A 14-Bit Linear DAC for VCM Driver

Kuok Ling, Chong Chi[+], Fred Trafton and Carlos Brito
Texas Instruments, Inc.
8360 LBJ Freeway
Dallas, Texas 75243, USA
[+]Cadence Design Systems, Inc.
5215 N. O'Connor, #1000
Irving, Texas 75039, USA

OVERVIEW

A conventional Hard Disk Drive (HDD) has two separate motors. The first is a Spindle Motor (SPM) which rotates the disk at high velocities (typically 5400-7200 RPM, though high-end HDD's in the 10-12K RPM range are currently available or under development). The second motor is called the Positioner or Voice Coil Motor (VCM). This motor moves the head assembly across the rotating disk to access individual packets of data written on the disk's magnetic media. This paper deals with the control mechanisms for the VCM, and specifically with a 14-bit DAC architecture used as the control mechanism for the VCM.

Data is written on the surface of the disk in concentric circles known as *tracks*. In order to read or write data at the disk's surface, the magnetic read/write head, located at the end of an arm driven by the VCM, must be positioned with high accuracy directly over the track requested by the computer's operating system. Obviously, the more tracks one can put onto the disk, the more data can be stored in the same area. For this reason, all HDD companies try to increase their Tracks Per Inch (TPI). A typical TPI today would be 10-15K TPI, though drives are currently in development which push this number much higher. The higher the TPI, the more accurately the VCM must be able to position the magnetic read/write head.

In order to do this, the VCM control loop must have two distinctly different functions:

1. **Seek** – in Seek mode, the VCM moves from one concentric data track to another. These tracks may be adjacent, or clear across the disk from one another. Special track numbering schemes have been devised which allow the VCM control loop to get feedback on its position relative to the disk's surface even while it is moving. The average time it takes to move from one track to another is one of the main operating parameters of the HDD, and is called the *seek time*. Optimizing this value is essential to be competitive.
2. **Track Follow** – in Track Follow mode, the VCM reads special magnetic markings on the disk surface known as *servo marks*. These marks are physically located on and near the data tracks to allow a position error signal to be developed. The system can tell how close the magnetic head is to dead-on-track from the information derived from these marks. The Track Follow loop attempts to servo the read head to being directly over the data as the disk spins underneath it.

The VCM control loop to perform both of these functions is depicted in Figure 1. A primary loop reads the servo marks from the disk surface, demodulates them, and supplies position error information to a Digital Signal Processor (DSP) IC. The DSP then calculates the current required

by the VCM to compensate for its position error, and communicates this information to the Servo IC via a Serial Port. The Servo IC's VCM driver is a transconductance loop which sets the VCM motor current to a value proportional to the Digital to Analog Converter (DAC) code. This is done by measuring the motor current indirectly from the voltage drop across a sense resistor, and then setting the voltage across the motor to the value required to achieve the desired current.

It is obvious, then, that the higher the track pitch, the higher the resolution of the positioner must be. At current TPI levels, the required resolution is 14 bits. This resolution can be achieved in several ways, including Pulse Width Modulation (PWM) DAC's, Sigma-Delta modulators, or Linear DAC's. For ease of programming and lower noise consideration, we have chosen a 14-bit Linear DAC. The specifications on this DAC call for guaranteed monotonicity, since it is in a feedback loop. However, absolute accuracy (Integral Non-Linearity or INL) is not so critical in this application since small non-linearities can be removed by the control loop. Therefore, while we are required to have 14 bits of Differential Non-Linearity (DNL) for this DAC, the requirement for INL is only equivalent to 10-bit level. This allowed us to make some trade-offs in the design in order to build an untrimmed on-chip 14-bit DAC.

Apart from the DNL and INL requirements, small die size, low current consumption, high speed of operation for a given power and small mid-scale error (the VCM loop servos around mid-scale during Track Follow mode) are also important for the DAC. The Linear VCM DAC design presented here accomplishes all the above goals: 14-bit resolution (DNL) with guaranteed monotonicity, 11-bit accuracy (versus 10-bit requirement), small die size (1300 sq. mils in a 1-μm, double-metal, double-POLY, Linear BiCMOS process), low current consumption (< 120 μA), high conversion speed (< 10 μs), and mid-scale error of 0.5 LSB (61 μV).

14-BIT LINEAR VCM DAC IMPLEMENTATION

The architecture chosen is the well-known dual resistor strings DAC [1], which is highly area-efficient when the resolution of the DAC is high; i.e. 10 bits and above. A dual resistor strings DAC consists of a coarse (MSB) resistor string and a fine (LSB) resistor string. Monotonicity, which is a requirement for the VCM DAC, is guaranteed in this type of DAC's since none of the resistors can have negative resistance. However, achieving accuracy above 11 bits without trimming is difficult with current process technologies.

A block diagram of the 14-bit linear DAC for the VCM driver is presented in Figure 2. In order to reduce power consumption and silicon area, all the circuits are powered by a 5V supply, except amplifier A2, which is powered by the 12V VCM motor supply (VM). Amplifier A1, together with the reference voltage Vref, establishes the full-scale range for the dual resistor strings DAC by feeding back the mid-scale voltage (mid-point of the course resistor string) to the negative input terminal of A1. Amplifier A2 shifts the intermediate output voltage of the DAC, Vo, up by the amount of VM/2, and performs a buffering function so that the DAC can drive a resistive load in the following stage. The voltage level-shift is needed because the reference voltage for the rest of the circuits in the VCM control loop is VM/2. There is an offset error in this DAC implementation, which comes from the input offset of amplifier A2; however, it is common for most VCM DAC designs due to the fact that they have to have a buffer amplifier in order to drive resistive load while maintaining high accuracy. The input offset of amplifier A1 only contributes to error in the full-scale range of the DAC, which is not very critical as long as

the minimum full-scale range can still command the maximum VCM motor current needed during Seek mode. The equation for the VCM DAC output voltage, VDACOUT, is as follows:

$$VDACOUT = Vo – VMID + VM/2,$$

where VMID is the mid-scale voltage of the course resistor string. The VCM DAC mid-scale voltage should be VM/2 ideally so that the motor current will be zero, and be of opposite polarities but with the same range above and below mid-scale. Thus, the positioner arm can be commanded by the DAC to move from the inner tracks to the outer tracks and vice versa. The voltage difference between Vo and VMID is only 0.5 LSB at mid-scale, therefore, VDACOUT will be equal to VM/2 plus 0.5 LSB (assuming amplifier A2 has zero input offset); refer to Figures 2 and 3 and explanations given later on in this article.

The theory of operation for a dual resistor strings DAC is fairly straightforward. The MSB input bits (N/2 bits normally, where N is the total number of bits of resolution) chooses the correct resistor along the coarse string and positions the fine string across the chosen resistor, where the remaining input bits (LSB) locate the output voltage along the fine resistor string. The INL of any resistor string DAC is mainly dependent on how well the process technology controls the matching of widely-separated resistor, and it is difficult to achieve accuracy above 11 bits at the present. On the other hand, the DNL of any resistor string DAC depends on the closely-placed resistor matching capability of the process technology and how well the layout of the resistors is carried out. Hence, when the decoding circuits are minimized and simplified, the resistors can be fitted in a smaller area, which enhances matching on top of saving silicon area.

An 8-bit version of the 14-bit dual resistor strings DAC is shown in Figure 3 for illustration purposes. The coarse resistor string has 16 resistors and is on the left hand side while the fine string is on the right with 15 resistors; the sixteenth resistor is split equally between the switch pair on the top and the switch pair at the bottom. The switch resistance of half the value of a unit fine string resistor is the reason why there is a mid-scale error of 0.5 LSB, however, it is also a way to guarantee 14-bit DNL in this architecture. The coarse string decoding circuits select the correct resistor along the coarse string, and connect the fine string across it for a given DAC input code. One of the ways to achieve small die size while getting high number of bits of resolution is to minimize the decoding circuits for the interface between the two resistor strings. This design accomplishes the decoding of the coarse string by partitioning it into two identical interleaving strings and manipulating the MSB control bits. By creating two identical interleaving strings, one less bit can be used to decode the coarse string. When the LSB (D0) of the coarse string control bits is a one, one is added to the three MSB control bits (D3D2D1), and the resulting bits (B3B2B1) are used to control the set of tap points along the coarse string that get multiplexed to the node V1. On the other hand, when it is a zero, the control bits are unchanged; i.e. (B3B2B1) is the same as (D3D2D1). The 3-bit add-one circuit is shown in Figure 4, while the 3-input multiplexer circuit is shown in Figure 5. This add-one scheme works fine except at the top of the coarse string where overflow occurs; thus, the top and the bottom tap points have to be decoded separately to avoid ambiguity. The decoding circuits for the top and the bottom switches of the coarse resistor string are given in Figure 6 (a) and 6 (b), respectively. The decoding circuits in Figure 6 prevent the bottom tap point to be selected when the three MSB control bits are all 1’s. The control bits for the other interleaved string along the coarse

string are always unchanged, and the set of points they (D3D2D1) control get multiplexed to the node V2.

A two-step decoding scheme is used to multiplex the tap points along both the coarse and the fine resistor strings to the respective outputs. Hence, only two series switches are needed to decode each string, which optimizes the trade-off between the number of switches needed (which affects the conversion speed) and the complexity and die size for the decoding circuits. In the two-step decoding scheme, the resistor string is first partitioned into 2^M blocks with 2^{N-M} resistors each, where M is the number of higher order bits; typically, N/2 (N is the total number of input bits) is chosen. One of the 2^M blocks is first selected with the M input bits for further decoding. The remaining input bits (N-M) are then used to locate the output in the chosen block among the 2^M blocks.

In order to improve the INL, each switch pair along the course string are scaled such that they give a total resistance that is equivalent to half the resistance of a unit resistor along the fine string. The switches must be scaled because the gate drive (Vgs) reduces from the bottom of the course resistor string to the top and the body effect increases the same way, therefore, the switch sizes increase from the bottom of the string to the top. All the decoding circuits are rather simple and small. Table 1 summarizes the total decoding circuits needed to realize the 14-bit DAC.

Switches	**291**
3-bit decoders	**3**
4-bit decoder	**1**
6-input multiplexer	**1**
6-bit add-one circuit	**1**
Glue logic gates (equivalent to a 2-input NAND each)	**9**

Table 1. Total decoding circuits used in the 14-bit VCM DAC.

Comparing the decoding circuits (Figures 4 to 6) for the 8-bit example to that of the 14-bit version in Table 1, it is clear that the six bits increase in resolution only resulted in very minimal increase in the size of the decoding circuits. The number of switches utilized to multiplex out the DAC voltage is quite critical also since more switches mean more die area and routing overhead.

MEASUREMENT RESULTS

Monte-Carlo simulations were run to obtain worst-case performance for the 14-bit Linear VCM DAC and the results are shown in Table 2 along with the measured data. The actual performance of the DAC comes very close to the simulated results. Figure 7 shows the measured DNL and INL data for a typical sample. As can be seen from the data, DNL achieved is less than 1/2 LSB, and the INL is better than 6 LSB; i.e. the DAC achieves 14 bits of resolution and 11 bits of accuracy, respectively. The current consumption for the amplifier A2 is excluded from the total current consumption of the VCM DAC shown in Table 2 due to the fact that it depends on the resistive load that it has to drive and it is not an intrinsic part of the DAC. The performance of the entire VCM control loop with the 14-bit Linear DAC included has also been verified to meet

all specifications. In addition, the yield of the VCM DAC has also being very good especially for a circuit with such high resolution and accuracy.

Performance	Simulated	Measured
DNL	**0.35 LSB**	**0.4 LSB**
INL	**6 LSB**	**6 LSB**
Current Consumption	**120 μA**	**100 μA**
Conversion speed	**8 μs**	**8 μs**

Table 2. Simulated (worst-case) vs. measured performance.

CONCLUSIONS

A general operational description of the Spindle Motor and the Voice Coil Motor in an HDD has been presented, particularly the electronics utilized to move the read/write head via the VCM control loop. The requirements imposed on the VCM DAC, due to the ever-increasing density (TPI) of an HDD, have also been explained. In addition, the actual circuit implementation of the 14-bit Linear VCM DAC has been described in detail, followed by the comparisons between the simulated worst-case results and the measured data. The die size consumed is slightly less than 1300 sq. mils in a 1.0-μm linear BiCMOS process technology. The current consumption for the DAC is 100 μA nominally while achieving conversion speed of 8 μs. The DNL and INL achieved are 0.4 LSB and 6 LSB, respectively, which is equivalent to achieving 14 bits of resolution and 11 bits of accuracy. The 14-bit Linear VCM DAC has also been confirmed to work well as an integral part in a VCM control loop for a state-of-the-art HDD system.

REFERENCES

[1] M. Tuthill, "A 16 Bit Monolithic CMOS D/A Converter," in ESSCIRC Dig. Tech. Papers, pp 353-355, Sept. 1980.

[2] Hans-Ulrich Post and Karl Schoppe, "A 14 Bit Monotonic NMOS D/A Converter," IEEE J. of Solid-State Circuits, VOL. SC-18, N0.3, June 1983.