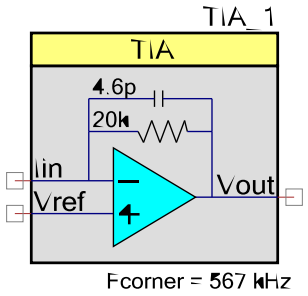
Trans-Impedance Amplifier (TIA)

1.90

Features



- Selectable conversion gain
- Selectable corner frequency
- Compensation for capacitive input sources
- Adjustable power settings
- Selectable input reference voltage

General Description

The Trans-Impedance Amplifier (TIA) component provides an opamp-based current-to-voltage conversion amplifier with resistive gain and user-selected bandwidth. It is derived from the SC/CT block.

The TIA is used to convert an external current to a voltage. Typical applications include the measurement of sensors with current outputs such as photodiodes. The conversion gain of the TIA is expressed in ohms, with the available range between 20 kΩ and 1.0 MΩ. Current output sensors, such as photodiodes, often have substantial output capacitance. This requires shunt feedback capacitance in the TIA to guarantee stability. The TIA has a programmable feedback capacitor to meet this need and provide bandwidth limiting to reduce broadband noise.

Input/Output Connections

This section describes the various input and output connections for the TIA. An asterisk (*) in the list of I/Os indicates that the I/O may be hidden on the symbol under the conditions listed in the description of that I/O.

Iin – Analog

The Iin is the input signal terminal. It is the sum of currents from the global inputs, which may include signals from a current output DAC.

Note This terminal name is `Iin` (uppercase `i`) **not** `lin` (lowercase `l`).

Vref – Analog

Vref is the input terminal for a reference signal. The reference may be an internal reference, internal VDAC value, or external signal.

Vout – Analog

Vout is the output signal terminal. Vout is determined by the following equation, where R_{FB} is resistive feedback:

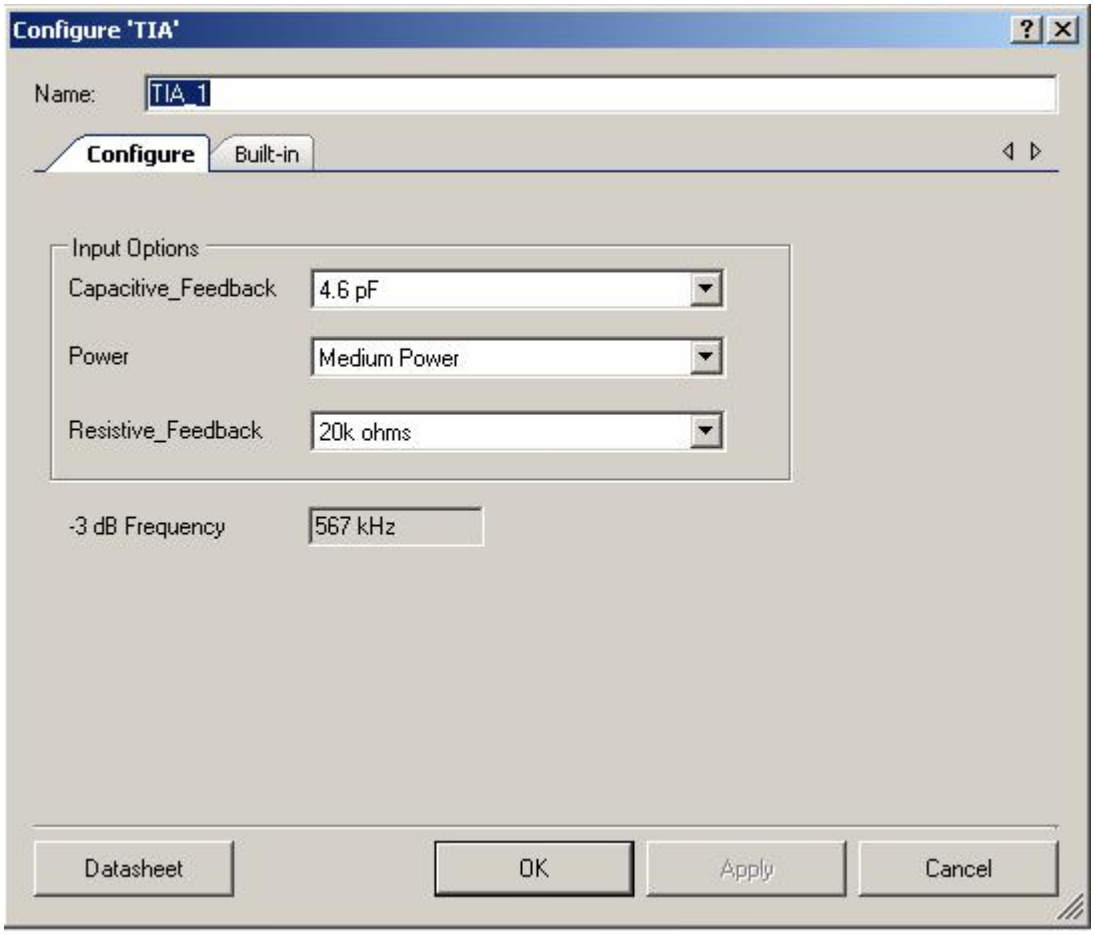
Vout = Vref – Iin × R_{FB}

Positive (from source) currents result in output voltage that is negative with respect to Vref.

Negative (into source) currents result in output voltage that is positive with respect to Vref.

Component Parameters

Drag a TIA component onto your design and double-click it to open the **Configure** dialog.



Capacitive_Feedback

This sets the capacitive feedback for the TIA. The capacitive feedback can be set to **None**, **1.3 pF**, **3.3 pF**, or **4.6 pF** (default). The –3 dB frequency for the TIA is calculated from the product of the values of resistive and capacitive feedback components.

Power

This sets the initial drive power of the TIA. The power determines the speed with which the TIA reacts to changes in the input signal. There are four power settings; **Minimum Power**, **Low Power**, **Medium Power** (default) and **High Power**. The **Minimum Power** setting results in the slowest response time and **High Power** the fastest. **Minimum** and **Low Power** settings have reduced drive currents and are not suitable for the lower values of feedback resistor.

Resistive_Feedback

This sets the nominal resistive feedback for the TIA. The resistive feedback may be selected from the following set of allowed values (in ohms): 20k (default), 30k, 40k, 80k, 120k, 250k, 500k and 1000k.

-3 db Frequency

This combobox is used to display the calculated value of bandwidth. This value depends on **Resistor_Feedback**, **Capacitive_Feedback** values and **Power** settings.

Application Programming Interface

Application Programming Interface (API) routines allow you to configure the component using software. The following table lists and describes the interface to each function. The subsequent sections cover each function in more detail.

By default, PSoC Creator assigns the instance name “TIA_1” to the first instance of a component in a given design. It can be renamed to any unique value that follows the syntactic rules for identifiers. The instance name becomes the prefix of every global function name, variable and constant symbol. For readability, the instance name used in the following table is “TIA.”

Function	Description
TIA_Start()	Powers up the TIA.
TIA_Stop()	Powers down the TIA.
TIA_SetPower()	Sets drive power to one of four levels.
TIA_SetResFB()	Sets the resistive feedback to one of eight values.
TIA_SetCapFB()	Sets the capacitive feedback to one of four values.
TIA_Sleep()	Stops and saves the user configurations.

TIA_Wakeup()	Restores and enables the user configurations.
TIA_Init()	Initializes or restores default TIA configuration.
TIA_Enable()	Enables the TIA.
TIA_SaveConfig()	Empty function. Provided for future use.
TIA_RestoreConfig()	Empty function. Provided for future use.

Global Variables

Variable	Description
TIA_initVar	Indicates whether the TIA has been initialized. The variable is initialized to 0 and set to 1 the first time TIA_Start() is called. This allows the component to restart without reinitialization after the first call to the TIA_Start() routine. If reinitialization of the component is required, then the TIA_Init() function can be called before the TIA_Start() or TIA_Enable() function.

void TIA_Start(void)

Description: Performs all of the required initialization for the component and enables power to the amplifier. The first time the routine is executed, the resistive and capacitive feedback and amplifier power are set based on the values provided during configuration. When called to restart the TIA following a TIA_Stop() call, the current component parameter settings are retained.

Parameters: None

Return Value: None

Side Effects: None

void TIA_Stop(void)

Description: Powers down TIA to its lowest power state and disables output.

Note This API is not recommended for use on PSoC 5 silicon. This device has a defect that causes connections to several analog resources to be unreliable when not powered. The unreliability manifests itself in silent failures (for example, unpredictably bad results from analog components) when the component using that resource is stopped. When using this silicon, all analog components in a design should be powered up (by calling their respective _Start() APIs, for instance TIA_Start()) at all times. Do not call the TIA_Stop() APIs.

Parameters: None

Return Value: None

Side Effects: Does not affect power, resistive or capacitive feedback settings

void TIA_SetPower(uint8 power)

Description: Sets the drive power to one of four settings; minimum, low, medium, or high.

Parameters: uint8 power: See the following table for valid power settings.

Power Setting	Notes
TIA_MINPOWER	Minimum active power and slowest reaction time
TIA_LOWPOWER	Low power and speed
TIA_MEDPOWER	Medium power and speed
TIA_HIGHPOWER	Highest active power and fastest reaction time

Return Value: None

Side Effects: None

void TIA_SetResFB(uint8 res_feedback)

Description: Set the amplifier resistive feedback value.

Parameters: uint8 res_feedback: See the following table for valid resistive feedback settings.

Gain Setting	Notes
TIA_RES_FEEDBACK_20K	Feedback resistor = 20k
TIA_RES_FEEDBACK_30K	Feedback resistor = 30k
TIA_RES_FEEDBACK_40K	Feedback resistor = 40k
TIA_RES_FEEDBACK_80K	Feedback resistor = 80k
TIA_RES_FEEDBACK_120K	Feedback resistor = 120k
TIA_RES_FEEDBACK_250K	Feedback resistor = 250k
TIA_RES_FEEDBACK_500K	Feedback resistor = 500k
TIA_RES_FEEDBACK_1000K	Feedback resistor = 1000k

Return Value: None

Side Effects: None

void TIA_SetCapFB(uint8 cap_feedback)

Description: Set the amplifier capacitive feedback value.

Parameters: uint8 cap_feedback: See the following table for valid capacitive feedback settings.

Gain Setting	Notes
TIA_CAP_FEEDBACK_NONE	No capacitive feedback
TIA_CAP_FEEDBACK_1_3PF	Feedback capacitor = 1.3 pF
TIA_CAP_FEEDBACK_3_3PF	Feedback capacitor = 3.3 pF
TIA_CAP_FEEDBACK_4_6PF	Feedback capacitor = 4.6 pF

Return Value: None

Side Effects: None

void TIA_Sleep(void)

Description: This is the preferred API to prepare the component for sleep. The TIA_Sleep() function saves the current component state. Then it calls the TIA_Stop() function and calls TIA_SaveConfig() to save the hardware configuration.

Call the TIA_Sleep() function before calling the CyPmSleep() or the CyPmHibernate() function. Refer to the PSoC Creator *System Reference Guide* for more information about power management functions.

Parameters: None

Return Value: None

Side Effects: None

void TIA_Wakeup(void)

Description: This is the preferred routine to restore the component to the state when TIA_Sleep() was called. The TIA_Wakeup() function calls the TIA_RestoreConfig() function to restore the configuration. If the component was enabled before the TIA_Sleep() function was called, the TIA_Wakeup() function will also re-enable the component.

Parameters: None

Return Value: None

Side Effects: Calling the TIA_Wakeup() function without first calling the TIA_Sleep() or TIA_SaveConfig() function may produce unexpected behavior.

void TIA_Init(void)

Description: Initializes or restores the component according to the customizer Configure dialog settings. It is not necessary to call TIA_Init() because the TIA_Start() routine calls this function and is the preferred method to begin component operation.

Parameters: None

Return Value: None

Side Effects: All registers will be set to values according to the customizer Configure dialog.

void TIA_Enable(void)

Description: Activates the hardware and begins component operation. It is not necessary to call TIA_Enable() because the TIA_Start() routine calls this function, which is the preferred method to begin component operation.

Parameters: None

Return Value: None

Side Effects: None

void TIA_SaveConfig(void)

Description: Empty function. Provided for future use.

Parameters: None

Return Value: None

Side Effects: None

void TIA_RestoreConfig(void)

Description: Empty function. Provided for future use.

Parameters: None

Return Value: None

Side Effects: None

Sample Firmware Source Code

PSoC Creator provides many example projects that include schematics and example code in the Find Example Project dialog. For component-specific examples, open the dialog from the Component Catalog or an instance of the component in a schematic. For general examples,

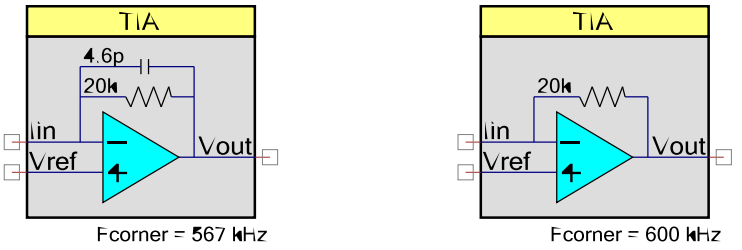
open the dialog from the Start Page or **File** menu. As needed, use the **Filter Options** in the dialog to narrow the list of projects available to select.

Refer to the “Find Example Project” topic in the PSoC Creator Help for more information.

Functional Description

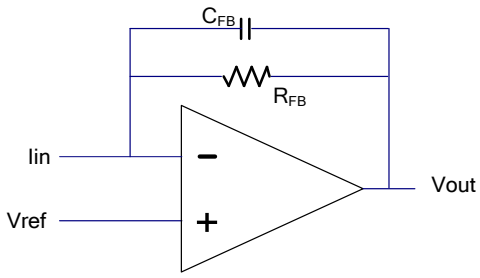
The TIA is constructed from a generic SC/CT block. The topology is an opamp with a selectable feedback resistor from the output to the inverting input. Optionally, a selectable feedback capacitor can also be connected between the output and the inverting input. Figure 1 shows the two possible TIA configurations.

Figure 1. TIA Configurations



The output voltage is controlled by adjusting the R_{FB} feedback resistor (see Figure 2). R_{FB} can be set to one of 8 values, between 20k and 1000k ohms, selectable in either the parameter dialog or using the TIA_SetResFB() API function.

Figure 2. TIA Schematic



The DC output level can be adjusted by adding current to the Iin terminal. Positive current (into the terminal) pushes the output negative; negative current (pulling current from the terminal) pushes the output positive. The source of the current can be an internal DAC.

The amplifier bandwidth is determined by the interaction between the feedback resistor R_{FB} and the selection of the capacitor in parallel with R_{FB}. The capacitive feedback value C_{FB} can be set to one of four values in either the parameter dialog or by using the TIA_SetCapFB() API function.

The –3 dB frequency for the amplifier is:

$$\textbf{Freq – 3 dB} = 1/(2\pi R_{FB} C_{FB})$$

Resources

The TIA uses one SC/CT analog block. Typically, the Vref input is routed from a voltage reference, a VDAC output, or an externally supplied reference on a GPIO.

API Memory Usage

The component memory usage varies significantly, depending on the compiler, device, number of APIs used and component configuration. The following table provides the memory usage for all APIs available in the given component configuration.

The measurements have been done with the associated compiler configured in Release mode with optimization set for Size. For a specific design the map file generated by the compiler can be analyzed to determine the memory usage.

Configuration	PSoC 3 (Keil_PK51)		PSoC 5 (GCC)		PSoC 5LP (GCC)	
	Flash Bytes	SRAM Bytes	Flash Bytes	SRAM Bytes	Flash Bytes	SRAM Bytes
Default	204	7	352	8	304	5

DC and AC Electrical Characteristics for PSoC 3

Specifications are valid for –40 °C ≤ T_A ≤ 85 °C and T_J ≤ 100 °C, except where noted. Specifications are valid for 1.71 V to 5.5 V, except where noted. Typical values are for T_A = 25 °C.

DC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
V_{IOFF}	Input offset voltage		–	–	10	mV
Rconv	Conversion resistance	R = 20k; 40-pF load	–25	–	+35	%
		R = 30k; 40-pF load	–25	–	+35	%
		R = 40k; 40-pF load	–25	–	+35	%
		R = 80k; 40-pF load	–25	–	+35	%
		R = 120k; 40-pF load	–25	–	+35	%
		R = 250k; 40-pF load	–25	–	+35	%
		R = 500k; 40-pF load	–25	–	+35	%
		R = 1M; 40 pF load	–25	–	+35	%
	Quiescent current		–	1.1	2.0	mA

AC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
BW	Input bandwidth (–3 dB)	R = 20k; –40-pF load	1500	–	–	kHz
		R = 120k; –40-pF load	240	–	–	kHz
		R = 1M; –40-pF load	25	–	–	kHz

DC and AC Electrical Characteristics for PSoC 5

Specifications are valid for –40 °C ≤ T_A ≤ 85 °C and T_J ≤ 100 °C, except where noted. Specifications are valid for 2.7 V to 5.5 V, except where noted. Typical values are for T_A = 25 °C.

DC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
V_{IOFF}	Input offset voltage		–	–	20	mV
R_{CONV}	Conversion resistance	R = 20k; 40-pF load	–25	–	+35	%
		R = 30k; 40-pF load	–25	–	+35	%
		R = 40k; 40-pF load	–25	–	+35	%
		R = 80k; 40-pF load	–25	–	+35	%
		R = 120k; 40-pF load	–25	–	+35	%
		R = 250k; 40-pF load	–25	–	+35	%
		R= 500k; 40-pF load	–25	–	+35	%
		R = 1M; 40-pF load	–25	–	+35	%
	Quiescent current		–	1.1	2	mA

AC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
BW	Input bandwidth (–3 dB)	R = 20k; 40-pF load	1000	–	–	kHz
		R = 120k; 40-pF load	230	–	–	kHz
		R = 1M; 40-pF load	23	–	–	kHz

Component Changes

This section lists the major changes in the component from the previous version.

Version	Description of Changes	Reason for Changes / Impact
1.90	Added PSoC 5LP support.	
	Added all APIs with the CYREENTRANT keyword when they are included in the .cyre file.	Not all APIs are truly reentrant. Comments in the component API source files indicate which functions are candidates. This change is required to eliminate compiler warnings for functions that are not reentrant used in a safe way: protected from concurrent calls by flags or Critical Sections.
	Updated DC and AC Electrical characteristics. Updated Resource and API memory usage sections.	

Version	Description of Changes	Reason for Changes / Impact
1.80	Modified source file to enable the charge pump when V_{DDA} is below 2.7 V	Charge pump should be enabled below 2.7 V
	Added DC and AC Electrical characteristics data for PSoC 5 to datasheet	
1.70	TIA_Stop() API modified for PSoC 5	Changes required to prevent the component from impacting unrelated analog signals when stopped on PSoC 5
	Added Debug window support	
	Added backward compatibility for register defines	To provide backward compatibility for TIA_1_10.
1.60	Updated the Configure dialog.	Created a customized interface. Added calculated bandwidth to customizer to support Bandwidth display.
	Removed Min-vdda parameter	Parameter for min Vdda is not required. Component will auto-recognize the voltage setting and set the block-internal switch pump accordingly.
	Updated TIA component symbol	TIA component symbol is updated to reflect Resistive Feedback, Capacitive Feedback, Fcorner value.
	Added characterization data to the datasheet.	
	Minor datasheet edits and updates	
1.50	Added Sleep/Wakeup and Init/Enable APIs.	To support low power modes, as well as to provide common interfaces to separate control of initialization and enabling of most components.
	TIA parameter Pull-down values are reordered in the ascending order.	The TIA parameter pull-down values are not in ascending order. The 80k ohm comes after 1000k ohm. Reordered the values accordingly.
	Changed the minus symbol to be the same length as horizontal stroke in the '+' character.	Updated the minus symbol to meet the industry standard.
	Updated a conditional statement to properly enable the charge pump clock for PSoC 3 Production silicon and PSoC 5 ES2 silicon or later.	The charge pump clock was not being enabled properly and therefore SC blocks were not working.

© Cypress Semiconductor Corporation, 2012. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, 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 products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

PSoC® is a registered trademark, and PSoC Creator™ and Programmable System-on-Chip™ are trademarks of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are property of the respective corporations.

Any Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

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.