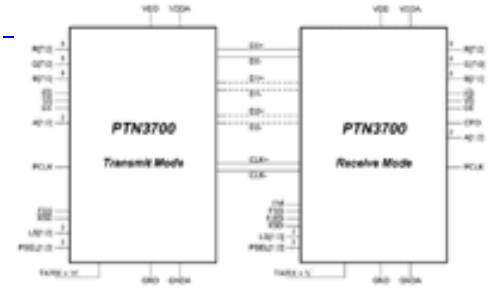
Designing Graphics-rich Mobile Devices: Part 2 of 2

By Nic Roozeboom

High-speed Serial Interfaces for Mobile Devices

High-speed serial interfaces replace parallel topologies in a wide array of applications today. Many of today's common interconnect standards such as USB and PCI



[1]

Express are based on serial transmission to achieve speed, physical compactness and link robustness, as do a vast array of implementations less visible to the consumer, such as notebook computer display interconnect, high-speed backplane interconnects and emerging memory bus architectures.

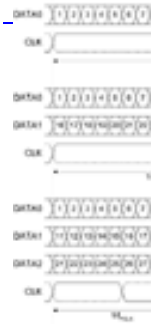
Though different in scope and optimized for best performance in specific environments, high-speed serial interconnects all make use of a few essential elements. Perhaps foremost, several important benefits are all at once achieved by using differential signaling, which provides a substantial reduction in noise emission and allows the signal swing to be substantially reduced, in turn reducing the amount of required signal power.

The ratio at which data is serialized is chosen such that per parallel word transmitted, all data bits (the "payload") plus any overhead (due to line coding &#151 more on this later &#151 and the addition of other useful bits such as parity or error correcting code) can be transferred within the parallel clock period. For example, to serially transmit one 24-bit video pixel (8-bits each for R, G and B color words) along with its synchronization bits (horizontal sync, vertical sync and data enable) without any other overhead, one would need the outgoing serial bit rate to be at least 27 times the incoming pixel clock rate. Let us assume that two additional general purpose bits will be added, as well as one parity bit (again, more later) to complete a total serial bit count of 30 bits per period of the parallel pixel clock.

The frequency at which video data will be transmitted is bound to two basic quantities: the physical implementation of the video display grid (number of

horizontal and vertical pixels), and the display refresh rate (the rate at which the entire display's grid of pixels and lines is refreshed once). Since the display and display driver need additional time between lines and between the end and start of a frame (also known as horizontal and vertical back and front porches, respectively) again some overhead is allowed for here. Ultimately, the pixel rate can be calculated by multiplying the display refresh rate by the number of horizontal and vertical pixels including overhead. The required serial bit rate can then be calculated by multiplying the pixel clock frequency by the number of serial bits per frame (see Table 1).

Should display sizes cause the serial bit rate to exceed what is desirable from an IC implementation or application standpoint (e.g. one might wish to limit the serial



interface's signaling rate to typical LVDS maximum rates of 650 Mb/s so that it can be realized in cheap and generic CMOS processes), then it is possible simply to distribute the payload over multiple serial lanes, reducing the absolute signaling rate per lane by that same factor. This makes it possible to scale from lower end display sizes such as QVGA (bitrates of about 120 Mb/s) all the way up to high-end display sizes such as XGA (bitrates of around 1250 Mb/s) simply by utilizing the necessary number of lanes as needed (see Table 1 for example bit rate calculations).

Depending on the specific end application for the video serial interface, additional overhead may or may not be needed, at the expense of complexity and efficiency. For traversing relatively short distances (several or tens of centimeters), the simplest solution is source-synchronous transmission, where the clock reference for the serial data is transmitted as a separate signal along with the data. When longer distances have to be reached (several meters), the difficulty of controlling skew, jitter and other timing issues will increase to the point where it is necessary to use line coding, a process in which the clock reference is embedded into the data stream. This in turn necessitates clock recovery from the data stream at the receiver end, also increasing complexity and inefficiency depending on the level of sophistication of the line coding scheme. Further complexity may be needed when it is necessary to encrypt data for intellectual property protection reasons. But for the purposes of this article, links usually remain short and internal to the mobile device. Moreover, any overhead added to the original data effectively increases the amount of power required to transport each bit, a consideration of paramount importance in battery powered handheld devices &#151 second to, most likely, space constraints.

For these reasons, the arguments are strong to opt for simple source-synchronous transmission as suitable and appropriate for the intended application space.

Landscape of Serial Interface IC Solutions

The MIPI Alliance has recognized the need to unify today's and tomorrow's technical requirements for mobile phones into comprehensive interface standards, so that vendors may benefit from optimal solutions that operate well together, are widely available and can be integrated into core chipsets and peripherals. As of today, fully ratified standards exist for the Display Serial Interface (DSI), the Display Command Set (DCS) and Camera Serial Interface (CSI-2), for example. For high-volume, cost-sensitive and performance-intensive applications such as cell phones, vendors of IC solutions (baseband processors, application/video coprocessors, display drivers, interface products) will emerge with increasingly versatile portfolios of MIPI-based solutions.

However, it can be expected that due to the drive for technical innovation and differentiation, vendors will seek solutions that are available now that solve the serial display

[3]

interface solution in an efficient and cost-effective way. In this environment, "pre-MIPI" solutions from many different IC vendors are seen to be used that, although not providing all desired benefits such as full integration, interchangeability, interoperability and multiple sourcing, they do allow set makers to advance their feature roadmaps faster and enable new functionality. Solutions such as the PTN3700 are designed as a stepping stone solution, allowing customers to use currently available products to migrate their product architectures to future MIPI-based solutions. Since redesigns of core chipsets to implement a new interface are costly and take time, add-on solutions mitigate both cost and risk by extending the useful economic life of the existing chipset, and allow quicker market introduction of advanced features and increased performance.

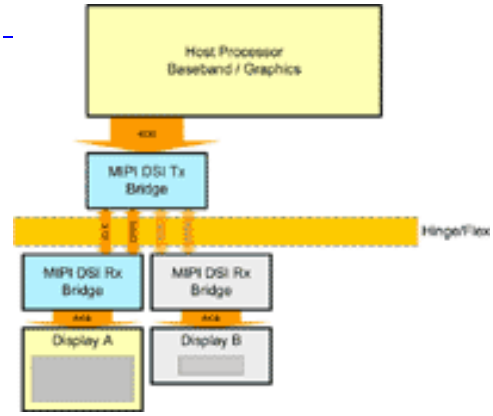
Even as MIPI standards achieve widespread adoption, a great many solutions will be able to benefit from full integration into core chipsets. In non-mainstream applications that incorporate high-resolution displays, it may not be economically attractive to redesign a processor or system-on-chip to reap the full benefits of a (MIPI) serial display interface. There may also be examples where a core IC is used across a variety of products within a platform (say, a printer platform), only a subset of which include a fully featured color display (e.g., a photo printer) and for that feature alone require a serial interface. In such scenarios, stand-alone MIPI

interfaces that can be added on to existing parallel architectures will be able to find useful applications.

Example Discrete Bridge IC Serial Solution

The NXP Semiconductors PTN3700 is an example of a configurable serializer and deserializer with features and characteristics optimized for use in space-constrained and power-conscious applications such as mobile phones and PDAs. The PTN3700 is configurable as either transmitter (serializer) or receiver (deserializer). The parallel I/O allows input and output of up to 24 bits of RGB video data, plus synchronization signals (horizontal, vertical sync, data enable and clock). The serial interface is organized as configurable between one and three differential data lanes, depending on what video bandwidth is needed to support the target display. For smaller displays, a single lane of data will suffice with pixel clocks between 4 to 15 MHz yielding up to 450 Mb/s data rate. Double the rate can be achieved using two-lane mode. For very high-end displays, three lanes could be used with pixel clock frequencies between 20 to 65 MHz for aggregate data rates of up to 1.95 Gb/s.

The data lanes are accompanied by a clock signal, for source-synchronous transmission of the data. Source synchronous transmission provides high transmission efficiency



[4]

because the need to embed the clock signal with the data can be avoided, as can the complexity of clock and data recovery on the receiver side. SubLVDS, a popular low-voltage flavor of LVDS &#151 adjusted for 1.8 V supply and I/O environments by lowering differential voltage swing and common mode range &#151 is used as the physical I/O for the serial interface.

Two transmission methods are possible with the PTN3700, different in their fundamental approach, but which are outwardly transparent to the user. Both are source-synchronous, meaning that the clock reference is explicitly sent with the data, and is necessary for determining valid data at the receiver. However one mode uses the original pixel clock edge to determine the absolute timing position of the first data bit sent in the serial stream. This means that there is no bit clock per se, and the receiver in order to correctly decode the incoming serial bit stream will have to regenerate a local bit clock reference by using a PLL, multiplying the received pixel clock frequency by the ratio of parallel-to-serial conversion used. This has the benefit that the position of bit #1 in the serial stream is known and simply

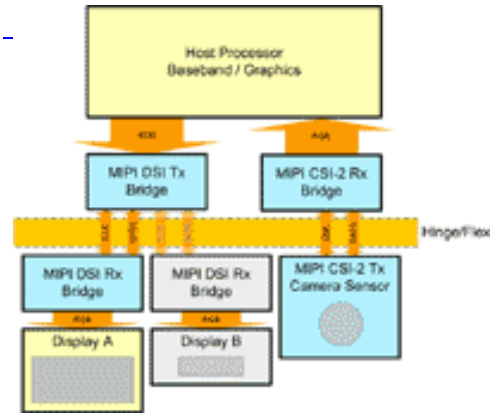
recognized by the receiver, which will align the remainder of the serial bits to it. The drawback is that skew between clock and data has to be very tightly aligned, and also this necessitates the use of a PLL in the receiver, which consumes a small but, in mobile applications, appreciable amount of power.

Figure 2 shows waveforms of a serial video data transmission method using the pixel clock edge as the timing reference for 30 bits of data, in configurations using 1, 2 and 3 data lanes respectively.

The second transmission mode option, rather than using one clock edge to delineate a series of bits, uses a double-data-rate bit clock (i.e., both clock edges are used) with each serial data bit, thus providing a clear timing reference for each bit sent across the serial link. This eliminates the need for a PLL in the receiver. However, this presents an apparent challenge: how to recognize the position of the first bit in the serial stream, such that the data is correctly decoded? In order to solve this, the PTN3700 makes use of embedded synchronization words, which are added to the data in the inactive portions of the video signal (i.e., when HS, VS and DE are active, during which no RGB video bits are displayed on the screen). These synchronization words are merged into the serial data by the transmitter, and decoded by the receiver which in this way is able to instantly lock to the incoming data stream (as it is fully source-synchronous). The receiver will instantly achieve synchronization by recognizing known unique synchronization words embedded in the data which tell the position of the first bit in a pixel and hence determines the position of the remaining bits. Besides the power savings of eliminating a receiver PLL, design constraints of the serial interconnect may be eased since a clock edge accompanies every serial bit sent.

Figure 3 shows the serial transmission method based on a double-data-rate clock to transmit data, in configurations of 1, 2 and 3 data lanes.

The PTN3700 is implemented as a standalone bridge IC, but its essential functions may be fairly easily integrated into host ICs such as application co-processors or display driver ICs, as it is specified around speeds and interfaces for which the use of non-exotic standard CMOS processes suffices. As such a standalone bridge IC can provide a



[5]

welcome complement to designs that use legacy parallel video interface,

e.g. providing an enhancement to an existing baseband or application co-processor by adding the capability to interface to a new display module equipped with a serial interface. Other scenarios in which a standalone bridge IC is useful are in the display module, where typically a higher voltage and larger geometry process might be used for the display driver (hence making it less suitable to host the high-speed, low-voltage differential serial interface). As size is paramount in a display module and board space is a rare luxury if at all available, the bridge chip is realized as bumped bare die, which is so tiny that it can be mounted directly and robustly onto the display module's flex foil.

Because the primary purpose of foil interconnects is their flexibility, more often than not only a single routing layer is available, otherwise the material will become too rigid. This introduces some challenges for the layout of the serial interconnect between the main PCB on the host side and the display module. Since signal crossings are not allowed, the order of differential clock and data lines must map exactly between transmitter and receiver. In order to address this fact as well as provide the application designer with additional freedom to place transmitter or receiver on either side of the PCB or flex foil, the PTN3700 is configurable (hardware pin-programmable) to a total of eight different mappings of "mirrored" input and output signals, including the serial interface signals as well as the parallel I/O. This allows the PTN3700 transmitter and receiver to be juxtaposed in the most convenient orientation in a given assembly design.

One "value-added" feature of the PTN3700 is Frame Mixing, an algorithm that converts 24-bit source video data to an 18-bit output stream while preserving the full array of 24-bit (16.7 Million color) resolution. The Frame Mixing algorithm can be enabled by a single pin and requires no programming as it is implemented in hardware in the PTN3700 and fully self-contained. When enabled, the algorithm encodes the color information of the two LSBs of the eight-bit-per-color (R, G, or B) data into the LSB of the new six-bit-per-color word by applying temporal and spatial modulation to the new six-bit LSB. As a result, for still and relatively slow-moving video, where color depth is the most critical, the full 16.7 million-color palette of the 24-bit source is fully rendered, producing visibly smoother color gradations and transitions. The cost of true 24-bit display modules is much greater than 18-bit implementations, since the complexity of both the display driver and the LCD are substantially increased: both have to support fourfold the number of levels per RGB color bit. Frame Mixing enables the use of cheaper and more widely available 18-bit display modules for 24-bit display applications, and as such is an example of how serial interface solutions can provide a cost benefit.

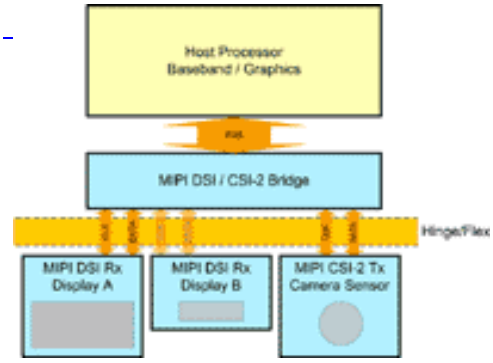
Future Trends

Thus far we have described an available serial display solution that accomplishes a few important basic goals: serialization of the interface combined with differential signaling, to achieve low wire count, low EMI and robust data transfer at high bandwidth. Over the next few years, solutions will become available on the market from multiple vendors that are designed around the MIPI standards for camera (CSI-2) and display (DSI) serial interfaces. This will create an environment where mobile handset makers can design complete solutions based on a modular approach using interoperable components and subsystems. To enable early

introduction of MIPI-enabled designs, bridge solutions may be considered during the time it takes for fully integrated core ICs, display modules and camera modules to become available. Bridge ICs can be useful for a variety of reasons depending on the specific design and product introduction environment. A few typical motivations are discussed here.

First of all, the use of new MIPI bridges allow accelerated realization of MIPI solutions in platforms based on legacy parallel interfaces. In this way, designers can achieve most if not all the technical benefits of high-speed serial solution: wire count reduction, low EMI, low-power transmission enabling high-resolution video to traverse a mechanically challenging interconnect for example.

Furthermore, attaching a bridge to a design with a legacy interface effectively enables that platform with a standard interface, which in turn can be mated with modules and



[6]

subsystems built to that standard (i.e., MIPI DSI or CSI-2) interface, realizing a wider and modular selection of components across multiple vendors and implementations.

As such, stand-alone bridges may be a valuable part of a flexible ecosystem of standard-based solutions, complementing and enabling platforms revolving around core processors and subsystems that may or may not feature integrated high-speed serial interfaces. This enhances product definition and market introduction effectiveness by adding expandability, flexibility and re-usability of modular solutions.

Especially when implementing high-speed interfaces in excess of 1 Gb/s signaling rates, the design of the basic IP for the physical layer interface, nor its implementation are quite so trivial. Often in cases where risky analog is being initially implemented on an expensive ASIC, a backup solution is desirable. A separate bridge can in such cases mitigate the risk of high-speed mixed-signal IP integration while providing insurance for timely market introduction of new essential functionality.

Conversely, add-on interfaces provide benefit by prolonging economic lifetime of core chip(set)s and thereby preserving significant investment in hardware and software infrastructure. Adoption rates of new interfaces often differ between core ASICs on one hand and peripherals such as displays and cameras on the other, and

a stand-alone bridge resolves the discrepancy by being able to re-use an existing host processor with parallel interface for a longer period.

MIPI Bridge Outlines

A few examples of high-speed serial interface bridges with possible applications in mobile devices are discussed below.

MIPI DSI Display Serial Interface Bridge

Figure 4 depicts a topology where a host processor with legacy (parallel) is enabled to interface with high-speed serial interface-equipped display module, typically consisting of one or two displays (one main, one auxiliary display. The display module itself may consist of a fully integrated display driver with serial interface or in its turn be realized using a separate high-speed serial receiver. Both transmitter and receiver serial interface bridges form a coordinated solution such that they are fully transparent to the host.

MIPI CSI-2 Camera Serial Interface Bridge

Figure 5 expands on the display-only topology by adding a receiver bridge capable of interfacing with a MIPI-CSI-2 equipped camera sensor, translating the Camera Serial Interface data stream to a parallel RGB format for the host processor. For a more integrated solution where the attachment rate of displays to cameras is 1:1, the solution shown in Figure 6 allows for a smaller footprint solution.

MIPI DSI/CSI-2 Integrated Host Bridge

The migration to higher resolution displays and camera sensors in mobile devices is advancing rapidly with the consumer demand for more content-rich media while on the go. While this trend necessitates the need for higher and higher data bandwidth between host processor and display or camera, despite the need for continuous miniaturization and lower power consumption, the technical challenges can be surmounted by the use of efficient high-speed serial interfaces. While integration of serial interfaces such as MIPI DSI or CSI onto host processors and peripheral devices is ideal for power and space considerations, early adoption is greatly helped by stand-alone devices able to bridge legacy devices to the new standard. Smartly designed stand-alone bridges, due to their small footprint, low power and easy applicability allow designers to realize most of the technical advantages of high-speed serial transmission, while offering benefits such as extending the useful economic life of existing chipsets and peripherals, allowing market introduction before integrated solutions are available, and providing greater flexibility to add high-speed serial capability across a wide range of platforms, modules and peripherals.

About the Author

Nic Roozeboom is technical marketing manager, High-Speed Interface, Product Line Interface Products, BL Standard ICs, for NXP Semiconductors.

Source URL (retrieved on *03/12/2014 - 4:15am*):

http://www.wirelessdesignmag.com/product-releases/2007/01/designing-graphics-rich-mobile-devices-part-2-2

Links:
[1] http://www.wirelessdesignmag.com/sites/wirelessdesignmag.com/files/legacyimages/0612/feat1b1_lrg.gif
[2] http://www.wirelessdesignmag.com/sites/wirelessdesignmag.com/files/legacyimages/0612/feat1b2_lrg.gif
[3] http://www.wirelessdesignmag.com/sites/wirelessdesignmag.com/files/legacyimages/0612/feat1b3_lrg.gif
[4] http://www.wirelessdesignmag.com/sites/wirelessdesignmag.com/files/legacyimages/0612/feat1b4_lrg.gif
[5] http://www.wirelessdesignmag.com/sites/wirelessdesignmag.com/files/legacyimages/0612/feat1b5_lrg.gif
[6] http://www.wirelessdesignmag.com/sites/wirelessdesignmag.com/files/legacyimages/0612/feat1b6_lrg.gif