

PS/2 protocol.

1. General requirements

We will use Altera's DE2 board and regular keyboard to implement hardware of PS/2 protocol. Keyboard will be connected to UP2 board. FPGA will receive data from keyboard, decode it and send the scan code to 7-segment led. Write the receiver that receives the data stream from keyboard and test-bench that checks the design, with error insertion. The keyboard receiver should be designed with fault recovery property, i.e. after error detecting on current packet, receiver may to continue in correct mode

2. About PS/2 protocol

The PS/2 keyboard implement a bidirectional synchronous serial protocol (LSB first). The bus is "idle" when both lines are high (open-collector). This is the only state where the keyboard/mouse is allowed begin transmitting data.

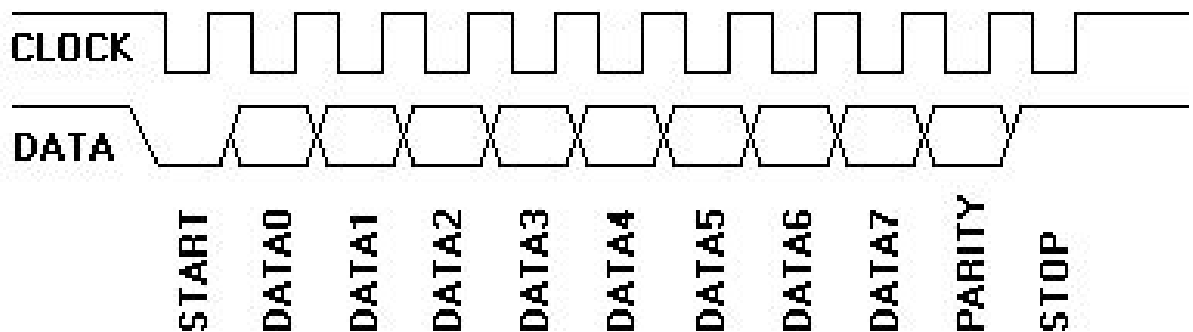


Figure 1. PS/2 signaling.

All data is transmitted one byte at a time and each byte is sent in a frame consisting of 11bits. These bits are:

- 1 start bit. This is always 0.
- 8 data bits, least significant bit first.
- 1 parity bit (odd parity).
- 1 stop bit. This is always 1.

The keyboard writes a bit on the Data line when Clock is high, and it is read by the host when Clock is low. The parity bit is set if there is an even number of 1's in the data bits and reset (0) if there is an odd number of 1's in the data bits. The number of 1's in the data bits plus the parity bit always add up to an odd number (odd parity.) This is used for error detection. The keyboard must check this bit and if incorrect it should respond as if it had received an invalid command.

Some tip: use start and stop bits for triplicate packed validity checking.



Figure 2: Scan code for the "Q" key (15h) being sent from a keyboard to the host (channel A - clock, channel B - Data).

The clock frequency is 10-16.7 kHz. The time from the rising edge of a clock pulse to a Data transition must be at least 5 msec. The time from a data transition to the falling edge of a clock pulse must be at least 5 microseconds and no greater than 25 msec. In high level the PS/2 protocol work by followed diagram.

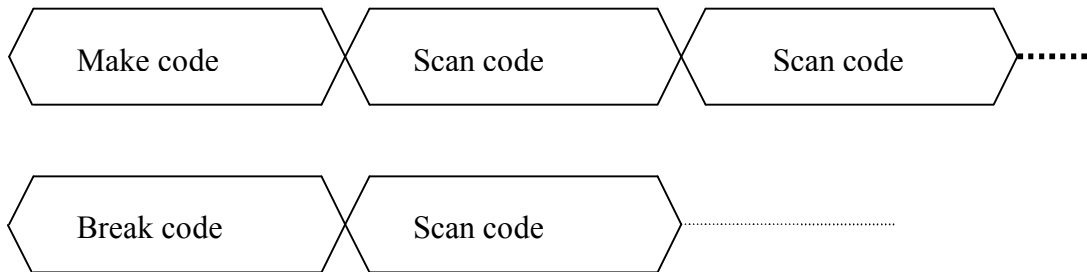


Figure 3. Key codes pattern.

Your keyboard's processor spends most of its time "scanning", or monitoring, the matrix of keys. If it finds that any key is being pressed, released, or held down, the keyboard will send a packet of information known as a "scan code" to your computer. There are two different types of scan codes: "make codes" and "break codes". A make code is sent when a key is pressed or held down. A break code is sent when a key is released. Every key is assigned its own unique make code and break code so the host can determine exactly what happened to which key by looking at a single scan code. The set of make and break codes for every key comprises a "scan code set" (table 1).

3. Seven segment indicator

One of the basic display components of digital circuits is the 7-segment display. It has (oddly enough) seven LED segments that are arranged in the pattern of an eight. To make a digit appear, the user must know which segments to turn on and which to leave off.

KEY	CODE	KEY	CODE	KEY	CODE	KEY	CODE
A	1C	2	1E	F2	6	D ARROW	E0,72
B	32	3	26	F3	4	R ARROW	E0,74
C	21	4	25	F4	0C	NUM	77
D	23	5	2E	F5	3	KP /	E0,4A
E	24	6	36	F6	0B	KP *	7C
F	2B	7	3D	F7	83	KP -	7B
G	34	8	3E	F8	0A	KP +	79
H	33	9	46	F9	1	KP EN	E0,5A
I	43	`	0E	F10	9	KP .	71
J	3B	-	4E	F11	78	KP 0	70
K	42	=	55	F12	7	KP 1	69
L	4B	\	5D	PRNT	E0,12,	KP 2	72
M	3A	BKSP	66	SCRN	E0,7C	KP 3	7A
N	31	SPACE	29			KP 4	6B
O	44	TAB	0D	SCROLL	7E	KP 5	73
P	4D	CAPS	58	PAUSE	E1,14,	KP 6	74
Q	15	L SHFT	12		E1,F0,	KP 7	6C
R	2D	L CTRL	14		F0,77	KP 8	75
S	1B	L GUI	E0,1F			KP 9	7D
T	2C	L ALT	11	[54]	5B
U	3C	R SHFT	59	INSERT	E0,70	;	4C
V	2A	R CTRL	E0,14	HOME	E0,6C	'	52
W	1D	R GUI	E0,27	PG UP	E0,7D	,	41
X	22	R ALT	E0,11	DELETE	E0,71	.	49
Y	35	APPS	E0,2F	END	E0,69		
Z	1A	ENTER	5A	PG DN	E0,7A		
0	45	ESC	76	U ARROW	E0,75	/	4A
1	16	F1	5	L ARROW	E0,6B		

Table 1. Scan code.

Inputs				Outputs							
0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	1	0	0	1	1	1	1	1
0	0	1	0	0	0	1	0	0	1	0	0
0	0	1	1	0	0	0	0	1	1	0	0
0	1	0	0	1	0	0	1	1	0	0	0
0	1	0	1	0	1	0	0	1	0	0	0
0	1	1	0	0	1	0	0	0	0	0	0
0	1	1	1	0	0	0	1	1	1	1	1
1	0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	1	0	0	0	0
1	0	1	1	1	1	0	0	0	0	0	0
1	1	0	0	0	1	1	0	0	0	0	1
1	1	0	1	1	0	0	0	0	1	0	0
1	1	1	0	0	1	1	0	0	0	0	0
1	1	1	1	0	1	1	1	0	0	0	0

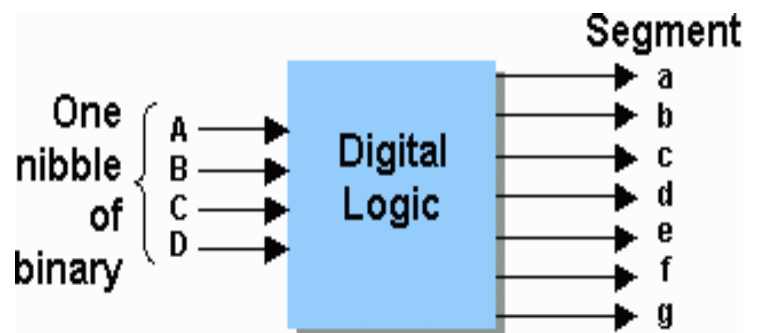
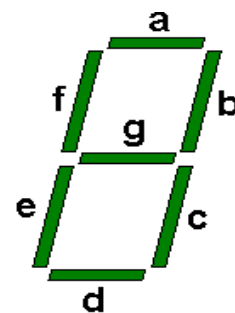


Figure 4. Truth table of hex to seven segment logic

4. Implementation tips

To use the signal that arrived from other clock (keyboard clock) domain (asynchronous signal) you need to sample it twice before referencing its value. This helps to prevent DFF's meta-stability. To determine the falling edge of signal, use the edge detector from Lab 3.

The output decoding implements by this way:

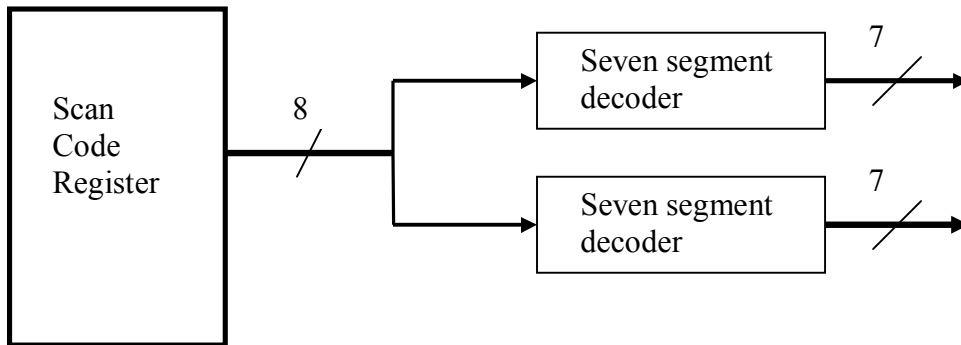


Figure 5. Output decoder block diagram.