PS/2 Keyboard Controller

Updated: January 2012
Source: Hamblen, Chapter 11 in *Rapid Prototyping of Digital Design* book, SOPC Edition, 2008.

Description:
This chapter explains the logic implemented in the "***keyboard.vhd***" file. It is used to interface with a PS/2 keyboard.

Communication Protocol:
When a key is pressed, a 'Make' code is sent. When a key is released, a 'Break' code is sent. For most keys, the 'Break' code is a data stream of F0 followed by the 'Make' code for the key.

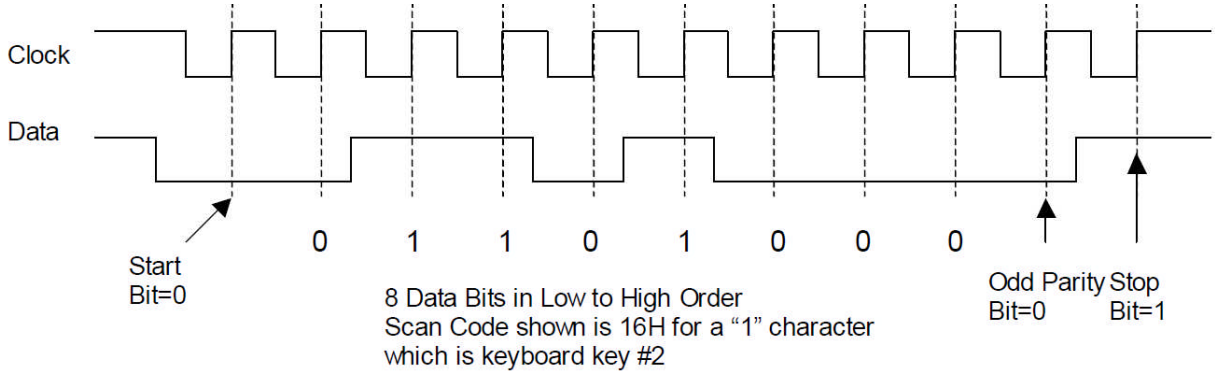
A PS/2 keyboard communicates with a host device using a data line and a clock line.

The scan codes are sent serially using 11 bits on the bi-directional data line. When neither the keyboard nor the computer needs to send data, the data line and the clock line are High (inactive). The transmission of a single scan code consists of:

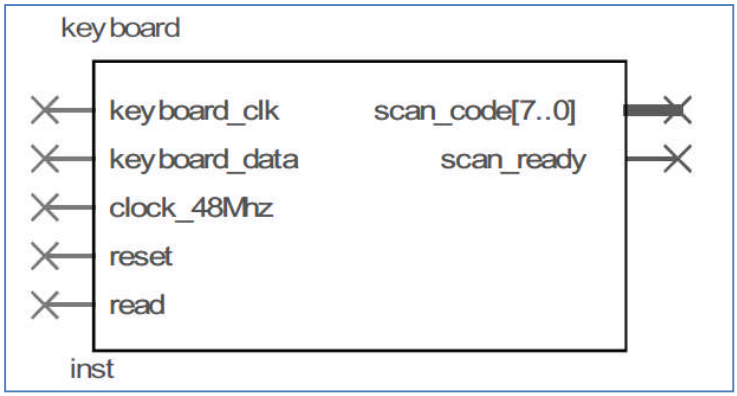
1. A start bit ('0')
2. Eight data bits containing the key scan code in low to high bit order
3. An odd parity bit
4. A stop bit ('1')

The following sequence of events occurs during a transmission of a command by the keyboard:.

1. The keyboard checks to ensure that both the clock and keyboard lines are inactive. Inactive is indicated by a High state. If both are inactive, the keyboard prepares the 'start' bit by dropping the data line Low.
2. The keyboard then drops the clock line Low for approximately 35us.
3. The keyboard will then clock out the remaining 10 bits at an approximate rate of 70us per clock (~14 kHz) period. The keyboard drives both the data and clock line.
4. The computer is responsible for recognizing the 'start' bit and for receiving the serial data. The serial data, which is 8 bits, is followed by an odd parity bit and finally a High stop bit. If the keyboard wishes to send more data, it follows the 12th bit immediately with the next 'start' bit. This pattern repeats until the keyboard is finished sending data at which point the clock and data lines will return to their inactive High state.



The PS/2 Keyboard controller (keyboard.vhd):




```
ENTITY keyboard IS
    PORT(   keyboard_clk, keyboard_data, clock_48MHz ,
            reset, read         : IN    STD_LOGIC;
            scan_code           : OUT   STD_LOGIC_VECTOR( 7 DOWNTO 0 );
            scan_ready          : OUT   STD_LOGIC);
END keyboard;
```

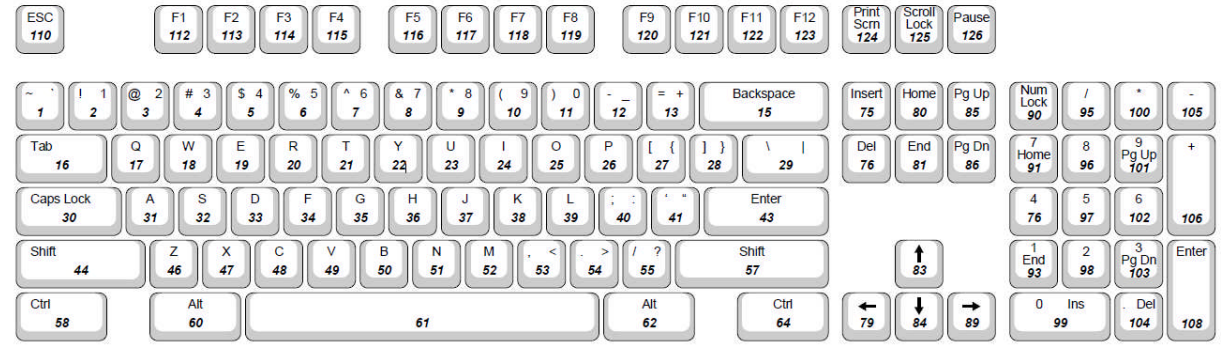
keyboard_clk: clock signal from a PS/2 keyboard.
keyboard_data: data signal from a PS/2 keyboard.
clock_48MHz: 50 MHz clock signal on a DE2-70 board.
reset: active low reset signal. Generally connected to KEY(0) on a DE2-70 board.
scan_code: 8-bit scan code received from a PS/2 keyboard.
scan_ready: an active high signal to user logic to indicate that the scan code is valid and ready.

read: an active high control signal from user logic to indicate the scan code hs been read. When read signal is asserted by user logic, scan_ready signal is reset to Low.

First, the 50 MHz clock signal is divided by two to produce a 25 MHz clock rate that is used internally. The keyboard controller operates with this 25 MHz clock signal. The controller monitors and filters the keyboard clock to reduce glitches. The keyboard clock signal is determined to be valid only if it has been High or Low for 8 successive 25 MHz clock periods. The filtered keyboard clock is used in conjunction with the keyboard data signal to receive and generate a valid scan code. The controller waits for the a start bit, converts the next eight serial data bits to parallel, and outputs the 8-bit scan code to the scan_code signal. scan_ready is set to '1' whenever a new scan code is received. The input signal read from user logic, resets the scan_ready handshake signal.

The user logic using this keyboard controller would need to wait until the scan_ready goes High. It should then read in the new scan code value, scan_code. Next, the user logic should set the signal read to High until scan_ready signal is Low. Then, the read signal should be reset to Low.

Scan Code Set 2 (not complete):



Key#	Make Code	Break Code	Key#	Make Code	Break Code	Key#	Make Code	Break Code
1	0E	F0 0E	31	1C	F0 1C	90	77	F0 77
2	16	F0 16	32	1B	F0 1B	91	6C	F0 6C
3	1E	F0 1E	33	23	F0 23	92	6B	F0 6B
4	26	F0 26	34	2B	F0 2B	93	69	F0 69
5	25	F0 25	35	34	F0 34	96	75	F0 75
6	2E	F0 2E	36	33	F0 33	97	73	F0 73
7	36	F0 36	37	3B	F0 3B	98	72	F0 72
8	3D	F0 3D	38	42	F0 42	99	70	F0 70
9	3E	F0 3E	39	4B	F0 4B	100	7C	F0 7C
10	46	F0 46	40	4C	F0 4C	101	7D	F0 7D
11	45	F0 45	41	52	F0 52	102	74	F0 74
12	4E	F0 4E	43	5A	F0 5A	103	7A	F0 7A
13	55	F0 55	44	12	F0 12	104	71	F0 71
15	66	F0 66	46	1A	F0 1A	105	7B	F0 7B
16	0D	F0 0D	47	22	F0 22	106	79	F0 79
17	15	F0 15	48	21	F0 21	110	76	F0 76
18	1D	F0 1D	49	2A	F0 2A	112	05	F0 05
19	24	F0 24	50	32	F0 32	113	06	F0 06
20	2D	F0 2P	51	31	F0 31	114	04	F0 04
21	2C	F0 2C	52	3A	F0 3A	115	0c	F0 0C
22	35	F0 35	53	41	F0 41	116	03	F0 03
23	3C	F0 3C	54	49	F0 49	117	0B	F0 0B
24	43	F0 43	55	4A	F0 4A	118	83	F0 83
25	44	F0 44	57	59	F0 59	119	0A	F0 0A
26	4D	F0 4D	58	14	F0 14	120	01	F0 01
27	54	F0 54	60	11	F0 11	121	09	F0 09
28	5B	F0 5B	61	29	F0 29	122	78	F0 78
29	5D	F0 5D	62	E0 11	E0 F0 11	123	07	F0 07
The remaining key codes are a function of the shift, control, alt, or num-lock keys.								