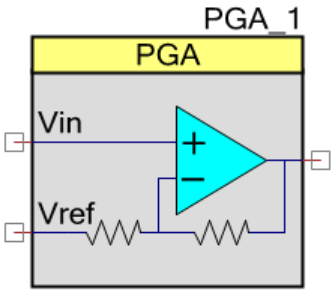
Please note that Cypress is an Infineon Technologies Company.
The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content
The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

Continuity of ordering part numbers
Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.

Programmable Gain Amplifier (PGA)

2.0



Features

- Gain steps from 1 to 50
- High input impedance
- Selectable input reference
- Adjustable power settings

General Description

The PGA implements an opamp-based, non-inverting amplifier with user-programmable gain. This amplifier has high input impedance, wide bandwidth and selectable input voltage reference. It is derived from the switched capacitor/continuous time (SC/CT) block.

The gain can be between 1 (0 dB) and 50 (+34 dB). The gain can be selected using the configuration window or changed at run time using the provided API. The maximum bandwidth is limited by the gain-bandwidth product of the opamp and is reduced as the gain is increased. The input of the PGA operates from rail to rail, but the maximum input swing (difference between Vin and Vref) is limited to V_{DDA}/Gain. The output of the PGA is class A, and is rail to rail for sufficiently high load resistance.

The PGA is used when an input signal has insufficient amplitude. A PGA can be put in front of a comparator, ADC, or mixer to increase the amplitude of the signal to these components. The PGA can be used as a unity gain amplifier to buffer the inputs of lower impedance blocks, including Mixers or inverting PGAs. A unity gain PGA can also be used to buffer the output of a VDAC or reference.

Input/Output Connections

This section describes the various input and output connections for the PGA. 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.

Vin – Analog

Vin is the input signal terminal.

Vref – Analog *

Vref is the input terminal for a reference signal.

The reference input can be connected to an external (to the component) reference or internal (to the component) V_{SS} (ground). When the reference is connected externally, the routing resistance is added to the internal resistors, slightly decreasing the gain and increasing the gain tolerance.

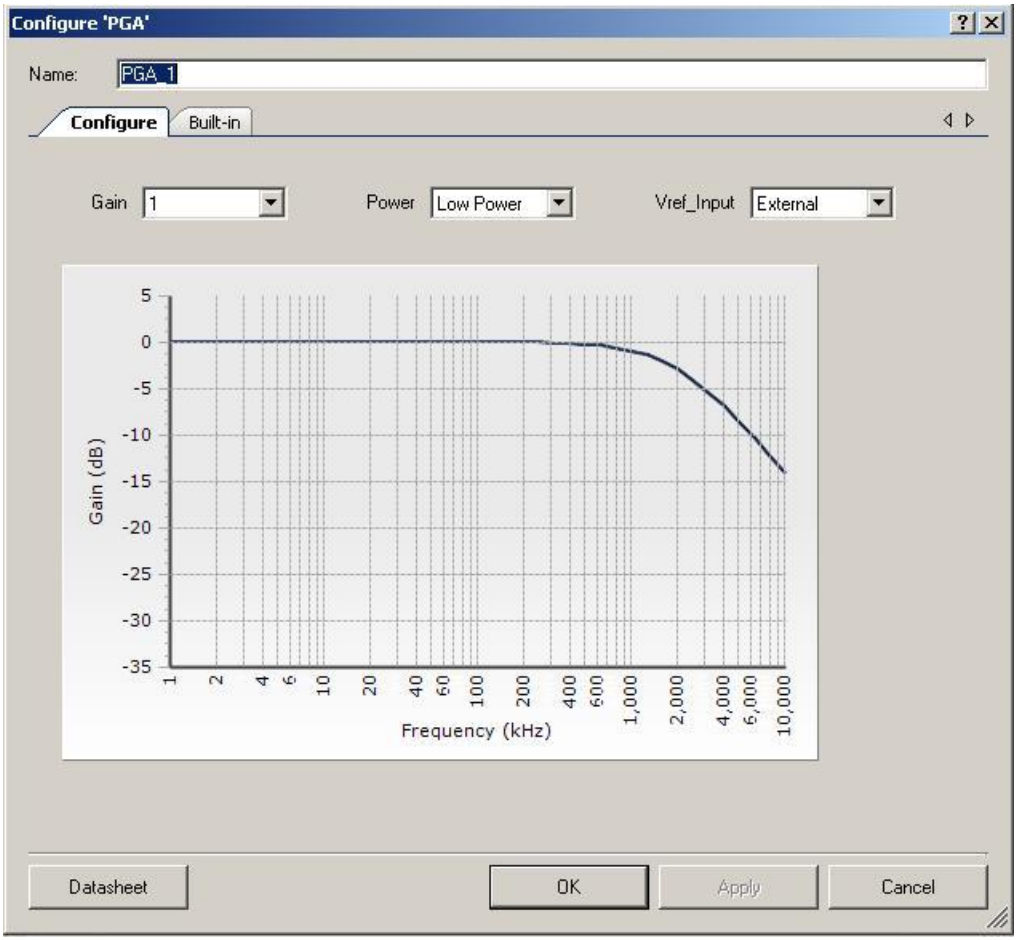
Vout – Analog

Vout is the output voltage signal terminal. Vout is a function of (Vin – Vref) times the specified Gain:

Vout = Vref + (Vin – Vref) × Gain

Component Parameters

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



Gain

This parameter sets the initial gain of the PGA. You can select the gain from the following set of allowed values: 1 (default), 2, 4, 8, 16, 24, 32, 48, and 50.

The following table shows the gain selection using Internal Resisters R_a and R_b

Gain	Rb	Ra
1	0	40k
2	40k	40k
4	120k	40k
8	280k	40k
16	600k	40k
24	460k	20k
32	620k	20k
48	470k	10k
50	490k	10k

Power

This parameter sets the initial drive power of the PGA. The power determines how fast the PGA reacts to changes in the input signal. There are four power settings: **Minimum Power**, **Low Power** (default), **Medium Power**, and **High Power**. A **Minimum Power** setting results in the slowest response time and a **High Power** setting results in the fastest response time.

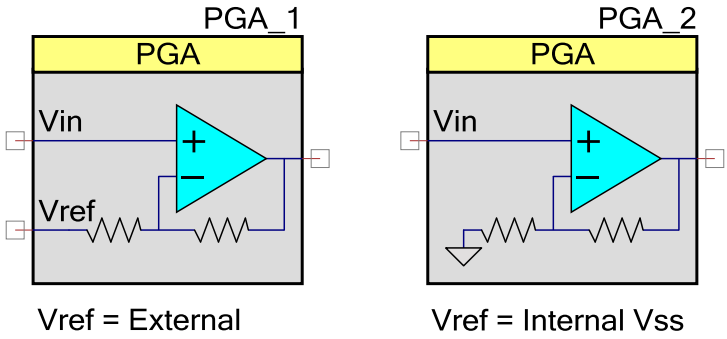
Vref_Input

This parameter is used to select the input voltage reference. The options include:

- **Internal Vss** – A ground signal internal to the component provides the amplifier reference.
- **External** (default) – A signal on the Vref terminal provides the amplifier reference.

The symbol displayed in PSoC Creator changes depending on the reference input selected.

Figure 1. PGA Configurations



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 “PGA_1” to the first instance of a component in a given design. You can rename it 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 “PGA.”

Function	Description
PGA_Start()	Starts the PGA
PGA_Stop()	Powers down the PGA
PGA_SetGain()	Sets gain to predefined constants
PGA_SetPower()	Sets drive power to one of four settings
PGA_Sleep()	Stops and saves the user configurations
PGA_Wakeup()	Restores and enables the user configurations
PGA_Init()	Initializes or restores default PGA configuration
PGA_Enable()	Enables the PGA
PGA_SaveConfig()	Empty function. Provided for future use.
PGA_RestoreConfig()	Empty function. Provided for future use.

Global Variables

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

void PGA_Start(void)

Description: This is the preferred method to begin component operation. This function turns on the amplifier with the power and gain based on the settings provided during the configuration or the current values after PGA_Stop() has been called.

Parameters: None

Return Value: None

Side Effects: None

void PGA_Stop(void)

Description: This function turns off PGA and enables its lowest power state.

Parameters: None

Return Value: None

Side Effects: None. Does not affect power or gain settings.

void PGA_SetPower(uint8 power)

Description: This function 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
PGA_MINPOWER	Minimum active power and slowest slew rate
PGA_LOWPOWER	Low power and slew rate
PGA_MEDPOWER	Medium power and slew rate
PGA_HIGHPOWER	Highest active power and fastest slew rate

Return Value: None

Side Effects: None

void PGA_SetGain(uint8 gain)

Description: This function sets the amplifier gain to a value between 1 and 50.

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

Gain Setting	Notes
PGA_GAIN_01	Gain = 1
PGA_GAIN_02	Gain = 2
PGA_GAIN_04	Gain = 4
PGA_GAIN_08	Gain = 8
PGA_GAIN_16	Gain = 16
PGA_GAIN_24	Gain = 24
PGA_GAIN_32	Gain = 32
PGA_GAIN_48	Gain = 48
PGA_GAIN_50	Gain = 50

Return Value: None

Side Effects: None

void PGA_Sleep(void)

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

Call the PGA_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 PGA_Wakeup(void)

Description: This is the preferred API to restore the component to the state when PGA_Sleep() was called. The PGA_Wakeup() function calls the PGA_RestoreConfig() function to restore the configuration. If the component was enabled before the PGA_Sleep() function was called, the PGA_Wakeup() function will also re-enable the component.

Parameters: None

Return Value: None

Side Effects: Calling the PGA_Wakeup() function without first calling the PGA_Sleep() or PGA_SaveConfig() function may produce unexpected behavior.

void PGA_Init(void)

Description: This function initializes or restores the component according to the customizer Configure dialog settings. It is not necessary to call PGA_Init() because the PGA_Start() API 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 PGA_Enable(void)

Description: This function activates the hardware and begins component operation. It is not necessary to call PGA_Enable() because the PGA_Start() API calls this function, which is the preferred method to begin component operation.

Parameters: None

Return Value: None

Side Effects: None

void PGA_SaveConfig(void)

Description: Empty function. Provided for future use.

Parameters: None

Return Value: None

Side Effects: None

void PGA_RestoreConfig(void)

Description: Empty function. Provided for future use.

Parameters: None

Return Value: None

Side Effects: None

MISRA Compliance

This section describes the MISRA-C:2004 compliance and deviations for the component. There are two types of deviations defined:

- project deviations – deviations that are applicable for all PSoC Creator components
- specific deviations – deviations that are applicable only for this component

This section provides information on component-specific deviations. Project deviations are described in the MISRA Compliance section of the *System Reference Guide* along with information on the MISRA compliance verification environment.

The PGA component does not have any specific deviations.

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.

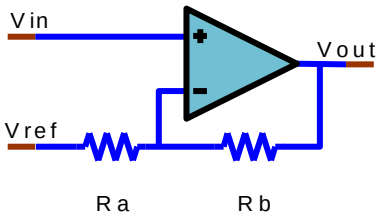
Refer to the “Find Example Project” topic in the PSoC Creator Help for more information.

Functional Description

The PGA is constructed from a generic SC/CT block. You can find details about this block in the applicable device datasheet and TRM, available on the Cypress website. The gain is selected by adjusting two resistors, Ra and Rb (see Figure 2. PGA Schematic). Ra may be set to either 10 kΩ, 20 kΩ, or 40 kΩ. Rb is set between 0 kΩ and 1000 kΩ to generate the gain values selectable in either the parameter dialog or the PGA_SetGain() function.

The block has a programmable capacitor in parallel with the feedback resistor, Rb. The value of the capacitor is configured for each gain selection to achieve guaranteed stability. Reassigning Rb values without also selecting the appropriate feedback capacitor value can result in PGA instability. Cypress strongly recommends that you use the provided APIs for gain changes.

Figure 2. PGA Schematic



The bandwidth of the PGA is determined by gain and power setting. Because of compensation capacitor and stability requirements, the bandwidth is somewhat reduced from the absolute maximum expected from the opamp's open loop gain-bandwidth.

Resources

The PGA component uses one SC/CT analog block.

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 5LP (GCC)	
	Flash Bytes	SRAM Bytes	Flash Bytes	SRAM Bytes
Default	228	22	332	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.

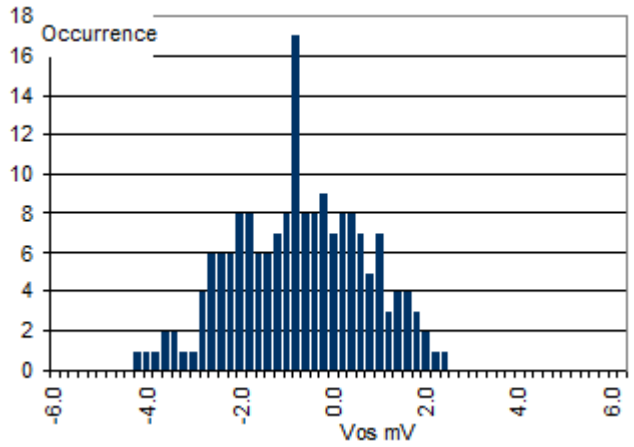
DC Characteristics

Parameter	Description	Conditions	Min	Typ	Max	Units
Vin	Input voltage range	Power Mode = minimum	Vssa	-	Vdda	V
V_{OS}	Input offset voltage	Power mode = high, gain = 1	-	-	10	mV
TCV_{OS}	Temp. coeff. input offset voltage, absolute value	Power mode = high, gain = 1	-	-	±30	µV/°C
Vonl	DC output nonlinearity	Gain = 1	-	-	±0.01	% of FSR
C_{IN}	Input capacitance		-	-	7	pF
Voh	Output voltage swing	Power mode = high, gain = 1, Rload = 100 kΩ to V_{DDA} / 2	V_{DDA}-0.15	-	-	V
Vol	Output voltage swing	Power mode = high, gain = 1, Rload = 100 kΩ to V_{DDA} / 2	-	-	V_{SSA}+0.15	V
Vsrc	Output voltage under load	Iload = 250 µA, Vdda ≥ 2.7V, power mode = high	-	-	300	mV
Ge1	Gain accuracy, deviation from nominal	G = 1, Vref internally connected to V_{SS}	–	0.01	0.15	+/-%
Ge2		G = 2, Vref internally connected to V_{SS}	–	0.1	1.0	
Ge4		G = 4, Vref internally connected to V_{SS}	–	0.5	1.35	
Ge8		G = 8, Vref internally connected to V_{SS}	–	0.6	1.6	
Ge16		G = 16, Vref internally connected to V_{SS}	–	0.7	2.5	
Ge32		G = 32, Vref internally connected to V_{SS}	–	0.85	5.0	
Ge50		G = 50, Vref internally connected to V_{SS}	–	2.1	5.0	
Gd1	Gain change versus temperature	G = 1, Vref internally connected to V_{SS}	N/A	1.2	2.5	ppm/°C
Gd2		G = 2, Vref internally connected to V_{SS}	–	8.6	20	
Gd4		G = 4, Vref internally connected to V_{SS}	–	13	29	
Gd8		G = 8, Vref internally connected to V_{SS}	–	15	35	
Gd16		G = 16, Vref internally connected to V_{SS}	–	18	40	
Gd32		G = 32, Vref internally connected to V_{SS}	–	38	75	

Parameter	Description	Conditions	Min	Typ	Max	Units
Gd50		G = 50, Vref internally connected to V_{SS}	–	167	400	
PSSR	Power supply rejection ratio		48	-	-	dB
I_{DD}	Operating current	Power mode = high	–	1.5	1.65	mA

Figures

Offset Voltage Histogram

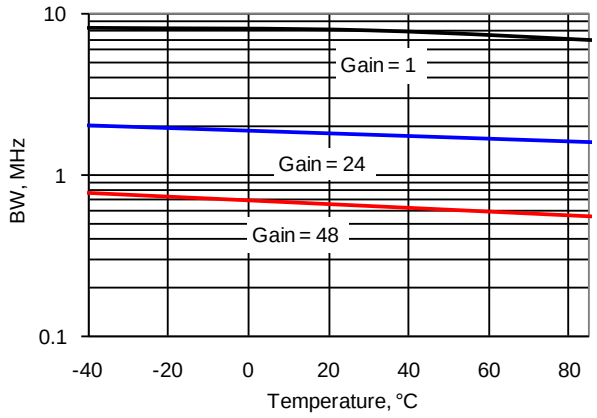


AC Characteristics

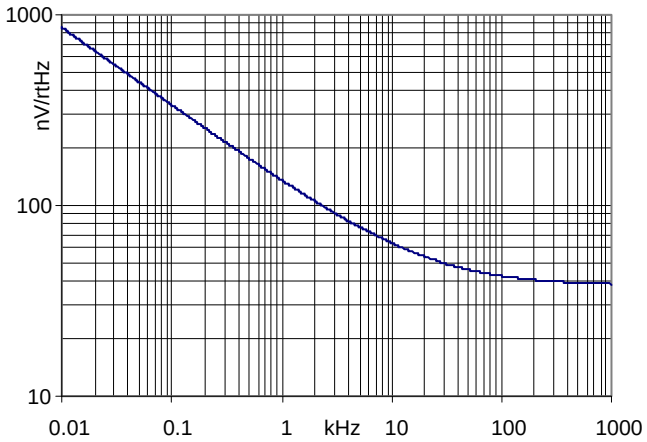
Parameter	Description	Conditions	Min	Typ	Max	Units
BW1	-3dB Bandwidth	Power mode = high, gain = 1, input = 100 mV peak-to-peak	6.7	8	-	MHz
SR_G1	Slew rate	20 - 80%, Gain = 1, P = High	3.0	4.8	N/A	V/µs
SR_G16		20 - 80%, Gain = 16, P = High	0.5	0.87	N/A	V/µs
SR_G50		20 - 80%, Gain = 50, P = High	0.25	0.84	N/A	V/µs
e_n	Input noise density	f = 100 kHz, P = High , Vdda = 5V	-	43	-	nV/sqrtHz

Figures

Bandwidth versus Temperature, at Different Gain Settings, Power = High



Voltage noise, V_{DDA} = 5 V, Power Mode = High



DC and AC Electrical Characteristics for PSoC 5LP

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.

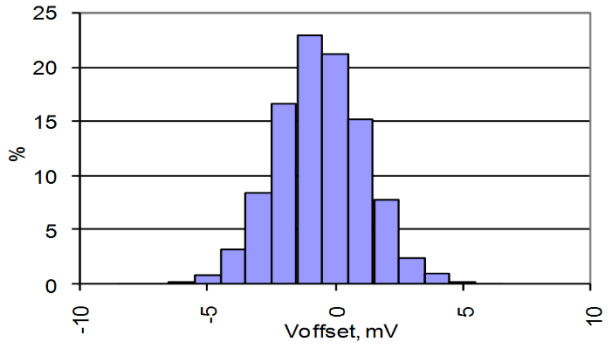
DC Characteristics

Parameter	Description	Conditions	Min	Typ	Max	Units
Vin	Input voltage range	Power mode = minimum	V_{SSA}	–	V_{DDA}	V
V_{OS}	Input offset voltage	Power mode = high, gain = 1	–	–	20	mV
TCV_{OS}	Input offset voltage drift with temperature	Power mode = high, gain = 1	–	–	±30	µV/°C
Ge1	Gain error, gain = 1		–	–	±2	%
Ge16	Gain error, gain = 16		–	–	±8	%
Ge50	Gain error, gain = 50		–	–	±10	%

Parameter	Description	Conditions	Min	Typ	Max	Units
V_{ONL}	DC output nonlinearity	Gain =1	–	–	±0.1	%of FSR
C_{IN}	Input capacitance		–	–	7	pF
V_{OH}	Output swing	Power mode = high, gain = 1, R_{LOAD} = 100 kΩ to $V_{DDA}/2$	V_{DDA} – 0.15	–	–	V
V_{OL}	Output swing	Power mode = high, gain = 1, R_{LOAD} = 100 kΩ to $V_{DDA}/2$	–	–	V_{SSA} + 0.15	V
V_{SRC}	Output voltage under load	I_{LOAD} = 250 µA, power mode = high	–	–	300	mV
I_{DDA}	Operating current	Power mode = high	–	1.5	1.65	mA
PSRR	Power supply rejection ratio		48	–	–	dB

Figures

Offset Voltage Histogram

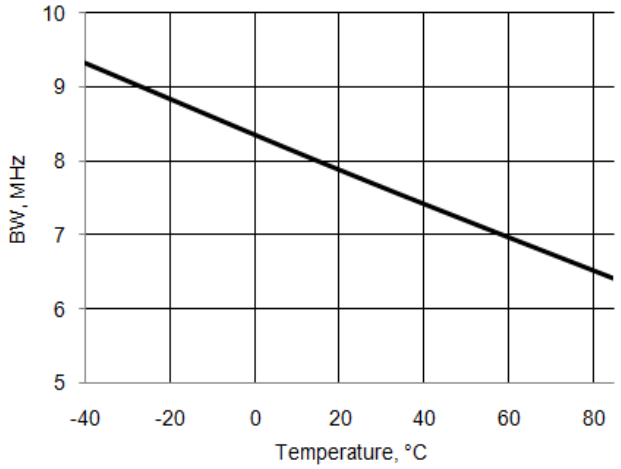


AC Characteristics

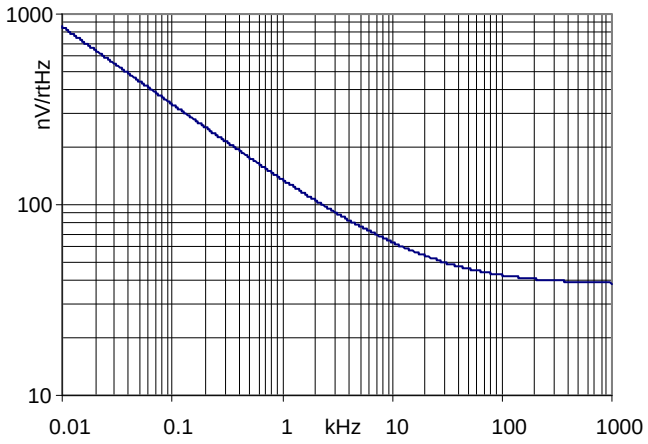
Parameter	Description	Conditions	Min	Typ	Max	Units
BW1	–3 dB bandwidth	Power mode = high, gain = 1, non-inverting mode, 300 mV ≤ V_{IN} ≤ V_{DDA} – 1.2 V, C_L ≤ 25 pF	6	8	–	MHz
SR1	Slew rate	Power mode = high, gain = 1, 20% to 80%	3.0	–	–	V/µs
e_n	Input noise density	Power mode = high, V_{DDA} = 5 V at 100 kHz	–	43		nV/sqrtHz

Figures

Bandwidth versus Temperature, Gain = 1, Power Mode = High



Noise versus Frequency, V_{DDA} = 5 V, Power Mode = High



Component Changes

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

Version	Description of Changes	Reason for Changes / Impact
2.0.d	Minor datasheet edit.	
2.0.c	Minor datasheet edit.	Fixed corrupted images/charts.
2.0.b	Minor datasheet edit.	
2.0.a	Edited datasheet to remove references to PSoC 5.	PSoC 5 has been replaced by the PSoC 5LP.
2.0	Added variable Vdda support.	
	Added MISRA Compliance section.	The component does not have any specific deviations.
1.90	Schematic and source file updates to use a single shared boost clock for all SC/CT block components.	A single shared clock will be used as boost clock by all SC/CT block components
	For low voltage VDDA operation uses a boost clock shared by all the SC/CT based components.	Reduces the number of analog clocks required in the system for boost clocks. With this change a single boost clock is shared instead of using a separate clock for each SC/CT based component.
1.80	Added PSoC 5LP support	
	Changed resistor combination for Gain = 24	Resistor values were incorrect
	Added all component APIs with the CYREENTRANT keyword	
	Minor GUI updates	
1.70.a	Added DC and AC Electrical characteristics data for PSoC 5	
1.70	Changed PGA_Stop() API for PSoC 5	Change required to prevent the component from impacting unrelated analog signals when stopped, when using PSoC 5.
	Updated PGA response graph	Change required to dynamically resize graph to fit window and to add horizontal and vertical grids.
1.60	Removed VDDA parameter from component customizer	VDDA setting in the component is redundant and unnecessary for multiple components. The parameter was removed and the component queries the global setting for minimum VDDA in the DWR and automatically enables the pump when necessary.
	Created configuration window to include frequency response graphs for an easier to use GUI.	Previous configuration window did not provide enough information for ease of use.

Version	Description of Changes	Reason for Changes / Impact
	Corrected SetGain constants in the header file	The constants provided for the SetGain API had incorrect values. These have been corrected.
	Added characterization data to datasheet	
1.50	Minor datasheet edits and updates	
	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.
	Removed Gain setting of 25.	The gain of 25 was too close to other values and therefore offered no value.
	Updated the symbol image and Configure dialog.	These were updated to comply with corporate standards.
	Changed the names of the registers by adding "_REG."	Updated to comply with coding guidelines.
	Added specification table and graphic placeholders	Data to be provided when characterization is complete.

© Cypress Semiconductor Corporation, 2012-2017. 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.