

13주 3강

마이크로 프로그램을 이용한 제어



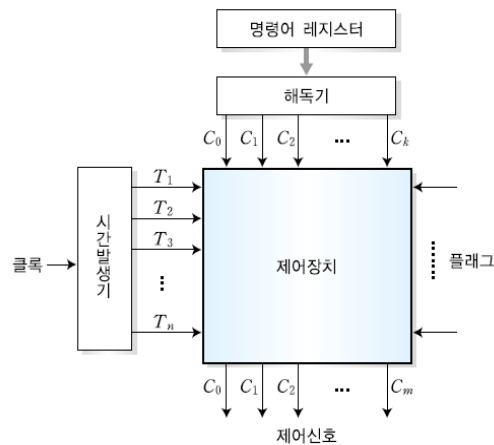
하드 와이어드 구현(hard wired implementation)



- 제어 신호 생성을 위해서 순차와 조합논리의 설계를 통해서 구성

- 하드 와이어드의 구조의 동작

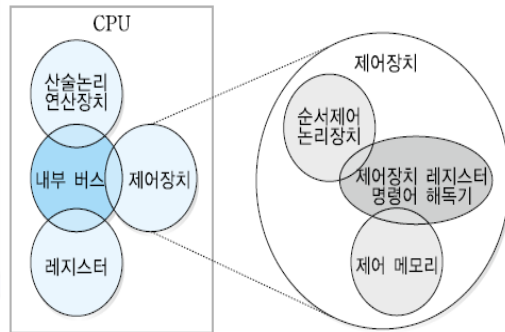
- 명령어 레지스터: 연산 코드를 해독하여 각 명령들이 여러 가지 연산을 수행
- 해독기: 각 연산 코드를 고유의 논리 입력을 가지도록 변환해주는 역할
- 클록: 마이크로 연산의 주기를 측정하는데 사용
- 클록 펄스의 주기는 신호가 데이터 경로와 중앙처리장치 회로를 통과하는 데 걸리는 시간만큼 길어야 한다.



마이크로 프로그램을 이용한 제어



- 하드 와이어드 제어 방식은 하드웨어의 설계와 검사가 복잡하며, 새로운 명령을 추가하기 위하여 설계를 변경해야 한다.
- 오늘날 제어장치의 구현은 마이크로 프로그램을 이용
- 마이크로 프로그램을 이용하는 제어장치의 구성
 - 순서제어 논리장치(sequencing logic)
 - 제어장치 레지스터들 (control unit registers)
 - 명령어 해독기(decoder)
 - 제어 기억장치(control memory)로 구성
- 제어장치 레지스터(control unit registers)
 - 제어 주소 레지스터(control address register)
 - 제어 버퍼 레지스터(control buffer register)
 - 서브루틴 레지스터(subroutine register) 등으로 구성

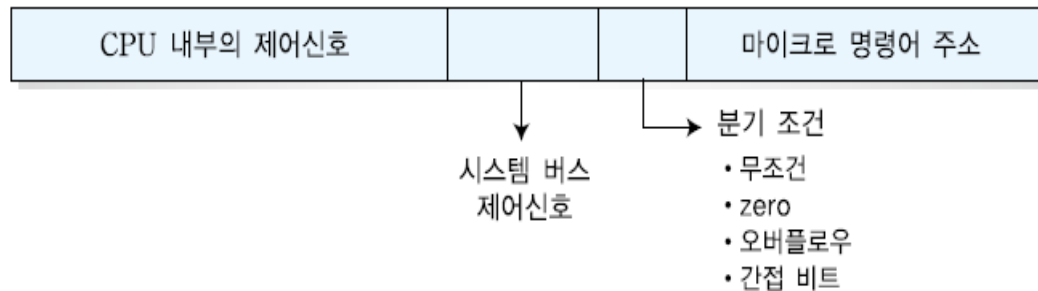


마이크로 명령어(micro instruction)



- 마이크로 프로그래밍 언어(micro-programming language)
 - 기계어의 마이크로 연산은 이진 비트로 표현, 사용의 편리성을 위해서 기호들(symbols)을 이용해서 표현한 것
- 마이크로 명령어(micro instruction)
 - 마이크로 연산 집합
 - 명령어 사이클의 각 부사이클은 여러 단계의 마이크로 연산들로 표현
 - 명령어 사이클은 마이크로 명령어들로 표현
- 마이크로 프로그램(micro-program)
 - 마이크로 명령어들을 이용하여 작성된 프로그램
 - 펌웨어(firmware)라고도 불린다.
- 루틴(routine)
 - 마이크로 프로그램에서 중앙처리장치의 특정 기능을 수행하기 위한 마이크로 명령어들의 그룹

●수평적 마이크로 명령어와 수직적 마이크로 명령어



(a) 수평적 마이크로 명령어

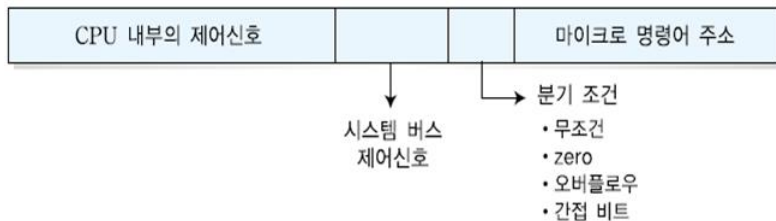


(b) 수직적 마이크로 명령어

수평적 마이크로 명령어



- 마이크로 명령어의 각 필드가 각 제어 신호에 대응되는 방식
 - CPU 내부의 제어 신호들과 시스템 버스의 제어 신호들에 필드가 할당.
 - 분기에서 사용될 조건을 나타내는 조건 필드와 분기 발생시 다음에 실행될 마이크로 명령어의 주소를 가지고 있는 필드가 있다.



- 수평 마이크로 명령어의 동작
 - 마이크로 명령어 실행을 위하여 비트 값이 1로 표시된 모든 제어 선을 ON, 비트 값이 0인 모든 제어 선을 OFF 시킨다.
 - 이 결과로 발생하는 제어 신호들은 한 개 이상의 마이크로 연산을 수행.
 - 분기 조건 비트의 값이 만족되지 않으면, 마이크로 명령어가 순서대로 실행된다.
 - 조건 비트의 값이 만족되면, 다음에 실행은 주소 필드에 의하여 지정된다.
- 수평 마이크로 명령어의 장단점
 - 하드웨어가 간단하고, 해독에 따른 지연 시간이 없다는 장점을 갖는다.
 - 마이크로 명령어의 비트 수가 길어서 큰 용량의 제어 기억장치가 필요하다.

- 코드화된 비트들을 이용하여 마이크로 명령어의 각 기능 코드를 구성

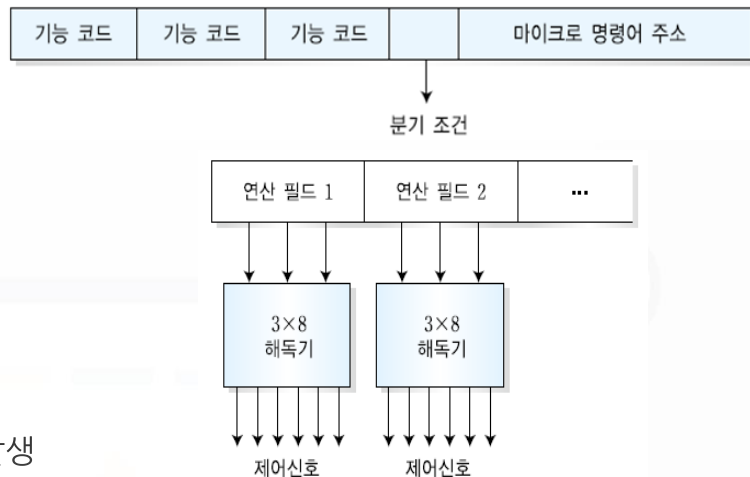
- 연산 필드에 적은 비트수의 코드화된 기능들을 포함시켜 제어 기억장치의 용량을 감소
- 해독기를 이용하여 코드를 필요한 수만큼의 제어 신호들로 확장하는 방식이다.

● 특징

- 마이크로 명령어의 비트 수가 감소되는 장점
- 제어 신호 발생을 위하여 코드화된 비트들을 해독하기 위한 지연이 발생

● 수직적 마이크로 명령어의 제어 신호 발생 방법

- 각 연산 필드의 내용은 코드화된 비트이고 해독기에서 그 내용이 해독되고 제어 신호들을 발생
- 3 X 8 해독기는 코드화된 3비트를 입력으로 받아들여서 8비트의 제어 신호가 되도록 코드를 해독



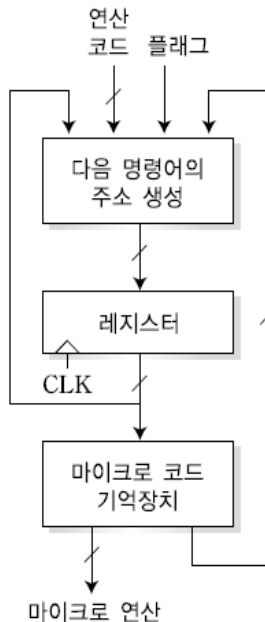
마이크로 프로그램을 이용한 구현



- 마이크로 명령어를 제어 기억장치에 저장하고 이것을 실행시켜서 제어 신호를 발생하는 방법이다.
- 마이크로 프로그램을 이용하여 구현된 대표적인 제어장치로는 마이크로 순서기(Micro-sequencing)가 있다.

● 마이크로 순서기의 구조

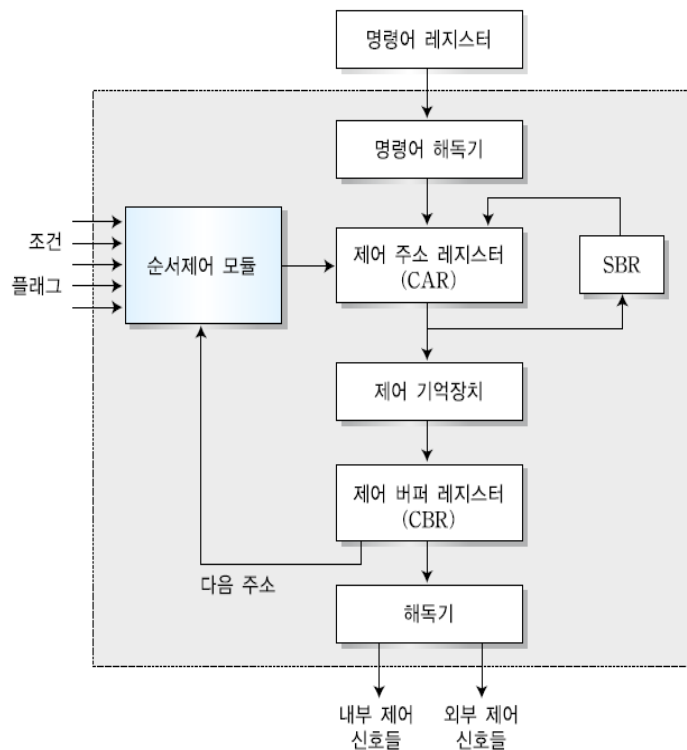
- 명령어 레지스터의 연산코드, 플래그, 마이크로 코드 기억장치의 출력이 다음 명령의 주소를 결정한다.
- 얻어진 마이크로 코드 기억장치의 주소는 레지스터에 임시로 저장되었다가 다시 이 주소에 근거하여, 마이크로 코드가 실행이 된다.
- 이 실행 결과의 출력이 제어 신호가 된다.



제어장치의 구조



- 명령어 해독기, 제어주소 레지스터, 제어 기억장치, 제어버퍼 레지스터, 서브루틴 레지스터, 순서제어 모듈로 구성



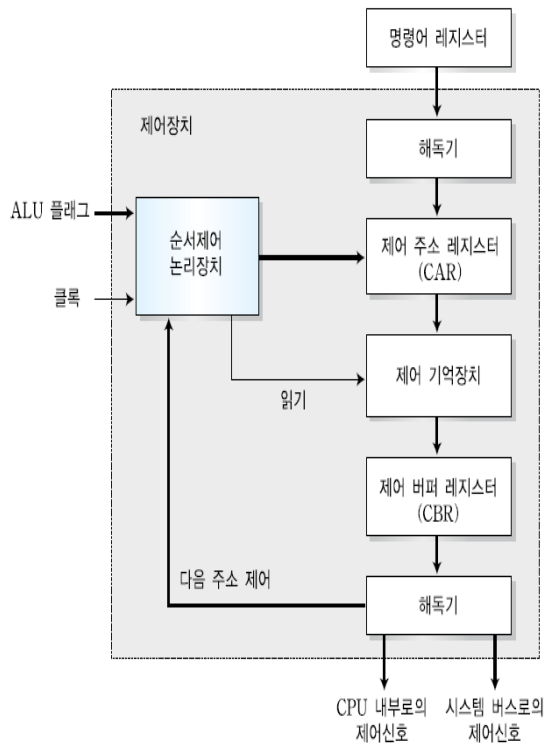


- 명령어 해독기(instruction decoder)
 - 명령어 레지스터(IR)가 보낸 명령어의 연산 코드를 해독하여 해당 연산을 수행하기 위한 루틴의 시작 주소를 결정하는 역할을 한다.
- 제어 주소 레지스터(CAR, Control Address Register)
 - 다음에 실행할 마이크로 명령어의 주소를 저장하는 레지스터다.
 - 저장된 주소는 제어 기억장치의 특정 위치를 지칭한다.
- 제어 기억장치(control memory)
 - 마이크로 명령어들로 이루어진 마이크로 프로그램을 저장하는 내부 기억장치다.
- 제어 버퍼 레지스터(CBR, Control Buffer Register)
 - 제어기억장치에서 읽은 마이크로 명령어 비트들을 일시적으로 저장.
- 서브루틴 레지스터(SBR, Subroutine Register)
 - 마이크로 프로그램에서 서브루틴이 호출되는 경우에 현재의 CAR 내용을 일시적으로 저장하는 레지스터다.
- 순서제어 모듈(sequencing module)
 - 마이크로 명령어의 실행 순서를 결정하는 회로들의 집합이다.

마이크로 프로그램을 이용한 제어장치의 동작과정



- 순서제어 논리장치(sequencing logic) 또는 순서제어 모듈이 제어 기억장치로 READ 명령어를 보낸다.
- 제어 주소 레지스터 CAR에 명시된 제어 기억장치 주소에 저장되어 있는 단어가 읽혀져 제어 버퍼 레지스터 CBR로 옮겨진다.
- 제어 버퍼 레지스터 CBR의 내용에 따라 제어 신호들과 다음 주소(next address) 정보가 발생된다.
- 순서제어 논리장치 또는 순서제어 모듈은 CBR의 내용과 ALU 플래그들에 근거하여 새로운 주소를 CAR에 적재한다. 그리고 다시 위의 과정을 반복하게 된다



다음 시간

14주. 병렬 컴퓨터 구조

