

## 11주 2강

# 명령어 사이클



# 어셈블리 프로그램의 분석

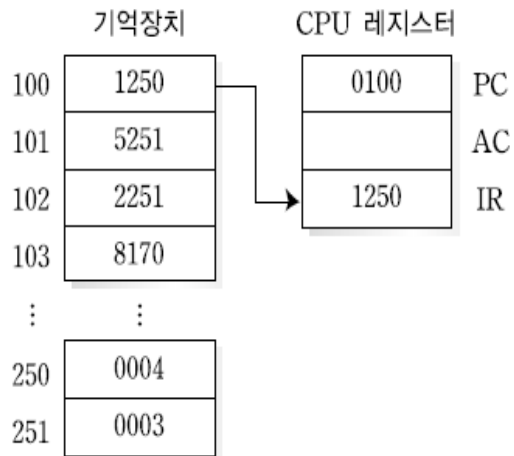


주소	명령어		기계 코드
100	LOAD	250	1250
101	ADD	251	5251
102	STOR	251	2251
103	JUMP	170	8170

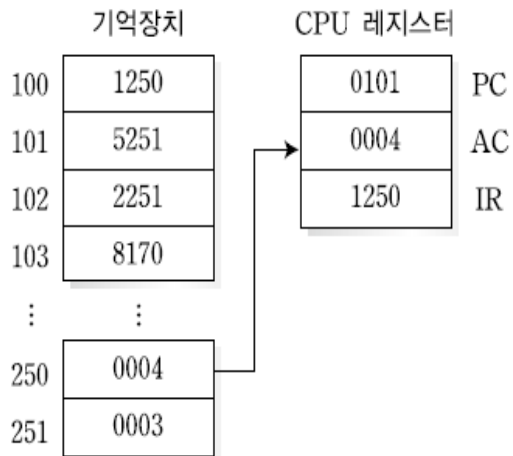
# LOAD 명령어의 동작



- 단계 1은 명령어 인출 단계이고, 단계 2는 명령어가 실행 단계



[단계 1]



[단계 2]

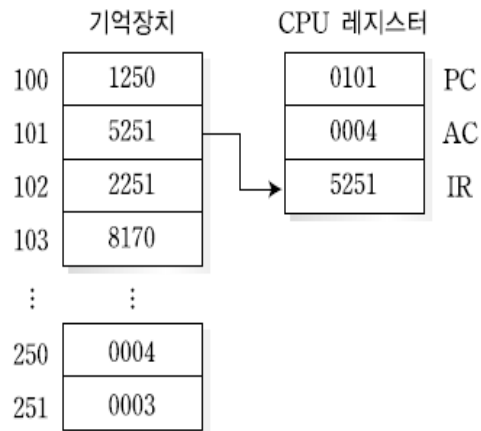
주소	명령어		기계 코드
100	LOAD	250	1250
101	ADD	251	5251
102	STOR	251	2251
103	JUMP	170	8170

# ADD 명령어의 동작

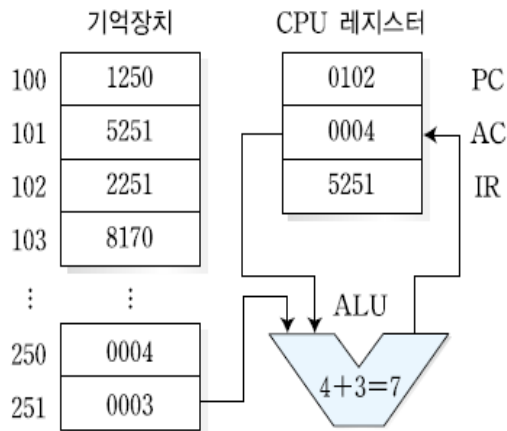


- PC에 저장되어 있는 데이터 값 101에 의해서 두 번째 명령어가 기억장치 101번지에서 인출되어 IR에 저장
- 이 명령어를 해독하고 실행해서, 누산기(AC)의 내용과 251번지의 내용을 더하고, 결과를 다시 누산기에 저장
- 그 다음 PC의 내용은 다음 명령어를 위해 102로 증가

주소	명령어		기계 코드
100	LOAD	250	1250
101	ADD	251	5251
102	STOR	251	2251
103	JUMP	170	8170



[단계 3]

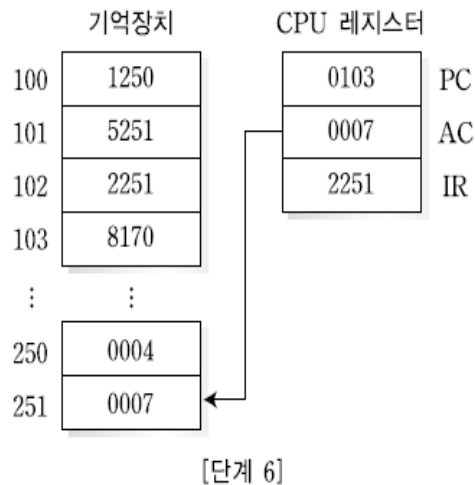
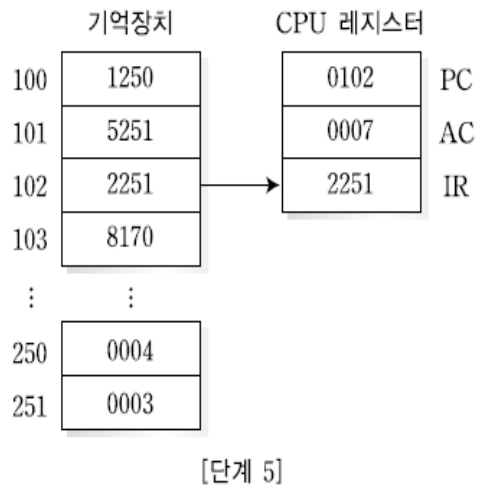


[단계 4]

# STORE 명령어의 동작



- 세 번째 명령어가 102번지로부터 인출되어 IR에 저장되고 해독
- 명령어 내용에 따라서 AC의 내용이 기억장치 251번지에 저장된다. 그리고 PC의 내용은 103으로 증가하게 된다.



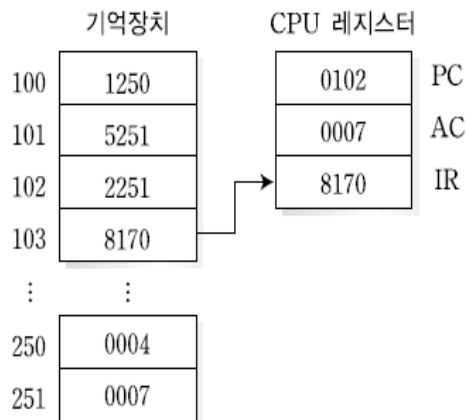
주소	명령어	기계 코드
100	LOAD 250	1250
101	ADD 251	5251
102	STOR 251	2251
103	JUMP 170	8170

# JUMP 명령어의 동작

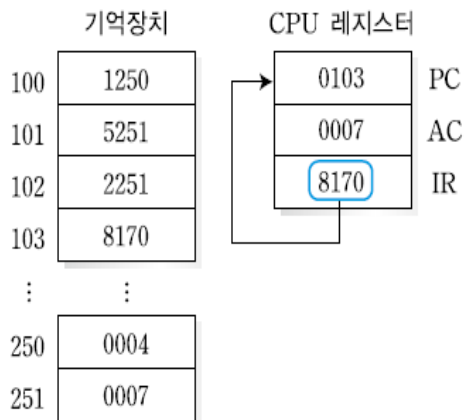


- 103번지에서 인출되어 IR에 저장된 후 명령어가 해독된다.
- 분기될 목적지 주소, 즉 IR의 하위 부분(170)이 PC로 적재된다.
- 다음 명령어 인출 사이클에서는 170 번지의 명령어가 인출이 되고 실행이 될 것이다.

주소	명령어	기계 코드
100	LOAD 250	1250
101	ADD 251	5251
102	STOR 251	2251
103	JUMP 170	8170



[단계 7]



[단계 8]

# 명령어 사이클의 개념

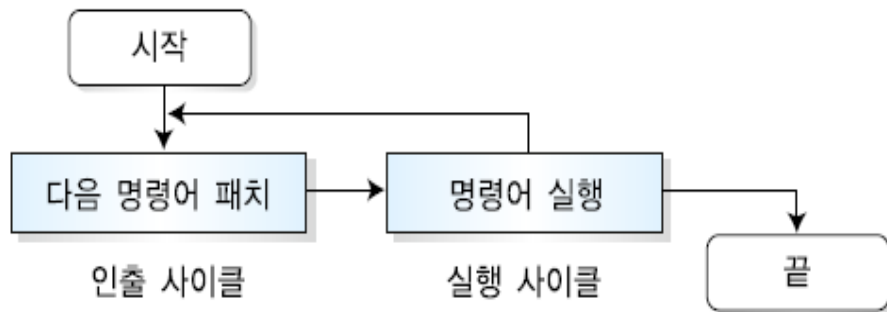


## ● 명령어 사이클(instruction cycle)

- 명령어는 두 단계를 하나의 사이클로서 해서 명령어를 수행
- 명령어 인출 단계는 인출 사이클(fetch cycle)
- 명령어 실행 단계는 실행 사이클(execute cycle)

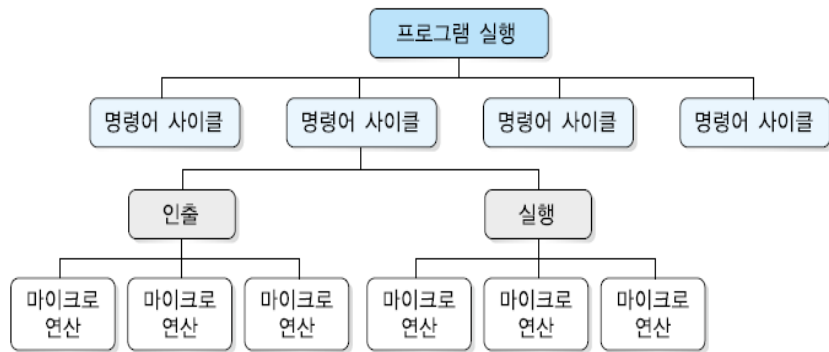


## 명령어 사이클의 동작과정





- ★ **마이크로-연산(Micro-Operations)**
  - 명령어를 실행하기 위한 가장 기본 단위의 프로그램 수행
  - 인출 사이클과 실행 사이클은 여러 단계(step)로 구성
    - 각 단계에서 수행되는 동작을 마이크로-연산(Micro-Operations)이라 한다.
  - 원자 연산(atomic operation)이라고도 한다



- ★ **프로그램이 실행되는 계층 구조**

- 명령어들은 명령어 사이클의 집합으로 구성
- 명령어의 사이클에는 두 개의 명령어 부 사이클이 존재
- 각 부 사이클의 수행도 여러 단계로 구분되며 이 구분이 마이크로 연산이다.



# 명령어 인출 사이클(Fetch Cycle)



## ● 중앙처리장치가 기억장치에서 명령어를 읽어오는 단계

- ① 프로그램 카운터(PC)는 다음에 인출할 명령어의 주소를 가지고 있다.
- ② CPU는 PC가 가리키는 기억장소에서 명령어를 인출하고 PC 내용을 증가
- ③ 명령어는 IR에 적재되고 CPU는 이를 해석하고, 요구된 동작을 수행한다.

## ● 명령어 인출 사이클에 대한 마이크로 연산

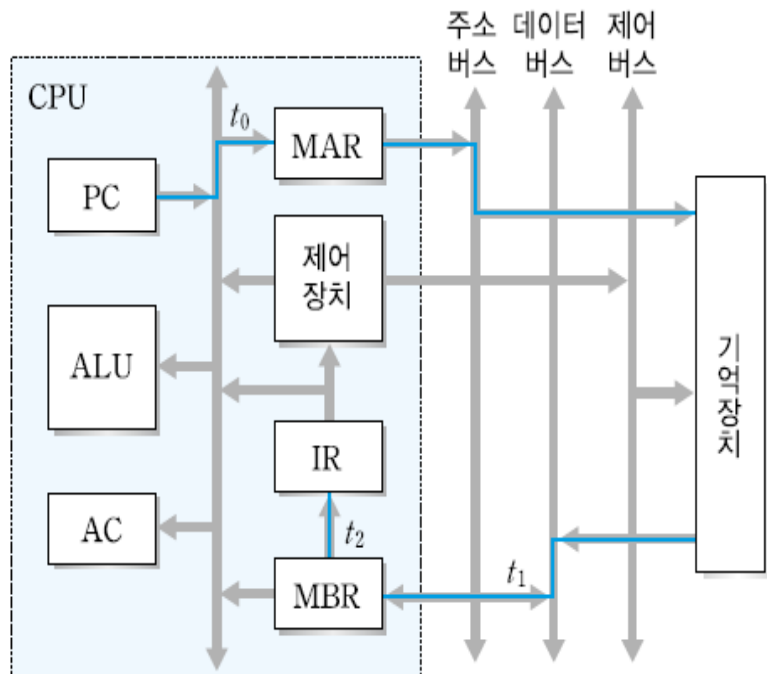
CPU 클럭	마이크로 연산	동작
$t_0$	$MAR \leftarrow PC$	PC 내용을 MAR로 전송
$t_1$	$MBR \leftarrow M(MAR)$ $PC \leftarrow PC + 1$	해당 주소 기억장치의 명령어가 MBR로 적재 PC의 내용을 1 증가
$t_2$	$IR \leftarrow MBR$	MBR에 있는 명령어가 IR로 이동

- 인출 사이클에서는  $t_0$ ,  $t_1$  그리고  $t_2$ 의 중앙처리장치 클럭의 주기가 필요
- 만약 중앙처리장치 클럭이 100MHz (클럭 주기 = 10ns)이면,  
인출 사이클 시간 :  $10\text{ns} \times 3 = 30\text{ns}$  소요

# 인출 사이클에서 주소와 명령어 흐름



- $t_0$ 에서는 주소가 기억장치로 전달
- $t_1, t_2$ 에서 기억장치에서 인출된 명령어가 MBR을 통해서 IR로 전달된다.



# 명령어 실행 사이클(Execution Cycle)



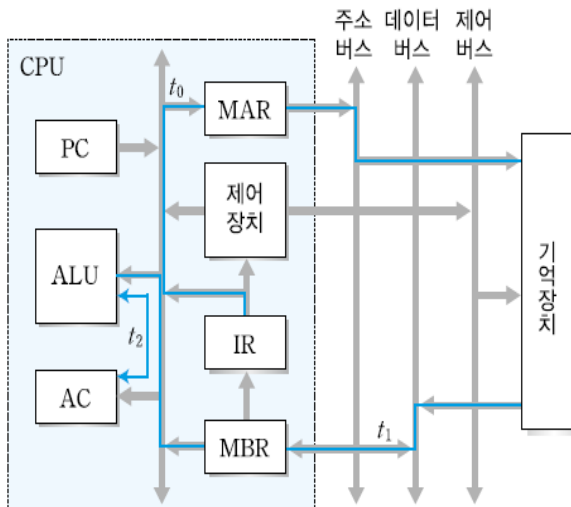
- 명령어를 실질적으로 실행하는 단계
  - ① 중앙처리장치와 기억장치 간에 데이터가 전송된다.
  - ② 중앙처리장치와 입출력 모듈 간에 데이터가 전송된다.
  - ③ 데이터에 대하여 지정된 산술 혹은 논리 연산이 수행된다.
  - ④ 제어동작으로 점프(jump)와 같이 실행될 명령어의 순서가 변경에 사용
- 실행 사이클에서의 마이크로-연산들은 명령어에 따라 다르다.
- ADD 명령어의 실행 사이클
  - 기억장치에 저장된 데이터를 AC의 내용과 더하고, 그 결과를 다시 AC에 저장하는 명령

CPU 클럭	마이크로 연산	동작
$t_0$	$MAR \leftarrow IR(addr)$	버퍼 레지스터 MBR에 저장될 데이터의 기억장치 주소를 MAR로 전송
$t_1$	$MBR \leftarrow M(MAR)$	저장할 데이터를 MBR로 이동
$t_2$	$AC \leftarrow AC + MBR$	MBR 데이터와 AC의 내용을 더하고 결과값을 다시 AC에 저장

# ADD 명령어 실행 사이클 동안의 정보 흐름



- $t_0$ 에서는 주소가 기억장치로 전달
- $t_1, t_2$ 에서는 명령어와 데이터의 흐름을 보여준다.



- 명령어 실행의 소요되는 시간: 명령어 인출 시간+ 명령어 실행 시간
- 하나의 명령어가 완전하게 실행되는 시간은 두 서브 사이클이 수행되는 시간의 합이다.

다음 시간

## 11주 3강. 명령어 집합

