**Advanced Computing System 기말고사**

**1. General questions [10 points] 아래의 설명이 참이면 O 틀리면 X로 답하시오.**

a. Fine-grain multithreading은 매 clock cycle마다 새로운 thread를 실행 시킨다. (O)

b. Vector processor는 memory에서 가져오는 instruction 수가 일반적이 processor에 비해 적다. (O)

c. 메모리 bank interleaving은 메모리의 간섭 문제를 줄일 수 있다. (O)

d. GPU는 graphic에 관련된 응용 프로그램만 성능을 향상시킬 수 있다. (X)

e. GPU의 block은 같은 Streaming multiprocessor에만 mapping된다. (O)

f. GPU에서는 여러 개의 thread가 동일한 instruction을 실행한다. (O)

g. Message passing multiprocessor는 shared memory multiprocessor보다 통신의 비용이 크다. (O)

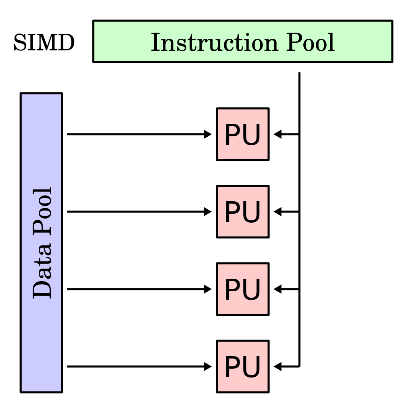
i. Multiprocessor에서 private data는 여러 cache에 공유가 된다. (X)

j. Flash memory는 read 와 write 시간이 틀리다. (O)

k. Flash memory에서는 garbage collection이 필요하다. (O)

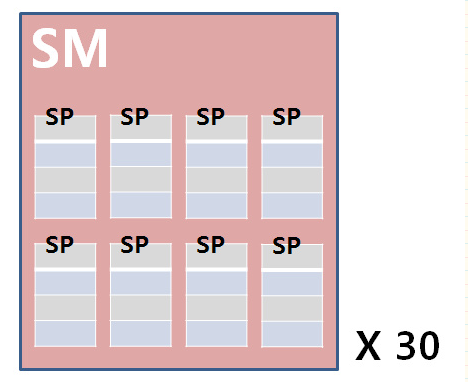
**2. Short answers [15 points]**

(a) Vector Processor는 SIMD 구조를 채택하고 있다. SIMD가 무엇인지 설명하여라.



: SIMD는 single instruction multiple data의 약자이며 [병렬 프로세서](https://ko.wikipedia.org/w/index.php?title=%EB%B3%91%EB%A0%AC_%ED%94%84%EB%A1%9C%EC%84%B8%EC%84%9C&action=edit&redlink=1)의 한 종류로, **하나의 명령어로 여러 개의 값을 동시에 계산하는 방식**이다. 또한, Vector Processor에서 많이 사용되는 방식이고 [비디오 게임 콘솔](https://ko.wikipedia.org/wiki/%EB%B9%84%EB%94%94%EC%98%A4_%EA%B2%8C%EC%9E%84_%EC%BD%98%EC%86%94)이나 [그래픽 카드](https://ko.wikipedia.org/wiki/%EA%B7%B8%EB%9E%98%ED%94%BD_%EC%B9%B4%EB%93%9C)와 같은 멀티미디어 분야에 자주 사용된다.

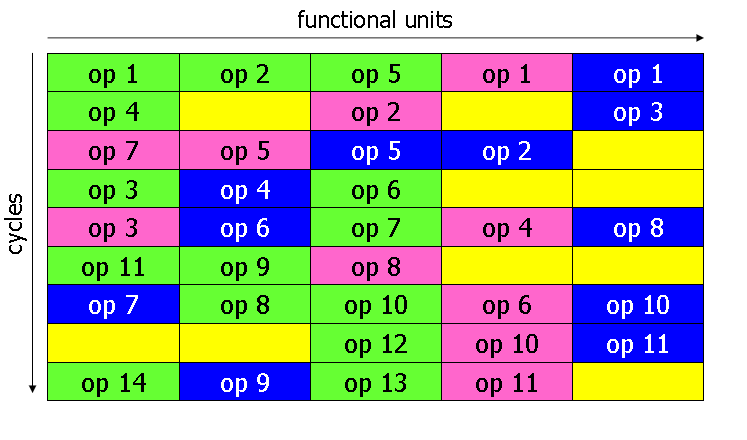
(b) GPU에서 WARP가 무엇인지 설명하여라.

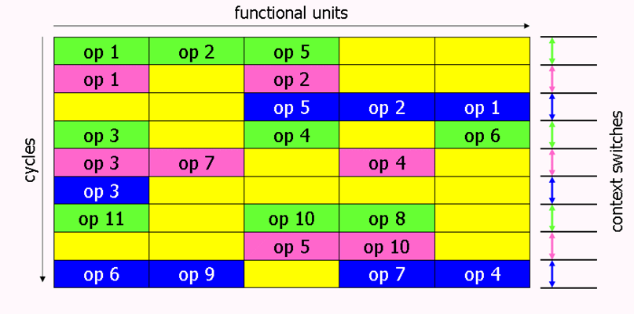


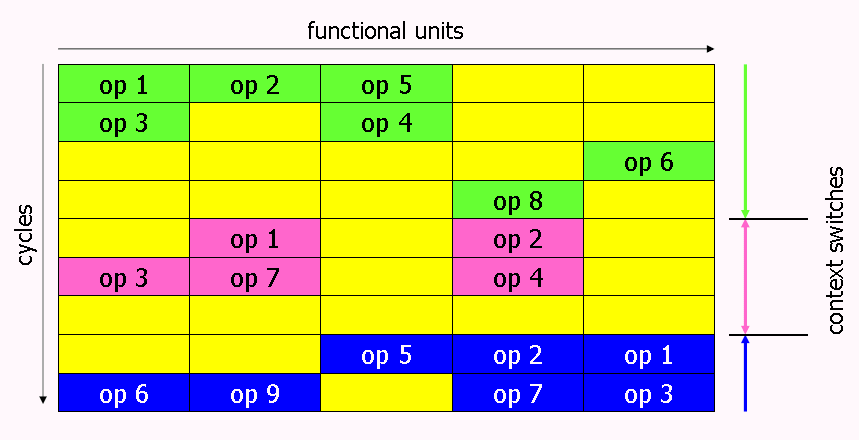
**WARP는 32개 스레드를 단위로 묶은 것이다.**

GPU는 SM의 집합으로 이루어져있다. SM은 Streaming Multiprocessor을 의미하며, SM은 8개의 SP로 이루어져있다. SP는 Scalar Processor 로, 4개의 스레드로 구성되어 있고, 실행한다. 때문에 **SM은 32개의 스레드로 구성되어 있는 것이다.** 즉, SM는 8개의 SP이며 1개의 SP는 4개의 스레드이다. **이 32개의 스레드는 워프(Warp)라는 단위로 불린다.** 16개의 스레드로 메모리참조를 하는 경우에는 하프워프(half-warp)라고 불린다.

(c) 일반적인 multithreading (fine-grain또는 coarse-grain)과 simultaneous multithreading의 차이에 대해 설명하여라.

SMT

 **Fine-Grained Multithreading**

 **Coarse-Grained Multithreading**

SMT(simultaneous multithreading)는 동시 멀티스레딩으로 하나의 실행에 여러 개의 독립 스레드를 허용한다.

반대로 일반적인 multithreading (fine-grain또는 coarse-grain)는 한 사이클 안에 한 개의 스레드만 실행이 가능하다.

**3. GPU [10 points]**

아래는 Ahmdal's law 이다. 이를 사용하여 다음 문제에 답하시오.



(a) GPU의 processor core 가 100개이다. 만일 GPU의 application의 95 %가 병렬화가 될 수 있다면 최대 얻을 수 있는 speedup은 얼마인가?

(b) GPU에서는 streaming application들이 global memory를 많이 접근하게 된다. GPU 가 큰 메모리 접근시간 문제를 해결하는 방법에 대해서 설명하여라.

**4. Multiprocessor [15 points]**

(a) 16개의 CPU core가 있는 multiprocessor에서 각 core는 2GHz에 구동이 된다. 각각의 core가 다른 코어에 속한 remote memory를 접근하는데 200 ns이 걸린다고 가정한다. 각 core의 base CPI is 0.5. 만일 1% 의 instructions (코드)가 remote memory를 접근(access)해야 한다면 최종 성능은 얼마인지 CPI로 계산하시오.

**200 ns/0.5 ns = 400 cycle. CPI = 0.5 + 0.01\* 400 = 4.5**

(b) Cache Coherency문제가 무엇인지 설명하여라.

**: Multiprocessor 에서 여러 cache 가 data를 공유할 때 한 cache 의 공유데이터가 수정될 때 inconsistency가 발생하는 문제**

**캐시 일관성**(cache coherence)이란 공유 메모리 시스템에서 각 클라이언트(혹은 프로세서)가 가진 로컬 캐시 간의 일관성을 의미한다.

**각 클라이언트가 자신 만의 로컬 캐시를 가지고 다른 여러 클라이언트와 메모리를 공유하고 있을 때, 캐시의 갱신으로 인한 데이터 불일치 문제가 발생한다.** 예를 들어 변수 X에 대해서 두 클라이언트가 변수 X를 공유하고 있고 그 값이 0이라고 하자. 이때 클라이언트 1(그림의 윗쪽)이 X에 1을 대입하였고 클라이언트 2(그림의 아래쪽)가 변수 X를 읽어들이게 되면 클라이언트 2는 클라이언트 1에 의해 수정된 값인 1을 받아들이는 것이 아니라 현재 자신의 로컬 캐시에 있는 0을 읽어들이게 된다. 따라서 캐시 1, 2는 같은 X라는 변수에 대해 다른 값을 가지게 되므로 데이터 불일치 문제가 발생한다. 캐시 일관성을 유지한다고 하는 것은 이러한 데이터 불일치 현상을 없애는 것을 의미한다. 캐시 일관성을 유지하기 위해서는 다른 프로세서가 갱신한 캐시 값을 곧바로 혹은 지연하여 다른 프로세서에서 사용할 수 있도록 해주어야 한다.

**5. Flash memory [10 points]**

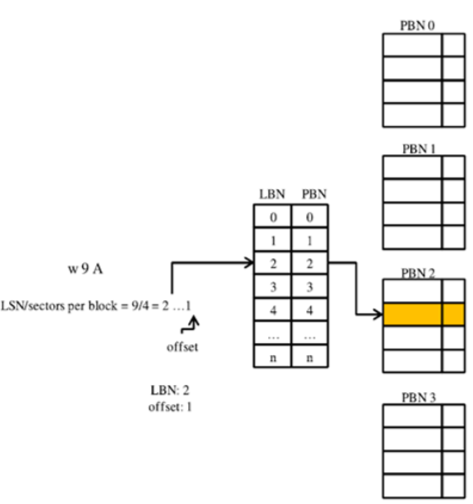
(a) Flash memory의 address 방식중 Page Mapping Method가 Block Mapping Method에 비해 장점과 단점을 기술하여라.

간략한 뜻:

**Page-level Mapping:** NAND Flash Memory의 Page단위로 Address Mapping을 관리한다. **Block-level Mapping:** NAND Flash Memory의 Block단위로 Address Mapping을 관리한다.

**Page-level Mapping Scheme의 장점은 새로 쓰기 요청이 들어온 데이터가 Physical Page(단, Free Page) 어디든지 쓰일 수 있다는 점이다.** 이러한 장점으로 인해 데이터 수정이 요청되었을 때 NAND Flash내에 사용 가능한 페이지(Free Page)가 존재한다면 데이터 카피(copy) 및 소거(erase)동작 없이 바로 쓰기동작이 가능하여, random write access pattern에서 높은 성능을 보이게 된다.  
**Page-level Mapping Scheme의 단점**은 쓰기요청이 계속되어 Free Space이 부족할 경우 invalid한 데이터가 저장되어 있는 space를 지우기 위해 많은 데이터 카피(copy) 및 소거(erase)동작이 한번에 요청된다는 것이다. 이럴 경우 쓰기 성능은 매우 떨어지게 된다. 또 다른 단점은 **Page Mapping Table의 크기가 커져 많은 RAM영역을 차지한다는 것인데**, 이는 embedded system의 가격경쟁력 측면에서 매우 불리하게 만든다.  
  
**Block-level Mapping의 장점은** Page-level Mapping 기법이 크기가 커지는 단점을 극복하기 위하여 제안된 방법으로**Page-level Mapping에 비해 Mapping Size가 작고 Sequential Write Access Pattern의 성능이 좋다는 점이다**. Sequential Write의 성능이 좋은 이유는 갱신하고자 하는 데이터가 모두 연속적인 PBN 및 Page Offset을 가지므로 Block Copy가 발생하지 않기 때문이다.   
**Block-level Mapping의 단점**은 Random Write Access Pattern시 잦은 Block Copy가 발생하므로 성능이 떨어진다. 

(b) 아래의 그림은 Blocking Mapping Scheme 을 나타내고 있다. W 9 B (page 9에 B로 overwrite을 하면 어떤 것들이 바뀌게 되는지 설명하여라.



블럭에는 4개의 page가 있고 9/4(블럭당 page) = 2 +1(나머지=data가 쓰일page 번호)

기존 A가 쓰여있는 곳에 인플레이스 write가 불가해서 다른 프리한 블록의 1번 패이지(같은 오프셋)에 B를 기록하고 2번 블록의 나머지 페이지들도 이동(머지) 하고 2번 블록 통째로 가비지컬렉션 대상(invalid)로 바꾸는거

1. empty상태의 블럭의 1번째 페이지에 A를 쓰고

2. valid상태로 바뀐다음

3. flash라서 같은 곳에 overwrite를 할수없기 때문에 A라고 쓰인 data를 읽은다음에 비어있는 블럭의 empty한 동일한 offset page 1번을 찾아서 B를 쓴다음,

4. A가 원래 쓰여있던 block은 invalid 상태로 만든다