Available online at www.elixirpublishers.com (Elixir International Journal)

Information Technology

Elixir Inform. Tech. 73 (2014) 25935-25937

ISSN: 2229-712X

Approaching towards Architectures for quantum computers

Imad Khan
Department of Computer Science, Bhauddin Zakriya University, Lahore.

ARTICLE INFO

Article history:
Received: 28 March 2014;
Received in revised form:
20 July 2014;
Accepted: 29 July 2014;

Keywords
Quantum,
Computers,
Relevant,
Algorithm.

ABSTRACT

In last few decades computer scientist has started taking interest in the quantum computing in order to resolve some of the problems in the recognizable classical computers. Relevant efforts have been put into the practice for two instructions: the basic assumptions behind the quantum mechanics has a residential affect, on the other hand a series of project has been used to show the series of implementation of all these algorithms. This will move the quantum from area of assumptions' to the actual level. The relationship between these two areas is generally the feasible algorithms for quantum, which settle down their model of quantum circuit.

Introduction

Most of the computational models focus on the physical inventory systems with a continues variable of different quantum systems that will be used as a computational spaces. The computational theory is not enough for a sensible computing for quantum computers.

Saunders (1999) stated that the probabilistic version of such an important language like (PGCL) to include all three levels of quantum primitives. These primitive will include Initialization, evaluation and the finalization. The language (SGL) is a communicative program that is sufficient for general quantum computers. It is an appealing technique for bringing changes in into the PGL programming into a corresponding but the reversible me.

Omar(1990) in his research has developed a technical formalism language However QSL is considered to be the understand environment. It will not be built on the top standard classical language. In this language the main purpose is to have insignificant unitary purpose. The regular operator construction and the generalization is mainly very hard to execute. For self operative determining there is no concept of parallelism.

The main features of the quantum model are discussed below:

Totality:

The language should be so influential that can articulate the quantum method. It shows that there should be some potential to code every valid algorithm of quantum. Each piece of the code can be communicated to the quantum qualitative methodology.

Conventional Extension:

A high level of traditional computational model should be added in order to gather up the quantum computation and the traditional processing method with a little effort. On the contrary the ad hoc language with tradition primitive is usually fall behind when improvement comes in normal encoding technology.

Seperability*:*

In language traditional and computing should be kept separated. This will help in avoid all calculations of traditional machine that is not required in quantum and cannot be accelerated in the quantum device. The language must be in high level coding so that quantam can get match with the way of programmers in a good way. The language should make it possible that the high level of coding can be converted in to the low level coding for quantum machines. This leads towards the compilation of codes from different quantum architectures without a significant intervention from the programmer in the process.

The first section of the paper will discuss generally about the language, together with the detail discussion about the language of the subject. Section 3 will discuss about the language rules of high quantum but normally for hardware primitive it will get reduced. Section 4 will show the clarity in hardware

QARM machine is considered to be the extension of RAM which can use quantum resources easily. It has the capability of executing all types of classical computing. This classical machine has mainly two roles: it can efficiently perform both pre and the post processing of data in easy quantum structure. It is better to keep the quantum execution time as limited as possible to avoid any kind of incoherence. Quantum system can be control by keeping it on Humiltonian which can generate needed unitary evolution for the system. It must be cleared that the quantum computing is not local but it can be possibly shared among different quantum machines to be used for quantum types of communication. These systems can be handled easily if the system get access to the hetro controlled sub systems. The quantum resources from its actually computing unit are used as an elementary terms known as qubit.

A scheme for a quantum language:

In order to perform quantum computation, the traditional core of quantum core must modify the state of element for the quantum system controls. In a proposed language these figures are treated as a unsigned integer numbers. Quantum memory in the system is organized as a physical structured array. The main objective of the quantum element is to device a linear structure. It is assume that quantum memory is originated as n array. The main purpose of this programming is to hide the details of the programmer while computation.

It is considered that no theorem can change the possibility of re devising the current generic of quantum computing. In later researches a new data type called quantum register has

Tele:
E-mail addresses: asma_logical@yahoo.com

introduced into the new type. Quantum register is basically the arbitrary collection of quantum bits. Arbitrary means the size are bounded with the amount of quantum available. Moreover different registers quantum can also overlap in a processing. So it is possible that the same address can be manipulated at different situations. A qubit will be considered free if its address is not referenced by the existing quantum register The operators of the system are basically the interface that is usually provided to the programmer for the purpose of handling all unitary transactions. The activity of converted quantum operator object into the quantum register will provide the long term transparency to the programmer. As the operator of the system is adding definition of the system through datum. Detum has the chances that it can possibly manipulate the ways easily in the processes.

Assumptions for quantum hardware:

The real construction of quantum computer is a real challenge for both of the theoretical and the experimental researcher of the field. It is currently not known that what type of the quantum mechanical method will be the most appropriate for the system. As it is also unknown that what kind of the system will be the most appropriate for quantum language. On the other side programmers are also unaware with the fact that which is the most appropriate lower architecture for the implementation of the quantum computing. The real computation is however dependent upon the capabilities of real quantum. Of hardware that is assembled with the system. Programming language should be as general as possible but on the other side it is adaptable by as many quantum machines as possible. Currently it can be implemented on local gates and apparently qubit location is needed to run the system efficiently. Hence the physical layout of the system is sometimes unknown to the high level program language this mean it is not possible to specify any closeness in the languages for the total system and the hardware language is not appropriate for this purpose in long run. The second assumptions of hardware are concerned about the implementation of parallelized gates in the system. The capability is required to perform fault tolerant quantum computing.

Quantum operator manipulation:

Quantum operators are the part of the proposed language of quantum circuits. This is mainly related to the unit transformation on the finite transformation Hilbert space. The action of quantum operator is basically to modify the quantum systems. All these unitary transform tics can be built by infinites by using matrix by acting one or two level of subsystems in computing. It is possible to use these approximate in matrices by using a finite gate method. Normally an identity operator object is constructed without using any particular parameters. Most of the time quantum operator communicates over the quantum registrar with the arbitrary size. It can be later on use for operator composition.

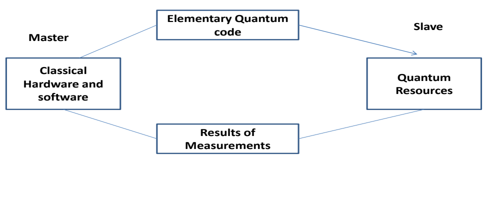

Figure1: QAR machines

Macro Quantum operators:

Macro quantum can become handy if the specific hardware is constructed for the implementation of macro quantum more efficiently. It can be gained by running the corresponding sequence for less specific low primitive. Quantum macro can become more efficient than the corresponding sequence for the less level of primitive.

Quantum operator = Fourier of transform into 7 q bits.

Line ordering of Qubit:

Operation quantum may require permutation under the registry they are operating. This can be done by exchanging the quantum bits which is referenced by q bits. The language most of the time will provide the fixed primitive of artery, this can perform certain exchanges.

Qop swap= Implementation of swap for first 5 qbits.

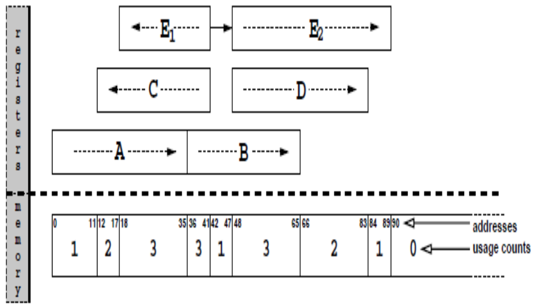

Figure 2: Organization of quantum memory without registry.

Controlled operators:

A controlled U operator is considered to be the quantum operator for CU which can easily implement the transformation of the CU|fx| y=|x U~x1….1|y, this is applied normally to the U for the second register where as first is found at the state 1….1. It will be very important high primitive quantum algorithm. This operator can be the unitary and the ad joint of some CU operator. This operator can be the unitary and has an impact on CU+ operator. Quantum operator has a some kind of controller to control such kind of activities. There is a strong need of putting ancillary qubits for their execution, which is normally controlled by the language internals transparent for the user.

For syntax three input adapters that will be strictly followed in quantum computing:

Table 1. Quantum registers, the Qreg objects (see Sect. 3.1).

	Prototype
The register class	`class Qreg;`
Type for a qubit address	`Qreg::address`
Type for a register size	`Qreg::size_type`
Type for a bit set	`Qbitset or unsigned integers`
Register constructors	`Qreg::Qreg(size_type s = 1, value v = 0);`
	`Qreg::Qreg(const Qbitset &the_bits);`
Register assignment	`void Qreg::operator=(value v) const;`
	`void Qreg::operator=(const Qbitset &the_bits) const;`
Measurement (blocking)	`Qbitset Qreg::measure(void) const;`
Register copy constructor	`Qreg::Qreg(const Qreg &a_register);`
Register destructor	`Qreg::~Qreg();`
Qubit addressing	`Qreg Qreg::operator[](address a) const;`
	`Qreg Qreg::operator()(address a, size_type s) const;`
Register concatenation	`Qreg operator&(const Qreg &r_1, const Qreg &r_2);`
	`Qreg &Qreg::operator&=(const Qreg &second_register);`
Register resizing	`Qreg &Qreg::operator+=(size_type the_size);`
	`Qreg &Qreg::operator-=(size_type the_size);`
Register size	`Qreg::size_type Qreg::size(void) const;`

```
Qop build_three_adder(int size) {
  Qop phase_shifts;
  for (int i=0; i<size; ++i)
    phase_shifts << QCondPhase(size-i, i+1).offset(i);
  Qop transform = (QFourier(size) & QSwap(size)).offset(size);
  Qop adder_2   = transform & phase_shifts & (! transform);
  Qop adder_3   = (adder_2 >> size);
  adder_3 << adder_2.split(size, size);
  return adder_3;
}
```

The main requirement of the quantum theory is to implement all the pseudo classical operators, that can transfer like Uf :|x|y> where f : Z2n— Z2m. Lecoff and Benefith (1990) in his study shown that any classical and potential quantum computing can be easily reversible. So reversible quantum can be easily converted into the quantum operators. A second remark means the function of the current context means that a mathematical mapping s must be marked in a context of mathematical functions. This definition may be more restrictive than any other usual meaning.

Conclusion:

This paper has proposed a scheme of language and a set of high level primitive programming for QAR machine. The high level primitive have been studied in detail for efficient and effective implementation.