Parallel Application of High Speed Link

In some cases it is desirable to design a system with a parallel configuration of High Speed Link chipsets. This is typically beneficial when the bandwidth of a single chipset is insufficient. For example, when the source clock is running at a frequency of 110 MHz, the clock and data to the FPD/Channel Link chipset may be split and transmitted over parallel paths to effectively double the bandwidth across the High Speed Link (with a maximum single chipset frequency of 65 MHz) and maintain the overall clock rate of 110 MHz. Each FPD/Channel Link transmitter/receiver pair will be operating at 55 MHz — half of the system clock frequency.

Take the Flat Panel Display Link application for dual pixel SXGA as an example. The clock and data from the graphic controller are "split" into odd and even pixels (data). The splitter function generates two clocks — CK1 and CK2 — at half of the graphic controller clock frequency (55 MHz). The splitter also separates the odd and even RGB data. This data splitting function is typically internal to the graphic controller, but may also be implemented as an ASIC external to the GUI. The odd data and CK1 are supplied to the data and clock inputs of one FPD Link transmitter; the even data and CK2 are supplied to the data and clock inputs of the second FPD Link transmitter. Data and clock of both FPD Link transmitters are sent across a cable, in parallel, to two FPD Link receivers. Timing and control signals are sent through only one of the transmitter/receiver pairs. At the output of the FPD Link receivers odd and even pixel data is sent to two parallel

flip-flops. An inverted clock signal (either CK1 or CK2) serves to latch all data into a second flip-flop stage. The data is then available to the timing controller, preserving the overall clock frequency of 110 MHz. (See *Figure 1*.)

The primary issue with a parallel configuration is the skew between the data paths and it's impact on the reconstruction of the parallel word. Data from each path must be available at the register inputs in time to be latched on the correct clock cycle. If the skew between parallel paths is too large, data will be misaligned and corrupted at the timing controller. The total skew between the two paths must be less than the time between adjacent clock edges (43% of clock period) to guarantee that data from both paths is at the register inputs *before* the inverted clock latches the data.

The skew between the two paths consists of chip-to-chip skew of the FPD Link transmitter- and receivers, skew of the cable, and system skew to the register inputs. Additional FIFO's before the register may be used to compensate for the skew. All parallel data will be available when the register clock arrives.

It is important that the clock signal to each of the transmitters be routed to reduce skew. Each clock signal should travel the same distance across the PCB from the output of the GUI (or splitter device) before reaching the clock input pins of the odd and even transmitters.
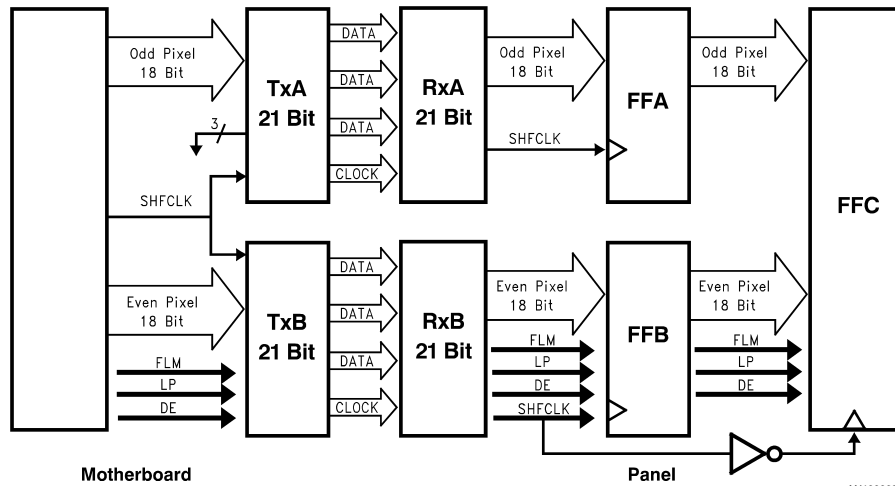

FIGURE 1. Dual Pixel Configuration with Flip-Flops and Register at Receiver Outputs

Definition of Skew Components:

1. Odd/even clock skew at input of FPD Tx. Dependent on dual/single clock source and electrical delay differences in signal layout to each transmitter.

2. Chip-to-chip clock skew at FPD Tx outputs. Defined as $T_{CCDmax} - T_{CCDmin}$ at single temperature and V_{CC} (per FPD Link datasheet).

3. Cable skew within a single cable (single cable interface is used — recommended) and defined as ps/unit length — or between two cables (if dual cable interface is used).

4. Chip-to-chip clock skew at the FPD Rx outputs. Defined as $R_{CCDmax} - R_{CCDmin}$ at single temperature and V_{CC} (per FPD Link datasheet).

Total budget for skew is the time between clock edges (falling and rising) in one clock cycle. Data is latched at the flip-flops on a falling edge; output of the flip-flop is then latched on the inverted rising edge.

For example, clock skew budget for SXGA:

   GUI clock = 110 MHz

   Shift clock frequency = 55 MHz

   T (shift clock period) = 18.2ns

   T * 0.43 = 7.8 ns (period multiplied by factor of 0.43 because RxCLKOUT duty cycle is 4:3 ratio)

Therefore, 1 + 2 + 3 + 4 should be less than 7.8 ns

Assume:

   #1 (clock layout skew) = 50 ps

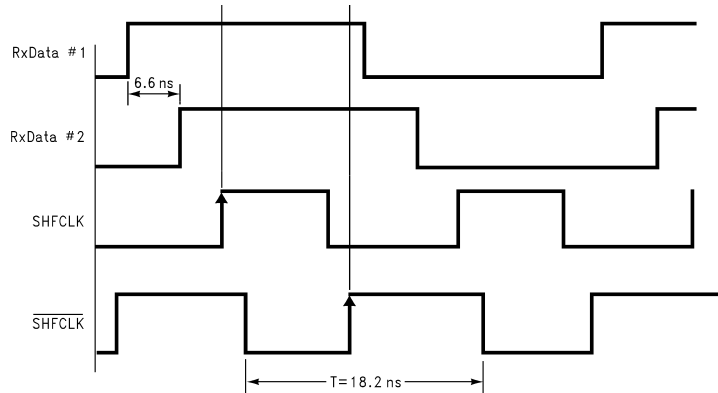   #2 (T_{CCD} window) = 2.5 ns (Note 1)

   #3 (cable skew) = 50 ps

   #4 (R_{CCD} window) = 4 ns (Note 1)

Note 1: Based on T_{CCD} and R_{CCD} specification for 3.3V FPD Link

Note 2: The flip flops should be integrated in the timing controller ASIC to minimize any additional delays and setup times.

Total skew = 6.6 ns

This value is less than 7.8 ns, therefore, the parallel word will be reconstructed correctly at the register.

AN100066-2

FIGURE 2. Data Correctly Reconstructed at 55 MHz

A maximum clock rate of 130MHz can be supported with a parallel configuration of the 65MHz FPD Link chipset. Due to the short clock period, FIFO's (or another method of aligning the parallel data paths) will be needed to guarantee correct reconstruction of the parallel word.

For example:

GUI clock = 130 MHz

Shift clock frequency = 65 MHz

T (shift clock period) = 15.4 ns

T * 0.43 = 6.6 ns

Using values for skew as in previous example, total skew is 6.6 ns. This is equal to the budget of 6.6 ns, therefore it may be necessary to align the data of the parallel paths (i.e., using FIFO's) to provide additional margin.
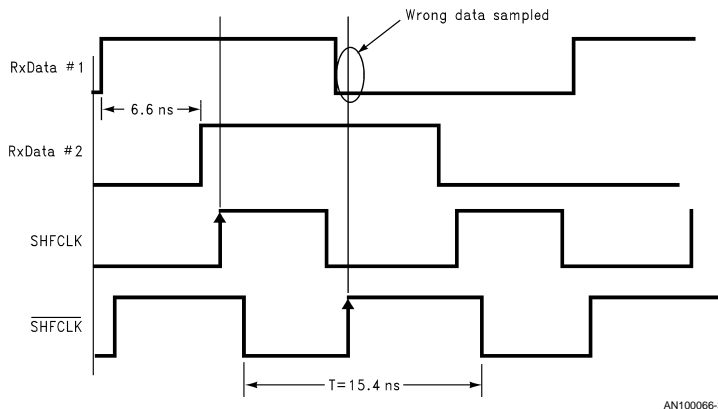

FIGURE 3. Data Incorrectly Reconstructed at 130 MHz

One way to align the data of parallel paths may be through the use of FIFO's. A FIFO may be added at the output of each receiver, before the inputs to the register. Logic would be included to monitor the "output register not empty" ($\overline{ORE}$) flags of each FIFO. $\overline{ORE}$ assertion indicates that the first parallel data word from the receiver outputs is available. When $\overline{ORE}$ of both FIFO's is asserted, the FIFO outputs are enabled and data from both paths are transmitted to the register inputs. This method would allow for up to one clock cycle of skew between the parallel paths.

There are several reasons for using a parallel configuration of high speed links. The most common is simply to gain bandwidth — two links deliver twice the bandwidth. Using two links also allows the interface to operate at half the data rate while maintaining the bandwidth of a single link running at full speed. Lowering the data rate allows the LVDS signals to be driven a longer distance — maximum cable length increases as data rate decreases. In addition, emissions are reduced at lower speeds and electromagnetic compliance is easier to achieve.

National's LDI (LVDS Display Interface) chipset provides an optimized solution for the dual pixel interface and high bandwidth requirement of emerging technologies. A single chipset includes a dual pixel interface, eliminating the need to address device-to-device clock skew and data alignment issues. The LDI chipset operates across a large frequency range (32.5MHz to 112MHz), with the flexibility to transmit data in either dual or single pixel mode. LDI also includes features to address the requirements of high speed data transmission over a long cable interface.

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.

2. A critical component in any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.