

[양자 컴퓨팅의 다양한 물리적 플랫폼] 측정 기반 양자컴퓨팅과 광집적회로 플랫폼

2025년 3월 21일

손영익



측정 기반 양자컴퓨팅과 광집적회로 플랫폼

2025년 현재, 구글이 기념비적 양자 우월성 실험 성과를 발표한 지 5년 이상의 시간이 흘렀다. 구글의 발표가 있었던 2019년을 기준으로, 해당 분야 전문가들 사이에서 수십 년간 진행되던 양자정보 연구가 빠르게 대중의 관심을 받게 되었다. 최근에는 양자컴퓨터의 중첩을 개념을 소개하기 위해, 예능 프로그램에서 중첩을 비유하여 '살았는지 죽었는지 모르는 산송장 컴퓨터'라는 별명을 붙여줄 정도로 대중적인 주제가 되었다.

특히, 과학기술 분야에 대한 많은 관심을 가진 분들은 비전문가라 할지라도 이미 양자컴퓨터의 기본적인 개념을 이해하고 있을 가능성이 높다. 그럼에도 불구하고, '측정 기반 양자컴퓨팅'이라는 개념은 많은 이들에게 생소할 것으로 보인다. 사실 측정 기반 양자 컴퓨팅은 양자 기술 전문가 그룹 내에서도 제대로 이해하는 사람을 찾아보기 드물 정도로 다소 복잡하고 어려운 개념이다. 본 연재에서는 이러한 '측정 기반 양자컴퓨팅' 개념을 명확하고 이해하기 쉽게 소개할 것이다. 이와 더불어, 측정 기반 양자컴퓨팅을 논할 때 빠질 수 없는 광자 기반 하드웨어 플랫폼도 함께 다루어 통합적으로 설명할 것이다.

3개의 양자컴퓨팅 아키텍처

범용 양자컴퓨팅, 즉 임의의 양자 정보를 다룰 수 있는 양자컴퓨터를 만드는 방법은 상당히 다양하다. 일반적으로 이러한 '다양성'을 논할 때 해당 분야 과학자들은 하드웨어의 종류를 가장 먼저 떠올린다. 구글 및 IBM의 초전도 큐비트는 전자 회로를 활용하고, IONQ의 이온 트랩은 원자를 쓴다는 점에서 큐비트 구성 물질과 그들을 구동하는 방식이 매우 다르다. 방금 예로 든 이러한 방식이 일반적으로 널리 알려진 하드웨어 종류에 따른 분류이다. 하지만 양자컴퓨터는 하드웨어 관점이 아닌 양자 정보를 처리하는 방식에 따라 체계를 분류하기도 한다. 이를 흔히 '아키텍처', '패러다임' 등의 용어로 부르며, 좀 더 풀어 쓴다면 '계산 방법론' 정도가 될 것이다. 본 연재에서는 아키텍처라는 용어를 주로 사용할 것이다.

양자정보 이론에 따르면, 범용 양자컴퓨팅은 세 가지 독립적인 아키텍처로 구현될 수 있다.: 회로 기반 양자 컴퓨팅 circuit-based quantum computing, CBQC, 단열 양자 컴퓨팅 adiabatic quantum computing, AQC, 그리고 본 글의 주제인 측정 기반 양자 컴퓨팅 measurement-based quantum computing, MBQC이 그것이다. 이 세 방식은 서로 동등한 양자 연산을 할 수 있다는 사실이 알려져 있다. 여기서 '동등하다'는 것은 어디까지나 학술의 영역에서 계산 복잡도를 따져봤을 때 그렇다는 것으로, 실제로 하드웨어 제작이나 구동 등 난이도가 같다는 뜻은 전혀 아니다. 그리고 이 세 방식은 어디까지나 정보를 처리하는 논리적인 구조에 관한 내용, 즉 '아키텍처'에 해당하며 특정 하드웨어와 직접적인 관련은 없다.

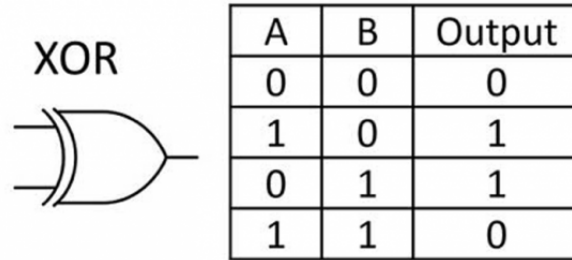
우리에게 보다 친숙한 반도체의 예를 들어 보자. 똑같은 기본 단위 반도체라도 서로 다른 아키텍처가 적용될 수 있다. 예를 들어, 현대 반도체 기술은 모두 실리콘 웨이퍼 프로세싱을 통해서 기본 단위인 트랜지스터를 제작하고 이를 복잡하게 연결하여 사용한다. 하지만 이들을 어떤 방식으로 구성하고 연결하는가에 따라서 서로 완전히 다른 아키텍처를 갖는 CPU central processing unit, GPU graphic processing unit, AP application processor 등으로 크게 나뉘볼 수 있다. 이러한 반도체 칩들은 모두 범용 정보 처리에 쓰일 수 있지만, GPU는 인공지능 모델 훈련에 특화되어 있고 AP는 모바일 프로세서 용도로 쓰이는 것처럼 각각의 성격 및 이에 따른 응용 분야는 매우 다르다. 즉, 반도체라는 하나의 하드웨어 기술임에도 불구하고 구성 및 조작 방식에 따라 여러 가지 아키텍처가 적용되고 서로 다른 특징을 갖게 된다.

이와 유사한 사례로 이번에는 양자컴퓨팅의 일종인 중성 원자 양자컴퓨터를 살펴보자. 중성원자 양자컴퓨터는 보다 널리 알려진 초전도 양자컴퓨터와 마찬가지로 회로기반 방식의 양자 컴퓨팅 연산을 수행할 수 있다. 하지만 이와 동시에 중성원자 플랫폼은 별개의 아키텍처인 단열 양자컴퓨팅 방식의 연산을 수행하기 위해 활용되기도 한다. 즉, 하드웨어 종류와 아키텍처 사이에는 1대1 짝짓기가 성립하는 것이 아니다.

반대로 하나의 아키텍처가 여러 가지 하드웨어 기술에 적용될 수도 있다. 세 아키텍처 중 가장 널리 쓰여지는 것은 회로 기반 방식으로, 구글, IBM, 아마존 등에서 개발 중인 초전도체 큐비트, 유명 스타트업 IONQ의 이온 트랩, 인텔의 반도체 양자점에 모두 쓰이고 있다. 회로 기반 아키텍처는 이렇듯 가장 널리 쓰이고 있을 뿐 아니라 세 가지 아키텍처 중 상대적으로 단순하여 이해하기 용이하다. 따라서 본 연재의 주제인 측정 기반 방식을 깊게 이해하기 위한 사전 작업으로써 초심자가 접근하기에 용이한 회로기반 방식을 먼저 다루어 보자. 회로 기반 아키텍처를 개략적으로 이해한 후 이를 바탕으로 측정 기반 방식을 좀 더 자세히 기술할 것이다.

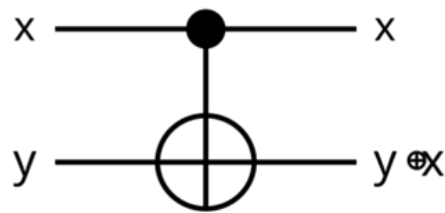
회로기반 고전 및 양자컴퓨팅

양자컴퓨팅의 동작 원리는 복잡하지만, 회로 기반 방식은 상대적으로 이해하기 쉽다. 비전문가일지라도 보다 알기 쉬운 고전 컴퓨터, 즉 반도체 기반 고전 디지털 컴퓨터 개념을 먼저 배우고 거기에 양자역학의 핵심 개념인 중첩과 얽힘을 추가하면 되기 때문이다. 아래의 고전 컴퓨터 예를 살펴보자. 고전 디지털 컴퓨터가 연산을 수행하는 방식은 각종 이진수 정보 처리 게이트들의 집합으로 이루어진다. 고전 컴퓨터의 게이트의 종류에는 AND, OR, NAND, NOR, XOR 등 다양한 요소들이 존재한다. 이 중 XOR 게이트는 아래와 같은 진리표를 가진다.



진리표는 왼쪽 두 열이 입력 정보, 출력이 마지막 열이며 그림으로 표현되어 있다. XOR 게이트는 서로 같은 숫자에 대해서는 0, 다른 숫자에 대해서는 1을 출력한다. 일반 디지털 컴퓨터는 이러한 게이트들을 매우 복잡한 방식으로 연결하여 정보를 처리한다고 이해할 수 있다.

회로기반 양자컴퓨터의 경우도 위의 예와 매우 비슷하데, 다만 고전 컴퓨터의 게이트들이 '양자' 게이트로 바뀌게 된다. 양자컴퓨터도 고전 컴퓨터처럼 다양한 게이트를 갖추고 있으며, 양자역학적 특성을 제외하면 고전 게이트와 유사하다. 예를 들어, Controlled-NOT_{CNOT} 게이트의 기호 및 진리표를 살펴보자.



input		output	
x	y	x	y+x
0⟩	0⟩	0⟩	0⟩
0⟩	1⟩	0⟩	1⟩
1⟩	0⟩	1⟩	1⟩
1⟩	1⟩	1⟩	0⟩

그림에 나타난 것처럼 양자 게이트 또한 진리표가 있고 이진수를 다루고 있다. 숫자 좌우에 위치한 수직 막대와 오른쪽 꺾쇠 기호는 양자 정보를 표기하기 위한 것이며, 현 시점에서 그리 중요한 것은 아니니 숫자 자체에 집중하자. 이 진리표는 출력이 1개가 아니라 2개라는 점에서 위 XOR 게이트와는 다르지만, 같은 점이 있다. CNOT 게이트의 두 출력 중 아래쪽 큐비트인 $y \oplus x$ 의 출력값은 디지털 컴퓨터의 XOR 게이트와 똑같다. XOR 게이트는 출력이 하나 뿐이지만, 만약 위쪽 입력값에 직접 연결되어 있는 출력부를 추가한다면 외견상 CNOT 게이트와 같은 결과를 갖게 되는 것으로 보인다. 하지만 이 두 경우는 고전 게이트와 양자 게이트라는 점에서 근본적으로 다르다.

그렇다면 고전 XOR 게이트와 양자 CNOT의 구체적인 차이점은 무엇일까? CNOT 게이트는 양자역학의 핵심 개념인 중첩과 얽힘이 구현될 수 있다는 점에서 고전 게이트와 구분된다. XOR 게이트의 경우는 진리표의 4줄 중 단 하나만이 입력되고 그에 해당하는 출력값이 얻어진다. CNOT 게이트는 4줄의 입력 경우의 수가 자유롭게 '중첩'되어 입력될 수 있다. 예를 들어, xy 의 입력값이 $|0\rangle|0\rangle + |1\rangle|0\rangle$ 라면 출력값도 이에 대응하는 값들의 중첩인 $|0\rangle|0\rangle + |1\rangle|1\rangle$ 로 나타나게 된다. 두 개 이상의 입력값이 동시에 들어올 수 있다는 개념을 '중첩'이라고 하며, $|0\rangle|0\rangle + |1\rangle|1\rangle$ 와 같이 표현되는 상태를 '얽힘'이라고 한다. 위 내용을 통해 중첩과 얽힘에 대한 기초 개념을 파악하였다면, 이를 바탕으로 양자컴퓨터의 범용성에 대하여 논의해 보자.

광기반 범용 universal 양자컴퓨터

범용 컴퓨터라는 개념은 디지털 컴퓨터에 먼저 존재했던 개념이다. 현대인이 가진 컴퓨터는 대부분 범용 컴퓨터라고 할 수 있다. '범용'이라는 용어는 쉽게 말해 그 어떤 종류의 디지털 정보처리도 (시간이 오래 걸릴 수는 있으나) 궁극적으로는 수행할 수 있다는 뜻이다. 우리 손안의 컴퓨터인 스마트폰으로 통화도 할 수 있지만, 이메일도 쓰고 영화도 볼 수 있는 등 온갖 용도로 쓸 수 있는 이유는 반도체 칩이 범용 연산 기기이기 때문이다. 반대로 공학용 계산기나 과거 유행했던 한영 전자사전 같은 것도 일종의 컴퓨터라 할 수 있지만, 특정 기능만을 수행한다는 점에서 범용이 아니다.

범용성을 제대로 이해하기 위하여 디지털 컴퓨터의 구체적인 동작 예를 살펴보자. 스트리밍 서비스로 영화를 시청한다고 할 때, 컴퓨터 내부에서는 어떤 일이 일어날까? 영화의 정보는 빛의 형태와 소리로 이루어지는데, 이 정보의 근원은 영화 촬영장에서 보고 들을 수 있는 아날로그 데이터이다. 영화 제작 시 이러한 아날로그 정보는 녹화 장치를 통해 전자 기기의 디지털 정보, 즉 0과1의 수많은 나열로 변환되어 기록된다. 이 때 아날로그 정보를 디지털로 바꾸는 과정에서 약간의 정보 손실이 생기게 될 수 밖에 없고, 따라서 이러한 일련의 과정은 일종의 수학적 근사라고 생각할 수 있다. 하지만 **충분한 비트 개수**를 사용하게 되면 **원정보에 한없이 가까운 디지털 정보**를 저장할 수 있다. 이어서 해당 정보는 인터넷 회선을 통해 빛의 형태로 우리의 범용 디지털 컴퓨터로 전달된 후 다시 전자 형태의 정보로 바뀌어 저장된다. 이어서 디지털 컴퓨터는 전송 받은 수많은 0과 1의 배열을 매우 복잡한 방식으로 처리하여 화면으로 출력하고 소리를 재생한다. 예를 들어, 화면에 빛을 출력하기

위하여 수많은 모니터 픽셀들의 광원들을 끄거나 (0) 혹은 켜는 (1) 명령을 컴퓨터가 수행하는데, 스트리밍을 통해 전송된 영화 파일의 디지털 정보를 모니터 픽셀의 구동 명령으로 바꾸는 과정이 바로 **컴퓨팅**, 즉 계산 과정에 해당된다.

이러한 컴퓨팅 과정은 일견 매우 복잡하게 들리지만, 오랜 시간의 연구개발을 통해 근본 원리가 잘 정립되어 있다. 0과1을 처리하는 과정은 불 함수Boolean function라는 개념으로 기술된다. 간단히 말해, 불 함수는 이진수 입력이 처리되어 이진수 출력이 나오는 과정을 의미한다. 이 때, 불 함수를 깊이 탐구해 보면 재미있는 결론에 도달하게 된다. 그 어떠한 불 함수라 할지라도 이를 **충분히 많은 수의 NAND 게이트만을 연결**하여 구성할 수 있다는 점이다. 디지털 정보 처리에 사용되는 NAND 게이트의 진리표는 아래와 같다.

$$X = \overline{A \cdot B}$$



A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

NAND 게이트라는 단 하나의 디지털 게이트를 충분히 많이 연결하여 임의의 디지털 정보 처리를 구현할 수 있다는 사실을 **범용성universality**이라고 표현한다. 흥미롭게도 기 언급되었던 XOR 과 같은 디지털 게이트는 그 자체로는 범용 게이트에 해당하지 않는다. 하지만 XOR 게이트와 AND 게이트 두 종류 게이트를 자유롭게 연결하여 사용하는 것이 허용된다면 다시 임의의 불 함수를 구현할 수 있게 된다. 따라서 게이트 집합 {XOR 게이트}는 범용성이 없지만, 게이트 집합 {XOR 게이트, AND 게이트}는 범용성을 갖게 된다. 이렇듯 범용 정보 처리를 구현할 수 있는 최소한의 게이트 집합을 찾는 작업은 매우 중요하다. 우리가 이해하고 다루기 쉬운 작은 단위의 객체를 충실히 구현한 후, 이들을 수없이 반복 연결하여 모든 종류의 정보 처리에 활용할 수 있기 때문이다.

양자컴퓨터에도 이와 동일한 개념이 있어, 일반 디지털 컴퓨터와 유사한 범용 양자 컴퓨터와 공학용 계산기 같은 특수 목적 양자 컴퓨터가 모두 존재한다. 본 연재의 주인공은 범용 양자 컴퓨터이므로 전자 위주로 논의를 전개해 보자. 범용이란 용어는 디지털 컴퓨터에서와 마찬가지로, 그 어떤 종류의 양자 정보 처리도 수행할 수 있다는 뜻이다. 쉽게 말해 만능 양자 정보 처리 기기라는 뜻이다.

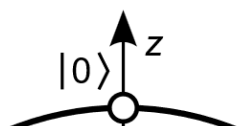
보다 구체적으로 만능 양자 정보 처리 기능은 어떤 뜻일까? 다소 난해할 수 있으나 정확성은 높은 학술적인 표현을 사용하면, '임의의 양자 시스템에 대하여 충분한 숫자의 큐비트와 범용 게이트들을 연결할 경우 **한없이 정확하게 그 동작을 근사**할 수 있다'라고 말할 수 있다. 보다 더 학술적인 양자 역학의 용어를 사용한다면, '양자 시스템 동작'을 '유니터리 변환unitary transformation' 이라는 용어로 대체할 수 있다. 유니터리 변환은 양자계가 시간을 지나며 겪는 변화를 나타내는 함수이다. 범용 양자 컴퓨터는 그 어떤 유니터리 변환이 주어지더라도 많은 숫자의 큐비트와 범용 게이트를 연결할 수만 있다면 충분한 정도의 정확도로 이를 근사할 수 있는 기계 장치라 할 수 있다.

위에서 디지털 컴퓨터가 범용성을 가지므로 임의의 정보 처리를 할 수 있다고 표현했으나, 이는 양자계에는 적용될 수 없다. 양자 시스템의 특성 상 디지털 컴퓨터로 모사하는 계산을 시도할 경우, 문제 복잡도가 증가함에 따라 거의 무한대에 가까운 메모리와 시간이 소요되기 때문에 수행이 불가능하다. 메모리, 시간, 하드웨어 개수 등 모든 측면에서 **유한한 자원**으로 문제를 해결할 수 있을 경우, 이를 학술적으로는 '**확장성scalability**'이라고 표현한다. 이는 양자컴퓨터를 실제로 구동한 하드웨어로 만들 수 있냐 없냐라는 공학적 질문과는 다소 다른 층위의 개념으로, 원리적으로 가능한가 따져보는 것이다. 따라서 디지털 컴퓨터로 임의의 양자계를 모사하는 접근법은 확장성이 없지만, 양자컴퓨터로 양자계를 근사하는 일은 유한한 자원을 사용해 해결할 수 있으므로 확장성을 갖게 된다. 실제로 양자컴퓨팅 분야가 시작된 핵심 이유가 바로 과학자들이 위와 같은 결론에 도달했기 때문이다.

이러한 중요성 때문인지, 양자 정보 이론의 기초를 배우다 보면 처음 접하게 되는 개념 중 하나가 바로 이 범용 양자컴퓨터이다. 범용 양자컴퓨터로 동작하기 위해서는 아래의 두 가지 간단한 조건만 만족하면 되는 것이 널리 알려져 있다.

1. 임의의 단일 큐비트 게이트를 수행할 수 있다.
2. CNOT 게이트 등 얽힘 2-큐비트 게이트를 수행할 수 있다.

앞서 {NOT 게이트, AND 게이트} 집합이 범용 디지털 게이트가 된 것 처럼, {단일 큐비트 게이트, CNOT 게이트} 집합은 범용성을 가진다. 즉, 그 어떤 유니터리 변환에 대해서도 충분한 개수의 단일 큐비트 게이트와 CNOT 게이트를 연결할 수 있다면 매우 정확하게 근사할 수 있게 된다. 여기서 기 언급되었던 중첩 및 얽힘과 같은 게이트 양자성 외에도 단일 큐비트 게이트는 디지털 게이트에 대응 개념이 없는 고유 특성을 보유하고 있다. 디지털 정보인 비트의 경우 0,1 의 두 가지 가능성만을 갖기 때문에 싱글 비트 게이트는 NOT 게이트만이 가능하다. 하지만 큐비트의 경우 블로흐 구Bloch sphere 라 불리는 단위 반지름 구 표면의 무한히 많은 점들이 모두 큐비트 정보의 후보가 되며, 따라서 보다 다양한 단일 큐비트 게이트가 존재할 수 있다. 따라서, 앞서 언급한 임의의 단일 큐비트 게이트는 블로흐 구상에서 일어나는 그 어떤 종류의 유니터리 변환이라도 수행할 수 있는 능력이 필요하다.



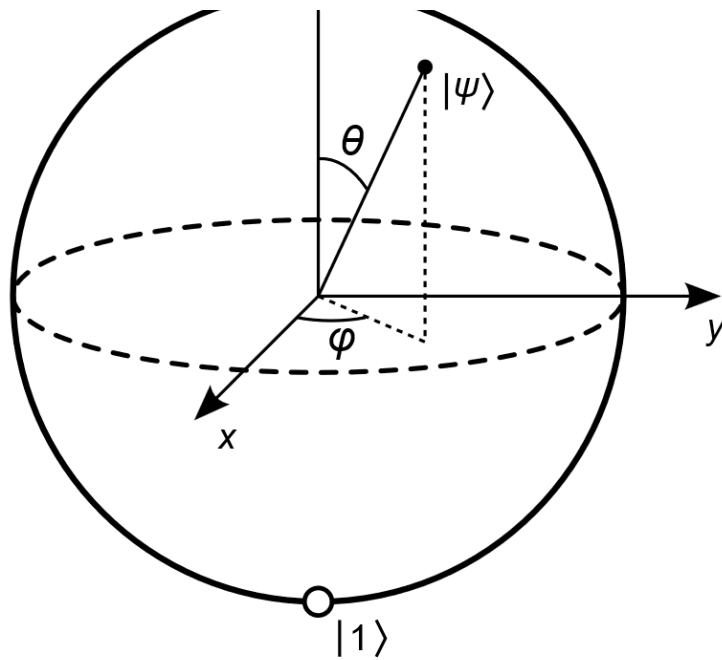


그림 1. 블로흐 구. 구 표면상의 무수히 많은 점들은 구현 가능한 큐비트 상태에 대응될 수 있다. 임의의 큐비트는 $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ 로 표현될 수 있으며, 이 때 $\alpha = \cos(\frac{\theta}{2}), \beta = e^{i\phi} \sin(\frac{\theta}{2})$ 의 대응 관계가 성립한다.

단일 큐비트의 경우, **임의의** 유니터리 변환이 요구된다는 점이 다소 부담스럽긴 하다. 하지만 임의의 유니터리 변환은 파울리 X, Z 게이트 등 다루기 쉬운 기본 단위 게이트로 분해될 수 있기 때문에 상대적으로 쉽게 구현할 수 있다. 실제로 실험에 사용되는 대부분의 물리계가 1번의 조건을 충족한다.

문제는 2번이다. 1번을 만족시키는 양자계는 매우 많음에도 불구하고 이들 모두가 양자 컴퓨팅 하드웨어 후보로 거론되지 않는 이유는 2번 조건을 제대로 만족시키지 못해서이다. 현재 기준으로는 광자, 초전도 큐비트, 반도체 양자점, 중성 원자, 이온 트랩 등 5 종이 범용 양자컴퓨터의 주요 후보 기술로 인정받아 경쟁하고 있다. (기타 물리계의 연구개발이 이루어지고 있긴 하나 산업계의 많은 관심을 끌 만큼의 기술적 진보가 이루어졌다고 보기는 어렵다.) 이온 트랩, 중성 원자, 반도체 양자점, 초전도 큐비트 하드웨어 모두 2번 조건인 2-큐비트 게이트 수행이 가능할 뿐 아니라 그 규모를 확장하기 위한 비교적 뚜렷한 청사진과 현실적 가능성이 존재한다. 이 때문에 세계 각지에서 이 플랫폼들에 대한 상당한 양의 투자가 이루어지고 있다. 그렇다면 본 연재의 주제인 광자 플랫폼은 어떨까?

현대의 과학 기술로 광자만을 활용해서는 2-큐비트 게이트를 만드는 것은 불가능에 가깝다. 그럼에도 불구하고 광자 기반 양자컴퓨팅 스타트업 중에는 유니콘 기업이 2개나 존재하며, 이온 트랩 플랫폼과 유사한 규모의 대규모 투자가 지속되고 있다. 2번의 조건을 만족하지 못하므로 범용 양자 연산이 불가능하고, 따라서 후보로 거론될 수 없을 것 같은데 왜 이런 현상이 벌어질까? 그 비밀이 MBQC, 즉 측정 기반 양자 컴퓨팅 아키텍처에 있다.

광자 기반 플랫폼과 측정 기반 아키텍처

서두에 언급한 것처럼 측정 기반 아키텍처와 광자 기반 하드웨어 플랫폼은 서로의 단짝이며, 이 사실 덕분에 광자 기반 양자컴퓨팅 기술이 각광받고 있다. 이러한 결론에 도달하려면 광자 플랫폼에서 사용되는 과학과 기술 전반에 대한 입체적인 이해가 필요하므로 이 내용을 함께 알아보려 한다.

우선, 광자 기반 기술이 왜 2-큐비트 게이트에 불리한 지 설명해 보자. CNOT 게이트의 경우, control 큐비트가 0 일 경우에 target 큐비트에는 아무 일도 일어나지 않고, 1일 경우에는 target 큐비트를 뒤집어 준다. 진리표를 통해 이러한 설명이 참임을 확인할 수 있다. 이제 이 기술을 하드웨어 관점에서 다시 설명해 보자. 여기에 활용되는 두 큐비트는 우주에 존재하는 두 입자라 볼 수 있다. 위 기술에 의하면, control 입자의 상태에 따라 target 입자의 상태가 영향을 받아야 한다. 이를 좀 더 단순하고 추상적으로 표현해 본다면, 적어도 두 입자는 '상호 작용'을 해야 한다. 좀 더 쉽게 일상의 용어로 이야기 하자면, 적어도 두 입자는 서로가 서로의 존재를 '느낄' 수 있어야 한다.

그렇지만 광자들은 물리학에서 '보존'이라 불리는, 일반적으로 상호 작용을 하지 않는 입자들이다. 지금 당장 손전등 A,B 2개를 들고 스타워즈의 광선검 전투 장면을 연출하려 한다면 가능하지 않을 것임을 우리는 일상의 경험을 통해 알고 있다. 이는 두 손전등에서 방출되는 광자가 서로 간섭하지 않고 독립적으로 움직이기 때문이다. 손전등 A에서 나오는 광자들은 손전등 B가 켜져 있던 꺼져 있던 아무런 영향을 받지 않는다. 다시 CNOT 게이트 이야기로 돌아와 손전등 B의 광자가 control 큐비트, 손전등 A의 광자가 target 큐비트라고 생각을 해 보면 왜 광자 큐비트들만으로는 CNOT 게이트가 불가능한지 알 수 있다.

따라서 측정 기반 아키텍처는 광자 기반 플랫폼의 한계점을 보완할 수 있는 구원투수와도 같다. 측정 기반 연산의 경우 2-큐비트 게이트를 애초에 사용할 필요가 없기 때문이다! 2-큐비트 게이트 대신 아래에 상술할 다른 방식의 하드웨어 운용을 요구하는데, 이를 현실적으로 구현할 수 있는 광자 신호 처리 기술은 이미 존재하며 기존 광통신 분야 연구개발을 통해 많은 진보가 이루어져 있다.

측정 기반 아키텍처의 발전 역사를 간단히 기술하자면 다음과 같다. 측정 기반 양자 연산의 이론적 기반은 2,000년을 전후하여 완성되었으며, 이 시점에 이미 광자가 범용 양자컴퓨팅 연산에 활용될 수 있음이 널리 알려지게 되었다. 다만 당시 기술로 고성능 범용 양자컴퓨터를 구현하려 한다면, 연재 후반부에 상술할 광집적회로라 불리는 소형화된 칩기술을 활용한다 하더라도 컴퓨터 크기가 거대한 산맥 수준으로 추산된다는 것이 큰 문제였다. 따라서 광자 기반 양자 컴퓨팅분야의 핵심 과제는 말도 안되게 커다란 컴퓨터의 크기를 인류가 건설할 수 있는 수준의 규모로 줄이기 위한 설계도 제작, 그리고 여기에서 요구하는 고성능의 광집적회로 하드웨어 생산 기술 혁신이라는 2가지로 요약할 수 있다.

2,000년 초반부터 지금까지, 약 20여 년간 지속적인 연구 개발의 거쳐 광자 기반 양자컴퓨팅은 2개의 과제를 매우 훌륭하게 수행하였다. 거대한 산맥 크기였던 설계도는 이제 슈퍼컴퓨터 설비 정도의 현실적인 규모로 줄어들었으며, 광집적회로 생산 기술은 발전을 거듭하여 이제는 삼성, 엔비디아, TSMC와 같은 세계 최고 반도체 기업들이 차세대 칩 개발에 활용할 정도로 진보하였다.

하지만 현재 설계도가 완벽한지, 현대 생산 공정 기술이 요구 성능을 모두 충족할 수 있을지 판단하는 것은 전문가에게도 어려운 과제다. 따라서 광기반 범용 양자컴퓨터가 가까운 미래에 실제로 제작될 수 있을지 예측하기는 매우 어렵다. 그렇지만 많은 전문가가 광자 기반 플랫폼을 가장 경쟁력 있는 기술로 평가하며, 이를 기반으로 광큐비트 기술이 범용 양자컴퓨팅의 유력한 후보로 자리 잡았다

측정 기반 아키텍처와 광자 기반 플랫폼

앞서 말했던 것처럼 아키텍처와 하드웨어 플랫폼은 별개의 분류 체계로서, 서로 1대 1대응이 되지 않는다. 하지만 그렇다고 모든 아키텍처가 모든 하드웨어 플랫폼과 쉽게 짝 지워 지는 것은 아니다. 특히, 광자 플랫폼의 경우 회로기반 및 단열연산 방식들을 제대로 구현하는 것이 불가능에 가깝기 때문에 현재로서는 측정 기반인 MBQC 만이 유일하게 시도해볼 만한 후보가 된다.

반대로 측정 기반 아키텍처는 여러 종류의 하드웨어 플랫폼으로 구현이 가능하다. 실제로 이온 트랩, 초전도 큐비트 모두 기초적인 측정 기반 양자 연산을 성능을 보여준 바 있다. 그 뿐 아니라, 보다 덜 알려진 광격자 내의 저온 원자와 같은 물리계도 MBQC 와 호환이 가능하다.

정리하자면, 범용 양자 연산을 위한 광자 플랫폼의 유일한 선택지는 측정 기반 아키텍처인데 반해, 측정 기반 아키텍처는 여러 하드웨어 플랫폼의 선택지가 있다. 따라서 아키텍처 - 하드웨어 짝짓기에서 일종의 불균형한 상황이 초래되었으니, 광자 플랫폼이 일방적으로 측정 기반 아키텍처를 원하는 상황으로 봐야 할까? 이에 대한 답은 '아니다' 이다.

여기에는 두 가지 이유가 있다. 첫째로, 광자를 제외한 나머지 4개의 플랫폼의 경우 회로 기반 아키텍처가 그들에게 훨씬 유리하다. 측정 기반 아키텍처에 대한 기초 연구개발을 할 수야 있겠지만, 이것이 회로 기반 아키텍처를 능가하는 우수성이 있지 않은 한, 양자컴퓨터 하드웨어 개발이 지상 목표인 기업의 입장에서는 별 관심이 가지 않을 것이다. 현재까지 이루어지고 있는 연구 개발 양상을 바탕으로 판단컨데, 측정 기반 아키텍처가 다수 하드웨어 플랫폼들의 최우선 선택지는 아니다.

두번째로, 측정 기반 아키텍처의 입장에서 살펴보자. 개별 하드웨어 플랫폼들의 입장은 일단 잊고, MBQC 알고리즘을 구현하기에 가장 효율적인 하드웨어 플랫폼은 무엇일까? 이를 학술적으로 탐구해 분명한 결론을 내리는 것은 쉽지 않은 일이겠으나, 필자의 경험과 감각을 바탕으로 판단할 때 광자 기반 플랫폼이 가장 우수한 것으로 짐작된다. 다시 말해, 측정 기반 아키텍처 입장에서 여러 하드웨어 선택지가 있긴 하지만, 이 중 광자 기반 기술이 가장 적합한 플랫폼이다.

따라서, 광자 기반 플랫폼과 측정 기반 아키텍처는 서로의 한계를 보완하는 최적의 조합이다. 이러한 주장을 납득하려면 측정 기반 아키텍처의 양자 정보 처리 방식에 대한 과학적 이해, 그리고 광자 기반 플랫폼의 하드웨어 기술에 대한 공학적 이해가 모두 필요하다. 이에 대한 내용을 후속 연재를 통해 다루어 볼 것이다.

