KEYPADA - Keypad Controller

Summary

This document provides detailed reference information with respect to the non-Wishbone Keypad Controller peripheral component, KEYPADA.

The Keypad Controller component (KEYPADA) provides a simple interface between a host processor and a 16-key (4Row x 4Column) keypad, such as that located on the NanoBoard-NB1. The Controller is fully synchronous.

Features

- Fully synchronous design
- Reset under microcontroller control
- 1 MHz operational frequency
- Continuous scanning of the keypad matrix – no triggering required
- Inherent key debouncing (12.3ms)

Availability

From a schematic document, the KEYPADA component can be found in the FPGA Peripherals integrated library (`FPGA Peripherals.IntLib`), located in the `\Library\Fpga` folder of the installation.

Functional Description

Symbol

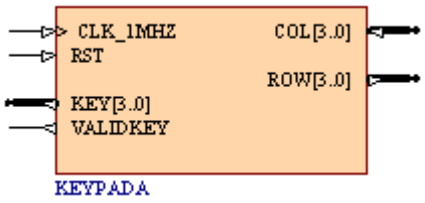

Figure 1. Keypad Controller Symbol.

Pin Description

Table 1. KEYPADA Pin description

Name	Type	Polarity/ Bus size	Description
Control Signals			
CLK_1MHZ	I	Rise	External clock signal. This signal must be 1 MHz in frequency. Clock dividers must be used within the design to achieve this frequency. The initial (undivided) signal can be sourced either from the system (NanoBoard) clock or some other reference clock in the design.
RST	I	High	External reset signal. This signal is issued by the microcontroller in order to clear the VALIDKEY and KEY outputs after reception of valid key data.
Microcontroller Interface Signals			
KEY	O	4	The 4-bit code identifying the actual key on the keypad that has been pressed.
VALIDKEY	O	High	Shows the status of a key on the keypad. This signal remains low until a key on the keypad is pressed and has been validated as such by the Controller's debounce circuitry. Once a 'valid key down' has been recognized, this output will go High and remain High until a reset signal from the microcontroller (on the RST input).
Keypad Interface Signals			
COL	I	4	Value returned by keypad matrix
ROW	O	4	Driving output to the keypad matrix

Hardware Description

Block Diagram

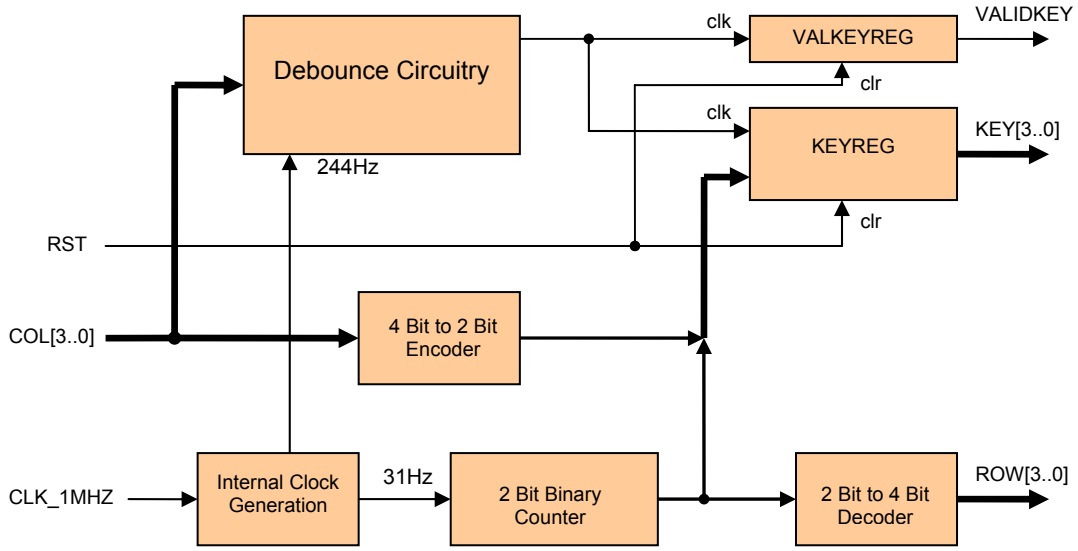

Figure 2. KEYPADA block diagram

Detailed Implementation Schematic

Figure 3 provides the detailed, underlying schematic sheet containing the circuitry used to implement the KEYPADA Controller.

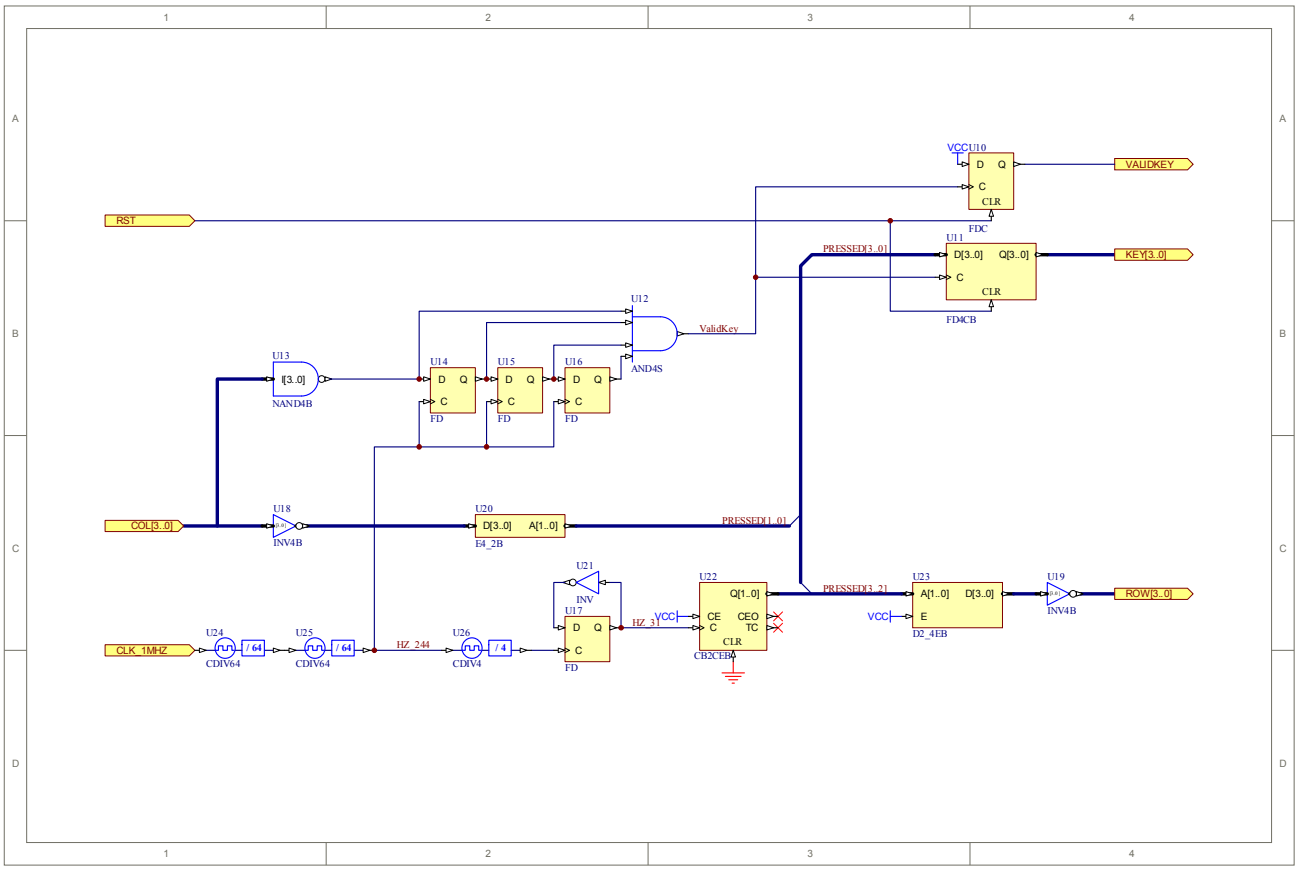

Figure 3. Implementation schematic for the KEYPADA Controller

Operation

Placing the KEYPADA Controller in a Design

The KEYPADA Controller provides a simple interface between a host microcontroller and the Keypad on the NanoBoard-NB1. Figure 4 shows an example of how the device is wired into a design.

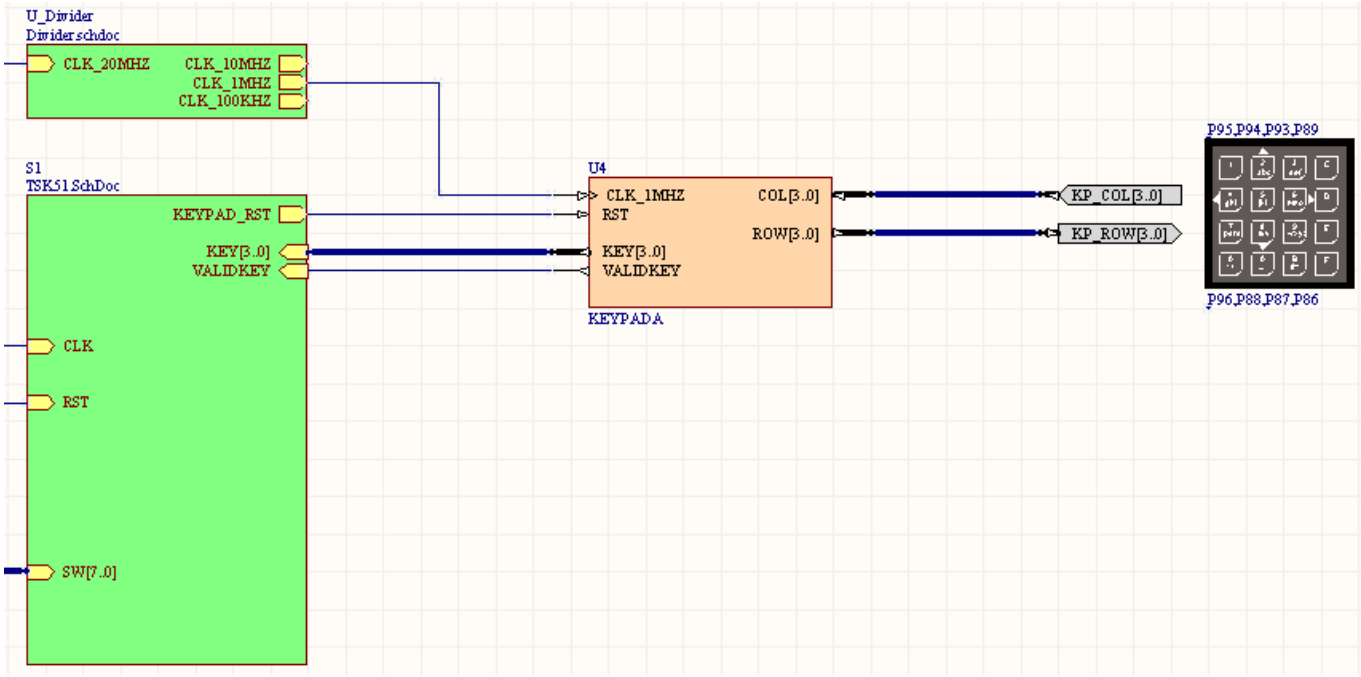

Figure 4. Providing the control interface between a microcontroller and the on-board keypad.

In the example circuit above, the microcontroller, which although not shown is actually a TSK51A_D, polls the VALIDKEY output from the Keypad Controller, which is wired into one of its ports. When the VALIDKEY signal goes High – corresponding to a key being pressed on the Keypad – the microcontroller will read in the 4-bit value on the KEY bus, which identifies the key that has been pressed, and will react by taking the appropriate action, as determined by the embedded code running within.

If the microcontroller used has external interrupt capability, the VALIDKEY signal can be wired directly to an interrupt pin. On reception of an active VALIDKEY signal (High), an interrupt routine would be entered within the embedded code running on the processor and the value on the KEY bus used as parametric input.

Scanning the Keypad Matrix

The Controller tests the keypad matrix for a valid key press, continuously, using the 4-bit ROW and COL buses. The detail of the keypad matrix and how these 4-bit signals are used can be seen in Figure 5.

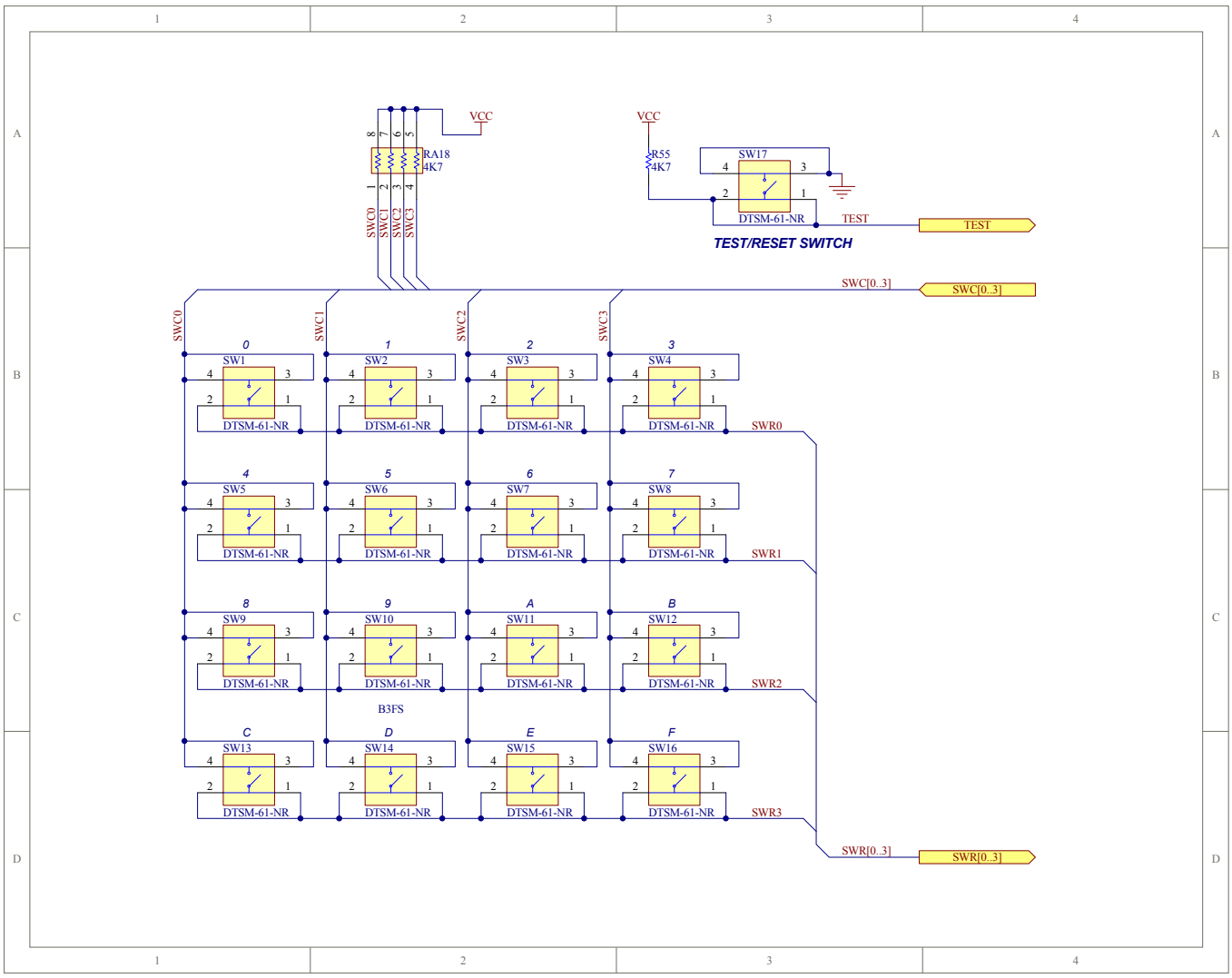

Figure 5. Keypad matrix and distribution of ROW and COL bus signals.

The Controller sends a 4-bit value on the ROW bus, which is simply the decoded output of a 2-bit binary counter, which is then subsequently inverted (Figure 6).

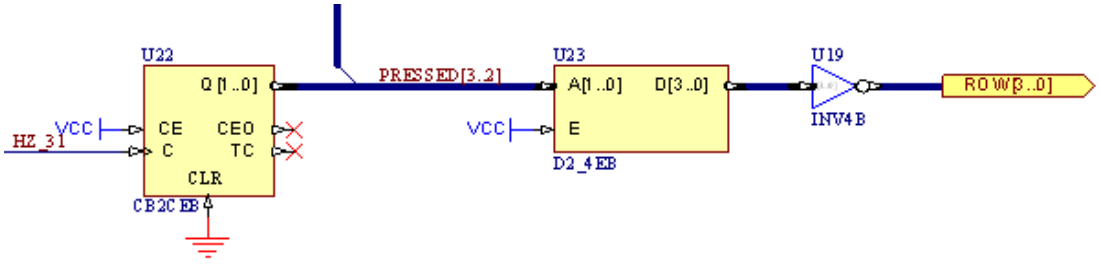

Figure 6. Generating the code to scan the rows of the keypad matrix.

The binary counter is clocked at a rate of 31 Hz. The value on the ROW bus is therefore changed approximately every 32 milliseconds. This provides continuous scanning of the matrix, by effectively testing each row in sequence and in a cyclic fashion.

Table 2 summarises the value sent to the keypad matrix on the ROW bus, based on the output of the binary counter.

Table 2. ROW value generation.

Output of Binary Counter	Row of matrix to be tested	Value sent to keypad matrix on ROW bus
00	0	1110
01	1	1101
10	2	1011
11	3	0111

The state of the keypad matrix is returned on the 4-bit COL bus. If a key has been pressed within a column, the corresponding bit on the bus will be a '0', otherwise the value '1' will be returned.

If no key has been pressed, all columns will return a '1'. The 4-bit value on the COL bus will therefore be 1111. The VALIDKEY signal will remain Low in this case and the Controller will continue with its scanning of the matrix, one row at a time.

If a key is pressed on the Keypad, the value on the COL bus will reflect the column in which the key was pressed. This value is passed to the Controller's debounce circuitry, to validate whether a key has been truly pressed.

The scan rate is approximately eight times slower than the clock used for the internal debounce circuitry. This ensures that the debounce test is completed in adequate time, before the next row of the matrix is tested.

Validating a Key Press - Debounce

The Keypad Controller has inherent key debouncing. The Controller's debounce circuitry (Figure 7) takes as input the 4-bit signal on the COL bus. If the value on the bus remains unchanged for at least 3 cycles of a 244 Hz clock, the key press is deemed to be a valid one.

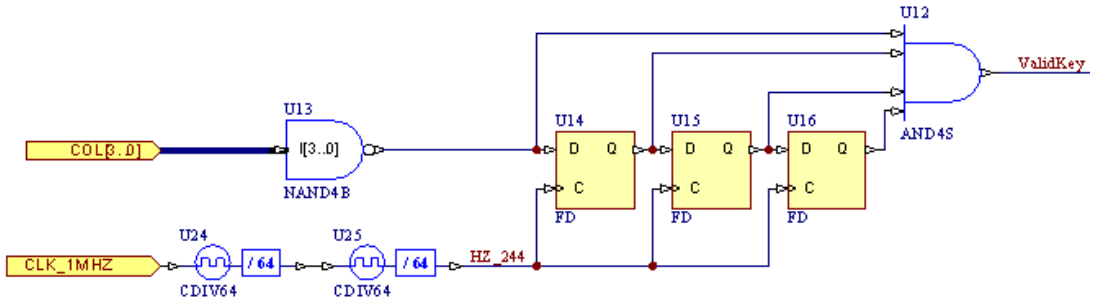

Figure 7. Debounce circuitry used to validate a key press.

Once a 'valid key down' situation has been recognized, the output of the debounce circuit (a 4-input AND gate) will go High. This signal is used, in turn, to clock the respective latches for the VALIDKEY and KEY signals – setting the VALIDKEY signal High, to flag to the microcontroller that a valid key has been pressed on the Keypad and presenting the 4-bit data which specifies the pressed key, on the KEY bus.

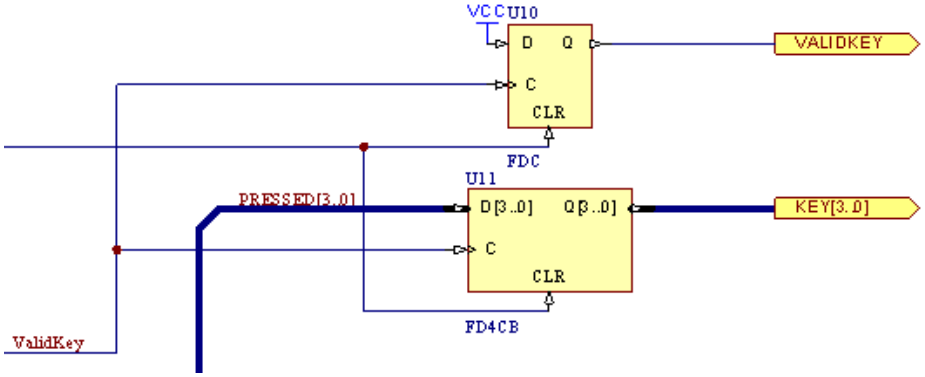

Figure 8. The output from the debounce circuit clocks the ValidKey and KEY latches.

Composing the Pressed Key Data Value

The return value from the Keypad Matrix is, simultaneous to the debounce testing, inverted and passed through a 4-bit to 2-bit priority Encoder (Figure 9).

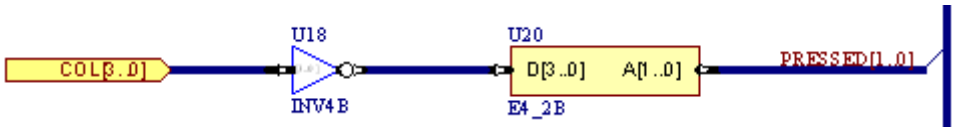

Figure 9. Encoding of the 4-bit COL value returned by the Keypad Matrix.

If the debounce circuitry determines the key press to be valid, the current 2-bit output value from the Encoder will be used to reflect the column in which the pressed key resides, when composing the 4-bit Row-Column data value presented to the microcontroller on the KEY bus.

Table 3 summarizes the 2-bit Column portion of the KEY signal, based on the value returned on the COL bus.

Table 3. Encoded column portion of the KEY signal.

Value on COL bus	Column of Keypad Matrix	Encoded value used for column portion of KEY signal
1110	0	00
1101	1	01
1011	2	10
0111	3	11

When the key-press has been recognized as valid, the current row being tested is the one used to reflect the row in which the pressed key resides. As the clock used for the debounce circuit is eight times faster than the clock used to provide scanning of the rows, there is no possibility of the next row being tested before the debounce validation has completed.

The 2-bit Row portion of the KEY signal is therefore the current output of the Binary Counter.

The separate 2-bit Row and Column values are then combined to form the single 4-bit KEY value, with bits 3..2 representing the Row and bits 1..0 representing the Column. This is the value that the microcontroller will read after being alerted to a valid key press by VALIDKEY going High. This gives the microcontroller the Row-Column intersect within the matrix, for the key that was pressed. The microcontroller can then take the appropriate action based upon its embedded code.

Table 4. Pressed key encoding.

Row\Column	00	01	10	11
00	1	2	3	C
01	4	5	6	D
10	7	8	9	E
11	A	0	B	F

The VALIDKEY signal will remain High and the current data present at the KEY port, until an active reset signal is issued by the microcontroller – appearing at the Keypad Controller's RST input.

Note: If more than one key on the keypad is pressed at the same time, nothing will happen. Both key presses will be ignored and the VALIDKEY signal will not go high.

Revision History

Date	Version No.	Revision
12-Feb-2009	1.0	Initial document release
30-Aug-2011	-	Updated template.