컴퓨터의 정보처리 기본 단위는 **비트**이다. 컴퓨터는 비트를 이용하여 명령어 집합에 해당하는 명령을 표현하고, 저장되는 정보 또한 암호화 (인코딩) 하여 표현한다. 각각의 비트는 0 또는 1의 값을 가지므로 n-bit는 2^n개의 정보를 표현할 수 있다.

정보를 비트로 표현하여 이를 처리하기 위해서는, 비트를 이용한 연산과정과, 이 연산을 수행할 물리적인 하드웨어가 필수적이다. 컴퓨터의 하드웨어 구조는, 계층에 따라 메모리/처리 장치->논리블럭->트랜지스터로 구성된다.
비트의 표현은 트랜지스터의 출력 전압을 이용하여 표현한다. 일반적으로 5V의 출력 전압을 논리값 1로 0V의 출력 전압을 논리값 0으로 사용한다. 실제 환경에서는 외부와의 상호작용 때문에 0V 초과 5V 미만의 전압값을 가지는 경우가 더 흔하기 때문에, 0.5V~1V정도의 여유를 두고 판단하는 것이 일반적이다. 이 트랜지스터들을 조합하면 논리블럭을 구성할 수 있고, 논리블럭을 조합하면 메모리/처리장치를 구성할 수 있다.

트랜지스터를 한데 모아, 만든 논리 블럭을 **집적 회로** 라고 한다. 집적회로에서는 여러 트랜지스터가 하나의 웨이퍼(wafer)라는 공간 안에서, 서로 정보를 주고 받으며, 비교적 큰 크기의 정보를 처리한다. 집적 회로의 정보 처리 성능은 트랜지스터의 개수에 비례하며, 따라서 한정된 면적에 더 많은 트랜지스터가 들어갈수록, 더 높은 성능을 보이게 된다. 하지만 트랜지스터의 크기가 줄어들면, 0과 1의 상태를 표현하는 전기신호를 결정하는 **채널** 또한 짧아지게 된다. 채널이 짧아지면, 누설되는 전류가 많아 전력 소모와 발열이 증가하고, 이러한 현상에 의해 출력 전압이 불안정해지고, 0과 1의 상태의 표현이 부정확해지게 된다. 이런 현상들을 **단채널 효과** 라고한다. 단채널 효과 때문에 트랜지스터의 소형화에는 한계가 존재하고, 따라서 집적 회로의 집적도에도 한계가 존재한다.집적도의 한계는 곧 장치의 정보 처리의 병렬성의 한계를 의미한다.현재 상용화된 컴퓨터는 집적도의 한계 때문에, 매우 큰 테라바이트 수준의 대용량 연산이나, 매우 복잡한 알고리즘을 수행하기에 적합하지 않다.

이에 대한 하나의 대안으로 양자 컴퓨팅이라는 개념이 20세기에 제시되었다. 양자컴퓨팅은 양자 컴퓨터를 이용한 정보 처리를 일컫는 말이다. 양자 컴퓨터는, 기존의 상용화된 컴퓨터와 비슷한 구조를 활용하여 정보를 처리한다. 비트 대신 큐비트를 이용하여 정보를 표현하고, 정보를 큐비트에 암호화하여 처리한다. 하지만 기존의 컴퓨터와는 달리 양자 컴퓨터에는 중첩과 얽힘이라는 특수한 성질이 존재한다. 큐비트는 0 또는 1의 확정된 상태로만 존재하는 것이 아니라, 각각의 상태를 모두 확률적으로 가진다. 가령 큐비트의 0 상태를 $|0\rangle$ 1상태를 $|1\rangle$이라고 하면, 큐비트는 $0.6|0\rangle+0.8|1\rangle$과 같은 상태를 가질 수 있으며, 이렇게 두 상태가 확률적으로 공존하는 상태를 **중첩**이라고 한다.또한 연산 시에는 **얽힘**을 이용하여 한번에 2개 이상의 큐빗에 담긴 정보를 처리할 수 있다. 이러한 점에서 더 효율적으로 정보를 표현 및 처리가 가능하다는 점에서, 양자 컴퓨터의 우위를 확인할 수 있다.

양자 컴퓨터는 이론적으로 정보 표현 및 처리의 관점에서 고전 컴퓨팅보다 효율적이지만, 큐빗의 물리적 특성 때문에 연산의 과정에서 결과를 믿을 수 있는 정도인 충실도가 불안정하고, 논리적 회로의 복잡도가 증가하면, 회로의 실행시간의 증가가 기존의 컴퓨터보다 심하다. 이러한 단점을 보완하기 위하여, **양자 오류 정정**과 **매핑**이라는 과정을 수행한다.

매핑은 논리적인 큐비트와 실제 하드웨어의 큐비트를 서로 대응시키는 과정을 일컫는다. 고전 컴퓨터와 달리 양자 컴퓨터의 HW에서는 이웃한 큐빗들 사이의 연산만 가능하다. 따라서 멀리 떨어진 두 큐빗 사이의 연산을 수행하기 위해서는 두 큐빗 사이의 정보를 논리적으로 교환하는 작업이 필수적이다. 논리적 교환을 수행하게 되면, 충실도가 저하되고, 지연되는 시간이 증가한다. 이러한 이유 때문에 매핑에서 논리적 교환은 가급적 지양하여야 한다. 하드웨어에 따라 직접 물리적인 큐비트를 옮기는 방식을 사용하기도 한다. 직접 옮기는 방식은 충실도의 손해가 없다는 장점이 있다, 하지만 양자역학적 성질 때문에, 중첩 상태를 유지할 수 있는 시간이 제한적이어서 옮기는 시간이 제한 시간 보다 짧을 때만, 논리적 교환에 비해 더 나은 성능을 보인다. 현재의 기술로는 이동시간이 길기 때문에, 제한시간의 제약을 만족시키는 것이 힘들다.

매핑은 논리적 교환이나, 물리적 큐비트의 이동을 이용하여, 보다 신속하고 정확한 양자적 정보 처리를 가능하게 한다.하지만 매핑 만으로는, 정확도가 더 중요한 정보를 처리하는 데 충분하지 않다. 공학자들은 이를 해결하기 위해 양자 오류 정정을 수행한다 고전 컴퓨터에서는 비트 문자열 정보에서 1의 개수를 활용하여 홀수 및 짝수 패리티

검사로 오류를 찾아내고, 해밍 코드를 이용하여 오류를 정정한다. 양자 컴퓨터 또한 이와 비슷한 방식으로 코드를 활용한 오류 정정을 수행한다. 대표적인 방식으로는 IBM의 바이시클 코드나, 서페이스 코드가 있다. 양자 오류 정정 또한 양자 회로로 수행된다. 최신의 연구에서는 양자 오류 정정도 더 효율적으로 수행하기 위해 양자 오류 정정을 위한 매핑 연구 또한 부상하고 있다.