컴퓨터구조

2018학년도 2학기 출석수업대체시험

핵심체크 및 출제예상문제

범위 : 교재 1 ~ 4장

제1장 컴퓨터 구조의 개요

1. 개요

우리는 '컴퓨터'라 불리는 기계를 가정에서 혹은 사무실에서 흔히 볼 수 있다. 오늘날의 컴퓨터는 데이터 처리의 중심도구로서 사용되고 있으며, 컴퓨터를 보다 정확하게 표현한다면 전자식 데이터 처리 시스템이라고 말할 수 있다.

2. 컴퓨터 시스템의 발전과정

(1) 제1세대 컴퓨터

① 진공관을 이용한 전자식 컴퓨터.
② 컴퓨터 제어가 단일 중앙처리장치에 집중됨
③ 어셈블리어(assembly language)의 사용

(2) 제2세대 컴퓨터

① 진공관을 대체한 트랜지스터 사용.
② 고급 프로그래밍 언어인 ALGOL, FORTRAN, COBOL을 사용.
③ 큰 용량의 기억장치를 가짐.
④ 입출력처리장치와 같은 특별한 처리장치의 도입으로 중앙처리장치의 시간 낭비를 줄임.
⑤ 컴퓨터 제조업자들이 컴파일러, 소프트웨어 라이브러리 등을 제공.
⑥ 하드웨어 설계시 모듈화 개념이 도입됨.
⑦ 자기디스크의 개발로 보조기억장치에 대한 직접 접근이 가능해짐.

(3) 제3세대 컴퓨터

트랜지스터를 대체한 집적회로의 등장.

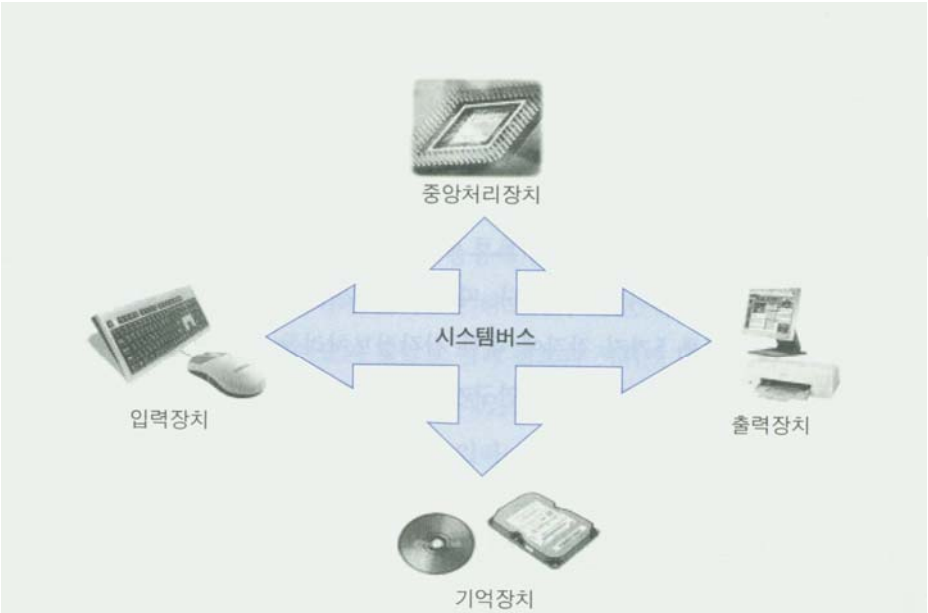
(4) 제4세대 컴퓨터

① 여러 개의 집적회로를 하나의 패키지에 집적한 대규모집적회로의 사용.
② 이후 반도체 기술의 발달로 초대규모집적회로가 개발되어 현재 컴퓨터에 사용.
③ 컴퓨터의 주요 구성요소들을 하나의 반도체 칩에 모두 집적시킨 마이크로 프로세서가 개발.
④ 개인용 컴퓨터가 널리 보급됨.
⑤ 컴퓨터를 사용한 업무의 효율성을 위한 자동화가 이루어지기 시작함.

(5) 1.2.3 차세대 컴퓨터

① 대규모 병렬처리 컴퓨터: 중복된 하드웨어를 효율적으로 활용할 수 있는 병렬처리기법을 도입.
② 광 컴퓨터: 하드웨어 기술이 지금까지의 속도로 발전한다면 불과 수년 내에 전자의 속도 도달.
③ 신경망 컴퓨터: 폰 노이만의 프로그램 내장형 컴퓨터 구조의 문제점을 해결하는 방법으로서, 인간의 신경계통 5가지 감각에 대한 감각정보처리를 모방한 컴퓨터 구조

3. 컴퓨터 시스템의 전체적 구성



〈그림 1-1〉 컴퓨터의 구성도

(1) 입력장치

사용자가 입력하는 데이터와 명령어를 받아서 컴퓨터가 알 수 있는 형태로 변환하여 기억 장치나 중아처리장치로 전달하는 장치(예: 키보드, 스캐너, 마이크, 마우스 등)

(2) 출력장치

입력된 데이터와 명령어를 처리하고 생성된 결과를 사람이 알아볼 수 있는 형태로 변환해 주는 장치(예: CRT화면, 프린터 등)

(3) 기억장치

① 주기억장치: 중앙처리장치가 처리할 데이터와 명령어들을 저장.(DRAM으로 구성)

② 캐시기억장치: 주기억장치와 중앙처리장치 사이에 위치한 기억장치로서 데이터가 중앙처리 장치에 보다 빨리 전달되게 한다. (SRAM의로 구성)

③ 보조기억장치: 대량의 데이터를 저장하기 위한 기억장치로서 캐시나 주기억장치와 달리 전원이 꺼져도 후에 사용할 수 있도록 해 준다.

(4) 중앙처리장치

컴퓨터의 두뇌에 해당하는 부분으로서 실제 연산을 수행하며 컴퓨터의 각 요소들을 순서에 맞추어 작동시키는 제어작용을 수행하는 장치

(5) 시스템버스

입력장치와 출력장치, 기억장치, 중앙처리장치 사이의 통신을 가능하게 해주는 통신 선로.

4. 컴퓨터 시스템의 분류

(1) 처리성능과 규모에 따른 분류

① 마이크로 컴퓨터

② 미니컴퓨터

③ 메인프레임컴퓨터

④ 슈퍼컴퓨터

(2) 구조에 따른 분류

① 파이프라인 슈퍼컴퓨터

② 대규모 병렬처리 컴퓨터

<1장 출제예상문제>

1. 컴퓨터 시스템의 주요 구성요소가 아닌 것은?

① 입력장치 ② 시스템버스

③ 중앙처리장치 ④ 링 장치

[해설] 시스템의 주요 구성요소

- 입력장치: 사용자가 입력하는 데이터와 명령어를 받아서 컴퓨터가 알 수 있는 형태로 변환하여 기억장치나 중앙처리장치로 전달하는 장치.(예. 키보드, 스캐너, 마이크, 마우스 등)
- 출력장치: 입력된 데이터와 명령어를 처리하고 생성된 결과를 사람이 알아볼 수 있는 형태로 변환해주는 장치.(예. CRT화면, 프린터 등)
- 기억장치: 주기억장치, 캐쉬, 보조기억장치
- 중앙처리장치: 컴퓨터의 두뇌에 해당하는 부분으로서 실제 연산을 수행하며 컴퓨터의 각 요소들을 순서에 맞추어 작동시키는 제어작용을 수행하는 장치.
- 시스템 버스: 입력장치와 출력장치, 기억장치, 중앙처리장치사이의 통신을 가능하게 해주는 통신 선로.

[정답] 4

2. 다음 중 컴퓨터 시스템의 구성요소와 그에 해당되는 인체의 구성요소의 연결이 가장 알맞게 된 것은?

① 감각기관-입력장치 ② 신경망-기억장치

③ 뇌세포-중앙처리장치 ④ 소화기관-출력장치

[정답] 1

3. 다음 중 인가전압에 따른 액정의 투과도 변화를 이용하여 각종 장치에서 발생하는 여러 가지 전기적인 정보를 시각정보로 변화시켜 전달하는 방법을 사용하는 출력장치는 무엇인가?

① CRT ② LCD

③ PDP ④ 프린터

[정답] 2

4. 다음 중 데이터 중앙처리장치에 신속하게 전달될 수 있도록 하기 위해 자주 사용되는 주기억장치 와 중앙처리장치 사이에 위치한 기억장치는 무엇인가?

① DRAM ② CD-ROM
③ 캐쉬 ④ 자기테이프

[해설] 캐쉬
주기억장치와 중앙처리장치 사이에 위치한 기억장치로서 데이터가 중앙처리장치에 보다 빨리 전달되게 한다. SRAM으로 구성

[정답] 3

5. 다음은 컴퓨터의 세대별 구성소자를 나타낸 것이다. 아닌 것은?

① 제1세대-진공관 ② 제2세대-트랜지스터
③ 제3세대-직접회로 ④ 제4세대-VLSI

[해설] 제 4세대 컴퓨터
- 여러 개의 집적회로를 하나의 패키지에 집적한 대규모 집적회로(LSI: Large Scale IC)의 사용
- 이후 반도체 기술의 발달로 초대규모 집적회로(VLSI: Very LSI)가 개발되어 현재 컴퓨터에 사용
- 컴퓨터의 주요 구성요소들을 하나의 반도체 칩에 모두 집적시킨 마이크로프로세서가 개발되어 널리 이용됨.
- 개인용 컴퓨터가 널리 보급됨
- 컴퓨터를 사용한 업무의 효율성을 위한 자동화가 이루어지기 시작함

④ 제4세대 - LSI

[정답] 4

6. 다음 중 미래형 컴퓨터의 형태는?

① 광 컴퓨터, 신경망 컴퓨터, UNIVAC
② 신경망 컴퓨터, TX-O, UNIVAC
③ 대규모 병렬처리 컴퓨터, 광 컴퓨터, PC-XT
④ 광 컴퓨터, 신경망 컴퓨터, 대규모 병렬처리 컴퓨터

[해설]
미래형 컴퓨터의 대표적인 형태로는 대규모 병렬처리 컴퓨터, 광 컴퓨터, 신경망 컴퓨터 등이 있음

[정답] 4

7. 다음 중 컴퓨터 시스템의 구성요소에서 시스템 버스에 대한 설명은?
① 데이터를 처리하여 사람이 알아볼 수 있는 형태로 변환한다.
② 중앙처리장치가 처리한 데이터를 기억한다.
③ 두 개 이상의 장치를 연결하는 통신선로이다.
④ 연산과 제어작용을 수행한다.

[해설]
버스란 두 개 이상의 장치들을 연결해주는 통신선로를 말함

① 출력장치, ② 주기억장치, ④는 중앙처리장치.

[정답] 3

제2장 컴퓨터 명령어

1 개요

* 본 장에서는 상업적인 범용 컴퓨터에서 볼 수 있는 가장 기본적인 명령어를 소개하고 설명함으로써 컴퓨터 명령어가 쉽게 이해될 수 있도록 한다. 또한 컴퓨터에서의 다양한 명령어 형식을 설명하고자 함

2 명령어의 구성

* 명령어는 필드라는 비트 그룹으로 이루어지며, 연산코드와 오퍼랜드필드로 나누어진다.

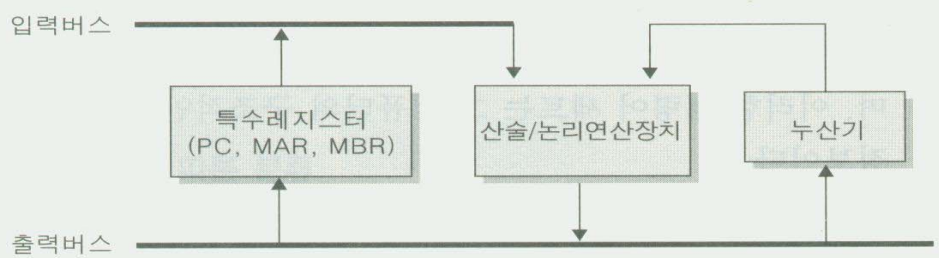
연산코드 (OP code)	오퍼랜드(operand)

3 명령어 형식

* 명령어는 컴퓨터의 내부구조에 따라 여러 가지 형식이 있다.

(1) 기억장소에 따른 명령어 형식

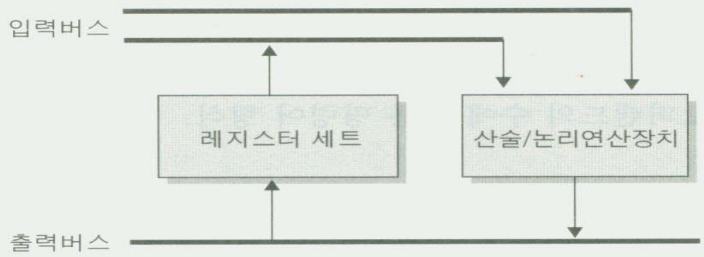
① 누산기를 이용한 명령어 형식: 누산기를 가진 명령어 구조는 함수연산기능의 명령어를 수행할 때 오퍼랜드들 중의 하나가 누산기에 기억되도록 하는 컴퓨터 구조에서 사용된다.



[누산기를 가진 컴퓨터 구조]

(예: ADD X ; AC <- AC + M[X])

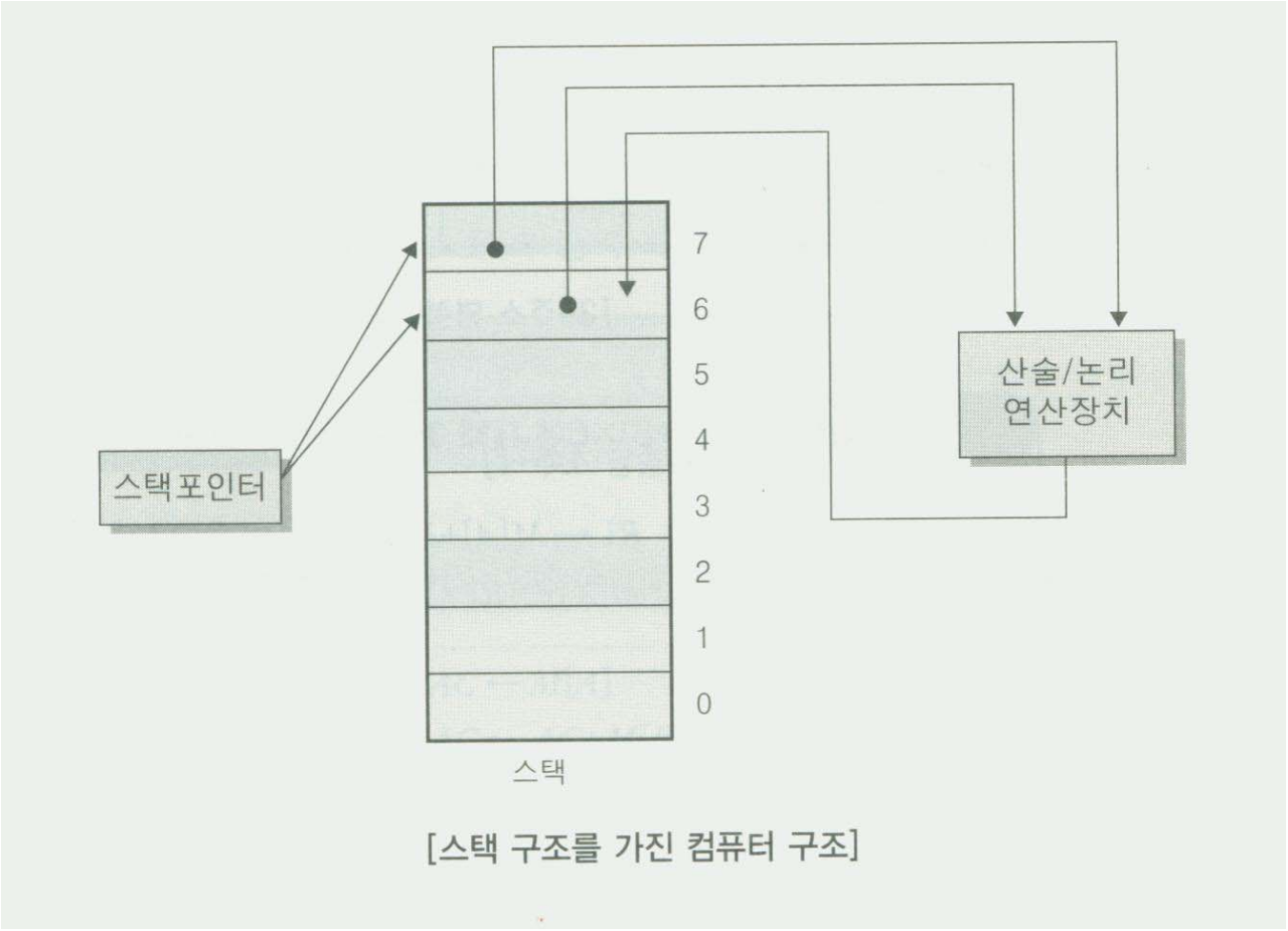
② 다중 레지스터를 이용한 명령어 형식: 다중 레지스터를 가진 컴퓨터 구조는 중앙처리장치 내에 여러 개의 레지스터를 가지고 있는 컴퓨터이다.



[다중 레지스터를 가진 컴퓨터 구조]

(예: ADD R1, R2, R3 ; R3 <- R1 + R2)

③ 스택 구조를 이용한 명령어 형식: 스택 구조 컴퓨터는 연산에 필요한 오퍼랜드들을 기억장치 스택에 기억시켜야 하고, 연산의 결과도 스택에 기억시키는 구조이다.



[스택 구조를 가진 컴퓨터 구조]

(예: ADD ; TOS <- TOS + TOS-1)

(2) 오퍼랜드의 수에 따른 명령어 형식

① 3-주소 명령: 오퍼랜드의 개수가 세 개인 명령어 형식이다.

② 2-주소 명령: 상업용 컴퓨터에서 가장 많이 사용되는 명령어 형식으로서 오퍼랜드 의 개수가 두 개인 명령어 형식이다.

③ 1-주소 명령: 오퍼랜드의 개수가 하나인 명령어 형식으로서, 기억장치로부터 오퍼 랜드를 가져오거나 연산결과를 저장하기 위한 임시적인 장소로 누산기 레지스터를 사용한다.

④ 0-주소 명령: 산술연산에 관련된 명령어들을 수행할 때, 기억장치 스택을 사용한다. 스택 구조를 이용한 0-주소 명령어 형식에서는 함수연산을 수행할 때 주소 필드를 사용하지 않는다.

4. 주소지정방식

- 주소지정방식이란 프로그램 수행시 오퍼랜드를 지정하는 방식으로서 오퍼랜드를 실제 참조하기 전에 명령어의 주소 필드를 변경하거나 해석하는 규칙을 지정하는 형식이다.
- 유효주소란 주소지정방식의 각 규칙에 의해 정해지는 오퍼랜드의 실제 주소를 말한다.

(1) 의미주소지정
명령어에서 주소 필드를 필요로 하지 않는 방식으로 연산코드 필드에 지정된 묵시적 의미 의 오퍼랜드를 지정한다. (예: ADD ; TOS <- TOS + TOS-1)

(2) 즉치주소지정
명령어 자체 내에 오퍼랜드를 지정하고 있는 방식이다. (예: LDI R1 ; R1 <- 100)
위 명령어는 레지스터 R1에 데이터 100을 초기화시키는 것으로서, 명령어 LDI자체 내에 100이라는 오퍼랜드를 포함하는 것이다.

(3) 직접주소지정과 간접주소지정
① 직접주소지정방식은 명령어의 주소 필드에 직접 오퍼랜드의 주소를 저장시키는 방식이다.
(예: LDA ADRS ; AC <- M[ADRS])
② 간접주소지정방식은 명령어의 주소 필드에 유효주소가 저장되어 있는 기억장치주소를 기억 시키는 방식으로서, 제어는 기억장치로부터 명령어를 가져온 후 주소 부분을 이용하여 다시 기억장치에 접근하여 유효주소를 읽어 낸다. (예: LDA[ADRS] ; AC <- <M[M[ADRS]])

(4) 레지스터와 레지스터 간접주소지정
① 레지스터 주소지정방식은 오퍼랜드가 레지스터에 저장되어 있는 방식이다. (예: LDA R1 ; AC <- R1)
② 레지스터 간접주소지정방식은 레지스터가 실제 오퍼랜드가 저장된 기억장치의 주소값을 갖고 있는 방식
(예: LDA (R1) ; AC <- M[R1])

(5) 상대주소지정
① 유효주소를 계산하기 위해 처리장치 내에 있는 특정 레지스터의 내용에 명령어 주소 필드 값을 더하는 방식이다.
② 유효주소 = 명령어 주소부분의 내용 + PC의 내용
(예: AC <- M[R1] ; AC <- M[ADRS + PC])

(6) 인덱스된 주소지정
① 인덱스 레지스터의 내용을 명령어 주소 부분에 더해서 유효 주소를 얻는다.
② 유효주소 = 명령어 주소 부분의 내용 + 인덱스 레지스터의 내용
(예: LDA ADRS(R1) ; AC <- M[ADRS + R1])

5. 명령어 종류
(1) 데이터 전송 명령어
① 한 장소에서 다른 장소로 단지 데이터를 전송하는 명령어
② 레지스터와 레지스터 사이, 레지스터와 기억장치 사이, 또는 기억장치와 기억장치 사이에 데이터를 이동하는 기능
③ 입출력 명령어가 포함

(2) 데이터 처리명령어
데이터에 대한 연산을 실행하고 컴퓨터에 계산능력 제공

① 산술명령어: 사칙연산에 대한 산술명령어
② 논리 및 비트 처리명령어
- 레지스터나 기억장치에 저장된 단어에 대해 2진 연산 수행
- 주로 2진 부호화 정보를 표현하는 비트 그룹이나 개별 비트를 처리하는데 사용
- 비트 값을 0으로 만들거나, 기억장치 레지스터에 저장된 오퍼랜드에 새로운 비트 값을 삽입하는 것 등이 가능
③ 쉬프트 명령어
- 쉬프트는 오퍼랜드의 비트를 왼쪽이나 오른쪽으로 이동시키는 동작
- 논리적 쉬프트와 산술적 쉬프트, 회전형 쉬프트 연산 등이 있음

(3) 프로그램 제어명령어
프로그램 수행의 흐름을 제어하거나 다른 프로그램 세그먼트(segment)로 분기할 수 있는 능력 제공

<2장 출제예상문제>

1. 다음 중 컴퓨터 명령어를 구성하는 필드가 아닌 것은?
① 수행할 연산을 나타내는 연산코드 필드
② 기억장치주소 혹은 처리장치 레지스터를 선택하기 위한 주소 필드
③ 오퍼랜드에 대한 해석방법을 지정하는 방식 필드
④ 인터럽트 요청을 위한 제어 필드

[해설]
하나의 컴퓨터 명령을 구성하는 필드들로는 수행할 연산을 나타내는 연산코드(operand code) 필드 및 기억장치주소나, 처리기 레지스터를 선택하기 위한 주소(address)필드, 주소필드를 해석하는 방법을 지정하는 방식(mode) 필드가 있음

[정답] 4

2. 다음의 프로그램과 관계가 깊은 명령어 형식은?

```
ADD    A,B,R1          ; R1 <- M[A] +M[B]
MUL    R1,C,X          ; M[X] <- R1 x M[C]
```

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

[해설] 3-주소 명령어(three-address instruction)
- 오퍼랜드의 개수가 세 개인 명령어 형식
- 장점: 산술식을 프로그램화하는데 있어서 그 프로그램의 길이가 짧아진다는 것
- 단점으로는 3-주소 명령어를 2진 코드화 했을 때 세 개의 오퍼랜드를 나타내기 위한 비트 수가 다른 주소 명령어 형식보다 많이 필요하다는 것

- 산술식 X=(Y+B)XC에 대해 3-주소 명령어를 이용한 프로그램
  ADD A, B, R1 ; R1 ← M[A] + M[B]
  MUL R1, C, X ; M[X] ← R1 + M[C]

[정답] 4

3. 다음은 1-주소 명령어 형식으로 작성된 프로그램이다. 빈곳에 들어갈 명령어는?

```
LOAD   A          ; AC <- M[A]
ADD    B          ; AC <- AC + M[B]
STORE  X          ; (                    )
LOAD   C          ; AC <- M[C]
MUL    X          ; AC <- AC x M[X]
STORE  X          ; M[X] <- AC
```

① M[X] <- AC ② AC <- M[X]
③ M[X] <- M[B] ④ M[B] <- M[X]

[해설]
위의 프로그램은 먼저 주소 A가 가리키는 메모리의 내용을 AC로 가져와서(LOADA; AC ← M[A]), 주소 B가 가리키는 메모리 내용과 합한 결과 값을 AC에 저장하고(ADDB; AC ← AC + M[B]), 주소 X가 가리키는 메모리에 AC의 내용을 저장한다(STORE X; M[X] ← AC). 따라서, X에는 A+B가 저장됨. 그리고, 주소 C가 가리키는 메모리의 내용을 AC로 가져와서(LOAD C; AC ← M[C]), 주소 X가 가리키는 메모리 내용과 곱한 결과값을 AC에 저장하고(MUL X; AC ← AC * M[C]), 주소 X가 가리키는 메모리에 AC의 값 ((A+B)*C)을 저장함(STORE X; M[X] ← AC)

[정답] 1

4. 3번 문제의 프로그램과 연관이 깊은 주소지정방식은 무엇인가?
① 의미주소지정 ② 즉치주소지정
③ 상대주소지정 ④ 간접주소지정

[해설]
ADD B와 같은 명령어는 AC의 값과 주소 B가 가리키는 메모리 내용과 합한 결과 값을 AC에 저장하라는 의미로, 명령어 ADD에 지정된 묵시적인 의미로 오퍼랜드 AC를 지정함. 이러한 방식을 의미주소지정방식이라함

[정답] 1

5. 다음 중 명령어의 오퍼랜드 필드에 유효주소가 저장되어 있는 기억장치주소를 기억시키는 주소지정방식은 무엇인가?
① 직접주소지정방식 ② 간접주소지정방식
③ 인덱스된 주소지정방식 ④ 레지스터 간접주소지정방식

[해설] 간접주소지정방식(indirect-addressing mode)
명령어의 주소필드에 유효주소가 저장되어있는 기억장치주소를 기억시키는 방식으로서, 제어는 기억장치로부터 명령어를 가져온 후 주소부분을 이용하여 다시 기억장치에 접근하여 유효주소를 읽어냄

[정답] 2

6. 다음 중 주소 지정방식과 유효주소에 대한 서술로 잘못된 것은?
① 즉치 주소지정방식에서는 유효주소가 없다.
② 레지스터 직접주소 지정방식에서는 유효주소가 없다.
③ 상대 주소지정방식에서는 유효주소가 PC 값에 따라 달라진다.
④ 유효주소는 해당 연산의 피연산자가 저장되어 있는 주기억장치의 주소이다.

[해설] 주소 지정 방식(addressing mode)
오퍼랜드를 실제로 참조하기 전에 명령어의 오퍼랜드를 변경하거나 해석하는 규칙을 지정하는 형식. 이러한 규칙의 적용에 의해 만들어진 오퍼랜드의 실제 주소를 유효주소(effective address)라 함

[정답] 1

7. 다음 중 데이터 전송 명령어가 아닌 것은 무엇인가?
① ST ② MOVE
③ DIV ④ LD

[해설] 대표적인 데이터 전송명령어

전송명령어	니모닉	기능
Load	LD	기억장치로부터 레지스터로의 전송
Store	ST	레지스터로부터 기억장치로의 전송
Move	MOVE	레지스터로부터 다른 레지스터로의 전송
Exchange	XCH	두 레지스터 간 또는 레지스터와 기억장치간의 데이터 교환
Push	PUSH	기억장치의 스택과 레지스터간의 데이터 전송
Pop	POP	
Input	IN	레지스터와 입출력장치 간의 데이터 전송
Output	OUT	

[정답] 3

8. [POP X]의 컴퓨터 명령에 대한 설명은?
① 입력 포트에서 레지스터로의 데이터 전송을 위한 명령어이다.
② 레지스터에서 출력 포트로의 데이터 전송을 위한 명령어이다.
③ 기억장치 스택에서 레지스터로의 데이터 전송을 위한 명령어이다.
④ 레지스터에서 기억장치 스택으로의 데이터 전송을 위한 명령어이다.

[해설]

③ POP 명령은 기억장치의 스택과 레지스터간의 데이터 전송에 관련함

[정답] 3

9. 다음 중 부호화되지 않은 수에 대한 분기 명령어에서 BE(Branch if Equal)를 실행시키기 위해서 조사하는 것은?

① N = 0 ② V = 0
③ C + Z = 1 ④ Z = 1

[해설] BE
상태 비트 Z = 1일 때 분기하는 명령어

[정답] 4

10. 다음 중 프로그램 인터럽트에 대한 설명이 아닌 것은?

① 프로그램 인터럽트는 내부 또는 외부 신호에 의해 시작된다.
② 내부 인터럽트는 사용자 모드와 시스템 모드의 전환 수단으로 사용된다.
③ 프로그램 인터럽트에서 해당 서비스 프로그램의 시작주소는 하드웨어적으로 결정된다.
④ 하드웨어 인터럽트는 입출력장치가 데이터 전송이 필요로 하는 경우, 한 작업에 할당된 시간이 종료한 경우, 전원 고장이 발생한 경우에 발생한다.

[해설]
② 소프트웨어 인터럽트는 사용자 모드와 시스템 모드의 전환 수단으로 사용됨

[정답] 2

11. 다음 중 기억장치로부터 오퍼랜드를 가져오거나 연산결과를 저장하기 위해 임시적 장소로 AC(accumulator)를 사용하는 것과 연관 깊은 명령어 형식은?

① 0-주소 명령 ② 1-주소 명령
③ 2-주소 명령 ④ 3-주소 명령

[해설] 1-주소 명령어(one-address instruction)
형식은 오퍼랜드의 개수가 하나인 명령어 형식으로 기억장치로부터 오퍼랜드를 가져오거나 연산결과를 저장하기 위한 임시적인 장소로 누산기 레지스터를 사용함. 1-주소 명령어에서 모든 연산은 누산기 레지스터와 기억장치에 저장된 오퍼랜드를 대상으로 수행되며, 이 경우 프로그램을 수행하기 위해 사용되는 명령어의 수는 더 증가한다.

[정답] 2

12. 데이터 전송 명령어는 무엇인가?

① NEG ② SHL

③ RET　　④ XCH

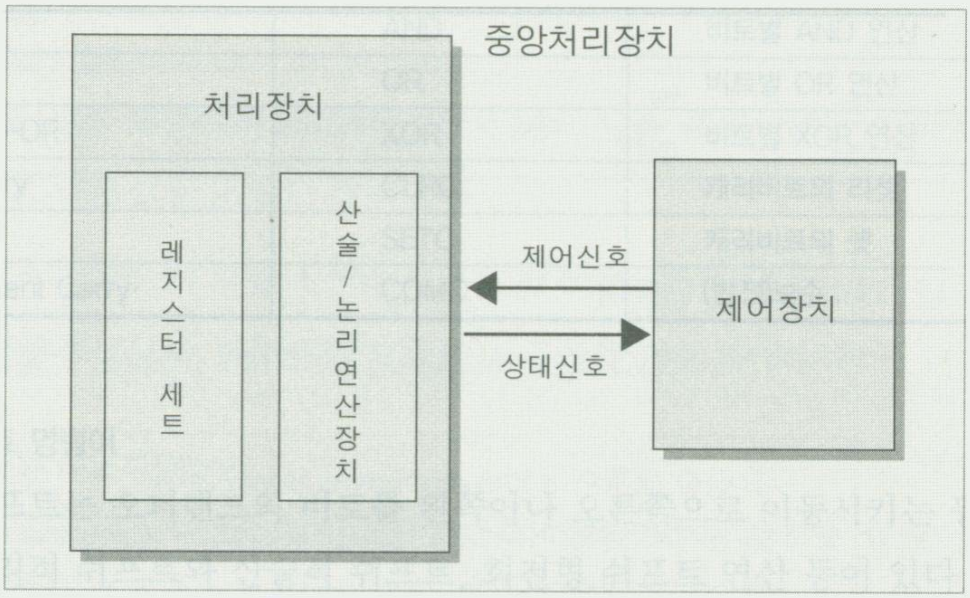
[해설] XCH(Exchange) 명령
두 레지스터간 또는 레지스터와 기억장치간의 정보를 서로 바꾸는 데이터 전송 명령임
① NEG(Negation)는 데이터 처리 명령(산술명령)
② SHL(Shift left)은 데이터 처리 명령(쉬프트명령)
③ RET(Return)는 프로그램 제어명령임

[정답] 4

제3장 처리장치

1. 개요

* 처리장치: 데이터를 처리하는 연산을 실행.
* 제어장치: 연산의 실행순서를 결정하는 역할.
* 중앙처리장치(CPU): 처리장치와 제어장치가 결합된 형태.



[중앙처리장치에서의 처리장치]

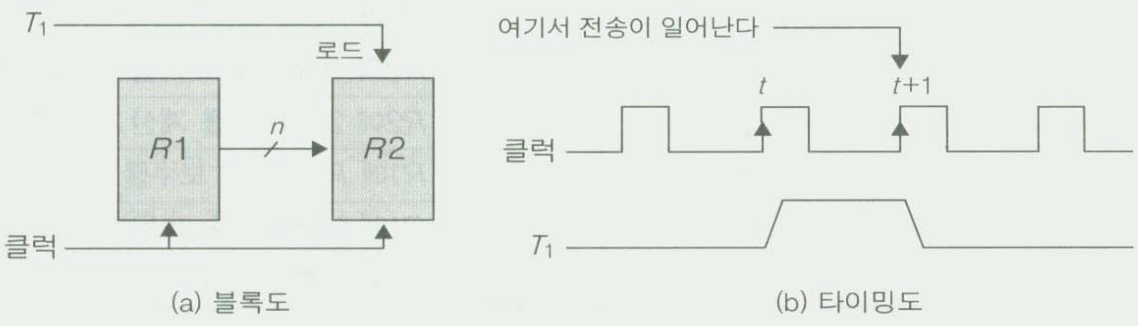
2. 마이크로연산

* 정의: 레지스터나 기억장치에 있는 데이터에 대해 이루어지는 기본적인 연산

(1) 레지스터 전송 마이크로연산

① 한 레지스터에서 다른 레지스터로의 데이터 전송은 연산자 '<-'로 표시 (예: R2 <- R1)
일 때 위 식은 레지스터 R1의 내용이 레지스터 R2로 전송됨을 나타낸다. 여기서 R1을 출발 레지스터라고 하고, R2를 도착레지스터라고 한다.

기호	의미	예
영문자(숫자 함께)	레지스터 표시	AR, R2, DR, IR
괄호	레지스터 일부분	R2(1), R2(7:0), AR(L)
대괄호	메모리에서의 어드레스	DR←M[AR]
화살표	자료 이동	R1←R2
쉼표	동시에 실행되는 두 개 이상의 마이크로연산 구분	R1←R2, R2←R1



[T_1=1인 상태에서 R1에서 R2로의 데이터 전송]

(2) 산술 마이크로연산

기본적인 산술연산으로는 덧셈, 뺄셈, 1 증가, 1 감소, 그리고 보수연산이 있다.

기호	의미
R0 ← R1 + R2	R1과 R2의 합을 R0에 저장
$R2 \leftarrow \overline{R2}$	R2의 보수(1의 보수)를 R2에 저장
$R2 \leftarrow \overline{R2} + 1$	R2에 2의 보수를 계산 후 저장
$R0 \leftarrow R1 + \overline{R2} + 1$	R1에 R2의 2의 보수를 더한 후 R0에 저장
R1 ← R1 + 1	R1에 1 더함(상승카운트)
R1 ← R1 − 1	R1에 1뺌(하강카운트)

(3) 논리 마이크로연산

① 기본적인 논리연산으로는 AND, OR, XOR, NOT 연산이 있음

② 레지스터에 저장되어 있는 비트의 데이터를 조작하는데 유용

기호	의미
$R0 \leftarrow \overline{R1}$	비트별 논리적 NOT(1의보수)
R0 ← R1∧R2	비트별 논리적 AND(비트 클리어)
R0 ← R1∨R2	비트별 논리적 OR(비트 세트)
R0 ← R1⊕R2	비트별 논리적 XOR(비트별 보수)

(4) 쉬프트 마이크로연산

① 데이터의 측면이동에 사용

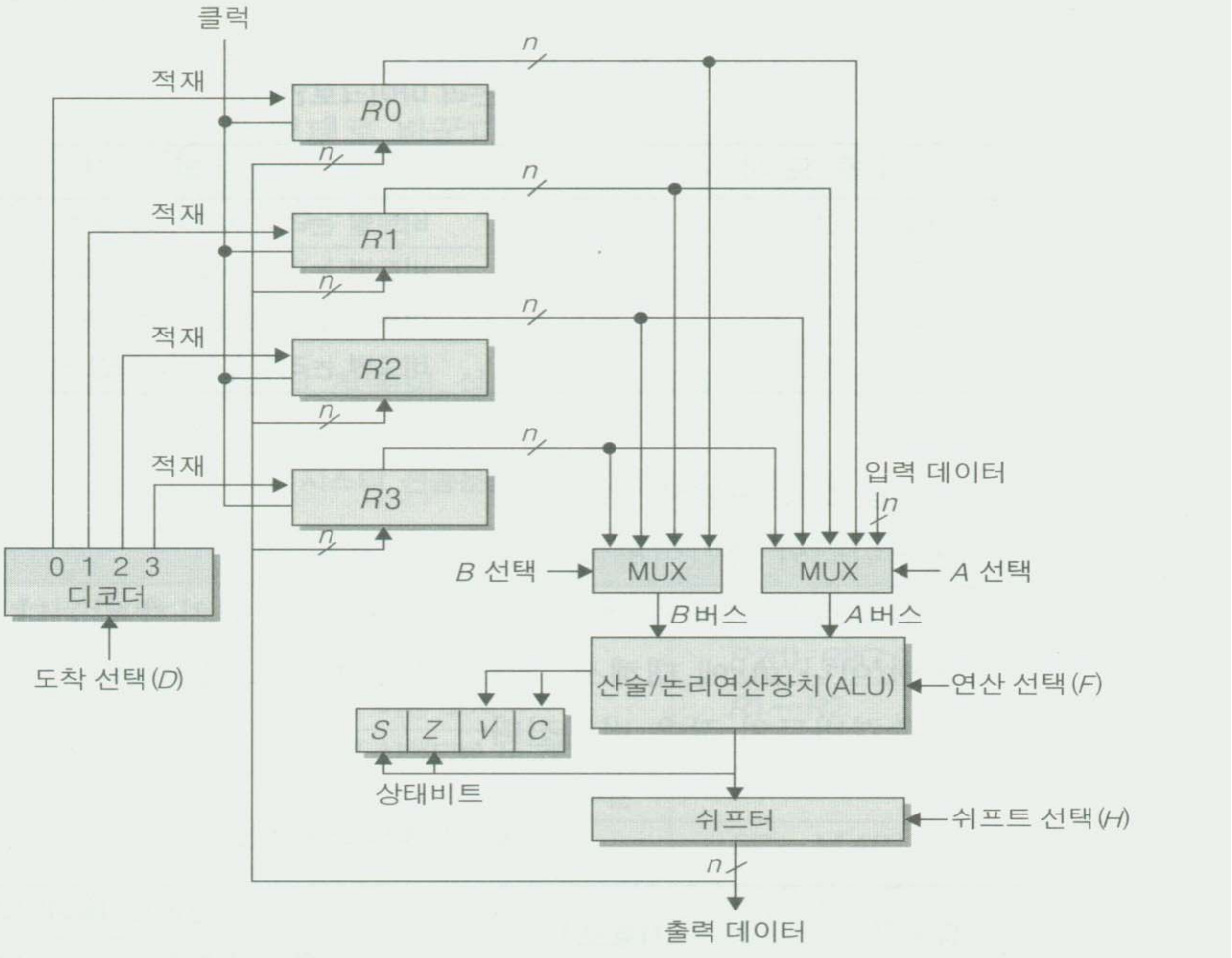
기호	의미	8비트 데이터의 경우	
		출발지R2	쉬프트 후: 목적지 R1
왼쪽 쉬프트	R1 ← sl R2	10011110	00111100
오른쪽 쉬프트	R1 ← sr R2	11100101	01110010

① 쉬프트 연산을 수행하더라도 R2의 값은 변하지 않는다.
② sr이나 sl에 대해서, 입력비트는 0으로 가정한다.
③ 출력비트의 값은 버려진다.

3. 처리장치의 구성요소

(1) 개요

① 여러 개의 레지스터(레지스터 세트)
② 산술/논리연산장치(ALU)
③ 내부버스(internaul bus)



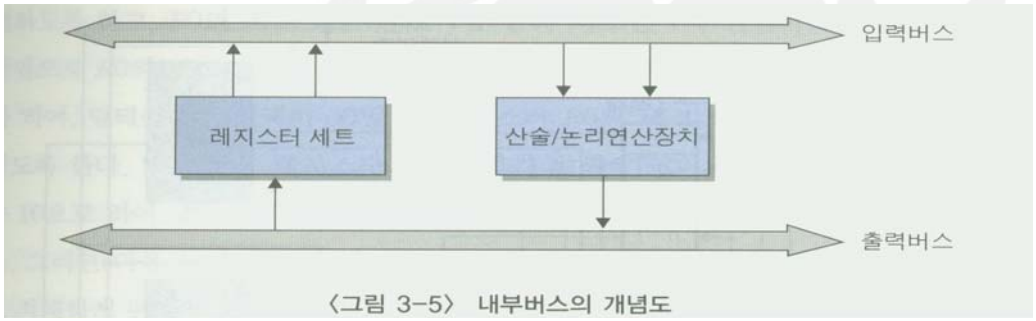
[간단한 처리장치의 내부구성도]

(2) 내부버스

① 레지스터들 간의 데이터 전송을 위한 공용선으로의 집합

② 내부버스를 구성하는 방법: 멀티플렉서와 디코더를 이용

③ 멀티플렉서는 출발 레지스터 선택, 디코더는 도착 레지스터를 선택

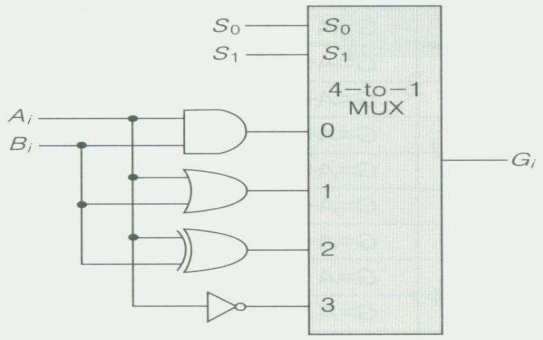


〈그림 3-5〉 내부버스의 개념도

(3) 산술/논리연산장치

① 산술연산회로: 가장 기본적인 요소는 병렬가산기이며 여러 개의 전자가산기 회로를 연속적으로 연결시켜 만듦. 병렬가산기로 들어가는 제어입력값을 선택하여 여러 가지 형태의 산술 연산을 실행

② 논리연산회로: 레지스터에 있는 각 비트를 독립된 2진 변수로 간주하여 비트별 연산을 실행



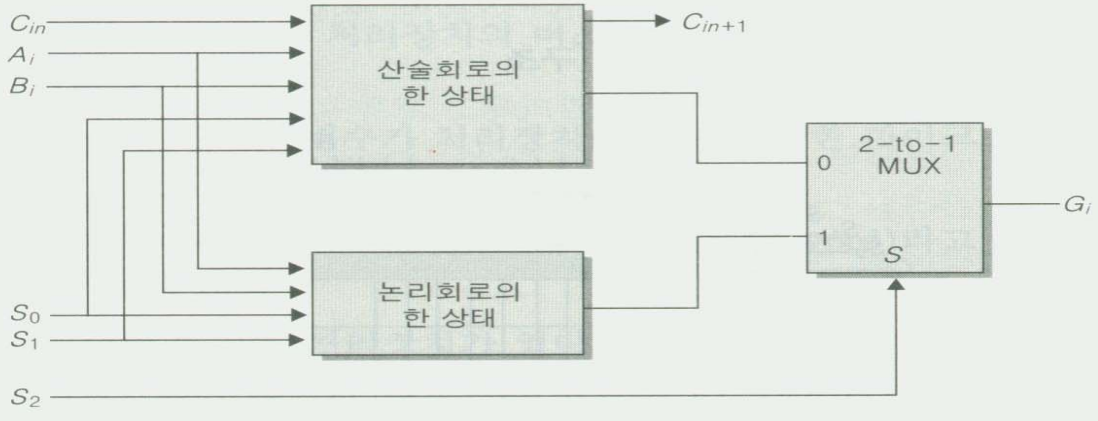
(a) 논리도

S_0	S_1	출 력	연 산
0	0	$G = A \wedge B$	AND
0	1	$G = A \vee B$	OR
1	0	$G = A \oplus B$	XOR
1	1	$G = \overline{A}$	NOT

(b) 함수표

[논리연산회로]

③ 산술/논리연산장치: 논리연산장치와 산술연산장치를 결합



[ALU의 한 단계 구성]

(4) 상태 레지스터

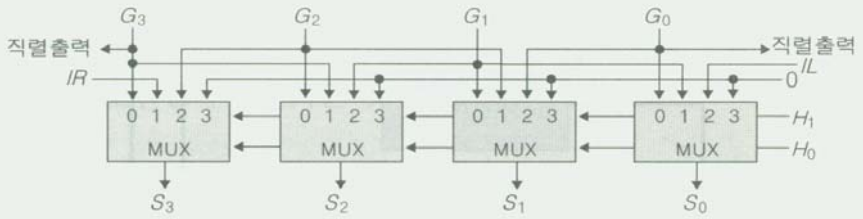
① ALU에서 산술연산이 수행된 후 연산결과에 의해 나타나는 상태값을 저장

② 상태 레지스터들은 C(carry bit), S(sign bit), Z(zero bit), V(overflow bit)로 구성

③ 상태 레지스터의 구조

(5) 쉬프터: 입력 데이터의 모든 비트들을 각각 서로 이웃한 비트로 자리를 옮기는 쉬프트 연산 수행

- 구조



[4비트 쉬프터]

- 쉬프트의 연산

〈쉬프터의 기능표〉

H_1	H_0	연 산	기 능
0	0	$S \leftarrow G$	쉬프트 없이 전송
0	1	$S \leftarrow shr\ G$	우측 쉬프트하여 전송
1	0	$S \leftarrow shl\ G$	좌측 쉬프트하여 전송
1	1	$S \leftarrow 0$	모든 출력비트에 0을 전송

(6) 제어단어

① 처리장치에 있는 선택신호는 처리장치 내에서 수행되는 마이크로연산을 선택하는 변수로서 처리장치의 버스, ALU, 쉬프터, 도착 레지스터 등을 제어

② 선택신호, 즉 제어변수가 처리장치 내에서 특정한 마이크로연산을 선택

③ 이러한 제어변수들의 묶음을 제어단어라고 함

④ 제어단어 생성을 위한 효과적인 방법: 작성된 제어단어를 기억장치에 저장하고, 기억장치의 출력을 처리장치의 각 구성요소의 선택신호로 연결하는 방법. 이렇게 하면 기억장치로부터 연속적인 제어단어를 읽음으로써 처리장치에서의 마이크로 연산이 정해진 순서대로, 연속적으로 수행하는 것이 가능함

<3장 출제예상문제>

1. 마이크로연산에 대한 설명으로 <u>틀린</u> 것은?

① 한 클럭 주기 동안 수행된다.

② 일련의 비트에 대하여 병렬로 수행된다.

③ 레지스터에 저장된 데이터를 대상으로 수행된다.

④ 마이크로연산은 여러 개의 마이크로 명령어로 수행된다.

[해설]
마이크로연산은 레지스터에 저장된 정보를 대상으로 수행하는 연산으로서 한 클럭주기 동안에 일련의 비트들에 대하여 병렬로 수행할 수 있는 가장 기본적인 연산이다. 한편, 하나의 마이크로 명령은 하나 또는 여러 개의 마이크로 연산을 의미함

[정답] 4

2. 다음은 내부버스에 대한 설명이다. 맞는 것은?
① 내부버스란 입출력 장치 간의 데이터 전송을 위한 공통선로의 집합이다.
② 내부버스는 멀티플렉서와 인코더를 이용해서 구성할 수 있다.
③ 내부버스는 멀티플렉서 대신에 세 상태 버퍼를 이용해서 설계할 수 있다.
④ 멀티플렉서를 이용해서 내부버스를 구성하는 방법이 연결선의 수를 줄이는 데 효과가 있다.

[해설]
① 내부버스란 레지스터들간의 데이터 전송을 위한 공용선로의 집합.
② 내부버스를 구성하는 방법 : 멀티플렉서와 디코더를 이용

[정답] 3

3. ALU의 기본 논리연산에 속하지 <u>않는</u> 것은?
① AND ② OR
③ XOR ④ NAND

[해설]
ALU는 산술논리연상장치를 말함(AND, OR, XOR)

[정답] 4

4. 쉬프터에 있는 2진수가 6번 좌로 쉬프트 되어 있다. 이 수는?
① number x 6 ② number / 6
③ number x 64 ④ number / 64

[정답] 3

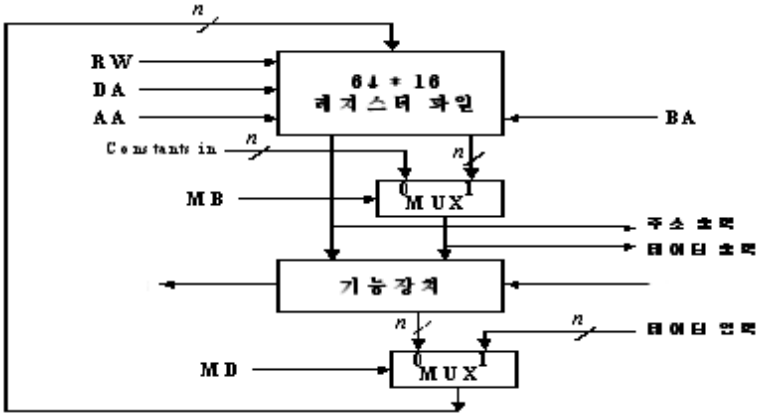
5. 다음에서 처리장치의 구성요소가 <u>아닌</u> 것은?
① 레지스터 ② 산술연산장치
③ 논리연산장치 ④ 시스템버스

[해설] <u>처리장치 구성요소</u>

- 여러 개의 레지스터(레지스터 세트)
- 산술논리연산장치(ALU)
- 내부 버스(internal bus)

[정답] 4

6. 다음의 그림은 처리장치의 블록도로 *n*의 비트는?



① 4 ② 16 ③ 32 ④ 64

[해설]
레지스터 파일의 크기가 64 * 16 인 것은 16 비트 레지스터가 64개 들어 있다는 것이므로, 결국 처리장치가 처리하게 될 데이터의 비트 수(즉, n)는 16

[정답] 2

7. 6번 문제의 그림에 대한 설명이 잘못된 것은?
① ALU는 산술연산회로와 논리연산회로로 구성된다.
② 레지스터 파일은 레지스터와 MUX 및 디코더로 구성된다.
③ 기능장치(function unit)의 주요 구성요소는 ALU와 쉬프터이다.
④ 레지스터 파일 내에서 출발 레지스터는 디코더에 의해, 도착 레지스터는 MUX에 의해 각각 결정된다.

[해설]
기능장치(function unit)는 ALU와 쉬프터를 의미하고 있으며, ALU는 산술연산과 논리연산을 수행하고 쉬프터는 쉬프트연산 수행. 레지스터 파일은 공용으로 사용하는 레지스터들과 디코더, 멀티플렉서 등으로 구성되는데, 도착(destination) 레지스터는 디코더에 의해, 출발(source) 레지스터는 멀티플렉서에 의해 각각 결정

[정답] 4

8. 다음 중 레지스터에 대한 설명으로 틀린 것은?
① 레지스터에 저장되어 있는데이터에 대해서 실행하는 연산을 마이크로연산이라 한다.

② 레지스터는 하나 이상의 연산을 수행할 수 있다.
③ 레지스터는 컴퓨터의 가장 기본적 구성요소이다.
④ 두 레지스터의 내용을 합하는 것은 마이크로연산에 해당하지 않는다.

[해설]
마이크로연산은 대개 한 클럭 사이클 동안 비트 열에 대해서 병렬로 실행되며 마이크로연산의 예를 들면, 한 레지스터의 내용을 다른 레지스터로 옮기는 것이나, 두 레지스터의 내용을 합하는 것, 그리고 레지스터의 내용을 1만큼 증가시키는 것 등

[정답] 4

9. 다음의 지문을 하는 산술 마이크로연산은?

R1에 R2의 2의 보수를 더한 후 R0에 저장

① $R0 \leftarrow R1 + \overline{R2} - 1$
② $R0 \leftarrow R1 + \overline{R2} + 1$
③ $R0 \leftarrow R1 + \overline{R2+1}$
④ $R0 \leftarrow R1 + R2$

[해설]
이진수의 2의 보수를 구하는 방법은 1의 보수를 구한후 1을 더하면 됨. 따라서 $R2$의 2의 보수는 $\overline{R2} + 1$

[정답] 2

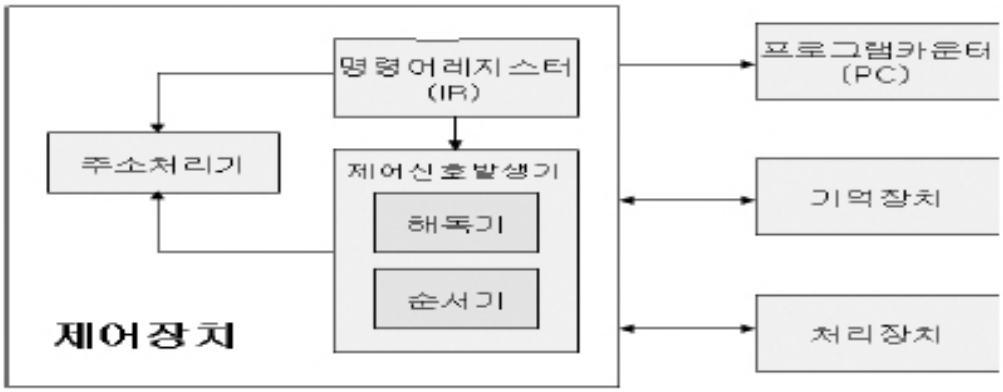
제4장 제어장치

1. 개요

특정한 데이터 연산을 실행할 수 있도록 처리장치에 마이크로연산을 구동시키는 여러 가지 신호들을 제공하는 장치

2. 제어장치의 구성

① 구성요소: 명령어 레지스터, 명령어해독기, 주소처리기, 순서제어기



3. 제어장치의 구현

(1) 마이크로프로그램에 의한 제어장치

① 제어단어와 같은 제어정보를 특별한 기억장치에 0과 1로 기억시킨 구조

② 속도가 느린 단점이 있지만, 제어신호를 수정하고자 할 때는 기억장치의 프로그램을 바꾸기만 하면 되므로 융통성이 뛰어난 장점

(2) 하드웨어에 의한 제어장치

① 순서회로와 같은 게이트와 플립플롭으로 이루어진 구조.

② 고속 동작이 가능하지만, 한번 만들어진 회로를 변경할 수 없으며 다시 설계해야 하는 단점

4. 마이크로프로그램의 의한 제어

마이크로프로그램에 의한 제어구조에서 제어기억장치에는 마이크로명령들이 저장되어 있으며, 저장된 마이크로명령은 하나 또는 다수의 마이크로연산을 나타내는 제어단어를 포함하고 있다.

(1) 제어장치의 구조

① 제어기억장치: 모든 제어정보를 항상 저장하고 있는 ROM이나 RAM

② 제어주소 레지스터: 제어기억장치에 있는 마이크로명령의 주소를 나타낸다.

③ 제어데이터 레지스터: 제어기억장치에서 읽어 온 마이크로명령을 저장하고 있다.

④ 다음주소 생성기(순서기): 제어기억장치에서 제어단어를 읽어 내는 순서를 결정한다.

(2) 마이크로명령어 형식

① A 필드: 처리장치의 레지스터 중 하나의 출발 레지스터 선택(3비트)
② B 필드: 처리장치의 레지스터 중 다른 하나의 출발 레지스터 선택(3비트)
③ D 필드: 처리장치의 레지스터 중 도착 레지스터 선택(3비트)
④ F 필드: 처리장치의 ALU 연산 선택(4비트)
⑥ H 필드: 처리장치의 쉬프트 연산 선택(3비트)
⑦ MUX1 필드: 0과 1로서 내부와 외부 주소를 선택(1비트)
⑧ MUX2 필드: 상태비트의 값에 따라 CAR을 구동(3비트)
⑨ ADRS 필드: ROM의 2진 번지에 해당하는 10진 주소(6비트)

(3) 마이크로프로그램의 작성

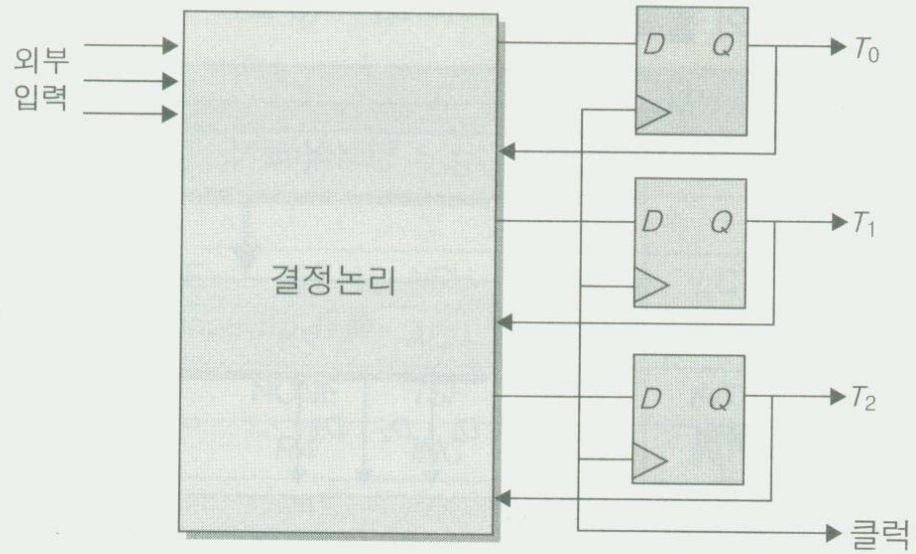
먼저 레지스터 R1의 내용에서 R2의 내용을 빼서 그 결과를 R3에 저장한다. 또한 R1과 R2의 내용 중에서 작은 수를 R4에 더하고, 만약 R1과 R2의 내용이 같으면 R4를 1증가시켜 R4의 내용을 처리장치의 출력단자로 내보낸다.

5. 하드웨어에 의한 제어

하드웨어에 의한 제어장치는 주어진 시간에 처리장치에서 수행할 마이크로 연산을 결정해 주는 제어상태를 갖는 순서회로이다.

(1) 상태 플립플롭을 이용한 제어

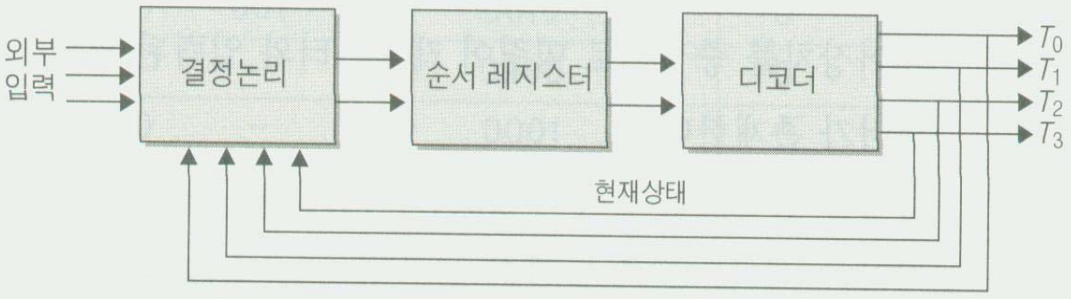
순서제어회로의 각 상태별로 각 한 개의 플립플롭을 할당하는 방법이다.



[상태 플립플롭 제어방식의 구성도]

(2) 순서 레지스터와 디코더를 이용한 제어

제어상태의 순서를 정하는 방법으로 레지스터를 사용하고, 각 상태마다 하나의 출력만을 나오도록 하기 위해 디코더를 사용하는 방법이다.



[순서 레지스터와 디코더를 이용한 제어방식의 구성도]

<4장 출제예상문제>

1. 제어장치의 구성요소가 아닌 것은?

① 순서제어논리 장치 ② 명령어 인출기

③ 명령어 해독기 ④ 제어 메모리

[해설] 제어장치 구성요소

명령어 레지스터, 명령어 해독기, 주소 처리기, 순서 제어기

[정답] 2

2. 다음에서 설명한 보기 중 그 특징이 나머지와 다른 것은?

① 기억장치에 여러 개의 단어로 2진 제어값을 저장하여 사용하는 제어장치이다.

② PC와 같은 레지스터가 존재하지 않는다.

③ 소형 디지털 시스템의 설계에는 비용이 많이 든다.

④ 하드웨어 구성이 한 번 만들어진 후 다른 제어순서를 구성하기 위해서는 연결 형태를 변화시킬 필요가 없다.

[해설]

①, ③, ④ 마이크로프로그램 제어방식에 관한 설명

② 하드웨어로 구현한 제어방식에 관한 설명이다. 하드웨어로 구현한 제어장치에서는 기억장치에서 명령어를 가져오거나 명령어의 실행순서를 처리하는데 관여하지 않으며 PC 와 같은 레지스터가 존재하지 않고 입력과 상태비트만으로 판단하여 연산순서와 그때 수행할 연산을 정함

[정답] 2

3. 다음 중 디지털 시스템에는 두 가지 서로 다른 유형의 제어장치가 있는데 알맞게 짝지은 것은?
① 소프트웨어로 구현한 제어장치 - 하드웨어로 구현한 제어장치
② 하드웨어로 구현한 제어장치 - 마이크로프로그램으로 구현한 제어장치
③ 마이크로프로그램으로 구현한 제어장치 - 소프트웨어로 구현한 제어장치
④ 마이크로프로그램으로 구현한 제어장치 - 마이크로오퍼레이션으로 구현한 제어장치

[해설]
전체적인 측면에서 보면, 디지털 시스템에는 두가지 서로다른 유형의 제어장치가 있음. 하나는 마이크로프로그램으로 구현한 제어장치이고 하나는 하드웨어로 구현한 제어장치

[정답] 2

4. 다음은 명령어 형식에 대한 설명이다. 틀린 것은?
① 명령어 형식에는 레지스터 형식과 즉치형식 두 가지가 있다.
② 목적지 레지스터는 계산된 결과가 저장되는 레지스터로 DR로 표시한다.
③ 즉치형식에서는 연산의 대상이 되는 오퍼랜드값이 2개 표시되어 있다.
④ 레지스터 형식에서는 연산의 대상이 되는 출발지 레지스터가 2개 표시되어 있다.

[정답] 3

5. 명령어해독기에 대한 설명이다. 맞는 것은?
① 명령어해독기는 조합논리회로로 명령어를 제어단어로 변환하여 처리장치에 공급한다.
② 명령어에서 목적지 레지스터는 제어단어의 필드 F로 변환된다.
③ 명령어에서 출발지 레지스터 A는 제어단어의 필드 D로 변환된다.
④ 명령어에서 출발지 레지스터는 B는 제어단어의 필드 H로 변환된다.

[정답] 1

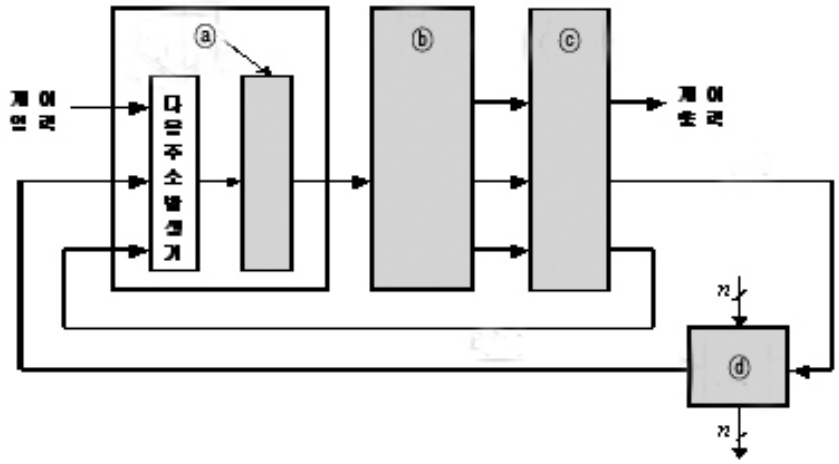
6. 다음 중 마이크로 프로그램 제어 구조는?
① 제어기억장치는 보통 ROM으로 구성하며, 각 단어는 컴퓨터 명령어이다.
② 제어논리의 변경이 필요한 경우 여러 요소들 사이의 와이어링을 바꾸어야 한다.
③ 하드웨어로 구현한 제어구조보다 속도가 빠르다.
④ 2진 제어변수를 제어기억장치에 저장하여 구현한다.

[해설]
마이크로 프로그램 제어구조는 제어함수나 제어단어와 같은 제어정보를 특별한 기억장치에 2진 정보로서 (0과 1로) 기억시킨 구조로서 하드웨어로 구현한 제어구조에 비해 연산의 수행속도가 느리다는 단점이 있지만, 제어논리의 변경이 필요한 경우 단순히 기억장치의 내용을 바꾸기만 하면 가능하다는 융통성이 있음. 반면에 하드웨어로 구현한 제어구조는 수행 속도가 빠르지만 일정한 설계법이 없으며 변경이 필요할

경우 하드웨어적으로 와이어링을 바꾸어야 하는 등 융통성이 떨어짐

[정답] 4

7. 다음의 그림에 대한 설명으로 옳은 것은?



① ⓐ - 제어주소를 보관하는 레지스터로서 다음에 수행할 마이크로 명령어의 주소를 저장하고 있다.
② ⓑ - 주기억장치로서 정적 마이크로 프로그래밍을 위해서 보통 ROM으로 구성된다.
③ ⓒ - 컴퓨터 명령어(instruction)를 잠시 보관하는 레지스터로서 이 컴퓨터 명령어는 제어출력, 처리장치로 입력되는 제어신호(A), 다음 주소를 생성하기 위한 정보(B)로 구성된다.
④ ⓓ - 캐쉬 기억장치로서 컴퓨터 명령어를 처리하는 동안의 중간 결과물을 저장한다.

[해설]
② 제어기억장치
③ 마이크로 명령어를 잠시 보관하는 레지스터이다.
④ 처리장치로서 제어단어(A)의 지시에 따라 n비트 입력 데이터를 처리하여 n비트 데이터로 출력하며, 이 때 발생되는 상태정보를 다음주소 발생기로 출력

[정답] 1

8. 4단계 파이프라인에서 실행될 다음 지문의 프로그램을 보고 NOP 명령어를 두는 이유는?

1. ADD R1, R0, R1 2. ADD R3, R2, R3 3. ADD R5, R4, R5 4. ADD R7, R6, R7 5. ADD R3, R1, R3 6. NOP 7 ADD R7, R5, R7 8. NOP 9. NOP 10. ADD R7, R3, R7	① 프로그램이 서브루틴으로 분기시 복귀주소를 저장하기 위해. ② 4단계 파이프라인에서 실행될 때 프로램이 원활하게 작동하게 하기 위해 ③ 프로그램을 간단하게 하기위해서 ④ 합산한 값의 보수를 구하기위해.

[해설]

지문의 프로그램에서 NOP 명령이 없다면, 4단계 파이프라인 실행시, 4번 명령의 결과가 저장되지 않은 상태에서 7번 명령어가 수행되므로, 데이터 해저드 현상이 발생하게 됨. 단순화된 파이프라인에서 이와 같은 오류를 피하기 위해 프로그램내에 NOP을 사용함.

[정답] 2