



이론, 실습, 시뮬레이션 

디지털 논리회로

개정 3판

Chapter 11. 레지스터

학습목표 및 목차

- 4가지 기본형 레지스터의 동작을 이해하고 구분하여 설명할 수 있다.
- 양방향 시프트 레지스터의 동작을 이해하고 활용할 수 있다.
- 레지스터의 주요 응용분야를 이해하고 활용할 수 있다.
- 시프트 레지스터 IC의 외부접속 방법을 이해하고 활용할 수 있다.

01. 레지스터의 분류

02. 직렬입력-직렬출력 레지스터

03. 직렬입력-병렬출력 레지스터

04. 병렬입력-직렬출력 레지스터

05. 병렬입력-병렬출력 레지스터

06. 양방향 시프트 레지스터

07. 범용 시프트 레지스터

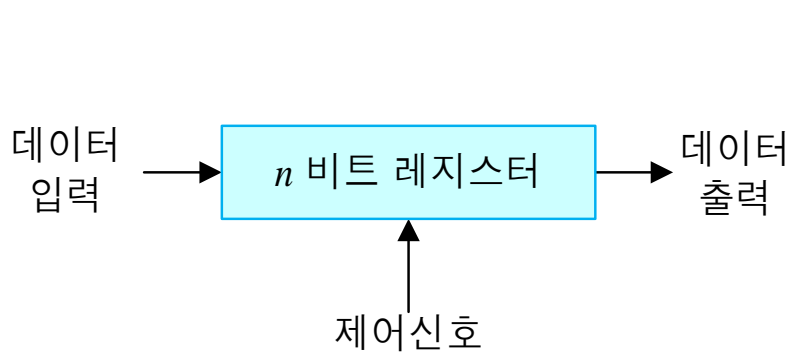
08. 시프트 레지스터의 응용

01 레지스터의 분류

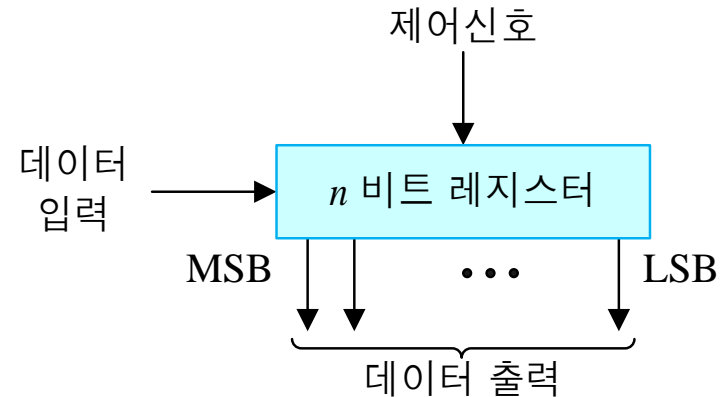
- 레지스터(register): 플립플롭 여러 개를 일렬로 배열하여 적당히 연결함으로써 여러 비트로 구성된 2진수를 저장할 수 있게 한 것
- 레지스터는 외부로부터 들어오는 데이터를 저장하거나 이동하는 목적으로 사용하며, 상태의 순서적인 특성을 갖는 것이 아니다.
- 카운터가 레지스터의 특별한 형태이지만 이름을 달리하여 레지스터와 구별하는 것이 보통이다.
- 레지스터는 다양한 종류의 카운터를 구성하는 데 사용될 뿐만 아니라 여러 비트를 일시적으로 저장하거나 저장된 비트를 좌측으로 또는 우측으로 하나씩 시프트(shift)할 때도 사용된다.
- 레지스터는 CPU 내부에서 연산의 중간 결과를 임시 저장하는 경우나 어떤 2진수의 보수를 구한다든지, 곱셈 또는 나눗셈을 하는 경우에도 사용

01 레지스터의 분류

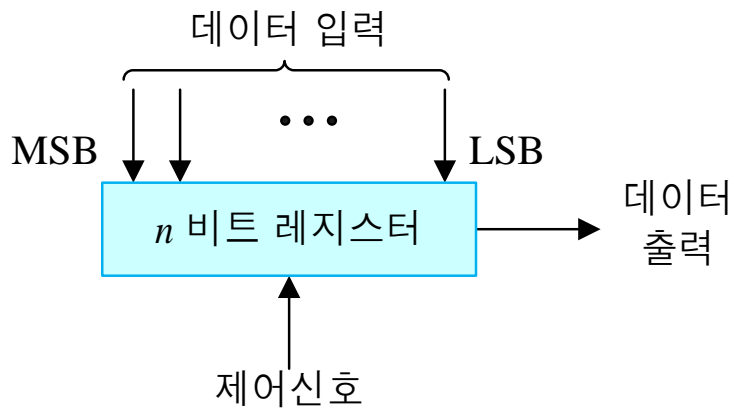
■ 레지스터의 종류



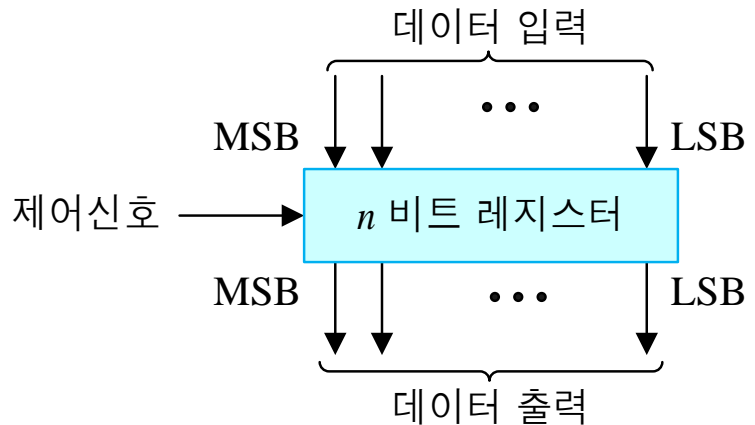
<직렬입력-직렬출력>



<직렬입력-병렬출력>



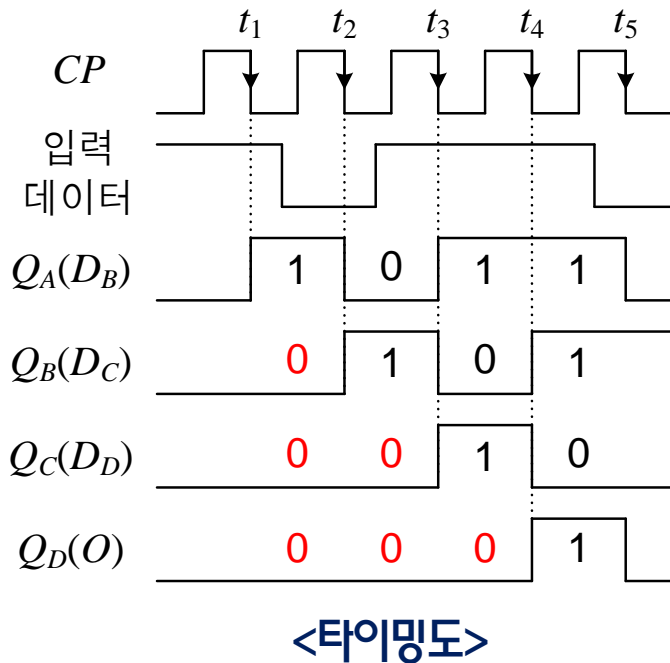
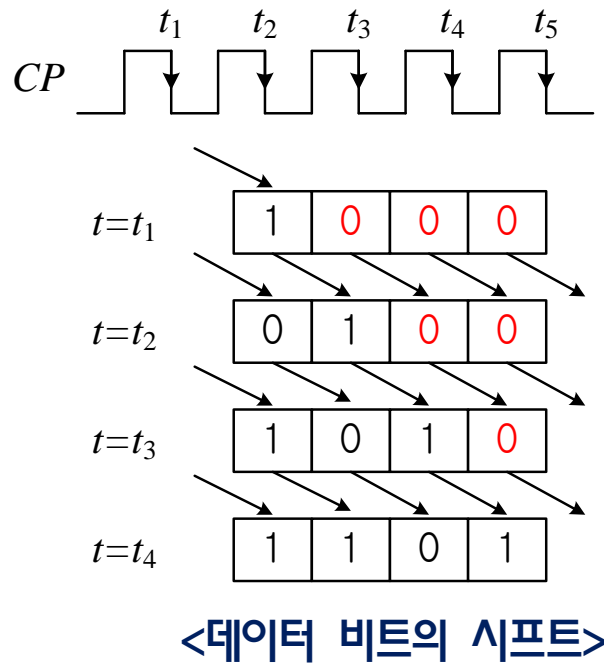
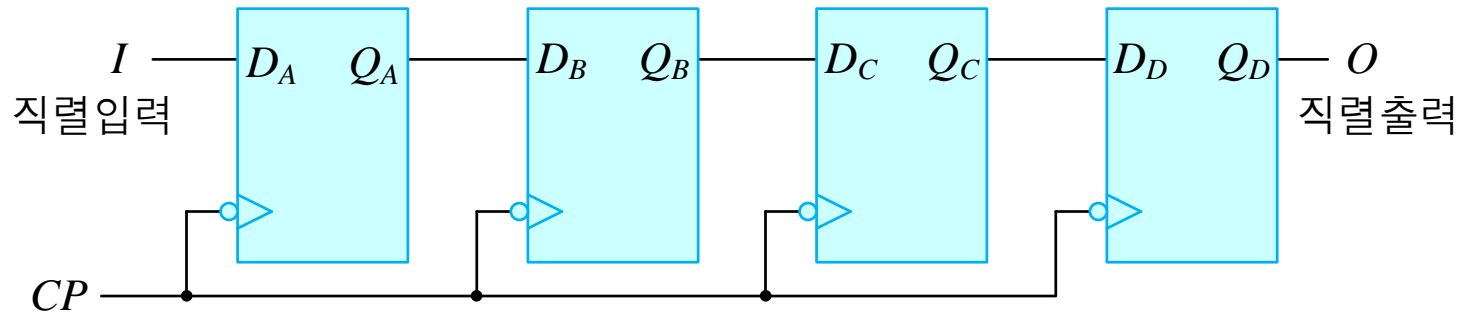
<병렬입력-직렬출력>



<병렬입력-병렬출력>

02 직렬입력-직렬출력 레지스터

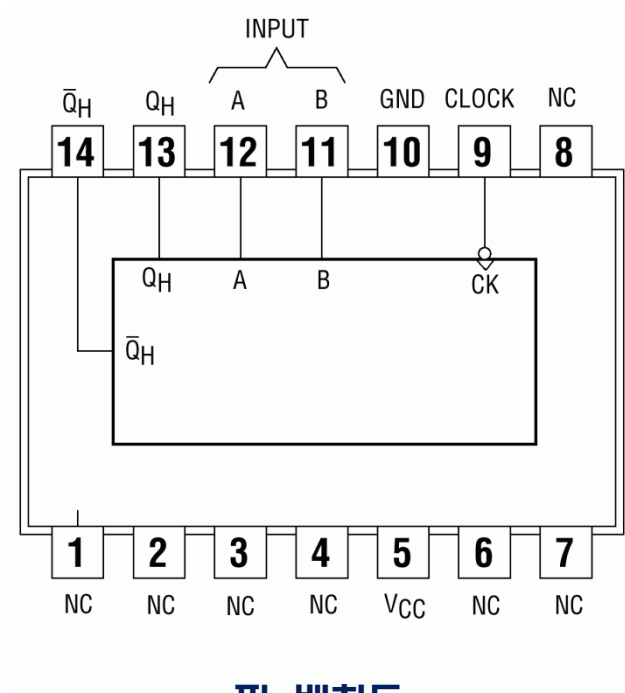
1. 4비트 직렬입력-직렬출력 레지스터 구조



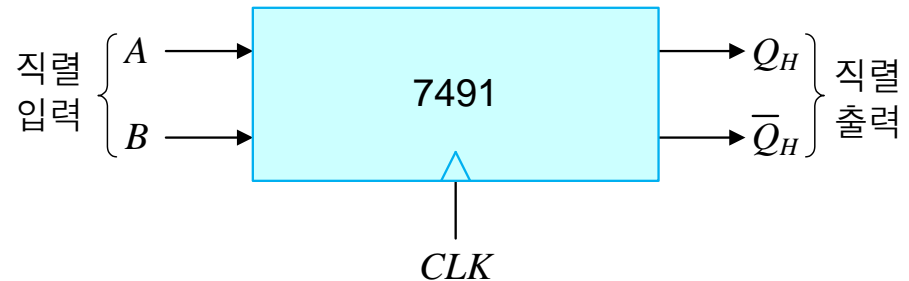
02 직렬입력-직렬출력 레지스터

2. 7491(8-Bit Shift Register)

- 8개의 *SR* 주종형 플립플롭을 직렬로 연결하여 구성한 직렬입력-직렬출력 레지스터
- 직렬 입력단자 : A, B 직렬 출력단자 : Q, \bar{Q}
- 직렬 데이터를 A 로만 받아들이려면 입력단자 B 를 논리 0으로 해야 한다.



<핀 배치도>

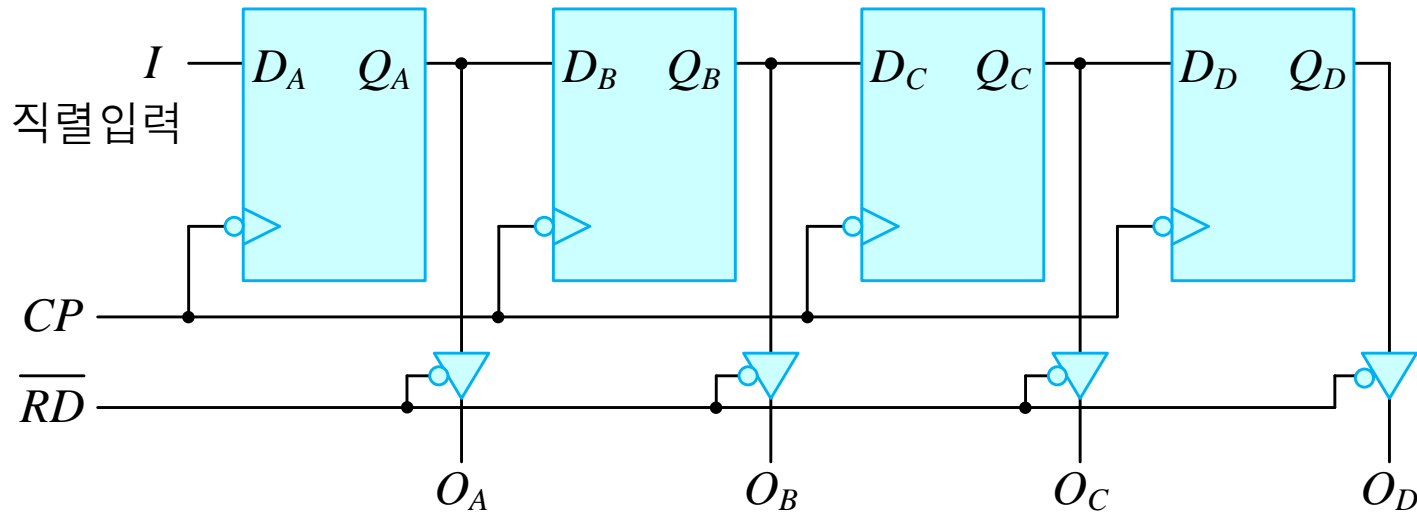


<블록도>

03 직렬입력-병렬출력 레지스터

1. 4비트 직렬입력-병렬출력 레지스터 구조

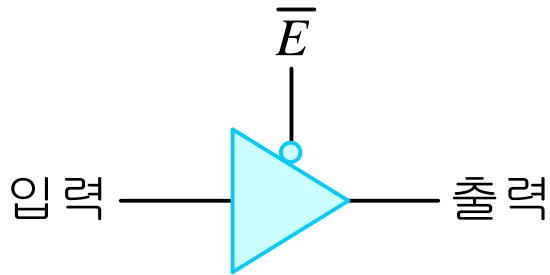
- 레지스터에 저장되어 있는 데이터의 출력은 새로운 4비트 데이터가 레지스터에 차게 되는 4번째 클록펄스, 8번째 클록펄스, 12번째 클록펄스 등에서 출력버퍼를 인에이블($\overline{RD} = 0$)하여 동시에 읽어내면 된다.



03 직렬입력-병렬출력 레지스터

■ 3상태 버퍼

- 출력이 3개 레벨(High, Low, Hi-Z) 중 하나를 가질 수 있다는 사실에서 유래

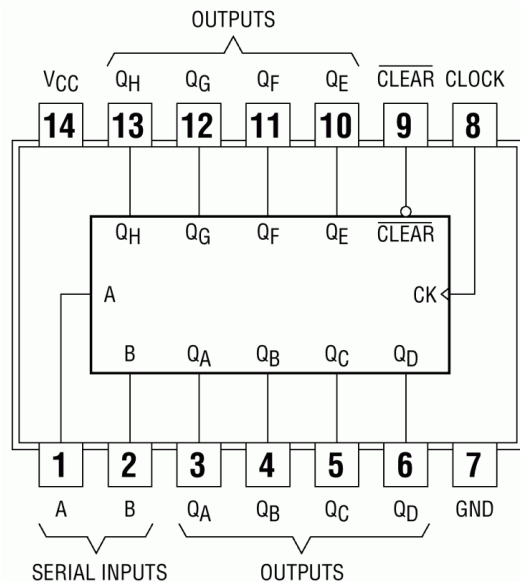


\overline{E}	입력	출력
0	0	0
0	1	1
1	0	Hi-Z
1	1	Hi-Z

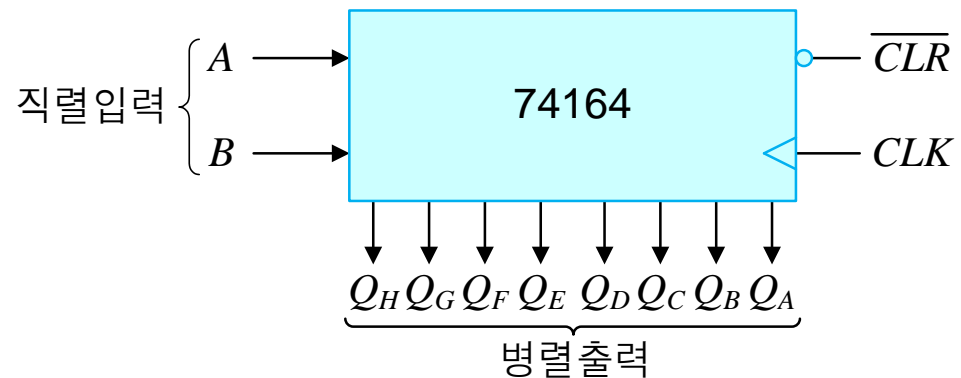
03 직렬입력-병렬출력 레지스터

2. 74164(8-Bit Parallel Output Serial Shift Registers)

- 8개의 *SR* 플립플롭으로 구성된 직렬입력-병렬출력 레지스터
- 직렬 입력단자 : *A*, *B* 병렬 출력단자 : $Q_H \sim Q_A$
- $\overline{CLR} = 0$ 이면, 모든 레지스터의 출력이 Clear.
- $\overline{CLR} = 1$ 이면, 정상동작. 클록의 상승에지마다 입력단자로 들어온 직렬 데이터가 시프트하여 저장



<핀 배치도>



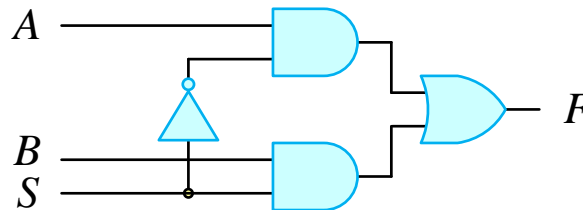
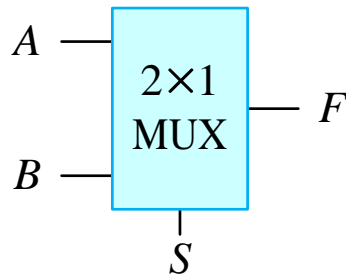
<블록도>

04 병렬입력-직렬출력 레지스터

1. 4비트 병렬입력-직렬출력 레지스터 구조

■ MUX의 동작

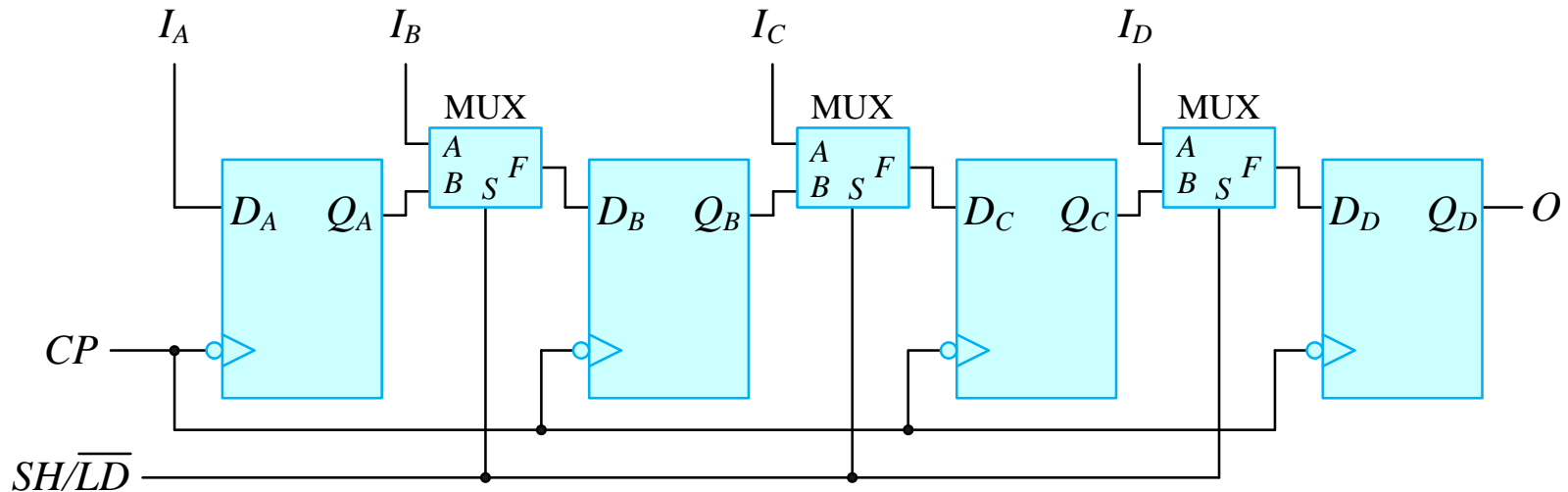
- $S=0$: 입력 A 와 출력 F 가 연결
- $S=1$: 입력 B 와 출력 F 가 연결



04 병렬입력-직렬출력 레지스터

■ 레지스터 동작

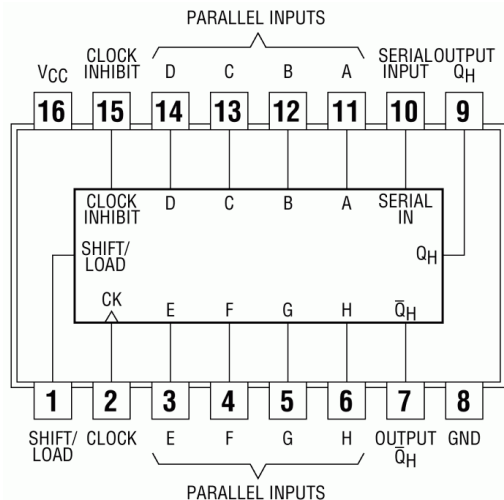
- $SH/\overline{LD}=0$: 입력 데이터 (I_D, I_C, I_B, I_A)이 각 플립플롭의 입력에 각각 연결되므로 클록펄스의 하강에지에서 입력 데이터의 각 비트가 동시에 샘플되어 대응하는 플립플롭의 출력 Q 에 저장
- $SH/\overline{LD}=1$: 클록펄스의 하강에지마다 레지스터 내용이 오른쪽으로 시프트



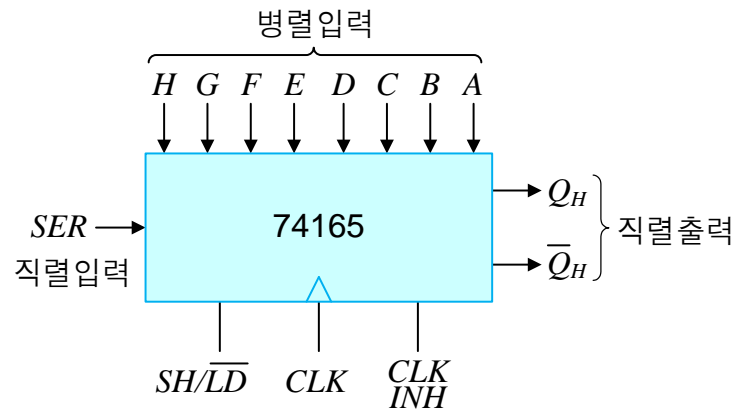
04 병렬입력-직렬출력 레지스터

2. 74165(Parallel Load 8-Bit Shift Registers)

- 8개의 *SR* 플립플롭으로 구성된 병렬입력-직렬출력 레지스터
- 병렬 입력단자 : $H \sim A$, 직렬 출력단자 : Q_H, \bar{Q}_H
- *SER*은 직렬로 시프트할 때 최하위 비트부터 직렬 데이터를 입력하기 위한 단자
- $SH/\bar{LD}=0$ 이면, 병렬 데이터가 입력
- $SH/\bar{LD}=1$ 이면, 시프트 동작
- $CLK\ INH$ 단자가 논리 0이면 클록펄스가 입력



<핀 배치도>

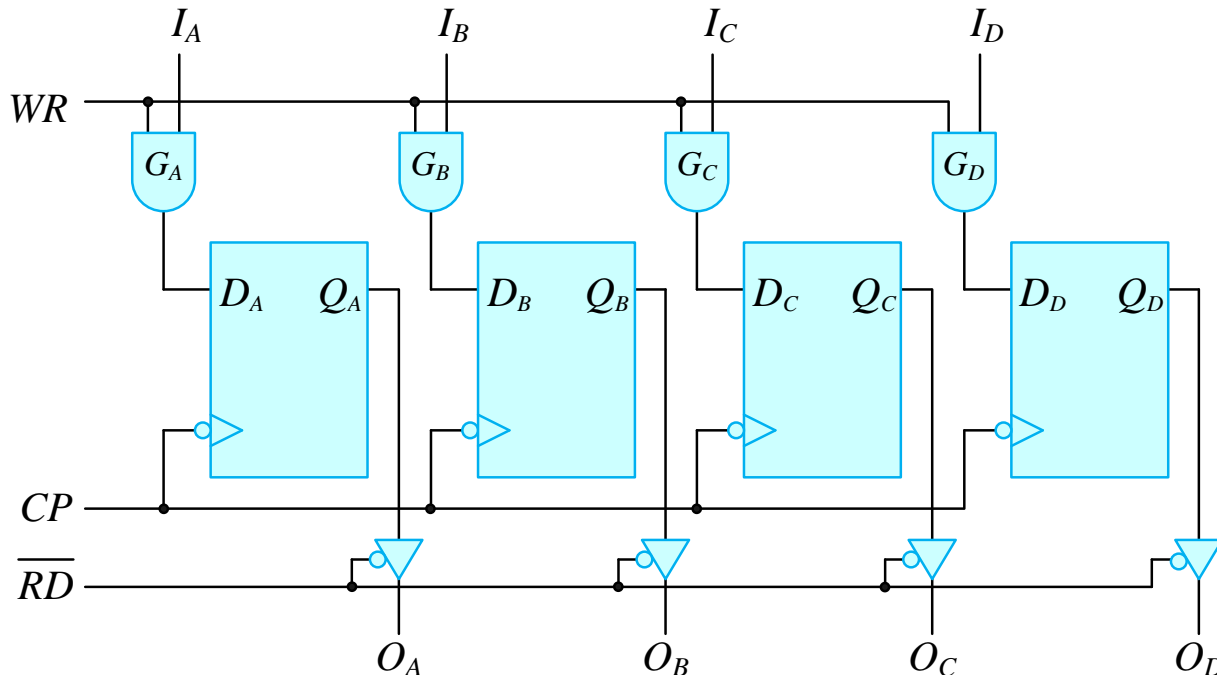


<블록도>

05 병렬입력-병렬출력 레지스터

1. 4비트 병렬입력-병렬출력 레지스터 구조

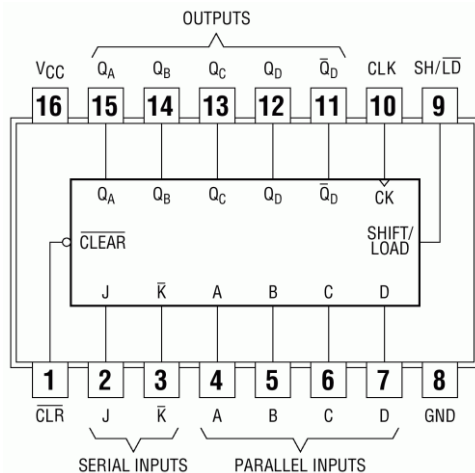
- $WR=1$ 이면 $I_D I_C I_B I_A$ 의 병렬 데이터는 각 AND 게이트를 통하여 동시에 각 플립플롭의 D 입력에 전송.
- $\overline{RD}=0$ 이면 각 플립플롭의 출력 데이터는 버퍼를 통하여 동시에 O_D, O_C, O_B, O_A 에 출력되며, $\overline{RD}=1$ 이면 출력되지 않는다.



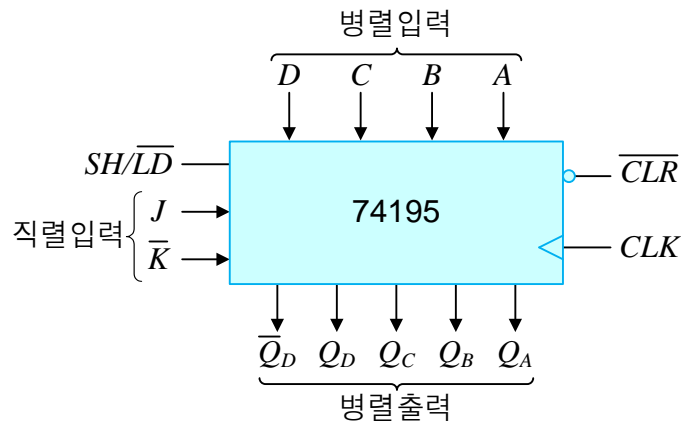
05 병렬입력-병렬출력 레지스터

2. 74195(Parallel Access 4Bit Shift Registers)

- 4비트 병렬입력-병렬출력 기능과 직렬 시프트 기능을 수행하는 레지스터
- 병렬입력-병렬출력 기능 순서
 - ① \overline{CLR} 단자를 논리 1로 한다.
 - ② 병렬입력 단자 D, C, B, A에 데이터를 병렬로 입력한다.
 - ③ $SH/\overline{LD}=0$ 으로 하여 레지스터에 로드한다.
 - ④ CLK 단자에 클록을 입력하면 상승에지에서 동작한다.
 - ⑤ 입력된 데이터는 레지스터에 로드되고 출력단자 Q_D, Q_C, Q_B, Q_A 로 데이터가 출력된다.



<핀 배치도>

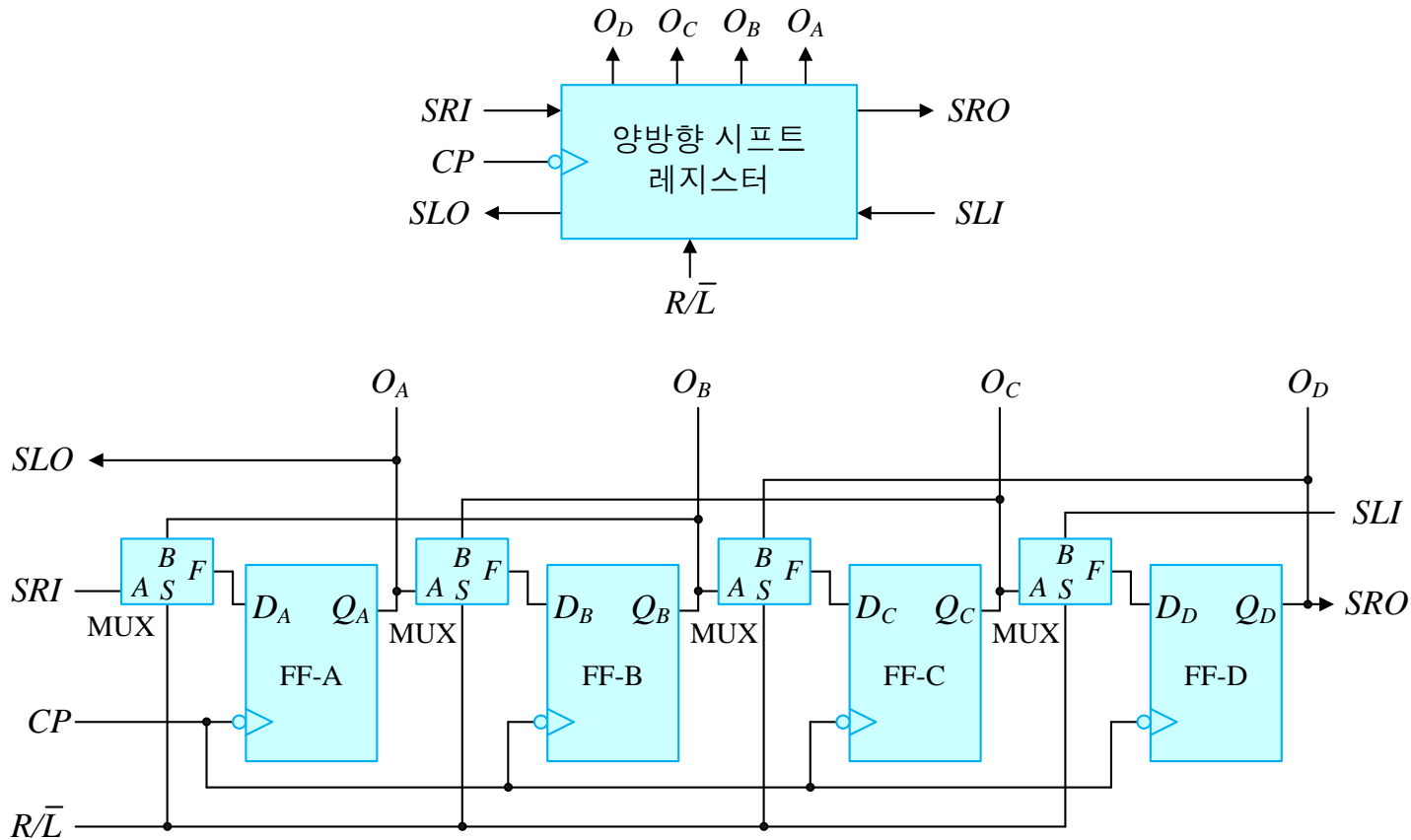


<블록도>

06 양방향 시프트 레지스터

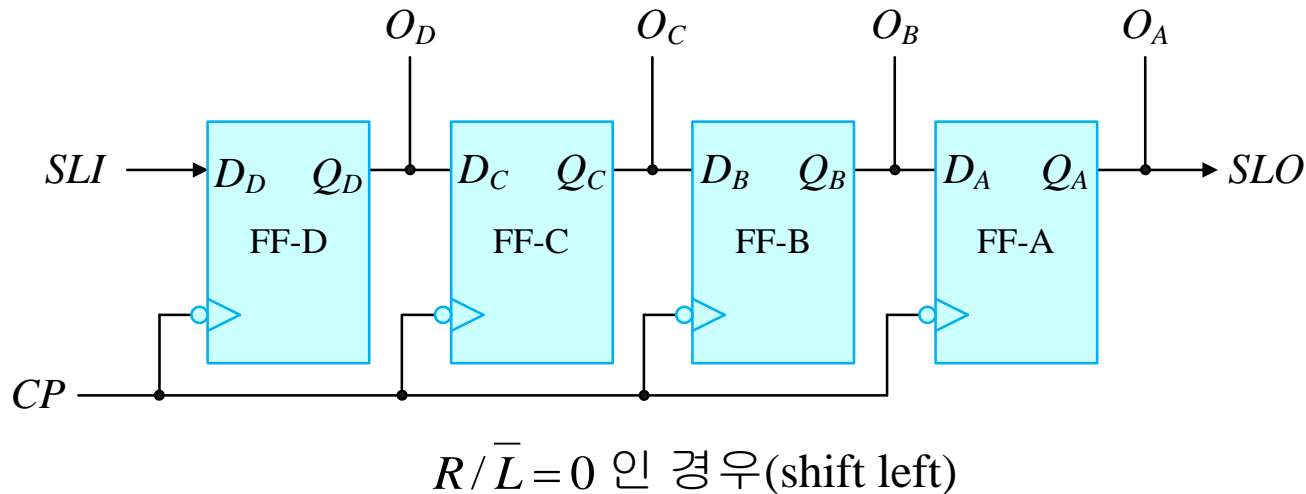
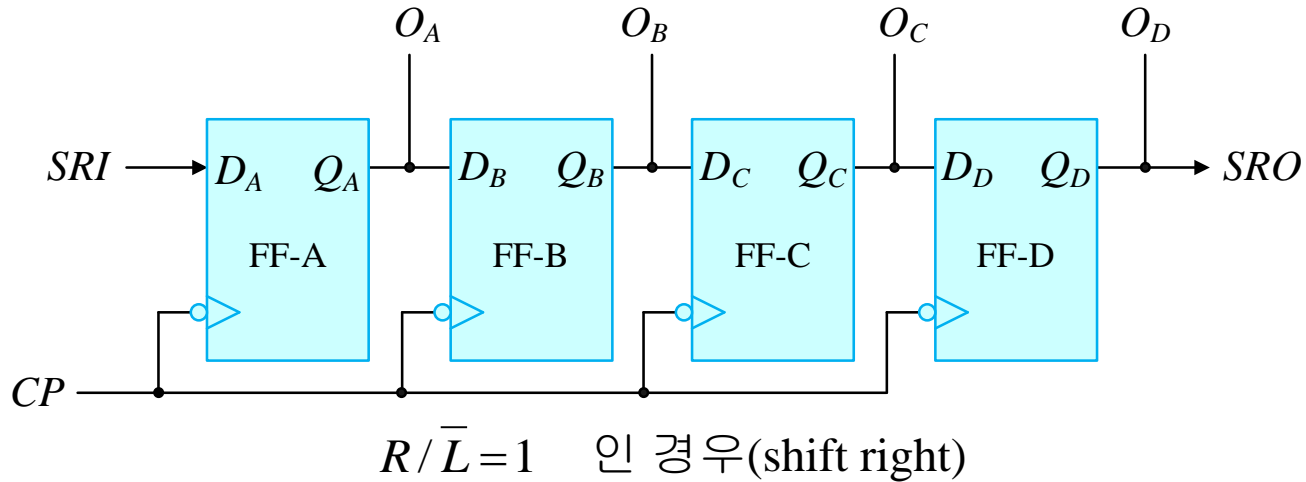
■ 레지스터 동작

- $R/\bar{L}=1$: 데이터를 SRI 에 입력시켜 오른쪽으로 시프트하면서 SRO 에서 출력
- $R/\bar{L}=0$: 데이터를 SLI 에 입력시켜 왼쪽으로 시프트하면서 SLO 에서 출력



06 양방향 시프트 레지스터

■ 제어 입력에 따른 쌍방향 시프트 레지스터 동작



07 범용 시프트 레지스터

1. 4비트 범용 시프트 레지스터 구조

■ 범용 시프트 레지스터 기능

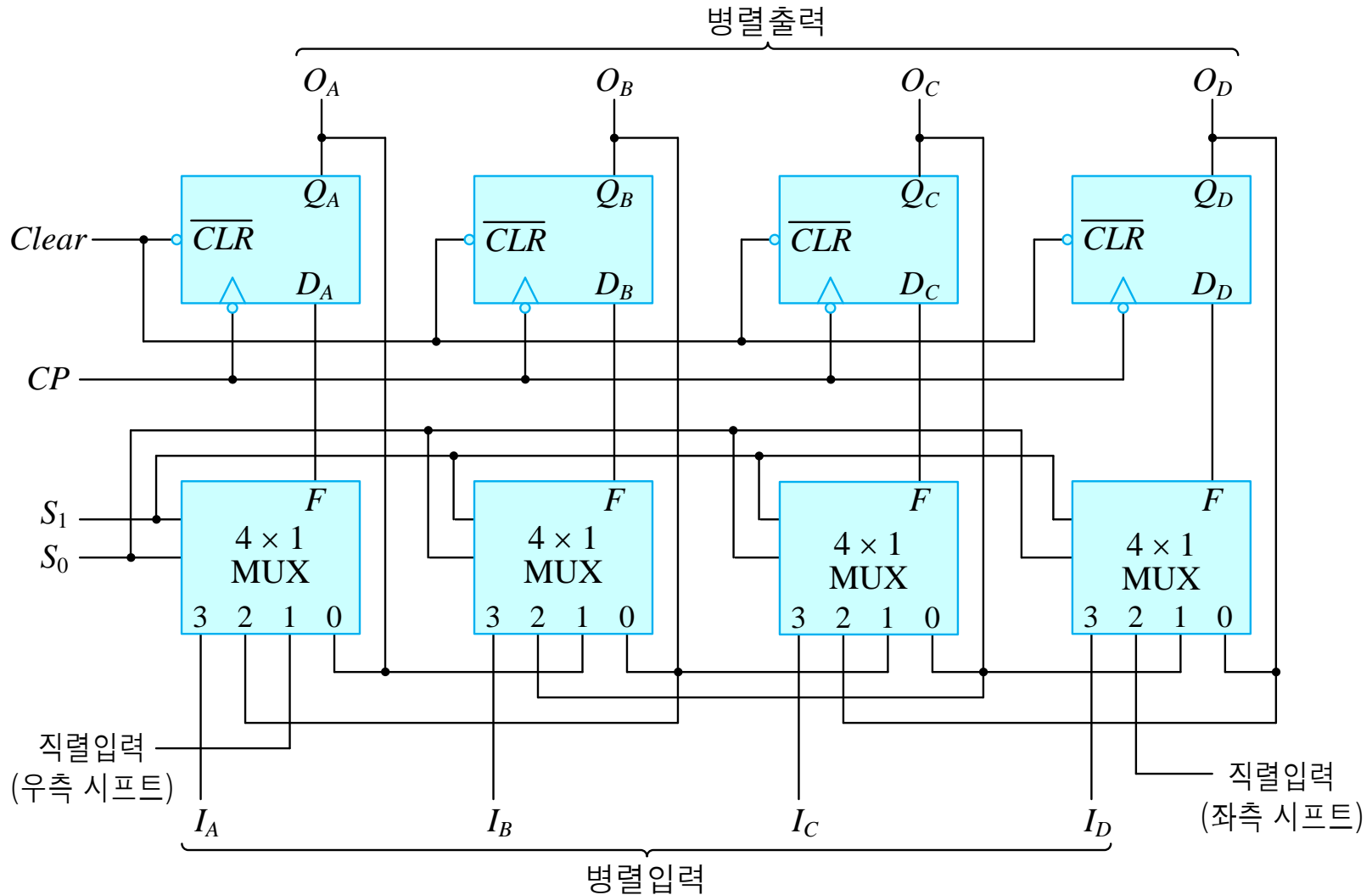
- 클리어 제어 입력신호에 의해 레지스터를 Clear할 수 있어야 한다.
- 모든 동작을 동기화 시키는 클록펄스를 입력시킬 수 있어야 한다.
- 자리 이동 제어 입력이 있어서 오른쪽 시프트와 왼쪽 시프트 및 직렬 입력과 직렬 출력을 수행
- 병렬 로드 입력신호에 의하여 병렬 전송과 병렬 입력을 수행
- 클록펄스에 관계없이 레지스터에 저장된 데이터를 변화 없이 유지

■ 범용 시프트 레지스터의 제어표

모드 제어		레지스터 동작
S_1	S_0	
0	0	불변 상태가 된다.
0	1	오른쪽 자리 이동이 수행된다.
1	0	왼쪽 자리 이동이 수행된다.
1	1	병렬 입력이 수행된다.

07 범용 시프트 레지스터

■ 범용 시프트 레지스터 회로



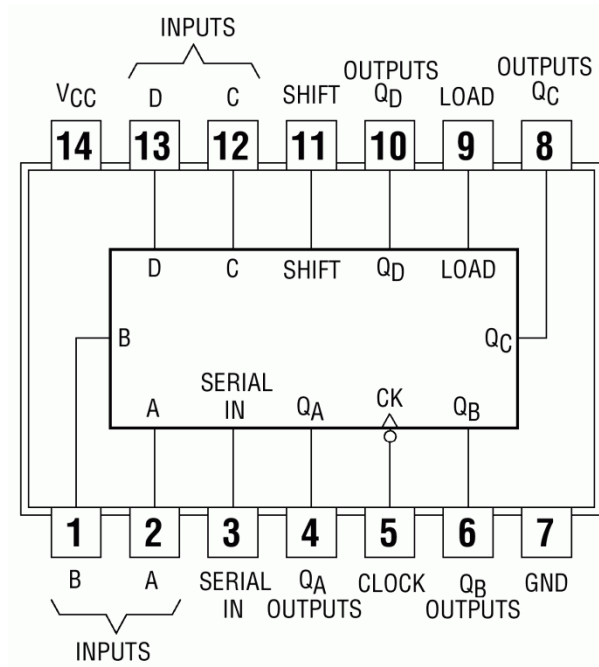
07 범용 시프트 레지스터

2. 74178(4Bit Parallel Access Shift Registers)

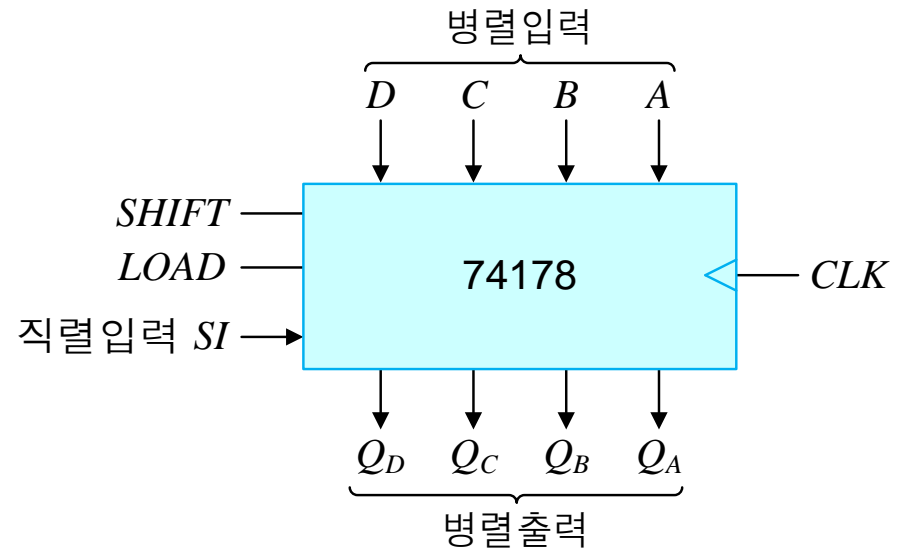
- PIPO, SISO, PISO, SIPO이 가능한 4비트 레지스터
- 동작상태
 - 병렬입력 : $SHIFT=0, LOAD=1$ 으로 하면 클록펄스의 하강에지에서 병렬데이터 $DCBA$ 가 들어간다.
 - 직렬입력 : $SHIFT=1, LOAD=x$ 로 하고 직렬데이터를 MSB부터 SI를 통하여 입력한다. 4개의 클록펄스 입력 후 4비트가 채워진다.
 - 직렬출력 : 저장된 직렬데이터는 클록펄스의 하강에지에서 MSB부터 Q_D 에서 출력된다.
 - 병렬출력 : 언제나 Q_D, Q_C, Q_B, Q_A 에서 동시에 출력될 수 있다.

입력		다음 상태				기능
$SHIFT$	$LOAD$	Q_D	Q_C	Q_B	Q_A	
0	0	Q_D	Q_C	Q_B	Q_A	불변 상태
0	1	D	C	B	A	병렬 입력 수행
1	x	SI	Q_D	Q_C	Q_B	오른쪽 시프트 수행

07 범용 시프트 레지스터



<핀 배치도>



<블록도>

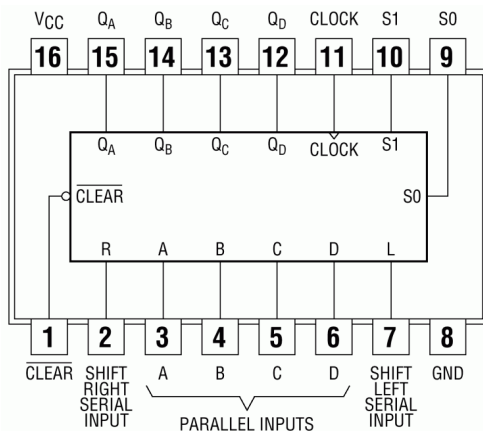
07 범용 시프트 레지스터

3. 74194(4Bit Bidirectional Universal Shift Registers)

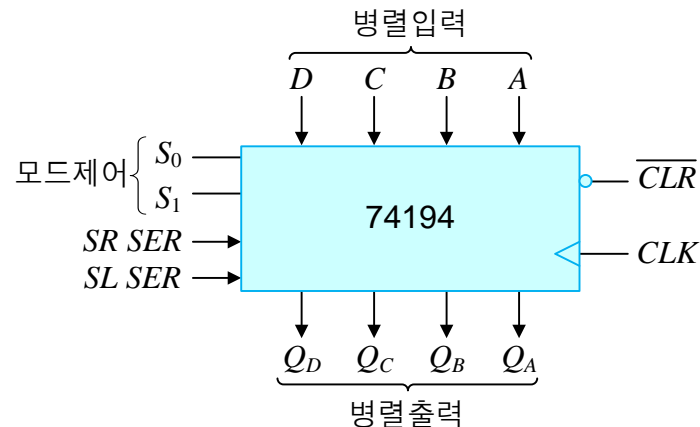
- PIPO, SISO, PISO, SIPO이 가능한 4비트 레지스터

모드 제어		기능
S_1	S_0	
0	0	불변 상태
0	1	우측 시프트 수행
1	0	좌측 시프트 수행
1	1	병렬 입력 수행

<제어표>



<핀 배치도>

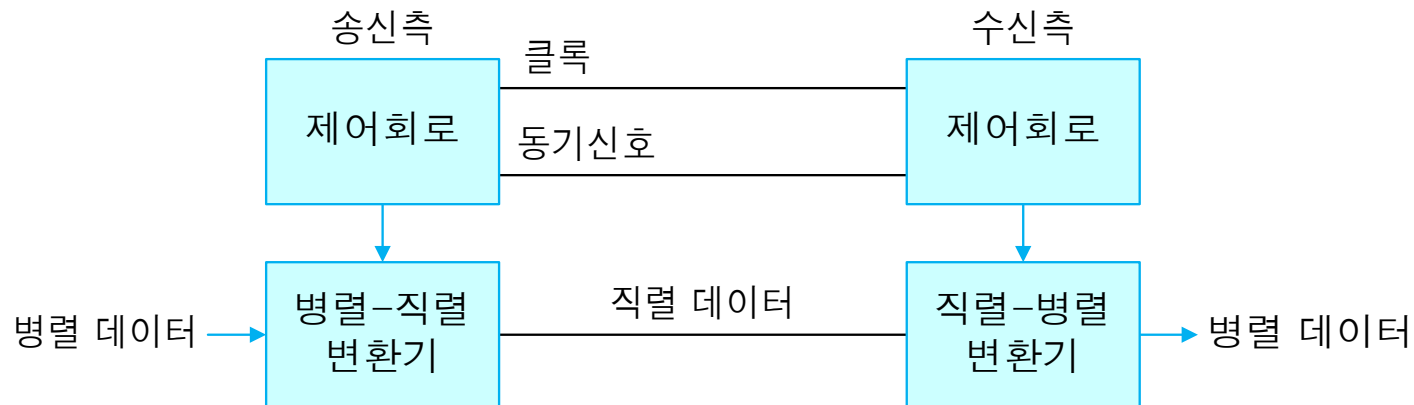


<블록도>

08 시프트 레지스터의 응용

1. 직렬 데이터 통신

- 시프트 레지스터는 음성통신을 위한 시스템에서 광범위하게 사용
- 전자 교환기는 각 전화가입자의 아날로그 음성신호를 ADC(Analog to Digital Converter)를 통하여 디지털 신호로 변환
- ADC는 입력 아날로그 신호를 매초 8000번 샘플링(sampling)하여 8비트 병렬데이터로 변환($8000 \times 8 = 64\text{Kbps}$)
- 이것은 다시 병렬입력-직렬출력 시프트 레지스터를 통해서 직렬데이터로 변환
- 중계선(trunk)의 전송방식에는 T1 방식과 E1 방식이 있음



08 시프트 레지스터의 응용

- 중계선의 전송속도

T1 방식

$$24 \times 64\text{Kbps} + 8\text{Kbps} = 1544\text{Kbps} = 1.544\text{Mbps}$$

E1 방식

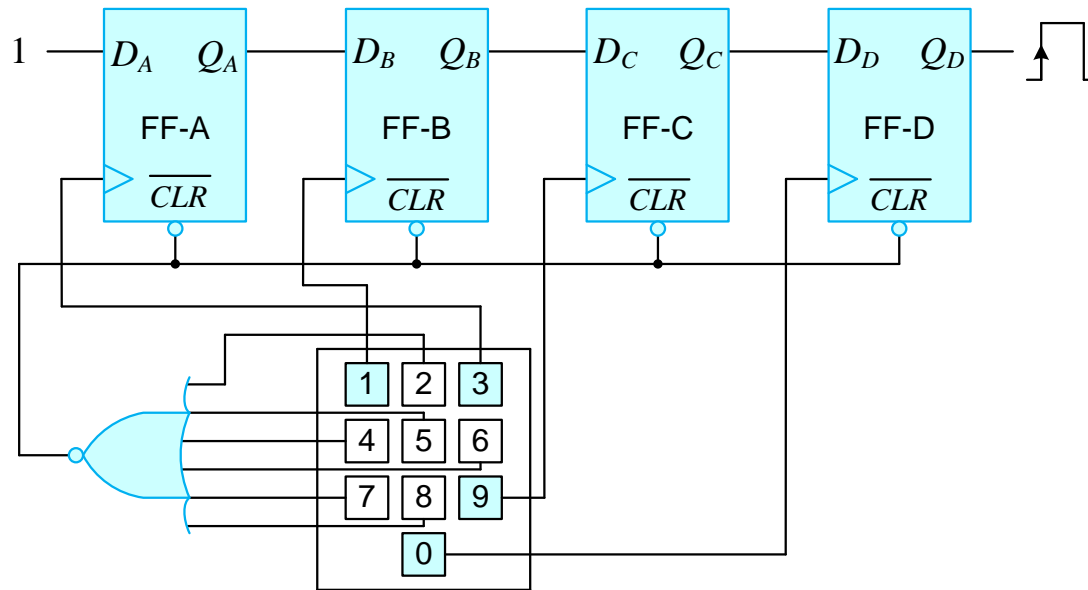
$$32 \times 64\text{Kbps} = 2048\text{Kbps} = 2.048\text{Mbps}$$

- 수신측의 전자교환기에서는 이 직렬데이터를 직렬입력-병렬출력 시프트 레지스터로 병렬 데이터로 변환
- 24채널(또는 32채널)로 디멀티플렉스하고 각 채널의 8비트 병렬데이터를 64kHz의 DAC(Digital to Analog Converter)에 의하여 원래의 아날로그 신호를 재생
- 비트 전송의 타이밍 기준을 제공하기 위한 클록을 보내는 선과 또 직렬데이터의 형태(format)를 정의하기 위한 동기신호를 보내는 선이 필요

08 시프트 레지스터의 응용

2. 디지털 금고

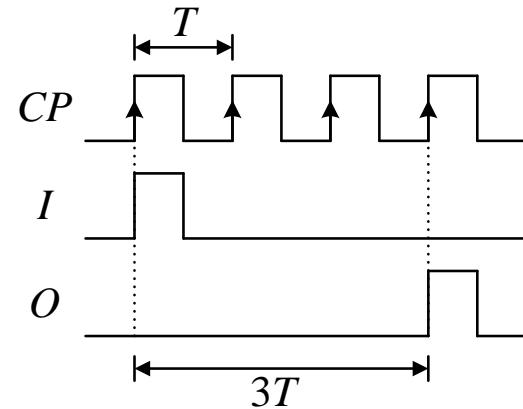
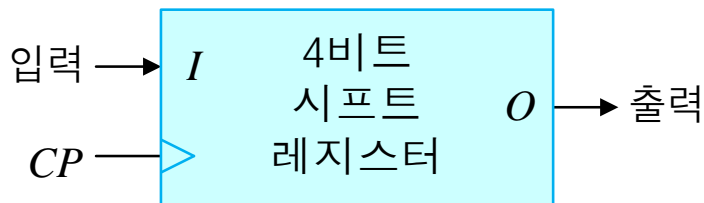
- 비밀번호가 "3, 1, 9, 0"인 경우를 가정
- 키 패드상의 키 3, 1, 9, 0은 각 플립플롭의 클럭입력에 연결
- 기타 키들은 NOR 게이트의 입력에 연결
- 비밀번호를 순서적으로 누르면 각 데이터가 오른쪽으로 시프트
- 마지막 키 0을 누르면 $Q_D=1$ 이 되어서 금고문이 열림



08 시프트 레지스터의 응용

3. 시간 지연회로

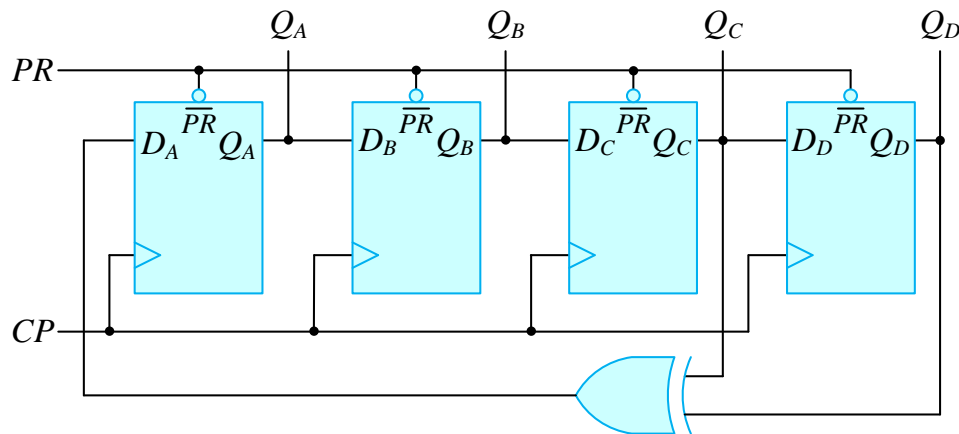
- n 비트 직렬입력-직렬출력 레지스터를 사용하면 입력에 가해진 펄스보다 $(n-1)T$ (T 는 클록의 주기)만큼 지연되어 출력에서 펄스가 나온다.
- 4비트 레지스터를 쓴 경우, 클록 주파수가 1MHz이면 $T=1\mu s$ ($=1/10^6$), 따라서 $3\mu s$ 지연되어 펄스가 나온다.
- 시간지연(time delay)을 더욱 증가하려면 레지스터를 필요한 개수만큼 직렬연결하고, 클록펄스를 공통으로 사용



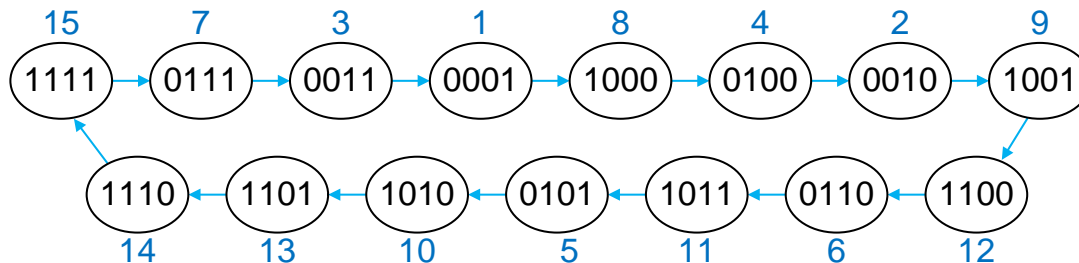
08 시프트 레지스터의 응용

4. 난수발생회로

- 임의의 랜덤(random)한 수열을 발생하는 회로
- $\overline{PR} = 0$ 후, $\overline{PR} = 1$ 하면, $Q_A Q_B Q_C Q_D = 1 1 1 1$
- 펄스를 입력함에 따라 상태도와 같이 동작



<회로도>



<상태도>

*초기상태는 0000 이외의 어떤 상태도 가능



감사합니다 ☺