

문제 1. [클럭] 다음 중 CPU 클럭에 대한 설명으로 옳지 않은 것은?

- ① CPU 클럭은 명령어 실행 타이밍을 결정하는 중요한 신호로, 주파수는 일반적으로 Hz 단위로 표기된다.
- ② 클럭 신호는 CPU 내부의 여러 연산 유닛이 동기화되어 올바른 순서로 작업을 수행하도록 돕는다.
- ③ 최신 CPU는 전력 관리와 성능 최적화를 위해 부하에 따라 클럭 속도를 자동으로 조절하는 기술을 적용한다.
- ④ 대부분의 CPU는 기본적으로 고정된 클럭 주기를 유지하며, 외부 환경 변화에 거의 영향을 받지 않는다.
- ⑤ 클럭 주파수는 CPU가 초당 반복하는 사이클 수를 나타내며, 시스템 성능의 중요한 지표로 활용된다.

문제 2. [코어 & 멀티코어] 다음 중 '코어'와 '멀티코어'에 대한 설명으로 옳지 않은 것은?

- ① 코어는 ALU, 제어장치, 레지스터 등 명령어 실행에 필요한 기본 기능 집합으로 구성된다.
- ② 멀티코어 시스템은 하나의 칩에 두 개 이상의 독립적인 코어를 포함하여, 동시 다중 작업 처리를 가능하게 한다.
- ③ 각 코어는 자체적으로 명령어를 해석하고 실행할 수 있어, 병렬 연산 효율을 높인다.
- ④ 멀티코어 시스템에서는 모든 코어가 항상 동시에 최대 성능으로 활성화되어 작동하는 경향이 있다.
- ⑤ 멀티코어 구조는 여러 작업을 동시에 처리할 수 있어 전체 시스템 성능과 응답성을 향상시킨다.

문제 3. [스레드: 하드웨어 vs 소프트웨어] 다음 중 하드웨어 스레드와 소프트웨어 스레드에 관한 설명으로 부적절한 것은?

- ① 하드웨어 스레드는 하나의 코어 내에서 동시에 여러 명령어 흐름을 처리할 수 있도록 지원하는 기능이다.
- ② 소프트웨어 스레드는 애플리케이션 내 독립적 실행 흐름을 의미하며, OS의 스케줄링에 따라 관리된다.
- ③ 멀티스레드 프로세서는 하드웨어 스레드 기술을 활용해 한 코어 내에서 다수의 스레드를 병렬로 처리할 수 있다.
- ④ 하드웨어 스레드는 CPU 설계에 따라 OS의 스케줄링 영향 없이 항상 독립적으로 동시에 실행된다.
- ⑤ 소프트웨어 스레드는 프로그래머나 운영체제에 의해 생성되고 관리되며, 실행 흐름 제어를 받는다.

문제 4. [멀티스레드 프로세서] 다음 중 멀티스레드 프로세서의 구조에 대해 부적절한 설명은 무엇인가?

- ① 멀티스레드 프로세서는 동시에 여러 스레드를 실행하여 프로세서 자원의 활용도를 높인다.
- ② 각 하드웨어 스레드는 일반적으로 독립적인 레지스터 집합을 할당받아, 서로 간섭 없이 연산을 수행할 수 있다.
- ③ 일부 멀티스레드 아키텍처에서는 특정 캐시나 보조 자원을 스레드 간에 공유하는 경우가 있다.
- ④ 멀티스레드 프로세서는 모든 스레드가 항상 독립적으로 동일한 연산 유닛을 완전히 분리된 방식으로 사용하도록 설계된다.
- ⑤ 이러한 구조 덕분에 여러 작업이 병렬 처리되어 전체 시스템 응답성과 처리 속도가 향상된다.

문제 5. [명령어 병렬 처리 – 파이프라인] 다음 중 명령어 파이프라인 구조에 대한 설명으로 옳지 않은 것은?

- ① 명령어 파이프라인은 인출, 해석, 실행, 결과 저장 등 여러 단계로 나누어 동시에 처리가 가능하도록 설계된다.
- ② 각 파이프라인 단계는 처리 시간이 다소 상이할 수 있어도, 전체적으로 명령어 처리 속도를 높인다.
- ③ 서로 다른 명령어들이 동시에 각기 다른 파이프라인 단계에서 실행되어 처리량이 증대된다.
- ④ 현대 CPU는 순차적 실행 뿐만 아니라 비순차적 실행 기법을 도입해 파이프라인 효율을 개선하고 있다.
- ⑤ 파이프라인 설계에서는 모든 단계가 연속적으로 작동해야 하며, 한 단계의 지연이 전체 실행 순서를 반드시 차질시키는 구조이다.

문제 6. [파이프라인 위험] 다음 중 명령어 파이프라인에서 전형적으로 고려되는 위험 요소에 대해 부적절한 설명은?

- ① 데이터 위험은 명령어 간 필요한 데이터가 아직 준비되지 않아 발생하는 문제를 의미한다.
- ② 제어 위험은 분기나 점프 명령어로 인해 다음 실행할 명령어의 주소를 예측하기 어려워 발생한다.
- ③ 구조적 위험은 여러 명령어가 동시에 제한된 하드웨어 자원을 요구할 때 발생하는 문제이다.
- ④ 캐시 위험은 전형적인 파이프라인 위험 중 하나로, 항상 명령어 실행에 직접적인 지연을 초래한다.
- ⑤ 데이터, 제어, 구조적 위험은 파이프라인 설계 시 주의 깊게 관리해야 하는 핵심 요소로 분류된다.

문제 7. [비순차적 명령어 처리] 다음 중 비순차적(Out-of-Order) 명령어 처리에 관한 설명으로 옳지 않은 것은?

- ① 비순차적 명령어 처리는 명령어 간 의존성이 낮은 경우 효과적으로 적용된다.
- ② 모든 명령어가 독립적이라고 가정하여, 언제나 순서를 임의로 변경해도 결과에 영향을 미치지 않는다.
- ③ 독립적인 연산 자원을 활용해 명령어 순서를 조정함으로써 전체 처리 효율을 높일 수 있다.
- ④ 이 기법은 실행 순서를 재조정하여 파이프라인 내 공백 시간을 최소화하는 데 기여한다.
- ⑤ 비순차적 실행은 의존성이 없는 명령어들의 재정렬을 통해 파이프라인 처리량을 증대시키는 역할을 한다.

문제 8. [CISC vs. RISC] 다음 중 CISC와 RISC 아키텍처에 대한 설명으로 부적절한 것은?

- ① RISC 아키텍처는 설계 단순성 덕분에 언제나 명령어 병렬 처리에서 CISC보다 우수한 성능을 보장한다.
- ② CISC 아키텍처는 다양한 명령어와 가변 길이 명령어 형식을 사용하여 복잡한 연산을 단일 명령어로 수행할 수 있다.
- ③ RISC 아키텍처는 단순화된 명령어 집합과 고정 길이 명령어 형식을 채택해 빠른 명령어 해석과 효율적 파이프라인 구성을 도모한다.
- ④ 현대의 CISC CPU는 내부적으로 명령어를 단순화해 파이프라인 처리를 개선하는 기술을 도입하고 있다.
- ⑤ 현대 CPU는 두 아키텍처의 장점을 적절히 혼합하여 최적의 성능과 전력 효율을 달성하는 방향으로 발전하고 있다.

- 1.정답: ④ 이유: 현대 CPU는 동적 주파수 조절(예: Turbo Boost, Dynamic Frequency Scaling) 기술을 통해 부하나 전력 상황에 따라 클럭 속도를 조정하므로 "대부분의 CPU가 고정된 주기를 유지한다"는 설명은 부정확합니다.
- 2.정답: ④ 이유: 실제로 현대의 멀티코어 CPU는 전력 효율 및 열 관리 등을 위해 필요에 따라 일부 코어만 활성화하거나 클럭을 조절하는 경우가 많으므로 "모든 코어가 항상 동시에 최대 성능으로 작동한다"는 설명은 옳지 않습니다.
- 3.정답: ④ 이유: 하드웨어 스레드 역시 OS의 스케줄러에 의해 실행 순서와 자원 배분이 관리되므로 "OS의 스케줄링 영향 없이 항상 독립적으로 동시에 실행된다"는 설명은 부정확합니다.
- 4.정답: ④ 이유: 실제 설계에서는 핵심 연산 유닛 등 일부 자원은 여러 스레드가 공유하는 경우가 많으며, 완전히 독립적인 분리 구조를 항상 보장하지 않으므로 D 항목은 부적절합니다.
- 5.정답: ⑤ 현대 CPU는 지연을 보완하기 위해 비순차적 실행, 예측 실행 등 다양한 기술을 사용하므로 "모든 단계가 연속적으로 작동해야 하며 한 단계의 지연이 전체 순서를 반드시 차질시킨다"는 설명은 부정확합니다.
- 6.정답: ④ 이유: 캐시 미스는 성능 최적화와 관련된 문제로, 전통적으로 파이프라인 위험(데이터, 제어, 구조적 위험)에는 포함되지 않으므로 ④ 항목의 설명은 부적절합니다.
- 7.정답: ② 이유: 모든 명령어가 독립적인 것은 아니므로, 의존성이 있는 명령어의 경우 순서 변경이 불가능하거나 결과에 영향을 줄 수 있어 D 항목은 옳지 않습니다.
- 8.정답: ① 이유: RISC의 단순한 설계가 병렬 처리에 유리한 점은 있으나, "항상 CISC보다 우수한 성능을 보장한다"는 절대적 주장은 실제 설계 및 응용 환경에 따라 달라지므로 부적절한 설명입니다.