

Short Stack with Syrup

Looking between the layers of the non-volatile Spartan-3AN family.



by Kevin Morris
Editor – FPGA and
Structured ASIC Journal
kevin@techfocusmedia.com

Non-volatile FPGA is one of today's oddest market segments.

Bound by the ill-defined characteristic of “non-volatility,” the field of available devices is diverse from the ground up, with strikingly different architectures, approaches, and benefits.

Actel, long the leader in non-volatile FPGAs, began with antifuse technology (which is one-time programmable) and followed later with flash-based FPGAs. QuickLogic fielded an antifuse-type approach from the beginning, and Lattice Semiconductor joined the fray over a year ago with their LatticeXP hybrid devices, embedding a flash boot PROM on the FPGA for rapid, on-chip configuration.

With the new, non-volatile Spartan™-3AN family, it is clear that Xilinx approached the non-volatile problem differently. Often, on your board, you'd pair a low-cost SRAM FPGA like a Spartan-3 device with a commodity flash memory for storing the bitstream and possibly other user data. Xilinx just took the die for that flash memory and the die for that FPGA, stacked them using packaging techniques now common in cell phones and other space-constrained devices, and dropped them into a single package. Voila! A non-volatile FPGA.

Hey, wait, that's totally cheating, isn't it? Does stacking the two die vertically instead of horizontally on your board somehow change a regular-old volatile SRAM FPGA into a newfangled non-volatile one? Aren't



there rules about these things? Somebody bring in a referee. What's the catch?

In reality, there is no “catch.” Like any engineering solution to a problem, there are trade-offs. To understand the ones involved in Spartan-3AN FPGAs, we need to examine our possible reasons for wanting a non-volatile FPGA in the first place, and then see how the various options stack up.

As long as we don't pop open the package and start asking too many questions (we'll discuss opening the package later, so keep those Dremel tools powered down for the moment, lil' hacker dudes), we can assume that a non-volatile FPGA is one that doesn't require an external boot PROM. You apply power to the pins and (after some time passes) the FPGA is ready to go, independent of any other devices on the board.

Why do we care? Well, we might want the smaller footprint of a single chip. We might want simpler board design. We might hope for lower power consumption. We might want additional design security. We might be trying to reduce total BOM cost. We might want instant (or near-instant) power on. Compared with Spartan-3A devices, how does the new Spartan-3AN family do on these issues? Yes, yes, no, sorta, and probably not.

Xilinx Spartan-3AN FPGAs are available in a subset of the package options of the Spartan-3A family. Probably because the stacked-die packaging process is more expensive, Xilinx isn't rolling out the entire fleet of packages for Spartan-3AN devices. The

company says they will probably be adding additional packaging options based on customer demand, so if the package you want is not on the list – vote early and often. The good news is that the package you choose will be the only FPGA-related thing you have to buy, inventory, place, and connect to your board. You won't need a separate boot PROM as you would with an SRAM-based FPGA. Board real estate and complexity is definitely reduced.

The FPGA portion of a Spartan-3AN device has the same power consumption as a corresponding 90-nm Spartan-3A device because – hey, it's the same die. As a review – Spartan-3A devices have some nice power-saving features for an SRAM FPGA. Its 90-nm technology gives it pretty good dynamic power consumption. In addi-

tion, Spartan-3A devices (and thus also Spartan-3AN devices) have two additional power modes called “Suspend” and “Hibernate” that allow you to drop static (standby) power by 40% with a fast wake-up time (suspend mode) or by 99% with a “normal” wake-up time (hibernate mode). This means that the device is still more power-hungry than other super-low-power non-volatile FPGAs like QuickLogic’s PolarPro or Actel’s Igloo, but it maintains other advantages. (Remember what we said about trade-offs?)

The flash portion of the Spartan-3AN FPGA will, of course, have the same power profile as the commodity flash memory device that it is. If you dropped one of those on your board next to a Spartan-3A device, your total power characteristics should be almost identical. One could argue that having the flash separate gives you more options on the flash side than defaulting to the built-in flash in Spartan-3AN FPGAs. If one argued that, one would be correct.

Security Risks

Design security is one of the issues most often cited for wanting a non-volatile FPGA over a conventional SRAM design. The important thing to remember here is that security is relative. Every option available on the market provides differing amounts and types of security. Almost nothing can protect against the best-funded security risks such as those funded by governments. As you move down the scale on attackers’ budgets and time constraints, however, security measures start to shake out at different levels.

The biggest design security risk with SRAM-based FPGAs is the availability of the configuration bitstream going between the boot PROM and the FPGA during startup. Since FPGAs are standard parts, if the contents of the boot PROM can be cloned, the design is stolen. The normal way to deal with this is to encrypt the bitstream in the boot PROM, and to have an encryption key stored in the FPGA that will allow the bitstream to be decrypted upon load. The problem with SRAM (and non-volatile devices in general) is that any encryption key

stored in SRAM is lost when the device is powered down. This leaves us with a problem in moving encrypted bitstreams that has been addressed two ways: 1) by attaching an external battery to the device to maintain the encryption key and 2) by adding permanent fuses to each device for storing the encryption key.

The Spartan-3AN device does neither of these. Not exactly, anyway. There is a permanent serial number stored in each device, and that serial number can be used to implement various security schemes as you see fit. It is not, however, used to decrypt the incoming bitstream. Bitstream protection is done by obscuring the connection between the FPGA and the boot PROM (now stacked on top of the FPGA inside a single package). While more secure than a separate boot PROM, the scheme is still far from impermeable as far as configuration stream protection goes. (OK, you can fire up those Dremel tools now.)

Because the Spartan-3AN FPGA has room for two complete configurations in the flash, it’s obvious that Xilinx thinks you’ll be updating the bitstream at some point in the product life cycle. The Spartan-3AN device even has a nice feature that’ll let it boot on the original “golden” bitstream if a bad download or other corruption causes the new, updated one to malfunction. The problem is, any mechanism you have for delivering an updated bitstream to the device will be snooperable by prying eyes (or prying logic analyzers, really). The bitstream obscuring method doesn’t hold up as well when the bitstream has to travel from your home base to the device in the field. Be sure to design your security scheme accordingly.

As far as the lower cost question goes, you’ll probably pay a bit of a premium for Spartan-3AN devices compared with the same Spartan-3A device with the same flash memory. When you factor in the other costs of an additional device such as board area and inventory and such, it may be a wash. On the positive side, you certainly won’t be taking a big BOM hit for using the new family. Xilinx projects prices in the \$5 range in volume.

It’s also important to look at Spartan-3AN devices in comparison with other non-volatile options. After all, from a marketing perspective, the Spartan-3AN FPGA is probably most important because it gets Xilinx into the non-volatile FPGA race. Before, a list of non-volatile FPGA vendors would have included Actel (for their antifuse and flash-based devices), QuickLogic (for their antifuse devices), Lattice (for their LatticeXP hybrid devices) and arguably even Altera for their Max II (which is really an FPGA marketed as a CPLD). Now, that list includes Xilinx with the Spartan-3AN family.

With a 90-nm SRAM-based FPGA at its core, Spartan-3AN devices will be faster and have higher density than the other entrants. Because it uses commodity flash bonded to the FPGA, it will offer more non-volatile storage than the other entrants. On the down-side, it will take longer to configure (in the range of 100 milliseconds because of streaming the configuration over a serial flash versus microseconds or even near-zero for truly non-volatile devices like antifuse and Actel’s flash). It will likely be less secure than the other options, and although its power consumption will be very good, it will consume more power in both active and standby modes than the other non-SRAM alternatives.

The Spartan-3AN family offers devices ranging from 50K to 1.4M “system gates” (corresponding with Spartan-3A devices XC3S50A to XC3S1400A) with flash memory ranging from 1M to 16M. Three of the devices, the XC3S200AN, XC3S700AN, and XC3S1400AN are available now, with the remaining devices (XC3S50AN and XC3S400AN) slated for second quarter.

Xilinx Spartan-3AN FPGAs are likely to be popular in space-, cost-, and power-constrained high-volume applications that need some of the features of non-volatile FPGAs with the performance and density of leading-edge SRAM devices. In addition to bringing Xilinx officially into the non-volatile game, the unique characteristics of the Spartan-3AN family gives us another valuable trade-off point in our FPGA selection. ●●●