FPGA based Embedded Processing Architecture for the QRD-RLS Algorithm

Deepak Boppana, Kully Dhanoa, Jesse Kempa
Altera Corporation, San Jose CA
Email: dboppana@altera.com, kdhanoa@altera.com, jkempa@altera.com

Abstract

A novel implementation of the QR decomposition based recursive least squares (RLS) algorithm on Altera Stratix FPGAs is presented. CORDIC (Coordinate Rotation by Digital Computer) operators are efficiently time-shared to perform the QR decomposition while consuming minimal resources. Back substitution is then performed on the embedded soft Nios processor by utilizing custom instructions to yield the final weight vectors. Analytical resource estimates along with actual implementation results illustrating the weight calculation delays are also presented.

1. Introduction

Adaptive weight calculation using the QRD-RLS algorithm is a well established technique for signal processing applications such as equalization, digital predistortion and beamforming [1]. These applications generally involve solving for an over specified set of equations, as shown below where m > N, using the least squares approach.

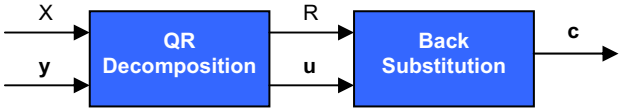
$$x_1(1)c_0 + x_2(1)c_1 + \ldots\ldots x_N(1)c_N = y(1) + e(1)$$
$$x_1(2)c_0 + x_2(2)c_1 + \ldots\ldots x_N(2)c_N = y(2) + e(2)$$
$$\vdots$$
$$x_1(m)c_0 + x_2(m)c_1 + \ldots\ldots x_N(m)c_N = y(m) + e(m)$$

The least squares approach attempts to find the set of coefficients c_n that minimizes the sum of squares of the errors, i.e. $\{\min \sum_m e(m)^2\}$. Representing the above set of equations in the matrix form, we have

$$X\mathbf{c} = \mathbf{y} + \mathbf{e} \qquad (1)$$

Direct computation of the coefficient vector **c** involves matrix inversion, which is generally undesirable for hardware implementation. QR decomposition based least squares algorithm avoids matrix inversions and is robust for hardware implementation. As shown in Figure 1, the algorithm transforms the matrix X into an upper triangular matrix R and the vector **y** into another vector **u** such that R**c**=**u.**

Figure 1. QR decomposition based least squares



The coefficients c_n are then computed using a procedure called back substitution that involves solving the equations shown below.

$$c_N = \frac{u_N}{R_{NN}} \qquad (2)$$

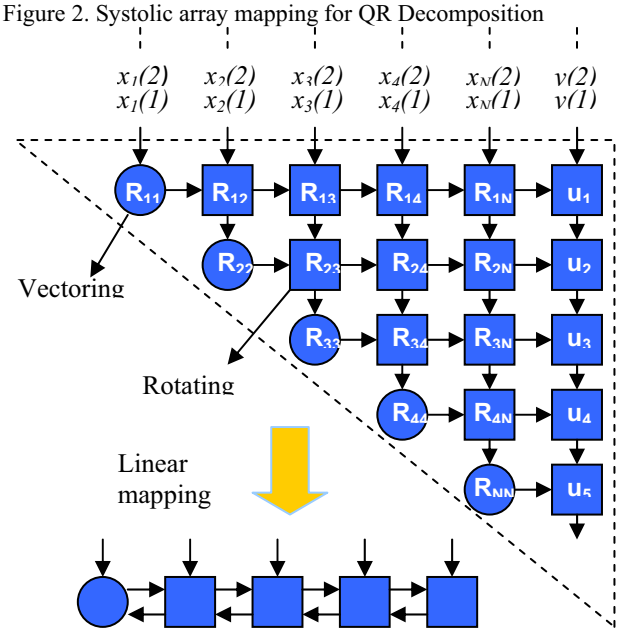
$$c_i = \frac{1}{R_{ii}}\left(u_i - \sum_{j=i+1}^{N} R_{ij}c_j\right) \text{ for } i = N-1, \ldots 1 \qquad (3)$$

The following sections describe the efficient implementation of the algorithm on Altera's Stratix FPGA, while consuming minimal resources and utilizing the concept of embedded soft core processing with custom instructions.

2. CORDIC based QR Decomposition

The QR decomposition of the input matrix X is performed using the systolic array architecture implemented with CORDIC blocks [2]. Altera's CORDIC IP blocks have a deeply pipelined parallel architecture enabling speeds over 250MHz on Stratix FPGAs in both vectoring and rotating modes. Direct mapping of the CORDIC blocks onto the systolic array yields enormous throughput that is generally not required. As shown in Figure 2, the resources required to implement the array can be reduced by trading throughput for resource consumption via mixed and discrete mapping schemes [3].

Figure 2. Systolic array mapping for QR Decomposition



Proceedings of the 12th Annual IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM'04)
0-7695-2230-0/04 $ 20.00 IEEE

IEEE COMPUTER SOCIETY
Authorized licensed use limited to: National Chung Hsing Univ.. Downloaded on October 04,2022 at 09:15:40 UTC from IEEE Xplore. Restrictions apply.

Table 1. Resource estimates for different mapping scemes

Implementation Technique	CORDIC usage		Throughput		Cost LE/update
	No. of Blocks	No of LEs	update delay (us)	updates/s	
Direct Mapping	54	70200	5.1	196078	0.358
Mixed Mapping	4	5200	250.85	3986	1.305
Discrete Mapping	2	2600	198.11	5047	0.515

Table 1 gives an overview of the resource estimates for the three different mapping schemes for an example scenario of m=64 and N=9 with 16-bit inputs to the CORDIC. It is observed that the discrete mapping scheme best exploits the pipelining ability of Altera's CORDIC block and offers the optimum tradeoff between resource consumption and throughput.

3. Back Substitution on Nios Soft Processor

The back substitution procedure, as seen from equations 2-3, primarily involves multiplication and division operations. The Nios embedded processor, with its custom multiply and divide instructions, is an ideal fit to implement this function.

The Nios processor is based on the revolutionary concept of embedding soft-core RISC processors within FPGAs and can operate at over 100 MHz on the Stratix FPGAs. This gives designers the flexibility and portability of high-level software design, while maintaining the performance benefits of parallel hardware operations in FPGAs.

Figure 3 outlines the procedure involved in computing the coefficients via back substitution. The CORDIC block performs the QR decomposition and stores the R and u values (both real and imaginary) in memory accessible to the Nios processor which then calculates the coefficient values using custom instructions and stores the results back into memory.

Figure 3. Nios processor for back substitution

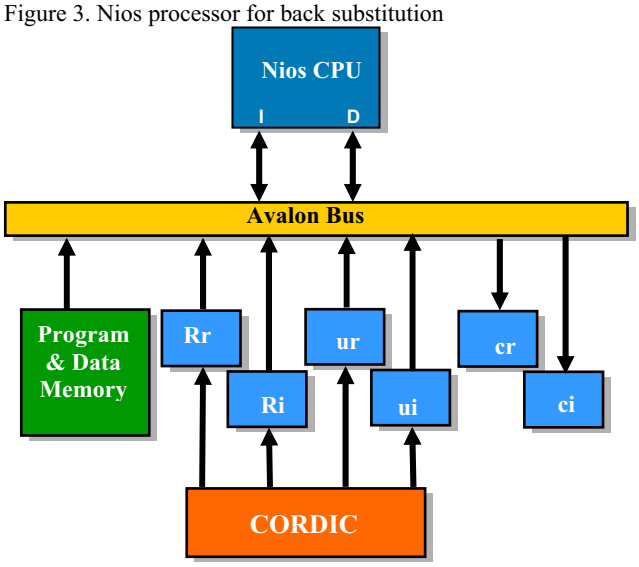
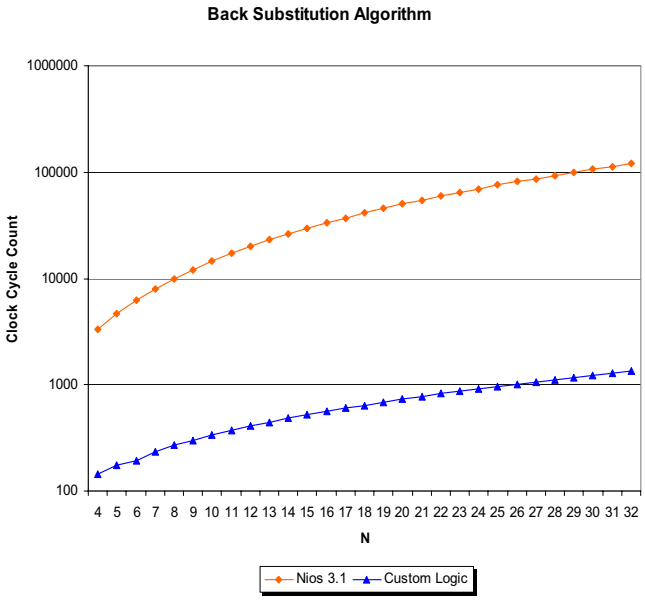


Figure 4 illustrates the number of clock cycles required for the Nios processor operating at 100MHz to calculate different number of complex coefficients via back substitution. The plot also shows the corresponding time taken by the Nios processor when the inner multiply-accumulate loop of equation 3 is accelerated in hardware and accessed as a custom peripheral by the processor.

Figure 4. Latency results for back substitution



It can be observed that for the previously considered example of N=9, the Nios processor takes approximately 12000 clock cycles or 120µs (software only approach) which is acceptable for many applications. For faster updates, the Nios processor with hardware acceleration approach can be used which requires only 300 clock cycles or 3µs, but at the expense of additional hardware resources. Moreover, the Nios processor can be used to implement other data and control functions on the FPGA during the time it is waiting for the outputs from the CORDIC block.

4. References

[1] Simon Haykin, *Adaptive Filter Theory*, Prentice Hall, Fourth Edition

[2] Gentleman, W.M. and Kung, H.T., "Matrix triangularization by systolic arrays" *Real-Time Signal Processing IV*, Proc. SPIE 298, 19-26.

[3] G.Lightbody, R.L.Walke, R.Woods, J.McCanny, "Novel mapping of a linear QR architecture", *Proc. ICASSP*, vol IV, pp.1933-6, 1999

IEEE COMPUTER SOCIETY

Authorized licensed use limited to: National Chung Hsing Univ.. Downloaded on October 04,2022 at 09:15:40 UTC from IEEE Xplore. Restrictions apply.