The Design of a Fault Tolerant GEQRNS Processing Element for Linear Systolic Array DSP Applications

Jeremy C. Smith
Department of Electrical Engineering
University of Florida

Fred J. Taylor
Department of Electrical Engineering
University of Florida

Abstract
In this work the design of a Galois Enhanced Quadratic Residue Number System [4] processor is presented, which can be used to construct linear systolic arrays. The processor architecture has been optimized to perform multiply-accumulate type operations on complex operands. The processor is also shown to have a high degree of tolerance to manufacturing defects and faults which occur during operation. A prototype integrated circuit has been fabricated in 1.5 \mu m CMOS technology, which is shown to operate at 40 MHz.

1 Introduction
The residue numbering system (RNS) has long been proposed as a means of achieving high-computational bandwidths in signal processing systems [5]. An integer X in the RNS is represented by an L-tuple of residues: \( X = (x_0, x_1, \ldots, x_{L-1}) \) where \( x_i = X \mod m_i \) is the i-th residue and \( m_i \) is the i-th modulus. The production rule for the i-th digit in a RNS computation is \( z_i = x_i \mod m_i, y_i = x_i \mod m_i \), where \( \mod m_i \) represents modular addition or multiplication respectively. The importance of this equation is that the computation of any digit in the L-tuple is independent of any other digit. This means that there are no carries between the residue channels. If the residue channels are of small wordlength, then high computation rates can be achieved in physical systems. The rule which maps computations over the implementation rings back to the base ring is the Chinese Remainder Theorem (CRT). The CRT is given as \( X = \sum_{i=1}^{L} m_i \mod M \) \( [m_i]^{-1} \mod m_i, x_i \in M \) where \( M = \prod_{i=1}^{L} m_i \), and for \( i, j \in \{1, 2, 3, \ldots, L\} \), \( \gcd(m_i, m_j) = 1 \) for \( i \neq j \), and \( m_i = M/m_i \), with \( m_i^{-1} \mod m_i \). The legitimate range of integers in a RNS system is the interval \([0, M-1]\).

1.1 Complex Arithmetic
The quadratic residue numbering system (QRNS) was first introduced by Leung [3], and is an efficient way of performing complex operations. The QRNS requires that the moduli be restricted to primes of the form 4k + 1. If this is so, the equation \( x^2 \equiv -1 \mod p \) has two solutions in the ring \( Z_p \), denoted \( j \) and \( j^{-1} \), which are additive and multiplicative inverses of each other. We define a forward mapping \( \theta : Z_p \rightarrow Z_p \times Z_p \) to be

\[
\theta(a + jb) = (z, z^*) \quad \text{where} \quad z \equiv a + jb \quad \text{and} \quad z^* \equiv a - jb
\]

We will call the \( z \) and \( z^* \) operands the normal and conjugate components, respectively.

The inverse mapping \( \theta^{-1} : Z_p \times Z_p \rightarrow Z_p \) is given by

\[
\theta^{-1}(z, z^*) = 2z^{-1}(z + z^*) > p + j > 2^{-1}j^{-1}(z - z^*) > p
\]

If \( (z, z^*), (w, w^*) \in Z_p \times Z_p \), then addition and multiplication operations in the ring \( Z_p \times Z_p \) are given by

\[
(z, z^*) + (w, w^*) = (z + w, z^* + w^*)
\]

\[
(z, z^*)(w, w^*) = (zw, z^*w^*)
\]

The \( z \) and \( z^* \) channels are independent and can be implemented in two separate, parallel channels. Complex arithmetic can thus be performed in one clock cycle.

The properties of Galois fields can be used to further simplify complex multiplication in the RNS. It is well known that for any prime modulus \( p \) there exists some \( \alpha \in Z_p \) that generates all non-zero elements of the field \( GF(p) \). That is, any non-zero element in \( Z_p \) can be represented by \( \alpha^k \), where \( k \in \{0, 1, 2, \ldots, p - 2\} \). Since we can represent all elements of \( GF(p) \) by exponents, multiplication can be performed via exponent addition. This is highly desirable from a hardware standpoint since n-bit adders tend to be smaller and faster than n-bit multipliers. A number theoretic logarithm table is used to obtain the power of \( \alpha \) for each QRNS operand, and an antilogarithm table is used to recover the summed powers (modulo \( p-1 \)).

2 Processing Element Architecture
Figure 1 depicts the architectural details of the GEQRNS processing element, and a die photograph is shown in Figure 2. Two eight-bit operands to be
multiplied, $x$ and $y$, are the exponents of elements $\sigma^x, \alpha^y \in GF(p) - \{0\}$. The $x$ operand bus (X-bus) supplies data to the multiplier and to the input of the data storage shift register (DSSR), which can be used to store up to sixteen operands which are known a priori. Once data has been loaded into the DSSR, it can be circulated continuously via an internal feedback path. The X-Bus can be freed for other global data-move operations when local prestored data is used.

Just prior to multiplication, it is necessary to check for a zero operand, since zero must be handled as an exception in the GEQRMNS. An unused binary code word is chosen as the GEQRMNS zero (i.e. some value between $p$ and $2^3 - 1$). In this case, $255_p$ was chosen, since zero detection will simply be the logical AND of all the data bits. If a zero is detected for either operand, a flag is exerted, which will physically hardwire the input of the exponentiation table’s pipeline register to zero after the next clock cycle. The output of the modular adder provides the address input to the exponentiation table. The table has the value of $\sigma^{x+y} \alpha^{-1}$ programmed at each corresponding address location. The QRNS product is thus obtained at the output of the ROM. The computed product is then fed to one input of another modulo adder, which reduces the computed sum $\bmod(p)$. This modulo adder can be used as an accumulator (i.e. with feedback for in-place algorithms), or can accept partial-sum QRNS results from adjacent processors.

3 Key Circuit Elements

3.1 ROMs

The schematic diagram of the exponentiation ROM is shown in Figure 3. The ROM is mask programmed by including a contact to connect a pre-existing programming (discharge) transistor to the bit line. Since these programming transistors are of minimal width, they will limit operational speed due to long bit line pull down delays. The ROM logic was partitioned so that the bit line height could be kept relatively short (i.e. reduced parasitic capacitance). Differential sensing techniques were also used, so that a logical zero could be determined well before the bit line has fully discharged. Figure 3 reveals that the ROM is precharged during the low clock level, which

![Figure 1: Processor Architecture](image1.png)

![Figure 2: Die photograph of processor (2.4mm x 2.4mm).](image2.png)

is just after new data has arrived at the address inputs. The data is stable during the evaluation (high) clock stage, when the wordline decoders become active. A precharge transistor has also been included at the sense amplifier/bit-line input, so that this point may fully charge to $V_{dd}$ (i.e. this point would slowly charge to approximately 3V through the two series NMOS devices). In this way, the low cycle duration of the clock can be shortened, and more time can be given to the high cycle (where bit line pull down occurs) without adding to the total clock period. A true-single-phase-clocking (TSPC) scheme is thus employed, in which new data is latched on the falling edge of the clock.

3.2 Modular Adders

Figure 4 depicts the basic construction of the modulo adders used for the multiplier and accumulator portions of the PE. Here, two binary adders are cascaded such that the output of the first adder is input to the second adder which has an appropriate offset added. Since the datapath is only eight bits, ripple carry adders are used and the offset added to the second adder is the value $(2^8 - p)$. The correct $\bmod p$ sum is selected from the outputs of either the first or second adder via a multiplexer controlled by the logical OR of the most significant carry bits of the first and second adders. Normally, two full adders (FA) are used, but this is wasteful as the offset value is known a priori. In this implementation, the offset has been bit programmed by only implementing the logic corresponding to an added zero or one at each bit position of the second adder. Basic $\bmod p$ adder primitive cells are then constructed from a FA and the logic corresponding to offset-zero and offset-one. A transmission gate adder was used to implement the FA circuit. These blocks can then be used to construct a $\bmod p$ adder of arbitrary value.
4 Fault Tolerance and Yield

In an effort to improve the survivability and manufacturability of large linear chains of PEs, electronic reconfiguration switches were incorporated into the PE architecture. These elements are the outermost transmission gates in Figure 1. One configuration latch is needed for all reconfiguration switches in the PE, since all busses are switched out simultaneously. It is important to note that the successful operation of the reconfiguration circuitry only depends on twenty-four transistors (i.e., 12 transmission gates) and twenty-four wire segments per PE.

The Negative Binomial Model has been used successfully to predict integrated circuit yield, since the effects of defect clustering are modeled. Clustering has been shown to apply to large areas of a wafer, which usually exceed the area of a typical chip (known as large area clustering). If large area clustering is assumed, it is common in analysis to assume that defects are randomly distributed within the cluster (i.e., within a chip). In our yield analysis, we will assume large area clustering. We will also consider average values of the model parameters only (i.e., averaged over all defects). A more accurate analysis would involve determining susceptible areas on a net by net basis for applicable defect mechanisms (i.e., for each mask), but this is beyond the scope of this presentation. The Negative Binomial Model is given as:

$$Y = \left( 1 + \frac{D_0 A}{\bar{a}} \right)^{-\bar{a}}$$  \hspace{1cm} (1)

where $\bar{a}$ represents the clustering parameter which usually ranges between 0.3 and 5, $D_0$ represents the average fatal defect density, and $A$ is the chip area. Conservative values for $D_0$ are between 1 and 2 fatal defects per cm$^2$, and $\bar{a} < 1$.

4.1 Yield Enhancement via Reconfiguration

Traditional yield models for circuits which have identical replicated cells of which $m$ out of $n$ must function, typically assume statistical independence of failures and that all failures can be corrected by the redundancy scheme. The yield of the reconfiguration circuitry is thus assumed to be 1. Assuming that the reconfiguration yield is 1 is increasingly being shown to be a bad assumption, particularly if the system area is large. What is really assumed is that $m$ out of $n$ cells are free from defects that affect reconfigurable nets and that all $n$ cells are free from defects that affect non-reconfigurable nets. A more accurate approach has recently been presented [1], which includes the yield of the reconfiguration circuitry. To be strictly correct, we must also consider defects that affect global signals such as power, ground, and clocking, as they cannot directly be reconfigured for. Again, this requires a very detailed consideration (layout extraction) of susceptible areas on a net by net basis. For our purposes, we can consider cell yield as being $Y_{cell} = (Y_{cell-r})(Y_{cell-n})$, where $Y_{cell-r}$ is the yield of the cell due to defects in reconfigurable nets, and $Y_{cell-n}$ is the yield due to at least one non-reconfigurable net defect. The block yield then becomes:

$$Y_b = \left[ \sum_{i=0}^{n-m} \binom{n}{i} (Y_{cell-r})^{n-i} (1 - Y_{cell-r})^i \right]$$
$$\times [Y_{cell-n}]^m$$  \hspace{1cm} (2)

Since it is desirable that a very high switch yield be exhibited, the layout ground rule used for these elements was less aggressive than that of other circuits in the chip. For example, interconnection lines were run in metal two, with increased line thickness and spacing and with no logic placed underneath the lines. Increasing the size or spacing of circuit elements greatly increases their survivability as the defect frequency tends to decrease with the cube of the defect radius. The layout of the switching elements was also greatly relaxed.

4.2 Yield Estimates

For evaluation purposes, we will consider a sixteen processor, four modulus system with one spare processor per modulus. The dynamic range of such a
system is greater than thirty bits, and with eight-bit operands (signed) and sixteen-bit (signed) filter coefficients, and inner product of length 120 could be computed (length 1.925 for twelve bit coefficients). Our areas are valid for a 0.8 μm process, with λ = 0.4 μm. We will consider the case of no redundancy first, and then compare this to the reconfigured yield. For simplicity, assume average fatal defect densities of 1, 1.5 and 2.0 per cm² for all areas on the chip except the reconfiguration switches in the PEs. Since the fatal defect density will be smaller for these areas (i.e. fewer masks steps), the fatal defect density is assumed to be one tenth that for logic [2], or 0.1, 0.15 and 0.2 per cm², respectively. For the non-redundant case, consider the entire chip area (1.39 cm²) at each value of the fatal defect density.

The yields for no redundancy are presented in Table 1 for various values of the clustering parameter $\sigma$. Equation 1 has been used to obtain the values in Table 1.

Let us now consider the case for the redundant sixteen processor array. The total area of each processing element is $0.008464$ cm² and the area of the switching elements in the PE is $0.002208$ cm² ($A_{PE-i}$). The difference between these two areas ($0.002256$ cm²) represents the area of the PE which can tolerate a circuit fault ($A_{PE-i}$). These areas are used in Equation 1 with each value of $D_0$ and $\sigma$, to produce $Y_{PE-i}$ and $Y_{PE-n}$. $Y_{PE-i}$ and $Y_{PE-n}$ are then substituted in the expression:

$$Y_{Mod} = \frac{[Y_{PE-i}]^{17} + 17(Y_{PE-i})^{16}(1 - Y_{PE-i})}{[Y_{PE-n}]^{17}}$$

(3)


\begin{tabular}{|c|c|c|c|}
\hline
\textbf{Projected Non-Redundant Chip Yields} & \\
\hline
$\alpha$ & Average Fatal Defect & Densities per [cm²] & \\
& & 1.0 & 1.5 & 2.0 & \\
\hline
0.25 & 0.6246 & 0.5717 & 0.5357 & \\
0.50 & 0.5139 & 0.4394 & 0.3901 & \\
0.75 & 0.4550 & 0.3684 & 0.3125 & \\
1.00 & 0.4179 & 0.3237 & 0.2641 & \\
2.00 & 0.3474 & 0.2392 & 0.1746 & \\
\hline
\end{tabular}

Table 1: Table of Non-Redundant Chip Yields.

\begin{tabular}{|c|c|c|c|}
\hline
\textbf{Projected Redundant Chip Yields} & \\
\hline
$\alpha$ & Average Fatal Defect & Densities per [cm²] & \\
& & 1.0 & 1.5 & 2.0 & \\
\hline
0.25 & 0.7647 & 0.6777 & 0.5992 & \\
0.50 & 0.7353 & 0.6343 & 0.5459 & \\
0.75 & 0.7222 & 0.6138 & 0.5199 & \\
1.00 & 0.7147 & 0.6017 & 0.5043 & \\
2.00 & 0.7021 & 0.5807 & 0.4764 & \\
\hline
\end{tabular}

Table 2: Table of Redundant Chip Yields.

We see that it is possible to increase the yield from 62.46 % to 76.47 % in the best case ($\sigma = 0.25$, $D_0 = 1$), and from 17.46 % to 47.64 % in the worst case ($\sigma = 2.0$, $D_0 = 2$). The increase in yield came from a modest increase in area $\approx 4.88 %$.

5 Conclusions

In this work, a high-throughput complex multiply-accumulate processor, which is shown to have a high degree of fault tolerance was presented. The chip was fabricated in a 1.5 μm CMOS technology and operates at 40 MHz. The processor architecture is amenable to signal processing applications such as convolution, Fourier Transform techniques, and more general matrix operations. A fault tolerant linear array, which consists of sixteen processors was also presented. Fault tolerance was provided by means of the reconfiguration of defective processors. This fault tolerance was introduced by a 5 % increase in circuit area, and can correct for faults which occur at the time of manufacture and those which occur in the field.

References


