# NEW APPROACHES TO HARDWARE ACCELERATION USING ULTRA LOW DENSITY FPGAs

A Lattice Semiconductor White Paper

August 2013

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

## *Introduction*

Ask system designers to list the problems they face – it doesn't matter whether they're building mobile consumer, automotive, industrial, medical or scientific applications – and inevitably they'll mention optimizing host processor performance. It's hardly surprising. The event-driven architecture of these MPUs allows them to multitask and address new priorities as they occur. But as the number of I/O continues to rise, it also places escalating demand on bandwidth. Tasked with managing a wider array of I/O as well as other system-wide command and control functions, today's host MPUs must remain operational for longer periods of time, thereby consuming precious power and compute resources.

Complicating this problem, many of the tasks host MPUs must address today are timing critical; they must be performed in a contiguous manner. Once the processing commences, it cannot be interrupted. Each task must be performed as the data reaches the MPU or the integrity of the data is compromised. Given that many of today's handsets or instruments may feature up to a half dozen or more sensors delivering data continuously over a SPI or I2C interface, performance requirements can climb dramatically.

Historically, designers have addressed these performance bottlenecks and allowed the host processor to devote more attention to command and control functions, by offloading math functions, particularly those that are predictable and repetitive, to general purpose devices such as DSPs or ASSPs. This partitioning of the design reduces processor overhead, but designers typically pay a price in some combination of increased board space, power consumption, cost, and development time. Furthermore, most of these general purpose solutions are far too large and generic to address the specific needs of today's timing critical applications.

Ideally designers require a programmable solution that can be dedicated to a specific task and delivers silicon optimized to perform that task with maximum efficiency in real-time logic.

## *Highly Targeted Solutions*

A new class of Ultra Low Density (ULD) FPGAs, developed by Lattice Semiconductor, offers designers the opportunity to develop highly targeted solutions that can be uniquely optimized for these specific timing critical and coprocessor-type applications. In effect, these new devices allow designers to develop very small, low power solutions aimed at accelerating very specific functions with very specific silicon. Unlike traditional general purpose coprocessors, these extremely small programmable devices offer the ability to implement algorithms that are highly optimized for a particular application and supported by interfaces targeted to that particular function. For instance, these applications might support timing critical functions for motor control or communications, or very specific hardware acceleration functions for sensor preprocessing or image rotation. While these highly targeted solutions offer more efficient bandwidth utilization, they also bring very attractive benefits in terms of lower power consumption, smaller product footprint and lower system cost.

At the same time ULD FPGAs give designers unparalleled design flexibility. By capturing these highly targeted functions in a very small, low power FPGA, designers can reduce core processor overhead opening up new options for system design. In some cases it allows them to use a lower performance and lower cost processor and keep the overall solution budget competitive. In others, it allows them to add new functionality into the system without disturbing the complex coding associated with the legacy core processor.

Finally, unlike traditional large FPGAs, these ULD devices offer designers the ability to deliver highly tailored programmable solutions occupying minimal board space while keeping system power demands and cost as low as possible. Utilizing a unique fabric that can perform functions in parallel, ULD FPGAs deliver excellent performance in devices as small as 1.71-mm x 1.71-mm while consuming an order of magnitude less power than a traditional applications processor.

The following design examples give insight into how ULD FPGAs might be used to address a wide variety of timing critical and hardware acceleration applications.

## *Image Scaling*

In a growing number of mobile applications today image outputs must be resized, either scaled down or up, or presented as picture-in-picture to support new applications or the use of multiple displays. Many embedded, medical and automotive applications, for instance, feature a display of one size on the front of the device and a second, different-sized display in another location on the product. By offloading this image scaling function from the core system processor, a circuit implemented in a ULD FPGA allows designers to use a lower cost core processor and reduce power consumption by minimizing overhead. Alternately, it can permit designers to extend the life of an existing design by adding new functionality without swapping out the core processor or taking on a major re-coding effort. The block diagram below depicts a typical image scaling circuit.

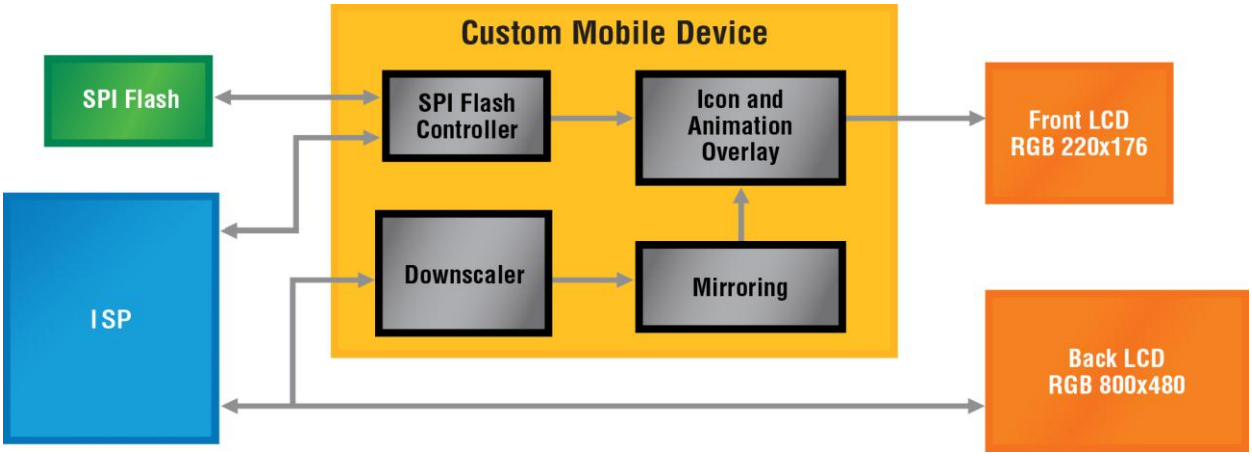


**Fig. 1: Image scaling circuit for a mobile device.**

## *WDR Preprocessing*

Wide Dynamic Range (WDR) preprocessing is a common technique used to improve image clarity. By lighting up dark areas of an image and diminishing the light in bright areas, this process improves the visibility of objects in an image. But this function can be highly compute intensive. The processor must combine short exposure frames and long exposure frames and develop a tone map to modify and improve the image.

Typically this process would require the significant logic resources of a high end Image Signal Processor (ISP).

The block diagram below illustrates a co-processing solution that drives down host ISP performance requirements by offloading the WDR function using an X02-4000 FPGA from Lattice Semiconductor. This highly targeted implementation takes image data from an image sensor with WDR capabilities over a sub-LVDS interface and performs the WDR function in a 4K LUT X02 FPGA. The HD video frame is stored in a low power SDRAM. Once the pre-processing is complete, the image is sent via parallel bus to the ISP. By performing this function before the data reaches the ISP, this highly optimized ULD FPGA allows designers to use a lower performance, more affordable ISP for the rest of the image pipeline.

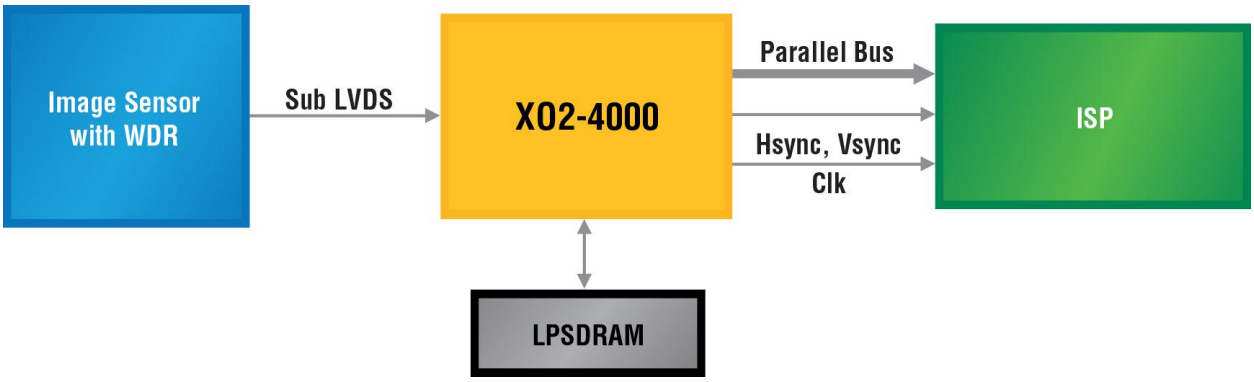


**Fig. 2: WDR coprocessor using a Lattice Semiconductor X02-4000 ULD FPGA.**

## *JTAG Extender*

As communications system functionality continues to grow by leaps and bounds, board design has become increasingly complex. Some communication boards today feature hundreds of components in densely packed layouts. Given this design trend, it has become extremely difficult to establish and use a single JTAG chain for an entire board.

In this example designers simplify the task of developing a JTAG chain for board initialization and test by breaking it into multiple smaller, more easily addressable pieces using an ULD FPGA. This multiple boundary scan Test Access Port (TAP) addressable

buffer is accessed through a standard 1149.1 interface. With three Local Scan Ports (LSPs), the function can be configured as hierarchical ports with the ability to add or remove local scan chains to streamline test flow. Each LSP can also be accessed individually or in combinations of two or three. This JTAG extender can be implemented in a ULD FPGA with less than 200 LUTs.


**Fig. 3: JTAG Extender using a Lattice Semiconductor ULD FPGA.**

## *Image Rotation*

Some lower cost processors used in a wide variety of industrial, medical, scientific and automotive applications are designed to only output an image in either portrait or landscape format. Designers can retain their cost-effective core processor and add image rotation capability by implementing the function in a ULD FPGA from Lattice Semiconductor with as little as 1K LUTs.

In the design below the iCE40 FPGA loads an image in landscape format (400 x 240) and a counter and buffers into SDRAM. Once the full image is loaded, the FPGA reads out the frame in rotated portrait format for display on the screen. The functions across the top of the FPGA manage setup and control. Larger screens would require FPGAs with densities of approximately 4K LUTs.

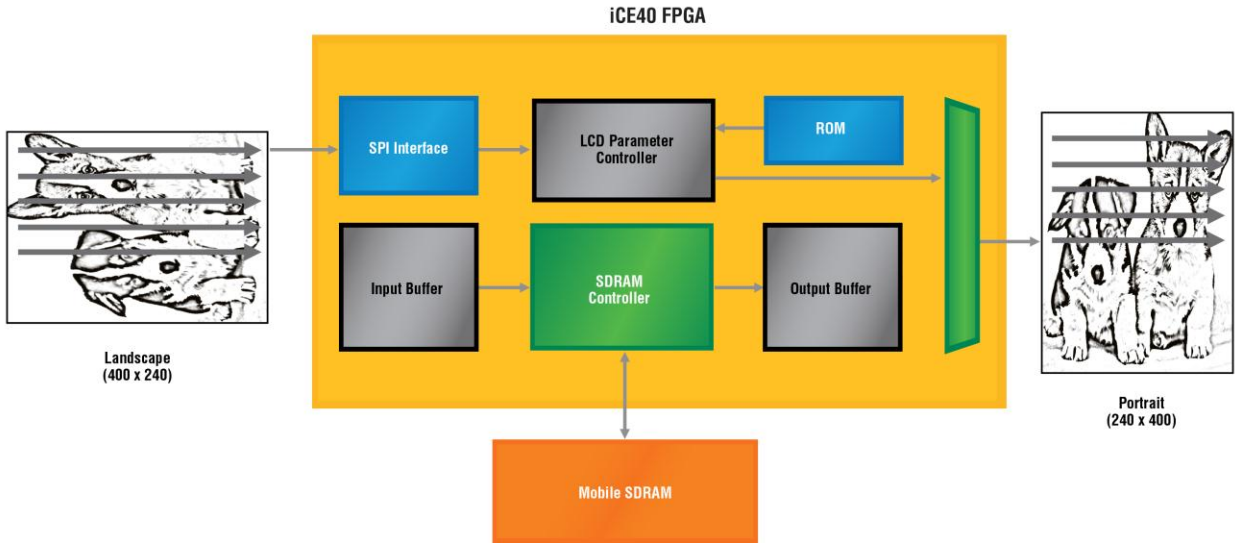


**Fig. 4: Image rotation circuit using iCE40 FPGA.**

## *Live System Updating*

Lattice Semiconductor's ULD FPGAs also bring a unique capability to time-critical applications where systems must be updated on-the-fly while the system continues to operate. Historically, one of the more attractive benefits of FPGAs has always been the ability to reconfigure device functionality without removing the device from the system. However, typically most of the methods used to reprogram FPGAs require a "major" disruption to system operation. How do users of high reliability systems that must be operational 24/7 update the FPGA without interrupting the operation of the rest of the board?

Lattice X02 ULD FPGAs feature a technology called Transparent Field Reconfiguration (TransFR) to help minimize system interruption. A function provided in Lattice's ispVM system software, TransFR makes use of the two sets of configuration storage built into Lattice's ULD FPGAs. Each FPGA maintains a working configuration of the FPGA in its on-chip SRAM while a copy of the configuration is retained in on-chip flash memory. Since the contents of the flash memory can be loaded into SRAM automatically at power up or any other time, there is the no need for external boot memory. FPGAs without internal non-volatile storage configure the SRAM from an external device such as a serial flash, microprocessor or EEPROM. Whether the SRAM is programmed by an

external device or on-chip Flash, the non-volatile storage can be programmed independently of the SRAM and each memory can be modified while the other remains unchanged. In a process called background programming, engineers program the non-volatile configuration memory while the SRAM continues to operate uninterrupted.

Lattice's TransFR technology combines this background programming capability with the ULD FPGA's powerful boundary scan functionality to support system updates with minimal disruption. First, the non-volatile memory is updated while the SRAM runs undisrupted. Next, I/O states are captured and driven to a user-defined level using JTAG commands. While the I/O states are under control, JTAG commands are used to initiate the transfer of new technology of new functionality from the non-volatile memory to the SRAM. Once the SRAM is reconfigured, I/Os are released from boundary scan control and returned to their specified functions.

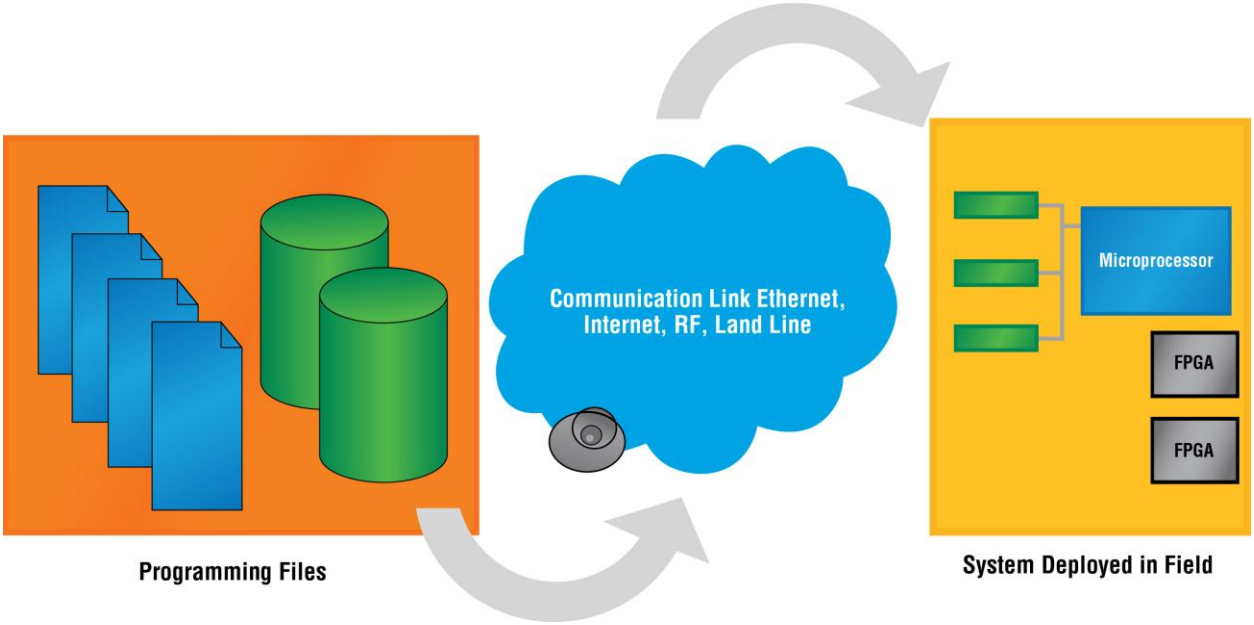


**Fig. 5: Live system updating solution.**

## *Conclusion*

The traditional notion of coprocessing requirements has arguably been summed by the concept, “the more computational power the better.” But many systems built today do not call for a monster coprocessor capable of delivering mountains of performance. More often than not what designers really need are solutions that are expressly targeted at their hardware acceleration needs. By using precisely targeted silicon optimized for a specific function, ULD FPGAs offer designers a new option – the ability to resolve their performance bottlenecks with minimal impact on system footprint, power consumption and cost.