A RESEARCH ON FAST HADAMARD TRANSFORM (FHT) DIGITAL SYSTEMS

Chen Anshi, Li Di, Zhou Renzhong
Beijing Institute of Technology, P. O. Box 327
Beijing 100081, China

ABSTRACT

A $2^m \times 2^m$ standard Hadamard matrix can be written as the *m*th power of one matrix. Based on this principle, the paper has presented a hardware digital system to do Fast Hadamard Transform (FHT), which is composed of one arithmetic unit, four random-access memories, two address generators, π_1 and π_2, and a control unit. Experiments show that the hardware system can finish FHT within 20ms for a picture of 4095 elements.

INTRODUCTION

Hadamard Transform has found applications in spectrum measurement and infrared imaging systems to replace the conventional sweeping measurement method such as spectrodiffractometers and vidicons. The procedure of Hadamard transform can be simply described as below:

1. Light intensities of the picture elements are multiplied by Hadamard matrix, which is done by an optical encoding device.

2. The values of above multiplication results is measured with a light detector, e. g., a photodiode, to get a set of data.

3. The datum set is multiplied by inverse Hadamard matrix to reconstruct the orignal picture.

In such a way, we can obtain a reconstructed picture of higher accuracy than in sweeping measurement method. The multiplication by Hadamard matrix of inverse Hadamard matrix said above means forward Hadamard transformation or inverse Hadamard transformation respectively. Forward Hadamard tramformation is done through an optical system which will not be discussed here. Inverse Hadamard transformation is generally done by an electronic system. Computers and software are often used to do inverse transformation. A computer may be fast enough for light spectrum measurements, but will be too slow for real-time image detection. For example, even high speed computer can not complete inverse Hadamard transform within 20ms for a picture of 63×65 elements. For this reason, it is required to have a hardware digital system to do fast (inverse) Hadamard transformation.

In this paper, the expnsion of standard Hadamard matrix is introduced at first. And then the FHT digital system based on Hadamard matrix expansion is described. At last the experimental results are shown.

HADAMARD MATRIX EXPANSION

A 2×2 standard Hadamard matrix, H_2, is defined as

$$\mathbf{H}_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Let $N = 2^m$, then a $2^m \times 2^m$ standard Hadamard matrix $\mathbf{H}_N$, is defined as

$$\mathbf{H}_N = \underbrace{\mathbf{H}_2 \otimes \mathbf{H}_2 \otimes \cdots \otimes \mathbf{H}_2}_{\text{for } m \text{ times}}$$

where symbol $\otimes$ means Kronecker multiplication. For example, $m=2$, $N=4$, then the Hadamard matrix, $\mathbf{H}_4$ is

$$\mathbf{H}_4 = \mathbf{H}_2 \otimes \mathbf{H}_2 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix},$$

Now we define a new matrix, $\mathbf{G}_4$, as below

$$\mathbf{G}_4 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & -1 \end{bmatrix}$$

multiplying $\mathbf{G}_4$ by itself, then yields

$$\mathbf{G}_4^2 = \mathbf{H}_4$$

In words, it is said that a Hadarmard matrix can be written as the multiplication of two matrices, which is called matrix expansion. It can be proved that Hadamard matix expansion exsists for $N=2^m (m=1, 2, 3, \cdots)$. Thus it follows

$$\mathbf{H}_N = \left[\underbrace{\begin{matrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & & & & \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 1 \end{matrix}}_{N/2} \quad \underbrace{\begin{matrix} 1 & 0 & 0 & \cdots & 0 \\ -1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ 0 & 0 & -1 & \cdots & 0 \\ \vdots & & & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & -1 \end{matrix}}_{N/2} \right]^m = \mathbf{G}_N{}^m$$

Multipying a given data vector, $\mathbf{f} = [f_0\, f_1\, f_2 \cdots f_{N-1}]^T$, with Hadamard matrix, $\mathbf{H}_N$, gives another data vector, $\mathbf{F} = [F_0\, F_1\, F_2 \cdots F_{N-1}]^T$. This procedure is called Hadamard Transform, i. e.

$$\mathbf{F} = \mathbf{H}_N \mathbf{f}$$

Since $\mathbf{H}_N = \mathbf{G}_N{}^m$, thus

$$\begin{aligned} \mathbf{F} &= \mathbf{G}_N{}^m \mathbf{f} \\ &= \underbrace{\mathbf{G}_N \mathbf{G}_N \cdots \mathbf{G}_N}_{m \text{ steps}} \mathbf{f} \end{aligned}$$

This equation reveans that Hadamard tramsformation can be divided into m steps of matrix multiplications in the same way. This is important to its application in fast Hadamard transform digital system presented in this paper.

To know more details about matrix expansion application in FHT system, we give an example of $m=2$ ($N=4$). The transformation is divided into two steps:

Step 1 :

$$\mathbf{G}_4 \mathbf{f} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \end{bmatrix} = \begin{bmatrix} f_0 + f_2 \\ f_0 - f_2 \\ f_1 + f_3 \\ f_1 - f_3 \end{bmatrix} = \begin{bmatrix} f_0' \\ f_1' \\ f_2' \\ f_3' \end{bmatrix} = \mathbf{f}'$$

Step2 :

$$\mathbf{G}_4 \mathbf{f}' = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} f_0' \\ f_1' \\ f_2' \\ f_3' \end{bmatrix} = \begin{bmatrix} f_0' + f_2' \\ f_0' - f_2' \\ f_1' + f_3' \\ f_1' - f_3' \end{bmatrix} = \begin{bmatrix} \mathrm{F}_0 \\ \mathrm{F}_1 \\ \mathrm{F}_2 \\ \mathrm{F}_3 \end{bmatrix} = \mathbf{F}$$

The flow diagram which shows this transrorm procedure is shown in Fig. 1.

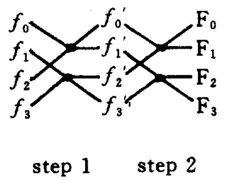
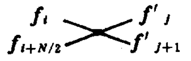


Fig. 1 FHT flow diagram ($m=2$)

Obviously, the butterfly algorithm used in FFT is also suited to FHT. So the total number of operations is 8 additions/subtractions. In every step, the loading of data, computing, and saving of data are the same, which make the design and construction of digital system much easier. In general, m steps are needed to finish FHT for $N=2^m$, where $m = 1, 2, \cdots$. The total number of operations is $m \times 2^m$ additions/subtractions. The butterfly algorithm used here is shown in Fig. 2.



$$j=2i \quad (i=0, 1, 2, \cdots, N/2-1)$$

Fig. 2 The butterfly algorithm

HARDWARE SCHEME

A FHT hardwave scheme for finishing matrix expansion is illustrated in Fig. 3. The orignal data $\mathbf{f}$, is

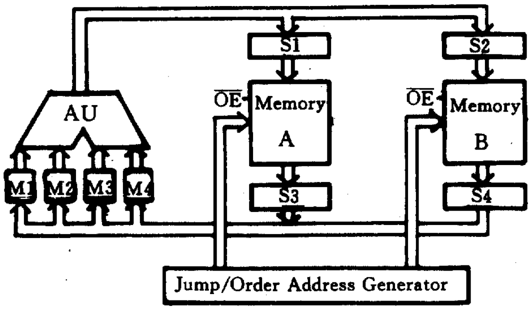


Fig. 3 The FHT hardware scheme

stored in memory A, and the first step operation result will be put into memory B. In this step, memory A is called source memory because it transmits data, and memory B is call receiver memory because it receives data. An arithmetic unit (AU) is used to do butterfly algorithm operations. The source memory A is given the addresses as below:

0, $N/2$, 1, $N/2+1$,2, ⋯, $N/2-1$, $N-1$, 0, 0.

They are produced by a Jump address generator. The receiver memory B is given the addresses as below:

0, 0, 0, 1, 2, 3, ⋯, $N-1$.

They are produced by an Order address generator. In this step, the tristate drivers S2 and S3 are on, S1 and S4 are off (in high impedence state). $\overline{\text{OE}}$ of A is low, and $\overline{\text{OE}}$ of B is high. Data in source memory A are transmitted through S3, latches, AU, and S2 to receiver memory B. The latches M1, M2, M3, and M4, are controlled by a shift register to read the data bus circulatively. After the start of first step, the operations are

1. (A: address 0) → M1;
2. (A: address $N/2$) → M3;
3. (A: address 1) → M2;
   (M1+M3) → B: address 0;
4. (A: address $N/2+1$) → M4;
   (M1−M3) → B: address 1;
5. (A: address 2) → M1;
   (M2+M4) → B: address 2;
6. (A: address $N/2+2$) → M3;
   (M2−M4) → B: address 3;
7. (A: address 3) → M2;
   (M1+M3) → B: address 4;

⋮

N. (A: address $N-1$) → M4;
   (M1−M3) → B: address $N-3$;
$N+1$. (M2+M4) → B: address $N-2$;
$N+2$. (M2−M4) → B: address $N-1$;

where "(A: address 0)→M1" means the datum at address 0 of memory A is loaded to latch M1, and "(M1+M3)→B: address 0" means that the data in latch M1 and latch M3 are added and stored to memory B at address 0 . $N+2$ cycles are used to finish the first step operations. In the second step, the working mode of memory A and B exchange with each other. Memory B becomes source memory, and A becomes receiver memory. The data in memory A now are useless. Tristate drivers S1 and S4 are on, S2 and S3 are off (in high impedence state). $\overline{\text{OE}}$ of B is low ,and $\overline{\text{OE}}$ of A is high . Memory B is switched to Jump address generator and memory A is switched to Order address generator. Operations in the second step are the same as those in the first step. In the third step, the working mode of memory A and B is altered again. m steps are needed to finish Hadamard transformation. The final result, **F**, will be in memory A if m is even, and in B if m is odd.

THE FHT DIGITAL SYSTEM

Fig. 4 shows the overall block diagram of FHT digital system which is designed to finish FHT (foward or inverse) within 20 ms for $N=4096$ ($m=12$). The system has an 8−bit data bus. Since optical device output, the forward Hadamard transformation result, is continuously transmitted to the FHT digital system, and the FHT output (final result) is also required to be continuous, four memories (A, A′, B and C) are needed. When A (or A′) is reading the optical device, A′ (or A) cooperates with B to do above described FHT operations. In the mean time memory C delivers the final result to a monitor or recording system.

An Overflow Preventer is added after the arithmetic unit to prevent overflow caused by adding or subtracting. Since the system has an 8−bit data bus, the datum values are within the range of $-128\sim+127$. If, in any step, one or more data of AU output values are beyond the range of $-64\sim+64$, then, in the next step, AU output may overflow. But the output will not overflow when it is divided by 2. A circuit is designed to detect the large data whose values are beyond the range of $-64\sim+64$. If large data are detected, the Overflow Preventer will, in the next whole step, divide AU output by 2 (right shifting for one bit). If no large data is detected, no dividing operations will be done throughout the next step. It is important to note that an error will be introduced if

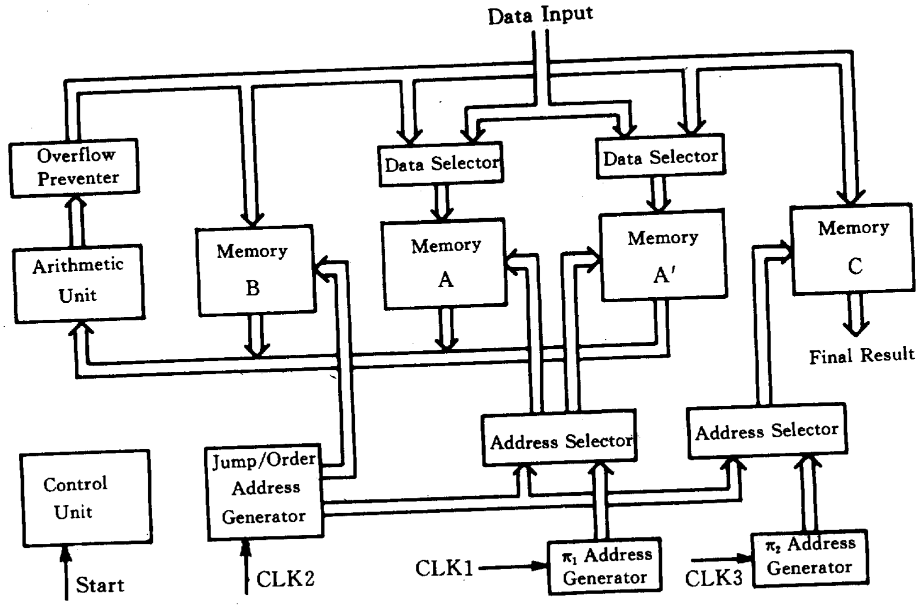


Fig. 4 The diagram of FHT digital system

there is a right shifting operation. The error analysis is presented in réfrence 2.

Three address generators are employed in the system， as is shown in Table 1.

It is important to know that if the input data come from an optical system which employs a left-cycling S matrix[1](not a standard Hadamard matrix)， a π_1 address generator is required to change the orders of input data. A π_2 address generator is also required to change the orders of output data[1].

A Control Unit is designed to control the address generators， transfer of data，memory read/write， arithmetic unit， latches and shift registers. Since the clock frequency of Jump and Order address generators is 2.5MHz， then it needs $T=0.4\mu s$(one colck cycle) to finish an adding operation or a subtracting operation. The number of cycles in one step is $N+2$. $m(N+2)$ cycles are required to finish m step. So the whole time needed to complete Hadamard transformation is

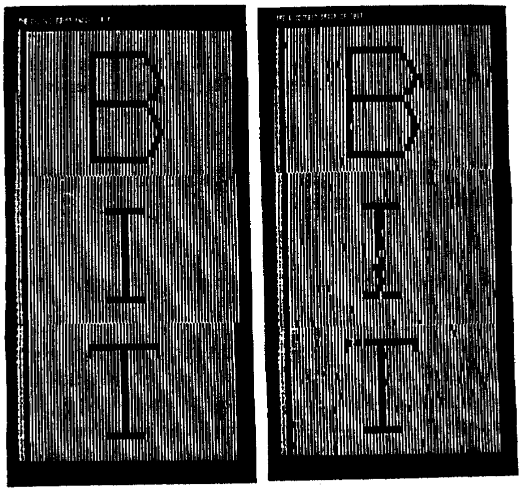
$$\begin{aligned} t &= m(N+2)T \\ &= 12 \times (4096+2) \times 0.4\mu s \\ &= 19.7\ \mathrm{ms} \end{aligned}$$

Table 1. Address generators in the FHT system

Addres Generator	Purpose	Clock	Clock Frequency
Jump address	For source memory	CLK3	2.5MHz
Order address	For receiver memory	CLK3	2.5MHz
π_1 address	For A and A′	CLK1	200kHz
π_2 address	For C	CLK2	200kHz

EXPERINENTAL RESULT

A picture ' BIT' is shown in Fig. 5a. Do forward Hadamard transformation by multiplying the picture with the left-cycling S matrix, which is done with a microcomputer. The result of forward transformation is the input data which is transmitted to memory A(or A') in FHT digital system. A start signal will trigger the system to begin (inverse) Hadamard transformation. An interval of 19. 7ms after the start signal ,the final result will be stored in memory C. A microcomputer is used to read the data from memory C, and display the result (recovered picture),as is shown in Fig. 5b. The noise in the recovered picture is caused by the error introduced by right shifting operations.



(a) (b)

Fig . 5 Experiment result

CONCLUSION

We may draw the following conclusions:

1. Hadamard matrix expansion method makes easier the design and construction of hardware system.

2. Transformation time is $m(2^m+2)T$. That is the minimum time required to finish Hadamard transformation.

3. The FHT digital system presented here will introduce a certain error, but it can be reduced by increasing the bit number of data bus.

REFERENCES

1. M. Harwit, N. J. A. Sloane,Hadamard Transform Optics, Academic Press,1979.

2. Chen Anshi, Decoding Methods in Hadamard Transform Imaging Techniques, M. S. thesis, Beijing Institute of Technology, 1988.