Using MAX II CPLDs as Analog Keyboard Encoders

July 2006, Version 1.0 Application Note 426

Introduction

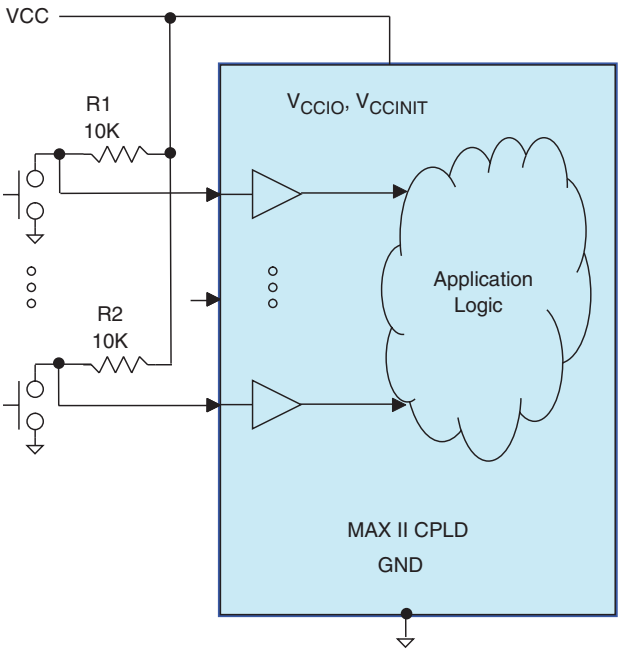
Keyboard encoders are a very common application for CPLDs. Typically a processor, ASSP, or ASIC does not have enough pins for keypads or keyboards. I/O expansion is a very common function for CPLDs and allows a processor to decode a very large keyboard with very few I/Os. Even though CPLDs like MAX® and MAX® II may have abundant low-cost I/Os, decoding a keypad or keyboard with one I/O per switch is not desirable. The advantage of decoding a keyboard with fewer wires is that it reduces the number of wires going from a keypad to a main circuit board or it reduces the complexity of a switch matrix in the keyboard assembly. This application note explains how the resources of a MAX II device can be used to decode a very large number of switches in a keypad or keyboard with only two I/Os and a `GND` pin. The decoding approach used works for as few as 4 switches and as many as 48 switches.

Keyboard Decode Methods

The most common way to encode a switch with a CPLD is by using a simple circuit that ties one end of the switch to `GND` and the other terminal to `VCC` through a pull-up resistor, typically 10 KΩ The `switch` and `resistor` node are connected to any CPLD input pin. For a pushbutton normally-open Single Pole, Single Throw (SPST) switch, the CPLD input is normally a logic one and is a logic zero when the switch is pressed.

Figure 1 shows the circuit for a simple one CPLD I/O per switch hookup. You can enhance this circuit by using a Schmitt trigger input buffer available in the MAX II device. The Schmitt trigger input reduces switching noise and makes switch debounce easier. Refer to *AN 422: Power Management in Portable Systems Using MAX II CPLDs* to learn how to integrate a power-up and power-down mechanism into the switch decode circuit.

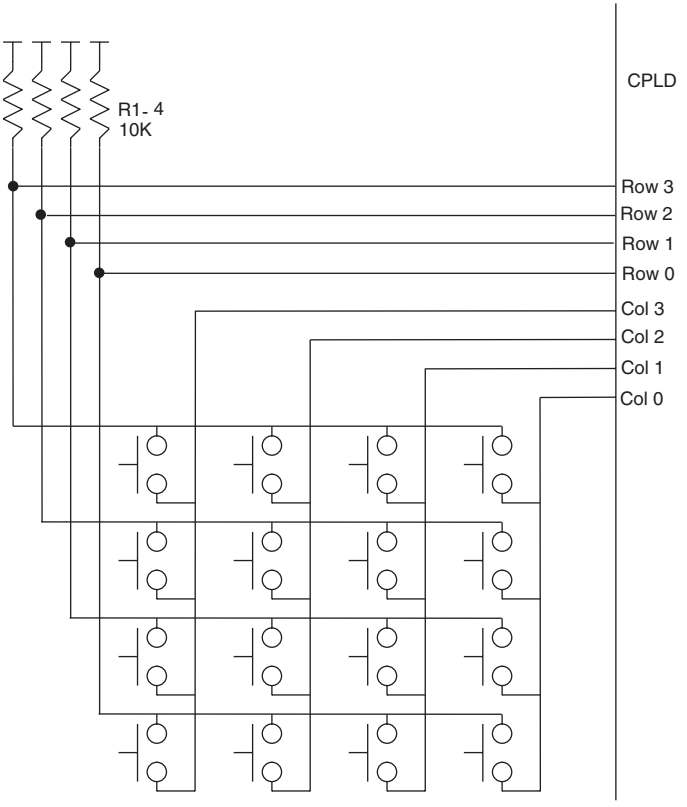
Figure 1. Simple One CPLD I/O per Switch Circuit



A common way to reduce the number of CPLD pins needed to decode a large switch keyboard is to connect the switches in a row-and-column matrix. With this approach, an array of $M \times N$ switches is decoded with only $M + N$ CPLD pins. An example is a 16-switch calculator keypad that is made into a 4x4 array and requires only 8 pins. This is half the number of pins required compared to the simple one-pin-per-switch approach.

Figure 2 shows the typical hook-up for decoding a switch array arranged in rows and columns. Each switch is connected to a respective row and column. The rows also have 10 KΩ pull-up resistors. A simple circuit is added to the programmable logic of the MAX II device.

Figure 2. Common Circuit for Decoding Keypad Using Rows & Columns



When the keypad is in an idle state, the column pins are driven low by the CPLD. The row pins are inputs and waiting for any switch to be pressed. When a switch is pressed and any row signal goes low, the logic in the CPLD begins to scan the switch array to determine which switch or switches are pressed.

The CPLD logic circuit drives all the columns to a high-Z state and then, one at a time, drives each column low. When each column is low, the respective row input pin indicates if the switch in the active column is pressed. The results of each column are loaded into a 4-bit register. Once

all four columns are scanned, a 16-bit register holds the logic value of all the switches. This approach is able to detect single-switch and multiple-switch combinations. When the register becomes all zeros, the scan logic may go back to the idle state, driving the columns low to save power.

The minimum ON pulse for a typical switch is > 3 ms. The typical human minimum response time pressing and releasing a switch is approximately 30 ms. The CPLD can easily scan the switch array in < 10 µS. Therefore, it should not be a problem for the CPLD to decode a switch before a user releases a switch.

Refer to *AN 422 Power Management in Portable Systems Using MAX II CPLDs* to learn how to integrate a power-up and power-down mechanism into the switch array decode circuit.

New Two-Wire Keypad Decider

The pin-per-switch and row-and-column switch array approaches work well, but they are not suitable for every application. They have some disadvantages. The pin-per-switch approach has the following drawbacks:

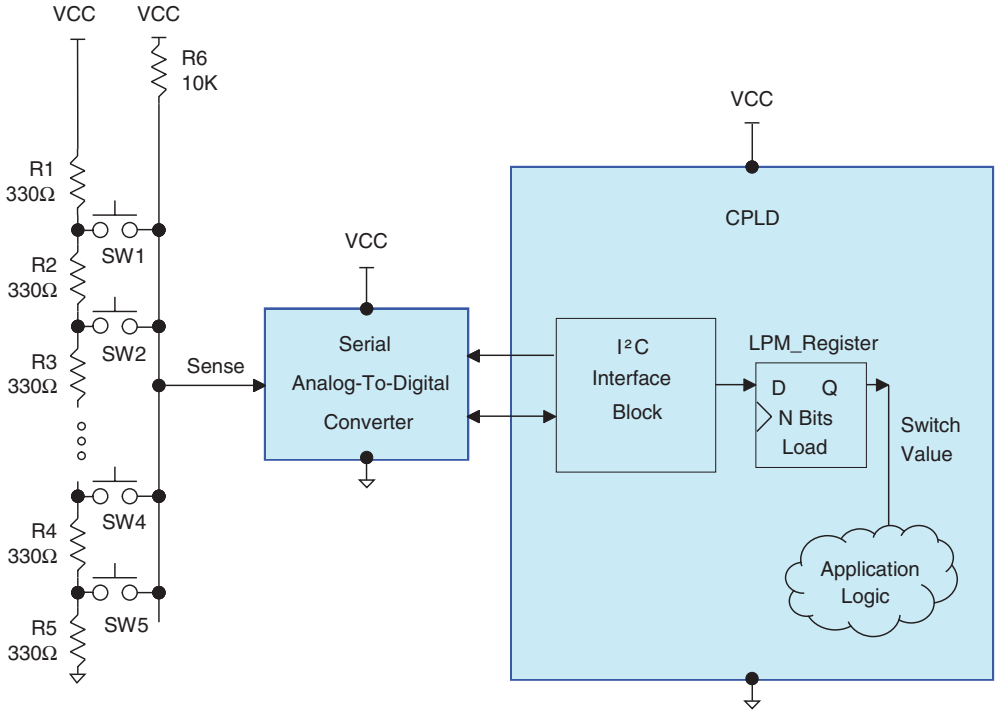
- Requires many CPLD I/O pins for a large keyboard
- Requires many parallel PCB traces to connect switches to the CPLD
- Requires costly wide high-pin count connectors between the membrane switch and the motherboard
- Requires discrete resistors per switch

The row-and-column switch array approach can significantly reduce the number of CPLD pins required and can reduce the number of wires required to connect a membrane switch to a mother board. However, the row-and-column switch array still has some disadvantages:

- Requires two-layer PCB breakout due to orthogonal row-and-column traces to connect switches
- Requires more costly two-layer membrane switch
- Receives only a small reduction in CPLD I/O pins for smaller keyboard arrays

You can use an analog-to-digital converter to decode a switch array. When a set of resistors connects in series between VCC and GND and a switch is connected to each resistor tap and a common node, if a switch is turned activated, the circuit generates a voltage proportional to the switch location in the resistor stack. You can use the output of an analog-to-digital converter to decode the pressed switch. The circuit for this type of decode is shown in Figure 3.

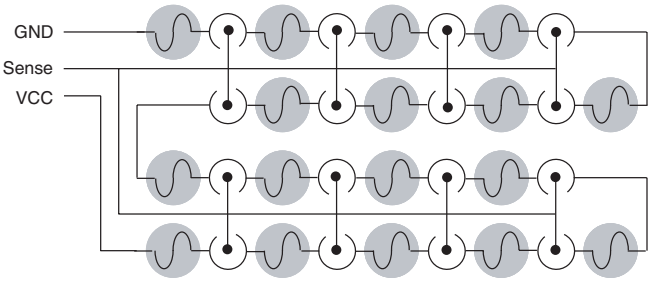
Figure 3. Decoding a Switch Array with an Analog-to-Digital Converter



In Figure 3, a resistor stack goes between VCC and GND. When a switch is closed, it connects the respective resistor tap to the serial analog-to-digital converter. You can program a CPLD to read the voltage sensed from the switch array. The CPLD drives the analog-to-digital converter with the right commands and protocol to capture the data and place it in a parallel register. This approach makes for a very simple switch array that only requires three wires: VCC, GND, and Sense.

As shown in Figure 4, it is very easy to integrate the series resistor into a single-layer membrane switch. You can make the traces in the membrane switch PCB resistive in many ways. The trace width can be made very narrow, a more resistive conductor can be used in resistor segments, and the trace can be etched thinner to increase resistance.

Figure 4. Membrane Switch PCB Layout for Analog Decode



This approach makes the membrane switch or a PCB switch layout very simple and low cost, but the added cost of an analog-to-digital converter may be unacceptable. It is possible to eliminate the analog-to-digital converter and use the MAX II device resources to decode the analog switch.

Two-Wire Keypad Decode Using MAX II Devices

A MAX II device uses its internal oscillator, Schmitt trigger I/O, and high-density arithmetic programmable logic fabric to replace an analog-to-digital converter to decode an analog switch array. The MAX II device only needs a simple, low cost, external capacitor to complete this function. Self-calibration eliminates the need for a high-quality capacitor. The circuit uses the same series of resistors structure as the analog-to-digital converter decode.

Figure 5 has a series of resistors R1 to R5. The resistors R2-R5 are 33 Ω each. There could be many resistors between R3 and R4. Resistor R1 is a multiple of 33 Ω R1 is the minimum resistance when sampling switches. R5 connects to ground.

Resistor R1 connects to the `PreCharge` node of the MAX II device. `PreCharge` is also connected to capacitor C1. The other end of capacitor C1 goes to ground. Each node of resistor stack R1-R5 connects to a normally open switch to a common node, `Sense`. `Sense` has a pull-up resistor R1 to `VCC` and connects to the MAX II device.

The values of R2 to R5 are set such that the total resistance $R2 + R3 + ... + R4 + R5$ is shorted to `Sense` through switch 1. The total resistance has to be low enough that the `Sense` voltage is less than the input low threshold of the `Sense` input of the MAX II device, as shown in Equation 1.

(1)
$$V_{\text{Sense}} = \frac{(R2 + R3 + \ldots + R4 + R5)}{(R2 + R3 + \ldots + R4 + R5 + R6)} < V_{\text{IL}}$$

The minimum resistance of R1 must be large enough to allow for the drive of the MAX II `PreCharge` output, as shown in Equation 2.

(2)
$$\frac{(VCC - V_{\text{OH}})}{R1 < I_{OH}}$$

In addition, the `Sense` drive must be strong enough when switch 1 is shorted, as shown in Equation 3.

(3)
$$\frac{(VCC - V_{OL})}{R1 < I_{OL}}$$

Resistor stack R1-R5 and capacitor C1 forms an RC time constant when `Sense` is tri-stated and no switch is pressed, as shown in Equation 4.

(4)
$$\tau = (R1 + R2 + R3 + ..R4 + R5) \times C1$$

When a switch is pressed and `Sense` is low, the RC time constant changes. If switch SW2 is pressed, then the RC time constant becomes faster, as shown in Equation 5.

(5)
$$\tau = (R1 + R2) \times C1$$

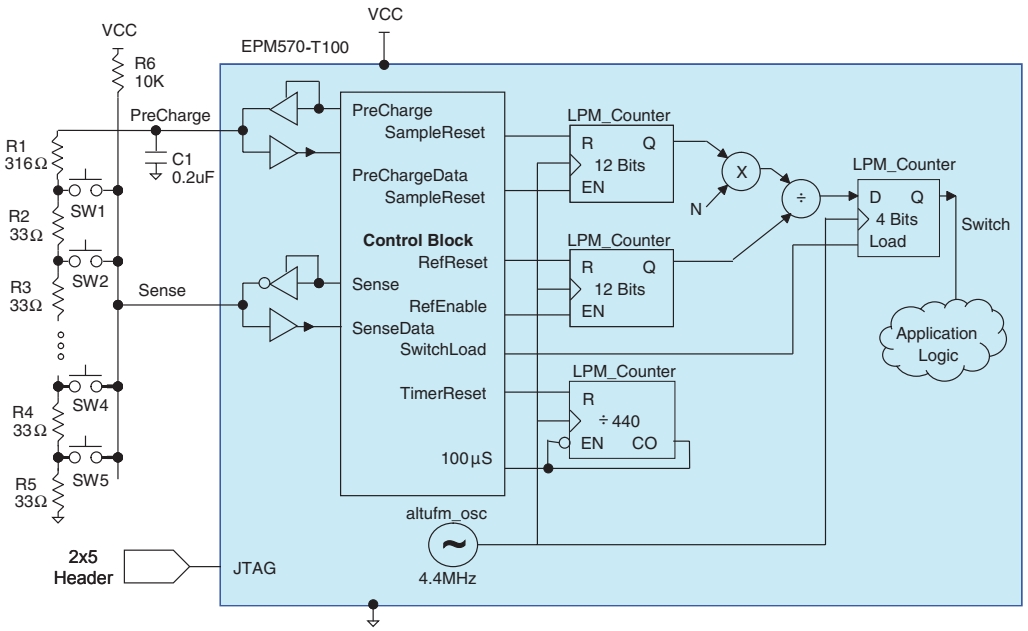
If switch SW4 is pressed, then the RC time constant becomes (Equation 6):

(6)
$$\tau = (R2 + R3 + \ldots + R4 + R5) \times C1$$

To determine the switch presses based on the time constant, use the internal oscillator of the MAX II device. Because the MAX II internal oscillator is only approximately 25% accurate but is consistent, the measurement of the RC time constant must be done in two parts.

First, you must measure a reference. This is the RC time constant when the PreCharge is pulsed **High** and the Sense signal is tri-stated. Then take a sample when the Sense is held low and the PreCharge is pulsed **High**. Based on the difference between the two time constants and the number of resistor unit values between PreCharge and GND, you can reliably decode the switch pressed.

Figure 5. Two Wire Keypad Decode Using MAX II Devices



The circuit in the MAX II device consists of a control block. There are three counters. A 12-bit counter is used to measure the *reference* τ and the switch τ; a 100 µS timer is used for debouncing. The arithmetic capabilities of the MAX II device are used to multiply the Sample count by *N*, where *N* is the number of total resistor units R5 in the resistor stack R1 to R5. R1 will usually be a multiple of R5. Then the *(<Sample> τ)*N* is divided by the *(<reference> τ)*. The resulting binary value corresponds to the switch pushed is stored in a holding register.

The equation to determine the switch value is (Equation 7):

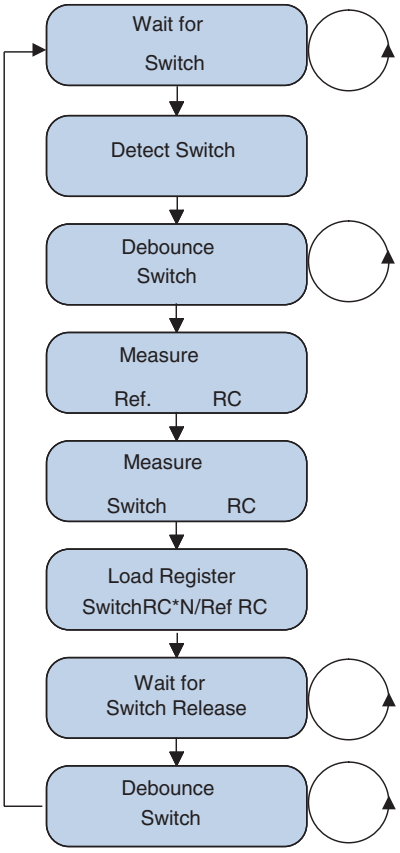
(7) $$Switch = \frac{Sample \times N}{Reference}$$

- *Sample* = Count of the counter measuring the discharge time when the `Sense` pin is driven low
- *Reference* = Count of the counter measuring the discharge time when the `Sense` pin is tri-state
- *N* = (Number of resistance units) R5 in the total resistance stack R1 to R5.

Use the capacitor value to adjust the typical discharge time. If the time is too long, the `Sense` and `Reference` counters can be increased from 12 bits. If the time is shortened, the `Sense` and `Reference` counters can be made with fewer bits. The minimum recommended counter size for the full `reference` stack is *N*4* or $Log_2(N*4)$ bits, (rounded up).

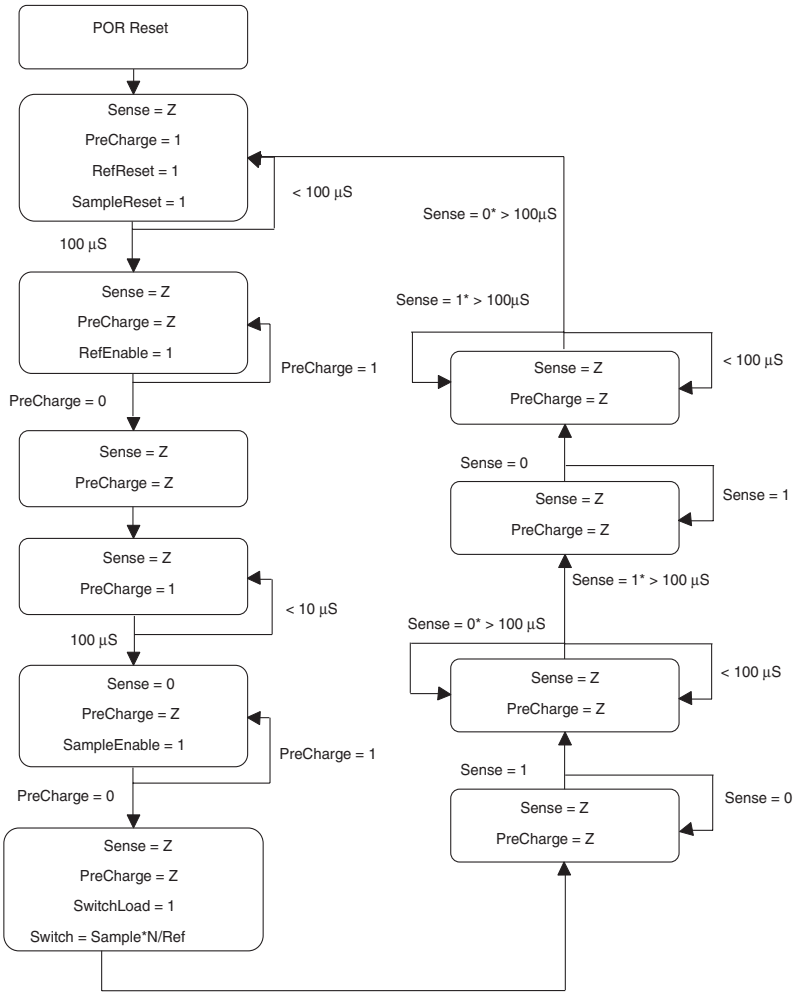
The flow chart in Figure 6 illustrates at a high level the switch decode process at a high level. The loops to the right of some states represent the device looping or waiting for a change in the switch state, or in the case of the debounce block, waiting for a fixed number of clock cycles for the switch to settle after a transition.

Figure 6. Flow Chart of a Two-Wire Switch Decode Process



The flow chart in Figure 7 describes the operation of the control block from Figure 5 in detail. The control block controls the timing of the sample measurement, the reference measurement, and the debounce of the switches. This diagram only specifies active signal outputs. All signals not specified are in a low or inactive state.

Figure 7. Control Block State Machine for a Two-Pin Keypad Decode(1)



Note to Figure 7:

(1) Outputs: PreCharge, Sense, RefReset, RefEnable, SampleReset, SampleEnable, SwitchLoad, TimerReset, Inputs, PreCharge, Sense, 100 µS.

You can improve the accuracy of the circuit if R1 is not an exact multiple of R5. R1 should be $N + 0.5 \times R5$. Making R1 an exact multiple of R5, the value coming from the multiply-divide function, is only one least significant bit (LSB) away from the next lower switch value. This means the division remainder is always 1 or 0. Therefore, a small amount of noise could make the circuit misread a switch if the count of the `Sense` or `Reference` counter is off by only one bit. By making the R1 value an *N*0.5* multiple, the division remainder will typically be $0.5 \times (\text{Refence} / N)$. Therefore, the `Sense` or `Reference` counters would need a lot of noise to be off by $0.5 \times (\text{Reference} / N)$ extra or fewer counts.

You can also improve the accuracy of the circuit by using a Schmitt trigger input for the `PreCharge` input buffer. The Schmitt trigger is less sensitive to noise on the relatively slow `PreCharge` node discharge.

Figure 8 shows how the keypad decode circuit of Figures 5, 6, and 7 can be combined with the power-up detect circuit described in *AN 422 Power Management in Portable Systems Using MAX II CPLDS*. This combined circuit has the advantage of a wire-efficient way to connect a multiple switch panel and a power-up detect feature that works for all switches with only one diode.

The switch matrix connection shown in Figures 1 and 2 requires one diode per switch or one diode per row. The only requirement for the new circuit to work is that the voltage divider created when a switch is pressed between R7, D1, SW1, and R2-5 provide a voltage to the gate of Q1 that is low enough to turn it on, as shown in Equation 8.

(8)
$$V_{GS} = \left(-1 \times \frac{(VCC - V_F)}{(VCC)} \times \frac{R7}{(R2 + R3 + \ldots + R4 + R5 + R7)} < \sim 0.7V\right)$$

- V_{GS} = The gate-to-source voltage of Q1
- V_F = The forward voltage drop of D1
- VCC = The battery supply voltage that is typically 3.0 V, but can be as low as 2.2 V, before the batteries are considered dead

Once the device is powered up, the `PWRDWN` node being low connected to the `Sense` node through the reverse biased diode D1 does not effect the operation of the switch decode process. Also, the switch decode process even in the slowest oscillator frequency process voltage temperature (PVT) is completed in < 1 ms. Therefore, the power-up time and the switch decode is easily completed in the 3 ms minimum switch on-time.

Figure 8. Two-Wire Keypad Decode with Power-Up & Power-Down Control

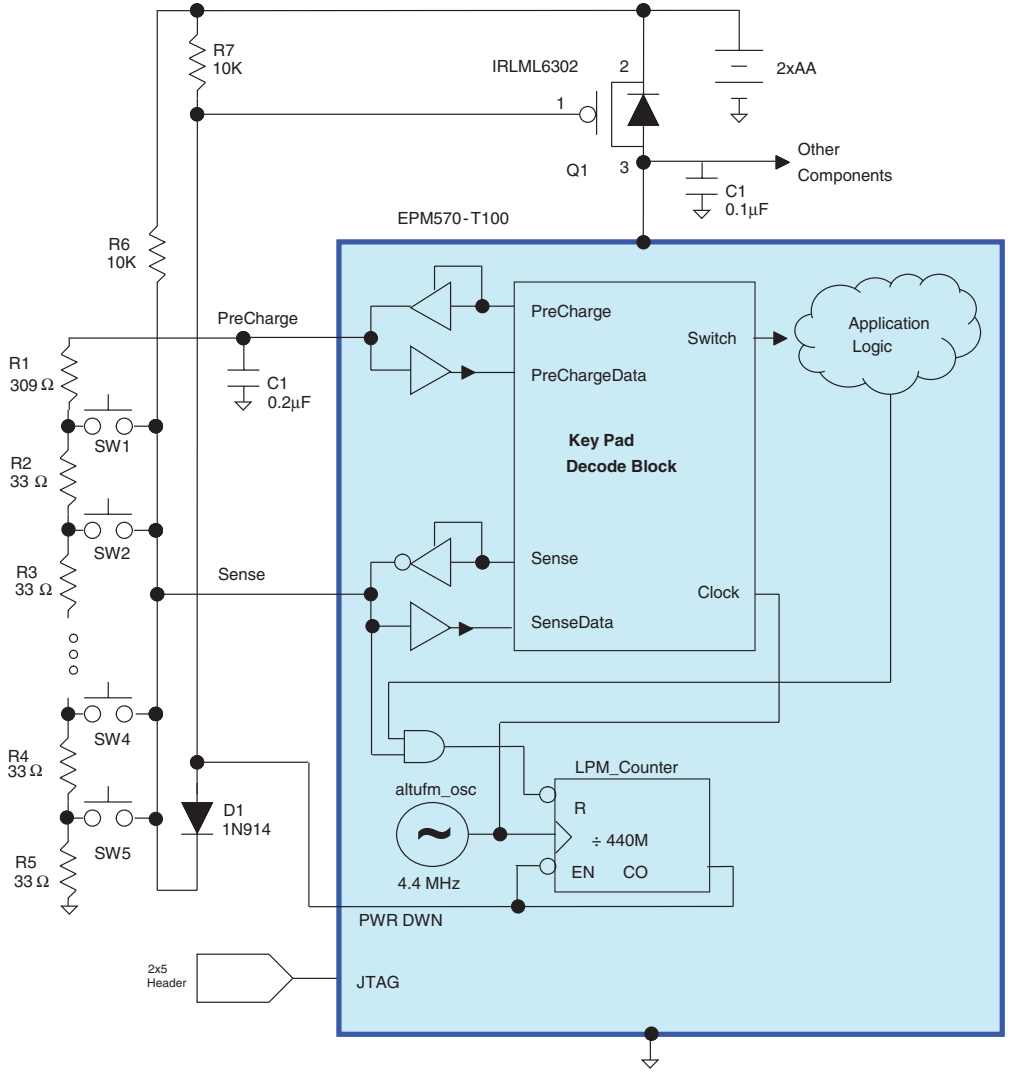
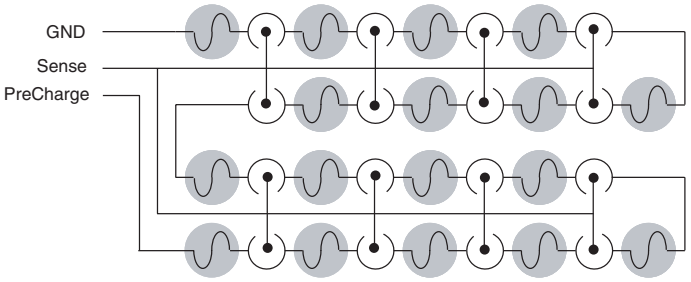


Figure 9 illustrates how the series resistance circuit required for decoding can be done very cost effectively using a membrane switch. The figure shows the layout of the traces below the membrane switches. The incomplete circle with a dot in the middle is the basic switch structure. The traces in the shaded area are high resistance traces that act as series resistors. The high resistance is achieved by using different material then connection traces or by using minimum line width.

Figure 9. Two-Wire Keypad Membrane Switch with Integrated Resistors



Advantages

An important advantage to using the MAX II device is that you have many choices when you need to decode a switch. A switch-per-pin may be the best choice for some applications while the two-pin-analog decode approach may be the most cost effective for some applications. The internal oscillator of the MAX II device is a very cost-effective way to debounce switches and enable complex decode operations without requiring an external oscillator. The Schmitt trigger feature of the MAX II device is very important in decoding switches using any method because, as in all methods, switch signals are very slow and noisy. The Schmitt trigger makes debounce easier and the analog switch decode more accurate. MAX II's density and arithmetic fabric makes the complex calculations required for the analog switch debounce easy and area efficient. The analog decode method is not possible with a traditional CPLD logic fabric.

Conclusion

MAX II CPLDs are the best choice for switch decode. With power-down techniques possible in all switch decode modes, a MAX II device is also the best choice for battery-powered portable applications.

101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
www.altera.com
Applications Hotline:
(800) 800-EPLD
Literature Services:
literature@altera.com