

챕터8 정리

Ch8: Combinational Circuit Design and Simulation Using Gates

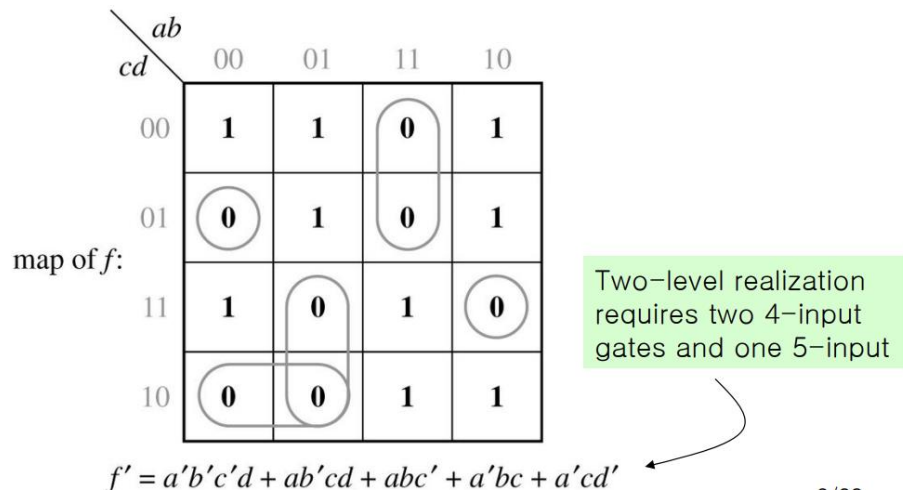
8.1 Review of Combinational Circuit Design

앞부분 복습이기 때문에 교수님이 수업에서 안 다루심 패스

8.2 Design of Circuits with Limited Gate Fan-in

• Limited Gate Fan-in

- Example: Realize $f(a, b, c, d) = \sum m(0, 3, 4, 5, 8, 9, 10, 14, 15)$ using 3-input NOR gate



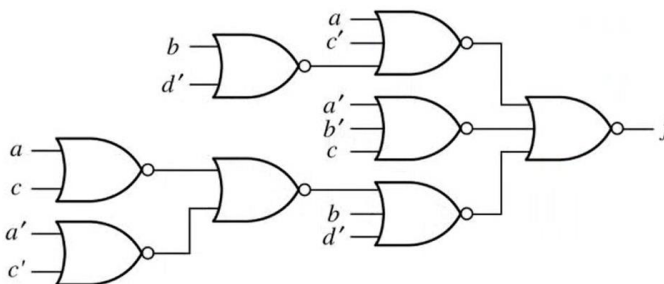
0에 초점을 맞춘 expansion

→ f' 은 4개의 인풋을 가진 AND 게이트와 5개의 인풋을 가진 OR 게이트가 필요함

근데 만약 5개의 입력을 가진 OR 게이트가 없다면? 최대 3인풋인 NOR 게이트만 있다면?

$$f' = b'd(a'c' + ac) + a'c(b + d') + abc'$$

$$f = [b + d' + (a + c)(a' + c')][a + c' + b'd][a' + b' + c']$$



→ 공통의 식을 묶어서 이런 식으로 가능. 중간과정이 생략되어 있다

- Fan-in과 Fan-out

Fan-in: 게이트에 입력할 수 있는 최대의 입력 개수

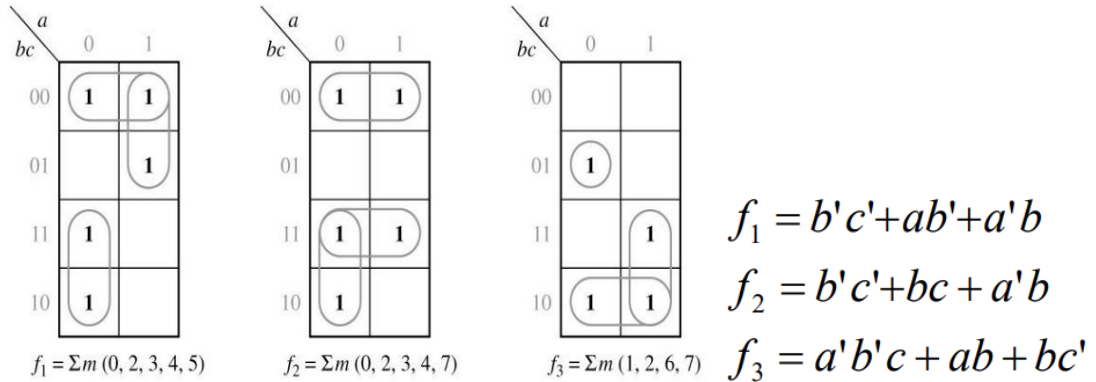
Fan-out: 팬인에 대응되는 개념. 어떤 게이트에서 아웃풋이 연결할 수 있는 최대 개수

선 연결이 된다는 것은 회로 개념에서 본다면 전류가 흐르는 것임

전류 양이 미미하면 연결될 수 있는 게이트의 개수가 제한될 수 있음

전류의 양에 따라서 연결할 수 있는 게이트의 수가 달라짐. 이런 현상과 관련됨

- Example: Realize the functions given in Figure, using only 2-input NAND gates and inverters



→ Each function requires a 3-input OR gate. 식 변형이 필요함

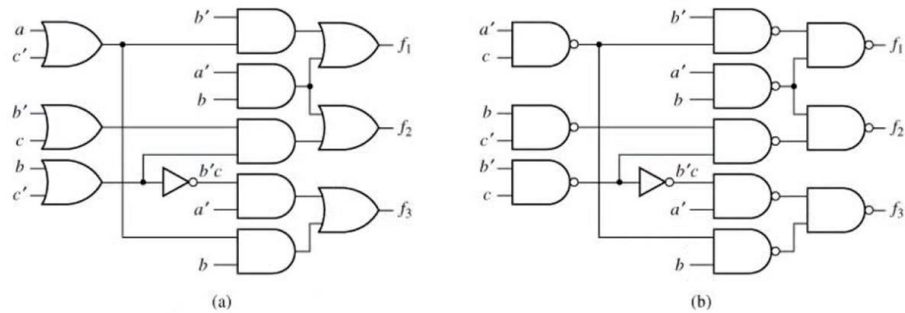


Figure 8-3: Realization of Figure 8-2

$$\begin{aligned}
 f_1 &= b'(a+c') + a'b \\
 f_2 &= b(a'+c) + b'c' \\
 \text{or } &= (b'+c)(b+c') + a'b \\
 f_3 &= a'b'c + b(a+c')
 \end{aligned}$$



$$\begin{aligned}
 f_1 &= b'(a+c') + a'b \\
 f_2 &= (b'+c)(b+c') + a'b \\
 f_3 &= a'(b+c')' + b(a+c')
 \end{aligned}$$

$$a'b'c = a'(b'c) = a'(b+c')'$$

6/28

→ 변형 과정: 공통인 요소들로 묶은 후 게이트를 NAND로 바꿔 줌

8.3 Gate delays and Timing Diagrams

• Gate delay

- Gate delay

AND든 OR이든 뭐든 간에 실제 내부 회로는 트랜지스터로 이루어지고 기능한다

트랜지스터가 기능한다는 것은 전압전류 신호들이 흐름으로써 동작특성을 한다는 소리

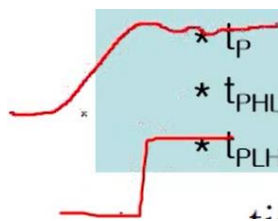
게이트들이 동작을 할 때 흐르는 과정에서 약간의 시간 간격이 필요함 (ns 수준일지라도)

현실에선 무시할 만한 수준이더라도 회로 설계의 측면에선 무시 불가함

그 시간 요소를 고려해야 하는데, 이 장은 이 내용에 관한 것임

신호가 low에서 high로 바뀔 때 직각으로 바뀌지 않음.

실질적으로는 상승하는데 시간간격이 필요함



→ 밑의 게 아니라 위의 것처럼 오른단 소리

신호뿐만 아니라 게이트가 입력이 들어와서 출력이 나올 때 까지의 시간도 필요함

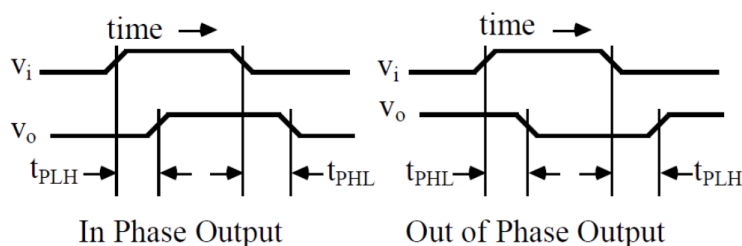
이를 propagation delay라고 함

- propagation delay

Real gates have delay, they do not respond immediately

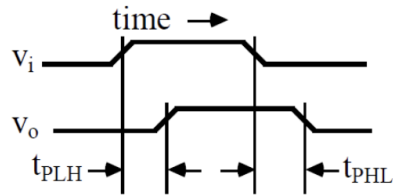
propagation delay: the time required for an input signal change to produce an output signal change

→ input signal change가 output signal change로 전달되는데 걸리는 시간



In Phase Output: 입력이 Low → High로 변함에 따라 출력도 같은 양상을 띠

Out of Phase Output: 입력이 Low → High로 변하면 출력은 반대로



In Phase Output

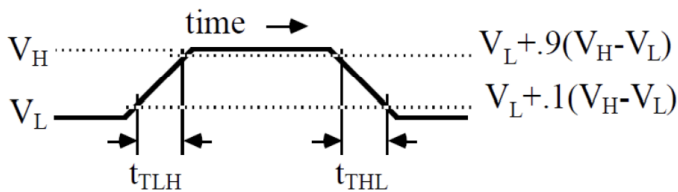
입력이 Low→High로 변할 때 출력도 Low→High로 바뀌는 propagation delay를 알아보자

이때의 propagation delay time을 t_{PLH} 라고 한다(Low-to-High propagation delay)

반대로 High→Low로 변할 때 걸리는 시간은 t_{PHL} 이라 함(High-to-Low propagation delay)

뭉뚱그려 t_p 라고 함 - 가장 큰 Worst case Time delay

- All signals take time to change



$t_{TLH} = t_{rise} =$ Time to make a low-to-high transition

→ low에서 high로 transition 하는데 걸리는 시간

$t_{THL} = t_{fall} =$ Time to make a high-to-low transition

→ high에서 low로 transition 하는데 걸리는 시간

올라감을 나타내는 척도로써 전압이 올라갈 때 Low Voltage와 High Voltage 사이의 간격이 있을 텐데, 그 간격의 10%부터 90%까지 도달하는 시간을 rise, falling time이라 함

Rise time과 Falling time은 일반적으로 같지 않음. 미세하게나마 약간의 차이가 있음

- Gate delay가 변하는 요인들

Temperature

Supply voltage(몇 볼트의 로직인가? - 낮을수록 줄어들게 됨)

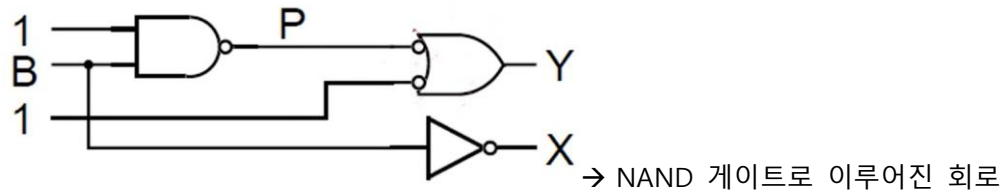
Capacitive load(선과 선 사이의 미세한 정전용량의 존재 - Capacitance)

Age(회로의 경과 시간)

연산 속도를 높이기 위해서 delay를 줄여야 함

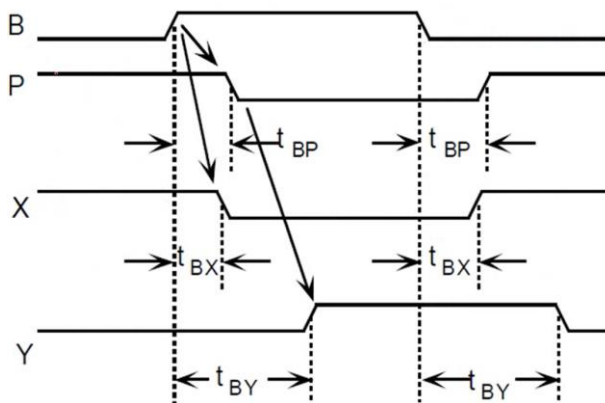
· Timing Diagrams

- Timing Diagrams – 간단한 예시를 살펴보자



입력 3개 중에 항상 1인 것이 2개이고, B는 0에서 1로, 1에서 0으로 아래 그래프와 같이 바뀐다고 하자

이때의 P, X, Y의 시그널 특성을 보이는지 propagation delay를 고려하여 살펴보자



P: (1 → 0 → 1) 초기 값은 1임. B가 0에서 1로 바뀌면 1에서 0으로 변함. B가 다시 0으로 바뀌면서 0이 됨. 그 과정에서 약간의 시간 간격이 필요함(propagation time)

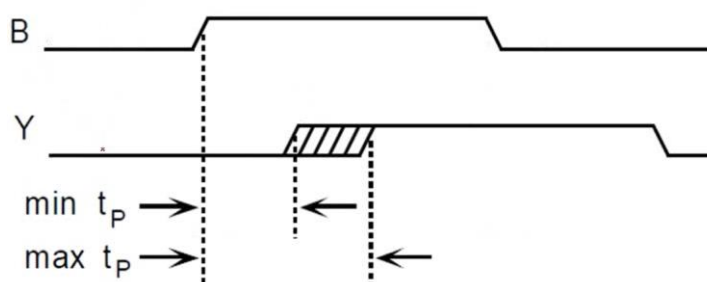
X: (1 → 0 → 1) P와 동일한 스펙의 NAND 게이트라면 딜레이 시간이 유사하겠지만 단순 인버터를 사용했다면 propagation delay가 차이 남

Y: (0 → 1 → 0) P의 변화가 Y에 영향을 미침. 그만큼 더 propagation delay가 필요함

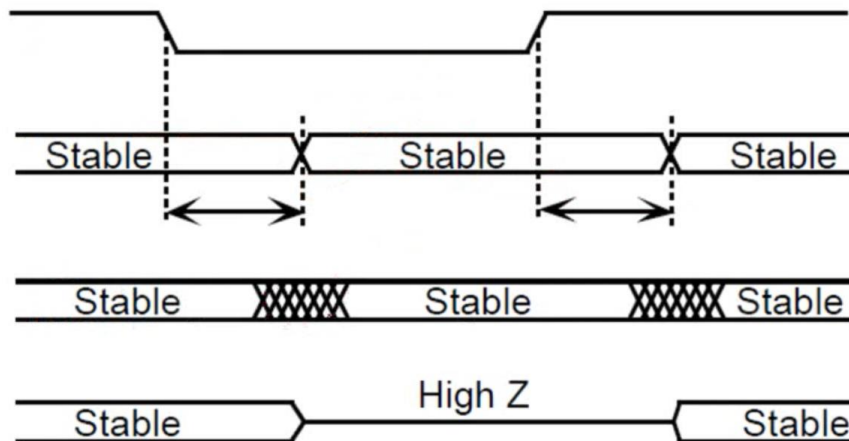
→ B의 변화에 따른 Y의 변화는 두 변화를 합친 값임

이러한 도식을 Timing Diagram이라 한다

B와 Y만을 따로 떼서 보자

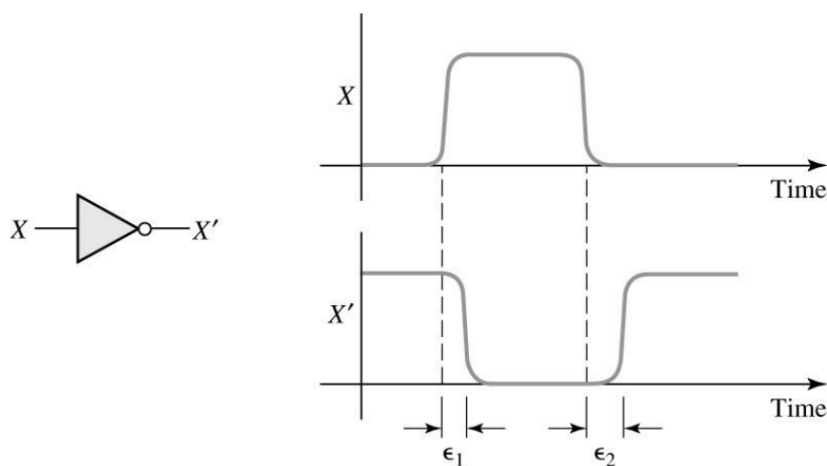


빗금은 뭐냐? → 최소의 propagation delay와 최대의 propagation delay 사이 간격임
 즉, Y가 0에서 1로 변하는 데 바뀔 지 안 바뀔 지 장담할 수 없는 영역(unstable 영역)
 회로 설계 시에 이런 상태에서의 다른 변환이 일어나지 않게 회로 구현을 해야함
 게이트들이 이런 특징을 갖고 있다면 사용할 수 있는 스피드의 제한이 있다
 High에서 Low로의 표현까지 있는데, 이를 한번에 표현한 것이 아래 그림임



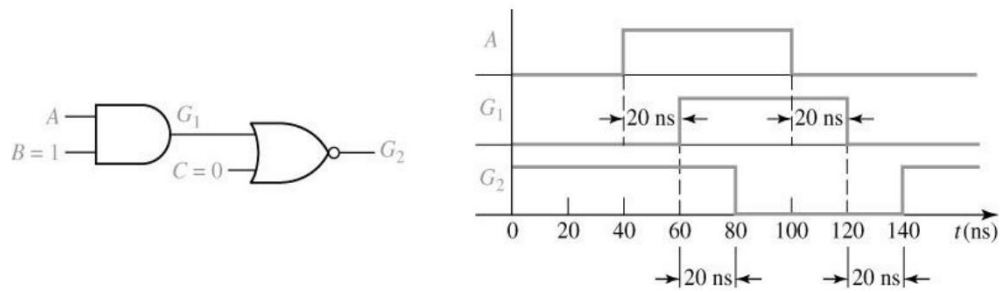
High Z라는 것은 임피던스를 의미함(Impedance: 저항적인 요소) → 하이 임피던스는 무한대의 저항(끊김과 동일), 플로팅 되어있는 상태라고도 함

- Propagation Delay in an Inverter



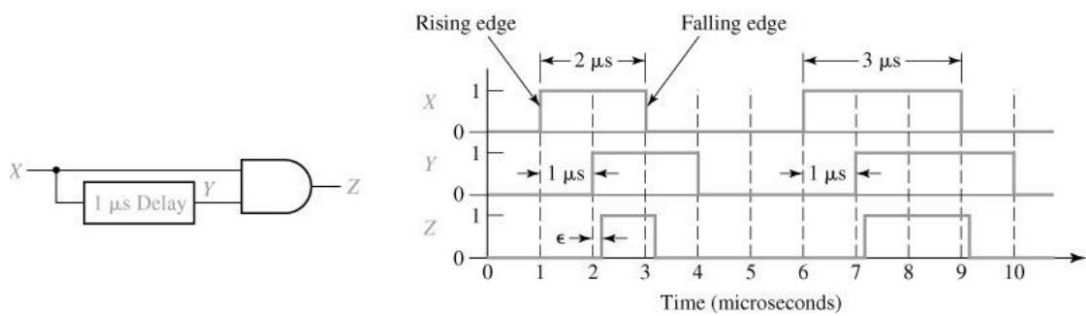
Propagation Delay 존재함. 미미하지만 $0 \rightarrow 1$ & $1 \rightarrow 0$ 일 때 일반적으로 다르다

- Timing Diagram for AND-NOR Circuit



→ 교재는 이해를 돕기 위해 단순히 시그널 변화를 직각으로 표현함. 실제로는 아님

- Timing Diagram for Circuit with Delay



AND 게이트 입력이 출력으로 전달되는 데까지의 걸리는 시간은 엡실론

딜레이 요소를 잘 고려하라는 뜻

Low → High: Rising edge 혹은 Positive edge 또는 Leading edge

High → Low: Falling edge 혹은 Negative edge 또는 Trailing edge

8.4 Hazards in Combinational Logic

· Hazards in Combinational Logic

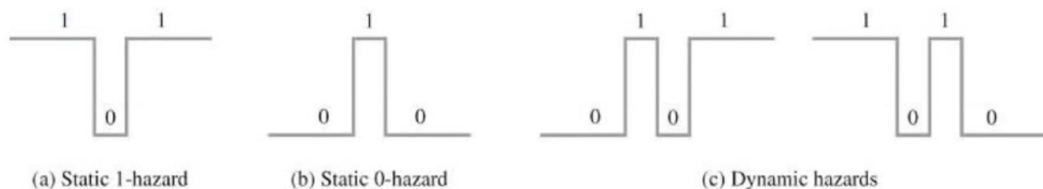
- Types of Hazards

타이밍 다이어그램으로 인한, 즉 Propagation Delay로 인해 전혀 예기치 못한 현상이 일어날 수 있음. 이것을 Hazards라고 함

항상 1로 유지되어야 하는데 타이밍 다이어그램 문제 때문에 순간적으로 0으로 빠졌다가 다시 1이 되는 현상이 일어날 수도 있고(Static 1-hazard)

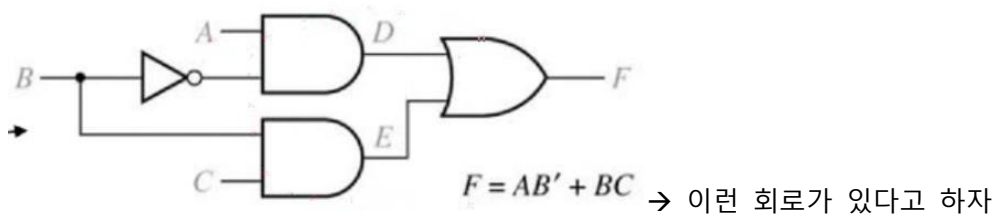
항상 0이어야 하는데 순간적으로 1이 되었다가 0이 되는 경우(Static 0-hazard)

0에서 1로 혹은 1에서 0으로 순조롭게 넘어가야 하는데 왔다갔다하다 바뀌는 현상(Dynamic hazards)



- Detection of a Static 1-Hazard

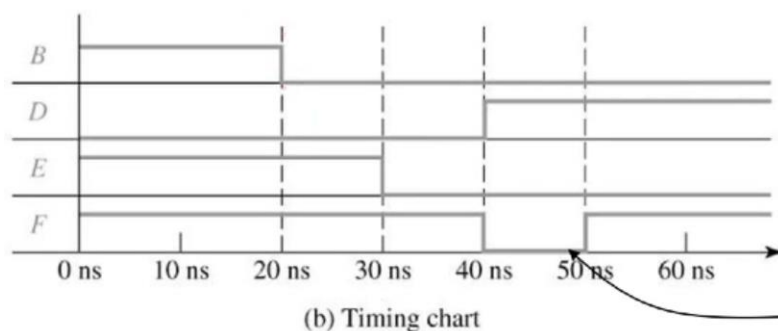
Static 1-hazard가 어떻게 일어나는지 araboja



A와 C는 항상 1이고, B는 1→0이다

따라서 D는 0→1, E는 1→0, 따라서 F는 D + E이므로 항상 1을 유지해야 함

그런데 이 Propagation Delay로 인해 0일 때가 있다



각 게이트마다 각각 10ns씩 걸린다고 했을 때 타이밍 차트는 다음과 같음

F는 10ns동안 1 해저드가 발생함(0-Spike)

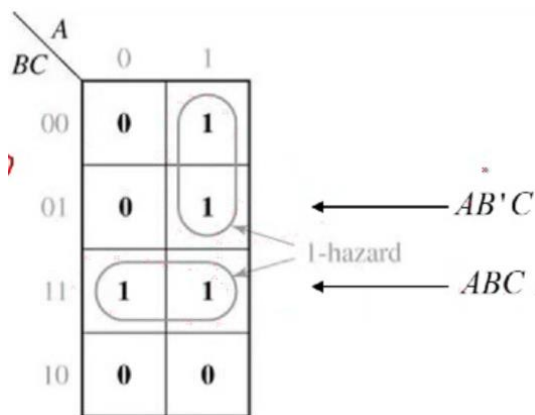
항상 1이어야 하는데 0인 경우가 발생함

왜? 게이트 딜레이로 인한 D와 E의 Transition 타이밍이 다르기 때문에

Spike 현상이 왜 일어나는가?

→ 어떤 Transition의 전후로 발생한다. 두번째가 on 되기 전에 첫번째가 먼저 off 되어서

카르노 맵으로 보면 다음과 같다



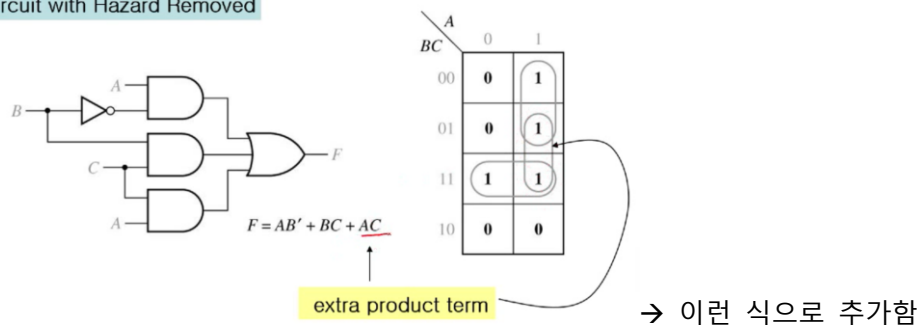
둘을 묶으면 AC. AC는 1로 변함이 없음

근데 B가 B'으로 바뀌는 과정에서 Hazard 현상이 일어남

그루핑할 때 그룹과 그룹 간의 트랜지션이 일어날 때 해저드 현상이 발생함

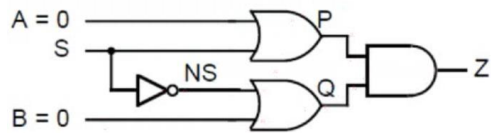
커버하는 방법? Extra Product Term을 추가함. 인접한 그루핑 사이의 공통의 추가의 그루핑을 해주면 트랜지션 현상 사이에도 해결됨

• Circuit with Hazard Removed

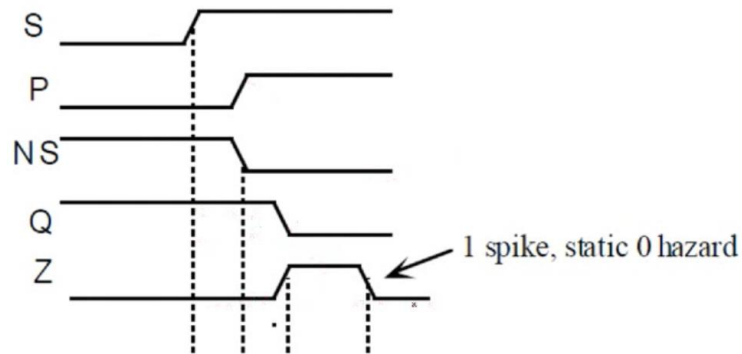


이런 해저드 현상은 SOP 형태에서 일어남

- Static 0-Hazard의 발생



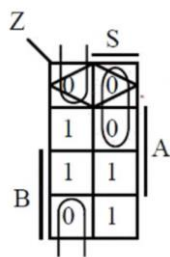
이러한 POS의 형태에서 일어난다



Analysis 과정 생략, 이렇게 1 Spike 현상이 일어남

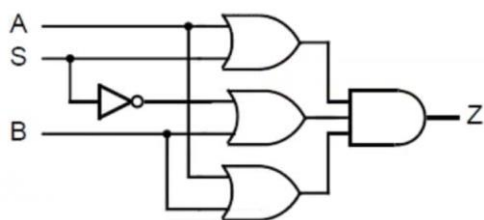
트랜지션 전과 후의 타임 딜레이가 달라서 발생한 것임

카르노 맵에서의 설명



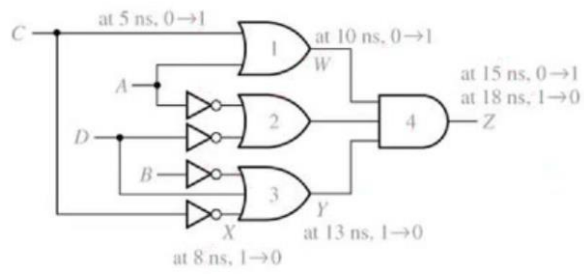
인접한 그룹핑 사이의 트랜지션이 일어날 때 생긴다

똑같이 엑스트라 OR게이트를 추가함

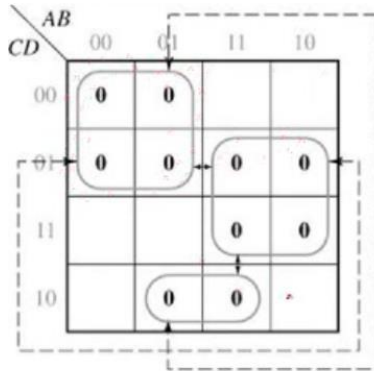


- Detection of a Static 0-Hazard

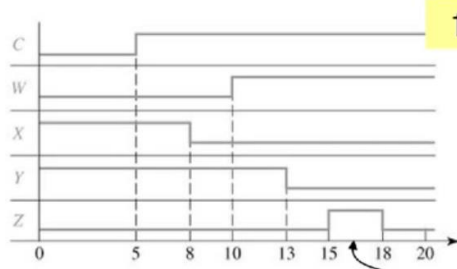
$$F = (A + C)(A' + D')(B' + C' + D)$$



(a) Circuit with a static 0-hazard

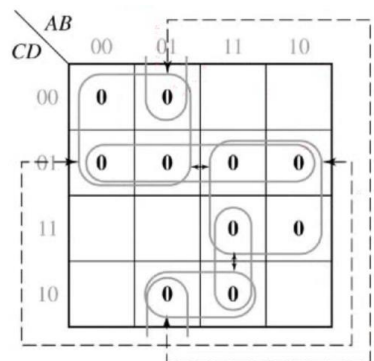


(b) Karnaugh map for circuit of (a) → 인접한 그룹 사이에 해저드 발생 우려



1-Spike, Static 0-Hazard

24/28

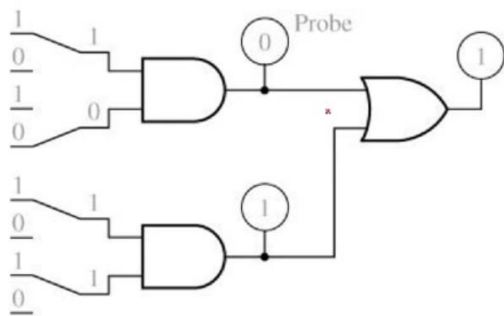


→ 인접한 그룹핑 사이에 추가의 OR 게이트를 추가함

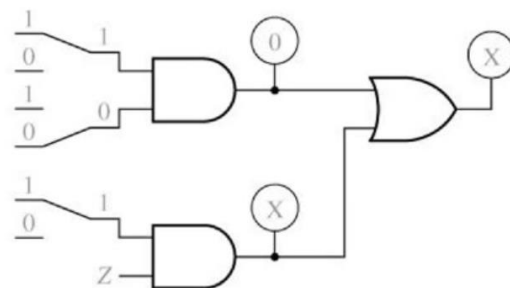
$$F = (A + C)(A' + D')(B' + C' + D)(C + D')(A + B' + D)(A' + B' + C')$$

8.5 Simulation and Testing of Logic Circuits

- Simulation and Testing of Logic Circuits
- Simulation and Testing of Logic Circuits



(a) Simulation screen showing switches



(b) Simulation screen with missing gate input

Probe: 시그널 값이 0인지 1인지 확인하는 소프트웨어

X: unknown value

Z: high impedance (hi-Z) connection (open circuit) → 선이 끊어진 상태

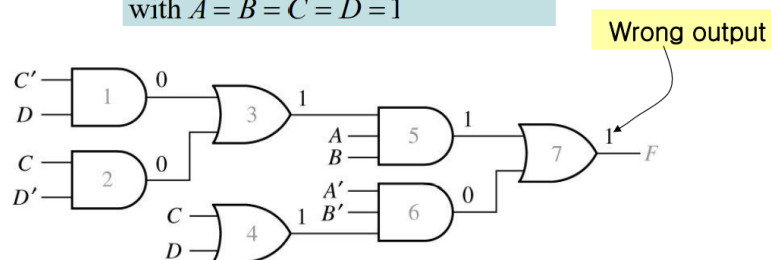
- And and OR Functions for Four-Valued Simulation

x	0	1	X	Z
0	0	0	0	0
1	0	1	X	X
X	0	X	X	X
Z	0	X	X	X

+	0	1	X	Z
0	0	1	X	X
1	1	1	1	1
X	X	1	X	X
Z	X	1	X	X

- Logic Circuit with Incorrect Output

Example: $F = AB(C'D + CD') + A'B'(C + D)$
with $A = B = C = D = 1$



Example

하드웨어적으로 구현하고 동작시켰을 때 오동작이 일어날 수 있다. 선이 끊어지거나 게이트가 고장 나거나 등등 → 어디가 잘못되었는지를 확인해야 한다(Troubleshooting이라

고 함)

Troubleshooting a Logic Circuit

→ Starting at the output gate and working back until the wrong connection or defective gate is located

예시에선 0이 나와야 하는데 1이 나옴 → 추적 부분부터 시작해서 역으로 추적해 들어감
보니까 7번 게이트는 문제없음 $0 \cdot 1 = 1$ 나왔으니까

그 다음 5번과 6번도 문제가 없음

4번을 보면 문제없음 근데 3번을 보니 0 0이 들어갔는데 1이 나왔네? 3번이 문제임

→ 3번 칩이 문제이니 바꿔 끼우면 되네