논문 21-46-09-01 The Journal of Korean Institute of Communications and Information Sciences '21-09 Vol.46 No.09
https://doi.org/10.7840/kics.2021.46.9.1373

양자 덧셈기를 활용한 양자 비교기 회로 구현방안

강 유 진◆, 허 준°

Circuit Implementation of Quantum Comparator Using a Quantum Adder

Yu-jin Kang◆, Jun Heo°

요 약

본 논문은 자리 올림 비트(carry bit)를 반환하는 양자 덧셈기를 활용하여 양자 비교기를 고안하고 회로 수준의 구현 방안을 제안하였다. 또한 3-큐비트 공간을 예로 들어 회로를 설계하고 이에 대한 시뮬레이션 결과를 제시하였다.

키워드 : 양자 비교기, 양자 덧셈기, 양자 회로
Key Words : Quantum comparator, Quantum adder, Quantum circuit

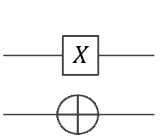
ABSTRACT

This research suggest the quantum comparator using the quantum adder which returns a carry bit and circuit implementation. Furthermore, we design a circuit of 3-qubit data space and present the simulation result.

Ⅰ. 서 론

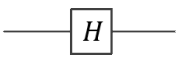
고전 디지털 컴퓨터의 최소 정보 단위는 비트라고 하며 0과 1의 값을 갖는다. 비트 단위의 정보는 NOT, AND, OR 등 다양한 연산 게이트를 통해 처리된다. 이에 상응하는 양자 컴퓨터는 큐비트라는 정보 단위를 가지며, 하나의 큐비트는 $|0\rangle$또는 $|1\rangle$의 상태를 갖거나, $|0\rangle$과 $|1\rangle$이 중첩된 $1/\sqrt{2}(|0\rangle+|1\rangle)$의 상태를 지닐 수도 있다. 양자 컴퓨터 또한 큐비트에 대한 연산 게이트를 정의할 수 있는데 주로 사용되는 게이트는 다음과 같다.

그림 1의 Pauli X 게이트는 주어진 $|0\rangle$상태나 $|1\rangle$상태에 NOT 게이트로 동작한다. 그림 2의 Hadamard 게이트는 $|0\rangle$이나 $|1\rangle$상태를 $|0\rangle$과 $|1\rangle$이 중첩된 상태로 만든다. 반대로 중첩된 상태를 입력할 경우 $|0\rangle$이나 $|1\rangle$상태로 만든다.



Input	X	Output
$\|0\rangle$	$\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}$	$\|1\rangle$
$\|1\rangle$		$\|0\rangle$

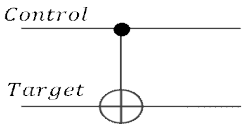
그림. 1. Pauli X 게이트의 회로도 및 진리표
Fig. 1. Circuit diagram and truth table of Pauli X gate



Input	H	Output
$\|0\rangle$	$\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$	$\|+\rangle = \frac{\|0\rangle + \|1\rangle}{\sqrt{2}}$
$\|1\rangle$		$\|-\rangle = \frac{\|0\rangle - \|1\rangle}{\sqrt{2}}$

그림. 2. Hadamard 게이트의 회로도 및 진리표
Fig. 2. Circuit diagram and truth table of Hadamard gate

※ 본 연구는 2021년도 한국과학기술정보연구원(KISTI) 주요사업 과제로 수행한 것입니다
◆ First Author : Korea University Department of Electrical Engineering yujin20@korea.ac.kr, 학생회원
° Corresponding Author : Korea University Department of Electrical Engineering, junheo@korea.ac.kr, 종신회원
논문번호 : 202103-043-A-RE, Received February 2, 2021; Revised May 26, 2021; Accepted June 1, 2021



Control	Target	CNOT	Output	
|0⟩	|0⟩	$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{bmatrix}$	|0⟩	|0⟩
|0⟩	|1⟩		|0⟩	|1⟩
|1⟩	|0⟩		|1⟩	|1⟩
|1⟩	|1⟩		|1⟩	|0⟩

그림. 3. Controlled NOT 게이트의 회로도 및 진리표
Fig. 3. Circuit diagram and truth table of Controlled NOT gate

그림 3의 Controlled NOT 게이트는 Control 큐비트가 $|1\rangle$일 경우 Target 큐비트에 NOT 연산을 취해주고 $|0\rangle$일 경우는 아무런 연산을 취하지 않는다. 이는 Target의 Output에서 볼 때, Control 큐비트와 Target 큐비트의 XOR(배타적 논리합) 연산을 취한 것과 같다.

앞서 살펴본 기본 게이트를 활용하여 더 복잡한 연산을 수행하는 회로를 구성할 수 있다. Thomas의 2017년 논문[1]에 제시된 자리 올림 비트를 반환하는 양자 상수 덧셈기는, n-비트의 상수 a와 변수 b의 덧셈 연산 $r=a+b$를 수행하고 r의 최상위 비트이자 자리 올림 비트만 반환한다. 해당 덧셈기는 a의 각 자리수가 1인 경우에만 CNOT 게이트와 NOT 게이트를 추가하여 회로를 구성하며 $n-1$개의 보조 큐비트 g를 사용한다. 본 논문에서는 1의 보수법을 사용한 뺄셈식이 자리 올림 비트를 사용한다는 것에 착안하여 양자 상수 덧셈기를 비교기로 변환하고, 3-큐비트 데이터 공간에 대한 회로도와 그에 따른 시뮬레이션 결과를 통해 제안한 비교기가 정상적으로 동작하는지 확인하였다.

Ⅱ. 본 론

2.1 1의 보수법을 이용한 양자 비교기 제안

1의 보수법에서 뺄셈 계산 시, 두 수의 뺄셈이 음수인 경우는 자리 올림 비트가 발생하지 않으며 그대로 다시 보수를 취한 후에 마이너스 기호를 붙여 주면 연산 결과를 얻을 수 있다. 반면 두 수의 뺄셈이 양수인 경우는 자리 올림 비트가 발생하므로 자리 올림 된 1을 더해 주어야만 제대로 된 연산 결과를 얻을 수 있다.

그림 4에서 $4-6$을 1의 보수법으로 환산하여 계산할 경우,

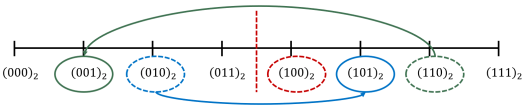


그림. 4. 3-비트 데이터 $(000)_2 \sim (111)_2$. 실선으로 된 초록색 원과 파란색 원은 각각 6과 2의 1의 보수
Fig. 4. 3-bit data $(000)_2 \sim (111)_2$. Green and blue circles drawn with solid lines represent 1's complements of 6 and 2, respectively

$$(100)_2+((110)_2)'=(100)_2+(001)_2=(101)_2 \quad (1)$$

으로 처리되며 자리 올림 비트가 발생하지 않으므로 최종 연산 결과는 다음과 같다.

$$((101)_2)'=-(010)_2=-2 \quad (2)$$

반면 $4-2$를 계산할 경우, 자리 올림 비트가 발생하며 다음 식과 같다.

$$(100)_2+((010)_2)'=(100)_2+(101)_2=(1001)_2 \quad (3)$$

따라서 자리 올림 비트를 최하위 비트에 더한 아래의 식이 최종 결과가 된다.

$$(001)_2+(001)_2=(010)_2=2 \quad (4)$$

n-비트 변수 a와 변수 b를 비교하는 데에 위와 같은 1의 보수 뺄셈식을 사용할 수 있다. 우선, $a-b$를 $a+b'$로 계산하고 자리 올림 비트를 y, 보조 큐비트를 g라고 하면 a,b,y,g는 다음과 같이 나타낼 수 있다.

$$y=\begin{cases}1, \text{if } a>b\\0, otherwise\end{cases} \quad (5)$$

$$a=(a_{n-1}\cdots a_1a_0)_2 \quad (6)$$

$$b=(b_{n-1}\cdots b_1b_0)_2 \quad (7)$$

$$g=(g_{n-1}\cdots g_1g_0)_2 \quad (8)$$

이를 회로로 구현 시, a는 변수이므로 각 자릿수가 1인 경우를 기준으로 설계하되, 제어 신호를 하나씩 추가한다. 또한 a와 b는 양자 상태 $|a\rangle$와 $|b\rangle$로 대응된다.

3-큐비트 공간에 대해, $a+b'$를 회로도를 구성해보면 그림 5로 나타낼 수 있다. 이때 변수 a와 변수 b는

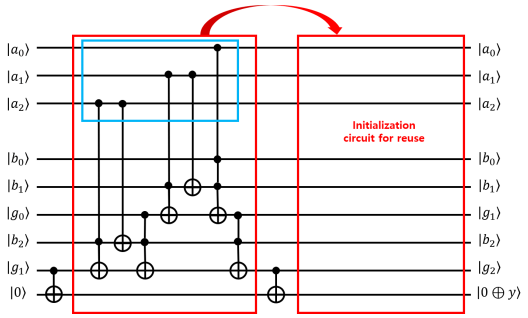


그림 5. 3-큐비트 비교기의 회로도. 변수 a 와 덧셈 시, 파란 박스 안의 제어 신호 추가. 재사용을 위한 초기화 시, 빨간 박스 안의 역연산(사용한 회로의 역순) 수행
Fig. 5. Circuit diagram of 3-qubit comparator. Add control signal when we calculate the sum with the variable a. Inverse operation (reverse order of the circuit used) is performed

$(000)_2$부터 $(111)_2$까지의 값을 가질 수 있다. 우선, $a-b$를 1의 보수법으로 계산하기 위해 b자리에는 b'으로 변환한 값을 대입한다. 따라서 회로도에 제시된 덧셈기는 $a+b'$연산을 수행하게 되며, y는 a의 값이 b보다 큰 경우 1, 작거나 같은 경우 0이 된다. y의 값은 Controlled NOT 게이트를 통해 $|0\rangle$로 초기화된 큐비트에 XOR 되어 해당 큐비트 측정 시, 확인할 수 있다.

2.2 QRAM을 이용한 입력 데이터 구성

사용자가 원하는 데이터는 초기에 고전 디지털 데이터 형태를 지닌다. 그러나 제안하는 양자 비교기의 입력은 비교하고자 하는 모든 입력 데이터가 양자 중첩된 형태이다. 이외에도 양자 연산을 수행하는 다른 연산 회로 또한 양자 중첩 상태의 데이터를 입력으로 갖는다. 따라서 고전 디지털 데이터를 양자 중첩 형태로 변환하는 과정이 필요하다. Giovannetti는 2008년 발표한 논문에서, Quantum Random Access Memory (QRAM) 모델을 제안하면서 해당 모델이 고전 데이터와 양자 데이터의 인터페이스가 될 수 있음을 보였다.[5]

QRAM은 고전 데이터를 불러와, 유니터리 연산을 통해 대용량의 양자 중첩 상태로 만들어준다. 이때 주소가 입력되면 해당 주소의 데이터를 반환하는 기존의 RAM과 비슷한 구조이나, 입력된 주소는 중첩형태여도 가능하다는 점에서 차이가 있다.

임의의 데이터 테이블을 T 라 할 때, 인덱스 i를 통해 대응되는 데이터 $T[i]$를 불러올 수 있다.

$|0\rangle$로 초기화된 상태에서 시작하여, 사용자가 중첩 상태의 주소(인덱스)를 입력하면 해당 주소의 데이터 $T[i]$를 양자 상태로 대응시킨다.

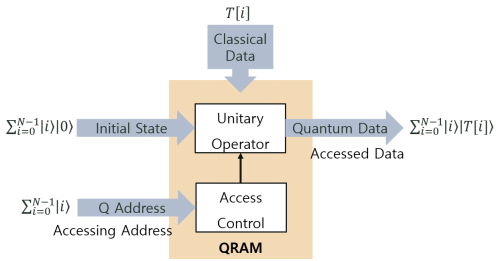


그림 6. QRAM의 구조
Fig. 6. Structure of QRAM

$$\sum_{i=0}^{N-1} |i\rangle \rightarrow \sum_{i=0}^{N-1} |i\rangle |T[i]\rangle \tag{9}$$

사용하는 QRAM 모델에 따라 양자 상태의 데이터 반환에 추가 복잡도가 소요되는데, 최대로 계산해보면 n-큐비트에 대해 $O(2^n)$의 복잡도가 필요하다. 해당 논문에서 Giovannetti는 $O(\log_2(2^n))$ 까지 복잡도 낮춘 "Bucket-brigade QRAM" 모델을 제안하였다. 이와 같이, QRAM의 복잡도는 사용되는 모델에 따라 달라질 수 있으며, 더 낮은 복잡도의 모델을 개발하는 연구도 활발히 진행 중이다.

QRAM을 제안하는 양자 비교기와 결합한다면 다음의 활용 방안을 제시할 수 있다. 우선, 비교를 원하는 모든 a와 b값의 주소를 입력하면 QRAM은 양자 중첩 상태의 a와 b값을 반환한다. 이를 다시 제안하는 양자 비교기의 입력으로 넣어주면, 모든 중첩 상태의 비교 결과를 얻을 수 있으므로 병렬 계산을 통한 양자 연산의 이득을 기대할 수 있다.

2.3 제안 양자 비교기의 응용 방안

최솟값을 찾는 양자 알고리즘인 Dürr-Høyer 알고리즘(DHA)에서, 최솟값을 찾는 과정 중 그림 7와 같은 고전 비교기가 필요하다[7]. 기존의 방식을 사용할 경우, 고전 비교기의 결과가 양자 탐색 알고리즘의 오

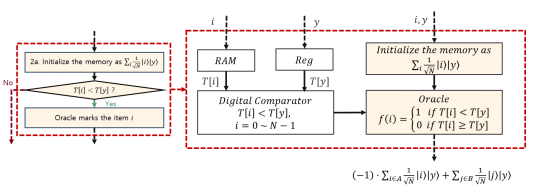


그림 7. 기존 DHA 알고리즘의 비교구문 수행 방식
Fig. 7. Conventional comparator in DHA algorithm

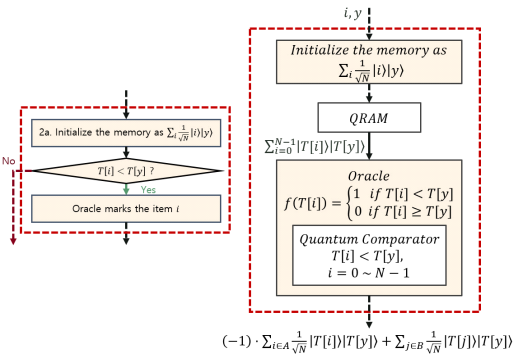


그림 8. 제안하는 방식의 DHA 알고리즘의 비교구문
Fig. 8. Proposed comparator in DHA algorithm

라클에 전달되어야만 원하는 최솟값을 찾을 수 있다. 그러나 그림 8처럼 양자 비교기를 사용할 경우, QRAM을 통해 중첩된 데이터를 입력할 수 있고 결과를 병렬로 얻을 수 있다는 장점을 갖는다.

Ⅲ. 실 험

제안하는 양자 비교기 검증에는, IBM Q에서 제공하는 양자 회로 시뮬레이션 라이브러리인 Qiskit을 사용하여 Python으로 진행하였다. 사용된 회로도는 그림 5에서 제시한 회로도를 토대로 b를 b'으로 변경하는 과정까지 포함하여 그림 9에 제시하였다. 이때, 데이터 및 ancilla 큐비트로 총 9 큐비트를 사용하며 controlled operation을 구현 시 추가의 ancilla 큐비트가 소요된다. 따라서 오류 및 큐비트 수 제한이 있는 실제 양자 컴퓨터 모델 대신, 이상적인 양자 컴퓨팅 시뮬레이션을 제공하는 qasm simulator에서 테스트하였다. 즉, 모든 기본 게이트 연산은 오류 없이 수행된다고 가정하며, 게이트 연산 오류로 발생할 수 있는 상태들의 확률은 0으로 표기된다.

그림 9에서 각 큐비트와 변수는 다음의 식으로 대응된다.

$$\begin{aligned} a &= (q_2 q_1 q_0)_2, \\ b &= (q_6 q_4 q_3)_2, \\ g &= (q_7 q_5)_2, \\ y &= q_8 \end{aligned} \tag{10}$$

해당 회로는 $a=(101)_2=5_{10}$인 경우를 기준으로 하였으며, 중첩된 상태의 모든 b값에 대해 비교하였다. 중첩된 b의 상태 중, $(000)_2$, $(001)_2$, $(010)_2$, $(011)_2$, $(100)_2$인 경우는 a보다 작기 때문에 $y=1$

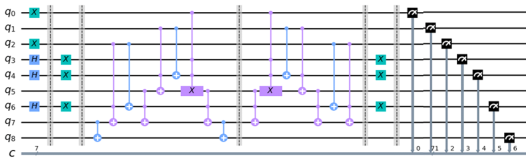


그림 9. $a=(101)_2=5_{10}$ 일 때, 3-큐비트 비교기 회로도.
Fig. 9. Circuit diagram 3-qubti comparator when $a=(101)_2=5_{10}$

의 값을 가질 것으로 예상할 수 있다. 그림 9의 회로로 1024번의 시뮬레이션을 수행한 결과, a와 b, y의 측정값 분포는 그림 10과 같다.

그림 10의 히스토그램에서, 측정 비트는 순서대로

$$(y)(b)(a)=(q_8)_2(q_6 q_4 q_3)_2(q_2 q_1 q_0)_2 \tag{11}$$

를 나타낸다. 제시된 측정값은 양자 비교기의 출력으로 가능한 모든 값을 의미하며, b 가 $a=(101)_2=5_{10}$ 보다 작은 $(000)_2$, $(001)_2$, $(010)_2$, $(011)_2$, $(100)_2$의 경우에만 최상위 비트인 y 가 1이 되는 것을 확인할 수 있었다. 또한 b 가 a보다 크거나 같은 경우 최상위 비트 y가 0인 것도 확인할 수 있으므로, b가 a보다 더 작은지는 y 값이 1인지를 통해 알아낼 수 있었다. 이 외에도 대소관계를 잘못 판단할 시 나타날 수 있는 $(1101101)_2$,$(1110101)_2$,$(0000101)_2$,... 과 같은 경우들은 발생 확률이 0이므로 히스토그램에 나타나지 않았다. 따라서 제안하는 양자 비교기는 양자 상태인 $|a\rangle$와 $|b\rangle$값을 비교하고, 그 결과는 추후 양자 탐색 알고리즘과 같이 데이터를 탐색하는 과정에서 원하는 값에 대한 조건 처리 시, 사용할 수 있다.

또한 제안하는 방식은 2007년 Oliveira에 의해 제시된 양자 비교기보다 1비트 간 비교 값을 저장하는 데에 더 적은 ancilla 큐비트를 사용한다.[8]

즉, 2007년에 제안된 방식은 3비트의 값 두 개의

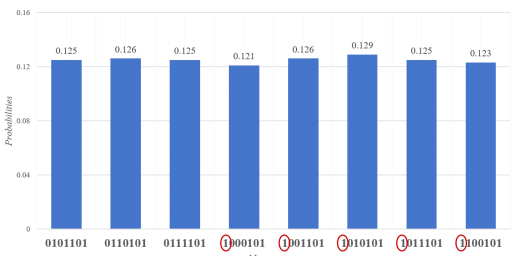


그림 10. 그림 9의 회로에 대한 측정값의 히스토그램
Fig. 10. Histogram of measurements on the circuit in Fig. 9

비교 결과를 저장하는 데에 8개의 ancilla 큐비트를 사용한 반면, 제안하는 방식은 더 적은 3개의 ancilla 큐비트를 사용한다.

Ⅳ. 결 론

본 논문에서는 기존의 양자 상수 덧셈기를 응용한 양자 비교기를 제안하였으며, 이는 1의 보수법에서 자리 올림 비트를 사용하는 것에 착안하였다. 또한 QRAM을 비교기 입력 데이터와의 인터페이스로 사용하여, 양자 알고리즘을 구현하는 활용 방안을 제시하였다. 마지막으로 제안한 비교기를 양자 회로 수준에서 구현한 회로도와 시뮬레이션 결과를 제시함으로써, 제안한 비교기가 예상한 대로 동작하는지 확인할 수 있었다.

References

[1] T. Haner, M. Roetteler, and K. M. Svore, “Factoring using 2n+2 qubits with toffoli based modular multiplication,” *Quantum Info. and Computation*, vol. 17, no. 7 & 8, 2017.

[2] S. A. Cuccaro, T. G. Draper, S. A. Kutin, and D. P. Moulton, “*A new quantum ripple-carry addition circuit*,” Retrieved Jan. 2021, from https://arxiv.org/abs/quant-ph/0410184, 2004.

[3] J. Hayes and I. L. Markov, “*Quantum approaches to logic circuit synthesis and testing*,” Retrieved Jan. 2021, from https://apps.dtic.mil/dtic/tr/fulltext/u2/a454812.pdf.

[4] M. A. Nielsen and L. I. Chuang, “*Quantum Computation and Quantum Information 10th Anniversary Edition*,” Cambridge University Press, 2010.

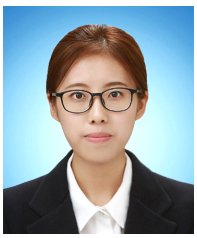
[5] V. Giovannetti, S. Lloyd, and L. Maccone, “Quantum random access memory,” *Phys. Rev. Lett.*, vol. 100, no. 16, Apr. 2008.

[6] C. Dürr and P. Høyer, “*A quantum algorithm for finding the minimum*,” Retrieved Jan. 2021, from https://arxiv.org/abs/quant-ph/9607014.

[7] Y. Kang and J. Heo, “Quantum minimum searching algorithm and circuit implementation,” *2020 ICTC*, pp. 214-219, Jeju Island, Korea, Aug. 2020.

[8] D. S. Oliveira and R. V. Ramos, “Quantum bit string comparator: Circuits and applications,” *Quantum Computers and Computing*, vol. 7, no. 1, pp. 17-26, Jan. 2007.

강 유 진 (Yu-jin Kang)



2018년 : 건국대학교 공학사
2020~현재 : 고려대학교 전기전자공학부 석박통합과정
<관심분야> 양자 정보 이론, 양자 컴퓨팅
[ORCID:0000-0001-5889-8210]

허 준 (Jun Heo)

1989년 : 서울대학교 공학사
1991년 : 서울대학교 공학석사
2002년 : University of Southern California 공학박사
2002~2007년 : 건국대학교 전자공학과 조교수
2007년~현재 : 고려대학교 전기전자공학부 교수
<관심분야> 양자 정보 이론, 양자 컴퓨팅, 통신 시스템