The purpose of this note is to point out that the state assignment techniques for synchronous sequential machines are equally good for the BSD's as well. The ordered pattern of BSD's makes it easier to choose an economical assignment, irrespective of the techniques used. However, this realization has to be compared with the SR detector realization.

Sureshchander
Dept. of Elec. Engrg.
Pant College of Technology
Pantnagar (Nainital), U. P.
India

Comment on “Computation of the Fast Walsh-Fourier Transform”

J. L. Shanks has given an algorithm for computing the discrete Walsh transform (abstract Fourier transform) of a sampled periodic function whose domain of definition is the set of integers modulo $2^n$. An algorithm of the same efficiency, using a much simpler notation, was given for the abstract Fourier transform in my correspondence published in 1963 in this TRANSACTIONS. This transform has an identical matrix representation; the only difference is that the function domain is represented (for computation purposes) by binary coded representations of the integers from 0 to $2^n-1$. These binary $n$-tuples form a group under vector addition, modulo two.

My correspondence noted that the Fourier transform (or Hadamard) matrix (and several other matrices of interest in the canonical expansion of functions of binary arguments) can be represented as the $n$th Kronecker power $[A]^n$ of a 2 by 2 matrix, $A$. It also described how to compute the transform by direct application of the following expansion theorem:

$$[A]^n = \prod_{k=0}^{n-1} (I^{2^n-1} \times A \times I^n).$$

Here $\prod$ represents the ordinary matrix product, $\times$ represents the Kronecker product, and $I^n$ represents a unit matrix of dimension $2^n$. By this theorem, the orthogonal transform matrix can be decomposed into the ordinary matrix product on $n$ commutative factors. It is easy to verify directly that multiplication by the factor $(I^{2^n-1} \times A \times I^n)$ corresponds to one stage of Shanks' signal flow graph topology for computing the fast Fourier transform (i.e., it produces the vector sum and difference of two $2^n$-tuples). The partial product $A \times I^n$ has the same form as the original $2 \times 2$ matrix $A$ with each scalar entry replaced by itself times a unit matrix of dimension $2^n$; the product $I^{(2^n-1)} \times A \times I^n$ is a quasidiagonal matrix with $2^{n-1}$ copies of $(A \times I^n)$ along its main diagonal.

The expansion of a domain index $i$ and an orthogonal function index $m$ as binary $n$-tuples permits a direct evaluation of the fast Fourier transform kernel function (or Walsh function) $\text{Wal}(m, i)$. Let the binary codes for $i$ and $m$ be $(x_1, x_2, \ldots, x_n)$ and a bit-reversed copy of $(w_1, w_2, \ldots, w_n)$, respectively. Then $\text{Wal}(m, i) = (-1)^v$ where $v = x_1w_1 + x_2w_2 + \ldots + x_nw_n$ is the ordinary inner product of two $n$-tuples. Since $(-1)^v$ changes sign if $v$ is odd or even, this inner product may be taken modulo two. Therefore, to compute $\text{Wal}(m, i)$, it is sufficient to take the parity function (modulo two sum) of the component-wise logical product of the two binary $n$-tuples representing $m$ and $i$.

Parity evaluation is an implicit but inaccessible function of most modern computers. Perhaps its utility in fast Fourier transforms will motivate more computer manufacturers to make it available as a machine instruction.

Robert J. Lechner
Sylvania Electronic Systems
Needham Heights, Mass. 02194

IEEE TRANSACTIONS ON COMPUTERS, FEBRUARY 1970

---