Customisation Of Arm Processor In Mixed Signal FPGA

Deepa Nath
Dept.of E&TC-MITCOE

Prof.V.V.Shete
Dept.of E&TC-MITCOE

Mr. Abhay Tambe
Managing Director

Abstract

Implementing designs on devices with robust security mechanism against theft and piracy is essential. Protecting the design through its entire product life cycle is important. So soft core implementation in the device along with hardware is an added feature. This paper discusses the Fusion series mixed signal FPGA-as a single chip solution. Single chip solution provides better EMI/EMC, reduced failures and improved noise immunity.

1. Introduction.

Customers demand for a new product to be launched in market in a shorter time span and at a low cost.The manufacturer who does this has a greater market share.
Competitors to achieve this target try to copy the products looking at the hardware and rewrite the code and sell at a lower price. So with this they bring the product in market in lesser time and at a cheaper price. Such malpractices can be stopped if we preserve the product in the form of Intellectual property and protect the product from being copied.
Over the past two decades, the FPGA has transitioned from a prototyping tool to a flexible production solution in both consumer and industrial applications. With FPGA logic complexity increasing from a few thousand gates to millions of gates, the devices can hold more of the key functions (i.e., the intellectual property, IP) of a system.
Today, designers can select FPGAs that employ various technologies to hold the configuration data—OTP (onetime programmable) antifuses, reprogrammable flash-based storage cells, and reprogrammable SRAM-based configurable logic cells. Both antifuse- and flash-based solutions provide relatively secure solutions, since the configuration data is stored on the FPGA chip
and there are mechanisms that prevent the stored data from being read out.

Having a core inbuilt in a FPGA and being offered as system on chip solution would help to achieve hardware privacy along with soft-core protection and this would definitely offer as an added feature for the industry people to prevent their product from getting copied. Single chip solution offers better EMI/EMC, improved noise immunity, reduced failures.

Secondly Microprocessor obsolescence[12] is a major concern for many companies. Programmable logic can provide a viable solution to this problem. By using soft core microprocessors embedded within a programmable logic device, one can own the processor core for use in any future devices and platforms also the design can be both flexible and scalable to suit different platforms.

2. Design Integration.

To reduce overall product costs, a route favoured by many companies is to devote time to reduce overall component count and reduce the number of devices on the Bill of Materials (BOM). Design integration (that is, the process of reducing component count by putting the effective system onto one chip) is one such method. Currently, this can be achieved by integrating many device-level functions into a single ASIC, trying to put as much functionality into software as possible or integrating many functions into an FPGA. While in the past the ASIC route was the natural choice this route is reserved for those who have huge production runs and are not impacted by time-to-market pressures. Choosing the correct

balance of processor peripherals might also be difficult because without going to a custom-made processor it is difficult to avoid wasted functions and hence wasted silicon.
Design integration delivers:
• Reduced BOM
• Reduced cost of order and inventory management
• May reduce PCB complexity and hence PCB cost

FPGA design integration also delivers:
• Faster time-to-market
• Same PCB for many projects
• Flexibility to change designs at any stage even in the field

Design Changes

Reconfigurable platforms cleverly partitioned between software and reprogrammable hardware will allow the designer, to change your choice of system bus or interface late in the design process and even in production.
The reconfigurable system concept also enables different standards and protocols to be tried, tested, and put on trials and if you don't find them to be suitable, you can just load in another bus interface and try it out until you find the best configuration.

Eliminating Processor Obsolescence

The biggest obsolescence headache is that of out-of-date microprocessors and micro-controllers [11]. Processors have shorter life spans than ever and are discontinued at short notice driven by the consumer market trends and the ever present need for speed enhancements.
Even if the design has been coded in "C" (which is always promoted as being "portable code"), verification (and, sometimes, architecture-specific) instructions and features can hamper the move from an obsolete processor to a next-generation device. The change-over process is further exacerbated by different package options and I/O configurations necessitating the need for a complete board re-spin.

Cost of Processor Obsolescence

There are several solutions to the problem of processor obsolescence. The applicability of any given solution depends upon a number of variables, including the value of the application software, the projected life of the system, and the amount of time and money available to solve the problem. The most radical and most expensive solution is to redesign the system around a new processor. Depending upon the volume of the code, a redesign can cost hundreds of man-years of time, much of it devoted to validation and testing.

Soft Processor Solution

A radical but robust new solution is emerging to eradicate processor obsolescence and preserve many years of legacy code and development. The new way is to own the soft processor core and embed it in FPGA fabric. Not only can you port the core to multiple FPGA platforms but you can "design" the peripheral set to meet the exact design requirements, thus eradicating architecture compromises and wasted peripherals.

For example, the designer might desire a processor with perhaps 05 UARTs, an interrupt controller and access to a block of external FLASH. While many off-the-shelf processors exist that would offer multiple UARTs and the other desired peripherals, they would typically be of sufficient complexity to have numerous other peripherals that would be unused in this system. Here we can now start with a processor core and build the peripheral set to meet their exact requirements. Silicon waste is reduced to zero since the designer will only implement what they need. Software design complexity is reduced because no code need ever be written to disable unwanted processor functionality. Even if after ten to fifteen years of field use, when the FPGA hardware might itself be nearing the end of its life, then the soft processor core can simply be dropped into its new FPGA "host" utilizing the same C code and almost all of the same hardware design files as well. The hardware platform might need some PCB modifications, but the legacy code remains usable and intact.

One of the considerations a designer has to face in designing an embedded processor is whether to use a soft or hard core. Soft cores or portable logic blocks, have many advantages over hard cores for some applications. Flexibility is the main advantage. A soft core processor has configurable parameters that enable to meet a variety of application needs. If one buys an off-the-shelf solution, there will be tradeoffs between the design goal and what is pre-canned in the off-the-shelf product. One may end up paying for peripherals that they don't want or they may end up with larger peripherals than they need for the specific task. In both cases one is wasting silicon and wasting

money. With a soft core we can get and pay for only what one wants.

3. Cortex-M1 Processor.

Cortex-M1 is a general purpose 32-bit microprocessor that offers high performance and small size in FPGA runs a subset of the Thumb-2 instruction set (ARMv6-M)

The Cortex-M1 is ARM's smallest processor core and it is optimized specifically for FPGA implementation. It is upwards code-compatible with the other Cortex processors. The Cortex line consists of three series of processors: A series (applications), R series (real time), and M series (microcontroller). All share the Thumb-2 blended 16/32-bit ISA meaning that we can run a full 32-bit implementation or narrow the width to conserve resources like, for example, on-chip FPGA memory, and still not run into instruction set problems.

The main blocks in Cortex-M1 includes the processor core, the Nested Vectored Interrupt Controller (NVIC), the AHB interface, and the debug unit. The processor core supports 13 general purpose 32-bit registers, including a Link Register (LR), a Program Counter (PC), a Program Status Register (xPSR), and two banked Stack Pointers (SP).
When combined with FPGA the small, fast and highly configurable Cortex-M1 processor offers a number of benefits

4. Mixed Signal FPGA.

Fusion mixed-signal FPGAs provide a unique solution, combining programmable logic, RAM, flash, and analog in a single chip. They range in density from 90,000 to 1,500,000 system gates

M1-enabled Fusion devices offer all the benefits of Actel nonvolatile, flash-based families: single chip, reprogrammable, live at power-up, secure, firm-error. Fusion series integrates configurable analog, large flash memory blocks, comprehensive clock generation and management circuitry and high-performance, programmable logic in a monolithic device.

Fusion offers up to 30 high-voltage-tolerant analog inputs, which enable direct connection to signals from –10.5 V to +12 V, eliminating the need for signal preconditioning. The Fusion analog-to-digital converter (ADC) is configurable and supports resolutions up to 12 bits, and sample rates up to 600,000 samples per sec.

Integration Advantages of Mixed Signal FPGA

- BOM integration
- Board design Simplification
- Reduced Points of Failure
- Reduced thermal loading
- Increased Design Flexibility
- Reprogramability

Features: Integrated A/D converter (ADC) with 8,10 and 12-bit resolution and 30 scalable analog input channels
• ADC accuracy better than1 percent
• On-chip voltage, current and temperature monitors
• In-system configurable analog supports a wide variety of applications
• Up to 1 MB of user flash memory
• Extensive clocking resources
• User nonvolatile FlashROM

5. Implementation Scheme.

The Cortex M1 system is build with help of smart design and the application code is written in the soft console software.

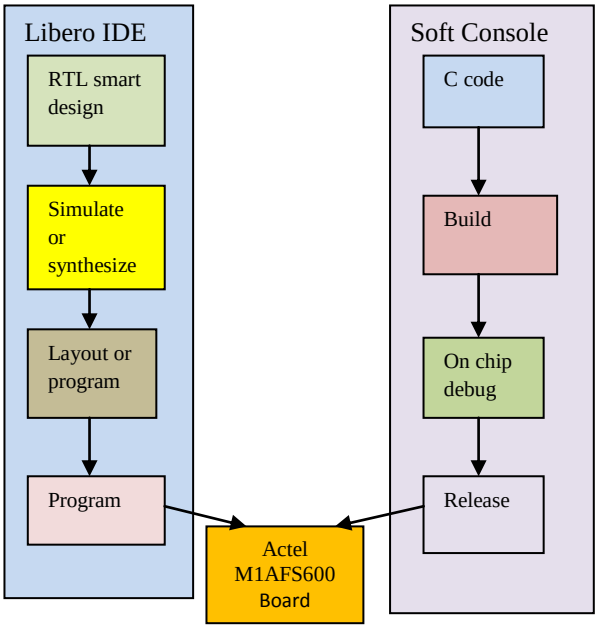
Fusion Board Description.

The Cortex-M1 processor system uses CoreAI, which allows the processor to configure, control and interact with the Analog Block inside the Actel Fusion FPGA. The UART in the system connects to an off-chip USB-to-UART chip, which allows communicating with the target system via a COM port on our machine (using HyperTerminal). Also included are 4 output bits to LEDs and 2 input bits from push-buttons or DIP switches.

Fusion board hardware details

- OLED display for display application
- 40 MHz oscillator for global clock input
- Active low reset input with Reset switch and LED indication
- On-board memory: 8-Mbit (256x32) SRAM
- 4 LEDs for output indication (driven by I/Os of the device)
- Two switches for I/P generation (provide input to device)
- Variable voltage potentiometer for analog voltage input.

- Temperature sensor for temperature measurement.
- On-board FlashPro3 for programming and debugging M1AFS part.
- Serial-to-USB adapter for communication with PC USB port.



" Figure 1. Hardware and Software tool flow"

Tools for customization.

a) Libero IDE.

The Libero design contains the top-level file, other source files, the Smart Design subsystem, constraints files, and builds scripts. These are used by Libero to compile, synthesize, and place-and-route the sample design.

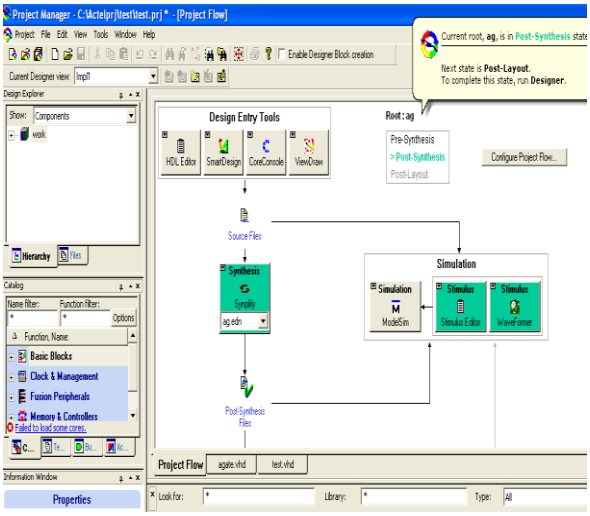
b) Smart Design.

It is a system-level development tool and IP deployment platform that simplifies the task of assembling and connecting IP for implementation in FPGAs. It enables us to select IP components from a database graphically stitch them together to build a processor-based system-level integration (SLI) design.It contains the processor, memory controllers, UART, Timer, Interrupt Controller, Analog Block and interface, Remap, GPIO, PWM, as well as instantiations of the AHB and APB buses.

c) Soft Console.

This provides a flexible and easy-to-use graphical user interface for managing embedded software development projects. One can quickly develop and debug software programs and implement them in FPGAs. Soft Console enables you to configure project settings, edit and debug software programs, and organize your files. With this tool we have simultaneous access to multiple tool windows and the ability to quickly switch editing and debug views.

Actual view of Libero Project Window :
Libero version 9.0



"Figure 2.Actel Libero Window Screen Shot"

Connect to the Target:

Before programming the FPGA, you will connect to the target board and setup HyperTerminal to communicate over the UART in your design. Perform the following steps to setup the Communication.

1. Open the HyperTerminal application (**Start** > **Programs** > **Accessories** > **Communications** >**HyperTerminal)**. Enter the Name field in the Connection Description dialog box and click **OK**.

2. Select the COM port you identified in the Getting Started Section of this tutorial and click **OK**.

3. Enter the following for the properties

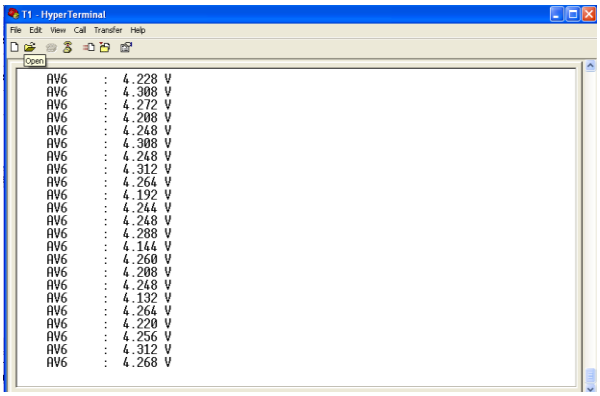
– Bits per second: 57600
– Data bits: 8
– Parity: None
– Stop bits: 1
– Flow control: None

6. Results:

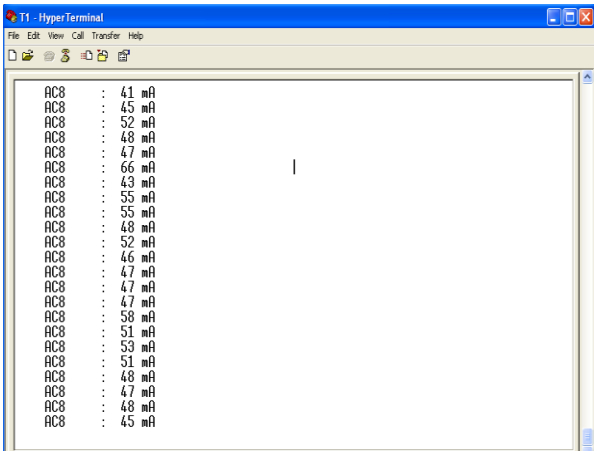
For varying the voltage, the user can vary the potentiometer R31 provided on the board. The user can see the following screen as an example for variation in voltage reading with the rotation of potentiometer.

R31 potentiometer

"Figure 3.Fusion Board photo"



"Figure 4.Voltage readings"



"Figure 5.Core Current Readings"

7. Conclusion.

Flash-based FPGAs, the mixed-signal M1 Fusion PSCs are virtually immune to tampering, assuring users that valuable IP will not be compromised or copied. Having a core inbuilt in a FPGA and being offered as system on chip solution would help to achieve hardware privacy along with soft-core protection and this would definitely offer as an added feature for the industry people.

1) It provides a single chip solution that reduces cost, power and board space and design complexity.

2) With ever increasing customers demand for the new product launch, competitors to achieve this target try to copy the products looking at the hardware and rewrite the code and sell at a lower price. So with this they bring the product in market in lesser time and at a cheaper price. Such malpractices can be stopped if we preserve the product in the form of Intellectual property and protect our product from being copied.

3) Implementing designs on devices with robust security mechanism against theft and piracy is essential. This provides enhanced security protecting the design through its entire product life cycle is important. So soft core implementation in the device is a popular approach followed by industries to protect their code.

8. Future scope.

The embedded processor market is fragmented when it comes to addressing the varying needs of embedded applications. Embedded processor vendors must offer many variations of their processors and supporting chipsets to meet specific peripheral requirement for every application they go after.

A CPU integrated into an FPGA lets embedded designers take full advantage of inherent parallelism of FPGAs to achieve high level of system performance.
Further scope could be use of multi processors in the device can execute code faster. FPGA also allow embedded designers to upgrade the embedded systems performance at any stage of the product life cycle without the need to redesign the board

References:

[1] Peter Yiannacouras, J.Gregory Steffan-"Exploration and Customization of FPGA-Based Soft Processors" in IEEE Transactions on Computer aided design of integrated circuits and systems-vol26, No.2, February 2007.
[2] P. Yiannacouras, J. G. Steffan, and J. Rose, "Application-specific customization of soft processor microarchitecture," in *Proc. Int. Symp. FPGA*, 2006, pp. 201–210.
[3] I. Kuon and J. Rose, "Measuring the gap between FPGAs and ASICs," in 2006 Fourteenth ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, pp. 21–30, USA, New York, NY: ACM Press, 2006.
[4] I. Kuon and J. Rose, "Measuring the gap between FPGAs and ASICs," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 26, no. 2, pp. 203–215, 2007.
[5] Jason Tong, Ian D.L.Anderson and Mohammed A.S.Khalid, "Soft core Processors for Embedded Systems" in Proc.of IEEE, Microelectronics, 2006
[6] J.Turley,"Survey:Who uses custom chips," Embedded System Programming, ol.18,no.8,Aug.2005
[7]Zoran Salcic- "Microprocessors and Microsystems", Science Direct -Volume 21,Issue 4,30 Dec 1997,Pages 249-256
[8] K. Parnell and R. Bryner. Comparing and Contrasting FPGA and Microprocessor System Design and Development. Xilinx, July 2004.
[9] Xilinx, Inc. MicroBlaze Soft Processor, August 2004.
[10]Actel Corp., "The Advantages of the 32-Bit Cortex-M1 Processor in Actel FPGA.
[11]www.xilinx.com/support/documentation/white_papers/wp213.pdf
[12]www.actel.com/documents/embedded_apps_M1_WP.pdf
[13]www.altera.com/literature/wp/wp-aab090805
[14]www.latticesemi.com/documents/doc30160x50.pdf