

제9장 순차논리회로

Youtube 주소

[9-1] <http://youtu.be/1bDzGSYPYQ8>

[9-2] <http://youtu.be/N99gbaZJ2Zs>

[9-3] <http://youtu.be/Srbaj-6QNXI>

[9-1(추가)] <http://youtu.be/3NXkSSAP3L4>

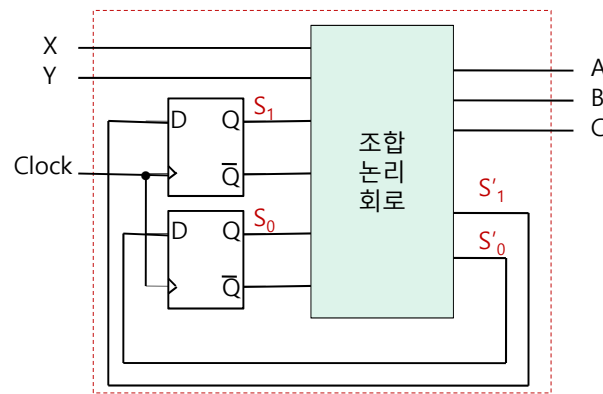


순차논리회로

- 조합논리회로
 - ✓ 입력 값의 조합에 따라서 출력 값 결정
- 순차논리회로
 - ✓ 회로 내부에 현재의 상태를 나타내는 값이 있음
 - ✓ 입력 값과 내부의 현재 상태 값에 따라서 출력 값과 다음상태를 결정
 - ✓ 조합논리회로가 순차논리회로의 일부분으로 사용됨
- 현재 상태 값의 저장
 - ✓ D 플립플롭 (flip-flop) 활용



순차논리회로의 구성 예

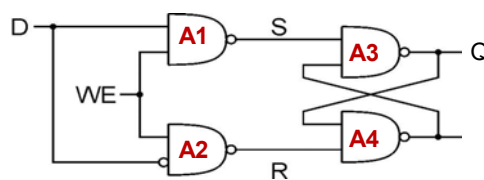


2019/11/5

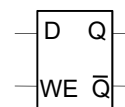
9-3

D 래치 (Latch)

- D Latch: 1비트의 상태나 값을 저장하는 장치
- WE (Write Enable): 래치에 새로운 값을 기록하려면 활성화



회로 구성

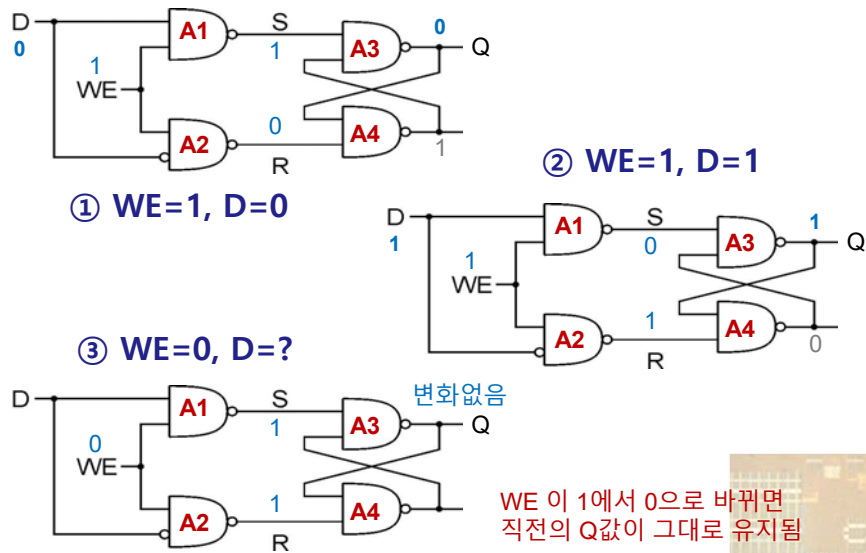


기호

2019/11/5

9-4

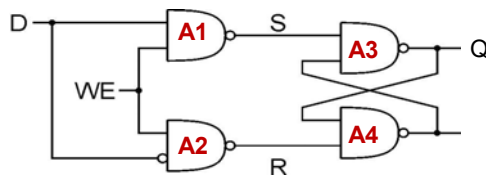
D Latch의 동작



2019/11/5

9-5

D Latch의 진리표



WE	D	S	R	Q
0	0	1	1	Q
0	1	1	1	Q
1	0	1	0	0
1	1	0	1	1

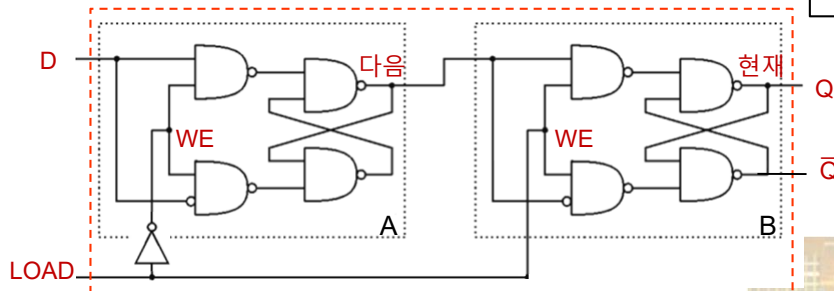
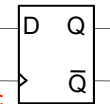
2019/11/5

9-6

D 플립플롭 (Flip-Flop)

- 출력과 입력을 시간적으로 분리
 ← Flip-flop의 출력이 조합논리회로를 거쳐서 다시 입력으로 연결되는 경우를 대비해서
- LOAD가 0에서 1로 변경되는 시점에 Q 값 변경

표시기호

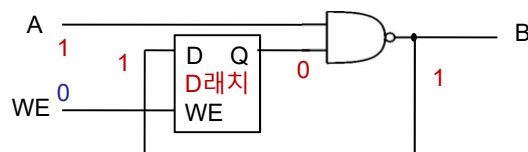


2019/11/5

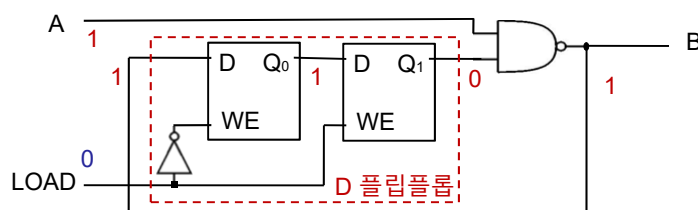
9-7

D Latch의 출력이 입력으로 연결되면?

D Latch 적용: WE=1이면 출력 B는?



D Flip-Flop 적용



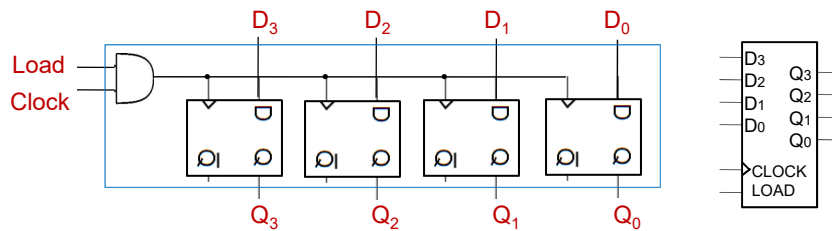
2019/11/5

9-8

레지스터 (Register)

- 프로세서 내부에 데이터를 임시로 저장하는 장치
- TOY: R0 ~ R6, PC 는 모두 32비트 레지스터
- D flip-flop들을 병렬로 연결하여 구성

표시기호



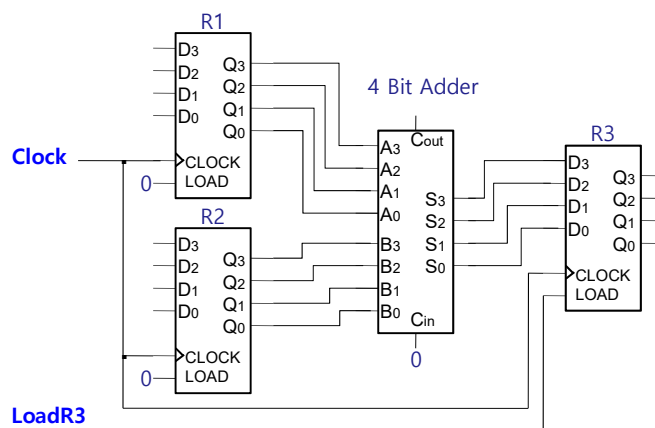
4비트 레지스터 회로

2019/11/5

9-9

레지스터 간의 ADD 연산처리

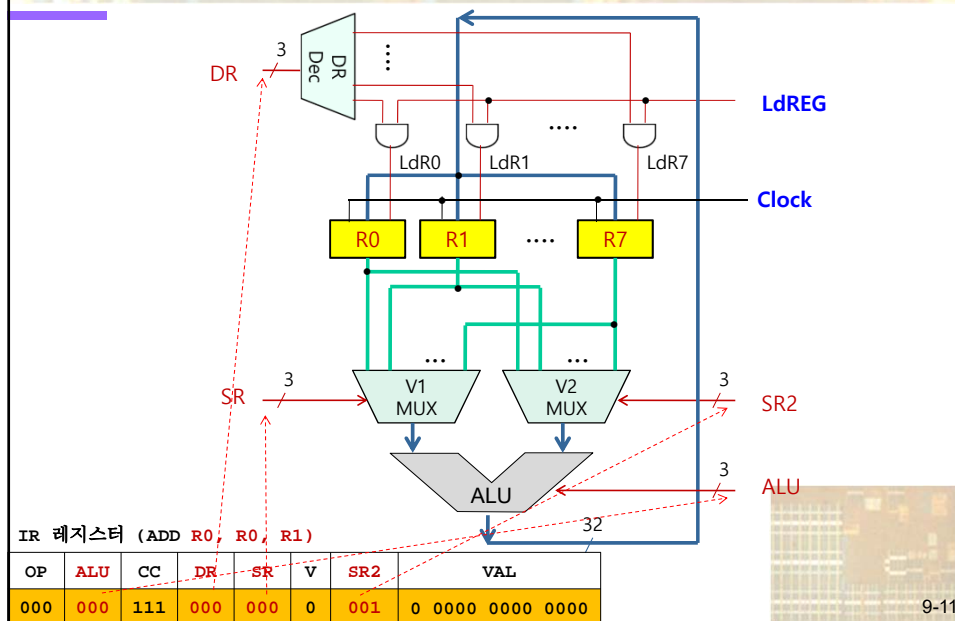
- R1의 값과 R2의 값을 더해서 R3에 저장하기
- ADD R3, R1, R2



2019/11/5

9-10

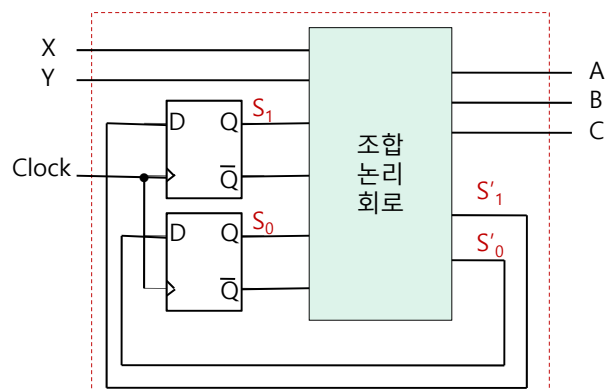
레지스터 간의 산술/논리 연산 회로



9-11

순차논리회로의 구성

- 외부입력: X, Y
- 클럭입력: Clock
- 외부출력: A, B, C
- 다음상태: S'1, S'0 (D 플립플롭의 입력으로 연결)



2019/11/5

9-12

순차논리회로의 설계 과정

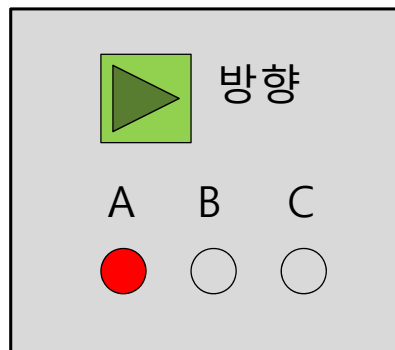
1. 구별해야 할 상태들과, 상태별 출력의 정의
2. 상태 간의 천이 조건 정의 (상태 천이도)
3. 상태별 고유 번호 부여, 번호 표현에 필요한 비트 수 만큼의 D 플립플롭 사용 (남는 상태들은 정상적인 상태 중의 하나로 천이하도록 처리 ← 모든 가능한 상태들을 포함해야)
4. 입력과 현재 상태에 따른 출력과 다음상태의 진리표 작성
5. 출력 신호와 다음 상태를 위한 조합논리회로 작성, 다음 상태는 D 플립플롭의 입력으로 연결

2019/11/5

9-13

교통안내 신호등 설계

- 방향스위치에 따른 불빛 흐름 안내
 - ✓ 0 이면 오른쪽으로 흐르고
 - ✓ 1 이면 왼쪽으로 흐른다.

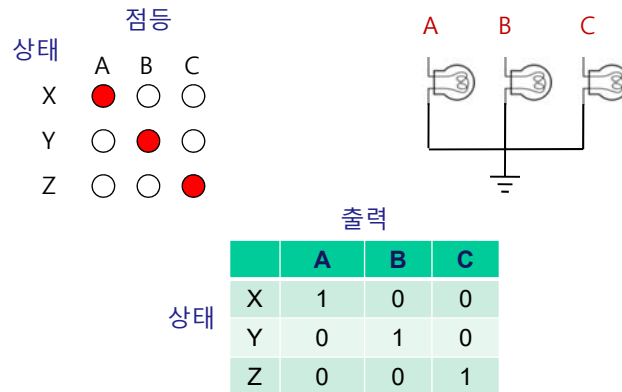


2019/11/5

9-14

1. 상태 및 출력 정의

- 구별해야 할 상태들과, 상태별 출력의 정의
 - ✓ 3가지의 상태가 있으며 각각 특정 1개 등을 켜다.
 - ✓ (참고) 이 예에서는 출력이 입력과 상관없이 상태에 따라 결정됨

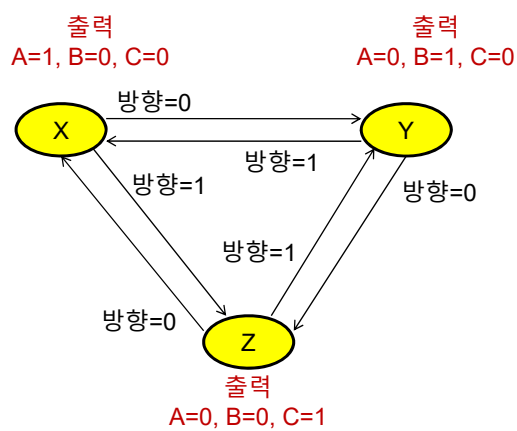


2019/11/5

9-15

2. 상태간의 천이 조건 정의

- 상태천이도 작성



2019/11/5

9-16

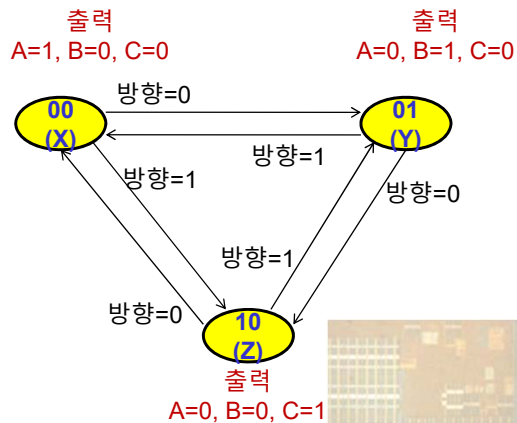
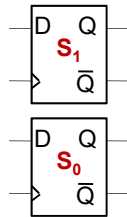
3. 상태별 고유번호 부여

- 상태별 고유 번호 부여
 - 번호 표현에 필요한 비트 수 만큼의 D 플립플롭 사용
- 2개의 D 플립플롭

✓ X: 00

✓ Y: 01

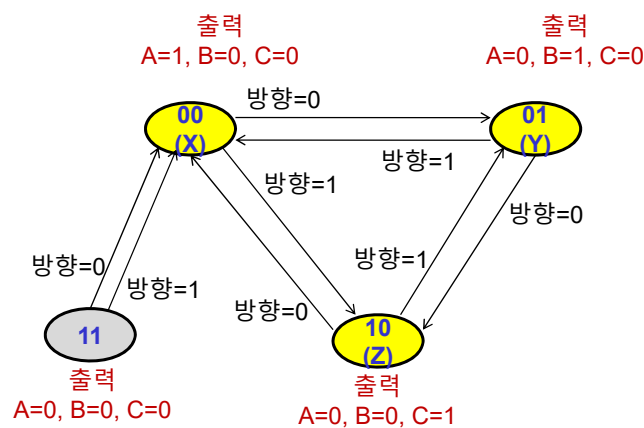
✓ Z: 10



2019/11/5

9-17

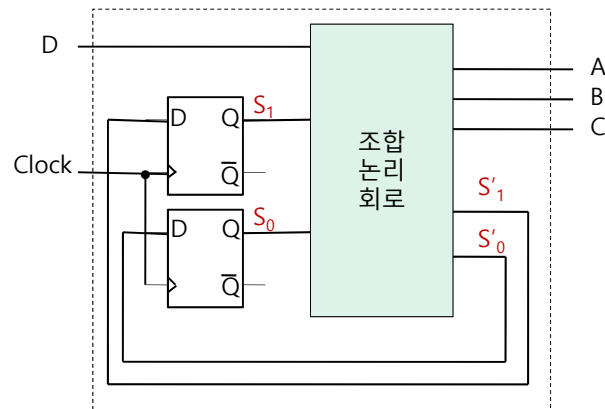
모든 상태를 포함한 상태 천이도



2019/11/5

9-18

전체 회로 구성



2019/11/5

9-19

4. 진리표 작성

입력		출력			
방향 (D)	상태 (S ₁ S ₀)	다음상태 (S' ₁ S' ₀)	A	B	C
0	0 0	0 1	1	0	0
0	0 1	1 0	0	1	0
0	1 0	0 0	0	0	1
0	1 1	0 0	0	0	0
1	0 0	1 0	1	0	0
1	0 1	0 0	0	1	0
1	1 0	0 1	0	0	1
1	1 1	0 0	0	0	0

$$A = \text{AND}(\text{NOT}(S_1), \text{NOT}(S_0))$$

$$B = \text{AND}(\text{NOT}(S_1), S_0)$$

$$C = \text{AND}(S_1, \text{NOT}(S_0))$$

$$S'_0 = \text{OR}(\text{AND}(\text{NOT}(D), \text{NOT}(S_1)), \text{NOT}(S_0), \text{AND}(D, S_1, \text{NOT}(S_0)))$$

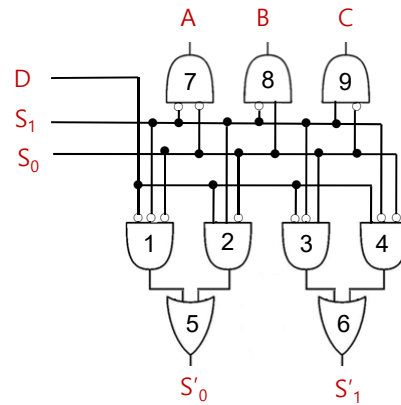
$$S'_1 = \text{OR}(\text{AND}(\text{NOT}(D), \text{NOT}(S_1), S_0), \text{AND}(D, \text{NOT}(S_1), \text{NOT}(S_0)))$$

2019/11/5

9-20

조합논리회로 부분

입력		출력			
방향 (D)	상태 ($S_1 S_0$)	다음 상태 ($S'_1 S'_0$)	A	B	C
0	0 0	0 1	1	0	0
0	0 1	1 0	0	1	0
0	1 0	0 0	0	0	1
0	1 1	0 0	0	0	0
1	0 0	1 0	1	0	0
1	0 1	0 0	0	1	0
1	1 0	0 1	0	0	1
1	1 1	0 0	0	0	0

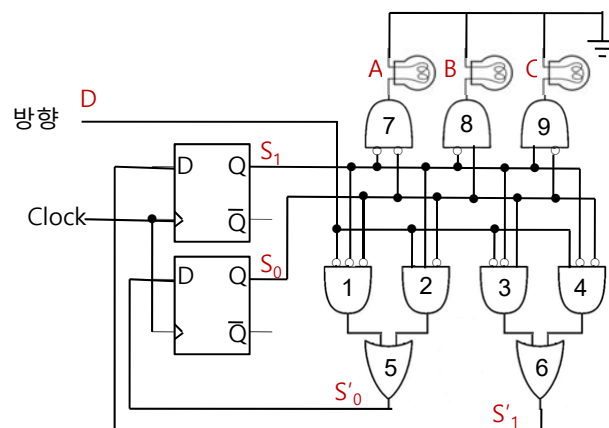


2019/11/5

9-21

5. 출력 및 다음상태 조합논리회로 작성

- 다음 상태는 D 플롭플롭의 입력으로 연결

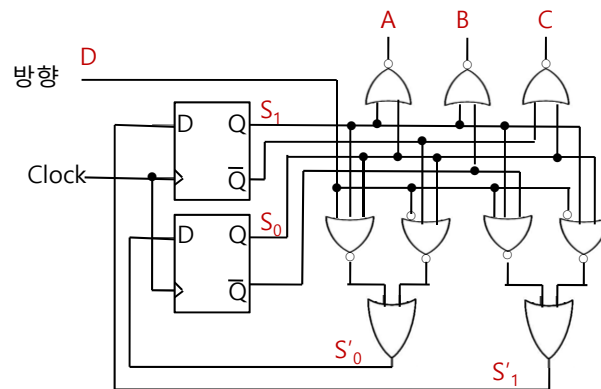


2019/11/5

9-22

회로의 개선

- D 플롭플롭의 반대 값 출력과 NAND/NOR 게이트 활용



2019/11/5

9-23

상태 번호를 변경하면?

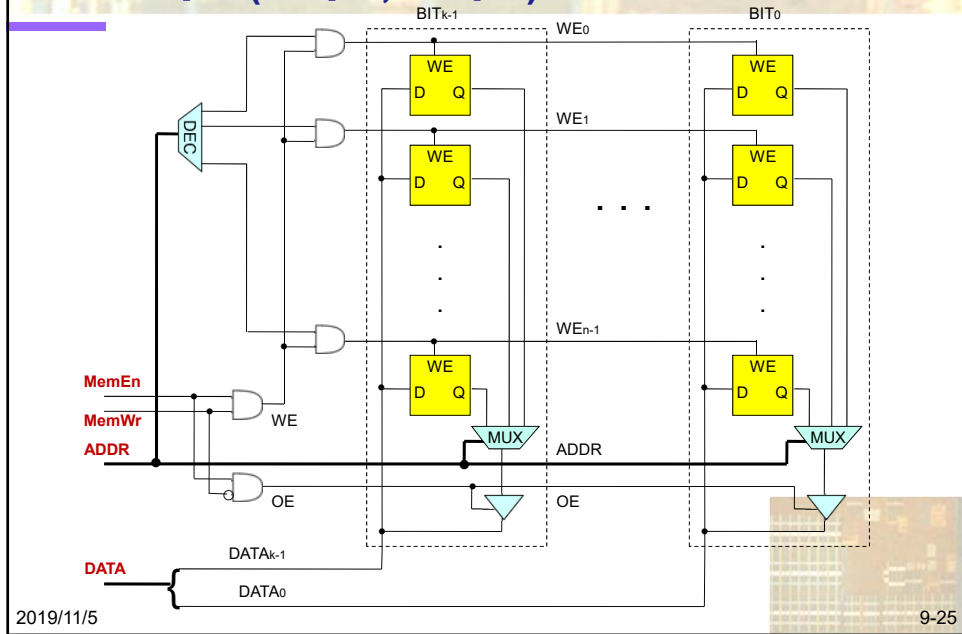
- X: 01, Y: 11, Z:10
- 사용하지 않는 00은 상태 X (01)로 천이

입력		출력			
방향 (D)	상태 ($S_1 S_0$)	다음상태 ($S'_1 S'_0$)	A	B	C
0	00				
0	01				
0	10				
0	11				
1	00				
1	01				
1	10				
1	11				

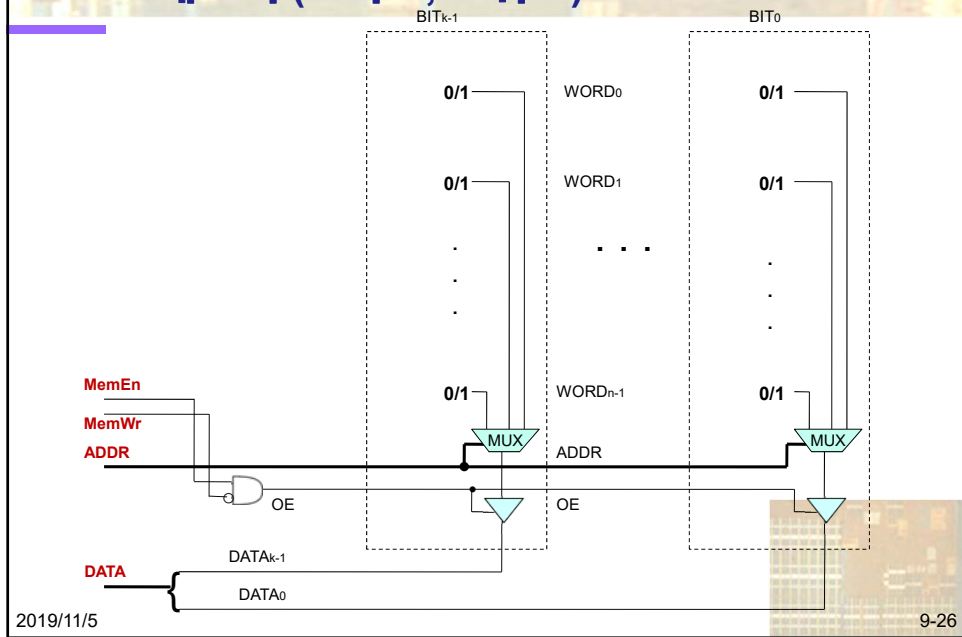
2019/11/5

9-24

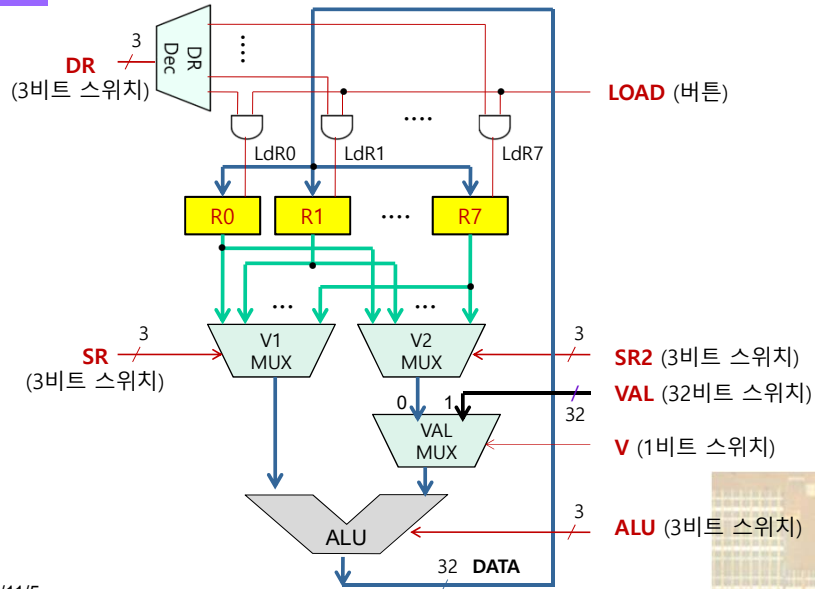
RAM 회로 (k 비트, n 워드)



ROM 메모리 (k 비트, n 워드)



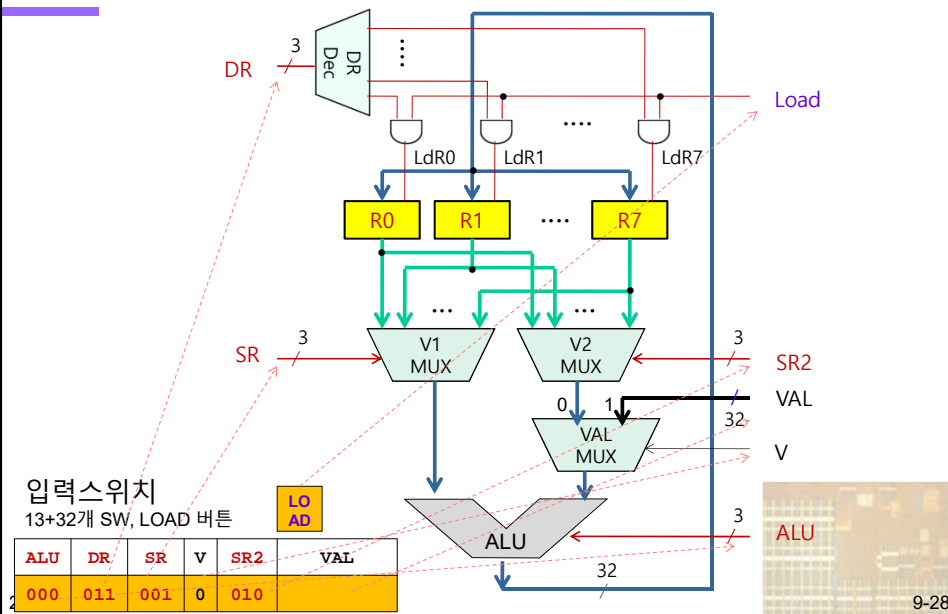
수동식 계산기 회로



2019/11/5

9-27

수동식 계산기 회로



9-28

수동계산기 조작

- 1부터 1000까지 더하기
 - ✓ COPY R0, 0
 - ✓ COPY R1, 1
 - ✓ 다음을 R1이 1001 이 될 때까지 반복
 - ✓ ADD R0, R0, R1
 - ✓ ADD R1, R1, 1
- 123 x 456
 - ✓ COPY R0, 123
 - ✓ COPY R1, 456
 - ✓ COPY R2, 0
 - ✓ 다음을 R1이 0 이 될 때까지 반복
 - ✓ ADD R2, R2, R0
 - ✓ SUB R1, R1, 1

2019/11/5

9-29