

2주차 3차시 부울 대수와 디지털 논리

【학습목표】

1. 부울 대수의 원리와 논리회로의 기능을 이해하고 설명할 수 있다.
2. 정보의 압축과 오류 방법을 이해하고 설명할 수 있다.

학습내용1 : 부울 대수

- 2진 연산 정보를 이용해서 컴퓨터에서 명제를 참과 거짓으로 판정하는 논리인 것을 강조
- 2진 연산 논리는 집합의 연산과 같음을 강조

1. 1947년 수학자 부울(George Boole)에 의해 고안

- ① 참/거짓(True/False)에 대한 논리 자료의 연산
- ② 부울 대수가 디지털 논리 회로에 적용 → 스위칭 대수에서 0은 전기가 흐름, 1은 전기가 끊긴 상태
- ③ AND(교집합), OR(합집합), NOT(여집합)(=기본 논리회로), XOR, NAND(=NOT + AND), NOR(=NOT + OR) 등
- ④ 집합과 다른점은 불 대수는 원소가 0(거짓)과 1(참)으로 구성되며, 인간의 지식이나 사고 과정을 수학적으로 해석한 것으로 논리대수라고도 함
- ⑤ 진리표(Truth Table)

입력		출력
0	0	0
0	1	0
1	0	0
1	1	1

(a) AND

입력		출력
0	0	0
0	1	1
1	0	1
1	1	1

(b) OR

입력		출력
0	0	0
0	1	1
1	0	1
1	1	0

(c) XOR

입력	출력
0	1
1	0

(d) NOT

학습내용2 : 디지털 논리회로

컴퓨터 내부에서 명제를 참과 거짓으로 판단하기 위해서는 논리 연산이 필요하고, 이를 하드웨어로 구현한 것이 논리 회로인 점을 강조

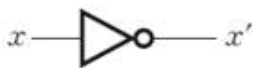
1. 논리 게이트(Gate)의 개념

* Gate : 가장 작은 단위의 디지털 논리 회로로서 부울 연산을 처리하는 장치로 한 개 이상의 입력과 하나의 출력으로 구성

- NAND 게이트와 NOR 게이트가 가장 자주 사용하는 게이트로 NOT 게이트(Inverter)에 의해 실현
- 인버터는 트랜지스터에 의해 실현, 과거에는 릴레이(Relay)와 진공관(Vacuum tube) 이용
- 인버터와 같이 작동하는 회로를 스위치(Switch)라 부름
 - 인버터를 직렬과 병렬로 연결하여 NAND회로, NOR회로 구성

* 디지털 논리 회로의 구성 요소인 게이트

NOT



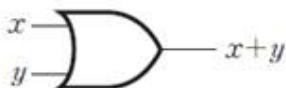
x	x'
0	1
1	0

AND



x	y	$x \cdot y$
0	0	0
0	1	0
1	0	0
1	1	1

OR



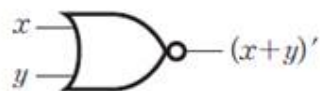
x	y	$x + y$
0	0	0
0	1	1
1	0	1
1	1	1

NAND



x	y	$(x \cdot y)'$
0	0	1
0	1	1
1	0	1
1	1	0

NOR



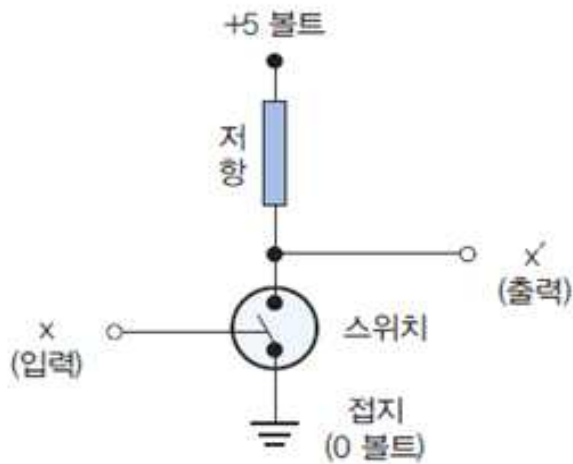
x	y	$(x + y)'$
0	0	1
0	1	0
1	0	0
1	1	0

XOR

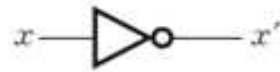


x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

* 스위치 기능을 수행하는 인버터(NOT 게이트)



(a) 구현방법



(b) 심볼

NOT

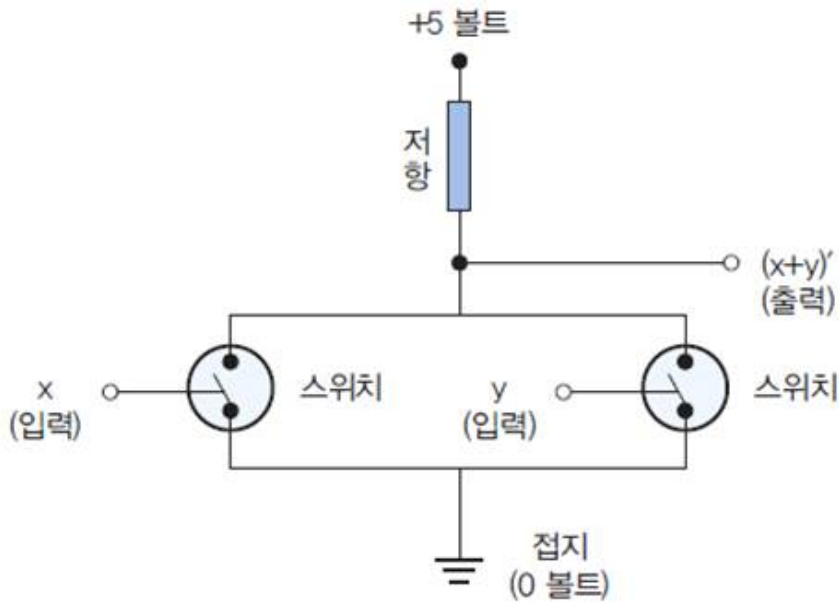
x	x'
0	1
1	0

(c) 진리표

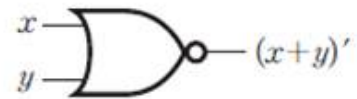


(d) 스위치를 실현하는 소자

- * NOR 게이트의 실현
- 두 개의 스위치를 병렬로 연결



(1) 구현방법



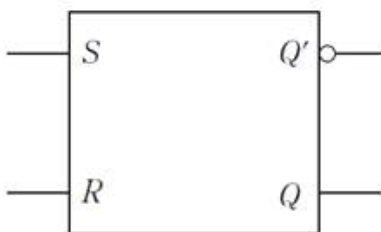
(2) 심볼

NOR

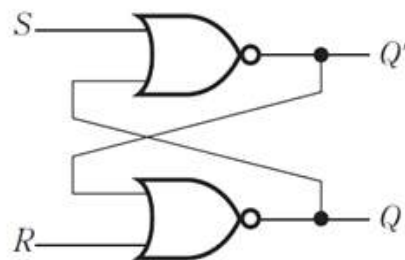
x	y	$(x+y)'$
0	0	1
0	1	0
1	0	0
1	1	0

(3) 진리표

- * Flip-Flop 회로
- 현 상태를 기억하는 디지털 논리 회로
- SR 플립-플롭(SR Latch): 1비트의 데이터를 저장
 - S: Set, R: Reset ... 순차 논리회로
 - 출력 값은 현재 입력되는 값뿐만 아니라 SR 래치의 이전 상태(즉 이전에 저장된 값)에 따라 다르게 작동



(a) 심벌



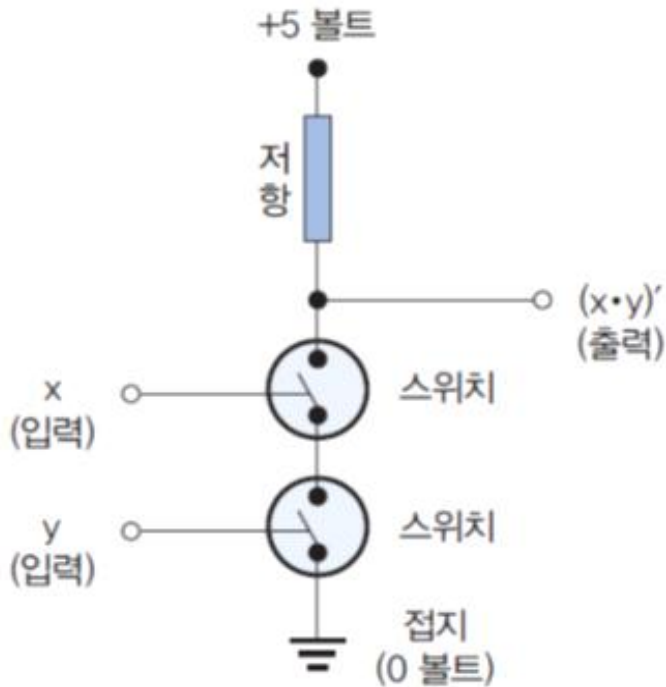
(b) 디지털 회로

S	R	$Q(t+1)$
0	0	$Q(t)$
0	1	0
1	0	1
1	1	undefined

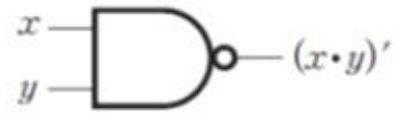
(c) 진리표

[SR 플립-플롭의 작동] (SET, RESET 장치)

- * NAND 게이트의 실현
- 두 개의 스위치를 직렬로 연결



(1) 구현방법



(2) 심볼

NAND

x	y	$(x \cdot y)'$
0	0	1
0	1	1
1	0	1
1	1	0

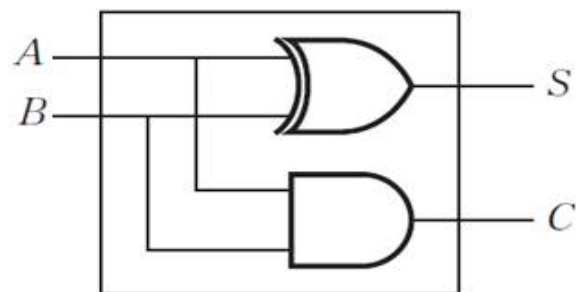
(3) 진리표

2. 논리 회로의 활용

- * (One bit) Full Adder
- 덧셈기는 컴퓨터의 논리회로 중 가장 기본인 연산회로
 - 덧셈기(Adder), 곱셈기(Multiplier), 나눗셈기(Divider)도 논리회로로 구성
- 반덧셈기(Half Adder)
 - 아랫자리에서 올라오는 캐리 비트가 없다고 가정하여 2비트 덧셈
 - S: Sum, C: Carry 비트

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

(a) 반덧셈기 진리표



S : Sum, C : Carry

(b) 반덧셈기의 논리회로

- 풀덧셈기(Full Adder)

- 아랫자리에서 올라오는 캐리 비트(Carry-in) 고려, 2비트와 자리올림 1비트를 더할 수 있는 덧셈기
- 2개의 반가산기와 1개의 OR회로로 구성
- 조합 논리회로

A	B	Carry In	Sum	Carry Out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

(a) 풀덧셈기 진리표



(b) 풀덧셈기의 논리회로

3. 정보의 압축과 오류

- 정보를 압축하는 필요성과 데이터 오류를 찾아내는 방법 중에 하나가 패리티 방법을 강조

1) 정보의 크기

가) 정보의 단위: KB, MB, GB, TB, PB, EB, ZB

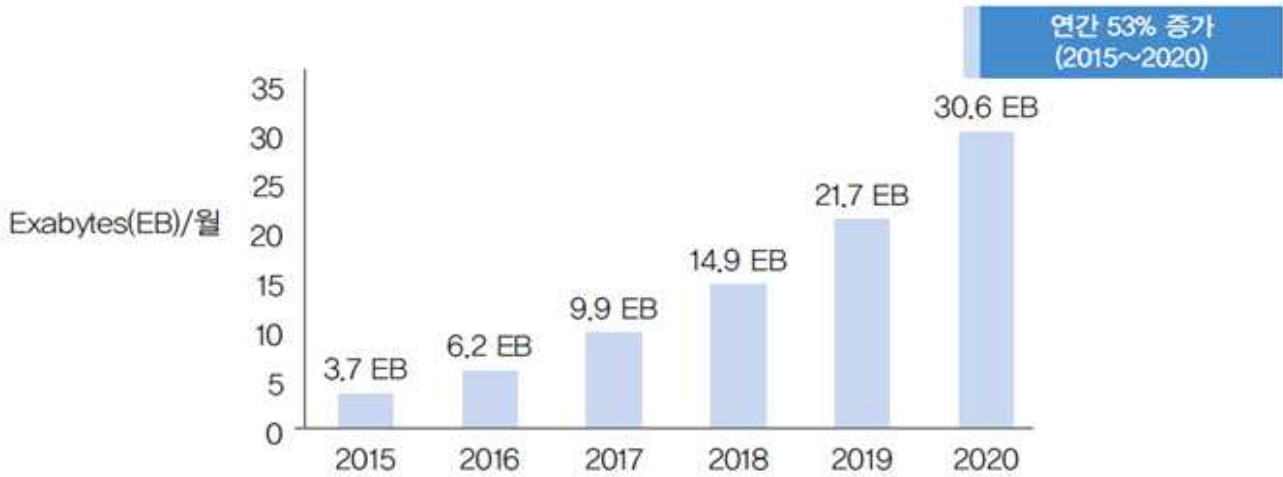
- 메모리 용량이 급증하고 특히 인터넷을 통해 소셜 미디어가 활성화되면서 최근 정보량이 기하 급수적으로 급격히 증가
- 전 세계의 정보량이 매년 21%씩 증가(2013~2018)

이름	약어	십진법	이진법
킬로(kilo)	K	$10^3 = 1000^1$	$1024^1 = 2^{10} = 1,024$
메가(mega)	M	$10^6 = 1000^2$	$1024^2 = 2^{20} = 1,048,576$
기가(giga)	G	$10^9 = 1000^3$	$1024^3 = 2^{30} = 1,073,741,824$
테라(tera)	T	$10^{12} = 1000^4$	$1024^4 = 2^{40} = 1,099,511,627,776$
페타(peta)	P	$10^{15} = 1000^5$	$1024^5 = 2^{50} = 1,125,899,906,842,624$
엑사(exa)	E	$10^{18} = 1000^6$	$1024^6 = 2^{60} = 1,152,921,504,606,846,975$
제타(zetta)	Z	$10^{21} = 1000^7$	$1024^7 = 2^{70} = 1,180,591,620,717,411,303,424$

[정보량의 단위]

나) 정보량의 증가

- 소셜미디어, SNS, 스마트폰, 사물인터넷(IoT)에 기인
- 특히, 동영상은 수 Giga

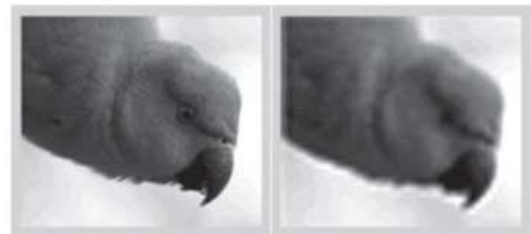
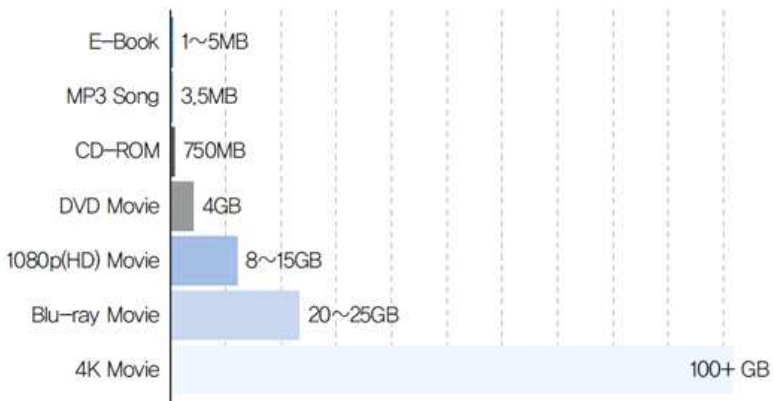


출처: Cisco VNI Mobile, 2016

[스마트폰의 미디어 용량의 증가]

다) 멀티미디어 정보 압축의 필요성

- 데이터 압축 → 데이터 전송 시간의 단축
- 다양한 압축 기법: 압축률이 높으면 복원 시 데이터 품질이 떨어짐
- 멀티미디어 데이터의 표준화



(a) 낮은 압축률

(b) 높은 압축률

[이미지의 압축률에 따른 이미지 품질의 비교]

라) 손실(Lossy)압축과 비손실(Lossless)압축

- 손실 압축: 온전한 복원 가능, 예) X-레이
 - 런-길이 부호화(Run-length Encoding),
 - 예) "aaaaabbbbbbbbaaaaccccc" → a5b7a4c6
- 비손실 압축: 예) JPEG 이미지, MP3 파일

마) 멀티미디어 데이터의 압축

- 이미지, 그래픽: BMP, TIFF, GIF, PNG, JPEG 등
- 사운드: WAV, Au, A-Law, MP3, AAC 등
 - 스트리밍 방식: RealAudio
- 동영상: MPEG-1, MPEG-2, MPEG-4, MPEG-7, H.264, MPEG-21 등
- 목적과 환경에 따라 다름

바) 정보의 오류 탐지(Error detection)

- 컴퓨터 내부에서 구성 요소 간의 전송 오류
- 원거리의 컴퓨터로 전송되는 디지털 데이터의 통신 오류
- 패리티 비트(Parity bit): 홀수 패리티, 짝수 패리티
 - 예) 홀수 패리티

1	0	1	1	0	1	1	1	0	1	1	0
---	---	---	---	---	---	---	---	---	---	---	---

비트 패턴에서 "1"의 개수가 짝수



(패리티 비트 "1"을 더함)

1	1	0	1	1	0	1	1	1	0	1	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---

비트 패턴에서 "1"의 개수가 홀수

패리티 비트

사) 오류 복원(Error correction)

- 데이터 전송 오류를 발견하여 원래의 올바른 데이터로 복원
- 해밍(Richard Hamming)이 제시
 - '1011101'과 '1001001'사이의 해밍 거리는 2이다. '2143896'과 '2233796'사이의 해밍 거리는 3이다.
 - "toned"와 "roses"사이의 해밍 거리는 3이다.
 - 해밍 거리(Hamming distance) 기법 적용
 - 데이터 전송시 발생한 오류를 원래의 데이터로 복원

【학습정리】

1. 기본 논리회로는 NOT 회로, AND회로, OR회로로 구성된다.
2. 회로는 XOR 회로는 두 입력 단자가 서로 다른 논리 값일 때만 참이 되는 회로이다.
3. 전가산기는 2개의 반가산기와 1개의 OR회로로 구성된다.