## **PS2** Keyboard

## About the guide

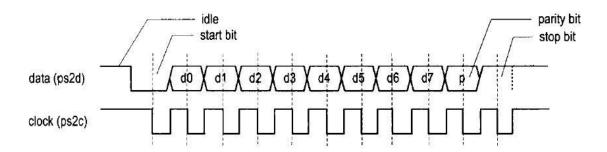
Aim of this guide is to make you familiar with ps2 protocol and building a keyboard module in verilog.

## Protocol

Protocol is synchronous using two pins one clock pin another data pin. Data is supposed to be sampled at falling edge of clock pin. Data is sent in packet of 11 bits. First start bit "0", next 8 bits data, next parity then ending bit "1".

Eight data bits are distinct for each key.

When you press a key keyboard continuously send data bits corresponding to that key continuously until it is pressed then keyboard sends ending byte 'f0'.



ESC F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12   76 05 06 04 0C 03 0B 83 0A 01 09 78 07	<b>1</b> 2075
$\begin{bmatrix} \cdot & \\ 0E & 16 \\ 16 & \\ 1E & 26 \\ 25 & 2E \\ 25 & 36 \\ 3D \\ 3E \\ 3E \\ 3E \\ 46 \\ 45 \\ 45 \\ 4E \\ 55 \\ 4E \\ 55 \\ 46 \\ 45 \\ 4E \\ 4E \\ 55 \\ 46 \\ 4E \\ 55 \\ 46 \\ 45 \\ 4E \\ 4E \\ 4E \\ 55 \\ 46 \\ 4E \\ 4E \\ 4E \\ 4E \\ 4E \\ 4E \\ 4E$	<b>→</b> <b>E0</b> 74
TAB Q W E R T Y U I O P [[{]} 1] 1]   0D 15 1D 24 2D 2C 35 3C 43 44 4D 54 5B 5D	<b>€0</b> 6B
$ \begin{array}{c} \textbf{CapsLock} \\ \textbf{58} \\ \textbf{1C} \\ \textbf{1B} \\ \textbf{23} \\ \textbf{2B} \\ \textbf{2B} \\ \textbf{34} \\ \textbf{33} \\ \textbf{3B} \\ \textbf{42} \\ \textbf{4B} \\ \textbf{4C} \\ \textbf{52} \\ \textbf{5A} \\ \textbf{5B} \\ $	<b>E</b> 0 72
$ \begin{array}{c c} & \textbf{Shift} \\ \hline & \textbf{12} \\ \hline & \textbf{12} \\ \end{array} \begin{array}{c} \textbf{Z} \\ \textbf{1A} \\ \hline & \textbf{22} \\ \end{array} \begin{array}{c} \textbf{C} \\ \textbf{21} \\ \hline & \textbf{2A} \\ \end{array} \begin{array}{c} \textbf{V} \\ \textbf{B} \\ \textbf{32} \\ \hline & \textbf{31} \\ \end{array} \begin{array}{c} \textbf{M} \\ \textbf{M} \\ \textbf{AA} \\ \hline & \textbf{41} \\ \hline & \textbf{49} \\ \hline & \textbf{4A} \\ \hline & \textbf{A} \\ \hline & \textbf{59} \\ \end{array} \right) $	
CtrlAltSpaceAltCtrl141129E011E014	

## For example if 'a' is pressed then 1c will be transmitted multiple times and bytes received are 1c,1c,1c.....1c,f0.If shift then a is pressed then bytes transmitted will be 12,12...12,1c,....1c,f0,12,.....,12,f0.

Verilog Code:-

module keyboard( input wire clk, // Clock pin form keyboard //Data pin form keyboard input wire data, output reg [7:0] led //Printing input data to led ); reg [7:0] data\_curr; reg [7:0] data\_pre; reg [3:0] b; reg flag; initial begin b<=4'h1; flag<=1'b0; data\_curr<=8'hf0;</pre> data\_pre<=8'hf0;</pre> led<=8'hf0; end always @(negedge clk) //Activating at negative edge of clock from keyboard begin case(b) 1:; //first bit 2:data\_curr[0]<=data; 3:data\_curr[1]<=data; 4:data\_curr[2]<=data; 5:data\_curr[3]<=data; 6:data curr[4]<=data; 7:data\_curr[5]<=data; 8:data\_curr[6]<=data; 9:data\_curr[7]<=data; 10:flag<=1'b1; //Parity bit 11:flag<=1'b0; //Ending bit endcase if(b<=10) b<=b+1; else if(b==11) b<=1; end always@(posedge flag) // Printing data obtained to led begin if(data\_curr==8'hf0) led<=data\_pre;</pre> else data\_pre<=data\_curr;</pre> end endmodule

Tips:-

1) Add "NET "[Keyboard clock pin]" CLOCK\_DEDICATED\_ROUTE = FALSE;" to your constraint file. 2) Pull up both clock and data pins in your constraint file.

3) Scan code send by keyboard are not ascii code you must manipulate it according to your use.