SLAC-PUB-639

COMMENTS ON "COMPUTATION OF THE FAST WALSH-FOURIER TRANSFORM" [1]

ABSTRACT

The matrix form of the Walsh functions as defined in the short note referred to can be generated by the modulo-2 product of two generating matrices, one of which is the natural binary code, and the other of which is the transpose of the bit-reversed form of the first. As a result, the coefficients of the Walsh transform occur in bit-reversed order.

By simply reordering the Walsh functions themselves to correspond to generation by the product of two such code matrices, neither or both in bit-reversed form, the Walsh coefficients occur in their natural order.

Index terms: code matrix, Walsh-Fourier transform, Walsh functions, Walsh matrix.

A useful relationship between the Walsh functions and the Cooley-Tukey algorithm for the fast Fourier transform has been established by Shanks [1], upon which Lechner has also commented [2]. As with the Cooley-Tukey algorithm, the coefficients of the Walsh-Fourier transform occur in bit-reversed order, unless the discrete data representing the function to be transformed are first rearranged in bit-reversed order [3].

This inconvenience can be circumvented by a very simple reordering of the Walsh functions themselves.

The writer has shown in previous correspondence that the binary Walsh functions in matrix form can be generated by modulo-2 multiplication of two simpler matrices representing some form of the reflected and/or natural binary code(s) [4].

By way of review, the basic binary Walsh matrix, possessing the property that each row has one more sign change than the preceding one (from 0 to 1 or from 1 to 0 in this case), is generated by any of the products:

$$\underset{\sim}{W} = \underset{\sim}{a}_i \odot \underset{\sim}{b}_{p-i}^t = \underset{\sim}{a}_{p-i} \odot \underset{\sim}{b}_i^t = \underset{\sim}{b}_i \odot \underset{\sim}{a}_{p-i}^t = \underset{\sim}{b}_{p-i} \odot \underset{\sim}{a}_i^t \qquad (1)$$

$$(i = 0,\ 1,\ \ldots\ p-1)$$

where $\underset{\sim}{a}_i$ denotes the reflected binary code and $\underset{\sim}{b}_i$ the natural binary code of p variables in the form of matrices of 2^p rows and p columns, and where $\underset{\sim}{a}_{p-i}$ and $\underset{\sim}{b}_{p-i}$ represent the respective matrices with the order of their columns reversed — or, in the contemporary terminology, in bit-reversed form. Examples are (21) and (22) of [4]. The superscript t denotes transposition of the matrix. The symbol $\odot$ denotes matrix multiplication, modulo-2.

Similarly, a modified binary Walsh matrix possessing a very simple and very useful recursive structure, perhaps first developed by Lechner [5] and shown in (17) of [4], is generated by either of the products:

$$\underset{\sim}{W}' = \underset{\sim}{b}_i \odot \underset{\sim}{b}_i^t = \underset{\sim}{b}_{p-i} \odot \underset{\sim}{b}_{p-i}^t \qquad (2)$$

in which neither of the code matrices is, or both are, in bit-reversed form.

Other forms of the Walsh matrix (some symmetrical and some unsymmetrical) can be generated by using other variations and combinations of the foregoing code matrices and matrices representing other codes. Still other forms can be obtained by independent reordering of the rows and/or columns of the Walsh matrix itself.

Now, the binary matrix representation of Fig. 1 of [1] (after replacing each 1 by 0 and each -1 by 1) is generated by either of the products:

$$\underset{\sim}{W}'' = \underset{\sim}{b}_i \odot \underset{\sim}{b}_{p-i}^t = \underset{\sim}{b}_{p-i} \odot \underset{\sim}{b}_i^t \qquad (3)$$

in which either one of the code matrices is in bit-reversed form, but not both. (This form of W is one presented, although not obtained from a product of code matrices, by Kaczmarz and Steinhaus [6].)

That this is so is also evident from (13) of [1], in which the k's are reversed in bit order with respect to the j's, and which could be written in the more compact matrix form:

$$\underset{\sim}{F} = \left[\underset{\sim}{U} - 2\underset{\sim}{W}''\right] \cdot \underset{\sim}{f} \qquad (4)$$

where $\underset{\sim}{f}$ is a column matrix of the discrete values f_n of the M-length real array whose transform is desired, and where $\underset{\sim}{F}$ is a column matrix of the transform coefficients F_m. Subtracting twice the binary Walsh matrix from the unit matrix $\underset{\sim}{U}$, all of whose elements are 1, effects the necessary transformation $0 \rightarrow 1$, $1 \rightarrow -1$, corresponding to (11) of [4]. The symbol $\cdot$ denotes ordinary matrix multiplication.

Since only one of the generating code matrices in either product in (3) is in bit-reversed form, it is not surprising that the coefficients F_m in (13) of [1] occur in bit-reversed form.

If the Walsh functions in Fig. 1 of [1] are rearranged in the order given by (2), corresponding to (16) of [4], then (4) can be rewritten as:

$$\underset{\sim}{F}' = \left[\underset{\sim}{U} - 2\underset{\sim}{W}'\right] \circ \underset{\sim}{f} \tag{5}$$

Then (12) and (13) of [1] could also be rewritten as:

$$\text{wal}'\,(j_2, j_1, j_0;\, k_2, k_1, k_0) = (-1)^{j_2 k_2}\,(-1)^{j_1 k_1}\,(-1)^{j_0 k_0} \tag{6}$$

$$F'\,(j_2, j_1, j_0) = \sum_{k_2=0}^{1} (-1)^{j_2 k_2} \sum_{k_1=0}^{1} (-1)^{j_1 k_1} \sum_{k_0=0}^{1} (-1)^{j_0 k_0}\, f(k_2, k_1, k_0) \tag{7}$$

$$(j_2, j_1, j_0 = 0 \text{ or } 1; \quad k_2, k_1, k_0 = 0 \text{ or } 1)$$

and the coefficients of the transform would then occur in the natural order instead of bit-reversed order.

Thus, in machine computation as described in [3], the computation may be done "in place," yet requires no "shuffling" of either the initial data sequence or the resulting spectrum.

The signal flow graph for (5) or (7) differs from that in Fig. 2 of [1] only in that the coefficients F_m at the right are now in their natural order, i.e., the same order as the data f_n at the left.

Keith W. Henderson
(Senior Member, IEEE)
Stanford Linear Accelerator Center*
Stanford University
Stanford, California 94305

REFERENCES

1. J. L. Shanks, "Computation of the fast Walsh-Fourier transform," IEEE Trans. Computers (Short Notes), vol. C-18, pp. 457-459; May 1969.
2. R. J. Lechner, "Comment on 'Computation of the fast Walsh-Fourier transform'," IEEE Trans. Computers (Correspondence), vol. C-19, p. 174; February 1970.
3. W. T. Cochran <u>et al.</u>, "What is the fast Fourier transform?" Proc. IEEE, vol. 55, pp. 1664-1674; October 1967.
4. K. W. Henderson, "Some notes on the Walsh functions," IEEE Trans. Electronic Computers (Correspondence), vol. EC-13, pp. 50-52; February 1964.
5. R. Lechner, "A transform theory for functions of binary variables," in "Theory of Switching," Harvard Computation Lab., Cambridge, Mass., Progress Rept. No. BL-30, Sec. X, pp. 1-37; November 1961.
6. S. Kaczmarz and H. Steinhaus, "Theorie der Orthogonalreihen," Chelsea Publishing Co., New York, N. Y., 2nd ed., pp. 125-134; 1951.

FOOTNOTE

* Operated under contract with the U. S. Atomic Energy Commission.