GROUP THEORETIC APPROACH TO THE OPTIMIZATION OF SPECTRAL REPRESENTATIONS FOR CIRCUIT SYNTHESIS

Radomir S. Stanković

Mathematical Institute of SASA, Knez Mihailova 36, Belgrade
e-mail: Radomir.Stankovic@gmail.com
Serbia

Abstract: Spectral representations are a suitable way to represent discrete signals that are inputs and outputs of discrete systems. Therefore, they can be used as a starting point in digital circuit design. In this paper, we discuss an approach to the optimization of spectral representations to reduce the circuit complexity.

Key words: Spectra representations, Circuit synthesis, Group representations, Fourier expressions.

1. INTRODUCTION

Functional decomposition is a standard part of many circuit design procedures. It is essentially important in various variants of memory bases synthesis, since the function to be realized must be split into subfunctions which can be mapped into standardized basic memory blocks [5]. Spectral representations are by their nature decomposition of a function into a linear combination of functions with known properties. Therefore, spectral representations are well suited as discrete functions descriptions for circuit synthesis [1].

An n-variable k-output discrete function $f(x) = f(x_1,x_2, \ldots , x_n) = (f_0,f_1, \ldots , f_k)$ is defined as a mapping

$$f = \times_{i=1}^{n} S_i \rightarrow L^k,$$

where each variable x_i takes values in a finite non-empty set S_i and each output takes values in another finite set L. It is a customary practice to assume that all the outputs take values in the same set mainly for practical reasons. Such a function can be viewed as a function on a finite group G which is a direct product of subgroups G_i with the support sets S_i. Therefore,

$$G = \times_{i=1}^{n} G_i, \quad g = \prod_{i=1}^{n} g_i,$$

where $g = |G|$ is the order of the group G expressed as the product of orders g_i of subgroups G_i. The range of the function f is usually assumed to be a field P which can be the field of complex numbers C, rational numbers Q, or a finite (Galois) field $GF(p)$.

The function f can be expressed as a series expansion

$$f = \sum_{i=1}^{n} a_i \phi_i(x), \quad i \in \{0,1,\dots,g-1\}, \quad a_i \in P, \qquad (1)$$

where the functions $\phi_i(x)$ are called the basis functions, short basis, and a_i are spectral coefficients.

The expression (1) is the basis for circuit synthesis from spectral representations [1].

2. CIRCUIT SYNTHESIS FROM SPECTRAL REPRESENTATIONS

Figure 1 shows the basic principle of circuit synthesis based on spectral representations. The circuit consists of a generator of basis functions $\phi_i(x)$, a memory module to store the spectral coefficients, a multiplier array for multiplying the non-zero coefficients with the basis functions to which they are associated, and an adder array to perform the addition and generate the function f.

This is obviously a programmable architecture sharing good features of such circuit realizations as reduced hardware development cost and design time [1]. This structure is convenient for fast prototyping and when frequent change of the function to be realized is required, since reprogramming is reduced to the change of the spectral coefficients, i.e., the memory content [5].

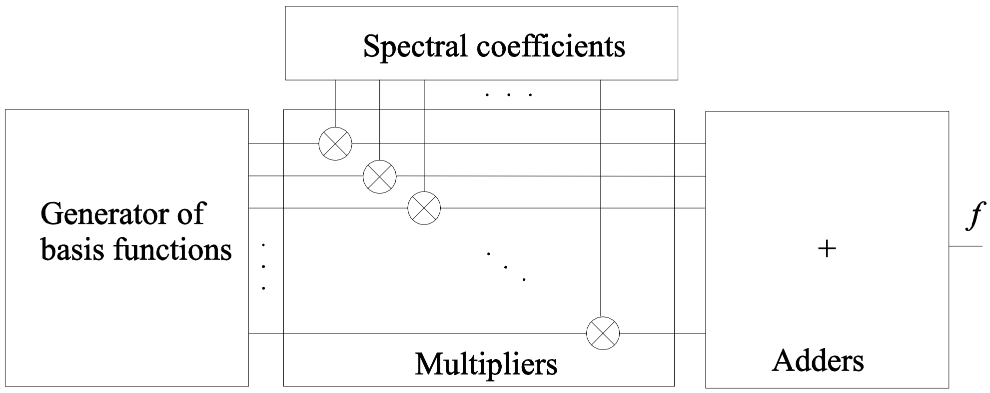

Fig. 1. The principle of circuit synthesis from spectral representations.

2.1. Optimization of spectral representations for circuit synthesis

From the structure of the circuit in Figure 1, immediately follow the optimization criteria as

1. Reduction of the number of non-zero spectral coefficients a_i, since this reduces the size of the memory, and the number of multipliers and adders.

2. Basis function should be selected such that can be easily generated in hardware, and possibly that can be expressed in terms of Boolean variables for simple realization with two state circuits.

3. For the selected basis, there exists a fast algorithm for computing the spectral coefficients a_i.

Classical optimization approaches are directed towards

1. Selecting a suitable basis.
2. When a basis selected, manipulation with basis functions, as for example their reordering.

In these approaches the assumed underlying group G is preserved as well as the range field P. For example, in the case of Boolean functions, the underlying domain group is the finite dyadic group C_2^n consisting of the set of 2^n n - tuples equipped with the operation modulo 2 addition, usually denoted as the Exclusive-OR (EXOR). For the range, the $GF(2)$ is assumed or the file of rational numbers Q.

In this case, among the widely used basis functions we mention the Walsh functions, the Haar functions, the arithmetic transform functions if the range is Q, and the Reed-Muller functions for $GF(2)$. Notice that the discrete Walsh functions are group characters of the finite dyadic group. Therefore, the Walsh basis is the Fourier basis on C_2^n. The other mentioned basis functions are viewed as the Fourier-like basis, since satisfy many although not all properties of the Fourier basis including the existence of the fast computing algorithms corresponding to the Fast Fourier transform (FFT). In all the mentioned examples, the basis functions can be expressed in terms of Boolean variables.

Despite these good features of the pointed out and related basis functions, there are many functions for which these spectral representations are still complex in the number of non-zero spectral coefficients. As another complexity measure, we use the number of 1-bits required to represent the non-zero coefficients. There might be cases where the number of non-zero coefficients is small, but their values are such that many non-zero bits are required to represent them.

The main drawback of this approach to the optimization of spectral representations is that given a function to be realized we do not know in advance or after some possibly simple analysis which basis to select for a spectral representation of it. In practice, a library of transforms, i.e., a collection of basis functions is provided and then spectral coefficients computed for each item in the library. The basis with the smallest number of non-zero coefficients, or the minimal number of 1-bits to represent them is selected.

3. GROPUT THEORETIC APPROACH TO THE OPTIMIZATION OF SPECTRAL REPRESENTATIONS

In this paper, we propose a different approach to the optimization of spectral representations. Instead of preserving the same domain group and change the basis, we change the domain group and work all the time with the Fourier basis. In this way, we can use all good features of the Fourier basis and still find a compact representation with acceptable complexity measures.

Changing the group structure means that we can encode variables taking values in initially specified subgroups G_i by variables with values in another set of subgroups whose product results in a group of the same order as the initial domain group. The following example illustrates this statement. For the definition of the groups used in this example, we refer to [8] or other related literature.

Example 1. A Boolean function in three variables $f(x_1,x_2,x_3)$ is defined as a function on the group $G = C_2^3$, where $C_2=(\{0,1\},\oplus_2)$, where $\oplus$ is the addition modulo 2. Each variable x_i, $i = \{1,2,3\}$ takes its values in the corresponding subgroup C_2. In this case, it is represented by the vector

$\mathbf{F} = [f(000), f(001), f(010), f(011), f(100), f(101), f(110), f(111)]^T$.

The same function can be viewed as a function on the group $G = C_2 \times C_4$, where $C_4 = (\{0,1,2,3\}, \oplus_4)$, i.e., it is the cyclic group of order 4. In this case, the function vector is encoded as $\mathbf{F} = [f(0,0), f(0,1), f(0,2), f(0,3), f(1,0), f(1,1), f(1,2), f(1,3)]^T$. Alternatively, the same function can be viewed as the function in a single eight-valued variable on the cyclic group $C_8 = (\{0,1,2,3,4,5,6,7\}, \oplus_8)$, or as a function on the quaternion group Q_2 which is a non-Abelian group and its Fourier basis is defined in terms of unitary irreducible representations instead of group characters as in the case of Abelian groups [8]. This results into a Fourier spectrum with 5 instead of 8 Fourier coefficients, but the fifth coefficient is a (2×2) matrix, thus with its elements we have again 8 values, since Fourier transform preserves the complete information about the function represented [8]. By changing the group structure, we change the basis, but it is again a Fourier basis, and might provide representation of different complexity.

Figure 2 shows the flow-graphs for the Fast Fourier transform (FFT) on the groups C_2^3, $C_2 \times C_4$, and Q_2. The difference in these graphs illustrates indirectly the differences in the structure of the groups. Table 1 shows the number of additions to perform the Fourier transform on these groups. The group $C_2 \times C_4$ requires the smallest number of steps, while the group Q_2 provides for the smallest number of additions.

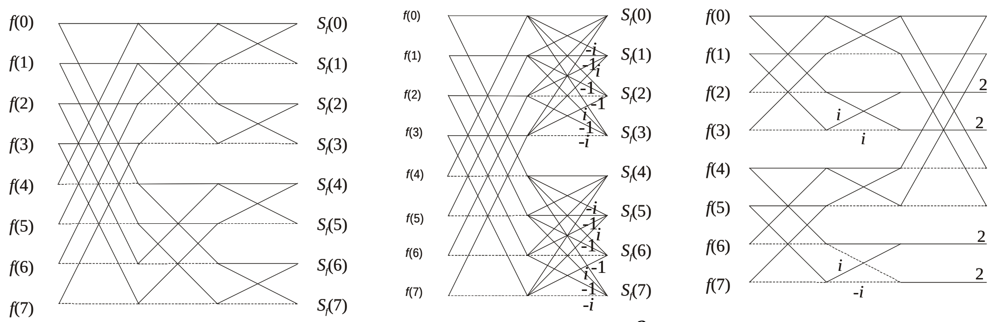

Fig. 2. The flow-graphs of FFT on groups C_2^3, $C_2 \times C_4$, and Q_2.

Table 1 The number of additions to perform FFT on C_2^3, $C_2 \times C_4$, and Q_2.

Group	Number of steps	Number of additions
C_2^3	3	24
$C_2 \times C_4$	2	32
Q_2	3	20

The Fourier transform matrices on cyclic groups C_2, C_4, the dihedral group D_4, and the quaternion group Q_2 are given in equations (2), (3), and (4). These groups and related Fourier expressions will be used in the examples bellow.

$$\mathbf{W}(1) = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \qquad \mathbf{V}(2) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{bmatrix}, \tag{2}$$

$$\mathbf{D}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 2 & 0 & -2 & 0 & 2 & 0 & -2 & 0 \\ 0 & -2 & 0 & 2 & 0 & 2 & 0 & -2 \\ 0 & 2 & 0 & -2 & 0 & 2 & 0 & -2 \\ 2 & 0 & -2 & 0 & -2 & 0 & 2 & 0 \end{bmatrix} \tag{3}$$

$$\mathbf{Q}_2 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 2 & -2i & -2 & 2i & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -2 & 2i & 2 & -2i \\ 0 & 0 & 0 & 0 & 2 & 2i & -2 & -2i \\ 2 & 2i & -2 & -2i & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4}$$

When a group G is represented as a direct product of some subgroups G_i, its group representations can be derived as the Kronecker product of representations of the constituent subgroups G_i. Notice that if two or more non-Abelian groups are used as subgroups, then the corresponding representations are not Fourier representations, however, still the term generalized Fourier representations is usually used [3]. Therefore, in this paper, we will simply speak about the Fourier representations also in such cases.

4. ILLUSTRATIVE EXAMPLES

In this section, we present some examples that illustrate the possibility of using groups of different structure to provide more compact Fourier representations in the number of non-zero coefficients and 1-bits to represent them. In these examples, we use the groups C_2^4, $C_2 \times C_2 \times C_4$, $C_4 \times C_4$, $C_2 \times D_4$, $C_2 \times Q_2$.

Example 2. Consider the function $f(x_1, x_2) = 1 \oplus_4 x_1 x_2$, where $x_1, x_2 \in \{0,1,2,3\}$ and the multiplication is modulo 4. The function vector of this function is

F = $[1,1,1,1,1,2,3,0,1,3,1,3,1,0,3,2]^T$. There are two zero values, and other values require 23 non-zero bits to represent them.

Table 2 shows the number of zero coefficients in Fourier representations of this function and the number of 1-bits to represent them on different groups. In this example, the Fourier representations on both non-Abelian groups considered require the smaller number of non-zero coefficients and 1-bits.

Table 2. Number of zero coefficients and 1-bits for the function f in Example 2.

Group	Number of zero coefficients	Number of 1-bits
C_2^4	2	22
$C_2 \times C_2 \times C_4$	2	22
$C_4 \times C_4$	2	22
$C_2 \times D_4$	6	15
$C_2 \times Q_2$	6	15

Example 3. The segment index encoding (SIE) function for *n* = 4 is specified by the function vector $F = [0,1,1,1,2,2,2,2,2,3,3,3,4,4,4,5]^T$ [5]. Table 3 shows the number of zero coefficients in Fourier representations of the SIE function and the number of 1-bits to represent them on different groups. In this example, the Fourier transform on non-Abelian groups require the smaller number of non-zero coefficients and 1-bits.

Table 3. Number of zero coefficients and 1-bits for the function SIE in Example 3.

Group	Number of zero coefficients	Number of 1-bits
C_2^4	0	36
$C_2 \times C_2 \times C_4$	0	33
$C_4 \times C_4$	0	43
$C_2 \times D_4$	2	30
$C_2 \times Q_2$	2	30

It is of course reasonable to expect that there are functions for which the Fourier representations on different groups do not provide more compact expressions. An example of these cases for the considered domain groups is the following.

Example 4 Consider the function longest prefix match which is for *n* = 4 specified by the function vector $\mathbf{F} = [5,5,5,5,2,2,3,3,1,4,4,4,4,4,4,4]^T$ [5]. There are no zero elements, and for the representaiton of function values 22 1-bits are required. None of the Fourier spectra of this function on groups considered in the above examples contains zero coefficients. For their representations, 40, 45, 55, 46, and 36, 1-bits are required. It is interesting to observe that among them the Fourier coefficients on the group $C_2 \times Q_2$ although are complex-valued require the smallest number of 1-bits.

5. OPTIMIZATION OF FOURIER REPRESENTATIONS

The above discussed examples illustrate that there are functions for which Fourier representations on finite non-Abelian groups provide more compact representations in the number of either non-zero coefficients, the number of 1-bits required to represent them, or both at the same time. These examples show that there is sense to consider domain groups of different structure to provide Fourier representations for circuit synthesis.

Main problem is that, as in many similar optimization problems in Switching theory, given a functon to be realized, there are no methods to determine in advance the domain group that will provide a compact representation for it. This is the same situation as for example selecting the Boolean expression for a given function [4], choose polarity of variables in Reed-Muller expessions, or determining the order of variables in representations by decision diagrams [7]. Further examples of this calss of problems are selection of the input and ouptut translations in optimisation of reversible logic circuits [2], determination of the linear transform of variables in minimization of circuits realzing index generating functions [6], and many others.

In the case of such problems, a search for an aceptable instead of the minimal representation is a reasonable approach recommended for practical applications. A corresponding algorithm can be formulated as follows.

Algorithm for determining the domain group

1. Given a function to be realized.

2. Depending on the values that each variable can take, determine the order of the domain group and possible combinations of orders of the domain subgroups the product of which is equal to the order of the initial domain group.

3. For each combination of subgroups, arrange the subgroups in the increasing value of their orders $g = \times_{i=1}^{n} g_i$, $g_0 \leq g_1 \leq \cdots \leq g_n$.

4. Start with the domain group represented as the direct product of subgroups of smallest orders and compute the Fourier representation on it. Determine the number of non-zero coefficients and the number of 1-bits to represent them.

5. If the obtained number of non-zero coefficients and 1-bits required to represent them is acceptable, then stop. Otherwise, replace the k subgroups of highest order by a subgroup of the corresponding order. Try different possibilities including non-Abelian subgroups.

6. After each replacement of subgroups compute the Fourier spectrum and read the number of zero coefficients and the number of 1-bits to represent them.

7. When the acceptable values of these parameters for the considered application are obtained stop the algorithm.

6. CLOSING REMARKS

Circuits derived from spectral representations have a regular structure, which is a feature usually required from present and expected in future computing technologies [9]. Further, such circuits are useful when frequent change of the functions implemented is required [5]. The optimization criteria follow from their structure and concern reduction of the number of non-zero coefficients and the number of 1-bits to represent them. In this paper, we point out that these parameters can be minimized by changing the structure of the domain group on which the function to be realized is considered. In particular, we suggest the usage of non-Abelian groups as the domain groups. For many functions, these groups can provide Fourier representations that are more compact in terms of the considered parameters than the Abelian groups.

REFERENCES

[1] Karpovsky, M.G., Stanković, R.S., Astola, J.T., *Spectral Logic and Its Application in the Design of Digital Devices*, Wiley, 2008.

[2] Miller, D.M., Dueck, G., Spectral techniques for reversible logic synthesis, *Proc. 6th International Symposium on Representations and Methodology of Future Computing Technologies*, March 2003, 56-62.

[3] Roziner, T. D., Karpovsky, M. G., Trachtenberg, L. A., "Fast Fourier transforms over finite groups by multiprocessor systems", *IEEE Trans. ASSP*, Vol. ASSP-38, 1990, pp. 226–240.

[4] Sasao, T., And-Exor expressions and their optimization, in Sasao, T., (ed.), *Logic Synthesis and Optimization*, Springer, 1993, 287-312.

[5] Sasao, T., *Memory Based Logic Synthesis*, Springer, 2011.

[6] Sasao, T., Index generating functions – minimization methods, *Proc. 47th Symp. on Multiple-Valued Logic*, Novi Sad, Serbia, May 22-24, 2017.

[7] Sasao, T., Fujita, M., *Representations of Discrete Functions*, Kluwer Academic Publishers, 1996.

[8] Stanković, R.S., Moraga, C., Astola, J.T., *Fourier Analysis on Finite Non-Abelian Groups with Applications in Signal Processing and System Design*, Wiley/IEEE Press, 2005.

[9] Stanković, R.S., Astola, J.T., Moraga, C., Fast Fourier transforms on finite groups as a method in synthesis for regularity, *Journal of Multiple-valued Logic and Soft Computing*, Vol. 23, No. 5-6, 2014, 463-483.

Information about the author:

Radomir S. Stanković – is a research professor at the Mathematical Institute of Sanu, Belgrade, Serbia. His research interests include spectral techniques, switching theory, multiple-valued logic, and signal processing.

Manuscript received on 27 June 2019