공학석사 학위논문

High Bandwidth Memory의 열 시뮬레이션 및 관리 기법

2018년 8월

서울대학교 대학원

전기·정보공학부

신 우 재

High Bandwidth Memory의 열 시뮬레이션 및 관리 기법

지도교수 이 혁 재

이 논문을 공학석사 학위논문으로 제출함
2018년 8월

서울대학교 대학원
전기·정보공학부
신 우 재

신우재의 석사 학위논문을 인준함
2018년 8월

위 원 장 최 기 영 (인)

부위원장 이 혁 재 (인)

위 원 김 장 우 (인)

초 록

프로세서의 연산 성능이 점점 발전해 옴에 따라 높은 대역폭을 가진 메모리에 대한 수요는 항상 존재해 왔다. 이러한 대역폭 요구량을 만족시키기 위한 다양한 종류의 메모리 중 High Bandwidth Memory(HBM)가 있다. HBM은 TSV를 사용하여 DRAM 다이를 수직으로 적층한 구조를 가지고 있다. 하지만 이러한 구조적인 특성으로 인하여 열의 배출이 원활하지 못하다는 단점이 있다. 본 논문에서는 HBM의 열 문제를 파악하고 이를 해결하기 위하여 HBM을 포함한 GPU 시스템의 어플리케이션 기반 열 시뮬레이션 환경을 구성한다. 그리고 구성한 시뮬레이션 환경을 이용하여 열 문제를 파악하고 주소 매핑 방식의 변화를 통한 열 관리 기법을 제안한다. GPGPUSim과 Hotspot을 활용하여 HBM의 온도를 시뮬레이션 할 수 있는 환경을 구성하였으며 2D 컨볼루션 어플리케이션 수행시 온도 시뮬레이션 결과를 살펴보았다. 그 결과 HBM에서 수직적인 온도 불균형이 관찰되었다. 최하부의 로직 다이에서 멀어질수록 온도가 감소하는 분포를 보였다. 이를 해소하기 위해 어플리케이션의 주소 접근 패턴을 프로파일링하고 이 결과와 HBM의 채널 배치 특성을 고려한 주소 매핑 방식을 제안했다. 제안한 방식을 통해 메모리 접근을 재분배하여 HBM내의 온도 편차를 감소시켰다. 재배치 정도에 따라 두 가지 매핑 방식을 제안하였으며 각각 0.87℃, 1.45℃의 온도 편차 감소 효과가 있었다. 또한 최대온도는 각각 7.84℃, 11.82℃ 감소하였다. 제안 방법을

사용할 때 동일한 오류율을 유지하면서 리프레시 전력을 최대 52%까지 절약 가능하다.

주요어 : HBM, 열 시뮬레이션, 열 관리 기법, GPGPUSim, Hotspot
학 번 : 2016-25219

목 차

표 목 차

그 림 목 차

제 1 장 서론

1.1 연구의 배경 및 내용

딥러닝과 빅데이터 기술의 등장 이래로 이러한 기술들은 매우 빠른 속도로 발전해왔다. 딥러닝에 사용되는 네트워크는 점점 더 깊고 복잡해졌으며 데이터의 크기 및 양 또한 빠르게 증가해왔다. 이러한 추세에 맞춰 병렬처리를 위한 GPGPU(General Purpose Graphics Processing Unit) 및 HPC(High Performance Computing)기술도 발전 해 왔다. 특히 다수의 코어로 구성된 GPU는 다수의 thread를 사용하여 연산을 병렬적으로 처리하기 때문에 이를 지원해줄 메모리도 높은 대역폭이 필요하다 [1]. 이러한 메모리와 프로세서 사이의 병목현상을 해결하기 위한 대역폭 요구량을 충족시키기 위해 메모리 기술도 꾸준히 발전해 다양한 종류의 메모리가 개발되어 왔다. 최근에는 High Bandwidth Memory(이하 HBM)[2]나 Hybrid Memory Cube와 같은 DRAM 다이를 수직으로 쌓아 올리는 방식이 주목받고 있다. 그러나 2D와 달리 3D에서는 그 구조적인 특성으로 인하여 열이 쉽게 빠져나가지 못한다는 문제가 있다. DRAM에서는 온도가 올라가면 타이밍 오류가 발생하거나 동작 수명에도 영향을 미치게 된다 [3]. 또한 누설 전류가 증가하여 데이터에 오류가 생길 확률도 증가하는 등 동작 전반에 걸쳐 문제가 발생하게 된다.

이에 따라 본 논문에서는 HBM의 열 문제를 파악하기 위해 GPU와 HBM을 사용한 환경에서 어플리케이션 수행 시 HBM의 온도를 시뮬레이션 할 수 있는 환경을 구성했다. 또한 시뮬레이션으로 파악된 온도 분포를 바탕으로 해결 방안을 제시하기 위한 주소 매핑 방식을 제안한다.

1.2 논문의 구성

본 논문은 다음과 같은 순서로 구성된다. 2장에서는 HBM의 구조적 특성과 이에 따라 발생할 수 있는 문제에 대해 설명한고 열 관리 기법에 관한 관련 연구를 소개한다. 3장에서는 시뮬레이션 환경 구성에 대해 설명한다. 1절에서는 우선 GPGPUSim과 Hotspot을 사용한 전반적인 시뮬레이션 환경을 구성하게 된 순서에 대해 설명하고 각각의 부분들의 역할에 대해 설명한다. 2절에서는 GPGPUSim 시뮬레이터의 기본적인 작동 원리와 본 실험에서 어떻게 사용되었는지에 대해 설명한다. 3절에서는 Hotspot 시뮬레이터의 동작 방식과 이를 HBM에 적용하는 방법에 대해 설명한다. 4장에서는 3장에서 구성한 시뮬레이션 환경을 이용한 시뮬레이션 결과에 대해 설명한다. 1절에서는 과도 구간에 대한 시뮬레이션 결과의 경향을 설명한다. 2절에서는 다양한 상황에 대한 정상 상태 시뮬레이션 결과에 대해 설명한다. 5장에서는 4장의 시뮬레이션 결과를 바탕으로 한 열 관리 기법에 대해 설명한다. 1절에서는 일반적인 주소 매핑 방식을 HBM에서 사용하고 분석한 결과를 제시한다. 2절에서는 제안한 주

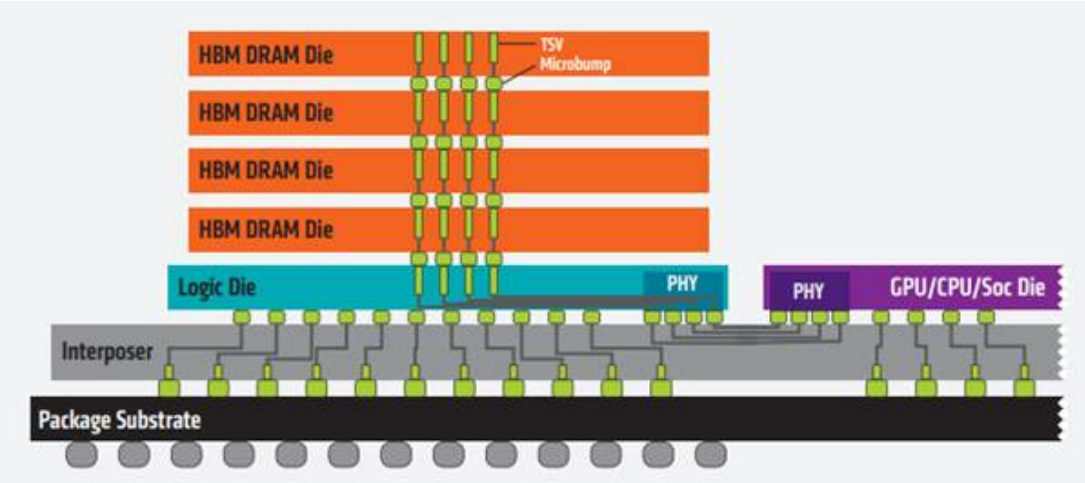
소 매핑 방식을 사용하여 시뮬레이션 한 결과로 HBM에서의 열 분포의 변화에 대해 설명한다. 6장에서는 마지막으로 본 논문의 제안 방식과 결과를 정리하여 결론을 맺는다.

제 2 장 배경 지식

2.1 HBM(High Bandwidth Memory)

HBM은 2013년에 JEDEC 표준으로 채택된 고대역폭의 메모리 인터페이스이다. DDR 인터페이스와 유사하게 동작하는 다수의 독립적인 채널로 이루어져 있다. 집적도의 향상을 위해 다수의 DRAM 다이를 수직으로 적층한 구조이며 이를 스택(stack)이라고 한다. HBM 스택은 인터포저를 통해 GPU나 CPU등의 프로세서와 통신하게 된다. DRAM 다이와 로직 다이, 인터포저와의 연결은 다수의 TSV(Through Silicon Via)를 통해 이루어지는데 이는 다이 사이에 미세한 구멍을 뚫어 와이어를 연결하는 방식이다. 다만 이러한 작업은 공정이 어렵고 불량률이 많아 수율이 떨어진다는 단점이 있다. 그렇지만 다이 간 연결의 길이가 짧아진다는 장점이 존재한다.

.



[그림 2-1] High Bandwidth Memory의 2.5D 구조 [4]

위에서 언급된 바와 같이 HBM은 CPU나 GPU 위에 DRAM이 직접 올라가 있지 않고 인터포저(interposer)를 통해 CPU 또는 GPU와 통신하게 된다. 이러한 이유로 HBM은 3D가 아닌 2.5D 구조라고 불린다. 이러한 구조는 **[그림 2-1]**에 나타나 있다. 2.5D 구조는 프로세서에 직접 구멍을 뚫지 않아도 된다는 장점이 있다. HBM의 스펙은 로직 다이에 대한 부분을 정의하고 있지는 않지만 테스트 기능 등 부가적인 기능을 위해 필요하다. 로직 다이는 DRAM 스택과 인터포저 사이에 위치해 있다.

HBM은 3차원으로 적층된 구조적 특징으로 인해 열이 쉽게 방출되지 못한다. 이에 따라 DRAM의 온도가 높은 수준으로 올라가게 되면 다양한 문제를 야기한다. 일반적인 DRAM의 동작온도는 85°C이다. DRAM 셀은 데이터를 보존하기 위해 주기적으로 전류를 재충전한다. 온도가 증

가하면 누설 전류의 증가로 인해 더 빠른 주기로 리프레시를 해주어야 한다. 잦은 빈도로 리프레시가 이루어지면 전력 소모도 많아질 뿐더러 리프레시 동작 동안에는 해당 뱅크를 사용 할 수 없으므로 성능에도 영향을 미친다. 이와 같은 이유로 DRAM의 온도를 파악하고 관리하는 것은 매우 중요하다. 이러한 문제를 해결하기 위한 열 관리 기법에 대해 연구할 때 HBM의 구조적 특성을 반영해야 한다. HBM은 기존의 메모리와는 다르게 수직으로 적층된 구조로 이루어져있기 때문에 이를 고려한 시뮬레이션 및 열 관리가 이루어져야 한다.

기존의 열 관리 기법은 HBM의 3D-stack 된 물리적 구조의 특성을 이용하지 못한다. 수직으로 적층된 DRAM 다이 사이의 온도 차이를 반영한 열 관리 기법이 필요하다. 이에 대한 실행 가능성을 연구하고 구현할 수 있는 방안을 제시한다.

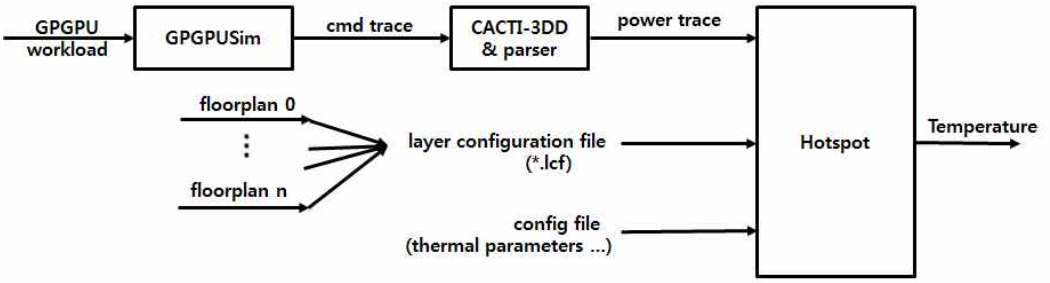
2.2 관련 연구

위에서 언급한 바와 같이 DRAM에서 온도를 관리 할 필요가 있기 때문에 다양한 상황과 다양한 구조의 DRAM에서의 온도 관리 방법에 대한 연구가 이루어져 왔다. [5]는 AMB(Advanced Memory Buffer)가 있는 Fully Buffered DIMM을 모델링하고 이 구조적 특성을 이용한 열 관리 기법을 제안했다. 이들은 단순히 사용하는 코어의 수를 제한하거나 동작 주파수와 동작 전압을 제한하는 방식의 DTM(Dynamic Thermal

Management)를 제안했다. [6]은 GPU에서 다양한 열 관리 기법의 성능을 평가했다. 그러나 이들의 열 시뮬레이션에서 메모리에 대한 부분은 제외되었다. [7]은 PIM을 포함한 3차원 메모리 구조에서 일정한 온도 이하로 유지하기 위한 전력의 한계점을 제시했다. 그러나 이 논문에서는 다이 간의 온도 분포에 대한 언급이 없고 각 다이에서 일정한 전력을 소모하는 것을 가정하였다. [8]는 CPU와 메모리를 포함한 구조의 열 시뮬레이션을 제시하였다. 본 논문에서는 이들의 선행연구와는 다르게 GPU 어플리케이션의 수행 결과를 바탕으로 하고, HBM의 수직 적층 구조 및 로직 다이를 고려한 열 모델링을 포함하고, HBM 구조에서의 열 관리 기법을 제시한다.

제 3 장 시뮬레이션 환경 구성

3.1 시뮬레이션 개요



[그림 3-1] 시뮬레이션 개념도

[그림 3-1]은 시뮬레이션 개요를 나타낸 개념도이다. 크게는 GPU를 시뮬레이션 하는 cycle 정확도를 가진 아키텍처 시뮬레이터인 GPGPUSim[9]과 IC의 열 모델인 Hotspot[10]을 기반으로하여 시뮬레이션 환경을 구성하였다.

우선 시뮬레이션에 사용할 GPGPU 워크로드를 GPGPUSim 시뮬레이터에서 수행하게 된다. GPGPUSim의 코드를 수정하여 특정 사이클 주기로 실행 결과를 주기적으로 기록하게 하였다. 또한 메모리 커맨드를 기록하도록 하여 각 사이클에 어떤 메모리 커맨드가 수행되었는지 알 수 있다. 이를 이용하여 메모리 커맨드 트레이스를 생성하고 CACTI-3DD[11]의 결과로 얻은 각 커맨드에 소요되는 에너지 값을 곱

하여 각 시간 구간에서의 전력을 계산한다. 커맨드 트레이스는 어떤 사이클에 메모리가 어떤 커맨드를 수행하는 지에 대한 정보를 포함하고 있다. 로우 커맨드로는 액티베이트(activate), 프리차지(precharge) 커맨드를 기록하고 컬럼 커맨드로는 읽기(read), 쓰기(write) 커맨드를 기록한다. 또한 메모리 주소를 어드레스 디코더에서 디코딩하여 나온 채널, 뱅크, 로우, 칼럼에 관한 정보를 포함하고 있다. 이러한 방식으로 생성된 메모리 커맨드 트레이스를 읽어 들이고 계산하여 전력 트레이스를 생성한다. 이렇게 생성된 전력 트레이스는 Hotspot 시뮬레이터의 입력으로 사용된다.

또한 HBM의 열 시뮬레이션을 위해 Hotspot 시뮬레이터는 물리적, 열적 특성에 대한 정보를 입력받게 된다. 이를 정의하기 위해 lcf(layer configuration file)파일을 구성한다. lcf파일은 HBM 각각의 레이어의 평면도(floorplan) 및 해당 레이어의 두께, 그리고 heat flow, specific heat, resistivity 등 물질의 열적 특성에 대한 정보를 포함한다. 모델링 된 HBM의 구조는 DRAM 다이와 그 사이를 채우는 TIM(Thermal Interface Material)이 차례로 적층되어 있는 구조이다. 각 레이어의 평면도는 flp(floorplan)파일에서 정의된다. 레이어를 이루고 있는 각 블록들의 위치 및 가로, 세로 길이 그리고 열적 특성에 변수 값들을 포함한다. DRAM 다이의 floorplan은 [12]의 die shot을 참조하여 모델링하였다. 또한 DRAM 다이 사이사이에 채워져 있는 물질 또한 모델링하여 순서

대로 쌓아 lcf 파일을 구성한다.

[7]에 따르면 냉각팬과 같은 active heat sink가 없는 경우에 로직 다이에서 동작 온도를 넘지 않는 전력은 8.5W이다. 따라서 1~10W 범위에서 로직 다이의 전력을 변화 시켰다.

실험에 사용한 워크로드는 polybench[13]의 2DConv를 사용했다. polybench는 누가 CUDA 및 OpenCL로 GPU 벤치마크이다. 이는 4096 x 4096 크기의 행렬을 3 x 3 크기의 필터로 컨볼루션(convolution)을 수행하는 프로그램이다. 컨볼루션 연산은 CNN(Convolution Neural Network)등 최근의 다양한 딥러닝 어플리케이션에서 이미지 인식을 위해 빠지지 않고 사용된다. 따라서 GPU에서 수행되는 컨볼루션 연산의 특성을 파악하고자 해당 워크로드를 사용했다.

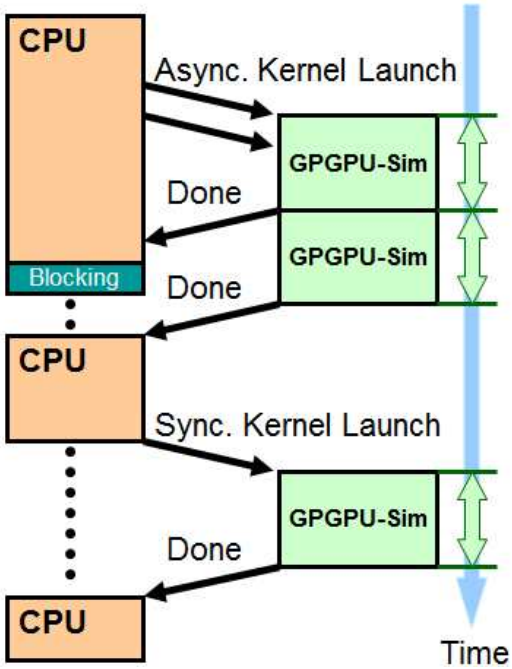
본 논문의 실험에서 사용된 HBM의 세팅은 [표 3-1]과 같다.

# of dies	4 DRAM + 1 logic
# of channel	2 /die
# of banks	8 /channel
# I/Os	128 / channel
I/O speed	500MHz DDR
Bandwidth	1024Gbps
Timing parameters (ns)	tRCD=15, tRAS=16, tRC=48,tCCD=4, tRRD=4

[표 3-1] 시뮬레이션에 사용된 HBM의 옵션

3.2 GPGPUSim 시뮬레이터

GPGPUSim은 GPU의 타이밍, 성능을 평가하기 위한 아키텍처 시뮬레이터로 CUDA와 OpenCL 프로그램의 실행 결과를 시뮬레이션 한다. CUDA 어플리케이션을 실행 하면 CPU에서 수행되는 부분은 실제로 CPU에서 수행된다. 반면 GPU에서 수행되는 CUDA 커널은 GPGPUSim에 설정된 GPU 모델에서의 수행을 CPU에서 시뮬레이션 한다. 따라서 실제 GPU를 사용하지 않고도 GPU의 다양한 수행 결과 정보를 얻을 수 있으며 가상의 GPU 구조를 설정하여 실험하는 것도 가능하다.



[그림 3-2] GPGPUSim의 실행 방식

[그림 3-2]은 GPGPUSim의 실행 방식을 나타내는 그림이다. 각각의 CUDA 커널들이 실행 될 때 원래는 GPU 하드웨어에서 실행되는 부분을 GPGPUSim이 대신 수행하게 된다.

GPGPUSim에서는 address mapping, cache, core 수 등의 다양한 옵션을 사용할 수 있다. GPU 옵션은 Titan X와 유사한 설정을 사용하였다 [14]. GPGPUSim에서는 GTX 480을 모델링한 설정 파일을 제공한다. 이를 Titan X의 설정으로 바꾸기 위해 설정한 파라미터의 내용은 [표 3-2]와 같다.

Configuration Option	GTX 480	GTX Titan X
-gpgpu_n_clusters	15	96
-gpgpu_clock_domains	700.0:700.0:700.0:924.0	1000.0:1000.0:1000.0:1753.0
-gpgpu_shader_registers	32768	16384
-gpgpu_shader_core_pipeline	1536:32	512:32
-gpgpu_pipeline_widths	2,1,1,2,1,1,2	1,1,1,1,1,1,2
-gpgpu_num_sp_units	2	1
-gpgpu_cache:dl1	32:128:4,L:L:m:N:H,A:32:8,8	16:128:4,L:L:m:N:L,A:32:8,8
-gpgpu_shmem_size	49152	24576
-gpgpu_cache:dl2	64:128:8,L:B:m:W:L,A:32:4,4:0,32	256:128:8,L:B:m:W:L,A:32:4,4:0,32
-gpgpu_tex_cache:l1	4:128:24,L:R:m:N:L,F:128:4,128:2	16:128:24,L:R:m:N:L,F:128:4,128:2
-gpgpu_const_cache:l1	64:64:2,L:R:f:N:L,A:2:32,4	128:64:2,L:R:f:N:L,A:2:32,4
-gpgpu_operand_collector_num_units_sp	6	4
-gpgpu_operand_collector_num_units_sfu	8	4
-gpgpu_num_reg_banks	16	8
-gpgpu_max_insn_issue_per_warp	1	2
-inter_config_file	config_fermi_islip.icnt	config_maxwell_islip.icnt
-gpgpu_dram_return_queue_size	116	160
-gpgpu_n_mem_per_ctrlr	2	4
-gpgpu_dram_buswidth	4	2
-gpgpu_num_sched_per_core	2	1

[표 3-2] GPGPUSim 옵션

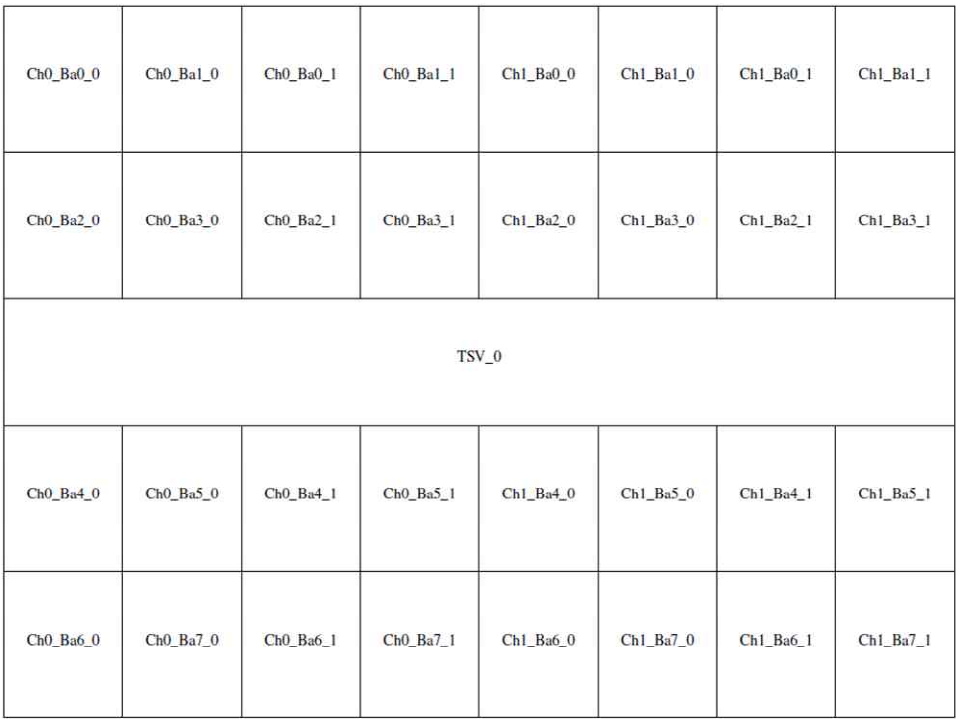
3.3 Hotspot 시뮬레이터

2D 집적회로의 경우도 집적도가 점점 향상되었고 앞서 언급된 바와 같이 3D 집적회로는 더 작은 면적에 전력이 집중되므로 열 측면에서 불리하다는 단점이 있다. Hotspot은 이러한 열 문제를 파악하기 위한 시뮬레이션을 제공하는 도구 중 하나이다. Hotspot은 각각의 블록 또는 그리드의 열 RC 모델링을 통한 열 시뮬레이션을 제공한다. 이 방식은 집적회로의 각 부분을 블록 또는 그리드로 나누어 각각의 열적 특성을 전기적 특성과의 쌍대성을 이용하여 변환하여 모델링한다. 열의 흐름은 전류에 대응되며, 온도차이는 전압에 대응되고 전기회로와 유사하게 이런 흐름은 열 저항에 의해 방해 받는다. 열 캐패시턴스는 RC회로에서의 캐패

시턴스의 역할과 같이 온도 변화의 지연을 모델링하기 위해 사용된다. 이렇게 모델링된 회로를 나타내는 미분방적식을 4차 Runge-Kutta 방법을 사용하여 수치해석적으로 풀어낸다.

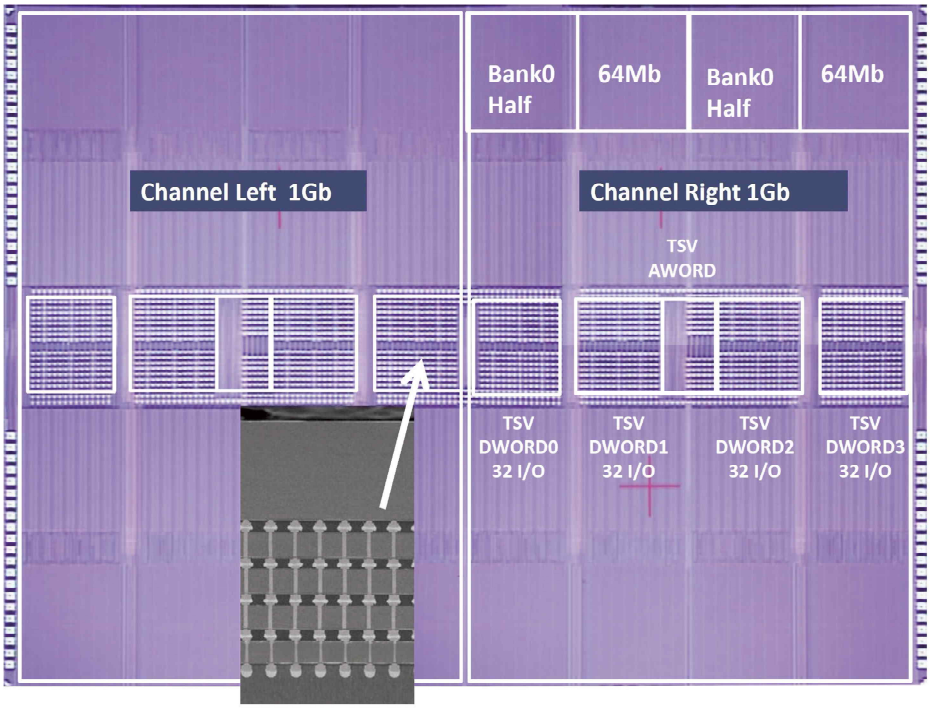
Hotspot의 시뮬레이션 방식은 블록 모드와 그리드 모드, 두 가지로 나누어진다. 블록 모드는 각각의 컴포넌트들을 하나의 블록으로 처리하여 대응되는 RC회로를 구성 한다. 반면에 그리드 모드는 칩을 일정한 간격의 격자로 나누어 RC회로를 구성 한 후 시뮬레이션을 진행한다. 블록 모드는 그리드 모드에 비해 모델이 간단하여 시뮬레이션 시간이 짧다는 장점이 있지만 그리드 모드에 비해 상대적으로 부정확하다는 단점이 있다. 본 논문에서 진행한 실험에서는 3D 구조의 HBM을 대상으로 진행하는데 3D 모드에서는 그리드 모드만 지원하기 때문에 그리드 모드를 사용한다. 4장의 시뮬레이션 결과를 살펴보면 한 레이어에서의 온도 표현이 64x64개의 그리드로 이루어 져있는 것을 확인할 수 있다. 블록 모드를 사용하는 경우 블록에 포함되는 그리드들이 하나의 블록을 이룬다. 3D 시뮬레이션의 경우 모드는 그리드 모드만 지원한다.

Hotspot으로 설정할 수 있는 파일들에는 floorplan file과 configuration 파일이 있다. 특히 3D 시뮬레이션에서는 2D로 이루어진 floorplan을 쌓아 올린 layer configuration file이 필요하다. cooling 방법, 각각의 요소들 간의 thermal conductivity와 dimension등의 정보가 필요하다.



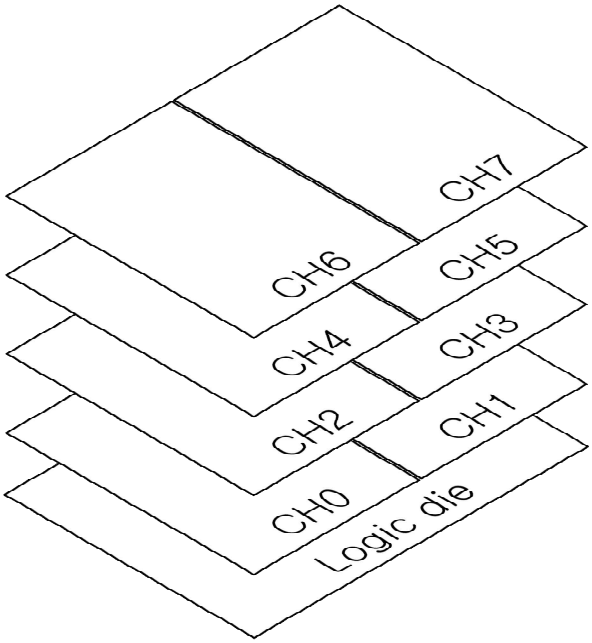
[그림 3-3] 생성된 HBM DRAM 다이의 flp 파일

[그림 3-3]는 모델링된 HBM의 DRAM 다이의 평면도이다. HBM의 DRAM 다이는 뱅크(bank) 단위로 나누어 모델링 했다. 이는 **[그림 3-4]**의 **[12]**의 연구에서 나온 DRAM die shot을 참조하여 생성되었다. DRAM 부분은 위 아래로 나누어져 있고 가운데로는 TSV 영역이 있다.



[그림 3-4] High Bandwidth Memory의 단면도[10]

수직으로 쌓아올린 DRAM 다이의 모습은 [그림 3-5]와 같다. 이때 각 채널은 한 레이어에 두 개씩 나누어져 배치된다. 높은 채널 번호를 가질수록 높은 위치에 위치하게 된다.



[그림 3-5] HBM 스택의 채널 배열

Hotspot 시뮬레이터에서는 냉각의 성능도 설정 가능하다. [8]에서는 [표 3-3]와 같이 냉각 성능을 네 가지로 구분하여 각각의 대류 열 전도도 값을 측정하였다. 이 값을 사용하여 각각의 냉각 성능에 따른 열 분포에 관한 시뮬레이션이 가능하다.

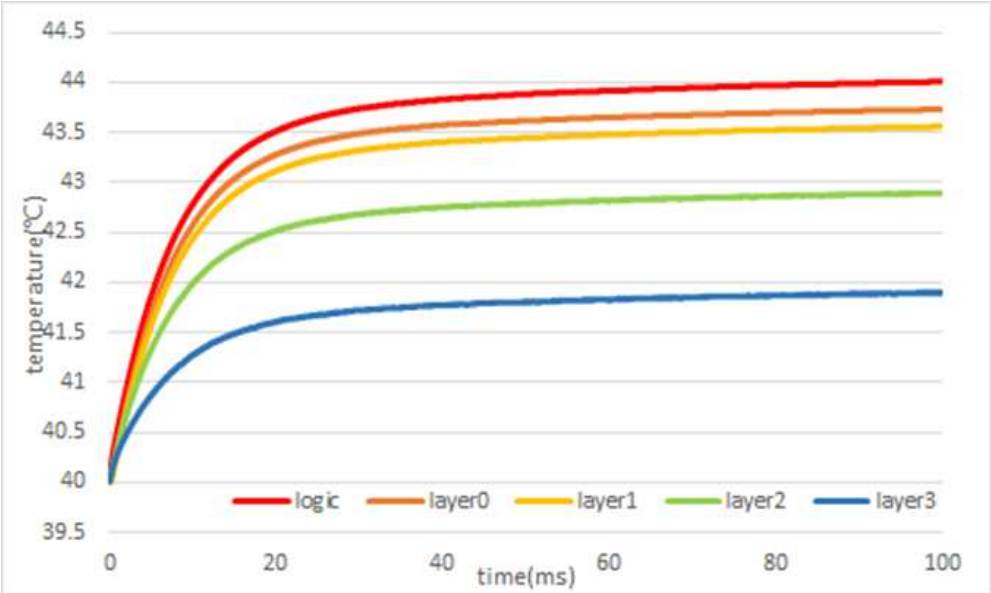
Cooling Solution	Convection Thermal Conductivity(℃/W)
Passive	4.0
Low-end active	2.0
Commodity server active	0.5
High-end server active	0.2

[표 3-3] 냉각 방식에 따른 대류 열 전도도[8]

[표 3-2]에 있는 네 가지 냉각방식 모두 기본적으로 히트 싱크(heat sink)를 사용한다. 첫 번째 passive의 경우 히트 싱크만 사용하는 냉각방식이다. 나머지 세 개의 냉각 방식은 히트 싱크와 냉각 팬을 함께 사용한 방식이다. 그리고 이 방식들의 성능에 따라 대류 열 전도도(convection thermal conductivity)가 달라진다. passive 냉각 방식에서 high-end server 냉각 방식으로 갈수록 높은 냉각 성능을 보이며 냉각에 필요한 비용 또한 늘어나게 된다.

제 4 장 시뮬레이션 결과

4.1 과도(transient) 구간 시뮬레이션



[그림 4-1] 과도 구간 시뮬레이션 결과

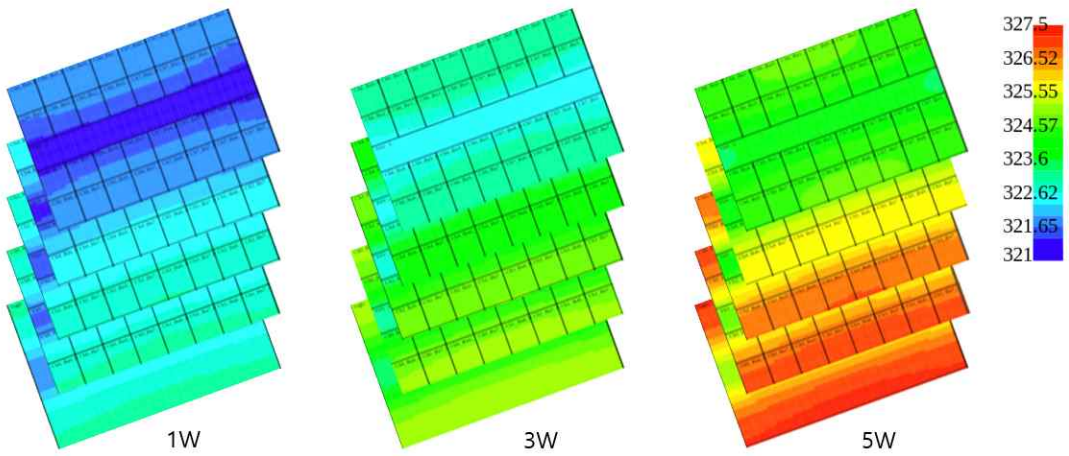
제 3장에서 구성한 시뮬레이션 환경을 이용하여 GPU에 장착된 HBM의 열 분포를 시뮬레이션 했다. **[그림 4-1]**은 주위온도가 45°C이고 로직 다이에서 소모되는 전력은 10W그리고 low-end active 냉각 방식을 사용한 상황에서의 과도(transient) 시뮬레이션을 수행하여 레이어별 최대 온도 값을 기록한 결과이다. 각 레이어의 최대 온도는 초기에는 빠르게

증가하다 로그 함수의 형태로 시간에 따라 점점 천천히 증가하는 경향을 확인 할 수 있다. 제일 하부에 위치한 로직 다이가 가장 높은 온도를 나타내며 제일 상부에 위치한 DRAM 다이(layer3)는 가장 낮은 온도를 나타낸다. 로직 다이와 가까울수록 온도 차이가 적으며, 멀어질수록 다이 사이의 온도의 차이도 커지는 분포를 보인다. 이는 히트싱크의 영향으로 상부는 열이 쉽게 방출되는데 반해 하부에는 로직 다이에서 발생하는 전력에 의한 열이 쉽게 방출되지 못하기 때문이다.

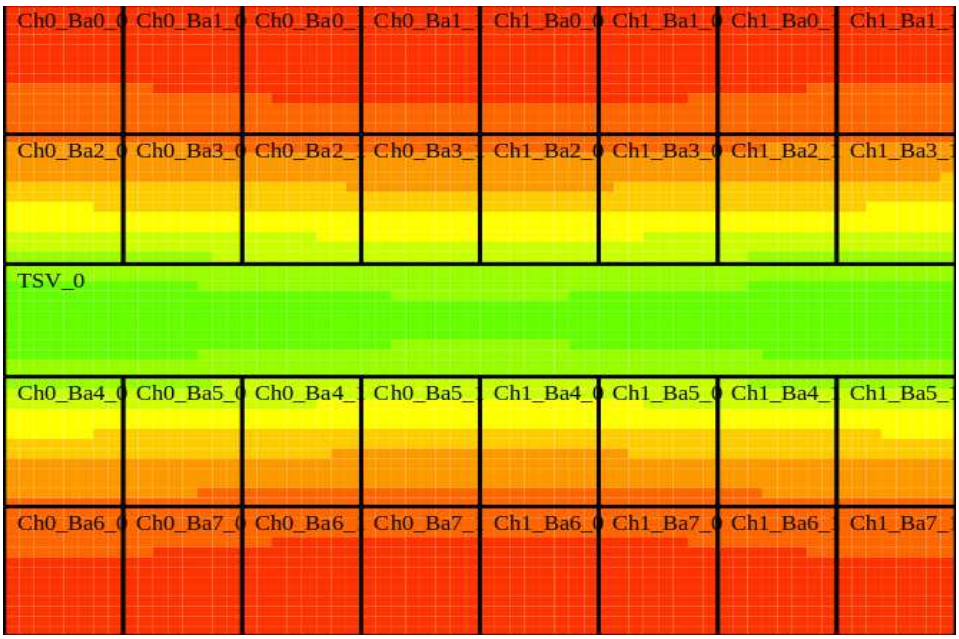
4.2 정상 상태(steady state) 시뮬레이션

[그림 4-2]는 정상 상태 시뮬레이션 결과를 시각화하여 나타낸 것이다. 제일 아래에 위치한 레이어가 로직 다이이며, 그 위로는 4개의 DRAM 다이가 쌓여있는 모습이다. 주위 온도는 45°C Commodity server active 냉각 방식, 그리고 로직 다이에서 소모되는 전력값을 각각 1W, 3W, 5W로 바꿔가며 시뮬레이션 하였다. 정상 상태 시뮬레이션 결과 또한 4.1의 과도 구간 시뮬레이션 결과와 유사한 경향을 보인다. 수직적인 온도 분포는 가장 하부에 위치한 로직 다이의 온도가 제일 낮으며, 가장 상부에 위치한 DRAM 다이의 온도가 제일 높은 것을 확인 할 수 있다. 또한 수평적 온도 분포는 모든 layer에 걸쳐서 TSV 영역의 온도가 제일 낮은 것을 확인 할 수 있다. 이는 TSV에서 발생되는 전력이 작은 이유도 있지만 높은 thermal conductivity를 가져 열이 더 원활하게

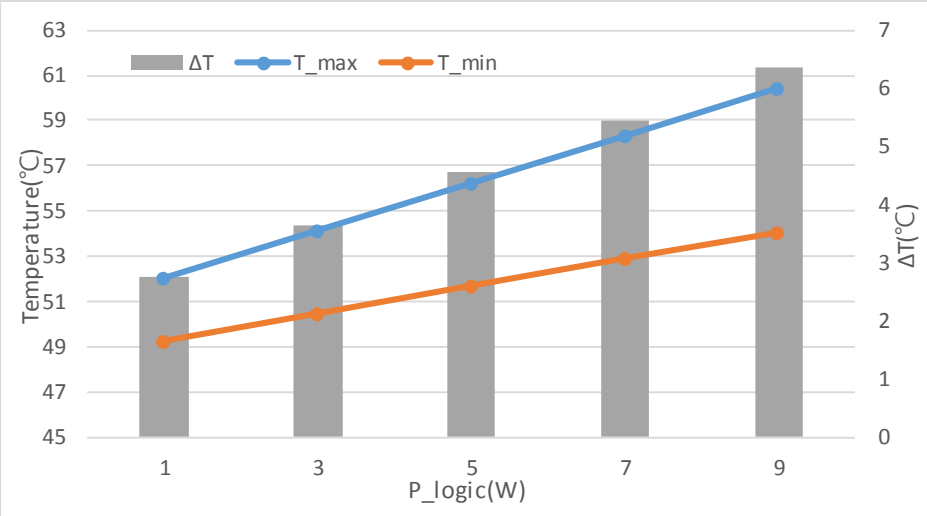
방출되는 영향도 있다. 이러한 수평적 온도 분포의 자세한 경향은 **[그림 4-3]**에서 확인 가능하다.



[그림 4-2] 로직 다이 전력에 따른 온도 HBM의 비교

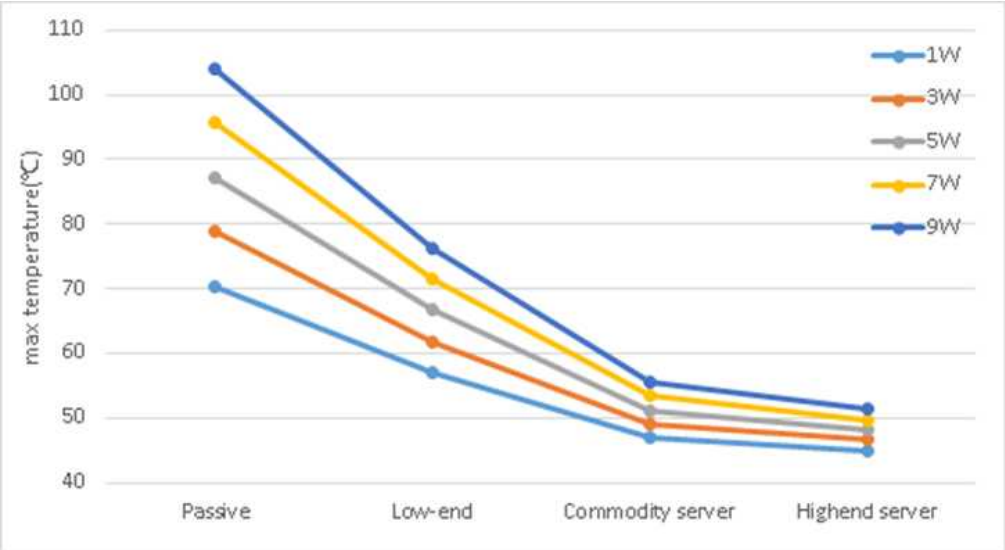


[그림 4-3] HBM 레이어 내의 수평적 온도 분포



[그림 4-4] 로직 다이 전력에 따른 최고,최소 온도 및 온도 편차

로직 다이의 전력에 따른 최대온도와 최소온도, 그리고 그 편차의 시뮬레이션 결과는 **[그림 4-4]**와 같다. 로직 다이의 소모 전력이 증가할수록 최대온도 및 최소온도는 증가한다. 다만 최대 온도가 증가하는 기울기가 더 크기 때문에 최하부 다이와 최상부 다이의 온도 차이는 점점 커진다. 이는 냉각을 하는 히트 싱크가 상부에 위치해 있기 때문에 상부의 DRAM 다이는 하부의 DRAM 다이에 비해 더 원활한 냉각이 이루어지기 때문이다.

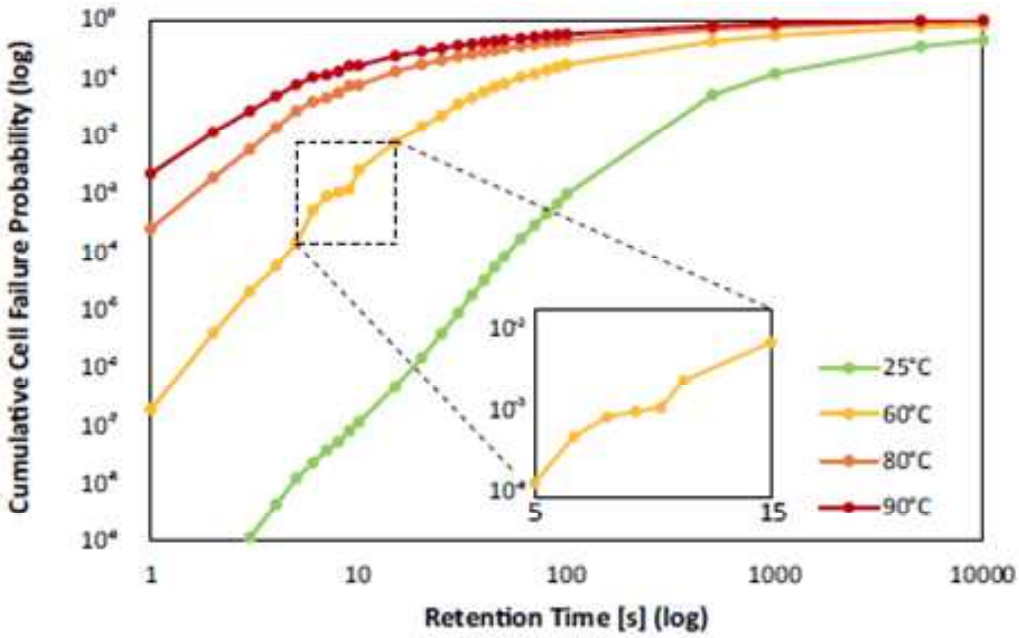


[그림 4-5] 냉각방식 및 로직 다이 전력에 따른 최고 온도

[그림 4-5]는 각각의 냉각방식을 적용한 시뮬레이션에서 로직 다이 전력에 따른 최고 온도를 나타낸다. 고성능의 냉각 방식을 사용 할수록 최대 온도가 낮아지며, 로직 다이 전력의 증가에 따른 온도의 증가 폭이 커지게 된다.

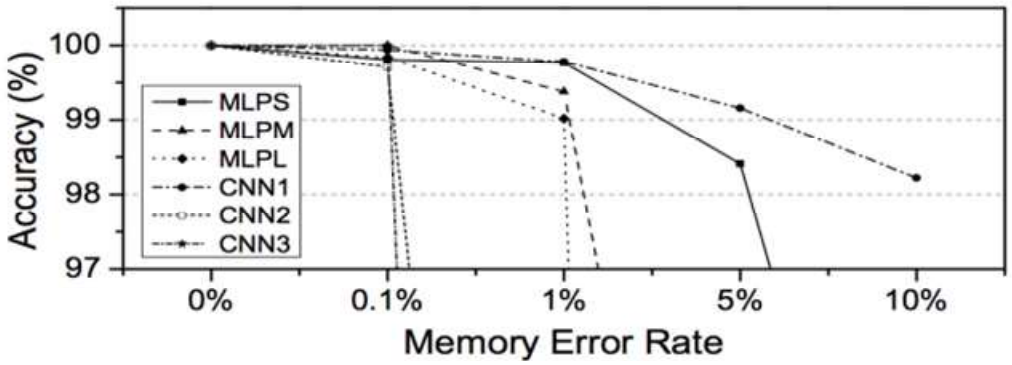
제 5 장 열 관리 기법

4장에서는 3장에서 구성한 시뮬레이션 환경을 이용하여 GPU에 HBM을 사용 했을 때 열적 분포에 대해 살펴보았다. 실험 결과 HBM의 상단에 위치한 다이와 하단에 위치한 다이 사이의 불균일한 열 분포를 확인했다. [15]에 따르면 온도가 올라갈수록 DRAM은 누설전류가 증가하기 때문에 오류가 발생할 확률이 증가한다. 따라서 더 잦은 리프레시 동작이 필요하게 된다.



[그림 5-1] 온도에 따른 retention time과 오류 확률[15]

또한 [그림 5-1]에서 볼 수 있듯이 DRAM에서의 오류 확률이 증가하면 신경망(Neural Network)의 추정(inference) 정확도가 급격히 감소하게 된다. 이와 같은 현상은 MLP(Multi Layer Perceptron)과 같은 신경망에서 보다 CNN과 같은 네트워크에서 두드러진다.



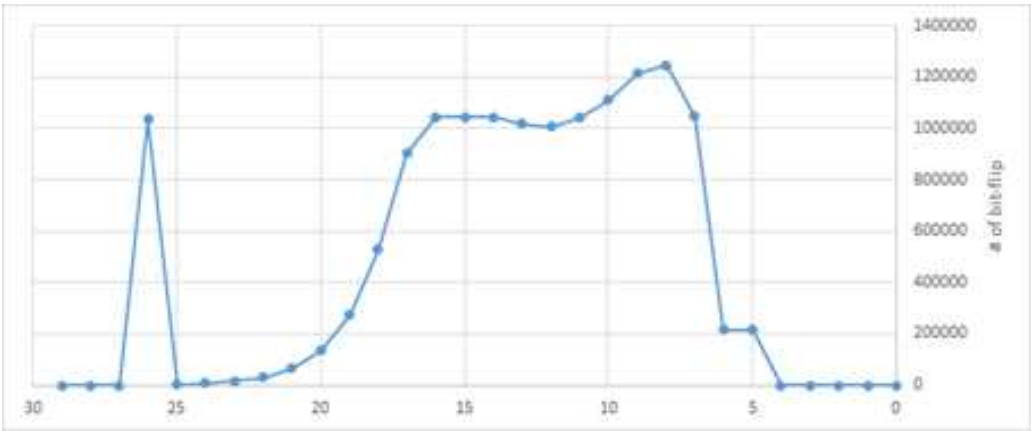
[그림 5-2] 메모리 오류율에 따른 신경망의 추정 정확도 [16]

5.1 HBM의 주소 매핑

본 장에서는 4장의 시뮬레이션 결과를 바탕으로 HBM 구조를 고려한 열 관리 기법을 제시한다. 그 방법으로 주소 매핑(address mapping)을 이용한다. 우선 워크로드의 특성을 파악하기 위해 프로파일링을 통해 비교 대상으로 사용할 베이스라인 매핑 방식을 결정한다.

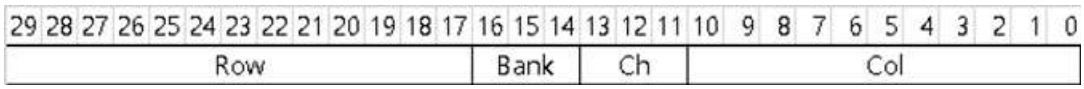
HBM은 독립적인 여러 채널(channel)로 구성되어 있으며, 채널 내에서는 뱅크(bank)라는 단위로 나누어진다. 뱅크는 다수의 로우(row)로 이

루어지며 로우는 다수의 컬럼(column)으로 이루어진 구조이다. HBM내의 특정 주소에 접근 할 때 접근하게 되는 로우를 빠르게 접근하기 위해서 로우 버퍼(row buffer)에 저장하는데 이를 액티베이트(activate)한다고 한다. 메모리에 접근하는 다음 주소가 이전과 같은 뱅크의 같은 로우에 해당한다면 빠르게 로우 버퍼를 참조하여 원하는 주소에 접근할 수 있으므로 빠르게 접근이 가능하다. 이를 로우 버퍼 히트(row buffer hit)라고 한다. 반면 메모리에 접근하는 다음 주소가 이전과 같은 뱅크의 다른 로우에 접근하는 경우에는 로우 버퍼에 들어있는 로우를 새로 접근할 로우로 대체해야 한다. 이를 로우 버퍼 미스(row buffer miss)라 한다. 이 경우 추가적인 동작이 필요하므로 지연시간이 증가하며 성능도 악화되게 된다. 따라서 효율적인 주소 맵핑 방식은 같은 뱅크, 다른 로우의 주소에 연속적으로 접근하는 것, 로우 버퍼 미스를 최소화 하도록 설계된다.



[그림 5-3] 주소 비트의 bit-flip 횟수

로우 버퍼 미스를 최소화하기 위한 매핑을 찾기 위해 **[그림 5-3]**과 같이 메모리 주소 접근 시 연속된 주소에서 bit-flip이 일어난 횟수를 특정했다. 그래프의 가로축은 메모리 어드레스 중 몇 번째 비트인지를 의미하고 세로축은 bit-flip이 일어난 횟수를 나타낸다. 각각의 다른 비트를 사용하는 횟수를 나타낸 것이다.



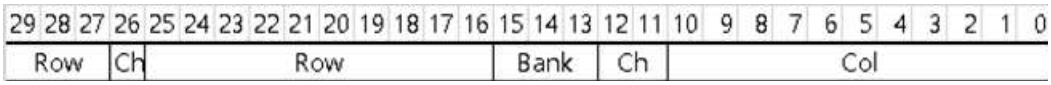
[그림 5-4] 기준 주소 매핑 (map1)

우선 bit-flip이 많이 일어나는 비트는 채널의 병렬성을 활용하기 위해 채널에 해당하는 비트에 맵핑된다. 뱅크 또한 병렬성을 활용하기 위해 다음으로 bit-flip이 빈번한 위치의 비트에 매핑된다. 다음으로 로우는 연속된 주소가 같은 로우에 접근하는 것이 유리하므로 bit-flip이 잘 일어나지 않는 위치에 매핑된다. 이와 같은 매핑 순서는 로우-뱅크-채널-컬럼 순서이다.

5.2 제안 주소 매핑 방식

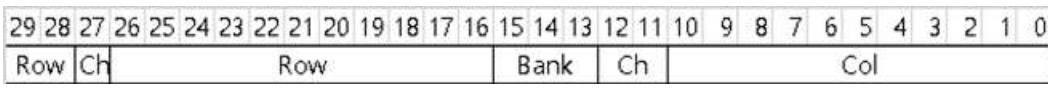
주소 매핑방식을 변경하여 HBM으로의 메모리 접근을 재분배 하기

위해 우선 주소의 각각의 비트를 사용하는 비율을 조사하였다. 그 결과 대부부의 비트는 균등한 비율로 사용되었으나, 26번 비트는 약 2:1의 비율로 1을 더 많이 사용하였으며, 27번 비트는 대부분 0을 사용하는 것을 확인 할 수 있었다. 따라서 주소를 디코딩하는 주소 디코더(address decoder)에 간단한 로직을 추가하여 해당 비트를 채널 비트로 사용한다면 각각의 채널에 접근하는 메모리 접근의 비율을 조절 할 수 있다. 이를 이용해 다음과 같은 주소 매핑 방식을 제안한다.



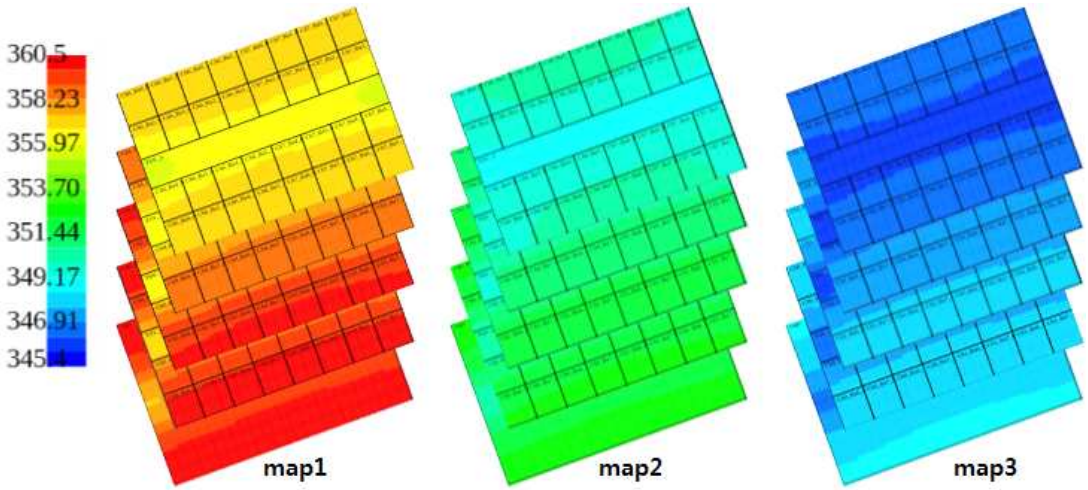
[그림 5-5] 제안 주소 매핑 1 (map2)

[그림 5-5]는 약 2대1의 비율로 1을 더 많이 사용하는 26번 비트를 채널의 최상위 비트로 사용하였다. 이에 따라 8개의 채널 중 상위 4개인 4, 5, 6, 7번 째 채널이 하위 4개의 채널인 0, 1, 2, 3번 채널보다 2배 많이 접근된다. 상위 채널은 물리적으로도 높은 곳에 위치하므로 메모리 접근이 HBM의 높은 쪽으로 치우치게 된다.



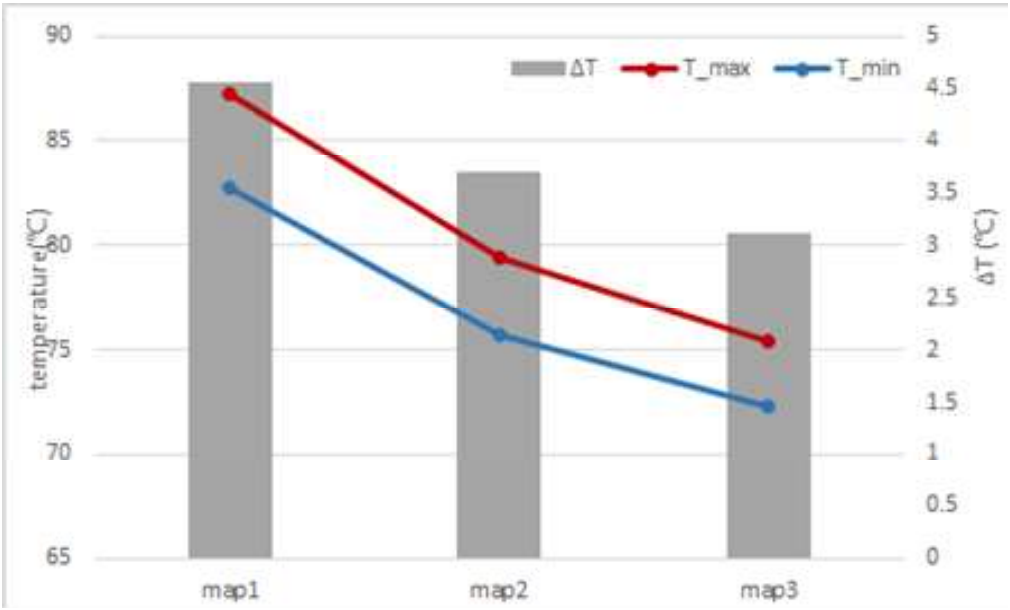
[그림 5-6] 제안 주소 매핑 2 (map3)

[그림 5-6]은 대부분 0만 사용하는 27번 비트를 채널의 최상위 비트로 사용하는 주소 매핑 방식을 나타낸다. 메모리 접근을 상위 번호의 채널로 집중시키기 위해 27번 비트를 flip하여 사용하였다. 따라서 대부분의 메모리 접근은 HBM의 상부 2개 레이어에 위치한 4개의 채널에 집중되게 된다. 위의 세 가지 주소 매핑 방식을 사용하여 Commodity Server 수준의 냉각 방식, 로직다이 전력은 5W인 상황에서 시뮬레이션 했다.



[그림 5-7] 주소 매핑 방식에 따른 HBM의 온도 분포

[그림 5-7]은 hotspot 시뮬레이션 결과를 눈으로 확인하기 쉽게 시각화한 결과이다. 이를 통해 불균형한 주소 매핑 방식을 사용할수록 전반적인 온도가 감소하는 경향을 확인 할 수 있다.



[그림 5-8] 주소 매핑 방식에 따른 최대, 최소 온도분포와 편차

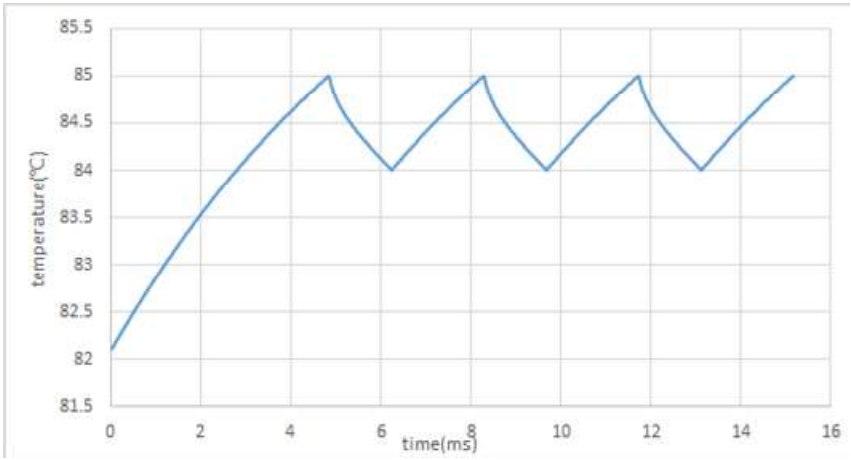
[그림 5-8]은 각각의 매핑 방법을 사용 할 때 HBM의 최대, 최소온도와 이 차이를 나타낸 그래프이다. 왼쪽 축은 HBM 메모리 내의 최대 온도와 최소온도를 나타내며 막대 그래프에 해당하는 오른쪽 축은 이 둘 간의 온도 편차를 나타낸다. 기준 주소 매핑(map1)에서의 온도 편차는 4,56℃에서 map2에서는 3.7℃로 map3에서는 3.12℃로 온도 편차가 줄어드는 것을 확인 할 수 있다. 여기에서도 마찬가지로 불균일한 주소 매핑을 사용할수록 최대 및 최소 온도도 함께 감소하는 것을 확인 할 수 있다. 최대온도는 map1에서는 87.26℃, map2에서는 79.42℃, map3에서는 75.45℃로 점점 감소하였다.

	map1	map2	map3
IPC	394.72	272.91	215.45

[표 5-1] 주소 매핑 방식에 따른 성능

[표 5-1]은 각 주소 매핑 방식에서의 IPC(Instruction per cycle) 값을 나타낸다. HBM의 채널을 골고루 모두 사용하지 않고 특정한 부분만 사용하는 매핑 방식일수록 때문에 최대의 대역폭을 활용할 수 없기 때문에 낮은 IPC를 갖게 된다.

[17]에서는 DTM(Dynamic Thermal Management)을 통한 온도 관리 기법을 사용하였는데, 이때 85℃에 도달하면 동작을 중지하는 thermal shutdown이나 대역폭을 제한하는 bandwidth throttling을 사용할 경우 성능이 최소 44.8% 감소하는 결과를 보였다. 다른 DTM을 사용할 때의 성능 저하보다. 85℃에서 DTM을 가동하고 84℃에서 DTM을 해제하는 방식이 성능저하가 가장 작은 것으로 나타났다.



[그림 5-9] thermal shutdown 사용 시 map1에서의 transient 시뮬레이션

[18]의 결과에 본 방법의 온도감소 효과를 적용하면 동일 수준의 오류율을 유지하면서 감소시킬 수 있는 리프레시 전력을 계산할 수 있다. 그 결과 동일 오류율에서 map2에서는 리프레시 전력을 최대 38.7%까지 map3에서는 최대 52.2%까지 감소시킬 수 있다는 결과를 얻었다.

제 6 장 결론

본 논문에서는 HBM을 포함한 GPU 시스템의 어플리케이션 기반 시뮬레이션 환경 구성, 그리고 주소 매핑 방식의 변화를 통한 열 관리 기법 및 시뮬레이션 결과에 대해 살펴보았다. GPGPUSim과 Hotspot을 활용하여 HBM의 온도를 시뮬레이션 할 수 있는 환경을 구성하였으며 2D 컨볼루션 어플리케이션 수행시 온도 시뮬레이션 결과를 살펴보았다. 그 결과 HBM에서 수직적인 온도 불균형이 관찰되었다. 최하부의 로직 다이에서 멀어질수록 온도가 감소하는 분포를 보였다. 이를 해소하기 위해 어플리케이션의 주소 접근 패턴을 프로파일링하고 이 결과와 HBM의 채널 배치 특성을 고려한 주소 매핑 방식을 제안했다. 제안한 방식을 통해 메모리 접근을 재분배하여 HBM내의 온도 편차를 감소시켰다. 재배치 정도에 따라 두 가지 매핑 방식을 제안하였으며 각각 0.87℃, 1.45℃의 온도 편차 감소 효과가 있었다. 또한 최대온도는 각각 7.84℃, 11.82℃ 감소하였다. 제안 방법을 사용할 때 동일한 오류율을 유지하면서 리프레시 전력을 최대 52%까지 절약 가능하다.

참 고 문 헌

[1] Dublish, Saumay, Vijay Nagarajan, and Nigel Topham. "Evaluating and mitigating bandwidth bottlenecks across the memory hierarchy in GPUs." *Performance Analysis of Systems and Software (ISPASS), 2017 IEEE International Symposium on. IEEE*, 2017.

[2] "JEDEC Solid State Technology Association-JESD 235", JEDEC Spec High Bandwidth Memory (HBM) DRAM Specification Rev, vol. 1, no. 30, Jan 2015.

[3] Skadron, Kevin, et al. "Temperature-aware microarchitecture." Computer Architecture, 2003. *Proceedings. 30th Annual International Symposium on. IEEE*, 2003.

[4] AMD, High-Bandwidth Memory (HBM) Reinventing Memory Technology

[5] Lin, Jiang, et al. Thermal modeling and management of DRAM memory systems. Vol. 35. No. 2. ACM, 2007.

[6] Sheaffer, Jeremy W., Kevin Skadron, and David P. Luebke. "Studying thermal management for graphics-processor architectures." *Performance Analysis of Systems and Software, 2005. ISPASS 2005. IEEE International Symposium on. IEEE*, 2005.

[7] Eckert, Yasuko, Nuwan Jayasena, and Gabriel H. Loh. "Thermal feasibility of die-stacked processing in memory." (2014).

[8] Zhu, Yuxiong, et al. Integrated Thermal Analysis for Processing In Die-Stacking Memory. *MEMSYS*, 2016.

[9] Ali Bakhoda, George Yuan, Wilson W. L. Fung, Henry Wong, Tor M. Aamodt, Analyzing CUDA Workloads Using a Detailed GPU Simulator, in *IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, Boston, MA, April 19-21, 2009.

[10] R. Zhang, M. R. Stan, and K. Skadron, "HotSpot 6.0: Validation, Acceleration and Extension." University of Virginia, Tech. Report CS-2015-04

[11] Chen, Ke, et al. "CACTI-3DD: Architecture-level modeling for 3D die-stacked DRAM main memory." *Proceedings of the Conference on Design, Automation and Test in Europe*. EDA Consortium, 2012.

[12] Lee, Dong Uk, et al. "25.2 A 1.2 V 8Gb 8-channel 128GB/s high-bandwidth memory (HBM) stacked DRAM with effective microbump I/O test methods using 29nm process and TSV." *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2014 IEEE International. IEEE*, 2014.

[13] Grauer-Gray, Scott, et al. "Auto-tuning a high-level language targeted to GPU codes." *Innovative Parallel Computing (InPar), 2012. IEEE*, 2012.

[14] Hall, Piriya Kristofer. "Adaptation of a GPU simulator for modern architectures." (2016).

[15] Jung, Matthias, et al. "A Platform to Analyze DDR3 DRAM's Power and Retention Time." *IEEE Design & Test* 34.4 (2017): 52-59.

[16] Zhao, Lei, Youtao Zhang, and Jun Yang. "AEP: An error-bearing neural network accelerator for energy efficiency and model protection." *Computer-Aided Design (ICCAD), 2017 IEEE/ACM International Conference on. IEEE*, 2017.

[17] Liu, Song, et al. "Hardware/software techniques for DRAM thermal management." *High Performance Computer Architecture (HPCA), 2011 IEEE 17th International Symposium on. IEEE*, 2011

[18] Liu, et al. An Experimental Study of Data Retention Behavior in Modern DRAM Devices: Implications for Retention Time Profiling Mechanisms. *ISCA* 2013

Abstract

Thermal Simulation of High Bandwidth Memory and Mangement Technique

Woojae Shin
Department of Computer and Engineering
The Graduate School
Seoul National University

High Bandwidth Memory is 3D stacked memory using TSV. And it is known to have thermal problems due to its 3D stacked architecture. This study has created an thermal model of HBM using Hotspot and simulated thermal information based on the memory command trace of the GPU application capture with GPGPUSim. With the simulation environment, the characteristics of vertical and horizontal temperature distribution of HBM were obtained. We also simulate the temperature of the HBM with various logic die power and cooling system. As a result of the simulation, unbalanced temperature distribution between DRAM dies can be found. To

alleviate this problem, profiling of GPGPU application is done. Address mapping considering the HBM's 3D architecture is proposed to redistribute the memory access. We proposed two types of address mapping scheme depending on its redistribution level. Each address mapping scheme show 0.87℃, 1.45℃ decrement in difference of maximum and minimum temperature in HBM stack. Also maximum temperature of HBM is decreased by 7.84℃, 11.82℃ each. With these proposed address mapping schemes, refresh power can be saved up to 52% maintaining same bit error rate.

keywords : High Bandwidth Memory, thermal simulation, thermal management, GPGPUSim, Hotspot

Student Number **: 2016-25219**