Controlled formalized operators with multiple control bits

Nikolay Raychev

Abstract—In this report is examined the construction of controlled operators, that include more than one control or target bit. In the research is examined a general class of n qubit operators, that use one or more control bits to conditionally apply an operator to a single target qubit. The conditional nature of these operators is controlled by a boolean function f, that acts on a set of control bits C.

Index Terms— boolen function, circuit, composition, encoding, gate, phase, quantum.

1 INTRODUCTION

In this report is examined the construction of controlled operators, that include more than one control or target bit, based on a formalized qubit operator. This work is part of the developed from the author formalized system for design of algorithmic models for quantum circuits, based on phase encoding, decoding and parameterization of primitive quantum operators. In previous publications of the author [6, 7, 8] were defined several sets of operators on the n qubit, that generalize certain classical characteristics: identity and logical negation. For the construction of the controlled operators, that include more than one control or target bit, may be needed nontrivial combinations of elementary operators. Instead of directly addressing the construction of such operators from elementary components, the focus will be on several methods for operation with formalized operators in a more extensive system of summary controlled operators. For development of the construction of such operators may be examined the work of Barenco. Here will be discussed a general class of n qubit operators, that use one or more control bits to conditionally apply an operator to a single target qubit. The conditional nature of these operators is controlled by a boolean function f, that acts on a set of the control bits C.

2 PARALLEL APPLICATION OF SINGLE QUBIT OPERATORS

When a sequence of single qubit operators do not target the same qubit, then Theorem 3.3.1.8 shows that they swap and may effectively be applied in a different order. Therefore it can be said that they can be applied in parallel and viewed as a single *n* qubit operator. The canonical example of such composition is the application of the Hadamard operator to all or most of the qubits in a register that occurs at the beginning and end of many quantum algorithms. Theorem 1 characterizes parallel arrays of elementary indexed operators and provides formulas for their aggregate encoding and amplitude effects.

Theorem 1 *If* $P = \{(t, \alpha, \gamma\iota\eta)\}$ *is a set of operator parameters such that no pair of the parameter sets in P share a target bit t. Then the operator* $U(P)$ *is the n qubit operator formed by the composition of the operators. If a given input basis* $x \in \mathbb{B}^n$ *and output basis* $y \in \mathbb{B}^n$,

$$\langle y|U(P)|x\rangle = (-1)^{\varepsilon*(P)_{x\oplus y}(x)} a^*(x, y, P) \quad (1)$$

With a function α^*, *defined as*

$$a^*(x, y, P) = \prod_{p\in P} \sqrt{a_p}^{x_{t_p}\oplus y_{t_p}} \sqrt{1 - a_p}^{x_{t_p}\oplus y_{t_p}} \quad (3.4.2)$$

and 2^n-tuple of n bit boolean functions $E^*(P)$ with the *i-th element* $E^*(P)_i$, *defined as*

$$\varepsilon * (P)_i(x) = \bigoplus_{p \in P} \varepsilon'\left((a, \gamma\iota\eta)_p\right)_{i_{t_p}} \left(x_{t_p}\right) \quad (2)$$

Proof. From corollary 3.1.7 it can be seen that for all $p \in P$, where the output data differ from the input at bit t_p, is applied an amplitude divider $\sqrt{1-\alpha_p}$, otherwise is applied $\sqrt{\alpha_p}$. This logic is captured by the formula

$$\sqrt{a_p}^{\overline{x_{t_p}\oplus y_{t_p}}} \sqrt{1 - a_p}^{x_{t_p}\oplus y_{t_p}}$$

Thus the function α^* correctly calculates the magnitude of $\langle y|U(P)|x\rangle$. In order to calculate correctly the phase of $\langle y|U(P)|x\rangle$ first should be noted that, when t_p differs in x and y, then the encoding function $\mathcal{E}((\gamma\iota\eta)_p)_1$ is used, otherwise - $\mathcal{E}((\gamma\iota\eta)_p)_0$ is used. As can be seen, the index of the encoding function is exactly $x_{t_p}\oplus y_{t_p}$. According to corollary 3.1.7, if it is given that the permutation function for one operator does not change the bit on which any other individual encoding function is acting, then each encoding function is applied to the unchanged value of x_{t_p}. Thus, it appears that $E^*(P)$ correctly calculates the phase encoding function of $\langle y|U(P)|x\rangle$. From Theorem 1 any arbitrary parallel array of elementary indexed operators can be characterized by an aggregate encoding and decoding operation. However, the encoding functions of $U(P)$ as calculated by $E^*(P)$ are unsimplified. Given that they are constructed in a conventional way and use elements of $\mathcal{B}^1$ it is possible to be a substantial simplification.

Simplification of Phase Encoding Functions

Since the CONST functions produce results, that are independent of their input data, it is always possible to reduce their presence in an encoding function to a single occurrence of ZERO or ONE.

Theorem 2 *If* $F = \{f_i | f_i \in CONST\}$. *Then x is a binary number in* $\mathbb{B}^n$, *such that*

$$\phi_F(x) = \bigoplus_{i=1}^{n} f_i x_i$$

and

$$\phi_F(x) = \begin{cases} ONE & \text{when } l \text{ is odd} \\ ZERO & \text{otherwise} \end{cases} \quad (3)$$

where $l = |\{i | f_i = ONE\}|$.

Proof. The CONST functions produce results that are independent of their input data. Therefore can be selected arbitrary bit b and replaced each x_i in $\phi_F(x)$ with b, reducing $\phi_F(x)$ from a function in $\mathbb{B}^n \to \mathbb{B}$ to a function in the Abelian group $(\mathcal{B}^1, \oplus)$. After this $\phi_F(x)$ can be rearranged such that all occurrences of *ZERO* to precede those of *ONE,* from where is obtained the formula

$\phi_F(x) = (ZERO^{\oplus k} \oplus ONE^{\oplus l})(b)$

where k, $l \geq 0$ are the number of occurrences of ZERO and ONE respectively and therefore $k + l = n$. From Theorem 2.2.2.3, $ZERO^{\oplus k} = ZERO$, leaving

$\phi_F(x) = (ZERO \oplus ONE^{\oplus l})(b) = ONE^{\oplus l}(b)$

Again, by Theorem 2.2.2.3, this is *ONE,* where l is odd, and *ZERO* – otherwise.

All occurrences of constant functions in an encoding function can be eliminated, so that the encoding function can be rewritten as a boolean expression of $m \leq n$ variables, comprised of *BAL* functions, combined with $\oplus$ or °.

Theorem 3 *If* Φ *is a boolean function over n bits, corresponding to an expression from f the formula*

$$\phi_F(x) = \bigoplus_{i=1}^{n} f_i x_i$$

Then there exists an equivalent formula over $m \leq n$ *bits that also expresses* Φ*, where* m *is the number of* f_i*, which are in BAL such that*

$$\phi_F(x) = \begin{cases} \bigoplus_{f_i \in BAL} f_i(x_i) & \text{if } \bigoplus_{f_j \in CONST} f_j(x_j) = ZERO \\ NOT\left(\bigoplus_{f_i \in BAL} f_i(x_i)\right) & \text{if } \bigoplus_{f_j \in CONST} = ONE \end{cases}$$

Proof. By theorem 2 can be reduced all k occurrences of *CONST* functions to *ZERO* or *ONE,* thus leaving an expression of $n - k$ *BAL* functions and one *CONST* function, combined with the operator $\oplus$ such that,

$\phi_F(x) \in$

$$\left\{ONE \oplus \left(\bigoplus_{f_i \in BAL} f_i(x_i)\right), ZERO \oplus \left(\bigoplus_{f_i \in BAL} f_i(x_i)\right)\right\}$$

Given that

$$ONE \oplus \left(\bigoplus_{f_i \in BAL} f_i(x_i)\right)$$

$$= NOT\left(\bigoplus_{f_i \in BAL} f_i(x_i)\right) \text{ and } ONE \left(\oplus \bigoplus_{f_i \in BAL} f_i(x_i)\right)$$

$$= \bigoplus_{f_i \in BAL} f_i(x_i)$$

then it's true that Φ_F reduces to a boolean function of $n - k \leq n$ variables.

Given that all operators extract their encoding functions $\mathcal{E}$ from $(BAL \times CONST) \cup (CONST \times BAL)$, Theorem 3.3.1.3 leads to a big simplification for the phase encoding functions.

Operators of the form $U^{\otimes n}$

If until now the parallel composition of operators was discussed in general, here will be paid attention to the application of a single operator with noninteger amplitude to n qubits. For such operators, the encoding function $\mathcal{E}'$ is the same as $\mathcal{E}$.

Corollary 1 *If* p^n *is the set containing n instances of* $p = (t, \alpha, \gamma\imath\eta)$*, that is the parameter set of n instances of a single operator applied in parallel, and* $H: \mathbb{B}^n \times \mathbb{B}^n \to \{0,1,\dots,n\}$ *is the function determining the Hamming distance between two n bit strings. Then the value of the amplitude measurement, as calculated by* α^**, is:*

$$a^*(x, y, p^n) = a^{\frac{n - H_{x,y}}{2}} (1 - a)^{\frac{H_{x,y}}{2}} \quad (4)$$

Proof. The above equation follows from 1, where α is the same for each operator.

A formula very similar to this one previously appeared in a paper by Grover [26]. In that paper it was used in a modification of its search, that uses operators like those, given in equation 1.2, as opposed to the standard Hadamard operator.

If $Par: \mathbb{B}^n \to \mathbb{B}$ is a parity function of $x \in \mathbb{B}^n$, such that $Par(x)$ is 1, when x is odd number 1, and 0 – otherwise. Then for $x, y \in \mathbb{B}^n$ the function $Par(x \oplus y)$ is equivalent of the parity function of the n bit binary representation of $H(x, y)$.

Corollary 2 *If* $I = \{i | (x \oplus y)_i = 1,\ 0 \leq i < n\}$ are the locations at which two n bit numbers x and y differ, and $\bar{I} = \{i | i \notin I,\ 0 \leq i < n\}$ the locations where they're the same. Then for the intensional operators, i.e. where $\gamma = 0$,

$$\varepsilon^*(p^n)(x) = \begin{cases} NOT\left(\bigoplus_{i \in I} \varepsilon'(\gamma\imath\eta)_1(x_i)\right) & i = 1 \text{ and } \mathrm{Par}(\overline{x \oplus y}) = 1 \\ \bigoplus_{i \in I} \varepsilon'(\gamma\imath\eta)_0(x_i) & otherwise \end{cases} \quad (5)$$

and for the extensional operators, i.e. where $\gamma = 1$,

$$\varepsilon^*(p^n)(x) = \begin{cases} NOT\left(\bigoplus_{i \in \bar{I}} \varepsilon'(\gamma\imath\eta)_0(x_i)\right) & n = 1 \text{ and } \mathrm{Par}(x \oplus y) = 1 \\ \bigoplus_{i \in I} \varepsilon'(\gamma\imath\eta)_1(x_i) & otherwise \end{cases} \quad (6)$$

Proof. Both forms follow from theorems 1 and 6.

Now can be determined general encoding functions for n qubit gates, composed of the parallel application of a single operator to one or more qubits.

$I = \{i | (x \oplus y)_i = 1,\ 0 \leq i < n\}$ are the locations at which two n bit numbers x and y differ, and $\bar{I} = \{i | i \notin I,\ 0 \leq i < n\}$ are the locations where they're the same.

Table 8.1: Encoding function for the parallel application of n instances of $U(\alpha, \gamma\imath\eta)$

$$\varepsilon^*((a, 000)^n)_{x \oplus y} = \bigoplus_{i \in I} ID(x_i)$$

$$\varepsilon^*((a, 001)^n)_{x \oplus y} = \bigoplus_{i \in I} NOT(x_i)$$

$$\varepsilon^*((a,010)^n)_{x\oplus y} = NOT^{Par(\overline{x\oplus y})} \circ \bigoplus_{i\in I} ID(x_i)$$
$$\varepsilon^*((a,011)^n)_{x\oplus y} = NOT^{Par(\overline{x\oplus y})} \circ \bigoplus_{i\in I} NOT(x_i)$$
$$\varepsilon^*((a,100)^n)_{x\oplus y} = \bigoplus_{i\in \bar{I}} ID(x_i)$$
$$\varepsilon^*((a,101)^n)_{x\oplus y} = NOT^{Par(x\bar{y})} \circ \bigoplus_{i\in \bar{I}} ID(x_i)$$
$$\varepsilon^*((a,110)^n)_{x\oplus y} = \bigoplus_{i\in \bar{I}} NOT(x_i)$$
$$\varepsilon^*((a,111)^n)_{x\oplus y} = NOT^{Par(x\bar{y})} \circ \bigoplus_{i\in \bar{I}} NOT(x_i) \quad (7)$$

Previously was discussed that one of the impulses for creation of the phase encoding/decoding system was the observation that for a given basis $|x\rangle$, $H^{\otimes n}|x\rangle$ encodes $(x\bullet b)$MOD2 to the phase of $|b\rangle$ in the resultant super-position. It should be noted that $H^{\otimes n} = U(\{(\frac{1}{2}, 100)\}^n)$, then in the formalized system for designing of algorithmic models for quantum circuits the encoding function is written as

$$\bigoplus_{i\in \bar{I}} ID(x_i)$$

The set $\bar{I}$ represents the indexes of places where b and x are the same. It can be seen that this formulation of the phase encoding for $|b\rangle$ is equivalent to

$$x\bullet b = \sum_{i=0}^{n} x_i b_i \text{ MOD } 2$$

First, $(\sum_{i=0}^{n} x_i b_i)$ is exactly the number of places where x and b are both 1, and so $x\bullet b$ can be formulated as the even of $x\oplus b$. On the other hand, $\bigoplus_{i\in \bar{I}} ID(x_i)$ will be reduced to *ID*, applied to the subset of $\bar{I}$ in which $x_i = 1$.

If there are k elements of $\bar{I}$, meeting this condition then $\bigoplus_{i\in \bar{I}} ID(x_i) = ONE^{\otimes k}$

Thus, the phase encoding/decoding formulation of the phase of $|b\rangle$ as a boolean function of x and b is equivalent to the already published formula as bit-wise inner product. Equation 8.8 provides an alternative formulation of the phase effects of $H^{\otimes n}$, as well as a generalization to other $U^{\otimes n}$ operators.

3 Controlled Operators with Multiple Control Bits

When a n qubit operator can be thought as a boolean function over all n variables, as all except the control bits being non-essential variables. For convenience will be used $f(C)$, designating the application of the boolean function f to the control bits.

Definition 1 *A n qubit conditional operator CU is formalized by the set of the bit indexes $C \subset \{0,1,\dots,n-1\}$, the target bit index t, with $t \notin C$, control function $f: \mathbb{B}^{|C|} \to \mathbb{B}$ and formalized single qubit operators A and B such that,*

$$\langle y|CU(C,t,f,A,B)|x\rangle = \begin{cases} \langle y|A_{[t]}|x\rangle & f(C)=0 \\ \langle y|B_{[t]}|x\rangle & f(C)=1 \end{cases} \quad (8)$$

Note 1 *When $f = ID$ and $C = \{c\}$, then $CU(C,t,f,A,B) = CU(c,t,ID,A,B) = CU(CU_{[c][t]}(A,B))$ and the generalized controlled operator of equation 8.9 captures the elementary two qubit controlled operator. If $A = U(\alpha_A, p_A)$ and $B = U(\alpha_B, p_B)$, then the matrix form of $C = CU(C,t,ID,A,B)$ is defined such that,*

$$\langle y|V|x\rangle = \begin{cases} (-1)^{\mathcal{E}(p_A)_0(i_t)}\sqrt{\alpha_A} & f(C)=0 \text{ and } j=i \\ (-1)^{\mathcal{E}(p_A)_1(i_t)}\sqrt{1-\alpha_A} & f(C)=0 \text{ and } j=i\oplus 2^t \\ (-1)^{\mathcal{E}(p_B)_0(i_t)}\sqrt{\alpha_B} & f(C)=1 \text{ and } j=i \\ (-1)^{\mathcal{E}(p_B)_1(i_t)}\sqrt{1-\alpha_B} & f(C)=1 \text{ and } j=i\oplus 2^t \\ 0 & otherwise \end{cases} \quad (10)$$

This generalizes the single control bit version by allowing multiple control bits and arbitrary control functions. For boolean function $f \in \mathcal{B}^m$, where $m < n$, the generalized controlled operator can be used to represent operators $U_{\mathcal{B}^m|_n}$, as well as α degree versions of these operators. Theorem 1 describes the construction of an operator in the set $U_{\mathcal{B}^m|_n}$ whereas theorem 4.2.2, describes the construction of an α degree version of a $U_{\mathcal{B}^m|_n}$ operator.

Theorem 4 *If operator $V = CU(C,t,f,A,B)$ is such that $A \in \text{Ext}_1, B \in \text{Next}_1$ and $f \in \mathcal{B}^m$, where $m < n$. Then $V \in U_{\mathcal{B}^m|_n}$.*

Proof. If it is assumed that $A \in \text{Ext}_1$ and $B \in \text{Next}_1$, then for each n qubit basis $|x\rangle$,

$$V|x\rangle = \begin{cases} \pm|x\rangle & f(C)=0 \\ \pm|x\oplus 2^t\rangle & f(C)=1 \end{cases} = \pm|x\oplus f(C)2^t\rangle \quad (9)$$

Theorem 5 *If operator $V = CU(C,t,f,A,B)$ is such that $A = (\alpha, p_A)$ and $B = U(1-\alpha, p_B)$. Then V is an α degree $U_{\mathcal{B}^m|_n}$.*

Proof. If it is assumed that $A = (\alpha, p_A)$ *and* $B = U(1-\alpha, p_B)$. Then, for each n qubit basis $|x\rangle$,

$V|x\rangle$

$$= \begin{cases} (-1)^{\mathcal{E}(p_A)_0(x_t)}\sqrt{\alpha}\,|x\rangle + (-1)^{\mathcal{E}(p_A)_1(x_t)}\sqrt{1-\alpha}\,|x\oplus 2^t\rangle & f(C)=0 \\ (-1)^{\mathcal{E}(p_B)_0(x_t)}\sqrt{\alpha}\,|x\oplus 2^t\rangle + (-1)^{\mathcal{E}(p_B)_1(x_t)}\sqrt{1-\alpha}\,|x\rangle & f(C)=1 \end{cases}$$
$$= \pm\sqrt{\alpha}|x\oplus f(C)2^t\rangle \; + \; \pm\sqrt{1-\alpha}|x\oplus \bar{f}(C)2^t\rangle \quad (10)$$

Theorem 1 leads to general means for constructing of α degree $U_{\mathcal{B}^m|_n}$ operators from an elementary indexed operator and an operator in $U_{\mathcal{B}^m|_n}$.

Theorem 6 *If V is an n qubit $U_{\mathcal{B}^m|_n}$ operator with target bit t and $A_{[t]}$ is indexed, formalized operator with an amplitude parameter α, such that $0 < \alpha < 1$. Then the operator VA is α degree $U_{\mathcal{B}^m|_n}$ operator.*

Proof. The proof follows from theorems 4 and 5, by noting that when $V = CU(C,t,f,U,W)$, then $VA = CU(C,t,f,UA,WA)$.

Theorem 6 is important because, it captures a common occurrence in the quantum algorithms: setting the target bit of an Oracle operator to a superposition and then applying the oracle operator. In the formalized system for designing of algo-

rithmic models for quantum circuits this can be addressed in the context of an α *degree* $U_{\mathcal{B}^m|n}$ *operator.*

3 CONCLUSION

This report describes a generalized controlled operator, that uses random boolean function over k < n bits, to control the application of a single qubit operator to a target bit in the n qubit space. This form generalizes the results of the already established two qubit operator, as well as the standard Oracle operators, used frequently in the quantum algorithms. The previous development of the controlled operators as $U_{\mathcal{B}^1|n}$ operators is extended and completed in Theorem 6, that shows a method for constructing an α degree of $U_{\mathcal{B}^m|n}$ operator from the new generalization of the controlled and indexed operators.

REFERENCES

[1] Barenco, A. A universal two-bit gate for quantum computation. Proceedings of the Royal Society of London A449 (1995), 679–683.
[2] Barenco, A., Bennett, C. H., Cleve, R., DiVincenzo, D. P., Margolus, N. H., Shor, P. W., Sleator, T., Smolin, J. A., and Weinfurter, H. Elementary gates for quantum computation. Physical Review A 52, 5 (1995), 3457–3467.
[3] Bernstein, E., and Vazirani, U. Quantum complexity theory. In Proceedings of the twenty-fifth annual ACM symposium on Theory of computing (New York, NY, USA, 1993), STOC '93, ACM, pp. 11–20.
[4] Bernstein, E., and Vazirani, U. Quantum complexity theory. SIAM J. Com- put. 26, 5 (1997), 1411–1473.
[5] Nikolay Raychev. Classical simulation of quantum algorithms. In International jubilee congress (TU), 2012.
[6] Nikolay Raychev. Unitary combinations of formalized classes in qubit space. In International Journal of Scientific and Engineering Research 04/2015; 6(4):395-398. DOI: 10.14299/ijser.2015.04.003, 2015.
[7] Nikolay Raychev. Functional composition of quantum functions. In International Journal of Scientific and Engineering Research 04/2015; 6(4):413-415. DOI:10.14299/ijser.2015.04.004, 2015.
[8] Nikolay Raychev. Logical sets of quantum operators. In International Journal of Scientific and Engineering Research 04/2015; 6(4):391-394. DOI:10.14299/ijser.2015.04.002, 2015.