A CMOS Continuous-Time Field Programmable Analog Array

C.A. Looby and C. Lyden

National Microelectronics Research Centre,
University College, Cork, Ireland
E-mail: *clooby@nmrc.ucc.ie, lyden@nmrc.ucc.ie*

Abstract

A circuit design for a Field Programmable Analog Array is presented which improves accuracy and repeatability compared to previous designs. Controlled by a configuration register, continuous-time signals are routed among programmable analog blocks to implement the user's chosen circuit. The configurable connections are realised by CMOS switches and a new innovation is that these are either buffered or nulled to cancel parasitic error. The function blocks are Op-Amps combined with passive networks which allow programmable transfer functions with accuracy insensitive to variations in process parameters and environment. The intended application area is the rapid development of analog circuits which are presently prototyped by PCBs stuffed with Op-Amp and passive components. The concept has been demonstrated on a CMOS IC and the resulting performance shows the feasibility of this approach to general purpose FPAA technology.

1 Introduction

The trend in modern hardware development of increasing integration and decreasing design turnaround time leads to the widespread adaption of field programmable devices[1]. While the bulk of electronic circuitry now consists of digital logic, there will also commonly be a portion which interfaces to the analog sensors and actuators that the real world necessitates.

The development of the analog portion requires field programmable devices to avoid large Non-Recurring Engineering(NRE) costs and to reduce turnaround time[2]. The digital circuitry has used FPGAs to improve flexibility and reduce project duration. The field programmable analog devices would need to offer ease-of-use and flexibility benefits similar to those of FPGAs.

There are fundamental difficulties in realising a versatile and useful Field Programmable Analog Array(FPAA) over and above those encountered with FPGAs. Apart from the usual challenges inherent in analog IC design (such as meeting linearity, noise and bandwidth specifications subject to process and environmental variations), the FPAA has two additional obstacles:

- The configurable routing should not affect the accuracy of the instantiated circuit
- The analog blocks must provide a variety of useful programmable functions

To date, general purpose FPAAs have tackled these problems using transconductor[3, 4] or switched-capacitor techniques[5, 6].

This paper describes a routing method that achieves wide bandwidth due to continuous-time operation. It offers accurate signal transfer without calibration since the routing is nulled and the function accuracy is limited only by device matching.

Section 2 details the routing elements used to configure the FPAA. The Function Blocks used in the FPAA are described in Section 3 including how their advantages in terms of accuracy and programmability is achieved. In Section 4, we discuss the overall architecture of a demonstrator FPAA using the elements already discussed. The results for the Function Blocks are presented in Section 5.

2 Routing elements

In general, the FPAA gets its configurability from the routing network which allows the user to control how signals flow through the available analog resources.

Firstly, CMOS switches are chosen for the routing mechanism because they are easy to configure digitally and have a simple dominant parasitic, the ON-resistance R_{on}[7]. This swamps any parasitics of any OFF switches attached to the same node at the sub-1MHz signal frequencies we address.

The problem of accurate FPAA operation is to eliminate error due to current flow in the R_{on} of the switches routing the signal between the diverse source and load impedances of active Function Blocks.

As an example, consider when the pass-switch approach is used for routing among the Function Blocks, as shown in Figure 1. The switch connects a source of finite impedance to a load of finite

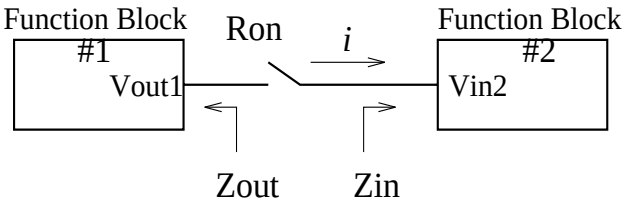

Figure 1: Illustration of the switch R_{on} problem

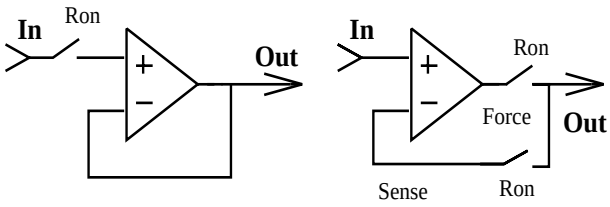

Figure 2: Buffer & Force-Sense nodes

impedance. These impedances will differ depending on the configuration. The signal transfer to Function Block 2 is given by:

$$V_{in2} = V_{out1} \frac{Z_{in}}{Z_{out} + R_{on} + Z_{in}}$$

The switch ON-resistance(R_{on}) is dependent on voltage, temperature and process variables. Therefore the transfer function of the node corresponds to a voltage and configuration dependent loss.

Prior solutions adopt transconductor[3] and switched-capacitor[5, 6] techniques to achieve a routing node with a well defined transfer function.

The switched capacitor approach transmits signals as charge which is not subject to loss due to R_{on}. However since it operates on sampled signals, the need for anti-aliasing filters constrains the bandwidth. The accuracy of the transfer function depends on the ratio of capacitors. Therefore this design method takes advantage of the matching devices available in IC technology.

The transconductor approach[4, 8] is a continuous time one and the transconductor is part of the transfer function. As such, there is no loss due to R_{on} but the voltage range is limited by transconductor linearity while the overall transfer function depends on active device transconductance and passive device impedance, the product of which is unlikely to be controlled in a given process or environment.

We have combined the benefits of both switched-capacitor and transconductor methods which are:

- The function accuracy depends on device ratios.
- Signals are continuous time.
- The routing has its R_{on} nulled.

Continuous-time voltage signals are routed using CMOS pass-switches into high-impedance internal nodes. These nodes are the inputs of unity-gain amplifiers. An Op-Amp with unity closed loop gain(Figure 2) will maximise linear voltage range and input impedance while its low output impedance means it is insensitive to loading. It eliminates current flow in the pass-switches, the main cause of inaccuracy in the internal nodes.

For external signals, it is desirable that the buffer also be output-switched so that each buffer can be fully switched to every route available. Otherwise redundant buffers would cause inefficient FPAA utilisation. By putting the output switches inside the feedback loop as in Figure 2, output switching is provided.

Even though this reintroduces current flow in the R_{on} of the Force switch, the redundant Sense switch means that the error is not seen at the output. It is now possible to configure pins to be inputs or buffered outputs. This Force-Sense approach is suitable for low impedance loads down to $5k\Omega$ because the Op-Amp gain ensures negligible source resistance.

Minimum size switches with their large R_{on} can be employed when buffered and the benefits are not only in layout area but parasitic capacitance and crosstalk. The node buffers are designed to have excess bandwidth compared to any other elements in the signal flow to ensure that they do not limit performance.

3 Programmed Function Blocks

In standard IC technology, it is difficult to build-in precision analog circuitry because component parameters are subject to process variations. While design methodologies such as switched-capacitor and master-slave-control circuits have circumvented this problem, they are suited to signal processing rather than a general purpose FPAA.

To achieve accurate transfer functions with standard IC processes, it is necessary to use ratios of passive device values, rather than the absolute values alone, as the controlling parameter. The suite of Function Blocks proposed here all use Op-Amps under passive device feedback to achieve this. Programmable parameters can be obtained by appropriate use of passive and switch networks.

An example of such a function is signal addition. In Figure 3, one signal is given a fixed gain and combined in a programmable ratio with another signal. It uses an R-2R CMOS DAC with 5 bits resolution designed to achieve 8 bit linearity.

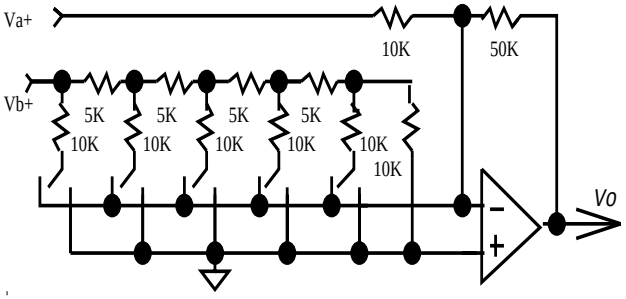

Figure 3: Programmable Adder and Gain Block

All the usual analog layout precautions were rigorously applied to this block design. The resistors of the R-2R DAC must match to eight bits. To achieve this, Gate Poly is used as it has very low voltage and temperature coefficients compared to other integrated resistor materials. The Poly sticks were deliberately designed as non-minimum width to ensure repeatability at the expense of extra area. Dummy resistors were also used to combat etching effects so

that every active resistor has identical neighbourhood. To ensure uniformity the DAC switches are replicated rather than geometrically scaled. The Op-Amp input devices are arranged as a common centroid to minimise random input offset.

The second function is subtraction which can be configured with integration, comparison or fixed gain(Figure 4). The DAC used this time is a resistor string type, because this is more suitable for voltage-mode operation than the R-2R ladder. The ratio of

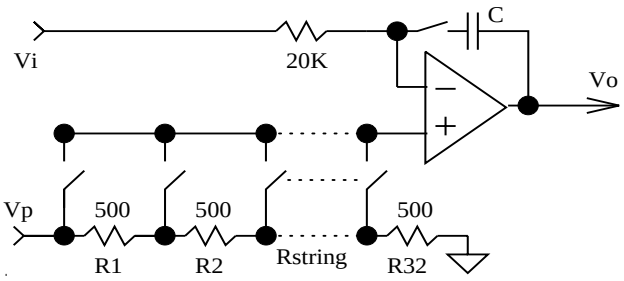

Figure 4: Programmable Subtractor Block

subtraction as well as the integration time constant is programmed by these two DACs. The Subtractor Function Block can act as a window comparator function simply by opening the switch in the feedback path. As with the Programmable Adder Block, thorough attention to detail was applied at the layout phase of the Subtractor Block to ensure the resistor matching.

The Subtractor uses two different elements, resistors and capacitors. Because only the resistors match to each other, the time constant is unavoidably dependent on the process capacitance. However this effect can be compensated using the programmable gain. The critical matching between the two channels is assured however.

4 Demonstrator FPAA IC

The Configuration Routing Elements and Function Blocks discussed above have been combined in a Demonstrator FPAA IC. The FPAA has an array of four Function Blocks, structured(Figure 5) as a $2x2$ arrangement of the Adder and Subtractor modules. A crossbar network using multiplexer switch boxes allows all inputs and outputs of the Function Block array to routed. The multiplexers consist of 8 switches and connect to buffered nodes. The external analog interface is comprised of six pins which connect to Force-Sense nodes in the crossbar network.

The Demonstrator FPAA IC(Figure 6) is fabricated in a 2.4μm N-WELL double-Poly CMOS process provided by Europractice[9] and occupies an area of $36mm^2$ in total. Power dissipation is $15mW$ and the design is core limited, having 29 I/O pins.

The configuration data is transferred serially as 118 bits and the configuration register is double buffered. This allows in-service reconfiguration. Safe power-up is ensured by the reset. Devices can be daisy-chained and partially reconfigured. The configuration stream has a read-back facility. The switch controls are differential to minimise crosstalk and common-mode effects.

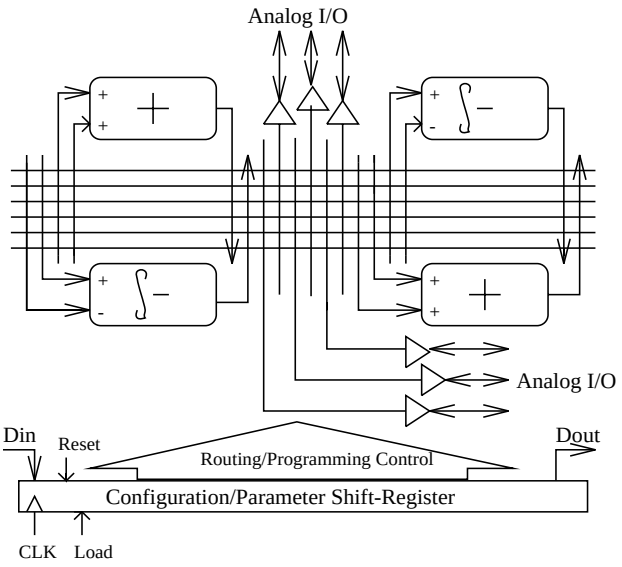

Figure 5: Structure of Demonstrator FPAA

The switch boxes and Function Blocks were individually designed as described in the previous sections. The overall layout of the IC was automatically Placed and Routed(PAR) with some manual intervention. This manual intervention consisted of guiding the placement of critical analog blocks and constraining the width, spacing and length of the analog signals, which was possible using Mentor ICgraph version A.4. The automatic PAR completed the routing of the configuration bits which was considered non-critical and too laborious for manual methods.

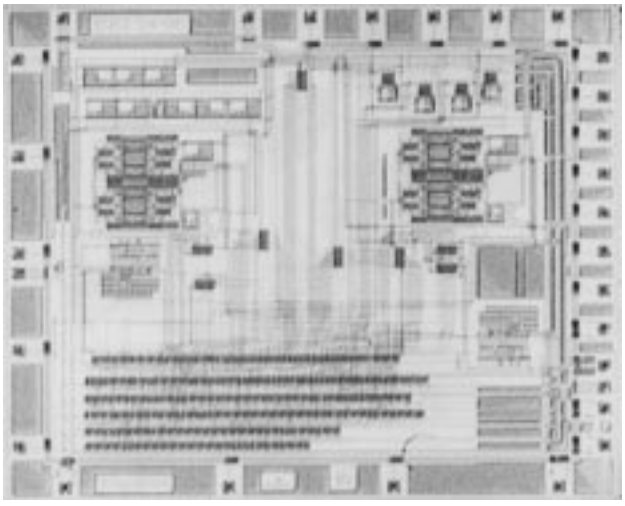

Figure 6: Microphoto of Demonstrator FPAA

A large part of the layout area is occupied by interconnect area and there is potential for compressing the layout significantly so that more Function Blocks could be accommodated.

5 Results

The Signal Adder circuit achieves 8 bit linearity as shown by Figures 7 and 8. Resolution is 5 bits and the measurements are within one-eighth of an LSB. This equates to the same monotonicity as

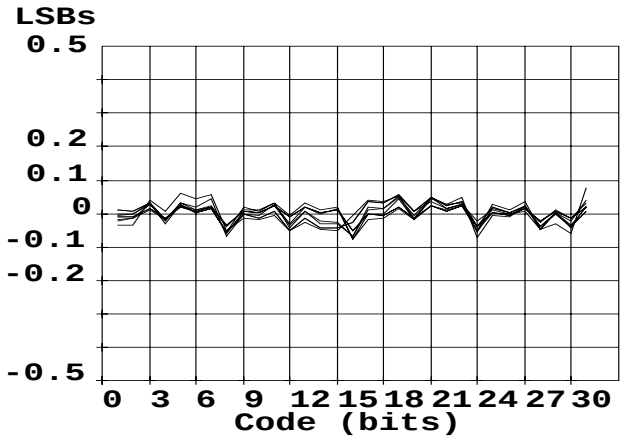

Figure 7: Adder: Integral Non-Linearity

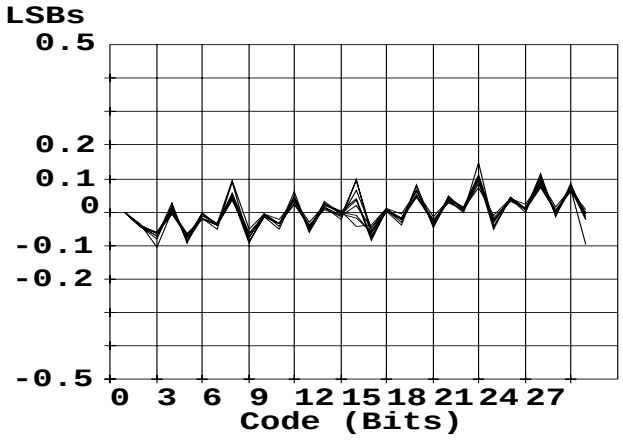

Figure 8: Adder: Differential Non-Linearity

an 8-bit accurate DAC would show at the transitions of the fifth bit. Offset is 10mV when adding both channels equally. The design approaches $1MHz$ bandwidth(Figure 9) independent of the amplifying factor. Compensation matches the gain setting for constant bandwidth.

However the Op-Amp in the case of the Subtractor only has a capacitive load and the dynamics are improved by having a lower drive output, relative to the Adder circuit. The common-mode gain response with various time-constant(RC) settings is shown in Figure 10. Figure 11 depicts the relative gain response of either channel with different gain ratios relative to the other.

The Adder and Subtractor Function Blocks are examples of suitable elements for the function array of an FPAA. The diversity and versatility of further Function Blocks has to be balanced with the unused layout area they occupy if the user doesn't achieve full utilisation.

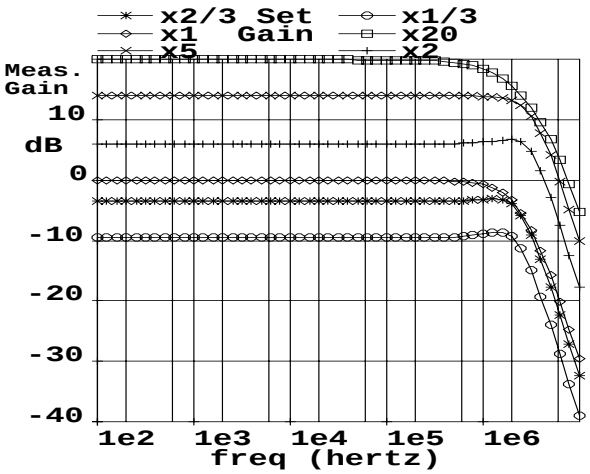

Figure 9: Adder: Gain Response

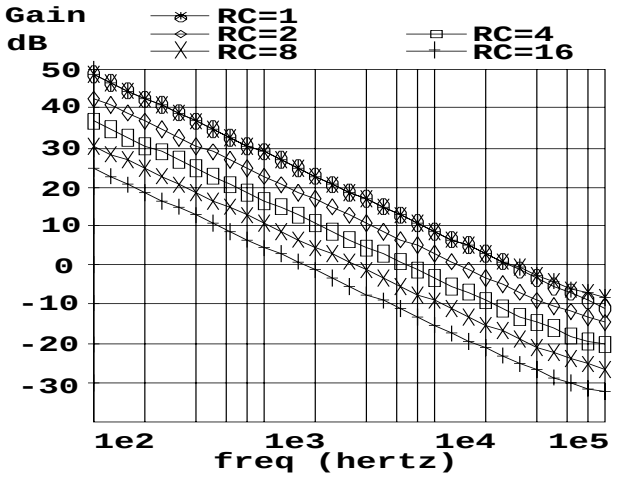

Figure 10: Subtractor: Common Response

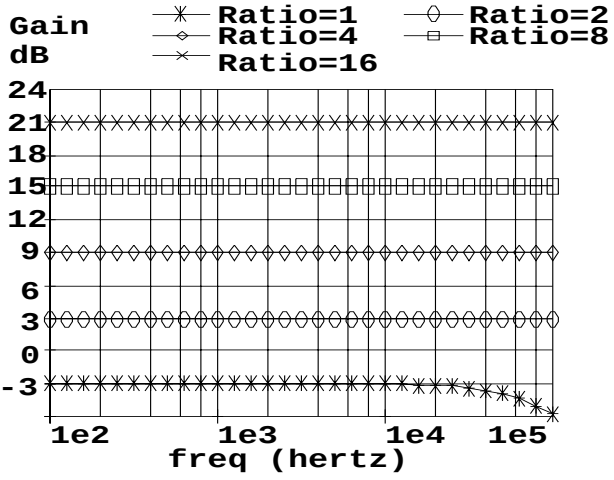

Figure 11: Subtractor: Relative Response

The two examples above, when arranged in a crossbar network of switch-buffers, are capable of implementing a large number of the glue circuits which combine signals linearly. The switch-buffer allows the programmable connection of Function Blocks. Accuracy can be maintained to an 8-bit level, sufficient for most target applications.

The results demonstrate that our architecture produces satisfactory performance in spite of the use of a low-cost, low complexity technology. This work has established a lower bound for the expected performance of FPAAs using our architecture and does not require any advanced fabrication steps.

However, by taking advantage of the lower parasitics, greater device complexity and higher density offered by modern, fine-geometry processes, this type of FPAA could offer superior performance and sufficient resources for a wide variety of analog circuit development. For example, a submicron BiCMOS implementation could be expected to offer increased bandwidth up to $20MHz$ and 50 Function Blocks within similar die area.

6 Conclusion

The essential functional elements of an FPAA have been described. After identifying accurate configuration and precision transfer functions as the major barriers to implementing the FPAA, the extent to which previously published FPAAs cope with these design challenges have been summarised.

The Op-Amp and programmable passive network have been proposed as the solution to the programmable parameter circuit for a general purpose FPAA. The buffered switch has been examined as a way of eliminating the configuration error due to ON-resistance R_{on}.

By demonstrating the complete concept on a small scale with an inexpensive technology, this FPAA when migrated to a small-geometry process is shown to have real potential. Measurement results show the feasibility of this approach to realising a general purpose Field Programmable Analog Array.

Acknowledgements

The authors would like to thank Europractice for IC fabrication.We are indebted to our colleagues John Doyle, Aidan Keady, Kevin Rowley and Sverre Lidholm for various reviews.

References

[1] S. van Tyle, “Market Facts”, Electronic Design, pp64, May 28, 1992,

[2] M. Sivilotti, “A Dynamically Configurable Architecture For Prototyping Analog Circuits ”, *Proc. Decennial Caltech Conf.*, Cambridge, MA, pp237–258, 1988.

[3] E.K.F. Lee and P.G. Gulak, “A Field-Programmable Analog Array based on MOSFET Transconductors”, *Electronics Letters* , Vol. 28, No. 1, Jan 2, 1992, pp28–29.

[4] E.K.F. Lee and P.G. Gulak, “A Transconductor-Based Field-Programmable Analog Array”, *1995 IEEE ISSCC Digest of Technical Papers*, pp. 198–199, Feb. 1995.

[5] H.W. Klein, “Circuit development using EPAC technology: an analog FPGA”, *Proceedings of the SPIE - The International Society for Optical Engineering*, vol.2607, pp. 136–44, 1995.

[6] A. Bratt and I. Macbeth, “Design and implementation of a Field-Programmable Analogue Array”, *Proceedings of the 1996 ACM SIGDA* 4^{th} *International Symposium on Field-Programmable Gate Arrays*, pp88–93, Feb. 11–13, 1996.

[7] S. Chang, B. Hayes-Gill and C. Paull, “Implementation of a Multi-function Signal Detection Block for a Field Programmable Analogue Array using Mietec 2.4μm CMOS Process and Mentor Graphics Software Version 8.2”, *1994* 4^{th} *Eurochip Workshop on VLSI design training*, pp. 226–231, Oct 17–19, 1994.

[8] P. Chow, P. Chow and P.G. Gulak, “A Field-Programmable Mixed-Analog-Digital Array”, *Proceedings of the 1995 ACM SIGDA* 3^{rd} *International Symposium on Field-Programmable Gate Arrays*, pp104–109, Feb. 1995.

[9] Mietec-Alcatel, $2.4\mu m$ CMOS Standard Cells Design manual, Rel. S2SM 2.0.0, 1989.