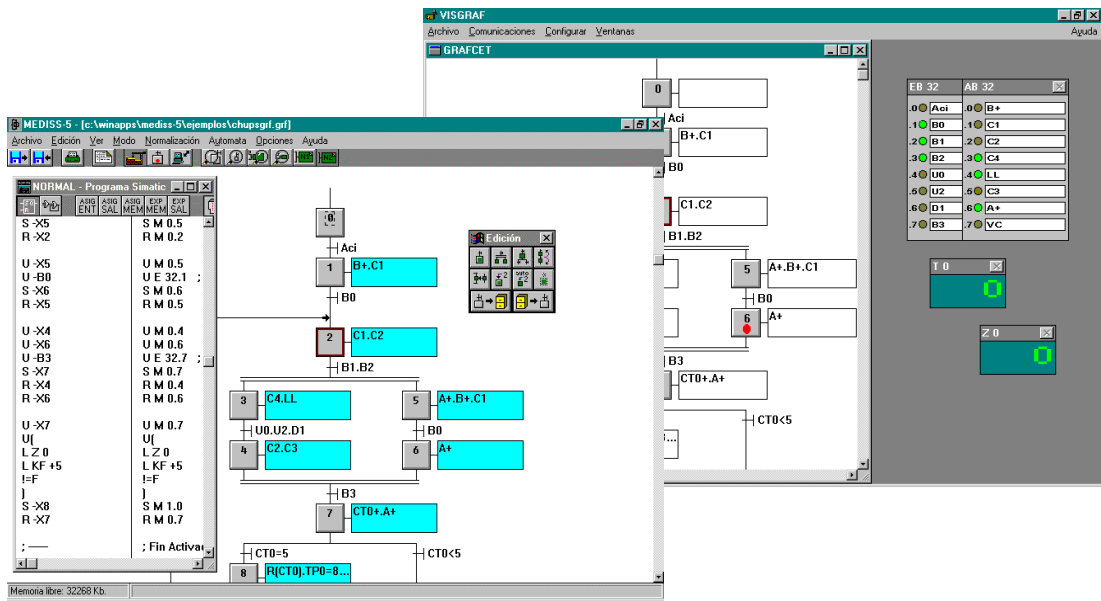


Fig. 6: Aspecto de las aplicaciones MEDISS y VISGRAF

Las ventanas con información relativa al conjunto de expresiones se ven en Fig. 7 y Fig. 8, mientras que en Fig. 9 se muestra la ventana con el código STEP-7 (Simatic S7-200) generado por la aplicación



Fig. 7: Asoc. Etapa-Biestable



Fig. 8: Asoc. Etapa-Relé de Memoria

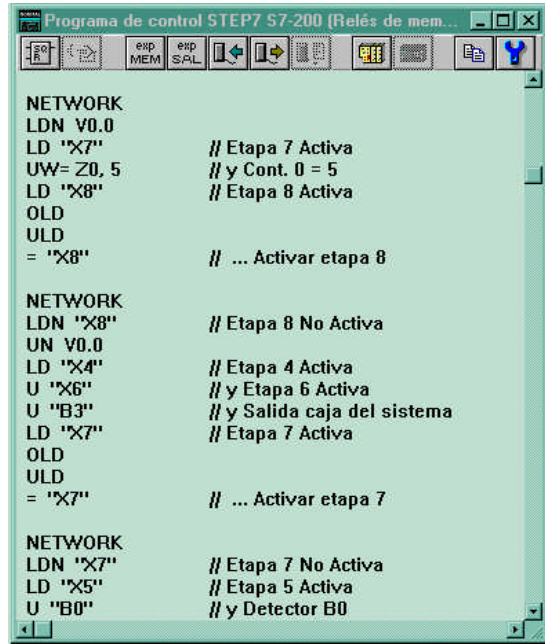


Fig. 9: Código S7-200

Referencias

- [1] *Diagramme Fonctionnel GRAFCET pour la Description des Systèmes Logiques de Commande* NF C 03-190 Union Technique de l'Electricité, 1982
- [2] G. Michel, *Autómatas Programables Industriales. Arquitectura y aplicaciones*, Marcombo 1990, pp. 256-272
- [3] *Preparation of Function Charts for Control System* International Standard CEI/IEC 848, 1988