

2017년 2학기 컴퓨터구조 기말고사 (54 점 만점) 2017.12.7.

1. 다음 빈칸을 채워라. (2점 x 12개 = 24점)

- (1) 파이프라이닝은 단일 작업(task)의 실행시간을 줄이지는 않고, 전체 작업부하(workload)의 ( 처리량(throughput) )을 높인다.
- (2) 동적 분기 예측에서 분기 시 타겟 주소를 계산하는 1-사이클 손실을 줄이기 위해 ( 분기 타겟 버퍼(BTB, Branch Target Buffer) )에 이 전에 계산한 타겟 주소를 캐시하여 사용한다.
- (3) 명령어 수준 병렬성을 증가시키는 방법은 파이프라인 깊이를 증가시키는 기법과 ( 다중 이슈 ( multiple issue ) ) 기법이 있다.
- (4) 프로그램은 어떤 특정 시간에는 주소 공간 내의 비교적 작은 부분에만 접근한다는 ( 지역성의 원칙 (principle of locality) )에 기반하여 메모리 계층이 설계된다.
- (5) 캐시의 성능 향상 시키는 기법으로 ( 연관성 (associativity) )을 증가시키는 기법과 다단계 캐싱 기법이 있다.
- (6) 페이지 테이블에서 최근에 사용된 가상 주소에서 실제 주소로 변환을 추적하는 특별한 주소 변환 캐시를 ( TLB (Translation-Lookaside Buffer) )라 한다.
- (7) 가상 메모리의 중요한 기능 중 하나는 여러 ( 프로세스 )가 각각의 주소 공간을 가지고 메인 메모리를 공유하면서도 서로 침범하지 않고 보호(Protection)되도록 하는 것이다.
- (8) ( 스누핑 프로토콜 (snooping protocol) )은 캐시 일관성 유지 (cache coherence) 프로토콜로 프로세서의 각 캐시들이 버스의 읽기/쓰기를 감시하고 쓰기 작업 시 다른 캐시의 복사본을 무효화시킨다.
- (9) 병렬 처리 프로그램의 작성이 어려운 이유로는 작업의 분할, 조정과 스케줄, ( 동기화 시간 최소화 ), 부하 균등(load balancing), 통신 오버헤드 등이 있다.
- (10) 멀티프로세서 구성에는 모든 프로세서가 단일 물리 주소 공간을 갖고 메모리를 공유하는 공유 메모리 멀티프로세서와 각 프로세서가 자신만의 전용 물리 주소 공간을 갖는 ( 메시지 전달 ) 멀티프로세서 구성이 있다.
- (11) ( CUDA )는 사용자가 직접 GPU 프로그래밍을 프로그램을 할 수 있도록 C 언어 기반의 엔비디아(Nvidia)사의 GPU 개발 플랫폼이다.
- (12) ( 클러스터(cluster) )는 독립된 컴퓨터들이 LAN과 같은 입출력 연결망으로 연결되어 작업을 분산 처리하는 컴퓨터 구성이다.

2. 다음 코드에서 아래를 기술하라. (10점)

- lw \$1, 40(\$6) ; ①
- add \$2, \$1, \$3 ; ②
- sub \$10, \$6, \$2 ; ③
- lw \$7, 44(\$9) ; ④

add \$6, \$4, \$7 ; ⑤

(1) 모든 데이터 종속성을 기술하라.

lw-add (①-②), add-sub (②-③), lw-add (④-⑤)

(2) 전방전달(forwarding)이 있다고 가정하고 해저드를 없애는 nop를 코드 사이에 추가하라.

lw \$1, 40(\$6) ; ①

nop

add \$2, \$1, \$3 ; ②

sub \$10, \$6, \$2 ; ③

lw \$7, 44(\$9) ; ④

nop

add \$6, \$4, \$7 ; ⑤

(3) 코드를 이동하여 nop를 없애는 스케줄링을 하여라.

lw \$1, 40(\$6) ; ①

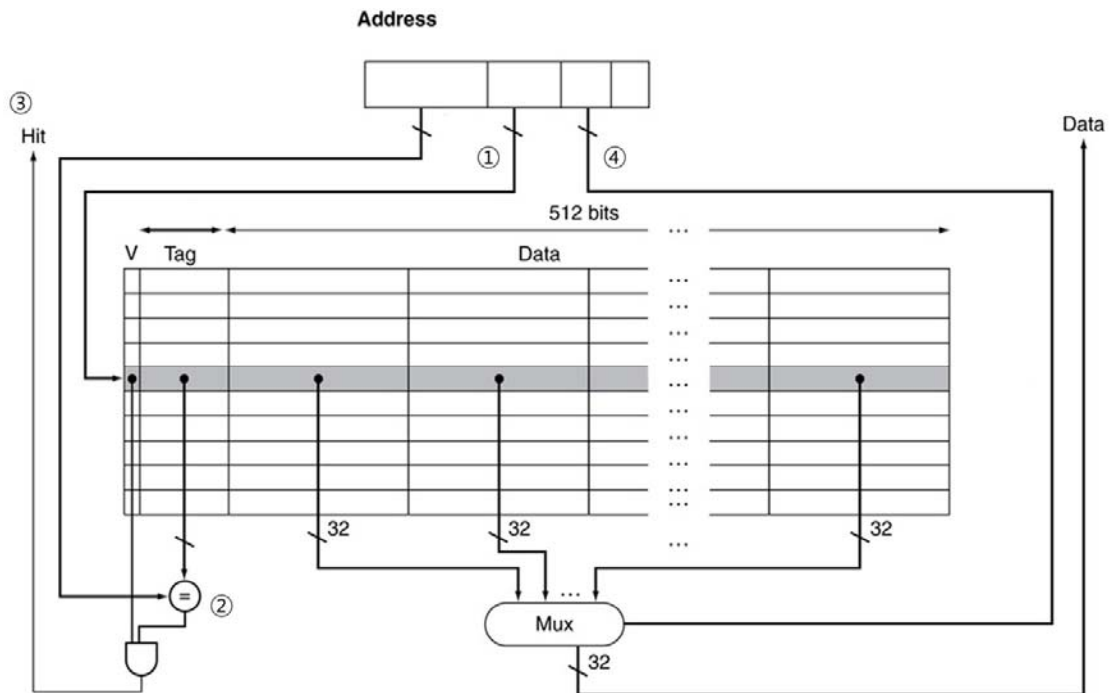
lw \$7, 44(\$9) ; ④

add \$2, \$1, \$3 ; ②

sub \$10, \$6, \$2 ; ③

add \$6, \$4, \$7 ; ⑤

3. 캐시의 데이터 크기가 16 KB, 블록의 크기가 16 워드 (512 비트)인 아래의 캐시 구조에서 다음을 기술하라. (10점)



(1) 캐시의 블록의 수와 32 비트 주소에서 태그, 인덱스, 블록 오프셋, 바이트 오프셋 크기를 계산하라 (계산 과정 포함).

$$\text{블록의 수} = 16 \text{ KB} / 16 \text{ 워드} = (2^4 \times 2^{10}) \text{ 바이트} / (2^4 \times 2^2) \text{ 바이트} = 2^8 = 256 \text{ 블록}$$

$$\text{인덱스} = \log_2 256 = 8 \text{ 비트},$$

$$\text{블록 오프셋} = \log_2 16 = 4 \text{ 비트}$$

$$\text{바이트 오프셋} = \log_2 4 = 2 \text{ 비트}$$

$$\text{태그} = 32 - (8+4+2) = 18 \text{ 비트}$$

(2) 캐시를 참조하는 ① ② ③ ④ 의 순서에 대해 각각 동작을 설명하라.

① : 인덱스 주소로 캐시의 블록을 참조

② : 태그를 비교하여 일치하는지 여부 조사

③ : 태그가 일치하고 유효한 블록이면 캐시 히트

④ : 블록 오프셋 주소로 블록의 16개의 워드 중 하나를 데이터로 선택

4. 10개의 스칼라 변수의 합과 10 X 10 2차원 행렬의 합을 구하는 프로그램에서 프로세서를 추가하였을 때 속도의 향상을 암달(Amdahl)의 법칙을 사용하여 측정하고자 할 때 다음을 기술하라. (10점)

(1) 암달의 법칙이란?

개선 후 실행 시간 = 개선에 영향을 받는 시간 / 개선의 크기 + 영향을 받지 않는 실행 시간

(2) 프로그램에서 순차 처리와 병렬 처리되는 부분

순차 처리: 10개의 스칼라 변수 합

병렬 처리: 10 X 10 2차원 행렬의 합

(3) 100개의 프로세서를 사용하는 경우 속도 향상

(힌트: 프로그램의 수행 시간은 덧셈의 연산 수행 시간을 t라고 할 때, 단일 프로세서의 수행 시간은 순차 처리와 병렬 처리되는 부분의 합하여 구한다. 100개의 프로세서 수행 시간은 암달의 법칙을 적용하여 병렬 처리 부분의 개선을 반영하여 계산한다)

단일 프로세서 수행 시간 = 10t (스칼라 덧셈) + 100t (행렬 덧셈) = 110t

100개의 프로세서 수행 시간 = 100t / 100 + 10t = 11t

속도 향상 = 110t / 11t = 10 배

(4) (3)의 결과가 의미하는 바를 설명

이상적인 속도향상 100배의 10% 밖에 속도 향상을 달성 못하였는데 이는 프로세서 등과 같은 하드웨어 자원을 추가해도 프로그램에서 순차적인 부분에 의해 성능이 제약됨을 보여준다. 즉, 단순한 하드웨어 자원을 추가해서 병렬 처리를 확장하는 어려움을 의미한다.