



2회 ▶ 산 02-1, 99-4

인스트럭션의 연산자 부분이 나타낼 수 있는 것으로 옳지 않은 것은?

- ① 인스트럭션의 순서
- ② 인스트럭션의 형식
- ③ 자료의 종류
- ④ 연산자

핵심이론

명령어의 구성

연산자부	자료부
------	-----

연산자부 (Operation Code)	<ul style="list-style-type: none"> 수행해야 할 명령이 들어 있다. 명령어 형식, 동작 코드, 데이터 종류 등을 표시한다. OP Code부라고도 한다.
자료부 (Operand)	<ul style="list-style-type: none"> 실제 데이터에 대한 정보가 들어 있다. 명령어 순서, 데이터, 데이터 주소, 데이터 주소를 구하는 데 필요한 정보 등을 표시한다. 번지부 또는 어드레스 필드(Address Field)라고도 한다.

유사문제

1회 ▶ 산 07-4

1. 명령 코드의 비트는 주소 필드(Field)를 가지고 있다. 이 주소 필드의 기능은?

- ① 누산기를 지정한다.
- ② 오퍼랜드를 선택할 수 있다.
- ③ 레지스터를 지정할 수 있다.
- ④ 수행할 동작을 명시할 수 있다.

1회 ▶ 산 12-2

2. 일반적인 컴퓨터의 명령어 형식에 포함되지 않은 필드(field)는?

- ① 연산코드 필드
- ② 주소 모드 필드
- ③ 주소 필드
- ④ 분기 필드

1회 ▶ 산 13-1

3. 명령 코드의 비트는 필드라고 불리는 몇 개의 그룹으로 나누어진다. 그 중 모드 필드(mode field)에 대한 설명으로 옳은 것은?

- ① 오퍼랜드나 유효번지가 결정되는 방법을 나타낸다.
- ② 메모리나 레지스터를 지정하는 방법을 나타낸다.
- ③ 수행하여야 할 동작을 나타낸다.
- ④ 명령을 수행하도록 제어 함수를 제공하는 방법을 나타낸다.

1회 ▶ 12-3

4. 명령어를 구성하는 명령어 내 비트들이 할당에 영향을 주는 요소가 아닌 것은?

- ① 버스 개수
- ② 주소지정 방식의 개수
- ③ 주소 영역
- ④ 연산 코드



1회 ▶ 05-4

명령어의 길이가 16bit이다. 이중 OP code가 5bit, operand가 8bit를 차지한다면 이 명령어가 가질 수 있는 연산자 종류는 최대 몇 개인가?

- ① 8개 ② 16개
③ 32개 ④ 256개

핵심이론

명령어의 개수

- 연산자부의 비트 수가 n Bit이면 명령어 개수는 최대 2^n 개이다.

예) OP code가 5비트이면 명령어 개수는 최대 $32(=2^5)$ 개이다.

유사문제

4회 ▶ 08-1, 07-1, 06-1, 02-1

1. OP-Code가 4비트면 연산자의 종류는 몇 개가 생성될 수 있는가?

- ① $2^4 - 1$ ② 2^4 ③ 2^3 ④ $2^3 - 1$

1회 ▶ 산 10-1

2. 16비트를 갖는 명령어 중 OP code가 5비트, operand가 8비트를 차지한다면 명령어의 최대 종류는?

- ① 5 ② 8 ③ 32 ④ 256

3회 ▶ 산 08-1, 05-1추, 02-3

3. OP 코드가 5비트, Operand가 11비트인 명령어가 갖는 매크로 연산의 종류는 몇 가지인가?

- ① 5가지 ② 32가지 ③ 128가지 ④ 2048가지

1회 ▶ 산 11-1

4. 명령어 형식이 다음과 같을 때 실행할 수 있는 명령의 수는?

OP 코드	간접	직접	주소
15	11 10	9 8	7 6
			0

- ① 16개 ② 32개 ③ 48개 ④ 64개

1회 ▶ 09-4

5. 1MByte의 기억장소를 가진 어떤 컴퓨터의 명령어 구성이 다음과 같을 때 이 명령어가 가질 수 있는 최대 Operation 수는?

Operation Code	Mode Bit	Register Selection Bit	Address Bit
5bit	1bit	2bit	20bit
28bit			

- ① 32개 ② 64개 ③ 128개 ④ 256개

1회 ▶ 07-2

6. 명령어의 길이가 16Bit이다. 이 중 OP Code가 6Bit, Operand가 10Bit를 차지한다면 이 명령어가 가질 수 있는 연산자의 종류를 최대 몇 개인가?

- ① 16개 ② 32개
③ 64개 ④ 256개

1회 ▶ 04-2

7. 명령어가 연산자(op code) 6비트, 주소 필드 16비트로 구성되어 있다. 이 명령어를 쓰는 컴퓨터는 최대 몇 가지 동작이 가능한가?

- ① 6 ② 16
③ 32 ④ 64

1회 ▶ 03-2

8. 명령어의 연산자 코드가 8비트, 오퍼랜드(operand)가 10비트 일 때 이 명령어로 몇 가지 연산을 수행하게 할 수 있는가?

- ① 8 ② 18
③ 256 ④ 1024

1회 ▶ 07-4

9. 16-Bit 컴퓨터 시스템에서 다음과 같은 2가지의 명령어 형식을 사용할 때 최대 연산자의 수는?



- ① 64 ② 72
③ 86 ④ 144

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



2회 ▶ 12-2, 10-1

인스트럭션 세트의 효율성을 높이기 위하여 고려할 사항이 아닌 것은?

- ① 기억 공간
- ② 레지스터의 종류
- ③ 사용 빈도
- ④ 주소지정 방식

핵심이론

명령어 설계 시 고려사항

- 연산자의 종류
- 데이터 구조
- 명령어 형식
- 주소지정 방식

명령어 세트의 효율성을 높이기 위한 고려사항

- 기억 공간
- 사용 빈도
- 주소지정 방식
- 주기억장치 बैं드폭 이용

유사문제

1회 ▶ 08-4

1. 인스트럭션의 설계 과정에서 고려해야 할 사항이 아닌 것은?

- ① interrupt 종류
- ② 연산자의 수와 종류
- ③ 데이터 구조
- ④ 주소지정 방식

2회 ▶ 05-1추, 05-1

2. 인스트럭션 세트의 효율성을 높이기 위하여 고려할 사항이 아닌 것은?

- ① 기억 공간
- ② 사용 빈도
- ③ 레지스터의 종류
- ④ 주기억장치 बैं드폭 이용



1회 ▶ 02-1

폰노이만(Von Neumann)형 컴퓨터의 연산자 기능으로서 적합하지 않은 것은?

- ① 병렬 처리 기능 ② 함수 연산 기능
③ 입출력 기능 ④ 전달 기능

핵심이론

연산자의 기능

- 함수 연산 기능
- 전달 기능
- 제어 기능
- 입·출력 기능

유사문제

1회 ▶ 05-4

1. 폰 노이만(Von Neumann)형 컴퓨터의 연산자 기능으로 옳지 않은 것은?

- ① 전달 기능 ② 제어 기능
③ 추적 기능 ④ 입출력 기능

4회 ▶ 10-4, 05-1, 02-2, 00-3

2. Von Neumann형 컴퓨터의 연산자들이 가져야 하는 기능 중 가장 거리가 먼 것은?

- ① 증폭 기능 ② 함수 연산(functional operation) 기능
③ 전달(transfer) 기능 ④ 제어(control) 기능

1회 ▶ 산 09-1

3. 폰 노이만(Von Neumann)형 컴퓨터 인스트럭션의 기능에 포함되지 않는 것은?

- ① 전달 기능 ② 제어 기능
③ 보존 기능 ④ 함수 연산 기능

2회 ▶ 04-2, 01-2

4. 컴퓨터에서 사용하는 명령어의 기능이 아닌 것은?

- ① 전달 기능 ② 제어 기능
③ 연산 기능 ④ 번역 기능

6회 ▶ 06-4, 04-4, 01-4, 산 14-3, 11-3, 00-4

5. 명령어에서 실행할 동작 부분을 나타내는 연산자(op code)의 기능과 관련 없는 것은?

- ① 함수 연산 기능 ② 입·출력 기능
③ 제어 기능 ④ 주소지정 기능

2회 ▶ 13-3, 03-1

6. 다음 중 OP-code의 기능이 아닌 것은?

- ① 주소 지정 ② 함수 연산
③ 전달 ④ 제어

1회 ▶ 산 99-2

7. 연산자의 기능이 아닌 것은?

- ① 함수 연산 기능 ② 번지 기능
③ 전달 기능 ④ 제어 기능

1회 ▶ 05-1추

8. 연산자 코드(operation code)의 기능이 아닌 것은?

- ① 입·출력 명령 수행 ② 제어 명령 수행
③ 유효 주소지정 기능 ④ 산술 연산 명령 수행

3회 ▶ 산 06-2, 03-2, 01-3

9. 컴퓨터의 연산자 기능이 아닌 것은?

- ① 기억 기능 ② 제어 기능
③ 전달 기능 ④ 함수 연산 기능

2회 ▶ 산 04-4, 01-2

10. 컴퓨터에서 사용되는 명령어들을 기능별로 분류할 때 분류 기준에 포함되지 않는 것은?

- ① 함수 연산 기능 ② 주소계산 기능
③ 전달 기능 ④ 입출력 기능



2회 ▶ 산 11-3, 05-2

다음 중 논리 연산만으로 짝지어진 것은?

- ① MOVE, AND, COMPLEMENT
- ② ROTATE, ADD, SHIFT
- ③ MOVE, EX-OR, SUBTRACT
- ④ MULTIPLY, AND, DIVIDE

핵심이론

함수 연산 기능

- 산술 연산
 - 연산의 대상 및 결과를 수치 데이터로 간주하고 행하는 수치적인 연산이다.
 - 연산자 종류 : ADD, SUB, MUL, DIV, 산술 Shift 등
- 논리 연산
 - 연산의 대상 및 결과가 0 또는 1 중 하나의 논리적인 값을 취하는 비수치적인 연산이다.
 - 연산자 종류 : AND, OR, XOR, Complement(NOT), 논리 Shift, Rotate(ROR, ROL), CPA, CLC 등

유사문제

1회 ▶ 01-3

1. 논리 연산에 들어가지 않는 것은?

- ① NOT
- ② Complement
- ③ OR
- ④ Load

3회 ▶ 산 05-1추, 01-1, 99-4

2. 레지스터에 있는 내용을 왼쪽으로 2비트 시프트 시키는 기능과 관계있는 것은?

- ① 제어 기능
- ② 연산 기능
- ③ 전송 기능
- ④ 레지스터 기능

2회 ▶ 03-2, 산 10-1

3. 컴퓨터에서 사용하는 명령어를 기능별로 분류할 때 동일한 분류에 포함되지 않는 것은?

- ① JMP(Jump 명령)
- ② ADD(Addition 명령)
- ③ ROL(Rotate Left 명령)
- ④ CLC(Clear Carry 명령)

1회 ▶ 산 99-2

4. 논리연산 기능으로만 나열된 것은?

- ① MOVE, AND, COMPLEMENT
- ② ROTATE, ADD, SHIFT
- ③ MOVE, EXCLUSIVE OR, SUBTRACT
- ④ MULTIPLY, AND, DIVIDE



1회 ▶ 산 13-1

다음 설명 중 틀린 것은?

- ① 중앙처리장치에서 연산한 결과 등을 일시적으로 저장해두는 레지스터를 누산기라 한다.
- ② 입출력 장치는 주변장치에 해당된다.
- ③ 레지스터에서 기억장치로 정보를 옮기는 것을 로드(load)라 한다.
- ④ 기억장치 내의 데이터를 다른 기억장치로 옮기는 것을 전송이라 한다.

핵심 이론

전달 기능

- CPU(레지스터)와 메모리 간에 자료를 전달하는 기능이다.
- Load : 메모리의 내용을 레지스터에 전달하는 명령이다.
- Store : 레지스터의 내용을 메모리에 전달하는 명령이다.
- Move : 레지스터 간에 자료를 전달하는 명령이다.
- Push : 스택에 자료를 저장하는 명령이다.
- Pop : 스택에서 자료를 인출하는 명령이다.

유사문제

5회 ▶ 산 14-3, 12-3, 12-2, 04-1, 99-1

1. 마이크로오퍼레이션에서 중앙처리장치의 정보를 기억장치에 기억시키는 동작은 무엇인가?

- ① LOAD
 - ② STORE
 - ③ TRANCE
 - ④ BRANCH

3회 ▶ 산 06-1, 05-4, 04-2

2. 레지스터의 내용을 메모리에 전달하는 기능을 무엇이라 하는가?

- ① Fetch
- ② Store
- ③ Load
- ④ Transfer

5회 ▶ 산 12-1, 10-2, 06-4, 05-1추, 02-3

3. 메모리의 내용을 레지스터에 전달하는 기능은?

- (① load ② fetch
③ transfer ④ store

2호 ▶ 10-1, 05-2

4. 연산자 기능에 대한 명령어를 나타낸 것 중 옳지 않은 것은?

- ① 함수 연산 기능 - ROL, ROR
- ② 전달 기능 - CPA, CLC
- ③ 제어 기능 - JMP, SMA
- ④ 입·출력 기능 - INP, OUT



유수의 조어

- CPA : (Acc) \leftarrow Acc' \rightarrow 누산기값을 1의 보수로 바꾸는 명령입니다.
- CLA : (Acc) \leftarrow 0 \rightarrow 누산기 레지스터의 값을 0으로 초기화하는 명령입니다.
- SMA : (Acc) \langle 0 then PC \leftarrow PC+2
 \rightarrow 누산기 값이 음수가 나오면 PC 레지스터의 값을 2 증가시켜서 명령의 순서를 바꾸는 제어명령에 해당합니다.

1호 ▶ 13-3

5. 통상적인 사용자 프로그램을 처리함에 있어서 중앙처리장치(CPU)가 가장 많이 실행하는 인스트럭션 종류는?

- ① 주기억장치와의 자료 전달(load, store)
- ② 수치적 및 논리적 연산(arithmetic, logical)
- ③ 입출력(input, output)
- ④ 조건 및 무조건 분기(branch)

1회 ▶ 산 09-1

6. 피연산자의 기억 장소에 따른 인스트럭션 분류 중 load 또는 store 인스트럭션의 사용 빈도가 매우 낮은 것은?

- ① 메모리-메모리 인스트럭션 형식
- ② 레지스터-레지스터 인스트럭션 형식
- ③ 레지스터-메모리 인스트럭션 형식
- ④ 스택 인스트럭션 형식



2회 ▶ 산 01-1, 00-2

다음의 어셈블리어로 나타낸 기본적인 명령(Instruction)중 제어 기능을 가진 명령만으로 짝지어진 것은?

- ① JMP X, ROL
- ② LAD X, SZC
- ③ SMA, JMP X
- ④ JMP X, LAD X

핵심이론

제어 기능

- 명령의 실행 순서를 변경할 때 사용하는 기능이다.
- 무조건 분기 명령 : GOTO, JMP(Jump) 등
- 조건 분기 명령 : IF, SPA, SNA(SMA), SZA 등
- Call : 부프로그램(서브루틴)을 호출하는 명령이다.
- Return : 부프로그램에서 주프로그램으로 복귀하는 명령이다.

유사문제

2회 ▶ 산 03-4, 01-2

1. 다음과 같은 명령어의 기능은?

JMP X

- ① 제어 기능
- ② 함수 연산 기능
- ③ 전달 기능
- ④ 입·출력 기능

1회 ▶ 산 13-2

2. 다음 명령(Instruction) 중에서 $PC \leftarrow X$ 와 같은 의미를 뜻하는 것은?

- ① JMP X
- ② ADD X
- ③ MOV X
- ④ STA X

3회 ▶ 산 12-1, 06-2, 99-1

3. 서브루틴과 연관되어 사용되는 명령은?

- ① Shift
- ② Call과 Return
- ③ Skip과 Jump
- ④ Increment와 Decrement

1회 ▶ 산 06-1

4. 다음 명령 중에서 번지 필드(Address Field)가 필요 없는 명령은?

- ① 데이터 전송 명령
- ② 산술 명령
- ③ 스킵(Skip) 명령
- ④ 서브루틴 Call 명령

1회 ▶ 산 10-4

5. 다음 중 번지필드가 필요 없는 명령은?

- ① skip
- ② branch
- ③ jump
- ④ call

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



1회 ▶ 산 11-1

다음 표에서 함수 연산 기능의 명령어를 수행하는 컴퓨터 구조와 피연산자의 기억장소가 올바르게 연결된 것은?

	컴퓨터의 구조	피연산자의 위치
ㄱ	AC(누산기) 구조	AC와 메모리
ㄴ	AC(누산기) 구조	메모리와 레지스터
ㄷ	범용 레지스터 구조	메모리와 레지스터
ㄹ	범용 레지스터 구조	메모리와 스택
ㅁ	스택 구조	스택

- ① ㄱ, ㄴ, ㄷ
- ② ㄱ, ㄷ, ㅁ
- ③ ㄴ, ㄷ, ㄹ
- ④ ㄷ, ㄹ, ㅁ

핵심이론

입 · 출력 기능

- CPU와 입 · 출력장치, 또는 메모리와 입 · 출력장치 간에 자료를 전달하는 기능이다.
- 명령어 종류 : INP(Input), OUT(Output)

유사문제

1회 ▶ 산 13-2

1. 함수 연산 기능 인스트럭션의 수행에 필요한 피연산자를 기억시킬 레지스터의 종류에 따라 컴퓨터 구조를 분류할 때, 이에 속하지 않은 것은?

- ① 스택 컴퓨터 구조
- ② AC 컴퓨터 구조
- ③ 리스트 컴퓨터 구조
- ④ 범용 레지스터 컴퓨터 구조



3회 ▶ 산 14-1, 05-2, 99-2

연산의 종류를 unary 연산과 binary 연산으로 구별할 때 다음 중 binary 연산을 하는 연산자가 아닌 것은?

- ① Complement
- ② OR
- ③ AND
- ④ Exclusive OR

핵심이론

단항(Unary) 연산

- 피연산자가 1개만 필요한 연산이다.
- 연산자 종류 : Complement(NOT), Shift, Rotate, Move 등

이항(Binary) 연산

- 피연산자가 2개 필요한 연산이다.
- 연산자 종류 : 사칙연산(ADD, SUB, MUL, DIV), AND, OR, XOR 등

유사문제

3회 ▶ 06-1, 산 08-2, 04-4

1. 다음 중 단항(Unary) 연산이 아닌 것은?

- ① Complement ② Rotate
- ③ AND ④ Shift

3회 ▶ 11-3, 06-2, 04-2

2. 컴퓨터 연산에서 단항(unary) 연산에 해당되지 않는 것은?

- ① Shift ② Complement
- ③ Rotate ④ OR

2회 ▶ 산 05-1, 99-4

3. 단항(Unary) 연산을 행하는 것은?

- ① SHIFT ② AND
- ③ OR ④ 4칙 연산

3회 ▶ 산 14-3, 07-4, 04-1

4. 다음 중 기능이 다른 연산자는?

- ① COMPLEMENT ② OR
- ③ AND ④ EX-OR

1회 ▶ 산 06-4

5. 이항 연산을 하는 연산자가 아닌 것은?

- ① Complement ② AND
- ③ OR ④ XOR

1회 ▶ 산 00-2

6. Binary 연산을 표시하는 것은?

- ① Complement ② Shift
- ③ AND ④ Rotate

1회 ▶ 산 05-4

7. 다음 중 바이너리(binary) 연산자는?

- ① ROTATE ② COMPLEMENT
- ③ OR ④ SHIFT

1회 ▶ 산 12-1

8. 다음 중 이항(binary) 연산자는 어떤 것인가?

- ① complement ② shift
- ③ AND ④ rotate



2회 ▶ 산 14-1, 07-2

다음 3가지 연산자(operator)가 혼합되어 나오는 식에서 시행(연산) 순서는? (단, 가장 왼쪽에 기술된 것이 가장 우선순위가 높다.)

- a) 관계 연산자(Relative operator)
- b) 논리 연산자(Logical operator)
- c) 산술 연산자(Arithmetic operator)

- ① $a \rightarrow b \rightarrow c$
- ② $b \rightarrow a \rightarrow c$
- ③ $c \rightarrow a \rightarrow b$
- ④ $a \rightarrow c \rightarrow b$

유사문제

1회 ▶ 산 05-1

1. 일반적으로 수식에서 3가지의 연산자(operator)가 혼합되어 나오는데 이들 연산자의 시행 순서가 옳은 것은?

- ① 산술 연산자 → 관계 연산자 → 논리 연산자
- ② 산술 연산자 → 논리 연산자 → 관계 연산자
- ③ 관계 연산자 → 산술 연산자 → 논리 연산자
- ④ 논리 연산자 → 관계 연산자 → 산술 연산자

1회 ▶ 산 05-1

2. 비수치적 연산을 설명한 것으로 옳은 것은?

- ① AND 연산은 레지스터 내에 특정 비트를 삽입하기 위한 연산으로 주로 사용된다.
- ② X-OR 연산은 레지스터 내의 모든 비트를 0으로 클리어(clear) 할 때 사용하는 연산이다.
- ③ COMPLEMENT 연산은 자료의 특정 비트 혹은 문자를 삭제하고자 할 때 사용된다.
- ④ MOVE 연산은 특정 레지스터의 내용을 다른 레지스터로 옮기고자 하는 경우 사용한다.

1회 ▶ 산 11-1

3. 컴퓨터 연산에 대한 설명 중 옳지 않은 것은?

- ① 한 번에 3개 이상의 데이터를 단일 연산기로 동시에 처리할 수 있다.
- ② 연산에 사용되는 데이터의 수가 한 개뿐인 것을 단항(unary)연산이라 한다.
- ③ 중앙처리장치(CPU)에서 연산에 사용될 데이터를 기억시켜 두는 장소를 레지스터라 한다.
- ④ 이동(move)과 회전(rotate)은 비수치적 연산에 속한다.

2회 ▶ 07-1, 02-2

4. 데이터 처리 명령어에 해당되지 않는 것은?

- ① 전송 명령어
- ② 로테이트 명령어
- ③ 논리 명령어
- ④ 산술 명령어

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