

◆ 16년 3월 고2 16~18번

[16~18] 다음을 읽고 물음에 답하시오.

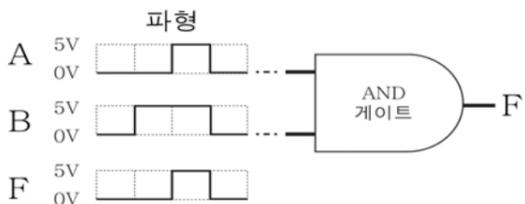
컴퓨터와 같은 디지털 장치는 1과 0밖에 구분하지 못한다. 더 정확하게 말하면 실제 숫자를 구분하는 것이 아니라 신호가 왔는지(1), 오지 않았는지(0)의 정보를 구분하는 것이다. 신호 여부는 두 개의 전압 레벨, 예를 들어 5V와 0V를 사용하여 구분한다. 즉 전압 레벨이 변화하는 것을 2진수로 처리하여 디지털 회로를 설계하는 것이다. 디지털 회로의 기본 요소는 논리 게이트인데, 논리 게이트는 하나 이상의 입력 값에 대한 논리 연산을 수행하여 출력 값을 얻는다.

디지털 회로는 출력을 결정하는 방법에 따라 조합 논리 회로와 순차 논리 회로로 나눌 수 있다. 조합 논리 회로는 현재의 입력 값들만 이용하여 출력 값을 결정한다. 즉 회로를 구성하는 논리 게이트들이 입력 신호들을 받는 즉시 그것들을 조합하여 출력 신호를 발생시킨다. 반면 순차 논리 회로는 과거의 출력 값이 현재의 출력에 영향을 미친다. 출력 값이 그 시점의 입력 값뿐만 아니라 이전 상태의 출력

[A] 값에 의해서도 결정되는 것이다. 가령 디지털 장치에서 수를 셀 때, 이전 상태의 출력 값과 현재의 값을 논리 연산하여 출력하므로 다음 상태로 변화할 때까지 현 상태를 기억하는 기능이 필요하다. 이러한 특성 때문에 순차 논리 회로는 조합 논리 회로와 달리 기억 기능을 가지고 있다. 이전 상태의 출력 값은 다음 단계의 순차 논리 회로 동작을 위해 피드백 경로를 통해 다시 순차 논리 회로의 입력으로 들어가게 된다.

조합 논리 회로이든 순차 논리 회로이든 디지털 회로의 설계는 다양한 논리 게이트들을 얼마나 효율적으로 연결하느냐가 중요하다. 가장 기본적인 논리 게이트로는 NOT 게이트, AND 게이트, OR 게이트가 있다. NOT 게이트는 보통 인버터라 부르며 출력 값이 입력 값과 반대가 되도록 변환한다. 예를 들어 입력 값이 0이면 출력 값은 1이고 입력 값이 1이면 출력 값은 0이 된다. 따라서 입력 가능한 조합은 1과 0, 두 개뿐이다. AND 게이트는 입력 단자를 통해 들어오는 입력 값이 모두 1일 때만 출력 값이 1이고, 만일 한 개라도 0이면 출력 값은 0이 된다. OR 게이트는 입력 값이 어느 하나라도 1이면 출력 값이 1이 되고, 입력 값이 모두 0일 때만 출력 값이 0이 된다. 논리 게이트들의 입력 가능한 조합의 수는 2의 거듭제곱을 따른다. 즉 입력 단자가 2개면 입력 가능 조합은 4개, 입력 단자가 3개면 입력 가능 조합은 8개가 된다.

논리 게이트의 입력과 출력은 전기적 신호가 바뀌는 모습으로 나타낼 수 있다. 아래 그림은 AND 게이트에 입력 신호가 들어왔을 때, 어떤 출력 신호가 나오는지 나타낸 것이다. A와 B의 파형이 각각의 입력 단자에 들어올 때, AND 게이트는 F와 같은 파형을 출력하게 된다. 여기서 파형이란 0과 1에 해당하는 전기적 신호(0V와 5V)가 시간에 따라 연속적으로 바뀌는 모습을 표현한 것을 말한다. 다른 게이트들도 이와 마찬가지로 전기적 신호가 바뀌는 모습을 표현하여 입력 신호와 출력 신호로 나타낼 수 있다.



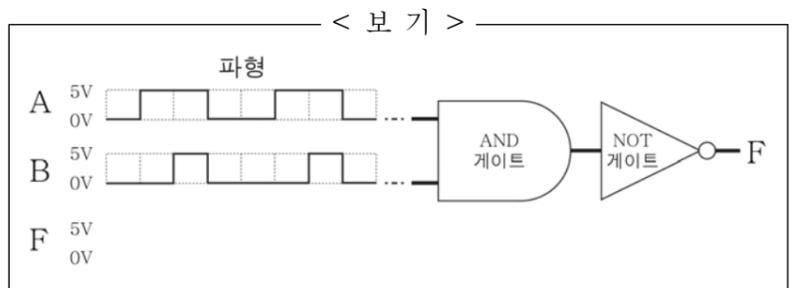
16. 밑글의 내용과 일치하지 않는 것은?

- ① 어떤 논리 게이트라도 출력 값은 0과 1, 둘 중 하나이다.
- ② 논리 게이트의 입력과 출력은 전기적 신호가 바뀌는 모습으로 나타낼 수 있다.
- ③ 어떤 논리 게이트에 n개의 입력 단자가 있으면 입력 가능한 조합의 수는 2ⁿ개이다.
- ④ NOT 게이트, AND 게이트, OR 게이트에서 입력 값이 모두 0이면 각각의 출력 값은 모두 0이다.
- ⑤ 조합 논리 회로와 순차 논리 회로는 둘 다 논리 게이트를 연결해 구현할 수 있다는 점에서는 같다.

17. [A]로 보아 출력을 결정하는 방법이 다른 하나는?

- ① 자동차의 문이 열리면 경고음이 울리는 경우
- ② 현관에 사람이 들어왔을 때 전등이 켜지는 경우
- ③ 사람이 다가가야만 움직이기 시작하는 에스컬레이터의 경우
- ④ 이용객이 안전띠를 착용하지 않아 운행되지 않는 놀이 기구의 경우
- ⑤ 은행에서 지폐를 세는 기계가 만 원권의 개수를 세어 총액을 나타내는 경우

18. 밑글을 참고할 때, <보기>에서 출력될 F의 파형으로 적절한 것은?



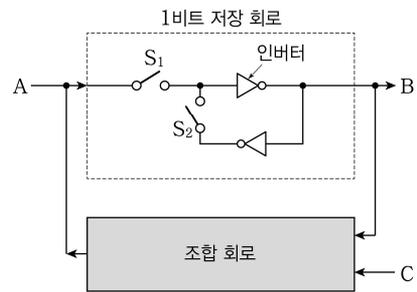
- ① F 5V 0V [Timing diagram showing a high pulse followed by a low pulse]
- ② F 5V 0V [Timing diagram showing a low pulse followed by a high pulse]
- ③ F 5V 0V [Timing diagram showing a high pulse followed by a low pulse]
- ④ F 5V 0V [Timing diagram showing a low pulse followed by a high pulse]
- ⑤ F 5V 0V [Timing diagram showing a high pulse followed by a low pulse]

◆ 15 LEET 언어이해 27~29번

[27~29] 다음 글을 읽고 물음에 답하시오.

컴퓨터의 CPU가 어떤 작업을 수행하는 것은 CPU의 '논리 상태'가 시간에 따라 바뀌는 것을 말한다. 가령, $Z = X + Y$ 의 연산을 수행하려면 CPU가 X와 Y에 어떤 값을 차례로 저장한 다음, 이것을 더하고 그 결과를 Z에 저장하는 각각의 기능을 순차적으로 진행해야 한다. CPU가 수행할 수 있는 기능은 특정한 CPU의 논리 상태와 일대일로 대응되어 있으며, 프로그램은 수행하고자 하는 작업의 진행에 맞도록 CPU의 논리 상태를 변경한다. 이를 위해 CPU는 현재 상태를 저장하고 이것에 따라 해당 기능을 수행할 수 있는 부가 회로도 갖추고 있다. 만약 CPU가 가지는 논리 상태의 개수가 많아지면 한 번에 처리할 수 있는 기능이 다양해진다. 따라서 처리할 데이터의 양이 같다면 이를 완료하는 데 걸리는 시간이 줄어든다.

논리 상태는 2진수로 표현되는데 논리 함수를 통해 다른 상태로 변환된다. 논리 소자가 연결된 조합 회로는 논리 함수의 기능을 가지는데, 조합 회로는 논리 연산은 가능하지만 논리 상태를 저장할 수는 없다. 어떤 논리 상태를 '저장'한다는 것은 2진수 정보의 시간적 유지를 의미하는데, 외부에서 입력이 유지되지 않더라도 입력된 정보를 논리 회로 속에 시간적으로 가둘 수 있어야 한다.



<그림> 순차 논리 회로

인버터는 입력이 0일 때 1을, 1일 때 0을 출력하는 논리 소자이다. <그림>의 점선 내부에 표시된 '1비트 저장 회로'를 생각해 보자. 이 회로에서 스위치 S_1 은 연결하고 스위치 S_2 는 끊은 채로 A에 정보를 입력한다. 그런 다음 S_2 를 연결하면 S_1 을 끊더라도 S_2 를 통하는 ㉠ 피드백 회로에 의해 A에 입력된 정보와 반대되는 값이 지속적으로 B에 출력된다. 따라서 이 회로는 0과 1 중 1개의 논리 상태, 즉 1비트의 정보를 저장할 수 있다. 이러한 회로가 2개가 있다면 00, 01, 10, 11의 4가지 논리 상태, n개가 있다면 2^n 가지의 논리 상태 중 1개를 저장할 수 있다.

그렇다면 논리 상태의 변화는 어떻게 일어날까? 이제 <그림>과 같이 1비트 저장 회로와 조합 회로로 구성되는 '순차 논리 회로'를 생각해 보자. 이 회로에서 조합 회로는 외부 입력 C와 저장 회로의 출력 B를 다시 입력으로 되받아, 내장된 논리 함수를 통해 논리 상태를 변환하고, 이를 다시 저장 회로의 입력과 연결하는 ㉡ 피드백 회로를 구성한다. 예를 들어 조합 회로가 두 입력이 같을 때는 1을, 그렇지 않을 경우 0을 출력한다고 하자. 만일 B에서 1이 출력되고 있을 때 C에 1이 입력된다면 조합 회로는 1을 출력하게

된다. 이때 외부에서 어떤 신호를 주어 S_2 가 열리자마자 S_1 이 닫힌 다음 다시 S_2 가 닫히고 S_1 이 열리는 일련의 스위치 동작이 일어나도록 하면, 조합 회로의 출력은 저장 회로의 입력과 연결되어 있으므로 B에서 출력되는 값은 0으로 바뀐다. 그런 다음 C의 값을 0으로 바꾸어주면, 일련의 스위치 동작이 다시 일어나더라도 B의 값은 바뀌지 않는다. 하지만 C에 다시 1을 입력하고 일련의 스위치 동작이 일어나도록 하면 B의 출력은 1로 바뀐다. 따라서 C에 주는 입력에 의해 저장 회로가 출력하는 논리 상태를 임의로 바꿀 수 있다.

만일 이 회로에 2개의 1비트 저장 회로를 병렬로 두어 출력을 2비트로 확장하면 00~11의 4가지 논리 상태 중 1개를 출력할 수 있다. 조합 회로의 외부 입력도 2비트로 확장하면 조합 회로는 저장 회로의 현재 출력과 합친 4비트를 입력받게 된다. 이를 내장된 논리 함수에 의해 다시 2비트 출력을 만들어 저장 회로의 입력과 연결한다. 이와 같이 2비트로 확장된 순차 논리 회로에서 외부 입력을 주고 스위치 동작이 일어나도록 하면, 저장 회로의 출력은 2배로 늘어난 논리 상태 중 하나로 바뀐다.

이 회로에 일정한 시간 간격으로 외부 입력을 바꾸고 스위치 동작 신호를 주면, 주어지는 외부 입력에 따라 특정한 논리 상태가 순차적으로 출력에 나타나게 된다. ④ 이런 회로가 N비트로 확장된 대표적인 사례가 CPU이며 스위치를 동작시키는 신호가 CPU 클록이다. 회로 외부에서 입력되는 정보는 컴퓨터 프로그램의 '명령 코드'가 된다. 명령 코드를 CPU의 외부 입력으로 주고 클록 신호를 주면 CPU의 현재 논리 상태는 특정 논리 상태로 바뀐다. 이때 출력에 연결된 회로가 바뀐 상태에 해당하는 기능을 수행하게 된다. CPU 클록은 CPU의 상태 변경 속도, 즉 CPU의 처리 속도를 결정한다.

27. 윗글의 내용과 일치하지 않는 것은?

- ① CPU가 수행할 수 있는 기능과 그에 해당하는 논리 상태는 정해져 있다.
- ② 인버터는 입력되는 2진수 논리 값과 반대되는 값을 출력하는 논리 소자이다.
- ③ 순차 논리 회로에서 저장 회로의 출력은 조합 회로의 출력 상태와 동일하다.
- ④ CPU는 프로그램 명령 코드에 의한 논리 상태 변경을 통해 작업을 수행한다.
- ⑤ 조합 회로는 2진수 입력에 대해 내부에 구현된 논리 함수의 결과를 출력한다.

28. ㉠과 ㉡에 대한 이해로 적절한 것은?

- ① ㉠은 조합 회로를 통해서, ㉡은 인버터를 통해서 피드백 기능이 구현된다.
- ② ㉠과 ㉡의 각 회로에서 피드백 기능을 위해 입력하는 정보의 개수는 같다.
- ③ ㉠과 ㉡은 모두 외부에서 입력되는 논리 상태를 그대로 저장하는 기능이 있다.
- ④ ㉠은 정보를 저장하기 위한 구조이며, ㉡은 논리 상태를 변경하기 위한 구조이다.
- ⑤ ㉠은 스위치 S_1 이 연결될 때, ㉡은 스위치 S_2 가 연결될 때 피드백 기능이 동작한다.

29. ㉠에서 N을 증가시켰을 때의 변화를 이해한 것으로 적절하지 않는 것은?

- ① 프로그램에서 사용 가능한 명령 코드의 종류가 증가한다.
- ② 조합 회로가 출력하는 논리 상태의 가짓수가 증가한다.
- ③ CPU가 가질 수 있는 논리 상태의 가짓수가 증가한다.
- ④ CPU에서 진행되는 상태 변경의 속도가 증가한다.
- ⑤ 동일한 양의 데이터를 처리하는 속도가 증가한다.

◆ 14년 6월 고1 28~30번

[28~30] 다음 글을 읽고 물음에 답하시오.

컴퓨터로 작업을 하다가 전원이 꺼져 작업하던 데이터가 사라져 낭패를 본 경험이 한 번쯤은 있을 것이다. 이는 현재 컴퓨터에서 주 메모리로 D램을 사용하기 때문이다. D램은 전기장의 영향을 받으면 극성을 띠게 되는 물질을 사용하는데 극성을 띠면 1, 그렇지 않으면 0이 된다. 그런데 D램에 사용되는 물질의 극성은 지속적으로 전원을 공급해야만 유지된다. 그래서 D램은 읽기나 쓰기 작업을 하지 않아도 전력이 소모되며, 전원이 꺼지면 데이터가 모두 사라진다는 문제점을 안고 있다.

이러한 D램의 문제를 해결할 수 있는 차세대 램 메모리로서 가장 주목을 받고 있는 것은 M램이다. M램은 두 장의 자성 물질 사이에 얇은 절연막을 끼워 넣어 접합한 구조로 되어 있다. 절연막은 일반적으로 전류의 흐름을 막는 것이지만 M램에서는 절연막이 매우 얇아 전류가 통과할 수 있다. 그리고 자성 물질은 자석처럼 일정한 자기장 방향을 가지는데, 아래 위 자성 물질의 자기장 방향에 따라 저항이 달라진다. 자기장 방향이 반대일 경우 저항이 커져 전류가 약해지지만 자기장 방향이 같을 경우 저항이 약해져 상대적으로 강한 전류가 흐르게 된다. M램은 이 전류의 강도 차이를 감지해 전류가 상대적으로 약할 때 0, 강할 때 1로 읽게 된다. 자성 물질은, 강한 전기 자극을 가하면 자기장 방향이 바뀌는데 이를 이용해 한쪽 자성 물질의 자기장 방향만 바꿈으로써 쓰기 작업도 할 수 있다.

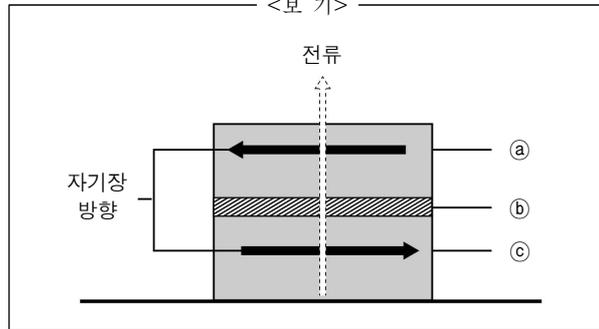
자성 물질의 자기장 방향은 전기 자극을 가해주지 않는 이상 변하지 않기 때문에 M램에서는 D램에서처럼 지속적으로 전원을 공급할 필요가 없다. 그렇기 때문에 D램에 비해 훨씬 적은 양의 전력을 사용하면서도 속도가 빠르며, 전원이 꺼져도 데이터를 잃어버릴 염려가 없다. 이런 장점들로 인해 M램이 일반화되면 컴퓨터뿐만 아니라 스마트폰이나 태블릿 PC와 같은 모바일 기기들의 성능은 크게 향상될 것이다.

그러나 M램이 일반화되기 위해서는 기술적 과제들도 많다. M램은 매우 얇은 막들을 쌓은 구조이기 때문에 이러한 얇은 막들이 원하는 기능을 하도록 제어하는 것은 기존의 반도체 공정으로는 매우 어렵다. 그리고 현재 사용하고 있는 자성 물질을 고도로 집적할 경우 자성 물질의 자기장이 인접한 자성 물질에 영향을 주는 문제도 있다. 이러한 문제를 해결할 수 있는 새로운 재료의 개발과 제조 공정의 개선이 이루어진다면 세계 반도체 시장의 판도도 크게 바뀔 것으로 보인다.

28. 밑줄의 내용과 일치하지 않는 것은?

- ① D램과 M램 모두 0 또는 1로 정보를 기록한다.
- ② M램은 자성 물질의 자기장이 강할수록 성능이 우수하다.
- ③ M램에서는 전류의 강도 차이를 감지해 데이터를 읽는다.
- ④ D램은 전원을 공급해주지 않으면 0의 값을 가지게 된다.
- ⑤ D램에서는 읽기나 쓰기 작업을 하지 않아도 전력이 소모된다.

29. <보기>는 M램의 구조를 나타낸 것이다. 이에 대한 이해로 적절한 것은? [3점]



- ① 현재 이 메모리에 기록된 정보는 1이다.
- ② 현재 전류는 상대적으로 큰 저항을 받고 있다.
- ③ ⑥는 전류가 원활하게 통과할 수 있게 해 준다.
- ④ ②와 ③의 자기장 방향은 전기를 가해 주었을 때만 나타난다.
- ⑤ 쓰기를 하려면 ②와 동시에 ③의 자기장 방향도 바꾸어야 한다.

30. 밑줄을 바탕으로 M램을 소개하려고 한다. 계획으로 적절하지 않은 것은?

제목: M램의 시대가 온다

M램의 장점

- 제조 비용이 저렴함 ㉠
- 전력 소모가 적어 모바일 기기에 적합 ㉡
- 전원이 꺼져도 데이터가 사라지지 않음 ㉢

M램이 넘어야 할 기술적 과제

- 제조 공정의 개선 ㉣
- 새로운 자성 물질의 개발 ㉤

- ① ㉠ ② ㉡ ③ ㉢ ④ ㉣ ⑤ ㉤

◆ 16년 6월 고1 20~22번

[20~22] 다음 글을 읽고 물음에 답하시오.

컴퓨터를 구성하고 있는 여러 가지 장치 중에서 가장 핵심적인 역할을 담당하고 있는 3가지 요소는 중앙처리장치(CPU), 주기억장치, 보조기억장치이다. 보통 주기억장치로 ‘램’을, 보조기억장치로 ‘HDD(Hard Disk Drive)’를 쓴다. 이 세 장치의 성능이 컴퓨터의 전반적인 속도를 좌우한다고 할 수 있다.

CPU나 램은 내부의 미세 회로 사이를 오가는 전자의 움직임만으로 데이터를 처리하는 반도체 재질이기에 때문에 고속으로 동작이 가능하다. 그러나 HDD는 원형의 자기디스크를 물리적으로 회전시키며 데이터를 읽거나 저장하기 때문에 자기디스크를 아무리 빨리 회전시킨다 해도 반도체의 처리 속도를 따라갈 수 없다. 게다가 디스크의 회전 속도가 빨라질수록 소음이 심해지고 전력 소모량이 급속도로 높아지는 단점이 있다. 이 때문에 CPU와 램의 동작 속도가 하루가 다르게 향상되고 있는 반면, HDD의 동작 속도는 그렇지 못했다.

그래서 HDD의 대안으로 제시된 것이 바로 ‘SSD(Solid State Drive)’이다. SSD의 용도나 외관, 설치 방법 등은 HDD와 유사하다. 하지만 SSD는 HDD가 자기디스크를 사용하는 것과 달리 반도체를 이용해 데이터를 저장한다는 차이가 있다. 그리고 물리적으로 움직이는 부품이 없기 때문에 작동 소음이 작고 전력 소모가 적다. 이런 특성 때문에 휴대용 컴퓨터에 SSD를 사용하면 전지 유지 시간을 늘릴 수 있다는 이점이 있다.

SSD는, 컴퓨터 시스템*과 SSD 사이에 데이터를 주고받을 수 있도록 연결하는 부분인 ‘인터페이스’, 데이터를 저장하는 ‘메모리’, 그리고 인터페이스와 메모리 사이의 데이터 교환 작업을 제어하는 ‘컨트롤러’, 외부 장치와 SSD간의 처리 속도 차이를 줄여주는 ‘버퍼 메모리’로 이루어져 있다. 이 중에 주목해야 할 것이 데이터를 저장하는 메모리다. 이 메모리를 무엇으로 쓰느냐에 따라 ‘램 기반 SSD’와 ‘플래시메모리 기반 SSD’로 나뉜다.

램 기반 SSD는 매우 빠른 속도를 발휘하는데, 이것을 장착한 컴퓨터는 전원을 켜 후 1~2초 만에 윈도우 운영체제의 부팅을 끝낼 수 있을 정도다. 다만 램은 전원이 꺼지면 저장 데이터가 모두 사라지기 때문에 컴퓨터의 전원을 끈 상태에서도 SSD에 계속해서 전원을 공급해 주는 전용 전지가 반드시 필요하다. 이런 단점 때문에 램 기반 SSD는 많이 쓰이지 않는다.

그래서 일반적으로 SSD는 플래시메모리 기반 SSD를 지칭한다. 플래시메모리는 전원이 꺼지더라도 기록된 데이터가 보존되기 때문에 HDD를 쓰던 것처럼 쓰면 된다. 그리고 플래시메모리 기반 SSD를 장착한 컴퓨터는 램 기반 SSD를 장착한 컴퓨터보다 느리긴 하지만 HDD를 장착한 동급 사양의 컴퓨터보다 최소 2~3배 이상 빠른 부팅 속도와 프로그램 실행 속도를 기대할 수 있다.

* 컴퓨터 시스템: CPU, 램 등 컴퓨터를 동작시키는 장치의 집합체.

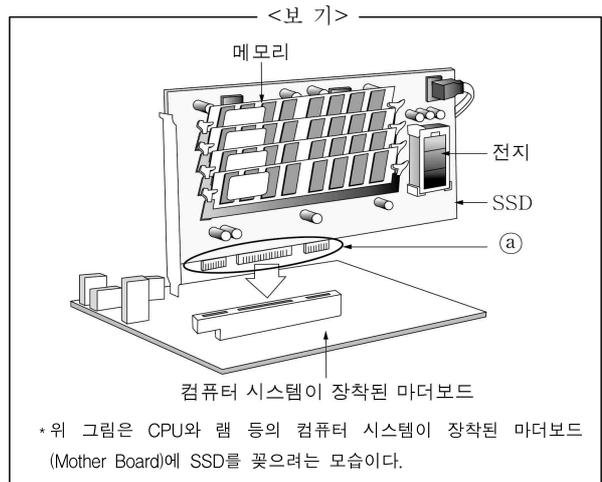
20. 밑글에서 확인할 수 있는 내용으로 적절하지 않은 것은?

- ① HDD의 발전 과정
- ② SSD의 구성 요소
- ③ 컴퓨터 속도를 결정하는 주요 장치
- ④ 램과 HDD의 데이터 처리 방식 차이
- ⑤ SSD를 휴대용 컴퓨터에 쓰면 좋은 이유

21. 밑글에 대한 이해로 적절한 것은?

- ① HDD를 설치하는 것보다 SSD를 설치하는 방법이 복잡하다.
- ② HDD는 데이터 처리 방식의 한계 때문에 속도의 향상이 더딘 편이었다.
- ③ SSD의 소음이 큰 이유는 데이터를 읽을 때 자기디스크가 회전하기 때문이다.
- ④ 운영체제를 빠르게 쓰고 싶다면, SSD보다 HDD를 보조기억장치로 쓰는 것이 낫다.
- ⑤ 전자를 움직여 데이터를 읽는 것보다 자기디스크를 움직여 데이터를 읽는 것이 전력을 적게 쓴다.

22. 밑글을 바탕으로 <보기>에 대해 이해한 것으로 적절하지 않은 것은? [3점]



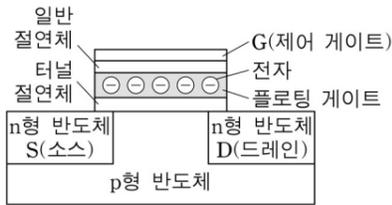
- ① <보기>의 SSD에는 컨트롤러와 버퍼 메모리 장치가 있다.
- ② a는 SSD가 컴퓨터 시스템과 데이터를 주고받는 부분이다.
- ③ <보기>의 SSD는 전지가 있는 것으로 보아 일반적으로 쓰이는 것이다.
- ④ <보기>의 SSD는 다른 종류의 SSD에 비해 데이터 처리 속도가 빠르다.
- ⑤ <보기>의 SSD에 전지가 없다면 컴퓨터 전원이 꺼졌을 때 메모리에 있는 데이터가 다 지워질 것이다.

◆ 14-6평 A형 19~21번

[19~21] 다음 글을 읽고 물음에 답하십시오.

플래시 메모리는 수많은 스위치들로 이루어지는데, 각 스위치에 0 또는 1을 저장한다. 디지털 카메라에서 사진 한 장은 수백만 개 이상의 스위치를 켜고 끄는 방식으로 플래시 메모리에 저장된다. 메모리에서는 1비트의 정보를 기억하는 이 스위치를 셀이라고 한다. 플래시 메모리에서 셀은 그림과 같은 구조의 트랜지스터 1개로 이루어져 있다. 플로팅 게이트에 전자가 들어 있는 상태를 1, 들어 있지 않은 상태를 0이라고 정의한다.

플래시 메모리에서 데이터를 읽을 때는 그림의 반도체 D에 3V의 양(+의) 전압을 가한다. 그러면 다른 한 쪽의 반도체인 S로부터 전자들이 D 쪽으로



이끌리게 된다. 플로팅 게이트에 전자가 들어 있을 때는 S로부터 오는 전자와 플로팅 게이트에 있는 전자가 마치 자석의 같은 극처럼 서로 반발하기 때문에 전자가 흐르기 힘들다. 한편 플로팅 게이트에 전자가 없는 상태에서는 S와 D 사이에 전자가 흐르기 쉽다. 이렇게 전자의 흐름 여부, 즉 S와 D 사이에 전류가 흐르는가로 셀의 값이 1인지 0인지를 판단한다.

플래시 메모리에서는 두 가지 과정을 거쳐 데이터가 저장된다. 일단 데이터를 지우는 과정이 필요하다. 데이터 지우기는 여러 개의 셀이 연결된 블록 단위로 이루어진다. 블록에 포함된 모든 셀마다 G에 0V, p형 반도체에 약 20V의 양의 전압을 가하면, 플로팅 게이트에 전자가 있는 경우, 그 전자가 터널 절연체를 넘어 p형 반도체로 이동한다. 반면 전자가 없는 경우는 플로팅 게이트에 변화가 없다. 따라서 해당 블록의 모든 셀은 0의 상태가 된다. 터널 절연체는 전류 흐름을 항상 차단하는 일반 절연체와는 다르게 일정 이상의 전압이 가해졌을 때는 전자를 통과시킨다.

이와 같은 과정을 거친 후에야 데이터 쓰기가 가능하다. 데이터를 저장하려면 1을 쓰려는 셀의 G에 약 20V, p형 반도체에는 0V의 전압을 가한다. 그러면 p형 반도체에 있던 전자들이 터널 절연체를 넘어 플로팅 게이트로 들어가 저장된다. 이것이 1의 상태이다.

플래시 메모리는 EPROM과 EEPROM의 장점을 취하여 만든 메모리이다. EPROM은 한 개의 트랜지스터로 셀을 구성하여 셀 면적이 작은 반면, 데이터를 지울 때 칩을 떼어 내어 자외선으로 소거해야 한다는 단점이 있다. EEPROM은 전기를 이용하여 간편하게 데이터를 지울 수 있지만, 셀 하나당 두 개의 트랜지스터가 필요하다. 플래시 메모리는 한 개의 트랜지스터로 셀을 구성하며, 전기적으로 데이터를 쓰고 지울 수 있다. 한편 메모리는 전원 차단 시에 데이터의 보존 유무에 따라 휘발성과 비휘발성 메모리로 구분되는데, 플래시 메모리는 플로팅 게이트가 절연체로 둘러싸여 있기 때문에 전원을 꺼도 1이나 0의 상태가 유지되므로 비휘발성 메모리이다. 이런 장점 때문에 휴대용 디지털 장치는 주로 플래시 메모리를 이용하여 데이터를 저장한다.

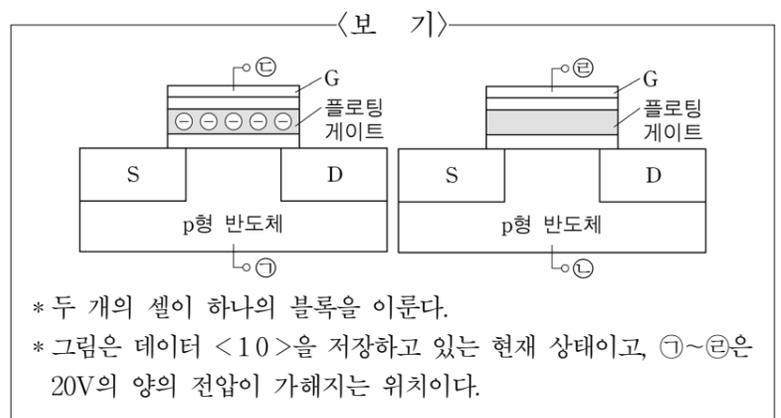
19. 윗글에 대한 설명으로 가장 적절한 것은?

- ① 대상의 구조를 바탕으로 작동 원리를 설명하고 있다.
- ② 대상의 장점을 설명한 뒤 사용 방법을 알려 주고 있다.
- ③ 대상의 크기를 기준으로 자세한 기능을 설명하고 있다.
- ④ 대상의 구성 요소를 설명한 뒤 제작 원리를 알려 주고 있다.
- ⑤ 대상의 단점을 나열하고 새로운 방식의 필요성을 제기하고 있다.

20. 윗글의 '플래시 메모리'에 대하여 추론한 내용으로 옳은 것은?

- ① D에 3V의 양의 전압을 가하면 플로팅 게이트의 전자가 사라진다.
- ② 터널 절연체 대신에 일반 절연체를 사용하면 데이터를 반복해서 지우고 쓸 수 없다.
- ③ 데이터 지우기 과정에서 자외선에 노출해야 데이터를 수정할 수 있다.
- ④ EEPROM과 비교되는 EPROM의 단점을 개선하여 셀 면적을 더 작게 만들었다.
- ⑤ 데이터를 유지하기 위해서는 전력을 계속 공급해 주어야 한다.

21. 윗글과 <보기>에 따라 플래시 메모리의 데이터 <10>을 <01>로 수정하려고 할 때, 단계별로 전압이 가해질 위치가 옳은 것은? [3점]



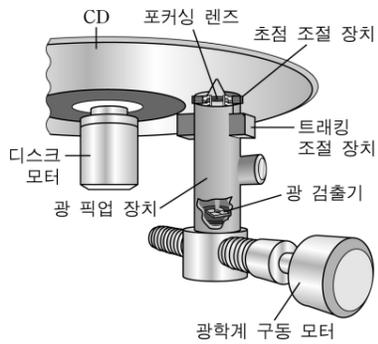
	1단계	2단계
①	㉠	㉤
②	㉡	㉣
③	㉠과 ㉣	㉤
④	㉣과 ㉡	㉤
⑤	㉡과 ㉤	㉣

◆ 14 수능 A형 28~30번

[28~30] 다음 글을 읽고 물음에 답하시오.

CD 드라이브는 디스크 표면에 조사된 레이저 광선이 반사되거나 산란되는 효과를 이용해 정보를 판독한다. CD의 기록면 중 광선이 흠어짐 없이 반사되는 부분을 랜드, 광선의 일부가 산란되어 빛이 적게 반사되는 부분을 피트라고 한다. CD에는 나선 모양으로 돌아 나가는 단 하나의 트랙이 있는데 트랙을 따라 일렬로 랜드와 피트가 번갈아 배치되어 있다. 피트를 제외한 부분, 즉 이웃하는 트랙과 트랙 사이도 랜드에 해당한다.

CD 드라이브는 디스크 모터, 광 픽업 장치, 광학계 구동 모터로 구성된다. 디스크 모터는 CD를 회전시킨다. CD 아래에 있는 광 픽업 장치는 레이저 광선을 발생시켜 CD 기록면에 조사하고, CD에서 반사된 광선은 광 픽업 장치



안의 광 검출기가 받아들인다. 광선의 경로 상에 있는 포커싱 렌즈는 광선을 트랙의 한 지점에 모으고, 광 검출기는 반사된 광선의 양을 측정하여 랜드와 피트의 정보를 읽어 낸다. 이때 CD의 회전 속도에 맞춰 트랙에 광선이 조사될 수 있도록 광학계 구동 모터가 광 픽업 장치를 CD의 중심부에서 바깥쪽으로 서서히 직선으로 이동시킨다.

CD의 고속 회전 등으로 진동이 생기면 광선의 위치가 트랙을 벗어나거나 초점이 맞지 않아 데이터를 잘못 읽을 수 있다. 이를 막으려면 트래킹 조절 장치와 초점 조절 장치를 제어해 실시간으로 편차를 보정해야 한다. 편차 보정에는 광 검출기가 사용된다. 광 검출기는 가운데를 기준으로 전후좌우의 네 영역으로 분할되어 있는데, 트랙의 방향과 같은 방향으로 전후 영역이, 직각 방향으로 좌우 영역이 배치되어 있다. 이때 각 영역에 조사되는 빛의 양이 많아지면 그 영역의 출력값도 커지며 네 영역의 출력값의 합을 통해 피트와 랜드를 구별한다.

레이저 광선이 트랙의 중앙에 초점이 맞은 상태로 정확히 조사되면 광 검출기 네 영역의 출력값은 모두 동일하다. 그런데 광선이 피트에 해당하는 지점에 조사될 때 트랙의 중앙을 벗어나 좌측으로 치우치면, 피트 왼편에 있는 랜드에서 반사되는 빛이 많아져 광 검출기의 좌 영역의 출력값이 우 영역보다 커진다. 이 경우 두 출력값의 차이에 대응하는 만큼 트래킹 조절 장치를 작동하여 광 픽업 장치를 오른쪽으로 움직여서 편차를 보정한다. 우측으로 치우쳐 조사된 경우에도 비슷한 과정을 거쳐 편차를 보정한다.

한편 광 검출기에 조사되는 광선의 모양은 초점의 상태에 따라 전후나 좌우 방향으로 길어진다. CD 기록면과 포커싱 렌즈 간의 거리가 가까워져 광선의 초점이 맞지 않으면, 조사된 모양이 전후 영역으로 길어지고 출력값도 상대적으로 커진다. 반면 둘 사이의 거리가 멀어지면, 좌우 영역으로 길어지고 출력값도 상대적으로 커진다. 이때 광 검출기의 전후 영역 출력값의 합과 좌우 영역 출력값의 합을 구한 후, 그 둘의 차이에 해당하는 만큼 초점 조절 장치를 이용해 포커싱 렌즈의 위치를 CD 기록면과 가깝게 또는 멀게 이동시켜 초점이 맞도록 한다.

28. 윗글에 나타난 여러 장치에 대한 설명으로 적절하지 않은 것은?

- ① 초점 조절 장치는 포커싱 렌즈의 위치를 이동시킨다.
- ② 포커싱 렌즈는 레이저 광선을 트랙의 한 지점에 모아 준다.
- ③ 광 검출기의 출력값은 트래킹 조절 장치를 제어하는 데 사용된다.
- ④ 광학계 구동 모터는 광 픽업 장치가 CD를 따라 회전할 수 있도록 해 준다.
- ⑤ 광 픽업 장치에는 레이저 광선을 발생시키는 부분과 반사된 레이저 광선을 검출하는 부분이 있다.

29. 윗글을 이해한 내용으로 적절하지 않은 것은?

- ① CD에 기록된 정보는 중심에서부터 바깥쪽으로 읽어야 하겠군.
- ② 레이저 광선은 CD 기록면을 향해 아래에서 위쪽으로 조사 되겠군.
- ③ 광 검출기에서 네 영역의 출력값의 합은 피트를 읽을 때보다 랜드를 읽을 때 더 크게 나타나겠군.
- ④ 렌즈의 초점이 맞지 않으면 광 검출기의 전 영역과 후 영역의 출력값의 차이를 이용하여 보정하겠군.
- ⑤ CD의 고속 회전에 의한 진동으로 인해 광 검출기에 조사된 레이저 광선의 모양이 길쭉해질 수 있겠군.

30. 윗글을 바탕으로 <보기>에 대해 설명한 내용으로 적절한 것은? [3점]

<보 기>

다음은 CD 기록면의 피트 위치에 레이저 광선이 조사되었을 때 <상태 1>과 <상태 2>에서 얻은 광 검출기의 출력값이다.

영역	전	후	좌	우
상태 1의 출력값	2	2	3	1
상태 2의 출력값	5	5	3	3

- ① 광 검출기에 조사되는 레이저 광선의 총량은 <상태 1>보다 <상태 2>가 작다.
- ② <상태 1>에서는 초점 조절 장치가 구동되어야 하지만, <상태 2>에서는 구동될 필요가 없다.
- ③ <상태 1>에서는 트래킹 조절 장치가 구동될 필요가 없지만, <상태 2>에서는 구동되어야 한다.
- ④ <상태 1>에서는 레이저 광선이 트랙의 오른쪽에 치우쳐 조사되고, <상태 2>에서는 가운데 조사된다.
- ⑤ <상태 1>에서는 포커싱 렌즈와 CD 기록면의 사이의 거리를 조절할 필요가 없지만, <상태 2>에서는 멀게 해야 한다.

◆ 15-9평 A형 19~21번

[19~21] 다음 글을 읽고 물음에 답하시오.

우리는 컴퓨터에서 음악을 들으면서 문서를 작성할 때 두 가지 프로그램이 동시에 실행되고 있다고 생각한다. 그러나 실제로는 아주 짧은 시간 간격으로 그 프로그램들이 번갈아 실행되고 있다. 이는 컴퓨터 운영 체제의 일부인 CPU(중앙 처리 장치) 스케줄링 때문이다. 어떤 프로그램이 실행될 때 컴퓨터 운영 체제는 실행할 프로그램을 주기억 장치에 저장하고 실행 대기 프로그램의 목록인 '작업큐'에 등록한다. 운영 체제는 실행할 하나의 프로그램을 작업큐에서 선택하여 CPU에서 실행하고 실행이 종료되면 작업큐에서 지운다.

한 개의 CPU는 한 번에 하나의 프로그램만을 실행할 수 있다. 그러면 A와 B 두 개의 프로그램이 동시에 실행되는 것처럼 보이게 하려면 어떻게 해야 할까? 프로그램은 실행을 요청한 순서대로 작업큐에 등록되고 이 순서에 따라 A와 B는 차례로 실행된다. 이때 A의 실행 시간이 길어지면 B가 기다려야 하는 '대기 시간'이 길어지므로 동시에 두 프로그램이 실행되고 있는 것처럼 보이지 않는다. 그러나 A와 B를 일정한 시간 간격을 두고 번갈아 실행하면 두 프로그램이 동시에 실행되는 것처럼 보인다.

이를 위해서 CPU의 실행 시간을 여러 개의 짧은 구간으로 나누어 놓고 각각의 구간마다 하나의 프로그램이 실행되도록 한다. 여기서 한 구간에서 프로그램이 실행되는 것을 '구간 실행'이라 하며, 각각의 구간에서 프로그램이 실행되는 시간을 '구간 시간'이라고 하는데 구간 시간의 길이는 일정하게 정한다. A와 B의 구간 실행은 원칙적으로 두 프로그램이 종료될 때까지 번갈아 반복되지만 하나의 프로그램이 먼저 종료되면 나머지 프로그램이 계속 실행된다.

한편, 어떤 프로그램의 구간 실행이 진행되는 동안, 다른 프로그램은 작업큐에서 대기한다. A의 구간 실행이 끝나면 A의 실행이 정지되고 다음번 구간 시간 동안 실행할 프로그램을 선택한다. 이때 A가 정지한 후 B의 실행을 준비하는 데 필요한 시간을 '교체 시간'이라고 하는데 교체 시간은 구간 시간에 비해 매우 짧다. 교체 시간에는 그때까지 실행된 A의 상태를 저장하고 B를 실행하기 위해 B의 이전 상태를 가져온다. 그뿐만 아니라 같은 프로그램이 이어서 실행되더라도 운영 체제가 다음에 실행되어야 할 프로그램을 판단해야 하므로 구간 실행 사이에는 반드시 교체 시간이 필요하다.

하나의 프로그램이 작업큐에 등록될 때부터 종료될 때까지 걸리는 시간을 '총처리 시간'이라고 하는데 이 시간은 순수하게 프로그램의 실행에만 소요된 시간인 '총실행 시간'에 '교체 시간'과 작업큐에서 실행을 기다리는 '대기 시간'을 모두 합한 것이다. ㉠ 총실행 시간이 구간 시간보다 긴 프로그램이 실행될 때는 구간 실행 횟수가 많아져서 교체 시간의 총합은 늘어난다. 그러나 총실행 시간이 구간 시간보다 짧거나 같은 프로그램은 한 번의 구간 시간 내에 종료되고 곧바로 다음 프로그램이 실행된다.

이제 프로그램 A, B, C가 실행되는 경우를 생각해 보자. A가 실행되고 있고 B가 작업큐에서 대기 중인 상태에서 새로운 프로그램 C를 실행할 경우, C는 B 다음에 등록되므로 A와 B

의 구간 실행이 끝난 후 C가 실행된다. A와 B가 종료되지 않아 추가적인 구간 실행이 필요하면 작업큐에서 C의 뒤로 다시 등록되므로 C, A, B의 상태가 되고 결과적으로 세 프로그램은 등록되는 순서대로 반복해서 실행된다.

이처럼 작업큐에 등록된 프로그램의 수가 많아지면 각 프로그램의 대기 시간은 그에 비례하여 늘어난다. 따라서 작업큐에 등록할 수 있는 프로그램의 수를 제한해 대기 시간이 일정 수준 이상으로 길어지는 것을 막을 필요가 있다.

19. 윗글의 내용과 일치하지 않는 것은?

- ① CPU 스케줄링은 컴퓨터 운영 체제의 일부이다.
- ② 프로그램 실행이 종료되면 실행 결과는 작업큐에 등록된다.
- ③ 구간 실행의 교체에 소요되는 시간은 구간 시간보다 짧다.
- ④ CPU 한 개는 한 번에 하나의 프로그램만 실행이 가능하다.
- ⑤ 컴퓨터 운영 체제는 실행할 프로그램을 주기억 장치에 저장한다.

20. ㉠의 실행 과정에 대한 이해로 적절하지 않은 것은?

- ① 교체 시간이 줄어들면 총처리 시간이 줄어든다.
- ② 대기 시간이 늘어나면 총처리 시간이 늘어난다.
- ③ 총실행 시간이 줄어들면 총처리 시간이 줄어든다.
- ④ 구간 시간이 늘어나면 구간 실행 횟수는 늘어난다.
- ⑤ 작업큐의 프로그램 개수가 늘어나면 총처리 시간은 늘어난다.

21. 윗글을 바탕으로 할 때, <보기>의 [가]에 들어갈 내용으로 적절한 것은? [3점]

—<보 기>—

운영 체제가 작업큐에 등록된 프로그램에 대해 우선순위를 부여하고 순위가 가장 높은 것을 다음에 실행할 프로그램으로 선택하면 작업큐의 크기를 제한하지 않고도 각 프로그램의 '대기 시간'을 조절할 수 있다.

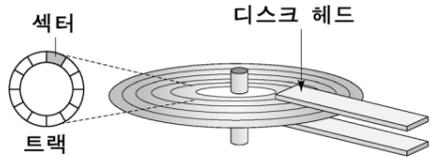
프로그램 P, Q, R이 실행되고 있는 예를 생각해 보자. P가 '구간 실행' 상태이고 Q와 R이 작업큐에 대기 중이며 Q의 순위가 R보다 높다. P가 구간 실행을 마치고 작업큐에 재등록될 때, P의 순위를 Q보다는 낮지만 R보다는 높게 한다. P가 작업큐에 재등록된 후 다시 P가 구간 실행을 하기 직전까지 [가] 을/를 거쳐야 한다.

- ① P에서 R로의 교체
- ② Q의 구간 실행
- ③ Q의 구간 실행과 R의 구간 실행
- ④ Q의 구간 실행과 Q에서 P로의 교체
- ⑤ R의 구간 실행과 R에서 P로의 교체

◆ 13-6평 44~46번

[44~46] 다음 글을 읽고 물음에 답하시오.

하드 디스크는 고속으로 회전하는 디스크의 표면에 데이터를 저장한다. 데이터는 동심원으로 된 트랙에 저장되는데, 하드 디스크는 트랙을 여러 개의 섹터로 미리 구획하고, 트랙을 오가는 헤드를 통해 섹터 단위로 읽기와 쓰기를 수행한다. 하드 디스크에서 데이터 입출력 요청을 완료하는 데 걸리는 시간을 접근 시간이라고 하며, 이는 하드



디스크의 성능을 결정하는 기준 중 하나가 된다. 접근 시간은 원하는 트랙까지 헤드가 이동하는 데 소요되는 탐색 시간과, 트랙 위에서 해당 섹터가 헤드의 위치까지 회전해 오는 데 걸리는 대기 시간의 합이다. 하드 디스크의 제어기는 '디스크 스케줄링'을 통해 접근 시간이 최소가 되도록 한다.

㉠ 200개의 트랙이 있고 가장 안쪽의 트랙이 0번인 하드 디스크를 생각해 보자. 현재 헤드가 54번 트랙에 있고 대기 큐*에는 '99, 35, 123, 15, 66' 트랙에 대한 처리 요청이 들어와 있다고 가정하자. 요청 순서대로 데이터를 처리하는 방법을 FCFS 스케줄링이라 하며, 이때 헤드는 '54 → 99 → 35 → 123 → 15 → 66'과 같은 순서로 이동하여 데이터를 처리하므로 헤드의 총 이동 거리는 356이 된다.

만일 헤드가 현재 위치로부터 이동 거리가 가장 가까운 트랙 순서로 이동하면 '54 → 66 → 35 → 15 → 99 → 123'의 순서가 되므로, 이때 헤드의 총 이동 거리는 171로 줄어든다. 이러한 방식을 SSTF 스케줄링이라 한다. 이 방법을 사용하면 FCFS 스케줄링에 비해 헤드의 이동 거리가 짧아 탐색 시간이 줄어든다. 하지만 현재 헤드 위치로부터 가까운 트랙에 대한 데이터 처리 요청이 계속 들어오면 먼 트랙에 대한 요청들의 처리가 계속 미뤄지는 문제가 발생할 수 있다.

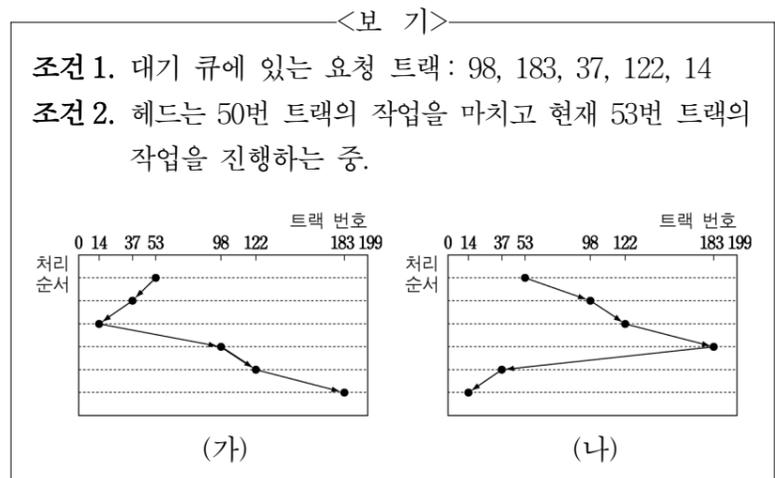
이러한 SSTF 스케줄링의 단점을 개선한 방식이 SCAN 스케줄링이다. SCAN 스케줄링은 헤드가 디스크의 양 끝을 오가면서 이동 경로 위에 포함된 모든 대기 큐에 있는 트랙에 대한 요청을 처리하는 방식이다. 위의 예에서 헤드가 현재 위치에서 트랙 0번 방향으로 이동한다면 '54 → 35 → 15 → 0 → 66 → 99 → 123'의 순서로 처리되며, 이때 헤드의 총 이동 거리는 177이 된다. 이 방법을 쓰면 현재 헤드 위치에서 멀리 떨어진 트랙이라도 최소한 다음 이동 경로에는 포함되므로 처리가 지나치게 늦어지는 것을 막을 수 있다. SCAN 스케줄링을 개선한 LOOK 스케줄링은 현재 위치로부터 이동 방향에 따라 대기 큐에 있는 트랙의 최솟값과 최댓값 사이에서만 헤드가 이동함으로써 SCAN 스케줄링에서 불필요하게 양 끝까지 헤드가 이동하는 데 걸리는 시간을 없애 탐색 시간을 더욱 줄인다.

* 대기 큐: 하드 디스크에 대한 데이터 입출력 처리 요청을 임시로 저장하는 곳.

44. 위 글의 내용과 일치하지 않는 것은?

- ① 데이터에 따라 트랙당 섹터의 수가 결정된다.
- ② 헤드의 이동 거리가 늘어나면 탐색 시간도 늘어난다.
- ③ 디스크 스케줄링은 데이터들의 처리 순서를 결정한다.
- ④ 대기 시간은 하드 디스크의 회전 속도에 영향을 받는다.
- ⑤ 접근 시간은 하드 디스크의 성능을 평가하는 척도 중 하나이다.

45. <보기>는 주어진 조건에 따라 ㉠에서 헤드가 이동하는 경로를 나타낸 것이다. (가), (나)에 해당하는 스케줄링 방식으로 적절한 것은?



- | | (가) | (나) |
|---|------|------|
| ① | FCFS | SSTF |
| ② | SSTF | SCAN |
| ③ | SSTF | LOOK |
| ④ | SCAN | LOOK |
| ⑤ | LOOK | SCAN |

46. 헤드의 위치가 트랙 0번이고 현재 대기 큐에 있는 요청만을 처리한다고 할 때, 각 스케줄링의 탐색 시간의 합에 대한 비교로 옳은 것은? [3점]

- ① 요청된 트랙 번호들이 내림차순이면, SSTF 스케줄링과 LOOK 스케줄링에서 탐색 시간의 합은 같다.
- ② 요청된 트랙 번호들이 내림차순이면, FCFS 스케줄링이 SSTF 스케줄링보다 탐색 시간의 합이 작다.
- ③ 요청된 트랙 번호들이 오름차순이면, FCFS 스케줄링과 LOOK 스케줄링에서 탐색 시간의 합은 다르다.
- ④ 요청된 트랙 번호들이 오름차순이면, FCFS 스케줄링이 SCAN 스케줄링보다 탐색 시간의 합이 크다.
- ⑤ 요청된 트랙 번호들에 끝 트랙이 포함되면, LOOK 스케줄링이 SCAN 스케줄링보다 탐색 시간의 합이 크다.

[32~36] 다음 글을 읽고 물음에 답하시오.

컴퓨터의 중앙처리장치인 CPU는 데이터를 처리하기 위해 주기억장치와 끊임없이 데이터를 주고받는다. 그런데 CPU는 처리 속도가 매우 빠른 반면, 주기억장치의 처리 속도는 상대적으로 느리다. 그렇기 때문에 CPU가 명령을 실행할 때마다 주기억장치로부터 데이터를 읽어 오면 두 장치의 처리 속도의 차이로 인해 명령을 빠르게 실행할 수가 없다. 그래서 캐시 기억장치를 활용하여 데이터 처리 속도를 향상시킨다. 캐시 기억장치는 CPU 내에 또는 CPU와 주기억장치 사이에 위치한 기억장치로 주기억장치보다 용량은 작지만 처리 속도가 매우 빠르다. 이러한 캐시 기억장치에 주기억장치의 데이터 중 자주 사용되는 데이터의 일부를 복사해 두고 CPU가 이 데이터를 사용하도록 하는 과정을 ‘캐싱(caching)’이라고 한다.

캐싱이 효율적으로 이루어지려면 CPU가 캐시 기억장치에 저장된 데이터를 반복적으로 사용하는 것이 중요한데 이를 위해 고려되는 것이 참조의 지역성이다. 참조의 지역성은 시간적 지역성과 공간적 지역성으로 나눌 수 있다. 시간적 지역성은 CPU가 한 번 사용한 특정 데이터가 가까운 미래에 다시 사용될 가능성이 높은 것을 말하고, 공간적 지역성은 한 번 사용한 데이터 근처에 있는 데이터가 곧 사용될 가능성이 높은 것을 말한다.

한편 주기억장치는 ‘워드(word)’ 단위로 데이터가 저장되고 캐시 기억장치는 ‘블록(block)’ 단위로 데이터가 저장된다. 이때 워드는 비트(bit)*의 집합이고 블록은 연속된 워드 여러 개의 묶음을 말한다. 주기억장치의 데이터가 캐시 기억장치에 저장되는 장소를 ‘라인(line)’이라고 한다. 캐시 기억장치는 일반적으로 하나의 라인에 하나의 블록이 들어갈 수 있도록 설계되어 있기 때문에 주기억장치에서 캐시 기억장치로 데이터를 전송할 때에는 블록 단위로 데이터를 전송한다. 캐시 기억장치의 용량은 주기억장치보다 훨씬 작기 때문에 주기억장치의 블록 중에서 일부만 캐시 기억장치에 저장될 수 있다. 그러므로 캐싱을 위해서는 주기억장치의 여러 블록이 캐시 기억장치의 하나의 라인을 공유하여 사용해야 한다.

예를 들어 어떤 컴퓨터의 주기억장치의 데이터 용량을 워드 2^n 개, 캐시 기억장치의 데이터 용량을 워드 M 개라고 가정해 보자. 이때 주기억장치의 블록 한 개가 K 개의 워드로 이루어져 있다고 하면 이 주기억장치의 총 블록 개수는 $2^n/K$ 개가 되며 각 워드는 n 비트의 주소로 지정된다. 그리고 캐시 기억장치의 각 라인은 K 개의 워드로 채워지므로 캐시 기억장치에는 총 M/K 개의 라인이 만들어진다.

캐싱이 이루어질 때 CPU가 요청한 데이터가 캐시 기억장치에 있는지 여부를 확인하고 해당 데이터를 불러오기 위해 주기억장치의 데이터 주소가 사용된다. 이 주소는 ‘태그 필드, 라인 필드, 워드 필드’의 형식으로 구성되어 있는데 ‘태그 필드’는 캐시 기억장치의 특정 라인에 주기억장치의 어떤 블록이 저장되어 있는지를 구분해 주는 역할을 한다. 그리고 ‘라인 필드’는 주기억장치의 블록이 들어갈 캐시 기억장치의 라인을 지정해 주며, ‘워드 필드’는 주기억장치의 각 블록에 저장되어 있는 워드를 지정해 준다.

주기억장치의 데이터를 캐시 기억장치에 저장하는 방식에는 여러 가지가 있는데 그중 하나가 ㉠ ‘직접 매핑’이다. 직접 매핑은 주기억장치의 데이터를 블록 단위로 캐시 기억장치

의 지정된 라인에 저장하는 방식이다. 직접 매핑 방식에서 캐싱이 이루어지는 과정은 다음과 같다. CPU가 ‘태그 필드, 라인 필드, 워드 필드’로 이루어진 주소를 통해 데이터를 요청하면, 우선 요청 주소의 라인 필드를 이용하여 캐시 기억장치의 해당 라인을 확인한다. 그리고 해당 라인에 데이터가 저장되어 있으면 그 라인의 태그와 요청 주소의 태그를 비교한다. 이때 두 태그의 값이 일치하는 경우를 ‘캐시 히트(cache hit)’라고 하며, 캐시 히트가 일어나면 주소의 워드 필드를 이용하여 라인 내 워드들 중에서 해당 데이터를 찾아 CPU에 보내 준다. 그런데 CPU가 요청한 주소의 태그와 캐시 기억장치 라인의 태그가 일치하지 않거나 해당 라인이 비어 있어서 요청한 데이터를 찾지 못하는 경우가 있다. 이는 CPU가 요청한 데이터가 캐시 기억장치에 저장되어 있지 않다는 의미로, 이 경우를 ‘캐시 미스(cache miss)’라고 한다. 캐시 미스가 일어나면 요청 주소에 해당하는 블록을 주기억장치에서 복사하여 캐시 기억장치의 지정된 라인에 저장한다. 그리고 주소의 태그를 그 라인의 태그 필드에 기록하고 요청된 데이터를 CPU에 보내 준다. 만약 그 라인에 다른 블록이 저장되어 있다면 그 블록은 지워지고 새롭게 가져온 블록이 저장된다.

[B]

직접 매핑은 CPU가 요청한 데이터가 캐시 기억장치에 있는지 확인할 때 해당 라인만 검색하면 되기 때문에 검색 속도가 빠르다. 그리고 회로의 구조가 단순하여 시스템을 구성하는 비용이 저렴한 장점이 있다. 하지만 같은 라인에 저장되어야 하는 서로 다른 블록을 CPU가 번갈아 요청하는 경우, 계속 캐시 미스가 발생해서 반복적으로 블록이 교체되므로 시스템의 효율이 ㉡ 떨어질 수 있다. 그래서 캐시 기억장치의 라인 어디에나 자유롭게 블록을 저장하는 ‘완전 연관 매핑’, 직접 매핑과 완전 연관 매핑을 혼합한 ‘세트 연관 매핑’ 등을 활용하기도 한다.

* 비트: 컴퓨터에서 정보를 나타내는 가장 기본적인 단위. 2진수의 0 또는 1이 하나의 비트.

32. 윗글의 내용과 일치하는 것은?

- ① 캐시 기억장치의 하나의 라인에는 하나의 워드만 저장될 수 있다.
- ② 캐시 기억장치는 주기억장치보다 용량이 크고 처리 속도가 느리다.
- ③ 캐시 기억장치에 저장된 데이터가 반복적으로 사용되어야 캐싱의 효율이 높아진다.
- ④ 시간적 지역성은 한 번 사용된 데이터 근처에 있는 데이터가 곧 사용될 가능성이 높은 것을 말한다.
- ⑤ 캐싱은 캐시 기억장치의 데이터 중 자주 사용되는 데이터의 일부를 주기억장치에 복사하여 사용하는 것을 말한다.

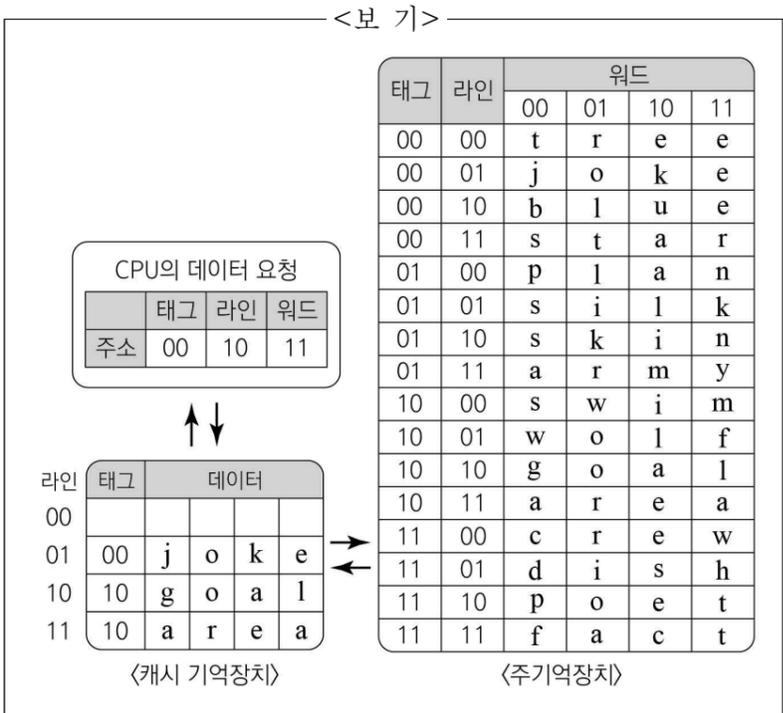
33. [A]를 참고할 때 <보기>의 ㉠~㉣에 들어갈 말을 바르게 짝 지은 것은?

<보 기>

주기억장치의 데이터 용량이 64개의 워드이고, 하나의 블록이 4개의 워드로 이루어져 있다면, 주기억장치는 총 16개의 (㉠)으로 구성되며, 각 워드는 (㉡)의 주소로 지정된다. 또한 캐시 기억장치의 데이터 용량이 16개의 워드라면 캐시 기억장치의 라인은 (㉢)가 만들어진다.

- | | | | |
|---|----|-----|----|
| | ㉠ | ㉡ | ㉢ |
| ① | 블록 | 6비트 | 4개 |
| ② | 블록 | 8비트 | 6개 |
| ③ | 워드 | 8비트 | 4개 |
| ④ | 라인 | 6비트 | 4개 |
| ⑤ | 라인 | 8비트 | 6개 |

34. <보기>는 '직접 매핑' 과정을 도식화한 것이다. [B]를 바탕으로 <보기>를 이해한 내용으로 적절하지 않은 것은? [3점]



- ① 요청된 주소의 '10'을 이용하여 캐시 기억장치의 라인을 확인한 후 태그 '00'이 그 라인의 태그와 일치하는지 확인하겠군.
- ② CPU가 요청한 데이터가 캐시 기억장치에 저장되어 있지 않으므로 캐시 미스가 일어나겠군.
- ③ 주기억장치의 데이터 블록 중에서 'b, l, u, e'가 복사되어 캐시 기억장치에 저장되겠군.
- ④ 캐시 기억장치의 라인 '01'에 저장되어 있는 데이터 블록이 삭제되겠군.
- ⑤ CPU의 데이터 요청에 의해 최종적으로 CPU로 보내지는 데이터는 'e'가 되겠군.

35. ㉠과 <보기>의 ㉡을 비교한 내용으로 가장 적절한 것은?

<보 기>

㉡완전 연관 매핑은 캐시 기억장치에 블록을 저장할 때 라인을 지정하지 않고 임의로 저장하는 방식이다. 이 방식은 필요한 데이터 위주로 저장할 수 있기 때문에 매핑 방식 중에 캐시 히트의 확률이 가장 높다. 그러나 히트 여부 확인이 모든 라인에 걸쳐 이루어져야 하므로 검색 시간이 가장 오래 걸린다. 그리고 회로의 구조가 복잡해서 시스템을 구성하는 비용이 높다. 주기억장치의 블록이 캐시 기억장치의 정해진 라인에 저장되는 것이 아니기 때문에 주기억장치의 주소는 태그 필드, 워드 필드로 이루어진다. 대신 블록이 교체될 때 어떤 블록을 삭제할지를 결정하는 블록 교체 알고리즘이 별도로 필요하다.

- ① ㉠과 달리 ㉡은 주기억장치의 주소에 태그 필드가 있다.
- ② ㉠과 달리 ㉡은 캐시 히트 여부를 확인하는 시간이 빠르다.
- ③ ㉡과 달리 ㉠은 블록 교체 알고리즘이 필요하다.
- ④ ㉡과 달리 ㉠은 라인을 지정하여 블록을 저장한다.
- ⑤ ㉠과 ㉡은 모두 회로의 구조가 복잡하다.

36. 문맥상 의미가 ㉠과 가장 가까운 것은?

- ① 엔진의 성능이 떨어져서 큰일이다.
- ② 소매에서 단추가 떨어져서 당황했다.
- ③ 감기가 떨어지지 않아 큰 고생을 했다.
- ④ 해가 떨어지기 전에 이 일을 마치기로 했다.
- ⑤ 굵은 빗방울이 머리에 한두 방울씩 떨어지기 시작했다.

27. 윗글을 바탕으로 ㉠~㉣에 대해 이해한 내용으로 적절한 것은?

- ① ㉠은 튜링 기계의 현재 상태와 다음 상태가 다르게 지정되어 있다.
- ② ㉡는 튜링 기계의 헤드가 읽는 기호와 기록할 기호가 동일하게 지정되어 있다.
- ③ ㉢와 ㉣는 튜링 기계의 헤드가 읽는 기호가 동일하게 지정되어 있다.
- ④ ㉠와 ㉡는 튜링 기계의 헤드가 기록할 기호가 다르게 지정되어 있다.
- ⑤ ㉠와 ㉡는 튜링 기계의 헤드가 이동할 방향이 동일하게 지정되어 있다.

28. 윗글과 [1진법의 덧셈을 하는 튜링 기계의 시작 모습]을 바탕으로 ㉠~㉣에 대해 이해한 내용으로 적절하지 않은 것은? [3점]

- ① ㉠에서 튜링 기계의 상태가 X일 때, ㉡에 따라 헤드는 오른쪽으로 한 칸 이동하고 기계는 상태를 유지하게 되겠군.
- ② ㉢에서 튜링 기계의 상태가 X일 때, ㉣에 따라 헤드는 빈칸에 1을 기록하고 기계는 상태를 바꾸게 되겠군.
- ③ ㉢에서 튜링 기계의 상태가 Y일 때, ㉣에 따라 헤드는 오른쪽으로 한 칸 이동하고 기계는 상태를 유지하게 되겠군.
- ④ ㉠에서 튜링 기계의 상태가 Z일 때, ㉡에 따라 헤드는 테이프에 기록된 1을 지우고 기계는 상태를 바꾸게 되겠군.
- ⑤ ㉢에서 튜링 기계의 상태가 Y일 때, ㉣에 따라 헤드는 왼쪽으로 한 칸 이동하고 기계는 상태를 바꾸게 되겠군.

29. 문맥상 ㉠~㉣와 바꾸어 쓰기에 적절하지 않은 것은?

- ① ㉠: 생각해 내게
- ② ㉡: 이루어진다
- ③ ㉢: 짜느냐에
- ④ ㉣: 퍼뜨려서
- ⑤ ㉣: 이바지한

◆ 22 사관학교 1차 15~18번

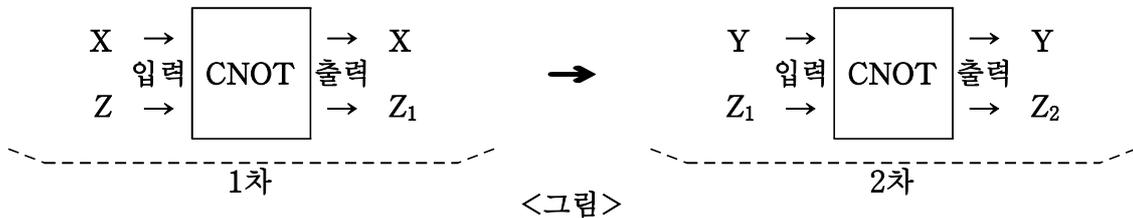
[15 ~ 18] 다음 글을 읽고 물음에 답하시오.

미시 세계에서 양자의 중첩은 여러 상태들이 겹쳐 있는 것이다. 이 중 특정 상태가 측정될 확률은 알 수 있지만 관측하기 전까지는 어떤 상태인지 정확히 알 수 없다. 어떤 상태인지 알기 위해서는 관측을 해야 하는데, 거시 세계의 관측 행위로 인해 양자의 중첩이 붕괴되어 중첩 상태 중 어느 한 상태로 확정된다. 예를 들어, 미시 세계의 철수는 앉아 있거나 서 있는 두 가지 상태가 중첩되어 있다면, 관측 이전에는 서 있는 철수가 관측될 확률과 앉아 있는 철수가 관측될 확률만 알 수 있다. 이처럼 미시 세계의 철수는 서 있는 상태와 앉아 있는 상태가 중첩되어 있는데, 이 중첩 상태는 거시 세계의 선생님이 관측했을 때 붕괴되어 비로소 서 있는 상태나 앉아 있는 상태 중 어느 한 상태로 확정된다.

양자 컴퓨터는 이러한 양자의 중첩 현상을 활용한 것이다. 기존의 컴퓨터는 정보의 기본 단위인 비트를 사용하며, 비트는 0 또는 1이라는 확정적인 값을 갖는다. 이와 달리 양자 컴퓨터는 큐비트를 사용하며, 큐비트는 0일 확률과 1일 확률을 가진 중첩된 상태를 갖는다. 따라서 기존 컴퓨터는 확정된 값을 입력해서 한 번에 하나씩 연산하여 확정된 값을 출력하지만, 양자 컴퓨터는 중첩된 큐비트를 한 번 입력함으로써, 중첩 상태가 가질 수 있는 모든 가능한 경우에 대한 연산을 한꺼번에 수행한다. 이런 원리상 양자 컴퓨터는 기존 컴퓨터보다 처리 속도가 빠르다.

기존 컴퓨터와 양자 컴퓨터는 오류를 검증하는 방법에도 차이가 있다. 기존 컴퓨터는 데이터를 저장할 때 동일 비트를 세 번 이상 저장한 후 다수결의 원리에 따라 오류를 보정할 수 있다. 일반적인 컴퓨터는 정상 비트일 가능성이 오류 비트일 가능성보다 높기 때문에, 0을 000으로, 1을 111로 저장해두면 그것이 설령 010, 001, 011, 101로 읽힌다고 하더라도 ㉠ 각각의 원래 값을 추정하기는 어렵지 않은 것이다. 그러나 양자 컴퓨터에서는 이러한 방식을 쓸 수 없다. 왜냐하면 중첩된 큐비트를 복사하거나 비교하려고 그 값을 관측하는 순간 중첩이 붕괴되기 때문이다.

이러한 문제를 해결하기 위해 양자 컴퓨터는 논리 게이트 CNOT(Controlled-NOT)을 통해 오류를 검증한다. 중첩된 큐비트가 오류일 가능성보다 정상일 가능성이 높은 양자 컴퓨터에서, 같은 내용의 연산을 통해 도출된 중첩된 큐비트 X와 중첩된 큐비트 Y가 있다고 하자. 연산에 오류가 없다면 X와 Y의 관측값은 같아야 한다. 관측은 중첩된 두 큐비트를 읽지 않고도 서로 같은지 다른지를 어떻게 판정할 수 있는가이다. <그림>의 CNOT에는 위아래로 두 개의 입력과 두 개의 출력이 있는데, 위의 입력은 입력된 그대로 출력된다. 위의 입력에 0이 들어오면 아래의 입력은 그대로 출력되며, 위의 입력에 1이 들어오면 아래의 입력이 반전되어 출력된다. 반전이 되는 경우, 0은 1로 1은 0으로 바뀐다.



먼저 큐비트 Z는 0으로 고정한다. 그리고 X와 Z를 CNOT에 입력한다. X가 1인 경우 Z를 반전하여 출력한다. 출력된 값을 각각 X와 Z₁이라고 하자. 다음으로, Y와 Z₁을 CNOT에 입력한다. Y가 1인 경우 Z₁을 반전하여 출력한다. 출력된 값을 각각 Y와 Z₂라고 하자. 만약 Z₂가 0이라면 X와 Y는 같고, 이는 연산에 오류가 있을 가능성이 낮다는 것을 의미한다. 반면 Z₂가 1이라면 X와 Y는 다르며, 이는 어디엔가 오류가 있음을 의미한다. 이러한 오류 검증의 과정에서 X와 Y의 값을 관측하지 않는다.

15. 밑글에 대한 설명으로 가장 적절한 것은? [3점]

- ① 상황을 가정하여 특정 대상의 향후 전망을 제시하고 있다.
- ② 특정 대상이 발전되어 온 과정을 통시적으로 설명하고 있다.
- ③ 다른 대상과의 차이점을 바탕으로 특정 대상을 설명하고 있다.
- ④ 서로 다른 이론을 적용하여 특정 대상의 장단점을 분석하고 있다.
- ⑤ 유사한 두 대상의 공통점을 제시한 후 각각의 의의를 서술하고 있다.

16. 밑글을 읽은 학생의 반응으로 적절하지 않은 것은? [3점]

- ① 거시 세계에서는 관측을 통해 양자의 중첩된 상태를 확인할 수 없겠군.
- ② 오류 비트일 가능성이 50% 이상이라면 다수결을 이용한 오류 검증은 실효성이 없겠군.
- ③ 미시 세계에서 중첩된 양자는 어떤 상태가 어느 정도의 확률로 있는지를 알 수 없겠군.
- ④ 기존 컴퓨터가 여러 번 수행해야 하는 연산을 양자 컴퓨터는 한 번에 수행할 수 있겠군.
- ⑤ 기존 컴퓨터의 오류 검증에서 동일 비트의 저장 횟수를 늘리면 검증의 정확도가 올라가겠군.

17. 문맥을 고려할 때 ㉠을 추정한 내용으로 적절한 것은? [3점]

- ① 0, 0, 0, 1 ② 0, 0, 1, 1 ③ 0, 1, 1, 1 ④ 1, 0, 1, 0 ⑤ 1, 1, 0, 0

18. 밑글을 바탕으로 <보기>를 이해한 내용으로 적절하지 않은 것은? [4점]

— < 보 기 > —

다음은 오류 검증을 위해 사용한 CNOT 게이트의 진리표*를 간략하게 제시한 것이다.

	X	Y	Z	Z ₁	Z ₂
㉠	(㉠)	0	0	0	0
㉡	0	1	0	0	(㉡)
㉢	1	0	0	1	(㉢)
㉣	(㉣)	1	0	1	0

* 진리표 : 논리연산의 가능한 모든 입력값의 조합을 열거하고, 각각의 조합에 대한 출력값을 나타내는 표.

- ① ㉠과 ㉣행의 Z₂가 각각 0이므로 ㉠은 0, ㉣은 1이다.
- ② ㉡과 ㉢행에서 X와 Y가 서로 다르므로, ㉡와 ㉢는 서로 다르다.
- ③ Z₁에 영향을 미치는 것은 X이다.
- ④ 오류 검증을 위해 관측하는 것은 Z₂이다.
- ⑤ X와 Y는 서로 영향을 미치지 않는다.

- 정답: 15. ③ 16. ③ 17. ② 18. ②

[27~32] 다음을 읽고 물음에 답하시오.

고전 역학에 ㉔ 따르면, 물체의 크기에 관계없이 초기 운동 상태를 정확히 알 수 있다면 일정한 시간 후의 물체의 상태는 정확히 측정될 수 있으며, 배타적인 두 개의 상태가 공존할 수 없다. 하지만 20세기에 등장한 양자 역학에 의해 미시 세계에서는 상호 배타적인 상태들이 공존할 수 있음이 알려졌다.

미시 세계에서의 상호 배타적인 상태의 공존을 이해하기 위해, 거시 세계에서 회전하고 있는 반지름 5cm의 팽이를 생각해 보자. 그 팽이는 시계 방향 또는 반시계 방향 중 한쪽으로 회전하고 있을 것이다. 팽이의 회전 방향은 관찰하기 이전에 이미 정해져 있으며, 다만 관찰을 통해 ㉕ 알게 되는 것뿐이다. 이와 달리 미시 세계에서 전자만큼 작은 팽이 하나가 회전하고 있다고 상상해 보자. 이 팽이의 회전 방향은 시계 방향과 반시계 방향의 두 상태가 공존하고 있다. 하나의 팽이에 공존하고 있는 두 상태는 관찰을 통해서 한 가지 회전 방향으로 결정된다. 두 개의 방향 중 어떤 쪽이 결정될지는 관찰하기 이전에는 알 수 없다. 거시 세계와 달리 양자 역학이 지배하는 미시 세계에서는, 우리가 관찰하기 이전에는 상호 배타적인 상태가 공존하는 것이다. 배타적인 상태의 공존과 관찰 자체가 물체의 상태를 결정한다는 개념을 받아들이기 힘들었기 때문에, 아인슈타인은 ㉖ “당신이 달을 보기 전에는 달이 존재하지 않는 것인가?”라는 말로 양자 역학의 해석에 회의적인 태도를 취하였다.

최근에는 상호 배타적인 상태의 공존을 적용함으로써 초고속 연산을 수행하는 양자 컴퓨터에 대한 연구가 진행되고 있다. 이는 양자 역학에서 말하는 상호 배타적인 상태의 공존이 현실에서 실제로 구현될 수 있음을 잘 보여 주는 예라 할 수 있다. 미시 세계에 대한 이러한 연구 성과는 거시 세계에 대해 우리가 자연스럽게 ㉗ 지니게 된 상식적인 생각들에 근본적인 의문을 ㉘ 던진다. 이와 비슷한 의문은 논리학에서도 볼 수 있다.

고전 논리는 ‘참’과 ‘거짓’이라는 두 개의 진리치만 있는 이치 논리이다. 그리고 고전 논리에서는 어떠한 진술이든 ‘참’ 또는 ‘거짓’이다. 이는 우리의 상식적인 생각과 잘 ㉙ 들어맞는다. 그러나 프리스트에 따르면, ‘참인 진술과 ‘거짓’인 진술 이외에 ‘참인 동시에 거짓’인 진술이 있다. 이를 설명하기 위해 그는 ‘거짓말쟁이 문장’을 제시한다. 거짓말쟁이 문장을 이해하기 위해 자기 지시적 문장과 자기 지시적이지 않은 문장을 구분해 보자.

[자기 지시적 문장]은 말 그대로 자기 자신을 가리키는 문장을 말한다. 예를 들어 “이 문장은 모두 열여덟 음절로 이루어져 있다.”라는 ‘참’인 문장은 자기 자신을 가리키며 그것이 몇 음절로 이루어져 있는지 말하고 있다. 반면 “페루의 수도는 리마이다.”라는 ‘참’인 문장은 페루의 수도가 어디인지 말할 뿐 자기 자신을 가리키는 문장은 아니다.

“이 문장은 거짓이다.”는 거짓말쟁이 문장이다. 이는 ‘이 문장’이라는 표현이 문장 자체를 가리키며 그것이 ‘거짓’이라고 말하는 자기 지시적 문장이다. 그렇다면 프리스트는 왜 거짓말쟁이 문장에 ‘참인 동시에 거짓’을 부여해야 한다고 생각할까? 이에 답하기 위해 우선 거짓말쟁이 문장이 ‘참’이라고 가정해 보자. 그렇다면 거짓말쟁이 문장은 ‘거짓’이다. 왜냐하면 거짓말쟁이

문장은 자기 자신을 가리키며 그것이 ‘거짓’이라고 말하는 문장이기 때문이다. 반면 거짓말쟁이 문장이 ‘거짓’이라고 가정해 보자. 그렇다면 거짓말쟁이 문장은 ‘참’이다. 왜냐하면 그것이 바로 그 문장이 말하는 바이기 때문이다. 프리스트에 따르면 어떤 경우에도 거짓말쟁이 문장은 ‘참인 동시에 거짓’인 문장이다. 따라서 그는 거짓말쟁이 문장에 ‘참인 동시에 거짓’을 부여해야 한다고 본다. 그는 거짓말쟁이 문장 이외에 ‘참인 동시에 거짓’인 진리치가 존재함을 뒷받침하는 다양한 사례를 제시한다. 특히 그는 양자 역학에서 상호 배타적인 상태의 공존은 이 점을 시사하고 있다고 본다.

고전 논리에서는 ‘참인 동시에 거짓’인 진리치를 지닌 문장을 다룰 수 없기 때문에 프리스트는 그것도 다룰 수 있는 비고전 논리 중 하나인 LP*를 제시하였다. 그런데 LP에서는 직관적으로 호소력 있는 몇몇 추론 규칙이 성립하지 않는다. 전진 긍정 규칙을 예로 들어 생각해 보자. 고전 논리에서는 전진 긍정 규칙이 성립한다. 이는 ㉚ “P이면 Q이다.”라는 조건문과 그것의 전진인 P가 ‘참’이라면 그것의 후진인 Q도 반드시 ‘참’이 된다는 것이다. 이와 비슷한 방식으로 LP에서 전진 긍정 규칙이 성립하려면, 조건문과 그것의 전진인 P가 모두 ‘참’ 또는 ‘참인 동시에 거짓’이라면 그것의 후진인 Q도 반드시 ‘참’ 또는 ‘참인 동시에 거짓’이어야 한다. 그러나 LP에서 조건문의 전진은 ‘참인 동시에 거짓’이고 후진은 ‘거짓’인 경우, 조건문과 전진은 모두 ‘참인 동시에 거짓’이지만 후진은 ‘거짓’이 된다. 비록 전진 긍정 규칙이 성립하지는 않지만, LP는 고전 논리에 대한 근본적인 의문들에 답하기 위한 하나의 시도로서 의의가 있다.

*LP: ‘역설의 논리(Logic of Paradox)’의 약자.

27. 문맥을 고려할 때 ㉖의 의미를 추론한 내용으로 가장 적절한 것은?

- ① 많은 사람들이 항상 달을 관찰하고 있으므로 달이 존재한다.
- ② 달은 질량이 매우 큰 거시 세계의 물체이므로 관찰 여부와 상관없이 존재한다.
- ③ 달은 관찰 여부와 상관없이 존재하므로 누군가 달을 관찰하기 이전에도 존재한다.
- ④ 달은 원래부터 있었지만 우리가 관찰하지 않으면 존재 여부에 대해 말할 수 없다.
- ⑤ 달이 있을 가능성과 없을 가능성이 반반이므로 관찰 이후에 달이 있을 가능성은 반이다.

28. 윗글을 바탕으로, <보기>의 ‘양자 컴퓨터’와 ‘일반 컴퓨터’에 대해 이해한 내용으로 적절한 것은?

<보 기>

양자 컴퓨터는 여러 개의 이진수들을 단 한 번에 처리함으로써 일반 컴퓨터보다 훨씬 빠른 속도로 연산을 수행한다. 연산 속도에 영향을 미치는 다른 요소들을 배제하면, 이진수를 처리하는 횟수가 적어질수록 연산 결과를 빨리 얻을 수 있기 때문이다.

n자리 이진수를 나타내기 위해서는 n비트가 필요하고 n자리 이진수는 모두 2^n 개 존재한다. 일반 컴퓨터는 한 개의 비트에 0과 1 중 하나만을 담을 수 있어, 두 자리 이진수인 00, 01, 10, 11을 2비트를 이용하여 연산할 때 네 번에 걸쳐 처리한다. 하지만 공존의 원리를 이용하는 양자 컴퓨터는 0과 1을 하나의 비트에 동시에 담아 정보를 처리할 수 있어 두 자리 이진수를 2비트를 이용하여 연산할 때 단 한 번에 처리가 가능하다. 양자 컴퓨터는 처리할 이진수의 자릿수가 커질수록 연산 속도에서 압도적인 위력을 발휘한다.

*비트(bit): 컴퓨터가 0과 1을 이용하는 이진법으로 연산을 수행하기 위해 사용하는 최소의 정보 저장 단위.

- ① 양자 컴퓨터는 상태의 공존을 이용함으로써 연산에 필요한 비트의 수를 늘릴 수 있다.
- ② 3비트를 사용하여 세 자리 이진수를 모두 처리하려고 할 때 양자 컴퓨터는 일반 컴퓨터보다 속도가 6배 빠르다.
- ③ 한 자리 이진수를 모두 처리하기 위해 1비트를 사용한다고 할 때, 일반 컴퓨터와 양자 컴퓨터의 정보 처리 횟수는 같다.
- ④ 양자 컴퓨터의 각각의 비트에는 0과 1이 공존하고 있어 4비트로 한 번에 처리할 수 있는 네 자리 이진수의 개수는 모두 16개이다.
- ⑤ 3비트의 양자 컴퓨터가 세 자리 이진수를 모두 처리하는 속도는 6비트의 양자 컴퓨터가 여섯 자리 이진수를 모두 처리하는 속도보다 2배 빠르다.

29. [자기 지시적 문장]에 대해 이해한 내용으로 적절한 것은?

- ① “붕어빵에는 붕어가 없다.”는 자기 지시적 문장이다.
- ② “이 문장은 자기 지시적이다.”라는 자기 지시적 문장은 ‘거짓’이 아니다.
- ③ “이 문장은 거짓이다.”는 이치 논리에서 자기 지시적인 문장이 될 수 없다.
- ④ 고전 논리에서는 어떠한 자기 지시적 문장에도 진리치를 부여하지 못한다.
- ⑤ 비고전 논리에서는 모든 자기 지시적 문장에 ‘참인 동시에 거짓’을 부여한다.

30. 윗글을 통해 ㉠에 대해 적절하게 추론한 것은?

- ① LP에서 P가 ‘참인 동시에 거짓’이고 Q가 ‘거짓’이면, ㉠은 ‘거짓’이다.
- ② LP에서 ㉠과 P가 ‘참인 동시에 거짓’이면, Q도 반드시 ‘참인 동시에 거짓’이다.
- ③ LP에서 ㉠과 P가 ‘참’ 또는 ‘참인 동시에 거짓’이면, Q도 반드시 ‘참’ 또는 ‘참인 동시에 거짓’이다.
- ④ 고전 논리에서 ㉠과 P가 각각 ‘거짓’이 아닐 때, Q는 ‘거짓’이다.
- ⑤ 고전 논리에서 ㉠과 P가 ‘참’이면서 Q가 ‘거짓’인 것은 불가능하다.

31. 윗글을 바탕으로 <보기>를 이해한 내용으로 적절하지 않은 것은? [3점]

<보 기>

A는 고전 논리를 받아들이고, B는 LP를 받아들일 뿐 아니라 양자 역학에서 상호 배타적인 상태의 공존이 시사하는 바에 대한 프리스트의 입장도 받아들인다.

A와 B는 아래의 (ㄱ)~(ㄴ)에 대하여 토론을 하고 있다.

(ㄱ) 전자 e는 관찰하기 이전에 S라는 상태에 있다.
 (ㄴ) 전자 e는 관찰하기 이전에 S와 배타적인 상태에 있다.
 (ㄷ) 반지름 5cm의 팽이가 시계 방향으로 회전한다.
 (ㄹ) 반지름 5cm의 팽이가 반시계 방향으로 회전한다.

(단, (ㄱ)과 (ㄴ)의 전자 e는 동일한 전자이고 (ㄷ)과 (ㄹ)의 팽이는 동일한 팽이이다.)

- ① A는 (ㄱ)이 ‘참’이 아니라면 ‘거짓’이고, ‘참’, ‘거짓’ 외에 다른 진리치를 가질 수 없다고 주장할 것이다.
- ② B는 (ㄱ)은 ‘참인 동시에 거짓’일 수 있다고 주장하지만, (ㄴ)은 ‘참’이 아니라면 ‘거짓’이라고 주장할 것이다.
- ③ A와 B는 모두 (ㄷ)이 ‘참’일 때 (ㄹ)도 ‘참’이 되는 것은 불가능하다고 주장할 것이다.
- ④ A는 B와 달리 (ㄴ)이 ‘참인 동시에 거짓’이 될 수 없다고 주장할 것이다.
- ⑤ B는 A와 달리 (ㄹ)이 ‘참’이 아니라면 ‘참인 동시에 거짓’이라고 주장할 것이다.

32. 문맥상 ㉠~㉣과 바꾸어 쓸 수 있는 말로 적절하지 않은 것은?

- ① ㉠: 의거(依據)하면 ② ㉡: 인지(認知)하게
- ③ ㉢: 소지(所持)하게 ④ ㉣: 제기(提起)한다
- ⑤ ㉤: 부합(符合)한다