Analog Engineer's Circuit

Voltage Margining and Scaling Circuit with a Voltage Output Smart DAC

Smart DAC *Katlynne Jones*

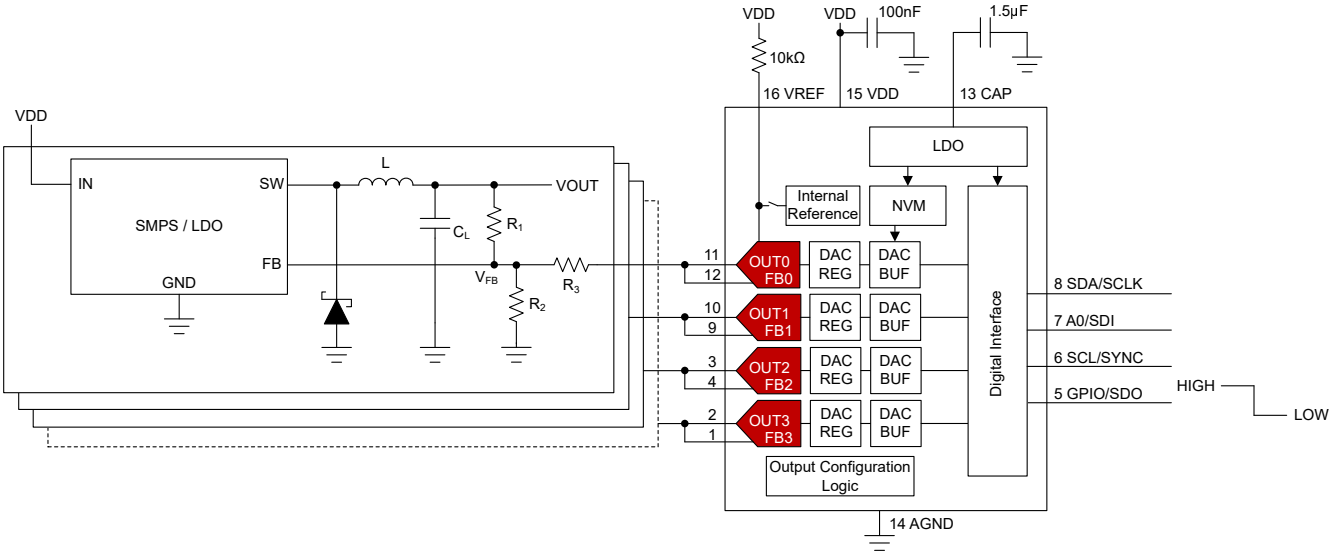
Design Objective

Key Input Parameter	Key Output Signal	Recommended Device
SPI or I^2C communication	0V to 1.2V analog DAC output, 3.3V ±10% SMPS output	DAC53204, DAC63204, DAC43204, DAC53004

Objective: *Provide a margin voltage for an SMPS output of ±10% the nominal value*

Design Description

This circuit uses a four-channel buffered voltage output DAC to voltage margin a switch-mode power supply (SMPS). A voltage margining circuit is used to trim, scale, or test the output of a power converter. Adjustable power supplies, such as low dropout regulators (LDOs), DC/DC converters, or SMPS provide a feedback (FB) input that is used to control the desired output. A precision smart DAC, such as the DAC53204 or DAC53004, provides linear control of the power supply output when the DAC is supplied and the output is powered on. Most DACs include an internal pull-down resistor at the voltage output when the DAC is supplied but the output is in power down mode. Also, when most DACs are completely powered off, the ESD cells on the output pin conduct current if the output is pulled away from ground which is the case in voltage margining circuits. The DAC53204 or DAC53004 provides a high impedance (Hi-Z) output when the DAC is powered off or when the output channel is in power down mode, meaning that the DACs draw very little current through the FB pin of the SMPS and the output is set at the nominal voltage. The DAC53204 and DAC53004 have a general-purpose input (GPI) pin that allows the DAC output to be toggled between a high and low voltage output. This allows the SMPS to be toggled within ±10% of the nominal output value. All register settings can be saved using the non-volatile memory (NVM) on the DAC53204 and DAC53004 meaning that the devices can be used without a processor, even after a power cycle. This circuit can be used in applications such as Communications Equipment, Enterprise Systems, Test and Measurement, and general-purpose power-supply modules.



Design Notes

1. The *DACx3204 12-Bit, 10-Bit, and 8-Bit, Quad Voltage and Current Output Smart DACs With Auto-Detected I2C, PMBus™, or SPI Interface Data Sheet* recommends using a 100nF decoupling capacitor for the VDD pin and a 1.5µF or greater bypass capacitor for the CAP pin. The CAP pin is connected to the internal LDO. Place these capacitors close to the device pins.
2. There are three reference options for the DAC53204:
   a. An external reference of 1.7V to V_{DD} can be applied to the VREF pin of the device. Connect a 100nF capacitor between the VREF pin and the AGND pin when using the external reference. Use a pullup resistor to V_{DD} when the external reference is not used.
   b. There is a precision 1.21V reference with ×1.5, ×2, ×3, and ×4 gain options.
   c. V_{DD} can be used as a reference.
3. The nominal voltage of the SMPS is set by resistors R_1 and R_2. The SMPS uses an internal 600mV reference voltage at the FB pin to determine the voltage at the output. Calculate R_1 and R_2 using the following equations:

$$R_1 = \frac{V_{NOMINAL} - V_{FB}}{I_{NOMINAL}}$$

$$R_2 = \frac{R_1 \times V_{FB}}{V_{NOMINAL} - V_{FB}}$$

   A nominal current of 100µA through R_1 and R_2 and 3.3V nominal output voltage is used. With these values the equations become:

$$R_1 = \frac{3.3V - 0.6V}{100\mu A} = 27k\Omega$$

$$R_2 = \frac{27k\Omega \times 0.6V}{3.3V - 0.6V} = 6k\Omega$$

4. To achieve the desired margin, the DAC53204 must sink or source additional current through R_1. This current (I_{MARGIN}) is calculated by:

$$I_{MARGIN} = \left(\frac{V_{NOMINAL} \times (1 + MARGIN) - V_{FB}}{R_1}\right) - I_{NOMINAL}$$

   For a ±10% margin, the equation becomes:

$$I_{MARGIN} = \left(\frac{3.3V \times (1 + 0.10) - 0.6V}{27k\Omega}\right) - 100\mu A = 12\mu A$$

5. The voltage output of the DAC53204 is turned into a current through the series resistor R_3. R_3 is calculated by:

$$R_3 = \frac{V_{DAC} - V_{FB}}{I_{MARGIN}}$$

   Avoid DAC codes near zero- and full-scale when determining the DAC output voltage range to avoid the zero- and full-scale errors at these codes. This design uses the internal 1.21V reference with a gain of ×1.5 giving a full-scale voltage of 1.82V. The DAC53204 has a max zero-code error of 15mV, so a 20mV minimum output is a safe choice. I_{MARGIN} is considered positive and is being sourced from the DAC when $V_{DAC} > V_{FB}$. I_{MARGIN} is considered negative and is being sunk by the DAC53204 when $V_{DAC} < V_{FB}$. R_3 is calculated to be 48.3kΩ:

$$R_3 = \frac{20mV - 0.6V}{-12\mu A} = 48.3k\Omega$$

   The maximum output becomes:

$$V_{DAC,\,MAX} = (R_3 \times I_{MARGIN}) + V_{FB}$$

$$V_{DAC,\,MAX} = (48.3k\Omega \times 12\mu A) + 0.6V = 1.18V$$

6. The DAC codes for $V_{DAC,MAX}$ and $V_{DAC,MIN}$ are stored in the DAC-MARGIN-HIGH and DAC-MARGIN-LOW registers. The codes programmed to these registers, in decimal, is calculated using:

$$\text{DAC\_MARGIN\_HIGH} = \frac{V_{DAC,\,MAX}}{V_{REF} \times \text{GAIN}} \times 1024$$

$$\text{DAC\_MARGIN\_LOW} = \frac{V_{DAC,\,MIN}}{V_{REF} \times \text{GAIN}} \times 1024$$

The equations become:

$$\text{DAC\_MARGIN\_HIGH} = \frac{1.18\text{V}}{1.21\text{V} \times 1.5} \times 1024 = 665.7\text{d}$$

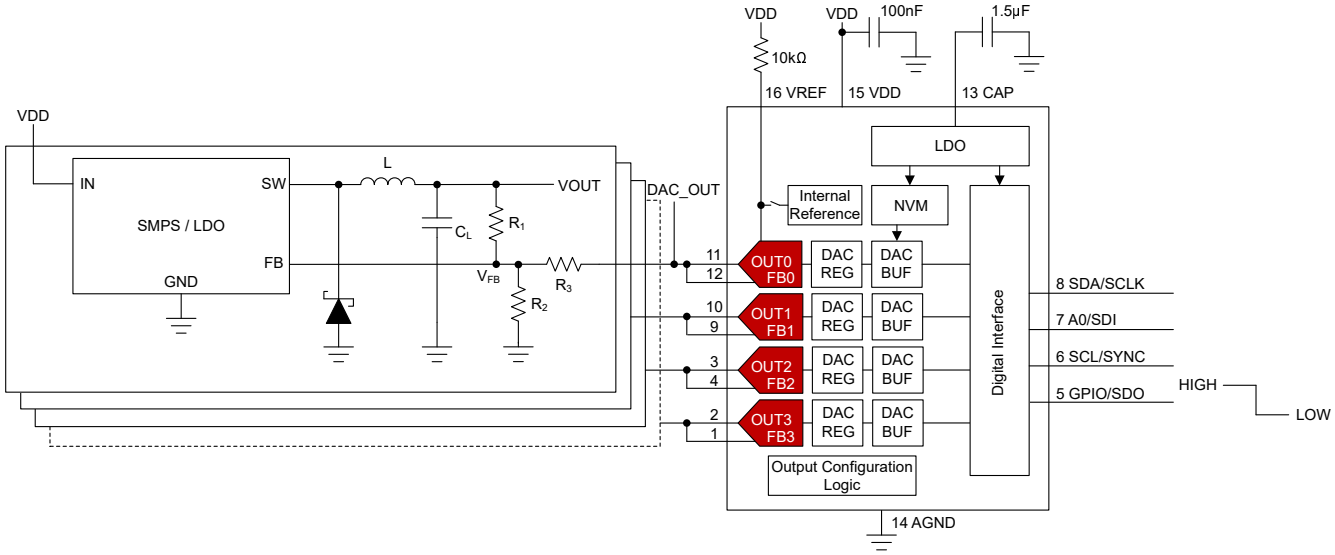
$$\text{DAC\_MARGIN\_LOW} = \frac{20\text{mV}}{1.21\text{V} \times 1.5} \times 1024 = 11.2\text{d}$$

This is rounded to 666d and 11d to give a $V_{DAC,MAX}$ of 1.18V and a $V_{DAC,MIN}$ of 19.5mV.

7. Using a 1.21V reference with a ×1.5 gain and the 10-bit DAC53204, the LSB size, or step size between each code, is about 1.8mV. Using lower reference voltages decreases the LSB size and thus increases the resolution of $V_{DAC,MAX}$ and $V_{DAC,MIN}$.
8. In this design, GPI is used for Margin High, Low function. A high on GPI sets the DAC output to $V_{DAC,MAX}$ and the SMPS V_{OUT} to margin low, or 2.97V. A low on GPI sets the DAC output to $V_{DAC,MIN}$ and the SMPS V_{OUT} to margin high, or 3.63V.
9. The DAC53204 can be programmed with the initial register settings described in the Register Settings section using I^2C or SPI. The initial register settings can be saved in the NVM by writing a 1 to the NVM-PROG field of the COMMON-TRIGGER register. After programming the NVM, the device loads all registers with the values stored in the NVM after a reset or a power cycle.

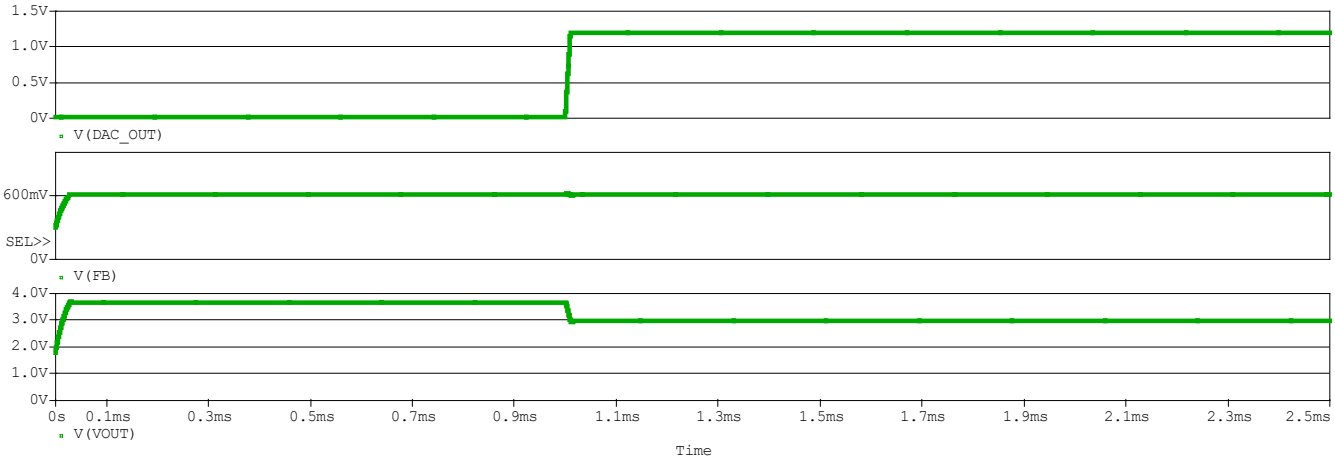
Design Simulations

This schematic is used for the following simulation of the DAC53204.



Transient Simulation Results

The simulation shows the SMPS output (V_{OUT}) responding to the changes on the DAC53204 output (DAC_OUT). When DAC_OUT is at $V_{DAC,MAX}$ the SMPS V_{OUT} goes to margin low, or 2.97V. When DAC_OUT is at $V_{DAC,MIN}$ the SMPS V_{OUT} goes to margin high, or 3.63V.



Register Settings

Register Settings for DAC53204 Voltage Margining

Register Address	Register Name	Setting	Description
0x1F	COMMON-CONFIG	0x1249	[15] 0b0: Write 0b1 to set window-comparator output to a latching output
			[14] 0b0: Write 0b1 to lock device. Unlock by writing 0b0101 to DEV-UNLOCK field in the COMMON-TRIGGER register
			[13] 0b0: Write 0b1 to set fault-dump read enable at address 0x01
			[12] 0b1: Enables the internal reference
			[11:10] 0b00: Powers-up VOUT3
			[9] 0b1: Powers-down IOUT3
			[8:7] 0b00: Powers-up VOUT2
			[6] 0b1: Powers-down IOUT2
			[5:4] 0b00: Powers-up VOUT1
			[3] 0b1: Powers-down IOUT1
			[2:1] 0b00: Powers-up VOUT0
			[0] 0b1: Powers-down IOUT0
0x24	GPIO-CONFIG	0x01F5	[15] 0b0: Write 0b1 to enable glitch filter on GPI
			[14] 0b0: Don't care
			[13] 0b0: Write 0b1 to enable output mode on GPIO pin
			[12:9] 0b0000: Selects the STATUS function setting mapped to GPIO as output
			[8:5] 0b1111: Enables GPI function on all channels
			[4:1] 0b1010: Selects GPI to trigger margin-high, margin-low
			[0] 0b1: Enables input mode for GPIO pin
0x20	COMMON-TRIGGER	0x0002	[15:12] 0b0000: Write 0b0101 to unlock the device
			[11:8] 0b0000: Write 0b1010 to trigger a POR reset
			[7] 0b0: Write 0b1 to trigger LDAC operation if the respective SYNC-CONFIG-X bit in the DAC-X-FUNC-CONFIG register is 1
			[6] 0b0: Write 0b1 to set the DAC registers and outputs to zero-code or mid-code based on the respective CLR-SEL-X bit in the DAC-X-FUNC-CONFIG register
			[5] 0b0: Don't care
			[4] 0b0: Write 0b1 to trigger fault-dump sequence
			[3] 0b0: Write 0b1 to trigger PROTECT function
			[2] 0b0: Write 0b1 to read one row of NVM for fault-dump
			[1] 0b1: Write 0b1 to store applicable register settings to the NVM
			[0] 0b0: Write 0b1 to reload applicable registers with existing NVM settings
0x01, 0x07, 0x0D, 0x13	DAC-X-MARGIN-HIGH	0xA680	[15:6] 0x29A: 10-bit data updates the MARGIN-HIGH code
			[5:0] 0b000000: Don't care
0x02, 0x08, 0x0E, 0x14	DAC-X-MARGIN-LOW	0x02C0	[15:6] 0x00B: 10-bit data updates the MARGIN-LOW code
			[5:0] 0b000000: Don't care

Pseudo Code Example

The following shows a pseudo code sequence to program the initial register values to the NVM of the DAC53204. The values given here are for the design choices made in the Design Notes.

Pseudo Code Example

```
//SYNTAX: WRITE <REGISTER NAME (Hex code)>, <MSB DATA>, <LSB DATA>
//Power-up voltage output on all channels, enables internal reference
WRITE COMMON-CONFIG(0x1F), 0x12, 0x49
//Configure GPI for Margin-High, Low function
WRITE GPIO-CONFIG(0x24), 0x01, 0xF5
//Write DAC margin high code (repeat for all channels)
WRITE DAC-0-MARGIN-HIGH(0x01), 0xA6, 0x80
//Write DAC margin low code (repeat for all channels)
WRITE DAC-0-MARGIN-LOW(0x02), 0x02, 0xC0
//Save settings to NVM
WRITE COMMON-TRIGGER(0x20), 0x00, 0x02
```

Design Featured Devices

Device	Key Features	Link
DAC53204	4-channel, 10-bit, VOUT and IOUT smart DAC with I^2C, SPI and Hi-Z out during power off	www.ti.com/product/DAC53204
DAC63204	4-channel, 12-bit, VOUT and IOUT smart DAC with I^2C, SPI and Hi-Z out during power off	www.ti.com/product/DAC63204
DAC43204	4-channel, 8-bit, VOUT and IOUT smart DAC with I^2C, SPI and Hi-Z out during power off	www.ti.com/product/DAC43204
DAC53004	Ultra-low-power, 4-channel, 10-bit, VOUT and IOUT smart DAC with I2C, SPI and Hi-Z out during power off	www.ti.com/product/DAC53004

Find other possible devices using the Parametric search tool.

Design References

See *Analog Engineer's Circuit Cookbooks* for TI's comprehensive circuit library.

Additional Resources

- Texas Instruments, Smart DAC Python Examples
- Texas Instruments, DAC63204 Evaluation Module
- Texas Instruments, DAC63204 EVM User's Guide
- Texas Instruments, Precision Labs - DACs

For direct support from TI Engineers, use the E2E™ forums:

e2e.ti.com

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265