論文 90-27-3-15

Design and Fabrication of a Processing Element for 2-D Systolic FFT Array

(고속 퓨리어변환용 2차원 시스토릭 어레이를 위한 ~~처리요소의 설계 및 제작)~~

李 文 基*, 辛 卿 旭*, 崔 炳 允*

(Moon Key Lee, Kyung Wook Shin, and Byeong Yoon Choi)

要　約

고속 퓨리어변환(Fast Fourier Transform)연산용 2차원 시스토릭 어레이의 기본 구성요소인 단위 처리요소(Unit processing element)를 집적회로로 설계, 제작하고 제작된 칩을 평가하였다. 설계된 칩은 FFT 연산을 위한 데이타셔플링 기능과 반쪽 버터플라이 연산기능을 수행한다. 약 6,500여개의 트랜지스터로 구성된 이 칩은 표준셀 방식으로 설계되었으며, 2미크론 이중 금속 P-Well CMOS 공정으로 제작되었다. 제작된 칩을 웨이퍼 상태로 프로브카드를 이용하여 평가하였으며 그 결과, 20MHz 클럭 주파수에서 반쪽 버터플라이 연산이 0.5 μsec에 수행됨을 확인하였다. 본 논문에서 설계, 제작된 칩을 이용하여 1024-point FFT를 연산하는 경우 11.2μsec의 시간이 소요될 것으로 예상된다.

Abstract

This paper describes the design and fabrication of a processing element that will be used as a component in the construction of a two dimensional systolic array for FFT. The chip performs data shuffling and radix-2 decimation-in-time (DIT) butterfly arithmetic. It consists of a data routing unit, internal control logic and HBA unit which computes butterfly arithmetic. The 6.5K transistors processing element designed with standard cells has been fabricated with a 2ừm double metal CMOS process, and evaluated by wafer probing measurements. The measured characteristics show that a HBA can be computed in 0.5 usec with a 20MHz clock, and it is estimated that the FFT of length 1024 can be transformed in 11.2 usec.

I. Introduction

A need to efficiently compute the Fast Fourier Transform (FFT) arises in many areas of digital signal and image processing. For the implementation of the FFT onto VLSI-oriented architectures, however, the requirements for complex global communication of the FFT is contrary to VLSI design principle.

Among various architectures for the FFT[1-5], mesh-connected array of processing elements(PE), which is essentially identical to the one proposed for ILLIAC-IV computer[6], is most appealing for VLSI implementation of the FFT. The reasons for this choice are as follows; First, it has local and regular interconnection of PEs. Second, it

*正會員, 延世大學校 電子工學科

(Dept. of Elec. Eng., Yonsei Univ.)

接受日字 : 1989年 12月 4日

performs data shuffling of an N-point FFT in 0 $(\sqrt{N})$ time with a systolic communication scheme, i.e., regular and nearest-neighbor communications in parallel and pipelined fashion.

In general, there are two approaches to the design of a special-purpose computing system. One is to design a dedicated hard-wired system, which can not be changed to perform other tasks besides the original task[8]. The other is to design a flexible system, which allows users to change it easily. The choice of implementation method tends to be driven by some factors such as performance considerations, flexibility requirements and implementation feasibility.

In this paper, we chose flexibility and implementation feasibility as major design considerations, rather than performance requirements. So, we have designed a processing element chip, which will be used as a basic component in the construction of the mesh-connected FFT array[9,10,11]. On the basis of observations on the way in which the FFT is computed on the 2-D array, we adopted half-butterfly arithmetic (HBA)[7] as the basic arithmetic function of each PE rather than conventional butterfly arithmetic in order to optimize array level performance.

This paper is organized as follows; Section II will briefly discuss the FFT computation algorithm on the mesh-connected array. In section III, the circuit and layout design of the chip will be described. In addition, design verifications such as switch-level logic simulation and timing simulation including parasitic effects due to layout will be presented. Finally, chip fabrication and some measurement results will be presented in section IV.

II. FFT Computation on the Mesh-Connected Array

The mesh-connected systolic array for the FFT of length $N=2^{m+n}$ is composed of $2^m x 2^n$ identical PEs, and computes the FFT as follows[7,11];

```
BEGIN
    initial data loading
    FOR q=1 TO m+n DO IN PARALLEL for all PEs
        BEGIN
            data shuffling
            half-butterfly arithmetic (HBA)
        END
    data I/O pipeline
END.
```

After initial input data are loaded into the array in row-major order, data shuffling and HBA computations are performed through $\log_2 N$ butterfly stages. Data shuffling operations are carried out in systolic communication scheme, i.e., local and regular communication in parallel and pipelined fashion. The HBA takes its name from the definition given in Eq. (1), in other words, it is half of the conventional radix-2 decimation-in-time (DIT) butterfly[7].

$$HBA^{+}=f(q-1,k)+f(q-1,k+D(q))\cdot W^{p}$$
$$HBA^{-}=f(q-1,k)-f(q-1,k+D(q))\cdot W^{p} \quad (1)$$
where, Twiddle factor $W^{p}=EXP(-j2\pi p/N)$
Shuffling distance $D(q)=N/2^{q}$

In eq. (1), k takes the values that satisfy the inequality of i MOD $(N/2^{q-1}) < N/2^{q}$ (where, i=0,1,...N-1). And, when $q=\log_2 N$, HBA+ and HBA-represent the final FFT results and f(0,k) represents initial input data. Since all the terms in Eq. (1) have complex values, the HBA can be represented as follows;

$$HBA\&= (Ar+jAi) \ \& \ (Br+jBi)\cdot(Wr+jWi) \quad (2)$$

where, $Ar=Re\{f(q-1, k)\}$, $Ai=I_m\{f(q-1, k)\}$
$Br=Re\{f(q-1, k+D(q))\}$, $Bi=I_m\{f(q-1, k+D(q))$
$W_r=Re\{W^{p}\}$, $W_i=I_m\{W^{p}\}$
operator & denotes '+' or '−'

The key concept in the HBA is to simultaneously compute the two HBAs in two PEs, rather than compute the conventional butterfly in a PE, in order to eliminate the data reshuffling operation and to achieve 100% PE utilization.

III. Circuit Design

The chip performs two functions, data shuffling then HBA computation. For the HBA computation of the q-th butterfly stage, two data f(q-1, k) and f(q-1, k+D(q)) must be shuffled by shuffling distance D(q) as can be seen from Eq.(1) After data shuffling, the chip computes the HBA&, whose operator & is determined by butterfly stage q and its position in the array.

The chip is composed of two data routing units (DRU), a HBA unit, data and coefficient registers, and internal control logic (ICL) as depicted in Fig. (1).

The DRU communicates with four nearest neighbor PEs to perform data shuffling and

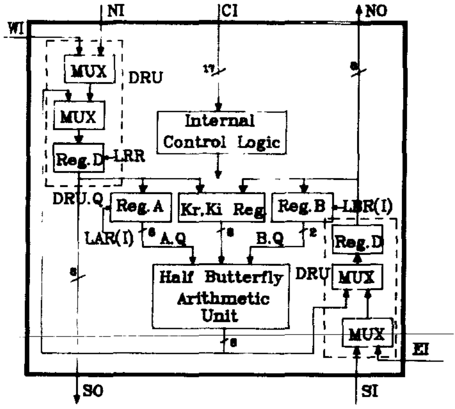


Fig.1. Internal structure of the chip.

twiddle factor loading. It consists of two multiplexers and a PIPO(Parallel-In Parallel-Out) register. The first multiplexer in the DRU controls data shuffling directions, i.e., horizontal shuffling or vertical shuffling depending on the butterfly stage. Based on the time performance analysis for the mesh-connected FFT array[7], a bit-parallel communication scheme is chosen for the DRU. Master-slave D-type flip-flops (MS-DFF) are used for the PIPO register.

As can be seen from Eq. (2), computational requirements for the HBA are four multiplications plus three additions and one subtraction or one addition and three subtractions depending on the HBA type. Thus direct implementation of the HBA onto silicon will be very inefficient due to large chip area when parallel arithmetic is used.

In this paper, distributed arithmetic technique[12] was employed in the HBAU design to achieve adder-based implementation of the HBA. When N-bit, fixed point, fractional two's complement arithmetic is used, the HBA computation in Eq. (2) can be expressed using distributed arithmetic concept as follow[11];

$$\begin{aligned} R_e\{HBA\&\} &= A_r \;\&\; 2^{-(N-1)}\cdot(W_r-W_i)/2+\sum_{n=0}^{N-1} Q_{rn}(B_{rn},B_{in})\cdot 2^{-n} \\ I_m\{HBA\&\} &= A_i \;\&\; 2^{-(N-1)}\cdot(W_r+W_i)/2+\sum_{n=0}^{N-1} Q_{in}(B_{rn},B_{in})\cdot 2^{-n} \end{aligned} \qquad (3)$$

where, subscripts r,i denote real and imaginary, respectively and Brn and Bin represent the n-th bit of Br and Bi, respectively.

In Eq. (3), the coefficients Qrn, Qin are determined by the combination of the n-th bits of Brn and Bin as tabulated in Table I.

Table I. Coefficient selection for distributed arithmetic.

Brn	Bin	Qrn		Qin	
		n=0	n≠0	n=0	n≠0
0	0	Ki	−Ki	Kr	−Kr
0	1	Kr	−Kr	−Ki	Ki
1	0	−Kr	Kr	Ki	−Ki
1	1	−Ki	Ki	−Kr	Kr

Where, Kr=(Wr+Wi)/2 and Ki=(Wr-Wi)/2

According to Eq. (3), the HBA can be computed by two separate accumulation operations, thus permitting the HBAU to be partitioned into two identical parts, one is responsible for real output and the other for imaginary output, as shown in the block diagram of the HBAU in Fig.(2).

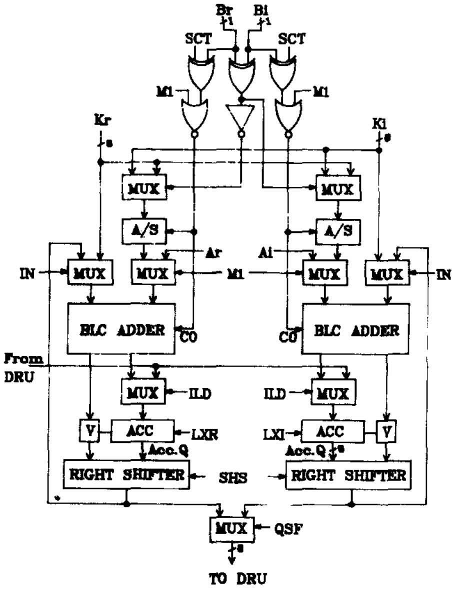


Fig.2. Block diagram of the HBAU.

The chip adopts an 8-bit word length for both data and coefficients, so it takes the HBAU 8 clock cycles to perform the coefficient selection and accumulation operations in Eq.(3) . In order to include the second term in Eq. (3) in the accumulation operation, distributed arithmetic begins with signal IN='HIGH' (see Fig.2) . The selection of Kr and Ki in Table-I is made on the basis of the exclusive-OR (XOR) or exclusive-NOR (XNOR) combination of Br and Bi, as shown in Fig.(2). Also, the signs of the coefficients are determined by Br for the real partial product and by Bi for the imaginary partial product. The signal SCT= 'HIGH' in Fig.(2) indicates the last accumulation step in distributed arithmetic (i.e., sign-bit(n=0) in Table-I), which requires the signs of coefficients Kr and Ki to be reversed. The final computational step in the HBA computation is to perform the & operation with the result of the distributed arithmetic and data Ar(Ai). The signal M1= 'HIGH' in Fig. (2) indicates this computational step.

In order to prevent overflow problem due to fixed register length in the HBAU, step-by-step scaling method was adopted, which scales down data by a factor of 2 (i.e., shift right 1-bit) at every butterfly stage. This scale down of data is performed before the distributed arithmetic begins. As a result, the HBA computation requires total 10 clock cycles from data scaling down to addition for & operation.

By using fixed-point, two's complement notation, subtraction due to -Kr or -Ki can be easily converted into addition. This conversion is performed by A/S block with CO='1' in Fig.(2). Furthermore, the adder used in the distributed arithmetic can be reused to carry out the & operation in Eq.(3), thus the HBAU can be efficiently realized with only two adders, which occupy smaller chip area than multiplier. Two 8-bit BLC (Binary Lookahead Carry)[13] adders were used to implement the distributed arithmetic.

The accumulators in the HBAU temporarily store the result of HBA computation during data shuffling operation, as well as accumulates intermediate results of distributed arithmetic.

IV. Layout Design

Chip layout was designed using standard cells with the aids of auto-routing tool, Sorcery[15]. Total 25 types of standard cells and one BLC macrocell were used. The floor plan of the chip is built up of two half-planes, one is responsible for real output and the other for imaginary output, as depicted in Fig.(4-a). Internal control logic (ICL) was placed between the two half-planes to equalize the distribution path of internally gated control signals. The cell placement was accomplished by hand, and the wiring was achieved by automatic router. The 1st level metal line was used for horizontal interconnections, and the 2nd level metal line for vertical interconnections.

Design verification was done at three levels; i) switch-level logic simulation, ii) timing analysis of critical delay path, iii) geometrical layout rule checking. The transistor and interconnection list, which was extracted from the chip layout, was used for switch-level logic simulation to verify the logic functions of the designed PE. It uncovered one design error in the coefficient initialization block, and this mistake was corrected by adding a simple combinational logic which generates clear/set inputs for the sign-bit of the coefficient registers Kr and Ki depending on the operator & in Eq.(3). From the logic simulations for the HBA+ and HBA- computation of the 1st butterfly stage, it was found that the simulation results differed from calculated values by only the least significant bit, and this discrepancy was caused by the scaling down operation. So, we could confirm the correctness of the design

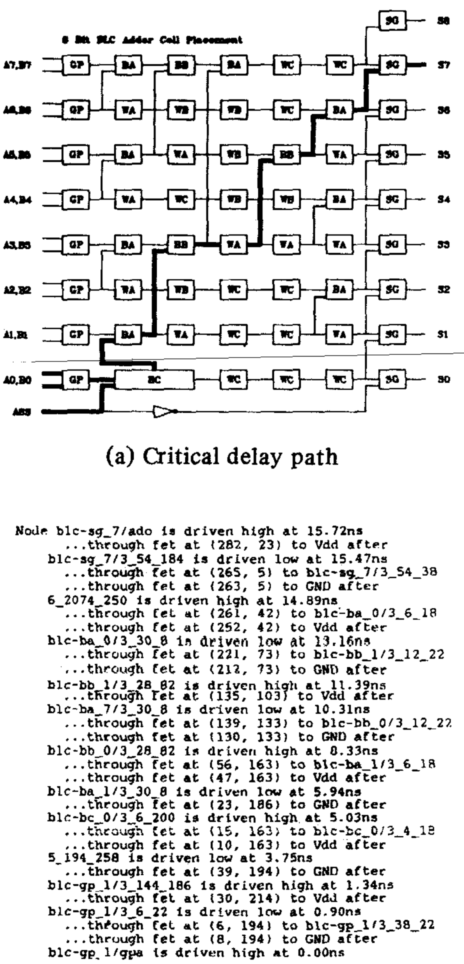
After the logic verification, to check that timing constraints were met, timing simulation based on the lumped RC delay model[14] was done on the circuit which takes account of the parasitic capacitances extracted from layout. Fig.(3) shows the critical path of the BLC adder found by timing simulator and its simulation result . It says that the 8-bit BLC adder takes 15.7nsec of critical delay. Some timing simulation results will be compared with measurement result of the fabricated chip in the next section.

Finally, the DRC (Design Rule Check) and ERC (Electrical Rule Check) were done to find out geometrical and electrical rule violations.

V. Fabrication and Measurement Results

1. Fabrication

The chip has been fabricated with 2um double metal P-Well CMOS process. Table-II summarizes some characteristics of the process used for the chip fabrication. The die photograph of the fabricated chip is shown in Fig.(4-b). The chip contains about 6,500 transistors in the area of 3.4 x 2.8mm^2 . The die size is 4.5 x 4.5 mm^2 including



(a) Critical delay path

```
Node blc-sg_7/ado is driven high at 15.72ns
     ...through fet at (282, 23) to Vdd after
    blc-sg_7/3_54_184 is driven low at 15.47ns
     ...through fet at (265, 5) to blc-sg_7/3_54_38
     ...through fet at (263, 5) to GND after
    6_2074_250 is driven high at 14.89ns
     ...through fet at (261, 42) to blc-ba_0/3_6_18
     ...through fet at (252, 42) to Vdd after
    blc-ba_0/3_30_8 is driven low at 13.16ns
     ...through fet at (221, 73) to blc-bb_1/3_12_22
     ...through fet at (212, 73) to GND after
    blc-bb_1/3_28_82 is driven high at 11.39ns
     ...through fet at (135, 103) to Vdd after
    blc-ba_7/3_30_8 is driven low at 10.31ns
     ...through fet at (139, 133) to blc-bb_0/3_12_22
     ...through fet at (130, 133) to GND after
    blc-bb_0/3_28_82 is driven high at 8.33ns
     ...through fet at (56, 163) to blc-ba_1/3_6_18
     ...through fet at (47, 163) to Vdd after
    blc-ba_1/3_30_8 is driven low at 5.94ns
     ...through fet at (23, 186) to GND after
    blc-bc_0/3_6_200 is driven high at 5.03ns
     ...through fet at (15, 163) to blc-bc_0/3_4_18
     ...through fet at (10, 163) to Vdd after
    5_194_258 is driven low at 3.75ns
     ...through fet at (39, 194) to GND after
    blc-gp_1/3_144_186 is driven high at 1.34ns
     ...through fet at (30, 214) to Vdd after
    blc-gp_1/3_6_22 is driven low at 0.90ns
     ...through fet at (6, 194) to blc-gp_1/3_38_22
     ...through fet at (8, 194) to GND after
    blc-gp_1/gpa is driven high at 0.00ns
```

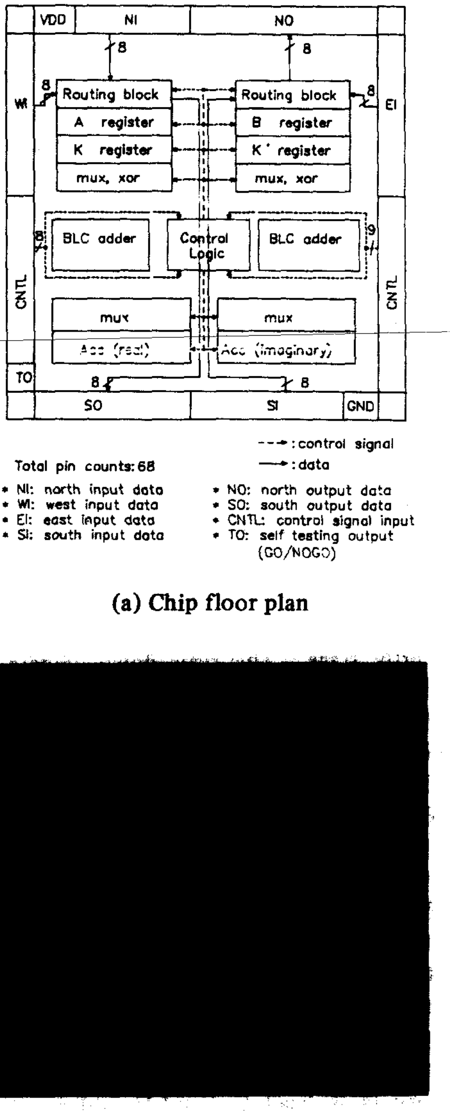
(b) Timing simulation result

Fig.3. Timing simulation of the BLC adder.

68 pads. Some test cells such as ring oscillators and BLC adder were placed at the leftmost part of the chip to measure some delay characteristics. Four types of ring oscillators were included; one 21 stage ring oscillator to measure the delay of a single inverter with Wp/Wn=10/5, and three 21 stage ring oscillators with 4,000 um interconnection lines of 1 st metal, 2 nd metal and poly in order to measure the delays of these interconnection lines. Table III summarizes some measured device characteristics.

2. *Measurement Results*

The function and performance of the fabricated chip were evaluated by wafer probing measure-



(a) Chip floor plan

(b) Die photograph

Fig.4. Floor plan and die photograph of the chip.

Table-II. Summary of process characteristics.

Technology	2μm, double metal, P-Well CMOS		
Effective channel length	1.6μm for NMOS, 1.44μm for PMOS		
Gate oxide thickness	38nm		
Nsub	2.0E16 for NMOS, 6.12E15 for PMOS		
Minimum width/space of	poly	1st metal	2nd metal
interconnection layer (μm)	2/2.5	3/2	4/3

Table-III. Summary of measured device characteristics.

Measured V_T (W/L=20/2)	PMOS		NMOS		
	−0.777		+0.870		
Measured g_m (W/L=20/2)	PMOS		NMOS		
	23.5E-6		50.0E−6		
Inverter (W_p/W_n=10/5) delay (nsec)*	Vdd=3V	4 V	5 V	6 V	7 V
	0.65	0.46	0.38	0.32	0.30
Interconnection line delay (psec/μm)*	1st metal	2nd metal		poly	
	1.25	0.78		8.1	

* measured using 21 stage ring oscillators

Table-IV. Comparison of measured delays with timing simulation results.

Block	measured delay (ns)			estimated delay by timing simulation (ns)
	Vdd=4V	Vdd=5V	Vdd=6V	
DRU register	19	13	12	11
Data register	5	5	5	5
Accumulator	8	11	10	5
BLC adder	32			15.7

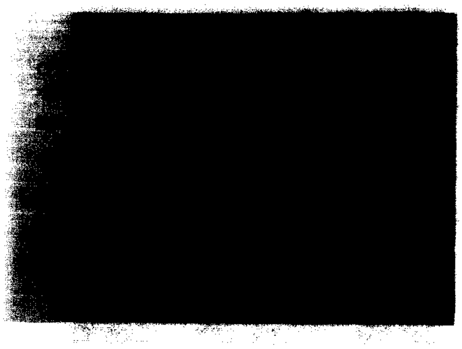


Fig.5. Measured waveforms of control signals for HBA computation.

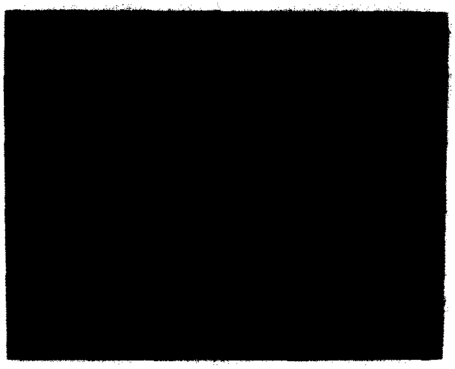


Fig.6. Measured waveforms for DRU register.

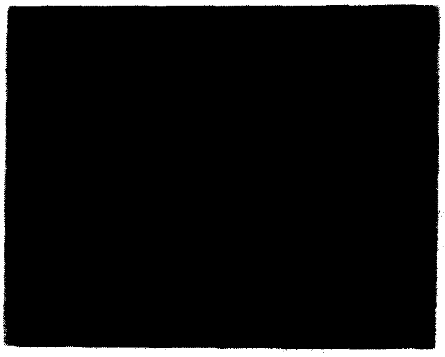


Fig.7. Measured waveforms for data register.

ments. Fig.(5) shows the measured waveforms of internally gated signals used for data scaling down, distributed arithmetic and & operation in Eq.(3). As can be seen from Fig.(5), the HBA computation requires total 10 clock cycles from pulse LAR(I) for data scaling down to the last pulse of LXR(I) for & operation in Eq.(3).

Fig.(6)-(8) show some measured characteristics for functional blocks of the fabricated chip. The measured delays for DRU block are 19 nsec, 13 nsec and 12 nsec at Vdd=4V, 5V, 6V, respectively as shown in Fig.(6) Also, 5 nsec delay for data register block shown in Fig.(7) and 11 nsec delay for accumulator shown in Fig.(8) were measured at room temperature with Vdd=5V.

Fig. (9) shows the measured propagation delay for the critical path of the 8-bit BLC adder shown in Fig.(3). Input pulse was applied to A0, and the critical path delay was measured at S7 with A7-A1=1111111, B7-B0=00000000 and ASS=1. The critical path delays for rising and falling waveforms were 26 nsec and 32 nsec, respectively.

A comparison of these measured characteristics with the delays estimated by timing simulation is shown in Table-IV. Some delay underestimation in timing simulation results might be caused by the delay model (lumped RC model) used in our timing simulator.

From Table-IV, we can recognize that the operating clock frequency of the chip is limited by the

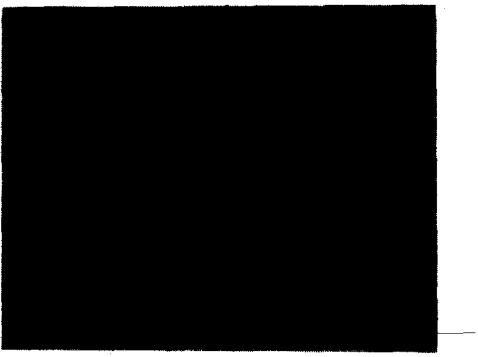


Fig.8. Measured waveforms for accumulator.

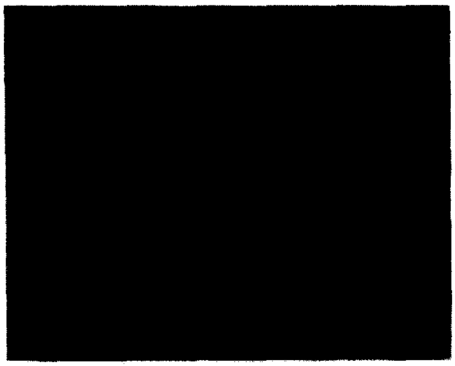


Fig.9. Measured waveforms for critical path delay of the BLC adder.

critical path delay of the BLC adder. To determine clock period T_{clk}, the relation of $T_{clk} > T_{blc} + 3\ T_{mux}$ must be satisfied as can be seen from Fig.(2). Thus, T_{clk} should be greater than about 40 nsec and the chip can operate with 20MHz clock frequency at Vdd=5V.

Based on this result, the HBA computation requiring total 10 clock cycles consumes 0.5 usec, and it is estimated that the FFT of length 1024 can be computed in 11.2 usec at 20MHz clock.

IV. Conclusions

A processing element chip to be used for mesh-connected systolic FFT array has been designed and fabricated with 2um double metal P-Well CMOS technology. The half-butterfly arithmetic concept was adopted as the basic arithmetic function of the chip in order to optimize array level performance. Some measured characteristics show that the chip can operate with 20MHz clock, and the HBA can be computed in 0.5 usec. Based on this result, it is estimated that a 1024-point FFT can be transformed in 11.2 usec thus, the chip can be used as a component in the construction of systolic FFT array using silicon-on-silicon hybrid package technology or wafer scale integration for high performance DSP applications.

References

[1] H.S. Stone, "Parallel processing with the perfect shuffle," *IEEE Trans. Comput.*, vol. C-20, pp. 153-161, Feb., 1971.

[2] Special Issue on sorting, *IEEE Trans. Comput.*, vol. C-34, no. 4, Apr. 1985.

[3] C.D. Thompson, "Fourier transform in VLSI," *IEEE Trans. Comput.*, vol. C-32, no. 11, pp. 1047-1057, Nov., 1983

[4] T. Wiely, et al, "A FFT systolic processor and its applications," *Proceedings of the IEEE ASSP-84*, vol. 2, Mar., 1984.

[5] K. Sapiecha, R. Jarocki, "Modular architecture for high performance implementation of FFT algorithm," *Proceedings of the 13th annual international symposium on computer architecture.* pp. 261-270, Jun. 1986.

[6] J.E. Stevens, "A fast Fourier transform subroutine for ILLIAC-IV," *Technical report, Center for advance computation, Illinois,* 1971.

[7] M.K. Lee, "Systolic array for FFT computation," *Multi Project Chip (MPC) development,* Dept., of electronic Eng., Yonsei Univ., Seoul, Korea, Aug., 1988.

[8] K.W. Shin, B.Y Choi, B.R. Kim, M.K. Lee, "A single chip 16-point FFT processor (YUSAF16)", *International conference on VLSI and CAD,* pp. 133-136, Oct., 1989

[9] B.Y. Choi, B.H. Kang, J.K. Lee, K.W. Shin, M.K. Lee, "VLSI implementation of two-dimensional FFT algorithm on systolic array," *Proceedings of the IEEE TENCON '87,* pp. 125-131, Aug., 1987.

[10] K.W. Shin, M.K. Lee, "A VLSI architecture for parallel computation of FFT," *Proceedings of the international conference on systolic arrays* pp. 116-125, May, 1989.

[11] K.W. Shin, B.Y. Choi, M.K. Lee, "A VLSI

architecture of systolic array for FFT computation," *Journal of the KITE,* vol. 25, no. 9, pp. 97-106, 1988.

[12] S.A. White, "A simple FFT butterfly arithmetic unit," *IEEE Trans. on Circuit and Systems,* vol. CAS-28, no. 4, pp. 352-355, Apr., 1981.

[13] N. Weste, K. Eshraghiam, "Principles of CMOS VLSI design," Addison Wesley, 1985

[14] J.K. Ousterhout, "A switch-level timing verifier for digital MOS VLSI," *IEEE Trans. computer-Aided Design,* vol. CAD-4, no. 3, pp. 336-349, Jul., 1985.

[15] "*SORCERY* user's manual," VLSI & CAD Lab., Yonsei Univ., SEOUL, KOREA.

著 者 紹 介

李 文 基 (正會員) 第25卷 第9號 參照
현재 연세대학교 전자공학과 교수

●

崔 炳 允 (正會員) 第25卷 第9號 參照
현재 연세대학교 대학원 전자공학과 박사과정 재학중

辛 卿 旭 (正會員) 第27卷 第1號 參照
현재 연세대학교 대학원 전자공학과 박사과정 재학중