# A CORDIC-based dynamically reconfigurable FPGA architecture for signal processing algorithms

Hongzhi Wang, Pierre Leray, and Jacques Palicot
SUPELEC/IETR
Campus de Rennes, Avenue de la Boulaie, CS 47601 F-35576 Cesson-Sévigné, France
Email: {hongzhi.wang, pierre.leray, jacques.palicot}@supelec.fr

## Abstract

CORDIC algorithm is very widely used in the digital signal processing systems. This article presents a CORDIC-based reconfigurable FPGA architecture that can be changed to perform different algorithms of signal processing. Two examples of CORDIC-based signal processing, such as FFT and SVD algorithms are surveyed in this paper. The fact that the common operator CORDIC is exploited in these algorithms makes them easily to be implemented on the same reconfigurable hardware. Specifically, the dynamic reconfiguration is performed for changing the interconnections between CORDIC. The dynamic reconfiguration method is used to improve both the reconfiguration time and the area efficiency.

## 1 Introduction

The COordinate Rotation DIgital Computer (CORDIC) was formally introduced by Volder [1]. It provides an effective method for computing iteratively the rotation of a two-dimensional vector using only add and shift operation. The CORDIC-based hardware are widely used in the digital processing systems such as digital filtering, modulation, fast Fourier transformations (FFT) and Singular Value Decomposition (SVD). In [2], it shows also how to use CORDIC to implement direct digital synthesizers, AM, PM, and FM analog modulators and ASK, PSK and FSK modulators,etc. It was shown that CORDIC-based architectures are a very appealing alternative to conventional multiply-and-add hardware. We have shown in our previous work [3] a reconfigurable MIMO V-BLAST decoder based on various number of CORDIC to adapt to different requirement of wireless systems.

Reconfigurable computing has been an active area of research over the past decade. [4] and [5] has been proposed in a large range of signal processing applications in order to improve high performance, flexibility and adaptability. The rapid growth of wireless communications has led to the demands for communication devices which can support multiple standards and have the capability of switching one to another. A key challenge of construction of a communication device is design of flexible device that can dynamically configure itself to run different algorithms as required to support the different standards, as in Software Defined Radio (SDR) systems.

There has been recently an increasing interest in the use of reconfigurable hardware chip, such as Field Programmable Gate Arrays (FPGAs), for computing signal processing because of their reconfigurability and support of parallelism. Moreover dynamically reconfigurable FPGAs offer new design space with a variety of benefits: flexibility and reusability at run time. The dynamic reconfiguration is closely related to partial reconfigurability of FPGA. Indeed, the partial reconfigurability allows to selectively change segments of the FPGA functionality without suspending operations of the remaining parts. There are several benefits of partial reconfiguration (PR). It reduces the configuration time and saves memory as the partial reconfiguration files (bitstreams) are smaller than full ones. We have introduced in [6] the dynamic reconfiguration into the MIMO V-BLAST decoder architecture. We especially detailed our design experiments to integrate the control of configuration into the MIMO decoding algorithm processing and an implementation of MIMO decoder in the Virtex-4 from Xilinx FPGA which improves considerbly all the performances: the reconfiguration time, the area, the size of bitstreams compared to Virtex-II previous implementation. In this paper we extend our previous work by presenting how to use this dynamic reconfigurable architecture to other signal processing algorithms such as FFT and SVD.

The rest of the paper is organized as follows. The CORDIC algorithm is briefly described in section 2. FFT and SVD algorithms are overviewed in section 3. Section 4 shows the CORDIC-based dynamic reconfigurable architecture and its configuration management (with the MicroBlaze implementation). The conclusions will be stated in section 5.

## 2 CORDIC algorithm

A more recent survey of CORDIC algorithm on FPGAs is presented in [7]. The CORDIC processor performs the vector rotation to compute a set of trigonometric functions. The principle idea of the CORDIC algorithm is to decompose a rotation into a sequence of micro-rotations over the angles $\theta_i$, where $\theta_i$ is restricted to $tan^{-1}(2^{-i})$ and $K_i = \cos(tan^{-1}(2^{-i}))$. The corresponding micro-rotation are expressed by the following iterative equation:

$$x_{i+1} = K_i \left[ \; x_i - y_i \cdot d_i \cdot 2^{-i} \; \right] \tag{1}$$

$$y_{i+1} = K_i \left[ \; y_i - x_i \cdot d_i \cdot 2^{-i} \; \right] \tag{2}$$

$$\theta_{i+1} = \left[ \; \theta_i - d_i \cdot tan^{-1}(2^{-i}) \; \right] \tag{3}$$

The CORDIC method can be employed in two different modes, known as the rotation mode and the vector mode. In the rotation mode, the co-ordinate components of a vector and an angle of rotation are given and the co-ordinate components of the original vector, after rotation through a given angle, are computed. In the vector mode, the co-ordinate components of a vector are given and the magnitude and angular argument of the original vector are computed.

## 3 CORDIC algorithm in signal processing applications

This section shows how CORDIC algorithm is used to perform two applications required in signal processing. This includes FFT and SVD algorithms.

### 3.1 FFT

The CORDIC-based multiplier can be employed to perform all the twiddle factor multiplications in FFT. The angle of the twiddle factor is decomposed into the weighted sum of a set of predefined elementary rotation angles. An implementation of the CORDIC based 8-FFT is described in [8], in which the butterfly structure of FFT remains. Its structure is illustrated in figure 1, which contains two type of CORDIC operators (type I and type II). The $\sqrt{2}^{log_2(N)}$ scaling of result is required compared with the MAC-based FFT, but it can be performed after the compuation of three stages.

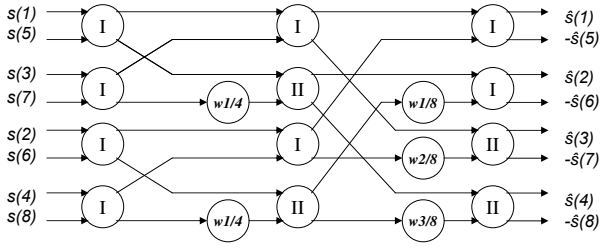


Figure 1: CORDIC based FFT

### 3.2 SVD

The SVD provides a wide range of application in signal processing such as direction estimation, spectrum analysis and systems identification. The Jacobi SVD algorithm is one of the most commonly used numerically stable algorithms for computing the SVD thanks to its high degree of potential parallelism. The SVD algorithm requires the computation of a rotation angle and subsequent operations performed by a squence of Givens transformations. The CORDIC provides an effective way for performing these transformations. In [9], a solution is described to implement an SVD processing element by a single conventional CORDIC system. Figure 2 shows a systolic array architecture for Jacobi SVD algorithm. In this algorithm, an $M \times M$ matrix is divided into $[M/2] \times [M/2]$ processing elements, each containing a $2 \times 2$ sub-matrix, which can be

performed by a two-dimension CORDIC processor. The Diagonal processor $P_{ii}$ and off-diagonal processor $P_{ij}$ perform on the sub-matrix a corresponding rotation computed by a two-dimension CORDIC processor.

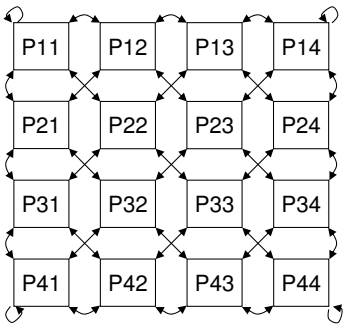


Figure 2: Systolic array architecture for the Jacobi SVD algorithm

## 4 Dynamic reconfigurable architecture and its configuration management

In our previous work [6], we have exploited a CORDIC-based reconfigurable architecture, containing different number of CORDIC to implement a MIMO V-BLAST decoder, as shown in fig.3. It was shown that the reconfiguration of Virtex-4 is indeed more efficient than the Virtex-II, thanks to the new features of Virtex-4. The reconfiguration time of dynamical reconfigurable part is equal to $50\mu$s which is very close to real-time reconfiguration. The purpose of this paper is to extend this architecture by changing dynamically the interconnections of CORDIC to perform different CORDIC-based algorithms such as FFT and SVD. Both FFT and SVD algorithms use the same CORDIC and only the interconnection between CORDIC is changed. Hence the reconfigurable architecture can be used for performing more CORDIC-based algorithms by changing the sequences of reconfiguration of interconnection between CORDIC.

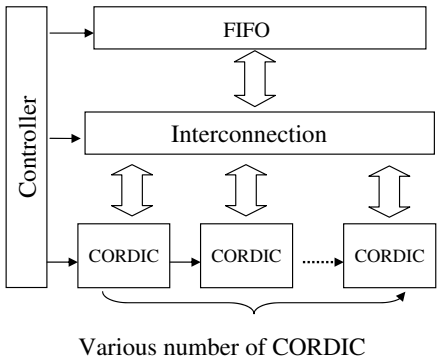


Figure 3: CORDIC-based reconfigurable architecture

The dynamic reconfiguration is used to share a group of configurable elements between several processing contexts. This is a resource multiplexing. Two types of dynamic reconfiguration can be distinguished according to data dependency between both functions in our previous work [6]. When there is data dependency between two configurations, the processing time depends on the reconfiguration time of two contexts of reconfigurable module. The partial reconfiguration allows to reduce the reconfiguration time and saves the storage memory as the size of partial reconfiguration files are smaller than the full ones.

There are several sorts of architectures possible to ensure the dynamic partial reconfiguration of FPGAs. Traditionally, an external configuration controller is used to perform remote updating by using an external configuration interface of the FPGA, the SelectMap port, like illustrated in the figure 4.1. But this approach requires fully reconfiguration of FPGA and results the longest possible reconfiguration period because a full device bitstream should be transmitted over the Host-to-device communication link. Another approach, as shown in the figure 4.2, is using an embedded processor within the user FPGA as configuration controller. In this approach, the Internal configuration Access Port (ICAP) is used to perform partial reconfiguration, avoiding the need of an external device.

In both approaches, the logic resources are separated into two parts, the fixed logic part and the dynamically

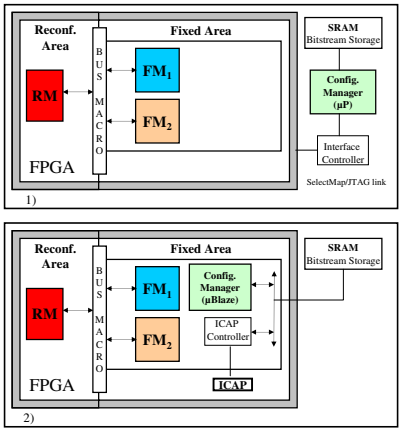


Figure 4: Two configuration managements to perform PR on FPGA.

reconfigurable part. The fixed area contains modules that remains the same during the reconfiguration. The physical separation between reconfigurable module with fixed module or an other reconfigurable one have to be realized by a special communication interface called Bus Macro which is provided by Xilinx to ensure the right place and routing of signal crossing over PR area.

## 5 Conclusion

We have presented in this paper a reliable, resources-saving and flexible means for designing a CORDIC based dynamically reconfigurable architecture. The use of partial reconfiguration shows the configuration time improvement, area efficiency and flexibility. Furthermore, the implementation within the FPGA of the reconfiguration management offers an additional gain in performances. The drawback of using dynamic reconfiguration is that the speed is reduced due to large reconfiguration time. This paper illustrated that the CORDIC could be used to implement different communication functions in signal processing algorithms such as FFT and SVD. So our architecture can be also used to realize these applications simply by changing the sequences of dynamic reconfigurations between CORDIC operators.

## References

[1] J. Volder, "The cordic trigonometric computing technique," *IRE Transactions on Electronic Computers*, vol. EC-8, no. 3, pp. 330–334, 1959.

[2] J. Valls and et al., "The use of cordic in software defined radios: A tutorial," *IEEE Communications Magazine*, vol. 44, no. 9, pp. 46–50, October 2006.

[3] H. Wang, P. Leray, and J. Palicot, "Reconfigurable architecture for mimo systems based on cordic operators," *Elsevier Comptes rendus, Physique*, vol. 7, pp. 735–750, 2006.

[4] K. Compton and S. Hauck, "Reconfigurable computing: a survey of systems and software." *ACM Comput. Surv.*, vol. 34, no. 2, pp. 171–210, 2002.

[5] R. Hartenstein, "A decade of reconfigurable computing: A visionary retrospective," *date*, vol. 00, p. 0642, 2001.

[6] H. Wang, P. Leray, and J. Palicot, "An efficient mimo v-blast decoder based on a dynamically reconfigurable fpga including its reconfiguration management," *ICC2008*, May 2008.

[7] R. Andraka, "A survey of cordic algorithms for fpgas," *FPGA '98. Proceedings of the 1998 ACM/SIGDA sixth international symposium on Field programmable gate arrays*, vol. 34, no. 2, pp. 191–200, Feb. 22-24 1998.

[8] B. Heyne and J. Goetze, "A pure cordic based fft for reconfigurable digital signal processing," *12th European Signal Processing Conference (Eusipco2004)*, vol. 7, September 2004.

[9] Z. Liu, K. Dickson, and J. V. McCanny, "A floating-point cordic based svd processor," in *ASAP*, 2003.