Functional composition of quantum functions

Nikolay Raychev

Abstract - The Boolean functions are a classic way for capturing one of the most basic computations. In this dissertation they are used as a means for specifying an information which is coded in the phase space of a quantum state. This, in turn, may serve for understanding of key examples of disturbance at quantum calculations based on the chain model. Here are examined the main properties and algebraic structures of single qubit Boolean functions and their composition, the combination of primitive Boolean functions with the operator for excluding or, ⊕ , as well as the expression of these functions and their negation as an expression, using ⊕ and B = {0, 1}, the set of the single-byte strings.

Index Terms - Boolean function, circuit, composition, encoding, gate, phase, quantum.

1 Introduction

The interpretation of the quantum computations as calculations from a physical process, and not as abstract control of symbols leads to a broader concept for computability. In accordance with the postulates of the quantum mechanics is identified also the concept for unitary transformations as the most fundamental paradigm for "physical computability". Unlike the classic symbol calculations, where each individual step of calculation may correspond to a bit string, the physical quantum calculation is in need of such a label only for the initial and final machine state. If a quantum computer is seen as probabilistic machine $\mathcal{M}$, the above mentioned requirements are equivalent to the countability of the sets for input and output, and $\mathcal{O}$. The quantum operators are further defined by the encoding of a binary information for their input data to the phase of their output data. This encoding is expressed as a Boolean function of n-bit strings, which correspond to the basic states of the n-qubit state. Furthermore, the interference between the operators is characterized in terms of these encoding functions with specific connections between the encoding functions, which produce a special case of an interference, called decoding.

2 Composition of Boolean functions

2.1 Single Qubit Boolean functions

The single qubit operators extract their encoding functions from the set of the single-byte Boolean functions.

Definition 1. *The set B^1 is equal to $\{f \,|\, f : \mathbb{B} \mapsto \mathbb{B}\}$ = {ID, NOT, ZERO, ONE}, where*

$ID: b \to b$

$NOT: b \to \bar{b}$

$ZERO: b \to 0$

$ONE: b \to 1$ *(1)*

and $\bar{b}$ indicates the negation of $b \in \mathbb{B}$

The set B^1 can be divided into two non-intersecting subsets.
BAL = {ID, NOT}
CONST = {ZERO, ONE}
The operation with these subsets, simplifies future discussions on the composition of elements of B^1 in functions for encoding the phase.

B^1 under composition

The expression of Boolean functions as a functional composition of elements of B^1 is a major part from the development and manipulation of functions for phase encoding.

Formal prerequisite 1. The set B^1 together with the binary operator for composition ∘ forms the monoid $(B^1, \circ)$.

Proof. Must be taken into account the Cayley table for $(B^1, \circ)$, given in Table 1.

Table 1: Cayley table for monoid (B^1, ∘)

∘	ID	NOT	ZERO	ONE
ID	ID	NOT	ZERO	ONE
NOT	NOT	ID	ONE	ZERO
ZERO	ZERO	ZERO	ZERO	ZERO
ONE	ONE	ONE	ONE	ONE

It is clear that *ID* is a neutral element and $(B^1, \circ)$ is closed under a composition. Since the composition as a whole is associative, $(B^1, \circ)$ is a monoid. The construction and the interaction of functions for phase encoding sometimes include a composition and therefore the knowledge of different means for simplifying the composition of the elements in B^1 is useful. The composition on the right of the CONST function *f* can be reduced to *f*, because it effectively "overshadows" the results of the function on the right. In other words, the result from the function on the right does not participate in the result from the CONST function *f*.

Formal consequence 1. *For function f* ∈ *CONST and g* ∈ B^1*, f* ∘ *g* ∈ *f*.
Proof. The proof follows from the Cayley table and the rows for CONST functions. The composition of the elements in B^1 the subset BAL is clearly defined as a group.

Formal consequence 2. *The set BAL together with the binary operator for composition* ∘ *forms the Abelian group (BAL,* ∘*).*

Proof. From the Cayley table and Formal prerequisite 1 it is clear that *(BAL,* ∘*)* is a monoid. Also the Cayley table for B^1 shows, that the elements of *BAL* are their own reverse elements and *(BAL,* ∘*)* is commutative. The composition of the

CONST elements is also clearly defined, but not that much well-structured as the composition in BAL.

Formal consequence 3. *The set CONST together with the binary operator for composition ∘ form the semigroup (CONST, ∘).*
Proof. From Formal prerequisite 2.2.1.1 and the Cayley table for $(B^1, \circ)$ follows that *(CONST, ∘)* is associative and closed under ∘.

A series of operators often have an encoding function, which effectively includes the reapplication of a Boolean function of one or more bits. That is why it is useful to be understand how and when the recomposition can be addressed or simplified.

Definition 2. *For f*∈ B^1, f^n is the iterated composition of *f.*

$$f^n = \begin{cases} ID & n = 0 \\ f \circ f^{n-1} & n > 0 \end{cases} \qquad (2)$$

An induction can be used, in order to be removed the recursion from Definition 2.

Formal prerequisite 2. *If f* ∈ B^1 *and n > 0,*

$$f^n = \begin{cases} f & f \in CONST \\ ID & f = ID \text{ } or \text{ } f = NOT \text{ } and \text{ } n \text{ } is \text{ } even \\ NOT & f = NOT \text{ } and \text{ } n \text{ } is \text{ } odd \end{cases} \qquad (3)$$

Proof. When *f = ID* , then at *n > 1* f^n= *ID*, where *ID* is the neutral element of the operator. That is why it is necessary to demonstrate that at *f = NOT* and *n > 1*, when *n* is even, then f^n= *ID* and when *n* is odd f^n= *NOT*. For *n* = 1 $NOT^{\,1} = NOT$ and then the assertions are true. For *n* = 2 *и* $NOT^{\,2} = NOT \circ NOT = ID$. If it is accepted that the Assertion is true for all *n* = *k*, then for (*k*+1) is obtained

$$NOT^{\,k+1} = NOT \circ NOT^{\,k} = \begin{cases} NOT \circ NOT & if \text{ } k \text{ } is \text{ } odd \\ NOT \circ ID & if \text{ } k \text{ } is \text{ } even \end{cases}$$

$$= \begin{cases} ID & if \text{ } k \text{ } is \text{ } odd \\ NOT & if \text{ } k \text{ } is \text{ } even \end{cases}$$

the second equality follows from the Statements for induction. .

2.2 Operators for excluding OR

It is possible to express a set B^1 with the help of the single-byte values 𝔹 and the operator for excluding or. This, in turn, will allow the combination of Boolean functions, and not only Boolean values, through the operator for excluding or. The algebraic structure of the Boolean values, combined by excluding or, is well known, but is expressed in a different way in
Formal prerequisite 3. *The set* 𝔹 *together with the operator for excluding or* ⊕ *forms the Abelian group* (𝔹, ⊕).

Proof. The Cayley table should be taken into account *(*𝔹, ⊕*).*

Table 1: Cayley table for (B, ⊕).

⊕	0	1
0	0	1
1	1	0

From the Cayley table follows, that 𝔹 is closed under ⊕ and that ⊕ is both associative and commutative. Even more, 0 is the neutral element and each element is its opposite.

The Group (𝔹, ⊕) captures the functions in the B^1.

$$ID(b) = b \oplus 0 = b$$
$$NOT(b) = b \oplus 1 = \bar{b}$$
$$ZERO(b) = b \oplus b = 0$$
$$ONE(b) = b \oplus NOT(b) = 1 \qquad (4)$$

The most common way for composing B^1 functions in phased functions is through the operator for excluding or. Equation 4 allows for expansion of ⊕ to B^1.

Definition 3. *The operator for excluding or* ⊕ *on the set* B^1 *is defined in such way that for each b* ∈ 𝔹 *and f, g* ∈ B^1

$$(f \oplus g)(b) = f(b) \oplus g(b)$$

The composition of elements from B^1, using ⊕, have a well-defined structure.

Formal prerequisite 4. *The set* B^1 *together with the operator for excluding or* ⊕ *forms the Abelian group* $(B^1, \oplus)$*.*

Proof. The Cayley table for $(B^1, \oplus)$ follows from equation 4.

Table 2.1: Cayley table for $(B^1, \oplus)$.

⊕	ID	NOT	ZERO	ONE
ID	ZERO	ONE	ID	NOT
NOT	ONE	ZERO	NOT	ID
ZERO	ID	NOT	ZERO	ONE
ONE	NOT	ID	ONE	ZERO

From table 2.2 follows that B^1 is closed under ⊕. In fact for *f, g* ∈ B^1 *is obtained*

$$f \otimes g \in \begin{cases} BAL & f \in BAL, g \in CONST \text{ } or \text{ } f \in CONST, g \in BAL \\ CONST & otherwise \end{cases} \qquad (5)$$

Furthermore, it is obtained that ⊕ is commutative, *ZERO* is the neutral element and each element from B^1 is its opposite. The associativity of ⊕ can easily be checked through Definition 3 and equation 4.

Note 4 *Equation 5 is useful in analysis of functions, defined by combination of elements from* B^1 *using the operator* ⊕*.*

Similar to ⊕ on 𝔹, the reapplication of ⊕ can be extended to B^1, as shown in definition 4 and Formal prerequisite 4.

Definition 4. *For f* ∈ B^1, $f^{\oplus n}$, is the iterated composition through ⊕ of *f.*

$$f^{\oplus n} = \begin{cases} ZERO & n = 0 \\ f \oplus f^{\oplus(n-1)} & n > 0 \end{cases} \qquad (6)$$

Formal prerequisite 5. *For f* ∈ B^1 *and n > 0,*

$$f^{\oplus n} = \begin{cases} f & if \text{ } n \text{ } is \text{ } odd \\ ZERO & if \text{ } n \text{ } is \text{ } even \end{cases} \qquad (7)$$

Proof. At n = 1 $f^{\oplus 1}$ = f and the Assertions is true. At n = 2 $f^{\oplus 2}$ = $f \oplus f$ = $ZERO.$ If it is accepted that the Assertion is true for all n = k, then for (k+1) is obtained

$$f^{\oplus(k+1)} = f \oplus f^{\oplus k} = \begin{cases} f \oplus f & if\ k\ is\ odd \\ f \oplus ZERO & if\ k\ is\ even \end{cases}$$

$$= \begin{cases} ZERO & if\ k\ is\ odd \\ f & if\ k\ is\ even \end{cases}$$

the second equality follows from the hypothesis for induction.

Functions for negation and addition in B^1

The addition or negation of functions play a key role at the approach for phase encoding and decoding to the models of disturbance.

The collating process of the function $f \in B^1$ to its addition $\bar{f}$ can be expressed in many different ways.

Formal prerequisite 6. *For each* $f \in B^1$,

$$\bar{f} = \begin{cases} f \oplus ONE & always \\ ONE \oplus f & always \\ NOT \circ f & always \\ f \circ NOT & f \in BAL \end{cases} \qquad (8)$$

Proof. The proof of the above equations follows from the Cayley tables for ⊕ and ∘, Definition 3 and Formal prerequisite 3.

It is also useful to be understood how the addition is distributed on ∘.

Formal consequence 4. *For the functions f and g in* B^1,

$$\overline{f \circ g} = \begin{cases} \bar{f} \circ g & always \\ f \circ \bar{g} & f \in BAL \end{cases} \qquad (9)$$

Proof. The distributivity of the addition of a function with ∘ derives from consequence 3.
The following Formal prerequisite is useful at the development of ideas for phase encoding and decoding, as it allows the negation of bits to be subtracted in the phase function and in this way allows for operation only with (B^1, ∘, ⊕).

Formal prerequisite 7. *If* $\bar{x}$ = $NOT(x)$ *at* $x \in \mathbb{B}$. *Then for each* $f \in B^1$, $f(\bar{x}) = (f \circ NOT)(x) = (f \oplus ONE)(x)$.

Proof. The fact that $f(\bar{x}) = f(NOT(x)) = (f \circ NOT)(x)$ *is clear.* The fact that $f(\bar{x}) = (f \oplus NOT)(x)$ *follows from the definition for NOT, given in equation 3.*

Structure of (B^1, ∘, ⊕)

The elements of phase functions, namely B^1, under composition by ∘ and ⊕ have well defined algebraic structure.

Formal prerequisite 8. *The structure* (B^1, ∘, ⊕) *is a right close ring.*

Proof. (B^1, ⊕) is Abelian group. (B^1, ∘) is a monoid and therefore semigroup. Finally, ∘ is distributed on ⊕ on the right, so that

$(a \oplus b) \circ c = (a \circ c) \oplus (b \circ c)$

In order to be seen this, are viewed three cases.

Where a, $b \in CONST$. According to equation 5 ($a \oplus b$) is in $CONST$ and therefore $(a \oplus b) \circ c = (a \oplus b) = (a \circ c) \oplus (b \circ c)$ for each $c \in B^1$ according to consequence 3.

Where a, $b \in BAL$. According to equation 5 ($a \oplus b$) is in $CONST$ and therefore $(a \oplus b) \circ c = (a \oplus b)$ according to consequence 2.2.1.2. When $c \in BAL$, then by Formal consequence 3 the composition of a or b with c or will define a and b, at $c = ID$, or will reverse their sign at $c = NOT$. It is easily proved that, in all cases, this will reduce $(a \circ c) \oplus (b \circ c)$ to $(a \oplus b)$.

When a and b are located in different subsets of B^1. Without losing the generality, it can be assumed that $a \in BAL$ and $b \in CONST$. From consequence 3 follows that $(a \circ c) \oplus (b \circ c) = (a \circ c) \oplus b$ and at $b = ZERO$, $(a \circ c) \oplus b = (a \circ c) = (a \oplus b) \circ c$. When $b = ONE$, according to consequence 2.2.2.5 $(a \circ c) \oplus b = (\bar{a} \circ c) = (a \oplus b) \circ c$.

3 Conclusion

One of the key components of this chapter, in contrast to the standard presentation of B^1, is showing the difference between the sets BAL and CONST. Although the express handling with these sets may look at this stage randomly, it is an important element of the functions for phase encoding. Often it is possible to be simplified the analysis of an encoding function or decoding connection only through the understanding of the ways in which BAL and CONST functions interact. The right close ring (B^1, ⊕) serves as a basis for construction of Boolean functions, that capture the encoding of binary information to the phase space of quantum states. The algebraic structures, presented in this chapter, explicitly emphasize the ways in which these functions can be manipulated.