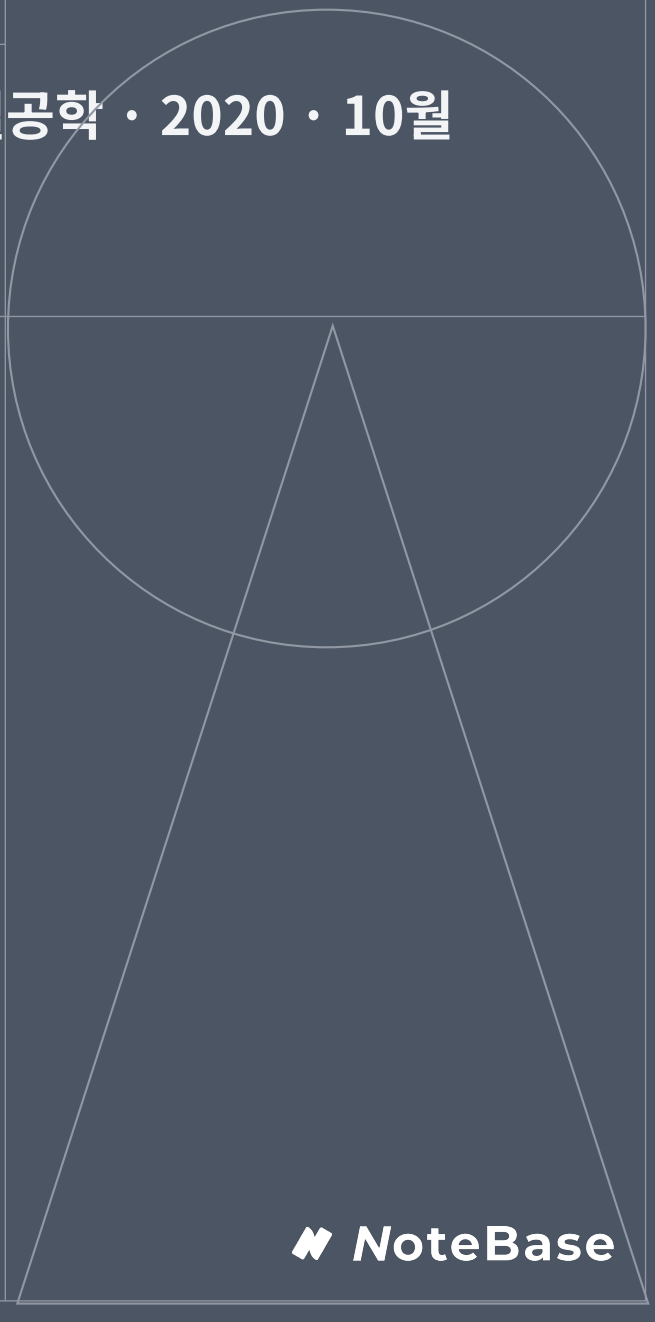
기출OX 해설집

공무원/서울시/7급 · 디지털공학 · 2020 · 10월



NoteBase

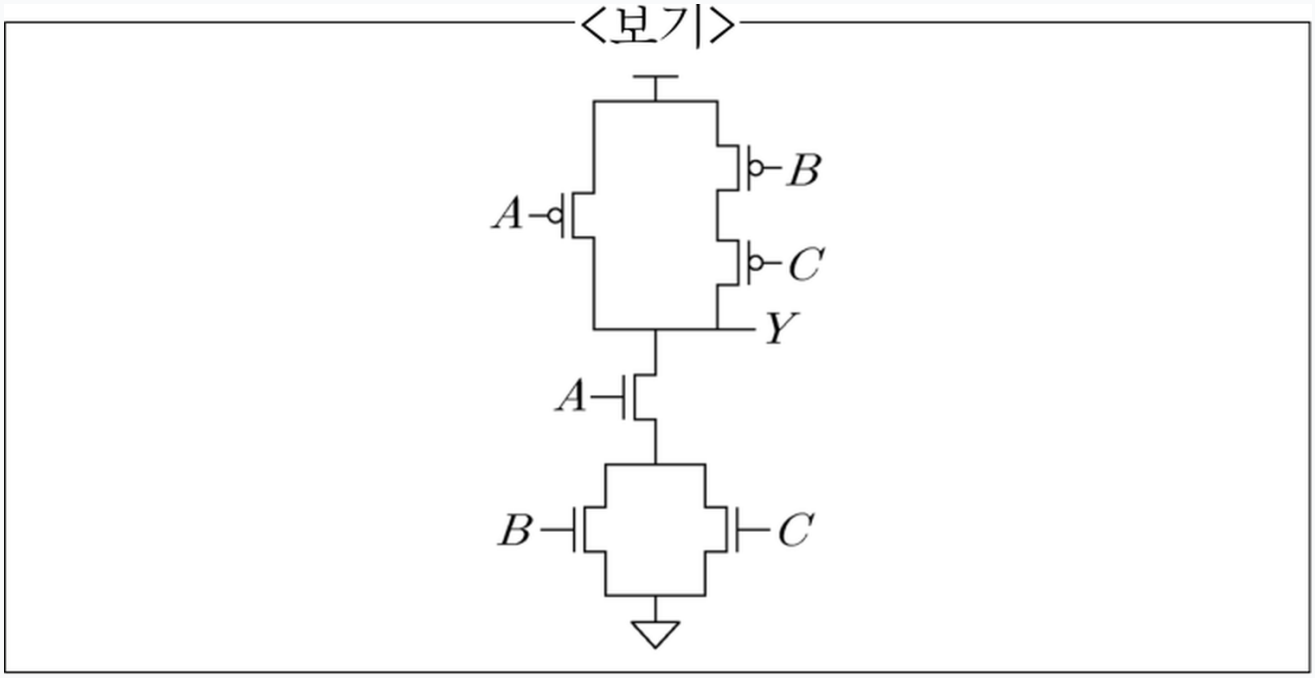
기출OX 해설집

공무원/서울시/7급 · 디지털공학 · 2020 · 10월

공무원/서울시/7급 · 디지털공학 · 2020 · 10월

문 1. <보기>와 같은 CMOS(complementary metal-oxide-semiconductor) 회로에 해당하는 부울 식(Boolean equation)은?

<보기>



① $Y = A + BC$

② $Y = A(B + C)$

③ $Y = \overline{A + BC}$

④ $Y = \overline{A(B + C)}$

정답 4번

출제 포인트

이 문제는 CMOS 회로의 풀다운 네트워크와 풀업 네트워크의 직렬·병렬 관계를 읽어 출력식 Y를 구하는 문제이다.

NoteBase

선지별 해설

① X: 1번은 $Y = A + BC$라고 보았지만, 아래 NMOS 풀다운 네트워크는 A가 켜지고 B 또는 C 중 하나가 켜질 때 접지로 연결된다. 따라서 접지 조건은 $A(B + C)$이고 출력은 그 보수이므로 $Y = A + BC$가 아니다. 틀린 표현은 '보수 없이 그대로 출력식으로 둔 점'이며, 올바르게는 $Y = \overline{A(B + C)}$이다.
② X: 2번은 풀다운 조건 $A(B + C)$ 자체를 출력식으로 적은 것이다. CMOS 출력 Y는 풀다운이 켜질 때 0이 되므로 출력식은 접지 조건의 보수여야 한다. 틀린 표현은 '$Y = A(B + C)$'이며, 올바르게는 $Y = \overline{A(B + C)}$이다.
③ X: 3번은 $A + BC$의 보수로 적었는데, 회로의 직렬·병렬 구조와 맞지 않는다. 아래쪽 NMOS는 A가 직렬이고 B, C가 병렬이므로 $A(B + C)$가 되어야 한다. 틀린 표현은 '$A + BC$'이며, 올바르게는 $A(B + C)$이므로 전체 출력식은 $Y = \overline{A(B + C)}$이다.
④ O: 아래 NMOS 풀다운 네트워크는 A 직렬과 B, C 병렬이므로 도통 조건이 $A(B + C)$이다. CMOS 출력은 이 조건에서 0이 되므로 최종 출력식은 $Y = \overline{A(B + C)}$이다. 위쪽 PMOS 풀업 네트워크도 A 병렬과 B, C 직렬의 듀얼 구조이므로 같은 결론이 나온다.

오답 포인트

NMOS 풀다운이 만드는 식을 그대로 Y로 적는 실수가 가장 흔하다. CMOS에서는 먼저 아래 NMOS가 언제 접지로 연결되는지를 식으로 쓰고, 마지막에 반드시 보수를 취해야 한다. 또한 직렬은 AND, 병렬은 OR로 읽되 위 PMOS와 아래 NMOS의 위치를 섞어 읽지 않아야 한다.

다시 나오면 이렇게 판단

먼저 아래 NMOS 풀다운 네트워크만 보고 직렬은 AND, 병렬은 OR로 접지 조건을 만든다. 그다음 CMOS 출력은 그 접지 조건의 보수라는 순서로 판단하면 된다.

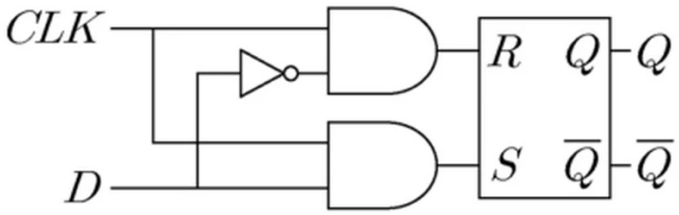
암기 포인트

CMOS는 '아래 NMOS 식을 먼저 쓰고 마지막에 전체 보수'로 정리된다.

NoteBase

문 2. <보기 1>에서 S-R 래치(latch)에 저장된 값(Q)이 1이라 가정하고, CLK 및 D 신호가 <보기 2>와 같이 주어질 때 각 경우별 $\overline{Q}$값을 옳게 짝지은 것은?

<보기 1>



<보기 2>

CLK	D	$\overline{Q}$
0	0	(가)
0	1	(나)
1	0	(다)
1	1	(라)

① (가): 0, (나): 0, (다): 0, (라): 1

② (가): 0, (나): 0, (다): 1, (라): 0

③ (가): 0, (나): 1, (다): 0, (라): 1

④ (가): 0, (나): 1, (다): 1, (라): 0

정답 2번

출제 포인트

이 문제는 게이트드 S-R 래치에서 $R = CLK \cdot \overline{D}$, $S = CLK \cdot D$를 먼저 구한 뒤, 초기값 $Q = 1$일 때 hold, reset, set 동작에 따라 $\overline{Q}$를 판단하는 문제이다.

NoteBase

선지별 해설

① X

이 선택지는 배열을 0, 0, 0, 1로 둔 점이 틀리다. $CLK = 0$인 (가), (나)에서는 $R = S = 0$이므로 초기 저장값 $Q = 1$이 그대로 유지되어 $\overline{Q} = 0$이다. 그러나 $CLK = 1, D = 0$인 (다)에서는 $R = 1, S = 0$이 되어 reset되므로 $Q = 0, \overline{Q} = 1$이어야 한다. $CLK = 1, D = 1$인 (라)에서는 $R = 0, S = 1$이 되어 set되므로 $Q = 1, \overline{Q} = 0$이어야 한다. 틀린 배열 → 올바르게는 0, 0, 1, 0이다.

② O

$CLK = 0$이면 D값과 무관하게 $R = S = 0$이므로 래치는 이전 상태를 유지한다. 초기값이 $Q = 1$이므로 (가), (나) 모두 $\overline{Q} = 0$이다. $CLK = 1, D = 0$이면 $R = 1, S = 0$이라 reset되어 (다)=1이고, $CLK = 1, D = 1$이면 $R = 0, S = 1$이라 set되어 (라)=0이다. 따라서 0, 0, 1, 0이 맞다.

③ X

이 선택지는 배열을 0, 1, 0, 1로 둔 점이 틀리다. 특히 $CLK = 0$인 (나)를 1로 본 것이 잘못이다. $CLK = 0$에서는 $D = 1$이어도 래치가 열리지 않으므로 이전값 $Q = 1$이 유지되어 $\overline{Q} = 0$이다. 또 $CLK = 1, D = 0$인 (다)는 reset이므로 0이 아니라 1이어야 한다. 틀린 배열 → 올바르게는 0, 0, 1, 0이다.

④ X

이 선택지는 배열을 0, 1, 1, 0로 둔 점이 틀리다. (다), (라)는 각각 reset과 set 결과를 맞게 썼지만, $CLK = 0, D = 1$인 (나)를 1로 둔 점이 잘못이다. $CLK = 0$이면 입력이 차단되어 이전값 $Q = 1$이 유지되므로 $\overline{Q} = 0$이어야 한다. 틀린 배열 → 올바르게는 0, 0, 1, 0이다.

오답 포인트

가장 흔한 실수는 $CLK = 0$일 때도 D가 직접 출력에 반영된다고 보는 점이다. 이 회로는 $CLK = 0$이면 두 게이트 출력이 모두 0이어서 래치가 이전 상태를 유지한다. 또 $\overline{Q}$를 묻는데 Q값만 계산하고 보수로 바꾸지 않는 실수도 자주 나온다.

다시 나오면 이렇게 판단

먼저 위쪽 경로와 아래쪽 경로를 읽어 $R = CLK \cdot \overline{D}, S = CLK \cdot D$를 만든다. 다음으로 $CLK = 0$이면 hold, $CLK = 1$이면 $D = 0 \rightarrow reset, D = 1 \rightarrow set$으로 판정한 뒤 마지막에 Q가 아니라 $\overline{Q}$를 적으면 된다.

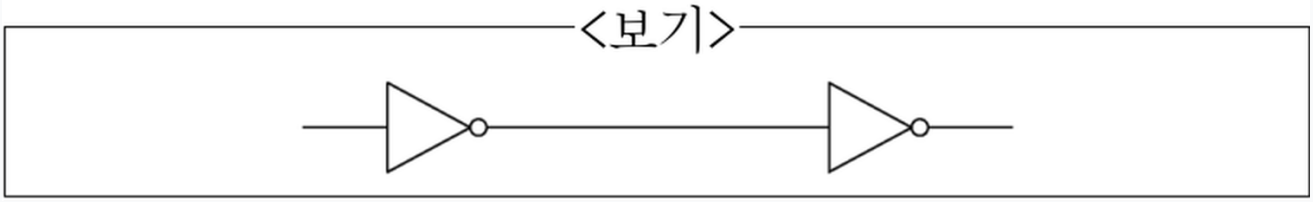
암기 포인트

게이트드 D 래치는 $CLK = 0$이면 유지, $CLK = 1$이면 $Q = D$가 되고, 따라서 $\overline{Q} = \overline{D}$가 된다.

NoteBase

문 3. 동일한 전기적 특성을 갖는 NOT 게이트(gate) 두 개를 <보기>와 같이 연결하였을 때 '논리 0' 신호에 대한 잡음 여유(noise margin)와 '논리 1' 신호에 대한 잡음 여유를 순서대로 바르게 나열한 것은? (단, NOT 게이트는 0~0.3[V] 범위의 출력 전압으로 '논리 0'을 표현하고 3.8~5[V]의 출력 전압으로 '논리 1'을 표현하며, 0~1.3[V]의 입력 전압을 '논리 0'으로 인식하고 3.1~5[V]의 입력 전압을 '논리 1'로 인식한다고 가정한다.)

<보기>



① 1[V], 0.7[V]

② 2.5[V], 1.8[V]

③ 2.8[V], 2.5[V]

④ 3.5[V], 1.8[V]

정답 1번

출제 포인트

이 문제는 동일한 NOT 게이트를 직렬 연결했을 때의 잡음 여유를 구하는 문제이다. 저레벨 잡음 여유와 고레벨 잡음 여유는 같은 논리 레벨에서 앞단 출력의 최악값과 뒷단 입력의 허용 한계값을 비교해 계산한다.

조건 정리

- $V_{OL(\max)} = 0.3V$
- $V_{OH(\min)} = 3.8V$
- $V_{IL(\max)} = 1.3V$
- $V_{IH(\min)} = 3.1V$
- 두 게이트가 동일하므로 첫 번째 게이트의 출력 한계와 두 번째 게이트의 입력 한계를 같은 논리 레벨끼리 비교한다.

사용할 공식

$NML = V_{IL(\max)} - V_{OL(\max)}$

$NMH = V_{OH(\min)} - V_{IH(\min)}$

이 공식을 쓰는 이유는 잡음 여유가 송신 게이트의 출력 보장 범위와 수신 게이트의 입력 판정 범위 사이의 안전 여유를 뜻하기 때문이다.

NoteBase

계산 과정

1. 논리 0에 대한 잡음 여유는 $NML = 1.3 - 0.3 = 1.0V$이다.
2. 논리 1에 대한 잡음 여유는 $NMH = 3.8 - 3.1 = 0.7V$이다.
3. 따라서 순서는 $1.0V$, $0.7V$이다.

선지별 해설

① O
- 계산 결과가 $1.0V$, $0.7V$이므로 선지와 정확히 일치한다.

② X
- $2.5V$와 $1.8V$는 같은 논리 레벨의 입출력 한계를 비교한 값이 아니라 서로 다른 경계값을 잘못 조합한 값이다.
- 틀린 표현은 "논리 0 잡음 여유 2.5[V], 논리 1 잡음 여유 1.8[V]"이고, 올바르게는 "논리 0 잡음 여유 1[V], 논리 1 잡음 여유 0.7[V]"이다.

③ X
- $2.8V$와 $2.5V$는 $3.1 - 0.3$, $3.8 - 1.3$처럼 낮은 레벨과 높은 레벨의 경계를 교차해서 뺀 값으로 잡음 여유 정의에 맞지 않는다.
- 틀린 표현은 "논리 0 잡음 여유 2.8[V], 논리 1 잡음 여유 2.5[V]"이고, 올바르게는 "논리 0 잡음 여유 1[V], 논리 1 잡음 여유 0.7[V]"이다.

④ X
- $3.5V$와 $1.8V$도 $3.8 - 0.3$, $3.1 - 1.3$처럼 출력 범위 폭이나 입력 공백 구간을 잘못 읽은 값이라 잡음 여유가 아니다.
- 틀린 표현은 "논리 0 잡음 여유 3.5[V], 논리 1 잡음 여유 1.8[V]"이고, 올바르게는 "논리 0 잡음 여유 1[V], 논리 1 잡음 여유 0.7[V]"이다.

오답 포인트

잡음 여유를 구할 때는 회로에 NOT 게이트가 두 개 있다는 사실 때문에 값을 두 번 계산하거나 높은 레벨 경계와 낮은 레벨 경계를 섞어 빼기 쉽다. 이 문제의 함정은 같은 논리 레벨끼리 비교해야 하는데 ③, ④처럼 서로 다른 경계를 교차해서 빼도록 유도한 데 있다.

다시 나오면 이렇게 판단

직렬 연결 회로의 잡음 여유는 앞단 출력과 뒷단 입력을 같은 논리 레벨끼리 비교해 판단한다. 저레벨은 $V_{IL(\max)} - V_{OL(\max)}$, 고레벨은 $V_{OH(\min)} - V_{IH(\min)}$를 바로 계산하면 된다.

NoteBase

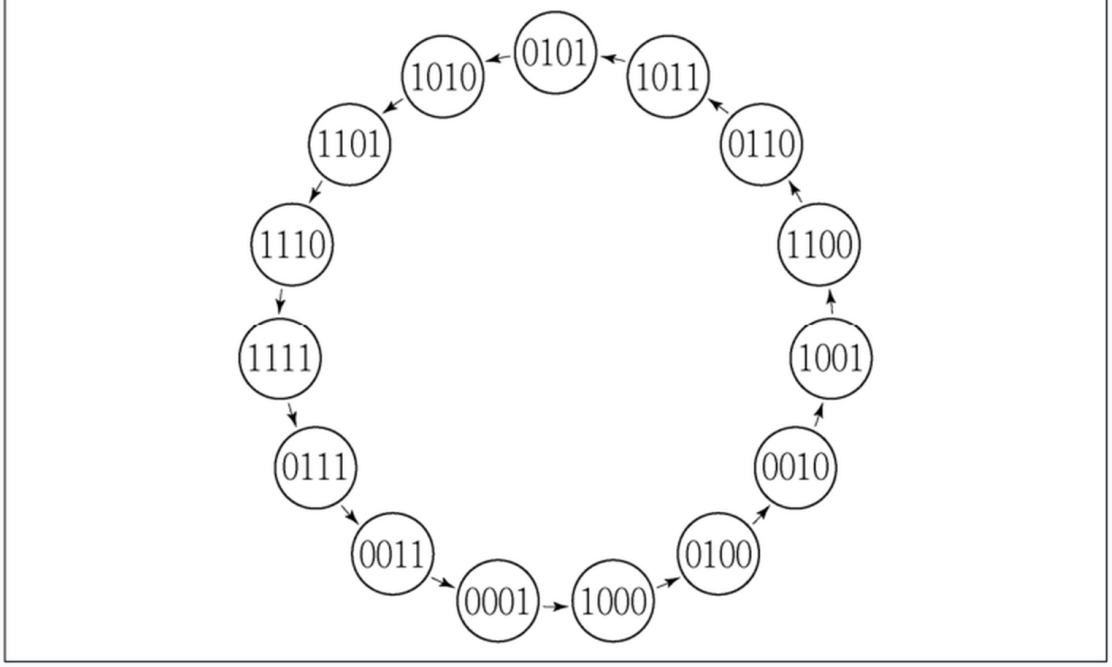
암기 포인트

잡음 여유는 같은 레벨끼리 뺀다. 저레벨은 입력 최대에서 출력 최대를 빼고, 고레벨은 출력 최소에서 입력 최소를 뺀 값이다.

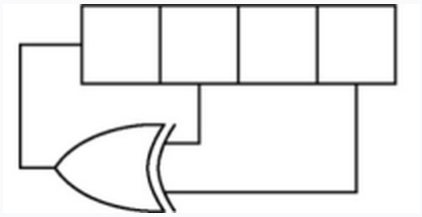
NoteBase

문 4. <보기>의 상태 천이도(state transition diagram)를 구현할 수 있는 회로는? (단, 선택지의 msb □ □ lsb 기호는 4-비트 우측 쉬프트 레지스터(shift-right register)로서 병렬 적재(parallel load) 기능이 있어 초기값을 임의의 값으로 설정할 수 있다고 가정한다.)

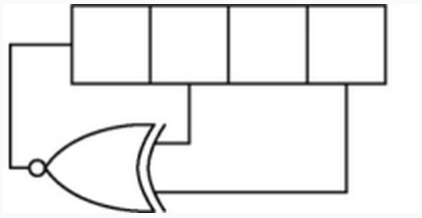
<보기>



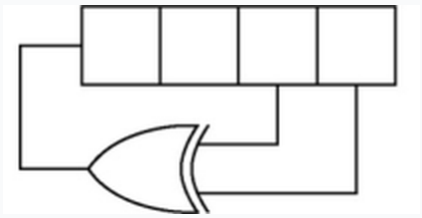
①



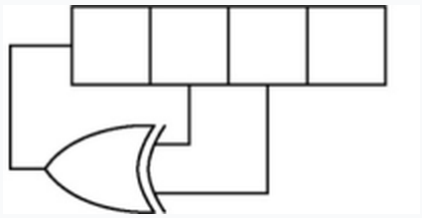
②



③



④



NoteBase

정답 3번

출제 포인트

이 문제는 4비트 우측 시프트 레지스터의 피드백 탭 위치와 XOR/XNOR 여부를 상태 전이도와 대조하여 최대길이 수열을 만드는 회로를 찾는 문제이다.

선지별 해설

① X: 1번 회로는 q_2와 q_0를 XOR하여 되먹임하므로 0010에서 피드백이 0이 된다. 따라서 다음 상태가 0001이 되어, 제시된 상태도의 0010→1001 전이와 맞지 않는다. 틀린 표현은 이 회로가 제시 상태열을 구현한다는 점이며, 올바르게는 0010에서 1001이 나오도록 $q_1 \oplus q_0$가 msb로 들어가야 한다.
② X: 2번 회로는 1번과 같은 탭을 쓰되 입력 반전이 있어 XNOR 계열로 동작한다. 0001에서 $q_2 = 0$, $q_0 = 1$이므로 피드백이 0이 되고 다음 상태가 0000이 되어, 상태도의 0001→1000과 바로 어긋난다. 틀린 표현은 XNOR 피드백으로 같은 수열이 나온다는 점이며, 올바르게는 이 상태도는 XOR 피드백 수열이다.
③ O: 3번 회로는 오른쪽 두 비트 q_1과 q_0를 XOR하여 msb로 되먹임하는 구조이다. 그래서 0001→1000, 0010→1001, 1111→0111이 모두 그림과 일치하고, 0000을 제외한 15개 상태를 순환하는 최대길이 수열이 된다.
④ X: 4번 회로는 q_2와 q_1를 XOR하여 되먹임하므로 0001에서 피드백이 0이 된다. 따라서 다음 상태가 0000이 되어 상태도의 첫 전이 0001→1000부터 맞지 않는다. 틀린 표현은 이 탭 배치가 제시 상태열을 만든다는 점이며, 올바르게는 lsb를 포함한 오른쪽 두 비트 q_1, q_0를 XOR해야 한다.

오답 포인트

도식이 비슷해 보여도 우측 시프트에서는 어떤 비트가 새 msb로 들어가는지만 보면 빠르게 갈린다. ①과 ③은 lsb를 포함한다는 점은 같지만 다른 탭 하나가 q_2인지 q_1인지에 따라 0010 다음 상태가 달라진다. ②는 XOR 모양에 입력 반전 버블이 붙어 XNOR로 바뀐다는 점이 함정이다.

다시 나오면 이렇게 판단

상태도에서 초반 전이 두 개만 잡아 피드백식을 역으로 확인하면 된다. 우측 시프트이므로 현재 상태에서 선택된 탭 비트를 XOR해 새 msb를 만들고, 0001→1000과 0010→1001을 동시에 만족하는 회로만 남기면 된다.

암기 포인트

4비트 최대길이 LFSR은 0000을 제외한 15개 상태를 순환한다. 이 문항의 핵심 되먹임은 오른쪽 두 비트의 XOR, 즉 $q_1 \oplus q_0$이다.

NoteBase

문 5. 아날로그 신호에 대비한 디지털 신호의 설명으로 가장 옳지 않은 것은?

① 데이터를 정확하게 계산할 수 있고, 암호화 및 보안이 용이하다.

② 재생(regeneration)이 용이하고, 오류의 검출 및 정정이 가능하다.

③ 시스템의 구성이 용이하고, 신뢰도, 융통성을 가진다.

④ 한번 잡음과 일그러짐이 존재하면 제거하기가 매우 어렵다.

정답 4번

출제 포인트

이 문제는 아날로그 신호와 대비되는 디지털 신호의 핵심 특성, 특히 잡음 내성과 재생 가능성을 구별하는 문제이다.

핵심 개념

아날로그 신호는 연속적인 크기로 정보를 표현하므로 잡음이 섞이면 원신호와 잡음을 완전히 분리하기 어렵다. 디지털 신호는 이산적인 0과 1로 정보를 표현하므로 일정 임계값으로 판정하고 재생하여 잡음의 영향을 줄일 수 있다.

선지별 해설

① O. 디지털 신호는 이산값으로 처리되므로 논리 연산과 수치 계산에 유리하고, 데이터 암호화·복호화 같은 보안 처리도 구현하기 쉽다. ② O. 디지털 신호는 중간에서 0과 1을 다시 판정해 재생할 수 있으므로 regeneration이 용이하고, 부가 비트를 이용한 오류 검출·정정도 가능하다. ③ O. 디지털 회로는 표준 논리소자를 조합해 시스템을 구성하기 쉬우며, 잡음에 비교적 강하고 프로그램적 변경도 가능해 신뢰도와 융통성이 높다. ④ X. 틀린 부분은 '잡음과 일그러짐이 존재하면 제거하기가 매우 어렵다'이다. 이것은 아날로그 신호의 단점에 가깝다. 디지털 신호는 잡음이 일부 섞여도 임계값 판정과 재생 과정을 통해 원래의 0과 1 형태로 복원하기가 상대적으로 쉽다.

오답 포인트

④는 잡음에 약한 아날로그 신호의 성질을 디지털 신호의 성질로 바꾸어 놓은 함정이다. 발문이 '디지털 신호의 설명'인지, 그리고 '옳지 않은 것'을 묻는지까지 함께 확인해야 정답이 뒤집히지 않는다.

다시 나오면 이렇게 판단

디지털 신호 선지가 나오면 0과 1의 이산 표현, 재생 가능, 오류 검출·정정 가능, 잡음에 비교적 강함을 먼저 연결한다. 반대로 잡음이 누적되면 복원이 어렵다는 설명은 아날로그 신호 쪽 특징으로 판단한다.

NoteBase

암기 포인트

디지털은 재생과 오류정정에 강하고, 아날로그는 잡음 누적에 약하다.

NoteBase

문 6. 십진수 999 이하의 자연수 또는 0을 2진수와 BCD 코드로 표현할 때 각각 요구되는 최소 비트수를 옳게 짝지은 것은?

① 2진수의 경우: 10, BCD코드의 경우: 16

② 2진수의 경우: 16, BCD코드의 경우: 12

③ 2진수의 경우: 10, BCD코드의 경우: 12

④ 2진수의 경우: 16, BCD코드의 경우: 10

정답 3번

출제 포인트

이 문제는 0부터 999까지의 표현 범위에 필요한 순수 2진수 비트수와 BCD의 자릿수별 비트수를 구별하는 문제이다.

조건 정리

주어진 범위는 0부터 999까지이므로 총 1000개의 값을 표현해야 한다.
구해야 할 값은 2진수 표현의 최소 비트수와 BCD 코드 표현의 최소 비트수이다.
BCD는 십진수 한 자리마다 4비트를 고정으로 사용한다.

사용할 공식

2진수 최소 비트수는 2^n이 표현 가능한 경우의 수이므로 $2^n \geq 1000$을 만족하는 최소 n으로 구한다.
BCD 최소 비트수는 십진 자릿수 $\times 4$비트로 구한다.
이 문제는 하나의 수 전체를 순수 2진수로 바꿀 때와 각 십진 자리를 따로 부호화하는 BCD를 비교하는 상황이다.

계산 과정

0부터 999까지는 총 1000개 값이다.
$2^9 = 512$이므로 부족하고, $2^{10} = 1024$이므로 2진수는 최소 10비트가 필요하다.
999는 세 자리 십진수이므로 BCD는 각 자리당 4비트씩 사용하여 $3 \times 4 = 12$비트가 필요하다.

NoteBase

선지 판단

① X: 2진수 10비트는 맞지만 BCD 16비트는 틀리다. 999는 세 자리 수이므로 BCD는 12비트가 맞다. 틀린 표현은 "BCD코드의 경우 16"이고, 올바르게는 "BCD코드의 경우 12"이다.
② X: BCD 12비트는 맞지만 2진수 16비트는 틀리다. 0부터 999까지는 $2^{10} = 1024$로 표현 가능하므로 10비트면 충분하다. 틀린 표현은 "2진수의 경우 16"이고, 올바르게는 "2진수의 경우 10"이다.
③ O: 2진수는 10비트, BCD는 12비트이므로 두 값이 모두 맞다.
④ X: 2진수 16비트도 틀리고 BCD 10비트도 틀리다. 순수 2진수는 10비트면 충분하고, 표준 BCD는 세 자리에 12비트가 필요하다. 올바르게는 "2진수의 경우 10, BCD코드의 경우 12"이다.

오답 포인트

BCD를 순수 2진수처럼 압축해서 생각하면 틀린다. BCD는 십진수 한 자리마다 4비트를 고정 사용하므로 세 자리 수 범위는 항상 12비트가 된다.

다시 나오면 이렇게 판단

표현 범위 문제가 나오면 먼저 값의 개수를 세고 순수 2진수는 2^n과 비교한다. BCD가 나오면 수 전체의 크기를 보지 말고 십진 자릿수에 4비트를 곱해 판단한다.

암기 포인트

순수 2진수는 2^n으로 경우의 수를 세고, BCD는 자릿수 ×4비트로 계산한다.

NoteBase

문 7. 하나의 착오만을 검출하고 정정할 수 있는 해밍 부호(Hamming code)를 128 비트의 데이터에 적용할 경우 필요한 패리티 비트의 개수는?

① 8

② 16

③ 40

④ 64

정답 1번

출제 포인트

이 문제는 해밍 부호에서 단일 오류를 검출·정정할 수 있도록 필요한 패리티 비트 수를 조건식으로 구하는 문제이다.

조건 정리

주어진 값: 데이터 비트 수는 128비트이다.
구해야 할 값: 필요한 패리티 비트 수이다.
주의할 조건: 해밍 부호는 오류가 없는 경우까지 포함해 모든 상태를 구별할 수 있어야 한다.

사용할 공식

패리티 비트 수를 r, 데이터 비트 수를 m이라 하면 $2^r \geq m + r + 1$을 만족해야 한다.

이 공식을 쓰는 이유: 해밍 부호는 $m + r$개의 각 비트 위치에서 발생할 수 있는 1비트 오류와 오류가 없는 상태 1개를 모두 구별해야 하므로 총 $m + r + 1$개의 상태를 표현할 수 있어야 한다.

계산 과정

1. $m = 128$을 대입하면 $2^r \geq 128 + r + 1$이 된다.
2. $r = 7$이면 $2^7 = 128$, 우변은 $128 + 7 + 1 = 136$이므로 조건을 만족하지 못한다.
3. $r = 8$이면 $2^8 = 256$, 우변은 $128 + 8 + 1 = 137$이므로 조건을 만족한다.
4. 조건을 처음 만족하는 최소값은 8이다.

NoteBase

선지별 해설

① O: 필요한 패리티 비트 수는 최소 8개이다. $r = 8$에서 처음으로 $2^r \geq m + r + 1$ 조건이 성립한다.
② X: 16개는 단일 오류 정정을 위해 필요한 최소 개수가 아니다. 틀린 표현은 "필요한 패리티 비트 수"이고, 올바르게는 "8개면 충분하다"이다.
③ X: 40개는 조건식으로 계산한 최소 패리티 비트 수와 전혀 맞지 않는다. 틀린 표현은 "40"이고, 올바르게는 "8"이다.
④ X: 64개는 데이터 비트의 절반을 패리티 비트로 둔 값에 가깝지만 해밍 부호의 필요 조건과는 무관하다. 틀린 표현은 "64"이고, 올바르게는 "8"이다.

오답 포인트

많이 틀리는 지점은 패리티 비트 개수를 단순히 데이터 비트의 절반이나 큰 2의 거듭제곱으로 감으로 고르는 데 있다. 이 문항은 반드시 $2^r \geq m + r + 1$을 세워 최소 r을 찾는지가 핵심이다.

다시 나오면 이렇게 판단

해밍 부호 문항은 먼저 $2^r \geq m + r + 1$ 조건을 세우고, 작은 r부터 대입해 처음 성립하는 값을 정답으로 판단한다.

암기 포인트

해밍 부호 패리티 비트는 $2^r \geq m + r + 1$을 만족하는 최소 r이다.

NoteBase

문 8. <보기>의 부울 식 Y를 최소항 또는 최대항으로 나타낸 것으로 가장 옳은 것은?

<보기>

$Y = AB + \overline{B}C$

① $Y = \Sigma(1, 3, 5, 7)$

② $Y = \Sigma(0, 2, 3, 4)$

③ $Y = \Pi(1, 3, 5, 7)$

④ $Y = \Pi(0, 2, 3, 4)$

정답 4번

출제 포인트

이 문제는 부울식 $Y = AB + \overline{B}C$의 진리값을 구해 표준 최소항 Σ와 표준 최대항 Π으로 바꾸는 문제이다.

조건 정리

주어진 식은 $Y = AB + \overline{B}C$이다.
변수는 A, B, C의 3개이므로 인덱스는 0부터 7까지이다.
최소항은 $Y = 1$인 행의 번호를 쓰고, 최대항은 $Y = 0$인 행의 번호를 쓴다.

사용할 공식

표준형 변환의 기준은 $Y = 1 \rightarrow \Sigma(m), Y = 0 \rightarrow \Pi(M)$이다.
이 공식을 쓰는 이유는 보기들이 모두 진리표의 인덱스를 이용한 표준 최소항 또는 최대항 형태이기 때문이다.

계산 과정

1. $B = 0$이면 $AB = 0, \overline{B}C = C$이므로 $Y = C$이다. 따라서 $(A, B, C) = (0, 0, 1), (1, 0, 1)$에서만 $Y = 1$이고 인덱스는 1, 5이다.
2. $B = 1$이면 $\overline{B}C = 0, AB = A$이므로 $Y = A$이다. 따라서 $(1, 1, 0), (1, 1, 1)$에서 $Y = 1$이고 인덱스는 6, 7이다.
3. 따라서 $Y = 1$인 최소항은 $\Sigma(1, 5, 6, 7)$이고, $Y = 0$인 최대항은 $\Pi(0, 2, 3, 4)$이다.

NoteBase

선지별 해설

① X: 최소항은 $Y = 1$인 인덱스를 적어야 하는데 6이 빠지고 3이 들어가 있다. 3번 인덱스 (0, 1, 1)에서는 $Y = 0$이므로 틀리다. 틀린 표현은 Σ(1, 3, 5, 7)이고, 올바르게는 Σ(1, 5, 6, 7)이다.
② X: 0, 2, 3, 4는 $Y = 0$인 인덱스이므로 최소항의 합이 아니라 최대항의 곱에 해당한다. 틀린 표현은 Σ(0, 2, 3, 4)이고, 올바르게는 Π(0, 2, 3, 4)이다.
③ X: 최대항은 $Y = 0$인 인덱스를 적어야 하는데 1, 5, 7은 모두 $Y = 1$인 행이다. 틀린 표현은 Π(1, 3, 5, 7)이고, 올바르게는 Π(0, 2, 3, 4)이다.
④ O: $Y = 0$인 인덱스가 정확히 0, 2, 3, 4이므로 표준 최대항 표현은 Π(0, 2, 3, 4)이다.

오답 포인트

최소항과 최대항의 기준을 뒤집으면 ②처럼 0이 되는 행을 Σ로 적는 실수를 하게 된다. 또 $B = 1$일 때 $Y = A$, $B = 0$일 때 $Y = C$로 나뉜다는 분기 판단을 놓치면 ①처럼 6 대신 3을 넣는 오류가 나온다.

다시 나오면 이렇게 판단

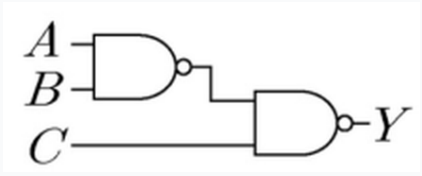
부울식을 표준형으로 바꾸는 문제는 먼저 한 변수 조건으로 식을 단순화한 뒤 $Y = 1$인 행과 $Y = 0$인 행을 나누어 적는다. 그다음 Σ는 1의 행, Π는 0의 행이라는 기준으로 선지를 바로 대조한다.

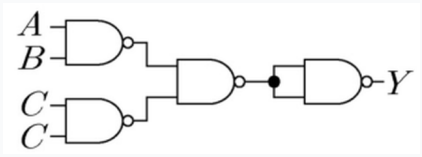
암기 포인트

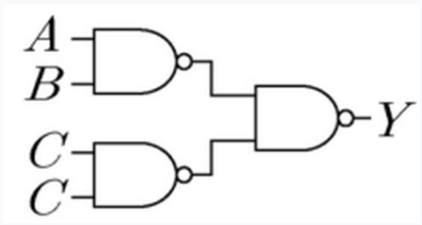
Σ는 1의 주소, Π는 0의 주소라는 기준만 고정하면 최소항과 최대항을 혼동하지 않는다.

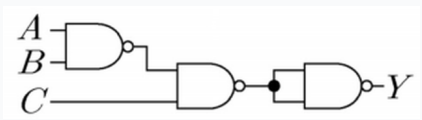
NoteBase

문 9. 논리식 $Y = AB + C$를 NAND 게이트만을 사용하여 구현한 것으로 가장 옳은 것은?

① 

② 

③ 

④ 

정답 3번

출제 포인트

이 문제는 NAND 게이트만으로 $Y = AB + C$를 만들기 위해 드모르간 법칙을 적용하고, 각 회로가 실제로 어떤 논리식으로 변환되는지를 판별하는 문제이다. $Y = AB + C = ((AB)' \cdot C')'$이므로, $(AB)'$와 C'를 만들어 마지막 NAND로 묶어야 한다.

NoteBase

선지별 해설

① X
첫 번째 NAND의 출력은 $(AB)'$이다. 이를 아무 반전 없이 그대로 들어온 C와 다시 NAND하면 $Y = ((AB)' \cdot C)' = AB + C'$가 된다. 목표 식 $AB + C$와 달리 C를 반전하지 않아 $AB + C'$를 만든 회로이므로 틀리다.

② X
아래쪽 NAND에 C와 C를 넣으면 출력은 C'가 된다. 가운데 NAND의 출력은 $((AB)' \cdot C')' = AB + C$로 목표 식과 같지만, 마지막 NAND를 두 입력 묶음으로 인버터처럼 사용하므로 최종 출력은 $(AB + C)'$가 된다. 가운데 단계까지는 맞아도 마지막 인버터 때문에 전체가 다시 반전되어 틀리다.

③ O
첫 번째 NAND의 출력은 $(AB)'$이고, 아래쪽 NAND는 C와 C를 같은 입력으로 받아 인버터처럼 동작하여 C'를 만든다. 마지막 NAND에서 $Y = ((AB)' \cdot C')' = AB + C$가 되어 목표 식과 일치한다.

④ X
첫 번째 NAND의 출력은 $(AB)'$이다. 이것을 반전 없이 그대로 들어온 C와 가운데 NAND로 묶으면 $((AB)' \cdot C)' = AB + C'$가 되고, 마지막 NAND 인버터를 거치면 최종 출력은 $(AB + C')'$가 된다. C를 반전하지 않은 데다 마지막 인버터까지 거쳐 목표 식과 어긋나므로 틀리다.

오답 포인트

NAND의 두 입력을 같은 값으로 묶으면 인버터가 된다는 점과, C를 그대로 넣는지 아니면 C'로 반전해서 넣는지에 따라 결과의 마지막 항이 C와 C'로 갈린다는 점을 놓치기 쉽다. 특히 ②는 가운데 출력이 목표 식 $AB + C$와 같아 보여도 마지막 인버터 때문에 전체 식이 다시 반전된다.

다시 나오면 이렇게 판단

각 NAND를 순서대로 (입력의 곱)'로 바꾸고, 두 입력이 같은 NAND는 인버터로 읽는다. 그다음 중간 노드의 식을 끝까지 추적하여 최종 출력이 $AB + C$인지 확인하면 된다.

암기 포인트

NAND만으로 OR를 만들 때는 필요한 보수를 먼저 만든 뒤 $Y = (X' \cdot Z')' = X + Z$ 형태로 합성한다.

NoteBase

문 10. 순차 논리회로에 대한 설명으로 가장 옳지 않은 것은?

① 클록의 트리거링 에지의 50% 시점 이후, 입력 데이터의 상승 50% 시점까지의 시간을 set-up 시간이라고 한다.

② 전파지연시간이란 입력신호가 가해진 후 출력에 변화를 가져오는 데 걸리는 시간을 의미한다.

③ 클록의 트리거링 에지의 50% 시점 이후에도 데이터 입력이 얼마 동안은 안정화되어 있어야 하는데, 이 시간을 hold 시간이라고 한다.

④ 회로의 전파지연시간이 짧을수록 고속동작이 가능하다.

정답 1번

출제 포인트

이 문제는 순차 논리회로의 타이밍 파라미터인 setup time, hold time, propagation delay의 정의를 정확히 구분하는지를 묻는 문제이다.

핵심 개념

setup time은 클록의 유효 트리거 에지 50% 시점 이전에 입력 데이터가 미리 안정되어 있어야 하는 최소 시간이다.
hold time은 클록의 유효 트리거 에지 50% 시점 이후에도 입력 데이터가 계속 안정되어 있어야 하는 최소 시간이다.
전파지연시간은 입력 변화가 발생한 뒤 출력이 이에 대응하여 변할 때까지 걸리는 시간이다.

선지별 해설

① X. 틀린 부분은 "클록의 트리거링 에지의 50% 시점 이후"이다. setup time은 에지 이후가 아니라 에지 이전에 입력 데이터가 안정되어 있어야 하는 시간이다. 올바르게는 "클록의 트리거링 에지 50% 시점 이전부터 입력 데이터의 상승 50% 시점까지 필요한 안정 시간"이다.
② O. 전파지연시간은 입력 신호가 변한 뒤 출력이 그 변화에 반응하기까지 걸리는 시간을 뜻하므로 설명이 맞다.
③ O. hold time은 클록의 트리거링 에지가 지난 뒤에도 데이터 입력이 일정 시간 유지되어야 하는 조건이므로 설명이 맞다.
④ O. 회로의 전파지연시간이 짧을수록 입력 변화에 대한 출력 응답이 빨라지므로 더 높은 클록 주파수에서 동작할 수 있어 고속동작에 유리하다.

오답 포인트

setup time과 hold time은 둘 다 클록 에지 주변의 안정 시간이라는 점이 같아서 방향을 뒤바꾸기 쉽다. 이 문항은 ①에서 "이후"라는 한 단어로 setup time을 hold time처럼 바꿔 놓은 점이 함정이다.

NoteBase

다시 나오면 이렇게 판단

클록 에지 이전 조건이면 setup time, 클록 에지 이후 유지 조건이면 hold time, 입력 변화 후 출력 응답까지의 시간이면 전파지연시간으로 판단한다.

암기 포인트

setup은 에지 전 준비, hold는 에지 후 유지, propagation delay는 입력 후 출력 지연이다.

NoteBase

문 11. 10진수 56에 대한 3진수 변환을 수행한 결과로 가장 옳은 것은?

① 1222_3

② 1202_3

③ 1212_3

④ 2002_3

정답 4번

출제 포인트

이 문제는 10진수 56을 3진수로 바꿀 때 연속 나눗셈과 자리값 검산을 정확히 할 수 있는지를 묻는 문제이다.

조건 정리

주어진 값: 10진수 56
구해야 할 값: 56의 3진수 표현
주의할 조건: 3진수의 각 자리에는 0, 1, 2만 올 수 있다.

사용할 공식

10진수를 3진수로 바꿀 때는 3으로 계속 나누어 나온 나머지를 아래에서 위로 읽거나,
$56 = a_3 \times 3^3 + a_2 \times 3^2 + a_1 \times 3 + a_0$ 꼴의 자리값 합으로 분해한다.
이 방법을 쓰는 이유는 3진수의 각 자리가 3의 거듭제곱 자리값을 뜻하기 때문이다.

계산 과정

56을 3으로 나누면 몫 18, 나머지 2이다.
18을 3으로 나누면 몫 6, 나머지 0이다.
6을 3으로 나누면 몫 2, 나머지 0이다.
2를 3으로 나누면 몫 0, 나머지 2이다.
나머지를 아래에서 위로 읽으면 2002_3이다.
자리값으로 검산하면 $2 \times 3^3 + 0 \times 3^2 + 0 \times 3 + 2 = 54 + 2 = 56$이므로 맞다.

NoteBase

선지별 해설

① X: 1222_3를 10진수로 바꾸면 $1 \times 27 + 2 \times 9 + 2 \times 3 + 2 = 53$이다. 틀린 표현은 1222_3이며, 올바르게는 2002_3이다.
② X: 1202_3를 10진수로 바꾸면 $1 \times 27 + 2 \times 9 + 0 \times 3 + 2 = 47$이다. 틀린 표현은 1202_3이며, 올바르게는 2002_3이다.
③ X: 1212_3를 10진수로 바꾸면 $1 \times 27 + 2 \times 9 + 1 \times 3 + 2 = 50$이다. 틀린 표현은 1212_3이며, 올바르게는 2002_3이다.
④ O: 2002_3는 $2 \times 27 + 0 \times 9 + 0 \times 3 + 2 = 56$이므로 10진수 56의 3진수 표현이 맞다.

오답 포인트

가장 큰 자리값인 $3^3 = 27$을 먼저 잡지 않고 숫자 모양만 고르면 오답이 된다. 56은 27이 두 번 들어가고 나머지가 2뿐이므로 가운데 두 자리는 0이 되어야 한다는 점이 핵심 함정이다.

다시 나오면 이렇게 판단

진법 변환 문제는 가장 큰 3의 거듭제곱부터 자리를 정하고, 연속 나눗셈의 나머지를 아래에서 위로 읽은 뒤 다시 10진수로 검산하면 된다.

암기 포인트

10진수를 n진수로 바꿀 때는 n으로 계속 나눈 나머지를 아래에서 위로 읽는다.

NoteBase

문 12. 2의 보수(2's complement)로 표현된 4비트 2진수 2개를 더해 4비트 결과를 저장한다고 할 때, 오버플로우(overflow)가 발생하는 경우로 가장 옳은 것은?

① $1010_2 + 0101_2$

② $0111_2 + 0001_2$

③ $1111_2 + 1001_2$

④ $1101_2 + 0010_2$

정답 2번

출제 포인트

이 문제는 4비트 2의 보수 덧셈에서 오버플로우가 생기는 조건을 묻는다.

조건 정리

4비트 2의 보수의 표현 범위는 -8부터 +7까지이다.
오버플로우는 같은 부호의 두 수를 더했는데 결과의 부호가 반대로 바뀌거나 실제 합이 표현 범위를 벗어날 때 발생한다.
부호가 다른 두 수의 덧셈에서는 오버플로우가 발생하지 않는다.

사용할 공식

판단 기준은 같은 부호끼리 더했을 때 결과 부호가 바뀌는가이다.
이 기준을 쓰는 이유는 2의 보수 덧셈에서 오버플로우가 최종 캐리 자체가 아니라 부호 비트의 모순으로 결정되기 때문이다.

계산 과정

① 1010은 -6, 0101은 +5이므로 합은 -1이다.
② 0111은 +7, 0001은 +1이므로 실제 합은 +8이다. +8은 4비트 범위를 벗어나므로 저장 결과 1000이 되어 오버플로우가 발생한다.
③ 1111은 -1, 1001은 -7이므로 합은 -8이다. -8은 표현 가능 범위 안이다.
④ 1101은 -3, 0010은 +2이므로 합은 -1이다.

NoteBase

선지 판단

① X: 1010은 -6, 0101은 +5이므로 부호가 다른 두 수의 덧셈이다. 결과는 -1로 범위 안이어서 오버플로우가 발생하지 않는다. 틀린 해석은 '오버플로우가 발생한다'이고, 올바르게는 '오버플로우가 발생하지 않는다'이다.
② O: 0111은 +7, 0001은 +1이므로 실제 합은 +8이다. 4비트 2의 보수로는 +8을 저장할 수 없어서 결과가 1000으로 뒤집히며 오버플로우가 발생한다.
③ X: 1111은 -1, 1001은 -7이므로 합은 -8이다. 두 음수를 더했지만 결과도 음수이고 -8은 저장 가능한 최솟값이므로 오버플로우가 아니다. 틀린 해석은 '오버플로우가 발생한다'이고, 올바르게는 '정상적으로 -8이 저장된다'이다.
④ X: 1101은 -3, 0010은 +2이므로 부호가 다른 두 수의 덧셈이다. 결과는 -1로 범위 안이어서 오버플로우가 발생하지 않는다. 틀린 해석은 '오버플로우가 발생한다'이고, 올바르게는 '오버플로우 없이 -1이 저장된다'이다.

오답 포인트

최종 캐리가 생기면 무조건 오버플로우라고 보면 틀린다. ③처럼 최상위 캐리가 있어도 실제 결과가 표현 범위 안이면 오버플로우가 아니다. 또한 ①, ④처럼 부호가 다른 두 수의 덧셈은 결과가 범위를 넘지 않는 한 오버플로우가 아니다.

다시 나오면 이렇게 판단

먼저 각 비트열을 10진값으로 바꾸어 부호를 확인한다. 그다음 같은 부호끼리의 덧셈인지 본 뒤 실제 합이 -8부터 +7 범위를 벗어나거나 결과 부호가 뒤집히는지 판단하면 된다.

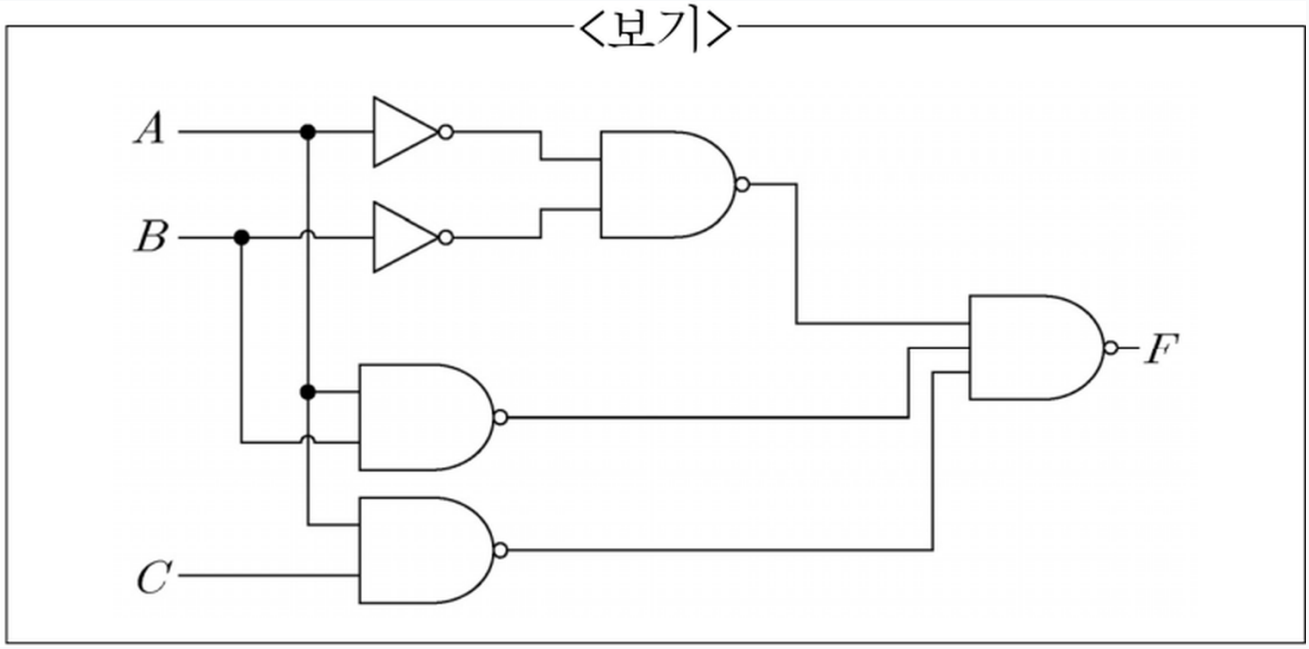
암기 포인트

2의 보수 오버플로우는 같은 부호 덧셈에서 결과 부호 반전으로 판단하고, 최종 캐리만으로는 판단하지 않는다.

NoteBase

문 13. <보기>의 부울 논리 게이트 회로는 3개의 입력 변수 A, B, C와 1개의 출력 변수 F를 갖는다. 이 논리 회로를 부울 논리식으로 표현한 것 중 가장 옳지 않은 것은?

<보기>



① $F = \overline{AB} + AB + AC$

② $F = \overline{AB} + AB\overline{C} + C$

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

④ $F = (A + \overline{B})(\overline{A} + B + C)$

정답 2번

출제 포인트

이 문제는 인버터와 NAND 게이트의 조합을 드모르간 법칙으로 읽어 회로식을 동치식으로 바꿀 수 있는지를 묻는다. 이 회로의 중간 출력은 각각 $A + B, \overline{A} + \overline{B}, \overline{A} + \overline{C}$이다.

NoteBase

선지별 해설

① O. 최종 출력은 $F = \overline{(A+B)(\overline{A}+\overline{B})(\overline{A}+\overline{C})}$이고, 이를 드모르간 법칙으로 풀면 $F = \overline{AB} + AB + AC$가 된다. 따라서 ①은 회로식과 정확히 같다.

② X. $\overline{AB} + AB\overline{C} + C$는 $C = 1$이면 항상 1이 되므로 원래 회로와 다르다. 예를 들어 $A = 0, B = 1, C = 1$이면 이 식은 1이지만 회로식 $\overline{AB} + AB + AC$의 값은 0이다. 틀린 표현은 $AB\overline{C} + C$ 부분이며, 올바르게는 $\overline{AB} + AB + AC$ 또는 그와 동치인 식이 되어야 한다.

③ O. $AC = ABC + A\overline{B}C$이고, 여기서 ABC는 이미 AB에 포함되어 흡수된다. 따라서 $\overline{AB} + AB + AC = \overline{AB} + AB + A\overline{B}C$가 되어 ③은 맞다.

④ O. 전개하면 $(A + \overline{B})(\overline{A} + B + C) = AB + AC + \overline{AB} + \overline{B}C$가 된다. 여기서 $\overline{B}C = A\overline{B}C + \overline{A}\overline{B}C$이고, 각 항은 각각 AC와 $\overline{AB}$에 흡수된다. 따라서 결국 $\overline{AB} + AB + AC$가 남으므로 ④도 맞다.

오답 포인트

최종 게이트도 NAND라는 점을 놓치면 중간 출력들을 그대로 더해 ②처럼 $+C$를 독립항으로 두기 쉽다. 또 아래 가지의 입력을 B, C로 잘못 읽으면 세 번째 중간식이 달라지므로, 배선이 실제로 A, C를 받는지부터 확인해야 한다.

다시 나오면 이렇게 판단

인버터를 지난 신호를 먼저 적고 각 NAND의 출력을 $\overline{XY} = \overline{X} + \overline{Y}$로 바꾼다. 그다음 마지막 NAND에서 드모르간 법칙을 한 번 더 적용하고, 끝에서 흡수법칙으로 중복항을 지우면 동치식을 빠르게 판별할 수 있다.

암기 포인트

NAND 회로는 가지별 출력을 먼저 읽고 마지막 NAND에서 한 번 더 드모르간을 적용하면 표준합 형태가 정리된다.

NoteBase

문 14. 디지털 로직에서 시프트가 좌측 또는 우측 방향으로 한 번에 1비트씩만 허용된다고 가정할 때, 2의 보수(2's complement)로 표현된 4비트 2진수 1101_2에 대해 1비트 우측 산술 시프트(Arithmetic Right Shift)를 수행한 결과로 가장 옳은 것은?

① 1110_2

② 1010_2

③ 0110_2

④ 1011_2

정답 1번

출제 포인트

이 문제는 2의 보수로 표현된 signed 수를 1비트 우측 산술 시프트할 때 부호 비트를 유지하는 원리를 묻는다.

조건 정리

주어진 값: 4비트 2의 보수 1101_2
구해야 할 값: 1비트 우측 산술 시프트 결과
주의할 조건: 논리 우시프트가 아니라 산술 우시프트이므로 빈 최상위 비트에 원래 부호 비트 1을 채운다.

사용할 원리

산술 우시프트는 모든 비트를 오른쪽으로 한 칸 이동시키고, 비어 있는 최상위 비트에는 원래의 부호 비트를 복사한다. 1101_2는 최상위 비트가 1이므로 음수 표현이며, 우시프트 후에도 최상위 비트는 1로 유지된다.

계산 과정

1. 1101_2의 부호 비트는 1이다.
2. 비트를 오른쪽으로 한 칸 이동하면 하위 세 비트는 110이 된다.
3. 산술 우시프트이므로 빈 최상위 비트에 1을 채워 결과는 1110_2가 된다.

NoteBase

선지 판단

① O: 산술 우시프트 결과가 1110_2이므로 옳다.
② X: 1010_2는 부호 비트를 유지한 우시프트 결과가 아니다. 이런 값은 시프트 방향이나 비트 이동 순서를 잘못 적용한 경우에 가깝다. 틀린 표현은 결과가 1010_2라는 점이며, 올바르게는 1110_2이다.
③ X: 0110_2는 최상위 비트를 0으로 채운 논리 우시프트 결과이다. 산술 우시프트는 0이 아니라 원래 부호 비트 1을 채워야 한다. 틀린 표현은 최상위 비트를 0으로 채운 점이며, 올바르게는 1110_2이다.
④ X: 1011_2는 오른쪽으로 한 칸 이동한 뒤 비트 배열을 잘못 적은 값이다. 산술 우시프트를 하면 하위 비트 배열은 110이 되어야 하므로 결과는 1011_2가 될 수 없다. 틀린 표현은 시프트 결과가 1011_2라는 점이며, 올바르게는 1110_2이다.

오답 포인트

산술 우시프트와 논리 우시프트를 혼동하면 ③처럼 0110_2를 고르기 쉽다. 2의 보수 문제에서는 먼저 signed 수인지 확인하고, 우시프트에서 빈 칸을 무엇으로 채우는지가 핵심 함정이다.

다시 나오면 이렇게 판단

2의 보수 시프트 문제는 먼저 산술 시프트인지 논리 시프트인지 구분하고, 산술이면 최상위 부호 비트를 그대로 복사한 뒤 나머지 비트를 한 칸 이동시켜 판단한다.

암기 포인트

산술 우시프트는 부호 유지, 논리 우시프트는 0 채움이다.

NoteBase

문 15. 유한상태 머신(FSM : Finite State Machine)은 대표적으로 밀리(Mealy) 머신과 무어(Moore) 머신으로 분류되는데, 이 두 가지 방식에 대한 설명으로 가장 옳지 않은 것은?

① Mealy 머신이 타이밍적으로 더 안정적이다.

② Mealy 머신이 더 빠른 출력 신호를 만들어 내는 데 유리하다.

③ 제어기 설계에서 Mealy 머신과 Moore 머신은 혼합하여 사용할 수 있다.

④ Mealy 머신의 출력 신호는 현재 상태와 입력 신호의 조합으로 생성된다.

정답 1번

출제 포인트

이 문제는 Mealy 머신과 Moore 머신의 출력 결정 방식과 타이밍 특성의 차이를 묻는 문제이다.

핵심 개념

Mealy 머신: 출력이 현재 상태와 현재 입력에 함께 의존하는 FSM이다. 결정 차이축: 출력이 입력 변화에 즉시 반응하는가이다. 키워드: 상태+입력, 빠른 반응이다.
Moore 머신: 출력이 현재 상태에만 의존하는 FSM이다. 결정 차이축: 출력이 상태에만 묶이는가이다. 키워드: 상태만, 안정적 출력이다.

선지별 해설

① X
틀린 부분은 "Mealy 머신이 타이밍적으로 더 안정적"이라는 표현이다. Mealy 머신은 입력 변화가 바로 출력에 반영되므로 순간적인 글리치나 흔들림이 나타나기 쉽다. 올바르게는 "Moore 머신이 타이밍적으로 더 안정적이다."이다.

② O
Mealy 머신은 출력이 상태와 입력의 조합으로 바로 결정되므로 입력 변화에 한 클록 빠르게 반응하는 형태를 만들기 쉽다. 따라서 더 빠른 출력 신호가 필요한 경우에 유리하다는 설명은 맞다.

③ O
실제 제어기 설계에서는 모든 출력을 한 방식으로만 만들 필요가 없다. 상태 기반의 안정적인 출력은 Moore 방식으로 두고, 즉시 반응이 필요한 일부 출력은 Mealy 방식으로 두는 혼합 설계가 가능하다.

④ O
Mealy 머신의 정의에 해당하는 설명이다. Mealy 출력은 현재 상태와 현재 입력을 함께 받는 조합논리의 결과로 생성된다.

NoteBase

오답 포인트

Mealy와 Moore를 구분할 때 학생들이 가장 많이 바꾸어 기억하는 부분은 속도와 안정성이다. Mealy는 빠르지만 입력 변화의 영향을 직접 받아 덜 안정적이고, Moore는 한 단계 느릴 수 있지만 출력이 상태에만 묶여 더 안정적이다.

다시 나오면 이렇게 판단

먼저 출력이 무엇에 의해 결정되는지를 본다. 출력이 상태만 보면 Moore이고 안정성 쪽 설명이 맞다. 출력이 상태와 입력을 함께 보면 Mealy이고 반응 속도 쪽 설명이 맞다. 문제에서 두 특성을 서로 바꾸어 놓으면 그 선지가 오답이다.

암기 포인트

Mealy는 상태+입력이라 빠르고, Moore는 상태만이라 안정적이다.

NoteBase

문 16. 새로 개발된 CPU인 'KS'가 32비트(4바이트) 단위로만 메모리 액세스 가능하도록 설계되었고 이 KS CPU의 주소 버스(address bus) 길이가 24비트로 주어졌다면 이 KS CPU를 이용해 액세스 가능한 메모리 최대 용량의 값[MiB]은? (단, 1[MiB]=2^{20}[Byte]를 의미한다.)

① 64

② 32

③ 24

④ 16

정답 1번

출제 포인트

이 문제는 주소 버스 비트 수와 한 번에 접근하는 메모리 단위를 이용해 전체 메모리 용량을 구하는 문제이다.

조건 정리

주어진 값:
- 주소 버스 길이: 24비트
- 메모리 접근 단위: 32비트 = 4바이트
- 1MiB = 2^{20}바이트

구해야 할 값:
- 액세스 가능한 최대 메모리 용량 [MiB]

주의할 조건:
- 이 CPU는 바이트 단위가 아니라 32비트 단위로만 메모리에 접근한다.

사용할 공식

최대 메모리 용량은
주소 개수 × 주소당 접근 단위 = 2^주소 버스 비트 수 × 단위 바이트 수
이다.
이 공식을 쓰는 이유는 주소 버스가 24비트이면 구분 가능한 주소가 2^{24}개이고, 각 주소가 가리키는 실제 메모리 단위가 4바이트이기 때문이다.

계산 과정

1. 주소 버스가 24비트이므로 주소 개수는 2^{24}개이다.
2. 한 주소가 가리키는 메모리 단위는 32비트이므로 4바이트이다.
3. 전체 용량은 $2^{24} \times 4 = 2^{24} \times 2^2 = 2^{26}$바이트이다.
4. 이를 MiB로 바꾸면 $2^{26}/2^{20} = 2^6 = 64$MiB이다.

NoteBase

선지별 해설

① O
- 계산 결과가 64MiB이므로 옳다.
- 주소 수 2^{24}개에 주소당 4바이트를 곱한 값을 정확히 반영한 선택지이다.

② X
- 32MiB는 전체 용량을 2^{25}바이트로 본 경우에 해당하므로 계산 결과와 다르다.
- 틀린 표현은 최대 용량을 32MiB로 본 점이다.
- 올바르게는 주소당 접근 단위를 4바이트로 적용하여 64MiB로 계산해야 한다.

③ X
- 24는 주소 버스의 비트 수일 뿐 메모리 용량 [MiB] 값이 아니다.
- 틀린 표현은 주소 버스 비트 수와 메모리 용량 수치를 같은 차원으로 본 점이다.
- 올바르게는 24비트 주소 버스로 2^{24}개의 주소를 만들고, 여기에 4바이트를 곱해 64MiB를 구해야 한다.

④ X
- 16MiB는 각 주소가 1바이트를 가리키는 바이트 주소 방식일 때의 값 2^{24}바이트이다.
- 틀린 표현은 이 CPU가 바이트 단위로 메모리에 접근한다고 본 점이다.
- 올바르게는 이 CPU는 32비트 = 4바이트 단위로만 접근하므로 최대 용량은 64MiB이다.

오답 포인트

가장 흔한 함정은 주소 버스가 24비트라는 이유만으로 2^{24}바이트, 즉 16MiB로 끝내는 것이다. 이 문항은 바이트 주소 방식이 아니라 32비트 단위 접근이라는 조건을 숨은 핵심으로 두고 있으므로, 주소 개수와 주소당 데이터 크기를 반드시 분리해서 계산해야 한다.

다시 나오면 이렇게 판단

메모리 용량 문제는 먼저 주소 버스로 주소 개수 2^n을 구하고, 다음에 한 주소가 가리키는 데이터 크기가 몇 바이트인지 확인한 뒤 곱해서 전체 용량을 계산하면 된다. 마지막에 바이트를 MiB나 KiB로 단위 변환하면 된다.

암기 포인트

메모리 용량은 주소 수 × 주소당 바이트 수이며, 워드 주소 방식이면 바이트 수를 반드시 한 번 더 곱한다.

NoteBase

문 17. <보기>의 부울함수식 $F(a,b,c,d)$를 가장 간략화한 것으로 옳은 것은?

<보기>

$F(a,b,c,d) = \bar{b}c\bar{d} + \bar{a}\bar{b}c + bc + a\bar{c}d + \bar{b}\bar{c}\bar{d} + \bar{b}cd$

① $c + a\bar{b} + ad + \bar{b}\bar{c}\bar{d}$

② $c + ad + \bar{b}\bar{d}$

③ $c + a\bar{b} + ad + \bar{b}\bar{d}$

④ $c + a\bar{b} + ad$

정답 2번

출제 포인트

이 문제는 흡수법칙과 여집합 결합을 이용해 부울식을 최소항으로 간략화하는 문제이다.

조건 정리

주어진 식은 $F(a,b,c,d) = \bar{b}c\bar{d} + \bar{a}\bar{b}c + bc + a\bar{c}d + \bar{b}\bar{c}\bar{d} + \bar{b}cd$이다.
구해야 할 값은 이 식을 가장 간단한 합의 형태로 줄인 결과이다.
핵심은 c를 먼저 묶어 만든 뒤, $c + \bar{c}X = c + X$를 적용하는 데 있다.

사용할 공식

$X\bar{d} + Xd = X$, $\bar{b} + b = 1$, $c + \bar{c}X = c + X$를 사용한다.
또한 컨센서스 정리 $XY + \bar{X}Z + YZ = XY + \bar{X}Z$를 이용하면 중복항을 제거할 수 있다.

계산 과정

1. $\bar{b}c\bar{d} + \bar{b}cd = \bar{b}c(\bar{d} + d) = \bar{b}c$이다.
2. 따라서 $\bar{b}c + \bar{a}\bar{b}c + bc = c$이다. 이미 c가 있으면 $\bar{a}\bar{b}c$도 흡수된다.
3. 원식은 $F = c + a\bar{c}d + \bar{b}\bar{c}\bar{d}$가 된다.
4. $F = c + \bar{c}(ad + \bar{b}\bar{d}) = c + ad + \bar{b}\bar{d}$이다.

NoteBase

선지별 해설

① X

$c + a\bar{b} + ad + \bar{b}\bar{c}\bar{d}$는 같은 함숫값을 만들 수는 있지만 $a\bar{b}$를 남겨 두어 최소식이 아니다.
$a\bar{b} = 1$인 경우는 $c = 1$이면 c가, $c = 0, d = 1$이면 ad가, $c = 0, d = 0$이면 $\bar{b}\bar{c}\bar{d}$가 이미 덮는다.
틀린 표현은 $a\bar{b}$를 그대로 둔 점이며, 더 줄이면 $c + ad + \bar{b}\bar{d}$가 된다.

② O

계산 결과가 그대로 $c + ad + \bar{b}\bar{d}$이므로 가장 간략화한 식이다.
c가 1이면 즉시 1이고, $c = 0$일 때만 ad와 $\bar{b}\bar{d}$를 검사하면 된다.

③ X

$c + a\bar{b} + ad + \bar{b}\bar{d}$에서 $a\bar{b}$는 ad와 $\bar{b}\bar{d}$가 있으면 필요 없는 컨센서스 항이다.
따라서 $ad + \bar{b}\bar{d} + a\bar{b} = ad + \bar{b}\bar{d}$가 되어 $a\bar{b}$를 삭제할 수 있다.
틀린 표현은 $a\bar{b}$를 남긴 점이며, 올바른 최소식은 $c + ad + \bar{b}\bar{d}$이다.

④ X

$c + a\bar{b} + ad$는 $\bar{b}\bar{d}$ 항이 빠져 있어 원식을 모두 표현하지 못한다.
예를 들어 $(a, b, c, d) = (0, 0, 0, 0)$이면 원식은 $\bar{b}\bar{c}\bar{d} = 1$이라서 1이지만, ④는 0이다.
틀린 표현은 $\bar{b}\bar{d}$를 누락한 점이며, 올바르게는 $c + ad + \bar{b}\bar{d}$이다.

오답 포인트

이 문항의 함정은 식이 같으냐와 식이 가장 간단하냐를 구분하지 못하는 데 있다.
①과 ③은 동치식일 수 있어도 중복항이 남아 있으므로 정답이 될 수 없다.
또한 c가 만들어진 뒤에도 $c + \bar{c}X = c + X$를 한 번 더 적용해야 최소식까지 내려간다.

다시 나오면 이렇게 판단

부울식 간소화 문제는 먼저 같은 문자만 다른 항을 $X\bar{d} + Xd = X$ 형태로 묶고, 그다음 $\bar{b}c + bc = c$처럼 1을 만드는 항을 찾는다.
마지막에는 $c + \bar{c}X = c + X$와 컨센서스 정리로 중복항이 남았는지 한 번 더 확인하면 된다.

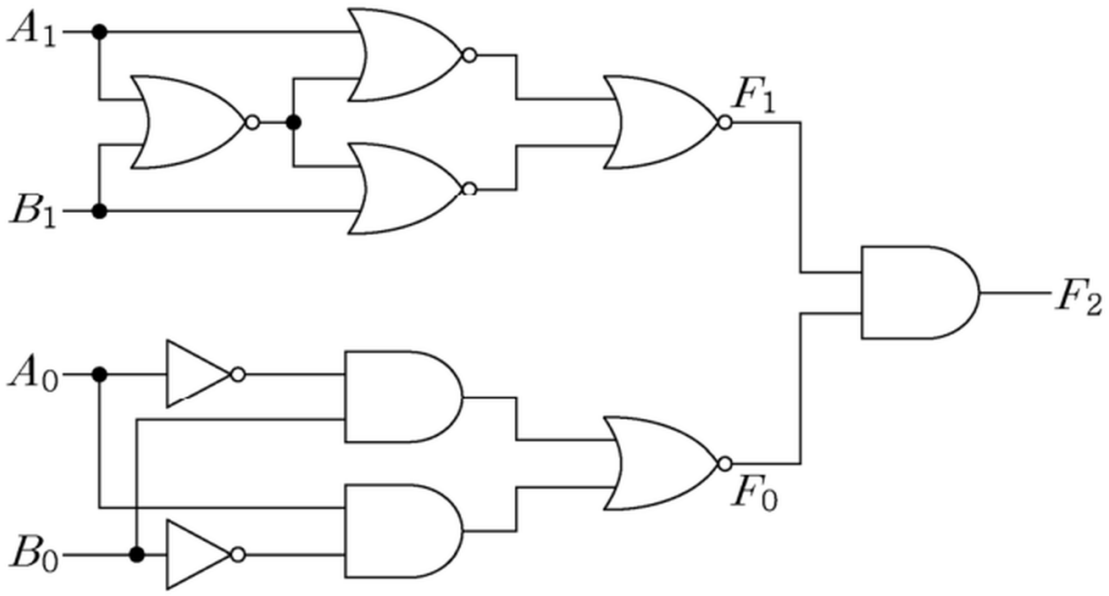
암기 포인트

$X\bar{d} + Xd = X$, $c + \bar{c}X = c + X$, 컨센서스 항 YZ는 지워질 수 있다는 점이 최소식 판별의 핵심이다.

NoteBase

문 18. 두 2진수 A, B는 $A = A_1A_0$, $B = B_1B_0$로 각각 2bit로 구성되어 <보기>의 입력 단에 연결되어 있다. 이 회로의 출력 F_0, F_1, F_2에 대한 설명으로 가장 옳지 않은 것은? (단, A_1은 A의 MSB이고 B_1은 B의 MSB이다.)

<보기>



① $B_1 = 0$이면, $F_1 = \overline{A_1}$

② $F_2 = 1$이면, $A > B$

③ $A_0 = 1$이면, $F_0 = B_0$

④ $F_2 = 0$이면, $A \neq B$

정답 2번

출제 포인트

이 문제는 회로도에서 각 게이트의 논리식을 단계적으로 정리하여 F_1, F_0, F_2가 각각 무엇을 뜻하는지 판별하는 문제이다.

NoteBase

선지별 해설

① O
- 상단 첫 게이트는 NOR이므로 출력은 $\overline{A_1 + B_1}$이다.
- 그 다음 두 NOR 게이트를 정리하면 각각 $\overline{A_1}B_1$, $A_1\overline{B_1}$가 되고, 마지막 NOR 출력은 $F_1 = A_1B_1 + \overline{A_1B_1}$이다.
- 따라서 F_1은 A_1과 B_1의 XNOR이다. $B_1 = 0$이면 $F_1 = \overline{A_1}$이므로 옳다.

② X
- 하단 회로도 같은 방식으로 정리하면 $F_0 = A_0B_0 + \overline{A_0B_0}$가 된다.
- 마지막 AND 게이트 때문에 $F_2 = F_1 \cdot F_0$이다. 즉 $F_2 = 1$은 MSB와 LSB가 모두 같다는 뜻이므로 $A = B$이다.
- 선지의 "$F_2 = 1$이면, $A > B$"가 틀렸고, 올바르게는 "$F_2 = 1$이면, $A = B$"이다.

③ O
- F_0는 A_0와 B_0의 XNOR이므로 $A_0 = 1$을 대입하면 $F_0 = B_0$가 된다.
- 따라서 옳다.

④ O
- $F_2 = F_1 \cdot F_0$는 두 비트가 모두 같은 경우에만 1이 된다.
- 따라서 $F_2 = 0$이면 적어도 한 비트가 다르므로 $A \neq B$이다.
- 따라서 옳다.

오답 포인트

상단과 하단 회로의 마지막 게이트가 모두 NOR이라는 점을 놓치면 XOR와 XNOR를 뒤바꾸기 쉽다. 특히 ②는 비교기 회로라는 이유만으로 F_2를 크기 비교 출력으로 성급하게 해석하도록 유도한 함정이다.

다시 나오면 이렇게 판단

회로 비교 문제는 각 블록을 바로 전체 의미로 읽지 말고 먼저 중간 출력식을 만든 뒤 마지막 게이트까지 정리한다. 두 비트 쌍에서 각각 XNOR이 나오고 마지막이 AND이면 전체 출력은 같음 판정 신호이다.

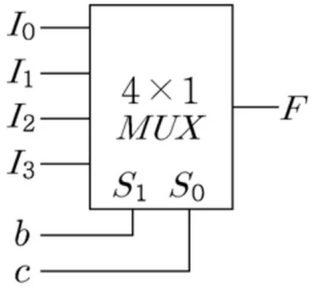
암기 포인트

XNOR는 같으면 1이고, 두 자리 비교에서 각 자리 XNOR의 AND는 $A = B$ 판정이다.

NoteBase

문 19. <보기 1>의 (A)에서 4×1멀티플렉서 회로가 (B)의 진리표를 만족한다. 멀티플렉서의 기능표가 <보기 2>와 같이 주어졌을 때, 멀티플렉서의 입력 I_0, I_1, I_2, I_3의 값으로 가장 옳은 것은?

<보기 1>



(A) 회로도

a	b	c	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

(B) 진리표

<보기 2>

S_1	S_0	F
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

① $I_0 = \overline{a},\ I_1 = 1,\ I_2 = 0,\ I_3 = a$

② $I_0 = I_2 = a,\ I_1 = 1,\ I_3 = \overline{a}$

③ $I_0 = I_2 = a,\ I_1 = 0,\ I_3 = \overline{a}$

④ $I_0 = a,\ I_1 = 0,\ I_2 = 1,\ I_3 = \overline{a}$

정답 3번

NoteBase

출제 포인트

이 문제는 4×1 멀티플렉서에서 선택선 S_1, S_0에 연결된 b, c를 기준으로 각 입력 I_0, I_1, I_2, I_3가 어떤 함수가 되어야 하는지를 진리표와 대응시키는 문제이다.

선지별 해설

① X: $bc = 00$일 때 출력은 I_0인데, 진리표에서 $a = 0$이면 0, $a = 1$이면 1이므로 $I_0 = a$이어야 한다. 선지의 $I_0 = \overline{a}$가 틀렸고, 올바르게는 $I_0 = a$이다. 또한 $bc = 10$일 때는 I_2가 선택되는데 진리표상 $I_2 = a$이므로 $I_2 = 0$도 틀렸고, 올바르게는 $I_2 = a$이다.
② X: $I_0 = a, I_2 = a, I_3 = \overline{a}$까지는 맞지만, $bc = 01$일 때 선택되는 I_1이 틀렸다. 진리표에서 $b = 0, c = 1$이면 a가 0이든 1이든 항상 $F = 0$이므로 올바르게는 $I_1 = 0$이다.
③ O: 선택선이 $S_1 = b, S_0 = c$이므로 $bc = 00 \rightarrow I_0$, $01 \rightarrow I_1$, $10 \rightarrow I_2$, $11 \rightarrow I_3$로 대응시킨다. 각 경우의 진리표를 보면 $I_0 = a, I_1 = 0, I_2 = a, I_3 = \overline{a}$가 되어 정확하다.
④ X: $I_1 = 0, I_3 = \overline{a}$는 맞지만 I_0와 I_2가 틀렸다. $bc = 00$에서는 $I_0 = a$가 되어야 하므로 선지의 $I_0 = a$는 맞더라도, $bc = 10$에서는 진리표가 a와 동일하게 변하므로 $I_2 = 1$이 아니라 $I_2 = a$가 맞다. 틀린 표현은 $I_2 = 1$이고, 올바르게는 $I_2 = a$이다.

오답 포인트

선택선의 순서를 거꾸로 읽거나, 각 bc 조합에서 a가 변할 때 F가 어떻게 달라지는지를 보지 않으면 I_1과 I_2를 상수로 잘못 두기 쉽다. 특히 $bc = 01$에서는 항상 0이고, $bc = 10$에서는 a를 그대로 따라간다는 점이 함정이다.

다시 나오면 이렇게 판단

먼저 선택선 연결을 확인해 S_1S_0와 입력 $I_0 \sim I_3$의 대응을 정한다. 그다음 진리표를 $bc = 00, 01, 10, 11$ 네 묶음으로 나누고, 각 묶음에서 a가 0일 때와 1일 때의 F를 비교해 $0, 1, a, \overline{a}$ 중 무엇인지 결정한다.

암기 포인트

멀티플렉서 치환은 선택선 조합별로 행을 묶어 보고, 남는 변수에 대해 출력이 $0, 1, a, \overline{a}$ 중 무엇인지 읽어내면 된다.

NoteBase

문 20. 순차 상태와 입력 신호에 따라 적절한 제어 신호를 생성하는 디지털 시스템 상태 제어기를 설계하는 방식으로 소프트웨어 알고리즘을 기술하는 순서도와 유사한 SM(state machine) 차트가 있다. SM 차트를 구성하는 기본 요소가 아닌 것은?

① 상태 박스(State box)

② 결정 박스(Decision box)

③ 입력 박스(Input box)

④ 조건 출력 박스(Conditional output box)

정답 3번

출제 포인트

이 문제는 SM 차트의 기본 구성 요소와 각 요소의 역할을 구별하는 문제이다.

핵심 개념

상태 박스는 한 상태에서 유지되는 동작과 기본 출력을 나타내는 중심 요소이다.
결정 박스는 입력 조건이나 내부 조건을 검사하여 다음 경로를 분기하는 요소이다.
조건 출력 박스는 특정 조건이 만족될 때만 출력이 발생하도록 표시하는 요소이다.
입력은 SM 차트에서 별도의 '입력 박스'라는 기본 요소로 두지 않고, 결정 박스에서 판단 조건으로 사용한다.

선지별 해설

① O. 상태 박스는 SM 차트의 기본 요소이며, 현재 상태에서 수행되는 동작과 출력을 표현한다.
② O. 결정 박스는 입력 신호나 조건값에 따라 다음 상태 또는 동작 경로를 선택하는 기본 요소이다.
③ X. 틀린 부분은 '입력 박스'이다. SM 차트의 기본 요소에는 입력 박스가 없고 입력 신호는 결정 박스에서 조건으로 판정한다. 올바르게는 '입력 조건은 결정 박스에서 검사한다'이다.
④ O. 조건 출력 박스는 조건이 참일 때만 출력이 발생하는 경우를 나타내는 기본 요소이다.

오답 포인트

순서도에는 입력과 출력 기호가 따로 있으므로 이를 SM 차트에도 그대로 대응시키기 쉽다. 이 문항은 '입력 신호를 다룬다'는 설명 때문에 입력 박스가 있을 것처럼 착각하는 지점을 노린 문제이다.

다시 나오면 이렇게 판단

SM 차트의 기본 박스는 상태 박스, 결정 박스, 조건 출력 박스로 정리한다. 입력은 박스 이름이 아니라 결정 박스에서 검사되는 판단 조건인지 먼저 보면 된다.

NoteBase

암기 포인트

SM 차트는 상태-결정-조건출력의 3요소이며 입력 박스는 없다.

NoteBase