

블럭정합알고리즘의 고속처리를 위한 VLSI 어레이의 설계

正會員 이 수 진*, 성 길 영**

Design of VLSI Array for High Speed Processing of Block Matching Algorithm

Su-Jin Lee*, Kil-Young Sung** *Regular Members*

요 약

본 논문에서는 블럭정합알고리즘의 고속처리를 위한 VLSI 어레이를 설계하였다. 블럭정합알고리즘의 단일할당 코드로부터 데이터존그래프를 구하고, 적당한 방향으로 투영시켜 이차원의 VLSI 어레이를 구하였다. 영상의 각 탐색블럭이 행과 열로 중첩되므로 구해진 이차원 어레이를 연결하여 어레이 프로세서를 구현하였다. 어레이프로세서 내의 각 처리요소들은 데이터를 재사용하므로 메모리 액세스와 입출력 핀의 수를 줄일 수 있다. 제안된 어레이프로세서의 동작을 컴퓨터시뮬레이션을 통하여 검증하였다.

ABSTRACT

In this paper, the VLSI array for high speed of Block Matching Algorithm(BMA) is designed. First of all the single assignment code is derived from the sequential algorithm of BMA, and then the three dimensional data dependance graph(DG) is derived from the single assignment code. The two dimensional VLSI array is derived by projecting the obtained DG along proper direction.

Since the search blocks are overlapped with columns as well as rows, the column data can be pipelined in BMA. The two dimensional VLSI array can be derived by cascading the obtained arrays. The PEs of this architecture reuse the columns and rows data overlapped, so the number of memory access and I/O pin can be reduced. The computer simulation is implemented for verifying performance of the correct operations and analysing the designed array.

*부경대학교 전자공학과
**경상대학교 정보통신공학과, 정보통신연구센터
論文番號:97454-1213
接受日字:1997年 12月 13日

I. 서 론

최근 컴퓨터 시스템의 발달과 사용자들의 다양한 욕구에 의해 멀티미디어의 사용이 증가하고 있다. 음성, 그림, 동영상 등의 멀티미디어 정보는 기존의 문자 정보에 비해 많은 양의 데이터를 갖는다. 제한된 전송 선로를 통한 HDTV(High Definition Television)의 디지털 방송이나 분산시스템에서의 멀티미디어 정보의 취득과 전송을 위해서는 데이터의 압축기술이 필수적으로 요구된다. 연속되는 프레임으로 이루어져 많은 양의 데이터를 갖는 동영상의 압축은 이동보상형 DPCM, 이산여현부호화(DCT transform coding) 기법과 엔트로피 부호화 기법을 병행하여 이루어진다.

이동벡터를 찾는 이동보상형 DPCM 단계는 연속된 두 영상의 각 블럭에 대하여 비교를 반복하므로 많은 계산량이 요구된다. 따라서 실시간 동영상 압축과 저장을 위하여 전체 부호화 시간의 대부분을 차지하는 이동벡터의 계산을 고속화하기 위한 VLSI 어레이 설계에 대한 연구가 많이 이루어지고 있다.

1989년 Yang^[1]은 이동보상 블럭정합알고리즘의 고속화를 위한 몇 가지 VLSI 어레이를 제안하였으며, Komarek^[2]는 시스토키어레이를 제안하였다. 1992년 Hsieh^[3]은 쉬프트레지스터를 이용하여 데이터를 직렬로 입력하는 시스토키어레이를 제안하였다. 이 구조는 입출력 핀의 수는 많이 줄일 수 있으나 계산시간이 길다는 단점이 있다. 1993년 Chan^[4]은 3단계 계층형탐색 알고리즘의 VLSI를 제안하였으며, Jehng^[5]은 트리구조의 VLSI를 제안하였다. 트리구조는 계산시간의 향상을 가져오지만 입출력 핀의 수가 많다는 단점을 갖는다. 1995년 Yeo^[6]은 탐색블럭(search block)의 일부가 여러 기준블럭(reference block)에 대해 반복해서 사용되는 성질을 이용하여 입출력 핀의 수가 작으며 계산속도의 향상을 갖는 구조를 제안하였다. 그러나 이 구조는 탐색블럭의 데이터들을 글로벌패스(Global Path)를 통해 전파하므로 실제 VLSI 구현시 어려움이 따른다.

본 논문에서는 블럭정합알고리즘에 근거하여 입출력 핀수의 증가를 제한하며 반복적으로 사용되는 탐색블럭(Search Block)의 열을 재사용하도록 시스토키어레이를 제안한다. 설계한 시스토키어레이는 이동벡터의 계산복잡도가 $O(N^6)$ 에서 $O(N^3)$ 으로 향상되

었으며, $O(N)$ 개의 입출력 핀을 갖는다.

본 논문에서는 블럭정합알고리즘에 대하여 고찰하고 기존의 순차적인 블럭정합알고리즘에서 데이터의존성을 분석하여 데이터의존그래프를 구하고 시간 및 공간 변환을 통해 시스토키어레이를 설계한다. 설계한 시스토키어레이와 그 처리요소의 정확성 및 유효성을 검증하기 위해 컴퓨터시뮬레이션을 수행하고 결과를 고찰하였다.

II. 블럭정합알고리즘

블럭정합알고리즘은 현재의 프레임을 $N \times N$ 의 정방형의 블럭으로 나누어 이전의 프레임 중에서 기준 블럭에 대응하는 블럭을 중심으로 주변 영역을 탐색하여 오차가 가장 작은 블럭으로의 벡터를 구한다. 현재의 프레임의 블럭을 기준블럭이라고 하며 기준 블럭과 비교할 이전 프레임의 블럭을 탐색블럭이라고 한다. 동영상의 연속되는 프레임에서 물체의 위치

```

do v=0 to  $N_v - 1$ 
  do h=0 to  $N_h - 1$ 
    MV(h,v) = (0,0)
    (h,v) =  $\infty$ 
    do m = -p to p
      do n = -p to p
        MAD(m,n) = 0
        do j = 0 to N-1
          do i=0 to N-1
            enddo i
            enddo j
            if  $D_{min}(h,v) > MAD(m,n)$ 
               $D_{min}(h,v) = MAD(m,n)$ 
              MV(h,v) = (m,n)
            end if
          enddo n
        enddo m
      enddo h
    enddo v
  
```

그림 1. 블럭정합알고리즘
Fig. 1 The block matching algorithm

가 조금씩만 변하므로 블록의 탐색거리를 p로 한정함으로써 전체 계산량을 줄일 수 있다.

기준블럭과 이동벡터 (m, n)에 해당하는 후보탐색블럭과의 오차의 누적값 MAD(Mean Absolute Difference)은 식(1)과 같다.

$$MAD(m, n) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |R(i, j) - S(m+i, n+j)| \quad (1)$$

(2p+1)²개의 후보 탐색블럭 중 MAD의 값이 최소가 되는 블럭으로 이동을 이동벡터라 한다. 블럭정합 알고리즘은 그림 1과 같다.

그림 1의 x(i, j)는 기준블럭의 해당 픽셀값을 표시하고 x(i+m, j+n)은 이동벡터가 (m, n)인 탐색블럭의 픽셀값을 표시한다. N_v와 N_h는 각각 프레임의 가로, 세로의 기본블럭의 갯수를 나타낸다.

III. 이차원 시스토키어레이의 설계

1. 알고리즘의 병렬화와 데이터의존그래프

알고리즘의 각 문장에서 데이터의존성이 존재한다면 그 문장을 병렬로 처리할 수 없다. 따라서 블럭정합알고리즘의 순차적인 알고리즘의 각 루프의 데이터의존성을 제거하기 위해 단일할당코드로 알고리즘을 변경한다. 데이터의존성이 존재하는 그림 1의 s1을 인덱스 확장을 통해 의존성을 제거한다. 기준블럭과 탐색블럭의 오차를 각 열별로 누적하여 그 결과를 합하여 MAD를 구한다. MAD를 계산하는 그림 1의 s1을 그림 2의 s1과 s2로 나누었다.

```

do v=0 to Nv-1
  do h=0 to Nh-1
    MV(h,v) = (0,0)
    Dmin(h,v) = ∞
    do m = -p to p
      do n = -p to p
        MAD(m,n) = 0
        do j = 0 to N-1
          do i=0 to N-1
            [Redacted]
          enddo i
        enddo n
      enddo m
    enddo h
  enddo v

```

```

enddo j
if Dmin(h,v) > MAD(m,n)
  Dmin(h,v) = MAD(m,n)
  MV(h,v) = (m,n)
end if
enddo n
enddo m
enddo h
enddo v

```

그림 2. 블럭정합알고리즘의 단일할당코드
Fig. 2 The single assign code of block matching algorithm

그림 2의 s1은 각 열을 통해 오차를 누적하는 기능을 갖고 s2는 그 차를 누적하여 MAD를 구한다. 그림 2의 단일할당코드에서부터 그림 3과 같은 3차원 데이터 의존 관계를 찾을 수 있다. 그림 3은 3×3의 블럭에 최대탐색거리가 2인 경우의 데이터 의존 그래프의 예이다. 3×3개의 기준블럭의 데이터는 [0 0 1] 방향으로 입력하고, 7×7개의 탐색블럭의 데이터는 다음 후보블럭에서 재사용할 수 있도록 [1 0 -1] 방향으로 입력한다. 흰색 원으로 표현한 처리요소 AD는 입력된 기준블럭과 탐색블럭의 데이터의 차를 누적하여 [0 1 1] 방향으로 진송하는 기능을 갖는다. 검은색 원으로 표현한 처리요소 A는 각 열의 처리요소 AD에서 누적한 값을 합하여 각 후보블럭에 대한 MAD를 계산한다. 처리요소 M은 각 후보블럭 중 최소의 MAD를 갖는 블럭을 기준블럭과 정합되는 탐색블럭으로 선택한다.

2. 어레이로의 시간 및 공간사상

그림 3의 데이터의존그래프에는 식(2)의 데이터의존벡터를 갖는다.

$$D = [\vec{e}_1 \ \vec{e}_2 \ \vec{e}_3 \ \vec{e}_4] \quad (2)$$

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

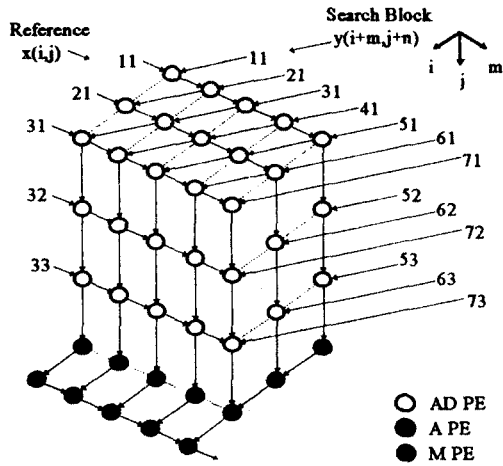


그림 3. FBMA의 데이터의존관계
Fig. 3 The data dependence of FBMA (where $N = 3, p = 2$)

그림 3의 데이터의존그래프를 [0 0 1] 방향으로 투영하여 입출력 핀의 수가 가장 작은 그림 4와 같은 시스토키어레이를 구한다. 이 경우의 최적의 스케줄 벡터는 $\vec{s}^T = [2 \ 1 \ 1]$ 이다.^[7] 각 에지의 점은 사각형은 단위시간지연을 나타낸다. 탐색영역은 왼쪽에서 오른쪽으로 전달되며 기준블럭은 처리요소 AD내에 저장된다. 가장 오른쪽 처리요소 A는 각 후보블럭의 MAD 값을 출력한다. 각 후보블럭의 MAD값을 비교하기 위한 처리요소 M은 생략했다. 그림 4의 왼쪽의 점선은 첫번째 탐색블럭의 인덱스를 표현한다. 하나의 기준블럭에 대해 $2p + 1$ 개의 후보블럭이 존재한다. 그림 4의 어레이는 탐색영역의 열을 기준으로 파이프라인화 시켰다. 탐색블럭의 열이 바뀌게 되면 p시간동안 의미없는 값을 출력한다.

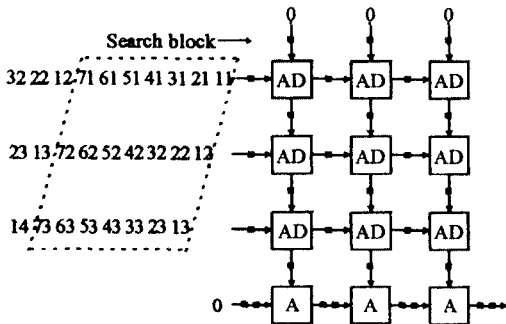


그림 4. 이차원 시스토키어레이
Fig. 4 The 2-D systolic array (where $N = 3, p = 2$)

3. 어레이의 확장

탐색블럭 픽셀값의 재사용을 높이도록 그림 4의 어레이를 연결하여 메모리 액세스를 줄이고 병렬성을 높일 수 있다. 그림 5는 탐색영역의 각 열의 입력에 대한 확장된 데이터의존그래프이다. 각 원은 그림 4의 어레이블럭을 나타낸다. 첫블럭에 입력된 N개의 열 중 $N-1$ 개는 열을 다음 블럭으로 전송하여 재사용한다. 이후의 각 어레이블럭은 하나의 열만을 새로 입력받는다.

각 탐색영역의 중복되는 열을 다음 어레이블럭으로 전달하여 전체화면의 기준블럭에 대한 파이프라인을 구성한다. 그림 5의 데이터의존그래프를 [0 1 0] 방향으로 투영시켜 메모리의 액세스가 최소가 되도록 어레이를 구성한다.

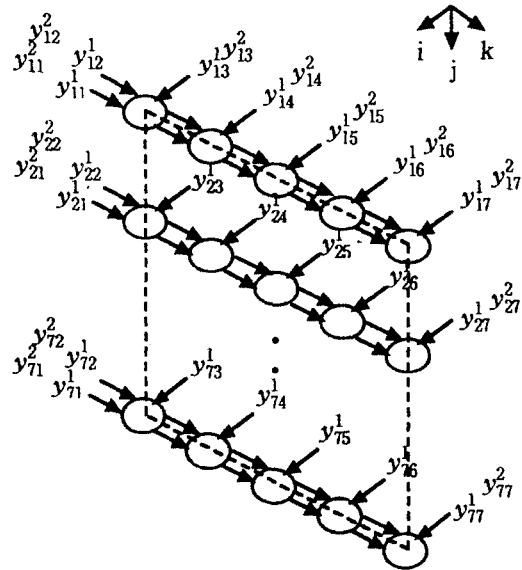


그림 5. 어레이블럭의 데이터의존그래프
Fig. 5 The data dependence graph of the array block (where $N = 3, p = 2$)

그림 5의 데이터의존그래프는 식(3)의 데이터의존 벡터가 존재한다.

$$D = [\vec{e}_1 \ \vec{e}_2] = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \quad (3)$$

이 경우 최적의 스케줄벡터 \vec{s} 는 [1 1 1]이다.

그림 4의 어레이를 그대로 연결할 경우 탐색블럭의 픽셀값을 입력하기 전 기준블럭의 픽셀값들을 각 처리요소 AD에 기억시켜야 한다. 각 어레이블럭마다 기준블럭을 픽셀값을 입력하는 편을 둔다면 전체 어레이의 입출력핀의 수가 많아진다. 각 블럭마다 기준블럭의 픽셀값을 입력하는 편을 두는 대신 탐색블럭의 픽셀과 같은 방향으로 다음 어레이블럭에 전달하면 입출력핀의 수를 줄일 수 있다. 탐색블럭의 픽셀

값은 기준블럭의 픽셀값이 안정화되는 $N-1$ 시간부터 입력된다. 그리고 다음 어레이 블럭으로 전달되는 기준블럭의 값과 타이밍을 맞추기 위해 어레이블럭들 사이에 탐색영역의 값을 전달하는데 한단위시간의 지연이 필요하다. 첫번째 기준블럭에 대한 이동벡터를 계산하는 시간은 식(4)와 같다.

$$T = [1 \ 1 \ 1] \begin{bmatrix} (2P+1) & -(N+1) \\ 2p+n & -1 \\ 2f(N+1)+N & -1 \end{bmatrix} + 1 + (N-1) \quad (4)$$

$$= 2N(p+1) + 6p - 2$$

그림 6은 $N=3, p=2$ 인 경우의 확장된 어레이의 구조를 나타냈다.

식(4)의 시간 이후 블럭파이프라인 주기인 $2p+N$ 시간마다 다음 기준블럭에 대한 이동벡터가 출력된다.

그림 7은 블럭인터페이스를 포함한 어레이블럭의 구조이다. 블럭인터페이스는 검은 사각형으로 표현한 탐색영역의 전달지연을 위한 레지스터와 현재 어레이블럭에서 계산된 MAD값과 위의 어레이블럭에서 계산하여 전달된 MAD값 중 작은 것을 선택하여 다음 어레이블럭으로 전달하는 기능을 포함한다.

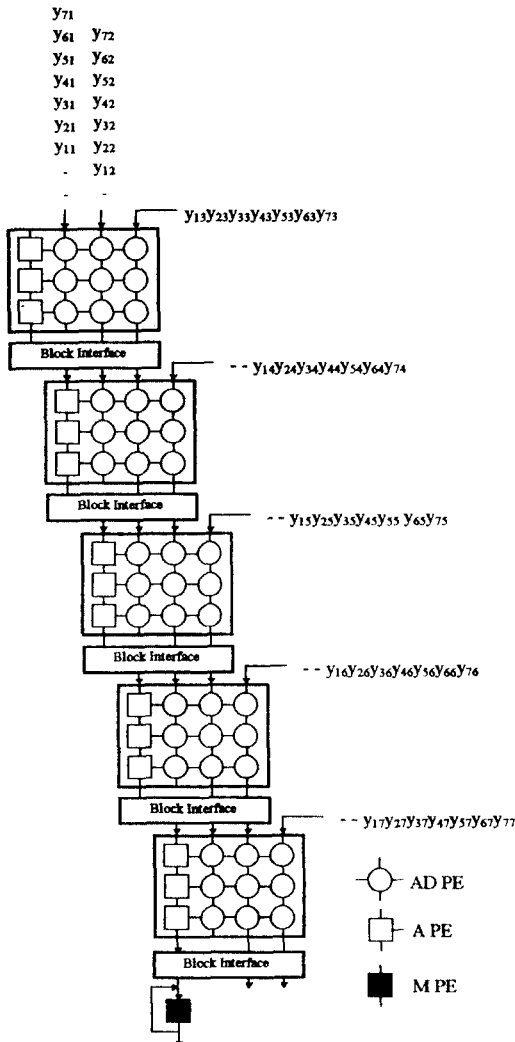


그림 6. 확장된 이차원 시스템 어레이
Fig. 6 The expanded 2-D systolic array (where $N=3, p=2$)

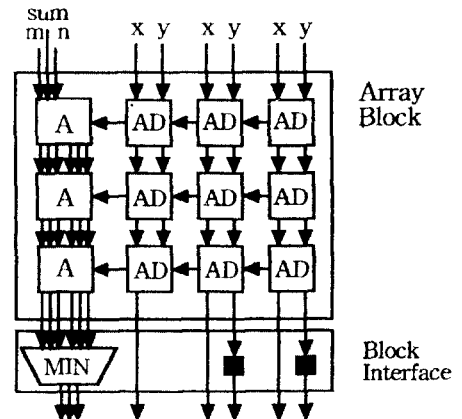


그림 7. 어레이블럭의 구조
Fig. 7 The structure of array block (where $N=3, p=2$)

IV. 결과 및 고찰

V. 결론

기준블럭의 크기가 $N \times N$ 이고 최대탐색거리가 p 인 경우 첫 번째 기준블럭의 계산시간은 식(5)이며 $2p + N$ 시간 이후 다음 기준블럭에 대한 계산결과가 출력된다. 그러므로 한 프레임에 대한 전체계산시간은 식(6)으로 구해진다.

$$2N(p + 1) + 6p - 2 \tag{5}$$

$$2N(p + 1) + 6p - 2 + (2p + N)(N_v N_h - 1) \tag{6}$$

표 1은 제안된 어레이와 기존의 어레이와의 성능을 비교하였다. 576×720 픽셀에 초당 30프레임울 갖는 비디오전송형태(CCIR Rec. 601)를 16×16 의 기준블럭과 최대탐색거리를 8로 두고 비교하였다.^{[7][8]}

표 1. 성능비교표(CCIR Rec 601.: 576×720 pixels, $N = 16$, $p = 8$)

Type	the number of PE	the number of PIN	the number of clock required	
			per first block	per frame
Komarek (AB2)	256	136	496	803,520
Komarek (AS2)	528	392	272	440,640
Hsieh & Lin	256	16	961	1,556,820
Jehng	512	2056	266	414,730
Yang	16	32	4,069	6,635,320
Chan	256	136	768	1,244,160
Yeo & Hu	256	32	256	423,936
Proposed method	4,624	392	334	52,142

제안한 어레이는 내부 처리요소의 수는 많으나 외부와의 입출력핀수의 증가를 억제하였다. 하나의 프레임에 대하여 시간 및 공간의 비교척도인 AT^2 을 비교하면 Yeo의 방법에 비해 약 3.7배의 성능향상을 가져왔다. 또한 하나의 프레임에 대한 처리요소의 이용률이 99%이상이다.

본 논문에서는 블럭정합알고리즘의 고속화를 위한 시스토키어레이를 설계하였다. 블럭정합알고리즘에서 단일할당코드로 변환하여 데이터 의존관계를 구하고 시간 및 공간변환을 거쳐 시스토키어레이를 구성하였다.

각 기준블럭에 대해 중첩되는 탐색블럭의 효율적인 재사용을 위해 어레이블럭을 중첩함으로써 메모리 액세스를 줄이며 파이프라인을 통해 프레임의 각 기준블럭에 대한 이동벡터의 계산을 고속화시켰다.

기준블럭이 $N \times N$ 이고 최대탐색거리가 p 로 주어질 경우, 이동벡터의 계산시간은 $O(N^6)$ 에서 $O(N^3)$ 으로 향상되었다. 전체처리요소의 수는 $N(N + 1)(2p + 1) + 1$ 개가 요구되며, 입출력핀의 수는 $(2(N + p) + 1) \times$ 비트수이다. 첫번째 기준블럭의 이동벡터가 $2N(p + 1) + 6p - 2$ 시간에 계산이 된 후 블럭파이프라인 주기 $2p + N$ 단위시간마다 다음 기준블럭의 이동벡터가 계산된다. 요구되는 처리요소의 수는 프레임의 픽셀수의 증가에 무관하며 단일방향의 데이터전송패스만이 존재하므로 VLSI 구현시 결합허용에 유리하다.

설계의 정확성과 유효함을 검증하기 위해 C 언어로 컴퓨터 시뮬레이션을 수행하였다. 각 단위시간에 변화하는 처리요소의 내부상태와 입출력되는 값들이 정확하게 변화하는 것을 확인하였다.

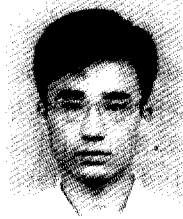
본 연구의 결과는 동영상압축기법인 블럭정합알고리즘의 고속처리를 위한 VLSI 구현에 대한 기초적인 설계기술로 제공될 것이다.

참고 문헌

1. K. M. Yang, M. T. Sun, and L. Wu, "A family of VLSI designs for the motion compensation block-matching algorithm," *IEEE Trans. Circuits Syst.*, vol. 36, pp. 1317-1325, Oct. 1989.
2. T. Komarek and P. Pirsch, "Array architectures for block matching algorithms," *IEEE Trans. Circuits Syst.*, vol. 36, pp. 1301-1308, Oct. 1989.
3. C. H. Hsieh and T. P. Lin, "VLSI architecture for block-matching motion estimation algorithm," *IEEE Tran. Circuits Syst. Video Technol.*, vol. 2,

pp. 169-175, June 1992.

4. E. Chan and S. Panchanathan, "Motion estimation architecture for video compression," IEEE Trans. Consumer Electron., vol. 39, No. 3, pp. 292-297, Aug. 1993.
5. Y. S. Jehng, L. G. Chen, and T. D. Chiueh, "An efficient and simple VLSI tree architecture for motion estimation algorithms," IEEE Trans. Signal Process., vol. 41, pp. 889-899, Feb 1993.
6. Hangu Yeo and Yu Hen Hu, "A novel modular systolic array architecture for full-search block matching motion estimation," IEEE Trans. Circuits. Syst., vol. 5, No. 3, Oct, 1995.
7. S. Y. Kung, VLSI array processors. Prentice Hall, Englewood Cliffs, N.J., 1988.
8. CCIR Recommend. 601, "Encoding parameters of digital television for studios," Recommend. pep, CCIR, Geneva, Switzerland, vol. XI, pt. 11TUS, 1982.



이 수 진(Su Jin Lee) 정회원
1995년 2월:부경대학교 전자공학과 졸업
1997년 2월:부경대학교 대학원 전자공학과 석사
1998년 현재:부경대학교 대학원 전자공학과 박사과정 재학중

※주관심분야:병렬처리, 영상압축, 컴퓨터구조



성 길 영(Kil Young Sung) 정회원
1980년 2월:경북대학교 전자공학과 졸업
1985년 2월:건국대학교 대학원 전자공학과 석사
1998년 현재:부경대학교 대학원 전자공학과 박사과정 재학중

1995년~현재:경상대학교 정보통신공학과 교수, 정보통신연구센터 연구원

※주관심분야:컴퓨터구조, 병렬처리, 신호처리