

Solución

a) La tabla de verdad solicitada se muestra en la Tabla 3, donde se puede observar el comportamiento solicitado:

- Si $INTA = 0$, no se activa ninguna de las líneas de aceptación, aunque haya peticiones pendientes.
- Si $INTA = 1$, se activa la línea de aceptación de la petición $INTR_i$ más prioritaria.

ENTRADAS									SALIDAS							
INTA	INTR ₇	INTR ₆	INTR ₅	INTR ₄	INTR ₃	INTR ₂	INTR ₁	INTR ₀	INTA ₇	INTA ₆	INTA ₅	INTA ₄	INTA ₃	INTA ₂	INTA ₁	INTA ₀
0	X	X	X	X	X	X	X	X	0	0	0	0	0	0	0	0
1	1	X	X	X	X	X	X	X	1	0	0	0	0	0	0	0
1	0	1	X	X	X	X	X	X	0	1	0	0	0	0	0	0
1	0	0	1	X	X	X	X	X	0	0	1	0	0	0	0	0
1	0	0	0	1	X	X	X	X	0	0	0	1	0	0	0	0
1	0	0	0	0	1	X	X	X	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	X	X	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	1	X	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Tabla 3: Tabla de verdad de $INTA_i$ (X representa 1 ó 0, indistintamente)

b) En este apartado hay que seleccionar, a partir de las líneas $INTR_i$ la petición más prioritaria. Una posible solución consiste en utilizar un codificador de prioridad seguido de un decodificador. En la Figura 2 se muestra el diseño propuesto, donde se puede observar que tanto el codificador de prioridad como el decodificador tienen sus

14 Estructura y Tecnología de Computadores II

entradas de habilitación H activadas.

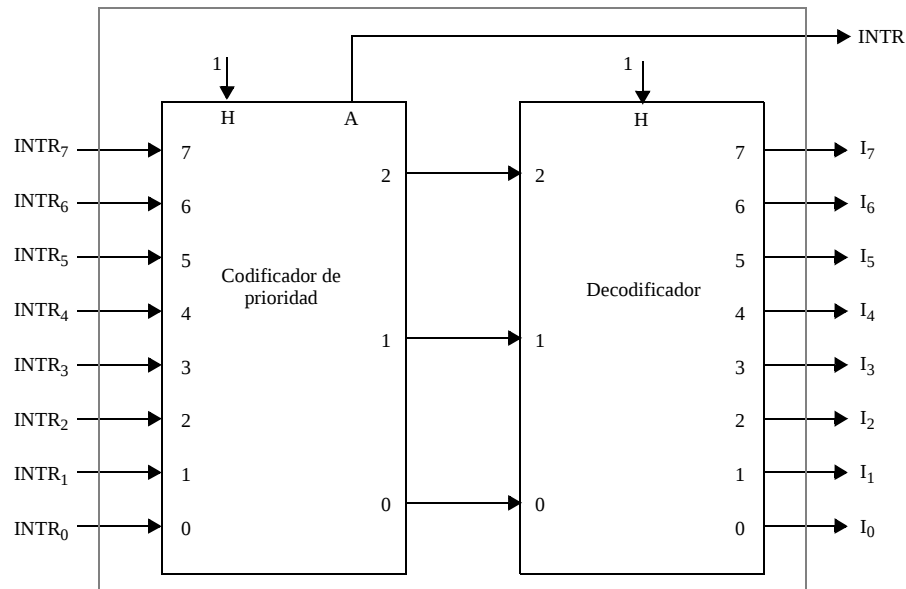


Figura 2: Mecanismo de gestión de prioridades

El codificador de prioridad genera la codificación binaria de la línea $INTR_i$ más prioritaria. A partir de aquí, por medio del decodificador se consigue que únicamente se active la línea I_i , y que las restantes salidas I_j , menos prioritarias, se encuentren a cero, independientemente de cómo se encuentren las correspondientes líneas $INTR_j$. Esto se recoge en la Tabla 4.

$INTR_7$	$INTR_6$	$INTR_5$	$INTR_4$	$INTR_3$	$INTR_2$	$INTR_1$	$INTR_0$	$INTR$	I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	X	X	X	X	X	X	X	1	1	0	0	0	0	0	0	0
0	1	X	X	X	X	X	X	1	0	1	0	0	0	0	0	0
0	0	1	X	X	X	X	X	1	0	0	1	0	0	0	0	0
0	0	0	1	X	X	X	X	1	0	0	0	1	0	0	0	0
0	0	0	0	1	X	X	X	1	0	0	0	0	1	0	0	0
0	0	0	0	0	1	X	X	1	0	0	0	0	0	1	0	0

Tabla 4: Tabla de verdad del mecanismo de gestión de prioridades (X representa 1 ó 0, indistintamente)

INTR ₇	INTR ₆	INTR ₅	INTR ₄	INTR ₃	INTR ₂	INTR ₁	INTR ₀	INTR	I ₇	I ₆	I ₅	I ₄	I ₃	I ₂	I ₁	I ₀
0	0	0	0	0	0	1	X	1	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1

Tabla 4: Tabla de verdad del mecanismo de gestión de prioridades (X representa 1 ó 0, indistintamente)

Cuando alguna de las líneas INTR_i esté a 1 significa que hay alguna petición de interrupción activa y que, por tanto, hay que activar la línea INTR conectada a la UCP para transmitir la petición. Para esto se utiliza la salida A del codificador de prioridad que se activa cuando alguna de sus entradas está a 1. El comportamiento de esta línea se puede ver también en la Tabla 4.

c) Cuando la UCP concede la interrupción activa su única línea INTA. El controlador de interrupciones tiene que activar en ese instante la línea de aceptación INTA_i correspondiente al dispositivo más prioritario que tenga pendiente una petición. Esta información se generó en el apartado anterior mediante las salidas I_i. Entonces, para generar las funciones lógicas de las ocho INTA_i que tiene el controlador de interrupciones que se está diseñando, basta con hacer la Y-lógica entre cada I_i y la línea INTA de aceptación generada por la UCP (ver Figura 3).

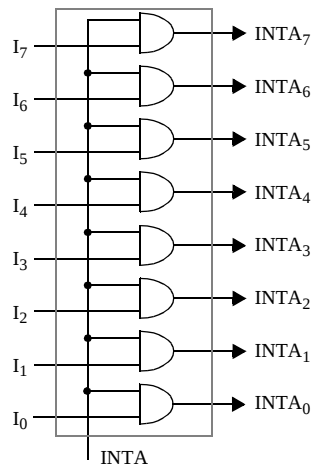


Figura 3: Generación de las señales de aceptación INTA_i

Para construir ahora el controlador de interrupciones solicitado tan solo hay que encadenar los módulos implementados en el apartado B (ver Figura 2) con el de la Figura 3, tal y como muestra la Figura 4.

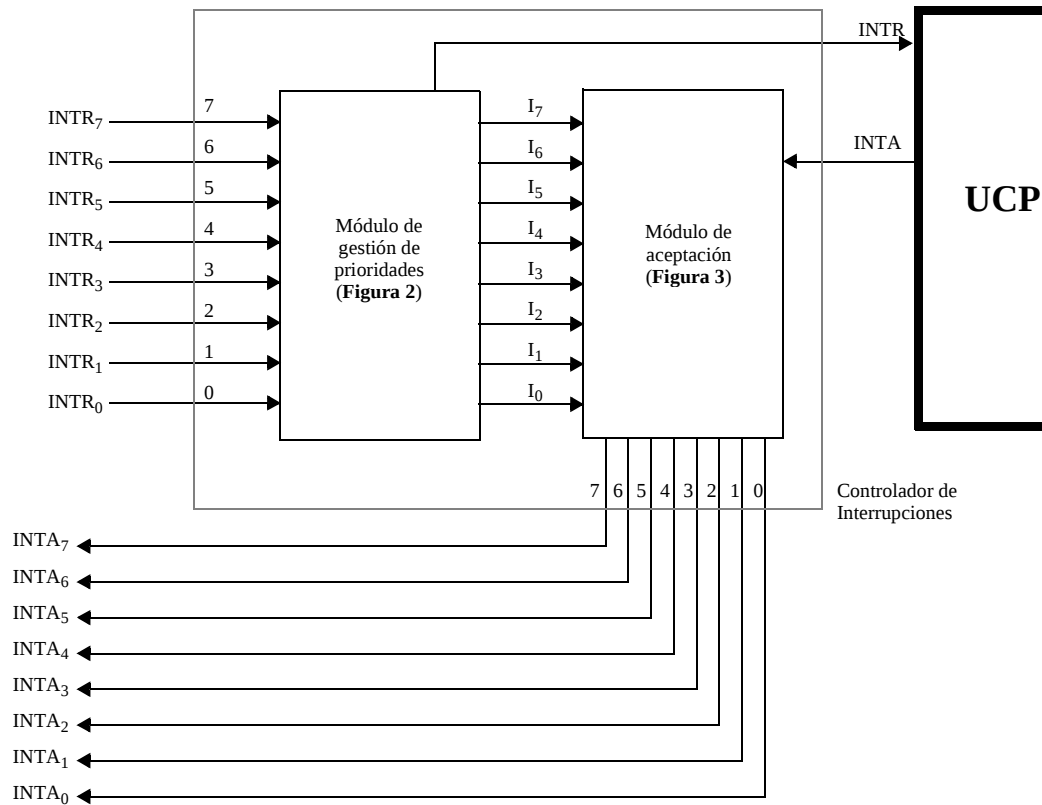


Figura 4: Controlador de interrupciones solicitado en el problema

Otra posible solución consiste en conectar la salida $INTA$ de la UCP a la entrada de habilitación H del decodificador de la Figura 2 de la página 14. De esta manera, las salidas del decodificador son ahora las señales $INTA_i$, tal como se muestra en la Figura 5.

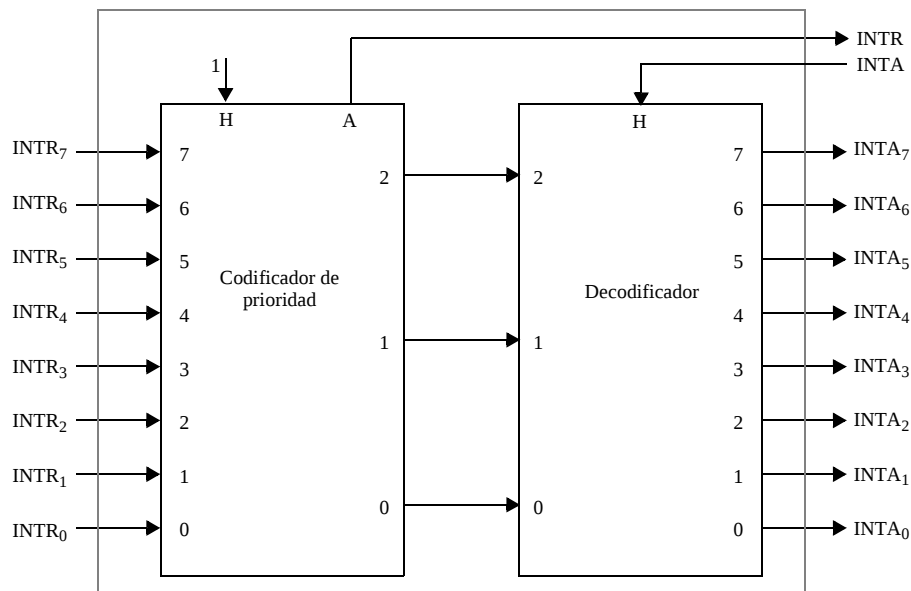


Figura 5: Diseño alternativo

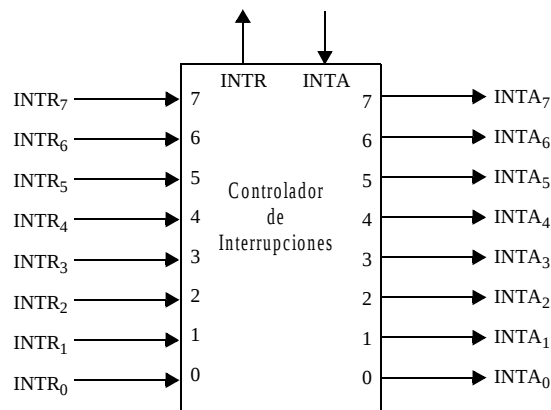


Figura 6: Representación del módulo controlador de interrupciones

d) En este apartado se va a representar el controlador de interrupciones como el módulo representado en la Figura 6. La forma de conectar dos de estos módulos para poder atender las peticiones de 15 dispositivos externos se muestra en la Figura 7. Denominando CI₁ al controlador de interrupciones 1 y CI₂ al controlador de interrupciones 2, el diseño tiene las siguientes características:

- Es capaz de atender peticiones de 15 dispositivos, tal y como requiere el enunciado del problema.

- La salida de petición de interrupción $INTR$ del CI_2 se ha conectado con la entrada $INTR_0$ del CI_1 , y la salida de aceptación $INTA_0$ del CI_1 se ha conectado con la entrada $INTA$ del CI_2 . De esta forma, las peticiones $INTR_7$ a $INTR_0$, solicitadas al CI_2 , sólo se atenderán cuando no haya peticiones activas en las líneas $INTR_8$ a $INTR_{14}$. Por tanto, la línea más prioritaria es la $INTR_{14}$ y la menos prioritaria es la $INTR_0$.

De esta manera se pueden encadenar tantos controladores de interrupciones como sean necesarios para atender a cualquier número arbitrario de peticiones.

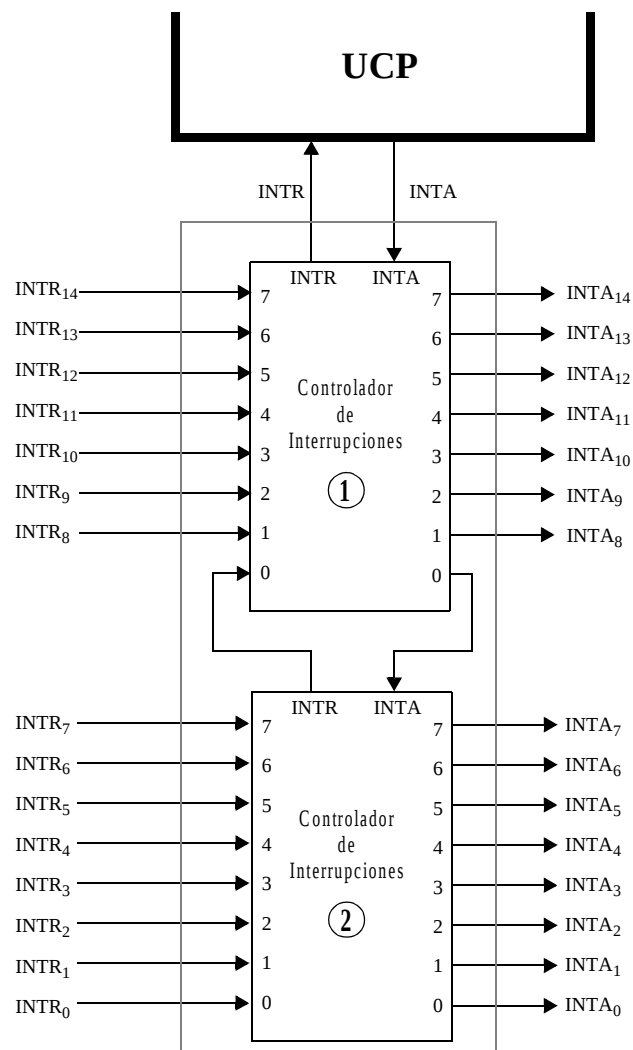


Figura 7: Controlador de interrupciones de 15 entradas

Nota importante:

A la hora de corregir este problema, el equipo docente de la asignatura se ha encontrado con bastante

frecuencia con el circuito que se muestra en la Figura 8. Se trata de una propuesta para el mecanismo de gestión de prioridades y generación de las señales de aceptación. Veamos por qué no funciona:

En la Figura 8 se ha representado una situación hipotética en la que están activas simultáneamente las peticiones $INTR_7$, $INTR_6$ e $INTR_5$, y la UCP ha concedido la interrupción mediante la activación de la línea de concesión $INTA$. Al ser la más prioritaria, la petición 7 debe ser atendida; efectivamente, el circuito propuesto genera $INTA_7 = 1$, lo que es correcto. La petición 6, al tener menor prioridad, no debe ser atendida: $INTA_6 = 0$, correcto. Pero $INTA_5$ es 1, cuando debería ser 0. El circuito está concediendo la interrupción simultáneamente a dos dispositivos diferentes, lo que es una situación errónea que puede ocasionar graves conflictos en el funcionamiento del computador.

Es fácil encontrar otras muchas situaciones en las que este dispositivo concede la interrupción a varios dispositivos simultáneamente. Se trata, por tanto, de una propuesta de solución no válida.

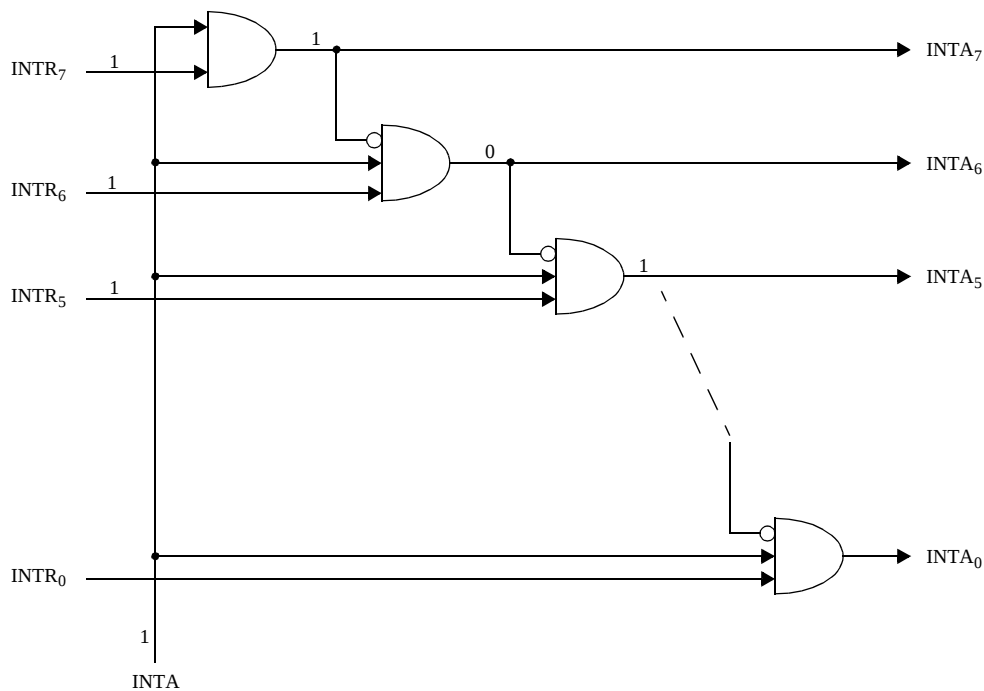


Figura 8: Circuito que no gestiona correctamente el esquema de prioridades