

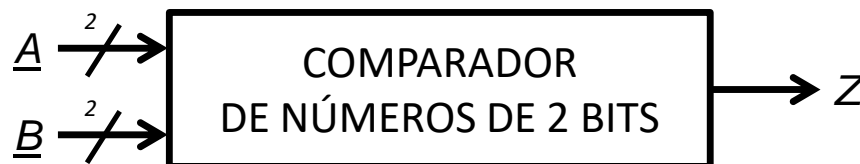


## FUNDAMENTOS DE COMPUTADORES

# PRÁCTICA 1: MONTAJE DE UN CIRCUITO COMBINACIONAL USANDO PUERTAS LÓGICAS

El objetivo de esta práctica es diseñar y montar en el laboratorio un sistema combinacional que realice la comparación de 2 números binarios sin signo de 2 bits. El circuito tendrá los puertos mostrados en la figura:

- Una entrada, A, por la que se recibe un número binario sin signo de 2 bits.
- Una entrada, B, por la que se recibe otro número binario sin signo de 2 bits.
- Una salida binaria, Z, que tomará el valor lógico “1” cuando A > B, y “0” en cualquier otro caso.



### Desarrollo de la práctica

#### 1. Fase de diseño

- El diseño del circuito debe realizarse en casa y quedar reflejado en el cuadernillo correspondiente.
- El circuito debe implementarse únicamente usando puertas NAND.

#### 2. Fase de montaje y depuración

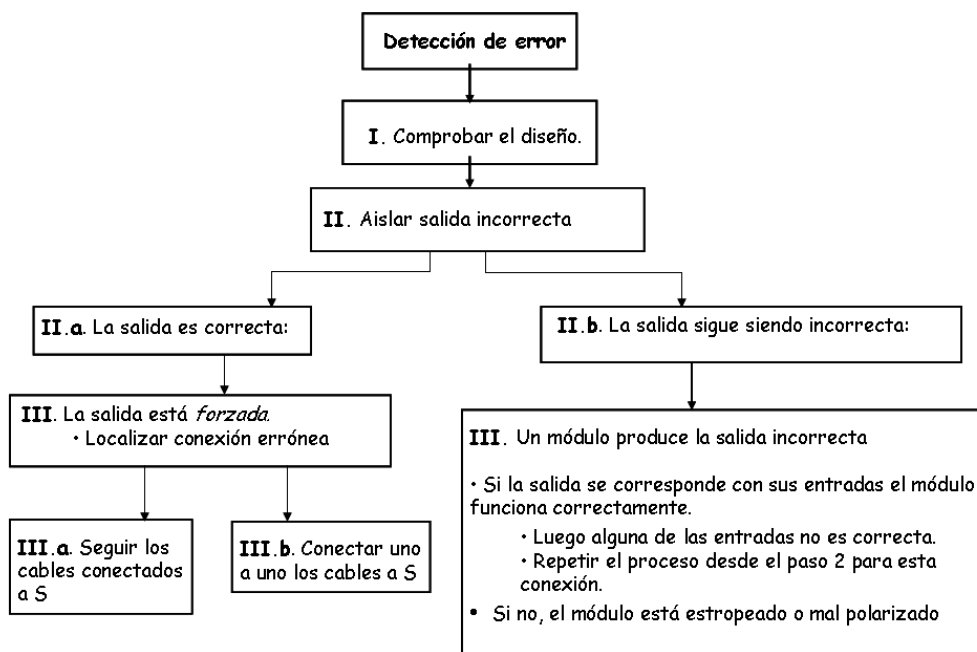
- Utilizar los siguientes chips para montar el circuito:
  - **7400**: 4 puertas NAND de 2 entradas.
  - **7410**: 3 puertas NAND de 3 entradas.
- Conectar todas las entradas a *switches* y todas las salidas a *leds*, según el orden usado en las tablas de verdad del cuadernillo.
- Una vez montado, **comprobar toda la tabla de verdad** del circuito para verificar que funciona correctamente. Si no es así, hay que depurarlo para encontrar los fallos y corregirlos.
- Cuando se haya comprobado que funciona correctamente, enseñar la práctica al profesor del laboratorio.

### ¿Cómo depurar un circuito combinacional?

Una vez montado el circuito, se encenderá el entrenador y se irán introduciendo mediante los switches cada una de las combinaciones de valores de entrada de la tabla de verdad, anotando sobre la propia tabla el resultado que se visualiza en los leds.

Si la tabla de verdad calculada por el circuito es la misma que la tabla de verdad fijada por la especificación, el circuito es correcto. En otro caso será incorrecto y deberá depurarse del siguiente modo:

1. Repasar la implementación lógica del circuito para descartar fallos en la fase de diseño. Si el diseño lógico es correcto, se debe pasar al punto 2.
2. Una vez que se está seguro de que el error se encuentra en la implementación física o en el montaje del circuito, se debe introducir al circuito la configuración binaria de entrada que produce la salida incorrecta. A continuación, se aísla la salida incorrecta S. Para ello se desconecta esta salida de todos los puntos del circuito donde estuviera conectada. En este momento pueden ocurrir dos cosas:
  - a. La salida S cambia y da el valor correcto. Luego esta salida seguramente esté conectada a otra salida del circuito que la está “forzando”. Pasar a la fase 3.
  - b. La salida S sigue dando un valor incorrecto, luego hay algún módulo en el árbol de cálculos de S que produce una salida incorrecta. Pasar a la fase 4.
3. La salida S está siendo “forzada” por la salida de otro módulo, está conectada a tierra o está conectada a alimentación. Para localizar donde está el error, hay dos procedimientos:
  - a. Seguir en la implementación de dónde viene cada uno de los cables conectados a la salida S.
  - b. Aislar la salida S e ir conectando, uno por uno, los cables a la salida. En el momento en el que el valor de la salida S vuelva a ser incorrecto se habrá identificado la conexión errónea. Corregir la conexión errónea.
4. Se ha localizado un módulo M que produce una salida incorrecta. Si la salida del módulo M es coherente con el valor de sus entradas, significa que el módulo está funcionando correctamente. Por tanto, alguna de las entradas debe tener un valor lógico incorrecto generado por la salida de un módulo anterior. Repítase el proceso a partir del paso 2 para esta nueva salida. Si la salida del módulo M no es coherente con el valor de las entradas, entonces la puerta no está funcionando correctamente debido a que el “chip” está mal polarizado o está estropeado.





## FUNDAMENTOS DE COMPUTADORES

# PRÁCTICA 2: DISEÑO Y MONTAJE DE UN SUMADOR BINARIO DE 2 BITS

El objetivo de esta práctica es diseñar y montar en el laboratorio un sumador binario de números de 2 bits. La práctica está dividida en dos partes:

- a) Diseñar un **sumador completo de un bit** e implementar el circuito usando puertas NAND y XOR.

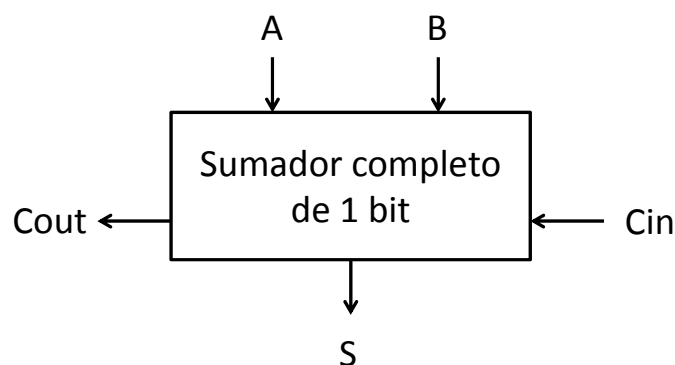


Figura 1: Sumador completo de un bit

- b) Diseñar un **sumador binario de números de 2 bits** e implementar el circuito encadenando dos sumadores completos de un bit (implementado cada uno a su vez con puertas NAND y XOR).

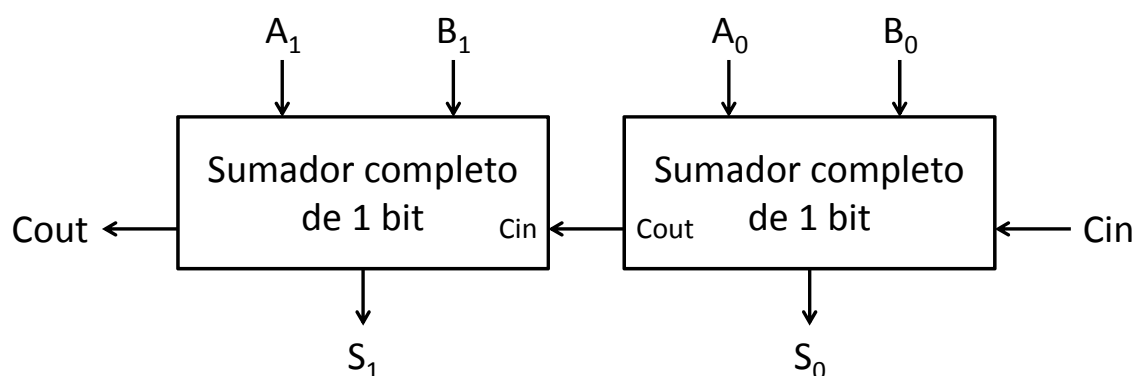


Figura 2: Sumador binario de números de 2-bits

### Desarrollo de la práctica

#### 1. Fase de diseño

- El diseño de los circuitos debe realizarse en casa y quedar reflejado en el cuadernillo correspondiente.

- Ambos sumadores deberán diseñarse con puertas NAND de 2 entradas y XOR de 2 entradas.

## 2. Fase de montaje y depuración

- Utilizar los siguientes chips para montar el circuito:
  - **7486**: 4 puertas XOR de 2 entradas.
  - **7400**: 4 puertas NAND de 2 entradas.
- Conectar todas las entradas a *switches* y todas las salidas a *leds*, según el orden usado en las tablas de verdad del cuadernillo.
- Una vez montado, **comprobar toda la tabla de verdad** del circuito para verificar que funciona correctamente. Si no es así, hay que depurarlo para encontrar los fallos y corregirlos.
- Cuando se haya comprobado que funciona correctamente, enseñar la práctica al profesor del laboratorio.



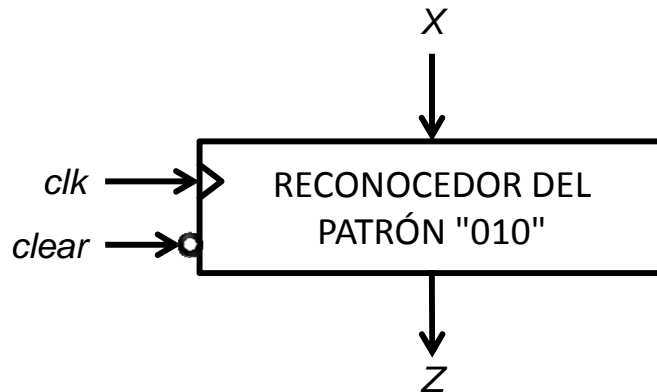
## FUNDAMENTOS DE COMPUTADORES

# PRÁCTICA 3: DISEÑO Y MONTAJE DE UN RECONOCEDOR DE PATRÓN

El objetivo de esta práctica es diseñar y montar en el laboratorio un circuito secuencial que reconozca el patrón “010”, modelando su comportamiento como una máquina de Moore.

Un circuito tendrá los puertos mostrados en la figura:

- Una entrada binaria,  $X$ , por la que reciben los bits en serie.
- Una salida binaria,  $Z$ , que tomará el valor lógico “1” cuando los últimos tres bits de entrada formen la secuencia “010”, y “0” en cualquier otro caso.
- Una entrada,  $clk$ , de reloj.
- Una entrada,  $clear$ , de inicialización asíncrona activa a baja.



### Desarrollo de la práctica

#### 1. Fase de diseño

- El diseño del circuito debe realizarse en casa y quedar reflejado en el cuadernillo correspondiente.
- El circuito deberá diseñarse como **máquina de Moore** usando biestables D, puertas NAND e inversores.

#### 2. Fase de montaje y depuración

- Utilizar los siguientes chips para montar el circuito:
  - **7474**: 2 biestables D (disparados por flanco de subida y señales de inicialización asíncrona a 0 ó 1 activas a baja).
  - **7400**: 4 puertas NAND de 2 entradas.
  - **7410**: 3 puertas NAND de 3 entradas.
  - **7404**: 6 puertas NOT.
- Conectar las entradas  $X$  y  $clear$  a *switches*.
- Conectar la entrada  $clk$  a un pulsador.
- Conectar la salida  $Z$  a un *led*.

- Comprobar que el circuito reconoce el patrón “010” con solapamiento, para ello hay que introducir por la entrada X una secuencia que presente varios patrones solapados en algún momento, como por ejemplo:

0 0 1 0 0 1 0 1 0 1 0 0 1 1

- Si el circuito no funciona correctamente hay que depurarlo para encontrar los fallos y corregirlos. Para depurar circuitos secuenciales hay que seguir los pasos explicados en el siguiente apartado.
- Cuando se haya comprobado que funciona correctamente, enseñar la práctica al profesor del laboratorio.

### Depuración de un circuito secuencial

A continuación se va a explicar un procedimiento para determinar de un modo sistemático a que son debidos los errores que se producen en el comportamiento de un sistema secuencial que se ha diseñado e implementado.

Para facilitar esta tarea es conveniente que los valores del estado actual (aunque NO SON SALIDAS del circuito) se conecten a los LEDs o al display para comprobar de manera rápida cual es la transición de estado que falla.

Si al comprobar el funcionamiento del circuito se detecta que el valor de una de las salidas no se corresponde con el esperado para el estado del circuito y la configuración binaria presente en las entradas, se debe buscar la causa del error. Para ello se procede del siguiente modo:

- i) Repasar el diseño que se está implementando buscando errores.
  - a. Si en la simulación el circuito funcionaba correctamente podemos asegurar que el error no se encuadra en el diseño.
- ii) Una vez que se está seguro que el error se encuentra en la implementación hardware o física del circuito, se debe analizar la secuencia de estados que sigue la máquina, para la secuencia de entradas en la que se ha detectado el error.
  - a. Si la secuencia es correcta entonces el error se encuentra en la red combinacional que produce la salida (SC1). Aplicar los pasos de depuración de una red combinacional sobre este módulo.
  - b. Si la secuencia no es correcta entonces falla o la red combinacional que genera el siguiente estado (SC2) o el conjunto de biestables. Pasar al punto iii.
- iii) Llevar el circuito al estado en el que se produce una transición incorrecta.

A continuación se plantea un ejemplo de cómo depurar un circuito secuencial cuando se observa un error en la secuencia de las salidas:

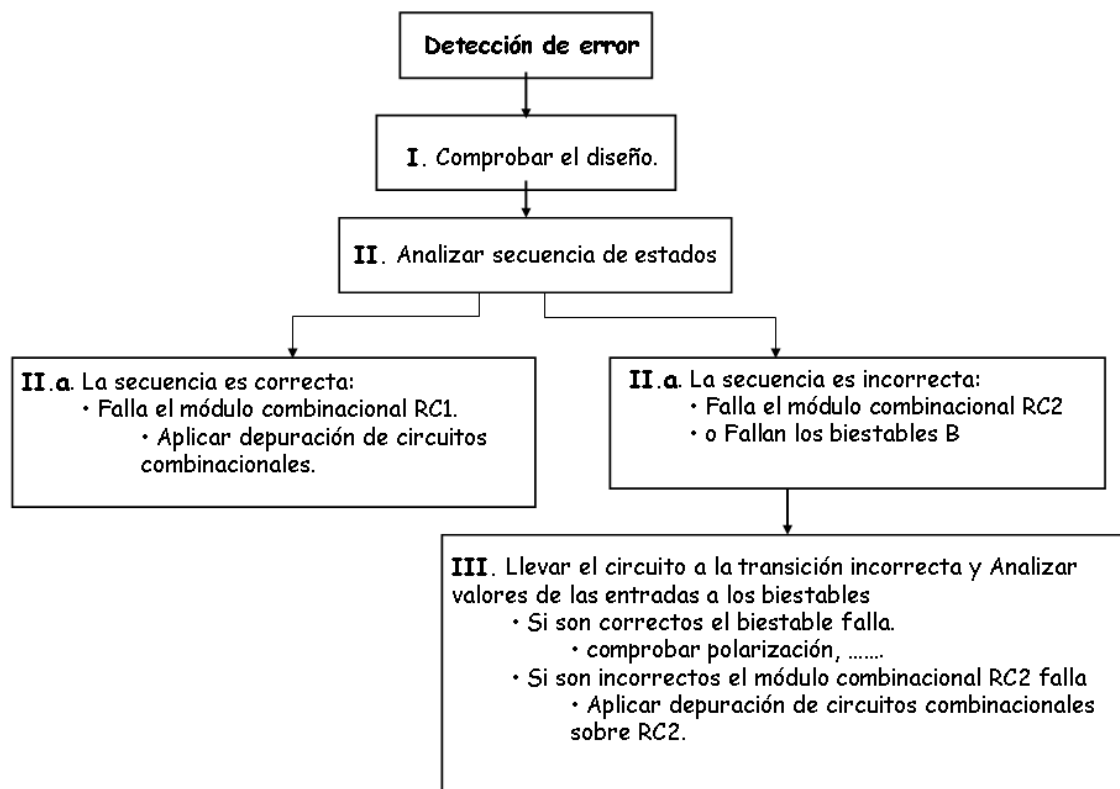
	Secuencia correcta				Secuencia observada		
	Q2	Q1	Q0		Q2	Q1	Q0
Estado inicial	0	1	0		0	1	0
Estado final	0	1	1		0	0	1

En este ejemplo se aprecia que el circuito que debería haber realizado la transición de estados  $2 \rightarrow 3$ , ha efectuado en realidad la transición  $2 \rightarrow 1$ . Lo primero es identificar el biestable o biestables que han efectuado una transición de estado incorrecta. En este ejemplo el biestable que realiza una transición incorrecta sería Q1 (es el único biestable que tiene un bit incorrecto).

Una vez identificado el biestable que falla:

- Si teniendo en cuenta los valores actuales de las entradas y la tabla de verdad del biestable que falla, se deduce que el próximo estado ha de ser el de la secuencia incorrecta observada, entonces el biestable está funcionando correctamente. El fallo se encuentra en la red combinacional (SC2) que genera el estado futuro. Aplicar los pasos de depuración de una red combinacional sobre esta red.
- Si las entradas del biestable que falla determinan que el próximo estado ha de ser el de la secuencia correcta entonces el biestable no funciona. Las posibles razones que no funcionen son:
  - No llega la señal de reloj.
  - Entradas asíncronas activadas.
  - Biestable sin polarizar o mal polarizado.
  - Chip estropeado.

A continuación se resume todo este proceso en forma de organigrama:

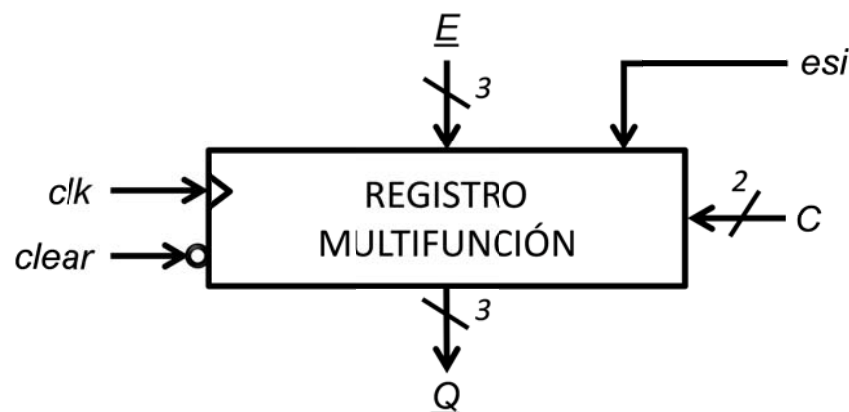




## FUNDAMENTOS DE COMPUTADORES

### PRÁCTICA 4: DISEÑO Y MONTAJE DE UN REGISTRO MULTIFUNCIÓN

El objetivo de esta práctica es diseñar y montar en el laboratorio un registro multifunción como el que se muestra en la figura capaz de realizar las funciones descritas en la tabla.



$C_1$	$C_0$	$\underline{Q}(t+1)$	Función
0	0	$\underline{E}(t)$	Carga paralela
0	1	$(Q_0(t), Q_2(t), Q_1(t))$	Rotación hacia la derecha
1	0	$(Q_1(t), Q_0(t), esi(t))$	Desplaza a la izquierda
1	1	$\underline{Q}(t)$	Conserva el valor

El circuito tendrá los puertos siguientes:

- Una salida paralela de datos de 3 bits,  $\underline{Q}$ , que muestra el contenido del registro.
- Una entrada paralela de datos de 3 bits,  $\underline{E}$ , por la que introducir el valor a cargar.
- Una entrada serie de datos,  $esi$ , por la que introducir el bit necesario para el desplazamiento a la izquierda.
- Una entrada de 2 bits,  $\underline{C}$ , para seleccionar el modo de funcionamiento.
- Una entrada,  $clk$ , de reloj.
- Una entrada,  $clear$ , de inicialización asíncrona a (000) activa a baja.

#### Desarrollo de la práctica

##### 1. Fase de diseño

- El diseño del circuito debe realizarse en casa y quedar reflejado en el cuadernillo correspondiente.
- El circuito deberá diseñarse usando biestables D y multiplexores 4 a 1.



## 2. Fase de montaje y depuración

- Utilizar los siguientes chips para montar el circuito:
  - **7474**: 2 biestables D (disparados por flanco de subida y señales de inicialización asíncrona a 0 ó 1 activas a baja).
  - **74153**: 2 multiplexores 4 a 1 (con señales de control comunes y señales de capacitación activas a baja separadas).
- Conectar las entradas *E*, *C*, *esi* y *clear* a *switches*.
- Conectar la entrada *clk* a un pulsador.
- Conectar la salida *Q* a *leds*.
- **Comprobar cada uno de los modos de funcionamiento** del registro para verificar que funciona correctamente. Si no es así, hay que depurarlo para encontrar los fallos y corregirlos.

## 3. Puesta en marcha

- Regular el generador de funciones del entrenador para que genere una onda cuadrada de 1 Hz de frecuencia.
- Conectar la entrada *clk* a la salida *TTL mode* del generador de funciones.
- Conectar la entrada de *clear* a un pulsador
- Cuando se haya comprobado que funciona correctamente, enseñar la práctica al profesor del laboratorio.



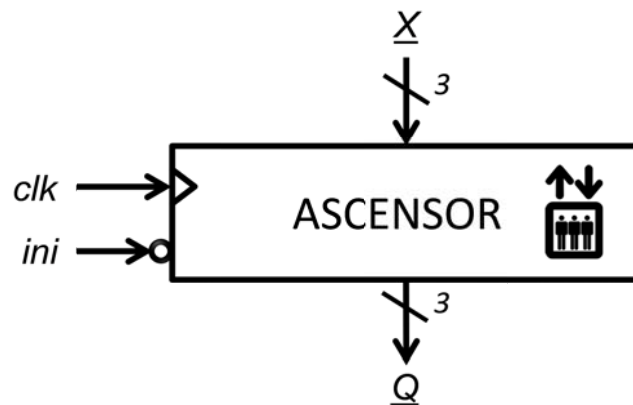
## FUNDAMENTOS DE COMPUTADORES

# PRÁCTICA 5: DISEÑO Y MONTAJE DE UN CIRCUITO EMULADOR DE UN ASCENSOR

El objetivo de esta práctica es diseñar y montar en el laboratorio un circuito secuencial síncrono que se comporte como el ascensor de una vivienda de 7 plantas.

El circuito tendrá los puertos mostrados en la figura:

- Una salida de 3 bits,  $\underline{Q}$ , que indica el piso en el que se encuentra el ascensor.
- Una entrada de 3 bits,  $\underline{X}$ , por la que se indica en binario el piso al que se desea ir.
- Una entrada,  $clk$ , de reloj.
- Una entrada,  $ini$ , de inicialización síncrona a (000) activa a baja.



Tanto  $\underline{Q}$  como  $\underline{X}$  codifican la planta en binario, es decir: (000) indica la planta baja, (001) indica la primera planta y así sucesivamente hasta (111) que indica la séptima planta.

La salida valdrá (000) en todos aquellos ciclos en los que  $ini$  valga 0. En los ciclos en los que  $ini$  valga 1, el comportamiento del sistema será como sigue:

- Si  $\underline{Q} = \underline{X}$ , la salida conservará su valor: el ascensor se encuentra en el piso en el que se desea ir y por tanto no se mueve.
- Si  $\underline{Q} > \underline{X}$ , la salida debe ir ciclo a ciclo decrementándose hasta que  $\underline{Q} = \underline{X}$ : el ascensor debe bajar porque la planta en la que se encuentra es superior a la planta a la que se desea ir.
- Si  $\underline{Q} < \underline{X}$ , la salida debe ir ciclo a ciclo incrementándose hasta que  $\underline{Q} = \underline{X}$ : el ascensor debe subir porque la planta en la que se encuentra es inferior a la planta a la que se desea ir.

### Desarrollo de la práctica

#### 1. Fase de diseño

- El diseño del circuito debe realizarse en casa y quedar reflejado en el cuadernillo correspondiente.

- Para emular el ascensor se utilizará un contador ascendente/descendente módulo 16 con carga paralela (chip 74169). Los 3 bits menos significativos de este contador equivalen a la salida  $Q$  del sistema.
- Para comparar la entrada  $X$  y la salida  $Q$  (y así determinar si el contador debe contar y en su caso si debe hacerlo ascendentemente o descendentemente) deberá usarse un sumador completo de 4 bits (chip 74283), inversores y puertas NAND de 3 entradas.

## 2. Fase de montaje y depuración

- Utilizar los siguientes chips para montar el circuito:
  - **74169**: 1 contador síncrono bidireccional módulo 16 (con señal de carga paralela activa a baja y prioritaria a las 2 señales de capacitación de cuenta activas a baja).
  - **74283**: 1 sumador completo de 4 bits.
  - **7410**: 3 puertas NAND de 3 entradas.
  - **7404**: 6 puertas NOT.
- Conectar la salida  $Q$  a un *display* de 7 segmentos derecho (que internamente incluye un conversor de código binario a código 7 segmentos).
- Conectar la entrada  $X$  a switches y adicionalmente al otro *display* de 7 segmentos izquierdo.
- Conectar la entrada *ini* a un *switch* adicional.
- Conectar la entrada *clk* a un pulsador.
- **Comprobar distintas combinaciones de planta solicitada y planta actual** para verificar que el sistema funciona correctamente. Si no es así, hay que depurarlo para encontrar los fallos y corregirlos.

## 3. Puesta en marcha

- Regular el generador de funciones del entrenador para que genere una onda cuadrada de 1 Hz de frecuencia.
- Conectar la entrada *clk* a la salida *TTL mode* del generador de funciones.
- Conectar la entrada de *ini* a un pulsador
- Cuando se haya comprobado que funciona correctamente, enseñar la práctica al profesor del laboratorio.