

프로세서와 중앙처리장치

프로세서(Processor)

정의: 프로세서는 데이터를 처리하고 명령을 실행하는 전기적 장치입니다. 일반적으로 CPU를 포함하여, GPU(그래픽 처리 장치)와 같은 다른 처리 장치도 포함될 수 있습니다.

구성: 프로세서는 제어 장치(Control Unit), 산술 논리 장치(ALU), 레지스터 등으로 구성되어 있습니다. 이들 구성 요소는 명령어를 해석하고, 데이터를 처리하며, 결과를 메모리로 전송하는 역할을 합니다.

종류: 프로세서는 범용 프로세서와 특정 목적에 맞춘 전용 프로세서로 나눌 수 있습니다. 예를 들어, 일반적인 컴퓨터에서 사용하는 CPU는 범용 프로세서이며, 특정 작업을 수행하기 위해 설계된 DSP(디지털 신호 프로세서)와 같은 전용 프로세서도 존재합니다.

중앙처리장치(CPU)

정의: 중앙처리장치(Central Processing Unit)는 컴퓨터의 두뇌 역할을 하며, 모든 계산과 명령 처리를 담당합니다. CPU는 프로그램의 명령어를 해석하고 실행하여 컴퓨터의 모든 작업을 수행합니다.

기능: CPU는 메모리에서 명령어를 가져와서 실행하고, 연산을 수행하며, 결과를 다시 메모리에 저장하는 등의 작업을 수행합니다. 이 과정에서 CPU는 기억, 해석, 연산, 제어의 네 가지 주요 기능을 수행합니다.

구성: CPU는 여러 개의 코어를 가질 수 있으며, 각 코어는 독립적으로 명령을 실행할 수 있습니다. 이는 멀티태스킹을 가능하게 하여 여러 작업을 동시에 처리할 수 있게 합니다.

결론

프로세서와 중앙처리장치는 밀접하게 관련되어 있지만, 프로세서는 더 넓은 의미로 사용되는 반면, CPU는 특정한 중앙처리장치를 지칭합니다. CPU는 프로세서의 한 종류로, 컴퓨터의 핵심적인 연산과 제어 기능을 수행하는 장치입니다. 프로세서는 CPU를 포함한 다양한 처리 장치를 포괄하는 용어로 이해할 수 있습니다.

명령어 사이클의 기본 과정

CPU는 다음과 같은 순차적인 단계로 명령어를 처리합니다.

:

1. Fetch (읽기)

- Program Counter(PC)가 가리키는 주소에서 명령어를 가져옴
- 가져온 명령어는 Instruction Register(IR)에 저장

- PC는 다음 명령어를 가리킬 수 있도록 증가

2. Decode (해석)

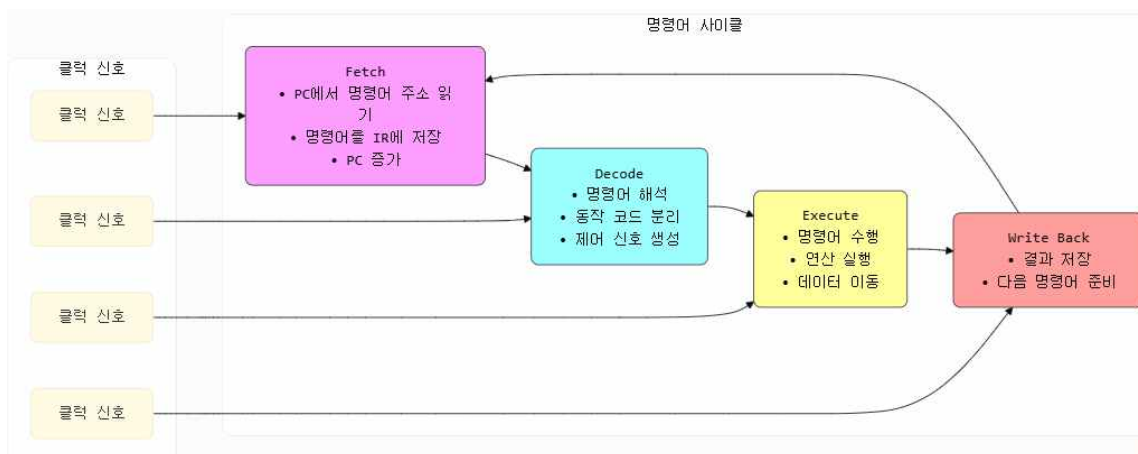
- IR에 저장된 명령어를 해석
- 명령어의 동작 코드(Op-code)와 오퍼랜드 분리
- 해석된 명령어에 따른 제어 신호 생성

3. Execute (실행)

- 해석된 명령어를 실제로 수행
- 산술/논리 연산이나 데이터 이동 수행
- 필요한 경우 메모리 접근

4. Write Back (기록)

- 실행 결과를 레지스터나 메모리에 저장
- 다음 명령어를 위한 준비



위 다이어그램에서 볼 수 있듯이, 각 단계는 클럭 신호에 의해 제어됩니다. 클럭 신호는 CPU의 심장과 같은 역할을 하며, 일정한 간격으로 전기적 펄스를 보내줌으로써 명령어 처리의 타이밍을 조정합니다

주요 구성 요소의 역할

1. 산술/논리 연산 장치 (ALU)

- 산술 연산 (덧셈, 곱셈 등)
- 논리 연산 (AND, OR 등)
- 레지스터의 데이터를 이용하여 연산 수행

2. 제어 장치

- 프로그램 카운터 (PC): 다음 명령어 주소 관리
- 명령어 레지스터 (IR): 현재 수행 중인 명령어 저장

- 명령어 해독기: 명령어 해석 및 제어 신호 생성

3. 레지스터

- IR (Instruction Register): 현재 명령어 저장
- PC (Program Counter): 다음 명령어 주소 저장
- AC (Accumulator): 연산 결과 임시 저장

성능 특성

1. 클럭과 성능

- 클럭 속도: 헤르츠(Hz) 단위로 측정
- 예: 2.5GHz = 1초에 25억 번의 명령어 처리 가능
- 클럭 신호는 명령어 사이클의 타이밍을 제어

2. 명령어 집합 구조

- RISC (Reduced Instruction Set Computer)
 - 단순한 명령어로 구성
 - 한 클럭당 한 명령어 처리
 - 파이프라이닝으로 성능 향상
- CISC (Complex Instruction Set Computer)
 - 복잡한 명령어로 구성
 - 여러 클럭에 걸쳐 처리
 - 하드웨어 구현이 복잡

아래 실제 예시 Assembly Language(어셈블리 언어=어셈블리어)

CISC 방식:

MULT 5:7, 4:3 // 한 번의 명령어로 곱셈과 저장

RISC 방식:

LOAD A, 5:7 // 메모리에서 레지스터로 데이터 로드

LOAD B, 4:3 // 메모리에서 레지스터로 데이터 로드

PROD A,B // 레지스터 간의 곱셈

STORE 5:7, A // 결과를 메모리에 저장

QUIZE

문제 1 : 프로세스란 무엇인가?

ㄱ) 실행 중인 프로그램을 의미한다.

- ㄴ) 하드웨어 장치의 일종이다.
- ㄷ) 메모리의 용량을 나타낸다.
- ㄹ) 운영체제의 종류를 의미한다.

정답: ㄱ)

정답 이유: 프로세스는 메모리에 적재되어 실행 중인 프로그램을 의미합니다. 즉, 프로세스는 프로그램이 실행되기 위해 필요한 자원을 할당받고, CPU에 의해 실행되는 작업의 단위입니다.

문제 2 : 중앙처리장치(CPU)의 주요 기능은 무엇인가?

- ㄱ) 데이터 저장
- ㄴ) 명령어 해석 및 실행
- ㄷ) 사용자 인터페이스 제공
- ㄹ) 네트워크 연결 관리

정답: ㄴ)

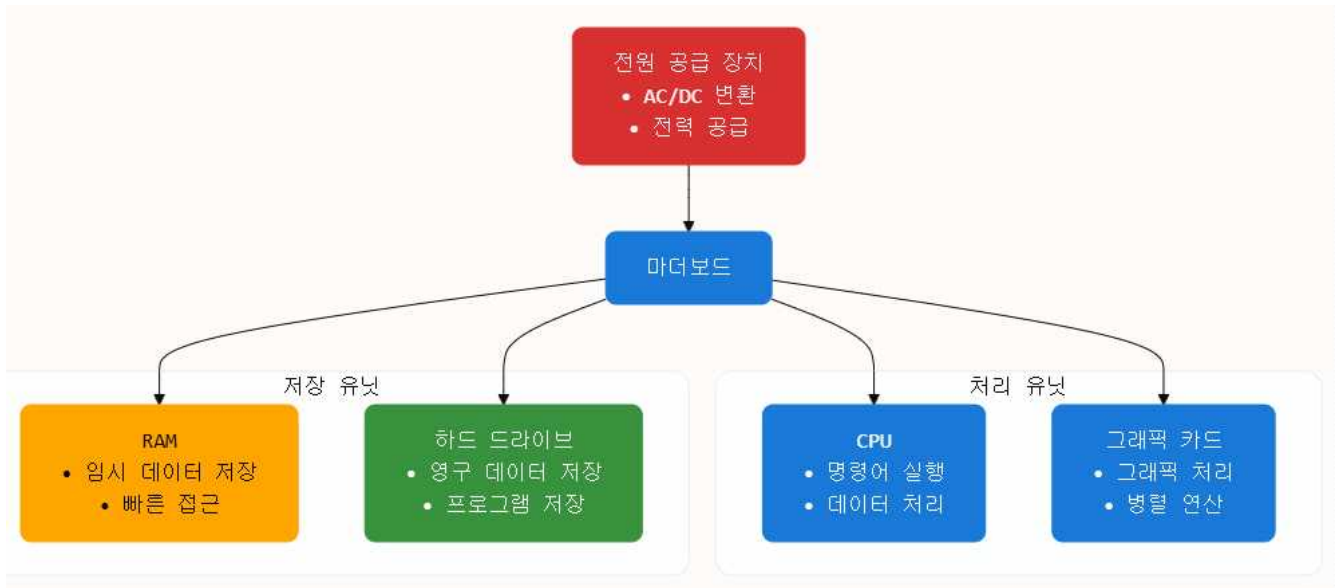
정답 이유: 중앙처리장치(CPU)는 컴퓨터의 두뇌로서, 주어진 명령어를 해석하고 실행하는 역할을 담당합니다. CPU는 기억, 연산, 제어의 기능을 통해 프로그램의 명령을 처리하고 결과를 생성합니다.

문제 3 : 프로세서와 프로세스의 차이점은 무엇인가?

- ㄱ) 프로세서는 소프트웨어이고, 프로세스는 하드웨어이다.
- ㄴ) 프로세서는 프로그램을 실행하는 하드웨어 유닛이고, 프로세스는 실행 중인 프로그램이다.
- ㄷ) 프로세서는 메모리의 용량을 의미하고, 프로세스는 CPU의 속도를 의미한다.
- ㄹ) 프로세서는 운영체제의 종류이고, 프로세스는 하드웨어의 종류이다.

정답: ㄴ)

정답 이유: 프로세서(Processor)는 컴퓨터 내에서 프로그램을 수행하는 하드웨어 유닛을 의미하며, 일반적으로 중앙처리장치(CPU)를 포함합니다. 반면, 프로세스(Process)는 메모리에 적재되어 CPU에 의해 실행 중인 프로그램을 의미합니다. 이 두 용어는 서로 다른 개념으로, 프로세서는 하드웨어적 측면을, 프로세스는 소프트웨어적 측면을 나타냅니다.



하드웨어 유닛 (Hardware unit)

문제 4 : 중앙처리장치(CPU)의 주요 구성 요소가 아닌 것은?

- ㄱ) 연산장치 (ALU)
- ㄴ) 제어장치 (CU)
- ㄷ) 레지스터
- ㄹ) 메모리 관리 장치

정답: ㄹ)

정답 이유: 중앙처리장치(CPU)는 주로 연산장치(ALU), 제어장치(CU), 그리고 레지스터로 구성됩니다. 메모리 관리 장치는 CPU의 일부가 아니며, 주로 운영체제의 역할로 메모리의 할당과 관리를 담당합니다. 따라서 메모리 관리 장치는 CPU의 구성 요소가 아닙니다.

문제 5 : 프로세스의 상태 중 CPU를 할당받기 위해 기다리는 상태는 무엇인가?

- ㄱ) 실행 상태
- ㄴ) 준비 상태
- ㄷ) 대기 상태
- ㄹ) 종료 상태

정답: ㄷ)

정답 이유: 프로세스는 CPU를 할당받기 위해 준비 상태에 있어야 하며, CPU가 사용 중일 경우 대기 상태로 전환됩니다. 대기 상태는 프로세스가 CPU를 기다리는 상태로, 자원을 요청하거나 입출력 작업을 수행하는 동안 CPU를 사용할 수 없는 상태입니다. 따라서 CPU를 할당받기 위해 기다리는 상태는 대기 상태입니다.

문제 6 : 중앙처리장치(CPU)의 주기억장치와의 속도 차이를 극복하기 위해 사용되는 장치는 무엇인가?

- ㄱ) 캐시 메모리
- ㄴ) 레지스터
- ㄷ) 하드 드라이브
- ㄹ) 보조 기억장치

정답: ㄱ)

정답 이유: 캐시 메모리는 CPU와 주기억장치 간의 속도 차이를 극복하기 위해 설계된 고속 메모리입니다. CPU는 캐시 메모리를 통해 자주 사용하는 데이터를 빠르게 접근할 수 있으며, 이는 전체 시스템의 성능을 향상시키는 데 중요한 역할을 합니다. 레지스터는 CPU 내부의 임시 저장소이지만, 캐시 메모리는 CPU와 주기억장치 간의 속도 차이를 줄이는 데 직접적으로 기여합니다.