

UM ALGORITMO BIFUNCIONAL PARA AVALIAÇÃO DOS ESPECTROS DE HADAMARD E HARTLEY

R.J. de Sobral Cintra, H.M. de Oliveira, R.M. Campello de Souza

CODEC - Grupo de Pesquisa em Comunicações
Departamento de Eletrônica e Sistemas
Universidade Federal de Pernambuco, Recife

Endereço eletrônico: rjsc@ee.ufpe.br, hmo@npd.ufpe.br, ricardo@npd.ufpe.br

RESUMO

Neste artigo, um novo algoritmo para o cálculo das transformadas discretas de Hartley e de Hadamard é apresentado, unindo as idéias propostas por Hsu-Wu e por Guo. Uma implementação em *hardware* utilizando somadores em paralelo é apresentada.

1. INTRODUÇÃO

Transformadas discretas definidas em corpos finitos ou infinitos ocupam um papel de destaque em engenharia. Neste artigo, são focalizadas duas transformadas em particular: a transformada de Hadamard e a transformada discreta de Hartley.

A transformada de Hadamard é uma ferramenta clássica utilizada principalmente em codificação algébrica, reconhecimento de padrões, análise espectral [13] e processamento de imagens [10].

A transformada discreta de Hartley, versão discretizada da transformada integral introduzida em 1942 por R. V. L. Hartley, foi definida em 1983 por R. Bracewell e encontrou aplicações iniciais no cálculo eficiente da transformada de Fourier [2] e em óptica (interferometria) [14].

Mais recentemente, no final da década de '90, a transformada de Hartley vem sendo usada em diversas aplicações, tais como:

- compressão de imagens biomédicas [9],
- sistemas de comunicação utilizando multiportadora (OFDM) e acesso múltiplo (CDMA) [1],
- modems ADSL [11, 15].

O crescente interesse na transformada de Hartley, impulsionado pela definição da transformada de Hartley de Corpo Finito [3], levou à introdução de recentes e importantes resultados no projeto de sistemas de multiplexação digital, de sistemas de múltiplo acesso e no estabelecimento de seqüências de espalhamento espectral [5].

Em 1987, Hsu e Wu introduziram um algoritmo para o cálculo da transformada de Walsh-Hadamard conjuntamente com o cálculo da transformada de Hartley [12].

Trabalhos recentes [7, 6] introduziram novos algoritmos rápidos para a transformada discreta de Hartley que atingem complexidades multiplicativas mínimas. Este cenário levou a uma revisão do algoritmo proposto por Hsu-Wu, visando otimizá-lo.

Outro trabalho recente realizado por Guo [11] introduz um projeto eficiente para a transformada discreta de Hartley através de uma implementação que se utiliza de adições paralelas.

Assim, no presente trabalho, o algoritmo de Hsu-Wu é revisado à luz dos novos algoritmos rápidos e é delineada uma implementação utilizando a técnica introduzida por Guo.

O espírito do algoritmo proposto é obter uma transformação que leve vetores do domínio de Hadamard ao domínio de Hartley e, assim, ter um algoritmo bifuncional. Este algoritmo é particularmente atrativo para implementação em CI's dedicados de baixo custo. O usuário tem opção de acessar diretamente a transformada de Hadamard ou a transformada de Hartley no mesmo *chip*, a depender da aplicação.

2. O ALGORITMO

A transformada discreta de Hartley de comprimento N relaciona dois vetores $\mathbf{v} \xleftrightarrow{\mathcal{H}} \mathbf{V}$ e é definida por [2]

$$V_k \triangleq \sum_{i=0}^{N-1} v_i \text{cas} \left(\frac{2\pi ki}{N} \right), \quad k = 0, 1, \dots, N-1,$$

onde $\text{cas } t \triangleq \cos t + \text{sen } t$. Esta transformada pode ser escrita em forma matricial da seguinte maneira:

$$\mathbf{V} = \mathbf{H}_N \mathbf{v},$$

onde \mathbf{H}_N é a matriz de Hartley, cujos elementos $h_{k,i}$ são da forma $\text{cas} \left(\frac{2\pi ki}{N} \right)$.

A transformada de Hadamard é dada por [13]

$$\Lambda = \mathbf{Had}_N \mathbf{v},$$

onde \mathbf{Had}_N é a matriz de Hadamard de comprimento N .

Para que essa transformada se preste para o cálculo da transformada de Hartley, deve-se ter

$$\mathbf{V} = \mathbb{H}_N \mathbf{Had}_N \mathbf{v},$$

onde $\mathbf{H}_N = \mathbb{H}_N \mathbf{Had}_N$. A matriz \mathbb{H} realiza o “ajuste” necessário à transformada de Hadamard para que ela se torne uma transformada de Hartley. É fácil verificar que

$$\mathbb{H}_N = \mathbf{H}_N (\mathbf{Had}_N)^T. \quad (1)$$

Uma restrição quanto aos comprimentos de transformada a serem considerados pode ser identificada, pois para todo comprimento N existe uma transformada de Hartley associada. Fato que não é verificado para as transformadas de Hadamard, conforme o seguinte teorema.

Teorema 1 (Ordem da Matriz de Hadamard) *Se uma matriz de Hadamard de ordem N existir, então N é igual a 1, 2 ou é múltiplo de 4 [13].*

Neste artigo, são consideradas matrizes cujos ordens são potência de 2 (construção clássica de Sylvester). Para essas matrizes de Hadamard do tipo de Sylvester, tem-se que $(\mathbf{Had}_N)^T = \mathbf{Had}_N$, o que nos permite escrever $\mathbb{H}_N = \mathbf{H}_N \cdot \mathbf{Had}_N$.

O método utilizado para obter o algoritmo rápido será ilustrado através de um exemplo. Assim, seja a transformada discreta de Hartley de comprimento 8, cuja matriz de transformação é descrita por [2]

$$\mathbf{H}_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \sqrt{2} & 1 & & - & -\sqrt{2} & & \\ 1 & 1 & - & - & 1 & 1 & - & - \\ 1 & & - & \sqrt{2} & & & 1 & -\sqrt{2} \\ 1 & - & 1 & - & 1 & - & 1 & - \\ 1 & -\sqrt{2} & 1 & & - & \sqrt{2} & & \\ 1 & - & - & 1 & 1 & - & - & 1 \\ 1 & & - & -\sqrt{2} & & & 1 & \sqrt{2} \end{bmatrix}$$

e a transformada de Hadamard de comprimento 8 que é dada por

$$\mathbf{Had}_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & - & 1 & - & 1 & - & 1 & - \\ 1 & 1 & - & - & 1 & 1 & - & - \\ 1 & - & - & 1 & 1 & - & - & 1 \\ 1 & 1 & 1 & 1 & - & - & - & - \\ 1 & - & 1 & - & - & 1 & - & 1 \\ 1 & 1 & - & - & - & - & 1 & 1 \\ 1 & - & - & 1 & - & 1 & 1 & - \end{bmatrix},$$

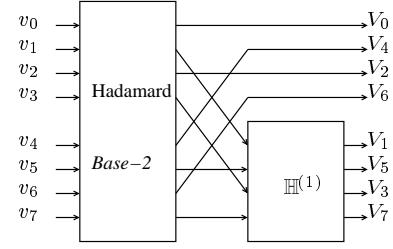


Figura 1: Diagrama do algoritmo para $N = 8$.

onde “-” representa -1 e os espaços em branco contêm o elemento zero.

Utilizando a Equação 1, encontra-se facilmente a matriz que se segue:

$$\mathbb{H}_8 = \begin{bmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & \frac{1}{2} + \frac{\sqrt{2}}{4} & \frac{1}{2} - \frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} & & \\ & & \frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} & \frac{1}{2} - \frac{\sqrt{2}}{4} & \frac{1}{2} + \frac{\sqrt{2}}{4} & & \\ & & & & \frac{1}{2} - \frac{\sqrt{2}}{4} & \frac{1}{2} + \frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} \\ & & & & 1 & & & \\ & & & & & & -\frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} & \frac{1}{2} + \frac{\sqrt{2}}{4} & \frac{1}{2} - \frac{\sqrt{2}}{4} \end{bmatrix}. \quad (2)$$

Realizando-se uma permutação na matriz \mathbb{H}_8 de modo a trocar a posição das linhas segundo uma reversão dos bits do número da linha em binário, obtém-se a seguinte matriz:

$$\tilde{\mathbb{H}}_8 = \begin{bmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & 1 & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & \frac{1}{2} + \frac{\sqrt{2}}{4} & \frac{1}{2} - \frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} \\ & & & & \frac{1}{2} - \frac{\sqrt{2}}{4} & \frac{1}{2} + \frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} \\ & & & & \frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} & \frac{1}{2} - \frac{\sqrt{2}}{4} & \frac{1}{2} + \frac{\sqrt{2}}{4} \\ & & & & -\frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} & \frac{1}{2} + \frac{\sqrt{2}}{4} & \frac{1}{2} - \frac{\sqrt{2}}{4} \end{bmatrix}. \quad (3)$$

Observa-se que a matriz $\tilde{\mathbb{H}}$ tem a seguinte descrição: $\text{diag}(\mathbf{I}_4, \tilde{\mathbb{H}}^{(1)})$, i.e., uma matriz diagonal em que os elementos da diagonal são matrizes. Este comportamento é verificado para qualquer comprimento potência de dois [8]. Para comprimento de bloco $N = 2^m$, tem-se a forma geral $\tilde{\mathbb{H}}_N = \text{diag}(\mathbf{I}_4, \tilde{\mathbb{H}}^{(1)}, \dots, \tilde{\mathbb{H}}^{(m-2)})$.

A Figura 1 apresenta um diagrama esquemático do algoritmo. É necessário agora desenvolver um algoritmo rápido

para realizar o cálculo da matriz $\tilde{\mathbb{H}}^{(1)}$, dada por

$$\begin{bmatrix} \frac{1}{2} + \frac{\sqrt{2}}{4} & \frac{1}{2} - \frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} \\ \frac{1}{2} - \frac{\sqrt{2}}{4} & \frac{1}{2} + \frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} \\ \frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} & \frac{1}{2} - \frac{\sqrt{2}}{4} & \frac{1}{2} + \frac{\sqrt{2}}{4} \\ -\frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} & \frac{1}{2} + \frac{\sqrt{2}}{4} & \frac{1}{2} - \frac{\sqrt{2}}{4} \end{bmatrix}.$$

Utilizando o mesmo procedimento descrito em [7], pode-se fatorar a matriz $\tilde{\mathbb{H}}^{(1)}$ na forma:

$$\begin{bmatrix} 1 & 1 \\ 1 & - \\ & 1 & 1 \\ & 1 & - \end{bmatrix} \cdot \text{diag} \left(\frac{1}{2}, \frac{\sqrt{2}}{4}, \frac{1}{2}, \frac{\sqrt{2}}{4} \right) \cdot \begin{bmatrix} 1 & 1 \\ & 1 & 1 \\ & & 1 & 1 \\ 1 & & & - \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 \\ 1 & - \\ & 1 & 1 \\ & & 1 & - \end{bmatrix}. \quad (4)$$

Esta fatoração reduz a complexidade aritmética de $\tilde{\mathbb{H}}^{(1)}$ para 2 multiplicações, 2 deslocamentos e 10 adições. A Figura 2 exibe um diagrama com esta implementação.

Numa implementação em arquitetura de 8 bits, o termo $\sqrt{2}/4$ é truncado da seguinte forma:

$$\frac{\sqrt{2}}{4} \approx 2^{-2} + 2^{-4} + 2^{-5} + 2^{-7} = 0,3515625.$$

Isto corresponde a um erro de cálculo de

$$\frac{\sqrt{2}/4 - 0,3515625}{\sqrt{2}/4} \approx 0,5631\%,$$

aproximadamente 1,7847 vez menor do que o erro oferecido pelo algoritmo anterior descrito na literatura [12]. O termo $1/2$ não contribui com erros, pois é um simples deslocamento à direita nos registradores. Esta análise de erro é mostrada na Tabela 1.

Tabela 1: Análise do erro cometido pelo algoritmo em arquitetura de 8 bits.

Número	Hsu-Wu	Proposto
0,854	0,254%	-
0,146	1,005%	-
1/2	-	0
$\sqrt{2}/4$	-	0,5631%

Complexidade Computacional. A complexidade aditiva do algoritmo completo é limitada inferiormente pelo número de adições necessárias para a transformada de Hadamard. Para comprimentos que são potência de dois, esse limite é $N \log_2 N$, que é obtido através do uso de algoritmos de dizimação base-2. Assim, a complexidade aritmética desse

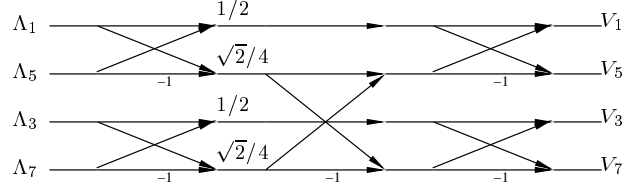


Figura 2: Diagrama do algoritmo para o cálculo da matriz $\tilde{\mathbb{H}}^{(1)}$

algoritmo é:

$$M_R(N) = \sum_{l=1}^{m-2} \tilde{M}_R \left(2^{(l+1)} \right), \quad (5)$$

$$A_R(N) = \underbrace{N \log_2 N}_{\text{Limite inferior}} + \sum_{l=1}^{m-2} \tilde{A}_R \left(2^{(l+1)} \right), \quad (6)$$

onde \tilde{M}_R e \tilde{A}_R são, respectivamente, as complexidades multiplicativa e aditiva de cada matriz $\tilde{\mathbb{H}}^{(l)}$.

3. IMPLEMENTAÇÃO POR SOMADORES

Uma implementação alternativa à exibida na Figura 2 para o cálculo da matriz $\tilde{\mathbb{H}}^{(1)}$ pode ser feita através do uso de somadores (*adders* e *carry saver adders* - CSA) [11]. O método baseia-se na observação de que

$$\begin{bmatrix} V_0 \\ V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} x & y & z & w \\ w & x & y & z \\ z & w & x & y \\ y & z & w & x \end{bmatrix} \begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix}, \quad (7)$$

é equivalente a

$$\begin{bmatrix} V_0 \\ V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} v_0 & v_1 & v_2 & v_3 \\ v_1 & v_2 & v_3 & v_0 \\ v_2 & v_3 & v_0 & v_1 \\ v_3 & v_0 & v_1 & v_2 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}. \quad (8)$$

No exemplo em questão, a matriz $\tilde{\mathbb{H}}^{(1)}$ possui os elementos $\left\{ \frac{1}{2} + \frac{\sqrt{2}}{4}, \frac{1}{2} - \frac{\sqrt{2}}{4}, \frac{\sqrt{2}}{4}, -\frac{\sqrt{2}}{4} \right\}$. Seguindo então o procedimento sugerido pela Equação 8, os elementos da matriz de transformação são invertidos. Antes disso, os elementos de $\tilde{\mathbb{H}}^{(1)}$ são representados em binário, visando a implementação final. Por simplicidade, usa-se nesse exemplo uma aritmética binária com palavras de 4 bits. Assim,

$$\begin{bmatrix} \frac{1}{2} + \frac{\sqrt{2}}{4} \\ \frac{1}{2} - \frac{\sqrt{2}}{4} \\ \frac{\sqrt{2}}{4} \\ -\frac{\sqrt{2}}{4} \end{bmatrix} \approx \begin{bmatrix} 0,85355 \\ 0,14644 \\ 0,35355 \\ -0,35355 \end{bmatrix} = \begin{bmatrix} 0,1101_{(2)} \\ 0,0010_{(2)} \\ 0,0101_{(2)} \\ -0,0101_{(2)} \end{bmatrix} + \text{erro}. \quad (9)$$

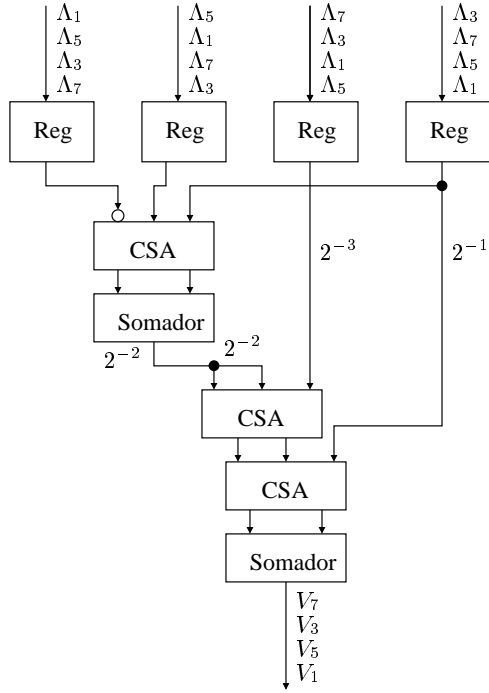


Figura 3: Implementação da matriz $\tilde{\mathbb{H}}^{(1)}$ via somadores.

Dessa maneira, tem-se que:

$$\begin{aligned}
 V_1 &= 0,1101_{(2)}\Lambda_1 + 0,0010_{(2)}\Lambda_5 + 0,0101_{(2)}\Lambda_3 - \\
 &\quad 0,0101_{(2)}\Lambda_7 \\
 &= 2^{-1}\Lambda_1 + 2^{-2}(\Lambda_1 + \Lambda_3 - \Lambda_7) + 2^{-3}\Lambda_5 + \\
 &\quad 2^{-4}(\Lambda_1 + \Lambda_3 - \Lambda_7) \\
 V_5 &= 0,1101_{(2)}\Lambda_5 + 0,0010_{(2)}\Lambda_1 + 0,0101_{(2)}\Lambda_7 - \\
 &\quad 0,0101_{(2)}\Lambda_3 \\
 &= 2^{-1}\Lambda_1 + 2^{-2}(\Lambda_5 + \Lambda_7 - \Lambda_3) + 2^{-3}\Lambda_1 + \\
 &\quad 2^{-4}(\Lambda_5 + \Lambda_7 - \Lambda_3) \\
 V_3 &= 0,1101_{(2)}\Lambda_7 + 0,0010_{(2)}\Lambda_3 + 0,0101_{(2)}\Lambda_1 - \\
 &\quad 0,0101_{(2)}\Lambda_5 \\
 &= 2^{-1}\Lambda_1 + 2^{-2}(\Lambda_7 + \Lambda_1 - \Lambda_5) + 2^{-3}\Lambda_3 + \\
 &\quad 2^{-4}(\Lambda_7 + \Lambda_1 - \Lambda_5) \\
 V_7 &= 0,1101_{(2)}\Lambda_3 + 0,0010_{(2)}\Lambda_7 + 0,0101_{(2)}\Lambda_5 - \\
 &\quad 0,0101_{(2)}\Lambda_1 \\
 &= 2^{-1}\Lambda_1 + 2^{-2}(\Lambda_3 + \Lambda_5 - \Lambda_1) + 2^{-3}\Lambda_7 + \\
 &\quad 2^{-4}(\Lambda_3 + \Lambda_5 - \Lambda_1).
 \end{aligned} \tag{10}$$

As equações acima reafirmam o fato de que todas as componentes espectrais relativas à matriz $\tilde{\mathbb{H}}^{(1)}$ podem ser calculadas usando a mesma arquitetura. A Figura 3 exibe a implementação do cálculo da transformada.

Um diagrama esquemático geral para o algoritmo é apre-

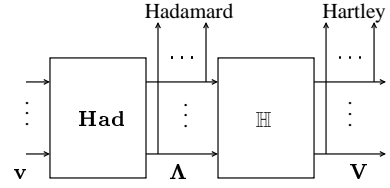


Figura 4: Diagrama esquemático do algoritmo geral.

sentado na Figura 4, enfatizando que ambas transformadas são disponibilizadas simultaneamente.

4. CONCLUSÕES

Foi apresentado um novo algoritmo que computa a transformada de Hadamard e a de Hartley simultaneamente. Limitações sobre os comprimentos de bloco para os quais este algoritmo pode ser aplicado foram evidenciadas. Uma implementação proposta revelou uma vantagem na análise de erro em relação ao já existente na literatura. Foi também sugerida uma implementação em *hardware* por meio de somadores paralelos. O cálculo da transformada inversa é trivial já que a transformada de Hartley é involucionária. Ademais cada uma das matrizes envolvidas, **Had** e **H**, também é involucionária (a menos de um possível ajuste de escala).

5. REFERÊNCIAS

- [1] BOGUCA, H., Effective Implementation of the OFDM/CDMA Base Station Transmitter using joint FHT and IFFT, **Proc. IEEE Workshop Signal Process. Adv. Wireless Commun.**, p.162–165, 1999.
- [2] BRACEWELL, R. N., **The Hartley Transform**, Oxford, 1986.
- [3] CAMPELLO DE SOUZA, R. M.; DE OLIVEIRA, H. M.; KAUFFMAN, A. N.; PASCHOAL, A. J. A., **Trigonometry in Finite Fields and a New Hartley Transform**, In: Proceedings of the 1998 IEEE Intern. Symp. on Info. Theory, p.293, ago. 1998.
- [4] DE OLIVEIRA, H. M.; CAMPELLO DE SOUZA, R. M., **Orthogonal Multilevel Spreading Sequence Design**, In: Coding Communications and Broadcasting, Farrell, P. G.; Darnell, M.; Honary, B., (Eds.), Hertfordshire, Inglaterra, Research Studies Press, John Wiley, p.291–301, 2000.
- [5] DE OLIVEIRA, H. M.; CAMPELLO DE SOUZA, R. M.; KAUFFMAN, A. N., **Efficient Multiples for Band-limited Channels: Galois Division Multiple Access**, In: Proceedings of the 1999 Workshop on Coding and Cryptography, WCC-99, Paris, p.235–241, jan. 1999.

- [6] DE OLIVEIRA, H. M.; DE SOBRAL CINTRA, R. J.; CAMPELLO DE SOUZA, R. M., **Multilayer Hadamard Decomposition of Discrete Hartley Transforms**, In: XVIII Simpósio Brasileiro de Telecomunicações, Gramado, RS, set. 2000.
- [7] ———, **A Factorization Scheme for Some Discrete Hartley Transform Matrices**, In: ICSECT 2001, International Conference on System Engineering, Communications and Information Technologies, Magas, B., (Ed.), Punta Arenas, Chile, Universidad de Magallanes, abr. 2001.
- [8] DE SOBRAL CINTRA, R. J., **Transformada Rápida de Hartley: Novas Fatorações e Algoritmo Aritmético**, Recife, 2001, Dissertação de Mestrado, Universidade Federal de Pernambuco.
- [9] DULEBA, I., Hartley Transform in Compression of Medical Ultrasonic Images, **Proceedings of the 10th International Conference on Image Analysis and Processing**, 1998.
- [10] GONZALEZ, R. C.; WOODS, R. E., **Digital Image Processing**, 3. ed., Addison-Wesley, 1992.
- [11] GUO, J.-I., An Efficient Design for One-Dimensional Discrete Hartley Transform Using Parallel Additions, **IEEE Transactions on Signal Processing**, v.48, 2000.
- [12] HSU, C.-Y.; WU, J.-L., The Walsh-Hadamard/discrete Hartley transform, **Int. J. Electronics**, p.747–755, v.62, 1987.
- [13] LIDL, R.; PILZ, G., **Applied Abstract Algebra**, Springer, 1998, Cap. 7, p.413–416.
- [14] VILLASENOR, J. D.; BRACEWELL, R., Optical Phase Obtained by Analogue Hartley Transform, **Nature**, p.735–737, v.330, 1987.
- [15] WANG, C. L.; CHANG, C. H., A Novel DHT-based FFT/IFFT Processor for ADSL Transceivers, **Proc. IEEE Int. Symp. Circuits Syst.**, p.51–54, v.1, 1999.