

리터럴 스위치에 의한 다중제어 유니터리 게이트의 새로운 함수 임베딩 방법

박동영*

A New Function Embedding Method for the Multiple-Controlled Unitary Gate based on Literal Switch

Dong-Young Park*

요 약

양자게이트 행렬은 치수가 r , 제어상태벡터 수가 n 및 표적상태벡터 수가 1인 경우에 $r^{n+1} \times r^{n+1}$ 차원 행렬이므로 n 증가에 따른 행렬 크기는 지수 함수적 증가 특성을 갖는다. 만약 제어상태벡터의 경우 수가 2^n 이라면 $2^n - 1$ 경우는 입력이 출력에 보존되는 단위행렬의 항등연산이고, 오직 한 개의 제어상태벡터 연산만이 표적상태벡터에 대한 유니터리 연산이다. 본 논문은 행렬차원 증가에 결정적 기여를 하는 $2^n - 1$ 개의 단위행렬 연산을 한 동작의 산술역승 연산으로 대체할 수 있는 새로운 함수 임베딩 방법을 제안한다. 제안한 함수 임베딩 방법은 다치 임계값을 갖는 2진 리터럴 스위치를 사용하므로 범용 하이브리드 MCU 게이트를 $r \times r$ 유니터리 행렬로 실현할 수 있다.

ABSTRACT

As the quantum gate matrix is a $r^{n+1} \times r^{n+1}$ dimension when the radix is r , the number of control state vectors is n , and the number of target state vectors is one, the matrix dimension with increasing n is exponentially increasing. If the number of control state vectors is 2^n , then the number of $2^n - 1$ unit matrix operations preserves the output from the input, and only one can be performed the unitary operation to the target state vector. Therefore, this paper proposes a new method of function embedding that can replace $2^n - 1$ times of unit matrix operations with deterministic contribution to matrix dimension by arithmetic power switch of the unitary gate. The proposed function embedding method uses a binary literal switch with a multivalued threshold, so that a general purpose hybrid MCU gate can be realized in a $r \times r$ unitary matrix.

키워드

$2^n - 1$ Times Of Unit Matrix Operations, Function Embedding, Arithmetic Power, Literal Switch, Hybrid MCU Gate
 $2^n - 1$ 회 단위 행렬 연산, 함수 임베딩, 산술 역승, 리터럴 스위치, 하이브리드 다중 제어 유니터리 게이트

1. 서 론

양자역학에서 시간 독립적인 정상상태(stationary

state)의 양자들은 에너지를 소모하지 않는다. 이와 같은 정상상태의 양자들을 행렬로 나타내면 유니터리 성질을 만족하는 대각선 행렬이 된다. 상대적으로 대

* 교신저자 : 강릉원주대학교 정보통신공학과
• 접수 일 : 2016. 12. 23
• 수정완료일 : 2017. 02. 13
• 게재확정일 : 2017. 02. 24

• Received : Dec. 23, 2016, Revised : Feb. 13, 2017, Accepted : Feb. 24, 2017
• Corresponding Author : Dong-Young Park
Dept. of Information and Telecommunication Eng., Gangneung-Wonju National University
Email : kouksundo@gwnu.ac.kr

각선을 벗어난 행렬요소들은 전이와 역전이 상태의 시간 의존적인 양자들을 나타낸다.

양자논리함수의 실현 방법은 함수 임베딩 방법 [1-7]과 직접적 사상(mapping)[8-11] 방법으로 나뉜다. 그동안 많은 함수 임베딩 방법이 제안되었으나 그 근간은 행렬 안에 양자게이트에 대한 함수적 기능을 부여하는 행렬 임베딩이다[1-7]. 그것은 양자역학에서 복소수로서의 양자 상태를 나타내는데 가장 적합한 수학적 공간이 힐버트 공간이란 가상의 복소수 공간이며 그것은 곧 행렬의 차원 공간이다. 양자의 스핀 (spin) 성질을 나타내기 위해서는 행렬 외에도 디랙 (Dirac)의 브라켓(braket, $\langle | \rangle$) 기호가 요구되며, 두 개 이상 양자들의 상호 작용을 나타내기 위한 텐서연산이 필요하다. 행렬임베딩 외에 유니터리 게이트를 제어함수의 산술 곱함수로 제어하는 임베딩 방법이 제안되었다[12]. 그러나 이 방법은 게이트 동작 임계값이 binary인 경우에만 적용할 수 있는 방법이다. 그 외에도 4차 $NCV-|v_1\rangle$ 라이브리리[3],[6-7]을 사용해 $2n+1$ 의 저비용 하이브리드(hybrid) 게이트로 OR, NOR, AND, NAND의 기본 논리게이트를 함수 임베딩 한 연구도 발표되었다[6-7]. 본 논문은 기존의 binary 산술곱셈 임베딩 방법[12]를 ternary qutrit와 quaternary qudit의 다치 소자에 적용할 수 있도록 확장한 연구이다. 본 논문은 II장에서 양자컴퓨팅 기초 지식을 논하고, III장에서 본 논문이 제안하는 스위치 함수 임베딩 알고리즘을 제시하였다. IV장에서는 제안한 알고리즘을 현존하는 대표적 양자 게이트 모델들에 적용한 상세설계를 논하고, 기존의 대표적 함수임베딩 방법과 비교하였다. V장은 본 논문의 결론이다.

II. 양자컴퓨팅 기본

2.1 유니터리 변환

양자정보통신에서 양자의 상태벡터는 다치(multiple-valued) 상태에 따라 부르는 명칭이 다르다. $|0\rangle$ 과 $|1\rangle$ 의 두 가지 상태를 갖는 양자는 큐비트(qubit)라 부르며, 세 개의 상태벡터 ($|0\rangle$, $|1\rangle$, $|2\rangle$)를 갖는 양자는 qutrit, 그리고 네 개 이상의 다치 상태를 갖는 양자는 qudit이라 부른다. qubit 양자 에너지가 $\frac{1}{2}h$ 인 양자의 업 스핀(up spin)과 다운(down) 스핀의 상태

벡터는 디랙 기호와 2×1 고유벡터(eigenvector) 행렬에 의해 $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ 과 $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ 로 표기한다. 수학적으로 힐버트(Hilbert) 공간의 큐비트 $|\psi\rangle$ 와 $|\phi\rangle$ 가 각각 $|0\rangle$ 과 $|1\rangle$ 상태에 있을 확률크기가 각각 α 와 β 및 γ 와 δ 는 일 때 식(1)과 식(2)과 같은 상태벡터로 표현할 수 있다. 각 상태의 확률은 확률크기 정규화로 절대제곱이며, 두 상태의 확률 합은 1이다.

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}, |\alpha|^2 + |\beta|^2 = 1 \quad (1)$$

$$|\phi\rangle = \gamma|0\rangle + \delta|1\rangle = \begin{pmatrix} \gamma \\ \delta \end{pmatrix}, |\gamma|^2 + |\delta|^2 = 1 \quad (2)$$

두 개 이상의 양자상태는 식(3)의 텐서 곱(tensor product, \otimes) 연산자를 사용해 나타낸다.

$$|\psi\rangle \otimes |\phi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \otimes \begin{pmatrix} \gamma \\ \delta \end{pmatrix} = \begin{pmatrix} \alpha\gamma \\ \alpha\delta \\ \beta\gamma \\ \beta\delta \end{pmatrix} \quad (3)$$

식(3)의 텐서 연산에서 좌측 양자 $|\psi\rangle$ 는 제어 양자로서 우측의 표적 양자 $|\phi\rangle$ 을 제어한다. 식(4)은 텐서 연산의 다양한 표기법을 나타낸다.

$$|\psi\rangle \otimes |\phi\rangle = |\psi\rangle |\phi\rangle = |\psi\phi\rangle \quad (4)$$

식(4)의 텐서 연산은 4개의 $2^2 \times 1$ 고유벡터 행렬을 생성한다. 일반적으로 n 개 큐비트의 텐서 곱은 2^n 개의 $2^n \times 1$ 고유벡터 행렬을 생성한다. 행렬에서 서로 수반(adjoint) 관계에 있는 두 행렬의 곱이 단위행렬을 이룰 때 이들을 유니터리 행렬이라 부른다.

$$U^\dagger U = U U^\dagger = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = I \quad (5)$$

단일 양자의 상태 변환을 위한 다양한 기본 연산자들이 있는데 그 중에서 식(6)의 NOT(X), 식(7)의 회전(Y) 및 식(8)의 선택적 반전('1') 기능을 갖는 파울리(Pauli) 연산자가 있다.

$$X = \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad (6)$$

$$Y = \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad (7)$$

$$Z = \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad (8)$$

위에서 X 와 Y 는 대각선 행렬이 아니므로 행렬 요소들이 시간 변화한다. 대각선 행렬이 아닌 임의 연산자 A 을 식(5)과 같은 수반관계 행렬 사이에 끼워 넣는 것을 유니터리 변환(unitary transformation)이라 한다. 이 유니터리 변환에서 $AU=UB$ 을 만족하면 대각선의 유니터리 행렬 B 가 구해진다.

$$U^\dagger AU = B, \text{ iff } AU = UB. \quad (9)$$

파울리 X 게이트에 대한 유니터리 변환은 대각선 행렬인 또 다른 파울리 게이트 Z 로 순환되며, 이의 역도 성립한다. 그림 1과 식(10)은 파울리 게이트의 유니터리 변환에 따른 순환 성질을 보여 준다. 식(10)은 그림 1의 행렬 연산이다.

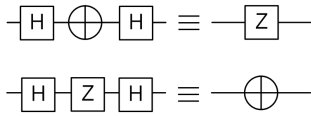


그림 1. 유니터리 변환과 순환 성질

Fig. 1 Unitary transformation and permutation property

$$H^\dagger XH = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = Z \quad (10)$$

, 여기서 $XH=HZ$, H 는 Hadamard 게이트.

텐서 곱 연산에는 최소 두 큐비트가 필요하므로 파울리 X 와 Z 게이트의 텐서 곱 연산을 위해 좌측에 제어큐비트 한 개를 추가한 것을 각각 cX (controlled- X)와 cZ (controlled- Z) 게이트라 부른다. 그림 2에서 cX 게이트는 cZ 게이트의 유니터리 변환을 통해 구해짐을 볼 수 있다. 식(11)은 그림 2의 행렬 연산이며, 이의 역도 성립한다.

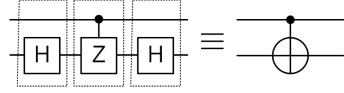


그림 2. 텐서연산을 위한 X 게이트의 유니터리 변환
Fig. 2 Unitary transformation to X gate for tensor product

$$\begin{aligned} (I_1 \otimes H_2) cZ (I_1 \otimes H_2) &= \begin{pmatrix} H & 0 \\ 0 & H \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & Z \end{pmatrix} \begin{pmatrix} H & 0 \\ 0 & H \end{pmatrix} \\ &= \begin{pmatrix} H^2 & 0 \\ 0 & HZH \end{pmatrix} = \begin{pmatrix} I & 0 \\ 0 & X \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = cX \end{aligned} \quad (11)$$

Toffoli 게이트는 cX 게이트에 제어 큐비트 한 개를 추가한 c^2X (double-controlled X) 유니터리 변환으로 식(12)과 같다[1-2].

$$\begin{aligned} (I_1 \otimes I_2 \otimes H_3) ccZ (I_1 \otimes I_2 \otimes H_3) &= \begin{pmatrix} H & 0 & 0 & 0 \\ 0 & H & 0 & 0 \\ 0 & 0 & H & 0 \\ 0 & 0 & 0 & H \end{pmatrix} \begin{pmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & Z \end{pmatrix} \begin{pmatrix} H & 0 & 0 & 0 \\ 0 & H & 0 & 0 \\ 0 & 0 & H & 0 \\ 0 & 0 & 0 & H \end{pmatrix} \\ &= \begin{pmatrix} H^2 & 0 & 0 & 0 \\ 0 & H^2 & 0 & 0 \\ 0 & 0 & H^2 & 0 \\ 0 & 0 & 0 & HZH \end{pmatrix} = \begin{pmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & X \end{pmatrix} = c^2X = U_{11} \end{aligned} \quad (12)$$

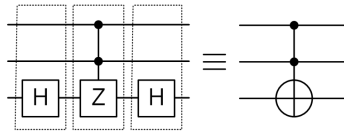


그림 3. Toffoli 게이트

Fig. 3 Toffoli gate

그림 3의 Toffoli 게이트는 식(13)과 같이 $2^3 \times 2^3$ Toffoli 행렬 U_{11} 에 위에서의 텐서 곱과 유니터리 변환에 의해 AND 연산 기능을 임베딩하고 있다. 이때 $|x_1 x_2 \rangle$ 와 $|x_t \rangle$ 는 각각 제어입력 큐비트와 표적입력 큐비트이며, 대응하는 출력 큐비트는 각각 $|y_1 y_2 \rangle$ 와 $|y_t \rangle$ 이다.

$$|y_1 y_2 y_t \rangle = U_{11} |x_1 x_2 x_t \rangle \quad (13)$$

$$, \text{여기서 } U_{11} = \begin{pmatrix} 10000000 \\ 01000000 \\ 00100000 \\ 00010000 \\ 00001000 \\ 00000100 \\ 00000010 \\ 00000010 \end{pmatrix}.$$

식(14)은 식(13)의 Toffoli 행렬 U_{11} 을 모든 상태벡터들로 확장한 범용 Toffoli 게이트이다.

$$|y_1 y_2 y_t \rangle = U_{x_1 x_2} |x_1 x_2 x_t \rangle \quad (14)$$

$$, \forall x_1 x_2 \in \{00, 01, 10, 11\}$$

$$, U_{x_1 x_2} = \begin{pmatrix} X & 0 & 0 & 0 \\ 0 & X & 0 & 0 \\ 0 & 0 & X & 0 \\ 0 & 0 & 0 & X \end{pmatrix}.$$

III. 스위치함수 임베딩 알고리즘

본 논문이 제안하는 게이트 행렬의 차원압축을 위한 스위치함수 임베딩 알고리즘과 함수실현 구조는 그림 4와 같다.

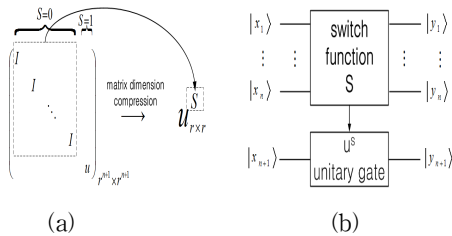


그림 4. 스위치함수에 의한 하이브리드 MCU 게이트 실현 블록도
(a) 알고리즘 (b) 구조

Fig. 4 Block diagram of the hybrid MCU gate realization based on the switch function
(a) algorithm (b) architecture

정의 1. r -치 논리 경우에 제어 qutrit $|x \rangle$ 에 대한 제어함수를 $f(x)$ 라 할 때 $d = r - 1$ 인 임계값을 갖는 2진 리터럴 스위치(literal switch)는 식(15)과 같다.

$$d^{f(x)} = 1 \text{ (on), iff } f(x) = d \quad (15)$$

$$= f(x), \text{ iff } d = 1$$

$$= 0 \text{ (off), otherwise.}$$

정의 2. n qubit 제어입력 상태벡터와 한 개의 qutrit 표적입력 상태벡터를 각각 $|x_1 x_2 \dots x_n \rangle$ 과 $|x_t \rangle$ 라 할 때 $n+1$ 하이브리드 MCT 게이트에서 $|x_1 x_2 \dots x_n \rangle$ 으로 조합 가능한 2^n 개 상태벡터 집합을 A 라고 하면 S_A 는 $a \in A$ 인 제어입력 상태벡터에 대한 n -tuple 리터럴 스위치 s_a 의 2진 AND이다.

$$S_A = \sum_{a=00\dots 0}^{11\dots 1} S_a = \sum_{a=00\dots 0}^{11\dots 1} d^{f(x_1)} d^{f(x_2)} \dots d^{f(x_n)}, \quad (16)$$

, 여기서 $A = \{|00 \dots 0 \rangle, |00 \dots 1 \rangle, \dots, |11 \dots 1 \rangle\}$.

$n+1$ 하이브리드 MCT 게이트는 식(17)과 같이 입력 출력 제어상태벡터는 binary qubit이고 MCT 게이트 내부는 ternary qutrit인 하이브리드 함수 임베딩이 가능하다.

정리. $\exists a \in A$ 일 때 S_A 가 u^{S_A} 과 같이 u 의 산술 곱함수라면 S_a 는 u^{S_a} 의 u 을 on/off 시킬 수 있는 단일 유니터리 스위치함수이며, S_A 는 u^{S_A} 에 대한 범용 유니터리 스위치함수로서 식(17)을 만족한다.

$$|y_1 y_2 \dots y_n y_t \rangle = U_A |x_1 x_2 \dots x_n x_t \rangle \quad (17)$$

$$= U_A |x_1 x_2 \dots x_n \rangle \otimes |x_t \rangle$$

$$= |x_1 x_2 \dots x_n \rangle \otimes u^{S_a \in A} |x_t \rangle$$

(증명)

$$u^{S_A} = u^{\sum_{a=00\dots 0}^{11\dots 1} S_a}$$

$$= u^{S_{0\dots 00}} \dots u^{S_{1\dots 11}}$$

$$= u^{(d^{f(x_1)} \wedge d^{f(x_2)} \wedge \dots \wedge d^{f(x_n)})} \dots u^{(d^{f(x_1)} \wedge d^{f(x_2)} \wedge \dots \wedge d^{f(x_n)})}$$

$$= (u^{d^{f(x_1)}} u^{d^{f(x_2)}} \dots u^{d^{f(x_n)}}) \dots (u^{d^{f(x_1)}} u^{d^{f(x_2)}} \dots u^{d^{f(x_n)}})$$

$$= \begin{pmatrix} u & 0 & \dots & 0 \\ 0 & u & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & u \end{pmatrix} = U_A$$

$$|y_1 y_2 \dots y_n y_t \rangle = U_A |x_1 x_2 \dots x_n x_t \rangle$$

$$= u^{S_A} |x_1 x_2 \dots x_n x_t \rangle$$

$$= u^{S_A} |x_1 x_2 \dots x_n \rangle \otimes |x_t \rangle$$

$$= |x_1 x_2 \dots x_n \rangle \otimes u^{S_A} |x_t \rangle$$

Q.E.D.

위의 정리는 그림 4 알고리즘의 구현이다. 즉 $S_a = 1$ 일 때 유니터리 스위치가 on 되어 표적상태벡터에 유니터리 연산 $u|x_t\rangle$ 을 실행하고, $S_a = 0$ 인 $2^n - 1$ 개 제어입력상태벡터들에 대해 $|y_t\rangle = |x_t\rangle$ 을 만족하는 2×2 단위행렬 연산을 실행함으로써 $2^{n+1} \times 2^{n+1}$ 게이트 연산이 2×2 유니터리 게이트 연산으로 구현될 수 있었다.

IV. 임베딩 알고리즘 적용 및 고찰

본 장에서는 대표적인 세 가지 양자게이트 모델 [2-3],[6-7]에 스위치함수 임베딩 방법을 적용할 경우의 상세설계를 제시한 후 행렬임베딩 방법과 비교하였다.

4.1 병렬 AND곱 MCU 게이트 모델[2]

그림 5와 표 1은 병렬 c^2U (double controlled unitary) 게이트 모델에 대한 스위치함수 임베딩 도식과 상세설계 표이다.

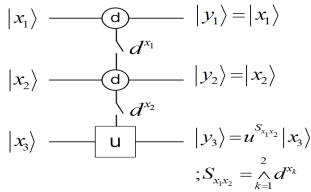


그림 5. 병렬 c^2U 게이트의 리터럴스위치 임베딩
Fig. 5 Literal switch embedding to the parallel c^2U gate

표 1. 그림 5의 임베딩 상세설계
Table 1. Embedding specifications on Fig. 5

Specifications	qubit	qutrit/qudit
radix r	2	3/4
threshold d	1	2/3
control input Ci	x_k	x_k
literal switch L	$d^{x_k} = x_k$	d^{x_k}
unitary switch S	$x_1 x_2$	$d^{x_1} d^{x_2}$
$ y_1 y_2 y_t\rangle = x_1 x_2\rangle \otimes u^{\bigwedge_{k=1}^2 d^{x_k}} x_t\rangle$		

4.2 범용 병렬 AND곱 MCU 게이트 모델[2]

그림 6과 표 2는 그림 5 모델의 제어입력 단에 과울리, M-S 및 $NCV-|v_1\rangle$ 라이브리리 게이트와 같은 $r \times r$ 제어 연산자 a_k 가 추가된 범용 병렬 c^2U 게이트 모델의 스위치함수 임베딩 도식과 상세설계 표이다.

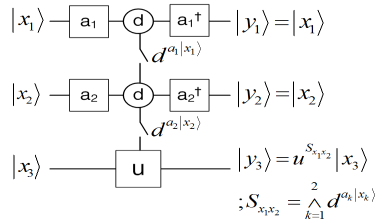


그림 6. 병렬제어연산자를 갖는 범용 c^2U 게이트의 리터럴스위치 임베딩
Fig. 6 Literal switch embedding to the universal c^2U gate with parallel control operators

표 2. 그림 6의 임베딩 상세설계
Table 2. Embedding specifications on Fig. 6

Spec	qubit	qutrit/qudit	hybrid
r	2	3/4	$2(3/4)^*$
d	1	2/3	$(2/3)^*$
Ci	$(x_k + a_k) \bmod r$		
L	$d^{(x_k + a_k) \bmod r}$		
S	$(x_k + a_k) \bmod$	$\bigwedge_{k=1}^2 d^{(x_k + a_k) \bmod r}$	
$ y_1 y_2 y_t\rangle = x_1 x_2\rangle \otimes u^{\bigwedge_{k=1}^2 d^{(x_k + a_k) \bmod r}} x_t\rangle$			

p.s : (internal)*

4.3 종속 AND곱 MCU 게이트 모델[2-3],[6-7]

그림 7은 Toffoli 게이트를 $NCV-|v_1\rangle$ 라이브리리를 사용해 저비용 종속 AND 구조로 실현한 것이다.

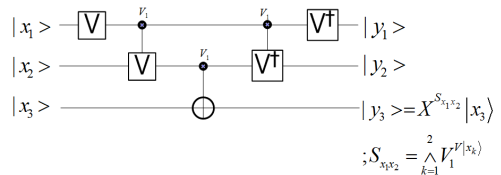


그림 7. Toffoli gate의 하이브리드 실현
Fig. 7 The hybrid realization of Toffoli gate

표 3은 중속 AND급 MCU 게이트 모델의 스위치 함수 임베딩 도식과 상세설계 표이다.

표 3. 그림 7의 임베딩 상세설계
Table 3. Embedding specifications of Fig. 7

Specifications	hybrid Toffoli gate
radix r	$2(4)$
threshold d	$V_1 = \frac{1}{2} \begin{pmatrix} 1-i \\ 1+i \end{pmatrix}$
control input	$V x_k >$
literal switch	$d^{ x_k >$
unitary switch S	$\bigwedge_{k=1}^2 d^{ x_k >$
	$ y_1 y_2 y_t > = x_1 x_2 > \otimes X^{\bigwedge_{k=1}^2 V_1^{ x_k > x_t >$

4.4 비교 및 고찰

표 4는 행렬임베딩 방법과 본 논문의 스위치함수 임베딩 방법을 비교한 표이다. 첫 번째는 Toffoli 게이트에 대한 실현 함수의 비교이다. 두 번째는 함수

표 4. 함수임베딩 비교표
Table 4. Comparison table of function embedding

comparison	Matrix embedding[1-7]	This paper, [12]
Toffoli gate	$U_{11} x_1 x_2 x_t >$	$ x_1 x_2 > \otimes u^S x_t >$
embedding	$2^{n+1} \times 2^{n+1}$ matrix U_A	arithmetic power u^{S_A} , $S_A = \sum_{a=00\dots 0}^{11\dots 1} S_a$ $S_a = \bigwedge_{k=1}^n d^{f(x_k)}$
		literal switch $d^{f(x_k)} = \begin{cases} 1, & \text{iff } f(x_k) = d \\ 0, & \text{otherwise} \end{cases}$
unit matrix operation	$U_f = \begin{pmatrix} I & 0 & \dots & 0 \\ 0 & \ddots & & 0 \\ \vdots & 0 & I & \vdots \\ 0 & 0 & \dots & u \end{pmatrix}$ $2^n - 1$ numbers of I 's operation $u = \begin{pmatrix} u_{00} & u_{01} \\ u_{10} & u_{11} \end{pmatrix}$	unitary switch $u^S = I$, iff $S_a = 0$ to $2^n - 1$ numbers of input state vectors $u^S = u$, iff $S_a = 1$

임베딩 비교로서 행렬임베딩 방법은 $2^{n+1} \times 2^{n+1}$ 행렬로 임베딩을 하였고, 본 논문은 두 개의 스위치함수를 사용하여 2×2 유니터리 게이트의 산술 먹 스위치함수로 임베딩을 하였다. 세 번째로 행렬차원을 증가시키는 원인으로 행렬임베딩 방법은 $2^n - 1$ 개의 $2^n \times 1$ 단위행렬 연산이 주요 원인임을 밝혀졌다. 본 논문은 산술 먹 스위치함수 알고리즘을 도입해 $2^n - 1$ 개 단위행렬 연산을 단일 산술 먹 연산으로 압축할 수 있음을 보였다.

V. 결 론

본 논문에서는 리터럴 스위치에 의한 MCU 게이트의 새로운 함수 임베딩 방법을 제안하고, 알고리즘의 타당성을 검증하였다. 본 논문은 양자 게이트의 행렬 임베딩 시에 행렬차원 증가의 주요 원인이었던 $2^n - 1$ 개의 $2^n \times 1$ 단위행렬 연산을 단일 2×2 유니터리 게이트의 산술 먹 2진 스위치 연산으로 임베딩 할 경우에 행렬차원의 효율적 압축이 가능함을 밝혔다. 본 논문이 제안한 함수 임베딩 방법은 $(n+1) \times (n+1)$ 차원 행렬연산을 2×2 차원의 유니터리 행렬 연산으로 압축하였으므로 큰 행렬을 사용하지 않고 양자논리회로의 함수식을 표현할 수 있다. 특히 본 논문 방법은 행렬 임베딩 방법이 제공하지 않는 게이트 내부의 제어함수 정보를 리터럴 스위치와 유니터리 스위치함수로 제공할 수 있어 직관적 설계와 해석이 요구되는 양자논리회로 설계 시의 함수 표현과 고장 탐지 등에 활용될 수 있을 것으로 사료된다. 하지만 본 논문의 함수 임베딩 방법은 힐버트 공간의 $2^{n+1} \times 2^{n+1}$ 행렬 연산을 2×2 차원 유니터리 행렬 연산으로 압축하였으므로 본래 $2^{n+1} \times 2^{n+1}$ 행렬이 갖고 있던 공간 및 시각적 정보 제공에는 제한적인 단점도 갖고 있음을 밝힌다.

References

[1] M. Bellac, *A Short Introduction to Quantum Information and Quantum Computing*. Cambridge: Cambridge University Press, 2006.
[2] M. Perkowski and D. Shah, *Design of Regular*

- Reversible Quantum Circuits*. Saarbrücken: LAP LAMBERT Academic Publishing, 2011.
- [3] Z. Sasanian, R. Wille, and D. Miller, "Realizing reversible circuits using a new class of quantum gates," In *The 49th Design Automation Conf.*, San Francisco, USA, June 3-7, 2012, pp. 36-41.
- [4] B. Mondal, P. Sarkar, P. Saha, and S. Chakraborty, "Synthesis of Balanced Ternary Reversible Logic Circuit," *IEEE 43rd Int. Symp. on Multiple-Valued Logic*, Toyama, Japan, May 22-24, 2013, pp. 334-339.
- [5] C. Moraga, "On some Basic Aspects of Ternary Reversible and Quantum Computing," *IEEE 44th Int. Symp. on Multiple-Valued Logic*, Bremen, Germany, May 19-21, 2014, pp. 178-183.
- [6] D. Park and Y. Jeong, "Realizing Mixed-Polarity MCT gates using $NCV-|v_1\rangle$ Library," *J. of the Korea Institute of Electronic Communication Science*, vol. 11, no. 1, 2016, pp. 29-35.
- [7] D. Park and Y. Jeong, "For new Duality Structure and its Application in the $NCV-|v_1\rangle$ Library," *J. of the Korea Institute of Electronic Communication Science*, vol. 11, no. 2, 2016, pp. 165-170.
- [8] Z. Zilic, K. Radecka, and A. Khazamiphur, "Reversible circuit technology mapping from non-reversible specifications," *Proc. Design Automation and Test in Europe*, Nice, France, Apr. 16-20, 2007, pp. 558-563.
- [9] D. Miller, R. Wille, and G. Dueck, "Synthesizing Reversible Circuits for Irreversible Functions," *12th Euromicro Conf. on Digital System Design/ Architectures, Methods and Tools*, Patras, Greece, Aug. 27-29, 2009, pp. 749-756.
- [10] S. Sultana and K. Radecka, "Rev-Map: A Direct Gateway from Classical Irreversible Network to Reversible Network," *IEEE 41th Int. Symp. on Multiple-Valued Logic*, Tuusula, Finland, May 23-25, 2011, pp. 147-152.
- [11] D. Park and Y. Jeong, "Gate Cost Reduction Policy for Direct Irreversible-to-Reversible Mapping Method without Reversible Embedding," *J. of the Korea Institute of Electronic Communication Science*, vol. 9, no. 11, 2014, pp. 1233-1240.
- [12] D. Park and Y. Jeong, "A New Functional Synthesis Method for Macro Quantum Circuits Realized in Affine-Controlled NCV-Gates," *J. of the Korea Institute of Electronic Communication Science*, vol. 9, no. 4, 2014, pp. 447-454.

저자 소개



박동영(Dong-Young Park)

1980년 인하대학교 전자공학과 졸업(공학사)

1985년 인하대학교 대학원 전자공학과 졸업(공학석사)

1995년 인하대학교 대학원 전자공학과 졸업(공학박사)

2017년 현재 강릉원주대학교 정보통신공학과 교수
 ※ 관심분야 : 다치논리 회로, 양자컴퓨팅, 양자정보통신

