웨이퍼스케일 프로세서 설계 - GPU 사례 연구

Saptadeep Pal* , 다니엘 페트리스코 † , Matthew Tomei † 및 Rakesh Kumar† Puneet Gupta‡ , Subramanian S. Iyer‡ , 캘리포니아
대학교 로스 앤젤레스 캠퍼스 컴퓨터 공학과 {saptadeep,puneetg}@ucla.edu
퓨터 공학과 ,ssiyer}@ucla.edu, {petrisk2,tomei2,rakeshk}@illinois.edu

개요 - 증가하는 통신 오버헤드는 이미 컴퓨터 시스템 확장을 위협하고 있습니다. 통신 오버헤드를 획기적으로 줄이는 한 가지 접근 방식은 웨이퍼 스케일 처리입니다.

그러나 웨이퍼스케일 프로세서 [1], [2], [3] 는 전통적인 통합 기술에 내재된 수율 문제 [1], [4] 로 인해 역사적으로 비실용적인 것으로 간주되었습니다 . 사전 제조된 다이가 실리콘 웨이퍼에 직접 결합되는 Si-IF(Silicon-Interconnection Fabric) [5], [6], [7] 와 같은 새로운 통합 기술 을 사용하면 해당 수율 문제. 따라서 waferscalar 아키텍처를 다시 검토해야 합니다. 이 백서에서는 웨이퍼 규모에서 오늘날의 아키텍처를 구축하는 것이 실현 가능하고 유용한지 연구합니다. 웨이퍼 스케일 GPU를 사례 연구로 사용하여 300mm 웨이퍼는 약 100개의 GPU 모듈(GPM)을 수용할 수 있지만 물리적 문제를 고려할 때 약 40GPM의 훨씬 축소된 GPU 아키텍처만 구축할 수 있음을 보여줍니다. 또한 웨이퍼 스케일 아키텍처의 성능 및 에너지 영향을 연구합니다. 우리는 웨이퍼스케일 GPU가 프로그래밍 모델을 변경하지 않고도 상당한 성능 및 에너지 효율성 이점(PCB에서 동등한 MCM-GPU 기반 구현에 비해 최대 18.9배 속도 향상 및 143배 EDP 이점)을 제공할 수 있음을 보여줍니다. 또한 웨이퍼 스케일 GPU 아키텍처를 위한 스레드 스케줄링 및 데이터 배치 정책을 개발합니다. 당사의 정책은 24GPM 및 40GPM 사례에 대해 각각 최대 2.88배(평균 1.4배) 및 1.62배(평균 1.11배)까지 최첨단 스케줄링 및 데이터 배치 정책을 능가합니다.

마지막으로 다이가 상호 연결된 최초의 Si-IF 프로토타입을 제작합니다.
우리는 프로토타입에서 성공적으로 연결되는 다이 간 상호 연결의 100%를 관찰합니다. Si-IF에서 다이 본딩에 대해 이전에 보고된 높은 수율과 함께 이것은 웨이퍼 스케일 GPU 아키텍처를 구축하기 위한 기술적 준비를 보여줍니다.

키워드 - 웨이퍼스케일 프로세서, GPU, 실리콘 인터커넥트 패브릭

I. 서론

새로운 애플리케이션 [8], [9], [10], [11], [12], 새로운 비즈니스 모델 [13], [14]) 및 새로운 데이터 처리 기술(예: 딥 러닝)의 출현으로, 병렬 하드웨어에 대한 필요성이 그 어느 때보다 강해졌습니다 [15,16,17]. 불행히도 똑같이 강력한 상쇄 추세가 있습니다. 병렬 하드웨어는 서로 다른 컴퓨팅 노드 간의 낮은 오버헤드 통신을 필요로 합니다. 그러나 통신의 오버헤드는 놀라운 속도로 증가하고 있습니다. 칩 간 통신을 지원하기 위해 IO 회로가 차지하는 영역 은 오늘날 일부 프로세서에서 이미 25%를 초과 합니다[18]. 이러한 IO의 전력 오버헤드는 일부 프로세서에서 30%를 초과합니다. 이러한 오버헤드는 통신 에너지, 대기 시간 및 대역폭 규모가 계산보다 훨씬 나빠짐에 따라 미래에 더 나빠질 것으로 예상됩니다 [19].

최근 작업의 대부분은 통신 병목 현상을 목표로 합니다. 집중 영역에는 에너지 비례 통신 패브릭 [20], [21], [22], 메모리 내 및 메모리 근처 계산 모델 [23], [24], [25], 통신 회피 알고리즘 [26] 및 새로운 패키징 이 포함됩니다. 2.5D 및 3D 통합과 같은 기술 [27,28]. 우리는 웨이퍼 스케일 프로세서 라는 다른 접근 방식을 탐구합니다 . 전통적으로 많은 카피

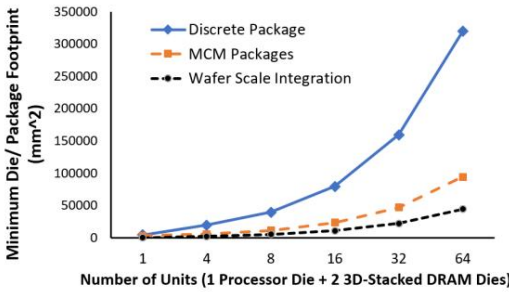
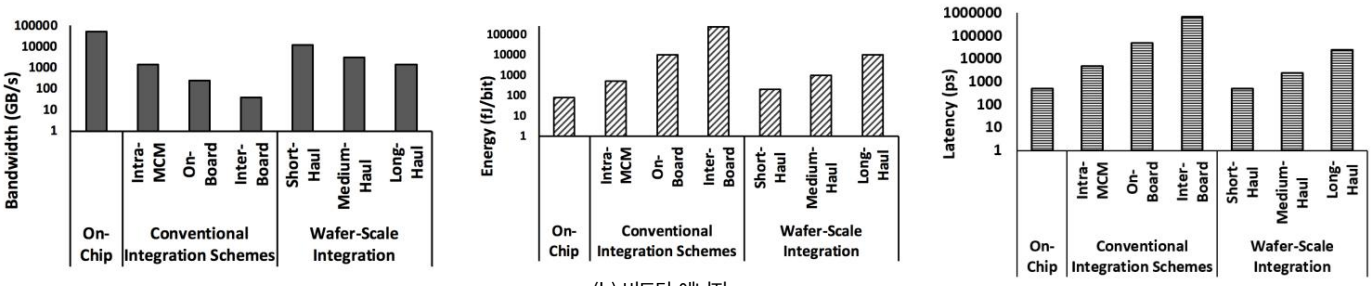


그림 1: 시스템당 프로세서 다이 수가 증가함에 따라 다양한 통합 체계에 대한 최소 다이/패키지 풋프린트가 표시됩니다.

단일 프로세서 다이의 수율에 따라 결정되는 다이의 최대 크기인 단일 웨이퍼에서 동시에 제조됩니다. 제조 후 웨이퍼는 개별 프로세서 다이로 잘려진 다음 패키징되어 인쇄 회로 기판(PCB)에서 이러한 패키징된 프로세서를 연결하는 IO 링크를 사용하여 병렬 시스템 에 통합됩니다. 반면에 웨이퍼스케일 프로세서에서는 웨이퍼가 프로세서입니다. 즉, 모놀리식 프로세서가 전체 웨이퍼 만큼 크게 설계 되거나 프로세서 세트가 웨이퍼와 프로세서 다이에 계속 상주 하도록 설계됩니다. 저비용 온웨이퍼 인터커넥트를 사용하여 웨이퍼 자체에 연결됩니다 .

웨이퍼스케일 프로세서는 면적, 성능 및 에너지 효율성 측면에서 기존 시스템에 비해 상당한 이점을 가질 수 있습니다. 그림 1은 여러 시나리오에서 컴퓨팅 다이의 총 면적 풋프린트를 보여줍니다 . 각 다이는 개별 패키지에 배치되고 , 멀티 칩 모듈(MCM) 패키지 및 웨이퍼 스케일 통합. 패키지의 면적 오버헤드는 일반적으로 상당히 높으며 고성능 시스템의 경우 패키지 대 다이 비율이 10:1 이상이 될 수 있습니다 [29]. MCM 패키지는 다이당 패키지 풋프린트를 줄이는 데 도움이 되지만 패키지 리스 웨이퍼스케일 통합은 면적당 컴퓨팅 측면에서 상당한 이점을 제공합니다. 그림 2는 웨이퍼 스케일 통합과 기존 통합 방식 및 온칩 인터커넥트에 대해 사용 가능한 통신 대역폭, 대기 시간 및 비트당 에너지를 비교합니다. 웨이퍼 스케일 통합에서 인터커넥트 는 온칩 인터커넥트와 유사한 피치를 갖습니다.

기존의 통합 방식에서는 MCM 패키지 내부의 다이 내 연결 이 미세한 와이어 피치를 가질 수 있지만 PCB 트레이스와 PCB 간 링크는 I/O가 제한됩니다. 이 con은 총 대역폭에 부담을 줍니다. 따라서 웨이퍼스케일 통합은 기존의 통합 체계가 제공할 수 있는 것보다 훨씬 더 큰 대역폭을 가능하게 합니다. 웨이퍼스케일 링크는 더 작고 밀도가 높기 때문에 상대적으로 더 낮은 주파수에서 많은 수의 링크가 실행되는 경우 간단한 병렬 통신 프로토콜을 사용할 수 있습니다 [6]. 이것은 SerDes 회로를 제거하는 데 도움이 되므로



(a) 대역폭 (b) 비트당 에너지 (c) 대기 시간

그림 2: 기존 통합 방식과 웨이퍼스케일 통합 방식에서 온칩 링크와 다양한 유형의 링크에 대한 통신 링크 대역폭, 비트당 에너지 및 대기 시간 비교.

통신의 에너지와 대기 시간을 줄입니다. 테스트 및 패키징 비용 측면 에서도 상당한 이점이 있습니다 [30].

당연히 웨이퍼스케일 프로세서는 80년대에 집중적으로 연구되었습니다. 또한 웨이퍼스케일 프로세서 [1], [4]를 구축하려는 여러 상업적 시도가 있었습니다. 불행하게도 이러한 약속에도 불구하고 이러한 프로세서는 수율 문제로 인해 주류 시장 에서 성공을 거두지 못했습니다 . 일반적으로 프로세서의 크기가 클수록 수율이 낮아집니다. 그 당시 웨이퍼 규모의 수율 은 쇠약해지고 있었습니다 [4].

우리는 그 이후로 제조 및 패키징 기술 에서 상당한 발전 이 이루어졌으며 이제 웨이퍼스케일 프로세서의 타당성을 재검토해야 할 때라고 주장합니다. 특히, 이제 사전 제조된 다이를 웨이퍼에 직접 안정적으로 접착하는 것이 가능합니다 [31,32,5]. 따라서 작고 높은 수율의 다이를 웨이퍼에 본딩하고 저비용 웨이퍼 레벨 인터커넥트(실리콘 인터커넥트 패브릭, Si- IF)를 사용하여 연결함으로써 수율이 높은 웨이퍼스케일 프로세서를 구축하는 것이 가능할 수 있습니다. 오늘날 의 높은(증가하는) 통신 오버헤드를 고려할 때 웨이퍼스케일 처리의 잠재적 이점이 훨씬 더 클 수 있다는 사실과 함께 오늘날 웨이퍼스케일 프로세서 구축의 이점과 과제 를 더 잘 이해하고 싶습니다 .

[7] 의 이전 작업에서는 Si-IF 기반 기판을 기반으로 하는 패키지리스 프로세서를 제안했으며 패키지 제거의 대역폭, 열 및 면적 이점에서 오는 상당한 성능 향상을 보여주었습니다. 그러나 이 작업 은 기존 크기의 단일 다이 프로세서 시스템 (~600mm2 , 150W)에만 초점을 맞췄습니다. 반면에 이 논문 은 300mm 웨이퍼 전체, 즉 70,000mm2 의 가용 면적 만큼 큰 GPU 시스템의 아키텍처에 초점을 맞춥니다 .

이 문서는 다음과 같은 기여를 합니다.

- 이것은 웨이퍼스케일 GPU 시스템을 구축하는 것이 실현 가능하고 유용한지를 연구하는 첫 번째 논문입니다. 우리는 새로운 통합 기술을 사용하는 300mm 웨이퍼가 약 100개의 GPU 모듈(GPM)을 수용 할 수 있지만 물리적 문제를 고려할 때 약 40GPM의 훨씬 축소된 GPU 아키텍처만 구축할 수 있음을 보여줍니다. • 다양한 물리적 제약 조건에서 웨이퍼 스케일 GPU에 대한 아키텍처 탐색을 수행합니다. 우리는 웨이퍼스케일 GPU가 전력 전달 네트워크 오버헤드로 인해 열 제약이 아닌 영역 제약을 받는다는 것을 발견했습니다. 우리는 105°C 의 접합 온도 제약 조건에 대해 300mm 웨이퍼에서 24GPM 아키텍처가 가능함을 보여줍니다 . 41GPM 아키텍처는 낮은 전압 및 주파수 에서 실행되는 각 GPM에서 4모듈 전압 스택이 허용될 때 활성화됩니다 . 우리는 또한 웨이퍼스케일 GPU 아키텍처가 링, 메시 또는 1D/2D 토러스 토폴로지로 지원될 수 있습니다. 크로스바와 같은 더 많은 연결된 토폴로지는 이러한 대형 프로세서의 배선 제한으로 인해 구축할 수 없습니다. • 우리는 웨이퍼스케일 GPU 아키텍처가 동등한 상호 연결된 개별 GPU 또는 심지어 상호 연결된 MCM-GPU 와 비교하여 많은 GPU 응용 프로그램에 대해 상당한 성능 및 에너지 효율성 이점 이 있음을 보여줍니다. 예를 들어, color [33] 는 24-GPM 및 40GPM 웨이퍼스케일 GPU에 대해 동등한 상호 연결된 MCM-GPU 기반 시스템에 비해 10.9배 및 17.8배의 속도 향상을 제공합니다. 모든 워크로드에서 40GPM 시스템의 평균 성능 및 에너지 효율성 이점은 각각 5.14배 및 22.5배입니다. • 웨이퍼 스케일 GPU 아키텍처에 대한 스레드 블록 스케줄링 및 데이터 배치의 영향을 연구합니다 . 배치 전략과 결합된 스레드 그룹 스케줄링 및 데이터 파티셔닝에 대한 당사의 기술 은 대규모 GPU 스케줄링 에서 최첨단 [34] 에 비해 최대 2.88배(평균 1.4배)의 성능 이점을 제공할 수 있습니다. EDP 측면에서 평균 이점은 24 및 40 GPM 시스템에서 각각 49% 및 20%입니다. • 마지막으로 상호 연결된 다이가 있는 최초의 Si-IF 프로토타입을 제시합니다. 100mm 웨이퍼 Si-IF에 대해 관찰한 다이 간의 100% 성공적인 상호 연결 10개의 상호 연결된 4mm2 다이가 있는 프로토타입 은 이전에 Si-IF에서 다이 본딩에 대해 보고된 높은 수율 과 결합되어 웨이퍼 스케일 GPU 아키텍처 를 구축하기 위한 기술적 준비 상태를 보여줍니다 .

II. 배경 및 기술 준비도

최근 몇 가지 통합 기술은 더 큰 시스템을 구축하는 데 목표를 두고 있습니다. 특히, TSMC CoWoS(인터포저 기반 솔루션) [35] 및 Intel의 EMIB [20] 와 같은 2.5D 통합 기술 을 사용하면 고대역폭, 저지연 상호 연결 기판에 여러 개의 고수익 다이를 통합하여 더 큰 시스템을 구축할 수 있습니다. 그러나 이러한 기술 에는 크기 제한이 있습니다. 인터포저는 얇은 실리콘을 사용 하므로 깨지기 쉽습니다. 따라서 인터포저 기반 시스템의 크기는 일반적으로 레티클의 크기로 제한됩니다. 레티클 크기 외에도 인터포저는 여러 개의 레티클을 스티칭하여 제작되는데, 이는 비용이 많이 들고 복잡한 공정이며 수율이 낮 습니다 [36,37]. 결과적 으로 오늘날 가장 큰 상용 인터포저 [38] 는 크기가 약 1230mm2 이고 단 하나의 GPU와 4개의 메모리 스택만 수용합니다. Intel의 EMIB 기술도 인터커넥트 기판에 약 5-10개의 다이만 통합할 수 있습니다 [20].

더 큰 시스템 을 가능하게 할 수 있는 또 다른 유망한 통합 기술은 Si-IF(Silicon Interconnect Fabric) [5], [6], [7]입니다. Si-IF는 유기 인쇄 회로 기판(PCB)을

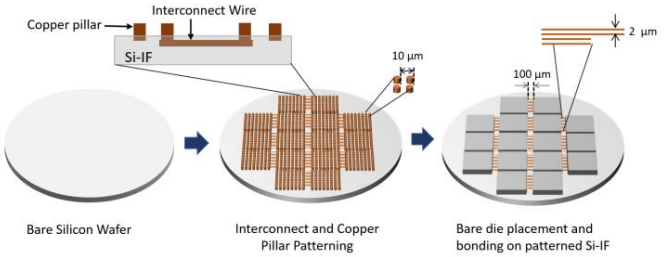


그림 3: 시스템 조립 프로세스 흐름이 표시됩니다. 베어 실리콘 웨이퍼를 처리하여 상호 연결 레이어와 구리 기둥을 만듭니다. 그런 다음 TCB를 사용하여 베어 다이를 웨이퍼에 접합합니다.

구리 기둥 기반 I/O 핀 을 사용하여 베어 실리콘 다이를 두꺼운 실리콘 웨이퍼에 직접 배치하고 본딩할 수 있습니다. 더 작고 높은 수율의 다이는 높은 수율을 보장하기 위해 성숙한 제조 기술을 사용하여 수동 상호 연결 기판(웨이퍼)에서 상호 연결됩니다. 프로세서, 메모리 다이와 주변 장치, VRM, 심지어 패시브(인덕터 및 커패시터)와 같은 비컴퓨팅 다이와 같은 다양한 시스템 구성 요소는 개별 구성 요소에 대한 패키지 요구 사항 없이 Si-IF에 직접 본딩될 수 있습니다 [19] , [5], [7].

그림 3은 Si-IF에서 시스템 어셈블리의 공정 흐름을 보여줍니다.
패키지 가 없기 때문에 다이 사이의 짧은 채널과 결합된 구리 기둥 을 사용하여 실리콘 웨이퍼에 실리콘 다이를 직접 결합 하면 Si-IF가 훨씬 더 성능이 뛰어나고 에너지 효율적인 통신을 달성할 수 있습니다(그림 3).

1

Si-IF는 실리콘 다이(및 기타 구성 요소)가 실리콘 웨이퍼에 직접 통합되기 때문에 웨이퍼스케일 통합을 위한 확실한 후보입니다. 그러나 웨이퍼 스케일 공정을 가능하게 하려면 Si-IF가 높은 시스템 수율을 제공해야 합니다. 아래에서는 Si-IF가 웨이퍼 스케일 통합에 필요한 높은 시스템 수율을 제공할 것이라고 주장합니다.

Si-IF 기반 웨이퍼 스케일 시스템의 수율에는 다이 수율, 구리 필러 본드 수율 및 Si-IF 기판 수율의 세 가지 구성 요소가 있습니다. KGD(Known-Good-Die) 테스트 기술 [39,40] 을 사용하여 Si-IF에서 조립에 사용할 다이를 미리 선택함으로써 다이에 대해 높은 수율 (>99%)을 보장할 수 있습니다 . 구리 기둥 본드의 수율도 99%에 근접할 것으로 예상됩니다. 구리 기둥 기반 I/O 오류의 기본 모드는 개방입니다. 구리 기둥은 납땜 기반 연결과 달리 돌출되기 쉽지 않으므로 단락이 불가능 합니다. 또한 기판과 다이가 모두 실리콘으로 만들어 졌기 때문에 큰 온도 변동으로 인해 구리 기둥 본드 에 큰 응력을 유발하는 열팽창 불일치 의 계수가 없습니다. 또한 프로토타입에 사용된 위치 및 결합 도구의 오정렬은 1μm 미만인 반면 기둥 간격은 5μm입니다. 따라서 단락 또는 오정렬로 인한 I/O 오류는 발생하지 않습니다. 사실, 이전 작업에서는 실제로 구리 기둥 수율이 99%보다 높은 것으로 관찰되었습니다 [5], [7]. 또한 미세 피치 구리 필러 (<10μm) 는 솔더 기반 연결 (>50μm 피치)을 사용 하는 오늘날의 통합 방식보다 최소 25배 더 많은 I/O를 허용하므로 리던던시 를 사용하여 시스템 수율을 개선할 수 있습니다. 영형. 또한, 네트워크 수준 복원력 기술 [41,42] 은 시스템 수율을 향상시키기 위해 웨이퍼의 결함이 있는 다이 및 상호 연결 주위로 데이터를 라우팅하는 데 사용할 수 있습니다 .

1솔더 기반 마이크로 범프와 같은 다른 본딩 기술도 사용할 수 있습니다 . 구리 기둥 을 사용하는 대신 Si-IF와 함께 솔더 기반 마이크로범프를 사용하는 것의 트레이드 오프 에는 더 거친 피치(25μ vs 5μm, 더 높은 전기 저항 (11-13μΩ-cm vs 1.7μΩ-cm), 더 높은 금속간화합물 및 피로 관련 가능성이 포함됩니다. 파손 및 용이한 탈착성(220-230°C 대 1000°C에서).

마지막으로, Si-IF 기판의 수율은 높을 것입니다 (> 90%) . 두꺼운 인터커넥트 와이어 (2um 폭, 4um 피치) 만 있고 능동 장치가 없는 수동 웨이퍼이기 때문입니다2 . 업계 표준 수율 모델링 방정식 1과 2 를 사용 하여 2μm 와이어 폭과 간격 으로 다양한 금속층 수와 금속층 활용 에 대한 Si-IF 기판(표 I 참조)의 예상 수율을 계산 합니다.

삼 적은 수의 금속층에 대한 기판 의 계산
4
된 항복 값 은 높습니다.

$$\text{수율} = (1+) \frac{D0 \times Fcrit \times \text{면적}}{\alpha} - \alpha \quad [44],[43] \qquad (1)$$

$$F_{\text{옆이 있는 차명타}} = \int_{0}^{\infty} (2r \quad p/2)^{*} \frac{2rc}{3r_{-}} dr = F_{\text{짧은 차명타}} \quad [45] \qquad (2)$$

표 1 : 상이한 수의 금속층에 대한 Si-IF의 수율

Si-IF 금속층 활용도(%) 1 10 20			

이전 Si-IF 프로토타입은 여러 다이에서 연결을 설정하지 않았습니다 . 따라서 Si-IF 기판의 수율 또는 상호 연결된 다이가 있는 Si-IF 기반 시스템 에 대한 시스템 수율에 대한 측정이 없었 습니다. Si-IF에서 다이간 상호 연결 의 실행 가능성을 평가하기 위해 우리 는 100mm 웨이퍼스케일 Si IF에서 연결성 테스트 다이렛을 결합한 프로토타입을 제작했습니다. 구리 기둥은 그림 4와 같이 다이렛 내부와 다이렛을 가로질러 구불구불한 방식으로 연결됩니다. 그림 5는 프로토타입의 현미경 사진을 보여줍니다. 2mm×2mm 크기의 dielet의 각 행 에는 200개의 구리 기둥(총 40,000개의 기둥)이 구불구불한 구조를 사용하여 연결되어 있습니다. 우리는 각각 4mm2인 5x2 다이렛 배열을 연결합니다 . Si-IF가 실제로 높은 수율로 많은 다이렛에 연결성을 제공 할 수 있는지 확인하기 위해 다이 전체에 전기적 연결성을 테스트했습니다 .

우리의 전기 테스트는 이 프로토타입의 인터커넥트의 100%가 연결되었음을 보여주며 Si-IF의 다이간 인터커넥트뿐만 아니라 구리 기둥 의 매우 높은 수율을 보여줍니다 .
본딩 후 열 주기 테스트는 온도 변화가 구리 기둥 본드에 미치는 영향을 테스트하기 위해 -40°C에서 125°C까지 수행되었으며 그 결과 모든 구리 기둥과 상호 연결 이 본드 접촉에서 눈에 띄는 열화 없이 열 주기를 견뎌냈습니다. 저항. Si-IF에서 다이 본딩에 대해 이전에 보고된 높은 수율과 함께 이 프로토타입에 대해 관찰한 높은 수율 은 웨이퍼 스케일 시스템 구축을 위한 기술적 준비 상태를 보여줍니다.

후속 섹션에서는 GPU 사례 연구를 사용하여 a) 실현 가능한 웨이퍼스케일 아키텍처의 공간을 탐색하고, b) 웨이퍼스케일 아키텍처의 성능 이점을 이해하고 최대화하며, c) 웨이퍼스케일 GPU를 위한 데이터 배치 및 스레드 스케줄링 전략을 개발합니다.

2반도체 제조에서 수율 손실의 대부분은 트랜지스터 층(front-end-of-the-line)과 작은 피치(≤200nm)의 처음 몇 개의 금속 층에서 발생합니다.

3D0 는 mm2 당 결함 밀도입니다. , α 는 결함 클러스터링 계수이며 계산을 위해 각각 ITRS 값 2200과 2 [43] 를 사용합니다. rc 는 임계 결함 크기, p 는 상호 연결 피치, Fcrit 는 결함이 발생하기 쉬운 임계 영역 의 비율 입니다.

4인터 다이 인터커넥트는 조밀한 간격의 다이 사이에서만 실행되기 때문에 인터커넥트 영역의 양은 전체 웨이퍼 영역 의 10% 미만이 될 것으로 예상됩니다 .

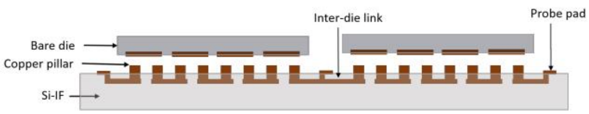


그림 4: 두 개의 서로 다른 다이의 구불구불한 구조 사이의 상호 연결이 있는 프로토타입의 개략도

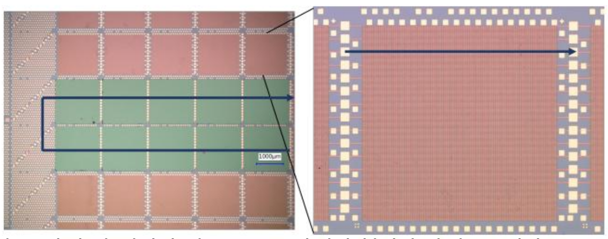


그림 5: 다이 간 연결이 있는 프로토타입의 현미경 사진. 10개의 4mm2 다이가 접착되어 다이 전체에서 신호의 연속성을 테스트합니다. 각 다이에는 회로도에 표시된 대로 구불구불한 구조의 행이 있습니다. 40,000개의 구리 기둥이 있는 Si-IF의 확대 사진도 표시됩니다.

III. 웨이퍼 스케일 GPU 아키텍처 사례

GPU 애플리케이션은 많은 양의 병렬성을 갖는 경향이 있습니다 [46,47]. 이와 같이 GPU 하드웨어 병렬 처리는 계속 증가 [34,48] 하며 냉각 및 수율에 의해서만 제한됩니다.
사실, 물리 시뮬레이션, 선형 대수학 및 기계 학습 영역의 많은 종류의 응용 프로그램은 100 개의 GPU에서 일상적으로 실행됩니다. 이러한 응용 프로그램은 다중 GPU 접근 방식이 프로그래밍 가능하기 때문에 GPU의 응용 크기를 늘리면 큰 이점 을 얻을 수 있습니다 [52]. 및 통신 오버헤드 [34]. MCM-GPU [34]와 같은 제안은 GPU의 유효 크기를 늘리려고 시도하지만 기존 통합 기술의 크기 제한에 의해 제한됩니다.

표 II: GPU 토폴로지

	스케일아웃 SCM-GPU	스케일아웃 MCM-GPU	웨이퍼스케일 GPU
GPM당 CU GPM당	64	64	64
L2 캐시 4MB 1.5TB/s 100ns 6pJ/비트 1		4MB	4MB
D램(HBM)		1.5TB/s 100ns 6pJ/비트	1.5TB/s 100ns 6pJ/비트
패키지당 GPM		4(링버스)	전체(메시)
GPM 상호 연결	없음	1.5TB/초 56ns 0.54pJ/비트	1.5TB/s 20ns 1.0pJ/bit
패키지 토폴로지	망사	망사	전체 시스템 패키지
패키지 상호 연결	256GB/s 96ns 10pJ/비트	256GB/s 96ns 10pJ/비트	없음

우리 는 표 II에 설명된 고도 병렬 GPU 시스템의 세 가지 구성을 평가 했습니다. 우리는 이러한 구성에서 가장 작은 하드웨어 장치를 GPM(GPU 모듈)이라고 생각하며, 이는 오늘날 사용 가능한 대형 GPU와 3D-DRAM 다이가 결합된 것과 거의 같습니다. 각 GPM에는 200W의 TDP와 GPU 다이 의 경우 500mm2 의 면적, 그리고 2개의 3D 스택 DRAM 다이의 경우 70W 및 200mm2 의 면적이 있습니다. 웨이퍼스케일 케이스 의 GPM 간 통신 에너지는 MCM-GPU의 온패키지 GPM 간 통신 에너지보다 높습니다 . 이것은 GPM이 평면도에서 죽기 때문입니다.

5E.g.NWChem, PSDNS, MILC, NAMD, QMCPAK, Chroma, GAMESS, MELD, AMBER 등은 Blue Waters [49], [50], [51] 의 3072 GPU에서 일상적으로 실행됩니다 . Nvidia는 창고 규모의 GPU 애플리케이션 을 위한 HPC 컨테이너도 제공합니다.

우리는 고려(섹션 IV-D 참조) 가 웨이퍼의 DRAM과 전압 조정기 모듈(VRM)에 의해 분리되므로 GPM 간 거리는 약 ~20mm 대 2-5mm입니다.

MCM 패키지(여기서 VRM은 일반적으로 패키지 외부 또는 다이 사이가 아니라 패키지 주변에 있음)

우리가 고려하는 첫 번째 구성은 각 GPM이 자체 패키지에 포함된 ScaleOut SCM-GPU(단일 칩 모듈 GPU)입니다. GPM은 기존 PCB의 2D 메시에 배치되어 QPI와 유사한 대역폭, 대기 시간 및 에너지 특성을 가진 패키지 간 링크를 통해 연결됩니다. 두 번째는 MCM-GPU 장치가 QPI와 같은 링크로 연결된 기존 PCB 의 2D 메시에 배치되는 MCM-GPU 의 확장인 ScaleOut MCM-GPU 입니다. 우리가 평가 하는 마지막 아키텍처는 Si-IF [6] 를 통해 연결된 GPM의 2D 메시를 포함하는 단일 웨이퍼 인 가상의 웨이퍼 규모 GPU (즉, 열 또는 전력 전달 제약을 고려하지 않음) 입니다. GPM 은 프로그래머의 관점 에서 단일 논리 GPU를 구성합니다 .

그림 6, 7은 SRAD와 Backprop의 두 가지 벤치마크에 대한 ScaleOut SCM-GPU 또는 ScaleOut MCM-GPU 접근 방식에 비해 웨이퍼스케일 GPU의 잠재적 이점을 보여줍니다 . 이 두 가지 애플리케이션은 의료 이미징 및 머신 러닝을 대표하도록 선택되었으며 , 두 분야 모두 웨이퍼 스케일 처리에서 상당한 이점을 얻을 것으로 예상됩니다. 우리의 시뮬레이션은 gem5-GPU [54] 를 사용하여 메모리 추적 및 활동 프로필을 생성 하여 수행되며, 이를 추적 기반 GPU 시뮬레이터에 제공합니다. 섹션 VI에서 실험 방법론을 확장합니다 .

Backprop 의 경우 단일 GPM 시스템에서 64GPM 웨이퍼스케일 GPU의 속도가 47.54배 향상되었습니다. 속도 향상은 최고 성능의 ScaleOut SCM-GPU 및 ScaleOut MCM-GPU 구성에 비해 각각 20.8배 및 21.13배입니다. 속도 향상 은 결국 메모리 전송 대기 시간에 의해 제한됩니다.

이러한 속도 향상은 다른 ScaleOut 시스템 통합 체계에서 달성되는 속도 향상과 달리 프로그래밍 모델을 변경할 필요 없이 달성됩니다.

ScaleOut SCM-GPU 및 ScaleOut MCM-GPU에 비해 waferscale GPU의 이점은 EDP를 고려할 때 더욱 분명해집니다. 64GPM Waferscale GPU는 단일 GPM에 비해 EDP가 31.54배 감소하고 하향 추세인 반면, ScaleOut MCM-GPU 및 ScaleOut SCM-GPM 시스템은 EDP가 9GPM을 넘어 증가합니다.

SRAD 의 경우 단일 GPM 시스템 에서 64GPM 웨이퍼스케일 GPU에 대해 42.56배 속도 향상이 관찰 됩니다. 이는 각각 3.57x 및 3.65x 속도 향상에서 포화되는 ScaleOut SCM-GPU 및 ScaleOut MCM-GPU와 비교됩니다. 또한 비용이 많이 드는 패키지 간 통신을 피함으로써 64GPM 웨이퍼스케일 GPU는 EDP의 24.88배 감소를 관리합니다. 이는 추가 GPM이 실제로 EDP를 증가시키는 ScaleOut MCM-GPU 및 ScaleOut SCM-GPU와 뚜렷한 대조를 이룹니다 .

위의 결과 는 GPU 애플리케이션의 성능 및 에너지 효율성 스케일링 이 동급의 상호 연결된 개별 GPU 또는 상호 연결된 MCM-GPU 보다 웨이퍼 스케일 GPU에서 훨씬 강력 하기 때문에 GPU 아키텍처가 웨이퍼 스케일에서 구축하기에 적합하다는 것을 보여줍니다 . 다음 섹션에서는 실현 가능한 웨이퍼 스케일 GPU 아키텍처의 공간을 탐색합니다.

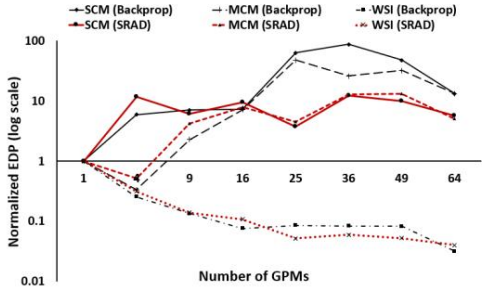


그림 6: Backprop 및 SRAD에 대한 정규화된 EDP

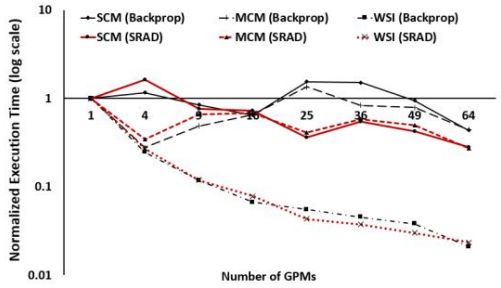


그림 7: Backprop 및 SRAD의 정규화된 실행 시간

IV. 웨이퍼 스케일 GPU 설계

웨이퍼스케일 GPU를 설계하는 것은 웨이퍼스케일 프로세서의 물리적 제약으로 인해 고유한 문제 입니다. 웨이퍼스케일 GPU 아키텍처는 킬로와트 전력에서 작동해야 합니다. 해당 아키텍처는 관련 열 및 전력 공급 문제가 있는 경우 실행 가능해야 합니다. 마찬가지로 웨이퍼 스케일 GPU는 막대한 상호 연결 리소스가 필요합니다(GPM 간의 연결 필요성으로 인해). 웨이퍼 스케일 GPU 아키텍처는 웨이퍼 스케일 수준에서 실현 가능한 네트워크 토폴로지를 지원해야 합니다.

이 섹션에서는 열, 전력 공급 및 연결 제약 조건이 있는 경우 실현 가능한 웨이퍼 규모 GPU 아키텍처를 식별하려고 시도합니다. 우리의 분석에서는 TDP가 각각 200W와 70W인 500mm2 GPU 다이와 200mm2 DRAM 다이 로 구성된 GPM 모듈을 고려합니다6 .

A. 열 제약 하의 웨이퍼스케일 GPU 아키텍처

이 하위 섹션에서는 다음과 같은 질문을 합니다. 대상 최대 접합 온도와 강제 공기 냉각 이 주어지면 300mm 웨이퍼에 수용할 수 있는 최대 GPM 수는 얼마입니까?

최대 허용 TDP를 결정하기 위해 강제 공기 대류 냉각으로 원형 300mm 웨이퍼를 덮는 1개 또는 2개의 정사각형 방열판을 사용하여 시스템을 냉각 한다고 가정합니다 . 그림 8은 웨이퍼에 부착된 두 개의 방열판의 개략도를 보여줍니다. 하나는 다이 바로 위에 있고 다른 하나는 웨이퍼 뒷면에 있습니다. 보조 방열판은 웨이퍼에 대한 기계적 지원을 제공할 뿐만 아니라 열 추출 효율을 높이는 데도 도움이 됩니다. 열 저항 모델은 그림 8에 나와 있습니다. 열 모델링 및 분석은 R-tools [55] 의 상용 CFD 기반 열 모델링 도구를 사용하여 수행됩니다. CFD는 도구에 사용되는 단순한 스파이스 기반 모델보다 더 정확한 결과 를 제공하는 것으로 알려져 있습니다. 핫스팟 [56] 과 같은 .

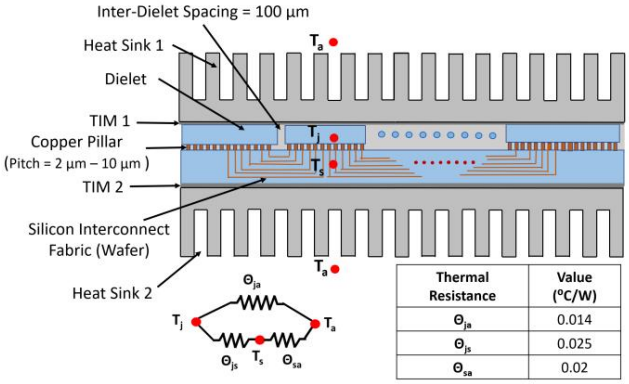


그림 8: 열 저항 모델과 함께 Si-IF의 웨이퍼스케일 시스템의 개략 단면도. Ta, Tj, Ts는 각각 주변, 칩 접합 및 실리콘 기판 온도를 나타냅니다. 다이에 직접 부착된 두 개의 방열판을 덮는 다른 후면 방열판이 표시됩니다. 방열판이 있는 300mm2 웨이퍼 스케일 시스템 의 열 저항 값 도 표시됩니다 [55].

70,000mm2 중 20,000mm2는 외부 연결 및 기타 인터페이싱 다이에 사용될 것이라고 생각 합니다 .

최대 TDP 추정을 위해 여러 열원(GPM 및 DRAM)이 50000mm2 크기의 표면에서 열을 발생시키는 것으로 간주합니다.

표 III에서는 위에서 설명한 다양한 시나리오에 대한 지속 가능한 TDP를 보여줍니다. 또한 전압 조정기 모듈(VRM) 이 있거나 없는 해당 열 예산 내 총 GPM 수를 제시합니다. VRM을 고려하지 않는 경우 유일한 열원은 GPM 모듈입니다. VRM이 웨이퍼에 배치되는 경우 VRM 비효율 로 인해 추가적인 열 손실이 발생 합니다. 여기서 우리는 on-Si-IF VRM이 약 85%의 효율을 가진다고 가정합니다 [59]. 따라서 효과적으로 VRM은 GPM당 48W의 추가 전력 손실을 초래합니다.

3개의 서로 다른 접합 온도 (Tj) 에 대해 하나의 방열판과 2개의 방열판을 모두 평가했습니다 . 전통적으로 85°C [57], [17] 및 105°C [58] 는 신뢰할 수 있는 Tj 로 사용됩니다 . 최신 방열판 솔루션을 시뮬레이션할 수 없었기 때문에 공정한 비교를 위해 프레임워크를 사용하여 [34] 에 설명된 최근에 게시된 다중 GPM 시스템(MCM-GPU)의 열 시뮬레이션을 수행했습니다 . 주변 온도가 25°C인 패키지 크기 77mm×77mm 의 방열판을 고려하면 Junction 온도는 121°C 입니다 .

결과적으로 Tj=120 ◦ 에 대해서도 분석합니다 .

우리의 분석 에 따르면 계산 목적으로 웨이퍼에서 50000mm2의 영역을 사용할 수 있지만 (~ 71GPM) 열 제한으로 인해 Si-IF에 배치할 수 있는 최대 GPM 수가 훨씬 더 낮아집니다. 이중 방열판 솔루션을 고려할 때 VRM에서 전력 손실이 없다고 가정하면 최대 34GPM을 지원할 수 있으며, 그렇지 않으면 숫자가 29GPM으로 줄어듭니다.

B. 전력 공급을 고려한 웨이퍼스케일 GPU 아키텍처

웨이퍼스케일 GPU 시스템은 방열판 기술에 의해 최대 약 9.3kW의 총 TDP로 제한됩니다. 고려하면

63D 스택 메모리는 웨이퍼 스케일에 필요하지 않습니다. 평면을 사용하여 메모리 다이는 단위 면적당 용량과 대역폭을 감소시킵니다.

7A 웨이퍼스케일 GPU는 하나 이상의 루트 컴플렉스에 연결된 다중(예: PCIe) 포트를 사용하여 외부 세계와 인터페이스할 수 있습니다.

표 III: 서로 다른 접합 온도에 대해 지원 가능한 GPM의 수

표적 접합 온도 (◦ C)	이중 방열판			단일 방열판		
	힘 (원)	숫자 없는 GPM VRM	숫자 VRM이 있는 GPM 29	힘 (원)	숫자 없는 GPM VRM 25	숫자 VRM 21 이 있는 GPM
120	9300	34	24 18	6900		
105	7600	28		5400	20	17
85	5850	21		4350	16	14

정격 TDP가 시스템 의 피크 전력의 0.75배 [60], [61] 가 되도록 하려면 전력 분배 네트워크(PDN)가 최대 12.5kW의 전력을 제공할 수 있어야 합니다 (최신 서버 보드의 경우 1-2kW에 비해). [62], [63], [64]) 피크 전력에서도 합리적인 효율성을 제공 합니다.

우리는 모든 GPM에 대해 효율적인 벅 컨버터를 사용하여 부하점(POL) 전력 변환이 있는 웨이퍼에 대한 48V, 12V, 3.3V 및 1.2V의 외부 전원 공급 대안을 탐색 합니다. 즉, GPM당 하나의 VRM이 있습니다. 일반적으로 입력 전압이 높을수록 PDN 회로 오버헤드가 커지지만 표 IV 및 V에 표시된 것처럼 전력을 공급하는 데 필요한 레이어 수가 적어 집니다(저항 손실도 낮아짐).

표 IV: 필요한 레이어 수 vs 웨이퍼에 대한 공급 전압. 10µm 두께의 금속은 RF I 2R 을 지원하는 대부분의 기술에서 사용할 수 있습니다. 입력 전압(V)

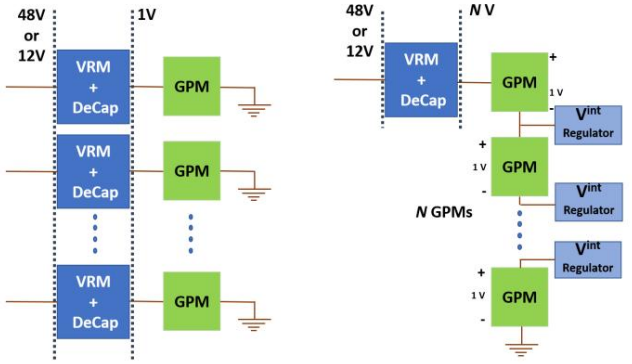
	손실(W)	레이어 수		
		두께 = 10µm	두께 = 6µm 68 16 8 4 2	두께 = 2µm
1	500	42		202
3.3	200	10		44
	500	6		18
12	100	2		10
	200	2		4
48	50	2	2	2
	100	2	2	2

표 V: GPM 입력당 VRM 및 디캡 오버헤드 VRM + GPM당 디캡 영역(mm2)

전압(V)				GPM 수		
	스택 없음 2단 4단 스택 없음 2단 4단 1 3.3 12 48					
	300	-	-	50	-	-
	1020	610	-	29	38	-
	1380	790	495	24	33	41
	2460	1330	765	15	24	34

우리는 [65] 에 주어진 전력 분배 메쉬 크기 조정 모델 을 사용하여 최소 면적과 전력 분배에 필요한 레이어 수를 결정합니다. 표 4에서 볼 수 있듯이 웨이퍼에 1V 및 3.3V를 공급하는 데 필요한 금속 층의 수가 매우 큰 I2R 손실의 경우에도 매우 많다는 것을 알 수 있습니다.
더욱이, 전력 전달을 위한 4개 이상의 금속층은 비용 및 제조 가능성 때문에 바람직하지 않다. 따라서 실행 가능한 유일한 옵션은 12V 또는 48V 외부 전원 입력입니다.
두 가지 외부 전원 입력 옵션(12V 및 48V)을 비교하기 위해 필요한 인덕턴스와 정전 용량 으로 인해 DC-DC 변환 및 조정을 위한 VRM의 크기가 매우 클 수 있음을 인식했습니다 . 합리적으로 높은 전력 변환 효율 (>90%) 에서 최신 48V-1V 변환기의 면적 효율은 1W/10mm2 - 1W/5mm2 (VRM 인덕터 포함) [66], [59] 범위입니다. PCB 기반 통합 을 사용하여 구현 되었습니다. 우리는 보수적 으로 48V에서 1V로의 전력 변환을 위해 VRM의 1W/6mm2 영역 오버헤드를 가정 합니다.

즉, 70W 3D 스택 로컬 DRAM(예: 최대 전력 360W)과 함께 TDP 200W로 GPM을 지원 하려면 48V-1V 옵션의 VRM 영역이 약 2160mm2 (360×6)가 됩니다! 우리는 또한



(a) PDN당 하나의 VRM 그림 (b) 누적 전압 공급

9: PDN 체계

1MHz [67] 의 주파수 에서 약 50A의 전류 부하 변동을 보상하는 데 필요한 표면 실장 디커플링 커패시터의 오버헤드 영역은 ~ 300mm2 입니다. 따라서 48V에서 1V로 변환 전략의 경우 웨이퍼 에서 사용 가능한 300mm2 있는 총 GPU 수 는 약 15개에 불과합니다 . 소스는 ~ 6.5kW 입니다. 즉, 85% VRM 효율성을 가정하고 I2R 손실을 설명하는 TDP는 4.9kW입니다. 따라서 시스템은 영역이 제한 되고 TDP가 제한되지 않습니다(최대 허용 TDP 가 약 9.3kW이므로). 이것은 단순히 전압 변환의 면적 효율 을 개선함으로써 웨이퍼 스케일 컴퓨팅에서 훨씬 더 높은 성능과 에너지 효율이 가능할 수 있음을 시사한다는 점 에서 두드러진 결과입니다.

12V 옵션의 경우 VRM의 크기가 더 작아질 것으로 예상됩니다 (~ 1W/3mm2). 따라서 약 24개의 GPM이
이제 총 약 10.3kW의 최대 전력(7.8kW TDP)에 해당하는 용량이 수용됩니다.
그러나 120°C Tj 의 열 제한에 따라 최대 29개 GPM을 수용할 수 없습니다 .

따라서 48V 공급의 경우와 마찬가지로 시스템은 열적으로 제한되지 않고 면적이 제한됩니다.
대체 전원 분배 전략은 그림 9와 같이 여러 GPU [68,69,70] 의 전압 스태킹입니다. N 개의 GPM이 스태킹된 경우 스택에 대한 공급 전압은 필요한 공급 전압의 N 배여야 합니다. 하나의 GPM과 동일한 전류가 적층된 GPM을 통해 흐릅니다. 이전 작업 [70] 에서 볼 수 있듯이 이 접근 방식은 인접한 GPM이 임의의 시간 간격에서 유사한 활동 및 전력 소모를 가질 것으로 예상되기 때문에 GPM 컨텍스트에서 실행 가능 합니다(좋은 데이터 배치 및 스케줄링 정책도 도움이 될 수 있음). 이제 N 개의 전압 조정기(GPM당 하나) 를 사용하는 대신 1/N 변환 비율을 가진 하나의 VRM이 N 개의 GPM 에서 공유 됩니다. 변환 비율이 감소함에 따라 동일한 효율을 위해 VRM 모듈의 크기를 줄일 수 있습니다. 완벽하게 균형 잡힌 스택은 안정적인 중간 노드 전압을 보장하지만 중간 노드 의 안정성 을 보장하기 위해 경량 전압 조정기(예: 정적 전력 손실을 최소화하면서 중간 레일 잡음을 줄일 수 있는 푸시-풀 조정기 [71]) 를 사용할 것으로 예상합니다. . 이러한 중간 전압 회로에 대한 자세한 내용은 [70]에 나와 있습니다.

이러한 중간 전압 노드 조정기는 전압 변환이 아닌 작은 전류 수요를 안정화하는 역할만 담당하므로 소형 스위치드 커패시터(SC) 기반

표 VI: 제안된 PDN 솔루션

대상 Junc. 온도 (◦ C)	이중 방열판			단일 방열판		
	열의	공급 전압(V)/리미트(W) # 스택당 GPM	최고 의 GPM 수	열의	공급 전압(V)/리미트(W) # 스택당 GPM	최고 의 GPM 수
120	9300 48/4	또는 12/2 7600	명목상 29	6900 48/2	또는 12/1 5400	명목상 21
105	48/2 또는 12/1 5850 48/2 또		24	48/2 또는 12/1 4350		17
85	는 12/1		18		48/1	14

표 VII: 스택당 12V 공급 및 4-GPM을 사용하는 41GPM의 작동 전압 및 주파수

대상 Junc. 온도 (◦ C)	이중 방열판			단일 방열판 작동 작동		
	GPM 전력(W) 125.75	운영 전압(mV) 877	작동 주파수 (MHz) 469.6	GPM 전력(W) 71.75	전압 주파수(mV)	(MHz)
120				44.75	752 364.2	664 570
105	92	805	408.2	24.5		291.4
85	51.5	689	311.7			216.2

레귤레이터 또는 선형 드롭아웃(LDO) 레귤레이터를 사용할 수 있습니다. 우리의 보수적인 추정치(경험 및 이전 작업을 기반으로 함) 는 이러한 중간 조정기가 약 200mm2의 면적 풋프린트를 가질 것이라고 제안합니다. 이를 통해 GPM 스택을 48V 전력으로 34GPM을 수용할 수 있음을 발견했습니다. 41GPM은 스택에 12V 공급 및 4GPM으로 수용할 수 있습니다. 이러한 결과는 전압 스태킹이 확장 가능한 웨이퍼 스케일 GPU 아키텍처를 가능하게 하는 유망한 기술 임을 보여줍니다 . 표 VI 는 제안된 다양한 PDN 설계 선택과 지원 가능한 GPM의 해당 수를 보여줍니다.

전압 스태킹을 사용하여 최대 41개의 GPM을 지원할 수 있지만 열 제한은 공칭 작동 조건에서 실행되는 29개의 GPM(VRM 포함)만 포장할 수 있다는 점을 기억하십시오. 따라서 우리 는 각 GPM의 공급 전압과 작동 주파수를 줄임으로써 GPM 의 수를 더욱 극대화할 수 있는 기회를 모색합니다 . 41개의 GPM을 수용하기 위해 총 전력이 최대 열 전력 예산 내에 있도록 이러한 GPM 의 작동 전압과 주파수를 찾습니다.

동일한 DRAM 전압 을 유지하면서 GPM 전압의 스케일링만 고려 했습니다. GPM 의 스케일링된 작동 전압 및 주파수는 표 VII에 나와 있습니다. GPM당 감소된 전압 을 지원하려면 VRM의 다운 변환 비율을 향상시켜야 합니다. 앞서 우리 는 12V/4V VRM의 크기를 12V/1V 변환 비율의 크기로 추정했으며, 이러한 다운 변환 비율(최대 12V/2.4V)의 증가는 이 크기의 VRM에서 쉽게 처리할 수 있습니다.

C. Waferscale GPU에 허용되는 네트워크 아키텍처

위의 분석은 GPM 모듈 간의 상호 연결을 고려하지 않습니다. 500mm2 (90mm 둘레)의 GPM 다이 크기, 4µm의 와이어 피치 및 와이어당 2.2GHz 의 유효 신호 속도 [6] (4.4GHz 신호 속도 의 접지-신호-접지)의 경우 레이어당 사용 가능한 총 대역폭 ~6TBps입니다 . 레이어 수를 늘리면 GPM 간 및 DRAM 대역폭이 증가하지만 수율은 낮아집니다8 .

[34] 에서 볼 수 있듯이 GPM에 대한 로컬 DRAM 대역폭을 1.5TBps 이상으로 늘리면 성능이 매우 적게 향상되지만 대역폭을 낮추면 상당한 성능 손실이 발생합니다. 이 DRAM 대역폭(1.5TB/s)을 염두에 두고 서로 다른 신호 금속 계층에 대해 실현 가능한 몇 가지 GPM 간 네트워크 토폴로지를 분석합니다.

[8]레이어 수를 늘리면 프로세스 복잡성이 증가할 뿐만 아니라 입자 결함에 취약한 임계 영역의 양 [72], [45]

Si-IF에 의존합니다(표 VIII 참조). 여기에서는 서로 다른 금속층 에 있는 신호선의 단락 및 개방으로 인한 수율 손실 만 고려합니다9 . GPM에서 로컬 DRAM 까지의 간격이 100-500µm 인 반면 5 ×5 GPM 배열은 약 16mm입니다. 우리는 KGD를 가정하고 구리 기둥 이중화 체계가 본딩 실패로 인한 수율 손실을 처리할 것이라고 가정합니다.

링, 메쉬 및 연결된 1D Torus의 세 가지 토폴로지는 하나의 레이어를 사용하여 구현할 수 있습니다. 2D Torus는 일부 링크가 GPM 어레이 주위로 라우팅되어야 하므로 주요 설계 및 신호 노력 없이는 하나의 금속 레이어를 사용하여 실현될 수 없습니다.

2계층 솔루션으로 이동하면 링 네트워크가 GPM 간 및 DRAM 대역폭으로 과도하게 프로비저닝되는 것을 볼 수 있습니다. 신호 레이어 수를 3개 레이어로 늘리면 보다 균형 잡힌 2D 토러스 네트워크가 가능하지만 수율이 이제 73.4% 까지 낮아질 수 있습니다.

요약하면, 수율 문제는 GPU의 총 금속 레이어 수를 제한하고 이는 다시 웨이퍼스케일 GPU에서 허용 가능한 네트워크 토폴로지 구성을 제한합니다. 이제 성능과 에너지를 극대화하면서 모든 물리적 제약 을 충족 하는 실행 가능한 웨이퍼스케일 GPU 아키텍처를 선택할 준비가 되었습니다 .

D. 전체 시스템 아키텍처 105°C 의 목

표 접합 온도에서 두 가지 구성을 고려합니다 . 하나는 1V 및 575MHz의 공칭 전압에서 실행되는 24GPM이고, 두 번째 는 805mV 및 469MHz의 감소된 전압에서 실행되는 40GPM입니다. 전자 의 경우 스택이 없는 12V 전원 공급 장치를 고려하고, 후자의 경우 스택에 4GPM이 있는 12V 전원 공급 장치를 고려합니다.

그림 11과 12에서 이 두 가지 옵션에 대한 평면도를 보여줍니다 . 각각 1GPM 과 2GPM의 중복을 고려한 25GPM과 42GPM의 평면도를 보여줍니다 .

당사의 inter-GPM 네트워크는 두 개의 금속 레이어를 사용하는 메쉬 네트워크입니다. 로컬 DRAM 대역폭과 각 GPM에 대한 GPM 간 대역폭은 1.5TBps 로 간주됩니다. GPM당 6TBps의 DRAM 대역폭은 6pJ/비트 메모리 액세스 에너지로 하나의 레이어로 지원할 수 있지만 총 DRAM 전력은 상당히 높습니다 (200W) [73]. 따라서 약 72W의 DRAM 전력이 발생하므로 1.5TBps의 DRAM 대역폭을 고려 했습니다. 이는 GPM당 가정 한 2개의 3D 스택 DRAM 모듈 [74] 의 총 TDP와 거의 같습니다.

첫 번째 경우, 스태킹이 사용되지 않는 경우 모든 GPM에는 2개의 로컬 3D 스택 DRAM 칩 세트, VRM 및 디커플링 커패시터가 있어 42mm x 49.5mm 크기의 타일을 형성합니다. 목표는 총 25개의 일반 타일을 배치하는 것입니다. 5 ×5 어레이 평면도는 300mm 원형 웨이퍼에 새길 수 있는 가장 큰 정사각형의 크기가 약 45000mm2 (~ 21 타일)에 불과하기 때문에 구현하기 어렵습니다. 따라서 이러한 25개의 타일을 배치할 수 있는 한 가지 가능한 평면도는 그림 11과 같습니다. 이 평면도는 메시 아키텍처와 매우 유사합니다.

9수율 은 섹션 II에 설명된 산업 표준 음이항 수율 모델 [45], [44], [43] 을 사용하여 계산됩니다. Inverse cubic 결함 밀도 분포 [72], ITRS [43] 에 따른 결함 밀도 값 및 4µm의 금속 피치를 사용합니다.

표 VIII: Inter-GPM 네트워크 토폴로지

레이어 수	토폴로지	메모리 대역폭(TBps) GPM 간 대역폭(TBps) 수율(%) 직경 평균 홉 길이 이등분 대역폭(TBps) 95.9 95.9 94.1 91.9 88.6 91.9 88.6					
1	반지	3	1.5		15	7.5	
	망사	3	0.75		10	4	3
	연결된 1D 토러스	3	0.5		8	3	3.75
2	반지	6	3		15	7.5	
	반지	3	4.5		15	7.5	
	망사	6	1.5		10	4	3.75
	망사	3	2.25		10	4	6 9 7.5 11.25
	연결된 1D 토러스	삼	1.5	84.3	8	~3	11.25
	2D 토러스	삼	1.125	79.6	5	~2.6	11.25
삼	2D 토러스	6	1.5	77.0	5	~2.6	15
	2D 토러스	삼	1.875	73.4	5	~2.6	18.75

코너 타일 없이. 우리는 또한 외부 인터페이스(CPU-GPU, 드라이버), 오실레이터 등을 위한 많은 시스템 레벨 블록을 수용할 System+I/O 영역을 고려했습니다.

마찬가지로 전압 스태킹이 있는 평면도의 경우 4GPM의 전압 스택마다 1개의 VRM + DeCap이 있는 42개의 다이와 3개의 중간 노드 전압 (Vint) 조정기를 배치합니다. 어레이 에 32개의 GPM을 배치하고 10개의 다른 GPM과 시스템+I/O 를 어레이 의 상단과 하단에 배치할 수 있었습니다. 메쉬 네트워크는 GPM을 연결합니다. KGD GPM, DRAM 및 VRM 다이가 I/O당 평균 수율이 99%이고 I/O 당 4개의 기둥이 있는 것을 고려하면 25 및 42 GPM 시스템의 예상 본드 수율은 각각 약 98% 및 96.6%입니다. 인터커넥트 길이에 따라 달라지는 Si-IF 기판 수율은 각각 92.3% 및 95%인데, 이는 GPM 간 와이어 가 40-GPM 평면도에서 더 작기 때문입니다. 따라서 전체 수율은 두 경우 각각 약 90.5% 및 91.8%가 될 것으로 예상됩니다 . 평면도에는 각각 25개 및 42개의 GPM이 있지만 최대 작동 GPM 수 는 열 예산으로 인해 25개 및 40개로 제한됩니다. 여분의 GPM 은 하나 또는 두 개의 GPM에 결함이 있는 경우 시스템 수율 을 개선하기 위해 여분의 GPM으로 사용할 수 있습니다 .

또한 여러 개의 웨이퍼 규모 GPU를 타일링하여 더 큰 GPU 시스템을 구축할 수 있습니다. 940mm의 웨이퍼 가장자리(300mm 웨이퍼 직경)와 외부 커넥터용으로 남은 20,000mm2 의 영역이 있다고 가정할 때 주변 의 절반이 전력을 전달하는 데 사용 된다고 가정하면 약 20개의 PCIe 소켓 커넥터를 주변에 수용할 수 있습니다 . x16 링크당 128GBps를 지원하는 PCIe 5.x를 사용하면 총 2.5TBps 의 오프 웨이퍼 대역폭을 지원할 수 있습니다.

시스템 통합: 하나의 시스템 통합 전략(그림 10)은 웨이퍼 어셈블리(습기 등으로부터 보호하기 위해 패시베이션됨)를 덮기 위해 기본 방열판과 옵션인 보조 방열판을 사용하는 것을 포함합니다. 보조 방열판을 사용하지 않는 경우 시스템을 감싸기 위해 뒷면의 단단한 금속판 이 사용됩니다. 섀시(서버/데스크탑 등) 에 설치하는 경우 일반 플러그 커넥터 또는 저강도 삽입 소켓 10을 사용하여 전체 시스템을 삽입할 수 있습니다. 또는 외부 금속 방열판을 섀시에 볼트로 고정하는 데 사용됩니다. 너비 19인치/깊이 36인치 표준 캐비닛의 행 하나에 방열판을 포함하여 2개의 300mm(12인치) 웨이퍼스케일 프로세서를 수용할 수 있는 것으로 추정 됩니다. 42U 캐비닛은 최대 6개의 행(즉, 12개의 WS-GPU)을 수용할 수 있습니다.

10실리콘은 PCB(압축 강도 3.2-3.4 GPa 대 370-400 MPa)에 사용되는 FR4 재료보다 훨씬 더 견고한 재료이며 후면 지지대(방열판 또는 플레이트 사용)를 사용하여 700 µm에서 1 mm 두께의 웨이퍼 를 플러그 커넥터의 일반적인 삽입력(수 십 MPa)을 쉽게 견딜 수 있습니다. 필요한 경우 연결을 시스템 I/O 패드에 와이어 본딩할 수도 있습니다.

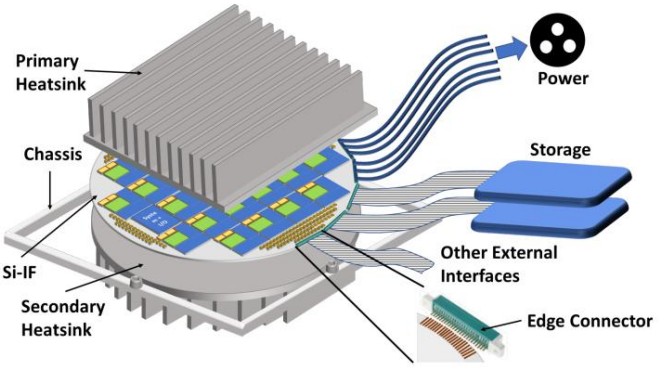


그림 10: 기본 및 후면 보조 방열판이 있는 Si-IF 시스템 어셈블리가 표시됩니다. 전체 시스템은 섀시에 볼트로 고정됩니다. 호스트 CPU는 외부에 연결되거나 웨이퍼 자체에 상주할 수 있습니다.

V. 스레드 블록 스케줄링 및 데이터 배치

웨이퍼 스케일 GPU 아키텍처의 성능 및 EDP는 컴퓨팅 및 데이터가 웨이퍼 스케일 시스템 에 어떻게 분산되는지에 따라 달라집니다 . 일반적으로 커널 실행 중에 GPU의 스레드 블록(TB)은 중앙 집중식 컨트롤러에 의해 CU 가용성에 따라 라운드 로빈 순서로 CU(컴퓨팅 유닛)로 디스패치됩니다. 그러나 이러한 세분화된 스케줄링 정책은 여러 GPM에 걸쳐 커널의 TB를 배치할 수 있습니다. 종종 연속적인 TB는 데이터 지역성에서 이점을 얻습니다. 따라서 이러한 정책은 GPM 간 네트워크에서 많은 수의 메모리 액세스가 필요하므로 웨이퍼 스케일 통합의 성능 및 에너지 이점을 파괴할 수 있습니다. 따라서 중앙 집중식 스케줄링 대신 분산 스케줄링을 사용합니다.

이 분산 정책에서는 커널의 인접한 TB 그룹 을 각 GPU에 할당하여 TB 간의 공간 데이터 집약성을 활용할 수 있습니다. 이러한 정책은 [34] 에 제시된 MCM-GPU 에 사용되었다. 연속적인 TB 그룹은 코너 GPM에서 시작하여 행을 먼저 이동 하는 GPM 어레이에 배치되었습니다 . 데이터 배치는 첫 번째 터치입니다. 즉, 특정 페이지에 대한 첫 번째 메모리 액세스가 완료되면 페이지는 메모리 참조가 만들어진 GPM의 로컬 DRAM으로 이동됩니다. 그러나 이 온라인 정책에 의해 악용될 수 없는 비이웃 TB 간에 강력한 공간 데이터 지역성이 여전히 존재할 수 있습니다. 다수의 GPM이 있는 웨이퍼스케일 GPU에서는 이웃하지 않는 GPM 간의 통신으로 인해 다중 홉 대기 시간이 높아져 성능이 저하될 수 있습니다. 따라서 데이터 액세스 대기 시간과 전체 네트워크 대역폭 사용 을 최소화하기 위해 많은 양의 데이터를 공유하는 TB를 인접한 GPM에 배치할 수 있도록 하는 정책이 필요합니다.

이 문제를 해결하기 위해 오프라인 파티셔닝 및 배치 프레임워크를 개발했습니다.

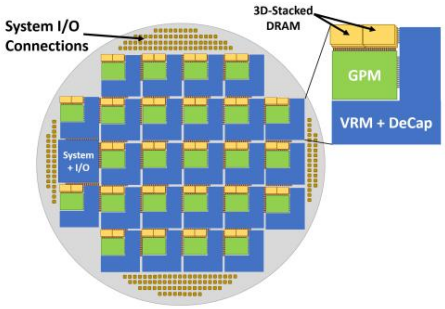


그림 11: 단위당 2개의 3D 스택 DRAM, VRM 단위 및 디커플링 커패시터로 구성된 25 GPM 단위(1개의 중복 단위)가 포함된 웨이퍼스케일 GPU.

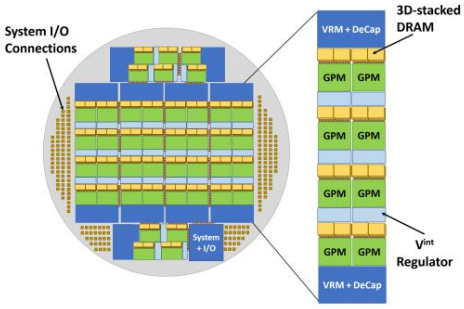


그림 12: 장치당 2개의 3D 스택 DRAM, VRM 장치 및 디커플링 커패시터로 구성된 42개의 GPM 장치(2개의 중복 장치)가 있는 웨이퍼스케일 GPU.

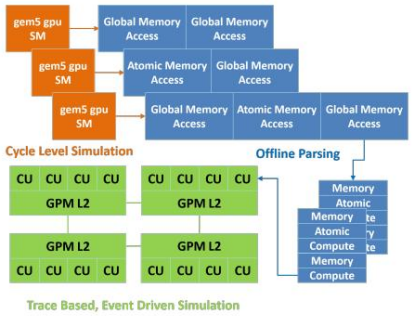


그림 13: 시뮬레이터 작업 흐름

원격 메모리 액세스 를 최소화하는 TB 및 DRAM 페이지에 대한 일정 및 GPM 할당 . 우리의 프레임워크는 완전히 자동화되어 있으며 TB-DRAM 페이지(TB-DP) 액세스 그래프를 입력으로 가져오고 TB에서 GPM으로의 매핑 및 데이터 배치를 출력합니다(그림 15 참조). TB-DP 액세스 그래프의 노드는 TB 또는 DRAM 페이지를 나타내며 TB 와 DRAM 페이지 사이의 가장자리는 특정 TB가 DRAM 페이지에 액세스함을 나타냅니다. 에지 가중치 는 총 액세스 수에 해당합니다. 이 그래프가 주어지면 목표는 분할 경계를 교차하는 가장자리 의 총 가중치 가 최소화 되도록 그래프를 k 분할로 분할하는 것입니다.

우리는 알고리즘 의 각 반복에서 N/k 노드로 하나의 파티션을 추출하는 Fiduccia - Mathhessey (FM) 분할 알고리즘 [75] 의 반복 형식을 사용하여 이 분할 문제를 해결합니다 . 구현 시 크기 비율 이 최대 ±2%까지 드리프트되도록 허용하여 파티션 절단을 추가로 최소화합니다. 이렇게 하면 파티션 전체에서 데이터 액세스를 최소화하는 응용 프로그램에 대한 해당 데이터 배치와 TB 일정이 생성 됩니다. 그러나 파티셔닝은 GPM 간 네트워크의 전체 로드를 최소화하는 문제를 해결하지 못하며 소수이지만 매우 원격(다수 홉) 액세스 가 여전히 남아 있는 경우 대기 시간 문제가 전체 성능 및 에너지 효율성에 영향을 미칠 수 있습니다. 따라서 네트워크 토폴로지 와 GPM 수가 주어진 경우 다음 단계는 TB-DP 클러스터를 이러한 GPM에 할당하는 것입니다. 이것이 클러스터 배치 문제입니다.

배치 문제의 경우 액세스 수와 액세스 소스와 대상 사이의 거리를 곱한 값인 원격 액세스 비용 메트릭 의 최소화를 고려합니다. 예를 들어, 5x5 GPM의 그리드와 (1,1) 및 (3,5) 위치의 GPM 간에 만들어진 5개의 액세스 를 고려해 봅시다 . 위치 간의 최소 맨해튼 홉 거리는 6홉입니다. 따라서 우리가 고려하는 비용은 30입니다. 총 비용은 네트워크의 총 대역폭 사용을 나타냅니다.

GPM 간 네트워크 및 홉 수 최소화는 본질적으로 액세스 대기 시간을 최소화합니다. 모의 어닐링 기반 배치를 사용하여 클러스터를 GPM 어레이의 적절한 GPM에 매핑합니다. 그림 14에서는 네트워크 토폴로지에 대한 이 비용을 40GPM 시스템에 대한 기본 런타임 동적 스케줄링 및 첫 번째 터치 페이지 배치(아래에서 설명)와 비교합니다.

우리의 오프라인 정책은 액세스 비용을 최대 57%까지 줄입니다.

파티셔닝 및 배치 정책은 공간 액세스 패턴에 따라 결정되었습니다. 시공간 액세스 패턴 에 기반한 정책 은 더 나은 최적화를 제공할 수 있지만 향후 작업으로 남겨둡니다. 게다가,

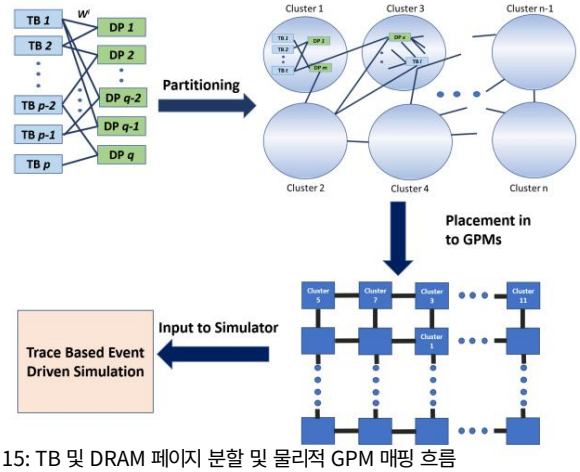


그림 15: TB 및 DRAM 페이지 분할 및 물리적 GPM 매핑 흐름

이 파티셔닝 체계에서는 로드 밸런싱을 명시적으로 고려하지 않았습니다 . 파티셔닝 알고리즘은 TB-DP 그래프 를 k개의 거의 동일한 파티션으로 나누므로 이 방식이 반드시 TB의 로드 밸런싱을 수행하지는 않습니다. 따라서 우리는 또한 정적 파티셔닝 위에 런타임 로드 밸런싱 체계를 사용합니다. 여기서 GPM의 대기열에서 대기 중인 TB(CU에 할당됨)가 있고 유휴 GPM이 있는 경우 대기 중인 TB는 가장 가까운 유휴 GPM으로 이전되었습니다.

기타 정책: 우리는 또한 첫 번째 TB 그룹 이 중앙 GPM에 배치되고 후속 그룹이 중앙 GPM에서 나선형으로 GPM 에 할당 되는 온라인 지역 인식 배치 정책 을 평가했습니다. 이 정책은 코너 GPM에서 시작하여 행을 먼저 이동하는 단순 배치 정책과 비교하여 ±3% 이내의 성능을 보였습니다.

오프라인 정책 의 경우 #access2 hop (가장 많이 연결된 TB 클러스터를 가장 가까이 배치할 수 있음) 및 hop2 #access (데이터 액세스의 최대 대기 시간 최소화) 의 합계와 같은 다른 액세스 비용 메트릭을 고려했습니다. 그러나 이러한 메트릭에서 생성된 배치는 #access*hop 메트릭에 비해 평균적으로 2% 낮은 성능을 보입니다. 단 , 불규칙한 애플리케이션이고 네트워크 대기 시간이 한정된 색상 에 대해 24-GPU 시스템에서 #access * hop2 를 사용할 때 7%의 이점 이 있습니다. . 따라서 섹션 VII에서는 단순 배치가 있는 온라인 정책과 클러스터 배치에 대한 데이터* 홉 메트릭이 있는 오프라인 정책에 대해서만 논의하고 분석합니다.

표 IX: 벤치마크

기준	모음곡	도메인
역전파	Rodinia 기계 학습 핫스팟	
	Rodinia 물리 시뮬레이션 lud	
	로디니아	선형 대수 입자 필터
순진한 Rodinia		의료 영상 분야
	로디니아	의료 영상 색상
	판노티아	그래프 색칠 BC
	판노티아	소셜 미디어

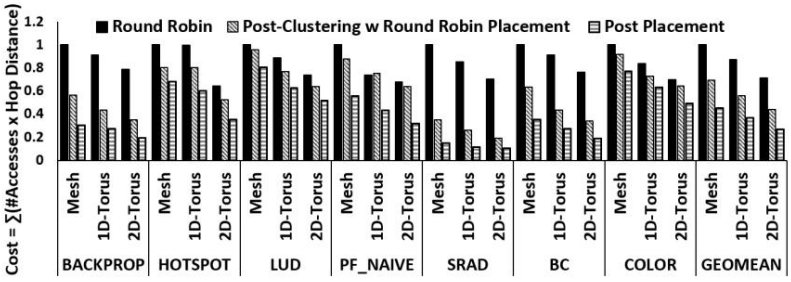


그림 14: 오프라인 파티셔닝 및 GPM 배치로 인한 액세스 비용 메트릭의 개선이 표시됩니다. 베이스라인은 데이터 위치 인식 분산 스케줄링 및 첫 접촉 데이터 배치입니다.

VI. 방법론

gem5 [76] 및 GPGPU-Sim [77] 과 같은 현재의 아키텍처 시뮬레이터 는 최대 12개의 컴퓨팅 유닛이 있는 시스템을 연구하도록 설계되었습니다 . 합리적인 시간 프레임에서 웨이퍼스케일 프로세서를 시뮬레이션할 수 없습니다. 웨이퍼 스케일 프로세서에서 GPU 애플리케이션의 확장성을 연구 하기 위해서는 보다 추상적인 시뮬레이션 모델이 필요합니다.

그림 13은 시뮬레이션 방법론을 보여줍니다. 웨이퍼 스케일 GPU 모델링의 첫 번째 단계는 애플리케이션 동작을 추출할 메모리 추적을 수집하는 것입니다. gem5-gpu Syscall Emulation(SE) 모드에서 8개의 CU(컴퓨팅 유닛) GPU를 생성하여 각 CU의 LSQ(Load-Store Queue)에 메모리 프로브를 배치합니다. 다음으로, 체크포인트11를 취하는 ROI(관심 영역)가 시작될 때까지 Linux 부팅을 통해 각 벤치마크를 빨리 감기로 실행합니다 .

마지막으로 세부 모드에서 애플리케이션의 전체 ROI를 실행하여 모든 전역 읽기, 쓰기 및 원자적 작업과 관련 스레드 블록의 메모리 추적을 수집합니다. 파일 은 추적 기반 시뮬레이터에 공급됩니다.

트레이스 기반 시뮬레이터는 먼저 트레이스를 파싱하고 상대 타이밍, 가상 주소, 작업 유형 (읽기/쓰기/원자) 및 메모리 액세스 크기를 수집하고 액세스의 블록 ID를 유지하지만 특정에 대한 유사성을 지웁니다. 컴퓨팅 유닛. 개인 컴퓨팅 시간은 원래 요청을 실행한 컴퓨팅 장치의 듀티 사이클을 곱한 비연속적인 메모리 액세스 사이에 소요 된 시간으로 추정됩니다 . 이 개인 계산 시간은 단순히 원시 계산이 아니라 블록 공유 메모리 액세스도 포함합니다.

시뮬레이터의 관점에서 두 작업 사이에는 차이가 없습니다. 이러한 컴퓨팅 요청은 전역 메모리 액세스 와 함께 스레드 블록으로 그룹화됩니다 . 스레드 블록을 실행할 때 컴퓨팅 요청은 모든 미결 메모리 요청이 완료 될 때까지 보수적으로 기다려야 합니다 . 반대로 새 메모리 요청은 처리할 미해결 컴퓨팅 요청 이 없을 때까지 기다려야 합니다. 이 가정은 스레드 블록 내에서 순서대로 워프를 실행하는 것을 기반으로 합니다. 실제로 로컬 워프 스케줄러는 캐시 미스 시 워프를 전환하여 계산 및 메모리 액세스를 겹칩니다.

소수의 계산 단위에 대해 gem5-gpu에 대해 추적 기반 시뮬레이터를 검증했습니다. 그림 16은 서로 다른 수의 CU에 대한 두 시뮬레이터의 정규화된 성능을 비교한 것입니다.12 그림 17은 정규화된 성능을 비교 한 것입니다.

11 각 애플리케이션에 대해 ROI는 추적이 약 20,000개의 스레드 블록을 생성하도록 충분히 큰 입력 크기로 실행되는 단일 연속 코드 섹션입니다.

12우리는 bc 및 색상 에 대한 검증 데이터를 생성할 수 없었습니다 . 워크로드 데이터 세트가 너무 커서 gem5-GPU 설정에서 완료할 수 없었습니다.

서로 다른 DRAM 대역폭 값에 대해 두 시뮬레이터에서 우리는 두 시뮬레이터에서 5%의 기하학적 평균과 28%의 최대 오류를 관찰합니다.

CU는 확장됩니다13. 우리는 7%의 기하 평균을 관찰하고
DRAM 대역폭이 조정될 때 두 시뮬레이터에서 최대 오류는 26% 입니다. 결과는 시뮬레이션 접근 방식의 유효성을 시사합니다(gem5-gpu와 비교).

추가 검증 단계로 우리 는 두 시뮬레이터에서 대역폭, 데이터 지역성 및 계산 리소스의 상호 작용을 시각적으로 표현하는 지붕선 플롯 [78] 을 만들었습니다.

8CU 시스템 의 루프라인 플롯인 그림 18의 육안 검사는 gem5-gpu와 트레이스 기반 시뮬레이터 간에 동일한 일반 특성과 애플리케이션 포지셔닝을 보여줍니다 .
이를 통해 컴퓨팅 대 메모리 비율, 데이터 지역성 및 대역폭 병목 현상과 같은 애플리케이션 특성 이 추적 기반 방법론에서 보존 된다는 확신을 더욱 강화할 수 있습니다.

섹션 VII의 결과를 위해 섹션 IV에서 살펴본 VFS 스케일링에 따라 575MHz에서 실행되는 24GPM 웨이퍼스케일 GPU와 408.2MHz에서 실행되는 40GPM 웨이퍼스케일 GPU를 평가 합니다. 단일 MCM GPU, 6개 패키지, 24GPM MCM-GPU 및 10개 패키지, 40GPM MCM-GPU를 비교합니다. 우리의 기본 스케줄러는 [34], [79] 에서 제안한 것과 유사한 동적 스케줄링입니다. 즉, GPM 첫 번째 및 첫 번째 터치 페이지 배치 내의 라운드 로빈으로 페이지가 처음 액세스되는 GPM에 매핑됩니다. 우리 시스템은 Rodinia [53] 의 5가지 벤치마크와 gem5-gpu에서 성공적으로 실행할 수 있는 Pannotia 제품군[33]의 2가지 비정규 워크로드에서 평가되었습니다(표 IX) . 우리 는 웨이퍼 스케일 GPM 간 네트워크에 메시 네트워크 토폴로지를 사용하고 스케일 아웃 MCM-GPU에 대해 온 보드 메시 네트워크를 사용하여 상호 연결된 MCM 패키지를 고려합니다.

VII. 결과

웨이퍼스케일 GPU 아키텍처의 성능 및 에너지 이점을 정량화하기 위해 제안된 웨이퍼스케일 아키텍처의 이점을 로컬 DRAM과 함께 각 MCM-GPU에 4GPM이 있는 여러 MCM-GPU를 사용하여 구축된 24 및 40GPM 시스템과 비교합니다. MCM-GPU는 기존의 PCB 기반 통합 기술을 사용하여 통합되는 것으로 가정합니다. 비교는 두 가지 스케줄링 및 데이터 배치 정책을 기반으로 수행됩니다. 하나는 스레드 그룹의 온라인 분산 스케줄링(라운드 로빈)입니다.

13일관성, 운영 체제 효과, 스레드 블록 스케줄링 런타임 및 정교한 메모리 병합 체계가 트레이스 시뮬레이터에서 정확하게 모델링되지 않아 오류가 발생합니다. 또한 메모리 대기 시간을 최소화하기 위해 로컬 공유 메모리 또는 워프 스케줄링 기술을 사용하는 것과 같은 최적화 는 모델링되지 않습니다.

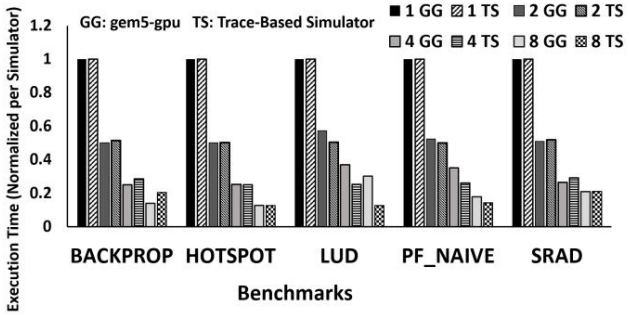


그림 16: Gem5-GPU 및 추적 기반 시뮬레이터에 대한 CU 크기 조정

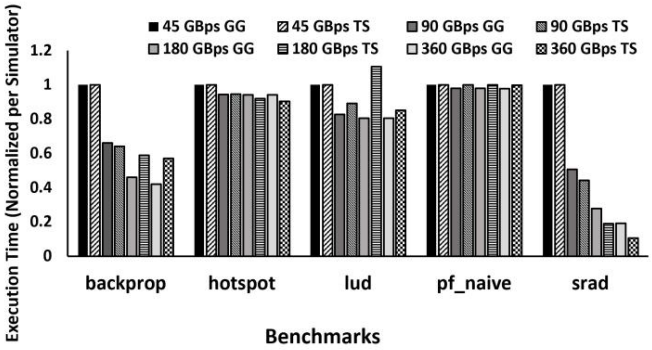


그림 17: Gem5-GPU에 대한 DRAM 대역폭 크기 조정 및 8CU에 대한 추적 기반 시뮬레이터

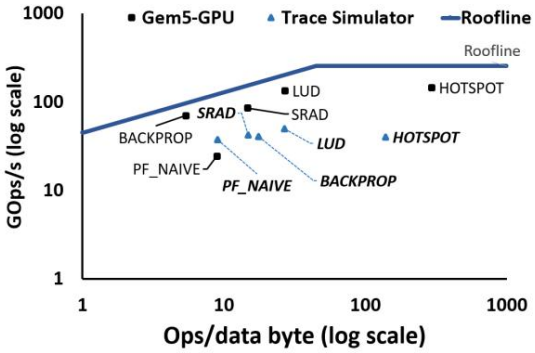


그림 18: Gem5-GPU와 Trace Simulator 간의 Roofline Plot 비교

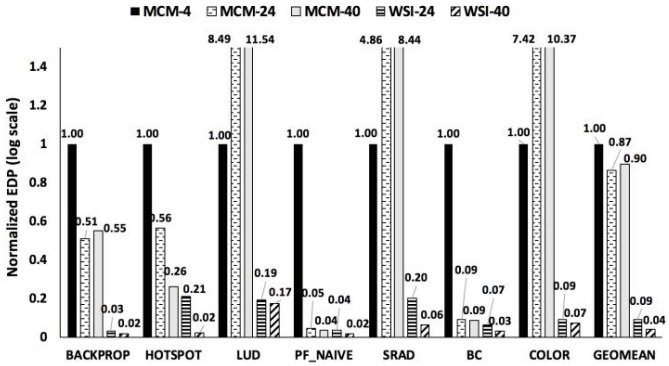


그림 20: 웨이퍼스케일 GPU용 EDP 대 MCM 패키지 기반 기존 시스템

GPM 먼저) [34] 에서 제안 된 첫 번째 터치 페이지 배치와 결합 (RR-FT)하고 다른 하나는 오프라인 파티셔닝 및 배치 접근 방식(MC-DP)(섹션 V)입니다.

먼저 MC-DP를 사용하여 결과를 비교합니다. 그림 19 및 20은 MC-DP를 사용할 때 다양한 벤치마크에서 서로 다른 구성에 대한 비교 원시 성능 및 EDP 결과를 보여줍니다 . 역전파, 핫스팟, 파티클 필터 나이브와 같은 응용 프로그램은 단일 MCM-GPU(4-GPM) 를 통해 MCM-GPU를 사용하여 각각 구현된 24-GPM 및 40-GPM 시스템에서 최대 4.9배 및 6.1배 속도 향상의 성능 이점을 보여줍니다.) . 그러나 lud 및 color와 같은 응용 프로그램은 MCM-24 또는 MCM-40 시스템 에서 실행될 때 성능 저하를 나타냅니다 . 이는 이러한 애플리케이션의 큰 메모리 공간과 불규칙한 액세스 패턴으로 인해 궁극적으로 성능 확장을 지배하는 중요한 MCM 간 데이터 전송이 발생하기 때문입니다.

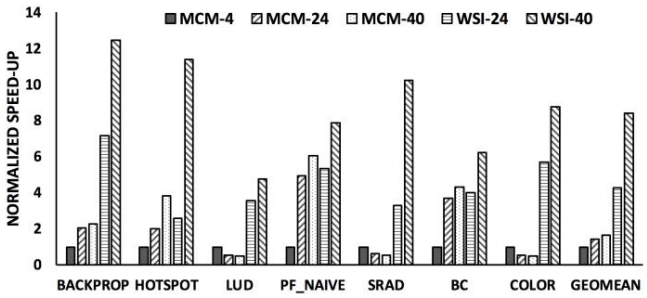


그림 19: Waferscale GPU와 MCM 패키지 기반 기존 시스템의 성능 개선

다중 MCM GPU에서 스케줄링이 처음 수행되는 동안 라운드 로빈 방식의 GPM 내에서 더 큰 커널은 로드 밸런싱 을 위해 여전히 분할되어 여러 MCM 모듈에 분산되었습니다. 또한 네트워크의 MCM 모듈 수가 증가함에 따라 멀티 홉 통신의 비율이 증가하고 이는 MCM 간 온보드 대역폭에 의해 더욱 병목 현상이 발생 합니다.

반면에 모든 응용 프로그램은 24-GPM 및 40-GPM 웨이퍼스케일 GPU 아키텍처에서 각각 최대 10.9배(평균 2.97배) 및 18.9배 (평균 5.2배)까지 상당한 속도 향상을 보입니다. MCM-GPU 기반 아키텍처. 이러한 큰 속도 향상은 전체 웨이퍼에서 사용 가능한 매우 높은 GPM 간 대역폭 때문입니다 . 마찬가지로 웨이퍼 스케일 통합에서 평균 EDP 이점 9.3x 및 22.5x를 얻을 수 있습니다. 이는 GPM 간 통신의 에너지 효율성이 10배 향상 되었을 뿐만 아니라 실행 시간이 크게 단축되었기 때문입니다 .

RR-FT 정책으로 이동하면 웨이퍼스케일 시스템과 MCM-GPU 기반 시스템 간의 성능 격차가 MC DP 보다 RR-FT를 사용할 때 약 2배 더 높다는 것을 알 수 있습니다. RR-FT에서 MC-DP로 전환할 때 MCM-GPU와 웨이퍼스케일 시스템 사이의 성능 격차가 감소 한다는 것은 스케일아웃 MCM-GPU 기반 시스템이 MC-DP를 사용할 때 웨이퍼스케일 기반 시스템보다 더 많은 이점이 있다는 것을 의미합니다. 이는 MCM-GPU 기반 시스템에서 MCM 간 온보드 통신 비용이 웨이퍼스케일 시스템 보다 훨씬 높기 때문입니다 . 결과적으로 지능형 스케줄링 및 데이터 배치를 통해 이러한 통신 을 줄이는 데 도움이 되는 우리의 오프라인 정책 은 스케일 아웃 MCM-GPU 시스템 의 성능을 크게 향상시킵니다 . 따라서 우리의 오프라인 정책 은 확장형 MCM-GPU 시스템에도 적합합니다.

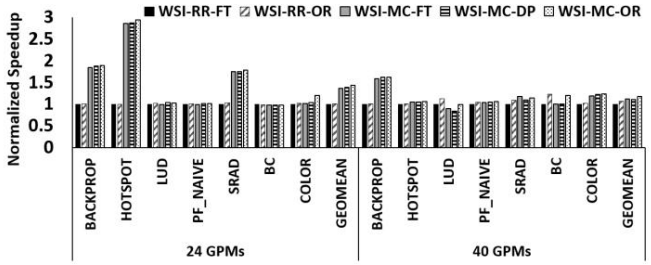


그림 21: 다양한 스케줄링 및 데이터 배치 정책의 성능

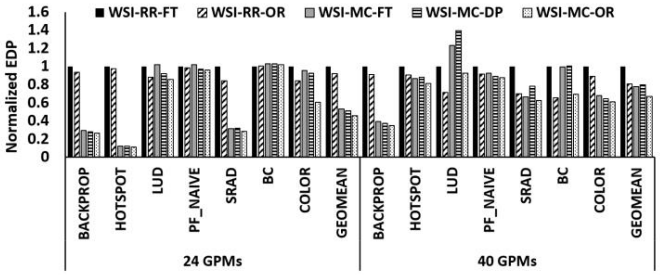


그림 22: 다양한 스케줄링 및 데이터 배치 정책의 EDP

이제 오프라인 파티셔닝 및 배치 접근 방식 을 웨이퍼 스케일 기반 시스템에 대한 기준 지역 인식 분산 스케줄링 정책 과 보다 자세히 비교 합니다. 현실적인 첫 번째 터치 페이지 배치(RR-FT 정책)와 원격 액세스에 오버헤드가 없거나 원격 액세스가 없음을 보장 하는 신탁 데이터 배치(RR-OR)로 기본 정책을 평가합니다.

시뮬레이션에서 모든 DRAM 페이지를 모든 GPM의 로컬 DRAM에 배치하여 RR-OR을 시뮬레이션했습니다. 마찬가지로 우리는 오프라인 접근 방식의 세 가지 변형을 고려했습니다. 첫 번째 경우(MC FT)에서는 첫 번째 터치 페이지 배치 정책 과 함께 오프라인 파티셔닝 결과에서 스레드 블록 일정만 사용되었습니다 . 두 번째 경우(MC-DP)에서는 파티셔닝 및 배치 프레임 작업의 데이터 배치 출력을 고려했습니다. 세 번째 경우(MC-OR)에서 우리는 이러한 스레드 블록 일정을 사용하여 가능한 최대 속도 향상을 고려했습니다. 즉, 다시 모든 DRAM 페이지가 모든 GPM의 로컬 DRAM에 배치되었습니다.

그림 21에서 볼 수 있듯이 런타임 RR-FT 정책은 RR-OR 정책 에 비해 평균 7% 더 나쁜 성능을 보입니다. 이는 MCM-GPU와 관련 하여 [34] 에서 관찰한 것과 유사합니다 . 또한 파티셔닝 및 GPM 배치 정책이 RR 기반 정책(FT 및 oracular 모두)을 훨씬 능가하는 것으로 관찰됩니다. 역전파, 핫스팟, srad 와 같은 애플리케이션 의 경우 24GPM의 경우 RR-FT를 통해 최대 2.88배의 큰 성능 이점을 얻을 수 있습니다. 혜택은 40 GPM의 경우 최대 1.62배입니다. 이러한 큰 이점은 우리의 파티셔닝 체계가 동일한 DRAM 페이지에 닿는 스레드 블록을 함께 클러스터링한다는 사실에서 비롯됩니다. 이렇게 하면 총 원격 액세스 수가 최소화됩니다. 이것은 또한 GPM 내의 데이터 지역성이 강력하게 보존되어 캐시 적중률이 증가하기 때문에 캐싱을 더욱 효과적으로 만듭니다. 전반적으로 평균적으로 우리의 오프라인 정책(MC-DP) 은 기본 RR-FT 정책에 비해 1.4배(24GPM) 및 1.11배(40GPM)의 이점이 있으며 가능한 최대 속도의 16% 이내입니다 (MC- 또는). 이것은 24 및 40 GPM 시스템에 대해 각각 49% 및 20%의 평균 EDP 이점을 초래합니다 (그림 22 참조). 이것은 또한 프로그래머 및 컴파일러 힌트 [80], [81] 에 기반한 온라인 스케줄링 최적화 가 도움이 될 수 있음을 나타냅니다. RR-FT 온라인 정책보다 WS-GPU에서 더 나은 성능을 달성합니다 . RR-FT에 비해 MC-DP의 상대적 이점은 24GPM 시스템에 비해 40GPM 시스템에서 더 작습니다. 이는 시스템 크기가 커짐에 따라 TB가 더 많은 GPM에 분산되어 캐싱의 이점이 감소 하기 때문일 수 있습니다 .

앞에서 언급했듯이 GPM의 작동 주파수로 575MHz를 사용합니다. 더 높은 GPM 주파수에서는 통신이 더 많은 병목 현상이 발생하므로 WSI-GPU 이점이 증가합니다.

데이터 액세스 요청 빈도가 증가함에 따라 GPM이 더 빠르게 실행되는 경우. 예를 들어, 평균적으로 WS-GPU-24 는 MCM-24보다 1GHz 대 575MHz에서 추가로 7% 더 뛰어난 성능을 보입니다.

40 GPM이 있는 토폴로지의 경우 비적층 구성도 고려했습니다. 비적층 구성에서 GPM은 훨씬 더 낮은 전압과 주파수(0.71V/360MHz)에서 실행되어야 합니다. 결과 시스템은 스택 구성에 비해 평균 14% 낮은 성능을 보입니다.

마지막으로 열 분석에서는 효율적인 강제 공기 냉각 모델을 가정했습니다. 액체 또는 상변화 냉각 솔루션은 지속 가능한 TDP를 증가시켜 훨씬 더 높은 컴퓨팅 밀도를 가능하게 합니다 [82]. 액체 냉각으로 인한 열 예산이 2배 증가 하면 기본 MCM-40에 비해 WS-GPU-40의 성능이 추가로 20~30% 증가할 수 있습니다.

VIII. 요약 및 결론

웨이퍼스케일 프로세서는 통신 오버헤드를 크게 줄일 수 있습니다. 그러나 그들은 허용할 수 없는 수율을 보였습니다. 사전 제조된 다이가 실리콘 웨이퍼에 직접 결합되는 Si-IF(Silicon Interconnection Fabric) 기반 통합 [5,6] 과 같은 새로운 통합 기술 은 해당 수율 문제 없이 웨이퍼스케일 프로세서를 가능하게 할 수 있습니다. 따라서 웨이퍼스케일 아키텍처를 다시 검토할 시간이 무르익었습니다 . 이 백서에서 우리는 현대 웨이퍼 스케일 시스템 을 설계하는 것이 실현 가능하고 유용하다는 것을 보여주었습니다 . 웨이퍼스케일 GPU를 사례 연구로 사용하여 300mm 웨이퍼 에 약 100개의 GPU 모듈(GPM)을 수용할 수 있지만 물리적 문제를 고려할 때 약 40GPM의 훨씬 축소된 GPU 아키텍처만 구축할 수 있음 을 보여주었습니다. 우리는 또한 웨이퍼 스케일 아키텍처의 성능 및 에너지 영향을 연구했습니다. 우리 는 웨이퍼스케일 GPU가 프로그래밍 모델을 변경하지 않고도 상당한 성능 및 에너지 효율성 이점(PCB에서 동등한 MCM-GPU 기반 구현에 비해 최대 18.9배 속도 향상 및 143배 EDP 이점)을 제공할 수 있음을 보여주었습니다. 또한 웨이퍼 스케일 GPU 아키텍처를 위한 스레드 스케줄링 및 데이터 배치 정책을 개발했습니다. 당사의 정책은 24GPM 및 40GPM 사례에 대해 각각 2.88배(평균 1.4배) 및 1.62배(평균 1.11배) 최신 스케줄링 및 데이터 배치 정책을 능가 했습니다. 마지막으로, 우리는 상호 연결된 다이가 있는 첫 번째 Si-IF 프로토타입을 제작했습니다. 프로토타입에 대해 관찰한 100% 수율과 이전 Si-IF 프로토타입에 대해 보고된 높은 본드 수율 은 웨이퍼 스케일 GPU 아키텍처 구축을 위한 기술적 준비 상태를 보여줍니다 .

IX. 승인

이 작업은 Defense Advanced에서 부분적으로 지원했습니다.
ONR 보조금을 통한 연구 프로젝트 기관(DARPA)
N00014-16-1-263, 교부금 MRP-17-454999를 통한 UCOP,

Futurewei Technologies, 삼성 전자 및 UCLA CHIPS 컨소시엄. 저자는 논문 의 Si-IF 프로토타입 을 도와준 SivaChandra Jangam과 Adeel A. Bajwa와 유용한 피드백과 제안을 해주신 익명의 검토자에게 감사드립니다.

참조

[1] "3부작 시스템." https://en.wikipedia.org/wiki/Trilogy 시스템, (2017년 11월 20일 액세스).

[2] RM Lea, "WASP: 웨이퍼 규모 대규모 병렬 처리기", 1990년 Proceedings. 웨이퍼 스케일 통합에 관한 국제 회의, pp. 36–42, 1990년 1월.

[3] D. Landis 및 J. Yoder 및 D. Whittaker 및 T. Dobbins, "A wafer scale programmable systolic data processor," Proceedings Ninth Biennial University/Government/Industry Microelectronics Symposium, pp. 252–256, 1991년 6월.

[4] JF McDonald, EH Rogers, K. Rose, AJ Steckl, " 웨이퍼 스케일 통합의 시련: 1960년대 WSI가 처음 시도된 이후 주요 기술 문제가 극복되었지만 상업 회사 는 아직 성공할 수 없습니다. 날다," IEEE Spectrum, vol. 21, pp. 32–39, 1984년 10월.

[5] A. Bajwa 및 S. Jangam 및 S. Pal 및 N. Marathe 및 T. Bai 및 T. Fukushima 및 M. Goorsky 및 SS Iyer, " 열 압축 본딩을 사용한 미세 피치 (≤10μm) 에서 이기종 통합", 2017 IEEE 67차 전자 부품 및 기술 컨퍼런스(ECTC), pp. 1276–1284, 2017년 5월.

[6] S. Jangam, S. Pal, A. Bajwa, S. Pamarti, P. Gupta, SS Iyer, "SuperCHIPS 통합 체계의 대기 시간, 대역폭 및 전력 이점", 2017년 IEEE 67차 전자 부품 및 기술 컨퍼런스 (ECTC), pp. 86–94, 2017년 5월.

[7] S. Pal, D. Petrisko, AA Bajwa, P. Gupta, SS Iyer 및 R. Kumar, "A case for packageless processors," 2018 IEEE International Symposium on High Performance Computer Architecture (HPCA), pp. 466–479, 2018년 2월.

[8] M. Panahiazar 및 V. Taslimitehrani 및 A. Jadhav 및 J. Pathak, "빅 데이터 및 시맨틱 웹 기술을 통한 맞춤형 의료 강화: 약속, 과제 및 사용 사례", 2014년 IEEE 빅 데이터 국제 회의(Big 자료), pp. 790–795, 2014년 10월.

[9] "NVIDIA가 가상 현실을 강화합니다." http://www.nvidia.com/object/virtual reality.html, (2017년 11월 21일 액세스).

[10] "NVIDIA Tesla 이미징 및 컴퓨터 비전." http://www.nvidia.com/ object/imaging comp vision.html, (2017년 11월 21일 액세스).

[11] Karen Simonyan 및 Andrew Zisserman, " 대규모 이미지 인식을 위한 매우 깊은 컨벌루션 네트워크" CoRR, vol. 복근/1409.1556, 2014.

[12] "Nvidia는 자사의 새로운 슈퍼컴퓨터가 최고 수준의 자동 운전을 가능하게 할 것이라고 말합니다." https://www.theverge.com/2017/10/10/16449416/ nvidia-pegasus-self-driving-car-ai-robotaxi, (2017년 11월 21일 액세스).

[13] "AWS의 고성능 컴퓨팅 소개(백서)." https://d0.awsstatic.com/whitepapers/Intro to HPC on AWS.pdf, (2017년 11월 21일 액세스).

[14] "Microsoft Azure"(2017년 11월 21일 액세스).

[15] “TOP500, 액셀러레이터 성장 모멘텀 보여.” https://insidehpc.com/2015/11/top500-shows-growing-momentum for-accelerators/, (2017년 11월 21일 액세스).

[16] "GPU 컴퓨팅을 위한 HPC 애플리케이션 지원." https://insidehpc.com/ 2015/11/ top500-shows-growing-momentum-for-accelerators/, (2017년 11월 21일 액세스).

[17] T. Vijayaraghavan 및 Y. Eckert 및 GH Loh 및 MJ Schulte 및 M. Ignatowski 및 BM Beckmann 및 WC Brantley 및 JL Greathouse 및 W. Huang 및 A. Karunanithi 및 O. Kayiran 및 M. Meswani and I. Paul and M. Poremba and S. Raasch and SK Reinhardt and G. Sadowski and V. Sridharan, " Exascale Computing 을 위한 APU 설계 및 분석 ", 2017 IEEE 고성능 컴퓨터 아키텍처 심포지엄(HPCA) , pp. 85–96, 2017년 2월.

[18] "Sandy Bridge(클라이언트) - 마이크로아키텍처 - Intel(Wikichip)." https: // en.wikichip.org/wiki/intel/microarchitectures/sandy bridge(클라이언트), (2017년 11월 20일 액세스).

[19] SS Iyer, "성능 및 확장을 위한 이기종 통합", 부품, 패키징 및 제조 기술에 관한 IEEE 트랜잭션 , vol. 6, pp. 973–982, 2016년 7월.

[20] R. Mahajan 및 R. Sankman 및 N. Patel 및 DW Kim 및 K. Aygun 및 Z. Qian 및 Y. Mekonnen 및 I. Salama 및 S. Sharan 및 D. Iyengar 및 D. Mallik, "EMIB(Embedded Multi-die Interconnect Bridge) – A High Density, High Bandwidth Packaging Interconnect" , 2016년 IEEE 제66회 전자 부품 및 기술 컨퍼런스 (ECTC), pp. 557–565, 2016년 5월.

[21] JW Poulton 및 WJ Dally 및 X. Chen 및 JG Eyles 및 TH Greer 및 SG Tell 및 JM Wilson 및 CT Gray, " 고급 패키징 응용 분야를 위한 28nm CMOS 의 0.54pJ/b 20Gb/ s 접지 참조 단일 종단 단거리 직렬 링크 ", IEEE 고체 회로 저널, 권. 48, pp. 3206–3218, 2013년 12월.

[22] Abts, Dennis and Marty, Michael R. and Wells, Philip M. and Klausler, Peter and Liu, Hong, "Energy Proportional Datacenter Networks," in the Proceedings of the 37th Annual International Symposium on Computer Architecture, ISCA '10, (뉴욕, 뉴욕, 미국), 338–347쪽, ACM, 2010년.

[23] Y. Huang 및 Y. Yesha 및 M. Halem 및 Y. Yesha 및 S. Zhou, "YinMem: 대규모 데이터 분석을 위한 분산 병렬 인덱싱 인메모리 계산 시스템 ", 2016 IEEE 빅 데이터 국제 회의 (빅데이터), pp. 214–222, 2016년 12월.

[24] J. Ahn, S. Hong, S. Yoo, O. Mutlu, and K. Choi, "A scalable processing-in-memory accelerator for parallel graph processing," 2015 ACM/IEEE 42nd Annual International Symposium on Computer 아키텍처(ISCA), pp. 105–117, 2015년 6월.

[25] E. Azarkhish 및 C. Pfister 및 D. Rossi 및 I. Loi 및 L. Benini, "스마트 메모리 큐브의 근거리 메모리 컴퓨팅을 위한 논리 기반 상호 연결 설계", VLSI(Very Large Scale Integration) 에 대한 IEEE 트랜잭션 시스템, vol. 25, pp. 210–223, 2017년 1월.

[26] K. Murthy 및 J. Mellor-Crummey, "Communication avoiding algorithms: Analysis and code generation for parallel systems," in 2015 International Conference on Parallel Architecture and Compilation (PACT), pp. 150–162, 2015년 10월.

[27] TG Lenihan, L. Matthew 및 EJ Vardaman, "2.5D 개발: 실리콘 인터포저의 역할", 2013년 IEEE 15차 전자 패키징 기술 컨퍼런스(EPTC 2013), pp. 53–55, 2013년 12월.

[28] V. Kumar 및 A. Naeemi, "3d 집적 회로 개요", RF, 마이크로파 및 테라헤르츠 애플리케이션(NEMO)을 위한 수치 전자기 및 다중물리 모델링 및 최적화에 관한 2017 IEEE MTT-S 국제 회의, pp. 311–313, 2017년 5월.

[29] A. Sodani, R. Gramunt, J. Corbal, HS Kim, K. Vinod, S. Chinthamani, S. Hutsell, R. Agarwal 및 YC Liu, "Knights Landing: 2세대 Intel Xeon Phi 제품," IEEE 마이크로, vol. 36, pp. 34–46, 2016년 3월.

[30] James B. Brinton 및 J. Robert Lineback, 패키징은 자본 설비를 통과하면서 조립에서 가장 큰 비용이 되고 있습니다. EE Times [온라인], 1999.

[31] LD Cioccio, P. Gueguen, R. Taibi, T. Signamarcheix, L. Bally, L. Vandroux, M. Zussy, S. Verrun, J. Dechamp, P. Leduc, M. Assous, D. Bouchu, F. de Crecy, L. Chapelon 및 L. Clavelier, "혁신적인 다이-웨이퍼 3d 통합 방식: 다이-웨이퍼 산화물 또는 구리 다이렉트 본딩과 평탄화 산화물 다이 간 충진", 2009년 IEEE 3D 시스템 통합에 관한 국제 회의 , pp. 1–4, 2009년 9월.

[32] A. Sigl, S. Pargfrieder, C. Pichler, C. Scheiring 및 P. Kettner, "고급 칩-웨이퍼 본딩: 최저 소유 비용을 제공하는 대량 3dic 생산을 위한 플립 칩-웨이퍼 본딩 기술 ," 2009년 전자 패키징 기술 고밀도 패키징 에 관한 국제 회의 , pp. 932–936, 2009년 8월.

[33] S. Che, BM Beckmann, SK Reinhardt 및 K. Skadron, "Pannotia: 불규칙한 gpgpu 그래프 응용 프로그램 이해", 2013 IEEE International Symposium on Workload Characterization(IISWC), pp. 185–195, 2013년 9월.

[34] Arunkumar, Akhil 및 Bolotin, Evgeny 및 Cho, Benjamin 및 Milic, Ugljesa 및 Ebrahimi, Eiman 및 Villa, Oreste 및 Jaleel, Aamer 및 Wu, Carole-Jean 및 Nellans, David, "MCM-GPU: Multi-Chip- 지속적인 성능 확장성을 위한 모듈 GPU" , 컴퓨터 아키텍처에 관한 제44회 연례 국제 심포지엄 절차, ISCA '17, (뉴욕, 뉴욕, 미국), pp. 320–332, ACM, 2017.

[35] YL Chuang, CS Yuan, JJ Chen, CF Chen, C. S. Yang 및 WP Changchien 및 CCC Liu 및 F. Lee, "Unified methodology for heterogeneous integration with CoWoS technology," in 2013 IEEE 63rd Electronic Components and Technology Conference, pp. 852–859, 2013년 5월.

[36] CL Lai 및 HY Li 및 A. Chen 및 T. Lu, " 유리 캐리어 CTE 및 패시베이션 두께 조정을 활용하는 TSV 없는 2.5D IC에 대한 실리콘 인터포저 휨 연구", 2016년 IEEE 66차 전자 부품 및 기술 컨퍼런스(ECTC) , 310~315쪽, 2016년 5월.

[37] Ying-Wen Chou 및 Po-Yuan Chen, Mincent Lee 및 CW Wu, "인터포저 기반 3차원 IC에 대한 비용 모델링 및 분석" , 2012년 IEEE 30차 VLSI 테스트 심포지엄(VTS), pp. 108–113 , 2012년 4월.

[38] John Hu, " 고성능 컴퓨팅 시스템을 위한 2.5D/3D 하이브리드 통합의 시스템 수준 공동 최적화 ", Semicon West 2016, 2016.

[39] R. Arnold 및 SM Menon 및 B. Brackett 및 R. Richmond, " 매우 신뢰할 수 있는 알려진 양호한 다이(KGD)를 생산하는 데 사용되는 테스트 방법", Proceedings. 1998년 멀티칩 모듈 및 고밀도 패키징에 관한 국제 회의(Cat. No.98EX154), pp. 374–382, 1998년 4월.

[40] HD Thacker 및 MS Bakir 및 DC Keezer 및 KP Martin 및 JD Meindl, "높은 핀 수의 칩 스케일 패키지를 테스트하기 위한 준수 프로브 기판", 2002년 제52회 전자 부품 및 기술 컨퍼런스(Cat. No.02CH37345), pp. 1188–1193, 2002.

[41] E. Wachter, A. Erichsen, A. Amory 및 F. Moraes, "토폴로지에 구애받지 않는 내결함성 noc 라우팅 방법", 유럽의 설계, 자동화 및 테스트 회의 절차, DATE '13, (San Jose, CA, USA), pp. 1595–1600, EDA 컨소시엄, 2013.

[42] D. Fick, A. DeOrio, G. Chen, V. Bertacco, D. Sylvester 및 D. Blaauw, "A high resilient routing algorithm for fault-tolerant nocs," in 2009 Design, Automation Test in Europe Conference 전시회, pp. 21–26, 2009년 4월.

[43] ITRS, "수율 향상," 2007년판.

[44] A. Kannan 및 NE Jerger 및 GH Loh, "Enabling interposer-based disintegration of multi-core processors," in 2015 48th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), pp. 546–558, 2015년 12월.

[45] I. Tirkel, "반도체 제조의 수율 학습 곡선 모델 ", 반도체 제조에 관한 IEEE 트랜잭션, vol. 26, pp. 564–571, 2013년 11월.

[46] J. Wang 및 S. Yalamanchili, "구조화되지 않은 GPU 애플리케이션의 동적 병렬 처리 특성화 및 분석", 2014 IEEE 워크로드 특성화에 관한 국제 심포지엄(IISWC), pp. 51–60, 2014년 10월.

[47] J. Sim, A. Dasgupta, H. Kim, and R. Vuduc, "A performance analysis framework for identification for potential benefits in gpgpu applications," in the 17th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, PPoPP '12, (뉴욕, 뉴욕, 미국), pp. 11–22, ACM, 2012.

[48] NVIDIA, “NVIDIA, GPU 로드맵 업데이트; 파스칼 발표”(2015).

[49] "푸른 바다." http://www.ncsa.illinois.edu/enabling/bluewaters, (2018년 7월 1일 액세스).

[50] "Blue Waters의 GPU 슈퍼컴퓨팅." http://developer.download.nvidia. com/compute/academia/whitepapers/Achievement2013 UIUC.pdf, (2018년 7월 1일 액세스).

[51] "Blue Waters의 작업량 분석." https://arxiv.org/ftp/arxiv/papers/1703/1703.00924.pdf, (2018년 7월 1일 액세스).

[52] M. Vinas, Z. Bozkus 및 BB Fraguela, "이기종 프로그래밍 라이브러리를 사용한 이기종 병렬 처리 활용", J. Parallel Distrib. 컴퓨터, vol. 73, pp. 1627–1638, 2013년 12월.

[53] Che, Shuai 및 Boyer, Michael 및 Meng, Jiayuan 및 Tarjan, David 및 Sheaffer, Jeremy W 및 Lee, Sang-Ha 및 Skadron, Kevin, "Rodinia: 이기종 컴퓨팅을 위한 벤치마크 제품군", Workload Characterization, 2009 . IISWC 2009. IEEE 국제 심포지엄 on, pp. 44–54, Ieee, 2009.

[54] J. Power, J. Hestness, MS Orr, MD Hill 및 DA Wood, "gem5-gpu: 이기종 cpu-gpu 시뮬레이터," IEEE Computer Architecture Letters, vol. 14, pp. 34–36, 2015년 1월.

[55] "R-tools 3-d 방열판 열 모델링." http://www.r-tools.com/.

[56] W. Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron 및 MR Stan, "Hotspot: 초기 단계 vlsi 설계를 위한 소형 열 모델링 방법론", 초대형 통합 에 대한 IEEE 트랜잭션 (VLSI) 시스템, vol. 14, pp. 501–513, 2006년 5월.

[57] J. Liu, B. Jaiyen, R. Veras 및 O. Mutlu, "Raidr: Retention-aware Intelligent Dram Refresh", 컴퓨터 아키텍처에 관한 제39회 연례 국제 심포지엄 절차, ISCA '12(워싱턴, DC, USA), pp. 1–12, IEEE 컴퓨터 학회, 2012.

[58] "NVIDIA GPU 최대 작동 온도 및 과열." https://http://nvidia.custhelp.com/app/answers/detail/a id/2752/ /nvidia gpu-maximum-operating-temperature-and-overheating, (2017년 11월 21일 액세스).

[59] M. Ahmed 및 C. Fei 및 FC Lee 및 Q. Li, "고효율 고전력 밀도 48/1V 시그마 컨버터 전압 레귤레이터 모듈", 2017 IEEE APEC(Applied Power Electronics Conference and Exposition), 2207–2212, 2017년 3월.

[60] V. Kontorinis, A. Shayan, DM Tullsen 및 R. Kumar, "테이블 기반 적응형 프로세서 코어로 피크 전력 감소" , 마이크로 아키텍처에 관한 제42회 연례 IEEE/ACM 국제 심포지엄 절차, MICRO 42, (뉴욕, 뉴욕, 미국), pp. 189–200, ACM, 2009.

[61] Intel Corp., 423핀 패키지의 Intel Pentium 4 프로세서 열 설계 지침, 2000년 11월.

[62] FC Lee 및 Q. Li 및 Z. Liu 및 Y. Yang 및 C. Fei 및 M. Mu, "자기 장치가 통합된 1kW 서버 전원 공급 장치를 위한 GaN 장치의 응용 ", CPSS Transactions on Power Electronics and Applications, vol. 1, pp. 3–12, 2016년 12월.

[63] "데이터 센터 전력 공급 아키텍처: 효율성 및 연간 운영 비용." http://www.vicorpower.com/documents/whitepapers/ 서버 효율성 vichip.pdf.

[64] "데이터 센터 및 네트워크실 인프라 의 총소유비용 결정 ." http://www.apc.com/salestools/CMRP 5T9PQG/CMRP-5T9PQG R4 EN.pdf.

[65] P. Gupta 및 AB Kahng, "강력한 배전 메쉬 의 효율적인 설계 및 분석 ", 19차 VLSI 설계 국제 회의(VLSID'06), pp. 6pp.–, 2006년 1월.

[66] MH Ahmed 및 C. Fei 및 FC Lee 및 Q. Li, "미래 데이터 센터를 위한 PCB 권선 매트릭스 변압기가 있는 48-V 전압 조정기 모듈", 산업용 전자 제품에 대한 IEEE 트랜잭션, vol. 64, pp. 9302–9310, 2017년 12월.

[67] J. Leng 및 Y. Zu 및 VJ Reddi, "GPU 전압 노이즈: GPU 아키텍처에서 공간 및 시간 전압 노이즈 간섭의 특성화 및 계층적 평활화", 2015 IEEE 21st International Symposium on High Performance Computer Architecture(HPCA), 161–173페이지, 2015년 2월.

[68] SK Lee, T. Tong, X. Zhang, D. Brooks 및 GY Wei, " 적응형 클럭킹 및 통합 전환 커패시터 dc x2013;dc 컨버터를 갖춘 16코어 전압 스택 시스템 ", IEEE Transactions on Very 대규모 통합(VLSI) 시스템, vol. 25, pp. 1271–1284, 2017년 4월.

[69] K. Mazumdar 및 M. Stan, "Breaking the power delivery wall using voltage stacking," in VLSI, GLSVLSI '12, (New York, NY, USA), pp. 51–54 , ACM, 2012.

[70] Q. Zhang, L. Lai, M. Gottscho 및 P. Gupta, "GPU용 다층 전력 분배 네트워크", 2016년 Design, Automation Test in Europe Conference Exhibition(DATE), pp. 451–456, 2016년 3월.

[71] E. Alon 및 M. Horowitz, "에너지 효율적인 디지털 회로에 대한 통합 규정", vol. 43, pp. 1795 – 1807, 09 2008.

[72] E. Papadopoulou 및 DT Lee, "보로노이 다이어그램을 통한 임계 영역 계산 ", 집적 회로 및 시스템의 컴퓨터 지원 설계에 관한 IEEE 트랜잭션 , vol. 18, pp. 463–474, 1999년 4월.

[73] N. Chatterjee, M. O'Connor, D. Lee, DR Johnson, SW Keckler, M. Rhu, WJ Dally, "Architecting an Energy-Efficient DRAM System for GPUs", 2017 IEEE International Symposium on High Performance 컴퓨터 아키텍처(HPCA), pp. 73–84, 2017년 2월.

[74] "삼성이 칩 대량 생산을 시작함에 따라 JEDEC, HBM2 사양 발표." https://www.anandtech.com/show/9969/jedec publishes-hbm2-specification.

[75] AE Caldwell, AB Kahng 및 IL Markov, "vlsi 넷리스트 파티셔닝을 위한 fiduccia-mattheyses 휴리스틱의 설계 및 구현", ALENEX '99, ALENEX '99(영국 런던) , UK), pp. 177–193, Springer-Verlag, 1999.

[76] N. Binkert, B. Beckmann, G. Black, SK Reinhardt, A. Saidi, A. Basu, J. Hestness, DR Hower, T. Krishna, S. Sardashti, R. Sen, K. Sewell, M . Shoaib, N. Vaish, MD Hill 및 DA Wood, “The gem5 시뮬레이터,” 시가르치 컴퓨터. 건축가. 뉴스, 권. 39, pp. 1–7, 2011년 8월.

[77] A. Bakhoda, GL Yuan, WWL Fung, H. Wong 및 TM Aamodt, "세부 GPU 시뮬레이터를 사용하여 cuda 워크로드 분석", 2009 IEEE International Symposium on Performance Analysis of Systems and Software, pp. 163–174 , 2009년 4월.

[78] S. Williams, A. Waterman 및 D. Patterson, "Roofline: 멀티코어 아키텍처를 위한 통찰력 있는 시각적 성능 모델" Commun. ACM, vol. 52, pp. 65–76, 2009년 4월.

[79] U. Milic, O. Villa, E. Bolotin, A. Arunkumar, E. Ebrahimi, A. Jaleel, A. Ramirez, D. Nellans, "Beyond the socket: Numa-aware gpus," Proceedings of 마이크로아키텍처에 관한 제50회 연례 IEEE/ACM 국제 심포지엄, MICRO-50 '17, (뉴욕, 뉴욕, 미국), pp. 123–135, ACM, 2017.

[80] N. Vijaykumar, E. Ebrahimi, K. Hsieh, PB Gibbons 및 O. Mutlu, "지역 설명자: gpus에서 데이터 지역성을 표현하는 전체적인 교차 계층 추상화", 2018 ACM/IEEE 45th Annual International 컴퓨터 아키텍처 심포지엄(ISCA), pp. 829–842, 2018년 6월.

[81] J. Wang, N. Rubin, A. Sidelnik 및 S. Yalamanchili, "Laperm: Locality Aware Scheduler for Dynamic Parallelism on GPU", 2016년 ACM/IEEE 43차 연례 컴퓨터 아키텍처 국제 심포지엄(ISCA), pp . 583–595, 2016년 6월.

[82] M. Skach, M. Arora, CH Hsu, Q. Li, D. Tullsen, L. Tang 및 J. Mars, "열 시간 이동: 상변화 재료를 활용 하여 창고 규모 컴퓨터의 냉각 비용 절감, ” 2015 ACM/IEEE 42 차 연례 컴퓨터 아키텍처 국제 심포지엄(ISCA), pp. 439–449, 2015년 6월.