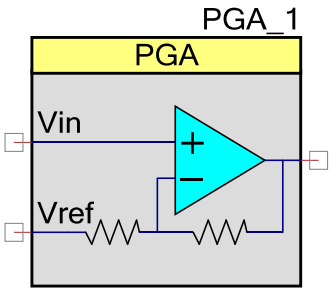
Programmable Gain Amplifier (PGA)

1.70

Features



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

General Description

The PGA implements an opamp-based, noninverting 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). You can select the gain using configuration or change it 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. You can put a PGA 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. You can also use a unity gain PGA 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.

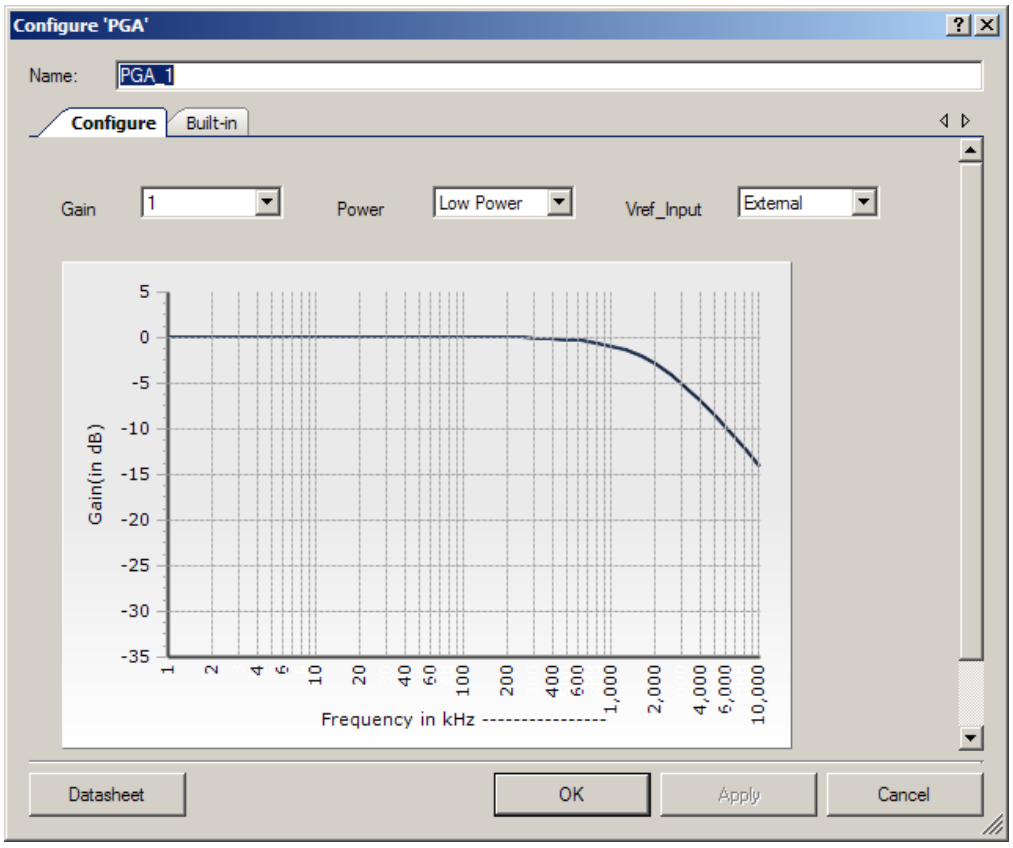
Vout – Analog

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

$$Vout = Vref + (Vin – Vref) \times Gain$$

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. The gain may be selected from the following set of allowed values: 1 (default), 2, 4, 8, 16, 24, 32, 48, and 50.

Power

This parameter sets the initial drive power of the PGA. The power determines the speed with which 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.

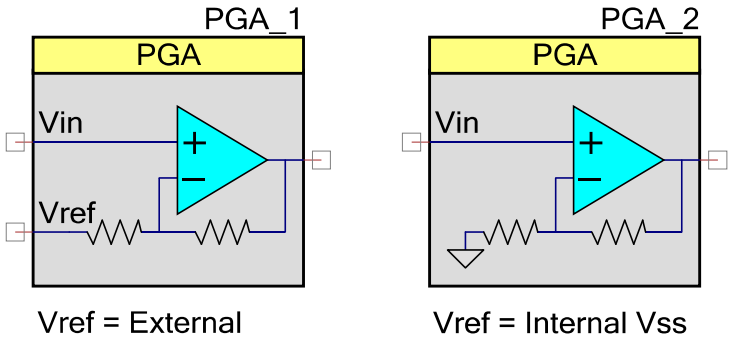
Vref_Input

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



Resources

The PGA uses one SC/CT block. You can find detailed information about this block in the applicable device datasheet and Technical Reference Manual (TRM). These documents are available on the Cypress website.

Analog Blocks	Digital Blocks					API Memory (Bytes)		Pins (per External I/O)
	Datapaths	Macro cells	Status Registers	Control Registers	Counter7	Flash	RAM	
1 SC/CT fixed block	N/A	N/A	N/A	N/A	N/A	381	20	3

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. Turns on the amplifier with the power and gain based on the settings provided during the configuration or the current values after a PGA_Stop() has been called.

Parameters: None

Return Value: None

Side Effects: None

void PGA_Stop(void)

Description: Turns off PGA and enables its lowest power state.

Note This API is not recommended for use on PSoC 3 ES2 and PSoC 5 silicon. These devices have 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 PGA_Start()) at all times. Do not call the PGA_Stop() APIs.

Parameters: None

Return Value: None

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

void PGA_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
PGA_MINPOWER	Minimum active power and slowest reaction time
PGA_LOWPOWER	Low power and speed
PGA_MEDPOWER	Medium power and speed
PGA_HIGHPOWER	Highest active power and fastest reaction time

Return Value: None

Side Effects: None

void PGA_SetGain(uint8 gain)

Description: 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: 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: 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

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.

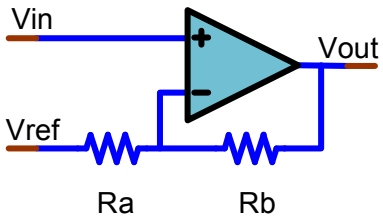
Refer to the “Find Example Project” topic in the PSoC Creator Help for more information.

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 20 kΩ or 40 kΩ. Rb is set between 20 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.

Registers

The PGA component configuration is implemented in registers SC[0..3]_CR0, SC[0..3]_CR1, and SC[0..3]_CR2. These can be accessed in your code by referring to the instantiated component name, for example, PGA_1_CR0_REG. The register contents can be reviewed in the PSoC Creator component debug window. See the applicable TRM, available on the Cypress website, for a detailed description of each register. The following registers are displayed in the PGA component debug window.

Register: PGA_1_CR0_REG

Name: Switched Capacitor Control Register 0

Description: Register bits 3:1 configure the switch capacitor block operating mode. This field is set to 110b for the PGA component.

Register: PGA_1_CR1_REG

Name: Switched Capacitor Control Register 1

Description: Register fields configure drive mode, compensation capacitor values, and gain setting of the switch capacitor block.

Register: PGA_1_CR2_REG

Name: Switched Capacitor Control Register 2

Description: Register fields configure the input impedance, feedback impedance, and the reference ground selection for the switch capacitor block.

Register: PGA_1_PM_ACT_CFG_REG

Name: Active Power Mode Configuration Register 9

Description: Register bits 3:0 enable power to each of the four switch capacitor blocks.

DC and AC Electrical Characteristics for PSoC 3

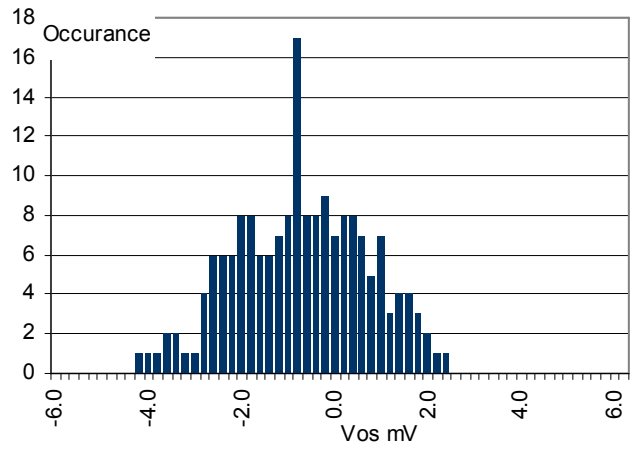
The following values are based on characterization data. Specifications are valid for –40 °C ≤ T_A ≤ 85 °C and T_J ≤ 100 °C except where noted. Unless otherwise specified in the following tables, all Typical values are for T_A = 25 °C, V_{DDA} = 5.0 V, Power = High, output referenced to analog ground, V_{SSA}.

5.0-V/3.3-V DC Electrical Characteristics

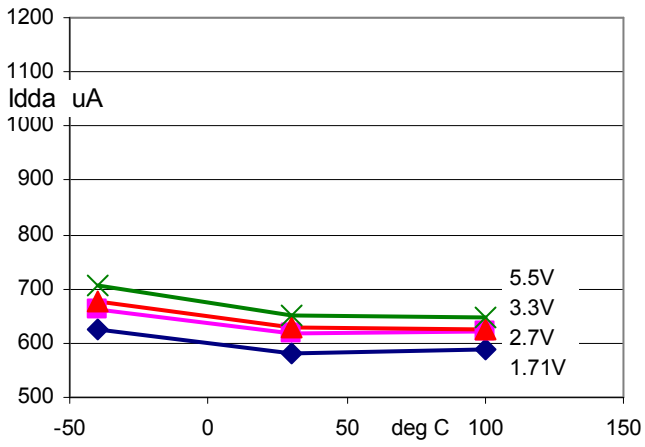
Parameter	Description	Conditions	Min	Typ	Max	Units
Input						
V_{OS}	Input offset voltage	All power modes (High, Medium, Low, Minimum)	N/A	3.5	10	mV
TCV_{OS}	Temp. coeff. input offset voltage, absolute value	All power modes (High, Medium, Low, Minimum)	N/A	6.0	12.3	µV/°C
C_{IN}	Input capacitance	Positive gain, noninverting input, not including pin and routing capacitance	N/A	2.0	N/A	pF
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	
Gd50		G = 50, Vref internally connected to V_{SS}	–	167	400	
Vout_range	Output swing	Gain = 1, R_{LOAD} = 100k to V_{DDA}/2, Difference from V_{DDA} or V_{SSA}	–	–	150	mV
I_{DDA}	Operating current	V_{DDA} = 1.71 V, P = Low	–	700	1000	µA
		V_{DDA} = 5.0 V, P = High	–	1100	1350	µA

Figures

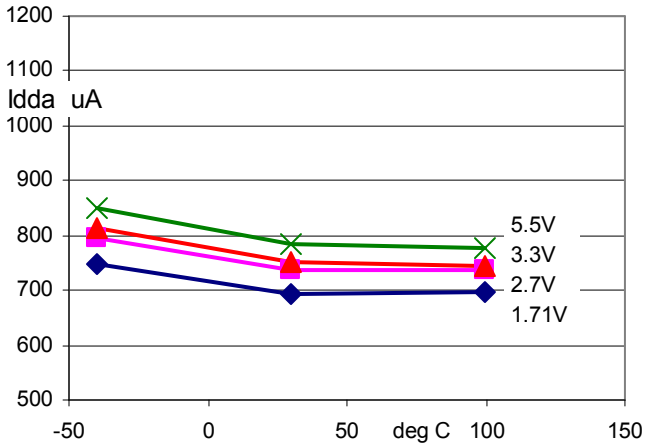
Histogram Input Offset Voltage



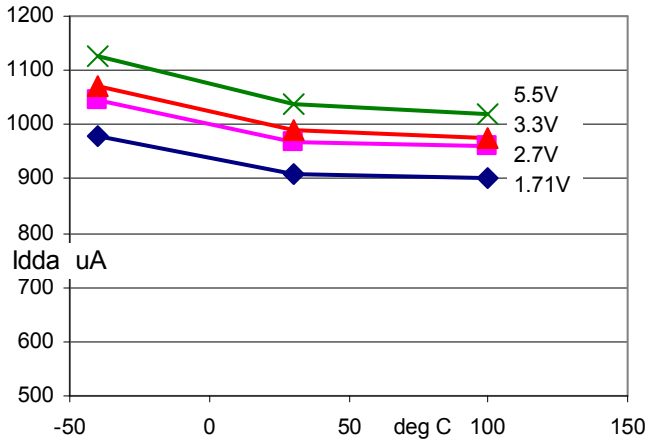
Typical Operating Current versus Temp, Power = Minimum



Typical Operating Current versus Temp, Power = Low



Typical Operating Current versus Temp, Power = High

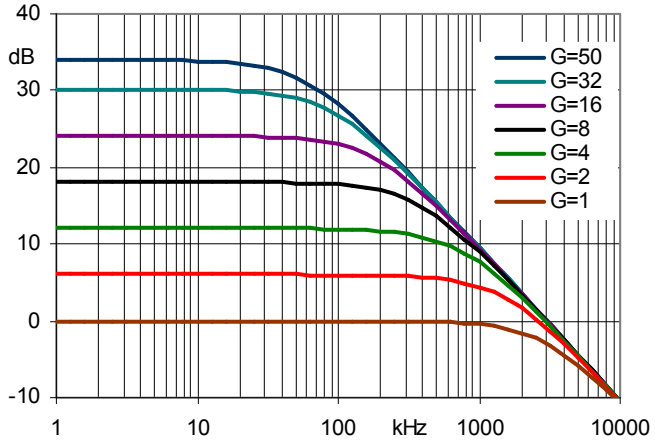


5.0-V/3.3-V AC Electrical Characteristics

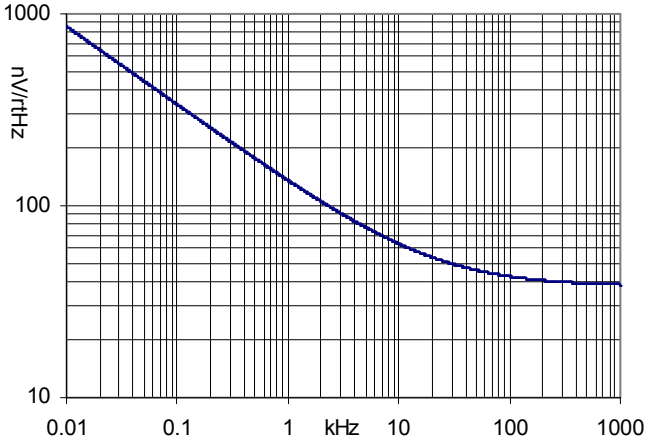
Parameter	Description	Conditions	Min	Typ	Max	Units
GBW_H	Gain bandwidth product, P = High	Gain = 1, V_{DDA} = 5.0 V, 25 °C	7.0	9.0	N/A	MHz
SR_G1	Slew rate	20 - 80%, Gain = 1, P = High	3.0	4.8	N/A	V/µs
SR_G16		20 - 80%, Gain = 16, P = High	0.5	0.87	N/A	V/µs
SR_G50		20 - 80%, Gain = 50, P = High	0.25	0.84	N/A	V/µs
PSRR_AC	Power supply rejection ratio	f = 100 kHz	48			dB
Vn		f = 100 kHz, P = High	na	42	N/A	nV/rtHz

Figures

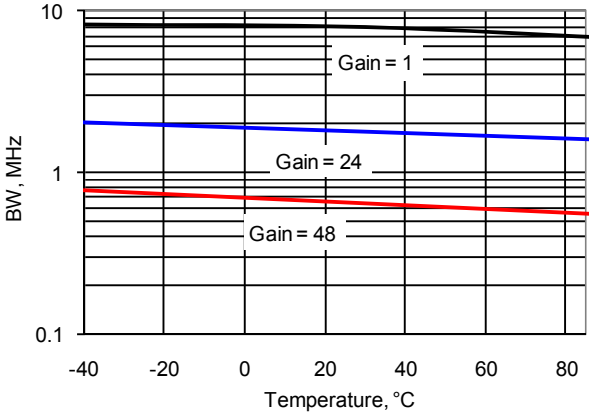
Typical Gain versus Frequency, Power = High



Voltage noise, VDDA = 5.0V, Power = High



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



DC and AC Electrical Characteristics for PSoC 5

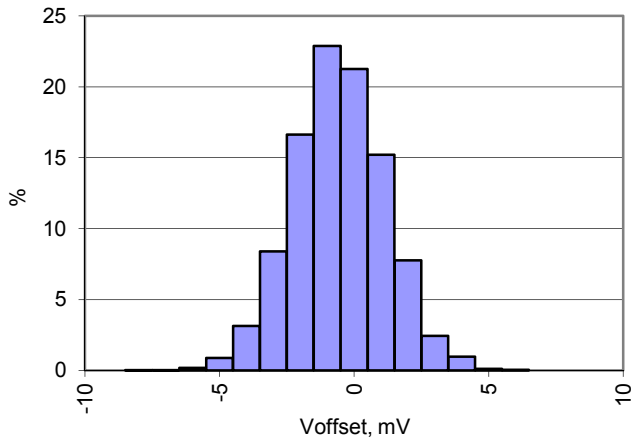
The following values are based on characterization data. Specifications are valid for –40 °C ≤ T_A ≤ 85 °C and T_J ≤ 100 °C except where noted. Unless otherwise specified in the following tables, all Typical values are for T_A = 25 °C, V_{DDA} = 5.0 V, Power = High, output referenced to analog ground, V_{SSA}.

5.0-V/3.3-V DC Electrical Characteristics

Parameter	Description	Conditions	Min	Typ	Max	Units
Input						
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
C_{IN}	Input capacitance		–	–	7	pF
Ge1	Gain error, gain = 1		–	–	±2	%
Ge16	Gain error, gain = 16		–	–	±8	%
Ge50	Gain error, gain = 50		–	–	±10	%
V_{ONL}	DC output nonlinearity	Gain =1	–	–	±0.1	%of FSR
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

PGA Voffset Histogram, 4096 samples/1024 parts

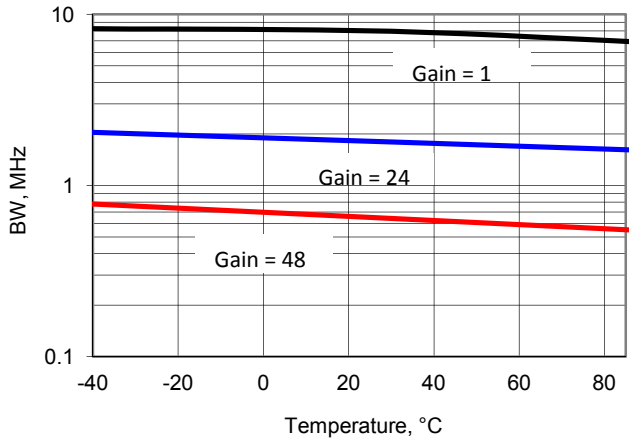


5.0-V/3.3-V AC Electrical Characteristics

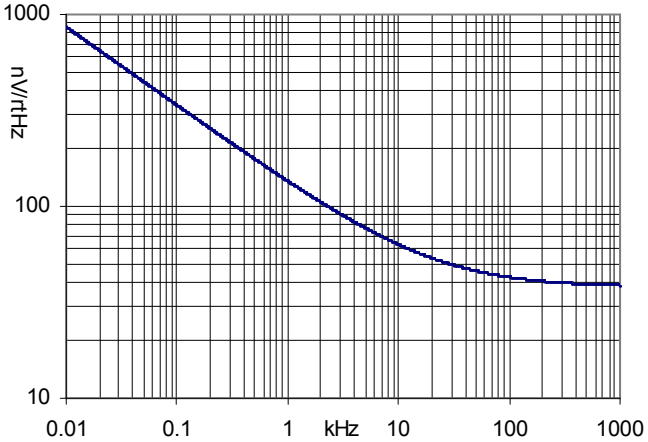
Parameter	Description	Conditions	Min	Typ	Max	Units
BW1	–3 dB bandwidth	Power mode = high, gain = 1, noninverting mode, 300 mV ≤ V_{IN} ≤ V_{DDA} – 1.2 V, C_L ≤ 25 pF	6	8	–	MHz
SR1	Slew rate	Power mode = high, gain = 1, 20% to 80%	3.0	–	–	V/µs
e_n	Input noise density	Power mode = high, V_{DDA} = 5 V at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

Figures

Bandwidth versus Temperature, at Different Gain Settings, 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
1.70.a	Added DC and AC Electrical characteristics data for PSoC 5	
1.70	PGA_Stop() API modified 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.
	Configuration window created to include. Frequency response graphs a better ease of use GUI.	Previous configuration window did not provide enough information for ease of use.
	SetGain constants corrected 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, 2011. 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.