Analog Engineer's Circuit

Programmable, High-Current Sink Circuit With DSBGA Smart DAC

Smart DAC

Katlynne Jones

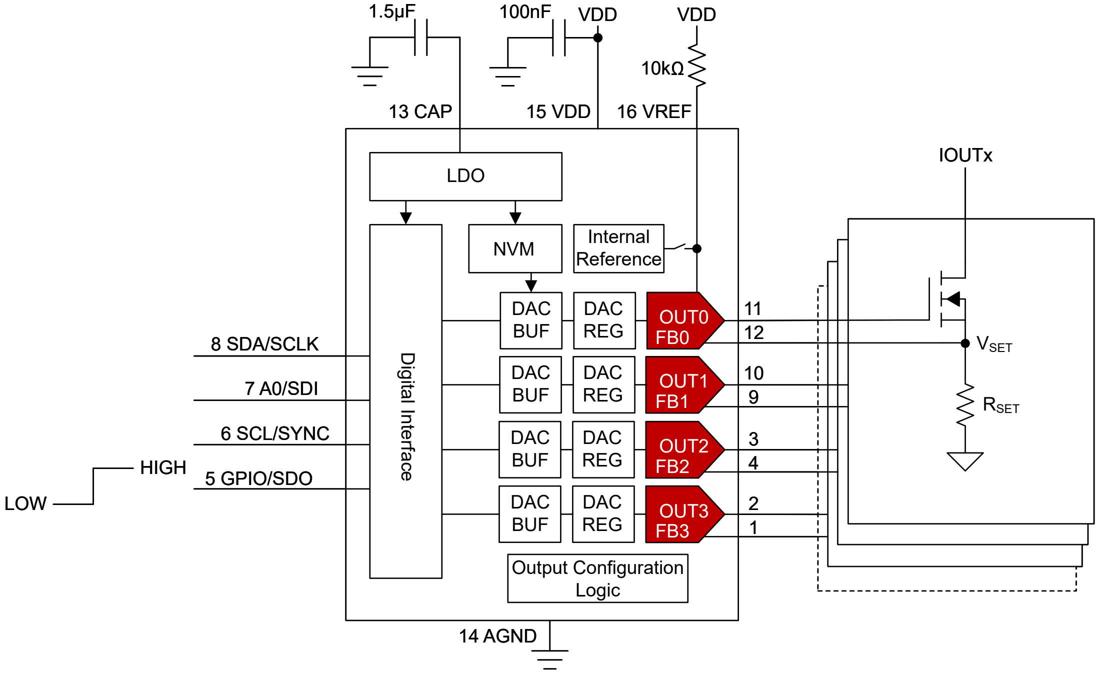
Design Objective

Key Input Parameter	Key Output Signal	Recommended Device
SPI or I^{2}C communication to control current output	0-mA to 200-mA programmable current sink	DAC63204W, DAC53204W, DAC63004W, DAC53004W,

Objective: 0-mA to 200-mA programmable current sink circuit with a 1.75-mm × 1.75-mm smart DAC.

Design Description

This design uses the force-sense outputs of the of the DAC63204W or DAC53204W (DACx3204W) with external FemtoFET™ MOSFETs to create a 4-channel programmable current sink with a 4 mm × 4 mm total size. The force-sense outputs of the DACx3204W compensate for the gate-source voltage drop caused by temperature, drain current, and aging of the MOSFET. A resistor, R_{SET}, connected to the source of the MOSFET sets the output current range. The DACx3204W has a general-purpose input/output (GPIO) pin that can be configured as a digital input to switch the output current between two values with a programmable slew rate. All register settings are saved using the integrated non-volatile memory (NVM), enabling the device to be used without runtime software, even after a power cycle or reset. This circuit can be used in optical module applications that require a high current output with a small size.



Design Notes

1. The *DACx3204W 12-Bit and 10-Bit, Quad Voltage and Current Output Smart DACs With Auto-Detected I2C, SPI, or PMBus® Interface in DSBGA Package* data sheet recommends using a 100-nF decoupling capacitor for the VDD pin and a 1.5-µF 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. Connect a 100-nF capacitor from VREF to GND if the external reference is used. Ramp up the external reference after VDD. Connect a pullup resistor from the VREF pin to VDD if the external reference is not used. This example uses the internal reference and the VREF pin is pulled up to VDD with a 10-kΩ resistor.
3. This design uses one CSD13380F3 FemtoFET per output channel. The CSD13380F3 has an ultra-small 0.73 mm × 0.64 mm footprint which keeps the overall application size small.
4. The force-sense configuration of the DAC63204W in this circuit eliminates most of the errors from the CSD13380F3 including the gate-source voltage drop caused by temperature, drain current, and aging of the MOSFET. Choose a MOSFET that has a high enough power dissipation rating, drain-to-source voltage rating, and drain current rating for the application.
5. V_{SET} is controlled by the DAC63204W to adjust the current output. R_{SET} sets the output range of the current source. Choose a small V_{SET} so that the power dissipation across R_{SET} is minimum. Calculate R_{SET} by:

   $$R_{SET} = \frac{V_{SET}}{I_{OUT}}$$

   A 0.6-V maximum V_{SET} is used in this example. R_{SET} is calculated to be 3 Ω:

   $$R_{SET} = \frac{0.6\,V}{200\,mA} = 3\,\Omega$$

   Choose an R_{SET} resistor with a power rating of at least 120 mW.
6. Choose the smallest reference and gain setting available based on V_{SET} to maximize the effective resolution of the DAC. This application uses the internal 1.21-V reference with the ×1.5 gain setting. The voltage step size of each DAC code (LSB size) for this reference and gain setting and the 12-bit DAC63204W is 443 µV:

   $$LSB = \frac{V_{REF} \times GAIN}{2^N}$$

   $$LSB = \frac{1.21\,V \times 1.5}{2^{12}} = 443\,\mu V$$

   The maximum DAC output voltage used in this application is 0.6 V. The effective resolution is calculated to be about 10 bits:

   $$Effective\ Resolution = log_2\left(\frac{V_{MAX}}{LSB}\right)$$

   $$Effective\ Resolution = log_2\left(\frac{0.6\,V}{443\,\mu V}\right) = 10.4\ bits$$
7. The DAC code for a given output voltage, reference, and gain setting is calculated by:

   $$DAC\_DATA = \frac{V_{OUT} \times 2^N}{V_{REF} \times GAIN}$$

   Where N is the resolution of the DAC. The DAC code for a 0.6-V V_{SET} is:

   $$DAC\_DATA = \frac{0.6\,V \times 2^{12}}{1.21\,V \times 1.5} = 1354d$$

8. The DAC63204W has a programmable slew rate feature. The programmable slew is configured by the CODE-STEP-X and SLEW-RATE-X fields in the DAC-X-FUNC-CONFIG register. The programmable slew is only available when toggling between two values stored in the DAC-X-MARGIN-HIGH and DAC-X-MARGIN-LOW registers.

   CODE-STEP-X defines the number of LSB steps used to transition from the starting code to the final output code. SLEW-RATE-X defines the time-period for each code step. The slew time is calculated by:

   $$t_{SLEW} = SLEW\_RATE \times CEILING\left(\frac{MARGIN\_HIGH - MARGIN\_LOW}{CODE\_STEP} + 1\right)$$

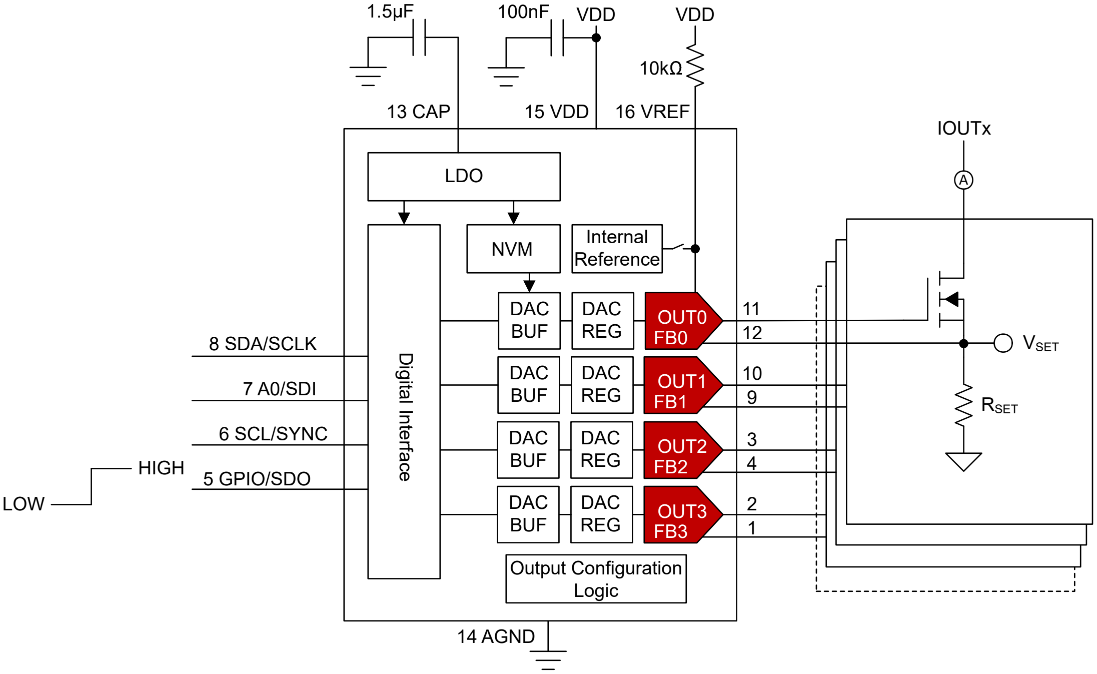
   This application uses a margin high code of 1354, margin low code of 0, SLEW-RATE of 8 µs/LSB and a CODE-STEP of 8 LSB to achieve a 1.36-ms slew time:

   $$t_{SLEW} = 8\left(\mu s/LSB\right) \times CEILING\left(\frac{1354 - 0}{8\ LSB} + 1\right) = 1.368\ ms$$

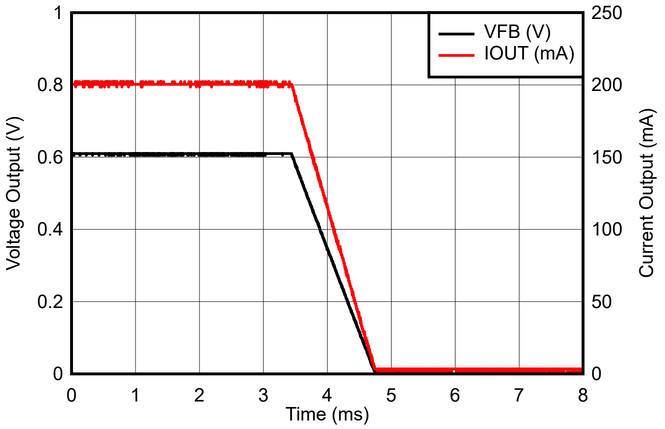
9. The GPIO pin can be configured as a digital input to switch the DAC63204W output between two output voltages with a programmable slew, or power the DAC output on and off. The GPI-EN bit in the GPIO-CONFIG register enables the GPIO pin as an input. The GPI-CH-SEL field selects which channels are controlled by the GPI. The GPI-CONFIG field selects the GPI function. Write 0b1010 to the GPI-CONFIG field to configure the GPIO pin to trigger margin-high or margin-low functions if programmable slew is needed, or write 0b0100 to configure the GPIO to power up or down the output if programmable slew is not needed.
10. The DAC63204W can be programmed with the initial register settings described in the Register Settings section using I^{2}C or SPI. Save the initial register settings 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 Results

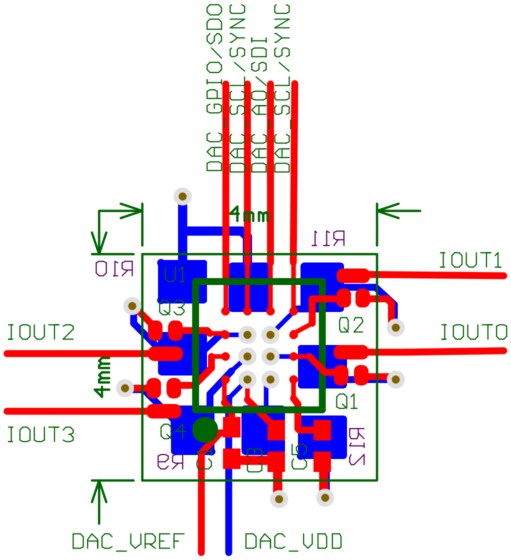
This schematic is used for the following design results of the DAC63204W. The V_{SET} and I_{OUT} signals are measured on an oscilloscope at the test points marked on the schematic.



This plot shows the high to low transition of the DAC63204W output with the 1.36-ms slew configured using the settings discussed in the Design Notes. The V_{SET} voltage measured at the FB0 pin of the DAC63204W slews from 0.6 V to 0 V which causes the output current measured at the drain of the CSD13380F3 to slew from 200 mA to 0 mA.



The following image shows the layout used for this application. The total size is 4 mm × 4 mm including the R_{SET} resistors and MOSFETS required for the high current output stage. The layout consists of the DACx3004W, four FemtoFET MOSFETs, four size-0603 resistors, and three size-0402 capacitors for the VDD and cap pins of the DAC. The R_{SET} resistors used in this example are size-0603 with a power rating of 0.33 W. Larger size resistors can be used to achieve a higher power rating which allows for a higher current output.



Register Settings

The following table shows an example register map for this application. The values given here are for the design choices made in the Design Notes section.

Register Settings for DAC63204W

Register Address	Register Name	Setting	Description
0x1F	COMMON-CONFIG	0x1249	[15] 0b0: Write 0b0 to set the window-comparator output to a non-latching output
			[14] 0b0: Device not locked
			[13] 0b0: Fault-dump read enable at address 0x00
			[12] 0b0: 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: Glitch filter disabled for GP input
			[14] 0b0: Don't care
			[13] 0b0: Disable output mode for 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: GP input configured to trigger margin high or low
			[0] 0b1: Enables input mode for GPIO pin

Register Settings for DAC63204W (continued)

Register Address	Register Name	Setting	Description
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: LDAC is not triggered
			[6] 0b0: DAC clear is not triggered
			[5] 0b0: Don't care
			[4] 0b0: Fault-dump is not triggered
			[3] 0b0: PROTECT function not triggered
			[2] 0b0: Fault-dump read not triggered
			[1] 0b1: Write 0b1 to store applicable register settings to the NVM
			[0] 0b0: NVM reload not triggered. Write 0b1 to reload applicable registers with existing NVM settings
0x03, 0x09, 0x0F, 0x15	DAC-X-VOUT-CMP-CONFIG	0x0800	[15:13] 0b000: Don't care
			[12:10] 0b010: Selects internal reference with ×1.5 gain
			[9:5] 0x00: Don't care
			[4] 0b0: Set OUTx pins as push-pull in comparator mode
			[3] 0b0: Comparator output consumed internally
			[2] 0b0: FBx input has high-impedance in comparator mode
			[1] 0b0: Comparator output not inverted
			[0] 0b0: Disable comparator mode
0x06, 0x0C, 0x12, 0x18	DAC-X-FUNC-CONFIG	0x0052	[15] 0b0: DAC-X clear mode set to zero-scale
			[14] 0b0: DAC-X output updates immediately after a write command
			[13] 0b0: Do not update DAC-X with broadcast command
			[12:11] 0b00: Phase set to 0°
			[10:8] 0b000: Selects sine wave mode
			[7] 0b0: Enable linear slew
			[6:4] 0b101: Selects 8 LSB CODE-STEP
			[3:0] 0x2: Selects 8 µs/step SLEW-RATE
0x01, 0x07, 0x0D, 0x13	DAC-X-MARGIN-HIGH	0x54A0	[15:4] 0x54A: 12-bit margin high code
			[3:0] 0x0: Don't care
0x02, 0x08, 0x0E, 0x14	DAC-X-MARGIN-LOW	0x0000	[15:4] 0x000: 12-bit margin low code
			[3:0] 0x0: Don't care

Pseudocode Example

The following shows a pseudocode sequence to program the initial register values to the NVM of the DAC63204W. The values given here are for the design choices made in the Design Notes section.

Pseudocode Example for DAC63204W

```
//SYNTAX: WRITE <REGISTER NAME (Hex code)>, <MSB DATA>, <LSB DATA>
//Set gain setting to 1.5x internal reference (1.8 V) (repeat for all channels)
WRITE DAC-0-VOUT-CMP-CONFIG(0x3), 0x08, 0x00
//Power-up voltage output on all channels and enable the internal reference
WRITE COMMON-CONFIG(0x1F),0x12, 0x49
//Configure GPI for Margin-High, Low trigger for all channels
WRITE GPIO-CONFIG(0x24), 0x01, 0xF5
//Set slew rate and code step (repeat for all channels)
//CODE_STEP: 8 LSB, SLEW_RATE: 8 µs/step
WRITE DAC-0-FUNC-CONFIG(0x06), 0x00, 0x52
//Write DAC margin high code (repeat for all channels)
//For a 1.8-V output range, the 12-bit hex code for 0.6 V is 0x54A. With 16-bit left alignment,
this becomes 0x54A0
WRITE DAC-0-MARGIN-HIGH(0x01), 0x54, 0xA0
//Write DAC margin low code (repeat for all channels)
//The 12-bit hex code for 0 V is 0x000. With 16-bit left alignment, this
becomes 0x0000
WRITE DAC-0-MARGIN-LOW(0x02), 0x00, 0x00
//Save settings to NVM
WRITE COMMON-TRIGGER(0x20), 0x00, 0x02
```

Design Featured Devices

Device	Key Features	Link
DAC63204W	4-channel, 12-bit, VOUT and IOUT smart DAC with I^2C, SPI, and Hi-Z out during power off in DSBGA package	DAC63204W
DAC53204W	4-channel, 10-bit, VOUT and IOUT smart DAC with I^2C, SPI, and Hi-Z out during power off in DSBGA package	DAC53204W
DAC63004W	4-channel, 12-bit, VOUT and IOUT smart DAC with I^2C, SPI, and Hi-Z out during power off in DSBGA package	DAC63004W
DAC53004W	4-channel, 10-bit, VOUT and IOUT smart DAC with I^2C, SPI, and Hi-Z out during power off in DSBGA package	DAC53004W

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, *DAC63004WCSP-Evaluation Module*
- Texas Instruments, *DAC63004WCSP-EVM User’s Guide*
- Texas Instruments, Precision Labs - DACs

For direct support from TI Engineers, use the E2E community:

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