

Práctica 2

Periféricos II

Teclado y ratón PS2

1 Introducción

La tarjeta de Altera UP2 incluye la posibilidad de incorporar un teclado o un ratón PS2 ya que incluye un conector PS2 que incorpora la interfaz eléctrica entre la FPGA FLEX y el ratón/teclado. Es necesario diseñar hardware que haga de interfaz lógica entre la FLEX y el periférico PS2.

1.1 Interfaz PS2

El conector PS2 está formado por 6 pines que funcionan con tensiones típicas TTL. Estos 6 pines incluyen los dos de alimentación que suministra el servidor: conexión a tierra y tensión de alimentación, dos líneas bidireccionales: una línea de datos y otra de reloj y dos pines que no se utilizan en es estándar PS2. En el caso de la tarjeta UP2 la línea de datos es el pin 31 de la FPGA y la línea de reloj el pin 30, siendo ambas señales en colector abierto y bidireccionales. La línea de reloj la controla normalmente el periférico pero también la puede activar el computador o en este caso el chip FLEX cuando se desea parar la transmisión de información desde el periférico. Tanto la FPGA como el periférico pueden controlar la línea de datos, siendo ésta la única fuente de intercambio de datos ente ambos. El periférico y el computador pueden intercambiar diversos comandos y mensajes. A nivel de protocolo el ratón y el teclado son muy similares y comparten varios comandos.

1.1 Códigos de teclado

Los teclados están normalmente codificados emplazando las teclas en una matriz de filas y columnas. Todas las filas y columnas son muestreadas periódicamente por un controlador a una frecuencia suficientemente alta para poder percibir cualquier cambio en el teclado. Los datos de la tecla se pasan en serie al computador desde el teclado utilizando su código *scan*, identificador único de cada tecla basado en la dirección (fila y columna) en la matriz de interruptores de teclado. Típicamente los teclados tienen un par de juegos de códigos *scan*. El juego por defecto es el que el teclado adopta una vez se suministra alimentación. Para pasar al juego alternativo el computador envía un comando al teclado.

Los códigos *scan* son de dos tipos: *make* y *break*. Se envía un código *make* cada vez que se presiona una tecla. Cuando se libera ésta se envía in código *break*. Para la mayoría de teclas el código *break* consiste en enviar F0 seguido del código *make* de dicha tecla. De esta manera se puede gestionar con facilidad el que se mantenga una tecla pulsada previamente a la pulsación de otras, ampliando así el juego de caracteres del teclado. Si una tecla se mantiene pulsada el teclado envía en cada muestreo el código *make* de la tecla pulsada hasta que se libera, enviando entonces el código *break*.

1.2 El protocolo de transmisión serie PS2

Los códigos *scan* se envían en serie usando 11 bits sobre la línea de datos bidireccional. Cuando ni el teclado ni el computador necesitan enviar datos tanto la línea de reloj como la de datos permanecen a nivel alto (inactivas). la siguiente trama de bits tiene la siguiente estructura:

1. Un bit de inicio '0'
2. 8 bits de datos conteniendo el código scan desde el menos significativo (que se envía primero) hasta el más significativo.
3. Un Bit de paridad impar, de manera que el número de unos del byte de información más el bit de paridad sea un número impar.
4. Un bit de stop '1'.

Cuando el **teclado** desea enviar un comando realiza las siguientes acciones:

1. Verifica que ambas señales bidireccionales están inactivas o a '1'. Entonces fija la señal de datos a 0, preparando el bit de inicio.
2. El teclado baja la señal de reloj un semiperiodo de 35us.
3. El teclado fija en la línea de datos los restantes 10 bits, validándolos mediante el flanco de subida del reloj con un periodo de aproximadamente 70us.
4. Si el teclado desea enviar más datos tras el undécimo bit (bit de stop a '1') se encadena el bit de inicio de la siguiente trama de 11 bits.
5. La transmisión finaliza cuando el teclado fija ambas señales a '1' de nuevo.

Un controlador de teclado deberá tener en cuenta que la señal de reloj puede ser débil e incluir ruido o reflexiones, lo cual puede hacer peligrar la fiabilidad de los datos.

Cuando el **computador** o la FPGA FLEX envía comandos al teclado PS2 se realizan las siguientes acciones.

1. El computador fija la señal de reloj a '0' para evitar que el teclado envíe cualquier dato.
2. El computador fija la señal de datos a '0' y libera la señal de reloj para indicar que se tienen datos para enviar al teclado.
3. El teclado genera los flancos de reloj necesarios para capturar los restantes bits (8 bits de comando + bit de paridad + bit de stop).
4. Después de fijar el bit de stop a '1' se libera la señal de datos y entonces el teclado deja de generar pulsos de reloj.

Después de la recepción con éxito de un comando por parte del teclado, éste envía un comando de reconocimiento (FA). En el caso de que la recepción haya sido errónea se enviará un comando de re-envío (FE o FC).

1.3 La interfaz con el ratón PS2

El protocolo de transmisión serie entre el ratón y el teclado es idéntico al descrito para el teclado. Internamente el ratón contiene un mecanismo de detección de movimiento en los dos ejes. El microcontrolador del ratón envía los datos de movimiento y los eventos relacionados con la pulsación de las teclas del ratón. En este caso es necesario que el controlador del computador (o la FPGA en este caso) envíe un comando de inicialización (F4) para que el ratón funcione normalmente. Esto hace que la interfaz sea más complicada que en el caso del teclado.

El ratón envía información al sistema en paquetes de tres bytes que contienen información sobre el movimiento y el estado de los botones. Esta información se desglosa en la tabla 1.

	MSB							LSB
Bit	7	6	5	4	3	2	1	0
Byte 1	Yv	Xv	Ys	Xs	1	M	R	L
Byte 2	X ₇	X ₆	X ₅	X ₄	X ₃	X ₂	X ₁	X ₀
Byte 3	Y ₇	Y ₆	Y ₅	Y ₄	Y ₃	Y ₂	Y ₁	Y ₀

L = Bit de estado del botón de la izquierda ('1' = pulsado)

M = Bit de estado del botón medio (No es estándar PS2)

R = Bit de estado del botón de la derecha ('1' = pulsado)

X₇-X₀ = Movimiento en el eje X en comp. a 2. (Hacia la Izda.= Negativo. Dcha. = Pos.)

Y₇-Y₀ = Movimiento en el eje Y en comp. a 2. (Hacia abajo = Negativo. Arriba = Pos.)

Xv = Bit de sobrecarga de datos X (*overflow*)

Yv = Bit de sobrecarga de datos Y (*overflow*)

Xs = Bit de signo X ('1' = negativo)

Ys = Bit de signo Y ('1' = negativo)

2 Trabajo de laboratorio

2.1 Práctica con el teclado

Se propone que el estudiante descargue inicialmente el fichero `teclado.zip`, lo descomprima en un directorio temporal y en particular abra el fichero `kbd.gdf`, que será el nivel más alto en la jerarquía del proyecto. El bloque `KEYBOARD` que se muestra en la figura 1, es el controlador de teclado que va a hacer las funciones del computador como interfaz PS2. Los restantes dos bloques son meros decodificadores hexadecimales a 7 segmentos que servirán para observar en los *displays* 7 segmentos de la FPGA el valor del último byte del código *scan* último recibido.

El controlador del ejemplo no envía comandos al teclado, con lo que la señal de datos y de reloj son siempre entradas y se utilizan los valores por defecto del teclado tras su alimentación.



Figura 1: controlador de teclado PS2

Se propone que se pruebe el correcto funcionamiento del módulo y se analice la estructura del controlador de teclado. El estudiante deberá responder a las cuestiones:

Justifica la estructura del primer `Process` del código. ¿Qué función realiza y cuál es su utilidad?

Justifica la estructura del segundo `Process`.

Es posible modificar la estructura del código del controlador para que, además de recibir los códigos scan, sea capaz de enviar códigos al teclado. De esta manera se valorará cualquier mejora del diseño del controlador que se proponga. Opcionalmente (y siempre al final de la práctica tras la sección del ratón) se pueden realizar los diseños:

Opción 1: *Incluir el módulo VGA y la ROM de generación de caracteres para mostrar por pantalla el código scan completo recibido.*

Opción 2: *Modificar el controlador de teclado para enviar comandos al teclado como encender los leds, reconfigurar los tiempos de retraso y la tasa de repetición de las teclas, etc. En la sesión de laboratorio se dispondrá de la tabla de códigos completa para enviar comandos al teclado.*

2.2 Práctica con el ratón

Se debe descargar y abrir en este caso el fichero raton.zip siendo en este caso la parte más alta de la jerarquía el fichero mice.gdf. En este ejemplo el controlador de ratón PS2 es bastante más complejo que el de teclado y no se pretende profundizar en el código (sería necesario una descripción completa de la temporización del protocolo PS2 para justificar los diversos estados del controlador).

Una vez compilado el diseño y programada la FLEX (con el ratón previamente conectado) se deberá comprobar el funcionamiento del sistema. El display más significativo de la FPGA muestra los 4 bits más significativos de la columna actual del cursor. El display menos significativo muestra los 4 bits más significativos de la dirección de la columna. El botón izquierdo y derecho desactivan los puntos de sendos displays. Cuando se inicializa el sistema el ratón se presenta situado en el centro de una pantalla de 640 x 480 puntos.

Parte obligatoria: *Se propone modificar el diseño incorporando el módulo VGA para mostrar la posición del ratón por la pantalla VGA. El ratón se mostrará como un rectángulo con un grosor de 1 carácter blanco sobre fondo negro. Para simplificar el diseño el movimiento será a través de la pantalla en modo texto de 30 líneas x 40 columnas. Adicionalmente se pretende que mientras se mantenga pulsado el botón de la izquierda del ratón cambie el cursor a color rojo. Cuando el botón pulsado sea el de la derecha el fondo cambiará a color azul, volviendo a los colores iniciales cuando los botones no estén presionados.*

Apéndice

Códigos scan para teclado inglés de 101, 102 y 104 teclas.

Tecla	MAKE	BREAK	-----	Tecla	MAKE	BREAK	-----	Tecla	MAKE	BREAK
A	1C	F0,1C		9	46	F0,46		[54	F0,54
B	32	F0,32		`	0E	F0,0E		INSERT	E0,70	E0,F0,70
C	21	F0,21		-	4E	F0,4E		HOME	E0,6C	E0,F0,6C
D	23	F0,23		=	55	F0,55		PG UP	E0,7D	E0,F0,7D
E	24	F0,24		\	5D	F0,5D		DELETE	E0,71	E0,F0,71
F	2B	F0,2B		BKSP	66	F0,66		END	E0,69	E0,F0,69
G	34	F0,34		SPACE	29	F0,29		PG DN	E0,7A	E0,F0,7A
H	33	F0,33		TAB	0D	F0,0D		U ARROW	E0,75	E0,F0,75
I	43	F0,43		CAPS	58	F0,58		L ARROW	E0,6B	E0,F0,6B
J	3B	F0,3B		L SHFT	12	F0,12		D ARROW	E0,72	E0,F0,72
K	42	F0,42		L CTRL	14	F0,14		R ARROW	E0,74	E0,F0,74
L	4B	F0,4B		L GUI	E0,1F	E0,F0,1F		NUM	77	F0,77
M	3A	F0,3A		L ALT	11	F0,11		KP /	E0,4A	E0,F0,4A
N	31	F0,31		R SHFT	59	F0,59		KP *	7C	F0,7C
O	44	F0,44		R CTRL	E0,14	E0,F0,14		KP -	7B	F0,7B
P	4D	F0,4D		R GUI	E0,27	E0,F0,27		KP +	79	F0,79
Q	15	F0,15		R ALT	E0,11	E0,F0,11		KP EN	E0,5A	E0,F0,5A
R	2D	F0,2D		APPS	E0,2F	E0,F0,2F		KP .	71	F0,71
S	1B	F0,1B		ENTER	5A	F0,5A		KP 0	70	F0,70
T	2C	F0,2C		ESC	76	F0,76		KP 1	69	F0,69
U	3C	F0,3C		F1	05	F0,05		KP 2	72	F0,72
V	2A	F0,2A		F2	06	F0,06		KP 3	7A	F0,7A
W	1D	F0,1D		F3	04	F0,04		KP 4	6B	F0,6B
X	22	F0,22		F4	0C	F0,0C		KP 5	73	F0,73
Y	35	F0,35		F5	03	F0,03		KP 6	74	F0,74
Z	1A	F0,1A		F6	0B	F0,0B		KP 7	6C	F0,6C
0	45	F0,45		F7	83	F0,83		KP 8	75	F0,75
1	16	F0,16		F8	0A	F0,0A		KP 9	7D	F0,7D
2	1E	F0,1E		F9	01	F0,01]	5B	F0,5B
3	26	F0,26		F10	09	F0,09		;	4C	F0,4C
4	25	F0,25		F11	78	F0,78		'	52	F0,52
5	2E	F0,2E		F12	07	F0,07		,	41	F0,41
6	36	F0,36		PRNT SCRN	E0,12, E0,7C	E0,F0, 7C,E0, F0,12		.	49	F0,49
7	3D	F0,3D		SCROLL	7E	F0,7E		/	4A	F0,4A
8	3E	F0,3E		PAUSE	E1,14,77, E1,F0,14, F0,77	-NONE-				