Quaternary Reed-Muller Expansions of Mixed Radix Arguments in Cryptographic Circuits

Ashur Rafiev, Julian P. Murphy, Alex Yakovlev
School of Electrical, Electronic & Computer Engineering, Newcastle University
{ashur.rafiev, j.p.murphy, alex.yakovlev}@ncl.ac.uk

Abstract

Circuits built using multi-valued fixed polarity Reed-Muller expansions based on Galois field arithmetic, in particular quaternary expansions over GF(4), normally display high efficiency in terms of power consumption, area, etc. However, security application specific gate level mapping shows inefficient results for uniform radix expansions. The idea of the research here is to consolidate binary and quaternary Galois field arithmetic within a single circuit in such a way that the mathematical representations can benefit down to the gate level model. A direct method to compute quaternary fixed polarity Reed-Muller expansions of mixed radix arguments is proposed and implemented in a synthesis tool. The results for the various types of power-balanced signal encoding catered for the security application are compared and analysed.

1. Introduction

The research presented in this article attempts at finding optimisation techniques for cryptographic logic synthesis where the key qualities are power, area and security metrics. Security is considered in the scope of side-channel attacks, e.g. differential power analysis [1]. Data independent (balanced) switching of wires improves the protection against differential power analysis attacks [2, 3] and can be achieved using switching-balanced data encoding, e.g. m-of-n.

M-of-n codes are an encoding scheme in which data is represented using n wires and where m of them are set to an active level (usually high). A protocol separating data using dummy symbols (spacers) is called a *spacer protocol*. Circuits based on m-of-n codes, typically 1-of-4 or 1-of-2 (dual-rail), over the years have been used in a number of areas of electronics, in particular clockless circuits and networks-on-chip [4].

M-of-n codes other than dual-rail imply multi-valued logic (MVL) synthesis. Unfortunately the conventional EDA flow considers neither MVL synthesis nor encoding of data signals, hence it is not directly applicable for the security aware design. From this point of view the use of enhanced synthesis techniques is definitely more desirable, in particular the use of logic synthesis based on Galois field arithmetic which is natural for cryptography.

Computation of the quaternary Reed-Muller expansions over Galois fields of radix 4 has a long research history [5, 6, 7, 8, 9, 10]. These expansions are popular due to the efficiency of their hardware implementations and testability. These expansions have a form of the sum of products in Galois field arithmetic. A computation algorithm gives the expansion in a form of mathematical equation. The next task is to efficiently decompose it into the hardware components.

The efficient mapping from mathematical equations into a gate level netlist becomes a significant problem since concrete gate level implementations of Galois field arithmetic components in different radices, encodings and trade-offs between balancing and power costs have different merits and demerits as discussed in Section 5. For example, efficient for data transfer multi-valued signals may introduce considerable overhead in the corresponding logic implementation. Hence it appears impossible to find a globally optimal choice for the radix with respect to security context.

The known solution to the problem is to combine arithmetic over GF(2) and GF(4) within a scope of one expansion to uncover an area for further optimisations. Thus the advantages of different components may be consolidated within a single circuit, and the optimisation can be based on considering the real power and area costs of the components. Previous research in Reed-Muller expansions tends to optimise the computation time while the area of mixed radix Galois field arithmetic has been barely explored [11, 12]. Most of mixed radix related works were dedicated to the radix reconversion approach. The main goal of the research presented in this article is a deeper investigation of mixing radices in Reed-Muller expansions

0195-623X/09 $25.00 © 2009 IEEE
DOI 10.1109/ISMVL.2009.21

+	0	1	A	B
0	0	1	A	B
1	1	0	B	A
A	A	B	0	1
B	B	A	1	0

×	0	1	A	B
0	0	0	0	0
1	0	1	A	B
A	0	A	B	1
B	0	B	1	A

Figure 1. Addition and multiplication over GF(4)

and analysis of possible benefits in terms of security application.

The main aspects of the article can be listed as follows:

1. Mixed radix optimisations within one expansion. One of the key benefits of the proposed mixed radix approach is that uniform mathematical representation of values allows avoiding the use of additional signal conversion logic between radices thus optimising the number of operations performed.

2. Gate mapping optimisations in various balanced encodings for security purposes. The idea is to use the flexibility of mixed radix approach to optimise across a number of the key parameters, hence the resultant circuits will derive less switching activity from the quaternary components and less area from the binary ones.

This paper is organised as follows: Section 2 defines basic notions for fixed polarity Reed-Muller expansions and shortly describes Green's direct method to compute the quaternary expansions. Section 3 starts with the definition of quaternary expansions of binary arguments and then introduces a general case of mixed radix Reed-Muller expansions. Sections 4 and 5 are related with the gate level mapping of expansions. Section 6 presents synthesis results and compares the applied methods. The last section concludes the work and suggests areas for future work.

2. Basic notions

Galois field denoted as GF(p) is an algebraic structure consisting of a set of p elements and operations of addition and multiplication. This article covers binary and quaternary Galois fields, namely GF(2) and GF(4). In GF(2) the operation of addition refers to the binary XOR operation, and the operation of multiplication refers to the binary AND. Denoting elements of GF(4) as 0, 1, A, and B, addition and multiplication over GF(4) can be defined as shown in Figure 1. Extended description and properties of Galois fields can be found in [13].

Binary and multi-valued functions can be represented using XOR sum of products, in particular case Reed-Muller (RM) expansions.

Definition 1 Literal $\widetilde{x}$ *of the* p*-valued variable* x *is the one of* p *possible polarity forms* $(x + c)$*;* c *is an element of GF(p) denoting the literal. For binary case the literal forms of the variable* x *are* $x + 0 = x$*,* $x + 1 = \overline{x}$ *over GF(2). In quaternary case the literals of* x *are* $x + 0 = x$*,* $x + 1 = \dot{x}$*,* $x + A = \ddot{x}$*,* $x + B = \overline{x}$ *over GF(4).*

In a *fixed polarity* RM expansion each variable must be represented by the same literal throughout the expansion.

Definition 2 *For an* n*-variable* p*-valued function* $f(x_1, \ldots, x_n)$ polarity number k *is defined as the decimal equivalent* $\langle k \rangle_{10}$ *of the* p*-nary number* $\langle k_n \ldots k_1 \rangle_p$ *where single digit* k_i *denotes the literal* $\widetilde{x}_i$*. Thus for a single fixed polarity RM expansion* k *is a constant, and there exist* p^n *fixed polarity RM expansions for any* n*-variable* p*-valued function.*

Definition 3 *General canonical RM expansion for an* n*-variable* p*-valued function is defined as follows:*

$$f(x_1, \ldots, x_n) = \sum_{i=0}^{p^n-1} a_i \left[\prod_{j=1}^{n} \widetilde{x}_j^{i_j} \right] \quad \textit{over GF}(p) \qquad (1)$$

where i *is a decimal equivalent of a* p*-nary number* $\langle i_n \ldots i_1 \rangle_p$*. Vector* $a = \begin{bmatrix} a_0 & \ldots & a_{p^n-1} \end{bmatrix}^t$ *is a* coefficient vector.

For example, the Reed-Muller expansion of zero polarity for a quaternary function of one argument takes the form (2).

$$f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 \quad \text{over GF(4)} \qquad (2)$$

According to Green's direct method [7] of computation of quaternary fixed polarity RM expansions the coefficient vector can be calculated using the following equation:

$$a = W^n_{\langle k \rangle} \cdot d \qquad (3)$$

$$W^n_{\langle k \rangle} = W_{k_n} \otimes W_{k_{n-1}} \otimes \ldots \otimes W_{k_1}$$

where d is the truth vector of the function $f(x_1, \ldots x_n)$, $\otimes$ is a Kronecker matrix product. Matrices W_0, W_1, W_2, W_3 are defined. The computation of quaternary RM expansions of n-variable function corresponds directly to the computation of the matrices $W^n_{\langle k \rangle}$ for all polarity numbers $k = \{0, \ldots, 4^n - 1\}$. More efficient RM computation algorithms than direct method exist [5, 6, 8, 9, 10]. However this article is based on the direct method as it is clear for understanding the basics of fixed polarity RM expansions.

+	0	1	A	B
0	0	1	A	B
1	1	0	B	A

×	0	1	A	B
0	0	0	0	0
1	0	1	A	B

Figure 2. Addition and multiplication of mixed radix operands

3. Mixed radix Reed-Muller expansions

3.1. Quaternary expansions of binary arguments

Any 4-valued variable x_j can be represented in an isomorphic way with a pair of 2-valued variables $[y_{2j-1}, y_{2j}]$.

An intuitive solution to accommodate different radices within one circuit is to use signal conversion. In other words, the circuit can be split into parts employing different radix logic connected using the components adapting signals from one radix to another. In Galois field arithmetic this conversion can be expressed in a convenient mathematical representation. For example, $\mathrm{GF}\left(N^2\right) \rightarrow \mathrm{GF}\left(N\right)$ correspondence is typically implemented as $\mathrm{GF}^2\left(N\right) \rightarrow \mathrm{GF}\left(N\right)$ [12].

However, $\mathrm{GF}\left(N\right) \rightarrow \mathrm{GF}\left(N^2\right)$ correspondence is trivial since N-valued variables can always be assigned to M-valued variables if $N \leq M$. In our case all binary variables can be considered as quaternary constrained to the values 0 and 1. For example, let's assume that the function $g\left(x\right)$ is similar to $f\left(x\right)$ in (2) but its argument can be assigned only 0 or 1. Then $x = x^2 = x^3$ and the expansion takes the form:

$$g\left(x\right) = c_0 + c_1 x \quad \text{over GF(4)}$$

where $c_0 = a_0$, $c_1 = a_1 + a_2 + a_3$; $c_0, c_1 \in$ GF(4). Consequently, considering x as a binary variable, the operations of mixed radix operands can be defined as shown in Figure 2. Regardless of the binary radix of the argument the multiplication by quaternary constants will produce a quaternary result for the function $g\left(x\right)$ thus defining the notion of quaternary function of binary arguments or binary-to-quaternary ($b \rightarrow q$) function for simplicity.

Replacing the argument x in (2) with two 2-valued arguments y_1, y_2 the function $f\left(x\right)$ takes the form:

$$f_{b\rightarrow q}\left(\left[y_1, y_2\right]\right) = b_0 + b_1 y_1 + b_2 y_2 + b_3 y_1 y_2$$

Term $y_1 y_2$ can be calculated over GF(2) since the arguments are binary.

Similarly to (3) the coefficient vector $b = \left[\begin{array}{cccc} b_0 & b_1 & b_2 & b_3 \end{array}\right]^t$ can be calculated as follows:

$$b = Q^2_{\langle 0\rangle} \cdot d$$

$$Q^2_{\langle 0\rangle} = \left[\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{array}\right] = \left[\begin{array}{cc} 1 & 0 \\ 1 & 1 \end{array}\right] \otimes \left[\begin{array}{cc} 1 & 0 \\ 1 & 1 \end{array}\right] = Q_0 \otimes Q_0$$

Consequently in general case (1) for the binary-to-quaternary function $f_{b\rightarrow q}\left(y_1, \ldots, y_{2n}\right)$ takes the form:

$$f_{b\rightarrow q}\left(\left[y_1, y_2\right], \ldots, \left[y_{2n-1}, y_{2n}\right]\right) = \sum_{i=0}^{2^{2n}-1} b_i \left[\prod_{j=1}^{2n} \widetilde{y}_j^{i_j}\right]$$

where i is a decimal equivalent of a binary number $\langle i_{2n} \ldots i_1\rangle_2$. The product part of the expression can be calculated over GF(2), and the rest of the expression can be calculated over GF(4).

A direct method to compute the coefficient vector $b = \left[\begin{array}{ccc} b_0 & \ldots & b_{2^{2n}-1} \end{array}\right]^t$ is similar to Green's:

$$b = Q^{2n}_{\langle k\rangle} \cdot d$$

$$Q^{2n}_{\langle k\rangle} = Q_{k_{2n}} \otimes Q_{k_{2n-1}} \otimes \ldots \otimes Q_{k_1}$$

$$Q_0 = \left[\begin{array}{cc} 1 & 0 \\ 1 & 1 \end{array}\right],\ Q_1 = \left[\begin{array}{cc} 0 & 1 \\ 1 & 1 \end{array}\right]$$

where polarity number k refers to binary literals, i.e. $\langle k\rangle_{10} = \langle k_{2n}, k_{2n-1}, \ldots, k_1\rangle_2$.

One can see that these equations are similar to the binary Reed-Muller expansions with the only exception that the truth vector d is quaternary producing quaternary coefficient vector b. The computational cost of the direct method is $9^n - 4^n$ additions over GF(4) and no multiplications (multiplications over GF(2) are simple choice operations) versus $11^n - 4^n$ additions and $2\left(11^n - 5^n\right)/3$ multiplications over GF(4) for the quaternary RM expansions. Moreover, the optimisation techniques can be derived from the binary Reed-Muller expansion methodology, e.g. [14, 15].

Example 1 *For an arbitrary function F defined by its truth vector $d = [0B111BABAABA100A]^t$ pure quaternary expansions show the best polarity is $\langle 10\rangle_{10} = \langle AA\rangle_4$. The coefficient vector in this case is $a = [B0010BB0BAB0000A]^t$, and the quaternary RM expansion takes the form:*

$$\begin{aligned} F_{q\rightarrow q}\left(x_1, x_2\right) &= B + \ddot{x}_1^3 + B\ddot{x}_1\ddot{x}_2 + B\ddot{x}_1^2\ddot{x}_2 + B\ddot{x}_2^2 \\ &\quad + A\ddot{x}_1\ddot{x}_2^2 + B\ddot{x}_1^2\ddot{x}_2^2 + A\ddot{x}_1^3\ddot{x}_2^2 \end{aligned}$$

where $\ddot{x}_1 = x_1 + A$ and $\ddot{x}_2 = x_2 + A$ are A-polarity forms of the arguments x_1, x_2.

For the case of the quaternary function of binary arguments the best polarity is $\langle 5\rangle_{10} = \langle 0101\rangle_2$ *producing* $b = [0BAA00010A0A01A0]^t$*, and the mixed radix RM expansion takes the form:*

$$\begin{aligned} F_{b\to q}(y_1,\ldots,y_4) &= B + A\overline{y}_1 + A\overline{y}_1 y_2 + \overline{y}_1\overline{y}_3 + Ay_2\overline{y}_3 \\ &\quad + By_4 + B\overline{y}_1 y_4 + Ay_2 y_4 + A\overline{y}_3 y_4 \\ &\quad + A\overline{y}_1 y_2\overline{y}_3 y_4 \end{aligned}$$

3.2. Expansions of mixed radix arguments

Pure quaternary expansions and quaternary expansions of binary arguments are the extremes of more general quaternary expansions of mixed radix arguments ($b, q \to q$) allowing both binary and quaternary arguments within a single circuit. Formally mixed radix arguments form a vector $Z = \left[\begin{array}{ccc} z_1 & \ldots & z_n \end{array}\right]^t$ where z_i can be either the quaternary argument x_i of the original function or a binary pair $[y_{i0}, y_{i1}]$ representing x_i.

Definition 4 Argument radix number r *of a mixed radix RM expansion of* n*-variable quaternary function is a decimal representation of a binary tuple* $\langle r\rangle_{10} = \langle r_n \ldots r_1\rangle_2$ *where* r_i *is 0 if* $z_i = x_i$*, or 1 if* $z_i = [y_{i0}, y_{i1}]$*. For pure quaternary expansions* $r = 0$*; for quaternary expansions of all binary arguments* $r = 2^n - 1$*.*

Defining equivalences between quaternary literals and pairs of binary literals as $x_i \equiv [y_{i0}, y_{i1}]$, $\dot{x}_i \equiv [\overline{y}_{i0}, y_{i1}]$, $\ddot{x}_i \equiv [y_{i0}, \overline{y}_{i1}]$, $\overline{x}_i \equiv [\overline{y}_{i0}, \overline{y}_{i1}]$ we can transform the quaternary canonical form (1), $p = 4$, to the following:

$$f_{b,q\to q}(Z) = \sum_{i=0}^{4^n-1} e_i \left[\prod_{j=1}^{n} \widetilde{z}_j^{i_j}\right]$$

where $\widetilde{z}_j^{i_j} = \widetilde{x}_j^{i_j}$ for $r_j = 0$; $\widetilde{z}_j^0 = 1$, $\widetilde{z}_j^1 = \widetilde{y}_{j0}$, $\widetilde{z}_j^2 = \widetilde{y}_{j1}$, $\widetilde{z}_j^3 = \widetilde{y}_{j0}\widetilde{y}_{j1}$ for $r_j = 1$.

The direct method to compute the coefficient vector $e = \left[\begin{array}{ccc} e_0 & \ldots & e_{2^n-1} \end{array}\right]^t$ is applicable here in the form:

$$e = S^n_{\langle k\rangle} \cdot d$$

$$S^n_{\langle k\rangle} = S_{k_n} \otimes S_{k_{n-1}} \otimes \ldots \otimes S_{k_1}$$

$$S_{k_i} = \begin{cases} W_{k_i}, & r_i = 0 \\ Q^2_{\langle k_i\rangle}, & r_i = 1 \end{cases}$$

Exhaustive search through 2^n argument radix numbers and computing for each of them 4^n fixed polarity expansions is a task of a very high complexity. An efficient computation for mixed radix argument RM expansions is a subject for future research. This article considers RM expansions of fixed argument radices, either binary or quaternary.

4. Decomposition

For mapping the RM expansions to the gate level the target is to decompose the expressions into the operations of multiplication ($x \cdot y$), addition ($x + y$), multiplication by a constant (cx), and addition to of a constant ($x + c$) over GF(2) or GF(4), where x, y are 2-valued or 4-valued variables; c is a constant value. This section describes a number of presented optimisation techniques related to the decomposition.

Minimisation of terms The optimisation applied to the decomposition process is a minimisation of RM expansion. For the quaternary case a number of methods are proposed, e.g. [16, 17]. The minimisation problem may also refer to the factorisation. However, the factorisation is not applicable to the described mixed radix circuits since it changes the order of additions and multiplications overriding operation radices. In our tool we used a first-order minimisation algorithm which extracts repeating subterms and treats them as temporary variables.

As can be observed from the example in Section 3, binary arguments produce larger number of terms, but the same terms tend to appear more frequently than in the case of pure quaternary thus having a greater potential for minimisation.

Propagation of binary radix Since GF(4) arithmetic operations of binary arguments also produce binary results, the target is to choose such a polarity and group terms in such a way that GF(2) propagates as far as possible. According to the properties of GF(4), $x^3 = 1$ for any non-zero x thus clamping the result of this operation to the binary range. Consequently all cubic forms of the arguments in quaternary RM expansions can use multiplications over GF(2) instead of GF(4). Similarly if a binary term in binary-to-quaternary sum of products is not multiplied by A or B it can be used as a binary argument to the following addition.

This optimisation approach does not affect the circuit structure, and it reduces area but not the energy consumption because it attempts to remove *unused* paths from the circuit, i.e. paths which never switch due to the properties of the original function.

Search for the best expansion Typically the expansion with the least number of non-zero terms is chosen as the best one [7, 10]. This approach minimises the number of additions in the resultant circuit but does not consider the total number of additions and multiplications. The exact number of operations is known only after the decomposition. Taking into account the proper values for switching energy and area for these operations the synthesis tool can search for the optimal solution with respect to the gate level characteristics.

Table 1. Encoded quaternary values

value	single-rail	dual-rail	1-of-4
0	00	01 01	0001
1	01	01 10	0010
A	10	10 01	0100
B	11	10 10	1000
spacer (NULL)	–	00 00	0000

However, for the large circuits the execution time can be infeasible if we decompose expansions for all polarities. Therefore the decomposition should be performed for a smaller number of the best expansion candidates selected using a simple criterion, e.g. the number of non-zero terms, which still might be a rough estimation criterion.

5. Component implementations

Arithmetic components for GF(2) and GF(4) can be implemented in different ways with respect to the selected encoding for binary and quaternary signals. Single-rail is a typical binary representation of signals. However, the focus of the paper is switching balanced codes, in particular 1-of-2 (dual-rail) and 1-of-4. Dual-rail encodes binary values using 2 wires: 0 is encoded as 01, 1 as 10. 00 is a spacer value. Quaternary values can be encoded as shown in Table 1.

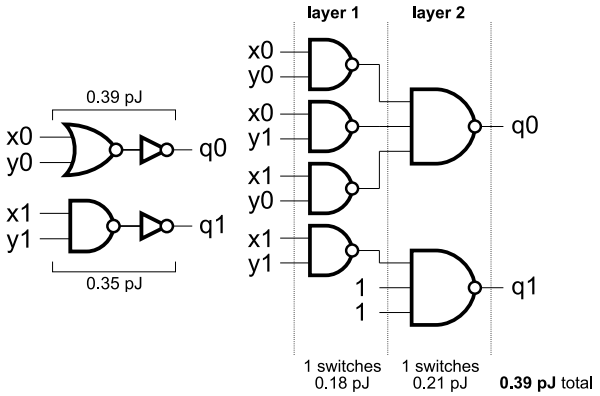
Generic approaches for m-of-n codes over Galois fields are patented in [11]. Since the primary attribute of m-of-n codes is a balanced switching, the components should also display this feature. Ideally the form and size of power signature of the component should be symmetric with respect to switching from a spacer to any data and vise versa. Usually this is made by introducing additional dummy-logic paths. However for real life examples an ideal symmetry is impossible, but the components can be "fully balanced" with respect to the technology capabilities.

For example, consider a GF(2) multiplier. In single-rail it can be represented with an ordinary AND gate while in dual-rail it takes the form:

$$\begin{aligned} q_0 &= x_0 + y_0 \\ q_1 &= x_1 y_1 \end{aligned} \tag{4}$$

where $\{q_0, q_1\}$ are wires of dual-rail encoded output, $\{x_0, x_1\}$ and $\{y_0, y_1\}$ are wires of dual-rail encoded operands x and y.

Mapping of the equation (4) into negative logic cells is shown in Figure 3(a). Switching the input $[x, y]$ from the spacer value to $[0, 0]$, $[0, 1]$ and $[1, 0]$ causes NOR gate to fire. Switching from the spacer to $[1, 1]$ fires NAND gate. NAND and NOR gates have different switching energy values thus the component balancing is not good in this case.



(a) relaxed balancing (b) fully balanced

Figure 3. Negative logic implementation of GF(2) multiplication in dual-rail

In order to balance it better we have to put additional logic paths making the structure of the component symmetric with respect to gates and input signals switching activity as shown in Figure 3(b). In the spacer state all inputs are set to low thus all outputs of 2-input NAND gates in the first layer are set to high precharging NAND gates in the second layer. Arrival of any data signal ($[0, 0]$, $[0, 1]$, $[1, 0]$, or $[1, 1]$) causes exactly one gate from the first layer to fire. This will produce only one 0 signal to the second layer switching one of the 3-input NANDs. Addition of constant inputs to certain gates guarantees that all gates in each layer are equal.

Although there are certain unavoidable aspects of the technology such as transistor level asymmetry which introduce little disbalance even to this design, an implementation is acceptable if it fits the requirements of the security standard [18]. For the same reason the structure shown in Figure 3(a) might also be sufficient since the difference in swithing energy is not large. This implies the approach of "relaxed" balancing when the security is slightly compromised for significant power and area gains.

For the exact implementations of other GF components the reader may refer to [19]. In our examples we used AMS C35 (0.35μm) library. Energy and area estimations of the components are shown in Table 2. These values are based on the RTL library specification.

6. Benchmark results

Approaches described in Sections 3.1, 4, and 5 are implemented in a tool which allows us to synthesise circuits using quaternary and binary-to-quaternary RM expansions. Component characteristics from Table 2 were used to find best polarities and to compute circuit characteristics. The precise evaluation requires placement and routing to be

Table 3. Synthesis results

logic balancing	expansion radix	AES S-box num of op-s	AES S-box switch energy pJ	AES S-box total area μm^2	Kasumi S7 num of op-s	Kasumi S7 switch energy pJ	Kasumi S7 total area μm^2	Misty S7 num of op-s	Misty S7 switch energy pJ	Misty S7 total area μm^2	Misty S9 num of op-s	Misty S9 switch energy pJ	Misty S9 total area μm^2
relaxed	binary	2229	839.28	**553826**	169	63.69	**41710**	167	62.88	**41494**	177	66.39	**45238**
	mixed	**1528**	**618.12**	1066396	**147**	**58.80**	89682	**155**	**61.95**	95830	**165**	**66.18**	110542
	q-ry	1640	661.83	1616841	417	168.00	414266	383	154.68	386018	533	215.55	541449
full	binary	2229	839.28	**779778**	169	63.69	**59190**	167	62.88	**58422**	177	66.39	**61614**
	mixed	**1528**	**618.12**	1210652	**147**	**58.80**	104402	**155**	**61.95**	110734	**165**	**66.18**	124894
	q-ry	1640	1026.99	2386231	417	267.96	627038	367	225.90	540268	533	331.89	789087

logic balancing	expansion radix	DES S-box 1 num of op-s	DES S-box 1 switch energy pJ	DES S-box 1 total area μm^2	DES S-box 2 num of op-s	DES S-box 2 switch energy pJ	DES S-box 2 total area μm^2	DES S-box 3 num of op-s	DES S-box 3 switch energy pJ	DES S-box 3 total area μm^2	DES S-box 4 num of op-s	DES S-box 4 switch energy pJ	DES S-box 4 total area μm^2
relaxed	binary	237	89.37	**58230**	173	65.07	**43326**	165	**62.34**	**39946**	226	85.14	**55932**
	mixed	186	75.42	135804	**133**	**53.85**	94298	**163**	65.61	101882	**176**	**71.31**	126550
	q-ry	**174**	**70.08**	167572	161	64.86	154912	168	67.77	163804	181	72.99	175218
full	binary	237	89.37	**83070**	173	65.07	**60438**	165	**62.34**	**57978**	226	85.14	**79116**
	mixed	186	**75.42**	152364	**133**	**53.85**	106626	**163**	65.61	119362	**176**	**71.31**	142558
	q-ry	**174**	109.92	251004	161	100.56	230282	168	104.94	241724	181	112.98	258798

Table 2. Switching energy and area for GF components

parameter	GF(2) dual-rail +	GF(2) dual-rail ×*	GF(2) dual-rail ×	GF(4) 1-of-4 +	GF(4) 1-of-4 ×*	GF(4) 1-of-4 ×
max sw. en., pJ	0.36	0.39	0.39	0.42	0.39	0.81
area, μm^2	330	182	366	1244	805	1699

* relaxed balancing

made, however it is a rather complex task. Currently we intend to use more generic evaluation.

Various S-boxes (DES, AES [20], Kasumi [21] and MISTY [22]) were chosen as typical examples of security circuits. They were synthesised in pure quaternary, pure binary, and binary-to-quaternary radix domains and mapped into fully balanced and relaxed components. The results are shown in Table 3. The switching energy parameter is a sum of switching energies of gates, and it does not consider the switching of wires. Since the encoding scheme restricts switching to one wire per data signal, the number of operations can be used to estimate the switching activity of intercomponent wires.

As can be observed from the examples, operations over GF(4) show considerable area overhead comparing to their GF(2) counterparts. The explanation can be as follows. The decomposition of quaternary operations to binary gates produces certain overhead while binary operations use the same radix domain as their gate level implementations. The quaternary domain logic might be used instead, for example n-valued dynamic logic [23], but this type of technology is not applicable for security.

In terms of power the results show variable efficiency for all radices. Kasumi and MISTY S-boxes are efficient in binary, DES S-Box 1 is good in quaternary. AES S-box shows the best results for mixed radix approach: the synthesised mixed radix circuit consume 26% less energy than the binary and occupy 34% less area than quaternary. This effect is related with the properties of the implemented function, and it appears impossible to analyse the efficiency of particular radix apriori, before the circuit is synthesised.

7. Conclusions

The method of generalising quaternary fixed polarity Reed-Muller expansions to the quaternary expansions of binary and mixed radix arguments is proposed. This type of expansions can be used to synthesise logic optimised with respect to exact parameter values of gate level components. Possible gate level mapping optimisation approaches, such as binary radix propagation, minimisation and implementation aware search of the best polarity, are also described.

The efficiency of the circuit depends on its function properties in relation to Galois field arithmetic. Benchmark re-

sults show improvement of up to 26% in switching energy and up to 84% in total area for mixed radix circuits over uniform radix, but in general the results may vary significantly. Nevertheless, binary-to-quaternary RM expansions are good for trade-off between hardware parameter costs and definitely should be considered as a possible synthesis technique.

To improve the runtime of the developed tool we need to apply more efficient algorithms for RM expansion computations and optimise the decomposition algorithm. An efficient methodology to compute the general case of mixed radix argument expansions is also a subject of future work.

Acknowledgement: This work is supported by EPSRC GR/F016786/1.

References

[1] P. Kocher, J. Jaffe, and B. Jun, "Introduction to differential power analysis and related attacks," 1998.

[2] A. Bystrov, D. Sokolov, A. Yakovlev, and A. Koelmans, "Balancing power signature in secure systems," in *Proc. 14th UK Asynchronous Forum*, 2003.

[3] S. Moore, R. Anderson, P. Cunningham, R. Mullins, and G. Taylor, "Improving smart card security using self-timed circuits," *Proc. of Asynchronous Circuits and Systems*, pp. 211–218, 2002.

[4] W. Bainbridge, W. Toms, D. Edwards, and S. Furber, "Delay-insensitive, point-to-point interconnect using m-of-n codes," in *Proc. of ASYNC'03*, 2003.

[5] B. Falkowski and S. Rahardja, "Efficient computation of quaternary fixed polarity Reed-Muller expansions," *Computers and Digital Techniques, IEE Proc.*, vol. 142, pp. 345–352, 1995.

[6] B. J. Falkowski and C. C. Lozano, "Quaternary fixed-polarity Reed-Muller expansion computation through operations on disjoint cubes and its comparison with other methods," *Computers & Electrical Engineering*, vol. 31, pp. 112–131, 2005.

[7] D. Green, "Reed-Muller expansions with fixed and mixed polarities over GF(4)," in *IEE Proc., Part E*, vol. 137, 1990.

[8] D. Jankovic and R. S. Stankovic, "Efficient calculation of fixed-polarity polynomial expressions for multiple-valued logic functions," in *Proc. of ISMVL '02*, p. 76, IEEE Comp. Soc., 2002.

[9] D. Jankovic, R. S. Stankovic, and C. Moraga, "Optimization of GF(4) expressions using the extended dual polarity property," in *Proc. of ISMVL '03*, p. 50, IEEE Comp. Soc., 2003.

[10] S. Rahardja and B. Falkowski, "Efficient algorithm to calculate Reed-Muller expansions over GF(4)," *Circuits, Devices and Systems, IEE Proc.*, vol. 148, pp. 289–295, 2001.

[11] UK Patent No. 0719455.8, "Cryptographic processing and processors." Newcastle University.

[12] Z. Zilic and Z. Vranesic, "Current-mode CMOS Galois field circuits," in *Proc. 23rd International Symp. on MVL*, pp. 245–250, 1993.

[13] T. C. Bartee and D. I. Schneider, *Computation with Finite Fields*, vol. 6 of *Inform. Contr.* June 1963.

[14] S. Purwar, "An efficient method of computing generalized Reed-Muller expansions from Binary Decision Diagram," *IEEE Trans. Comput.*, vol. 40, no. 11, pp. 1298–1301, 1991.

[15] E. C. Tan and H. Yang, "Optimization of fixed-polarity Reed-Muller circuits using dual-polarity property," *Circuits, systems, and signal processing*, vol. 19, no. 6, pp. 535–548, 2000.

[16] S. Yanushkevich, D. Popel, V. Shmerko, V. Cheushev, and R. Stankovic, "Information theoretic approach to minimization of polynomial expressions over GF(4)," in *Proc. of ISMVL '00*, p. 265, 2000.

[17] Y. Zhang and P. W. Rayner, "Minimisation of Reed-Muller polynomials with fixed polarity," *IEE Proc.*, vol. 131, pp. 177–186, 1984.

[18] "Federal information processing standards FIPS 140-3 (draft)." National Institute of Standards and Technology.

[19] A. Rafiev, J. Murphy, and A. Yakovlev, "RTL implementations of GF(2) and GF(4) arithmetic components," tech. rep., Newcastle University, 2008.

[20] *Specification for the Advanced Encryption Standard (AES)*, Nov 26, 2001. Federal Information Processing Standards Publication 197.

[21] *3GPP Technical Specification 35.202*, 2001. v3.1.1.

[22] M. Matsui, "Block encryption MISTY." Communications Science and Techniques, ISEC96-11, 1996.

[23] Intrinsity, Inc., *Technology White Papers*, ch. 8: N-ary Circuits: Robust Gate Design. www.intrinsity.com, 2006.