Lab Assignment 2

Interfacing Keyboard with FPGA Board

Oskar Westman
elt13owe@student.lu.se

What happens when you press/release a key?

- **Key pressed = make code is generated.**
- **Key released = break code is generated.**

key	make	break
A	‘1C’h	‘F0’h ‘1C’h
B	‘32’h	‘F0’h ‘32’h
C	‘21’h	‘F0’h ‘21’h

How does the keyboard send data?

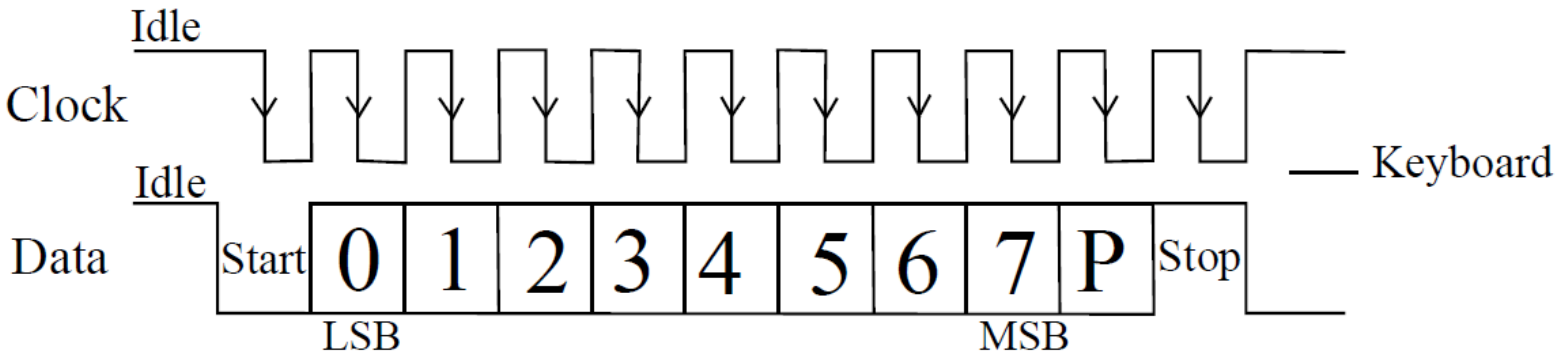
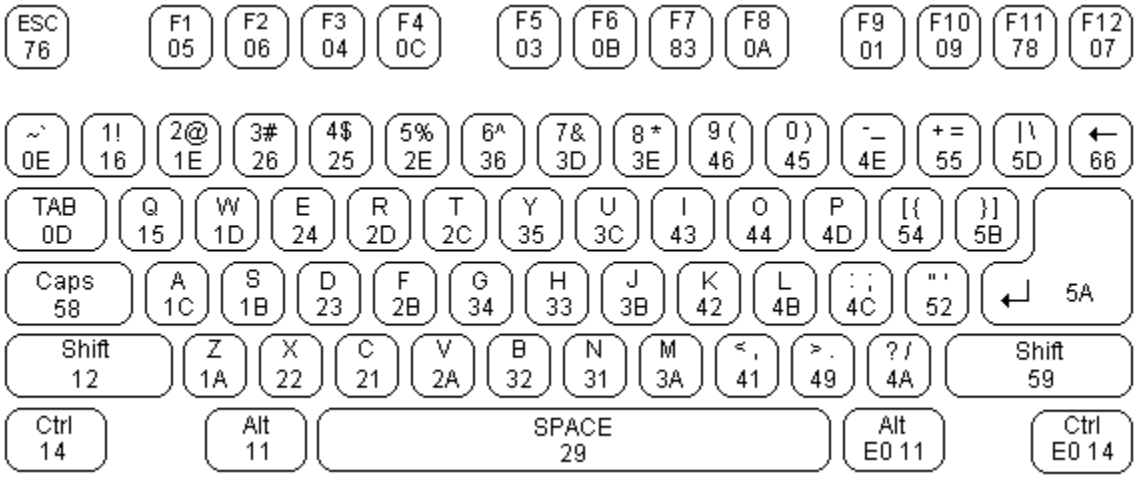


Figure 1: Key Board interface waveform

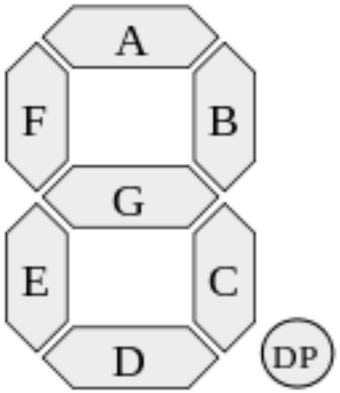
Scan codes

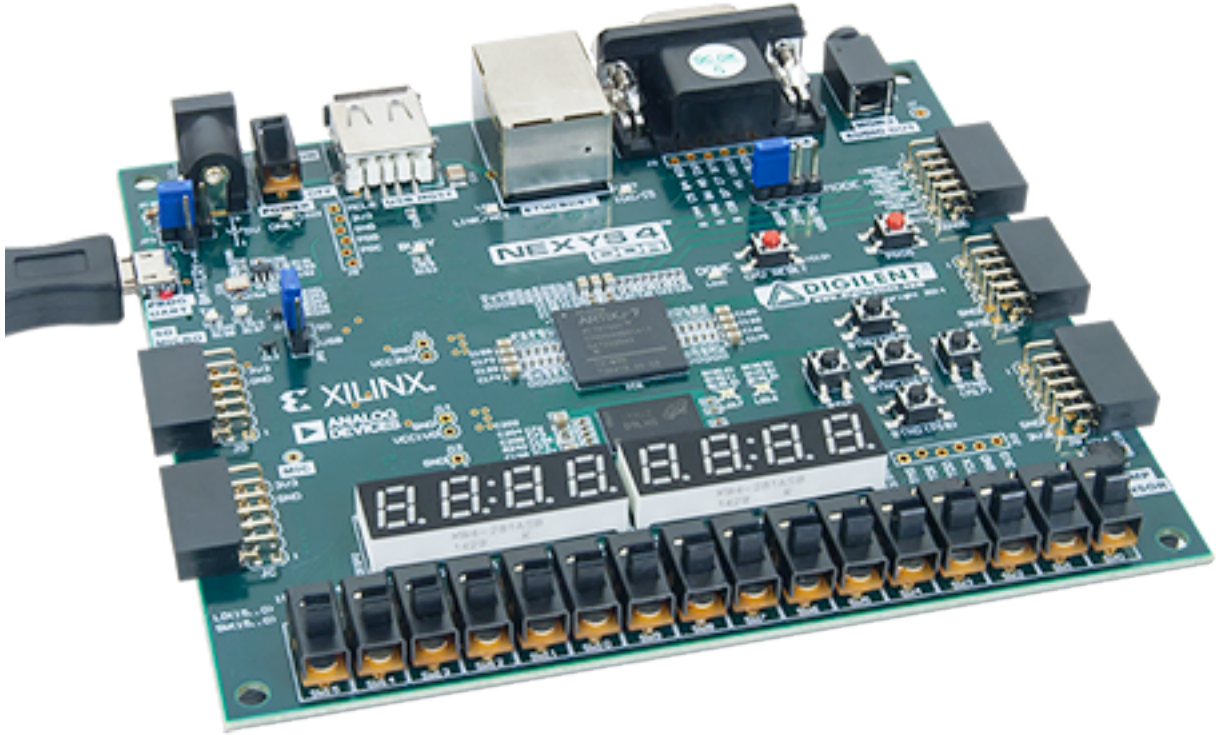
- **Each key is assigned a unique scan code.**



FPGA

- **Pressed key shows on FPGA -- 7 SEGEMENT DISPLAY**





Demostration

- Check the reset condition, all display should be switched off at reset.
- Pressing "123456" one by one, the display should look the following:

Seven Segement Display

DISPLAY OFF				press (sw0) reset
			1	press 1
		1	2	press 2
	1	2	3	press 3
1	2	3	4	press 4
2	3	4	5	press 5
3	4	5	6	press 6

Main Processes

- **Synchronizing Keyboard with FPGA.**
- **Detection of falling edge of keyboard Clock. (why?)**
- **Storing of relevant Scan Code from keyboard (meaning?).**
- **Display of ’numbers’ keys on Seven Segment.**
- **Shift the previous key left and display the current number when the next number key is pressed.**

Why falling edge?

1.2 Reading Keyboard Scan Codes Through the PS/2 Interface on the Board

The keyboards provided in the lab has a USB interface, however, don't worry, we won't ask you to implement a USB host, which is a very complex design. In our case fortunately the FPGA board has a USB host (PIC microcontroller) implemented already. This USB host converts the keyboard interface to old PS/2 format interface, and we will design the controller for this interface.

The old PS/2 PC keyboard transmits data in a clocked serial format consisting of a start bit, 8 data bits with least significant bit (LSB) first, an odd parity bit and a stop bit. The clock signal is only active during data transmission. The generated clock frequency is usually in the range of 10 - 30 kHz. Each bit should be read on the falling edge of the clock.

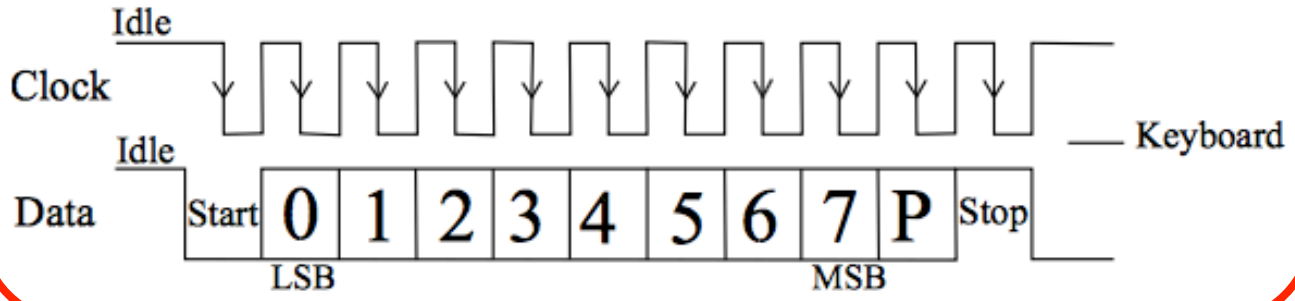
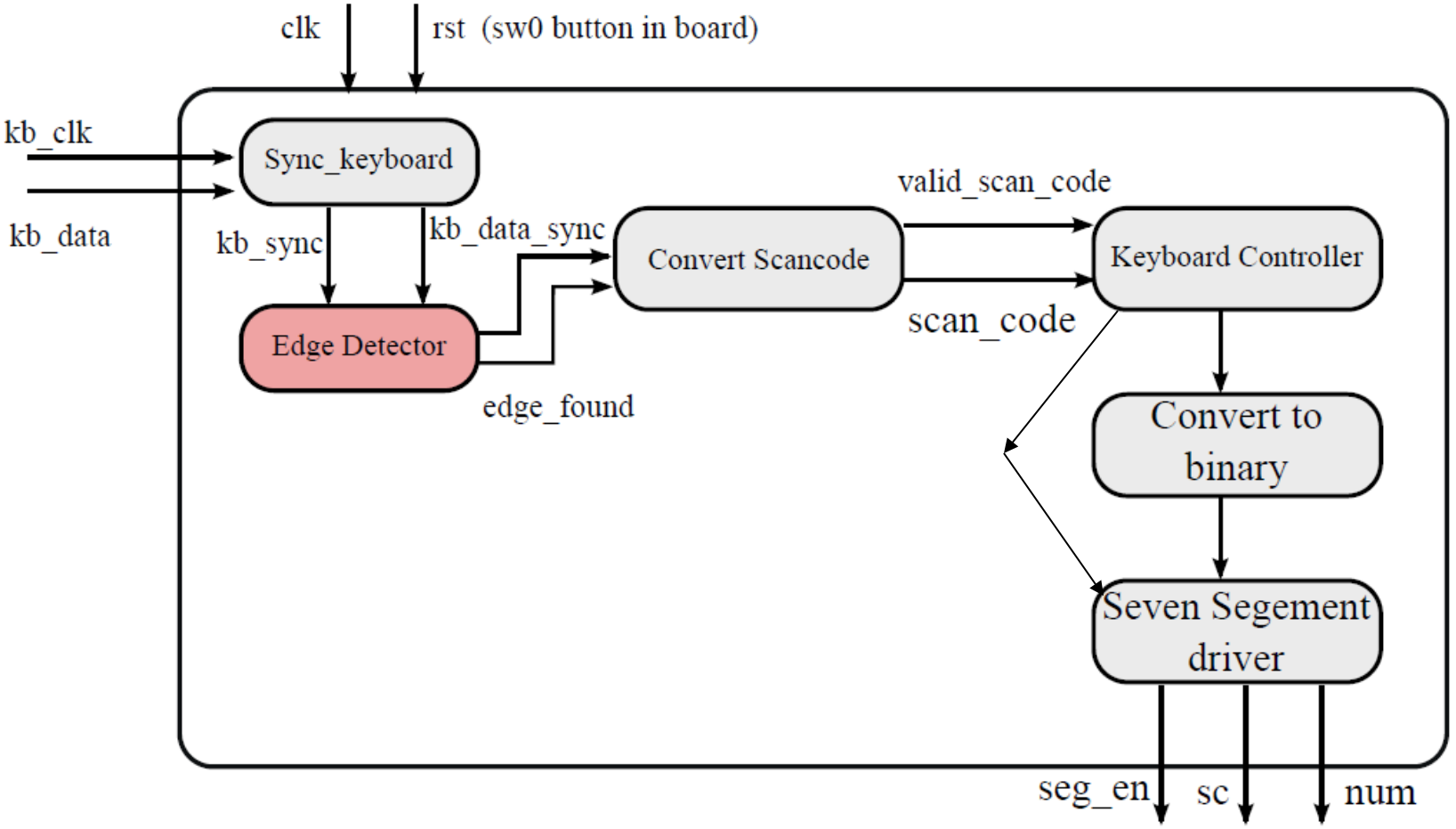
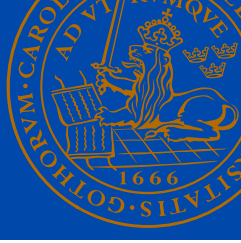


Figure 1: Key Board interface waveform

Tasks


clk
rst (sw0 button in board)
kb_clk
kb_data
Sync_keyboard
kb_sync
kb_data_sync
Edge Detector
edge_found
Convert Scancode
valid_scan_code
scan_code
Keyboard Controller
Convert to binary
Seven Segement driver
seg_en
sc
num

Advice



- **Build your own test bench for each component and TEST IT!**
- **Work your way through the components, confirm function and continue.**
- **Document your work along, present to the TA.**
- **Bit-shift is a good operation in this lab. Read about it in the book or google it!**
- **NO LATCHES!**

VHDL File Lists

File Name	Function
keyboard_top.vhd	Top level integration file
sync_keyboard.vhd	Synchronize keyboard data
edge_detector.vhd	Edge detection circuit
convert_scancode.vhd	Convert serial data to parallel
keybaord_ctrl.vhd	Keyboard controller
convert_to_binary.vhd	Convert scan code to binary
binary_to_sg.vhd	Seven segment driver
keyboard_top.xdc	Pin mapping to FPGA board.
tb_pkg.vhd	Required for some functions in testbench
tb_keyboard.vhd	test bench to drive stimulus
input.txt	Keys to be sent to design via testbench.