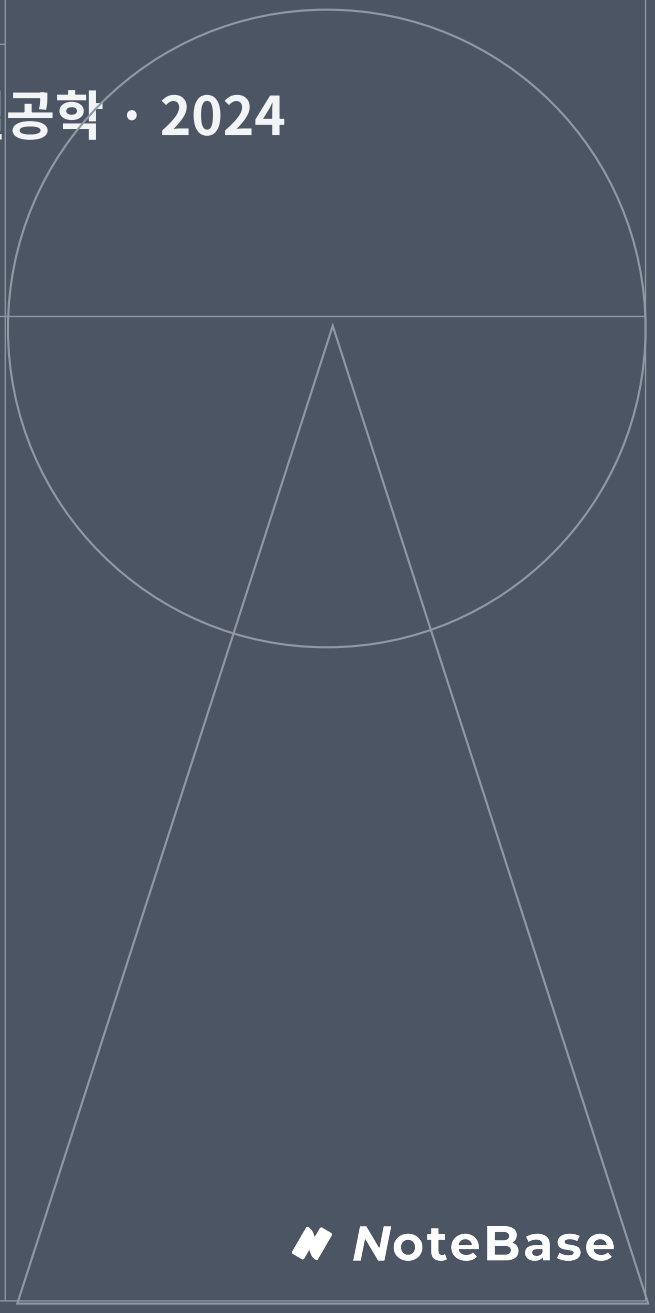
기출OX 해설집

공무원/서울시/7급 · 디지털공학 · 2024



NoteBase

기출OX 해설집

공무원/서울시/7급 · 디지털공학 · 2024

공무원/서울시/7급 · 디지털공학 · 2024

문 1. 16진수 $AF930C_{(16)}$를 8진수로 변환한 값은?

① $53711414_{(8)}$

② $11506444_{(8)}$

③ $10101111100100110000110 0_{(8)}$

④ $15706444_{(8)}$

정답 1번

출제 포인트

이 문제는 16진수를 2진수로 바꾼 뒤 3비트씩 묶어 8진수로 변환할 수 있는지를 묻는 문제이다.

조건 정리

주어진 값은 $AF930C_{16}$이다.
구해야 할 값은 이 수를 8진수로 변환한 결과이다.
주의할 조건은 16진수 한 자리는 4비트, 8진수 한 자리는 3비트에 대응한다는 점이다.

사용할 공식

$16 \rightarrow 2 \rightarrow 8$ 순서로 변환한다.
$A = 1010,\ F = 1111,\ 9 = 1001,\ 3 = 0011,\ 0 = 0000,\ C = 1100$으로 바꾼다.
이 공식을 쓰는 이유는 16진수와 8진수는 직접 자리값을 맞추기보다 둘 다 2의 거듭제곱 진수이므로 2진수로 거치면 가장 정확하게 변환할 수 있기 때문이다.

계산 과정

1. $AF930C_{16} = 1010\ 1111\ 1001\ 0011\ 0000\ 1100_2$이다.
2. 이를 왼쪽부터 3비트씩 묶기 좋게 정리하면 $101\ 011\ 111\ 001\ 001\ 100\ 001\ 100_2$이다.
3. 각 묶음을 8진수 한 자리로 바꾸면
$101 = 5,\ 011 = 3,\ 111 = 7,\ 001 = 1,\ 001 = 1,\ 100 = 4,\ 001 = 1,\ 100 = 4$이므로 결과는 53711414_8이다.

NoteBase

선지 판단

1번 O: $AF930C_{16}$를 2진수로 바꾼 뒤 3비트씩 묶으면 53711414_8가 되므로 옳다.
2번 X: "11506444"가 틀린 표현이다. 3비트 단위 변환 결과와 일치하지 않는다. 올바르게는 53711414_8이다.
3번 X: "10101111100100110000110 0(8)"이 틀린 표현이다. 이 값은 8진수가 아니라 $AF930C_{16}$를 2진수로 전개한 형태이다. 올바르게는 2진수로 $10101111100100110000110 0_2$, 8진수로는 53711414_8이다.
4번 X: "15706444"가 틀린 표현이다. 3비트씩 정확히 끊어 읽은 결과가 아니므로 맞지 않는다. 올바르게는 53711414_8이다.

오답 포인트

16진수 각 자리를 바로 8진수로 옮기려 하거나, 2진수로 바꾼 뒤 3비트가 아니라 다른 단위로 끊으면 오답이 나온다. 특히 3번처럼 2진수 전개 결과를 그대로 8진수라고 적는 실수가 자주 나온다.

다시 나오면 이렇게 판단

진수 변환 문제는 먼저 기준 진수를 2진수로 바꾼 뒤, 목표 진수의 비트 단위에 맞춰 다시 묶어 판단한다.
16진수는 4비트, 8진수는 3비트라는 대응만 지키면 된다.

암기 포인트

16진수 한 자리는 4비트, 8진수 한 자리는 3비트이므로 16 → 2 → 8 변환이 가장 안전하다.

NoteBase

문 2. <보기>의 진리표를 이용해 출력 F를 곱의 합 형태로 최대로 간략화한 논리식은? (단, X는 무정의(don't care)를 뜻한다.)

<보기>

입력 $ABCD$	출력 F	입력 $ABCD$	출력 F
0000	0	1000	1
0001	1	1001	0
0010	0	1010	0
0011	0	1011	0
0100	X	1100	1
0101	1	1101	X
0110	0	1110	0
0111	1	1111	1

① $F = \overline{A}B + \overline{C}D$

② $F = AB + BC + ACD$

③ $F = BD + \overline{A}CD + A\overline{CD}$

④ $F = B\overline{C} + BD + \overline{A}BC + CD$

정답 3번

출제 포인트

이 문제는 진리표의 1과 무정의(don't care)를 카르노 맵에 옮겨 출력 F를 곱의 합(SOP) 형태로 최대한 간략화할 수 있는지를 묻는 문제이다. 발문이 \"최대로 간략화한 곱의 합\"을 요구하므로, 1이 되는 행을 모두 포함하면서 don't care는 묶음을 키우는 데에만 활용해 항의 수와 변수의 수를 최소로 만드는 식을 골라야 한다.

NoteBase

선지별 해설

1번: X. $F = \overline{A}B + \overline{C}D$ 형태인데, $\overline{A}B$는 $A=0, B=1$인 0100, 0101, 0110, 0111을 모두 1로 만든다. 그러나 진리표에서 0110(m_6)은 $F=0$이므로 0인 행까지 켜는 모순이 생긴다. 또한 이 식에는 $A=1$인 항이 없어 1이어야 하는 1000(m_8), 1100(m_{12}), 1111(m_{15})을 전혀 만들지 못한다. 올바른 식은 $F = BD + \overline{AC}D + A\overline{CD}$이다.

2번: X. AB는 1100, 1101, 1110, 1111을, BC는 0110, 0111, 1110, 1111을 1로 만드는데, 0인 행인 1110(m_{14}), 0110(m_6)까지 켜 버린다. 반대로 0001(m_1), 0101(m_5), 1000(m_8)은 어느 항으로도 덮지 못해 누락된다. 올바른 식은 $BD + \overline{AC}D + A\overline{CD}$이다.

3번: O. BD는 $B=1, D=1$인 0101(m_5), 0111(m_7), 1111(m_{15})을 한 묶음으로 덮고, don't care인 1101(m_{13})까지 같은 묶음에 포함시켜 4셀짜리 큰 묶음을 만든다. $\overline{AC}D$는 $A=0, C=0, D=1$인 0001(m_1)과 0101(m_5)을 덮고, $A\overline{CD}$는 $A=1, C=0, D=0$인 1000(m_8)과 1100(m_{12})을 덮는다. 세 항을 합치면 1이어야 하는 $m_1, m_5, m_7, m_8, m_{12}, m_{15}$만 정확히 포함하고 0인 행은 하나도 켜지 않으며, 항·변수 수도 가장 적어 최소 곱의 합 조건을 만족한다.

4번: X. $\overline{B}C$는 $B=0, C=1$인 0010, 0011, 1010, 1011을, CD는 $C=1, D=1$인 0011, 0111, 1011, 1111을 1로 만드는데, 진리표에서 0010(m_2), 0011(m_3), 1011(m_{11}) 등은 $F=0$이므로 0인 행까지 켜 버린다. 또한 1000(m_8), 1100(m_{12})처럼 실제로 1이어야 하는 행을 덮지 못한다. 올바른 식은 $BD + \overline{AC}D + A\overline{CD}$이다.

오답 포인트

이 문제의 함정은 1이 되는 행 몇 개만 눈으로 맞춰 보고 식을 고르면 안 된다는 점이다. 각 곱항이 어떤 셀을 덮는지 모두 따져, 0인 행(예: 0110, 1110, 0011)까지 켜는 항이 들어 있지는 않은지 반드시 대입해 확인한다. don't care(0100, 1101)는 1 옆에서 묶음을 키우는 데만 보조로 쓰며, 그 자체를 반드시 1로 만들 필요는 없다.

다시 나오면 이렇게 판단

진리표 최소화 문제는 먼저 1의 위치(최소항 번호)를 모두 적고 X는 보조로만 사용한다. 그다음 카르노 맵에서 가장 큰 묶음을 만든 뒤 각 묶음에서 변하지 않는 변수만 남긴다. 선지를 검증할 때는 각 곱항이 덮는 셀을 모두 적어 1인 행을 빠짐없이 덮고 0인 행은 하나도 켜지 않는지 교차 확인하면 정답 식을 안정적으로 찾을 수 있다.

암기 포인트

곱의 합 최소화는 1을 묶고 X는 묶음 확대에만 쓰며, 묶음 안에서 변하지 않는 변수만 남긴다. 후보 식은 덮는 셀로 검산한다.

NoteBase

문 3. <보기>의 논리식을 최소항식 또는 최대항식으로 표현한 것으로 가장 옳지 않은 것은? (단, m은 최소항, M은 최대항, 괄호 안의 숫자는 최소항 또는 최대항의 번호, 기호 Σ는 논리합, 기호 Π는 논리곱을 의미한다.)

<보기>

$\overline{A}\overline{B}C + \overline{A}B\overline{C} + \overline{A}BC + A\overline{B}\overline{C} + A\overline{B}C$

① $\Sigma m(1, 2, 3, 4, 5)$

② $\overline{\Pi M(1, 2, 3, 4, 5)}$

③ $\overline{\Sigma m(0, 6, 7)}$

④ $\overline{\Pi M(0, 6, 7)}$

정답 4번

출제 포인트

이 문제는 주어진 논리식을 최소항 번호로 바꾸고, 그 여집합으로 최대항식을 연결할 수 있는지를 묻는다.

조건 정리

주어진 식을 F라 두면 $F = \bar{A}\bar{B}C + \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}\bar{C} + A\bar{B}C$이다.
각 곱항은 A, B, C의 조합 하나에 대응하는 최소항이다.
3변수이므로 가능한 번호는 0부터 7까지이다.

사용할 공식

최소항의 번호 집합을 S라 하면 $F = \Sigma m(S) = \Pi M(S^c)$이다.
또한 $\overline{\Sigma m(S^c)} = \Sigma m(S)$, $\overline{\Pi M(S)} = \Sigma m(S)$이다.
이 관계를 쓰는 이유는 보기들이 원식과 보수 표기를 함께 섞어 놓았기 때문이다.

계산 과정

1. $\bar{A}\bar{B}C = 001$이므로 m_1이다.
2. $\bar{A}B\bar{C} = 010$, $\bar{A}BC = 011$, $A\bar{B}\bar{C} = 100$, $A\bar{B}C = 101$이므로 각각 m_2, m_3, m_4, m_5이다.
3. 따라서 $F = \Sigma m(1, 2, 3, 4, 5)$이다.
4. 3변수 전체 집합은 $\{0, 1, 2, 3, 4, 5, 6, 7\}$이므로 여집합은 $\{0, 6, 7\}$이고, $F = \Pi M(0, 6, 7)$이다.
5. 따라서 $\overline{\Sigma m(0, 6, 7)} = F$이고, $\overline{\Pi M(0, 6, 7)} = \bar{F}$이다.

NoteBase

선지 판단

1번: O. $\Sigma m(1, 2, 3, 4, 5)$는 원식에서 1이 되는 최소항 번호를 그대로 모은 표현이므로 맞다.
2번: O. $\Pi M(1, 2, 3, 4, 5)$는 1,2,3,4,5에서 0이 되는 함수이므로 전체에 보수를 취한 $\overline{\Pi M(1, 2, 3, 4, 5)}$는 1,2,3,4,5에서 1이 되는 F와 같다.
3번: O. $\Sigma m(0, 6, 7)$는 F의 보수 $\bar{F}$이므로, 전체에 보수를 취한 $\overline{\Sigma m(0, 6, 7)}$는 다시 F가 된다.
4번: X. $\Pi M(0, 6, 7)$ 자체가 이미 F인데 여기에 다시 보수를 취하면 $\bar{F}$가 된다. 틀린 표현은 $\overline{\Pi M(0, 6, 7)}$이고, 올바르게는 $\Pi M(0, 6, 7)$이다.

오답 포인트

최대항식은 0이 되는 번호를 적는다는 점을 놓치면 0의 집합 $\{0, 6, 7\}$을 구해 놓고도 보수를 한 번 더 붙이는 실수를 하기 쉽다. 이 문항의 함정은 4번처럼 이미 맞는 최대항식 앞에 막대를 한 번 더 붙여 F와 $\bar{F}$를 뒤바꾸는 데 있다.

다시 나오면 이렇게 판단

곱항을 먼저 이진수로 읽어 최소항 번호를 만든 뒤, 1이 되는 번호 집합과 0이 되는 번호 집합을 분리해서 본다. 그다음 Σm은 1의 집합, ΠM은 0의 집합이라는 규칙으로 판단하면 보수 기호에 흔들리지 않는다.

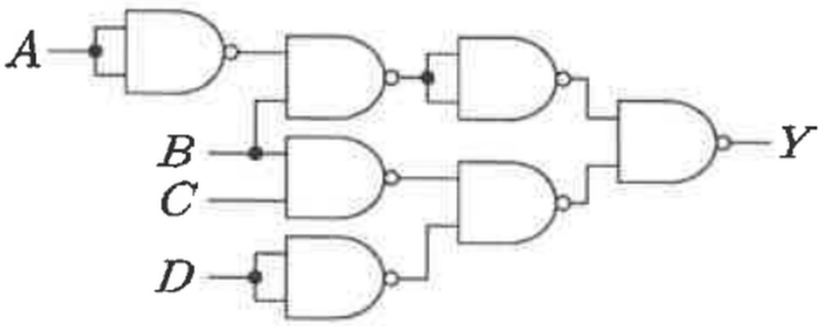
암기 포인트

Σm은 1의 주소, ΠM은 0의 주소를 적는 표기이다.

NoteBase

문 4. <보기>는 $Y(A, B, C, D)$의 논리회로이다. Y의 최소항식에 포함되지 않는 최소항은?

<보기>



① m_2

② m_5

③ m_{11}

④ m_{12}

정답 2번

출제 포인트

NAND 게이트를 인버터와 조합회로로 해석해 부울식을 단순화하고, 그 결과에서 포함되는 최소항과 포함되지 않는 최소항을 판별하는 문제이다.

선지별 해설

① X. m_2는 $A, B, C, D = 0, 0, 1, 0$이다. 이때 $B' = 1$이므로 $Y = A + B' + C'D' = 0 + 1 + 0 = 1$이다. 따라서 m_2는 최소항식에 포함된다.

② O. m_5는 $A, B, C, D = 0, 1, 0, 1$이다. 이때 $A = 0, B' = 0, C'D' = 1 \cdot 0 = 0$이므로 $Y = 0 + 0 + 0 = 0$이다. 따라서 m_5는 최소항식에 포함되지 않는다.

③ X. m_{11}은 $A, B, C, D = 1, 0, 1, 1$이다. $A = 1$이므로 다른 항과 무관하게 $Y = 1$이다. 따라서 m_{11}은 최소항식에 포함된다.

④ X. m_{12}는 $A, B, C, D = 1, 1, 0, 0$이다. $A = 1$이므로 $Y = 1$이다. 따라서 m_{12}는 최소항식에 포함된다.

NoteBase

오답 포인트

입력을 묶은 NAND를 인버터로 보지 않으면 위쪽 가지의 $(A'B)'$가 다시 $A'B$로 복원된다는 점을 놓치기 쉽다. 또한 $(BC)'$과 D'을 NAND 한 결과를 $BC + D$로 바꾸는 단계는 드모르간 정리 $(X'Y')' = X + Y$를 적용한 것임을 분명히 해야 한다. 마지막으로 문제는 최소항식에 포함되는 항이 아니라 포함되지 않는 항을 묻기 때문에, 최종적으로는 $Y = 0$이 되는 조합을 찾아야 한다.

다시 나오면 이렇게 판단

입력을 묶은 NAND는 먼저 인버터로 바꾸고, 각 가지의 출력을 식으로 정리한 뒤 마지막에 $Y = 0$이 되는 입력 조합만 추려 최소항식에서 빠지는 번호를 찾는다.

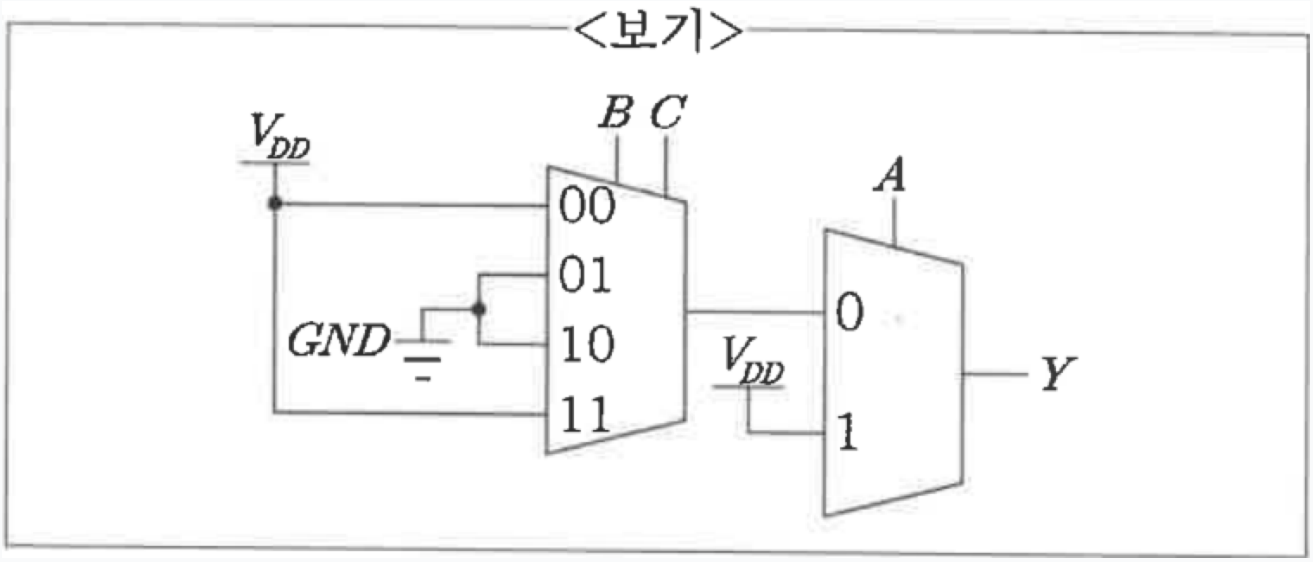
암기 포인트

입력 묶음 NAND는 인버터이고, 포함되지 않는 최소항은 $Y = 0$인 조합의 번호이다.

NoteBase

문 5. <보기>는 멀티플렉서(Multiplexer)를 활용한 회로이다. 이 회로의 출력 Y에 대한 논리식으로 가장 옳은 것은? (단, V_{DD}는 논리값 1, GND는 논리값 0, A~C는 선택 입력 값이며, B는 MSB, C는 LSB이다.)

<보기>



① $Y = A + \overline{BC} + \overline{B}C$

② $Y = \overline{A} + BC + \overline{BC}$

③ $Y = A + \overline{A}C + B\overline{C}$

④ $Y = A + BC + \overline{BC}$

정답 4번

출제 포인트

이 문제는 두 단계 멀티플렉서의 연결을 보고 각 입력선이 만드는 논리식을 합성할 수 있는지를 묻는 문제이다.

선지별 해설

① X. $B\bar{C} + \bar{B}C$는 B와 C가 서로 다를 때 1이 되는 XOR식이다. 그림의 왼쪽 멀티플렉서는 00과 11에서 1이므로 XOR이 아니라 XNOR이어야 한다. 틀린 표현 $B\bar{C} + \bar{B}C \rightarrow$ 올바르게는 $BC + \bar{B}\bar{C}$이다.

② X. 오른쪽 2:1 멀티플렉서는 $A = 1$일 때 1번 입력인 1을 그대로 내보내므로 앞항은 $\bar{A}$가 아니라 A가 된다. 틀린 표현 $\bar{A} \rightarrow$ 올바르게는 A이다.

③ X. 왼쪽 멀티플렉서의 출력은 B, C만으로 정해지므로 $\bar{A}C$ 같은 항이 직접 생기지 않는다. 또한 10에서 0, 11에서 1이므로 $B\bar{C}$ 형태도 맞지 않는다. 틀린 표현 $\bar{A}C + B\bar{C} \rightarrow$ 올바르게는 $BC + \bar{B}\bar{C}$가 포함되어야 한다.

④ O. $A = 1$이면 오른쪽 멀티플렉서가 1을 선택하므로 $Y = 1$이 되고, $A = 0$이면 왼쪽 멀티플렉서 출력 $BC + \bar{B}\bar{C}$가 그대로 나온다. 이를 합치면 $Y = A + BC + \bar{B}\bar{C}$가 된다.

NoteBase

오답 포인트

왼쪽 4:1 멀티플렉서의 1이 들어간 위치가 00과 11이라는 점을 놓치면 XOR과 XNOR를 뒤바꾸기 쉽다. 또한 오른쪽 2:1 멀티플렉서에서 1번 입력이 상수 1이라는 점을 놓치면 A와 $\bar{A}$를 거꾸로 쓰는 실수가 나온다.

다시 나오면 이렇게 판단

멀티플렉서 회로는 각 단계별로 따로 본다. 먼저 선택선과 각 데이터 입력값을 보고 앞단 멀티플렉서의 식을 만든 뒤, 그 출력을 다음 멀티플렉서의 0번 입력과 1번 입력에 대입해 최종식을 합성하면 된다.

암기 포인트

4:1 MUX에서 00,11만 1이면 $BC + \bar{B}\bar{C}$인 XNOR이고, 01,10만 1이면 $B\bar{C} + \bar{B}C$인 XOR이다.

NoteBase

문 6. <보기>의 짝수 병렬 패리티 시스템에서 데이터 블록의 에러 위치를 나타낸 것으로 가장 옳은 것은?

<보기>

data \ bit	7	6	5	4	3	2	1	0	패리티 비트
0	0	1	0	1	0	1	1	1	1
1	1	1	1	0	0	0	1	1	1
2	0	1	0	0	0	0	0	1	1
3	1	1	1	1	0	0	0	0	0
4	1	0	1	1	1	0	1	1	0
5	0	0	0	0	0	1	1	0	0
6	1	1	1	1	1	1	1	0	1
7	0	1	1	1	1	0	0	0	0
패리티 워드	0	0	0	1	1	1	1	0	0

① data 3, bit 5

② data 5, bit 3

③ data 2, bit 5

④ data 5, bit 2

정답 3번

출제 포인트

이 문제는 짝수 병렬 패리티 표에서 행 패리티와 열 패리티가 동시에 어긋나는 위치를 찾아 단일 비트 오류를 특정할 수 있는지를 묻는다.

자료 읽는 순서

1. 각 data 행의 1의 개수와 행 패리티 비트를 합쳐 짝수가 되는지 확인한다.
2. 각 bit 열의 1의 개수와 아래 패리티 워드를 합쳐 짝수가 되는지 확인한다.
3. 어긋난 행과 어긋난 열이 만나는 칸을 오류 위치로 판단한다.

NoteBase

선지별 해설

1번 X: data 3행은 1이 4개이고 패리티 비트가 0이어서 합이 4로 짝수이다. 따라서 오류 행이 아니다. 또한 bit 5열 자체는 오류 열이지만 행이 맞지 않으므로 정답이 아니다. 틀린 표현은 "data 3"이고, 올바르게는 data 2, bit 5이다.
2번 X: data 5행은 1이 2개이고 패리티 비트가 0이어서 합이 2로 짝수이다. 따라서 오류 행이 아니다. 틀린 표현은 "data 5, bit 3"이고, 올바르게는 data 2, bit 5이다.
3번 O: data 2행은 1이 2개인데 패리티 비트가 1이어서 합이 3으로 홀수가 된다. bit 5열은 열의 1 개수가 5개이고 아래 패리티 워드가 0이어서 전체가 홀수가 된다. 따라서 교차점인 data 2, bit 5가 오류 위치이다.
4번 X: data 5행은 오류 행이 아니고, bit 2열은 열의 1 개수가 3개인데 아래 패리티 워드가 1이어서 합이 4로 짝수이므로 오류 열도 아니다. 틀린 표현은 "data 5, bit 2"이고, 올바르게는 data 2, bit 5이다.

오답 포인트

행 패리티만 보고 data 2까지는 찾고도 열 패리티를 끝까지 세지 않거나, bit 번호를 오른쪽의 0부터 읽지 않고 반대로 읽으면 다른 선택지로 흔들리기 쉽다.

다시 나오면 이렇게 판단

병렬 패리티 문항은 먼저 이상이 있는 data 행을 찾고, 다음으로 이상이 있는 bit 열을 찾은 뒤, 두 위치의 교차점을 단일 오류 비트로 판단한다.

암기 포인트

단일 비트 오류는 이상 행 1개와 이상 열 1개의 교차점으로 잡힌다.

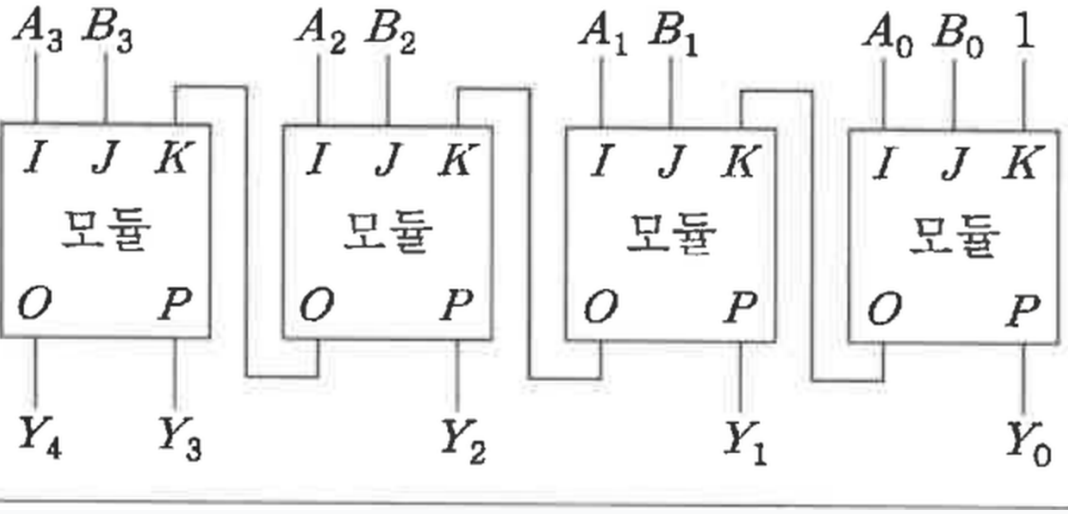
NoteBase

문 7. <보기 1>은 입력 I, J, K에 대한 출력 O, P의 진리표이며, 이를 따르는 모듈 4개를 <보기 2>와 같이 연결했다. $A_3A_2A_1A_0 = 1011_{(2)}$, $Y_4Y_3Y_2Y_1Y_0 = 11001_{(2)}$일 때, $B_3B_2B_1B_0$의 값은?

<보기 1>

입력			출력	
I	J	K	O	P
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

<보기 2>



① $1101_{(2)}$

② $1100_{(2)}$

③ $1011_{(2)}$

④ $1110_{(2)}$

NoteBase

정답 1번

출제 포인트

이 문제는 <보기 1>의 진리표가 어떤 회로인지 먼저 규정한 뒤, <보기 2>의 직렬 연결에서 자리올림이 어떻게 전달되는지를 추적해 $B_3B_2B_1B_0$를 구하는 문제이다. 진리표를 보면 출력 O는 입력 I, J, K 중 1이 두 개 이상일 때 1이 되는 다수결(majority) 함수이고, 출력 P는 1의 개수가 홀수일 때 1이 되는 $I \oplus J \oplus K$이다. 즉 각 모듈은 O가 자리올림(carry), P가 합(sum)인 전가산기이며, $O = IJ + JK + IK$, $P = I \oplus J \oplus K$로 정리된다. <보기 2>에서는 한 모듈의 O가 왼쪽 모듈의 K로 들어가는 4비트 리플 캐리 가산기이고, 맨 오른쪽 모듈의 K가 1로 고정되어 초기 자리올림이 1이다. 따라서 합 비트는 $Y_0 = P_0$, $Y_1 = P_1$, $Y_2 = P_2$, $Y_3 = P_3$(맨 왼쪽 모듈의 P), 최종 자리올림은 $Y_4 = O$(맨 왼쪽 모듈의 O)가 되어 전체 출력은 $A + B + 1$을 나타낸다.

선지별 해설

① O

- 전체 관계식은 $A + B + 1 = Y_4Y_3Y_2Y_1Y_0$이다. $A = 1011_{(2)} = 11$, $Y_4Y_3Y_2Y_1Y_0 = 11001_{(2)} = 25$이므로 $B = 25 - 11 - 1 = 13 = 1101_{(2)}$이다.
- 검산하면 $1011 + 1101 + 1 = 11001$로 주어진 $Y_4Y_3Y_2Y_1Y_0$와 정확히 일치한다. 비트별로 보아도 초기 carry 1에서 $Y_0 = 1$이 되려면 $B_0 = 1$, 이어 $Y_1 = 0$에서 $B_1 = 0$, $Y_2 = 0$에서 $B_2 = 1$, $Y_3 = 1$에서 $B_3 = 1$이 되어 $B_3B_2B_1B_0 = 1101$이다.

② X

- $1011 + 1100 + 1 = 11000$이므로 $Y_0 = 0$이 되어 주어진 출력 $Y_0 = 1$과 다르다.
- 1100은 정답 1101에서 B_0을 0으로 잘못 둔 값으로, 초기 carry 1을 빼먹고 계산하면 나오기 쉬운 오답이다.

③ X

- $1011 + 1011 + 1 = 10111$이므로 주어진 11001과 맞지 않는다.
- 1011은 A와 같은 값으로, B를 따로 구하지 않고 입력 A를 그대로 옮겨 적은 오답이다.

④ X

- $1011 + 1110 + 1 = 11010$이므로 $Y_1 = 1$, $Y_0 = 0$이 되어 주어진 출력과 어긋난다.
- 1110은 carry 전달 방향을 반대로 보거나 비트 자리를 어긋나게 맞췄을 때 나오는 값이다.

오답 포인트

진리표에서 O(carry)와 P(sum)의 역할을 거꾸로 읽으면 합 비트와 자리올림 비트를 뒤바꿔 오답이 나온다. 또한 맨 오른쪽 $K = 1$을 초기 자리올림으로 보지 않으면 $A + B + 1$이 아니라 $A + B$로 계산해 B_0부터 어긋난다. 각 모듈을 독립적으로 보지 말고 오른쪽에서 왼쪽으로 carry가 전달되는 연결 구조를 함께 읽어야 한다는 점이 함정이다.

NoteBase

다시 나오면 이렇게 판단

입력 3개와 출력 2개의 진리표가 나오면 먼저 O가 다수결(carry), P가 $I \oplus J \oplus K$(sum)인 전가산기인지 확인한다. 그다음 맨 오른쪽 초기 carry부터 sum과 carry를 한 자리씩 추적하거나, 회로 전체를 A+B+(초기 carry)로 한 번에 보고 미지수만 역산하면 빠르게 풀린다.

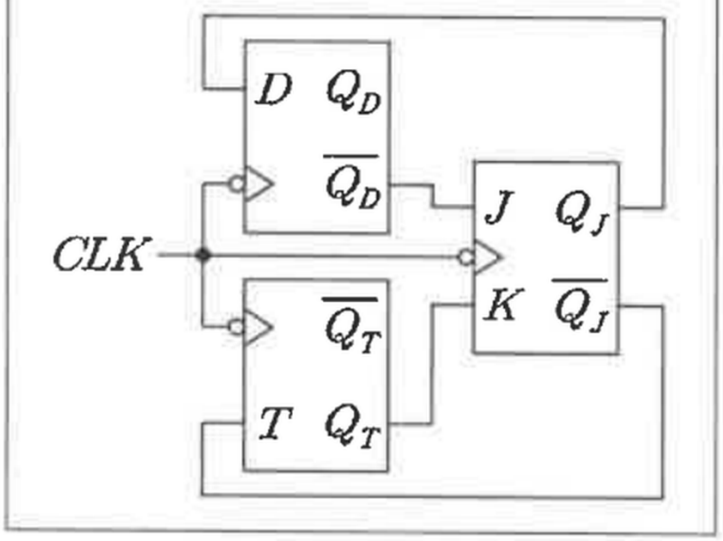
암기 포인트

전가산기에서 $O = IJ + JK + IK$(carry), $P = I \oplus J \oplus K$(sum)이다. 전가산기 4개를 직렬로 연결하면 리플 캐리 가산기가 되고, 맨 오른쪽 carry 입력이 1이면 전체 출력은 $A + B + 1$이 된다.

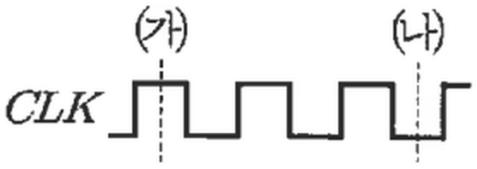
NoteBase

문 8. <보기 1>의 Q_D, Q_T, Q_J는 <보기 2>의 (가) 시점에 모두 0이다. (나) 시점에 Q_T, Q_J의 값을 옳게 짝지은 것은? (단, (가), (나) 시점에 Q_D, Q_T, Q_J는 정상상태(steady state)이다.)

<보기 1>



<보기 2>



① Q_T: 0, Q_J: 0

② Q_T: 0, Q_J: 1

③ Q_T: 1, Q_J: 0

④ Q_T: 1, Q_J: 1

정답 1번

출제 포인트

D·T·JK 플립플롭이 혼합된 동기식 순서 회로에서, 초기 상태 $Q_D = Q_T = Q_J = 0$으로부터 CLK 하강 에지 4회 후 $Q_T \cdot Q_J$ 값을 계산한다. 회로도의 J·K 핀 앞 버블(인버터 기호)을 정확히 읽어 유효 입력을 결정하는 것이 핵심이다.

NoteBase

조건 정리

(가) 시점 초기 상태: $Q_D = 0,\ Q_T = 0,\ Q_J = 0$
세 플립플롭 모두 CLK 하강 에지에 동기 동작한다.
회로 연결:
· D FF 입력: $D = \bar{Q}_J$
· T FF 입력: $T = Q_J$
· JK FF: J 핀 앞 버블(인버터) → 유효 $J = Q_D$（$\bar{Q}_D$가 버블에서 재반전）, K 핀 앞 버블 → 유효 $K = Q_T$（$\bar{Q}_T$가 버블에서 재반전）
<보기 2> 타이밍 다이어그램에서 (가)~(나) 사이에 CLK 하강 에지 4회 발생

사용할 공식

D FF: $Q_D(t+1) = D = \bar{Q}_J(t)$
T FF: $Q_T(t+1) = Q_J(t) \oplus Q_T(t)$
JK FF: $J = 0,\ K = 0$→유지 / $J = 0,\ K = 1$→리셋($Q_J = 0$) / $J = 1,\ K = 0$→셋($Q_J = 1$) / $J = 1,\ K = 1$→토글
단, J·K는 버블 처리 후 유효값 $J = Q_D,\ K = Q_T$를 대입한다.

NoteBase

계산 과정

초기: $(Q_D,\ Q_T,\ Q_J) = (0,\ 0,\ 0)$

[1번째 하강 에지]
$D = \bar{Q}_J = 1 \rightarrow Q_{D,\text{next}} = 1$
$T = Q_J = 0 \rightarrow Q_{T,\text{next}} = 0 \oplus 0 = 0$
유효 $J = Q_D = 0,\ K = Q_T = 0 \rightarrow$ JK 유지 $\rightarrow Q_{J,\text{next}} = 0$
상태: $(1,\ 0,\ 0)$

[2번째 하강 에지]
$D = \bar{Q}_J = 1 \rightarrow Q_{D,\text{next}} = 1$
$T = Q_J = 0 \rightarrow Q_{T,\text{next}} = 0$
유효 $J = Q_D = 1,\ K = Q_T = 0 \rightarrow$ JK 셋 $\rightarrow Q_{J,\text{next}} = 1$
상태: $(1,\ 0,\ 1)$

[3번째 하강 에지]
$D = \bar{Q}_J = 0 \rightarrow Q_{D,\text{next}} = 0$
$T = Q_J = 1 \rightarrow Q_{T,\text{next}} = 1 \oplus 0 = 1$
유효 $J = Q_D = 1,\ K = Q_T = 0 \rightarrow$ JK 셋 유지 $\rightarrow Q_{J,\text{next}} = 1$
상태: $(0,\ 1,\ 1)$

[4번째 하강 에지]
$D = \bar{Q}_J = 0 \rightarrow Q_{D,\text{next}} = 0$
$T = Q_J = 1 \rightarrow Q_{T,\text{next}} = 1 \oplus 1 = 0$
유효 $J = Q_D = 0,\ K = Q_T = 1 \rightarrow$ JK 리셋 $\rightarrow Q_{J,\text{next}} = 0$
상태: $(0,\ 0,\ 0)$

(나) 시점 정상상태: $Q_T = 0,\ Q_J = 0$

선지 판단

① $Q_T = 0,\ Q_J = 0$: (나) 시점 상태 $(0, 0, 0)$과 일치한다. ○
② $Q_T = 0,\ Q_J = 1$: 2번째 에지 후 상태 $(1, 0, 1)$에서 $Q_T = 0 \cdot Q_J = 1$이지만, (나)는 4번째 에지 후이므로 틀리다. ☒
③ $Q_T = 1,\ Q_J = 0$: 이 회로의 상태 순서 (0,0,0)→(1,0,0)→(1,0,1)→(0,1,1)→(0,0,0)에서 $Q_T = 1$이면서 $Q_J = 0$인 상태는 나타나지 않는다. ☒
④ $Q_T = 1,\ Q_J = 1$: 3번째 에지 후 상태 $(0, 1, 1)$이지만, 4번째 에지 후에는 이 상태에서 벗어난다. ☒

NoteBase

오답 포인트

J·K 핀 앞 버블을 무시하고 $J = \bar{Q}_D \cdot K = \bar{Q}_T$로 계산하면 상태 전이가 달라져 오답에 도달한다. 버블은 반드시 인버터로 처리해야 하며, $\bar{Q}_D$를 버블 있는 J 핀에 연결하면 유효 J 입력은 Q_D가 된다. 또한 <보기 2> 다이어그램에서 하강 에지를 3회 또는 5회로 잘못 세면 ④ 또는 ①($Q_D = 1$인 다른 상태)에 도달한다.

다시 나오면 이렇게 판단

①회로도에서 각 플립플롭 입력 연결을 추적하고 버블 유무를 확인한다. ②타이밍 다이어그램에서 유효 CLK 에지(이 회로는 하강 에지) 수를 정확히 센다. ③동기식 회로에서는 에지 직전의 현재 상태로 모든 플립플롭을 동시 갱신하는 상태 전이표를 에지 수만큼 단계별로 작성한다.

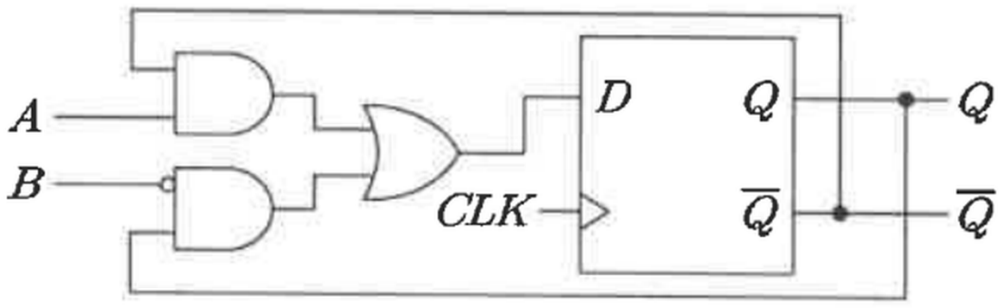
암기 포인트

JK FF 진리표: J=0·K=0=유지, J=0·K=1=리셋, J=1·K=0=셋, J=1·K=1=토글. T FF: T=0=유지, T=1=토글. D FF: 다음 Q=D. 핀 앞 버블=인버터. $\bar{Q}$를 버블 있는 핀에 연결하면 유효 입력은 Q이다.

NoteBase

문 9. <보기>의 회로에 대한 설명으로 가장 옳지 않은 것은? (단, $Q(t)$는 현재의 Q값, $Q(t+1)$은 CLK 신호의 다음 상승 에지 직후의 Q값을 의미한다.)

<보기>



① $Q(t)=0$에서 $Q(t+1)=0$이 되려면 현재 $A=0, B=0$이면 된다.

② $Q(t)=0$에서 $Q(t+1)=1$이 되려면 현재 $A=1$이어야 하며, B는 아무 값이어도 된다.

③ $Q(t)=1$에서 $Q(t+1)=0$이 되려면 현재 $A=0, B=1$이면 된다.

④ $Q(t)=1$에서 $Q(t+1)=1$이 되려면 현재 $A=1, B=1$이면 된다.

정답 4번

출제 포인트

D 플립플롭의 입력식을 현재 상태 $Q(t)$와 입력 A, B로 나타내고, 그 식으로 다음 상태 $Q(t+1)$를 판정하는 문제이다.

선지별 해설

① O. $Q(t)=0$이면 $Q(t+1)=A$이다. 따라서 $A=0, B=0$이면 $Q(t+1)=0$이 된다. 이 선지는 참이며, 더 정확히는 B와 무관하게 A=0이면 된다.

② O. $Q(t)=0$이면 $Q(t+1)=A$이므로 0에서 1로 가려면 A=1이어야 한다. B는 식에 들어가지 않으므로 아무 값이어도 된다.

③ O. $Q(t)=1$이면 $Q(t+1)=\overline{B}$이다. 따라서 $A=0, B=1$이면 $Q(t+1)=0$이 된다. 이 선지는 참이며, 더 정확히는 A와 무관하게 B=1이면 된다.

④ X. $Q(t)=1$이면 $Q(t+1)=\overline{B}$이므로 B=1일 때 다음 상태는 1이 아니라 0이 된다. 틀린 표현은 \"A=1, B=1이면 $Q(t+1)=1$\"이고, 올바르게는 B=0이어야 $Q(t+1)=1$이며 A는 아무 값이어도 된다.

NoteBase

오답 포인트

하단 입력의 버블을 놓치면 $\overline{B}$ 대신 B로 읽어 ④를 맞는 선지처럼 오해하기 쉽다. 또한 ①과 ③은 필요조건이 아니라 충분조건을 말하는 문장인데, 이를 필요충분조건처럼 읽으면 불필요하게 의심하게 된다.

다시 나오면 이렇게 판단

먼저 피드백을 포함한 D 입력식을 만든 뒤 현재 상태별로 나누어 본다. 이 회로는 $Q(t) = 0$이면 $Q(t+1) = A$, $Q(t) = 1$이면 $Q(t+1) = \overline{B}$로 바로 단순화되므로 각 선지를 즉시 판정할 수 있다.

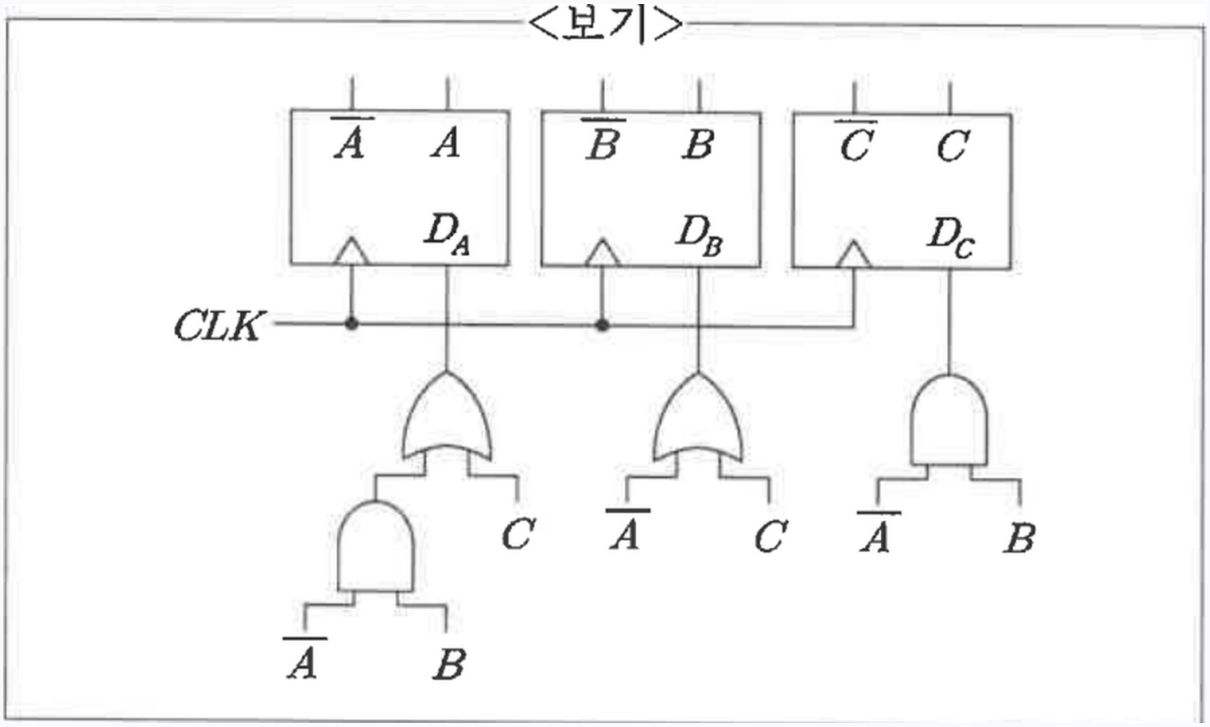
암기 포인트

D 플립플롭은 다음 상태가 곧 D이고, 입력 버블은 보수를 뜻하므로 이 회로의 핵심식은 $Q(t+1) = A\overline{Q}(t) + \overline{B}Q(t)$이다.

NoteBase

문 10. <보기>는 D 플립플롭 기반의 임의순서 카운터이다. ABC의 초기값이 000일 때, CLK 신호의 상승에지가 여러 번 지나가더라도 절대 ABC의 값이 될 수 없는 것은? (단, A는 MSB, C는 LSB이다.)

<보기>



① 110

② 010

③ 001

④ 111

정답 3번

출제 포인트

이 문제는 D 플립플롭 입력식을 회로도에서 읽어 현재 상태 000에서 시작하는 다음 상태 순환을 추적하는 문제이다.

NoteBase

선지별 해설

① X: 110은 도달 가능한 상태이다. 000 → 010 → 111 → 110으로 3번째 상승 에지 후 나타난다. 틀린 판단은 '110은 절대 될 수 없다'이고, 올바르게는 '110은 순환 상태에 포함된다'이다.
② X: 010은 도달 가능한 상태이다. 초기상태 000에서 1번째 상승 에지 직후 바로 010이 된다. 틀린 판단은 '010은 절대 될 수 없다'이고, 올바르게는 '010은 최초로 도달하는 상태'이다.
③ O: 001은 순환에 포함되지 않는다. 000에서 출발한 상태천이가 000, 010, 111, 110만 반복하므로 001은 여러 번의 클록이 지나도 나타나지 않는다.
④ X: 111은 도달 가능한 상태이다. 000 → 010 → 111로 2번째 상승 에지 후 나타난다. 틀린 판단은 '111은 절대 될 수 없다'이고, 올바르게는 '111은 순환 상태에 포함된다'이다.

오답 포인트

출제자가 노린 함정은 각 비트를 비동기적으로 바뀐다고 착각하거나, 한 플립플롭의 출력 변화가 같은 클록 안에서 다른 플립플롭 입력에 즉시 반영된다고 오해하는 지점이다. 이 회로는 동기식이므로 현재 상태의 A, B, C를 기준으로 세 D 입력을 먼저 계산한 뒤, 상승 에지에서 동시에 갱신해야 한다.

다시 나오면 이렇게 판단

회로도형 카운터는 먼저 각 플립플롭의 다음 상태식 (D_A, D_B, D_C)를 읽고, 초기상태를 대입해 한 클록씩 상태표를 만든다. 한 번 순환이 닫히면 그 밖의 상태는 도달 불가능 상태로 판단할 수 있다.

암기 포인트

동기식 D 플립플롭 카운터는 현재 상태로 다음 상태를 계산하고, 닫힌 상태순환에 없는 코드는 도달할 수 없는 상태이다.

NoteBase

문 11. 비가중치 코드에 해당하지 않는 것은?

① 3 초과 코드(excess-3 code)

② BCD 코드(BCD code)

③ 그레이 코드(gray code)

④ 5중 2코드(2-out-of-5 code)

정답 2번

출제 포인트

이 문제는 가중치 코드와 비가중치 코드를 구분할 수 있는지를 묻는 문제이다.

핵심 개념

가중치 코드는 각 비트 자리에 정해진 가중치가 있어 그 합으로 숫자값을 읽는 코드이다. 비가중치 코드는 비트 패턴에 고정된 자리 가중치가 없어 합으로 값을 직접 읽지 않는 코드이다. BCD는 대표적인 가중치 코드이고, Excess-3 코드·그레이 코드·5중 2코드는 비가중치 코드로 분류한다.

선지별 해설

1번 O: 3 초과 코드는 각 10진수에 3을 더해 표현하는 코드로, 비트 자리의 합으로 값을 직접 읽는 방식이 아니다. 따라서 비가중치 코드에 해당한다.
2번 X: 틀린 부분은 '비가중치 코드에 해당한다'는 판단이다. BCD 코드는 8421처럼 각 비트에 고정 가중치가 부여되는 대표적인 가중치 코드이다. 올바르게는 'BCD 코드는 비가중치 코드가 아니라 가중치 코드이다.'이다.
3번 O: 그레이 코드는 인접한 수가 1비트만 변하도록 만든 코드로, 비트의 가중치 합으로 수를 읽는 코드가 아니다. 따라서 비가중치 코드에 해당한다.
4번 O: 5중 2코드는 5비트 중 1의 개수가 항상 2개가 되도록 만든 코드로, 자리 가중치의 합으로 값을 나타내는 코드가 아니다. 따라서 비가중치 코드에 해당한다.

오답 포인트

BCD가 10진수 표현용 코드라는 이유만으로 비가중치 코드라고 착각하면 틀린다. 이 문제의 함정은 '숫자를 표현하는 코드'인지가 아니라 '각 비트에 고정 가중치가 있느냐'이다.

다시 나오면 이렇게 판단

코드 이름보다 먼저 각 비트 자리에 8, 4, 2, 1처럼 고정된 자리값이 있는지를 본다. 자리값의 합으로 숫자를 읽으면 가중치 코드이고, 그렇지 않으면 비가중치 코드로 판단한다.

NoteBase

암기 포인트

BCD는 가중치, Excess-3·Gray·2-out-of-5는 비가중치로 정리된다.

NoteBase

문 12. 8비트 시스템에서 $(-16) \div 2^2$을 구현하는 시프트 연산으로 가장 옳은 것은? (단, 음수는 2의 보수로 표현하고, >> 기호는 논리 오른쪽 시프트(Logical shift right) 연산자, >>> 기호는 산술 오른쪽 시프트(Arithmetic shift right) 연산자를 의미하고, 기호 뒤의 숫자는 시프트하는 비트의 개수이다.)

① $00010000_{(2)} >> 2$

② $00010000_{(2)} >>> 2$

③ $11110000_{(2)} >> 2$

④ $11110000_{(2)} >>> 2$

정답 4번

출제 포인트

이 문제는 8비트 2의 보수 표현에서 음수를 2^n으로 나눌 때 산술 오른쪽 시프트를 사용해야 함을 묻는 문제이다.

조건 정리

주어진 값은 -16과 $2^2 = 4$이다. 8비트에서 -16의 2의 보수 표현은 `11110000(2)`이다. 구해야 할 값은 $(-16) \div 4 = -4$를 만드는 시프트 연산이다. 음수 나눗셈이므로 부호 비트를 유지하는 산술 오른쪽 시프트가 필요하다.

사용할 공식

2의 보수로 표현된 정수를 2^n으로 나눌 때는 산술 오른쪽으로 n비트 시프트한다. 이 문제는 -16을 4로 나누는 상황이므로 오른쪽으로 2비트 이동시키면서 최상위 부호 비트 1을 유지해야 한다.

계산 과정

1. -16을 8비트 2의 보수로 쓰면 `11110000(2)`이다.
2. 이를 산술 오른쪽으로 2비트 시프트하면 `11111100(2)`가 된다.
3. `11111100(2)`는 8비트 2의 보수에서 -4이므로 $(-16) \div 2^2$와 일치한다.

NoteBase

선지별 해설

① X: `00010000(2)`는 $+16$의 표현이므로 피연산자부터 틀렸다. 논리 오른쪽 시프트를 해도 `00000100(2)`가 되어 $+4$가 나온다. 틀린 표현은 `00010000(2)`이고, 올바르게는 `11110000(2)`를 사용해야 한다.
② X: 산술 오른쪽 시프트를 선택한 점은 맞지만 시작 비트열 `00010000(2)`가 $+16$이어서 문제의 -16을 구현하지 못한다. 틀린 표현은 `00010000(2)`이고, 올바르게는 `11110000(2)`를 산술 오른쪽으로 2비트 시프트해야 한다.
③ X: `11110000(2)`는 -16의 표현이라 시작값은 맞지만 논리 오른쪽 시프트는 왼쪽을 0으로 채워 `00111100(2)`가 되므로 $+60$이 된다. 틀린 표현은 `논리 오른쪽 시프트`이고, 올바르게는 `산술 오른쪽 시프트`를 사용해야 한다.
④ O: `11110000(2)`를 산술 오른쪽으로 2비트 시프트하면 부호 비트 1이 유지되어 `11111100(2)`가 된다. 이는 -4이므로 $(-16) \div 2^2$를 정확히 구현한다.

오답 포인트

음수의 2의 보수 표현과 논리 시프트, 산술 시프트를 한 번에 묶어 헷갈리기 쉬운 문제이다. 특히 `11110000(2)`까지는 맞게 잡아 놓고도 논리 오른쪽 시프트를 선택하면 부호가 깨져 전혀 다른 양수가 나온다는 점이 함정이다.

다시 나오면 이렇게 판단

음수를 2^n으로 나누는 시프트 문제는 먼저 원수를 2의 보수 비트열로 바꾸고, 그다음 부호 유지가 필요한지 확인한 뒤, 음수이면 산술 오른쪽 시프트를 적용하는 순서로 판단한다.

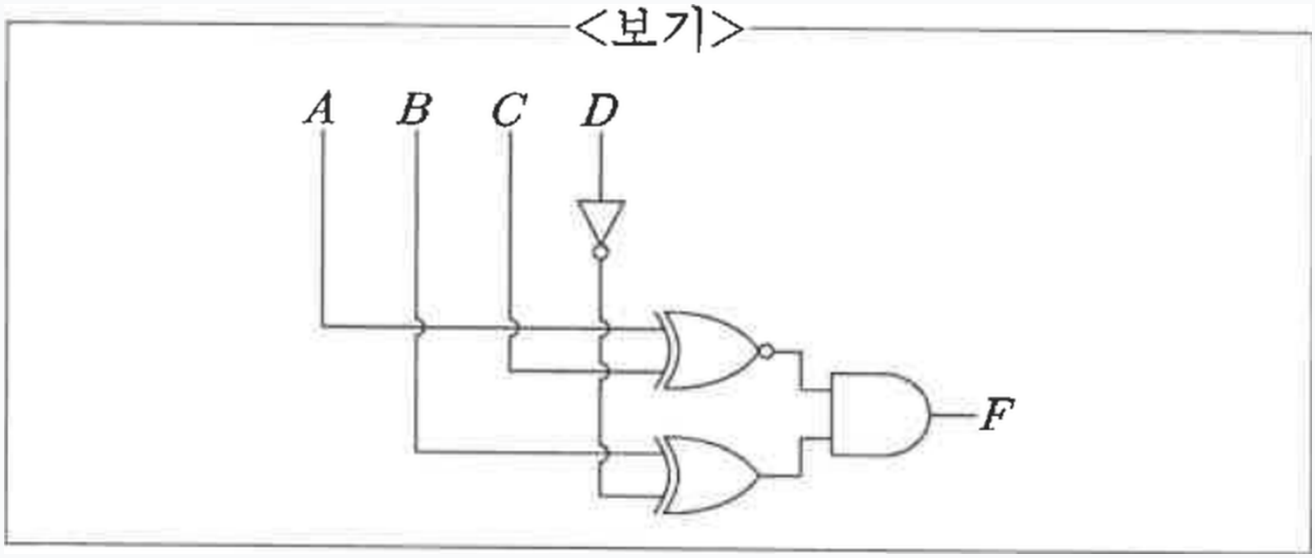
암기 포인트

음수의 2^n 나눗셈은 2의 보수 표현 후 산술 오른쪽 시프트이다.

NoteBase

문 13. <보기>에 제시한 논리회로의 부울함수식을 최소항의 합 형태로 나타낼 때 가장 옳은 것은?

<보기>



① $F = \overline{A}BC\overline{D} + \overline{A}B\overline{C}D + A\overline{B}C\overline{D} + ABCD$

② $F = \overline{A}BC\overline{D} + \overline{A}B\overline{C} + A\overline{B}\overline{D} + ABCD$

③ $F = \overline{A}B\overline{C}\overline{D} + \overline{A}B\overline{C}D + A\overline{B}C\overline{D} + ABCD$

④ $F = \overline{A}B\overline{C} + A\overline{B}C\overline{D} + ABCD$

정답 1번

출제 포인트

이 문제는 회로도에서 게이트 종류와 입력 연결을 읽은 뒤, 부울식을 전개하여 최소합의 형태로 쓸 수 있는지를 묻는다.

조건 정리

위쪽 게이트는 출력에 버블이 있는 XOR이므로 A와 C의 XNOR이다. 아래쪽 게이트는 B와 인버터를 지난 D의 XOR이다. 마지막 게이트는 두 출력의 AND이므로 최종 함수는 두 중간식의 곱으로 정리된다.

사용할 공식

XNOR는 $A \odot C = AC + \overline{A}\overline{C}$이다. XOR는 $B \oplus \overline{D} = B\overline{\overline{D}} + \overline{B}\overline{D} = BD + \overline{B}\overline{D}$이다. 이 문제는 각 게이트 출력을 먼저 식으로 바꾼 뒤 분배법칙으로 전개하면 최소합을 바로 확인할 수 있다.

NoteBase

계산 과정

위쪽 출력은 $X = A \odot C = AC + \overline{AC}$이다. 아래쪽 출력은 $Y = B \oplus \overline{D} = BD + \overline{BD}$이다. 따라서 $F = XY = (AC + \overline{AC})(BD + \overline{BD})$이다. 전개하면 $F = ABCD + A\overline{B}C\overline{D} + \overline{A}B\overline{C}D + \overline{ABCD}$이다. 이는 카르노맵으로 보면 $m(0, 5, 10, 15)$만 1인 형태라 서로 인접한 묶음이 없어 더 줄어들지 않는다.

선지 판단

1번: O. $\overline{ABCD} + \overline{A}B\overline{C}D + A\overline{B}C\overline{D} + ABCD$로 전개 결과와 정확히 일치한다.
2번: X. ②의 $\overline{A}B\overline{C}$처럼 3변수항이 들어가면 D 조건이 사라져 없는 조합까지 1로 만들게 된다. 이 회로의 1은 네 개의 고립된 최소항뿐이므로 3변수항으로 줄어들 수 없다. 올바르게는 $\overline{ABCD} + \overline{A}B\overline{C}D + A\overline{B}C\overline{D} + ABCD$이다.
3번: X. ③은 첫 항이 $\overline{A}B\overline{CD}$로 되어 있어 $m0$를 놓치고 $m4$를 잘못 포함한다. 첫 항은 $\overline{ABCD}$이어야 하므로 전체 식이 틀린다.
4번: X. ④의 $\overline{A}B\overline{C}$는 역시 3변수항이라 D의 조건을 잃는다. 실제로는 $\overline{A}B\overline{C}D$와 $\overline{ABCD}$처럼 4변수 최소항 두 개가 따로 남아야 한다.

오답 포인트

출력 버블이 달린 XOR를 XNOR로 읽지 못하거나, 아래 게이트 입력을 $B \oplus D$로 읽으면 보수 극성이 전부 뒤바뀐다. 또 카르노맵에서 고립된 1을 무리하게 묶어 3변수항으로 줄이면 ②, ④ 같은 오답이 된다.

다시 나오면 이렇게 판단

회로형 부울식 문제는 게이트 종류 확인, 각 입력선 추적, 중간식 작성, 최종식 전개, 카르노맵 인접 여부 확인의 순서로 판단하면 안정적으로 풀린다.

암기 포인트

XNOR는 같으면 1, XOR는 다르면 1이며, 카르노맵에서 고립된 1은 4변수 최소항으로 그대로 남는다.

NoteBase

문 14. <보기>는 서로 다른 부울 변수 A, B, C에 대한 함수 G, H이다. $G+H$를 최대로 간략화한 논리식으로 가장 옳은 것은?

<보기>

$G(A,B,C)=m_0+m_2+m_5+m_7$

$H(A,B,C)=M_1M_3$

① $A+B+C$

② ABC

③ $A+\overline{C}$

④ $A\overline{C}$

정답 3번

출제 포인트

이 문제는 최소항과 최대항으로 주어진 불 함수를 표준형에서 간략형으로 바꾼 뒤, 두 함수를 OR하여 흡수법칙으로 최종식을 정리할 수 있는지를 묻는다.

조건 정리

주어진 식은 $G(A,B,C)=\Sigma m(0,2,5,7)$, $H(A,B,C)=\Pi M(1,3)$이다.
구해야 할 값은 $G+H$의 최대 간략형이다.
Σm은 해당 최소항에서 값이 1임을 뜻하고, ΠM은 해당 최대항에서 값이 0임을 뜻한다.

사용할 공식

최소항 묶기, 최대항의 곱형 전개, 흡수법칙 $X+XY=X$, 항등식 $(X+B)(X+\overline{B})=X$를 사용한다.
이 공식을 쓰는 이유는 G는 1이 되는 조합을 묶어 간단히 만들 수 있고, H는 0이 되는 조합을 최대항으로 표현한 뒤 바로 간략화할 수 있기 때문이다.

계산 과정

1. $G=m_0+m_2+m_5+m_7$이므로 m_0, m_2를 묶으면 $\overline{AC}$, m_5, m_7을 묶으면 AC가 되어 $G=\overline{AC}+AC$가 된다.
2. $H=M_1M_3=(A+B+\overline{C})(A+\overline{B}+\overline{C})$이다. 여기서 $X=A+\overline{C}$로 두면 $(X+B)(X+\overline{B})=X$이므로 $H=A+\overline{C}$가 된다.
3. 따라서 $G+H=(\overline{AC}+AC)+(A+\overline{C})$이다.
4. $A+AC=A$, $A+\overline{AC}=A+\overline{C}$이므로 최종적으로 $G+H=A+\overline{C}$가 된다.

NoteBase

선지 판단

1번 X: $A+B+C$는 범위가 너무 넓다. 예를 들어 $(A,B,C)=(0,1,1)$이면 1번 식은 1이지만, 이 경우는 m_3이고 $H=0, G=0$이므로 실제 $G+H=0$이다. 틀린 표현은 $B+C$가 불필요하게 포함된 점이고, 올바르게는 $A+\overline{C}$이다.
2번 X: ABC는 $(1,1,1)$ 한 경우만 1이 되는 너무 좁은 식이다. 예를 들어 $(1,0,0)$이면 $H=A+\overline{C}=1$이어서 실제 $G+H=1$인데 2번 식은 0이다. 올바르게는 $A+\overline{C}$이다.
3번 O: 계산 결과가 그대로 $A+\overline{C}$이므로 옳다. $A=1$이거나 $C=0$이면 출력이 1이 되는 형태와 정확히 일치한다.
4번 X: $\overline{A}C$는 오히려 H가 0이 되는 영역을 나타낸다. $(A,C)=(0,1)$일 때만 1이 되는데, 실제 정답식 $A+\overline{C}$는 그 경우 0이 된다. 틀린 표현은 부정 위치가 반대라는 점이고, 올바르게는 $A+\overline{C}$이다.

오답 포인트

최소항은 1의 위치, 최대항은 0의 위치라는 점을 뒤섞으면 바로 오답이 된다. 특히 4번처럼 $\overline{A}C$를 보면 정답식의 여집합 영역을 그대로 고른 실수가 나오기 쉽다.

다시 나오면 이렇게 판단

불 함수 간략화 문제는 먼저 Σm인지 ΠM인지 구분하고, 각 함수를 따로 간략화한 뒤 마지막에 흡수법칙으로 한 번 더 줄이는 순서로 판단한다.

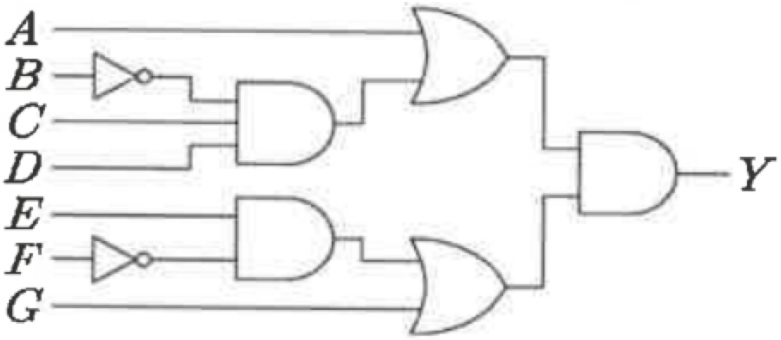
암기 포인트

Σm은 1의 위치, ΠM은 0의 위치이며, $(X+B)(X+\overline{B})=X$, $X+XY=X$가 간략화의 핵심이다.

NoteBase

문 15. <보기>는 Y의 논리회로이다. 최소 곱의 합 형태로 나타낸 $\overline{Y}$의 논리식에 포함되는 항은?

<보기>



① $\overline{AC}$

② $\overline{AB}$

③ $A\overline{D}$

④ $\overline{E}G$

정답 1번

출제 포인트

이 문제는 게이트 회로를 식으로 바꾼 뒤 드모르간 법칙으로 $\overline{Y}$를 최소 곱의 합 형태로 전개할 수 있는지를 묻는다.

선지별 해설

① O: 회로식은 $Y = (A + \overline{B}CD)(E\overline{F} + G)$이다. 따라서 $\overline{Y} = \overline{A + \overline{B}CD} + \overline{E\overline{F} + G} = \overline{A}(B + \overline{C} + \overline{D}) + (\overline{E} + F)\overline{G}$이고, 이를 전개하면 $\overline{A}B + \overline{AC} + \overline{AD} + \overline{EG} + F\overline{G}$가 된다. 따라서 $\overline{AC}$는 포함된다.

② X: $\overline{AB}$ 항은 나오지 않는다. 첫 번째 괄호를 보수 취하면 $\overline{A}(B + \overline{C} + \overline{D})$가 되므로 B는 그대로 살아 있고 $\overline{B}$로 바뀌지 않는다. 틀린 표현은 $\overline{B}$이고, 올바르게는 B이므로 포함되는 항은 $\overline{A}B$이다.

③ X: 전개 결과에는 $\overline{AD}$가 포함되고 $A\overline{D}$는 포함되지 않는다. 틀린 표현은 A이고, 올바르게는 $\overline{A}$이다.

④ X: 두 번째 괄호의 보수는 $(E\overline{F} + G)' = (\overline{E} + F)\overline{G}$이므로 포함되는 항은 $\overline{EG}$ 또는 $F\overline{G}$이다. 틀린 표현은 G이고, 올바르게는 $\overline{G}$이다.

오답 포인트

이 문제의 함정은 회로에서 바로 $\overline{Y}$를 읽으려 하거나, 드모르간 적용 뒤 문자의 부정을 한 번 더 잘못 붙이는 데 있다. 특히 $(A + X)' = \overline{A}\overline{X}$에서 $\overline{X}$를 전개할 때 $\overline{B}CD$가 $B + \overline{C} + \overline{D}$로 바뀐다는 점을 놓치면 ②처럼 $\overline{AB}$를 고르게 된다.

NoteBase

다시 나오면 이렇게 판단

최종 게이트부터 거꾸로 읽어 Y를 먼저 식으로 적고, 물은 식이 $\overline{Y}$이면 전체에 한 번만 드모르간을 적용한 뒤 각 괄호를 SOP로 전개해 포함 항을 확인하면 된다.

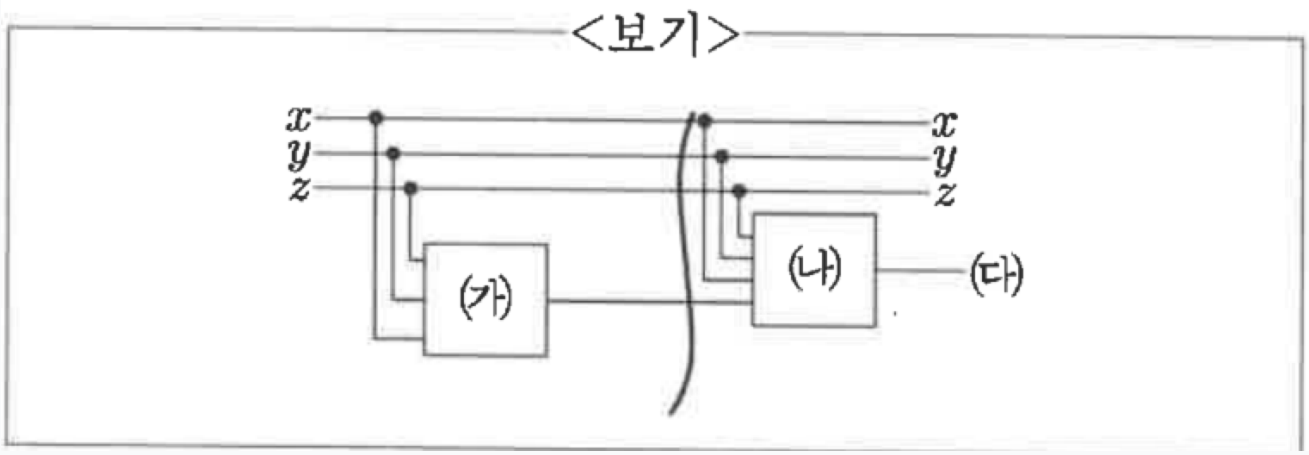
암기 포인트

OR의 보수는 AND의 합으로, AND의 보수는 OR의 합으로 바뀐다. $(X+Y)' = \overline{X}\,\overline{Y}$, $(XYZ)' = \overline{X} + \overline{Y} + \overline{Z}$가 핵심이다.

NoteBase

문 16. <보기>의 굽은 곡선을 중심으로 좌측이 송신부, 우측이 수신부라 할 때, 송신부에서 총 3비트의 데이터 신호 x, y, z 전송 시 발생하는 1비트의 오류를 수신부가 인지할 수 있도록 하는 (가), (나), (다)를 옳게 짝지은 것은? (단, (가), (나)는 들어갈 논리게이트 종류와 입력 신호의 개수를 의미하고, (다)는 전송 오류가 발생했음을 알리는 값이다.)

<보기>



① (가): XOR3, (나): XOR4, (다): 0

② (가): XNOR3, (나): XOR4, (다): 0

③ (가): XOR3, (나): XNOR4, (다): 1

④ (가): XNOR3, (나): XNOR4, (다): 0

정답 2번

출제 포인트

3비트 데이터 x, y, z에 패리티 비트를 추가해 1비트 오류를 검출하는 회로에서, 송신부 게이트 (가)의 종류와 수신부 게이트 (나)의 종류, 그리고 오류 발생 시 (나)의 출력값 (다)를 계산해 올바르게 짝지을 수 있는지를 묻는다.

조건 정리

송신부: x, y, z (3비트 데이터) → (가) → 패리티 비트 p 생성. 전송 경로(굵은 곡선)를 거쳐 수신부에 x, y, z, p 4비트가 도달한다. 수신부: x, y, z, p → (나) → (다). (다)는 1비트 오류가 발생했을 때 출력되는 값이다.

사용할 공식

XOR: 입력 중 1의 개수가 홀수이면 1, 짝수이면 0. XNOR: 입력 중 1의 개수가 짝수이면 1, 홀수이면 0(XOR의 부정). 1비트 오류 발생 시 전체 비트의 XOR 합이 반전된다.

NoteBase

계산 과정

$D = x \oplus y \oplus z$로 정의한다.

②번 (가)=XNOR3, (나)=XOR4인 경우:
패리티 비트 $p = \overline{D}$
정상 수신 시: XOR4$(x, y, z, p) = D \oplus \overline{D} = 1$
1비트 오류 시(데이터 한 비트가 반전되어 $D \to D', D' \neq D$):
- $D = 0$이면 $D' = 1, p = \overline{0} = 1$: $1 \oplus 1 = 0$
- $D = 1$이면 $D' = 0, p = \overline{1} = 0$: $0 \oplus 0 = 0$
→ 오류 발생 시 출력은 항상 0이므로 (다) = 0이다.

선지 판단

① XOR3, XOR4, 0: $p = D$, 정상 수신 시 XOR4 = $D \oplus D = 0$, 1비트 오류 시 = 1. 오류 알림값은 1이어야 하므로 (다)=0은 틀린 것이다. ☒
② XNOR3, XOR4, 0: $p = \overline{D}$, 정상 수신 시 XOR4 = $D \oplus \overline{D} = 1$, 1비트 오류 시 = 0. 오류 알림값이 0이므로 (다)=0은 맞는 것이다. ✓
③ XOR3, XNOR4, 1: $p = D$, 정상 수신 시 XNOR4 = $\overline{D \oplus D} = 1$, 1비트 오류 시 = 0. 오류 알림값은 0이어야 하므로 (다)=1은 틀린 것이다. ☒
④ XNOR3, XNOR4, 0: $p = \overline{D}$, 정상 수신 시 XNOR4 = $\overline{D \oplus \overline{D}} = 0$, 1비트 오류 시 = 1. 오류 알림값은 1이어야 하므로 (다)=0은 틀린 것이다. ☒

오답 포인트

①과 ②는 수신부가 모두 XOR4이지만 송신부 게이트가 다르다. XOR3(짝수 패리티)을 쓰면 정상 수신 시 출력이 0이고 오류 시 1이 되어 (다)=0이 아니라 (다)=1이 맞는 값이 된다. ③은 계산 방향은 맞지만 오류 알림값을 0이 아닌 1로 뒤집어 놓은 함정이다. ④는 XNOR3+XNOR4 조합에서 정상 수신 시 출력이 0이 되어 오류 알림값이 오히려 1이 된다.

다시 나오면 이렇게 판단

(가) 게이트로 패리티 비트 p를 먼저 구한다. 그다음 (나) 게이트에 4비트 모두를 넣었을 때 오류가 없는 정상 수신의 출력값을 계산한다. 그 값의 반대가 1비트 오류 발생 시의 알림값 (다)가 된다.

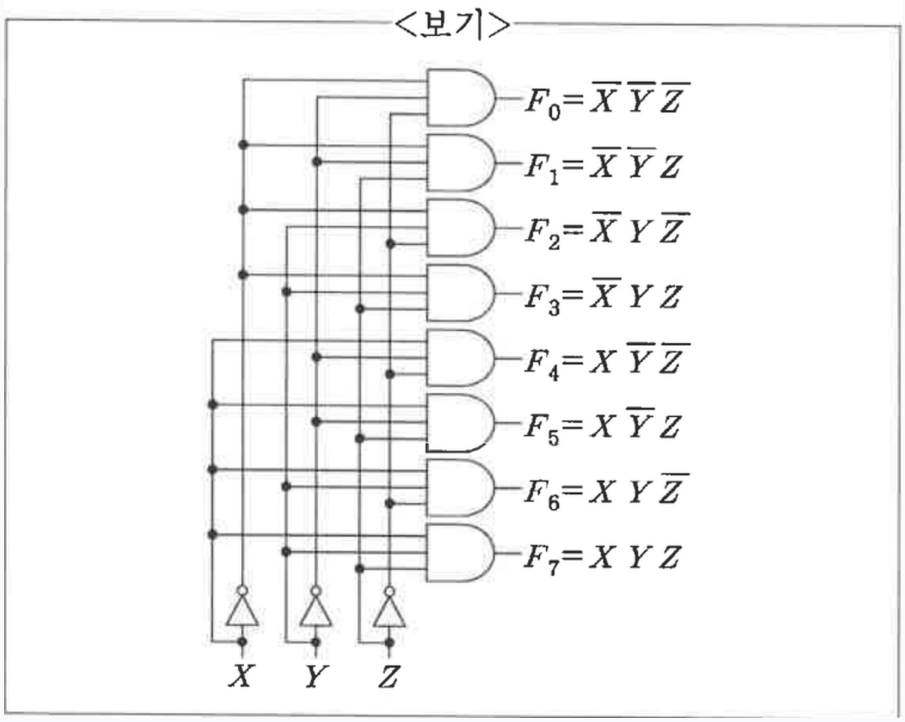
암기 포인트

XNOR3+XOR4 조합(홀수 패리티): 정상 수신 시 XOR4 출력=1, 오류 시=0. XOR3+XOR4 조합(짝수 패리티): 정상 수신 시 XOR4 출력=0, 오류 시=1. 송신부 게이트가 XOR이면 오류 알림값은 1, XNOR이면 오류 알림값은 0(수신부가 XOR일 때 기준)이다.

NoteBase

문 17. <보기>의 3-to-8 디코더와 3개의 OR 게이트를 이용하여 $F(X, Y, Z) = \overline{X}Y + \overline{X}Z + YZ$를 나타내고자 한다. OR 게이트의 입력으로 들어가야 하는 디코더의 출력을 모두 고른 것은?

<보기>



① F_1, F_2, F_4, F_5

② F_1, F_2, F_3, F_7

③ F_1, F_5, F_6, F_7

④ F_2, F_3, F_6, F_7

정답 2번

출제 포인트

3-to-8 디코더의 각 출력은 3변수 최소항(minterm) 하나에 정확히 대응한다. 주어진 불 함수를 최소항의 합(Sum of Minterms) 형태로 전개하면, 그 최소항 번호에 해당하는 디코더 출력을 OR 게이트로 묶어 함수를 구현할 수 있다.

조건 정리

구현할 함수: $F(X, Y, Z) = \bar{X}Y + \bar{X}Z + YZ$
디코더 출력 대응:
$F_0 = \bar{X}\bar{Y}\bar{Z}(m_0), F_1 = \bar{X}\bar{Y}Z(m_1), F_2 = \bar{X}Y\bar{Z}(m_2), F_3 = \bar{X}YZ(m_3)$
$F_4 = X\bar{Y}\bar{Z}(m_4), F_5 = X\bar{Y}Z(m_5), F_6 = XY\bar{Z}(m_6), F_7 = XYZ(m_7)$

NoteBase

사용할 공식

불 함수의 각 곱항(product term)에 빠진 변수를 $(A + \bar{A}) = 1$로 보완해 최소항으로 전개한다. 최소항 번호는 변수를 (X, Y, Z) 순서로 2진수로 읽은 값이다.

계산 과정

각 항을 최소항으로 전개한다.

$\bar{X}Y = \bar{X}Y(Z + \bar{Z}) = \bar{X}YZ + \bar{X}Y\bar{Z} = m_3 + m_2$
$\bar{X}Z = \bar{X}Z(Y + \bar{Y}) = \bar{X}YZ + \bar{X}\bar{Y}Z = m_3 + m_1$
$YZ = YZ(X + \bar{X}) = XYZ + \bar{X}YZ = m_7 + m_3$

전체 합집합(중복 제거): $F = \sum m(1, 2, 3, 7)$

진리표 교차 검증:
(0,0,1)=m_1: $\bar{X}Z = 1 \rightarrow$ F=1 ✓
(0,1,0)=m_2: $\bar{X}Y = 1 \rightarrow$ F=1 ✓
(0,1,1)=m_3: 세 항 모두 1 → F=1 ✓
(1,0,1)=m_5: $\bar{X} = 0, Y = 0 \rightarrow$ F=0 ✓ (포함 안 됨)
(1,1,0)=m_6: $\bar{X} = 0, Z = 0 \rightarrow$ F=0 ✓ (포함 안 됨)
(1,1,1)=m_7: $YZ = 1 \rightarrow$ F=1 ✓

OR 게이트 입력은 F_1, F_2, F_3, F_7이다.

선지 판단

① F_1, F_2, F_4, F_5: $m_4 = (1, 0, 0)$과 $m_5 = (1, 0, 1)$은 F에 포함되지 않고 m_3이 빠져 있다. 틀린 선지이다.
② F_1, F_2, F_3, F_7: m_1, m_2, m_3, m_7 모두 전개 결과와 일치한다. 맞는 선지이다.
③ F_1, F_5, F_6, F_7: $m_5 = (1, 0, 1)$과 $m_6 = (1, 1, 0)$은 F에 포함되지 않고 m_2, m_3이 빠져 있다. 틀린 선지이다.
④ F_2, F_3, F_6, F_7: $m_6 = (1, 1, 0)$은 F에 포함되지 않고 m_1이 빠져 있다. 틀린 선지이다.

오답 포인트

①번은 $\bar{X}Y$를 'X=1, Y=0'으로 혼동해 m_4, m_5를 선택할 때 발생하는 오류이다. 변수 보수(bar)의 위치를 정확히 확인하는 것이 핵심이다.
③·④번은 YZ 항에서 'X가 1인 경우(m_7)만 선택하면 된다'고 오해해 m_5, m_6 등을 포함시키거나 m_1을 빠뜨리는 오류이다.

NoteBase

다시 나오면 이렇게 판단

디코더+OR 게이트 구현 문제는 반드시 함수를 최소항 합(SOM)으로 전개하는 것이 첫 단계이다. 각 곱항에 빠진 변수를 $(A + \bar{A})$로 보완하고, 최소항 번호(XYZ를 2진수로 읽은 값)를 구한 뒤 해당 번호의 디코더 출력을 선택한다. 전개 후 반드시 2~3행 진리표로 교차 검증한다.

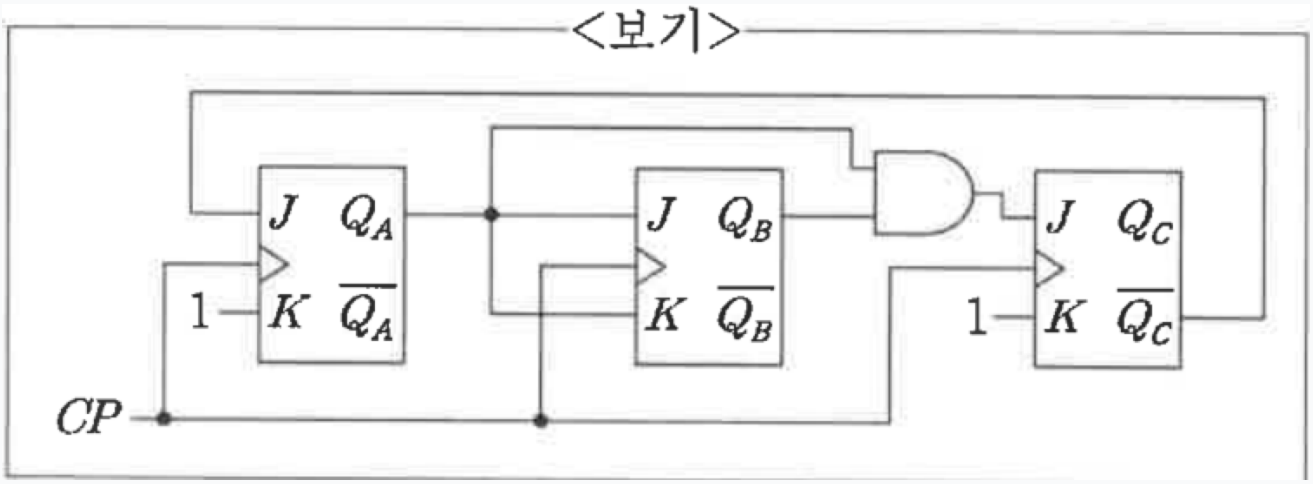
암기 포인트

3-to-8 디코더 출력 F_i는 10진수 i에 해당하는 최소항 m_i이다. 예: $F_3 = \bar{X}YZ(011_2 = 3)$. 불 함수를 SOM으로 나타내면 $F = \sum m(1, 2, 3, 7)$이고 이것이 OR 게이트로 묶을 디코더 출력 목록이 된다.

NoteBase

문 18. <보기>와 같은 동기 순서논리회로가 있다. 이 회로의 $Q_CQ_BQ_A$의 값을 시간 순으로 나타낸 것으로 가장 옳은 것은? (단, $Q_CQ_BQ_A$의 초기상태는 000이며, CP는 클록 펄스를 의미하고, Q_C는 MSB, Q_A는 LSB이다.)

<보기>



① 000 → 001 → 010 → 011 → 100 → 000 → ...

② 000 → 100 → 011 → 010 → 001 → 000 → ...

③ 000 → 111 → 110 → 101 → 100 → 000 → ...

④ 000 → 001 → 011 → 101 → 111 → 000 → ...

정답 1번

출제 포인트

JK 플립플롭 3개로 구성된 동기 순서논리회로의 회로도를 분석하여 클록마다의 상태 전이 순서를 구한다. 각 플립플롭의 J·K 입력 연결을 회로도에서 읽은 뒤, 초기 상태에서 출발하여 JK 특성표를 한 클록씩 적용하여 다음 상태를 계산한다.

조건 정리

초기 상태: $Q_CQ_BQ_A = 000$

FF_A: $J_A = \bar{Q}_C$(Q_C 보수 피드백), $K_A = 1$

FF_B: $J_B = Q_A$, $K_B = Q_A$

FF_C: $J_C = Q_A \cdot Q_B$(AND 게이트 출력), $K_C = 1$

모든 플립플롭이 공통 클록 CP에서 동시에 전이하는 동기 회로이다.

사용할 공식

JK 플립플롭 특성표: $J = 0, K = 0 \rightarrow$ Q 유지 / $J = 0, K = 1 \rightarrow Q = 0$(리셋) / $J = 1, K = 0 \rightarrow Q = 1$(세트) / $J = 1, K = 1 \rightarrow$ Q 반전(토글)

NoteBase

계산 과정

상태 000 ($Q_C=0, Q_B=0, Q_A=0$):
$J_A=\bar{Q}_C=1,\ K_A=1 \rightarrow Q_A$ 토글: 0→1
$J_B=Q_A=0,\ K_B=0 \rightarrow Q_B$ 유지: 0→0
$J_C=0\cdot 0=0,\ K_C=1 \rightarrow Q_C$ 리셋: 0→0
→ 다음 상태: 001

상태 001 ($Q_C=0, Q_B=0, Q_A=1$):
$J_A=\bar{Q}_C=1,\ K_A=1 \rightarrow Q_A$ 토글: 1→0
$J_B=Q_A=1,\ K_B=1 \rightarrow Q_B$ 토글: 0→1
$J_C=1\cdot 0=0,\ K_C=1 \rightarrow Q_C$ 리셋: 0→0
→ 다음 상태: 010

상태 010 ($Q_C=0, Q_B=1, Q_A=0$):
$J_A=\bar{Q}_C=1,\ K_A=1 \rightarrow Q_A$ 토글: 0→1
$J_B=Q_A=0,\ K_B=0 \rightarrow Q_B$ 유지: 1→1
$J_C=0\cdot 1=0,\ K_C=1 \rightarrow Q_C$ 리셋: 0→0
→ 다음 상태: 011

상태 011 ($Q_C=0, Q_B=1, Q_A=1$):
$J_A=\bar{Q}_C=1,\ K_A=1 \rightarrow Q_A$ 토글: 1→0
$J_B=Q_A=1,\ K_B=1 \rightarrow Q_B$ 토글: 1→0
$J_C=1\cdot 1=1,\ K_C=1 \rightarrow Q_C$ 토글: 0→1
→ 다음 상태: 100

상태 100 ($Q_C=1, Q_B=0, Q_A=0$):
$J_A=\bar{Q}_C=0,\ K_A=1 \rightarrow Q_A$ 리셋: 0→0
$J_B=Q_A=0,\ K_B=0 \rightarrow Q_B$ 유지: 0→0
$J_C=0\cdot 0=0,\ K_C=1 \rightarrow Q_C$ 리셋: 1→0
→ 다음 상태: 000

최종 순서: 000→001→010→011→100→000→⋯

NoteBase

선지 판단

① 000→001→010→011→100→000→⋯ : 계산 결과와 일치한다. 정답.
② 000→100→011→010→001→000→⋯ : 오답. 초기 상태 000에서 $J_C = Q_A \cdot Q_B = 0, K_C = 1$이므로 Q_C는 0으로 리셋되어, 첫 전이 후 $Q_C = 1$(100)이 될 수 없다.
③ 000→111→110→101→100→000→⋯ : 오답. 초기 상태 000에서 $Q_A = 0$이므로 $J_B = 0, K_B = 0$이 되어 Q_B는 변하지 않는다. 000→111로의 전이는 성립하지 않는다.
④ 000→001→011→101→111→000→⋯ : 오답. 상태 011에서 계산하면 다음 상태는 100이지만, ④는 101을 다음 상태로 제시하여 틀렸다.

오답 포인트

상태 011에서 세 비트가 동시에 변하는 점을 놓치기 쉽다. $Q_A = 1, Q_B = 1$이므로 AND 게이트 출력이 1이 되어 $J_C = K_C = 1$로 Q_C가 토글(0→1)되고, 동시에 $J_A = K_A = 1$로 Q_A가 토글(1→0), $J_B = K_B = 1$로 Q_B가 토글(1→0)된다. 따라서 011 다음 상태는 100이며, 101·110·111이 아니다.

다시 나오면 이렇게 판단

회로도에서 각 플립플롭의 J·K 입력 연결을 먼저 정리한다. 피드백 배선($\bar{Q}_C \rightarrow J_A$)이 있으면 어느 상태에서 리셋 조건이 발생하는지 파악하면 카운터의 modulus를 빠르게 확인할 수 있다. AND 게이트가 있으면 해당 조건($Q_A = Q_B = 1$)이 충족되는 시점에 Q_C가 변한다는 점에 주목한다.

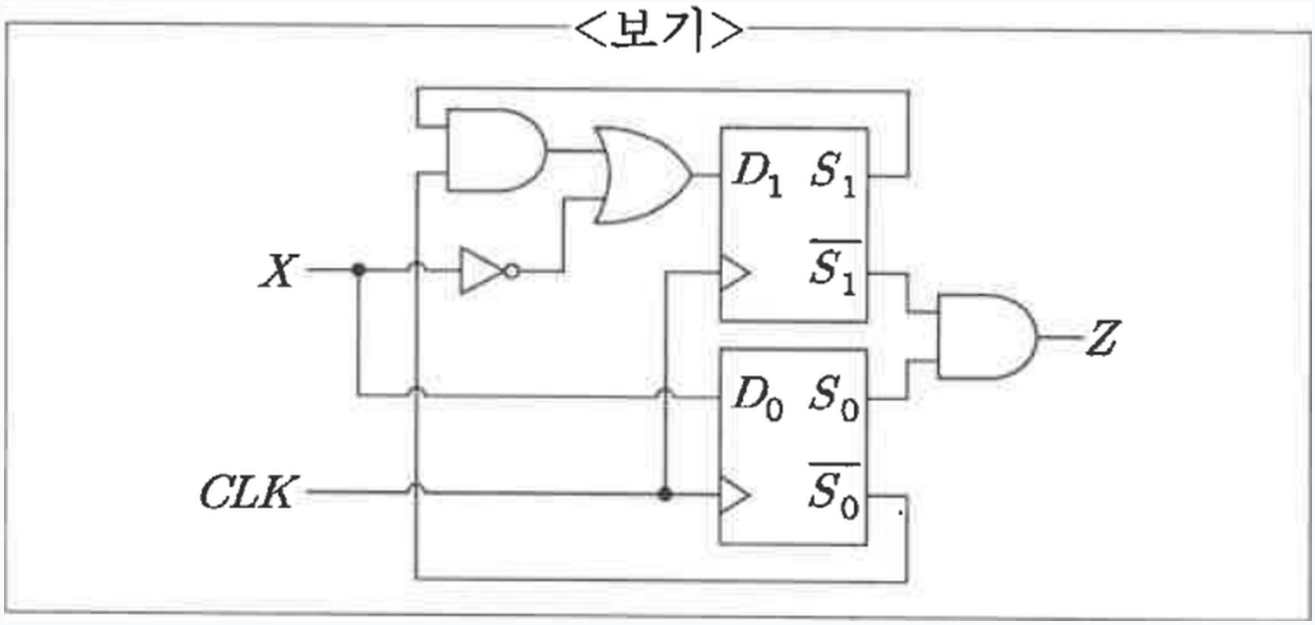
암기 포인트

JK 플립플롭 특성: $J = K = 1$→토글, $J = 0, K = 1$→리셋(0), $J = 1, K = 0$→세트(1), $J = K = 0$→유지.
이 회로는 mod-5 동기 이진 업 카운터이다. 0(000)→1(001)→2(010)→3(011)→4(100)→0(000)의 5개 상태를 순환한다.

NoteBase

문 19. <보기>의 D 플립플롭 기반 유한상태머신(FSM)에 대한 설명으로 가장 옳지 않은 것은?

<보기>



① 무어머신이다.

② 현재 상태가 $S_1S_0 = 00$이면, 다음 상태는 $S_1^+S_0^+ = 11$이 될 수 없다.

③ 현재 상태가 $S_1S_0 = 11$이고 입력이 $X = 0$이면, 다음 상태는 $S_1^+S_0^+ = 00$이다.

④ 현재 상태가 $S_1S_0 = 01$이고 입력이 $X = 1$이면, 출력 $Z = 1$이다.

정답 3번

출제 포인트

이 문제는 회로도에서 D 입력식과 출력식을 읽어 무어머신 여부, 다음 상태, 출력을 판정하는 문제이다.

선지별 해설

1번 O: 출력 Z가 현재 상태변수 S_1, S_0만으로 결정되고 입력 X가 출력으로 직접 들어가지 않으므로 무어머신이다.

2번 O: 현재 상태가 00이면 $S_1^+ = \overline{X} + 0 = \overline{X}, S_0^+ = X$이다. 따라서 $X = 0$이면 10, $X = 1$이면 01이 되어 11은 될 수 없다.

3번 X: 현재 상태가 11이고 $X = 0$이면 $S_1^+ = \overline{0} + 1 = 1, S_0^+ = 0$이므로 다음 상태는 10이다. 틀린 표현은 '다음 상태는 00이다'이고, 올바르게는 '다음 상태는 10이다'이다.

4번 O: 현재 상태가 01이면 $Z = \overline{S_1}S_0 = 1 \cdot 1 = 1$이다. 출력은 상태만으로 결정되므로 $X = 1$일 때도 $Z = 1$이 맞다.

NoteBase

오답 포인트

3번은 $X=0$이면 두 플립플롭이 함께 0으로 초기화된다고 착각하게 만드는 선지이다. 실제로는 D_0만 X를 그대로 받고, D_1에는 $\overline{X}$가 OR로 들어가므로 $X=0$일 때 S_1^+는 1이 된다.

다시 나오면 이렇게 판단

회로형 FSM은 먼저 각 플립플롭의 D 입력을 다음 상태식으로 적고, 그다음 출력식을 따로 적어 현재 상태와 입력을 대입해 판단한다. 출력선에 입력이 직접 연결되지 않으면 무어머신으로 본다.

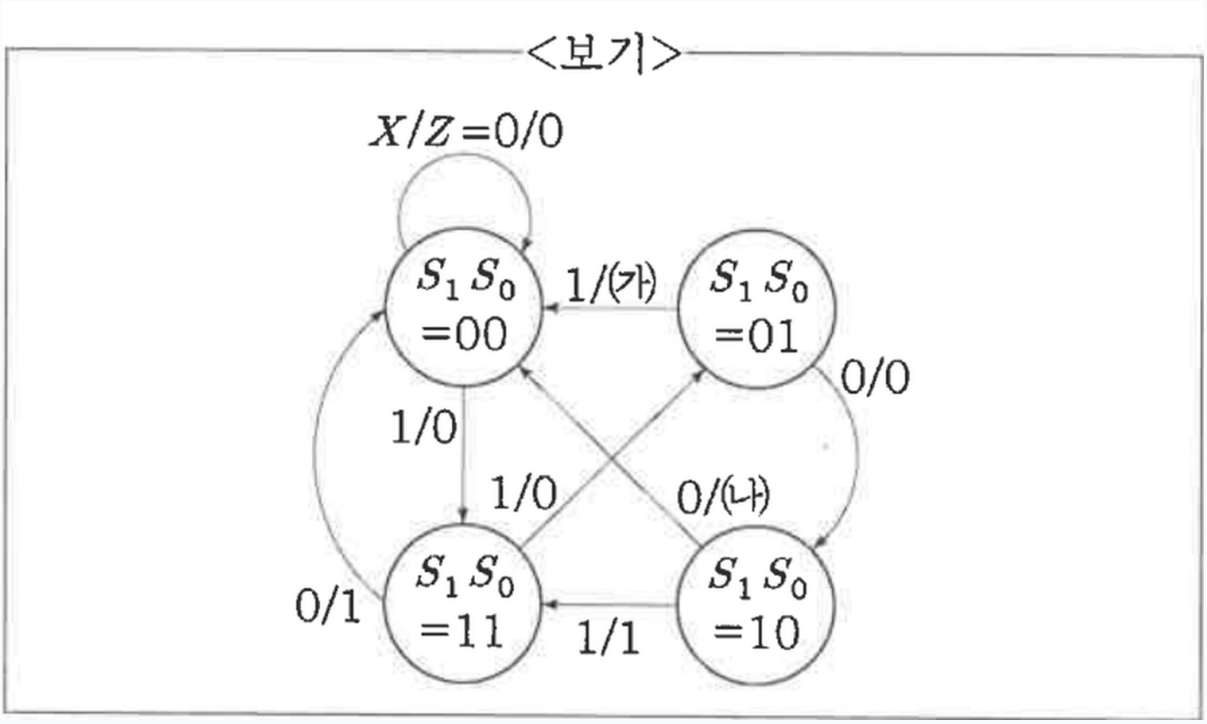
암기 포인트

D 플립플롭에서는 D 입력이 곧 다음 상태이다. 이 회로는 $S_1^+ = \overline{X} + S_1, S_0^+ = X, Z = \overline{S_1} S_0$이다.

NoteBase

문 20. <보기>는 입력 X와 출력 Z를 갖는 유한상태머신(FSM)의 상태도이다. $Z = S_1\overline{X} + S_1\overline{S_0} + \overline{S_1}S_0X$일 때, (가), (나)를 옳게 짝지은 것은?

<보기>



① (가): 0, (나): 0

② (가): 0, (나): 1

③ (가): 1, (나): 0

④ (가): 1, (나): 1

정답 4번

출제 포인트

FSM의 출력 논리식 $Z = S_1\overline{X} + S_1\overline{S_0} + \overline{S_1}S_0X$에 특정 상태값과 입력값을 대입하여 상태 전이도의 빈칸 출력 (가), (나)를 계산하는 문항이다. (가)는 상태 S_1S_0=01에서 X=1일 때의 출력, (나)는 상태 S_1S_0=10에서 X=0일 때의 출력이다.

조건 정리

(가): 상태 S_1S_0=01 (S_1=0, S_0=1), 입력 X=1
(나): 상태 S_1S_0=10 (S_1=1, S_0=0), 입력 X=0

NoteBase

사용할 공식

$Z = S_1\overline{X} + S_1\overline{S_0} + \overline{S_1}S_0X$
세 항은 OR 관계이므로 항 하나라도 1이면 Z=1이다.

계산 과정

(가) — S_1=0, S_0=1, X=1 대입:
$Z = 0\cdot 0 + 0\cdot 0 + 1\cdot 1\cdot 1 = 0 + 0 + 1 = 1$
따라서 (가) = 1

(나) — S_1=1, S_0=0, X=0 대입:
$Z = 1\cdot 1 + 1\cdot 1 + 0\cdot 0\cdot 0 = 1 + 1 + 0 = 1$
따라서 (나) = 1

선지 판단

① (가)=0, (나)=0: 둘 다 틀렸다.
② (가)=0, (나)=1: (나)=1은 맞으나 (가)가 틀렸다. 세 번째 항 $\overline{S_1}S_0X = 1\cdot 1\cdot 1 = 1$이 살아남아 Z=1이므로 (가)=0이 될 수 없다.
③ (가)=1, (나)=0: (가)=1은 맞으나 (나)가 틀렸다. 첫 번째 항 $S_1\overline{X} = 1\cdot 1 = 1$과 두 번째 항 $S_1\overline{S_0} = 1\cdot 1 = 1$이 모두 1이므로 Z=1이 되어 (나)=0이 될 수 없다.
④ (가)=1, (나)=1: 계산 결과와 일치하여 옳다.

오답 포인트

①②③은 (가) 또는 (나)를 0으로 잘못 판단한 경우이다. 출력 논리식은 세 항의 합(OR)이므로 항 하나라도 1이 되면 전체 Z=1이다. (가) 계산에서는 S_1=0이어서 처음 두 항이 모두 0이 되지만, 세 번째 항 $\overline{S_1}S_0X$만으로 Z=1이 된다. (나) 계산에서는 S_1=1, S_0=0이어서 첫 번째·두 번째 항이 동시에 1이 된다.

다시 나오면 이렇게 판단

상태도에서 빈칸 레이블이 있는 전이 화살표를 찾아 출발 상태의 S_1S_0값과 그 화살표의 X값을 읽은 뒤, 출력 논리식에 직접 대입하여 Z를 계산한다. 상태도의 X/Z 표기에서 슬래시 앞이 입력(X), 뒤가 출력(Z)이다.

암기 포인트

$Z = S_1\overline{X} + S_1\overline{S_0} + \overline{S_1}S_0X$의 세 항 조건: ①$S_1$=1이고 X=0이면 Z=1 ②$S_1$=1이고 S_0=0이면 Z=1 ③S_1=0이고 S_0=1이고 X=1이면 Z=1. 세 조건 중 하나라도 해당하면 Z=1이다.

NoteBase