# Fixed Polarity Pascal Transforms with Symbolic Computer Algebra Applications

Kaitlin N. Smith and Mitchell A. Thornton
*Department of Electrical and Computer Engineering*
*Southern Methodist University*
Dallas, Texas, U.S.A.
{KNSmith, Mitch}@SMU.edu

***Abstract*—The fixed polarity forms of the Reed-Muller (RM) transform exist in $2^n$ different polarities. The integer-valued Pascal transform is related to the binary-valued RM transform through the Sierpinski fractal, calculated by performing the modulo-2 operation on Pascal's triangle, as it appears in the lower triangular portion of the positive-polarity RM transform. We generalize the relationship between the fixed-polarity forms of the RM transform and introduce associated forms of the Pascal transform that are characterized by a polarity value allowing for a family of fixed-polarity Pascal (FPP) transform matrices to be defined. We observe and prove several properties of the FPP transforms and their inverses. An application of FPP transforms in the area of symbolic computer algebra that enables very fast decomposition of real-valued polynomials as weighted sums of different binomials raised to a power as compared to manual symbolic manipulation is described. The decomposition weights can be considered to be the inverse FPP spectrum with respect to a real-valued polynomial since they are computed using one of the linear orthogonal FPP transformation matrices.**

## I. INTRODUCTION

The Reed-Muller (RM) transform is used in the information and switching theory communities for a variety of applications. When the RM transform is used to analyze a switching function that depends upon $n$ variables, there are $2^n$ different versions of the transformation matrix that are collectively known as the fixed-polarity transformation matrices of size $2^n \times 2^n$. The polarity number is an integer parameter $p$ where $p \in [0, 2^n - 1]$. The value $n$ represents the number of dependent variables of a particular switching function and $p$ specifies the exact RM transformation matrix. One of the common applications of the polarity-$p$, or fixed-polarity RM (FPRM) transform is to find the value of $p$ such that a polarity-$p$ transformation of a switching function results in an Exclusive-OR Sum of Products (ESOP) expression with a minimized number of product terms.

Sierpinski's gasket is a fractal structure that is closely related to Pascal's triangle as each component of Sierpinski's structure is the modulo-2 form of the corresponding integer value in Pascal's triangle [1], [2]. Furthermore, the positive-polarity RM transform (PPRM) is the form of the RM transform when the polarity value is zero ($p = 0$), resulting in a lower triangular transformation matrix where the lower triangular portion of the PPRM transform matrix is equal to Sierpinski's gasket. Thus, there is a close and inherent mathematical relationship between the PPRM transformation matrix and Pascal's triangle via the relationship both structures have with Sierpinski's gasket [3]. The Pascal transform is described in [4], [5] and has been applied to image processing. Additionally, the Pascal matrix has been found useful in filter design for $s$ to $z$ domain conversion [6]. Pascal transforms are of interest in the switching theory community as described in [7] [3]. Fast algorithms for Pascal and other related transforms have been considered recently [8] [9].

In this work, we describe a family of Pascal transforms characterized by an integer value that is analogous to the polarity number $p$ of the fixed-polarity RM (FPRM) transforms. We provide some background definitions, observations, and mathematical results that are then used to justify and formulate this family of Pascal-like transforms that we refer to as "fixed-polarity Pascal" (FPP) transforms. We also describe an application of FPP transforms in the field of symbolic computer algebra for factoring polynomials into a weighted sum of binomials raised to an integer power. Finally, we consider a further generalization of the inverse FPP transforms that results in a corresponding generalization of the binomial decomposition problem that enables any arbitrary set of binomials to be considered.

## II. BACKGROUND, PROPERTIES, AND NOTATION

### A. Pascal's Triangle

Pascal's triangle is a well-known structure that is infinite in size, comprised of integer values, and displays many interesting properties. Pascal's triangle can be constructed with the well-known recursive formula that yields values in the $i^{th}$ row based on the sum of two values in the preceding $(i-1)^{th}$ row. As an example, the first seven rows of Pascal's triangle are shown below.

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | 1 | | | | | | | $n=0$ |
| | | | | | 1 | | 1 | | | | | | $n=1$ |
| | | | | 1 | | 2 | | 1 | | | | | $n=2$ |
| | | | 1 | | 3 | | 3 | | 1 | | | | $n=3$ |
| | | 1 | | 4 | | 6 | | 4 | | 1 | | | $n=4$ |
| | 1 | | 5 | | 10 | | 10 | | 5 | | 1 | | $n=5$ |
| 1 | | 6 | | 15 | | 20 | | 15 | | 6 | | 1 | $n=6$ |
| $\cdot^{\cdot^{\cdot}}$ | | $\vdots$ | | $\vdots$ | | $\vdots$ | | $\vdots$ | | $\vdots$ | | $\vdots$ | $\ddots$ |

One of the properties that is of interest in this paper is the relationship of each row in the triangle to the well-known binomial theorem

$$(x+y)^n = \sum_{i=0}^{n} \left( \begin{array}{c} n \\ i \end{array} \right) x^i y^{n-i}. \quad (1)$$

Pascal's triangle is an explicit arrangement of the combinatorial coefficients in Eqn. 1. The $n^{\text{th}}$ row of the triangle contains elements beginning with coefficient of the $n^{\text{th}}$-degree monomial on the left to the $0^{\text{th}}$-degree monomial (or constant) on the right, or vice-versa.

We refer to the values in each row, the binomial coefficients, as $a_k$ where $k = 0, \cdots, n$. These values are also the coefficients of the simplified polynomial form of $(x+1)^n$. We use the term "simplified polynomial" to denote that the polynomial is comprised of only a single term for each degree value $n$. That is, each instance of the monomial, $a_k x^k$, for a specific $k$ appears only once in the simplified and expanded polynomial of $(x+1)^n$ as shown in Table I.

TABLE I
PASCAL'S TRIANGLE WITH CORRESPONDING BINOMIALS AND SIMPLIFIED POLYNOMIALS

| Binomial Power | Binomial | Polynomial |
|---|---|---|
| $n=0$ | $(x+1)^0$ | $(1)x^0$ |
| $n=1$ | $(x+1)^1$ | $(1)x^1 + (1)x^0$ |
| $n=2$ | $(x+1)^2$ | $(1)x^2 + (2)x^1 + (1)x^0$ |
| $n=3$ | $(x+1)^3$ | $(1)x^3 + (3)x^2 + (3)x^1 + (1)x^0$ |
| $n=4$ | $(x+1)^4$ | $(1)x^4 + (4)x^3 + (6)x^2 + (4)x^1 + (1)x^0$ |

A modified form of Pascal's triangle exists that corresponds to binomials of the form $(x-1)^n$. This well-known alternative of Pascal's triangle is identical to that given previously with the exception that each alternating value in the triangle is negated. The implementation of the modified Pascal's triangle is found in in Table II.

TABLE II
MODIFIED PASCAL'S TRIANGLE IN TERMS OF SIMPLIFIED BINOMIALS

| Binomial Power | Binomial | Polynomial |
|---|---|---|
| $n=0$ | $(x-1)^0$ | $(1)x^0$ |
| $n=1$ | $(x-1)^1$ | $(1)x^1 - (1)x^0$ |
| $n=2$ | $(x-1)^2$ | $(1)x^2 - (2)x^1 + (1)x^0$ |
| $n=3$ | $(x-1)^3$ | $(1)x^3 - (3)x^2 + (3)x^1 - (1)x^0$ |
| $n=4$ | $(x-1)^4$ | $(1)x^4 - (4)x^3 + (6)x^2 - (4)x^1 + (1)x^0$ |

### B. The Pascal Transform

The Pascal transform refers to a linear transformation matrix that contains components based upon the Pascal triangle [10], [11]. The Pascal's transform matrix can be expressed in a lower triangular, $\mathbf{P}_L$, or a upper triangular, $\mathbf{P}_U$, form. Pascal's matrices are infinite in dimension, however, it is convenient and common to utilize truncated versions of finite dimension based upon the degree of the polynomial of interest. The rows and columns of the Pascal matrices, and all matrices referred to in this paper, have indices that begin at zero (0) so that the index value values are equivalent to related to the binomial exponent $n$ in the triangle. In truncated form, the two different forms of the Pascal transformation matrices are denoted as a finite dimensioned $k \times k$ form as $\mathbf{P}_{Lk}$ and $\mathbf{P}_{Uk}$ where the indices are $i, j = 0, 1, \ldots, k-1$. Eqn. 2 contains the $4 \times 4$ truncated versions of the Pascal transformation matrices.

$$\mathbf{P}_{L4} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 \\ 1 & 3 & 3 & 1 \end{bmatrix} \quad \mathbf{P}_{U4} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

The construction definitions for the lower and upper triangular forms based on the binomial coefficients are provided in Eqn. 3 and Eqn. 4.

$$\begin{array}{ll} \mathbf{P}_L = [p_{L;i,j}] & \\ p_{L;i,i} = p_{L;i,0} = 1 \forall i = 0, \ldots, n-1, & p_{L;i,j} = 0 \text{ if } j > i \\ p_{L;i,j} = \left( \begin{array}{c} i \\ j \end{array} \right) & \forall i, j = 0, \ldots, n-1 \text{ and } i > j \end{array} \quad (3)$$

$$\begin{array}{ll} \mathbf{P}_U = [p_{U;i,j}] & \\ p_{U;i,i} = p_{U;0,i} = 1 \forall i = 0, \ldots, n-1, & p_{U;i,j} = 0 \text{ if } j < i \\ p_{U;i,j} = \left( \begin{array}{c} j \\ i \end{array} \right) & \forall i, j = 0, \ldots, n-1 \text{ and } i < j \end{array} \quad (4)$$

With respect to the binomial theorem, modified Pascal matrices can be formed with respect to the modified Pascal triangle that contain rows corresponding to simplified polynomial forms of $(x-1)^n$. These are provided in Eqn. 5.

$$\mathbf{P}_{ML4} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \quad \mathbf{P}_{MU4} = \begin{bmatrix} 1 & -1 & 1 & -1 \\ 0 & 1 & -2 & 3 \\ 0 & 0 & 1 & -3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

The modified Pascal matrices may also be defined using the binomial coefficients.

$$\begin{array}{l} \mathbf{P}_{ML} = [p_{ML;i,j}] \\ p_{ML;i,i} = p_{ML;i,0} = (-1)^{i+j} \forall i = 0, \ldots, n-1, p_{ML;i,j} = 0 \text{ if } i < j \\ p_{ML;i,j} = (-1)^{i+j} \left( \begin{array}{c} i \\ j \end{array} \right) \forall i, j = 1, \ldots, n-1 \text{ and } i > j \end{array} \quad (6)$$

$$\begin{array}{l} \mathbf{P}_{MU} = [p_{MU;i,j}] \\ p_{MU;i,i} = p_{MU;0,i} = (-1)^{i+j} \forall i = 0, \ldots, n-1, p_{MU;i,j} = 0 \text{ if } i > j \\ p_{MU;i,j} = (-1)^{i+j} \left( \begin{array}{c} j \\ i \end{array} \right) \forall i, j = 1, \ldots, n-1 \text{ and } i < j \end{array} \quad (7)$$

### C. Pascal Matrix Properties

The various Pascal and modified Pascal matrices have properties that are useful for the analyses in this work. Proofs have been omitted to keep this section concise. The proofs, however, can be provided by the authors if requested.

*Observation 1:* The Pascal transformation matrices $\mathbf{P}_L = [p_L; i, j]$ and $\mathbf{P}_U = [p_U; i, j]$ are "unitriangular" meaning they are triangular matrices with diagonal elements $p_L; i, i = p_U; i, i = 1$.

*Lemma 1:* The determinants of $\mathbf{P}_L$, $\mathbf{P}_U$, $\mathbf{P}_{ML}$, and $\mathbf{P}_{MU}$, are unity, +1.

□

*Lemma 2:* The determinants of $\mathbf{P}_S$ and $\mathbf{P}_{MS}$ are unity, +1.

□

*Lemma 3:* The Pascal matrices $\mathbf{P}_U$ and $\mathbf{P}_L$ are orthogonal.

□

*Observation 2:* From the constructive definitions, we observe the following with respect to transposes of the Pascal and modified Pascal transformation matrices.

$$\mathbf{P}_U = \mathbf{P}_L^T$$
$$\mathbf{P}_{MU} = \mathbf{P}_{ML}^T$$

□

Next, we address the issue of Pascal matrix inverses. We Note that since $\mathbf{P}_U = \mathbf{P}_L^T$, it is only necessary to consider one of the unitriangular forms of the matrices. We further adopt the convention of Knuth *et. al.* whereby we consider $x^0 = 1$ for all $x$ since this is a condition that must be true if the binomial theorem is to be valid when it is stated using variables $x$ and $y$ as $(x+y)^n$ for the cases where $x = y = 0$ and $x = -y$ [12] [13] [10].

*Theorem 1:* The two matrices $\mathbf{P}_U$ and $\mathbf{P}_{MU}$ are inverses of one another.

$$\begin{aligned}\mathbf{P}_U^{-1} &= \mathbf{P}_{MU}\\ \mathbf{P}_{MU}^{-1} &= \mathbf{P}_U\end{aligned} \tag{8}$$

□

*Corollary 1:* The two matrices $\mathbf{P}_L$ and $\mathbf{P}_{ML}$ are inverses of one another.

□

### D. The Fixed-polarity Reed-Muller Transforms

The RM transform was originally devised for use in error detection and correction and is also used for transforming binary switching functions in sum of products (SOP) form into a type of ESOP form [14] [15]. Spectral methods allow switching functions to be expressed as the sum of weighted set of basis functions, and the RM transform in particular produces the weighting coefficients $r_i \in \{0, 1\}$ resulting in a representation of the switching function where product terms in the basis set are combined with the XOR operation. The set of basis functions comprise $2^n$ product terms that are formed from a set of $n$ literals. Each literal may be in the form of a complemented or an uncomplemented variable, but not both. Thus, the choice of assignment of each variable appearing in either complemented or uncomplemented form corresponds to the RM transform polarity number.

The basis set includes all product terms that are each comprised of $\binom{n}{i}$ literals where $i = 0, 1, \ldots, n$. Therefore, the basis set contains $N_{\text{total}} = 2^n$ total product terms since $N_{\text{total}} = \sum_{i=0}^{n} \binom{n}{i} = 2^n$.

The RM representation is comprised of $2^n$ of product terms weighted by the RM spectral coefficients, $r_i$ and combined with the exclusive-OR operator to implement addition in $GF(2)$. In general form, the full RM expansion of a function of three variables is given by

$$f(x_1, x_2, x_3) = r_0(1) \oplus r_1(\dot{x}_1) \oplus r_2(\dot{x}_2) \oplus r_3(\dot{x}_3)$$
$$\oplus r_{12}(\dot{x}_1\dot{x}_2) \oplus r_{13}(\dot{x}_1\dot{x}_3) \oplus r_{23}(\dot{x}_2\dot{x}_3) \oplus r_{123}(\dot{x}_1\dot{x}_2\dot{x}_3).$$

In the expression above, the notation $\dot{x}_i$ indicates that every time a variable appears in the general form of the RM expansion, it will always be either positive polarity, $x_i$, or negative polarity, $\overline{x}_i$. The actual polarity assigned to each variable in the expression depends on the type of RM transform implemented. For example, switching functions can be transformed using a positive-polarity RM (PPRM), negative-polarity RM (NPRM), or more generally, a fixed-polarity RM (FPRM) matrix. The spectrum or coefficients of the RM expansion, $\mathbf{r}$, are found by computing $(\mathbf{Rf}) = \mathbf{r}$ using the RM transformation matrix, $\mathbf{R}$, and a vector, $\mathbf{f}$, containing the switching function $f$ valuations present in the function truth table. The resulting spectrum is the set of basis function weighting coefficients $r_i$ in the vector $\mathbf{r}$ where each $r_i \in \mathbb{B} = \{0, 1\}$.

The PPRM transformation matrix for a function of one variable is $\mathbf{R}_1$. Transformation matrices for functions of $n$ variables can be calculated as the outer product of $n$ $\mathbf{R}_1$ matrices. This outer product expansion that yields higher-ordered RM transformation matrices is given by

$$\mathbf{R}_1 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, \mathbf{R}_n = \bigotimes_{i=1}^{n} \mathbf{R}_1. \tag{9}$$

With PPRM, all variables in the RM expansion are in an uncomplemented form, $\dot{x}_i = x_i$. Similarly, the NPRM transformation matrix for one variable, $\mathbf{R}_{-1}$ and its outer product expansion for higher-ordered, multi-variable functions is

$$\mathbf{R}_{-1} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \mathbf{R}_{-n} = \bigotimes_{i=1}^{n} \mathbf{R}_{-1}. \tag{10}$$

With NPRM, all variables in the RM expansion are in complemented form, $\dot{x}_i = \bar{x}_i$.

There are $2^n$ different and unique RM transformation matrices for functions of $n$ variables. These matrices are conveniently specified by an $n$-bit unsigned integer value referred to as the "polarity number." The polarity number is formed by choosing some consistent order of the variables of a switching function and assigning a zero (0) to those variables that are uncomplemented and a one (1) to those variables that are to be complemented in the RM expansion. With this convention, the PPRM is then the RM expansion of a switching function where all literals are uncomplemented and thus corresponds to the polarity-zero (zero) expansion. Likewise, the NPRM expansion is one where all literals appear in complemented form. Thus, the NPRM polarity value is $2^n - 1$ since it corresponds to the value represented by a string of $n$ ones (1) in a binary string.

The general case, where some literals are complemented and others are not can be obtained through use of the appropriate fixed-polarity RM (FPRM) transformation matrix. To derive the FPRM transform for a function of size $n$, a polarity number $p$ is specified that indicates which variables will be complemented and which will not. The polarity number $p$ is formed by assigning the $i^{\text{th}}$ bit in the overall string $p$ to one (1) if the $i^{\text{th}}$ variable $x_i$ is to appear as a complemented literal and to zero (0) otherwise. Next, $\mathbf{R}_1$ and $\mathbf{R}_{-1}$ transformation matrices must be combined with the outer product and in the appropriate order as used to define the polarity number, $p$. For example, if a polarity-5 FPRM transform is desired for a function of three variables, $p = 101_2$ causes the variable polarity to be fixed so that $\dot{x}_3 = \overline{x}_3$, $\dot{x}_2 = x_2$, and $\dot{x}_1 = \overline{x}_1$ in the final RM expansion. Using $p$, the FPRM transformation matrix can be computed to be

$$\mathbf{R} = \mathbf{R}_{-1} \otimes \mathbf{R}_1 \otimes \mathbf{R}_{-1} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix},$$

$$\mathbf{R}_5 = \begin{bmatrix} 1&1&0&0&1&1&0&0 \\ 0&1&0&0&0&1&0&0 \\ 1&1&1&1&1&1&1&1 \\ 0&1&0&1&0&1&0&1 \\ 0&0&0&0&1&1&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&1&1&1&1 \\ 0&0&0&0&0&1&0&1 \end{bmatrix}.$$

In general the expression for generating the FPRM transform matrix with polarity $p$ is given as

$$\mathbf{R}_p = \bigotimes_{i=1}^{n} \mathbf{R}_{(-1)^{p_i}}. \quad (11)$$

### *E. The Sierpinski Gasket Fractal*

Sierpinski's triangle can be formed by replacing each value in Pascal's triangle with its modulo-2 result thus containing a one (1) for odd values in Pascal's triangle and zero (0) for even values. This fractal has been shown to have a relationship with the Reed-Muller forms in a large body of past work [16]–[18]. Here, we only describe a subset of the relationships that are relevant to the work described in this paper. Specifically, it is noted that Sierpinski's gasket is formed from the PPRM. The size of the fractal, measured as the number of rows is related to the dimension of the PPRM matrix. Since the dimension of the PPRM matrix are $2^n \times 2^n$, the number of resultant rows in the Sierpinski gasket as formed using the lower diagonals of the PPRM matrix is $2^n$.

*Example 1:* Consider the $8 \times 8$ PPRM matrix, $\mathbf{R}_3$, as

$$\mathbf{R}_3 = \bigotimes_{i=1}^{3} \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 1&0&0&0&0&0&0&0 \\ 1&1&0&0&0&0&0&0 \\ 1&0&1&0&0&0&0&0 \\ 1&1&1&1&0&0&0&0 \\ 1&0&0&0&1&0&0&0 \\ 1&1&0&0&1&1&0&0 \\ 1&0&1&0&1&0&1&0 \\ 1&1&1&1&1&1&1&1 \end{bmatrix}. \quad (12)$$

Extracting the rows of Sierpinski's gasket fractal from $\mathbf{R}_3$ results in the rows corresponding to $n = 0$ through $n = 7$ and is shown below.

```
              1
            1   1
          1   0   1
        1   1   1   1
      1   0   0   0   1
    1   1   0   0   1   1
  1   0   1   0   1   0   1
1   1   1   1   1   1   1   1
```

□

## III. Computer Algebra Applications

Computer algebra refers to a collection of algorithms that are useful for implementing automated symbolic mathematical processing. Examples of such systems include the Mathematica, Wolfram Alpha, Maple, and GAP software.

### *A. Pascal's Matrix for Polynomial Simplification*

We observe that the process of selecting the polynomial coefficients for $n = 5$ could be accomplished through use of one of the triangular matrix forms of the Pascal matrix and multiplying by a vector, $\mathbf{b}$, that has a one (1) in the fifth index to select the needed values. We will choose to use $\mathbf{P}_U$ so that column vectors can be used to select the appropriate row. The following example illustrates this process using a calculation of $\mathbf{a} = \mathbf{P}_U \mathbf{b}$.

*Example 2:* Consider finding the simplified polynomial form of $(x+1)^5$ using $\mathbf{P}_U$. In this case, any truncated matrix could be used so long as the dimensions are at least $6 \times 6$ or larger. Thus, we will use the $7 \times 7$ matrix $\mathbf{P}_{U7}$ in this example. A vector is formed that represents the expression of interest that we denote as $\mathbf{b}$. The $\mathbf{b}$ vector has components that represent the binomial $(x + 1)$ raised to various powers.

Using the representation of $\mathbf{b}$, the selection of the appropriate row from Pascal's triangle that corresponds to $(x+1)^5$ when the triangle is represented by the matrix $\mathbf{P}_U$ is shown in Eqn. 13. The matrix multiplication is carried out resulting in the vector $\mathbf{a}$ that contains the coefficients of the simplified and expanded polynomial form of $(x + 1)^5$ which are the coefficients of $x^5 + 5x^4 + 10x^3 + 10x^2 + 5x + 1$.

$$\mathbf{P}_{U7} = \begin{bmatrix} 1&1&1&1&1&1&1 \\ 0&1&2&3&4&5&6 \\ 0&0&1&3&6&10&15 \\ 0&0&0&1&4&10&20 \\ 0&0&0&0&1&5&15 \\ 0&0&0&0&0&1&6 \\ 0&0&0&0&0&0&1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 5 \\ 10 \\ 10 \\ 5 \\ 1 \\ 0 \end{bmatrix} \quad (13)$$

The coefficients of the simplified vector are present in the resultant product vector $\mathbf{a}$. □

Example 2 provides the motivation for the first result of this work. The $\mathbf{b}$ vector can be used to specify linear combinations of various binomials of the form $(x + 1)^n$ and with a single vector-matrix direct product operation, the expanded and simplified polynomial form can be computed from $\mathbf{a}$. The polynomial is determined with significantly less effort than required through manual calculations and is also suitable for implementation in a computer algebra algorithm.

The modified Pascal's triangle can be interpreted as providing the coefficients for the simplified and expanded polynomial form of $(x - 1)^n$ for the integer $n \geq 0$. Thus, we can use the modified Pascal's transformation matrices $\mathbf{P}_{ML}$, $\mathbf{P}_{ML}$, and $\mathbf{P}_{ML}$ in the same manner that we used $\mathbf{P}_U$ in the preceding example.

*Example 3:* Consider finding the simplified polynomial form of the expression $2(x - 1)^5 + 3(x - 1)^3 - 4(x - 1)^2 - 7$. First, we note that the largest degree in the expression is five

(5) indicating that a truncated Pascal matrix of at least $6 \times 6$ should be used. For consistency with the previous example, we will use $\mathbf{P}_{MU7}$. The vector $\mathbf{b}$ is composed based upon the coefficients of the expression. Note that the constant, 7, in the expression is the coefficient of the binomial $(x-1)^0$. Eqn. 14 contains the calculation of the simplified and expanded polynomial form of the given expression by calculating the vector $\mathbf{a} = \mathbf{P}_{MU7}\mathbf{b}$.

$$\mathbf{a} = \begin{bmatrix} 1 & -1 & 1 & -1 & 1 & -1 & 1 \\ 0 & 1 & -2 & 3 & -4 & 5 & -6 \\ 0 & 0 & 1 & -3 & 6 & -10 & 15 \\ 0 & 0 & 0 & 1 & -4 & 10 & -20 \\ 0 & 0 & 0 & 0 & 1 & -5 & 15 \\ 0 & 0 & 0 & 0 & 0 & 1 & -6 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 7 \\ 0 \\ -4 \\ 3 \\ 0 \\ 2 \\ 0 \end{bmatrix} = \begin{bmatrix} -2 \\ 27 \\ -33 \\ 23 \\ -10 \\ 2 \\ 0 \end{bmatrix} \quad (14)$$

□

### B. Fixed Polarity Pascal's (FPP) Matrices

*Observation 3:* A family of the upper triangular version of Pascal's transformation matrix may be formulated by selecting some column vectors from $\mathbf{P}_U$ and the remaining column vectors from $\mathbf{P}_{MU}$. The resulting matrix, denoted $\mathbf{P}_{pU}$ is characterized by an integer $p$ referred to as the "polarity number" of this form of Pascal's matrix. The polarity number is the decimal equivalent of the binary string formed by assigning a zero (0) to those columns from $\mathbf{P}_U$ and a one (1) to those columns from $\mathbf{P}_{MU}$. The string is read from left to right with the most significant bit corresponding to the binomial exponent $n = 0$. We refer to these forms of Pascal's transformation matrices as the fixed-polarity form since each polarity number refers to a specific matrix structure. □

The fixed-polarity Pascal matrix can be used to find the simplified polynomial expansion of a linear combination of binomials raised to a power wherein those binomials may be either $(x+1)^n$ or $(x-1)^n$. For those values of $n$ where the binomial $(x+1)$ is present, the corresponding column vector is obtained from the $\mathbf{P}_U$ matrix and where $(x-1)$ is used, the column vector corresponds to that from the $\mathbf{P}_{MU}$ matrix. If no binomial is specified for a particular values of $n$, any arbitrary column vector may be used for this particular application as it is considered a don't care.

*Example 4:* Consider the linear combination of binomials, $2(x-1)^5 + 3(x+1)^3 - 4(x-1)^2 + 7$. If it is desired to find the simplified polynomial expansion of this expression, a fixed-polarity Pascal's matrix can be formulated wherein columns 0 and 3 are assigned from the $\mathbf{P}_U$ matrix since these columns correspond to the terms $7(x+1)^0$ and $3(x+1)^3$ respectively. Likewise columns 2 and 5 are assigned from the $\mathbf{P}_{MU}$ matrix since they correspond to the terms $-4(x-1)^2$ and $2(x-1)^5$ respectively. In terms of polarity number, the bit string is 0X10X1X where "X" is a don't care. For simplicity, we will assign the don't cares be to zero, although this is an arbitrary assignment. After assigning the don't care values to zero, the polarity number becomes $0010010_2$ or $18_{10}$ in decimal. Likewise, the vector $\mathbf{b}^T = [7, 0, -4, 3, 0, 2, 0]$ as obtained from the given linear combination of binomials raised to various powers. Eqn. 15 show the calculation, $\mathbf{a} = \mathbf{P}_{18U7}\mathbf{b}$.

$$\mathbf{a} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & -1 & 1 \\ 0 & 1 & -2 & 3 & 4 & 5 & 6 \\ 0 & 0 & 1 & 3 & 6 & -10 & 15 \\ 0 & 0 & 0 & 1 & 4 & 10 & 20 \\ 0 & 0 & 0 & 0 & 1 & -5 & 15 \\ 0 & 0 & 0 & 0 & 0 & 1 & 6 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 7 \\ 0 \\ -4 \\ 3 \\ 0 \\ 2 \\ 0 \end{bmatrix} = \begin{bmatrix} 4 \\ 27 \\ -15 \\ 23 \\ -10 \\ 2 \\ 0 \end{bmatrix} \quad (15)$$

The result $\mathbf{a}^T = [4, 27, -15, 23, -10, 2, 0]$ indicates that the simplified and expanded polynomial is $2x^5 - 10x^4 + 23x^3 - 15x^2 + 27x + 4$. □

*Definition 1:* In considering a fixed-polarity and truncated Pascal's transformation matrix, each column vector is associated with a binomial $a_n(x \pm 1)^n$ for $n = 0, 1, \ldots, k$. Furthermore, when the matrix is being formed for the purpose of calculating the coefficients of a simplified polynomial expansion that corresponds to a weighted linear combination of binomials where the weights are denoted by $a_i \in \mathbb{R}$. When there is not a binomial present in the corresponding weighted linear combination of binomials, this is equivalent to the binomial being present in the expression with a weighting coefficient that is zero valued, $a_i = 0$. The column vector in the truncated fixed-polarity Pascal transformation matrix is said to be a "don't care column vector." □

*Lemma 4:* The maximum number of different fixed-polarity Pascal's transformation matrices is $2^{k+1}$ when considering the family of fixed-polarity matrices that are truncated to $(k+1) \times (k+1)$ and when don't care column vectors are all set to the same vector that is arbitrarily chosen.

□

*Corollary 2:* The matrix $\mathbf{P}_U$ is the positive polarity Pascal's transformation matrix with a polarity number of $p = 00\ldots0_2$. □

*Corollary 3:* The matrix $\mathbf{P}_{MU}$ is the negative polarity Pascal's transformation matrix with a polarity number of $p = 11\ldots1_2$.

□

### C. Decomposition of Polynomials into Linear Combinations of Binomials

We have previously considered the application of fixed polarity Pascal transformation matrices for the purpose of determining the simplified polynomial expansions of weighted linear combinations of powers of binomials of the form $(x+1)^n$ and $(x-1)^n$. While this method offers a convenient method and one that potentially requires less effort than manual symbolic polynomial multiplication, the inverse problem of finding a decomposition of a polynomial into a weighted sum of binomials is more difficult to calculate symbolically since it requires the computation of polynomial division operations.

*Theorem 2:* A polynomial in simplified and expanded form can be factored into a linear combination of binomials in the $(x+1)^n$ by calculating a single direct matrix-vector product. □

*Example 5:* Consider the polynomial $3x^5 - 7x^4 - 7x^3 + 10x - 1$. It is desired to decompose this polynomial into a linear combination of binomials of the form $(x+1)^n$ for $n = 0, 1, \ldots, 5$. For this example, we will use a transformation matrix of size $7 \times 7$. The corresponding coefficient vector is $\mathbf{a}^T = [-1, 10, 0, -7, -7, 3, 0]$. Since we are finding binomial coefficients from polynomial coefficients, we are performing the calculation of $\mathbf{b} = (\mathbf{P}_{U7})^{-1}\mathbf{a} = \mathbf{P}_{MU7}\mathbf{a}$ using the identity found in Eqn. 8. This calculation is shown below in Eqn. 16.

$$\mathbf{b} = \mathbf{P}_{MU7}\mathbf{a} = \begin{bmatrix} 1 & -1 & 1 & -1 & 1 & -1 & 1 \\ 0 & 1 & -2 & 3 & -4 & 5 & -6 \\ 0 & 0 & 1 & -3 & 6 & -10 & 15 \\ 0 & 0 & 0 & 1 & -4 & 10 & -20 \\ 0 & 0 & 0 & 0 & 1 & -5 & 15 \\ 0 & 0 & 0 & 0 & 0 & 1 & -6 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 \\ 10 \\ 0 \\ -7 \\ -7 \\ 3 \\ 0 \end{bmatrix} = \begin{bmatrix} -14 \\ 32 \\ -51 \\ 51 \\ -22 \\ 3 \\ 0 \end{bmatrix} \quad (16)$$

Therefore, the decomposition vector is $\mathbf{b}^T = [-14, 32, -51, 51, -22, 3, 0]$ indicating the decomposition is that shown in Eqn. 17.

$$\begin{aligned} &3x^5 - 7x^4 - 7x^3 + 10x - 1 \\ &= 3(x+1)^5 - 22(x+1)^4 + 51(x+1)^3 \\ &-51(x+1)^2 + 32(x+1) - 14 \end{aligned} \quad (17)$$

□

The dual relationship also holds in that a polynomial may be decomposed into a weighted linear combination of binomials of the form $(x-1)^n$. In this case, the $\mathbf{P}_U$ matrix is used to find the decomposition of a simplified polynomial expansion in the form of a linear combination of weighted $(x-1)^n$ binomials raised to various integer powers $n$ as it is the inverse of $\mathbf{P}_{MU}$.

### D. Polynomial Decomposition for General Polarity Values

We have shown how FPP transformation matrices can be computed for all polarity values $p = 0, \cdots, 2^{n-1} - 1$ allowing for weighted sums of a mixture of binomials of the form $(x+1)^n$ and $(x-1)^n$ to be easily simplified into the form of an $n^{\text{th}}$-degree polynomial. In the case of $p = 0$ and $p = 2^{n-1} - 1$, the FPP matrices are $\mathbf{P}_U$ and $\mathbf{P}_{MU}$. For all other polarity values, the FPP matrix is easily composed using column vectors from $\mathbf{P}_U$ and $\mathbf{P}_{MU}$ in accordance with the binary form of the polarity number.

We have also shown how the $\mathbf{P}_U$ and $\mathbf{P}_{MU}$ matrices can be used to decompose a polynomial into a weighted sum of binomials. This is a more difficult task to achieve symbolically and is sometimes referred to as "factoring a polynomial." In these cases, we utilized $\mathbf{P}_U^{-1} = \mathbf{P}_{MU}$ and $\mathbf{P}_{MU}^{-1} = \mathbf{P}_U$ thus enabling the inverse computations yielding the $\mathbf{b}$ vector binomial weights to be achieved given the polynomial coefficients in the $\mathbf{a}$ vector. The remaining case is that of decomposing a polynomial into a weighted sum of mixed binomials that utilize $(x+1)^n$ for some values of $n$ and $(x-1)^n$ for the other values of $n$. In this case, the determination of the inverse of the FPP matrix for polarity values other than 0 and $2^{n-1} - 1$ requires a computation and is not achievable through simply composing a matrix of various column vectors from either $\mathbf{P}_U$ or $\mathbf{P}_{MU}$. Fortunately, due to the structure and properties of the FPP matrices, efficient algorithms are available for constructing the inverse matrix. These methods have a polynomial complexity for finding the FPP inverse matrix, making the decomposition of a polynomial into the the more general case of mixed $(x+1)^n$ and $(x-1)^n$ binomials a practical and viable decomposition technique.

The process for decomposing a polynomial into a weighted sum of $(x+1)^n$ and $(x-1)^n$ can be stated as follows.

1) Choose values of $n$ that correspond to $(x+1)$ and $(x-1)$ and form the polarity number.
2) Given the polarity number and the degree of the polynomial to be decomposed, construct the truncated FPP transformation matrix using column vectors from $\mathbf{P}_U$ and $\mathbf{P}_{MU}$.
3) Determine the inverse of the truncated FPP matrix using an efficient algorithm that takes advantage of FPP matrix properties.
4) Form the $\mathbf{a}$ vector from the coefficients of the polynomial to be decomposed.
5) Multiply the FPP matrix inverse with the $\mathbf{a}$ vector to obtain the $\mathbf{b}$ vector that contains the weights of the $(x+1)^n$ and $(x-1)^n$ binomials.

### E. Inverse of FPP Matrix

From Observation 1, the FPP matrices are unitriangular matrices. The unitriangular structure allows the FPP matrix inverse to be efficiently computed.

*Definition 2:* The general notation for a $k \times k$ truncated FPP matrix with polarity value $p$ is $\mathbf{P}_{pUk}$. The polarity value $p$ indicates whether the $i^{\text{th}}$ column vector of $\mathbf{P}_{pUk}$ is equivalent to that of $\mathbf{P}_{Uk}$ or $\mathbf{P}_{MUk}$. The dimension of the matrix $k$ is related to the largest integer power $n$ of the polynomial or binomial that can be processed by $\mathbf{P}_{pUk}$ matrix as $k = n+1$. □

*Lemma 5:* The eigenvalues of the $k \times k$ truncated FPP matrices, $\mathbf{P}_{pUk}$, are unity with a multiplicity of $k$.

□

*Theorem 3:* We express $\mathbf{P}_{pUk} = \mathbf{I} + \mathbf{T}_{pUk}$ where $\mathbf{T}_{pUk}$ is an upper triangular matrix with all diagonal values equal to zero. The inverse of a truncated FPP matrix can be expressed in closed form as given in Eqn. 18.

$$\mathbf{P}_{pUk}^{-1} = \mathbf{I} - \mathbf{T}_{pUk} + \mathbf{T}_{pUk}^2 - \mathbf{T}_{pUk}^3 + \cdots + (-1)^{k-1}\mathbf{T}_{pUk}^{k-1} \quad (18)$$

□

Although we can use the result of Theorem 3 to compute the inverse matrix $\mathbf{P}_{pUk}$, it is often the case that linear equations can be used for $\mathbf{a} = \mathbf{P}_{pUk}\mathbf{b}$ when $\mathbf{a}$ and $\mathbf{P}_{pUk}$ are known. It is further the case that $\mathbf{P}_{pUk}$ is triangular allowing for the linear equations to be solved iteratively beginning with the row of $\mathbf{P}_{pUk}$ containing a single non-zero element and back substituting into the next row that has two non-zero elements, *etc.* This iterative process allows for the components of a $k$-dimensional $\mathbf{b}$ vector to be obtained with $k$ back substitutions and is very efficient. In this way it is not required to compute the various powers of the matrix $\mathbf{T}_{kUp}$ as required when using Eqn. 18.

*Example 6:* Consider the polynomial $3x^5 - 7x^4 - 7x^3 + 10x - 1$. It is desired to decompose this polynomial into a linear combination of binomials of the form $(x+1)^n$ for $n \in \{3, 4, 5\}$ and for $(x-1)^n$ for $n \in \{0, 1, 2\}$.

First the appropriate FPP transform matrix is constructed. Since the largest binomial degree is five (5), it is necessary to use a truncated Pascal's matrix of at least dimension $6 \times 6$. The truncated $6 \times 6$ matrices corresponding to the decomposition into $(x+1)$ and into $(x-1)$ are shown below in Eqns. 19 and 20.

$$\mathbf{P}_{MU6} = \begin{bmatrix} 1 & -1 & 1 & -1 & 1 & -1 \\ 0 & 1 & -2 & 3 & -4 & 5 \\ 0 & 0 & 1 & -3 & 6 & -10 \\ 0 & 0 & 0 & 1 & -4 & 10 \\ 0 & 0 & 0 & 0 & 1 & -5 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (19)$$

$$\mathbf{P}_{U6} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 0 & 1 & 3 & 6 & 10 \\ 0 & 0 & 0 & 1 & 4 & 10 \\ 0 & 0 & 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (20)$$

We formulate the polarity number by composing a bit string beginning with the most significant bit first that has a zero (0) in the $k^{\text{th}}$ position for instances of $(x+1)^k$ and a one (1) in the $k^{\text{th}}$ position for instances of $(x-1)^k$. In the case of this example where we wish to have a decomposition with $(x+1)^k$ for $k \in \{3, 4, 5\}$ and $(x-1)^k$ for $k \in \{0, 1, 2\}$. Thus, the polarity number is $000111_2 = 7_{10}$. Next, we determine the transformation matrix for the decomposition, $\mathbf{P}_{7U6}$ by using the $k^{\text{th}}$ column vector in $\mathbf{P}_{MU6}$ when the $k^{\text{th}}$ bit in the polarity number is one (1) and the $k^{\text{th}}$ column vector in $\mathbf{P}_{U6}$ when the $k^{\text{th}}$ bit in the polarity number is zero (0) as shown in Eqn. 21.

$$\mathbf{P}_{7U6} = \begin{bmatrix} 1 & 1 & 1 & -1 & 1 & -1 \\ 0 & 1 & 2 & 3 & -4 & 5 \\ 0 & 0 & 1 & -3 & 6 & -10 \\ 0 & 0 & 0 & 1 & -4 & 10 \\ 0 & 0 & 0 & 0 & 1 & -5 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (21)$$

For this example, we will use a transformation matrix of size $6 \times 6$. The corresponding coefficient vector is $\mathbf{a}^T = [-1, 10, 0, -7, -7, 3]$. The calculation is shown below in Eqn. 22.

$$\mathbf{a} = \mathbf{P}_{7U6}\mathbf{b} = \begin{bmatrix} -1 \\ 10 \\ 0 \\ -7 \\ -7 \\ 3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & -1 & 1 & -1 \\ 0 & 1 & 2 & 3 & -4 & 5 \\ 0 & 0 & 1 & -3 & 6 & -10 \\ 0 & 0 & 0 & 1 & -4 & 10 \\ 0 & 0 & 0 & 0 & 1 & -5 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{bmatrix} \quad (22)$$

Eqn. 22 can be solved by first computing the closed form of $\mathbf{P}_{7U6}^{-1}$ using the result of Theorem 3 or it can be solved through iterative back substitution. Through back substitution, we observe that

$$(1)b_5 = 3 \rightarrow b_5 = 3,$$

$$(1)b_4 + (-5)b_5 = -7 \rightarrow b_4 = -7 + 15 = 8,$$

$$(1)b_3 + (-4)b_4 + (10)b_5 = -7, \\ \rightarrow b_3 = -7 + 32 - 30 = -5,$$

$$(1)b_2 + (-3)b_3 + (6)b_4 + (-10)b_5 = 0, \\ \rightarrow b_2 = -15 - 48 + 30 = -33,$$

$$(1)b_1 + (2)b_2 + (3)b_3 + (-4)b_4 + (5)b_5 = 10, \\ \rightarrow b_1 = 10 + 66 + 15 + 32 - 15 = 108,$$

$$(1)b_0 + (1)b_1 + (1)b_2 + (-1)b_3 + (1)b_4 + (-1)b_5 = -1, \\ \rightarrow b_0 = -1 - 108 + 33 - 5 - 8 + 3 = -86.$$

Therefore, the decomposition vector is $\mathbf{b}^T = [-86, 108, -33, -5, 8, 3]$ indicating the decomposition is that shown in Eqn. 23.

$$\begin{aligned} &3x^5 - 7x^4 - 7x^3 + 10x - 1 \\ &= 3(x-1)^5 + 8(x-1)^4 - 5(x-1)^3 \\ &-33(x+1)^2 + 108(x+1) - 86 \end{aligned} \quad (23)$$

The decomposition holds and the polynomial coefficient vector $\mathbf{a}^T = [-1, 10, 0, -7, -7, 3]$ decomposes into the binomial coefficient vector $\mathbf{b}^T = [-86, 108, -33, -5, 8, 3]$ for the fixed polarity value $p = 7$. □

### F. Generalization of Methods for $(x \pm a)^n$

In the previous sections, methods for efficiently deriving the simplified and expanded polynomial from a binomial form and vice-versa were discussed for the homogeneous and mixed polarity cases when binomial forms are $(x \pm 1)^n$. These techniques may be generalized for the case when $(x \pm a)^n$ since it is simple to modify Pascal's triangle as described by the Binomial Theorem. Accommodation for $a$ is made by multiplying the diagonals of Pascal's triangle by $a$ raised to some power, $a^d$, as seen below.

$$\begin{array}{ccccccccccc} & & & & & 1a^0 & & & & & \\ & & & & 1a^0 & & 1a^1 & & & & \\ & & & 1a^0 & & 2a^1 & & 1a^2 & & & \\ & & 1a^0 & & 3a^1 & & 3a^2 & & 1a^3 & & \\ & 1a^0 & & 4a^1 & & 6a^2 & & 4a^3 & & 1a^4 & \\ 1a^0 & & 5a^1 & & 10a^2 & & 10a^3 & & 5a^4 & & 1a^5 \end{array}$$

Using this version of Pascal's triangle, standard and modified versions of the $\mathbf{P}_U$ and $\mathbf{P}_L$ transformation matrices can be defined. With these, polynomial simplification or decomposition using the aforementioned techniques can be implemented.

## IV. Conclusion

We have introduced the concept of a fixed polarity Pascal (FPP) transformation matrix that is analogous to the fixed-polarity Reed-Muller (FPRM) transformation matrices. This concept resulted in the definition of a family of linear transformations based upon the Pascal transformation that are characterized by an integer value referred to as the "polarity number." We introduced and proved several characteristics of this family of transformation matrices and showed how the inverse FPP transforms can be calculated efficiently in closed form. We applied these results to a computer algebra application resulting in methods that are suitable for implementation as efficient algorithms to both simplify polynomials and to decompose polynomials into weighted sums of binomials raised to a power.

In future work, we will investigate methods that attempt to find the FPP polarity number, and hence the specific FPP transformation matrix that results in a binomial decomposition that has as few binomial terms as possible. This problem is a direct analogy to the well-known and well-studied problem of finding the polarity number for a FPRM transform that results in a switching function expressed in a RM expansion that has as few product terms as possible. We also believe that the FPP transforms can be used as the core operation in many other computer algebra applications and we intend to investigate those as well.

## REFERENCES


[1] W. Sierpiński, "Sur une courbe cantorienne dont tout point est un point de ramification," *C.R. Acad. Sci. Paris*, vol. 160, p. 302, 1915.

[2] B. Pascal, *"Traité du triangle arithmetique" in Œuvres de Blaise Pascal (Reprint) volume 3, 445–503*. Hachette, 1904–1914, 1665.

[3] C. Moraga, R. Stanković, and M. Stanković, "The Pascal triangle (1654), the Reed-Muller-Foutier transform (1992), and the discrete Pascal transform (2005)," in $46^{th}$ *Int. Symp. on Multiple-valued Logic*. IEEE Computer Press, 2016, pp. 229–234.

[4] M. F. Aburdene and J. E. Dorband, "Unification of Legendre, Laguerre, Hermite, and binomial discrete transforms using Pascal's matrix," *Multidimensional Systems and Signal Processing*, vol. 5, no. 3, pp. 301–305, 1994.

[5] M. F. Aburdene and T. J. Goodman, "The discrete Pascal transform and its applications," *IEEE Signal Processing Letters*, vol. 12, no. 7, pp. 493–495, 2005.

[6] T.-B. Deng, S. Chivapreecha, and K. Dejhan, "Unified pascal matrix for first-order $s-z$ domain transformations," *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2130–2139, 2009.

[7] R. S. Stanković, J. Astola, and C. Moraga, "Pascal matrices, Reed-Muller expressions and Reed-Muller error correcting codes," *Zbornik Radova*, vol. 18, no. 26, pp. 145–172, 2015.

[8] A. N. Skodras, "Fast discrete Pascal transform," *Electronics Letters*, vol. 42, no. 23, 2006.

[9] D. B. Gajić and R. S. Stanković, "Fast computation of the discrete Pascal transform," in $47^{th}$ *Int. Symp. on Multiple-valued Logic*. IEEE Computer Press, 2017, pp. 149–154.

[10] G. S. Call and D. J. Velleman, "Pascal's matrices," *The American Mathematical Monthly*, vol. 100, no. 4, pp. 372–376, 1993.

[11] A. Edelman and G. Strang, "Pascal matrices," http://web.mit.edu/18.06/www/Essays/pascal-work.pdf, MIT, Cambridge, MA, Tech. Rep., 2003.

[12] D. E. Knuth, "Two notes on notation," *American Mathematical Monthly*, vol. 97, pp. 403–422, 1992.

[13] R. L. Graham, D. E. Knuth, and O. Patashnik, *Concrete Mathematics: A Foundation for Computer Science*, 2nd ed. Addison & Wesley, 1994.

[14] I. S. Reed, "A class of multiple-error-correcting codes and the decoding scheme," *Transactions of the IRE Professional Group on Information Theory*, vol. 4, pp. 38–49, 1954.

[15] D. E. Muller, "Application of Boolean algebra to switching circuit design and to error detection," *IRE Transactions on Electronic Computers*, vol. 3, pp. 6–12, 1954.

[16] G. W. Dueck, V. P. Shmerko, J. T. Butler, and S. N. Yanushkevich, "On the number of generators for transeunt triangles," *Discrete Applied Mathematics*, vol. 108, no. 3, pp. 309–316, 1999.

[17] D. Maslov, "A method to find the best mixed polarity Reed-Muller expansion," Master's thesis, The Faculty of Computer Science, The University of New Brunswick, Fredericton, New Brunswick, Canada, 2001.

[18] G. W. Dueck, D. Maslov, V. P. Shmerko, J. T. Butler, and S. N. Yanushkevich, "A method to find the best mixed polarity Reed-Muller expression using transeunt triangle," in $5^{th}$ *Int. Reed-Muller Workshop*. unpublished workshop proceedings, 2001, pp. 82–92.