SIMD와 MIMD가 결합된 구조를 갖는 병렬처리시스템

이 형[†] · 최 성 혁[††] · 김 중 배[†††] · 박 종 원[††††]

요 약

영상에 관련된 다양한 응용 시스템들을 구현하는 많은 연구들이 진행되어 왔지만, 그러한 영상 관련 응용 시스템을 구현함에 있어서 처리 속도의 저하로 인하여 많은 어려움을 겪고 있다. 이를 해결하기 위해 대두된 여러 방법들 중에서 최근 하드웨어 접근 방법에 고려한 많은 관심과 연구가 진행되고 있다. 본 논문은 영상을 실시간으로 처리하기 위하여 하드웨어 구조를 갖는 병렬처리시스템을 기술하며, 또한 병렬처리시스템을 얼굴 검색 시스템에 적용한 후 처리 속도 및 실험 결과를 기술한다. 병렬처리시스템은 SIMD와 MIMD가 결합된 구조를 갖고 있기 때문에 다양한 영상 응용시스템에 대해서 융통성과 효율성을 제공하며, 144개의 처리기와 12개의 다중접근기억장치, 외부 메모리 모듈을 위한 인터페이스와 외부 프로세서 장치(i960Kx)와의 통신을 위한 인터페이스로 구성되어 있다. 다중접근기억장치는 메모리 모듈선택회로, 데이터 라이팅회로, 그리고, 주소계산 및 라우팅회로로 구성되어 있다. 또한, 얼굴 검색 시스템을 병렬처리 시스템에 적합한 병렬화를 제공하기 위해 메쉬방법을 이용하여 전처리, 정규화, 4개 특징값 추출, 그리고 분류화로 구성하였다. 병렬처리시스템은 하드웨어 모의실험 패키지인 CADENCE사의 Verilog-XL로 모의실험을 수행하여 기능과 성능을 검증하였다.

Parallel Processing System with combined Architecture of SIMD with MIMD

Hyung Lee[†] · Sung-Hyuk Choi[††] · Jung-Bae Kim[†††] · Jong-Won Park[††††]

ABSTRACT

There have been studied on building various application systems related to image. However, low image processing speed in many systems previously developed has been the problem. We have investigated a parallel processing system for improving the processing speed and applied to many related applications. In this paper, we address parallel processing system designed as hardware for meeting the demand on image applications to be processed in real time. And, facial recognition system was selected and attempted using the proposed system in order to verify its performance. The parallel processing system is similar to a combined architecture of SIMD with MIMD. Since this combined architecture has flexibility and efficiency, it should be applied to many related image applications. The parallel processing system proposed consisted of 144 processing elements, and 12 multi-access memory systems, and two interface modules ; one is for an external processing unit, Intel960Kx processor, the other for external memory modules. The multi-access memory system we introduced is made up of a memory module selection module, a data routing module, and an address calculation and routing module. In order to verify the system, we developed a parallel algorithm for the facial image recognition using the mesh feature. The algorithm consists of a preprocessing to extract feature regions, normalizing the final feature region, extracting 4 feature vectors from the previous feature regions and the normalized region, and a classification. This algorithm was performed on the system. The system and all processing procedures of the algorithm on it were simulated and evaluated by the CADENCE Verilog-XL hardware simulation package.

키워드 : SMID, MIMD, Parallel Processing System, Facial Recognition

1. Introduction

Multimedia processing is becoming increasingly important because of the wide variety of application. Each media in a multimedia environment requires different processes, techniques, algorithms and hardware. Hence, it is crucial to design processor architectures, that meet the computing requirements of the various media type, and the convenient data structures for a given class of applications implemented at the hardware [1, 2].

In recent years many researchers have been interested in various image applications. One of them is the facial image recognition. Up to date the investigations were focused on the automatic recognition and identification of the human faces, including the facial expression analysis [3, 4]. Although tremendous amount of work was done in building systems

---

※ This work was partly supported by IDEC.

[†] 정 회 원 : 충남대학교 대학원 컴퓨터공학과 박사과정 수료
[††] 정 회 원 : 아진전자산업
[†††] 정 회 원 : 옥성전자 이사
[††††] 정 회 원 : 충남대학교 정보통신공학과 교수

논문접수 : 2000년 6월 26일, 심사완료 : 2001년 1월 5일

for facial image recognition, there have been many difficulties to process the developed system in real time. One of them is in low image processing speed. In order to speed up the recognition of a facial image an attached special purpose SIMD (Single Instruction Multiple Data stream) processor which can communicate with a host computer via a fast PCI (Peripheral Component Interconnect) local bus system is required.

We have been studied and developed a parallel processing system to improve image processing speed. Generally it is similar to a combined architecture of an SIMD with an MIMD (Multiple Instruction Multiple Data stream). The parallel processing system consists of a processing unit ; a local memory which stores instructions and common data ; N processing elements which are instructed to be operated synchronously by the processing unit ; and a multi-access memory system which provides data to the N processing elements simultaneously.

Several authors considered multi-access memory systems [5-9]. In particular, Parks and Harper [9] proposed a memory system for the SIMD construction of a Gaussian pyramid, where the number of processing elements of the SIMD processor and the number of memory modules of the memory system are 2^n and 2^n+1, respectively. The memory system provides a simultaneous access to 2^n data elements whose access types are block, row, or column, where the interval of the block and the column is 1 and the interval of the row is of power of two.

In this paper, we propose a parallel processing system involving a multi-access memory system for a simultaneous access to the data elements within a row with a constant interval. It then suggests a parallel algorithm for facial image recognition to be applied to the system. The algorithm consists of a preprocessing and extracting feature region, normalizing feature regions, extracting 4 feature vectors from the previous feature regions and the normalized feature region with binary image, and a classification.

This paper is organized as follows. Section 2 addresses a multi-access memory system. A parallel processing system and a parallel algorithm for facial image recognition are detailed in Section 3 and Section 4, respectively. Section 5 presents experimental results and a simulation of the parallel processing system performed by using a hardware simulation package CADENCE Verilog-XL is demonstrated. Finally, we conclude this paper in Section 6 followed by the references.

2. Multi-Access Memory System

The memory system consists of a memory module selection circuitry, a data routing circuitry for write, an address and a routing circuitry, $(pq+1)$memory modules, and a data routing circuitry for read. In order to distribute the data elements of the $M \times N$ array $I(*,*)$ among m memory modules, a memory module assignment function must place in distinct memory modules the array elements that are to be accessed simultaneously. Also, an address assignment function must allocate different addresses to array elements assigned to the same memory module.

2.1 Memory Module Assignment Function

A memory module assignment function, which determines the index of memory module of an element, is $\mu(i,j)=(iq+j)//m$. Here the notation $x//y$ is used to denote the non-negative remainder that results from the integer division of x by y.

2.2 Address Assignment Function

The address assignment function which determines the address of an element within a memory module is $a(i,j)=(i.j)\times S+j/q$, where S is any integer satisfying $s \geq \lceil X \rceil \times N/2q$ and $S \times M/2q \leq c$. c is the capacity of each memory module.

2.3 Address Calculating Circuit

This section discusses the address calculating circuit that computes pq addresses of a row with a constant interval r. The differences of pq addresses of pq the data elements within a row with a constant interval r from the base address, $a(i,j)$, are

$$\begin{aligned} AROW(i,j,r,a) &= a(i,j+ar)-a(i,j), \\ &= (j//q+ar)/q, \qquad 0 \leq a < pq. \end{aligned} \tag{1}$$

The address calculating circuit computes m addresses by using the m adders provided that the base address $a(i,j)$ and the address differences are supplied to input A and B of the m adders, respectively.

2.4 Address and Data Routing

The address routing circuit receives m addresses from the register A1 in the address calculating circuit and moves the m addresses to the m memory modules through the register A2. The index numbers of the memory modules for the m addresses are represented as follows for the different access patterns, where the interval $r > 0$ and $r // m \neq 0$:

$$INROW(i,j,r) = (\mu(i,j) + br) // m, \ 0 \leq b \leq pq-1. \qquad (2)$$

The address difference of the row of the μth module $ADROW(\mu)$ is obtained from (1) and (2) :

$$ADROW(\mu) = (i//q + ((\mu - iq - i) \times r' // m) \times r)/q, \quad 0 \leq \mu < pq. \qquad (3)$$

By substituting $(\mu + \mu(0))//m (= ((\mu - iq - j) \times r' // m + (iq \ + j)//m)//m$ instead of μ into (3), we get the address differences in ascending order of memory modules from $\mu(0)$ as follows :

$$ADROW(\mu + \mu(0)//m) = (i//q + (((\mu - iq - j) \times r' // m + (iq + j)//m)//m) \times r)/q, \quad 0 \leq \mu < pq, \text{ where } r \times r' = 1//m. \qquad (4)$$

For routing the calculated addresses to the memory modules, it is required for the address differences stored in the address difference memory module to be rotated by the index number of the memory module in the first data element. Prearranging and storing the address differences in the address difference memory module make the address routing circuitry inexpensive and simple to control. The role of the date routing circuit is to align data elements read from or to the memory modules. After aligning the data elements as follows, right-rotation is performed by $\mu(i,j)$ times in order for the index numbers of the memory modules of those data elements to be in the ascending order : $D2((kr)//, m \leftarrow D1(k) \ 0 \leq k \leq pq-1$, where $D1$ and $D2$ registers are the date register and a temporary register, respectively. For the READ operations, the routing patterns are reversed.

2.5 Memory Module Selection

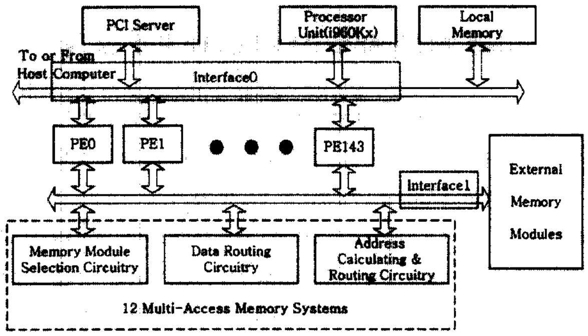
The memory module selection circuit enables pq memory modules whose index numbers are represented in (4).

Three ROM are needed in implementation of the multi-access memory system. One ROM is for storing results of the memory module selection functions because it takes a higher space complexity on the device. The others are for synchronizing three modules. In order to obtain valid addresses to each memory modules, the time complexity of address calculation and routing module is higher than other two modules. One of two is for basic addresses to each memory module and the other for $a(i,j)$ in the equation (1).

3. Parallel Processing System

If an algorithm consists of a lot of identical operation on different image points, an SIMD architecture in which N processing elements are instructed to be operated synchronously under the control of one processing unit is adequate for the algorithm. If an algorithm needs to access data in any direction with an interval corresponding to the length of between data, a multi-access memory system, which provides simultaneous access to N data elements with access types, is adequate for the algorithm. The type may be a block, a row or a column with some intervals. These properties are involved in some image processing algorithm or some parts of an algorithm.

Sometimes, SIMD architecture can have a side effect to work some image processing algorithms because a lot of the algorithm or parts of it do not enough to satisfy parallel properties. Thus, these take the lower degree of parallelism. In order to solve it, the parallel processing system has to be exchanged MIMD architecture and SIMD architecture during the processing. In this case, a processor unit asserts the mode register in each processing element. In MIMD mode, each processing element works as the same as a stand-alone processor and directly access to memory modules not going through a multi-access memory system.



(Figure 1) The block diagram of parallel processing system

A parallel processing system with consists of a processing unit that is Intel960Kx processor, a local memory, external memory modules, 144 processing elements, 2 interface modules, and 12 multi-access memory systems is represented in (Figure 1). The processing unit fetches and decodes an instruction in a local memory in which lists of instructions for an algorithm are stored. And not only synchronously issues data or an instruction to processing elements but also controls processing elements. A processing element can execute basic instructions as same as ones of a basic computer, which are introduced in [10]. Registers in a processing element have 8 bits and each processing element has 16 registers ; 4 special registers, 9 general registers for MIMD mode, and 3 general registers for SIMD mode. Special registers are used to enable/disable a processing element to be operated and to indicate the state of a processing element to a processing unit. The state register and the wait register are needed a processor unit to synchronize processing elements in MIMD mode because each other instruction to be doing in processing elements has its own processing times. Two interfaces are needed to control data routing according to mode specified. Interface0 module is of arbitration modules to control signals between processing unit and processing elements. This module consists of 12 general registers, 3 special register, and submodules for arbitration. Interface1 module is of switching modules, which is comprised of MUX-based combinational circuits to support memory access operations according to a specific mode.

The parallel processing system we designed provides logically two dimensional addressing mode which is used in (r,c)-based image domain. Therefore, as previously mentioned, most of image processing in spatial domain is done with enough processing power in the parallel processing system.

4. Parallel Algorithm for Facial Image Recognition

To verify the performance of the proposed parallel processing system, specific applications to the system are investigated. One of them is to recognize facial images, which was the application attempted using the proposed the system. The algorithm for facial image recognition that we proposed consists of a preprocessing and extracting feature region, normalizing feature regions, extracting 4 feature vectors from the previous feature regions and the normalized feature region with binary image, and a classification. The details are as follows.

4.1 Extraction of Feature Region

Facial images to be processed are allowed to have some limitations. Those remained in gray level without a background, those looking forward without glasses, and the part of facial components was not to be covered by the hair. In the preprocessing step hair on the face scanned was eliminated using histogram method. This method helps to distinguish between hair color and face color. The next step is to extract the region involved eyes, nose, and mouth whose we interested. In order to extract the region of interest, a small region containing eyes was first detected using the histogram of the gray intensity of an image after Sobel edge operation. Using Sobel edge operator a contour and outlines of facial components are detected and location of eyes are found by both vertical and horizontal histogram.

Based on the length between outsides of eyes, the region containing eyes, nose, and mouth was extracted. In the knowledge base, the height of the region that includes nose and mouth is not greater than one and half of the length in the general. The final feature region we concerned is established by the methods that the former region was equalized and binarized using a combined technique of P_title global binarization with local window based method. Thus, The final region is of a binary image. From the region, the first feature $f_1=\frac{h}{w}$, where is the width and h is the height of the feature region, was yielded.

4.2 Normalization

We normalized the final feature region to compare the same facial images with different sizes. Size normalization is a transformation of image of arbitrary size ($x1 \times x2$) into an image of fixed pre-specified size (for example, 48×48) which results in dimensional changes by factors of $48/x1$ and $48/x2$. This process is a crucial preprocess to obscure scale variation of facial images presented to a recognizer.

4.3 Extraction of Feature Vectors

The normalized region yielded 3 feature matrices as features and itself became the fourth feature, $0 \le i,\ j \le 48$. Each matrix element, which is the second feature, was calculated for each 3×3 mesh of the normalized region, $f_2=I_2(i,j)=\sum_{k=i}^{3}\sum_{l=j}^{3} I_4(k,l),\ 0\le i,j\le 16$. This is processed by horizontal access type with interval 3 and 12 in the multi-access memory system. The third feature was calculated using 6×6 mesh, $f_3=I_3(i,j)=\sum_{k=i}^{2}\sum_{l=j}^{2} I_2(k,l)$, $0\le i,j\le 8$, by horizontal access type with interval 2 and 6.

4.4 Classification

Four feature vectors are sequentially compared to those of facial image stored in database using equation of which form is $d_i=|f_i-f'_i|$, $1\le i\le 4$, where f'_i is the th feature of some facial image in database. The order of comparison is f_1, f_3, f_2, and f_4 with each limitation value l_i. If d_i is came within l_i percents of total compared images, it becomes the candidate facial image in next comparison because it is the most similar facial image. All comparisons need a horizontal access type with interval 1.

5. Experiments and Simulation

To implement the parallel processing system involving the multi-access memory system as previously presented for facial image recognition, some parameters should be defined by values based on the algorithm of facial image recognition. These values for parameters are in an application specific and system design specific. These parameters and values are in <Table 1>.

After deciding system parameters, the considerable design issue is the data and addresses routing path. This routing path is affected by the first 5 parameters in <Table 1>. The block diagram for the routing circuitry is depicted in (Figure 2).

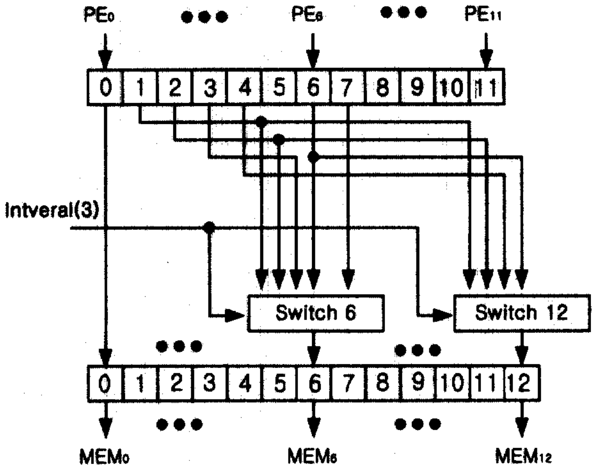
<Table 1> Parameters and Values for the parallel processing system

Parameter	Value	Description
M	144	Size of the row
N	54	Size of the column
Interval	1, 2, 3, 6, 12	Intervals
P	4	Basic parameter
Q	3	Basic parameter
S	18	Basic parameter for address calculation
data bus	6	Bandwidth of data bus
No. of ports	72	No. of PEs x data bus
No. of memory modules	13	No. of memory modules
No. of PEs	12	No. of PE x No. of PEs = 144

<Table 2> Lists of Primitive instruction for facial image recognition

Controls	Clr, CtoR, RtoR, Add, Dec, Inc, Xor, Cond1, Cond2, Cond3
Processing	PeClr, PeRtoT, PeCtoR, PeInc, PeAdd, PeCmp, PeCond1. PeCond2
Others	Mask, Halt, CutoPe, Read, Write

Basic instructions for the algorithm are listed to design a processing element because the role of processing element is likely to a processor. To do so, the facial recognition algorithm was analyzed and decomposed to detail instructions. Then, lists of instructions are classified to design a processing element and shown as <Table 2>. With the general point of view, we thank about additional instructions to the lists because other applications will be carried out on the system. Additional instructions were established from a lot of methods for image processing being used for a long time. Therefore, some image applications satisfied to these instruction lists can be carried out on this system.



(Figure 2) In case of interval 3, block diagram of the routing circuitry

One hundred image scans were randomly acquired from an alumni album. 4 features extracted from each acquired image are stored in database. Database is not of commercial one and is just file systems stored features sequentially. For the test, 53 out of 100 images were randomly selected and their size reduced to four- ninth of the original image. The results were shown in <Table 3>.

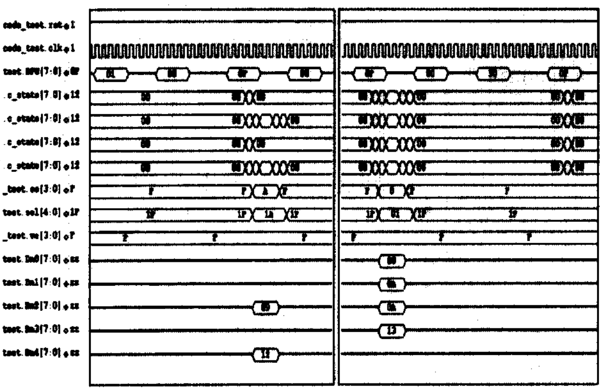
<Table 3> The results of facial recognition

	The first order	The second order	Failure	Total
No. of images	47	5	1	53
Rate (%)	88.6	9.6	1.8	100

In <Table 3>, one failure is the result that reducing image size is cause to blur the image. This blurred image is the facial image that eyes in it attached to eye bowls and eye bowls to hair. Therefore, the algorithm we designed was not detected the region of interest included eyes.

A simulation of the parallel processing system is performed by using CADENCE Verilog-XL and the wave form obtained by the simulation is illustrated in (Figure 4).

In (Figure 4), c_state signals are to control the state of processing elements, and present which instructions are currently running in processing elements. Other control signals are oe_, sel_, and, we_ to control external memory modules. One of them, sel_ signal, is generated by multi-access memory system, others by processing elements. Dm 0~4 are of data lines between Interface1 module and external memory modules. In (Figure 4-a), two processing elements are running arithmetic operations, which are addition and increment, and others WRITE operations with different addresses. Four processing elements are running addition after read data elements, READ operation, which are depicted in (Figure 4-b).



(a) In MIMD mode (b) In SIMD mode

(Figure 4) Wave forms obtained through the simulation : in cases of MIMD and SIMD

On the result of simulation, we compared serial processing with parallel one as following :

$$
\begin{aligned}
Tss &= Tsr + (Tss1 + Tss2 + Tss3 + Tss4) \times 53 \\
&= T + 85899750ns, \\
Tps &= T + Tps1 + Tps2 + Tps3 + Tps4 \times 5 + Tpci \times 2 \\
&= T + 1385720ns
\end{aligned}
$$

where *Tss* is the total processing time of the serial facial recognition, *Tps* is the total processing time of the parallel one, and *T* is the preprocessing time processed on the host. The results showed that the parallel processing was about 61.9 times faster than the serial processing, where the number of PE's was 144.

6. Conclusions and Discussion

The demands for processing multimedia data in real-time using unified and scalable architecture are ever increasing with the proliferation of multimedia applications. We presented a parallel processing system to achieve the demands for speedups and applied to the facial image recognition, which was recognized time-consuming works. It was confirmed that the system had been able to support enough scalability to the applications and improved their processing speed in real-time processing. If large number of PE is equipped with the system, it can be speculated that the speedup performance be pushed up to the limitation predicted by Amdahl's Law.

Although the comparison values previously mentioned were obtained through simulations and speedup performances during the application on the system were achieved, it was just estimated values because the proposed system has not yet been manufactured into a circuit board.

Unfortunately, some problems occurred in transferring a lot of data elements from the host to the system and vise versa during processing the application on the system. That is, the time for transferring data allocated more than the processing time. To solve this, the bus bandwidth needs to be improved on the system side and new specific methods to the system be developed on the method side.

References

[1] Sethuraman(Panch) Panchanathan, "Architectural Approaches for Multimedia Processing," 4th International ACPC'99, Salzburg, Austria, Feb. 1999.

[2] Edwige Passaloux, "On Parallel Reconfigurable Architecture for Image Processing," 4th International ACPC'99, Salzburg, Austria, Feb. 1999.

[3] Ashok Samal and Pransana A. Iyenger, "Automatic recognition and analysis of human faces and facial expression : A survey," Pattern Recognition, Vol. 25, No.1, pp.65-77, 1992.

[4] Jeffry R. Bach, Santanu Paul, and Ramesh Jaim, "A visual information management system for the interactive retrieval of faces," IEEE Transactions on Knowledge and Data Engineering, Vol.5, No.5, pp.619-628, Aug. 1993.

[5] P. Budnik and D. J. Kuck, "The Organization and use of parallel memories," IEEE Transactions on Computer, Vol.C-20, pp.1566-1569, Dec. 1971.

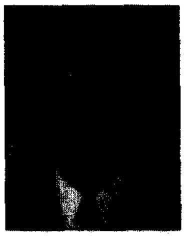
[6] D. C. Van Voorhis and T. H. Morrin, "Memory systems for image processing," IEEE Transactions on Computer, Vol.C-27, pp.113-125, Feb. 1978.

[7] D. H. Lawrie and C. R. Vora, "The prime memory system for array access," IEEE Transactions on Computer, Vol. C-31, pp.435-442, May 1982.

[8] J. W. Park, "An efficient memory system for image processing," IEEE Transactions on Computer, Vol.C-35, pp.669-674, July 1986.

[9] J. W. Park and D. T. Harper III, "An efficient memory system for the construction of a Gaussian Pyramid," IEEE Transactions on Parallel and Distributed Systems, Vol.7, No.7, pp.855-860, July 1996.

[10] M. MORRIS MANO, 'Computer System Architecture', 2nd Edition, Prentice hall, 1982.

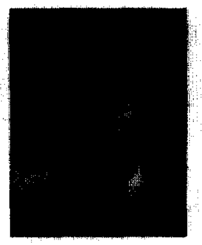


이형

e-mail : hyung@crow.cnu.ac.kr
1995년 충남대학교 컴퓨터과학과 졸업(학사)
1997년 충남대학교 컴퓨터공학과 졸업(공학석사)
2000년 충남대학교 컴퓨터공학과 박사과정 수료
관심분야 : 영상처리, 병렬처리, 반도체설계

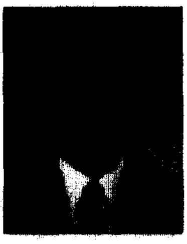
최성혁

e-mail : csh@ajinelec.com
1991년 한국항공대학교 정보통신공학과 졸업(학사)
1991년~2000년 전자통신연구원 연구원
1999년 충남대학교 전자공학과 졸업(공학석사)
2001년~현재 아진전자산업 연구원
관심분야 : 병렬처리, 데이터 통신



김종배

e-mail : jbkim@wooksung.com
1989년 한남대학교 전자계산기공학과 졸업(학사)
2000년 충남대학교 정보통신공학과 졸업(공학석사)
1989년~2000년 한국전자통신연구원 선임연구원
2000년~현재 욱성전자 이사
관심분야 : 병렬처리, 반도체설계



박종원

e-mail : jwpark@crow.cnu.ac.kr
1979년 충남대학교 전자공학과 졸업(학사)
1981년 한국과학기술원 전산학과 졸업(공학석사)
1991년 한국과학기술원 전산학과 졸업(공학박사)
1995년~현재 충남대학교 공과대학 정보통신공학과 교수
관심분야 : 영상처리, 병렬처리, 의공학