Quantum Gate Array Architecture Design Using Photons

Jia-Lin Chen, Edoardo Charbon, Lingli Wang*, Wen-Qing Zhao

Abstract — A modularized method is proposed to design quantum gate arrays using Destructive-CNOT gates, AND modules, and several auxiliary photons. Two quantum gate array architectures are demonstrated to implement arbitrary Boolean functions of both the canonical sum-of-product and Reed-Muller forms. In particular, our work shows how to cascade some parity check modules to generate product terms of input qubits. By using these n-input AND modules and several Destructive-CNOT gates, we can easily implement any Boolean functions.

Index Terms — Quantum gate array, input parity check, photons, SOP and RM expansions

I. INTRODUCTION

Implementing Boolean functions on a quantum computer is very important for the exploration of quantum systems. There have been efforts to find automatic ways to create efficient quantum circuits implementing Boolean functions. Iwama et al. proposed a transformation set for Boolean circuits using extra auxiliary qubits[1]. Younes and Miller[2] presented a method to convert Boolean circuits into their quantum equivalents as Reed-Muller expansions. However, the design of quantum gate array (QGA) architectures for arbitrary Boolean functions is seldom studied. The physical implementation of quantum circuits is also a field that deserves more attention.

From this physical aspect, the single-photon representation of a qubit is of special interest, due to the robustness against decoherence and ease of single-qubit operation [3, 4]. In addition, with the emergence of new solid-state arrays of single-photon detectors, the prospect of a practical implementation of quantum systems becomes increasingly realistic [5]. Two-qubit measurement, especially a measurement device that does not destroy each other’s state is a very powerful tool, since it allows entanglement distillation and efficient quantum computing based on measurements for optical quantum. Quantum computation and communication will therefore benefit greatly from non-destructive two-qubit measurements [6]. This can be achieved by quantum nondemolition detectors (QND) using cross-Kerr medium which has a Hamiltonian of the form:

\[ \hat{H}_K = \hbar \chi \hat{n}_a \hat{n}_c, \] (1)

where \( \hat{n}_a, \hat{n}_c \) are number operators for mode \( a \) and \( c \) respectively. A photon in mode \( c \) will then accumulate a phase shift \( \theta = \chi t \) with \( t \) being the interaction time that is proportional to the number of photons in mode \( a \).

A large Kerr nonlinearity with \( \theta = \pi \) is impossible, but recently a nonlinearities of magnitude \( \sim 10^{-2} \) have become available with electromagnetically induced transparency (EIT). Nemoto, Munro, et al [6][8] had proposed a method to construct CNOT and Bell state detection with such small-but-not-tiny Kerr nonlinearities.

In this paper we will demonstrate that quantum gate arrays can be designed based on parity check to implement a Boolean function as both SOP (Sum-of-Products) and RM (Reed-Muller) expansions. The advantage of our design is easy to construct with high scalability.

II. BASIC MODULES OF QUANTUM GATE ARRAY

There are two basic operations in the traditional Boolean functions: XOR operation and NAND operation. Correspondingly, there are two basic modules in the QGA architectures we propose to implement these operations.

A. Destructive-CNOT gate

The qubit we consider in this paper refers to the polarization state of photons. We define the polarization state \( |H\rangle \) as logic 0 and \( |V\rangle \) as logic 1. In addition, \( |+\rangle \) and \( |-\rangle \) states equal to \( 1/\sqrt{2}(|H\rangle + |V\rangle) \) and \( 1/\sqrt{2}(|H\rangle - |V\rangle) \) respectively.

A probabilistic logic device, referred to as a Destructive-CNOT [7], is shown in Fig. 1. It consists of two PBS and two single-photon detectors. Our goal is to flip the polarization state of the target qubit if the control qubit is \( |V\rangle \) polarized, and do nothing if it is \( |H\rangle \) polarized. Consider an initial state of the target qubit and control qubit as form \( \{a, |H\rangle + b, |V\rangle\} \otimes \{a, |H\rangle + b, |V\rangle\} \). Interacting with PBS (+/-) in the \( |+\rangle \) state, we have:

\[ |\psi\rangle_{sa} = \frac{1}{2}((a_d a_i + b_i b_d)(|++\rangle + |--\rangle) + (a_d b_i + a_i b_d)(|+-\rangle - |-+\rangle)) + \frac{1}{\sqrt{2}}|\psi\rangle_c \]

(2)

where \( |\psi\rangle_c \) represents the state that mode 3 and mode 4 do not have 1AO1 (1 and only 1) photon. Rewriting the amplitudes in mode 4 back in the HV basis lead to

\[ |\psi\rangle_{sa} \rightarrow \frac{1}{2}((a_d |a\rangle |H\rangle + b_i b_d |H\rangle + a_d b_i |V\rangle + a_i b_d |V\rangle) + (a_d b_i |V\rangle + b_i b_d |V\rangle + a_d b_i |H\rangle + a_i b_d |H\rangle)) + \frac{1}{\sqrt{2}}|\psi\rangle_c \]

(3)

This work was supported by the National Natural Science Foundation of China under Grant No.60676020 and Shanghai Pujiang Program 2008.

Jia-Lin Chen, Lingli Wang, Wen-Qing Zhao are with the State Key Laboratory of ASIC & System, Fudan University, Shanghai 201203, P. R. China (e-mail: 082052041, llwang, wqzhao@fudan.edu.cn).

Edoardo Charbon is with EPFL, 1015 Lausanne, Switzerland, (email: edoardo.charbon@epfl.ch)

* Corresponding author
So if we accept that \( D^H \) receives 1AO1 photon and \( D^V \) receives zero photon, the desired CNOT operation is completed. This coincidence occurs with a probability of \( 1/4 \).

But if we accept that \( D^V \) receives 1AO1 photon and \( D^H \) receives zero photon, we only need a simple bit-flip operation to accomplish the transmission. Thus, the total success is with a probability of \( 1/2 \). From the result we can find that this device has performed an XOR operation between target photon and control photon.

B. AND function based on Parity Check

Nemoto and Munro showed in [8] that a parity-checking device can be constructed by several cross-Kerr nonlinearities and a general homodyne-heterodyne measurement. As shown in Fig. 2, consider two polarization qubits with initial states of

\[
|\psi_1\rangle = a_1 |HH\rangle + b_1 |VV\rangle \quad \text{and} \quad |\psi_2\rangle = a_2 |HH\rangle + b_2 |VV\rangle
\]

respectively. Interacted with PBS, cross-Kerr nonlinearities and an X homodyne measurement conditions \( |\psi_1\rangle |\psi_2\rangle \) to

\[
|\psi_1\rangle = f(X, \varphi) [a_1 a_2 |HH\rangle + b_1 b_2 |VV\rangle] + f(X, \varphi \cos \theta) [a_1 b_2 e^{i\varphi(x)} |HV\rangle + a_2 b_2 e^{-i\varphi(x)} |VH\rangle]
\]

where

\[
f(X, \varphi) = \exp[-\frac{1}{4}(X - 2\varphi^2) / (2\pi)^{1/4}]
\]

and

\[
\phi(X) = \frac{\varphi x \sin \theta - \varphi^2}{\varphi^2} \sin 2\theta \mod 2\pi.
\]

We define the measured homodyne value \( X > X_0 \), where \( X_0 = \varphi(1 + \cos \theta) \), as situation-1, and \( X < X_0 \) as situation-2. If the measured value is in situation-1, then the output is \( a_1 a_2 |HH\rangle + b_1 b_2 |VV\rangle \) with a large probability. Otherwise, for situation-2 the output is \( a_1 b_2 e^{i\varphi(x)} |HV\rangle + a_2 b_2 e^{-i\varphi(x)} |VH\rangle \). The phase shift \( \phi(x) \) associated with the two-photon components depends on the outcome of the homodyne measurement. This can be corrected by applying the phase shift operation \( \exp[-i\phi(x) \hat{n}_x] \), conditional on the obtained value of \( X \) [6].

The key of this installment is that the homodyne-heterodyne measurement cannot distinguish \( +\theta \) from \( -\theta \), we can use this point to generate a function of AND operation. Suppose we want to implement a classic Boolean product term, for example \( y = x_0 x_1 \). In fact, it is equal to use a homodyne-heterodyne to measure \( x_0, x_1, y \) with the following operation:

\[
\begin{pmatrix}
1 & 1 & 1 \\
1 & e^{-i\theta} & e^{i\theta} \\
1 & e^{i\theta} & e^{-i\theta}
\end{pmatrix}
\]

This unity operator can be implemented in Fig. 3: two IPC modules are cascaded and a post operation \( T_\theta = \begin{bmatrix} 1 & 0 \\ 0 & e^{-i\theta} \end{bmatrix} \) is adopted if the measured value of the first IPC is \( X > X_0 \). \( T_\theta \) can be realized easily in Fig. 4. We name this device AND-2 because it generates the result of AND function of two inputs.

More generally, the device with \( n \) IPC modules cascading in this fashion can be called AND-\( n \). For example, AND-3 is shown in the Fig. 5.
happens, the output will be the NAND result of three input.
A bit-flip operation should be taken to get the correct result.
We can easily find that if we cascade more IPC modules, the
result is also the same, that is, we just need to check the
measured value of last module and take the corresponding
operations. As the probability of each IPC is nearly unity, the
probability of this device is about unity as well.

III. QUANTUM GATE ARRAY ARCHITECTURES
To implement a classic Boolean function, for example,
\( f = x_0x_1x_2 \), as in Fig. 5, we can let \(|\varphi_1\rangle, |\varphi_2\rangle, \) and
\(|\varphi_3\rangle\) be \( x_0, x_1 \) and \( x_2 \) respectively. \(|\varphi\rangle\) can be an arbitrary
photon with the state of \(|\varphi\rangle = a|H\rangle + b|V\rangle\), according to
the measured value, \(|\varphi\rangle\) will be collapsed to the desired
result.

Based on AND-3 module, SOP terms of three logic
variables can be implemented as shown in Fig. 6. Here we
cascade \( x_0, x_1 \) and \( x_2 \) to the next AND-3 module and as
we use Gray Code to order all minterms, each AND-3 just
needs one bit-flip gate on one input. \(|\varphi_1\rangle \sim |\varphi_8\rangle\) are eight
arbitrary photons to be used to generate all eight minterms.

Fig. 7 shows a RM pi-term implementation for a Boolean
function. We use three IPC-2 modules and one AND-3
module to generate seven RM pi-terms and use a bit-flip
gate if term 1 is needed. Three photons are for three logic
variables and four arbitrary photons are needed in this
implementation compared by 11 photons used in Fig. 6, but
at the cost of more Destructive-CNOT gates.

In both Fig. 6 and Fig. 7, the dashed lines represent the
possible light paths, while the solid lines represent the real
inputs connected to the Destructive-CNOT gates. A hollow
cycle means there is no “interconnection” between
horizontal lines and vertical lines. The interconnection
can be changed by a special program, which implies the
horizontal light path will be vertical if the circle is solid.
This can be easily achieved by using an appropriate optical
appliance to change the light path. The constraint rules of
interconnects are shown as follows:

Rule 1: Since the Destructive-CNOT gate takes an XOR
operation between two inputs, only two solid lines can
connect to any Destructive-CNOT gate. One of two inputs
comes from the SOP terms or RM pi-terms, and the other
one can be either product term or the output of preceding
gate.

Rule 2: Because photons have no fan-out, the output of
every AND module can connect one Destructive-CNOT at
most, therefore only one solid circle for one horizontal line
if we want to use it.

Let us take \( f(x_0, x_1, x_2) = x_0x_1x_2 + x_0x_2 + x_1x_2 \)
as discussed in Appendix A for an example. In Fig. 6, we make
four solid cycles to connect four SOP terms
\( x_0x_1x_2, \ x_0x_2, \ x_1x_2 \) and \( x_0x_2x_2 \) to three
Destructive-CNOT gates respectively while in Fig. 7 we link
pi-terms \( x_0x_1, \ x_1x_2 \) and \( x_2 \) to two Destructive-CNOT
gates respectively. It is obvious that the RM implementation
is better than that of SOP representation for this function.

IV. CONCLUSION
In this paper, we have shown methods to implement
Boolean functions on quantum gate array architectures.
Basic modules such as Destructive-CNOT gate and IPC are
demonstrated to implement XOR operation and AND
operation respectively. In addition, architectures for QGA
are proposed to implement Boolean functions as both
canonical SOP and RM representations. By using AND
modules based on parity check, we can achieve high success
probability using a simple structure. Finally, our method is
modularized and easy to scale. We hope this work will
motivate the study for complex quantum gates
implementation with higher success probability and less post
operations.

Appendix A: SOP and RM expansions
Any \( n \)-variable Boolean function can be expressed
canonically by SOP forms as Eq. A.1:
\[
f(x_0, x_1, \ldots, x_{n-1}) = \sum_{i=0}^{2^n-1} a_i m_i \tag{A.1}
\]
where the subscript \( i \) can also be expressed in a binary form
as \( i = (i_0, i_1, \ldots, i_{n-1}) \), “\( \sum \)” is the OR operator, SOP term \( m_i \)
can be expressed as \( m_i = \hat{x}_{i_0}\hat{x}_{i_1} \cdots \hat{x}_{i_{n-1}} \), and
\[
\hat{x}_j = \begin{cases} x_j, & i_j = 0 \\ \overline{x}_j, & i_j = 1 \end{cases}, \quad j = 0, 1, \ldots, n-1. \tag{A.2}
\]
Alternatively, the function can be expressed by the
Positive Polarity Reed-Muller (PPRM) expression as follows:
\[ f(x_0, x_1, \cdots, x_{n-1}) = \bigoplus_{j=0}^{2^n-1} b_j \pi_j \]  
(A.3)

where “\( \bigoplus \)” is the XOR operator, a RM pi-term is \( \pi_j = \bar{x}_i x_i \cdots \bar{x}_{j-1} x_{j-1} \), where

\[
\bar{x}_j = \begin{cases} 
1, & i_j = 0 \\
\bar{x}_j, & i_j = 1, \ j = 0, 1, \ldots, n-1.
\end{cases}
\]  
(A.4)

Both the SOP and RM forms have \( 2^n \) on-set and off-set product terms. For example, for a 3-variable function

\[ f(x_0, x_1, x_2) = x_0 x_1 \bar{x}_2 + x_0 x_2 + \bar{x}_1 x_2 \]  
(A.5)

Its canonical SOP form is

\[ f_{SOP}(x_0, x_1, x_2) = \bar{x}_0 \bar{x}_1 x_2 + x_0 \bar{x}_1 x_2 + x_0 x_1 \bar{x}_2 + x_0 x_1 x_2 \]  
(A.6)

with 4 on-set minterms and the equivalent PPRM form is

\[ f_{PPRM}(x_0, x_1, x_2) = x_0 x_1 \bigoplus x_1 x_2 \bigoplus x_2 \]  
(A.7)

with 3 on-set pi-terms.

**Appendix B: Quantum circuits**

The CNOT gates family, including NOT gate, Cont-NOT gate and Toffoli gate, are commonly used in quantum circuits. A general Boolean circuit can be represented as a sequence of CNOT gates[2].

For the SOP form, since the minterms are mutually exclusive, the OR operation on minterms is equivalent to the XOR operation. Therefore, it is convenient to implement Boolean functions of SOP forms in the quantum gates mentioned above. Eq.(A.6) can be represented as Fig. A.1.

![Quantum Boolean circuit in SOP form](image1)

Fig. A.1. Quantum Boolean circuit in SOP form

For the RM form, it is easier to implement it using quantum gates as suggested in [2]. Similarly, Eq.(A.7) can be represented as Fig. A.2.

![Quantum Boolean circuit in RM form](image2)

Fig. A.2. Quantum Boolean circuit in RM form

**References**


[8] Kae Nemoto, W. J. Munro, “Nearly Deterministic Linear Optical Controlled-NOT Gate” PRL 93, 250502, 2004