Lecture 10

레지스터

개요

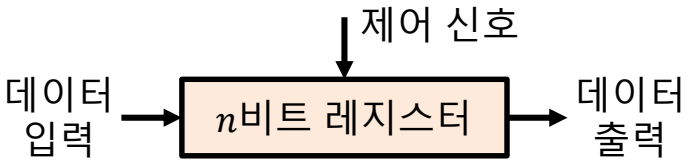
- 플립플롭
  - 1비트를 저장할 수 있음
- 레지스터(**register**)
  - 플립플롭 여러 개를 일렬로 배열하여 적당히 연결함으로써 여러 비트로 구성된 2진수를 저장할 수 있음
  - n비트 레지스터는 플립플롭 n개로 구성되며, 2진 정보 n비트를 저장할 수 있음
  - 배운 카운터는 레지스터의 특별한 형태라고 할 수 있음
  - 주로 여러 비트를 일시적으로 저장하거나 저장된 비트를 왼쪽 또는 오른쪽으로 하나씩 **시프트**(shift)할 때 사용함
    - 예, 계산기에서 숫자 367을 입력할 때

개요

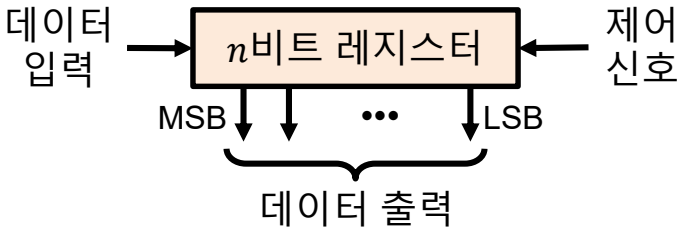
- 레지스터(**register**)
  - CPU 내부에서 연산의 중간 결과를 임시 저장하거나 CPU 상태를 저장하는 데 사용함
  - 레지스터에 데이터를 입력, 출력할 때 2가지 방법이 있음
    - 병렬 시프트 : 병렬로 동시에 시프트하는 방법
    - 직렬 시프트 : 한 비트씩 직렬로 시프트하는 방법
  - 데이터를 입력, 출력하는 방법에 따라 레지스터의 4종류가 있음
    ① 직렬 입력 – 직렬 출력
    ② 직렬 입력 – 병렬 출력
    ③ 병렬 입력 – 직렬 출력
    ④ 병렬 입력 – 병렬 출력

개요

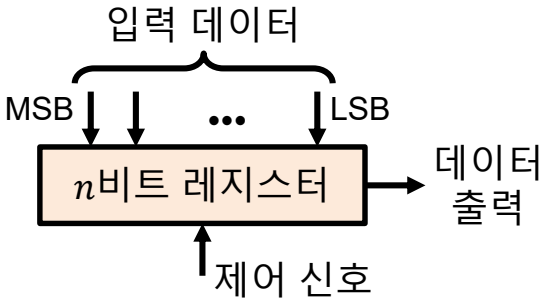
- 레지스터 종류



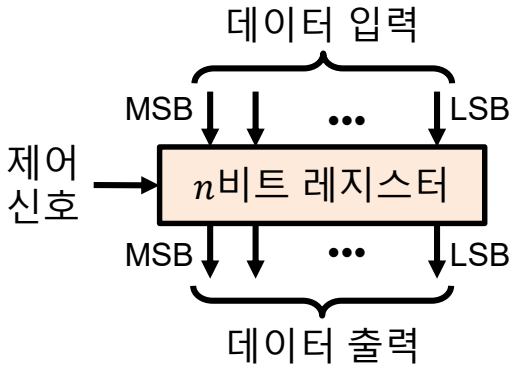
직렬 입력 – 직렬 출력



직렬 입력 – 병렬 출력



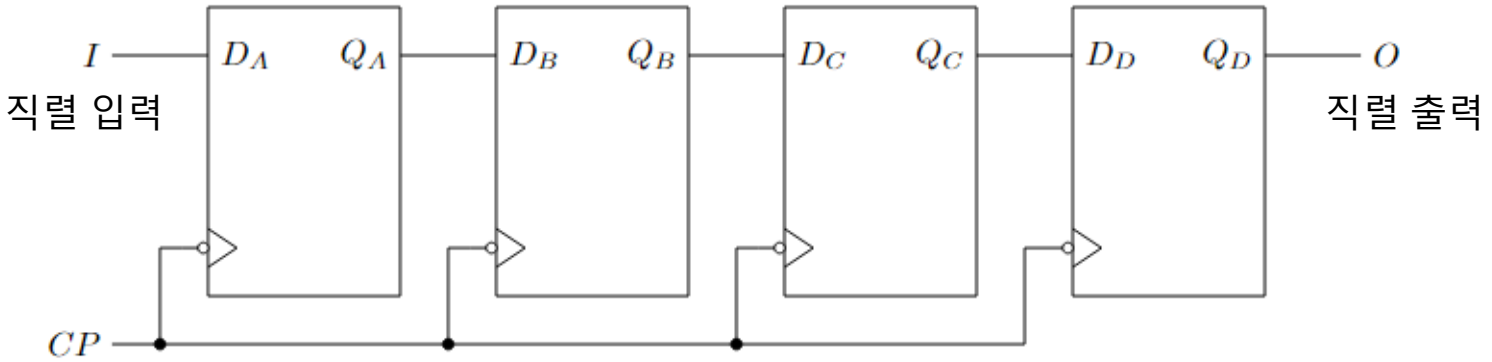
병렬 입력 – 직렬 출력



병렬 입력 – 병렬 출력

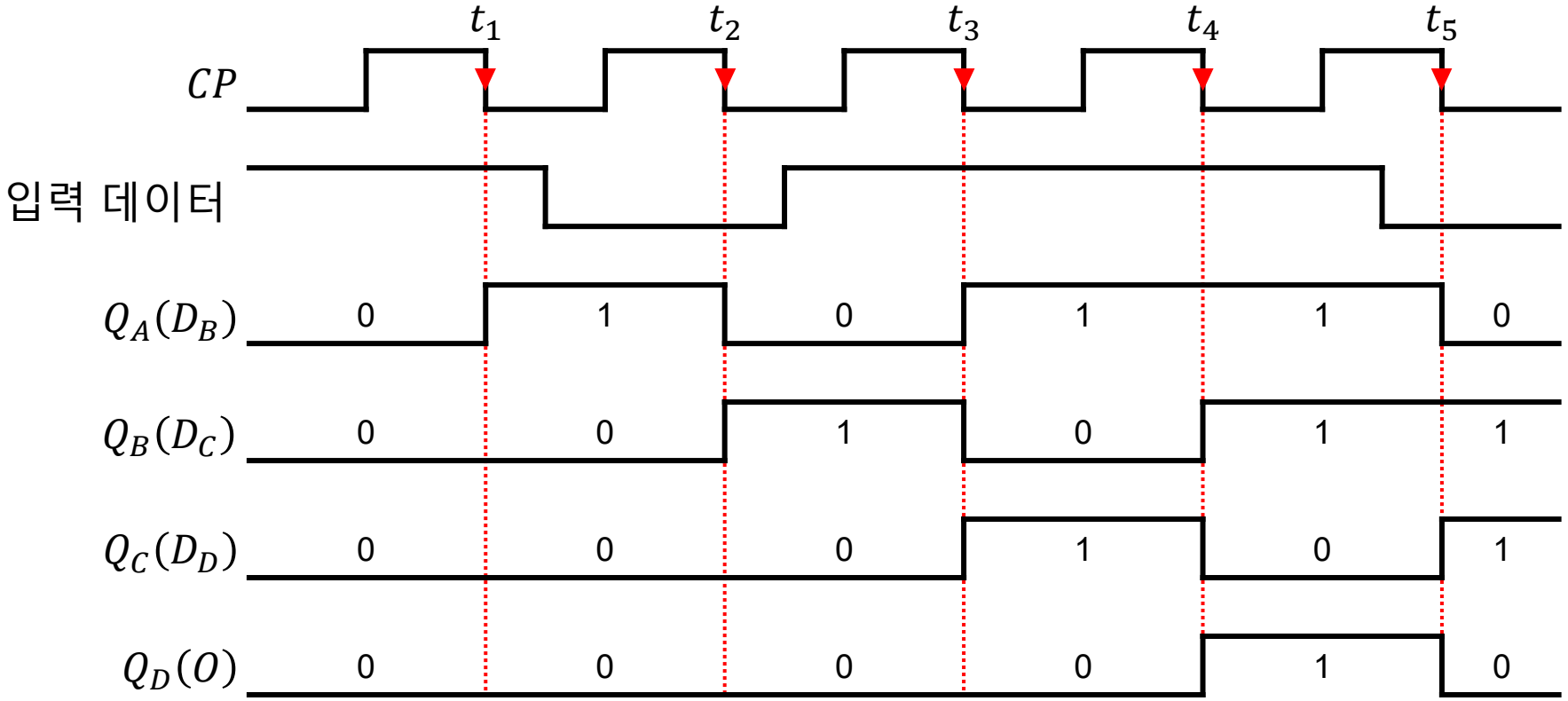
직렬 입력 – 직렬 출력

- 4비트 직렬 입력 – 직렬 출력 레지스터
  - 첫 번째 클록 펄스의 하강 에지에서 입력 데이터가 샘플되어 첫 번째 플립플롭 출력인 Q_A로 시프트됨
  - 두 번째 클록 펄스의 하강 에지에서 Q_A는 Q_B로, Q_B는 Q_C로, Q_C는 Q_D로 시프트됨과 동시에 새로운 데이터가 샘플되어 Q_A에 입력됨
  - 위 과정은 새로운 클록 펄스의 하강 에지마자 반복되므로 네 번째 클록 펄스의 하강 에지에서 비로소 Q_D에 처음에 입력된 데이터가 나타남



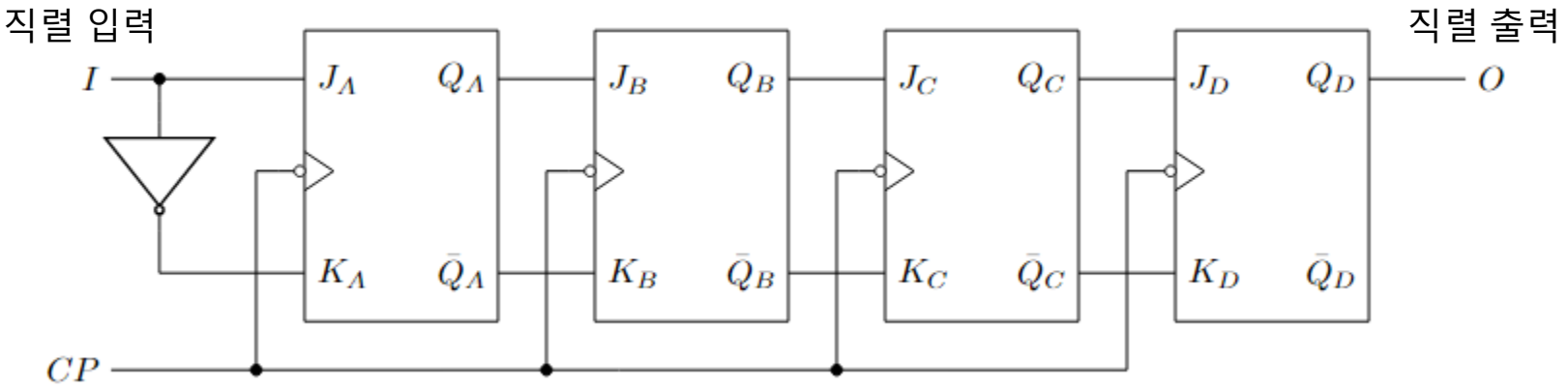
직렬 입력 – 직렬 출력

- 4비트 직렬 입력 – 직렬 출력 레지스터



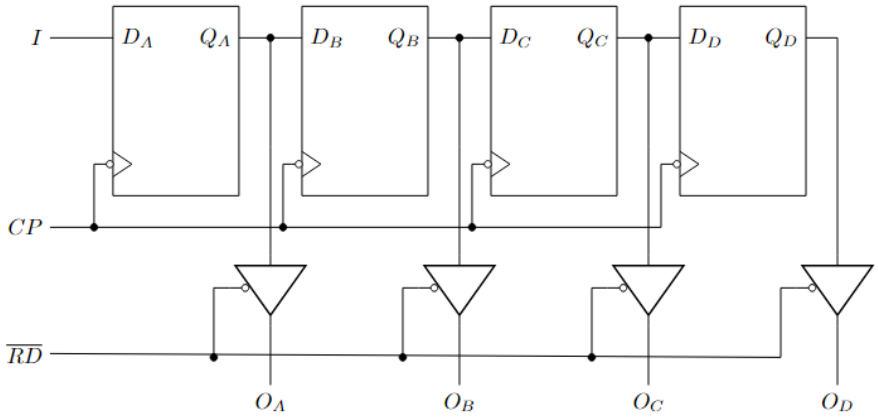
직렬 입력 – 직렬 출력

- 4비트 직렬 입력 – 직렬 출력 레지스터
  - JK 플립플롭을 사용해 직렬 입력 – 직렬 출력 레지스터를 구성할 수 있음

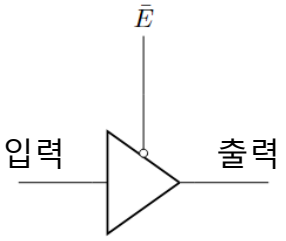


직렬 입력 – 병렬 출력

- 4비트 직렬 입력 – 병렬 출력 레지스터
  - 직렬 입력 – 직렬 출력 레지스터와 기본적으로 동일한 구조이며, 각 출력에 **3상태**(tri-state) 버퍼가 연결된 점만 다름
    - 3상태 버퍼 : 출력 상태는 High 상태, Low 상태 그리고 **하이 임피던스**(high impedance) 3가지이며, 하이 임피던스 상태는 입력과 출력이 연결되어 있지 않다는 의미임
  - 클록 펄스 4개가 입력되면 4비트 직렬 입력 데이터가 레지스터에 모두 저장됨
  - $\overline{RD} = 0$이면 각 플립플롭에 저장되어 있던 데이터가 출력 O_A, O_B, O_C, O_D에 동시에 출력됨
  - $\overline{RD} = 1$이면 3상태 버퍼가 하이 이피던스 상태가 되어 입력과 출력 사이의 연결이 완전히 끊어져서 출력되지 않음



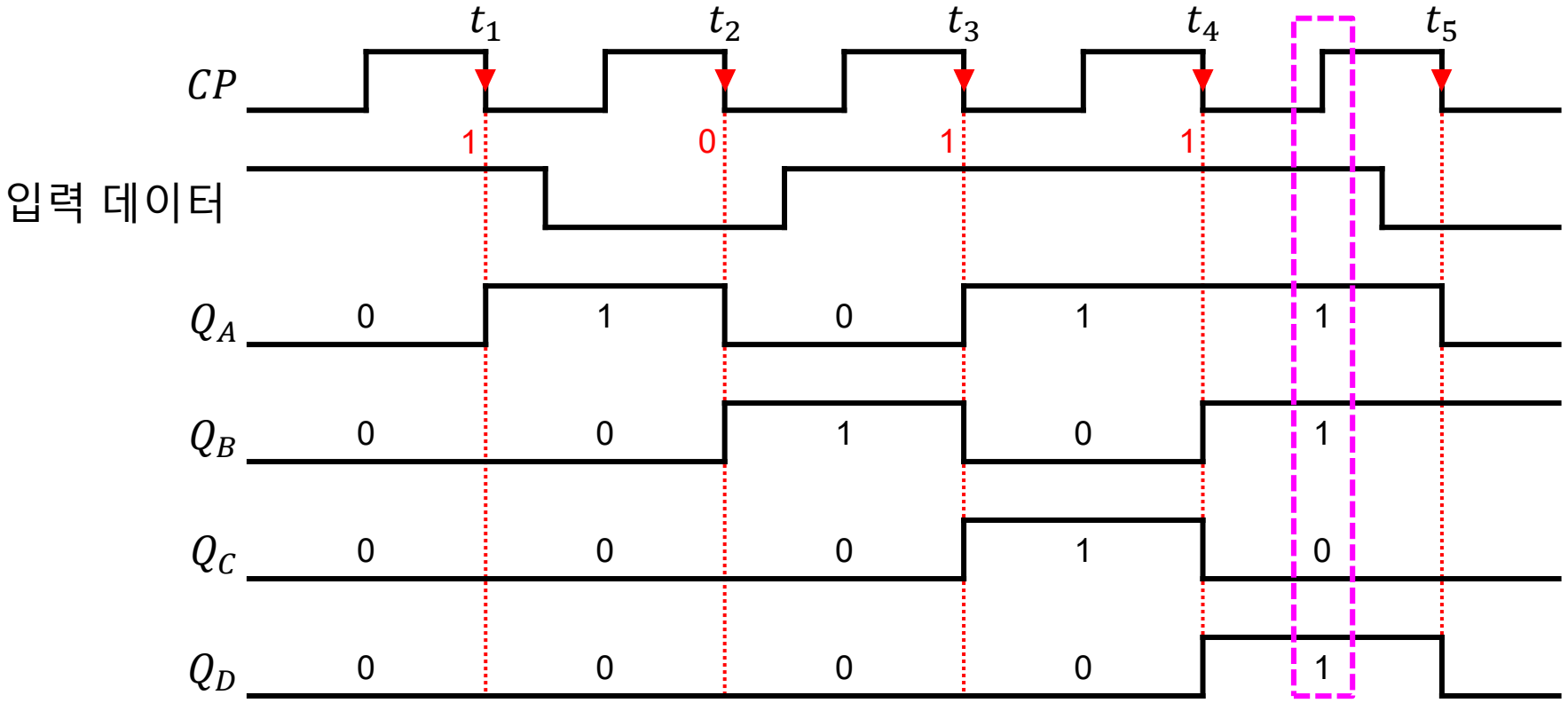
$\overline{E}$	입력	출력
0	0	0
0	1	1
1	0	HighZ
1	1	HighZ



직렬 입력 – 병렬 출력

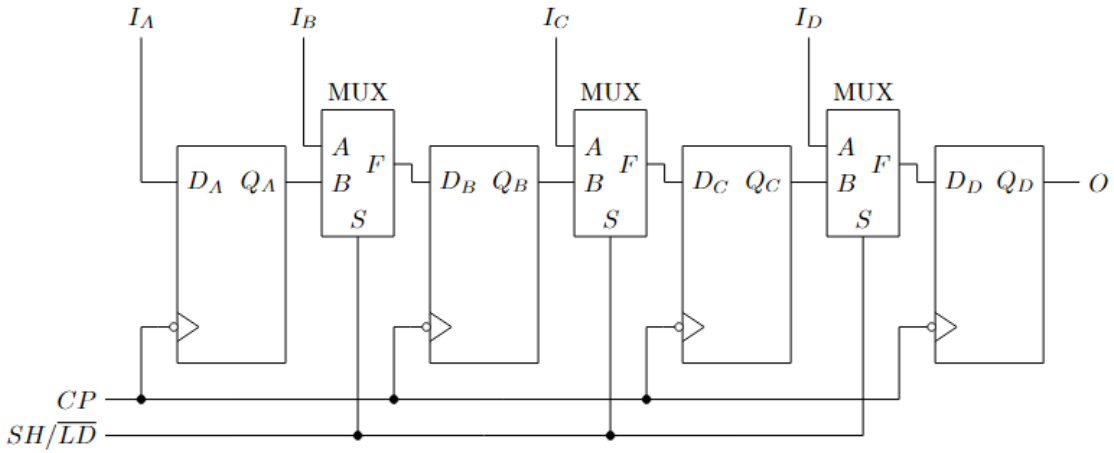
- 4비트 직렬 입력 – 병렬 출력 레지스터

$\overline{RD} = 0$이면 $O_D O_C O_B O_A = 1011$인
병렬 출력을 받을 수 있음

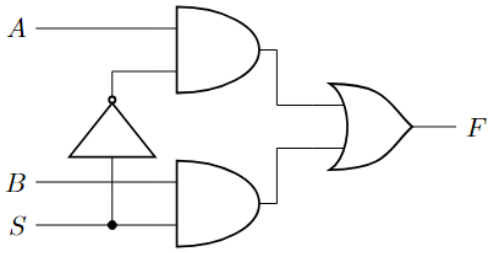


병렬 입력 – 직렬 출력

- 4비트 병렬 입력 – 직렬 출력 레지스터
  - MUX의 동작 : $S = 0$이면 입력 A와 출력 F가 연결됨, $S = 1$이면 입력 B와 출력 F가 연결됨
  - $SH/\overline{LD} = 0$이면 입력 데이터 I_A, I_B, I_C, I_D가 각 플립플롭의 입력 D_A, D_B, D_C, D_D와 각각 연결되므로 클록 펄스의 하강 에지에서 입력 데이터의 각 비트가 동시에 샘플되어 대응하는 플립플롭의 출력 Q에 저장됨
  - $SH/\overline{LD} = 1$로 하면 클록 펄스의 하강 에지마다 레지스터 내용이 오른쪽으로 시프트되어 마지막 단 레지스터 출력에서 데이터 비트가 직렬로 나옴
  - 입력 데이터는 클록 펄스 4개마다 샘플해야 함

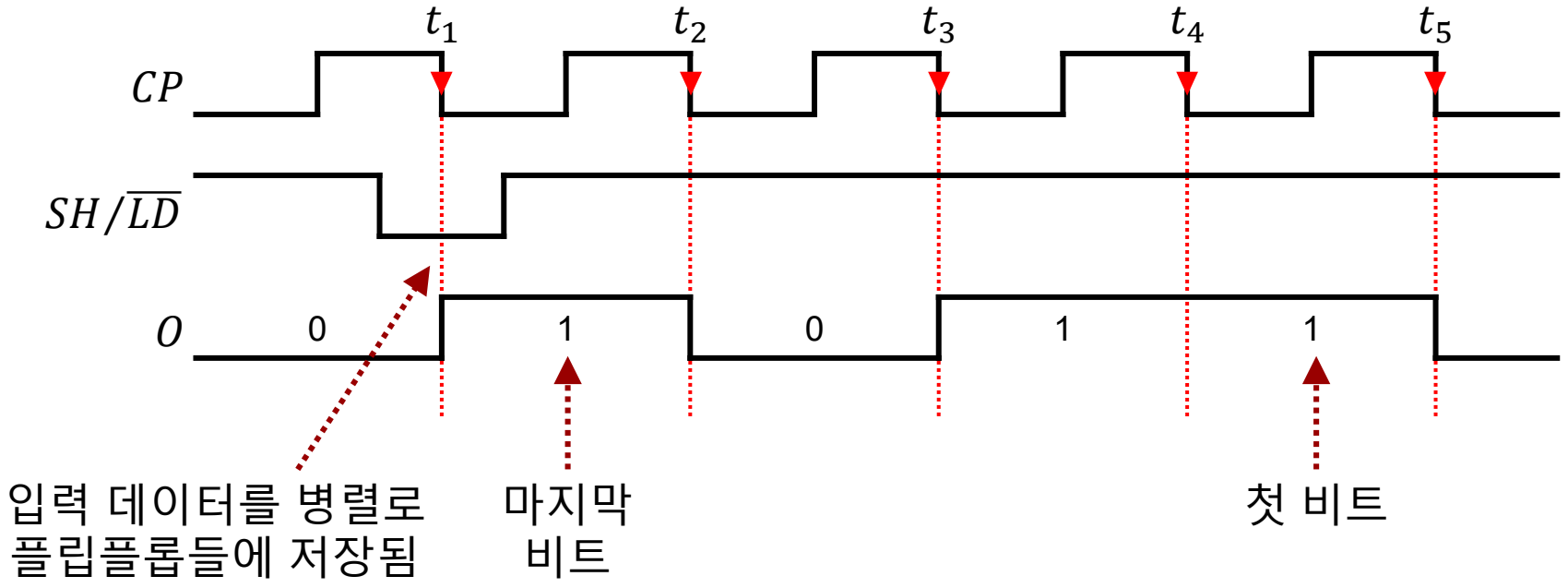


MUX의 회로



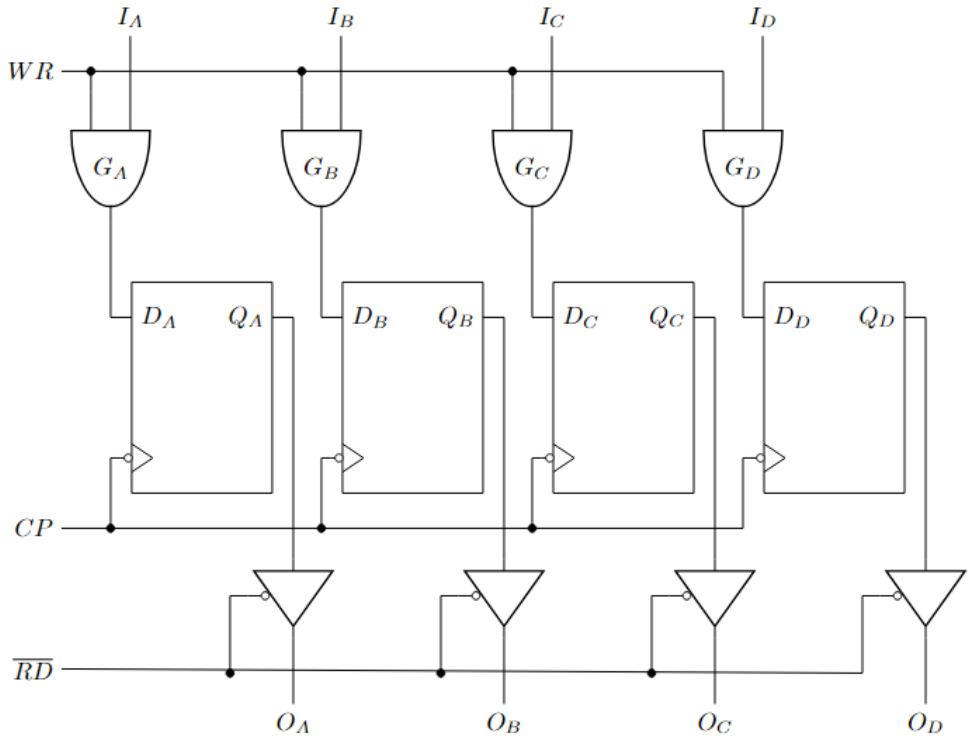
병렬 입력 – 직렬 출력

- 4비트 병렬 입력 – 직렬 출력 레지스터
  - 입력 $I_A I_B I_C I_D = 1101$인 경우



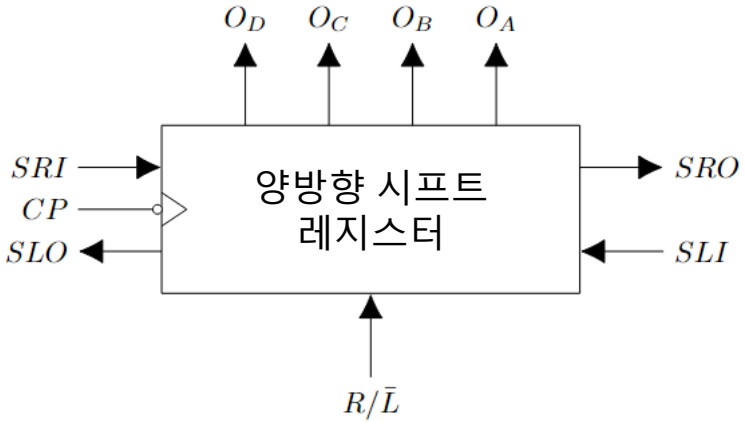
병렬 입력 – 병렬 출력

- 4비트 병렬 입력 – 병렬 출력 레지스터
  - 병렬 입력을 제어하는 WR 신호와 병렬 출력을 제어하는 $\overline{RD}$ 신호가 있음



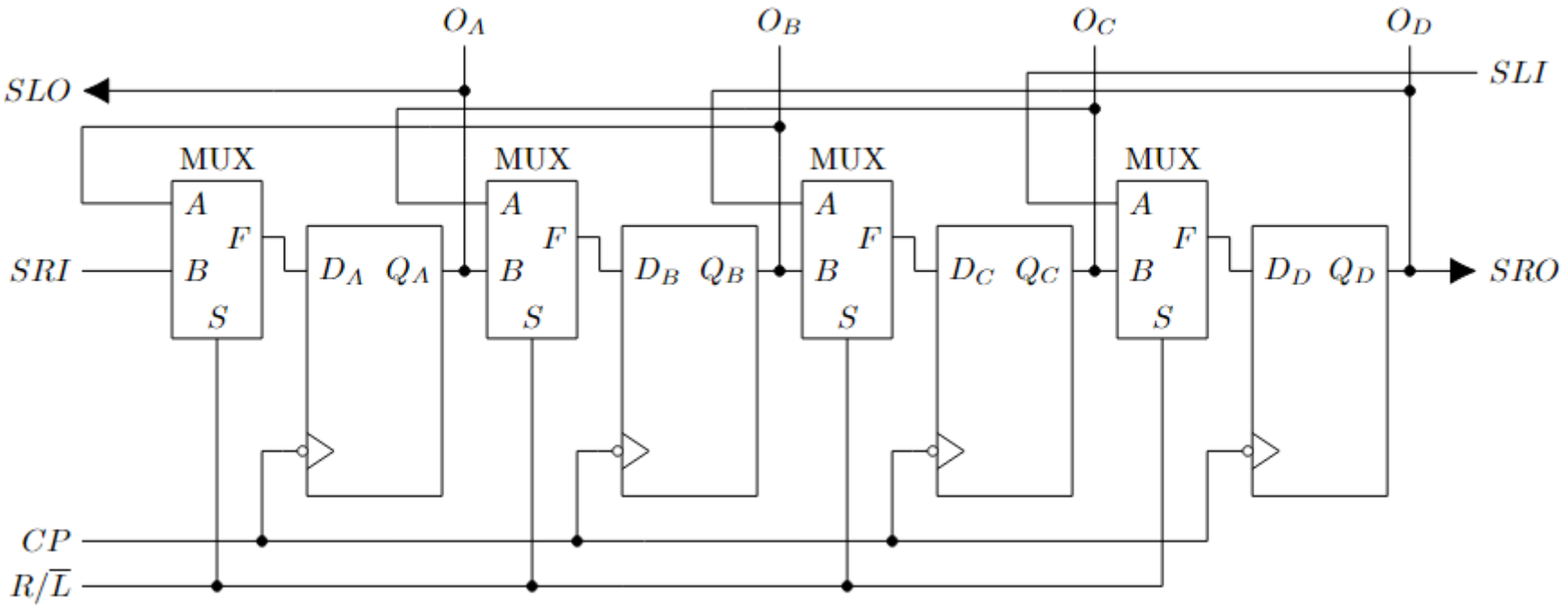
양방향 시프트 레지스터

- 양방향 시프트가 가능한 직렬 입력 – 병렬 출력 레지스터
  - SRI(shift right input) : 오른쪽 시프트 입력을 나타냄
  - SLO(shift left output) : 왼쪽 시프트 출력을 나타냄
  - SLI(shift left input) : 왼쪽 시프트 입력을 나타냄
  - SRO(shift right output) : 오른쪽 시프트 출력을 나타냄
  - $R/\bar{L}$: 시프트 방향을 제어함
    - $R/\bar{L} = 1$일 경우 데이터를 SRI에 입력시켜 오른쪽으로 시프트하면서 SRO에서 출력함
    - $R/\bar{L} = 0$일 경우 데이터를 SLI에서 입력해 왼쪽으로 시프트하면서 SLO에서 출력함



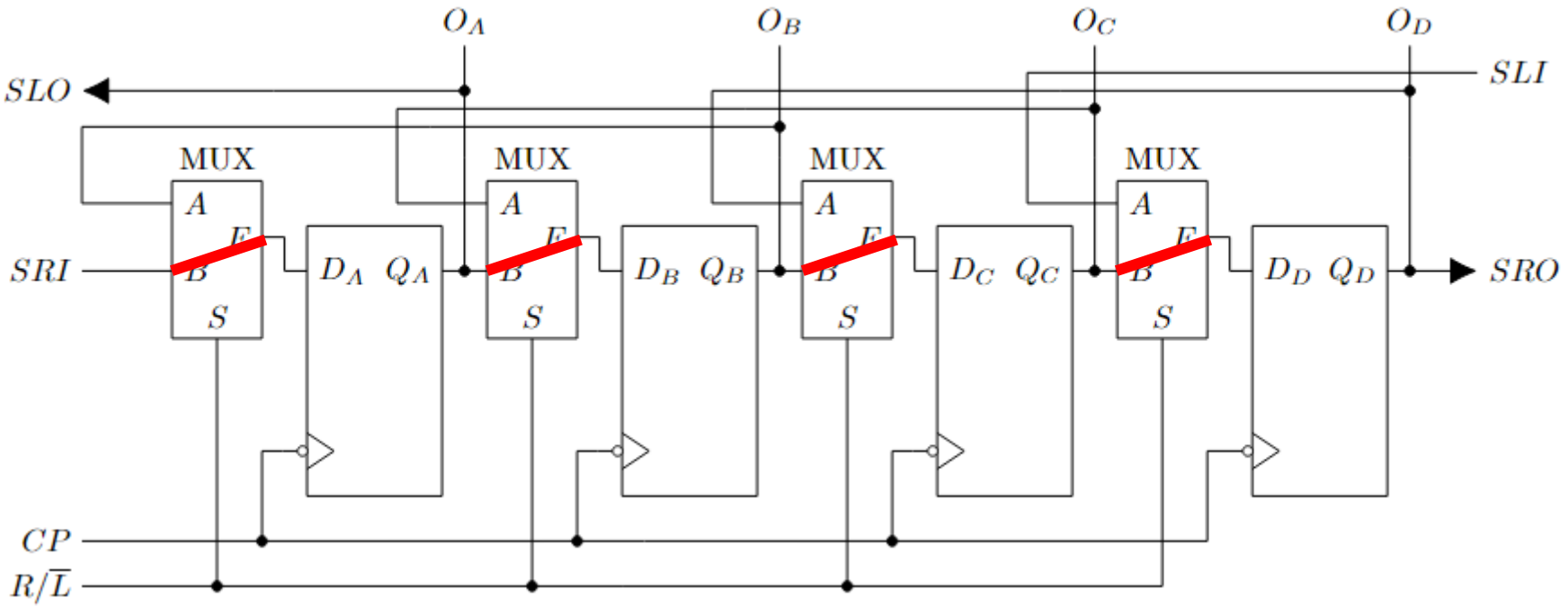
양방향 시프트 레지스터

- 양방향 시프트가 가능한 직렬 입력 – 병렬 출력 레지스터



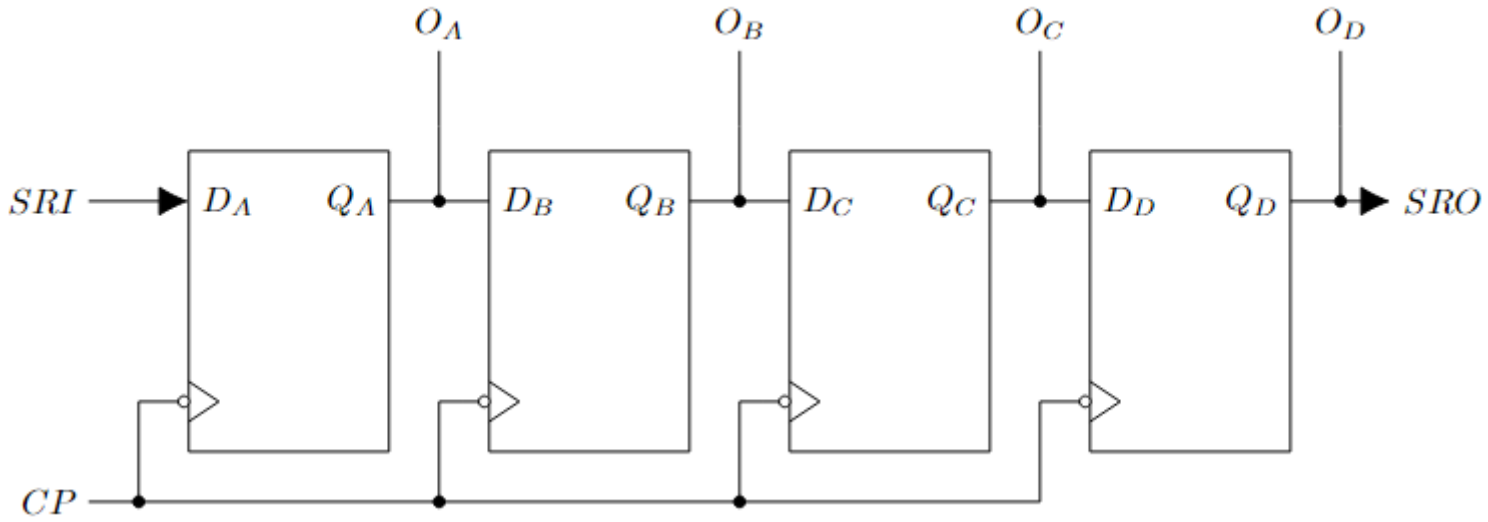
양방향 시프트 레지스터

- 양방향 시프트가 가능한 직렬 입력 – 병렬 출력 레지스터
  - $R/\overline{L} = 1$일 경우 $SRI \rightarrow D_A$, $Q_A \rightarrow D_B$, $Q_B \rightarrow D_C$, $Q_C \rightarrow D_D$로 연결됨



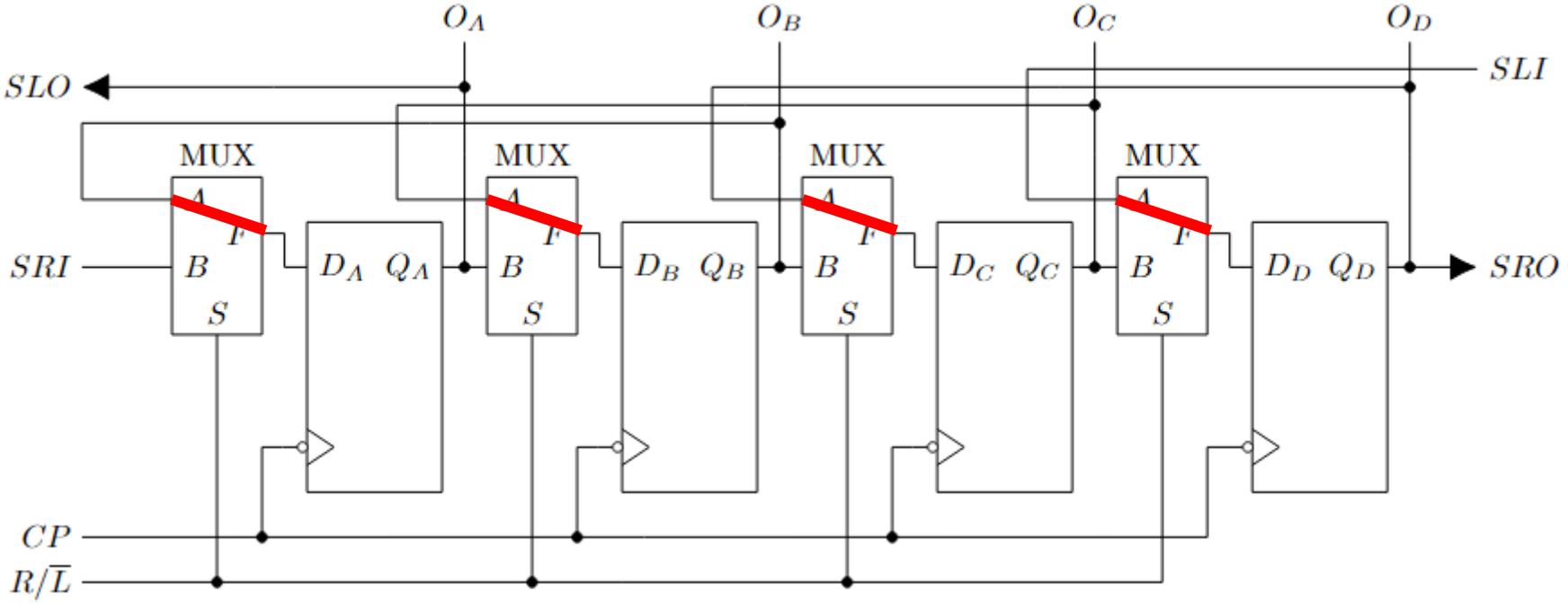
양방향 시프트 레지스터

- 양방향 시프트가 가능한 직렬 입력 – 병렬 출력 레지스터
  - $R/\bar{L} = 1$일 경우 $SRI \rightarrow D_A$, $Q_A \rightarrow D_B$, $Q_B \rightarrow D_C$, $Q_C \rightarrow D_D$로 연결됨
  - 오른쪽 시프트 레지스터가 됨



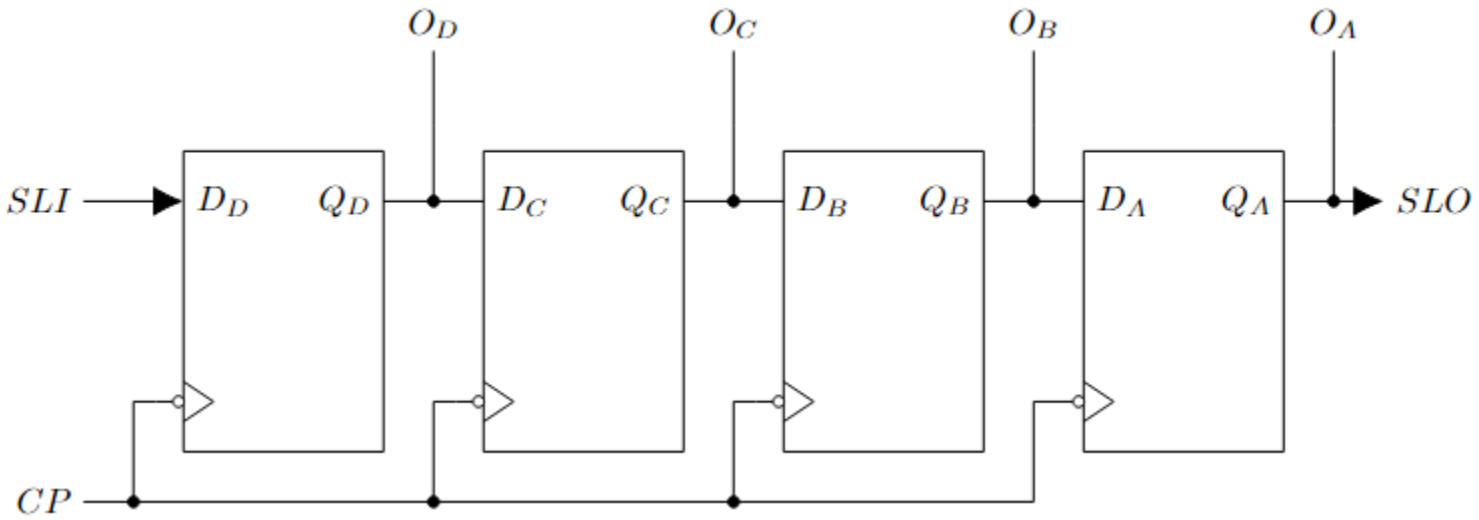
양방향 시프트 레지스터

- 양방향 시프트가 가능한 직렬 입력 – 병렬 출력 레지스터
  - $R/\overline{L} = 0$일 경우 $SLI \rightarrow D_D$, $Q_D \rightarrow D_C$, $Q_C \rightarrow D_B$, $Q_B \rightarrow D_A$로 연결됨



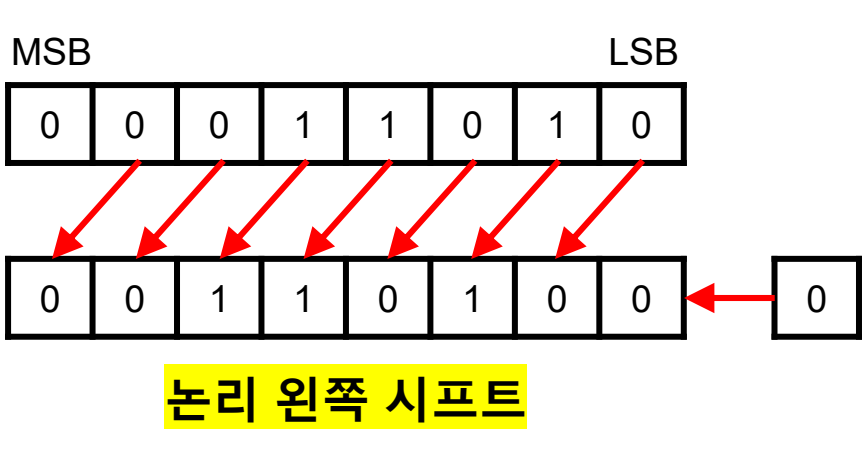
양방향 시프트 레지스터

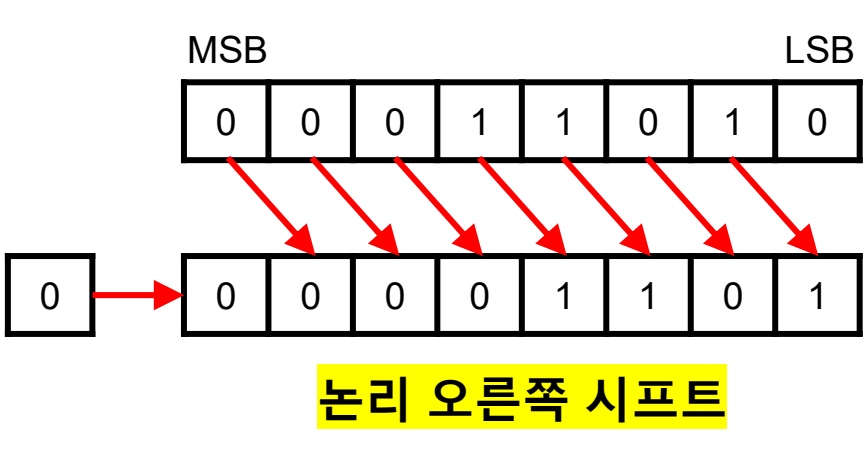
- 양방향 시프트가 가능한 직렬 입력 – 병렬 출력 레지스터
  - $R/\bar{L} = 0$일 경우 $SLI \rightarrow D_D$, $Q_D \rightarrow D_C$, $Q_C \rightarrow D_B$, $Q_B \rightarrow D_A$로 연결됨
  - 왼쪽 시프트 레지스터가 됨



양방향 시프트 레지스터

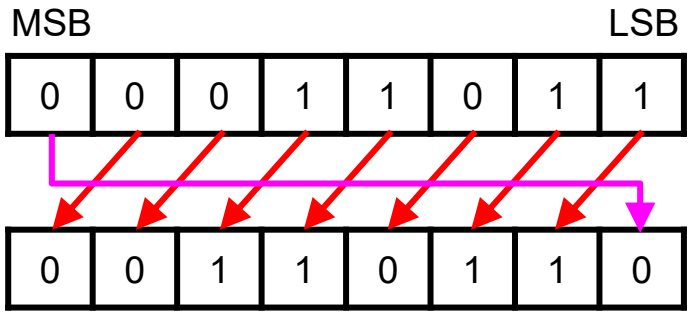
- 시프트 연산
  - 논리 시프트(logical shift)
    - 논리 왼쪽 시프트 : 비트들이 왼쪽으로 한 칸씩 이동하고, 최상위 비트는 버리게 되고, 최하위 비트에는 0이 들어옴
    - 논리 오른쪽 시프트 : 모두 비트들이 오른쪽으로 한 칸씩 이동하게 되고, 최상위 비트에는 0이 들어오고, 최하위 비트는 버리게 됨



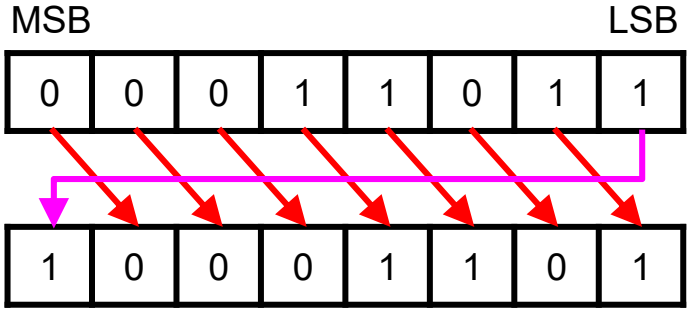


양방향 시프트 레지스터

- 시프트 연산
  - 회전 시프트(rotate shift) 또는 순환 시프트(circular shift)
    - 논리 시프트와 근본적으로 같지만 최상위 비트 또는 최하위 비트를 버리지 않고 반대편 끝에 있는 비트 위치로 들어가게 함



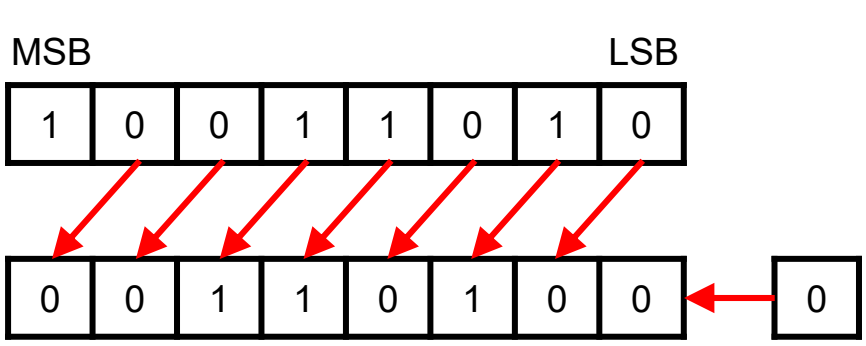
회전 왼쪽 시프트



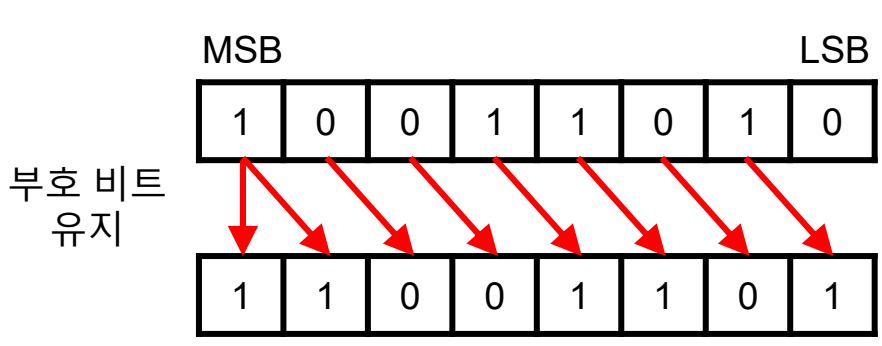
회전 오른쪽 시프트

양방향 시프트 레지스터

- 시프트 연산
  - 산술 시프트(arithmetic shift)
    - 레지스터에 저장된 데이터가 2의 보수 표현될 경우 부호 비트를 고려하여 수행되는 시프트를 말함
    - 산술 왼쪽 시프트는 논리 왼쪽 시프트와 동일하지만, 산술 오른쪽 시프트는 부호 비트를 유지한 채 시프트함



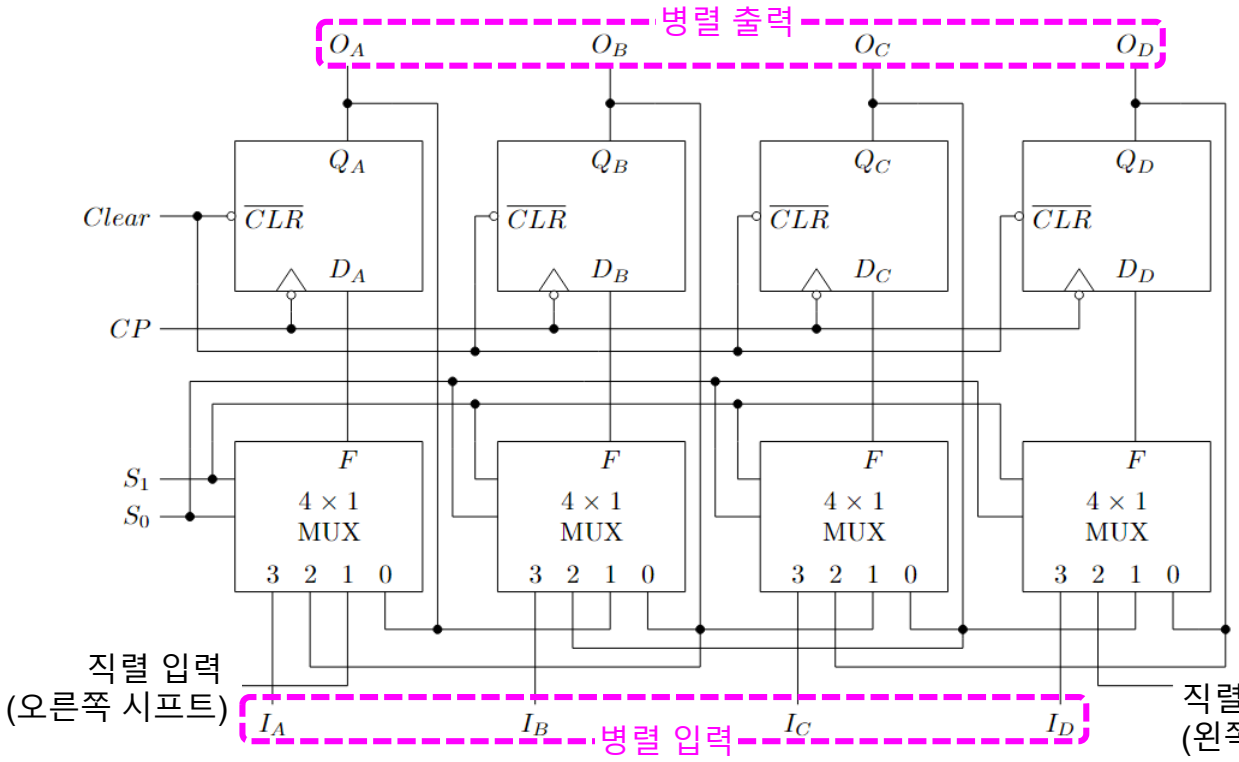
산술 왼쪽 시프트



산술 오른쪽 시프트

범용 시프트 레지스터

- 4비트 범용 시프트 레지스터
  - 직렬 데이터를 병렬로 또는 병렬 데이터를 직렬로 변환함



모드 제어		레지스터 동작
S_1	S_0	
0	0	불편 상태가 됨
0	1	오른쪽 시프트가 수행됨
1	0	왼쪽 시프트가 수행됨
1	1	병렬 입력이 수행됨

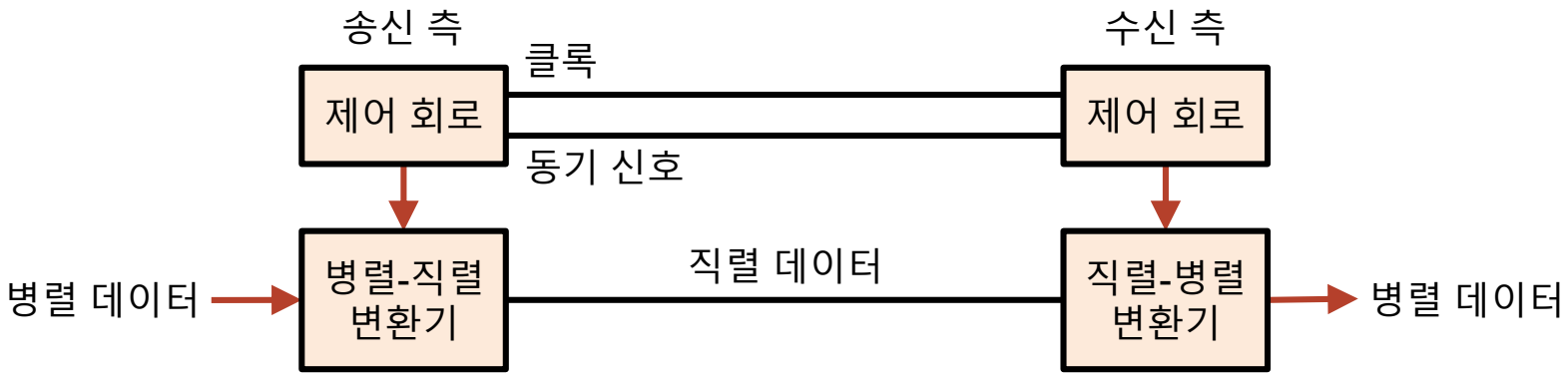
4 × 1 MUX는 입력 4개를 선택적으로 1개만 통과시킴

범용 시프트 레지스터

- 4비트 범용 시프트 레지스터
  - $S_1S_0 = 00$인 경우
    - 각 멀티플렉서 입력 채널 0이 선택되어 레지스터는 현재 출력값이 다시 플립플롭 D 입력에 공급되어 클록 펄스가 입력되어도 현재 상태를 유지함
  - $S_1S_0 = 01$인 경우
    - 각 멀티플렉서 입력 채널 1이 선택되어 각 플립플롭의 출력 Q는 오른쪽 플립플롭의 D 입력에 연결되어 오른쪽 시프트를 수행하며, 맨 왼쪽 플립플롭의 D 입력에는 직렬 데이터가 입력됨
  - $S_1S_0 = 10$인 경우
    - 각 멀티플렉서 입력 채널 2가 선택되어 각 플립플롭의 출력 Q는 왼쪽 플립플롭의 D 입력에 연결되어 왼쪽 스피트를 수행하며, 맨 오른쪽 플립플롭의 D 입력에는 직렬 데이터가 입력됨
  - $S_1S_0 = 11$인 경우
    - 각 멀티플렉서 입력 채널 3이 선택되어 병렬 입력($I_D \sim I_A$)의 2진 데이터는 클록 펄스가 입력될 때 레지스터에 로드(load)됨

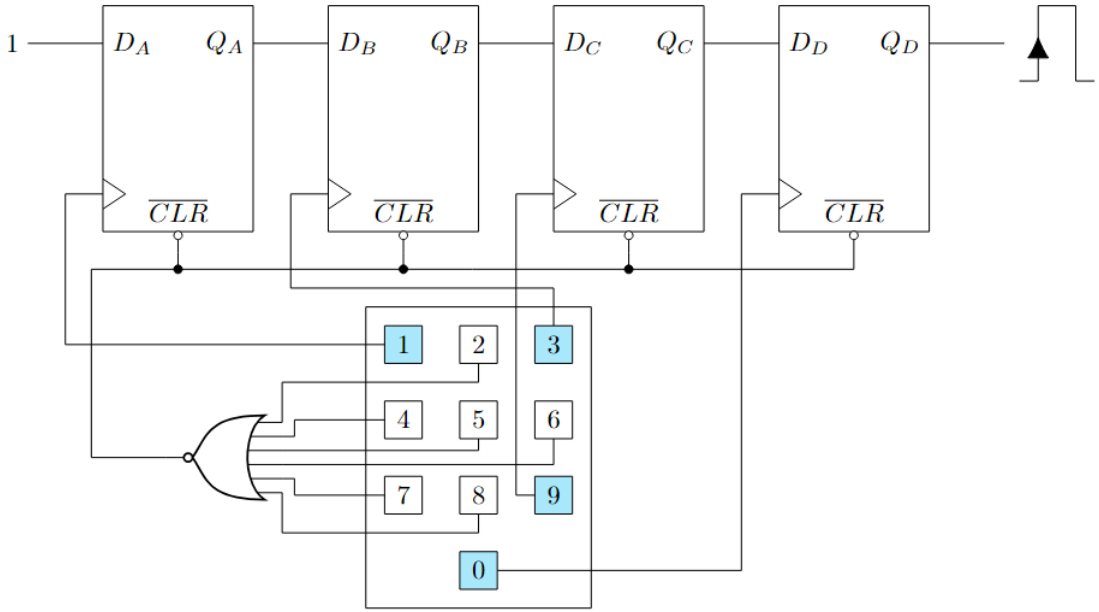
시프트 레지스터 응용

- 직렬 데이터 통신
  - 송신 측
    - ADC를 통해 아날로그 신호를 초당 샘플링(sampling)하여 병렬 데이터로 변환함
    - 병렬 입력 – 직렬 출력 시프트 레지스터를 통해 직렬 데이터로 변환함
  - 수신 측
    - 직렬 입력 – 병렬 출력 시프트 레지스터를 통해 받은 직렬 데이터를 병렬 데이터로 변환함
    - DAC를 통해 원래의 아날로그 신호를 재생함



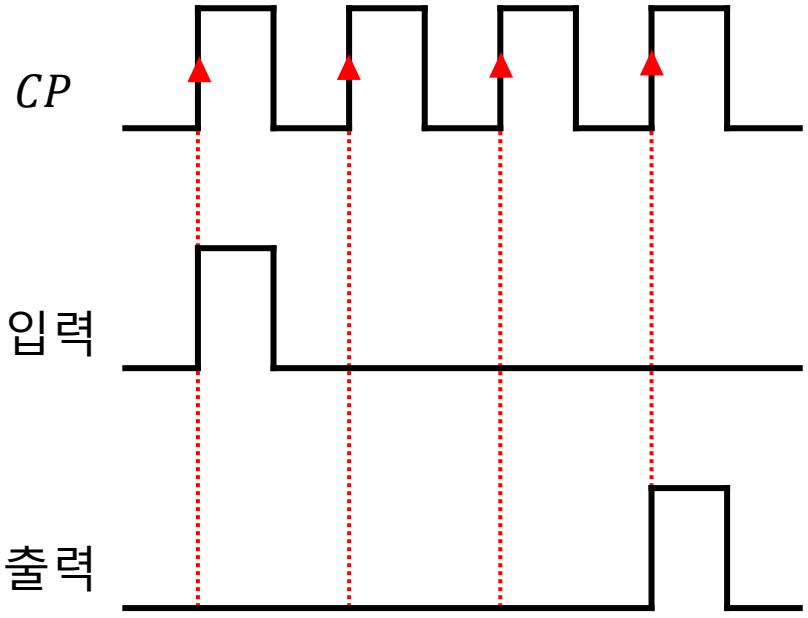
시프트 레지스터 응용

- 디지털 금고
  - 디지털 금고는 미리 정해진 비밀번호를 그 순서대로 키를 누를 때에 한해서 금고가 열리도록 한 것임
  - 예, 비밀번호가 3, 1, 9, 0인 경우



시프트 레지스터 응용

- 시간 지연 회로
  - n비트 직렬 입력 – 직렬 출력 레지스터를 사용하면 입력에 가해진 펄스보다 $(n-1)T$(T는 클록의 주기)만큼 지연되어 출력에서 펄스가 나옴



시프트 레지스터 응용

- 난수 발생 회로(pseudo-random number generator)
  - 랜덤(random) 수열을 발생하는 회로