Application Note

LCD Digital VCOM Using the DAC43701

Dylan Zheng

ABSTRACT

In the LCD, the panel's common plane voltage (VCOM) is often used to maximize image quality and provide hybrid vision. It can be challenging for an integrated VCOM chip to meet the increasing customized requirements, or it needs additional cost. To address these requirements, this application note outlines how to design a digital VCOM solution using the DAC43701. The operation principle of a digital VCOM circuit is introduced, and a method of updating NVM is presented to conveniently modify the DAC's default output state and minimize panel's flicker. Finally, an application example is used to demonstrate these concepts.

Table of Contents

List of Figures

List of Tables

Trademarks

All trademarks are the property of their respective owners.

1 Introduction

LCD panels are widely used in notebooks, TV, automotive display and so on. A typical LCD panel subsystem diagram block is shown in Figure 1-1. VCOM is the reference voltage for each pixel in the LCD panel. VCOM is typically used to maximize contrast and image quality during operation. A VCOM with different amplitudes and frequencies are needed to achieve different LCD electrical characteristics, optical characteristics and functional requirements.

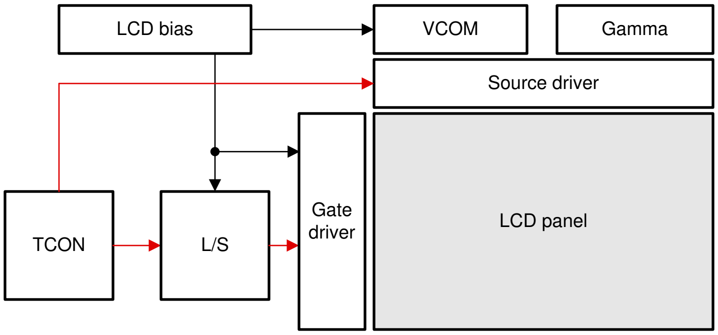


Figure 1-1. Typical LCD Panel Subsystem Diagram Block

Figure 1-2 shows the LCD digital VCOM circuit using the DAC43701. It consists primarily of a DAC43701 and inverting summer circuit. The DAC output can be dynamically adjusted by I2C, and the inverting summer circuit outputs the required VCOM with specific gain and offset.

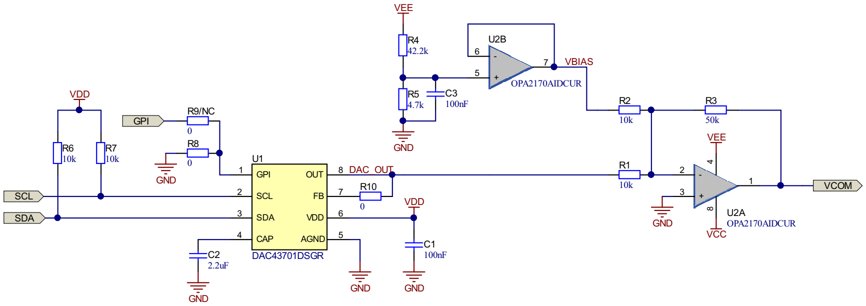


Figure 1-2. LCD Digital VCOM Using the DAC43701

The DAC43701 uses the power supply (VDD) as a reference by default. It also contains an internal reference that can be enabled by writing a 1 to the REF_EN bit in the GENERAL_CONFIG register (address D1h). The DAC transfer function when VDD is used as reference is given in Equation 1.

$$V_{DAC\_OUT} = \frac{DAC\_DATA}{2^N} \times V_{DD} \tag{1}$$

Where, N is the DAC resolution, either 8-bits (DAC43701) or 10-bits (DAC53701). DAC_DATA is the decimal equivalent of the data in the DAC_DATA register (address 21h), ranges from 0 to 2^N-1.

The transfer function of the inverting summer circuit is given in Equation 2.

$$V_{COM} = -\frac{R_3}{R_1} \times V_{DAC\_OUT} - \frac{R_3}{R_2} \times V_{BIAS} = -\frac{R_3}{R_1} \times V_{DAC\_OUT} - \frac{R_3}{R_2} \times \left(\frac{R_5}{R_4 + R_5} \times V_{EE}\right) \tag{2}$$

Assuming R_1 and R_2 are equal, Equation 1 and Equation 2 became Equation 3

$$V_{COM} = -\frac{R_3}{R_1} \times \left(\frac{DAC\_DATA}{2^N} \times V_{DD} + V_{BIAS}\right) \tag{3}$$

Thus, different amplitudes and frequencies of VCOM can be controlled by a dedicated DAC_DATA value, which is configured in real time by I2C. Figure 1-3 shows the VCOM voltage characteristic when VDD is 5 V.

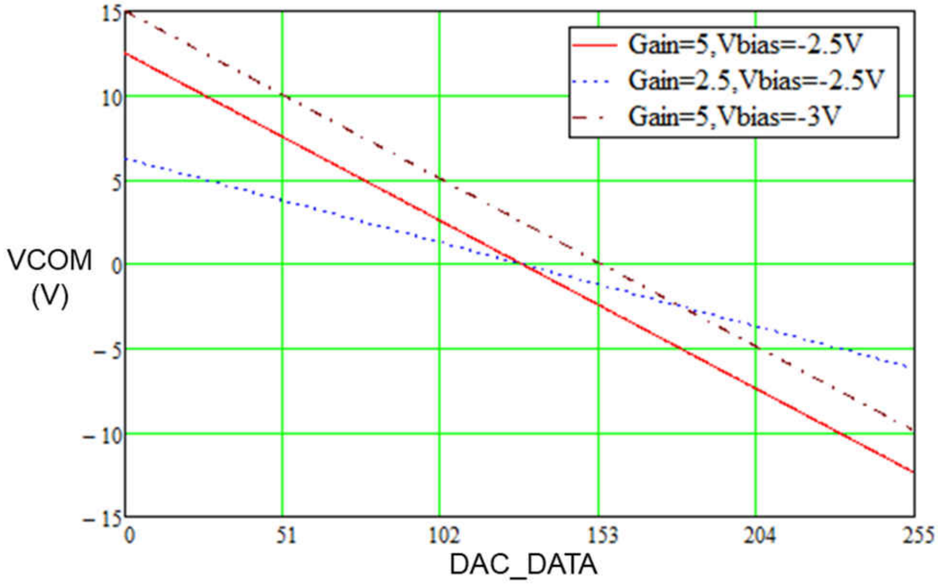


Figure 1-3. LCD Digital VCOM Voltage Characteristic

2 NVM Online Update to Minimize Flicker

After the VDD supply has been established, and before the communication between the MCU and DAC43701 is valid, the default state of DAC_OUT is powered down to high impedance. During this time,the VCOM will be $-\frac{R_3}{R_1} \times V_{BIAS}$. If this duration is too long, it will cause a visible flicker at the LCD panel. When the system powers up, VCOM should be set to 0V as soon as possible to minimize flicker. Therefore, the DAC_OUT default state should be modified, which can be done by updating the NVM of the DAC43701.

A common method of programming the NVM is by monolithic flash in the factory. This method can require additional customized tools and mass production procedures. In addition, if the DAC43701 needs to be replaced, the new device will need to be reprogrammed which can be an inconvenience for subsequent maintenance. If the application uses an MCU or other I2C masters, there is a simpler method available to program the NVM. This method to update the NVM with the LCD system's existing MCU (NVM online update) is described below.

As shown in Equation 3 the DAC_DATA register should be set to $hex\left(\frac{-V_{BIAS}}{V_{DD}} \times 2^N\right)$ to set VCOM to 0V. And the DAC_PDN bits in the GENERAL_CONFIG register (address = D1h) should be 00b to enable the DAC output. Figure 2-1 shows the flow of the NVM online update.

1. After each POR delay, read back register 0x21 and 0xD1.
2. Compare the read back values with the target values. Execute steps 3-5 if the values DO NOT meet the above target values.
3. Update DAC_DATA(address = 21h) to customized value.
4. Write 00b to DAC_PDN (address = D1h) to power up the DAC output.
5. Assert NVM_PROG (address = D3h) to initiate NVM write.
6. After programing and a power cycle, the DAC_OUT will have the target default state.

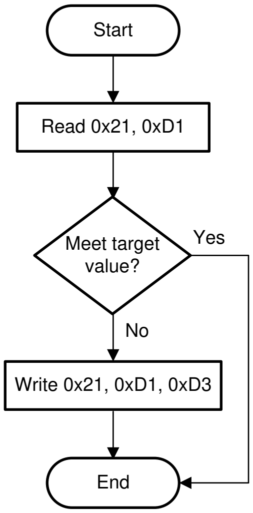


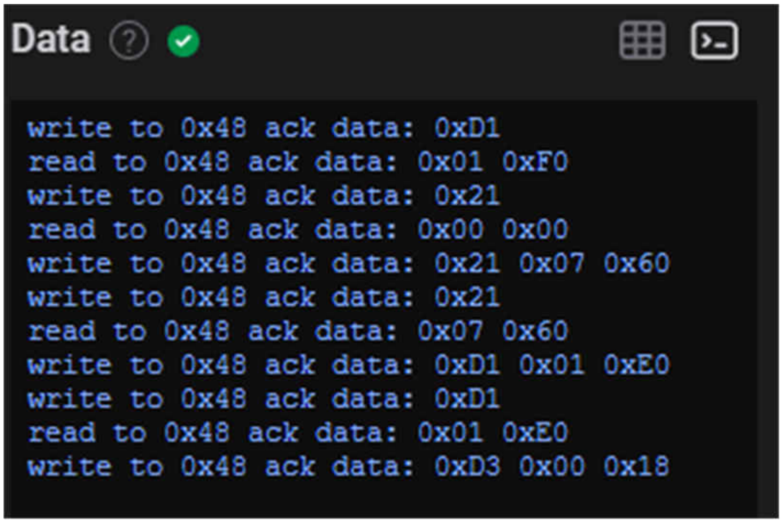
Figure 2-1. Work Flow of the NVM Online Update

With the above flow, the MCU will automatically update the NVM in situations where there is a deviation from the target value such as replacing the DAC43701 with a new device, or an unsuccessful initialization caused by noise.

The following shows a pseudo code example of Figure 2-1 to program the initial register values to the NVM. The users can modify it for their dedicated usages. Figure 2-2 gives the I2C log when the target value of DAC_DATA (address = 21h) is 0x0760, and GENERAL_CONFIG (address = D1h) is 0x01E0.

Pseudo Code Example for NVM Online Update

```
//SYNTAX:WRITE <REGISTER NAME (REGISTER ADDRESS)>, <MSB DATA>, <LSB DATA>
//SYNTAX:READ <REGISTER NAME (REGISTER ADDRESS)>, <MSB DATA>, <LSB DATA>
READ DAC_DATA(0x21)
READ GENERAL_CONFIG (0xD1)
//Whether 'target value' or not
IF(0xD1=01E0h & 0x21= $hex\left(\frac{-V_{BIAS}}{V_{DD}} \times 2^{N}\right)$ )
{}
ELSE
{
//Write DAC code (12-bit aligned)
WRITE DAC_DATA(0x21), $hex\left(\frac{-V_{BIAS}}{V_{DD}} \times 2^{N}\right)$
//Enable output, DAC_PDN=00,
WRITE GENERAL_CONFIG (0xD1), 0x01, 0xE0
//Program the NVM
WRITE TRIGGER(0xD3), 0x00, 0x18
//DAC output will go to target voltage at power up, and will have target output voltage next POR.
}
```



```
write to 0x48 ack data: 0xD1
read to 0x48 ack data: 0x01 0xF0
write to 0x48 ack data: 0x21
read to 0x48 ack data: 0x00 0x00
write to 0x48 ack data: 0x21 0x07 0x60
write to 0x48 ack data: 0x21
read to 0x48 ack data: 0x07 0x60
write to 0x48 ack data: 0xD1 0x01 0xE0
write to 0x48 ack data: 0xD1
read to 0x48 ack data: 0x01 0xE0
write to 0x48 ack data: 0xD3 0x00 0x18
```

Figure 2-2. I2C Log of the Work Flow

3 Application Example

This section gives short design notes for generating a square wave VCOM using the circuit shown in Figure 1-2. The design goals are shown in Table 3-1.

Table 3-1. Design Goals

VDAC-	VDAC+	VCOM-	VCOM+	fsquare	VCC	VEE	VDD
0.2V	2.2V	-5V	5V	140Hz	12V	-12V	2.6V

3.1 Design Note

For this design, the gain required should be:

$$G_V = \left|-\frac{R_3}{R_1}\right| = \left|\frac{V_{COM+} - V_{COM-}}{V_{DAC+} - V_{DAC-}}\right| = 5 \tag{4}$$

And the bias voltage should be:

$$V_{BIAS} = \frac{R_5}{R_4 + R_5} \times V_{EE} = -\frac{V_{DAC+} + V_{DAC-}}{2} = -1.2V \tag{5}$$

Select reasonable resistance values for R_3 and R_5

$$R_3 = 50k\Omega,\ R_5 = 4.7k\Omega \tag{6}$$

From Equation 4 through Equation 6, the required R_1 and R_4 can be calculated.

$$R_1 = R_2 = 10k\Omega,\ R_4 \approx 42.2k\Omega \tag{7}$$

Some parameters should be checked to make the circuits stable and robust.

Calculate the small signal circuit bandwidth using Equation 8 to ensure it is greater than the 140Hz requirement.

$$BW = \frac{GB_{OPA2170}}{NG} = \frac{1.2MHz}{1 + \frac{R_3}{R_1||R_2}} = 109kHz \tag{8}$$

Where, NG is the noise gain of the inverting summer circuit.

Calculate the minimum slew rate required using Equation 9.

$$SR > 2 \times \pi \times f_{square} \times V_{COM+} = 4.4mV/\mu s \tag{9}$$

The slew rate of OPA2170 is 0.4V/µs which meets the requirements.

For better stability, the zero created by the gain setting resistors and input capacitance of the OPA2170 should be greater than the bandwidth calculated in Equation 8.

$$f_{zero} = \frac{1}{2 \times \pi \times \left(C_{cm} + C_{diff}\right) \times \left(R_1||R_2||R_3\right)} = 5.8MHz > \ = 109kHz \tag{10}$$

Where, C_{cm} and C_{diff} are the common-mode input capacitance and differential input capacitance of the OPA2170 respectively.

3.2 Optimize Power Sequence

In addition, the common-mode voltage range should be considered to avoid the amplifier working in a nonlinear state. The common-mode voltage of the circuit in Figure 1-2 is 0V, and the common-mode voltage range of the OPA2170 is from (VEE-0.1V) to (VCC-2V). As the bias voltage is divided from VEE, if the VEE supply is applied earlier than VCC, the clipping voltage will be observed at the VCOM output as shown in Figure 3-1, which may cause unexpected flicker.

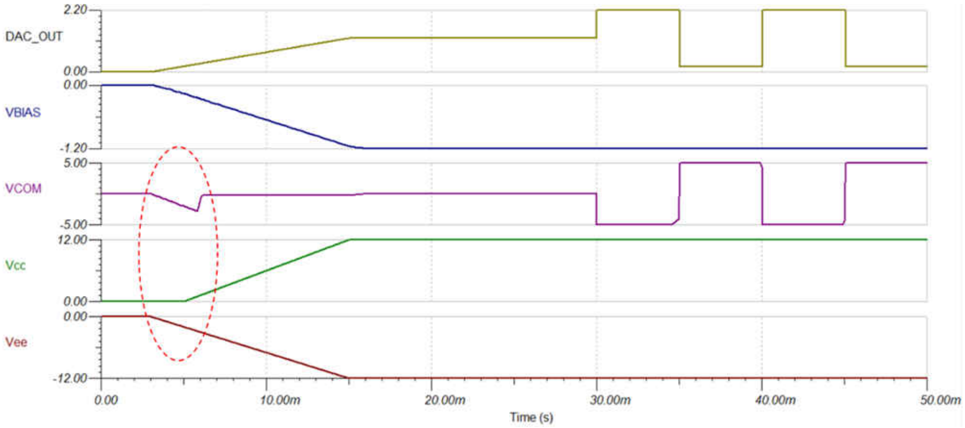


Figure 3-1. Transient Simulation Results When VEE Rises up Earlier

To avoid the clipping issue, VCC should power on before VEE, and ensure VCC is greater than 2 V(typical) when VEE powers on, as shown in Figure 3-2.

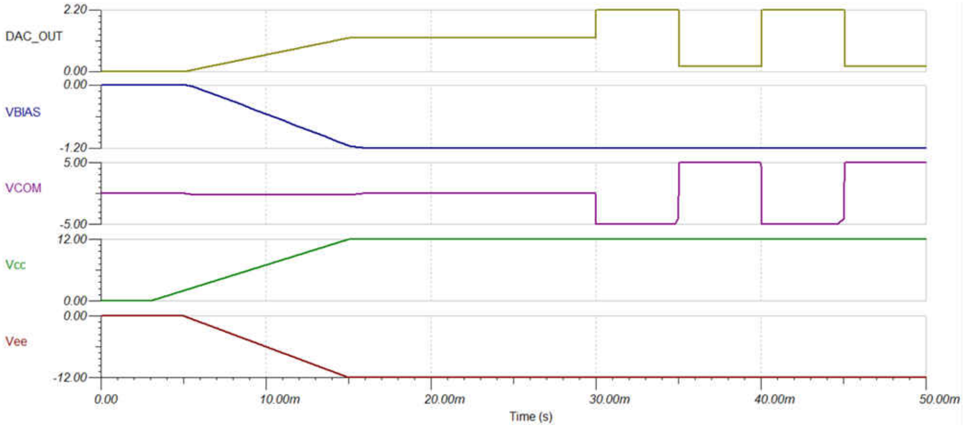


Figure 3-2. Transient Simulation Results When VCC Rises up Earlier

3.3 DAC Pseudo Code Example

This section gives a pseudo code example for the design goals in Table 3-1. From Equation 1, the DAC_DATA register should be 14h and D9h for VDAC- and VDAC+ respectively. To generate the customized square wave, MCU can update the DAC_DATA at a frequency of 140Hz through I2C.

Pseudo Code Example for Programmable Square Wave VCOM

```
//SYNTAX:WRITE <REGISTER NAME (REGISTER ADDRESS)>, <MSB DATA>, <LSB DATA>
//Enable output, use VDD as DAC reference,
WRITE GENERAL_CONFIG (0xD1), 0x01, 0xE0
WHILE ()
{
//Write DAC code (12-bit aligned), For 2.6-V VDD as DAC reference, the 8-bit hex code for 0.2 V is 0x14
//Left aligned this becomes 0x0140
WRITE DAC_DATA(0x21), 0x01, 0x40
//Duration is 1/280 second
DELAY(1/280)
//Write DAC code (12-bit aligned), For 2.6-V VDD as DAC reference, the 8-bit hex code for 2.2 V is 0xD9
//Left aligned this becomes 0x0D90
WRITE DAC_DATA(0x21), 0x0D, 0x90
//Duration is 1/280 second
DELAY(1/280)
}
```

Figure 3-3 shows the test results

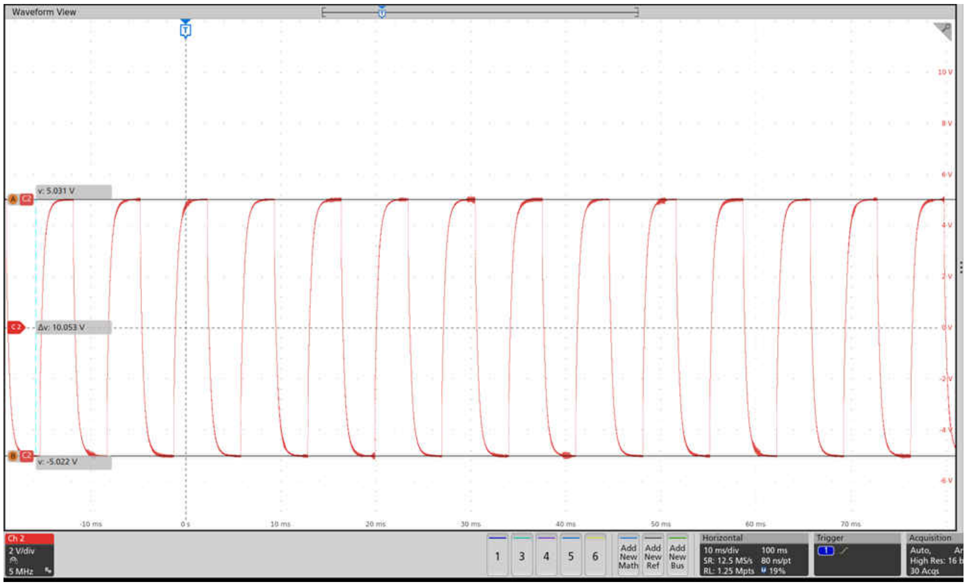


Figure 3-3. VCOM Test Result

4 Summary

This application note described a digital VCOM circuit using a smart DAC and its NVM online update solution. It is important to consider the DAC default output and circuit power sequence to maximize LCD image quality. An application example is given to validate the performance and help users change to suit their applications.

5 References

1. Texas Instruments, DAC43701 data sheet.
2. Texas Instruments, DAC53701 data sheet.
3. Texas Instruments, Analog Engineer's Circuit Cookbook: Amplifiers.
4. Texas Instruments, OPA2170 data sheet.

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