

# Implementing a Keypad Controller on an IGLOO FPGA

## 1.0 Introduction

The main objective of this design is to detect a key pressed event. A standard 6-rows-by-3-columns scan keypad logic has been designed and implemented.

## 2.0 General Implementation Overview

The key press will be stored in three 8-bit registers which can be read out/displayed using a Two-Wire Serial Interface provided. The design may be modified for active low key detection or active high key detection as desired.

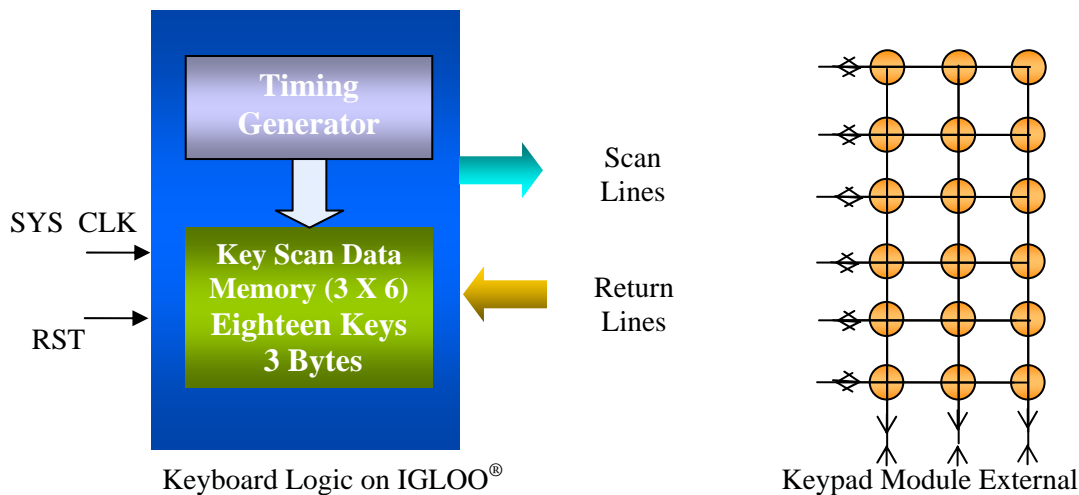


Figure 1. Keypad Driver

## 3.0 Reference Design Implementation

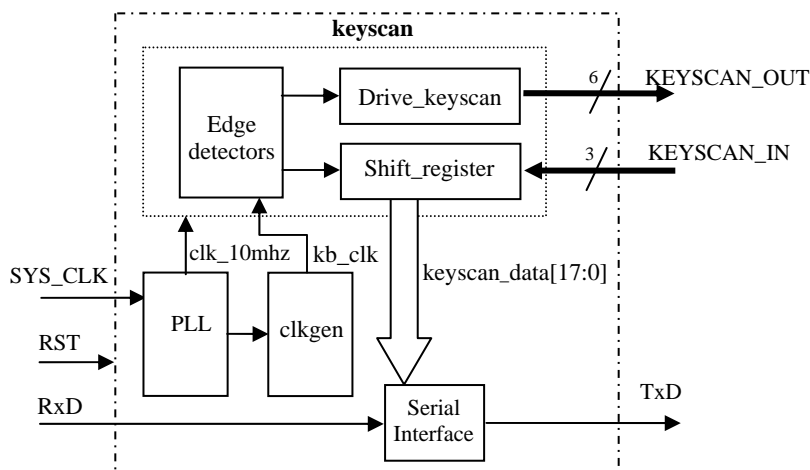


Figure 2. Keypad Driver Logic Block Diagram

A 3-bit counter is used to generate scan out signals in a one-hot encoded fashion on the rising edge of the kb\_clk. The return lines are captured on the falling edge of the kb\_clk. The data is latched on the last sequence of the scan refresh cycle.

Special serial interface logic is provided to recover the latched data and transmit on the TxD line. This is provided so that the key press event can be polled through a software interface as desired. The serial interface block diagram is depicted below in Figure 3.

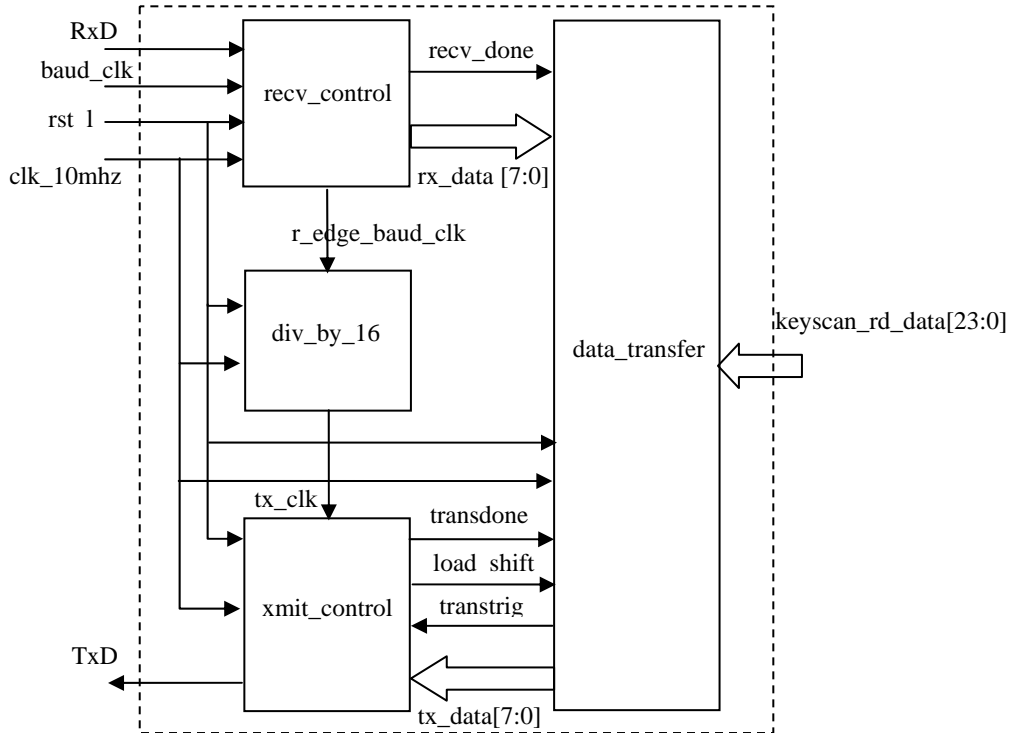


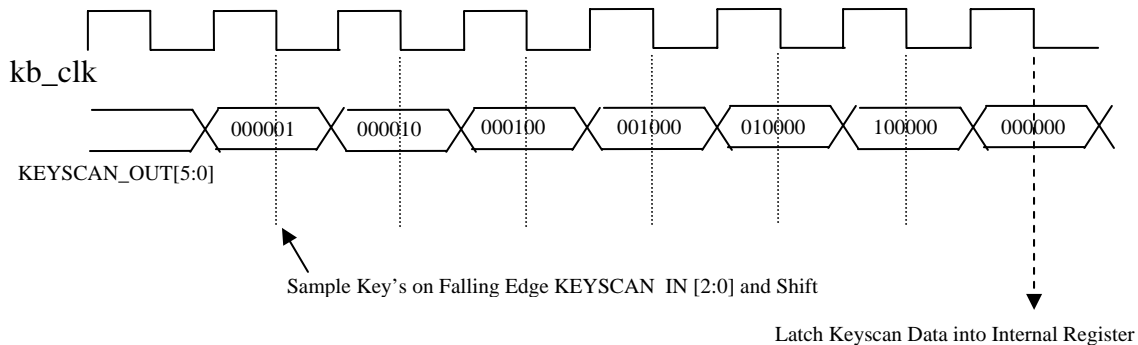
Figure 3. Serial I/F for Keypad Driver

#### 4.0 I/Os

The following table describes the main I/Os in the design:

Signal	Input/Output	Description
KEYSCAN_OUT[5:0]	Output	6 Scan Lines – Output
TXD	Output	Transmit – Serial Output
KEYSCAN_IN[2:0]	Input	3 Return Lines – Input
RXD	Input	Receive – Serial Input
SYS_CLK	Input	System Clock – 20Mhz.
SYS_RESET_N	Input	Master Reset – Active Low

## 5.0 Waveforms



The row counter is incremented on the rising edge of the keyboard clock (kb\_clk) and the return lines are sampled on the falling edge of the kb\_clk, as shown above. The final data is latched into an internal 18-bit register, on the falling edge of the seventh count.

## 6.0 Conclusion

There are several control functions that an FPGA can perform, enabling the integration of multiple functions on a chip and thus reducing the board space, power requirements, and bill-of-materials (BOM) cost. Keyboard control is an essential function of many portable devices in the market. This design can be targeted at any application requiring a keyboard interface in a matrix form. The row-column matrix combination can be easily changed to cater to the designer's requirements. The scan method can also be integrated into an interrupt-based logic if interfacing to a processor.

## Appendix A – Keypad Controller Design Example

### Design Files Summary

Files	Functionality
clk_by_2.v	Divides input clock by 2 – Toggle F/F
clk_gen.v	Clock Generator
data_transfer.v	Transfers the data according to commands given by PC
recv_control.v	This block receives data serially on RxD
xmit_control.v	This block transmit data serially on TxD
serial.v	This block connects xmit_control, recv_control & data_transfer
div_by_5.v	Divide by 5 – Derived clock for internal use
div_by_16.v	Divide by 16 block for serial communication – Baud Clock
keyscan.v	This block scans keys
keypad_ip.v	This block interconnects keyscan and serial block
top_keypad_ip.v	This block interconnects keypad_ip and PLL
top_tb.v	Test_bench for keypad_ip

## About Ishnatek

Ishnatek offers FPGA design and hardware prototyping services. Ishnatek offers Design and Verification Services for embedded solutions. We have IPs such as 8031, RTC, Timers, Enhanced PWM, UART/SIO/IrDA, I2C, LED Driver and Key scan, Parallel Port, ECP/EPP, etc., which can be building blocks for your embedded controller solutions.