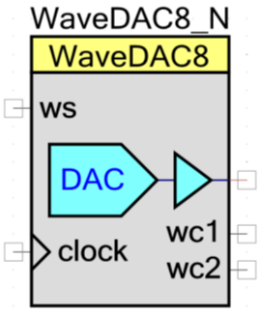
Please note that Cypress is an Infineon Technologies Company.
The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the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

8-Bit Waveform Generator (WaveDAC8)

2.10



Features

- Supports standard and arbitrary waveform generation
- Arbitrary waveform may be drawn manually or imported from file
- Output may be voltage or current, sink or source
- Voltage output can be buffered or direct from DAC
- Hardware selection between two waveforms
- Waveforms may be up to 4000 points
- Predefined sine, triangle, square, and sawtooth waveforms

General Description

The WaveDAC8 component provides a simple and fast solution for automatic periodic waveform generation. A high-level interface allows you to select a predefined waveform or a custom arbitrary waveform. Two separate waveforms can be defined then selected with an external pin to create a modulated output. The input clock can also be used to change the sample rate or modulate the output.

When to use a WaveDAC8

Use the WaveDAC8 anytime a periodic waveform needs to be generated.

Input/Output Connections

This section describes the various input and output connections for the WaveDAC8. 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.

Wave – analog output (the terminal label is hidden)

The Wave terminal is connected directly to the DAC's output, except when the buffered range is chosen, then the WaveDAC's output is buffered. It may be routed to any analog compatible pin on the PSoC.

ws – Input

The Wave Select (ws) input selects which waveform will be generated. It can be used to switch quickly between two waveforms to generate an FSK signal.

clock – Input*

The clock input allows you to use an alternate clock source. When internal clock is selected, this input is not visible.

wc1 – Output

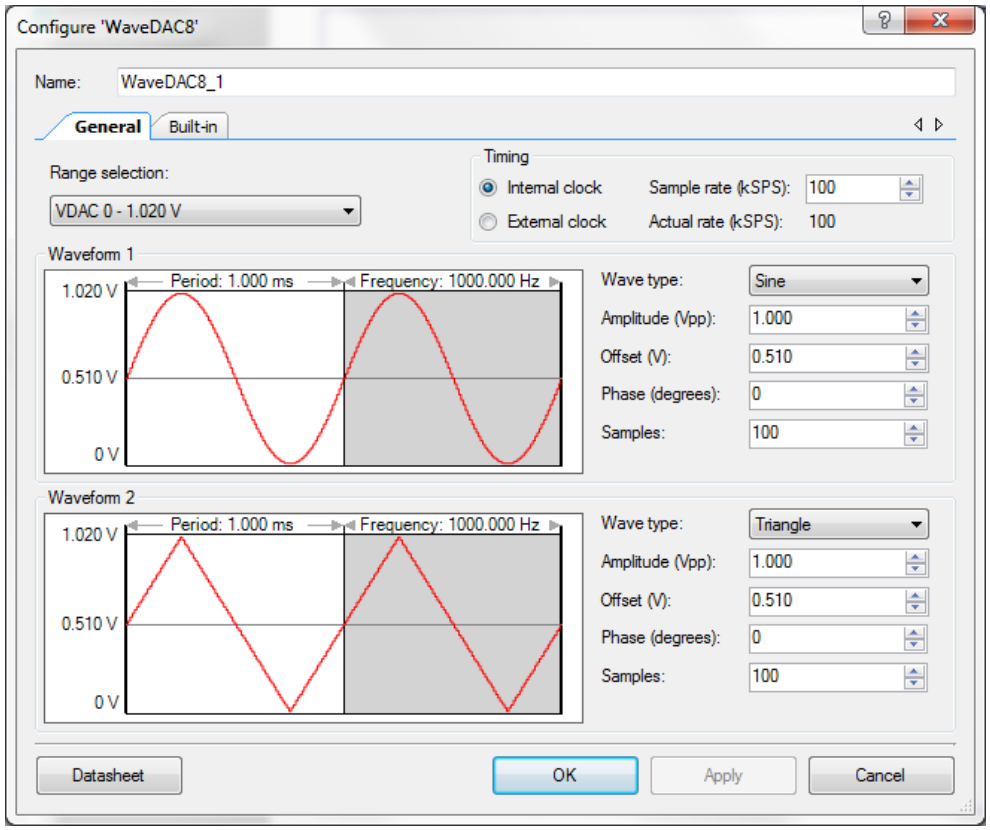
The Wave Complete 1 (wc1) signal goes high for two bus clocks at the end of waveform 1.

wc2 – Output

The Wave Complete 2 (wc2) signal goes high for two bus clocks at the end of waveform 2.

Parameters and Setup

Drag a WaveDAC8 component onto your design and double-click it to open the Configure dialog.



The WaveDAC8 component provides the following parameters.

Range Selection

This parameter selects the output mode and range of the internal DAC.

Range	Mode	Ouput	Step Size
VDAC 0 – 1.020 V	Voltage	0 to 1.020 Volts	4 mV
VDAC 0 – 1.020 V (Buffered)	Voltage	0 to 1.020 Volts	4 mV
VDAC 0 – 4.080 V	Voltage	0 to 4.080 volts	16 mV
VDAC 0 – 4.080 V (Buffered)	Voltage	0 to 4.080 volts	16 mV
ISink 0 – 2.040 mA	Current Sink	0 to 2.040 mA	8 uA
ISink 0 – 255 uA	Current Sink	0 to 255 uA	1 uA
ISink 0 – 32 uA	Current Sink	0 to 32 uA	0.125 uA
ISource 0 – 2.040 mA	Current Source	0 to 2.040 mA	8 uA
ISource 0 – 255 uA	Current Source	0 to 255 uA	1 uA
ISource 0 – 32 uA	Current Source	0 to 32 uA	0.125 uA

Timing

Clock Source (Internal / External)

Use this parameter to select whether the clock source is internal or external. When an internal clock is selected, the clock pin will not be visible.

Sample Rate (kSPS)

Use this parameter to select the sampling frequency rate in kSPS. The maximum sample rate is 6 MSPS for Current Mode, 1 MSPS for Voltage Mode 1 V ranges, and 250 kSPS for Voltage Mode 4 V ranges. The Waveform Period and Frequency can be calculated as follows:

$$Waveform_Period = \frac{Samples}{SampleRate}, \quad Waveform_Frequency = \frac{SampleRate}{Samples}$$

Note The WaveDAC8 component uses DMA to transfer data from a lookup table in Flash memory to the DAC. The DMA channel in the WaveDAC8 shares the bus with other DMA channels and the CPU. Each sample requires at least 10 bus clock cycles to transfer the data from Flash to the DAC. Make sure that the bus clock is at least 10 or more times faster than the WaveDAC8 sample rate. The bus clock can be set in the **Clocks** tab of the Design-Wide Resources (DWR) (*<project>.cydwr*) file in PSoC Creator.

For sample rates greater than 4 Msps, it is recommended to copy the waveform data into SRAM, and use the WaveDAC8_Wave1Setup() or WaveDAC8_Wave2Setup() API to configure the

WaveDAC8 to use a SRAM-based lookup table. This will eliminate any wait state delay incurred by reading Flash.

If using multiple WaveDAC8s in a single design, make sure that the sum of all sample rates of all the WaveDAC8s is at least 10 to 15 times the bus frequency. Also, evaluate any other DMA channels and take into account their utilization as well.

Actual Rate (kSPS)

This read-only field displays the actual calculated sample frequency rate. This may vary from the requested sample rate depending on the ability to create the clock based on the integer division of a higher frequency clock in the system.

Waveform 1 & 2

Both waveforms have identical parameters:

Wave Type

This parameter selects one of six waveforms, four are fixed and two allow the user to provide a custom waveform:

- Sine
- Square
- Triangle
- Sawtooth
- Arbitrary (Draw)
- Arbitrary (From File)

  Note The format of the wave source file is .csv (comma separated values). This is a simple text file that contains integer values that range from 0 to 255, separated by commas. For example: “0,1,2, … ,254,255”. The data is loaded from the file by the component when you click “open” *.csv file. It is not reloaded on clean/build project, etc.

Amplitude

This parameter defines the peak-to-peak amplitude for the non-arbitrary waveforms.

Offset

This parameter defines the offset of the middle for the non-arbitrary waveforms relative to zero level (0V or 0mA).

Phase

This parameter defines the phase shift (in degrees) of the waveform relative to the generation start point.

Samples

This parameter defines the number of waveform data samples.

Resources

The WaveDAC8 uses one viDAC8 block, digital demux, DFF trigger, two DMA channels, and an optional clock and/or opamp:

Configuration	Resource Type				
	UDB Macrocells	DMA Channels	VIDAC Fixed Blocks	Opamp Fixed Blocks	Digital Clock Dividers
Internal clock, VDAC mode not buffered	3	2	1	0	1
Internal clock, VDAC mode buffered	3	2	1	1	1
Internal clock, IDAC mode	3	2	1	0	1
External clock, VDAC mode not buffered	3	2	1	0	0
External clock, VDAC mode buffered	3	2	1	1	0
External clock, IDAC mode	3	2	1	0	0

The Flash/RAM usage is:

Configuration	PSoC 3 (Keil_PK51)		PSoC 5LP (GCC)	
	Flash Bytes	SRAM Bytes	Flash Bytes	SRAM Bytes
Internal clock, VDAC not buffered	851	11	850	10
Internal clock, VDAC buffered	902	12	932	10
Internal clock, IDAC	884	11	914	10
External clock, VDAC not buffered	815	11	794	10
External clock, VDAC buffered	866	12	876	10
External clock, IDAC	848	11	858	10

The listed flash sizes do not include waveform data arrays (2*100 bytes by default).

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 "WaveDAC8_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 "WaveDAC8".

Functions

Function	Description
void WaveDAC8_Start(void)	Starts the DAC and DMA channels.
void WaveDAC8_Stop(void)	Disables DAC and DMA channels.
void WaveDAC8_Init(void)	Initializes or restores the component according to the customizer Configure dialog settings.
void WaveDAC8_Enable(void)	Activates the hardware and begins component operation.
void WaveDAC8_Wave1Setup(uint8 * wavePtr, uint16 sampleSize)	Sets the array and size of array used for waveform generation for waveform 1.
void WaveDAC8_Wave2Setup(uint8 * wavePtr, uint16 sampleSize)	Sets the array and size of array used for waveform generation for waveform 2.
void WaveDAC8_StartEx(uint8 * wavePtr1, uint16 sampleSize1, uint8 * wavePtr2, uint16 sampleSize2)	Sets the arrays and sizes of arrays used for waveform generation for both waveforms and starts the DAC and DMA channels.
void WaveDAC8_SetSpeed(uint8 speed)	Set drive mode / speed of the DAC.
void WaveDAC8_SetRange(uint8 range)	Set current or voltage range.
void WaveDAC8_SetValue(uint8 value)	Set 8-bit DAC value.
void WaveDAC8_DacTrim(void)	Set the trim value for the given range.
void WaveDAC8_Sleep(void)	Stops and saves the user configuration.
void WaveDAC8_Wakeup(void)	Restores and enables the user configuration.
void WaveDAC8_SaveConfig(void)	This function saves the component configuration. This function will also save the current component parameter values, as defined in the Configure dialog or as modified by appropriate APIs. This function is called by the WaveDAC8_Sleep() function.
void WaveDAC8_RestoreConfig(void)	This function restores the component configuration. This function will also restore the component parameter values to what they were before calling the WaveDAC8_Sleep() function.

void WaveDAC8_Start(void)

Description: Performs all of the required initialization for the component and enables power to the block. The first time the routine is executed, the range, polarity (if any), and power (speed) settings are configured for the operating mode selected in the design. When called to restart the WaveDAC8 following a WaveDAC8_Stop() call, the current component parameter settings are retained.

When the external clock is used – this function should be called before the clock is started to cause correct waveform generation. Otherwise the first sample may be undefined.

Parameters: None

Return Value: None

Side Effects: None

void WaveDAC8_Stop(void)

Description: Turn off the WaveDAC8 block.

Parameters: None

Return Value: None

Side Effects: Does not affect WaveDAC8 type or power settings

void WaveDAC8_Wave1Setup(uint8 *WavePtr, uint16 SampleSize)

Description: Selects a new waveform array for the waveform 1 output. The WaveDAC8_Stop function should be called prior to calling this function and WaveDAC8_Start should be called to restart waveform.

Parameters: uint8 *WavePtr: Pointer to array containing waveform data

uint16 SampleSize: Size of waveform array pointed to by WavePtr. (Maximum sample size is 4000, minimum is 4)

Return Value: None

Side Effects: Does not affect WaveDAC8 type or power settings

void WaveDAC8_Wave2Setup(uint8 *WavePtr, uint16 SampleSize)

Description: Select a new waveform array for the waveform 2 output. The WaveDAC8_Stop function should be called prior to calling this function and WaveDAC8_Start should be called to restart waveform.

Parameters: uint8 * WavePtr: Pointer to array containing waveform data

uint16 SampleSize: Size of waveform array pointed to by WavePtr. (Maximum sample size is 4000, minimum is 4)

Return Value: None

Side Effects: Does not affect WaveDAC8 type or power settings

void WaveDAC8_StartEx(uint8 *WavePtr1, uint16 SampleSize1, uint8 *WavePtr2, uint16 SampleSize2)

Description: Select new waveform arrays for both waveform outputs and starts the WaveDAC8. The WaveDAC8_Stop function should be called prior to calling this function.

Parameters: uint8 *WavePtr1: Pointer to array containing waveform1 data

uint16 SampleSize1: Size of waveform1 array pointed to by WavePtr1. (Maximum sample size is 4000, minimum is 4)

uint8 * WavePtr2: Pointer to array containing waveform2 data

uint16 SampleSize2: Size of waveform2 array pointed to by WavePtr2. (Maximum sample size is 4000, minimum is 4)

Return Value: None

Side Effects: Does not affect WaveDAC8 type or power settings

void WaveDAC8_Init(void)

Description: Initializes or restores the component according to the customizer Configure dialog settings. It is not necessary to call WaveDAC8_Init() because the WaveDAC8_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 WaveDAC8_Enable(void)

Description: Activates the hardware and begins component operation. It is not necessary to call WaveDAC8_Enable() because the WaveDAC8_Start() API calls this function, which is the preferred method to begin component operation.

Parameters: None

Return Value: None

Side Effects: None

void WaveDAC8_SetSpeed(uint8 speed)

Description: Sets the drive mode / speed to one of the settings.

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

Power Setting	Notes
WaveDAC8_LOWSPEED	Lowest active power and slowest slew rate.
WaveDAC8_HIGHSPEED	Highest power and fastest slew rate.

Return Value: None

Side Effects: None

void WaveDAC8_SetRange (uint8 range)

Description: Sets the DAC range to one of the settings.

Parameters: uint8 range:

For VDAC mode:

Range Setting	Notes
WaveDAC8_VDAC8_RANGE_1V	1.02V
WaveDAC8_VDAC8_RANGE_4V	4.08V

For IDAC mode:

Range Setting	Notes
WaveDAC8_IDAC8_RANGE_32uA	32 uA.
WaveDAC8_IDAC8_RANGE_255uA	255 uA.
WaveDAC8_IDAC8_RANGE_2mA	2.04 mA

Return Value: None

Side Effects: The range value defines are applicable only for their DAC modes, e.g. if you try to use WaveDAC8_IDAC8_RANGE_32uA in VDAC mode, then a compilation error will be generated.

void WaveDAC8_SetValue (uint8 value)

Description: Sets the output of the DAC to the desired value. It is preferable to use this function when the clock is stopped. If this function is used during normal operation (clock is running), the predefined waveform may be interrupted.

Parameters: uint8 value: 8-bit DAC value from 0 to 255.

Return Value: None

Side Effects: None

void WaveDAC8_DacTrim(void)

Description: Sets the proper predefined trim calibration value for the present DAC mode and range.

Parameters: None

Return Value: None

Side Effects: None

void WaveDAC8_Sleep(void)

Description: This is the preferred API to prepare the component for sleep. The WaveDAC8_Sleep() API saves the current component state. Then it calls the WaveDAC8_Stop() function and calls WaveDAC8_SaveConfig() to save the hardware configuration.

Call the WaveDAC8_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 WaveDAC8_Wakeup(void)

Description: This is the preferred API to restore the component to the state when WaveDAC8_Sleep() was called. The WaveDAC8_Wakeup() function calls the WaveDAC8_RestoreConfig() function to restore the configuration. If the component was enabled before the WaveDAC8_Sleep() function was called, the WaveDAC8_Wakeup() function will also re-enable the component.

Parameters: None

Return Value: None

Side Effects: Calling the WaveDAC8_Wakeup() function without first calling the WaveDAC8_Sleep() or WaveDAC8_SaveConfig() function may produce unexpected behavior.

void WaveDAC8_SaveConfig(void)

Description: This function saves the component configuration. This function will also save the current component parameter values, as defined in the Configure dialog or as modified by appropriate APIs. This function is called by the WaveDAC8_Sleep() function.

Parameters: None

Return Value: None

Side Effects: None

void WaveDAC8_RestoreConfig(void)

Description: This function restores the component configuration. This function will also restore the component parameter values to what they were before calling the WaveDAC8_Sleep() function.

Parameters: None

Return Value: None

Side Effects: None

Global Variables

Function	Description
uint8 WaveDAC8_initVar	The initVar variable is used to indicate initial configuration of this component. This variable is prepended with the component name. The variable is initialized to zero and set to 1 the first time WaveDAC8_Start() is called. This allows for component initialization without reinitialization in all subsequent calls to the WaveDAC8_Start() routine. If reinitialization is required, then the WaveDAC8_Init() function can be called before the WaveDAC8_Start() or WaveDAC8_Enable() function.

Sample Firmware Source Code

PSoC Creator provides numerous 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.

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.

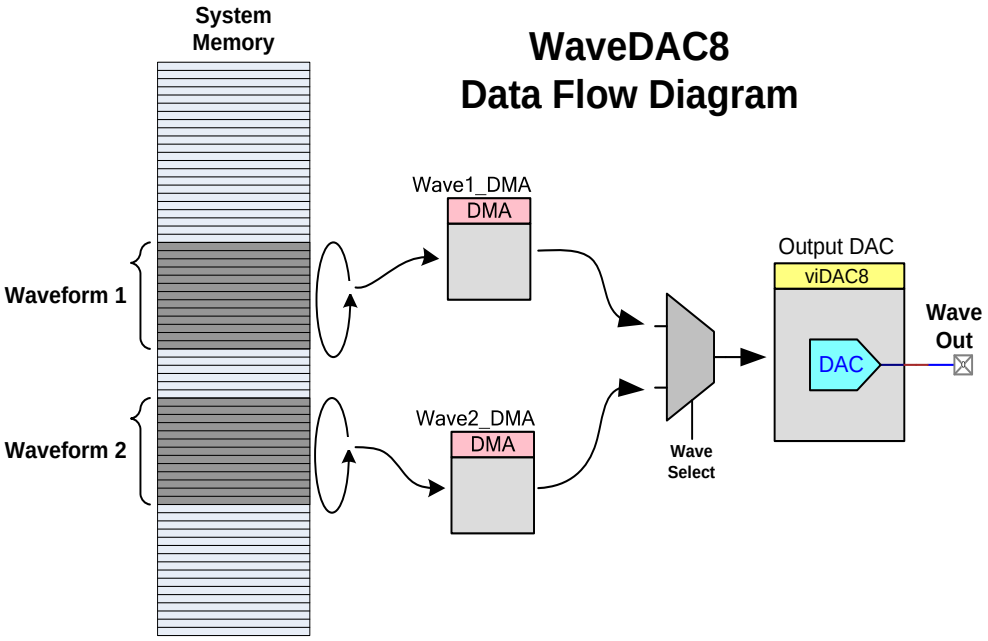
The WaveDAC8 component does not have any specific deviations.

Functional Description

The core of WaveDAC8 is the DAC. It will either be a standard VDAC8 (voltage DAC) or IDAC8 (Current DAC) depending on the range selected. The two DMA channels Wave1_DMA and Wave2_DMA are used to transfer the waveform array data in memory to either the IDAC or VDAC. When the user configures a waveform with the user interface, the component automatically configures each of the DMA channels to transfer the data. Both of these DMA channels transfer data to the DAC, but only one can operate at a time. The wave select “ws” input selects which of these DMA channels is triggered by the clock, using the demultiplexer

“DMA Select” to route the signal to the corresponding DMA channel. The two wave complete outputs “wc1” and “wc2” can be used to signal that the DMA channel has transferred the last value from the waveform table, or that one full waveform period has been completed.

Block Diagram and Configuration



The optional output buffer eliminates the load influence on the output voltage in VDAC mode.

DC and AC Electrical Characteristics

DC Characteristics (VDAC Mode) for CY8C38 Family

Parameter	Description	Conditions	Min	Typ	Max	Units
	Resolution		-	-	8	bits
Vout	Output voltage range, code = 255	1 V scale	-	1.02	-	Volt
		4 V scale, Vdda = 5 V	-	4.08	-	Volt
INL	Integral non linearity	1 V scale	-	±2.1	±2.5	LSB
DNL	Differential non linearity	CL=15 pF	-	±0.3	±1	LSB
Rout	Output resistance (for non-buffered ranges)	1 V scale	-	4	-	kΩ
		4 V scale	-	16	-	kΩ
	Monotonicity		-	-	Yes	-

Parameter	Description	Conditions	Min	Typ	Max	Units
Vos	Zero scale error		-	0	±0.9	LSB
Eg	Gain error	1 V scale	-	-	±2.5	%
		4 V scale	-	-	±2.5	%
TC_Eg	Temperature coefficient, gain error	1 V scale	-	-	0.03	%FSR / °C
		4 V scale	-	-	0.03	%FSR / °C
Idd	Operating current	Low speed mode	-	-	100	µA
		High speed mode	-	-	500	µA

DC Characteristics (IDAC Mode) for CY8C38 Family

Parameter	Description	Conditions	Min	Typ	Max	Units
	Resolution				8	bits
Iout	Output current	Range = 2mA, code = 255, VDDA ≥2.7 V, Rload = 600 Ω	-	2.04	-	mA
		Range = 2mA, high speed mode, code = 255, VDDA ≤2.7 V, Rload = 300 Ω	-	2.04	-	mA
		Range = 255 µA, code = 255, Rload = 600 Ω	-	255	-	µA
		Range = 32 µA, code = 255, Rload = 600 Ω		31.875		µA
	Monotonicity				Yes	
INL	Integral non linearity	Sink mode, range = 255 µA, Codes 8 – 255, Rload = 2.4 kΩ, Cload = 15 pF	-	±0.9	±1	LSB
		Source mode, range = 255 µA, Codes 8 – 255, Rload = 2.4 kΩ, Cload = 15 pF		±1.2	±1.6	LSB
DNL	Differential non linearity	Sink mode, range = 255 µA, Rload = 2.4 kΩ, Cload = 15 pF	-	±0.3	±1	LSB
		Source mode, range = 255 µA, Rload = 2.4 kΩ, Cload = 15 pF		±0.3	±1	LSB
Ezs	Zero scale error		-	0	±1	LSB
Eg	Gain error	Range = 2 mA, 25 °C	-	-	±2.5	%
		Range = 255 µA, 25 ° C	-	-	±2.5	%
		Range = 32 µA, 25 ° C	-	-	±3.5	%

Parameter	Description	Conditions	Min	Typ	Max	Units
TC_Eg	Temperature coefficient of gain error	Range = 2 mA			0.04	% / °C
		Range = 255 µA			0.04	% / °C
		Range = 32 µA			0.05	% / °C
Vcompliance	Dropout voltage, source or sink mode	Voltage headroom at max current, Rload to VDDAor Rload to VSSA, Vdiff from VDDA	1	-	-	V
IDD	Operating current, code = 0	Low speed mode, source mode, range = 32 µA	-	44	100	µA
		Low speed mode, source mode, range = 255 µA,		33	100	µA
		Low speed mode, source mode, range = 2 mA		33	100	µA
		Low speed mode, sink mode, range = 32 µA		36	100	µA
		Low speed mode, sink mode, range = 255 µA		33	100	µA
		Low speed mode, sink mode, range = 2 mA		33	100	µA
		High speed mode, source mode, range = 32 µA		310	500	µA
		High speed mode, source mode, range = 255 µA		305	500	µA
		High speed mode, source mode, range = 2 mA		305	500	µA
		High speed mode, sink mode, range = 32 µA		310	500	µA
		High speed mode, sink mode, range = 255 µA		300	500	µA
		High speed mode, sink mode, range = 2 mA		300	500	µA

DC Characteristics (VDAC Mode) for CY8C58LP family

Parameter	Description	Conditions	Min	Typ	Max	Units
	Resolution		-	8	-	bits
Vout	Output voltage range, code = 255	1 V scale	-	1.02	-	Volt
		4 V scale, Vdda = 5 V	-	4.08	-	Volt

Parameter	Description	Conditions	Min	Typ	Max	Units
INL	Integral non linearity	1 V scale	-	±2.1	±2.5	LSB
		4 V scale	-	±2.1	±2.5	LSB
DNL	Differential non linearity	1 V scale	-	±0.3	±1	LSB
		4 V scale	-	±0.3	±1	LSB
Rout	Output resistance	1 V scale	-	4	-	kΩ
		4 V scale	-	16	-	kΩ
	Monotonicity		-	-	Yes	-
Vos	Zero scale error		-	0	±0.9	LSB
Eg	Gain error	1 V scale	-	-	±2.5	%
		4 V scale	-	-	±2.5	%
TC_Eg	Temperature coefficient, gain error	1 V scale	-	-	0.03	%FSR / °C
		4 V scale	-	-	0.03	%FSR / °C
Idd	Operating current	Low speed mode	-	-	100	µA
		High speed mode	-	-	500	µA

DC Characteristics (IDAC Mode) for CY8C58LP family

Parameter	Description	Conditions	Min	Typ	Max	Units
	Resolution				8	bits
Iout	Output current	Range = 2mA, code = 255, VDDA ≥2.7 V, Rload = 600 Ω	-	2.04	-	mA
		Range = 2mA, high speed mode, code = 255, VDDA ≤2.7 V, Rload = 300 Ω	-	2.04	-	mA
		Range = 255 µA, code = 255, Rload = 600 Ω	-	255	-	µA
		Range = 32 µA, code = 255, Rload = 600 Ω		31.875		µA
	Monotonicity				Yes	
INL	Integral non linearity	Sink mode, range = 255 µA, Codes 8 – 255, Rload = 2.4 kΩ, Cload = 15 pF	-	±0.9	±1	LSB
		Source mode, range = 255 µA, Codes 8 – 255, Rload = 2.4 kΩ, Cload = 15 pF	-	±1.2	±1.6	LSB

Parameter	Description	Conditions	Min	Typ	Max	Units
		Source mode, range = 32µA, Codes 8 - 255, Rload = 20 kΩ, Cload = 15 pF	-	±0.9	±2	LSB
		Sink mode, range = 32µA, Codes 8 - 255, Rload = 20 kΩ, Cload = 15 pF	-	±0.9	±2	LSB
		Source mode, range = 2mA, Codes 8 - 255, Rload = 600 Ω, Cload = 15 pF	-	±0.9	±2	LSB
		Sink mode, range = 2mA, Codes 8 - 255, Rload = 600 Ω, Cload = 15 pF	-	±0.6	±1	LSB
DNL	Differential non linearity	Sink mode, range = 255 µA, Rload = 2.4 kΩ, Cload = 15 pF	-	±0.3	±1	LSB
		Source mode, range = 255 µA, Rload = 2.4 kΩ, Cload = 15 pF	-	±0.3	±1	LSB
		Source mode, range = 31.875 µA, Rload = 20 kΩ, Cload = 15 pF	-	±0.2	±1	LSB
		Sink mode, range = 31.875 µA, Rload = 20 kΩ, Cload = 15 pF	-	±0.2	±1	LSB
		Source mode, range = 2.0 4 mA, Rload = 600 Ω, Cload = 15 pF	-	±0.2	±1	LSB
		Sink mode, range = 2.0 4 mA, Rload = 600 Ω, Cload = 15 pF	-	±0.2	±1	LSB
Ezs	Zero scale error		-	0	±1	LSB
Eg	Gain error	Range = 2 mA, 25 °C	-	-	±2.5	%
		Range = 255 µA, 25 ° C	-	-	±2.5	%
		Range = 32 µA, 25 ° C	-	-	±3.5	%
TC_Eg	Temperature coefficient of gain error	Range = 2 mA			0.045	% / °C
		Range = 255 µA			0.045	% / °C
		Range = 32 µA			0.05	% / °C
Vcompliance	Dropout voltage, source or sink mode	Voltage headroom at max current, Rload to Vdda or Rload to Vssa, Vdiff from Vdda	1	-	-	V

Parameter	Description	Conditions	Min	Typ	Max	Units
IDD	Operating current, code = 0	Slow mode, source mode, range = 32 µA	-	44	100	µA
		Slow mode, source mode, range = 255 µA,		33	100	µA
		Slow mode, source mode, range = 2 mA		33	100	µA
		Slow mode, sink mode, range = 32 µA		36	100	µA
		Slow mode, sink mode, range = 255 µA		33	100	µA
		Slow mode, sink mode, range = 2 mA		33	100	µA
		Fast mode, source mode, range = 32 µA		310	500	µA
		Fast mode, source mode, range = 255 µA		305	500	µA
		Fast mode, source mode, range = 2 mA		305	500	µA
		Fast mode, sink mode, range = 32 µA		310	500	µA
		Fast mode, sink mode, range = 255 µA		300	500	µA
		Fast mode, sink mode, range = 2 mA		300	500	µA

AC Characteristics (VDAC Mode) for both CY8C38 and CY8C58LP families

Parameter	Description	Conditions	Min	Typ	Max	Units
Fdac	Update rate	1V mode	-	-	1	Msps
	Update rate	4V mode	-	-	250	Ksps
TsettleP	Settling time to 0.1%, step 25% to 75%	1 V scale, Cload = 15 pF	-	0.45	1	µs
		4 V scale, Cload = 15 pF	-	0.8	3.2	µs
TsettleN	Settling time to 0.1%, step 75% to 25%	1 V scale, Cload = 15 pF	-	0.45	1	µs
		4 V scale, Cload = 15 pF	-	0.7	3	µs
	Voltage noise	Range = 1 V, High speed mode, VDDA= 5 V, 10 kHz	-	750	-	nV/sqrtHz

AC Characteristics (IDAC Mode) for both CY8C38 and CY8C58LP families

Parameter	Description	Conditions	Min	Typ	Max	Units
Fdac	Update rate		-	-	6	Msps
Tsettle	Settling time to 0.5 LSB	Range = 32 µA or 255 µA, full scale transition, High speed mode, 600Ω15-pF load	-	-	125	ns
	Current noise	Range = 255 µA, source mode, High speed mode, VDDA= 5 V, 10 kHz	-	340	-	pA/sqrtHz

Component Changes

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

Version	Description of Changes	Reason for Changes / Impact
2.10.a	Minor datasheet edits.	
2.10	MISRA violation 19.8 is fixed. Added an explanation about the file format for the “Arbitrary (From File)” Wave Type option.	Undocumented MISRA violation. Datasheet clarification.
2.0.b	Added explanation for possible Sample Rate / DMA issues.	Make the datasheet more detailed to improve user experience with the component.
2.0.a	Edited datasheet to add Component Errata section.	Document that the component was changed, but there is no impact to designs.
2.0	First release of component as part of PSoC Creator.	This component was previously made available in the Cypress Community Forums and in Application Note AN69133.

© Cypress Semiconductor Corporation, 2016-2017. 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.