THEME
111 명령어 형식

1회 ▶ 11-1

다음은 명령어 형식에 대한 설명이다. 옳은 것은?

① 명령은 보통 OP 코드부분과 오퍼랜드 부분으로 나누며 오퍼랜드는 수행해야 할 동작을 명시하는 부분이고, OP코드는 연산의 대상물이다.

② 기억장치의 주소나 레지스터를 지정하거나 실제 데이터 값을 가지고 있는 부분이 오퍼랜드이다.

③ 오퍼랜드의 비트 수가 n 비트인 경우 2^n 가지의 서로 다른 동작을 수행할 수 있다.

④ 오퍼랜드에는 유효번지를 결정하기 위한 모드 비트를 가질 수 없다.

핵심이론

명령어 형식(Instruction Format)

- 명령어는 연산자부(Operation Code)와 자료부(Operand)로 구성된다.
- 명령어는 자료부의 개수에 따라 0-주소 명령어, 1-주소 명령어, 2-주소 명령어, 3-주소 명령어 형식으로 구분된다.

유사문제

1회 ▶ 산 04-2

1. 한 명령을 두 부분으로 나누면?

① 호출과 실행

② 연산과 논리

③ 번지와 데이터

④ operation과 operand

1회 ▶ 02-4

2. computer 시스템에서 1-address machine, 2-address machine, 3-address machine으로 나눌 때 기준이 되는 것은?

① operation code

② 기억장치의 크기

③ register 수

④ operand의 address

1회 ▶ 03-1

3. 명령어 형식(instruction format)이 opcode, addressing mode, address의 3부분으로 되어있는 컴퓨터에서 주기억장치가 1024 워드일 경우 명령의 크기는 몇 비트로 구성되어야 하는가? (단, op-code는 4비트이며, addressing mode는 직접 및 간접 주소지정 방식 구분에만 사용한다라고 가정한다.)

① 10

② 15

③ 20

④ 25

[정답] 핵심문제 ② / 유사문제 1. ④ 2. ④ 3. ②

THEME
112 0-주소 명령어

2회 ▶ 03-1, 00-3
0-주소 인스트럭션 형식을 사용하는 컴퓨터의 특징은?
① 연산 후에 입력 자료가 변하지 않고 보존된다.
② 연산에 필요한 자료의 주소를 모두 구체적으로 지정해 주어야 한다.
③ 모든 연산은 스택에 있는 자료를 이용하여 수행한다.
④ 연산을 위해 입력 자료의 주소만을 지정해 주면 된다.

핵심이론

0-주소 명령어(0-Address Instruction)

OP-Code

- 기억장치에 접근할 Operand부가 없이 OP-code부만으로 구성된다.
- 명령어의 길이가 짧아서 기억공간을 적게 차지한다.
- 모든 연산은 스택(Stack)에 있는 자료를 이용하여 수행되며, 연산의 결과도 스택에 저장된다.
- 스택 포인터(Stack Pointer)가 가리키는 Operand를 이용하기 때문에 자료의 주소를 지정할 필요가 없다.
- 연산에 사용된 모든 피연산자 값을 상실한다.
- 0-주소 명령어 형식을 갖는 컴퓨터 구조 원리를 스택 구조(Stack Architecture)라고 한다.

유사문제

2회 ▶ 산 08-1, 05-1추
1. 인스트럭션 형식 중 자료의 주소를 지정할 필요가 없는 형식은?
① 1-주소 ② 2-주소 ③ 3-주소 ④ 0-주소

3회 ▶ 06-1, 05-2, 99-1
2. 0-주소 인스트럭션에 필요한 것은?
① 스택(Stack) ② 색인 레지스터(index register)
③ 큐(queue) ④ 기본 레지스터(base register)

1회 ▶ 산 14-3
3. 0-주소 명령어의 형식에서 결과 자료는 어디에 저장되는가?
① 스택 ② 누산기 ③ 범용 레지스터 ④ 명령 레지스터

3회 ▶ 산 08-4, 04-2, 01-2
4. 명령 형식 중에서 스택(stack)을 필요로 하는 것은?
① 3주소 명령어 ② 2주소 명령어
③ 1주소 명령어 ④ 0주소 명령어

1회 ▶ 산 05-1
5. PUSH, POP 명령어 처리와 가장 가까운 명령어 형식은?
① 0-주소 명령어 ② 1-주소 명령어
③ 2-주소 명령어 ④ 3-주소 명령어

4회 ▶ 05-4, 03-2, 산 02-2, 99-1
6. stack이 갖는 주소지정 방식은?
① 0-Address ② 1-Address
③ 2-Address ④ 3-Address

6회 ▶ 06-2, 05-1, 04-1, 02-2, 01-1, 00-1
7. 0-번지 명령형(zero-address instruction format)을 갖는 컴퓨터 구조 원리는?
① An accumulator extension register
② Virtual memory architecture
③ Stack architecture
④ Micro-programming

2회 ▶ 02-1, 99-2
8. 스택 머신(stack machine)은?
① zero address machine ② one address machine
③ two address machine ④ three address machine

1회 ▶ 03-4
9. 기억장치의 구조가 stack 구조를 가질 때 가장 밀접한 관계가 있는 명령어는?
① one-address 명령어 ② two-address 명령어
③ three-address 명령어 ④ zero-address 명령어

1회 ▶ 07-4
10. Stack 구조의 컴퓨터에서 사용하는 연산 명령어의 주소지정 방식은?
① 0-Address ② 1-Address ③ 2-Address ④ 3-Address

1회 ▶ 산 03-2
11. 데이터 연산시 스택(stack)만을 사용하는 인스트럭션은?
① zero-address ② one-address ③ two-address ④ three-address

[정답] 핵심문제 ③ / 유사문제 1. ④ 2. ① 3. ① 4. ④ 5. ① 6. ① 7. ③ 8. ① 9. ④ 10. ① 11. ①

THEME
113 Stack 구조

2회 ▶ 산 11-3, 99-1

0-주소 인스트럭션 형식의 특징이 아닌 것은?

① 연산을 위한 스택을 갖고 있으며, 연산 수행 후에 결과를 스택(stack)에 보존한다.
② 자료를 얻기 위하여 스택에 접근할 때는 top이 지정하는 곳에 접근한다.
③ unary 연산 경우에는 2개의 자료가 필요하고, top이 지정하는 곳의 자료를 처리하고, 결과는 top 다음 위치에 기억한다.
④ binary 연산인 경우 2개 자료가 필요하고, 스택 상단부 2자리에 저장한다.

핵심이론

스택(Stack)

- 자료의 삽입, 삭제가 top이라고 부르는 한쪽 끝에서만 이루어지는 자료 구조이다.
- 가장 나중에 삽입된 자료가 가장 먼저 삭제되는 후입선출(LIFO ; Last In First Out) 방식으로 자료를 처리한다.
- 스택에 자료를 삽입하는 명령은 Push, 스택에서 자료를 인출하는 명령은 Pop이다.

유사문제

2회 ▶ 산 02-3, 00-4

1. 명령어의 형식 가운데 연산에 사용된 모든 피연산자 값을 상실하는 명령어 형식은?

① 3-주소 형식 명령어 ② 2-주소 형식 명령어
③ 1-주소 형식 명령어 ④ 0-주소 형식 명령어

1회 ▶ 09-4

2. 스택 컴퓨터의 특성에 대한 설명으로 옳지 않은 것은?

① 피연산자를 나타내지 않기 때문에 인스트럭션의 길이가 짧아서 기억 공간의 이용이 효율적이다.
② 스택에 기억된 데이터만을 이용하여 연산하므로 인스트럭션 수행 시간이 짧다.
③ 함수 연산에 필요한 데이터를 미리 처리되는 순서대로 기억시켜 놓아 편리하다.
④ 스택에 레지스터의 수가 적을 때에는 전달 기능의 인스트럭션인 PUSH와 POP를 사용해야 되는 비효율성이 있다.

1회 ▶ 산 05-4

3. 스택(Stack)에 관한 설명 중 옳지 않은 것은?

① 역 폴리쉬형의 산술식을 처리하는데 효과적이다.
② 재귀적 프로그래밍에서 복귀 주소를 저장하는데 효과적이다.
③ LIFO라고 부르기도 한다.
④ 스택은 스택 포인터를 사용함으로써 1-주소 방식의 명령어 처리에 적합하다.

1회 ▶ 산 04-1

4. STACK에 관하여 올바르게 설명한 것은?

① 복귀 번지를 저장하기 위한 메모리이다.
② PUSH 명령으로 의해 데이터를 꺼낸다.
③ 1-address 구조를 갖는다.
④ FIFO 구조를 갖는다.

1회 ▶ 산 12-1

5. 부프로그램(Sub-program)에서 주프로그램(Main-program)으로 복귀할 때 필요한 주소를 기억할 때 적합한 것은?

① Queue ② Dequeue ③ Stack ④ Buffer

1회 ▶ 산 01-1

6. PUSH, POP 명령어 처리시 처리되는 메모리 주소는?

① 스택의 데이터 ② AX의 데이터
③ SI의 데이터 ④ DI의 데이터

5회 ▶ 14-2, 14-1, 11-3, 01-2, 산 05-1추

7. 다음 중 피연산자의 위치(기억장소)에 따라 명령어 형식을 분류할 때 instruction cycle time이 가장 짧은 것은?

① 레지스터-메모리 instruction ② AC instruction
③ 스택 instruction ④ 메모리-메모리 instruction

1회 ▶ 산 04-4

8. 일반적인 컴퓨터의 CPU 구조 가운데 처리할 수식을 미리 처리되는 순서인 역 polish(또는 postfix) 형식으로 바꾸어야 하는 CPU 구조는?

① 범용 레지스터 구조 CPU ② 단일 누산기 구조 CPU
③ 스택 구조 CPU ④ 모든 CPU 구조

3회 ▶ 10-4, 03-4, 산 10-2

9. 일반적인 컴퓨터의 CPU 구조 가운데 수식을 계산할 때 수식을 미리 처리되는 순서인 역 polish(또는 postfix) 형식으로 바꾸어야 하는 CPU 구조는?

① 단일 누산기 구조 CPU ② 범용 레지스터 구조 CPU
③ 스택 구조 CPU ④ 모든 CPU 구조

[정답] 핵심문제 ③ / 유사문제 1. ④ 2. ③ 3. ④ 4. ① 5. ③ 6. ① 7. ③ 8. ③ 9. ③

THEME
114 1-주소 명령어

1회 ▶ 07-2

1-주소 명령어의 특징으로 올바른 것은?

① 모든 명령은 누산기에 기억되어 있는 자료를 사용한다.
② 스택의 사용이 필수적이다.
③ 2개의 오퍼랜드를 가지고 있다.
④ 2-주소 명령어와 비교하여 프로그램의 길이가 최소 2배 이상이 된다.

핵심이론

1-주소 명령어(1-Address Instruction)

OP-Code	Operand 1

- OP-code부와 1개의 Operand부로 구성된다.
- 모든 연산은 누산기(Accumulator)에 의해 수행되며, 연산의 결과도 누산기에 저장된다.

유사문제

2회 ▶ 05-2, 산 03-1

1. 연산 결과를 항상 누산기(Accumulator)에 저장하는 명령어 형식은?

① 0-주소 명령어 ② 1-주소 명령어
③ 2-주소 명령어 ④ 3-주소 명령어

4회 ▶ 14-1, 12-2, 07-1, 99-3

2. 명령어의 구성 형태 중 하나의 오퍼랜드만 포함하고 다른 오퍼랜드나 결과값은 누산기에 저장되는 명령어 형식은?

① 0-주소 명령어 ② 1-주소 명령어
③ 2-주소 명령어 ④ 3-주소 명령어

1회 ▶ 04-1

3. 반드시 누산기가 필요한 주소지정 방식은?

① 0-Address 주소지정 방식
② 1-Address 주소지정 방식
③ 2-Address 주소지정 방식
④ 3-Address 주조지정 방식

2회 ▶ 산 04-2, 00-1

4. 주소 부분이 하나밖에 없는 1-주소 명령 형식에서 결과 자료를 넣어 두는데 사용하는 레지스터는?

① 어큐뮬레이터(accumulator)
② 인덱스(index) 레지스터
③ 범용 레지스터
④ 스택(stack)

1회 ▶ 13-3

5. 1-주소 명령어에서는 무엇을 이용하여 명령어 처리를 하는가?

① 누산기
② 가산기
③ 스택
④ 프로그램 카운터

[정답] 핵심문제 ① / 유사문제 1. ② 2. ② 3. ② 4. ① 5. ①

THEME
115 1-주소 명령어 형식

3회 ▶ 산 10-1, 06-2, 04-2

다음과 같은 명령어는 어떤 유형의 번지 명령 방식인가?

```
LOAD A    ADD B    SOTRE C
```

① 0-주소
② 1-주소
③ 2-주소
④ 3-주소

핵심이론

1-주소 명령어 형식

예 C = A + B

LOAD A	AC ← M[A]
ADD B	AC ← AC + M[B]
STORE C	M[C] ← AC

유사문제

1회 ▶ 산 07-2

1. 다음 명령어 중 형식이 다른 것은?

① ADD A
② SUB A
③ PUSH A
④ LOAD A

1회 ▶ 산 09-2

2. 다음과 같이 산술식으로 표현된 명령을 누산기를 이용하는 1-주소 명령으로 옳게 표현한 것은?

```
X = ( A + B ) * C
```

① LOAD A
ADD B
MUL C
STORE X

② LOAD B
MUL C
ADD A
STORE X

③ ADD A
LOAD B
MUL C
STORE X

④ ADD A, B
MUL C
STORE X

1회 ▶ 11-2

3. 다음 중 1-주소 명령어 형식을 따르는 명령어 MUL A를 가장 적절하게 설명한 것은? (단, M[A]는 기억장치와 A번지의 내용을 의미하고 MUL은 곱셈을 나타낸다.)

① AC ← AC × M[A]
② R1 ← R2 × M[A]
③ AC ← M[A]
④ M[A] ← AC

2회 ▶ 산 10-4, 00-2

4. 다음 명령어 형식에 대한 특성 중 옳지 않은 것은?

① 3-주소 명령어 형식은 명령어 길이가 증가한다.
② 2-주소 명령어 형식은 오퍼랜드가 2개 필요하다.
③ 1-주소 명령어 형식은 스택이 필요하다.
④ 0-주소 명령어 형식은 PUSH, POP 명령이 필요하다.

[정답] 핵심문제 ② / 유사문제 1. ③ 2. ① 3. ① 4. ③

THEME
116 2-주소 명령어

2회 ▶ 산 02-2, 01-1

계산 결과를 시험할 필요가 있을 때 계산 결과가 기억장치에 기억 될 뿐 아니라 중앙처리장치에도 남아 있어서 중앙처리장치 내에서 직접 시험이 가능하므로 시간이 절약되는 인스트럭션 형식은?

① 3주소 인스트럭션 형식
② 2주소 인스트럭션 형식
③ 1주소 인스트럭션 형식
④ 0주소 인스트럭션 형식

핵 심 이 론

2-주소 명령어(2-Address Instruction)

OP-Code	Operand 1	Operand 2

- OP-code부와 2개의 Operand부로 구성된다.
- 연산의 결과는 Operand 1에 기록되므로 Operand 1에 기록되어 있던 원래의 자료가 지워진다.
- 계산 결과를 시험할 필요가 있을 때 계산 결과가 기억장치에 기억될 뿐 아니라 중앙처리장치에도 남아 있어서 중앙처리장치 내에서 직접 시험이 가능하므로 시간이 절약된다.

유 사 문 제

1회 ▶ 산 12-3

1. 명령어 형식 중 컴퓨터에서 가장 널리 사용되는 형식으로, 입력 자료가 연산된 후에는 보존되지 않지만 실행속도가 빠르고 기억 장소를 많이 차지하지 않는 형식으로 오퍼랜드 1의 내용과 오퍼랜드 2의 내용을 더해 오퍼랜드 1에 기억시키는 형식은?

① 0-address 명령 형식
② 1-address 명령 형식
③ 2-address 명령 형식
④ 3-address 명령 형식

1회 ▶ 09-1

2. 2-주소 명령어 형식으로 다음 연산을 표와 같이 수행했을 때 각 ( )에 알맞은 것은? (단, R1, R2은 레지스터를 나타냄)

[연산] Y = (A + B) * (C + D)

연산코드	주소필드1	주소필드2
MOV	R1	A
ADD	R1	B
MOV	R2	C
ADD	R2	D
MUL	R1	R2
(가)	(나)	(다)

① (가) : MOV, (나) : Y, (다) : R1
② (가) : MOV, (나) : A, (다) : B
③ (가) : ADD, (나) : Y, (다) : R1
④ (가) : ADD, (나) : A, (다) : B

[정답] 핵심문제 ② / 유사문제 1. ③ 2. ①

THEME
117 3-주소 명령어

1회 ▶ 산 03-4

여러 개의 범용 레지스터를 가진 컴퓨터에 사용되며, 연산 후에 입력 자료가 변하지 않고 보존되는 인스트럭션의 형식은?

① 0주소 인스트럭션의 형식
② 1주소 인스트럭션의 형식
③ 2주소 인스트럭션의 형식
④ 3주소 인스트럭션의 형식

핵심이론

3-주소 명령어(3-Address Instruction)

OP-Code	Operand 1	Operand 2	Operand 3

- OP-code부와 3개의 Operand부로 구성된다.
- 연산의 결과는 Operand 1(또는 Operand 3)에 기록되므로 연산 후에 입력 자료가 변하지 않고 보존된다.
- 여러 개의 범용 레지스터를 가진 컴퓨터에 사용된다.
- 전체 프로그램의 길이는 짧아지지만, 명령어 한 개의 길이는 길어진다.

유사문제

2회 ▶ 산 09-4, 06-4

1. 명령 구성 형식 중 연산에 이용된 자료가 연산 후에도 기억장치에 그대로 보존되는 형식은?

① 1-주소 명령형식
② 2-주소 명령형식
③ 3-주소 명령형식
④ 0-주소 명령형식

1회 ▶ 06-2

2. 3-주소 명령어의 장점에 해당하는 것은?

① 프로그램 길이가 짧아진다.
② 1개의 명령어만을 사용하여 프로그램을 작성해야 한다.
③ 주소지정을 할 수 있는 기억장치 주소 영역이 증가한다.
④ 임시 저장 장소가 필요하다.

2회 ▶ 09-1, 01-3

3. 3-주소 명령어의 설명으로 옳은 것은?

① 결과는 1st operand에 남는다.
② 결과는 2nd operand에 남는다.
③ 결과는 3rd operand에 남는다.
④ 결과는 임시 구역에 남는다.

1회 ▶ 11-3

4. 3주소 명령어 연산에서 결과는 어디에 저장되는가?

구조

op	operand1	operand2	operand3

① PC(Program Counter)
② stack
③ operand1
④ 임시저장장소

[정답] 핵심문제 ④ / 유사문제 1. ③ 2. ① 3. ③ 4. ③

THEME
118 인스트럭션의 성능

2회 ▶ 14-2, 07-4

명령어 수행시간이 10ns이고, 명령어 패치시간이 5ns, 명령어 준비시간이 3ns이라면 인스트럭션의 성능은 얼마인가?

① 0.1
② 0.3
③ 0.5
④ 1.25

핵심이론

인스트럭션의 성능

- 인스트럭션 성능 = 수행시간 / (패치시간 + 준비시간)

  예 인스트럭션 수행시간이 15ns, 인스트럭션 패치시간이 3ns, 인스트럭션 준비시간이 2ns일 경우
  인스트럭션 성능 = 15ns / (3ns+2ns) = 3

유사문제

1회 ▶ 08-4

인스트럭션 수행 시간이 20ns이고, 인스트럭션 패치시간이 5ns, 인스트럭션 준비시간이 3ns이라면 인스트럭션의 성능은 얼마인가?

① 0.4
② 0.6
③ 2.5
④ 4.0

[정답] 핵심문제 ④ / 유사문제 1. ③

THEME
119 주소 설계 시 고려사항

2회 ▶ 00-2, 99-2

전자계산기 기억장치의 주소 설계시 고려사항이 아닌 것은?

① 주소를 효율적으로 나타내야 한다.
② 주소 byte를 정확하게 해야 한다.
③ 사용자에게 편리하도록 해야 한다.
④ 주소 공간과 기억 공간을 독립시킬 수 있어야 한다.

핵심이론

주소 설계 시 고려사항

- 주소를 효율적으로 나타내야 한다.
- 사용자에게 편리하도록 해야 한다.
- 주소 공간과 기억 공간을 독립시킬 수 있어야 한다.

유사문제

1회 ▶ 05-2

1. 주소 설계시 고려할 점으로 옳지 않은 것은?

① 주소 공간과 기억 공간을 항상 종속시켜야 한다.
② 주소를 효율적으로 나타낼 수 있어야 한다.
③ 프로그램이나 데이터가 그 컴퓨터 기억장치 내의 어느 곳에 기억되어 있더라도 수행이 가능해야 한다.
④ 주소 공간과 기억 공간을 독립시킬 수 있어야 한다.

2회 ▶ 14-1, 05-4

2. 주소 설계 시 고려해야 할 점이 아닌 것은?

① 주소를 효율적으로 나타낼 수 있어야 한다.
② 주소 공간과 기억 공간을 독립시킬 수 있어야 한다.
③ 전반적으로 수행속도가 증가될 수 있도록 해야 한다.
④ 주소 공간과 기억 공간은 항상 일치해야 한다.

1회 ▶ 14-3

3. 컴퓨터 기억장치의 주소설계 시 고려사항으로 옳지 않은 것은?

① 주소를 효율적으로 나타내야 한다.
② 주소 표시는 16진법으로 표기해야 한다.
③ 사용자에게 편리하도록 해야 한다.
④ 주소 공간과 기억 공간을 독립시킬 수 있어야 한다.

1회 ▶ 산 10-4

4. 주소 설계에서 일반적으로 고려해야 할 사항으로 가장 거리가 먼 것은?

① 주소를 효율적으로 나타낼 수 있어야 한다.
② 가상기억 공간을 확보해야 한다.
③ 주소 공간과 기억 공간을 독립시켜야 한다.
④ 사용자에게 편리해야 한다.

[정답] 핵심문제 ② / 유사문제 1. ① 2. ④ 3. ② 4. ②

THEME
120 인스트럭션 기타 문제

1회 ▶ 산 06-2

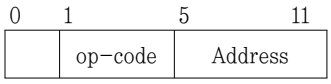
RRI(Register to Register Instruction)의 전체 가능한 수를 A, MRI(Memory Reference Instruction)의 전체 가능한 수를 B라고 할 때, 상호 관계로 올바른 것은?

① B, A = 0
② B 〉 A
③ B = A
④ B 〈 A

유사문제

1회 ▶ 산 08-2

1. 다음과 같은 명령 형식을 사용하는 컴퓨터에서 가능한 MRI(Memory Reference Instruction)의 개수는?



① 4
② 8
③ 16
④ 32

1회 ▶ 산 08-4

2. 산술 마이크로 동작 R1←R2+R3을 실행하기 위해 필요한 레지스터의 개수는?

① 1
② 2
③ 3
④ 4

1회 ▶ 03-1

3. 3-cycle 인스트럭션에 속할 수 없는 것은?

① ADD
② JUMP
③ LOAD
④ STORE

1회 ▶ 산 13-2

4. 다음 중 범용 레지스터를 사용하여 기억할 수 없는 것은?

① 연산할 데이터
② 연산된 결과
③ 실행될 명령어
④ 주기억장치에서 보내온 데이터

[정답] 핵심문제 ② / 유사문제 1. ③ 2. ③ 3. ② 4. ③