PSoC® 1 Programmable Bipolar Analog Current Source

Author: Dave Van Ess, Rajiv Vasanth Badiger

Associated Part Family: CY8C24x23, CY8C24x94, CY8C27x43, CY8C28xxx, CY8C29x66

Related Application Notes: AN2041

To get the latest version of this application note, or the associated project file, please visit http://www.cypress.com/go/AN36179.

The unique configuration of PSoC® 1 switched capacitor blocks enables construction of a programmable bipolar current source. This application note explains how to build a programmable current source using a couple of basic PSoC User Modules.

Contents

1 Introduction

Many applications need an accurate and controllable current source. The unique PSoC MCU architecture enables programmable current sources that source and sink 40 mA. This application note provides:

- A brief explanation of instrumentation amplifier based (In-Amp) current sources.
- Examples of analog current sources implemented with PSoC analog blocks.

2 PSoC Resources

Cypress provides a wealth of data at www.cypress.com to help you to select the right PSoC device for your design, and quickly and effectively integrate the device into your design. In this document, PSoC refers to the PSoC 1 family of devices. To learn more about PSoC 1, refer to the application note AN75320 - *Getting Started with PSoC 1*.

The following is an abbreviated list for PSoC 1:

- **Overview:** PSoC Portfolio, PSoC Roadmap
- **Product Selectors:** PSoC 1, PSoC 3, PSoC 4, or PSoC 5LP. In addition, PSoC Designer includes a device selection tool.
- **Datasheets:** Describe and provide electrical specifications for the PSoC 1 device family.
- **Application Notes and Code Examples:** Cover a broad range of topics, from basic to advanced level. Many of the application notes include code examples.
- **Technical Reference Manuals (TRM):** Provide detailed descriptions of the internal architecture of the PSoC 1 devices.
- **Development Kits:**
  - CY3215A-DK In-Circuit Emulation Lite Development Kit includes an in-circuit emulator (ICE). While the ICE-Cube is primarily used to debug PSoC 1 devices, it can also program PSoC 1 devices using ISSP.
  - CY3210-PSOCEVAL1 Kit enables you to evaluate and experiment Cypress's PSoC 1 programmable system-on-chip design methodology and architecture.
  - CY8CKIT-001 is a common development platform for all PSoC family devices.
- The MiniProg1 and MiniProg3 devices provide an interface for flash programming.

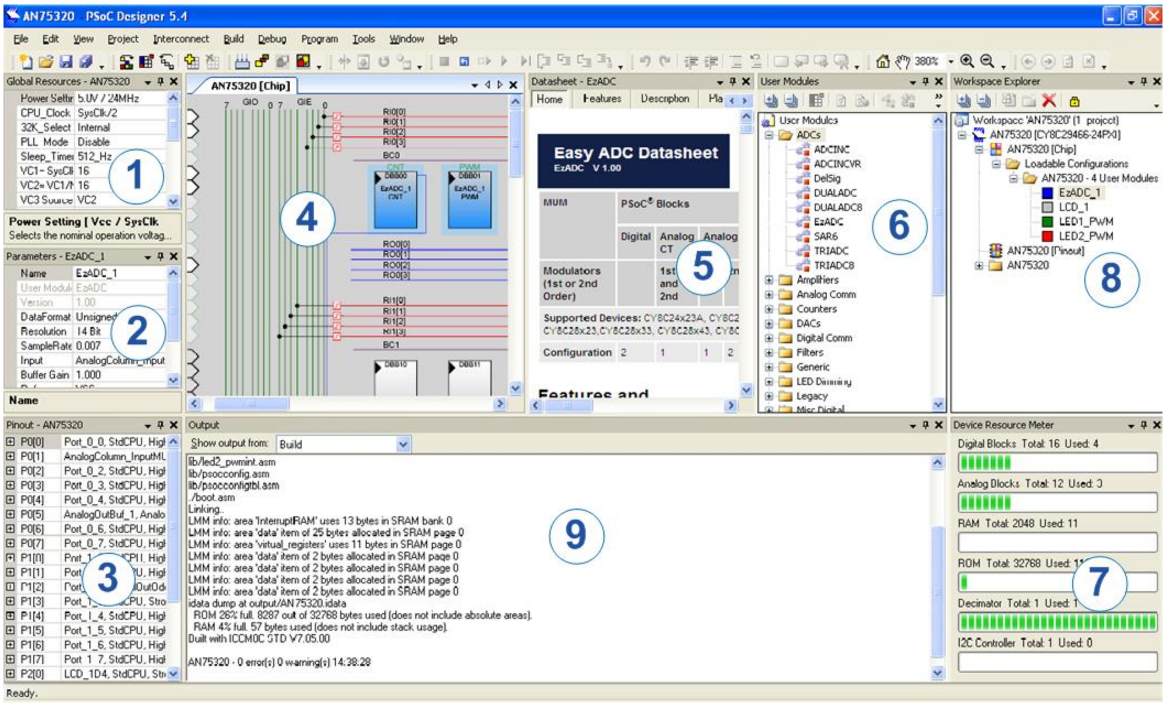
2.1 PSoC Designer

PSoC Designer is a free Windows-based Integrated Design Environment (IDE). Develop your applications using a library of pre-characterized analog and digital peripherals in a drag-and-drop design environment. Then, customize your design leveraging the dynamically generated API libraries of code. Figure 1 shows PSoC Designer windows. **Note:** This is not the default view.

1. **Global Resources –** all device hardware settings.
2. **Parameters –** the parameters of the currently selected User Modules.
3. **Pinout –** information related to device pins.
4. **Chip-Level Editor –** a diagram of the resources available on the selected chip.
5. **Datasheet –** the datasheet for the currently selected UM
6. **User Modules –** all available User Modules for the selected device.
7. **Device Resource Meter –** device resource usage for the current project configuration.
8. **Workspace –** a tree level diagram of files associated with the project.
9. **Output –** output from project build and debug operations.

Note: For detailed information on PSoC Designer, go to **PSoC® Designer > Help > Documentation > Designer Specific Documents > IDE User Guide**.

Figure 1. PSoC Designer Layout


AN75320 - PSoC Designer 5.4
File Edit View Project Interconnect Build Debug Program Tools Window Help
Global Resources - AN75320
Power Setting [Vcc / SysClk
Selects the nominal operation voltag...
Parameters - EzADC_1
Name
AN75320 [Chip]
Datasheet - EzADC
Home Features Description
Easy ADC Datasheet
EzADC V 1.00
User Modules
Workspace Explorer
Pinout - AN75320
Output
Device Resource Meter
Ready.
1
2
3
4
5
6
7
8
9

Power Settir	5.0V / 24MHz
CPU_Clock	SysClk/2
32K_Select	Internal
PLL Mode	Disable
Sleep_Timer	512_Hz
VC1= SysCl	16
VC2= VC1/N	16
VC3 Source	VC2

Name	EzADC_1
User Modul	EzADC
Version	1.00
DataFormat	Unsigned
Resolution	14 Bit
SampleRate	0.007
Input	AnalogColumn_Input
Buffer Gain	1.000

Supported Devices: CY8C24x23A, CY8C2 CY8C28x23,CY8C28x33, CY8C28x43, CY8C

Show output from: Build

```
lib/led2_pwmint.asm
lib/psocconfig.asm
lib/psocconfigtbl.asm
./boot.asm
Linking..
LMM info: area 'InterruptRAM' uses 13 bytes in SRAM bank 0
LMM info: area 'data' item of 25 bytes allocated in SRAM page 0
LMM info: area 'virtual_registers' uses 11 bytes in SRAM page 0
LMM info: area 'data' item of 2 bytes allocated in SRAM page 0
LMM info: area 'data' item of 2 bytes allocated in SRAM page 0
LMM info: area 'data' item of 2 bytes allocated in SRAM page 0
LMM info: area 'data' item of 2 bytes allocated in SRAM page 0
idata dump at output/AN75320.idata
  ROM 26% full. 8287 out of 32768 bytes used (does not include absolute areas).
  RAM 4% full. 57 bytes used (does not include stack usage).
Built with ICCM8C STD V7.05.00

AN75320 - 0 error(s) 0 warning(s) 14:38:28
```

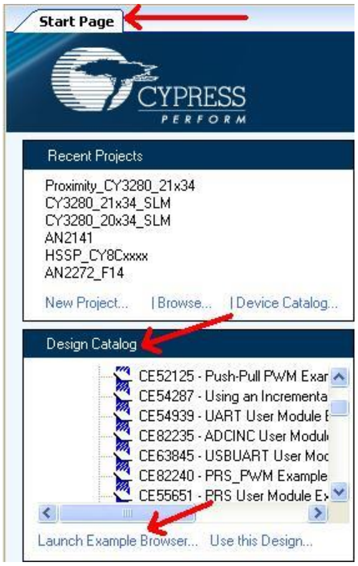
Digital Blocks Total: 16 Used: 4
Analog Blocks Total: 12 Used: 3
RAM Total: 2048 Used: 11
ROM Total: 32768 Used: 1
Decimator Total: 1 Used: 1
I2C Controller Total: 1 Used: 0

2.2 Code Examples

The following webpage lists the PSoC Designer based Code Examples. These Code Examples can speed up your design process by starting you off with a complete design, instead of a blank page and also show how PSoC Designer User modules can be used for various applications. For more information on PSoC 1 code examples, visit http://www.cypress.com/go/PSoC1Code Examples.

To access the Code Examples integrated with PSoC Designer, follow the path **Start Page > Design Catalog > Launch Example Browser** as shown in Figure 2.

Figure 2. Code Examples in PSoC Designer


Start Page
CYPRESS
PERFORM
Recent Projects
New Project... | Browse... | Device Catalog...
Design Catalog
Launch Example Browser... Use this Design...

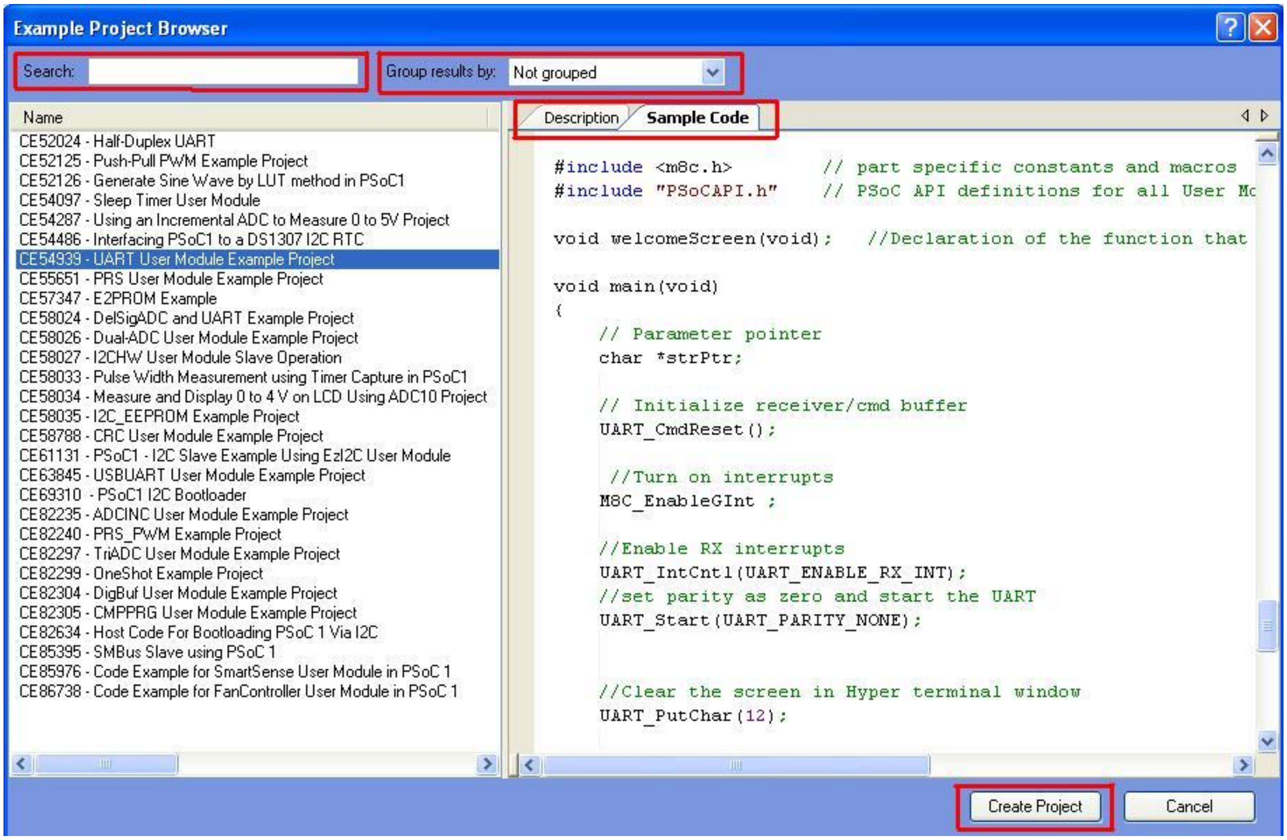
Proximity_CY3280_21x34
CY3280_21x34_SLM
CY3280_20x34_SLM
AN2141
HSSP_CY8Cxxxx
AN2272_F14

CE52125 - Push-Pull PWM Exar
CE54287 - Using an Incrementa
CE54939 - UART User Module E
CE82235 - ADCINC User Modul
CE63845 - USBUART User Moc
CE82240 - PRS_PWM Example
CE55651 - PRS User Module E›

In the Example Projects Browser shown in Figure 3, you have the following options:

- Keyword search to filter the projects.
- Listing the projects based on Category.
- Review the datasheet for the selection (on the Description tab).
- Review the code example for the selection. You can copy and paste code from this window to your project, which can help speed up code development, or
- Create a new project (and a new workspace if needed) based on the selection. This can speed up your design process by starting you off with a complete, basic design. You can then adapt that design to your application.

Figure 3. Code Example Projects, with Sample Codes



Group results by: Not grouped

CE52024 - Half-Duplex UART
CE52125 - Push-Pull PWM Example Project
CE52126 - Generate Sine Wave by LUT method in PSoC1
CE54097 - Sleep Timer User Module
CE54287 - Using an Incremental ADC to Measure 0 to 5V Project
CE54486 - Interfacing PSoC1 to a DS1307 I2C RTC
CE54939 - UART User Module Example Project
CE55651 - PRS User Module Example Project
CE57347 - E2PROM Example
CE58024 - DelSigADC and UART Example Project
CE58026 - Dual-ADC User Module Example Project
CE58027 - I2CHW User Module Slave Operation
CE58033 - Pulse Width Measurement using Timer Capture in PSoC1
CE58034 - Measure and Display 0 to 4 V on LCD Using ADC10 Project
CE58035 - I2C_EEPROM Example Project
CE58788 - CRC User Module Example Project
CE61131 - PSoC1 - I2C Slave Example Using EzI2C User Module
CE63845 - USBUART User Module Example Project
CE69310 - PSoC1 I2C Bootloader
CE82235 - ADCINC User Module Example Project
CE82240 - PRS_PWM Example Project
CE82297 - TriADC User Module Example Project
CE82299 - OneShot Example Project
CE82304 - DigBuf User Module Example Project
CE82305 - CMPPRG User Module Example Project
CE82634 - Host Code For Bootloading PSoC 1 Via I2C
CE85395 - SMBus Slave using PSoC 1
CE85976 - Code Example for SmartSense User Module in PSoC 1
CE86738 - Code Example for FanController User Module in PSoC 1

```
#include <m8c.h>        // part specific constants and macros
#include "PSoCAPI.h"    // PSoC API definitions for all User Mc

void welcomeScreen(void);   //Declaration of the function that

void main(void)
{
    // Parameter pointer
    char *strPtr;

    // Initialize receiver/cmd buffer
    UART_CmdReset();

     //Turn on interrupts
    M8C_EnableGInt ;

    //Enable RX interrupts
    UART_IntCntl(UART_ENABLE_RX_INT);
    //set parity as zero and start the UART
    UART_Start(UART_PARITY_NONE);


    //Clear the screen in Hyper terminal window
    UART_PutChar(12);
```

2.3 Technical Support

If you have any questions, our technical support team is happy to assist you. You can create a support request on the Cypress Technical Support page.

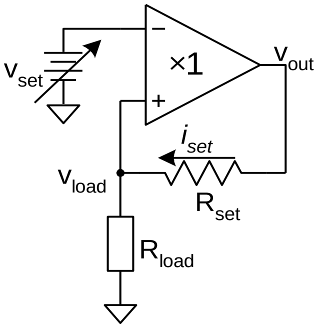
You can also use the following support resources if you need quick assistance.

- Self-help
- Local Sales Office Locations

3 Current Source

Figure 4 shows a classic design of a current source built with an In-Amp.

Figure 4. Instrumentation Amplifier Based Current Source



Equation 1 shows the value of i_{set} given V_{out} and V_{load}.

Equation 1 $$i_{set} = \frac{V_{out} - V_{load}}{R_{set}}$$

If the gain of the In-Amp is set to one, the Equation 2 holds true.

Equation 2 $$V_{out} = V_{load} - V_{set}$$

Combining these two equations results in Equation 3.

Equation 3 $$i_{set} = \frac{-V_{set}}{R_{set}}$$

Independent of the load resistance, the voltage across R_{set} is always equal to $–V_{set}$. Equation 3 defines current. This is independent of load resistance, considering the voltage across the resistor as constant.

Equation 3 is true only when Equation 2 holds true. The amplifier attempts to make the output, the difference of the inputs. V_{out} and V_{load} cannot exceed the legitimate operating range for the specific In-Amp. For example, you cannot drive a finite current into an infinite load. Therefore, it is possible to build a programmable current source with an In-Amp, a resistor, and a reference voltage.

4 Working with In-Amps and MIPS

There are several good In-Amps (Analog Devices AD620 and Burr Brown INA128). The PSoC INSAMP User Module is not capable of operating with only a gain of one. However, it is possible to build a unity gain difference amplifier (DiffAmp) using a Switched Capacitor Block User Module (SCBLOCK). Although the DiffAmp is not as robust as the In-Amp, you can include the following for the same price:

- A DAC to set the reference voltage
- A 4 MIPS CPU
- 4K to 32 Kbytes of Flash
- 256 bytes to 2K RAM
- 4 to 16 digital blocks
- 4 to 12 analog blocks

These are the prerequisites for building a complete application.

5 PSoC Implementation

Figure 5 shows the PSoC implementation of a current source built with a DAC, DiffAmp, and an external current setting resistor. PSoC Designer project is provided for this implementation.

DAC6_1 controls V_{set}. It feeds the negative input of **DiffAmp**. The amplifier's output is buffered (**Buf0**) and brought out on Port 0[3]. The input is brought in on Port 2[1] and connects to the positive input of **DiffAmp**. When the input is connected to the output via R_{set}, a current is sourced into R_{load}. Figure 6 shows the user module placement for the example.

The DiffAmp is the SCBLOCK User Module configured to be a differential unity gain amplifier. Figure 7 shows the parameter selection.

Note For background information on switch capacitor design, refer to the application note AN2041 – Understanding PSoC 1 Switched Capacitor Analog Blocks.

Note For DAC6 input between –28 and –31, the DAC column clock must be limited to 250 kHz. For a DAC6 input greater than –28, the column clock must be limited to a frequency not greater than 2 MHz.

Figure 5. PSoC Current Source Implementation

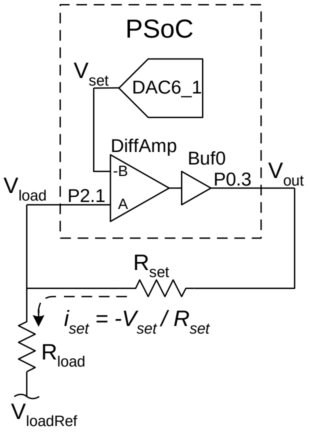


Figure 6. Current Source User Module Placement

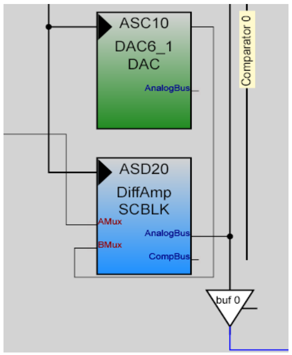
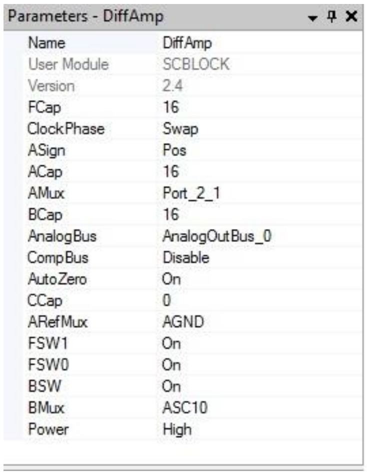


Figure 7. Parameter Selection for DiffAmp


Parameters - DiffAmp

Name	DiffAmp
User Module	SCBLOCK
Version	2.4
FCap	16
ClockPhase	Swap
ASign	Pos
ACap	16
AMux	Port_2_1
BCap	16
AnalogBus	AnalogOutBus_0
CompBus	Disable
AutoZero	On
CCap	0
ARefMux	AGND
FSW1	On
FSW0	On
BSW	On
BMux	ASC10
Power	High

6 Supply Rails

The circuit shown in Figure 5 is capable of driving a current into an increasing resistive load until V_{out} reaches infinity or at least attempts to exceed the V_{dd} one that comes first. As mentioned in the Working with In-Amps and MIPS section, the amplifier attempts to make the output the difference of the inputs. It does so if all the voltages are kept in the linear ranges.

There are four unique voltage nodes for this circuit. They are:

- $V_{loadRef}$

  This is the return path of the load resistor. It is a value you choose. It is V_{ss} (gnd), V_{dd}, or AGND. You can have any other voltage outside the supplies' rails as long as the other three voltages are in their linear range.

- V_{set}

  This voltage sets the current level. It is an input into a switched capacitor block and accepts inputs up to, and including, the supply rails as show in Equation 4.

  Equation 4 $$V_{ss} \le V_{set} + AGND \le V_{dd}$$

  DAC6 generates V_{set} value in the range of AGND ± V_{ref}. It is important that this voltage does not exceed the limits set in Equation 4.

- V_{load}

  Figure 5 shows this signal as an input into a switched capacitor block and accepts inputs up to, and including, the supply rails. Equation 5 defines this.

  Equation 5 $$V_{ss} \le V_{laad} \le V_{dd}$$

  Equation 6 shows that V_{load} is dependent on $V_{loadRef}$, R_{load}, and I_{set}.

  Equation 6 $$V_{load} = V_{load\mathrm{Re}\,f} + i_{set} \bullet R_{load}$$

  It is not possible to source a current into R_{load}, if $V_{loadRef}$ is set to V_{dd} or to sink a current from R_{load} if $V_{loadRef}$ is set to V_{ss}.

- V_{out}

  Figure 5 shows this signal as an output of an analog output buffer. Equation 7 shows it as dependent on i_{set}, R_{load}, and R_{set}.

  Equation 7 $$V_{out} = V_{loadRef} + i_{set} \cdot (R_{load} + R_{set})$$

  The limitation of this buffer's range is a function of the current it supplies. Each has its own limits which affects the correct linear operation.

  Figure 8 shows the typical range of an analog output buffer as a function of load impedance.

Figure 8. Buffer Range Vs Load Resistance, V_{dd} = 5 V

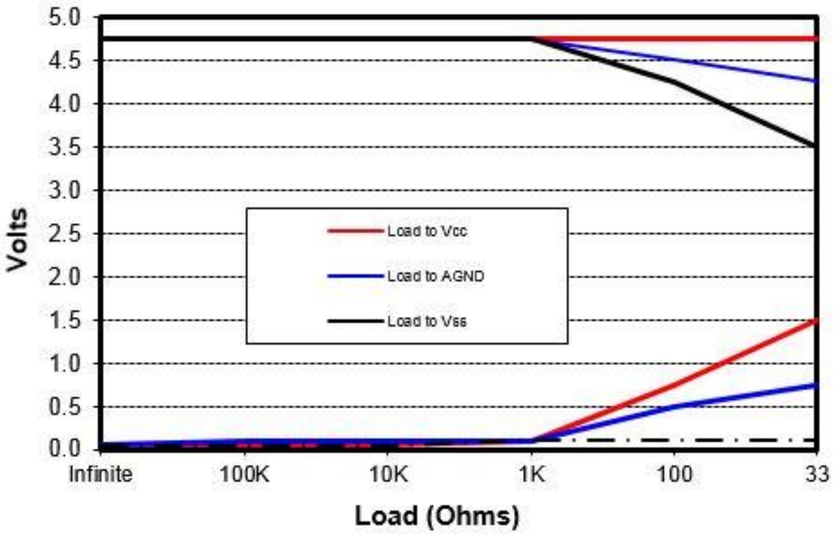
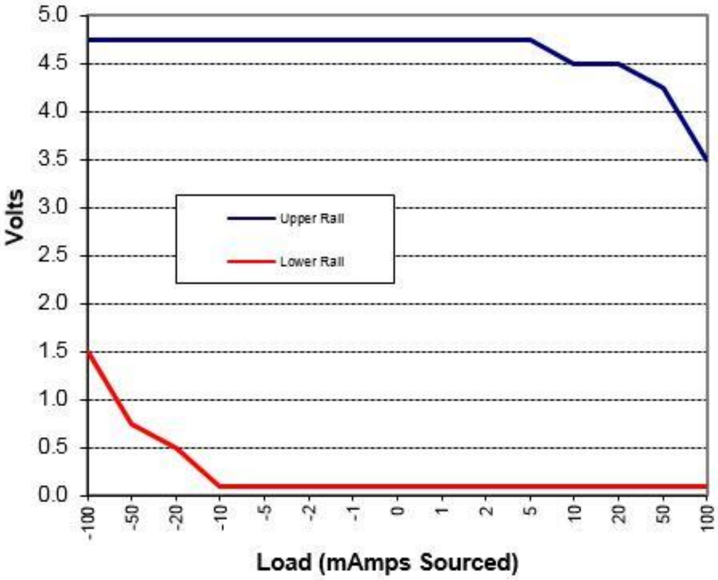


Figure 9 shows the range of an analog output buffer as a function of sourced load current.

Figure 9. Buffer Range Vs Load Current, V_{dd} = 5 V



Of course, "typical" is defined as the results from measuring one at the bench. V_{out} falls in the allowable ranges for Figure 8 and Figure 9.

7 Application Example

Figure 10 shows a typical transducer that converts some arbitrary physical parameter to a resistance.

It is ground-referenced and varies from 0 to 300 ohms. The manufacturer recommends a 4-mA stimulus to generate 0-V to 1.2-V response. This response is then digitized. This requires:

- Generating a 4 mA current.
- Shifting response to a valid PSoC ADC range.

V_{dd} of PSoC is set to 5.0 V. $V_{loadRef}$ is set to V_{ss} (gnd). AGND is set to ½V_{dd} or 2.5 V. The DAC6 sets V_{set} to 1.25 V. This meets the requirements set in Equation 4. Setting R_{set} to 316 ohms and using Equation 2, the current is set to 4 mA. V_{load} has a range from 0 to 1.2 V meeting the requirements set in Equation 5. Figure 9 shows that for a sourced current of 4 mA, V_{out} has a range of 0.1 to 4.75 V. V_{out} is 1.25V larger than V_{load} or 1.25V to 2.45 V, falling within the range margins.

Figure 10. Typical Transducer Requirements

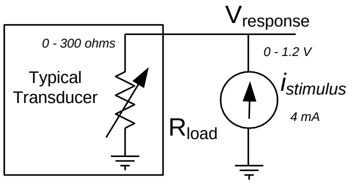
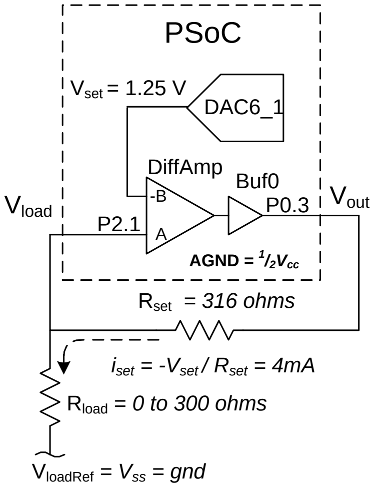


Figure 11. PSoC Implementation



8 Solutions for Port 2

The previous solution has V_{load} coming in on one of the port 2 switched-capacitor inputs. This enables a range that includes the supply rails. However, not all PSoC 1 devices have port 2. An alternate design is presented in Figure 12.

Figure 12 looks identical to the implementation in Figure 5 except that V_{load} now feeds a Programmable Gain Amplifier (PGA) configured as a unity gain **Buffer** that has its input on P0[1] and output feeding **DiffAmp2**.

Figure 12. PSoC Current Source, Sans Port 2

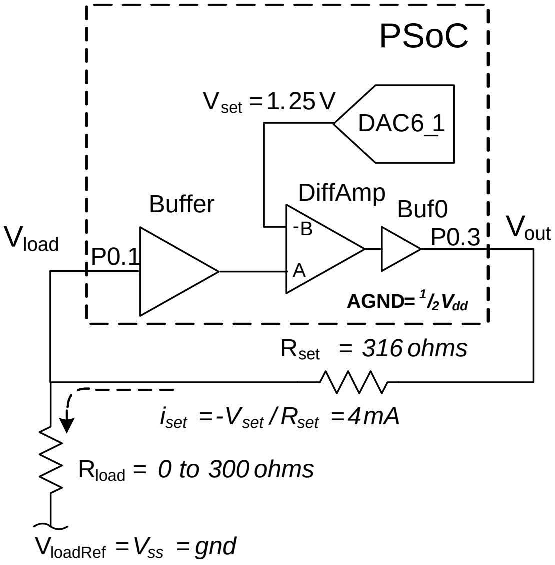
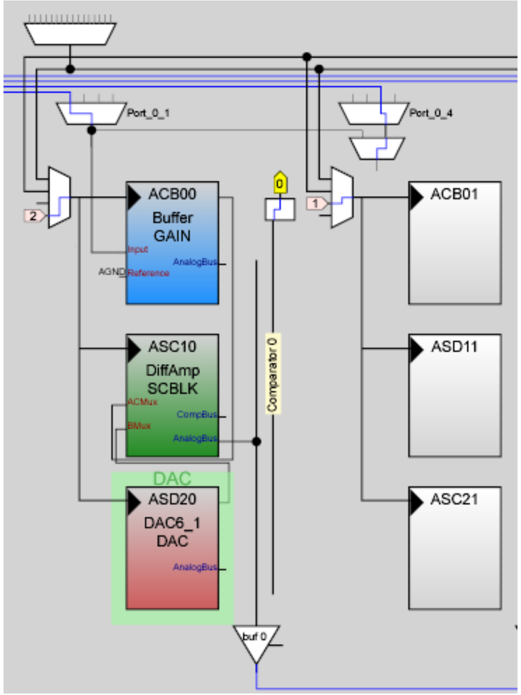


Figure 13. PSoC Implementation, Sans Port 2



The primary difference is the acceptable voltage range for V_{load}. This signal is now an input to the continuous time analog block. Equation 8 defines the allowable range for V_{load} taken from the specifications in the device datasheet.

Equation 8 $$(V_{ss} + 0.5) \le V_{load} \le (V_{dd} - 1)$$

The voltage across R_{load} ranges from 0 to 1.2 V. This violates the conditions of Equation 8. A simple solution is to add a 130 ohm resister in series with the transducer load to boost V_{load} by 0.52 V.

This ensures:

- V_{load} of span of 0.52 to 1.72 V
- V_{out} of span of 1.77 to 2.97 V

Both meet the range requirements set in Equation 8 and Equation 7. Of course, subtract the contributions of R_{shift} from your result.

9 Summary

Understanding instrumentation amplifiers permits the construction of bipolar current sources. The PSoC MCU architecture further enables you to build a complete programmable current source with only one external resistor.

About the Author

Name: Rajiv Vasanth Badiger

Title: Applications Engineer Staff

Document History

Document Title: AN36179 - PSoC® 1 Programmable Bipolar Analog Current Source

Document Number: 001-36179

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	1406903	SFV	03/30/2010	New application note.
*A	2552494	DWV	08/14/2008	Updated Section on DAC Clock Frequency.
*B	2901738	YARA	04/01/2010	Added reference to AN2089 in Document Properties. AN2089 is an older version of this application note.
*C	3739205	RJVB	09/10/2012	Updated in new template.
*D	4664928	ASRI	02/18/2015	Updated the project to PSoC Designer 5.4.
*E	4835488	ASRI	07/24/2015	Updated document template and reorganized the content. Added PSoC Resources. Updated the project to PSoC Designer 5.4 SP1. Updated the document title. Updated Supply Rails and Solutions for Port 2.

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

Automotive	cypress.com/go/automotive
Clocks & Buffers	cypress.com/go/clocks
Interface	cypress.com/go/interface
Lighting & Power Control	cypress.com/go/powerpsoc
Memory	cypress.com/go/memory
PSoC	cypress.com/go/psoc
Touch Sensing	cypress.com/go/touch
USB Controllers	cypress.com/go/usb
Wireless/RF	cypress.com/go/wireless

PSoC® Solutions

psoc.cypress.com/solutions

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

Cypress Developer Community

Community | Forums | Blogs | Video | Training

Technical Support

cypress.com/go/support

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

Phone : 408-943-2600
Fax : 408-943-4730
Website : www.cypress.com

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.