

강의	정보처리 필기	강사	조대호
차시 명	[CA-06강] 프로세서와 명령어	차시	6차시

학습내용

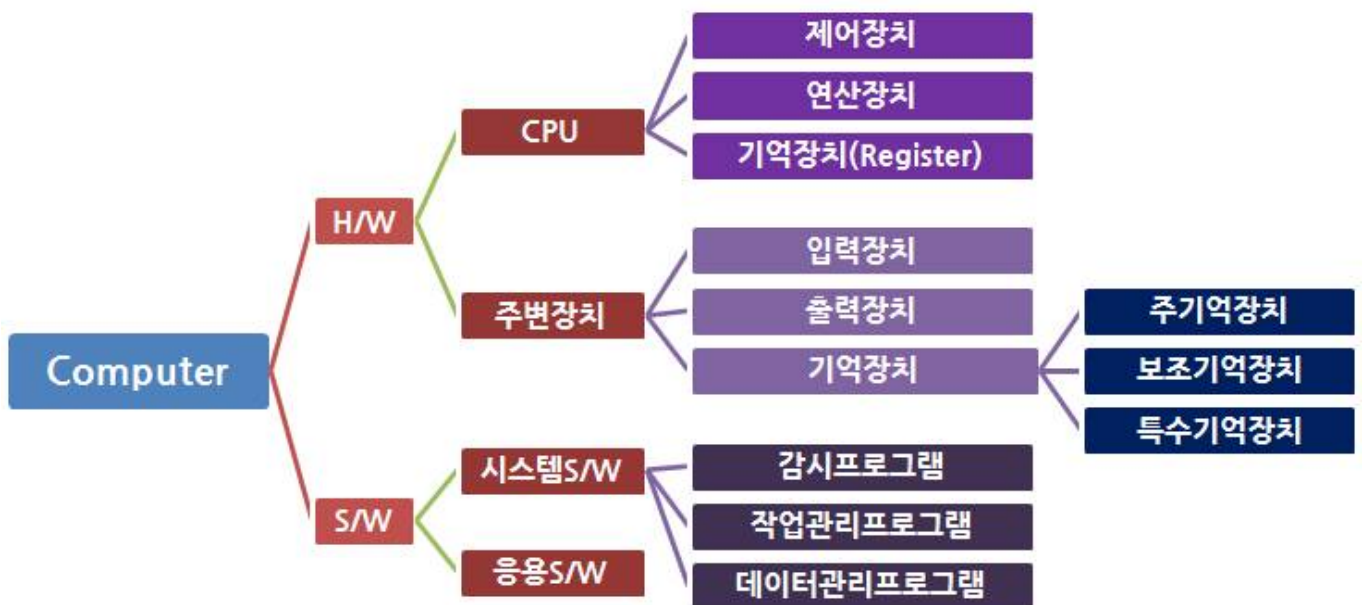
☞ 프로세서와 명령어

학습목표

- ☞ 컴퓨터의 구조와 프로세서에 대해 이해 할 수 있다
- ☞ 컴퓨터의 명령어에 대해 이해 할 수 있다

학습내용

1. 컴퓨터의 구성



2. 중앙처리장치(Central Processing Unit)



제어장치

주 기억장치에 기억된 명령을 꺼내서 해독하고, 시스템 전체에 지시 신호를 내는 장치
명령 코드가 명령을 수행할 수 있게 필요한 제어 기능을 제공

제어장치의 구성

- 프로그램 카운터, 명령어 레지스터, 부호기, 명령해독기, 순서기, 주소 처리기

연산장치

제어장치의 명령에 따라 실제로 연산을 수행하는 장치

산술연산, 논리연산, 관계연산, 이동(Shift)등을 수행

연산장치의 종류

- 가산기, 누산기, 보수기, 데이터 레지스터, 오버플로우 검출기, 시프트 레지스터

기억장치(Register)

중앙처리장치 내부에서 처리할 명령어나 연산의 중간 결과값 등을 일시적으로 기억하는 임시기억장소

버스(Bus)

CPU, 메모리, I/O 장치 등과 상호 필요한 정보를 교환하기 위해 연결하는 공동의 전송선
제어버스, 주소버스, 데이터버스(CPU와 메모리 또는 I/O장치 사이에서 데이터 전송)

3. 레지스터의 종류 및 기능

종 류	기 능
프로그램 카운터 (PC, Program Counter)	<ul style="list-style-type: none"> 차기 명령(Next Instruction)의 번지를 지시 (다음에 실행할 명령의 번지를 가짐) 프로그램 실행 도중 분기가 발생하면 CPU 내의 PC의 내용을 먼저 변화시켜야 함 (분기명령어가 실행되는 경우에는 그 목적지 주소로 갱신됨)
명령 레지스터 (IR, Instruction Register)	<ul style="list-style-type: none"> 현재 실행중인 명령의 내용을 기억 OP code 명령호출이 IR로 이동
누산기 (AC, Accumulator)	<ul style="list-style-type: none"> 연산장치에 있는 레지스터(register)의 하나로 연산 결과를 일시적으로 기억하는 장치 주소 부분이 하나밖에 없는 1-주소 명령 형식에서 결과 자료를 넣어 두는데 사용하는 레지스터
상태 레지스터 (Status Register)	<ul style="list-style-type: none"> 컴퓨터의 내부 상태를 나타냄
PSWR (Program Status Word Register)	<ul style="list-style-type: none"> PSW를 저장하고 있는 레지스터 ※ PSW : 시스템 내부의 순간순간의 상태를 기록하고 있는 정보
플래그 레지스터	<ul style="list-style-type: none"> CPU 내부에서 방금 행한 연산의 결과로 나타나는 상태(결과가 0인지 여부, 부호(음수인지 양수인지), 캐리 및 오버플로의 발생 여부 등의 상태)를 나타내는 플립플롭
메모리 주소 레지스터 (MAR, Memory Address Register)	<ul style="list-style-type: none"> 기억장치를 출입하는 데이터의 번지를 기억하는 레지스터
메모리 버퍼 레지스터 (MBR, Memory Buffer Register)	<ul style="list-style-type: none"> 기억장치를 출입하는 데이터가 잠시 기억되는 레지스터
인덱스 레지스터 (Index Register)	<ul style="list-style-type: none"> 어드레스의 수정, 서브루틴의 연결, 반복 계산 수행 등의 역할을 하는 레지스터
Shift Register	<ul style="list-style-type: none"> 자료의 병렬 전송을 직렬 전송으로 변환 2배 길이 레지스터(double-length register)라고도 불림
Major State Register	<ul style="list-style-type: none"> CPU가 무엇을 하고 있는지 나타냄

4. 명령어(Instruction) 의 구성

연산자(Operation Code)부	자료(Operand)부
----------------------	--------------

실행할 명령이 들어 있음

연산자부의 비트수가 n Bit 일 때 2의 n승 개의 명령어(연산자) 수행 가능

자료부 = 어드레스 필드 = 주소부

실제 데이터에 대한 정보를 표시하는 부분

어드레스 필드의 크기 = 최대 메모리 용량

예> OP-Code : 6bit, 어드레스 필드 16bit 일 때,
최대 메모리 용량은?

64K Word (2의 16승 = 65536 = 64K)

5. 연산자(OP-Code, Operation Code) 의 기능

연산자(Operation Code)부	자료(Operand)부
----------------------	--------------

함수 연산 기능

중앙처리장치에서 데이터를 처리하는 기능

산술, 논리 연산 명령(ADD, AND, CPA, CPC, CLC, ROR, ROL 등)

자료 전달 기능

중앙처리장치와 기억장치 사이에서 정보를 교환하는 기능

Load : 메모리의 내용을 CPU로 전달

Store : CPU의 정보를 메모리에 기억

Move : 특정 레지스터의 내용을 다른 레지스터로 옮기는 명령

Push, Pop : 스택에 자료를 저장, 인출하는 명령

제어 기능

프로그램의 수행 흐름을 제어하는데 사용

무조건 분기 명령 : GOTO, JMP(Jump)

조건 분기 명령 : IF, SPA, SNA, SZA

부 프로그램 호출 및 복귀 : Call, Return

입,출력 기능

6. 명령어 형식

3-주소 명령어

OP-Code	Operand 1	Operand 2	Operand 3
	자료1의 주소	자료2의 주소	결과의 주소

Operand 부가 3개로 구성

연산 후에 입력자료가 변하지 않고 보존됨

전체 명령어를 읽어오는 시간 단축

프로그램의 길이가 짧아짐

하나의 명령을 수행하기 위해서 최소한 4번 기억장소에 접근해야 하므로 전체적인 수행시간 길어짐

2-주소 명령어

OP-Code	Operand 1	Operand 2
	자료1의 주소	자료2의 주소
	결과의 주소	

Operand 부가 2개로 구성

3-주소 명령어에 비해 길이가 짧음

계산 결과를 시험할 필요가 있을 때 기억장치에서 기억 될 뿐만 아니라 CPU에도 남아 있어 CPU내에서 직접 시험이 가능하므로 시간이 절약됨.

전체 프로그램의 길이가 길어짐

1-주소 명령어

OP-Code	Operand 1
---------	-----------

자료1의 주소

Operand 부가 1개로 구성

반드시 누산기(Accumulator)가 필요한 주소지정방식

모든 명령은 누산기에 기억되어있는 자료를 사용함

예> $C = A + B$

LOAD A
ADD B
STORE C

0-주소 명령어

OP-Code

Operand 부가 없이 OP-code 부만으로 구성

모든 연산은 스택(Stack)에 있는 자료를 이용하여 수행

7. 기타 시험에 잘 나오는 것 요약정리

단항 연산자와 이항 연산자

- 단항연산자(Unary Operator) : 피 연산자가 1개만 필요
NOT, Complement, Shift, Rotate, MOVE 등

- 이항연산자(Binary Operator) : 피 연산자가 2개 필요
사칙연산, AND, OR, XOR, XNOR 등

명령어 설계시 고려 사항

연산자의 종류, 주소 지정 방식, 해당 컴퓨터 시스템 단어(Word)의 크기(비트수)

스택(Stack)

자료의 삽입(Push), 삭제(Pop)이 한쪽에서 가능한 자료구조, LIFO 구조

0주소 명령어 형식 뿐 아니라 부 프로그램(서브루틴) 호출 시 복귀 주소 저장, 인터럽트 발생시 복귀 주소 저장, 재귀(Recursive) 프로그램의 순서 제어, 역 polish 형 산술식의 처리 등에 쓰임

0-주소 명령형을 갖는 컴퓨터 구조의 원리로 스택 머신(Stack Machine)이라고도 함

재귀(Recursion) 프로그램

한 루틴이 자기를 다시 불러 실행하는 프로그램

이터레이션(Iteration)

해당 루틴이 정확한 결과를 산출할 때까지 해당 루틴에서 발생한 결과를 가지고 처음에 사용한 자료를 다시 수정하여 계산 작업을 반복적으로 수행하는 것

프로그램 디버깅(Debugging)

프로그램의 오류를 발견하고 그 원인을 밝히는 작업

- 트레이서(tracer) : 프로그램의 실행과정을 출력하는 프로그램
- 덤프(dump) : 주기억장치의 내용을 그대로 화면이나 프린터, 디스크 등에 출력하는 것

부 프로그램(서브루틴, Subroutine)과 매크로

부 프로그램 : 반복적으로 사용되는 코드를 필요할 때 호출하여 사용할 수 있도록 한 프로그램

매크로 : 반복적으로 사용되는 코드를 프로그램 내에 삽입하여 언제든지 호출하여 사용할 수 있도록 한 것
프로그래머가 어셈블리어언어로 프로그램을 작성할 때 반복되는 것을 효과적으로 하기 위함

공통점 : 여러 번 중복되는 부분을 별도로 작성하여 사용

데이터 처리 명령어 종류

산술 명령어 : ADD, SUB, MUL, DIV, 산술 Shift

논리 명령어 : NOT, AND, OR, XOR, 논리적 Shift, ROTATE, COMPLEMENT, CLEAR

연산자의 우선순위 : 산술연산자 → 관계연산자 → 논리연산자

요점정리

1. 컴퓨터의 구조와 프로세서에 대해 이해 할 수 있다
2. 컴퓨터의 명령어에 대해 이해 할 수 있다

다음차시예고

수고하셨습니다. 다음 7주차에서는 “[CA-7강] 주소지정방식, 명령실행과 제어”에 대해서 학습하도록 하겠습니다.