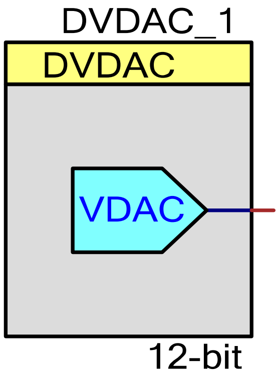
9 to 12 bit Dithered Voltage DAC (DVDAC)

Version 1.4

Features

- Two voltage ranges, 1 and 4 volts
- Adjustable 9, 10, 11, or 12 bit resolution
- Dithering used DMA for zero CPU overhead
- DNL less than 1 from 9 to 12 bits
- Uses a single DAC block



General Description

The DVDAC component has a selectable resolution between 9 and 12 bits. Dithering is used to increase the resolution of its underlying 8-bit VDAC. Only a small output capacitor is required to suppress the noise generated by dithering.

Input/Output Connections

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

Vout – Analog

The Vout terminal is the connection to the DAC's voltage output. It may be routed to any analog compatible pin on the PSoC.

An external capacitor may be required between this output and Vssa to reduce the noise generated by the voltage dithering. The dithered signal is only at the LSb level so the noise is only 4mV for the 1 volt range and 16 mV for the 4 volt range. The actual dither frequency varies with the resolution of the DAC. If a 2 MHz dither clock is used for the PWM and the period is set to 4 (10-bits) the actual dither frequency is about 250 KHz (1 MHz/4). The output resistance of the DAC is about 4K for the 1 volt range and 16K ohms for the 4 volt range.

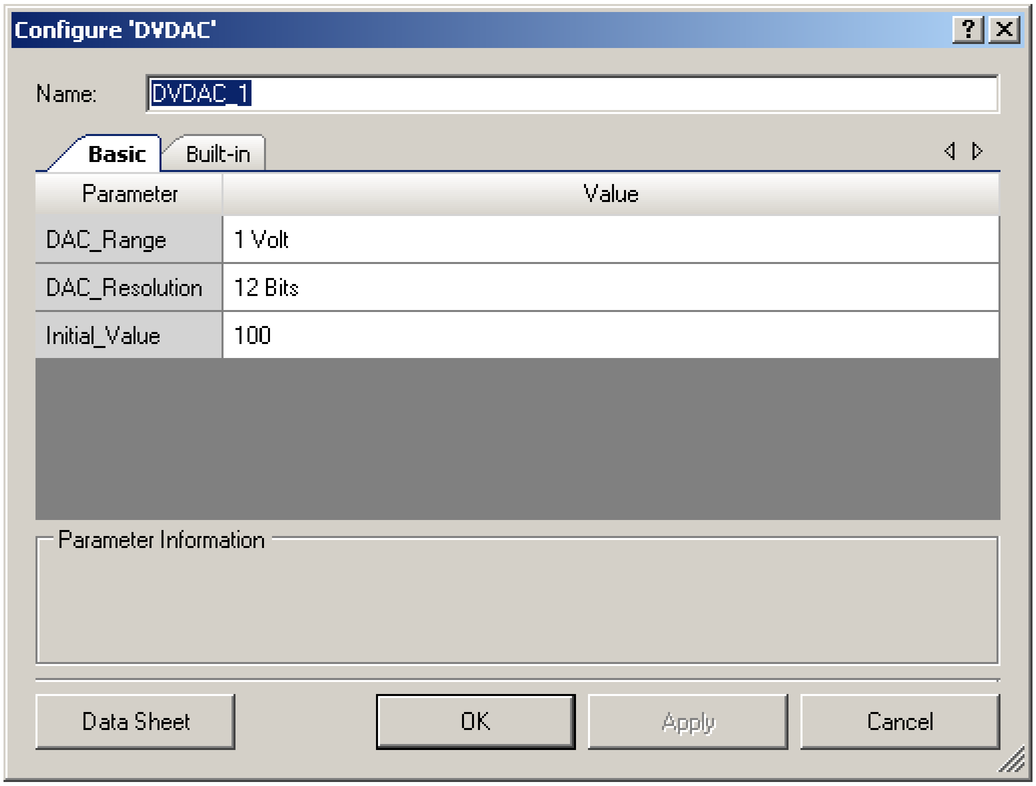
Table 4 below shows example capacitor values and frequency response for both the 1 and 4 volt ranges for each of the resolution settings.

Table 1 Capacitor value and dither frequency

Resolution (bits)	9	10	11	12
Cap Value (1 volt range)	160pF	630pF	2.5nF	0.01uF
Cap Value (4 volt range)	630pF	2.5nF	0.01uF	0.04uF
Dither Frequency (1 volt range)	500 kHz	250 kHz	125 kHz	62.5 kHz
Dither Frequency (4 volt range)	125 kHz	62.5 kHz	31.3 kHz	15.6 kHz

Parameters and Setup

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



Configure 'DVDAC'

Name: DVDAC_1

Parameter	Value
DAC_Range	1 Volt
DAC_Resolution	12 Bits
Initial_Value	100

The DVDAC component provides the following parameters.

DAC_Range

This parameter allows you to set one of two voltage ranges as the default value. The range may be changed at any time during runtime with the SetRange() function.

Table 2 DVDAC ranges

Range	Input value range	Output Equation
1 Volt	0 to 1.023 Volts	Vout = (value/2^{bits}) * 1.024 Volts
4 Volts	0 to 4.095 Volts	Vout = (value/2^{bits}) * 4.096 Volts

Initial_Value

This is the initial value the DVDAC will present after the Start() command is executed. The SetValue() function or a direct write to the DAC register will override the default value at anytime.

Table 3 DVDAC range and resolution parameters

Resolution	Valid input value range	Step Size (1 volt range)	Step Size (4 volt range)
9	0 to 510	2 mV	8 mV
10	0 to 1020	1 mV	4 mV
11	0 to 2040	0.5 mV	2 mV
12	0 to 4080	0.25 mV	1 mV

DAC_Resolution

This parameter allows you to select one of four resolutions (9, 10, 11, or 12).

Resources

The DVDAC component uses one viDAC8 analog block and a DMA channel.

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 "DVDAC_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 "DVDAC".

Function	Description
void DVDAC_Start(void)	Initialize the DVDAC with default customizer values.
void DVDAC_Stop(void)	Disables the DVDAC and sets it to the lowest power state.
void DVDAC_SetValue(uint16 value)	Sets DAC level. See Table 2 above for valid range.
void DVDAC_SetRange(uint8 range)	Sets range to 1 or 4 volts.
void DVDAC_Init(void)	Initializes or Restores default DVDAC configuration
void DVDAC_Enable(void)	Enable the DVDAC.
void DVDAC_Sleep(void)	Stops DVDAC and saves the user configuration.
void DVDAC_WakeUp(void)	Restores and enables the user configuration.
void DVDAC_SaveConfig(void)	Empty function. Provided for future usage.
void DVDAC_RestoreConfig(void)	Empty function. Provided for future usage.

Global Variables

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

void DVDAC_Start(void)

Description: Initialize the DVDAC with default customizer values. Enable and power up the DVDAC to the given power level.

Parameters: None

Return Value: None

Side Effects: None

void DVDAC_Stop(void)

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

Parameters: None

Return Value: None

Side Effects: None

void DVDAC_SetRange(uint8 range)

Description: Sets range to 1 or 4 volts.

Parameters: (uint8) range: Sets full scale range for DVDAC. See table below for ranges.

Option	Description
DVDAC_RANGE_1V	Set full scale range of 1 V
DVDAC_RANGE_4V	Set full scale range of 4 V

Return Value: None

Side Effects: None

void DVDAC_SetValue(uint16 value)

Description: Sets output voltage.

Parameters: (uint16) value: Output value. See Table 2 above for valid input range.

Return Value: None

Side Effects: None

void DVDAC_Init(void)

Description: Initializes/restores default DVDAC configuration.

Parameters: None

Return Value: None

Side Effects: All registers will be set to their initial values. This will re-initialize the component. Calling the Init() function requires a call to SetValue() if you intend to set a new value other than what is currently in the register.

void DVDAC_Enable(void)

Description: Enables the DVDAC.

Parameters: None

Return Value: None

Side Effects: None

void DVDAC_Sleep(void)

Description: Stops the operation. Saves the user configuration and the component enable state. Should be called just prior to entering sleep.

Parameters: None

Return Value: None

Side Effects: None

void DVDAC_Wakeup(void)

Description: Restores the configuration registers and component enable state. Should be called just after awaking from sleep.

Parameters: None

Return Value: None

Side Effects: None

void DVDAC_SaveConfig(void)

Description: Saves the user configuration.

Parameters: None

Return Value: None

Side Effects: None

void DVDAC_RestoreConfig(void)

Description:	Restores the user configuration.
Parameters:	None
Return Value:	None
Side Effects:	None

DMA

The DVDAC component is not compatible with DMA. It uses a lookup table and DMA to dither between two values. The value should only be changed by using the SetValue( ) function.

Sample Firmware Source Code

The following is a C language example demonstrating the basic functionality of the DVDAC component. This example assumes the component has been placed in a design with the default name "DVDAC_1."

Note If you rename your component you must also edit the example code as appropriate to match the component name you specify.

```
#include <device.h>

void main()
{
    DVDAC_1_Start();                      // Enable DVDAC
    DVDAC_1_SetRange(DVDAC_1_RANGE_1V);   // Set full scale range to 1.024V
    DVDAC_1_SetValue(100);                // Set value to 200 mV (9-bit mode)
}
```

Functional Description and Block Diagram

The DVDAC uses voltage dithering to increase the resolution of standard 8-bit voltage DAC found in PSoC3 and PSoC5 parts. When the API SetValue( ) function is called, it creates a RAM table of values. DMA is used to read this table and periodically write the values to the DMA data register. The clock periodically latches the last value written to the DAC, and at the same time triggers the next DMA cycle to update the DAC with a new value. The clock frequency is 2 MHz for the 1 volt range and 500 KHz for the 4 volt range.

The figure below shows the schematic for the DVDAC component.

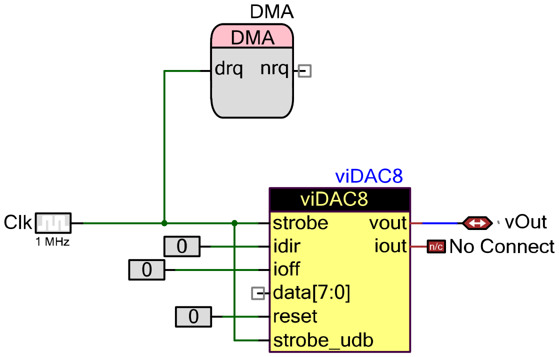


Figure 1 DVDAC schematic

Registers

Since the DVDAC makes use of software and DMA to create the added resolution, direct register access is not advised. The DAC register information is provided here for reference purposes only.

Table 4 DVDAC_CR0

Bits	7	6	5	4	3	2	1	0
Value	RSVD			mode	Range[1:0]		hs	RSVD

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

Table 5 DVDAC_CR1

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

- mx_data: Select 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

Table 6 DVDAC_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

The following values are indicative of expected performance and based on initial characterization data.

DC Characteristics

Parameter	Description	Conditions	Min	Typ	Max	Units
	Output current					
Iout	High	Vdda ≤ 2.7 V, RL 300 Ω	-	-	2.047	mA
	Medium	RL 600 Ω	-	-	255	µA
INL	Integral non linearity	RL 600 Ω, CL=15 pF , 9-bit	-	± 1	-	LSb
		RL 600 Ω, CL=15 pF , 10-bit	-	± 2	-	LSb
		RL 600 Ω, CL=15 pF , 11-bit		± 4	-	LSb
DNL	Differential non linearity	RL 600 Ω, CL=15 pF , 9-bit	-	0.35	-	LSB
		RL 600 Ω, CL=15 pF , 10-bit	-	0.6	-	LSB
		RL 600 Ω, CL=15 pF , 11-bit	-	1.0	-	LSB
Ezs	Zero scale error		-	0	±1	LSB
Eg	Gain error	Uncompensated	-	-	2.5	%
		Temperature compensated	-	-	TBD	%
IDAC_ICC	DAC current low speed mode	Code = 0	-	-	200	µA
IDAC_ICC	DAC current high speed mode	Code = 0	-	-	1000	µA

AC Characteristics

Parameter	Description	Conditions	Min	Typ	Max	Units
Fdac	Update rate (1 volt range)		-	1		Msps
	Update rate (4 volt range)			250		Ksps
Tsettle	Settling time to 0.5LSB					
	1 volt range	9 bit resolution	-	0.9		uS
		10 bit resolution		5.3		uS
		11 bit resolution		28		uS
		12 bit resolution		140		uS
	4 volt range	9 bit resolution		3.5		uS
		10 bit resolution		21		uS
		11 bit resolution		112		uS
		12 bit resolution		560		uS

Reference Component Changes

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

Version	Description of Changes
1.0	First release
1.2	Add no-connect on current outputs in component
1.4	Updated DMA component to support PSoC Creator 2.1 SP1

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

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

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

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.