

Using field-programmable analog to build adaptable RFID readers

Field-programmable analog array design platforms can be used to create a high-performance RFID design that is highly configurable and adaptive under digital control.

By Tegid Roberts

Typical radio frequency identification (RFID) reader design challenges include multistandard adoption, installation issues, high tag failure rate, and an increasingly crowded frequency space. Using software-controllable silicon allows designers to standardize hardware design across a product range, provide adaptable in-field solutions, and vastly reduce time to market.

Although, RFID technology has been in use for more than a decade, miniaturization of tags and lower-cost manufacturing techniques have fueled an explosion of uses and markets. Trials are under way in Japan and Europe in which school children and hospital patients are being tagged with RFID chips for security reasons. Transport for London (TfL) is set to spend up to £34 million on new digital enforcement technologies that

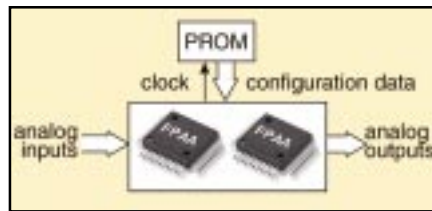


Figure 1. Statically programmed FPAAs.

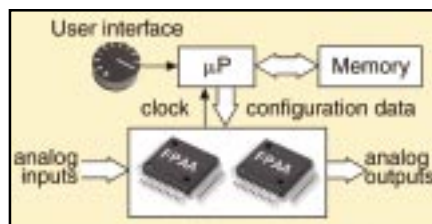


Figure 2. Reconfiguring an FPAAs.

Real-time programmable devices would allow designers to have microprocessor control over analog functionality, thus enabling improved tag detection algorithms.

could include RFID tags for the Congestion Charging Scheme (CCS).

NEC has signed a contract with the Bank of Nagoya in Japan for an RFID-based document management system to launch in April 2005, representing a new application for RFID technology. And also in 2005, European visas and passports will be required to carry an RFID-type chip containing the digitized facial image of the rightful holder. When passing through customs, the document will be scanned by a reader that will compare the image on the chip to the photograph on the document.

Given the increase in RFID usage, many new challenges face design engineers who have been restricted by discrete solutions in a search for the analog functionality of an RFID reader. Currently, these challenges include:

- multiple tag standards;
- 20% tag failure rate;

■ installation and placement issues and the need for cost-effective management and maintenance of readers;

■ reader size reduction that allows them to be imbedded into structures and handheld devices; and

■ intellectual property protection and secure access control protocols.

Designers would be better able to tackle these challenges if they could separate the functionality of an RFID product from its hardware, much like field programmable gate arrays (FPGAs) allow designers to reuse hardware for different functions by changing the run-time program data. A programmable analog solution would allow hardware design to remain largely static across product lines, leaving the functionality to be determined by the program loaded onto the product.

Real-time programmable devices

would allow designers to have microprocessor control over analog functionality, thus enabling improved tag detection algorithms. Installation and placement of devices can be better managed by having predefined setup software running during installation and also maintained on the network. Replacing multiple discrete components with a single IC device would result in a smaller footprint, not only allowing for the design of smaller readers, but also saving on the bill-of-materials (BOM). Additionally, intellectual property and security methods could also be better protected by 'encoding' program data.

The programmable RFID device is no pipe dream. In fact, field-programmable analog arrays (FPAAs) address all of these design challenges.

Programmable analog approach

Within an FPAAs, a wholly analog CMOS-based switched-capacitor (SC) network is combined with a programming layer that allows designers to implement an extremely wide variety of signal processing functions, applied using digital configuration data. SC techniques are well established as a means of implementing integrated CMOS analog circuits, where circuit function is dependent only on the matching of components, thereby delivering precision between 0.1% and 1%, significantly better than discrete analog



Figure 3. FPAAs design tool and CAM parameter setting screens.

components. SC also offers excellent linearity that typically exceeds 94 dB or 0.002% in a filter block, and typical bandwidths range from dc to 2 MHz. This allows designers to use the FPAA for low-frequency RFID solutions as well as with the subbands of higher frequency standards.

The nature of the FPAA is that it is field programmable. Moreover, it is dynamically and partially reconfigurable. This means that by applying the appropriate configuration data, the FPAA can perform:

- a single function when powered up;
- a sequence of different functions over time; and
- a dynamic function that can partially change over time, or can have its attributes adjusted freely without interrupting operation.

The next section outlines the process of designing an FPAA application. The result of this process is typically either a set of configuration data, a collection of data sets, or a set of software instructions.

In the simplest case, a design may be fixed having completed a prototype, and a single set of data made available to configure the device. The FPAA is volatile, and like an FPGA it is programmed when powered up. The simplest way to do this is to couple it to an electrically erasable programmable read-only memory (EEPROM) chip containing the data. When power is applied, the FPAA downloads configuration data from the EEPROM (Figure 1).

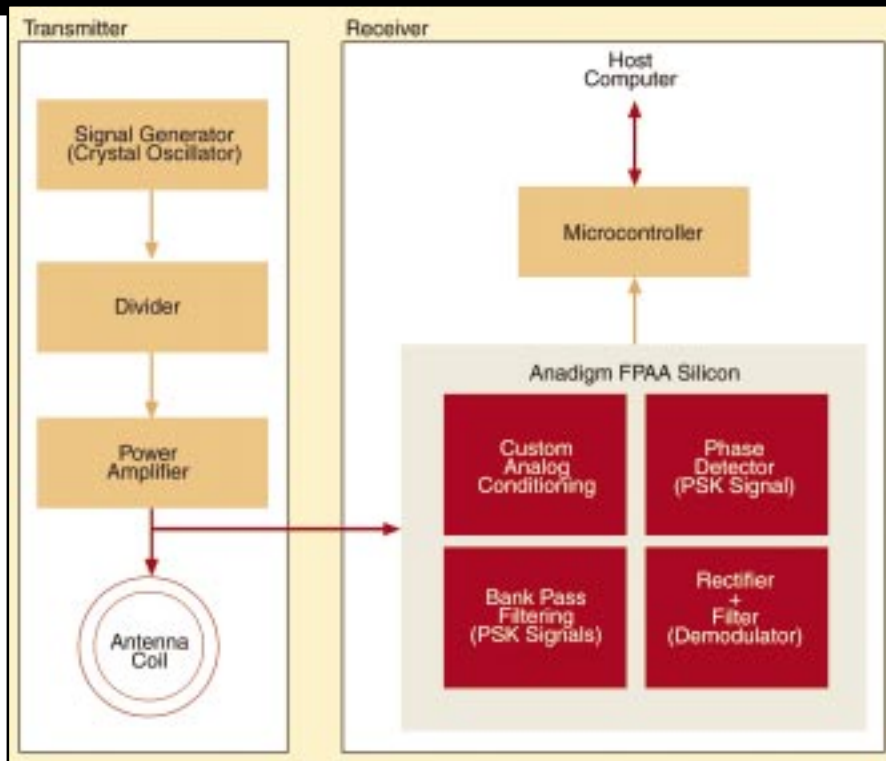


Figure 4. Block diagram of an RFID reader incorporating an FPAA.

More flexible operation is afforded with the use of some form of logic controller, which can be selective of the data to be loaded into the device. For example, a microcontroller that has been programmed to perform reconfiguration of part of the

FPAA can apply configuration data to an RFID device selectively based on controls or system conditions (Figure 2).

The user interface might be anything from a simple installation setup program to a sophisticated RFID reader network software management tool. For both, the circuit design and reconfiguration control development process are the same, and are straightforward. These are described in the next section.

Designing an FPAA circuit

In general, when designing an FPAA circuit, the designer does not need to see how the circuit is realized. This is handled by the design tools in which a circuit is constructed by the designer using circuit building blocks called configurable analog modules (CAMs).

Examples of CAMs are filter stages, gain stages, summing/difference stages, voltage multiplication, phase/voltage comparators, rectifiers, oscillators, peak detectors, and references. In all, some 50 to 60 such building blocks are available, and Anadigm is constantly adding to these libraries of modules. The FPAA also supports non-linear processing functions including modulators, arbitrary transfer functions, and programmable soft-knee compression.

The CAMs, which are accessed using a robust design tool, are contained within a design library and can be dragged and dropped onto a schematic 'canvas' within the design tool—the FPAA device that will realize the final design. The designer sets various attributes for that CAM, such as filter corner frequency, via a parameter dialog box, and

the system algorithmically calculates the 'best fit' component values and circuit structure to meet requirements (e.g., see CAM subcircuit in Figure 3). Thus, the designer never needs to be concerned with low-level circuit design because the FPAA software assembles the configuration data set(s) for the entire design and delivers it in a form ready for delivery from EEPROM, or host microprocessor.

Using an FPAA in an RFID reader

In Figure 4, the FPAA replaces application-specific, discrete circuitry on the receiver end of the device that is tailored to one specific set of functions. This device is controlled by the host microprocessor primarily used to process received data from the FPAA. This is only one example of many possible configurations in which the FPAA can be used, and shows example functional blocks relevant to the RFID engineer, including filtering, rectification and signal conditioning.

RFID designers have found uses for FPAA in applications including those in the receiving arm of inductively coupled systems. For example, the following functions can easily be incorporated and tested using the associated electronic design automation (EDA) tool:

- bandpass filter, fully programmable and either steep-edged or notch-centered around 500 kHz;
- dynamic amplifier section;
- demodulator/rectifier; and
- low-pass filter and gain staging.

And, in the transmitter arm for inductively coupled systems, a <134 kHz oscillator system can be subjected to high demands regarding phase stability and sideband noise, avoiding a worsening of noise ratio in relation to extremely weak signals picked up by that arm. Other potential applications for the FPAA include demodulating FSK or PSK datastreams directly, and closed loop applications such as adaptive systems useful for active filtering.

FPAA benefits for RFID

Design engineers using FPAA technology can bring many much-needed improvements, such as a streamlined design process, improvements in manufacture, and quality control. Some of these include:

- The ability to produce one hardware design to feed a whole product line, made possible by a change of software.
- Dynamic optimization that allows for increased range and performance in electrically noisy environments, allowing improved RFID tag readability.
- Increased ability of engineers to customize solutions that allow for more advanced

features, such as anti-collision techniques, to detect multiple tags in close proximity, which will be particularly important in the retail marketplace where the desire to read many tags per second is dramatically increasing.

■ Easy support of changing and/or multiple protocols because the current RFID standards are constantly being reviewed and added to the FPAA methodology.

■ Excellent stability in the FPAA over degradation through temperature and time

fluctuations improves reliability in the field, which will be particularly important for readers placed in hostile environments.

■ A layer of design protection not currently available in discrete design that is a result of the programmable nature of the device at boot up, when it downloads 'encoded' program data information that describes the process algorithms. In particular, the access control market requires a very high level of secrecy to maintain product

integrity as these devices are used to protect property and are often used as electronic locks.

■ Allowing the consolidation of components and reduction of board space, leading to smaller end products and an easy retrofit into existing solutions.

In addition, cost savings can be expected by implementing the FPAA into RFID reader product lines. These result from board design consolidation and standardization across a range of products and easier inventory

control. Similarly, board footprints can be reduced by replacing discrete components with a single device. Due to the extremely high tolerance of the device, this can also improve quality control, particularly by replacing the need for expensive component matching and tolerance testing at the manufacture stage. Other software benefits are precise control over last-minute changes to the algorithm design, easily employed by reprogramming devices without the need for hardware changes.

During the design phase, the real-time nature of the software design tool greatly reduces design time from months to days, and in some cases, even minutes. An engineer can rapidly prototype designs in the software domain, and using the built-in simulation tools, can even test a design topology without having to implement hardware changes. By exploiting the real-time programmable nature of the FPAA device, the installation of RFID readers can be greatly simplified for the installation or maintenance engineer. It is possible to move away from screwdriver calibration methods and have either a software calibration system or even a network-monitored self-calibrating system. This approach can greatly reduce setup and maintenance costs at installation and in the future.

Summary

Discrete analog circuitry does not easily allow for digital programmability and dynamic control. Digital signal processing (DSP) can fill this need but cannot provide cost-effective solutions in the RF domain. FPAA not only provides a cost-effective and efficient platform for creating high-performance RFID solutions, it also allows for designs that are highly configurable and adaptive under digital control.

Thus, benefits typically associated with DSP—such as the ability to differentiate, “future-proof” and tune products by exploiting software-based configurations, and to maintain security of design by use of an integrated solution—are available while maintaining an analog signal path throughout.

The software design tools facilitate development of both the circuit and control interface to a host microprocessor or can be controlled from an EEPROM.

The uses of programmable analog in RFID applications are only limited by the imagination of the design engineer, while the EDA tool greatly encourages the ability to try novel and exciting ideas that can be translated to manufacture, installation and long-term maintenance. **RFID**

ABOUT THE AUTHOR

Tegid Roberts was with Sony Corp. for seven years working on in-house projects. He left Sony in 2002 to start his own consultancy practice. Tegid is a graduate of York University with a degree in electronic engineering.

