



UNIVERSIDAD DE BUENOS AIRES

**Facultad de Ingeniería  
Departamento de Electrónica**

***Estructura del Computador (66.70)***

# **Controlador de Ascensores**

**Cuatrimestre y año:** 2º 2010

**Profesor Titular Teóricas:** Ing. Jorge Mazzeo

**Docentes a cargo de las prácticas:**

- Ing. Ricardo Arias
- Sr. Daniel Trichero
- Srita. Marina Zambrano

## Integrantes:

<b><i>Padrón</i></b>	<b><i>Nombre</i></b>	<b><i>Email</i></b>
88153	Elvio Tocalino	elviotocalino@gmail.com
87944	Lucas Robles	lrobles@fi.uba.ar
82341	Maximiliano Monzón	maximilianohm@gmail.com

## Objetivo:

El objetivo del presente trabajo práctico es el de diseñar un módulo de control de llamadas de ascensor.

## Descripción del Problema:

Un edificio de cuatro pisos (subsuelo, planta baja, primer y segundo piso) posee dos ascensores que permiten la movilidad entre todos los pisos. Cada piso, cuenta con acceso a ambos ascensores, un pulsador único que permite llamar a un ascensor (junto con una luz azul que indica que se ejecuto la llamada) y una luz verde encima de cada ascensor que indica que dicho ascensor está respondiendo a la llamada.

El sistema pide determinar cual ascensor llamar cuando se aprieta el botón, generando el comando para que ocurra y encender las luces o apagarlas según sea necesario. Adicionalmente se debe tener en cuenta:

- El ascensor a llamar debe ser el que se encuentra lo más cerca del piso desde donde se apretó el botón, estando este libre.
- En caso de que uno esté libre y el otro no, el ascensor a llamar es aquel que esté libre sin importar su distancia al pulsador accionado.
- Si los dos ascensores están ocupados se ignora la llamada.

Para simplificar la lógica se utiliza dos integrados: El "Lift Interface" que controla la mecánica de los ascensores (uno por cada ascensor) y otro que genera un pulso angosto cada vez que un ascensor cambia de piso independientemente de si es en ascenso o en descenso.

## Convenciones de nombres

- Ya que la puesta en práctica de esta solución involucraría la confección de una placa PCB a ser instalada en cada uno de los 4 pisos del edificio, en el presente trabajo se usa el término “control de ascensores” o “controlador” para referirse a dicha placa.
- Se designa como “usuarios” a aquellas personas que operaran el controlador. Puede pensarse en los inquilinos que habitan el edificio, si se descarta el uso de la botonera del ascensor.
- Se usa el nombre de “pulsador” para el dispositivo al alcance de los usuarios que emite un pulso para llamar a los ascensores.
- Se usan los términos “ascensor1” y “ascensor2” para identificar los ascensores. Cabe destacar que, aunque se podría asignar arbitrariamente cuál ascensor será el ascensor1, este tendrá prioridad posteriormente durante la toma de decisiones (es decir, ante empates, se llama al ascensor1).

## Hipótesis del trabajo

La solución descrita fue ideada asumiendo:

1. El piso “P” en que se sitúa el controlador es un dato constante.
2. La luz verde se prende cuando el ascensor es elegido y se apaga cuando el ascensor arriba al piso del controlador.
3. Siempre que haya una luz verde prendida en un piso, su luz azul estará encendida.
4. Conocemos los estados libre/ocupado de cada ascensor por las variables “L1” y “L2”. Esto se debe a que aún no hemos diseñado un mecanismo que permita proveer dicha información. Delegamos esta tarea para el futuro.
5. No nos vamos a ocupar de la manera en que los cuatro controladores (uno por cada piso) podrían comunicarse con cada Lift-Interface simultáneamente sin crear conflictos. Asumimos que tal tarea será cumplida por alguien más.
6. Para un ascensor en estado libre, leer “D” de su Lift-Interface permite conocer en qué piso se encuentra.

## Descripción conceptual de la solución

Para evitar confusiones se proponen algunos nombres que serán usados frecuentemente durante la descripción de la solución y en los diagramas.

pulsoE: el pulso que es emitido cuando un usuario oprime el pulsador, interpretado como un flanco de bajada.

pulsoS: el pulso que será enviado desde el controlador hacia el Lift-Interface de uno (y solo uno) de los ascensores para llamarlo.

L1, L2: señal booleana que indica si el ascensor1, ascensor2 está libre (uno) u ocupado (cero).

PA1, PA2: piso en que actualmente se encuentra el ascensor1, ascensor2.

P: piso en que se instaló el controlador.

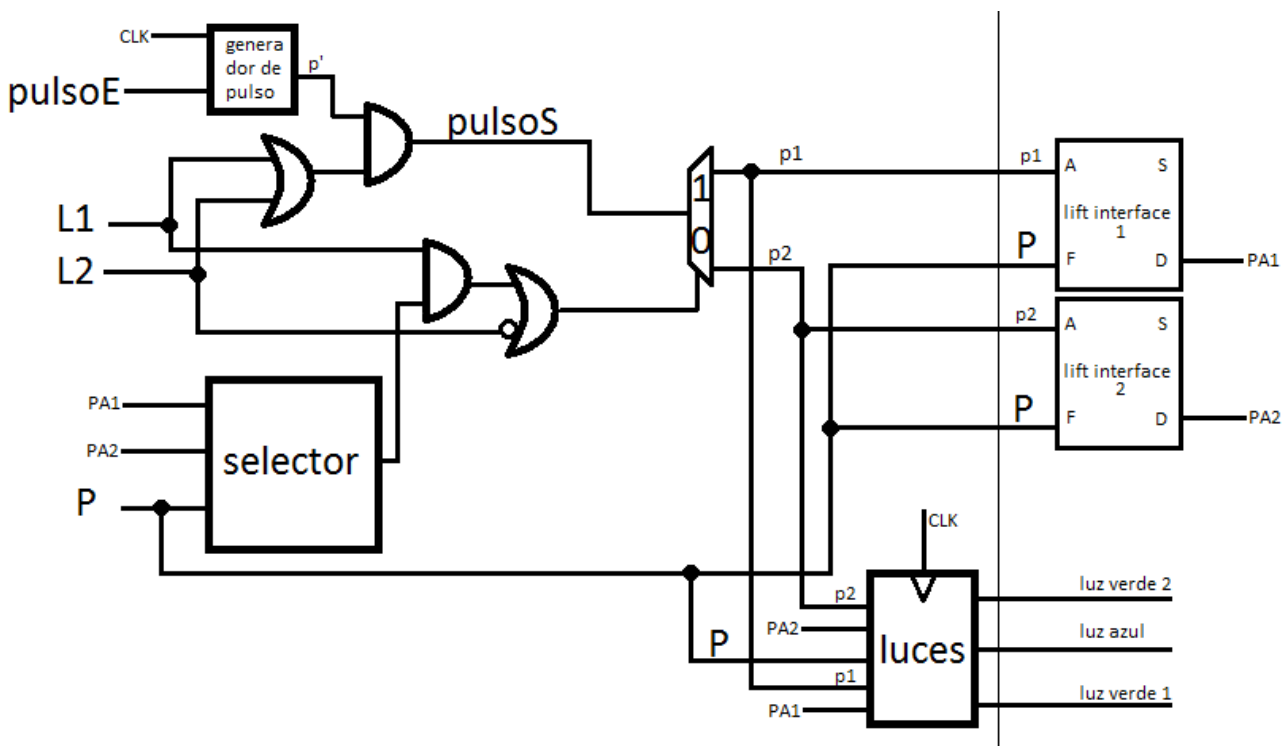
p1,p2: el pulsoS enviado/por-enviar al ascensor1,ascensor2.

El pulsoE es el disparador del circuito, pero no es confiable: su duración depende del usuario. Para evitar problemas que pudieran surgir por esto generamos el pulsoS, de un período del CLOCK de duración, por cada flanco de bajada del pulsoE.

Razonando sobre los posibles casos a manejar cuando se produce un pulsoE, concluimos que dependiendo de los estados libre/ocupado de cada ascensor se llama a uno u otro, o se ignora la llamada. Organizamos pues nuestras posibles respuestas en función de esta información:

ascensor1 libre (L1)	ascensor2 libre (L2)	respuesta
0	0	ignoro
1	0	ascensor1
0	1	ascensor2
1	1	selector(PA1,PA2,P)

Para poder ignorar o atender una llamada trabajamos con el pulsoS, asociando una señal *enabling* que lo neutralice cuando sea necesario. De pulsoS se derivan p1 y p2 que van a los ascensores. Además, de la hipótesis 6 se obtiene que PA1 y PA2 pueden leerse de las Lift-Interface de los ascensores. Verdaderamente, las variables P, pulsoE, L1 y L2 son las verdaderas entradas del circuito solución. El resto son calculadas y reusadas. Se muestra a continuación un diagrama que corresponde al planteo de la solución:



En este diagrama puede apreciarse la separación entre las partes que corresponden al controlador (a la izquierda de la línea vertical) y aquellos

componentes externos con los cuales este debe interactuar. Cabe distinguir, sin embargo, que de la parte derecha las Lift-Interface de los ascensores son compartidas entre los cuatro controladores, mientras que las luces no.

Como se mencionó antes, el pulsoE es transformado en un pulso “p'” sincronizado con el CLOCK del circuito, y cuya duración es de un período del CLOCK. El pulso p' es neutralizado cuando  $L1=L2=0$ , y en caso contrario su valor es usado como pulsoS. Además, puede verse que el selector actúa constantemente, pero su valor solo es significativo cuando  $L1=L2=1$ . Esto último no es evidente en el diagrama, por lo que vamos a derivarlo ahora:

Sea “R” la respuesta del selector, entonces la señal de control del demultiplexor 1a2 que actúa sobre pulsoS resulta  $f = f(L1,L2,R)$  tal que  $f=1$  para elegir al ascensor1 y  $f=0$  para elegir al ascensor2. Si usamos Karnaugh para sintetizar la función f:

r	L1,L2	.00	.01	.11	.10
0		x	0	0	1
1		x	0	1	1

Gracias a las redundancias (debidas al caso  $L1=L2=0$  ya explicado) obtenemos

$$f(L1,L2,R) = not(L2) + R * L1$$

Los componentes dibujados en el diagrama, de nombres “luces”, “generador de pulso” y “selector”, se presentan brevemente a continuación.

### **selector:**

Este componente es el encargado de decidir a qué ascensor llamar cuando ambos están libres.

Recibe tres entradas de dos bits cada una, que son interpretadas como los pisos de los ascensores y el piso del controlador. La salida de un bit será “1” para llamar al ascensor1 y “0” para el ascensor2.

Este componente permite conocer en todo momento cuál sería el ascensor a llamar si se produjera un pulsoE con ambos libres.

El selector se implementa casi matemáticamente, haciendo la diferencia en módulo del piso de cada ascensor con el piso P del controlador, y luego comparando los resultados para ver cual fue menor.

### **luces:**

Este componente usa las hipótesis 2 y 3 para mantener las luces del piso en que está el controlador funcionando como lo requiere el enunciado del TP.

### **generado de pulso:**

Un pequeño componente que transforma el pulso emitido por el usuario, que es irregular y asincrónico, en un pulso de duración constante y sincronizado.

Sigue una descripción minuciosa de estos componentes, detallando su comportamiento e implementación.

## El Selector

El selector elige entre dos ascensores, cuando ambos están libres.

Entradas:

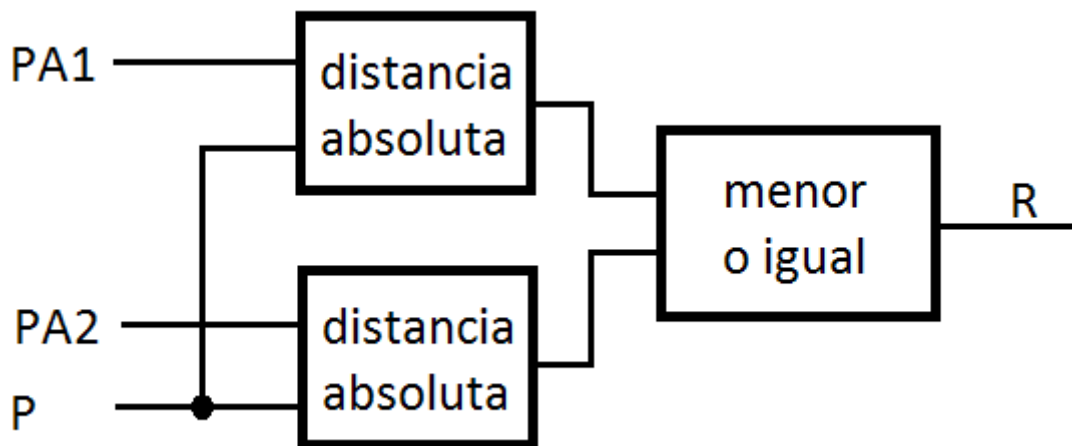
P1 (2 bits): piso en que se encuentra el ascensor 1, libre.

P2 (2 bits): piso en que se encuentra el ascensor 2, libre.

P (2 bits): piso en que se produjo el llamado.

Salidas:

R (1 bit): señal que indica que hay que llamar al ascensor 1 si es uno, o al 2 si es cero.



Como se puede ver, el selector es en realidad una articulación de sub-componentes. R se calcula a partir de la entrada matemáticamente, usando las funciones implementadas por los componentes “distancia absoluta” y “menor o igual”, que se detallan a continuación:

### ***DISTANCIA ABSOLUTA:***

Sinopsis: Calcula la distancia absoluta entre dos números, ambos entre 0 y 3 inclusive.

Ecuación:  $\text{distAbs}(A,B) == |A - B|$

Entradas:

A (dos bits) = A1,A0

B (dos bits) = B1,B0

Salidas

D (dos bits) = D1,D0 número entre 0 y 3 inclusive.

Para D1

B1 B0	.00	.01	.11	.10
A1 A0				
.00	0	0	1	1
.01	0	0	1	0
.11	1	1	0	0
.10	1	0	0	0

Para D0

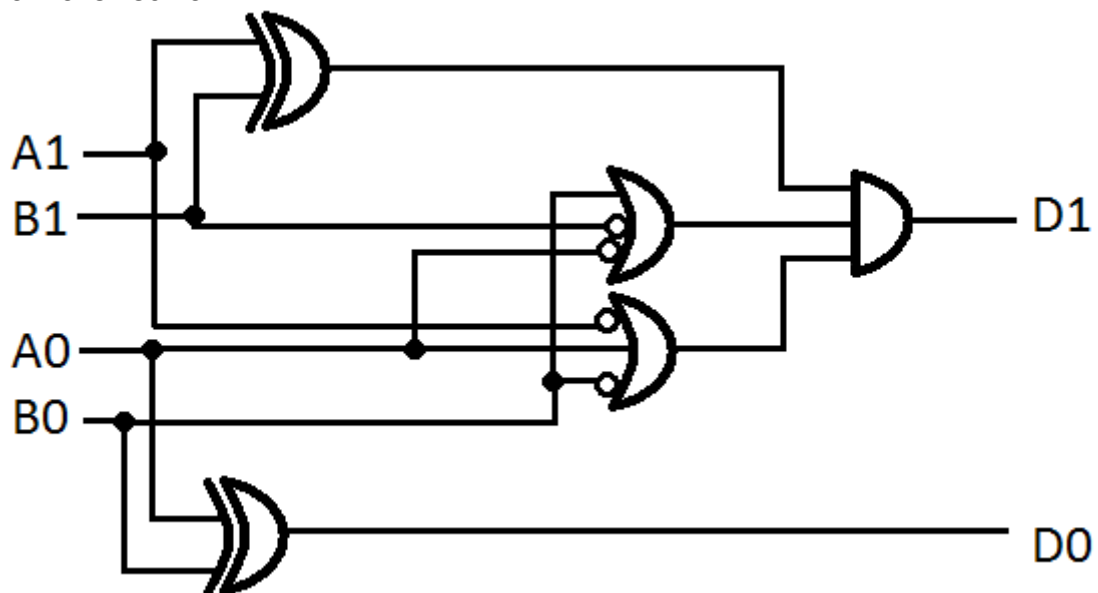
B1 B0	.00	.01	.11	.10
A1 A0				
.00	0	1	1	0
.01	1	0	0	1
.11	1	0	0	1
.10	0	1	1	0

Expresión booleana:

$$D_1(A_1, A_0, B_1, B_0) = (A_1 \text{ xor } B_1) * (\text{not}(A_1) + A_0 + \text{not}(B_0)) * (\text{not}(A_0) + \text{not}(B_1) + B_0)$$

$$D_0(A_1, A_0, B_1, B_0) = A_0 \text{ xor } B_0$$

Diagrama circuital:



## MENOR O IGUAL:

Sinopsis: Indica si A es menor o igual que B, para A y B números entre 0 y 3 inclusive.

Ecuación:  $\text{menorOIgual}(A,B) === (A < B) \text{ ó } (A = B)$

Entradas:

A (dos bits) = A1,A0

B (dos bits) = B1,B0



Salidas:

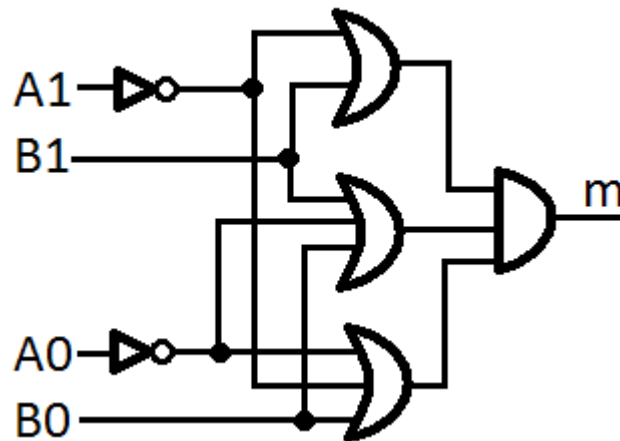
m = señal de un bit que estará en 1 cuando A sea menor o igual a B, y 0 cuando B sea menor que A.

B1 B0	.00	.01	.11	.10
A1 A0				
.00	1	1	1	1
.01	0	1	1	1
.11	0	0	1	0
.10	0	0	1	1

Expresión booleana:

$$m(A_1, A_0, B_1, B_0) = (\text{not}(A_1) + B_1) * (\text{not}(A_1) + \text{not}(A_0) + B_0) * (\text{not}(A_0) + B_1 + B_0)$$

Diagrama circuital:



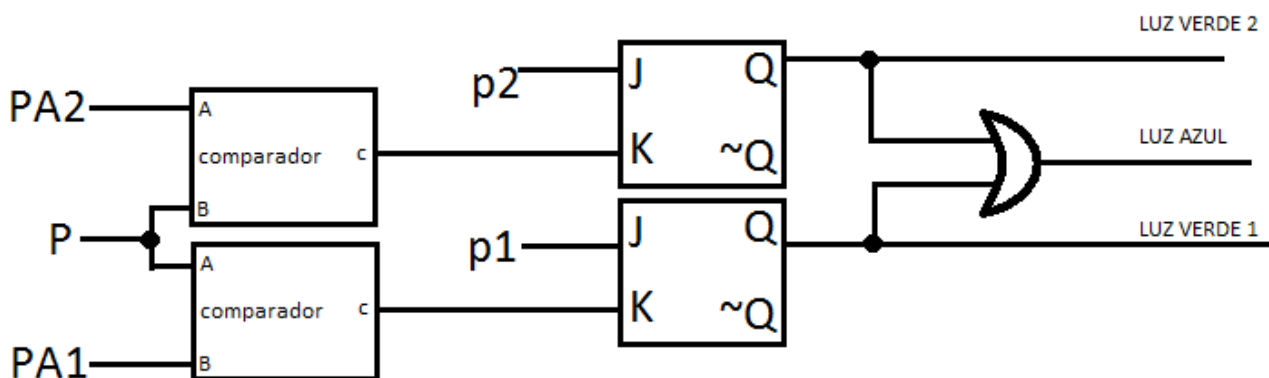
## Las Luces

Este componente se ocupa de activar, y mantener activas, una serie de señales que representan el estado prendido/apagado de las tres luces correspondientes al piso en que es instalado el controlador. Para mantener el estado de las luces se usan FF-JK, provistos para el TP. Gracias a la hipótesis 3, bastan 2 FF para mantener el estado de las 3 luces. La novedad de este componente respecto de los ya descritos es que sus entradas son todas pulsos. Tanto p1 como p2 son derivados de pulsoS, el cual se deriva de pulsoE. Este es un pulso de entrada al circuito solución, por lo que su duración es incierta. Además S1 y S2 son leídos de la salida “S” del Lift-Interface del respectivo ascensor, también pulsos.

Las entradas son:

PA1, PA2 (leídas del “D” de cada Lift-Interface), P (ver hipótesis 1), y los pulsos p1 y p2 derivados de pulsoS.

Se muestra a continuación el circuito solución:



Donde p1/p2 son mutuamente excluyentes, y serán “1” si y solo si al menos uno de los dos ascensores está libre.

Ambos FF están conectados al mismo CLOCK (algo que el diagrama no muestra) que el resto del controlador.

### Comparador

Sinopsis: compara por igualdad dos números de dos bits cada uno.

Ecuación:  $\text{comparador}(A,B) == A = B$

Entradas:

A (dos bits)= A1, A0

B (dos bits)= B1, B0

Salida:

C (un bit) : valdrá 1 (uno) si A y B son iguales, y 0 (cero) si no lo son.

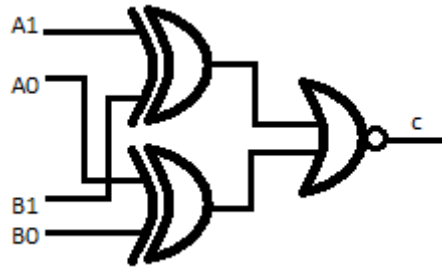
B1 B0	.00	.01	.11	.10
A1 A0				
.00	1	0	0	0
.01	0	1	0	0
.11	0	0	1	0
.10	0	0	0	1

$$C = (\text{not}(A_1) + B_1) * (A_1 + \text{not}(B_1)) * (\text{not}(A_0) + A_0) * (A_0 + \text{not}(B_0))$$

que se expresa mejor como :

$$C(A_1, A_0, B_1, B_0) = \text{NOR}(\text{xor}(A_1, B_1), \text{xor}(A_0, B_0))$$

Diagrama circuital:



## El generador de pulso

Usando un FF-D, emite un pulso alto “p'” de duración de un período del CLOCK por cada flanco de bajada del pulsoE.

El diagrama expresa la solución:

