Shifting flat panel display data for improved images

FPGAs can add flexibility to the design of high speed display interfaces, writes **Gordon Hands**

With flat panel displays increasingly important in a range of products, the need to increase colour depth and resolution of the LCD panels creates a formidable challenge for display designers: how to move increasingly large amounts of data from video sources to the display. And, once inside the display, how should data be moved between boards in the system?

To address the need to transmit data to displays a range of standards has evolved. The digital visual interface (DVI) standard combines analogue VGA and digital display data in the same connector. Digital data is transmitted across up to two links, each comprising three channels of encoded data running at a raw data rate up to 1.6Gbit/s and a common clock running up to 165MHz. The more recent high definition media interface (HDMI) standard uses the same underlying transport mechanism, but in a digital only standard. Looking ahead, the universal display interface (UDI) and Display Port are vying to be the next generation standard, providing over 10Gbit/s of bandwidth in a single cable.

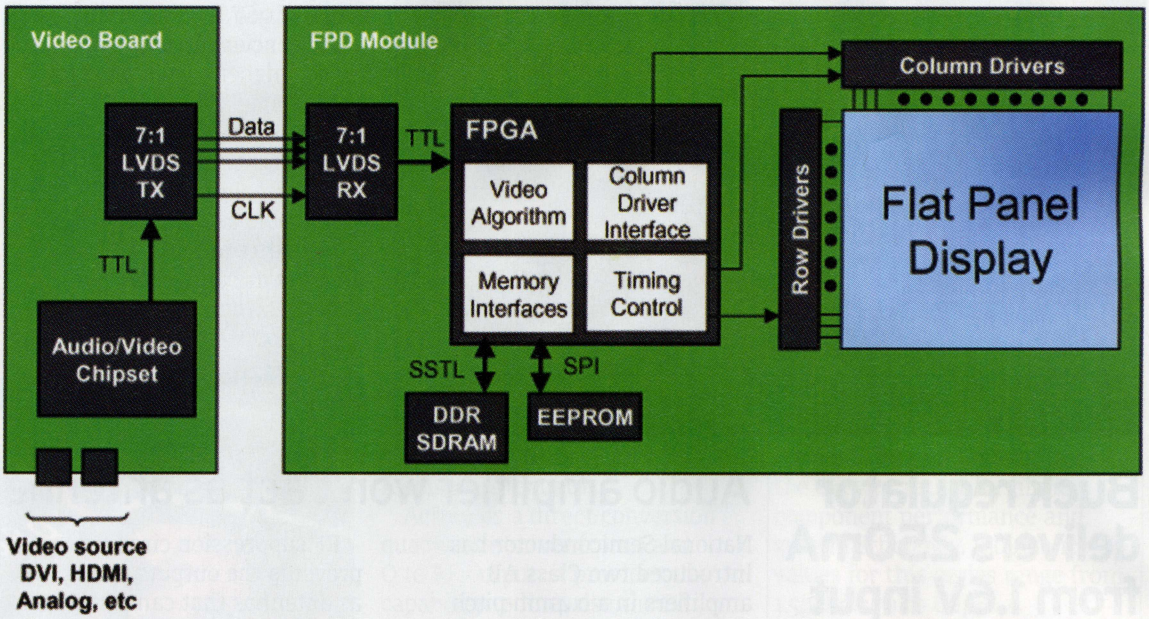

Figure 2: FPGAs can simplify design options in display applications.

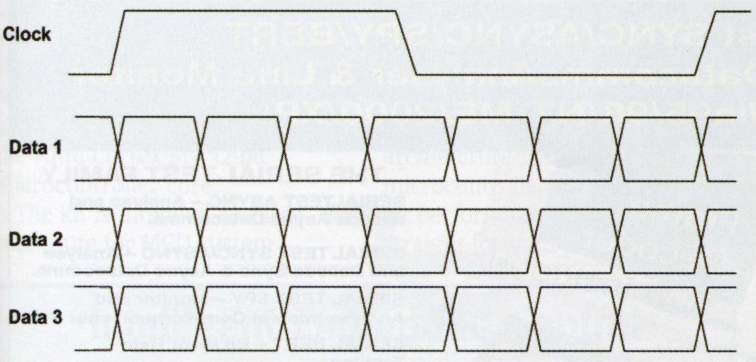

Figure 1: Basic timing of 7:1 LVDS Interface.

LCD displays in laptop computers were among the first applications to pose the challenge of communicating large amounts of display data from one board to another. The need to fit the cabling through the laptop hinge limited the number of wires that could be used. To solve this problem at the transmitter, data is serialised and presented as a serial stream, along with a copy of the low speed clock. At the receiver, this data is deserialised and presented to the rest of the system in parallel form.

Chips implementing this interface, often branded under the FlatLink and Channel Link names, quickly became the *de facto* standard for this application. Today this type of interface is commonly used in many board-to-board display applications. With these interfaces, seven data bits are serialised for each cycle of the low speed clock. Figure 1 shows the basic timing of these interfaces. Although conceptually simple, the need to implement high speed clocks for data serialisation and de-serialisation with tight timing budgets has caused some designers to implement these interfaces with application specific standard products.

Standard display drivers and interface chip-sets exist, but designers must format data from audio/video modules for the display, adjusting as necessary for the user's preferences. Given the range of displays available, a standard chip for this purpose often is not an option. Short product life cycles and a product's inherent variability mean that the number of shipments for each display type can be relatively low.

This has created a conundrum for

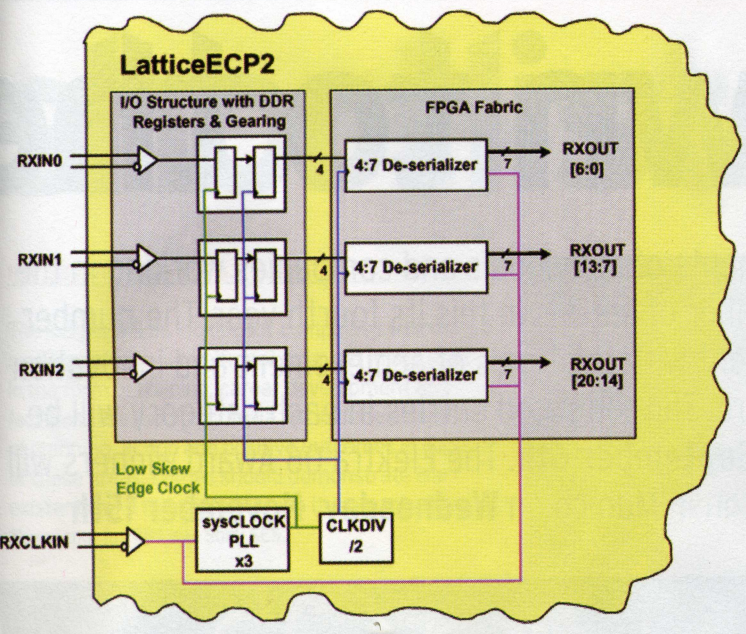

Figure 3: High speed board-to-board interface in LatticeECP2.

system designers who find the Asic approach no longer viable. One option is an FPGA-based design.

The requirements for an FPGA-based approach to the Channel Link and FlatLink style interfaces consist of four key components: a high speed LVDS buffer, a PLL for generating the de-serialization clock, input data capture and gearing and, finally, data formating. The input data and clock are presented to the device in LVDS format, with the data at relatively high speed. The FPGA must provide LVDS input buffers capable of providing the desired speed. The LatticeECP2 device provides input buffers suitable for operation up to 840Mbit/s.

The low speed clock provided with the data must be multiplied 3.5 times in order to capture the data, assuming the input capture circuitry operates in the double data rate (DDR) mode, in which data is captured on both the rising and falling edges of the clock. If the input capture circuitry operates only on one edge of the clock, a multiplication factor of seven must be used. An alternative method of operation would be to generate seven phase-shifted versions of the low speed clock and use these to capture the input data with seven different registers. However, the challenges of clock generation and distribution argue against this for an FPGA approach. With the example implementation, a PLL is used to multiply the clock by 3.5.

The registers that follow the LVDS input buffer must capture the data. Tight control of the clock and data relationship is important given the high speed of the incoming data stream. It also is necessary to gear, or reduce, the speed of the data before it is passed to the FPGA fabric. Most low cost FPGAs limit the operation of individual circuit elements to around 350MHz, with the practical limit for any reasonable amount of logic in the 225MHz and below range. The greater the gearing that can be done in the I/O structure, the lower the likelihood the FPGA fabric will be the limit on overall performance. In the example, the DDR registers within the LatticeECP2 I/O structure are used to capture the data, providing an inherent gearing factor of two. Additional gearing logic present in the I/O circuitry also is used to provide an extra gearing factor of two. The overall combined gearing factor of four allows the input to operate up to 900Mbit/s without exceeding the practical FPGA fabric speed of 225MHz.

The final step is to take the data from the I/O cells and format it into the original 7-bit width clocked by the low-speed clock. This logic can be constructed within the FPGA fabric. In the example design, the FPGA logic takes 4-bit wide data running at 1.75 times the low speed clock and converts it to 7-bit data running at the low-speed clock rate.

FPGAs with embedded serdes are becoming more common. This technology creates the potential for bringing interface standards such as DVI, HDMI, Display Port and UDI directly into the FPGA. Such integration promises to further simplify systems and reduce costs.

GORDON HANDS IS STRATEGIC MARKETING MANAGER AT LATTICE SEMICONDUCTOR