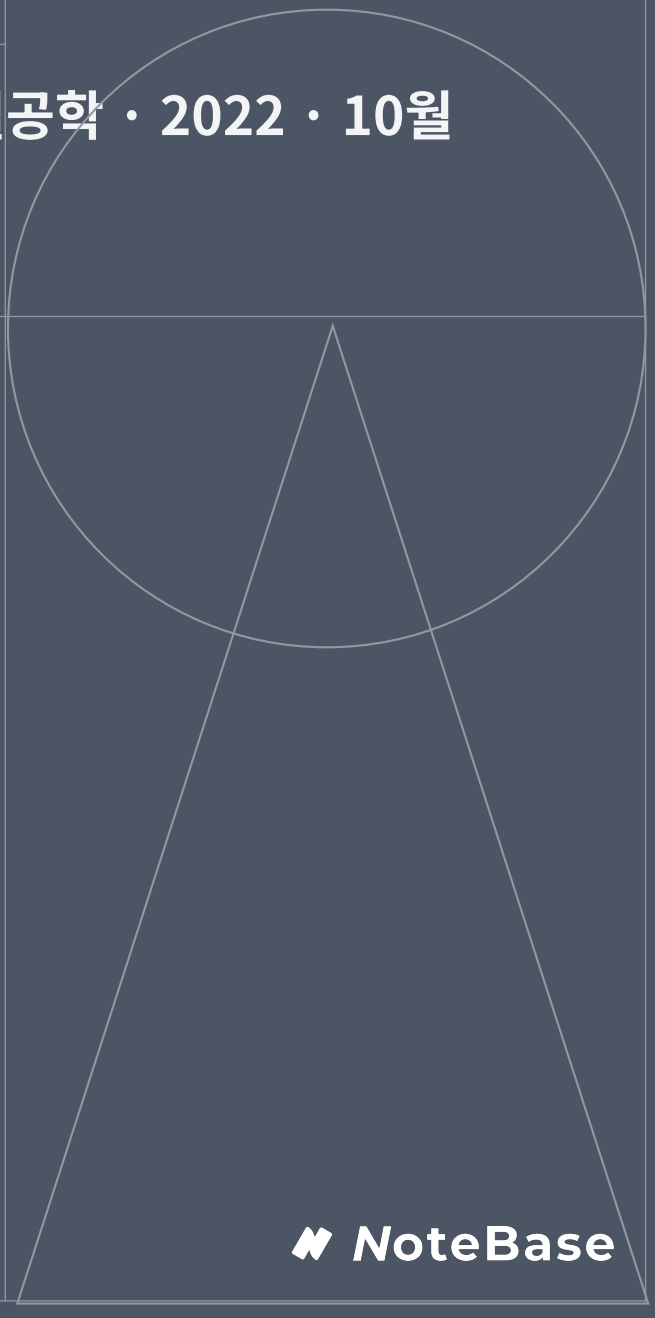
기출OX 해설집

공무원/서울시/7급 · 디지털공학 · 2022 · 10월



NoteBase

기출OX 해설집

공무원/서울시/7급 · 디지털공학 · 2022 · 10월

공무원/서울시/7급 · 디지털공학 · 2022 · 10월

문 1. 디지털 시스템과 디지털 신호에 대한 설명으로 가장 옳지 않은 것은?

① 일반적으로 아날로그 신호를 디지털 신호로 변환할 때에는 약간의 정보 손실과 일그러짐을 수반한다.

② 디지털 시스템에서 사용하는 펄스의 상승하는 부분을 리딩 에지(leading edge 또는 positive edge)라고 한다.

③ 디지털 신호는 아날로그 신호에 비하여 잡음과 일그러짐에 훨씬 강인한 장점이 있다.

④ 높은 전압 레벨을 '0'으로 표시하고 낮은 전압 레벨을 '1'로 표시하는 논리를 정논리(positive logic)라 한다.

정답 4번

출제 포인트

디지털 신호의 특성과 에지 명칭, 정논리와 부논리의 대응 관계를 정확히 구별하는 문제이다.

핵심 개념

아날로그 신호는 연속적인 값으로 정보를 표현하고, 디지털 신호는 이산적인 레벨로 정보를 표현한다.
아날로그 신호를 디지털 신호로 변환할 때는 샘플링과 양자화가 일어나므로 양자화 오차에 따른 약간의 정보 손실이 생길 수 있다.
리딩 에지는 펄스가 낮은 레벨에서 높은 레벨로 올라가는 상승 구간이다.
정논리는 높은 전압 레벨을 1, 낮은 전압 레벨을 0에 대응시키는 방식이고, 부논리는 높은 전압 레벨을 0, 낮은 전압 레벨을 1에 대응시키는 방식이다.

선지별 해설

① O: 아날로그 신호를 디지털 신호로 바꿀 때는 샘플링과 양자화 과정이 필요하므로 양자화 오차가 생기고, 원래의 연속 정보가 일부 손실될 수 있다.
② O: 디지털 펄스의 상승하는 부분은 리딩 에지이며 positive edge라고도 한다.
③ O: 디지털 신호는 임계값으로 0과 1을 판정하고 재생이 가능하므로 아날로그 신호보다 잡음과 일그러짐에 강하다.
④ X: 틀린 표현은 "높은 전압 레벨을 '0'으로 표시하고 낮은 전압 레벨을 '1'로 표시하는 논리를 정논리라 한다"이다. 실제로는 높은 전압 레벨을 1, 낮은 전압 레벨을 0으로 표시해야 정논리이고, 문제의 설명은 부논리(negative logic)에 해당한다.

NoteBase

오답 포인트

④는 전압의 높고 낮음을 뒤집어 놓고 용어만 정논리라고 붙인 함정이다. 수험생이 정논리와 부논리의 정의를 전압 크기 자체로만 막연히 기억하면 쉽게 속게 된다.

다시 나오면 이렇게 판단

발문이 '옳지 않은 것'인지 먼저 확인한 뒤, 용어형 선지는 정의를 바로 대응시켜 판정한다. 정논리는 high=1, low=0이고 부논리는 high=0, low=1이라는 기준을 대입하면 바로 판단된다.

암기 포인트

정논리=high 1, low 0 / 부논리=high 0, low 1 / 상승 에지=leading edge

NoteBase

문 2. 2의 보수로 표현된 6비트 2진수 2개를 더한 6비트 결과를 저장한다고 할 때, 오버플로우(overflow)가 발생하는 경우로 가장 옳은 것은? (단, 최상위 비트는 부호 비트이다.)

① $(111011)_2 + (101100)_2$

② $(001110)_2 + (110000)_2$

③ $(111001)_2 + (110011)_2$

④ $(010111)_2 + (011011)_2$

정답 4번

출제 포인트

6비트 2의 보수 덧셈에서 오버플로우가 발생하는 조건은 같은 부호의 두 수를 더했는데 결과의 부호가 바뀌는지, 또는 표현 범위를 벗어나는지 판정하는 것이다.

조건 정리

주어진 값: 각 피연산자는 6비트 2의 보수이며 최상위 비트는 부호 비트이다.
구해야 할 값: 각 선지의 덧셈에서 오버플로우 발생 여부이다.
주의할 조건: 6비트 2의 보수의 표현 범위는 -32부터 31까지이다.

사용할 공식

오버플로우 판정은 '두 피연산자의 부호 비트가 같고 결과의 부호 비트가 다르면 오버플로우'이다.
같은 뜻으로 합이 6비트 2의 보수 범위 -32부터 31까지를 벗어나면 오버플로우이다.

이 공식을 쓰는 이유:
2의 보수 덧셈은 마지막 자리올림 자체가 아니라 부호 반전과 표현 범위 초과 여부로 오버플로우를 판단하기 때문이다.

계산 과정

1. 6비트 2의 보수의 표현 범위가 -32부터 31까지임을 확인한다.
2. 각 선지를 10진수로 바꾸면 ① 111011_2+101100_2=-5+(-20)=-25, ② 001110_2+110000_2=14+(-16)=-2, ③ 111001_2+110011_2=-7+(-13)=-20, ④ 010111_2+011011_2=23+27=50이다.
3. ①, ②, ③은 결과가 범위 안에 있어 오버플로우가 없고, ④는 합이 50으로 31을 넘어서 오버플로우가 발생한다.

NoteBase

선지별 해설

① X: 111011_2=-5, 101100_2=-20이므로 합은 -25이다. -25는 6비트 2의 보수 범위 안이어서 오버플로우가 발생하지 않는다. 틀린 표현은 '①에서 오버플로우가 발생한다'이며, 올바르게는 '①은 오버플로우가 발생하지 않는다'이다.
② X: 001110_2=14, 110000_2=-16이므로 합은 -2이다. 서로 부호가 다른 두 수의 덧셈은 오버플로우가 발생하지 않는다. 틀린 표현은 '②에서 오버플로우가 발생한다'이며, 올바르게는 '②는 오버플로우가 발생하지 않는다'이다.
③ X: 111001_2=-7, 110011_2=-13이므로 합은 -20이다. -20은 범위 안이고 음수+음수의 결과도 음수이므로 오버플로우가 없다. 틀린 표현은 '③에서 오버플로우가 발생한다'이며, 올바르게는 '③은 오버플로우가 발생하지 않는다'이다.
④ O: 010111_2=23, 011011_2=27이므로 합은 50이다. 50은 6비트 2의 보수 최대값 31을 넘는다. 저장 결과는 110010_2가 되어 부호가 음수로 바뀌므로 오버플로우가 발생한다.

오답 포인트

최종 캐리가 생기면 무조건 오버플로우라고 보는 실수가 잦다. 2의 보수에서는 마지막 캐리보다 같은 부호끼리 더했는데 결과 부호가 반대로 바뀌는지가 핵심이며, 표현 범위 -32부터 31까지를 함께 확인해야 한다.

다시 나오면 이렇게 판단

먼저 각 수의 부호와 10진값을 확인한 뒤 같은 부호의 덧셈인지 본다. 같은 부호 덧셈에서 결과가 표현 범위를 벗어나거나 결과 부호가 반대로 바뀌면 오버플로우로 판단한다.

암기 포인트

2의 보수 오버플로우는 '같은 부호 덧셈, 결과 부호 반전'으로 판정한다.

NoteBase

문 3. 10진수 -20을 부호가 있는 2의 보수로 변환하여 16비트 시프트 레지스터(shift register)에 저장하고 왼쪽으로 4비트만큼 시프트하였을 때, 레지스터에 저장된 값을 10진수로 변환한 값은?

① -320

② -80

③ -5

④ 80

정답 1번

출제 포인트

16비트 2의 보수로 표현된 음수를 왼쪽으로 4비트 시프트했을 때의 10진수 값을 묻는 문제이다.

조건 정리

주어진 값: 10진수 -20을 16비트 2의 보수로 저장한다.
구해야 할 값: 왼쪽으로 4비트 시프트한 뒤 레지스터 값의 10진수 표현이다.
주의할 조건: 16비트 범위에서 왼쪽으로 4비트 시프트하면 오른쪽에 0이 채워지고, 결과가 표현 범위를 넘지 않으면 2^4배와 같다.

사용할 공식

$x << n = x \times 2^n$

이 공식을 쓰는 이유: 2의 보수 정수를 왼쪽으로 n비트 시프트하면 값이 2^n배가 된다. 이 문제의 결과는 -20×2^4이며 16비트 범위 안에 있으므로 그대로 적용할 수 있다.

계산 과정

1. 10진수 20의 16비트 표현은 0000 0000 0001 0100이다.
2. 이를 2의 보수로 바꾸면 1111 1111 1110 1100이므로 -20이다.
3. 이를 왼쪽으로 4비트 시프트하면 1111 1110 1100 0000이 된다.
4. 이 값을 2의 보수로 다시 읽으면 -320이므로 정답은 ①이다.

NoteBase

선지 판단

① O
- 계산 결과가 -320이므로 맞다.

② X
- '-80'이 틀린 표현이다. 올바르게는 '-320'이다.
- -80은 4비트 시프트를 2^2배처럼 잘못 처리하거나 시프트 비트 수를 혼동할 때 나오는 값이다.

③ X
- '-5'가 틀린 표현이다. 올바르게는 '-320'이다.
- -5는 왼쪽 시프트를 오른쪽 시프트처럼 해석해 값을 줄여 읽을 때 생기는 오답이다.

④ X
- '80'이 틀린 표현이다. 올바르게는 '-320'이다.
- 80은 시프트 후에도 최상위 비트가 1이라는 점을 놓치고 부호를 양수로 잘못 읽을 때 나오는 값이다.

오답 포인트

왼쪽 시프트를 단순한 자리 이동으로만 보고 부호 해석을 빼먹으면 ④처럼 양수로 읽기 쉽다. 또 4비트 시프트를 2^4배가 아니라 다른 배수로 착각하면 ②, 왼쪽과 오른쪽 시프트를 혼동하면 ③이 나온다.

다시 나오면 이렇게 판단

음수 시프트 문제는 먼저 2의 보수 비트열을 만든 뒤 시프트 후 비트열을 다시 읽는다. 결과가 표현 범위 안에 있으면 왼쪽 n비트 시프트는 2^n배라는 규칙으로 빠르게 검산할 수 있다.

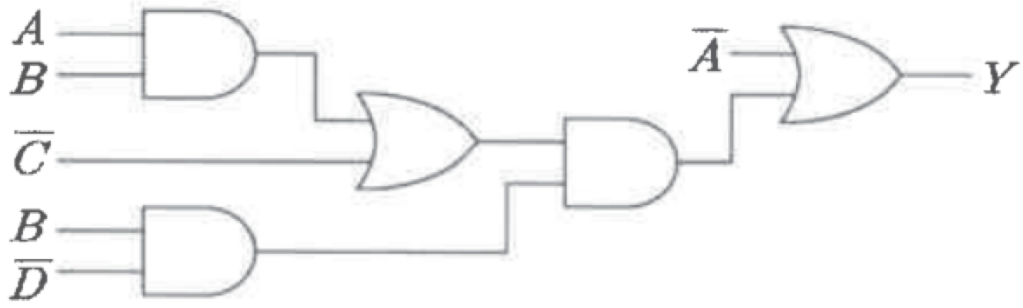
암기 포인트

2의 보수 정수는 오버플로가 없을 때 왼쪽 1비트 시프트마다 값이 2배가 된다.

NoteBase

문 4. <보기>의 회로를 최대로 간략화한 것으로 가장 옳은 것은?

<보기>



① $Y = \overline{A}$

② $Y = \overline{A} + B\overline{D}$

③ $Y = \overline{A} + BC\overline{D}$

④ $Y = \overline{A} + AB\overline{D} + B\overline{CD}$

정답 2번

출제 포인트

이 문제는 회로도의 각 게이트를 논리식으로 옮긴 뒤 전개, 흡수, 포함항 제거를 거쳐 최종식을 가장 짧게 정리할 수 있는지를 묻는다.

선지별 해설

① X. $Y = \overline{A}$만 남으려면 아래쪽의 $B\overline{D}$ 경로가 출력에 영향을 주지 않아야 하는데, 실제 회로에서는 그 경로가 끝까지 살아 있다. 틀린 표현은 $Y = \overline{A}$이고, 올바르게는 $Y = \overline{A} + B\overline{D}$이다.

② O. 회로식을 세우면 $Y = \overline{A} + ((AB + \overline{C}) \cdot B\overline{D})$이고, 이를 정리하면 $Y = \overline{A} + B\overline{D}$가 된다. 선택지 ②가 최대로 간략화한 결과와 일치한다.

③ X. $BC\overline{D}$는 전개 과정에서 나오는 중간항일 뿐 최종항이 아니다. $BC\overline{D} \subseteq B\overline{D}$이므로 $\overline{C}$ 조건은 최종식에서 사라진다. 틀린 표현은 $+BC\overline{D}$이고, 올바르게는 $+B\overline{D}$이다.

④ X. $\overline{A} + AB\overline{D} + B\overline{CD}$는 전개식 그대로이므로 최대 간략형이 아니다. $\overline{A} + AB\overline{D} = \overline{A} + B\overline{D}$이고 $B\overline{CD}$도 $B\overline{D}$에 흡수된다. 틀린 표현은 전개식을 최종식으로 본 점이고, 올바르게는 $Y = \overline{A} + B\overline{D}$이다.

오답 포인트

전개식 $\overline{A} + AB\overline{D} + B\overline{CD}$에서 멈추면 ④에 끌리기 쉽다. 출제자는 전개까지만 하고 흡수법칙과 포함항 제거를 끝까지 하지 않는 실수를 노렸다.

NoteBase

다시 나오면 이렇게 판단

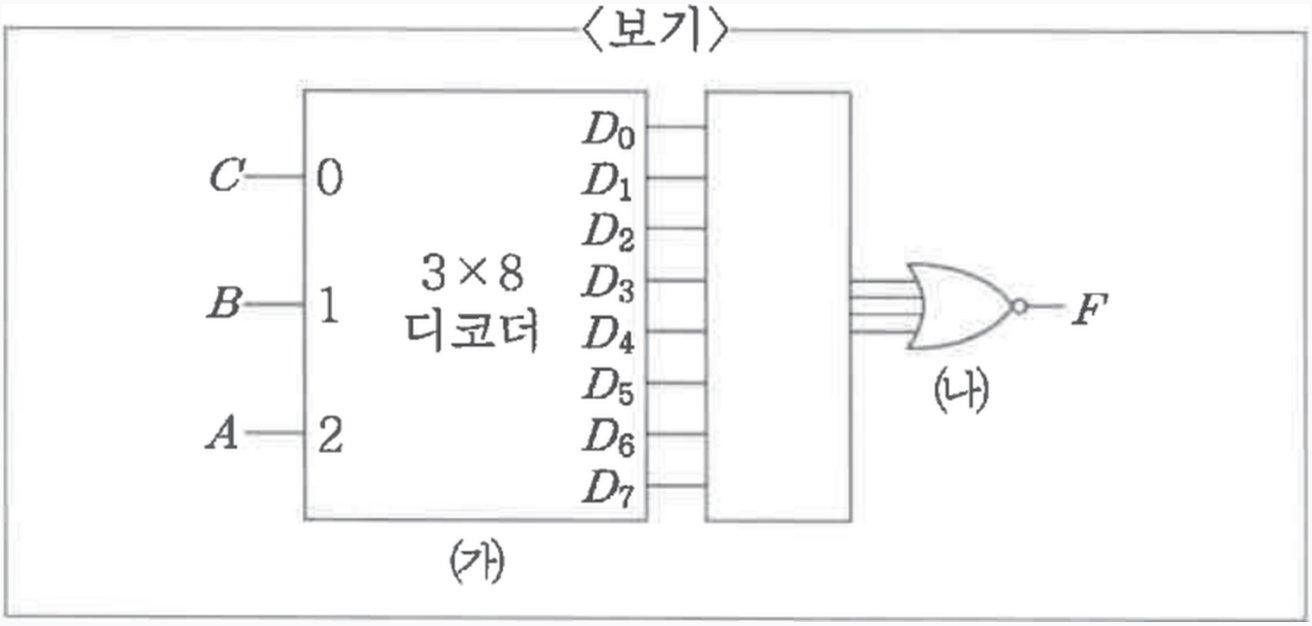
회로 간략화 문제는 게이트를 식으로 바꾼 뒤 괄호를 전개하고, 같은 변수의 중복 제거, 흡수법칙 적용, 더 강한 조건의 항이 더 약한 조건의 항에 포함되는지 확인하는 순서로 정리하면 된다.

암기 포인트

회로식은 전개식이 아니라 흡수 뒤의 식이 답이다. $X + XY = X$, $\overline{X} + XY = \overline{X} + Y$가 간략화의 핵심이다.

문 5. <보기>의 (가)와 (나)를 활용해 논리함수 $F = A \oplus B \oplus C$를 구현하고자 한다. 이때 (나)의 입력으로 연결되어야 할 (가)의 출력을 모두 고른 것은?

<보기>



① D_0, D_2, D_4, D_6

② D_0, D_3, D_5, D_6

③ D_1, D_2, D_4, D_7

④ D_1, D_3, D_5, D_7

정답 2번

출제 포인트

이 문제는 3×8 디코더의 minterm 출력과 출력 버블이 있는 OR, 즉 NOR 게이트의 동작을 연결하여 $F = A \oplus B \oplus C$를 구현하는 문제이다.

NoteBase

선지별 해설

① X: D_0, D_2, D_4, D_6는 짝수 패리티 전체가 아니라 000,010,100,110만 포함하므로 011과 101를 빠뜨리고 001과 111도 반영하지 못한다.
틀린 표현은 출력 집합 선정이다. 올바르게는 D_0, D_3, D_5, D_6를 연결해야 한다.
② O: D_0, D_3, D_5, D_6는 $A \oplus B \oplus C = 0$인 입력 000,011,101,110에 해당한다. 이 네 출력을 NOR에 넣으면 이 경우에만 0이 되고, 나머지 001,010,100,111에서는 1이 되어 $F = A \oplus B \oplus C$가 구현된다.
③ X: D_1, D_2, D_4, D_7는 XOR가 1인 minterm 자체이다. 이것들을 NOR에 연결하면 $A \oplus B \oplus C$가 아니라 그 보수인 $\overline{A \oplus B \oplus C}$가 나온다.
틀린 표현은 NOR 입력에 XOR의 1인 minterm을 직접 연결한 점이다. 올바르게는 XOR가 0인 D_0, D_3, D_5, D_6를 연결해야 한다.
④ X: D_1, D_3, D_5, D_7는 C의 값만 기준으로 홀수·짝수를 구분한 집합이 아니어서 XOR의 0 경우와도 일치하지 않고 1 경우와도 일치하지 않는다.
틀린 표현은 minterm 선택 기준이다. 올바르게는 패리티가 짝수인 000,011,101,110, 즉 D_0, D_3, D_5, D_6를 선택해야 한다.

오답 포인트

출력 버블을 놓치고 (나)를 OR로만 읽으면 D_1, D_2, D_4, D_7를 고르게 된다. 이 문제의 함정은 XOR의 1이 되는 minterm을 찾는 것에서 끝나지 않고, 마지막 게이트가 NOR라는 점까지 반영해야 한다는 데 있다.

다시 나오면 이렇게 판단

디코더 문제는 먼저 함수값이 1이 되는 입력 조합을 minterm으로 적고, 마지막 결합 게이트가 OR인지 NOR인지 확인한 뒤 그대로 묶을지 여집합을 묶을지 결정한다.

암기 포인트

디코더에서 OR는 1인 minterm을 묶고, NOR는 0인 minterm을 묶는다.

NoteBase

문 6. 부호가 있는 2의 보수 방식의 수 체계에서 8비트 2진수로 표현할 수 있는 최댓값과 최솟값을 10진수로 나타낸 값은?

① 최댓값: 127, 최솟값: -128

② 최댓값: 127, 최솟값: -127

③ 최댓값: 128, 최솟값: -128

④ 최댓값: 128, 최솟값: -127

정답 1번

출제 포인트

이 문제는 8비트 2의 보수 표현 범위를 이용해 최댓값과 최솟값을 구할 수 있는지를 묻는다.

조건 정리

주어진 값: 8비트, 부호가 있는 2의 보수 방식
구해야 할 값: 표현 가능한 최댓값과 최솟값의 10진수 값
주의할 조건: 2의 보수는 양수 범위보다 음수 범위가 1만큼 더 크다.

사용할 공식

n비트 2의 보수 범위는 최솟값이 -2^{n-1}, 최댓값이 $2^{n-1}-1$이다.
이 공식을 쓰는 이유: 최상위 1비트가 부호 비트로 쓰이는 2의 보수 표현은 남은 비트 수에 따라 표현 범위가 정해지기 때문이다.

계산 과정

1. 8비트이므로 $n=8$이다.
2. 최솟값은 $-2^{8-1}=-2^7=-128$이다.
3. 최댓값은 $2^{8-1}-1=2^7-1=127$이다.
4. 따라서 가능한 범위는 -128부터 127까지이다.

선지 판단

① O: 최댓값 127, 최솟값 -128은 8비트 2의 보수 범위와 정확히 일치한다.
② X: "최솟값 -127"이 틀렸다. 올바르게는 최솟값이 -128이다.
③ X: "최댓값 128"이 틀렸다. 올바르게는 최댓값이 127이다.
④ X: "최댓값 128, 최솟값 -127"이 모두 틀렸다. 올바르게는 최댓값 127, 최솟값 -128이다.

NoteBase

오답 포인트

부호 비트를 따로 떼어 생각해 −127부터 127까지라고 착각하기 쉽다. 2의 보수는 0이 하나만 존재하므로 음수 쪽이 1개 더 많아 -128까지 표현된다.

다시 나오면 이렇게 판단

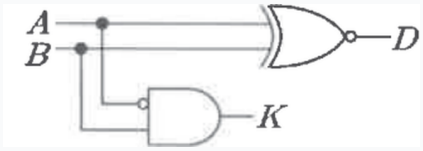
부호 있는 2의 보수 범위를 묻는 문제는 먼저 -2^{n-1}와 $2^{n-1}-1$ 공식을 적용해 음수 쪽이 1만큼 더 넓다는 점까지 함께 확인한다.

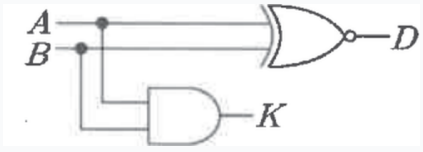
암기 포인트

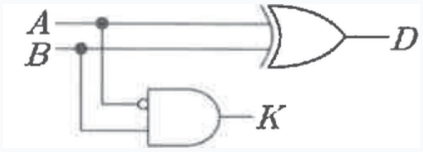
8비트 2의 보수 범위는 -128부터 127까지이다.

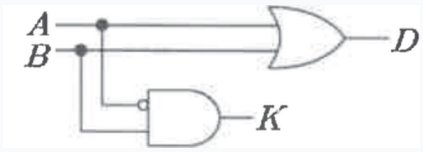
NoteBase

문 7. 반감산기 회로도로 가장 옳은 것은? (단, A와 B는 입력, D는 차이, K는 빌림수이다.)

① 

② 

③ 

④ 

정답 3번

출제 포인트

반감산기의 출력식 $D = A \oplus B$, $K = \overline{A}B$를 게이트 모양과 버블 위치로 판별하는 문제이다.

NoteBase

선지별 해설

① X
- D 게이트가 출력 버블이 있는 XOR 계열이므로 $D = \overline{A \oplus B}$인 XNOR이다.
- 반감산기의 차이는 $A \oplus B$이어야 하므로 D가 틀렸다. K는 $\overline{A}B$로 맞지만, 전체 회로로는 오답이다.
- 틀린 표현은 D 부분이다. 올바르게는 출력 버블이 없는 XOR이어야 한다.

② X
- D가 XNOR이므로 차이 출력이 틀렸다.
- K도 두 입력이 그대로 AND에 들어가므로 $K = AB$이다. 반감산기의 빌림수는 AB가 아니라 $\overline{A}B$이다.
- 틀린 표현은 D와 K 모두이다. 올바르게는 D는 XOR, K는 A 반전 후 B와 AND이다.

③ O
- D가 출력 버블 없는 XOR이므로 $D = A \oplus B$이다.
- K가 A 입력에만 버블이 있는 AND이므로 $K = \overline{A}B$이다.
- 반감산기의 차이와 빌림수 식을 모두 만족하므로 옳다.

④ X
- D 게이트가 XOR의 이중 곡선 입력선이 없는 OR 게이트이므로 $D = A + B$이다.
- 반감산기의 차이는 OR가 아니라 XOR이어야 한다.
- K는 $\overline{A}B$로 맞더라도 D가 틀려 전체 회로는 오답이다.
- 틀린 표현은 D 부분이다. 올바르게는 OR가 아니라 XOR이어야 한다.

오답 포인트

XOR와 OR, XOR와 XNOR를 버블 유무와 입력측 이중 곡선으로 구분하지 못하면 ①이나 ④를 고르기 쉽다. 특히 빌림수 $K = \overline{A}B$만 보고 ①을 정답으로 착각하기 쉽지만, 반감산기는 D와 K가 동시에 맞아야 한다.

다시 나오면 이렇게 판단

반감산기 문제는 먼저 식을 $D = A \oplus B$, $K = \overline{A}B$로 고정한 뒤, D는 XOR인지, K는 A만 반전된 AND인지 순서대로 확인하면 된다.

암기 포인트

반감산기는 차이 XOR, 빌림수는 앞자리 A를 반전한 $\overline{A}B$이다.

NoteBase

문 8. 2진 코드를 그레이 코드로 변환하는 회로에 2진 코드 '1011'이 들어올 때 출력은?

① $(0010)_{gray}$

② $(0101)_{gray}$

③ $(1110)_{gray}$

④ $(1001)_{gray}$

정답 3번

출제 포인트

이 문제는 2진수를 그레이 코드로 바꾸는 규칙 $G_3 = B_3, G_i = B_{i+1} \oplus B_i$를 적용해 출력 코드를 구하는 문제이다.

조건 정리

주어진 값: 2진 코드 1011
구해야 할 값: 이에 대응하는 그레이 코드
주의할 조건: 그레이 코드의 최상위 비트는 원래 2진수의 최상위 비트와 같고, 그 다음 비트부터는 인접한 두 2진 비트를 XOR한 값으로 만든다.

사용할 공식

$G_3 = B_3, \quad G_2 = B_3 \oplus B_2, \quad G_1 = B_2 \oplus B_1, \quad G_0 = B_1 \oplus B_0$
이 규칙은 2진수를 그레이 코드로 변환할 때 각 비트가 바로 앞 비트와의 XOR로 정해지기 때문에 사용한다.

계산 과정

1. 입력 2진수는 $B_3B_2B_1B_0 = 1011$이다.
2. $G_3 = B_3 = 1, G_2 = B_3 \oplus B_2 = 1 \oplus 0 = 1$이다.
3. $G_1 = B_2 \oplus B_1 = 0 \oplus 1 = 1, G_0 = B_1 \oplus B_0 = 1 \oplus 1 = 0$이다.
4. 따라서 출력 그레이 코드는 1110이다.

NoteBase

선지별 해설

① X: 출력은 1110이어야 하므로 0010은 틀리다. 그레이 코드의 최상위 비트는 입력의 최상위 비트와 같아야 하므로 첫 비트가 0인 점부터 맞지 않는다. 틀린 표현은 '0010'이고, 올바르게는 '1110'이다.
② X: 0101은 계산 결과와 다르다. 특히 최상위 비트가 0이어서 $G_3 = B_3 = 1$ 규칙에 어긋난다. 틀린 표현은 '0101'이고, 올바르게는 '1110'이다.
③ O: $G_3 = 1$, $G_2 = 1 \oplus 0 = 1$, $G_1 = 0 \oplus 1 = 1$, $G_0 = 1 \oplus 1 = 0$이므로 1110이 된다. 따라서 ③이 옳다.
④ X: 첫 비트 1은 맞지만 나머지 비트가 규칙과 다르다. 둘째 비트는 $1 \oplus 0 = 1$, 셋째 비트는 $0 \oplus 1 = 1$, 넷째 비트는 $1 \oplus 1 = 0$이어야 하므로 1001은 틀리다. 틀린 표현은 '1001'이고, 올바르게는 '1110'이다.

오답 포인트

그레이 코드를 단순히 오른쪽으로 한 칸 민 값처럼 보거나, 각 비트를 원래 2진수와 그대로 대응시키면 오답이 된다. 이 문항에서는 최상위 비트는 그대로 두고 나머지는 인접 2진 비트 XOR이라는 규칙을 놓치면 ①, ②, ④가 그럴듯해 보인다.

다시 나오면 이렇게 판단

2진수-그레이 코드 변환 문제는 먼저 최상위 비트를 그대로 적고, 그 다음 비트부터는 왼쪽 비트와 현재 비트를 XOR해 순서대로 채우면 된다. 계산이 끝나면 보기의 각 비트를 한 자리씩 대조하면 된다.

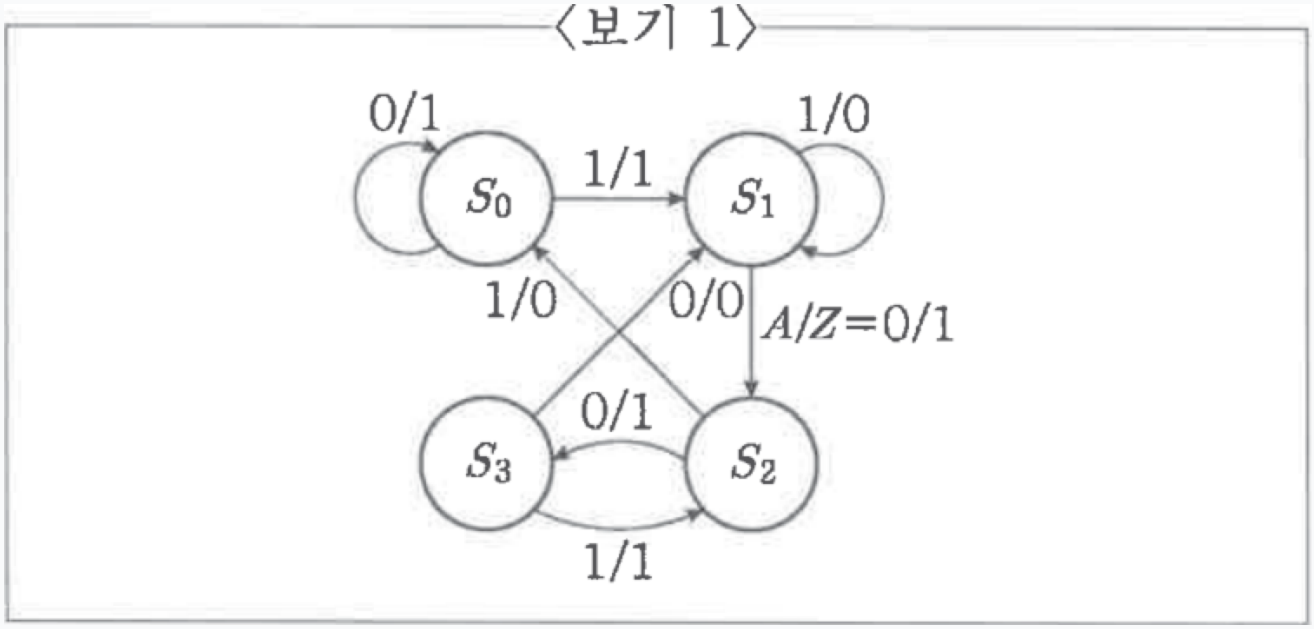
암기 포인트

2진수→그레이 코드는 맨 앞은 그대로, 뒤는 인접 비트 XOR이다.

NoteBase

문 9. 입력 A와 출력 Z를 가지는 한 FSM(Finite State Machine)이 <보기 1>의 상태도를 따르고 <보기 2>의 조건을 만족한다. 이때, <보기 2>의 ㉡에 해당하는 FSM의 상태는?

<보기 1>



<보기 2>

(가) FSM의 현재 상태 = S_2이고 입력 A = ㉠일 때, 출력 Z = 0이다.

(나) FSM의 현재 상태 = ㉡이고 입력 A = ㉠일 때, 차기 상태 = ㉡이다.

① S_0

② S_1

③ S_2

④ S_3

정답 2번

출제 포인트

이 문제는 FSM 상태도에서 입력/출력 라벨을 읽어 ㉠의 입력값과 ㉡의 상태를 역으로 추론하는지를 묻는다.

NoteBase

선지별 해설

① X: S0는 입력 1에서 S1로 이동하므로 차기 상태가 S0가 아니다. 틀린 표현은 '현재 상태가 S0이면 그대로 유지된다'는 가정이다. 올바르게는 '현재 상태가 S0이고 입력 1이면 차기 상태는 S1이다.'이다.
② O: (가)에서 S2에서 출력 Z가 0이 되는 입력은 1뿐이므로 ㉠=1이다. (나)에서 입력 1일 때 차기 상태가 자기 자신인 상태는 S1의 1/0 자기루프뿐이므로 ㉡=S1이다.
③ X: S2는 입력 1에서 S0로 이동하고 출력 0을 낸다. 틀린 표현은 'S2가 입력 1에서 자기 자신으로 유지된다'는 점이다. 올바르게는 '현재 상태가 S2이고 입력 1이면 차기 상태는 S0이다.'이다.
④ X: S3는 입력 1에서 S2로 이동하므로 차기 상태가 S3가 아니다. 틀린 표현은 'S3에 입력 1을 넣으면 자기루프가 형성된다'는 점이다. 올바르게는 '현재 상태가 S3이고 입력 1이면 차기 상태는 S2이다.'이다.

오답 포인트

S2에서 출력이 0이라는 조건만 보고 곧바로 S2를 답으로 고르면 틀린다. 이 문항은 먼저 (가)에서 ㉠=1을 확정한 뒤, 다시 (나)에서 입력 1에 대한 자기루프가 있는 상태를 찾아야 한다.

다시 나오면 이렇게 판단

상태도 문제는 먼저 주어진 출력 조건으로 입력값을 확정하고, 그 입력값에 해당하는 간선들만 다시 훑어 현재 상태와 차기 상태의 관계를 확인하면 된다. 자기 자신으로 돌아오라는 조건이 나오면 자기루프가 있는지부터 본다.

암기 포인트

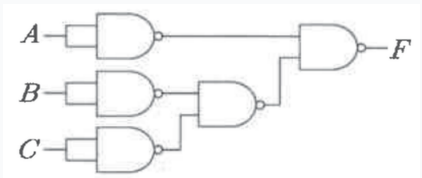
출력 조건으로 입력을 먼저 정하고, 유지 조건은 자기루프로 판별한다.

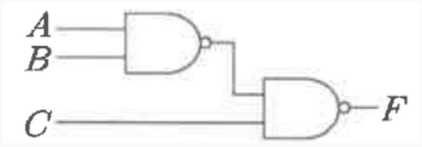
NoteBase

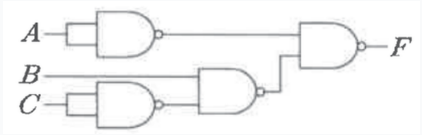
문 10. <보기>의 진리표를 보고 NAND 게이트만을 사용하여 논리회로를 구성한 것은?

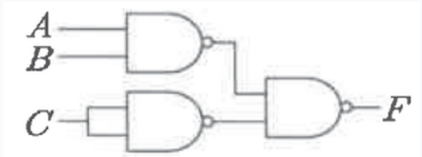
〈보기〉

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

① 

② 

③ 

④ 

정답 4번

출제 포인트

이 문제는 진리표를 식으로 정리한 뒤 각 NAND 회로의 출력식을 환산하여 같은 함수를 찾는 문제이다.

NoteBase

선지별 해설

① X
- A는 자기입력 NAND라서 A'가 되고, B와 C도 각각 반전된 뒤 NAND를 거쳐 $B + C$가 된다. 마지막 NAND 출력은 $(A'(B + C))' = A + B'C'$이다.
- 이 회로의 출력식은 $A + B'C'$이므로 보기의 $F = C + AB$와 다르다. 올바르게는 C 또는 AB가 1이면 출력이 1이 되어야 한다.

② X
- 첫 게이트 출력은 $(AB)'$이고, 마지막 NAND 출력은 $((AB)'C)' = AB + C'$이다.
- 이 회로는 C가 0일 때 1이 되기 쉬운 구조이므로 보기와 반대 방향의 조건을 가진다. 올바르게는 C'가 아니라 C가 OR 항으로 들어가 $AB + C$가 되어야 한다.

③ X
- A는 반전되어 A'가 되고, C는 반전되어 C'가 된다. 가운데 NAND는 B와 C'를 받아 $(BC')'$를 만들고, 마지막 NAND 출력은 $(A'(BC')')' = A + BC'$이다.
- 이 회로의 출력식은 $A + BC'$이므로 $A = 1$이면 거의 항상 1이 되어 보기와 다르다. 올바르게는 A 단독 항이 아니라 AB가 C와 OR 관계를 이루어야 한다.

④ O
- 첫 게이트는 $(AB)'$, 아래 게이트는 C'를 만든다. 마지막 NAND는 $((AB)'C')' = AB + C$가 된다.
- 드모르간 법칙으로 바로 $AB + C$가 나오므로 보기의 진리표와 정확히 일치한다.

오답 포인트

자기입력 NAND를 인버터로 보는 과정과 마지막 NAND에 드모르간 법칙을 적용하는 과정에서 자주 틀린다. 특히 ②를 $AB + C$로 잘못 읽기 쉬운데 실제로는 $((AB)'C)' = AB + C'$이므로 C의 부정 위치를 끝까지 확인해야 한다.

다시 나오면 이렇게 판단

진리표를 먼저 간단한 식으로 줄인 뒤, 각 선택지의 각 NAND 출력을 한 단계씩 식으로 바꾸어 마지막 출력식만 비교하면 된다. 자기입력 NAND는 반전, NAND의 연속 연결은 드모르간 변환으로 정리하는 방식이 가장 빠르다.

암기 포인트

NAND만 있는 회로는 중간출력을 먼저 〃 표기로 적고, 마지막에 드모르간 법칙으로 +와 곱을 바꾸면 정답식이 빠르게 보인다.

NoteBase

문 11. 2진수 $(101101)_2$를 16진수로 변환한 값은?

① $(45)_{16}$

② $(30)_{16}$

③ $(2D)_{16}$

④ $(D2)_{16}$

정답 3번

출제 포인트

이 문제는 2진수를 4비트씩 묶어 16진수로 바꾸는 진법 변환 원리를 묻는 문제이다.

조건 정리

주어진 값: 101101_2
구해야 할 값: 16진수 값
주의할 조건: 오른쪽부터 4비트씩 묶고, 왼쪽이 부족하면 0을 채운다.

사용할 공식

2진수는 4비트씩 묶으면 바로 16진수 한 자리로 대응된다.
$(101101)_2 = (00101101)_2 = (2D)_{16}$
이 공식을 쓰는 이유: 16은 2^4이므로 2진수 4자리가 16진수 1자리와 정확히 대응되기 때문이다.

계산 과정

1. 101101_2를 오른쪽부터 4비트씩 묶기 위해 00101101_2로 맞춘다.
2. 앞의 0010_2는 16진수 2이다.
3. 뒤의 1101_2는 16진수 D이다.
4. 따라서 전체 값은 $2D$이므로 $(101101)_2 = (2D)_{16}$이다.

선지 판단

① X: 45_{16}는 틀리다. 45_{16}는 10진수로 69이므로 101101_2의 값인 45와 다르다. 틀린 표현은 "45"이며 올바르게는 "2D"이다.
② X: 30_{16}는 틀리다. 30_{16}는 10진수로 48이므로 계산 결과와 맞지 않는다. 틀린 표현은 "30"이며 올바르게는 "2D"이다.
③ O: $2D$_{16}가 맞다. 0010 → 2, 1101 → D로 대응되므로 계산 결과와 일치한다.
④ X: $D2$_{16}는 틀리다. 이는 두 묶음의 순서를 거꾸로 읽은 값이다. 틀린 표현은 "D2"이며 올바르게는 "2D"이다.

NoteBase

오답 포인트

왼쪽 비트가 4자리가 안 될 때 0을 채우지 않거나, 4비트 묶음의 순서를 바꾸면 ④ 같은 오답이 나온다. 또 2진수의 10진수 값 45를 그대로 16진수 45로 적으면 ① 같은 오답이 된다.

다시 나오면 이렇게 판단

2진수를 16진수로 바꾸는 문제는 오른쪽부터 4비트씩 묶고, 부족한 왼쪽은 0으로 채운 뒤 각 묶음을 16진수 한 자리로 바꾸면 된다.

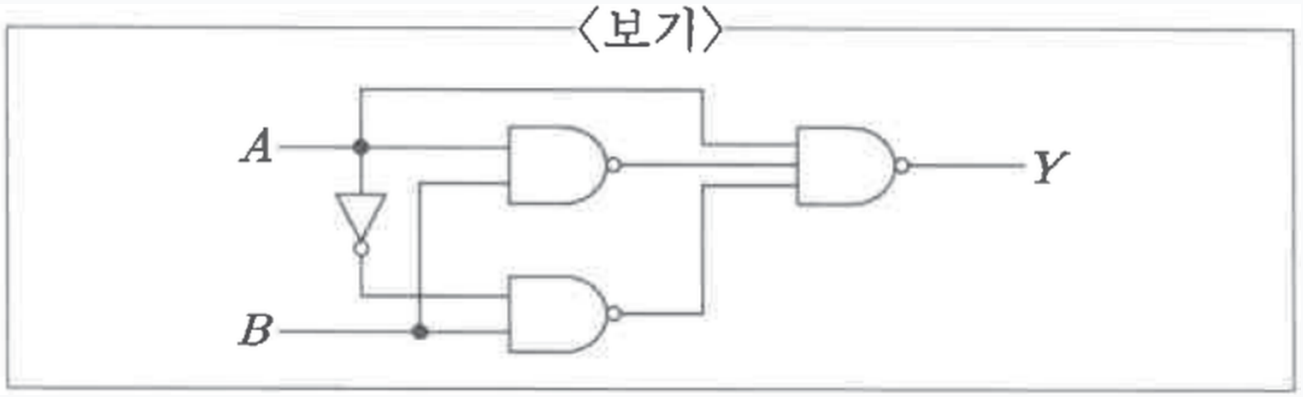
암기 포인트

16진수 1자리는 2진수 4자리와 대응하므로 2진수→16진수는 4비트 묶음 변환으로 처리된다.

NoteBase

문 12. <보기> 회로의 출력 Y가 0이기 위한 입력값의 조건을 가장 옳게 짝지은 것은? (단, x는 무관조건(don't care condition)을 의미한다.)

<보기>



① A=0, B=x

② A=1, B=0

③ A=1, B=1

④ A=x, B=1

정답 2번

출제 포인트

이 문제는 회로도에서 각 NAND 게이트와 인버터의 연결을 불 대수식으로 바꾸어 $Y=0$이 되는 입력 조건을 찾는 문제이다.

선지별 해설

① X: $A=0$이면 $Y=\overline{A}+B=1+B=1$이므로 B 값과 무관하게 0이 될 수 없다. 틀린 표현은 "A=0, B=X"이고, 올바르게는 "A=1, B=0"일 때만 $Y=0$이다.

② O: $A=1, B=0$이면 $Y=\overline{A}+B=0+0=0$이므로 출력이 0이 된다. $Y=0$이 되려면 보수식 $\overline{AB'}$의 내부가 1이어야 하므로 $A=1, B=0$이 정확하다.

③ X: $A=1, B=1$이면 $Y=\overline{A}+B=0+1=1$이므로 출력은 1이다. 틀린 표현은 "A=1, B=1"이고, 올바르게는 B가 1이 아니라 0이어야 한다.

④ X: $B=1$이면 $Y=\overline{A}+1=1$이므로 A 값과 무관하게 0이 될 수 없다. 틀린 표현은 "A=X, B=1"이고, 올바르게는 "A=1, B=0"이다.

NoteBase

오답 포인트

최종 게이트에 들어가는 A의 직접 입력을 빼먹거나, $Y = 0$ 조건을 찾으면서 중간 NAND 출력이 0이 되는 경우만 따지면 오답이 된다. 이 문항은 각 게이트 출력을 먼저 식으로 바꾼 뒤 마지막 NAND 전체를 한 번에 정리해야 흔들리지 않는다.

다시 나오면 이렇게 판단

회로 문제는 입력선의 분기와 버블 위치를 먼저 확인한 뒤 각 중간 출력을 식으로 쓰고, 마지막 게이트 직전 식을 불 대수로 단순화하여 Y를 한 변수식으로 만든 다음 조건을 대입해 판단한다.

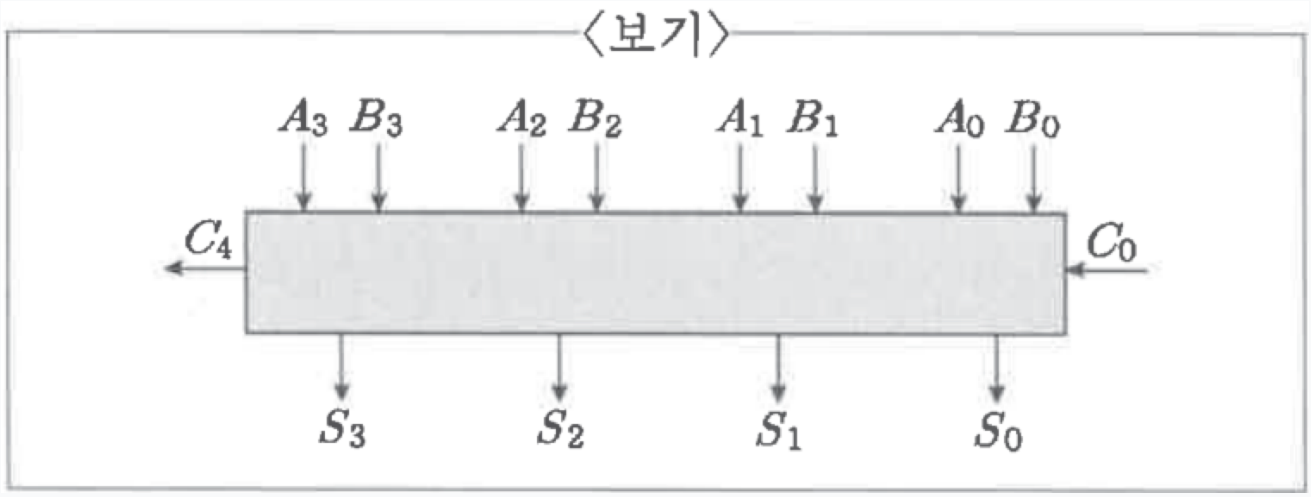
암기 포인트

NAND 회로는 중간식을 먼저 구하고 마지막 버블을 드모르간 법칙으로 정리하면 $Y = 0$ 조건이 가장 빠르게 보인다.

NoteBase

문 13. ROM만을 이용하여 <보기>와 같은 4비트 덧셈기 회로를 구현하고자 할 때, 필요한 ROM 메모리의 용량은?

<보기>



① 9 × 32[비트]

② 9 × 32[바이트]

③ 512 × 5[비트]

④ 512 × 5[바이트]

정답 3번

출제 포인트

이 문제는 ROM으로 조합회로를 구현할 때 입력 수로 주소 수를 정하고 출력 수로 워드 길이를 정하는 원리를 묻는다.

조건 정리

주어진 입력: A3, B3, A2, B2, A1, B1, A0, B0, C0의 9개 입력이다.
구해야 할 값: ROM의 전체 용량이다.
주의할 조건: ROM 용량은 주소 수 × 워드당 비트 수로 계산하며 단위는 비트이다.

사용할 공식

주소 수는 2^n, 워드 길이는 출력 비트 수, ROM 용량은 $2^n \times m$비트이다.
이 공식을 쓰는 이유: ROM은 입력 조합마다 하나의 주소가 필요하고, 각 주소에서 회로의 모든 출력값을 한 번에 내보내기 때문이다.

NoteBase

계산 과정

1. 덧셈기의 입력은 A와 B의 각 4비트 8개와 자리올림 입력 C0 1개이므로 주소선 수는 9개이다.
2. 따라서 필요한 주소 수는 $2^9 = 512$개이다.
3. 출력은 합 S3, S2, S1, S0의 4비트와 최종 자리올림 C4의 1비트이므로 총 5비트이다.
4. 따라서 필요한 ROM 용량은 512×5비트이다.

선지 판단

① X: 입력 수 9를 그대로 곱하고 출력 수 5에만 $2^5 = 32$를 적용한 형태이다. ROM에서는 입력 수가 주소선 수이므로 2^9가 되어야 한다. 틀린 표현은 "9×32비트"이고, 올바르게는 "512×5비트"이다.
② X: ①과 같은 입력·출력 뒤바뀜 오류에 더해 단위를 바이트로 잘못 적었다. 5는 5바이트가 아니라 5비트 출력이다. 틀린 표현은 "9×32바이트"이고, 올바르게는 "512×5비트"이다.
③ O: 입력 9개로 주소 수는 512개이고, 출력은 S0~S3와 C4의 5비트이므로 512×5비트가 맞다.
④ X: 주소 수 512와 출력 수 5 자체는 맞지만 단위를 바이트로 잘못 적었다. 틀린 표현은 "5바이트"이고, 올바르게는 "5비트"이다.

오답 포인트

입력선 개수와 출력선 개수를 뒤집어 9×2^5처럼 계산하거나, 출력선 수를 비트가 아니라 바이트로 착각하면 틀린다.

다시 나오면 이렇게 판단

ROM 구현 문제는 먼저 입력선 개수 n을 세어 주소 수를 2^n으로 정하고, 다음에 출력선 개수만큼 워드 길이를 정한 뒤 단위를 비트로 붙여 판단한다.

암기 포인트

ROM 용량은 입력선은 주소 수로, 출력선은 워드 길이로 바꿔 2^n × 출력비트수로 계산한다.

NoteBase

문 14. 서로 다른 부울 변수 A, B, C, D에 대한 함수 F, G가 있다. 함수 F, G의 4변수 카르노 맵이 각각 <보기 1>, <보기 2>와 같을 때, $F \oplus G$의 부울식은?

〈보기 1〉

CD \ AB	00	01	11	10
00	0	1	0	1
01	1	0	0	0
11	1	1	1	1
10	1	1	0	1

〈보기 2〉

CD \ AB	00	01	11	10
00	1	1	0	0
01	0	0	0	1
11	0	1	1	0
10	0	1	0	0

① $\overline{B}$

② CD

③ $A + BC$

④ $(A + B) + C\overline{D}$

정답 1번

출제 포인트

이 문제는 두 4변수 카르노 맵을 칸별로 XOR한 뒤 반복되는 열 패턴만 남겨 최종 부울식을 읽어내는 문제이다.

선지별 해설

① O: $\overline{B}$는 $B = 0$일 때만 1이다. 결과 맵이 AB=00, 10 열에서만 모두 1이므로 정확히 일치한다.
② X: CD라면 CD=11인 한 행에서만 1이 나와야 한다. 실제 결과 맵은 특정 행이 아니라 AB=00, 10 두 열 전체가 1이므로 CD가 아니다. 틀린 표현은 "CD"이고, 올바르게는 "$\overline{B}$"이다.
③ X: $A + BC$는 AB=10, 11이면 항상 1이 되거나, B와 C의 조합에 따라 값이 달라진다. 실제 결과 맵은 AB=11 열이 전부 0이고 AB=00 열이 전부 1이므로 맞지 않는다. 틀린 표현은 "$A + BC$"이고, 올바르게는 "$\overline{B}$"이다.
④ X: $(A + B) + CD$는 B=1인 AB=01 열에서 이미 1이 되어야 한다. 실제 결과 맵의 AB=01 열은 모든 칸이 0이므로 반대 패턴이다. 틀린 표현은 "$(A + B) + CD$"이고, 올바르게는 "$\overline{B}$"이다.

오답 포인트

원래의 F, G를 각각 간소화하려고 들면 시간이 오래 걸린다. 이 문항은 먼저 XOR를 칸별로 계산해 결과 맵의 반복 패턴을 보는 것이 핵심이며, 행 전체가 아니라 열 전체가 반복된다는 점을 놓치면 ②처럼 CD에 끌리기 쉽다.

NoteBase

다시 나오면 이렇게 판단

두 카르노 맵의 XOR나 XNOR를 묻는 문항은 원래 식을 따로 구하지 않고 같은 위치의 값을 바로 비교한다. 그다음 결과 맵에서 행 전체가 반복되는지, 열 전체가 반복되는지를 먼저 보고 남는 변수를 결정한다.

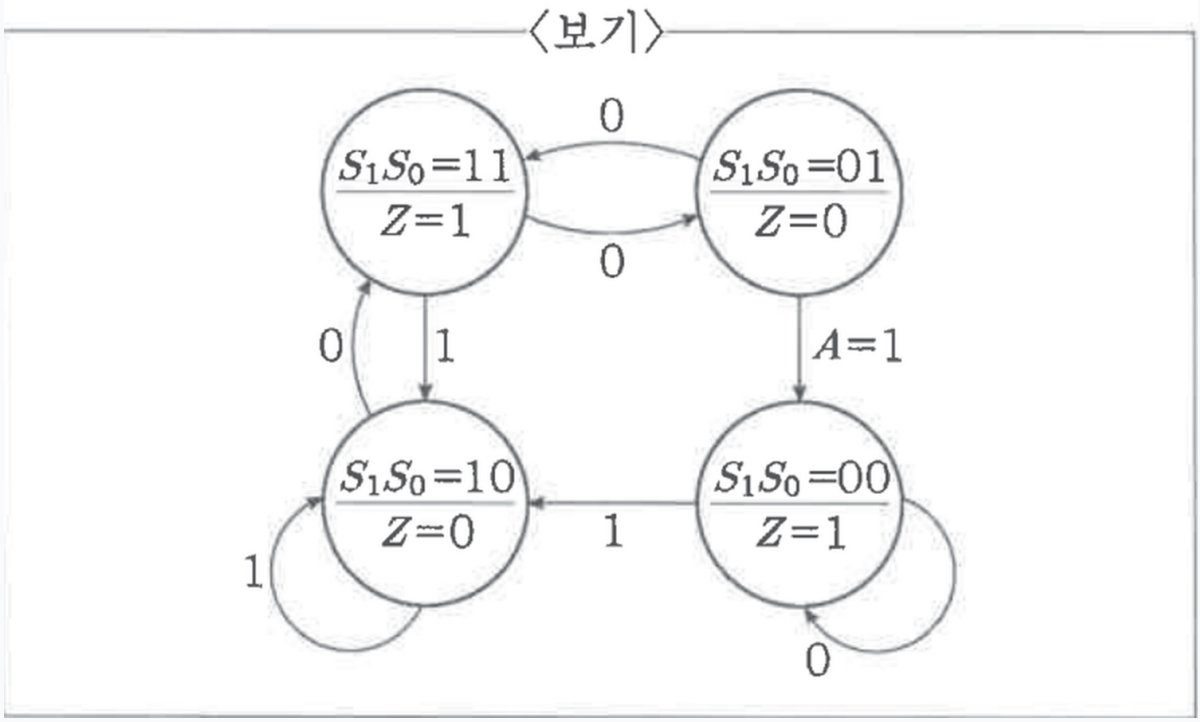
암기 포인트

카르노 맵의 전 열이 1이면 그 열 조건만 남고, AB=00·10만 1이면 $B = 0$, 즉 $\overline{B}$이다.

NoteBase

문 15. <보기>는 한 FSM(Finite State Machine)의 상태도이고 S_1S_0은 현재 상태, A는 입력, Z는 출력을 의미한다. 이때, Z의 논리식은?

<보기>



① $Z = S_1 \oplus S_0$

② $Z = \overline{S_1S_0} + S_1S_0$

③ $Z = S_1 \oplus S_0 \oplus A$

④ $Z = \overline{S_1S_0A} + \overline{S_1}S_0 + S_1\overline{S_0A}$

정답 2번

출제 포인트

이 문제는 FSM 상태도에서 각 상태에 붙은 출력값을 읽어 Moore형 출력식 Z를 현재 상태 변수 S_1, S_0만으로 표현하는지를 묻는 문제이다.

NoteBase

선지별 해설

① X

- $S_1 \oplus S_0$는 상태 01, 10에서 1이 되고 00, 11에서 0이 된다. 그림은 그 반대로 00, 11에서 $Z = 1$이므로 맞지 않는다.
- 틀린 표현은 "배타적 논리합"이다. 올바르게는 $Z = \overline{S_1 S_0} + S_1 S_0$이다.

② O

- 상태 00과 11에서만 $Z = 1$, 상태 01과 10에서 $Z = 0$이므로 $Z = \overline{S_1 S_0} + S_1 S_0$가 정확하다.
- 이는 두 상태 비트가 같을 때 1이 되는 XNOR 식과 동일하다.

③ X

- 이 식은 입력 A를 포함한다. 그러나 그림의 출력 Z는 각 상태 원 안에 고정되어 있으므로 현재 상태만으로 정해지는 Moore형 출력이어야 한다.
- 틀린 표현은 "$\oplus A$" 부분이다. 올바르게는 A가 빠진 $Z = \overline{S_1 S_0} + S_1 S_0$이다.

④ X

- 이 식도 A를 포함하므로 상태가 같아도 입력에 따라 출력이 바뀌게 된다. 그림의 Z는 상태 00, 11에서 항상 1이고 01, 10에서 항상 0이므로 이런 형태가 될 수 없다.
- 틀린 표현은 입력 A에 의존하는 항들이다. 올바르게는 상태 일치만 반영한 $Z = \overline{S_1 S_0} + S_1 S_0$이다.

오답 포인트

전이 화살표에 적힌 0, 1, $A = 1$을 보고 출력식에도 입력 A가 들어간다고 착각하기 쉽다. 이 문제는 전이조건이 아니라 상태 원 안에 적힌 Z 값을 읽는 문제이므로, 입력이 포함된 ③과 ④는 구조적으로 먼저 제외된다.

다시 나오면 이렇게 판단

FSM에서 출력이 상태 원 안에 적혀 있으면 먼저 Moore형인지 확인한다. Moore형이면 출력식은 현재 상태 변수만으로 세우고, 각 상태에서 출력이 1인 경우의 최소항만 합하면 된다.

암기 포인트

Moore 출력은 상태만 본다. 이 문항의 Z는 00, 11에서 1이므로 XNOR, 즉 $\overline{S_1 S_0} + S_1 S_0$이다.

NoteBase

문 16. 3 초과 코드(excess-3 code)로 나타낸 값 1010011010111001의 9의 보수를 3 초과 코드로 표현한 것은?

① (0101100101001010)_3 초과
② (0101100101000111)_3 초과
③ (0101100101001001)_3 초과
④ (0101100101000110)_3 초과

정답 4번

출제 포인트

이 문제는 3 초과 코드의 자기보수성으로 9의 보수를 구할 수 있는지를 묻는 문제이다.

조건 정리

주어진 값: 1010 0110 1011 1001
구해야 할 값: 이 값의 9의 보수를 3 초과 코드로 나타낸 비트열
주의할 조건: 3 초과 코드는 4비트씩 한 자리이며, 각 자리의 실제 10진수 값은 코드값에서 3을 뺀 값이다.

사용할 공식

3 초과 코드는 각 10진수 자리 d를 $d + 3$의 4비트 코드로 나타낸다.
또한 자기보수 코드이므로 한 자리의 9의 보수 $9 - d$는 해당 3 초과 코드의 비트 반전으로 얻어진다.
이 원리를 쓰는 이유는 9의 보수를 묻는 문항이라 원래 값을 풀어 다시 부호화해도 되고, 각 4비트를 바로 반전해도 같은 결과가 나오기 때문이다.

계산 과정

1. 4비트씩 끊으면 1010 0110 1011 1001이다.
2. 각 묶음에서 3을 빼면 10-3=7, 6-3=3, 11-3=8, 9-3=6이므로 원래 값은 7386이다.
3. 7386의 9의 보수는 2613이다.
4. 2613을 3 초과 코드로 바꾸면 2→0101, 6→1001, 1→0100, 3→0110이므로 0101100101000110이다.
5. 자기보수 코드 성질로 원래 코드 1010011010111001의 각 비트를 반전해도 0101100101000110이 되어 같은 답이 나온다.

NoteBase

선지 판단

① X
- 0101 1001 0100 1010은 2, 6, 1, 7을 뜻하므로 2617이다.
- 틀린 표현은 마지막 4비트 1010이다. 올바르게는 3의 3 초과 코드인 0110이어야 하므로 전체는 0101100101000110이다.

② X
- 0101 1001 0100 0111은 2, 6, 1, 4를 뜻하므로 2614이다.
- 이는 9의 보수 2613이 아니라 10의 보수 쪽 결과에 해당한다.
- 틀린 표현은 마지막 4비트 0111이다. 올바르게는 0110이다.

③ X
- 0101 1001 0100 1001은 2, 6, 1, 6을 뜻하므로 2616이다.
- 마지막 자리 6을 그대로 둔 꼴이라 6의 9의 보수인 3이 반영되지 않았다.
- 틀린 표현은 마지막 4비트 1001이다. 올바르게는 0110이다.

④ O
- 0101 1001 0100 0110은 2, 6, 1, 3을 뜻하므로 2613이다.
- 원래 코드 1010 0110 1011 1001을 비트 반전해도 같은 값이 나오므로 9의 보수 코드와 일치한다.

오답 포인트

3 초과 코드는 비트열 자체가 아니라 각 4비트 묶음의 코드값에서 3을 빼서 읽어야 한다. 또한 9의 보수와 10의 보수를 혼동하면 ②처럼 마지막 자리를 4로 적기 쉽고, 자기보수 코드 성질을 놓치면 ③처럼 마지막 자리를 원래 값 6으로 남기는 실수가 나온다.

다시 나오면 이렇게 판단

3 초과 코드의 보수 문제는 4비트씩 끊어 원래 10진수를 읽고, 각 자리의 9의 보수를 구한 뒤 다시 3을 더해 부호화하면 된다. 3 초과 코드는 자기보수 코드이므로 각 비트를 반전한 값과 같은지도 함께 검산할 수 있다.

암기 포인트

3 초과 코드는 자기보수 코드이므로 9의 보수는 각 4비트의 비트 반전과 같다.

NoteBase

문 17. 서로 다른 부울 변수 A, B에 대한 함수 F의 최소항 전개식은 $F(A,B) = m_0 + m_2$이고, 함수 G의 최대항 전개식은 $G(A,B) = M_2M_3$이다. 함수 $Z = FG$일 때, Z의 최대항 전개식은?

① $Z(A,B) = M_0$

② $Z(A,B) = M_2$

③ $Z(A,B) = M_0M_3$

④ $Z(A,B) = M_1M_2M_3$

정답 4번

출제 포인트

이 문제는 최소항 전개식과 최대항 전개식을 진리표로 바꾼 뒤, 곱 함수 $Z = FG$를 구하여 다시 최대항 전개식으로 바꿀 수 있는지를 묻는다.

조건 정리

주어진 값: $F(A,B) = m_0 + m_2$, $G(A,B) = M_2M_3$이다.
구해야 할 값: $Z = FG$의 최대항 전개식이다.
주의할 조건: 최소항은 함수값이 1인 위치를, 최대항은 함수값이 0인 위치를 나타낸다.

사용할 공식

$F = \Sigma m(0, 2)$이면 0번과 2번에서만 1이고, $G = \Pi M(2, 3)$이면 2번과 3번에서만 0이다.
따라서 두 함수를 먼저 진리표로 읽은 뒤 $Z = F \cdot G$를 구하고, 마지막에 $Z = 0$이 되는 인덱스를 모아 최대항 전개식으로 적는다.

계산 과정

두 변수의 인덱스는 $00 \to m_0$, $01 \to m_1$, $10 \to m_2$, $11 \to m_3$이다.
$F = m_0 + m_2$이므로 F는 00,10에서 1이어서 $F = \overline{B}$이다.
$G = M_2M_3$이므로 G는 10,11에서 0이어서 $G = \overline{A}$이다.
따라서 $Z = FG = \overline{B} \cdot \overline{A} = \overline{AB} = m_0$이다.
Z는 0번에서만 1이고 1, 2, 3번에서 0이므로 최대항 전개식은 $M_1M_2M_3$이다.

NoteBase

선지 판단

① X: M_0는 0번에서 0이고 나머지에서 1인 함수이다. 구한 Z는 0번에서만 1이므로 일치하지 않는다. 틀린 표현은 최대항 M_0만으로 나타낸 부분이며, 올바르게는 $Z(A,B) = M_1M_2M_3$이다.
② X: M_2는 2번에서만 0이고 나머지에서 1인 함수이다. Z는 1, 2, 3번에서 모두 0이므로 부족하다. 틀린 표현은 2번만 0으로 본 부분이며, 올바르게는 $Z(A,B) = M_1M_2M_3$이다.
③ X: M_0M_3는 0번과 3번에서 0인 함수이다. Z는 0번에서 1이어야 하므로 정반대 위치를 포함한다. 틀린 표현은 0번을 최대항으로 넣은 부분이며, 올바르게는 $Z(A,B) = M_1M_2M_3$이다.
④ O: $Z = \overline{AB} = m_0$이므로 1이 되는 곳은 0번뿐이다. 따라서 0이 되는 1, 2, 3번을 곱한 최대항 전개식 $M_1M_2M_3$가 맞다.

오답 포인트

최소항 m_i와 최대항 M_i의 의미를 뒤바꾸면 ①처럼 잘못 고르기 쉽다. 또한 $Z = m_0$라는 결과를 얻고도 이를 그대로 최대항으로 쓰면 최대항 전개식 요구를 놓치게 된다.

다시 나오면 이렇게 판단

부울 전개식 문제는 먼저 각 항이 1을 뜻하는지 0을 뜻하는지 확인하고, 진리표에서 결과 함수의 1 위치와 0 위치를 구분한 뒤, 최소항을 묻는지 최대항을 묻는지에 맞춰 마지막 표현만 바꿔 적으면 된다.

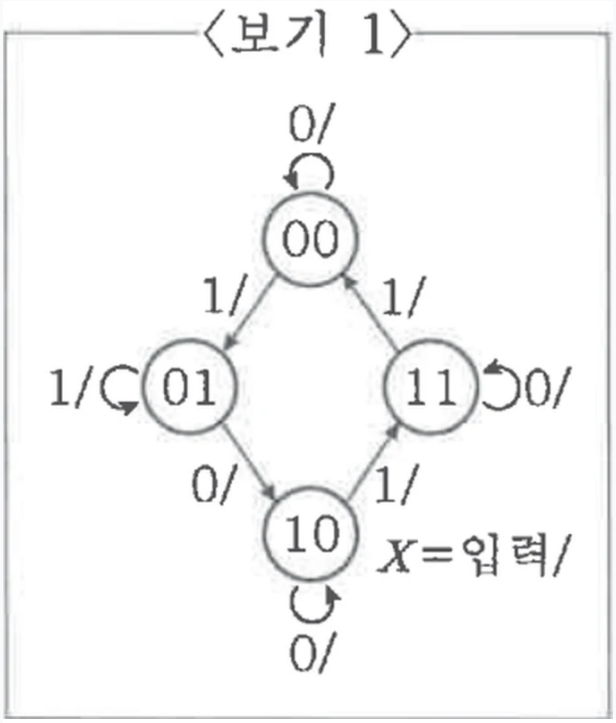
암기 포인트

최소항은 1의 위치, 최대항은 0의 위치를 모은 표현이다. $Z = m_0$이면 같은 함수의 최대항 전개식은 $M_1M_2M_3$이다.

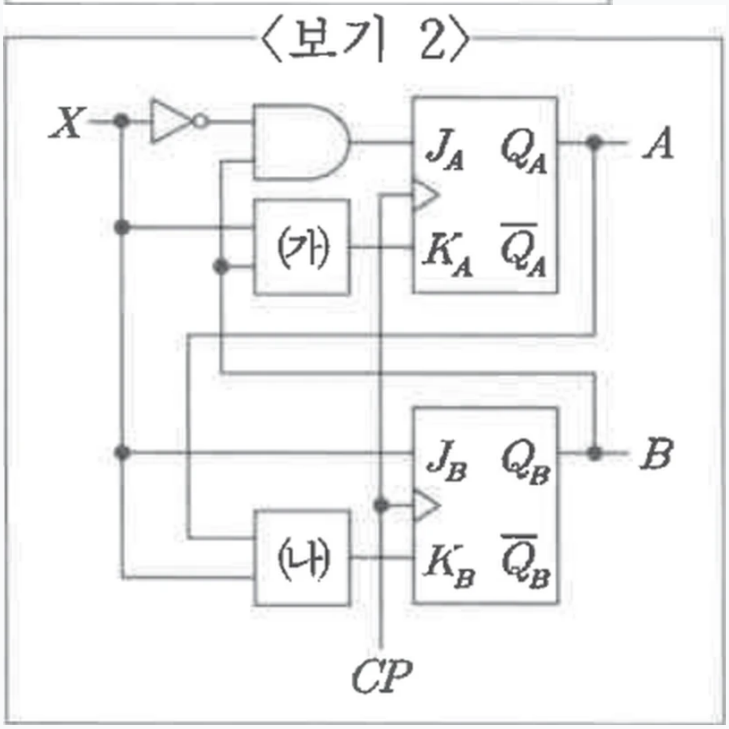
NoteBase

문 18. <보기 1>의 상태도를 나타내는 동기 순서논리회로를 2개의 $J-K$ 플립플롭을 이용하여 <보기 2>와 같이 구현할 때, <보기 2>의 (가)와 (나)에 들어갈 게이트를 옳게 짝지은 것은? (단, Q_A가 MSB, Q_B가 LSB이다.)

<보기 1>



<보기 2>



① (가) (나)



NoteBase

② 

③ 

④ 

정답 1번

출제 포인트

이 문제는 상태도에서 각 비트의 다음 상태를 읽어 JK 플립플롭의 여기식으로 바꾼 뒤, 빈 칸 (가)와 (나)에 들어갈 게이트 종류를 판단하는 문제이다.

선지별 해설

① O

(가)는 AND 게이트, (나)는 XNOR 게이트이다. 상태도에서 Q_A가 0에서 1로 바뀌는 경우는 01 상태에서 $X=0$일 때뿐이므로 $J_A=\overline{X}Q_B$가 된다. 또한 Q_A가 1에서 0으로 바뀌는 경우는 11 상태에서 $X=1$일 때뿐이므로 $K_A=XQ_B$가 되어 (가)는 AND가 된다. Q_B는 $Q_B=1$인 상태에서 다음 값이 0이 되는 경우가 01에서 $X=0$, 11에서 $X=1$일 때이므로 $K_B=\overline{Q_A\oplus X}$가 되어 (나)는 XNOR가 된다.

② X

(가)를 OR로 두면 K_A가 $X+Q_B$ 꼴이 되어 10 상태처럼 $Q_B=0$이어도 $X=1$이면 $K_A=1$이 된다. 그러면 10에서 11로 가야 할 때 Q_A가 1에서 0으로 잘못 토글될 수 있다. (가)는 OR가 아니라 AND이어야 한다. 또한 (나)를 XOR로 두면 01에서 $X=0$일 때 $K_B=0$이 되어야 하는데 실제로는 1이 필요하므로 맞지 않는다. (나)는 XOR가 아니라 XNOR이어야 한다.

③ X

(가)는 AND라서 맞지만 (나)를 AND로 두면 K_B가 두 입력이 모두 1일 때만 1이 된다. 그러나 상태도에서는 01에서 $X=0$일 때도 Q_B를 1에서 0으로 만들어야 하므로 $K_B=1$이 필요하다. (나)는 AND가 아니라 XNOR이어야 한다.

④ X

(가)를 OR로 두는 순간 K_A 조건이 과도하게 넓어져 Q_A가 불필요하게 리셋된다. 또한 (나)를 OR로 두면 01에서 $X=1$처럼 Q_B를 유지해야 하는 경우에도 $K_B=1$이 될 수 있어 상태도가 깨진다. (가)는 OR가 아니라 AND, (나)는 OR가 아니라 XNOR이어야 한다.

NoteBase

오답 포인트

JK 플립플롭 문제에서 많은 경우 J와 K를 동시에 같은 종류의 게이트로 맞추려다 틀린다. 이 문항은 각 입력을 독립적으로 여기표에 대입해야 하며, 특히 K_B가 01에서 $X=0$, 11에서 $X=1$일 때만 1이 된다는 점을 놓치면 XOR와 XNOR를 뒤집어 고르게 된다.

다시 나오면 이렇게 판단

상태도에서 현재상태와 입력으로 다음상태를 먼저 표로 정리한 뒤, 각 비트에 대해 0→1이면 $J=1$, 1→0이면 $K=1$만 뽑아낸다. 그다음 1이 되는 조합만 묶어 식을 만들면 게이트 종류를 바로 고를 수 있다.

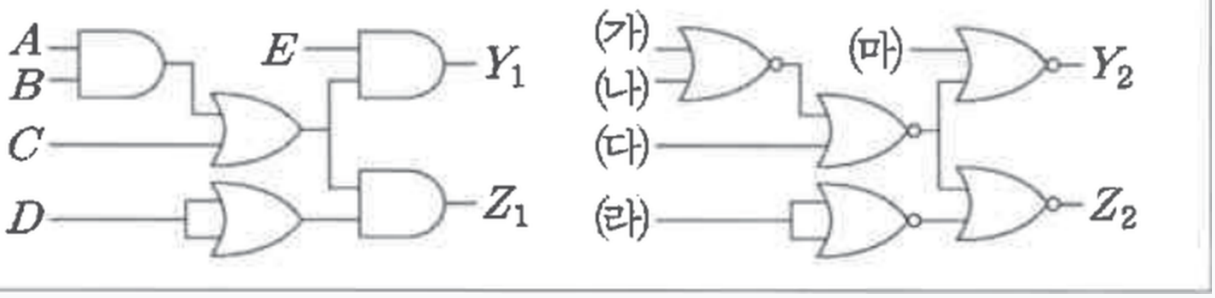
암기 포인트

JK 구현은 상태천이표를 여기표로 바꾸는 문제이며, 이 문항의 핵심 식은 $K_A = XQ_B$, $K_B = \overline{Q_A \oplus X}$이다.

NoteBase

문 19. <보기>에서 $Y_1 = Y_2, Z_1 = Z_2$일 때, (가)~(마)를 옳게 짝지은 것은?

<보기>



① (가): A, (나): B, (다): $\overline{C}$, (라): $\overline{D}$, (마): E

② (가): $\overline{A}$, (나): $\overline{B}$, (다): $\overline{C}$, (라): D, (마): E

③ (가): $\overline{A}$, (나): $\overline{B}$, (다): C, (라): D, (마): $\overline{E}$

④ (가): $\overline{A}$, (나): $\overline{B}$, (다): $\overline{C}$, (라): $\overline{D}$, (마): $\overline{E}$

정답 3번

출제 포인트

드모르간 법칙과 NOR 게이트의 자기입력 결선을 이용해 AND, OR, NOT을 NOR만으로 등가 변환할 수 있는지를 묻는 문제이다.

선지별 해설

① X: (가)=A, (나)=B이면 첫 NOR 출력이 $\overline{A+B}$가 되어 AB를 만들지 못한다. 이어서 가운데 NOR는 $\overline{\overline{A+B}+\overline{C}} = (A+B)C$가 되어 왼쪽 회로의 $AB+C$와 달라진다. 틀린 표현은 (가)=A, (나)=B, (다)=C☒, (라)=D☒, (마)=E의 조합이다. 올바르게는 (가)=A☒, (나)=B☒, (다)=C, (라)=D, (마)=E☒이어야 한다.

② X: (가)=A☒, (나)=B☒는 맞지만 (다)=C☒, (마)=E가 틀리다. 이때 첫 NOR는 AB를 만들지만 가운데 NOR 출력의 보수는 $AB+\overline{C}$가 되고, 위쪽 마지막 NOR에 E가 직접 들어가므로 Y_2가 $E(AB+C)$와 같아지지 않는다. 틀린 표현은 (다)=C☒, (마)=E이다. 올바르게는 (다)=C, (마)=E☒이어야 한다.

③ O: (가)=A☒, (나)=B☒이면 첫 NOR가 $\overline{\overline{A}+\overline{B}} = AB$를 만든다. 여기에 (다)=C를 넣으면 가운데 NOR 출력이 $\overline{AB+C}$가 되고, (마)=E☒를 위 NOR에 넣으면 $Y_2 = \overline{\overline{E}+\overline{(AB+C)}} = E(AB+C) = Y_1$이다. 또 (라)=D이면 아래 NOR 출력이 $\overline{D}$가 되어 $Z_2 = \overline{\overline{(AB+C)}+\overline{D}} = D(AB+C) = Z_1$이다.

④ X: (가)=A☒, (나)=B☒, (마)=E☒는 맞지만 (다)=C☒, (라)=D☒가 틀리다. 이 경우 가운데 NOR 출력의 보수는 $AB+\overline{C}$, 아래 NOR 출력은 D가 되어 Y_2, Z_2 모두 왼쪽 회로와 달라진다. 틀린 표현은 (다)=C☒, (라)=D☒이다. 올바르게는 (다)=C, (라)=D이어야 한다.

NoteBase

오답 포인트

첫 NOR로 AB를 만들려면 입력이 A, B가 아니라 A□, B□여야 한다는 점과, 같은 신호를 두 입력에 묶은 NOR가 NOT이 된다는 점을 놓치기 쉽다. 또한 C와 E의 보수 위치를 한 칸씩 잘못 놓으면 $AB + C$가 아니라 $AB + \bar{C}$가 되어 전체 식이 바로 어긋난다.

다시 나오면 이렇게 판단

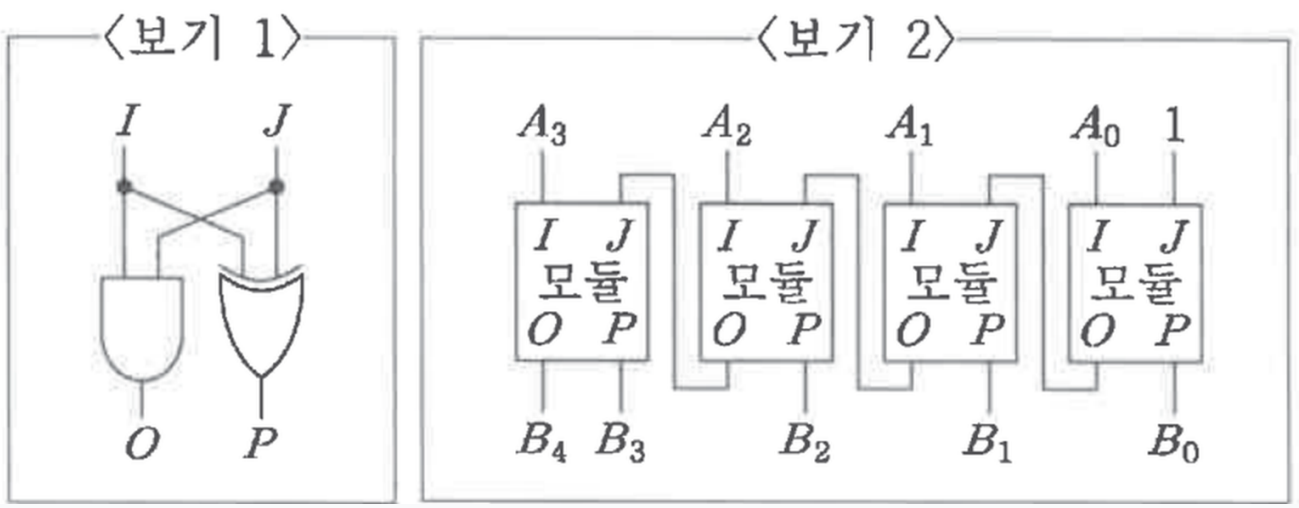
같은 입력 두 개가 들어간 NOR를 먼저 NOT으로 바꾼다. 그다음 첫 NOR가 AB를 만들 수 있도록 보수 입력 A□, B□를 찾고, 가운데 NOR가 $\overline{AB + C}$가 되도록 C를 넣는다. 마지막 단계에서 위쪽은 E□, 아래쪽은 D에서 만든 D□가 들어가야 각각 Y_1, Z_1과 같아진다.

암기 포인트

NOR 회로 변환의 핵심은 $\overline{X + X} = \overline{X}$, $\overline{\overline{X} + \overline{Y}} = XY$ 두 식이다.

NoteBase

문 20. <보기 1>의 모듈 4개를 <보기 2>와 같이 연결하였다. $A_3A_2A_1A_0 = (1011)_2$일 때, $B_4B_3B_2B_1B_0$의 값은?



① $(01100)_2$

② $(01110)_2$

③ $(10101)_2$

④ $(11111)_2$

정답 1번

출제 포인트

이 문제는 모듈의 논리 기능을 읽어 오른쪽의 상수 1이 왼쪽으로 전달하는 자리올림을 계산하는 문제이다.

NoteBase

선지별 해설

① O
- 오른쪽부터 계산하면 $A_0 = 1$과 1이 들어가므로 $B_0 = P = 1 \oplus 1 = 0$, 자리올림 O=1 1=1이 생긴다.
- 다음 모듈은 $A_1 = 1$, 자리올림 1이므로 $B_1 = 1 \oplus 1 = 0$, 다시 자리올림 1이다.
- 세 번째 모듈은 $A_2 = 0$, 자리올림 1이므로 $B_2 = 0 \oplus 1 = 1$, 자리올림은 0이 된다.
- 마지막 모듈은 $A_3 = 1$, 자리올림 0이므로 $B_3 = 1 \oplus 0 = 1$, 최종 자리올림 B_4=1 0=0이다. 따라서 $B_4B_3B_2B_1B_0 = (01100)_2$이다.

② X
- $(01110)_2$가 되려면 B_1이 1이어야 하는데, 두 번째 모듈의 입력은 1,1이므로 XOR 출력은 0이다.
- 틀린 표현은 결과를 $(01110)_2$로 본 부분이다. 올바르게는 $(01100)_2$이다.

③ X
- $(10101)_2$는 $B_0 = 1$, $B_4 = 1$로 본 값인데, 오른쪽 끝 모듈의 $1 \oplus 1$은 0이고 마지막 모듈의 1 0도 0이다.
- 틀린 표현은 첫 비트와 최종 자리올림을 1로 본 부분이다. 올바르게는 $(01100)_2$이다.

④ X
- $(11111)_2$는 모든 출력이 1이라고 본 값이다. 그러나 B_0와 B_1은 각각 $1 \oplus 1 = 0$이라서 0이고, B_4도 최종 자리올림이 없어 0이다.
- 틀린 표현은 전 비트가 1이라고 본 부분이다. 올바르게는 $(01100)_2$이다.

오답 포인트

모듈의 두 출력 중 어느 것이 합이고 어느 것이 자리올림인지 바꾸어 읽으면 오답이 나온다. 이 그림에서는 O가 AND 출력이라 자리올림이고, P가 XOR 출력이라 합이다. 특히 오른쪽 끝의 1이 먼저 A_0와 결합한다는 점을 놓치면 ②, ③ 같은 값으로 잘못 간다.

다시 나오면 이렇게 판단

회로형 이진 연산 문제는 먼저 각 모듈의 논리식부터 정리한 뒤, 최하위 비트에서 시작해 합 출력과 자리올림 출력을 한 단계씩 왼쪽으로 넘겨 계산하면 된다.

암기 포인트

반가산기는 합이 XOR, 자리올림이 AND이며, 증분기는 맨 오른쪽에 1을 넣어 +1을 만든다.

NoteBase