6주차 2차시 CPU의 논리회로

【학습목표】

1. CPU의 논리회로 설계를 설명할 수 있다.
2. 연성장치의 개념을 설명할 수 있으며, 논리연산회로의 내부 구성과 함수 테이블을 설명할 수 있다.

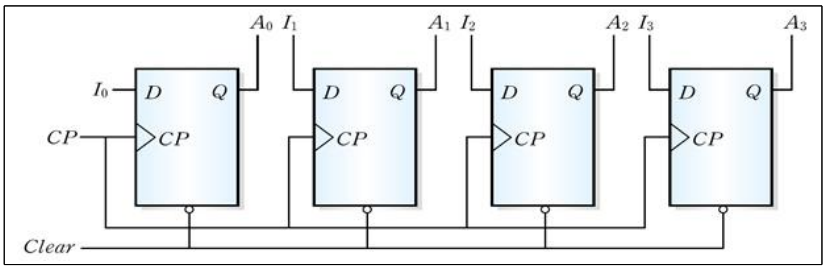
학습내용1 : 레지스터

1. CPU의 논리회로 설계

<CPU는 논리 회로로 설계된 중앙처리장치를 하나의 집적 회로 칩으로 만든 것>

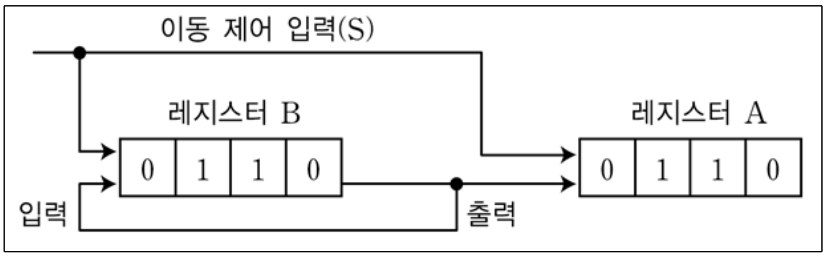
1) 레지스터
- 고속으로 동작할 수 있도록 플립플롭으로 구성
- 레지스터의 설계
    - 일반적으로 D 플립플롭은 레지스터를 제작하는 구성 요소로 사용됨
    - 입력신호 D가 클록 펄스에 동기 되어 그대로 출력에 전달되는 특성이 있음

2) 4비트 레지스터의 구성
- 각 플립플롭은 공통의 클록을 갖고 있음
- 클록이 플립플롭에 입력될 때 마다, 4비트의 입력 I_0 ~ I_3 가 저장
- 출력 측 A_0 ~ A_3 에서는 언제나 저장된 값을 참조할 수 있음



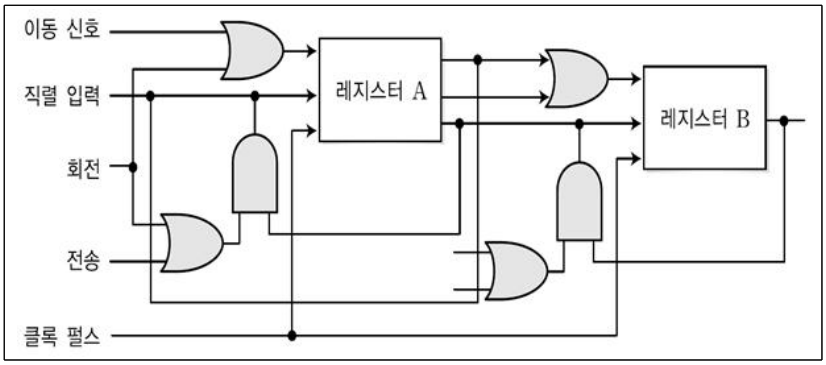
3) 레지스터의 전송

- 레지스터 간 전송
  - 다른 레지스터에 데이터를 쓰거나 저장된 데이터를 읽는 동작으로 직렬전송과 병렬전송으로 구분됨
- 직렬전송 방식
  - 이동 레지스터(shift register)임
  - 레지스터 B에서 레지스터 A로 데이터가 직렬 전송되는 개념임
  - 전송하는 레지스터의 내용을 보존하기 위해서는 자신의 직렬출력을 다시 직렬로 입력하여 모든 비트가 원래의 위치에 있도록 하여야 함



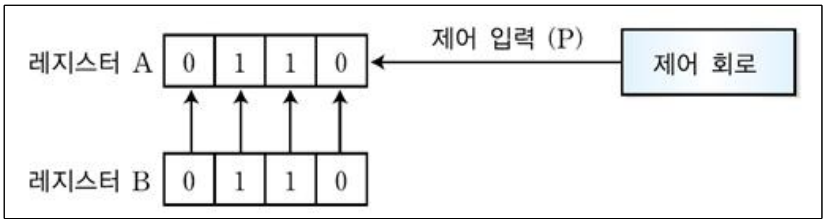
4) 게이트가 추가된 레지스터 B에서 레지스터 A로의 직렬전송

- 레지스터 A의 내용을 잃어버리지 않도록 하기 위해 게이트를 추가한 회로임
- 이동 신호는 레지스터 A의 내용만 이동 시키고 레지스터 B에는 영향을 주지 않는 단자임
- 전송 신호는 레지스터 A를 회전시키고 레지스터 B만 시프트 시키는 단자임
  - A의 데이터가 그대로 남아 있으면서 B로 이동 복사됨
- 회전 신호는 레지스터 A의 내용이 직렬로 출력된 후 다시 직렬로 입력되어 회전되지만 레지스터 B에는 아무런 영향을 미치지 않도록 설계된 단자임

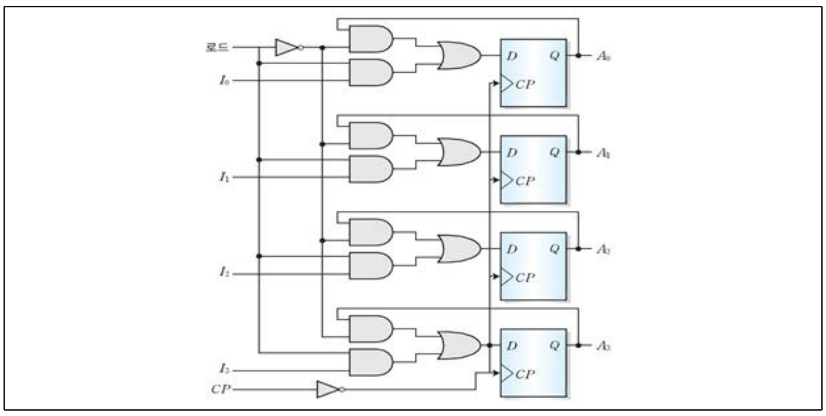


5) 레지스터 간 전송

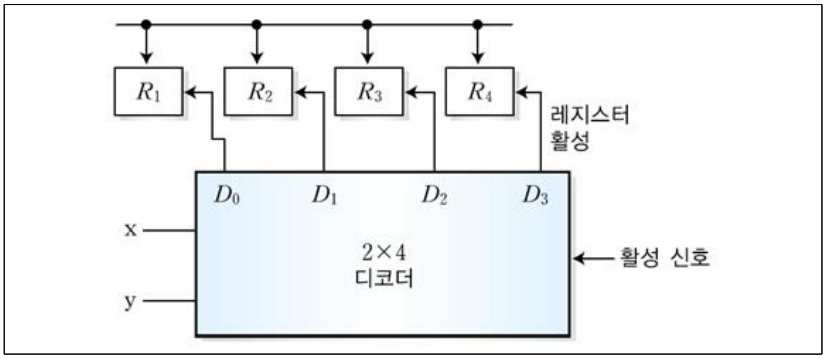
- 병렬전송 방식(Parallel Transfer)
  - 레지스터에 기억된 전체 내용을 하나의 제어 신호로 다른 레지스터에 동시에 전송하는 방식임



- 4비트 레지스터의 병렬전송
  - 로드(Load)가 1일 경우 4비트 입력 I_0 ~ I_3 은 4개의 플립플롭에 각각 저장됨
  - 로드(Load)가 0인 경우 I_0 ~ I_3 의 입력은 차단되고, 플립플롭의 결과가 다시 플립플롭으로 입력됨

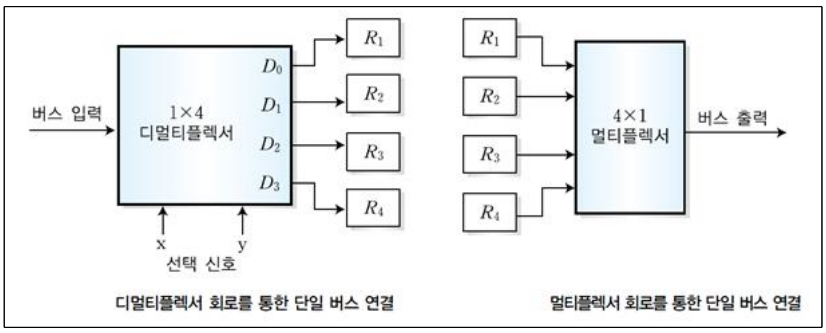


6) 버스전송 방식
- 단일 버스로 연결된 경우와 병렬 버스로 연결된 경우로 분류
  - 병렬 버스 전송의 경우에서는 버스 내의 선의 개수와 레지스터를 구성하는 플립플롭의 개수가 일치해야 함
  - 단일 버스로 전송되는 경우에는 버스로 사용하는 회선이 1개이므로 1비트 신호인 제어 신호를 전송하거나 직렬 전송만이 가능함
- 디코더를 활용한 단일 버스 연결
  - 레지스터 4개를 단일 버스를 통해서 데이터를 공동으로 전송할 경우에는 레지스터를 선택하기 위해 2비트가 입력되는 디코더를 사용함
    : 4개의 레지스터 R1 ~ R4이 디코더에 연결됨
    : 선택 신호 x와 y가 디코더에서 D_0 ~ D_3 까지의 출력 중 하나를 선택하여 연결된 레지스터를 활성화함



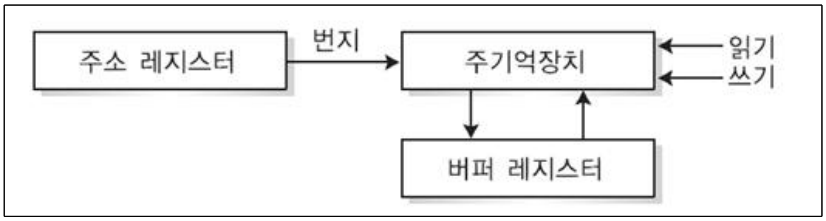
- 멀티플렉서와 디멀티플렉서의 단일버스 연결
  - 단일 버스가 여러 레지스터 중 선택된 하나의 레지스터에 수신하는 방법으로 디멀티플렉서 회로를 사용함
    : 선택 신호에 의해 디멀티플렉서 회로의 출력 D_0~D_3 가 생성되므로 이를 직접 레지스터에 연결하여 수신하도록 함

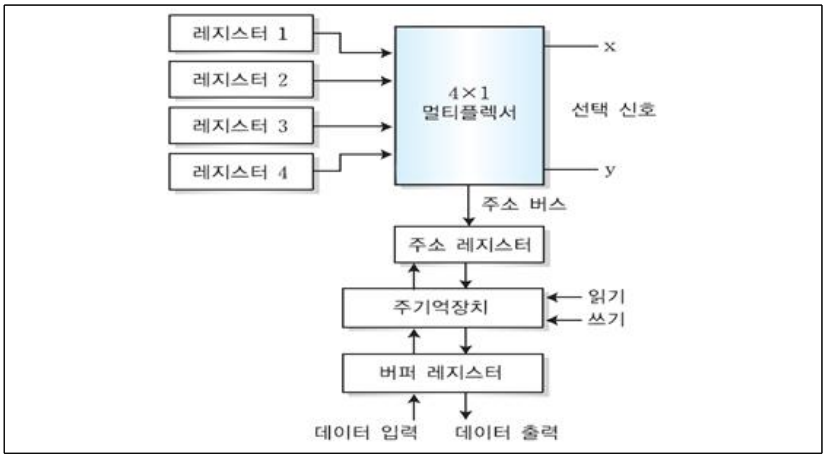
- 레지스터에 저장된 데이터를 단일 버스로 송신할 때는 멀티플렉서를 사용
  - 레지스터 4개가 멀티플렉서를 통하여 단일 회선인 버스에 연결
  - x와 y가 지적하는 회로에 의해 R_1 ~ R_4 중 하나가 선택, 선택된 레지스터는 버스로 데이터를 출력함



7) 기억장치전송 방식

- 주기억장치에 데이터를 쓰기 동작과 읽는 동작을 수행하기 위해서는 해당 위치를 알려주는 주소번지가 필요
  - 주소번지를 저장하는 기억장치 주소 레지스터가 필요하고 주기억장치에서 읽혀지거나 기록할 때 임시적으로 저장되는 기억장치 버퍼 레지스터가 필요함
- 4개 레지스터가 주소번지를 저장
  - 멀티플렉서 하나를 선택하여 기억장치 주소 레지스터로 전달하는 경우임

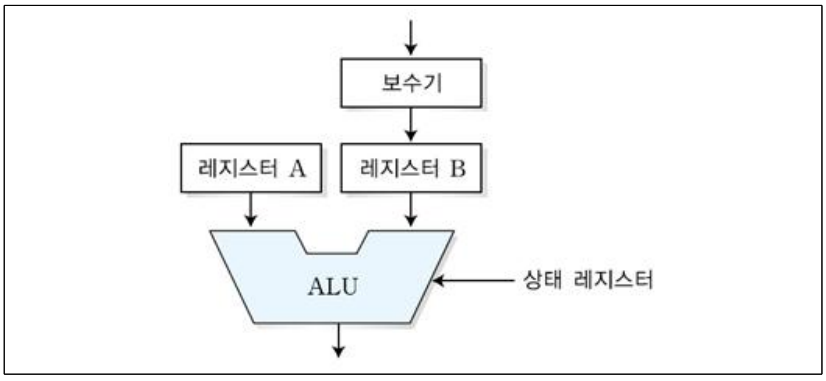




학습내용2 : 연산장치

1. 연산장치의 개요

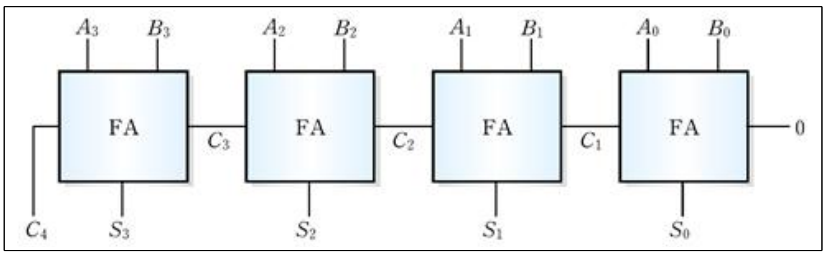
1) 연산장치
- 연산될 데이터와 연산한 결과를 기억시킬 레지스터가 필요함
- 연산의 상태를 나타내기 위한 상태 레지스터들의 연결이 필요함
- 산술논리연산장치(ALU)
    - 덧셈을 하기 위한 가산기임
    - 연산에 이용되는 데이터나 연산 결과 등을 일시적으로 보관하기 위한 누산기
    - 데이터를 보관하는 기억장치 버퍼 레지스터 등이 필요함
    - 보수를 만들기 위한 보수기, 계산 결과의 상태를 점검하기 위한 상태 레지스터 등으로 구성됨



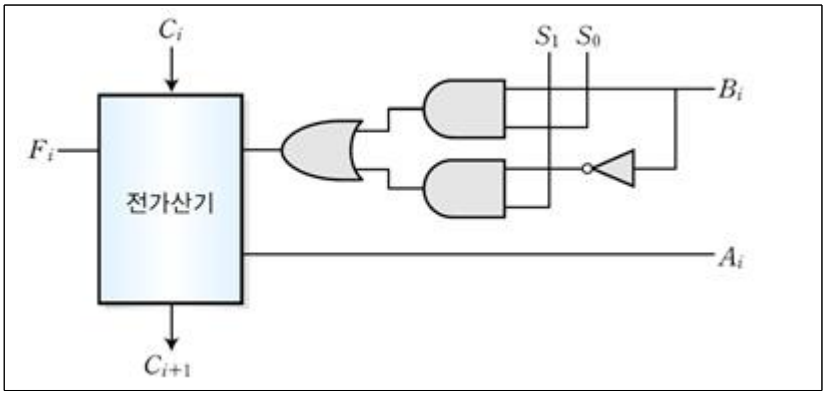
2) ALU에서의 연산회로

- 산술연산회로

- 4비트의 병렬 가산기로 구성됨



- 병렬 가산기가 단순한 덧셈 기능뿐만 아니라 여러 가지 연산을 수행하기 위해서는 구성요소인 전가산기의 한쪽 입력단자에 논리 회로를 추가해야 함



- 두 개의 선택 신호 S_1과 S_0 그리고 자리올림 C_i에 의해서 8가지 연산을 수행함

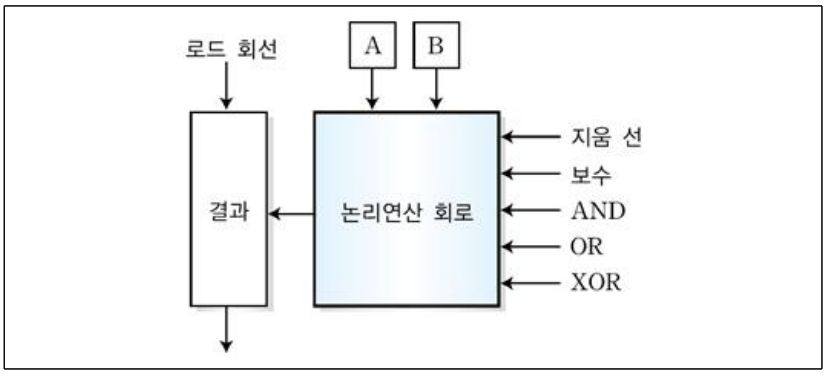
3) 산술연산 논리회로 함수 테이블

선택 신호와 자리 올림			출력	연산 동작
S_1	S_0	C_i		
0	0	0	$F=A$	A를 전송
0	0	1	$F=A+1$	A의 증가 캐리를 포함한 덧셈
0	1	0	$F=A+B$	A와 B의 가산
0	1	1	$F=A+B+1$	A와 B의 캐리 가진 가산
1	0	0	$F=A+B'$	A와 B의 1의 보수 가산
1	0	1	$F=A+B'+1$	감산(A와 B의 2의 보수 덧셈)
1	1	0	$F=A-1$	A를 1 감소
1	1	1	$F=A$	A를 전송

4) 논리연산 회로

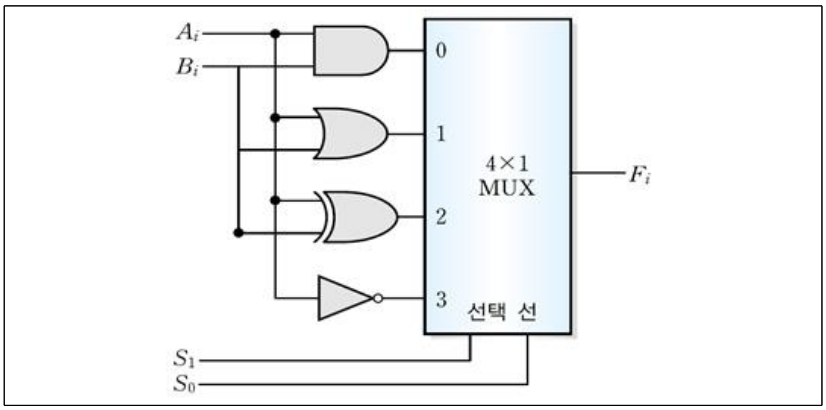
- 논리 게이트들을 이용하여 조합논리 회로를 구성하면 다양한 논리연산을 수행함
  - 데이터 A와 B를 입력하는 회선과 연산의 종류를 선택하는 회선이 존재함
  - 입력되는 데이터 A와 B의 내용을 연산 지시에 의해 AND, OR, XOR 및 NOT 연산을 수행함
  - 연산 지시 신호는 논리연산의 선택뿐만 아니라 결과가 기억될 장소를 0으로 지우는 지움 선과 NOT 연산으로 1의 보수가 되도록 하는 보수 회선이 존재할 수 있음

- 논리연산 회로의 구성도



5) 논리연산회로의 내부 구성과 함수 테이블

- 논리연산 회로 내부
  - 논리 게이트의 조합논리 회로와 멀티 플렉서로 구성 됨
  - 4개의 논리연산이 선택신호 S_0와 S_1에 의해서 하나가 선택되고 출력하게 함



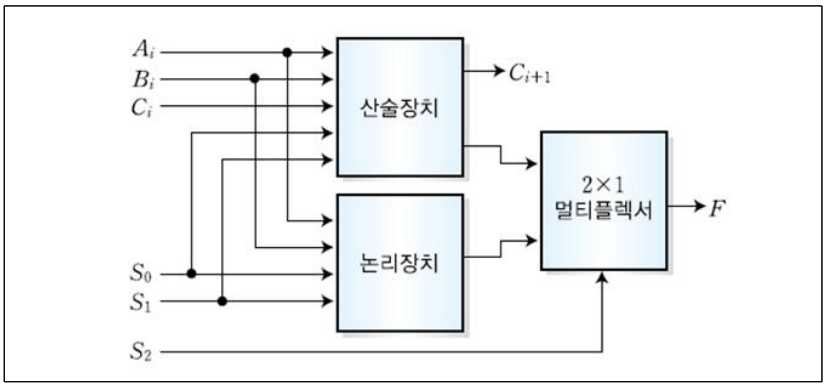
- 논리 연산의 함수 테이블

S_1	S_0	출력	연산 동작
0	0	$F = A \wedge B$	AND
0	1	$F = A \vee B$	OR
1	0	$F = A \oplus B$	XOR
1	1	$F = A'$	NOT

2. CPU의 논리회로 설계

1) ALU

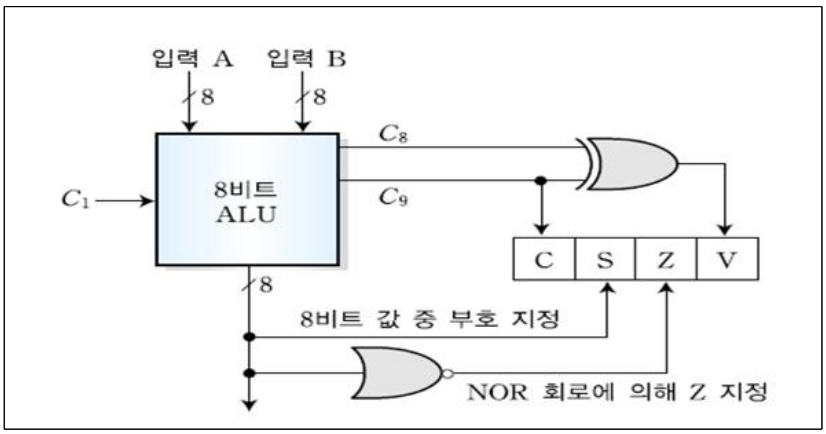
- 설계된 산술연산 회로와 논리연산 회로를 조합하면 ALU가 완성
- 선택선 S_1과 S_0 는 산술연산 회로와 논리연산 회로가 공통으로 사용
- 선택선 S_2는 두 회로 중 하나를 선택하는데 사용
- S_2가 0이면 산술연산을, S_2가 1이면 논리연산을 수행



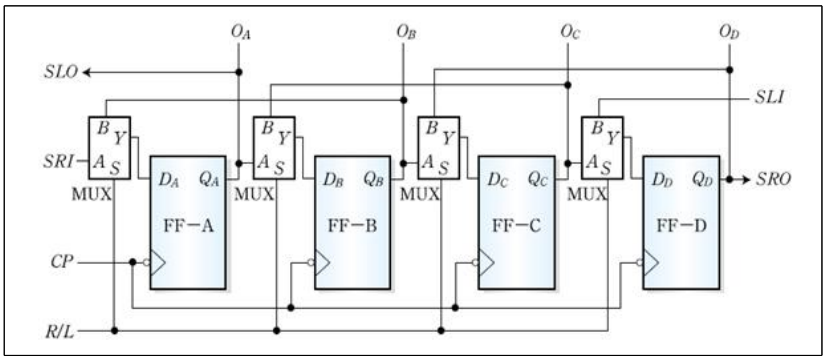
2) 상태 비트

- 플래그(flag) 또는 조건 코드(condition code)라고도 불림
- CPU를 설계하는 과정에서 상태 비트는 여러 종류가 존재
    - 자리올림(Carry : C), 오버플로우(Overflow : V),
      제로(Zero : Z), 부호(Sign : S)의 4가지는 필수적임
    - C(Carry) : 자리올림 비트가 1이면 자리올림수가 발생한 함
- CPU를 설계하는 과정에서 상태 비트는 여러 종류가 존재
    - S(Sign) : 부호비트가 1이면 음수이고, 0이면 양수 상태를 나타냄
    - Z(Zero) : ALU의 연산결과 모든 비트의 출력이 0이면 제로 비트는 1이 되고, 그렇지 않으면 제로 비트는 0이 됨
    - V(Overflow) : ALU의 두 자리 올림 수 C_8, C_9 를 XOR를 한 결과가 1이면 오버 플로우가 발생한 것이고, 그렇지 않고 0이면 오버 플로우는 발생하지 않은 상태임

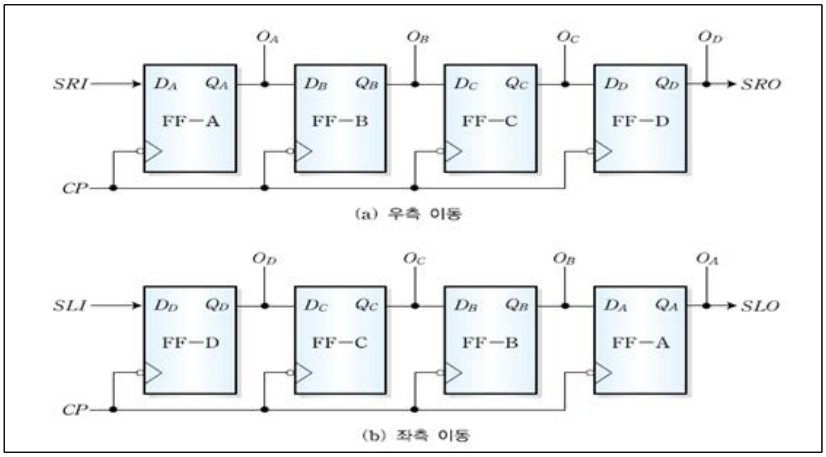
3) 8비트 ALU와 상태 레지스터
- 오버 플로우 비트를 위해서는 XOR 게이트가 필요
- 제로 비트를 동작하게 하기 위해서는 NOR 게이트가 필요



4) 이동기(shifter)
- 이동 방향은 왼쪽과 오른쪽이고, 이동 방향의 마지막 비트 값은 바깥으로 밀려남, 이동 레지스터라고도 함
- 양방향 이동 레지스터



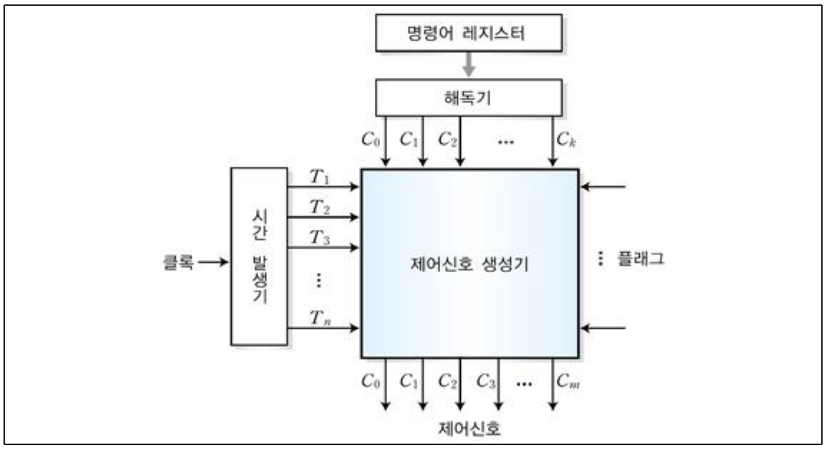
5) 양방향 이동 레지스터의 좌측 이동과 우측 이동
- 오른쪽 이동은 왼쪽에서 새로운 비트 값이 입력되고 오른쪽에서는 마지막 비트가 배출됨
- 왼쪽 이동은 오른쪽에서 새로운 비트 값이 입력되고 왼쪽에서 마지막 비트가 배출됨



학습내용3 : 연산장치와 제어장치

1. CPU의 논리회로 설계

1) 하드와이어적 제어장치
- 제어장치의 유형
    - 하드웨어만으로 설계된 하드와이어적 제어장치(hardwired control unit)
    - 소프트웨어가 포함된 마이크로 프로그램 된 제어장치(micro-programmed control unit)
- 논리 회로에 의해 제작된 하드와이어적 제어장치
    - 하드와이어적 제어장치는 제어 신호의 생성과정에서 지연이 매우 작다는 장점을 갖음
    - 구현 논리 회로는 명령 코드 및 주소 지정 모드 등에 따라 매우 복잡하다는 단점을 갖음



【학습정리】

1. 레지스터의 전송방식은 레지스터 간 전송방식, 버스 전송방식, 기억장치 전송 방식등이 있다.

2. 연산장치는 산술연산회로와 논리연산회로로 구분할 수 있다.

3. 제어장치의 유형에는 하드웨어만으로 설계된 하드와이어적 제어장치와 소프트웨어가 포함된 마이크로프로그램된 제어장치 등이 있다.