

컴퓨터 생성 홀로그래를 위한 VLSI 구조

종신회원 서영호*, 정회원 최현준**, 종신회원 김동욱**

VLSI Architecture for Computer-Generated Hologram

Young-Ho Seo* *Lifelong Member*, Hyun-Jun Choi** *Regular Member*,
Dong-Wook Kim** *Lifelong Member*

요약

본 논문에서는 실시간으로 컴퓨터 생성 홀로그래를 생성할 수 있는 VLSI 구조를 제안하고 하드웨어로 구현하였다. 고속으로 디지털 홀로그래를 생성할 수 있는 수정된 알고리즘을 도입하고, 하드웨어 구현을 위해 재해석하였다. 수치 및 시각적인 정밀도 분석으로부터 하드웨어 내부의 비트 너비를 구하였다. CGH 알고리즘의 분석과 정밀도 분석 결과로부터 CGH 셀의 구조를 제안하였다. CGH 셀의 구조와 알고리즘의 특성으로부터 동작 순서를 분석하였고, 파이프라인 구조와 동작적인 타이밍을 제안하였다.

Key Words : Digital Hologram, Computer-generated Hologram, VLSI, Hardware Design

ABSTRACT

In this paper, we proposed a new VLSI architecture which can generate computer-generated hologram (CGH) in real-time and implemented to hardware. The modified algorithm for high-performance CGH was introduced and re-analyzed for designing hardware. From both numerical and visual analysis, the internal number system of hardware was decided. CGH algorithm and precision analysis enabled to propose a new cell architecture for CGH. The operational sequence was analyzed with the architecture of CGH cell and the characteristics of the modified CGH algorithm, and finally the pipelined architecture and the operational timing were proposed.

I. 서론

홀로그래피는 1948년 Gabor에 의해 최초로 제안된 이래 3차원 정보를 기록할 수 있다는 특징 때문에 많은 연구자들의 관심을 끌었다. 기존의 홀로그래피는 홀로그래 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 홀로그래의 제작을 연구해 오고 있다^[1].

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

CGH를 이용하여 한 프레임에 해당하는 하나의 객체를 위한 홀로그래가 생성되기 위해서는 많은 연산량과 시간이 소요되기 때문에 소프트웨어로 처리하는 것은 불가능하고 하드웨어로 구현되어야만 한다^[3]. 따라서 지금까지 많은 연구가 진행되어 왔

* 본 논문은 교육인적자원부, 산업자원부, 노동부의 출연금 및 보조금으로 수행한 최우수실험실지원사업의 연구결과입니다.

** 본 논문은 2008 JCCI 우수논문으로 추천되었습니다.

* 광운대학교 교양학부 IT전공 (yhseo@kw.ac.kr), **광운대학교 전자재료공학과 ({hjchoi, dwkim}@kw.ac.kr)
논문번호 : KICS2008-05-221 접수일자 : 2008년 5월 19일, 최종논문접수일자 : 2008년 7월 9일

다⁴⁻⁷). [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 알고리즘을 수정보완하기 보다는 이를 하드웨어로 구현할 때 보다 효율적이고 고속으로 동작할 수 있는 설계에 그 목적을 두고 있다.

본 논문은 다음과 같이 구성되어 있다. II장에서는 CGH의 원리와 하드웨어 구현을 위해 수정된 CGH 알고리즘을 설명한다. III장에서는 제안한 하드웨어 구조를 설명하고 IV장에서는 구현결과를 보이고 기존 연구와 비교한다. 마지막으로 V장에서는 결론을 맺는다.

II. 컴퓨터 생성 홀로그램의 개요

2.1 CGH 알고리즘

홀로그램은 광학계를 이용하여도 취득할 수 있지만 이러한 광학계 자체를 수학적으로 모델링한 후에 연산에 의해서 취득할 수도 있다. 이러한 수학적 연산을 통해 얻어진 홀로그램을 컴퓨터 생성 홀로그램(computer-generated hologram, CGH)이라고 한다. 여러 종류의 CGH가 있지만 본 논문에서는 “Phase” 방식을 사용한다. 객체에서 CCD로 입사되는 파에서 위상 성분만을 이용하여 홀로그램을 생성하는 것이고 그 증명³은 본 논문에서 다루지 않는다.

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.

CGH는 식 (1)과 같이 정의되는데 앞서서 설명한 것과 같이 홀로그램의 위상(θ_H)으로부터 홀로그램의 강도(I_α)를 얻는 방법이다. 여기서 N은 3차원 객체의 광원수를 뜻한다. θ_H 는 kR_{α_j} 로 정의되고 이는 식 (2)와 같다. 여기서 k는 참조파의 파수로 $2\pi/\lambda$ 로 정의되고 λ 는 532nm의 값을 사용한다. x_α 와 y_α 는 홀로그램내의 위치를 뜻하고 x_j , y_j , 및 z_j 는 3차원 객체의 위치를 나타낸다. p는 픽셀의 크기를 나타내고 10.4 μ m의 값을 사용한다.

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

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

식 (2)는 Fresnel 근사화 기법에 의해서 식 (3)으로 근사화된다^[8].

$$R_{\alpha_j} \cong z_j + \frac{p^2}{2z_j}(x_{\alpha_j}^2 + y_{\alpha_j}^2) \quad (3)$$

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

$$\begin{aligned} kR_{\alpha_j} &= k(z_j + \frac{p^2}{2z_j}(x_{\alpha_j}^2 + y_{\alpha_j}^2)) \quad (4) \\ &= 2\pi(\frac{z_j}{\lambda} + \frac{p^2}{2\lambda z_j}(x_{\alpha_j}^2 + y_{\alpha_j}^2)) \\ &= 2\pi(\theta_Z + \theta_{XY}) \end{aligned}$$

식 (4)에 의해서 식 (1)은 식 (5)와 같이 나타낼 수 있다. 식 (5)와 같이 유도하는 이유는 코사인 함수는 2 π 를 주기로 하여 반복되므로 ($\theta_Z + \theta_{XY}$)의 결과에서 정수부분을 고려하지 않아도 되기 때문이다.

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

소수부분의 연산만 수행함을 나타내는 연산자 mod[]를 도입한다면 식 (5)에서 θ_H 항은 식 (6)과 같이 유도할 수 있다. 식 (5)와 식 (6)에서 θ_H 항에

큰 차이는 없어 보이지만 정수부분의 연산이 없어 지므로 곱셈과 덧셈 연산에서 상당히 많은 연산량이 감소한다.

$$\begin{aligned}
 kR_{\alpha_j} &= k(z_j + \frac{p^2}{2z_j}(x_{\alpha_j}^2 + y_{\alpha_j}^2)) \quad (6) \\
 &= \text{mod} \left[k(z_j + \frac{p^2}{2z_j}(x_{\alpha_j}^2 + y_{\alpha_j}^2)) \right] \\
 &= \text{mod} \left[2\pi \left(\frac{z_j}{\lambda} + \frac{p^2}{2\lambda z_j}(x_{\alpha_j}^2 + y_{\alpha_j}^2) \right) \right] \\
 &= \text{mod} [2\pi(\theta_Z + \theta_{XY})] \\
 &= 2\pi \times \text{mod} [\theta_Z + \theta_{XY}]
 \end{aligned}$$

식 (6)을 이용하여 식 (5)을 재정리하면 식 (7)과 같다.

$$I_\alpha = \sum_j^N A_j \cos(2\pi \times \text{mod}[\theta_Z + \theta_{XY}] + \Phi_\alpha + \Phi_j) \quad (7)$$

식 (7)은 컴퓨터 프로그래밍에 의해서 수행되고 복원될 수 있다. 컴퓨터 프로그래밍에 의해서 뿐만 아니라 SLM을 이용하여 광학적으로 복원할 수도 있다.

2.2 수정된 CGH 알고리즘

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)Δ)을 보정하여 해당 위치에서의 홀로그램 값 (I_α)을 구한다. 가로줄의 첫 번째 위치에서 연산되어야 하는 과정은 식 (7)과 같다.

$$\begin{aligned}
 I_\alpha &= \sum_j^N A_j \cos(\theta_H + \Phi_\alpha + \Phi_j) \\
 &= \sum_j^N A_j \cos(2\pi \times \text{mod}[(\theta_Z + \theta_{XY})] + \Phi_\alpha + \Phi_j) \\
 &= \sum_j^N A_j \cos(2\pi \times \text{mod} \left[\frac{z_j}{\lambda} + \frac{p^2}{2\lambda z_j}(x_{\alpha_j}^2 + y_{\alpha_j}^2) \right] + \Phi_\alpha + \Phi_j) \quad (8)
 \end{aligned}$$

만일 그 이후의 위치에 대해 홀로그램을 구하고자 할 경우에 x_α 는 $(x_\alpha - d)$ 와 같이 둘 수 있고 이를 식 (6)에 대입하여 정리하면 식 (9)과 같이 유도

할 수 있다.

$$\begin{aligned}
 \theta_{XY}(x_{\alpha_j} + d, y_{\alpha_j}, z_{\alpha_j}) &= \text{mod} \left[\frac{p^2}{2\lambda z_j}((x_{\alpha_j} + d)^2 + y_{\alpha_j}^2) \right] \quad (9) \\
 &= \text{mod} \left[\frac{p^2}{2\lambda z_j}(x_{\alpha_j}^2 + 2dx_{\alpha_j} + d^2 + y_{\alpha_j}^2) \right] \\
 &= \text{mod} \left[\frac{p^2}{2\lambda z_j}(x_{\alpha_j}^2 + y_{\alpha_j}^2) + \frac{p^2}{2\lambda z_j}(2dx_{\alpha_j} + d^2) \right] \\
 &= \text{mod} [\theta_{XY}(x_{\alpha_j}, y_{\alpha_j}, z_{\alpha_j}) + \Gamma_d]
 \end{aligned}$$

식 (9)를 식 (5)에 대입하면 I_α 는 식 (10)과 같다.

$$I_\alpha = \sum_j^N A_j \cos(2\pi \times \text{mod}[\theta_{XY} + \theta_Z + \Gamma_d] + \Phi_\alpha + \Phi_j) \quad (10)$$

식 (10)을 살펴보면 $\theta_{XY}(x_{\alpha_j}, y_{\alpha_j}, z_{\alpha_j})$ 항은 이미 이전에 구한 결과이고 이 결과에 Γ_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 Γ_d 를 더해주기만 하면 된다. 이러한 방법으로 연산의 간소화를 이룬다.

또한 Γ_d 의 값을 실제로 구해보면 식 (10), (11), 그리고 (12)의 결과를 얻을 수 있고 이 식들의 분석으로부터 식 (13)과 같은 일반화를 시킬 수 있다.

$$\Gamma_1(x_{\alpha_j}, z_j) = \text{mod} \left[\frac{p^2}{2\lambda z_j}(2x_{\alpha_j} + 1) \right] \quad (11)$$

$$\begin{aligned}
 \Gamma_2(x_{\alpha_j}, z_j) &= \text{mod} \left[\frac{p^2}{2\lambda z_j}(4x_{\alpha_j} + 4) \right] \quad (12) \\
 &= \text{mod} \left[\frac{p^2}{2\lambda z_j}(2x_{\alpha_j} + 1) + \frac{p^2}{2\lambda z_j}(2x_{\alpha_j} + 1) \right] \\
 &\quad + \frac{p^2}{2\lambda z_j} \times 2 \\
 &= \text{mod} [\Gamma_1 + \Gamma_1 + \Delta]
 \end{aligned}$$

$$\begin{aligned}
 \Gamma_3(x_{\alpha_j}, z_j) &= \text{mod} \left[\frac{p^2}{2\lambda z_j}(6x_{\alpha_j} + 9) \right] \quad (13) \\
 &= \text{mod} \left[\frac{p^2}{2\lambda z_j}(4x_{\alpha_j} + 4) + \frac{p^2}{2\lambda z_j}(2x_{\alpha_j} + 1) \right] \\
 &\quad + \frac{p^2}{2\lambda z_j} \times 4 \\
 &= \text{mod} [\Gamma_2 + \Gamma_1 + 2\Delta]
 \end{aligned}$$

객체의 첫 번째 광원으로 중간 홀로그램을 구하여 저장하고, 두 번째 광원으로 두 번째 중간 홀로그램을 구한 후 첫 번째 중간 홀로그램과 더한다. 객체를 구성하는 광원의 수 만큼 이러한 과정을 반복한다.

$$\Gamma_d(x_{\alpha j}, z_j) = \text{mod}[\Gamma_{d-1} + \Gamma_1 + (d-1)\Delta] \quad (14)$$

$$\left(\text{where } \Delta(z_j) = \text{mod}\left[\frac{p^2}{\lambda z_j}\right] \right)$$

III. 제안한 하드웨어 구조

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

3.1 연산의 분석

하드웨어로 구현하기 위해서 식 (10)을 재정의한다. 식 (10)을 다시 정리한 식 (13)에서 처음에 계산된 후 동일한 값을 유지하는 부분과 변화되는 부분을 구분한다. 각각은 초기항(i_{init})과 업데이트항(i_{update})이라 부르고 식 (14)와 (15)로 정의한다.

식 (14)와 (15)으로부터 객체 내의 하나의 광원에 의한 중간 홀로그램 내의 한 점의 크기는 식 (16)과 같다.

$$I_{\alpha} = \sum_j^N A_j \cos(2\pi(\theta_z + \theta_{XY,d=0} + \Gamma_d)) \quad (13)$$

$$= \sum_j^N A_j \cos(2\pi(\theta_z + \theta_{XY,d=0} + \Gamma_1 + \Gamma_{d-1} + (d-1)\Delta))$$

$$i_{init} = \theta_z + \theta_{XY,d=0} + \Gamma_1 \quad (14)$$

$$i_{update} = \Gamma_{d-1} + (d-1)\Delta \quad (15)$$

$$I_{phase} = A_j \cos(2\pi(i_{init} + i_{update})) \quad (16)$$

식 (14), (15), 그리고 (16)을 구하기 위해 세부적인 단계로 나누면 표 1과 같다. 초기항에서 구해진 값들은 DR1, DR2, 및 DR3의 레지스터에 저장하고 다음 연산에서 계속적으로 사용한다. 표 1에서 step은 연산의 시간적인 순서를 의미한다.

3.2 CGH 셀의 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 대기 지연 시간 이후부터는 한 클록에 하나의 값을 연속적으로 출력할 수 있다.

표 1. 연산 단계
Table 1. Arithmetic steps

Classification	Steps	Arithmetic	REG
i_{init} Calculation	Step1	$\Theta_{XY,d=0}, \Theta_z$	DR1
	Step2	$\Theta_{XY,d=0} + \Theta_z$	DR2
	Step3	Γ_1	DR3
i_{update} Calculation	Step4	Γ_d	DR4
	Step5	$\Theta_{XY,d=0} + \Theta_z + \Gamma_d$	
i_{phase} Calculation	Step6	$\cos 2\pi\Theta_H$	
	Step7	$A_j \cos 2\pi\Theta_H$	

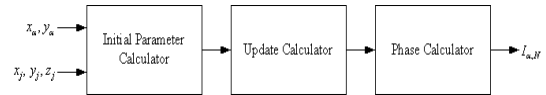


그림 1. 제안한 CGH 셀의 구조
Fig. 1. The proposed architecture of CGH cell

이전에 발표되었던 연구결과들과 비교하여 다음과 같은 특성 혹은 장점을 가진다.

- 100% 파이프라인 구조
- 독립적인 단위 연산기 구조
- 연산기의 확장에 따라 성능이 비례적으로 증가

제안한 CGH 셀은 식 (14), (15), 그리고 (16)의 세 단계로 구성된다. 각 단계는 초기 파라미터 연산기, 업데이트 연산기, 그리고 위상 연산기라 명명한다. 이러한 구조를 그림 1에 나타냈다. 초기 파라미터 연산기는 중간 홀로그램의 한 행의 가장 처음 위치(d=0)에서만 동작된다. 이 후의 위치(d>0)에서는 업데이트 연산기와 위상 연산기가 동작한다. 따라서 파이프라인도 업데이트 연산기와 위상 연산기에만 적용된다.

그림 2는 그림 1의 각 블록을 상세하게 나타낸 것이다. 표 1의 동작을 순서대로 수행할 수 있는 구조를 구성하였다. 그림 2에 각 중간 연산 단계를 표시하여 CGH 알고리즘이 하드웨어로 사상되는 방법을 확인할 수 있도록 하였다.

식 (8)과 (10)을 살펴보면 일부 동작을 LUT (look-up table)로 미리 만들어서 연산과정을 간략화할 수 있으므로 알 수 있다. 식 (8)로부터 가장 먼저 알 수 있는 것은 코사인 함수에 대한 연산을 LUT로 만드는 것이다. 이는 일반적으로 신호처리 분야에서 사용하는 방법이다. 또한 z_j 와 Δ 를 LUT로 만들 수 있음을 알 수 있다. z_j 는 0.5에서 1.5 사이의 값을 가지는 깊이 정보로써 최대 256가지를 가

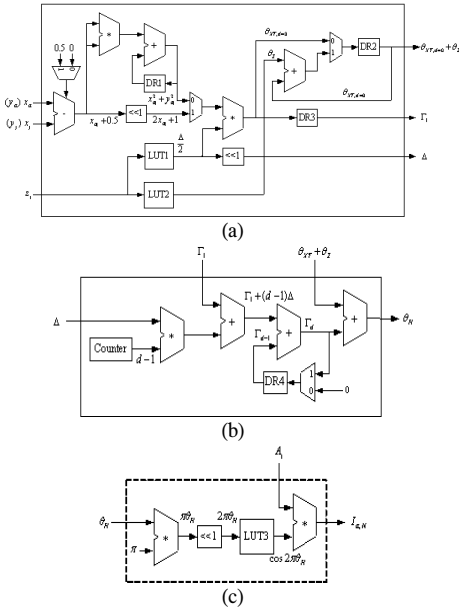


그림 2. CGH 셀의 세부 구조 (a) 초기 파라미터 연산기 (b) 업데이트 연산기 (c) 위상 연산기
 Fig. 2. Detail architecture of CGH cell (a) initial parameter calculator (b) update calculator (c) phase calculator

진다. 이들 값은 256개의 주소를 가지는 LUT로 미리 만든 후 연산을 하지 않고 결과를 출력한다. 만일 z_j 가 너무 많은 종류의 값을 가진다면 LUT로 만드는 것이 더욱 하드웨어 자원과 효율에 있어서 단점을 가져올 것이다.

3.3 파이프라이닝

하드웨어 내의 조합회로의 지연시간을 고려하고 임계경로를 분석하여 표 2와 같이 CGH 셀의 내부를 단위 연산으로 분리하였다. 총 7개의 단계를 나누는 것을 볼 수 있다.

표 2에서 분석한 CGH 셀의 내부 단위 연산시간을 바탕으로 그림 3과 같이 CGH 셀을 파이프라인화하였다. 총 7 단계의 파이프라인 단계를 가지므로 7 클럭의 대기지연시간 이후에 한 클럭 당 하나의 결과를 출력할 수 있다. 그림에서 R이 파이프라인 레지스터에 해당한다. 앞서 설명한 것과 같이 일반적으로 홀로그램을 재생할 경우에 홀로그램과 3차원 객체의 초기 혹은 오프셋 위상($\phi_\alpha + \phi_j$)은 사용하지 않는다. 파이프라인을 적용하는데 있어서 누적

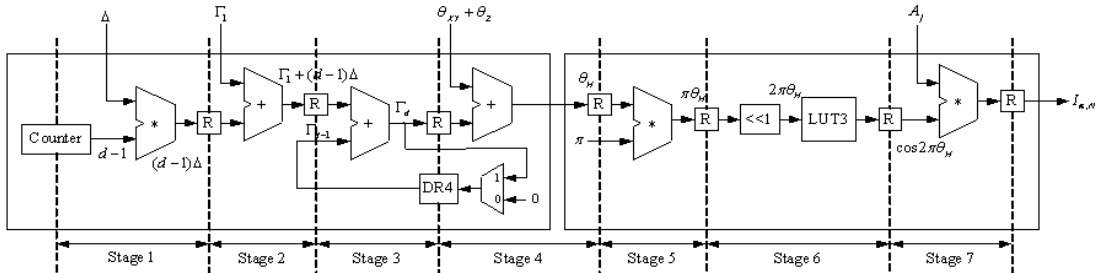


그림 3. 파이프라인을 적용한 CGH 셀의 구조
 Fig. 3. Pipelined architecture of CGH cell

표 2. 파이프라인 단계 스케줄링
 Table 2. Pipeline stage scheduling

Cycle	Step4			Step5	Step6		Step7
	Stage1	Stage2	Stage3	Stage4	Stage5	Stage6	Stage7
1	0Δ						
2	1Δ	$\Gamma_1 + 0\Delta$					
3	2Δ	$\Gamma_1 + 1\Delta$	Γ_1				
4	3Δ	$\Gamma_1 + 2\Delta$	Γ_2	$\theta_{H,d=1}$			
5	4Δ	$\Gamma_1 + 3\Delta$	Γ_3	$\theta_{H,d=2}$	$\pi\theta_{H,d=1}$		
6	5Δ	$\Gamma_1 + 4\Delta$	Γ_4	$\theta_{H,d=3}$	$\pi\theta_{H,d=2}$	$\cos(2\pi\theta_{H,d=1})$	
7	6Δ	$\Gamma_1 + 5\Delta$	Γ_5	$\theta_{H,d=4}$	$\pi\theta_{H,d=3}$	$\cos(2\pi\theta_{H,d=2})$	$\cos(2\pi\theta_{H,d=1})$
8	7Δ	$\Gamma_1 + 6\Delta$	Γ_6	$\theta_{H,d=5}$	$\pi\theta_{H,d=4}$	$\cos(2\pi\theta_{H,d=3})$	$\cos(2\pi\theta_{H,d=3})$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	$(N-1)\Delta$	$\Gamma_1 + N\Delta$	Γ_N	$\theta_{H,d=N-1}$	$\pi\theta_{H,d=N-2}$	$\cos(2\pi\theta_{H,d=N-3})$	$\cos(2\pi\theta_{H,d=N-3})$

덧셈이나 데이터를 보유하는 동작이 파이프라인 경로에 의해서 흐트러지지 않도록 유의해야 한다.

IV. 구현결과

제안한 하드웨어 구조는 VHDL을 이용하여 설계하였다. 구현에 이용된 실험 환경 및 변수들의 값을 표 3에 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 60%의 CLB를 사용한다. 각 FPGA 당 하나의 SDRAM이 필요하고 On-Chip DP SRAM은 약 500 Kbyte를 사용한다.

표 3에서 Hologram Size, Pixel Pitch, 그리고 Wavelength 등의 값은 임의로 정한 것이 아니고 본

연구팀이 보유한 디지털 홀로그램 재생 장치의 사양이 그와 같기 때문이다.

표 4에는 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 클럭 당 1개의 결과를 출력할 수 있다.

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 7.3%의 성능 향상이 이루어진 것을 볼 수 있다. 또한 실시간 처리를 하기위해 CGH 셀을 5,632개 사용할 경우에 제안한 하드웨어는 62.9개의 홀로그램을 생

표 3. 실험 환경 및 변수

Table 3. 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 4. 구현결과 및 이전 연구와의 비교

Table 4. 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

성할 수 있는 반면에 [4]는 41.10개의 홀로그램을 생성할 수 있다. 따라서 제안한 하드웨어가 더욱 좋은 성능을 보이는 것을 확인할 수 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 제안하였다. Xilinx의 XC2VP70 FPGA 칩 4개를 이용하였고, 각 FPGA 칩 당 320개의 CGH 커널이 맵핑되고 약 60%의 CLB를 사용하였다. 각 FPGA 당 하나의 SDRAM이 필요하고 On-Chip DP SRAM은 500 Kbyte를 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. 추후에 이들을 전용 블록으로 설계한 후에 ASIC화를 통해 범용성을 증대하고 면적을 최소화하고자 한다.

참 고 문 헌

[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 displayunit for three-dimensional display by use 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] T. Shimobaba, and T. Ito, "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.

서 영 호 (Young-Ho Seo)

종신회원



1999년 2월 광운대학교 전자재료공학과 졸업(공학사)
 2001년 2월 광운대학교 일반대학원 졸업(공학석사)
 2000년 3월~2001년 12월 인터스닷컴(주) 연구원
 2004년 8월 광운대학교 일반대학원 졸업(공학박사)
 2003년 6월~2004년 6월 한국전기연구원 연구원
 2004년 12월~2005년 8월 유한대학 연구교수
 2005년 9월~2008년 2월 한성대학교 조교수
 2008년 3월~현재 광운대학교 조교수
 <관심분야> 2D/3D 영상 및 비디오 처리, 디지털 홀로그램, SoC 설계, 워터마킹/암호화

최 현 준 (Hyun-Jun Choi)

정회원



2003년 2월 광운대학교 전자재료공학과(공학사)

2005년 2월 광운대학교 전자재료공학과(공학석사)

2005년 3월~현재 광운대학교 전자재료공학과 박사과정

<관심분야> Image Processing,

디지털 홀로그래, 디지털 콘텐츠 보안, FPGA/ASIC 설계

김 동 옥 (Dong-Wook Kim)

종신회원



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