

# Una Introducción a los Circuitos Digitales

Ing. Arturo Miguel de Priego Paz Soldán

<http://www.tourdigital.net>

[amiguel@pucp.edu.pe](mailto:amiguel@pucp.edu.pe)

Setiembre de 2014

Esta guía presenta el proceso de diseño digital a través de un ejemplo. Muestra cómo se analiza, especifica, diseña y simula un controlador de las luces de un semáforo. Conocerás las funciones lógicas más simples, contarás desde 0 hasta 15 en el sistema de numeración binario, y construirás unos circuitos lógicos para controlar las luces de un semáforo. Finalmente, usarás circuitos integrados para construir el semáforo.

Comencemos por analizar cómo funciona un semáforo. En la siguiente tabla puedes ver los estados de un semáforo típico. Este semáforo tiene cuatro estados o combinaciones diferentes de luces. Necesita 12 faros, pero los faros que apuntan en direcciones opuestas siempre muestran el mismo estado, por lo tanto, solamente debemos controlar 6 luces, tres para cada sentido (Norte-Sur y Este-Oeste). Las señales que controlan el encendido y apagado de las luces se llaman salidas del circuito. Nota que el semáforo tiene cuatro estados que duran unos períodos definidos (T1, T2, T3, T4).

	Sentido Norte Sur			Sentido Este Oeste			Período
	Rojo	Ámbar	Verde	Rojo	Ámbar	Verde	
	ON	OFF	OFF	OFF	OFF	ON	T1
	ON	OFF	OFF	OFF	ON	OFF	T2
	OFF	OFF	ON	ON	OFF	OFF	T3
	OFF	ON	OFF	ON	OFF	OFF	T4

Cada faro puede tener su luz encendida (ON) o apagada (OFF). Cuando un faro esté encendido representaremos ese estado con un 1 (ON), y cuando esté apagado con un 0 (OFF).



apagado = OFF = 0



encendido = ON = 1

Un bit es un tipo de número que puede tener sólo dos valores: 0 y 1. Entonces las salidas de control para cada faro pueden codificarse con seis bits en total:

Sentido Norte Sur			Sentido Este Oeste			Período
Rojo	Ámbar	Verde	Rojo	Ámbar	Verde	
1	0	0	0	0	1	T1
1	0	0	0	1	0	T2
0	0	1	1	0	0	T3
0	1	0	1	0	0	T4

Cuando un bit está en 1 indica que la luz de ese faro debe encenderse en el período correspondiente, y cuando el bit esté en 0 la luz del faro deberá apagarse.

Todo circuito digital tiene bits de entrada y bits de salida. Los bits de las luces de los semáforos son bits de salida, porque son bits que el circuito lógico envía al exterior. Los bits de entrada son bits que el circuito recibe del exterior. Muchas veces los bits se agrupan en grupos de bits llamados buses para representar cantidades mayores a 0 y 1. Por ejemplo, un byte es el nombre que se le da a un grupo de 8 bits.

Ahora vamos a darles nombres a los bits de salida. Mirando la tabla previa, vamos a asignar de izquierda a derecha los siguientes nombres: RojoNS, AmbarNS, VerdeNS, RojoEO, AmbarEO, VerdeEO. El sufijo NS hace referencia al sentido Norte-Sur y el sufijo EO al sentido Este-Oeste. Observa que:

- ❑ RojoNS es 1 cuando el período es T1 o T2 y es 0 en los períodos T3 y T4.
- ❑ AmbarNS es 1 cuando el período es T4, y es 0 en el resto de períodos.
- ❑ VerdeNS es 1 cuando el período es T3, y es 0 en el resto de períodos.
- ❑ RojoEO es 1 cuando el período es T3 o T4, y es 0 en los períodos T1 y T2.
- ❑ AmbarEO es 1 cuando el período es T2, y es 0 en el resto de períodos.
- ❑ VerdeEO es 1 cuando el período es T1, y es 0 en el resto de períodos.

Como los bits solamente pueden tener dos estados basta con escribir estas relaciones del siguiente modo (asumiendo que si la condición para que sea 1 no se cumple, entonces se cumple la condición para que sea 0):

- ❑ RojoNS es 1 cuando el período es T1 o T2.
- ❑ AmbarNS es 1 cuando el período es T4.
- ❑ VerdeNS es 1 cuando el período es T3.
- ❑ RojoEO es 1 cuando el período es T3 o T4.
- ❑ AmbarEO es 1 cuando el período es T2.
- ❑ VerdeEO es 1 cuando el período es T1.

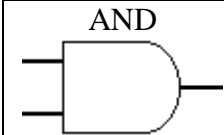
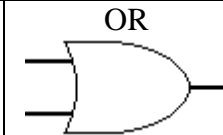
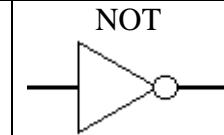
Existe una manera formal para indicar estas relaciones si usamos las siguientes convenciones:

- ❑ Si para que ocurra el evento F es necesario que simultáneamente el evento A y el evento B ocurran, entonces decimos que  $F = A \text{ AND } B$ .
- ❑ Si para que ocurra el evento F es necesario que el evento A o el evento B ocurran, entonces decimos que  $F = A \text{ OR } B$ . Observa que si ambos eventos ocurren también ocurre F.
- ❑ Si para que ocurra el evento F es necesario que no ocurra el evento A, entonces decimos que  $F = \text{NOT } A$ .
- ❑ Si para que ocurra el evento F es necesario que ocurra el evento A, entonces decimos que  $F = A$ .

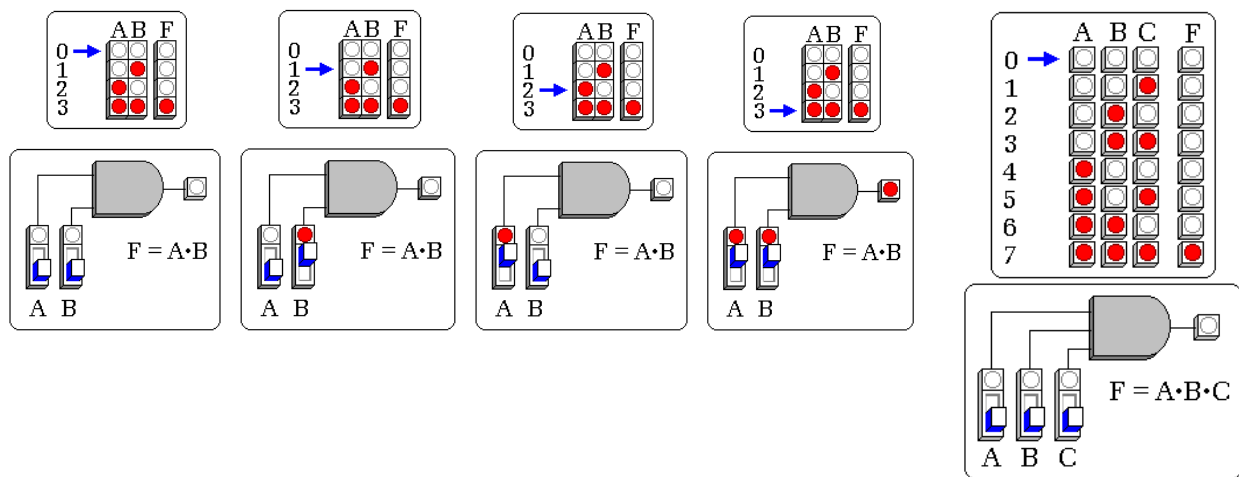
De acuerdo a esto, si representamos con 1 cuando ocurre un evento y con 0 cuando no ocurre, podemos transformar las sentencias anteriores en nuevas sentencias equivalentes:

RojoNS es 1 cuando el período es T1 o T2	$\text{RojoNS} = T1 \text{ OR } T2$
AmbarNS es 1 cuando el período es T4	$\text{AmbarNS} = T4$
VerdeNS es 1 cuando el período es T3	$\text{VerdeNS} = T3$
RojoEO es 1 cuando el período es T3 o T4	$\text{RojoEO} = T3 \text{ OR } T4$
AmbarEO es 1 cuando el período es T2	$\text{AmbarEO} = T2$
VerdeEO es 1 cuando el período es T1	$\text{VerdeEO} = T1$

Las operaciones AND, OR y NOT son las funciones básicas de la lógica matemática, y juegan un papel fundamental en el diseño de los circuitos integrados. En el diseño lógico estas funciones tienen símbolos que facilitan su identificación en un diagrama esquemático (en los diagramas usaremos símbolos realzados para una presentación diferente):

AND 	OR 	NOT 
Función Y	Función O	Función No
Producto	Suma	Complemento
. (punto)	+ (más)	' (apóstrofe) - (raya superior)

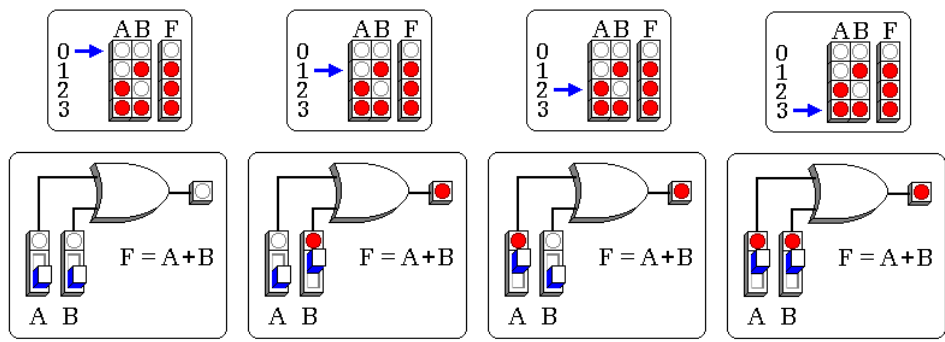
Puedes entender mejor como trabaja la función AND observando las siguientes figuras. Nota que un led rojo indica encendido y uno blanco indica apagado. Presta atención a la posición de los interruptores: la posición abajo lo desconecta e indica 0 (OFF) y la posición arriba lo conecta e indica 1 (ON).



La función AND de dos entradas tiene la salida en 1 solamente cuando ambas entradas están en 1, y en otro caso la salida es 0.

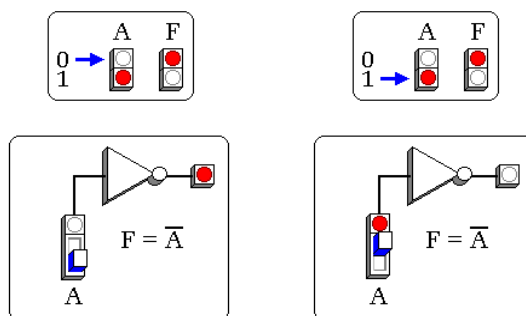
Ahora, observa cómo se comporta la puerta AND de tres entradas. Intenta deducir la regla de comportamiento para esta AND de tres entradas y deduce la regla para una AND de cualquier número de entradas.

Para el caso de la función OR ocurre algo similar:

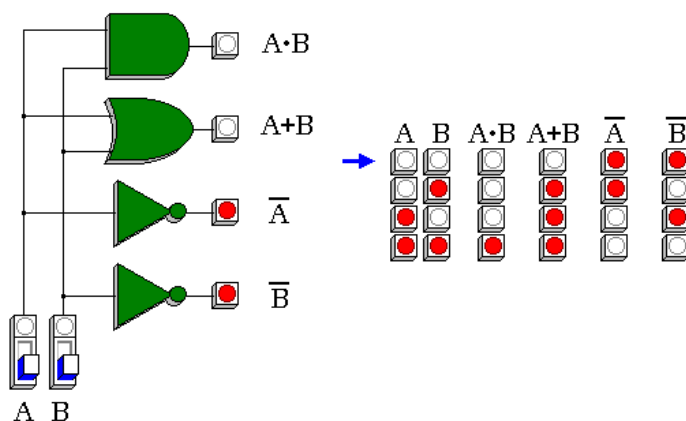


La función OR de dos entradas tiene la salida en 1 si cualquier entrada está en 1, y en otro caso la salida es 0 (cuando ambas entradas están en 0). ¿Qué sucede en una OR de más de dos entradas?

Para la función NOT tenemos:



Ahora, compara estas tres funciones (AND, OR, NOT) analizando la siguiente figura:



En un circuito lógico existen entradas (interruptores por ejemplo), salidas (indicadores luminosos o leds por ejemplo) y circuitos que realicen funciones lógicas (AND, OR, NOT, y otras más). Para el diseño digital existen más tipos de entradas, más tipos de salidas y una gran variedad de funciones lógicas.

















Volviendo al diseño del controlador de luces del semáforo, intenta responder esta pregunta: ¿T1, T2, T3 y T4 son entradas, funciones o salidas? La respuesta es que no son ni entradas ni salidas. Los períodos de tiempo en este caso son resultado de estados internos del controlador de las luces.

Un estado en un circuito lógico queda determinado por las entradas actuales del circuito y por su estado anterior, es decir, por las condiciones de funcionamiento pasadas. En el caso del semáforo hemos definido cuatro estados: T1, T2, T3 y T4.

Si fueras policía de tránsito y tuvieras que controlar manualmente el semáforo apretando unos botones, ¿cómo harías para saber cuánto tiempo dejas al semáforo en cada estado? Asumiendo que el tráfico es similar en ambos sentidos, quizás tomarías tu reloj y asignarías intervalos de tiempos en segundos, por ejemplo, 25 segundos para T1 y T3, y 5 segundos para T2 y T4. Eso haría que el ciclo del semáforo se repita cada minuto. En nuestro semáforo particular vamos a hacer que el ciclo dure 16 segundos.

Los circuitos digitales emplean fácilmente el sistema binario para contar. El sistema binario es el sistema de numeración en base 2. Nosotros estamos familiarizados con el sistema decimal, el que usamos para contar con los dígitos del 0 al 9. En el sistema binario contamos con bits, con dígitos del 0 al 1.

Para que puedas comparar ambos sistemas, en la siguiente tabla puedes ver la cuenta binaria para una cuenta decimal equivalente desde 0 hasta 15 (o sea, para 16 cuentas).

	Cuenta Binaria	Cuenta Decimal
	q <sub>3</sub> q <sub>2</sub> q <sub>1</sub> q <sub>0</sub>	
	0000	0
	0001	1
	0010	2
	0011	3
	0100	4
	0101	5
	0110	6
	0111	7
	1000	8
	1001	9
	1010	10
	1011	11
	1100	12
	1101	13
	1110	14
	1111	15

Esta tabla nos permitirá hallar una relación entre los estados del semáforo, los períodos de tiempo, las entradas y salidas para el circuito lógico.

Los bits de la cuenta binaria los vamos a representar con q<sub>3</sub>, q<sub>2</sub>, q<sub>1</sub> y q<sub>0</sub>. El bit q<sub>3</sub> es el más significativo, ubicado más a la izquierda del número. El bit q<sub>0</sub> es el menos significativo, ubicado más a la derecha del número. Estos bits van a cambiar automáticamente cada segundo, es decir, cada cuenta va a permanecer activa por un segundo y luego el contador hará que la cuenta se incremente. Cuando la cuenta sea 15 la cuenta siguiente será 0 y el ciclo se repetirá.

A continuación, vamos a dividir los 16 segundos en cuatro períodos. A T<sub>1</sub> y T<sub>3</sub> le daremos 7 segundos y a T<sub>2</sub> y T<sub>4</sub> le vamos a dar 1 segundo.

¿Tienes una idea de cómo relacionar estos tiempos con el contador? La forma más sencilla de hacerlo es así:

Cuando el contador está:	El período es:
En el rango de 0 a 6	T <sub>1</sub>
en 7	T <sub>2</sub>
En el rango de 8 a 14	T <sub>3</sub>
En 15	T <sub>4</sub>

En la tabla siguiente juntamos todo, la cuenta binaria en cada período y las luces de salida:

Cuenta Decimal	Cuenta Binaria	Período	RojoNS	AmbarNS	VerdeNS	RojoEO	AmbarEO	VerdeEO
	q <sub>3</sub> q <sub>2</sub> q <sub>1</sub> q <sub>0</sub>							
0	0 0 0 0	T1	1	0	0	0	0	1
1	0 0 0 1		1	0	0	0	0	1
2	0 0 1 0		1	0	0	0	0	1
3	0 0 1 1		1	0	0	0	0	1
4	0 1 0 0		1	0	0	0	0	1
5	0 1 0 1		1	0	0	0	0	1
6	0 1 1 0		1	0	0	0	0	1
7	0 1 1 1	T2	1	0	0	0	1	0
8	1 0 0 0	T3	0	0	1	1	0	0
9	1 0 0 1		0	0	1	1	0	0
10	1 0 1 0		0	0	1	1	0	0
11	1 0 1 1		0	0	1	1	0	0
12	1 1 0 0		0	0	1	1	0	0
13	1 1 0 1		0	0	1	1	0	0
14	1 1 1 0		0	0	1	1	0	0
15	1 1 1 1	T4	0	1	0	1	0	0

Ahora haremos que los bits de salida del circuito pasen a ser función de los bits de salida de un contador binario. Presta atención a cada bit de la cuenta binaria y a las luces: ¿Cuándo se prende la luz RojoNS? Cuando q<sub>3</sub> está apagado, que es cuando ocurren los períodos T1 o T2, tal como habíamos visto antes. Es decir, la función:

$$\text{RojoNS} = \text{T1 OR T2}$$

podemos cambiarla por la función:

$$\text{RojoNS} = \text{NOT } q_3$$

Mira lo que pasa con VerdeEO. Presta atención solamente a la combinación de bits de la cuenta binaria que hacen que VerdeEO sea 1. ¿Puedes hallar una relación?

Para que VerdeEO sea 1 q<sub>3</sub> debe ser 0 pero además los bits restantes no pueden ser 1 al mismo tiempo. En cualquier otra situación VerdeEO es 0. Es decir, VerdeEO se enciende cuando q<sub>3</sub> no se enciende y además q<sub>2</sub>, q<sub>1</sub> y q<sub>0</sub> tampoco se encienden al mismo tiempo. Eso podemos representarlo así:

$$\text{VerdeEO} = (\text{NOT } q_3) \text{ AND } (\text{NOT } (q_2 \text{ AND } q_1 \text{ AND } q_0))$$

De manera similar, intenta deducir lo siguiente:

$$\text{RojoEO} = q_3$$

$$\text{AmbarEO} = (\text{NOT } q_3) \text{ AND } (q_2 \text{ AND } q_1 \text{ AND } q_0)$$

$$\text{VerdeNS} = (q_3) \text{ AND } (\text{NOT } (q_2 \text{ AND } q_1 \text{ AND } q_0))$$

$$\text{AmbarNS} = (q_3) \text{ AND } (q_2 \text{ AND } q_1 \text{ AND } q_0)$$

Notarás que la expresión  $q_2 \text{ AND } q_1 \text{ AND } q_0$  se repite en cuatro casos. Siendo así, resulta conveniente asignar esta expresión a una función X, con lo cual tendremos:

$$\text{RojoNS} = \text{NOT } q_3$$

$$\text{RojoEO} = q_3$$

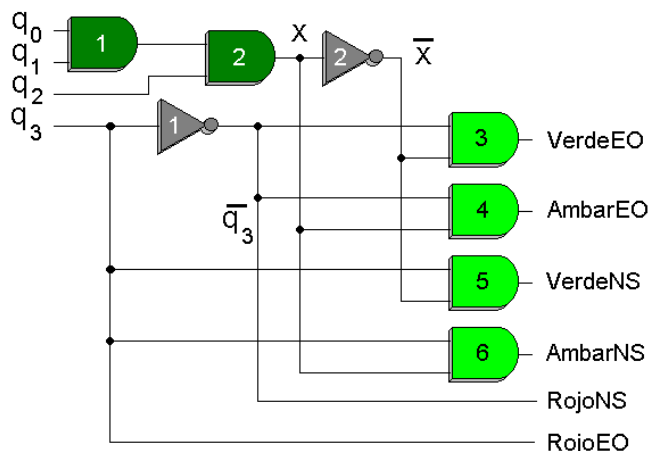
$$X = q_2 \text{ AND } q_1 \text{ AND } q_0$$

$$\text{VerdeEO} = (\text{NOT } q_3) \text{ AND } (\text{NOT } X)$$

$$\text{AmbarEO} = (\text{NOT } q_3) \text{ AND } (X)$$

$$\text{VerdeNS} = (q_3) \text{ AND } (\text{NOT } X)$$

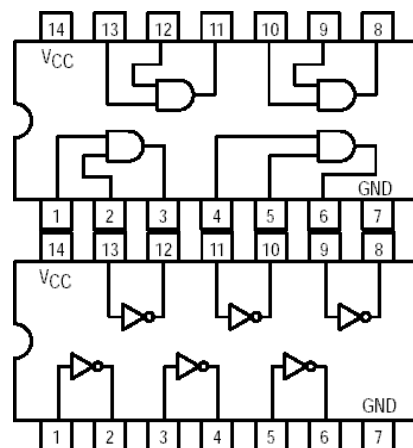
$$\text{AmbarNS} = (q_3) \text{ AND } (X)$$



A partir de estas expresiones podemos hallar los diagramas lógicos (comprueba que cada función esté representada por el diagrama lógico correspondiente).

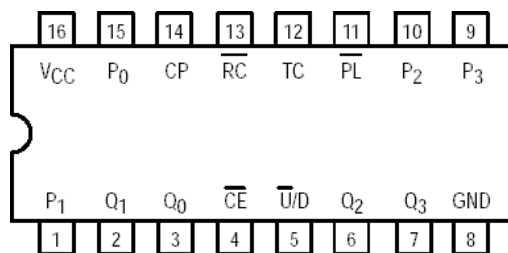
Para construir este circuito lógico se necesitan circuitos integrados (chips) que realicen las funciones lógicas que hemos determinado. En este caso vamos a usar un 74LS191 para contar de 0 a 15 cíclicamente, dos 74LS08 para las puertas AND, y un 74LS04 para las puertas NOT. Todos los chips deben ser polarizados para funcionar correctamente, es decir, deben conectarse a una batería o alimentación de voltaje. Estos chips operan con voltajes de 4.5V a 5V. Comenzando con el pin inferior izquierdo, los pines se cuentan desde 1 hasta el último siguiendo el sentido contrario a las manecillas del reloj.

Observa la figura a la derecha. Un 7408 tiene cuatro puertas en el chip, cada una puede realizar una función AND de dos entradas. El pin de VCC es el 14, el de GND el 7. El pin de VCC debe conectarse con el terminal positivo de la batería y el pin de GND con el terminal negativo. En los demás chips debe hacerse lo mismo.



Un circuito integrado 74LS04 tiene 6 puertas NOT. El pin de VCC es el 14, el de GND el 7.

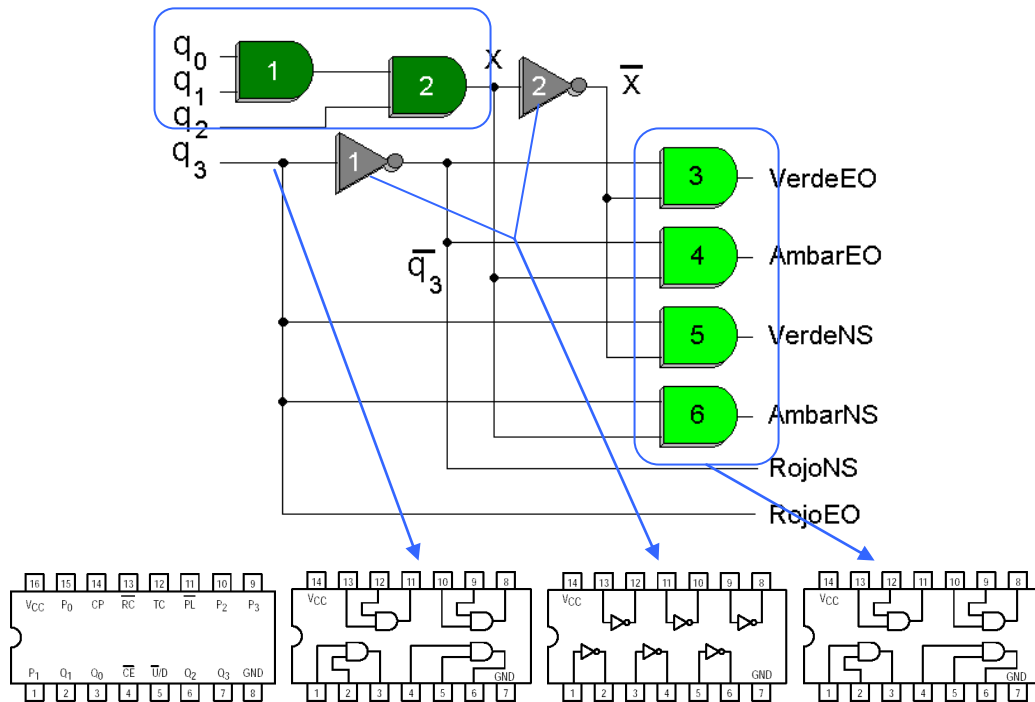
El circuito integrado 74LS191 es un contador binario de 4 bits. El contador binario provee las señales  $q_3, q_2, q_1$  y  $q_0$ . El 74191 tiene 16 pines, el pin 16 se conecta a VCC y el pin 8 a GND. Los pines 4 y 5 los conectaremos a GND para que el contador realice cuentas en sentido ascendente. Los bits  $q_3, q_2, q_1$  y  $q_0$  se encuentran en los pines 7, 6, 2 y 3, respectivamente. El pin 14 se conecta a la señal de reloj, que en este caso será de una frecuencia de 1Hz. Esta señal de 1Hz es muy rápido para un semáforo normal. Con un circuito integrado LM555 puedes variar la frecuencia para hacer las cuentas más lentas (en otra sesión aprenderás a construir osciladores con este circuito).



Ahora veremos cómo ubicar las funciones lógicas en los circuitos integrados. Puedes elegir cualquier puerta lógica en el 74LS08 y 74LS04 para las puertas AND y NOT, respectivamente.

Para este diseño, vamos a hacer lo siguiente. Las AND 1 y 2 se arman en un 74LS08 y las AND 3, 4, 5 y 6 en el otro 74LS08. Las NOT 1 y 2 se construyen sobre un 74LS04.

En la siguiente figura verás las asignaciones:



Nota que los bits de salida del contador proveen las entradas para esta etapa de decodificación. En la siguiente figura se muestra un esquema del circuito construido en un simulador digital. Para observar la cuenta binaria, los bits  $q_3$ ,  $q_2$ ,  $q_1$  y  $q_0$  se han conectado a los leds amarillos. Este programa está en <http://www.tourdigital.net/SimuladorTTLconEscenarios.html>

