

1. 프로세서 내부를 구성하고 있는 세 가지 요소는 무엇인가?

산술논리연산장치(ALU) : 각종 산술 연산들과 논리 연산들을 수행하는 회로들로 이루어진 하드웨어 장치

레지스터 : 프로그램을 수행에서 데이터를 일시적으로 저장하는 범용레지스터와 프로그램 수행에서 특수한 기능을 수행하는 특수레지스터가 있다.

제어장치 : 명령어 수행에서 필요한 제어 신호를 발생시키는 장치

2. 시스템 버스에 대하여 각각 설명하라.

데이터버스, 주소버스, 제어버스

3. 프로세서에서 범용 레지스터의 역할은 무엇인가?

데이터 또는 명령을 일시적으로 저장하는 역할

4. 레지스터를 전용과 범용으로 나눈다면 전용 레지스터의 예를 들어라.

프로그램카운터(PC) :

누산기(AC)

상태레지스터

명령어레지스터

주소레지스터

데이터레지스터

5. 레지스터 중에서 다음에 인출할 명령어의 주소를 담고 있는 것은 무엇인가?

프로그램카운터

6. 명령어 레지스터의 역할을 설명하라.

인출한 명령어를 일시적으로 저장하는 레지스터

7. 프로그램 카운터(PC)란 무엇인가?

다음 인출해올 명령어 주소를 저장하고 있는 레지스터

8. 명령어 실행 후에 각종 상태를 표시하는 레지스터는 무엇인가? 또한 표시하는 상태에는 어떤 것들이 있는가?

상태레지스터, Zero, 음수, Overflow, 캐리 등의 상태를 나타냄

9. 스택은 어떤 경우에 사용하는가? 이때 스택을 사용하면 어떤 장점이 있는가?

인터럽트처리, 서브루틴 콜과 리턴, 0-주소명령어 등을 사용

스택은 가장 최근에 저장한 자료부터 순서대로 사용해야하는 경우에 적합한 구조이다.

10. 외부 메모리를 이용해 스택을 구현하려고 한다면 이를 위해서 프로세서에서 구조적으로 지원해 줘야 하는 것들에는 어떤 것이 있는가?

11. 명령어 실행 사이클을 4단계로 구분한다면 이들은 무엇인가?

명령어 인출

명령어 해독

명령어 실행

결과 저장

12. 메모리에 저장된 프로그램의 일부를 프로세서로 읽어들이는 과정을 명령어 실행 사이클에서 무엇이라고 부르는가?

명령어 인출

13. 프로세서 내부의 각종 장치가 어떻게 동작해야 하는지를 제어하는 데 필요한 신호를 만들기 위해서는 어떤 명령어가 명령어 레지스터

에 저장되어 있는지를 알아야 한다. 어떤 종류의 명령어인지 알아내기 위한 이 과정을 명령어 실행 사이클에서 무엇이라고 부르는가?

명령어 해독

14. 덧셈 명령어를 실행하기 위해서는 덧셈을 당하는 피연산자가 필요하다. 이 데이터가 메모리에 저장되어 있다면, 이 피연산자를 프로세서로 불러들이는 과정은 명령어 실행 사이클의 어느 과정에서 처리돼야 하는가?

프로세서마다 차이가 있지만, 이 경우에는 명령어 실행 이전에 피연산자를 인출한다.

15. 프로세서에서 명령어 실행 사이클의 1단계 명령어 인출을 마친 상태에서 외부 장치에 의해서 인터럽트가 발생했다. 프로세서는 언제 이 인터럽트 처리를 해야 하는가?

명령어 수행 중에 인터럽트가 발생하며 현재 수행 중인 명령어를 종료하고 인터럽트를 수행한다. 명령어 실행 사이클의 마지막에 인터럽트 요구를 체크한다.

16. 명령어에 대해서 네 가지 세부 단계가 실행될 때, 프로그래머가 관여해 세부 단계의 실행을 멈추거나 지연시키는 등의 행위를 할 수 있는가?

프로그래머가 세부 단계를 실행을 멈추거나 지연시키는 권한은 없다.

17. 명령어 형식을 크게 두 부분으로 구분하면 무엇인가?

연산코드, 한 개이상의 오퍼랜드

18. 명령어가 어떤 연산을 수행해야 하는지를 지정하는 부분을 무엇이라고 하는가?

연산코드

19. 명령어 형식에서 연산을 수행하는 데 필요한 데이터나 데이터가 들어 있는 곳의 주소를 표시하는 부분을 무엇이라고 하는가?

오퍼랜드

20. 많은 연산을 지원하고 큰 메모리를 참조할 수 있도록 명령어의 크기를 크게 만들려고 한다. 어떤 단점이 있는가?

명령어형식의 길이가 길어짐,

21. 명령어 형식을 설계하는 데 있어서 오퍼랜드가 차지하는 필드를 1개만 두려고 한다. 이를 위해서는 구조적으로 어떤 지원이 있어야 하는가?

오퍼랜드가 1개 포함하는 1-주소 명령어에서는 모든 명령은 누산기에 저장되어 있는 데이터를 사용한다. 누산기는 하나밖에 없으므로 누산기에 데이터가 저장되어 있음을 나타낼 필요가 없다.

연산에서 피연산자의 하나는 누산기에 저장하고, 연산 결과값도 누산기에 저장한다.

22. 명령어 연산 코드가 8비트면, 이 프로세서가 수행할 수 있는 연산의 종류는 최대 몇 가지인가?

최대 연산의 종류는 256개

23. 명령어 형식에서 오퍼랜드 10비트에 2의 보수 형태로 데이터를 저장한다면, 그 데이터의 표현 범위는 얼마인가?

-512 ~ +511

24. 명령어 형식에서 0-주소명령어, 1-주소 명령어, 2-주소 명령어, 3-주소 명령어으로 나눌 때 기준이 되는 것은 무엇인가?

오퍼랜드 수

24. 0-주소 명령어 형식을 이용한 연산을 하기 위해서는 어떤 자료구조를 응용해야 가능한가?

스택 구조

25. 1-주소 명령어에서 연산 코드가 6비트고, 10비트의 오퍼랜드가 메모리의 주소를 지정한다면 메모리 용량은 최대 얼마인가?

1024Byte

26. 명령어 형식의 주소 필드의 개수가 시스템 성능에 미치는 영향을 설명하라.

명령어 형식의 비트가 길어지면 하나의 명령어를 수행하기 위해서 더 많은 메모리로 접근해야하므로 수행시간이 길어진다.

27. 시프트 명령어는 데이터 이동에 관한 명령어의 범주에 속하는가, 아니면 산술논리 연산의 범주에 속하는가? 그 이유는 무엇인가?

산술논리연산장치에 속한다.

28. 명령어 형식에서 많은 종류의 주소지정 방식을 두는 이유는 무엇인가?

프로세서는 한번에 처리할 비트를 크기는 그 프로세서의 워드의 크기와 일치하는 것이 일반적이다. 제한된 수의 명령어 비트를 이용하여 다양한 방법으로 오퍼랜드를 지정하고, 메모리의 접근도 더 큰 용량의 메모리로 접근하도록 다양한 주소지정방식을 둔다.

29. 다음 명령어에서 R1에 저장되는 값은 무엇인가?

MOV R1, #300

R1 = 300

30. 즉시 주소지정 방식의 장점은 무엇인가? 또한 한계는 무엇인가?

상수를 정의 하거나, 변수의 초기값을 설정할 때 사용하는 것으로 오퍼랜드에 숫자 데이터를 직접 넣은 것으로 별도의 메모리 접근을 할 필요가 없어 빠르게 실행할 수 있다. 한계로는 오퍼랜드의 필드의 비트수가 사용할 수 있는 숫자의 크기를 제한한다.

31. 직접 주소지정 방식이 메모리에 들어 있는 데이터를 가져오기 위한 가장 쉬운 방법인 것 같다. 이 방법의 문제점은 무엇인가?

오퍼랜드의 내용이 메모리 주소를 지정하고 있는 유효주소이므로 메모리에 저장된 데이터를 인출하기 위해서는 한 번의 메모리 접근이 필요하다. 따라서 메모리 유효주소를 계산하는 과정이 없이 간단한 장점이 있다.

단점은 오퍼랜드의 내용이 메모리의 주소가 되므로 오퍼랜드의 비트 수가 제한되어 있으므로 직접 접근할 메모리 공간이 제한된다는 것이 단점이다.

32. 메모리를 이용한 간접 주소지정 방식을 사용하면 큰 메모리라도 모두 접근할 수 있다. 단점은 무엇인가?

메모리를 두 번 접근함으로써 실행 시간이 많이 걸린다.

33. 레지스터를 이용한 간접 주소지정 방식의 장점을 설명하라.

레지스터 간접 주소지정 방식은 오퍼랜드에는 레지스터 번호를 지정하고, 지정된 레지스터의 내용은 메모리 주소를 나타낸다. 레지스터 비트의 크기에 따라서 메모리로 접근할 용량이 결정된다. 메모리 데이터를 인출하기 위해서 한 번의 메모리 접근이 필요하다.

34. 상대 주소지정 방식은 어떤 경우에 사용되는가?

변위 값은 현재 실행 중인 명령어 위치를 기준으로 한 상대적인 값으로 분기 명령, 점프 명령에 사용된다.

35. 인덱스 레지스터를 이용한 변위 주소지정 방식은 어떤 경우에 사용되는가?

메모리에 저장된 데이터 배열의 시작 주소를 지정할때 변위를 사용한다. 인덱스레지스터의 값은 배열의 시작주소로부터 각 데이터까지의 크기를 나타낸다.

36. 상대 주소지정 방식과 변위 주소지정 방식을 비교 설명하라.

변위주소지정방식에서 두 개의 오퍼랜드를 가지며, 첫 번째 오퍼랜드는 레지스터의 번호를 지정하고, 두 번째 오퍼랜드는 변위를 나타낸다. 유효주소는 레지스터의 값과 변위를 더한 값이 된다.

상대주소지정방식에서 유효 주소는 오퍼랜드의 변위 값과 프로그램카운터(PC) 값을 더한 값이다.

37. 다음과 같은 레지스터와 메모리 주소에 대한 데이터 값이 주어졌다. 각 명령어를 실행한 후 레지스터 R1 값을 구하라.

레지스터 값 R1 = 100, R2 = 200, PC = 100

- | | |
|-----------------------|-----|
| (1) MOV R1, #100 즉치 | 100 |
| (2) MOV R1, (100) 간접 | 70 |
| (3) MOV R1, [R2] 레지스터 | 60 |
| (4) MOV R1, 1[R2] 변위 | 70 |
| (5) MOV R1, 2(PC) | 40 |

메모리 주소	데이터
99	10
100	201
101	302
102	40
199	50
200	60
201	70
202	90

38. 30비트의 주소로 지정할 수 있는 메모리 장소의 최대 수는 얼마인가?

$$2^{30} = 1\text{Gbyte}$$

39. 메모리의 LOAD, STORE 동작을 설명하라.

메모리의 LOAD :

- 메모리로부터 읽을 데이터워드 주소를 MAR(Memory Address Register)에 적재한다.
- 프로세서에서 'Read'라는 신호를 메모리로 전달하며 MAR에 위치한 데이터를 읽어 나와 MDR(Memory Data Register)로 적재한다.
- 프로세서는 MDR의 저장된 데이터를 사용하게 된다.

메모리의 STORE :

- 프로세서는 메모리에 저장한 데이터를 먼저 데이터 레지스터(MDR)에 적재한다.
- 프로세서는 데이터를 저장할 메모리 주소를 MAR에 적재한다.
- 프로세서의 'Write' 제어신호는 MDR에 있는 데이터를 메모리 주소 MAR 위치에 저장한다.

40. 명령어의 분류 및 기능을 설명하라.

명령어의 종류는 산술논리연산 명령어, 데이터이동 명령어, 제어명령어, 입출력 명령어 등으로 분류할 수 있다.