

한국방송통신대학교 자료포털 노우존

www.knouzone.com



컴퓨터구조

2018학년도 2학기 기말시험

핵심체크 및 출제예상문제



범위 : 교재 전 범위

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.

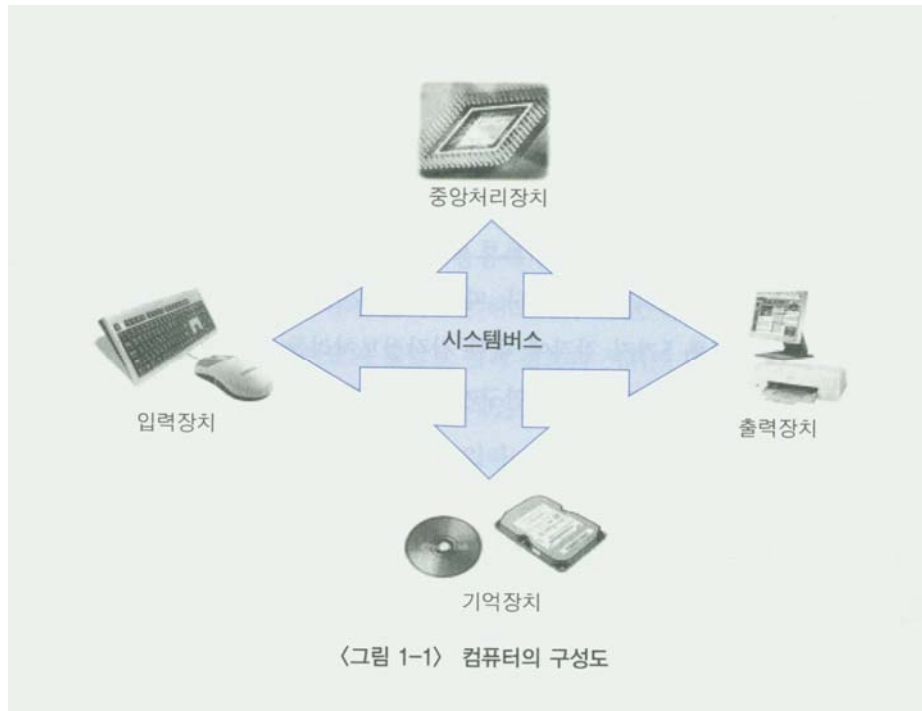
(4) 제4세대 컴퓨터

- 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



(1) 입력장치

사용자가 입력하는 데이터와 명령어를 받아서 컴퓨터가 알 수 있는 형태로 변환하여 기억 장치나 중앙처리장치로 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. (SRAM으로 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.

[정답] 2

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

- [illegible]

[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-집적회로 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]

버스는 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 나뉘어진다.

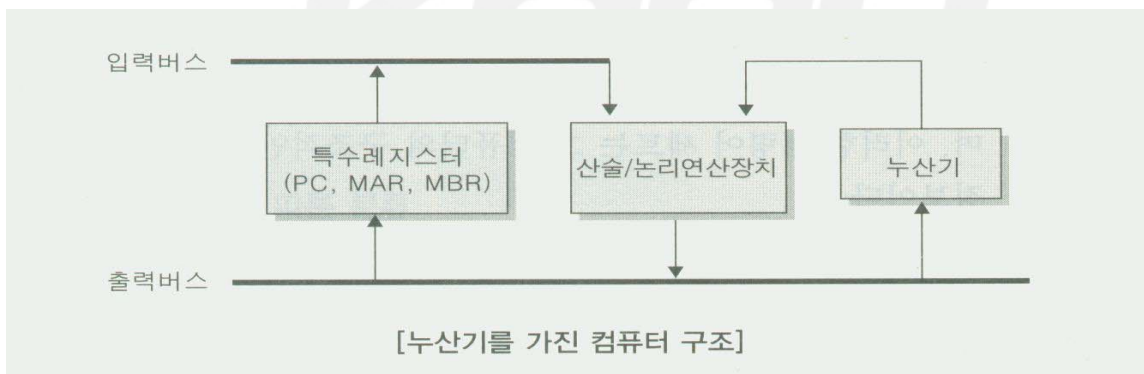
연산코드 (OP code)	오퍼랜드(operand)
----------------	---------------

3 명령어 형식

- * 명령어는 컴퓨터의 내부구조에 따라 여러 가지 형식이 있다.

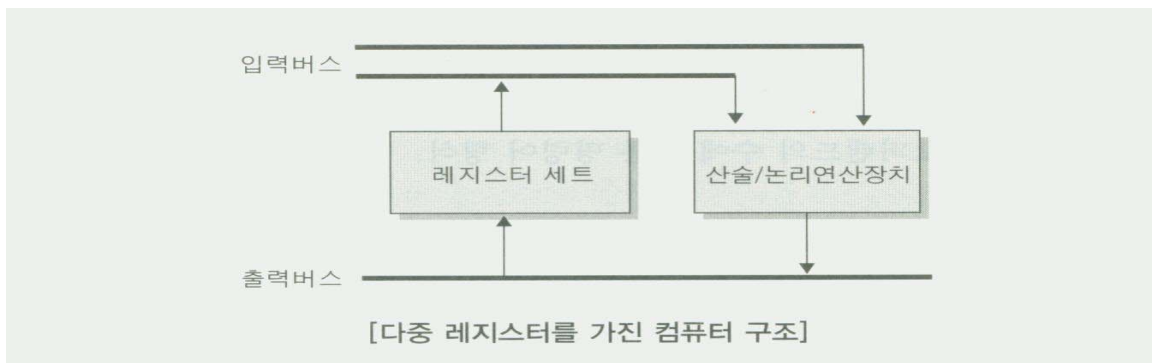
(1) 기억장소에 따른 명령어 형식

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



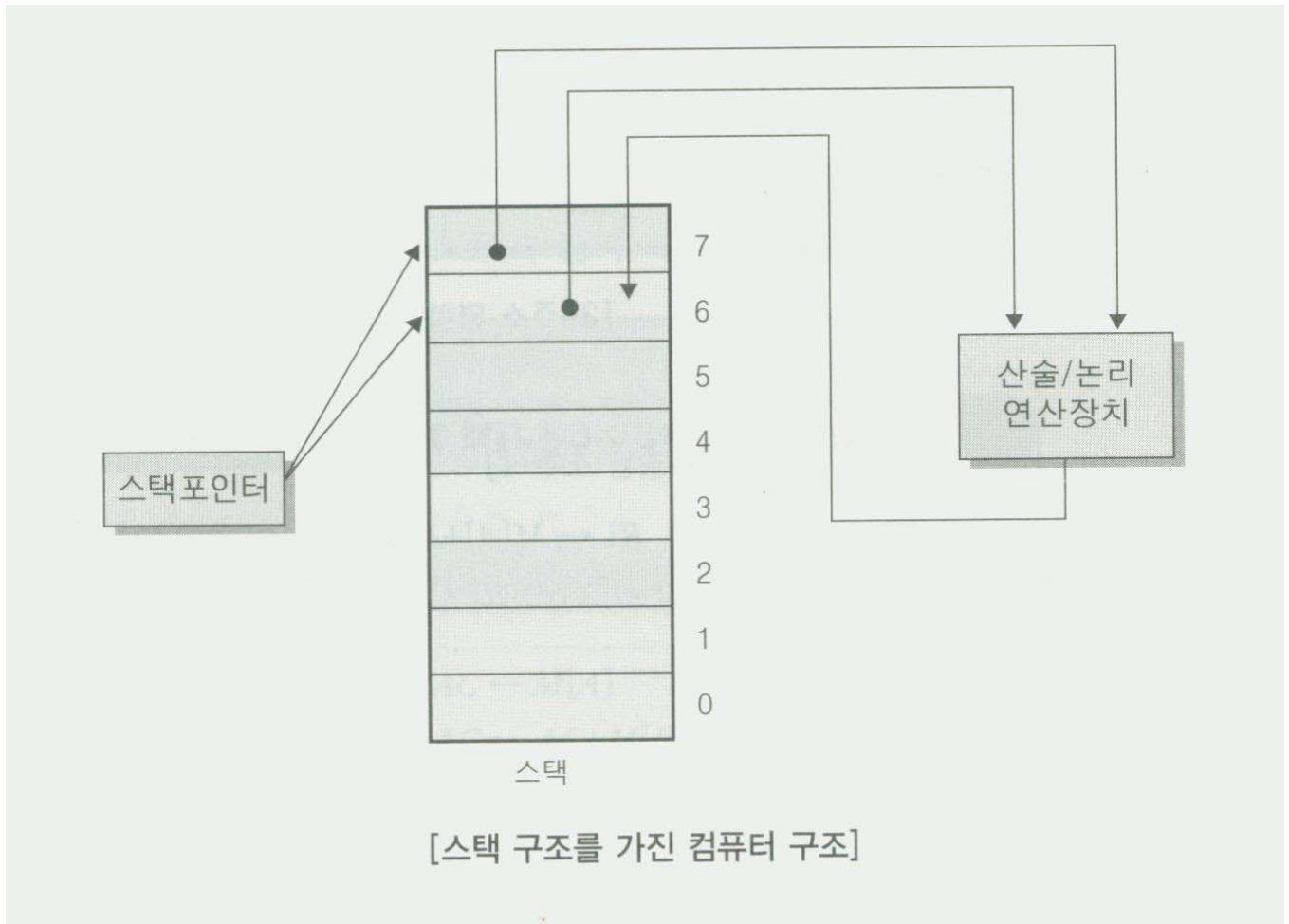
(예: ADD X ; AC ← AC + M[X])

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



(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.



(예: ADD ; TOS \leftarrow 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 오퍼랜드의 개수가 두 개인 명령어 형식이다.
- ③ 1-주소 명령: 오퍼랜드의 개수가 하나인 명령어 형식으로서, 기억장치로부터 오퍼랜드를 가져오거나 연산 결과를 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 방식으로 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 의미의 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 LDI 자체 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 방식으로, 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 $X=(Y+B)XC$ 에 대해 3-주소 명령어를 이용한 프로그램

ADD A, B, R1 ; $R1 \leftarrow M[A] + M[B]$

MUL R1, C, X ; $M[X] \leftarrow R1 + M[C]$

[정답] 4

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

LOAD	A	; AC \leftarrow M[A]
ADD	B	; AC \leftarrow AC + M[B]
STORE	X	; ()
LOAD	C	; AC \leftarrow M[C]
MUL	X	; AC \leftarrow AC \times M[X]
STORE	X	; M[X] \leftarrow AC

- ① $M[X] \leftarrow AC$ ② $AC \leftarrow M[X]$
③ $M[X] \leftarrow M[B]$ ④ $M[B] \leftarrow M[X]$

[해설]

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

[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

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)

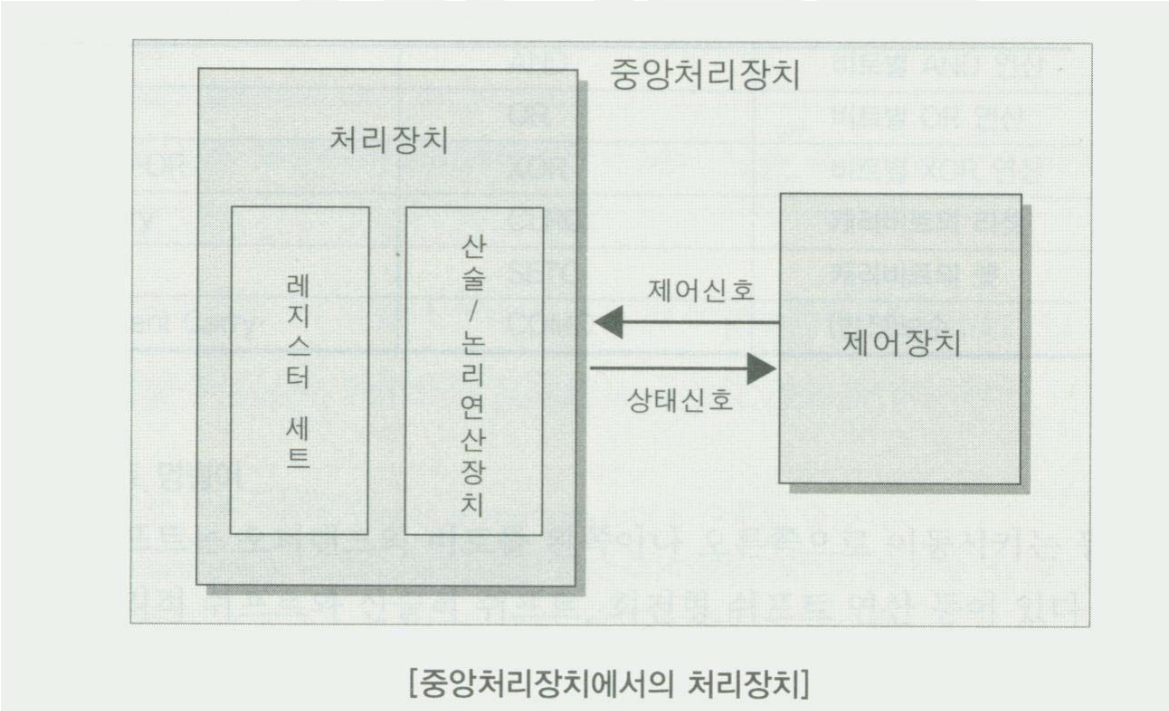
③ RET(Return)는 프로그램 제어명령임

[정답] 4

제3장 처리장치

1. 개요

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



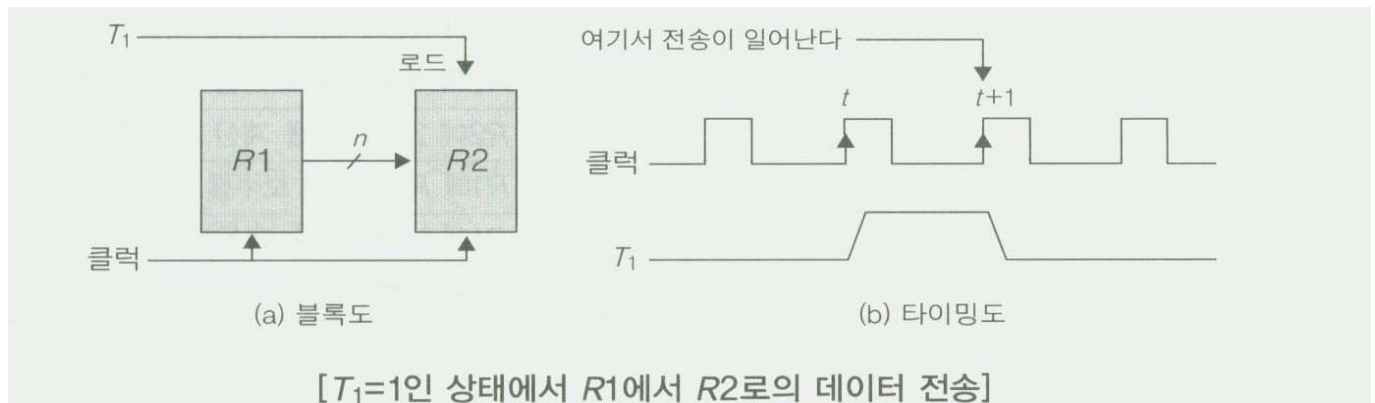
2. 마이크로연산

- * 정의: 레지스터나 기억장치에 있는 데이터에 대해 이루어지는 기본적인 연산

(1) 레지스터 전송 마이크로연산

- ① 한 레지스터에서 다른 레지스터로의 데이터 전송은 연산자 ' \leftarrow '로 표시 (예: $R2 \leftarrow R1$)
- 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
침표	동시에 실행되는 두 개 이상의 마이크로연산 구분	R1←R2, R2←R1



(2) 산술 마이크로연산

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

기호	의미
$R0 \leftarrow R1 + R2$	R1과 R2의 합을 R0에 저장
$R2 \leftarrow \overline{R2}$	R2의 보수(1의 보수)를 R2에 저장
$R2 \leftarrow R2 + 1$	R2에 2의 보수를 계산 후 저장
$R0 \leftarrow R1 + \overline{R2} + 1$	R1에 R2의 2의 보수를 더한 후 R0에 저장
$R1 \leftarrow R1 + 1$	R1에 1 더함(상승카운트)
$R1 \leftarrow R1 - 1$	R1에 1뺀(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)
$R0 \leftarrow R1 \wedge R2$	비트별 논리적 AND(비트 클리어)
$R0 \leftarrow R1 \vee R2$	비트별 논리적 OR(비트 세트)
$R0 \leftarrow R1 \oplus R2$	비트별 논리적 XOR(비트별 보수)

(4) 쉬프트 마이크로연산

- ① 데이터의 측면이동에 사용

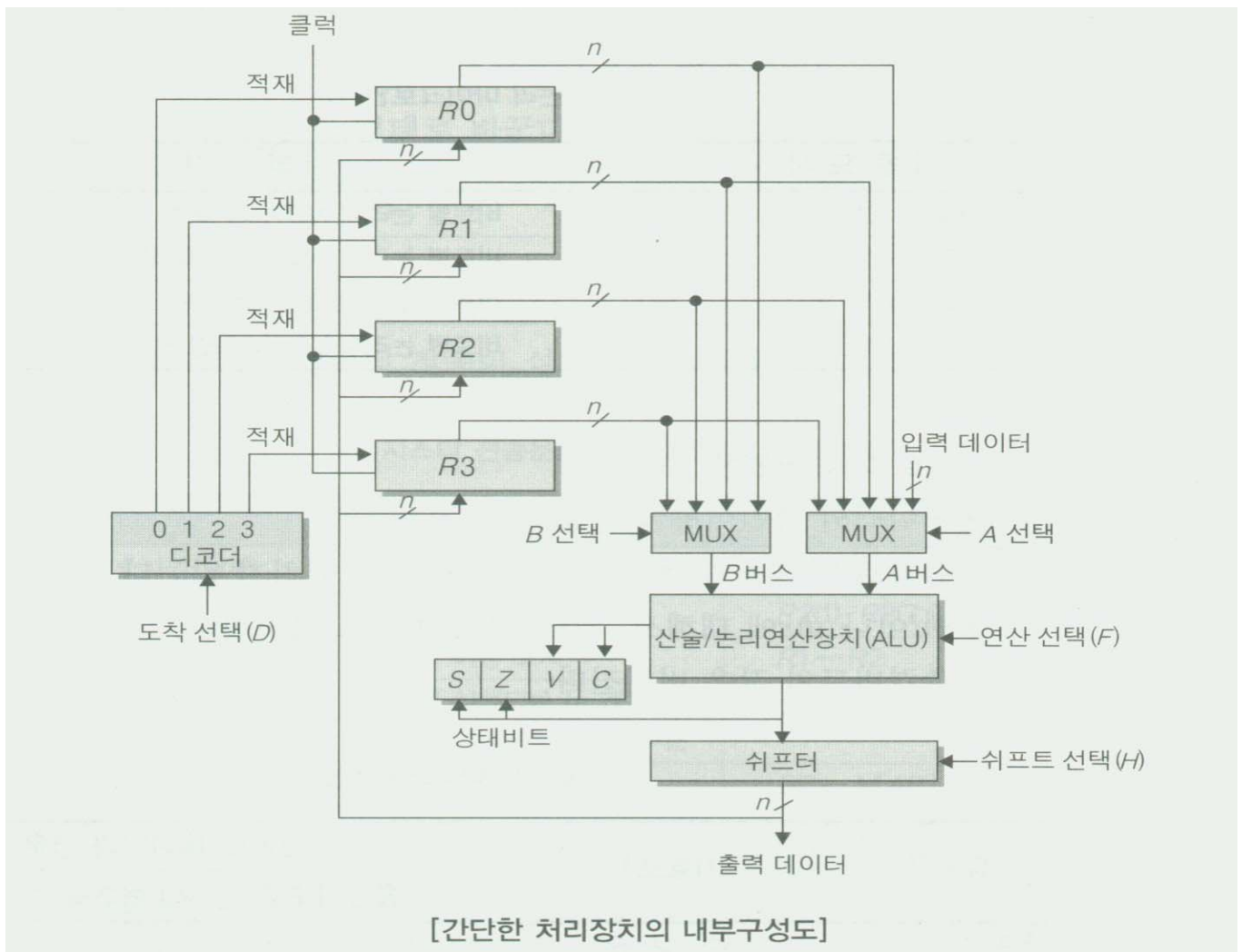
기호	의미	8비트 데이터의 경우	
		출발지 R2	쉬프트 후: 목적지 R1
왼쪽 쉬프트	$R1 \leftarrow sl\ R2$	10011110	00111100
오른쪽 쉬프트	$R1 \leftarrow sr\ R2$	11100101	01110010

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

3. 처리장치의 구성요소

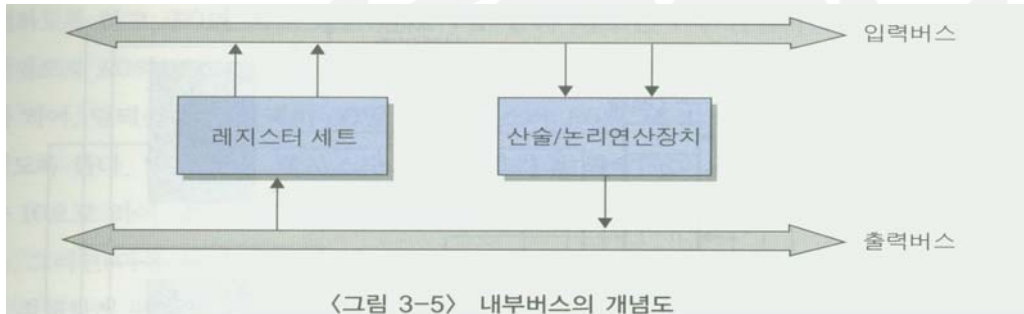
(1) 개요

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



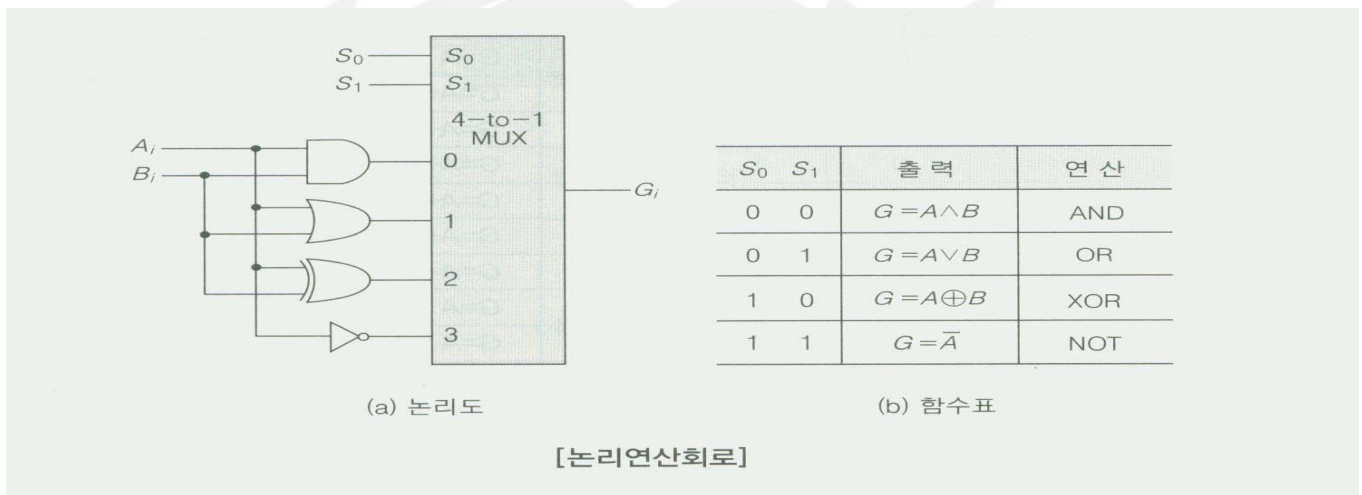
(2) 내부버스

- ① 레지스터들 간의 데이터 전송을 위한 공용선으로의 집합
- ② 내부버스를 구성하는 방법: 멀티플렉서와 디코더를 이용
- ③ 멀티플렉서는 출발 레지스터 선택, 디코더는 도착 레지스터를 선택

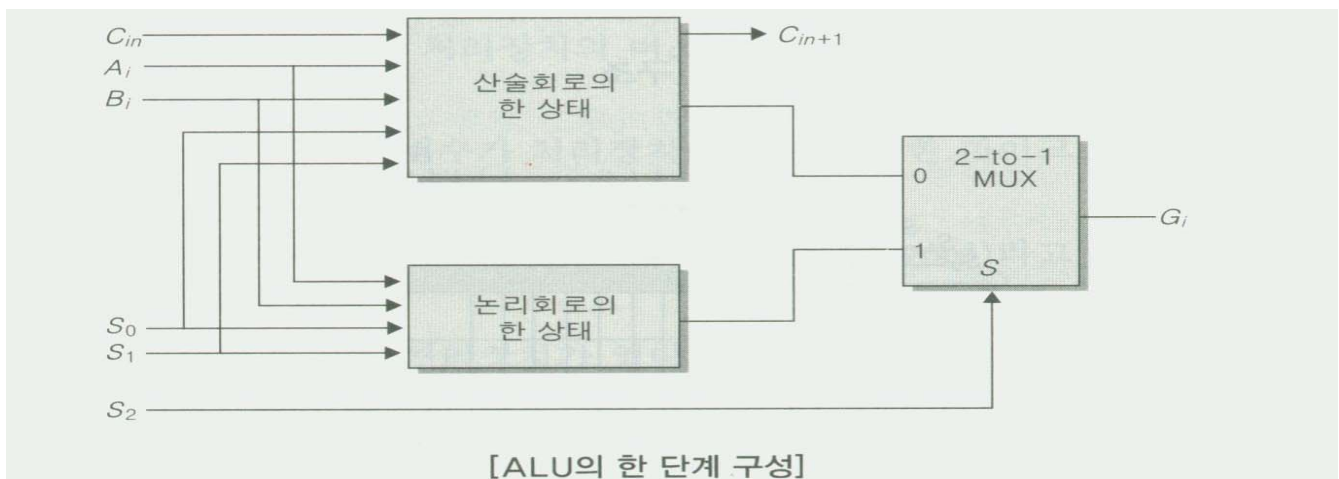


(3) 산술/논리연산장치

- 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



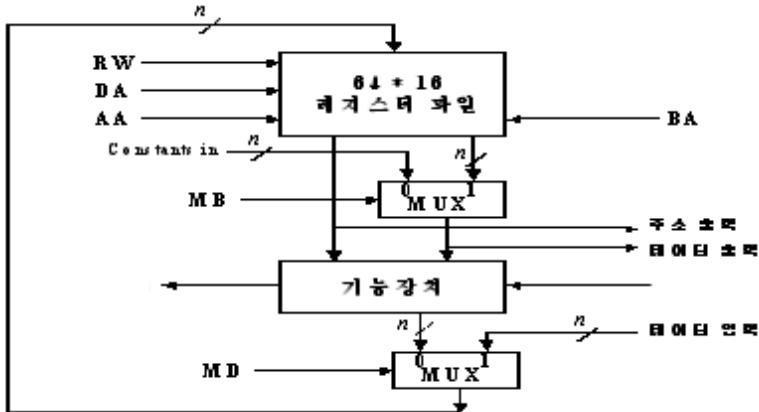
- ③ 산술/논리연산장치: 논리연산장치와 산술연산장치를 결합



- 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 쉬프트이다.
 ④ 레지스터 파일 내에서 출발 레지스터는 디코더에 의해, 도착 레지스터는 MUX에 의해 각각 결정된다.

[해설]

기능장치(function unit)는 ALU와 쉬프트를 의미하고 있으며, ALU는 산술연산과 논리연산을 수행하고 쉬프트는 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 있는 데이터에 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 R2의 2의 보수는 $\overline{R2} + 1$

[정답] 2

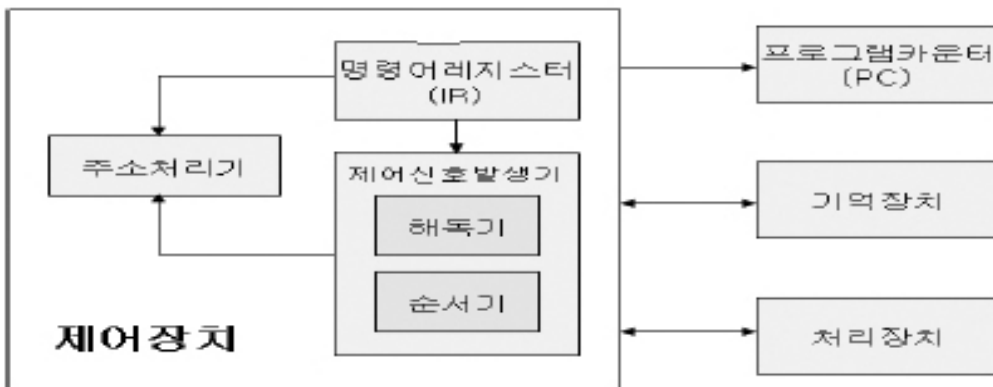
제4장 제어장치

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

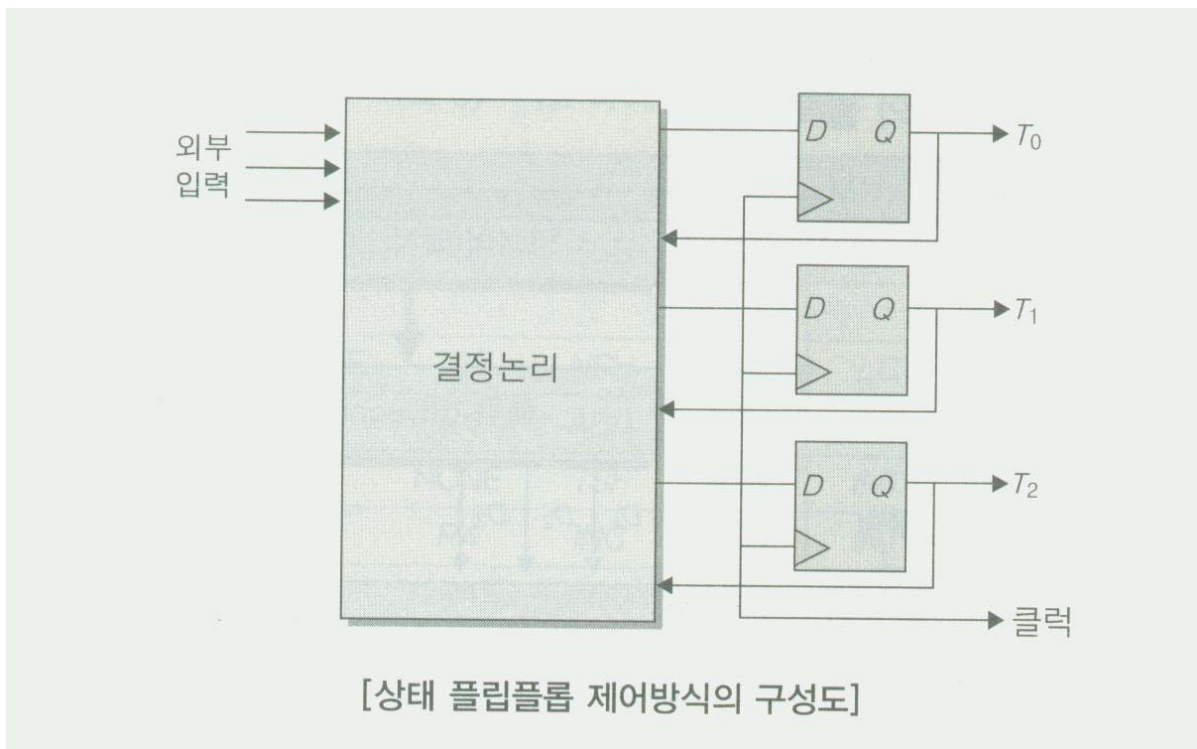
먼저 레지스터 R1의 내용에서 R2의 내용을 빼서 그 결과를 R3에 저장한다. 또한 R1과 R2의 내용 중에서 작은 수를 R4에 더하고, 만약 R1과 R2의 내용이 같으면 R4를 1증가시켜 R4의 내용을 처리장치의 출력단자로 내보낸다.

5. 하드웨어에 의한 제어

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

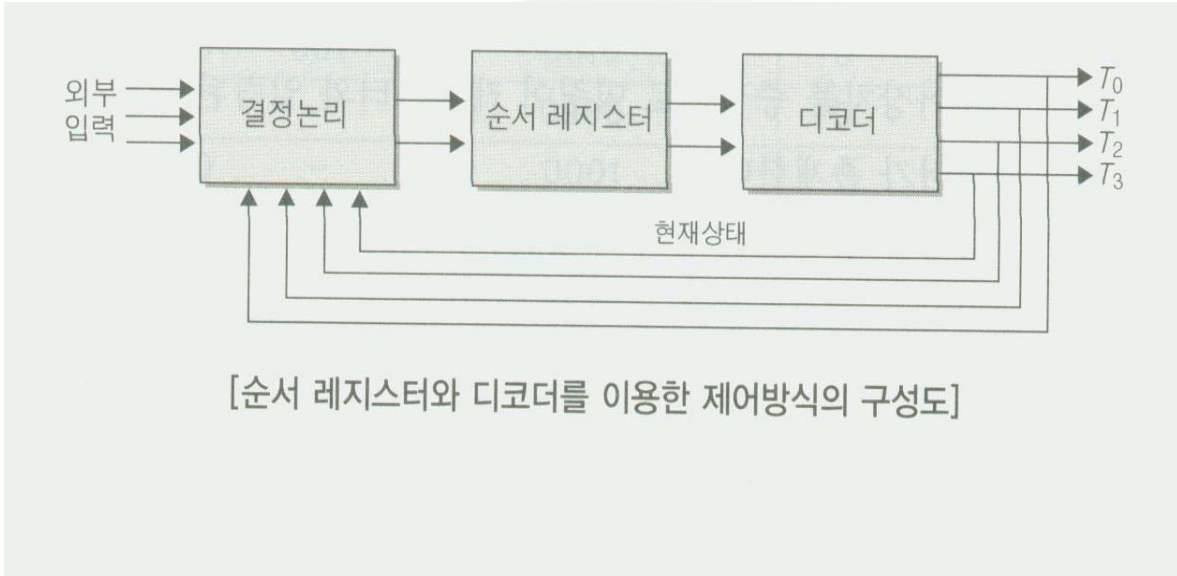
(1) 상태 플립플롭을 이용한 제어

순서제어회로의 각 상태별로 각 한 개의 플립플롭을 할당하는 방법이다.



(2) 순서 레지스터와 디코더를 이용한 제어

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



<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.

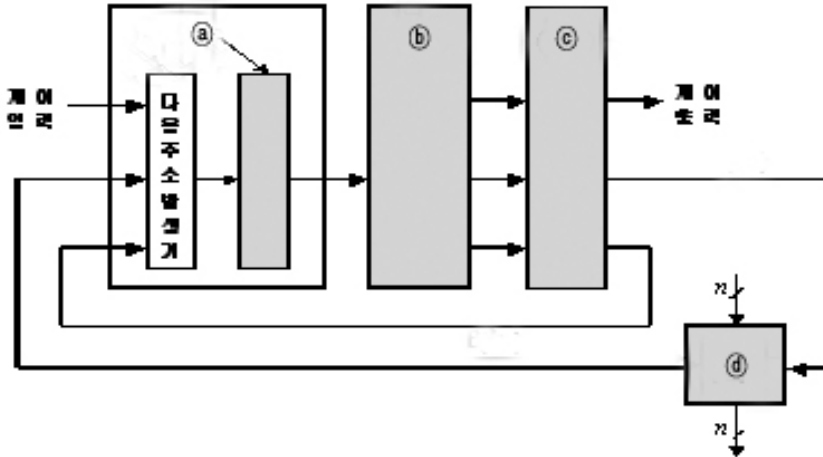
[해설]

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 ㉓ - 제어주소를 보관하는 레지스터로서 다음에 수행할 마이크로 명령어의 주소를 저장하고 있다.
- ② ㉔ - 주기억장치로서 정적 마이크로 프로그래밍을 위해서 보통 ROM으로 구성된다.
- ③ ㉕ - 컴퓨터 명령어(instruction)를 잠시 보관하는 레지스터로서 이 컴퓨터 명령어는 제어출력, 처리장치로 입력되는 제어신호(A), 다음 주소를 생성하기 위한 정보(B)로 구성된다.
- ④ ㉖ - 캐시 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 발생하는 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 프로그램이 원활하게 작동하게 하기 위해
- ③ 프로그램을 간단하게 하기위해서
- ④ 합당한 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

제5장 중앙처리장치

1. 개요

컴퓨터에서 명령어를 수행하고 데이터를 처리하는 장치로서 프로그램이 수행되는 전반적인 과정을 제어하고 처리장치와 제어장치로 나눌 수 있다.

2. 중앙처리장치의 내부구조

- ① 산술/논리연산장치: CPU에서 연산장치.
- ② 레지스터 세트: CPU 내부의 레지스터.
- ③ 제어장치: 프로그램에 의해 주어지는 연산의 순서를 차례대로 실행하기 위해 제어신호 발생.
- ④ 내부버스: ALU와 레지스터 간의 데이터 전송을 위한 데이터버스와 제어장치로부터 발생하는 제어신호를 위한 버스로 구성

3. 레지스터

(1) 범용 레지스터

- ① 여러 가지 연산을 위해 데이터 저장, 주소 저장과 같은 일반적인 목적을 위해 사용되는 레지스터
- ② CPU 내부에 있는 소규모의 일시적인 기억장치로 프로그램의 진행 도중 가까운 시간내에 사용할 데이터나 연산결과를 일시적으로 기억시키는데 사용한다.
- ③ 데이터를 연산할 때 메모리로부터 데이터를 인출할 경우 호출시간이 많이 걸리기 때문에 CPU 내부의 레지스터에 데이터를 기억시켜두고 연산한다.

(2) 특수 레지스터

- ① 프로그램 카운터:
 - 프로그램 카운터는 데이터가 저장되어 있는 기억장치의 주소를 지정해 주는 레지스터
 - 현재 처리하려고 하는 데이터를 인출한 후에는 자동적으로 1 증가하여 다음에 수행해야 할 명령어가 저장된 기억장치주소를 지정해 줌
- ② 명령어 레지스터
 - 프로그램의 수행 중에 가장 최근에 기억장치로부터 인출되어진 명령어를 갖고 있음
 - 명령어 레지스터가 가지는 비트수는 명령어의 연산코드의 비트수와 같음
- ③ 누산기
 - 데이터를 일시 저장하는 레지스터
 - 입력장치로부터 데이터를 받아들이거나, 출력장치로 데이터를 전송하는데 사용됨
 - CPU가 연산을 수행한 후 그 결과는 반드시 누산기에 저장함
- ④ 기억장치주소 레지스터: 기억장치주소를 일시 저장하는 레지스터
- ⑤ 기억장치 버퍼 레지스터: 기억장치로 쓰일 데이터나 혹은 기억장치로부터 읽힐 데이터를 임시로 저장하는 레지스터
- ⑥ 스택 포인터
 - 스택 주소지정방식에서 사용됨
 - 스택 영역의 번지를 지정해 주는 포인터
 - 프로그램 카운터와 같은 크기의 비트수를 가짐

4. 명령어 사이클

한 개의 명령어를 CPU에서 수행하는 데 필요한 전체 수행 과정

(1) 명령어 인출 사이클

- ① 기억장치에 기억되어 있는 명령어를 인출하는 과정
 - ② 명령어 인출 사이클을 마이크로연산 표현을 이용하여 살펴보면 다음과 같은 세 단계로 이루어 짐
- ```
T0 : MAR ← PC
T1 : MBR ← M[MAR], PC ← (PC)+1
T2: IR ← MBR
```

##### (2) 명령어 실행 사이클

- ① 명령어를 실행하는 단계
- ② 이 과정에서는 명령어 인출과정을 통하여 IR 레지스터에 실린 명령어를 해독하고, 해독한 명령어에 따라 필요한 연산이 수행됨
- ③ LOAD 명령어의 실행 사이클 동안의 마이크로연산 표현은 다음과 같음

```
T0 : MAR ← IR(adrs)
T1 : MBR ← M[MAR]
T2: AC ← MBR
```

##### (3) 간접 사이클

- ① 간접주소지정방식을 사용하는 명령어에서 오퍼랜드 부분의 유효주소를 결정하는데 사용됨
- ② 간접 사이클의 마이크로연산은 다음과 같으며, 세 개의 클럭으로 완성됨

```
T0 : MAR ← IR(adrs)
T1 : MBR ← M[MAR]
T2: IR ← MBR
```

##### (4) 인터럽트 사이클

- ① CPU의 정상적인 동작 중에 입출력장치로부터의 인터럽트 처리의 요청이 발생하였을 때 실행됨
- ② 인터럽트: CPU가 현재 처리 중인 프로그램 루틴을 중단시키고 다른 동작을 수행하도록 하는 것을 말함
- ③ 다음은 인터럽트 사이클의 마이크로연산을 나타냄

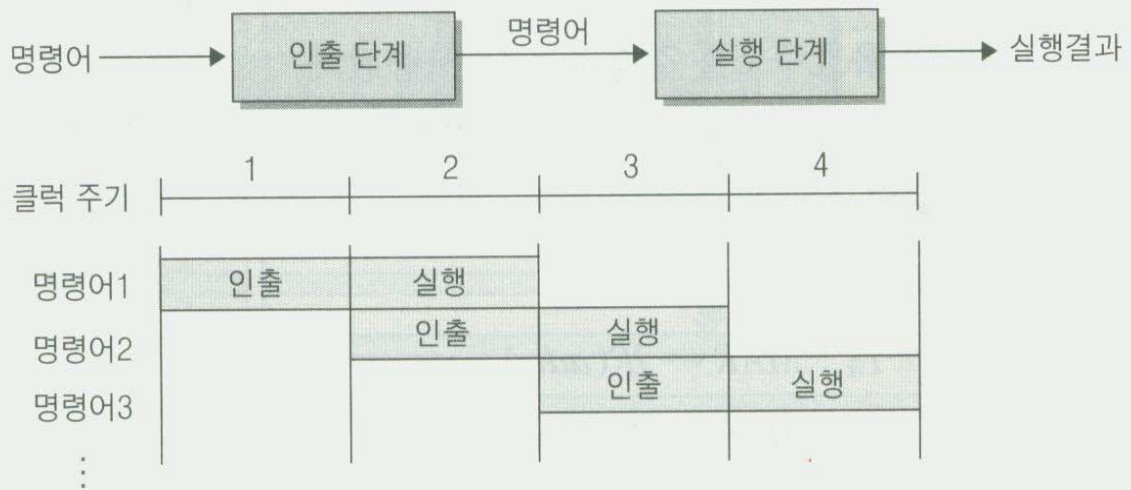
```
T0 : MBR ← PC
T1 : MBR ← SP, PC ← ISR(adrs)
T2: M[MAR] ← MBR
```

#### 5. 간단한 구조의 컴퓨터 설계

##### (1) 명령어 분석

- ① 명령어 인출 단계
- ② 명령어 실행 단계
- ③ 제어 논리

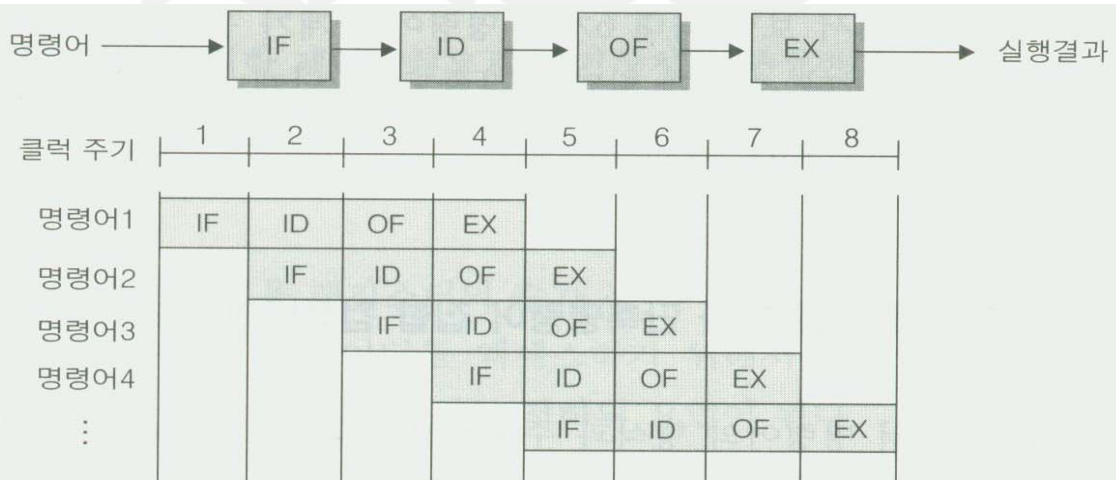




[2단계 명령어 파이프라인 실행 흐름도]

## (2) 4단계 명령어 파이프라인

명령어 수행 단계를 4단계, 즉 명령어 인출(IF), 명령어 해독(ID), 오퍼랜드(OF), 실행단계(EX)로 나누어 수행하는 방법



[4단계 명령어 파이프라인 실행 흐름도]

## (3) 6단계 명령어 파이프라인

명령어 수행을 여섯 단계, 즉 명령어 인출하는(FI) 단계, 명령을 해독하는(DI) 단계, 오퍼랜드를 계산하는(CO) 단계, 오퍼랜드를 인출하는(FO) 단계 명령어를 실행하는(EI) 단계, 연산된 결과 오퍼랜드에 저장하는(WO) 단계로 나누어 수행하는 방법

## 7. 복합명령어 집합 컴퓨터와 단축명령어 집합 컴퓨터의 비교

### (1) CICS와 RISC의 특징





- ③ ALU에서는 제어신호에 의해 덧셈을 수행하고 그 결과를 누산기에 저장한다.
- ④ 계산결과는 외부 시스템버스를 통해 다시 캐시메모리로 전달된다.

[정답] 4

4. 4단계 명령어 파이프라인에 해당되지 않는 단계는?

- ① 연산코드 인출                      ② 명령어 해독
- ③ 명령어 인출                      ④ 실행 단계

[해설]

- 명령어 인출 단계(IF : Instruction Fetch stage): 명령어를 기억장치로부터 인출하는 과정
- 명령어 해독 단계(ID : Instruction Decoding stage): 해독기를 이용해 인출된 명령어를 해석하는 과정
- 오퍼랜드 인출 단계(OF : Operand Fetch stage): 해석된 명령어에 의해 메모리 참조가 필요하다면 오퍼랜드를 인출해 오는 과정
- 명령어 실행 단계(EX : Execution stage): 해석된 결과로서 지정된 연산을 수행하는 과정

[정답] 1

5. 다음은 RISC의 특징이다. 맞는 것은?

- ① 상대적으로 많은 수의 명령어 집합을 가지고 있다.
- ② 고정된 길이의 명령어로 구성되어 있다.
- ③ 제어장치가 마이크로프로그램 방식으로 설계되었다.
- ④ 파이프라인 수행에 적합하지 않다.

[해설] RISC

- 고정길이의 명령어 형식을 사용하기 때문에 명령어 집합이 단순하다.
- 모든 명령어가 1 워드 단위로 고정되어 있기 때문에 1개 명령어를 수행하는데 1개 클럭만이 소요됨
- 약 30여 개의 명령어만을 가지고 있다.

[정답] 2

6. 다음 중 중앙처리장치가 기억장치에서 명령어를 가져오는 것은?

- ① direct cycle                      ② interrupt cycle
- ③ fetch cycle                      ④ execute cycle

[해설] 명령어 수행 사이클

- ② 명령어 해석: 인출되어진 명령어가 무엇을 수행하라고 지시하는지를 알기 위하여 명령어를 해독하는 과정. 이 과정에서 명령어 종류에 따라 필요한 경우에 오퍼랜드의 인출이 일어남
- ③ 명령어 인출: 기억장치로부터 명령어를 가져오는 것을 말한다. 인출되어진 명령어의 주소는 프로그램 카운터(PC)에 들어있으며, 인출되어진 명령어는 명령어 레지스터(IR)에 옮겨짐
- ④ 명령어 실행: 지정된 연산 수행. 이때 간접주소방식의 명령어인 경우에는 기억장치로부터 데이터가 들어

있는 유효주소를 가져옴

[정답] 3

7. 프로그램 카운터는?

- ① 마이크로명령을 저장한다.
- ② 프로그램 데이터를 저장한다.
- ③ 프로그램을 카운트하여 +1씩 증가한다.
- ④ 실행을 기다리는 명령의 번지를 저장한다.

[해설]

프로그램 카운터(program counter)는 다음에 수행되어질 명령어의 주소를 갖고 있는 특수 레지스터

[정답] 3

8. 다음 중 일반적으로 CPU는 크게 어떻게 나뉘는가?

- ① 처리장치와 제어장치
- ② 시스템 버스 와 기억장치
- ③ 출력장치와 제어장치
- ④ 기억장치와 처리장치

[해설]

일반적으로 CPU는 대개 두 부분 즉, 처리장치와 제어장치로 나눌 수 있음. 처리장치는 레지스터들, ALU, 쉬프트, 내부버스들로 구성되어 있으며, 내부버스는 레지스터들과 ALU, 쉬프트, 다른 컴퓨터 구성요소들간에서 데이터를 전송하기 위한 경로를 제공함. 처리장치는 파이프라인화 될 수도 있고 그렇지 않을 수도 있음. 제어장치는 프로그램 카운터, 명령어 레지스터, 제어논리로 구성되어 있으며, 마이크로프로그램화되거나 하드와이어화되어 구현될 수 있음

[정답] 1

9. 다음 중 명령어 형식에 대한 설명으로 그 성격이 다른 하나는 무엇인가?

- ① 명령어들 중 몇 개는 처리하는데 여러 클럭사이클이 요구하는 복잡한 동작을 수행한다
- ② 명령어의 대부분은 기억장치 내용을 오퍼랜드로 직접 동작시킬 수 있다
- ③ 8개의 주소지정 방식들과 두 가지 명령어 형식길이를 가지고 있다.
- ④ 각 명령어 형식은 32비트 워드 1개로 구성된다.

[해설]

- ①, ②, ③ CISC 구조를 명확하게 식별할 수 있는 특징
- ④ RISC 에 대한 설

[정답] 4

## 제6장 기억장치

### 1. 개요

기억장치는 프로그램과 데이터를 저장하는 장치로서 컴퓨터 시스템의 필수적인 요소이다.

#### (1) 지역성의 원래

- ① 시간적 지역성: 어떤 내용이 한번 참조되면, 곧바로 다시 참조되기 쉽다.
- ② 공간적 지역성: 어떤 내용이 참조되면, 그 내용에 가까운 곳에 잇는 다른 내용들이 곧바로 참조 되기 쉽다.

#### (2) 기억장치 계층

지역성의 원리를 이용하기 위해 기억장치를 계층으로 구현하며, 기억장치 계층은 속도와 크기가 다른 기억장치를 여러 층으로 구성한 것.

#### (3) 기억장치 성능평가

- ① 기억용량
- ② 접근시간
- ③ 사이클 시간
- ④ 기억장치 대역폭
- ⑤ 가격

### 2. 주기억장치

컴퓨터 시스템에서 주기억장치는 필수적인 하드웨어 요소이며 주기억장치의 접근시간은 컴퓨터 시스템 성능의 중요한 제한요소에 해당된다. 특히 접근시간과 기억장치 대역폭, 기억용량은 주기억장치의 성능평가에는 가장 중요한 요소라고 할 수 있다.

#### (1) 복수모듈기억장치

기억장치모듈을 여러 개 가지고 있는 기억장치이다.

#### (2) 연관기억장치

기억장치에 기억되어 있는 데이터의 내용을 이용하여 기억장치에 접근할 수 있기 때문에 내용주소화 기억장치라고도 한다.

#### (3) 캐시기억장치

주기억장치보다 속도가 빠른 기억소자로 구성되어 중앙처리장치와 주기억 장치 간의 속도 차이를 줄여 주는 역할을 한다.

#### (4) 가상기억장치

보조기억장치와 같이 기억용량이 큰 기억장치를 마치 주기억장치처럼 사용 하는 개념이다.

### 3. 보조기억장치

당장 필요하지는 않지만 필요할 때에 주기억장치로 옮겨 사용할 수 있는 자료를 저장하는 장치이다.

#### (1) 자기디스크

금속이나 플라스틱으로 만들어진 원형판을 자화된 물질로 코팅한 기억매체.

#### (2) 자기테이프

디스크에서처럼 데이터는 물리적 레코드라고 불리는 연속적인 블록 단위로 읽기연산과 쓰기연산이 실행된다.

#### (3) 광기억장치

빛을 이용한 보조기억장치로 콤팩트디스크 디지털 오디오시스템기술을 컴퓨터 데이터 저장장치에 응용한 장치이다.

① CD: 디지털화된 음향정보 저장

② CD-ROM: CD보다 재생장치가 좀 더 정교하며 오류 수정코드 가짐

③ CD-R: 한 번 데이터를 기록하여 그 다음부터는 읽기만 가능

④ CD-RW: 재기록이 가능한 CD.

⑤ DVD: 영상 저장매체

#### (4) RAID 시스템

현재 대형 컴퓨터에서 사용되고 있는 고가의 대용량 디스크 시스템을 의미하는 SLE(Single Large Expensive Disk)를 저가의 작은 디스크 배열로 대체하기 위해 RAID (Redundant Arrays of Inexpensive Disks) 시스템이 개발되었다. RAID 시스템은 크기가 작고, 저가인 여러 개의 하드디스크를 묶어 하나의 기억장치처럼 사용하는 디스크 시스템이다.

-RAID 0레벨

-RAID 1레벨

-RAID 2레벨

-RAID 3레벨

-RAID 4레벨

-RAID 5레벨

-RAID 6레벨

## <6장 출제예상문제>

1. 다음의 내용과 관련 깊은 것은?

“어떤 내용이 한 번 참조되면, 곧바로 다시 참조되기 쉽다.”

① 공간적 지역성

② 시간적 지역성

③ 내부적 참조성

④ 외부적 참조성





의 고 대역폭을 필요로 하는 어떠한 애플리케이션이건 RAID 0용으로 아주 적합하다

- ② RAID 2는 몇 개의 디스크에 데이터를 스트라이핑한 다음, 별개의 디스크나 디스크들에 ECC(error-correction code)를 기록한
- ④ RAID 3에는 최소 3개의 드라이브(두 개의 데이터, 하나의 패리티)가 필요하며, 데이터 드라이브들을 가로 질러 정보를 스트라이핑하면서 동시에 스트라이핑을 위한 패리티 정보를 별개 드라이브에 기록하도록 만들어졌다.

[보충학습] RAID 시스템

크기가 작고, 저가인 여러 개의 하드디스크를 묶어 하나의 기억장치처럼 사용하는 디스크 시스템

[정답] 2



## 제7장 입출력 시스템

### 1. 개요

컴퓨터는 입출력 시스템에 의해 외부 데이터를 중앙처리장치나 기억장치로 읽어들이기도 하고 외부로 출력시키기도 한다. 이러한 컴퓨터 입출력 시스템은 중앙 처리장치와 마찬가지로 기술의 발달에 따라 영향을 받는다.

### 2. 입출력 시스템의 구성요소

#### (1) 입출력장치

중앙처리장치 또는 기억장치와 외부의 입출력 매체 사이에 정보를 전송하는 것이 주된 임무이다.

#### (2) 입출력장치 제어기와 입출력 제어기

① 입출력 제어기: 주기억장치와 입출력장치 사이에 데이터 전달이 이루어지게하는 제어기.

② 입출력장치 제어기: 입출력장치의 하드웨어 기능을 제어하는 목적.

#### (3) 입출력장치 인터페이스

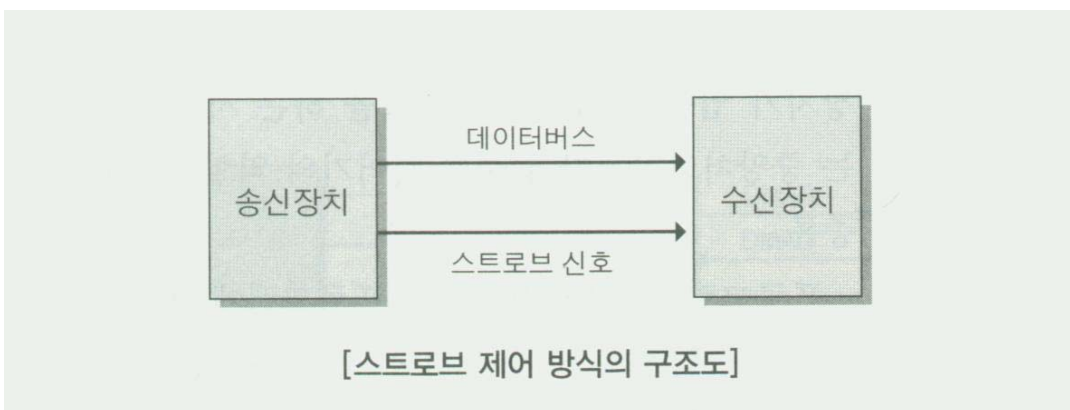
입출력 포트라고 부르기도 하며, 입출력 시스템의 기능 중에서 각 입출력장치의 고유한 특성에 관련되는 것을 포함하는 하드웨어 장치이다.

### 3. 비동기 입출력 데이터 전송

비동기 입출력장치는 자율적으로 동작하므로 중앙처리장치의 동작과 일치하지 않는다. 따라서 비동기적인 장치 사이에서의 데이터 전송은 데이터를 일방적으로 전송하면 수신측에서 데이터가 수신되지 않을 경우에도 수신된 것으로 착각할 수 있다. 그러므로 제어선을 사용하여 상대방에게 신호를 보내 주어 준비를 하도록 해야 한다.

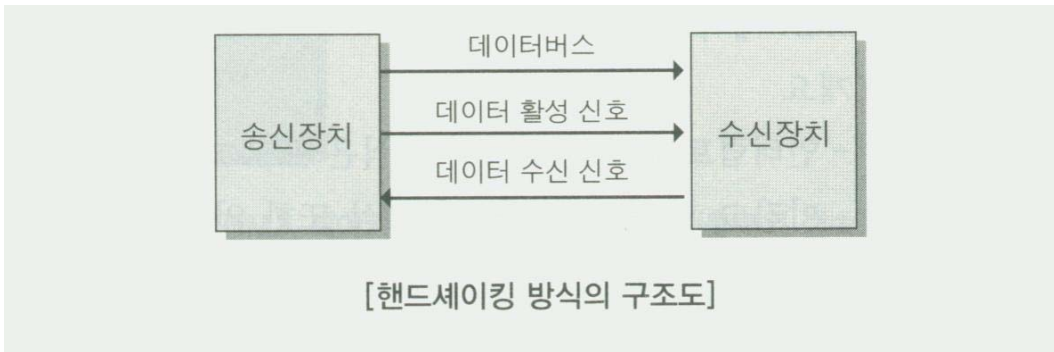
#### (1) 스트로브 제어

비동기적 데이터 전송 방식의 하나인 스트로브 제어 방식은 데이터 전송의 시간을 맞추기 위해 데이터 전송 회선 이외에 하나의 제어회선을 더 갖는다.



#### (2) 핸드셰이킹

스트로브 신호 방식의 단점을 보완해 줄 수 있는 방식으로 높은 신뢰성과 융통성을 가진 방식이다.



#### 4. 입출력 제어 방식

##### (1) 중앙처리장치에 의한 제어 방식

가장 간단한 입출력 시스템은 독립된 입출력 제어기가 없고, 중앙처리장치가 입출력 제어기의 역할을 하는 시스템이다.

##### (2) DMA 제어 방식

인터럽트에 의한 입출력 방식은 프로그램에 의한 방식보다는 효율적이라고 할 수 있으나, 이 방식 또한 인터럽트 처리를 위한 중앙처리장치의 추가적인 작업이 필요하기 때문에 중앙 처리장치의 많은 시간을 낭비하게 된다.

##### (3) 채널에 의한 제어 방식

채널이라고 하는 일종의 입출력 프로세서를 이용하여 입출력 작업을 전담하도록 하는 방식이다.

### <7장 출제예상문제>

1. 다음 중 컴퓨터 내부장치와 입출력장치의 차이점이 아닌 것은?

- |                |                 |
|----------------|-----------------|
| ① 데이터 전송속도의 차이 | ② 데이터 처리 단위의 차이 |
| ③ 에러율의 차이      | ④ 데이터 종류의 차이    |

[해설] 내부 장치와 입출력장치의 차이점

- 데이터 전송 속도의 차이
- 데이터 처리 단위의 차이
- 에러율의 차이

[정답] 4

2. 다음 중 컴퓨터 시스템 내에 있는 버스의 계층구조에 속하지 않는 것은?

- |               |            |
|---------------|------------|
| ① 중앙처리장치 내부버스 | ② 주기억장치버TM |
| ③ 제어버스        | ④ 입출력버스    |



- ② 인터럽트로 시작되는 데이터 전송에 사용되는 통신방식이다.
- ③ 전송오류가 발생하면 time-out 기법으로 검출할 수 있다.
- ④ 송수신장치의 능동적 참여로 데이터 전송의 신뢰성이 있다.

[해설]

두 개의 핸드셰이킹선은 송신장치가 발생하는 데이터 준비(data ready)선과 수신장치가 발생하는 인지(acknowledge)선. 핸드셰이킹은 송,수신장치의 능동적 참여로 데이터를 전송하기 때문에 유연성과 신뢰성이 매우 높다. 또한 데이터 전송오류를 검출하기 위해서 time-out 기법을 사용한다.

[정답] 2

6. 다음 중 DMA제어기와 연관 없는 것은?

- ① 채널
- ② MCG(memory cycle grant)
- ③ 사이클 스틸
- ④ MCR(memory cycle request)

[해설]

DMA 제어기가 주기억장치에 접근하려면 우선 주기억장치 접근을 위하여 주기억장치 사이클을 요청(MCR: memory cycle request)해야 하며, 그 요청이 허용(MCG:memory cycle grant)될 때 읽기 쓰기 제어 신호를 전달해야함. DMA 제어기와 중앙처리기는 같은 주기억장치 버스를 이용하여 주기억장치에 접근하는데, 중앙 처리 장치는 프로그램을 수행중일 때에는 매 사이클마다 주기억장치에 접근하려 하기 때문에 DMA 방식에 의하여 입출력이 일어날 때에는 DMA 제어기와 중앙 처리 장치는 경쟁적으로 주기억장치 사이클을 사용하게 된다. 이러한 경쟁을 해결하는 방법으로 DMA 방식에 의한 입출력 시스템에서는 사이클 스틸(cycle steal) 방법을 채용한다.

[정답] 1

# 제8장 병렬처리 시스템

## 1. 개요

병렬처리란 하나 이상의 연산을 동시에 수행함으로써, 연산속도를 증가시키려는 처리방법을 말하고 이러한 기법으로 구현한 컴퓨터 구조를 병렬처리기라고 한다.

## 2. 컴퓨터 시스템의 성능

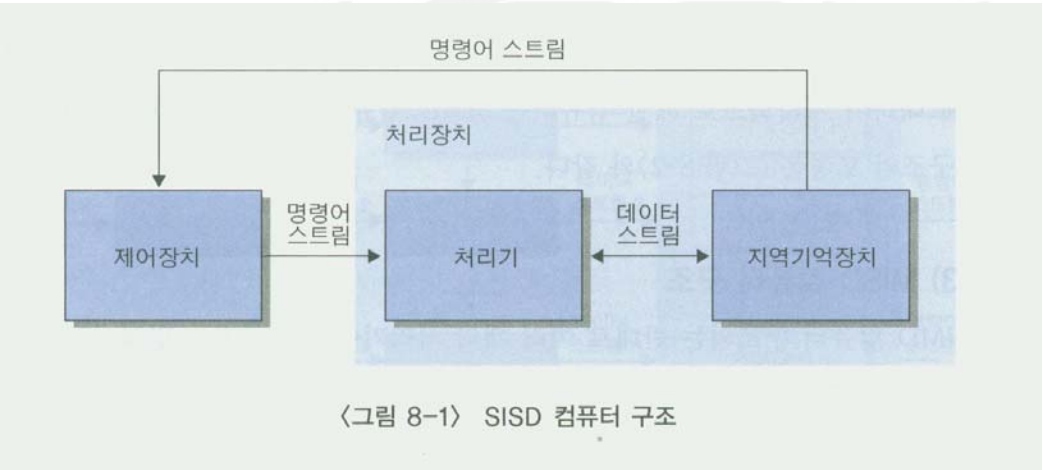
일반적으로 병렬처리기가 이전의 순차처리기에 비해서 놀라운 성능향상을 가져왔다고 간주한다. 여기서 두 컴퓨터를 비교하는 데 있어서의 기준이 되는 성능의 의미를 이해하는 것이 병렬처리를 파악하는 데 있어서 출발점이 된다.

## 3. 병렬처리 시스템의 분류

### (1) 플린의 분류

#### ① SISD 컴퓨터 구조: 단일 명령어 스트림-단일 데이터 스트림 컴퓨터

- 현재의 컴퓨터는 대개 이러한 구조로 되어 있으며 명령어가 순차적으로 하나씩 수행된다. 허나 명령어를 수행할 때 여러 수행 단계로 나누어서 중첩 수행 혹은 파이프라인에 의한 수행을 할 수도 있다.



#### ② SIMD 컴퓨터 구조: 단일명령어 - 복수 데이터 스트림 컴퓨터

- 배열 처리기 구조, 여러 개의 처리기가 하나의 제어 처리기에 의하여 제어되는 구조를 가지고 있다.



③ MISD 컴퓨터 구조: 복수 명령어 스트림 - 단일 데이터 스트림 컴퓨터

- SIMD 컴퓨터 구조와는 반대로 여러 개의 처리기에서 수행되는 명령어는 서로 다르지만 전체적으로 하나의 데이터 스트림을 가지고 있는 형태이다. 실제로 이러한 처리 방식이 이용되기는 어렵기 때문에 이러한 구조의 컴퓨터는 없다.

④ MIMD 컴퓨터 구조: 복수 명령어 스트림 - 복수 데이터 스트림 컴퓨터

- 다중 처리기 컴퓨터 시스템이나 다중 컴퓨터 시스템은 이러한 구조에 속하며, 여러 개의 처리기는 각각 다른 데이터 스트림에 대하여 다른 명령어 스트림을 수행한다.

(2) 팡의 분류

팡(Feng)은 컴퓨터 구조를 병렬수행 정도에 따라 분류하였다.

- ① 단어별 순차, 비트별 순차 처리
- ② 단어별 병렬, 비트별 순차 처리
- ③ 단어별 순차, 비트별 병렬 처리
- ④ 단어별 병렬, 비트별 병렬 처리

(3) 구조에 의한 분류

- ① 공유기억장치 시스템: 공유기억장치를 가지는 다중처리기 구조는 처리기와 기억장치 모듈 사이에 완전한 연결성이 있는 강결합 시스템이다.
- ② 메시지 전달 시스템: 전형적인 메시지 전달 방식은 여러 개의 컴퓨터 모듈과 상호연결망으로 구성되며, 데이터 통신은 공유변수가 아니라 메시지를 통하여 이루어진다.

4. 병렬처리기의 종류

(1) 파이프라인 처리기

프로그램 내에 내재하는 시간적 병렬성을 활용하기 위하여 프로그램 수행에 필요한 작업을 시간적으로 중첩하여 수행시키는 처리기를 의미한다.





[해설] MISD 컴퓨터 구조

SIMD 컴퓨터 구조와는 반대로 여러 개의 처리기에서 수행되는 명령어는 서로 다르지만 전체적으로 하나의 데이터 스트림을 가지고 있는 형태. 실제로 이러한 처리 방식이 이용 되기는 어렵기 때문에 이러한 구조의 컴퓨터는 없음

[정답] 1

2. 다음 중 구조에 의해 분류한 것으로 옳은 것은?

- ① SISD, SIMD, MISD, MIMD
- ② 다중장치 처리기, 파이프라인 처리기
- ③ 버스, 크로스바, 트리, 입방체
- ④ 공유기억장치 시스템, 메시지 전달 시스템

[해설] 구조에 의한 분류

- 다중처리기: 오직 하나의 운영체제를 가지면서 기억장치 공간의 입출력 자원을 공유
- 다중 컴퓨터: 자신만의 처리기, 기억장치, 입출력, 운영체제를 가진 여러 개의 컴퓨터로 구성
- 다중프로세싱: 병렬적이면서 비동기적인 컴퓨터 시스템에서 동시에 여러 개의 태스크를 수행

[정답] 4

3. 프로그램 수행에 필요한 작업을 시간적으로 중첩하여 수행시키는 처리기는?

- ① 파이프라인 처리기
- ② 배열 처리기
- ③ 다중처리기
- ④ 다중배열처리기

[해설] 파이프라인 처리기

- 프로그램 내에 내재하는 시간적 병렬성(temporal parallelism)을 활용하기 위하여 프로그램 수행에 필요한 작업을 시간적으로 중첩하여 수행시키는 처리기를 의미
- 파이프라인 처리원리를 명령어 수행에 적용시킨 것을 명령어 수행 파이프라인(instruction execution pipeline 혹은 instruction pipeline)이라고 부름

- ② 배열처리기: 한 컴퓨터 내에 여러 개의 처리장치(processing element: PE)들을 배열 형태로 가지고 있을 때 이를 배열처리기라 부름
- ③ 다중처리기: 프로그램의 수행 속도를 포함한 성능의 개선이 주된 목표이지만 신뢰성(reliability), 유연성(flexibility), 그리고 가용성(availability)등을 개선하는 것도 그 목표에 포함됨

[정답] 1

4. 다음 중 병렬처리기를 처리기의 상호연결망 구조에 따라 분류한 것으로 옳은 것은?

- ① SISD, SIMD, MISD, MIMD
- ② 다중장치 처리기, 파이프라인 처리기
- ③ 버스, 크로스바, 트리, 입방체



④ 공유기억장치 시스템, 메시지 전달 시스템

[해설]

① Flynn 분류법, ② 중점기법, ③ 처리기 상호연결구조, ④ 구조에 따라 병렬처리를 분류

[정답] 3

5. 일반적으로 배열처리기 구조라고도 하는데, 여러 개의 처리기가 하나의 제어처리에 의하여 제어되는 구조를 가지고 있는 구조는?

- ① 단일 명령어 스트림-단일 데이터 스트림 컴퓨터(SISD)
- ② 단일 명령어 스트림-복수 데이터 스트림 컴퓨터(SIMD)
- ③ 복수 명령어 스트림-단일 데이터 스트림 컴퓨터(MISD)
- ④ 복수 명령어 스트림-복수 데이터 스트림 컴퓨터(MIMD)

[해설] SIMD 컴퓨터 구조

배열 처리기 구조, 여러 개의 처리기가 하나의 제어 처리기에 의하여 제어되는 구조를 가지고 있음

[정답] 2

6. 다음 중 정적 상호 연결망 구조에 해당하는 것은?

- ① 성형 구조                      ② 버스 구조
  - ③ 크로스바 구조                ④ 다단계 네트워크

[해설] 정적 상호연결망

- 성형 구조(star topology): 허브(hub)라고 하는 특별히 할당된 노드와 연결되는 연결도 1의 노드들로 구성
- 완전 상호연결: 각 프로세서들과 각 기억장치 모듈들 사이에 가능한 연결이 모두 존재하는 것
- 선형 구조(linear topology): 이용 가능한 가장 단순한 구조
- 링 구조(ring topology): 양끝이 연결되어 있는 선형 네트워크. 결함 노드가 있는 경우에 메시지는 다른 길로 감으로써 그 노드를 피해갈 수 있음
- 트리 구조: 노드의 연결도는 일정하게 유지하면서 확장 가능한 구조라는 장점이 있지만 통신 거리는 상대적으로 김
- 메쉬(mesh)나 토러스(torus) 구조: 프로세서 배열에 적합
- 하이퍼큐브(hypercube) 구조는 기하학적으로는  $n$ 차원 공간에서 정의되는 큐브의  $2n$ 개의 꼭지점에 노드를 가진 구조라고 정의될 수 있음

②, ③, ④ 동적 상호연결망에 속함

[정답] 1

7. 다음 중 동적 상호연결망 구조에 해당하는 것은?

- ① 성형 구조                      ② 버스 구조

③ 메시 또는 토러스 구조

④ 다단계 네트워크

[해설] 동적 상호연결망 구조

- 버스구조: 단일 버스 구조는 사용되는 프로세서의 수가 적을 때, 즉 최대 버스 교통량이 버스 대역폭에 비해서 적을 때 좋은 성능을 발휘. 버스 사이클 당 하나의 연결만을 제공한다는 단점이 있다.
- 크로스바:  $P \times M$  크로스바(crossbar)는  $P$ 개의 프로세서와  $M$ 개의 기억장치 모듈과 동시에 연결할 수 있으며, 완전 크로스바는 스위치의 수가 프로세서의 수와 기억장치의 수의 곱으로 증가하기 때문에 네트워크의 규모가 커지면 비용이 매우 커진다는 단점이 있음
- 다단계(multistage) 네트워크: 근원지와 목적지 사이에 여러 개의 스위치 요소가 있는 상호연결망으로서 보통 많은 연결이 필요하고 프로세서의 수가 많을 때 이용됨

[정답] 2, 4