# 실시간 디지털 홀로그래피를 위한 고성능 CGH 프로세서

서영호* · 최현준** · 김동욱**

## FImplementation of RF Controller based on Digital System for TRS Repeater

Young-Ho Seo* · Hyun-Jun Choi** · Dong-Wook Kim**

이 논문은 2005년 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임.
(KRF-2005-041-D00594)

요 약

본 논문에서는 하드웨어의 구현을 위해 수정된 CGH(Computer Generated Hologram) 알고리즘을 바탕으로 디지털 홀로그램을 생성할 수 있는 하드웨어 구조를 제안하고 FPGA(Field Programmable Gate Array)를 기반으로 구현하고자 한다. 먼저 CGH 알고리즘을 분석한 후에 디지털 홀로그램을 효율적으로 연산할 수 있는 CGH 셀 (cell)의 구조를 제안하고 CGH 셀의 확장을 통해서 CGH 커널 (kernel)을 구현한다. 그리고 최종적으로 CGH 커널과 SDRAM Controller, DMA 등의 블록들을 결합하여 CGH 프로세서를 구현한다. 제안한 구조는 CGH 커널 내 CGH 셀의 단순한 추가를 통해서 성능을 비례적으로 증가시킬 수 있다. 이는 CGH 셀들이 독립적으로 동작하기 때문이다. 제안한 하드웨어는 Xilinx의 XC2VP70 FPGA를 이용하여 구현하였고 200 MHz의 동작속도에서 40,000개의 광원으로 구성된 3차원 객체를 0.205초에 1,280×1,024크기의 홀로그램으로 생성할 수 있다.

ABSTRACT

In this paper, we propose a hardware architecture to generate digital hologram using the modified CGH (Computer Generated Hologram) algorithm for hardware implementation and design to FPGA (Field Programmable Gate Array) platform. After analyzing the CGH algorithm, we propose an architecture of CGH cell which efficiently products digital hologram, and design CGH Kernel from configuring CGH Cell. Finally we implement CGH Processor using CGH Kernel, SDRAM Controller, DMA, etc. Performance of the proposed hardware can be proportionally increased through simply addition of CGH Cell in CGH Kernel, since a CGH Cell has operational independency. The proposed hardware was implemented using XC2VP70 FPGA of Xilinx and was stably operated in 200MHz clock frequency. It take 0.205 second for generating 1,280×1,024 digital hologram from 3 dimensional object which has 40,000 light sources.

키워드

digital hologram, 3D, CGH, VLSI, FPGA

* 한성대학교 정보통신공학과, 미디어연구실
접수일자 : 2007. 5. 29
** 광운대학교 전자재료공학과, 디지털 설계 및 테스트 연구실

## Ⅰ. 서 론

홀로그래피는 1948년 Gabor에 의해 최초로 제안된 이래 3차원 정보를 기록할 수 있다는 특징 때문에 많은 연구자들의 관심을 끌어왔다. 기존의 홀로그래피는 홀로그램 필름에 3차원 정보를 기록하고, 현상된 필름을 사용하여 3차원 물체를 복원하는 방식을 사용함으로써 그 응용이 크게 제한되었다. 이러한 단점을 극복하기 위한 새로운 접근방법으로 1966년 이후 많은 연구자들이 컴퓨터에 의한 홀로그램의 제작을 연구해 오고 있다[1].

컴퓨터에 의한 홀로그램의 제작은 1960년대 중반이후부터 시작된 것으로, 물체파(object wave)와 기준파(reference wave)의 간섭에 의해 생성되는 간섭항을 계산함으로써 현실에서는 불가능한 이상적인 특성을 가진 부품을 제작하거나 특성시험 등을 위해 개발되었다[2].

컴퓨터 생성 홀로그램(CGH)으로 제일 먼저 개발된 것은 Brown과 Lohmann에 의한 "Detour Phase"방식이다. 이것은 홀로그램을 동일하게 구획된 셀로 분할하고, 각 셀 내에 사각형의 개구를 만들어, 이 개구의 크기와 위치를 홀로그램의 위상과 진폭에 대응하여 조정하여 제작하는 것으로, 기준파가 없이 물체파의 푸리에 변환에 의한 회절광의 성분만을 이용하므로, 원리상 평면의 패턴과 같은 단순 2차원적 물체의 홀로그램 제작만 가능하여 광 신호처리를 위한 공간 필터 제작 등에 응용된다. 그 다음이 Lee에 의해 개발된 1) 간섭항의 복소 진폭 및 위상을 "Detour Phase" 방식에서와 같은 각 셀을 4개의 등간격 칸으로 분리하고, 각 칸에 복소 진폭 및 위상의 크기와 부호를 고려하여 개구를 만드는 방식과 2) 간섭항의 진폭을 1로 두고, 1/2을 더하여 진폭 투과율 분포로 하여 주어진 문지방 값보다 크거나 같은 경우는 1 그 외는 0으로 하여 간섭무늬를 그리는 방식이 있고, 간섭항의 위상을 r(x,y,z)를 일정하다고 하고, 피사체의 형태에 관한 좌표를 이용하여 o(x,y,z)를 직접 계산하는 "Phase"방식이 있다. 이 방식은 홀로그램 상에 기록되는 실질적인 정보인 위상을 기하 광학적인 방법으로 계산하기 때문에 이전의 방식에 비해 많은 계산 시간을 필요로 하나, 피사체의 형태에 구해 없이 적용이 가능하며, 가장 실질적인 홀로그램에 근사하는 방식이므로, 홀로그래픽 비디오 홀로그램 제작에 응용되고 있다. 그 외 Kinoform은 위상에 의해 주어지는 간섭항의 진폭변화를 필름의 표면 요철로 하여 기록하는 방식이어서 디지털적 홀로그램의 제작이 가능하므로, 프린팅 방식을 이용한 각종 엠보스 홀로그램의 제작에 많이 이용된다. 이들 CGH 이외에도 단순히 격자벡터의 방향과 주기가 다른 격자의 조각으로 합성시킨 격자 홀로그램도 CGH의 하나이다[3].

CGH를 이용하여 한 프레임에 해당하는 하나의 객체를 위한 홀로그램이 생성되기 위해서는 많은 연산량과 시간이 소요되기 때문에 소프트웨어로 처리하는 것은 불가능하고 하드웨어로 구현되어야만 한다. 따라서 지금까지 많은 연구가 진행되어 왔다[4~7]. [4]에서는 4개의 Xilinx FPGA (XC2VP70)를 사용하는 전용 PCB 보드를 제작하여 Fresnel Transform CGH를 구현하였다. 홀로그램의 x축 해상도만큼의 단위 연산기를 병렬로 배열(1,408개)하는 구조를 가지고 166MHz의 클록 주파수에서 한 프레임의 홀로그램을 0.0679초에 생성할 수 있다. [5]에서는 1개의 Altera FPGA (EP20K300E)를 사용하는 전용 PCB 보드를 제작하여 Fresnel Transform CGH를 구현하였다. 27개의 단위 연산기를 병렬로 배열하는 구조를 가지고 35MHz의 클록 주파수에서 한 장의 홀로그램을 0.15초에 생성할 수 있다. [6]과 [7]에서는 nVIDIA의 GeForce 6600이라는 상용 그래픽 보드에 펌웨어 형태의 CGH 프로그램을 다운로드하여 Fresnel Transform CGH를 구현하였다. 이 두 연구의 결과는 전용 칩셋을 설계한 것이 아니므로 CGH를 생성하는 시간은 앞서 소개된 두 구현결과에 비해서 뒤쳐지는 성능을 보인다.

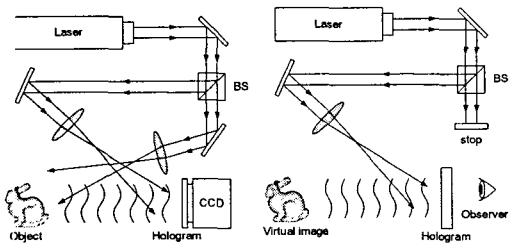
본 논문에서는 하드웨어 구현을 위해 수정된 CGH 알고리즘을 바탕으로 100% 파이프라인 형태의 하드웨어 구조를 제안한다. 제안된 하드웨어는 FPGA로 구현하였다. 본 논문에서는 기존의 CGH 알고리듬을 수정보완하기 보다는 이를 하드웨어로 구현할 때 보다 효율적이고 고속으로 동작할 수 있는 설계에 그 목적을 두고 있다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 CGH의 원리와 하드웨어 구현을 위해 수정된 CGH 알고리즘을 설명한다. 3장에서는 제안한 하드웨어 구조를 설명하고 4장에서는 구현결과를 보이고 기존 연구와 비교한다. 마지막으로 5장에서는 결론을 맺는다.

## Ⅱ. 컴퓨터 생성 홀로그램의 개요

디지털 홀로그램은 광학장비 대신 전자장비를 이용하는 방식으로, 홀로그래피의 간섭무늬(fringe pattern)를

CCD 카메라에 기록하고 비디오 신호로 전송하여 수신단에서 SLM(Spatial Light Modulator)에 표시된 간섭무늬 데이터에 레이저광을 조사함으로써 영상을 재생하는 기법이다. 그림 1에서 디지털 홀로그램의 시스템 구성을 보이고 있다. 디지털 홀로그램은 기존의 광학 홀로그램에 의한 기법과 동일하게 레이저광을 집광 렌즈로 평행광을 만들고, 빔 분리기로 참조파와 객체파로 나눈다. 객체파는 피사체(object)에 조명된 다음 참조파와 직접 CCD에 조사되어 간섭무늬을 형성한다. 간섭무늬 정보는 SLM에 인가되고 여기에 평행광을 조사하면 1차 회절광이 발생하여 3차원 객체를 재생할 수 있다[2].




(a) (b)

그림 1. 디지털 홀로그림 (a) 기록 (b) 복원

Fig. 1. Digital hologram (a) recoding (b) reconstruction

홀로그램은 그림 1과 같은 광학계를 이용하여도 취득할 수 있지만 이러한 광학계 자체를 수학적으로 모델링한 후에 연산에 의해서 취득할 수도 있다. 이러한 수학적인 연산을 통해 얻어진 홀로그램을 컴퓨터 생성 홀로그램이라고 한다. 여러 종류의 CGH가 있지만 본 논문에서는 "Phase" 방식의 계산 방법을 사용한다. 그림 1(a)에서 객체에서 CCD로 입사되는 파에서 위상 성분만을 이용하여 홀로그램을 생성하는 것이고 그 증명[3]은 본 논문에서 다루지 않는다.

이미 언급한 것과 같이 파면의 간격이 시간과 공간에 걸쳐 일정하게 유지되는 가 간섭성 광이 어떤 형태를 가진 물체에 부딪쳐 반사될 때는 그 반사된 광의 파면은 물체의 형태에 비례하여 달라진다. 즉 파면의 위상 변화가 물체의 형상에 따라 변화하게 된다. 그러므로 물체의 형상에 관한 좌표의 정보가 있으면, 형상에 따라 변화하는 파면의 위상변화는 기하광학적인 광선추적에 의해 쉽게 계산된다. 그 계산 과정의 순서는 다음과 같다

① 계산하고자 하는 물체의 표면을 3차원 좌표 내에 나열된 점들로 표시하고, 각 점에서부터 홀로그램 판상의 각 점까지의 거리를 계산한다.

② 호이겐스 원리에 따라, 물체 표면의 한 점에서 고유 위상을 가진 광파가 나온다고 가정할 수 있으므로, 홀로그램 상의 한 점에 입사하는 물체로부터의 광파는 물체의 전 표면에서 오게 되므로 먼저 계산한 물체의 각 점에서 이 점까지의 거리를 각 광파의 위상으로 변환하여 각 점으로부터의 광파를 합치게 되면, 홀로그램의 한 점에서 물체파가 계산된다.

③ 기준파의 경우는 평면 기준파가 홀로그램에 수직방향에 있어서는 수직으로 입사하고, 수평방향은 어떤 각을 가지고 입사한다고 가정하면, 기준파는 수평방향으로만 위치별로 위상차를 가지게 된다.

④ 홀로그램의 한 점에서 계산된 물체파와 이 기준파를 합쳐서, 빔강도를 구하고 시간 평균을 취하게 되면 이점에서의 상대적인 빔 강도가 구해지므로, 판의 다른 점에서의 빔 강도를 구하여 같은 빔 강도를 가지는 점들을 연결하게 되면 간섭무늬가 얻어진다. 이것이 컴퓨터 생성 홀로그램이다.

⑤ 물체의 각 점에서 나오는 광파의 진폭은 위치에 따라 달리 할 수도 있으나 결과에 큰 영향을 미치지 않는다.

### 2.1. CGH 알고리즘

본 절에서는 위상을 이용한 CGH 생성 방법에 대해서 설명한다. CGH는 식 (1)과 같이 정의되는데 앞서서 설명한 것과 같이 홀로그램의 위상($\theta_H$)으로부터 홀로그램의 빛의 강도($I_\alpha$)를 얻는 방법이다. 여기서 N은 3차원 객체의 광원수를 뜻한다.

$$I_\alpha = \sum_j^N A_j \cos(\theta_H + \Phi_\alpha + \Phi_j) \qquad (1)$$

$$R_{\alpha j} = \sqrt{(px_\alpha - px_j)^2 + (py_\alpha - py_j)^2 + z_j^2} \qquad (2)$$

$\theta_H$는 $kR_{\alpha j}$로 정의되고 이는 식 (2)와 같다. 여기서 $k$는 참조파의 파수(wave number)로 $2\pi/\lambda$로 정의되고 $\lambda$는 $532nm$의 값을 사용한다. $x_\alpha$와 $y_\alpha$는 홀로그램내의 위치를 뜻하고 $x_j$, $y_j$, 및 $z_j$는 3차원 객체의 위치를 나타낸다. $p$는 픽셀의 크기를 나타내고 본 논문에서는 $10.4\mu m$의 값을 사용한다. 식 (2)는 Fresnel Approximation

에 의해서 식 (3)으로 근사화 될 수 있다[8].

$$R_{\alpha j} \cong z_j + \frac{p^2}{2z_j}(x_{\alpha j}^2 + y_{\alpha j}^2) \tag{3}$$

$kR_{\alpha j}$에 해당하는 항은 식 (4)와 같이 정리할 수 있다. 여기서 $x_{\alpha j}$와 $y_{\alpha j}$는 $x_\alpha - x_j$와 $y_\alpha - y_j$이다.

$$\begin{aligned} kR_{\alpha j} &= k(z_j + \frac{p^2}{2z_j}(x_{\alpha j}^2 + y_{\alpha j}^2)) \\ &= 2\pi(\frac{z_j}{\lambda} + \frac{p^2}{2\lambda z_j}(x_{\alpha j}^2 + y_{\alpha j}^2)) \\ &= 2\pi(\theta_Z + \theta_{XY}) \\ &= \theta_H(x_{\alpha j}, y_{\alpha j}, z_{\alpha j}) \end{aligned} \tag{4}$$

식 (4)에 의해서 식 (1)은 식 (5)와 같이 나타낼 수 있다. 식 (5)와 같이 유도하는 이유는 코사인 함수는 $2\pi$를 주기로 하여 반복되므로 $(\theta_Z + \theta_{XY})$의 결과에서 정수부분을 고려하지 않아도 되기 때문이다. 소수부분의 연산만 수행함을 나타내는 연산자 mod[]를 도입한다면 식 (5)에서 $\theta_H$항은 식 (6)과 같이 유도할 수 있다.

$$\begin{aligned} I_\alpha &= \sum_j^N A_j \cos(k(z_j + \frac{p^2}{2z_j}(x_{\alpha j}^2 + y_{\alpha j}^2)) + \Phi_\alpha + \Phi_j) \\ &= \sum_j^N A_j \cos(\theta_H + \Phi_\alpha + \Phi_j) \\ &= \sum_j^N A_j \cos(2\pi(\theta_Z + \theta_{XY}) + \Phi_\alpha + \Phi_j) \end{aligned} \tag{5}$$

식 (5)와 식 (6)에서 $\theta_H$항에 큰 차이는 없어 보이지만 정수부분의 연산이 없어지므로 곱셈과 덧셈 연산에서 상당히 많은 연산량이 감소한다.

$$\begin{aligned} \theta_H(x_{\alpha j}, y_{\alpha j}, z_{\alpha j}) &= kR_{\alpha j} \\ &= k(z_j + \frac{p^2}{2z_j}(x_{\alpha j}^2 + y_{\alpha j}^2)) \\ &= \mathrm{mod}\left[k(z_j + \frac{p^2}{2z_j}(x_{\alpha j}^2 + y_{\alpha j}^2))\right] \\ &= 2\pi(\frac{z_j}{\lambda} + \frac{p^2}{2\lambda z_j}(x_{\alpha j}^2 + y_{\alpha j}^2)) \\ &= \mathrm{mod}\left[2\pi(\frac{z_j}{\lambda} + \frac{p^2}{2\lambda z_j}(x_{\alpha j}^2 + y_{\alpha j}^2))\right] \\ &= 2\pi(\theta_Z + \theta_{XY}) = \mathrm{mod}[2\pi(\theta_Z + \theta_{XY})] \\ &= 2\pi \times \mathrm{mod}[\theta_Z + \theta_{XY}] \end{aligned} \tag{6}$$

### 2.2. CGH 알고리즘의 수정

본 절에서는 앞 절에서 설명한 CGH 알고리즘의 효율을 높이기 위해 제안된 방법을 소개한다[9]. 이 수정된 CGH 알고리즘은 하나의 가로줄 단위로 연산이 이루어진다. 먼저, 가로줄의 첫 번째 홀로그램 위치(d=0)에 대해 $\theta_{XY}(x_{\alpha j}, y_{\alpha j}, z_{\alpha j})$를 연산하고 그 이후의 위치(d>0)에 대해서는 가로줄이 끝날 때 까지 이전 위치에서 연산된 결과($\Gamma_{n-1}(x_{\alpha j}, z_j)$)에 일부 값($(n-1)\Delta$)을 보정하여 해당 위치에서의 홀로그램 값($I_\alpha$)을 구한다. 가로줄의 첫 번째 위치에서 연산되어야 하는 과정은 식 (7)과 같다.

$$\begin{aligned} I_\alpha &= \sum_j^N A_j \cos(\theta_H + \Phi_\alpha + \Phi_j) \\ &= \sum_j^N A_j \cos(2\pi \times \mathrm{mod}[(\theta_Z + \theta_{XY})] + \Phi_\alpha + \Phi_j) \\ &= \sum_j^N A_j \cos(2\pi \times \mathrm{mod}\left[\frac{z_j}{\lambda} + \frac{p^2}{2\lambda z_j}(x_{\alpha j}^2 + y_{\alpha j}^2)\right] \\ &\quad + \Phi_\alpha + \Phi_j) \quad (d=0) \end{aligned} \tag{7}$$

만일 그 이후의 위치에 대해 홀로그램을 구하고자 할 경우에 $x_\alpha$는 $(x_\alpha - d)$와 같이 둘 수 있고 이를 식 (6)에 대입하여 정리하면 식 (8)과 같이 유도할 수 있다.

$$\begin{aligned} \theta_{XY}(x_{\alpha j} + d, y_{\alpha j}, z_{\alpha j}) &= \mathrm{mod}\left[\frac{p^2}{2\lambda z_j}((x_{\alpha j} + d)^2 + y_{\alpha j}^2)\right] \\ &= \mathrm{mod}\left[\frac{p^2}{2\lambda z_j}(x_{\alpha j}^2 + 2dx_{\alpha j} + d^2 + y_{\alpha j}^2)\right] \\ &= \mathrm{mod}\left[\frac{p^2}{2\lambda z_j}(x_{\alpha j}^2 + y_{\alpha j}^2) + \frac{p^2}{2\lambda z_j}(2dx_{\alpha j} + d^2)\right] \\ &= \mathrm{mod}\left[\theta_{XY}(x_{\alpha j}, y_{\alpha j}, z_{\alpha j}) + \Gamma_d\right] \end{aligned} \tag{8}$$

식 (8)을 살펴보면 $\theta_{XY}(x_{\alpha j}, y_{\alpha j}, z_{\alpha j})$ 항은 이미 이전에 구한 결과이고 이 결과에 $\Gamma_d$항이 추가된 결과를 보인다. 따라서 가로줄의 첫 번째 위치 이후부터는 첫 번째 위치의 결과인 $\theta_{XY}(x_{\alpha j}, y_{\alpha j}, z_{\alpha j})$에 $\Gamma_d$를 더해주기만 하면 된다. 이러한 방법으로 연산의 간소화를 이룬다.

또한 $\Gamma_d$의 값을 실제로 구해보면 식 (9), (10), 그리고 (11)의 결과를 얻을 수 있고 이 식들의 분석으로부터 식 (12)와 같은 일반화를 시킬 수 있다.

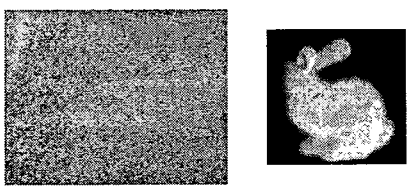
$$\Gamma_1(x_{\alpha j}, z_j) = \mathrm{mod}\left[\frac{p^2}{2\lambda z_j}(2x_{\alpha j}+1)\right] \quad (9)$$

$$\begin{aligned}\Gamma_2(x_{\alpha j}, z_j) &= \mathrm{mod}\left[\frac{p^2}{2\lambda z_j}(4x_{\alpha j}+4)\right] \quad (10)\\ &= \mathrm{mod}\left[\frac{p^2}{2\lambda z_j}(2x_{\alpha j}+1)+\frac{p^2}{2\lambda z_j}(2x_{\alpha j}+1)+\frac{p^2}{2\lambda z_j}\times 2\right]\\ &= \mathrm{mod}\left[\Gamma_1+\Gamma_1+\Delta\right]\end{aligned}$$

$$\begin{aligned}\Gamma_3(x_{\alpha j}, z_j) &= \mathrm{mod}\left[\frac{p^2}{2\lambda z_j}(6x_{\alpha j}+9)\right] \quad (11)\\ &= \mathrm{mod}\left[\frac{p^2}{2\lambda z_j}(4x_{\alpha j}+4)+\frac{p^2}{2\lambda z_j}(2x_{\alpha j}+1)+\frac{p^2}{2\lambda z_j}\times 4\right]\\ &= \mathrm{mod}\left[\Gamma_2+\Gamma_1+2\Delta\right]\end{aligned}$$

$$\Gamma_n(x_{\alpha j}, z_j) = \mathrm{mod}\left[\Gamma_{n-1}+\Gamma_1+(n-1)\Delta\right] \quad (12)$$
$$\left(where\ \Delta(z_j) = \mathrm{mod}\left[\frac{p^2}{\lambda z_j}\right]\right)$$

그림 2에는 앞서 설명된 알고리즘을 바탕으로 제작된 디지털 홀로그램과 HoloVision[10]을 이용하여 복원한 결과를 나타내었다.



(a) (b)

그림 2. CGH 및 복원 결과 (a) 디지털 홀로그램(1,280×1,024) (b) 복원된 객체(200×200)
Fig. 2. CGH and reconstruction result (a) digital hologram (1,280×1,024) (b) reconstructed object (200×200)

## III. 제안한 하드웨어 구조

본 장에서는 수정된 CGH 알고리즘을 바탕으로 하여 새로운 하드웨어 구조를 제안한다.

### 3.1. 하드웨어 구현의 필요성

H×V 크기의 3차원 객체에 대한 M×N 크기의 홀로그램을 만들기 위해서는 M×N×H×V 만큼의 기본 반복수가 필요하다. 한 번의 반복수에는 또 세부적으로 다수의 곱셈과 누적 덧셈 연산이 필요하다. 이러한 이유로 인해서 한 장의 홀로그램을 소프트웨어를 이용해서 생성하고자 할 경우에 최소 수 분 이상의 시간이 소요된다[10]. 광학계를 모델링하여 수학적인 연산을 통해 디지털 방식으로 홀로그램을 생성하고자 할 경우에는 아무리 좋은 컴퓨터를 사용한다 할지라도 소프트웨어로는 실시간 처리가 불가능하다. 따라서 CGH를 실시간으로 생성하기 위해서는 하드웨어로의 구현이 필수적이다.

### 3.2. 정밀도 분석

CGH는 다른 신호처리 기술들과 다르게 연산되는 변수들이 너무나 다른 수의 범위를 가진다. 즉 소수점이 거의 없고 정수부만 큰 값을 가지는 변수와 정수부는 거의 없소 소수부만을 큰 자리수를 가지는 변수 사이의 연산이 빈번하다. 이러한 연산은 연산기의 크기만 증가시키면서 효율성은 떨어뜨리는 결과를 가져온다. 따라서 하드웨어를 구현하기에 앞서서 중간 연산들 사이의 수 범위와 정밀도를 분석하여 구현하고자 하는 하드웨어의 비트너비(버스크기)를 정한다.

중간 연산결과들의 수 범위에 따른 비트수를 표 1에 정리하였다. 본 논문에서는 중간 연산결과를 모두 반영할 수 있는 비트수를 선정하였기 때문에 소프트웨어로 연산한 결과와 하드웨어 연산 결과는 일치한다.

표 1. 정밀도 분석에 따른 중간 연산 단계별 비트수
Table 1. Bit width on intermediate arithmetic steps by precision analysis

| | Item | Miin. Value | Max. Value | Bit Width | |
|---|---|---|---|---|---|
| | | | | Int | Dec |
| 1 | $\frac{p^2}{2\lambda z_j}$ | 0.00056956292 | 0.0001708688 | 0 | 32 |
| 2 | $x_{\alpha j}^2 = (x_\alpha - x_j)^2$ | 0 | 1,638,400 | 21 | 0 |
| 3 | $\frac{z_j}{\lambda}$ | 78989.4155 | 2369668.246 | 22 | 16 |
| 4 | $\frac{p^2}{2\lambda z_j}(x_{\alpha j}^2+y_{\alpha j}^2)$ | 0 | 358.33801 | 9 | 20 |
| 5 | $\frac{p^2}{2\lambda z_j}(2x_{\alpha j}+1)$ | 0.136438799 | 0.700306097 | 0 | 32 |
| 6 | $z_j$ | 0.5 | 1.5 | 1 | 8 |

### 3.3. LUT를 이용한 동작의 간략화

식 (7)과 표 1을 분석하면 일부 동작을 LUT (Look-Up Table)로 미리 만들어서 연산과정을 간략화 할 수 있음으로 알 수 있다. 식 (7)로부터 가장 먼저 알 수 있는 것은 코사인 함수에 대한 연산을 LUT로 만드는 것이다. 이는 일반적으로 신호처리 분야에서 사용하는 방법이다. 또한 표 1을 살펴보면 1, 3번 항목을 LUT로 만들 수 있음을 알 수 있다. 1, 3번 항목에서 $z_j$는 0.5에서 1.5 사이의 값을 가지는 깊이 정보로써 최대 256가지를 가진다. 만일에 $z_j$가 너무 많은 종류의 값을 가진다면 LUT로 만드는 것이 더욱 하드웨어 자원과 효율에 있어서 단점을 가져올 것이다. 따라서 이들 값은 256개의 주소를 가지는 LUT로 미리 만든 후 연산을 하지 않고 결과를 출력한다.

### 3.4. 제안한 하드웨어 구조

본 절에서는 앞서 설명한 수정된 CGH 알고리즘과 LUT 등을 이용하여 하드웨어 구조를 제안한다. 먼저 본 논문의 핵심에 해당하는 CGH 셀의 구조를 제안하고 동작을 분석한다. 제안한 CGH 셀은 홀로그램의 하나의 가로줄을 연산할 수 있는 단위 연산기로서 동작의 독립성을 보장하여 단순한 추가에 의해서 연산 성능을 비례적으로 증가시킬 수 있다. 또한 전체를 파이프라인화하여 10클록의 대기지연시간 이후부터는 한 클록에 하나의 값을 연속적으로 출력할 수 있다.

이전에 발표되었던 연구결과들과 비교하여 다음과 같은 특성 혹은 장점을 가진다.

① 100% 파이프라인 구조
② 독립적인 단위 연산기 구조
③ 연산기의 확장에 따라 성능이 비례적으로 증가

#### 3.4.1. CGH 셀의 구조 제안

제안한 CGH 셀은 d=0일 때의 초기값 연산과 d>0일 때의 보정 연산을 모두 포함한다. d=0에서의 초기값을 연산하는 것(그림 3에서 점선 부분)은 파이프라인 경로에 포함시키지 않는다. 왜냐하면 이 연산은 초기의 준비 단계에서 한번만 이루어진 후에 다시 연산되지 않기 때문이다. 이를 파이프라인 경우에 포함시키면 불필요한 대기지연시간만 추가하는 것이 된다.

초기값을 연산하는 블록은 CGH 셀에 포함시키지 않고 미리 다른 블록에서 연산해 둘 수 있다. 이렇게 구성을 한다면 하드웨어 자원은 절약될 것이다. 그러나 CGH 셀의 독립성이 떨어지기 때문에 확장을 할 경우에 따로 다른 제어 동작이 추가되거나 부가 하드웨어가 필요하게 된다. 따라서 확장성과 독립성을 위해서 포함시키는 것으로 하였다.

그림 3에 앞서 설명한 식을 바탕으로 하여 하드웨어의 연산 방법을 표시하였다. 5개의 곱셈기와 7개의 덧셈기(뺄셈기)를 사용한다. CGH 셀에서 곱하기 2 등의 연산은 쉬프트 동작을 이용하여 처리하였고 Scalar는 실제로 사용되지 않는다. DR은 누적 연산 혹은 값을 보유할 경우에 사용하는 데이터 레지스터이다. 추후에 파이프라인 레지스터와 구별하기 위해서 명시하였다.

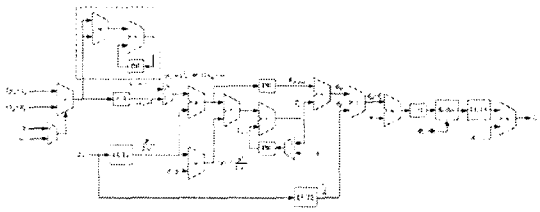


그림 3. 제안한 CGH 셀의 구조
Fig. 3. The proposed architecture of CGH cell

하드웨어내의 조합회로의 지연시간을 고려하고 임계경로를 분석하여 그림 4와 같이 CGH 셀의 내부를 단위 연산으로 분리하였다. 총 10개의 단계로 나뉘는 것을 볼 수 있다.

그림 4에서 분석한 CGH 셀의 내부 단위 연산시간을 바탕으로 그림 5와 같이 CGH 셀을 파이프라인화하였다. 총 10 단계의 파이프라인 단계를 가지므로 10클록의 대기지연시간 이후에 한 클록 당 하나의 결과를 출력할 수 있다. 그림에서 R이 파이프라인 레지스터에 해당한다. 앞서 설명한 것과 같이 일반적으로 홀로그램을 재

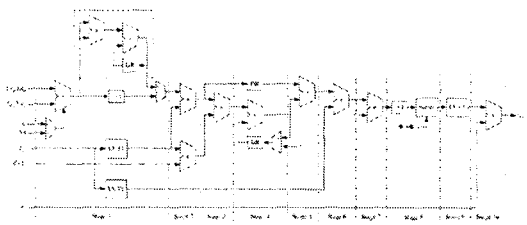


그림 4. 파이프라인을 위한 제안한 CGH 셀의 타이밍 분석
Fig. 4. Timing analysis of CGH cell for pipeline

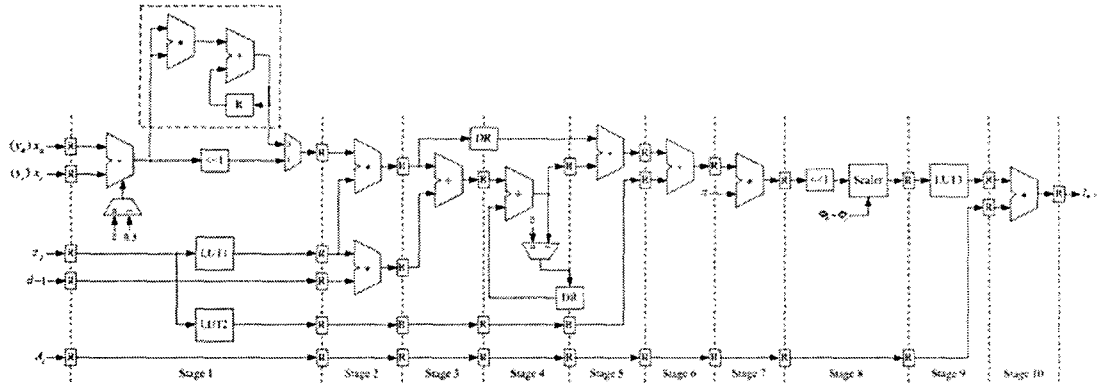


그림 5. 파이프라인화된 CGH 셀의 구조
Fig. 5. Pipelined CGH cell architecture

생할 경우에 홀로그램과 3차원 객체의 초기 혹은 오프셋 위상($\Phi_\alpha + \Phi_j$)은 사용하지 않는다. 즉 8번째 파이프라인 단계는 생략될 수 있고 이 경우에 대기지연시간은 9클록이 된다. 이는 대기지연시간의 감소일 뿐이므로 성능과의 거의 무관하다.

앞서서 설명하였던 결과 식들과 그림 5를 살펴보면 식들에서 공통적으로 사용되거나 미리 연산되어야 하는 것들($\frac{z_j}{\lambda}$, $\frac{p^2}{2\lambda z_j}$)이 그림 5에서도 미리 연산된 후 파이프라인 경로를 따라가면서 중간 단계들에서 사용되는 것을 볼 수 있다.

파이프라인을 적용하는데 있어서 누적 덧셈이나 데이터를 보유하는 동작이 파이프라인 경로에 의해서 흐트러지지 않도록 유의해야 한다.

3.4.2. CGH 커널의 구조 제안

제안한 CGH 커널은 CGH 셀 0~(N-1), Zero-Value Controller, Bypass, 그리고 Scheduler로 구성된다. 앞 절에서 설명한 CGH 셀이 핵심 블록에 해당하고 각각의 CGH 셀이 홀로그램의 하나의 가로줄을 담당한다. 본 논문에서는 1,280×1,024 크기의 홀로그램을 생성하고 있으므로 1,280개의 CGH 셀을 사용하고 있다. 만일 연산하여도 결과를 변경하지 않는 입력의 경우(즉, 광원이 아닐 경우)에 Zero-Value Controller에서 의해서 검출되고 Bypass를 블록을 통해서 단순히 타이밍만이 조절되어 출력된다. 각 CGH 셀들과 Bypass 블록을 통해 출력된 결과들은 Scheduler를 통해서 정렬된 후에 CGH 커널로부터 출력된다.

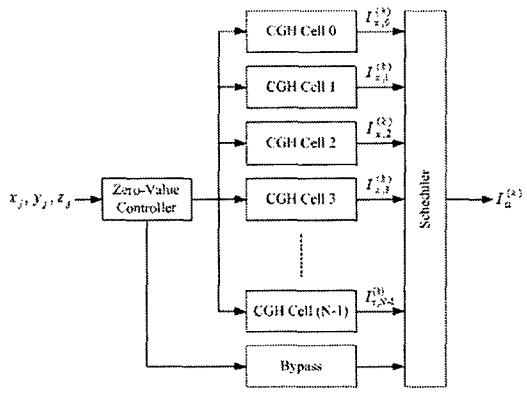



그림 6. 제안한 CGH 커널의 구조
Fig. 6. The proposed architecture of CGH kernel

3.4.3. CGH 프로세서의 구조

제안한 CGH 프로세서는 Input Interface, Output Interface, Main Controller, SDRAM Controller, DMA 등으로 구성된다. 입력된 3차원 객체에 대한 정보는 먼저 SDRAM에 저장된다. 그 후 다음 3차원 객체에 대한 정보가 입력될 때 이전에 저장된 3차원 객체의 정보를 호출하여 홀로그램을 생성한다. 앞서서 설명한 것과 같이 CGH 커널을 통해서 홀로그램은 생성된다. 중간 연산결과들은 On-Chip DP SRAM에 저장되고 최종적인 홀로그램을 생성할 때까지 데이터의 저장소로 이용된다. 최종적으로 홀로그램이 생성되면 CGH 커널은 다음 3차원 객체에 대한 홀로그램의 생성을 준비하게 되고 이와 동시에 DMA에 의해서 DP SRAM에 저장된 홀로그램은 Output Interface를 거쳐 외부로 출력된다.

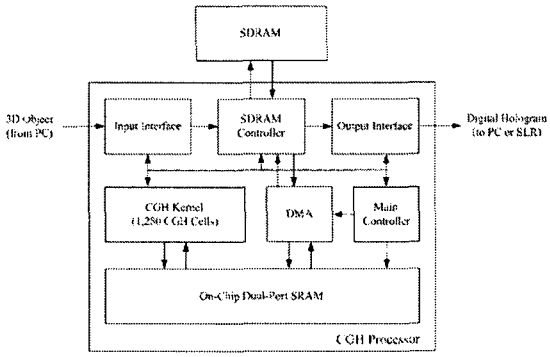



그림 7. 제안한 CGH 프로세서의 구조
Fig. 7. The proposed architecture of CGH processor

## Ⅳ. 구현결과

제안한 하드웨어 구조는 VHDL을 이용하여 설계하였다. 구현에 이용된 실험 환경 및 변수들의 값을 표 2에 나타냈다. 이전 연구들과의 비교를 위해서 타겟으로 한 FPGA 칩은 Xilinx의 XC2VP70으로 700만 게이트급에 해당한다. 총 4개의 FPGA 칩을 이용하여 구현하였고 각 FPGA 칩 당 320개의 CGH 커널이 맵핑되고 약 70%의 CLB를 사용한다. 각 FPGA 당 하나의 SDRAM이 필요하고 On-Chip DP SRAM은 320 Kbyte를 사용한다. 또한 CGH 커널 내의 Scheduler에 의해서 160 Kbyte의 SRAM이 사용된다.

표 2에서 Hologram Size, Pixel Pitch, 그리고 Wavelength 등의 값은 임의로 정한 것이 아니고 본 연구팀이 보유한 디지털 홀로그램 재생 장치의 사양이 그와 같기 때문이다.

표 2. 실험 환경 및 변수
Table 2. Experimental environment and variables

| Item | Specification |
|---|---|
| Object Point | 40,000 |
| Hologram Size | 1,280×1,024 |
| Pixel Pitch (p) | 10.4μm |
| Wavelength (λ) | 532nm (Green Laser) |
| FPGA | Xilinx XC2VP70 |

표 3에는 제안한 하드웨어의 구현 결과를 서론에서 소개되었던 이전 연구들과의 비교하였다.

구현된 하드웨어는 3가지 사양으로 결과를 나타내었는데 이 중에서 첫 번째 세로줄에 나타난 결과가 실제 결과이고 나머지 두 결과는 이전 연구결과들 중에서 가장 좋은 결과를 보이는 [4]의 결과와 비교하기 위한 예측 결과이다. 구현한 하드웨어는 1,280개의 파이프라인화된 CGH 셀을 이용하여 클록 당 1개의 결과를 출력할 수 있다. 따라서 식 (13)과 같이 한 장의 홀로그램을 생성하는데 시간이 소요되는 것을 알 수 있다.

$$1{,}024(1{,}280\ CGH\ Cells)\times 5ns\times 40{,}000 \quad (13)$$
$$+4\times 5ns(initial\ value)=0.205\text{sec}$$

표 3. 구현결과 및 이전 연구와의 비교
Table 3. Implementation results and comparison with the previous results

| Item | Implementation Results | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | [4] | | [5] | [6] | [7] | Ours | | |
| Object Point | 10,000 | | 400 | 70 | 1,000 | 40,000 | 10,000 | 10,000 |
| Hologram Size | 1,408×1,050 | | 800×600 | 800×600 | 960×600 | 1,280×1,024 | 1,408×1,050 | 1,408×1,050 |
| Target H/W | 4 FPGA (XC2VP70) | | 1 FPGA (EP20K300E) | nVIDIA's GeForce 6600 | nVIDIA's GeForce 7800 | 4 FPGA (XC2VP70) | | |
| Frequency (MHz) | 166MHz | | 35MHz | - | - | 200MHz | 166MHz | 166MHz |
| Time(sec)/CGH | 0.0679 | 0.0232 | 0.15 | 0.0067 | 0.1 | 0.205 | 0.0633 | 0.0159 |
| Hologram/Time (sec) | 14.73 | 43.10 | 6.667 | 149.254 | 10 | 4.88 | 15.80 | 62.90 |
| Hardware Feature | Parallel Unit Cells | | 27 Parallel Unit Cells | Commercial Graphic Board | Commercial Graphic Board | Parallel Unit Cells | | |
| | 1,408 | 5,632 | | | | 1,280 | 1,408 | 5,632 |

먼저 구현 결과를 살펴보면 200MHz의 동작속도로 동작이 가능하였고 1,280개의 CGH 셀을 이용하여 4만개의 광원을 가지는 3차원 객체에 대해서 1,280×1,024 크기의 홀로그램을 0.205초에 한 장씩 생성할 수 있다. 즉, 초당 4.88개의 홀로그램을 만들 수 있는 속도이다. [4]의 결과를 살펴보면 초당 30 프레임 이상의 실시간 동작을 하기 위해서는 총 5,632개의 병렬 단위 셀이 필요하다. 만일 구현된 하드웨어와 [4]의 하드웨어를 동일한 조건으로 둘 경우에 홀로그램 생성속도는 0.0046(=0.0679-0.0633)초 더 빠른 결과를 보이기 때문에 약 7%의 성능 향상이 이루어진 것을 볼 수 있다. 또한 실시간 처리를 하기위해 CGH 셀을 5,632개 사용할 경우에 제안한 하드웨어는 62.9개의 홀로그램을 생성할 수 있는 반면에 [4]는 41.10개의 홀로그램을 생성할 수 있다. 따라서 제안한 하드웨어가 더욱 좋은 성능을 보이는 것을 확인할 수 있다.

구현된 하드웨어를 이용하여 생성된 디지털 홀로그램은 광학 시스템을 통해서 재생하여 검증하였다. 그림 8에 광학 시스템을 위한 장치들을 보이고 있고 그림 9에는 재생된 디지털 홀로그램을 디지털 카메라를 이용하여 촬영한 사진을 보이고 있다.

## V. 결 론

본 논문에서는 하드웨어의 구현을 위해 수정된 CGH 알고리즘을 바탕으로 홀로그램을 생성할 수 있는 하드웨어를 구현하고 검증하였다.

디지털 홀로그램을 효율적으로 연산할 수 있는 CGH 셀의 구조를 제안한 후에 CGH 셀의 확장을 통해서 CGH 커널을 구현하였다. 그리고 CGH 커널과 기타 하드웨어 장치들을 이용하여 CGH 프로세서를 구현하여 디지털 홀로그램을 생성할 수 있었다. Xilinx의 XC2VP70 FPGA 칩 4개를 이용하여 CGH 프로세서를 구현하였다. 각 FPGA 칩 당 320개의 CGH 커널이 맵핑되고 약 70%의 CLB를 사용하였다. 각 FPGA 당 하나의 SDRAM이 필요하고 On-Chip DP SRAM은 320 Kbyte를 사용하였고 CGH 커널 내의 Scheduler에 의해서 160 Kbyte의 SRAM이 사용하였다. 구현된 하드웨어는 0.205초의 시간동안에 한 장의 홀로그램을 생성할 수 있었고 CGH 셀의 확장을 통해서 실시간 동작이 가능한 것을 확인하였다. 이 결과는 기존 연구 중 가장 성능이 우수한 방법과 비교할 때 7% 정도의 성능향상을 보여, 본 논문에서 제안하는 방식의 우수성을 입증하였다.

본 논문에서 제안되고 구현한 하드웨어는 FPGA를 타겟으로 하고 있고 설계를 하는데 있어서 많은 부분이 FPGA가 제공하는 기능들을 이용하였다. 따라서 이들을 전용 블록으로 설계한 후에 ASIC화가 필요할 것으로 보인다. ASIC화를 통해서 큰 하드웨어 자원을 사용하는 단점을 해소해야 한다. 또한 곱셈기에 걸쳐있는 임계경로의 축소를 위해서 파이프라인화된 병렬 곱셈기의 구현 및 적용이 필요하다. 마지막으로 디지털 홀로그램의 재생 특성 혹은 디지털 홀로그램 자체의 특성을 분석하여 하드웨어 내 비트수의 최적화 과정이 수반될 경우에 더욱 최적화된 하드웨어 구현이 가능할 것으로 판단되며, 추후에 이러한 점을 보완하고자 한다.

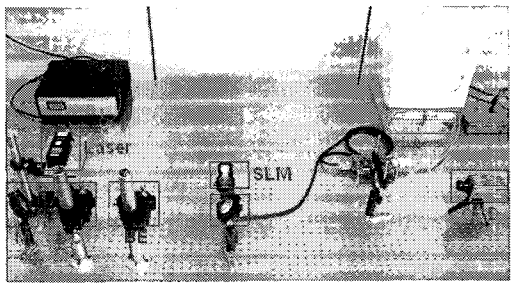



그림 8. 구현한 하드웨어를 통해 생성된 디지털 홀로그램의 재생을 위한 광학 시스템
Fig. 8. Optical system for display of digital hologram generated by the implemented hardware

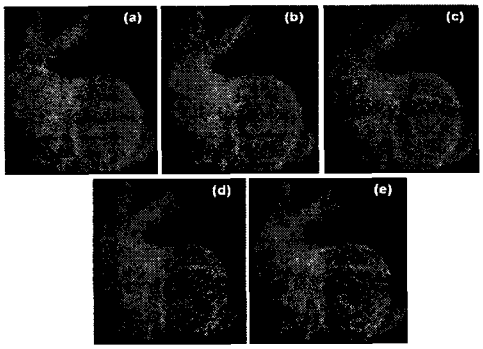



그림 9. 재생된 디지털 홀로그램; 복원거리 (a) 1,060㎜, (b) 1,070㎜, (c) 1,100㎜, (d) 1,130㎜, (e) 1,140㎜
Fig. 9. Displayed digital hologram; reconstruction distance (a) 1,060㎜, (b) 1,070㎜, (c) 1,100㎜, (d) 1,130㎜, (e)1,140㎜

## 참고문헌


[1] T. Motoki, H. Isono, and I. Yuyama, "Present Status of Three-Dimensional Television Research," Proc. IEEE 83(7): 1009-1021(July 1995).

[2] J. K. Chung and M. H. Tsai, Three-Dimensional Holographic Imaging, John Wiley & Sons, Inc., 2002.

[3] P. Hariharan, Basics of Holography, Cambridge University Press, May 2002.

[4] T. Ito, N. Masuda, K. Yoshimura, A. Shiraki, T. Shimobaba, and T. Sugie, "Special-Purpose computer HORN-5 for a real-time electroholography," Optics Express, Vol. 13, No. 6, March 2005.

[5] T. Shimobaba, A. Shiraki, N. Masuda, and T. Ito, "Electroholographic display unit for three-dimensional display by use of special-purpose computational chip for holography and reflective LCD panel," Optics Express, Vol. 13, No. 11, May 2005.

[6] N. Masuda, T. Ito, T. Tanaka, A. Shiraki, and T. Sugie, "Computer generated holography using a graphics processing unit," Optics Express, Vol. 14, No. 2, January 2006.

[7] L. Ahrenberg, P. Benzie, M. Magnor, and J. Watson, "Computer generated holography using parallel commodity graphics hardware," Optics Express, Vol. 14, No. 17, August 2006.

[8] W. G. Joseph, Introduction to Fourier Optics (3rd edition), Roberts & Company, 2005.

[9] http://www2.edge.no/projects/index.php?expn=2&target=holovision/about.php

[10] T. S., and T. I., "An Efficient Computational Method Suitable for Hardware of Computer-Generated Hologram with Phase Computation by Addition," Computer Physics Communications, Vol. 138, pp. 44-52, 2001.


## 저자소개

**서 영 호(Young-Ho Seo)**

1999년 2월 광운대학교 전자재료공학과 졸업(공학사)
2001년 2월 광운대학교 일반대학원 졸업(공학석사)
2000년 3월~2001년 12월 : 인티스닷컴(주) 연구원
2004년 8월 : 광운대학교 일반대학원 졸업(공학박사)
2003년 6월~2004년 6월 : 한국전기연구원 연구원
2004년 12월~2005년 8월 : 유한대학 연구교수
2005년 9월~현재 : 한성대학교 전임강사
※관심분야 : 2D/3D 영상 및 비디오 처리, 디지털 홀로그램, SoC 설계, 워터마킹/암호화

**최 현 준(Hyun-Jun Choi)**

2003년 2월 광운대학교 전자재료공학과 졸업(공학사)
2005년 2월 광운대학교 일반대학원 졸업(공학석사)
2005년 3월~현재 광운대학교 일반대학원 박사과정
※관심분야 : 영상압축, 워터마킹, 암호학, FPGA/ASIC 설계, Design Methodology

**김 동 욱(Dong-Wook Kim)**

1983년 2월 : 한양대학교 전자공학과 졸업(공학사)
1985년 2월 : 한양대학교 대학원 졸업(공학석사)
1991년 9월 Georgia공과대학 전기공학과 졸업(공학박사)
1992년 3월~현재 광운대학교 전자재료공학과 정교수. 광운대학교 신기술 연구소 연구원
2000년 3월~2001년 12월 인티스닷컴(주) 연구원
※관심분야 : 디지털 VLSI Testability, VLSI CAD, DSP 설계, Wireless Communication