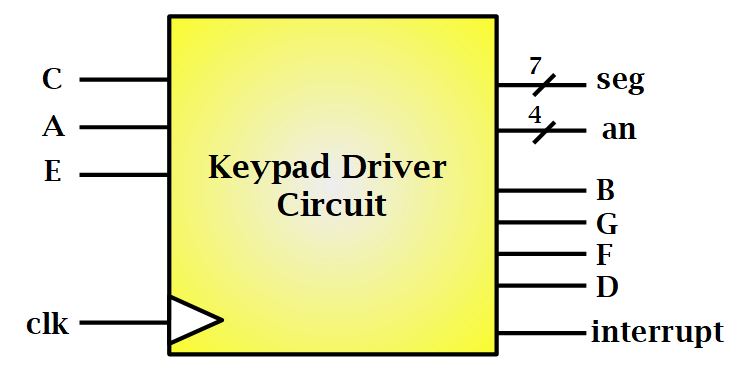
Peripheral Assignment 3 – Keypad Driver

Marina Dushenko & Luis Gomez

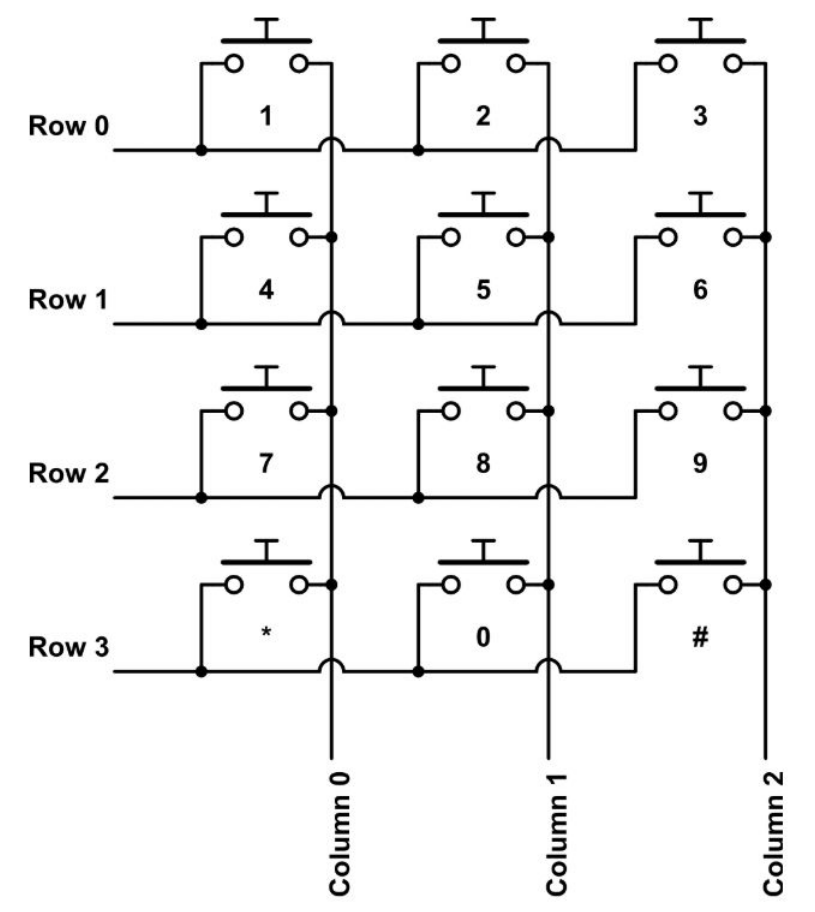
**Black Box Diagram**



**Figure 1:** Black Box Diagram for Key Pad Driver

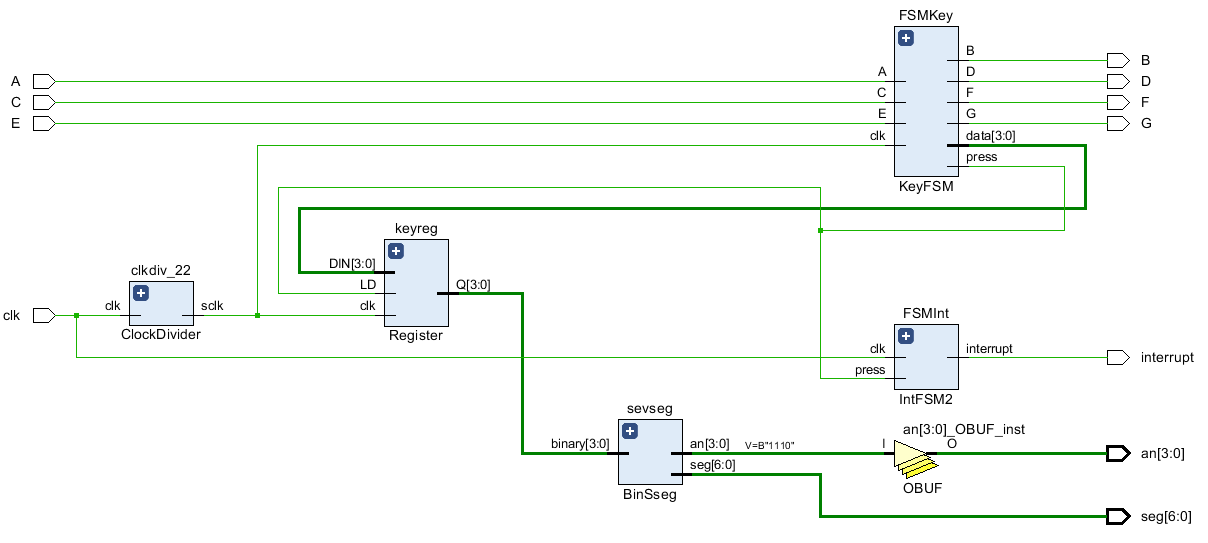
**Behavior Description**

The following program expands the capabilities of the RAT architecture by adding an additional peripheral. This peripheral receives a 1-bit button-press input and providin back to the user output via the Basys3 board. The complexity of the keypad was reduced by the carful arrangements of rows and columns that are demonstrated in the figure 5 below. The keypad itself is a passive device and detects user input based on specific row and column placement. Upon a press, the keypad sends an interrupt to the RAT MCU, and specific instruction is executed. This allows the peripheral to only be on when a user input is present. The program then cycles through the specific row and column until the exact value is reached. This makes the peripheral more user-friendly, and less burdensome then constantly cycling through all possible values of the keypad.



**Figure 2:**  Keypad Circuit

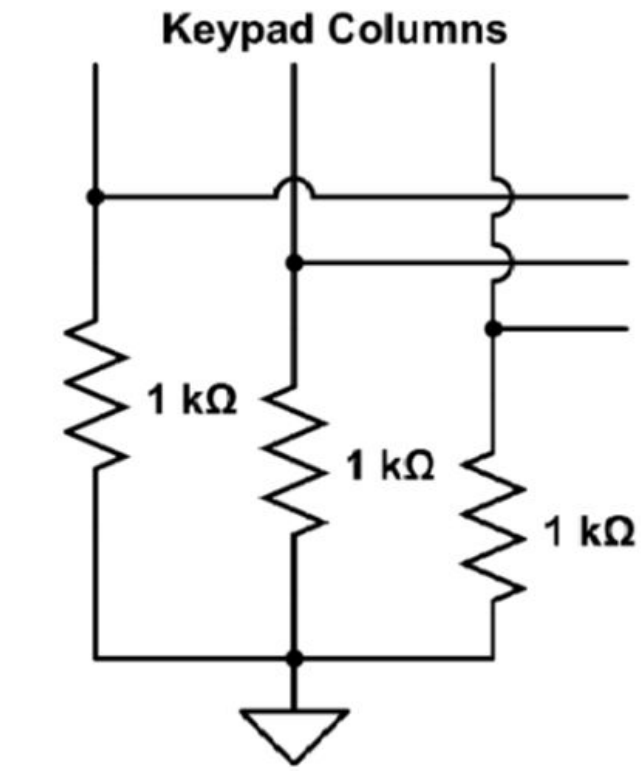
**Structural Design**



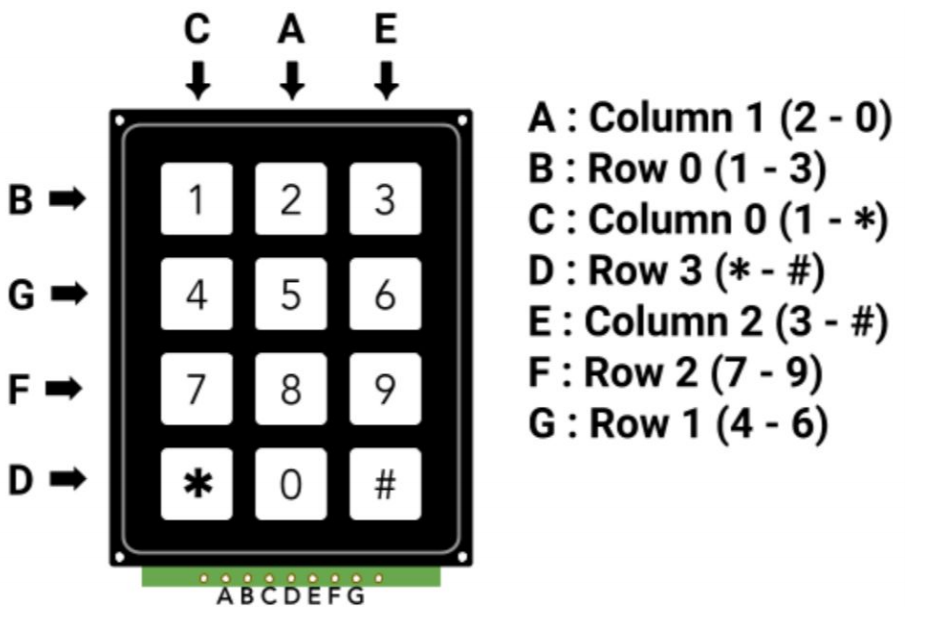
**Figure 3:** Keypad Elaborated Design Schematic

**Specifications**

Referencing figure 4 below it is essential that one connects a pull-down resistor to the keypad prior to the Basys3 board. The pulldown resistor prevents the input signal from floating. This prevents the keypad driver from receiving unpredictable feedback. The remainder of the keypad is mapped as shown in figure 5. The outputs of the Basys board correspond with input ports of the keypad driver. When a specific button is activated the driver send an interrupt signal to the RAT MCU. This signal can be observed on an oscilloscope by a 60ns pulse. The pulse itself runs off a 100MHz clock. The driver only allows one button to be pressed at a single time. Therefore, before the driver allows another interrupt to pass, the previous button must be released. The program also comes standard with a debounce capability. This slows down the rate at which the program detects the press and treats the input as a single value.



**Figure 4:** Pull down Resistor model



**Figure 5:** Keypad Pinout Schematic

**SystemVerilog Source Code**

**Example Use Code**