

論 文

저 전송률 비디오 코덱용 실시간 8×8 이차원 DCT 처리기의 VLSI 구현

正會員 權 用 武* 正會員 金 炯 坤*

A VLSI Implementation of Real-time 8×8 2-D DCT Processor for the Subprimary Rate Video Codec

Yong Moo KWON*, Hyoung Gon KIM* *Regular Members*

要 約 본 논문에서는 저 전송률 비디오 코덱의 영상 데이터 압축을 위한 실시간 이차원 이산여현변환기 구현에 대해 기술한다. 제안된 구조는 벡터 내적 연산의 병렬 처리에 효율적인 분산연산을 이용하였으며 동시성을 최대로 활용하고 있어 CCITT에서 제안하는 완전 CSIF 30 프레임/초의 처리성능을 만족한다. 또한 제안된 구조를 비트 수준으로 모의시험을 수행하여 CCITT에서 제안하는 IDCT 정확도 사양을 만족함을 보였다. 실제로 효율적인 VLSI 실현을 위해 설계방법론을 연구하고 SUN3/150C를 중심으로 모듈발생기 지향적 설계환경을 구축하였다. 구축된 설계환경을 이용하여 제안된 구조의 핵심모듈을 이중 금속선 2μm CMOS 기술로써 구현하였으며 설계된 이차원 DCT 칩의 크기는 약 3.9mm×4.8mm이다.

ABSTRACT This paper describes a VLSI implementation of real-time two dimensional DCT processor for the subprimary rate video codec system. The proposed architecture exploits the parallelism and concurrency of the distributed architecture for vector inner product operation of DCT and meets the CCITT performance requirements of video codec for full CSIF 30 frames/sec. It is also shown that this architecture satisfies all the CCITT IDCT accuracy specification by simulating the suggested architecture in bit level. The efficient VLSI design methodology to design suggested architecture is considered and the module generator oriented design environments are constructed based on SUN3/150C workstation. Using the constructed design environments, the suggested architecture have been designed by double metal 2 micron CMOS technology. The chip area fo designed 8×8 2-D DA-DCT(Distributed Arithmetic DCT) processor is about 3.9mm×4.8mm.

I. 서 론

*韓國科學技術研究院 집적시스템研究室
Korea Institute of Science and Technology.
論文番號 : 90-08(接受1989. 10. 6)

최근 위성통신시스템, 광통신시스템 등의 도입
에 의해 고속 디지털회선의 사용이 확대되고

또한 ISDN에 의해 회선설정이 용이한 통신시스템이 가능함에 따라 동화상통신은 새로운 서비스로 큰 기대를 받고있다. 그러나 동화상 통신의 대중화를 위해서는 많은 정보처리를 요구하는 영상신호의 압축 및 처리기술이 핵심이 되며 현재 CCITT에서는 저 전송률 비디오 코덱의 영상 데이터 압축방법으로 이산여현변환을 채택하고 있다. 또한 반도체 기술에 힘 입어 실시간 영상 데이터 압축 전용의 특수 초고집적회로의 개발이 가능해 짐에 따라 이산여현변환기 및 그 역 변환기의 초고집적회로화에 대한 연구가 계속되고 있다.⁽¹⁻¹⁰⁾

이산여현변환 처리기의 연산구조는 근본적으로 행렬 벡터 내적 승산에 근거를 두며 이의 초고집적회로화를 위한 하드웨어 구조에는 고속 DCT 처리 알고리즘을 이용한 구조 및 벡터 내적 승산 처리구조가 있다. 16×16이상의 비교적 큰 블록에 효율적인 고속 DCT 처리 알고리즘을 이용한 구조로는 Chen 알고리즘을 이용한 구조, Lee 알고리즘을 이용한 구조⁽⁶⁾ 및 회전연산기를 이용한 구조^(6,7)가 있으며 벡터 내적 승산처리구조로는 고속 병렬 승산기를 이용한 시스토릭 어레이 구조, 병렬 승산누적기(Multiplier Accumulator: MAC)처리 구조⁽⁹⁾ 및 분산연산처리 구조^(3, 4, 6, 10)를 들 수 있다.

고속 DCT 처리 알고리즘은 승산동작은 감소되었다는 이점이 있으나 VLSI 구현이라는 측면에서 보면 기본 연산구조 간의 연결(interconnection)이 증가되어 이로인한 overhead가 단점으로 지적되고 있다. 또한 벡터 내적 승산처리구조 중 시스토릭어레이 구조 및 병렬 MAC 처리구조는 요구되는 하드웨어 량에 의해 VLSI 실현이 어렵다는 단점을 갖고 있다. 한편 VLSI 지향적 처리기 구조로 메모리를 집중적으로 이용하는 구조가 최근 주목되고 있는데 그 한 방식이 분산연산을 이용하는 구조이다.⁽¹²⁾ 즉 승산동작은 분산연산 (DA: Distributed Arithmetic)이라 불리는 비트 수준의 연산동작에 의해 병렬로 처리되며 이는 LUT를 이용해 효율적으로 실현된다. 또한 분산연산은 동시(concurrent) 동작으

로 실시간 처리가 가능하며 규칙적인 구조를 가지므로 VLSI 설계가 용이하다는 장점을 갖는다.

본 논문에서는 CCITT에서 권고하고 있는 저 전송률 비디오 코덱의 핵심부인 8×8 이차원 이산여현변환 및 이의 역변환을 위한 완전주문형 초고집적회로 처리기 구현에 관한 것으로 그 하드웨어 구조로는 처리성능과 VLSI 설계의 용이성을 고려하여 분산연산을 이용한 처리구조를 바탕으로 하고 있다. 서론에 이어 제2장에서는 분산연산의 기본 개념을 설명하고 분산연산을 이용한 8×8 이차원 이산여현변환기 및 그 역변환기의 구조를 설명하고 제3장에서는 제안된 구조의 성능 및 정확도 모의시험에 대해 기술한다. 제4장에서는 VLSI 설계방법론 및 구축된 설계환경에 대해 기술하고 제5장에서는 CMOS 기술을 이용한 이산여현변환기 및 역변환기 설계에 대해 기술한다.

II. 분산연산을 이용한 이산여현변환기 구조

순방향 이산여현 변환기(FDCT)는 식(1)로 역방향 이산여현 변환기(IDCT)는 식(2)로 정의된다. 이식에서 FDCT 입력 및 IDCT의 출력은 9비트, FDCT의 출력 및 IDCT의 입력은 12비트가 될 것으로 보인다.⁽¹²⁾

$$F(u,v) = 1/4C(u)C(v) \sum_{x=0}^7 \sum_{y=0}^7 f(x,y) \cos[\pi u(2x+1)/16] \cos[\pi v(2y+1)/16] \quad \dots \quad (1)$$

$$f(x,y) = 1/4 \sum_{u=0}^7 \sum_{v=0}^7 C(u)C(v)F(u,v) \cos[\pi u(2x+1)/16] \cos[\pi v(2y+1)/16] \quad \dots \quad (2)$$

단, u,v,x,y는 0,1,...,7인 정수

C(u), C(v)는 u,v가 0이면 $1/\sqrt{2}$, 아니면 1의 값을 갖음.

한편 분산연산의 기본 개념은 다음과 같다.

입력 데이터 $f(x)$ 가 2진 보수 수치계를 이용해서 L 비트 sequence로 표현되는 경우 $f(x,i)$ 를 $f(x)$ 의 값을 나타내는 sequence의 i번째 비트를 의미한다면 그 값은 다음과 같이 나타내어진다.

$$f(x) = -f(x, L-1) 2^{L-1} + \sum_{i=0}^{L-2} f(x, i) 2^i \quad \dots (3)$$

이 식을 8×1 일차원 DCT 변환식에 적용하면 아래 식에서 보는 바와 같이 $F(u)$ 는 입력 데이터의 비트 각각에 대한 기본 변환 벡터의 합에 의해 구해진다.

$$F(u) = 1/2 C(u) \left[-\sum_{x=0}^7 f(x, L-1) \cos\{2x+1\}u\pi / 16 \right] 2^{L-1} + \sum_{i=0}^{L-2} \sum_{x=0}^7 f(x, i) \cos\{(2x+1)u\pi / 16\} 2^i \quad \dots (4)$$

N개의 입력 데이터가 동시에 인가되는 경우 각 입력 데이터의 같은 위치(weighting value)의 비트로 이루어 지는 비트패턴에 해당하는 기본 벡터의 합을 먼저 계산하여 메모리에 저장하여 놓고 N개의 입력 데이터가 이루는 비트패턴에 의해 읽어지는 결과를 누적함으로써 변환계

수를 얻을 수 있다. 이러한 분산연산 구조를 이용하는 경우 N개의 벡터 내적 연산이 입력 데이터의 비트 수 만큼의 누적 동작에 의해 구해지며 또한 병렬성과 동시성을 최대로 활용할 수 있으므로 고성능을 얻을 수 있다.

LUT의 크기는 입력 데이터의 비트 수에 따르지 않고 입력 데이터 수에 따르므로 $N=8$ 인 경우 2^8 word가 된다. 그러나 LUT의 크기는 Chen 알고리즘에 사용된 행렬분해(matrix decomposition)를 1단만 사용하면 N개의 비트 순차형 가감산기를 사용함에 의해 비트 pattern을 반으로 줄일 수 있어 LUT의 크기를 반으로 줄일 수 있다. 즉 이 방식을 이용하는 경우 $N=8$ 인 경우 16 word 메모리 N개로 LUT가 구성된다. 그림1은 행렬 분해방식을 이용한 8×1 일차원 DA-FDCT(Distributed Arithmetic Forward DCT) 처리기의 구조이며 그림2는 8×1 일차원 DA-IDCT(Distributed Arithmetic Inverse DCT) 처리기 구조이다. 그림 1에 나타낸 바와 같이 DCT의 특성상 FDCT의 경우는, 입력데이터에 대해 가감산기를 적용하며 IDCT의 경우는 누적기의 출력에 가감산기를 적용해야 한다. 그림3은 행렬분해에 의해 감축된 4×4 행렬식이다.

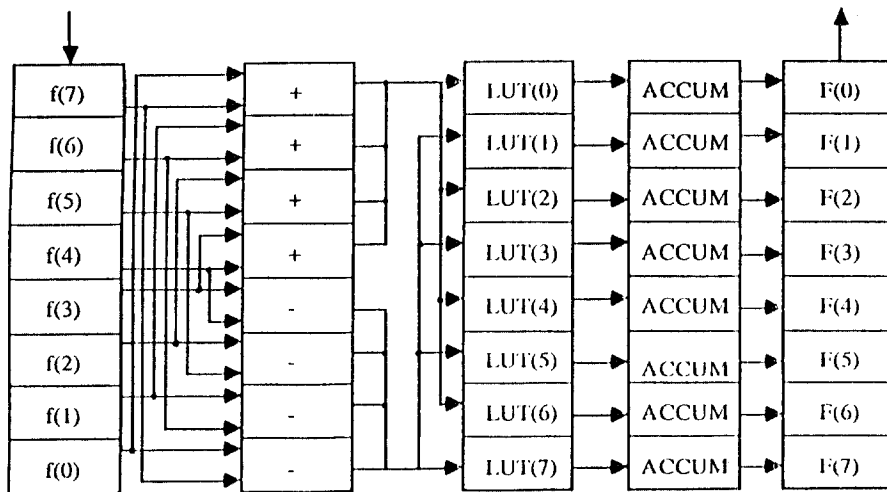


그림 1 DA-FDCT 구조
Architecture of DA-FDCT

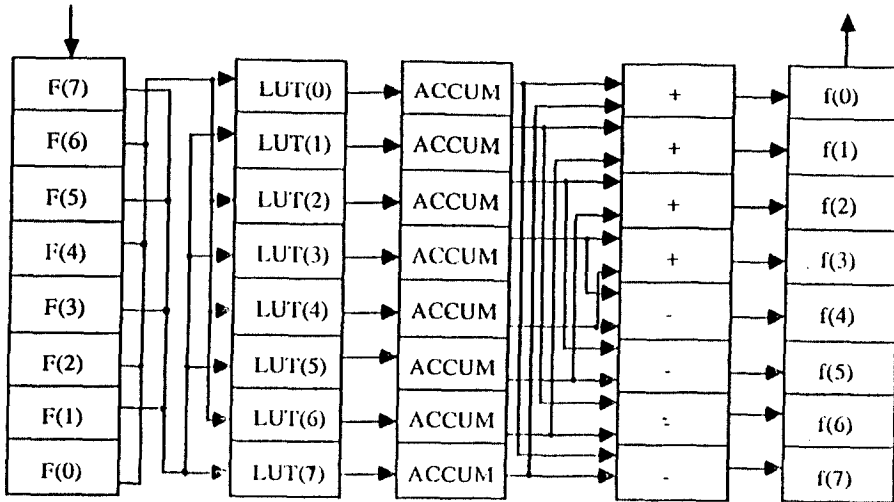


그림 2 DA-IDCT 구조
Architecture of DA-IDCT

FOR FDCT DA_LUT

$$\begin{pmatrix} C4 & C4 & C4 & C4 \\ C2 & C6 & -C6 & -C2 \\ C4 & -C4 & -C4 & C4 \\ C6 & -C2 & C2 & -C6 \end{pmatrix} \begin{pmatrix} f[0] + f[7] \\ f[1] + f[6] \\ f[2] + f[5] \\ f[3] + f[4] \end{pmatrix} \begin{matrix} \text{:LSB DA_FDCT_LUT[0]} \\ \text{DA_FDCT_LUT[2]} \\ \text{DA_FDCT_LUT[4]} \\ \text{:MSB DA_FDCT_LUT[6]} \end{matrix}$$

$$\begin{pmatrix} C7 & C5 & C3 & C1 \\ -C5 & -C1 & -C7 & C3 \\ C3 & C7 & -C1 & C5 \\ C1 & C3 & -C5 & C7 \end{pmatrix} \begin{pmatrix} f[3] - f[4] \\ f[2] - f[5] \\ f[1] - f[6] \\ f[0] - f[7] \end{pmatrix} \begin{matrix} \text{:LSB DA_FDCT_LUT[1]} \\ \text{DA_FDCT_LUT[3]} \\ \text{DA_FDCT_LUT[5]} \\ \text{:MSB DA_FDCT_LUT[7]} \end{matrix}$$

FOR IDCT DA_LUT

$$\begin{pmatrix} C4 & C2 & C4 & C6 \\ C4 & C6 & -C4 & -C2 \\ C4 & -C6 & -C4 & C2 \\ C4 & -C2 & C4 & -C6 \end{pmatrix} \begin{pmatrix} F[0] \\ F[2] \\ F[4] \\ F[6] \end{pmatrix} \begin{matrix} \text{:LSB DA_IDCT_LUT[0]} & [0]+[7]-f[0] \\ \text{DA_IDCT_LUT[1]} & [1]+[6]-f[1] \\ \text{DA_IDCT_LUT[2]} & [2]+[5]-f[2] \\ \text{:MSB DA_IDCT_LUT[3]} & [3]+[4]-f[3] \end{matrix}$$

$$\begin{pmatrix} C7 & -C5 & C3 & -C1 \\ C5 & -C1 & C7 & C3 \\ C3 & -C7 & -C1 & -C5 \\ C1 & C3 & C5 & C7 \end{pmatrix} \begin{pmatrix} F[1] \\ F[3] \\ F[5] \\ F[7] \end{pmatrix} \begin{matrix} \text{:LSB DA_IDCT_LUT[4]} & [3]-[4]-f[4] \\ \text{DA_IDCT_LUT[5]} & [2]-[5]-f[5] \\ \text{DA_IDCT_LUT[6]} & [1]-[6]-f[6] \\ \text{:MSB DA_IDCT_LUT[7]} & [0]-[7]-f[7] \end{matrix}$$

그림 3 행렬분해에 의해 감축된 4×4 DCT 행렬
4×4 DCT matrix using matrix decomposition

또한 DCT계수들을 이용하여 구해진 LUT의 위치 순서도 표시하였다.

본 연구에서는 각 요소가 9비트로 표현되는 8×8크기의 이차원 DCT 처리기 구조를 설계하였다. 이차원 DCT 처리기는 길이 N인 두개의 일차원 DCT 처리기로 실현되며 8×1 DCT 처리기의 핵심부분은 16 word×16 bit로 구성되는 8개의 LUT와 각 LUT의 결과를 사용하는 비트 수 만큼 누적시키는 8개의 누적기로 이루어진다. 또한 입출력 접속을 위해 N개의 지연선로(delay line)와 각 지연선로 요소의 출력을 비트 순차적으로 변환 시키는 N개의 parallel-in serial-out 레지스터 및 LUT의 크기를 감소하기 위해 이용되는 순차형 가감산기가 필요하다. 이러한 일차원 DCT 처리기를 칩화하는 경우 N개의 누적기 출력이 병렬로 외부로 전달되어야 하므로 비경제적인 구조를 갖게된다. 따라서 입출력 수를 줄이고 주변에 필요한 회로를 줄이기 위해 순차형 입출력을 갖도록 두개의 일차원 DCT 처리기가 포함된 칩을 설계하는 것이 효율적이며 이를 위해서 행렬 변환부도 포함시켰다.

한편 현재 설계된 칩은 FDCT와 IDCT를 별도의 칩으로 설계하였으나 주후 입출력 핀 및 내부 버스의 제어 기능을 포함시켜 FDCT와 IDCT 공용의 칩으로 개선하고자 한다.

III. 성능 및 정확도 모의시험

DCT 실시간 구현을 위해서 필요한 처리속도는 화상의 해상도 및 전송률에 따라 달라지며 이에 대한 기준으로서 CCITT에서는 표1과 같이 실시간 DCT 처리를 위한 블록 당 최대 처리시간

표 1. 실시간 DCT 처리를 위한 블록당 최대 처리시간
Specification for 8×8 subblock DCT processing time

해상도	Full CSIF	4/9 CSIF	1/4 CSIF
30 (frames/sec)	14 μ s	31.5 μ s	56 μ s
15 (frames/sec)	28 μ s	63.0 μ s	112 μ s
7.5(frames/sec)	56 μ s	126 μ s	224 μ s

간 사양을 제시하였다.⁽¹¹⁾

표 1에 표시한 바와 같이 완전(full) CSIF(Common Source Input Format)의 해상도를 갖는 경우 44×36 블록의 밝기성분과 각각 22×18블록의 색차 신호를 최대전송률인 초당 30화면을 보낼 경우 블록당 처리시간은 14 μ sec로 계산된다.

설계된 DA-DCT(Distributed Arithmetic DCT: DA-DCT는 DA-FDCT 및 DA-IDCT를 모두 포함하는 의미로 표기하였다.)는 Row/Column 일차원 변환을 pipeline처리하며 데이터의 입출력 또한 pipeline 처리한다. 또한 8×1 DCT를 병렬처리해준다. 이와 같은 경우 처리 성능은 다음과 같다. 즉 1 cycle time을 40 nsec라 할때 8×1 DCT를 처리하는데 걸리는시간 LUT의 출력을 누적하는데 17 cycle, 누적된 출력을 출력하는데 16 cycle 즉 총 33 cycle로서 1.32 μ sec가 걸린다. 따라서 8×8 DCT에는 10.56 μ sec가 걸리므로 30 F/S의 완전 CSIF 사양을 만족하게 된다.

DA-DCT 비트 수준의 정확도 모의시험을 기능 단계에서 수행하였다. CCITT p×64 kbit/s 비디오 코덱의 IDCT 정확도 사양⁽¹³⁻¹⁴⁾은 다음과 같다. 즉 3가지 데이터 범위(-5~5, -256~255, -300~300)에 대해 64-bit 부동소수점 연산 정확도로서 이차원 FDCT를 행렬 내적승산에 의해 결과를 구하고 그 결과를 12비트로 clipping하여 IDCT 입력 데이터를 만든 후 이 IDCT 입력 데이터에 대해 다시 64-bit 부동소수점 연산정확도로서 이차원 IDCT를 행렬 내적승산에 의해 결과를 구하고 그 결과를 다시 9-bit로 clipping하여 reference IDCT 데이터를 구한다. 한편 제안된 구조에 대해 앞서 구한 IDCT 입력 데이터를 이용하여 이차원 IDCT를 수행하고 그 결과를 9 bit로 clipping하여 test IDCT 데이터를 구한다. 앞에서 구한 reference IDCT 데이터에 대해 test IDCT 데이터는 IDCT 정확도 사양을 만족해야 한다.

본 연구에서는 제안된 DA-IDCT 구조의 비트 수준의 IDCT 정확도 모의시험을 위해 그림 3

의 4×4 행렬을 이용하여 그림 4와 같은 LUT 데이터를 만드는 DA-DCT-LUT() 함수와 실제 LUT와 누적기를 이용하여 분산연산을 수행하는 DA-ACCUM() 함수를 작성하였으며 그림 5에서 DA-IDCT-LUT의 비트 수를 변수로 하여 IDCT 정확도 모의시험을 수행하였다. 모의시험을 수행해본 결과 CCITT IDCT 정확도 사양을 만족하기 위해서는 DA-IDCT-LUT는 최소한 15 bit

를 가져야했다. 그림5에서 (2.x)은 소숫점 이상 2 bit 소숫점 이하 x bit를 의미한다. 본 연구에서는 VLSI 설계를 고려하여 DA-DCT-LUT bit 수를 16 비트(2.14)로 하였다. 그림 6-1은 DA-IDCT-LUT bit 수가 16비트 (2.14)이며 데이터 범위가 -256~ +255인 경우의 모의시험 결과이다. 또한 그림 6-2는 DA-IDCT의 LUT bit 수와 동일하게 16비트(2.14)를 사용한 경우

```

DA_DCT_LUT(TYPE, ROM_NO, ADDR)
char TYPE;
int ROM_NO, ADDR;
{
    double cf_sum = 0;
    int ADDR_BIT = 4; /* Number of address bit */
    int i, SF = exp(rom_bsize - 2);
    char BIT_SEQ[ADDR_BIT + 1], *ltoBIN();

    /* Make Binary Bit Pattern of ROM_ADDR */
    strcpy (BIT_SEQ, ltoBIN(ADDR, ADDR_BIT));

    for ( i = 0; i < ADDR_BIT; i++){

        if ( (TYPE == 'F') && (BIT_SEQ[i] == '1')) { /* For FDCT */
            if((ROM_NO % 2) == 0) cf_sum += DA_fCF1[ROM_NO/2][i];
            else cf_sum += DA_fCF2[ROM_NO/2][i];
        }

        else if ((TYPE == 'I') && (BIT_SEQ[i] == '1')) { /* For IDCT */
            if (ROM_NO < 4) cf_sum += DA_iCF1[ROM_NO][i];
            else cf_sum += DA_iCF2[ROM_NO - 4][i];
        }
    }
    return( rounding(cf_sum * SF) );
}

```

```

DA_ACCUM(row, col, bit_no)
int row, col, bit_no;
/* output : F[row, col], input : f[0, col] ~ f[N-1, col] */
{
    int add, i, sum = 0;

    for(i = 0; i < (bit_no - 1); i++){
        addr = mkaddr(col, i);
        sum += (DA_DCT_LUT[row][addr] * exp(i));
    }
    addr = mkaddr(row, col, bit_no - 1);
    sum -= (DA_DCT_LUT[row][addr] * exp_2(bit_no - 1));
}
return (sum);
}

```

그림 4 BIT 수준의 DA-DCT 기능 모의시험을 위한 분산연산 처리부의 프로그램
Program for DA-DCT functional simulation in bit level

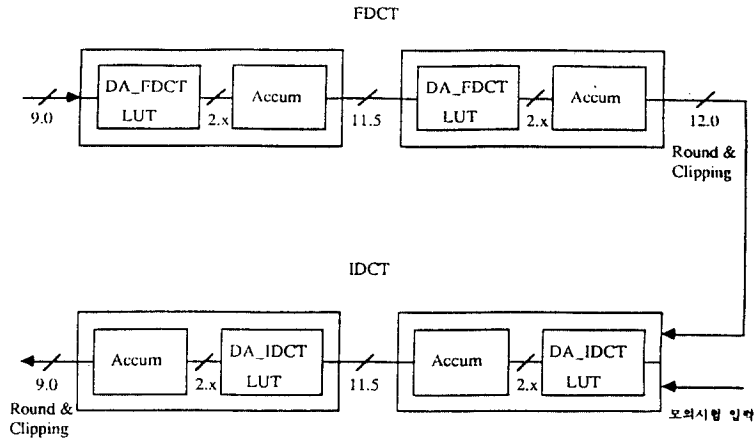


그림 5 분산연산을 이용한 DCT처리
DCT based on Distributed Arithmetic

- * 데이터 범위 : -256 to +255
- * DA_IDCT_LUT bit 수 : (2,14)
- * 데이터 수 : 640000 random integer pixels

Mean Error versus pixel Location (x 1000)

2 (0)	1 (1)	0 (0)	1 (0)	0 (0)	1 (0)	1 (0)	1 (0)
0 (-1)	-2 (0)	1 (1)	0 (1)	0 (0)	1 (0)	0 (0)	1 (0)
1 (1)	1 (1)	0 (-1)	1 (2)	0 (0)	0 (0)	0 (0)	0 (1)
0 (3)	0 (-2)	1 (0)	-1 (1)	1 (0)	1 (0)	0 (0)	0 (0)
0 (1)	0 (0)	0 (0)	1 (0)	0 (0)	0 (1)	0 (1)	1 (0)
1 (1)	0 (-1)	1 (0)	1 (0)	0 (1)	0 (0)	0 (0)	0 (0)
1 (0)	0 (1)	-1 (0)	-2 (1)	1 (1)	0 (0)	0 (0)	-1 (1)
1 (0)	0 (1)	0 (0)	0 (0)	0 (0)	0 (1)	0 (-2)	0 (1)

Mean Square Error versus pixel Location (x 1000)

9 (10)	10 (9)	9 (8)	9 (7)	7 (8)	10 (7)	8 (7)	8 (8)
9 (9)	9 (8)	8 (8)	10 (8)	8 (8)	8 (11)	8 (7)	9 (8)
9 (7)	10 (8)	8 (8)	9 (7)	9 (9)	9 (7)	10 (9)	9 (8)
9 (9)	7 (8)	8 (8)	9 (8)	8 (8)	10 (9)	9 (10)	8 (10)
8 (8)	10 (9)	9 (9)	9 (8)	7 (8)	10 (8)	8 (8)	10 (10)
8 (8)	9 (11)	8 (8)	9 (7)	7 (7)	7 (8)	10 (9)	9 (8)
9 (9)	9 (9)	9 (8)	9 (9)	8 (9)	7 (7)	8 (7)	9 (8)
9 (8)	8 (6)	7 (9)	7 (8)	8 (9)	9 (8)	8 (8)	8 (8)

	random data	opposite sign
Peak Error :	1	1
Number of Error of 1 :	5482 (0.86%)	5239 (0.82%)
Number of Error above 1 :	0	0
Peak Pixel Mean Square Error :	0.01	0.01
Overall Mean Square Error :	0.0086	0.0082
Peak Pixel Mean Error (in magnitude) :	0.0023	0.0027
Overall Mean Error :	0.000063	0.000020

- All-zeros in produce all-zeros out.

그림6-1 CCITT IDCT 정확도 모의시험 결과
Result of CCITT IDCT accuracy simulation

- * 데이터 범위 : -256 to +255
- * DA_IDCT_LUT bit 수 : (2,14)
- * 데이터 수 : 640000 random integer pixels

Mean Error versus pixel Location (x 1000)

2 (0)	1 (1)	0 (0)	1 (0)	0 (0)	1 (0)	1 (0)	1 (0)
0 (-1)	-2 (0)	1 (-1)	0 (1)	0 (0)	1 (0)	0 (0)	1 (0)
1 (1)	1 (1)	0 (-1)	1 (2)	0 (0)	0 (0)	0 (0)	0 (1)
0 (3)	0 (-2)	1 (0)	-1 (1)	1 (0)	1 (0)	0 (0)	0 (0)
0 (1)	0 (0)	0 (0)	1 (0)	0 (0)	0 (1)	0 (1)	1 (0)
1 (1)	0 (-1)	1 (0)	1 (0)	0 (1)	0 (0)	0 (0)	0 (0)
1 (0)	0 (1)	-1 (0)	-2 (1)	1 (1)	0 (0)	0 (0)	-1 (1)
1 (0)	0 (1)	0 (0)	0 (0)	0 (0)	0 (1)	0 (-2)	0 (1)

Mean Square Error versus pixel Location (x 1000)

9 (10)	10 (9)	9 (8)	9 (7)	7 (8)	10 (7)	8 (7)	8 (8)
9 (9)	9 (8)	8 (8)	10 (8)	8 (8)	8 (11)	8 (7)	9 (8)
9 (7)	10 (8)	8 (8)	9 (7)	9 (9)	9 (7)	10 (9)	9 (8)
9 (9)	7 (8)	8 (8)	9 (8)	8 (8)	10 (9)	9 (10)	8 (10)
8 (8)	10 (9)	9 (9)	9 (8)	7 (8)	10 (8)	8 (8)	10 (10)
8 (8)	9 (11)	8 (8)	9 (7)	7 (7)	7 (8)	10 (9)	9 (8)
9 (9)	9 (9)	9 (8)	9 (9)	8 (9)	7 (7)	8 (7)	9 (8)
9 (8)	8 (6)	7 (9)	7 (8)	8 (9)	9 (8)	8 (8)	8 (8)

	random data	opposite sign
Peak Error :	1	1
Number of Error of 1 :	5482 (0.86%)	5239 (0.82%)
Number of Error above 1 :	0	0
Peak Pixel Mean Square Error :	0.01	0.01
Overall Mean Square Error :	0.0086	0.0082
Peak Pixel Mean Error (in magnitude) :	0.0023	0.0027
Overall Mean Error :	0.000063	0.000020

- All-zeros in produce all-zeros out.

그림6-2 CCITT IDCT 정확도 모의시험 결과
Result of CCITT IDCT accuracy simulation.

의 DA-FDCT 정확도 모의시험 결과이다. 본 연구에서 수행한 모든 모의시험결과는 참고문헌 (1)에 있다.

IV. DA-DCT 설계방식 및 설계환경

DA-DCT 초고집적회로의 구현을 위한 설계환경 및 구현 방식은 다음과 같다. 먼저 제작공정이 확정되지 않았으므로 제작공정에 따라 설계변경이 용이하도록 하여야 하므로 모듈발생기 개념⁽¹⁵⁾을 도입하여 설계 규칙(design rule) 및 설계조건들을 변수화하여 이용한다.⁽¹⁶⁻¹⁷⁾ 실제 핵심부분의

설계는 CIF(Caltech Intermediate Form)를 출력할 수 있는 모듈설계언어 (MDL: Module Design Language)를 C언어를 이용해 정의하였으며 이를 이용해 기본셀을 계층적 구조로 정의하여 이용하였다. 이들은 CIF 형태의 출력을 내며 CMOS 제작공정에 따라 필요한 layer들을 자동 생성하며 설계를 용이하게 하기 위하여 abstract layer 개념을 이용하였다. 또한 출력된 결과를 plotter를 이용해 그리기 위하여 CIF 계층제거 프로그램 및 plotter와의 접속 프로그램을 개발하였다. 구축된 설계환경은⁽¹⁸⁾ 그림7과 같다. 회로의 모의시험은 Cray2S에서 HSPICE와 IBM-PC386에서 PSPICE를 이용하여 수행하였

다.

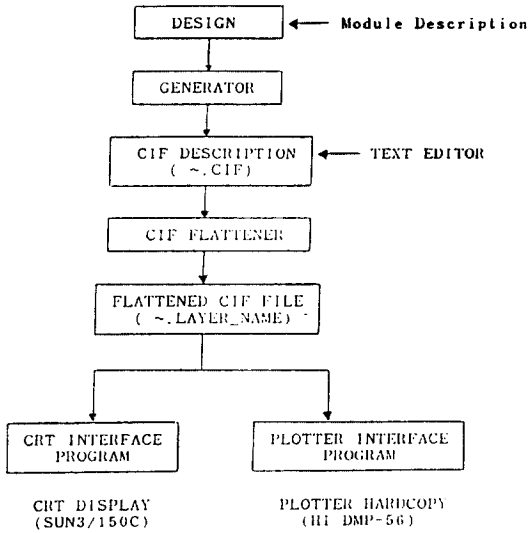


그림 7 SUN3/150C를 중심으로 구축된 설계환경
Constructed design environments based on SUN 3/150C

V. DA-DCT CMOS 모듈의 설계 및 검증.

본 장에서는 제안된 8×8 이차원 DA-DCT 처리기를 실현하기 위해 앞 장에서 설명한 설계 방법 및 환경을 이용하여 수행한 핵심모듈의 설계 및 모의시험에 대해 기술한다. 핵심모듈을 설계하기에 앞서 2μm 선폭을 사용하는 이중 금속선 CMOS 회로의 기본 특성을 조사하고 이를 기본으로 하여 중요 핵심부분을 설계하였다. 중요핵심모듈은 입력지연 및 데이터 변환부, DA-DCT Look-Up Table용 ROM, 순차형 누적기, 출력 데이터 변환 및 지연부, 행렬 변환부, 제어부 등으로 구성된다. 그림 8은 8×8이차원 DA-FDCT의 전체 layout을 나타낸 것으로 내부 주요 모듈을 bounding box로 표시하였다. 표 2는 설계된 DA-FDCT 특성표이다.

표 2 DA-FDCT 특성표
Characteristics of DA-FDCT.

Technology	2-μm 이중 금속선 CMOS
Core 크기	3.9 x 4.8 mm ²
총 Tr 갯수	약 70,000 개
최대 동작속도	20 MHz

입력지연 및 데이터 변환부;

입력지연부는 순차적으로 입력되는 화소 값들을 병렬로 DA DCT 변환부에 인가하기 위해 필요하며 사용되는 위치에 따라 9/12/16 비트로 구성되는 word를 8번 shift하면서 저장하는 기능을 갖는다. 입력 데이터 형태 변환부는 비트

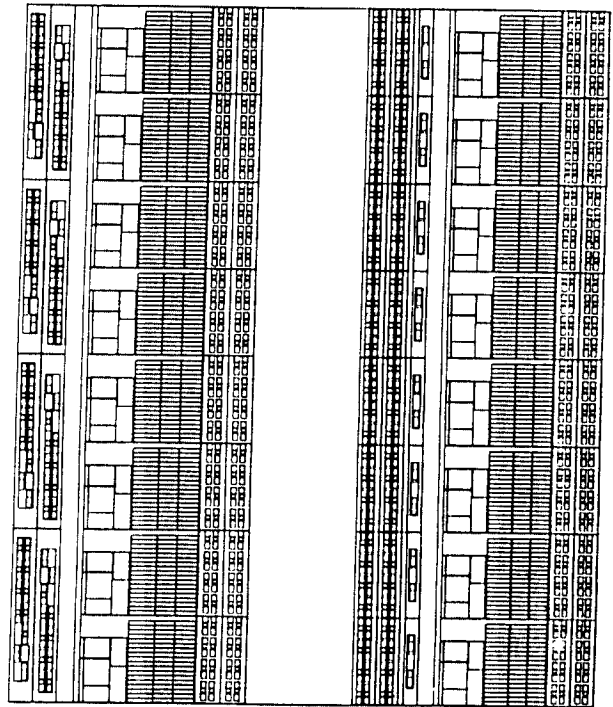


그림 8 8×8 이차원 DA-FDCT 처리기 배치도
(칩 크기는 3.9mm×4.8mm: 단 주변 pad는 제외)
8×8 2-D DA-FDCT processor layout
(The chip area is 3.9mm×4.8mm excluding pad)

병렬로 입력되는 화소값들을 분산연산 처리시 필요한 비트순차형으로 바꾸는 기능을 수행하며

기본적으로는 parallel-in serial-out 래치로 구성된다. 본 설계에서는 CMOS static 래치 회로를 이용하여 지연소자 및 데이터 변환소자를 동시에 실현하는 입력 기본셀을 정의하고 이들을 이용하

여 입력부를 구성하였다. 입력지연부와 word-parallel bit-serial 변환부를 동시에 형성하기 위한 입력 기본셀은 그림9와 같이 두개의 D래치와 2:1 multiplexer 로 구성하였다.

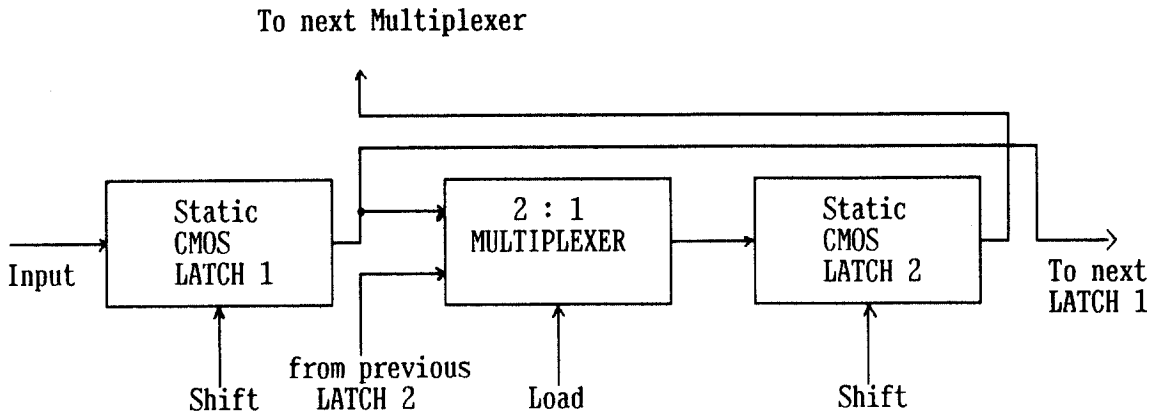


그림 9 입출력 기본 셀 구성도
Block diagram of in/out basic cell.

가감산기부:

행렬분해를 위해 사용한 비트 순차형 가산기는 전가산기부와 carry 및 sum 신호를 저장하는 레지스터로 구성되며 전가산기부는 동작성능 및 면적을 작게 차지하는 N-P CMOS 동적 논리 회로를 이용하였다. N-P CMOS 동적 논리 회로를 이용한 전가산기부는 순차형 누적기에도 사용된다. 또한 순차형 가산기의 구성은 순차형 가산기의 구성과 동일하나 가감산 동작 초기조건과 종 래치의 반전된 신호를 이용하여 2진 보수 수치계의 가산 동작을 실현한다. 이를 위해 carry 신호를 저장하는 래치는 초기에 preset 가능하여야 하며 종 래치의 출력은 반전된 출력도 함께 제공하여야 한다.

DA-DCT Look-Up Table 부:

LUT는 ROM, PLA, 다단 논리 게이트 등으로 실현가능하나 ROM을 이용한 구조가 가장 효율적이다. ROM은 크게 나누어 ROM 어레이부, 어드레스 디코더부 및 출력 감지증폭부로 구성되

며 ROM 셀은 한 비트 기억장치로서 하나의 트랜지스터를 사용할 수 있으며 일반적으로 NOR 어레이로 설계된다. ROM의 프로그램 방법으로는 접촉형(contact) 프로그램, 트랜지스터의 유무, 이온주입(implantation)등이 있으나 word 선의 용량이 적어 동전원 소모가 적게되는 트랜지스터의 유무에 의한 프로그램 방법을 사용하였다. 어드레스 디코더부는 동작속도를 고려하여 동적(dynamic) NOR 디코더 방식을 사용하였으며 비트선의 출력 버퍼로서 요구되는 감지증폭기에는 기하학적 비를 고려한 반전기를 이용하여 설계하였다. 이러한 LUT는 ROM 발생기에 의해 설계되었으며 ROM 발생기의 입력 변수로는 word 수, 비트 수, 감지증폭기의 β 비를 갖는다. ROM 발생기에 의해 설계된 ROM 어레이는 16 words × 16비트로 구성되어 있으며 word 선은 poly이며 비트선은 금속선으로 구성하였다. 또한 ROM에 대한 spice 모의시험 수행 결과 그림 9와 같이 설계사양인 40ns access 시간을

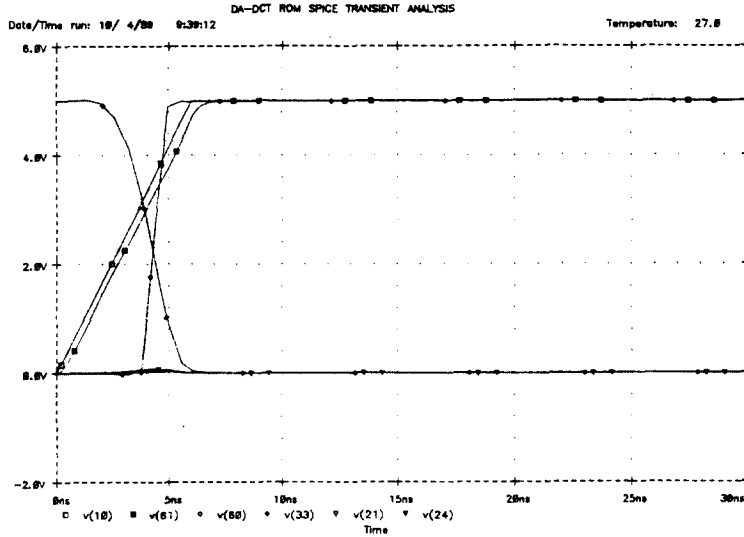


그림 9 LUT spice 모의시험
Spice simulation of LUT

만족하였다.

Carry save 순차형 누적기부:

Carry save 순차형 누적기는 carry save 수치계를 이용하여 redundant 상태를 이용하여 carry 신호의 전파를 비트수준으로 고정하는 누적기 구조로 누적되는 수의 비트수가 많을 수록 효율적인 동작을 하게된다. 즉 각 자리수는 (0,1,2)의 상태를 가질수 있으며 이를 위해 두 비트가 사용되므로 각 비트마다 하나의 redundant 상태를 가지며 이를 이용하면 carry 신호의 전파없이 누적연산을 수행할 수 있으며 최종출력을 얻기 위해서는 이러한 carry save 수치계를 binary 수치계로 바꾸기 위한 병렬 가산기가 필요하다. 본 설계에서는 이러한 수치계의 변환을 병렬가산기를 이용하지 않고 가산-shift로 실현하였다. 사용된 전가집산기 셀은 N-P CMOS 논리회로를 이용하였다.

데이터 출력부 및 제어부:

데이터 출력부는 순차형 누적기의 출력을 비트 병렬 형태로 바꾸어 주는 serial-in parallel-out 데이터 변환부와 출력 데이터들을 순차적으로 외부로 전달하는 출력지연선으로 구성된다. 데이

타 변환소자 및 지연소자를 동시에 실현하는 출력 기본셀은 입력 기본셀과 마찬가지로 두개의 CMOS D 래치와 2:1 multiplexer로 구성된다.

제어부는 내부의 제어신호 및 외부와의 접속을 위한 제어신호로 구성되며 내부제어 신호는 qualified clock 회로를 이용한다. 외부와는 비동기식에 의한 handshake 방식이 고려되고 있으며 외부에 사용되는 메모리 제어신호도 고려대상이 된다. 이러한 제어신호에 대한 상세한 설계는 계속 연구되어야 할 사항이다.

행렬 변환부:

이차원 DCT를 하나의 칩에 실현 하고자 하는 경우 앞서 설명한 대로 행렬변환부가 포함되어야 한다. 행렬 변환부는 입출력이 순차적으로 주어지는 경우 write 어드레스WA와 read 어드레스RA의 관계는 $RA = (WA \times N) \text{ Mod } N$ 로 주어지 N이 8인 경우 3비트 위치만 shift 시켜 이용할 수 있다. 한편 분산연산을 이용하는 구조는 N개의 출력이 동시에 발생되므로 이를 순차형으로 변환하여 이용하는 경우 parallel-in serial-out 변환기와 serial-in parallel-out 변환기가 이중으로 사용되어 칩면적이 많이 필요하게 된다. 본

연구에서는 N개의 입력과 N개의 출력을 동시에 처리하는 행렬변환기 구조로서 RAM 모듈을 사용한다. 행렬 변환부는 64-word×16-bit 크기의 RAM으로 하였으며 RAM 셀은 6 트랜지스터 형태를 사용하였다.¹⁰⁾ 또한 read 회로로서 감지증폭기를 사용하였으며 write 회로에는 래치를 사용하였다. 어드레스 디코더로는 동작속도 및 전력소모 등을 고려하여 동적 NOR 디코더를 사용하였다.

VI. 결 론.

본 연구에서는 p×64 kbps 급의 저전송률 비디오 코덱 시스템에서 실시간 처리의 핵심대상이 되는 8×8 이차원 이산여현변환기 및 역변환기의 VLSI 실현을 위해 분산연산을 이용한 DA-FDCT 및 DA-IDCT 구조를 제안하였다. 제안된 구조는 CCITT에서 제안하는 실시간 처리성능을 만족하며 비트 수준으로 정확도 모의 시험을 수행하여 IDCT 정확도를 만족하도록 설계하였다. 제안된 구조는 초고집적회로 기술을 이용해 모듈발생기로 실현 하였으며 이를 위해 모듈설계언어 및 CIF(Caltech Intermediate Form) 처리기를 개발하였다. 설계된 DA-FDCT 및 DA-IDCT 발생기는 ROM 발생기를 중심으로한 DA-DCT Look-Up Table 모듈, LUT의 출력을 누적하기 위한 carry save 순차형 누적기 모듈, 입력변환 및 지연부모듈, 행렬분해를 위한 가감산기 모듈, 출력 변환 및 지연부모듈, 행렬 변환을 수행하는 행렬분환부 등의 모듈로 구성된다.

설계된 DA-FDCT 및 DA-IDCT는 앞으로 특정 공정에 따른 조정작업이 필요할 것이며 이를 위해 제조공정 및 설계규칙에 관련된 조건들을 변수화하여 사용되는 조건에 쉽게 대처하도록 하였다.

參 考 文 獻

1. 영상압축용 특수 IC 설계 및 개발, 연구보고서, 한국과학기술연구원, 1989.
2. 권용무, 김형근, 분산연산을 이용한 실시간 이차원 DCT 처리기의 VLSI 구현, 전자공학회 통신공학회 합동 영상통신분과 연구발표회, pp.95-98, Aug. 1989.
3. M.T.Sun, T.C. Chen, A.M. Gottlieb, VLSI Implementation of a 16×16 Discrete Cosine Transform Chip, IEEE Trans. on CAS, Vol.36, No.4 pp.610-617, April 1989.
4. M.T.Sun, L.Wu, and M.L.Liou, A Concurrent Architecture for VLSI Implementation of Discrete Cosine Transform, IEEE Trans. on Circuit and Systems, Vol. CAS-34, No. 8, pp. 992-994, Aug. 1987.
5. A.Ligtenberg, J.H. O'Neil, A Single Chip Solution for an 8×8 Two-Dimensional DCT, Proc. of the IEEE 1987 Int'l, Symp. on Circuit and Systems, Dunfey City Line Hotel, Philadelphia, pp.1128-1131, May 1987.
6. B.Sikstrom, L.Wanhammer, M.Afghhi and J.Pencz, A High speed 2-D discrete cosine transform chip, Integration the VLSI journal, Vol.5, pp.159-169, 1987.
7. M.Vetterli, A. Ligtenberg, A Discrete Cosine Transform Chip, IEEE Journal on Selected Areas in Comm., Vol. SAC-4, No. 1, pp.49-61, Jan. 1986.
8. Data Sheet: STV3208 8×8 DCT, SGS THOMSON Microelectronics, May 1989.
9. Advance Information: IMS A121 2-D Discrete Cosine Transform Image Processor, inmos, June 1989.
10. TCAD: a 27 MHz 8×8 Discrete Cosine Transform Chip, CCETT, 1989.
11. Description of Ref. Model 8, CCITT SGXV Working Party XV/4 Document #525, June 1989.
12. S.A. White, Applications of Distributed Arithmetic to Digital Signal Processing: A Tutorial Review, IEEE ASSP Magazine, Vol. 6, No. 3, pp.4-19, July 1989.
13. Precision for IDCT Calculation and Refresh Cycle, CCITT SGXV Working Party XV/1 Documents #321R, March 1988.
14. Specification of Inverse DCT, CCITT SGXV Working Party XV/1 Annex 3 to Document #445R,

- p.34, Dec. 1988.
15. D.D. Gajski, Silicon Compilation, Addison Wesley, pp. 49-94, 1988.
 16. Amar Mukherjee, Introduction to nMOS & CMOS VLSI Systems Design, *Prentice-Hall*, 1986.
 17. Neil Weste et al., Principles of CMOS VLSI Design, *Addison-Wesley*, 1985.
 18. 권용무, 김형곤, 모듈발생기 지향적 VLSI 설계환경 구축, 한국정보과학회, Vol. 16, No.6, pp.539-549, Nov. 1989.
 19. D. A. Pucknell, K. Eshraghian, Basic VLSI Design, *Prentice-Hall*, 1988.



權用武(Yong Moo KWON) 正會員
1957年4月27日生
1980年2月:漢陽大 電子科 卒業
1983年2月:漢陽大學院 電子科卒業(工學碩士)
1986年3月~現在:漢陽大學院 電子科 博士課程
1983年1月~現在:韓國科學技術研究院 集積시스템研究室 研究員



金炯坤(Hyong Gon KIM) 正會員
1952年4月5日生
1974年2月:韓國航空大 電子科 卒業
1982年8月:英國 Kent 大 電子科 工學 碩士
1985年5月:英國 Kent 大 電子科 工學 博士
1977年10月~現在:韓國科學技術研究院 集積시스템研究室長