AN58827

PSoC® 3 and PSoC 5LP Internal Analog Routing Considerations

Author: Mark Hastings

Associated Project: No

Associated Part Family: All PSoC 3 and PSoC 5LP parts

Software Version: PSoC Creator™ 1.0 or Higher

Related Application Notes: None

AN58827 discusses how internal trace and switch resistance can affect the performance of a design and how these issues can be avoided by understanding a few basic details about the PSoC® 3 and PSoC 5LP internal analog architecture. Trace and switch resistance are not a concern for most applications. However, this application note teaches the designer when resistance in the signal path may cause measurement errors.

Introduction

When you add a wire or trace on your PCB, you add some amount of resistance in the signal path. This is also true when adding signal paths and switches in an integrated circuit. The only difference is the scale, but Ohms law still holds true.

The PSoC 3 and PSoC 5LP parts are possibly the most flexible mixed signal controllers on the market today. The internal analog switch matrix provides many options when routing signals between analog blocks and GPIO (General Purpose Input and Output) pins. The signal paths and switches required to provide this flexibility also add resistance between the signal source and its destination. A detailed diagram of the analog blocks, GPIOs, and switch matrix is located in the Appendix at the end of the application note.

Calculating Path Resistance

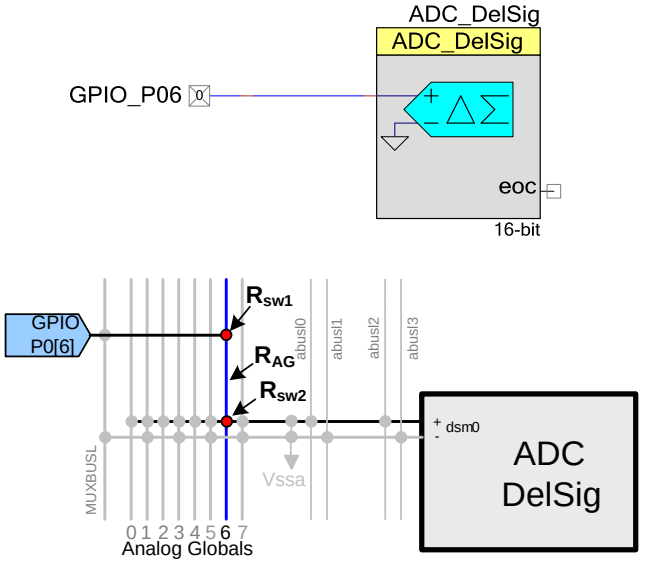
To determine the resistance of a path between an analog block and a GPIO pin, add up the resistance for each switch and trace in the signal path. Table 1 gives an approximation for the worse case resistance of each analog path and switch.

Table 1. Typical Resistance of Internal Paths and Switches

Item	Label	Typical Resistance
Small Switch	(colored white)	500 - 700 Ω
Large Switch	(colored red)	200 - 350 Ω
XLarge Switch	(colored green)	~ 50 Ω
Analog Global	AGL[7:0], AGR[7:0]	~ 200 Ω
Analog Mux Bus	AMUXBUSR, AMUXBUSL	~ 100 Ω
Analog Local Bus	AbusR[3:0], AbusL[3:0]	~ 100 Ω

For a simple path between a GPIO pin and the ADC, an estimation of the signal path resistance can be calculated. See Figure 1 for a representation of a simple circuit where P0[6] is connected to the positive input of the DelSig ADC.

Figure 1. Simple Signal Route



The upper half of Figure 1 shows a schematic in PSoC Creator™. The lower half of Figure 1 shows one possible path routed by PSoC Creator to connect a GPIO pin to the positive input of the ADC block. If we analyze the signal path, the signal will pass through two switches and an analog global bus. The total resistance of this path may be approximated by the following equation for this example:

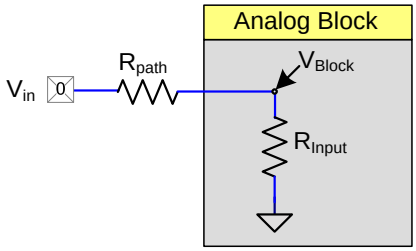
$$R_{Total} = R_{SW1} + R_{AG} + R_{SW2} \qquad \text{Equation 1}$$

$$R_{Total} = 250_{SW1} + 200_{AG} + 250_{SW2} = 700\,\Omega \qquad \text{Equation 2}$$

Substituting the actual resistance values in the equation, you get a signal path resistance of about 700 Ω. The actual resistance may be less, because the resistance of the analog global shown in Table 1 is the worst case. Values between the absolute minimum and maximum were used for this example. Most signals follow only a section of the global, not the entire length, so its resistance will be less.

After calculating an approximate signal path resistance, the designer must ask himself, “Does the 700 Ω affect the design?” The buffered input to the ADC has an input resistance of greater than 100 MΩ. Figure 2 shows how a voltage divider is created for any analog block that has an input resistance less than infinity and more than zero.

Figure 2. Design with Equivalent Resistance



The two resistors R_{path} and R_{Input} form a voltage divider so the actual voltage seen at the input by the ADC (V_{Block}) is not the same as the input voltage, V_{in}. The following equation calculates the error created due to these resistors:

$$\%Error = \frac{R_{path}}{(R_{input} + R_{path})} * 100 \qquad \text{Equation 3}$$

Substituting with the actual resistor values;

$$\%Error = \frac{700}{(100{,}000{,}000 \;\; + 700\;)} * 100 = 0.0007\,\% \qquad \text{Equation 4}$$

Because the input resistance is greater than 100 MΩ and the actual path resistance is less than 700 Ω, the error introduced by the path resistance is less than 0.0007% or 7 ppm, which is insignificant for most applications. This is also true when using the Comparator, Opamp, and PGA where the input resistance is also greater than 100 MΩ.

When does Resistance Matter?

The input resistance for most PSoC 3 and PSoC 5LP blocks is shown in Table 2. When the input resistance is greater than 100 MΩ as in the example described earlier, the error is insignificant. On the other hand, where the input resistance is less than a 100 MΩ, the path resistance may be significant. A couple of examples where this is a concern is the un-buffered mode of the DelSig ADC and the inverting input of the PGA.

Table 2. Typical Input Resistance to Some Analog Blocks

Analog Block	Input Resistance	Affected by Routing Resistance
DelSig ADC (Buffered Input and Internal source)	>100 MΩ	No
DelSig ADC (Buffered Input and GPIO source)	>10 MΩ	No, but the difference here is because of the GPIO pin leakage
DelSig ADC (Un-buffered Input)	>80 KΩ	Input resistance is a function of ADC clock and input capacitor
Op Amp	>100 MΩ	No
PGA	>100 MΩ	No

Table 2. Typical Input Resistance to Some Analog Blocks (continued)

Analog Block	Input Resistance	Affected by Routing Resistance
Inverting PGA	20 KΩ or 40 KΩ	Yes
Mixer	20 KΩ or 40 KΩ	Yes
SAR ADC (PSoC 5LP only)	>150 KΩ	Input resistance is a function of the sample rate
Comparator	>100 MΩ	No

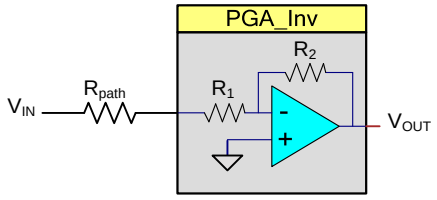
When using the DelSig ADC in the un-buffered mode, the input resistance may be low enough to adversely affect the accuracy of the design. The input resistance of the DelSig ADC is a function of the ADC clock and input capacitance. The input capacitance of the DelSig ADC is between 1 and 16 pF depending on the range and resolution. The actual input capacitance for each range of the DelSig ADC may be found in the ADC's datasheet. The following equation is used to calculate the input resistance of the DelSig ADC.

$$R_{input} = \frac{1}{(C_{input} * F_{clk})} \quad \text{Equation 5}$$

For example, if the DelSig ADC clock is 3.0 MHz, the resolution is 15 bits and the range is set to ±1 V, the input capacitance is about 4 pF. Using this equation, the input resistance is about 83 KΩ. Using Equation 3, a path resistance of about 700 Ω can introduce an error of 0.7%, which may be significant. The user has the option to either use the ADC's input buffer to eliminate this error, or to continue using the unbuffered input and compensate using firmware.

In the case of the Inverting PGA, the input trace resistance affects the gain. The path resistance adds to R_1 in Figure 3.

Figure 3. PGA Schematic with Path Resistance



Normally the gain equation for the Inverting PGA is:

$$Gain = \frac{R_2}{R_1} \quad \text{Equation 5}$$

If you add the input path resistance to the equation, it is easy to see how it affects the gain in Equation 5.

$$Gain = \frac{R_2}{(R_1 + R_{path})} \quad \text{Equation 6}$$

R_1 is either 20 K or 40 KΩ for in the Inverting PGA. If the path resistance is as high as 700 Ω and R_1 is 20 KΩ the gain error is about 3.5% less than expected or about 1.7% if R_1 is 40 KΩ.

Example Project

The example project demonstrates a method to measure the temperature of an external diode, called Delta-V_{BE}. This is a popular method to measure the die temperature of large CPUs and FPGAs. Manufacturers of these parts place a P-N junction on the die and expose the two terminals from this junction to pins on the package. The Delta-V_{BE} method is immune to V_{BE} offsets and does not require temperature calibration. All that is required is an accurate voltmeter and an adjustable current source. The current source only needs to apply two different currents with a known ratio. The absolute value of these currents is not important, just the ratio of the two currents. When each of the currents is applied, the voltage across the P-N junction is measured. The ratio of the currents should be about 10 or more so that the difference in V_{BE} is large enough to make an accurate measurement.

PSoC 3 and PSoC 5LP contain the two components required to perform this measurement, an accurate ADC and an adjustable current source. The following equation is used to calculate the temperature for this method.

$$\Delta V_{BE} = \frac{KT}{(q \bullet \ln(N))} \quad \text{Equation 7}$$

Where:

Δ **V_{BE}** is the difference in junction voltage measured at each current

K is Boltmann's constant (1.380658x10^{-23}) joules/K

q is the charge of an electron (1.602176x10^{-19}) Coulombs

T is absolute temperature in Kelvin

N is the ratio of the two currents

First solve for degrees Kelvin:

$$T_{\circ K} = \frac{\Delta V_{BE}}{\ln(N)} \cdot \frac{q}{K} \quad \text{Equation 8}$$

Next, covert to degrees Centigrade:

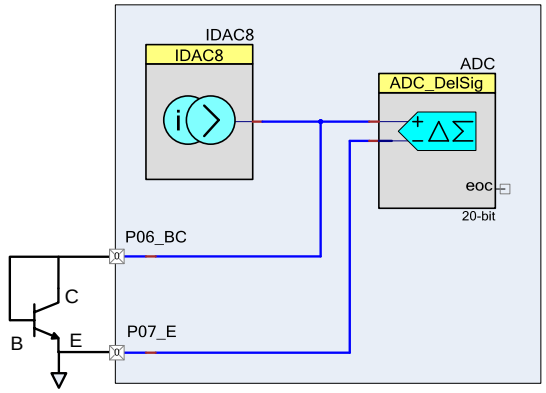
$$T_{\circ C} = \left(\frac{\Delta V_{BE}}{\ln(N)} \cdot 11604\right) - 273 \quad \text{Equation 9}$$

If we fix the current ratio to 10, the equation gets even simpler.

$$T_{\circ C} = (\Delta V_{BE} \cdot 5040) - 273 \quad \text{Equation 10}$$

Now that the method and equation are fixed, all that is left is to implement it with a PSoC. Figure 4 shows the schematic from PSoC Creator with an external 2N3904 NPN transistor, as the temperature sensor.

Figure 4. Temperature Sensor Schematic



Test the Project

The project is implemented, compiled, and downloaded, but the temperature measurement is much higher than expected, over 60 °C. You had expected something closer to 25 °C, room temperature. This is an error of about 35 °C, what could have caused this error?

Because the circuit uses a current source and we know that switches and internal traces are resistive, an I*R drop is the most likely suspect. Adding the actual resistors to the schematic that are inherent from routing generates a more accurate schematic as shown in Figure 5. The input resistance to the ADC is relatively high; therefore, the voltage drop across R2, R4, and R5 are insignificant. There is a voltage drop across R1, but it is out of the measurement path. R3 on the other hand, is directly in the measurement path. The current path between the current source (IDAC8) and the external transistor is directly through R3, which causes a significant I*R drop across R3. When the ADC measures the Base-Emitter voltage (V_{BE}) of the external transistor, it also measures this voltage drop across R3.

Figure 5. Schematic Showing Routing Resistance

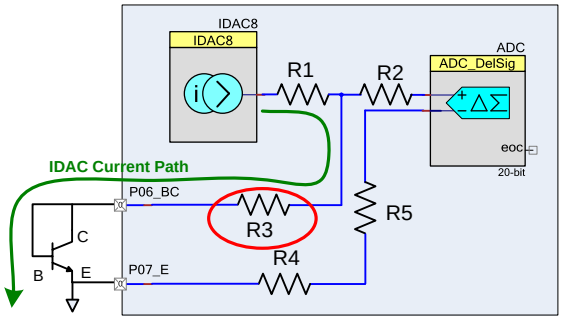
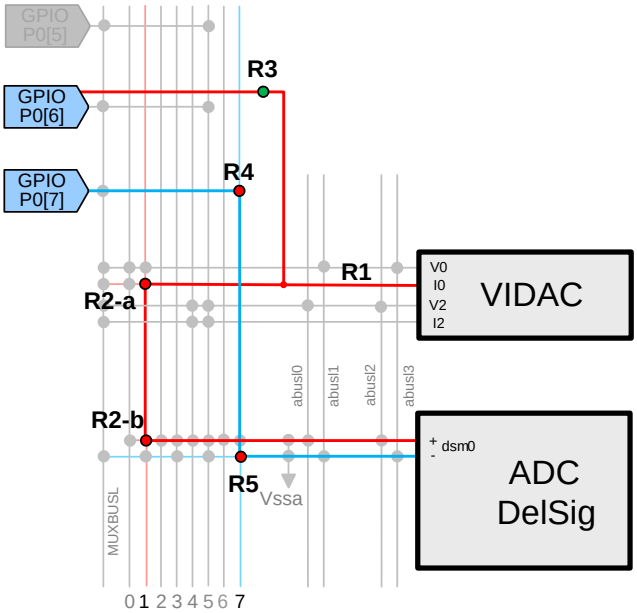


Figure 6 shows the actual internal routing for this project and correlates the schematic resistors with the actual switches.

Figure 6. Actual Routing of Project



With a current ratio of 10, the volts per degree Centigrade is approximately 200 µV/C. The temperature error is the measured temperature minus the actual temperature. In this case, it is about 35 °C (60 $°C_{measured}$ – 25 $°C_{actual}$). So, the voltage error is 35 °C * 200 µV/°C or about 7 mV. This voltage error is the I*R drop across R3.

The resistance of the switches and the path is estimated by dividing the error voltage by the difference in current. Remember, this method switches between 10 µA and 100 µA so the current delta is 90 µA. The resistance is the voltage drop divided by the current delta, 7 mV/90 µA = 78 Ω. This can easily be explained by the resistance of a switch (XLarge) and some internal routing resistance defined in Table 1 on page 1, which confirms that the error is caused by the routing resistance.

Fixing the Problem

Now that the mystery is solved, how do you fix the problem? One way is to route the current source out to its own pin. This eliminates the switch resistance that was shared by both the current source and measurement path. The ADC now measures only V_{BE} and not the extra voltages caused by an I*R drop. See Figure 7 for the new schematic. When the project is rebuilt, compiled, and run, the calculated temperature is within a couple degrees of the actual temperature expected. This small change in the circuit has a big impact on the accuracy of the design, but does cost an extra GPIO pin. This design is a corner case, but it emphasizes how a designer should always be aware of trace resistance both inside and outside the device. Review the internal routing in Figure 8.

Figure 7. Schematic with Separate Current Path

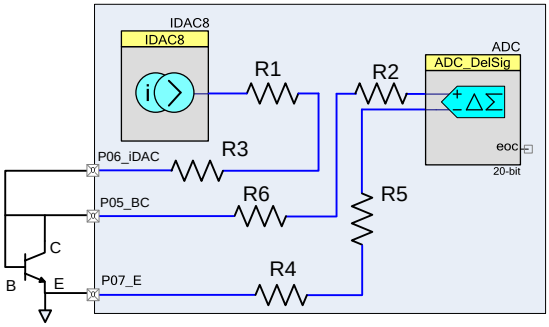
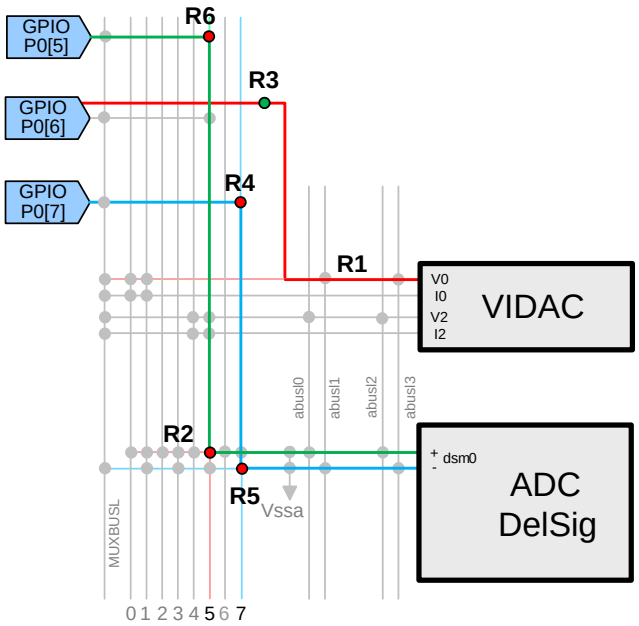


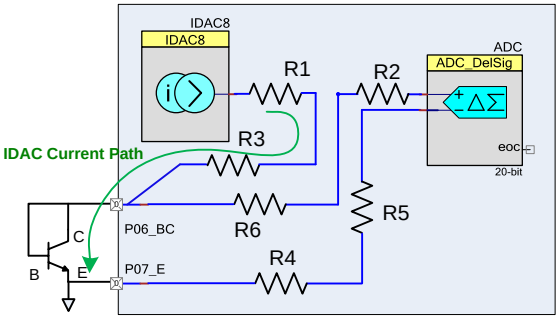
Figure 8. Actual Routing of Solution



Alternative Solution

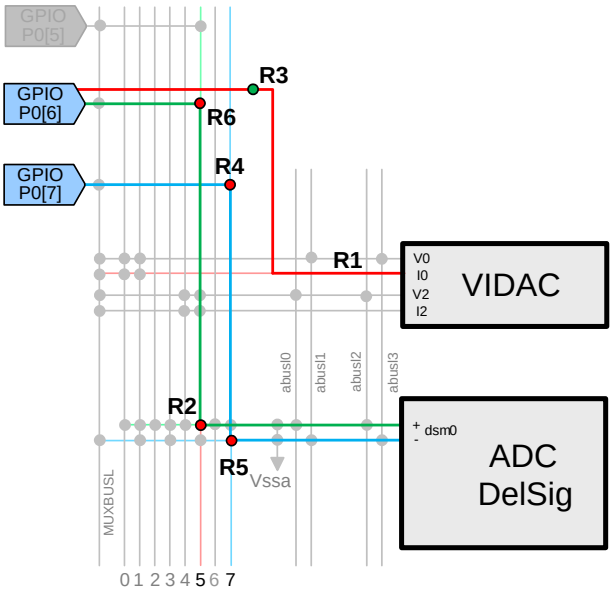
With PSoC there is almost always an alternative solution to every design. In the previous solution the temperature error was eliminated, but at the cost of an additional GPIO pin. This may not always be an alternative if your design is tight on pins. Perhaps a better solution is to use the fact that there is both the standard analog global connection as well as the dedicated high current connection to GPIO P0[6]. This way, the path from the IDAC8 and the path from the ADC can be separate until the actual connection to the pad at GPIO P0[6], and not require a separate pin. See the routing schematic at Figure 9.

Figure 9. Alternative Solution Schematic



If you evaluate the internal routing in Figure 10, you can see that the current and measurement paths are truly independent up to P0[6]. If the external diode sensor is near the PSoC, or the PCB trace resistance is low, this is a good solution. If the sensor is connected via a resistive path from the PSoC, the initial solution may be a better alternative, since sharing the current and measurement path is what originally caused the problem.

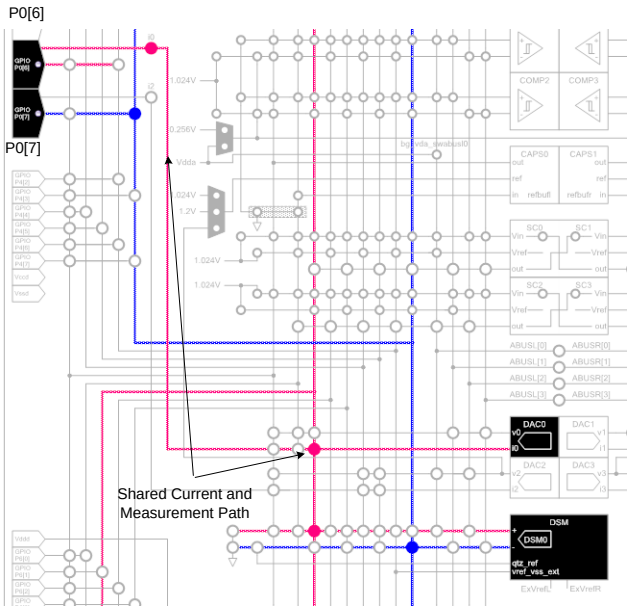
Figure 10. Routing of Alternative Solution



The schematic of the alternative solution and the original have the same schematic Figure 4, so how do you force this different route? As of PSoC Creator 2.0, an additional tool was added to allow the designer to review and change the route taken by any path. This tool is called the “Analog Device Editor”. It allows the designer to alter a signal path, and to select alternate analog blocks. The route and block can then be locked so that the route will be static even if you rebuild the project.

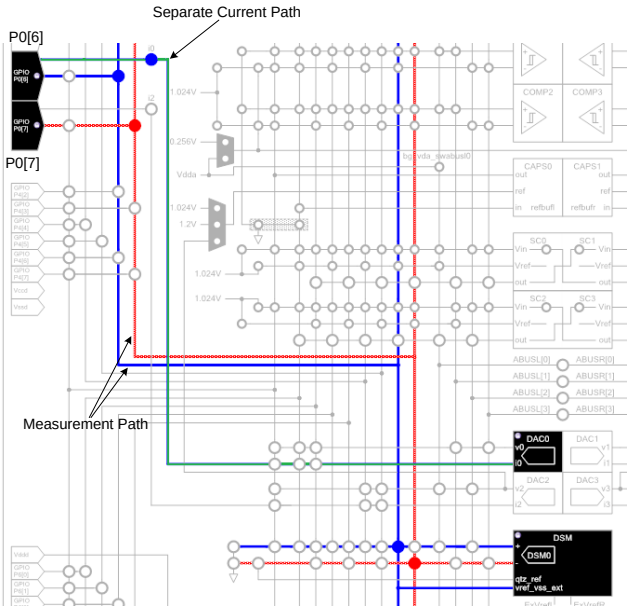
Figure 11 shows a partial view from the Analog Device Editor of the original temperature sensor design. Notice that the template is very similar to the Analog Interconnect Diagram provide in the appendix, Figure 15.

Figure 11. Analog Device Editor View of Original Project



By rerouting the design, you can see the difference in the actual route. The voltage measurement point becomes the actual GPIO, P0[6]. No current from the IDAC8 flows in the measurement path from the ADC. Although these two solutions are the exact same schematic, they are routed differently. The Analog Device Editor allows you to make minor changes in just a few minutes. In the case of this example project, it reduced the temperature measurement error from totally unacceptable to down to a degree or two.

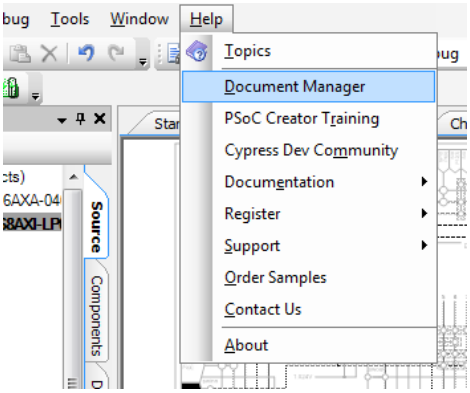
Figure 12. Analog Device Editor view of Alternative Solution



Analog Device Editor Documentation

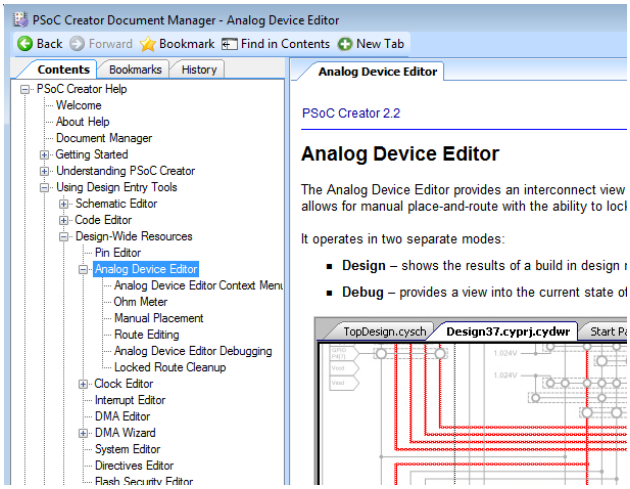
A complete description of the Analog Device Editor and how to use it can be found in the PSoC Creator Document Manager. To find this document, click on the “Help” menu and select “Document Manager”. See Figure 13.

Figure 13. Document Manager Location



In the Document Manager, select “PSoC Creator Help” under the “Contents” tab and navigate down to *Using Design Entry Tools/Design-Wide Resources/Analog Device Editor*. This section of the help manual will show you how to examine and to edit your circuit’s analog. See Figure 14.

Figure 14. Analog Device Editor Documentation



Some Analog Device Editor features include:

- View actual signal paths
- Examine Analog Mux routing and configuration
- Measure the resistance of a signal path
- Lock components to a specific analog block
- Change which analog blocks are used for a given component
- Re-route signal paths and analog mux routes
- View individual switch resistance
- Display individual switch control register address and mask values

Summary

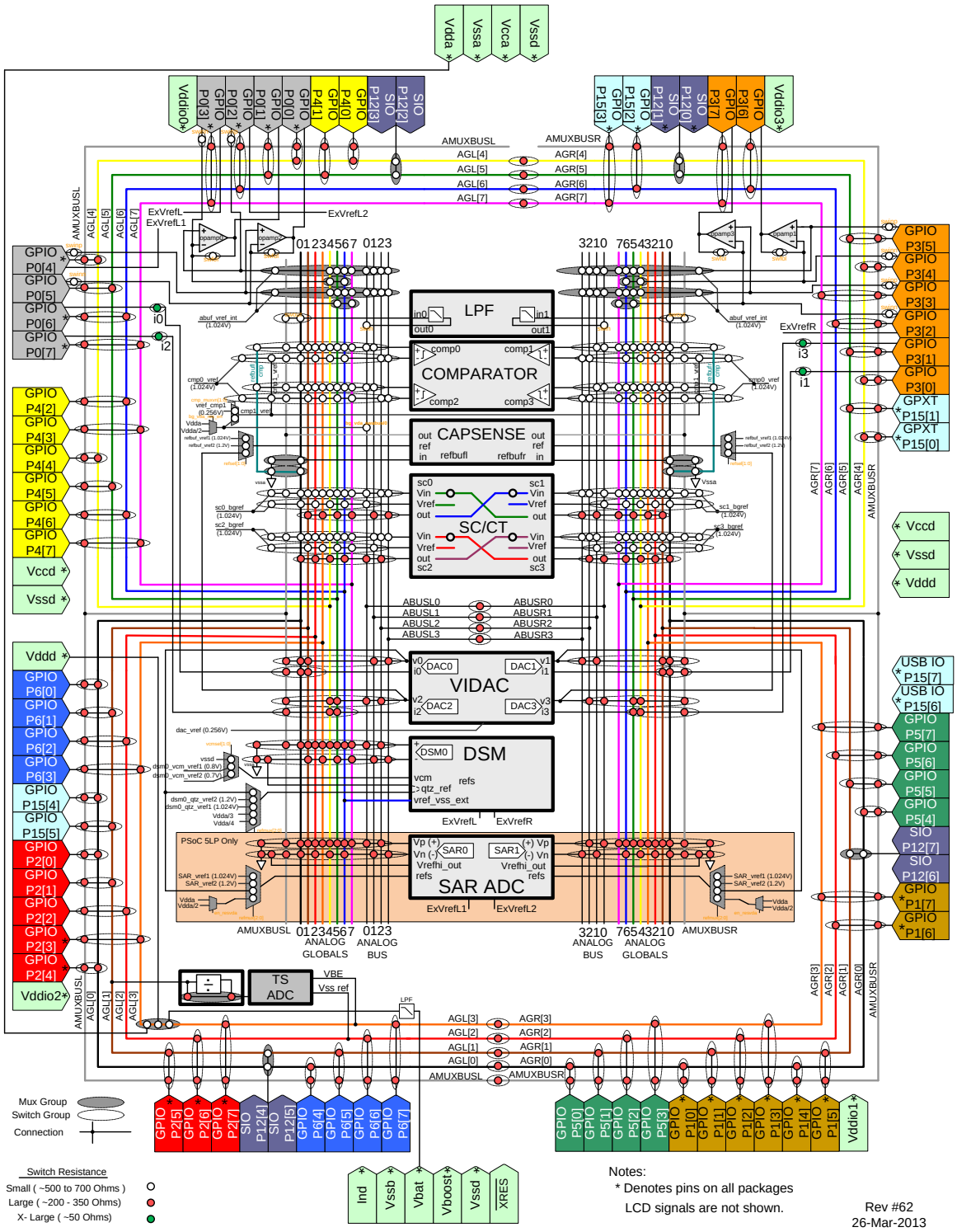
This application note helps you to think about what is going on inside the PSoC. Usually, the current through the switches and internal traces is so low, that you can completely ignore voltage drops that may occur. When dealing with current sources or inputs that do not have high input resistance, pay extra attention to the signal path and any voltage drops that may occur. Also, remember that Ohms Law is the same whether the circuit is implemented inside the PSoC or the old way, using discrete components on a PCB. Make use of the Analog Device Editor to review how you design has been routed. If you do find an issue, this powerful tool can help you force a more desirable route to achieve optimal performance.

About the Author

Name:	Mark Hastings
Title:	Applications Engineer MTS
Background:	Mark Hastings received his BSEE degree from Washington State University in 1984. For most of the last 29 years he has been involved in embedded and mixed signal designs. In his free time he can be found hiking the North Cascades of Washington State.
Contact:	meh@cypress.com

Appendix

Figure 15. Analog Interconnect Diagram



Document History

Document Title: AN58827 – PSoC® 3 and PSoC 5LP Internal Analog Routing Considerations

Document Number: 001-58827

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	2859800	MEH	01/20/2010	New application note
*A	2991568	SRIH	07/22/2010	Fixed branding discrepancies
*B	3132534	MEH	12/08/2010	Changed title to “Internal Routing Considerations for PSoC® 3 and PSoC 5 Analog Designs”. Updated description of Figure 1. Updated Analog Interconnect Diagram and added caption to diagram.
*C	3506342	MEH	01/23/2012	Updated template according to current Cypress standards. Changed the title. Updated Figure 15. Several minor updates.
*D	3811873	MEH	11/15/2012	Updated Associated Part Family as “All PSoC 3 and PSoC 5LP parts”. Replaced PSoC 5 with PSoC 5LP in all instances across the document.
*E	3956041	MEH	4/05/2013	Add alternative routing solution and introduce the reader to the PSoC Creator analog editor. Updated the Analog Interconnect Diagram.
*F	5688104	AESATMP8	04/19/2017	Updated logo and Copyright.

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

ARM® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6

Cypress Developer Community

Forums | WICED IOT Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

© Cypress Semiconductor Corporation, 2010-2017. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.