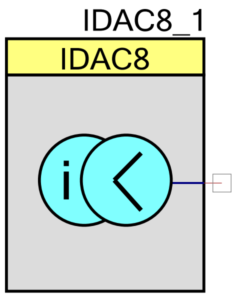
8-Bit Current Digital to Analog Converter (IDAC8)

1.80

Features



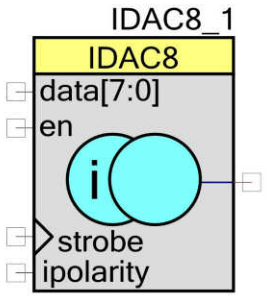
- Three ranges: 2040 µA, 255 µA, and 31.875 µA
- Current sink or source selectable
- Software- or clock-driven output strobe
- Data source may be CPU, DMA, or UDB

General Description

The IDAC8 component is an 8-bit current output DAC (Digital to Analog Converter). The output can source or sink current in three ranges. The IDAC8 can be controlled by hardware, software, or by a combination of both hardware and software.

Input/Output Connections

This section describes the various input and output connections for the IDAC8. 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.



Iout – Analog

The Iout terminal, the terminal on the right side of the symbol, is the connection to the DAC's current source/sink. It can be routed to any analog-compatible pin on the device. When the highest current range is selected (2040 µA), the output should only be routed to a specific set of pins that provide a direct low resistive path. These port pins are P0[6], P0[7], P3[0], or P3[1].

data[7:0] – Input *

This 8-bit-wide data signal connects the IDAC8 directly to the DAC bus. The DAC bus may be driven by UDB-based components or control registers, or routed directly from GPIO pins. Enable this input by setting the **Data Source** parameter to **DAC Bus**. If you select the **CPU or DMA** option instead, the bus connection disappears from the component symbol.

Use the data[7:0] input when hardware can set the proper value without CPU intervention. When using this option, you should enable **Strobe Mode** as well.

For many applications this input is not required, but instead the CPU or DMA will write a value directly to the data register. In firmware, you may use the IDAC_SetValue() function or directly write a value to the IDAC8_1_Data register (assuming an instance name of "IDAC8_1").

en – Input*

The en input is an UDB control input pin. Connecting this pin to logic '1' (ON) switches on the current to flow through the output terminal. Connecting to logic '0' (OFF) switches off the current at the output terminal. If the **Hardware Enable** check box is selected, this pin will be visible and must be connected to either logic '1' or logic '0'.

strobe – Input *

The strobe input is an optional signal input and is selected with the **Strobe Mode** parameter. If **Strobe Mode** is set to **External**, the pin is visible and must be connected to a valid digital source. In this mode, the data is transferred from the IDAC8 register to the DAC on the next positive edge of the strobe signal.

If this parameter is set to **Register Write** the pin disappears from the symbol and any write to the data registers is immediately transferred to the DAC.

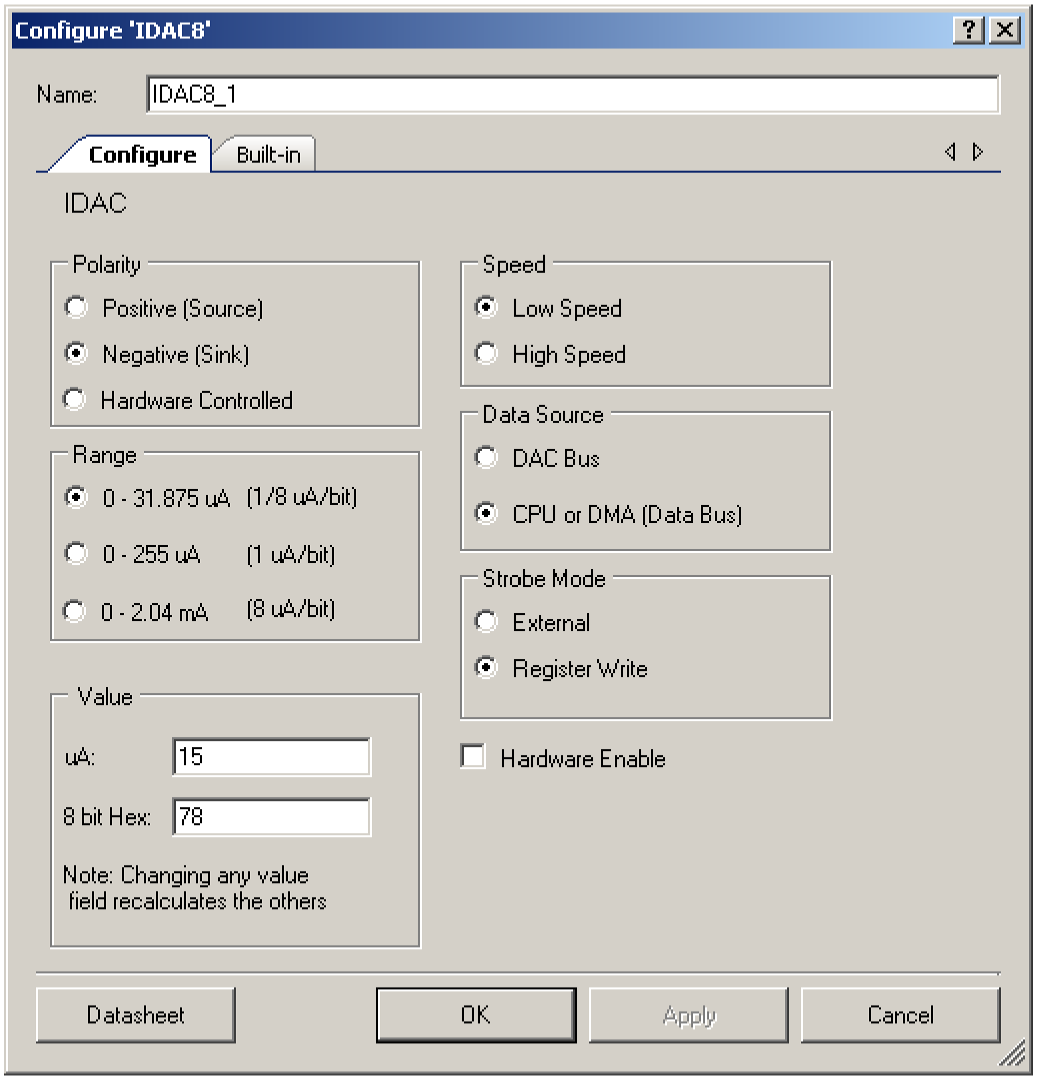
For audio or periodic sampling applications, the same clock used to clock the data into the DAC can also be used to generate an interrupt. In this case, each rising edge of the clock transfers data to the DAC and causes an interrupt to get the next value loaded into the DAC register.

Ipolarity – Input*

The ipolarity input is a UDB control input pin. This is used to control the direction of the current, either source or sink to its load. When this pin is connected to logic '0' (source), the output of the DAC sources current to a load that is connected to V_{SS} or other voltage that is at least 1.0 V below V_{DDA}. If the pin is connected to logic '1' (sink), it supplies current to a load that is connected to V_{DD} or other voltage at least 1.0 V above V_{SS}.

Component Parameters

Drag an IDAC8 component onto your design and double click it to open the **Configure** dialog.



Name: IDAC8_1

Polarity: ( ) Positive (Source) [✓ Negative (Sink)] ( ) Hardware Controlled

Range: [✓ 0 - 31.875 uA (1/8 uA/bit)] ( ) 0 - 255 uA (1 uA/bit) ( ) 0 - 2.04 mA (8 uA/bit)

Value: uA: 15 — 8 bit Hex: 78

Note: Changing any value field recalculates the others

Speed: [✓ Low Speed] ( ) High Speed

Data Source: ( ) DAC Bus [✓ CPU or DMA (Data Bus)]

Strobe Mode: ( ) External [✓ Register Write]

☐ Hardware Enable

The IDAC8 component provides the following parameters.

Polarity

The **Polarity** parameter allows you to select whether the IDAC8 sinks or sources current to its load. When the **Positive (Source)** option is selected, the output of the DAC sources current to a load that is connected to V_{SS} or other voltage that is at least 1.0 V below V_{DDA}. In the **Negative (Sink)** mode, it supplies current to a load that is connected to V_{DD} or other voltage at least 1.0 V above V_{SS}. Depending on which polarity is selected, the symbol shows the direction of the current.

The **Hardware Controlled** option in the **Polarity** parameter is used to control the direction of the current, either **Source** or **Sink** from UDB control. Logic '0' (source) as input specifies the current direction as **Source**. Logic ‘1’ (sink) as input specifies it as current **Sink**. When **Hardware Controlled** is selected then the “ipolarity” pin will be visible as an input.

Range

This parameter allows you to set one of three current ranges as the default value. The range may be changed at any time during run time with the IDAC8_SetRange() function. If you select the highest current range, **0 – 2040 uA**, you should route the output to one of the special pins that provide a low resistive path. These pins are P0[6], P0[7], P3[0], and P3[1].

Range	Lowest Value	Highest Value	Step Size
0 – 31.875 uA	0.0 µA	31.875 µA	0.125 µA
0 – 255 uA	0.0 µA	255 µA	1 µA
0 – 2040 uA	0.0 µA	2040 µA	8 µA

Value

This is the initial value the IDAC8 presents after the IDAC8_Start() command is executed. The IDAC_SetValue() function or a direct write to the DAC register overrides the default value at any time. Legal values are between 0 and FF, inclusive. The **uA** field represents IDAC8 source and sink current in microamps. **8 bit Hex** represents IDAC8 input data value in hexadecimal format.

Data Source

This parameter selects the source of the data to be written into the DAC register. If the CPU (firmware) or the DMA writes data to the IDAC8, then select **CPU or DMA (Data Bus)**. If data is written directly from the UDBs or a UDB-based component, then select **DAC Bus**. When **DAC Bus** is selected, the input is indicated on the IDAC8 symbol. There is only one DAC Bus, so multiple IDACs cannot have independent hardware (UDB) data sources. When **Data Source** is set as **DAC Bus**, the customizer automatically sets the **Strobe Mode** to **External** and disables the option so that you cannot change it.

Note For PSoC 5 silicon, a write of a new value to the DAC may result in an indeterminate value on the DAC output. To output the desired value, write or strobe the DAC twice with the same value. Because the first write may result in an indeterminate output, the time between the two writes should be minimized. This applies to writes by CPU, DMA, and strobe. The API IDAC8_SetValue() writes the value provided twice to mitigate this issue for CPU writes.

Speed

This parameter provides two settings for the designer, **Low Speed** and **High Speed**. In the Low Speed mode, the settling time is slower but it consumes less operating current. In the High Speed mode, the current settles much faster, but at a cost of more operating current.

Strobe Mode

This parameter selects whether the data is immediately written to the DAC as soon as the data is written into the IDAC8 data register. This mode is selected when the **Register Write** option is selected. When the **External** option is selected, a clock or signal from UDBs controls when the data is written from the DAC register to the actual DAC.

Hardware Enable

This parameter provides the UDB control to switch on or off the current flow at the output terminal. Logic '1' (ON) as input specifies that current flows through the output terminal. Logic '0' (OFF) as input specifies that current doesn't flow through the output terminal. When the **Hardware Enable** check box is selected, the "en" pin will be visible as an input.

Resources

	Digital Blocks					API Memory (Bytes)		
Analog Blocks	**Datapaths**	**Macro cells**	**Status Registers**	**Control Registers**	**Counter7**	**Flash**	**RAM**	**Pins (per External I/O)**
1 VIDAC fixed block	N/A	N/A	N/A	N/A	N/A	417	3	1

The IDAC8 uses one vIDAC8 analog block.

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 "IDAC8_1" to the first instance of a component in a given design. You can rename the instance 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 "IDAC8."

Function	Description
IDAC8_Start()	Initializes the IDAC8 with default customizer values. Enables and powers up the IDAC8.
IDAC8_Stop()	Disables the IDAC8 and sets it to the lowest power state.
IDAC8_SetSpeed()	Sets DAC speed.
IDAC8_SetPolarity()	Sets the output mode to current sink or source.

Function	Description
IDAC8_SetRange()	Sets full-scale range for IDAC8.
IDAC8_SetValue()	Sets value between 0 and 255 with the given range.
IDAC8_Sleep()	Stops and saves the user configuration.
IDAC8_WakeUp()	Restores and enables the user configuration.
IDAC8_Init()	Initializes or restores default IDAC8 configuration
IDAC8_Enable()	Enables the IDAC8.
IDAC8_SaveConfig()	Saves the current configuration
IDAC8_RestoreConfig()	Restores the configuration.

Global Variables

Variable	Description
IDAC8_initVar	Indicates whether the IDAC8 has been initialized. The variable is initialized to 0 and set to 1 the first time IDAC8_Start() is called. This allows the component to restart without reinitialization after the first call to the IDAC8_Start() routine. If reinitialization of the component is required, then the IDAC8_Init() function can be called before the IDAC8_Start() or IDAC8_Enable() function.

void IDAC8_Start(void)

Description: This is the preferred method to begin component operation. IDAC8_Start() sets the initVar variable, calls the IDAC8_Init() function, and then calls the IDAC8_Enable() function. Enables and powers up the IDAC8 to the given power level. A power level of 0 is the same as executing the stop function.

Parameters: None

Return Value: None

Side Effects: If the initVar variable is already set, this function only calls the IDAC8_Enable() function.

void IDAC8_Stop(void)

Description: Powers down IDAC8 to lowest power state and disables output.

Parameters: None

Return Value: None

Side Effects: None

void IDAC8_SetSpeed(uint8 speed)

Description: Sets DAC speed.

Parameters: uint8 speed: Sets DAC speed, see the following table for valid parameters.

Option	Description
IDAC8_LOWSPEED	Low speed (low power)
IDAC8_HIGHSPEED	High speed (high power)

Return Value: None

Side Effects: None

void IDAC8_SetPolarity(uint8 polarity)

Description: Sets output polarity to sink or source. This function is valid only if the Polarity parameter is set to either source or sink.

Parameters: uint8 polarity: Sets current sink or source functionality, see the following table.

Option	Description
IDAC8_SOURCE	Set mode as current source.
IDAC8_SINK	Set mode to current sink.

Return Value: None

Side Effects: None

void IDAC8_SetRange(uint8 range)

Description: Sets full-scale range for IDAC8

Parameters: uint8 range: Sets full-scale range for IDAC8. See the following table for ranges.

Option	Description
IDAC8_RANGE_32uA	Set full scale range to 31.875 µA
IDAC8_RANGE_255uA	Set full scale range to 255 µA
IDAC8_RANGE_2mA	Set full scale range to 2.040 mA

Return Value: None

Side Effects: None

void IDAC8_SetValue(uint8 value)

Description:	Sets value to output on IDAC8. Valid values are between 0 and 255.
Parameters:	uint8 value: Value between 0 and 255. A value of 0 is the lowest (zero) and a value of 255 is the full-scale value. The full-scale value depends on the range, which is selected with the IDAC8_SetRange() API.
Return Value:	None
Side Effects:	On PSoC 3 ES2, PSoC 3 Production, and PSoC 5, the IDAC8_SetValue() function should be called after enabling the power to the IDAC8.

void IDAC8_Sleep(void)

Description:	This is the preferred API to prepare the component for sleep. The IDAC8_Sleep() API saves the current component state. Then it calls the IDAC8_Stop() function and calls IDAC8_SaveConfig() to save the hardware configuration. Call the IDAC8_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 IDAC8_Wakeup(void)

Description:	This is the preferred API to restore the component to the state when IDAC8_Sleep() was called. The IDAC8_Wakeup() function calls the IDAC8_RestoreConfig() function to restore the configuration. If the component was enabled before the IDAC8_Sleep() function was called, the IDAC8_Wakeup() function will also re-enable the component.
Parameters:	None
Return Value:	None
Side Effects:	Calling the IDAC8_Wakeup() function without first calling the IDAC8_Sleep() or IDAC8_SaveConfig() function may produce unexpected behavior.

void IDAC8_Init(void)

Description:	Initializes or restores the component according to the customizer Configure dialog settings. It is not necessary to call IDAC8_Init() because the IDAC8_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. This will reinitialize the component. Calling the IDAC8_Init() function requires a call to IDAC8_SetValue() if you intend to set a new value other than what is currently in the register.

void IDAC8_Enable(void)

Description:	Activates the hardware and begins component operation. It is not necessary to call IDAC8_Enable() because the IDAC8_Start() API calls this function, which is the preferred method to begin component operation.
Parameters:	None
Return Value:	None
Side Effects:	None

void IDAC8_SaveConfig(void)

Description:	This function saves the component configuration and nonretention registers. It also saves the current component parameter values, as defined in the Configure dialog or as modified by appropriate APIs. This function is called by the IDAC8_Sleep() function.
Parameters:	None
Return Value:	None
Side Effects:	None

void IDAC8_RestoreConfig(void)

Description:	This function restores the component configuration and nonretention registers. This function also restores the component parameter values to what they were before calling the IDAC8_Sleep() function.
Parameters:	None
Return Value:	None
Side Effects:	Calling this function without first calling the IDAC8_Sleep() or IDAC8_SaveConfig() function may produce unexpected behavior.

Sample Firmware Source Code

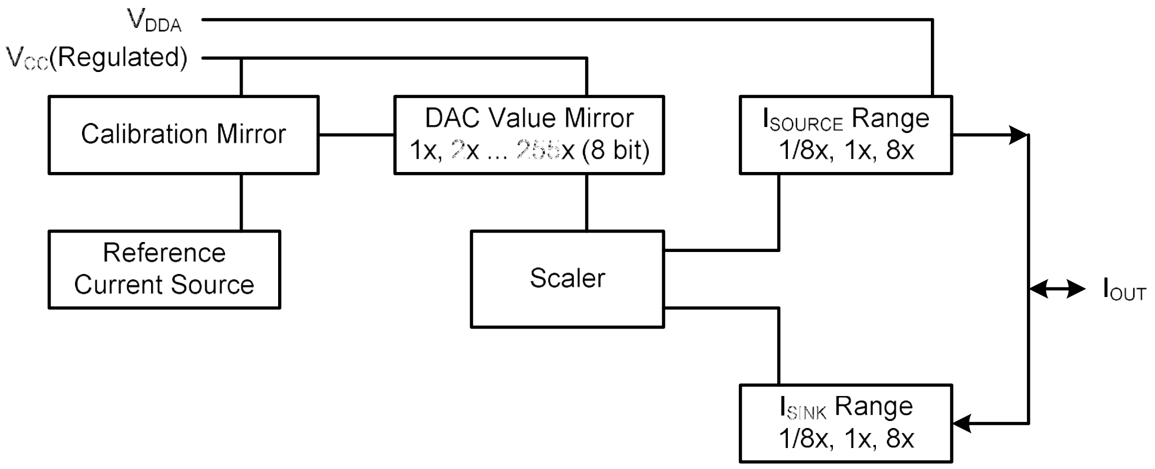
PSoC Creator provides many example projects that include schematics and example code in the Find Example Project dialog. For component-specific examples, open the dialog from the Component Catalog or an instance of the component in a schematic. For general examples, open the dialog from the Start Page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

IDAC8 functionality is implemented using the PSoC vidac block. This block is an 8 bit digital analog converter capable of either voltage or current output. The output from the IDAC8 is single-ended. The functional block diagram is shown in Figure 1.

Figure 1. Block Diagram



IDAC8 can be used as a current source or sink. It is built using current mirror architecture; current is mirrored from a reference source to a mirror IDAC8. Calibration and value current mirrors are responsible for the 8-bit calibration and the 8-bit IDAC8 value. The current is then diverted into the scaler to generate the current corresponding to the IDAC8 value. The IDAC8 value can either be obtained from the IDAC8 data register or from eight lines from the UDB. The IDAC8 can convert up to 8 Msps to generate sinusoids.

The two current mirrors provide either a current sink or source. The IDAC8 can be configured to operate in one of three ranges:

- 0 to 2.040 mA, 8 µA/bit
- 0 to 255 µA, 1 µA/bit
- 0 to 31.875 µA, 0.125 µA/bit

For each level, there are 255 equal steps of M/256 where M = 2.040 mA, 255 µA, or 31.875 µA. The output can be delivered into any resistance or to a fixed voltage, as long as the minimum headroom requirement of 1.0 V is met. This means that the maximum voltage for sourced current is V_{DDA} – 1.0 V and the minimum output voltage for sunk current is 1.0 V above V_{SSA}.

The IDAC8 is strobed to get its output to change for the input code. You can select the strobe sources for the IDAC8 from the bus write strobe, analog clock strobe, or any UDB signal strobe.

DMA

IDAC8 components do not require implementation of a DMA Request signal. The data rate to IDAC8 components should be controlled externally. The DMA Wizard can be used to configure DMA operation as follows:

Name of DMA Source/Destination in DMA Wizard	Direction	DMA Req Signal	DMA Req Type	Description
IDAC8_Data_PTR	Destination	N/A	N/A	Stores the DAC value between 0 to 255

Registers

The functions provided support most of the common run-time functions that are required for most applications. The following register reference provides a brief description for the advanced user. The IDAC8_Data register may be used to write data directly to the DAC without using the API. This may be useful for either the CPU or DMA.

IDAC8_CR0

Bits	7	6	5	4	3	2	1	0
Value	reserved			mode	range[1:0]		hs	reserved

- mode: Sets DAC to either voltage or current mode
- range[1:0]: DAC range settings
- hs: Sets data speed

IDAC8_CR1

Bits	7	6	5	4	3	2	1	0
Value	reserved		mx_data	reset_udb_en	mx_idir	idirbit	Mx_ioff	ioffbit

- mx_data: Selects data source
- reset_udb_en: DAC reset enable

- mx_idir: Mux selection for DAC current direction control
- idirbit: Register source for DAC current direction
- mx_off: Mux selection for DAC current off control
- ioffbit: Register source for DAC current off

IDAC8_DATA

Bits	7	6	5	4	3	2	1	0
Value	Data[7:0]							

- Data[7:0]: DAC data register

DC and AC Electrical Characteristics for PSoC 3

Unless otherwise specified: Typical = 25 °C, V_{DDA} = 5.0 V, headroom = 1.0 V minimum, specifications apply to all ranges: 0 to 31.875 µA, 0 to 255 µA, 0 to 2.04 mA.

IDAC8 DC Characteristics

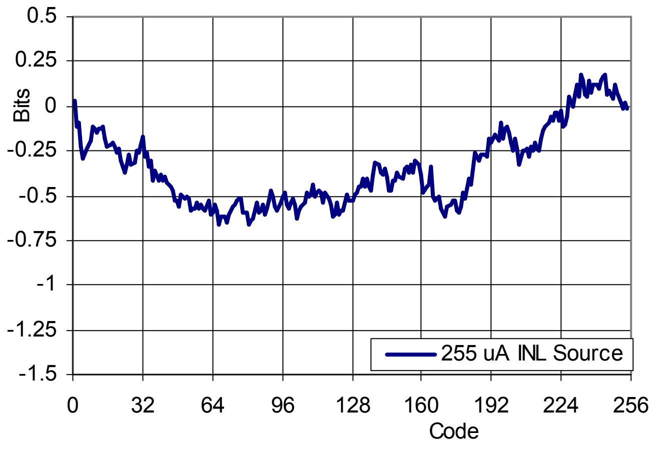
Parameter	Description	Conditions	Min	Typ	Max	Units
	Resolution		–	–	8	bits
I_{OUT}	Output current at code = 255	Range = 2.040 mA, code = 255, $V_{DDA} \geq 2.7$ V, R_{LOAD} = 600 Ω	–	2.040	–	mA
		Range = 2.040 mA, High mode, code = 255, $V_{DDA} \leq 2.7$ V, R_{LOAD} = 300 Ω	–	2.040	–	mA
		Range = 255 µA, code = 255, R_{LOAD} = 600 Ω	–	255	–	µA
		Range = 31.875 µA, code = 255, R_{LOAD} = 600 Ω	–	31.875	–	µA
	Monotonicity		–	–	Yes	
Ezs	Zero scale error		–	0	±1	LSB
Eg	Gain error	Range = 2.04 mA, 25 °C	–	–	±2.5	%
		Range = 255 µA, 25 °C	–	–	±2.5	%
		Range = 31.875 µA, 25 °C	–	–	±3.5	%

Parameter	Description	Conditions	Min	Typ	Max	Units
TC_Eg	Temperature coefficient of gain error	Range = 2.04 mA	–	–	0.04	%/°C
		Range = 255 µA	–	–	0.04	%/°C
		Range = 31.875 µA	–	–	0.05	%/°C
INL	Integral nonlinearity	Sink mode, range = 255 µA, Codes 8 to 255, R_{LOAD} = 2.4 kΩ, C_{LOAD} = 15 pF	–	±0.9	±1	LSB
		Source mode, range = 255 µA, Codes 8 – 255, R_{LOAD} = 2.4 kΩ, C_{LOAD} = 15 pF	–	±1.2	±1.5	LSB
DNL	Differential nonlinearity	Sink mode, range = 255 µA, R_{LOAD} = 2.4 kΩ, C_{LOAD} = 15 pF	–	±0.3	±1	LSB
		Source mode, range = 255 µA, R_{LOAD} = 2.4 kΩ, C_{LOAD} = 15 pF	–	±0.3	±1	LSB
Vcompliance	Dropout voltage, source or sink mode	Voltage headroom at max current, R_{LOAD} to V_{DDA} or R_{LOAD} to V_{SSA}, V_{DIFF} from V_{DDA}	1	–	–	V
Idev	Voltage-dependent current deviation	Source mode, V_{OUT} = 0.0 V Sink mode, V_{OUT} = V_{DD}	–	1.0%	–	µA
I_{DD}	Operating current, code = 0	Slow mode, source mode, range = 31.875 µA	–	44	100	µA
		Slow mode, source mode, range = 255 µA,	–	33	100	µA
		Slow mode, source mode, range = 2.04 mA	–	33	100	µA
		Slow mode, sink mode, range = 31.875 µA	–	36	100	µA
		Slow mode, sink mode, range = 255 µA	–	33	100	µA
		Slow mode, sink mode, range = 2.04 mA	–	33	100	µA
		Fast mode, source mode, range = 31.875 µA	–	310	500	µA
		Fast mode, source mode, range = 255 µA	–	305	500	µA
		Fast mode, source mode, range = 2.04 mA	–	305	500	µA
		Fast mode, sink mode, range = 31.875 µA	–	310	500	µA

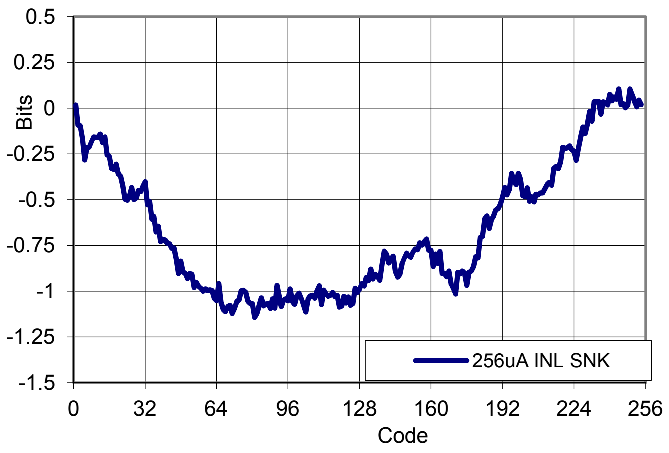
Parameter	Description	Conditions	Min	Typ	Max	Units
		Fast mode, sink mode, range = 255 µA	–	300	500	µA
		Fast mode, sink mode, range = 2.04 mA	–	300	500	µA

Figures

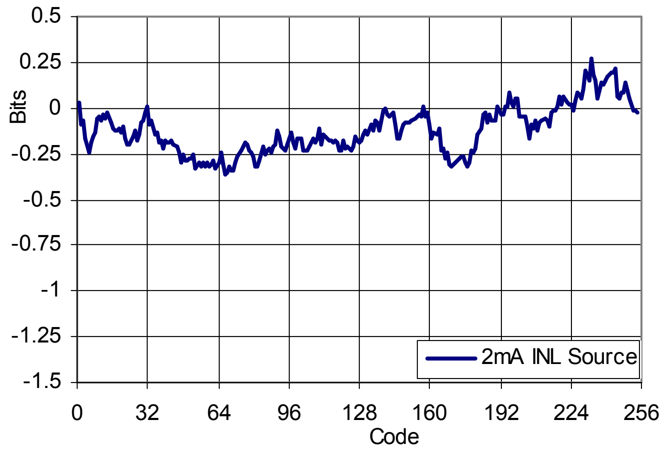
INL versus DAC Code, Range = 255 µA, Source Mode



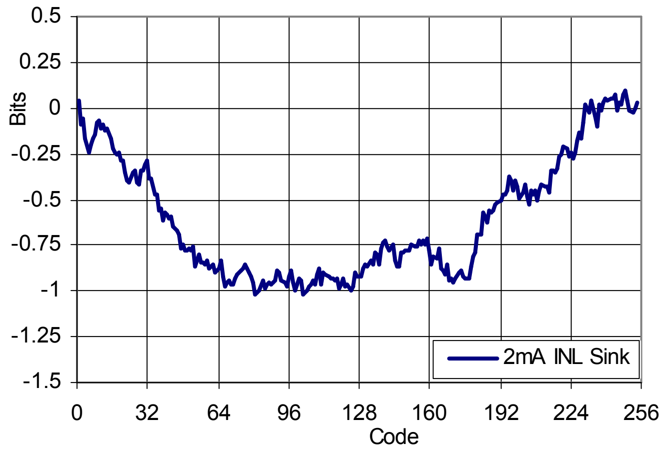
INL versus DAC Code, Range = 255 µA, Sink Mode



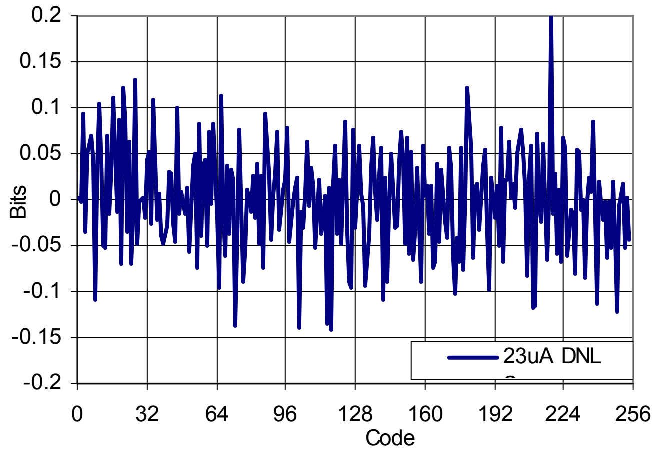
INL versus DAC Code, Range = 2.040 mA, Source Mode



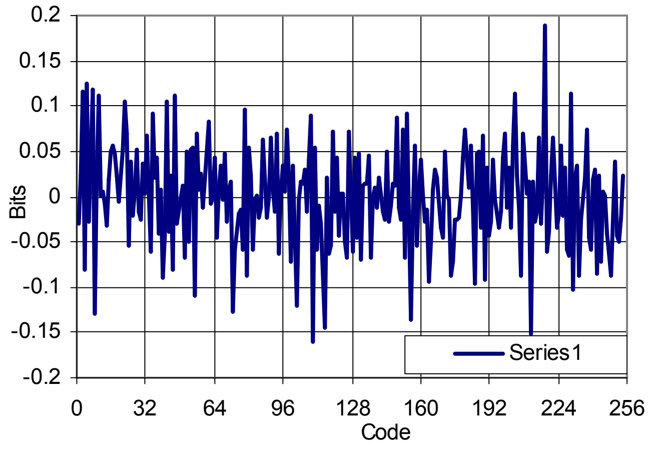
INL versus DAC Code, Range = 2.040 mA, Sink Mode



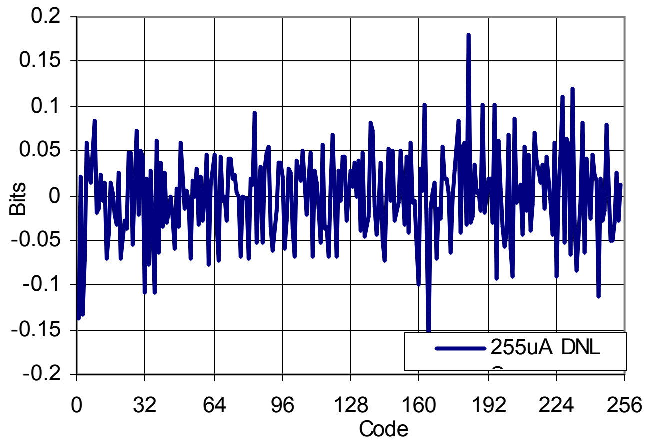
DNL versus DAC Code, Range = 31.875 µA, Source Mode



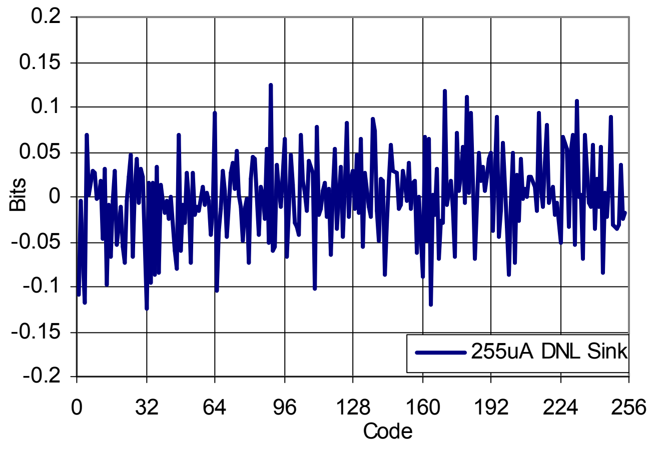
DNL versus DAC Code, Range = 31.875 µA, Sink Mode



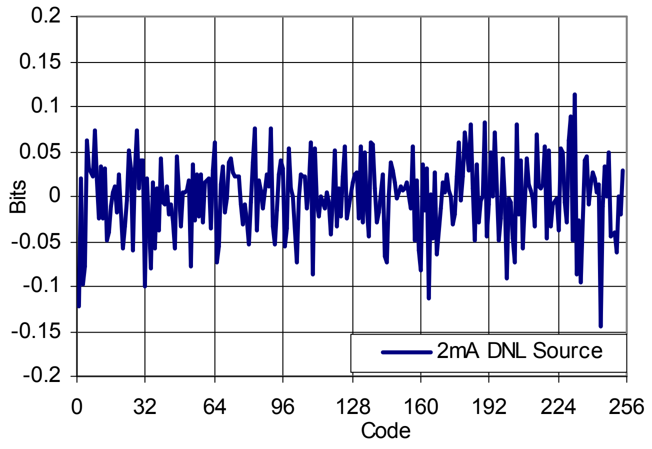
DNL versus DAC Code, Range = 255 µA, Source Mode



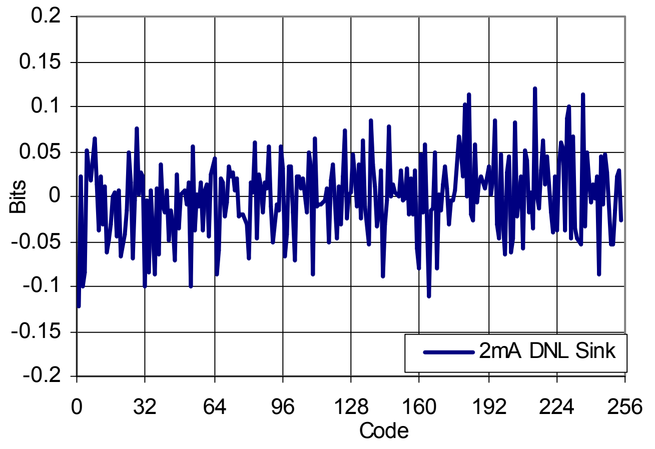
DNL versus DAC Code, Range = 255 µA, Sink Mode



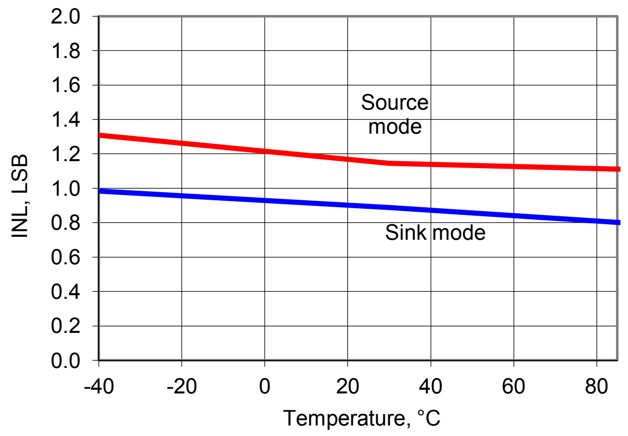
DNL versus DAC Code, Range = 2.04 mA, Source Mode



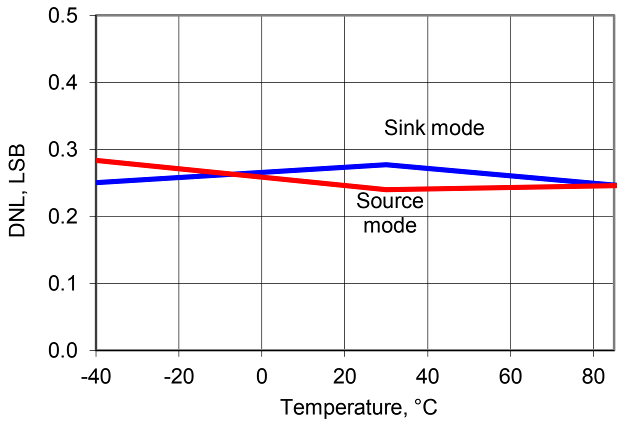
DNL versus DAC Code, Range = 2.04 mA, Sink Mode



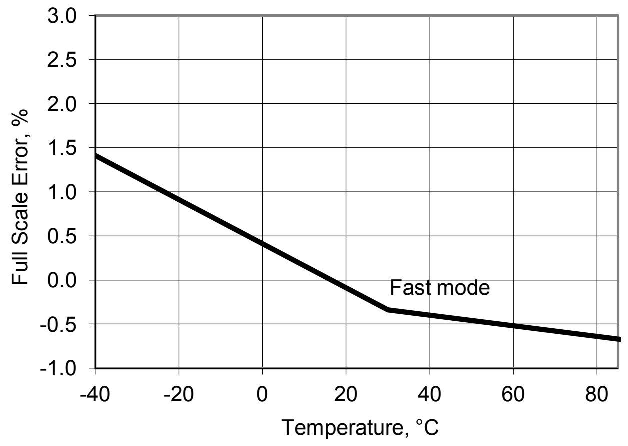
IDAC INL versus Temperature, Range = 255 µA, Fast Mode



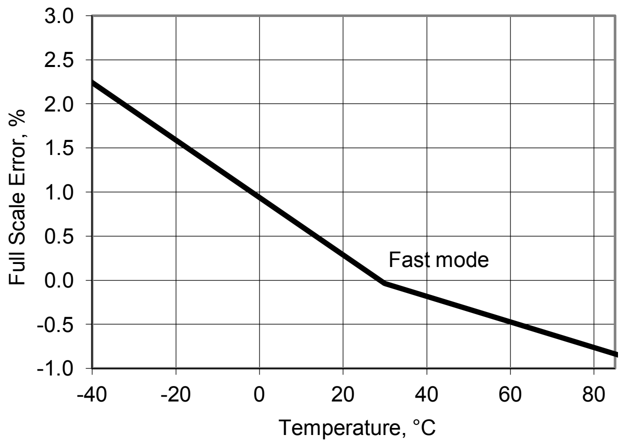
IDAC DNL versus Temperature, Range = 255 µA, Fast Mode



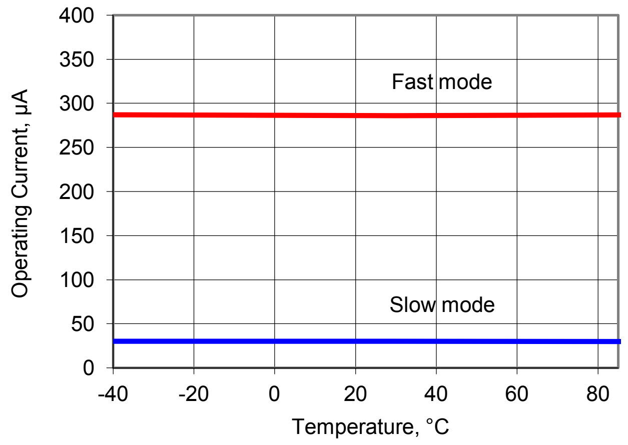
IDAC Full Scale Error versus Temperature, Range = 255 µA, Source Mode



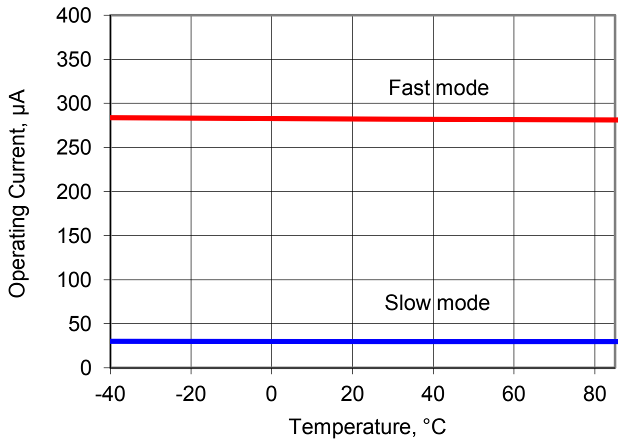
IDAC Full Scale Error versus Temperature, Range = 255 µA, Sink Mode



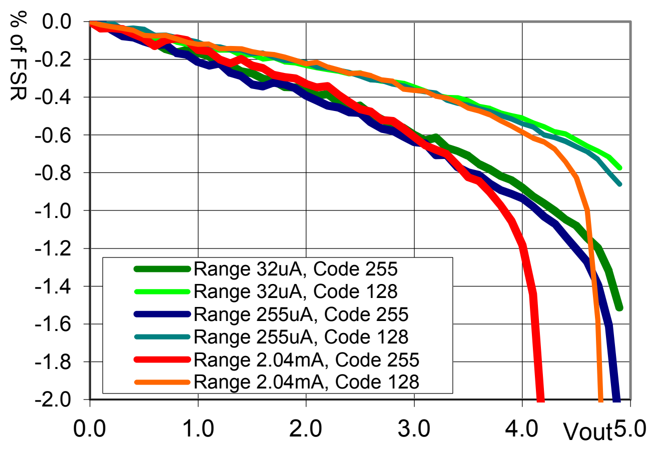
IDAC Operating Current versus Temperature, Range = 255 µA, Code = 0, Source Mode



IDAC Operating Current versus Temperature, Range = 255 µA, Code = 0, Sink Mode



Current Difference from Nominal into Vout (% of FSR), Vdd = 5.5 V, T = 25 °C

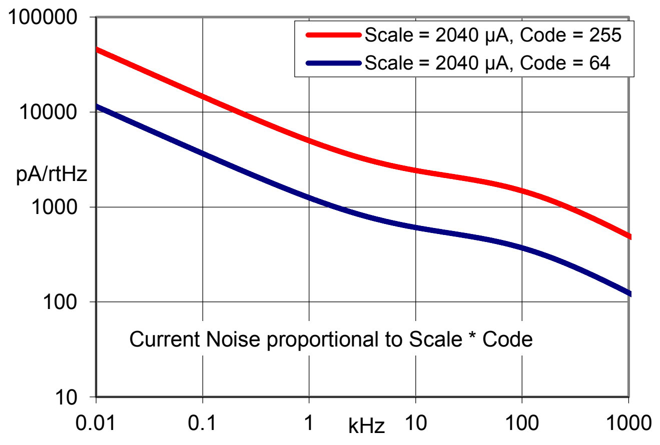


IDAC8 AC Characteristics

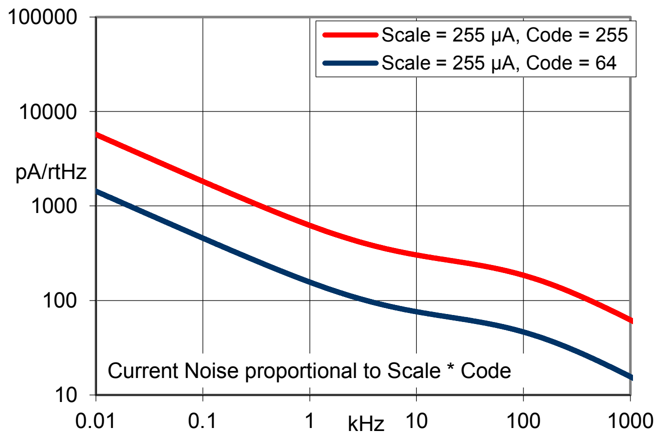
Parameter	Description	Conditions	Min	Typ	Max	Units
F_{DAC}	Update rate		–	–	8	Msps
T_{SETTLE}	Settling time to 0.5 LSB	Independent of IDAC range setting (I_{OUT}), full-scale transition, 600-Ω load, C_L = 15 pF, Fast mode	–	–	125	ns
		Independent of IDAC range setting (I_{OUT}), full-scale transition, 600-Ω load, C_L = 15 pF, Slow mode	–	–	1000	ns
In2040 µA	Current Noise	Fast mode, source mode, range = 2040 µA, code = 255, V_{DDA} = 5 V, 10 kHz	–	2.7	–	nA/rtHz
In255 µA		Fast mode, source mode, range = 255 µA, code = 255, V_{DDA} = 5 V, 10 kHz	–	340	–	pA/rtHz
In32 µA		Fast mode, source mode, range = 31.875 µA, code = 255, V_{DDA} = 5 V, 10 kHz	–	40	–	pA/rtHz

Figures

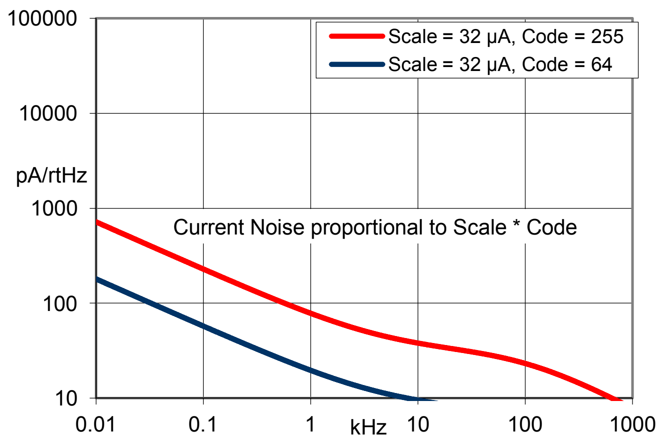
Noise versus Frequency, 2040 µA



Noise versus Frequency, 255 µA



Noise versus Frequency, 32 µA



DC and AC Electrical Characteristics for PSoC 5

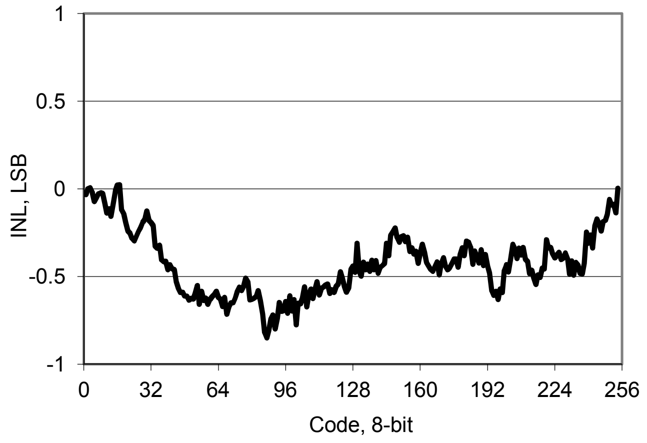
Unless otherwise specified: Typical = 25 °C, V_{DDA} = 5.0 V, headroom = 1.0 V minimum, specifications apply to all ranges: 0 to 31.875 µA, 0 to 255 µA, 0 to 2.04 mA.

Parameter	Description	Conditions	Min	Typ	Max	Units
	Resolution		–	–	8	bits
I_{OUT}	Output current at code = 255	Range = 2.040 mA, code = 255, R_{LOAD} = 600 Ω	–	2.040	–	mA
		Range = 255 µA, code = 255, R_{LOAD} = 600 Ω	–	255	–	µA
		Range = 31.875 µA, code = 255, R_{LOAD} = 600 Ω	–	31.875	–	µA
	Monotonicity		–	–	Yes	
Ezs	Zero scale error		–	0	±2.5	LSB
Eg	Gain error		–	–	±5	%
TC_Eg	Temperature coefficient of gain error	Range = 2.04 mA	–	–	0.04	%/°C
		Range = 255 µA	–	–	0.04	%/°C
		Range = 31.875 µA	–	–	0.05	%/°C
INL	Integral nonlinearity	Range = 255 µA, Codes 8 to 255, R_{LOAD} = 600Ω, C_{LOAD} = 15 pF	–	–	±3	LSB
DNL	Differential nonlinearity	Range = 255 µA, R_{LOAD} = 600Ω, C_{LOAD} = 15 pF	–	–	±1.6	LSB
Vcompliance	Dropout voltage, source or sink mode	Voltage headroom at max current, R_{LOAD} to V_{DDA} or R_{LOAD} to V_{SSA}, V_{DIFF} from V_{DDA}	1	–	–	V
I_{DD}	Operating current, code = 0	Slow mode, source mode, range = 31.875 µA	–	44	100	µA
		Slow mode, source mode, range = 255 µA,	–	33	100	µA
		Slow mode, source mode, range = 2.04 mA	–	33	100	µA
		Slow mode, sink mode, range = 31.875 µA	–	36	100	µA
		Slow mode, sink mode, range = 255 µA	–	33	100	µA
		Slow mode, sink mode, range = 2.04 mA	–	33	100	µA

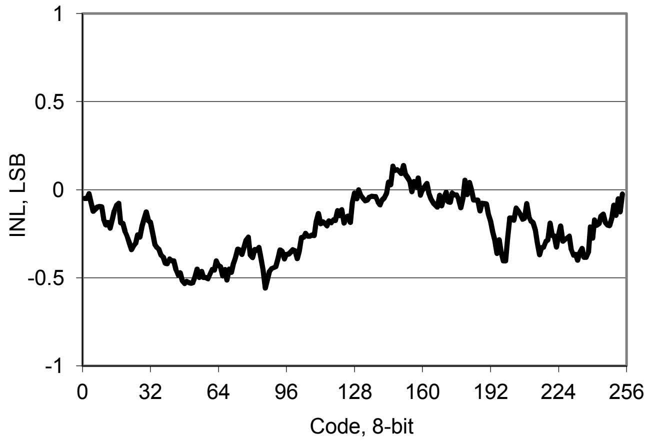
Parameter	Description	Conditions	Min	Typ	Max	Units
		Fast mode, source mode, range = 31.875 µA	–	310	500	µA
		Fast mode, source mode, range = 255 µA	–	305	500	µA
		Fast mode, source mode, range = 2.04 mA	–	305	500	µA
		Fast mode, sink mode, range = 31.875 µA	–	310	500	µA
		Fast mode, sink mode, range = 255 µA	–	300	500	µA
		Fast mode, sink mode, range = 2.04 mA	–	300	500	µA

Figures

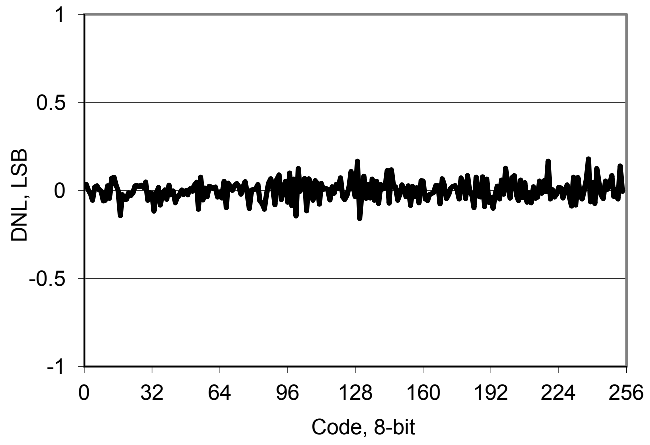
INL versus DAC Code, Range = 255 µA, Source Mode



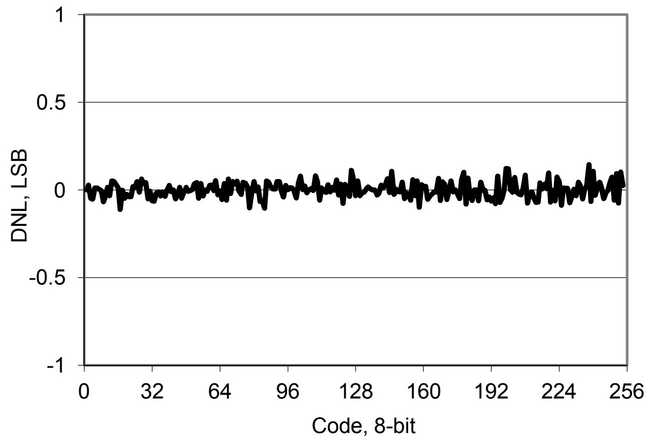
INL versus DAC Code, Range = 255 µA, Sink Mode



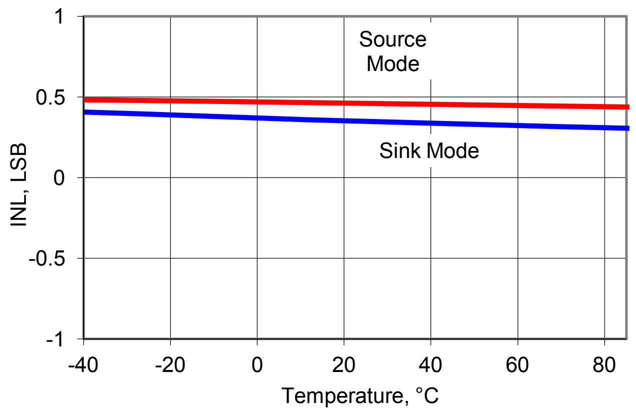
DNL versus DAC Code, Range = 255 µA, Source Mode



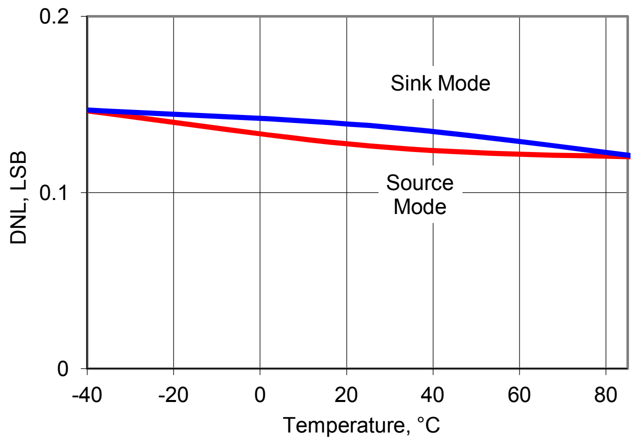
DNL versus DAC Code, Range = 255 µA, Sink Mode



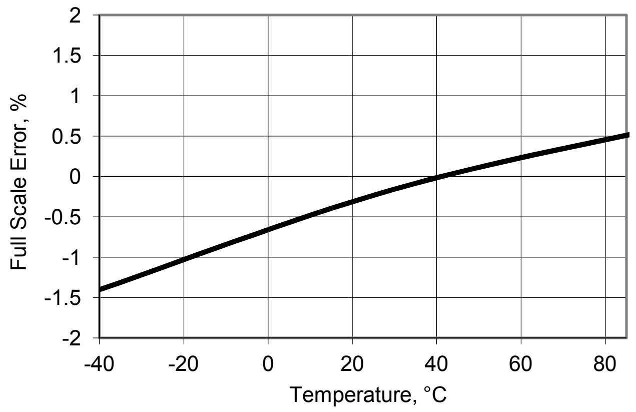
IDAC INL versus Temperature, Range = 255 µA, Fast Mode



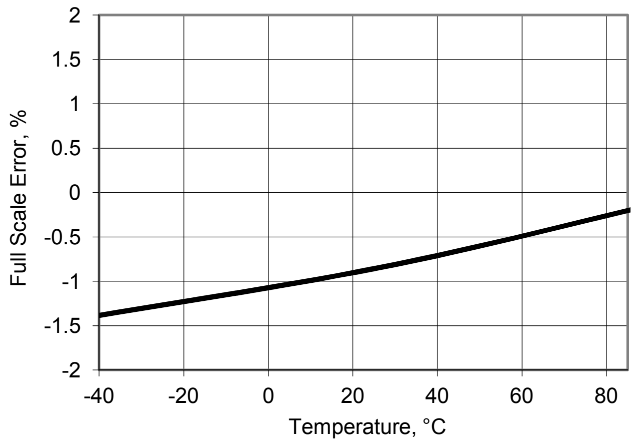
IDAC DNL versus Temperature, Range = 255 µA, Fast Mode



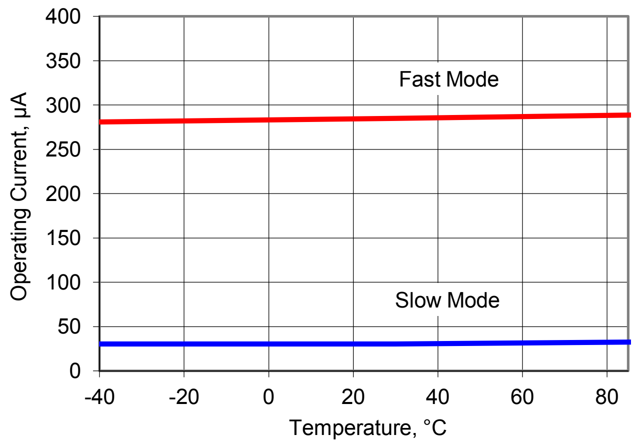
IDAC Full Scale Error versus Temperature, Range = 255 µA, Source Mode



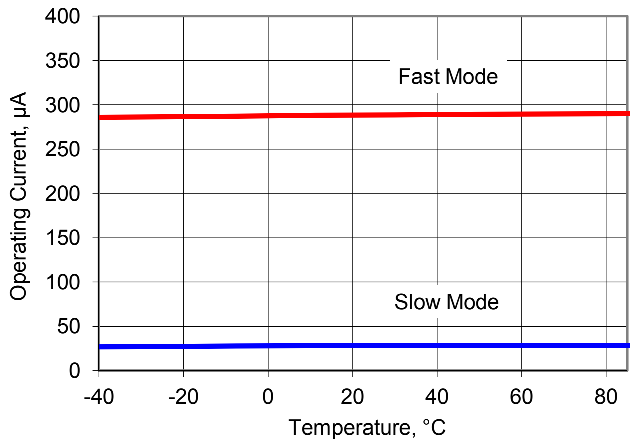
IDAC Full Scale Error versus Temperature, Range = 255 µA, Sink Mode



IDAC Operating Current versus Temperature, Range = 255 µA, Code = 0, Source Mode



IDAC Operating Current versus Temperature, Range = 255 µA, Code = 0, Sink Mode



IDAC8 AC Characteristics

Parameter	Description	Conditions	Min	Typ	Max	Units
F_{DAC}	Update rate		–	–	5.5	Msps
T_{SETTLE}	Settling time to 0.5 LSB	Range = 31.875 µA or 255 µA, full scale transition, fast mode, 600 Ω 15-pF load	–	–	180	ns
In255 µA	Current Noise	Fast mode, source mode, range = 255 µA, code = 255, V_{DDA} = 5 V, 10 kHz	–	340	–	pA/sqrtHz

Component Changes

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

	Description of Changes	Reason for Changes / Impact
1.80	Updated IDAC8 customizer GUI, so that Hardware Controlled and Hardware Enable options are provided.	Allows the user to control the current direction (source or sink) and current flow (switch ON or OFF) through UDB control.
	Added PSoC5 DC and AC Electrical Characteristics specifications to datasheet	
	Minor datasheet edits and updates	
1.70	IDAC8_Stop() API changes for PSoC 5	Change required to prevent the component from impacting unrelated analog signals when stopped, when used with PSoC 5.
	Updated IDAC8 customizer GUI	▪ To allow user to enter float values in uA field. ▪ To force the Strobe mode to External when Data Source is selected as DAC Bus. ▪ To make IDAC8 GUI consistent with VDAC8 GUI.
1.60	Added a GUI Configuration Editor	Previous configuration window did not provide enough information for ease of use.
	Added characterization data to 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.
	Added DMA capabilities file to the component.	This file allows the IDAC8 to be supported by the DMA Wizard tool in PSoC Creator.