Combining Low-Cost & Non-Volatility To Deliver No Compromise FPGAs

A Lattice Semiconductor White Paper

February 2005

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

The Unmet Demand For Low-Cost, Non-Volatile FPGAs

Although SRAM FPGAs dominate the FPGA market, the vast majority of designers would prefer a non-volatile, reprogrammable FPGA solution -- provided the associated cost premium is not too great. This is illustrated in Figure 1, which summarizes the results of a Lattice Semiconductor survey of FPGA designers. Their desire for a non-volatile solution is driven by multiple factors, including the need for:

- Smaller board area and the simplicity of a single-chip solution
- Rapid availability of logic after power-up
- Higher security than is possible with traditional FPGAs
- Real time reprogrammability

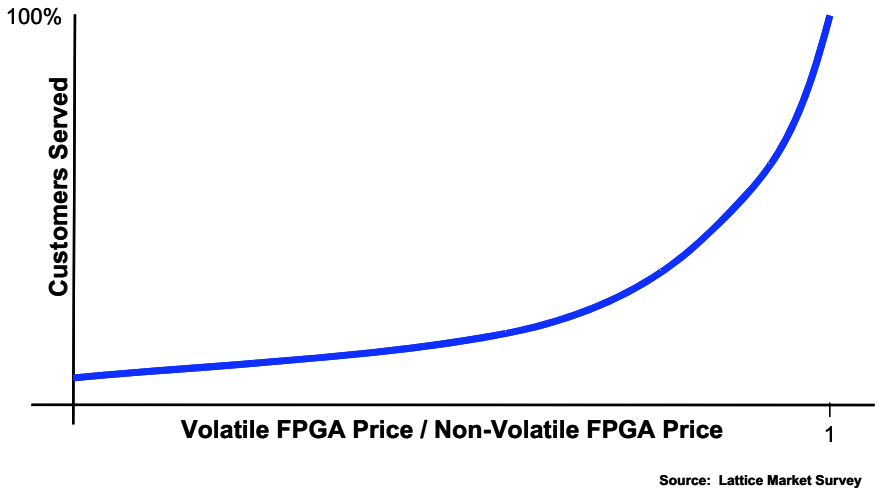

Figure 1 – Demand for Non-Volatile Reprogrammable FPGAs

LatticeXP (eXpanded Programmability) FPGAs deliver the benefits of non-volatility at an economical price point. This has been achieved by combining a low-cost 130nm embedded FLASH technology with the optimized FPGA architecture found in LatticeEC (EConomy) FPGAs. This combination has enabled Lattice to reduce the die size over 80% between its first generation non-volatile FPGAs and the LatticeXP devices. This is illustrated in Figure 2.

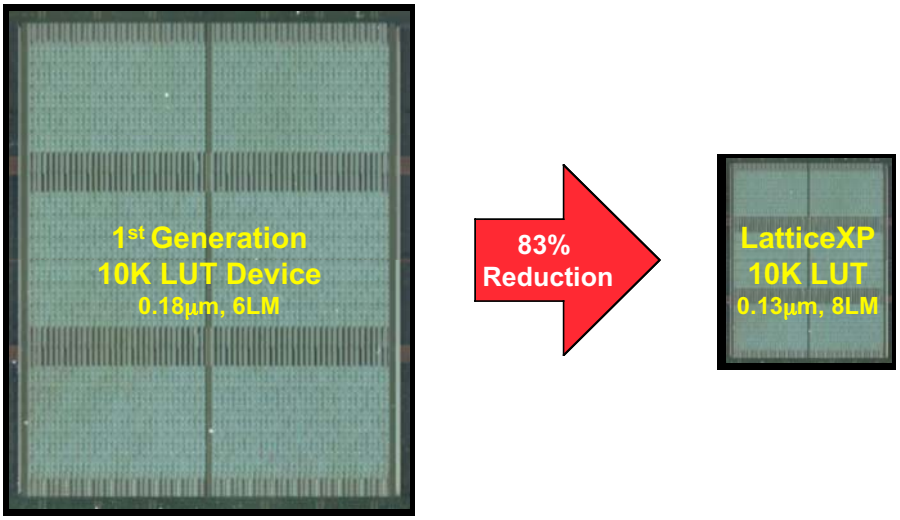

Figure 2 – 2nd Generation Lattice Non-Volatile FPGAs Die Size Reduction

Applications For Non-Volatile, Reprogrammable FPGAs

Non-volatile, reprogrammable FPGAs are well suited for implementing system logic in a wide variety of end markets including communications, consumer, industrial, computing, military and automotive. These FPGAs are especially well suited when there is a requirement for the rapid availability of logic, reduced parts count, high-security or real time programming.

Rapid Availability of Logic

On-chip, non-volatile memory typically allows devices to be ready for operation within one millisecond of power good, in contrast to SRAM-based FPGA devices that typically require tens or hundreds of milliseconds for configuration. The rapid logic availability of non-volatile FPGAs is a desirable characteristic in many common applications:

System Heartbeat Logic: Most systems have critical control logic that must be operational prior to reset being released for the microprocessor. Typical functions include power-up control logic and microprocessor address and bus decode. The rapid availability of logic that non-volatile devices provide makes them ideal for these applications.

Soft Event Upset Control: As silicon geometries shrink and FPGA complexity increases, designers are increasingly concerned about soft event upset (SEU). A common technique for managing SEU is to periodically reload SRAM configuration bits. Non-volatile devices allow this to be done in a short period of time, minimizing system downtime and the corresponding impact on performance.

Rapid Power Cycling For Lower Power: Many system designers turn off devices in their systems for some percentage of the system duty cycle in order to reduce overall power consumption. Non-volatile devices, with their rapid availability of logic after power-up, are ideal for these applications. SRAM-based FPGAs, however, consume a substantial portion of the duty cycle for configuration, as illustrated in Figure 3.

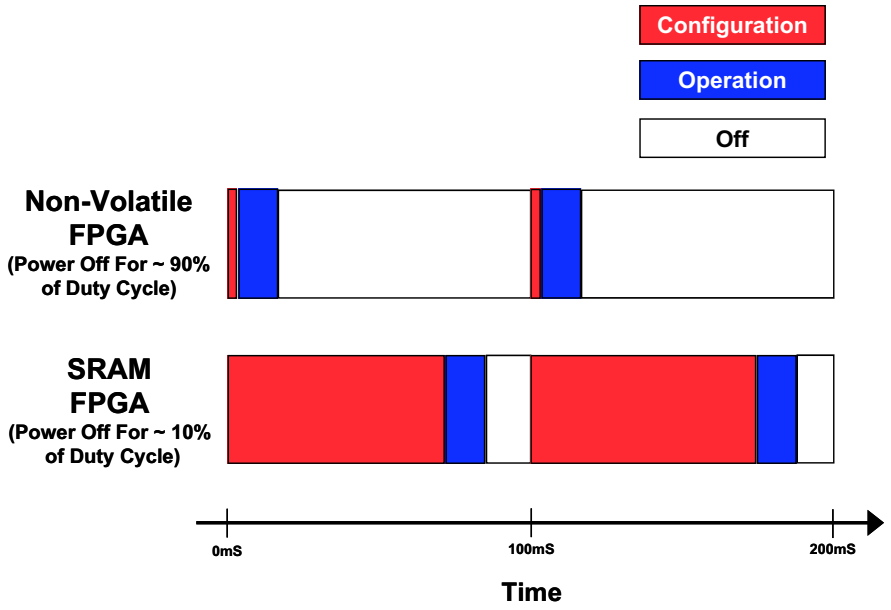

Figure 3 – Improving Power Off Duty Cycle With Non-Volatile FPGAs

Single Chip

Traditional SRAM FPGAs require a boot memory to load the SRAM configuration at power-up. Sometimes configuration loading is done via the on-board microprocessor, while in other applications a stand-alone boot memory is required. Neither solution is ideal. Booting from the system microprocessor introduces additional interdependencies between hardware and software development. It also requires the microprocessor to be up and running prior to the configuration of the FPGAs, precluding their use for system

heartbeat functions. Using a stand-alone boot memory increases the board area footprint of the solution and the bill of materials (BOM), as well as associated costs. By integrating the boot memory on-chip, non-volatile FPGAs provide an alternative and more elegant solution.

Security

In today's complex systems, FPGAs are being used increasingly to replace functions traditionally performed by ASICs and even microprocessors. Ten years ago, the FPGA was at the periphery of the system--today it is at the heart. With current FPGA technology gate counts running into the millions, FPGAs are an attractive target for piracy. FPGA designers are increasingly worried about issues such as:

- Cloning
- Reverse engineering
- Overbuilding
- Theft of service

The SRAM FPGAs most commonly used by system designers need to be configured from a boot device every time the system powers up. This link between the boot device and the FPGA represents a significant security risk because the configuration data is exposed and vulnerable to piracy during system power up. Using non-volatile FPGAs eliminates this security risk.

Real Time Programming

It is increasingly common for designers to develop systems in which the programmable logic devices (PLDs) are updated in real time during equipment operation. Often the equipment is connected to a communication link and updated data and commands are issued via that link. In other cases, a disk or other media is used to provide the update information. This approach provides the ability to update logic in the field: a capability that obviously can be used to fix bugs in logic. As important, though, it also provides a simple means to respond to changes in standards and to add features and capabilities in pay for services environments.

Devices that combine non-volatile and SRAM configuration bits on the same chip are well suited for these applications because they allow the FLASH memory to be updated in real time while the device is operating from the SRAM configuration. The new configuration can then be applied rapidly, either at user command or during the next power cycle. This concept is shown in Figure 4.

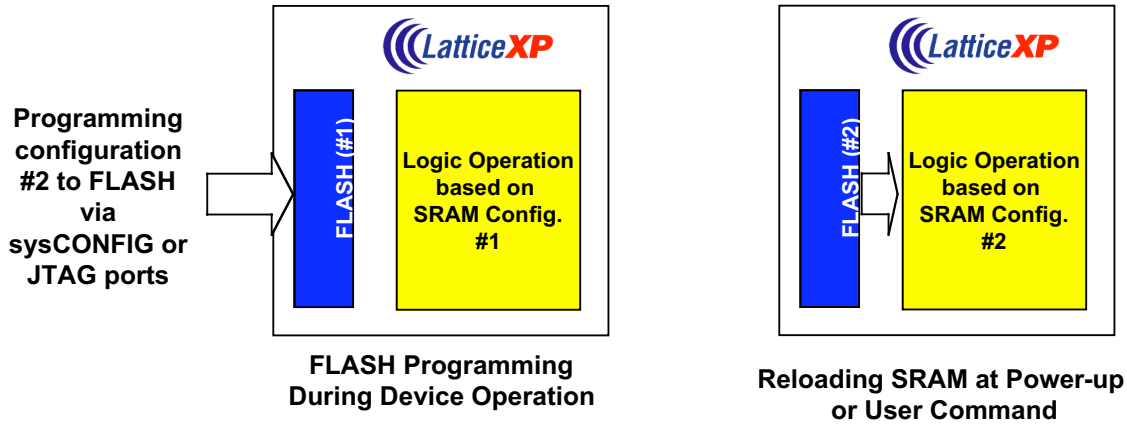

Figure 4 – Real Time Programming

The LatticeXP Device Family

LatticeXP devices add blocks of non-volatile memory to an optimized FPGA architecture to provide a low-cost, non-volatile solution. The devices are manufactured on an economical 130nm Flash process, which provides good intrinsic soft error rate immunity.

LatticeXP Architecture

At the core of the LatticeXP devices are Programmable Function Units (PFUs) that allow the implementation of logic and, for 25% of the blocks, distributed memory. Logic is implemented using four input look-up tables (LUT-4s) and register pairs, which is the *de facto* standard for the FPGA industry and well understood by system designers and logic synthesis tool suppliers.

Distributed memory provides an efficient method for designers to implement small blocks of scratch pad memory. Rows of sysMEM Embedded Block RAM (EBR) provide 9kb blocks of memory for use implementing larger memory blocks. At the end of the

rows of sysMEM memories are sysCLOCK PLLs that allow clocks to be aligned for improved set-up and clock-to-out times and new clocks to be synthesized.

Around the periphery of the device are sysIO interfaces that allow the device to interconnect to a variety of I/O standards, including LVCMOS, PCI, LVTTL, LVDS, SSTL and HSTL. Additionally, LVPECL, BLVDS and RSDS interface standards can be emulated with the addition of external resistors. DLL calibrated DQS delay blocks, DDR registers, and clock transfer circuitry allow the easy implementation of high performance DDR memory interfaces up to 333Mbps. Generic DDR interfaces up to 700Mbps can also be implemented with the device. The various functional blocks of the architecture are interconnected with a routing fabric that provides an optimum balance among speed, flexibility and cost. Figure 5 illustrates the overall architecture.

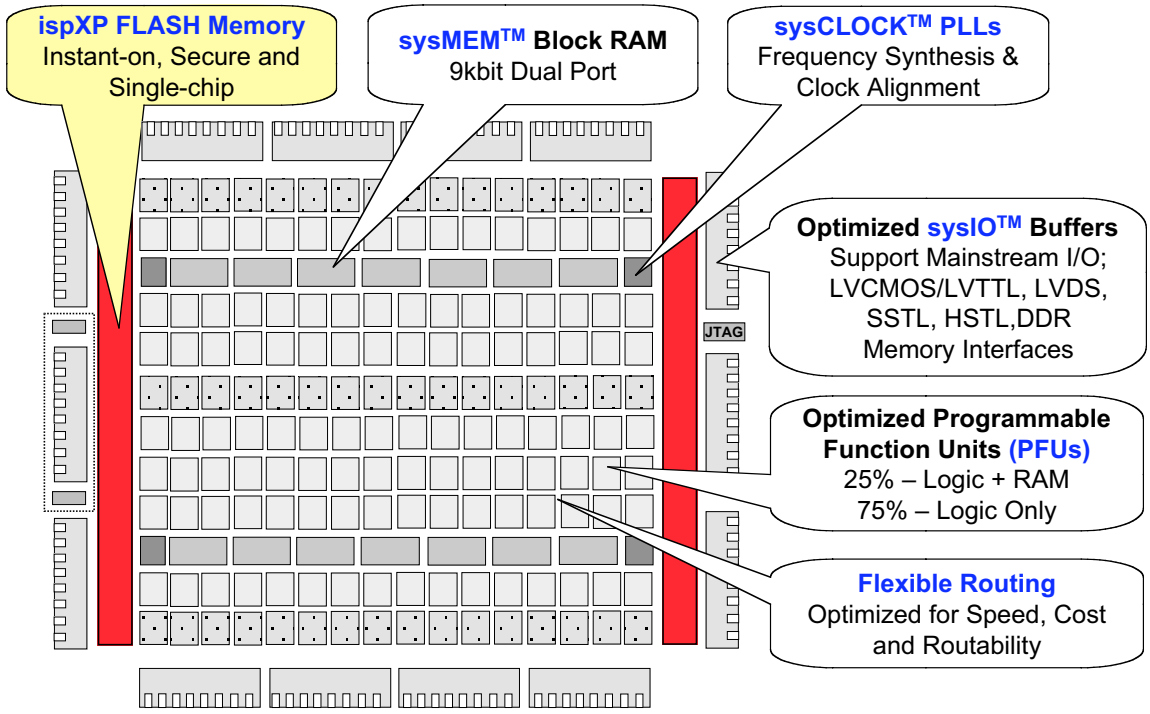

Figure 5 – LatticeXP Architecture

Non-Volatile and SRAM Memory

SRAM configuration bits control the operation of the LatticeXP devices. At power-up, these bits are loaded via the on-chip, non-volatile memory, resulting in logic availability in less than 1ms after power good. During device operation the SRAM may be

reconfigured from the FLASH by toggling a pin or issuing the correct commands through the device configuration ports. Both the FLASH memory and the SRAM memory can be reprogrammed/reconfigured via either a JTAG port or a sysCONFIG port (microprocessor style interface). Configuration of the SRAM via these ports takes between tens and hundreds of milliseconds, depending on the chosen interface. The FLASH memory can be programmed in as little as 2 seconds. Figure 6 shows the operation of the different memories within the LatticeXP devices.

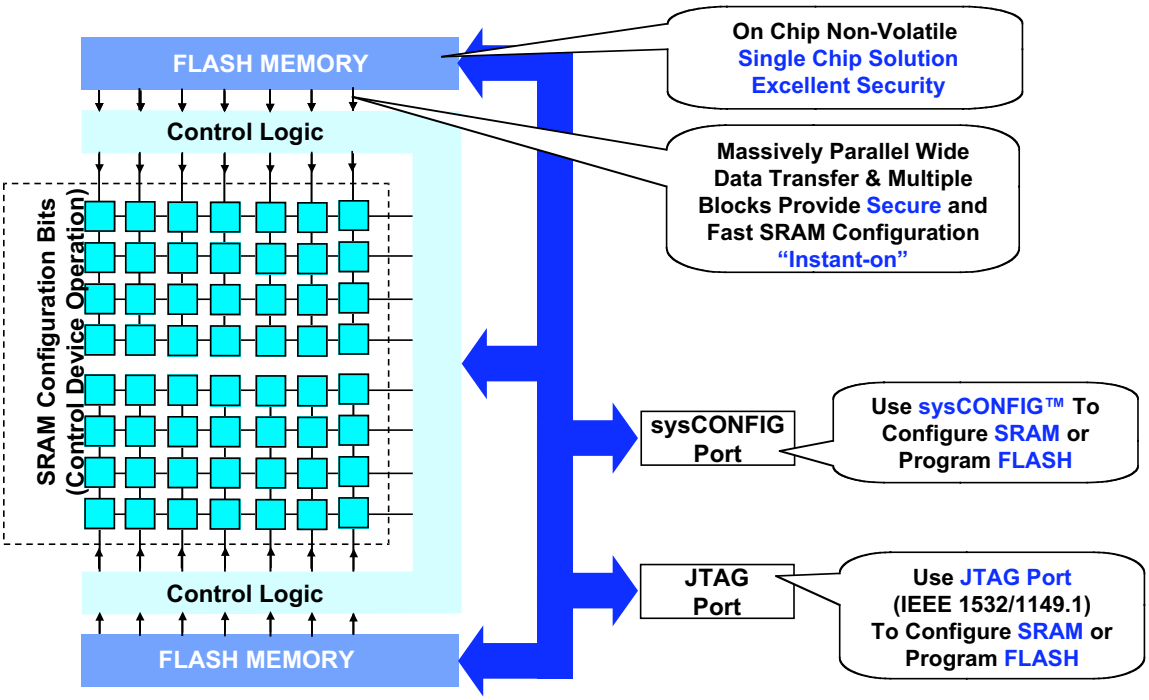

Figure 6 – Configuration Memories in the LatticeXP

Device Family

The LatticeXP family supports logic capacities between 3K and 20K Look-up Tables (LUTs) and 62 to 340 I/O in a variety of popular package options. All devices in the same package are pinout compatible, allowing device capacity to be adjusted to match changing design requirements without re-laying out the circuit board. The devices are available in two power supply options, one supporting 3.3/2.5/1.8-volt operation and the other supporting 1.2-volt operation. The higher voltage version allows designers to benefit from the lower power, higher speed and lower cost of 130nm technology without

migrating to lower power supply voltages. Table 1 shows the different devices available in the LatticeXP family.

Device	XP3	XP6	XP10	XP15	XP20
LUTs (K)	3.1	5.8	9.7	15.4	19.7
sysMEM Blocks	6	10	24	32	46
sysMEM (Kbits)	54	90	216	288	414
Distributed RAM (Kbits)	12	23	39	61	79
Voltage (V)	1.2/1.8/2.5/3.3V				
PLLs	2	2	4	4	4
Package I/O Combinations					
100-pin TQFP (14x14mm)	62				
144-pin TQFP (20x20mm)	100	100			
208-pin PQFP (28x28mm)	136	142			
256-ball fpBGA (17x17mm)		188	188	188	188
388-ball fpBGA (23x23mm)			244	268	268
484-ball fpBGA (23x23mm)				300	340

Table 1 – LatticeXP Family Members

Summary

Non-volatile, reprogrammable FPGAs, available at a minimal cost premium compared to SRAM FPGAs, represent the “affordable ideal” for many system designers, particularly those concerned with rapid logic availability, real time programming, security, board area and part counts. The LatticeXP FPGA combines a low-cost architecture with a 130nm FLASH process to deliver a unique combination of non-volatility and low-cost to FPGA designers.

###