

5주 2강

CPU의 조직과 CPU 논리회로



1 명령어를 해독하는 기능과 제어 신호를 해당장치에 전달하는 역할을 수행

2 명령어의 형식

연산 코드

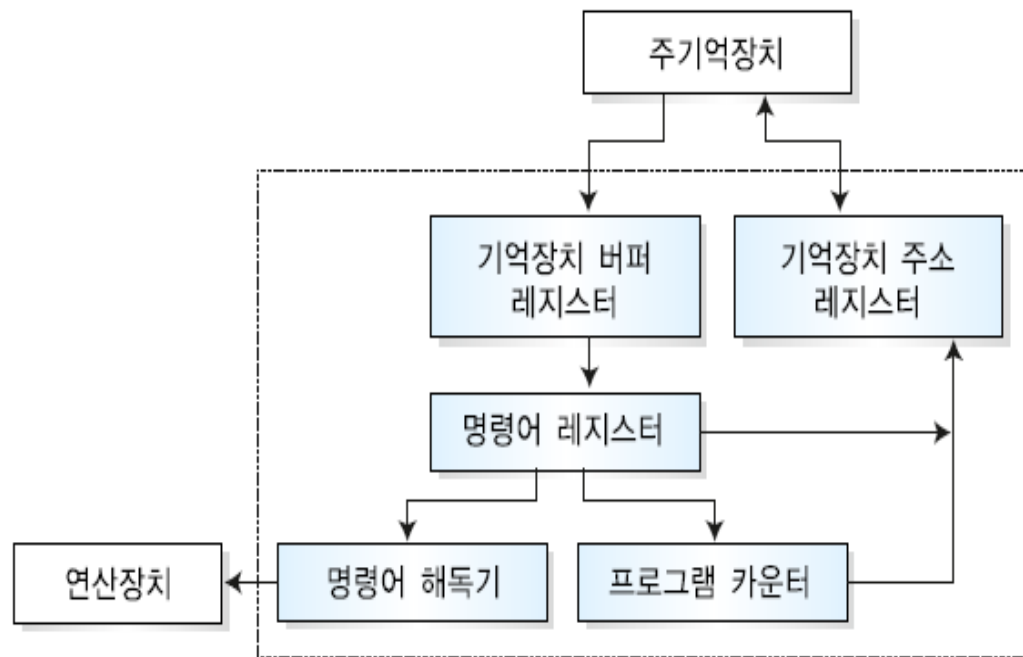
기억장치 주소

- 연산 코드필드는 수행되어야 할 연산이 지정되어 있는 필드
- 기억장치의 주소 필드는 해당 연산을 수행할 때 데이터가 저장되어 있는 주소

3 제어장치의 구성요소

- 기억장치 버퍼 레지스터 : 주기억장치로부터 읽어온 명령어를 임시적으로 저장
- 명령어 레지스터 : 명령어를 저장하는 곳
- 명령어 해독기 : IR에 저장된 명령어의 연산 코드 필드를 전달받아서 명령어를 해독하여 수행할 연산을 결정한다.
- 기억장치주소 레지스터 : 명령어 레지스터에 저장된 명령어의 주소 번지를 저장한다.
- 프로그램 카운터 : 다음에 수행할 명령어의 주소 번지를 저장하는 곳

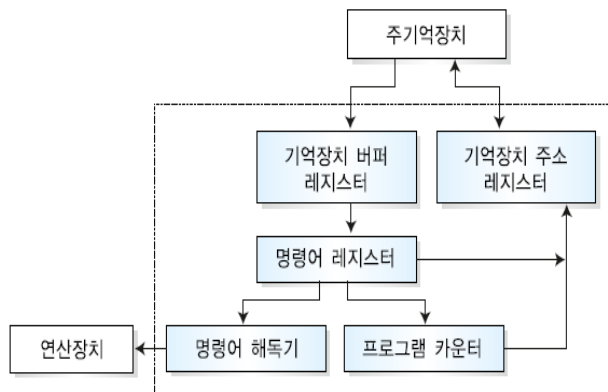
제어장치의 구성



내부 CPU 버스(Internal CPU Bus)



- 이동경로제공
 - ALU와 레지스터들 간의 데이터 이동
 - ALU와 제어장치 간의 데이터 이동
 - 제어장치와 레지스터들 간의 데이터 이동
- 데이터 버스와 제어버스로 구성
- 반드시 버퍼 레지스터들 시스템 버스 인터페이스 회로를 통하여 시스템 버스와 접속하는 특징을 가지고 있다.
- 기억장치 버퍼 레지스터와 기억장치 주소 레지스터는 CPU내부와 외부 장치 간에 속도 차이를 극복하기 위한 버퍼 역할



CPU의 논리 회로 설계

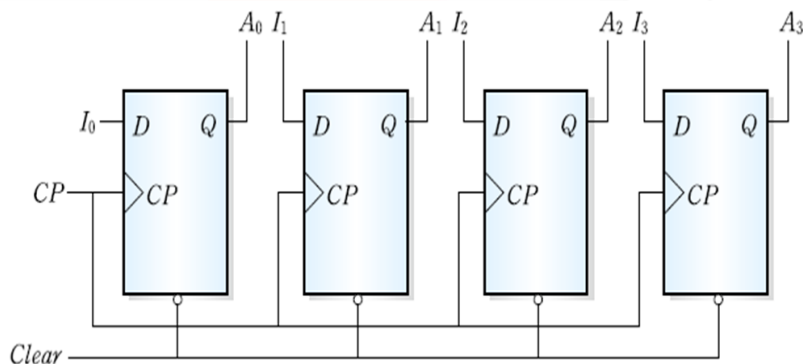


★ 마이크로프로세서는 논리 회로로 설계된 중앙처리장치를 하나의 집적 회로 칩으로 만든 것

- 논리 회로를 통해서 설계된 중앙처리장치의 구성요소들을 공부하는 것이 마이크로프로세서를 이해하는 것.

★ 레지스터

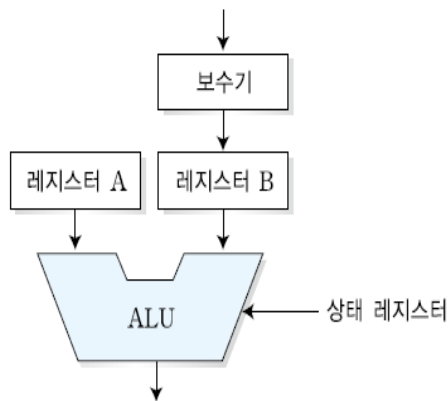
- 고속으로 동작할 수 있도록 플립플롭으로 구성
- 4비트 레지스터의 구성



1 연산될 데이터와 연산한 결과를 기억시킬 레지스터가 필요 또한 연산의 상태를 나타내기 위한 상태 레지스터들의 연결도 필요

2 산술논리연산장치(ALU)

- 덧셈을 하기 위한 가산기
- 연산에 이용되는 데이터나 연산 결과 등을 일시적으로 보관하기 위한 누산기
- 데이터를 보관하는 기억장치 버퍼 레지스터 등이 필요
- 이 밖에도 보수를 만들기 위한 보수기, 계산 결과의 상태를 점검하기 위한 상태 레지스터 등으로 구성

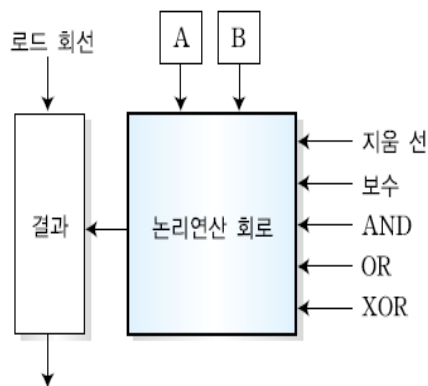




1 논리 게이트들을 이용하여 조합논리 회로를 구성하면 다양한 논리연산을 수행할 수 있다.

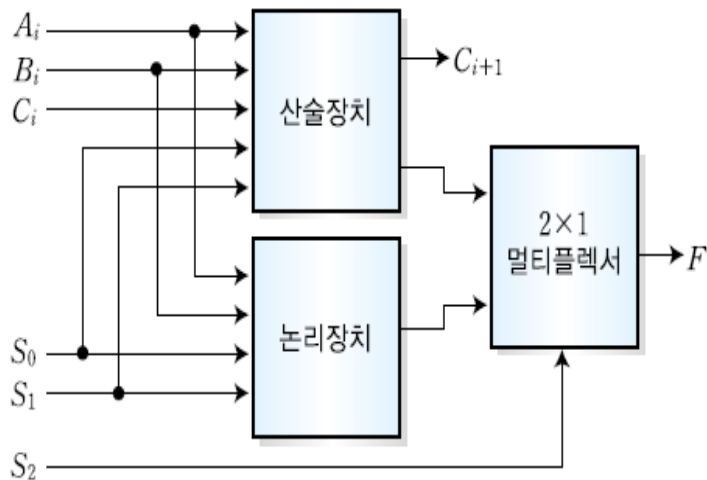
- 데이터 A와 B를 입력하는 회선과 연산의 종류를 선택하는 회선이 존재
- 입력되는 데이터 A와 B의 내용을 연산 지시에 의해 AND, OR, XOR 및 NOT 연산을 수행
- 연산 지시 신호는 논리연산의 선택뿐만 아니라 결과가 기억될 장소를 0으로 지우는 지움선과 NOT 연산으로 1의 보수가 되도록 하는 보수 회선이 존재할 수 있다.

2 논리연산 회로의 구성도



☆ 설계된 산술연산 회로와 논리연산 회로를 조합하면 ALU가 완성

- 선택선 S_1 과 S_0 는 산술연산 회로와 논리연산 회로가 공통으로 사용
- 선택선 S_2 는 두 회로 중 하나를 선택하는데 사용
- S_2 가 0이면 산술연산을, S_2 가 1이면 논리연산을 수행한다.

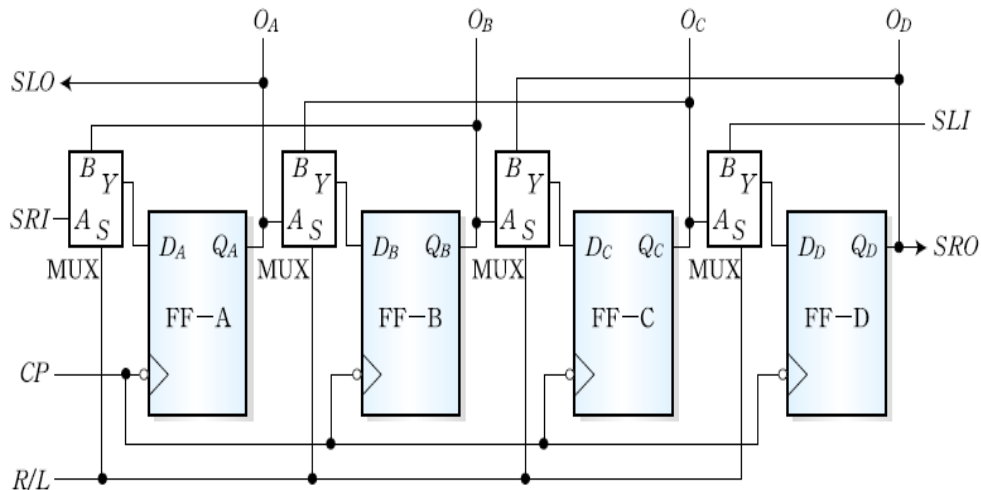


이동기(shifter)



★ 이동 방향은 왼쪽과 오른쪽이고, 이동 방향의 마지막 비트 값은 바깥으로 밀려난다. 이동 레지스터라고도 한다.

★ 양방향 이동 레지스터

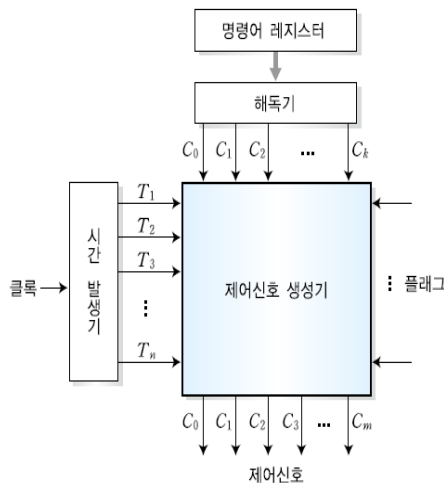


1 제어장치의 유형

- 하드웨어만으로 설계된 하드와이어적 제어장치(hardwired control unit)
- 소프트웨어가 포함된 마이크로 프로그램된 제어장치(micro-programmed control unit)

2 논리 회로에 의해 제작된 하드와이어적 제어장치

- 하드와이어적 제어장치는 제어 신호의 생성과정에서 지연이 매우 작다는 장점을 갖는다.
- 구현 논리 회로는 명령 코드 및 주소 지정 모드 등에 따라 매우 복잡하다는 단점을 갖는다.



다음 시간

5주 3강. CPU의 기능과 동작 그리고 성능

