Reed-Solomon Codes for Low Power Communications

Lionel Biard, Dominique Noguet
CEA-LETI, Minatec, 17 rue des Martyrs, 38054 Grenoble Cedex 9, France
Email: lionel.biard@cea.fr ; dominique.noguet@cea.fr

Abstract—**Power consumption is a critical issue for many applications running on autonomous battery operated devices. In the context of low power communications, the use of Forward Error Correction (FEC) schemes shall therefore not only be considered as a way to improve communication robustness but also as a way to reduce transmit power. The benefits of this approach can be assessed by analyzing the tradeoff between the additional performance gained and the power consumption overhead resulting from extra computations.**

In this paper, a selection method is described to size the parameters of a low power Reed-Solomon (RS) code. The rationale highlights the influence of code parameters on computational complexity and performance of the code. An efficient set of code parameters can be deduced from this analysis to reduce the global power consumption of the system. Among the RS codes considered, the shortened RS(40,32,4) code over GF(2^8) is selected for further implementation analysis. Several design improvements are investigated step by step, and the relative power savings achieved with each enhancement are quantified. These results tend to illustrate the improvements that can be reached with only little design efforts.

I. Introduction

The development of low power communication systems, like Low Data Rate (LDR) Wireless Personal Area Networks (WPAN), leads to strong requirements on the power consumption of the transceivers. For such systems, the traditional way of selecting a FEC scheme only based on its performance shall be reconsidered. Indeed, its correcting capabilities shall be seen more as a way to decrease the transmit power than as a way to improve communication robustness. However, a high error correction capability usually comes at the cost of a significant power consumption overhead caused by the additional encoding/decoding computations. Therefore, the most advanced FEC schemes specifically designed to reach very high performances might not be the most appropriate choice in the context of this paper. In fact, the challenging issue in choosing the FEC scheme for a low power communication system is to find the best trade-off between its performance and power consumption.

Based on this trade-off, different types of codes, including soft-decoding methods, have been compared in [1]. However, several code specimen selected are not targeted for low power consumption. On the contrary, the current paper focuses only on one type of codes, and aims at choosing an optimal set of code parameters to meet low power constraints. RS codes have been selected in this study because they accommodate well with hardware complexity restrictions of LDR WPAN transceivers. The impact of code parameters is first analysed in Section II. For each parameter, both performance and power consumption issues are evaluated. A set of guidelines are finally provided to choose a suitable low power RS code.

After selecting a specific RS code, Section III of this paper investigates some hardware enhancements to further reduce the power consumption of the RS decoder implementation. Some topics have been theoretically described in [11]-[15], but the practical power consumption report of the current paper allows to quantify the real impact of each modification. This study is based on an FPGA implementation, but results can easily be extended to other targets.

This paper is an extended version of [2], and presents a more detailed study and refined conclusions compared to the previous version.

II. Selection of a Low Power Reed-Solomon Code

In this section, the impact of RS code parameters is analysed. Both performance and power consumption issues are tackled. Subsection A introduces the assumptions and methods used in the following to evaluate both of these aspects. Subsection B provide the detailed analysis itself, and is followed by recommendations to select a low power RS code in subsection C. Finally, a specific case study based on numerical results is presented in subsection D.

A. Methodology and Assumptions

RS codes [3][4] are defined by the set of three parameters (n,k,t), k and n being the number of symbols respectively before and after encoding, and t=(n-k)/2 the number of symbols which can be corrected among n. The code rate is denoted by R=k/n. Symbols take their values in a Galois Field GF(2^m), and are thus represented with m bits. The n parameter is bounded by 2^m. A lower value for n specifies a shortened RS code.

From the different existing decoding methods, frequency domain algorithms traditionally show the lowest computational complexity [5]. The classical architecture for this kind of algorithms is depicted in

TABLE I. COMPUTATIONAL COMPLEXITY OF REED-SOLOMON CODES BUILDING BLOCKS.

		GFinv	GFmul	GFmulαi	GFadd	GFreg	GFmem
Encoder				n.(2.t)	n.(2.t)	n.(2.t)	
Syndrome calculation				n.(2.t)	n.(2.t)	n.(2.t)	
BMA		2.t-1	$(2.t-1).(2.t+1) + t^2$		$(2.t-1).(2.t) + t^2$	$(2.t-1).(5.t-1) + t$	
EEA		t	t.(4.t)		t.(4.t)	t.(6.t+1)	
PGZ	for t = 1	1	1		2		
	for t = 2	1	9		4	4	
	for t = 3	1	27		6	15	
Chien Search				n.(2.t-1)	n.(2.t-1)	n.(2.t-1)	
Forney Algorithm		t	t				
Error correction					t		
Delay line							k
Total (with EEA)		2.t	t.(4.t+1)	n.(6.t-1)	n.(6.t-1)+ t.(4.t+1)	n.(6.t-1)+ t.(6.t+1)	k

figure 1. This block diagram is deeply described in section III, where implementation issues are discussed. For the current section, the main concern is the computational complexity of the RS building blocks. These complexity figures, obtained after a first implementation analysis, are displayed in table I. They are expressed in terms of Galois Field (GF) operations. GFadd represents a Galois Field addition. GFmulαi corresponds to the multiplication by a specific Galois Field element α^i, whereas GFmul is the multiplication of two unspecified Galois Field elements. GFinv provides the inverse of an element. And finally, register storage and memory storage of a Galois Field element are differentiated by GFreg and GFmem, because of their significantly different power consumption. For the Key Equation Solving, which is the core of the decoder, several algorithms can be used [3], Berlekamp-Massey Algorithm (BMA), Extended Euclidean Algorithm (EEA) or Peterson-Gorenstein-Zierler (PGZ) algorithm. Considering a classical implementation, it can be seen from table I that EEA [6] requires the least computations, except for low values of t ($t \leq 3$) where PGZ algorithm [7] performs better.

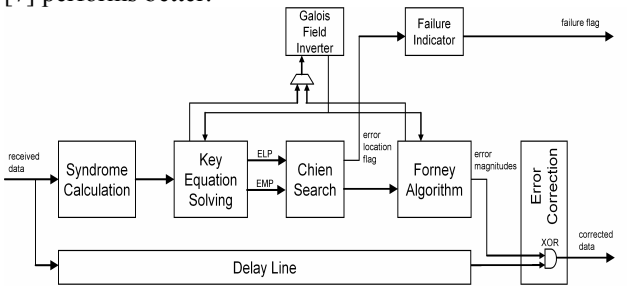

Figure 1. Block diagram of the Reed-Solomon decoder

From this table, the overall Computational Complexity per Information Bit (CCIB) can be obtained. This expression will be used in the next subsections to compare the digital power consumption for different set of code parameters. With EEA, it can be expressed as a function of t, R and m as follows:

$$CCIB = \frac{1}{m}\left[\frac{1-R}{2R}.(4t+1).GFmul + \frac{1}{R}.(6t-1).GFmul\alpha i\right] + \frac{1}{m}\left[\left(\frac{1}{R}.(6t-1)+\frac{1-R}{2R}.(4t+1)\right).GFadd + \frac{1-R}{R}.GFinv\right] + \frac{1}{m}\left[\left(\frac{1}{R}.(6t-1)+\frac{1-R}{2R}.(6t+1)\right).GFreg + 1.GFmem\right] \quad (1)$$

To evaluate this expression, the computational cost of the different GF operators is required. This is detailed in table II, along with their estimated power consumption, obtained from the specific FPGA implementation described in section III and from [1]. It can be noticed that the complexity of GF operators is *O(m)* or $O(m^2)$.

TABLE II. COMPUTATIONAL COST OF GF OPERATORS

GF operator	Computational Complexity	Power consumption (pJ)
1 GFadd	m XOR	0.4 m
1 GFmulαi	m.(m-2)/2 XOR	0.4 m(m-2)/2
1 GFmul	m^2 AND $3.(m-1)^2/2$ XOR	$0.4\ (m^2+3(m-1)^2/2)$
1 GFinv	m ROM read	8 m
1 GFreg	m REG write	2 m
1 GFmem	m RAM read and write	10 m

At last, for the performance analysis, the BER vs. E_b/N_0 curves from the next subsection have been obtained by simulation. A very simple transmission scheme has been used, including a BPSK modulation over an AWGN channel.

B. Performance and Power Consumption Analysis

Firstly, by comparing shortened RS codes with the same code rate R over the same Field GF(2^m), it can be noticed that the CCIB grows like *O(t)*. Figure 2 illustrates this behavior for several RS codes over GF(2^8) with the same rate R=0.8. From a strict computational complexity point of view, it is thus more interesting to choose a low-t RS code, implying a low value for n and k.

A similar comparison is shown in figure 3 from a performance point of view for the same set of RS codes. Same code rate implies that the error correction capability per information symbol t/k is constant. This t/k ratio provides of course a first estimation of the correction capability of the RS scheme used. But it can be noted in figure 3 that for low BER, even if this ratio is the same for every code, the larger the block size k (or n), the better the performance. Indeed, as errors are not equally distributed, it can be understood that a large block size allows better correction.

Combining both aspects, it can be deduced that, for a given code rate R, an optimal (n,k) pair should be determined in order to achieve a good trade-off between the computational complexity and the performance of the

code. However, this first conclusion should be related to other aspects analyzed hereafter.

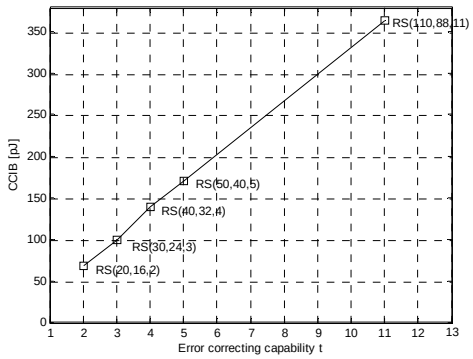

Figure 2. Consumption of RS codes over GF(2^8) with R=0.8

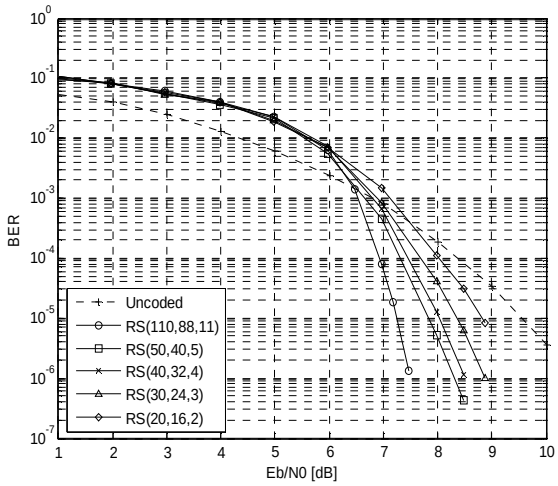

Figure 3. Performance of RS codes over GF(2^8) with R=0.8

In the same way, RS codes can be compared by fixing the error correction capability t, and choosing different code rates. From equation (1), it can be seen that the computational complexity has a 1/*x* dependence with R, which leads the choice towards a rather high value for R. Figure 4 illustrates this effect for RS codes over GF(2^8) with an error correction capability t=4 and code rates ranging from R=0.5 to R=0.97.

On the other hand, figure 5 illustrates the performance for the same set of codes with t=4. As it is known, the code rate introduces a *10.log(1/R)* shift in the performance curve which decreases the coding gain. Therefore, a high code rate is preferable in order to reduce the shift of the curve. However, concerning the slope of the curve, two antagonist effects can be distinguished. With a fixed t, a higher code rate corresponds to a larger block size n, but also results in a smaller t/k ratio. The former aspect, like before, has a positive impact on performances, while a smaller t/k decreases the slope of the curve. As a matter of fact, these two effects are more or less counterbalanced depending on the values of the parameters. With t=4, it can be seen that the former aspect is preponderant, strengthening the choice of a high code rate. However, the influence of the t/k ratio can not be neglected, as emphasized hereafter.

From these results, it can be concluded that a high code rate seems to be the best solution to achieve not only a low computational cost but also better performances, when considering a fixed error correction capability.

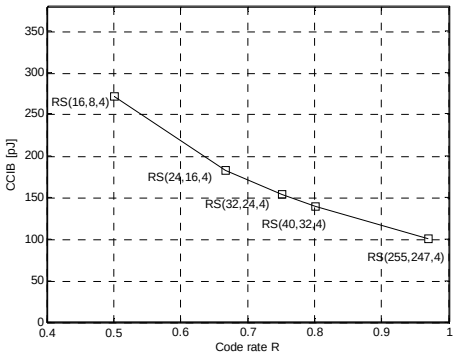

Figure 4. Consumption of RS codes over GF(2^8) with t=4

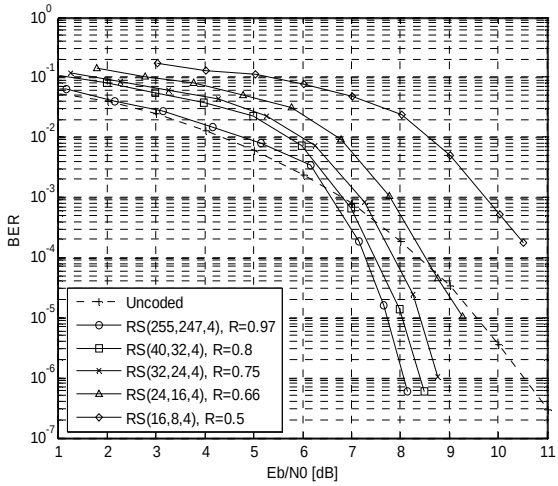

Figure 5. Performance of RS codes over GF(2^8) with t=4

Finally, RS codes over GF(2^8) with a fixed data block size n=255 and different error correcting capabilities t are compared below. The CCIB can be re-expressed as a function of n and t, considering that *R=1-2t/n*. This shows again the strong influence of t on the digital power consumption as depicted in figure 6.

As suggested before, the main interest when considering a fixed data block size lays in the performance analysis of figure 7. In this case, the curves are again subject to the shift due to R. However, the influence of the block size is less perceptible as n is fixed, and their slope are strongly related to the t/k ratio. The weight of this last effect is thus well illustrated. For low values of t/k, this impact is clearly visible on the graph, but it can be noticed that the slope is not significantly improved any more when t/k reaches a certain level. Therefore, a too high value for t/k is not advised, as it increases the shift of the curve without improving its slope. It can be noticed for example that RS(255,205,25) and RS(255,225,15) have a comparable slope, despite the gap between their respective t/k ratio. Besides, RS(255,185,35) does not reach the same level of performances despite its even higher t/k ratio, because the impact of R becomes determinant in this case. From this

point, it can be concluded that an optimum value must be found for t to maximize performances.

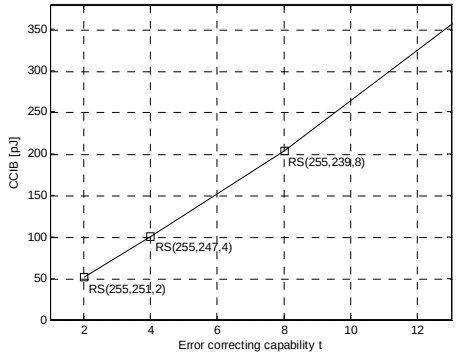

Figure 6. Consumption of RS codes over GF(2^8) with n=255

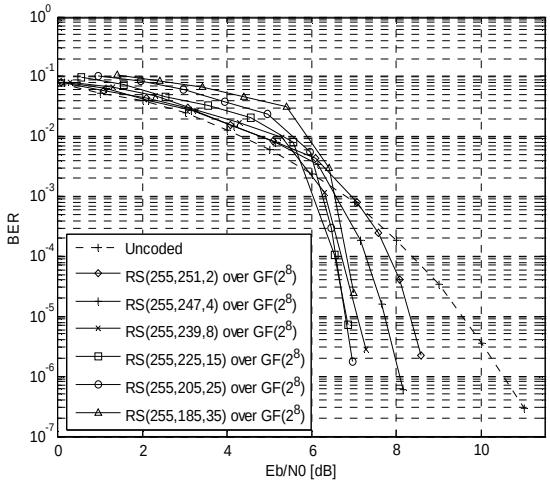

Figure 7. Performance of RS codes over GF(2^8) with n=255

In a last step, the influence of the GF size is analysed. This parameter is mathematically quite independent from the other parameters choice. Regarding the CCIB, only GFmul and GFmulαi, which are $O(m^2)$, are concerned by the GF size. As they do not represent the major part of the CCIB, modifying m do not have a strong influence on power consumption (see figure 8).

From a performance point of view, figure 9 shows the BER achieved with a shortened RS(40,32,4) code over different GF. It appears that the smallest GF code performs slightly better. Indeed, when the number of errors considered in one data block of a large GF code exceeds its error correction capability, a smaller GF code might still be able to correct them, as the errors might be dispatched over several of its shorter input data blocks. Consequently, depending on the requirements, a small Galois Field might be selected, as it improves both computational cost and performance.

C. Recommandations for RS Code Parameters Selection

To sum it up, some general trends can be drawn from the previous analysis. On one hand, computational complexity is reduced by choosing a low t and by increasing R. On the other hand, for performance optimization, a high code rate R is preferable to reduce the shift of the curve, while a high t parameter will improve its slope. However, this last effect has not a very

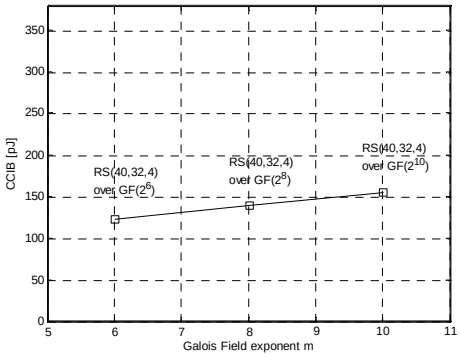

Figure 8. Consumption of RS codes over different Galois Fields

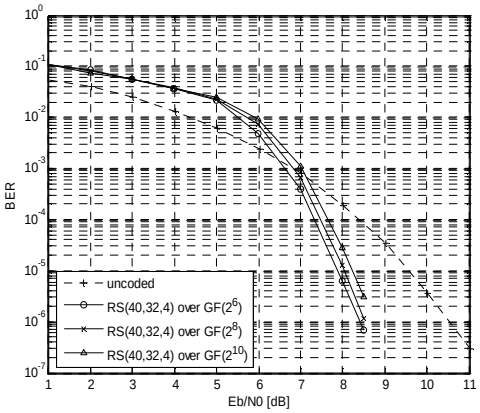

Figure 9. Performance of RS codes over different Galois Fields

significant impact for high values of t/k, and thus a limited value should be chosen for t, improving the computational complexity at the same time. Concerning the Galois Field size, a small value for m improves both computational complexity and performance. However, a too small GF size would limit the data block size (bounded by 2^m) and thus the code rate, which is not in line with the previous suggestions.

From these recommendations, a trade-off still needs to be determined for the t parameter. In the following, the computational complexity values and coding gain results previously obtained are compared in order to rule on an optimum value for t. To compare the CCIB and the coding gain G on a fair basis, the transmit energy per bit reduction ΔE_b associated with the coding gain can be extracted from:

$$G[dB] = -10.\log\left[1 + \frac{\Delta E_b}{E_b}\right] \quad (2)$$

where E_b is the transmit energy per bit used for the uncoded system. Replacing ΔE_b by ΔE_b+CCIB in this formula, the "equivalent gain" corresponding to the overall energy savings achieved with the coded system can thus be obtained as:

$$\Delta G[dB] = G - 10.\log\left[1 + \frac{CCIB}{10^{-G/10}.E_b}\right] \quad (3)$$

Figure 10 illustrates this relation for several RS code over GF(2^8) and GF(2^6). With respect to the previous advices, the highest value has been selected for n in order to maximize the coding rate. Values leading to optimum

or sub-optimum performances, as explained in subsection B, have been selected for t. A BER of 10^{-5} is considered to evaluate the coding gain G.

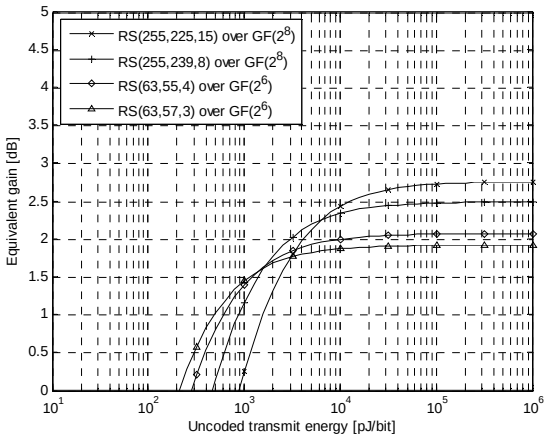

Figure 10. Equivalent gain of several low-power RS codes

This graph shows that the RS code selection depends on the transmit power of the system considered. For high power values, the computational complexity is negligible and the t parameter should be selected so as to maximize the coding gain. On the contrary, when considering lower power values, the computational complexity has a stronger influence, and a lower value for t should be selected in order to limit this effect. Therefore, depending on the system specifications, RS(255,225,15) over GF(2^8) or RS(63,57,3) over GF(2^6) could be advised. For very low power systems, no coding at all should be used.

It shall be reminded that this comparison is based on the power consumption values and the simplified channel scheme previously used. However, assuming that the use of a different technology would just introduce a scaling factor in the CCIB, the resulting effect on the previous graph would simply be a shift of the curves (see equation 3). Their relative position and the conclusions previously drawn can therefore be extrapolated to other systems.

D. Example of a Typical WPAN Transceiver

For illustration purpose, a Zigbee [8] transceiver is considered, which is now rather typical for WPAN applications. Amongst several existing implementations, the "Letibee" device shows the lowest output power of -3dBm [9]. Assuming a bit rate of 250 kbps, the corresponding uncoded transmit energy Eb=2nJ/bit is obtained. With these specifications, the RS(255,239,8) code over GF(2^8) is the most attractive solution to achieve some power savings (figure 10).

To go further with this example, the code parameters selection can be even more focused to the WPAN application specified. Particularly, the data block size k can be cross-optimized with the average transmitted message length. Indeed, when considering higher layers, the frames to be transmitted rarely fit exactly in a multiple of RS data blocks. Some padding has to be performed which increases the consumption uselessly. With the previous Zigbee example, let's assume that data frames always have the maximum 128 bytes length, and that one third of the communications are acknowledged, using 5 bytes acknowledgement frames. Command and beacon frames are ignored. The average CCIB can be computed as:

$$CCIB_{avg} = \sum_{L} p_L . CCIB_L \tag{4}$$

where p_L represents the probability of a message of length L, and $CCIB_L$, the actual CCIB associated with it:

$$CCIB_L = \frac{1}{L}.\left(CCIB.k.m.\left\lceil \frac{L}{k.m} \right\rceil \right) \tag{5}$$

With this adjustment, the equivalent gain is submitted to some modifications, as depicted in figure 11. Compared to figure 10, the curves are shifted by a different value for each code, depending on k, m and L.

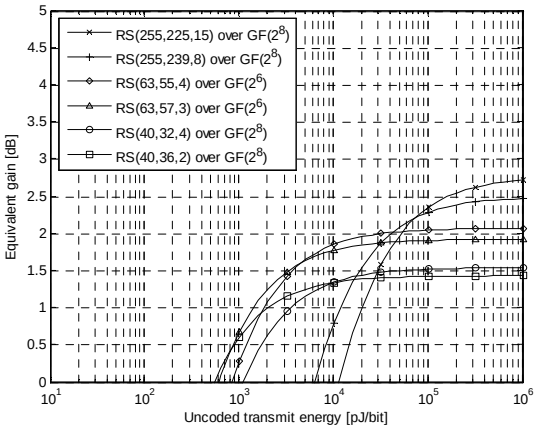

Figure 11. Equivalent gain with adjustments due to padding operations

RS codes with a large block size, like RS(255,239,8) over GF(2^8) previously mentioned, present a significant loss because the block size is not adapted to the frames to transmit. A smaller block size like for example RS(40,32,4) and RS(40,36,2) over GF(2^8) is more appropriate as the padding will be limited. However, with a comparable block size and computational complexity, RS(63,57,3) and RS(63,55,4) over GF(2^6) present the best choices, because of their higher coding gain. This example shows the difficulty to select a generic RS code, suitable for every low power application. However, a small block size will generally limit the computational waste due to padding.

Finally, it should be stressed that the general conclusions of this paper are quite in accordance with the choice of the IEEE 802.15.4a standardization committee for low rate WPANs [10], which opted for a RS(63,55,4) code over GF(2^6).

III. A Low Power Implementation of the Shortened RS(40,32,4) Code over GF(2^8)

As a case study, the RS(40,32,4) code over GF(2^8) is selected to quantify the worth of some low-power implementation improvements. Several approaches are successively considered, delay line mapping, switch-off and parallelization strategies for Syndrome Calculation and Chien Search units, and finally Composite Galois Field operations.

TABLE III. POWER CONSUMPTION OF THE RS DECODER WITH INCREMENTAL DESIGN IMPROVEMENTS.

Config.1: no error; Config.2: four errors in the first quarter of the message; Config.3: four errors in the three first quarters; Config.4: four errors until the end of the message (3 errors in the last quarter).

	LE power consumption (mW)				RAM power consumption (mW)	Hardware complexity	Max. clock frequency
	Config.1	Config.2	Config.3	Config.4			
Preliminary design	0.48	1.58	1.59	1.47	4.24	1577 LE 344 bits RAM	86.5 MHz
+ Delay line optimization	0.50	1.60	1.61	1.49	0.95	1550 LE 288 bits RAM	92 MHz
+ Syndrome optimization (original version)	0.48	1.72	1.73	1.61	0.95	1575 LE 576 bits RAM	83 MHz
+ Syndrome optimization (hybrid version)	0.45	1.63	1.64	1.52	0.95	1567 LE 288 bits RAM	83 MHz
+ Chien Search optimization	0.45	1.45	1.59	1.52	0.95	1569 LE 288 bits RAM	83 MHz
+ Parallelization optimization	0.66	1.77	2.02	2.00	0.69	1961 LE 288 bits RAM	87 MHz
+ GF operators optimization	0.47	1.31	1.46	1.45	0.95	1382 LE 288 bits RAM	83 MHz

The power consumption figures have been obtained with the Altera PowerPlay Analyser tool (QuartusII v5.0), after place and route of the design onto a Cyclone FPGA (EP1C6 speed grade 6). Results correspond to the decoding process of one data block (n.m bits). With the 50 MHz clock used, the computation time of the decoder is 13.4 µs. The main interest here is not the absolute power consumption values, but the relative gain obtained with each hardware optimization, which can be more easily extrapolated to other technologies.

A. Delay Line Mapping

The decoder block diagram is depicted in figure 1. This architecture has been thoroughly described in the literature [3]. The Syndrome Calculation unit generates a set of syndromes and acts as an error detector. In the next step, the syndromes are input to the Extended Euclidean Algorithm (EEA) to compute the Error Locator and Error Magnitude Polynomials (ELP and EMP). The degree of the ELP indicates the number of corrupted symbols detected (if lower than t), and its roots determine their location in the received data block. An exhaustive search of the roots of the ELP is then performed in the Chien Search unit, in order to find the position of the errors. The Chien Search is also applied to the EMP, and the resulting values are used by Forney's algorithm to compute the error magnitudes. At last, these magnitudes are added to the corresponding altered data to recover the initial message. A delay line is of course required to synchronize the error magnitudes with the corresponding received data. Additionally, a failure indicator unit is implemented, which indicates if the received data block can be corrected or not. It simply compares the number of error locations found in the Chien Search unit with the degree of the ELP. If different, a decoder failure is asserted. Finally, the GFinv operator is shared between EEA and Forney units, because its implementation results in a rather high hardware complexity, when using a straightforward 256-entry look-up table. Both modules never request it simultaneously with the pipelined architecture used.

In a preliminary design, the delay line has been mapped to the RAM memory blocks provided on the FPGA. All other modules have been mapped to Logic Elements (LE), which correspond to the programmable registered and combinational logic available on the FPGA. A first power consumption analysis shows the large consumption of RAM blocks compared to LEs (see table III, line 1). This remark is also valid regardless of the technologies and targets, and should lead the first design efforts towards an optimization of the delay line mapping. With the FPGA implementation described, a substantial 78% gain in RAM consumption has been obtained by designing precise memory read enable signal and by avoiding storing parity symbols (table III, line 2). Obviously, these savings are highly dependent of the technology used, and the results are provided just as an example.

B. Switch-Off Strategies

In a next step, some switch-off strategies have been applied to avoid some unnecessary computations. In the preliminary design, some simple techniques have already been used to reduce signal activity. First, each unit is disabled when unused in the pipeline scheme. In addition, Forney's algorithm and the error correction unit are disabled as well when parity symbols are processed, as they do not need any correction. The Chien Search unit, however, is still required during parity symbols computations, in order to drive the failure indicator unit until the end of the message. Finally, the GFinv unit, which introduce an important switching activity with its look-up table implementation, is only triggered when an error is detected during the Chien Search, and of course during the EEA.

Additionally, some less trivial power savings can be achieved by inhibiting some computations in the Syndrome Calculation unit [11] and in the Chien Search unit [12].

Concerning the Syndrome Calculation unit, only one half of the syndromes needs to be computed to detect any errors. In case of nullity, no error has been detected and the second half does not need to be computed, as well as

the rest of the decoder. The received data can be provided unchanged at the output. From table III (line 2), it can be noticed that the careful design of the preliminary version lead to a behavior similar to the intended objective, as the power consumption is reduced when an uncorrupted message is processed. In fact, this is also due to the inherent lower signal activity in the different modules resulting from the nullity of the syndromes.

Nevertheless, the described switch-off approach has been included in the design in order to evaluate the additional gain. Actually, the pipeline scheme is slightly modified, and the latency of the decoder is doubled. A memory twice as large is also required. For fair comparison, the power consumption results of this design have been extrapolated to fit to the same simulation duration as for the other designs. But this solution does not lead to significant power savings (see table III, line 3). In fact, the switch-off results in a 0.14 mW power reduction in the main computation units of the decoder, but the control overhead due to the highest latency leads to an extra 0.12 mW power consumption. This almost counterbalances the benefit of the optimization, and explains the higher consumption of the design when errors are received.

Therefore, a hybrid solution has been developed, where the switch-off is not used in the Syndrome Calculation unit itself, but is applied to all other elements of the decoder. In this case, latency is not increased, and a small 0.05 mW (10%) LE power reduction is achieved (Table III, line 4). However, the syndrome unit itself has a small 0.03 mW worse power consumption than the initial design, due to the additional test logic. This increases slightly the power consumption when errors are received. In summary, this optimization shows its benefit for high quality channels, for which most of the data blocks do not contain any errors. For corrupted data blocks, the power overhead is however negligible compared to the global decoding computation amount.

Concerning the Chien Search unit, it is not powered down in the preliminary design, in order to detect errors even on parity symbols. In fact, it can be powered down once all error locations have been found [12]. Furthermore, this feature requires no extra logic, since the failure indicator unit can already provide this information. With this improvement, power consumption depends of course on the position of the last error. Results from table III are therefore presented for different configuration of errors. This modification is combined with the previous syndrome optimization (table III, line 5). In the best case, when errors are gathered at the beginning of the message (config. 2), savings exceed 10% of the LE power consumption. They reach 75% for the Chien Search itself. Besides, it can be mentioned that errors which affect parity symbols are not corrected by Forney unit, saving some costly GF inversions. This reduces of course the consumption in config. 4, where 3 errors among 4 affect parity symbols. Of course, this switch-off optimization, like for syndrome switch-off shows its advantage for high quality channels, when errors are occasional. With a poor quality channel, errors are more likely to occur, and Chien Search will often be turned off late, leading to a negligible gain.

C. Hardware Parallelization

In addition to the previous modifications, some parallelism might be introduced in the decoder to reduce the number of register and memory accesses. Initially, this approach has been used rather to reach high data rates for optical transmissions than to reduce power consumption [11][13]. With a parallel-input RS decoder, the architecture of the Syndrome Calculation and Chien Search unit are deeply modified [11], as illustrated in figure 12 for a basic cell of the Syndrome Calculation unit. Instead of a sequential computation, symbols are input in parallel and the result is finally computed in a lower number of steps. This technique requires more logic, because operators are duplicated for each parallel input. But the same number of computations is performed in both cases, except for the number of register accesses which depends on the level of parallelism. Therefore, by tuning the level of parallelism, a trade-off between hardware complexity and power consumption is achieved.

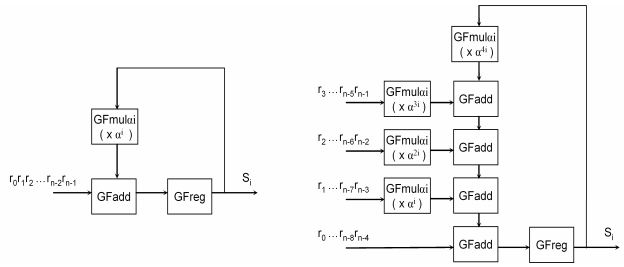

Figure 12. Parallelization in the Syndrome Calculation unit

With the 4-input architecture implemented, the power analysis shows first a notable 27% RAM power reduction (table III, line 6). Indeed, although the memory entries are 4 times wider, the number of memory accesses is divided by 4, which globally leads to a smaller power consumption. On the other hand, contrary to the expectations, the LE power consumption is increased. This is due to serialization and de-serialization processes, which are taken into account in this example, but this solution requires also more control logic, and finally less optimizations are achieved by the synthesis tool. As a result, compared to the previous version of the design, the overall power consumption is reduced by less than 5% (0.05 mW power reduction) with an uncorrupted message, whereas it can increase by up to 9% (0.22 mW power increase) when corrections are required. This modification can naturally be used for high quality channels, but considering the 25% extra hardware required, it will not be taken into account in the remaining analysis.

C. Composite Galois Field Approach

At last, a specific effort can be made to enhance the design of GF operators, and especially the power consuming multipliers and the inverter. The choice of a Composite Galois Field (CGF) approach [14]

theoretically achieves a good computational complexity reduction, especially for high order fields. Instead of computing operations over GF(2^8), an isomorphic Composite Galois Field GF((2^4)^2) is used. An operation in this field decomposes in a number of simpler operations in its ground field GF(2^4) of smaller order, as depicted in figure 13, and results in a global complexity reduction [15].

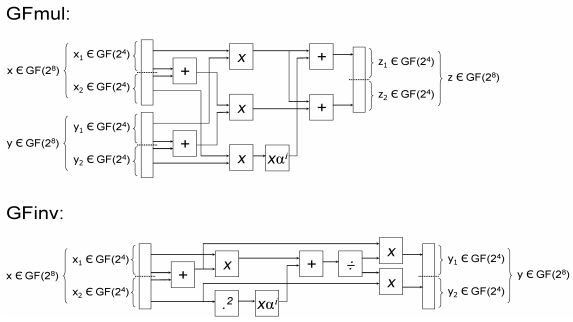

Figure 13. GF((2^4)^2) operators implementations

With this modification combined with the previous optimizations, our detailed reports revealed in fact that GFmulαi multipliers have a higher consumption than the traditional bit-parallel multipliers. It appears that the synthesis tool is not able to perform as much logic simplifications as with the previous design. This might be due to the sequential nature of their architecture, which prevents some advanced resource sharing between the different GFmulαi instantiated in the design, and which involves more logic layers for each operator. This explains the small power consumption increase observed for config. 1 (see table III, line 7), where the computations only involve the Syndrome Calculation unit and its GFmulαi operators. The same conclusion applies to GFmul operators, mainly used in EEA unit. But in this case, the sequential architecture of the operators allows to reduce some important signal fan-outs, and apparently seriously contributes to power savings in this unit.

Concerning GFinv, the new analytical method is far more efficient. A further enhancement has even been achieved in this module by keeping a classical look-up table architecture for the ground field inverter, instead of using direct formulas as suggested in [15]. Indeed, for a small ground field like GF(2^4), it is more interesting to keep a 16-entry look-up table which is severely optimised by the synthesis tool, for a hardware implementation. This slight change reduces the power consumption of the GFinv operator by 15%. These remarks explain that despite the higher GFmulαi consumption, an overall 10% LE power reduction is observed with the use of CGF, compared with the design including switch-off optimizations (table III, line 5 and 7). The exception is config. 4, where power savings are smaller (5%), because GFinv is not so much involved in the computations as above. From this analysis, it can be expected that the CGF approach might show a higher interest for very large Galois Fields, unfortunately not considered for low power RS codes.

D. Discussion

The whole study shows that for our specific FPGA implementation of the RS(40,32,4) over GF(2^8), the different switch-off and Composite Galois Field features, in conjunction with a careful mapping of the delay line, lead to a global 61% power reduction. The enhanced design also shows a slightly smaller hardware complexity, thanks to the CGF operators. However, apart from the delay line, which optimization highly depends on the target used, the respective 18% and 6% LE power reduction achieved in this study for corrupted and uncorrupted messages might be quite representative of the savings that can be expected by applying the same techniques to a different target.

Finally, the power consumption of the optimized hardware implementation can be linked to the results obtained in section II. As a first remark, it should be mentioned that in the theoretical analysis from section II, the whole decoding computations were taken into account in the power consumption estimations of RS codes. With the real hardware implementation, however, the power consumption is particularly reduced when receiving an uncorrupted message. As mentioned earlier, this is mainly due to the reduced signal activity in the different modules when the syndromes equal zero. Therefore, when considering low BER, even the preliminary version of the design will show a surprisingly low power consumption compared to the estimations from section II.

As an example, let's consider a coded BER of 10^{-5}. The corresponding uncoded BER is equal to $2\ 10^{-4}$. In a first approximation, the probability of an error in an input data block of the RS decoder is thus $6.4\ 10^{-2}$. The average consumption of the FEC system can be estimated, using the values from table III. The 80 pJ/info bit obtained for the preliminary design is far below the estimated consumption of 138 pJ/info bit from section II. This point highlights the importance of a careful design including some disable control logic. Considering the enhanced design, the hardware improvements described lead to a small additional 8% LE power reduction in average. The resulting gain curve is compared in figure 14 with the estimated curves from section II.

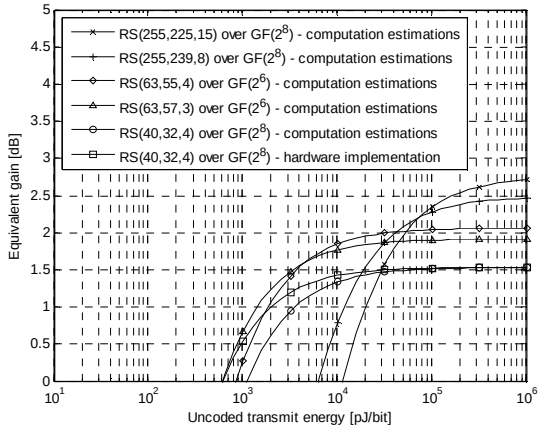

Figure 14. Equivalent gain with adjustments due to padding

This example clearly shows that a careful design of RS codes, similarly to the switch-off strategies, have an

appreciable impact on power consumption. More specifically, it shows the advantage of a FEC system which includes an error detector, and can be partly powered down when an uncorrupted message is received. This characteristic shows its importance when aiming communication systems operating at very low power ranges.

IV. Conclusion

This article describes methods to choose and design Reed-Solomon codes for low power transceivers. It highlights the impact of code parameters (n,k,t) as well as Galois Field size on the computational complexity and performance of the code. It shows that low power codes can be obtained with a limited error correction capability t combined with a high (n,k) pair. However, for an optimal code selection, some application-specific factors should be examined. The transmit power range aimed, and an approximate power consumption model of the technological target used, sets a kind of operating point which determines a variable power efficiency for each RS code. Furthermore, the average transmitted message length might also be analysed, in order to optimize the (n,k) parameters. A shortened code or a small Galois Field size might thus be worthwhile to avoid useless computations on padding bits.

Low power architecture improvements are investigated and illustrated with an FPGA implementation of the shortened RS(40,32,4) code over GF(2^8). The significant impact of some simple design issues is emphasized with the optimization of the delay line mapping, which represents the largest part of the overall power consumption for this FPGA implementation. Besides, further advanced considerations like switch-off strategies and Composite Galois Field approach resulted in up to 18% extra logic power reduction. On the contrary, the parallelization techniques did not give satisfying results. Above all, this hardware implementation showed the benefits of FEC systems which can be partly powered down when an uncorrupted message is received. Indeed, when targeting low BER, most of the time only the error detection unit is activated, leading to important power savings.

Acknowledgment

This work has been done in the frame of the **MAGNET Beyond** European ICT project of the 6th Framework Program. MAGNET Beyond is a R&D project within Mobile and Wireless Systems and Platforms Beyond 3G (www.ist-magnet.org).

References

[1] C. Desset, "Selection of channel coding for low-power wireless systems", *Vehicular Tech. Conf.*, vol.3, pp.1920-1924, Apr. 2003.

[2] L. Biard, D. Noguet, "Choice and Implementation of a Reed-Solomon Code for Low Power Low Data Rate Communication Systems", *IEEE Radio and Wireless Symp.*, pp.365-368, Jan. 2007.

[3] S. B. Wicker, *Error control systems for digital communication and storage*, Prentice Hall, 1995.

[4] S. Lin, D. J. Costello Jr., *Error Control Coding, Second Edition*, Prentice Hall, 2004.

[5] S. Choomchuay, B. Arambepola, "Time domain algorithms and architectures for Reed-Solomon decoding", *IEE proc. Communications Speech and Vision*, vol.140, no.3, pp.189-196, Jun. 1993.

[6] H. Lee, M.-L. Yu, L. Song, "VLSI design of Reed-Solomon decoder architectures", *IEEE Int. Symp. Circuits and Systems*, vol.5, pp.705-708, May 2000.

[7] S.-F. Wang, H.-Y. Hsu, A.-Y. Wu, "A very low-cost multi-mode Reed-Solomon decoder based on Peterson-Gorenstein-Zierler algorithm", *IEEE Workshop Signal. Processing Systems*, pp. 37-48, Sept. 2001.

[8] IEEE 802.15.4 standard, http://standards.ieee.org/getieee802/download/802.15.4-2003.pdf

[9] E. Mercier, F. Dehmas, "Principles of Communications in Wireless SensorNets – The Physical Layer", *Handbook of Wireless Mesh & Sensors Networking*, MacGraw Hill, 2007.

[10] IEEE 802.15.4, Draft P802.15.4a/D4, Aug. 2006.

[11] H.-C. Chang, C.-C. Lin, C.-Y. Lee, "A low-power Reed-Solomon decoder for STM-16 optical communications", *IEEE Asia-Pacific Conf. on ASIC*, pp.351-354, Aug. 2002.

[12] Y. Wu, "Low Power Decoding of BCH Codes", *IEEE Int. Symp. Circuits and Systems*, vol.2, pp.369-372, May 2004.

[13] L. Song, "10- and 40- Gb/s Forward Error Correction Devices for Optical Communications", *IEEE journal of solid-state circuits*, vol.37, no11, pp.1565-1573, Nov. 2002.

[14] C. Paar, *Efficient VLSI Architectures for Bit-Parallel Computation in Galois Fields*, PhD thesis, Inst. Experimental Math., U. of Essen, Germany, Jun. 1994.

[15] C. Paar, M. Rosner, "Comparison of Arithmetic Architectures for Reed-Solomon Decoders in Reconfigurable Hardware", *IEEE Symp. FPGAs for Custom Computing Machines*, pp.219-225, Apr. 1997.

Lionel Biard was born in France, in 1978. He received his M.S. degree from the National Graduate Engineering School in Electronics from Grenoble, in 2002. He pursued his studies in the telecommunication department from the CEA-LETI (Laboratory of Electronics, Technology and Information), in Grenoble, France, where he achieved a post-graduate French research diploma in 2005. He is now working in the area of digital architectures for wireless communications, at the LETI.

Dominique Noguet (MSc 1994, PhD 1998) is currently the head of the 'Digital Architectures and Prototypes' laboratory (LASP) at the Laboratory of Electronics, Technology and Information (LETI) of the French nuclear agency (CEA), where he also acts as senior expert on architecture design for signal processing and communication systems. He has been involved in digital architecture and system design for image processing, then for digital communication including CDMA and OFDM systems. He authored or co authored more than 25 papers in international journals and conferences. He was a member of the Technical Program Committee of various conferences including ISSSTA, PIMRC and EUROMICRO. He was awarded by a best paper award and the best PhD price from Institut National Polytechnique de Grenoble. His current field of interest is on flexible digital radio design.