11주 3강

명령어 집합

명령어 집합의 개념

- 명령어 집합(instruction set)
  - 중앙처리장치가 수행할 동작을 정의하는 2진수 코드로 된 명령들의 집합
  - 기계 명령어(machine instruction)라고도 함.
- 명령어 집합은 중앙처리장치의 사용목적, 특성에 따라 결정된다.

명령어 집합 설계를 위해 결정되어야 할 사항들

- 연산 종류
  - 중앙처리장치가 수행할 연산들의 수와 종류 및 복잡도 등을 결정.
- 데이터 형태
  - 연산을 수행할 데이터들의 형태, 데이터의 길이(비트 수), 수의 표현 방식 등을 고려
- 명령어 형식
  - 명령어의 길이, 오퍼랜드 필드들의 수와 길이 등을 고려
- 주소지정 방식
  - 피 연산자의 주소를 지정하는 방식을 고려

명령어 집합의 특성

명령어는 연산 코드(Operation Code), 오퍼랜드(Operand)로 구성

- 연산 코드(Operation Code)
  - 수행될 연산을 지정하며 연산자라고도 한다.
  - 함수 연산 기능, 전달 기능, 제어 기능, 입출력 기능으로 분류
- 오퍼랜드(Operand)
  - 연산을 수행하는 데 필요한 데이터 혹은 데이터의 주소
  - CPU의 레지스터, 주기억장치/입출력장치 등에 저장된 데이터 또는 주소
  - 다음 명령어 주소(Next Instruction Address)

명령어 형식(instruction format)

- 여러 필드들로 구분되며, 각 필드는 일련의 비트 패턴에 의해 표현
- 명령어 형식
  - 필드들의 수와 배치 방식 및 각 필드의 비트 수에 의해서 명령어가 표현됨.

- 세 개의 필드들로 구성된 16비트 명령어
  - 한 개의 연산코드 와 두 개의 오퍼랜드

명령어 집합에서 연산의 종류

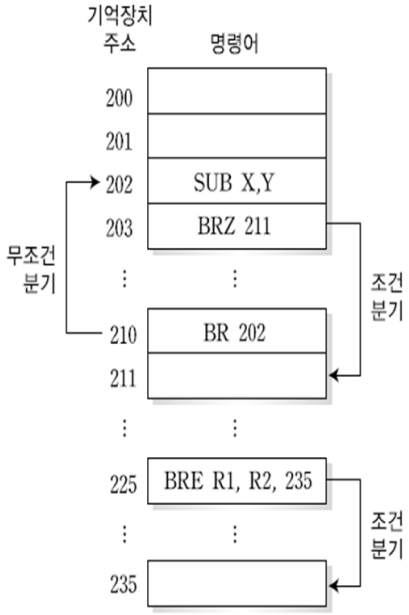
- 함수 연산 기능(Functional Operation)
  - 함수의 연산 기능을 담당하는 산술 연산이나 논리 연산 명령 등이 해당된다.
- 전달 기능(Transfer operation)
  - CPU와 주기억장치 사이, 레지스터 간의 정보교환과 적재, 저장기능을 수행.
  - 정확한 데이터 전송을 위해 근원지와 목적지 오퍼랜드의 위치가 명시되어야 함.
- 입출력 기능(Input/Output Operation)
  - 입출력 명령어는 중앙처리장치와 외부장치들 간의 데이터 이동을 위한 명령어.
  - 분리형 입출력의 경우는 별도의 입출력 명령어 사용
  - 기억장치-사상 입출력의 경우에는 일반 데이터 이동 명령어들을 동일하게 사용
- 제어 기능(Control Operation)
  - 제어장치에서 수행되며 프로그램의 수행흐름을 제어하는 데 사용.
  - 프로그램 내의 명령어의 실행 순서를 변경하는 명령어로 분기, 서브루틴 호출이 대표적이다

분기 명령어

- 오퍼랜드는 다음에 실행할 명령어 주소를 가지고 있다.
- 무조건 오퍼랜드의 주소로 이동하거나 조건 만족 시에만 이동
  - 조건 분기는 조건 코드(condition code)와의 부합 여부에 따라서 분기가 결정
  - 조건 코드에는 zero(0), 부호(+, -), 오버플로우 플래그 등이 있다

다양한 분기의 형태

- 203: BRZ(branch if zero) 211
  - 조건코드가 0이면 211로 분기
- 210: BR 202
  - 무조건 분기 명령어로 무조건 202번지로 분기하라는 명령
- 225: BRE(branch if equal) R1, R2, 235
  - 실행 중에 비교(검사)와 분기 처리를 수행
  - 레지스터 R1과 레지스터 R2의 내용이 같다면 235번지로 분기한다.



서브루틴 호출 명령어

- 호출 명령어(CALL)는 현재 PC 내용을 스택에 저장하고 서브루틴의 시작 주소로 분기하는 명령어이다.
- 복귀 명령어(RET)는 CPU가 원래 실행하던 프로그램으로 되돌아가도록 하는 명령어이다.

CALL 명령어에 대한 마이크로-연산

t_0 : MBR ← PC

t_1 : MAR ← SP, PC ← X

t_2 : M[MAR] ← MBR, SP ← SP - 1

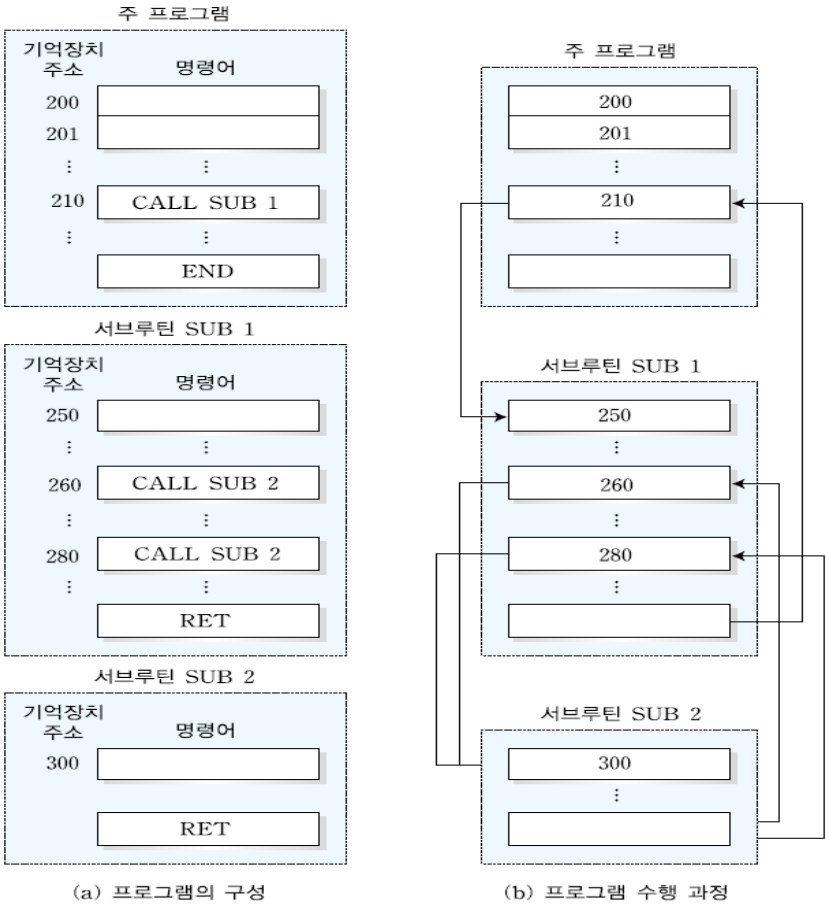
RET 명령어의 마이크로-연산

t_0 : SP ← SP + 1

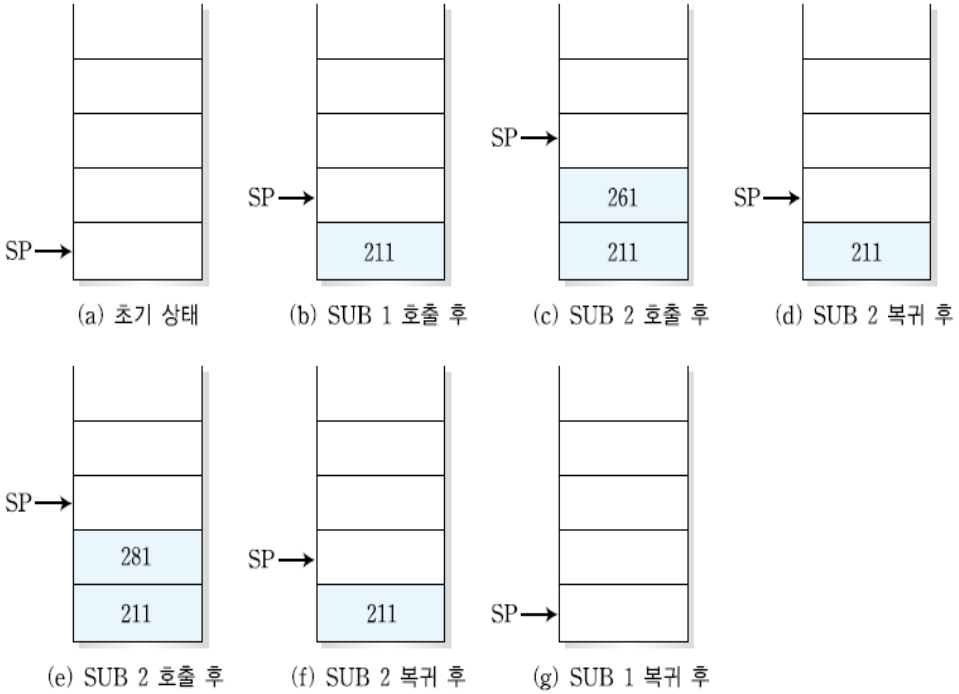
t_1 : MAR ← SP

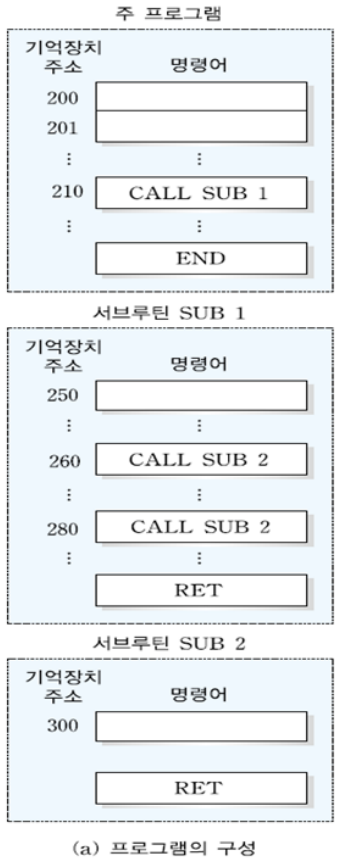
t_2 : PC ← M[MAR]

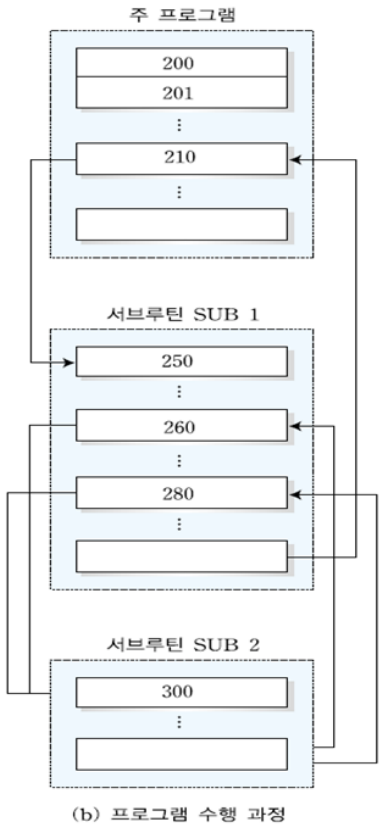
서브루틴의 호출과 복귀과정



서브루틴 수행 과정에서 스택의 변화







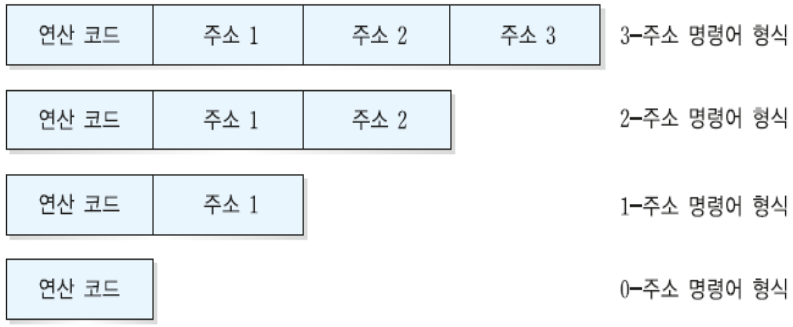
오퍼랜드 형태

- 오퍼랜드에 저장될 데이터 형태는 주소, 수, 문자, 논리 데이터 등이 될 수 있다.
  - 주소(addresses): 주기억장치의 주소이거나 레지스터의 주소다.
  - 수(numbers): 정수 혹은 고정-소수점 수, 부동-소수점 수등을 사용
  - 문자(characters): ASCII 코드가 사용된다.
  - 논리 데이터(logical data): 비트(bit) 혹은 플래그(flag) 등으로 사용된다.

- 오퍼랜드가 주소를 나타낼 때,
  - 오퍼랜드 수에 따라 3, 2, 1, 0 주소 방식이 있다.
  - 0 주소 방식은 스택을 사용하는 컴퓨터에서 사용된다.

오퍼랜드의 분류

- 주소 수에 따른 명령어 분류



- 주소 수에 따른 명령어 표현 (어셈블리어 표현)

명령어 분류	어셈블리 언어로 명령어 표현
1-주소 명령어	LOAD X
2-주소 명령어	MOV X, Y
3-주소 명령어	ADD X, Y, Z

*0-주소 명령어는 어셈블리어로 표현하지 않는다.

다음 시간

12주. 컴퓨터 명령어를 효과적으로 실행하기 위한 기법들