

고속 홀로그램 생성 하드웨어를 위한 메모리 접근

이윤혁¹ · 박성호² · 서영호³ · 김동욱^{1*}

Memory Access for High-Performance Hologram Generation Hardware

Yoon-Hyuk Lee¹ · Sung-Ho Park² · Young-Ho Seo³ · Dong-Wook Kim^{1*}

¹Department of Electronic Material Engineering, Kwangwoon University, Seoul 139-701, Korea

²Uniquify Inc, San Jose, USA

³College of Liberal Arts, Kwangwoon University, Seoul 139-701, Korea

요 약

본 논문은 이전 논문들에서 제안한 고속 홀로그램 생성기 구조의 입출력을 분석하여 가상의 마스터(Virtual Master, VM)를 구현하여 홀로그램 생성기의 입출력 신호 패턴을 생성하고, 이를 이용하여 AXI(Advanced eXtensible Interface)기반의 시스템과 연동하여 메모리 접근에 대한 분석하였다. 또한 메모리에 맵핑방법을 통하여 메모리 접근 시 레이턴시를 줄이는 방법을 제안하고 구현한 시스템을 통하여 메모리 접근에 대하여 분석하였다. 제안한 메모리 맵핑 방법을 통하여 분석하였을 때 약 3배 가량 행 활성화(Activation)을 줄여 레이턴시를 줄일 수 있었다.

ABSTRACT

In this paper we analysis for in out signal by previous study and implement virtual master that generate CGH processor signals. Also, we propose memory address mapping. By constructing the system model of our method and by analyzing the latencies according to the memory access methods in a system including our model and several other models, the low-latency memory access method has been obtained. The proposed method is reduce number of activation in DRAM.

키워드 : CGH, SoC, AXI 기반 시스템, 메모리접근

Key word : CGH, SoC, AXI-based System, Memroy Access

접수일자 : 2013. 12. 31 심사완료일자 : 2014. 01. 13 게재확정일자 : 2014. 01. 27

* **Corresponding Author** Dong-Wook Kim(dwkim@kw.ac.kr, Tel:+82-2-940-5167)

Department of Electronic Material Engineering, Kwangwoon University, Seoul 139-701, Korea

Open Access <http://dx.doi.org/10.6109/jkice.2014.18.2.335>

print ISSN: 2234-4772 online ISSN: 2288-4165

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright © The Korea Institute of Information and Communication Engineering.

I. 서 론

홀로그래피는 1948년 Gabor에 의해 최초로 제안된 이후 3차원 정보를 기록할 수 있다는 특징 때문에 많은 연구자들의 관심을 끌어들였다. 기존의 홀로그래피는 홀로그램 필름에 3차원 정보를 기록하고, 현상된 필름을 통해 3차원 물체를 복원하는 방식을 사용함으로써 그 응용이 크게 제한되었다. 이러한 단점을 극복하기 위한 새로운 접근방법으로 1966년 이후 많은 연구자들이 컴퓨터 연산에 의한 홀로그램(computer-generated hologram, CGH)의 제작을 연구해 오고 있다[1]. 이 기술은 물체파(object wave)와 기준파(reference wave)의 간섭에 의해 생성되는 간섭향을 계산함으로써 현실에서는 불가능한 이상적인 특성을 가진 부품을 제작하거나 특성시험 등을 위해 개발되었다[2-3].

CGH를 이용하여 한 프레임에 해당하는 홀로그램을 생성하기 위해서는 많은 연산량과 시간이 소요되기 때문에 고속의 연산방법이 필요하다. 이런 고속 CGH를 위한 여러 알고리즘들이 개발되어 왔다[4-6]. MIT Media lab의 Spatial Imaging Group(지금은 Object-based Media Group)은 고속 CGH를 위한 연구를 가장 먼저 시작한 연구그룹이다[4]. 여기서는 HPO(horizontal-parallax-only) 방식의 CGH 기법을 이용해 디지털 홀로그램을 생성하는 연구를 수행하였다. 이 연구에서는 LUT(Look-up Table)방식과 parallel supercomputer를 사용해 10,000 point의 light source 영상으로부터 1초에 한 장씩 디지털 홀로그램(해상도: 6M)을 생성하였다. 일본 Nihon대학의 Yoshikawa교수는 root연산을 Taylor 급수로 전개하여 변형한 후 CGH 수식의 근사화를 통해 고속화 알고리즘을 정리하였다[5]. 이 연구는 최근 연구되고 있는 고속 CGH 알고리즘들의 이론적 토대가 된다는 점에서는 상당히 가치가 있지만, 연산속도 측면에서는 큰 의미가 없다. 일본의 Chiba 대학의 연구팀은 Yoshikawa교수의 이론을 변형시켜 x축에서 반복덧셈만을 수행하여 CGH를 계산하는 알고리즘을 제안하고 FPGA를 이용한 하드웨어로 구현하였다[6].

이와 같이 CGH 연산량이 너무나 방대하기 때문에 실제로 소프트웨어로 CGH를 실시간으로 처리하는 것은 불가능하고 하드웨어로 구현되어야 하며, 지금까지 CGH를 위해 하드웨어를 활용한 많은 연구가 진행되어 왔다[7-14]. 이러한 연구들은 GPU(general-purpose

processing unit) 기반의 소프트웨어 방식[7-11]과 FPGA(field programmable gate array) 기반의 하드웨어 칩 구현 방식[6],[12-14]으로 나뉘어진다. GPU를 이용한 방식은 FPGA 기반의 방식에 비해서 구현이 비교적 쉽고 개발기간이 짧다는 장점이 있다. FPGA를 이용하면 구현과정이 매우 복잡하고 개발기간이 오래 걸린다는 단점이 있다. 또한 한번 구현하면 구조를 변경하거나 성능을 개선하기 어렵다. 그러나 GPU 방식에 비해서 성능은 수십에서 수백 배 가량 높은 성능을 갖는다. 특히 GPU에 대한 연구가 최근에 활발히 이루어지고 있다. 싱가포르대[9]는 CGH 수식을 지수함수를 이용한 복소수 형태로 변환한 후에 연산을 분리하는 알고리즘을 제안하였다. 분리된 항을 각각 LUT로 만든 후에 연산을 고속화시켰고, 이를 nVidia의 GPU로 구현하였다. 1,000 (1K)개의 object point를 갖는 객체에 대해 1024×768크기의 홀로그램을 0.3초당 한 장씩 생성할 수 있었다. 중국 Zhongshan 대학의 Wang[10]은 3D mesh-model을 기반으로 GPU를 이용해 CGH를 수행한 연구를 발표했다. 또한 일본 Chiba대의 Shimobaba[11]는 AMD의 GPU(Graphic Processing Unit)를 기반으로 하여 이전 연구에서 제안한 알고리즘[6]을 사용하면서 GPU 프로그래밍 기법을 활용하여 고속화하였고, HD크기의 홀로그램을 0.31초당 한 장씩 생성할 수 있었다. 또한 4개의 Xilinx FPGA (XC2VP70)를 사용하는 전용 PCB 보드를 제작하여 Fresnel Transform CGH를 구현하였다[12]. 홀로그램의 x축 해상도만큼의 단위 연산기를 병렬로 배열(1,408개)하는 구조를 가지고 166MHz의 클럭 주파수에서 한 프레임의 홀로그램을 0.0679초에 생성할 수 있다. 최근에는 CGH를 연산하기 위한 전용 연산 시스템인 HORN-6 특수 컴퓨터가 제안되기도 하였다[13]. 또한 100% 파이프라인(pipeline) 구조를 기반으로 하는 CGH 프로세서가 제안되었다[14]. Fresnel 변환을 수행하기 위한 CGH 셀(cell)의 하드웨어 구조를 제안한 후에 이를 확장하여 CGH 커널(kernel)을 구성하였고, 이를 다시 확장하여 CGH 프로세서를 구현하였다. 파이프라인 구조[14]의 하드웨어는 이전 하드웨어[12]보다 최대 87.32%의 높은 성능을 갖는다.

이전 논문[12][14-16]에서 사용하고 있는 홀로그램의 최대 크기는 1408×1050이다. 픽셀 크기가 10 μ m의 SLM(Spatial Light Modulator)을 사용한다면 홀로그램의 크기는 1.408×1.050cm²이다. 광학적인 확대를 고려

한다고 해도 너무나 작은 홀로그램이기 때문에 CGH 연산을 위해 지금보다 훨씬 높은 성능을 갖는 하드웨어가 필요하다. CGH 프로세서는 연산량 뿐만 아니라 외부 메모리로의 접근 량이 매우 많다. 이는 AP (application processor)와 같은 여러 기능을 수행하는 프로세서와 같이 사용할 경우 데이터 전송으로 인한 전체 시스템 성능을 감소시키는 영향을 줄 수 있다. 따라서 CGH 프로세서와 같은 데이터 전송이 많이 이루어지는 프로세서의 경우 데이터 전송을 분석해야할 필요가 있다.

많은 프로세서간의 데이터 전송을 위한 구조는 대표적으로 버스구조[17-26]와 NoC (Network on Chip) [27-32]가 있다. 버스구조를 여러 레이어로 분리하거나 [17-18] 데이터 전송간의 arbiter를 개선하고[21] 프로세서의 배치[22-26]를 통하여 데이터 전송간의 지연시간을 줄일 수 있는 연구가 이루어져 왔다. 또한 AXI (Advanced eXtensible Interface) 프로토콜 버스의 데이터 통신량을 분석할 수 있는 구조를 구현하였다[20].

II. 컴퓨터 생성 홀로그램 하드웨어 구조

2.1. CGH 알고리즘

컴퓨터 생성 홀로그램은 광학계를 수학적으로 모델링하여 구한 수식을 통하여 홀로그램을 생성한다. 본 논문에서 사용하는 CGH는 여러 CGH 중 위상을 이용한 방식을 사용하였다. 파면의 파장이 광원과의 거리와는 독립적으로 일정하게 유지되는 가간섭성(coherent) 광원 참조파가 어떠한 물체에 조사될 때 반사된 물체파는 물체의 표면의 형태에 따라 파면이 변화한다. 이때 참조파와 물체파간의 간섭현상을 통하여 물체에 대한 정보를 계산할 수 있다[3]. 식 (1)은 CCD (charge-coupled device)로 입사되는 두 파의 간섭 현상을 기록하는 홀로그램의 하나의 화소의 밝기(I)를 구하는 CGH 식이다. N 개의 물체파의 광원(A_j)에 대한 각 좌표는 (x, y, z) 이고 각 홀로그램 화소의 좌표는 (u, v) 이다. k 는 참조파의 파수($2\pi/\lambda$)이고 λ 는 참조파의 파장이다. 또한 p 는 CCD의 한 화소의 크기이다.

$$I(u, v) = \sum_j^N A_j(x, y, z) \exp\{jk\sqrt{(pu - px)^2 + (pv - py)^2 + z^2}\} \quad (1)$$

식 (1)에서 $(pu - px_j), (pv - py_j) \ll z_j$ 의 조건이 만족하는 Fresnel 영역일 경우 식 (2)와 같이 Taylor 급수의 3차 항 이후를 0으로 근사화하여 구할 수 있다.

$$A_j \exp\left[j\frac{2\pi}{\lambda}\sqrt{p^2(u-x)^2 + p^2(v-y)^2 + z^2}\right] \approx A_j \exp\left[j2\pi\left\{\frac{z}{\lambda} + \frac{p^2}{2\lambda z}\{(u-x)^2 + (v-y)^2\}\right\}\right] \quad (2)$$

식 (1)과 (2)를 통하여 구한 복소수항의 홀로그램을 오일러 공식을 이용하여 전개하여 실수부만을 취하면 식 (3)을 구할 수 있다. $\frac{p^2}{2z\lambda}(u-x)^2$ 과 $\frac{p^2}{2z\lambda}(v-y)^2$ 그리고 $\frac{z}{\lambda}$ 를 각각 $\theta_x, \theta_y, \theta_z$ 로 하여 각 좌표에 해당하는 좌표 항으로 나타낼 수 있다[3].

$$\begin{aligned} RE[I(u, v)] &= \sum_j^N A_j(x, y, z) \cos\left[2\pi\left\{\frac{z}{\lambda} + \frac{p^2}{2\lambda z}\{(u-x)^2 + (v-y)^2\}\right\}\right] \\ &= \sum_j^N A_j(x, y, z) \cos[2\pi\{\theta_z + \theta_x + \theta_y\}] \end{aligned} \quad (3)$$

2.2. 병렬화 방법

식 (3)의 CGH를 통한 홀로그램을 구하기 위해서 크게 두 가지의 연산방법이 존재한다. 그림 1에 두 가지 방법을 나타냈다. 그림 1(a)는 물체점 기반의 연산 방법으로 하나의 광원에 대하여 모든 홀로그램 화소를 연산하여 이를 누적하는 형태의 연산 방법이고, 그림 1(b)는 홀로그램 화소 기반의 연산 방법으로 하나의 홀로그램 화소를 모든 광원에 대하여 연산하여 누적하는 방법이다. 두 방법의 연산량은 같지만 연산을 위한 메모리 접근량에 큰 차이가 생긴다. N 개의 물체점을 이용하여 $U \times V$ 의 홀로그램을 생성하고, 한 번에 생성할 수 있는 홀로그램 생성기의 병렬 셀의 개수가 $U' \times V'$ 일 경우 총 연산 횟수 I 는 식 (4)와 같다.

$$I = N(\lceil U/U' \rceil \times \lceil V/V' \rceil) \quad (4)$$

그림 1(a)와 같은 물체점 기반의 연산 방식은 하나의 물체점에 대하여 연산 후 다음 물체점에 대하여 연산하

기 위해 이전의 물체점에 대하여 연산한 홀로그램을 메모리로부터 읽어와야 하고, 누적 결과를 메모리에 저장해야 한다.

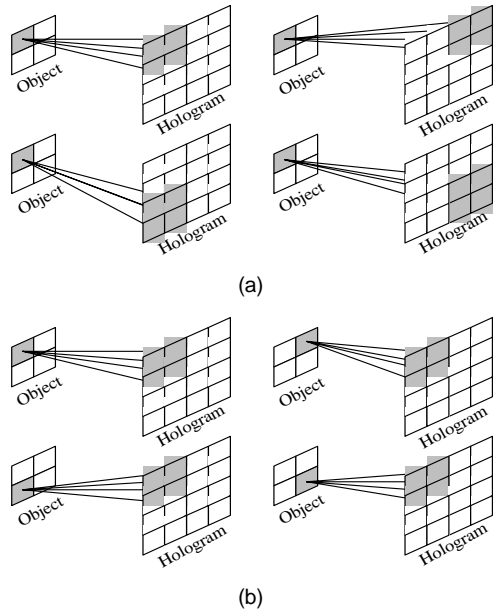


그림 1. 홀로그램 생성 방법 (a) 물체점 기반, (b)홀로그램 화소 기반
Fig. 1 Calculation scheme (a) object point-based, (b) Hologram pixel-based

즉, 하나의 물체점에 대해 메모리 접근은 병렬 셀당 2번이 필요하다. 또한 N 개의 물체점을 로드하기 위하여 N 번의 물체점에 대한 메모리 접근이 필요하다. 따라서 식 (4)로부터 한 장의 홀로그램을 생성하기 위한 총 메모리 접근량 M 은 식 (5)와 같이 나타낼 수 있다.

$$M = 2IU'V' + N \approx (2UV + 1)N \quad (5)$$

반면 그림 1(b)와 같이 홀로그램 화소 기반의 연산 방식은 홀로그램에 대하여 모든 물체점에 대하여 연산하기 때문에 누적 연산을 위한 메모리 접근이 필요 없다. 한 번의 연산에 필요한 메모리 접근 횟수는 물체점을 로드하기 위한 한 번의 접근이 필요하고, 모든 물체점에 대하여 연산할 때 마다 병렬 셀만큼 접근이 필요하다. 따라서 한 장의 홀로그램을 생성하기 위한 총 메모리 접근량 M 은 식 (6)과 같이 나타낼 수 있다.

$$M = I + U'V'(\lceil U/U' \rceil \times \lceil V/V' \rceil) \approx I + UV \quad (6)$$

그림 2는 연산 방법에 따른 메모리 접근의 타이밍도를 나타내었다. 그림 2(a)는 물체점 기반의 연산 방법으로 한 번의 연산을 위해 병렬 셀의 2배 만큼의 메모리 접근이 필요하다. 만약 파이프라인화된 프로세서를 이

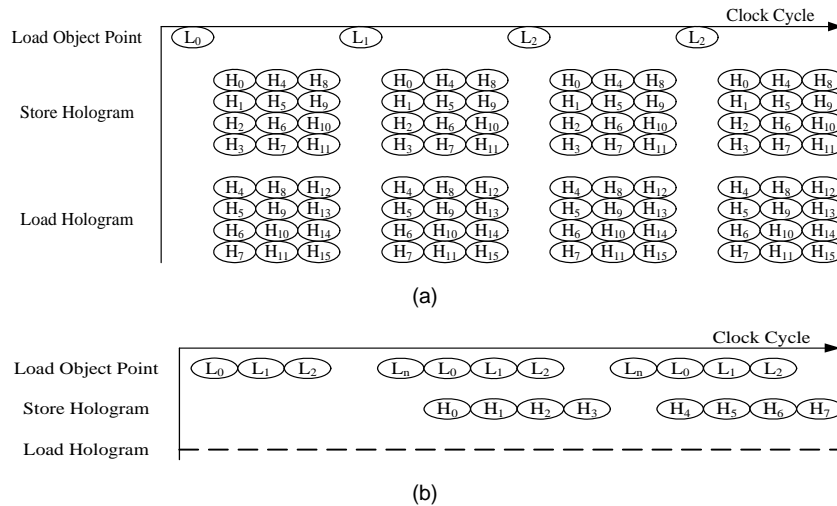


그림 2. 연산방법에 따른 타이밍도 (a) 물체점 기반, (b) 홀로그램 화소 기반
Fig. 2 Timing diagram (a) object point-based, (b) hologram pixel-based

용하여 연산을 할 경우 클럭 당 병렬 셀의 두 배의 메모리 접근이 필요하다. 그림 2(b)는 홀로그래프 화소 기반의 연산 방법으로 N개의 물체점을 연산하는 동안 병렬 셀의 개수만큼과 N번의 메모리 접근이 필요하다.

III. 메모리 접근 분석을 위한 시스템 모델

SoC 칩 내에는 많은 IP들과 IP들 간 통신을 위한 회로 등이 존재한다. 하지만 모든 IP를 구현하는 것은 매우 어려운 일이다. 따라서 본 장에서는 본 연구팀에서 구현한 IP들을 대신하여 가상의 마스터가 배치된 AXI (advanced extensible interface) 프로토콜을 지원하는 시스템 모델을 제안한다.

3.1. AXI 프로토콜

제안한 시스템 모델은 AXI 프로토콜을 기반으로 구성하였다. AXI 프로토콜은 기본적으로 5개의 독립적인 채널로 구성되어 있으며 쓰기를 위한 어드레스 채널과 데이터 채널 그리고 응답 채널로 구성되고, 읽기를 위한 어드레스 채널과 데이터 채널로 구성한다. 모든 채널은 ready 신호와 valid 신호의 핸드-셰이킹을 통하여 데이터 전송이 이루어지는데, 그림 3과 같이 이 두 신호가 '1'일 경우 다음 클럭에서 해당 정보를 캡처한다.

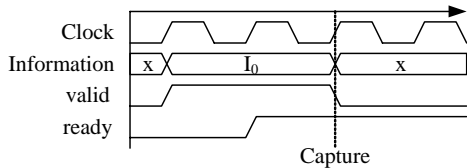


그림 3. 핸드-셰이킹
Fig. 3 Hand-shaking

또한 각 채널은 해당 정보(데이터)의 고유 ID가 있는데 각 채널에 동일한 ID는 같은 정보를 가진다. 즉, 데이터 채널에서 이전의 주소 채널의 ID와 같은 ID를 전송받을 경우 해당 주소에 데이터를 쓰기 혹은 읽기 동작을 수행한다. 따라서 같은 ID를 통해 여러 주소를 전송할 경우 데이터는 전송한 주소의 순서대로 전송된다. 반대로 서로 다른 ID를 통하여 주소를 전송할 경우

순서에 상관없이 데이터가 전송될 수 있다. 그림 4는 비순차 방식의 데이터 전송을 나타내고 있는데, 비순차 방식을 통하여 지연시간이 긴 데이터 사이에 길이가 짧은 데이터를 보내어 전체적인 지연시간을 줄일 수 있다.

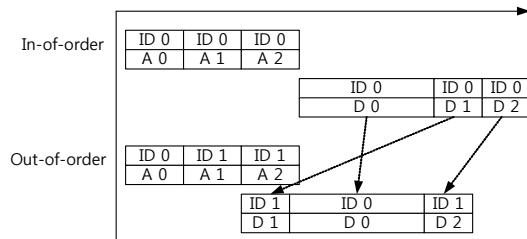


그림 4. 비순차 전송
Fig. 4 Out-of-order transfer

AXI 프로토콜의 컨트롤 신호(Burst, Length, Size)는 어드레스 채널과 함께 전송된다. AXI 프로토콜은 Burst 형태의 정보를 전송할 수 있다. Burst 모드는 하나의 어드레스 정보를 이용하여 여러 개의 데이터를 전송하는 것이다. 이때 Burst 신호의 형태의 따라 고정된 어드레스, 혹은 하나씩 증가 또는 일정 구간 내에서만 증가하는 전송이 가능하다. Length 신호의 경우 Burst 전송을 할 경우 전송데이터의 개수를 설정할 수 있다. 이때 슬레이브는 Length 신호를 확인하여 해당하는 개수만큼 데이터를 전송 받게 된다.

Size 신호는 하나의 Burst 내의 유효한 데이터의 크기이다. 쓰기 데이터 채널에서는 해당 데이터와 STRB 신호 및 Last 신호가 존재하는데 STRB 신호는 데이터에서 유효한 데이터에 마스킹하고, Last 신호는 해당 데이터가 Burst 전송의 마지막 신호임을 나타낸다. 응답 채널은 슬레이브에서 마스터로 보내는 신호로 입력 어드레스채널에 해당하는 데이터채널이 전송결과를 출력한다. 읽기 데이터 채널은 슬레이브에서 마스터로 보내는 신호로 쓰기 데이터 채널의 STRB 신호 대신 응답 신호를 출력한다.

3.2. 시스템 모델

제안한 시스템 모델 System Verilog로 구현 하였으며 여러 개의 가상의 마스터를 가지는 모델로 슬레이브는 DRAM 컨트롤러로 구성하였다. 구현한 시스템 모델

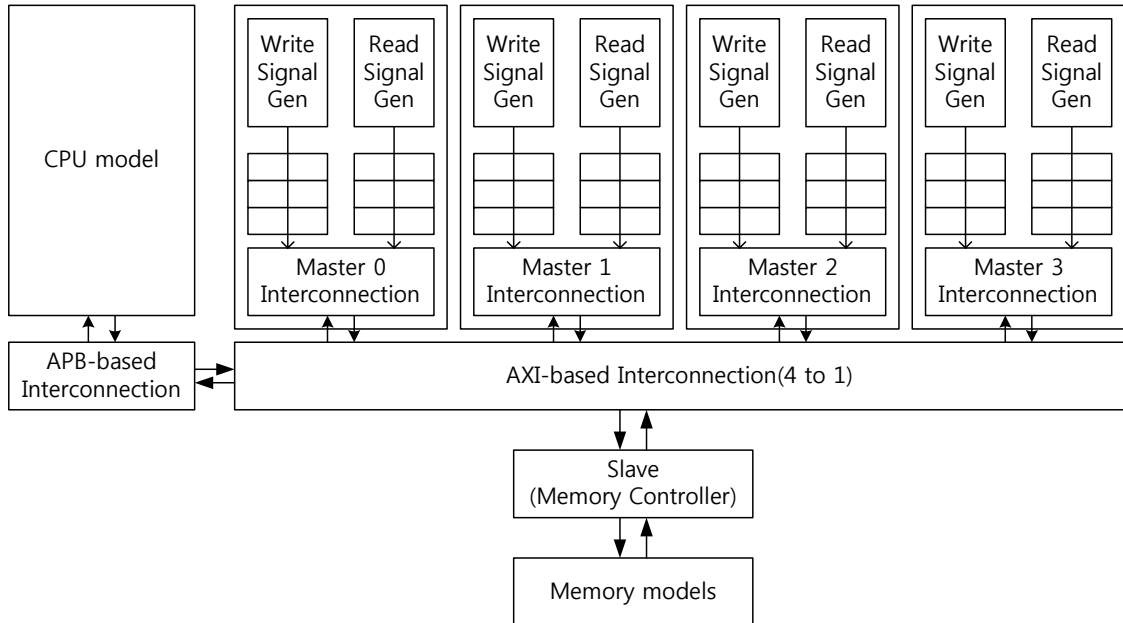


그림 5. 제안한 시스템 모델
 Fig. 5 The proposed system model

은 AXI 프로토콜 기반의 시스템으로 전체적인 구조를 그림 5에 나타내었다. 쓰기/읽기 신호 생성기에서 각 채널(쓰기: 어드레스, 데이터, 응답, 읽기: 어드레스, 데이터)에 해당하는 신호를 생성하여 임시 저장 공간인 큐(queue)에 입력한다.

시스템의 초기화 이후 마스터 상호접속 모듈에서 큐에서 데이터를 읽어서 AXI 프로토콜을 통하여 데이터를 디코딩하여 전송한다. 상호접속 모듈에서 슬레이브에서 오는 ready 혹은 valid 신호를 통하여 데이터 전송이 이루어지면 다음 신호를 큐로부터 읽어 디코딩 한다. 큐에 모든 데이터가 마스터 상호접속 모듈을 통하여 전송이 완료되면 해당 마스터의 전송은 중단된다.

각 신호 생성기는 단일 동작에 대한 AXI 프로토콜의 신호를 생성하고 동작신호와 비동작신호를 생성한다.

동작 신호는 입력된 데이터와 주소 및 제어 신호를 큐로 입력하고 비동작 신호 생성은 모든 신호를 '0'으로 하고 ready 혹은 valid 신호를 '0'으로 하여 마스터의 전송이 없을 경우에 해당한다. 다양한 신호 생성을 위하여 각 채널의 큐로 입력되는 데이터와 상호접속 모듈을 통하여 전송된 데이터를 카운트를 하여 이 값을 통하여

여러 조건의 신호 생성을 가능하도록 하였다.

IV. 메모리 접근 방법

홀로그램 생성기와 같이 메모리 접근 횟수가 많은 IP의 경우 접근 횟수뿐만 아니라 접근 방법에 따라 메모리 접근 지연시간을 줄여 전체 시스템의 성능을 높일 수 있다. 특히 멀티미디어 SoC와 같이 많은 기능을 수행하는 칩의 경우에 메모리 접근은 매우 중요한 이슈이다. 따라서 본 장에서는 앞장에서 설명한 시스템 모델을 이용하여 홀로그램 생성기가 다른 IP들과 같이 혹은 여러 홀로그램 생성기가 같이 구동되는 조건에 대한 메모리 접근 방법에 대하여 설명한다.

4.1. DRAM 구조

그림 6과 같이 DRAM은 어떠한 메모리 지점에 접근하기 위해서 해당 행, 해당 행에 접근하여 그 행의 모든 데이터를 활성화시킨다. DRAM 내의 한 비트를 저장하는 셀은 캐패시터의 충전 혹은 방전을 통해 데

이터를 저장한다. 이때 충/방전에 필요한 시간을 줄이고 셀의 집적도를 높이기 위해 커패시터의 용량을 작게 만드는데, 메모리에 접근할 때 이 커패시터의 용량이 작아서 저장된 데이터의 레벨을 확인하는 것이 어렵다. 따라서 레벨을 확인하기 위해서 증폭과정이 필요한데, 이 과정이 activation 과정이다. activation 과정은 해당 बैं크에서 하나의 행에 해당하는 데이터를 증폭 시키는데, 이 과정이 끝나야 해당 열에 접근할 수 있다.

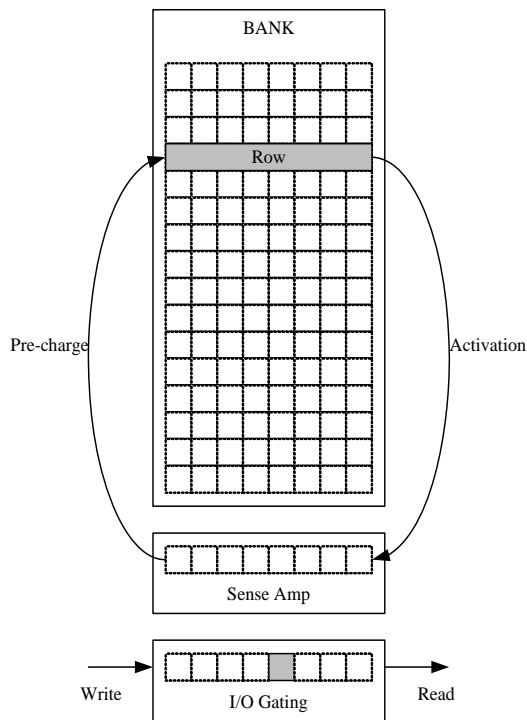


그림 6. DRAM의 구조
Fig. 6 DRAM architecture

또한 증폭 되어져 있는 행에 접근하여 읽기 혹은 쓰기 동작으로 변형된 데이터는 बैं크 내의 데이터와 다른 데이터이므로 증폭된 행을 반드시 다시 बैं크로 업데이트하는 과정이 필요하다. 이 과정이 pre-charge 과정이고 어떤 행에서 다른 행으로 접근이 필요할 경우 이 과정을 수행해야 한다. 따라서 DRAM에 접근할 때 많은 지연시간을 갖는 것은 activation과 pre-charge 과정 때문이므로, 한 번의 행을 activation 시키고 많은 읽

기/쓰기 동작을 수행하는 것이 지연시간을 줄일 수 있는 방법이다. 또한 DRAM의 각 बैं크는 독립적으로 위 과정을 수행할 수 있으며 다른 बैं크의 다른 행을 접근할 경우에는 원래의 자리로의 업데이트 없이 접근이 가능하다.

4.2. 메모리 접근 방법

앞서 설명한 것과 같이 하나의 행에 대하여 많은 접근을 하는 것이 유리하다. 하지만 2D기반의 홀로그래프 생성기의 경우 홀로그래프를 저장할 때 작은 블록을 연산하여 데이터를 저장하기 때문에 작은 블록을 그대로 저장할 경우 많은 행에 대하여 접근한다. 이 경우 많은 Activation 과정이 필요하기 때문에 메모리 접근할 때 지연시간이 증가한다. 그림 7은 DRAM의 접근 량이 같을 경우 접근 지연시간을 줄이기 위해 2D 형태의 영상 데이터를 1D 형태로 주소를 변형하는 방법을 나타내었다. 2D 형태의 총 데이터 량이 DRAM의 한 행의 데이터 량보다 작을 경우 영상 데이터의 행과 열의 상위 비트를 치환하여 한 행에 2D 블록 전체를 저장할 수 있다.

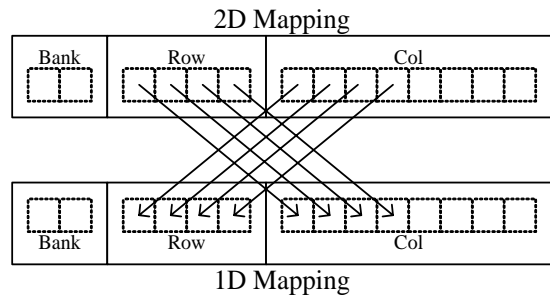


그림 7. 2D에서 1D로의 메모리 맵핑
Fig. 7 Memory address mapping from 2D to 1D

홀로그래프 생성기에서 광원의 데이터 형태는 밝기가 '0'이 아닌 유효한 광원에 대하여 연산한다. 따라서 2D 형태가 아닌 1D형태의 데이터로 구성된다. 이 경우 여러 बैं크에 대하여 접근할 경우 Activation 과정의 지연시간은 독립적으로 수행되므로 숨길 수 있다. 또한 접근할 बैं크의 수만큼 행의 길이가 길어지는 효과를 얻을 수 있어 지연시간을 줄일 수 있다.

V. 메모리 접근 방법

그림 8은 홀로그램 화소값을 저장하기 위한 메모리를 접근할 때 서브블록 행이 메모리의 행주소로 맵핑할 경우와 블록전체를 메모리 주소의 행으로 맵핑하여 저장할 때 512x512 홀로그램을 생성할 때 블록의 크기별로 activation 동작을 측정한 결과이다. 홀로그램 화소값을 저장할 때는 블록 전체를 메모리 주소의 행으로 하여 저장하면 activation 동작이 약 1/3로 줄어드는 것을 확인 할 수 있다. 이는 적은 activation 동작으로 같은 량의 데이터를 저장하는 것이므로 블록단위로 DRAM의 행 주소를 설정하는 것이 유리하다는 것을 확인 할 수 있다.

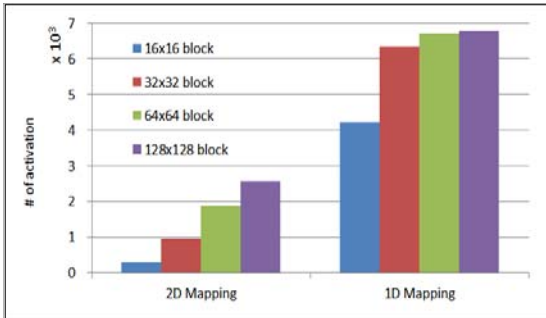


그림 8. 홀로그램 화소값을 저장하기 위한 Activation 수
 Fig. 8 Number of activations in storing hologram pixel values

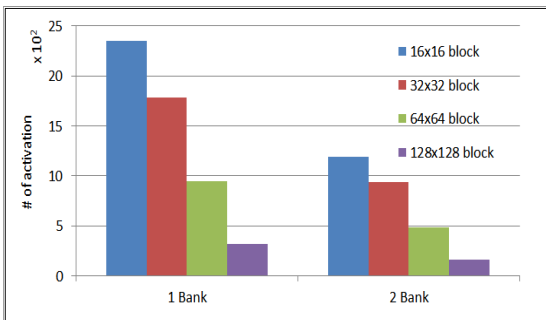


그림 9. 광원 데이터를 로드하기 위한 Activation 수
 Fig. 9 Number of activations in loading light source data

그림 9는 광원 데이터를 읽어 오기 위한 메모리 접근 시 블록의 크기별로 광원에 할당된 बैं크를 1개와 2개일

경우 बैं크 당 평균 activation을 측정한 결과이다. बैं크를 많이 사용할수록 activation 동작이 1/2 정도로 줄어드는 것을 확인 할 수 있다. 이는 사실상 총 activation 동작은 같으나 서로 다른 बैं크에서는 독립적으로 동작할 수 있으므로 बैं크를 여러 개 설정하는 것이 유리하다는 것을 확인할 수 있다.

VI. 결론

본 논문에서는 이전 연구들에서 구현한 고속 홀로그램 생성기의 신호 패턴을 분석하여 구현한 시스템 모델을 통하여 메모리 접근에 대하여 분석하였다. 구현한 시스템 모델은 AXI 프로토콜 기반의 시스템으로 분석한 고속 홀로그램 생성기의 신호를 큐(Queue)형태에 메모리에 저장한 뒤 AXI 프로토콜에 의한 전송하는 구조로 구현 하였다. 메모리 접근을 분석하기 위해 DRAM 내에서 하나의 행을 활성화(Activation)의 수를 CGH 프로세서의 신호 패턴을 이용하여 측정하였다. 홀로그램 화소를 저장하기 위한 메모리 주소를 1D형태로 맵핑할 때 Activation 수가 3배 감소할 수 있었고, 광원을 불러오기 위한 메모리 बैं크를 증가 시킬 경우 2배 감소하였다.

감사의 글

본 연구는 지식경제부, 방송통신위원회 및 한국 산업기술평가관리원의 산업원천기술개발사업(정보통신)의 일환으로 수행하였음. [KI002058, 대화형 디지털 홀로그램 통합비스 시스템의 구현을 위한 신호처리요소 기술 및 하드웨어 IP 개발]

REFERENCES

[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] Mark Lucente, "Interactive Computation of Holograms Using a Look-up Table", *Journal of Electronic Imaging*, vol. 2, #1, pp.28-34, Jan. 1993.
- [5] H. Yoshikawa, S. Iwase, and T. Oneda, "Fast Computation of Fresnel Holograms employing Differences", *Proceeding of SPIE*, vol. 3956, 2000.
- [6] T. Shimobaba, 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.
- [7] 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.
- [8] 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.
- [9] Y. Pan, X. Xu, S. Solanki, X. Liang, R. Bin, A. Tanjung, C. Tan, and T.-C. Chong, "Fast CGH computation using S-LUT on GPU", *Optics Express*, vol. 17, No. 21, pp. 18543-18555, Oct. 2009.
- [10] Y.-Z. Liu, J.-W. Dong, Y.-Y. Pu, B.-C. Chen, H.-X. He, and H.-Z. Wang, "High-speed full analytical holographic computations for true-life scenes", *Optics Express*, vol. 18, no. 4, pp. 3345-3351, Feb. 2010.
- [11] T. Shimobaba, T. Ito, N. Masuda, Y. Ichihashi, and N. Takada, "Fast calculation of computer-generated-hologram on AMD HD5000 series GPU and OpenCL", *Optics Express*, vol. 18, no. 10, pp. 9955-9960, May. 2010.
- [12] 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.
- [13] Y. Ichihashi, H. Nakayama, T. Ito, N. Masuda, T. Shimobaba, A. Shiraki, and T. Sugie, "HORN-6 special-purpose clustered computing system for electroholography", *Optics Express*, vol. 17, no. 16, pp. 13895-13903, Aug, 2009.
- [14] Y.-H. Seo, H.-J. Choi, J.-S. Yoo, and D.-W. Kim, "An architecture of a high-speed digital hologram generator based on FPGA", *Journal of Systems Architecture*, Vol. 56, pp. 27-37, Dec. 2009.
- [15] Y.-H. Seo, H.-J. Choi, J.-S. Yoo, and D.-W. Kim, "A New Parallelizing Algorithm and Cell-based Hardware Architecture for High-speed Generation of Digital Hologram", *Journal of Systems Architecture*, Vol. 16, pp. 54-63, Jan. 2011.
- [16] Y.-H. Seo, Y.-H. Lee, J.-S. Yoo and D.-W. Kim, "Hardware architecture of high-performance digital hologram generator on the basis of a pixel-by-pixel calculation scheme", *APPLIED OPTICS*, Vol.51 Issue 18, pp. 4003~4012, Jun. 2012.
- [17] Y. S. Chen, S.W. Lo, "Multi-layer bus minimization for SoC", *Journal of System and Software*, Vol.83, pp.121~132, July 2009.
- [18] R. Lu, A. Cao, C. K. Koh, "SAMBA-Bus: A High Performance Bus Architecture for System-on-Chip", *IEEE*, Vol.15, No.1, pp.69~79, Jan 2007.
- [19] K. K. Ryu, V.J. Mooney, "Automated Bus Generation for Multiprocessor SoC Design", *IEEE*, Vol.23, No.11, Nov 2004.
- [20] H. M. Kyung, G. H. Park, J. W. Kwak, T. J. Kim, S. B. Park, "Design and implementation of Performance Analysis Unit(PAU) for AXI-based multi-core System on Chip(SoC)", *Microprocessors and Microsystems*, Vol.34, pp.102~116, March 2010.
- [21] S. Kallakuri, A. Doboli, S. Doboli, "Applying stochastic modeling to bus arbitration for systems-on-chip", *Integration, the VLSI journal*, Vol.40, pp.183~191, Feb 2006.
- [22] J. H. Yoo, S. J. Yoo, K. Y. Choi, "Topology/Floorplan/Pipeline Co- Design of Cascaded Crossbar Bus", *IEEE*, Vol.17, No.8, pp.1034~1047, Aug 2009.
- [23] R. Wang, Y. Zhan, N. Chou, F. Y. Young, "Bus Matrix Synthesis Based on Steiner Graphs for Power Efficient System-on-Chip Communications", *IEEE*, Vol.30, No.2, pp.167~179, Feb 2011.
- [24] S. Pasricha, Y. H. Park, F. J. Kurdahi, N. Dutt, "CAPPS: A Framework for Power-Performance Tradeoffs in Bus-Matrix-Based On-Chip Communication Architecture Synthesis", *IEEE*, Vol.18, No.2, pp.209~221, Feb 2010.
- [25] W. Sheng, S. Dong, "Multi-bend bus-driven floorplanning considering fixed-outline constraints", *Integration, the VLSI journal*, Vol.46, pp.142~152, March 2012.
- [26] S. Pasricha, N. D. Dutt, E. Bozorgzadeh, M. B. Romdhane, "FABSYN: Floorplan-Aware Bus Architecture Synthesis", *IEEE*, Vol.14, No.3, pp.241~253, March 2006.

- [27] S. H. Lee, Y. C. Yoon, S. Y. Hwang, "Communication-aware task assignment algorithm for MPSoC using shared memory", *Journal of System Architecture*, Vol.56, pp.233~241, March 2010.
- [28] Y. C. Lan, H. A. Lin, S. H. Lo, Y. H. Hu, S. J. Chen, "A Bidirectional NoC(BiNoC) Architecture with Dynamic Self-Reconfigurable Channel", *IEEE*, Vol.30, No.3, March 2011.
- [29] C. Wang, W. H. Hu, S. E. Lee, N. Bagherzadeh, "Area and power-efficient innovative congestion-aware Network-on-Chip", *Journal of System Architecture*, Vol.57, pp.24~38, Oct 2010.
- [30] S. Jovanovic, C. Tanougast, C. Bobda, S. Weber, "CuNoC: A dynamic scalable communication structure for dynamically reconfigurable FPGAs", *Microprocessor and Microsystems*, Vol.33, pp.24~36, Aug 2008.
- [31] M. Bakhouya, S. Suboh, J. Gaber, T. El-Ghazawi, S. Niar, "Performance evaluation and design tradeoffs of on-chip interconnect architecture", *Simulation Modeling Practice and Theory*, Vol.19, pp.1496~1505, Nov 2010.
- [32] A. Abousamra, A. K. Jones, R. Melhem, "Codesign of NoC and Cache Organization for Reducing Access Latency in Chip Multiprocessors", *IEEE*, Vol.23, No.6 pp.1038~1046, June 2012.



이윤혁(Yoon-Hyuk Lee)

2012 광운대학교 전자재료 공학과 공학사 졸업
 2014 광운대학교 일반대학원 공학석사 졸업
 ※관심분야 : 디지털 홀로그래프, SoC 설계



박성호(Sung-Ho Park)

2000 광운대학교 전자재료공학과 공학사 졸업
 2004 광운대학교 일반대학원 공학석사 졸업
 2004 ~ 2013 LG전자 SIC 사업팀 HPM Gr 선임연구원
 2013 ~ 현재 Uniquify 책임연구원
 ※관심분야 : 영상압축, 워터마킹, 암호화, FPGA/ASIC 설계, Design Methodology



서영호(Young-Ho Seo)

1999 광운대학교 전자재료 공학과 공학사 졸업
 2001 광운대학교 일반대학원 공학석사 졸업
 2004 광운대학교 일반대학원 공학박사 졸업
 2003 ~ 2004 한국전기연구원 연구원
 2005 ~ 2008 한성대학교 조교수
 2008 ~ 현재 광운대학교 교양학부 조교수
 ※관심분야 : 실감미디어, 2D/3D 영상신호처리, 디지털 홀로그래프



김동욱(Dong-Wook Kim)

1983 한양대학교 전자공학과 공학사 졸업
 1985 한양대학교 전자공학과 공학석사 졸업
 1991 Georgia 공과대학 전기공학과 공학박사 졸업
 1992 ~ 현재 광운대학교 전자재료공학과 정교수
 2000 ~ 2001 인티스닷컴(주) 연구원
 2009 ~ 현재 광운대학교 실감미디어 연구소장
 2006 ~ 현재 (사)실감미디어 산업협회 이사
 ※관심분야 : 3D 영상처리, 디지털 홀로그래프, 디지털 VLSI Testability, CAD, DSP 설계, Wireless Communication