

디지털 부속품

2.1 집적 회로

◆ 디지털 게이트를 구성하는 전자 부품들을 포함하는 실리콘 반도체

- 칩 안의 많은 게이트들은 회로의 요구에 따라 서로 연결되어 있고 핀을 통해 외부 회로나 신호에 연결

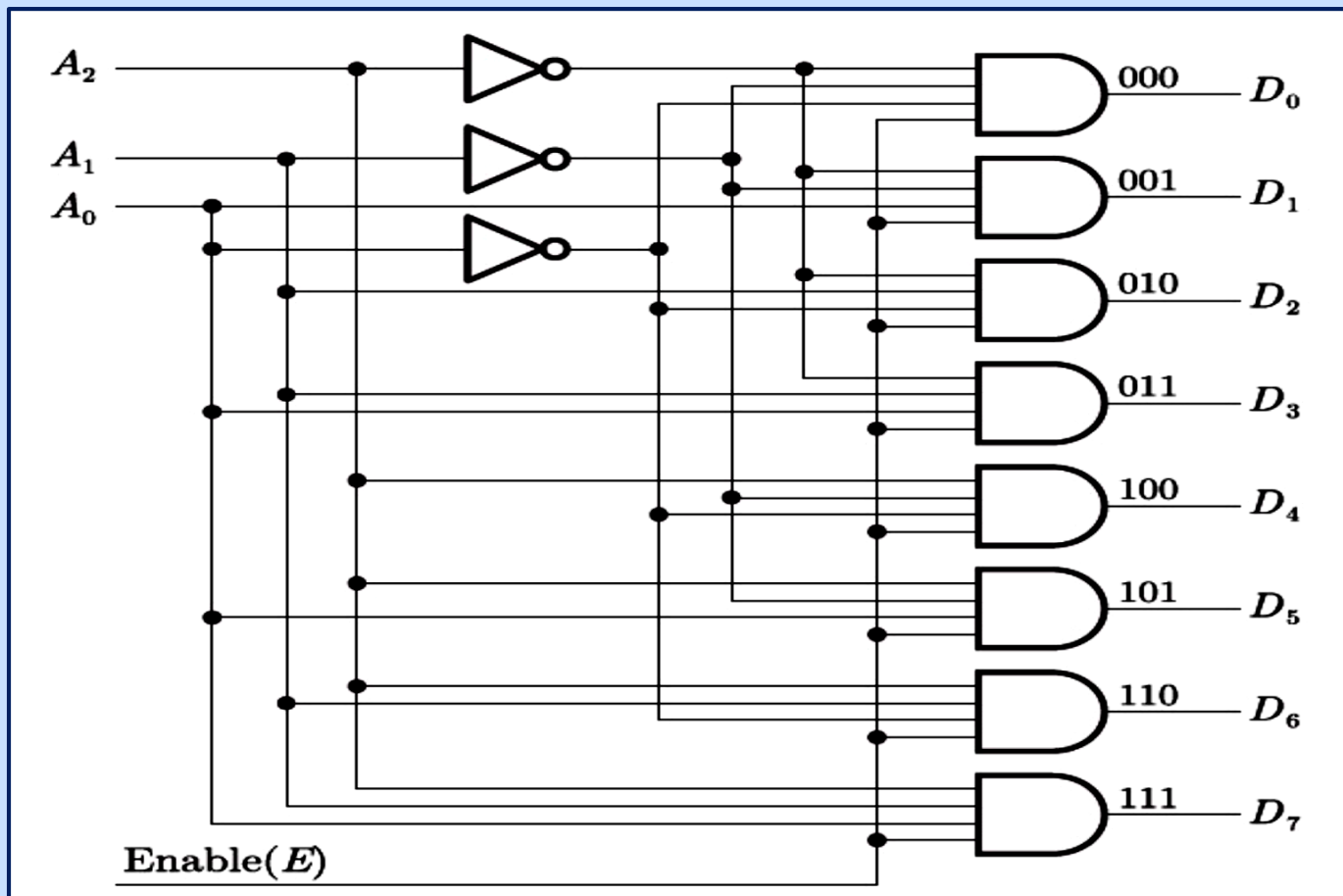
◆ 집적 회로 기술이 발전함에 따라 칩 안에 담을 수 있는 게이트 수가 크게 증가

- 소규모 집적 (SSI) 장치 : 10개 이하의 게이트가 하나에 칩에 들어가 있음
- 중규모 집적 (MSI) 장치 : 10에서 200개까지의 게이트를 집적하여 디코더, 가산기, 레지스터 같은 기본적인 디지털 장치를 구현
- 대규모 집적 (LSI) 장치 : 200에서 1000개까지의 게이트를 집적하여 프로세서나 메모리 칩 같은 디지털 시스템을 형성
- 초대규모 집적 (VLSI) 장치 : 수천 개의 게이트를 집적하여 대형 메모리나 복잡한 마이크로 컴퓨터 칩을 형성

2.2 디코더

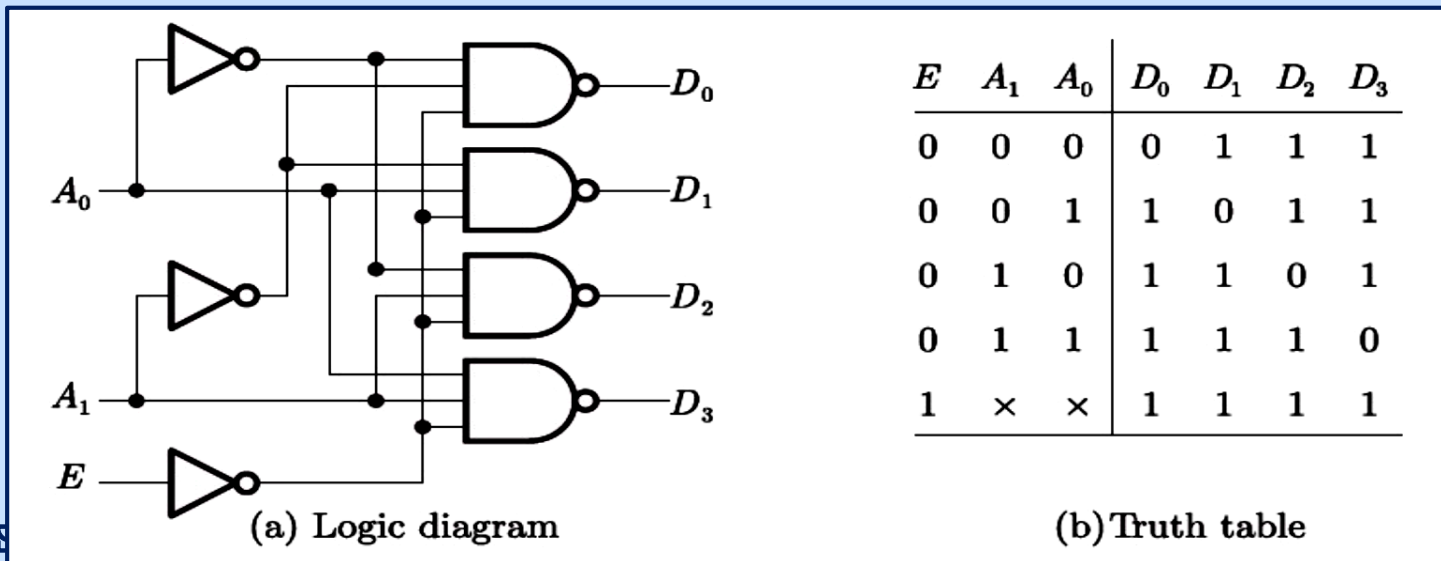
◆ n 비트로 코딩된 이진 정보를 2^n 개의 서로 다른 출력으로 바꾸어 주는 회로

- 3 x 8 디코더 : 3개의 인버터, 8개의 AND 게이트, 인에이블 입력(E)

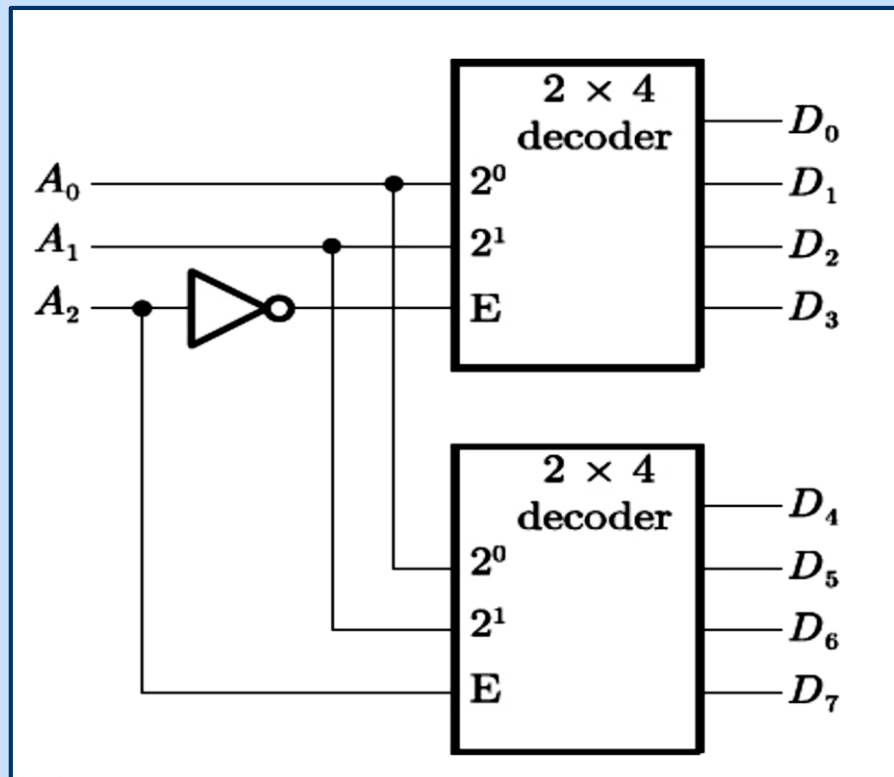


Enable	Inputs			Outputs							
E	A ₂	A ₁	A ₀	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
0	x	x	x	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	0	1	0
1	0	1	0	0	0	0	0	0	1	0	0
1	0	1	1	0	0	0	0	1	0	0	0
1	1	0	0	0	0	0	1	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0	0
1	1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0

– 보수화된 형태의 출력이 경제적이므로 NAND 게이트 디코더 형태로 구성



- ◆ 두 개 이상의 디코더를 동일한 인에이블 입력에 연결해 한 개의 큰 디코더를 구성할 수 있음
- 두 개의 2×4 디코더로 3×8 디코더를 구성한 예

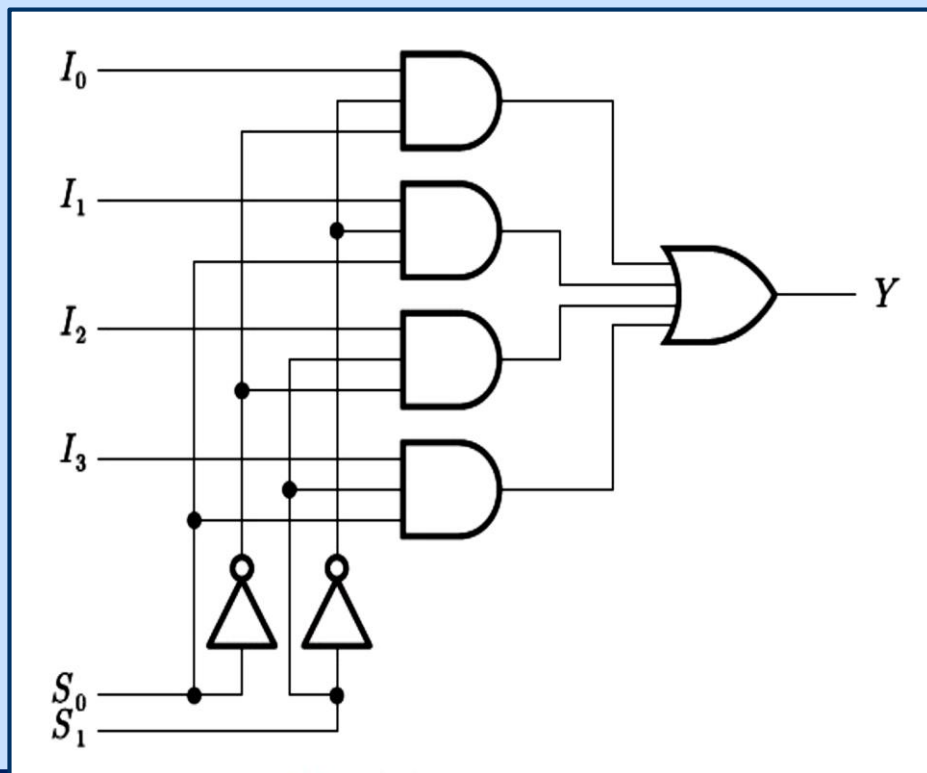


2.3 멀티플렉서

◆ n 개의 선택 입력에 따라 2^n 개의 입력을 하나의 출력에 선택적으로 연결시켜

주는 조합 회로로 데이터 선택기라고도 함 (MUX)

– 4 × 1 멀티플렉서 : 4개의 입력, 2개의 선택 입력

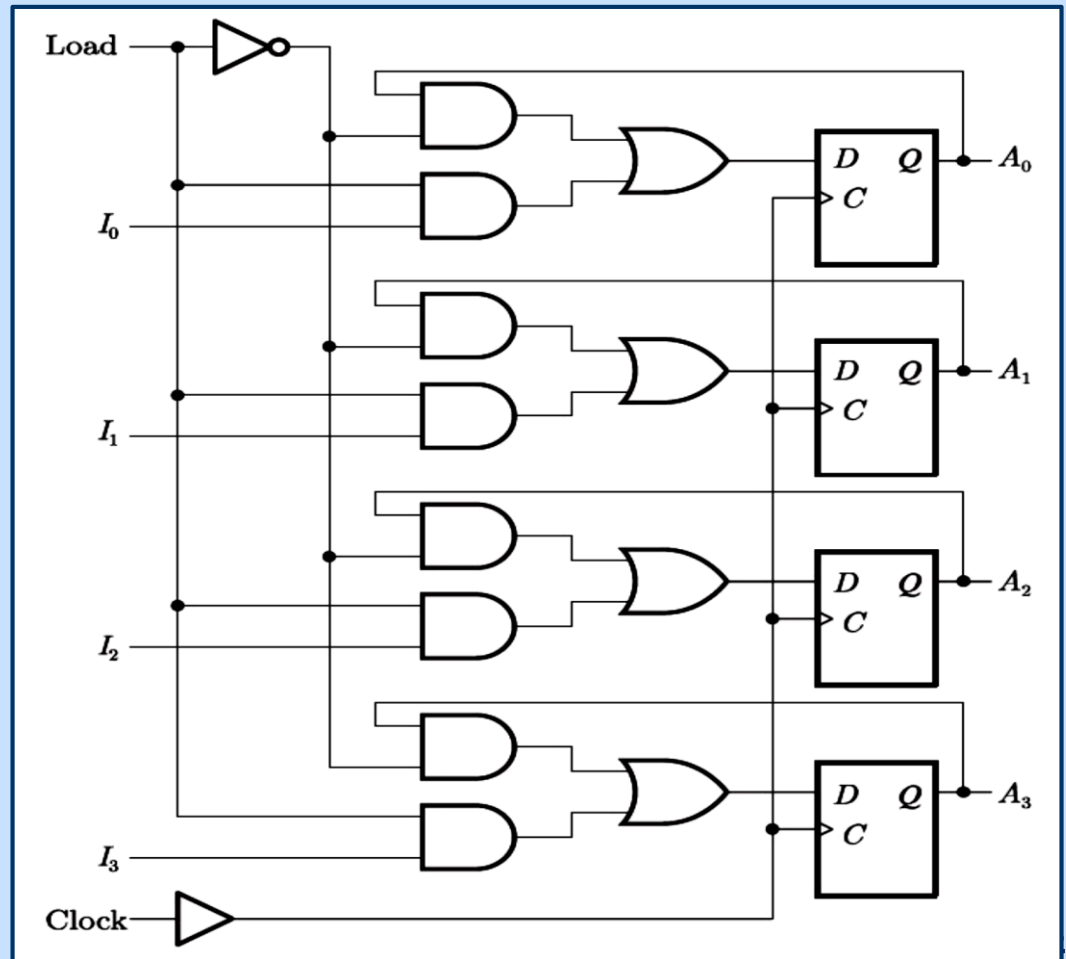
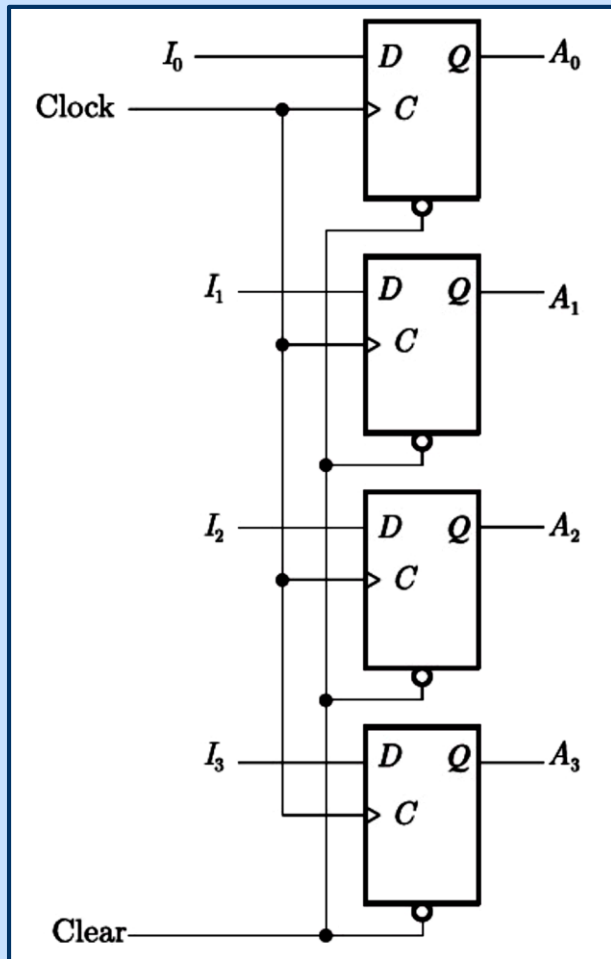


Select		Output
S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

2.4 레지스터

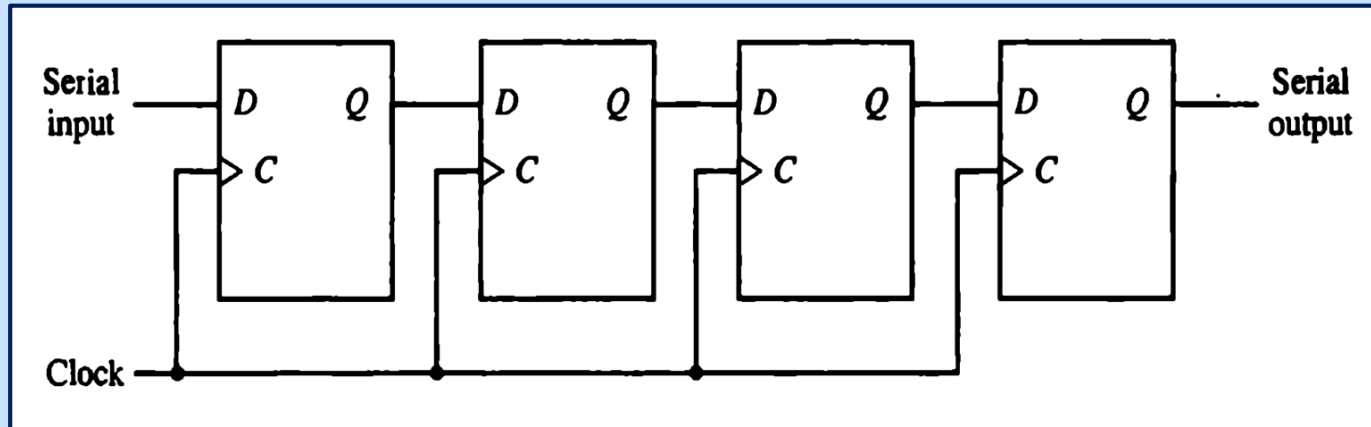
◆ n비트의 정보를 저장하기 위해 n개의 플립플롭과 조합 회로로 구성

- 병렬 로드를 가진 4비트 레지스터 : 로드 입력 1이고 상승 에지에서 동작



2.5 시프트 레지스터

◆ 레지스터에 저장되어 있는 이진 정보를 단방향 또는 양방향으로 이동시킴

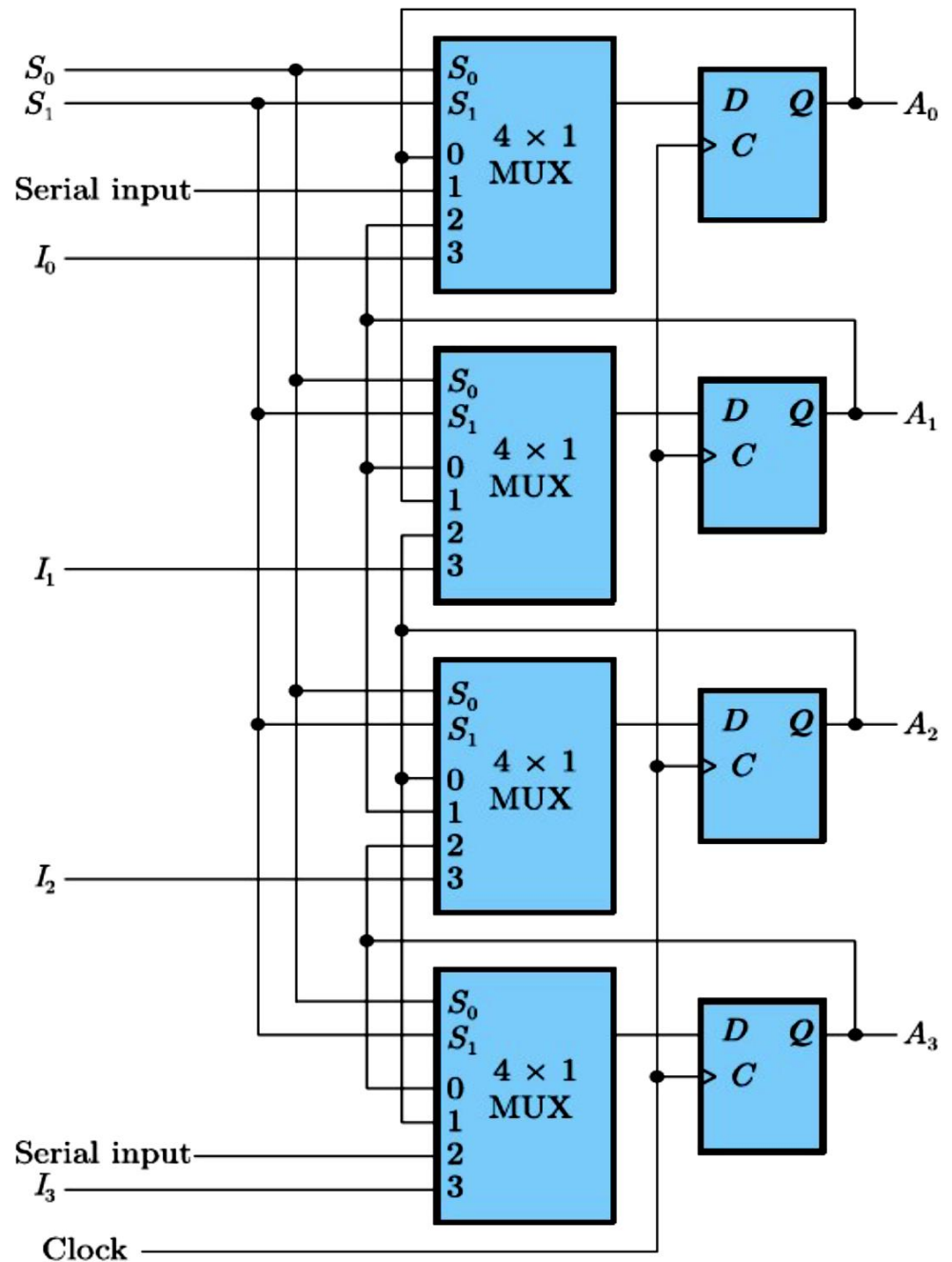


◆ 일반적인 시프트 레지스터의 기능 (최소 하나의 기능을 포함)

- 모든 동작을 동기화하기 위한 클럭 펄스 입력, n개의 병렬 출력 라인
- 직렬 입력 라인을 가지고 있으며 오른쪽 시프트 동작을 수행
- 직렬 입력 라인을 가지고 있으며 왼쪽 시프트 동작을 수행
- 병렬 전송을 위한 n개의 입력 라인을 가지고 있으며 병렬 로드 동작을 수행
- 클럭 펄스가 입력되더라도 레지스터의 정보를 불변하도록 하는 제어 상태

- 병렬 로드를 가진 4비트 양방향 시프트 레지스터

Mode control		Register Operation
S_1	S_0	
0	0	No change
0	1	Shift left
1	0	Shift right
1	1	Parallel load

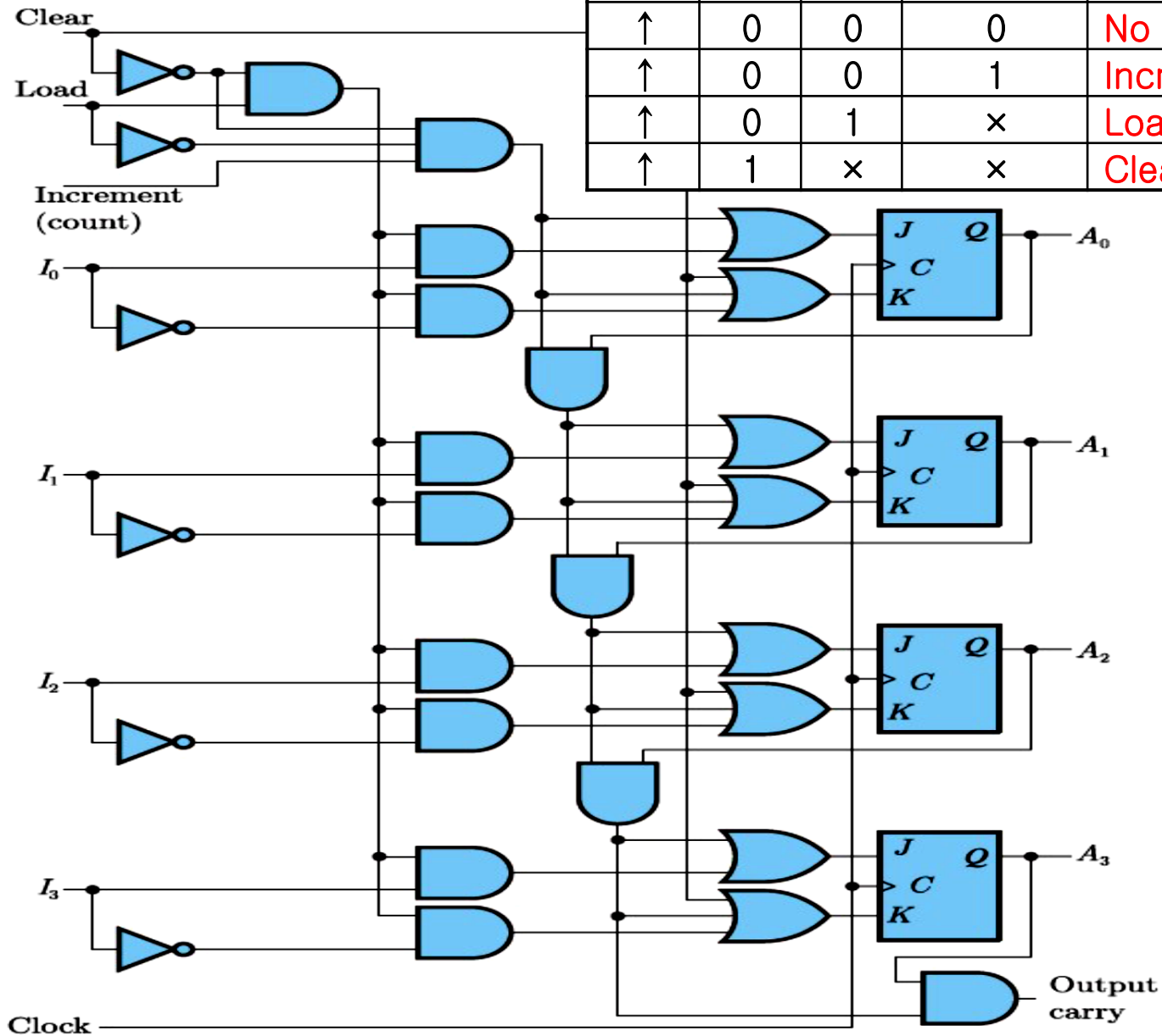


- 시프트 레지스터는 주로 원격지 시스템 사이에 데이터를 전송할 때 사용
- 원거리의 전송에는 병렬 전송보다 직렬 전송이 더 경제적임
- 시프트 레지스터를 통해 송신측에서는 병렬-직렬 변환을 하여 데이터를 전송하고 수신측에서는 전송 받은 직렬 데이터를 다시 병렬로 변경

2.6 이진 카운터

- ◆ 입력 펄스에 따라 미리 정해진 순서대로 상태 변화가 진행되는 레지스터
 - 이벤트 발생 횟수를 세거나 동작 순서를 제어하는 타이밍 신호 생성에 사용
 - 이진 카운터는 이진수의 순서를 따르는 카운터로 n 비트 이진 카운터는 0에서 (2^n-1) 까지 카운터함
 - 병렬 로드를 가진 이진 카운터 : 클리어와 로드 입력이 0이고 인크리먼트 입력이 1일 때 정상적인 이진 카운터 동작을 수행

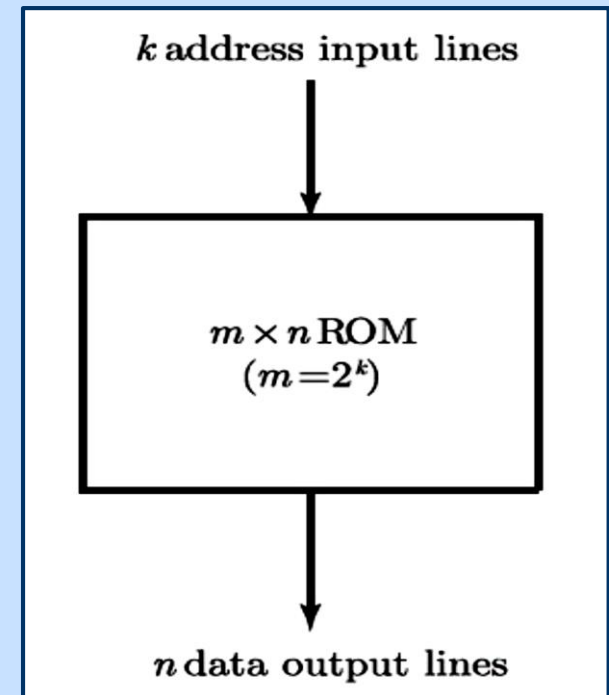
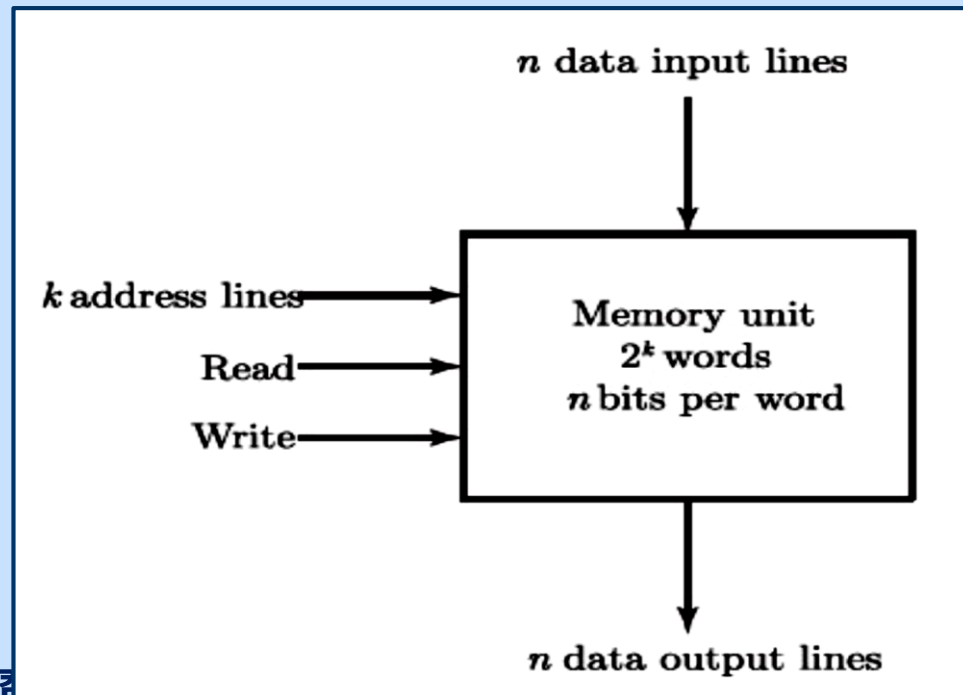
Clock	Clear	Load	Increment	Operation
↑	0	0	0	No change
↑	0	0	1	Increment count by 1
↑	0	1	x	Load inputs $I_0 \sim I_3$
↑	1	x	x	Clear outputs to 0



2.7 메모리 장치

◆ 메모리의 각 워드는 0에서 2^k-1 까지의 주소를 가지고 k 개의 주소 입력의 값에 따라 특정 워드가 선택됨

- RAM은 워드의 물리적인 위치에 관계없이 접근 절차나 접근 시간이 동일
- ROM 안에서 어떤 패턴이 형성되면 전원이 끊어지더라도 데이터는 계속 남아 있게 됨



데이터의 표현

3.3 고정 소수점 표현

◆ 이진 소수점이 항상 동일한 위치에 존재하는 것으로 분수를 표현하기 위해
레지스터 맨 왼쪽에 두거나 정수를 표현하기 위해 레지스터의 맨 오른쪽에 둠

- 정수 표현 방법

부호절대값 표현 : 부호비트 + 절대값 ($-14 \rightarrow 1\ 0001110$)

부호화된 1의 보수 표현 : 0은 1로, 1은 0으로 변환 ($-14 \rightarrow 1\ 1110001$)

부호화된 2의 보수 표현 : 1의 보수 + 1 ($-14 \rightarrow 1\ 1110010$)

- 1의 보수는 논리 연산에 주로 사용되고 음수나 뺄셈 연산에 2의 보수 사용

- 음수가 2의 보수로 표시될 때 뺄셈은 감수를 2의 보수로 변환 후 더하면 됨

$$(\pm A) - (-B) = (\pm A) + (+B), (\pm A) - (+B) = (\pm A) + (-B)$$

$$11111010 - 11110011 = 11111010 + 00001101 = 00000111 \text{ (캐리 제거)}$$

- ◆ n비트 두 수의 덧셈과 뺄셈 연산에서 결과가 n+1 비트이면 오버플로우 발생
 - 레지스터의 길이가 한정되어 있으므로 연산 결과가 오버플로우가 발생하는지를 검사하고 플래그 레지스터의 해당 비트에 저장하는 것이 필요
 - 부호 비트 밑에서 부호 비트로 올라온 캐리와 부호 비트로부터 생긴 캐리가 서로 다르면 오버플로우가 발생

3.4 부동 소수점 표현

- ◆ 부동소수점은 가수와 지수로 표현되고 지수는 소수점의 위치를 알려줌
 - 이진 부동소수점의 정규화 표현 : $m \times 2^e$ ($1001.11 \rightarrow (.1001110)_2 \times 2^4$)
 - 부동 소수점의 산술 연산은 고정 소수점 연산에 비하여 복잡하고 시간이 많이 걸림, 매우 작은 수와 매우 큰 수를 표시하기에 편리
 - 부동 소수점 연산을 위한 하드웨어 블록이 없는 컴퓨터는 부동 소수점 연산을 위한 서브루틴을 사용

3.6 에러 검출 코드

◆ 이진 정보를 전송할 때 외부 잡음이 들어가면 1이 0으로 0이 1로 변하는
에러가 발생할 가능성이 있음

– 가장 일반적인 에러 검출 방법은 패리티 비트를 사용하는 것으로 1비트
에러를 검출할 수 있음

Message(xyz)	P(odd)	P(even)
000	1	0
001	0	1
010	0	1
011	1	0
100	0	1
101	1	0
110	1	0
111	0	1

$$P_e = (x \oplus y \oplus z)$$

$$P_o = (x \oplus y \oplus z)'$$

$$P_e = 1 \oplus 1 \oplus 1 = 0 \oplus 1 = 1$$

$$P_o = (1 \oplus 1 \oplus 1)' = 0$$

