

19강_제어 유닛의 제어방식과 기능 및 구조

🕒 Created	@Aug 13, 2020 12:04 PM
≡ Property	다시
≡ Tags	

학습목표

- » CPU 내의 제어유닛의 제어 방식의 종류와 명령어 코드를 해독하기 위한 기능 및 구조를 설명할 수 있다.
- » 마이크로코드 형식에서 필드 별 2진 비트 패턴으로 구성된 마이크로명령어의 집합체인 마이크로프로그램 작성 방법을 설명할 수 있다.
- » 마이크로명령어 해독 후 실행을 위한 순서 제어 방법을 이해하고, 또한 마이크로명령어 제어 형식을 설명할 수 있다.

학습내용

- » 제어 유닛의 제어방식과 기능 및 구조
- » 마이크로프로그램
- » 마이크로프로그램 실행 순서제어

제어 유닛

○ 하드와이어 제어(Hardwired Control)

- 복잡한 Datapath → 복잡한 제어 설계 → 복잡한 프로세서 설계
- 명령어 세트 설계의 단순화 → 제어 과정을 단순화
- 제어 유닛은 모든 명령어를 수행함에 있어서 매 클럭 주기마다 Datapath가 해야 할 일을 정한다.
- 이 과정을 특정화하여 표현한 것이 Finite-state diagram이다.
- 각 state마다 1 클럭 소요되므로 명령어 수행은 다수의 state로 완료된다.
- Complexity of Control
= State × Control inputs × Control output
- 입력: IR(function) + Datapath(condition) + state의 수
- 출력: X bit 패턴 구성

제어 유닛

○ 하드와이어 제어(Hardwired Control)

제어 유닛

$$S_0 : MAR \leftarrow PC$$

$$S_1 : MDR \leftarrow M[MAR], PC \leftarrow PC+1$$

$$S_2 : IR \leftarrow MDR$$

여기서, S_0, S_1 및 S_2 는 state

하드와이어 제어 / 마이크로 프로그램을 이용한 제어 두 가지로 구분 가능

Datapath가 복잡하기 때문에 제어가 복잡, 결과적으로 프로세서도 복잡하게 설계가 된다
이를 단순화 시키기 : Datapath를 명령어를 설계할 때부터 단순화 시켜서, 제어과정을 단순하게 만들자

→ 제어 유닛 : 모든 명령어를 수행할 때 매 클럭마다 Datapath가 해야 할 일을 정한다.(모든 명령어를 수행하는 단계 표준화를 만들어서 State를 일정하게 만든다)

Finite-state diagram : 이러한 과정들을 특정화해서 표현한 것. 각 state는 1 클럭씩 소요되게 만들어서 명령어가 수행될 때 다수의 state로 완성이 되게끔 단순화 시키는 것

Complexity of Control : State의 수 * Control inputs * Control output 으로 복잡도 따짐

제어 유닛

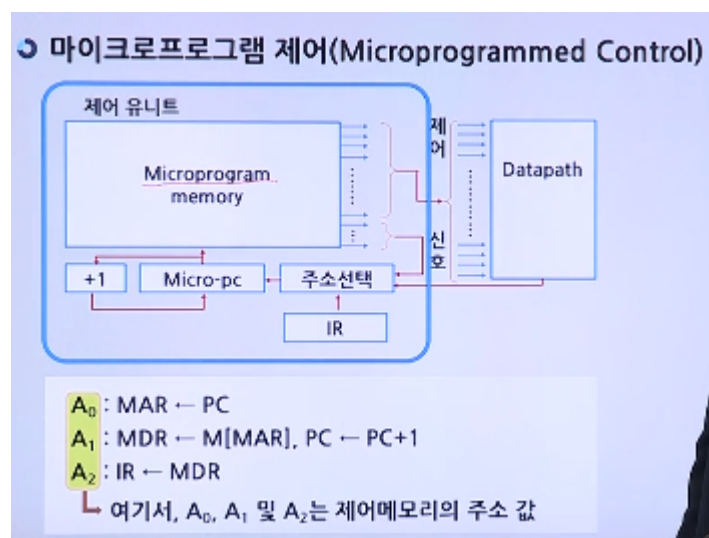
마이크로프로그램 제어(Microprogrammed Control)

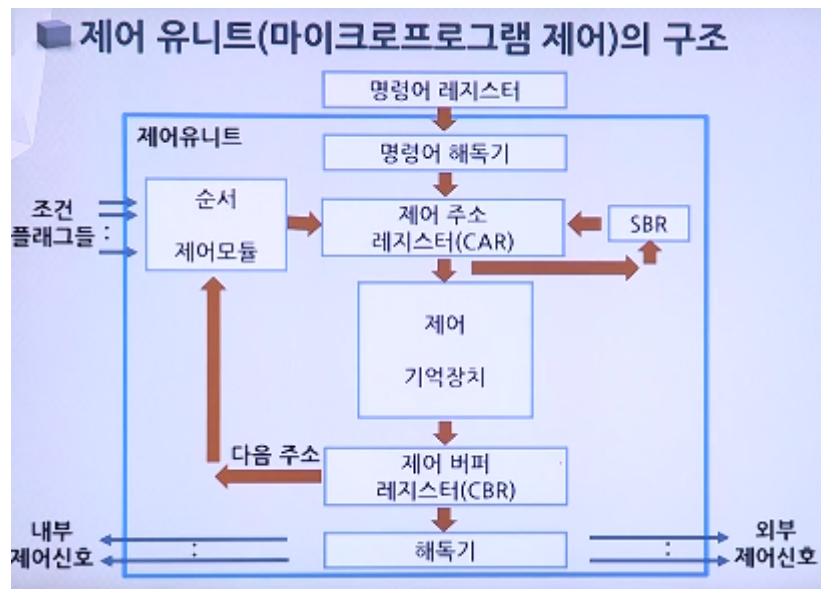
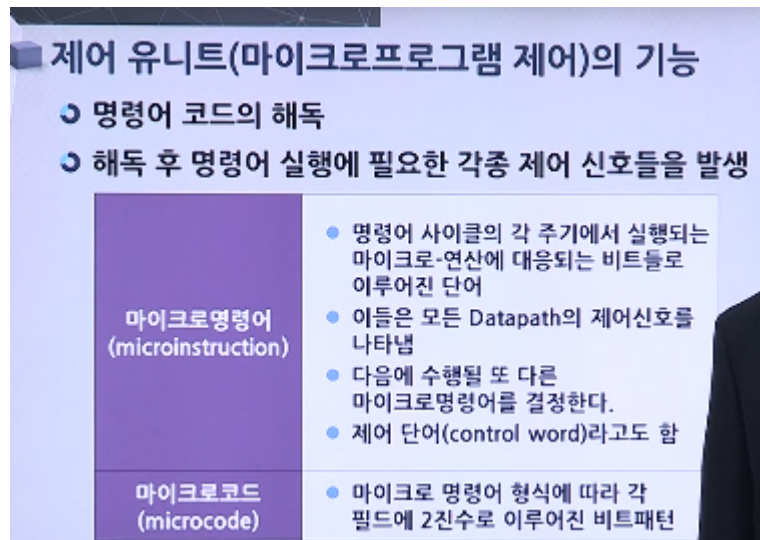
- 제어 유닛 → datapath 제어용 축소된 컴퓨터 (miniature computer)
- 프로그램 내장형 컴퓨터(stored-program computer)
- microprogramming → microinstruction, microcode, microprogram
- 제어신호 그룹 → 필드(비트 패턴) → microinstruction format
- 프로그래밍 → 주소체계 유지, 조건점프 등등
- 명령어에 해당되는 각각의 마이크로-연산을 제어 메모리(ROM)의 할당된 주소에 한 개씩 microcode(비트패턴)로 작성된다.

마이크로 프로그램 제어 == 프로그램 제어 회로로 제어하는 것이 아닌 비트패턴~프로그램 제어

CPU 내에 컨트롤 유닛이 있음. 그 자체가 하나의 조그만 컴퓨터다~

미니어처 컴퓨터(프로그램 내장형 컴퓨터)





마이크로프로그램 제어의 전체적인 구조 간략하게 보여줌
 명령어가 인출사이클을 통해서 최종적으로 가져온 위치가 명령어 레지스터
 CAR은 일종의 PC(프로그램 카운터)
 SBR

제어 유닛(마이크로프로그램 제어)의 기능

명령어 해독기 (Instruction Decoder)	<ul style="list-style-type: none"> 명령어 레지스터(IR)로부터 읽혀진 op-code를 해독한다. 해독된 연산을 실행하기 위한 마이크로서브루틴의 시작 주소를 결정한다.
제어 주소 레지스터 (Control Address Register)	<ul style="list-style-type: none"> 다음 번에 실행할 마이크로명령어의 주소 정보가 저장되는 레지스터이다. 주소는 제어 기억장치의 위치를 말한다.
제어 기억장치 (Control Memory)	<ul style="list-style-type: none"> 마이크로명령어들로 이루어진 마이크로프로그램을 저장하는 내부 기억장치이다.

제어 유닛(마이크로프로그램 제어)의 기능

제어 버퍼 레지스터 (Control Buffer Register)	<ul style="list-style-type: none"> 제어 기억장치로부터 읽혀진 마이크로명령어를 일시적으로 저장하는 레지스터이다.
서브루틴 레지스터 (Subroutine Register)	<ul style="list-style-type: none"> 마이크로프로그램에서 서브루틴이 호출되는 경우에 현재의 제어 주소 레지스터(CAR) 내용을 일시적으로 저장하는 레지스터이다.
순서제어 모듈 (Sequencing Module)	<ul style="list-style-type: none"> 마이크로명령어의 실행 순서를 결정하는 회로들의 집합이다.