

GITAM

(DEEMED TO BE UNIVERSITY)

**TITLE: VLSI PROJECT**

Subtitle: KEYPAD SCANNER USING FPGA KIT

**NAME:** GREESHMA C K

**REG.NO:**BU21EECE0100523

**NAME:** D.GOWHASRI

**REG.NO:** BU21EECE0100486

**NAME:** B.SATHISH CHANDRA KUMAR

**REG.NO:**BU21EECE0100338

**NAME:**A.SRAVANI

**REG.NO:**BU22EECE0100447

**UNDER THE GUIDANCE OF :**

DR.ARUN KUMARAssociate professor, ECE department ,GITAM university BLR

DR.KARTHICK.SAssociate professor, ECE department ,GITAM university BLR

**INTRODUCTION:**

Keypad scanners are commonly used in various applications like security systems, calculators, and user interfaces where a set of buttons needs to be read and processed. Implementing a keypad scanner using an FPGA (Field Programmable Gate Array) offers the benefits of flexibility, parallel processing, and customizability, making it an attractive choice for embedded system designers.

### **KEYPAD MATRIX BASICS:**

A typical keypad is arranged in a matrix format, usually a 3x3 or 4x4 grid. Each key connects a specific row to a specific column. By scanning this matrix, you can detect which key is pressed. The process involves setting columns one by one to a low state and reading the row states to determine the pressed key.

### **FPGA OVERVIEW:**

An FPGA is a type of digital logic chip that can be programmed to perform any logic function. Unlike microcontrollers, which execute sequential instructions, FPGAs operate on a parallel architecture, making them suitable for tasks that require concurrent operations. FPGAs are widely used in applications requiring high-speed data processing and custom hardware configurations.

### **Why to Use an FPGA for Keypad Scanning?**

1. **Parallel Processing:** FPGAs can handle multiple tasks simultaneously, which is useful for scanning multiple keys at once.
2. **Customizability:** You can design custom logic tailored to your specific application needs.
3. **Performance:** FPGAs can achieve higher processing speeds compared to microcontrollers for certain tasks.
4. **Reconfigurability:** FPGAs can be reprogrammed as needed, allowing for updates and changes without hardware modifications.

### 

### 

### **KEYPAD SCANNER IMPLEMENTATION:**

**Implementing a keypad scanner on an FPGA involves the following steps:**

1. **Connecting the Keypad:** Physically connecting the rows and columns of the keypad matrix to the FPGA I/O pins.
2. **Writing HDL Code:** Using a hardware description language (HDL) such as Verilog or VHDL to describe the scanning logic.
3. **Synthesizing the Design:** Converting the HDL code into a format that can be loaded onto the FPGA.
4. **Assigning Pins:** Mapping the FPGA pins to the keypad matrix connections.
5. **Programming the FPGA:** Loading the synthesized design onto the FPGA and testing the implementation.

**OBJECTIVES:**

The main objectives of implementing a keypad scanner using an FPGA are as follows:

1. **Understand Keypad Matrix Interfacing**
2. **Design Keypad Scanning Logic**
3. **Synthesize and Implement on FPGA**
4. **Real-Time Key Detection**
5. **Enhance FPGA Design Skills**
6. **Optimizing the Performance**
7. **Document the Design Process**

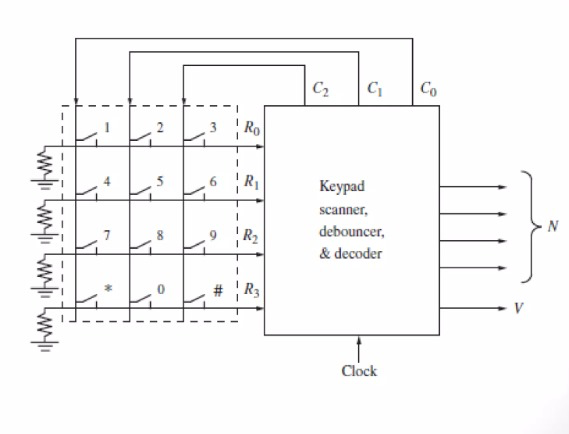
**SOFTWARE:**

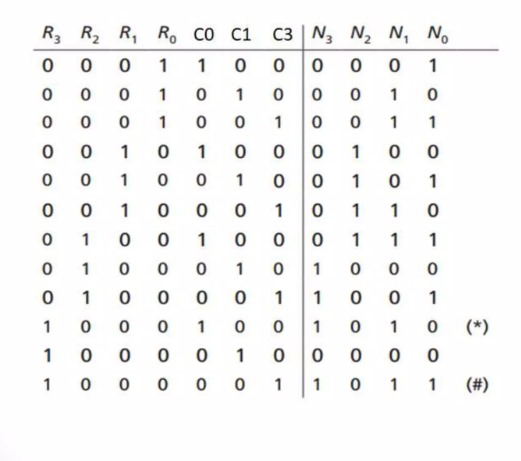
\*Vivado Design Suite (for Xilinx FPGAs)

**HARDWARE:**

\*Xilinx FPGA

**BLOCK DIAGRAM:**

****

**TRUTH TABLE:**

**CODE:**

module keypad\_scanner(n3, n2, n1, n0, r0, r1, r2, r3, c0, c1, c2);

input r0, r1, r2, r3, c0, c1, c2;

output n3, n2, n1, n0;

assign n3=(r2 & ~c0) | (r3 & ~c1);

assign n2 = r1 | (r2 & c0);

assign n1 = (r0 & ~c0)|(r2 & c2)|(~r1 & ~r0& c0);

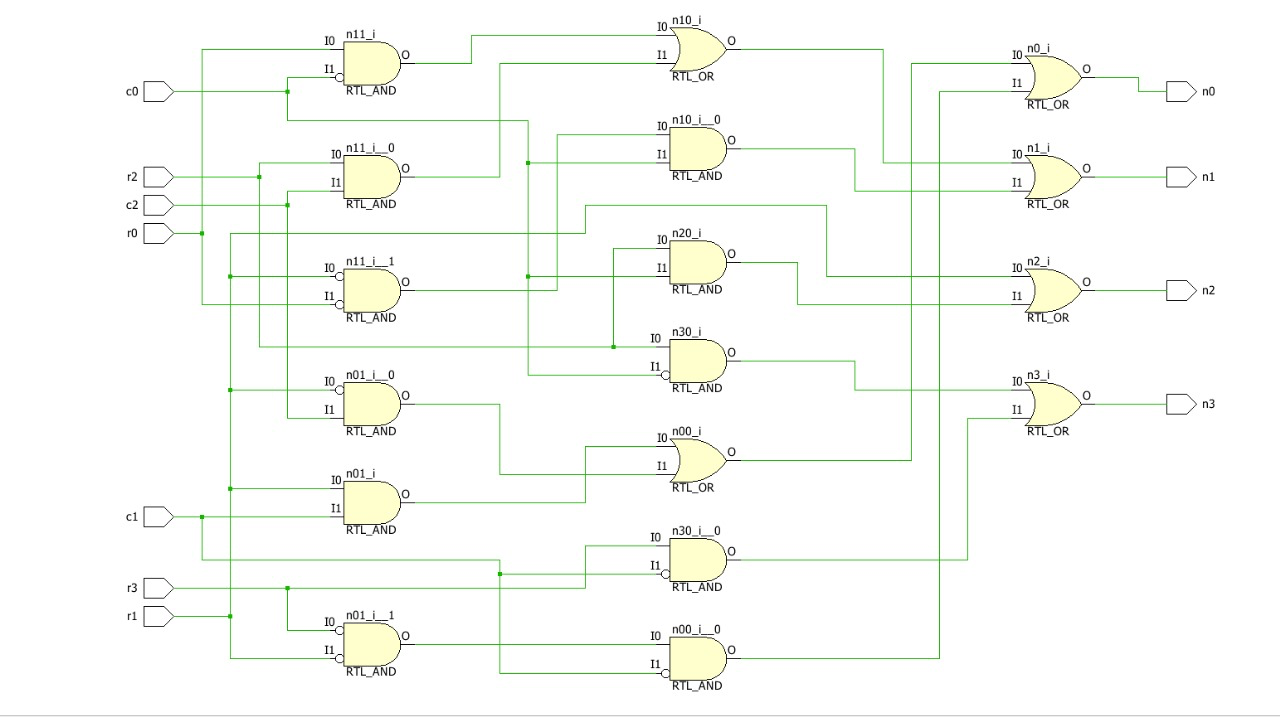
assign n0 = (r1 & c1)|(~r1 & c2)|(~r3 & ~r1 & ~c1);

endmodule

**SIMULATION:**



**RTL SCHEMATIC SIMULATION:**

****

**FPGA Implementation:**

Input - R[1000], C[100] | Output - N[1000]

**DRIVE LINK:**

https://drive.google.com/file/d/1AlzCl5nDDxhg9CcZAgquGCocD86PXydh/view?usp=sharing

**CONCLUSION:**

The implementation of a keypad scanner using an FPGA provides an efficient and flexible solution for reading input from a keypad matrix. This approach leverages the strengths of FPGAs, such as parallel processing and customizability, to create a robust and high-performance keypad scanning system. Whether for educational purposes or practical applications, understanding and implementing this system can be a valuable addition to your skill set in digital design and embedded systems.