LOW-ENERGY HETEROGENEOUS DIGIT-SERIAL REED-SOLOMON CODECS

Leilei Song, Keshab K. Parhi

Department of Electrical
and Computer Engineering
University of Minnesota
Minneapolis, MN 55455, USA
E-mail:{*llsong, parhi*}*@ece.umn.edu*

Ichiro Kuroda, Takao Nishitani

Digital Signal Processing Lab
NEC Corporation
4-1-1 Miyazaki, Miyamae-Ku
Kawasaki 216, Japan
Email: {*kuroda, takao*}*@dsp.cl.nec.co.jp*

ABSTRACT

Reed-Solomon (RS) codecs are used for error control coding in many applications such as digital audio, digital TV, software radio, CD players, and wireless and satellite communications. This paper considers software-based implementation of RS codecs where special instructions are assumed to be used to program finite field multiplication datapaths inside a domain-specific programmable digital-signal processor (DS-PDSP). A *heterogeneous digit-serial* approach is presented, where the *heterogeneity* corresponds to the use of different digit-sizes in the multiply-accumulate (MAC for polynomial multiplication) and degree reduction (DEGRED for polynomial modulo operation) subarrays. The salient feature of this digit-serial approach is that only the digit-cells are implemented in hardware, the finite field multiplications are performed digit-serially in software by dynamically scheduling the internal digit-level operations in RS encoders and decoders. It is concluded that, for 2-error-correcting RS(n, k) codec implementations over finite field $GF(2^8)$, a parallel MAC unit (of digit-size 8) and a DEGRED unit with digit-size 2 is the best datapath, with respect to least energy consumption and energy-delay products; with this datapath architecture and appropriate digit-serial scheduling strategies, more than 60% energy reduction and more than $1/3$ energy-delay reduction can be achieved compared with the parallel multiplication datapath based approach.

1. INTRODUCTION

The arithmetic over finite fields are the underlying operations in many error-control coding algorithms. Sophisticated error correcting codes, such as Reed-Solomon codes, are extremely important in both broadband and narrowband digital communication systems. There has been considerable interests in designing dedicated circuits for RS encoders and decoders. Design of universal RS codecs which can operate over different finite fields with variable code rate has practical importance and still remains an active research topic [1]. With current scaled technologies, many DSP algorithms based on binary arithmetic can be realized using domain-specific programmable digital signal processors (DS-PDSP) optimized for targeted applications. This processor based software approach is a very efficient design alternative and it reduces the time-to-market and the design costs. If finite field arithmetic would be implemented in a programmable DSP datapath, the universal RS codecs (and other finite field based systems) could be easily implemented in software. The finite field $GF(2^m)$ is generally used in practice. In $GF(2^m)$, addition and subtraction are bit-independent and can be computed using array of XOR gates. Inversion (as well as division) can be computed iteratively using multiplication. Therefore, to design DSP datapath for finite field arithmetic, we concentrate only on the multiplication. One combined binary and finite field arithmetic datapath architecture has been proposed in [2].

With the increased levels of integration and the current focus on wireless systems, minimization of energy consumption in the presence of performance constraints has become increasingly important. Energy reduction techniques have been proposed at all levels of the design hierarchy, from algorithmic and architectural to circuits and technological innovations. Both energy and energy-delay product are important metrics for analyzing the system performance since optimizing performance and power are tightly interwoven. The energy consumption of a system depends on both the hardware components and the software programs of a system. In this paper, we use a *hardware software co-design* methodology and achieve power reduction at algorithmic and architectural level.

Digit-serial architectures are best suited for systems requiring moderate sample rate and where area and power consumption are critical. Two types of digit-serial multiplication schemes have been presented in [3] for the design of dedicated digit-serial finite field multipliers. In this paper, we consider the design of low-energy high-performance Reed-Solomon encoders and decoders using a novel *heterogeneous digit-serial* approach. In this approach, one parallel finite field multiplier is decomposed into MAC and DEGRED subarrays, and different digit-sizes, $D1$ and $D2$ ($D1 \geq D2$) are assigned to each of them, respectively. The salient feature of this digit-serial approach is that only the digit-cells, MAC_D1 and DEGRED_D2 are implemented in hardware, the finite field multiplications are performed digit-serially in software by dynamically scheduling the internal digit-level operations. By using this approach, the dataflow control tasks in digit-serial systems are moved from hardware to software and each digit-cell is considered as an independent unit, as illustrated in Figure 1. The advantage of this heterogeneous digit-serial approach is two fold. First, various datapath architectures can be obtained by varying digit-sizes $D1$ and $D2$, and the design space can be explored to a great extent for a given design constraint, such as area or energy consumption. Second, various digit-serial multiplication algorithms can be used to dynamically schedule the MAC and $DEGRED$ operations. Moreover, in RS encoders and decoders, digit-level operations can be treated beyond the limitation of one multiplica-

This research was supported in parts by the National Science Foundation under grant number INT-9600372 and the Army Research Office under grant number DA/DAAH-94-G-0405.

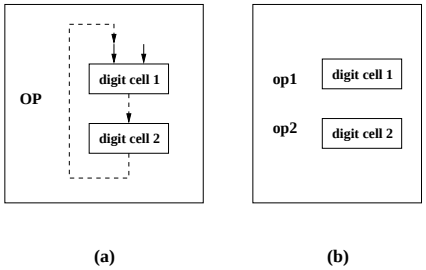

Figure 1. (a). Hardwired digit-serial unit; (b). Software digit-serial approach based on independent digit-cells

tion operation and optimally scheduled at system level such that the system energy consumption and latency can be reduced. Low-energy Reed-Solomon codecs are designed in software based on various finite field datapath architectures. It is concluded that, for 2-error-correcting RS(n, k) codec implementations over finite field $GF(2^8)$, a parallel MAC unit (of digit-size 8) and a DEGRED unit with digit-size 2 is the best datapath, with respect to least energy consumption and energy-delay products; with this datapath architecture and appropriate digit-serial scheduling strategies, more than 60% energy reduction and more than $1/3$ energy-delay reduction can be achieved compared with the parallel multiplication datapath based approach.

This paper is organized as follows. Section 2 gives a brief overview of some basic concepts, algebraic structures of finite field. Section 3 presents the *heterogeneous* digit-serial finite field datapaths in DS-PDSP processors. Section 4 addresses the efficient energy reduction approaches and their application to the design of low-energy high-performance RS codecs.

2. FINITE FIELD FUNDAMENTALS

Knowledge of basic finite field concepts and properties is assumed, as covered in [4], [5], [6].

Finite field $GF(2^m)$ contains 2^m elements. It is an extension field of $GF(2)$, which has elements 0 and 1. All finite fields contain a zero element, a unit element, a primitive element and have at least one primitive polynomial $p(x) = x^m + p_{m-1}x^{m-1} + \cdots + p_1x + p_0$ over $GF(2)$ associated with it. The primitive element α generates all nonzero elements of $GF(2^m)$ and is a root of the primitive polynomial $p(x)$, i.e.,

$$\alpha^m = p_{m-1}\alpha^{m-1} + p_{m-2}\alpha^{m-2} + \cdots + p_1\alpha + p_0. \quad (1)$$

The finite field $GF(2^m)$ can either be represented in *exponential representation* as:

$$GF(2^m) = \{0, \alpha^0, \alpha^1, \cdots, \alpha^{2^m-2}\}, \quad (2)$$

or in *polynomial representation* as:

$$GF(2^m) = \{A | A = a_{m-1}\alpha^{m-1} + a_{m-2}\alpha^{m-2} + \cdots + a_1\alpha + a_0, where\ a_j \in GF(2),\ 0 \le j \le m-1\}. \quad (3)$$

The polynomial representation is generally used for finite field arithmetic operation, where addition is carried out using bit-independent XOR operations, and multiplication is carried out using polynomial multiplication and modulo operations over $GF(2)$. The polynomial representation is based on the *basis representation*, where $\{1, \alpha, \alpha^2, \cdots, \alpha^{m-1}\}$ is known as the *standard basis* which is also referred to as the polynomial basis or conventional basis. The reader may refer to [7] for a detailed description of parallel finite field multiplication algorithms.

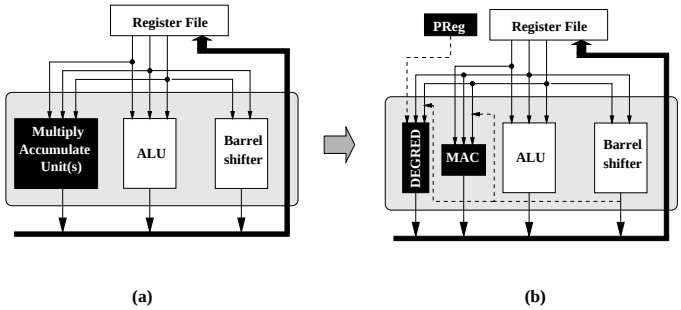

Figure 2. Finite Field datapath architectures in a DS-PDSP processor (a). Parallel datapath; (b). Digit-serial datapath

3. DIGIT-SERIAL FINITE FIELD MULTIPLICATIONS

In this section, we present a *heterogeneous* digit-serial approach for the design of finite field datapath in the DS-PDSP processor. Since most practical RS codecs are operated over $GF(2^8)$, we only consider finite field multiplication over $GF(2^8)$ in this section. The corresponding datapath architectures are built for $GF(2^8)$. However, they can be programmed to compute multiplications over finite field $GF(2^m)$, for $m \le 8$, by applying zero-padding and extra shifting operations to the operands.

3.1. Digit-serial datapath architectures

A domain-specific programmable DSP processor (DS-PDSP) [8] is assumed whose datapath is specialized for finite field operations, especially multiplications. There are two major operations involved in finite field multiplication, namely polynomial multiplication and polynomial modulo operation over $GF(2)$. They can either be implemented as a whole in one parallel multiplier as shown in the parallel datapath architecture in Figure 2(a), or they can be implemented separately as shown in the digit-serial datapath architecture in Figure 2(b), where the MAC array is for polynomial multiplication and has a digit-size $D1$, the DEGRED array is for polynomial modulo operation and has a digit-size $D2$. The heterogeneous digit-serial approach is based on the datapath architecture shown in Figure 2(b).

3.2. Heterogeneous digit-serial finite field multiplications

In the heterogeneous digit-serial approach, one $MAC\ 8 \times D1$ array of digit-size $D1$ and one $DEGRED\_D2$ array with digit-size $D2$ are assumed in the DSP datapath, and two corresponding instructions, MAC and $DEGRED$ are assigned to them. $MAC8 \times D1$ array computes the product of two polynomials over $GF(2)$ with degree 7 and degree $D1 - 1$, respectively, and accumulates their product to some initial value or some intermediate result from previous computation using tree-type structure. $DEGRED\_D2$ array performs polynomial modulo operation and reduces the degree of input polynomial by $D2$. It consists of $D2$ linearly connected rows of $DEGRED1$, whose structure is as shown in Figure 3, where wl denotes the input wordlength, and p_i ($0 \le i \le 7$) are the coefficients of the primitive polynomial $p(x)$. Various digit-serial multipliers can be obtained by varying $D1$ and $D2$. To simplify the analysis, we assume that $D1$ can only be divisors of $M = 8$, i.e., 8, 4, 2, 1; $D2$ can only be divisors of $D1$, Hence, there are totally 10 possible combinations of hardware units: (MAC8, DEGRED7), (MAC8, DEGRED4), (MAC8, DEGRED2), (MAC8, DEGRED1), (MAC4, DEGRED4), (MAC4, DEGRED2), (MAC4, DEGRED1), (MAC2, DEGRED2), (MAC2, DEGRED1)

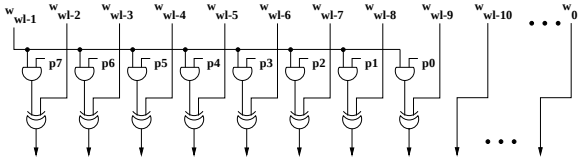

Figure 3. Circuit for One row of $DEGRED1$ array

and (MAC1, DEGRED1). Note that when $D1 = M = 8$, the maximum degree of the intermediate result is equal to 15 which requires degree reduction by 7 only. Hence instead of having $D2 = D1 = 8$, we set maximum value of $D2$ to 7.

Example 3.1 *Assume that the finite field datapath consists of (MAC4, DEGRED4). One multiplication over $GF(2^8)$ can be performed as follows:*

MAC4
MAC4
DEGRED4
DEGRED4

4. SOFTWARE-BASED LOW-ENERGY RS CODECS

In this section, we consider the design of low-energy high-performance RS codecs based on various datapath architectures presented in last section, and the selection of the datapath architecture with the digit-sizes combination that leads to the least energy and energy-product for RS codecs. Energy consumption are estimated using the HEAT tool proposed in [9] at a clock frequency of 50MHZ, using NEC 0.35μ technology, at $3.3Volts$ supply voltage.

4.1. RS(n, k) codes

Reed-Solomon codes are optimal multiple-error-correcting codes and are efficient for correcting both burst and random errors. A t-error-correcting primitive RS(n, k) code with symbols from $GF(2^m)$ has codewords of length $n = 2^m - 1$ and satisfies $n - k = 2t$. Any primitive RS code can be shortened, that is, changed from (n, k) code (with $n = 2^m - 1$) to an $(n - b, k - b)$ code by dropping b most significant information symbols from each codeword where $b < k$. Because the dropped symbols in a shortened code are always set to zero, they do not need to be transmitted, and the receiver can assume them to be zero for decoding. The decoding process is then simplified. Error-correction capability of the shortened RS code is the same as the original RS code, while the code rate is slightly reduced from $R = \frac{k}{n}$ to $R = \frac{k-b}{n-b}$.

In this section, 2-error-correcting RS(n, k) codes over $GF(2^8)$ are considered, where $n \leq 255$ and $k = n - 4$. RS(n, k) codes with $n < 255$ is shortened from RS(255,251) code. A systematic generator-matrix based method is used for RS encoding, which is based on vector-matrix multiplications over $GF(2^8)$. Peterson-Gorenstein-Zierler algorithm [4] is used for RS decoding, where the syndrome computation is based on vector-matrix multiplications over $GF(2^8)$ using precomputed lookup table which requires storage of $4n$ bytes. The reader may refer to [4] for a detailed explanation of RS encoding and decoding algorithms.

4.2. Energy reduction approaches

As can be seen, vector-vector (or vector-matrix) multiplications are the common and frequently used computations in both RS encoders and decoders. Consider the vector-vector multiplication over $GF(2^m)$,

$$\begin{aligned} &\begin{bmatrix} A_0 A_1 \cdots A_{N-1} \end{bmatrix} \begin{bmatrix} B_0 \\ B_1 \\ \cdots \\ B_{N-1} \end{bmatrix} \\ &= A_0B_0 + A_1B_1 + \cdots + A_{N-1}B_{N-1}, \end{aligned} \quad (4)$$

where A_i and B_i, $0 \leq i \leq N-1$, are elements of $GF(2^m)$ with wordlength equal to m. With one parallel multiplier in the datapath (as shown in Figure 2(a)), this vector-vector multiplication requires N multiply-accumulate operations and consumes $12.256N\ pJ$ energy if non-pipelined fully-parallel multiplier is used, or $8.986N\ pJ$ energy if one-level pipelined fully-parallel multiplier is used. Notice that in finite field vector-vector multiplication, polynomial modulo operation can be delayed till the last step, and the intermediate result can be obtained by polynomial multiplication and accumulation only. As a result, using the digit-serial datapath in Figure 2(b), the vector-vector multiplication only requires N $MAC8$ and 1 $DEGRED7$ operations. One $MAC8$ operation consumes $2.858pJ$ energy and one $DEGRED7$ operation consumes $7.657pJ$ energy. Therefore, compared with the non-pipelined parallel multiplier datapath, about 77% energy consumption can be saved at the expense of increasing instruction count by one; compared with the one-level pipelined parallel multiplier datapath, about 68% energy consumption can be saved with the same latency.

Therefore, by separating MAC and $DEGRED$ operations and scheduling them beyond the limitation of one multiplication operation at system level, system energy consumption can be reduced dramatically with little overhead. Furthermore, MAC and $DEGRED$ can be operated in parallel on different operands, which leads to reduction in latency. These schemes are used to program low-energy RS encoders and decoders in the next section.

4.3. Heterogeneous digit-serial low-energy 2-error-correcting RS(n, k) codecs over $GF(2^8)$

RS($n, n-4$) encoders and decoders are programmed following the energy reduction rules presented in last subsection. It turns out that RS codecs based on different datapaths requires same number of instructions other than MAC and $DEGRED$, and same number of memory accesses. Therefore, in Table 1 we only summarize and compare multiplication related operations in terms of the codeword length n. From Table 1, we can conclude that the RS codecs based on datapaths with (MAC8, DEGRED4) units, or with (MAC8, DEGRED2) have the least energy and energy-delay products.

Comparing the performance of RS codecs based on the parallel datapath (Figure 2(a)) and the digit-serial datapath containing MAC8+DEGRED2, we can observe that the digit-serial RS encoders only consume 23.3% of the energy of the parallel approach with slight increase in latency; the digit-serial RS decoders consume about 27.6% of the energy of the parallel approach with 1.67 times latency. The energy-delay products of RS(n, k) encoders and decoders are plotted as a function of the block length n in Figure 4 assuming the the clock cycle time in both datapaths are the same. From Figure 4 we can see that the energy-delay products of RS encoders can be reduced by $2/3$ and the energy-delay products of RS decoders can be reduced by at least $1/3$ using the digit-serial approach.

Note that the critical path of each digit-cell (MAC and DEGRED) is shorter than that of the parallel multiplier. Therefore, the digit-serial datapath can be operated at higher clock rate, which reduces the computation delay ($latency \times T_{clk}$) of the digit-serial RS

Table 1. Comparison of the energy consumption and latency of RS(n, k) encoders and decoders based on various datapath architectures

Datapath	Crit. path	Area	Encoder		Decoder	
			Energy	Latency	Energy	Latency
			(pJ)	(# cycles)	(pJ)	(# cycles)
Parallel-Multiplier	$19D$	1680	$49.024n - 196.096$	$4n - 16$	$73.536n + 453.472$	$6n + 37$
(MAC8, DEGRED7)	$14D$	1680	$11.432n - 14.632$	$4n - 12$	$29.838n + 432.254$	$7n + 79$
(MAC8, DEGRED4)	$8D$	1344	$11.432n - 21.2$	$4n - 8$	$23.488n + 363.29$	$8n + 121$
(MAC8, DEGRED2)	$5D$	1120	$11.432n - 26.448$	$4n$	$20.315n + 308.186$	$10n + 205$
(MAC8, DEGRED1)	$5D$	1008	$11.432n - 27.388$	$4n + 12$	$19.530n + 298.316$	$13n + 331$
(MAC4, DEGRED4)	$8D$	896	$12.448n - 25.264$	$8n - 24$	$24.758n + 372.688$	$12n + 123$
(MAC4, DEGRED2)	$4D$	672	$12.448n - 30.512$	$8n - 16$	$21.585n + 317.584$	$14n + 207$
(MAC4, DEGRED1)	$4D$	560	$12.448n - 31.452$	$8n - 4$	$20.80n + 307.714$	$17n + 333$
(MAC2, DEGRED2)	$4D$	448	$15.072n - 41.008$	$16n - 48$	$24.865n + 341.856$	$22n + 211$
(MAC2, DEGRED1)	$3D$	336	$15.072n - 41.948$	$16n - 36$	$24.08n + 331.986$	$25n + 337$
(MAC1, DEGRED1)	$2D$	224	$21.312n - 66.908$	$32n - 100$	$31.88n + 389.706$	$41n + 345$

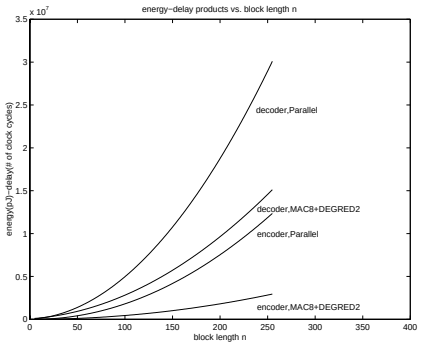

Figure 4. Comparison of the energy-delay products of RS codecs over $GF(2^8)$ based on the parallel datapath and the digit-serial datapath containing MAC8+DEGRED2

codecs without increasing the energy consumption. (Recall that in a digital CMOS circuit, 90% energy consumption are due to dynamic switchings and can be estimated using $E = \alpha CV^2$, where α is the activity factor, C is the load capacitance and V is the supply voltage [10]). Hence, further energy-delay reduction is possible for the digit-serial approach by either operating the digit-serial system at higher clock rate (to reduce the total computation delay) or using lower supply voltage (to reduce the energy consumption).

5. CONCLUSIONS AND FUTURE WORK

In this paper, a hardware-software co-design approach has been used to design the datapath for a domain-specific DSP processor. Special datapath architectures for finite field arithmetic operations have been constructed with low-energy RS codecs application in mind. A novel heterogeneous digit-serial approach was proposed to design digit-serial finite field datapaths. The salient feature of this digit-serial approach is that only the digit-cells are implemented in hardware, the finite field multiplications are performed digit-serially in software by dynamically scheduling the internal digit-level operations. Low-energy high-performance Reed-Solomon codecs are designed in software based on various finite field datapath architectures. It is concluded that, for 2-error-correcting RS(n, k) codec implementations over finite field $GF(2^8)$, a parallel MAC unit (of digit-size 8) and a DEGRED unit with digit-size 2 is the best datapath, with respect to least energy consumption and energy-delay products; with these datapath architectures and appropriate digit-serial scheduling strategies, more than 60% energy reduction and more than $1/3$ energy-delay reduction can be achieved compared with the parallel multiplication datapath based approach. Future work will be directed towards further energy-delay optimization by taking signal correlations into consideration.

6. REFERENCES

[1] S. B. Wicker and V. K. Bhargava, *Reed-Solomon Codes and Their Applications*, IEEE Press, New York, NY, 1994.

[2] W. Drescher and G. Fettweis, "VLSI Architectures for Multiplication in $GF(2^m)$ for Application Tailored Digital Signal Processors", *in Proc. of 1996 VLSI Signal Processing*, pp. 55–64, October 1996.

[3] L. Song and K. K. Parhi, "Efficient Finite Field Serial/Parallel Multiplication", *in Proc. of International Conf. on Application Specific Systems, Architectures and Processors*, pp. 72–82, Chicago, Aug 1996.

[4] R. E. Blahut, *Theory and Practice of Error Control Codes*, Addison Wesley, 1984.

[5] W. W. Peterson and E. J. Weldon, *Error-Correcting Codes*, The MIT Press, 1972.

[6] R. J. McEliece, *Finite Fields for Computer Scientists and Engineers*, Kluwer Academic, 1987.

[7] S. K. Jain, L. Song, and K. K. Parhi, "Efficient Semi-Systolic Architectures for Finite Feild Arithmetic", *IEEE Trans. on VLSI Systems*, vol. , to appear.

[8] I. Verbauwhede and M. Touriguian, "A Low Power DSP Engine for Wireless Communications ", *in Proc. of VLSI Signal Processing, IX*, pp. 471–480, San Francisco, CA, October 1996.

[9] J. H. Satyanarayana and K. K. Parhi, "HEAT: Hierarchical Energy Analysis Tool", *in Proc. 33rd ACM/IEEE Design Automation Conf.*, pp. 9–14, Las Vegas, June 1996.

[10] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low-Power CMOS Digital Design", *IEEE Journal of Solid State Circuits*, vol. 27, pp. 473–483, April 1992.