



Smart DAC

Katlynn Jones

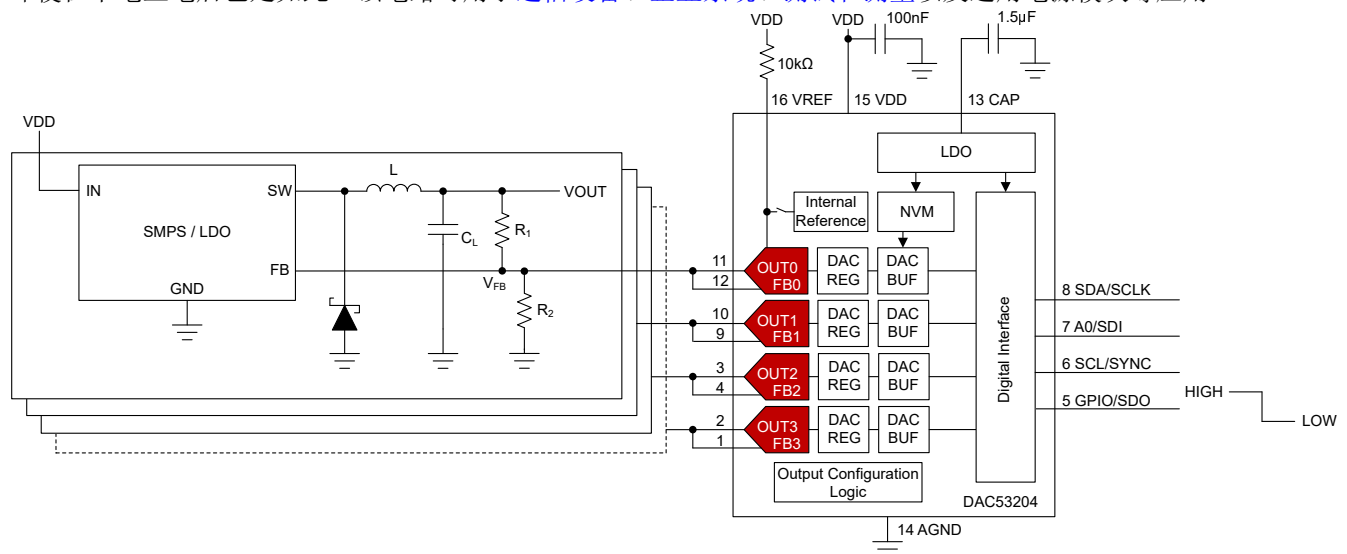
设计目标

主要输入参数	主要输出信号	推荐器件
SPI 或 I ² C 通信	±12μA 模拟 DAC 输出, 3.3V ±10% SMPS 输出	DAC43204、DAC53204、DAC53004

目标：为 SMPS 输出提供 ±10% 标称值的裕量电压。

设计说明

该电路使用四通道缓冲电流输出 DAC 为开关模式电源 (SMPS) 提供电压裕度。电源裕量调节电路用于修整、调节或测试电源转换器的输出。低压降稳压器 (LDO)、直流/直流转换器或 SMPS 等可调节电源提供了反馈 (FB) 输入, 用于控制所需的输出。DAC43204 等精密智能 DAC 可在 DAC 已供电且输出已上电启动时实现对电源输出的线性控制。当 DAC 已供电但输出处于断电模式时, 大多数 DAC 会在电压输出端添加一个内部下拉电阻器。此外, 当大多数 DAC 完全断电时, 如果输出被拉离接地电位, 则输出引脚上的 ESD 单元会传导电流, 电压裕量调节电路中就是如此。当 DAC 断电或输出通道处于断电模式时, DAC43204 提供高阻抗 (Hi-Z) 输出, 这意味着 DAC 通过 SMPS 的 FB 引脚消耗非常小的电流, 且输出设置为标称电压。DAC43204 有一个通用输入 (GPI) 引脚, 此引脚可使 DAC 输出在高电流输出和低电流输出之间切换。这样可在标称输出值的 ±10% 范围内切换 SMPS。可使用 DAC43204 的非易失性存储器 (NVM) 保存所有寄存器设置, 这意味着可在无处理器时使用器件, 即使在下电上电后也是如此。该电路可用于通信设备、企业系统、测试和测量以及通用电源模块等应用。



设计说明

1. **DACx3204 具有自动检测型 I²C、PMBus™ 或 SPI 接口的 12 位、10 位和 8 位四路电压和电流输出智能 DAC** 数据表建议：将 100nF 去耦电容器用于 VDD 引脚，将 1.5μF 或更高的旁路电容器用于 CAP 引脚。CAP 引脚连接至内部 LDO。将这些电容器靠近器件引脚放置。
2. SMPS 的标称电压可通过电阻 R₁ 和 R₂ 设置。SMPS 在 FB 引脚上使用内部 600mV 基准电压来确定输出电压。使用以下公式计算 R₁ 和 R₂：

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

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

使用通过 R₁ 和 R₂ 的 100μA 标称电流以及 3.3V 标称输出电压。代入这些值，公式变为：

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

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

3. 为了实现所需的裕度，DAC43204 必须通过 R₁ 灌入或提供额外的电流。此电流 (I_{MARGIN}) 可通过以下公式计算：

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

对于 ±10% 的裕度，公式变为：

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

4. ±I_{MARGIN} 的 DAC 代码存储在 DAC-MARGIN-HIGH 和 DAC-MARGIN-LOW 寄存器中。使用以下公式计算以十进制形式编程到这些寄存器的代码：

$$DAC_MARGIN_HIGH = \frac{I_{DAC,MAX} - I_{MIN}}{I_{MAX} - I_{MIN}} \times 256$$

$$DAC_MARGIN_LOW = \frac{I_{DAC,MIN} - I_{MIN}}{I_{MAX} - I_{MIN}} \times 256$$

使用 ±25μA 的 I_{OUT} 范围，公式变为：

$$DAC_MARGIN_HIGH = \frac{12\mu A - (-25\mu A)}{25\mu A - (-25\mu A)} \times 256 = 189.44d$$

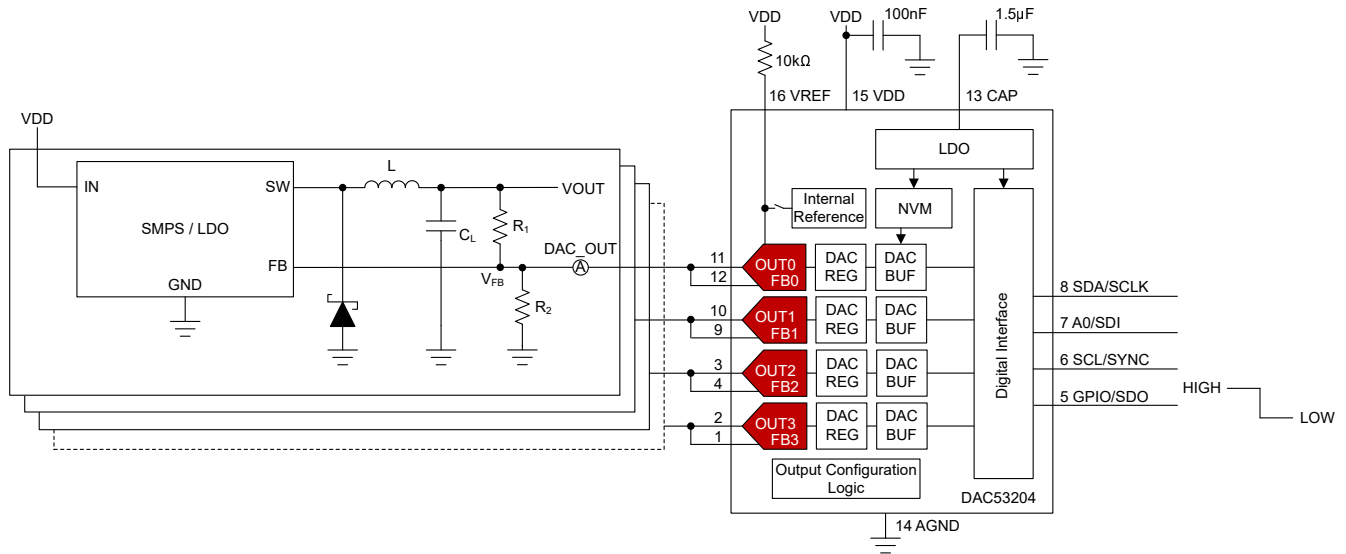
$$DAC_MARGIN_LOW = \frac{-12\mu A - (-25\mu A)}{25\mu A - (-25\mu A)} \times 256 = 66.56d$$

这两个值四舍五入为 189d 和 67d，得出 I_{DAC,MAX} 为 11.9μA，I_{DAC,MIN} 为 -11.9μA。

5. 在此设计中，GPI 用于高/低裕度功能。GPI 上的高电平会将 DAC 输出设置为 I_{DAC,MAX}，将 SMPS V_{OUT} 设置为低裕度 2.97V。GPI 上的低电平会将 DAC 输出设置为 I_{DAC,MIN}，将 SMPS V_{OUT} 设置为高裕度 3.63V。
6. 使用 I²C 或 SPI，根据 **寄存器设置** 部分所述的初始寄存器设置对 DAC43204 进行编程。将初始寄存器设置保存在 NVM 中，方法是将 1 写入 COMMON-TRIGGER 寄存器的 NVM-PROG 字段。对 NVM 进行编程后，器件将在重置或下电上电之后加载具有 NVM 所存储值的所有寄存器。

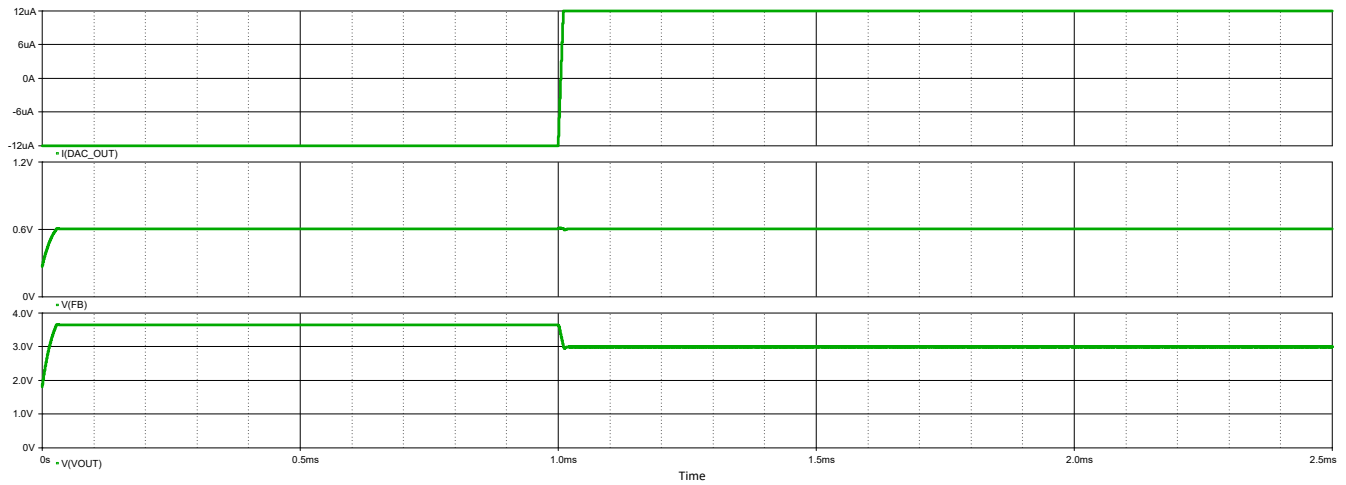
设计仿真

该原理图用于如下 DAC43204 仿真。



瞬态仿真结果

仿真显示了 SMPS 输出 (V_{OUT}) 随 DAC43204 输出 (DAC_OUT) 变化而变化的结果。当 DAC_OUT 为 $I_{DAC,MIN}$ 时, SMPS V_{OUT} 升高裕度 3.63V。当 DAC_OUT 为 $I_{DAC,MAX}$ 时, SMPS V_{OUT} 降低裕度 2.97V。



寄存器设置

DAC43204 电压裕量的寄存器设置

寄存器地址	寄存器名称	设置	说明
0x1F	COMMON-CONFIG	0x1DB6	[15] 0b0 : 写入 0b1, 将窗口比较器输出设置为锁存输出
			[14] 0b0 : 写入 0b1, 锁定器件。将 0b0101 写入 COMMON-TRIGGER 寄存器的 DEV-UNLOCK 字段, 以进行解锁
			[13] 0b0 : 写入 0b1, 在地址 0x01 处设置故障转储读取使能
			[12] 0b1 : 启用内部基准
			[11:10] 0b11 : 将 VOUT3 断电
			[9] 0b0 : 将 IOOUT3 上电
			[8:7] 0b11 : 将 VOUT2 断电
			[6] 0b0 : 将 IOOUT2 上电
			[5:4] 0b11 : 将 VOUT1 断电
			[3] 0b0 : 将 IOOUT1 上电
			[2:1] 0b11 : 将 VOUT0 断电
[0] 0b0 : 将 IOOUT0 上电			
0x24	GPIO-CONFIG	0x01F5	[15] 0b0 : 写入 0b1, 在 GPI 上启用干扰滤波器
			[14] 0b0 : 无关
			[13] 0b0 : 写入 0b1, 在 GPIO 引脚上启用输出模式
			[12:9] 0b0000 : 选择映射到 GPIO 作为输出的 STATUS 功能设置
			[8:5] 0b1111 : 在所有通道上启用 GPI 功能
			[4:1] 0b1010 : 选择 GPI, 触发高/低裕度
[0] 0b1 : 启用 GPIO 引脚的输入模式			
0x20	COMMON-TRIGGER	0x0002	[15:12] 0b0000 : 写入 0b0101, 解锁器件
			[11:8] 0b0000 : 写入 0b1010, 触发 POR 复位
			[7] 0b0 : 如 DAC-X-FUNC-CONFIG 寄存器中相应 SYNC-CONFIG-X 位为 1, 则写入 0b1, 触发 LDAC 运行
			[6] 0b0 : 写入 0b1, 基于 DAC-X-FUNC-CONFIG 寄存器中相应 CLR-SEL-X 位, 将 DAC 寄存器和输出设置为零代码或中间代码
			[5] 0b0 : 无关
			[4] 0b0 : 写入 0b1, 触发故障转储序列
			[3] 0b0 : 写入 0b1, 触发 PROTECT 功能
			[2] 0b0 : 写入 0b1, 读取 NVM 的一行进行故障转储
[1] 0b1 : 写入 0b1, 将适用寄存器设置存储到 NVM			
[0] 0b0 : 写入 0b1, 使用现有 NVM 设置重新加载适用寄存器			
0x01、0x07、0x0D、0x13	DAC-X-MARGIN-HIGH	0xBD00	[15:8] 0xBD : 8 位数据更新 MARGIN-HIGH 代码
			[7:0] 0x00 : 无关
0x02、0x08、0x0E、0x14	DAC-X-MARGIN-LOW	0x4300	[15:8] 0x43 : 8 位数据更新 MARGIN-LOW 代码
			[5:0] 0x00 : 无关

伪代码示例

下面所示为将初始寄存器值编程到 DAC43204 的 NVM 的伪代码序列。此处给出的值基于在 [设计注意事项](#) 中所做的设计选择。

GPI 至 PWM 的伪代码示例

```
//SYNTAX: WRITE <REGISTER NAME (Hex code)>, <MSB
DATA>, <LSB DATA> //Power-up current output on all channels, enables
internal reference WRITE COMMON-CONFIG(0x1F), 0x1D, 0xB6 //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), 0xBD, 0x00
//write DAC margin low code (repeat for all channels) WRITE DAC-0-MARGIN-LOW(0x02),
0x43, 0x00 //save settings to NVM WRITE COMMON-TRIGGER(0x20), 0x00, 0x02
```

设计中采用的器件

器件	主要特性	链路
DAC43204	具有 I ² C、SPI 且在断电期间处于高阻态输出的 4 通道 8 位 VOUT 和 IOUT 智能 DAC	www.ti.com.cn/product/cn/DAC43204
DAC53204	具有 I ² C、SPI 且在断电期间处于高阻态输出的 4 通道 10 位 VOUT 和 IOUT 智能 DAC	www.ti.com.cn/product/cn/DAC53204
DAC53004	具有 I ² C、SPI 且在断电期间处于高阻态输出的超低功耗 4 通道 10 位 VOUT 和 IOUT 智能 DAC	www.ti.com.cn/product/cn/DAC53004

使用 [参数搜索工具](#) 查找其他可能的器件。

设计参考资料

有关 TI 综合电路库的信息，请参阅 [模拟工程师电路手册](#)。

其他资源

- 德州仪器 (TI), [智能 DAC Python 示例](#)
- 德州仪器 (TI), [DAC63204 评估模块](#)
- 德州仪器 (TI), [DAC63204 EVM 用户指南](#)
- 德州仪器 (TI), [高精度实验室 - DAC](#)

如需 TI 工程师的直接支持，请使用 **E2E™** 论坛：

e2e.ti.com

重要声明和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做任何明示或暗示的担保，包括但不限于对适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他功能安全、信息安全、监管或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的应用。严禁对这些资源进行其他复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，TI 对此概不负责。

TI 提供的产品受 [TI 的销售条款](#) 或 [ti.com](#) 上其他适用条款/TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

邮寄地址：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265

Copyright © 2024，德州仪器 (TI) 公司