

BIT SLICE SIGNAL PROCESSOR 를 이용한 DCT 의 구현

김 동 륙<sup>\*\*</sup> 고 석 번<sup>\*</sup> 백 승 권<sup>\*</sup> 이 태 수<sup>\*</sup> 민 병 구<sup>\*\*</sup>  
 서울 대학교 의과대학 의공학교실<sup>\*\*</sup>, 공과대학 제어계측공학과<sup>\*</sup>

(Implementation of DCT using Bit Slice Signal Processor)

Dong L. Kim, Seok B. Go, Seung K. Paek, Tae S. Lee, Byong G. Min  
 Dept. of Biomed. Eng., Dept of Control & Instr. Seoul Nat'l Univ.

A microprogrammable Bit Slice Sinal Processor for image processing is implemented. Processing speed is increased by the parrallelism in horizontal microprogram using 120bits microcode, pipelined architecture, 2 bank memory switching that interfaces with the Host through DMA, a variable clock control, overflow checking H/W,look-up table method and cache memory.

With this processor,a DCT algorithm which uses 2-D FFT is performed. The execution time for 512 x 512 x 8 image is 12 sec when 16 bit operation is runned, and the recovered image has acceptable quality with MSE 0.276 %.

I. 서론

병원전산망에서는 D.R.(Digital Radiography)등을 이용한 X-ray 화상 데이터량이 방대하게 되므로 화상 신호의 압축이 필요하게 된다. 그러나 화상신호의 압축및 재생에는 많은 시간이 소요되므로, 각 workstation 에서는 화상신호 처리용 processor가 필요하게 된다. 본 논문에서는 서울대학교 병원내 workstation을 위한 화상처리용 bit slice signal processor를 개발, DCT에 응용한 결과를 논한다.

II.FCT (Fast Cosine Transform)

Markhoul은 Narasimha 와 Peterson 이 제시한 방법을 발전시켜서, N 이 2의 급수 라는 제한성을 없애고

일반화 시켰다.(N 이 짝수나 홀수 ) 또, Inverse FCT도 구했고,2-Dimension 의 경우로 확장시켰는데, 이 알고리즘은 Chen 의 것과 비교할 때 비슷한 성능을 가진다.[1,2]

2-D 의 경우 그 알고리즘을 정리하면 다음과 같다.

. FCT procedure (N1 x N2 matrix)

$$1) \quad v(n1, n2) = x(2n1, 2n2) \quad 0 \leq n1 \leq \left\lfloor \frac{N1-1}{2} \right\rfloor, \quad 0 \leq n2 \leq \left\lfloor \frac{N2-1}{2} \right\rfloor$$

$$v(n1, n2) = x(2N1-2n1-1, 2n2) \quad \left\lfloor \frac{N1+1}{2} \right\rfloor \leq n1 \leq N1-1, \quad 0 \leq n2 \leq \left\lfloor \frac{N2-1}{2} \right\rfloor$$

$$v(n1, n2) = x(2n1, 2N2-2n2-1) \quad 0 \leq n1 \leq \left\lfloor \frac{N1-1}{2} \right\rfloor, \quad \left\lfloor \frac{N2+1}{2} \right\rfloor \leq n2 \leq N2-1$$

$$v(n1, n2) = x(2N1-2n1-1, 2n2-2n2-1) \quad \left\lfloor \frac{N1+1}{2} \right\rfloor \leq n1 \leq N1-1, \quad \left\lfloor \frac{N2+1}{2} \right\rfloor \leq n2 \leq N2-1$$

로써 v(n1, n2) 를 구한다.

2) V(k)=2-D DFT of v(n1, n2) 를 구한다.

$$3) \quad C(k1, k2) = 2\text{Re}\{W_{4N1}^{k1} [W_{4N2}^{k2} V(k1, k2) + W_{4N2}^{-k2} V(k1, N2-k2)]\}$$

$$\text{또는} \quad = 2\text{Re}\{W_{4N2}^{k2} [W_{4N1}^{k1} V(k1, k2) + W_{4N1}^{-k1} V(N1-k1, k2)]\}$$

로써 C(k1, k2) 를 구한다.

III. Microprogrammable Signal Processor

1. 구성방법

본연구에서는 Am2900 family bit slice I.C.를 사용하여 pipelining을 고려한 구조를 갖고,

# BIT SLICE SIGNAL PROCESSOR를 이용한 DCT의 구현

horizontal microprogram에 의해 parallelism을 충분히 살린 신호처리용 고속 processor를 제작하였다.

Processor는 Writable Control store (WCS)로 구성하였으므로 control unit에 내장되는 microprogram의 내용에 따라 여러가지 digital 신호처리 알고리즘을 빠른 시간 내에 수행할 수 있다.[3]

## 2. 구조

본 processor는 다음과 같이 7 부분으로 크게 나누어 진다.

- . ALU 1 (Am 2901C X 4, Am 2902 X 1)
- . ALU 2 (Am 2903A X 4, Am 2902 X 1, Am 2904, Am 29517, look up table, cache memory)
- . Data path (MSI 소자)
- . Processor control unit (Am 2910, Intel 2148 X 30 =120 bit)
- . Main memory 및 memory control (80K word X 2 bank)
- . Clock (Am 2925)
- . Interface to Host computer (DMA)

System의 전체 block도는 그림 1과 같다.

## 3. 소프트웨어

본 processor의 개발을 위해 작성된 program은 크게 2가지로 나뉜다.

A. Bit-slice Development tool— 이것은 Bit slice processor의 Microprogram memory 및 main memory를 조작하여 Host computer가 Bit slice processor의 monitor 역할을 하도록 작성된 것이다.

B. Microprogram support tool [4]— 본 processor를 사용하여 수행 될 디지털신호처리 알고리즘은 Meta Assembler program을 사용하여 Symbolic language를 Microcode로 변환하여 이루어진다. 사용된 Meta Assembler의 구조는 phase I과 phase II로 나뉘어진다.

phase I은 Micropinstruction을 정의해 주고, 정의된 Symbol들을 table로 만들어 Meta Assembler phase II가 실행될 때 참조된다.

phase II는 phase I에서 정의한 명령을 이용하여, 알고리즘을 따라 작성된 Symbolic Microprogram을 Microcode로 변환시켜준다.

## IV. Microprogramming

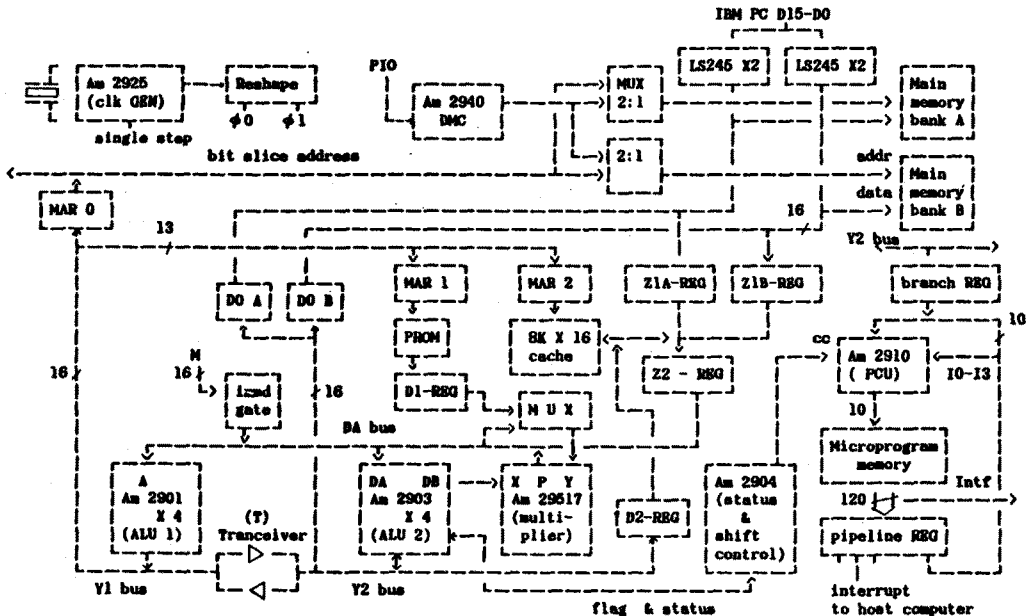


그림 1. 전체 블럭도

1. FFT 의 경우

데이터의 조건에 따라 분기하는 곳이 다르므로, 정확한 명령수행 횟수는 셀 수 없지만 각 분기점에서 평균한 Butterfly loop 내의 명령 수행횟수는 표 1 과 같다. 표 1은 memory access 때 2배로 연장되는 clock 의 수까지 고려한 것이다.

과 정	실행 명령 수
W * Xm(q)곱셈	19
Normalization	2
Butterfly덧셈	9
overflow set	0.5
Butterfly 마무리	8.5
합	39.0

표 1. Butterfly 내 명령 수

2. 응용program과의 결합

응용 program 은 현재 C 언어로 되어 있으며, program 도중에 쉽게 호출 하도록 작성하였다.

A. BSP initialization

```
mp=BSPinit(fname1,fname2,...)
```

```
int *mp
```

```
fname1,fname2,... : μ-program object file names
```

로서 호출한다. 함수 BSPinit()는 BSP(Bit Slice Processor)를 동작시키기 위하여 Interface 회로의 PIO 및 초기치를 설정하고, 인수로 넘겨받은 μ-program들을 μ-memory에 download한다.

B. 2-D FFT 의 호출

```
. BSPFFT2(rx,ix,n,s,tr,*mp,bank)
```

로 호출한다.

함수BSPFFT2()는 rx,ix의 데이터 배열 및 n, s, tr을 선택된 BSP main memory bank에 DMA로 보내고, 2-D FFT μ-program 의 시작번지(\*mp)를 jamming address로 내보낸 후, BSP의 동작을 시작하게 한다.

C. BSPDCT.C main program

BSPDCT 에서는, 위에서 설명한 program 들을 이용하여 적합한 bank switching 을 하면서 DCT 를

행하게 된다. 본 논문에서는 Markhoul 의 알고리즘을 적용하는데, PC 의 disk로부터 data file을 읽어온 후,  $v(n1,n2)=vr(n1,n2)+jvi(n1,n2)$  로 reordering해 주고 나서 BSPFFT2를 호출한다. 물론 맨 먼저 BSPinit() 를 해 주어야 한다. BSP가 2-D FFT를 수행하는 동안, 이미 처리되어서 또다른 bank 에 있는 데이터를

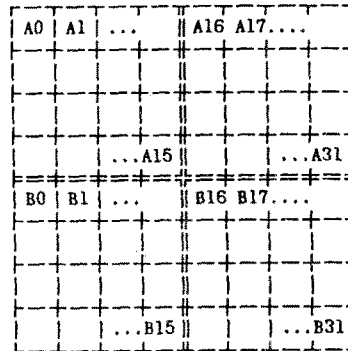


그림2. disk내 화상데이터의 구조

읽어들여 정리한 후 disk 에 써 넣고 나서, 다시 disk로부터 그 다음 데이터를 읽어오고 reordering 한 후 BSPFFT2 를 호출한다. disk 에 있는 화상 file은 그림2 과 같이 64개의 블록으로 나누어져 있고, 각 블록의 크기는 64 x 64 이다. memory bank 가 2개로 나누어져 있기 때문에 블록도 A, B 두부분으로 나누어서 고대로 선택되게 하였다. BSPDCT 의 알고리즘을 정리하면 그림 3와 같다.

V 결과

본 processor 에서는 512 X 512 X 8 bit Chest X-ray 화상을 처리하기 위하여 64 X 64 크기의 블록으로 나누어 계산 하였다. 또한 알고리즘의 효율적인 작성과 검토클 위해 C 언어로 정수형 계산을 simulation 하였다.

그림 4는 서울대병원 의공학과에서 개발한 DR(Digital Radiology) 시스템으로부터 얻은 Chest X-Ray 화상이다. 그림 5 는 원화상(그림 4) 을 BSP를 사용하여 정수형 계산으로 DCT 한 것이고, 그림 6 은 그것을 다시 inverse DCT 한 것이다.

BIT SLICE SIGNAL PROCESSOR를 이용한 DCT의 구현

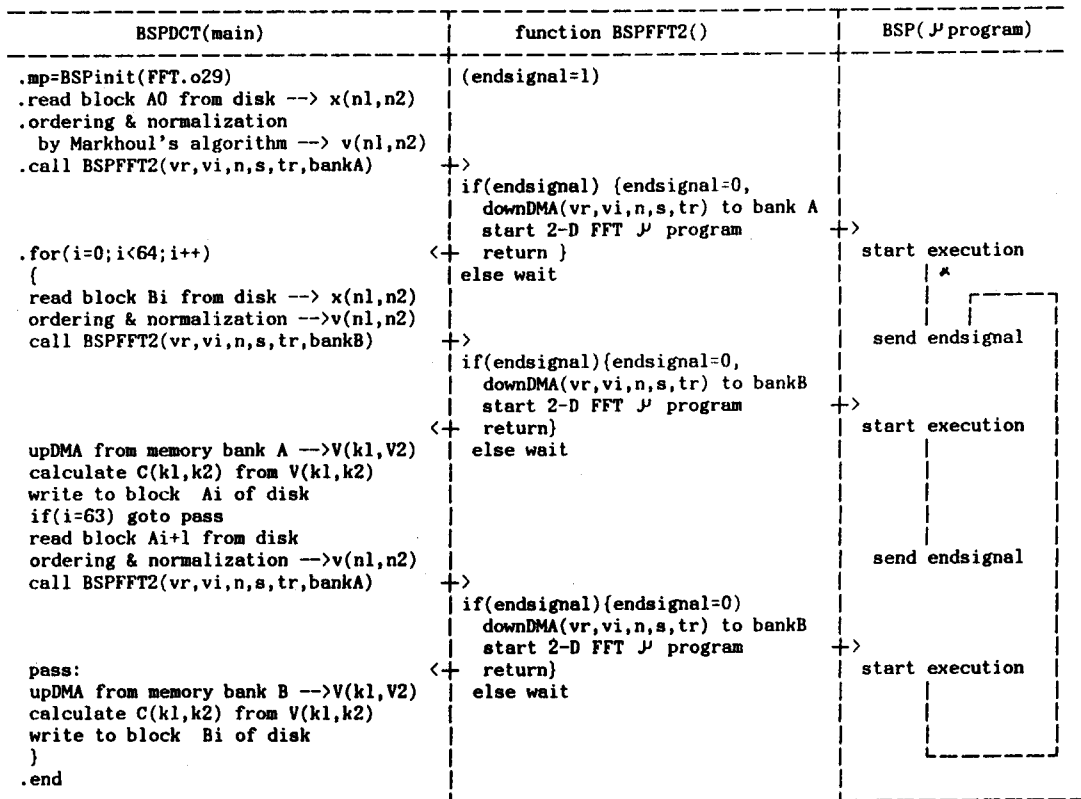


그림3. BSPDCT 의 알고리즘

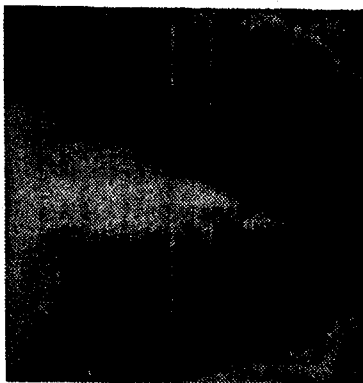


그림 4.  
human phantom  
D.R. image

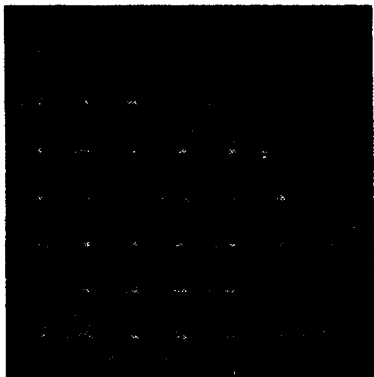


그림 5.  
integer DCT  
using BSP  
(block size  
= 64 x 64)

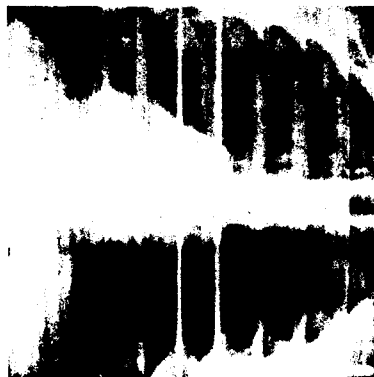


그림 6.  
integer IDCT  
using BSP

복원된 화상의 원화상과의 오차비교는 표 2 과 같다.

화 상	오차가 발생한 화소의 수	RMS 값 (grey level)
부동소숫점연산	1	0.001953
정수형 연산	9165	0.275910

표 2. 각 화상의 오차비교

. 실행시간

Host Computer 로 사용한 IBM PC/AT 의 disk access time 은 C 언어의 level 1 의 경우 4 kwords/0.6sec

이다(read/write평균). 또한, 본 processor 에 의한 32 point FFT 의 실행시간이 약 0.7msec 이다. 또, DCT 의 경우 64 X 64 블록을 다 계산 하려면 32 point FFT 를 호출하면서 (data가 실수이므로) 가로로 한번, 세로로 한번씩 각각 변환을 할 때

$$(64 + 64) \times 0.7 = \text{약 } 90 \text{ msec}$$

가 소요된다. 그런데 memory bank 가 2 개로 되어 있어 switching 을 하므로 disk access time 과 processor의 실행시간과는 겹치게 된다. disk access time 이 64 X 64 화상을 읽고 쓰는데 약 125 msec 걸리므로 결과적으로 한 블록의 처리시간은 processor 가 처리하는 시간, 즉 125 msec 가 된다. 512 X 512 의 화상을 처리하기 위해서는 64 블록의 데이터를 처리해야 하므로, 화상 한 화면의 처리 속도는 125 X 64 = 8 sec 가 된다. 실제로는 host computer 의 data ordering 및 reordering time 이 추가되므로 약 12 초 소요된다.

## VI 결론

본 논문에서는 화상신호를 디지털 신호처리하기 위한 Bit Slice Signal Processor 의 제작 및 알고리즘 구현에 대하여 논하였다. Microprogram으로 algorithm 을 구현하였을 때 512X512 의 화상을 DCT 로 계산하는데 약 12초가 걸렸다. 이 processor 는 전산망의 workstation 에 부착시켜 사용할 수 있다. 각 workstation 에서 요구하는 데이터는 database 에

저장되어 있다가 전산망을 통해 workstation 에 도달하고, workstation 에서는 본 processor 를 사용하여 화상을 복원한다. 그 때 걸리는 시간이 12 초 정도라면 충분히 짧은 시간이라고 생각된다. 정수형계산 외에 부동소수점 연산을 행할 수도 있고, 32bit 연산도 가능하므로 오차를 더 줄일 수도 있다. 앞으로 다른 여러가지 디지털 신호처리 알고리즘의 수행에 본 processor 를 편리하게 이용하기 위한 다양한 program 의 개발이 과제이다.

## 참고문헌

- [1] Markhoul, J. A fast discrete cosine transform in one - and two-dimensions. IEEE Trans. Acoust., Speech, Signal Process. ASSP-28, 27-34, 1980
- [2] Chen, W., Smith, C., and Fralick, S. A fast computational algorithm for the discrete cosine transform. IEEE Trans. Commun. COM-25, 1004-1009, 1977
- [3] J. Mick and J. Brick "Bit Slice Microprocessor Design", McGraw-Hill 1980
- [4] 박 제현 "Bit Slice MDS 의 설계 " 제어계측공학과 서울대학교 공과대학, 1985
- [5] S.C.Lo, and H.K.Huang Radiological Image compression: Full Frame Bit-Allocation Technique Radiology 1985 155:811-817