



UNAM. Facultad de Ciencias  
Máquinas Digitales con Laboratorio

Profesor: Arturo León Romanos  
Alumno: Mario Tonatiuh Zamarron

## *Reporte final:*

*Diseño de un Semáforo Inteligente con un microcontrolador PICAXE*





*Una vida llena de errores no sólo es más honorable sino que es  
más provechosa que una vida sin haber hecho nada.*

George Bernard Shaw



# Presentación

El presente reporte reúne los temas vistos y actividades realizadas en la materia optativa de Máquinas Digitales con Laboratorio de la Facultad de Ciencias de la UNAM durante el semestre 2010-1.

La primera parte del reporte trata sobre la teoría de los circuitos de conmutación. En la segunda parte se aborda el problema de diseñar un semáforo inteligente (es decir, un semáforo mejorado que pueda tomar decisiones respecto al tráfico) por medio de componentes electrónicos discretos e integrados. En la tercera parte se retoma el problema anterior pero esta vez utilizando un microcontrolador PICAXE.

Quiero agradecer al profesor y M. en I. Arturo León Romanos por su paciencia, entusiasmo y entrega a lo largo del curso.

M. T. Z.



# Contenido

## **1ª parte. Teoría de los circuitos de conmutación (TCC)**

Antecedentes	1
Objetivos de la TCC	1
Relaciones, funciones y operaciones binarias	2
Álgebra booleana	3
Teoremas importantes	3
Álgebra de conmutación	4
Funciones booleanas: Formas SOP y POS	4
Funciones booleanas: formas canónicas	4
Mintérminos	4
Maxtérminos	6
Equivalencia de funciones booleanas al ser expresadas con mintérminos y maxtérminos	7
Métodos de simplificación de funciones booleanas	8
Método del mapa de Karnaugh	8
Método de Quine–McCluskey	8
Problemas resueltos del libro “Introduction to switching circuit theory” de Donald D. Givone	9
Bibliografía y referencias	12

## **2ª parte. Diseño de un semáforo inteligente con componentes electrónicos discretos e integrados**

Definición del problema	13
Sistemas y subsistemas del semáforo inteligente	13
SISTEMA CONTADOR-COMPARADOR	14
Conexión general entre sus distintos elementos (diagrama)	15
Subsistema de Control del contador y del latch	15
Multivibradores	15
Temporizadores	16
Desarrollo	17
Resultados	19
Discusión	20



Conexión entre los Subsistemas Contador, Comparador y de Control del contador y del latch	20
Representación del Subsistema de Control del Contador y del Latch con la conversión de BCD a 7 segmentos	22
<b>SISTEMA DE CONTROL DE LAS LUCES DEL SEMÁFORO</b>	<b>23</b>
Implementación del sistema de control de las luces del semáforo utilizando flip-flops JK	24
Tabla de verdad del Semáforo Inteligente	26
Mapas de Karnaugh del Semáforo Inteligente	28
Ecuaciones obtenidas con los K-Mapas	34
Circuito del Sistema de Control de las Luces del Semáforo Inteligente	35
Bibliografía y referencias	36

### **3ª parte. Diseño del semáforo inteligente con un microcontrolador PICAXE**

Objetivo	37
Introducción	37
Desarrollo	37
1 – Diseño del circuito impreso	37
a) Diseño del diagrama de conexiones	37
b) Preparación de la placa para circuitos impresos	40
c) Impresión láser del circuito en papel satinado	40
d) Planchado de la impresión sobre una placa para circuitos impresos (placa con una capa de cobre)	41
e) Remoción del papel	41
f) Eliminación del cobre sobrante	41
g) Perforación de las superficies de conexión en la placa	42
h) Ensamblado y soldado de los componentes electrónicos	42
2 – Diseño del programa de control del PICAXE-14	42
Programa con Diagramas de Flujo	43
Programa en BASIC	45
3 – Descarga del programa en el PICAXE-14M y prueba de funcionamiento	46

### **APÉNDICE**

Distribución de pines del temporizador 555	47
Distribución de pines del 74143	48
Distribución de pines del comparador 7485	50

## 1ª parte

# Teoría de los circuitos de conmutación (TCC)

La teoría de los circuitos de conmutación (TCC) puede considerarse una rama de una teoría más general denominada teoría de autómatas. Mientras que la teoría de autómatas, de manera global, se encarga del estudio del procesamiento de información entre las terminales de entrada y salida de sistemas, la teoría de los circuitos de conmutación se enfoca en sistemas discretos, sus estructuras y las relaciones entre sus comportamientos y sus estructuras. Un sistema discreto es aquel en el que hay un número contable de valores de señales asociados con las terminales de entrada y salida. Además, un circuito de conmutación consiste de un número finito de nodos, cada uno de los cuales tiene en cada momento sólo un valor de un número contable de valores de señal. Las computadoras, redes de teléfono y sistemas de control de procesos son ejemplos de circuitos de conmutación o sistemas de conmutación.

Es común dividir en dos clases a los circuitos de conmutación. El primer tipo de denomina circuito combinacional. Este tipo de circuito se caracteriza por el hecho de que las salidas en cualquier instante son funciones únicamente de las entradas en ese mismo instante.

El segundo tipo de circuitos de conmutación se denomina circuito secuencial. Estos circuitos se caracterizan por el hecho de que las salidas es cualquier instante no sólo son funciones de las entradas en ese instante, sino también de las entradas en tiempos anteriores. Este tipo de circuitos tienen una propiedad de memoria, y el orden en el que los valores de entrada son aplicados es importante, a diferencia de los circuitos combinacionales.

## Antecedentes

La TCC inició en 1938, con la tesis de maestría de Claude Shannon, *Un análisis simbólico sobre los relevadores y los circuitos de conmutación*, en el Massachusetts Institute of Technology (MIT). Desarrolló un álgebra capaz de describir redes de relevadores. Además, mediante manipulaciones de las expresiones algebraicas asociadas con las redes de relevadores fue posible describir redes equivalentes. De esta forma, el diseño de redes pudo lograrse mediante métodos algebraicos. El álgebra de Shannon se asemejaba al cálculo de proposiciones que había sido derivado del álgebra de la lógica desarrollada por George Boole en 1854. En 1949 Shannon extendió sus ideas tempranas y métodos. Al poco tiempo, el valor de la TCC como un auxiliar en el diseño de redes discretas fue claramente reconocido, y muchos avances y mejoras al álgebra inicial aparecieron en la literatura.

## Objetivos de la TCC

El principal objetivo de la TCC es desarrollar métodos formales para el diseño de sistemas discretos. Esto se logra creando un modelo matemático para un problema físico dado. Una vez que el modelo es establecido, puede ser manipulado de tal manera que se obtenga una forma matemática equivalente. Esta nueva forma, correspondiente a una red de conmutación, debería ser idealmente más simple, más económica, más rápida y más confiable. Si bien, no todos estos objetivos suelen satisfacerse simultáneamente sí se logra una mejoría.

Debe notarse que, una vez que el modelo matemático es obtenido, el diseño implica la manipulación del modelo, y el significado físico del sistema puede excluirse hasta las etapas finales del diseño. Por lo tanto, las técnicas de diseño son aplicables a cualquier sistema que pueda describirse con el mismo sistema algebraico ya que gran parte del diseño se hace en el sistema algebraico. Por ejemplo, en el caso de circuitos de conmutación electrónicos, inicialmente basta con considerar las propiedades lógicas de los componentes y no las propiedades eléctricas. Es por esto que la TCC es de gran valor en el diseño de varios tipos de sistemas físicos tales como las computadoras y las redes de conmutación telefónicas.

Nuestra atención se centra en sistemas en los que sólo existe uno de dos posibles valores discretos de las señales en cualquier punto de la red. Esta restricción es una consecuencia directa del tipo de componentes usados en la construcción de una red. De hecho, se pueden concebir sistemas más generales que permitan un mayor número de valores discretos de señales, pero no son de nuestro interés. Existen muchos componentes que son de naturaleza binaria, y son estos los que se usan principalmente en los circuitos de conmutación.

Debe notarse que, si bien, la TCC se desarrolla con principios matemáticos, estos no son el objetivo principal. El interés fundamental es la síntesis de sistemas discretos utilizando dichos principios. Debido a esto, las técnicas desarrolladas tienen un gran potencial para utilizarse en el diseño de cualquier sistema que satisfaga los mismos requisitos iniciales. En este sentido la TCC deja de ser sólo matemáticas y se convierte en ingeniería.

## **Relaciones, Funciones y Operaciones Binarias**

Cualquier subconjunto  $R$  del conjunto  $S \times T$  es definido como una relación binaria, o simplemente relación, entre los conjuntos  $S$  y  $T$ . El principio detrás de una relación binaria es que  $(s, t)$ , donde  $s \in S$  y  $t \in T$ , esté dentro del conjunto  $R$  si  $s$  y  $t$  están relacionados, usualmente por alguna regla conocida que describe la relación. Por ejemplo, que  $s > t$ ,  $s < t$ ,  $s = 2t$ , etc.

Un concepto muy cercano al de relación es el de función. Una función  $F$ , o mapeo, desde un subconjunto de  $S$  en un conjunto  $T$  es una relación binaria, es decir un subconjunto de  $S \times T$ , que cumple con dos propiedades: (1) la relación binaria no es un conjunto vacío, y (2) si  $(s, t_1)$  y  $(s, t_2)$  son ambos miembros de la relación, entonces  $t_1 = t_2$ .

Una operación binaria  $\circ$  en un conjunto  $S$  es un mapeo del conjunto  $S \times S$  al conjunto  $S$ . Por lo tanto, una operación binaria  $\circ$  en el conjunto  $S$  es una función en la que  $\forall (a, b) \in S \times S$  existe un único elemento  $c \in S \ni c = a \circ b$ .

### *Sistema matemático*

Un sistema matemático  $M$  es un conjunto  $M = \{S, R, P\}$  en el que  $S$  es un conjunto no vacío de elementos,  $R$  es un conjunto de relaciones (usualmente operaciones) en  $S$ , y  $P$  es un conjunto de postulados sobre  $S$  y  $R$ .

El sistema matemático utilizado en los circuitos de conmutación es un álgebra booleana. Este sistema hace uso relaciones y funciones de conjuntos, siendo las operaciones binarias unas de las funciones más importantes.

## Álgebra booleana

Un sistema matemático que consta de un conjunto (B) de elementos junto con dos operaciones cerradas binarias (+) y (•) se dice que es un álgebra booleana si y sólo si cumple con los siguientes postulados  $\forall a, b, c \in B$ :

*P1. Conmutatividad:*

$$\begin{aligned}a + b &= b + a \\a \cdot b &= b \cdot a\end{aligned}$$

*P2. Distributividad:*

$$\begin{aligned}a + (b \cdot c) &= (a + b) \cdot (a + c) \\a \cdot (b + c) &= (a \cdot b) + (a \cdot c)\end{aligned}$$

*P3. Elemento Neutro:*

$$\begin{aligned}a + 0 &= a \\a \cdot 1 &= a\end{aligned}$$

*P4. Elemento Inverso:*

$$\begin{aligned}a + a' &= 1 \\a \cdot a' &= 0\end{aligned}$$

## Teoremas importantes

1. *Principio de dualidad.* Todo teorema o identidad algebraica deducible de los postulados del álgebra booleana es transformable en un segundo teorema o identidad algebraica válidos si todas las operaciones (+) y (•) y los elementos neutros 0 y 1 son intercambiados entre sí.

Derivados de los cuatro postulados y del principio de dualidad existen una serie de teoremas muy importantes, entre ellos:

2.  $a'$  es único.

7.  $a + ab = a$   
 $a(a + b) = a$

3.  $a + 1 = 1$   
 $a \cdot 0 = 0$

8.  $a + a' b = a + b$   
 $a(a' + b) = a + b$

4.  $0' = 1$   
 $1' = 0$

9.  $a + (b + c) = (a + b) + c$   
 $a(bc) = (ab)c$

5.  $a + a = a$   
 $aa = a$

10. Leyes de Morgan:  
 $(a + b + \dots + n)' = a' b' \dots n'$   
 $(ab \dots n)' = a' + b' + \dots + n'$

6.  $(a')' = a$

A partir de estos postulados y principalmente estos teoremas es posible transformar funciones booleanas en otras equivalentes, con igual o distinto número de términos que las originales. Una función equivalente presenta los mismos valores que la función original para iguales estados

lógicos de cada variable en ambas funciones.

## Álgebra de conmutación

Hasta este punto se ha considerado un sistema matemático llamado álgebra booleana. Sin embargo, no se han definido las dos operaciones binarias postuladas. Con el fin de establecer una relación entre un álgebra booleana y los circuitos de conmutación, se presentará una forma particular de álgebra booleana definiendo las operaciones binarias.

*Teorema.* El conjunto  $B = \{0, 1\}$  junto con las operaciones por  $(+)$ ,  $(\bullet)$  y  $(')$  definidas por

$+$	$0$	$1$	$\bullet$	$0$	$1$	
$0$	$0$	$1$	$0$	$0$	$0$	$0' = 1$
$1$	$1$	$1$	$1$	$0$	$1$	$1' = 0$

es un álgebra booleana.

## Funciones booleanas: Formas SOP y POS

Las funciones de conmutación en la forma de suma de productos (SOP) se construyen al sumar (OR) términos producto (AND), donde cada término producto se forma mediante el AND de varias variables complementadas o sin complementar, cada una de las cuales es una literal. Un ejemplo de forma SOP de una función de cuatro variables es

$$f(A, B, C, D) = \overline{A}\overline{B}C + \overline{B}\overline{D} + \overline{A}C\overline{D}$$

Las funciones de conmutación en la forma de producto de sumas (POS) se construyen al considerar el producto (AND) de términos suma (OR), donde cada término suma se obtiene mediante el OR de varias literales. Un ejemplo de la forma POS de una función de cuatro variables es

$$f(A, B, C, D) = (A + \overline{B} + C)(\overline{B} + \overline{D})(\overline{A} + C + \overline{D})$$

## Formas canónicas

Las formas canónicas de las funciones de conmutación son ciertas formas SOP y POS con características especiales. Si bien, es posible representar funciones de conmutación mediante expresiones de conmutación diferentes pero equivalentes, las formas canónicas SOP y POS son únicas para cada función (i.e., una función no puede tener dos o más formas canónicas SOP o POS).

**Mintérminos.** Para una función de  $n$  variables, si un término producto contiene cada una de las  $n$  variables exactamente una vez, ya sea en forma complementada o no complementada, el término producto es un *mintérmino*. Si una función se representa como una suma sólo de mintérminos, decimos que la función tiene la forma de *suma canónica de productos (SOP canónica)*. Por ejemplo,

$$f_{\alpha}(A, B, C) = \overline{A}\overline{B}\overline{C} + A\overline{B}\overline{C} + \overline{A}B\overline{C} + ABC \quad (1)$$

es una función en la forma SOP canónica, con cuatro mintérminos.

Para simplificar la escritura de la forma canónica SOP. Con frecuencia se usa una notación especial, en la que cada mintérmino se representa mediante un código binario de  $n$  bits. Cada bit representa una de las variables del mintérmino como sigue: 1 si la variable es no complementada y 0 si sí es complementada. Las variables se enumeran en el mismo orden en cada mintérmino. Lo importante de esta notación es que, para que un mintérmino valga 1, cada variable no complementada del mintérmino debe valer 1 y cada variable complementada debe valer 0. Con este código, podemos escribir los mintérminos de la ecuación (1) en una de las siguientes formas equivalentes:

Mintérmino	Código del mintérmino	Número del mintérmino
$\overline{A}\overline{B}\overline{C}$	010	$m_2$
$A\overline{B}\overline{C}$	110	$m_6$
$\overline{A}B\overline{C}$	011	$m_3$
$ABC$	111	$m_7$

Escribimos cada mintérmino en forma abreviada como  $m_i$ , donde  $i$  es el entero decimal igual al código binario correspondiente para el mintérmino. Así, podemos escribir la ecuación (1) en forma compacta como

$$f_{\alpha}(A, B, C) = m_2 + m_3 + m_6 + m_7 \quad (2)$$

Podemos simplificar aún más si escribimos la función en forma de lista de mintérminos como sigue:

$$f_{\alpha}(A, B, C) = \sum m(2, 3, 6, 7) \quad (3)$$

Las tres ecuaciones (1), (2) y (3) ilustran tres formas diferentes, pero equivalentes de representar la forma canónica SOP para  $f_{\alpha}(A, B, C)$ .

El orden de las variables en la notación funcional de las ecuaciones (2) y (3) es muy importante, ya que determina el orden de los bits en los números del mintérmino. Podemos demostrar este hecho fácilmente, cambiando la relación de orden de las variables en la función  $f_{\alpha}(A, B, C)$  a  $f_{\alpha}(B, C, A)$  como sigue:

$$\begin{aligned} f_{\alpha}(B, C, A) &= \sum m(2, 3, 6, 7) \\ &= m_2 + m_3 + m_6 + m_7 \\ &= 010 + 011 + 110 + 111 \end{aligned}$$

$$\begin{aligned}
&= \overline{B}C\overline{A} + \overline{B}CA + B\overline{C}\overline{A} + BCA \\
&= \overline{A}\overline{B}C + \overline{A}BC + A\overline{B}C + ABC \\
&\neq f_{\alpha}(A, B, C)
\end{aligned} \tag{4}$$

Se ve claramente que las ecuaciones (1) y (4) son distintas.

Podemos deducir fácilmente la tabla de verdad de una función a partir de su forma SOP canónica. Esto se hace considerando lo siguiente: cada fila se numera según su código decimal, y los únicos unos que aparecen en la tabla corresponden a las filas  $i$ , corresponden a los mintérminos  $m_i$ . Obsérvese la siguiente tabla de verdad:

Fila núm. (i)	Entradas A B C	Salidas $f_{\alpha}(A, B, C) = \sum m(2, 3, 6, 7)$	Complemento $\overline{f_{\alpha}}(A, B, C) = \sum m(0, 1, 4, 5)$
0	0 0 0	0	1 $\leftarrow m_0$
1	0 0 1	0	1 $\leftarrow m_1$
2	0 1 0	1 $\leftarrow m_2$	0
3	0 1 1	1 $\leftarrow m_3$	0
4	1 0 0	0	1 $\leftarrow m_4$
5	1 0 1	0	1 $\leftarrow m_5$
6	1 1 0	1 $\leftarrow m_6$	0
7	1 1 1	1 $\leftarrow m_7$	0

Obsérvese que todos los mintérminos compuestos por tres variables ( $8 = 2^3$ ) están en la lista de mintérminos para  $f_{\alpha}(A, B, C)$ .

La lista de mintérminos para  $f_{\alpha}(A, B, C)$  más la de  $\overline{f_{\alpha}}(A, B, C)$  da en total ocho mintérminos que son todos los mintérminos posibles que se pueden obtener con tres variables ( $2^3$ ). En general, cada uno de los  $2^n$  mintérminos de  $n$  variables siempre aparecerá en la forma SOP canónica para  $f(x_1, x_2, \dots, x_n)$  más la de  $\overline{f}(x_1, x_2, \dots, x_n)$ .

**Maxtérminos.** Si un término suma de una función de  $n$  variables contiene una de las  $n$  variables exactamente una vez en forma complementada o no complementada, el término suma es un maxtérmino. Si representamos una función como producto de términos suma, cada uno de los cuales es un maxtérmino, decimos que la función tiene la forma canónica de producto de sumas (POS canónica). Por ejemplo,

$$f_{\gamma}(A, B, C) = (A + B + C)(A + B + \overline{C})(\overline{A} + B + C)(\overline{A} + B + \overline{C}) \tag{5}$$

Es la forma POS canónica de una función  $f_{\gamma}(A, B, C)$  que tiene cuatro maxtérminos.

Adoptamos una notación especial para los maxtérminos, como hicimos con los mintérminos, con la diferencia de que el código se intercambia como sigue: 0 si la variable es no complementada y 1 si sí lo es. Lo importante de esta notación es que, para que un maxtérmino valga 0, toda variable no complementada del maxtérmino debe valer 0 y toda variable complementada debe valer 1. Así, podemos representar los maxtérminos de  $f_{\gamma}(A, B, C)$  como sigue:

Maxtérmino	Código del maxtérmino	Número del maxtérmino
$A + B + C$	000	$M_0$
$A + B + \overline{C}$	001	$M_1$
$\overline{A} + B + C$	100	$M_4$
$\overline{A} + B + \overline{C}$	101	$M_5$

Escribimos cada maxtérmino en forma abreviada como  $M_i$ , donde  $i$  es el entero decimal del código binario correspondiente para el maxtérmino. Así,

$$f_\gamma(A, B, C) = M_0 M_1 M_4 M_5 \quad (6)$$

$$= \prod M(0, 1, 4, 5) \quad (7)$$

La última forma es la forma de lista de maxtérminos. Las ecuaciones (5), (6) y (7) son formas POS canónicas equivalentes para  $f_\gamma(A, B, C)$ . Como en el caso de las ecuaciones (2) y (3), el orden de las variables en las ecuaciones (6) y (7) es muy importante. La tabla de verdad para  $f_\gamma(A, B, C)$  es

Salida Máx. núm. $f_\gamma(A, B, C)$	Entradas A B C					
0	0 0 0	0	1	1	1	0
1	0 0 1	1	0	1	1	0
2	0 1 0	1	1	1	1	1
3	0 1 1	1	1	1	1	1
4	1 0 0	1	1	0	1	0
5	1 0 1	1	1	1	0	0
6	1 1 0	1	1	1	1	1
7	1 1 1	1	1	1	1	1

Cada fila de la tabla se numera según el código decimal, como lo hicimos con los mintérminos. Obsérvese que los únicos ceros de la tabla aparecen en los renglones  $i$  correspondientes a los maxtérminos  $M_i$ . Por lo tanto, como hicimos con los mintérminos, podemos generar la tabla de verdad mediante una inspección directa de la lista de maxtérminos. Al comparar las tablas de verdad para  $f_\alpha(B, C, A)$  y  $f_\gamma(A, B, C)$  vemos que

$$\begin{aligned} f_\alpha(B, C, A) &= \sum m(2, 3, 6, 7) \\ &= f_\gamma(B, C, A) \\ &= \prod M(0, 1, 4, 5) \end{aligned} \quad (8)$$

Por lo tanto, las funciones  $f_\alpha(B, C, A)$  y  $f_\gamma(A, B, C)$  son iguales y la ecuación (8) muestra sus formas SOP y POS canónicas.

## Métodos de simplificación de funciones booleanas



## Método del mapa de Karnaugh

El método del mapa presenta un procedimiento simple y directo para minimizar funciones booleanas. Este método fue propuesto primero por Edward W. Veitch en 1952 y modificado ligeramente por Maurice Karnaugh al año siguiente, se conoce como el “diagrama de Veitch” o el “mapa de Karnaugh” (abreviado como K-Mapa o KV-Mapa).

Un mapa de Karnaugh es un diagrama utilizado para la minimización de funciones algebraicas booleanas.

Los mapas K aprovechan la capacidad del cerebro humano de trabajar mejor con patrones que con ecuaciones y otras formas de expresión analítica. Externamente, un mapa de Karnaugh consiste de una serie de cuadrados, cada uno de los cuales representa una línea de la tabla de verdad. Puesto que la tabla de verdad de una función de  $N$  variables posee  $2N$  filas, el mapa K correspondiente debe poseer también  $2N$  cuadrados. Cada cuadrado alberga un 0 ó un 1, dependiendo del valor que toma la función en cada fila. Las tablas de Karnaugh se pueden utilizar para funciones de hasta 6 variables, más allá, ya no es tan práctico.

Por ejemplo, considérese la siguiente función de 4 variables:

$$f = \bar{A}\bar{B}CD + \bar{A}B\bar{C}D + \bar{A}BC\bar{D} + \bar{A}BCD + A\bar{B}\bar{C}\bar{D} + A\bar{B}C\bar{D} + AB\bar{C}\bar{D} + ABC\bar{D}$$

Su K-mapa es:

AB \ CD		00	01	11	10
00				1	1
01			1	1	
11	1	1			
10		1	1	1	

La función minimizada con el K-mapa queda

$$f(A,B,C,D) = B\bar{C}D + A\bar{D} + \bar{A}CD + B\bar{C}\bar{D}$$

## Método de Quine–McCluskey

El Algoritmo Quine–McCluskey es un método de simplificación de funciones booleanas desarrollado por Willard Van Orman Quine y Edward J. McCluskey. Es funcionalmente idéntico a la utilización del mapa de Karnaugh, pero su forma tabular lo hace más eficiente para su implementación en lenguajes computacionales, y provee un método determinístico de conseguir la mínima expresión de una función booleana.

El método consta de dos pasos:

1. Encontrar todos los implicantes primos de la función.
2. Usar esos implicantes en una tabla de implicantes primos para encontrar los implicantes primos esenciales, los cuales son necesarios y suficientes para generar la función.

Aunque es más práctico que el mapa de Karnaugh, cuando se trata de trabajar con más de cuatro variables, el tiempo de resolución del algoritmo Quine-McCluskey crece de forma exponencial con el aumento del número de variables. Se puede demostrar que para una función de  $n$  variables el límite superior del número de implicantes primos es  $3n/n$ . Si  $n = 32$  habrá más de  $6.5 \times 10^{15}$  implicantes primos. Funciones con un número grande de variables tienen que ser minimizadas con otros métodos heurísticos.

## Problemas resueltos del libro “Introduction to switching circuit theory” de Donald D. Givone.

**2.9** Complementar cada una de las siguientes expresiones booleanas:

(a)  $(a'bc + d)'f + g$

$$[(a'bc + d)'f + g]' = \{[(a'bc + d)']' + f'\} g' = [(a'bc + d) + f'] g' = (a'bc + d + f') g'$$

(b)  $\{[ab(c' + d)] + e'\}fg'$

$$\begin{aligned} \{[ab(c' + d)] + e'\}fg' &= \{[ab(c' + d)] + e'\}' + f' + (g')' = \{[ab(c' + d)]' (e')'\} + f' + g \\ &= \{[a' + b' + (c' + d)'] e\} + f' + g = \{[a' + b' + (c')' d'] e\} + f' + g \\ &= (a' + b' + cd') e + f' + g \end{aligned}$$

(c)  $(a + b' + c)(d' + e')' + f'g + h'i'$

$$\begin{aligned} [(a + b' + c)(d' + e')' + f'g + h'i']' &= \{[(a + b' + c)' + [(d' + e')']'][(f')' + g'][(h')'(i')']\} \\ &= \{[(a + b' + c)' + (d' + e')]\}(f + g')(h + i) = (a' b c' + d' + e')(f + g')(h + i) \end{aligned}$$

**2.19** Encontrar la representación canónica disyuntiva (SOP canónica) de la siguiente función definida por la siguiente tabla de combinaciones

<i>a</i>	<i>b</i>	<i>c</i>	<i>f</i>
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

$$f(a, b, c) = a'b'c + a'bc' + a'bc + ab'c$$

**2.21** Escribe cada una de las siguientes expresiones canónicas disyuntivas en forma algebraica y determina sus tablas de combinaciones correspondientes.

(a)  $f(x, y, z) = \sum m(0, 2, 4, 5, 7)$

$x$	$y$	$z$	$f$
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

$$f(x, y, z) = xy'z' + xy'z + xy'z' + xyz + xyz$$

(b)  $f(w, x, y, z) = \sum m(1, 3, 7, 8, 9, 14, 15)$

$w$	$x$	$y$	$z$	$f$
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

$$f(w, x, y, z) = w'x'y'z + w'x'yz + w'xyz + wx'y'z' + wx'y'z + wxyz' + wxyz$$

**2.23** Encontrar la representación canónica conjuntiva (POS canónica) de la función definida por la tabla de combinaciones del problema 2.19.

$$f(a, b, c) = (a + b + c) (a' + b + c) (a' + b' + c) (a' + b' + c')$$

**2.28** Expresar el complemento de las siguientes funciones de conmutación en sus formas canónicas disyuntiva y conjuntiva.

	w	x	y	z
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

**(a)**  $f(x, y, z) = \sum m(0, 1, 6)$

$$f(x, y, z) = m_0 + m_1 + m_6$$

$$\begin{aligned} \Rightarrow f'(x, y, z) &= m_0' m_1' m_6' = (x'y'z')' (x'y'z)' (xyz')' \\ &= (x + y + z)(x + y + z')(x' + y' + z) \\ &= M_0 M_1 M_6 = \prod M(0, 1, 6) \end{aligned}$$

Forma canónica conjuntiva

Pero también otra forma de expresar el complemento es la siguiente

$$\begin{aligned} f'(x, y, z) &= \sum m(2, 3, 4, 5, 7) = m_2 + m_3 + m_4 + m_5 + m_7 \\ &= x'yz' + x'yz + xy'z' + xy'z + xyz \end{aligned}$$

Forma canónica disyuntiva

**(b)**  $f(x, y, z) = \prod M(1, 2, 3, 5, 7)$

$$f(x, y, z) = M_1 M_2 M_3 M_5 M_7$$

$$\begin{aligned} \Rightarrow f'(x, y, z) &= M_1' + M_2' + M_3' + M_5' + M_7' \\ &= (x + y + z')' + (x + y' + z)' + (x + y' + z')' + (x' + y + z')' + (x' + y' + z')' \\ &= x'y'z + x'yz' + x'yz + xy'z + xyz \\ &= m_1 + m_2 + m_3 + m_5 + m_7 = \sum m(1, 2, 3, 5, 7) \end{aligned}$$

Forma canónica disyuntiva

Pero también

$$\begin{aligned} f'(x, y, z) &= \prod M(0, 4, 6) = M_0 M_4 M_6 \\ &= (x + y + z)(x' + y + z)(x' + y' + z) \end{aligned}$$

Forma canónica conjuntiva

**(c)**  $f(w, x, y, z) = \sum m(1, 4, 5, 6, 9, 13, 14)$

$$\begin{aligned}
f(x,y,z,w) &= m_1 + m_4 + m_5 + m_6 + m_9 + m_{13} + m_{14} \\
\Rightarrow f'(x,y,z,w) &= m_1' m_4' m_5' m_6' m_9' m_{13}' m_{14}' \\
&= (w'x'y'z')(w'xy'z')(w'xyz')(wx'y'z')(wxy'z')(wxyz') \\
&= (w+x+y+z')(w+x'+y+z)(w+x'+y+z')(w+x'+y'+z)(w'+x+y+z') \\
&\quad (w'+x'+y+z')(w'+x'+y'+z)
\end{aligned}$$

Forma canónica conjuntiva

Y también

$$\begin{aligned}
f'(x,y,z,w) &= \sum m(0,2,3,7,8,10,11,12,15) \\
&= m_0 + m_2 + m_3 + m_7 + m_8 + m_{10} + m_{11} + m_{12} + m_{15} \\
&= w'x'y'z' + w'x'yz' + w'xyz + w'xyz + wx'y'z' + wx'yz' + wx'yz \\
&\quad + wxyz' + wxyz
\end{aligned}$$

Forma canónica disyuntiva

$$(d) f(w,x,y,z) = \prod M(2,3,8,10,15)$$

$$\begin{aligned}
f(w,x,y,z) &= M_2 M_3 M_8 M_{10} M_{15} \\
\Rightarrow f'(w,x,y,z) &= M_2' + M_3' + M_8' + M_{10}' + M_{15}' = m_2 + m_3 + m_8 + m_{10} + m_{15} \\
&= w'x'yz' + w'x'yz + wx'y'z' + wx'yz' + wxyz \\
&= \sum m(2,3,8,10,15)
\end{aligned}$$

Forma canónica disyuntiva

Y también

$$\begin{aligned}
f'(w,x,y,z) &= \prod M(0,1,4,5,6,7,9,11,12,13,14) = M_0 M_1 M_4 M_5 M_6 M_7 M_9 M_{11} M_{12} M_{13} M_{14} \\
&= (w+x+y+z)(w+x+y+z')(w+x'+y+z)(w+x'+y+z')(w+x'+y'+z) \\
&\quad (w+x'+y'+z')(w'+x+y+z')(w'+x+y'+z')(w'+x'+y+z) \\
&\quad (w'+x'+y+z')(w'+x'+y'+z)
\end{aligned}$$

Forma canónica conjuntiva

## BIBLIOGRAFÍA Y REFERENCIAS

CONSTANTINI, S. *Mapas de Karnaugh*. [En línea]. 2001. Disponible en: <http://medusa.unimet.edu.ve/sistemas/bpis03/mdkminimizacion.htm> [Consultado el 8 de octubre de 2009].

GIVONE, D. *Introduction to switching circuit theory*. New York: McGraw-Hill Computer Science Series, 1970.

NELSON, V.; NAGLE H. *Análisis y diseño de circuitos lógicos digitales*. Prentice Hall, 1997.

Wikipedia, la enciclopedia libre. *Algoritmo Quine–McCluskey*. [En línea]. 24 de julio de 2009. Disponible en: [http://es.wikipedia.org/wiki/Algoritmo\\_Quine–McCluskey](http://es.wikipedia.org/wiki/Algoritmo_Quine–McCluskey) [Consultado el 10 de octubre de 2009].

Wikipedia, la enciclopedia libre. *Mapa de Karnaugh*. [En línea]. 6 de octubre de 2009. Disponible en: [http://es.wikipedia.org/wiki/Mapa\\_de\\_Karnaugh](http://es.wikipedia.org/wiki/Mapa_de_Karnaugh) [Consultado el 8 de octubre de 2009].

## 2ª parte

# Diseño de un semáforo inteligente con componentes electrónicos discretos e integrados

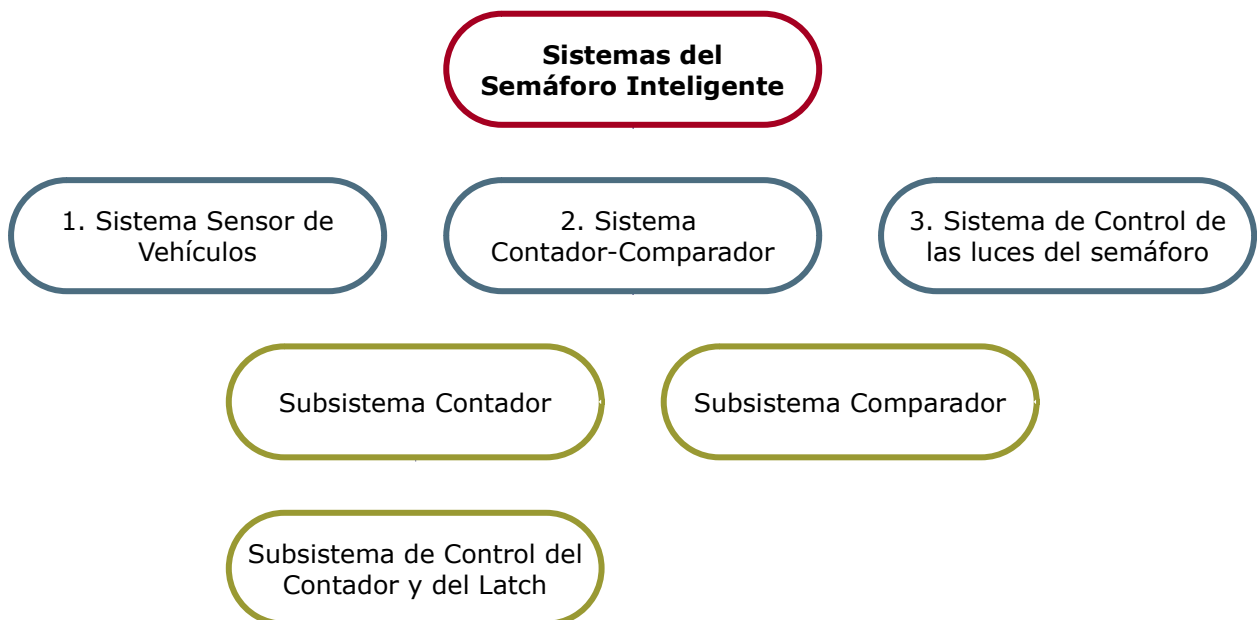
### Definición del problema

El problema consiste en diseñar un semáforo que pueda tomar decisiones respecto al tráfico que se presente, utilizando sensores que medirán la cantidad de vehículos que pasan cada cierto tiempo. Para una primera aproximación se considera lo siguiente:

- El semáforo opera en un crucero en el que las dos calles (A y B) que cruzan son de un solo sentido.
- Las calles son de un solo carril.
- No hay peatones que crucen las calles.

### Sistemas y subsistemas del semáforo inteligente

El sistema completo consta de 3 sistemas y 2 subsistemas; su estructura se muestra en el siguiente organigrama:



### Descripción

1. **Sistema Sensor de Vehículos.** Su objetivo es detectar cada vehículo que circula en las

calles A y B en un cierto tiempo. Se busca que genere un pulso por cada vehículo detectado. Este será el único sistema que no se desarrollará durante el curso, por lo que se mencionan dos ideas sobre cómo podría ser: Puede ser un sensor que reaccione al ser presionado cuando los vehículos pasen sobre él; su forma sería alargada y se instalaría sobre el asfalto como si fuera un tope pero de altura muy pequeña. Otra idea, aunque tal vez más complicada, es que el sensor funcionara con cámaras de video y que a partir de las imágenes identificara al número de vehículos circulantes.

2. **Sistema Contador-Comparador.** Cuenta el número de pulsos -correspondientes a cada vehículo en A y B- mandados por el Sistema Sensor de Vehículos. A su vez compara estas cantidades decidiendo si  $A > B$ ,  $A = B$  o  $A < B$ \*. El sistema se conforma por los siguientes subsistemas:

***Subsistema Contador.*** Es el que se encarga de contar los vehículos (pulsos) en A y B. Está conformado, principalmente, por un contador y una memoria (latch).

***Subsistema de Control del Contador y del Latch.*** Es el que se encarga de controlar al Subsistema Contador.

***Subsistema Comparador.*** Es el que se encarga de comparar el número de vehículos en A y B.

3. **Sistema de Control de las luces del semáforo.** Decide cuándo y en qué orden encender las luces del semáforo según la información dada por el Subsistema Comparador.

## 2. SISTEMA CONTADOR-COMPARADOR

Para diseñar este sistema se necesitarán básicamente un contador, una memoria (latch) que guarde durante un tiempo el número contado de vehículos en A y B, y un comparador. Para esto se utilizarán dos circuitos integrados 74143, un 555, un 74123 y un 7485.

Como ya se mencionó este sistema se compone de dos subsistemas. El subsistema Contador se forma con un 555, un 74123, dos 74143 (que incluyen cada uno un contador de 4 bits, latch y un decodificador BCD a siete segmentos) y dos displays. El 74123 y el 555 trabajarán mutuamente para activar el contador y el latch, respectivamente.

El subsistema Comparador lo constituye únicamente un 7485 que se encargará de comparar las cantidades A y B proporcionadas por los dos 74143.

La figura 1 ilustra de manera general la conexión entre los distintos elementos.

\* El número de vehículos que circulan en cada calle se denota igual que la notación de las calles.

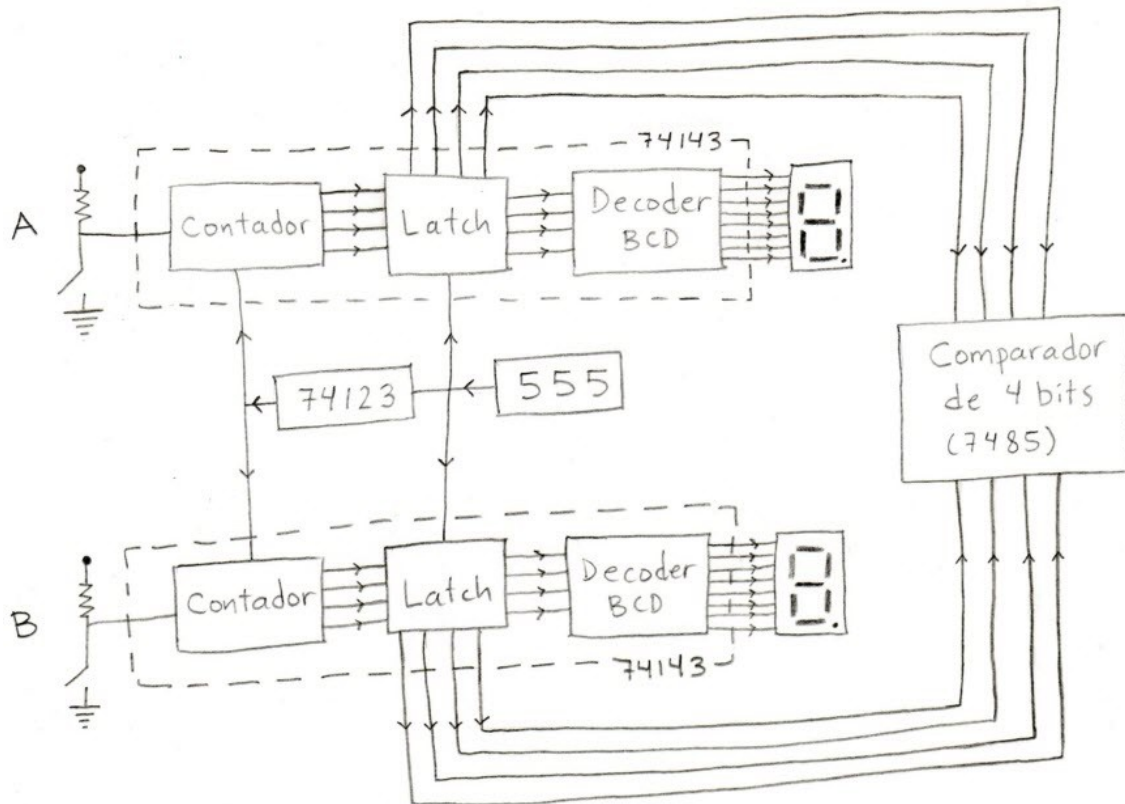


Figura 1. Sistema Contador-Comparador.

## ***SUBSISTEMA DE CONTROL DEL CONTADOR Y DEL LATCH***

### ***Multivibradores***

En electrónica, un multivibrador es un circuito oscilador capaz de generar una onda cuadrada. Según su funcionamiento, los multivibradores se pueden dividir en dos clases:

- De funcionamiento continuo, astable o de oscilación libre: genera ondas a partir de la propia fuente de alimentación.
- De funcionamiento impulsado: a partir de una señal de disparo o impulso sale de su estado de reposo.

Si posee dos de dichos estados, se denomina biestable; si posee uno, se le llama monoestable.

### ***Monoestable***

El monoestable es un circuito multivibrador que realiza una función secuencial consistente en que al recibir una excitación exterior, cambia de estado y se mantiene en él durante un periodo que viene determinado por una constante de tiempo. Transcurrido dicho periodo de tiempo, la salida



del monoestable vuelve a su estado original. Por tanto, tiene un estado estable (de aquí su nombre) y un estado casi estable.

## **Monoestables integrados**

Se encuentran monoestables integrados en varias familias lógicas, tanto TTL (9601, 74121, 74123 y otros) como CMOS (4047, 4528, etc.). Son circuitos que comprenden parte analógica, que es la generación del pulso, y parte digital, que proporciona varias funciones lógicas entre las entradas y las salidas digitales. La duración del pulso es función de  $R \cdot C$ , aunque la dependencia exacta depende del modelo.

La parte digital les añade distintas prestaciones, produciendo diversos tipos de monoestables:

- \* *Restaurable o resetable*: Una entrada de reset permite interrumpir el pulso en cualquier momento, dejando el dispositivo preparado para un nuevo disparo.

- \* *Redisparable (retriggerable)*: Permite reiniciar el pulso con un nuevo disparo antes de completar la temporización. Digamos que se tiene un temporizador de 4 ms, pero a los 2 ms de iniciado el pulso se realiza un nuevo disparo; la duración que se obtiene es de  $2 + 4 = 6$  ms. Los monoestables *no redisparables* sólo permiten el disparo cuando no existe ninguna temporización en curso. Es decir, en el ejemplo anterior ignoraría el segundo disparo y se obtendría un pulso de 4 ms solamente.

- \* *Monoestable-Multivibrador*: Son monoestables dobles (Dos, normalmente independientes) en la misma cápsula que permiten su conexión de forma que el fin del pulso generado por uno de ellos dispara al otro. Permiten el control preciso e independiente de los tiempos alto y bajo de la señal de salida.

Además de los circuitos anteriores, existen circuitos con una parte digital muy reducida, que se pueden utilizar bien como monoestables o como multivibradores y existen tanto en tecnología bipolar como CMOS. El NE555 es el paradigma de este tipo de circuitos.

## **Temporizadores**

Un temporizador es un aparato mediante el cual, podemos regular la conexión ó desconexión de un circuito eléctrico pasado un tiempo desde que se le dio dicha orden. Los temporizadores se pueden clasificar en: Térmicos, neumáticos, de motor síncrono y electrónicos.

El principio básico de funcionamiento de los temporizadores electrónicos se basa en la carga o descarga de un condensador (C) mediante una resistencia (R), en lo que se conoce como circuitos RC.

## **Temporizador 555**

El NE555 (también el LM555, CA555 y MCI555) es el circuito integrado de temporización más comúnmente usado. Este circuito puede funcionar en dos modos: monoestable (un estado estable) o astable (sin estados estables). En modo monoestable produce retardos de tiempo muy precisos que van desde microsegundos a horas. En modo astable produce señales rectangulares con ciclos de trabajo variables.

## Funcionamiento en astable

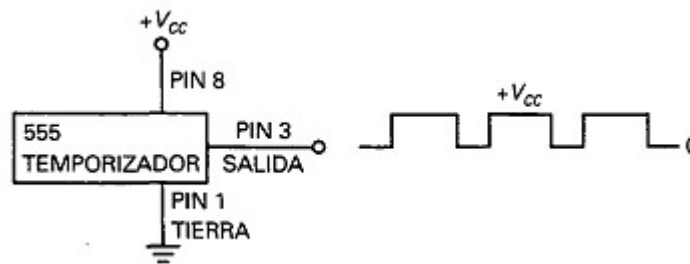


Figura 2. Temporizador 555 funcionando en modo astable (oscilación libre).

El temporizador 555 puede ser conectado para funcionar como un multivibrador astable. Cuando se usa con esta configuración no posee ningún estado estable en el que pueda permanecer indefinidamente. Dicho de otra manera, cuando funciona en modo astable oscilará produciendo un pulso rectangular de salida. La figura 2 muestra un temporizador 555 funcionando en modo astable. Como se observa, la salida es una serie de pulsos rectangulares. Como no es necesaria una señal de disparo para conseguir esta salida, este modo de funcionamiento es denominado también multivibrador de oscilación libre.

## DESARROLLO

Configurar el 555 en astable sin el capacitor de  $0.01 \mu\text{F}$  (es decir, la entrada 5 directamente a tierra), ver figura 3 o revisar la hoja de datos.

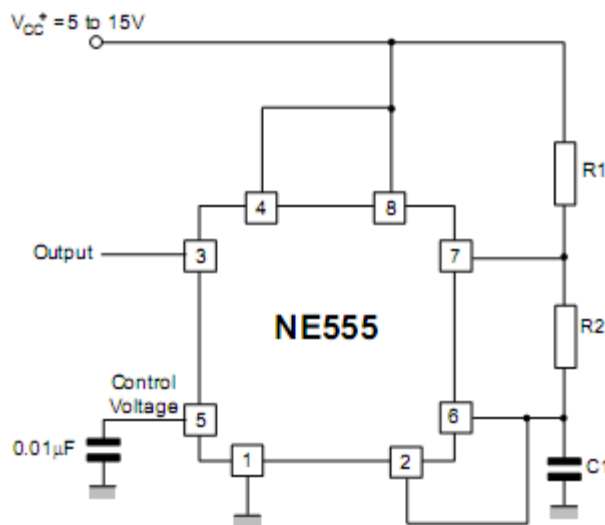


Figura 3. Configuración de un NE555 en astable.

La duración  $t_1$  en estado alto (tiempo de carga del capacitor  $C_1$ ) es:

$$t_1 = 0.693 (R_1 + R_2)C_1 \quad (9)$$

La duración  $t_2$  en estado bajo (tiempo de descarga de  $C_1$ ) es:

$$t_2 = 0.693 (R_2)C_1 \quad (10)$$

donde  $R_1$  y  $R_2$  son los valores de dos resistencias distintas y  $C_1$  el valor del capacitor.

Obsérvese que el tiempo en nivel bajo ( $t_2$ ) sólo depende de una resistencia ( $R_2$ ) mientras que el tiempo en nivel alto  $t_2$  depende de las dos resistencias  $R_1$  y  $R_2$ . A nosotros nos interesa que  $t_1$  sea muy pequeño (del orden de  $\mu s$  o  $ns$ ) con un valor fijo desde el principio y que  $t_2$  sea grande (alrededor de 30 s) pero en un principio queremos que sólo dure aproximadamente 4 segundos para ver que el circuito funcione correctamente. Para esto se tomaron los siguientes valores:  $R_1 = 47 \text{ k}\Omega$ ,  $R_2 = 10 \text{ M}\Omega$ ,  $C_1 = 47 \times 10^4 \text{ pF}$ .

Una vez armado el circuito con el 555 conectarlo al 74123 como en la figura 4. La salida del 555 se conecta en el pin 3 (clear) del 74123. La salida que nos interesa en el 74123 es la del pin 4 ( $\bar{Q}_1$ ). Conectar un osciloscopio de dos canales al pin 3 del 555 y al pin 4 del 74123 para visualizar las señales de entrada y salida –se espera obtener señales semejantes a las de la figura 5-. Verificar que el osciloscopio opere en modo CC.

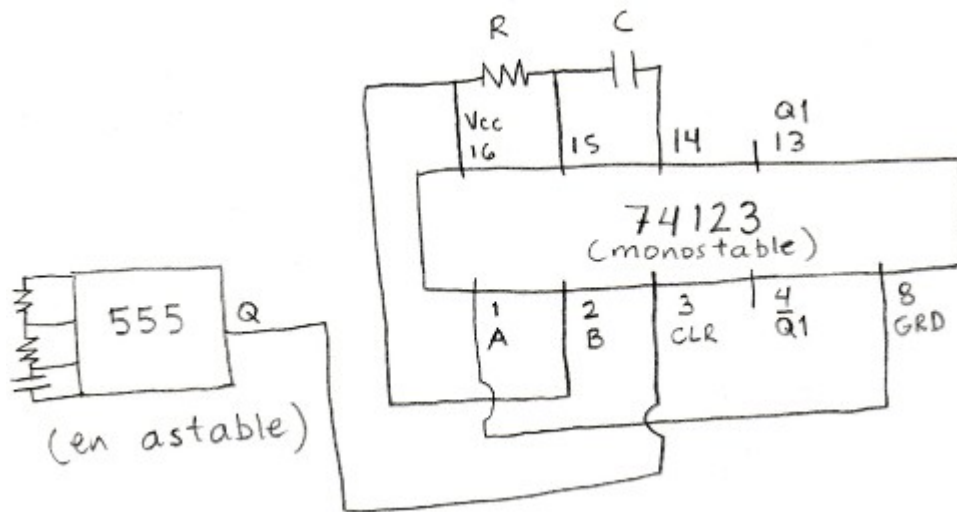


Figura 4. Subsistema de control del contador y del latch: Conexión entre el 555 y el 74123.

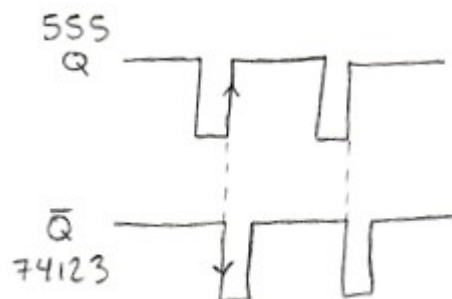


Figura 5. Señales esperadas del subsistema de control del contador y del latch. La señal del 74123 depende del 555.

Para comprender por qué la señal de salida debe ser como en la figura 5 obsérvese en la tabla de verdad del 74123 (ver figura 6 o la hoja de datos) el último renglón. **A** está a tierra y por lo tanto siempre está en nivel de voltaje bajo (L), **B** está conectado a  $V_{cc} = 5V$  (alimentación) y por lo tanto siempre está en nivel de voltaje alto (H) y **CLR** está conectado a la salida del 555 en astable, el cual genera una señal rectangular periódica. Cuando en CLR la señal pasa de nivel bajo a alto, el pin 4 ( $Q_1$ ) del 74123 genera (Trigger) un pulso (pasa de nivel bajo a alto por un breve tiempo, volviendo a nivel bajo) mientras que su negado ( $\overline{Q}_1$ ), el pin 13, hace lo opuesto, pasa de nivel alto a bajo por un breve tiempo, volviendo a nivel alto.

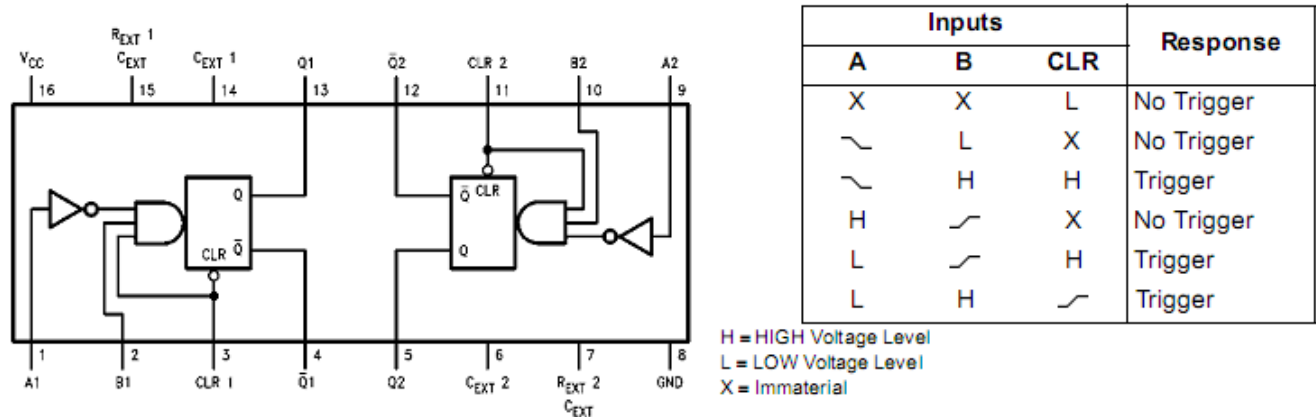


Figura 6. Diagrama de conexiones de un DM74123 y tabla de verdad.

## RESULTADOS

Los resultados obtenidos con el Subsistema de Control del Contador y del Latch son los siguientes:

Tabla 1. Datos y resultados obtenidos con un 555D y un DM74LS123N.

555D (en astable)				DM74LS123N (monostable)			
$R_1=$ 47 k $\Omega$ $R_2=$ 10 M $\Omega$ $C_1=$ 47x10 <sup>4</sup> pF				$R=$ 47 k $\Omega$ $C=$ 300 $\mu$ F			
	Teórico	Experimental	e%		Teórico	Experimental	e%
$t_1$ (s) =	3.27	4	22.3%	$T_w$ ( $\mu$ s) =	3.95	7	77.3%
$t_2$ (ms) =	15.3	16	4.6%				

$t_1$  es la duración en estado alto del 555,  $t_2$  es la duración en estado bajo y  $e\%$  es el error porcentual.  $T_w$  es la duración de la señal de disparo del 74123 (nivel en estado bajo por corresponder a la salida  $\overline{Q}_1$ ).

$$T_w = KRC (1 + 0.7/R), \text{ donde } K \approx 0.28$$

Tanto  $t_1$ ,  $t_2$ ,  $f$  y  $T_w$  fueron medidos con un osciloscopio Tektronix TDS 1002.

## DISCUSION

El único problema que se presentó fue que en un principio se obtuvieron idénticas las señales de entrada y salida en el 74123 (iguales a la señal Q del 555). Esto sucedió porque los capacitores se utilizaron intercambiados, es decir el de  $47 \times 10^4$  pF se usó en el 74123 y el de 3 nF en el 555. Al volver a intercambiarlos se solucionó el problema.

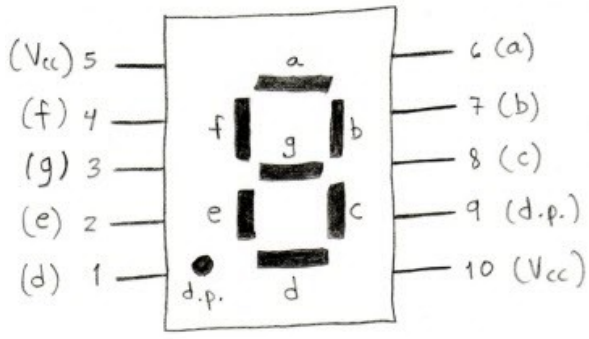
## CONEXIÓN ENTRE LOS SUBSISTEMAS CONTADOR, COMPARADOR Y DE CONTROL DEL CONTADOR Y DEL LATCH

Ya armado el Subsistema de Control del Contador y del Latch sigue su conexión con los Subsistemas Contador y Comparador, que juntos formarán el Sistema Contador-Comparador. El Contador se forma con dos 74143 y el Comparador con un 7485. Para entender de manera general cómo debe quedar el circuito véase el diagrama de la figura 1.

La descripción de los pines del 74143 se encuentra en el apéndice, al final del reporte. En la tabla 2 se indica cómo conectar los pines del 74143 incluyendo las conexiones con el Subsistema de control del Contador y el Latch (555 y el 74123) y con el Comparador 7485:

*Tabla 2. Conexiones del 74143 con el 555, el 74123 y el 7485.*

	<b>74143</b>	
<b>FUNCIÓN</b>	<b>PIN #</b>	<b>INSTRUCCIONES</b>
CLEAR INPUT	3	Se conecta a la salida $\overline{Q}$ del 74123 (esta salida la comparten los dos 74143).
CLOCK INPUT	2	Se conecta a un switch o a un generador de funciones cuadradas (para simular el Sistema Sensor de Vehículos).
PARALLEL COUNT ENABLE INPUT (PCEI)	23	A tierra.
SERIAL COUNT ENABLE INPUT (SCEI)	1	A tierra.
MAXIMUM COUNT OUTPUT	22	Se deja desconectado.
LATCH STROBE INPUT	21	Se conecta a la salida Q del 555 (esta salida la comparten los dos 74143).
LATCH OUTPUTS	17, 18, 19,	Al comparador (7485) y/o a un led c/u. Las

(Q <sub>A</sub> , Q <sub>B</sub> , Q <sub>C</sub> , Q <sub>D</sub> )	20	<p>conexiones con el comparador son las siguientes:</p> <p>En un 74143 (para A):  Q<sub>A</sub> (pin17) con el A0 (pin 10) del 7485.  Q<sub>B</sub> (pin 18) con el A1 (pin 12) del 7485.  Q<sub>C</sub> (pin 19) con el A2 (pin 13) del 7485.  Q<sub>D</sub> (pin 20) con el A3 (pin 15) del 7485.</p> <p>En el otro 74143 (para B):  Q<sub>A</sub> (pin17) con el B0 (pin 9) del 7485.  Q<sub>B</sub> (pin 18) con el B1 (pin 11) del 7485.  Q<sub>C</sub> (pin 19) con el B2 (pin 14) del 7485.  Q<sub>D</sub> (pin 20) con el B3 (pin 1) del 7485.</p>
DECIMAL POINT INPUT	7	A tierra.
BLANKING INPUT (BI)	5	A tierra.
RIPPLE-BLANKING INPUT (RBI)	4	Se deja desconectado.
RIPPLE-BLANKING OUTPUT (RBO)	6	Se deja desconectado
LED/LAMP DRIVER OUTPUTS (a, b, c, d, e, f, dp)	15, 16, 14, 9 11, 10, 13, 8	<p>Se conectan al display. Las conexiones son las siguientes (véase el siguiente diagrama):</p> <p>Pin 15 (a) con el pin 6 del display.  Pin 16 (b) con el pin 7 del display.  Pin 14 (c) con el pin 8 del display.  Pin 9 (d) con el pin 1 del display.  Pin 11 (e) con el pin 2 del display.  Pin 10 (f) con el pin 4 del display.  Pin 13 (g) con el pin 3 del display.  Pin 8 (g) con el pin 9 del display.</p>  <p>The diagram shows a 7-segment display with pins 1 through 10. The segments are labeled a (top), b (top-right), c (bottom-right), d (bottom), e (bottom-left), f (top-left), and g (middle). The decimal point (d.p.) is at the bottom left. Pin 1 is connected to the decimal point, pin 2 to segment e, pin 3 to segment g, pin 4 to segment f, pin 5 to Vcc, pin 6 to segment a, pin 7 to segment b, pin 8 to segment c, pin 9 to the decimal point, and pin 10 to Vcc.</p>

## Representación del Subsistema de Control del Contador y del Latch con la conversión de BCD a 7 segmentos

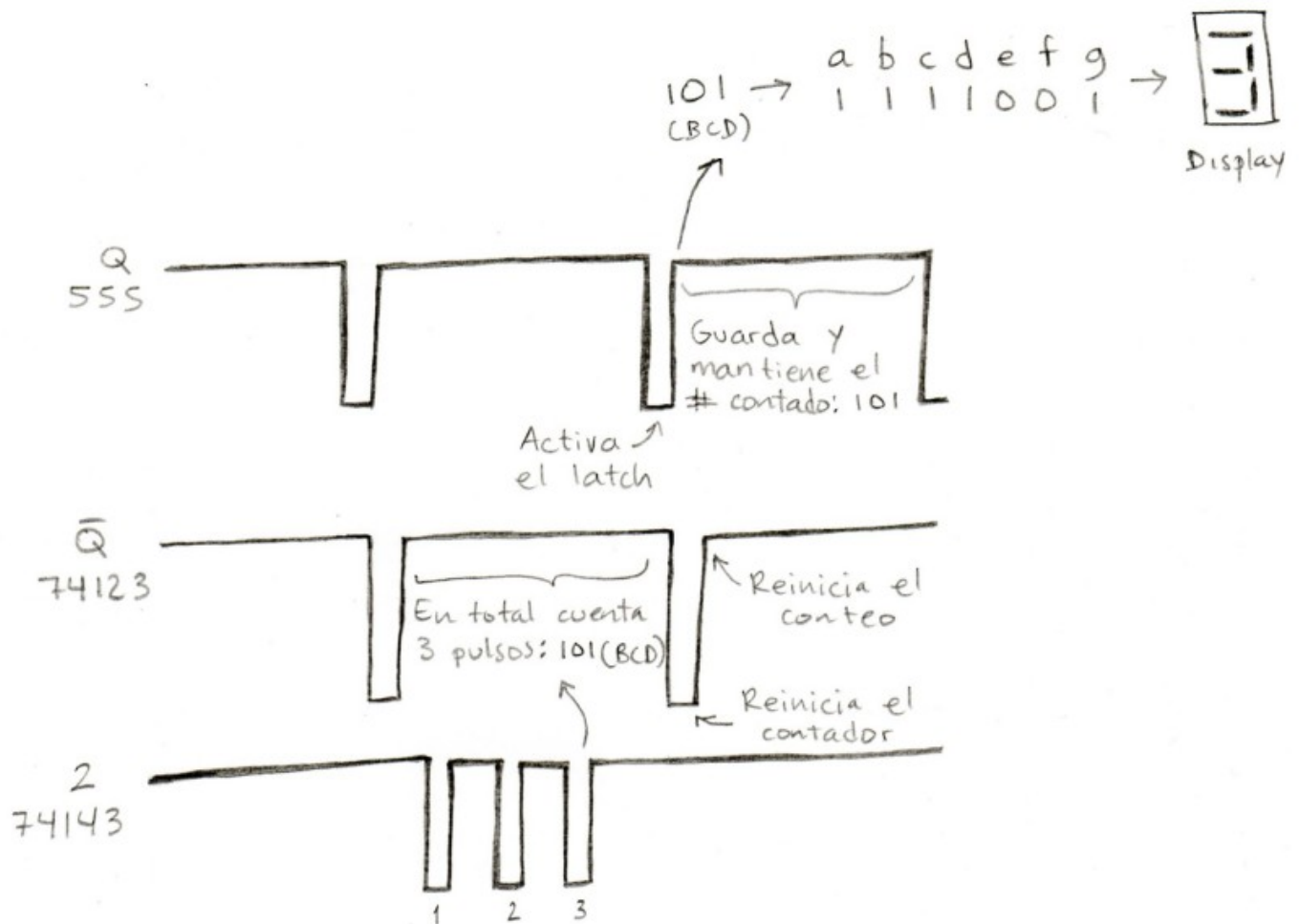


Figura 7. Representación de las señales del Subsistema de control del contador y del latch.

La señal  $Q'$  del 74123 controla los contadores de los dos 74143; mientras que la señal  $Q$  del 555 controla la memoria latch también de los dos 74143. En el pin 2 del 74143 se reciben los pulsos correspondientes a cada vehículo que pasa en alguna de las calles, por ejemplo en A. En la figura 7 se ven 3 pulsos (3 estados en nivel bajo). Durante el tiempo en que se producen los pulsos la salida  $Q'$  del 74123 está en nivel alto, es en este tiempo cuando se mantiene activo el contador del 74143 y cuenta los 3 pulsos en código binario (BCD) o sea 101 pulsos; Durante ese mismo tiempo el 555 se encuentra en estado alto pero este pasa a nivel bajo poco antes de que la señal del 123 también pase a nivel bajo (son señales ligeramente desfasadas); en nivel bajo el 555 activa la memoria (latch) del 74143 y guarda el número contado (101) anteriormente; el 555 pasa otra vez a estado alto y durante ese estado el latch retiene el número contado (en BCD), en ese mismo tiempo el decoder lo decodifica a siete segmentos para verlo en decimal (3) en un display conectado al 74143. Durante ese tiempo la salida  $Q'$  del 123 pasó a estado bajo lo que reinicia el contador; luego pasa a estado alto por lo que inicia nuevamente el conteo y todo el proceso se repite.

La siguiente tabla muestra las equivalencias entre los sistemas decimal, BCD y de siete segmentos.

Decimal	BCD	Siete segmentos						
0	0000	a	b	c	d	e	f	g
1	0001	1	1	1	1	1	1	0
2	0010	0	1	1	0	0	0	0
3	0011	1	1	0	1	1	0	1
4	0100	1	1	1	1	0	0	1
5	0101	0	1	1	0	0	1	1
6	0110	1	0	1	1	1	1	1
7	0111	1	1	1	1	0	0	0
8	1000	1	1	1	1	1	1	1
9	1001	1	1	1	1	0	1	1

Tabla 3. Equivalencia entre los sistemas decimal, BCD y de siete segmentos. A la derecha se muestran los segmentos de un display con sus correspondientes letras de identificación.

### 3. SISTEMA DE CONTROL DE LAS LUCES DEL SEMÁFORO

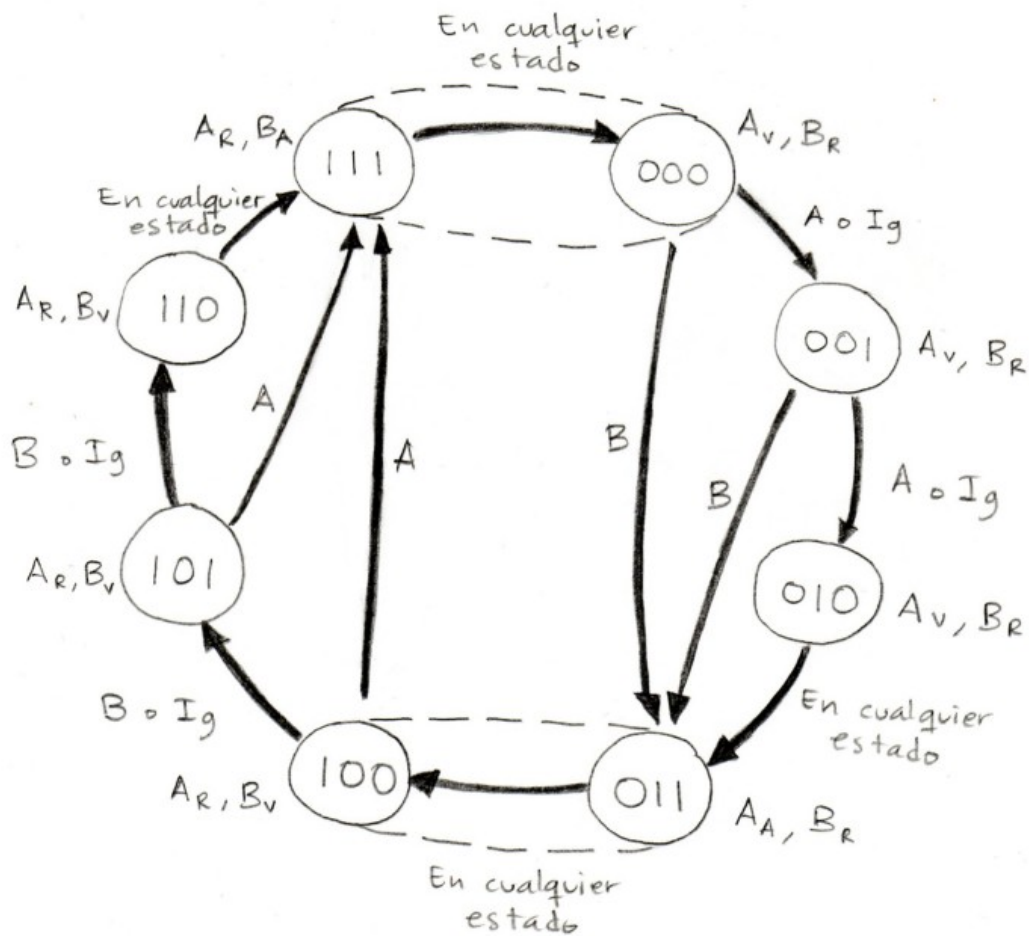


Figura 8. Sistema de control de las luces del semáforo. Notación:  $A$  significa  $A > B$ ,  $B$  significa  $B > A$  y  $I_g$  significa  $A = B$ . Los subíndices en  $A$  y  $B$  indican el color de la luz del semáforo, R rojo, A amarillo y V verde.



De manera general el sistema funciona de la siguiente forma: El sistema toma decisiones a partir de la cantidad de vehículos que hay en cada calle (A y B). Si  $A > B$  permite que pasen los vehículos en A, si  $B > A$  permite que circulen los vehículos en B. Cuando  $A = B$  el semáforo que haya estado en verde inmediatamente antes permanecerá igual. Sin embargo, ningún semáforo podrá estar en verde más de tres veces consecutivas; después de la tercera vez ese semáforo se pondrá en rojo y el otro pasará a amarillo y enseguida a verde. En los siguientes estados el sistema vuelve a aplicar estas reglas comparando nuevamente si  $A > B$ ,  $B > A$  o  $B = A$ .

La figura 8 muestra el ciclo completo. Iniciamos en el estado 000 en el que el semáforo de A está en verde ( $A_V$ ) y el de B está en rojo ( $B_R$ ), para pasar a otro estado pueden ocurrir dos cosas: 1) Que el tráfico en B sea mayor que en A ( $B > A$  denotado sólo por la letra B); de aquí los semáforos pasan al estado 011 en el que las luces de A cambian a amarillo ( $A_A$ ) y las de B a rojo ( $B_R$ ), y enseguida pasan al estado 100 en el que las luces de A cambian a rojo ( $A_R$ ) y las de B a verde ( $B_V$ ). 2) Que el tráfico en A sea mayor que en B ( $A > B$  denotado sólo por la letra A) o que el tráfico sea igual en A y en B ( $A = B$ , denotado por Ig); en cualquiera de estos casos las luces se mantendrán igual ( $A_V$  y  $B_R$ ) pero será un nuevo estado denominado 001. Para pasar del estado 001 a otro podría ocurrir otra vez lo mismo que en los incisos 1) y 2), aunque en este último inciso pasaríamos al estado denominado 010. Ahora bien, obsérvese que al llegar al estado 010 el semáforo de A ya ha estado en verde 3 veces consecutivas por lo que en el siguiente estado -sin importar si  $A > B$ ,  $B > A$  o  $A = B$ - las luces de A cambian a amarillo ( $A_A$ ) y las de B a rojo ( $B_R$ ), y enseguida las de A pasan a rojo ( $A_R$ ) y las de B a verde ( $B_V$ ); estos son los mismos estados 011 y 100 del inciso 1), respectivamente. Nótese que los estados 011 y 100 pueden considerarse como uno solo porque el 011, además de durar sólo 4 o 5 segundos, debe ser seguido siempre por el estado 100.

A partir del estado 100 el análisis es semejante al del principio (estado 000), sólo que ahora los papeles se intercambian, es decir el semáforo de A está en rojo ( $A_R$ ) y el de B en verde ( $B_V$ ).

## Implementación del sistema de control de las luces del semáforo utilizando flip-flops JK

Para implementar el sistema de control de las luces del semáforo con circuitos digitales se utilizarán flip flops. Existen diversos tipos de flip flops: JK, SR, T o D.

Las siguientes tablas de verdad describen el comportamiento general de los flip flops JK, SR, T y D. Las salidas  $Q_t$  indican que estas ocurren en un tiempo t, mientras que las salidas  $Q_{t+1}$  ocurren un tiempo después en un nuevo estado de las entradas J, K, S, R, T o D.

Cada par de valores de salida  $Q_t$  y  $Q_{t+1}$  puede obtenerse únicamente con la combinación correspondiente de valores de entradas J, K, S, R, T o D. Las X pueden tomar cualquier valor binario (0 o 1).

J	K	$Q_t$	$Q_{t+1}$	S	R	$Q_t$	$Q_{t+1}$
0	X	0	0	0	X	0	0
1	X	0	1	1	0	0	1
X	1	1	0	0	1	1	0
X	0	1	1	X	0	1	1

<b>T</b>	<b>Q<sub>t</sub></b>	<b>Q<sub>t+1</sub></b>
0	0	0
0	1	1
1	0	1
1	1	0

<b>D</b>	<b>Q<sub>t</sub></b>	<b>Q<sub>t+1</sub></b>
0	0	0
0	1	0
1	0	1
1	1	1

Nosotros utilizaremos flip flops JK y necesitaremos 3 de ellos porque esta es la cantidad necesaria y suficiente para generar los 8 ( $= 2^3$ ) estados del semáforo inteligente (ver figura 8).

El siguiente paso consiste en realizar una tabla de verdad de cada una de las variables que intervienen. En la siguiente tabla se muestran los valores que tomarían los flip flops JK, SR, T y D si generaran los 8 estados del semáforo inteligente.

Hay que mencionar que  $J_1$ ,  $K_1$ ,  $J_2$ ,  $K_2$ ,  $J_3$  y  $K_3$  dependen de  $A$ ,  $B$ ,  $Ig$  y de las entradas de  $F_1$  ( $Q_1$  en  $t$ ),  $F_2$  ( $Q_2$  en  $t$ ) y  $F_3$  ( $Q_3$  en  $t$ ). Las luces del semáforo ( $Av$ ,  $Aa$ ,  $Ar$ ,  $Bv$ ,  $Ba$  y  $Br$ ) dependen, también, de  $A$ ,  $B$ ,  $Ig$  y además de las salidas de  $F_1$  ( $Q_1$  en  $t+1$ ),  $F_2$  ( $Q_2$  en  $t+1$ ) y  $F_3$  ( $Q_3$  en  $t+1$ ).

Para obtener expresiones booleanas más simplificadas de los FFs JK y de las luces del semáforo, se trazan sus mapas de Karnaugh.



Tabla 4. Tabla de verdad del semáforo inteligente para distintos FFs: JK, SR, T y D.

Entradas (en t)							Salidas (en t+1)			Entradas de FFs JK						Entradas de FFs SR						Entradas de FFs T			Entradas de FFs D			Salidas Luces del semáforo								
A	B	Ig	F <sub>1</sub>	F <sub>2</sub>	F <sub>3</sub>		F <sub>1</sub>	F <sub>2</sub>	F <sub>3</sub>	J <sub>1</sub>	K <sub>1</sub>	J <sub>2</sub>	K <sub>2</sub>	J <sub>3</sub>	K <sub>3</sub>	S <sub>1</sub>	R <sub>1</sub>	S <sub>2</sub>	R <sub>2</sub>	S <sub>3</sub>	R <sub>3</sub>	T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>	A <sub>v</sub>	Aa	Ar	Bv	Ba	Br			
0	0	0	0	0	0		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
0	0	0	0	0	1		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
0	0	0	0	1	0		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
0	0	0	0	1	1		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
0	0	0	1	0	0		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
0	0	0	1	0	1		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
0	0	0	1	1	0		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
0	0	0	1	1	1		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
0	0	1	0	0	0		0	0	1	0	X	0	X	1	X	0	X	0	X	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	1	
0	0	1	0	0	1		0	1	0	0	X	1	X	X	1	0	X	1	0	0	0	1	0	1	0	1	0	1	0	1	0	0	0	0	0	1
0	0	1	0	1	0		0	1	1	0	X	X	0	1	X	1	0	X	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0	0	0	1
0	0	1	0	1	1		1	0	0	1	X	X	1	X	1	X	0	0	1	0	1	0	1	1	1	1	0	1	0	0	0	1	1	0	0	0
0	0	1	1	0	0		1	0	1	X	0	0	X	1	X	X	0	0	X	1	0	0	1	0	1	0	1	0	1	0	0	1	1	0	0	0
0	0	1	1	0	1		1	1	0	X	0	1	X	X	1	X	0	1	0	0	1	0	1	1	1	1	0	1	0	0	1	1	0	0	0	0
0	0	1	1	1	0		1	1	1	X	0	X	0	1	X	X	0	X	0	1	0	1	0	0	1	0	1	0	1	0	0	1	0	1	0	0
0	0	1	1	1	1		0	0	0	X	1	X	1	X	1	0	1	0	1	0	1	0	1	1	1	0	0	0	0	0	0	0	0	0	1	1
0	1	0	0	0	0		0	1	1	0	X	1	X	1	X	0	X	1	0	1	0	1	0	1	1	0	1	1	0	1	0	0	0	0	1	1
0	1	0	0	0	1		0	1	1	0	X	1	X	X	0	0	X	1	0	X	0	1	0	1	0	1	0	1	1	0	1	0	0	0	1	1
0	1	0	0	1	0		0	1	1	0	X	X	0	1	X	1	0	X	0	1	0	1	0	0	1	1	1	1	0	1	0	0	0	0	1	1
0	1	0	0	1	1		1	0	0	1	X	X	1	X	1	X	0	0	1	0	1	0	1	1	1	1	0	0	1	0	0	1	1	0	0	0
0	1	0	1	0	0		1	0	1	X	0	0	X	1	X	X	0	0	X	1	0	0	1	1	1	0	1	0	1	0	0	1	1	0	0	0
0	1	0	1	0	1		1	1	0	X	0	1	X	X	1	X	0	1	X	0	1	0	1	1	1	0	1	1	0	0	1	1	0	0	0	0
0	1	0	1	1	0		1	1	1	X	0	X	0	1	X	X	0	X	0	1	0	1	0	0	1	1	1	1	0	0	1	0	1	0	0	0
0	1	0	1	1	1		0	0	0	X	1	X	1	X	1	0	1	0	1	0	1	0	1	1	1	0	0	0	0	0	0	0	0	0	1	1
0	1	1	0	0	0		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0	1	1	0	0	1		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0	1	1	0	1	0		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0	1	1	0	1	1		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0	1	1	1	0	0		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0	1	1	1	0	1		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0	1	1	1	1	0		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

[illegible]

### K-Mapa de $J_1$

Entradas									
$F_1 F_2 F_3$		000	001	011	010	110	111	101	100
A B	$I_g$								
0 0	0	X	X	X	X	X	X	X	X
0 0	1	0	0	1	0	X	X	X	X
0 1	1	X	X	X	X	X	X	X	X
0 1	0	0	0	1	0	X	X	X	X
1 1	0	X	X	X	X	X	X	X	X
1 1	1	X	X	X	X	X	X	X	X
1 0	1	X	X	X	X	X	X	X	X
1 0	0	0	0	1	0	X	X	X	X

$$\Rightarrow J_1 = \overline{F_1} F_2 F_3$$

### K-Mapa de $K_1$

Entradas									
$F_1 F_2 F_3$		000	001	011	010	110	111	101	100
A B	$I_g$								
0 0	0	X	X	X	X	X	X	X	X
0 0	1	X	X	X	X	0	1	0	0
0 1	1	X	X	X	X	X	X	X	X
0 1	0	X	X	X	X	0	1	0	0
1 1	0	X	X	X	X	X	X	X	X
1 1	1	X	X	X	X	X	X	X	X
1 0	1	X	X	X	X	X	X	X	X
1 0	0	X	X	X	X	0	1	0	0

$$\Rightarrow K_1 = F_1 F_2 F_3$$

### K-Mapa de $J_2$

Entradas									
$F_1 F_2 F_3$		000	001	011	010	110	111	101	100
A B	Ig								
0 0	0	X	X	X	X	X	X	X	X
0 0	1	0	1	X	X	X	X	1	0
0 1	1	X	X	X	X	X	X	X	X
0 1	0	1	1	X	X	X	X	1	0
1 1	0	X	X	X	X	X	X	X	X
1 1	1	X	X	X	X	X	X	X	X
1 0	1	X	X	X	X	X	X	X	X
1 0	0	0	1	X	X	X	X	1	1

$$\Rightarrow J_2 = \overline{F_1}F_3 + B\overline{F_1} + F_1F_3 + AF_1 = AF_1 + B\overline{F_1} + F_3$$

### K-Mapa de $K_2$

Entradas									
$F_1 F_2 F_3$		000	001	011	010	110	111	101	100
A B	Ig								
0 0	0	X	X	X	X	X	X	X	X
0 0	1	X	X	1	0	0	1	X	X
0 1	1	X	X	X	X	X	X	X	X
0 1	0	X	X	1	0	0	1	X	X
1 1	0	X	X	X	X	X	X	X	X
1 1	1	X	X	X	X	X	X	X	X
1 0	1	X	X	X	X	X	X	X	X
1 0	0	X	X	1	0	0	1	X	X

$$\Rightarrow K_2 = \overline{F_1}F_3 + F_1F_3 = F_3$$

### K-Mapa de $J_3$

Entradas										
$F_1$	$F_2$	$F_3$	000	001	011	010	110	111	101	100
A	B	$I_g$								
0	0	0	X	X	X	X	X	X	X	X
0	0	1	1	X	X	1	1	X	X	1
0	1	1	X	X	X	X	X	X	X	X
0	1	0	1	X	X	1	1	X	X	1
1	1	0	X	X	X	X	X	X	X	X
1	1	1	X	X	X	X	X	X	X	X
1	0	1	X	X	X	X	X	X	X	X
1	0	0	1	X	X	1	1	X	X	1

$$\Rightarrow J_3 = 1$$

### K-Mapa de $K_3$

Entradas										
$F_1$	$F_2$	$F_3$	000	001	011	010	110	111	101	100
A	B	$I_g$								
0	0	0	X	X	X	X	X	X	X	X
0	0	1	X	1	1	X	X	1	1	X
0	1	1	X	X	X	X	X	X	X	X
0	1	0	X	0	1	X	X	1	1	X
1	1	0	X	X	X	X	X	X	X	X
1	1	1	X	X	X	X	X	X	X	X
1	0	1	X	X	X	X	X	X	X	X
1	0	0	X	1	1	X	X	1	0	X

$$\Rightarrow K_3 = \overline{A}F_1 + \overline{B}F_1 + F_2$$



### K-Mapa de Av

		Salidas F <sub>1</sub> F <sub>2</sub> F <sub>3</sub>							
		000	001	011	010	110	111	101	100
A B	Ig								
0 0	0	X	X	X	X	X	X	X	X
0 0	1	1	1	0	1	0	0	0	0
0 1	1	X	X	X	X	X	X	X	X
0 1	0	1	X	0	X	0	0	0	0
1 1	0	X	X	X	X	X	X	X	X
1 1	1	X	X	X	X	X	X	X	X
1 0	1	X	X	X	X	X	X	X	X
1 0	0	1	1	0	1	X	0	X	0

$$\Rightarrow Av = \overline{F_1}\overline{F_2} + \overline{F_1}F_2\overline{F_3} = \overline{F_1}(F_2 + F_2\overline{F_3})$$

### K-Mapa de Aa

		Salidas F <sub>1</sub> F <sub>2</sub> F <sub>3</sub>							
		000	001	011	010	110	111	101	100
A B	Ig								
0 0	0	X	X	X	X	X	X	X	X
0 0	1	0	0	1	0	0	0	0	0
0 1	1	X	X	X	X	X	X	X	X
0 1	0	0	X	1	X	0	0	0	0
1 1	0	X	X	X	X	X	X	X	X
1 1	1	X	X	X	X	X	X	X	X
1 0	1	X	X	X	X	X	X	X	X
1 0	0	0	0	1	0	X	0	X	0

$$\Rightarrow Aa = \overline{F_1}F_2F_3 = \overline{F_1}(F_2F_3)$$

### K-Mapa de Ar

		Salidas F <sub>1</sub> F <sub>2</sub> F <sub>3</sub>							
		000	001	011	010	110	111	101	100
A B Ig	0 0 0	X	X	X	X	X	X	X	X
	0 0 1	0	0	0	0	1	1	1	1
	0 1 1	X	X	X	X	X	X	X	X
	0 1 0	0	X	0	X	1	1	1	1
	1 1 0	X	X	X	X	X	X	X	X
	1 1 1	X	X	X	X	X	X	X	X
	1 0 1	X	X	X	X	X	X	X	X
	1 0 0	0	0	0	0	X	1	X	1

$$\Rightarrow Ar = F_1$$

### K-Mapa de Bv

		Salidas F <sub>1</sub> F <sub>2</sub> F <sub>3</sub>							
		000	001	011	010	110	111	101	100
A B Ig	0 0 0	X	X	X	X	X	X	X	X
	0 0 1	0	0	0	0	1	0	1	1
	0 1 1	X	X	X	X	X	X	X	X
	0 1 0	0	X	0	X	1	0	1	1
	1 1 0	X	X	X	X	X	X	X	X
	1 1 1	X	X	X	X	X	X	X	X
	1 0 1	X	X	X	X	X	X	X	X
	1 0 0	0	0	0	0	X	0	X	1

$$\Rightarrow Bv = F_1 \overline{F_2} + F_1 F_2 \overline{F_3} = F_1 (\overline{F_2} + F_2 \overline{F_3})$$

### K-Mapa de Ba

		Salidas $F_1 F_2 F_3$							
		000	001	011	010	110	111	101	100
A B	Ig								
		00	01	11	10	10	11	01	00
0	0	X	X	X	X	X	X	X	X
0	1	0	0	0	0	0	1	0	0
1	1	X	X	X	X	X	X	X	X
1	0	0	X	0	X	0	1	0	0
1	1	X	X	X	X	X	X	X	X
1	1	X	X	X	X	X	X	X	X
1	0	X	X	X	X	X	X	X	X
1	0	0	0	0	0	X	1	X	0

$$\Rightarrow Ba = F_1 F_2 F_3 = F_1 (F_2 F_3)$$

### K-Mapa de Br

		Salidas $F_1 F_2 F_3$							
		000	001	011	010	110	111	101	100
A B	Ig								
		00	01	11	10	10	11	01	00
0	0	X	X	X	X	X	X	X	X
0	1	1	1	1	1	0	0	0	0
1	1	X	X	X	X	X	X	X	X
1	0	1	X	1	X	0	0	0	0
1	1	X	X	X	X	X	X	X	X
1	1	X	X	X	X	X	X	X	X
1	0	X	X	X	X	X	X	X	X
1	0	1	1	1	1	X	0	X	0

$$\Rightarrow Ar = \overline{F_1}$$

En resumen, las ecuaciones obtenidas con los K-Mapas son:

$$\begin{array}{ll}
 J_1 = \overline{F_1}F_2F_3 & Av = \overline{F_1}(\overline{F_2} + F_2\overline{F_3}) \\
 K_1 = F_1F_2F_3 & Aa = \overline{F_1}(F_2F_3) \\
 J_2 = AF_1 + B\overline{F_1} + F_3 & Ar = F_1 \\
 K_2 = F_3 & Bv = F_1(\overline{F_2} + F_2\overline{F_3}) \\
 J_3 = 1 & Ba = F_1(F_2F_3) \\
 K_3 = \overline{A}F_1 + \overline{B}\overline{F_1} + F_2 & Ar = \overline{F_1}
 \end{array}$$

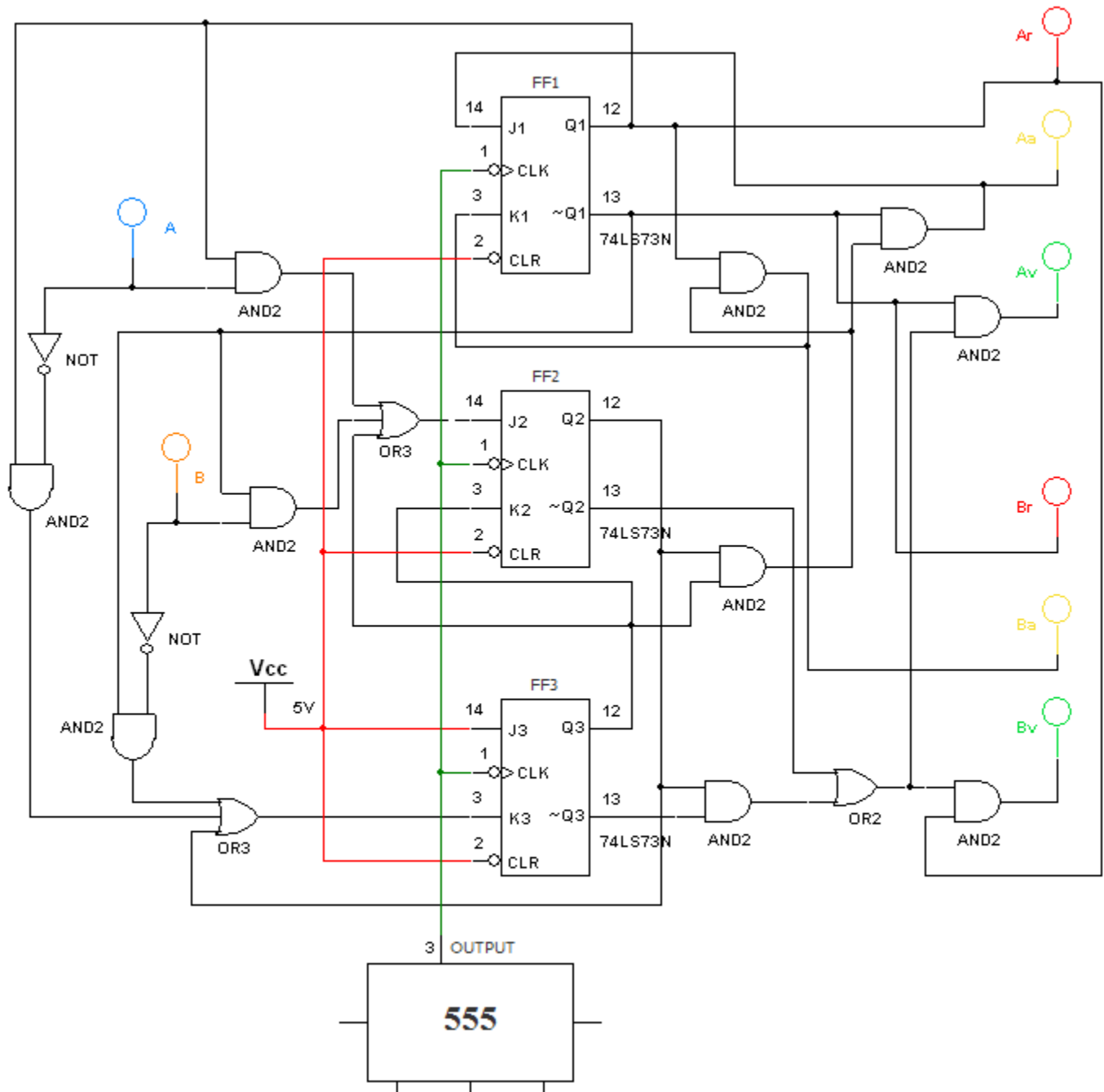
Es interesante el hecho de que ninguna de las funciones depende de la variable  $Ig$ .

A partir de estas ecuaciones se diseña el circuito del Sistema de Control de las Luces del Semáforo utilizando compuertas lógicas y 3 flip-flops JK como se muestra en la figura 9.

Nótese que en la parte inferior de la figura 9, el temporizador 555 es el mismo que el del Sistema Contador-Comparador (ver figuras 1 o 4). Por otra parte, en los nodos de los indicadores A y B (en color azul y naranja, respectivamente) se conectan las salidas  $A > B$  (pin 5) y  $B > A$  (pin 7), respectivamente, del comparador 7485 (también, del Sistema Contador-Comparador).

De esta forma se concluye la elaboración del Semáforo Inteligente utilizando componentes electrónicos discretos e integrados.

Figura 9. Circuito del Sistema de Control de las Luces del Semáforo con FFs JK.



## BIBLIOGRAFÍA Y REFERENCIAS

CONSTANTINI, S. *Mapas de Karnaugh*. [En línea]. 2001. Disponible en: <http://medusa.unimet.edu.ve/sistemas/bpis03/mdkminimizacion.htm> [Consultado el 8 de octubre de 2009].

Datasheetcatalog.com. *555 datasheet*. [Documento PDF]. 2009. Disponible en: [http://www.datasheetcatalog.com/datasheets\\_pdf/N/E/5/5/NE555N.shtml](http://www.datasheetcatalog.com/datasheets_pdf/N/E/5/5/NE555N.shtml) [Consultado el 28 de agosto de 2009].

Datasheetcatalog.com. *74123 datasheet*. [Documento PDF]. 2009. Disponible en: [http://www.datasheetcatalog.com/datasheets\\_pdf/D/M/7/4/DM74123.shtml](http://www.datasheetcatalog.com/datasheets_pdf/D/M/7/4/DM74123.shtml) [Consultado el 28 de agosto de 2009].

Datasheetcatalog.com. *7485 datasheet*. [Documento PDF]. 2009. Disponible en: [http://www.datasheetcatalog.com/datasheets\\_pdf/S/N/7/4/SN7485N.shtml](http://www.datasheetcatalog.com/datasheets_pdf/S/N/7/4/SN7485N.shtml) [Consultado el 20 de septiembre de 2009].

Datasheetcatalog.com. *74143 datasheet*. [Documento PDF]. 2009. Disponible en: [http://www.datasheetcatalog.com/datasheets\\_pdf/S/N/7/4/SN74143.shtml](http://www.datasheetcatalog.com/datasheets_pdf/S/N/7/4/SN74143.shtml) [Consultado el 20 de septiembre de 2009].

MALVINO, A. *Principios de Electrónica*. 6ª edición: España: McGraw-Hill, 2000. p. 959-960.

MOLINA, M. *Temporizadores*. [En línea]. 2009. Disponible en: <http://www.profesormolina.com.ar/electromec/temporizadores.htm> [Consultado el 29 de agosto de 2009].

NELSON, V.; NAGLE H. *Análisis y diseño de circuitos lógicos digitales*. Prentice Hall, 1997.

Wikipedia, la enciclopedia libre. *Monoestable*. [En línea]. 16 de agosto de 2009. Disponible en: <http://es.wikipedia.org/wiki/Monoestable> [Consultado el 28 de agosto de 2009].

Wikipedia, la enciclopedia libre. *Multivibrador*. [En línea]. 17 de julio de 2009. Disponible en: <http://es.wikipedia.org/wiki/Multivibrador> [Consultado el 28 de agosto de 2009].

### 3ª parte

## Diseño del semáforo inteligente con un microcontrolador PICAXE

### OBJETIVO

Realizar el sistema de control del semáforo inteligente -planteado en la 2ª parte de este reporte- utilizando un microcontrolador PICAXE 14M.

### INTRODUCCION

La utilización de un chip PICAXE simplifica enormemente el proceso de diseño del Semáforo Inteligente. Por una parte, por tratarse de un microcontrolador ya no es necesario, en este caso, utilizar compuertas lógicas, flip-flops, multivibradores, comparadores, etc., es decir, todos los chips utilizados son sustituidos por un solo PICAXE.

Esto se debe a que un microcontrolador es un circuito integrado que incluye en su interior las tres unidades funcionales de una computadora: unidad central de procesamiento, memoria y unidades de E/S (entrada/salida). El microcontrolador es capaz de llevar a cabo procesos lógicos; estos procesos o acciones son programados, generalmente en lenguaje ensamblador por el usuario, y son introducidos en este a través de un programador.

Por otra parte, el PICAXE es un microcontrolador mejorado debido a que simplifica la programación al usar lenguaje BASIC en lugar de ensamblador. Además, utiliza un jack stereo en lugar de un programador (que es mucho más costoso) para descargar los programas en el PICAXE. Todo esto, manteniendo las mismas funciones de los microcontroladores habituales.

### DESARROLLO

El desarrollo se divide en 3 partes principales:

- 1 – Diseño del circuito impreso.
- 2 – Diseño del programa de control del 14M.
- 3 – Descarga del programa en el 14M y prueba de funcionamiento.

#### 1 – Diseño del circuito impreso

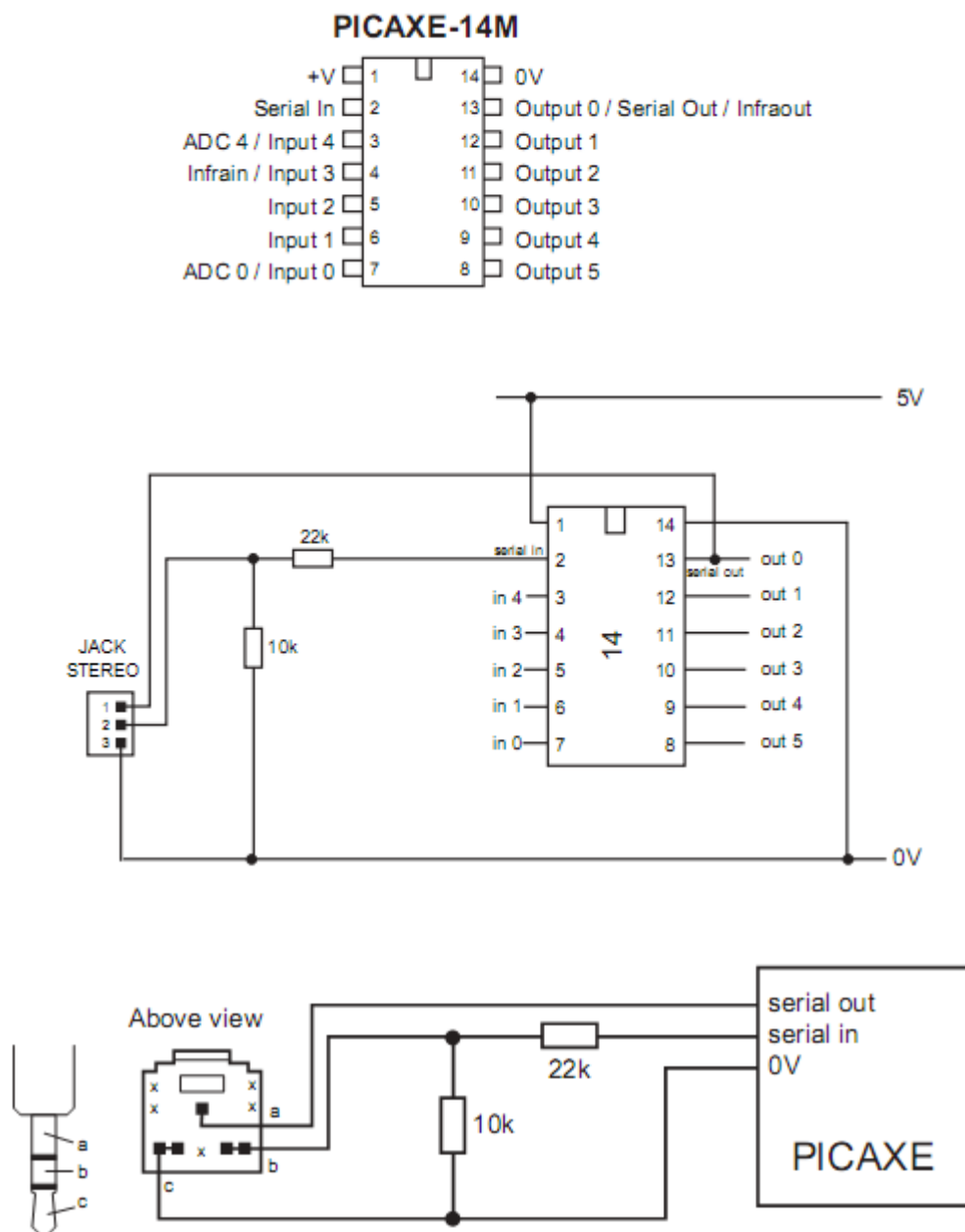
Consiste en varios pasos:

##### *a) Diseño del diagrama de conexiones.*

El manual 1 del PICAXE viene incluido en el software *Programming Editor*, pero también puede descargarse individualmente desde: <http://www.rev-ed.co.uk/PICAXE/es/index.htm>

En él buscamos la disposición de los pines del 14M, así como el circuito mínimo de operación.

Figura 10. Disposición de pines del 14M y circuito mínimo de funcionamiento.



**Nota:**

La salida 0 (pin 13) se usa durante la descarga de programas, pero puede utilizarse también como una salida de propósito general una vez que la descarga ha finalizado. En placas de circuitos impresos puede usarse un jumper para poder conectar el pin 13 al socket de descarga o a la salida.

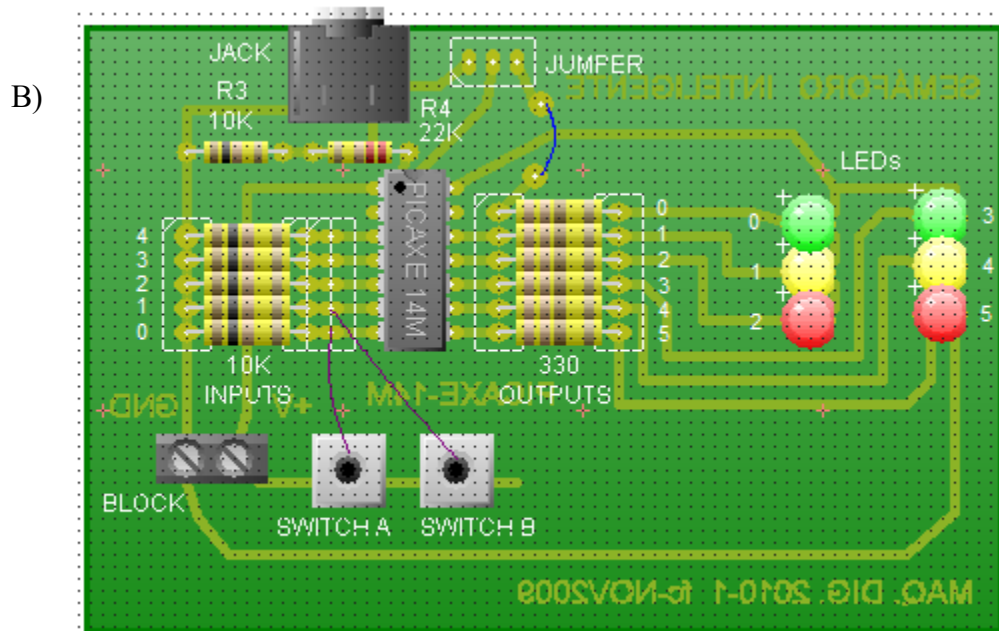
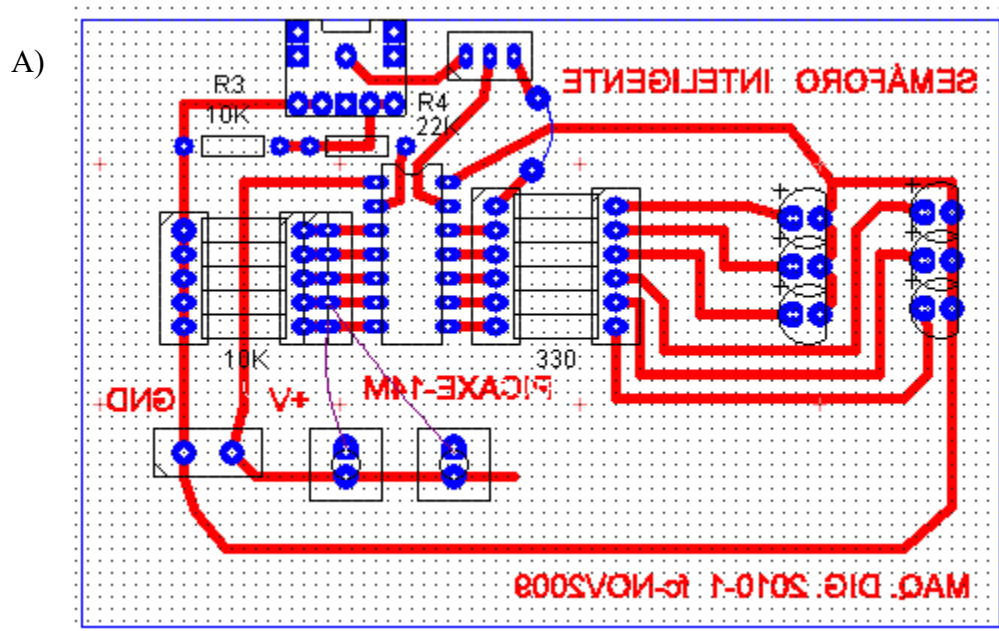
Para realizar el diseño del circuito se utilizará el programa *PCB Wizard* 3.5. La placa debe contener los siguientes componentes (ver figura 11):

- 1 PICAXE 14M o un zócalo de 14 pines.
- 1 Jack stereo para PICAXEs.
- 5 Resistores de 10 k $\Omega$  en las 5 entradas del 14M.



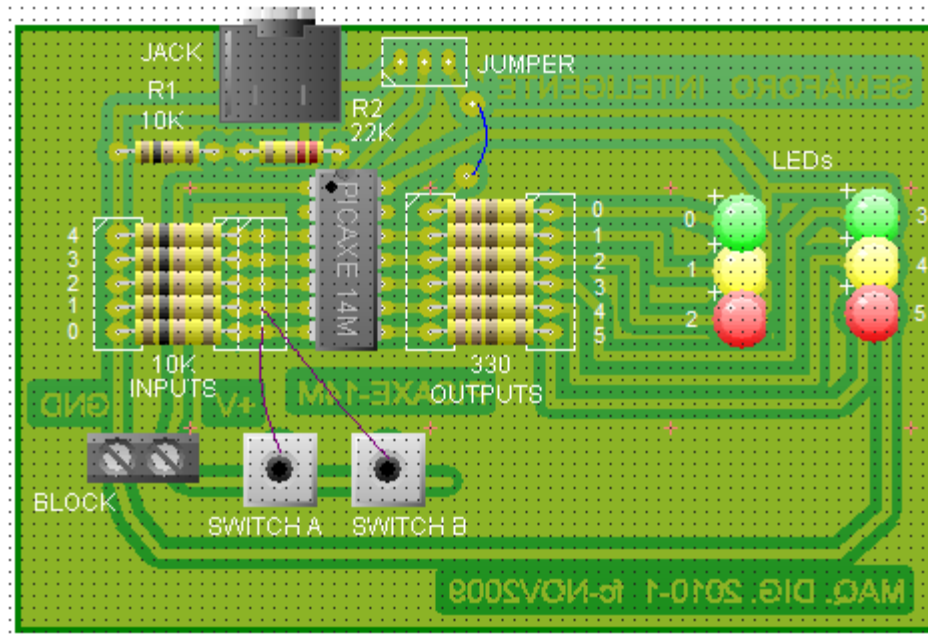
- 1 Resistor de 10 k $\Omega$  y 1 de 22 k $\Omega$  para el jack stereo (ver figura 11 B).
- 1 Header de 3 pines con 1 jumper y 1 cable conectado como en la figura 11.
- 6 Resistores de 330 $\Omega$  en las 6 salidas del 14M.
- 6 LEDs de colores verde, rojo y amarillo (2 de c/u) correspondientes a las luces del semáforo.
- 2 Interruptores con 2 cables conectados a las entradas 0 y 1 que simularan el paso de vehículos en A y B.
- 1 Block de 2 terminales.

Figura 11. Diseño del circuito impreso del semáforo inteligente. Diseño realizado con PCB Wizard 3.5. A) Vista "Normal". B) Vista "Real World".



Al realizar el diseño existe la opción de dejar con cobre las zonas que no se vayan a utilizar (ver figura 12). Esto permitirá aumentar la duración del cloruro férrico ( $\text{FeCl}_3$ ) y acabar más rápido cada vez que se utilice, ya que la superficie sobre la que actúe será mucho menor.

Figura 12. El circuito de la figura 11, inciso B) recubierto de cobre en las zonas sin uso.



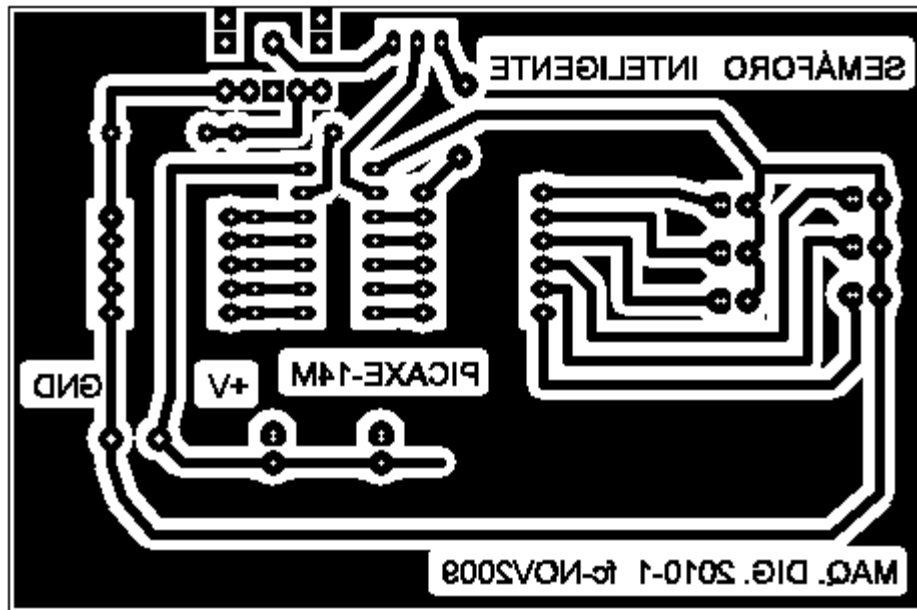
#### b) Preparación de la placa para circuitos impresos.

Cortar la placa virgen a las medidas de nuestro diseño utilizando una segqueta para metal. Limpiar escrupulosamente la placa hasta que el cobre quede brillante, libre de suciedad y sin grasa. Una vez limpia, tener cuidado de no tocar con los dedos la superficie de cobre. Utilizaremos para ello una lana de acero, que debe ser lo mas fina posible para que no queden rayas. Emplear polvo limpiador o algún agente químico pueden resultar útiles, como por ejemplo un limpia metales y un trapo en lugar de la lana de acero. Podemos probar distintos métodos, de acuerdo a los elementos que tengamos al alcance.

#### c) Impresión láser del circuito en papel satinado.

Una vez realizado el diseño del circuito en *PCB Wizard*, se escoge la vista “Artwork” y se imprime en papel satinado (en el lado brillante), utilizando una impresora láser. Estos dos requisitos son importantes, sin ellos la impresión no podrá adherirse correctamente a la placa para circuitos. El diseño impreso debe verse como en la figura 13.

Figura 13. Diseño del circuito impreso del semáforo inteligente. Vista “Artwork” para imprimir.



*d) Planchado de la impresión sobre una placa para circuitos impresos (placa con una capa de cobre).*

Se toma la hoja impresa y se coloca el lado impreso en contacto con la zona de cobre de la placa para circuitos. La hoja puede fijarse con un poco de cinta adhesiva si se considera necesario.

A continuación, se plancha la hoja sobre la placa, utilizando una plancha para ropa a temperatura máxima. El planchado debe ser lento, presionando con la punta de la plancha, especialmente sobre las pistas del circuito. El tiempo de planchado varía entre 5 y 10 minutos aproximadamente.

Si el planchado se realizó correctamente la impresión se habrá adherido íntegramente sobre la placa. Todas las zonas en negro de la impresión (ver figura 13) cubrirán el cobre de la placa para circuitos, excepto en las zonas en blanco que quedarán descubiertas.

*e) Remoción del papel.*

Para remover el papel simplemente se sumerge la placa en agua y se presiona empujando con los dedos el papel adherido. No debe quedar papel en la placa.

*f) Eliminación del cobre sobrante.*

Verter en un recipiente de vidrio entre 300 y 700 ml de cloruro férrico. Sumergir la placa dentro de él y agitar suavemente con alguna varilla de plástico o vidrio hasta que haya desaparecido el cobre de las zonas blancas de la impresión (entre 5 y 15 minutos aproximadamente). Durante este tiempo se puede calentar un poco el recipiente con el percloruro (entre 30 y 50 °C) utilizando una parrilla eléctrica, una estufa o una pistola de aire para el pelo, esto acelerará el proceso. Para finalizar, lavar la placa con abundante agua.

**Nota:**

Hay que tener cuidado con esta sustancia ya que es tóxica, ácida y muy corrosiva. Es importante lavar inmediatamente las manos y otras superficies que hayan estado en contacto con la disolución para evitar daños. Conviene además, usar el cloruro férrico en un lugar ventilado y por supuesto evitar salpicaduras, ya que puede manchar permanentemente la ropa.

**g) Perforación de las superficies de conexión en la placa.**

Perforar con un mini taladro (tipo Dremel) los puntos donde irán conectados los componentes electrónicos.

**h) Ensamblado y soldado de los componentes electrónicos.**

Finalmente, colocar los componentes electrónicos sobre el lado de la placa sin cobre (como en la figura 11 B) y a la vez ir soldando con un cautín y estaño.

## **2 – Diseño del programa de control del PICAXE-14**

Los microcontroladores PICAXE son programados usando el software gratuito *Programming Editor*. Este software proporciona un ambiente completo para generar programas para los PICAXE. Los programas pueden ser editados, compilados y descargados a los chips PICAXE (de todos los tamaños) desde el mismo software.

El software corre en cualquiera de los siguientes sistemas operativos Windows: Windows 95, 98, ME, NT, 2000, XP. Para usuarios de Linux y Mac existe un software similar: *AXEpad*. *Programming Editor* puede descargarse desde esta dirección:

<http://www.rev-ed.co.uk/PICAXE/progedit.htm>

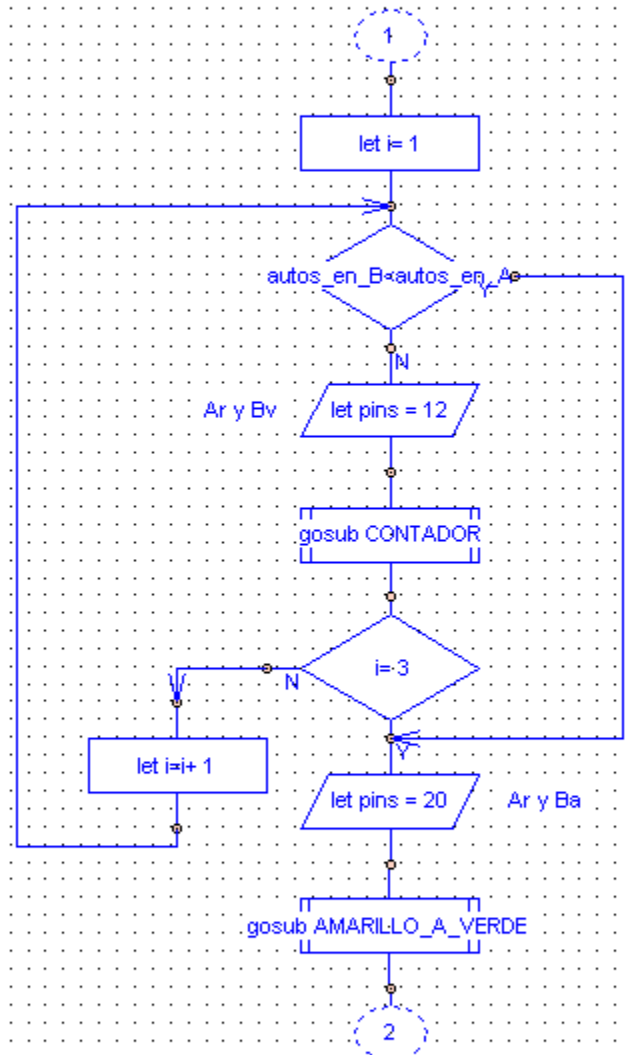
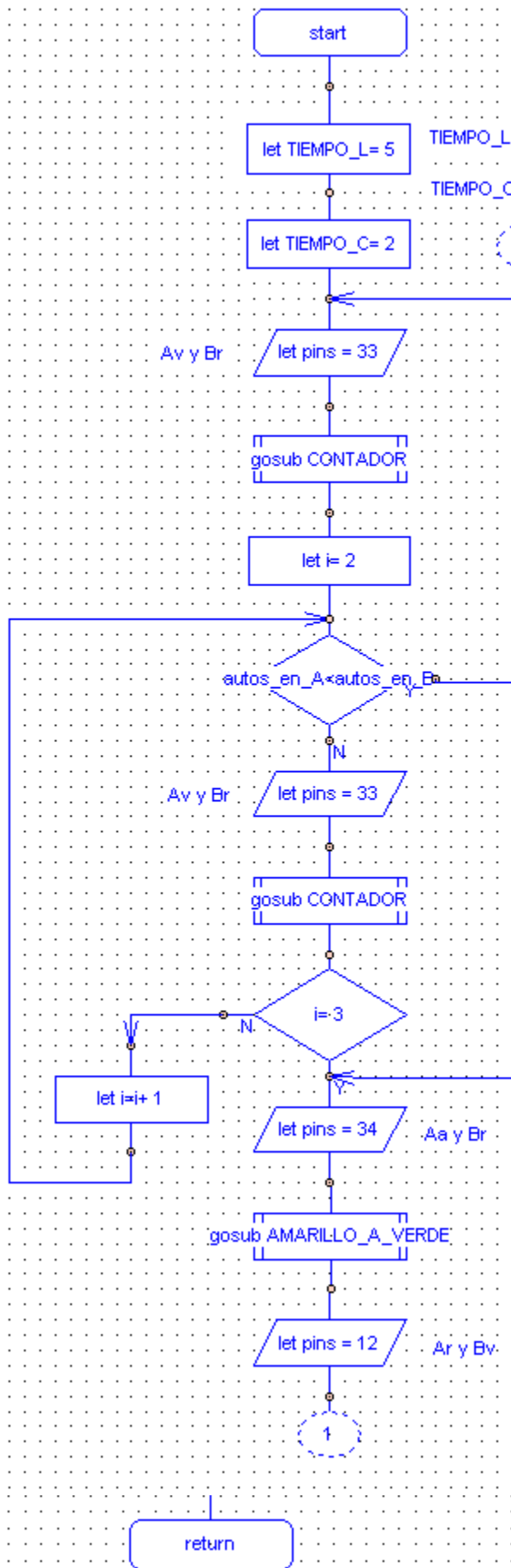
*Programming Editor* soporta dos formas de programar: Mediante texto en BASIC y en forma gráfica con diagramas de flujo. Ambos métodos usan los mismos comandos y sintaxis de BASIC. El método de diagramas de flujo permite programar gráficamente, pero usa un menor subconjunto de comandos de BASIC y es más recomendable para estudiantes jóvenes o principiantes.

Los dos métodos permiten simular gráficamente el programa antes de descargarlo en el PICAXE

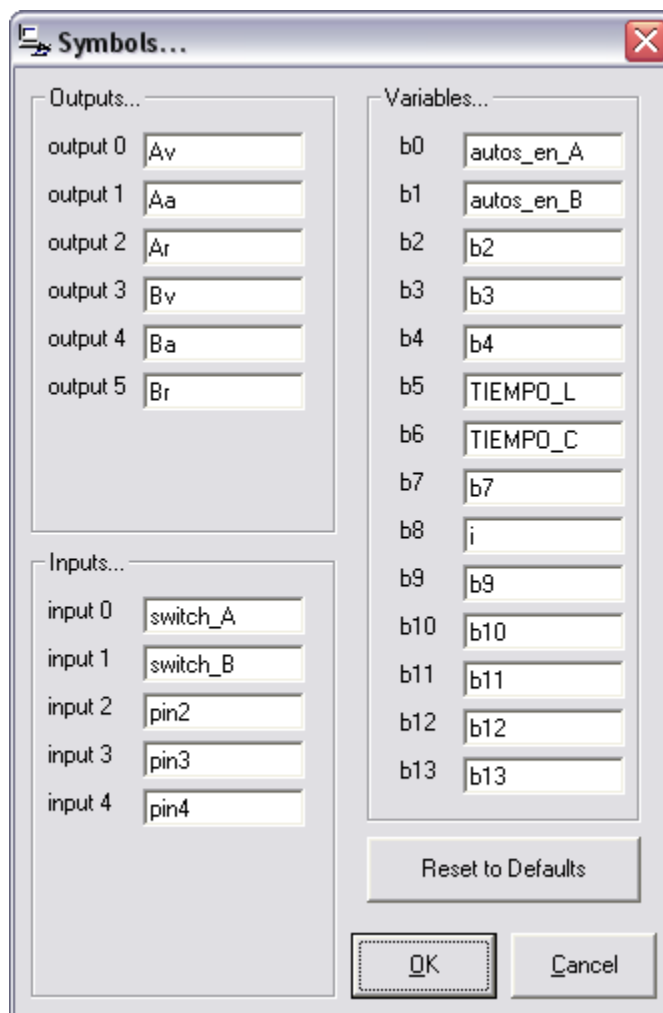
En los primeros 2 manuales (incluidos en el *Programming Editor*) se explica con detalle cómo programar utilizando estos métodos.

El programa del Semáforo Inteligente se realizó con ambos métodos. Primero se hizo con diagramas de flujo, después se convirtió a BASIC y por último se corrigieron algunos errores que presentó el programa tras ser convertido, además se cambiaron los nombres de las etiquetas nombradas automáticamente.

### **Programa con Diagramas de Flujo**



Al programar con diagramas de flujo existe la opción de renombrar las entradas y salidas del PICAXE, así como las variables “b” desde una sola ventana, en lugar de tener que escribir el comando symbol para cada variable. Esto se hace seleccionando en el menú: Flowchart> Flowchart Symbol Table. En la siguiente imagen se muestran los cambios realizados para el programa del Semáforo Inteligente.



## Programa en BASIC

```

'SEMAFORO INTELIGENTE

'Av = 0  (output 0)  Av es la luz verde de A
'Aa = 1  (output 1)  Aa es la luz amarilla de A
'Ar = 2  (output 2)  Ar es la luz roja de A
'Bv = 3  (output 3)  Bv es la luz verde de B
'Ba = 4  (output 4)  Ba es la luz amarilla de B
'Br = 5  (output 5)  Br es la luz roja de B

symbol autos_en_A = b0      'Para guardar la cantidad contada de autos en A
symbol autos_en_B = b1      'Para guardar la cantidad contada de autos en B
symbol TIEMPO_L = w5 'Tiempo Largo: es la duraciOn de las luces verdes y rojas
                        'de A o B
symbol TIEMPO_C = w6 'Tiempo Corto: es la duraciOn de las luces amarillas
                        'de A o B

symbol i = b8

Main:
    let TIEMPO_L=5000 'Valor de prueba. En realidad debe valer
                        'aprox. 50 segundos
    let TIEMPO_C=1500 'Valor de prueba. En realidad debe valer
                        'aprox. 4 segundos

Salto_1: let i=1
        let pins = %00100001 'Enciende Av y Br, los demAs apagados
        gosub CONTADOR
        let i= 2

Salto_2: if autos_en_A<autos_en_B then Salto_3
        let pins = %00100001 'Enciende Av y Br, los demAs apagados
        gosub CONTADOR
        if i= 3 then Salto_3
        let i=i+1
        goto Salto_2

Salto_3: let pins = %00100010 'Enciende Aa y Br, los demAs apagados
        gosub AMARILLO_A_VERDE
        let pins = %00001100 'Enciende Ar y Bv, los demAs apagados
        let i= 1

Salto_4: if autos_en_B<autos_en_A then Salto_5
        let pins = %00001100 'Enciende Ar y Bv, los demAs apagados
        gosub CONTADOR
        if i= 3 then Salto_5
        let i=i+1
        goto Salto_4

Salto_5: let pins = %00010100 'Enciende Ar y Ba, los demAs apagados
        gosub AMARILLO_A_VERDE
        goto Salto_1

CONTADOR:
    count 0,TIEMPO_L,w0 'Cuenta los pulsos en el pin 0 (input)
                        'que estA conectado al switch A
    count 1,TIEMPO_L,w1 'Cuenta los pulsos en el pin 1 (input)
                        'que estA conectado al switch B

    let autos_en_A=w0
    let autos_en_B=w1
    pause TIEMPO_L

```

```

        return

AMARILLO_A_VERDE:
    pause TIEMPO_C
    return

```

Antes de descargar el programa (en BASIC o en diagramas de flujo) al PICAXE 14M se puede simular (seleccionando “Simulate” en los botones del menú).

### 3 – Descarga del programa en el PICAXE-14M y prueba de funcionamiento.

- Conectar el cable serial o USB para PICAXE a la PC.
- Iniciar el software *Programming Editor* (clic en Inicio>Programas>Revolution Education>Programming Editor). Después, en el menú hacer clic en View>Options para mostrar la ventana de opciones (aunque podría aparecer automáticamente en el inicio). En la pestaña “Mode” seleccionar el tipo correcto de PICAXE (14M). En la pestaña “Ports” seleccionar el puerto serial COM en el que se conectó el cable serial (en caso de usar el cable USB consultar la ayuda que se muestra en esa pestaña).
- Colocar el jumper en los dos pines más cercanos al jack stereo del circuito.
- Conectar el cable al jack stereo.
- Conectar 4.5 V (3 pilas AA) o una fuente de poder a 5 V al bloc de 2 terminales del circuito.
- En el menú de *Programming Editor* seleccionar PICAXE>Program para descargar el programa al 14M. Cuando la descarga finalice el programa se ejecutará inmediatamente en el 14M.
- El cable puede ser desconectado del jack stereo
- Colocar el jumper en los dos pines más alejados al jack stereo.

Tras este último paso los LEDS del circuito deberán prenderse en el orden previsto.

## BIBLIOGRAFÍA Y REFERENCIAS

Manuales 1 y 2 de los PICAXE. Vienen incluidos en Programming Editor (menú>Help) pero también pueden descargarse desde: <http://www.rev-ed.co.uk/PICAXE/es/index.htm>

PALAZZESI, A. *Como construir tus propios PCB*. [En línea]. 19 de septiembre de 2008. Disponible en: [http://www.ucontrol.com.ar/wiki/index.php/Como\\_construir\\_tus\\_propios\\_PCB](http://www.ucontrol.com.ar/wiki/index.php/Como_construir_tus_propios_PCB) [Consultado el 2 de diciembre de 2009]

Software empleado:

PCB Wizard 3.5. Web: <http://www.new-wave-concepts.com/pr/wizard.html>

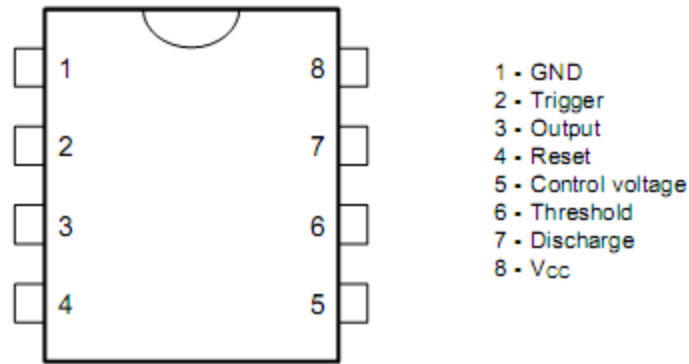
Programming Editor 5.2.7. Web: <http://www.rev-ed.co.uk/PICAXE/progedit.htm>

Windows XP SP3. Web: <http://www.microsoft.com/latam/windowsxp/default.msp>

## APÉNDICE



## Distribución de pines del temporizador 555



*Distribución de pines en un NE555N de STMicroelectronics.*

### 1 - Tierra o masa.

**2 - Disparo:** En esta patilla se establece el inicio del tiempo de retardo, si el 555 es configurado como monostable.

**3 - Salida:** Aquí veremos el resultado de la operación del temporizador 555, ya sea que esté conectado como monostable, astable u otro. Cuando la salida es alta, el voltaje de salida es el voltaje de aplicación ( $V_{cc}$ ) menos 1.7 V. Esta salida se puede obligar a estar en casi 0 V con la ayuda de la patilla # 4 (reset).

**4 - Reset:** Si se pone a un nivel por debajo de 0.7 V, pone la patilla de salida # 3 a nivel bajo. Si por algún motivo esta patilla no se utiliza hay que conectarla a  $V_{cc}$  para evitar que el 555 se "resetee".

**5 - Control de voltaje:** Cuando el temporizador 555 se utiliza en el modo de controlador de voltaje, el voltaje en esta patilla puede variar casi desde  $V_{cc}$  (en la practica como  $V_{cc} - 1$  V) hasta casi 0 V (en la practica aprox. 2 V). Así es posible modificar los tiempos en que la patilla # 3 esta en alto o en bajo independiente del diseño (establecido por las resistencias y condensadores conectados externamente al 555).

El voltaje aplicado a la patilla # 5 puede variar entre un 45% y un 90 % de  $V_{cc}$  en la configuración monostable.

Cuando se utiliza la configuración astable, el voltaje puede variar desde 1.7 volts hasta  $V_{cc}$ . Modificando el voltaje en esta patilla en la configuración astable causará la frecuencia original del astable sea modulada en frecuencia (FM).

Si esta patilla no se utiliza, se recomienda ponerle un condensador de 0.01 $\mu$ F para evitar las interferencias.

**6 - Umbral:** Es una entrada a un comparador interno que tiene el 555 y se utiliza para poner la salida (Pin # 3) a nivel bajo.

**7 - Descarga:** Utilizado para descargar con efectividad el condensador externo utilizado por el temporizador para su funcionamiento.

**8 - V+:** También llamado  $V_{cc}$ , es el pin donde se conecta el voltaje de alimentación que va de 4.5

V hasta 16 V (máximo). Hay versiones militares de este integrado que llegan hasta 18 V.

## Distribución de pines del 74143

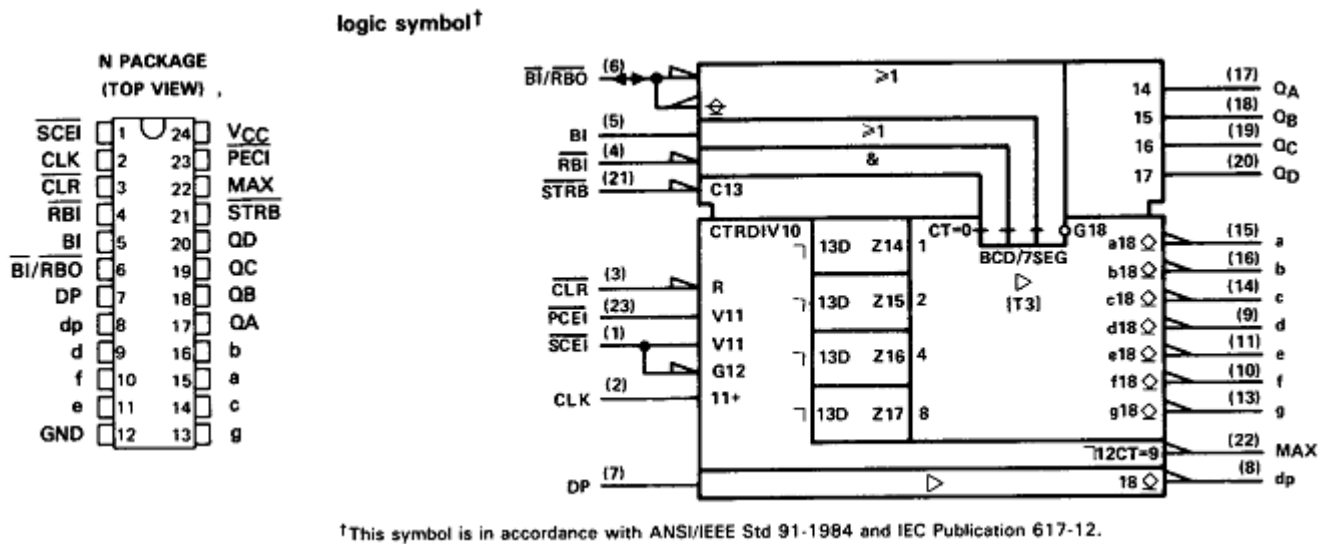
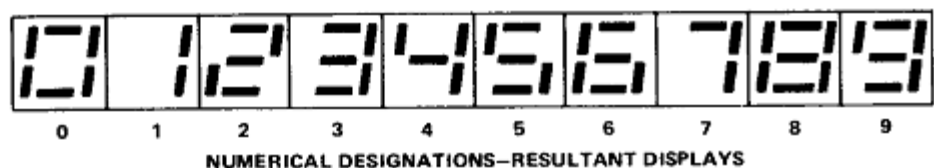


Tabla 5. Descripción de las funciones de los pines del 74143.

FUNCIÓN	PIN NÚM.	DESCRIPCIÓN
CLEAR INPUT	3	En nivel bajo reinicia y mantiene al contador en cero. Debe estar en nivel alto para un conteo normal.
CLOCK INPUT	2	Cada transición positiva (de nivel bajo a alto) incrementa el contador si el circuito está en modo normal de conteo (PCEI y SCEI en nivel bajo y clear en nivel alto).
PARALLEL COUNT ENABLE INPUT (PCEI)	23	Debe estar en nivel bajo para el modo normal de conteo. En nivel alto el contador será deshabilitado. El nivel lógico no cambiará cuando el reloj (clock) esté en nivel bajo.
SERIAL COUNT ENABLE INPUT (SCEI)	1	Debe estar en nivel bajo para el modo normal de conteo, también debe estar en este nivel para que la salida de conteo máximo (maximum count output) esté en nivel bajo. En nivel alto el contador será deshabilitado y la salida de conteo máximo estará en nivel alto. El nivel lógico no cambiará cuando el reloj esté en nivel bajo.

MAXIMUM COUNT OUTPUT	22	Bajará de nivel cuando el contador esté en 9 y SCEI esté en nivel bajo. Regresará a nivel alto cuando el contador cambie a 0 y permanecerá en este nivel durante el conteo de 1 a 8. Permanecerá en nivel alto (inhabilitado) mientras SCEI esté en nivel alto.
LATCH STROBE INPUT	21	En nivel bajo, el latch sigue los datos del contador. En nivel alto, el latch guarda estos datos y los mantiene constantes y el contador puede trabajar independientemente.
LATCH OUTPUTS (Q <sub>A</sub> , Q <sub>B</sub> , Q <sub>C</sub> , Q <sub>D</sub> )	17, 18, 19, 20	Los datos BCD que maneja el decodificador pueden ser guardados en el latch de 4 bits y están disponibles en estas salidas para utilizarse con otro circuito o procesador. Los pesos binarios de estas salidas son: Q <sub>A</sub> =1, Q <sub>B</sub> =2, Q <sub>C</sub> =4, Q <sub>D</sub> =8.
DECIMAL POINT INPUT	7	Debe Estar en nivel alto para que se muestre el punto decimal. El punto decimal no se muestra cuando esta entrada está en nivel bajo o cuando el display está en blanco (blanked).
BLANKING INPUT (BI)	5	En nivel alto, pondrá en blanco (apagará) el display y forzará a RBO a nivel bajo. Debe estar en nivel bajo para que el display trabaje correctamente. Puede ser pulsado para implementar un control de intensidad del display.
RIPPLE-BLANKING INPUT (RBI)	4	En nivel bajo, si los datos en el latch son BCD 0, apagará el display y forzará a RBO a nivel bajo. Esta entrada no tiene efecto si los datos en el latch son distintos de 0.
RIPPLE-BLANKING OUTPUT (RBO)	6	
LED/LAMP DRIVER OUTPUTS (a, b, c, d, e, f, dp)	15, 16, 14, 9 11, 10, 13, 8	Salidas para utilizar lámparas o LEDs de siete segmentos y sus puntos decimales. Ver más adelante la identificación de cada segmento y su representación en el display.



# Distribución de pines del comparador 7485

SN5485, SN54LS85, SN54S85 . . . J OR W PACKAGE  
SN7485 . . . N PACKAGE  
SN74LS85, SN74S85 . . . D OR N PACKAGE  
(TOP VIEW)

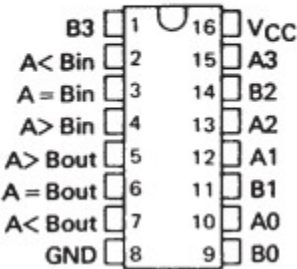


Tabla 6. Tabla de verdad del comparador 7485.

FUNCTION TABLE									
COMPARING INPUTS				CASCADING INPUTS			OUTPUTS		
A3, B3	A2, B2	A1, B1	A0, B0	A > B	A < B	A = B	A > B	A < B	A = B
A3 > B3	X	X	X	X	X	X	H	L	L
A3 < B3	X	X	X	X	X	X	L	H	L
A3 = B3	A2 > B2	X	X	X	X	X	H	L	L
A3 = B3	A2 < B2	X	X	X	X	X	L	H	L
A3 = B2	A2 = B2	A1 > B1	X	X	X	X	H	L	L
A3 = B3	A2 = B2	A1 < B1	X	X	X	X	L	H	L
A2 = B3	A2 = B2	A1 = B1	A0 > B0	X	X	X	H	L	L
A3 = B3	A2 = B2	A1 = B1	A0 < B0	X	X	X	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	H	L	L	H	L	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	L	H	L	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	X	X	H	L	L	H
A3 = B3	A2 = B2	A1 = B1	A0 = B0	H	H	L	L	L	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	L	L	L	H	H	L