Features

- VHDL source code of a PS/2 keyboard interface
- Outputs the PS/2 make and break codes received from a keyboard
- Sets flag output when new code is available
- Validates parity, start, and stop bits of PS/2 transactions
- Configurable system clock frequency
- Synchronizes between PS/2 and system clock domains
- Debounces incoming PS/2 signals

Introduction

This details a PS/2 keyboard interface component for use in CPLDs and FPGAs, written in VHDL. The component receives data transactions from a PS/2 keyboard and provides the keyboard make and break codes to user logic over a parallel interface. It was designed using Quartus II, version 12.1. Figure 1 illustrates a typical example of the PS/2 keyboard interface integrated into a system. An example design that implements this PS/2 keyboard interface component to make a PS/2 keyboard to ASCII converter is available here 339 . The PS/2 keyboard interface is itself a simplified version of the PS/2 Host Transceiver available here 239 .

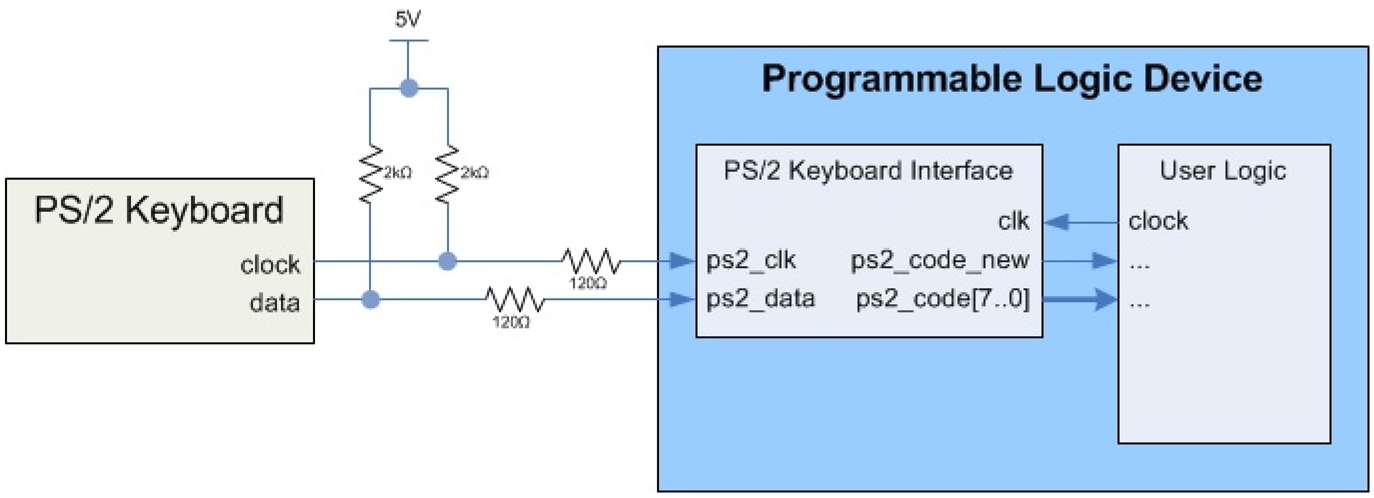


Figure 1. Example Implementation

Background

PS/2 (IBM Personal System/2) is an interface for keyboards and mice to PC compatible computer systems via a 6-pin Mini-DIN connector. The computing system must provide the keyboard or mouse with 5V source and ground connections. Communication occurs over a 2-wire serial interface, consisting of a clock line and a data line. Both lines require pull-up resistors (2kΩ shown in Figure 1). The 120Ω series resistors in Figure 1 are required to interface the 3.3V FPGA I/O to the 5V signals.

Once powered, the keyboard goes through a self- initialization sequence. Upon completion, it is ready to communicate keyboard events over the PS/2 interface.

Figure 2 illustrates the transaction format. Both clock and data signals are logic level high when inactive. The keyboard provides both the clock and data. The clock has a frequency between 10 kHz and 16.7 kHz (i.e. a 60-100us period). The data begins with a start bit (logic low), followed by one byte of data, a parity bit, and finally a stop bit (logic high). The data is sent LSB first. Each bit should be read on the falling edge of the clock signal. Once complete, both the clock and data signals return to logic level high.

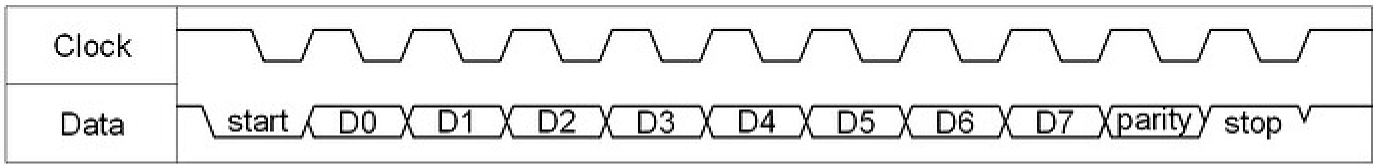


Figure 2. PS/2 Keyboard Transmission Timing Diagram

The data byte represents part of a keyboard scan code: either a make code (key press) or a break code (key release). Three different sets of scan codes exist, however the vast majority of keyboards use Scan Code Set 2, which is provided in the Appendix. A make code usually consists of either one or two bytes. If a make code uses two bytes, the first byte is x“E0”. A given key’s break code is typically the same as its make code, except that break codes include an additional x“F0” byte as the 2nd to last byte. (The PAUSE and PRNT SCRN keys are exceptions to the above.)

While it is possible to send data to a keyboard (to change its settings, etc.), this is generally unnecessary and is not included in this VHDL component. However, the PS/2 Host Transceiver component available here 239 can be used to do this.

Theory of Operation

Figure 3 conceptually illustrates the PS/2 keyboard interface component’s architecture. The clock and data signals from the keyboard are first synchronized and debounced. (The debounce component VHDL is provided above and documentation is available here 133 .) The resultant internal PS/2 data signal is then serially loaded into a shift register on falling edges of the PS/2 clock. An idle counter determines when the transaction is finished, defined by the PS/2 clock remaining at a high logic level for more than 55us, i.e. longer than half of the worst-case PS/2 clock period. Combinational error checking logic verifies the start bit, stop bit, and parity bit with the data. When the PS/2 port is idle and the data is valid, the component outputs the received PS/2 code and sets the *ps2_code_new* signal high to indicate that a new code is available on the *ps2_code* bus. The code remains available on the bus until another code is received.
The *ps2_code_new* signal remains high until another PS/2 transaction begins (when a low PS/2 clock signal clears the idle counter).

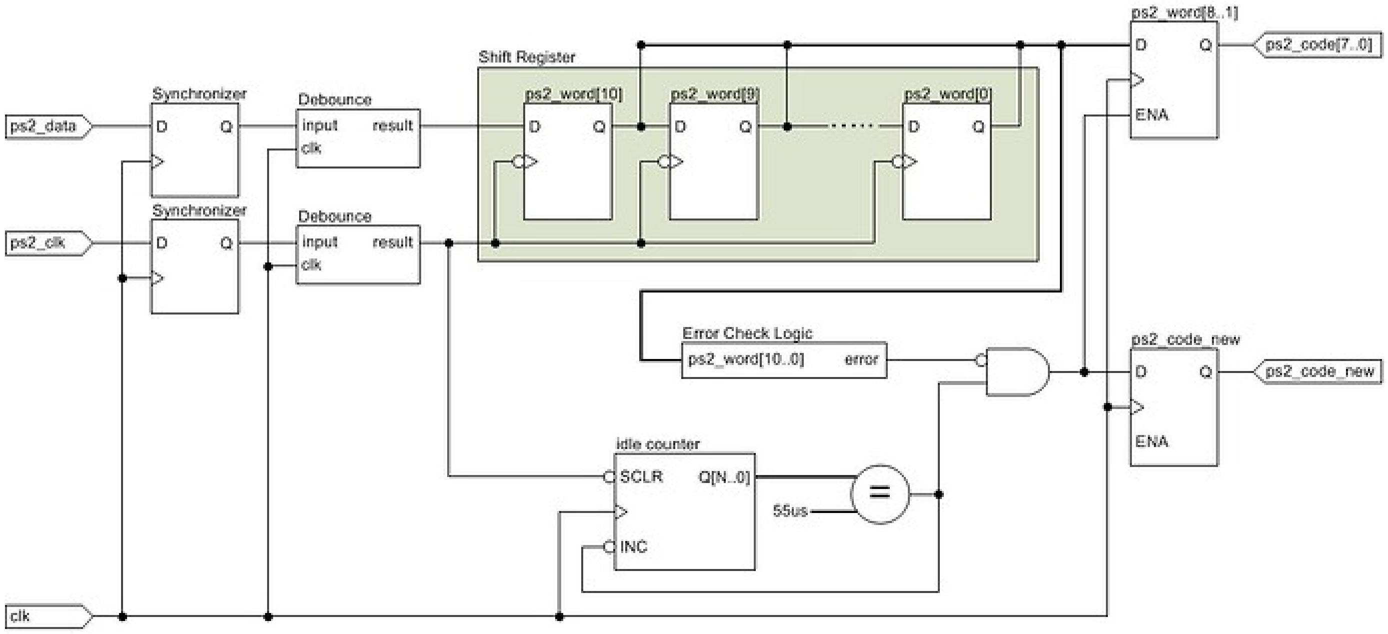
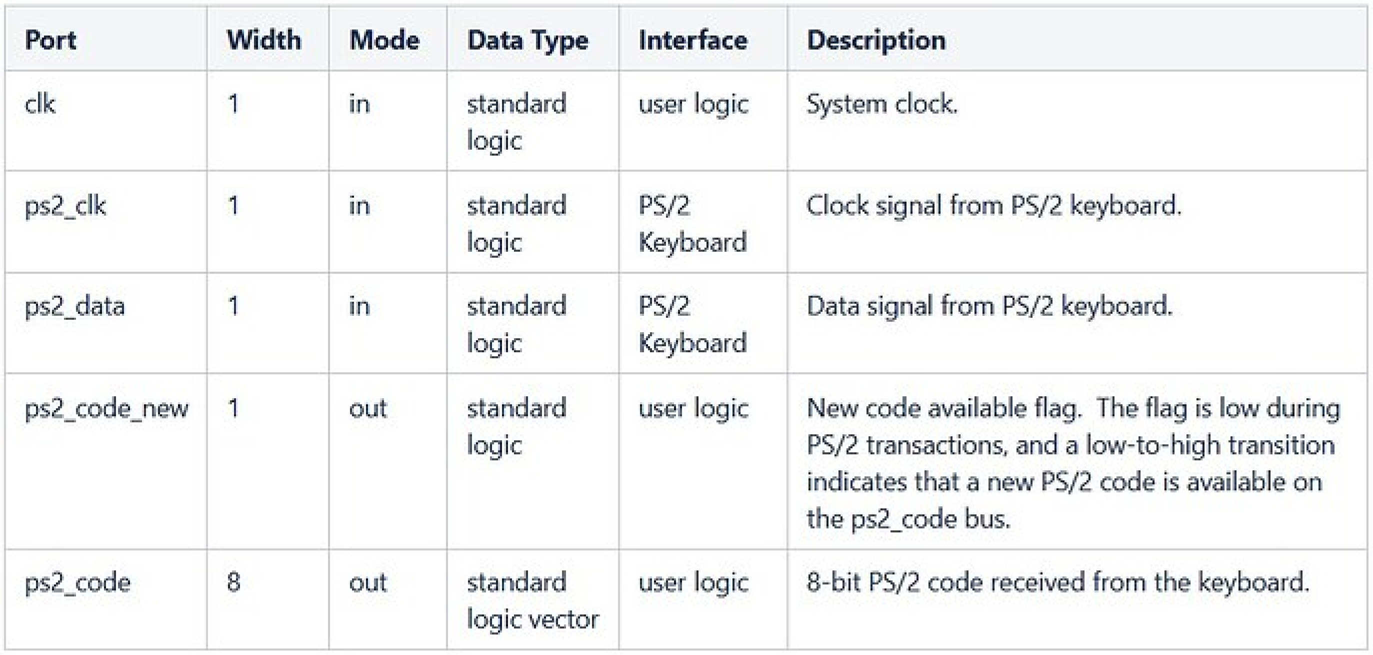


Figure 3. PS/2 Keyboard Interface Logic Architecture

Port Descriptions

Table 1 describes the PS/2 keyboard interface's ports.

Table 1. Port Descriptions



Port	Width	Mode	Data Type	Interface	Description
clk	1	in	standard logic	user logic	System clock.
ps2_clk	1	in	standard logic	PS/2 Keyboard	Clock signal from PS/2 keyboard.
ps2_data	1	in	standard logic	PS/2 Keyboard	Data signal from PS/2 keyboard.
ps2_code_new	1	out	standard logic	user logic	New code available flag. The flag is low during PS/2 transactions, and a low-to-high transition indicates that a new PS/2 code is available on the ps2_code bus.
ps2_code	8	out	standard logic vector	user logic	8-bit PS/2 code received from the keyboard.

Setting the Timing Parameters

The system clock speed affects the debounce time and the idle counter timing. The two GENERIC parameters declared in the ENTITY, *clk_freq* and *debounce_counter_size* must be set appropriately for the component to operate correctly. The *clk_freq* parameter must be set to the system clock *clk* frequency in Hz. The default setting in the provided code is 50 MHz (the frequency at which the component was simulated and tested). The *debounce_counter_size* parameter must be set such that 2^*debounce_counter_size* / *clk_freq* = 5us, as described in the documentation for the debounce component here 38 . For a 50 MHz system clock, *debounce_counter_size* = 8.

Example Transaction

Figure 4 shows the timing diagram of an example transaction. Once the PS/2 clock signal goes low, the *ps2_code_new* flag deasserts to indicate that a new PS/2 keyboard transaction is in progress. When the transaction completes, the *ps2_code_new* flag asserts to indicate that a new PS/2 code has been received and is available on the *ps2_code* bus. In this case, the PS/2 code received is x"1C", which is the make code for the "A" key.

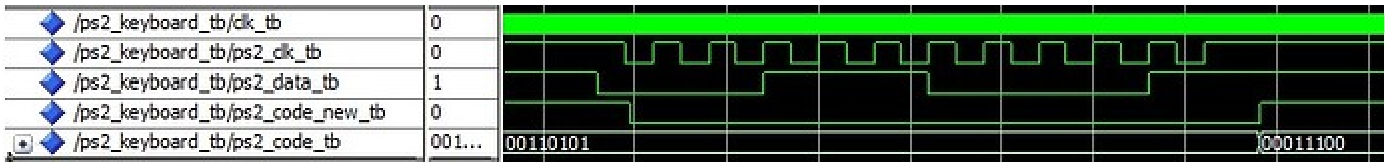


Figure 4. Example PS/2 Transaction

Conclusion

This PS/2 keyboard interface is a programmable logic component that receives transactions from PS/2 keyboards. It synchronizes the clocks domains, debounces the input signals, performs error checking, and notifies the user logic when new codes from the keyboard are available on its parallel output bus.

Appendix: Scan Code Set 2

Table A1. Keyboard Scan Code Set 2

Key	Make Code	Break Code
A	1C	F0,1C
B	32	F0,32
C	21	F0,21
D	23	F0,23
E	24	F0,24
F	2B	F0,2B
G	34	F0,34
H	33	F0,33
I	43	F0,43
J	3B	F0,3B
K	42	F0,42
L	4B	F0,4B
M	3A	F0,3A
N	31	F0,31
O	44	F0,44
P	4D	F0,4D
Q	15	F0,15
R	2D	F0,2D
S	1B	F0,1B
T	2C	F0,2C
U	3C	F0,3C
V	2A	F0,2A
W	1D	F0,1D
X	22	F0,22
Y	35	F0,35
Z	1A	F0,1A
0	45	F0,45
1	16	F0,16
2	1E	F0,1E
3	26	F0,26
4	25	F0,25
5	2E	F0,2E
6	36	F0,36
7	3D	F0,3D
8	3E	F0,3E
9	46	F0,46
`	0E	F0,0E

Key	Make Code	Break Code
-	4E	F0,4E
=	55	FO,55
|5D	F0,5D	
BKSP	66	F0,66
SPACE	29	F0,29
TAB	0D	F0,0D
CAPS	58	F0,58
L SHFT	12	FO,12
L CTRL	14	FO,14
L GUI	E0,1F	E0,F0,1F
L ALT	11	F0,11
R SHFT	59	F0,59
R CTRL	E0,14	E0,F0,14
R GUI	E0,27	E0,F0,27
R ALT	E0,11	E0,F0,11
APPS	E0,2F	E0,F0,2F
ENTER	5A	F0,5A
ESC	76	F0,76
F1	5	F0,05
F2	6	F0,06
F3	4	F0,04
F4	0C	F0,0C
F5	3	F0,03
F6	0B	F0,0B
F7	83	F0,83
F8	0A	F0,0A
F9	1	F0,01
F10	9	F0,09
F11	78	F0,78
F12	7	F0,07
PRNT SCRN	E0,12,E0,7C	E0,F0,7C,E0,F0,12
SCROLL	7E	F0,7E
PAUSE	E1,14,77,E1,F0,14,F0,77	-NONE-
[	54	FO,54
INSERT	E0,70	E0,F0,70
HOME	E0,6C	E0,F0,6C
PG UP	E0,7D	E0,F0,7D
DELETE	E0,71	E0,F0,71
END	E0,69	E0,F0,69

Key	**Make Code**	**Break Code**
PG DN	E0,7A	E0,F0,7A
U ARROW	E0,75	E0,F0,75
L ARROW	E0,6B	E0,F0,6B
D ARROW	E0,72	E0,F0,72
R ARROW	E0,74	E0,F0,74
NUM	77	F0,77
KP /	E0,4A	E0,F0,4A
KP *	7C	F0,7C
KP -	7B	F0,7B
KP +	79	F0,79
KP EN	E0,5A	E0,F0,5A
KP .	71	F0,71
KP 0	70	F0,70
KP 1	69	F0,69
KP 2	72	F0,72
KP 3	7A	F0,7A
KP 4	6B	F0,6B
KP 5	73	F0,73
KP 6	74	F0,74
KP 7	6C	F0,6C
KP 8	75	F0,75
KP 9	7D	F0,7D
]	5B	F0,5B
;	4C	F0,4C
’	52	F0,52
,	41	F0,41
.	49	F0,49
/	4A	F0,4A