PS/2 Keyboard Controller

The PS/2 controller can be used to communicate with a PS/2 keyboard . It provides an interface to the PS/2 protocol, handling the data transmission and the timing control.

The PS/2 controller interface consists of the PS/2 clock and the PS/2 data input(11 bits). The 11-bit data port is for receiving data from the PS/2 device . All data is transmitted one byte at a time and each byte is sent in a frame consisting of 11.

These bits are:

-1 start bit.This is always 0;

-8 data bits, least significant bit first;

-1 parity bit (odd parity);

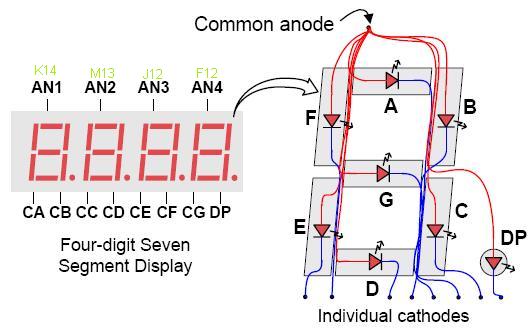
-1 stop bit.  This is always 1;

The key board is connected to the FPGA board through the PS/2 port.

|  |  |  |
| --- | --- | --- |
| Male  http://www.computer-engineering.org/ps2protocol/spindin.JPG  (Plug) | Female  http://www.computer-engineering.org/ps2protocol/spindin1.JPG  (Socket) | **6-pin Mini-DIN (PS/2):**  1 - Data  2 - Not Implemented  3 - Ground  4 - Vcc (+5V)  5 - Clock  6 - Not Implemented |

When a button is pressed on the keyboard, it will generate a 11-bit code from which the 8 data bits are decoded and sent to the 7-segment LED display.The character pressed on the keyboard will then appear on the LED display.

The 7-segment display is 0-active.



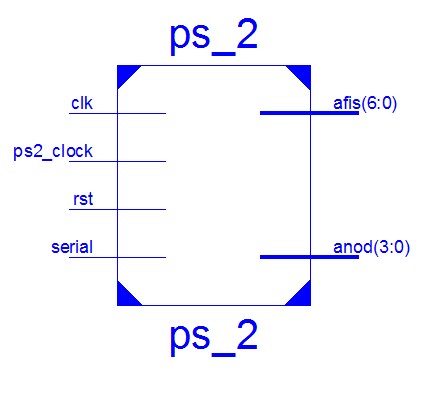
For example when the key ’A’ is pressed the keyboard will send the following code ”00011100” along with the start bit,parity bit and stop bit.The code is then decoded and the letter ’A’ is shown on the LED display.

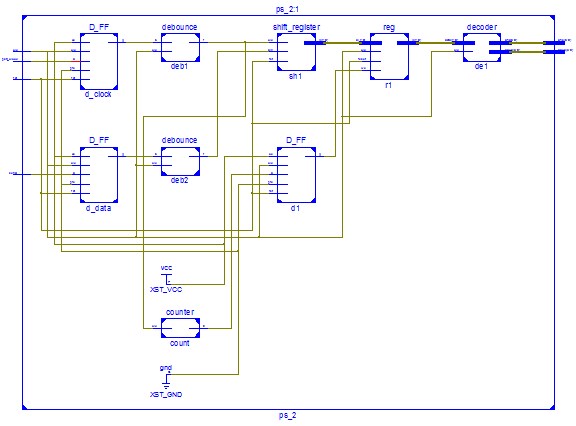
**Synchronization Timing:**

The timing control necessary for the PS/2 communication is handled by the controller.The ps2\_clock and the serial(data input) are sent by the keyboard.The ps2\_clock it’s sent between 10kHZ and 30kHz depending on the keyboard. Clock source on Basys2 FPGA Board has a 50 MHz source.The serial enters the d\_data flip-flop , it’s synchronized with the internal clock of the FPGA board and the new serial will be serial\_data and ps2\_clock enters the d\_clock flip-flop , it’s synchronized with the internal clock of the FPGA board and the new serial will be ps2\_clock\_sync.

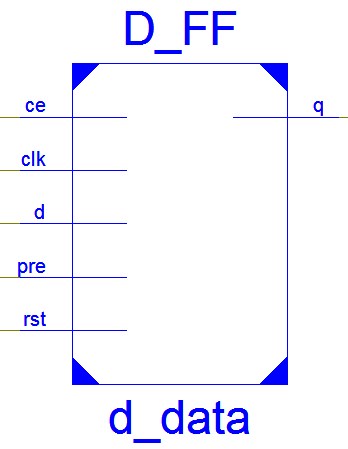
After these two signals are synchronized they need to be debounced (using deb1 respectively deb2) before they are further used ,the debounced ps2\_clock\_sync becomes ps2\_clock\_debounced and the debounced serial\_data\_debounced.

**PS/2 Keyboard controller components:**

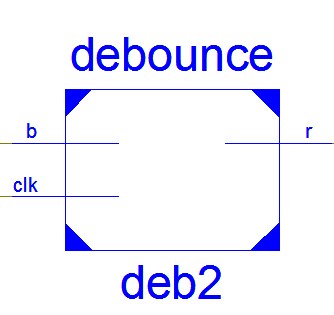
****



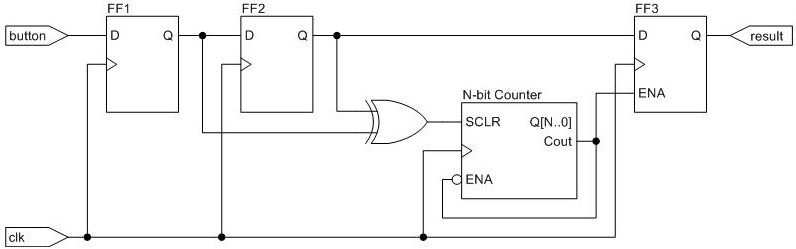
-d\_clock and d\_data flip-flops are used to synchronise the ps2\_clock signal, respectively the serial signal;



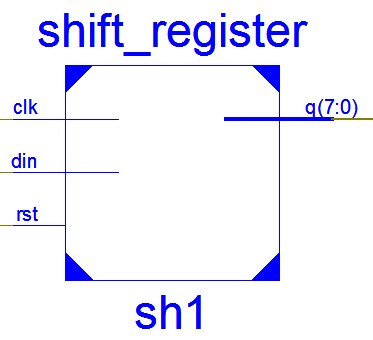
-deb1 and deb2 are used to debounce the signals which are outputed from d\_clock, respectively d\_data;



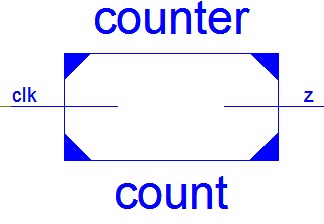
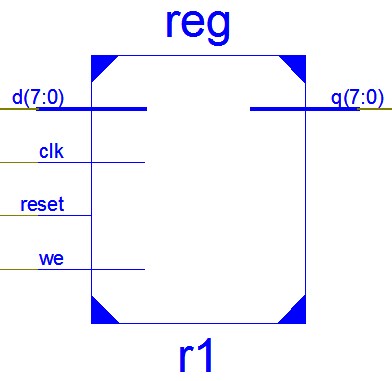
The circuit continuously clocks the button’s logic level into FF1 and subsequently into FF2.  So, FF1 and FF2 always store the last two logic levels of the button.  When these two values remain identical for a specified time, then FF3 is enabled, and the stable value is clocked through to the result output.



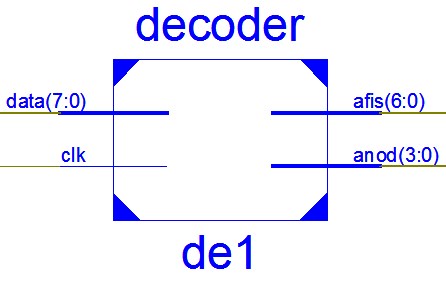
-sh1 receives the debounced data from the deb2 bit by bit on the rising edge of the ps2\_clock\_debounced;



-count (counter in range 0-10) receives ps2\_clock\_debounced from deb1 and after it counts 11 clock pulses, it sends a signal to d1 flip-flop which delays it and it’s sent further to enable r1 for receiving the 8 data bits from sh1. When the r1 (register) is full, the code is sent instantaneously to de1;

-de1 (decoder) compares the code received from the register (r1) with a specific code for each character. If it matches, the character will be displayed on the 7-segment LED else the character „-” will be displayed.



**Pins** Connections:

NET "clk" LOC="b8"; -- input signal clock

NET "rst" LOC="a7"; -- input signal reset

NET "anod<3>" LOC="k14"; --output signal

NET "anod<2>" LOC="m13"; --output signal

NET "anod<1>" LOC="j12"; --output signal

NET "anod<0>" LOC="f12"; --output signal

NET "afis<6>" LOC="l14"; --output signal (G)

NET "afis<5>" LOC="h12"; --output signal (F)

NET "afis<4>" LOC="n14"; --output signal (E)

NET "afis<3>" LOC="n11"; --output signal (D)

NET "afis<2>" LOC="p12"; --output signal (C)

NET "afis<1>" LOC="l13"; --output signal (B)

NET "afis<0>" LOC="m12"; --output signal (A)

NET "ps2\_clock" LOC="b1"; -- input signal keyboard clock

NET "serial" LOC="c3"; --input signal keyboard data

**Instructions:**

1) You need to connect the FPGA to the power supply and to the keyboard.

2) You should Power On the FPGA and upload the proogram on the FPGA chip.

3) To change the character displayed on the 7-segment LED you have to press a button from the keyboard.

**Upgrade Possibilities:**

This controller can be upgraded by adding an error detection and correction code and Host-to-Device Communication(this is opposite of what occours in device-to-host communication).

Authors:

Giurgiuman Andrei

Bota Paul

Group: 30412