

IMPLEMENTACIÓN DE ALGORITMOS DE CONTROL DE TRÁFICO VEHICULAR CON REDES DE PETRI NO BINARIAS UTILIZANDO DIAGRAMAS EN ESCALERA

Jhony Andrés Arroyave Díaz ¹, Manuel Alejandro Zuleta Bermúdez ², Rubén Darío Vásquez Salazar ³

¹ Ingeniero Electromecánico. Inspector de Ultrasonido en el equipo de Predictivo y Gestión del Área Mantenimiento Equipos de Empresas Publicas de Medellín E.S.P. jhony.arroyave@gmail.com

² Ingeniero Electromecánico. Ingeniero Analista de Equipos en Geominas S.A. zule0536@hotmail.com

^{1, 2} Instituto Tecnológico Metropolitano (Institución Universitaria) – ITM, Calle 73 No 76A – 354 (Vía El Volador), Medellín, Colombia, en 2011.

³ Magíster en Ingeniería – Automatización Industrial - Universidad Nacional de Colombia sede Manizales. Docente de tiempo completo - Facultad de Ingenierías – Politécnico Colombiano Jaime Isaza Cadavid, carrera 48 No 7-151, Medellín, Colombia. rdvasquez@elpoli.edu.co.

RESUMEN

Este artículo se concentra en el estudio de las redes de Petri (RdP) no binarias, sus componentes, capacidad de modelar sistemas a eventos discretos y principalmente la viabilidad y limitaciones para su programación en controladores lógicos programables (PLCs). Se hace un comparativo de las redes de Petri binarias con las no binarias, encontrándose principalmente 2 diferencias: la cantidad de marcas en los estados y el peso de los arcos que afecta la evolución del marcaje. Se propone entonces un método para implementar las no binarias en los PLCs. Dicha metodología se aplica al control de sistemas de tráfico vehicular, teniendo en cuenta la cantidad de vehículos que esperan sobre 2 vías que se cruzan, para modificar los tiempos en verde y en rojo de cada semáforo y así optimizar el tránsito sobre la intersección. Al final se presentan los resultados de la simulación hecha en lenguaje Ladder.

Palabras clave: Control de tráfico vehicular, controladores lógicos programables, redes de Petri no binarias, sistemas a eventos discretos.

Recibido: 12 de junio de 2012.
Received: June 12th 2012.

Aceptado: 16 de octubre de 2012.
Accepted: October 16th, 2012.

IMPLEMENTATION OF LOGIC CONTROL ALGORITHMS WITH NON-BINARY PETRI NETS BY USING LADDER DIAGRAMS

ABSTRACT

This article focuses on the study of non binary Petri nets, its components, ability to model discrete event systems and especially the feasibility and limitations for its programming by using programmable logic controllers (PLCs). A comparison between binary Petri nets and non-binary is made, finding primarily 2 differences: the number of tokens in the states and the weight of the arcs that affect the evolution of the marking. Thus, a method for implementing non-binary on this language is proposed. This methodology is applied to control traffic systems, taking into account the number of vehicles that wait for 2-way cross, to change the time in green and red in every stoplight and optimize the traffic on the intersection. At the end, the results of the obtained simulation in Ladder language are presented.

Keywords: *discrete event systems, non-binary Petri nets, programmable logic controllers, traffic control.*

1. INTRODUCCIÓN

A medida que los procesos de tecnificación se hacen más complejos tanto en su estructura como en el funcionamiento de las partes que lo integran, eventos y sistemas que controlan y costos derivados de la construcción de prototipos, es necesario contar con sistemas de modelación gráfica, que permitan caracterizar sus partes y conexiones; o bien, permitan desarrollar un esquema en el cual se observe las reacciones del sistema frente a variables y acontecimientos externos.

Dentro de los controladores lógicos programables (PLC) las RdP son un modelo gráfico, formal y abstracto para describir el flujo de información de un proceso [1], además han sido muy útiles para realizar la programación en PLC de la especificación del proceso dada.

Existen diferentes metodologías para traducir RdP binarias a lenguaje Ladder [2], en las cuales se toma cada elemento de la red y se asigna a un elemento de programación en lenguaje Ladder.

Además existe una metodología para traducir RdP no binarias a lenguaje Ladder, la cual toma cada lugar como un contador al que se le incrementa o decrementa una unidad dependiendo del disparo de las transiciones simulando la llegada o salida de marcas [3], pero no utiliza el peso de los arcos para realizar la evolución de varias marcas de forma simultánea.

Las RdP no binarias tienen entonces algunas diferencias significativas respecto a las binarias, entre ellas la cantidad de marcas que tienen en cada uno de sus lugares, 0 ó 1 para las binarias, y enteros positivos para las no binarias, y el peso de los arcos, lo cual representa la cantidad de marcas que un arco transporta de un lugar a una transición o viceversa.

Ello trae consigo cambios en la traducción a lenguaje Ladder, ya que no se pueden utilizar los mismos elementos binarios como contactos, bobinas *set* y *reset* que se utilizaban para las binarias.

Se inicia entonces con la definición teórica de las redes de Petri, se hace una analogía entre las RdP y el lenguaje de programación Ladder según las metodologías existentes, se resaltan las principales

diferencias entre las binarias y las no binarias a través de un breve ejemplo y se propone la metodología, la cual se aplica a un sistema de tráfico vehicular.

2. SISTEMAS A EVENTOS DISCRETOS Y REDES DE PETRI

La frase "*Eventos Discretos*" está generalmente asociada con algunos formalismos muy populares como autómatas de estados finitos, RdP, grafos de eventos (Event Graphs), Statecharts, GRAFCET, etc [4]. Desafortunadamente, no todos ellos pueden representar todo tipo de sistemas a eventos discretos ya que algunos de ellos poseen características muy particulares.

Estos lenguajes gráficos se limitan a sistemas con un número finito de estados posibles, mientras que en este caso se requiere una herramienta más general. De todas formas, tal formalismo general existe y se conoce como DEVS (Discrete Event System Specification). El formalismo DEVS fue desarrollado por Bernard Zeigler a mediados de los setentas. El uso de DEVS en relación con sistemas continuos no está aún muy difundido y es prácticamente desconocido en las comunidades de métodos numéricos y control. Sin embargo, DEVS es ampliamente utilizado en ciencias de la computación donde ha recibido un muy importante desarrollo en sus aspectos teóricos y prácticos [4].

DEVS permite representar todos los sistemas cuyo comportamiento entrada-salida puede ser descrito por secuencias de eventos sujeto a que el estado tenga un número finito de cambios en cualquier intervalo finito de tiempo. Las RdP representan el método más efectivo, tanto para el diseño como para el control de DEVS [3].

Las RdP surgieron en 1962 con el trabajo doctoral de Carl Adam Petri "*Kommunikation mit Automaten*" (Comunicación con autómatas), en Alemania. En éste Petri formuló la base para una teoría de comunicación entre componentes asincrónicos de un sistema de cómputo.

Las ideas de Petri atrajeron la atención de un grupo de investigadores del *Applied Data Research Inc.* Dirigido por Anatol Holt y que trabajaban en el proyecto "*Information System Theory Project*". El

grupo, desarrolló la teoría del proyecto conocido como "Systemics". Este trabajo fue el que proporcionó la teoría primaria, notación y representación de las Redes de Petri [5].

2.1 Definición

Fundamentalmente una Red de Petri ordinaria (RdPO), N , es un cuádruplo [6].

$N = \langle P, T, Pre, Post \rangle$, donde:

$P = \{p_1, p_2, \dots, p_m\}$, es un conjunto finito y no vacío de lugares;

$T = \{t_1, t_2, \dots, t_n\}$, es un conjunto finito y no vacío de transiciones;

$P \cap T = \emptyset$ y $P \cup T = \emptyset$

$Pre : P \times T \rightarrow \{0,1\}$, es el conjunto de lugares de entrada a T

$Post : T \times P \rightarrow \{0,1\}$, es el conjunto de lugares de salida de T

Una RdPO marcada es un par $Nm = \langle N, M_0 \rangle$ en el cual N es una Red de Petri Ordinaria y M_0 es el marcado inicial.

2.2 Peso de los arcos

Una RdP controlada por pesos de los arcos está definida por una tupla:

$\varrho = (P, T, W, C, B)$, donde:

P , es un conjunto finito de lugares.

T , es un conjunto finito de transiciones.

$W : (P \times T) \cup (T \times P) = N$, es una función que indica el peso de los arcos conectores entre los lugares y las transiciones (donde N es un entero no negativo del conjunto de los enteros).

C , es un conjunto finito de lugares de control.

$B : (C \times T) = \{0,1\}$, es el peso binario de un arco dirigido; asociando los lugares con las transiciones.

El peso de un arco desde un lugar p_i hasta una transición t_j es definido por $w(p_i, t_j)$ y el peso desde un arco desde t_j hasta p_i por $w(t_j, p_i)$.

Cuando una transición dispara, toma la cantidad de marcas requeridas para la salida y distribución hacia los siguientes lugares según su peso del arco. Si los pesos de los arcos son iguales para todos los lugares de salida, las marcas se mueven a través de la transición. Si existe alguna diferencia en los pesos de los arcos, las marcas pueden desaparecer o ser creadas según el lugar de salida al que estas se dirijan. De hecho una transición al disparar lo hace produciendo y eliminando marcas con lo que se puede decir que la transición trabaja en función del peso del arco.

3. RED DE PETRI Y LENGUAJE DE PROGRAMACIÓN

3.1 Programa y lenguaje de programación

Un PLC es un dispositivo capaz de leer el entorno a través de sus entradas y ejecutar unas acciones a través de sus salidas. En medio de las entradas y salidas se encuentra la lógica de control, la cual es un *programa* diseñado por un experto utilizando una sintaxis estructurada, utilizando instrucciones, símbolos y órdenes.

Los *lenguajes de programación* se han desarrollado y expandido de igual manera que los PLC. Los actuales poseen nuevas y más versátiles aplicaciones y con mayor capacidad de computación. Por ejemplo, los PLC pueden transferir bloques de datos de una localización de memoria a otra, mientras que llevan a cabo operaciones lógicas y matemáticas en otro bloque. Por lo que manejan datos de una manera más fácil y eficiente.

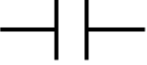
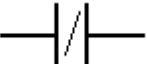




El lenguaje en escalera, comúnmente conocido como diagrama de contactos o por su traducción al inglés, Ladder, es uno de los más utilizados en los autómatas programables. Su simbología se compone de elementos de entrada y de salida, cada uno de los cuales se relaciona con una variable del PLC [7]. Al igual que otros 4 lenguajes está contenido en el estándar internacional IEC61131-3. Adicional a esto se han desarrollado nuevos módulos de entradas y salidas que han obligado a cambiar las instrucciones existentes [8].

3.2 Elementos de programación

Para programar un PLC con lenguaje Ladder, es necesario conocer cada uno de los elementos que hace parte de este [8]. La Tabla 1 muestra algunos

de los elementos básicos del lenguaje con su respectiva descripción.

Tabla 1. Elementos básicos de programación en lenguaje Ladder

Símbolo	Nombre	Descripción
	Contacto NA	Se activa cuando hay un uno lógico en el elemento que representa, esto es, una entrada (para captar información del proceso a controlar), una variable interna o un bit de sistema.
	Contacto NC	Su función es similar al contacto NA anterior, pero en este caso se activa cuando hay un cero lógico, cosa que deberá tenerse muy en cuenta a la hora de su utilización.
	Bobina NA	Se activa cuando la combinación que hay a su entrada (izquierda) da un uno lógico. Su activación equivale a decir que tiene un uno lógico. Suele representar elementos de salida, aunque a veces puede hacer el papel de variable interna.
	Bobina NC	Se activa cuando la combinación que hay a su entrada (izquierda) da un cero lógico. Su activación equivale a decir que tiene un cero lógico. Su comportamiento es complementario al de la bobina NA.
	Bobina SET	Una vez activa (puesta a 1) no se puede desactivar (puesta a 0) si no es por su correspondiente bobina en RESET. Sirve para memorizar bits y usada junto con la bobina RESET dan una enorme potencia en la programación.
	Bobina RESET	Permite desactivar una bobina SET previamente activada.

3.3 Relación entre las RdP y el lenguaje Ladder de programación

La principal relación es mostrar un procedimiento de diseño de diagramas capaz de resolver los problemas en las fases de construcción, test y mantenimiento de sistemas complejos con ayuda de las RdP. De acuerdo con los resultados de desarrollos antes realizados, los métodos propuestos se centraban principalmente en la fase de construcción en perjuicio de los demás. Sus metodologías estaban destinadas casi exclusivamente a trasladar las RdP a diagramas de contactos (Fig. 1). Sin embargo algunos problemas de las fases de test y de construcción son muy significativos. Por tanto es vital el desarrollo de una metodología que permita obtener programas más legibles en fase de ejecución, que permitan al programador conocer directamente en el programa o en tablas de observación el estado de la RdP y con ésta el del proceso controlado.

Además, si la unión entre la RdP y su correspondiente diagrama de contactos se encuentra adecuadamente estandarizada, el sistema de RdP será capaz de monitorear y diagnosticar el sistema real a un nivel más alto por medio del estado del PLC en tiempo real. Por tanto será posible reducir no solo el tiempo de diseño sino también la depuración y mantenimiento por medio de una metodología adecuada.

En primer lugar, el método debe ser unificado y aplicable a los PLC de diferentes marcas. A pesar que los diagramas de contactos son ampliamente utilizados con PLC, no se trata de un lenguaje de programación estándar. Muchos fabricantes de PLC incluyen diferentes sintaxis e instrucciones para aumentar la funcionalidad de los diagramas de contactos. Además, el mismo programa puede dar lugar a comportamientos diferentes cuando se ejecuta en autómatas distintos debido a sus modos de funcionamiento respectivos.

Para establecer una metodología unificada, que dé lugar al mismo resultado en diversos PLC, las múltiples características de los distintos sistemas deben ser consideradas con anterioridad. Por ejemplo, se debe evitar el uso de instrucciones no estándar. Sólo se eligen las instrucciones fundamentales tales como contactos, bobinas, etc. De esta forma se consigue un método más portable [9]. Esta portabilidad se ve limitada incluso en PLCs que cumplan con el estándar IEC61131-3, debido a que los protocolos de comunicación para

transferencia del programa al PLC y otros elementos que el estándar no define, dan libertad a los fabricantes, lo que impide la total unificación, incluso entre diferentes gamas del mismo

fabricante. Por lo tanto la portabilidad es gráfica pero la traducción y codificación debe realizarse manualmente.



Fig. 1. Relación entre RdP y diagrama Ladder de programación [9]

Por otro lado el programa debe ser fácil de probar y depurar. Además de generar el diagrama de contactos capaz de proporcionar el comportamiento deseado, el método de conversión debe preservar, en lo posible, la topología de la RdP original de forma que el programa resulte más fácil de seguir y ampliar con posterioridad. El diseño en diagrama de contactos debe ser capaz de ayudar a la RdP en su nivel superior para su uso en el monitoreo y diagnóstico de operaciones en tiempo real. Esto se logra si los estados de la RdP se representan por variables globales, de modo que puedan ser observadas desde cualquier programa o bloque del PLC. Al interior del PLC las marcas o banderas están definidas como variables globales, y pueden ser de diferentes tipos de acuerdo a su cantidad de bits y formatos, cumpliendo con las exigencias antes mencionadas. Por lo tanto, se eligen las marcas M tipo entero como elementos para representar dichos estados. Su sintaxis es "MW".

Por último, el método desarrollado debe ser aplicable a los casos más amplios y complejos. Por tanto debe ser considerada la conversión de las topologías más generales de las RdP con objeto de garantizar la diversidad de aplicaciones. Además de permitir la construcción de la secuencia general de control, el tabulado sistemático de las condiciones de entrada desde el exterior y de las salidas de control permitirá completar el procedimiento de diseño. Este diseño extendido puede ser utilizado como una herramienta adicional para la generación de acciones de control en

diagrama de contactos con vista a su posterior utilización en la monitorización y el diagnóstico. Por lo tanto, la metodología debe soportar las estructuras básicas de las RdP como las convergencias y divergencias AND y OR.

Tanto las RdP como los diagramas de contactos son herramientas apropiadas para representar modelos de estado y condiciones. En las RdP cada posición puede ser asociada a un posible estado del sistema. El marcado representa el estado en que se encuentra el sistema y las transiciones son las condiciones necesarias para que se produzca un cambio de estado. De la misma forma en diagrama de contactos una salida puede representar el estado de una posición de la RdP y la combinación de contactos de entrada puede representar las transiciones en la RdP. Siguiendo estas reglas las RdP pueden ser fácilmente traducidas a diagramas de contactos (Tabla 2).

4. REDES DE PETRI NO BINARIAS

Las RdP no binarias se rigen bajo los mismos conceptos básicos de las binarias, estas difieren de las últimas en que la evolución del marcado, aunque se da de igual manera a través de la RdP, depende del peso del arco de entrada y salida de cada una de las transiciones de la red, es decir, si el peso del arco de entrada a la transición es igual a 3 y el de salida es igual a 4, cada que un evento desaparece esta transición, este arco de entrada le

quitará tres marcas al lugar anterior y el arco de salida le dará cuatro marcas al siguiente lugar (Fig. 2).

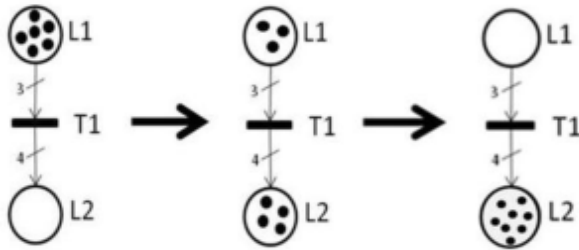


Fig. 2. Arco de peso no unitario en la RdP.

Cuando el peso del arco de salida es menor que la cantidad de marcas del lugar, de este último solo se tomará la cantidad de marcas que represente el peso del arco. Cuando un evento dispara una transición, el arco de entrada al siguiente lugar creará la cantidad de marcas necesarias según su peso (Fig. 2).

Dentro de la RdP no binaria se puede presentar ganancia o pérdida de marcas después del disparo de una transición, aunque este evento está sujeto al peso de los arcos de entrada y salida de cada uno de los lugares. Si el peso del arco de entrada a una transición T1 es menor que el peso del arco de salida de la misma, al disparar dicha transición habrá una ganancia de marcas (Fig. 2).

Tabla 2. Ejemplo de RdP binaria

EJEMPLO 1	EJEMPLO 2
<p>P1=M0.0 P2=M0.1 P3=M0.2 T1=E124.0 T2=E124.1*T0*E124.1 SALIDA1=A124.0 SALIDA2=A124.1</p>	<p>P1=M0.0 P2=M0.1 P3=M0.2 T1=E124.0 T2=E124.1 SALIDA1=A124.0 SALIDA2=A124.1</p>
<p>REPRESENTACIÓN DIAGRAMA LADDER O DE CONTACTOS</p>	<p>REPRESENTACIÓN DIAGRAMA LADDER O DE CONTACTOS</p>

Caso contrario sucede si el peso del arco de entrada a la transición T1 es mayor que el peso del arco de salida, habrá pérdida de marcas (Fig. 3).

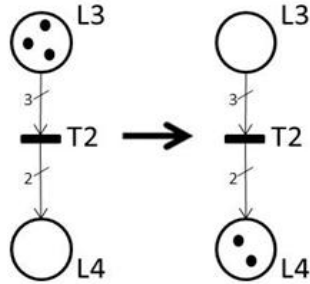


Fig. 3. RdP no binaria que pierde marcas después de su evolución

4.1 Elementos necesarios en la RdP no binaria

En las redes no binarias dos de los elementos presentan algunos cambios en comparación de las binarias, lo que da esta característica de no binaria (Fig. 4).

La cantidad de marcas en los lugares, anteriormente equivalente a una variable booleana (*bool*) y que en esta propuesta se representa por un entero (INT) de 16 bits. Sería posible representar un lugar por una variable de tamaño BYTE, y podría almacenar hasta 255 marcas, pero generalmente los PLCs no poseen comparadores de este tipo de datos. En cambio con el formato entero, un lugar podría almacenar hasta 32767 marcas y para éste existen diferentes tipos de comparadores.

El otro elemento que tiene cambios son los arcos, los cuales incluyen el peso que representa la cantidad de marcas que éstos transportan. A través de operaciones aritméticas se pueden representar estos arcos, realizando suma cuando un arco sea de entrada a un lugar, o resta cuando el arco sea de salida [10].

4.2 Traducción de RdP no binarias a lenguaje Ladder

Los elementos básicos para traducción a lenguaje Ladder y programación de las RdP binarias (Tabla 1), se complementan con las funciones para las RdP no binarias (Tabla 2).

Cada uno de los elementos de la RdP no binaria se puede representar por una función específica en el lenguaje Ladder de programación (Tabla 3). La Tabla 4 muestra la traducción paso a paso de estos elementos, formando así los segmentos necesarios para lograr una programación requerida en un PLC.

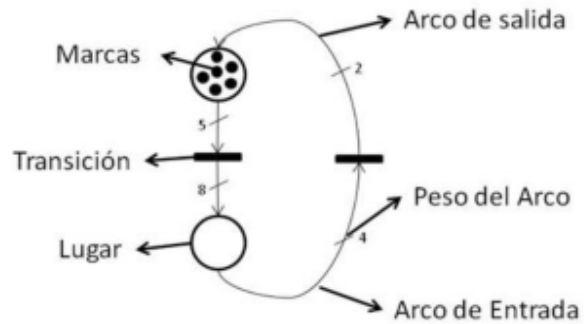

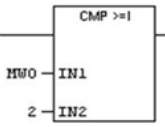

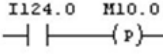

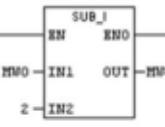
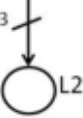



Fig. 4. Elementos de la RdP no binaria.

Tabla 3. Elementos de programación de RdP no binarias

Símbolo	Nombre	Descripción
(P)	Flanco positivo	Permite detectar solo una vez la señal cada vez que hay un cambio de <i>off</i> a <i>on</i> en el código.
ADD_I	Adición	Permite sumar cantidades que representan un valor A y llevarlos a un valor B, almacenándolos en una dirección determinada. Esta función reemplaza el elemento SET en las RdP o binarias.
SUB_I	Sustracción	Permite restar cantidades que representan un valor A y llevarlos a un valor B, almacenándolos en una dirección determinada. Esta función reemplaza el elemento RESET en las RdP no binarias.
CMP > I	Comparación	Permite evaluar el estado anterior de los segmentos en el lenguaje Ladder.

Tabla 4. Traducción de elementos de una RdP no binaria a lenguaje Ladder

RdP no binaria	Traducción lenguaje Ladder	Descripción
		La función comparación equivale a la revisión de la cantidad de marcas disponibles en el lugar analizado en la RdP no binaria. En este ejemplo L1; equivalente a MW0, debe tener mayor número de marcas en IN 1 que en IN 2 para que pueda quedar sensibilizada la transición siguiente.
		Un evento o un conjunto de eventos representados por contactos, temporizadores, entre otros, seguido de un flanco positivo, equivale a la transición de la RdP en lenguaje Ladder. En este ejemplo el evento I124.0 y el flanco positivo M10.0, configuran la transición T1.
		La sustracción es la función que equivale al arco de salida de un lugar y su peso. En este ejemplo el número 2 indica que al disparar la transición, este arco toma dos marcas de MW0 o L1.
		La adición es la función que equivale al arco de entrada al lugar siguiente de la transición y su peso. En este ejemplo el número 3 indica que al disparar la transición, este arco añade tres marcas a MW2 o L2.

5. EJEMPLO DE APLICACIÓN

5.1 Traducción de RdP no binarias a lenguaje Ladder

Una red de semáforos de por lo menos dos cruces puede ser una de las posibles aplicaciones de una

RdP no binaria. La RdP se utiliza en aquellos sistemas en los cuales algunas etapas de su funcionamiento dependen de prioridades. Las prioridades en estos sistemas se pueden representar dando valores enteros a cada uno de los posibles tipos de vehículos que transiten a través de los carriles. Un vehículo más grande tendrá asignado un mayor peso que le permitirá darle mayor prioridad a la vía, reflejándose esto en un mayor tiempo en el color verde [10]. Así, cada lugar de la RdP representa la cantidad de vehículos que hay sobre cada carril. Los arcos de entrada al lugar representan vehículos que llegan al carril, y los arcos de salida los vehículos que salen. El peso de los arcos es entonces el tipo e importancia del vehículo: peso 1 para motocicletas, peso 2 para automóviles y peso 3 para camiones o vehículos pesados (Fig. 5).

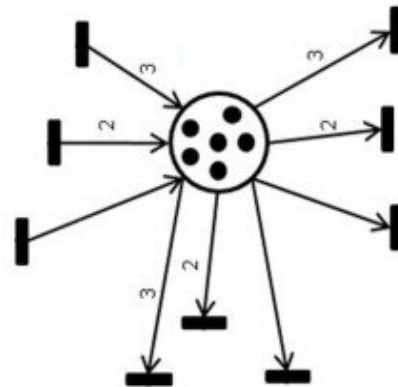


Fig. 5. RdP no binaria para un cruce [10].

5.2 Descripción del sistema

Como condición inicial deben comenzar en rojo los semáforos del carril 1 del cruce 1-2 y 1-3, del carril 2 del cruce 2-4 y del carril 4 del cruce 3-4 y en verde los semáforos del carril 2 del cruce 1-2, del carril 3 del cruce 1-3 y 3-4 y del carril 4 del cruce 2-4 (Fig. 6).

Cada semáforo situado en los carriles de los cruces debe tener un tiempo igual para realizar el cambio de color. Entre los cambios de verde a rojo y viceversa, los semáforos deben pasar por un color amarillo que permanecerá encendido para ambos un tiempo mínimo.

Si a uno de los carriles llega algún tipo de vehículo este tomará un mayor valor, permitiendo el cambio de una manera rápida de rojo a verde, minimizando el tiempo de espera para el carril con más afluencia vehicular.

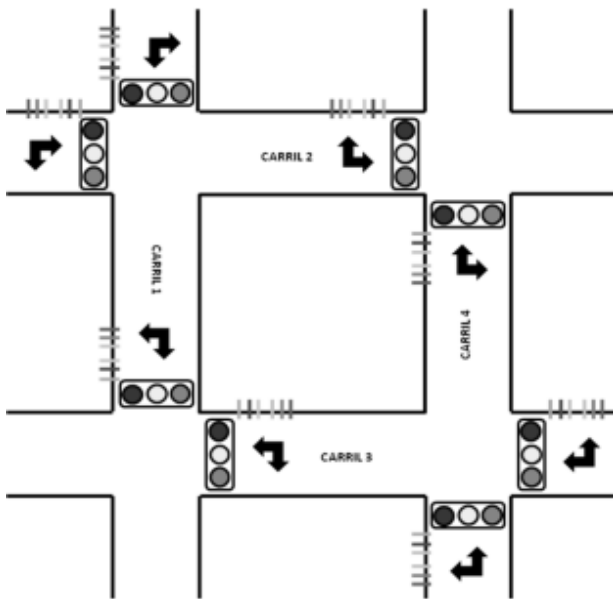


Fig. 6. Semáforo con 4 cruces [10].

Si después de la partida vehicular de los cruces el valor de los carriles es igual, los semáforos deben regresar a sus intervalos de tiempos iniciales.

5.3 Tabla de símbolos

Tabla 5. Aparte de tabla de símbolos de red de semáforos de cuatro cruces

Símbolo	Dirección	Tipo de dato
Amarillo 1 y 2 a verde 1	M 0.1	BOOL
Amarillo 1 y 2 a verde 2	M 0.3	BOOL
Amarillo 1	A 124.1	BOOL
Amarillo 2	A 124.4	BOOL
Cantidad 1	MW 2	INT
Cantidad 2	MW 4	INT
Complete Restart	OB 100	OB 100

5.4 Conversión a diagrama de contactos

El diagrama de contactos (Fig. 7) muestra la regla de cambio de luces en los semáforos, de acuerdo a unas condiciones establecidas por los temporizadores y la cantidad de vehículos presentes en los cruces. Se observa que de no haber ningún tipo de vehículo, el cambio de color en los cruces depende de un T0 de 20 segundos, si existe algún tipo de vehículo en uno de los dos cruces, el programa asignará un valor superior al semáforo que lo controla y dará prioridad a su movilización, cuando el cruce que estaba vacío

contenga una cantidad superior de vehículos que el anterior, se cederá la priorización a este y después de cumplir con un tiempo mínimo T1 de 5 segundos el semáforo que lo controla cambiará para darle vía.

6. CONCLUSIONES

La evolución del marcaje en una RdP depende de la cantidad de marcas existente en los lugares de dicha red. En un lugar el marcaje debe ser superior al peso del arco de salida, si esta condición no se cumple la transición no estará sensibilizada y no habrá evolución del marcaje.

El desarrollo de las RdP no binarias y sus aplicaciones han sido poco trabajadas en comparación con otros tipos de redes, sin embargo son muy útiles para el modelamiento de sistemas complejos y funcionalidades más exigentes.

La metodología propuesta para la programación de redes de Petri no binarias en lenguaje Ladder resulta similar en estructura a otras metodologías existente para las binarias, sin embargo requiere de otro tipo de datos y elementos para programar la evolución del marcaje, como datos analógicos para sus lugares y comparadores y operaciones aritméticas para sus arcos y transiciones.

La programación de RdP no binarias propuesta en la cual los estados se representan por marcas globales tipo entero, permite que desde cualquier programa o bloque del PLC se conozca el estado de la red y así se abran las posibilidades a sistemas de monitoreo de más alto nivel.

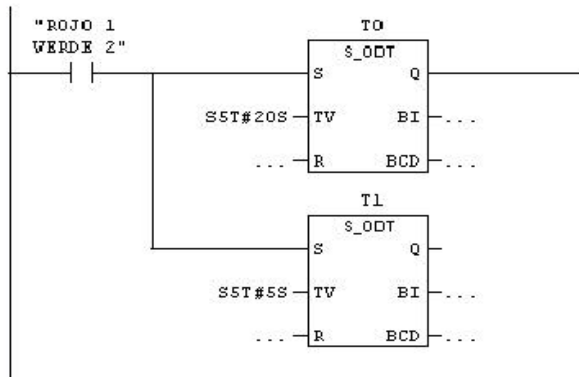
7. AGRADECIMIENTOS

A los profesores Andrés García Higuera de la universidad Castilla de la Mancha (España), James Brusey de Coventry University (Reino Unido) y Murat Uzam de Meliksah University (Turquía), por la disposición que tuvieron al brindar valiosas fuentes de información.

Al Instituto Tecnológico Metropolitano (Institución Universitaria) por el aporte y apoyo a la investigación, a la formación profesional y por la utilización de sus laboratorios, especialmente el de Automatización Industrial para realizar las simulaciones y pruebas requeridas para obtener los resultados.

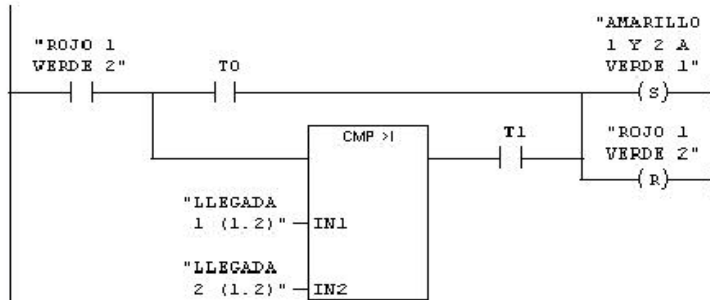
Segm. 1 : SEMAFORO CUATRO CRUCES

COMIENZA EL SISTEMA CON CARRIL 1 EN ROJO Y CARRIL 2 EN VERDE, CRUCE 1-2



Segm. 2 : SEMAFORO CUATRO CRUCES

ACTIVACIÓN DE TEMPORIZADORES PARA CAMBIO DE SEMAFORO EN AMBOS CARRILES A AMARILLO, CRUCE 1-2



Segm. 6 : SEMAFORO CUATRO CRUCES

ACTIVACIÓN DE TEMPORIZADORES PARA CAMBIO DE SEMAFORO EN AMBOS CARRILES A AMARILLO, CRUCE 1-2

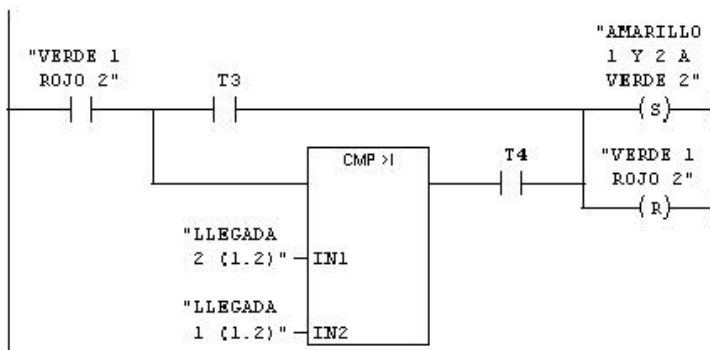


Fig. 7. Aparte de diagrama Ladder de red de semáforos de cuatro cruces [10].

8. REFERENCIAS BIBLIOGRAFICAS

- [1] Ferreira Contreras, J. (s.f.). Instituto Tecnológico de Morelia. Recuperado el 15 de Mayo de 2011, de <http://es.scribd.com/doc/38131473/Redes-Petri>
- [2] Zapata, G. y Carrasco, E. Estructuras generalizadas para controladores lógicos modeladas mediante redes de Petri. Dyna 135, 2002.
- [3] Uzam, M., Jones, A. H. y Ajlouni, N. (1996) Conversion of Petri Net Controllers for Manufacturing Systems into Ladder Logic Diagrams. Emerging Technologies and Factory Automation, 1996. EFTA '96. Proceedings., 1996 IEEE Conference on. Vol 2. 1996
- [4] Kofman, E. (s.f.). Simulación y Control de Sistemas Continuos por Eventos Discretos. Tesis de Doctorado . Rosario, Santa Fe, Argentina.
- [5] Petri, C. A. (1966). Communication with Automata. New York: Griffiss Air Force Base.
- [6] Silva, M. (1985). Las Redes de Petri en la Automática y la Informática. Editorial AC.
- [7] Vásquez Salazar, R. D. (2010). Control Lógico Programable. Medellín: Textos Académicos, Fondo Editorial ITM.
- [8] Escuela Politecnica Nacional de Ecuador. (s.f.). Repositorio Digital Escuela Politecnica Nacional. Recuperado el 10 de Abril de 2011, de <http://bibdigital.epn.edu.ec/>
- [9] García Higuera, A. (2005). El Control Automático en la Industria. Cuenca: Universidad de Castilla - La Mancha.
- [10] Arroyave Díaz, J. A., & Zuleta Bermúdez, M. A. (Junio de 2011). Investigación sobre la utilización de las Redes de Petri no binarias en controladores lógicos programables utilizando lenguaje Ladder. Medellín, Antioquia, Colombia.