Leveraging MIPI D-PHY-based Peripherals in Embedded Designs

A Lattice Semiconductor White Paper

May 2014

Lattice Semiconductor
5555 Northeast Moore Ct.
Hillsboro, Oregon 97124 USA
Telephone: (503) 268-8000
www.latticesemi.com

Ultra-Low Density FPGA-based Bridging Opens Doors to New Applications for MIPI Components

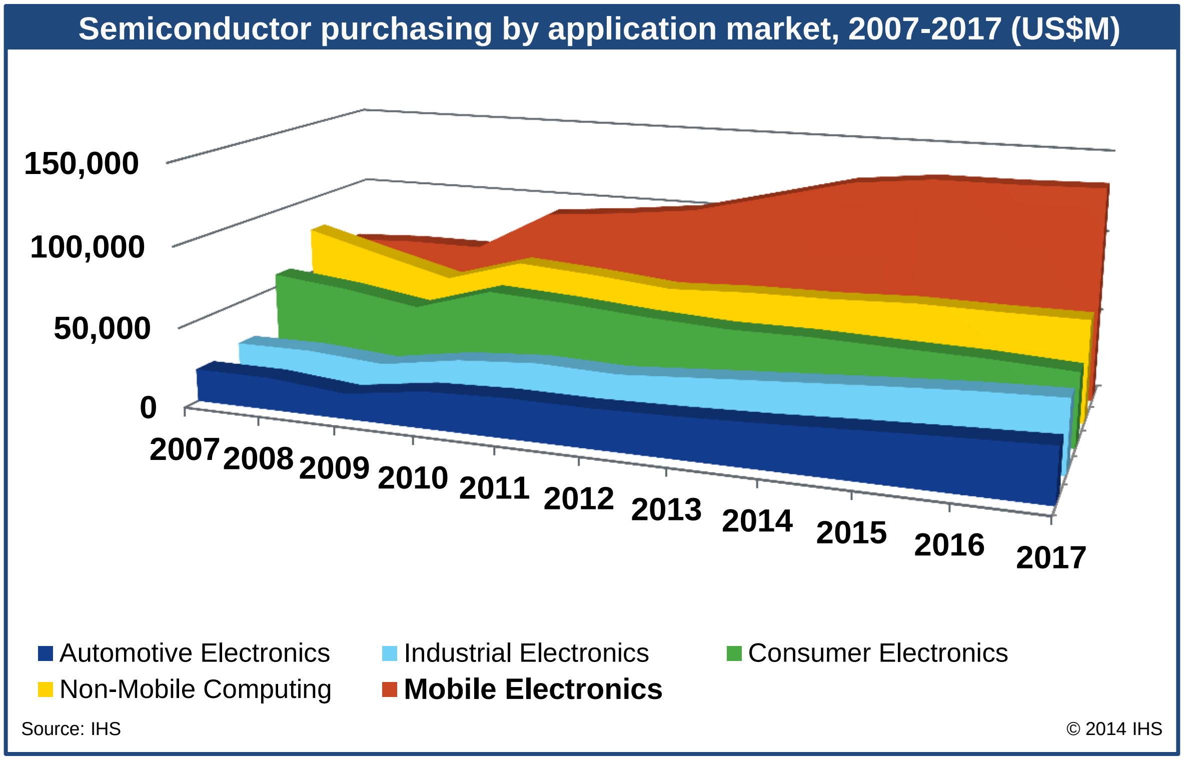
Embedded systems designers face an ongoing dilemma. On the one hand they need to drive down systems costs. On the other they cannot exploit manufacturing economies of scale because their systems are targeted at relatively narrow, low volume applications. While high volume consumer markets offer components capable of performing similar tasks at much lower cost, embedded designers are restricted from taking advantage of those components by their systems' reliance on highly specialized, legacy interfaces optimized for the embedded environment. This issue plays out most vividly in displays, cameras and application processors where low-cost, MIPI-compliant components for mobile platforms with interfaces based on the D-PHY physical bus cannot communicate with embedded system processors that typically feature LVDS, RGB or SPI interfaces for their display and digital parallel, subLVDS or HiSPi interfaces for an image sensor.

This paper looks at potential solutions to this problem and explores how embedded systems designers can leverage the cost/performance advantages system developers in high volume consumer markets have already learned. In particular, it examines the use of a new class of Ultra Low Density (ULD) FPGAs and accompanying reference designs developed by Lattice Semiconductor to create low-cost, highly configurable bridging solutions for the use of D-PHY-based, MIPI-compliant peripherals in embedded designs.

History Lessons

A few decades ago when embedded system designers wanted to reduce costs, they turned to the high-volume architecture of their day. By utilizing standardized hardware and software originally developed for the popular PC architecture, embedded designers discovered they could drive down development costs while using highly reliable components proven in tens of millions of consumer applications. Components and subsystems built for the PCI bus offered a highly attractive, lower-cost alternative to complex, expensive embedded bus architectures such as VME, STD and Multibus. Adapting the PCI and later the PCIe bus allowed embedded developers to take advantage of established, easy-to-use design tools and open source operating systems on a highly familiar architecture. Ultimately, this strategy paid multiple dividends, enabling designers to pass lower costs on to their end customers, while shortening their development cycle and more easily meeting time-to-market windows.

Those days are long gone. The PC architecture is no longer the high volume leader (Figure 1), that mantle has been handed to mobile computing. Today, if embedded developers want to drive down their system costs, they have to look to the components and interfaces used in today's high volume segment -- the rapidly growing smart phone and tablet market.



Source: IHS

© 2014 IHS

Most smart phones and tablets on the market today use buses and interfaces defined by the Mobile Industry Processor Interface (MIPI) Alliance. Founded in 2003, the MIPI Alliance was established to promote component interoperability by defining specifications for standard hardware and software interfaces between an applications processor and the wide range of peripherals used in mobile systems including image sensors, memories, displays, RF components and other devices in mobile platforms. Figure 2 illustrates typical MIPI-compliant interfaces in a mobile platform.

Perhaps the most commonly-used components that embedded designers can leverage from the MIPI-compliant mobile market are displays, cameras and application processors. As Figure 2 illustrates, mobile designs today typically use the Display Serial Interface (DSI) for the LCD screen and the Camera Serial Interface (CSI-2) for the camera image sensor. The primary challenge embedded designers face to utilizing these lower cost mobile components is building bridges between the mature interfaces commonly used in embedded applications and the MIPI Alliance-defined interfaces used in mobile. In the embedded market, for instance, displays often feature a LVDS, RGB or SPI interface while image sensors typically use a digital parallel, subLVDS or HiSPi interface. Most embedded processors do not offer a DSI display interface. Accordingly, designers will need a bridge device to connect their embedded processor to a DSI display. Similarly, embedded design teams who want to integrate a low cost image sensor originally targeted at the mobile market will need a bridge to the CSI-2 interface.

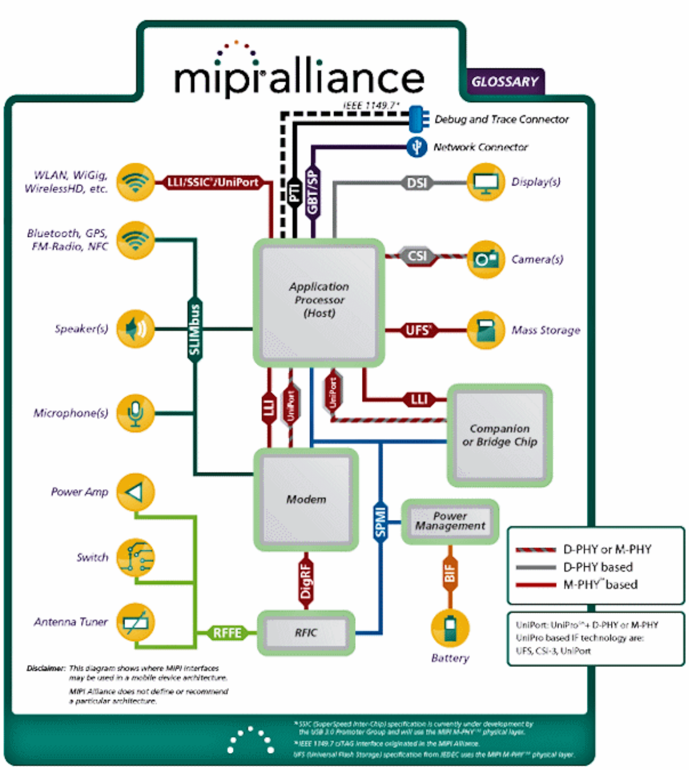

mipi alliance
GLOSSARY
IEEE 1149.7*
Debug and Trace Connector
Network Connector
WLAN, WiGig, WirelessHD, etc.
LLI/SSIC*/UniPort
PTI
GBT/SP
DSI
Display(s)
Bluetooth, GPS, FM-Radio, NFC
CSI
Camera(s)
Speaker(s)
SLIMbus
Application Processor (Host)
UFS*
Mass Storage
Microphone(s)
LLI
UniPort
UniPort
LLI
Companion or Bridge Chip
Power Amp
Modem
SPMI
Power Management
Switch
DigRF
BIF
Antenna Tuner
RFFE
RFIC
Battery
D-PHY or M-PHY
D-PHY based
M-PHY based
UniPort: UniPro + D-PHY or M-PHY
UniPro based IF technology are: UFS, CSI-3, UniPort
Disclaimer: This diagram shows where MIPI interfaces may be used in a mobile device architecture.
MIPI Alliance does not define or recommend a particular architecture.
*SSIC (SuperSpeed Inter-Chip) specification is currently under development by the USB 3.0 Promoter Group and will use the MIPI M-PHY physical layer.
*IEEE 1149.7 cJTAG interface originated in the MIPI Alliance.
UFS (Universal Flash Storage) specification from JEDEC uses the MIPI M-PHY physical layer.

Figure 2: Interfaces for mobile platforms defined by the MIPI Alliance.

The D-PHY Interface

Both the MIPI CSI-2 and DSI interfaces are based on a unique physical bus called a D-PHY. Featuring a differential clock and one to four differential data lines, the D-PHY is a center-aligned, source synchronous interface which is clocked on both the rising and falling edge. One device serves as a transmitter and another as the receiver. The bus offers the unusual ability to change “on-the-fly” from differential to single-ended signaling. Typically differential or High Speed (HS) mode is used to drive high-performance video, while the single-ended low power (LP) mode is used to transmit control data.

The DSI HS interface operates electrically as a Scalable-Low-Voltage-Signally (SLVS) standard device with a 200-mV common-mode voltage. The clock is DDR source synchronous and the number of data lanes on the interface can vary from one to four. The higher the display resolution and refresh rate, the more data lanes and speed required. The DSI interface uses LP mode on data lane 0 to configure the register map setting of the screen. This mapping is referred to as the Display Command Set (DCS). Accordingly, designers building bridges to DSI must not only map the video or graphics data to HS mode, they also need a mechanism to configure the display screen in LP mode. Note that this marks a crucial distinction between the DSI and CSI-2 interfaces. Rather than use LP mode, the CSI-2 image sensor employs a separate I^{2}C bus to program the image sensor.

While the current generation of application processors offers very attractive advantages in terms of functionality, high integration and low power, many embedded designers cannot make the transition from generations of legacy processors because of their enormous investments in software and peripheral functions. . More often than not, the software development costs of switching to another processor are simply too high.

These designers can still, however, leverage some of the lower cost components used in mobile products. As an example, consider a microcontroller-based embedded design where the existing investment in software is too high to justify a transition to a new applications processor. Let's assume that the existing controller uses a CMOS RGB or LVDS flatlink bus to link to a LCD screen. The designers would like to move to a low cost DSI display, but cannot because it uses a D-PHY bus which is incompatible with the controller's existing interface.

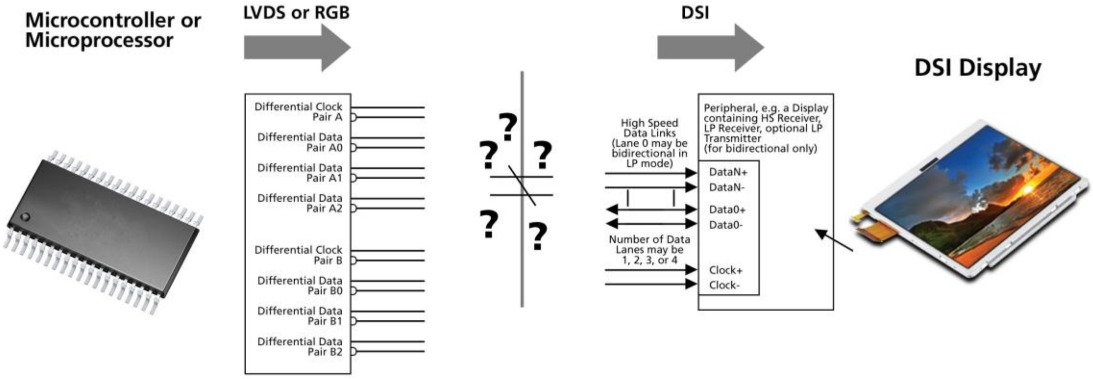

Microcontroller or Microprocessor
LVDS or RGB
DSI
DSI Display
Differential Clock Pair A
Differential Data Pair A0
Differential Data Pair A1
Differential Data Pair A2
Differential Clock Pair B
Differential Data Pair B0
Differential Data Pair B1
Differential Data Pair B2
? ? ? ? ?
High Speed Data Links (Lane 0 may be bidirectional in LP mode)
Number of Data Lanes may be 1, 2, 3, or 4
Peripheral, e.g. a Display containing HS Receiver, LP Receiver, optional LP Transmitter (for bidirectional only)
DataN+
DataN-
Data0+
Data0-
Clock+
Clock-

Figure 3: Many of the legacy microcontrollers used in the embedded market do not offer MIPI Alliance-compatible interfaces.

New Bridging Options

Until recently designers would have to justify the relatively high cost of a dedicated ASIC to take advantage of a low cost DSI display in their design. In most cases the cost of the ASIC would be far too high, and the development cycle too long, to support a design change of that order. Instead, designers would have to opt for a more expensive display.

Today designers have the option of using a highly configurable, ultra-low-density FPGA-based solution to construct a D-PHY bridge. To address this problem and simplify development of bridging solutions, Lattice Semiconductor has developed a family of reference designs that allow OEMs to take advantage of the low cost of MIPI-enabled camera, application processor and display components. Lattice currently offers four reference designs for MIPI camera and display applications. They include a MIPI DSI transmit bridge to drive a DSI-receiving devices such as a DSI display, a DSI receive bridge to enable an applications processor to interface to a screen not designed for mobile applications, a CSI-2 transmit bridge to enable an applications processor to interface to a non-CSI-2 image sensor, and a CSI-2 receive bridge to enable a CSI-2 image sensor to interface to an embedded ISP. Using Lattice's ULD FPGAs, developers can quickly build programmable bridging solutions that can be modified to address different requirements. They simply specify the interface they need on a form on the Lattice website and they receive an HDL netlist targeting any one of Lattice's MachX02, MachXO3 or ECP3 FPGAs.

How would this work? Let's re-examine the design using an embedded microcontroller that's built to support a CMOS RGB or LVDS flatlink bus to a LCD screen. Assuming the microcontroller features a CMOS RGB888 (24-bit color bus) display interface, the first step would be to determine how to program the configuration registers of the DSI display. Typically, the microcontroller will use an I²C bus to perform this function. However, the MIPI DSI specification does not accept I²C for display configuration. Instead, DSI uses the Low Power mode of the serial data lane 0 for the Display Command Set (DCS). In this case the FPGA bridge must convert the I²C commands from the microcontroller to a series of DCS commands to configure the DSI screen. After the screen is configured, the FPGA must be set up to receive the data from the RGB888 interface. If the resolution of the bus and the display is identical, the FPGA would then convert the parallel bus to the serial DSI bus. If the resolutions differed, the FPGA would scale the image up or down. In either instance, it would be necessary to configure the number of output data lanes for the DSI interface. Once that is accomplished, the FPGA could output a DSI transmit interface to drive a DSI display.

What if the embedded systems designer wants to integrate a low-cost CSI-2 image sensor into a design, but the design's image signal processor (ISP) only features a CMOS interface? As mentioned earlier, one of the important distinctions between the DSI and CSI-2 interfaces is how they map image sensor register data. CSI-2 uses a separate I²C bus to perform this task. This difference simplifies the designer's task by transmitting the register configuration directly from the image sensor to the ISP via an I²C bus. In this scenario the CSI-2 image sensor data would act as an input bus to the bridge FPGA and the CMOS parallel bus on the ISP would be driven by the FPGA (see figure 4).

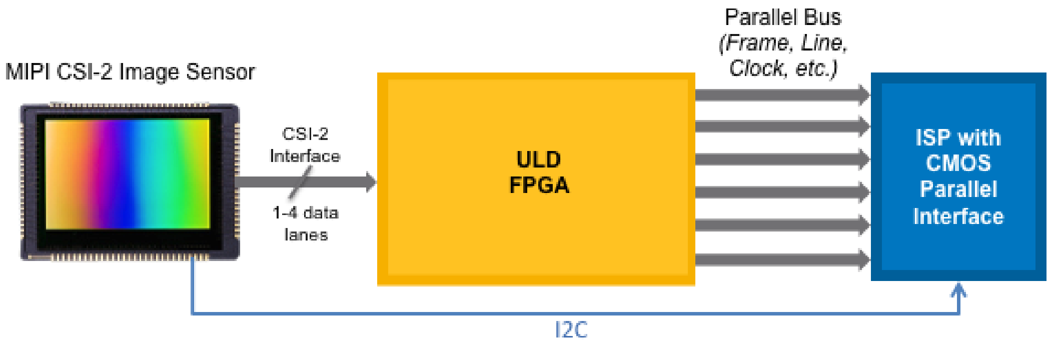

MIPI CSI-2 Image Sensor
CSI-2 Interface
1-4 data lanes
ULD FPGA
Parallel Bus (Frame, Line, Clock, etc.)
ISP with CMOS Parallel Interface
I2C

Figure 4: A CSI-2 image sensor bridge based on a ULD FPGA.

The same CSI-2 bridge solutions can be used to expand the functional capabilities of existing embedded systems. A growing number of applications, ranging from 3D stereoscopic video to black box automobile recorders require multiple image sensors. Unfortunately, many of the popular ISPs on the market today only feature a single image sensor port, even though they can deliver the performance needed to support two sensors.

Using the dual sensor bridge depicted below (see figure 5), designers can expand ISP port configuration and support many of these new applications. This example illustrates an image sensor bridge for a black box car driver recorder which uses two cameras, one looking out the front windshield and a second pointed at the driver. The bridge interfaces to each image sensor's parallel CMOS bus. The bridge can then output the two merged 720P images in either a top/bottom or left/right configuration. In this instance the bridge is implemented in a 1300-LUT MachX03 FPGA which synchronizes and manages the two image sensors. It then outputs the data in a CSI-2 format the ISP or application processor can accept.

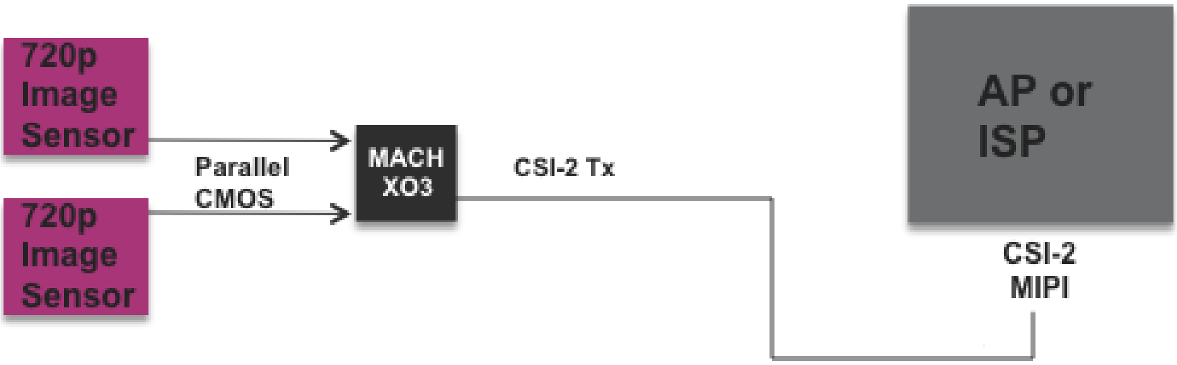


Figure 5: Dual sensor bridge expands existing system functionality.

Conclusion

Developers in almost any applications arena are under constant pressure to use every technology edge to minimize system cost, footprint and power without compromising reliability. The embedded market is no exception. Much like the PC market in the 1990s, today's rapidly growing MIPI-compliant smart phone and tablet market offers embedded designers an opportunity to integrate a broad range of lower cost and proven components if they can resolve compatibility issues. Using a new set of programmable bridging solutions based on Lattice Semiconductor's ULD FPGAs, designers can now eliminate those obstacles and leverage recent advances in MIPI displays, image sensors and application processors to drive down system costs.