



Innovative Non-Volatile Configuration Memory Technology Enables Low-cost, Multi-time Programmable ULD FPGAs

A Lattice Semiconductor White Paper
Dec 2013

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

It's hardly surprising that flash-based FPGAs have earned a solid reputation since their entry into the market. They offer an impressive array of attractive features. These non-volatile devices retain configuration memory without external assistance and can achieve instant-on, live-at-power-up status as soon as system voltage reaches its minimum level. At the same time flash-based FPGAs offer a high level of reliability for system and mission-critical functions and, given the particular physical characteristics of their silicon, better protection against hacking and other security threats. However, there are significant price and power penalties for using embedded flash in an FPGA.

Moreover, some common misconceptions have emerged about flash-based FPGAs as well. Some developers believe an FPGA can only be programmed up to 10,000 times via embedded flash memory or that applications requiring an FPGA to boot up in-system instantly can only do so via embedded flash. Others believe that the only technology that will support the implementation of a different chip configuration is embedded flash.

Replacing embedded flash

Those ideas are simply not true. In fact, new technologies are emerging that promise to offer many of the inherent advantages of flash-based FPGAs without the technology's relatively high power consumption, footprint and cost. To help achieve that goal, Lattice Semiconductor is leveraging a proven technology utilized in its widely adopted iCE FPGA family that employs a proprietary Non-Volatile Configuration Memory (NVCM) as a replacement for embedded flash. This FPGA architecture is compelling because it uses internal NVCM and SRAM to maximize programmability at a low cost. With a smaller die size, this new technology not only reduces cost, it also helps drive down static and dynamic power, and utilizes compact BGA packaging to shrink system footprint.

Lattice calls its NVCM update process Multi Time Programmability (MTP). Currently MTP technology can be used to store two different configurations of the FPGA. This

does not mean the FPGA cannot be infinitely reprogrammable. While the NVCM cannot be programmed a near-infinite number of times like SRAM, designers can eliminate continual NVCM updates by using different system partitioning schemes. Given that the vast majority of designs use a processor and an extensive memory subsystem that typically includes external flash, all the key components are already resident on the system board. Using MTP designers gain a number of important benefits. A smaller die size than flash memory enables smaller, lower cost packages. It also drives down static and dynamic power.

For system designers looking for new ways to cut their BOM cost, ULD FPGAs with MTP offer a low cost replacement for expensive embedded flash memory. One way this technology cuts cost is by eliminating duplicity in the system. Why use embedded flash in an FPGA when external flash is already resident in a memory subsystem? For example, FPGAs with embedded flash are widely used to store a “golden” image of the design in some form of a non-volatile memory to configure the FPGA on system startup so the system can always return to a known state. With Lattice’s MTP technology, designers can now perform the same function at lower cost while utilizing less space and power. This approach has significant cost implications over the long term. By adopting an architecture that uses MTP and external flash, a designer can cut FPGA and system cost today and into the future because the technology is scalable with lower cost silicon process geometries. And this approach can be utilized whether it is deployed in small consumer mobile devices, large blade servers or process control systems.

System configuration options

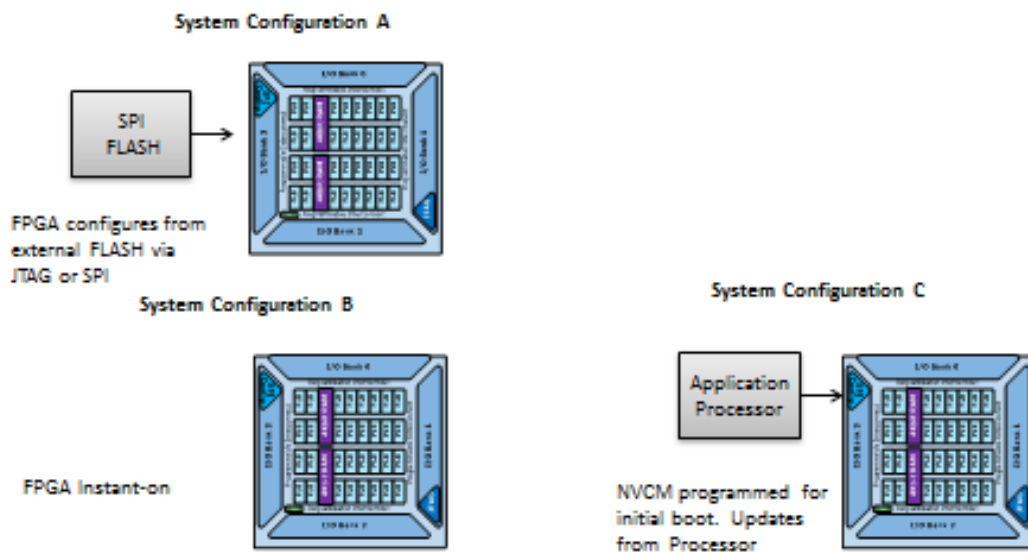
The key to achieving this reduced cost without sacrificing flexibility is the designer’s willingness to develop the hardware and software hooks needed to configure the FPGA from external flash memory. Booting an FPGA from an external memory is a usage model employed widely across the electronics industry. This approach essentially trades off a longer FPGA configuration time for less flash duplicity and lower system cost. For example, a system that uses an embedded memory to configure a 2K LUT FPGA performs that task in less than one millisecond (ms). The same system loading

the configuration data via SPI external flash will perform the task in approximately 18ms.

To see how this would work in different system configurations we explore three common start-up procedures for FPGA system usage:

- A) The FPGA is not needed until after processor is up and running and is configured from external memory.
- B) The FPGA has to start up on its own with instant-on in less than 1ms to ensure it is alive before the processor boots up.
- C) The FPGA has to start up on its own, but it is not time critical.

System Configuration Schemes



When tradeoffs matter

Start-up procedure A is not an issue because the FPGA can wait to be loaded from a processor via the on-board external flash. Start-up procedure B can be addressed by using the NVCM to store a “golden” image to load the FPGA. This initial image allows the system to start up in a known good state and then allows the processor to update the FPGA if newer code is available in external flash. In those instances when the system demands that the FPGA must start up with a complete set of the latest code, this procedure may require an FPGA with embedded flash. In those cases, the designer will have to bear the higher cost and power consumption of an embedded flash FPGA.

Systems employing start-up procedure C can use either of two strategies. They can load a “golden” image from NVCM and then allow the processor to perform a code update via external flash, just as in case B. Or they can load the FPGA from a dedicated SPI flash memory. While this adds a small cost to the design, it also simplifies the solution.

Table 1: Comparing System Configurations

Configuration	A	B	C
Non Volital Memory	External FLASH	Internal FLASH	NVCM & External FLASH
Pros	Low Cost, Lower Power	Faster FPGA Configuration Time	Flexible Solution, Lower Power
Cons	Slower FPGA Configuration Time	Higher Cost, Higher Power	Slower FPGA Configuration Time

In options B and C designers must take care to ensure the FPGA is properly updated from a processor and external flash. It is extremely important that the FPGA update process does not obstruct the operation of the rest of the system. Designers can use

several approaches to enable those hooks. For example, peripheral chips that could be impacted can be placed in a wait state or sleep mode, or held in reset.

Some developers will balk at the idea of implementing the hardware and software hooks in their system to access external flash for this function, particularly those designing high reliability solutions in infrastructure and server backplane applications. Yet the use of external flash to configure FPGAs via JTAG is a standard practice in the system development cycle. Therefore, many of the challenges developers fear have already been resolved. Once a system moves into production and must be updated in the field, it will likely require more than two updates. In that case designers can still reduce costs by utilizing non-volatile memory resident in the system to run those updates. But they may need to retain intelligence implemented during development so the system recognizes when it has exceeded its second update and must access data from an off-chip memory.

Conclusion

Today designers of systems of all sizes, from servers and industrial systems to instrumentation and consumer goods, face the same constraints. They are continually under pressure to shrink system size and weight, minimize component cost to remain competitive, and rein in power consumption to extend system life and reliability. For designers willing to leverage existing on-board external flash memory and the processor, Lattice's MTP technology offers an exciting new way to replace expensive flash-based FPGAs with ULD FPGAs and, in the process, optimize system footprint, cost and power.