ANALOG DEVICES

AN-777
APPLICATION NOTE

One Technology Way • P.O. Box 9106 • Norwood, MA 02062-9106 • Tel: 781/329-4700 • Fax: 781/461-3113 • www.analog.com

Dual AD9884A Design Guideline to Achieve UXGA Resolutions

by Del Jones

INTRODUCTION

Pixel clock speeds in excess of 140 MHz can be achieved with the AD9884A by using a dual chip "ping-pong" configuration. A dual chip solution is different from an alternate pixel sampling solution in that full refresh rates can be maintained.

There are many ways to implement a dual AD9884A design. This application note serves to make the user aware of the considerations that should be weighed when implementing this ping-pong configuration. Among the variables are layout and routing constraints, clock selection, graphics controller requirements, and maximum speed requirements.

Analog Input Layout and Routing

When laying out and routing the analog inputs (R, G, B, and HSYNC), several factors should be considered. The trace lengths of the R, G, and B inputs should be kept as equal as possible, while also keeping the routes direct (no zig-zagging) to maintain equal propagation delays. The branches to each of the AD9884's analog inputs should be kept as short as possible. The 75 Ω terminators on the RGB inputs should be placed as close to the branch junctions as possible. Finally, each R, G, and B branch requires its own coupling capacitor. These considerations are illustrated in Figure 1.

Clock Source Selection

There are three methods that can be used for clocking data. An external clock source can be used to clock both AD9884As as well as data-latching devices (graphics controllers). This method requires external PLL circuitry and special high speed clock layout and routing considerations.

Another option is to use the PLL in chip 1 to drive chip 2. This method would require chip 2 to be configured for external clock operation, using the negative edge of chip 1's DATACK to sample the RGB data. This method employs the most direct routing of HSYNC. The HSYNC could be routed directly to chip 1, then routed to the second device. (Although the second device does not use HSYNC to generate a clock, it is still needed to provide a timing reference for other functions, such as clamping.) The problem with this option is that it causes difficulty in setting the correct clock phase of chip 2 because of the added propagation delay between HSYNC and chip 1's data clock output. It can also cause ongoing clock phase difficulty in chip 2 because of the variability of chip 1's data clock propagation delay over time and temperature.

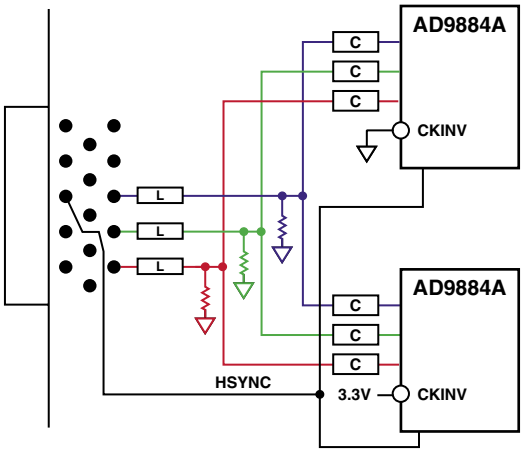

Figure 1. Analog Input Routing

The recommended method for clocking is to use the PLLs in both chips. This method requires special attention to HSYNC input layout, as illustrated in Figure 1. If very careful attention is paid to keeping the branch lengths identical (avoiding zig-zagging), then the skew between the two chips' sampling clock and digital outputs will be negligible.

Sampling Clock Inversion

All three clocking methods described earlier require chip 2 to sample RGB data 180° out of phase with chip 1. *The recommended method for doing this is to use the CKINV input (Pin 27)*. This method requires that chip 2 have its CKINV input pulled high so that it clocks RGB data using the opposite edge of chip 1. As illustrated in Figure 1, this allows both chips to run their clocks at half of the effective data rate. Chip 1 captures odd data on its sampling edge, while chip 2 captures even data on its sampling edge (180° out of phase from chip 1). An additional benefit of this method is the ability to align the data between the two chips through internal pipeline delay adjustment.

An alternative method to implement the chip 2 clock inversion is to use the phase adjustment control. The initial phase setting of chip 2 would be offset 180° (16 steps) from chip 1's initial phase setting. Using this method, the data output from chip 2 will be shifted by 1/2 pixel clock. This results in less setup and hold time margin when using a single data capture clock.

Another option is to invert the clocks externally, but this requires the use of an external PLL.

The timing issues associated with data capture are negligible if the capture device has separate capture clocks for each data port (odd and even data).

Clock Phase Adjustment
Although the internal clock delays should be the same, each chip's phase will need to be adjusted separately. Phase differences between the two chips can come from layout trace length variations in HSYNC or RGB inputs and from normal internal chip variations.

Since each chip is running at 1/2 speed, the phase adjustment step sizes are doubled with respect to the full speed clock. This results in 1/2 the number of useable phase adjustment steps (16 instead of 32) since the phase adjustment range will now cover two full speed pixels rather than one.

Interpart Difference Adjustments
Dual ADC applications are sensitive to the differences between the two ADCs, which can come from gain, offset, and linearity.

Gain and Offset
Dual AD9884A applications are highly sensitive to gain and offset errors between the two chips. Any difference between odd and even pixels is highly visible. *Therefore, accurate gain and offset adjustment is required for each chip.*

Linearity/Dithering
Dual ADC applications are also sensitive to differences in linearity between the two devices. If the application's requirements prove to be stringent, dithering may be required to minimize the vertical "banding" that may occur. The dithering method requires that on every other data frame, the even and odd device be swapped. For example, during frame one, device 1 processes odd pixels and device 2 processes even pixels. During frame two, device 1 processes even pixels while device 2 processes odd pixels. *Dithering allows the eye to essentially "average" the effect of the linearity differences, as well as differences in offset and gain*.

Device Addressing
Each AD9884A requires a different serial bus address.

Data Mode Selection
As with any AD9884A design, the output data mode must be considered. For a dual AD9884A design, there are three modes of operation. Each device can operate in single-channel mode or each device can run in dual-channel (demultiplexed outputs) mode. The third alternative is to have chip 1 operate in dual-channel mode when using just one chip, and then in single-channel mode when using both chips.

Single-Channel Mode
Single-channel mode requires 48 data output traces and is recommended if using a single graphics controller that has only 48 data input lines. In this mode, run in single-chip mode for frequencies up to 100 MHz, and run in ping-pong mode for frequencies over 100 MHz. Data switching limits the maximum operating speed to 200 MHz.

Dual-Channel Mode
This mode requires 96 data output traces and either two graphics controllers or a single controller with 96 data inputs. In this mode, run in single-chip mode for frequencies up to 140 MHz, and run in dual-chip mode for frequencies over 140 MHz. This mode offers the highest maximum operating frequency, 280 MHz.

Chip 1 Dual-Channel, Chip 2 Single-Channel
This mode requires 64 data output traces and a graphics controller that has a third 24-bit data port. In this mode, run in single-chip, dual-channel mode for frequencies up to 140 MHz, and run in single-channel, dual-chip mode for frequencies over 140 MHz. Data switching limits the maximum operating speed to 200 MHz. Because the maximum operating frequency doesn't improve over single-channel mode, this mode is not recommended since it requires more complexity.

For all three modes, the user must determine how to capture data. The large quantity of timing possibilities precludes listing all of them. For more information, refer to the timing diagrams in the AD9884A data sheet.

Reference Designs
Ping-pong reference designs using the AD9884A are currently available from several leading graphics controller companies. For access to these designs, general design help, or for more information, contact the factory via email at flatpanel_apps@analog.com.

AN05439–0–3/05(0)